
Hardware and User Profiling for Multi-factor Authentication

PhD Thesis

Adeeb Ali Alnajjar

This thesis is submitted in partial fulfillment of the requirements for the
Doctor of Philosophy

Software Technology Research Laboratory

Faculty of Technology
De Montfort University

June 2013

Declaration of Authorship

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy in computer security at Software Technology Research Laboratory (STRL), Faculty of Technology, De Montfort University, Leicester, United Kingdom. No part of the material described in this thesis has been submitted for the award of any other degree or qualification in this or any other university or college of advanced education.

DEDICATION

The author of this work would like to show his appreciation and thanks to the following people for their contributions and support during this thesis.

To my **Mother and father.**

To my **Wife.**

To my **children**

To my **Family.**

People who took the time to take part in my studies.

I hope that by obtaining my PhD I can draw a smile on their faces.

Thank you.

Abstract

Most software applications rely on the use of user-name and passwords to authenticate end users. This form of authentication, although used ubiquitously, is widely considered unreliable due to the users inability to keep them secret; passwords being prone to dictionary or rainbow-table attacks; as well as the ease with which social engineering techniques can obtain passwords.

This can be mitigated by combining a variety of different authentication mechanisms, for example biometric authentication such as fingerprint recognition or physical tokens such as smart cards. The resulting multi-factor authentication is typically stronger than any of the techniques used individually. However, it may still be expensive or prohibited to implement and more difficult to deploy due to additional accessories cost, e.g, finger print reader.

Multi-modal biometric systems are those which utilise or are capable of utilising, more than one physiological or behavioural characteristic for enrolment, verification, or identification. So, in this research we present a multi-factor authentication scheme that is based on the user's own hardware environment, e.g. laptop with fingerprint reader, thus avoiding the need of deploying tokens and readily available biometrics, e.g., user keystrokes. The aim is to improve the reliability of the authentication using a multi-factor approach without incurring additional cost or making the deployment of the solution overly complex.

The presented approach in this research uses unique sequential hardware information available from the user's environment to *profile* user behaviour. This approach improves upon password mechanisms by introducing a novel Hardware Authentication and User Profiling (HAUP) in form of Multi-Factor Authentication MFA that can be easily integrated into the traditional authentication methods. In addition, this approach observes the advantage of the correlation between user behaviour and hardware environment as an implicit verification identity procedure to discriminate user-name and password usage, in particular hardware environment by specific pattern. So, the proposed approach uses hardware information to profile the user's environment when user-name and password are typed as part of the log-in process. These Hardware Manufacture Serial Part Numbers (*HM-SPNs*) profiles are then correlated with the users behaviour, e.g., key-stroke behaviour that allows the system to profile user's behaviour dependent on their environment. As a result of this approach, the access control system can determine a particular level of trust for each user and base access control decisions on it in order to reduce potential identity fraud.

Keywords

Authentication, Profiling, Keystroke Recognition, User behaviour security, Multi-factor, Attack.

Acknowledgements

First of all, I would like to start by praising Allah (God) for all his bounties, blessings and for providing me with faith, patience and commitment to complete this research. Without him, none of this work would have been possible. Then, I would like to thank my supervisor Dr. Helge Janicke for his expert guidance and help in providing a stimulating and remarkably hassle free environment in which this research could be pursued. I am grateful for his careful reading and constructive comments on our joint paper.

Also I would like to express my sincere gratitude to Software Technology Research Laboratory staff, at De Montfort University with a special mention for the technical director, Professor Hussein Zedan, for his guidance, persistent support, and encouragement throughout this research. Moreover, they gave me motivation for starting my new topic and freedom in my research interests.

I wish to thank all researchers and colleagues of the STRL. During this work, I have collaborated with many colleagues for whom I have great regard. Finally, I would like to mention my beloved children Abdullah, Jomanah and Raghad who have given me happiness during difficult periods of my study.

Thank you all.

Leicester, United Kingdom /2013

Adeeb Ali Alnajjar

Publications

1- Adeeb Alnajjar and Helge Janicke “Multi-Factor Authentication Using Hardware Information and User Profiling Techniques”, in HAISA 2012 : Sixth International Symposium on Human Aspects of Information Security and Assurance 2012, Crete, Greece.

2- Adeeb Alnajjar “New Profiling Technique to Authenticate Internet And Computer Network Users”, in the 5th Saudi International Conference SIC05, June 2011, University of Warwick, Coventry, United Kingdom.

3- Adeeb Alnajjar “Using Hardware Information in Profiling Users Activity for Attack Tracking”, in the 4th Saudi International Conference SIC04, July 2010, Manchester university, Manchester, United Kingdom.

List of Abbreviation

MFA : Multi-Factor Authentication

HAUP : Hardware Authentication and User behaviour

HMSPNs : Hardware Manufacture Serial Part Numbers

eID : Electronic Identification

SID : Super Identity

PINs : Personal Identification Numbers

OTP : One Time Password

HW : Hardware

Bh : Behaviour

MAC : Media Access Control

BIOS : Basic Input and Output System

HDD : Hard Disc Drive

c_u : Previous user hardware configuration.

\bar{c}_u : Current user hardware configuration.

b_u : Previous user pattern.

\bar{b}_u : Current user behaviour.

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	v
Publications	vi
List of Abbreviation	vii
List of Figures	xiii
List of Tables	xvi
1 Introduction	1
1.1 Background	2
1.2 Problem Statement	3
1.3 Research Aim and Objectives	4
1.4 Research Questions	5
1.5 Scope of the Research	7
1.6 Research Methodology	8

1.7	Contributions	10
1.8	Success Criteria	11
1.9	Thesis Structure	12
2	Literature Review	18
2.1	Authentication in Access Control	20
2.1.1	The Limitations in Authentication Factors and Approaches	22
2.1.2	Single Factor Authentication	23
2.1.3	Multi-Factor Authentication	25
2.2	User Profiling	30
2.2.1	Profiling Services	32
2.2.2	Detecting User Behaviour	33
2.2.3	Cookies in Profiling Users	35
2.2.4	Keystroke and Profiling Users	36
2.3	Hardware Information Overview	40
2.3.1	MAC address	41
2.3.2	Storage Media Numbers	43
2.3.3	Motherboard Serial Numbers	44
2.4	Using HW Information in Authentication	45
2.5	Neural Network Recognition	52
2.5.1	Neural Network Analysis	52
2.5.2	Neural Network for Profiling	54
2.5.3	Neural Network for Security	55
2.5.4	Neural Networks and Intrusion Detection Systems	56
2.5.5	Neural Network Anomaly Detection	57
2.5.6	Neural Network Misuse Detection	57
2.6	Summary	58

3	Methodology	61
3.1	Method	63
3.2	HW Authentication in HW Life Cycle	65
3.3	Authentication HW Analysis	69
3.4	Design HAUP Framework	73
3.5	Analysing Hardware Authentication and User Behaviour	75
3.6	HAUP Approach Analysis	78
3.7	HW Information Contribution in HAUP Approach	80
3.8	HAUP Approach Analysis Requirements	80
3.9	HAUP Success Criteria	81
3.10	Summary	82
4	Hardware Authentication and User Behaviour Framework	84
4.1	HAUP Framework	85
4.2	Main HAUP System Components and Architecture	88
4.2.1	HAUP Recognizer	91
4.2.2	Client Behaviour recognizer	94
4.3	HAUP Authentication Process	96
4.4	Summary	100
5	Mathematical Model of HAUP	101
5.1	Formal HAUP Analysis	102
5.1.1	Domains	102
5.1.2	Hardware configuration	104
5.1.3	Sequential Log-in Context	105
5.2	Illustrative Example	106
5.2.1	Lmitations	109

5.3	Behaviour Trust Based on Hardware Information	110
5.3.1	Back Propagation Algorithm	111
5.4	Summary	115
6	Implementation	117
6.1	Hardware Authentication Scenario in Access Control	118
6.2	System Procedures	120
6.3	HAUP Analysis	123
6.3.1	Sequence Diagram of Behaviour Modelling	124
6.4	System Interaction	127
6.5	Summary	132
7	Set of Experiments and Evaluation	134
7.1	Proposed Set of Experiments and Evaluation Criteria	136
7.2	Profiling User Behaviour in Typing Log-in Keys Using The Same HW	138
7.3	Set of Experiments Using HAUP Approach . . .	140
7.3.1	One User Uses The Same Hardware . . .	141
7.3.2	One User Using Two Different Sets Of Hardware	143
7.3.3	Two Users Using the Same Password and Hardware	144
7.4	The Relation Between Log-in Time and Hardware Usage	145
7.5	Priority Classes Threshold	146
7.5.1	Overall Level of Trust	150
7.6	Neural Network Analysis in Matlab	152

7.6.1	Using Neural Network to Compare Between Users Behaviour When Two different HW are Used	156
7.6.2	Neural Network Analysis Experiment For Group of Users	162
7.7	Summary	174
8	Conclusion	177
8.1	Achievements	178
8.2	Contribution	180
8.3	Success Criteria Revisited	182
8.4	Limitations	183
8.4.1	Ways in which the solution might fail . .	184
8.5	Future Work	185

List of Figures

2.1	Network Layers and Security Service [1]	20
2.2	Authentication Techniques [2]	29
3.1	HW and Users Bh Life Cycle	67
3.2	HW Method for HW Approach	69
3.3	HW Method in User Authenticate	73
3.4	Framework Overview for Designing HAUP Aspect	74
3.5	Main Authentication Factors	77
3.6	Overview of Dependencies of HW Authentica- tion Factors	78
4.1	HAUP framework	86
4.2	High Level HAUP Architecture	89
4.3	Details of HAUP System Architecture	91
4.4	HW History During the Day	92
4.5	client's Keystroke Patterns in Particular Hardware	95
4.6	Profiling User's Keystroke Behaviour	96
4.7	General Overview of HAUP Flow Diagram be- tween System Components	98
5.1	How Using HW Information in Trust User	109
5.2	neural network procedures and hidden layer. . .	114
6.1	System Scenario	120

6.2	Diagram of HAUP Procedures	123
6.3	Sequence Diagram of Using HAUP First Time	125
6.4	Sequence Diagram of using hardware After First Time	126
6.5	Keystroke Biometric Behaviour Capture	128
6.6	“MAC Address” collection from Client HW	129
6.7	HAUP Profiling Database	130
6.8	HAUP Encrypted Database	131
6.9	Plot Measure Code to Show Users Keystroke Behaviour and Pattern in a HW	132
7.1	User Behaviour in Typing Log-in “Password” Keys by Using The Same Hardware	139
7.2	User Pattern in Typing “Password” Using The Same Hardware	140
7.3	Hardware Usage and Profile Against Keystroke Pattern	142
7.4	Hardware Usage And Profile of The Delay in Keystroke Behaviour	143
7.5	Hardware Profiling and Recognizing User Behaviour Against One User Using Two Different Hardware	144
7.6	HW Usage and Profile Against Two Users Use Same Log-in Information “Password” in One Hardware	145
7.7	Time “Signature” in Using a Particular Hardware	146
7.8	Priority Classes in HAUP Factors	149
7.9	Overall Level of Trust	151
7.10	Users Keystroke Speed Analysis Using Two Different Hardware	155

7.11	Neural Network Training Performance From Two Users Patterns Using the Same Hardware	157
7.12	Analysis of User Behaviour When Using the HAUP Prototype and Neural Network With Respect to a Particular Hardware	159
7.13	Comparison Between Users Keystroke Analysis When Using Two Different HW Devices and The Same Password Authentication Keys	161
7.14	Neural Network Analysis Including Input, Output and Hidden Layers	163
7.15	Neural Network Analysis Two Hardware Recognizing user Patterns	169
7.16	Neural Network recognizes which Hardware is Used	170
7.17	User 4400773 Recognizing And Testing	171
7.18	User abdull Recognizing And Testing	173
8.1	Support Vector Machine recognizer	187

List of Tables

2.1	Common Network Cards “MAC address” Companies and their HW Part Numbers	42
2.2	“Media Storage” Companies and their HW Parts Numbers [3]	44
2.3	Some “BIOS” Manufacturers and their HW Part Numbers[4]	44
3.1	Profiling User’s HW	71
4.1	Keystroke Profiling Against HW Configuration .	95
7.1	Users Pattern Analysis by Neural Network When Same password Keys and twoHW are Used 3 layers	164
7.2	User Pattern Analysis by Neural Network When the Same PW Keys and two HW are Used 5 layers	166
7.3	Analysis of User Patterns Using Neural Networks Without Determining the Users Hardware and Using Three Layers	168
7.4	Analysing And Testing Users Pattern Using Neural Network	174

Chapter 1

Introduction

Objectives

-
1. Motivate the research.
 2. Clarify the assumption.
 3. Describe the contributions.
 4. Outline the thesis structure.
-

1.1 Background

Computer Security aims to provide Confidentiality, Integrity and Availability in Information Systems in order to serve the information technology user. A key component in Information Systems to ensure Confidentiality and Integrity is Access Control. Access Control determines if a user is permitted to access a specific system resource, and how this resource can be used. A key factor for making access control decisions is to establish the identity of the user making this request, a process that is referred to as Authentication.

There are various established mechanisms to authenticate users, but by far the most widely deployed is the authentication by username and password. However it is becoming clear that this particular method is inadequate and prone to various forms of attack [5]. For example, short passwords can be easily broken using brute force techniques due to the increased computation power of today's personal computers or rented services in the Cloud. Dependent on the type of password chosen by a user dictionary attacks or rainbow-table attacks (reviewed in Section 1.2) can be used to assume another user's identity with relative ease. Another very effective way of stealing someone's identity are social engineering attacks, that trick unsuspecting users to divulge sufficient personal information about themselves so that the attacker can assume their identity or simply guess

information that is required to reset their password information within the information system.

Authentication can rely on three factors: what a user knows e.g. “username/password”, what a user has e.g. debit card and what a user is e.g. fingerprint. Using a combination of these factors is often referred to as Multi Factor Authentication (MFA). Most MFA are difficult to deploy due to the cost or logistic reasons.

This thesis considers a MFA approach that is based on hardware (HW) and User Profiling paired with well established username and password mechanisms that overcomes some of the problems of traditional MFA approaches. The following sections outline the problem statement and research aims.

1.2 Problem Statement

Identity fraud is estimated to affect 1.8 million UK residents and having an annual cost the UK economy of 2.7 billion [6]. This type of fraud is mainly utilising authentication vulnerabilities in access control mechanism. For example, compromise the Internet service provider by spoofing using another user’s authentication keys. The widely used “username/password” authentication is considered unreliable due to users’ inability to keep passwords secret; in addition passwords are prone to

dictionary [7] or rainbow-table attacks [8] as well as the ease with which social engineering [9] techniques can obtain passwords. Moreover, the cost of additional authentication factors is an obstacle to the deployment of MFA. For example, using fingerprints in MFA cost £2 GBP for the cheapest fingerprint reader to be used with users' computers [10]. One of challenges faced in MFA involves selecting characteristics of a user's identity without additional cost or inconveniencing the user. In addition, MFA should protect the user identity from spoofing, and respect user privacy.

Based on this problem statement the following aims and objectives of this research are established.

1.3 Research Aim and Objectives

The resulting MFA approach should consider cost and impact on existing environments, whilst providing resistance against attacks.

To achieve the aim of this research, a HW and user behaviour profiling approach is developed for modelling dynamic user behaviour based on user's HW environment. This approach clarifies HW advantages in profiling a user behaviour in order to reduce potential identity fraud and provide the trust between the user and service providers. This trust focuses in

the grater successful log-in attempt using same HW is grate the level of trust. Theses trust aid the automated verification of actions against security policies. [11]. The main objectives to achieve the research aim are:

O.1. Show the feasibility of profiling techniques in MFA.

O.2. Select suitable characteristic for profiling.

O.3. Develop a computational model for authentication based on the selected characteristics and provides a mathematical model for profiling that establishes a level of trust in which authentication is based.

O.4. Develop authentication framework that supports the mathematical model for profiling the selected characteristics.

O.5. Create authentication prototype based on the selected characteristics for data collection and evaluation.

O.6. Evaluate the prototype based on the approach against profiling and authentication approaches.

1.4 Research Questions

The following questions are related to authentication and profiling user behaviour to support “username/password” mechanism. So, this research will discuss the following questions:

Q.1. How can HW information be used to profile the user? Chapters 2 and 3 demonstrate that some HW can be used to observe user activity. This will address objective O.1. by showing profiling technique in MFA.

Q.2. What HW information is suitable for profiling in the context of authentication? Chapter 3 demonstrates a profiling method using unique hardware manufacture serial part numbers. This will address objectives O.2. by providing the hardware characteristics to be used in profiling.

Q.3. Can profiling be combined with traditional “username and password” mechanisms? Chapters 3, 4 and 5 provide the ability of develop profiling by HW information. This will clarify the objectives O.1. and O.3. by providing hardware authentication framework.

Q.4. What characteristics can be collected? Can additional accessories do this? Chapter 3 and 7 investigate the use of accessories, such as fingerprint scanner, in combination with hardware parts to profile usage characteristics. This will concentrate on the objective O.4. by providing authentication factors and creating framework based on hardware authentication for profiling.

Q.5. What is the added cost of a profiling approach? How can costs to be avoided? Chapter 3 illustrates the developed

profiling cost compared with the cost of current profiling technologies. This will focus on the objective O.5. by implementing MFA approach and demonstrates the cost of the authentication factors.

Q.6. How much profile information needs to be available to improve authentication? Chapters 4 and 6 determine when a profile is reliable. This will address objective O.5 by analysing the authentication prototype and factors in authentication.

Q.7. What is the impact of multiple users using various devices? This is discussed in Chapter 7. This will address objective O.6. by providing on evaluation for this approach.

1.5 Scope of the Research

This work addresses identity fraud in traditional authentication approach that is username and password. In this research we look how servers authenticate the clients. Profiling user patterns has many techniques to recognize user activity. For example, using cookies by means of collecting information about their behaviour during authentication, whilst taking their HW context into account.

Other traditional MFA factors such as fingerprint or physical tokens such as smart cards are not considered because of

the cost of deploying these devices. In this research we focus on available user hardware to investigate profiling user characteristics.

1.6 Research Methodology

As in majority of computer science approaches the described research belongs to the constructive research field where the constructive refer to knowledge contribution being developed as new framework [12]. So, this research uses a constructive approach to analyse and explore problem then provide solution and develop new approach to solve it. [13]. Four main steps constitute the methodology proposed.

Step 1: Critical Literature Review

Background research is conducted with critical review using hard and digital resources. For example, using Google scholar search for E-books and focusing in latest related published papers. In addition, using libraries and focus on specific and related journal, conferences and symposium in order to expand knowledge in research scope. For example, Association for Computing Machinery (ACM), Springer and IEEE Security and Privacy Magazine. This step enhanced understanding of main factors and approaches in authentication including HW profiling to provide evidence for the research objectives O.1.,

O.2. and related to research questions Q.1., Q.2., Q.3. and Q.4.

Step 2: Hardware System Methodology

Focuses on designing a system architecture using UML [14] to capture the research objectives O.3. and O.4. and answer research questions Q.5. and Q.6. provides a formal specification of the authentication approach using a mathematical model to support the approach.

Step 3: Implementation

This step aims to implement HW Authentication and User Profiling (HAUP) prototype to be integrated into a MFA framework. This HW prototype is implemented by Java code which has virtual machine specification and has access to hardware information in order to illustrate the approach in this research. This step clarifies the profiling influence in authentication decisions and finds a mechanism to observe profile influence based on a trust-model. Finally, this step implements the system prototype and components in order to achieve the objectives O.5. and O.6. of this research and is related to research question Q.5.

Step 4: Set of Experiments and Evaluation

In this step the research will implement the system prototype to collect information from the set of experiments. This

information is about user behaviour in typing “username/password” using a variety of hardware. This step will analyse this information which comes as result of the set of experiments in HAUP approach. In this step the research determines the main criteria by comparing between the result of HW profiling approach and neural network analysis results that related to research question Q.6 and Q.7.

1.7 Contributions

This research develops an authentication approach to help to protect the user from identity fraud. The key contribution of this research is using HW information together with user behaviour in profiling a user to improve “username/password” based on authentication mechanism in MFA.

This research builds a framework to profile a MFA model to analyse user HW environments and behaviours in order to profile a user. The contribution is a novel authentication technique that analyses HW information and user behaviour. This approach develops the modelling of dynamic behaviour of the user to support profiling and then establishing trust in the user. The technical contributions of this research are:

C.1. Chapter [3](#) provides the built framework for Multi-factor authentication based on hardware and user behaviour.

That addressed object O.4. by providing the framework to demonstrate HW authentication methods. This contribution addresses questions Q.1., Q.2. and Q.4. by giving the method and requirements of building authentication method in this research.

C.2. Chapter 4 integrates the new authentication mechanism traditional “username/password” mechanism. That clarifies objective O.3. by providing the methodology of using HW information in authentication. This contribution in Chapter 4 answers question Q.3.

C.3. Chapter 5 provides a mathematical model for trust that combines profiling information addressing objective O.3. and answering question Q.5.

C.4. Chapter 7 demonstrates the feasibility of the approach by implementing a set of experiments to address objective O.6. and answer questions Q.6. and Q.7 by showing the advantages and impact of the HW authentication approach.

1.8 Success Criteria

The thesis success criteria are as follows:

S.C.1. Critical literature review of access control.

S.C.2. Framework for access control using Hardware Authentication and User Profiling.

S.C.3. Evaluation of the development of proposed framework using various scenarios.

1.9 Thesis Structure

The thesis is organised as follows:

The second chapter reviews authentication techniques and issues to provide the related work of profiling user behaviour techniques in authentication approaches. Section [2.1](#) analyses current authentication techniques and clarifies the limitations in authentication factors in order to contrast this research against related work. Section [2.2](#) provides background about authentication and analysing profiling user approaches to illustrate the influence of profiling requirements in cost and user convenience. Section [2.3](#) provides an overview of HW Manufacture Serial Part Numbers (HMSPNs) characteristics and their utilisation in profiling techniques that is determined to profile user behaviour in the developed approach. After that, section [2.4](#) illustrates and reviews current authentication techniques which are depending on HW information to profile a user to compare between the developed and current HW authentication. Finally, this chapter highlights the neural network

analysis to evaluate user behaviour when HW authentication is used. Section 2.5 provides the related work of neural network utilisation to profile the user.

Chapter three describes the methodology of this work to identify computer HW modeling as an authentication factor (Ownership factor) in MFA. This methodology is based on particular modeling and developing a framework solution to build a trust model. Section 3.1 illustrates the key method of Hardware Authentication and User Profiling *HAUP* approach to develop “username/password” mechanism. This method is using HW information which are considered as fixed unique numbers and difficult to temper with as a one of the platform-unique information and has been used as a platform identifier for several public services [15]. Section 3.2 demonstrates HAUP technique to observe HW information and user behaviour during the HW life cycle in order to use HW information in profiling a user. Section 3.3 explains how to profile the user with respect to analyse HW characteristics. Section 3.4 illustrates the general HAUP framework. Section 3.5 provides HAUP components to clarify the method to capture user HW and behaviour. Then, section 3.6 analyses HAUP procedures to check user behaviour based on particular HW. Next, section 3.7 presents HW MFA technique to analyse users patterns in particular HW properties

that establish HAUP authentication key for the user. Finally, this chapter determines the general requirements to implement HAUP prototype in section 3.8.

Chapter four proposes HAUP system architecture and framework. This chapter discusses HAUP system procedures to analyse user patterns with respect to HW information. Section 4.1 explains the HAUP system framework to recognize HW authentication procedures and implement HW authentication system. Section 4.2 clarifies HAUP system architecture map and determines the procedures between HAUP system components. After that, section 4.3 provides HAUP system procedures to read user's HW and observe user's behaviour using traditional "*username/password*" authentication following by explains how to profile a user to determine a level of trust.

Chapter five addresses the mathematical model of HAUP approach with respect to the motivation of using HAUP system method. This chapter provides mathematical procedure to profile the user using HAUP authentication factors. Section 5.1 clarifies HW information in mathematical expression by computing HW weight to trust the user and provides formal

assumptions and analysis for hardware profiling in HAUP approach. Section 5.2 provides illustrative examples using hardware information factors in HAUP mathematical model based on given weight of trust for HAUP factors. Then, section 5.3 provides mathematical equations to present user HW influence and profiling user behaviour in a mathematical model. Section 5.3.1 provides a mathematical model using back propagation algorithm to analyse user behaviour.

Chapter six provides the modeling of the software to implement HAUP system components. This MFA prototype is based on describes the implementation of profiling HW information and user behaviour as followed in software engineering systems. This chapter implements the HAUP prototype for gathering information to be analysed and then evaluate HW authentication approach in chapter 7. Section 6.1 provides a technical scenario for HW authentication profiling to explain analysis procedure when log-in procedure “username/-password” in progress. Section 6.2 presents procedures that are used to compare between user behaviour and patterns. This comparison assists to profile and trust a user when a user behaviour has similarity with his/her pattern in using same HW. Section 6.3 provides techniques to compute level of trust for a user followed by addressing implementation steps to analyse

user HW and behaviour. Finally, section 6.4 defines the system interaction of the deployed software.

Chapter seven analyses and evaluates the HAUP prototype results to explore the advantage of using HW information in profile the user in MFA. This chapter determines evaluation criteria in section 7.1. This evaluation criteria compare between current profiling user behaviour in authentication approaches and using HW information as profiling factor. Section 7.2 provides data analysis for user behaviour in typing “username and password” keys. Section 7.3 presents set of experiments using two different HW to illustrate user behaviour analysis result in every particular HW. Section 7.4 illustrates log-in time to support analysing users HW by to explore the difference in user’s behaviour recognition when the user moves between more than one piece of HW. Section 7.5 the ability of using priority class base on profiling HW information to illustrate trust improvement based on HAUP approach factors. Finally, the chapter ends with using neural network for the analysis of user behaviour when a variety of hardware are used to evaluate the result of HW approach in section 7.6.

Chapter eight provides a conclusion for this research and

discusses the success criteria with respect to the obtained results. Then, this chapter discuss potential improvement that would enhance the proposed approach as part of future work. Section [8.1](#) clarifies the research achievements to determine the advantages of HAUP. Section [8.2](#) presents the contribution to knowledge. Section [8.3](#) revisits the success criteria of this work to compare between HAUP and contemporary MFA approaches. Section [8.4](#) addresses the limitation and weakness of HAUP. Finally, this chapter explores future work in section [8.5](#).

Chapter 2

Literature Review

Objectives

1. Background.
 2. Authentication limitations.
 3. Profile anomaly detection.
 4. Overview of hardware authentication.
 5. Neural network analysis.
-

This chapter provides background information on current authentication approaches and analyses authentication factors. It also reviews the related work in the current authentication approach and focuses on the limitations and difficulties experienced which affected the decision as to whether to develop a new authentication approach. In this chapter, Multi-Factor Authentication (*MFA*) approaches and mechanisms that improve profiling in access control are reviewed by exploring the limitations and difficulties of developing a profiling approach. This chapter demonstrates Neural Network usage and analysis in profiling systems and provides the background to neural networks utilisation in computer security and profiling systems.

Section 2.1 analyses authentication factors in access control and provides the limitations and difficulties of deploying these factors. Section 2.2 addresses the relationship between current authentication approaches and profiling techniques that strengthen authentication and improve the level of trust [16]. Section 2.3 provides an overview of computer HW which contains significant information in respect of profiling users. Then, section 2.4 illustrates profiling approaches and describes the different mechanism used for profiling users, depending on the content of the HW information. Finally, section 2.5 analyses the neural network methods which are used to recognize user behaviour.

2.1 Authentication in Access Control

Security systems which are installed in computers, switches, routers, firewall devices and security services are all providing protection to Information Technology (*IT*) services. To protect *IT* services from any misuse, illegal authority and cyber threats, there are many built in and pre-programmed security procedures both in computers and computer networks which are in place specifically. Some of these procedures aim to support user privacy issues and some are there to protect the *IT* services from malicious misuse. Figure 2.1 illustrates an example of the security services and mechanism in place to protect computer services and focuses on authentication approaches in the application layer [17].

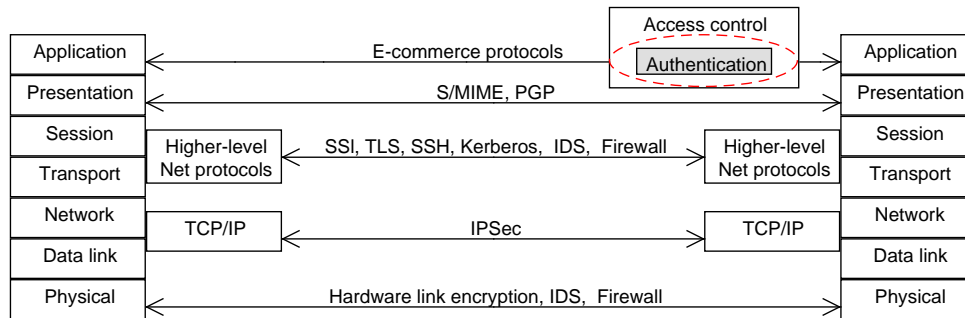


FIGURE 2.1: Network Layers and Security Service [1]

Access control is the prevention of unauthorised use of a resource, including the prevention of use of a resource in an

unauthorised manner [18]. Granting a user authentication in computer applications provides assurance of the stated identity of an entity [19]. This authentication procedure has a crucial process in access control. However, user identity compromised is the main concern in any authentication procedure. In the access control application, at the application layer that is shown in Figure 2.1 there are three main authentication security identification factors that need to be determined before a level of trust can be applied to the user [20].

These factors are:

- A. Something the user knows, e.g., username and password.
- B. Something the user has, e.g., debit card.
- C. Something the user is or does, e.g., fingerprint.

The “username and password” are an essential and traditional identification technique and a popular authentication approach that is based on knowledge factors [21]. Strong password must consist of letters (i.e. capital letters and lowercase), symbols and numbers. Using additional factors such as a credit card number as an ownership factor is required to profile and authenticate the user as part of the MFA approach. In MFA approaches, the user has to identify at least two of the three identities to verify and profile a user. MFA authentication is

a more valuable authentication identity. However, the cost of additional authentication in MFA is a difficult to meet [22].

Physiological and behavioural biometrics are the main biometrics authentication factors [23]. These authentication factors are considered as inherence factors [24]. Physiological biometrics authentication identifies physiological user characteristics which are unique inherence information. For example, fingerprints and eye retina/iris scanning. In contrast, behavioural biometrics is the process of detecting the behavioural features of the user [25]. For example, digital signature and keystroke dynamics.

