Secure Authentication Procedures Based on Timed Passwords, Honeypots, Honeywords and Multi-Factor Techniques

By

Omar Zeyad Akif

Supervised by
Professor Hamed Al-Raweshidy

A Thesis Submitted

in Partial Fulfilment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Department of Electronic and Computer Engineering

College of Engineering, Design and Physical Sciences

BRUNEL UNIVERSITY LONDON

London, United Kingdom

December 2017

Abstract

A time-based password generating technique has been adopted and applied to protect sensitive datasets as the first technique used in this thesis. It specifically mitigates attacks and threats by adding time as a part of the password, which is generated using the shift-key. This in turn raises the possible combinations for the password and enhances the system's security. The Password Quality Indicator (PQI) was implemented to evaluate security improvement. Results showed that contemporary password techniques were up to 200% more secure than the traditional methods.

The second method, 'honeypot', is based on web-session management. The authentication process is triggered if the web-session is initiated correctly when the first webpage is requested; legitimate users must perform the correct session through a precise links' sequence to be compatible with the session management that has been saved in the server side. The honeypot will present a sequence of links to lure the attacker into performing the authentication procedure directly from the login box. When compared to conventional methods, it was found that using the new method has improved user security by 200%.

Additionally, a multi-factor authentication approach was tested, where combination of the timing password and the honeypot techniques was used. The outcomes were calculated and the results demonstrated that the passwords' strength was enhanced when using and increasing the number of links and the quantity of dwell time periods as a result of probabilities and complication. This approach yielded passwords that are 300% more secure than traditional methods would generate.

Finally, a honeywords-generation method (decoy passwords) was also applied to detect attacks against the databases of hashed passwords. With an aim of achieving flatness, the original password for each user account was stored with many honeywords in order to confuse and mislead cyber-attackers. This technique relies on the abnormal generation method to achieve flatness among real password. A survey involving 820 participants was conducted to quantify how many users were able to recognise the real password among several honeywords. The results have shown that the new generation method was an improvement on traditional methods by 89.634% and attained sufficient flatness to confuse the attackers.

Dedication

I dedicate my thesis to:

My **Dad**: You encouraged me to take on adventures, such as my PhD.

My **Mom**: I learned from you that where there is a will, there is a way.

In loving memory my **Uncle**: I still cherish in my heart the love and laughter we shared.

My **Brothers**: you are the greatest backers that I can ever rely on.

Ann: You are the best gift God has given me.

My **Teachers**: thank you for making my world a better place.

Declaration

I declare that this thesis is my own work and is submitted for the first time to the Post-Graduate Research Office. The study was originated, composed and reviewed by myself and my supervisors in the Department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences, Brunel University London UK. All the information derived from other works has been properly referenced and acknowledged.

Omar Zeyad Akif
December 2017

Acknowledgements

To begin with, I wish to thank my supervisor, Professor Hamed Al-Raweshidy for the constant support during my PhD study. Thank you for your patient approach and for sharing your immense knowledge. Moreover, I will never forget your kindness and consideration in my times of need. I could not have hoped for a better mentor and friend for my PhD study.

Professor Geoff J. Rodgers is another person who this PhD might not have been successful without. His unconditional support and guidance helped me overcome most, if not all, the difficulties I faced with the research aspect of my study. I cannot think of enough ways to express my gratitude for you. I must also extend my appreciation to my second supervisor, Professor Maozhen Li for his words of encouragement and support.

I would also like to thank my sponsor, the Iraqi Ministry of Higher Education and Scientific Research, as represented by the Iraqi Cultural Attache in London, for giving me this opportunity to complete my study.

Additionally, I must express my sincere gratitude to Dr Hussein Al-Ali for his on-going support, encouragement and all the fun time we spent together. In you, I have gained a friend for life; one who I can rely on in difficulties and share my happy moments with. My gratitude must also be given to Dr Firas Abdul-Hameed who was always willing to help and offer his best advice.

Last but not least, a heartfelt thanks to my really supportive and lively friends: Bilal, Laith, Shireen, Raad, Younis and Yousif who made my time at university enjoyable and special. I will always remain indebted for their understanding and support during the times when I really needed them.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AKE	Authenticated Key Exchange
BTPS	Biometric Template Protection Schemes
CKD	Continuous Keystroke Dynamics
CLR-eCK	Challenge-dependent Leakage Resistant
CP	Click Patterns
CY	Cloud Honey
DGAs	Domain Generation Algorithms
DNA	Deoxyribonucleic acid
DoS	Denial of Service
DPE	Difference Percentage Equation
DWPT	Discrete Wavelet Packet Transformation
ECC	Elliptic Curve Cryptography
EER	Equal Error Rate
GUI	Graphical User Interface
HIHAT	High-Interaction Honeypot Analysis Toolkit
HTTPS	Hyper Text Transfer Protocol Secure
ICT	Information and Communication Technologies
IDS	Intrusion Detection Systems
KD	Keystroke dynamics
LRPE	Leakage-Resilient Password Entry
MFA	Multi-Factor Authentication
MINIBAG	mini-batch bagging
OTP	One-Time Password
PIN	Personal Identification Numbers
PKI	Public Key Infrastructure
PQI	Password Quality Indicator
QIM	Quantisation Index Modulation
SET	Secure Electronic Transaction
SKD	Static Keystroke Dynamics
SMS	Short Message Service
SMshing	Phishing Through SMS
SPHF	Smooth Projective Hash Functions
SQL	Structured Query Language
SSL	Secure Socket Layer
TLS	Transport Layer Security
TMIS	Telecare Medical Information System
TS	User Time Signatures
URL	Uniform Resource Locator
XSS	Cross-Site Scripting

Chapter One: Overview

1.1. Introduction

There are several security issues that present significant challenges in any computer system, including privacy, authorisation, verification, access control, configuration of the system, the storage of information, and management [1]. When a user needs to have network access, with most web applications these security issues are pertinent and authentication is required to gain access [2]. Authentication is the process of verifying legitimate users before obtaining access to secure resources [3]. The traditional user authentication can be defined as a combination of username and password. Username refers to the identity of the user, by which the user is identified, while the password is the information that has been related with it that proves the users identity and the two are used together to obtain permission for access, for without them, it is not granted for a secure system [4]. There are several mechanisms of authentication and each has different strengths, but all have weaknesses too depending on the context their usage. The most popular authentication types come under three main categories: knowledge-based (for example, passwords); token-based (for example, credit cards); and biometric-based (for example, fingerprint) as well as their combinations [5].

The traditional way of authentication is a text-based password, which is the main knowledge-based authentication method [6]. Even after many attempts to exchange it for other alternatives, the password is still the most popular user authentication technique deployed today [7]. A good password has to have two features: the user can remember it, and it is difficult to guess [8]. Unfortunately, these two work against each other, such that a password that is easy to remember is generally short and hence, easy to surmise [9]. Textual passwords were recognised as a point of weakness in information system security by Morris and Thompson in 1979. They found that most passwords were weak, in fact, representing a percentage of 86% of the total. Most passwords were too short, containing just lowercase letters, only digits, or a combination of these and so, they could easily found in dictionaries [10].

Owing to the fast development of wireless communication and information technologies, in particular, the rapid growth of the Internet, the amount data passing

through networks is now vast and exponentially on the increase [11][12]. This, along with the speed of creating these data has led to a new concept called "Big Data", which are data sets with sizes beyond the ability of commonly used software tools for capturing, creating, managing and data processing within a "tolerable elapsed time" [13]. While many techniques are used to secure passwords, most of them are insufficient in the face of attackers' tools [14]. However, there are many technologies that have emerged to make the password stronger. Honeypots have been developed as good defenders and for detecting nature of the attacks. However, they occupy only a minor position among other security technologies, such as firewalls and Intrusion Detection Systems IDS [15]. This is surprising given that they can provide unique attack information, unavailable by any other means [16]. Specifically, a honeypot is a computer security mechanism set to detect, deflect and/or in some way, attract attempts at unauthorised use of information systems [17]. Honeywords is an approach to improving the password technology, involving techniques that make them more difficult to crack by an adversary, even if the password file has been compromised [18]. Moreover, they can act as decoy passwords, having the function of being a proactive tool for detecting the attacks based on compromised password files [19].

Finally, the primary goal of the adversary is compromising and collecting the passwords of the system so as to get access to a sensitive dataset [20]. Hence, passwords should be strong against attacks, which include dictionary attacks, guessing attacks and brute-force attacks. To achieve this, four authentication techniques are applied in this thesis: behavioural timing passwords, honeypots, multi-factors, and honeywords respectively. In sum, these techniques are proposed for addressing the problems of password attacks and will be applied to a sensitive dataset.

1.2. Password Security and Usability

User authentication is the first step of communication for users of security sensitive systems, and alphanumeric passwords are most popular for fulfilling this purpose. In the past, the authentication systems were for security purpose and adversaries largely unable to develop patterns for making attacks and systems were generally stand alone. However, over time adversaries have developed increasingly sophisticated patterns for launching attacks and given the interconnected online world

we now live in they can cause extreme damage if they manage to crack authentication and thus, gain access to secure systems. [21].

Clearly, there are public standards for defining product usability; however, none such exist for password usability. Usability can be defined as the level of using any product by particular users, with the aim being to achieve pre-specified goals in relation to effectiveness, efficiency and satisfaction, in line with the desired usage [22]. The general weakness of passwords is well known, but regardless of this, there are many website passwords of users that are very weak and can be broken in a short time by using software widely available for cracking them [23]

There are several issues that need to be considered when assessing password usability:

- **Time for the password creation:** The time that passes from loading the password creating webpage and submitting one;
- **Creation attempts:** The number of times the user has to create a password before one is accepted by the system;
- **Difficulties of password login:** The number of times the user has to enter the password before access owing to the password needing be remembered correctly;
- **Password storing:** The way in which the password is stored (for example, writing it down or saving it in the browser) [24].

1.3. Passwords in the Wild

A number of concerns have been raised about real-life passwords and their application in the current environment; hence this matter has been referred to by some as “security in the wild”. This refers to how security is important to people in their everyday daily lives and hence, they should address the following question, ‘‘is this system secure enough for what I want to do?’’ [25]. Some features of passwords have been changed in recent years, such that those selected have tended to be longer than before and people have been urged to ensure that they contain a combination of uppercase and lower case letters as well as special characters and digits [26].

For any verification or registration process over the Internet, the Transport Layer Security (TLS) is used, which is an encrypted channel and hence, passwords sent through it are more secure than unencrypted ones. However, TLS is risky to use, as it is vulnerable to a range of attacks, including phishing and man in the middle attacks. Consequently, Hyper Text Transfer Protocol Secure (HTTPS) is usually used in the connection (e.g. HTTP will be over TLS to obtain an encrypted connection) to send sensitive information (for example password) over the Internet. On the authentication server side, the TLS protocol depends on whether the user is really trusted by the server (server certificate), which if so, will allow for access to the Uniform Resource Locator (URL) and be subsequently trusted with the Public Key Infrastructure (PKI). In sum, the security level of passwords transported over the Internet rely on the secure channel used for transportation, which depends on TLS [27].

Additionally, there are other services that play a vital role in increasing the security of Internet traffic for various applications, including Secure Socket Layer (SSL) and Secure Electronic Transaction (SET). However, the most commonly used as a standard in web data communication is SSL, which is deployed to provide high secure access to websites. A combination of both public key and private key encryption methods can be used with SSL, which allows for the respective advantages of both technologies to be exploited [28].

1.3.1. Enhancing the Login Process

There are numerous services provided to users through public networks and despite their benefits, security remains the main concern given their public nature [29]. For, given this open environment an adversary can easily launch various types of attacks such as eavesdropping, intercepting and modification of the channel. Hence, providing a secure channel for transforming the messages and sensitive information is required to protect the information [30]. People usually create their passwords either from a dictionary or their memories so as to give them the ability to remember them and most users for convenience have the same password for several accounts [31].

1.4. User-Centred Security

In the field of security, the user plays a central role in the design and evaluation of mechanisms for protecting data. Recently, Bruce Schneier, a cryptographer, was quoted in his book as registering a deep concern about security, when he said: "If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology" [32]. When devices are being used in over networks, there are programs that help ensure that information is kept secure. These programs are aimed at making people see the importance of understanding how security works when devices are working, especially when they are connected to networks [33].

1.5. Web-Session Management

Several responses and requests are involved as part of a *web session*, whereby a web server and a browser exchange information on a semi-permanent basis. Generally, stateful web session areas can be expected to be bound to a *cookie*, which in turn is retained in the user's web browser. Valid and accurate information is provided when the user provides authentication to access a specific website through a particular web page by means of, for instance, a password and a username pair, then the server will generate a new *cookie*, sending it to the web browser. In all following requests that originate from the browser, this cookie is included and is used as proof of an established, password-authenticated session. Within this process, the cookie opens the possibility of a hijack, as it acts as a password for requests made to the webserver. That is, its value may be hijacked and utilised in place of the user consequently, with there being no need to compromise the server, or the low-level network connection [34].

Web applications have become a vital part of the web and often contain sensitive data that has to be protected and secured properly [35]. Web application attacks can involve security misconfigurations, broken authentication or session management, among other issues. Moreover, some of the most common and pernicious attacks involve improper validation or the inputting of malicious text or domain-specific code. Attacks of this type include Cross-Site Scripting (XSS), and

SQL injection attacks, among others [36]. Session management is usually performed through the secured cryptographic, cookies-based, ticket technique [37].

1.5.1. Web Threats

Two main categories of attack are handled and prevented by web security services and solutions, traditionally speaking. These two categories are *network attackers* and *web attackers*. At least one server is controlled by the web attacker when it responds to an HTTP(S) request as part of a web attack. This contains malicious, arbitrary information and data selected by the attacker in question. Web attack capacities are extended through network attacks owing to the capability of intercepting and distinguishing traffic between the endpoints of the networks concerned. These attacks are not able to break down cryptography; however, all HTTP traffic may be forged, corrupted and inspected during such an attack. Web attacks are probably easier and simpler to undertake compared to a network attack. The latter can wreak considerable damage and have negative repercussions, for through these attacks, the attacker is given complete control over HTTP-served web pages [38].

1.6. Keystroke Authentication

By recognising the rhythm of the users' typing and key inputs, keystroke dynamics can be deployed to recognise individual people, thereby making the password-username logging on mechanism stronger and harder to break. That is, the rhythm through which the person inputs the keys on the PC when entering the password (this can also be used regarding username inputs) is taken into account, thus the rhythm and the username/password both have to be correct [39].

Since the keyboard is still utilised by the user following the logging on process, for instance, when using social media, for web browsing and for document writing etc., ongoing authentication can be maintained through the keystroke dynamic procedure. The primary difference between the logging-on procedure use of keystroke dynamics (this process is referred to as SKD: static keystroke dynamics) and ongoing authentication processes (CKD: continuous keystroke dynamics) is that the former

involves fixed typing information, whilst for the latter, there is no fixed information [40].

Ease-of-use can be combined with username-password procedures through employment of keystroke dynamics, which introduces a further element of security regarding the soundness and trustworthiness of biometric means and processes. In one study, the extant literature on keystroke dynamics and shift-key patterns was reviewed. The authors hoped that understanding these patterns could deliver credential hardening of user-passwords. The findings were validated with the use of a keystroke dynamics dataset, including nearly nine thousand different input sequences, a number greater than that used in related research and literature [41].

1.7. Passwords Based on User Behaviours

It is the reality that the password has become one of the weakest links in the user security chain. Moreover, trying to meet the requirements of some security systems has become very hard, especially in terms of asking users to create complex long passwords because of their inability to memorise them [42]. Hence, password vulnerability still persists and sometimes working to increase security can lead to the opposite effect being the outcome owing to user behaviour. For instance, mandating users to change their passwords periodically will lead to them either writing the passwords down for easy access or forgetting them due to the number of times they have been created and changed [43]. Often, the user is habitually impatient, not being prepared to undertake the various optional actions available for protecting data. For instance, tasks such as reading a manual and checking the safety of input devices might be lead users to be impatient are seen by many as tedious and as consuming valuable time [44]. Various characteristics relating to the user, such as location and motion have been deployed to increase system security. However, user behaviour is unpredictably changeable, which has made it even more challenging to develop or improve the systems that are based on it [45]. In sum, users, whilst wanting their systems to be secure, do not want to have to go through long drawn out procedures every time they log on to their devices. In other words, typical user behaviour works in the opposition direction to the level of security.

1.8. Problem Statement and Motivations

Access control techniques that have been widely applied, to gain access to secure systems include username-passwords, tokens, and biometrics. Their main purpose is to protect these systems from adversarial users gaining access. Nevertheless, the password is still the extensively used method of the available authentication techniques for securing authorised access to secure systems. However, as aforementioned, many passwords are weak and for the reasons previously discussed, it is difficult to get users to create much stronger ones. User behaviour is a serious problem associated with the weakness of passwords, whereby most users create short ones and they traditionally use a dictionary or a memorable personal item. Another reason why passwords are weak is because many people use them for several accounts and hence, they can be used as gateways to multiple accounts. If users insist on this behaviour, then they need to be advised on how to make their password stronger by including special characters, numbers along with upper and lower case letters in it. However, many people find it difficult to remember complex passwords of this nature. In sum, users prefer to create an easy password rather than a strong one due to memory limitations.

Leaking of passwords remains one of the major threats in terms of unauthorised access to sensitive information. Moreover, password attacks have been on the increase in recent years. Dictionary attacks, brute-force attacks, man-in-the-middle attacks and guessing attacks are examples of attacks. Whilst many researchers have proposed solutions to threats against password-based user authentication, there is no solution that has been widely adopted. One major challenge is that user' need security systems that is both secure and practicable. Unfortunately, it is difficult to incorporate these two features into systems to increase security and at the same time be popular with users. By adding other elements to the traditional password that do not impose complexity for the user, sensitive datasets can be better protected, which is aim of this thesis. In sum, this work involves applying different techniques for improving password security and hence, making them more resilient to attacks.

1.9. Aims and Objectives

The main aim of this thesis is to improve the traditional method of authentication, namely, the text-based password, thereby mitigating the influence of attacks. This will be achieved by making adversary detection more effective through the addition of more elements as part of the passwords. To this end, this research is focused on new techniques developed based on text-based password authentication, but different to traditional password creation, which involves the following.

1- Review and measure the behaviours of users in relation to the authentication process in order to identify their strengths and weaknesses. To achieve this, two basic surveys were carried out. They provide a comprehensive review of the respondents' behaviour, and their thought processes when creating their passwords. In addition, the users' behaviour when using the Shift-key to type in their password during the login process is investigated. The second survey was carried out to measure the strength of a proposed Honeywords generation algorithm and whether the participants have the ability to guess the real password among the fake Honeywords.

2- Developing mathematical models derived from another model, which are used to measure the strength of traditional passwords. These new models can be used generally to measure the password strength for any special cases based on character have special characteristics. Moreover, another mathematical model is to be used to measure the strength of honeypots based on session management. Subsequently, the outcomes of comparative results will be presented.

3- Creating a new password with new mechanisms to increase its strength and demonstrating that it can improve resilience against password attacks over traditional methods.

4- It is generally believed that it is hard to get people to change their behaviour. However, in the current research evidence will be provided that demonstrates how users can be instructed in modifying their behaviour so as to make the devices and systems they work with more secure.

5- Developing a special program in C# that is able to measure the time period for shift-key usage. Usually, the Shift-key and Ctrl-key are neglected when used

in traditional keystroke methods as they require an additional key to be pressed in order to execute a given action.

6- Increasing the possible combinations for the password. The traditional password can involve one or more of the categories of uppercase letters, lower case letters, special characters and digits. However, new techniques will be introduced to increase the number of permutations, thus making the password more difficult to identify by an adversary.

1.10. Research Outcomes and Contributions

The main contributions of this thesis can be summarised as follows:

- Studying the behaviours of users according to two aspects. The first provides understanding whether the users type uppercase letters by pressing the Caps-Lock key or by using the Shift-key and if do the latter, ascertaining how long they hold it down before releasing it. This survey will be used with the timed password to determine the start of the first dwell time period, while the second enquiry relates to defining what users think about passwords created by others. Furthermore, this survey will be used in the honeywords technique to measure the flatness of the new honeywords generation method. The data have been collected for these investigations through two surveys.
- By including timing in the password entry process, this calls for a change in the behaviour of users, which is in contrast to systems learning, where a behaviour profile is created for each user, such as in dynamic keystroke authentication. The main outcome was increased the possible combinations probabilities of the passwords.
- Deriving new formulas to develop three mathematical models that are applied to obtain the results. In addition, these formulas can be used in any special cases for measuring the strength of a password. The traditional password measurement has been used in the timed password, honeypots and multi-factor techniques to compare the traditional results with a new mathematical model for each technique aforementioned.
- A new honeywords generation method has been developed to confuse and lure the adversary, which delivers an improved detection system. Furthermore,

mixed techniques are used to generate these honeywords based on the behaviour of users when they create their passwords. For instance, a dictionary attack has been used in the new generation method for protection, rather than being deployed to compromise the password.

- A new high-interaction honeypot technique based on web-session management with webpage links is provided. These links will be ordered, and each user has a unique sequence that will be part of their password. The main achievements were improved the authentication technique and detected the attacks.
- Using a multi-factors method that involves the mixing of the two new techniques to create a form of authentication. Specifically, both the behavioural timing aspect and honeypots based on web-session management will become part of a password. The benefits of the both techniques have been gained especially with the password strength was increased. Finally, a mathematical model has been developed for comparison purposes.

1.11. Thesis Overview

In this chapter an introduction has been provided and then there was a general discussion about authentication techniques and passwords. Moreover, the motivations as well as the contributions have also been explained in this chapter. The rest of thesis is organised as follows:

- **Chapter Two:** In this chapter, a literature review and the background regarding authentication techniques and their characteristics are provided. The popular authentication threats and attacks are listed and discussed in this chapter. Furthermore, the current authentication schemes are evaluated, including honeypots, keystrokes, biometrics, honeywords, and multi-factor authentication techniques, by drawing on recent research and considering their applications.
- **Chapter Three:** The first contribution is presented in this chapter, which is a timed password based on changing user behaviours. This chapter begins with a description of the methodology for the new technique and then, the Petri nets model is explained, being accompanied by a Petri nets diagram.

Subsequently, password strength measurements for both the traditional method and the new password technique involving a mathematical model are calculated and discussed. The new password method's strength measurements are divided into three scenarios: fixed positions, two variant positions, and three variant positions, for the timed keystrokes. The results are presented and analysed, then the capacity for successful password attacks on this new technique is considered too. Finally, a summary of this chapter is provided. The results of this chapter have been published in SAI Intelligent Systems Conference (IntelliSys), IEEE 2017, London.

- **Chapter Four:** In this chapter, password authentication based on honeypot web-session management is presented as the second contribution. The chapter starts with an explanation of the methodology of the new technique, and then the Petri nets process for this method is explained along with a diagram being given to provide clarity. Furthermore, the mathematical model is explained for the traditional password measurement and that of the new technique. Additionally, the results of the new technique are illustrated after the mathematical models have been implemented, and subsequently analysed. Password attacks on this new technique are investigated and finally, there is a chapter summary.
- **Chapter Five:** A multi-factor authentication approach is presented in this chapter as the third contribution of this thesis. Initially, the methodology is described and then the mathematical models for measuring the strength of the traditional password and the new password technique are explained. The results are presented in graphical form and subsequently analysed. Finally, there is a chapter summary.
- **Chapter Six:** In this chapter, the fourth contribution is presented: a new honeywords generation method based on user behaviour to achieve flatness, as explained. The key issues covered relating to the honeywords technique are the honeychecker, review of honeywords, and their limitations. The traditional honeywords generation methods are explained along with their problems and then, the new honeywords generation technique is described in detail. Furthermore, attacks on the new honeywords generation technique are discussed, with the testbed being subsequently explained and the results

of the survey being presented and analysed. Finally, this chapter will be summarised. The results of this chapter have been published in FTC Future Technology Conference, IEEE 2017, Vancouver, Canada.

- **Chapter Seven:** In this chapter, the thesis conclusions are presented and suggestions for future research avenues put forward.

Chapter Two: Literature Survey

2.1. Introduction

In the previous chapter, the importance of user authentication and Internet security has been explained. In this chapter, the main significant threats and attacks on passwords will be introduced. Moreover, some advantages and disadvantages of some related works' authentication techniques will be considered in this chapter too, especially those related to password based-text.

The main concern with any data is their security, for if they are breached this will compromise their confidentiality and integrity. Hence, authentication of the user is of primary importance. The most commonly used way of ensuring that a user is legitimate and thus can have access, is the password. Whilst the classical login/password-based systems have been simple to implement, they are vulnerable to attacks. Currently, the number and types of attacks for breaking passwords are increasing, hence a new technique that has the ability to hold-out against these is needed, particularly when the password provides access to important sensitive datasets [46]. Some people believe that any password is weak, but they prefer to use them in preference to any other authentication procedure, because they are easy to use by all. Unfortunately, many users make their passwords very simple, which make it easy for attackers to steal or break them [47].

2.2. Popular Authentication-Related Threats

There are two types of the security attacks: human-based and technology-based. With the human-based attacks, an adversary will interact with the target (victim), who has valuable information, such as through social engineering attacks. On the other hand, regarding technology–based attacks, an adversary can gain access to secret information by using non-interactive means, for instance, a phishing email [48].

Password attacking involves different character combinations being tried until a match with the correct password is found. There are many types of password attacks, some of the most important of which are described next [49]. In this section,

the popular threats are discussed, and then they will be performed on the new authentications techniques that have been proposed in this thesis to explain how they can hold out against these attacks.

2.2.1. Brute-Force Attack

It is not easy to protect against brute-force attack, which is implemented on a great number of combinations using a trial-and-error process. In this type of attack, all the possible combinations of the password are applied to break it. It can also be applied to crack encrypted passwords wherever they are saved in the form of encrypted text [50]. In contrast to the dictionary attack (see below), it even focuses on unknown key combinations. When the key size is not large, passwords are able to be broken easily. Brute-force attack, however, is very time consuming when faced with a strong password or a large number of key combinations. The implementation of such attacks is via either a computer program or it is ready-made. There has to be a high configuration of the computer in order for a brute-force attack to be fast and effective [51].

2.2.2. Dictionary Attack

A dictionary attack is carried out on verification data by trying out every word in the dictionary. This type of attack is targeted at sites with a high probability of success, such as those with weak passwords or with only a few key combination numbers. This attack is quicker than one of brute force and is more successful when a weak, commonly used or short password is used. However, when the password contains special characters, it becomes more complex to hack into web sites [52].

Moreover, a dictionary attack is the commonest method for hacking password hashes. There is a wide use of dictionary words by attackers aimed at analysing passwords, which can quickly crack hashes [53]. That is, this involves using either very big dictionary files that contain potential passwords in their millions or a combination of words in the dictionary. It works by calculating the hash value of every dictionary file password and using it in comparison with the hash value input of any unknown one [54]. When a match is discovered between the dictionary text and

the hash value, the input password will be the same. Whilst this is a faster than other methods of attack, it is not as successful. It has a generally good success rate when used for common passwords, which is why numerous passwords are continually cracked using this method by attackers [55].

2.2.3. Phishing Attacks

This is where an attacker attempts to retrieve legitimate users' confidential and sensitive credentials fraudulently by mimicking electronic communications from a trustworthy or public organization in an automated fashion [56]. The aim of phishing is to steal sensitive information, such as online banking passwords and credit card information from Internet users. These attacks use a combination of social engineering and technical spoofing techniques that persuade users into giving away sensitive information that the attacker then uses to make a financial profit [57].

2.2.4. Shoulder-Surfing Attack

Shoulder-surfing is when someone sees the inputted password over the shoulder of a person. There have been many attempts to resist this type of attack, including eye-gaze entry, tactile/haptic (vibration) patterns and digitally signing in with pressure on a touch screen [58].

2.2.5. Guessing Attacks

Knowledge-based verification is where users with weak passwords that attackers can simply guess at, are at risk of online attacks by someone guessing the password. It is also possible to predict the password when offline, usually when attackers have database access [59]. Some websites put measures in place when successive password entry attempts are unsuccessful, such as disabling any further attempts or raising delay times in the system's response [60]. Whilst these safety measures are useful for reducing the incidence of unauthorised password guessing, they can alienate genuine users who might have simply forgotten their password. In addition, these approaches could cause a denial of service (DoS) attack on users'

accounts; barring them from entry. In order to remember passwords, users choose short and memorable ones or replicate or modify an existing password [61].

2.2.6. Social Engineering Attacks

The efficiency of protecting sensitive information has been increased, but people still represent the weakness element regarding its security. A social engineering attack targets this weakness, whereby an adversary deploys several techniques of manipulation to obtain the sensitive information [62] as follows:

- Physical approaches: The information will be collected about the future victim, based on physical action taken by the adversary and this information can take various forms, for instance, valid credentials of the victim, date of birth or even the password written on a post-it note [63].
- Social approaches: The attacker uses sociopsychological techniques and the main principle that he/she relies on it is curiosity. This can be achieved, for instance, by phishing the victim through using any social media platform [64].
- Reverse social engineering: In this attacks the attacker will lure the victim by make him/her believe they are in a trustworthy environment. The aim of this attack is to make the victims that is possible to attack will approach him, for instance to ask for help. This type of attack comprises of three main parts: sabotage, advertising and assisting. Firstly, the company's computers will be faced a problem, then the second stage will come which is the attacker advertise to solve the problem, then the final stage will come which is the victim will ask to get the help by attacker.
- Technical approaches: There is no doubt that the most password attacks are carried out via Internet. Moreover, most users are using one password for several accounts and the most passwords are simple to be easy remembered. Thus, gathering the information that are related to the victim via searching in some social media programs or using search engine to gather some information related to a victim and potential to be a password is currently using.

- Socio-technical approaches: This type of social engineering attacks are often used the combination of several or all the methods that have been discussed above [65].

2.2.7. Eavesdropping

Eavesdropping can be defined as when the attacker deploys spying tools of in a specific network to hack into the communication channel [66]. The main goal of eavesdropping is so the adversary can capture the behaviour of the network traffic as well as getting the network map. It is a very risky threat, which can lead to the collapse (break down) of integrity and confidentiality, thereby causing serious economic damage [67]. Intruders are increasingly eavesdropping on communications between legitimate users and servers as well as masquerading as authorised users or remote servers so as to be able to steal sensitive information [68].

2.2.8. Denial of Service Attack

A denial of service attack (DoS) gives an adversary access to the network services, thus preventing the authorised users from doing so [69]. Once in the system, he/she will use intensive computation tasks against the victim, thereby exploiting system vulnerability. Another method is flooding the system with a huge amount of useless packets and as a consequence, the victim can be forced out of service for from a few minutes to several days [70].

2.2.9. Rainbow Table

Hashing the plaintext or password is a one-way function, which makes it hard to find the required password [71]. However, rainbow tables, which are massive tables filled with hash values, can be used to find a required password, whereby a hacker employs them to find it by reversing the hashing function. Despite a rainbow table taking up a lot of storage when holding it, attackers can usually crack the password in a shorter amount of time than when applying the brute force technique [72].

2.3. Evaluation of Authentication Schemes

User authentication is a service provided to the user to guarantee that he/she has permission to access system in question [73]. Authentication is usually performed based on the following:

1- Something you know (for example, password);
2- Something you are (for example, fingerprint);
3- Something you have (for example, hardware token) [74].

The first type is the most popular in computer systems, whereby the user inputs his/her password by typing it on a keyboard [75]. Whilst there are some other authentication methods, these are much less commonly used than the password. When two or more methods are combined together, this will generate a new case called "multi-factor authentication" [76].

2.3.1. Knowledge-Based Authentication

Knowledge-based authentication is based on some secret information, which is not shared with people and hence, only the user has access to it [77]. The passwords that are used by users are vulnerable to attack due to some issues relating to the password itself and /or human behaviour. For instance, the limitations of the user's memory, whereby many cannot remember strong passwords (meaningless words) and hence, use an easy one that is remember, but also simple to guess [78]. An acceptable password should be one that takes into account end user memory limitations as well as the security requirements of the system [79]. Minimising the user's mnemonic load, but not using an obvious password linked to the easily accessed attributes of the user, can deliver stronger authentication secrets. For example, using pre-existing knowledge that is well-known in the user's memory rather than asking him/her to memorise a random alphanumeric string can be effective. That is, with this method, the user can recall the relevant information easily as it is intrinsic to them, whereas the intruder will not have access to it [80].

An additional security layer can be added to protect passwords against phishing or malware. This single password authentication has been performed on

cloud storage and is based on optimising either the online service performance, or the storage server [81].

Password recovery of public encryption application is a significant practical advancement in relation to retrieving a password which the user has forgotten as well as supporting the implementation of data forensics. One method to enhance password recovery is a technique based on a rainbow table attack, which has the capacity to crack long passwords. Specifically, two methods are combined together, namely, dictionary generator and a rainbow table attack to generate an effective and smart method to this end [82].

2.3.1.1. The Password-Authenticated Key Exchange (AKE)

The Password-Authenticated Key Exchange (AKE) allows users to produce a robust cryptographic key, which is generated based on sharing the human-memorable password without the need to use the public key platform. This is one of the essential and popular usages in the cryptographic field. However, theft still exists even when the password has been salted and hashed, because several attacks e.g. a rainbow table attack can crack such passwords successfully. A secure protocol has been proposed, the database of passwords will be shared between two servers or more, such that the authentication procedure will require a distributed computation, with the client and servers thus being involved. In addition, regarding this scenario, if one server is compromised, this will not be a problem, because the attack is not valuable to the attacker due to only the shared database of the password having been revealed. That is, under these circumstances a brute force guess at the password will not work without additional interaction by the attacker with the user [83].

A smart card has been used with the AKE protocols to store sensitive information, which is usually used for authentication or to generate the session key. The main principle behind this idea is based on the assumption that there is a tamper-resistant property with smart cards. Nevertheless, there is a risk for sensitive information that has been stored in the cards in that it can be extracted from them by using side channel attacks. Obviously, if the intruder has ability to steal the user's smart card, then he/she will have an opportunity to impersonate the victim user, or the adversary will have the ability to launch further attacks, known as a Stolen Smart Card Attack. A three-party password authentication key exchange protocol has been

proposed to protect the cards from this type of attack, whereby even though the user card has been stolen or lost or the information on the card has been extracted, the information will be still secure against the attack [84].

Amongst their numerous real-world uses, authenticated key exchange (AKE) protocols have been employed to ensure communication channel security. In one paper, several additions to the existing literature were provided. Leakage-resistant AKE protocol security modelling was reviewed and assessed, with it being demonstrated that extant models failed to capture leakage attacks to a sufficient extent or else that they impose somewhat unnatural demands on the user. Hence, these authors proposed new, strong security model, referred to as the eCK (CLR-eCK) challenge-dependent leakage resistant model. This model captures leakage attacks on ephemeral (randomness) and long-term secret key types, which are challenge dependent. Additionally, a general framework for the construction of one-round CLR-eCK-secure AKE protocols was proposed as part of one study, based on SPHFs (that is, smooth projective hash functions). Lastly, a general framework based on the Decisional Diffie-Hellman assumption not including a random oracle was presented practicable instantiation. Regarding the computation overhead and the communication, the findings demonstrate that the instantiation is efficient and that, it can capture a greater number of leakage attacks [85].

2.3.1.2. Password Vulnerabilities and Protections

Leakage of the password is one of the main concerns to the user as a real threat for the password-based user authentication. Despite of this problem having been investigated by researchers, there is no adequate solution. The leakage of the password generally happens during the authentication process, when the user inputs the password, which is the rationale underpinning the designing of Leakage-Resilient Password Entry (LRPE). Certain criteria have been used to build an effective LRPE scheme, which will not only cover leakage resilience, but also keep most of the features of usability benefits of old style passwords [86].

Traditional password based authentication schemes are vulnerable to many types of attacks, one being a shoulder surfing attack. In this case, the adversary can see the authorised during the login process when inputting the password, which can then be reused to get illegal access to the system and hence, be able to take some

malicious action. Whilst there are many techniques to prevent such attack, the attacker has ability to observe these as either fully or partially. Some researchers are working on preventing these attacks, while others are working on detecting the attackers [87]. The timing password technique deployed in this thesis will make a shoulder surfing attack very difficult to apply.

In a distributed environment, most protection methods are focused on the server's protection rather than the information sharing. Usually, a single server has been used with the traditional authentication of biometrics-based password mechanisms, which are easy to expose. Using a dual stage authentication technique has been demonstrated as a good alternative to a stage application. Furthermore, the protocol has been enhanced by ensuring communication between the master server and authentication server is via the secure link [88].

Authentication based on employing friends to help users has been proposed (for instance, trustee-based social authentication), which has been shown to provide a good backup in the authentication process. In this system, the user's trustee friends are carefully selected from among his/her friends to be associated with him/her. Furthermore, when the user tries to login onto the system through his/her account there is different verification codes have been sent by the service provider. A recovery threshold k that will be represent the minimum number of verification codes obtained for the user from the trustee before resetting his/her password. However, this mechanism is vulnerable to what are called forest fire attacks. In the beginning, the attacker will try to breach a small number of the users, and then, will continuously try to attack the rest of the users by compromising the trustee involved in social authentication [89].

To crack probabilistic passwords, multi-word patterns and keyboard patterns (when there are two words or more in the alphabetic password part) have been added regularly to context-free grammars. The results have shown that the enhancement of the new system has been up to 22% compared with the original one. Furthermore, the protection against a dictionary attack has been increased up to 33% [90]. Researchers are continuing to try to find techniques make the password resilient against cracks or attacks, and in this thesis, the timing password and honeypot techniques are employed to keep passwords safe from dictionary attacks.

Memory-hard functions need a lot of space for efficient evaluation and when used for password hashing, they significantly increase the time and cost of offline dictionary attacks. The memory-hardness of the Balloon algorithm was analysed through the use of "random sandwich graphs". Moreover, general techniques have been developed to give a proof of security for both the script and Argon2i password-hashing functions in the random-oracle model [91]. Argon2i is concerned with secrets protection of low-entropy without using the secret keys [92]. A computation sequential model has been used to analyse the security system, which helped to capture the attacks running over single-core machines [91].

2.3.1.3. The Cloud-Based Authentication

With the mobile cloud there are various authentication procedures methods used, which can be classified into two main categories. The first is cloud side authentication, whilst the second pertains to the user side. Moreover, each category can be divided into two sub-categories, according to the types of authentication credentials. The best definition for a credential is a unique identifier used for node authentication. Within this classification, there are two credentials types, the first being identity-based credentials, while the second refers to the context-based ones. Most authentication steps on the cloud side are managed by the cloud server. On the other hand, with the user-side authentication methods, most steps are performed in mobile devices. Regarding comparison of smartphones with conventional PCs, the former are connected to the internet most of the time, and using the network along more than one path, which will make the user's smartphone more susceptible to threats. Some features to enhance authentication performance require flexible processing and extra storage capacity. Additionally, more than one authentication factor can be combined together, according the user security requirements [93].

Whilst cloud-based authentication introduces some benefits in terms of performance and usability, it introduces some security and privacy issues. In an era when mobile users are consuming cloud services from a plethora of different cloud vendors who store their data in multiple instances around the globe (redundant data for data safety), the user's private authentication information, such as password and biometrics, are highly exposed to risk. Consequently, a robust authentication method is a critical requirement for a mobile cloud environment. Whilst there are some

advantages regarding to the performance and usability in the cloud based authentication, there are some issues relating to the security and privacy [94].