2.1.1 The Limitations in Authentication Factors and Approaches

Having a record of a user's personal and key information is the authentication factor's knowledge identity. Using knowledge factors alone is considered as *Single Factor Authentication* [26] and one of the traditional authentication approaches [9]. Knowledge factor keys are not enough to authenticate the user because they are prone to dictionary or rainbow-table attacks as well as the ease with which social engineering techniques can obtain passwords. Association ownership factors coupled with knowledge factors can increase the probability of reducing any potential identity fraud. This association is one example

of *MFA* approaches [27]. Visual identity processes can also be used to verify user authentication, however, this, and other *MFA* approaches require additional costs to produce.

2.1.2 Single Factor Authentication

The most common Electronic Identification (*eID*) models are widely available on the market and there are a number of reasons as to why this particular ownership authentication model was chosen as the preferred authentication method. Initially, it was widely assumed that there would be much competition between providers which drove production costs down and therefore made this the most common and available model [28]. As a result, *eIDs* are regarded as an extremely important driver in respect of e-service development and currently are very widely used in the world of IT solutions. The *eID* server may be operated by the service provider or a third party. Some *eIDs* provide an authentication certificate as a warranty to verify it as the ownership factor supplied to the organisation or individual who rely on the service.

On the client side, a card reader, client software package and other additional accessories, e.g. fingerprint reader, are required to provide user profiling [29]. Basic card readers leave the responsibility of recording and monitoring user interaction to the software and do not record any visual or behavioural

evidence for this interaction. Advanced card readers, however, have their own PIN entry keypad to protect against malware attacks. The client software allows the protected communication between the card and the eID server, displays authorisation certificates and allows the user to restrict access to eID data fields. In addition, the chip on the ID card verifies the user's PIN and the authorisation certificate of the eID server and release of the information are authorised[30]. So, using eID relies on securing the identity and this may require additional profiling, especially in the online authentication process.

Another authentication identity project in recent ownership factor authentication approaches is Super Identity (*SID*) [31]. This SID depends on a comprehensive identification method in order to improve the trust in user's identity. SID project investigates the relations between offline and online identities, the cross-disciplinary association ranges from biometric measures through to management of on-line identities [32]. This approach defines the set of identity measures of interest and gathers relevant datasets either from existing resources or through active data collection from participants across diverse demographic populations[33]. These measures of interest fall into two categories:

- a) static and behavioural measure in real world; and
- b) static and behavioural measure in cyber world.

SID provides a step-change in the current thinking and ideas regarding the processes which are more effective in identity and identification monitoring, and places much value on the impact that this has in the real world. This approach aims to implement comprehensive authentication to trust the user by using behavioural measures.

Naji et.al. [34] enhanced authentication security in access control systems by using handwritten signatures as behavioural biometrics to strengthen and protect authentication keys from identity fraud. Their system employs the static and dynamic features of the signature to make a decision about the identity of the signature through a combination of matching statistical models to analyse them [35]. As result, handwritten signature processing and extracting their features is time consuming and requires dedicated HW environment at the user side. Chapter 3 in this research provides authentication approach based on username and password authentication technique and handwritten by recognizing user keystroke and explains the method of employing dynamic features in username and password authentication technique.

2.1.3 Multi-Factor Authentication

In MFA, a combination of methods from at least two of the basic authentication factors is used to get the authorisation;

for example, a bank card and Personal Identification Number (PIN). In some approaches, users are required to provide a password number from a security token [36].

One of the motivations of using MFA is to improve the single factor based Authenticated Key Exchange (AKE) by combining two or even more factors in one system [37]. These MFA approaches are based on a single factor and in recent times, MFA has come forward as an active research topic [38]. However, extra caution should be taken as current approaches to MFA are expensive and difficult to deploy [39].

Integrating the credit card payment system with biometrics in MFA has given support for more efficient verification. This method proposes to employ fingerprint verification with a credit card in a MFA [40]. Doing this would need the installation of additional equipment that would increase the cost.

Employing biometrics when using a credit card in authentication as a MFA procedure is another access control approach [41]. This system approaches time that affects the user acceptability for the system and using fingerprint authentication comes at low to medium cost with a medium level of accuracy.

The card reader is an additional level of HW security that can use a One Time Password *OTP* [42]. The chip on the client user card generates the OTP, with the caveat that the account

is rendered inaccessible if the card is lost or stolen. This additional challenge-response mechanism is run over a separate channel and removes the need for security questions to confirm transactions and also helps to prevent fraud. To embed the OTP in an SMS by using a mobile phone as the token reader requires accessories in the user's computer and depends on additional secure channels [43] as these will also come with additional costs [44]. With the ubiquity of mobile phones, sending an SMS text or voice messages that includes an OTP is, in effect, extending the card reader approach [45]. Here, the mobile phone is considered a secure channel, albeit with the increasing connectivity of smart phones this cannot be considered as independent as the original card reader [46, 47]. Whilst this approach reduces the cost in deploying readers, it adds additional costs on the extra communication channels and requires these channels to be accessible to the user [48].

Pennam K. [49] improved new models of accessories by using particular chips and models of improved new models of accessories by using particular chips and models of Liquid Crystal Display (*LCD*) as a method to obtain a reliable authentication factors. This approach is collaborated with the Global System for Mobile (*GSM*) messages which is implemented to decipher the fingerprint in the OTP verification LCD as a method to obtain a reliable authentication factors. This approach is collaborated with the GSM messages which is implemented to

decipher the fingerprint in the OTP verification [50, 51]. However, this technique needs to notify the Automated Teller Machine (*ATM*) and requires additional secure information from the user to deliver the OTP.

In the overall analysis of multi and single factor authentication approaches, authentication factors and features can be classified into four categories which are static or dynamic and physical or knowledge-based biometrics. On the one hand, physical biometrics is associated with the inherited physiological characteristics of the human body which is 'something the user is'. This technique employs the characteristics of fingerprints, palm prints or faces which are considered static physical biometrics. On the other hand, behavioural biometrics occur from activities carried out by the user either spontaneously or specifically learned. Dynamic or behavioural biometric techniques include handwritten signatures, keystroke dynamics, gait patterns and lip movement. Techniques that use passwords or PINs' are dynamic knowledge-based biometrics, whereas 'something the user has' techniques that utilise magnetic cards and smart cards are considered static and physical-based biometrics [52]. Figure 2.2 factors and features in current authentication illustrate the techniques.

Chapter 4 in this research declares authentication approach based on avoiding user inconvenience.

2.2 User Profiling

In order to profile a user a set of personal data relating to the specific user must be collected. In information technology (IT), this data refers to a person's identity by providing digital illustration. During the process of profiling the user the description of the characteristics of a person will be stored. A profile will comprise of a set of parameters because the variation on just one single parameter may be not be enough in itself to signal an alert. Exploiting this information by taking into account the person's characteristics and preferences can also support the identity. For example, using adaptive hypermedia systems that personalise the individual computer communication can profile the user. A computer demonstration of a user model can measure the user profile. User profiles can be found on operating systems, computer programs, or dynamic websites [55]. As a result, the authentication process is the procedure followed to profile a user by evaluating the data generated by their methods and patterns of behaviour.

Using Credit Card Verification 2 (*CCV2*) to dodge generating valid card numbers is another method used in physical

profiling predominantly to protect users when ordering goods and services over the internet. This technique supplies a card number by providing evidence that the user is in physical possession of the card which profiles the usage in order to trust him or her and this approach has been used since 1998. However, the payment card may be stolen or spoofed in which case it makes it virtually impossible to detect fraudulent usage [56].

Applying the Long Credit Card Verification 2 (*LCCV2*) to implement random CCV2 in a credit card is an improved method of saving the security keys. This approach depends on the user to secure the random CCV2 keys that requires the user to keep additional password keys [57]. These security procedures are profiling the user activity and improve the level of trust using additional secret keys. These additional keys also rely on the user's memory and ability to remember more additional password keys. This level of trust cannot differ from the real user and any another user who compromises user identity. Moreover, this level of information is required additional cost, e.g., credit cards and user memory. So, further profiling is required to recognize user behaviour.

2.2.1 Profiling Services

Profiling information can be exploited by a system taking into account the person's characteristics and preference. For example, using adaptive hypermedia systems to learn user behaviour that personalise the individual user's pattern can profile the user. Some of the relationships between identities and users, identities and service profiles and identities and devices which published standard profiling services are not yet fully understood and, without doubt, have not yet been verified in user identification services. Moreover, these standards have been fixed by requirements from particular communication domains. Applying these techniques to new multimedia applications, Next Generation Networking (*NGN*) terminals and to Web-based services will produce interesting services and yet unimagined effects [58].

R. Copeland [59] stated that the area of user identities and service profiles is beginning to be extended to support internet protocol (*IP*), multi-media sub-system (*IMS*) and Web integration. Web-based authentication and Single Sign On (*SSO*) can already be integrated with *IMS*; group management standards allow re-using groups across many applications and user profiles can accommodate data from social network websites. So, user profiling techniques for authentication should provide support for a secure information technology environment.

2.2.2 Detecting User Behaviour

Detecting user behaviour and activity, e.g., user's typing speed in the keyboard and which device the user normally uses is one of the profiling methods used to trust the user. This detecting is observed by analysing the user behaviour records that can explore user patterns. These records provide support to the unauthorised detection function as Intrusion Detection System (*IDS*) in two behaviours [60]. Firstly, the IDS approach must make a decision on a number of metrics that can be used to determine user behaviour. Analysis of review records over a period of time can be used to determine the activity profile of the average user. Thus, the review records are supplied by the definition of typical behaviour. Secondly, the current review records are the input methods used to detect intrusion. That is the intrusion detection analysis incoming review records to determine variation from average behaviour.

Using mouse biometric behaviour to verify the user by observing movement is extracting angle-based metrics to profile the user. However, this approach requires additional procedures from the user which is using the mouse as an authentication factor in order to verify the authority. Current authentication techniques are used to support the usability for the user instead of additional verification techniques [61]. One of main drawback in behaviour detection is producing many false

alarms because the user's pattern and system activity can vary too widely to be recognized. Additional drawback is the difficulty of establishing a normal definition for acceptable activity [62].

Profiling based anomaly detection system requires profiling user characteristics. In authentication approaches, profiling user characteristics should be aware of authentication factors availability. Profiling user characteristics needs additional cost to be implemented in both the server and client sides.

There are two types of Statistical Anomaly Detection techniques [63]: First: Threshold Detection System which involves counting the number of incidents of specific event type over an interval of time. If the count surpasses what is considered to be a sensible number that one misuse might be expected to occur. Second: Profile Based Anomaly Detection System which relates to user profile focuses on characteristics of the past behaviour of a user or related groups of users and then detecting significant deviation.

Profiling the user in authentication can use any free or available data resources to improve the trust. For example, profiling a user using cookies requires lower cost than using fingerprint scanner profiling. This research discusses detecting user behaviour using Profile Based Anomaly Detection System based

on hardware information which can be considered a free resource to profile the user.

2.2.3 Cookies in Profiling Users

Cookies can enclose unique identification data for the user to recognize and remembers users as profiling servers. There are many genuine uses in cookies, such as storing users' preferences and items in online shopping carts. Cookies allow websites to track the activities of users within the site in order to improve the site or to suggest products based on users' browsing histories [64].

Cookies are site-specific; however they can still be used to track user's behaviour across multiple sites. A website can allow a third party to place a cookie on a user's hard drive in order to authenticate the user. For example, adding network double click might place a cookie on a user's computer when the user visits a website that displays ads supplied by double click [65].

Browser's controls have a standard to allow the user to delete inapplicable cookies [66]. However, at least two permutations the flash cookie evade simple deletion. For example, Adobe's of Flash software allows websites to store up to twenty-five times the amount of information of a regular cookie [67].

This permits large sound and video files to preload enough information to ensure smooth playback. The software can also store data from cookies, recreating cookies with the same unique identification number even after a user deletes the originals .

The basic function of a cookie is to allow web servers to store and retrieve information on the user's machine. Although, there is no major security consideration in using these cookies however there are privacy and usability issues which affect their deployment [68]. So, using cookies' information depends on temporary keys that are stored in users' devices. Cookies keys cannot be fixed identification for user environments to profile the user because of the user ability to delete cookies information from the computer device and therefore, cannot determine user context in using a particular device

2.2.4 Keystroke and Profiling Users

Recognizing user keystroke behaviour is one of the biometric behaviour recognition processes [69]. This recognition is based on the hypothesis of different people as they type in unique and different typing measures [70]. There are many basic methods [71, 72] which are used to analyse keystrokes and thus, keystroke dynamics can be used as behavioural biometrics for users. This is the technique used for analysing users' typing behaviour and where keyboard input is monitored [73, 74]. This

technique is good to visualise the significant pattern differences between the different user's keystroke behaviour and infers that analysing the keystroke dynamics is a very encouraging method to identify a user [75, 76].

False Accept Rate (*FAR*) or false match rate is the probability that the system miss-matches the input pattern to non-matching criteria. It measures the presence of invalid inputs which are incorrectly accepted as being valid. In the case of the similarity scale, if the person is compromised in reality, i.e., if the matching score is higher than the threshold, then the user is treated as genuine and that increases the *FAR* and accordingly, performance also depends upon the selection of the threshold value [77]. In contrast, False Reject Rate (*FRR*) or false non-match rate is the probability that the system fails to detect a match between the input pattern and a matching template in the database. It measures the percentage of valid inputs which are incorrectly rejected [78, 79].

Profiling keystroke behaviour approaches is mostly characterised by the error rates in these following precision cases based on *FAR*, *FRR* [80]. For example, *FAR* is applied when user keystroke behaviour is not combined with user keystroke key patterns. So, it is insufficient to be an objective authentication factor. This implies that keystroke dynamics is a very encouraging method to identify user using *FRR* [81, 82, 76].

Chapter 7 of this research authentication approach is improving the FAR discrimination of user behaviour by further clarifying user patterns using the user's context when the behaviour is recognized and pattern is captured.

Statistical [70] and *neural network* [83] techniques are the main two keystroke approaches and there are some combinations of both approaches [84, 71]. Statistical approaches compare a reference set of typing characteristics of a specific user with a test set of typing characteristics of the same user. Neural Networks use historical data that comes from the previous usage, and then uses this data model to predict the result of a new test or to classify a new observation [85, 86, 87].

To reduce the HW environment factor that may affect user behaviour in keystroke, *Maxion and Killourhy* [88] explored a number pad input using a single finger. They tried to discriminate the users' typing style using FAR and FRR scope and suggest a low level of security that authentication using keystroke biometrics can be used in particular environment [89, 90].

Keystroke dynamics alone is insufficient to be an objective authentication factor as some drawbacks have been exposed by other research [91] which re-valued that inhibited keystrokes can come from the real word applications. This inhibited keystroke came about because of the user's environment which were influenced when the keystrokes were provided.

One research experiment explored the possibility of using modified keyboards that were based on pressure sensors to recognize users keystroke [92]. This pressure sensor keyboard has the ability to capture the password sequence when a key is pressed down, however, this feature is not available in an ordinary keyboard and would therefore come with additional costs, e.g., surface touch keyboard.

Incident response is addressing and managing the aftermath of a security breach, gaining authority or attack. This response is understood only based on the exploits used after an incident occurs [93]. So, the only data that has been gathered is what is left on the compromised system. Unfortunately, this information has delayed, limited and tells us little about the overall threat for example in "cyber crime" scenario to obtain illegal authorisation the most important weak point is hackers are not at the crime location to be profiled in early stage using any available information resources. Profiling the user remotely requires variety of method to be compatible with user devices to trust the user reliably for example if a user uses just traditional authentication "username and password" the access control system should have additional biometric recognizing technique to authenticate the user. One of the challenges faced in authentication involves alternative profiling specifically in traditional authentication "username and password". HW

authentication can support the incident response by giving HW information as crime tools in cyper crime at early stage.

Recognizing user behaviour can support and improve profiling techniques. Keystroke profiling techniques are related to providing authentication keys and can be observed by monitoring user behaviour. However, profiling keystrokes requires additional factors to recognize user activity and decipher user behaviour. Some of these factors may cause an inconvenience for the user, however, this research will discuss alternative profiling factors which can be used to recognize user keystrokes and profile the user at a low cost and with little or no inconvenience to the user. Chapter 4 discusses the method in further detail and provides examples of additional methods used to recognize users' keystroke behaviour.

2.3 Hardware Information Overview

Hardware (*HW*) is any physical computer part, e.g., mouse, screen or case, as physical systems have physical outputs. Each computer device is created as a set of HW parts, for example, the motherboard and media storage. Some of these parts are mandatory parts and others could be accessories. These parts are fixed and are not easy to tamper with.

Manufacturers of computer parts have to register their parts under the manufacturer name with a unique serial number. These numbers are considered as fixed HW information and could be made by one manufacturer or more in the same computer. In authentication systems these parts can play a significant factor to determine user privileges to gain authentications. Network card numbers or “MAC address”, hard disc drives (*HDDs*) and motherboards are all examples of HW parts.

2.3.1 MAC address

The MAC address is a 6-byte, 12- digit hexadecimal number which is divided into two parts. The manufacturers identifier is the first half of this address. A manufacturer is assigned a range of MAC addresses to use when HW part numbers are serialized. The second half of the MAC address is a serial number the manufacturer has assigned to the device. The MAC is considered to be a unique identifier attached to network adapters called Network Interface Cards (*NIC*). It is a number that serves as an identifier for a particular network adapter. Network cards (or built-in network adapters) in any two different computers will have different MAC addresses, as would an Ethernet adapter and a wireless adapter in the same computer. However, it is possible to change the MAC address in the computer device,

often referred to as MAC spoofing or cloning which is an illegal hacker's tool used to obtain unauthorised privilege [94]. Table 2.1 illustrates some common MAC address examples.

First three bytes of MAC addresses	Manufacturer
00000C	Cisco
0000A2	Bay Network
0080D3	Shiva
00AA00	Intel
02608C	3Com
080009	Hewlett-Packard
080020	Sun
08005A	IBM

TABLE 2.1: Common Network Cards “MAC address” Companies and their HW Part Numbers

The MAC data communication protocol sub-layer is a sub-layer of the Data Link Layer (*DLL*) specified in the seven-layer OSI model (layer 2). It provides addressing and channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multi-point network, typically a local area network (*LAN*) or metropolitan area network (*MAN*). The HW that implements the MAC is referred to as a MAC. The MAC sub-layer acts as an interface between the Logical Link Control (*LLC*) sub-layer and the network's physical layer. The MAC layer emulates a full-duplex logical communication channel in a multi-point network [95].

Using the MAC address alone is not reliable enough to identify the user because the user has the ability to change this hardware information that is HMSPNs. However, by using additional hardware information, this can reduce the spoofing potential. In this research project, the information from three separate hardware parts were used to provide the authentication prototype and this is discussed in chapter 7. In addition, the work discussed in chapter 8 clarifies the ability to implement a comprehensive approach that recognizes all the hardware parts used.

2.3.2 Storage Media Numbers

Storage media drivers are storage devices that store digitally encoded data. Early hard drives have removable media; however, the hard drives which are predominantly used today are typically sealed units.

There are at least 200 international companies that manufacture media storage units. Many of these companies have also now started to support new, smaller form factors that are compatible with the ever-reducing physical sizes of modern day computing and IT devices. These HW parts have unique and fixed manufactured serial part numbers. Table 2.2 illustrates some storage media manufacturer serial part numbers [3, 96].

Hard disk manufacturers numbers	Manufacturer
ST3750640AS	Seagate
WD5000AAKS	Western Digital
HDS722516VLAT20	Hitachi

TABLE 2.2: “Media Storage” Companies and their HW Parts Numbers
[3]

2.3.3 Motherboard Serial Numbers

The motherboard manufacturer serial number is another example of hardware information. This is known as the Basic Input and Output System (*BIOS*) serial number. This number could be shown on screen during the memory count when the computer is turned on. Table 2.3 illustrates some BIOS numbers.

BIOS manufacturers numbers	Manufacturer
2A5LAH09C	Award BIOS
51-0505-001437-00111111	AMI BIOS

TABLE 2.3: Some “BIOS” Manufacturers and their HW Part Numbers[4]

HW information has some characteristics of the user environment when the user has entered authentication keys to use or log on to the IT services. These HW part numbers contain significant information about the computer. For example, one or a number of specific HW parts are used when a particular action or series of actions are carried out by the user. As such,

the HW context information is profiling the user's behaviour when the IT services are used.

2.4 Using HW Information in Authentication

HW has been used to facilitate authentication for a long time. The idea is that owners/users register their devices based on their MAC address so that, the devices themselves are authenticated, rather than their users. MAC addresses are used in the cryptography of files, authentication and integrity networks to support the security of data transportation. This technique uses the MAC address as a key authentication factor to secure the communication session with the Internet Protocol (*IP*) address to reach the device destination [97].

Filtering MAC addresses to secure the wireless network is essential in giving users access to the wireless network. Doing so will give precise control to wireless users connected with the Access Point (*AP*) associated with their MAC address [98]. If this filtering is not applied and the MAC address of the client is not given, the client will not be granted access to the wireless network. So, MAC addresses of the client computer device gives the authorisation needed for a wireless connection which is between the client and server [99].

Spoofing attack is a situation in which one person or program successfully masquerades as another user by falsifying data and thereby gaining an illegitimate advantage [100]. Spoofing of MAC is usually beyond the average wireless user's experience. In order to carry out spoofing on a MAC address, the client needs to be associated with a particular AP. As result, using the MAC address in wireless security depends on filtering the MAC address of the client without determining the user's characteristics.

Another method of HW authentication usage is storage media drivers such as HDDs . Each storage media item has a unique HMSPN as an identifier product code that can be used in profiling [101]. These HMSPNs are already actively used for identification, albeit that they can be modified at firmware level and thus are susceptible to spoofing. For example, Microsoft products send product and HW identifiers during the activation process [102, 103]. So, this HW information provides the opportunity to profile the user's computing environment.

Port security is a mechanism which is used to restrict the MAC addresses that connect via a particular port switch. This tool allows defined and specific access to a particular port to allow a unique MAC addresses, or a range of MAC addresses. To connect to the LAN port, it will allow access of MAC addresses which belong to a range according to a configured list. When a frame arrives to the switch it will compare the MAC

addresses with the MAC addresses on the configured allowed list. If the MAC address matches one of items on the list then the packet is allowed to go through. In contrast, if the MAC address does not belong to the configured list the port will drop the packet. So, MAC addresses can be specified to connect to a certain port. This type of firewall can support authentication [104]. This level of information has some characteristics of the user's HW environment which can profile the user activity by using particular HW.

In "Active Directory Integrated Media Access Control" based wireless authentication, the Internet Authentication Source (*IAS*) needs to be installed on a domain controller to ensure that the domain controller belongs to the Remote Access Service (*RAS*) and IAS source group. To proceed with this process, a Security Group in Active Directory is created which should have the MAC address of the laptop's Wireless Cards. These are identified as "Wireless MACs".

Users are created by using the MAC address as a USERNAME and the AP is shared by a secret password. These users should be controlled by a security group created earlier by the network administrator. After creating a remote access policy in the IAS, this will permit remote access through the membership in the Windows group that was made previously. This course of action has been taken earlier in "authenticate wireless MAC accounts, based on group membership" [105]. A unique

and constant MAC address is transmitted by 802.11 devices and thus are identifiable. It was recently proposed to replace such identifiers with pseudonyms, i.e. temporary names which were unable to be linked to the IT device due to the fact that implicating identifiers or identifying characteristics of 802.11 networks traffic can identify many users with high accuracy [106].

Another profiling technique uses four implicit identifiers visible to the piece of HW to quantify how well a passive adversary can identify users. A lower boundary is placed on how accurately users can be identified implicitly by using the following:

1. Identifying four previously unrecognized implicit identifiers: network destinations, network names advertised in 802.11 probes, differing configurations of 802.11 options and sizes of broadcast packets that hint at their contents.
2. Develop an automated procedure to identify users which quantifies how much information is revealed via implicit identifiers, both singularly and in multiples, and which can reveal about several hundred users in three empirical 802.11 traces.
3. The evaluation shows users produce highly discriminating implicit identifiers. Even a small sample of network traffic can identify them, i.e. more than half (56%) of the time in public networks. Moreover, it is most unlikely that they would

be mistaken as being the source of other network traffic (1% of the time). Since adversaries will obtain multiple traffic samples from a user over time, this high level of accuracy in traffic classification enables them to track many users with even higher accuracy than in common wireless networks.

4. It is the first time it has been shown with empirical evidence that design considerations beyond eliminating explicit identifiers, such as unique names and addresses, must be addressed to protect anonymity in wireless networks.

During the course of this research it was [106] noted that by considering a subset of all possible identifiers and a weak, passive adversary, the results only place a lower boundary on the accuracy with which users can be profiled. The efforts are continuing to uncover implicit identifiers exposed in 802.11, such as those exposed by timing channels. The accuracy of the implicit identifiers over longer timescales and across different locations will be evaluative, since this study analysis is limited by the duration and location of the traces.

In 1998 the University of Pittsburgh established a network connection to residence hall students because the number of residence hall beds had increased to 6,000 and the connection rate had continued to increase to 74 percent of resident students. Students were implementing a manual process to assign static IP addresses and record each computer's MAC address.

This then required the entry of a username and password each time the user established a connection. After that, the 2000 Dynamic Host Configuration Protocol Automated Teller Machine (DHCPATM) was used to provide IP addresses for each student in conjunction with registration software to record the necessary machine information. This technique, however, was considered to be too time consuming for tracking security activity [107]. Point-to-Point Protocol over Ethernet “PPPoE” technology was used to improve the ability of secure access to the wireless network. So, a single and easy system can be configured and used for all users. In spite of this the wireless or traditional wired ports connection must be implemented in order to avoid confusion and to offer users flexibility in public areas without needing to re-authenticate or switch to a different authentication mechanism wireless network [108, 109]. Therefore, using additional HW information may support this access control approach to avoid the confusion of roaming from wireless to traditional wired ports in LAN.

Another technique uses specific network security devices. Network security devices are connected between a protected client and a network. The network security device negotiates a session key with another protected client. Then, all communications between the two clients are encrypted. The device is self-configuring and locks itself to the IP address of its client. Thus, the client cannot change its IP address once this has

been set and therefore cannot emulate the IP address of another client. When a packet is transmitted from the protected host, the security device translates the MAC address of the client to its own MAC address before transmitting the packet into the network. Packets addressed to the host contain the MAC address of the security device [110].

In order to verify the client's username and password the Secure Remote Password protocol (*SRP*) [111] modular performs large integer exponentiations. This task requires many operations and consumes a large part of the total execution time of software implementations of the SRP protocol that are affected by HW performance. Modifying or designing a suitable HW environment to accelerate the exponentiations modular in the SRP protocol [112, 113] is associated to user's HW and affects in observing user behaviour.

A mouse is a dynamic biometric that is similar to keystroke dynamics. The mouse is very important for graphical user interface (*GUI*). In contrast, the keyboard is essential for command line based applications. The behaviour of both these devices can be combined in a common detector. Adapting keystroke technology by addressing issues such as passive and dynamic monitoring could improve the detection [114]. However both detectors may be affected by the keyword and mouse environment that motivate the focus in users' devices which affect user detection.

A user's HW can support a reduction in digital identity fraud. However, because of natural or analytic HW authentication, this level of information is related to the user's confidentiality and integrity which are a primary concern and thus, any implementation of a new authentication method will have to be aware of this. In this research, HW information is used as the authentication factor.

2.5 Neural Network Recognition

A neural network is a set of simple processing elements that exhibit complex global behaviour determined by the connections between the processing elements and element parameters [115]. A neural network is used to learn procedures through mapping approximation function about a user's behaviour. Neural network tools have techniques to achieve high capability of probability systems [116]. So, the neural network has an adaptive rate of learning and contains popular techniques to analyse and profile user behaviour which supports the process of authenticating the use [117].

2.5.1 Neural Network Analysis

Artificial neural networks are the self-processed "training" of connecting artificial neurons. Artificial neural networks can be

used to gain an understanding of biological neural networks without necessarily creating a model of real biological systems [118]. The biological nervous system is highly complex and artificial neural network algorithms attempt to abstract this complexity and focus on what may hypothetically matter most from an information processing point of view. Neural network performance is mimicking human error patterns [119]. Good performance, e.g., as measured by good predictive ability, low generalisation error, or performance human error patterns, can then be used as one source of evidence towards supporting the hypothesis that the abstraction really captured something important from the point of view of information processing in the brain. [120] Another incentive for the neural network is to reduce the amount of computation required to simulate artificial neural networks, so as to allow one to experiment with larger networks and train them on larger data sets.

Artificial Neural Network (ANN) is an information processing model that is stimulated by the way biological nervous systems process information. The key element of this model is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unanimity to solve specific problems [121].

2.5.2 Neural Network for Profiling

Multilayer Perceptions (*MLP*) neural network and Radial Basis Function (*RBF*) networks have become the most widely used network architectures in pattern classification problems. The general difference between the two neural networks is that MLP is a more distributed approach compared to RBF, which only responds to a limited section input space [122, 123].

Neural networks have the ability to do learn from examples by using generalising algorithms. This learning can be used to identify data that have not been seen in the system. A new type of attack or compromising authorisation can be identified by the neural generalising algorithm and this ability can help in the investigation of crimes. In addition, forecasting is used to predict what will probably occur in the future based on current information for example a neural network with a forecasting algorithm can give an output predicting who is likely to engage in bad behaviour [124]. Thus, forecasting algorithms might be useful for investigators by providing a list of suspicious people. However, such algorithms might not be beneficial for attacks perpetrated by outsiders, however are more likely to be useful for illegal behaviour by people who work in the same organisation.

2.5.3 Neural Network for Security

Neural networks have recently been applied to computer security and are seen as an improvement over expert systems [125]. Expert systems use a set of security rules acquired from the knowledge of human experience. They are able to detect attacks which are defined by these rules, however if a new type of attack is launched, this system may ignore it, leading potentially to great damage to the system. Therefore, an expert system needs to be regularly updated to correspond to the improved methods by which assailants may try to break down these systems. Indeed, such updating may sometimes not be sufficient, because even if an improvement is made to the system, it may still not recognize an attack which is made after this updating has taken place. Due to the ability of neural networks to deal with new events, basing a computer security system on an ANN has the advantage of being able to detect any kind of attack or obtain illegal authority. This ability improves the mechanism to make the system safe from any new methods that attackers try to establish. Sammany in [126] state that a neural network is able to detect users patterns which have never before been seen in the network system, because it has the property of generalisation.

2.5.4 Neural Networks and Intrusion Detection Systems

A network system needs to be monitored to detect any attacks that might harm the system such as unauthorised access of intruders to the network. These attacks are usually detected by an intrusion detection system [127].

Intrusion detection has two main techniques: anomaly and misuse detection. The anomaly technique is used to detect intrusion by seeking unusual behaviour in the network while misuse detection searches for actions that match descriptions of assaults known as 'signatures' which have been applied to this technique [128]. There are three stages to the creation of a neural network ID [129]:

1. Gathering data from training: for each day and user, there should be a vector that represents how often a command is regularly executed by a user. This can be achieved by having audit logs applied to each user over several days.
2. Training: the user is recognized by training the neural network depending on the commands represented by the vectors.
3. Execution: when the detector identifies a user, it will state whether this is a known user. If not, the system recognizes this user as an attacker.

In this research, user behaviour information will be collected by monitoring typing patterns on the computer keyboard. Neural networks will learn this information based on particular environment.

2.5.5 Neural Network Anomaly Detection

In anomaly detection systems, artificial neural networks have been applied as a substitute for statistical analysis. In statistical analysis techniques, an attacker is recognized by comparing normal with current behaviour [130]. Neural networks were specifically proposed to identify the typical characteristics of system users and identify statistically significant variations from the user's established behaviour.

2.5.6 Neural Network Misuse Detection

The methods of network assaults are continuously changing, so a system is required which is flexible in defence and protection and which is able to analyse the huge amount of data in the network. Neural networks have the ability to analyse information from the network, even if it is incomplete or inaccurate. Furthermore, its learning ability enables it to detect dangerous attacks in cases where many attackers strike the network at the same time. ANNs are also very fast, allowing them to detect an

attack before it can cause great damage [131]. Their learning ability protects the network from any attack that has been seen before because the system has discovered the attributes of this attack from previous events.

2.6 Summary

The previous chapter presented background about authentication factors and approach in access control in Section 2.1 followed by focusing in profiling factors and technique to vitrify the user and answering research question Q.1. by addressing hardware usage in authentication. Then, Section 2.4 declares hardware information and hardware usage in authentication which indicates research question Q.2. and clarifies hardware characteristics to be used in profiling the user. Finally, Section 2.5 demonstrate neural network usage in profiling and recognizing user behaviour in security systems.

This chapter established one of the important weak point in computer authentication. This weak point is how to protect authentication identity keys from been used fraudulently and are users/hackers protected from being captured. As result of the literature review in previous chapter, the following limitation need to be addressed to improve authentication methods:

L.1. Password authentication is not reliable due to the users inability to keep them secret; passwords are prone to dictionary or rainbow-table attacks as well as the ease with which social engineering techniques can obtain passwords that is clarified in section [2.1.1](#).

L.2. Current authentication approaches to MFA are expensive and difficult to deploy which is increase the cost to profile the use (See Section [2.2](#)).

One of extending behavioural profiling research in information security is measuring deviant behaviour by data collection and measurement issues, e.g., improving methods for collecting and measuring security related data to capture actual behaviour [[132](#)].

Profiling user behaviour can improve the authentication method to trust the user because of the profiling identity is referring to a person which has description of the characteristics of a person. This profiling is based on using methods of recognition to analyse and then identify specific user behaviour, e.g., keystrokes in typing the authentication keys. However, profiling user behaviour requires additional accessories and needs system capability to observe user patterns in any context to determine user behaviour.

Computer HW environments have physical characteristics. These physical characteristics are a) fixed and physical gates

to use digital resources for one user or more and b) difficult to tamper with. So, how should we use computer device characteristics to improve profiling techniques in authentication?. Profiling HW information can be used to discriminate against the valid use of password credentials against the misuse of password credentials by an attacker, without complicating the authentication process or incurring large extra costs.

The next chapter will discuss the methodology of profiling user's activity using HW information. This discussion determines HW authentication and user profiling "HAUP" approach frameworks to build new authentication approaches. The next chapter provides *HMSPNs* contributions to improve the traditional username and password authentication.

Chapter 3

Methodology

Objectives

1. Define authentication in HW life cycle.
 2. Introduce HW activity and user's behaviour.
 3. Formalise analyse and explore HW approach.
 4. Explain framework and requirements.
-

This chapter provides an overview of the proposed framework which is concerned with the process of building the “authentication approach” based on the use of HW information in traditional “username and password” technique. Section 3.1 provides key methods and motivations which determine the HW technique and characteristics to be used in the authentication framework. Section 3.2 explains the HW authentication procedures which are required to observe HW information and monitor user behaviour (Bh) during the HW *life cycle*

Section 3.3 explains how to analyse HW characteristics to profile the user. Then, section 3.4 outlines the HW authentication framework. Section 3.5 provides HW approach procedures to be implicated in the traditional *user name and password* technique. Section 3.6 provides the HW authentication approach to check user HW and Bh as authentication procedure in access control. Following this, section 3.7 presents the HW contribution in profiling a user to reduce the potential of identity fraud. Finally, section 3.8 determines the main requirements which are necessary to implement the HW authentication approach.

3.1 Method

Users can't assume the pattern of somebody else when he or she deals with a particular computer device because of the profiling inability to have comprehensive observation for user's environment when the computer device is used. However, there is a possibility to establish a profiling map to trust the user. Users' computer devices which have been used to get the authority may hold significant information about the user, e.g., the computer device may been used in successful login attempts by the same user or other users in previous usage. Can the security systems use computer devices characteristics in authentication?

The main physical and digital requirements which allow users of information technology to compromise other users are computer device and authentication keys (See Chapter 2 Section 2.6). This raises the question of how to access the control techniques that profile the user using these basic requirements?

Each person has their own usage pattern when he or she uses any computer machine or smart device (See Chapter 2, Section 2.2.2). A user's Bh and pattern is recognized by their performance and device analysis which is a significant factor in profiling the user. If this is the case, this also raises the question of whether the computer device develops the profiling by exploring the relation between user behaviour and pattern every successful log-in attempt.

Every user has the right to keep using a particular or variety of HW to perform authority in IT services. In this approach, physical information is considered as the user's contextual environment during the authentication process. This approach records an advanced impression of the occurrence of misuse which is considered as a user's HW environment and contains a user's characteristic. These characteristics recognize a user's physical Bh for profiling aims. This recognition of user Bh came as a result of using a particular computer device by a user. This information can be reused together with user Bh and usage profiles as an authentication approach to identify malicious access behaviour.

The HW approach is the process of profiling the way in which a user's device has been utilised to obtain the authority to access control for every successful log-in attempt. This process motivates a user's machine to analyse and determine the user's Bh by using, in particular, HW as an authentication factor.

This research explores the use of HW information as an embodied identity to recognize and analyse a user's environment in order to use the results for profiling user behaviour. In addition, the research also clarifies the similarity between user's Bh and patterns based on using specific HW by a user to get access.

Following HW characteristics motivate to use HW information in profiling user's activity to develop profiling a user in traditional "username/password" authentication approach:

M.1. Computer HW has significant information (HMSPNs).

M.2. Computer HW information is difficult to tamper with.

M.3. Computer HW information is considered a user environment during performing access requests.

M.4. Computer HW information has particular characteristics that are encouraged to be used in profiling at access control.

Moreover, HW configuration can be reused, together with user activity and usage profiles, to identify malicious behaviour. Profile usage can be obtained by correlating the HW configuration and user Bh when accessing information technology services. Chapter 4 clarifies the HW profiling technique to authenticate the user in access control threshold.

3.2 HW Authentication in HW Life Cycle

HW parts also have a specific usage history by sorting the users based on HW usage. Some computer HW parts have not changed and have been used by the manufacturer for a long time. Every computer device has a history which is tracked

during the time of its life cycle. In other words, every single computer's HW has a specific record of usage by all users from manufacture to destruction.

If a user has been using the same device and following the same log-in procedure for a long time this user will have a particular pattern in using a particular device. Therefore, the user has a particular pattern range that will be used to recognize user Bh based on specific HW. If the number of users of a particular device increases, the access control system recognizes HW performance to recognize how users behave when the authority is taken. For example, user keystroke Bh is captured when the username and password keys are typed by calculating the keystroke speed and user typing rhythm, even if a group of users use the same username and password. Of course, the sharing of accounts is bad practice, but is still commonly encountered in both domestic and corporate environments over which the service provider has little influence. Figure 3.1 shows the users the two users Bob and Colin used John's HW, however they have different behaviours in dealing with same HW.

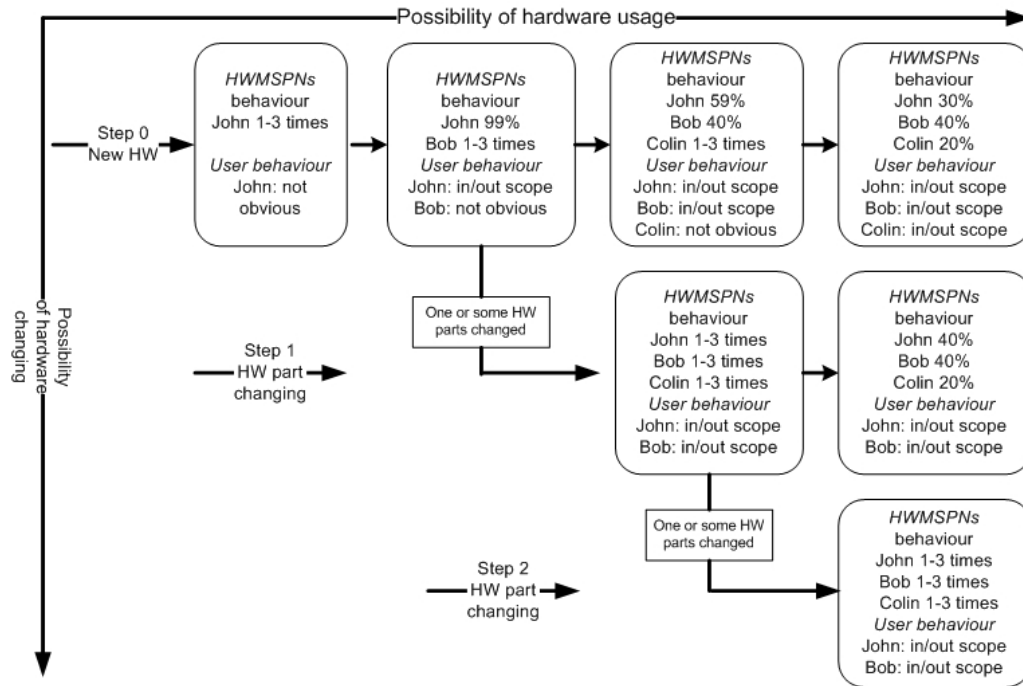


FIGURE 3.1: HW and Users Bh Life Cycle

HW *life cycle* in Figure 3.1 explains conceptually the HW usage. HW usage provides the opportunity to learn users' Bh depending on a particular HW configuration. However, the HW parts may change over time resulting in configurations that are distinctive to previous log-in attempts by their users. For example, the log-in may be typed on the touch screen or (after attaching the tablet to a docking station) through a physical keyboard. These changes in HW configurations affect the user profiling. "Step 1" and "Step 2" in Figure 3.1 reflect changing HW parts and thus a change of user's environments. Therefore, the HW approach has to recognize changes in HW and determine users' HW at every log-in attempt. As a result, using

HW information in the HW authentication approach in access control could be a new factor to profile and verify the user that is detected by a user's computer HW configuration behaviour.

The HW authentication approach records and analyses the different patterns of users' Bh when they use the same HW environment and the same username and password. A HW authentication technique maps users' computer HW *environments* in order to recognize the user patterns in particular HW.

Profiling Bh techniques are based on recognizing users' unique biometric Bh denoted in chapter [2.2](#). However, profiling behavioural techniques cannot determine when, and which, particular physical feature is executed that recognizes the user's environment when observing user Bh techniques. HW profiling focuses on user behaviour depending on a particular environment and observes the methods of behaviour. The HW approach provides a level of trust which depends on the profile of a user's computer behaviour in traditional username and passwords authentication with respect to HW activity. Figure [3.2](#) illustrates the HW profiling technique and factors in HW authentication approach.

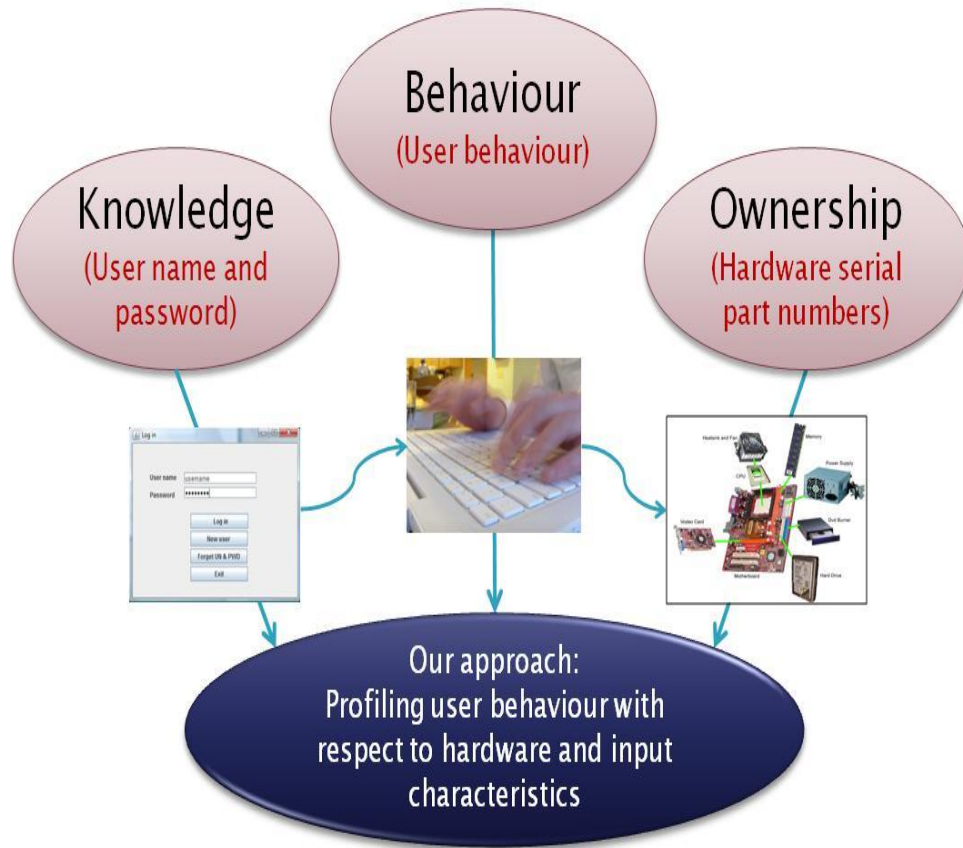


FIGURE 3.2: HW Method for HW Approach

3.3 Authentication HW Analysis

When a user attempts to log-in using identity authentication, namely username and passwords, HW authentication analyses user's HW environment by collecting three HW manufacture serial parts numbers. This HW configuration explores whether the user has used the current HW in log-in procedure previously. If so the HW authentication approach calculates how

many times the user has used the current HW log-in. If the user did not use the current log-in previously, the HW authentication approach deals with the current HW as “manufactured HW first usage” and this approach is the technique to be used to observe user’s behaviour.

So, if “First usage” the HW authentication approach cannot profile user behaviour due to the lack of the user’s patterns being previously recorded. In if “First usage” the approach can redirect the user to another verification approach, e.g., additional password. However, the HW approach begins to learn the user’s pattern from first usage to recognize the user’s patterns to be used in the following profiling user’s behaviour in the next successful log-in attempt.

Following Table 3.1 demonstrates an example of particular HW usage by 6 users. In this example, three particular HW parts are assumed to being the user environment when successful login attempt occurred. These HW parts are a) Media Access Control MAC Address; b) Storage Media Manufacture Serial Part Number; and c) Motherboard or BIOS Number. Using three hardware parts simplifies HW authentication approach in this research and will be improved by using more HW parts in further development stages.

TABLE 3.1: Profiling User's HW

#	User	Successful Log-in	MAC	Storage	BIOS
1	Peter	3	00-1C-C0-E6-38-4C	82566DM-2	AZCB927006JW
2	Linda	546	00-1C-C0-E6-38-4C	82566DM-2	AZCB927006JW
3	Cress	255	00-27-0E-20-6F-0D	SAMSUNGHD	<i>CNF8375GR0</i>
4	Antonio	456	00-1E-68-F3-44-C4	<i>3COM113321d</i>	2A5LAH09C
5	Antonio	1	00-1C-C0-6D-6E-AA	<i>3COM212121N</i>	2A5LAH09C
6	Colin	0	<i>00-1C-C0-E6-38-4C</i>	3COM113321d	<i>CNF8375GR0</i>

In Table 3.1 in first example we assume that there are three users, namely: Linda HW number of successful log-in attempts (546), Cress HW number of successful log-in attempts (255) and Antonio HW number of successful attempts (456) that have used their HW to access their accounts for many successful attempts. If HW is used for many log-in attempts, HW is trusted as a log-in environment and the HW authentication approach recognizes the physical HW. This HW information can disseminate user behaviour in a particular HW environment.

In the second example, we note that users Linda and Peter use the same HW at every successful log-in. The HW information recognizes that group of users use same HW.

In the third example, Antonio HW number of last successful log-in attempt (1) has changed two of his computer HW parts (MAC address and Storage media), so the level of trust in the HW authentication approach should recognize HW changing has influenced the profiling of the user's behaviour.

In the fourth example, we assume Colin tries to compromise another user's HW or perhaps Colin is using his friend computer. Colin has used HW information from another users HW which affects the HW trust. Trusting a user's HW can play a significant factor in recognizing user's behaviour using particular HW. User movement between one or many HW affects the ability to observe biometric behaviour. This HW information requires additional factors to trust the users' which is based on the HW environment.

If a user keeps using particular HW during every successful log-in attempts, the HW approach learns that the user is familiar with this particular HW and the HW authentication approach determines the HW user pattern based on particular HW. The level of trust can be increased because the user behaviour is observed by the HW approach through using the same HW at every successful log-in attempt because the user will have particular pattern in using particular HW. However, if the user behaviour in using particular hardware is not similar to user pattern the level of trust is decreased and may required additional verification method to authenticate the user. Figure 3.3 illustrate how HW method can support the level of trust to authenticate the user.

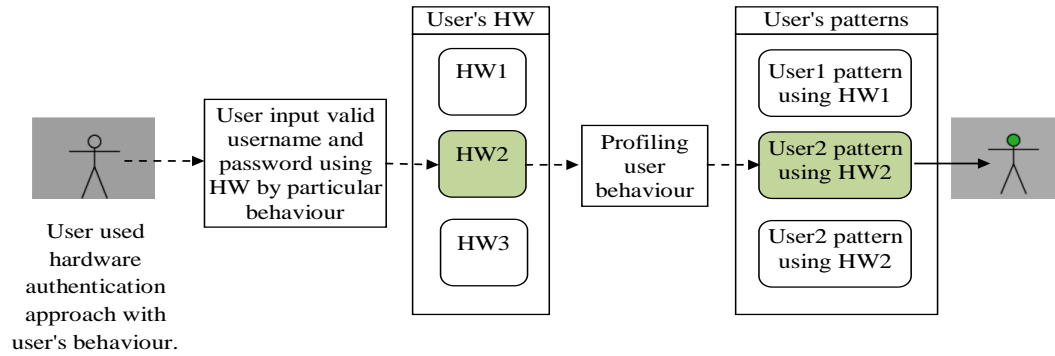


FIGURE 3.3: HW Method in User Authenticate

The level of trust a user is increases dramatically when HW approach learns user pattern and the user Bh is not change every successful log-in. So, In HW authentication approach the level of trust a user will be based on profiling the following three factors:

F.1. Successful log-in attempt using “*username/password*”.

F.2. User HW that is used at every successful log-in attempt procedure.

F.3. User Bh every successful log-in procedure including.

3.4 Design HAUP Framework

The HAUP framework is located in 'access control edge'. The HAUP framework starts when the user has requested to gain

the authority using the log-in application from a particular computer device and finishes with giving a level of trust for the user. The HAUP framework has two parts; the first part is located in the client device which collects the user's HW and behaviour. The second part is allocated in the server side to analyse HW usage and behaviour from stored patterns in the HW database that has previous HW usage analysis. In the server side the HAUP controller calculates the similarity between the current user Bh and the previous user's patterns to present the level of trust. Figure 3.4 shows the general framework for HAUP approach.

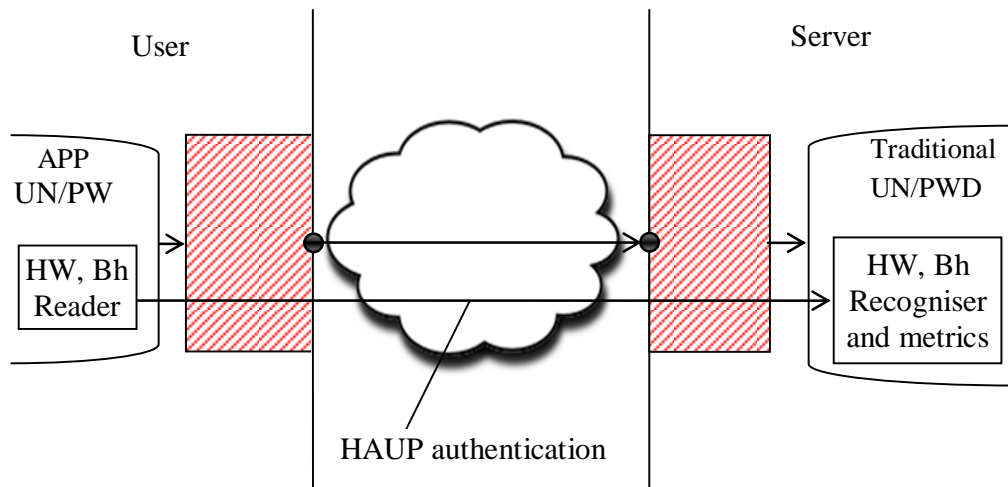


FIGURE 3.4: Framework Overview for Designing HAUP Aspect

3.5 Analysing Hardware Authentication and User Behaviour

HW authentication focuses on Analysing Hardware Authentication and User Profiling (HAUP). The HAUP approach checks users computer HW authentication and focuses on analysing hardware authentication and user Bh. The HAUP approach checks users' computer HW parts at every successful log-in attempt which may have been affected by a change either partly or completely of these HW parts depending on the time of the day, week or month that the user has used the computer which could, in turn, affect the user's Bh. For example, if the user has changed his keyboard, the HAUP approach have to recognize the new keyboard to create the required level of trust because this change may effect in recognizing userbehaviour.

The HW authentication approach depends on a minimum of three mandatory HW part numbers. For example, when a user uses more than one device with the same log-in during the day e.g. desk top computer at work, smart mobile phone and laptop, the HW authentication approach should recognize the changing of user's Bh environment because the method of HW observation also have changed. In addition, the HAUP approach learns user Bh sequentially from first usage and this learning is reflected in the user's level of trust which increases at every successful log-in attempt.

Analysis of user-typing patterns on a particular HW is discriminating username and password keys by monitoring user Bh in dealing with particular keyboards. Moreover, every computer device has particular profiling about user's pattern. Recognizing user Bh should be aware of HW to profile the user's reliability.

This correlation between user's Bh and HW is reducing the False Accept Rate and False Reject Rate rates and allows the approach to be deployed throughout heterogeneous approach which are comprised of various HW interfaces. For example, in Figure 3.5 the user uses four different HW during the day. The user uses the same username and password to log-in to the system by particular HW at specific time. HAUP is profiling the user based on the HW information which is related to particular time. This analysis discriminates user Bh in particular HW at particular time which observe user pattern when the use performance change during the day.

Analysis of user-typing patterns on a particular HW is based on monitoring user Bh in dealing with particular keyboards. Moreover, every computer device has particular profiling about user's pattern. Recognizing user Bh should be aware of HW to profile the user reliably. This correlation is reducing the FAR and FRR rates and allows the approach to be deployed throughout heterogeneous approach which are comprised of various HW interfaces. For example, in Figure 3.5 the

user uses four different HW during the day. The user uses same “username/password” to log-in to the system by particular HW at specific time. HAUP is profiling the user based on the HW information which is related to particular time. This analysis discriminates user Bh in particular HW at particular time which observe user pattern when the use performance change during the day.

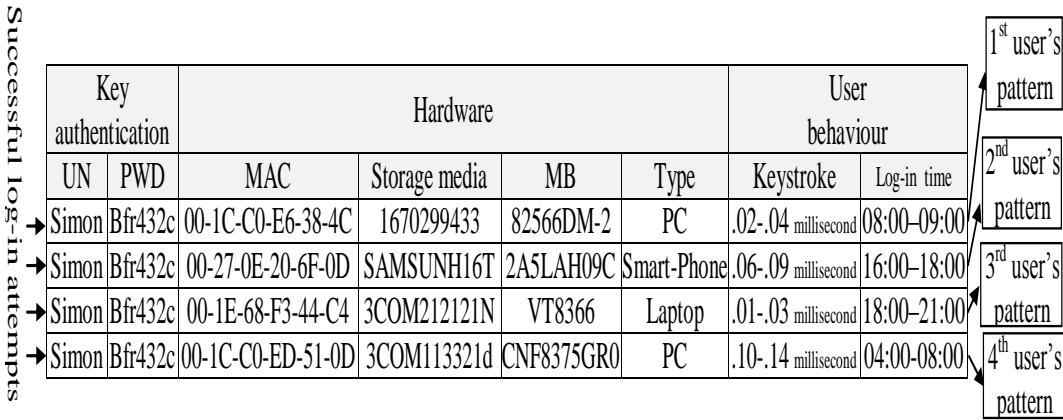


FIGURE 3.5: Main Authentication Factors

HW and user Bh profiling combinations have three main factors to authenticate a user. The first factor is the traditional authentication factor which is the username and password. The second factor is user’s Bh within a particular HW environment.

The third factor is the user’s HW which is the infrastructure element which is necessary to learn user Bh. Figure 3.6 illustrates the HAUP steps to determine the level of trust in a

user. If more than one user uses the same HW for log-in, the HAUP approach is required to determine how the HW have an influence in the user Bh.

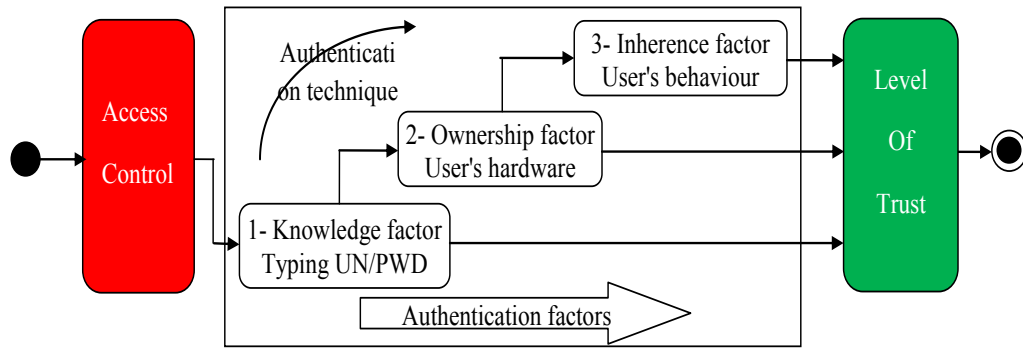


FIGURE 3.6: Overview of Dependencies of HW Authentication Factors

3.6 HAUP Approach Analysis

The HAUP approach provides two components in the log-in procedure. Whilst the user is typing his or her username and password, the first component captures the current user Bh by calculating the keystrokes (both key-press and release) speed when username and password are typed. The second component collects the HW information which consists of the user's current HW configuration. As the user or other security software installed on the client device can prevent the gathering of HW information, we consider this to be optional information. However, if this information is not provided it has detrimental

effects on the accuracy of the HAUP approach as the HW profiling information is coupled with the selection of the user-profile for keystroke recognition.

When the user performs access using particular computer HW, the approach begins to analyse and compare the current HW configuration with the established profile of that user that is stored in the server side to determine the similarity of the Bh. If the user has used the current HW before, the approach computes the similarity between the current keystroke Bh of the user and the Bh that has been recorded against this hardware configuration previously. If the current HW configuration is not in the database, the component compares the user pattern against all known keystroke Bhs for that user, indiscriminate of the HW configuration. This obviously reduces the efficiency of this approach.

As a result, the HW similarity test reflects the idea that the HW that has been previously used by the same user increases the likelihood of the user being genuine, as this rules out attacks in which passwords have been observed by shoulder surfing or rainbow table attacks. Uncharacteristic use of HW, e.g. the use of a company PC that has regularly been used during office hours for a period of 6 months which now has an access taking place at 2am in the early hours of the morning, will be flagged up by a low trust level in the HW.

3.7 HW Information Contribution in HAUP Approach

The HAUP is a MFA technique that discriminates traditional username and password factors and, in particular, Bh factors by analysing HW characteristics. The HAUP approach binds between these three sequential and dependent factors based on the traditional username and password log-in function. HAUP concentrates on user HW environments that affect user performance. Using HW information discriminates the profiling technique to accurately determine usage patterns carried out by the user.

The HAUP approach is built in “*MHSPNs*” accessories ‘on their own’ to improve the authentication technique by determining user environments and by using low profiling costs. Additional factors that are implicit in monitoring users’ HW Bh in the particular HW are given specific log-in keys text, i.e., the username and password.

3.8 HAUP Approach Analysis Requirements

Implementing the HAUP approach necessitates the following requirements:

R.1. Collecting HW information from user’s devices.

R.2. Analysing the history of the usage of HW.

R.3. Capture and collect user Bh after every successful log-in attempt in the access control threshold in user's devices.

R.4. Analysing user Bh in using current HW from previous successful log-ins.

R.5. Find the similarity between users' patterns and current users' Bh when the same HW is used.

R.6. Determine the level of trust for the user based on the correlation between users' patterns and Bh.

3.9 HAUP Success Criteria

HW information can support profiling systems in access control if HW is considered as a user environment to recognize user Bh. HW information can support profiling a user if the user keeps using particular HW at every successful log-in attempt. So, the HAUP system criteria success is based on the following:

1) Hardware availability at every log-in attempt which is essential to identify user Bh in particular HW.

2) Profiling Bh ability to clarify user Bh. This profiling requires more than one technique, e.g., recognizing typing rhythm and speed, which supports to observe user patterns.

3) User ability to perform his/her Bh at every successful log-in attempt that related to his/her normal pattern. This ability gives the opportunity to learn user patterns and find the similarity between user Bh and patterns at every successful log-in attempt.