2.3.2. One-Time Password

Robust security is always demanded by the users, especially regarding crucial systems, for instance, financial ones, in order to secure the accounts of clients. To this end, one of the best solutions is to implement the one-time password (OTP) [95]. The main purpose in this case is to overcome the key significant problems of the traditional password. Nevertheless, to implement this technique special hardware is required, and this is its main drawback in that it will raise the cost, and create problems regarding availability [96]. The main idea behind using a one-time password is to make it encoded for single use with there being a unique password generated for each login process or transaction [97].

There are two commonly in OTP techniques used: *token-based* and *tokenless*. There are some disadvantages with the token-based technique, such as it requires hardware, which raises the cost. Moreover, regarding this type of OTP, it is difficult to deal with multiple tokens, and easy to lose or forget them due to their small size. On the other hand, the key advantage with is this type of OTP technique is that it is safe against keylogger [98]. Tokenless OTP can be divided into two types: soft-token (software) and Short Message Service (SMS). The main advantage with soft-token is that it uses the existing infrastructure, while the main disadvantage with this type is that is vulnerable to keylogger and malware. Despite SMS -tokenless having the advantage of giving the user the opportunity to use the same phone device, there are some drawbacks relating to this technique. For instance, owing to this depending on the user's phone, a poor cellular network will hinder its usage. Furthermore, a delay of the message could affect this type of technique, and another disadvantage is there will be no authentication in-browser. Another disadvantage relating to this technique, is that there will be no support for a non-SIM based device [99]. The OTP is outside the scope of this thesis.

2.3.3. Graphical Password

Graphical passwords are much easier to remember than knowledge-based ones and hence, they have become more popular to use than previously [100]. Some researchers contend that they are a good alternative to text ones for user authentication. That is, they believe the best way to improve the security and usability is by employing "culturally familiar pictures". Results have shown that culturally familiar graphical passwords used with unfamiliar decoys are easier to remember than such passwords with familiar ones [101]. There are two main categories of graphical authentication mechanisms:

1- Recall-based: The first stage is about drawing or identifying the locations of the image.
2- Recognition based: To recognise a group of images among a larger set [102].

Microsoft's Picture Password is a technique for authenticating a user that avoids the need of having to type a character based password. The password comprises a set of gestures drawn on an image, with the position, direction and order of these gestures making up the password. In addition to being easy to use on touch screen devices, this method delivers better memorability as well as enhancing password strength against phishing attacks. Moreover, researchers have demonstrated that for portrait pictures people are strongly attracted to using facial features as gesture types. By collecting a set of Picture Passwords and employing computer vision methods, a list of password guesses with decreasing probability order can be constructed. It has been found that guessing in the set order increases the likelihood of finding a password in a restricted number of guesses [103].

Some researchers have concentrated on evaluating the Persuasive Cued Click Points graphical password system, with the usability and security having been assessed on three different levels. The aim is to make the authentication system have the ability to support the users in selecting a better password and accordingly, the value of the effective password space should be expanded. When a user chooses a poor password in click-based graphical passwords, this will lead to a hotspots problem emerging; there are some parts of the image that have a higher probability of being selected points, which will allow the attackers to be more successful when using

dictionary attacks. Hence, some researchers are encouraging users to select the points in the image randomly, which will make them difficult for attackers to guess [104].

A graphical password scheme can suffer from shoulder surfing attacks, and most users prefer use textual passwords despite the problems associated with them. Text based graphical passwords have emerged to overcome some of those problems related with textual ones. To improve the text-based shoulder surfing resistant graphical password scheme and to make the login procedure more efficient, a colours technique has been proposed. The results show that this new technique minimises the threat of shoulder surfing attacks [105]. That is, the elements added to a textual password, such as the techniques used in this thesis, will increase its resistance to password's attacks.

Static digital images, those commonly utilised as part of password systems that use graphical password system incorporation tailored/personalised physical tokens, are now being replaced by the PassBYOP, a contemporary public-terminal password scheme. As part of this process, the digital image is shown on a user-device, such as a tablet or a smartphone; the user then uses their system's camera to take an image after which they are able to input their password according to a series of choices using a live video of the token. Passwords employed as part of this process have highly distinct visual elements. Within one study, three PassBYOP feasibility studies are assessed regarding reliability and usability and security from observation. According to the reliability study, the appropriate systematic limit for image-feature passwords, which are shown to be viable, need to include a minimum of seven aspects and of these features, so that they can be judged as equivalent, forty percent must geometrically match the originals retained on an authentication server. Completion times (at 7.5s) and error rates (at nine percent) have been revealed by the usability study. These figures are, in the main, comparable with existing static-digital imaging using graphical passwords. The PassBYOP's resistance to certain attacks, namely observation attack-three attacks, is because attackers cannot utilise shoulder surfing, malware or observations using a camera to compromise the security of a passwords, as can be seen in the conclusions of this study. Hence, security through the use of a PassBYOP has been shown to be beneficial by these findings for existing graphical password schemes [106]. However, in this thesis, graphical passwords will be considered no further as they fall outside of its remit.

2.3.4. Biometrics Authentication

Several methods have been utilised to limit database and/or application access by the user in the form of a biometrics pattern-recognition scheme. There are two main categories of these regarding the biometric features stored or captured from the user input. First, there are physiological features, those aspects that concern the physical features of the user in question, whilst the second pertains to behavioural features, which capture the user's behavioural traits. Of the former, facial recognition, hand geometric measurement, palm prints, DNA and fingerprints are examples, as is odour/scent and iris recognition, the lattermost having replaced the older retina recognition feature. Regarding behavioural features, gait, vocal, and typing rhythm are exemplars. There are two modes when biometric systems are in operation: verification, which is when the system ensures that the user is actually the individual he/she purports to be, and smartcards, identification and an identifying number are all examples of a verification tool used for access. These are also deployed in the second mode, which is identifying an unknown user trying to access a secure system [107]. Figure 2-1 shows the biometric technique classification as utilised in several systems and for many applications. There are advantages and drawbacks to each of these biometric methods.

Deoxyribonucleic acid (DNA): Using DNA is unique to the person in question (with the singular exception being identical twin, who shares a DNA code and pattern). It is, therefore, arguably an optimal one-of-a-kind code, barring the mentioned exception. Nevertheless, at the current time, its utilisation is predominantly seen in the forensic field. Its employment as an alternative application as a biometric is subject to the following three concerns: 1) sensitivity and contaminants; 2) immediate real-time problems when used for recognition; and 3) problems regarding privacy [108].

Ear: Essentially, this method is one that measures and then matches the space between salient locations on the pinna from a specific point on the ear. However, the characteristics of ears are not unique to a particular individual [109].

Face Recognition: This is the most widely used of the biometric characteristics recognition techniques. Verification of the face can be controlled, static or dynamic uncontrolled. The most common applications are: the shape and

positioning of the eyebrows, chin, nose and eyes; or comprehensive assessment of the face through the compilation of a set of authentic images to provide a single one [110].

Thermogram of hand veins: individuals have a particular arrangement regarding their bodily thermal radiation and its distribution. Indeed, these features may be photographed using an infrared camera without much difficulty or intrusion. No contact is needed when a thermogram-base system is used, and the process is not an invasive one [111].

Fingerprints: this is the traditional method of identification that has been utilised for over a hundred and twenty years as it is very accurate. Individual fingers of an individual do not share fingerprints, nor do identical twins [112].

Gait: this concerns the identification of an individual according to the very particular way in which they walk/move. This method of biometric identification utilises a spatial temporal measurement. This generally used in lower-security protocols due to its inaccuracy. It is a biometric identification technique that uses biometric gate per recordings of an individual moving/walking to gauge particular distinct movements [113].

Hand geometric measurements: This method of identification determines the measurements of an individual's hand through scanning and recognition according to palm size, the width/length of the fingers, among other characteristics. Owing to the size of the hand's base geometry this method of identification cannot be employed everywhere, for instance, it is not suitable for laptops. The process can involve a video signal of the geometry of one hand being scanned, a computer then digitises the image of the hand or a video of it can be acquired [114].

Iris: In the eye, the iris is surrounded by the sclera (whites of the eye) and on the inside, there is the pupil. This annular area has a visual texture, which is generated when the foetus is developing, with this arrangement being subsequently stabilised throughout the first and second year of the individual's life. Iris based recognition systems are encouragingly expedient and precise; indeed, larger identification systems per iris recognition seem feasible at this time. Like fingerprints, irises are different, even when it comes to identical twins, with those of every individual being unique [115].

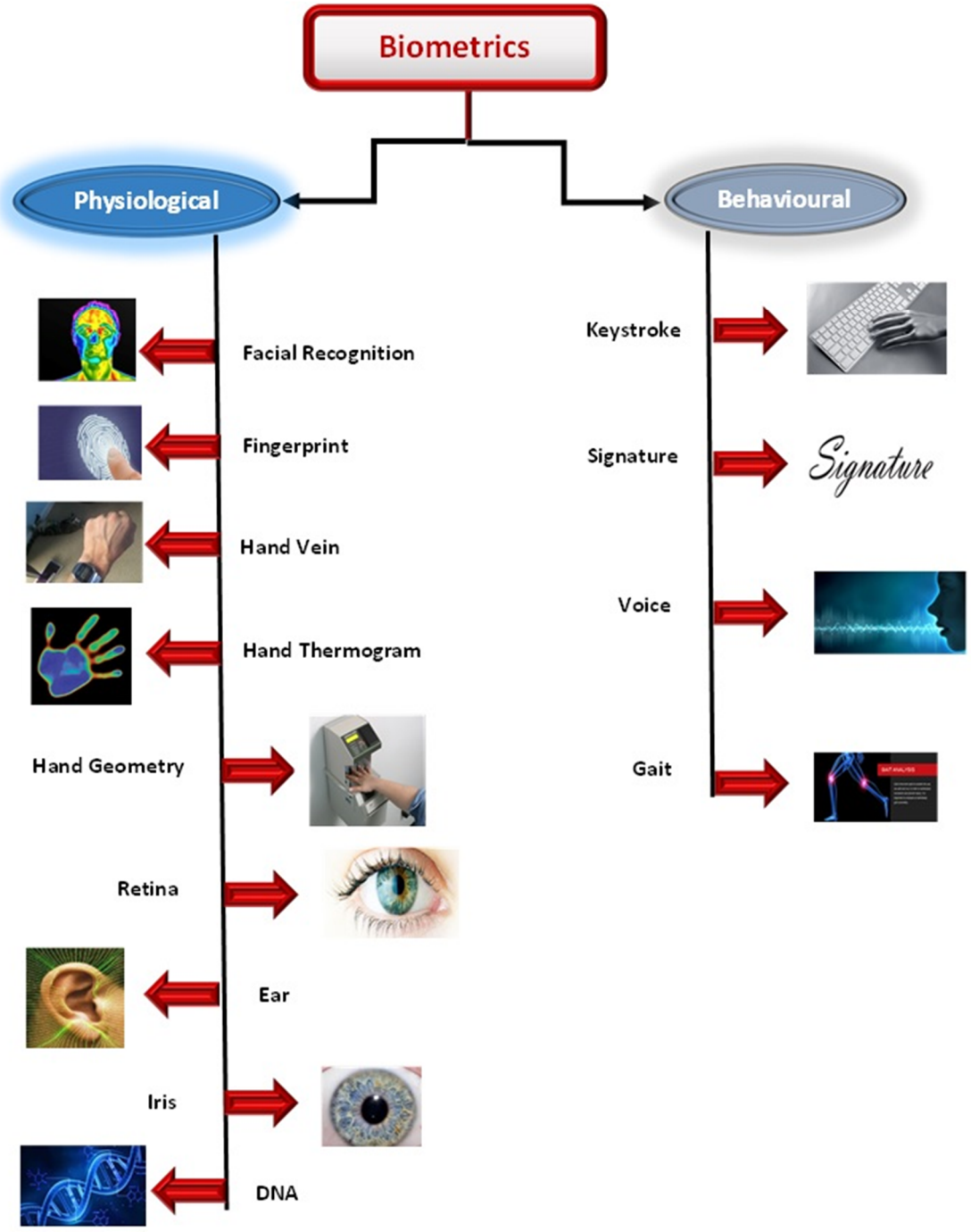

Figure 2-1 The classification of main biometrics techniques

Retina: this type of scan identifies the pattern of the blood vessels in an individual's retina and predates the other eye-based technology, namely, the iris scan. Since the retina is not immediately visible, coherent infrared light sources are required so that it can be illuminated before scanning. Because the immersion of the infrared wavelength light energy is dissipated more expediently compared to surrounding tissues, this then allows an image of the blood vessels themselves to be made, after which an analysis is conducted in relation to particular points within such an

arrangement. Some diseases can have an impact on the retina scan, unlike the iris scan, but they are very uncommon [116].

Signature: this is utilised so that the handwriting style of an individual can be used for identification purposes. This method looks at the person's signature when written by hand, which is deemed to be a behavioural biometric and is included within the categorising of dynamic verification technologies. The stroke, the speed, the pressure of the pen and further timing information are all incorporated in this assessment when an individual is providing his/her signature [117].

Voice: This technology uses sound rather than aesthetical identifiers. Regarding which, the sound sensations of individuals are used, with the information then being compared and contrasted to an extant set of data. The verification process generally demands that the individual say a specific phrase or code, which is then utilised within the process for verification [118].

Despite biometrics authentication mechanisms having been put into widespread deployment, users are still concerned regarding their security. Morphing attacks, refer to the infiltration of artificial images that have been created by using biometrics information consisting of the combination of two or more objects. Morphed images have been demonstrated as representing a specific threat to civil security, whereby a wanted criminal can use an authorised passport to enter a country with a fake identity [119]. Lastly, authentication through biometric means is vulnerable to presentation attacks, where authentic biometric data are shown to the sensor such that access is secured through illegitimate means. Of all the means for breaking a biometric identification system, this is the most frequently used [120].

Several spoofing methods of biometrics ID technologies have been utilised by hackers, who always rise to a challenge. In one example, the ID information for accessing a phone was secured through the 3D printing of a latex glove matching the 'victim's' fingerprints and consequently, the 'victim's' phone was successfully hacked by using this glove with its fingerprint reader. Another example involved hackers using merely a picture of a person's hand, meaning that they did not even need to gain access to the user in question for the hacking to be successful. Even facial recognition technology can be potentially hacked by holding up a photograph of a person to a scanning machine's camera, which works because many of these

machines work in just two dimensions. Users are now requested to smile or to blink according to newer more advanced systems; however, compared to fingerprint biometrics, facial recognition processes are more easily spoofed [121].

To sum up, it would appear that the biometrics authentication techniques all need special devices (sensors) to recognise the biometrics, but not all computers include these and hence, not all authorised users are able to login into systems. Moreover, biometrics can be attacked by an adversary. Table 2-1 illustrates the characteristics of the most popular biometrics technologies [122].

Table 2-1 Characteristics of Biometrics technologies

Characteristics	Fingerprint	Hand Geometry	Retina	Iris recognition	Face	Signature	Voice
Ease of Use	H	H	L	M	M	H	H
Incidence of Error	Dryness, dirt, age	Hand injury, age	Wearing glasses	Lighting	Lighting, age, glasses, hair	Signature Changing	Noise, voice changing (having a cold)
Accuracy	H	H	V.H	V.H	H	H	H
Acceptance of User	M	M	M	M	M	H	H
Long-Term Stable	H	M	H	H	M	M	M

V.H= Very High H= High M= Medium L= Low

2.3.5. Keystroke Password

Keystroke dynamics (KD) can be used to minimise the threat and to protect data against malicious attacks. It has been used to analyse the behaviour of users and also as a retraining method for ensuring fraud patterns are absent at the registration time. The insider threat will be mitigated by retraining boosts for whole system performance [123]. KD is a biometric method for user authentication; however, in computer security, it has limited use owing to greater typing time and the need for

more practice sessions than the traditional method. Nevertheless, it has been proposed that the conventional method can be enhanced through examination of KD and Click Patterns (CP), thereby increasing the security level without the need for complex password. Moreover, User Time Signatures (TS) were identified subsequent to analysing user KD and CP. According to their capacity to adhere to their specific TS, they have been classified as beginner, standard or expert. During login, the user inputs are matched with the relevant database records in order to authenticate them [124].

Digraph, which involves the temporal information concerning two consecutive keystrokes, is among the most significant keystroke dynamics biometric features. This process may be separated into *Flight Time* and *Dwell Time.* The latter of these refers to the time that passes between pressing and the releasing when making a single keystroke for a single key, that is, the length of time the key in question is being pushed down. Whilst the former is pertains to the time between one key and the next, that is, the time between the pressing of two keys in consecutive order. Secondary features remain relatively unsearched. The Delete and Backspace keystrokes information was recorded and collated to determine the regularity of mistakes made. Moreover, the shift and the letter keys were monitored to ascertain the order in which the many keys were released as people typed capital letters into their keyboards [125].

Keyboard typing identification has been suggested according to the individual proposals. Additionally, there are several algorithms that have been devised for machine learning, which are then coupled with a pairwise user coupling process/technique suggested for gauging the efficacy of each technique's performance as well as for assimilating more than one technique when used in conjunction. The findings demonstrated that a bottom-up tree schematic structure with use of the pairwise coupling was the most efficacious. Keystroke data were used to validate the various techniques. A publicly conducted comprehensive analysis was performed by means of a publicly available examination accessible through the internet. Seven percent higher accuracy regarding the keystroke dynamic's dataset was attained compared to a cutting-edge and up-to-date result employing the same set of data. That is, when utilising normative typing behaviour, about ninety percent accuracy was attained [126].

The process of discerning the various typing patterns of users has been considered through the utilisation of a keystroke dynamics features learning

algorithm. Specifically, mini-batch bagging (MINIBAG) has been suggested with the 1-class naive Bayes (AR-ONENB) algorithm. MINIBAG chunks every single attribute from the set of data into numerous sub-datasets through the pre-processing stage. AR-ONENB arranges the various characteristics according to the time-frame throughout the same stage (pre-processing), thereby ensuring efficacious and proficient clarification. The results yielded from experiments utilising these algorithms for user-authentication benchmarking in relation to keystroke dynamics. MINIBAG allows for machine learning algorithms with a set of mini-batch multiple loads, whilst AR-ONENB calibrates the log-likelihood value from the keystroke index order to estimate anomalies, thereby utilising the fact that everybody's typing speed is different [127].

There are numerous factors that impact on keystroke production many of which have not been given sufficient attention in the extant literature, two of which are contextualisation and linguistic context. Authentication experiment accuracy can be increased by including the linguistic context when the keystrokes are undertaken. Additionally, the Equal Error Rate EER baseline can be reduced to 0.0309 from 0.0483 by 36% using a 486-user dataset through the utilisation of just digraph intervals and unigraph holds. Moreover, through the reduction of the feature-set size by various means, the EER results were improved to 0.0232. Hence, the significance of context through typing authentication is evidenced by these findings [128].

Strong, ongoing and unobtrusive user authentication means are potentially attainable by free text keystroke dynamics. This method is one of behavioural biometrics, when referring to the categorisations mentioned above. Users can type any information they wish during the authentication process as part of free-text biometrics. However, not all user exhibit stability of the patterns utilised to differentiate these keystrokes from those of others. Indeed, if an individual is playing a game on their computer then there may be "unstable" keystroke inputs, or if the user is typing nonsensical text. Regarding this, in one study, the author devised a hypothesis asserting that certain forms of text have a negative effect on authentication keystroke dynamics and consequently, they need to be vetted, i.e. filtered out. The effect of nonsensical text or "gibberish" on the process of authentication was then investigated. That is, using a bespoke set of keystroke data, where it was found that almost a quarter of all keystrokes needed to be categorised as nonsensical "gibberish"

(23.3 percent). Additionally, it emerged that there was no impact on the false accept rate according to nonsensical keystrokes, there was significant positive one in the false reject rate [129].

2.3.6. Multi-Factor Authentication

Multi-factor authentication is an efficient mechanism used to protect sensitive data. Multiple authentications techniques can be combined together to provide an effective security system [130]. An authentication process is invariably used as essential protection against any unauthorised access to any devices or applications. Furthermore, single factor authentication is increasingly becoming adequate for protecting the systems as threats to security become more sophisticated. Hence, multi-factor authentication can be used as a viable choice for protecting computers and others devices using the Internet. There are many types of authentication mechanisms with the different security levels and they are available for several kinds of devices. That is, the well-known multi-factor authentication mechanisms have been used to improve security systems in the various applications [131].

A multi factor environment can be used to solve the problem of user registration in several servers, but this makes it difficult for users, as they have to remember different usernames and passwords. A secure multi factor authentication protocol can be based on smart cards, biometrics and/or Elliptic Curve Cryptography (ECC). There are some problems regarding the classical system:

1- It can suffer from Denial of Service (DoS) and inside user attacks;
2- The whole system can break down due to incorrect work by the register centre, thereby failing to deliver robustness [132].

A telecare medical information system (TMIS) is very desirable from the perspective of users, as they can gain access to information or services of a medical nature remotely. However, preserving the privacy of users and authentication remain difficult. Contemporaneously, certain schemes, like two-factor authentication, or smart card-based authentication, have been put forward. When using a 2-factor authentication scheme with a TMIS, this does not secure the system from dictionary attacks or stolen smart cards [133].

Utilising a multi-factor *user-centred* data-backup scheme has been suggested which involves the user choosing a symmetrical key before separating it into a further three shares. The key is then eradicated; however, it can be remade simply by assimilating and combining stored shares retained in the laptop of the user. In the eventuality of the laptop, or the smartcard going awry, the key is remains recoverable through biometrics and the password. This scheme has not only been shown to be practicable and reliable, for it also has demonstrated the ability to meet security requirements [134].

A multi-factor authentication (MFA) has been devised using PINs (personal identification numbers), a speaker biometric of speech watermarks and an OTP (one-time password). Regarding the speech watermarks, digital speech watermarking is utilised to capture robust and semi-fragile watermarks concurrently for speech, thereby proving voice ownership and intruder identification. Specifically, the QIM (Quantisation Index Modulation) and the DWPT (Discrete Wavelet Packet Transformation) are employed, both of which are blind semi-fragile watermarking procedures, for the speech information, so that an angle of the wavelet's sub-bands can be determined and thus, yield more and more specific information [135].

2.3.7. Honeypot

A honeypot refers to part of an information system that can be used to detect unauthorised or illegal use of a resource. It is a decoy computer system that has been designed to look like a real system, which attracts an adversary to break into it, while unknown to him/her; he/she is being covertly observed. The imposter is unaware of precisely where the honeypot is and hence, they can be very effective [136]. This mechanism tempts the attacker into seeking vulnerabilities in the decoy system, whilst the observer probes his/her computer system in order to learn about the strategies and tools used, thereby subsequently being able to improve system security [137]. With a honeypot it is possible for the developer of the security to test and analyse attacks, thereby gaining useful information about the major direction from which they are being launched [138].

Honeypots can be classified in two ways: client and server. Client honeypots reveal an in-depth insight into client-side attacks, and are also known as active

honeypots or honeyclients. In contrast, server honeypots are more knowledgeable about server-side attacks, which are a type of passive honeypot. The technology of both types of honeypot is being heavily researched when it comes to cyber security. The differences between the client and server honeypots are as follows:

- Client-side: this emulates and drives the client-side software and does not allow server-based services to be exposed to attack;
- Active: it cannot attract attacks to itself, but rather, has to institute actively remote service interaction in order for this to happen;
- Identifying: whilst any access to a traditional honeypot is automatically a malicious act, the client-side honeypot is able to distinguish between the adversary behind it and a benign server [139]

For any information source, safeguarding the access, availability and integrity of data is a fundamental security requirement. If these are compromised in any way, then there is the risk of intrusion into the system and security threats associated with that. A honeypot is a sophisticated decoy-based technology, which offers attractive opportunities for those involved in computer security. It can also be considered a universal concept, as it can provide solutions with other security technologies. Moreover, an advanced hybrid honeypot with unique content has been proposed. These features allow it to be flexible when it comes to deployment systems, based on the assembled parameters of a system. Through a process of replicating vulnerabilities and a lack of security, the honeypot entices attackers. Once interaction has been established, the system will monitor an activity undertaken by the attackers and analyse the collected information so as to enhance computer security [140].

Network-layer honeypots traditionally fall into one of two categories: low and high-interaction honeypots. The former are aimed at accurately replicating the services and certain behavioural traits of a system. No effort is spent on other services, and the replicated services might not initiate the full set of features of the service. Because these honeypots have restricted use, high-interaction honeypots are favoured [141]. In contrast to low-interaction honeypots, the high-interaction model presents a fully interactional system. Not only does it replicate the functionality, operating system and services, for it also provides actual system services. This enables the attacker to carry out a full honeypot with system control, thereby allowing for more to be discovered about the methods, tools and motives of this attacker. High-

interaction honeypots can act as decoys that drain the attacker's resources as they are harder to detect. Whilst this has its advantages, attackers can hijack honeypots and use them to attack other systems on the Internet. One example of a high-interaction honeypot application for the Internet is the High-Interaction Honeypot Analysis Toolkit (HIHAT), which utilises the real web system to operate, transforming any PHP application into a very interactive honeypot [142].

It is generally agreed that the most effective honeypots are high-interaction models. The attacking element may or may not be automated, whatever the case, they provide an authentic interaction with the machine that is targeted. However, ordinary honeypots are not strong enough to assess their behaviour on the target machine when it comes to automated attacks caused by malware. Researchers have revealed how honeypots with high interaction can be enhanced with certain features to power them to develop the reverse engineering qualities required to stave off and analyse threatening attacks [143].

Cloud computing is at grave risk from security breaches and other issues. Being such an elaborate networking system, some security systems can be implemented, but there is no safeguard fix to all every security threat. The cloud is at risk from integrated and intelligent attacks, hence hosts have to ensure that they can not only detect these threats, but also, prevent and repair them. Some researchers' focus has been to introduce honeypots into the cloud computing systems in order to analyse the pattern of attacks. The concept of honeypots is relatively new and hence, this type of technology has not been considered in depth in the field of security protection. One exception to this is Cloud Honey (CY), which is an open-source framework that sustains both high-interaction and low-interaction honeypots within the cloud. CY collates and analyses information on attacks, which helps to build up profiles of the attackers [144].

Despite the numerous defence solutions to combatting malicious websites that have been devised, countless numbers of them remain active. This has been attributed to the *ecosystem of malicious redirection*, with one study aiming to comprehend the way in which this has been evolving through long-term assessment. With this goal in mind, a honeypot monitoring system was devised by the authors and deployed over a 4-year period, which specialised in URL-redirection behaviour monitoring. More than 100,000 malicious redirected URLs were collated over this 4-year period, being

extracted from a total of 776 different websites. There were three major findings: 1) A further attacker motive has emerged in the form of click-fraud, which encourages URL redirection; 2), web-based domain generation algorithms (DGAs) are increasing in popularity and use, whereby they are employed to boost redirect URL entropy and thus, circumvent URL blacklisting; and, finally, for immediate site of direction chains and the deployment of these IP-flux and domain flux are used in concurrence with one another so that the redirection robustness can be increased. Given these findings, the authors have suggested several preventative measures concerning malicious URL redirections. From the details and data collated using the honeypot-biased monitoring solution, it is now possible for network and security operations to leverage beneficial, utilisable information. One instance of this, is the potential disruption concerning web-based attack infrastructures, which the system achieves by closing those domain names once they have been extracted from the monitoring system. Identification of individuals that undertake these attacks can also be made through the collation of the tracking IDs and web advertising IDs [145].

A honeypot needs to be created by hackers, if they are to be successful in cracking passwords. To store collated passwords and IDs, a Structured Query Language SQL statement is first needed by the honeypot, as is a table in the database. Whilst the development of information and communication technologies ICT has been significant in regard to convenience and ease-of-use, there are many negative repercussions that remain. Malicious code spreading and E-mail or SMS-based lures sent to users to draw them towards false websites through the utilisation of social engineering, are known of and have been used maliciously in the past. SMshing (phishing through SMS), pharming and phishing are becoming increasingly disparate and diversified. Most websites are processed through the easy expedient of utilising passwords and IDs; it is because of this that attackers have started to alter their means and methods. When, in the process of authentication, users of the internet are more susceptible to attack as they are often seen employing the same password and/or IDs across a number of different websites [146].

Finally, a honeypot can be used on both sides, first, there can be an attack luring the victim by the use of a link, image or any other phishing method. On the other hand, a honeypot can be used towards improving the security system by luring the adversary to perform an action that will not only fend off the attack, but also, help

in the prevention of future attacks on the system. In this thesis a high-interaction honeypot based on web session management is used to enhance the authentication technique.

2.3.8. Honeywords

The idea behind honeywords is to create a relation between the real password and decoy hashed passwords, such that for every user the latter look like real passwords. That is, the honeywords are these decoys. An attacker can recognise the presence of honeywords in a password file, as it is very unusual to have multiple passwords for a single user account. However, even if the attacker can crack multiple passwords associated with a user, he or she does not know which are honeywords, and which the real ones [147].

Most existing Biometric Template Protection Schemes (BTPS) do not offer as strong security as cryptographic tools. Moreover, they are unable to determine whether or not a probe template has been downloaded from the database by an imposter or an authentic user. Consequently, the "honeywords" idea was proposed to detect the cracking of hashed password databases. In particular, an extra layer of protection is needed with biometric feature schemes, as these have been shown to be flawed. A honey template protection scheme relating to faces has been proposed and evaluated as representing an improvement on existing schemes [148].

Many researchers have pointed out that most password hashes are not safe against hackers and hence, the method of honeywords (decoy passwords) has been used to detect attacks against hashed password databases. Furthermore, cracking hashed passwords has become easier for an intruder, who wants to enter the account through an authenticated user. In addition, a user's password can be recovered by an intruder through using a brute-force attack on the hashed password. A user's real password can be distinguished among honeywords for each user by using a secure server called a "honeychecker", which triggers an alarm when a honeyword is used [149].

A further advantage provided through honeywords, is that how clients perform their particular passwords can be gleaned from distributed records. Subsequently, the person trying to crack the password is able to reassess the client watchword

determination models and then prepare for a more expedient word-splitting calculation as a result. Within extant frameworks, all password encoding is stored by the authors using encryption components. Decoding methods with standard calculations are used to secure a secret key by programmers. Through these means, prospective assaults can be assisted through each security key break. Some model working regarding prospective hash makers are discombobulated and convoluted regarding certain systems used and devised in the honeymoon 'era', for instance, chaffing systems, can darken word decisions from real clients. Indeed, potentially, certain honeywords will, to some degree, bother likelihood dispersions could be drawn upon through beneficial and deliberate acts to sloppy the attackers "client information" structure decisions [150].

2.4. Behavioural Password

Generally speaking, users tend to determine and set their passwords according to personal and identifiable details, knowledge and information. Whilst many are aware that they are potentially insecure, users still employ important and significant words and information when coming up with their password. Coping behaviour are concerned with the precise self-management of the resources of the user and also, the fact that the vast majority of users' obliviousness regarding efficacious password practice. Additionally, the disinformation regarding passwords and their strength remains a notable reason why many users have weak passwords. Moreover, user very often draws on easily recalled information in their memory when forming a password, which invariably has a personal link to them. In general, dictionary words and personalised information can be guessed with ease, if an attacker has sufficient computational power and a sufficiently large number of guessing attempts [151].

A context-autonomous ongoing authentication system has been the concern in [152]; one that reacts according to all user actions. An algorithm for a robust dynamic trustworthiness model is one contribution that is applicable for all ongoing authentication systems, regardless of the biometric modality they may employ. Further, the authors have also provided an innovative performance reporting technique for the continual authentication context. A unique behavioural biometric dataset was used, along with considerable and extensive experimentation when

validating the suggested approach. A total of 53 users' data were collated to form a dataset within entirely uncontrolled conditions through the researchers' data collation software. Mouse utilisation and keystroke patterns were both considered to stop a potential attacker from escaping detection through a single-input device restriction method as the system will, in this example, just check to alternative input device. A technique for feature selection is devised for application for alternative pattern recognition patterns throughout this study. Of all the results in this aspect of the study, the best is that fifty from 53 users were never locked-out by the system mistakenly; however, three (5.7 percent of users) did occasionally get locked out over a mean of 2,265 different actions. Additionally, only 0.1 percent of the imposters managed to avoid detection, that is, three out of a total of 2,756 over a mean of 252 different actions [152].

User interactions and behaviour regarding a multi modal authentication system is the focus of one study. That is, in order to gain access, the legitimate user of a web application had to apply multiple authentication methods. Digital certificates, passwords and OTPs, are methods proposed in this respect. According to preference, users can choose the method they like, referred to as personalised authentication. The method's availability will remain adaptive, however. The authentication request's character and nature are determined by the system through computation, as are the user's location and behaviour traits as well as the application's trustworthiness, which are utilised as determinant factors. Moreover, users' help-desk support, databases and system log data retrieval are used to determine the interactions and behaviours. Live authentication systems and relief users are used to generate the data. User interactions and user behaviour are analysed for the following reasons: they provide an indication of the risk and profile of the respective authentication means, whilst minimising security and usability imbalance regarding prospective development and future design [153].

2.5. Web-Session Security Management

It is possible to launch an attack on a web session through numerous layers, because web security is complicated. Hence, the presumptions made should first be discussed and their importance regarding security considered.

1- ***Perfect cryptography*:** Web sessions may be damaged through a man-in-the-middle attack or network 'sniffing' at the network layer. The HTTPS protocol can be used to protect web traffic, this service being responsible for wrapping the traffic in an encrypted channel.
2- ***The attacker is not able to compromise the web browser:*** Generally, the available mechanisms for protection, as offered by normal, standard web browsers, are relied upon by the web applications, e.g. HTTPOnly cooking attribute or the same-origin policy.
3- ***Content injection vulnerabilities can impact negatively on trusted web applications:*** As can be learnt from the past, it is impossible to give a cast iron guarantee that a web application is immune to this type of threat [154].

The first application of micro-policies to web-based security will be presented in [155]. This is achieved by the study of a core browser mode and determining its efficacy when securing web sessions. A web session security is a micro policy that enforces simple declarative information to be expressed and translated. Subsequently, a flexible, secure and elegant browser-side enforcement mechanism can be realised and generated, one that, for web developers, retains its accessibility. In [155] it was shown how, through adopting such an approach, a large array of web session attacks can be prevented with efficacy and uniformity. Additionally, and as part of the proposed Google Chrome extension, a proof-of-concept implementation of an important core of this extension shall be developed and proposed. This extension, Michrome, can be configured easily and to ensure that strong security policies can be enforced without the need to compromise the website's functionality [155].

Figure 2-2 shows the types of web application vulnerabilities and various forms of attacks [156]. In this figure, the main parts concerning this thesis have been shaded grey colure and this starts with session management vulnerabilities.

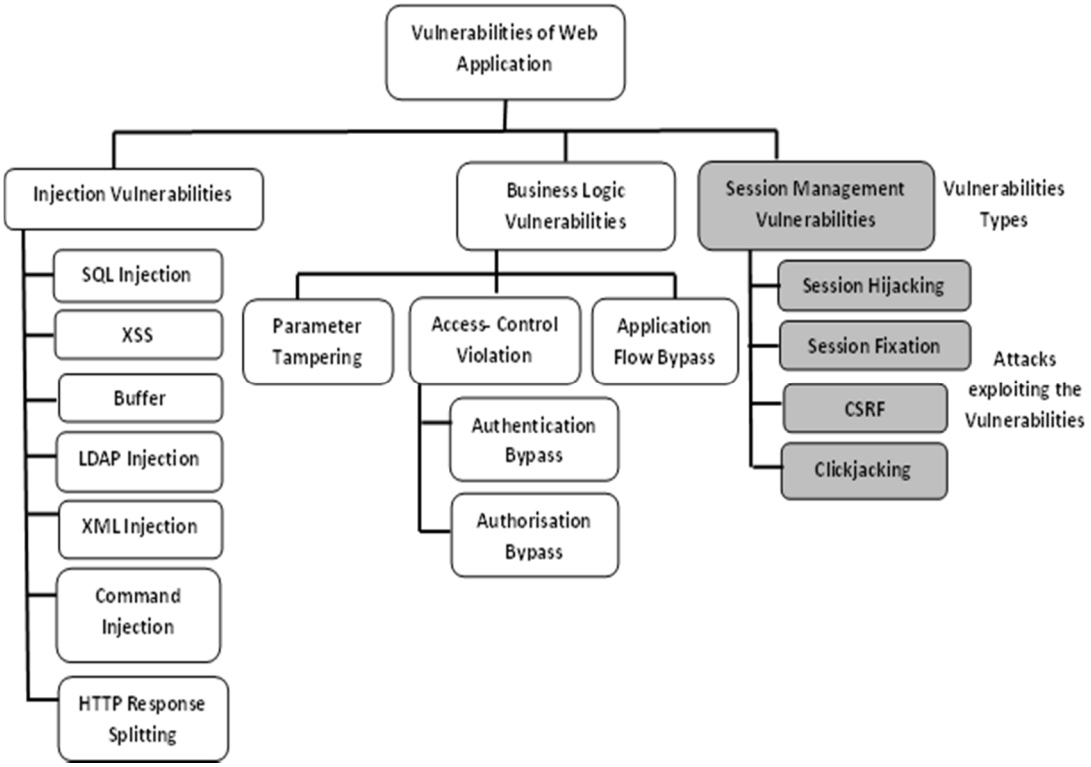

Figure 2-2 The types of vulnerabilities and the attacks exploiting these

2.6. Summary

In this chapter, the most commonly used authentication methods and their attacks have been reviewed. Authentication attacks can be divided into two types, human-based and technology-based. The human attacks can be riskier than the technology-based attack due to the attacker's phishing attempts on the victim; this type of phishing is not a phishing attack but categorised differently, for instance to obtain help. Another example of how human attacks carry more risk than the technology attacks is when employees mistakenly leak information or on purpose. Furthermore, many types of attacks targeting the password are mentioned in this chapter, however, the most widely used and effective means to compromise the user's passwords comprises of the phishing attacks (e.g. when sending a link via email). Other types of attacks require special skills to reveal the password, such as the guessing attacks, social engineering attacks and eavesdropping. Additionally, there are other types of attacks which are based on probabilities, for instance brute-force attacks and dictionary attacks.

The taxonomy of the authentication techniques can be divided into three categories: 1) knowledge-based 2) token-based 3) biometrics. Figure 2-3 illustrates the taxonomy of the authentication techniques and few examples relating to each technique.

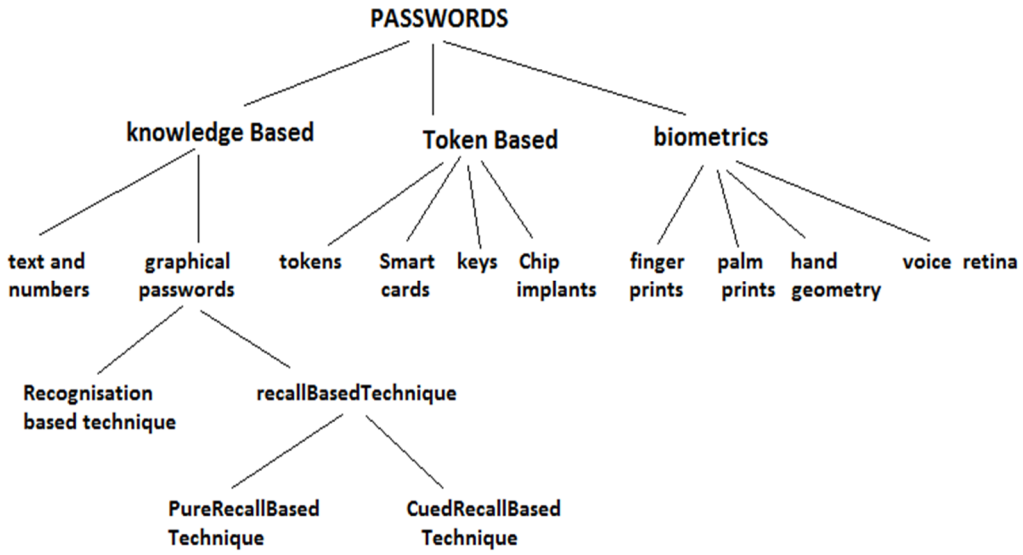

Figure 2-3 Taxonomy of the Authentication Techniques

The text-based passwords are still the main authentication technique that has been used by most users for several reasons. These include the fact that they can be created easily, not requiring additional equipment, being easy to remember and revoked. Thereby, other types of authentications have been emerged. However, text-based passwords are vulnerable for many attacks as mentioned before; therefore, to avoid the weaknesses of these passwords, some techniques were added to passwords to increase their resistance to attacks.