3.10 Summary

This chapter clarified new method to improve traditional authentication approach using HW information. Section 3.1 determined HW information characteristics to be used as authentication factor in access control. Then, section 3.2 demonstrated HW information availability during log-in procedure to be used in profiling the user that provided clear answer for research question Q.3. by addressing the methodology of using HW information to profile the user which aims to research objective O.2. by selecting suitable characteristic for profiling. After that, 3.3 presented more clarification and examples to analyse user HW in access control to profile the user that answer the research question Q.4 by addressing HW characteristics. Section, 3.4 clarified general framework to use HW in traditional authentication approach which is the first contribution C.1.(framework for Multi-factor authentication based on hardware and user Bh) of this research and conducted the research objective O.4. by implementing set of experiments.

Section 3.5 and section 3.6 focus in analysing HW and user Bh that improved profiling the user to determine user pattern. Section 3.7 demonstrated HAUP approach advantages that clarified research question Q.5 (What is the added cost of a profiling approach?) answer by decreasing profiling cost and determined the main analysis requirements to successes HAUP approach in section 3.8. Finally, section 3.9 determine the success criteria to evaluate HAUP. So, using HW information as profiling factors to determine a user's Bh can increase the effectiveness of observing a user's patterns. Profiling a user's pattern in particular HW identifies hackers misuse in user Bh.

The HAUP approach profiles a user's Bh based on HW information to categorise user Bh in particular HW environments. This approach has the potential to reduce identity fraud without additional accessories' costs or inconveniencing the user. This chapter presented the HW authentication approach analysis to draw the HAUP framework.

The next chapter will discuss the HAUP system design and procedures to declare HAUP architecture and then implement the HAUP prototype in chapter 6. The next chapter will provide an authentication system analysis using HW and Bh recognizer when the traditional "*username/password*" approach is in progress at the access control threshold.

Chapter 4

Hardware Authentication and User Behaviour Framework

Objectives

-
1. Introduce the framework.
 3. Describe the components.
 2. Define the architecture.
-

This chapter discusses the HAUP framework that provides user pattern analysis based on HW information. This framework uses profiling of user behaviour methods based on an analysis of user HW activity in accordance with “*HMSPN*’s”. Section [4.1](#) provides an overview of the HAUP framework to recognize user HW activity and profile users’ methods and behaviour. After that, section [4.2](#) provides HAUP system architecture and determines the procedures between the HAUP system components. Section [4.3](#) provides HAUP system procedures to read users’ HW and recognize users’ behaviour when using the traditional username and password authentication.

4.1 HAUP Framework

When following the authentication process, the client uses the application to contact the server. The server responds by sending the security requirement; for example, using the Secure Socket layer (SSL) and Digital Certificate. This procedure secures the authentication identity during the communication session. For example, using an encrypted method (such as MD5) and private and public keys to secure every log-in session between the client and server.

The HAUP framework in Figure [4.1](#) builds on the current username and password and include HW information and the

user's behaviour in the authentication procedure as the user's profiling procedure. The HAUP framework has an additional controller component in the client side which has two main procedures. The first procedure reads the user's HW information which is the "HW observer" and observes the client's behaviour when typing the traditional username and password keys using a "behaviour observer". This identity information is associated and encrypted to be sent as 'log-in identities' with the username and password' through the network.

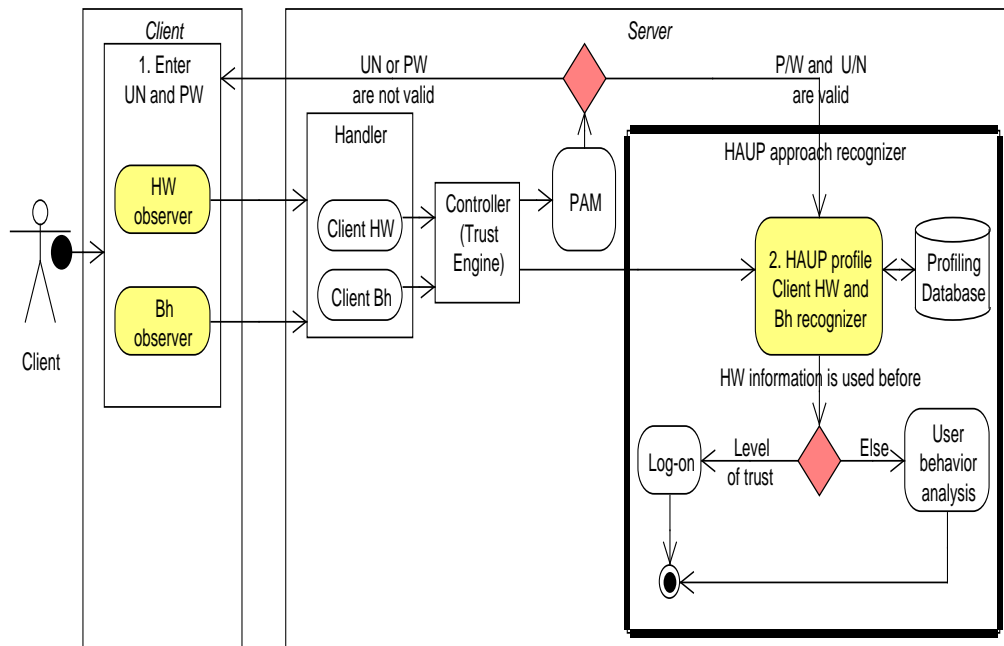


FIGURE 4.1: HAUP framework

After the security certificate procedures are applied and the client's identities reach the server, the HAUP framework in the

server side receives the client's log-in request which includes three identifiers username and password, HW information and client's behaviour. A handler component in the server side then receives every log-in request separately. For example, using a thread in java script in the server to serve more than one client at the same time. Then, the handler component sends the client's identity to a trust engine component controller to manage the authentication keys.

The trust engine component controller sends the username and password to 'Pluggable Authentication Module' (PAM) components in order to check if the username and password are valid or not. If they are not valid, the system sends an 'incorrect password' message to inform the client and ask the client to re-enter a valid password and/or username. If the username and password are valid, the engine controller sends the client's HW and Bh to the "recognizer" components. This recognizer component determines the client's patterns from the profiling database component using a profiling component which searches for the previous client pattern in the same received HW information.

If the HW has not been used by the client, this means the client did not use the current HW previously and the client pattern is not observed. In this case, the HAUP framework redirects the client to another verification question or approach. If the HW has been used by the client and is found in profiling

database, the profiling component observes the client pattern from the previous usage. Finally, the trust engine controller performs a comparison by profiling the current client behaviour and profiling the previous pattern based on the same HW and calculating the similarity between the client behaviour and pattern. This calculation shows the level of trust based on the HAUP authentication factors.

4.2 Main HAUP System Components and Architecture

The HAUP system depends on two components in the both server and client sides to profile the client during the log-in procedure in the client's devices. The first component captures and observes the client's behaviour and HW from client side. This information is encrypted and sent to the server side. Figure 4.2 shows the high level HAUP architecture including components.

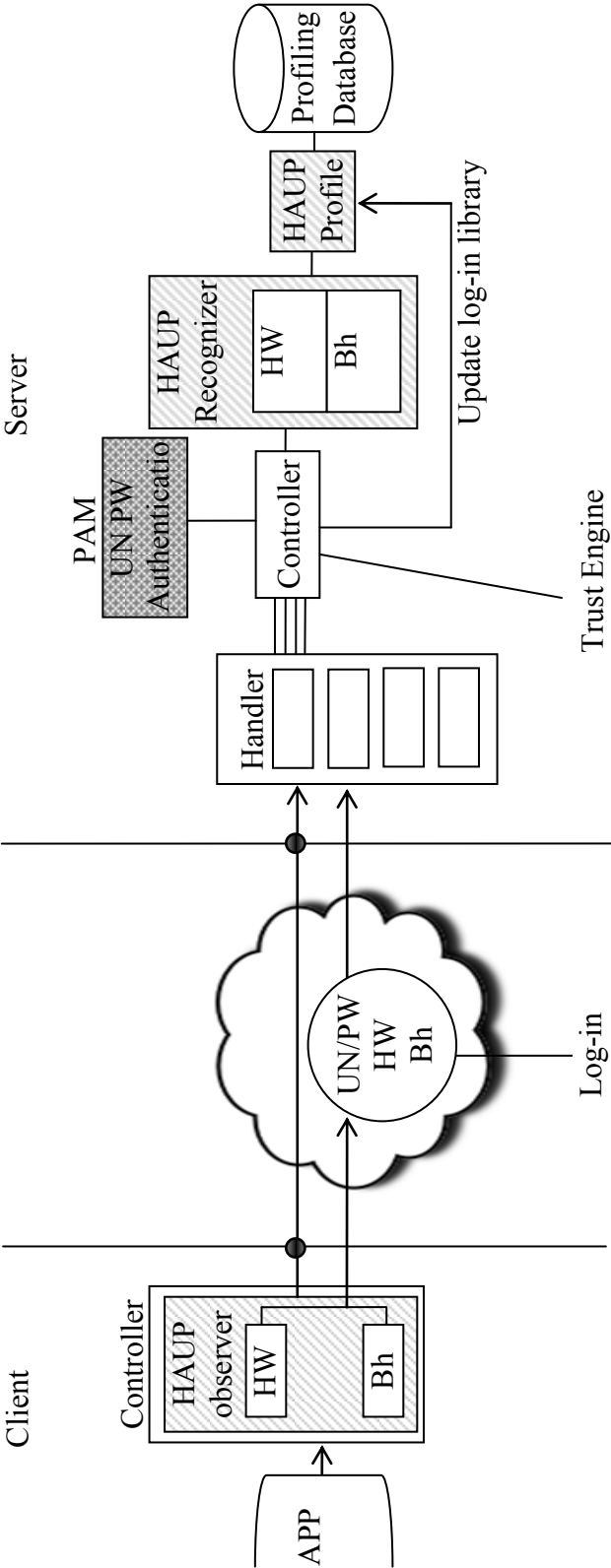


FIGURE 4.2: High Level HAUP Architecture

In the server side the HAUP component carries out profiling component checks to establish if the client used the current HW before by accessing the client's profiling database. If so, this component analyses the relationship between the current and the previous client's HW. The second component relates to the level of trust and here the component observes the similarity between the current client behaviour and the previous client's pattern when the client used the current HW in a previously successful log-in attempt. This component determines the level of trust by highlighting the relationship between the client's behaviour and pattern in respect of the 'HMSPNs' environment.

Figure 4.3 shows details of HAUP system architecture and declares recognizing client's behaviour procedures when the traditional username and password checking procedure is in progress.

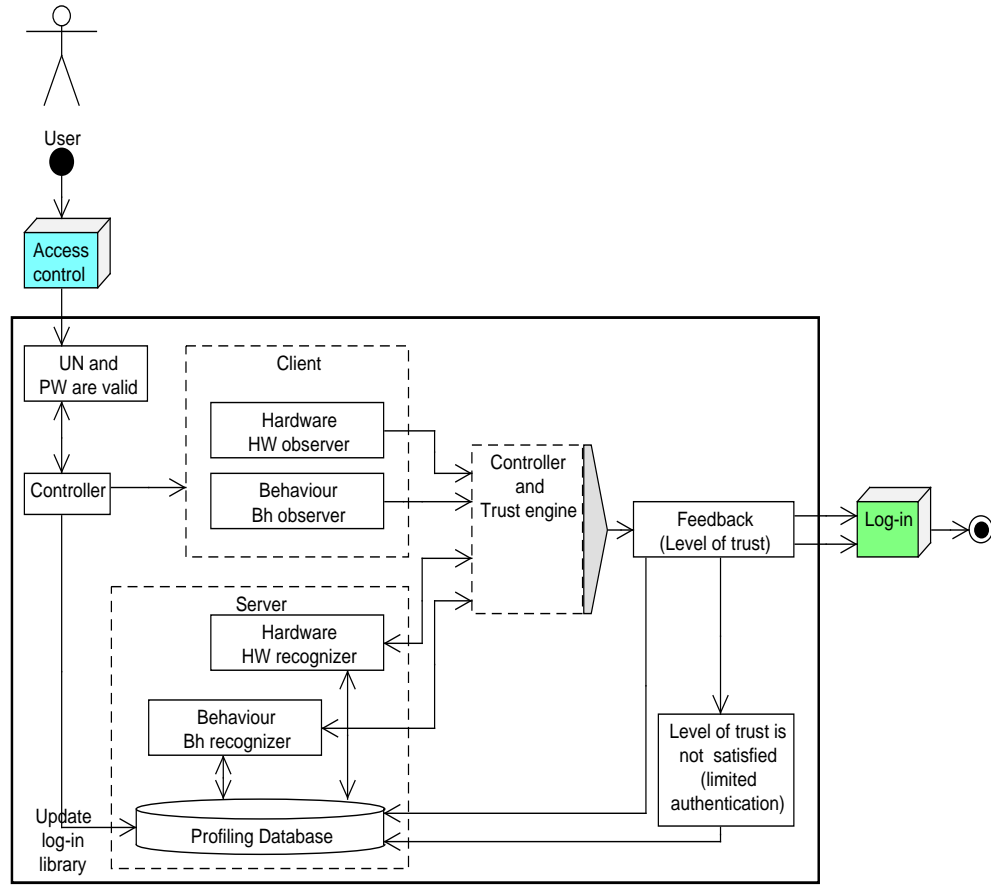


FIGURE 4.3: Details of HAUP System Architecture

4.2.1 HAUP Recognizer

The HAUP process takes into account previous HW usage and client patterns over time and also considers other aspects such as concurrent usage of the same HW configuration in different log-in processes which, for example, could indicate a spoofing attack. HAUP system architecture process requirements are:

1. Trust level based on usage of HW configurations.
2. Known HW configurations for use in behaviour recognition (or matching configuration).
3. Cross log-in analysis for unauthenticated detection.

The trust level is computed against the history of previous log-in-attempts and their associated HW configurations which is drawn from the sequence of previous successful log-in attempts by this client.

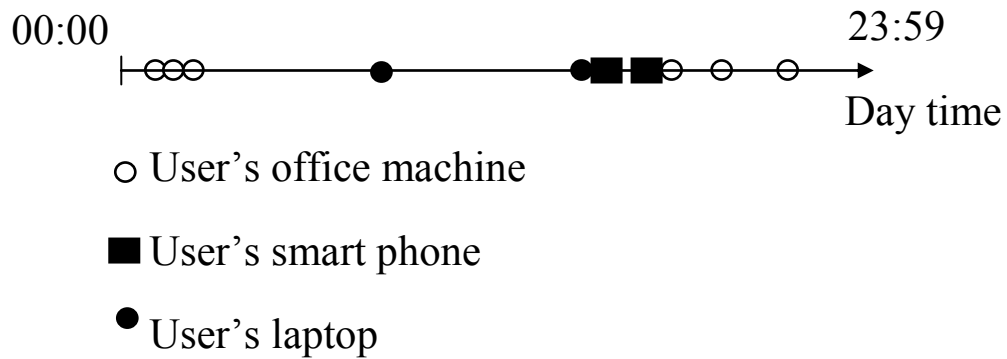


FIGURE 4.4: HW History During the Day

Figure 4.4 shows a simplified example. Every node on the timeline represents a successful log-in by the client in question. The HW configuration that is used by particular client during the day is depicted by the shape of the node, e.g. the empty circle is the client's office machine, the square is a mobile device and the filled circle is the client's home computer.

1. The first step is to decide whether the HW has been used before. This is important for the keystroke recognizer in subsequent checks as it establishes a baseline trust for the access in case the HW is known.

2. Secondly, the access is viewed in the context of the other accesses (left neighbours), the time and the day of the access. We chose metrics based on the time of day and the day in the week as these constitute the majority of repetitions we have encountered during the HW analysis usage. Currently HAUP system architecture doesn't support more complex analysis of these events in the HAUP prototype, but envisage the use of neural networks or support vector machines to establish a behaviour baseline against which the check can be performed. Based on the "fit" of the HW configuration used in the log-in, the trust level is adjusted.

3. Thirdly, the HW recognizer maintains a cache of recent and current log-in activities over the entire client-base. If there is a current log-in from the same HW configuration or configurations that share particular HW components there is a chance that one of the log-ins could be fraudulent and based on spoofed HW information. It is known that some HW manufacturers fail to provide unique serial numbers for their components. For the known cases there is a blacklist of manufacturer IDs which are excluded from this analysis step. A collision of using another

client's HW here reduces the trust level established by the HW *HW recognizer*.

4.2.2 Client Behaviour recognizer

The HW client behaviour recognizer considers the press and release times as a proof of concept and does not use other correlations between subsequent key press events that may be further improving the accuracy. As the contribution of this research is not a novel keystroke recognition scheme however is the integration of multiple approaches, this mechanism can be replaced with more sophisticated techniques such as specific keystroke recognition [69].

The keystroke recognizer takes the current keystroke behaviour entered by the client behaviour and matches it against the previous recorded keystroke behaviour of that client using that HW.

In the HAUP system, a keystroke pattern is characterised by the press and release times of the keys that are used in entering the username and password and is gathered on the client side. Figure 4.5 gives an example of such a pattern.

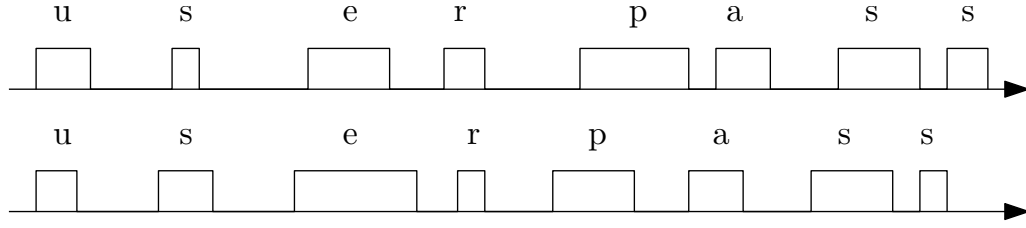


FIGURE 4.5: client's Keystroke Patterns in Particular Hardware

In this research the HW authentication builds trust-metrics which are based on whether the current keystroke pattern fits the users profile information where the profile is created based on the previous user inputs. For example, with respect to Figure 4.5 the first key event is the time the letter “u” is pressed. Previous log-ins, e.g., the recorded times in Table 4.1 which forms the user profile, as depicted in Figure 4.6.

TABLE 4.1: Keystroke Profiling Against HW Configuration

#	1	2	3	4	5	6	7	8	9	10	11
u↓	10	8	9	11	15	8	10	8	11	6	12
u↑	10	8	9	11	15	8	10	8	11	6	12
s↓	6	5	7	8	9	6	7	6	8	5	8
s↑	15	10	10	12	20	12	11	10	12	12	10

The HAUP approach looks at the variance of the data and the percentile into which the current keystroke pattern falls with respect to each key press and release event and computes an accumulated trust level over all events contained in the keystroke pattern. In comparison to, for example, specific

keystroke recognition [73]. This is a very simple approach which we plan to refine in the future.

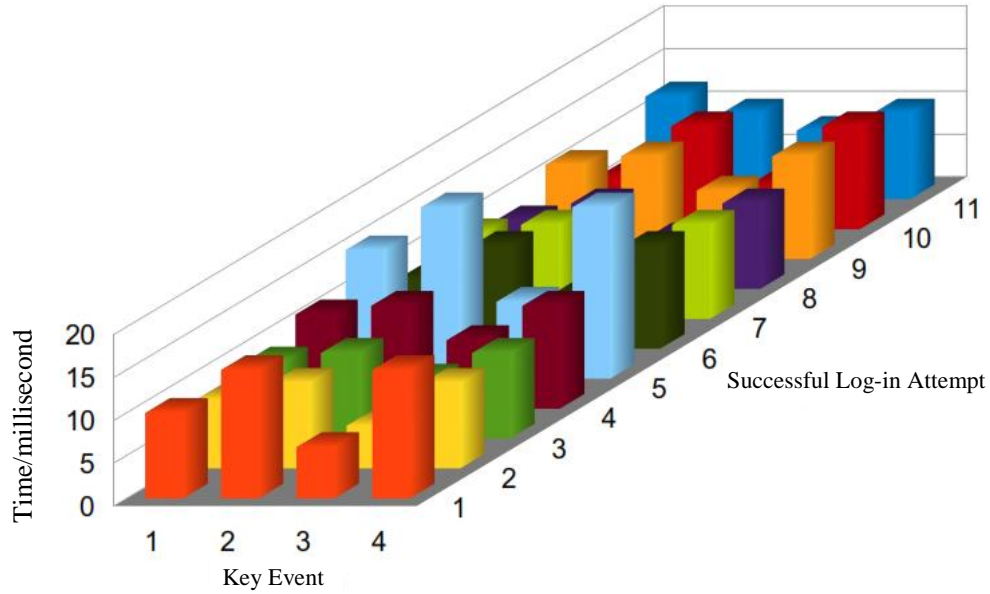


FIGURE 4.6: Profiling User's Keystroke Behaviour

4.3 HAUP Authentication Process

The HAUP system has two procedures in the client side when the traditional username and password log-in procedure is in progress. Whilst the user 'u' is typing the username and password the first procedure captures the user's behaviour by calculating the keystroke (both key press and release) speed when the username and password are typed. The second procedure collects the HW information which consists of the user's current

HW configuration. As the security software installed on the client machine can prevent the gathering of HW information, the HAUP system considers this to be optional information. However, if this information is not provided, it has a detrimental effect on the accuracy of the HAUP mechanism as the HW profiling information is coupled with the selection of the user profile for keystroke recognition.

In the server side the HAUP has two additional procedures. When the log-in identity is received and approved as valid in the server side, the first HAUP procedure checks if the HW has been used by the current user or not. If used, the second procedure recognizes the similarity between user's behaviour and pattern based on the HW usage and calculate level of trust for the user. If the HW is new for the HAUP system or is unknown, HAUP redirects the user for another verification approach. Figure 4.7 illustrates the HAUP procedures and interaction [133].

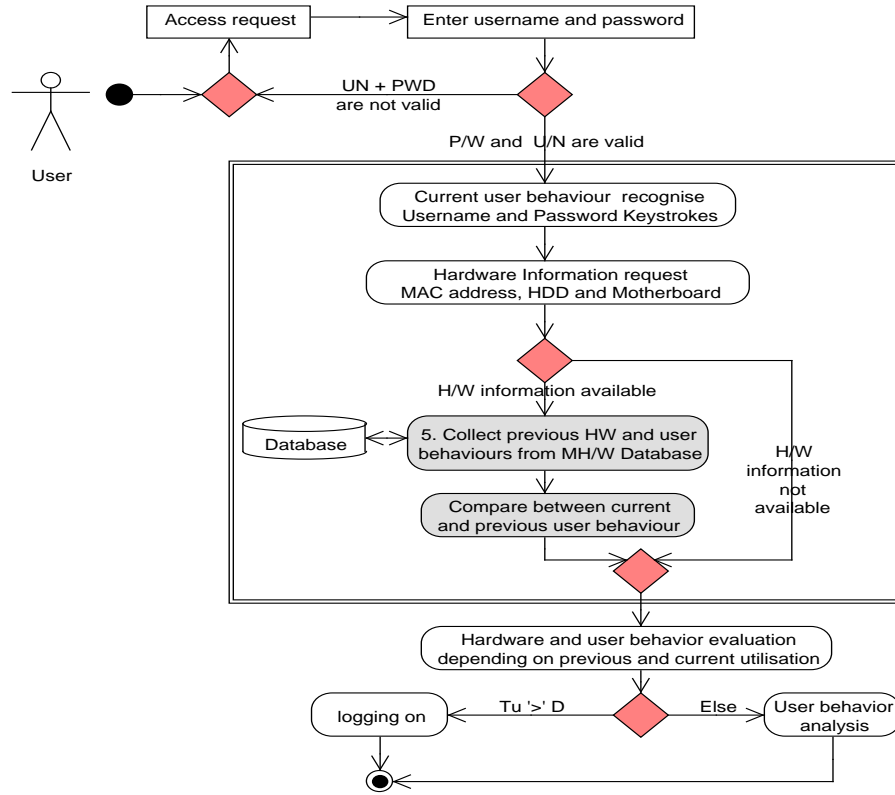


FIGURE 4.7: General Overview of HAUP Flow Diagram between System Components

If the user provides access to the HW profile, the HAUP system begins to analyse and compare the previous HW configuration with the established profile of that user is used in current successful log-in attempt to determine their similarity. If the user has used the current HW before, the HAUP system computes the similarity between the current keystroke behaviour of the user and in particular the HW that has been recorded against this HW configuration previously.

Given that the username and password checks are successfully passed, the HAUP system computes the similarity between the HW configuration and the associated keystroke behaviour similarity to profile two levels of trust. If only the keystroke information is available then only one level of trust is being used. Similarly, the keystroke behaviour is evaluated and linked against the used HW configuration if available.

The HAUP system authenticates normally if the username and password are correct and a threshold in both levels of trust is passed. If the username and password do not match, the authentication is considered as failed. If the username and password are correct and only a low level of trust is established based on the HW or keystroke behaviour, the system can be configured to adapt to the level of trust. For example, the authentication can be failed; the user can be authenticated with reduced privileges such as only being able to view his account details; the HAUP system can increase the threshold for an intrusion detection system that identifies fraudulent activity based on the transactions that are undertaken or even redirect the user to a honey pot trapping system to explore if the user is a hacker using a spoofed username and password. In an e-banking context, this could mean to delay the transactions and attempt to contact the user via a different media such as email or phone.

4.4 Summary

This chapter introduced the HAUP framework in section 4.1 including the main components to implement HW authentication approach. This framework achieve the research objective O.4. (Develop a framework that supports the mathematical model) by detecting the system components and architecture that provide the contribution C.4. (Demonstrate the feasibility of the approach by implementing a set of experiments) by providing feasible HAUP framework that implement this work and collect data from set of experiments. Section 4.2 determined and explained the HAUP architecture and included the HAUP system components which are required to implement the HAUP prototype based on HW characteristics. Section 4.3 presented the mechanism of HAUP components to be implicated in the traditional username and password authentication including the flow digram procedures that clarified answer for the research question Q.5.(What is the added cost of a profiling approach? How can costs to be avoided?) by noting the HW information affect to implement the approach.

The next chapter provides a formal model to define HAUP profiling mathematically and support the HAUP authentication process to build a level of trust.

Chapter 5

Mathematical Model of HAUP

Objectives

1. Provide model description.
 2. Explain the domain.
 3. Present the formal model.
 4. Describe the backward propagation.
-

This chapter provides the mathematical model that defines the HAUP approach. This mathematical model determines the authentication factors to be calculated and combined in the level of trust. Section 5.1 provides assumptions and analysis of the domain for hardware profiling. Section 5.2 provides an illustrative example using hardware information to explain the mathematical model. Section 5.3 provides the mathematical modelling for user behaviour illustrating the hardware influence in recognizing users behaviours. Sub section 5.3.1 describes the neural network for the analysis of user behaviour.

5.1 Formal HAUP Analysis

The HAUP trust function is based on a data model for comparing profiling information obtained in previous and current log-in procedures. In the HAUP approach, hardware information and user behaviour are considered the main authentication factors.

5.1.1 Domains

The following declares the domain model:

- 1- Trust is modelled as a weighted function T that maps from a log-in context $l \in L$ and an existing hardware profile

$\alpha \in A$ and Behaviour profile $\beta \in B$ to a value in $[0, 1]$. So, $T : L \times A \times B \rightarrow [0, 1]$ and U is user who perform log-in procedure.

2- S is sequence of log-in attempts l , initially $S = \{\}$. So, $S = \{l_1, l_2, l_3, \dots, l_n\}$. Every log-in has four factors $l = \{p_i, c_i, b_i, t_i\}$ where i is log-in sequence index. So, the user has previous log-in attempts and $l_1 \dots l_{n-1}$. The current log-in attempt is l_n . The factors are: p_i user's username and password in current log-in attempt using c_n by specific behaviour b_n at particular time t_n .

$$S = \{p_1, c_1, b_1, t_1\}, \{p_2, c_1, b_1, t_2\}, \dots \{p_n, c_n, b_n, t_n\}$$

S_c is the sequence of log-in attempts using particular configuration c

$$S_c = \{l \in S | l.c = c\}$$

Similarity $S_{c,p,t}$: is the sequence for user $u \in U$ used c configuration in successfully log-in at particular time t . These authentication factors has similarity characteristics about the user activity. The similarity of this factors in every successful log-in attempt can be used to profile the user and determine the access control threshold.

3- w_α and w_β are weights that can be chosen to vary the influences of Hardware Trust and Behaviour Trust.

$$\# 1) \quad T(\ell, \alpha, \beta) = \begin{cases} T_\alpha(\ell, \alpha) w_\alpha + T_\beta(\ell, \beta) w_\beta & \text{if } p \text{ matches} \\ 0 & \text{else.} \end{cases}$$

Where $w_\alpha + w_\beta = 1$

Let T_α be a function modelling Hardware Trust, and T_β be a function modeling Behaviour Trust.

$$T_\alpha : L \times A \rightarrow [0, 1]$$

Mapping from a log-in context $l \in L$ and a hardware profile $\alpha \in A$ to a trust value in $[0, 1]$

$$T_\beta : L \times B \rightarrow [0, 1]$$

Similarly T_β is a function modelling Behaviour trust, mapping from a log-in context $l \in L$ and a behaviour profile $\beta \in B$ to trust value $[0, 1]$.

5.1.2 Hardware configuration

Every computer is assumed to have particular hardware (HW) parts. HW is a set of all possible hardware in user machine.

Let C be the set of all possible configurations and define a configurations to be three of distinct HW . In HAUP, three hardware parts are considered (Network card part number, Hard disk number and BIOS number).

$$c = \{\langle m_i, m_j, m_k \rangle : m_i, m_j, m_k \in HW \wedge m_i \neq m_j \neq m_k\}$$

We define the similarity function to determine the similarity between two configurations c and \bar{c} that are used in log-in l procedures. However, number of HW parts can increase in the future work that discriminates the user behaviour in particular HW Characteristics. So,

$$\text{If } c = \{m_i, m_j, m_k\} \text{ and } \bar{c} = \{\bar{m}_i, \bar{m}_j, \bar{m}_k\}$$

We assume $d = 3 - |c \cap \bar{c}|$, d is the difference between the two configurations.

And if c and \bar{c} were same then $d = 0$

However, if c and \bar{c} were not same then, $d = 3$, which is $(c \cap \bar{c}) = \emptyset$

5.1.3 Sequential Log-in Context

U_c is set of uses that used C in log-in successfully.

$$U_c = \{l.u | l \in S \wedge l.c = c\}$$

$|U_c|$ is total number of using c in successful log-in by same user.