Graphical password and OTP are examples of authentication techniques used by users to prevent certain types of attacks. Biometric techniques are based on either user behaviours (e.g. keystroke password and signature) or on users physiological recognition (e.g. face and retina). Moreover, the honeypots and honeywords are other authentication techniques implemented to detect the attackers and protect the systems. Lastly, a combination between two techniques or more will lead to another technique, referred to as “multi-factor authentication technique”, which benefits from several techniques to increase the strength of the authentication procedure.

Chapter Three: Timed Password Based on Changing User's Behaviours

3.1. Methodology

The Password Quality Indicator (PQI) is used to measure the duration required to find the correct match. Here, the shift-key has been chosen due to it not delivering any action when pressed by a user until am additional key is pressed in combination. Moreover, this key is often used to generate the uppercase letters and some special characters.

A simple survey was carried out to determine the shortest time taken to press the shift-key *Dwell Time* (*Dwelling Period*) using a special program written using Visual C# to collect the data. 261 people were asked to enter different passwords, the length being at least eight characters, and containing three uppercase letters. One further condition for those taking part in the survey was that "there are no two uppercases letters next to each other". Participants were not aware that use of the shift-key and its duration constituted the main goal of the experiment. 4% (11 people) of those surveyed did not use the shift-key to type the required uppercase letters, but instead, employed the "Caps Lock" to type these letters. However, the rest of the participants (250 people) released the shift-key immediately after inserting the required uppercase letter, with the maximum dwell time being recorded at 978 milliseconds. The users who took part were mostly students or staff at Brunel University London. 87.2% (218 people) from 250 participants recorded 120 milliseconds or less, while 29 people (11.6%) recorded between 121- 250 milliseconds. The remaining 3 participants (1.2%) recorded more than 251 milliseconds. Figure 3-1 illustrates the Graphical User Interface GUI for the special program used in the survey; the dwell time shown in this graph is 224 milliseconds.

The new timed password technique keeps all the features of the traditional password, but with an added a dwell time with a special characteristic based on user behaviour changing as a special element. The proposed technique requires two or three uppercase letters or special characters in the password, being typed using the shift-key. The legitimate users have to generate a required period of dwell time when they press

the shift-key to make these special strokes, which is the period between hitting the uppercase letter or special character and letting go of the shift-key.

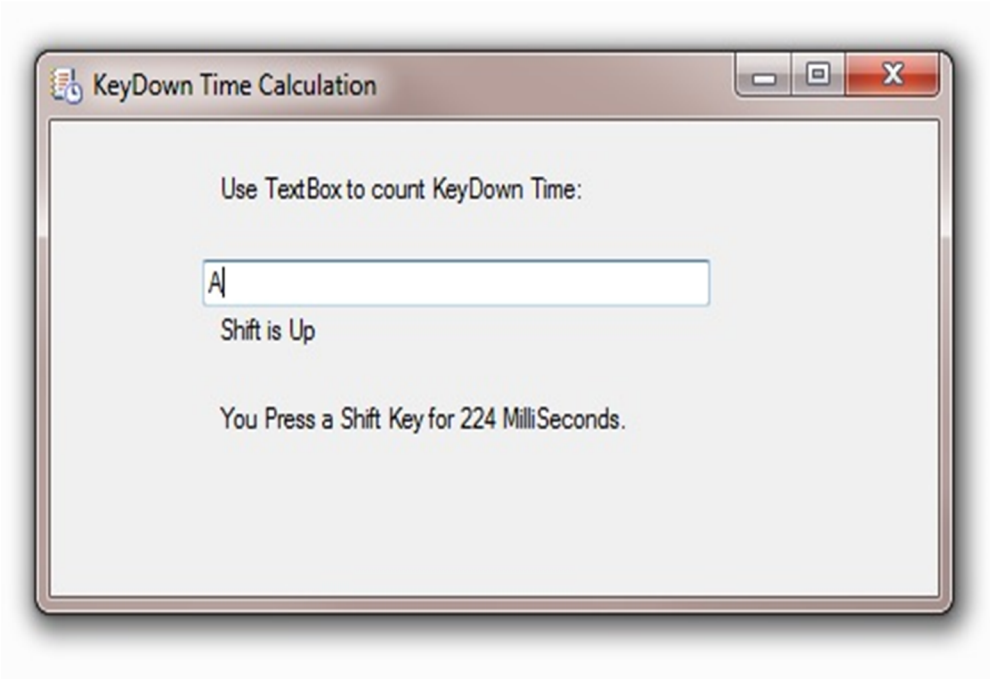

Figure 3-1 The GUI and result of the C# program

As a result of the aforementioned survey outcomes having a maximum time delay of 978 milliseconds, the decision was taken to divide these delays into four groups measured in milliseconds, starting at 2,000 (2,000 - 4,000, 4,001 - 6,000, 6,001 - 8,000 and 8,001 - 10,000). Clearly, these categories can be changed or extended if found inappropriate for the contexts in which they are to be used. If the user wants to type "A" (as a password letter) and the required time for "A" to be authentic is between 4,001-6,000 milliseconds, he/she should hold down the shift-key, type "A" and then wait for this time window to generate the specific dwell time period before releasing the shift key. Admittedly, the proposed password method is not a particularly user-friendly one and people will need to be trained to use it correctly. However, as previously pointed out, it is only aimed at a limited number of users who need access to sensitive datasets. Nevertheless, people using traditional passwords get faster in entering them the more they use them and thus, it is predicted that the new password

will become increasingly easier to use over time. Furthermore, the authorised user can use a real watch (e.g. hand watch or computer watch) to provide the correct dwell time period. Depicted in Figure 3-2, is a flowchart of the timed password technique that will be generated based on the shift-key.

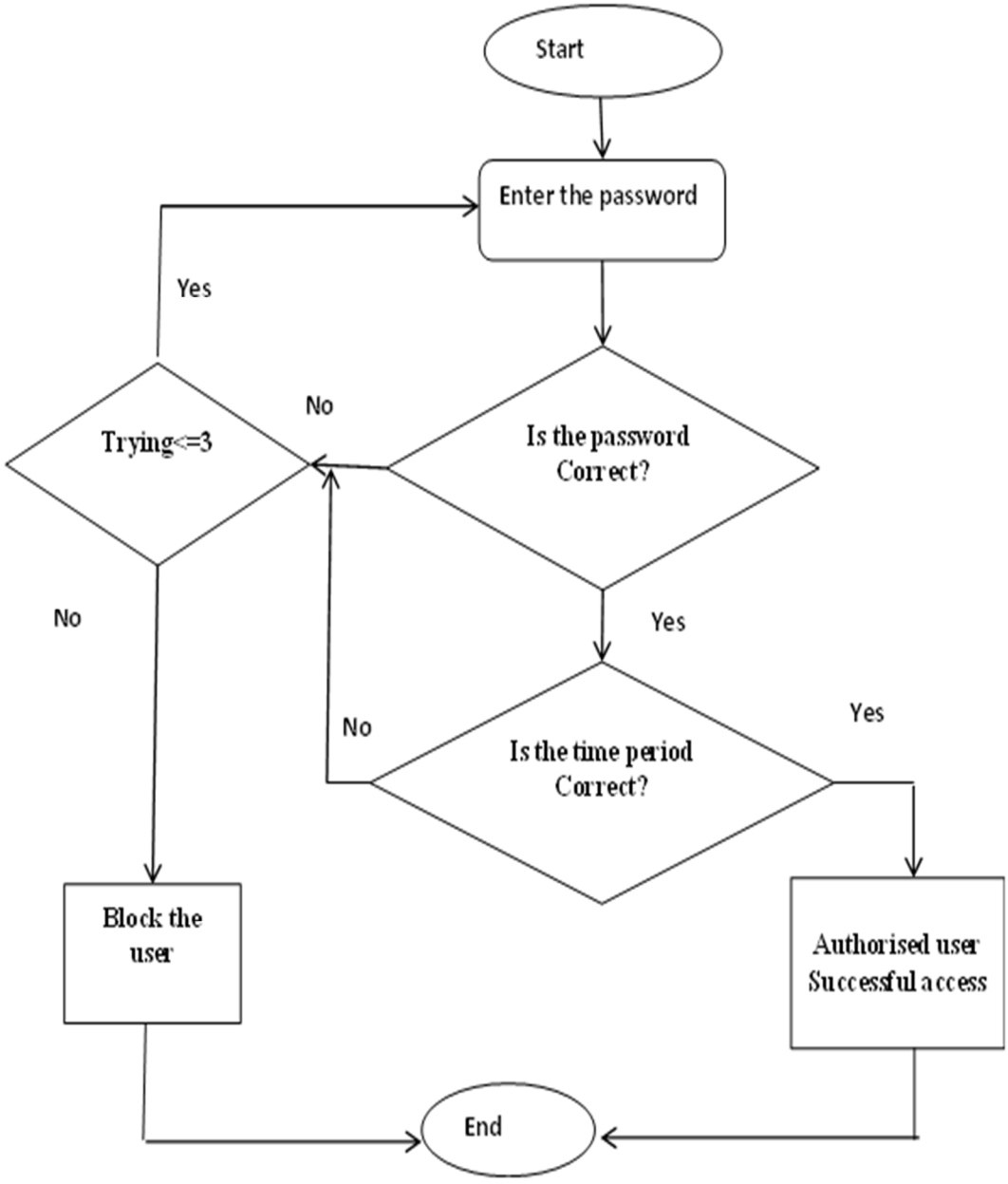

Figure 3-2 A flowchart of the timed password technique

As with the traditional password, the first step is that the username must be entered correctly, and then the password stage will come. In this stage, the authorised user has to go through two conditions, rather than one, as in the traditional password. The first stage is entering the correct password to be matched with the same password that has been saved in the server, which is the same as for the traditional password. The second condition is generating the required dwell time period in the specific positions to be compared with the same positions at time periods for the dwell time that has been saved on the server side. Finally, if the both conditions are met, then the user will be granted access to the sensitive dataset, otherwise, the login process will be rejected.

The main difference between the new password technique and the conventional keystroke technique is that the traditional keystroke technique requires the computer system to interact with human behaviour, learning how each user will type their passwords, whereas the latter does not. Moreover, with the new technique, the user needs to modify their behaviours according to the dwell time period required as explained in this section. The dwell time period will be fixed for each user to be the main part of the password.

3.1.1. Petri Nets Model

Petri nets can be defined as graphical and mathematical modelling tools that can be applied in several systems. Petri nets can be described and studied for the processing of information systems that are considered to be concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. Furthermore, Petri nets is a graphical tool that can be used as a visual-communication aid similar to flowcharts, block diagrams and networks with two main differences. Firstly, Petri nets are dynamic graph discretions and, secondly, they can be used in the event-driven systems. Petri nets can control the system's behaviours via the use of mathematical models [157].

To describe how the new system works, Petri Nets have been developed for each of its stages and Figure 3-3 illustrates the Petri Nets model for the new authentication method when just one character is to be entered. Clearly, the transactions that have orange shadow borders (t3, t4, t10, t11, t12, t5, t6) are in a

conflict situation and hence, either t3 or t4 could be firing when the token is in P3, but not both. Similarly, when the token is in P4, then only one of t10, t11, or t12 could be firing. In fact, the transaction filled in orange is the first that will be fired. Usually, Petri nets are formally defined as a 5-tuple N = (P, T, I, O, M0) and these tuples can de described in this system as follows:

1- $P = \{p_1, p_2, p_3, \ldots, p_{12}\}$ is a finite set of places;

2- $T = \{t_1, t_2, t_3, \ldots, t_{13}\}$ is a finite set of transitions, $P \cup T \neq \emptyset$, and $P \cap T = \emptyset$;

3- I: $P \times T \rightarrow N$ is an *input function* that defines directed arcs from places to transitions, where N is a set of nonnegative integers;

I: {P1⟶ t1, P2 ⟶ t2, P3 ⟶ t3, P3 ⟶ t4, P4 ⟶ t10, P4 ⟶ t11,
P4 ⟶ t12, P5 ⟶ t5, P5 ⟶ t6, P8 ⟶ t8, P10 ⟶t6, P11 ⟶ t5,
P12 ⟶ t13, P6 ⟶ t7, P9 ⟶ t9};

4- O: $T \times P \rightarrow N$ is an *output function* that defines directed arcs from transitions to places

O: {t1⟶ P2, t2 ⟶ P3, t2 ⟶ P4, t3 ⟶ P5, t4 ⟶ P8, t10 ⟶ P10,
t11 ⟶ P11, t12 ⟶ P12, t5 ⟶ P6, t6 ⟶ P6, t8 ⟶ P9, t13 ⟶ P9,
t7 ⟶ P7, t9 ⟶P7}

5- M_0: $P \rightarrow N$ is the *initial marking*.
$M_0 = \{1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\}$

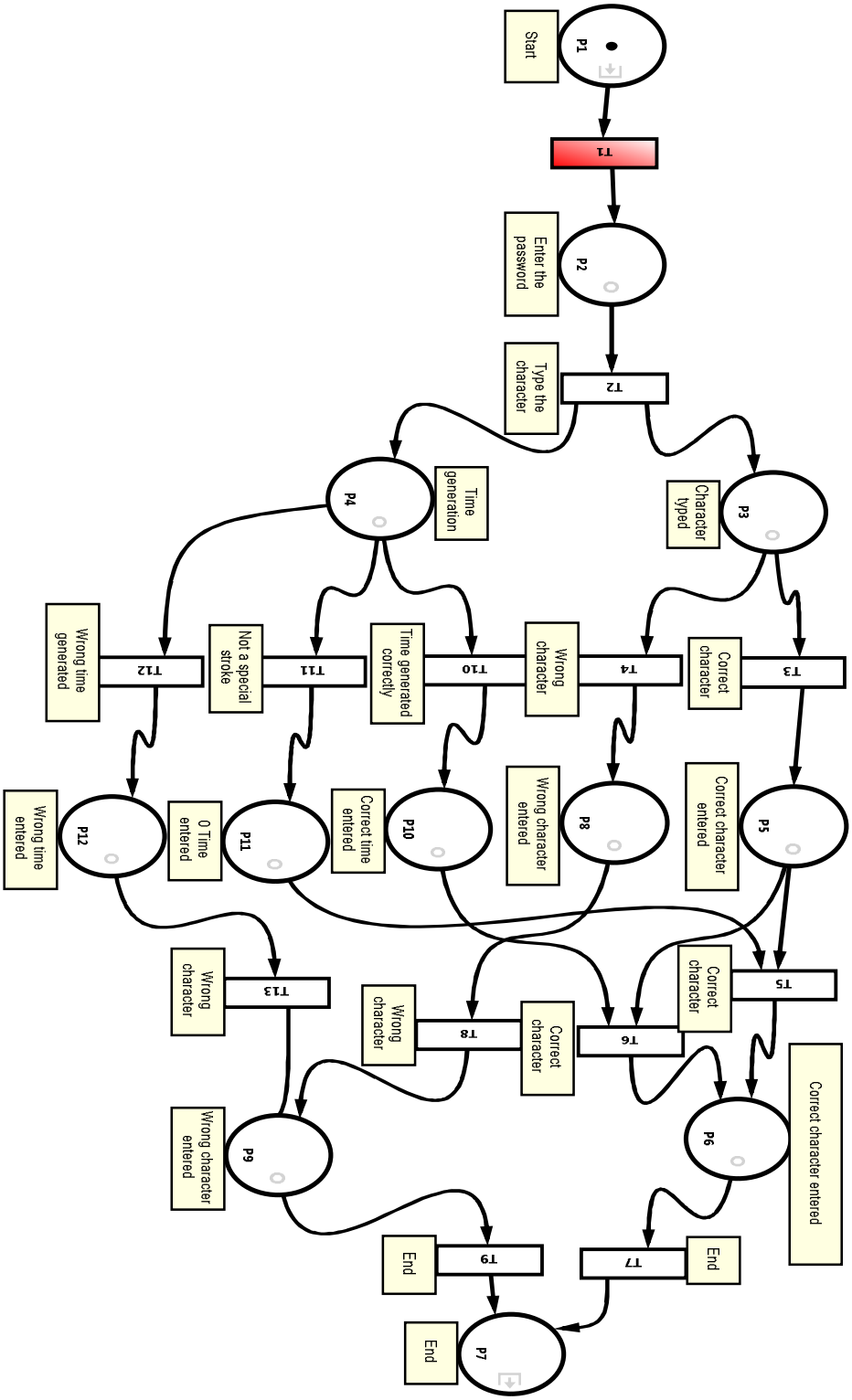

Figure 3-3 The Petri Nets model for checking when one character is entered

3.2. Password Strength Measurement

The strength of the traditional password will be compared with the new timed password using Password Quality Indication (PQI) for measurement.

3.2.1. Traditional Password Strength Measurement

3.2.1.1. Entropy of Guessing a Password

The term “entropy” is taken from information theory, being defined as the degree of "uncertainty" or "randomness” of a phenomenon. Accordingly, “password entropy” is used to measure the level of security of a password, i.e. the higher the entropy the greater the security [158]. Password-strength judgement, when accurate, is essential when trying to make security and safeguarding more efficacious. Password strength, in the past, has generally been determined by informational entropy. However, this method can be inadequate in ascertaining or capturing resistance, when it comes to intelligent hacking through guessing attempts. Password guessability, according to more contemporary research, is the passwords capacity to endure and withstand guesses through a certain password cracker through security metrics, or other training data. Regarding the benefits of this method, one notable observation is that of modelling knowledge that a real-life adversary could have employed, in addition to potential bounding attempts that may have been used; however, the selected setup determines the quality of the results. That is, the number of guesses that are needed before a certain algorithm is cracked and the extent of training setup required before the particular password can be reached are determining factors [159].

Nowadays, because they are unable to reflect the ease by which a password can be practicably cracked, the more traditional entropy metrics have become less popular. Instead, the simulation or running of a certain cracking algorithm, as provided with a perimeter through several types of training data, is now utilised to determine the strength of a password. There are two main benefits from such a method: firstly, it can determine, though calculation, the password’s ‘guessability’ on an individuated basis, facilitating data-driven estimations as to its strength as the password itself is generated. Secondly, the method can estimate real-world security

from existing real data, not merely idealised, adversarial methods and processes. The simulated algorithm used for cracking is not sufficiently trainable to outwit a real attacker, which constitutes one of the method's shortcomings, for this can result in inaccurate predictions and estimations concerning the security and strength of the password in question [160].

3.2.1.2. Password Quality Indicator

Several types of password attack exist; essentially, various arrays or combinations of certain characters are used, with the aim of securing a successful one, such that a password can be successfully matched, as part of the password-attacking process. Several strategies need to be in place for a password to be successfully cracked. To start with, generally, certain notorious and uncomplicated combinations will be utilised before all password combinations are employed as a brute-force assessment of the potential password candidates. The order below gives the most commonly seen and used password cracking route:

1. Existing terminologies words are taken from the dictionary;
2. Using singular and perhaps double character variations of these dictionary words;
3. Endeavouring to enumerate every potential smaller character set, options and varieties, for instance, utilising just-all-lowercase or just-all-uppercase letters, then adding digital characters,
4. Employing a brute-force enumeration method for every potential candidate commination for the password utilising a full character set (that is, a total of 93 characters).

The time it takes to determine the correct password match represents a password's efficacy. That is, the longer the time taken to find a match, then the more robust the password is against being cracked. Hence, the extent to which the password deviates from words featured in the dictionary, its length and the size of the password set can be utilised to determine the quality of a given password. A precise measurement as to the disparity between two things can be ascertained through Levenshtein's editing distance, which is able to calculate the minimal number of individual character manipulations needed to determine the distance between two similar entities. This is determined through deletion, insertion and modification until

the two become the same. An example can be shown as: the difference, or distance, between the word “see” and “bee” is a single letter, and thus there is a value of 1, whereas “more” and “lure” have a dissimilarity of 2. So that the difference between a password and extant words in the dictionary can be measured, the dictionary words are compiled, after which the Levenshtein’s editing distance between the password and all these lines is ascertained. The distance between these base-level words found in the dictionary and the password is considered a being the minimum. The number of characters that feature within the password determines the password’s length; this is essential when determining the time-frame needed before a password to be successfully cracked.

According to the PQI, a password is made of characters, which are from certain groups, e.g. all characters, lower case alphabet characters, or digit characters, which are character sets. The 93 printable characters on the British keyboard can be grouped into four sets:

1. 26 lower case letters (abcdefghijklmnopqrstuvwxyz).
2. 26 uppercase letters (ABCDEFGHIJKLMNOPQRST UVWXYZ).
3. 31 special characters (~!@#$%^&*()_-+={}[]|\: ; ”<> ? ’, . /).
4. 10 digits (0 1 2 3 4 5 6 7 8 9).

The Password Complexity Index (PCI) of a password entirely derived from the characters in set 1 is 26, whilst for set 2 it is 26, for set 3 it is 31 and for set 4 it is 10. The PCI is additive, so, for instance, a password made up characters from set 1 and 4 has a PCI value of 26+31=57. For a password Ω, which has a PCI value C and length m, the number of all Possible Password Candidates (PPC) of the same format is:

$$PPC = C^m. \quad (3\text{-}1)$$

To have the same number of password candidates in standard password format, with a PCI value of 10, it is necessary to find out the length (L) of the password candidates in this format and thus, this becomes:

$$C^m = 10^L. \quad (3\text{-}2)$$

Therefore,

$$L = m * \log_{10} C. \quad (3\text{-}3)\ [161]$$

where, L is the effective length of password P, m is the length of password P, and C is the PCI.

The PQI of a password is a pair λ = (D, L), where D is Levenshtein's editing distance of the password to the base dictionary words and L is the effective password length. When D ≥ 3 and L ≥ 14, this is pertains to a good password. D ≥ 3 means that the password is at least 3 characters different from base dictionary words, and L ≥ 14 means that there are at least $\mathbf{10^{14}}$ possible candidates to be tried to crack the password [161].

In this thesis, the passwords have been analysed using an entropic approach which is well suited to dealing with passwords generated at random. When the password is determined in a more deterministic manner, then it is difficult to analyse the generation of the password analytically.

3.2.2. A New Password Strength Measurement Technique

Having explained the traditional password measurement by using the PQI mentioned in the previous section (3.2.1.2), the mathematical equations need to be modified to account for the time element in the new password technique. The number of characters that can be candidates when using the shift-key is 42, i.e. all uppercase letters, 26, and 16 special characters (~ ! @ $ % ^ & ( ) _ | : ” <> ?). Three equations are applied for the new method, one with fixed positions and two variant ones. Regarding the latter, one has two variant characters, whilst the other has three.

3.2.2.1. New password Technique with Two Fixed Positions

One way to apply the new technique is to use the shift-key with characters in two fixed positions within the password. So, for instance, the shift-key could be applied to the characters in the first and fourth positions in the password, so the first and fourth characters can have any of 42 possible uppercase characters and any one of l possible time periods. Hence, the characters are drawn from a character set of $42l$. The fixed position technique will be easily remembered by all the users due to the same position being repeated in all passwords. However, a fixed position password will be easier to break compare to one with variant positions. For a new password Ω^*, the difference between the effective length of traditional password L and the effective

length of the new password L^*, i.e. $\Delta \boldsymbol{L}$, can be calculated. To calculate the value of $\Delta \boldsymbol{L}$, for finding the password length with the most effective strength, $\Delta L = L^* - L$.

Characters in every position $=10^L$

For a standard password with effective length L and total characters available G, we have

$$G = 10^L. \tag{3-4}$$

If the password has length m and the characters are drawn from a character set C then $G = C^m$.

For the new password, with the same character set and length:

$$G = C^m,\ G^* = C^{m-2}(xl)^2$$

$$\Delta L = L^* - L$$

$$\Delta L = \log_{10} G^* - \log_{10} G$$

$$\Delta L = \log_{10} C^{m-2}\,(xl)^2 - \log_{10} C^m$$

$$\Delta L = \log_{10}\left(\frac{C^{m-2}(xl)^2}{C^m}\right) \text{ [since } \log_{10} a - \log_{10} b = \log_{10}\frac{a}{b}\text{]}$$

$$\Delta L = \log_{10}\left(\frac{xl}{C}\right)^2$$

$$G^* = 10^{L^*} \text{ and } G^* = C^{m-2}(x\,l)^2. \tag{3-5}$$

Hence

$$\Delta L = L^* - L = 2\log_{10}\left(\frac{x\,l}{C}\right). \tag{3-6}$$

3.2.2.2. Variant Positioning with Two Special Cases of the New Password Technique

Variant positioning gives legitimate users more flexibility when using the new password technique. Moreover, different positions increase the unlikelihood of the password being cracked. The final equation for measuring the strength of this type of new password is:

$$G = C^m,\ G^* = \binom{m}{2} C^{m-2}(xl)^2 \tag{3-7}$$

$$\Delta L = L^* - L$$

$$\Delta L = \log_{10} G^* - \log_{10} G$$

$$\Delta L = \log_{10}\binom{m}{2} C^{m-2}(xl)^2 - \log_{10} C^m$$

$$\Delta L = \log_{10}\left(\frac{\binom{m}{2} C^{m-2}(xl)^2}{C^m}\right)$$

$$\Delta L = \log_{10}\left(\frac{m(m-1)(m-2)!}{2!\,(m-2)!}\left(\frac{xl}{C}\right)^2\right)$$

$$\Delta L = \log_{10}\left(\frac{m(m-1)}{2}\left(\frac{xl}{C}\right)^2\right)$$

From equation (3-5) $G^* = 10^{L^*}$ then

$$\Delta L = log_{10}\left(\left(\frac{xl}{C}\right)^2 \frac{m(m-1)}{2}\right) \quad (3\text{-}8)$$

3.2.2.3. Variant Positioning with Three Special Cases of the New Password Technique

The same technique as above is applied, but here the number of special keystroke characters using a time period is three with variant positions, instead of two. The final equation for measuring the strength of this type of new password is:

$$\Delta L = log_{10}\left(\frac{(x\,l)^3}{C^3} \frac{m(m-2)(m-1)}{3!}\right) \quad (3\text{-}9)$$

3.3. The Results

MATLAB has been used to calculate all the above equations for:

- Traditional password (Equation 3-3)

$L = m * \log_{10} C$.

- New password technique with fixed positions (Equation 3-6)

$\Delta L = 2\left[\log_{10}\left(\frac{x\,l}{C}\right)\right]$.

- Variant positions with two special cases of the new password technique (Equation 3-8)

$$\Delta L = log_{10}\left(\left(\frac{xl}{C}\right)^2 \frac{m(m-1)}{2}\right).$$

- Variant positions with three special cases of the new password technique (Equation 3-9)

$$\Delta L = log_{10}\left(\frac{(x\,l)^3}{C^3}\frac{m(m-2)(m-1)}{3!}\right).$$

The results are presented according to the C values of relevance to this research and the total number is eight, based on uppercase letters being part of the password, as follows:

- When C=26: password contains uppercase letters (Figure 3-4).
- When C=36: password contains uppercase letters (26) + digits (10) =36 (Figure 3-5).
- When C=52: password contains uppercase letters (26) + lower case letters (26) =52 (Figure 3-6).
- When C=57: password contains uppercase letters (26) + special characters (31) =57 (Figure 3-7).
- When C=62: password contains uppercase letters (26) + digits (10) + lower case letters (26) =62 (Figure 3-8).
- When C=67: password contains uppercase letters (26) + digits (10) + special characters (31) =67 (Figure 3-9).
- When C=83: password contains uppercase letters (26) + lower case letters (26) + special characters (31) =83 (Figure 3-10).
- When C=93: password contains uppercase letters (26) + lower case letters (26) + special characters (31) + digits (10) =93 (Figure 3-11).

The x-axis below illustrates the length of the password (m), and the possible password candidates needing to be tried to crack the password are represented by y-axis (shown with log scale).

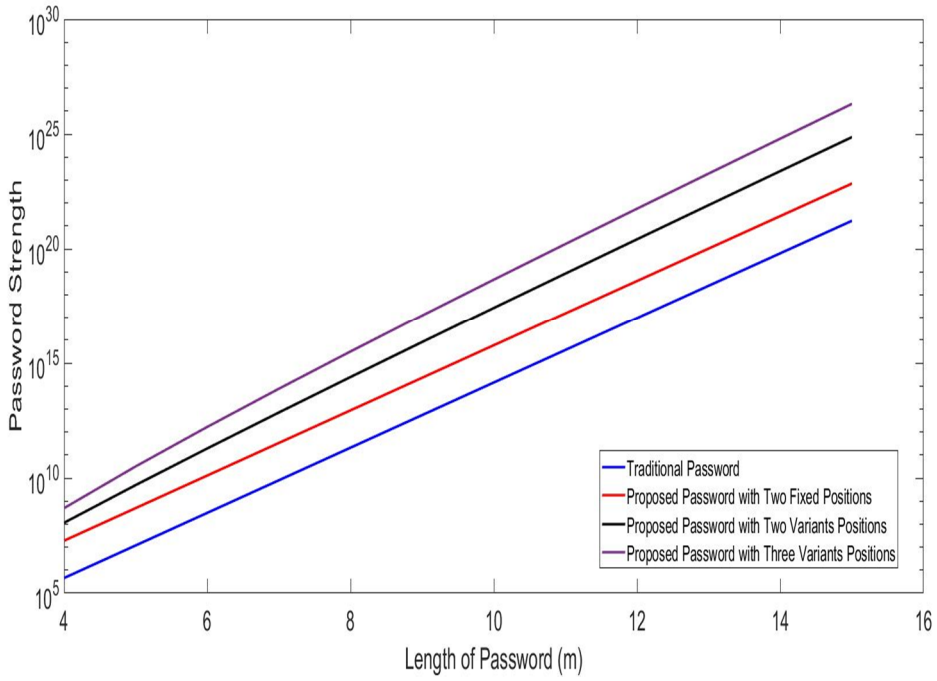

Figure 3-4 C=26: password contains uppercase letters

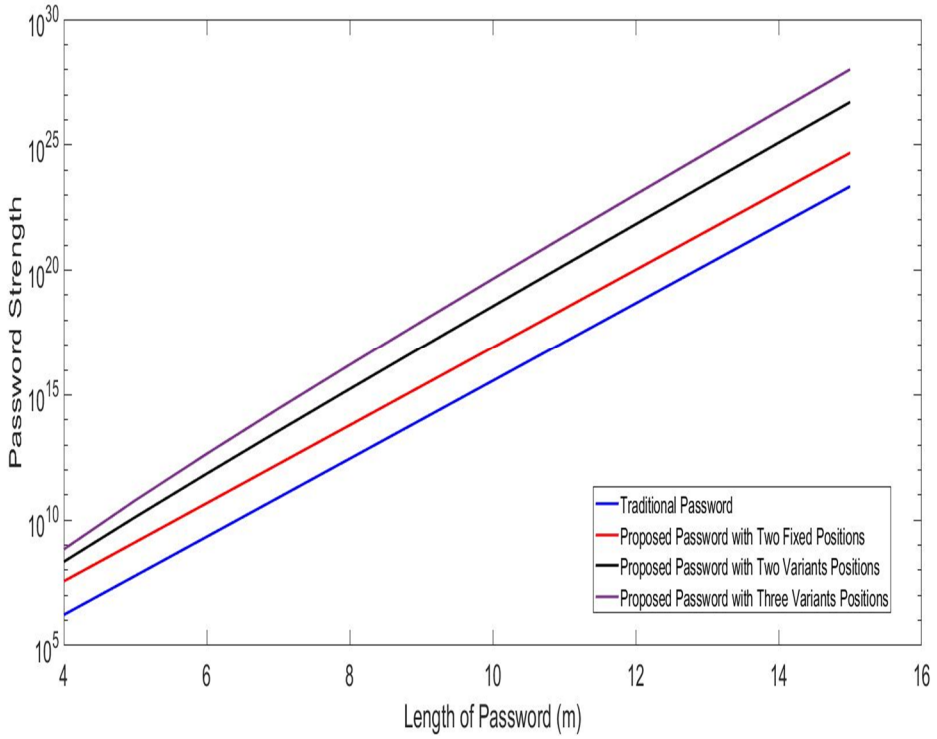

Figure 3-5 C=36: password contains uppercase letters (26)+digits(10)=36

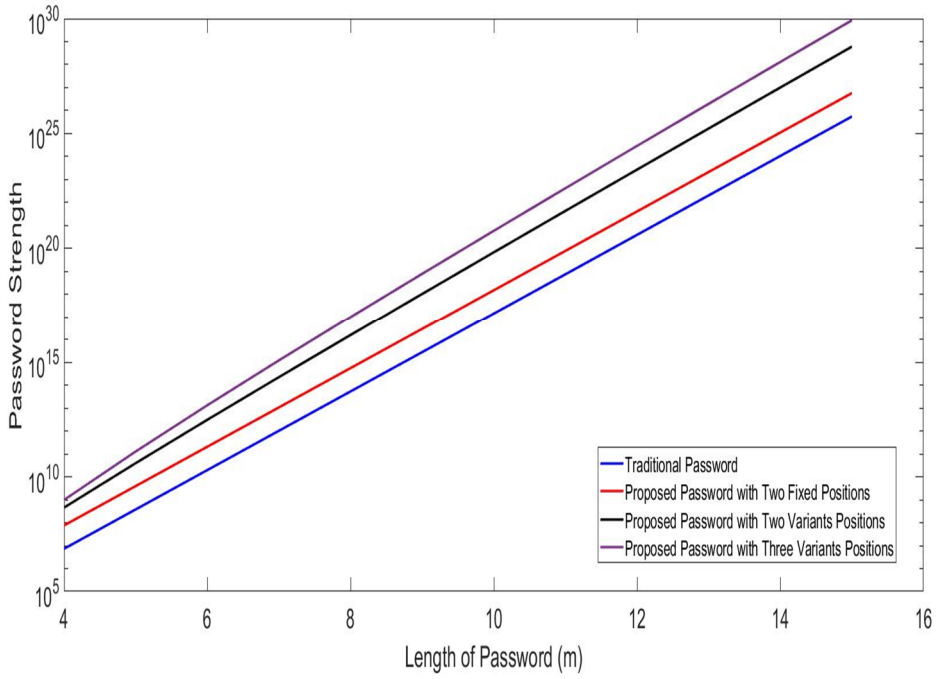

Figure 3-6 C=52: password contains uppercase letters (26) + lower case letters (26)=52

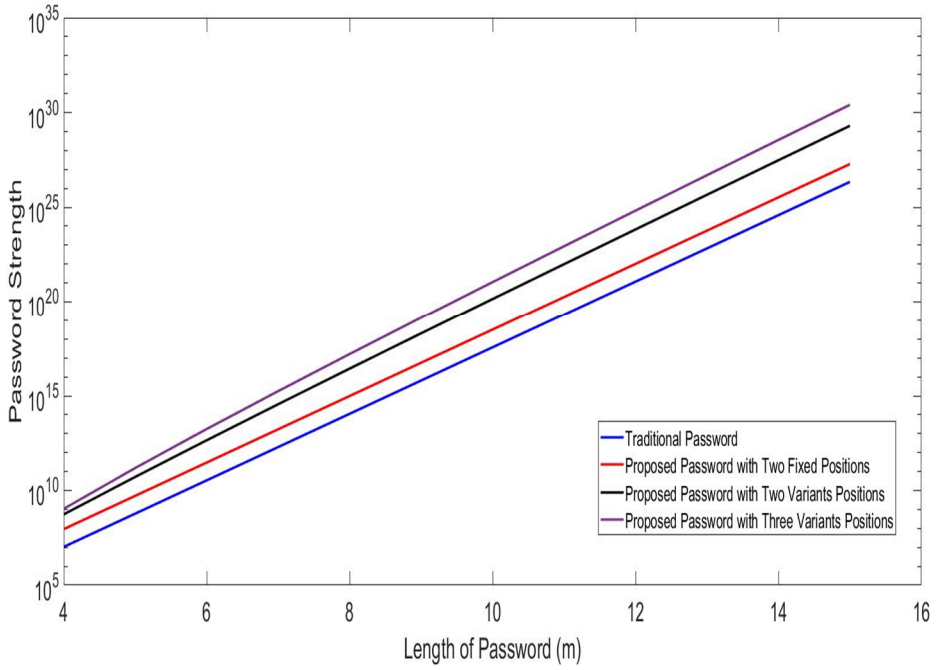

Figure 3-7 C=57: password contains uppercase letters (26)+ special characters (31)=57

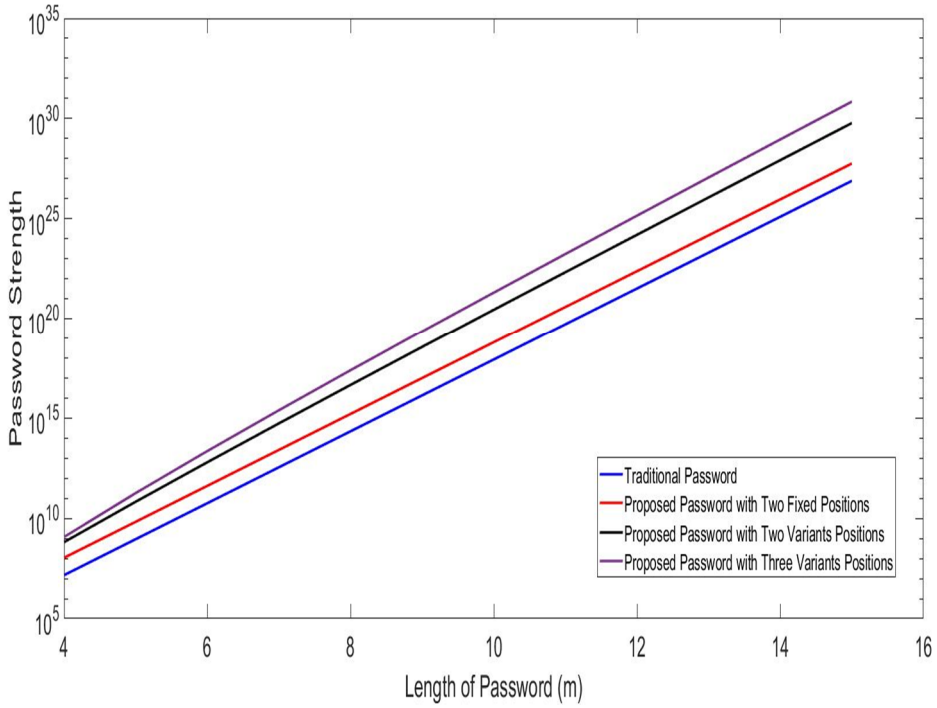

Figure 3-8 C=62: password contains uppercase letters (26)+ digits (10)+ lower case letters (26)=62