An important matrices for HW trust is how often c was used by u among all users of c . This is computed by

$N_{u,c} = |\{l \in S | l.u = u \wedge l.c = c\}|$ and is used as relative to the total numbers of usages of c , N_c

$$N_c = |\{l \in S | l.c = c\}|$$

5.2 Illustrative Example

Following analysis example of filtering HAUP factors. Filtering map in log-in L sequence S which begin by check the “*username/password*”. Then HAUP checks the \bar{c} including log-in time t . Finally the user behaviour \bar{b} is checked. The result of this analysis should be related to current users log in attempt which has four weighted factors $S_n = \{\bar{p}_n, \bar{c}_n, \bar{b}_n, \bar{t}_n\}$. Following procedure declares every filtering target in this example:

When the user log-in to HAUP approach, the log-in factors is filtered by the username and password which noted by p factor.

$$S_p = \{l \in S | l.p = p\}$$

If the user uses more than one HW during the day/week, HAUP approach filters the log-in based in HW factor which is used in current successfully log-in attempt \bar{c} . So, HAUP explores user pattern if the hardware been used in successful log-in attempt before. If so, that mean $N_c > 1$ which three

main hardware parts information from current hardware are used before in successful log-in attempt by the same user.

$$S_c = \{l \in S | l.c = c\}$$

If the user has particular behaviour at particular time during the day/week based on particular hardware, HAUP approach filters the log-in based in behaviour factor. So, if the user's behaviour in current log-in attempt has similarity to previous user behaviour in using same hardware which means $(b_u \cap \bar{b}_n) = t_u$. In addition, if analysing and capturing the user behaviour in particular time during the day, that means the user is using same hardware at particular time. That is

$$S_t = \{l \in S | l_t = t\}$$

So, HAUP approach have three factors weight based on username and password validation factor.

$$S_{\bar{p}_n, \bar{c}_n, \bar{b}_n, \bar{t}_n} = [0, 1] \dots \dots \dots (1)$$

Figure 5.1 shows a sequence of HW usage and how this affects the user's level of trust. Figure 5.1 shows the first checking point which is exploring whether the user used the current HW (\bar{c}_u) in previous successful log-in attempts (c_u) by find out the similarity between them. If the HW is used, the HAUP approach compares the user's behaviour (\bar{b}_u) during the current successful log-in using current HW configuration from previous usage b_u . So, if user's behaviour is similar to the user pattern,

the approach has profiled the user. Furthermore, to recognize the user, the HW change during the day time should be recognized as part of the profile of user HW.

When the user keeps using particular group of HW the user will have specific behaviour when every particular HW is used. This familiarity discriminates the user behaviour in particular HW, e.g., the user has keystroke pattern in using particular keyboard. So, recognizing user behaviour in using specific HW provides profiling for the user during the day/week. This profiling is support to trust the user when the user has unique pattern in using particular HW. For example, every user has particular pattern in using mobile touch screen and has another pattern in using desktop keyboard.

The HAUP approach merges HW usage, as explained in Figure 5.1, with biometrics behaviour and log-in times during the day/week. This yields a HAUP approach in which the user's biometrics can be correlated with the HW that is used during log-in process.

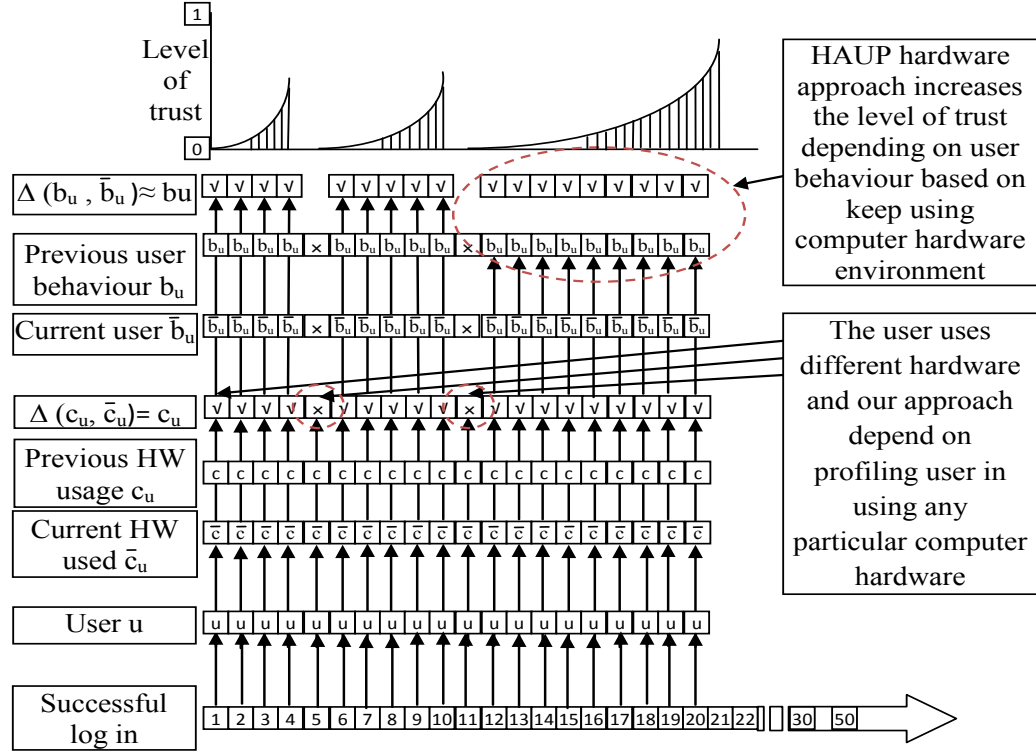


FIGURE 5.1: How Using HW Information in Trust User

5.2.1 Limitations

If a user has successful log-in attempt using particular hardware then

$$S_{p,c} = \{l \in S | l.p \wedge l.c = c\}$$

If user's hardware is unknown to be recognized, then the user is expected to have same behaviour at particular time during the day which is considered the user is using particular hardware at same time during the day.

$$S_{p,c,b} = \{l \in S | l.p \wedge l.c = c \wedge l.b = b\}$$

If user's hardware and behaviour is unknown to be recognized with previous user's pattern at any time during the day/week, then the user can not be trusted using HW authentication approach.

5.3 Behaviour Trust Based on Hardware Information

User pattern is keystroke behaviours b in typing every single log-in keys in successfully log-in attempts.

$$b_u = b_{u,1}, b_{u,2}, b_{u,3}, \dots, b_{u,n}$$

User behaviour in current successfully log-in attempt \bar{b}_u is keystroke behaviour r in typing every single log-in keys.

$$\bar{b}_u = \bar{b}_{u,1}, \bar{b}_{u,2}, \bar{b}_{u,3}, \dots, \bar{b}_{u,n}$$

The relation between previous pattern and current user's behaviour in r is:

If \bar{b}_u in b_u We say $\bar{b}_u \in |b_u|$

If $(b_u \cap \bar{b}_u) = b_u$ or $(b_u \cap \bar{b}_u) \approx b_u$

We say, b_u and \bar{b}_u are typical

$$b_u = \{l.b | l \in Sl.u = u, l.p = up\}$$

$$b_u \cap \bar{b}_u = (|b_{u_r1} - \bar{b}_{u_r1}|)^2, (|b_{u_r2} - \bar{b}_{u_r2}|)^2, \dots, (|b_{u_rn} - \bar{b}_{u_rn}|)^2$$

$$b_u \cap \bar{b}_u = \sum_{i=1}^n (|b_i - \bar{b}_i|)^2$$

n is number of username and password characters

However, if the user behaviour is related to particular c_u , so profiling user behaviour b_u is based on particular c_u .

$$T_\alpha \cap T_\beta = b_{c_u} \cap \bar{b}_{\bar{c}_u}$$

$T_\beta(l, b)$ is represented by a neural network for each P and c . One neural network for each p and c by feeding users behaviour (keystroke) speed. The typical matching gets higher weight which is closed to 1 and the completely different matching gets lower value which is closed to 0.

5.3.1 Back Propagation Algorithm

The important reason for using back-propagation algorithm was that, it was considered as a supervised learning algorithm which is used to learn user keystroke pattern. It was used for multi-layer perceptions to change the respected weights that were connected with the total hidden neuron layers. This algorithm used the computed output errors to update the weight

values in backward direction. For retrieving the total error, forward propagation was done earlier. During the forward propagation the neurons would be (activated function) as shown below:

$$f(.) = 1 / ((1 + \exp(-X)))$$

Where:

X - The input.

exp - exponential

The Back Propagation algorithm worked based on the following 4 steps:

1- It performed forward propagation phase with respect to the input pattern and calculate the error output.

2- Changed all weight values of each weight matrix using the formula.

$$W_{k+1} = W_k + (Error \times O)$$

$$V_{k+1} = V_k + (Error \times O)$$

where :

k is number of iterations.

$k + 1$ = next iteration.

O - output of the network

W - Weight for input layer.

W_{k+1} weight at $k + 1$ time

V - Weight for hidden layer.

V_{k+1} weight at $k + 1$ time

3- Repeated step 1

4- This process of algorithm ended once all the out patterns match their target patterns. This process required a time stamp to calculate all of measurements in the neural network. this time is called number of Epoch.

Back propagation procedures

1- Collect error of output neurons: $E = O(1 - O)(d - O)$ where d is the desired signal (user's keystrokes behaviour)

2- Changes output layer weight.

$$W_{k+1} = W_R + \lambda EO$$

$$V_{k+1} = V_R + \lambda EO$$

where λ - is learning rate.

3- Calculate (back-propagate) hidden layer error.

$E_h = O(1 - O)(EW_k + E\Delta w_k)$ where Δw_k is the change of the weight with respect to time = dw_k/dk

where E_h is the error in hidden layers

4- Change hidden layer weight

$W_{k+1} = W_R + \lambda EX$ where X: is users response time in keyboard pressing

$V_{k+1} = V_R + \lambda EO$ (output for input layer)

where

W_R is the response input weight.

V_R is the response hidden weight.

Graph 5.2 clarifies neural network procedures and hidden layer.

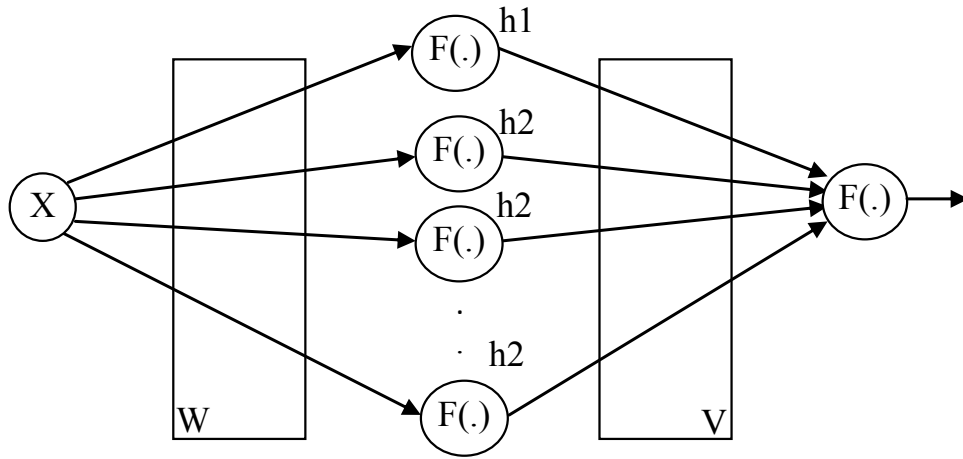


FIGURE 5.2: neural network procedures and hidden layer.

This procedure is used in chapter 7 to explore the hardware influence in profiling user behaviour. This evaluation enhanced hardware information to be authentication factor for profiling proposed. So, this value is express user behaviour matching when the value is closed to 1. In contrast if the value is closed to zero that is considered user's pattern and behaviour are not same.

5.4 Summary

This chapter presented a formal analysis and assumption to provide HAUP procedure mathematically in section 5.1. This mathematical model is using the similarity between user's current hardware configuration \bar{c} and previous configuration that is used in any previous successful log-in attempts. This is exploring the intersection between current user behaviour \bar{b} and previous user pattern b in log-in keys (username and password keystrokes) at particular time when the log-in attempt is successful. Section 5.2 illustrate the similarity between user's HW and behaviour by giving illustrative example using hardware and user behaviour factors in HAUP mathematical model based on given weight of trust for HAUP factors. This weight factors answers the research question Q.3. by combining HW information factor weight with traditional "username and password" in mathematical model to provide authenticate and profile the

user. After that, section 5.3 provides mathematical modelling for user behaviour in using particular hardware at particular time that addressing the research contribution C.3. by providing a mathematical model for trust. Finally, section 5.3.1 clarifies using neural network function to analysis user behaviour when two different hardware are used by same an different users.

Equation #1 provides level of trust to authenticate the user. The level of trust is between zero and one. This level of trust is closed to one when intersection between user behaviour and pattern has more similar factors. In contrast, the result is closed to zero when similar factors are less.

The following chapter will provide the practical steps of building HAUP prototype and clarify technical procedures to deploy HAUP software.

Chapter 6

Implementation

Objectives

-
1. Establish HAUP software requirements.
 2. Explain HAUP system interaction.
 3. Implementing HAUP.
-

This chapter describes the HAUP authentication prototype to develop the HAUP approach as MFA based on the traditional username and password authentication procedure. Section 6.1 provides a technical HAUP authentication scenario which is running in traditional user-name and password authentication in the client device. Section 6.2 analyses HAUP program codes that are running with traditional username and password to profile users' hardware and observe user behaviour. Then, section 6.3 provides HAUP analysis methods to present the relationship between user behaviour and previous users' patterns in typing the successful authentication keys using particular HW. Finally, section 6.4 provides the HAUP analysis steps in a sequence diagram.

6.1 Hardware Authentication Scenario in Access Control

In a HW authentication scenario, the HAUP prototype monitors users' behaviour in the client device whilst the user is typing the authentication keys and reads HW information (See chapter 4.2). This information is sent in an encrypted form by hash function algorithm [134] with the username and password information from the user's device to the server to analyse the log-in attempt. After that, HAUP checks the authentication

keys username and passwords and HW information by searching in the HW log-in database to determine if the user used current HW in any previous successful log-in attempts.

The HAUP prototype collects HW information “*HMSPNs*” from user devices. The HAUP prototype assumes some HMSPNs but no specific HW parts. User behaviour is monitored and collected from user devices. HW and user behaviour information is used to evaluate the HAUP authentication approach to show HAUP authentication analysis and compares HAUP results with current authentication approaches. If a user has used current HW before, every particular time and user pattern is observed based on this HW by the HAUP prototype. The HAUP prototype calculates the similarity between the previous user’s pattern and current user’s behaviour by using keystroke profiling. This similarity in the HAUP prototype focuses on keystroke speed in the authentication keys username and passwords. This profiling supports the authentication decision to give a user a particular level of trust. Figure 6.1 shows the HW authentication scenario between user’s device and the server.

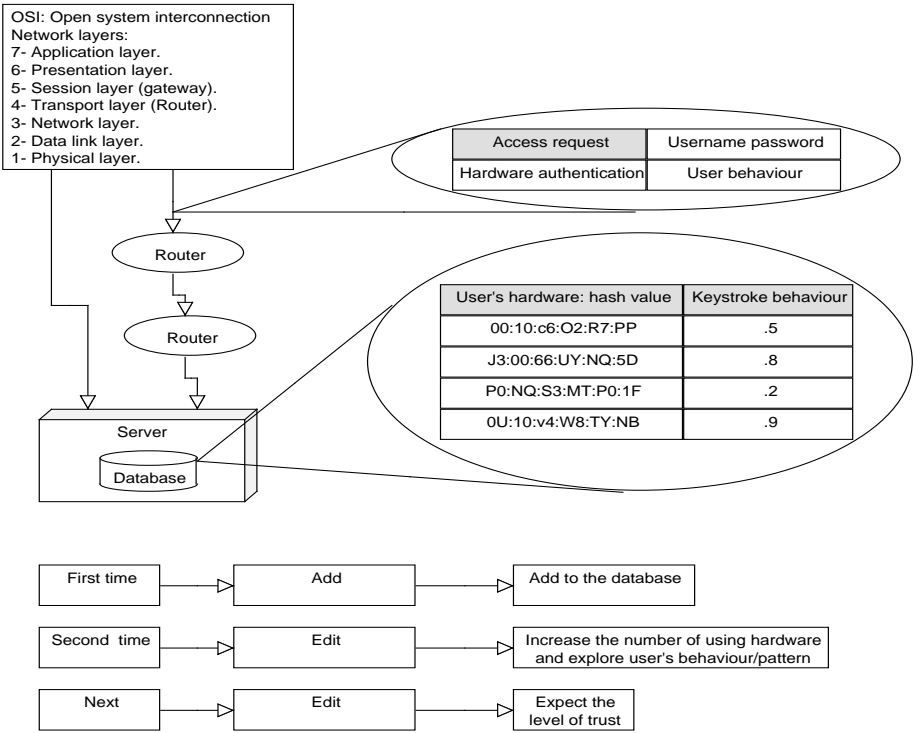


FIGURE 6.1: System Scenario

6.2 System Procedures

With every successful log-in attempt the HAUP prototype captures user behaviour and collects HW information from the user's device. The collected HW information are “*MAC address, Storage media and BIOS*”. Following this the HAUP prototype analyses the user's HW information by searching in the user's HW database in the server side and explores if the user has used the current HW parts before. After that, the

HAUP prototype compares the similarity between the current user's behaviour and previous user's patterns based on using the same HW. However, if a user's HW was not determined in the current log-in attempt the HAUP prototype determines the log-in time to analyse user patterns in the previous log-in based on the particular time during the day. This analysis declares the relationship between the user's behaviour when the log-in took place at that particular time. This comparison provides a particular level of trust for the user.

As result of analysing the HAUP procedures, the HAUP prototype collects and analyses HW information using the following procedures:

P.1. Profiling the user's keystroke behaviour when the username and password keys are typed by determine the response time between key-press and key-release as part of the process of capturing biometric behaviour (See Section [4.2.2](#)).

P.2. Profiling the user's HW by reading HMSPNs from the user's devices as part of the process of capturing user behaviour (See Section [4.2.1](#)).

P.3. Collecting the user's HW and observing user behaviour from the client side then sending the encrypted results to the server side.

P.4. Analysing the user's HW in the server side by exploring how often the user used current HW before user and recognizing user behaviour (keystrokes speed in typing the log-in keys) in the user's device.

P.5. Determining the log-in time that support to recognize user's HW that is used and the time of using a particular HW during the day/week. This time stamp motivates to recognize user behaviour and refer the user behaviour to particular HW.

To collect HW information the HAUP prototype requires amending the user operating system by sending and implementing particular file in client device. This file is suitable with users operating system. A HAUP relationship diagram can work on any method of physical storage, whether it be disk, CD or USB as they can all be used for storing data. Figure 6.2 shows general diagram of HAUP authentication procedures, (1) enrolment, (2) identification and (3) verification. In contrast, user behaviour implements a particular function to calculate the time response of the user's keystroke when entering the username and password in the log-in form. Both reading HW and monitoring user behaviour implement a visual analysis to show user's behaviour. This analysis shows the level of trust in the user based on the current HW.

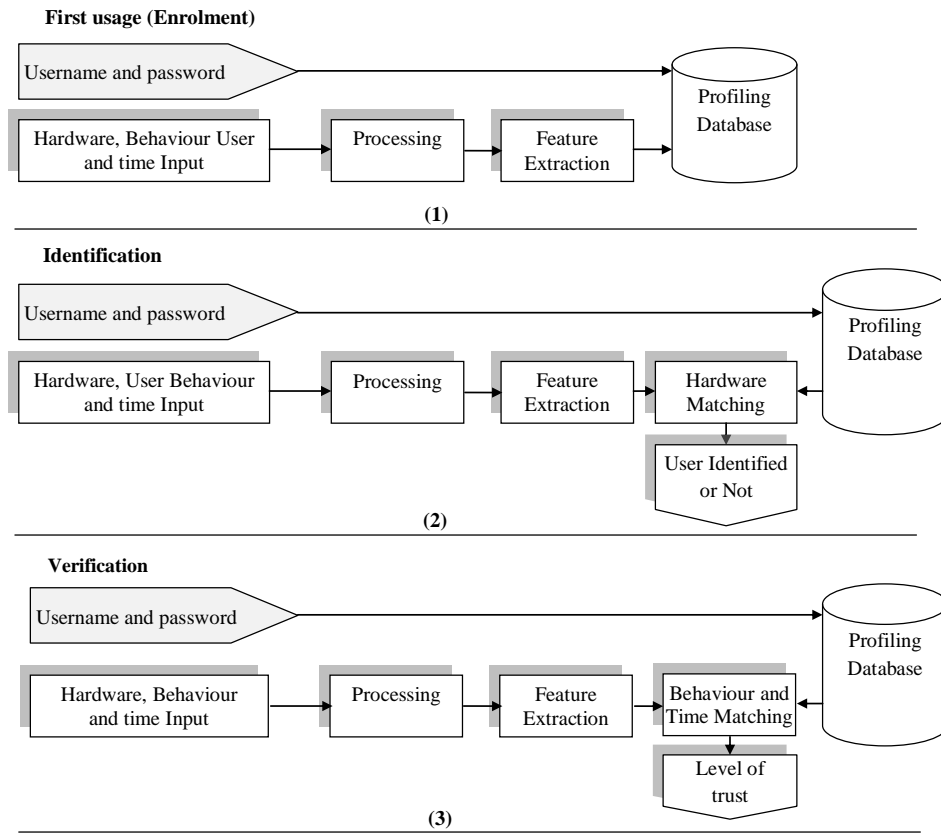


FIGURE 6.2: Diagram of HAUP Procedures

6.3 HAUP Analysis

The HAUP authentication technique depends on the matching of the current HW against the user's previous HW usage the associated user's behaviour against the previous user's behaviour as part of the log-in procedure.

At the client side the log-in prompt performs three data-collection functions. Firstly the username and passwords are collected in the traditional way. Secondly, the keystroke behaviour of the user is gathered and saved during the typing of the username and password. Functions like auto-completion and the 'copy & paste' functions are turned off, as they would effectively disable the recognition of the keystroke behaviour. Thirdly, during log-in, HAUP reads the HW configuration from the user's operating system. This requires the user to download the log-in software or the server address from which the log-in prompt is loaded.

On the server side the HAUP checks the username and password hash against the stored credentials. If this is successful, the additional two components - HW recognizer and keystroke recognizer - are invoked to further validate the log-in request, thus providing additional scrutiny. The HW recognizer checks the database to establish whether the user has used before. If the user has used before, the system determines the similarity between the current keystroke behaviour and the previous keystroke pattern.

6.3.1 Sequence Diagram of Behaviour Modelling

Beginning from the early stage of using a particular HW configuration, the HAUP system can ask additional questions for

verification because of the new and unknown HW. Unknown HW is an obstacle which hinders the ability to recognize user behaviour in previous usage. As such, the HAUP system collects HW information and user behaviour for the first time and use additional verification questions. However, in the next log-in the system resumes using use HW authentication. Figure 6.3 illustrates the sequence diagram of first time usage of the HAUP system.

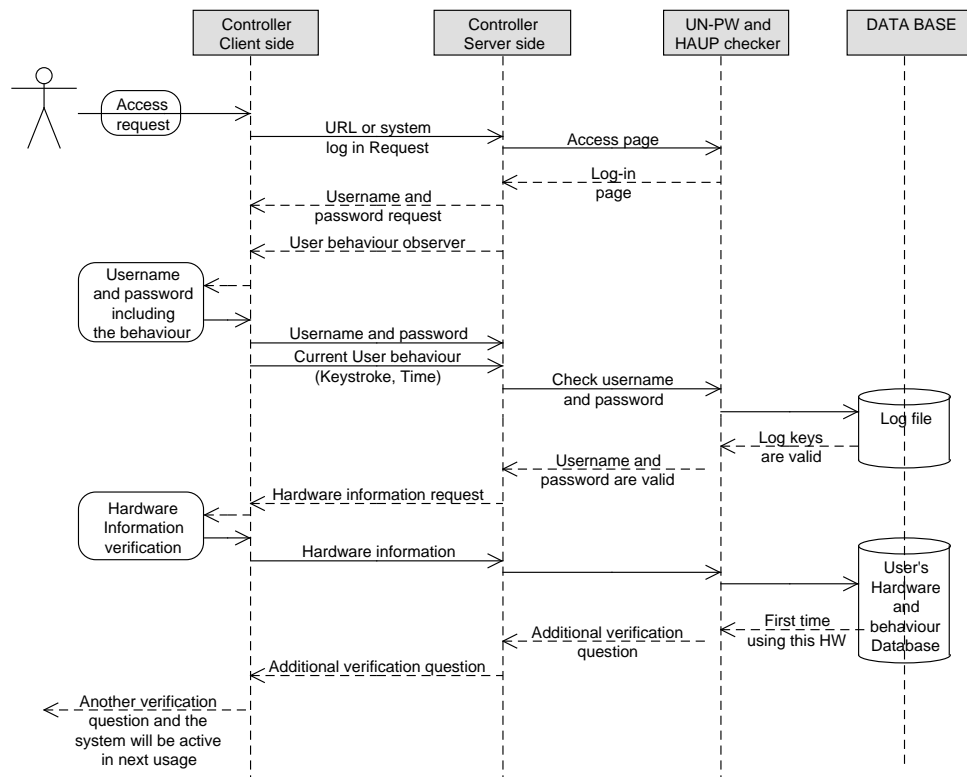


FIGURE 6.3: Sequence Diagram of Using HAUP First Time

The HAUP prototype checks the user HW at every successful log-in. In this way the HW can be tracked during subsequent successful log-in attempts. So, the HAUP approach collects the user's HW and monitors the user's behaviour. At every successful log-in, the server side profiles the user's HW and recognizes the user's behaviour based on the HW information. Figure 6.4 shows the sequence diagram of HW usage when the HAUP when applied by the user.

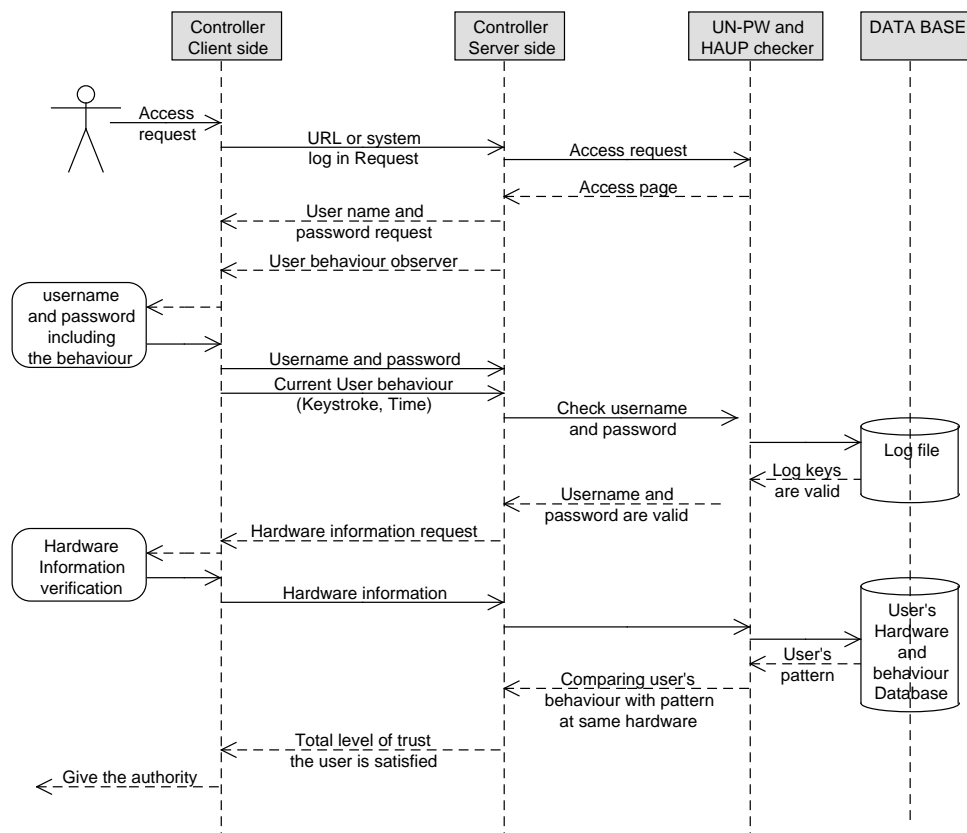


FIGURE 6.4: Sequence Diagram of using hardware After First Time

6.4 System Interaction

As previously discussed, when the user types their username and password in the log-in field, HAUP captures the user's behaviour and calculates the response time between each key press and release of every single keystroke contained within the username and password to observe the user's behaviour at every successful log-in attempt (See Section 6.2). Keystroke response time is calculated by millisecond in order to observe user's pattern however, some programming language can calculate the nanosecond which is faster and can give more details about keystroke that can improve HAUP approach to recognize user's keystroke behaviour. Figure 6.5 shows HAUP code to capture user's keystrokes.

To implement HAUP authentication prototype a Java programming language is type of safe language. This programming language can support different environment application and consider a virtual machine, platform independence, support web and network, network applications support with high performance [135]. However the code access security aims to evaluate HAUP and was not a consideration for the prototype demonstration proof of concept. In addition, in real application the system will consider in system interrupt time in order to observe user behaviour. This development will be used when the system uses nanosecond to recognize user behaviour.


```

336 AccountUsaerNameText.addKeyListener(new KeyAdapter() { private void keyPressed(KeyEvent e) {
337     int keyCode = e.getKeyCode();
338     MiliSecondP = System.currentTimeMillis();
339     if (COUNTER1==0) {MiliSecondTotalrhythm= System.currentTimeMillis();
340     }
341     KeyStrocked[COUNTER1] +=e.getKeyChar()+"";
342     if(COUNTER1!=0){ Res[COUNTER1]= MiliSecondP-MiliScndW;}
343     }
344     private void keyReleased(KeyEvent e) {
345         CurrentLOGINBehaviorInfo[COUNTER1] = System.currentTimeMillis() - MiliSecondP;
346         if(""+CurrentLOGINBehaviorInfo[COUNTER1]+"==null)
347             {CurrentLOGINBehaviorInfo[COUNTER1]=0;}
348         MiliScndW = System.currentTimeMillis();
349         COUNTER1++;
350     private void keyTyped(KeyEvent e) {   });
351     AccountUsaerNameText.addActionListener( new ActionListener() { private void actionPerformed( ActionEvent e )
352     {
353         if (COUNTER1<5)
354             {JOptionPane.showMessageDialog(frame,"Please type your username");
355             AccountUsaerNameText.setText("");
356             AccountUsaerNameText.grabFocus();
357             AccountPasswordText.setText("");
358             COUNTER1=0;}
359             else {AccountPasswordText.grabFocus(); }
360         }
361     AccountPasswordText.addActionListener( new ActionListener() { private void actionPerformed( ActionEvent e )
362     {
363         else {LoginButton.grabFocus();
364         LoginButton.doClick(); } } });
365     AccountPasswordText.addKeyListener(new KeyAdapter() {
366     public void keyPressed(KeyEvent e) {
367         int keyCode = e.getKeyCode();
368         MiliSecondP = System.currentTimeMillis();
369         KeyStrocked[COUNTER] +=e.getKeyChar()+"";
370         KeyStrockedBehaviourSpeed[COUNTER3] +=e.getKeyChar()+"";
371         if(COUNTER2!=16){Res[COUNTER2]= MiliSecondP-MiliScndW;}
372     }
373     private void keyReleased(KeyEvent e)
374     {
375         CurrentLOGINBehaviorInfo[COUNTER2] = System.currentTimeMillis() - MiliSecondP;
376         if(""+CurrentLOGINBehaviorInfo[COUNTER2]+"==null)
377             {CurrentLOGINBehaviorInfo[COUNTER2]=0; }
378         MiliScndW = System.currentTimeMillis();
379         COUNTER2++;COUNTER3++;COUNTER++;
380     }
381     private void keyTyped(KeyEvent e) {   });

```

FIGURE 6.5: Keystroke Biometric Behaviour Capture

The HAUP prototype collects three HW part numbers (MAC address, Storage media and BIOS) in order to profile the user's HW activity. For example, when the HAUP is executed in the Windows operating system environment a physical MAC address is given by following the Java code (Figure 6.6).

```
28 public String getMacAddress() throws IOException
29 {
30     String macAddress = null;
31     String command = "ipconfig /all";
32     Process pid = Runtime.getRuntime().exec(command);
33     BufferedReader in = new BufferedReader(new InputStreamReader(pid.getInputStream()));
34     while (true) {
35         String line = in.readLine();
36         if (line == null) break;
37         Pattern p = Pattern.compile(".*Physical Address.*: (.*)");
38         Matcher m = p.matcher(line);
39         if (m.matches()) {
40             macAddress = m.group(1);
41             break;
42         }
43     }
44     in.close();
45     return macAddress;
46 }
```

FIGURE 6.6: “*MAC Address*” collection from Client HW


The code (Figure 6.6) to collect the user’s HW information (physical address or MAC Address) from their device by obtaining, for example, the motherboard and hard disk drive manufacturer serial numbers. The complete Java code is available in appendix (A). However, the collection code is based on the user operating system such as Windows, iOS and Android.

To recognize the time when a particular HW is used, the HAUP prototype collects log-in times for every successful log-in attempt. The system then analyses this information. For example, the user may use a desktop device whilst at work, a laptop at home and may use a mobile device whilst on the move or in public areas.


To store HW information, the system saves the HW information of users in an encrypted database. The store procedure begins with the first successful log-in using new HW and is named by the user's username, as entered at the log-in. Then, the HAUP prototype saves the user's HW and behaviour in a particular table at every subsequent successful log-in. The full codes is available in appendix (B).

Figure 6.7 shows the user's HW database which includes the user's HW and biometric behaviour [136].

Key0	Key1	Key3	Key4	Key16	Key17	Key18	Key19	Key20	Key21	Key22	Key23	MacAddress	StorageMedia	BIOSNo	UserName
62	47	78	0	78	63	31	63	93	62	63	78	00-1C-C0-6D-6E-AA	-128640614	BQJO8280076R	andy
47	78	78	0	63	62	47	47	109	47	63	78	00-1C-C0-6D-6E-AA	-128640614	BQJO8280076R	andy
94	78	79	0	78	93	46	47	94	63	62	63	00-1C-C0-6D-6E-AA	-128640614	BQJO8280076R	andy
94	94	79	0	93	78	31	63	63	47	79	63	00-1C-C0-6D-6E-AA	-128640614	BQJO8280076R	andy
93	78	62	0	78	94	31	47	78	62	63	46	00-1C-C0-6D-6E-AA	-128640614	BQJO8280076R	andy



User's behaviour in typing the username and password keys every successful log-in attempts.



User's hardware is used in successful log-in attempt.

FIGURE 6.7: HAUP Profiling Database

In HAUP, the database of HW information is considered to the property of the user as this is a record of the user behaviour and as such is subject to the user's privacy rights. Exposure or leakage of this information would possibly harm the HAUP integrity. In order to ensure that a user's privacy is maintained and protected, all private information relating to the user is transferred from the user's computer to the server in

an encrypted form using hashing encryption technique. Figure 6.8 demonstrates one example of how encrypted information is transported for HW by using the manufacturer serial part numbers. in addition, HAUP prototype is created to do the set of exarments which will be improved in the real system that have to improve the encryption method in order to provide saver environment for HW and users behaviour information. Further information about the database architecture is in Appendix (C).

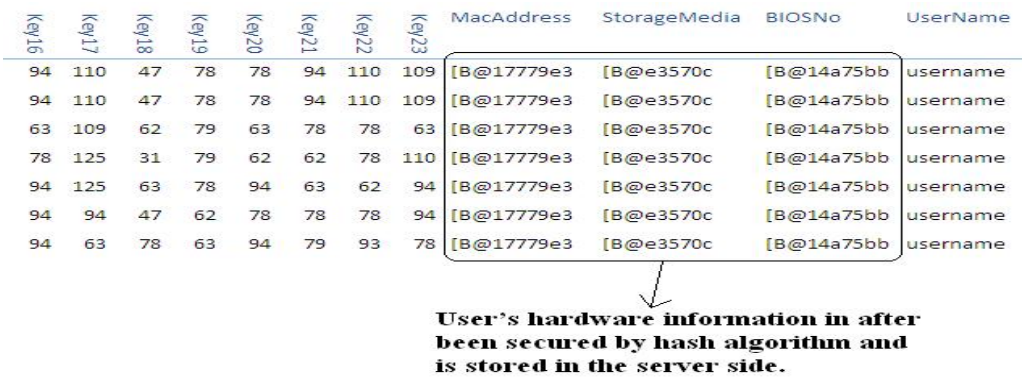


FIGURE 6.8: HAUP Encrypted Database

To measure user’s behaviour and patterns the HAUP prototype draws a chart to record the user’s keystroke speed for a particular HW at every successful log-in. In addition, this chart shows the average of the user’s keystroke pattern in comparison with previous successful log-in attempts and the user’s keystroke behaviour of the current log-in. Figure 6.9 shows Java

code which displays user behaviours and pattern. Further information about full HAUP prototype code for analysing user's hardware and behaviour is in Appendix (B).

```
285 g2.setPaint(Color.green);
286 for(int j = 0; j < data.length; j++)
287 { if(data[j]!=0) { int x = x0 + (int)(xScale * (j+1));
288   int y = y0 - (int)(yScale * data[j]);
289   g2.fillOval(x-2, y-2, 4, 4);
290   XPasswordPoint=x;
291   YPasswordPoint=y;}
}
```

FIGURE 6.9: Plot Measure Code to Show Users Keystroke Behaviour and Pattern in a HW

6.5 Summary

This chapter described key parts of the HAUP software prototype. Section 6.1 provided HAUP technical scenario in access control to authenticate the user using HW information. Section 6.2 described the main system tasks and requirements of HAUP including highlighted HAUP procedures including sequence diagram of HAUP modelling that answer the research question Q.6. by addressing the profiling factors in HAUP approach to authenticate the user. Section 6.3 revealed that this HAUP prototype is executable in a Windows operating system by using the Java code to show that HAUP analysed and evaluated the HAUP approach. Following this, the chapter illustrated the storing methods to save user's HW and behaviour. Section

6.4 presented a method of reading user's hardware and monitoring user behaviour at every successful log-in attempt and discussed and analysed safe and effective ways of transferring user data ensuring user's privacy was maintained. Finally, this chapter provides observing function to determine the similarity between user behaviour and pattern every successful log-in attempt.

The next chapter will provide set of experiments which give detailed examples of when the HAUP system has been used followed by an evaluation of these set of experiments.

Chapter 7

Set of Experiments and Evaluation

Objectives

1. Define evaluation criteria.
 2. Determine priority classes.
 3. Implementing set of experiments.
 4. Presenting evaluation.
-

This chapter provides set of experiments using the HAUP prototype. In these set of experiments, group of users who have (eight postgraduate students) used IT services before and have good experience in providing their pattern in typing keyboard keys .The HAUP prototype illustrates user's HW and behaviour in diagram analysis. Then, neural networks are used to observe the user's behaviour data to analyse user patterns based on the particular HW in order to evaluate the HAUP profiling technique. Section 7.1 determines and describes the evaluation criteria to assess HAUP. Section 7.2 shows the user's behaviour diagram when using a particular HW. This section provides the HAUP profiling technique to clarify the difference between HAUP profiling and the current authentication profiling.

Section 7.3 provides the data collected by the HAUP set of experiments. Section 7.4 discusses the log-in timing behaviour to support the HAUP analysis of the user's HW activity which affects the user's behaviour when a particular HW is used at particular time. Section 7.5 addresses the ability of the HW environment to support the profiling of user behaviour when the same log-in keys are used.

To analyse user's behaviour using additional recognizing techniques, section 7.6 determines the neural network analysis which explores the differentiation between neural networks analysis and the HAUP prototype to recognize user behaviour. Then, subsection 7.6.1 clarifies the neural network comparison

between the user's behaviour in typing the same authentication password keys when the same and different HW is used. After that, subsection 7.6.2 illustrates the neural network analysis and comparison of users' behaviours when group of users use the same HW and password keys, followed by analysing these user behaviours when another HW are used.

7.1 Proposed Set of Experiments and Evaluation Criteria

These set of experiments proposes to apply the HAUP prototype as the authentication method with the traditional username and password. Then, following experiments will be applied to evaluate HAUP approach:

1. In the first experiment, one user who has experience in IT services, e.g., user who familiar with bank account services. This user will use two different HW and same authentication keys "username-password" in performing access procedure for two hundred times. This experiment profiling user behaviour in typing log-in keys using the same HW and shows the HAUP prototype demonstrates how to recognize user behaviour and pattern every successful log-in attempt. This experiment aims to recognize the different on observing user pattern in the two HW.

2. In the second experiment, two users use same HW and same authentication keys "username-password" for two hundred times by every user. This experiment aims to explore the different in these users pattern when same HW recognize every user pattern. This experiment shows the HW performance using HAUP prototype to observe each user pattern in keystroke behaviour.

3. In the third experiment, group of users will use two different HW and same authentication keys "username-password" for hundred times. This experiment provides HAUP prototype and neural network techniques to show the different in recognizing user pattern using the two different HW.

So, in a specific identity authentication application when we are looking for the potential biometric to be used, the following three criteria must be evaluated [137] to clarify the advantages and disadvantages of the developed approach:

E.C.1. Acceptability, which indicates that people have accepted the process of using the HAUP system.

E.C.2. Circumvention, which identifies how it is possible to circumvent the authentication system.

E.C.3. Performance, which specifies the achievable identification (verification) accuracy and resources needed to achieve an acceptable level of accuracy [138, 139].

7.2 Profiling User Behaviour in Typing Log-in Keys Using The Same HW

Using particular password keys in the log-in procedure by and using specific HW can improve recognizing user behaviour in specific user context environment (See Section 3). In the HAUP prototype the user's behaviour is assessed by monitoring the user's keystroke speed which is assumed to be between zero and one thousand milliseconds by determining the time response between each keystroke and each key release which captures every username and password key in every successful log-in attempt. This assumption determines the user pattern domain which develops the ability to recognize user behaviour and declares the relationship between user behaviour and real user patterns. The HAUP prototype learns user's behaviour by recording every keystroke used at every successfully log-in attempt.

In this learning, the HAUP monitors user's keystrokes. Furthermore, capturing user behaviour with respect to the user's HW can build profiling signatures or characteristics for the user. Figure 7.1 shows input user behaviour in typing password keys using the same HW and determines particular domains for the user's keystroke speed. So, the user's keystrokes in the username and password keys has particular characteristic about the user which are based on HW analysis.

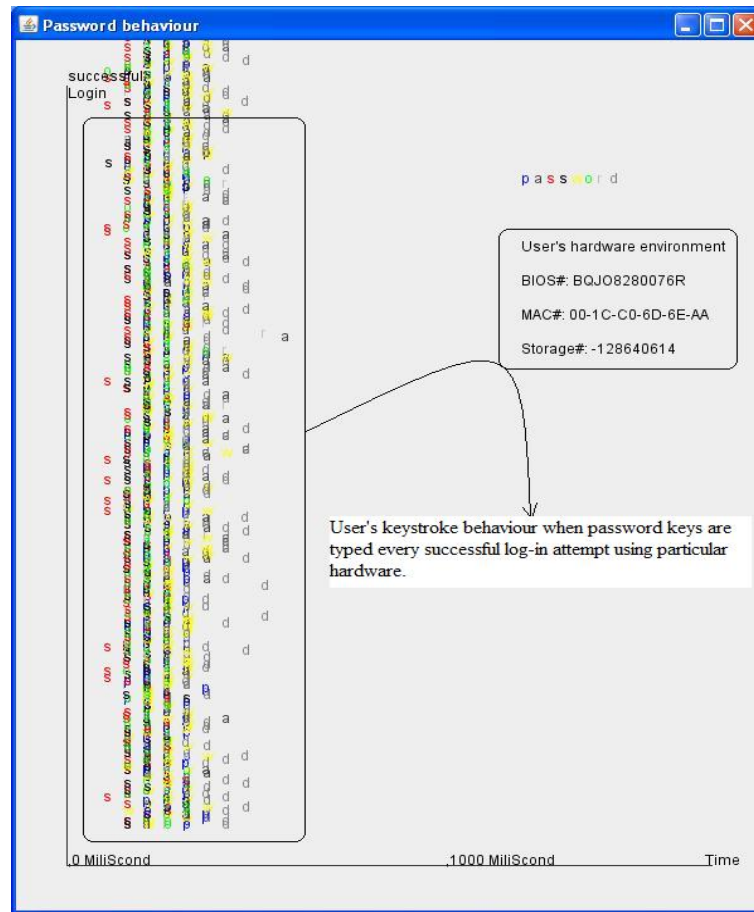


FIGURE 7.1: User Behaviour in Typing Log-in “Password” Keys by Using The Same Hardware

User behaviour is recognized by calculating the password keystrokes’ response time which is the delay time between each key press and each key release. This password-keystrokes analysis should be limited by the user pattern after the user becomes familiar with using particular HW. Figure 7.2 shows the user keystroke speed boundaries after capturing user behaviour in 200 successful log-ins.

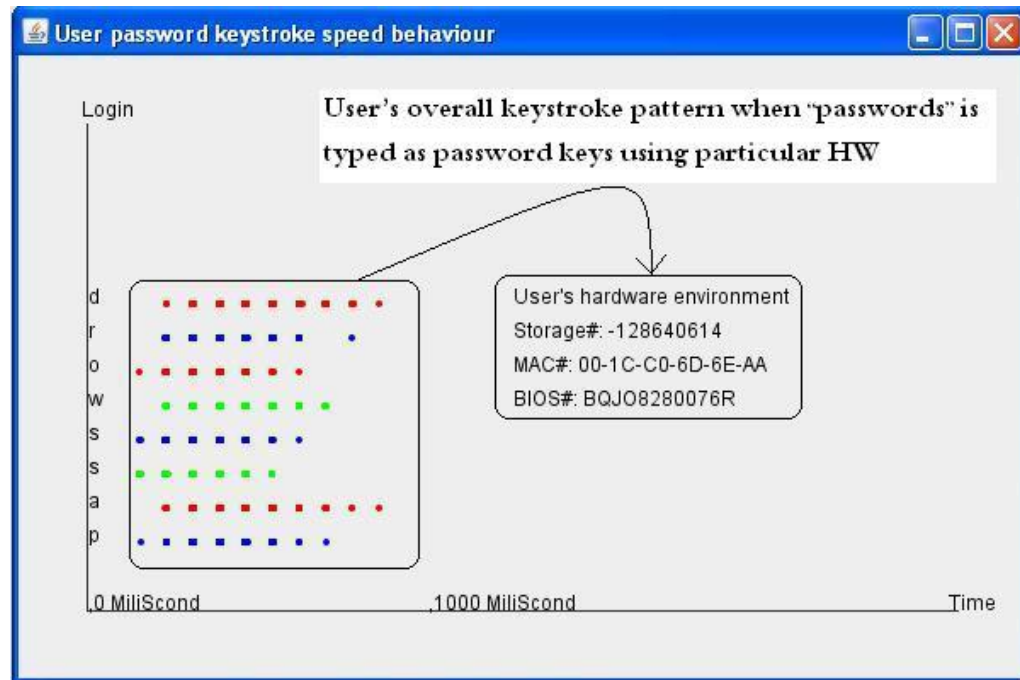


FIGURE 7.2: User Pattern in Typing “Password” Using The Same Hardware

7.3 Set of Experiments Using HAUP Approach

The HAUP prototype was designed to record each user’s HW and behaviour as an implicit identity in the log-in procedure. When a user inputs the correct username and password the HAUP prototype collects three HMSPNs. These are date BIOS, MAC address and the hard disk manufacturer’s serial numbers. When the user logs in to the system and the HW information is determined, the HAUP prototype searches in the user’s HW

database to determine if the user has used the current HW previously or if this is the first time. If the user has used it before, HAUP shows the percentage of usage of the current HW from other user usage the HAUP prototype analyses user behaviour and provides a level of trust to authenticate the user if the user has used current HW before.

7.3.1 One User Uses The Same Hardware

In this experiment, the user has had more than four hundred successful log-in attempts to the HAUP prototype using a particular computer device's HW. The HAUP prototype clarifies the similarity between the current users behaviour and the previous user's pattern to explore the trust result and establish if the user has used the current username and password previously. Figure 7.3 shows the percentage of HW usage and the ability to recognize user behaviour in the current successful log-in attempt. User behaviour is reflected through red dots that appear in Figure 7.3 and observes if the current user behaviour in typing the username and password is related to the user pattern which is reflected through the domain between yellow dots, as shown in Figure 7.3. This observation declares if the user behaviour is within the user's pattern domain or not.

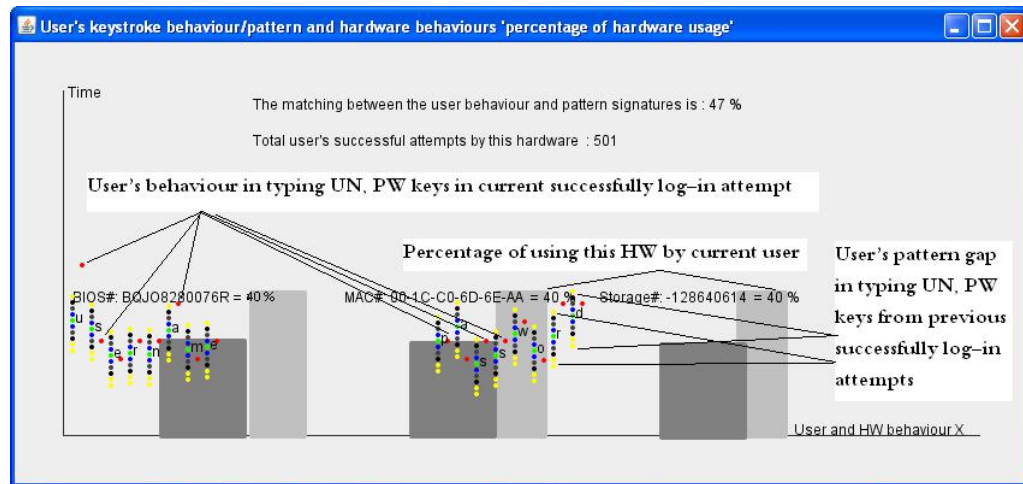


FIGURE 7.3: Hardware Usage and Profile Against Keystroke Pattern

In contrast, if the user's keystroke behaviour has changed this could possibly indicate a hacker or fraudulent usage which may have compromised the log-in keys. The HAUP prototype recognizes the difference between user patterns and behaviour in typing the username and password keys. Figure 7.4 clarifies the delay in user's keystroke response times when the username and password is typed. So, the HAUP prototype observes the delay in the user's current behaviour which is reflected through red dots that appear in Figure 7.4. Moreover, the total user behaviour is changing in the user's keystroke signature and the HAUP prototype shows the user's behaviour in keystroke speed is not similar to the user's pattern.

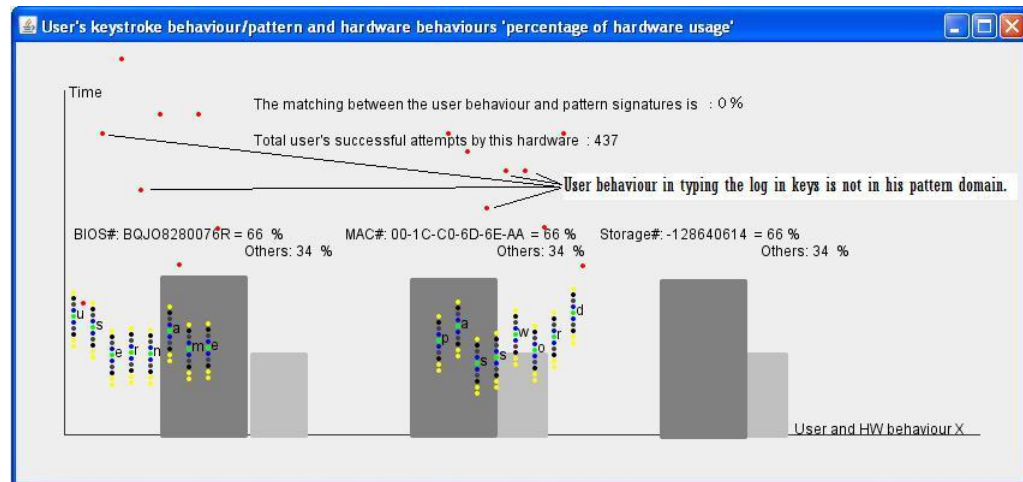


FIGURE 7.4: Hardware Usage And Profile of The Delay in Keystroke Behaviour

7.3.2 One User Using Two Different Sets Of Hardware

In this experiment the HAUP prototype improves the ability of observing the difference in user behaviour when different HW are used. In this experiment, the user has performed 50 successful log-ins using the same username and password keystrokes as in the previous log-in procedure. Also in this experiment, the HAUP prototype observes user behaviour and patterns in the second computer device is faster than the first computer observation. Figure 7.5 shows keystroke behaviour and patterns in the first HW set on the left side. In contrast; the right hand side of Figure 7.5 indicates the second HW set which has been

affected by the keystrokes and rhythm of the second HW set when log-in keys are typed.

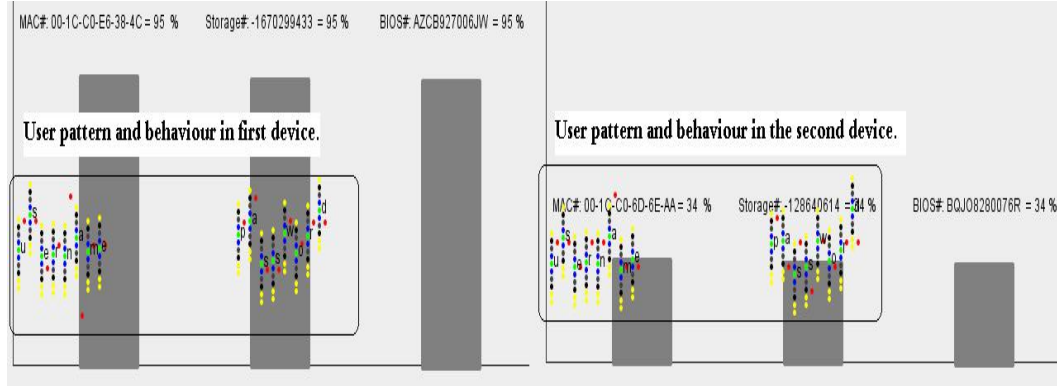


FIGURE 7.5: Hardware Profiling and Recognizing User Behaviour Against One User Using Two Different Hardware

7.3.3 Two Users Using the Same Password and Hardware

In this experiment, two users used the same HW and same password for fifty successful log-in attempts. The HAUP prototype recognizes the HW effect by observing user's keystroke behaviour. The HAUP prototype compares users based on the their patterns in using a particular HW. Figure 7.6 show the HW effect in user behaviour and pattern stamps.

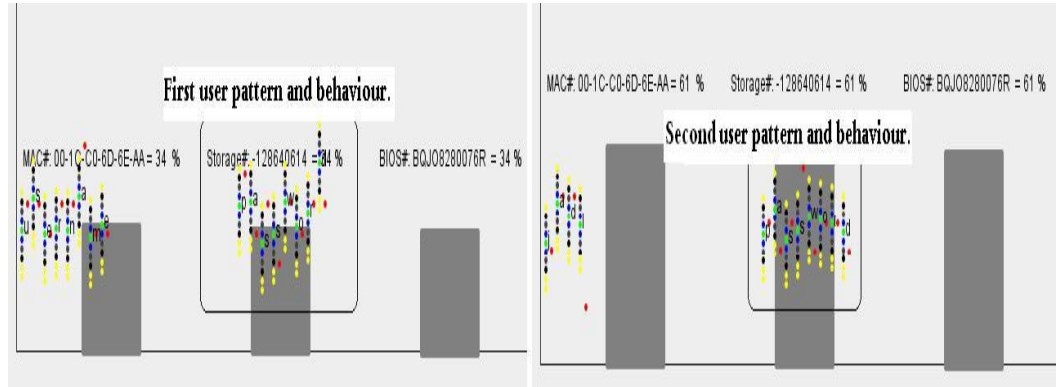


FIGURE 7.6: HW Usage and Profile Against Two Users Use Same Log-in Information “Password” in One Hardware

7.4 The Relation Between Log-in Time and Hardware Usage

The log-in time during the day and week could be an additional factor to support HAUP. This time stamp can clarify user behaviour and how this changes during the day time. For example, the user may move between their desktop at their workplace then use their laptop at home and their smart phone in between. In this case the user’s pattern is affected by the user HW context during the day times which appears in the performance of recognizing user behaviour. In addition, the time factor recognizes the user HW changing by recording and monitoring the user HW changing in the previous log-in attempts. For example, if the user used to use a particular HW from 9:00 am to 04:00 pm every day the system could expect

to see a specific pattern in the user's HW by the analysis of the previous record of the user's behaviour at these times, even if the HAUP couldn't read the user HW. The HAUP system can read the HAUP prototype and adds the time factor to support the profiling of changes in the user's HW. Figure 7.7 show the number of users using a particular HW at a specific time.

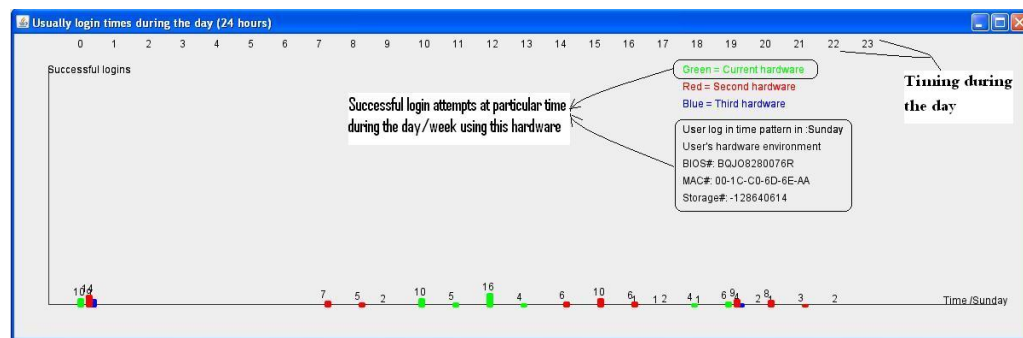


FIGURE 7.7: Time "Signature" in Using a Particular Hardware

7.5 Priority Classes Threshold

In the HAUP prototype, the user behaviour calculations are based on recognizing the user's HW. For example, if the user has more than one HW device, i.e. desk-top, mobile and lap-top, the HAUP system should be aware of the different environments and the different times these have been used throughout the day.

In the HAUP authentication, the priority classes are given a level of trust and the HAUP's focus is on the valid username

and password. If valid, the HAUP classifies the user HW to give the level of trust to the user. If the HW has been used by the user before; the HAUP further increases the user's level of trust. If the user behaviour related to the user has stored the patterns from the previous usage, the level of trust increases and the user can get more priority. Moreover, the level of trust is increased if the user used the HW by his pattern at the same time during the day/week. However, if one of these factors was not reliable in relation to previous successful attempts, the level of trust will decrease because of the system inability to bind between the user's pattern and current user behaviour. Thus, the ability of implementing MFA has the benefit of calculating the level of trust for the user at every log-in attempt. Priority classes have contributed to the implementation of the HAUP approach. Priority classes provide percentage results to trust the user based on the user's data in the HW log-in database. This result clarifies the level of trust by determining the percentage of trust and partial trust to provide full success for user authentication. This percentage begins with the denial of the service which is zero percentage if the username or password are not valid. Then the HAUP system gives partial success if the HAUP approach is accepted with the username and password validation. However, for the first usage of the new HW the authentication system may require an additional verification question. After that, when the user accepts the HAUP

approach in the access control, the system includes the user behaviour and time stamp factors to implement the HAUP authentication approach as the MFA. So, if the user keeps using the same HW at a particular time and applying the same behaviour and patterns so, the HAUP level of trust will increase. Figure 7.8 shows priority the classes for every log-in procedure using HAUP factors.