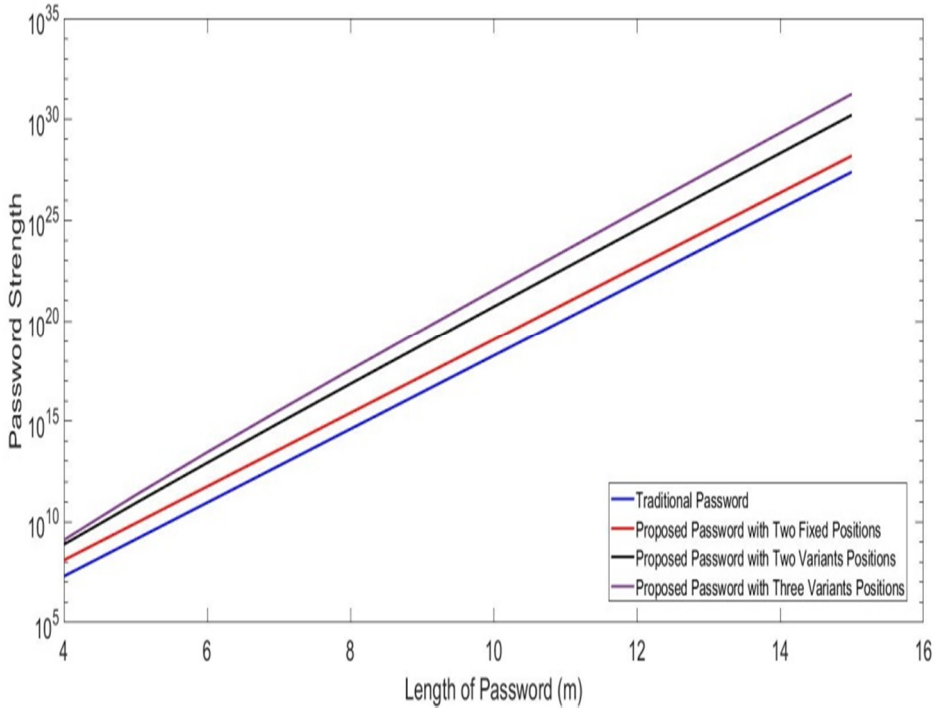

Figure 3-9 C=67: password contains uppercase letters (26)+ digits (10)+ special characters =67

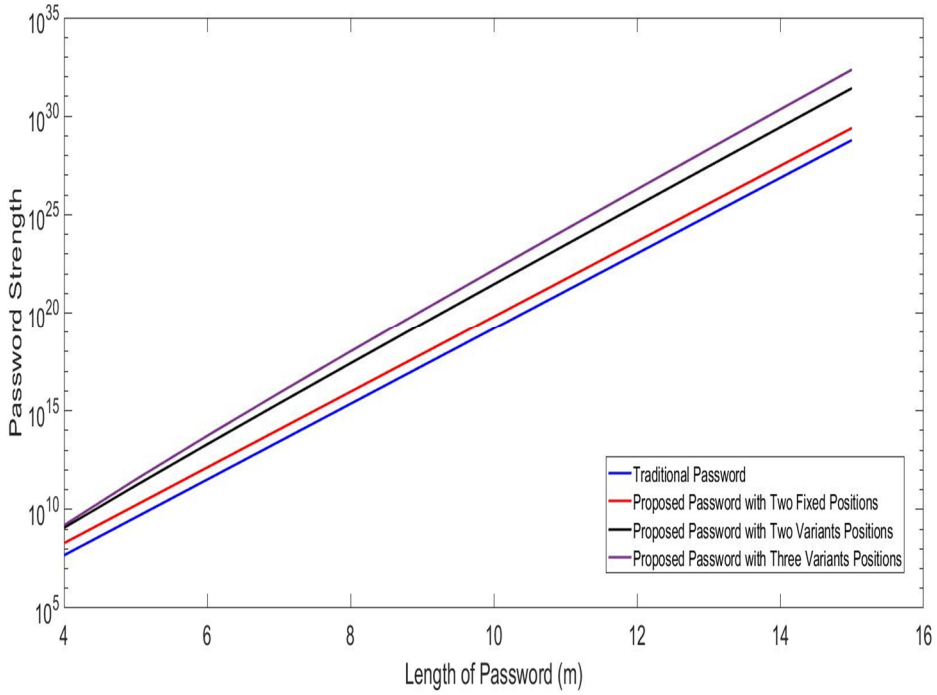

Figure 3-10 C=83: password contains uppercase letters (26) + lower case letters (26) + special characters (31) =83

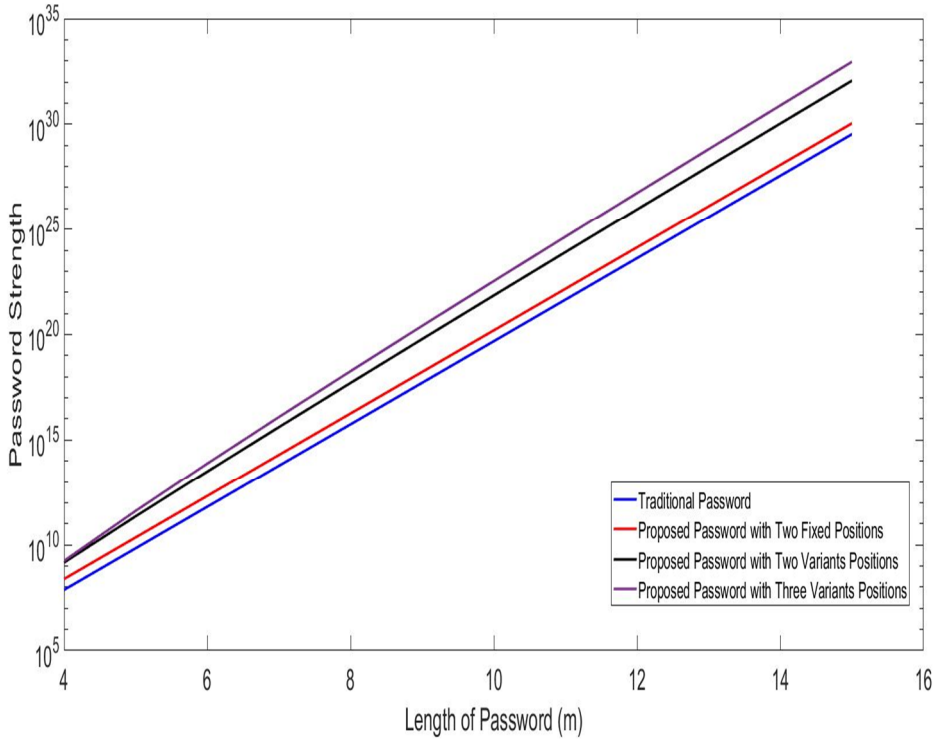

Figure 3-11 C=93: password contains uppercase letters (26) + lower case letters (26) + special characters (31) + digits (10) =93

3.3.1. The Difference Percentage Equation (DPE)

The Difference Percentage Equation (DPE) has been used to calculate by how much the new password technique is better than the traditional one.

$$\text{DPE} = \left(\frac{|V1-V2|}{(V1+V2)/2}\right) * 100\%. \quad (3\text{-}10)$$

V1: The value of the traditional password.

V2: The value of the new technique password.

Table 3-1 illustrates when equation (3-10) was applied to the password when C = 93, with the following results being obtained.

Table 3-1 Strength comparison results between the new password and the traditional one (by percentage) when DPE was applied and C = 93

Password Length (m)	Fixed Positions %	Two Variant Positions %	Three Variant Positions %
4	106.175	180.563	183.726
5	106.175	188.107	193.328
6	106.175	191.992	196.636
7	106.175	194.247	198.071
8	106.175	195.670	198.792
9	106.175	196.624	199.194
10	106.175	197.295	199.435
11	106.175	197.784	199.589
12	106.175	198.151	199.692
13	106.175	198.435	199.763
14	106.175	198.658	199.814
15	106.175	198.836	199.851

Other values of C give quantitatively similar results.

3.3.2. Results Analysis

From the results, it can clearly be seen that the most secure password is when C = 93, involving a combination of, uppercase letters, lower case letters, special characters and digits, for all three options tested, which is to be expected. However, the arrangement with the two fixed position characters only performed 106% better than the conventional method. By contrast, both the variant positioning setups had security values impressively almost double the traditional password technique. Specifically, the

two variant types ranged from 180.5% to 198.8% over the conventional method, according to password length and the corresponding interval for the three variant arrangements was 183.7% to 199.9%. The latter, although impressive, required a password length of 15 characters, which obviously would be difficult to memorise and would be more likely to be entered incorrectly than a shorter one. Notably, an eight letter character password performs only 0.3% worse than one with 15. Taking these observations into account, it would appear that a password of around eight characters is the optimum for the proposed new technique. Moreover, this number of characters is one that a high proportion of users choose for the traditional password method.

3.4. Analysis of Password Attacks

Regarding the common password attacks that have been discussed earlier in chapter two, the effect of these on the new timed password will be discussed in this section. According to the brute-force attack, all the attempted combination possibilities will be implemented based on a trial-and-error process. However, when the time is added as part of this password, this number of trial-and-error becomes substantially greater. Furthermore, the same problem will be faced with the dictionary attack, also due to it not being concern about the time as part of the password.

Phishing attacks that lure the victim into entering his/her password or providing sensitive information as a form of social engineering have proved to be very effective. Nevertheless, phishing attacks will not work on the new timed password, because it is based on a specific time period (dwell time) for pressing the shift-key and even though the adversary may have obtained the real characters of the password, he/she will not know this duration of time. The same problem will occur with guessing attacks and eavesdropping, whereby the generation a dwell time period could prevent these attacks from being successful. For a shoulder-surfing attack, the attacker will need to be very focused on how the authorised user is typing the password. However, it will be very difficult to ascertain the dwell period (the specific time period) due to the adversary being unlikely to detect how keys are being entered determining the correct use of the password.

3.5. Summary

In this chapter, a new password technique that incorporates time has been presented. Specifically, it relies on the dwell time, whereby the user spends a specified time between typing an uppercase letter or special character and releasing the shift-key as part of the password entry process. The aim is to increase the possible combinations for the password, thereby improving security when compared to traditional measures. First, a survey was carried out to determine the appropriate settings for the new password system. Clearly, this method is more complex than the traditional one, but it is not aimed at the general user. That is, it is targeted at those who require access to sensitive datasets needing high levels of security. They will need to be trained to be able to take the specified time interval for releasing the shift-key, as explained.

A mathematical model has been built and the effective length of password was used to compare the outcomes generated by the model with those from the traditional password method. The difference percentage equation was employed to demonstrate that the new password technique is better than the traditional one. The results of the mathematical model have provided evidence that the new password technique is better than the traditional one by nearly a factor of two (199.851%) in the scenario when three variant uppercase letters or special characters were used to form the password at a length of 15. In fact, all three arrangements tested using this technique provided security outcomes that were substantially better than when using the traditional method. Two fixed positions gave the permanent improvement value 106.175% for all passwords length.

Chapter Four: Password Authentication Based on Honeypot Session Management with Web Page Links.

4.1. Methodology

A high interaction honeypot is developed based on website session management to achieve full tracking for each user (sequence of links). Specifically, the session management is worked as a honeypot to detect attacks by being a part of the authentication process with the login box given those intruders often use this to compromise passwords. In addition, the level of security can be increased by making the number of possibilities for creating the session very high when the user tries to gain access to modify a sensitive dataset. For each user, there is a username u_i, password p_i and session s_i, which are combined together to achieve authorised access. Table 4-1 illustrates a comparison between two authentication files on the server side, with the first being the traditional one (Table 4-1-a), whilst the second is the proposed file (Table 4-1-b). The authentication procedure is divided into two parts, with the first involving the creation of a correct session based on the order of the link sequences embedded within it. However, this session will only be created, if the first webpage has been demanded by the user, whilst if the adversary goes directly to any webpage other than the first (main webpage) through the URL, then the session will not start and the value "0" will be sent to the server, thereby identifying an imposter who is trying to login.

Table 4-1 Traditional authentication method and the new method saving on the server side

Username	Password
u_1	p_1
u_2	p_2
. .	
u_n	p_n

a. Traditional information on the server side

Username	Password	Session Management
u_1	p_1	s_1
u_2	p_2	s_2
. .	. .	. .
u_n	p_n	s_n

b. New information on the server side

Each link on the webpage has a dedicated number in ascending order from 1 to n on each webpage. If the user starts from the first webpage and clicks on the link, then at this moment, the order number of this link will be added to the session that was created before. The legitimate user will have to create the session by requesting the main webpage, followed by going through the three specific orders of the link sequences (four webpages). In the end, the session will give the information about the full tracking of browsing in the webpages, according to the links that have been clicked on by this user, to move from webpage to another. The second part takes the form of the traditional way of logging in, that of the user name u_i and password p_i. On the server side, each legitimate user wanting to have access to the database must do so by providing three components: a unique username u_i; a traditional password p_i; and the session s_i. That person must complete all the component information correctly and this is checked to see whether it is compatible with the information sent from the client side before giving this user access to modify the database.

Moreover, the login box will appear as part of all the webpages, the main purpose of which is to enter the user ID and password. However, there is another objective for this arrangement, which is a honeypot used to lure an adversary to log into the system, but fails to use the correct sequence of the session management. For, usually, an adversary uses the login box directly when he/she tries to hack into the system with what he/she believes to be the correct password. Hence, a high interaction honeypot will make for easier attacker detection as this type is integral to the real system and consequently, much less likely to be identified as such by an adversary than with a low interaction one.

In addition, starting the session from the first webpage gives the system an additional level of protection, because the adversary will not be able to create the session and hence, the whole authentication system will fail. This procedure will help an Intrusion Detection System (IDS) to suspect and detect intruders. For instance, if the user performs login on the first webpage, then the full tracking for the session will be non-existent, because no link has been chosen from the first webpage. In this case, the session will be sent to the server without any number being registered regarding it. In contrast, if the user clicks the links without the correct sequence, then the session management will be sent without the right number of links, which will not be

compatible with session management s_i for this username u_i and password p_i in the information saved on the server side and so, the login process will be denied.

Obviously, the links for each webpage are all fakes except for that which is part of the authorisation sequence for the legitimate user's password. In other words, the rest of the links could be the initial part of the password of another user, who will need to follow a different pre-established sequence. However, as explained, the legitimate user has to go through the correct sequence of three links before trying to enter the username and password. In addition, the IDS has the responsibility of monitoring the session management of the user through the links of the website during the login process. To achieve this, it checks the sessions of the webpages (full tracking) for each user. The type of IDS used in the proposed system is signature based (misuse detection) rather than anomaly based, because it knows the pattern of the session management for each user from when it was created. Consequently, this system just has to match the session management that has been sent from the client side (user) with that saved on the server side. Hence, the patterns of attacks relating to the session management are predefined from the beginning. Each legitimate user has a specific sequence of links that must be gone through before entering his/her ID and password into the login box during the login process. That is, a legitimate user has to log into the system correctly through the following procedure.

1- The first step is a request for the required website by the user from the server.
2- The server will respond to this request and there are three links in the specific sequence for each user. One will be chosen from each webpage, each of which has n links, one being the correct one and the remainder are fakes for this user. Nevertheless, as aforementioned, one of these links could be a part of the correct sequence for other users. In addition, another user has a different sequence of links, so the link that has been specified for the first user might well be fake for a second user, given the different link sequence for each.
3- The session management (tracking the links) will be started on the first (main) webpage and at the moment when the website is on the client side. Java script is used to carry this out.
4- The first specific link will be chosen and clicked on from the main webpage, at which moment its order number of will be saved and added to the session management.

5- The second specific link will be chosen and clicked on from the webpage that appears after clicking (chosen) the first link, the order number of which will be saved and added to the session management along with the first link that was selected in step 4.

6- The third specific link will be chosen and clicked on from the webpage that appears after clicking (chosen) the second link, with the order number of this link being saved and added to the session management along with the first and second ordered links that were selected in steps 4 and 5.

7- At this step, after choosing the order of the sequence of links, the user should enter the correct user name (ID) and password into the login box.

8- The session management s_i that has been created during the steps from 3 to 6, username u_i and password p_i will be sent to the server to check whether they are correct or not.

9- The user will have access to the original database if the session (sequence of links), ID and password are correct.

10- Each authorised user has a permutation to follow and when just one link out of the three link sequence is incorrect, for instance, missing the order of the sequence inside the session management, then the legitimate user will need to go back to the first (main) webpage. At this point, an uppercase letter "A" will be added to the session management as an indicator character for the first webpage. At the end, when the server checks the session management $\boldsymbol{s_i}$ that was created during the sign up process, if it is correct, the user will have the access to the original database. Otherwise, he/she will have access to the fake one, if he/she has been able to breach the password, rather than gaining access to the original database.

11- The main idea behind this procedure is to make detecting an adversary easier, for if he/she is able to steal the username u_i and password p_i, then he/she will try to login in to the login box without being concerned about the links and order of the session management s_i for the user i. That is, access to the session will not be successful and the fake database will appear, with the original one being locked, despite the imposter having u_i and p_i. Figure 4-1 illustrates the authorised login procedure that must be followed by the legitimate user to get access to the genuine database.

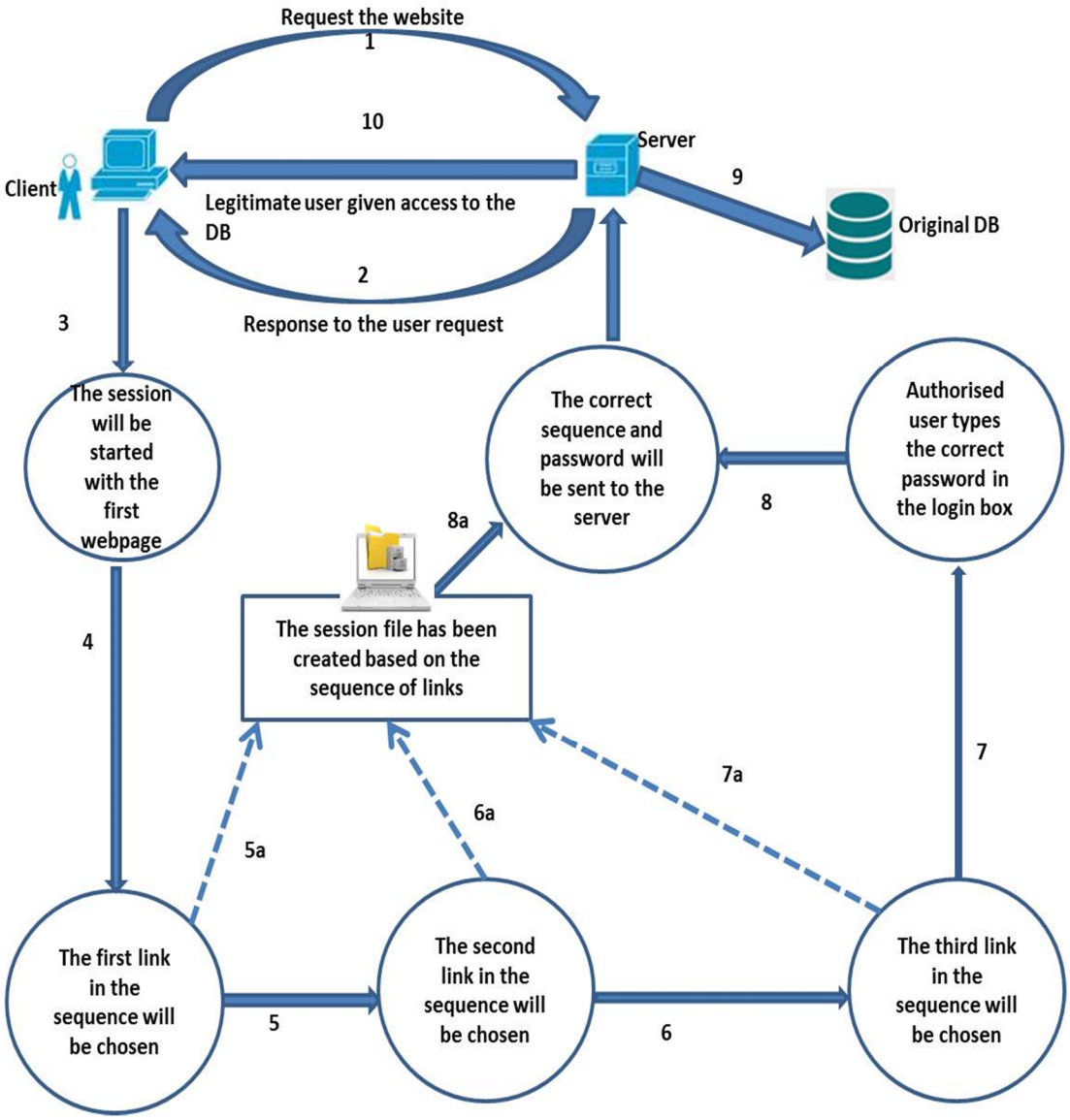

Figure 4-1 Authorised login by a legitimate user

4.1.1. Petri Nets Model

To describe how the new system works, a Petri net has been developed for each of it stages and Figure 4-2 illustrates the Petri nets model for the new authentication method. Usually, Petri nets are formally defined as a 5-tuple $N = (P, T, I, O, M_0)$, and these tuples can de described in this system as follows:

1- $P = \{p_1, p_2, p_3, \ldots, p_{12}\}$ is a finite set of places;

2- $T = \{t_1, t_2, t_3, \ldots, t_{16}\}$ is a finite set of transitions, $P \cup T \neq \emptyset$, and $P \cap T = \emptyset$;

3- I: $P \times T \rightarrow N$ is an *input function* that defines directed arcs from places to transitions, where N is a set of nonnegative integers;

I: {P1 ⟶ t1, P1 ⟶ t2, P2 ⟶ t3, P2 ⟶ t4, P3 ⟶ t5, P3 ⟶ t6,
P4 ⟶ t11, P4 ⟶ t17, P5 ⟶ t9, P5 ⟶ t10, P5 ⟶ t17, P6 ⟶ t11,
P7 ⟶ t7, P8 ⟶ t13, P9 ⟶ t12, P10 ⟶ t14, P10 ⟶ t15,
P11 ⟶ t16, P11 ⟶ t8};

4- O: $T \times P \rightarrow N$ is an *output function* that defines directed arcs from transitions to places

O: {t1 ⟶ P2, t1 ⟶ P4, t2 ⟶ P3, t3 ⟶ P4, t3 ⟶ P5, t4 ⟶ P6,
t5 ⟶ P9, t6 ⟶ P7, t7 ⟶ P3, t8 ⟶ P1, t9 ⟶ P6, t10 ⟶ P4,
t10 ⟶ P5, t11 ⟶ P8, t12 ⟶ P8, t13 ⟶ P10, t14 ⟶ P13, t15 ⟶ P11,
t16 ⟶ P12;

5- M_0: $P \rightarrow N$ is the *initial marking*.

$$M_0 = \{1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\}$$

It is clear that T1 and T2 are in a conflict situation, hence either T1 will be fired or T2, but not the both and the transactions that are in this situation have been marked with an orange border.

Table 4-2 illustrates the notifications that have been used with the Petri nets module.

Table 4-2 The notation used in the Petri nets

Notations	Meaning
WB	Webpage
U and P	Username and Password
SF	Session File
DB	Database

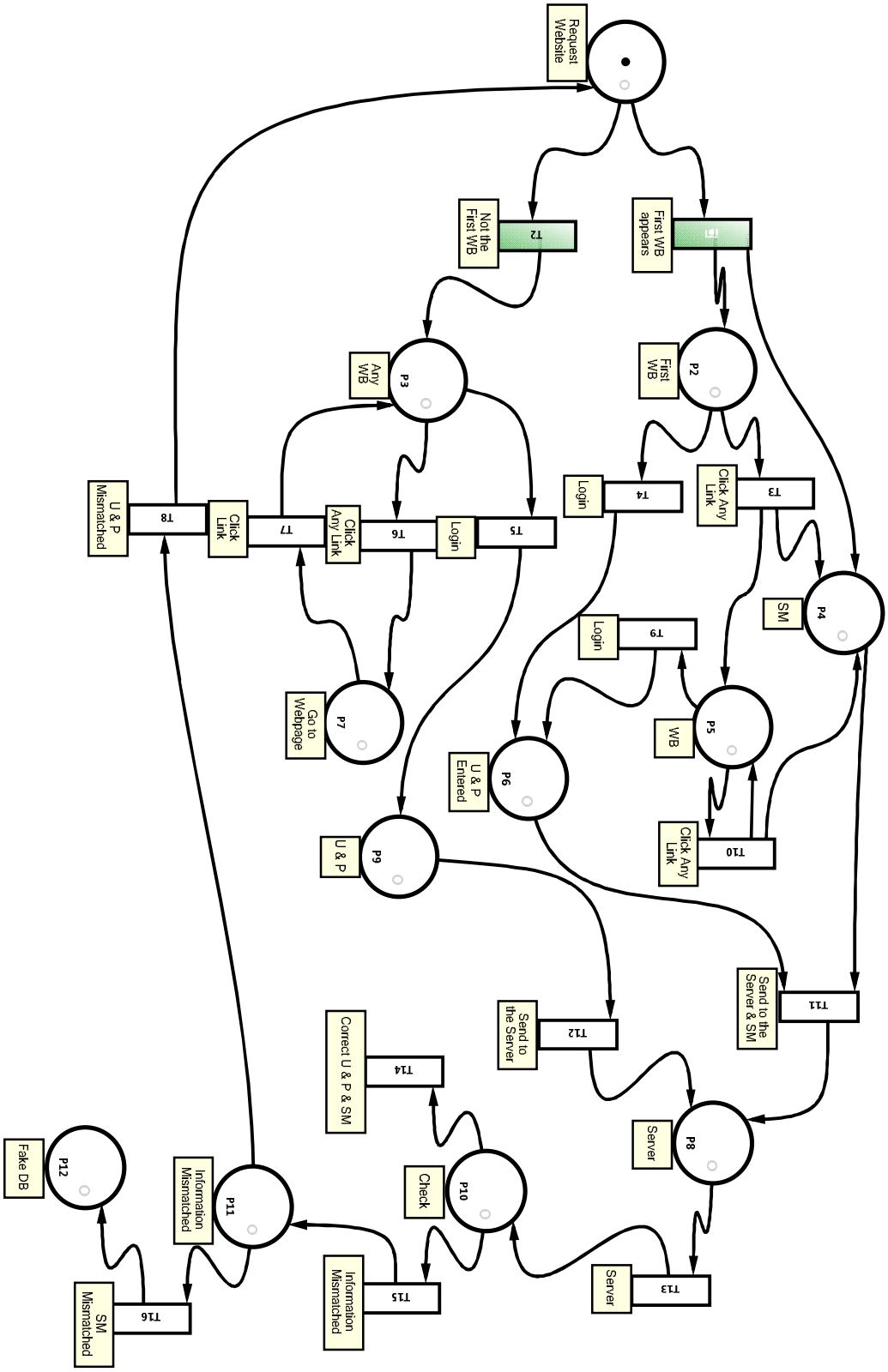

Figure 4-2 A Petri nets model of the new authentication method

4.2. Mathematical Model

4.2.1. Traditional Password Strength Measurement

The same PQI that has been used in the previous chapter is used in this section to measure the traditional method and so, the same equation (3-3) is applied:

$$L = m * \log_{10} C.$$

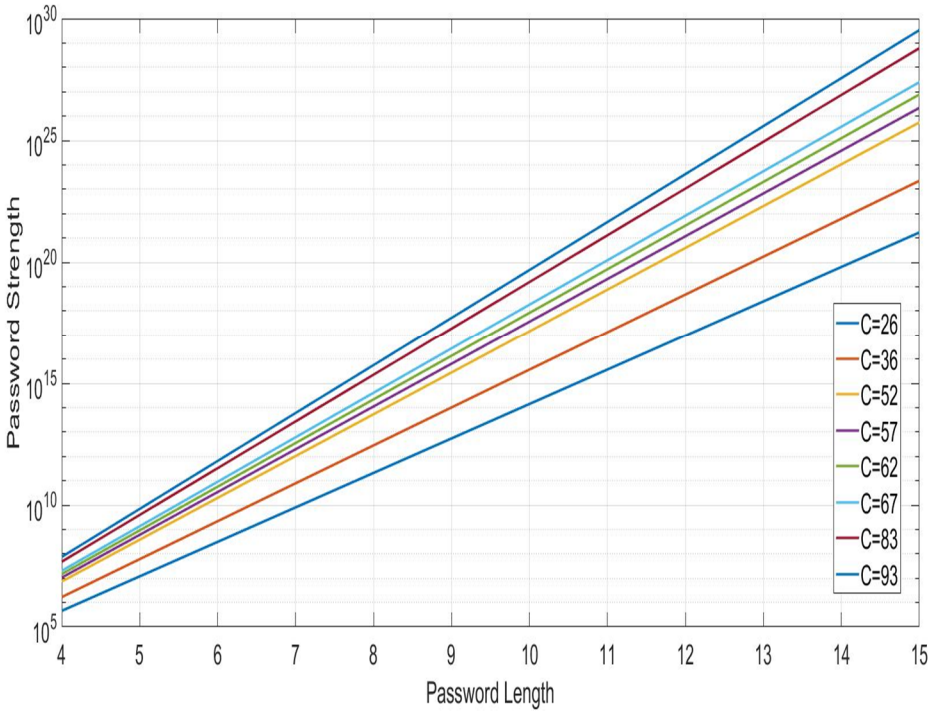

Figure 4-3 The traditional password results

Figure 4-3 illustrates the results of all possible L after equation 3-3 has been applied, with 10^L being calculated for each of this values, thereby covering all the available Cs. The x-axis represents the length of the password m, and the window period is taken from 4 to 15, whereas the y-axis refers to10^L. Clearly, from the Figure 4-3 results, the weakest password is when C = 26 and $m = 4$, whilst the strongest is when C = 93 and $m = 15$.

4.2.2. New Password Measurements

Having explained the traditional password measurement before, the mathematical equations (equation No. 3-3) need to be modified to account for the password after the links and session management have been added. For a new password Ω^*, the effective length of the new password 10^{L^*} will be calculated. Regarding the new method, the total number of links in the all websites will be represented by the symbol "Γ". As aforementioned, for each user there are three links that have been followed by the legitimate user to make the sequence correctly and hence, the new formula will be:

$$L^* = \log_{10}(C^m \Gamma^3) \quad (4\text{-}1)$$

Then, the possible candidates trying to crack the password will be 10^{L^*}. Figure 4-4 shows the results of all possible L^* after applying equation 4-1 and then, 10^{L^*} has been calculated, which means all the values of C have been taken. The x-axis pertains to the length of the password m from 4 to 15, whereas the y-axis is 10^{L^*}, and $\Gamma = 50$.

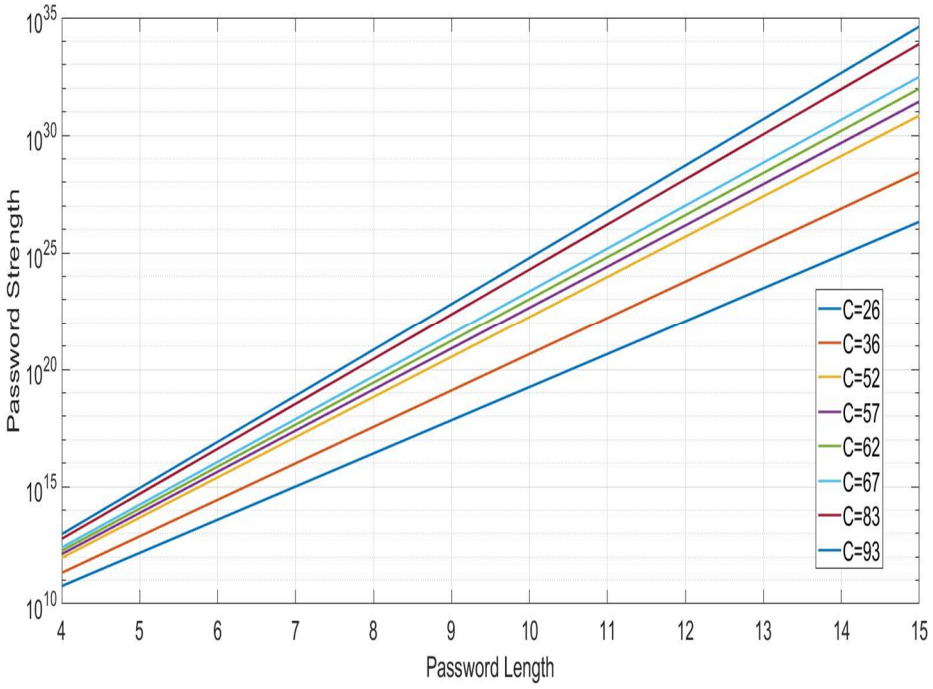

Figure 4-4 The results of the new authentication method

4.3. Results

There are eight results covering all the probabilities of C, but only three of them are considered in this paper due to several of their values being close together. Figure 4-5 illustrates the proposed authentication method when $40 \leq \Gamma \leq 80$, C = 26 and $4 \leq m \leq 15$ after calculating 10^{L^*}, which demonstrates the new authentication method when equation 4-1 has been applied. In this figure, there are some values that have been selected randomly and entered to demonstrate the changes in the values for the different Γ's and *m*s, because these are not easy to see with the naked eye 10^{L^*}.

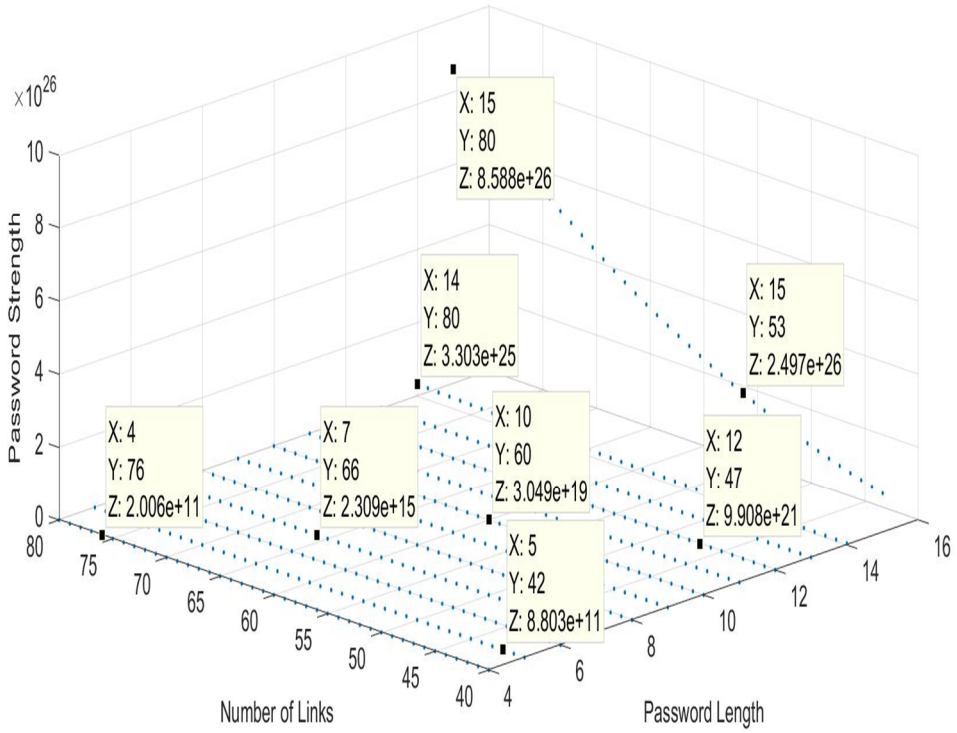

Figure 4-5 Proposed authentication method when $40 \leq \Gamma \leq 80$, C=26 and $4 \leq m \leq 15$ after calculating 10^{L^*}

While Figure 4-6 demonstrates the proposed authentication method when $40 \leq \Gamma \leq 80$, C = 26 and $4 \leq m \leq 15$ before calculating 10^{L^*} and Figure 4-7 illustrates it when $40 \leq \Gamma \leq 80$, C = 67 and $4 \leq m \leq 15$, Figure 4-8 shows the proposed authentication method when $40 \leq \Gamma \leq 80$, C = 67 and $4 \leq m \leq 15$, before calculating 10^{L^*}, whilst Figure 4-9 shows the method when $40 \leq \Gamma \leq 80$, C = 93 and $4 \leq m \leq 15$ and Figure 4-10 demonstrates it when $40 \leq \Gamma \leq 80$, C = 93 and $4 \leq m \leq 15$, before calculating 10^{L^*}.

There are eight possible values of C (explained in chapter 3). The following figures provide comparisons between the traditional password and the new authentication method for all these Cs and a fixed number of links for the new authentication method, which is 50 (Γ=50) for the all results shown in this thesis (Figure 4-11 to Figure 4-18). Obviously, the results of a new authentication method will change according to the number of links (Γ), i.e. when increasing (Γ > 50) or decreasing them (Γ < 50). In addition, the x-axis illustrates the length of password (m), and the possible password candidates to be tried to crack it are given on the y-axis in a log scale.

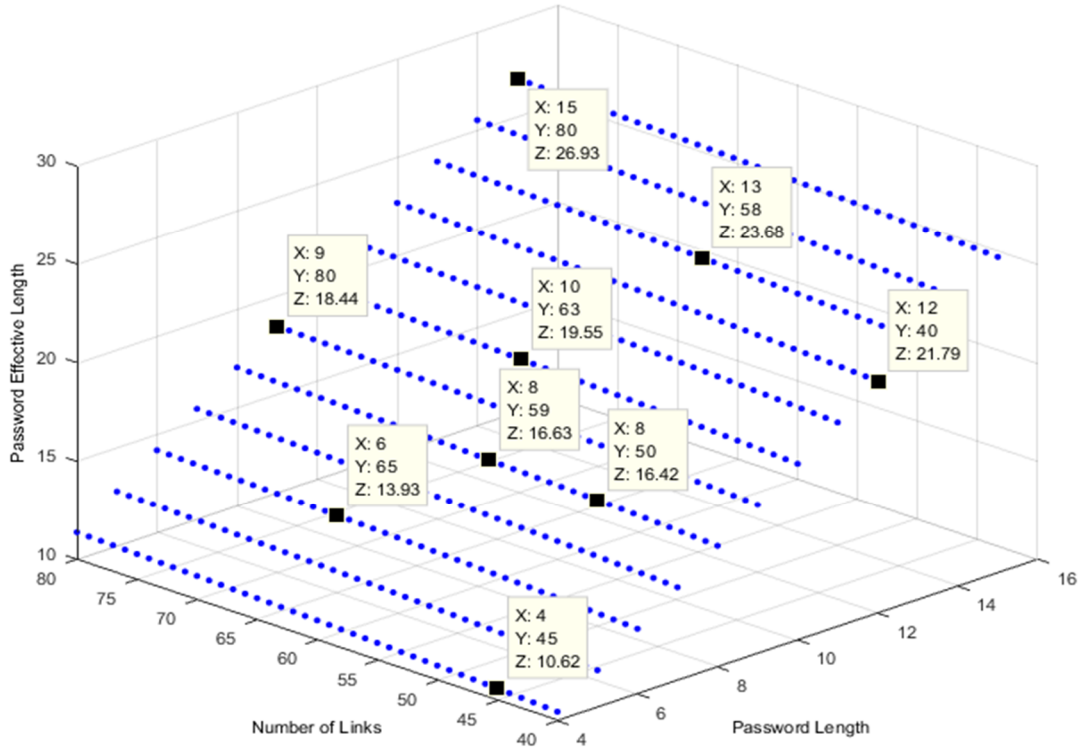

Figure 4-6 Proposed authentication method when 40 ≤ Γ ≤ 80, C = 26 and 4 ≤ m ≤ 15 of *L*

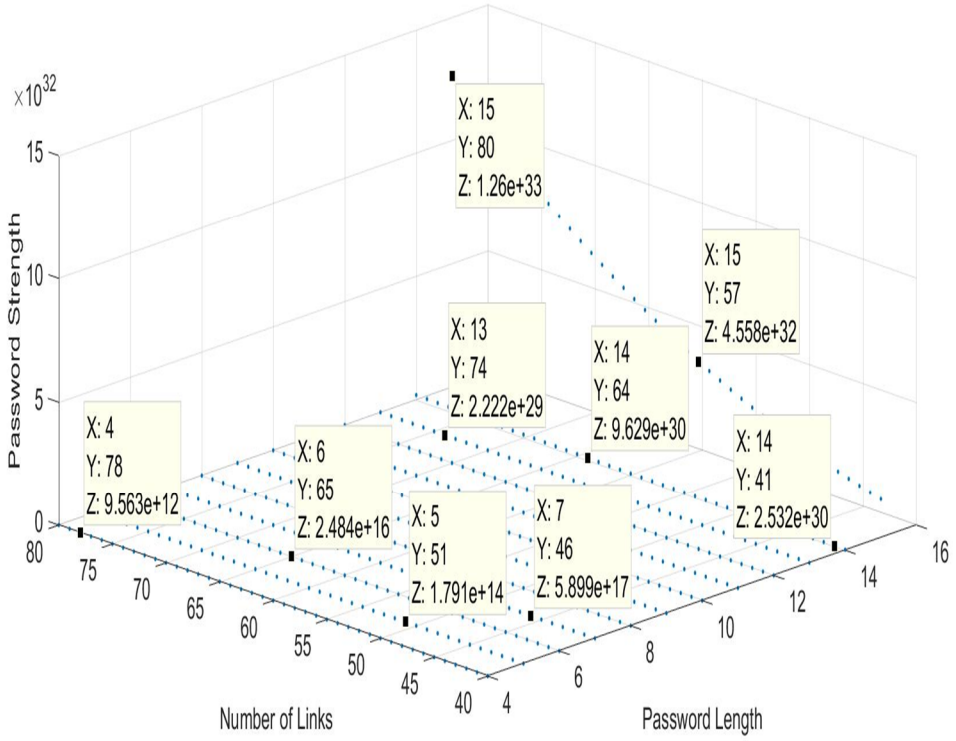

Figure 4-7 Proposed authentication method when 40 ≤ Γ ≤ 80, C = 67 and 4 ≤ m ≤ 15 after calculating 10^{L^*}

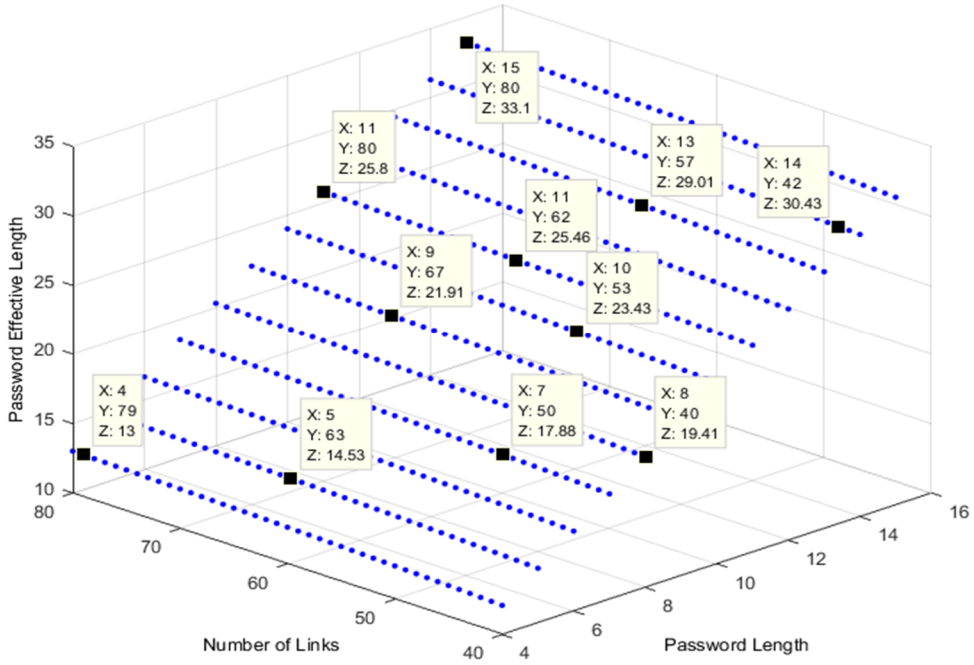

Figure 4-8 Proposed authentication method when 40 ≤ Γ ≤ 80, C = 67 and 4 ≤ m ≤ 15 of *L*

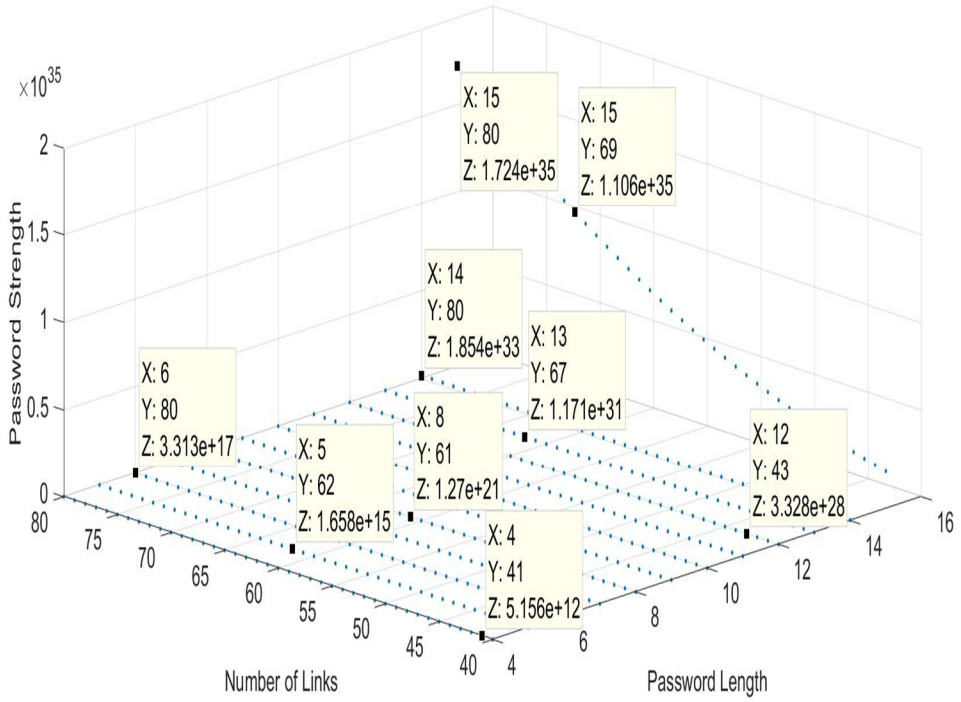

Figure 4-9 Proposed authentication method when $40 \leq \Gamma \leq 80$, C = 93 and $4 \leq m \leq 15$ after calculating 10^{L^*}

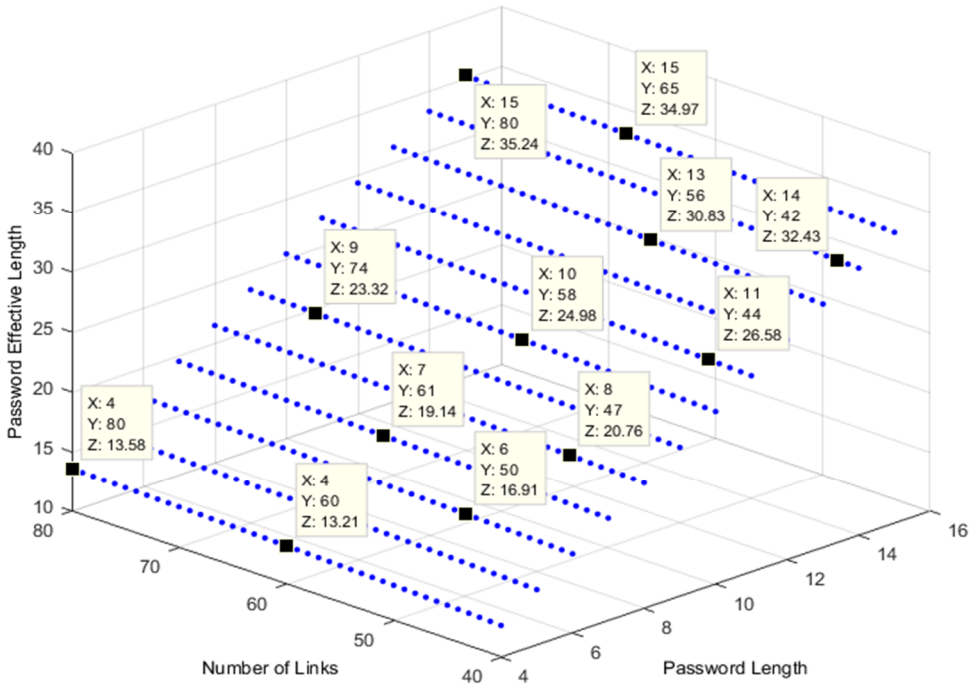

Figure 4-10 Proposed authentication method when $40 \leq \Gamma \leq 80$, C=93 and $4 \leq m \leq 15$ of L

Figure 4-11 illustrates the comparative results when C = 26, while Figure 4-12 provides them for when C = 36. Moreover, Figure 4-13 and Figure 4-14 illustrate the results when C = 52 and C = 57, respectively. In addition, Figure 4-15 shows the results when C = 62 and Figure 4-16 when C = 67. Finally, the last two values are C = 83 and C = 93, as shown in Figure 4-17 and Figure 4-18, respectively

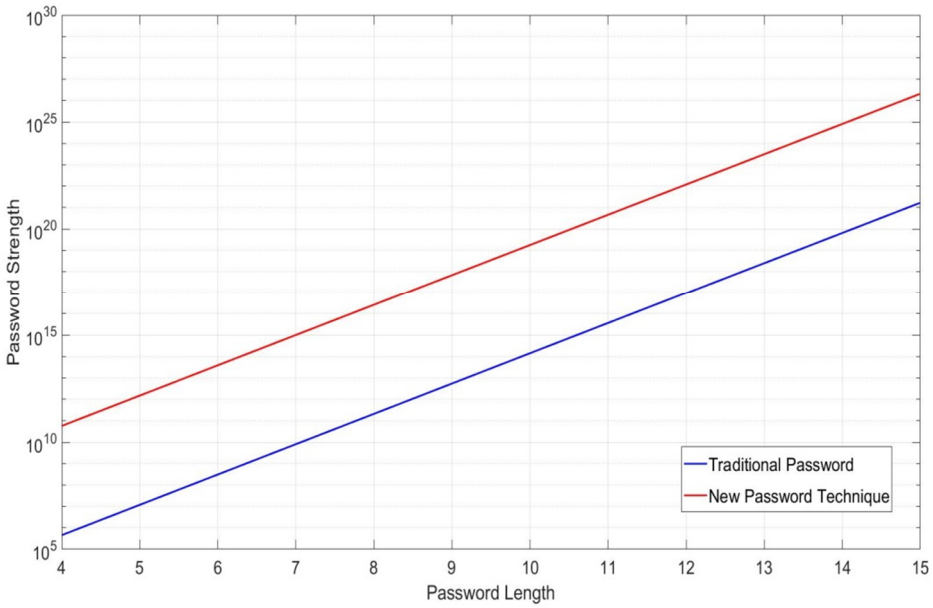

Figure 4-11 C = 26; password contains uppercase letters or lower case letters and Ɽ = 50

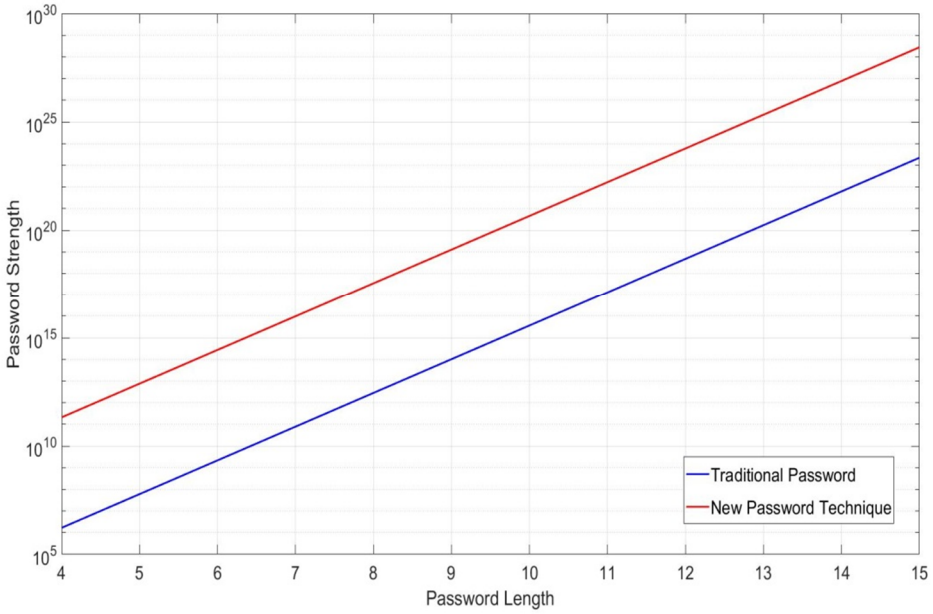

Figure 4-12 C = 36; password contains uppercase letter(s) or lower case letter(s) (26) + digit(s) (10) = 36 and Ɽ = 50

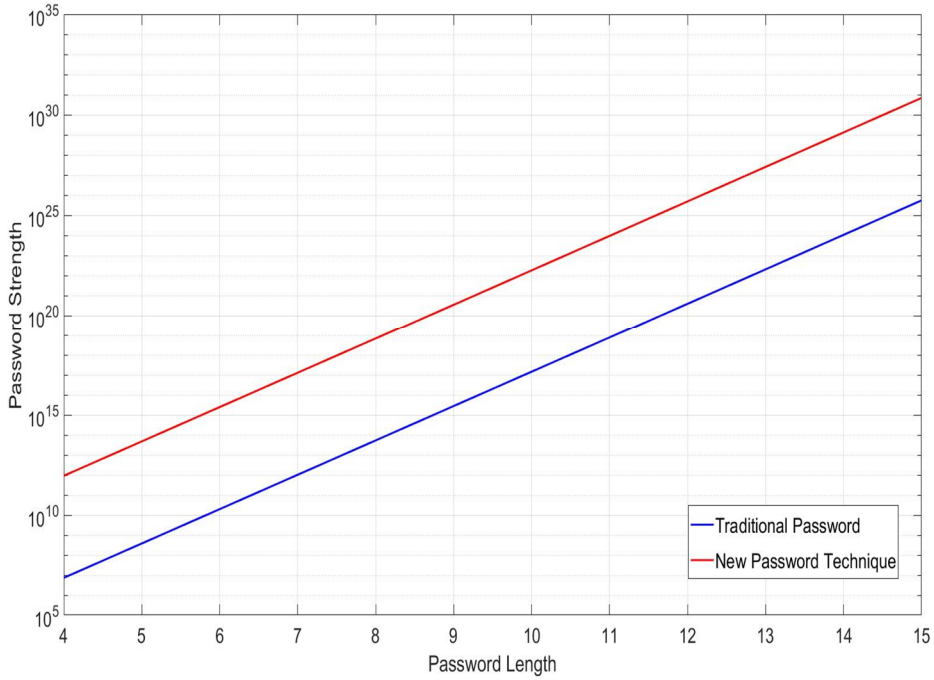

Figure 4-13 C = 52; password contains uppercase letter(s) (26) + lower case letter(s) (26) =52 and Ɽ=50

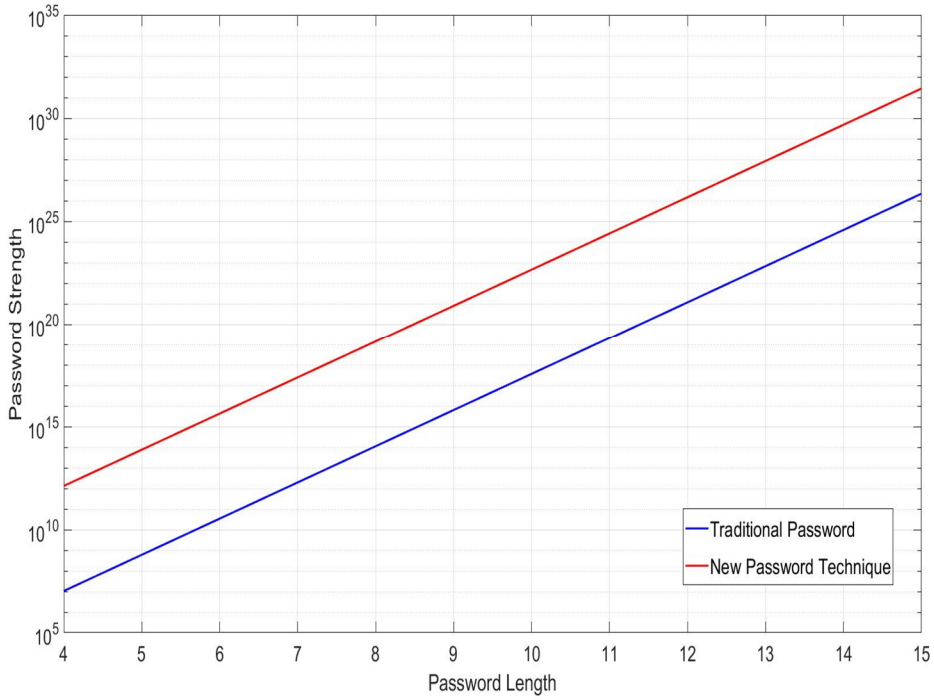

Figure 4-14 C=57; password contains uppercase letters or lower case letters (26) + special character(s) (31) = 57 and Ɽ= 50

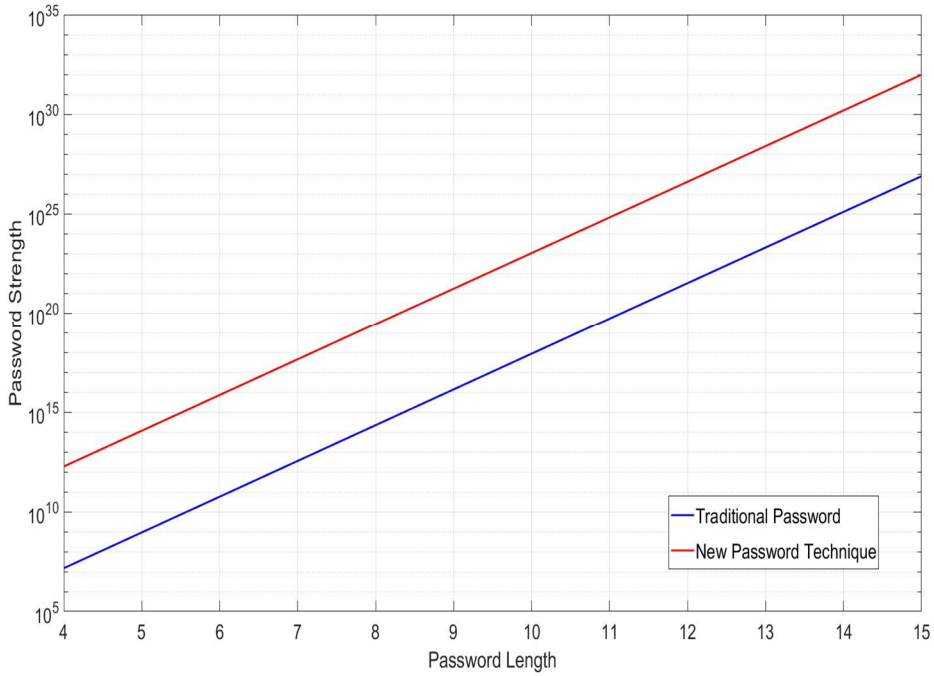

Figure 4-15 C = 62; password contains uppercase letter(s) (26) + digits (10) + lower case letter(s) (26) = 62 and Γ = 50

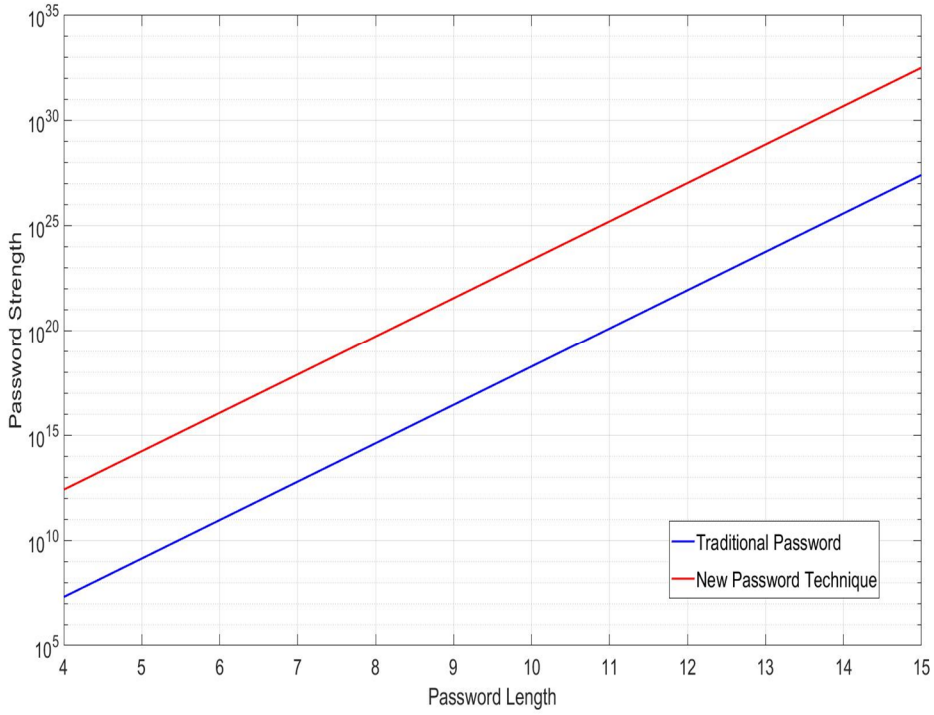

Figure 4-16 C = 67; password contains uppercase letters or lower case letters (26) + digit(s) (10) + special character(s) (31) = 67 and Γ = 50

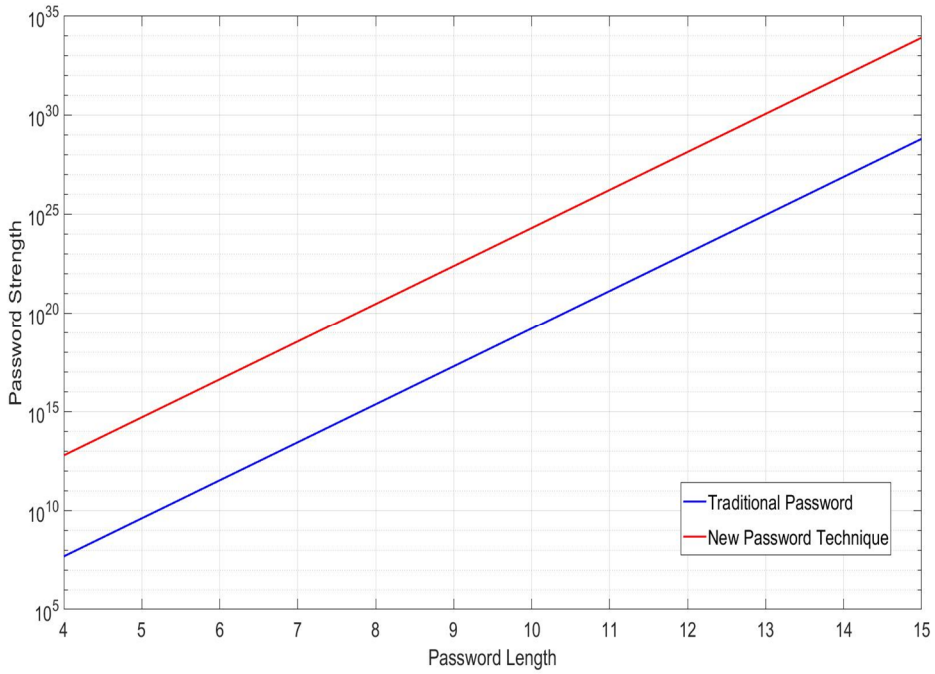

Figure 4-17 C = 83; password contains uppercase letter(s) (26) + lower case letter(s) (26) + special character(s) (31) = 83 and Γ = 50

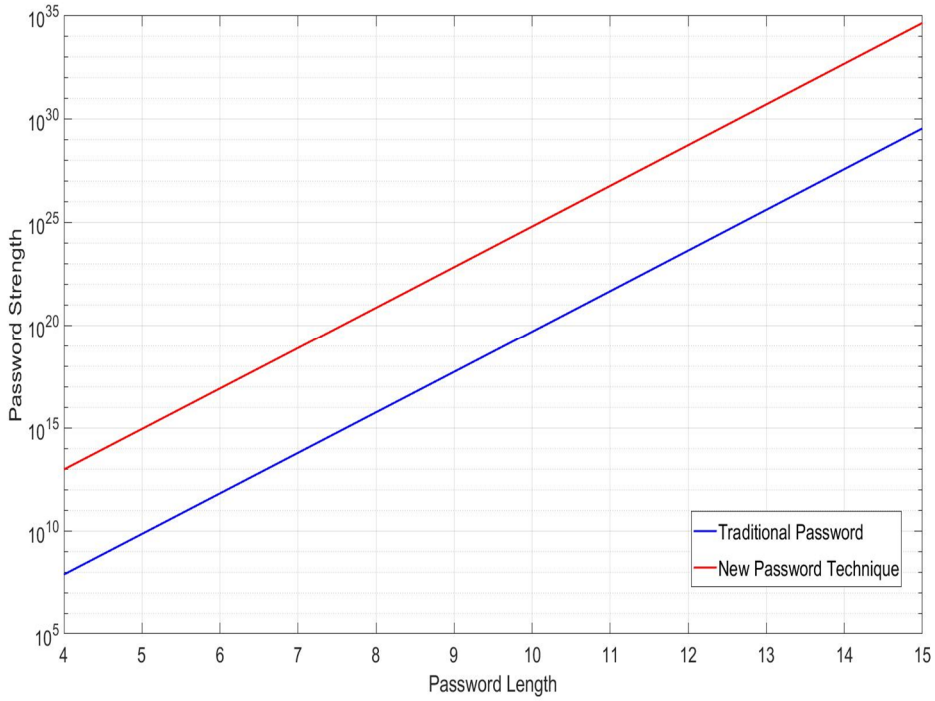

Figure 4-18 C = 93; password contains uppercase letter(s) (26) + lower case letter(s) (26) + special character(s) (31) + digit(s) (10) = 93 and Γ = 50

Clearly, for each value of C the weakest password occurs when the length of password $m = 4$, whilst the strongest emerges when $m = 15$, with the best being when C = 93 and $m = 15$.

4.3.1. The Difference Percentage Equation (DPE)

The Difference Percentage Equation (DPE) 3-10 has been used to calculate by how much the new password technique is better than the traditional one.

Figure 4-19 illustrates the results when DPE is applied to obtain a comparison between $L^*$and L. The results show the percentage by which the new method is better than the traditional one and they demonstrate that the former is approximately 200% better than the latter.

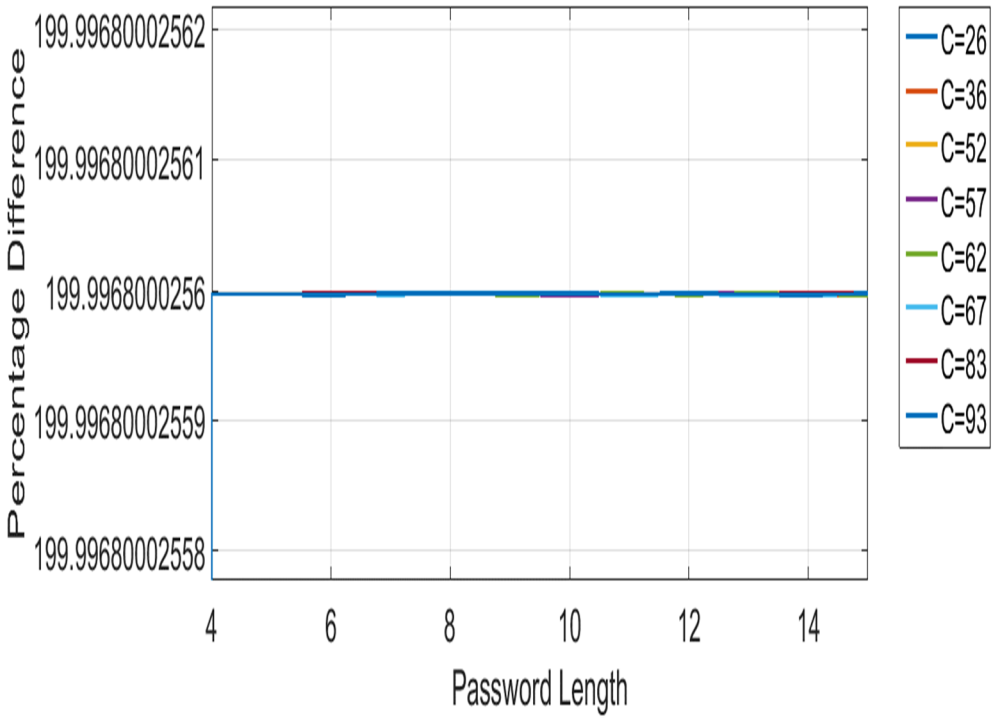

Figure 4-19 The results after DPE is applied

4.3.2. Results Analysis

From the results, when all possible probabilities of C are taken, it can clearly be seen that the strongest authentication password is when C = 93 and the number of links is 80. This is the maximum number of links Γ has to have to obtain these outcomes. When C = 93, this involves a combination of, uppercase letters, lower case letters, special characters and digits, with the new authentication method. Moreover, when making a comparison between the traditional authentication method and the new one, the number of links in the website is fixed at 50. As aforementioned, the results have shown that the new method is better than the traditional one by 200%. Finally, the weakest new authentication value emerged when C = 26. Additionally, the strength of the password in this technique was based on the number of links in the system Γ. However, when Γ is high, the possible combinations for the password increased too.

4.4. Analysis of Password Attacks

As a result of this approach, all password attacks (brute-force, dictionary, shoulder surfing and guessing attacks) will not be able to succeed due to the session management not being a concern of any of them. Furthermore, brute-force attacks and dictionary attacks are not concerned with links as a part of any authentication method, seeing as they are entirely focused on words and numbers. Shoulder surfing, will need to have a very high accuracy in terms of monitoring how the authorised user is going through the correct sequence links. However, if the attacker is able peep at the sequence of links and can recognised that they are a part of an authentication process, then will be too difficult to stop him/her. Nevertheless, usually attackers monitoring passwords try to enter them in the login box, which is used here to lure the attacker to login and he/she can be easily detected.

Moreover, a guessing attack deploys words rather than links and how to create the session management, so it is not be able to succeed with the proposed approach. Usually, sessions are starting after the login process, but in this system the session will be triggered when the user goes to the main webpage (first webpage). Furthermore, the signature based type IDS has been linked with the session

management to check and monitor the full tracking (order of the user link sequences) during the login process under the proposed arrangement.

4.5. Summary

In this chapter, a new password technique based on honeypots using the session management has been developed for allowing the legitimate user to have access to a database. Specifically, the proposed approach mitigates attacks and threats and will be used as a decoy against the imposter user. The session will be created when the first (main) webpage is requested by user, with each link having a unique order number on each webpage. The session that has been created in advance will enable the system to have a full browsing track for each user and then, the order number for each link that has been clicked by the user will be embedded in the session. The correct sequence of the links must be performed by the legitimate user compatible with the session management that has been saved in the server through the sign up process, for the user have access to the dataset. Hence, the username u_i, password p_i and session management s_i have to be correct and compatible with the information of user i before access to the database is granted. The aim is to create honeypots to prevent password attacks by increasing the number of possible combinations for password authentication, thereby improving security when compared to traditional measures. A mathematical model has been built and the Password Quality Indicator (PQI) used to compare the outcomes generated by the model with those from the traditional password method. Further, the mathematical model was applied in MATLAB to obtain the results shown in the graphs provided. Moreover, whilst the number of links can vary, they have been fixed in the procedure to make the results more comprehensible. The difference percentage equation has been employed to determine whether the new authentication of password technique is better than the traditional method and the results show that the former is better than the latter by 200%. In addition, Petri nets have been used to explain how the whole new system works.

Chapter Five: A Multi-Factor Authentication Approach Based on Honeypot Web Session Management and Time-Period Generation

5.1. Methodology

Multi-factor authentication is widely using in authentication systems to protect sensitive information. As aforementioned, multi-factor authentication consists of two or more techniques combined together. In this chapter, two methods mentioned before in chapters three and four, are integrated to create a new multi-factor authentication method. The new method starts with creating the correct session management s_i by the authorised user to obtain the correct first step in the login processing. The next step is entering the username and password to perform the login process. In this stage the authorised user must enter a special stroke, which is either an uppercase letter or special character. Moreover, this time period Tp_i must be generated by using the shift-key.

In the first step to make an authorised user have a correct login and then, access to a sensitive dataset, this user u_i, must generate the correct session management s_i. To this end, the authorised user should request the first webpage of the website, at this moment the web session management will be begin. The next step for the web session management will be to choose the correct links, there being a specific sequence of three. This sequence of links has been determined before, with each user having a different one. If the first webpage is not requested, then the web session management will not be started and the authentication process will have failed, with the signature IDS system recognising that this is might be an attack. Additionally, the login box will appear on all webpages to lure the adversary user to enter the username and password on any one of these (usually the first webpage) and thus, incorrect session management. In contrast, an authorised user will enter the correct username u_i and the correct timed password Tp_i. Finally, the whole procedure in the form of interaction honeypot lures the attacker into committing a mistake, which makes it easy for IDS to detect him/her. The same procedure as described in chapter four will be applied.

For each user, there will be a password including uppercase letter(s) and/or special character(s) generated by the shift-key and as described in chapter three, the time taken to release the shift-key after typing one of the aforementioned keys becomes an element of the password. That is, the same procedure as that presented in chapter three will be applied in this chapter.

To sum up, the multi-factor authentication that has been used in this chapter will make the authentication process much stronger than the traditional password. However, the time consumed to have access to a sensitive dataset will be increased, which could frustrate the users. Furthermore, the procedure of generating Tp_i is difficult for the normal users due to being based on changing user behaviour. To overcome this, training will be required or the use of a timing device to calculate the window period. Table 5-1 illustrates a comparison between two authentication files on the server side, with the first being the traditional one (Table 5-1-a), whilst the second is the proposed file (Table 5-1-b).

Table 5-1 Traditional authentication method and new method saving in the server

Username	Password
u_1	p_1
u_2	p_2
. .	
u_n	p_n

a. Traditional information on the server side

u_i	Tp_i	s_i
u_1	Tp_1	s_1
u_2	Tp_2	s_2
. .	. .	. .
u_n	p_n	s_n

b. New information on the server side

Figure 5-1 shows the steps that must be followed by the authorised user to gain access to the system.

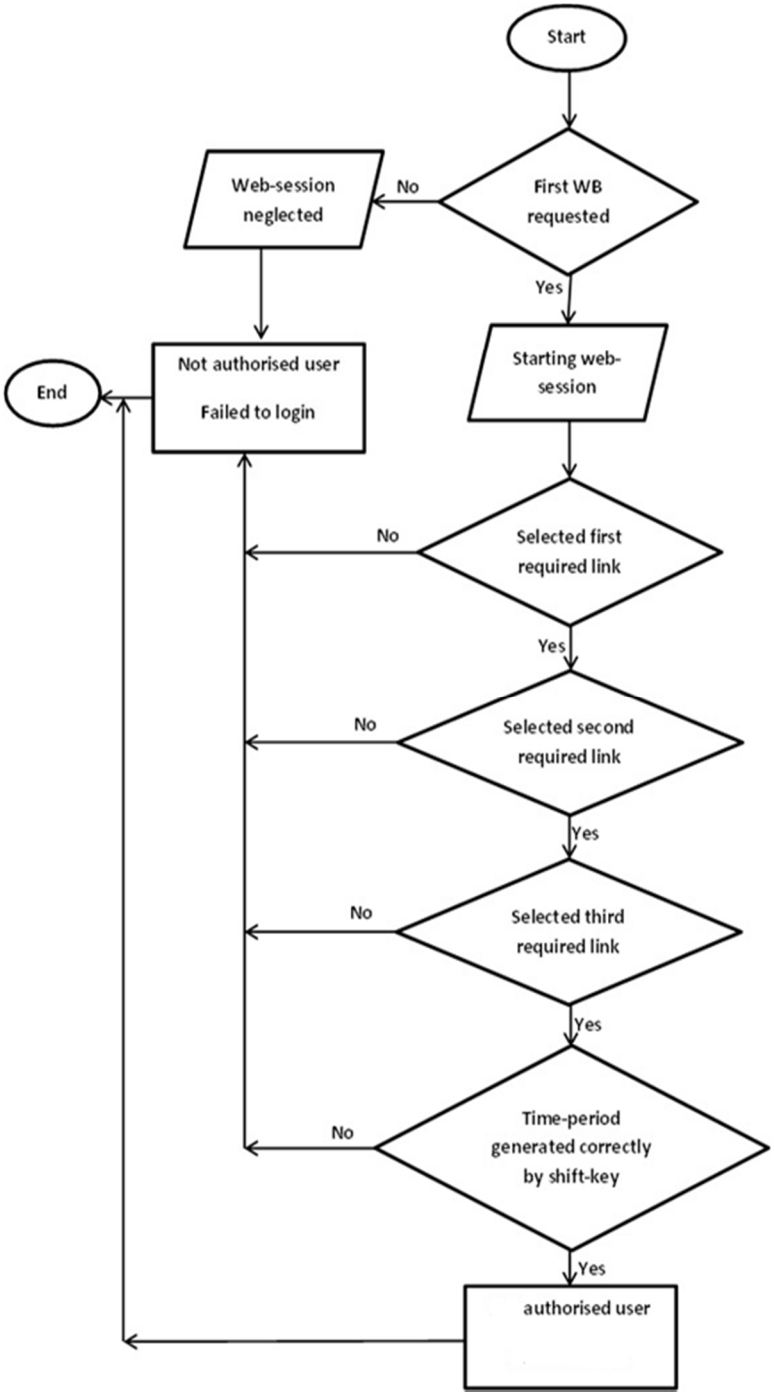

Figure 5-1 How an authorised user can gain access to the system

5.2. Mathematical Model

5.2.1. Traditional Password Measurement

The same PQI that was used in the two previous chapters is applied in this section to measure the traditional password and then the comparison with the results for the multi-factor method made after the new mathematical model has been be derived. The formula for traditional password measurement is the same equation 3-3 as before.