Login sequential	Knowledge factor		Ownership factor			Inherence factor		Result	Level of Trust	
	Traditional authentication		Hardware profile			Behavior pattern				
	User-name	Password	BIOS	MAC	Storage	Keystroke speed and rhythm	Log in time			
1	x		x	x	x	x	x	Denied	0	Built in level of trust
2	✓		New	New	New	First time	Particular time	Partial Success	10%	
3	✓		✓	✓	✓	Second time	At same time	Partial Success	20%	
4	✓		✓	✓	✓	✓	At same time	Partial Success	30%	
5	✓		✓	✓	✓	✓	At same time	Success User logs In	40%	
6	✓		✓	✓	✓	✓	✓	Success User logs In	50%	
7	✓		✓	✓	✓	✓	✓	✓	60%	
8	✓		✓	✓	✓	✓	New Particular time	Partial Success	30%	Built in new Regular time
9	✓		✓	✓	✓	✓	At same new time	Partial Success	40%	
10	✓		✓	✓	✓	✓	At same new time	Success! User logs In	50%	
11	✓		✓	✓	✓	✓	✓	✓	60%	
12	✓		✓	✓	x	x	x	Partial Success	30%	Hardware environment changed.
13	✓		✓	x	x	x	x	Partial Success	20%	
14	✓		New	New	New	First time	Particular time	Partial Success	10%	
15	✓		✓	✓	✓	Second time	At same time	✓	20%	
16	✓		✓	✓	✓	Third time	At same time	✓	30%	
17	✓		✓	✓	✓	✓	At same time	✓	40%	
18	✓		✓	✓	✓	✓	✓	✓	50%	
19	✓		✓	✓	✓	✓	✓	✓	60%	

FIGURE 7.8: Priority Classes in HAUP Factors

7.5.1 Overall Level of Trust

Giving percentage measurements for HAUP authentication factors can represent a particular level of trust for the user. These factors are collaborated with the username and password authentication approach. In the HAUP prototype a trust level of 75% is given if a user keeps using the same three HW parts (BIOS, MAC, HDD) every successful log-in. This weight of percentage (75%) is summarise of the three parts weight which are (BIOS=25% , MAC=25% , HDD=25%) that focuses on the user's HW as the main factor to profile the user in HAUP. So, HAUP give 75% for HW information because this value is representing main factor in HAUP approach and this factor has influenced in user behaviour and log-in time. If the user used same HW, a trust level of 15% is given for the user behaviour because of the relation between user HW and behaviour that be supported by user's HW which been used before. So, user the behaviour level is increasing to 90% based on the matching between the user's patterns and behaviour. After that an additional 10% is given if the user has logged-in at the same time because the time has influenced to recognize user behaviour and not related user behaviour but can support to recognize user context during the day. However, these percentages will be recalculated if some factors has significant and more influence in comparing with another HAUP factors which can improve the approach assistant in future work. This time percentage is

increased when the user continues to log-in at the same particular time every day or week and may eventually reach up to a 99% of level of trust. As result of analysing these authentication factors, the HAUP prototype supports the username and password authentication approach by determining the similarity percentage between the previous and current user's HAUP factor. Figure 7.9 illustrates the overall level of trust based on HAUP prototype factors.

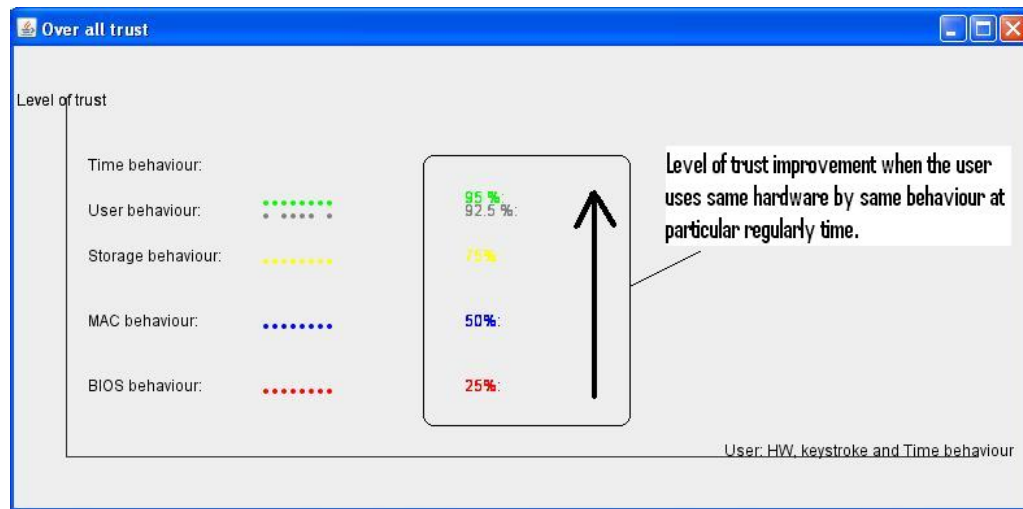


FIGURE 7.9: Overall Level of Trust

As mentioned in the mathematical model (Chapter 5 section 5.1.1) $W_\alpha + W_\beta = 1$, user HW weight to be trusted can be $T_\alpha = .5$ if the user used the HW in previous successful log-in attempts. User behaviour weight can be $W_\beta = .5$ if the user behaviour is similar to user pattern in previous successful log-in attempts which illustrated by user behaviour weight

is close to .5. However, if user behaviour weight was close to zero that means, $W_\alpha = .5$ and $W_\beta + W_\alpha = .0$. In contrast, if the user did not use current HW before $W_\alpha = 0$ which means $W_\alpha + W_\beta = 0$ because of HAUP inability to determine user pattern in new HW. These weight values may have additional weight factor, e.g., time and can be change based on the factor influence in the level of trust. For example HW weight can be $W_\alpha = .75$ and the user behaviour is $W_\alpha = .25$ when the HW can has important characteristics to trust the user.

7.6 Neural Network Analysis in Matlab

As been mentioned in Chapter 2, the neural network is used in profiling user's behaviour and has specific parameters to evaluate recognition approaches by adaptive learning. The adaptive learning rate is used to recognize user behaviour. One of adequate techniques to identify user behaviour systems is neural networks in Mat-lab application. Neural network has just two random parameters which are learning rate and number of hidden layers. These numbers of parameters are less than fuzzy and genetic parameters. A neural network criterion declares the difference between the user's behaviours based on a particular HW. Some of these criteria are 1) mean square error (MSE) which the different between the desired signal "HW" and the

neural network outputs, 2) Epoch which is time required to measure all neural network calculation.

When training by minimum error, this represents the maximum number of iterations [140]; performance which is mimicking human error patterns and measured the behaviour of the system as minimum or maximum error that calculated by mean square error. [119]; validation which is a technique for assessing how the results of a statistical analysis will generalise to an independent data set [141], and gradient which is a first-order optimization algorithm to finding a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point [142]. These criteria clarifies HW influences in profiling user behaviour. So, neural network analysis is used for analysing the HAUP authentication results. This result determines the user keystroke speed when typing password keys in two different computers. This analysis investigates the differences in user keystroke patterns in every computer environment. This investigation declares the keystroke speed at every successfully log-in attempt on every computer.

In HAUP studies, neural network analysis has five hidden layers to learn user patterns using fitting techniques [143]. This fitting learning uses a back propagation technique which is adaptive rate learning, has minimum learning error and has

two random parameters. These parameters are a number of hidden layers and learning rates. Furthermore, the HAUP prototype is trying to obtain optimal performance to learn the user's behaviour which can determine the user's pattern in a particular HW environment. So, this experiment focuses on learning the user's keystroke behaviour in a variety of HW to prove the differentiation in profiling user behaviour when the HW changed.

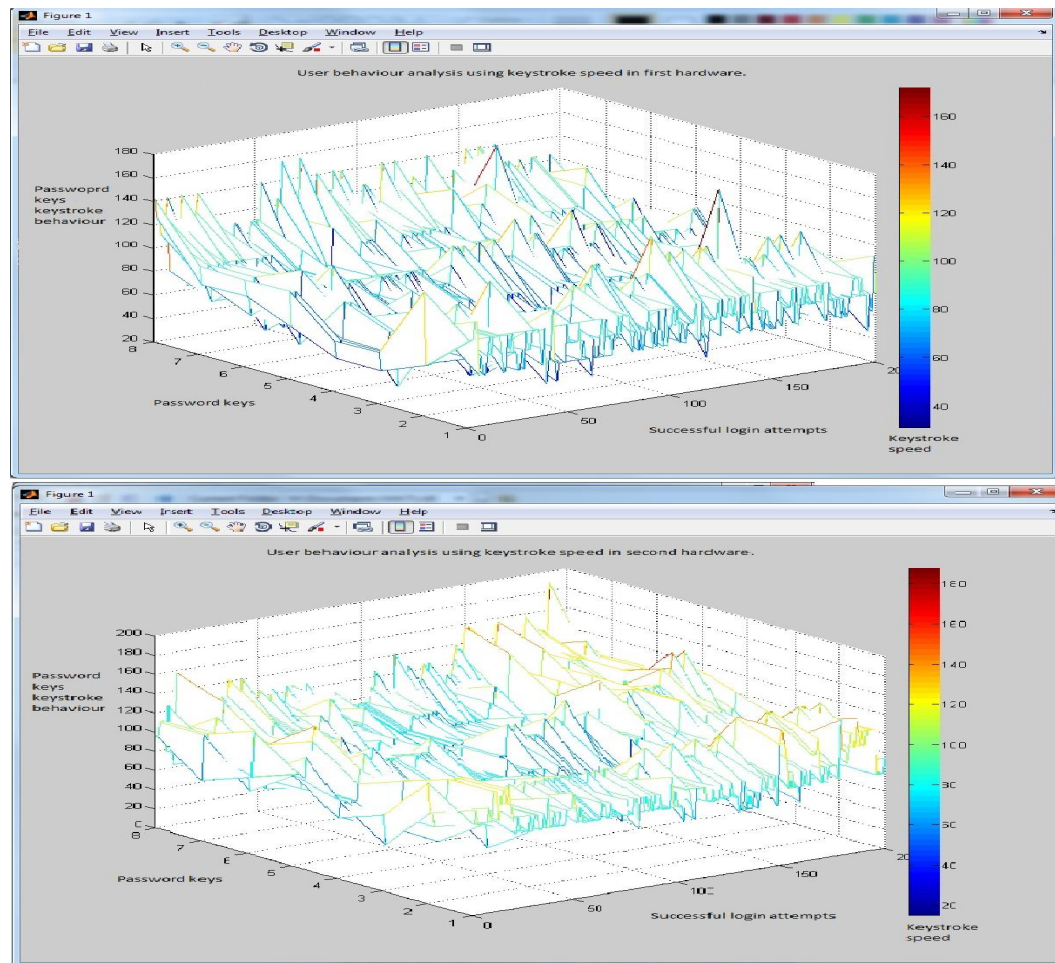


FIGURE 7.10: Users Keystroke Speed Analysis Using Two Different Hardware

Figure 7.10 illustrates the user pattern by determining the user's keystroke speeds in the user password at every successful log-in. In this experiment, more than two hundred successful attempts are monitored using the same password keys using two different HW. So, the keystroke speed in the first computer is

spending less time than the keystroke speed in the second computer. In addition, every keystroke key has a specific average speed in every computer HW.

7.6.1 Using Neural Network to Compare Between Users Behaviour When Two different HW are Used

In this experiment neural network recognition is used to learn the user's behaviour when two different HW and the same password keys are used for more than two hundred successful log-in attempts. In these successful log-in attempts, the user entered the password as the required key log to be granted authorisation.

Figure 7.11 shows neural network training to learn two user's patterns. On the above side the first user required 6 epochs to reach the maximum number of errors to learn the user pattern and the best validation performance was 3.7856e-011. However, the second user on the bottom of figure 7.11 has a validation performance of.00023138 which is the best validation performance and needed 11 epochs to be recognized.

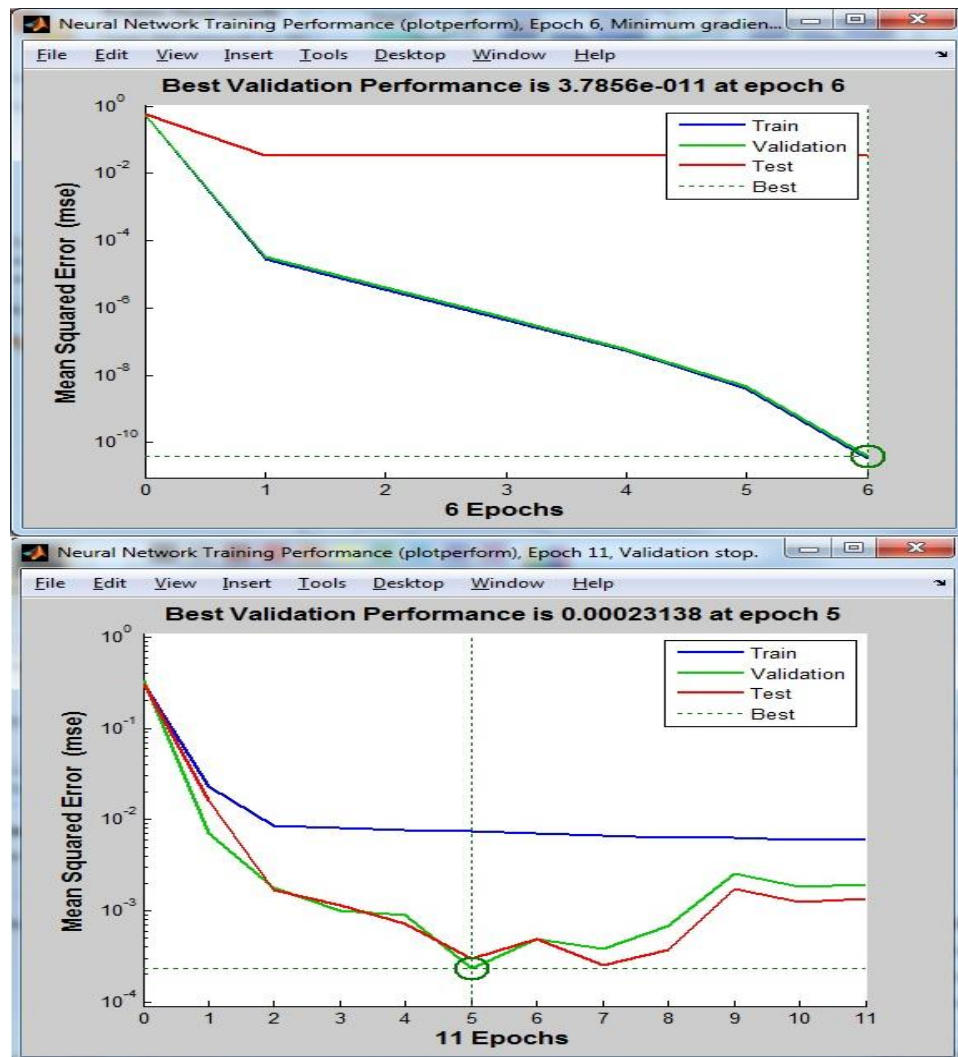


FIGURE 7.11: Neural Network Training Performance From Two Users Patterns Using the Same Hardware

In order to form a comparison between the HAUP and neural network Analysis the keystroke behaviour of each user has been collected and the users behaviour has been compared based on both users using the same HW and password keys.

Both users have at least fifty successful attempts during the log-in procedure.

The result of analysing the users keystroke patterns and behaviour in the successful log-in is obtained by monitoring the users keystroke speed signature as a sample of recognizing the users behaviour, based on a particular HW. In addition, the neural network provides a specific fitting for each user which is indicated through second column in Figure 7.12.

Figure 7.12 presents the users behaviour and pattern analysis when using the same HW. This experiment shows the HW influence in recognizing user behaviour, even when the same username and password were used.

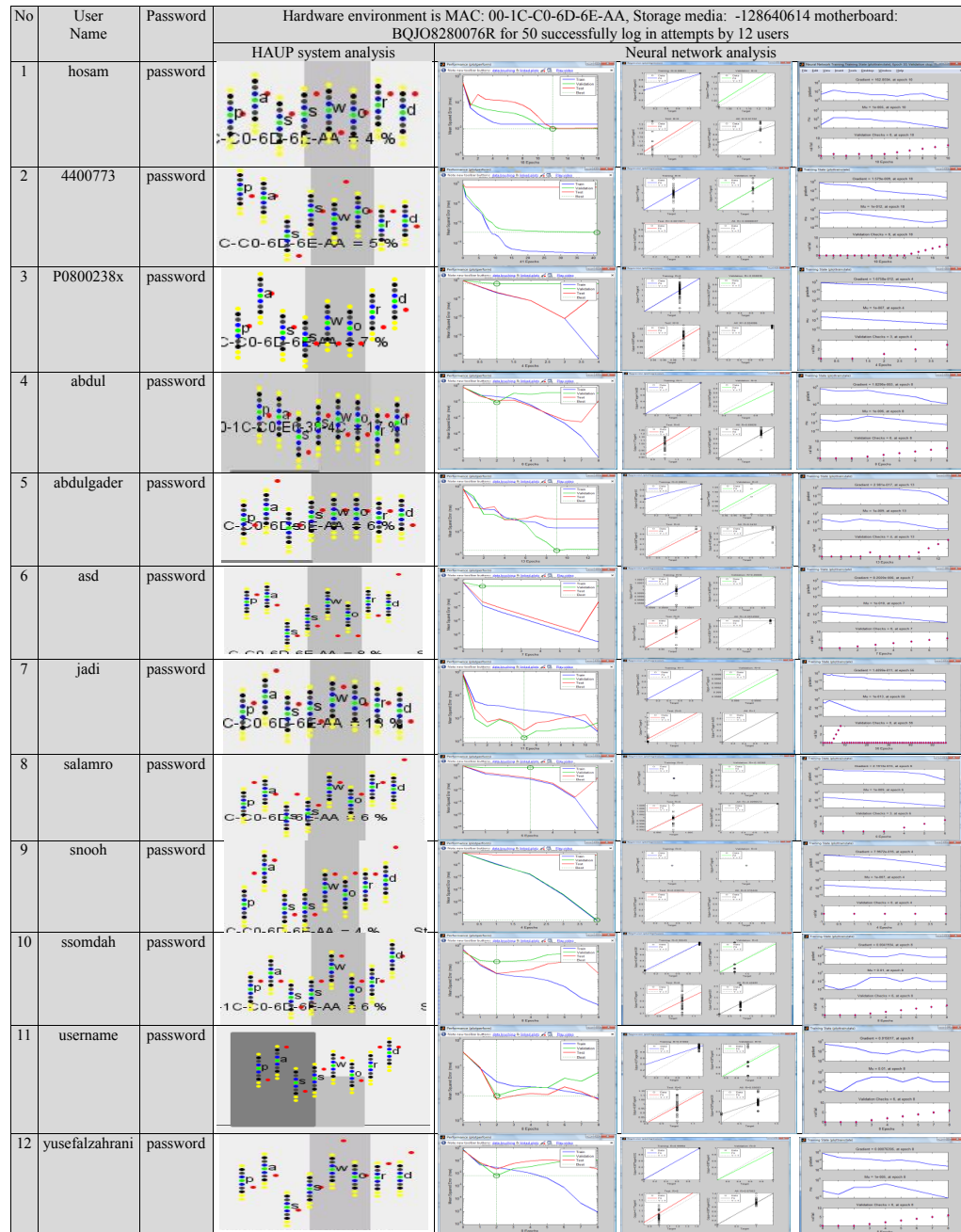


FIGURE 7.12: Analysis of User Behaviour When Using the HAUP Prototype and Neural Network With Respect to a Particular Hardware

The comparison in Figure 7.13 illustrates the HAUP process of recognizing and demonstrating the differentiation in observing user keystrokes when using different HW and how this may affect the monitoring of user behaviour. In this experiment, every users behaviour was monitored during 50 successful log-ins when using the same username and password. Then HAUP system extracts the user patterns based on the HW factor. User patterns are reflected in the figures displayed in the fourth and fifth columns of Figure 7.13. These figures clarify the user patterns in the two HW contexts, using the HAUP prototype.

No	User Name	Password	Hardware environments	
			00-1C-C0-6D-6E-AA, -128640614 and BQJO8280076R	00-21-5D-13-F5-9A, -998986574 and CNF8375GR0
1	hosam	password		
2	4400773	password		
3	P0800238x	password		
4	abdul	password		
5	abdulgader	Password		
6	asd	password		
7	jadi	password		
8	salamro	password		
9	snooh	password		
10	ssomdah	password		
11	username	password		
12	yusefzahrani	password		

FIGURE 7.13: Comparison Between Users Keystroke Analysis When Using Two Different HW Devices and The Same Password Authentication Keys

7.6.2 Neural Network Analysis Experiment For Group of Users

To observe the difference between users patterns using the same HW the neural network was used. In order to determine the user patterns and then compare them against the neural network recognition, group of user's keystroke samples were used which were the number of users who used the traditional username and password authentication approach. The username was chosen by the user and the password word was set at password. Then, following 50 successful log-in attempts on a particular HW device, the user pattern is then been learned. The neural network analyses data from eight key samples which are the keystroke behaviour of typing the password keys using the same computer device. Then, 15% of the validation and testing mechanisms are considered which build the training of a set of independent measurements which is the overall percentage of recognition of the user pattern based the on neural network back probation. 15% of this recognition is determined because of the similarity in user behaviour when using two different HW devices that require a high level of recognition to recognize the users behaviour. So, neural network training is based on a minimum error percentage of 30% to learn the users pattern.

The optimal recognition assumed in the case was by 50 inputs and 50 outputs. In addition, applied neural network

layers in two experiments aim to show how the HW influences the process of recognizing user pattern in more than one neural network learning. However, the aim of using neural network analysis in HAUP set of experiments is not to explore the optimal recognition process in order to learn the user pattern. The aim is to highlight the differences when observing user behaviour when using HW factors and how this changes when observing how the HW influences the neural network learning. The successful log-in attempts observes the differences in user behaviour when different HW devices are used which is be shown in Figure 7.14 below. However, the neural network calculated the response time consume by millisecond to learn user. Which have zero result in some cases that means close to zero.

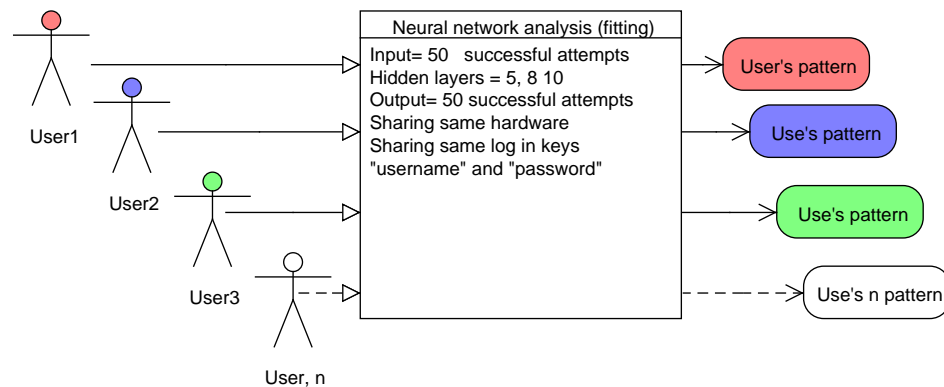


FIGURE 7.14: Neural Network Analysis Including Input, Output and Hidden Layers

TABLE 7.1: Users Pattern Analysis by Neural Network When Same password Keys and twoHW are Used 3 layers

#	User name	HW	Epoch	Time Millisecond	Min Error	Performance	Gradient	Mu	Validation
1	$b_{(4400773)}$	1	11	00:00:01	.769	.283	8.07e-05	1.00e-07	6
	$b_{(4400773)}$	2	8	00:00:00	.428	5.03e-05	1.14e-06	1.00e-11	6
2	$b_{(abdul)}$	1	8	00:00:00	.39	1.61e-24	1.63e-13	1.00e-11	0
	$b_{(abdul)}$	2	14	00:00:00	.801	.0253	.000468	.000100	6
3	$b_{(abdulgader)}$	1	8	00:00:00	.475	.000174	1.45e-06	1.00e-11	6
	$b_{(abdulgader)}$	2	1000	00:00:14	.622	.0147	3.12e-10	1.00e-09	0
4	$b_{(asd)}$	1	8	00:00:00	.473	.280e-12	7.62e-12	1.00e+11	1
	$b_{(asd)}$	2	5	00:00:00	.342	8.83e-15	2.93e-15	1.00e+08	6
5	$b_{(hosam)}$	1	59	00:00:01	.694	.0147	1.10e-10	1.00e-10	0
	$b_{(hosam)}$	2	45	00:00:00	.763	4.44e-17	9.94e-11	1.00e-13	0
6	$b_{(jadi)}$	1	6	00:00:00	.253	7.01e-28	2.65e-16	1.00e-09	0
	$b_{(jadi)}$	2	11	00:00:00	.383	.0269	.00353	.000100	6
7	$b_{(P0800238x)}$	1	19	00:00:00	.617	1.81e-18	8.90e-11	1.00e-14	0
	$b_{(P0800238x)}$	2	18	00:00:00	.248	3.72e-19	1.38e-14	1.00e-14	0
8	$b_{(ssomdah)}$	1	134	00:00:02	.354	7.49e-18	9.91e-11	1.00e-13	0
	$b_{(ssomdah)}$	2	10	00:00:00	.431	.0184	8.03e-05	1.00e-07	6

As a result of using neural network fitting, Tables 7.1 and shows the difference in neural network factors when neural networks learn the user patterns using three layers.

Tables 7.2 and shows the difference in neural network factors when neural networks learn user pattern using five layers.

TABLE 7.2: User Pattern Analysis by Neural Network When the Same PW Keys and two HW are Used 5 layers

#	User name	HW	Epoch	Time Millisecond	Min Error	Performance	Gradient	Mu	Validation
1	$b_{(4400773)}$	1	18	00:00:00	.4751	3.1803e-01	8.041e-06	1.00e-14	0
	$b_{(4400773)}$	2	8	00:00:01	.9798	.12492	6.659e-06	1.00e-11	0
2	$b_{(abdul)}$	1	8	00:00:00	.669	.0450	1.83e-05	1.00e06	6
	$b_{(abdul)}$	2	14	00:00:00	.654	.0291	.000302	.000100	6
3	$b_{(abdulgader)}$	1	13	00:00:00	.756	.0148	2.98e-17	1.00e-09	4
	$b_{(abdulgader)}$	2	21	00:00:00	.346	.0147	3.91e-11	1.00e-11	4
4	$b_{(asd)}$	1	143	00:00:02	.362	.0147	1.82e-10	1.00e+10	0
	$b_{(asd)}$	2	16	00:00:00	.427	4.09e-05	7.62e-06	1.00e+11	6
5	$b_{(hosam)}$	1	4	00:00:00	.928	3.52e-27	3.93e-15	1.00e-15	0
	$b_{(hosam)}$	2	5	00:00:00	.831	1.71e-30	5.22e-17	1.00e-08	0
6	$b_{(jadi)}$	1	6	00:00:00	.467	4.24e-05	2.64e-16	1.00e-09	4
	$b_{(jadi)}$	2	9	00:00:00	.468	.00888	.00314	1.00e-09	6
7	$b_{(P0800238x)}$	1	9	00:00:01	1.11	4.79e-31	2.04e-17	1.00e-12	0
	$b_{(P0800238x)}$	2	16	00:00:00	.367	.0147	5.94e-11	1.00e-14	3
8	$b_{(salamro)}$	1	12	00:00:00	.526	.0227	2.12e-07	1.00e-06	6
	$b_{(salamro)}$	2	9	00:00:00	.562	6.06e-32	2.00e-17	1.00e-12	6

From the neural network analysis in Tables 7.1 and 7.2 we note the user ($b_{ssomdah,password}$) has a different attitude or pattern (mean squared error, epoch and gradient) that emerges from the neural network when the HW is changed. The gradient result is changing in both tables for every user when the HW environment changes. For example, the first user $b_{(4400773)}$ in the Table 7.2 has 41 epoch in the first HW analysis, in contrast the epoch value was 2 in the second HW. In addition, user minimum error, performance, gradient and momentum learning rate have different analysis results which clarifies the HW influence in recognizing users pattern.

Furthermore, the neural network analysis and recognition have different results to those of discrimination that appear in the previous tables. This result corroborates the theory that the users computer HW environment provide boundaries that has an efficient capacity to measure any discrimination when analysing user behaviour.

Table 7.3 shows the users behaviour by using neural network criteria without profiling a particular computer HW which is present in another neural network analysis to learn the users pattern. This result insists the users context value in profiling user behaviour. Moreover, using keystroke patterns to analyse user behaviour can be supported by further behaviour analysis. For example, we can include the users keystroke rhythm

and mouse signature which provides more opportunity to analyse the users behaviour accurately with respect to a particular HW.

TABLE 7.3: Analysis of User Patterns Using Neural Networks Without Determining the Users Hardware and Using Three Layers

#	User name	Epoch	Time ms	Min Err	Performance	Gradient	Mu	Val
1	$b_{(4400773)}$	10	00:00:00	.843	.0118	.00200	.100	6
2	$b_{(abdul)}$	113	00:00:02	.500	.00714	9.93e-11	1.00e-10	0
3	$b_{(abdulgader)}$	26	00:00:01	.741	.00714	4.32e-11	1.00e-12	0
4	$b_{(asd)}$	8	00:00:00	.621	.0132	.0129	.00100	6
5	$b_{(hosam)}$	7	00:00:00	.403	.0179	.000658	1.00e-05	6
6	$b_{(jadi)}$	9	00:00:00	1.09	.0133	.0132	.100	6
7	$b_{(P0800238x)}$	32	00:00:00	.454	.00714	9.93e-12	1.00e-12	0
8	$b_{(ssomdah)}$	13	00:00:00	.258	6.24e-20	9.26e-11	1.00e-14	0

Additional experiment provides additional prove to explore HW influence in recognizing user behaviour by determining the similarity in profiling the user when particular HW is used. In this experiment neural network is doing training for the input data in two different HW using one hidden layer and three neurons. In this experiment, the neural network is learning user pattern when "password" keys are typed for forty five successful log-in attempts using the two different HW. User "password" keystrokes are interred to the neural network and the target are two categories which are user pattern in every (HW1, HW2). As result of neural network separate users pattern in two categories. First category is the user pattern in using first HW and

the second category for the second. Figure 7.15 show the input and output steps of the neural network to train user pattern and explore different HW pattern

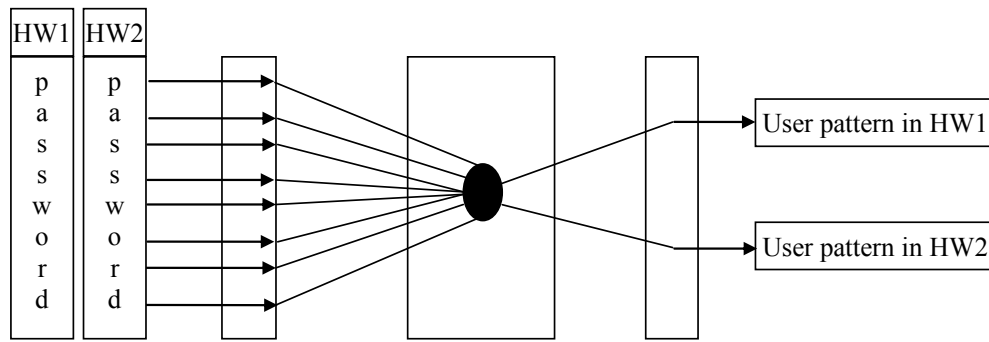


FIGURE 7.15: Neural Network Analysis Two Hardware Recognizing user Patterns

When network learn user pattern in particular HW the next experiment is testing the neural network by typing new log-in attempt to show the HW influence in profiling the user. In this testing the user inputs new data "password keystrokes" and the output is neural network determining which HW been used. Figure 7.16 shows the input and output steps of the neural network test to determine user HW that been used in the tested data.