5.2.2. New Password Measurement

5.2.2.1. New Password Technique with Two Fixed Positions Timed Password based on Web-Session Management

In this subsection, the authentication procedure granting login to a sensitive dataset is described. First, the correct session management s_i is established by requesting the first webpage and going through the correct sequence. To complete the login process, the authorised user must enter username u_i and password p_i. The password should be come with the required time period generated by shift-key, in this case, two fix positions special strokes are used to grant access to the dataset. For example, the first position and fifth position in the password can be special keystrokes relating to the timed element. The mathematical model is derived from the PQI traditional model measurements.

$$G = C^m, G^* = \Gamma^3 C^{m-2}(xl)^2$$

$$\Delta L = L^* - L$$

$$\Delta L = \log_{10} G^* - \log_{10} G$$

$$\Delta L = \log_{10} \Gamma^3 C^{m-2}\,(xl)^2 - \log_{10} C^m$$

$$\Delta L = \log_{10}\left(\frac{\Gamma^3 C^{m-2}(xl)^2}{C^m}\right)$$

$$\Delta L = \log_{10}\left(\frac{\Gamma^3 C^{m-2}(xl)^2}{C^m}\right)$$

$$\Delta L = \log_{10}\left(\frac{\Gamma^3 (xl)^2}{C^2}\right) \quad (5\text{-}1)$$

Since:

Γ: The number of links in the all webpages, in this system will be fixed on 50.

x: The number of special keystrokes characters (42), 26 uppercase letters + 16 special characters can be typed by shift-key.

l: The number of period time, in this method is 4, and the unit is milliseconds (2001-4000), (4001-6000), (6001-8000), (8001-1000).

C: Password Complexity Index (PCI).

5.2.2.2. New Password Technique with a Two Variant Positions Timed Password based on Web-Session Management

The same login procedure as that explained in the previous section is applied with a slight change, whereby the positions of the keystrokes characters the timed keystrokes can be at any position in the password, rather than fixed. That is, the authorised user can type the special keystrokes at any two positions in the password. The traditional measurement of the PQI has been used to derive the new mathematical model to obtain the results.

$$G = C^m, G^* = \Gamma^3 \binom{m}{2} C^{m-2}(xl)^2$$

$$\Delta L = L^* - L$$

$$\Delta L = \log_{10} G^* - \log_{10} G$$

$$\Delta L = \log_{10}\Gamma \binom{m}{2} C^{m-2}(xl)^2 - \log_{10} C^m$$

$$\Delta L = \log_{10}\left(\frac{\Gamma^3 \binom{m}{2} C^{m-2}(xl)^2}{C^m}\right)$$

$$\Delta L = \log_{10}\left(\frac{\Gamma^3 m(m-1)(m-2)!}{2!\,(m-2)!}\left(\frac{xl}{C}\right)^2\right)$$

$$\Delta L = \log_{10}\left(\frac{\Gamma^3 m(m-1)}{2}\left(\frac{xl}{C}\right)^2\right) \quad (5\text{-}2)$$

5.2.2.3. New Password Technique with a Three Variant Positions Timed Password based on Web-Session Management

The same above login procedure with three variant positions rather than two is deployed next, so the new formula is:

$$\Delta L = \log_{10}\left(\frac{\Gamma^3 m\,(m-2)(m-1)}{3!}\left(\frac{xl}{C}\right)^3\right) \quad (5\text{-}3)$$

5.3. Results

There are eight sets of results that cover all probabilities of C, with three of them being shown in this chapter, i.e. when C = 26, C = 62, C = 93. Moreover, for each C there are three figures provided, the fixed position, two variant positions and three variant positions scenarios, as described above. Further, the results are presented for $40 \leq \Gamma \leq 80$ to explain how the number of links can impact on the new authentication technique. That is, when the number of links is high, the authentication technique will give a strongest results due to the number of possibilities increasing Figure 5-2 illustrates the results of C = 26 for the new authentication method with two variant positions applied based on equation 5-2, while $40 \leq \Gamma \leq 80$, and $4 \leq m \leq 15$, after calculating 10^{L^*}. Figure 5-3 shows when $40 \leq \Gamma \leq 80$ and $4 \leq m \leq 15$, after calculating 10^{L^*} for three variant positions, and equation 5-3 has been applied. In some figures, there are several values that have been selected randomly and entered to demonstrate the changes in the values for the different Γ's and *m*'s, because these are not easy to see with the naked eye.

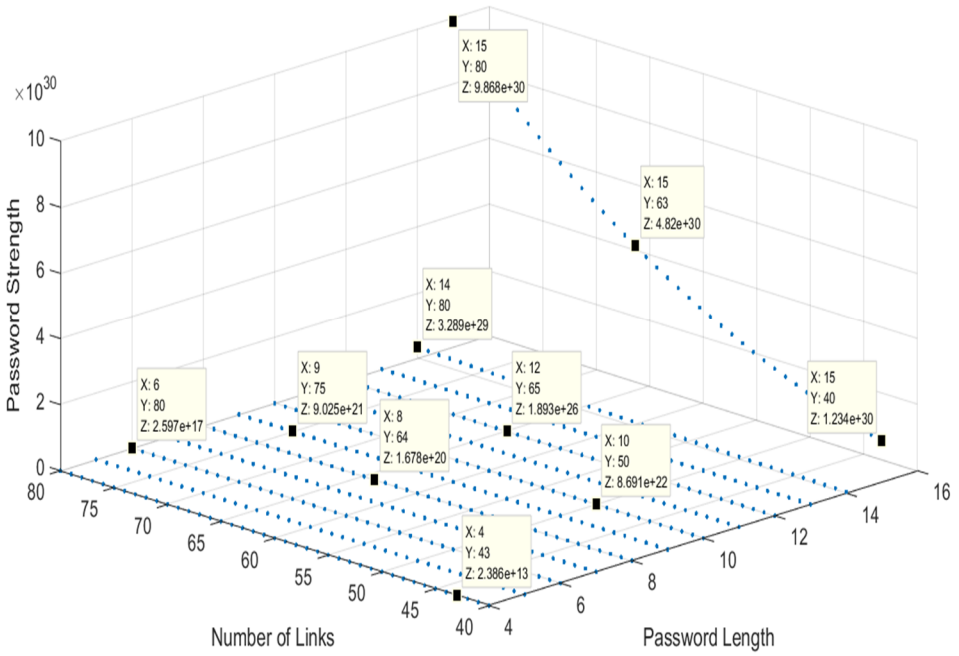

Figure 5-2 The password strength of two variant positions when C = 26, 40 ≤ Γ ≤ 80 and 4 ≤ m ≤ 15, after calculating 10^{L^*} and equation 5-2 has been applied

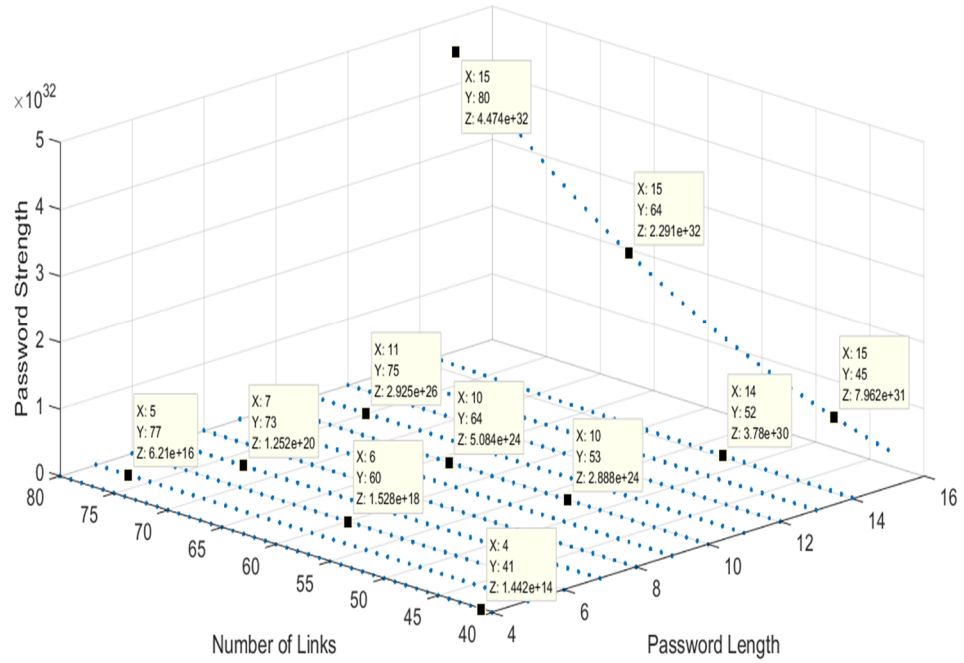

Figure 5-3 The password strength of three variant positions when C = 26, 40 ≤ Γ ≤ 80, 4 ≤ m ≤ 15, after calculating 10^{L^*} and equation 5-3 has been applied

Figure 5-4 illustrates the results when C = 62 with two variant positions for 40 ≤ Γ ≤ 80, and 4 ≤ m ≤ 15, after calculating 10^{L^*}. Obviously, the strength of the password will be better than this when C = 62, with three variant positions and 40 ≤ Γ ≤ 80, and 4 ≤ m ≤ 15, after calculating 10^{L^*}, the results for which are shown in figure 5-5. Figure 5-6 and Figure 5-7 show the two variant positions and three variant positions scenarios, respectively. Clearly, the strength of the password is increasing when C is changed to the higher value and the number of links is increased.

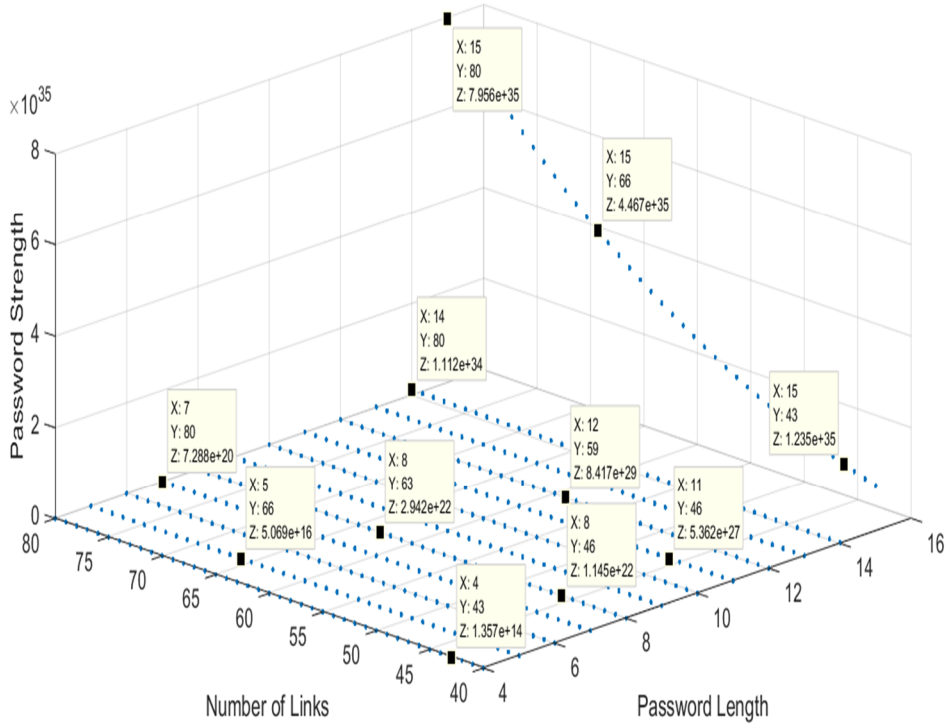

Figure 5-4 The password strength of two variant positions, when C = 62, 40 ≤ Γ ≤ 80, 4 ≤ m ≤ 15, after calculating 10^{L^*} and equation 5-2 has been applied

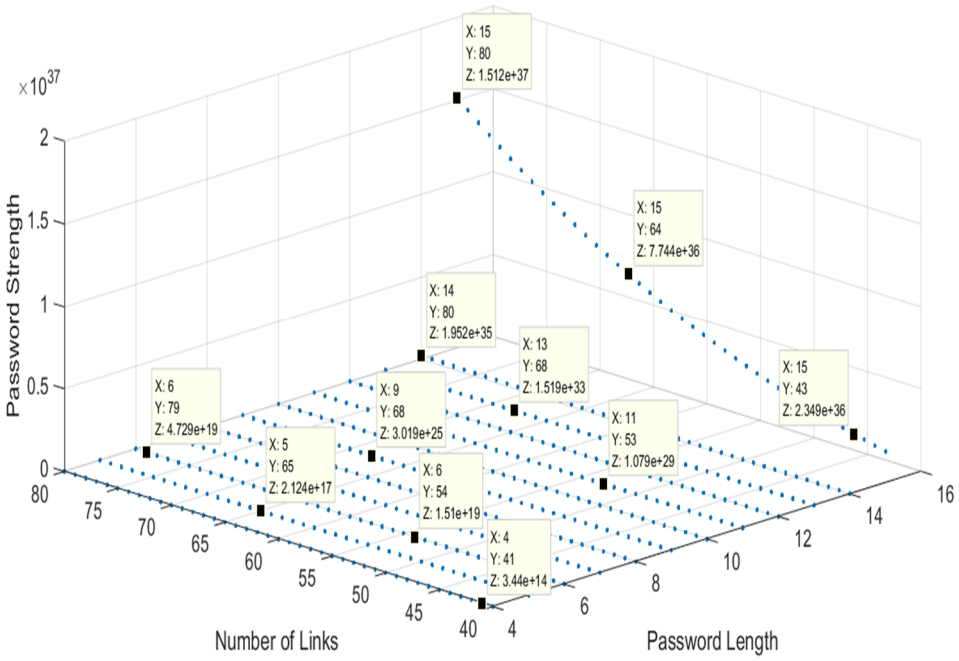

Figure 5-5 The password strength of three variant positions, when C = 62, 40 ≤ Γ ≤80, 4 ≤ m≤ 15, after calculating 10^{L^*} and equation 5-3 has been applied

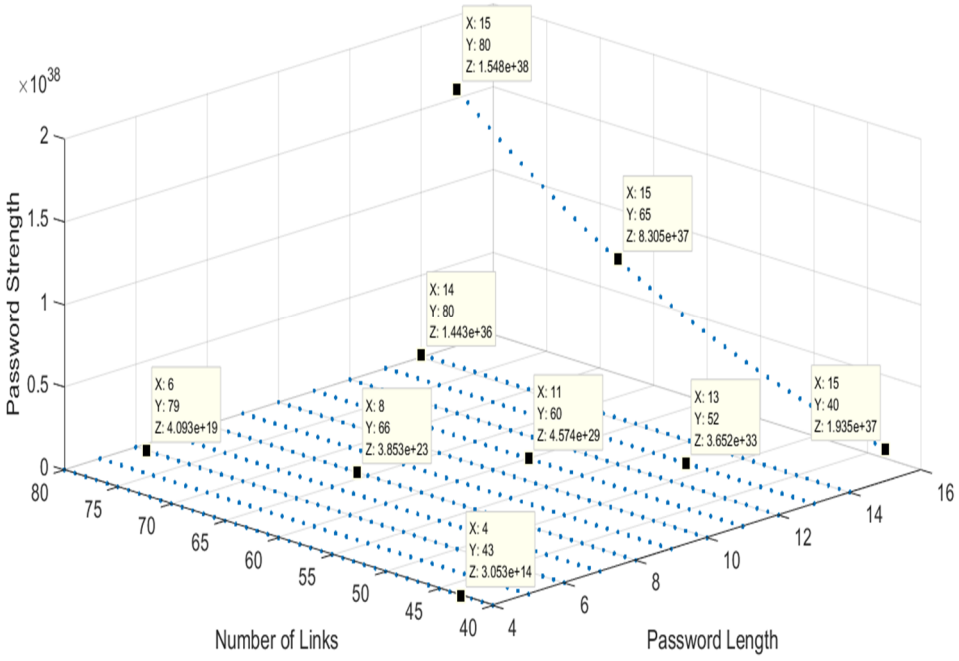

Figure 5-6 The password strength of two variant positions, when C = 93, 40 ≤ Γ ≤ 80, 4≤m≤15, after calculating 10^{L^*} and equation 5-2 has been applied

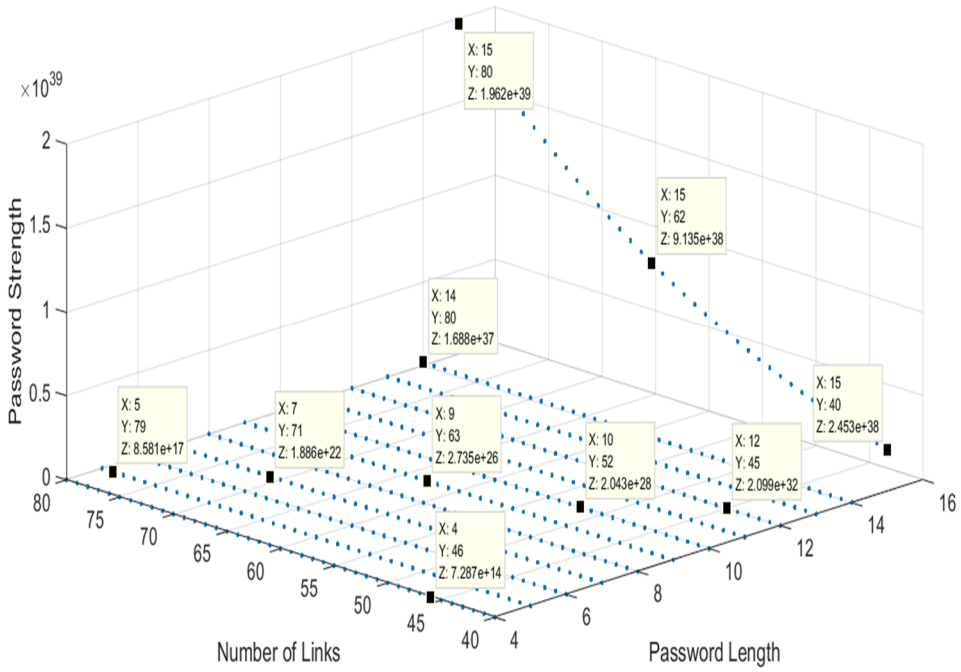

Figure 5-7 The password strength of three variant positions, when C = 93, 40 ≤ Γ ≤ 80, 4 ≤ m ≤ 15, after calculating 10^{L^*} and equation 5-3 has been applied

To provide a comparison of the results between the multi-factor authentication method and those of the traditional password, three out of eight have been selected to illustrate the strength of password. The C values that have been selected are C = 26, C = 62 and C = 93, with the number of links being fixed at Γ=50 for applying to MATLAB to draw the comparisons between the traditional password, fixed positions password, two variant positions password, and three variant positions password. Figure 5-8 illustrates the results when C = 26, whilst Figure 5-9 shows for when C = 62, and Figure 5-10 gives them for when C = 93.

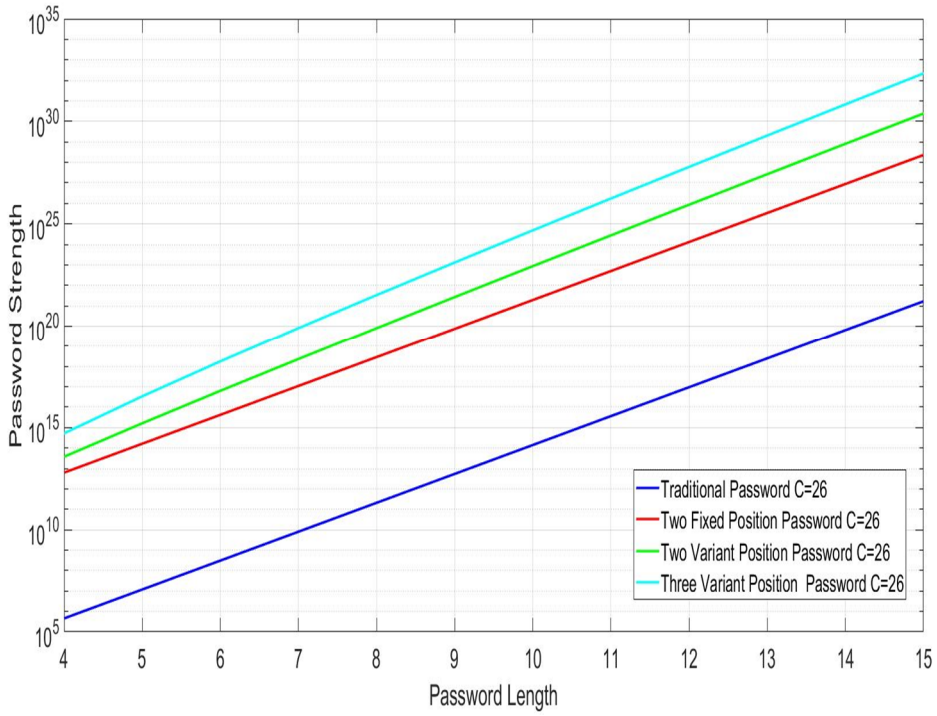

Figure 5-8 The results when the value of C = 26 and the number of links is fixed at Γ = 50

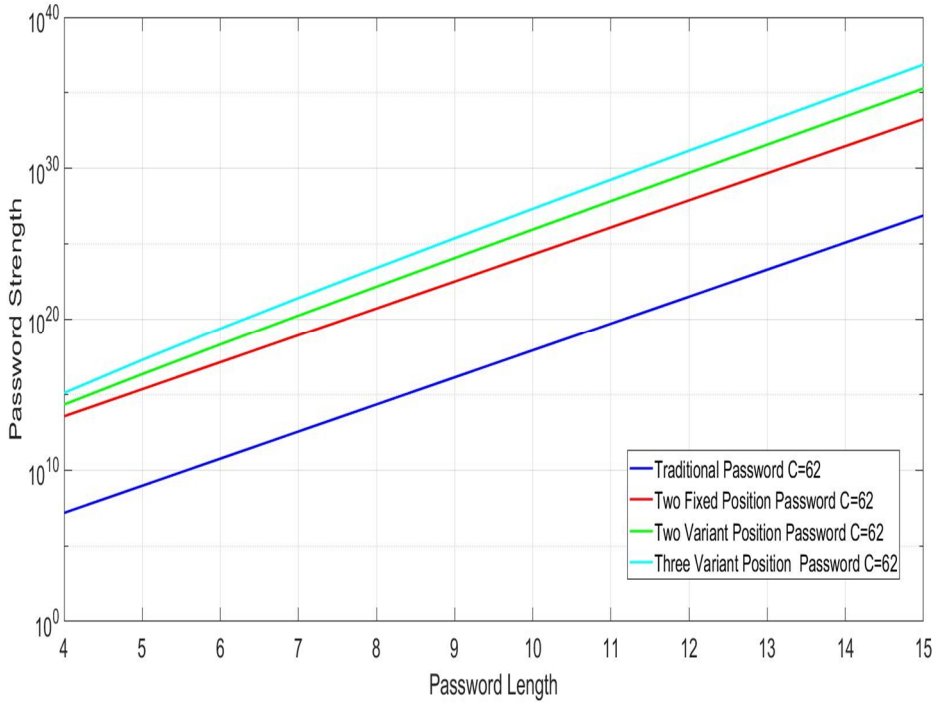

Figure 5-9 The results when the value of C = 62 and the number of links is fixed as Γ = 50

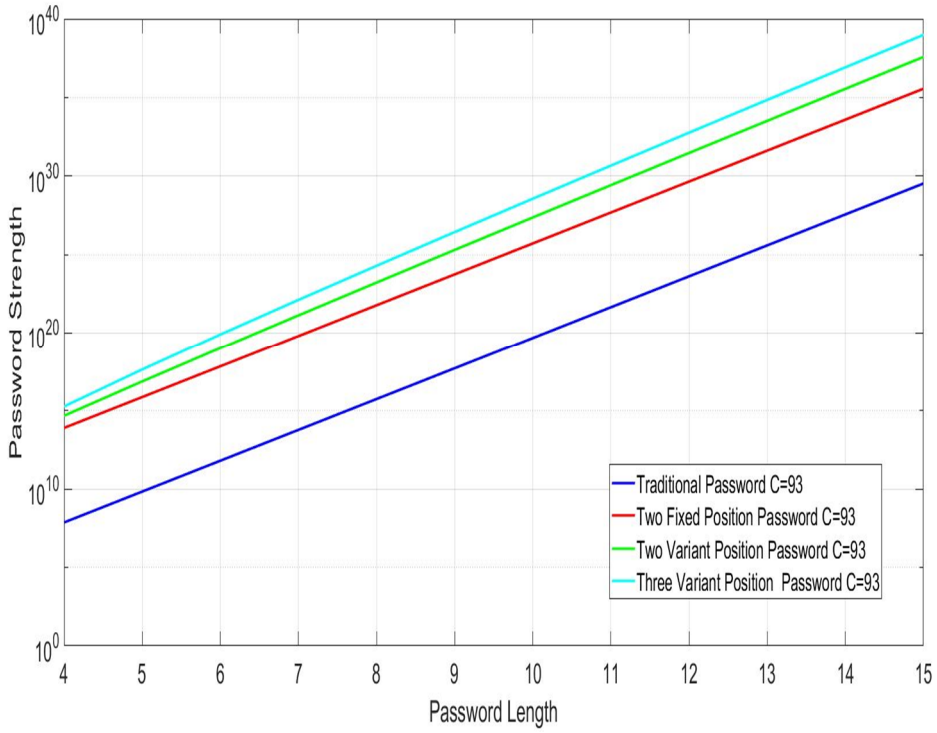

Figure 5-10 The results when the value of C =93 and the number of links is fixed as Γ = 50

5.3.1. Results Analysis

In the new multi-factor authentication, there are four elements that can affect the results. The first is the value of C, whereby there are eight probabilities for it and when C is high the password will be stronger. The second element is the positioning of the special keystrokes in the password in terms of whether they are fixed or variant. The results have shown when the special stroke positions are variant the strength of password will be stronger than when fixed position. Furthermore, the third element that can influence the strength of the password is the number of links Γ, such that when this is high the strength of password will be high too due to the increased number of possible combinations when trying to discover the correct links sequence. Finally, the fourth element is the length of the password m, whereby the strength of the password will be increased when the m value is high. The general results have shown that the new multi-factor authentication method is better than the traditional, timed password and session management authentication method when these are used separately rather than together.

5.4. Summary

In this chapter, two authentications methods that have been described before have been combined to create a new authentication method, which has multiple factors. The first factor is high interaction honeypot web-session management, where the user has to request the first webpage from the website to create the session management. The authorised user must choose the correct three step sequence to be added to the session and the login box will be shown on all the webpages to be used as a honeypot given the required links sequence procedure.

The next step, after the correct session management has been provided, is to enter the username and password, because now, the bone fide user is on the correct web page. The authorised user will enter the username followed by the timed password. If all three elements are correct, then the user will have access to the sensitive dataset, otherwise it will be denied. The multi-factor authentication method is not appropriate for everyone or to protect normal datasets and training in how to generate the correct time by using the shift-key will be needed. Finally, the results have shown that the multi-factor authentication method is better than traditional authentication, web-session management, and the time password used singularly. The new technique's results demonstrated that the multi-factor technique is 300% more superior than the traditional password technique. The timed password and honeypots technique, on the other hand, eclipsed the traditional technique by approximately 200%. These outcomes will increase the possible combinations for a given password. Furthermore, the multi-factor technique can only be used in the presence of two conditions: a) in a website containing links, b) on special types of users with abilities to adapt the new technique by changing their behaviours.

Chapter Six: Honeywords Generation Method for Passwords Based on User Behaviours to Achieve Flatness

6.1. Honeychecker

There is a very helpful service called honeychecker to ascertain whether the password entered by an authorised user through the login process is correct or a decoy (honeyword). Honeywords are stored randomly in the honeychecker system for any user along with the password. The term "sweetwords" refers to all the honeywords and password for each user [162]. The honeychecker is created to help the secure system developer to use honeywords, if attackers hack into the computer system and steal the file of password hashes, whether salted or deployed using other hashing parameters. The honeychecker is a separate computer system containing the secret information. It will communicate with the computer system through the login process, or when the user tries to change the password over a dedicated and secure channel. The honeychecker has the capability of detecting any anomalous access. It can trigger an alarm when something irregular has been detected, which will be followed by a reaction by administrator or computer system, such as the login being denied [163].

6.2. Review of Honeywords

The concept of honeywords is to provide a technique to detect whether someone accessing password files has been genuinely invited in. Essentially, the scenario of honeywords is one where any user u_i is provided with a list of k words, called "*sweetwords"*, which are denoted as $W_i = \{w_1, w_2, w_3, \dots, w_k\}$. One of these sweetwords (for instance w_j) is the authentic user's password, while the rest of the list $(k-1)$ are fake and called *honeywords*. The main architecture feature is a new server, "the honeychecker", which contains a database for each user u_i and an index $c(i) = j$, where $w_j \in W_i$ is the correct password of u_i.

The real password of the authorised user will be generated and entered during the registration stage. On the basis of this password, the system generates and adds $(k-1)$ honeywords. Moreover, the honeywords generation algorithm is targeted at creating decoy passwords that are the same as the real one, so that an intruder will not

be able to recognise them from the real password. Accordingly, the system chooses a random $1 \leq j \leq k$, gives the real password to $w_j$and populates the set W_i with the generated honeywords. Finally, the password along with the honeywords are "hashed" and saved in the password file in the form $H_i = \{h_1 = \text{hash}(w_1), \dots, h_k = \text{hash}(w_k)\}$, while the index $c(i) = j$ is stored by the honeychecker.

When u_i logs onto the system, he or she should enter the password and then, the system will check $\text{hash}(p)$ against each hashed sweetword in H_i. If the password that has been entered does not match with any elements of H_i, the connection is rejected. In contrast, let j be such that $\text{hash}(p) = h_j$, then the pair u_i, j will be sent to the honeychecker. Hence, if $j = c(i)$, then the authentication succeeds, and the honeychecker will send back its "approval", whilst otherwise an alarm is triggered, as the password file has probably been attacked. Figure 6.1 explains the main idea of the decoy passwords [164], whilst Table 6-1 illustrates the related notation.

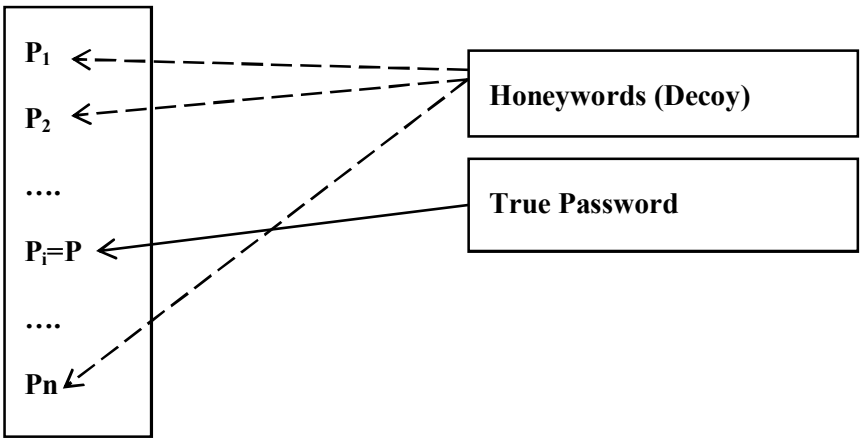

Figure 6-1 illustration of the main idea of the decoy passwords

Table 6-1 Related notation

Notations	Meaning
u_i	i^{th} user in system
P_i	password of i^{th} user
W_i	tuple of passwords stored for u_i
k	number of elements in W_i
$c(i)$	index of correct password in W_i
sweetword	each element of W_i

6.3. Limitations of Honeywords

Despite of the fact that current honeyword based methodologies can provide security against brute force attacks, they do have some limitations, which are described below.

a. ***Co-relational hazard:*** If a relationship exists between the username and password, then the real password of the user can easily be recognised from the list W_i. In such cases honeywords cannot protect the original password, because of this association.

b. ***Distinguishable well-known password patterns:*** If a user chooses a password linked to some well-known object/fact, then an adversary can simply recognise the real password. For example, some of the passwords belonging to this category are bond007, james007, 007bond and 007007, which were found in a list of 10,000 or 500 most common passwords (these will be used to generate the honeywords in this chapter).

c. ***Issue related to DoS resistivity:*** If an adversary knows the real password of the user, then he or she can recognise the honeywords and then, can intentionally submit honeywords to produce a false negative feedback signal by the "honeychecker". If the adversary obtains these honeywords from several accounts, then the entire web server may become blocked. This is known as a Denial-of-Service (DoS) attack and the real password of the user should be not giving any knowledge about system generated honeywords to avoid one.

d. ***The issue relating to Multiple System Vulnerability:*** If a user uses the same password in two (or more) different systems, and if two systems are employing the same honeyword generation algorithm, when an adversary gets access to both systems, then Multiple System Vulnerability can occur. In this case, the adversary may obtain two lists of W_i for the user u_i. Let $W_i^{S_j}$ refer to the list of sweetwords for user u_i in the system S_j. So, if the honeywords that have been generated belong to $W_i^{S_p}$ and $W_i^{S_q}$ (where p ≠ q) are different (probability of which is close to 1), then by performing the connection operation $W_i^{S_p} \cap W_i^{S_q}$ the unauthorised user can obtain the real password. This is known as Multiple System Vulnerability (MSV) of the honeyword based authentication technique [165].

6.4. Personal Information in Passwords and Human Behaviours

A text-based password is the most common authentication method and is likely to remain so for the foreseeable future. Whilst users have been recommended different types of authentication mechanisms, passwords are still considered the best way to protect access to a system. That is, none of the alternative technique can provided all the benefits of passwords without introducing extra burden to the users. However, passwords have been criticised as being one of the weakest techniques in relation to authentication. One of the key reasons for this weakness can be put down to the limitations of the human memory. For, as a consequence, most passwords rather than being real random strings and hence, quite strong, are simple so they are easy to remember [166]. Basically, people prefer to create passwords according to their personal information, because of the limitation of their memory capacity and a random password can be difficult to remember [167].

6.5. List of the Worst Passwords

Not only do most users create an easy password because they can easily remember it, for they often also use the same one in several systems. Whatever the case, frequently they are easy to guess by the intruder. A list of the 500 worst passwords has been created by researchers to help users avoid selecting them. Unfortunately, one in nine users used one from this list and one in 50 uses a password from the top 20 [168].

6.6. Honeywords Generation Method and Discussion

In this section, some of the honeywords generation methods are discussed.

6.6.1. Chaffing-by-Tweaking

This method involves tweaking the real password by selecting the character positions that will be tweaked to produce the honeywords, so the user password will be the seed of the generator algorithm. The same type of character will be selected:

letters are replaced by letters, digits by digits, and special characters by special characters. For instance, when $t = 3$ and the last t characters have been selected for tweaking, the method for the generator algorithm is $Gen(k, t)$. While another approach called "*chaffing-by-tweaking-digits*" is carried out by tweaking the last t positions that contain digits. For instance, if the last algorithm has been used, then for the password *42hungry* and $t = 2$, the honeywords *12hungry* and *58hungry* may be generated.

Remark 1. Most people prefer to choose the numbers involved in passwords relating to a special date (birthday, anniversary or any other historical event). For this reason, it is highly probable that such a password includes a digit sequence like *19xx*, *20xx* or *xx*, where *xx* represents the last two digits of the date. Hence, for those passwords that involve applying the *chaffing-by-tweaking-digits* method, the date digits will be replaced with randomly selected digits. Basically, an adversary will recognise the true password easily among the honeywords. For example, assume the honeywords are generated with $t = 4$ and $k = 9$ for the password *omar1974*. It can clearly be seen below that the digits in the honeywords do not relate to a specific date and hence the correct password, *omar1974*, is logically deducible by an adversary [169].

omar8372 omar9168 **omar1974**

omar2107 omar9607 omar5782

omar1439 omar8274 omar9510

6.6.2. Chaffing-with-a-Password Model

In this technique, the generator algorithm takes the password from the user, and then a probabilistic model of the original passwords is relied upon to generate the honeywords. To give an example of applying this technique, known as *modelling syntax*, the model divides the real password into character sets. For example, the password *mice3blind* is decomposed as four-letters + one-digit + five-letters (L4+D1+L5) and is replaced with the same structure, such as in *gold5rings*.

Remark 2. There are some well-known patterns that have appeared when a password database has been leaked. For example, all of the following passwords are

included in the list of the 10,000 most common passwords and in the 500 worst passwords list.

bond007 james007

007bond 007007

So, the adversary will easily identify the real password, if it is one of these generic passwords [169].

6.6.3. Hybrid Method

This method involves combining of the strength of different honeyword generation methods, e.g. *chaffing-with a-password model* and *chaffing-by-tweaking-digits*. For instance, let the original password be *apple1903*, then the honeywords *angel2562* and *happy9137* might be produced as seeds to *chaffing-by-tweaking-digits*. For $t = 3$ and $k = 4$, for each seed, the honeywords will be as follows:

happy9691 apple1242 angel2656

happy9787 **apple1903** *angel2360*

happy9370 apple1271 angel2498

happy9129 apple1927 angel2625

Remark 3. This method will reduce the chance of an adversary recognising the real password. Nevertheless, the previous remarks are still valid for this case. That is, an intruder may make reasonable guesses regarding the real password [169].

6.7. The Proposed Honeywords Generation Algorithm

In the proposed honeywords generation algorithm, dictionary attacks, personal questions-answers, the 500 worst passwords list and character shuffles have been used to generate the honeywords. The aim is to increase the flatness of the honeywords, thereby making the adversary confused when trying to identify or recognise the real password. To this end, the new honeywords generation algorithm has been created. The scenario for honeywords generation is the same as the traditional one, whereby a list of k honeywords is provided for user u_i, denoted as $W_i = \{w_1, w_2, w_3, \dots, w_k\}$. One of these honeywords (for instance w_j) is used as the real password, while the

remaining W_i $(k-1)$ are fake, with the aim being as aforementioned to increase the flatness.

The proposed honeywords generation method with a password includes at least one letter and one digit. Figure 6-1 illustrates of the whole structure of the proposed honeywords generation method.

Step 1

Analysing the password:

a- How many digits?

b- How many letters (Uppercase and lower case)?

c- How many special characters?

Step 2

Generating the Honeywords

1- Creating a database containing the public personal questions (50-60 questions), which are divided into two parts according to the type of answers. The first part is associated with the names, and will be generated as letters (for example, your nickname, city you like, your favourite team, pet's name and so on). Whilst the second part will be relating to digits (date of birth, anniversary, best year in your job and so on). Six questions will be chosen randomly from the database (three from each part). Then, five honeywords will be generated by combining the first part answers with the second. Any user can ignore any question, if he/she does not want to answer it and immediately, this question will be replaced by another. In addition, if there are just two digits in the original password, then the algorithm will select that number for the honeywords from the digits answers (This group is called G1).

For example:

a- Letters part: Nickname: Mero, Child's name: Peter, City: London

b- Digits part: Best year in your job?: 2005, In which year was your father born?: 1948, In which year did you have your last long journey? 2014

The honeyword results will be:

Mero2005 London1948 Peter2005 Mero1948 London2014

2- This type of honeywords is generated based on a dictionary attack, with four being created for this type of group. The principle behind how to make

suitable honeywords is about searching through the dictionary attack and using the real password with a difference of up to three digits or letters (This group is called G2).

Note: Some passwords are not applicable with this type of group due to their being too difficult to find in the dictionary attack, in which case four honeywords will be generated from the other groups.

3- This group of honeywords is made according to the 500 worst passwords list; with five being chosen randomly from this list (This group is called G3).

4- This type of honeywords is made by shuffle and then some letters or digits from the ID user name are mixed with this. Subsequently, the real password together with some digits and letters are generated to be inserted in the honeywords, with meaningless words then being generated. In this step ten honeywords are created (This group is called G4).

Special cases

5- If there is a special character(s) included in the password, then the honeywords will contain the same number of these generated randomly.

6- The number of uppercase letters in the password will equal the number in the honeywords.

If there are two words the same in list W_i for the user u_i, then the algorithm explained in Figure 6-2 will applied. Basically, if the original password is one of these two words then the honeyword will be replaced. The repeated words are not allowed in this algorithm as these may give an indication to the attacker about where the real password is.

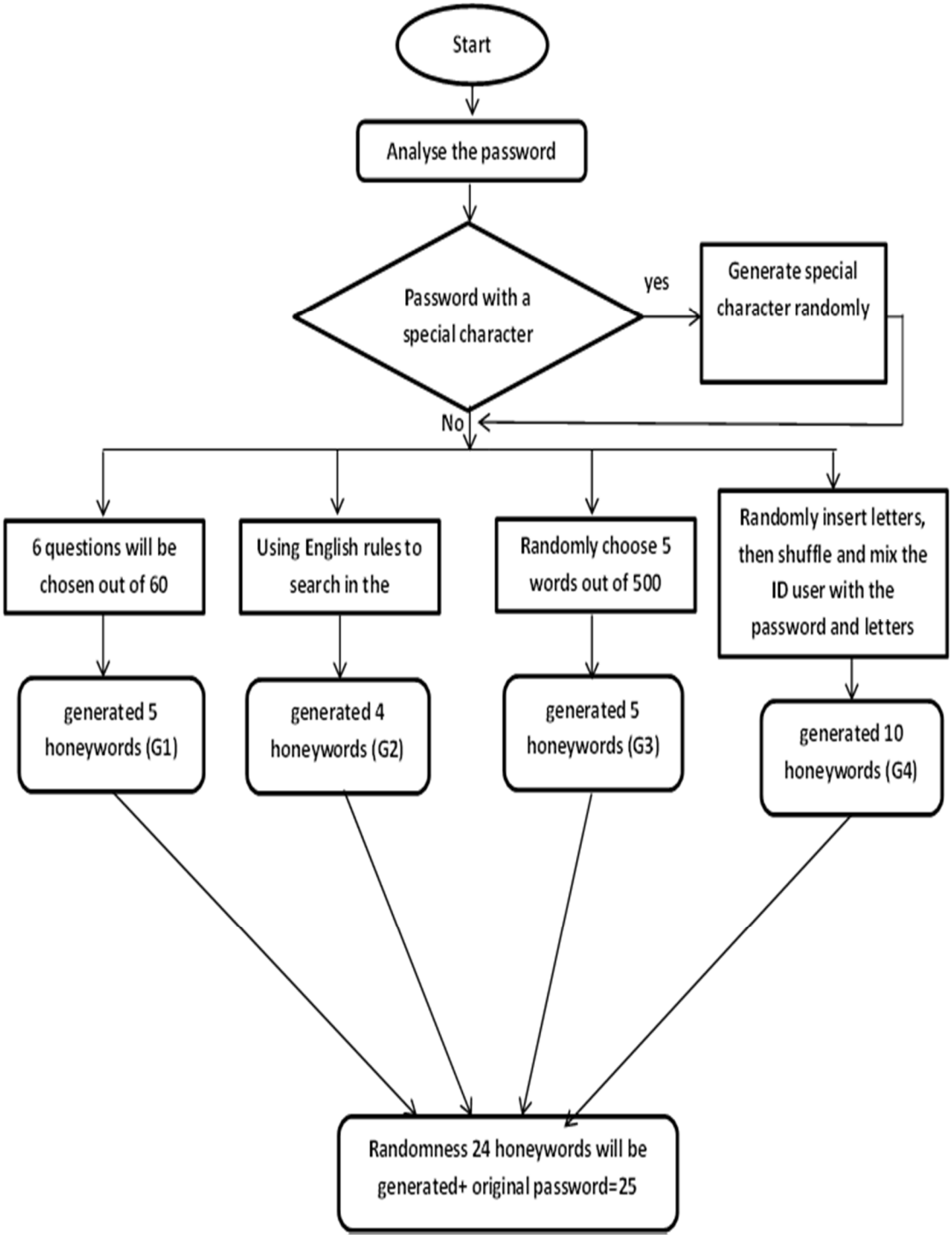

Figure 6-2 The general structure of the proposed honeywords generation method

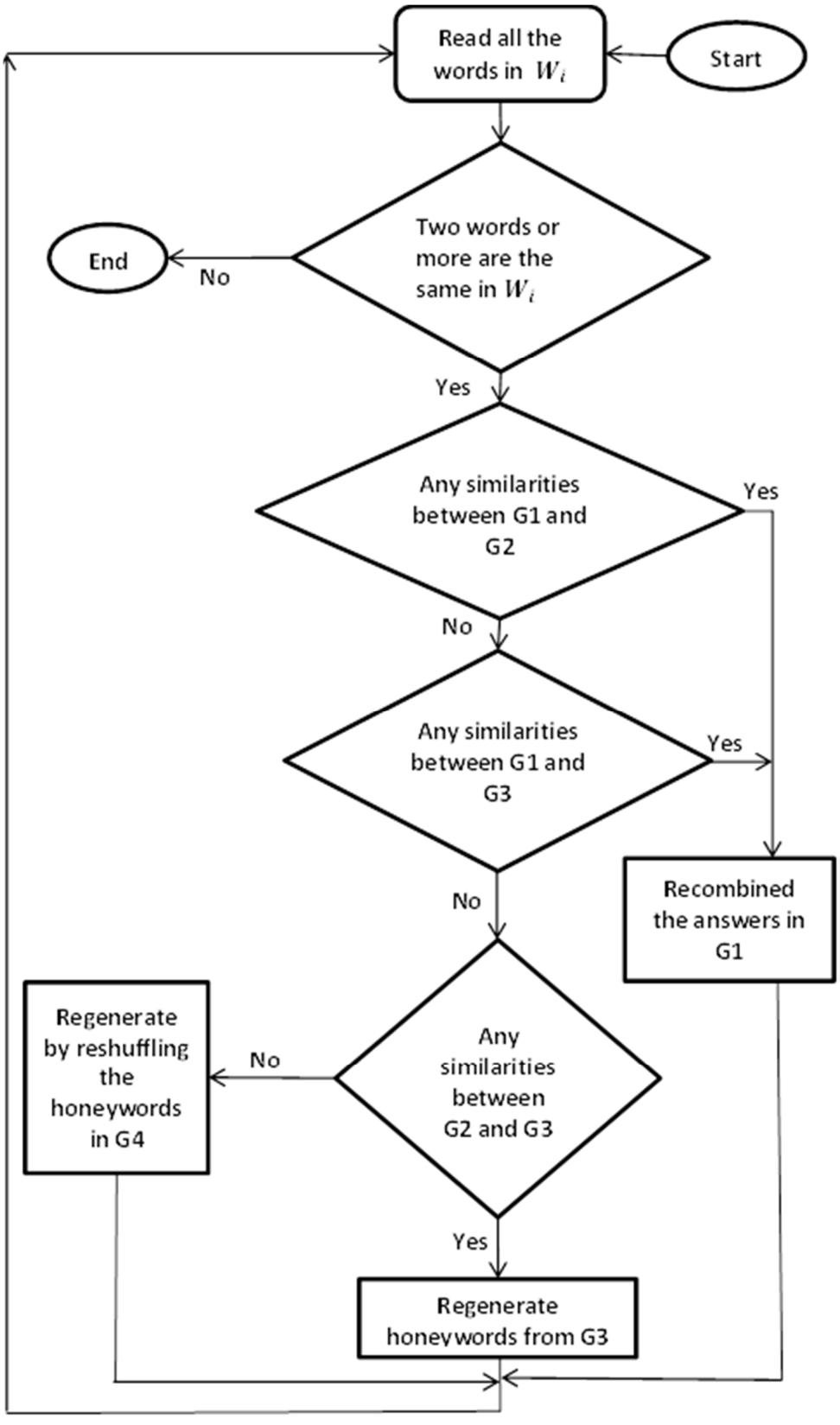

Figure 6-3 The algorithm when two or more honeywords are the same in W_i

6.8. Discussion the Flatness in the New Honeywords Generating Method

The flatness in the proposed method and how an adversary tries to analyse the honeywords in W_i for each $u_i$is discussed in this section. Obviously, the adversary does not have any predefined information about the password; however, he/she will try to analyse W_i to find any information about the c_i. The honeywords created in this first group are associated with personal questions, which will most probably lead to personal answers. In this case, six answers, which are either in letter or digit form and the letters and digits are then randomly mixed to produce five honeywords. The high level of association of these honeywords with a user $u_i's$ real answers will make it difficult for the adversary to identify which one is the real password, i.e. this increases the flatness. In contrast, the traditional methods do not take into consideration whether there is a personal password, because all the honeywords are generated by tweaking some letters or digits in the real password.

It is clear that *chaffing-by-tweaking* has many problems; the first relates to when a digit is replaced by another. That is, generation of a honeyword does not relate to human dates, starting either with 19xx or 20xx. The second problem is that not only is the digits tweaking easily recognised by the adversary, but also, he/she can easily find the password when letters are replaced. For instance, when t=3, this means three letters will be replaced randomly by others, which results in the meaning of the original password not being present in the honeywords and hence, they are completely distinct from the former. So, recognising the original password, which is the meaningful word among meaningless ones, will be very easy. For these reasons, the first group has been generated based on personal information, because most users continue to use this when they create their passwords.

Dictionary attacks are commonly used to break passwords, but in the proposed method they are used to generate the honeywords. Such an attack involves most of the passwords that have been created by users around the world, by using an algorithm based on English language rules to make the search in this dictionary to find honeywords very close to the original password. This algorithm tries to find words in the dictionary attack with the most same letters or digits with up to plus or minus three characters or digits. The first priority to find the different words will be

regarding the digits if they are present, otherwise the four words with the closest letter sounds to the actual password are applied. For example, **ch** is mostly pronounced either as /**k**/, as in **ch**aracter, **ch**ord, or as /**tʃ**/, as in **ch**icken, **ch**est. Almost all words containing "chi" or "che" are pronounced with /tʃ/ (note exceptions like "chiropractor" /ˈkaɪərəʊpræktə/ **kaay**-roh-præk-tə and "chemistry" /ˈkemɪstri/ (**kem**-ist-ree), but there's no reliable rule for "cha", "cho", and "chu". The main benefit of using a dictionary attack is that all the honeywords that will be chosen are real passwords generated by users in the past, so the flatness will be very high in this group of honeywords as well.

As aforementioned, there are some passwords that are used commonly by users and researchers have collated them into one list, calling it the worst passwords in the world. Choosing some of them randomly and making a combination of them is a popular procedure for adversaries. Consequently, group three will be generated according to this list.

In addition, a minority of users have a strong password, whereby they select some letters randomly and create a meaningless one. However, most users in this group still select letters or digits from their names and/or personal dates. To avoid these types of passwords, with the proposed method and the goal of flatness, a fourth group is created. Some characters are chosen from the original password and ID and then some digits and letters are generated randomly to be inserted into the honeywords.

6.9. Analysis of the Security of the proposed Generation Method: Denial of Service Attack

Using the proposed method will partially reduce the number of DoS attacks. That is, this is an improvement on the current method, because it provides greater resistance by increasing the flatness in the list of honeywords and the real password W_i makes the proposed method stronger than the traditional methods against these attacks. Because w_j in W_i can be either a honeyword or the real password, the flatness will make the attacker confused when trying to guess the real one. In the existing methods, an attacker has to know that the honeywords pertaining to a particular

password have all been generated by random tweaking, which is possible and then he/she can run a DoS attack. With the proposed method the honeywords are generated according to several procedures, and not randomly.

6.10. Discussion of Attacks on the New Method

Table 6-2 illustrates how an attacker could compromise the password by nominating some words from the honeywords table and then, analysing the results.

Table 6-2 discussion on the types of attacks against passwords

	Good Password	Personal Password	Generic Password
Dictionary Attack	Not Working	Not Working	Not Working
Brute-Force Attack	Not Working	Not Working	Not Working
Password Guessing Attack	If the attacker chooses all good honeywords with the password, then he/she will obtain 11 words, one of which is the real password (1/11). In addition, If the attacker is going to choose just the meaningful words, he will obtain 14 words and fortunately the real password is not one of them (0/14). This type of password is not popular, because most users prefer to use passwords that are easy to remember.	If the attacker chooses just the meaningful words. he/she will obtain 15, among which the real password will be included. However, the flatness is very high because the honeywords are coming from real password lists; some of them relating to the user, some of them chosen from the dictionary attack and the final set are chosen from the public list of passwords (1/15). As a result, the guessing method will be chosen by the attacker.	If the attacker chooses just the meaningful words, he/she will obtain 15, among which the real password will be included. However, the flatness is very high, because the honeywords are coming from real password lists, some of which are related to the user, some are chosen from the dictionary attack, and the final set is selected from the public list of passwords (1/15). As a result, the guessing method will be chosen by the attacker.
Clever Attacker	If the attacker chooses all good honeywords with the password, then he/she will obtain 11 words, one of which is	If the attacker chooses just the meaningful words, he/she will obtain 15, among which the real password will be included. However, the flatness is very high,	If the attacker chooses just the meaningful words, he will obtain 15, among which the real password will be included. However, the flatness is very high, because the honeywords are coming from real password

	the real password (1/11). In addition, If the attacker is going to choose just the meaningful words he/she will obtain 14 words, but fortunately the real password is not one of them (0/14). This type of password is not popular, for most users prefer to use those that are easy to remember.	because the honeywords are coming from real passwords lists, some of which are related to the user, some are chosen from the dictionary attack and the final set is selected from the public list of passwords (1/15). Now the attacker has just one choice, which is to try to analyse the words and nominate some of them as the real password.	lists, some of which are related to the user, some of them are chosen from the dictionary attack, and the final set is chosen from the public list of passwords (1/15). Now the attacker has just one choice, which is to try to analyse the words and nominate some of them as the real password.

6.11. The Survey

The survey has been carried out based on the new honeywords generation technique and two traditional techniques (Chaffing-by-Tweaking and Chaffing-by-Tweaking digits) by including 820 participants. Appendix A illustrates 50 samples of this survey from each method. Most of these were students at Brunel University London and the survey was applied through the following procedure:

1. There were many tables of the W_i list containing 24 honeywords and one real password which have been generated by using the new generation method. The passwords were divided into three groups for each W_i (generic, personal and strong passwords). Furthermore, there were other tables for the traditional techniques W_i which were prepared to be used in this survey (Chaffing-by-Tweaking and Chaffing-by-Tweaking digits)
2. First of all, each participant was asked if they partook in a similar survey before. If they have, the user will then not be enlisted in this survey
3. One from each type (generic, personal and strong passwords) of the new generation method list W_i was presented to user at one time
4. For each W_i that was presented to users, there were two requirements to be completed by each user:
   a- To nominate any word(s) they thought were the real password (these were represented by the Blue line in the results and labelled “total nominated”).

b- If the participant chose the real password as one of their selected word in part a-, then this will be registered in the survey and represented as part of the results (the red line titled “frequency”).

c- From the words that have been nominated by each participant in step a-, the user must then select only one word they believed was the real password. If they choose the real password correctly, then the result will be registered and illustrated in the results as green line in the graphs.

The outcomes of the survey demonstrated that most participants were not aware that strong passwords can be used by some users. As a result, participants did not choose or guess the real password when it was a strong one. Though, most participants selected the maximum number of the remaining words (shown in W_i) and this is how flatness was achieved. However, most participants chose the generic honeywords and thought that they were the real password. The results will be discussed in the next section (6.12).

6.12. Testbed and Results

It is a difficult to measure how people are thinking when they are creating a password, because it depends on unpredictable user behaviour. To address this, a testbed engaged with by 820 people was developed to determine whether users can recognise the real password among honeywords. The scenario involved dividing the passwords into three groups: good, personal, and generic. Then, the participants were provided with the W_i, and ask to nominate words that could be passwords, this column being titled “nomination”. The idea behind this step was to ascertain how many people would nominate the real password among the honeywords, and how many words they would choose amongst which they believed the password would be found. Having chosen their words, they were asked to identify the single one that they thought was the real password and if they got it wrong then Intrusion Detection System IDS would trigger attempted intrusion, but if successful access was granted.

The first type, namely the good password, was strong, being created with random letters, digits, and special characters. The results showed that this type of password is very strong, as most people who participated in the testbed experiment

did not choose it among the honeywords, i.e. no one was able to guess the real password when it was good and random.

The second type of password is the personal password, which was created based on information relating to the users. The testbed revealed that the new method is better than the traditional ones. Finally, with the same scenario, the third type of password, i.e. the generic password, was applied.

Figure 6-4 illustrates the results of the strong password for the new method, with the total nominated representing how many words the participants chose, while the frequency is how many people selected a particular amount of passwords. For example, the number of people who nominated 14 words was 224 (27.317%), whereas 11 words were nominated by 78 (9.512%). No one guessed the real password, even if they had nominated it, which shows it was very strong and flat.

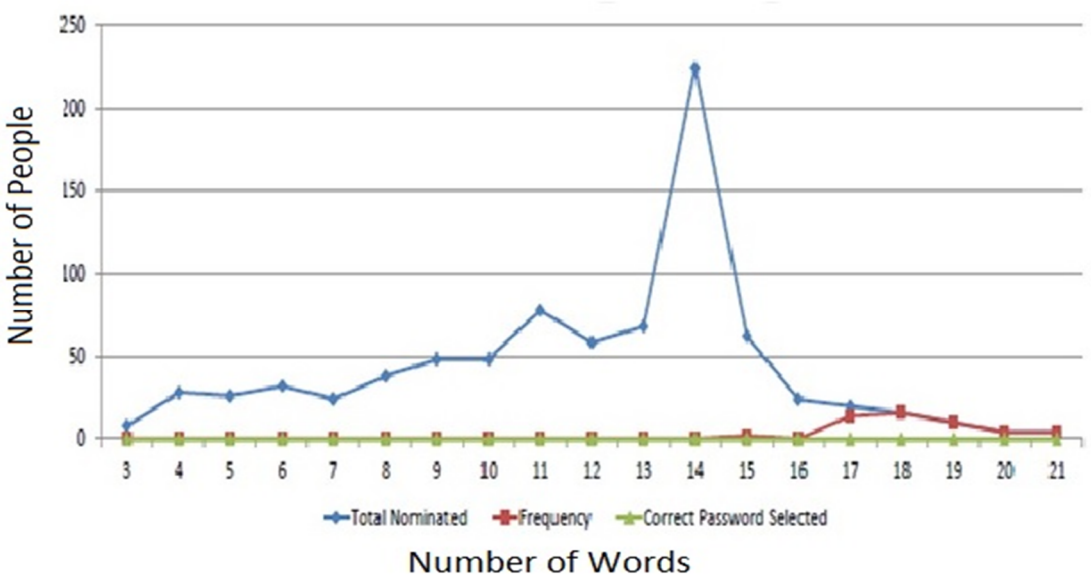

Figure 6-4 The results of the proposed method when a strong password was applied

Figure 6-5 shows the results of the proposed method when the real password is the personal information type and clearly, the number of people who nominated the password amongst their choices increased, being 502 out of 820 (61.219%). Moreover, there were two people who guessed the correct password (0.244%).

Figure 6-6 illustrates the new method when a generic password was the real password and the results show that this type provides the worst outcomes of the three, but the new method still gives better results than with the traditional one. Specifically,

the total number who chose this password was 630 out of 820 (76.829%), and it was guessed correctly 21 times (2.561%). This implies that most attackers focus on generic words.

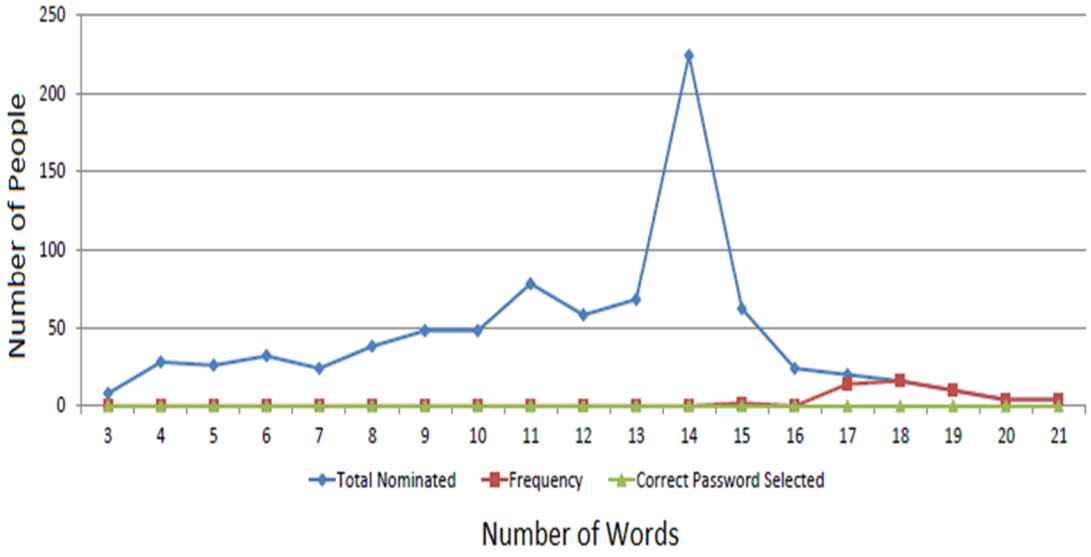

Figure 6-5 The testbed results when the real password contained personal information

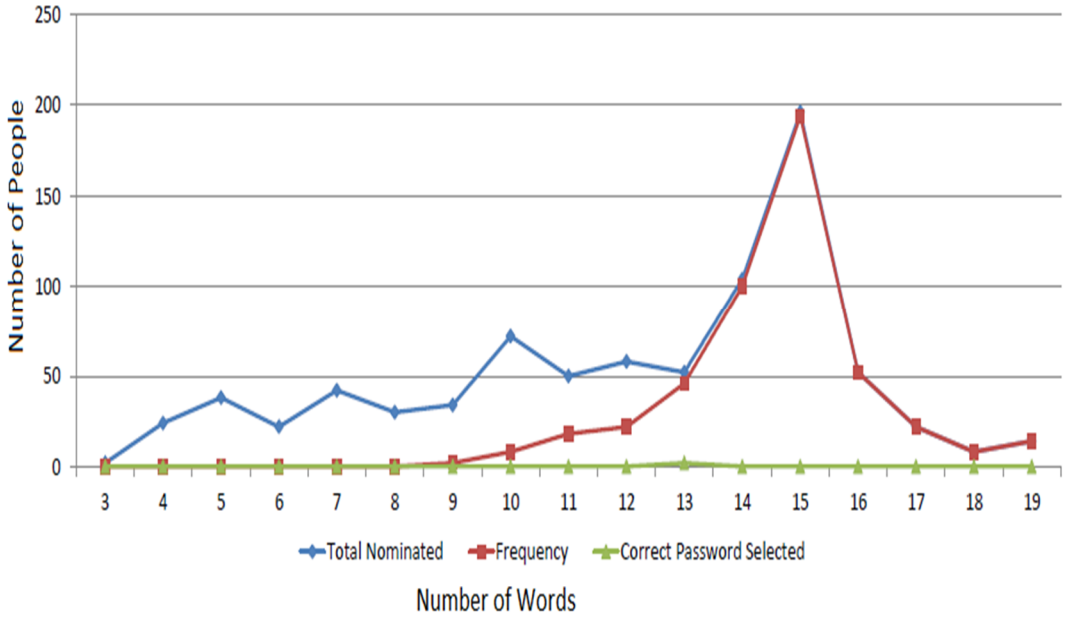

Figure 6-6 The testbed results for the proposed method when the real password is generic

In Figure 6-7 showing the outcomes when Chaffing-by-Tweaking was applied in the testbed, it is clear that the number of participants guessing the real password

was very high, standing at 794 times out of 820 (96.829%), whilst the number who nominated was 812 (99.024%). Moreover, most people nominated just one or two words out of 25 (3.048%) in W_i and no one nominated more than six, which suggests that many were confident from the beginning which was the correct password.

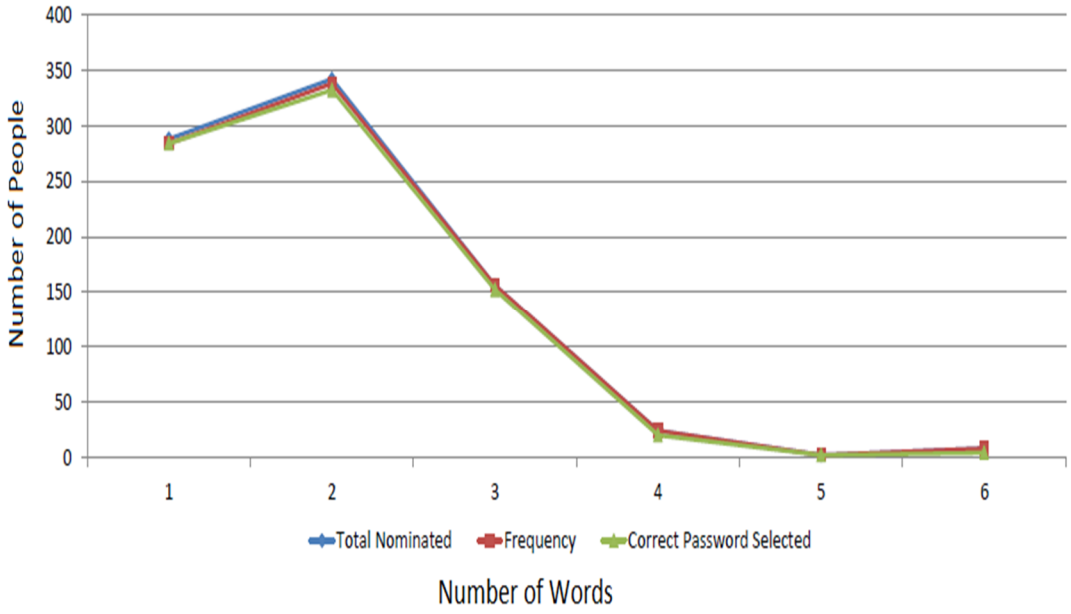

Figure 6-7 The testbed results for the traditional method of "Chaffing-by-Tweaking

In Figure 6-8 the results for the traditional method of Chaffing-by-Tweaking-Digits are shown. This method provides slightly better results than Chaffing-by-Tweaking in that the password was guessed correctly 756 times out of a possible 820 (92.195%). To give an example of how the proposed method generates the honeywords, in Table 6-3 the password is “Ujemgzae91#e”. Clearly, the first row contains honeywords generated based on personal information, while the second has those created based on the worst password list. The rest of the table was generated by shuffling the letters and digits. A dictionary attack was not used in this table, because no word is similar this password.

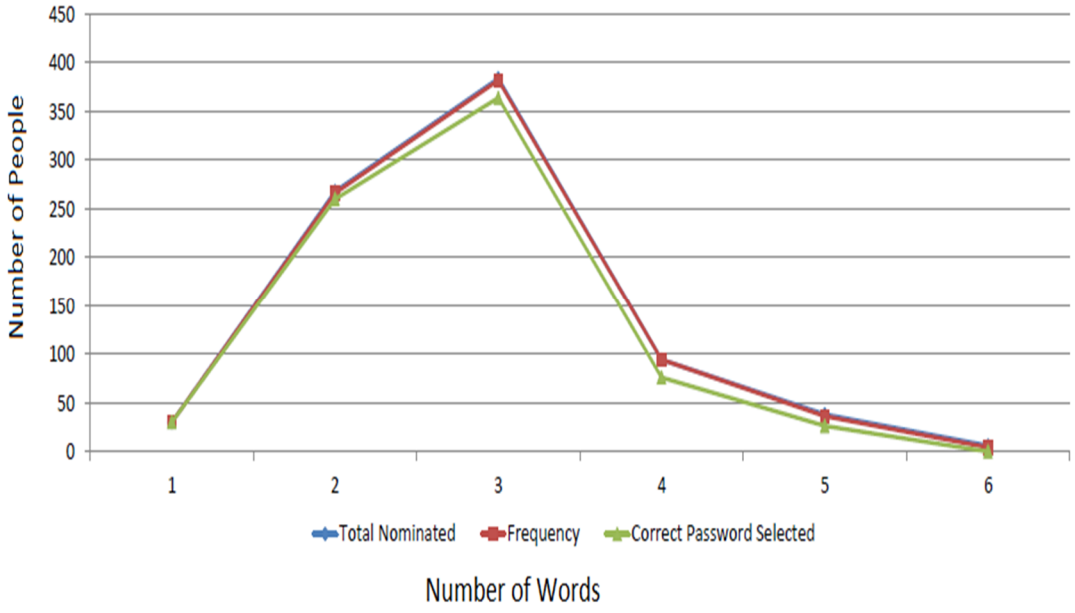

Figure 6-8 The testbed results for the traditional method of "Chaffing-by-Tweaking-Digits"

Table 6-3 Testbed with the new method and a good password

Prestol#70	Jordy$86	Steves@75	Mechaill$81	Anna^1945
Liverpool@2005	Football&1234	Password*1111	Music@6666	bond@007
Booboo&75	Love&2014	Mustang@16	Zme1qo@55req	Epalm#1999ks
Pufna*37xy	Msac^hs31	Neadjg_69	Vlpheo$10r	Kp#12zxme
Ltcbas!00j	Tg36$ewba	**Ujemgzae91#e**	Rpnq#fxg	Lsczyr&12

Table 6-4 illustrates an example when the testbed was applied with the generic password, “password222”, being drawn from the list of worst passwords. The honeywords in the second row were generated based on a dictionary attack.

Table 6-4 Testbed with the new method and a generic password

StationRoad1960	Church2016	Morgan2010	Stevs1958	Andy2000
Alunaliceza	Andralice2004	Anasialice1977	Anaalice85	Hello131313
Nicholas123	Andrew 1212	**Password222**	Welcome777	Alice1974
ElArzd204	O9lefcm7ss	Oxsr15dox	Z7erpmc0	Enm12q
Movxg20w	Qica12r00	Hvagjr4193	Nlpqroo1870	Zaqu2w88

6.13. Summary

In this chapter, a new honeywords generation method has been proposed, which was developed to overcome the problems that exist with the traditional methods. The proposed method is based on personal information, dictionary attacks, the worst password list (generic passwords) and shuffling the characters. User behaviour is the underpinning principle of the new method, because creation of the passwords differs from one user to another. Some limitations regarding the extant honeywords methods have been discussed and these have been overcome by the proposed method, as has been explained. A testbed was applied to obtain the results using 820 participants and these have shown that the generic passwords are the weakest type of passwords due to being easy recognised. 76.829% of the participants nominated the password within the honeywords, whereas 2.561% participants targeted the correct password. The strongest type of passwords are termed "good passwords" due to not being guessed by any participant, however, this type of password is not commonly used by the users due to users' behaviours (e.g. forgetting the password). Personal information passwords are widely used by users, but this type of passwords provided 15.61% better results when compared with the generic password according to the nomination of the passwords and 2.317% according to correct choice of the password.

Based on the traditional methods that have been tested in this survey (Chaffing-by-Tweaking and Chaffing-by-Tweaking-Digits), it has been demonstrated that the new method is better than the traditional ones by 89.634% in the worst case which is based on passwords being chosen correctly. Additionally, the results have shown that Chaffing-by-Tweaking-Digits is better than Chaffing-by-Tweaking by 4.634%.

Chapter Seven: Conclusion and Future work

7.1. Introduction

This chapter summarises the thesis outcomes for all four techniques contributed to the field through this work. Furthermore, future work proposals and recommendations are also covered.

7.2. Conclusion

In this thesis, new authentication techniques based on dwell time, web-session honeypots, multi-factors, and honeywords have been introduced. Clearly, sensitive datasets must be protected by strong methods than traditional one. Hence, two out of the four novel techniques, which are targeted at protecting sensitive datasets, are based on changing user behaviour.

For the first technique, the authentication password relies on a specific dwell time generated by using shift-key to type the uppercase letters that are included in the password (two or three letters). To measure the behaviour of users when typing the uppercase letters, a survey was carried out. The main goal of this technique is to increase the possible combinations for the password, thus mitigating attacks more effectively than with the traditional password. A mathematical model has been derived from the traditional password measurements and PQI was used to compare the new method with the traditional password in terms of performance. The results pertained to three mathematical models: fixed positions, two variant positions, and three variant positions, for the timed keystroke approach. The findings show that the proposed technique is better than the traditional password by 192 %, when C = 93 and the length of the password is 4, whilst this improvement is up to 200% when the length is 15.

The second authentication technique applied in this thesis is a honeypot based on web-session management. The legitimate user has to go through a specific sequence before performing the login process, which is available on all the webpages so as to lure an adversary to undertake the login process without going through the correct sequence and hence, detection of such anomalous behaviour will be easy.

Another aim of this technique is to mitigate the attacks on the password by increasing the possible of password combinations. PQI has been used to make a comparison between its results for the traditional password and those of the new honeypot technique. It has been pointed out that the effectiveness of the novel honeypot technique is associated with the number of links, whereby increasing their number will increase the strength of the authentication process. The results have shown that the new technique is better than that of the traditional password by up to 200%

The third contribution is a multi-factor authentication technique, where the two previous authentications techniques have been combined together. For this method, a legitimate user must go through a predetermined specific sequence and when he/she reaches the require webpage the login with username and password must be carried out. The login process must include the dwell time generated by the shift-key for the specific uppercase letters that are a part of a password. The results have been calculated when $\Gamma = 50$ (the number of links) with fixed positions, two variant positions, and three variant positions regarding the specified keystrokes. The results have shown that the multi-factor technique is better than the traditional password authentication technique as well as the dwell time and honeypot techniques when they are applied separately.

The fourth and last technique provided in this work is honeywords (decoy passwords), a new generation method based on the user's behaviour. The new honeywords have included words from users' personal information, dictionary attacks, the worst and most common passwords lists and finally, shuffling and inserting letters and digits. The aim of this technique is to achieve flatness to confuse the adversary when trying to recognise the real password among the honeywords. The limitations of traditional honeywords generation methods have been discussed and explained. The outcomes were obtained through a survey of 820 people. The passwords were divided into three types: good passwords, public passwords, and personal passwords. In sum, the results indicate higher flatness for the new honeywords generation method than with the traditional methods.

In this thesis, two main goals were achieved with relation to enhancing the security of the authentication technique. The first of which was the detection of intruders as demonstrated in the honeypots, the honeywords and the multi-factor techniques. The second goal was to increase the possible combinations for the

password in order to make it strongest than before. The results showed that the multifactor technique had a 100% improvement of security when compared to the timed password and the honeypots techniques. Overall, a solid conclusion which can be drawn from the attained results is that the strongest password was acquired when C=93 and m=15, while the weakest password for each technique was when C=26 and m=4. Additionally, all techniques in which the dwell-time period was used (timed-password) were based on uppercase letters.

7.3. Future Work

The following recommendations are put forward regarding future authentication techniques.

The first would be to combine the proposed dwell time technique with a graphical password. The first step would involve selecting a picture among a group of them. Then, the legitimate user would have to move the mouse to specific locations (points) on the chosen picture by clicking at a particular (locations) and holding on for a specified dwell time.

The second recommendation is create a new multi-factor technique based on honeypots, for which two techniques, such as fingerprint and dwell time are applied together. Because of the fingerprint biometric technique is could be compromised, as explained in chapter two, adding a new element would make the authentication process much stronger in protecting sensitive datasets. Specifically, the authentication process would be started with the use of a fingerprint, followed by a message confirming that the user has passed the fingerprint stage successfully. However, this is a fake message if the fingerprint has matched and in fact is a honeypot for luring the user into thinking that his/her figure has been read correctly. In contrast, the legitimate user knows that he/she must keep his/her finger on the sensor for the correct dwell time.

The third proposal is to expand the number of ways for selecting honeywords. To achieve this, neural networks and fuzzy logic will be added to the learning system to achieve higher accuracy when selecting the words from dictionary attacks, the

common 10,000 passwords list, and the worst 500 passwords list. In other words, new rules will be used to increase the flatness.

7.4. Research Impact

There is no doubt that protecting sensitive information is the main concern for the users and programming developers. Attackers are constantly improving their techniques for hacking into systems by compromising users' passwords and hence, ongoing enhancement of security techniques is essential to protect such sensitive information. The outcomes from this research can be applied to a sensitive dataset saved in big data, with the timed password requiring legitimate users to have the ability to change their behaviour to order to access to the sensitive information for editing or viewing. In the event of applying web-session management to an extremely sensitive dataset and links that are available on a website, the session management will enhance the password's sophistication, making it very difficult for the adversary to compromise the password. Moreover, a multi-factor technique can be utilised if the legitimate users have the ability to change their behaviour and use correct dwell time periods as well as web-session management. While the honeywords technique can be successfully applied in all password-based authentication systems, it requires a bigger size of memory than traditional methods.

Appendix A

Appendix A illustrates samples of 50 participants out of 820 for each generation method was contributed in this survey.

#	Good Password		
	Nominations	Password Selected	Correct password
1	17	0	0
2	15	0	0
3	14	0	0
4	18	1	0
5	20	1	0
6	11	0	0
7	9	0	0
8	15	0	0
9	15	0	0
10	11	0	0
11	15	0	0
12	18	1	0
13	14	0	0
14	14	0	0
15	15	0	0
16	19	1	0
17	14	0	0
18	19	1	0
19	10	0	0
20	14	0	0
21	11	0	0
22	15	0	0
23	16	0	0
24	18	1	0
25	15	0	0
26	12	0	0
27	20	1	0
28	21	1	0
29	16	0	0
30	15	0	0
31	12	0	0
32	14	0	0
33	15	0	0

Personal Password		
Nominations	password Selected	Correct password
12	1	0
15	1	0
14	1	0
16	1	0
19	1	0
11	0	0
14	1	0
15	1	0
17	1	0
12	0	0
15	1	0
17	1	0
16	1	0
15	1	0
15	1	0
17	1	0
15	1	0
19	1	0
10	0	0
15	1	0
13	1	0
15	1	0
16	1	0
16	1	0
15	1	0
13	0	0
19	1	0
19	1	0
18	1	0
15	1	0
10	0	0
15	1	0
15	1	0

34	17	0	0	16	1	0
35	15	0	0	16	1	0
36	14	0	0	15	1	0
37	10	0	0	12	0	0
38	15	0	0	15	1	0
39	14	0	0	15	1	0
40	14	0	0	16	1	0
41	12	0	0	15	0	0
42	19	1	0	17	1	0
43	14	0	0	15	1	0
44	14	0	0	16	1	0
45	14	0	0	11	1	0
46	12	0	0	12	1	0
47	14	0	0	15	1	0
48	21	1	0	19	1	0
49	15	0	0	13	1	0
50	14	0	0	15	1	0

#	Good Password			Personal Password		
	Nominations	Password Selected	Correct password	Nominations	password Selected	Correct password
1	17	0	0	12	1	0
2	15	0	0	15	1	0
3	14	0	0	14	1	0
4	18	1	0	16	1	0
5	20	1	0	19	1	0
6	11	0	0	11	0	0
7	9	0	0	14	1	0
8	15	0	0	15	1	0
9	15	0	0	17	1	0
10	11	0	0	12	0	0
11	15	0	0	15	1	0
12	18	1	0	17	1	0
13	14	0	0	16	1	0
14	14	0	0	15	1	0
15	15	0	0	15	1	0
16	19	1	0	17	1	0
17	14	0	0	15	1	0
18	19	1	0	19	1	0
19	10	0	0	10	0	0
20	14	0	0	15	1	0
21	11	0	0	13	1	0
22	15	0	0	15	1	0
23	16	0	0	16	1	0
24	18	1	0	16	1	0
25	15	0	0	15	1	0
26	12	0	0	13	0	0
27	20	1	0	19	1	0
28	21	1	0	19	1	0
29	16	0	0	18	1	0
30	15	0	0	15	1	0
31	12	0	0	10	0	0
32	14	0	0	15	1	0
33	15	0	0	15	1	0
34	17	0	0	16	1	0
35	15	0	0	16	1	0
36	14	0	0	15	1	0
37	10	0	0	12	0	0
38	15	0	0	15	1	0
39	14	0	0	15	1	0
40	14	0	0	16	1	0
41	12	0	0	15	0	0
42	19	1	0	17	1	0

43	14	0	0
44	14	0	0
45	14	0	0
46	12	0	0
47	14	0	0
48	21	1	0
49	15	0	0
50	14	0	0

15	1	0
16	1	0
11	1	0
12	1	0
15	1	0
19	1	0
13	1	0
15	1	0

#	Generic Password			Chaffing-by-Tweaking		
	Nominations	password Selected	Correct password	Nominations	password Selected	Correct password
1	13	1	0	1	1	1
2	12	0	0	1	1	1
3	15	1	0	2	1	1
4	16	1	0	1	1	1
5	19	1	0	1	1	1
6	15	1	0	1	1	1
7	15	0	0	2	1	1
8	15	1	0	1	1	1
9	16	1	0	1	1	1
10	16	1	0	3	1	0
11	14	1	0	1	1	1
12	17	1	0	1	1	1
13	15	1	0	1	1	1
14	16	1	0	2	1	1
15	14	1	0	2	1	1
16	17	1	0	1	1	1
17	15	1	0	2	1	1
18	17	1	0	1	1	1
19	9	0	0	1	1	1
20	15	1	0	1	1	1
21	12	1	0	2	1	0
22	15	1	0	1	1	1
23	16	1	0	1	1	1
24	16	1	0	1	1	1
25	15	1	0	1	1	1
26	15	1	0	1	1	1
27	16	0	0	2	1	1
28	20	1	0	1	1	1
29	17	1	0	1	1	1
30	14	1	0	3	1	1
31	11	0	0	1	1	1
32	15	1	0	1	1	1
33	14	1	0	1	1	1
34	16	1	0	1	1	1
35	15	1	0	2	1	1
36	16	0	0	1	1	1
37	11	0	0	2	1	0
38	16	1	0	1	1	1
39	15	1	0	1	1	1
40	15	1	1	2	1	1
41	14	0	0	1	1	1
42	20	1	0	2	1	1

43	15	1	0	2	1	1
44	11	1	1	1	1	1
45	10	0	0	2	1	1
46	11	0	0	2	1	0
47	15	1	0	1	1	1
48	20	1	0	1	1	1
49	15	1	0	2	1	1
50	14	0	0	3	1	1

#	Chaffing-by-Tweaking-Digits		
	Nominations	password Selected	Correct password
1	3	1	1
2	3	1	1
3	2	1	1
4	3	1	1
5	2	1	1
6	2	1	1
7	3	1	1
8	2	1	1
9	2	1	1
10	3	1	1
11	1	1	1
12	2	1	1
13	3	1	1
14	2	1	1
15	2	1	1
16	2	1	1
17	3	0	0
18	4	1	1
19	1	1	1
20	3	1	1
21	3	1	1
22	1	1	1
23	3	1	1
24	2	1	1
25	3	1	0
26	1	1	1
27	1	1	1
28	3	1	1
29	3	1	1
30	3	1	1
31	1	1	1
32	3	1	1
33	3	1	1
34	2	1	1
35	1	1	1
36	2	1	1
37	3	1	1
38	2	1	1
39	2	1	1
40	3	1	1
41	1	1	1
42	2	1	1

43	2	1	1
44	3	1	1
45	3	1	1
46	2	1	1
47	3	1	1
48	2	1	1
49	3	1	1
50	2	1	1

References

[1] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 10–28, 2017.

[2] S. M. Gurav, L. S. Gawade, P. K. Rane, and N. R. Khochare, "Graphical password authentication: Cloud securing scheme," *Proc. - Int. Conf. Electron. Syst. Signal Process. Comput. Technol. ICESC 2014*, pp. 479–483, 2014.

[3] H. J. Highland, "Security in computing," *Comput. Secur.*, vol. 16, no. 3, p. 181, 1997.

[4] B. Rannala, *The art and science of species delimitation*, vol. 61, no. 5. Addison-Wesley Longman Publishing Co., Inc., 2015.

[5] M. Belk, P. Germanakos, C. Fidas, and G. Samaras, "A Personalization Method Based on Human Factors for Improving Usability of User Authentication Tasks," in *22nd Conference on User Modeling, Adaptation and Personalization*, 2014, pp. 13–24.

[6] M. E. Kabay, C. Christian, K. Henry, and S. Schneider, "Professional Certification and Training in Information Assurance," *Comput. Secur. Handbook, Sixth Ed.*, pp. 71–74, 2014.

[7] W. Meng, W. Li, L. F. Kwok, and K. K. R. Choo, "Towards enhancing click-draw based graphical passwords using multi-touch behaviours on smartphones," *Comput. Secur.*, vol. 65, pp. 213–229, 2017.

[8] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," in *Proceedings - IEEE Symposium on Security and Privacy*, 2014, pp. 689–704.

[9] F. Wei, P. Vijayakumar, J. Shen, R. Zhang, and L. Li, "A provably secure password-based anonymous authentication scheme for wireless body area networks," *Comput. Electr. Eng.*, vol. 0, pp. 1–10, 2017.

[10] V. Taneski, M. Hericko, and B. Brumen, "Password security - No change in 35 years?," in *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014 - Proceedings*, 2014, pp. 1360–1365.

[11] L. Ma and M. Jiang, "Chances and challenges confronting securities industry and the countermeasures in big data and cloud computing era," *Proc. 9th Int. Conf. Comput. Sci. Educ. ICCCSE 2014*, pp. 177–182, 2014.

[12] F. Wei, P. Vijayakumar, J. Shen, R. Zhang, and L. Li, "A provably secure password-based anonymous authentication scheme for wireless body area networks," *Comput. Electr. Eng.*, vol. 0, pp. 1–10, 2017.

[13] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, "Networking for Big Data: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 1, pp. 531–549, 2017.

[14] C. Ghaoui, *Usability Evaluation of Online Learning Programs*. IGI Global, 2002.

[15] A. Chuvakin, "'Honeynets: High value security data,'" *Netw. Secur.*, vol. 2003, no. 8, pp. 11–15, 2003.

[16] A. J. Fabiano and J. Qiu, *Post-stereotactic radiosurgery brain metastases: A*

review, vol. 59, no. 2. 2015.

[17] R. P. Reyes Ch and E. R. C. Fonseca, "How easy is to break password protection: A preliminary empirical study," in *2016 IEEE Ecuador Technical Chapters Meeting, ETCM 2016*, 2016, vol. 1, pp. 1–6.

[18] A. Juels and T. Ristenpart, "Honey encryption: Security beyond the brute-force bound," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8441 LNCS, P. Q. Nguyen and E. Oswald, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 293–310.

[19] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh, "Kamouflage: Loss-resistant password management," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6345 LNCS, D. Gritzalis, B. Preneel, and M. Theoharidou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 286–302.

[20] J. Bradley, *OS X Incident Response: Scripting and Analysis*. Syngress, 2016.

[21] P. Dunphy, *Usable, Secure and Deployable Graphical Passwords*, no. November. Paul Dunphy, 2012.

[22] K. K. Greene, J. Kelsey, and J. M. Franklin, "Measuring the Usability and Security of Permuted Passwords on Mobile Platforms," *Natl. Inst. Stand. Technol. Interag. Rep.*, vol. 8040, 2016.

[23] A. Vance, D. Eargle, K. Ouimet, and D. Straub, "Enhancing password security through interactive fear appeals: A web-based field experiment," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 2988–2997, 2013.

[24] W. Melicher *et al.*, "Usability and Security of Text Passwords on Mobile Devices," *Proc. 2016 CHI Conf. Hum. Factors Comput. Syst. - CHI '16*, pp. 527–539, 2016.

[25] P. Dourish, R. E. Grinter, J. D. De La Flor, and M. Joseph, "Security in the wild: User strategies for managing security as an everyday, practical problem," *Pers. Ubiquitous Comput.*, vol. 8, no. 6, pp. 391–401, 2004.

[26] C. Shen, T. Yu, H. Xu, G. Yang, and X. Guan, "User practice in password security: An empirical study of real-life passwords in the wild," *Comput. Secur.*, vol. 61, pp. 130–141, 2016.

[27] F. Kiefer, "Advancements in password-based cryptography." University of Surrey, 2016.

[28] M. Hussain, A. W. Abdul Wahab, I. Batool, and M. Arif, "Secure password transmission for web applications over internet using cryptography and image steganography," *Int. J. Secur. its Appl.*, vol. 9, no. 2, pp. 179–188, 2015.

[29] O. León, J. Hernández-Serrano, and M. Soriano, "Securing cognitive radio networks," *Int. J. Commun. Syst.*, vol. 23, no. 5, pp. 633–652, 2010.

[30] Y. Choi, "Security Enhanced Anonymous Multi-Server Authenticated Key Agreement Scheme using Smart Card and Biometrics," *Iacr*, vol. 2014, 2014.

[31] N. Fleischhacker, M. Manulis, and A. Azodi, "A Modular Framework for Multi-Factor Authentication and Key Exchange," in *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, vol. 8893, pp. 190–214.

[32] S. Cherry, *Secrets and lies: digital security in a networked world [Books]*, vol. 37, no. 10. John Wiley & Sons, 2000.

[33] S. Dodier-Lazaro, R. Abu-Salma, I. Becker, and M. Sasse, "From paternalistic to user-centred security: Putting Users First with Value-Sensitive Design," in *In: Position Papers. Values In Computing. (2017)*, 2017.

[34] T. Arce, P. E. Román, J. Velásquez, and V. Parada, "Identifying web sessions with simulated annealing," *Expert Syst. Appl.*, vol. 41, no. 4 PART 2, pp. 1593–1600, 2014.

[35] O. B. Al-Khurafi and M. A. Al-Ahmad, "Survey of Web Application Vulnerability Attacks," in *Proceedings - 2015 4th International Conference on Advanced Computer Science Applications and Technologies, ACSAT 2015*, 2016, pp. 154–158.

[36] D. Mitropoulos, P. Louridas, M. Polychronakis, and A. D. Keromytis, "Defending Against Web Application Attacks: Approaches, Challenges and Implications," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99. pp. 1–1, 2017.

[37] K. Gutzmann, "Access control and session management in the HTTP environment," *IEEE Internet Comput.*, vol. 5, no. 1, pp. 26–35, Jan. 2001.

[38] M. Bugliesi, S. Calzavara, and R. Focardi, "Formal methods for Web security," *J. Log. Algebr. Methods Program.*, vol. 1, pp. 1–17, 2016.

[39] A. Kołakowska, "Usefulness of keystroke dynamics features in user authentication and emotion recognition," in *Advances in Intelligent Systems and Computing*, vol. 551, Springer, 2018, pp. 42–52.

[40] P. Bours and S. Mondal, "Continuous Authentication with Keystroke Dynamics," *Nor. Inf. Secur. Lab. NISlab*, vol. 2, pp. 41–58, 2015.

[41] M. Karnan, M. Akila, and N. Krishnaraj, "Biometric personal authentication using keystroke dynamics: A review," *Appl. Soft Comput. J.*, vol. 11, no. 2, pp. 1565–1573, 2011.

[42] D. Ferbrache, "Passwords are broken – the future shape of biometrics," *Biometric Technol. Today*, vol. 2016, no. 3, pp. 5–7, 2016.

[43] B. Grawemeyer and H. Johnson, "Using and managing multiple passwords: A week to a view," *Interact. Comput.*, vol. 23, no. 3, pp. 256–267, 2011.

[44] S. Ruoti, J. Andersen, and K. E. Seamons, "Strengthening Password-based Authentication.," in *Way@ Soups*, 2016.

[45] Y. Yang, J. Sun, and L. Guo, "PersonaIA: A Lightweight Implicit Authentication System based on Customized User Behavior Selection," *IEEE Trans. Dependable Secur. Comput.*, pp. 1–1, 2016.

[46] S. Prabhu and V. Shah, "Authentication using session based passwords," *Procedia Comput. Sci.*, vol. 45, no. C, pp. 460–464, 2015.

[47] D. Mishra, A. K. Das, A. Chaturvedi, and S. Mukhopadhyay, "A secure password-based authentication and key agreement scheme using smart cards.,"

J. Inf. Sec. Appl., vol. 23, pp. 28–43, 2015.

[48] Y. Hedin and E. Moradian, "Security in multi-agent systems," *Procedia Comput. Sci.*, vol. 60, no. 1, pp. 1604–1612, 2015.

[49] A. Jesudoss and N. P. Subramaniam, "A Survey on Authentication Attacks and Countermeasures," *Indian J. Comput. Sci. Eng.*, vol. 5, no. 2, pp. 71–77, 2014.

[50] X. Yang, Z. Shen, X. Hu, and W. Hu, "Chaotic Encryption Algorithm Against Chosen-Plaintext Attacks in Optical OFDM Transmission," *IEEE Photonics Technol. Lett.*, vol. 28, no. 22, pp. 2499–2502, 2016.

[51] J.-S. Cho, Y.-S. Jeong, and S. O. Park, "Consideration on the brute-force attack cost and retrieval cost: A hash-based radio-frequency identification (RFID) tag mutual authentication protocol," *Comput. Math. with Appl.*, vol. 69, no. 1, pp. 58–65, 2015.

[52] E. I. Tatli, "Cracking more password hashes with patterns," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 8, pp. 1656–1665, 2015.

[53] D. Vishwakarma and C. E. V. Madhavan, "Efficient dictionary for salted password analysis," in *IEEE CONECCT 2014 - 2014 IEEE International Conference on Electronics, Computing and Communication Technologies*, 2014, pp. 1–6.

[54] J. Jose, T. T. Tomy, V. Karunakaran, V. Anjali Krishna, A. Varkey, and C. A. Nisha, "Securing passwords from dictionary attack with character-tree," in *Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016*, 2016, pp. 2301–2307.

[55] H. Yang *et al.*, "TapLock: Exploit finger tap events for enhancing attack resilience of smartphone passwords," in *IEEE International Conference on Communications*, 2015, vol. 2015–Septe, pp. 7139–7144.

[56] I. Uusitalo, J. M. Catot, and R. Loureiro, "Phishing and countermeasures in Spanish online banking," in *Proceedings - 2009 3rd International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2009*, 2009, pp. 167–172.

[57] L. Wu, X. Du, and J. Wu, "Effective Defense Schemes for Phishing Attacks on Mobile Computing Platforms," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6678–6691, 2016.

[58] L. Zhang-Kennedy, S. Chiasson, and R. Biddle, "Password advice shouldn't be boring: Visualizing password guessing attacks," in *eCrime Researchers Summit, eCrime*, 2013, pp. 1–11.

[59] X. Yu, Z. Wang, Y. Li, L. Li, W. T. Zhu, and L. Song, "EvoPass: Evolvable graphical password against shoulder-surfing attacks," *Comput. Secur.*, vol. 70, pp. 179–198, 2017.

[60] R. B. Varne and R. V. Mane, "CAPTCHA: A robust approach to resist online password guessing attacks," in *Proceedings - 2014 IEEE International Conference on Advances in Communication and Computing Technologies, ICACACT 2014*, 2014, pp. 1–6.

[61] C. Adams, G. V. Jourdan, J. P. Levac, and F. Prevost, "Lightweight protection

against brute force login attacks on web applications," in *PST 2010: 2010 8th International Conference on Privacy, Security and Trust*, 2010, pp. 181–188.

[62] F. Mouton, L. Leenen, and H. S. Venter, "Social engineering attack examples, templates and scenarios," *Comput. Secur.*, vol. 59, pp. 186–209, 2016.

[63] M. Junger, L. Montoya, and F. J. Overink, "Priming and warnings are not effective to prevent social engineering attacks," *Comput. Human Behav.*, vol. 66, pp. 75–87, 2017.

[64] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, "Advanced social engineering attacks," *J. Inf. Secur. Appl.*, vol. 22, pp. 113–122, 2015.

[65] R. Heartfield, G. Loukas, and D. Gan, "You Are Probably Not the Weakest Link: Towards Practical Prediction of Susceptibility to Semantic Social Engineering Attacks," *IEEE Access*, vol. 4, pp. 6910–6928, 2016.

[66] M. Masdari and S. Ahmadzadeh, "A survey and taxonomy of the authentication schemes in Telecare Medicine Information Systems," *J. Netw. Comput. Appl.*, vol. 87, pp. 1–19, 2017.

[67] A. Aldairi and L. Tawalbeh, "Cyber Security Attacks on Smart Cities and Associated Mobile Technologies," *Procedia Comput. Sci.*, vol. 109, no. 2016, pp. 1086–1091, 2017.

[68] J. Zheng and D. Wang, "Cryptanalysis and improvement of a SIP authentication scheme," *Proc. 2nd Int. Conf. Inf. Technol. Electron. Commer. ICITEC 2014*, pp. 199–203, 2014.

[69] M. Ibrahim Salim and T. A. Razak, "A study on IDS for preventing denial of service attack using outliers techniques," in *Proceedings of 2nd IEEE International Conference on Engineering and Technology, ICETECH 2016*, 2016, pp. 768–775.

[70] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 447–456, 2014.

[71] H. Kumar *et al.*, "Rainbow table to crack password using MD5 hashing algorithm," in *2013 IEEE Conference on Information and Communication Technologies, ICT 2013*, 2013, pp. 433–439.

[72] H. M. Ying and N. Kunihiro, "Decryption of frequent password hashes in rainbow tables," in *Proceedings - 2016 4th International Symposium on Computing and Networking, CANDAR 2016*, 2017, pp. 655–661.

[73] K. Skračić, P. Pale, and Z. Kostanjčar, "Authentication approach using one-time challenge generation based on user behavior patterns captured in transactional data sets," *Comput. Secur.*, vol. 67, pp. 107–121, 2017.

[74] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung, "Fourth-factor authentication," in *Proceedings of the 13th ACM conference on Computer and communications security - CCS '06*, 2006, p. 168.

[75] D. Palacios, T. España, F. Almenárez, and D. Díaz-sánchez, "Comparing Password Management Software," no. October, 2016.

[76] M. Lim, A. B. J. Teoh, and J. Kim, "Biometric feature-type transformation: Making templates compatible for secret protection," *IEEE Signal Process.*

Mag., vol. 32, no. 5, pp. 77–87, 2015.

[77] A. Abdellaoui, Y. I. Khamlichi, and H. Chaoui, "A Novel Strong Password Generator for Improving Cloud Authentication," *Procedia Comput. Sci.*, vol. 85, pp. 293–300, 2016.

[78] A. Sahu, S. B., & Singh, "Survey on Various Techniques of User Authentication and Graphical Password," *Int. J. Comput. Trends Technol.*, vol. 16, no. 3, pp. 98–102, 2014.

[79] S. Mount and R. Newman, "Energy-Efficient Brute Force Password Cracking," in *Proceedings - 2015 European Intelligence and Security Informatics Conference, EISIC 2015*, 2016, p. 189.

[80] K. Renaud, D. Kennes, J. Van Niekerk, and J. Maguire, "SNIPPET: Genuine knowledge-based authentication," in *Information Security for South Africa, 2013*, 2013, pp. 1–8.

[81] T. Acar, M. Belenkiy, and A. Küpçü, "Single password authentication," *Comput. Networks*, vol. 57, no. 13, pp. 2597–2614, 2013.

[82] L. Zhang, C. Tan, and F. Yu, "An Improved Rainbow Table Attack for Long Passwords," *Procedia Comput. Sci.*, vol. 107, pp. 47–52, 2017.

[83] O. Blazy, C. Chevalier, and D. Vergnaud, "Mitigating server breaches in password-based authentication: Secure and efficient solutions," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9610, K. Sako, Ed. Cham: Springer International Publishing, 2016, pp. 3–18.

[84] C. M. Chen, L. Xu, W. Fang, and T. Y. Wu, "A three-party password authenticated key exchange protocol resistant to stolen smart card attacks," in *Smart Innovation, Systems and Technologies*, vol. 63, J.-S. Pan, P.-W. Tsai, and H.-C. Huang, Eds. Cham: Springer International Publishing, 2017, pp. 331–336.

[85] R. Chen, Y. Mu, G. Yang, W. Susilo, and F. Guo, "Strongly leakage-resilient authenticated key exchange," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9610, pp. 19–36.

[86] Q. Yan, J. Han, Y. Li, J. Zhou, and R. H. Deng, "Leakage-resilient password entry: Challenges, design, and evaluation," *Comput. Secur.*, vol. 48, pp. 196–211, 2015.

[87] N. Chakraborty and S. Mondal, "Tag Digit Based Honeypot to Detect Shoulder Surfing Attack," in *International Symposium Security in Computing and Communications (SSCC)*, 2014, pp. 101–110.

[88] M. Boopathi and M. Aramudhan, "Secure server-server communication for dual stage biometrics - based password authentication scheme," *Alexandria Eng. J.*, 2016.

[89] N. Z. Gong and D. Wang, "On the security of trustee-based social authentications," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 8, pp. 1251–1263, Aug. 2014.

[90] S. Houshmand, S. Aggarwal, and R. Flood, "Next Gen PCFG Password

Cracking," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 8, pp. 1776–1791, 2015.

[91] D. Boneh, H. Corrigan-Gibbs, and S. Schechter, "Balloon hashing: A memory-hard function providing provable protection against sequential attacks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10031 LNCS, J. H. Cheon and T. Takagi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 220–248.

[92] A. Biryukov, D. Dinu, and D. Khovratovich, "Argon2: New generation of memory-hard functions for password hashing and other applications," in *Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016*, 2016, pp. 292–302.

[93] M. Alizadeh, S. Abolfazli, M. Zamani, S. Baaaharun, and K. Sakurai, "Authentication in mobile cloud computing: A survey," *J. Netw. Comput. Appl.*, vol. 61, pp. 59–80, 2016.

[94] Y. Wang, I.-R. Chen, and D.-C. Wang, "A Survey of Mobile Cloud Computing Applications: Perspectives and Challenges," *Wirel. Pers. Commun.*, vol. 80, no. 4, pp. 1607–1623, Feb. 2015.

[95] K. K. Kumbhare and K. V. Warkar, "A Review on Noisy Password, Voiceprint Biometric and One-Time-Password," *Phys. Procedia*, vol. 78, pp. 382–386, 2016.

[96] D. Zhao and W. Luo, "One-time password authentication scheme based on the negative database," *Eng. Appl. Artif. Intell.*, vol. 62, pp. 396–404, 2017.

[97] A. Mitra, A. Kundu, M. Chattopadhyay, and S. Chattopadhyay, "A cost-efficient one time password-based authentication in cloud environment using equal length cellular automata," *J. Ind. Inf. Integr.*, vol. 5, pp. 17–25, 2017.

[98] H. Alsaiari, M. Papadaki, and P. S. Dowland, "A Review of Graphical Authentication utilising a Keypad Input Method," in *Proceedings of the Eighth Saudi Students Conference in the UK*, 2015, p. 359.

[99] R. Anbuvizhi and V. Balakumar, "Credit / Debit Card Transaction Survey Using Map Reduce in HDFS and Implementing Syferlock to Prevent Fraudulent," *Int. J. Comput. Sci. Netw. Secur.*, vol. 16, no. 11, pp. 106–110, 2016.

[100] A. Danish, L. Sharma, H. Varshney, and A. M. Khan, "Alignment based graphical password authentication system," in *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on*, 2016, pp. 2950–2954.

[101] H. M. Aljahdali and R. Poet, "Challenge set designs and user guidelines for usable and secured recognition-based graphical passwords," in *Proceedings - 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2014*, 2015, pp. 973–982.

[102] C. Katsini, M. Belk, C. Fidas, N. Avouris, and G. Samaras, "Security and Usability in Knowledge-based User Authentication," *Proc. 20th Pan-Hellenic Conf. Informatics - PCI '16*, pp. 1–6, 2016.

[103] A. Sadovnik and T. Chen, “A visual dictionary attack on Picture Passwords,” in *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings*, 2013, pp. 4447–4451.

[104] A. Nayak and R. Bansode, “Analysis of Knowledge Based Authentication System Using Persuasive Cued Click Points,” *Procedia Comput. Sci.*, vol. 79, pp. 553–560, 2016.

[105] A. Jain, R. Khetan, K. Dubey, and H. Rambade, “Color Shuffling Password Based Authentication,” *Int. J.*, vol. 10528, 2017.

[106] A. Bianchi, I. Oakley, and H. Kim, “PassBYOP: Bring Your Own Picture for Securing Graphical Passwords,” *IEEE Trans. Human-Machine Syst.*, vol. 46, no. 3, pp. 380–389, Jun. 2016.

[107] S. Bhatt and T. Santhanam, “Keystroke dynamics for biometric authentication-A survey,” in *Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, PRIME 2013*, 2013, pp. 17–23.

[108] K. Halunen, J. Häikiö, and V. Vallivaara, “Evaluation of user authentication methods in the gadget-free world,” *Pervasive Mob. Comput.*, vol. 40, pp. 220–241, 2017.

[109] A. F. Abate, M. Nappi, and S. Ricciardi, “I-Am: Implicitly Authenticate Me Person Authentication on Mobile Devices Through Ear Shape and Arm Gesture,” *IEEE Trans. Syst. Man, Cybern. Syst.*, 2017.

[110] I. Traor??, Y. Nakkabi, S. Saad, B. Sayed, J. D. Ardigo, and P. M. De Faria Quinan, “Ensuring online exam integrity through continuous biometric authentication,” in *Information Security Practices: Emerging Threats and Perspectives*, I. Traore, A. Awad, and I. Woungang, Eds. Cham: Springer International Publishing, 2017, pp. 73–81.

[111] P. Gupta, S. Srivastava, and P. Gupta, “An accurate infrared hand geometry and vein pattern based authentication system,” *Knowledge-Based Syst.*, vol. 103, no. Supplement C, pp. 143–155, 2016.

[112] C. Yuan, X. Sun, and R. Lv, “Fingerprint liveness detection based on multi-scale LPQ and PCA,” *China Commun.*, vol. 13, no. 7, pp. 60–65, 2016.

[113] I. Bouchrika, “Evidence evaluation of gait biometrics for forensic investigation,” in *Intelligent Systems Reference Library*, vol. 115, A. E. Hassanien, M. Mostafa Fouad, A. A. Manaf, M. Zamani, R. Ahmad, and J. Kacprzyk, Eds. Cham: Springer International Publishing, 2017, pp. 307–326.

[114] M. Pleva, P. Bours, D. Hladek, and J. Juhar, “Using current biometrics technologies for authentication in e-learning assessment,” in *ICETA 2016 - 14th IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings*, 2016, pp. 269–274.

[115] S. Thavalengal and P. Corcoran, “User Authentication on Smartphones,” *IEEE Consum. Electron. Mag.*, vol. 5, no. April, pp. 87–93, 2016.

[116] A. Bhuiyan, A. Hussain, A. Mian, T. Y. Wong, K. Ramamohanarao, and Y. Kanagasingam, “Biometric authentication system using retinal vessel pattern and geometric hashing,” *IET Biometrics*, vol. 6, no. 2, pp. 79–88, 2017.

[117] M. Diaz, A. Fischer, M. A. Ferrer, and R. Plamondon, "Dynamic Signature Verification System Based on One Real Signature," *IEEE Trans. Cybern.*, pp. 1–12, 2016.

[118] A. K. Jain, K. Nandakumar, and A. Ross, "50 years of biometric research: Accomplishments, challenges, and opportunities," *Pattern Recognit. Lett.*, vol. 79, no. C, pp. 80–105, 2016.

[119] M. Gomez-Barrero, C. Rathgeb, U. Scherhag, and C. Busch, "Is your biometric system robust to morphing attaeks?," in *Proceedings - 2017 5th International Workshop on Biometrics and Forensics, IWBF 2017*, 2017, pp. 1–6.

[120] S. Bhilare, V. Kanhangad, and N. Chaudhari, "A study on vulnerability and presentation attack detection in palmprint verification system," *Pattern Anal. Appl.*, pp. 1–14, 2017.

[121] T. Ring, "Spoofing: are the hackers beating biometrics?," *Biometric Technol. Today*, vol. 2015, no. 7, pp. 5–9, 2015.

[122] U. Bakshi, R. Singhal, and M. Malhotra, "Biometric Technology: A Look and Survey at Face Recogntion." IJESI, 2014.

[123] M. B. Bondada and S. M. S. Bhanu, "Analyzing user behavior using keystroke dynamics to protect cloud from malicious insiders," in *2014 IEEE International Conference on Cloud Computing in Emerging Markets, CCEM 2014*, 2015, pp. 1–8.

[124] Neha and K. Chatterjee, "Efficient Remote User Authentication Technique for Internet Based Applications Using Keystroke Dynamics," in *Smart Trends in Information Technology and Computer Communications: First International Conference, SmartCom 2016, Jaipur, India, August 6--7, 2016, Revised Selected Papers*, A. Unal, M. Nayak, D. K. Mishra, D. Singh, and A. Joshi, Eds. Singapore: Springer Singapore, 2016, pp. 881–888.

[125] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Futur. Gener. Comput. Syst.*, vol. 16, no. 4, pp. 351–359, 2000.

[126] S. Mondal and P. Bours, "Person Identification by Keystroke Dynamics Using Pairwise User Coupling," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 6, pp. 1319–1329, 2017.

[127] J. Ho and D. K. Kang, "Mini-batch bagging and attribute ranking for accurate user authentication in keystroke dynamics," *Pattern Recognit.*, vol. 70, pp. 139–151, 2017.

[128] A. Goodkind, D. G. Brizan, and A. Rosenberg, "Utilizing overt and latent linguistic structure to improve keystroke-based authentication," *Image Vis. Comput.*, vol. 58, pp. 230–238, 2017.

[129] J. Huang, D. Hou, S. Schuckers, and S. Upadhyaya, "Effects of text filtering on authentication performance of keystroke biometrics," in *8th IEEE International Workshop on Information Forensics and Security, WIFS 2016*, 2017, pp. 1–6.

[130] Z. Ba and K. Ren, "Addressing Smartphone-Based Multi-factor Authentication via Hardware-Rooted Technologies," in *Proceedings - International Conference on Distributed Computing Systems*, 2017, pp. 1910–1914.

[131] D. Dasgupta, A. Roy, and A. Nag, "Toward the design of adaptive selection

strategies for multi-factor authentication," *Comput. Secur.*, vol. 63, pp. 85–116, 2016.

[132] M. Zhang, J. Zhang, and W. Tan, "Remote three-factor authentication protocol with strong robustness for multi-server environment," *China Commun.*, vol. 14, no. 6, pp. 126–136, 2017.

[133] H. Xiong, J. Tao, and C. Yuan, "Enabling Telecare Medical Information Systems with Strong Authentication and Anonymity," *IEEE Access*, vol. 5, no. 8, pp. 5648–5661, 2017.

[134] Y. Liu, Q. Zhong, L. Chang, Z. Xia, D. He, and C. Cheng, "A secure data backup scheme using multi-factor authentication," *IET Inf. Secur.*, 2016.

[135] M. A. Nematollahi, H. Gamboa-Rosales, F. J. Martinez-Ruiz, J. I. De la Rosa-Vargas, S. A. R. Al-Haddad, and M. Esmaeilpour, "Multi-factor authentication model based on multipurpose speech watermarking and online speaker recognition," *Multimed. Tools Appl.*, vol. 76, no. 5, pp. 7251–7281, 2017.

[136] E. Kheirkhah, S. M. P. Amin, H. A. Sistani, and H. Acharya, "An experimental study of SSH attacks by using Honeypot Decoys," *Indian J. Sci. Technol.*, vol. 6, no. 12, pp. 5567–5578, 2013.

[137] C. Rong and G. Yang, "Honeypots in Blackhat Mode and its Implications," in *Parallel and Distributed Computing, Applications and Technologies, PDCAT Proceedings*, 2003, pp. 185–188.

[138] A. V. Arzhakov, S. S. Troitskiy, N. P. Vasilyev, and D. S. Silnov, "Development and implementation a method of detecting an attacker with use of HTTP network protocol," in *Proceedings of the 2017 IEEE Russia Section Young Researchers in Electrical and Electronic Engineering Conference, ElConRus 2017*, 2017, pp. 100–104.

[139] S. Kumar, R. Sehgal, and J. S. Bhatia, "Hybrid honeypot framework for malware collection and analysis," in *2012 IEEE 7th International Conference on Industrial and Information Systems, ICIIS 2012*, 2012, pp. 1–5.

[140] L. Vokorokos, P. Fanfara, J. Radusovsky, and P. Poor, "Sophisticated Honeypot mechanism - The autonomous hybrid solution for enhancing computer system security," in *SAMI 2013 - IEEE 11th International Symposium on Applied Machine Intelligence and Informatics, Proceedings*, 2013, pp. 41–46.

[141] S. Litchfield, D. Formby, J. Rogers, S. Meliopoulos, and R. Beyah, "Rethinking the Honeypot for Cyber-Physical Systems," *IEEE Internet Comput.*, vol. 20, no. 5, pp. 9–17, 2016.

[142] D. K. Rahmatullah, S. M. Nasution, and F. Azmi, "Implementation of low interaction web server honeypot using cubieboard," in *ICCEREC 2016 - International Conference on Control, Electronics, Renewable Energy, and Communications 2016, Conference Proceedings*, 2017, pp. 127–131.

[143] M. Bombardieri, S. Castanò, F. Curcio, A. Furfaro, and H. D. Karatza, "Honeypot-powered malware reverse engineering," in *Proceedings - 2016 IEEE International Conference on Cloud Engineering Workshops, IC2EW 2016*, 2016, pp. 65–69.

[144] H. Gjermundrod and I. Dionysiou, "CloudHoneyCY - An Integrated Honeypot Framework for Cloud Infrastructures," in *Proceedings - 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing, UCC 2015*, 2015, pp. 630–635.

[145] M. Akiyama, T. Yagi, T. Yada, T. Mori, and Y. Kadobayashi, "Analyzing the ecosystem of malicious URL redirection through longitudinal observation from honeypots," *Comput. Secur.*, vol. 69, pp. 155–173, 2017.

[146] H. J. Mun and K. H. Han, "Blackhole attack: user identity and password seize attack using honeypot," *J. Comput. Virol. Hacking Tech.*, vol. 12, no. 3, pp. 185–190, 2016.

[147] P. R. L. R. M. I. T. Dr. Ari Juels RSA Professor Ronald L. Rivest MIT. Dr. Ari Juels RSA, "For Stronger Password Security, Try a Spoonful of Honeywords," 2013.

[148] E. Martiri, B. Yang, and C. Busch, "Protected honey face templates," in *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft fur Informatik (GI)*, 2015, vol. P-245, pp. 1–7.

[149] M. J. Bhole, "Honeywords : A New Approach For Enhancing Security Manisha Jagannath Bhole," *Int. Res. J. Eng. Technol.*, vol. 2, no. 8, pp. 1563–1566, 2015.

[150] A. S. G. Sujeeth Kumar, "Modelos animales de dolor neuropático," *Dolor*, vol. 31, no. 2, pp. 70–76, 2016.

[151] L. Zhang-Kennedy, S. Chiasson, and P. Van Oorschot, "Revisiting password rules: Facilitating human management of passwords," in *eCrime Researchers Summit, eCrime*, 2016, vol. 2016–June, pp. 81–90.

[152] S. Mondal and P. Bours, "A study on continuous authentication using a combination of keystroke and mouse biometrics," *Neurocomputing*, vol. 230, pp. 1–22, 2017.

[153] G. R. Haron, D. Maniam, L. Mat Nen, and N. I. Daud, "User behaviour and interactions for multimodal authentication," in *2016 14th Annual Conference on Privacy, Security and Trust, PST 2016*, 2016, pp. 309–316.

[154] S. Calzavara, R. Focardi, M. Squarcina, and M. Tempesta, "Surviving the Web," *ACM Comput. Surv.*, vol. 50, no. 1, pp. 1–34, 2017.

[155] S. Calzavara, R. Focardi, N. Grimm, and M. Maffei, "Micro-policies for web session security," in *Proceedings - IEEE Computer Security Foundations Symposium*, 2016, vol. 2016–Augus, pp. 179–193.

[156] G. Deepa and P. S. Thilagam, "Securing web applications from injection and logic vulnerabilities: Approaches and challenges," *Inf. Softw. Technol.*, vol. 74, no. Supplement C, pp. 160–180, 2016.

[157] M. Tadao, "Petri Nets: Properties, Analysis and Applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[158] M. M. Taha, T. A. Alhaj, A. E. Moktar, A. H. Salim, and S. M. Abdullah, "On password strength measurements: Password entropy and password quality," in *Proceedings - 2013 International Conference on Computer, Electrical and Electronics Engineering: "Research Makes a Difference", ICCEEE 2013*,

2013, pp. 497–501.

[159] M. L. Mazurek *et al.*, "Measuring password guessability for an entire university," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13*, 2013, pp. 173–186.

[160] B. Ur *et al.*, "Measuring real-world accuracies and biases in modeling password guessability," in *24th USENIX Security Symposium (USENIX Security 15)*, 2015, pp. 463–481.

[161] W. Ma, J. Campbell, D. Tran, and D. Kleeman, "Password entropy and password quality," in *Proceedings - 2010 4th International Conference on Network and System Security, NSS 2010*, 2010, pp. 583–587.

[162] M. Tech, "Web Application:(with) HoneyWords and HoneyEncryption," vol. 4, no. 2, p. 2313, 2013.

[163] A. Juels and R. L. Rivest, "Honeywords : Making Password-Cracking Detectable," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 145–159.

[164] L. Catuogno, A. Castiglione, and F. Palmieri, "A honeypot system with honeyword-driven fake interactive sessions," in *Proceedings of the 2015 International Conference on High Performance Computing and Simulation, HPCS 2015*, 2015, pp. 187–194.

[165] N. Chakraborty and S. Mondal, "A New Storage Optimized Honeyword Generation Approach for Enhancing Security and Usability," *arXiv Prepr. arXiv1509.06094*, p. 8, 2015.

[166] Y. Li, H. Wang, and K. Sun, "A study of personal information in human-chosen passwords and its security implications," in *Proceedings - IEEE INFOCOM*, 2016, vol. 2016–July, pp. 1–9.

[167] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proceedings - IEEE Symposium on Security and Privacy*, 2012, pp. 538–552.

[168] Riva11, "The Top 500 Worst Passwords of All Time," *Symantec*, 2010. [Online]. Available: http://www.symantec.com/connect/blogs/top-500-worst-passwords-all-time.

[169] I. Erguler, "Achieving Flatness: Selecting the Honeywords from Existing User Passwords," *IEEE Trans. Dependable Secur. Comput.*, vol. 13, no. 2, pp. 284–295, 2016.

Publications:

1- O. Z. Akif and H. S. Al-Raweshidy and G. J. Rodgers, "Protecting a Sensitive Dataset Using Time Based Password in Big Data," *SAI Intelligent Systems Conference (IntelliSys), IEEE 2017*, London.

2- O. Z. Akif and H. S. Al-Raweshidy and G. J. Rodgers, " Achieving Flatness: Honeywords Generation Method for Passwords Based on User Behaviours " *FTC Future Technology Conference, IEEE 2017*, Vancouver, Canada.