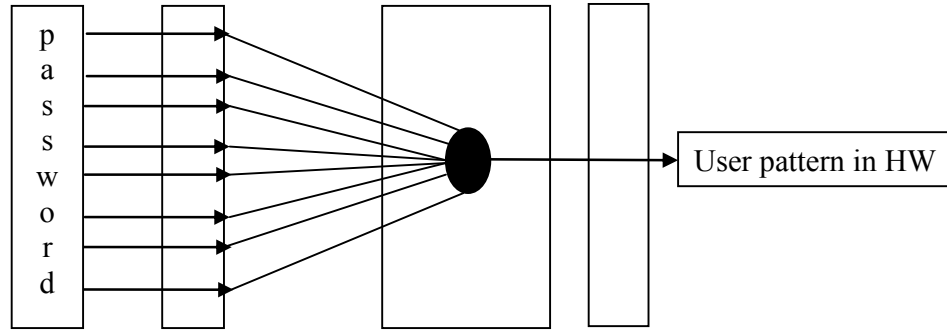


FIGURE 7.16: Neural Network recognizes which Hardware is Used

The neural network function that learn user pattern and then test user behaviour using ten-fold cross validation [144] is available in appendix (D). This function recognize user pattern from the key strokes of forty five successful log-in attempt using the two HW and using another five attempt to test the user behaviour and bind between user behaviour and pattern based on particular HW. For example, when the user types the "password" keys using the first PC1 so the result of recognizing which PC been used should give PC1 high test value (PC1=.7 and PC2= .3). In this experiment the tested data is coming from HW1 Figure 7.17 shows neural network analysis when the user $b_{(4400773)}$ pattern is learned tested.

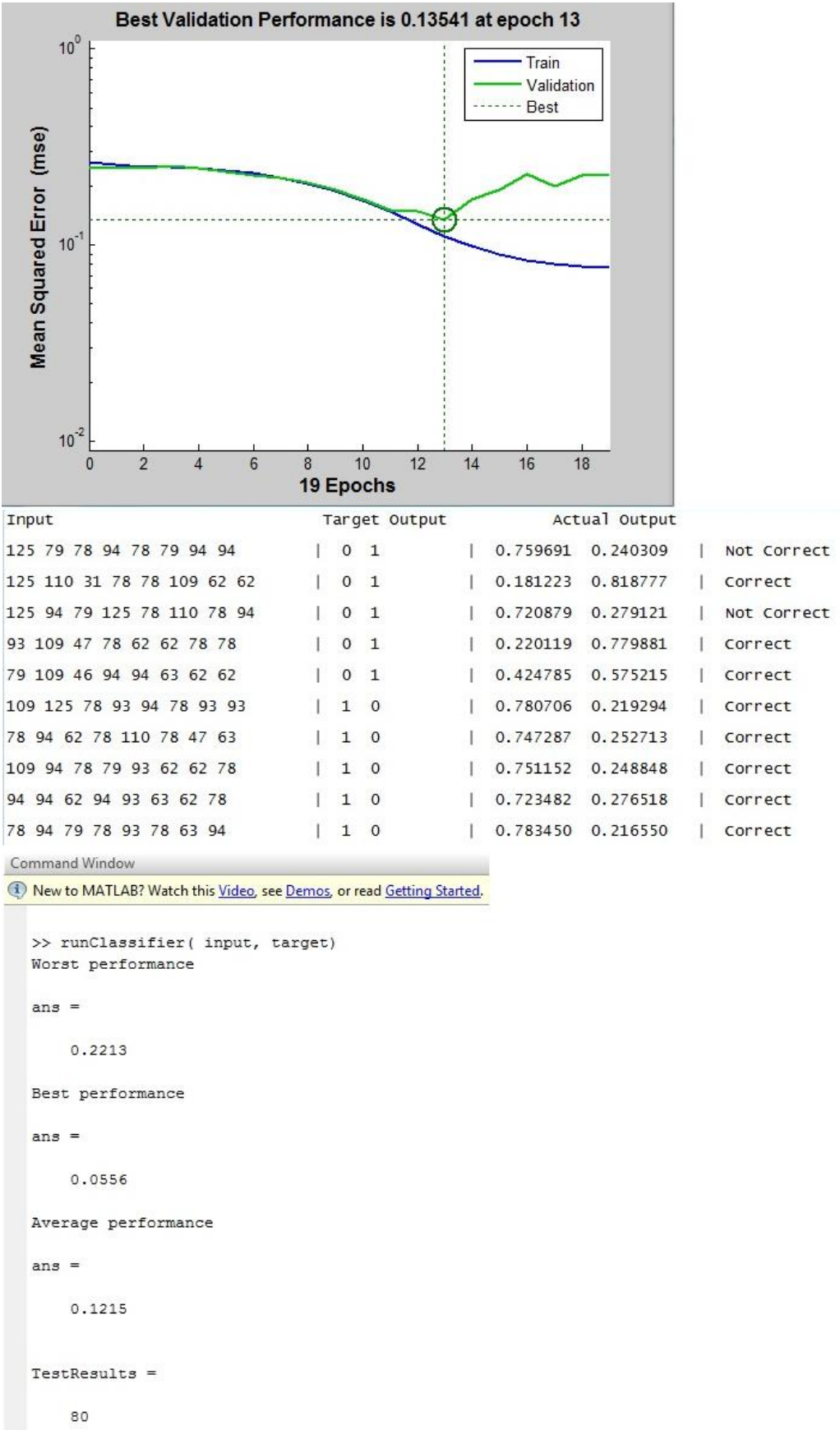


FIGURE 7.17: User 4400773 Recognizing And Testing

Figure 7.18 shows another user analysis result. In this experiment neural network analysis provides two invalid test result (third and forth test). However the rest of the result prove the user pattern is related to the valid pattern in the correct HW.

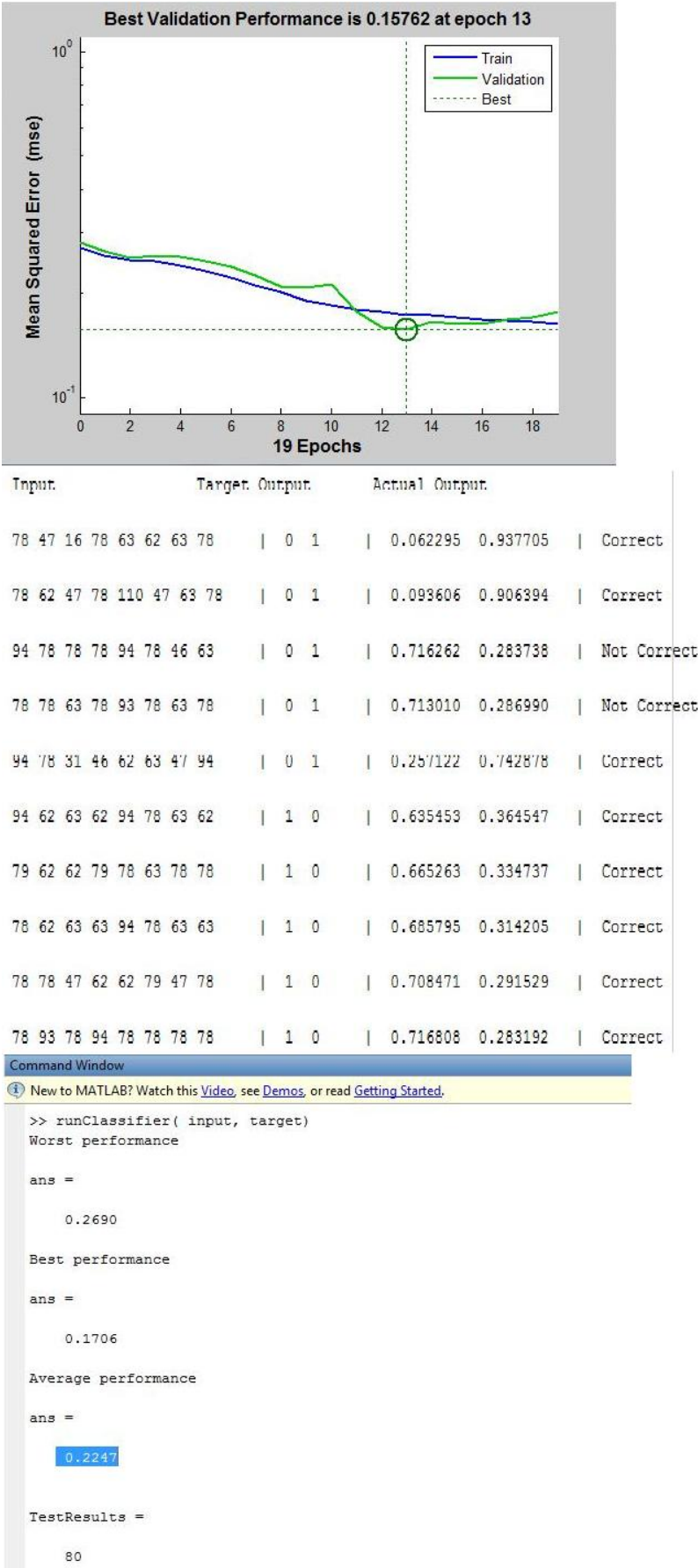


FIGURE 7.18: User abdull Recognizing And Testing

Table 7.4 shews the result of recognizing user behaviour and provide the best performance to classify user pattern based on particular HW. Then, the group of users testing result is provided after learning user behaviour for ten time. The testing result in the table is showing the percentage of correction cases after ten time learning attempts.

TABLE 7.4: Analysing And Testing Users Pattern Using Neural Network

#	User name	Worst Performance	Best Performance	Average	Test
1	$b_{(4400773)}$	0.236	0.008	0.107	80%
2	$b_{(abdul)}$	0.2690	0.1706	0.2247	80%
3	$b_{(abdulgader)}$	0.2721	5.2885e-008	0.1316	90%
4	$b_{(asd)}$	0.2872	0.2178	0.2495	60%
5	$b_{(hosam)}$	0.2539	0.0962	0.1857	80%
6	$b_{(jadi)}$	0.4536	0.1496	0.2268	70%
7	$b_{(P0800238x)}$	0.3329	0.0769	0.2152	70%
8	$b_{(ssomdah)}$	0.4506	0.1783	0.2390	80%

7.7 Summary

HW observation has affected the profiling of user behaviour biometrics based on success criteria 7.1. This influence comes as a result of changing the users computer HW context that clarified in section 7.2. The HAUP observation has developed user profiling based on recognizing the users HW context and log-in time which demonstrated in section 7.4. This HAUP result clarifies the familiarity of using a particular computer's

HW. In addition, the HAUP technique profiles users behaviour based on recognizing a particular. Finally, section 7.6 clarified that profiling users HW has significant characteristics about a users identity to support user authentication in the access control threshold that.

From the previous set of experiments is noted the HW influence in profiling user behaviour, e.g, user's performance in the first and second tables 7.1 and shows different analysis to train users behaviour when the users use different HW.

HW information has significant factors to profile the user and by comparing this with current authentication approach, this work has three main criteria which is used to improve normal trust methods. Firstly, current authentication and profiling techniques depend on system delay to capture user behaviour; however the HAUP method captures user behaviour in an access control threshold which is stored before moving into the system services. Secondly, observing user behaviour should be affected by the user environment HW. So, using behaviour biometrics may not determine user behaviour if the user environment was not recognized. Thirdly, there is a low cost associated with the implementation of MFA which the HAUP the prototype does not need as it does not require any additional accessories to carry out the multi-authentication approach.

The next chapter summaries this research and provides the

conclusion. It also explains the main results and achievements of using HW information as the token key in the MFA approach. The next chapter also discusses the main requirements to improve this authentication approach and mentions future work.

Chapter 8

Conclusion

Objectives

1. Achievements.
 2. Contribution
 3. Limitations.
 4. Future work.
-

This chapter provides the conclusion to this research work by presenting the main achievements and contributions and highlighting the limitations experienced and discussing future work. Section 8.1 highlights the achievements of this work and provides a contribution approach in authentication to support the authentication in access control. Section 8.2 addresses authentication development when hardware devices profile the user to discuss the contribution to the authentication in access control. Section 8.3 provides a review of the success criteria of the HAUP approach, followed by the main limitations of this approach in Section 8.4. Finally, section 8.5 confers the potential for further development.

8.1 Achievements

This research clarified some of the limitations and difficulties in current authentication factors and approaches in chapter 2. This research explored HW information availability and user's behaviour to provide an authentication approach based on this available information. This research described HW information and provided a brief history of HW usage in security proposes. So, this research analysed authentication HW information to explore profiling physical behaviour based on particular computer HW and carried out an analysis and design-activities to model the HW authentication system.

This research provided a framework in Chapter 3 to use HW information in profiling user behaviour. Chapter 6 discussed HAUP prototype implementation which is using profiling techniques to present a trust-model that takes into account users HW information and behaviour when the "username and password" keys are typed. The HAUP prototype is of course a proof of concept that shows that the techniques can be combined and that their combination yields a positive influence on the accuracy of the detection. This research provided a java-based prototype implementation of the HAUP authentication system and presented a set of experiments as a proof of concept for this work.

This research presented the HAUP profiling technique in a prototype system. This research evaluated the HAUP approach by comparing HAUP with MFA approaches. Then, Chapter 7 focused on the user's manufacturer serial part numbers as the user environment to provide a high level of confidence in profiling a user at the access control threshold. This authentication approach is evaluated by implementing the HAUP prototype to get simple results and compared the HAUP result with the neural network analysis to recognize user behaviour and patterns in a particular computer HW.

8.2 Contribution

To answer the research questions Q.1 and Q.2, an automated MFA HW and biometric behaviour authentication system has been built and tested based on the framework throughout the research in Chapters 6 and 7. This MFA authentication is the HAUP prototype which is the subject of this work, the objective of which was to investigate the integration of the HW signature with user behaviour. The results of this analysis are answer questions Q.3 and Q.4 and gives a detailed breakdown of the methodology of the HW approach in Chapters 3 and 4. This approach achieved a better performance that may not have been achievable with single biometric behaviour alone such as, for example, keystroke, and also improves the traditional username and password authentication approach with less cost. These assessments provide the answers to question Q.5 by evaluating the HAUP approach in Chapter 7 and mathematical model in chapter 5.

The experimental investigations, which combined the feature level and decision level fusions, have improved the final authentication performance. This is addressed in question Q.7 by evaluating the HAUP approach. Therefore, it has been shown that the proposed hybrid approach offers considerable improvements to the accuracy of authentication approaches.

As result of analysing and evaluating this work, HW information has significant factors to profile the user and by comparing this with current authentication approach, this work has three main criteria which is used to improve normal trust methods. Firstly, current authentication and profiling techniques depend on system delay to capture user behaviour; however the HAUP method captures user behaviour in an access control threshold which is stored before moving into the system services. Secondly, observing user behaviour should be affected by the user environment HW. So, using behaviour biometrics may not determine user behaviour if the user environment was not recognized. Thirdly, there is a low cost associated with the implementation of MFA which the HAUP the prototype does not need as it does not require any additional accessories to carry out the multi factor authentication approach.

This work has improved the username and password authentication technique and reduce potential fraud by strengthening authentication based on HW authentication. This work has identified the new HW authentication approach in demonstrating that HAUP uses HW manufacturer serial part numbers in MFA form. This work developed authentication methods to enforce available and low-cost resources in order to implement the MFA approach.

The development of this system has improved profiling techniques using the HW information configuration as authentication keys in technology services as first step to profile the user in the HAUP approach. As a result, the traditional username and password authentication approach can be improved to protect the user from potential identity fraud.

The proposed solution is a type of authentication for access to computer services which is presented in the HAUP approach. The HAUP approach can discover a user's behaviour when an illegal access occurs. This is possible with any account when the username and password is used by the hacker. This solution can map the Internet network, even if the total number of HW information reaches into millions and more. This approach combines the password based authentication process with HW profiling and keystroke recognition that then provides an MFA scheme which does not require additional devices to be deployed. In addition, the HAUP approach adds little cost to the deployment authentication approach.

8.3 Success Criteria Revisited

To answer the research questions that we pointed out in Chapter 1, framework for Hardware Authentication and User Profiling has been provided then implemented in prototype and

evaluated throughout the thesis as following:

S.C.R.1. To answer research questions Q.1. Q.2. background about authentication in access control field is covered in Chapter 2. Then, some limitations and difficulties in current authentication approaches are highlighted.

S.C.R.2. To answer research questions Q.3. and Q.4., methodology of proposed framework and system architecture are shown and anglicised in Chapters 3 and 4.

S.C.R.3. To answer the research question Q.5., we implemented HW investigation with user behaviour prototype upon traditional authentication username and password approach in Chapter 6.

S.C.R.4. To answer Q.5. Q.6. and Q.7, we provided set of experiments to evaluate the advantages of integrating HW authentication approach in traditional username and password approach. Then, comparison between HW and current authentication factors is presented and evaluated using neural network in Chapter 7.

8.4 Limitations

Implementing this work requires the user's HW information which can be very difficult to collect from the user's devices

because of the variety of user's operating in the system environment and due to client privacy issues. In addition, during the research steps, the time taken to collect information about the user's data could not provide more HW information. However with more time, more comprehensive analysis could be undertaken. Furthermore, analysing user behaviour requires more components and real user patterns at every successful log-in attempt to observe user behaviour in keystroke typing behaviour.

There are also extensions factors which may affect user behaviour. For example, HW performance may have different values to recognize user patterns. However, HAUP approach factors has a significant influence to recognize user behaviour and the development of the HAUP approach can include user HW performance in the level of trust.

8.4.1 Ways in which the solution might fail

This work shown that, the availability of the HW information configuration on its own could enhance security and trust. However the work presented in this research show that by capturing a wide range of HW information is possible to perform an analysis of behaviour characteristics. The prototype software has shown the level of trust that can be declared from HW information usage. So, the proposed solution might not

work if approach is not able to get the HW information of the user that requires a specific technique or procedure to get a *HMSPNs*. Furthermore, reading specific HW configuration address procedure may break privacy laws.

8.5 Future Work

In the future development of this research the profiling techniques used will be refined in the HAUP framework and the possibility of implementing techniques based on support vector machines will also be explored.

This research also will also investigate the use of the profile information in attack attribution, as the HW profiles can provide an indication about fraudulent users. In addition, this research will look at geo-spatial information and its integration in the HW recognizer. The idea is that successive log-ins from different geographical areas are not plausible and can indicate fraudulent activity. In this line of investigation future work will also actively deploy honey-pots to further identify behavioural traits of the user. This information can then be used twofold, a) to provide additional attribution information about the attacker, and b) to retrospectively authorise the actions performed if the user is deemed to be genuine. In addition, the

following future work can develop HAUP approach as future possibilities:

FW1 - by sorting more HW information and determining a black list for example, switch and router HW information, the system determines the spoofed HW information by creating HW taxonomy for user's hardware that improve the security in authentication. In addition, the system will check the clients' HW information to authenticate the user. This level of information can be supported to protect users inside a network from the outside and protect any public websites. These public websites provides wide usage that will eventually evaluate this work.

FW2 - to use additional behavioural recognition approaches, e.g., fuzzy language, to find more definition for any user behaviour. In addition, using a Support Vector Machine to hold opposing views between user behaviour that is part of the user pattern and explore hacker's behaviours, as shown in Figure [8.1](#)

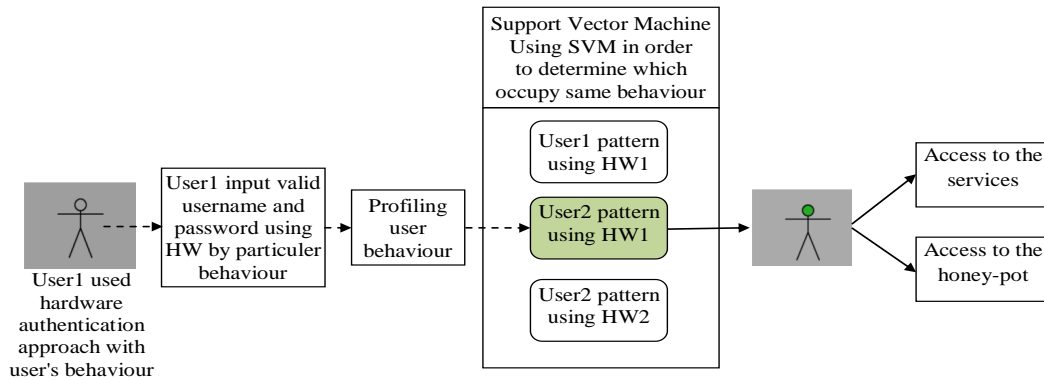


FIGURE 8.1: Support Vector Machine recognizer

FW3 - to extending the username and password authentication keys to long text will provide more opportunities to observe user behaviour as opposed to shorthand. So the HAUP system can recognize user behaviour at the point of log-in. Furthermore, if we refine the individual techniques and adopt, for example, keystroke recognition approaches so as those that have been presented in [73], this can also improve the ability of recognizing user patterns.

FW4 - to implement the HW approach as an authentication protocol in a network low-level layer which is required as an additional procedure through the network. For example, the system should determine how many hardware parts are mandatory to reflect user behaviour. The system should determine whether this hardware and user behaviour information will be carried and transferred in the header of each packet. The encryption store and save method of hardware information

should be secure. The HAUP approach can develop the HW authentication technique to be used as authentication protocol.

FW5 - to use hardware information in forensics by neural network analysis which is capable of solving problems related to patterns by using several techniques such as clustering, classification and generalising. They are also able to predict future events on the basis of events that have occurred in the past [145, 146]. These abilities may be useful for forensics where they can be used to collect evidence after a crime has been committed; e.g., HW information. Classification is used to distinguish between two items based on the degree of similarity between them, such as the distinction between legal and illegal transactions. Therefore, classification is a helpful algorithm for investigators as this will enable them to determine illegal activities that have been conducted within the system [147]. In addition, clustering is used to group data in accordance with resemblances among aspects and characteristics, e.g., matching a group of users who have used similar HW or a group of users which share similarities behaviour [124]. Thus, clustering may have benefits for analysts who wish to group similar unauthorised techniques on a particular system or systems. Grouping crimes in this way makes it easier to deal with a new attack which is similar to earlier ones, because these have been investigated and analysed.

Bibliography

- [1] L. Klander and J. Press, *Hacker proof: the ultimate guide to network security*. Taylor and Francis, 1998.
- [2] D. Muramatsu, M. Kondo, M. Sasaki, S. Tachibana, and T. Matsumoto, “A markov chain monte carlo algorithm for bayesian dynamic signature verification,” *Information Forensics and Security, IEEE Transactions on*.
- [3] Kaushik, “How to interpret hard disk model numbers,” *Instant fundas*, August 2007. [Online]. Available: <http://www.instantfundas.com/2009/02/how-to-interpret-hard-disk-model.html>
- [4] M. Corporation, “Deciphering the bios serial number,” <http://www.hardwaresecrets.com/article/34>, January 1999. [Online]. Available: <http://www.hardwaresecrets.com/article/34>

- [5] W.-C. Ku and S.-M. Chen, “Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards,” *Consumer Electronics, IEEE Transactions on*, vol. 50, no. 1, pp. 204–207, 2004.
- [6] T. Newburn, *Handbook of policing*. Willan, 2012.
- [7] J. Yan, “A note on proactive password checking,” in *Proceedings of the 2001 workshop on New security paradigms*. ACM, 2001, pp. 127–135.
- [8] S. Marechal, “Advances in password cracking,” *Journal in computer virology*, vol. 4, no. 1, pp. 73–81, 2008.
- [9] Jones and Bartlett, “Elementary information security,” *Crypto Smith*, February 1997. [Online]. Available: <http://www.cryptosmith.com/node/219>
- [10] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. springer, 2009.
- [11] M. Blaze, J. Ioannidis, and A. D. Keromytis, “Experience with the keynote trust management system: Applications and future directions,” in *Trust Management*. Springer, 2003, pp. 284–300.
- [12] E. B. Wilson, *An introduction to scientific research*. Courier Dover Publications, 1952.

- [13] E. Labro and T. Tuomela, “On bringing more action into management accounting research: process considerations based on two constructive case studies,” *European Accounting Review*, vol. 12, no. 3, pp. 409–442, 2003.
- [14] M. Fowler, *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional.
- [15] K. Lee, K. Lee, J. Byun, S. Lee, H. Ahn, and K. Yim, “Extraction of platform-unique information as an identifier,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 3, no. 4, pp. 85–99, 2012.
- [16] L. C. Attila M, Zoltn B, “Strengthening passwords by keystroke dynamics,” *IEEE*, 2007. [Online]. Available: www.knt.vein.hu
- [17] D. Nordell, “Terms of protection: The many faces of smart grid security,” *Power and Energy Magazine, IEEE*, vol. 10, no. 1, pp. 18–23, 2012.
- [18] W. Stallings, L. Brown, M. Bauer, and M. Howard, *Computer security: principles and practice*. Pearson Prentice Hall, 2008.
- [19] G. Koien, “Access security in 3gpp-based mobile broadband systems,” 2010.

- [20] R. Chandramouli, “Determining authentication strength for smart card-based authentication use cases,” in *ICDS 2012, The Sixth International Conference on Digital Society*, 2012, pp. 153–158.
- [21] S. Chiasson, P. Van Oorschot, and R. Biddle, “A usability study and critique of two password managers,” in *15th USENIX Security Symposium*, 2006, pp. 1–16.
- [22] A. Gelb and C. Decker, “Cash at your fingertips: Biometric technology for transfers in developing and resource-rich countries,” *Center for Global Development Working Paper*, no. 253, 2011.
- [23] L. O’Gorman, “Comparing passwords, tokens, and biometrics for user authentication,” *Proceedings of the IEEE*.
- [24] W. Shen, M. Surette, and R. Khanna, “Evaluation of automated biometrics-based identification and verification systems,” *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1464–1478, 1997.
- [25] V. Matys and Z. Riha, “Toward reliable user authentication through biometrics,” *IEEE Security and Privacy*, pp. 45–49, 2003.
- [26] R. Song, “Advanced smart card based password authentication protocol,” *Computer Standards and Interfaces*, vol. 32, no. 5, pp. 321–325, 2010.

- [27] P. Singh and G. Thakur, “Enhanced password based security system based on user behavior using neural networks,” *International Journal of Information*, vol. 4, 2012.
- [28] B. Brandherm, J. Hauptert, A. Kroner, M. Schmitz, and F. Lehmann, “Roles and rights management concept with identification by electronic identity card,” in *Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010 8th IEEE International Conference on. IEEE, 2010, pp. 768–771.
- [29] C. Chang and I. Lin, “Remarks on fingerprint-based remote user authentication scheme using smart cards,” *ACM SIGOPS Operating Systems Review*, vol. 38, no. 4, pp. 91–96, 2004.
- [30] A. Poller, U. Waldmann, S. Vowé, and S. Türpe, “Electronic identity cards for user authentication promise and practice,” *IEEE Security and Privacy*, vol. 10, no. 1, p. 46, 2012.
- [31] V. Sassone, “Sid: An exploration of superidentity,” 2012. [Online]. Available: <http://www.southampton.ac.uk/cybersecurity/projects/sid.page?#overview>

- [32] D. Hodges, S. Creese, and M. Goldsmith, “A model for identity in the cyber and natural universes,” in *Intelligence and Security Informatics Conference (EISIC), 2012 European*, aug. 2012, pp. 115–122.
- [33] D. H. Sadie Creese, “Oxford university, computer science, super identity,” 2012.
- [34] A. W. Naji, A. S. Housain, B. B. Zaidan, A. A. Zaidan, and S. A. Hameed, “Security improvement of credit card online purchasing system,” *Scientific Research and Essays*, vol. 6(16), pp. 3357–3370, 2011.
- [35] A. Naji, A. Housain, B. Zaidan, A. Zaidan, and S. Hameed, “Security improvement of credit card online purchasing system,” *Scientific Research and Essays*, vol. 6, no. 16, pp. 3357–3370, 2011.
- [36] E. Council, “Federal financial institutions examination council,” *Stat*, vol. 2160, pp. 22–50, 1994.
- [37] M. Hwang, S. Chong, and T. Chen, “Dos-resistant id-based password authentication scheme using smart cards,” *Journal of Systems and Software*, vol. 83, no. 1, pp. 163–172, 2010.
- [38] X. Li, J. Niu, J. Ma, W. Wang, and C. Liu, “Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards,” *Journal*

- of Network and Computer Applications*, vol. 34, no. 1, pp. 73–79, 2011.
- [39] S. Wu and Y. Zhu, “Improved two-factor authenticated key exchange protocol,” *The International Arab Journal of Information Technology*, vol. 8, no. 4, pp. 430–439, 2011.
- [40] D. Kumar, Y. Ryu, and D. Kwon, “A survey on biometric fingerprints: The cardless payment system,” in *Biometrics and Security Technologies, 2008. ISBAST 2008. International Symposium on*. IEEE, 2008, pp. 1–6.
- [41] H.-D. Ihmaidi, A. Al-Jaber, and A. Hudaib, “Securing online shopping using biometric personal authentication and steganography,” in *Information and Communication Technologies, 2006. ICTTA’06. 2nd*, vol. 1. IEEE, 2006, pp. 233–238.
- [42] D. Parameswari and L. Jose, “Set with sms otp using two factor authentication,” *Journal of Computer Applications (JCA)*, vol. 4, no. 4, 2011.
- [43] J. Brainard, A. Juels, R. Rivest, M. Szydlo, and M. Yung, “Fourth-factor authentication: somebody you know,” in *Conference on Computer and Communications Security: Proceedings of the 13 th ACM conference on Computer and communications security*, vol. 30, 2006, pp. 168–178.

- [44] S. Ravi, P. C. Kocher, G. Lee, Ruby B. McGraw, and A. Raghunathan, “Security as a new dimension in embedded system design.” in *DAC*, S. Malik, L. Fix, and A. B. Kahng, Eds. ACM, 2004, pp. 753–760. [Online]. Available: <http://dblp.uni-trier.de/db/conf/dac/dac2004.html/RaviKLMR04>
- [45] M. Alzomai and A. Josang, “The mobile phone as a multi otp device using trusted computing,” in *Proceedings of the Fourth International Conference on Network and System Security*. IEEE Computer Society.
- [46] T. D. Ajakaiye and K. S. K. Krause, “Online based authentication and secure payment methods for m-commerce applications,” 2011.
- [47] F. Aloul, S. Zahidi, and W. El-Hajj, “Two factor authentication using mobile phones,” in *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*. IEEE, 2009, pp. 641–644.
- [48] A. Smith, “Online payment service providers and customer relationship management,” *International Journal of Electronic Finance*, vol. 2, no. 3.
- [49] P. Krishnamurthy and M. Redddy, “Implementation of atm security by using fingerprint recognition and gsm.”

- [50] J. Cheng and J. Tian, “Fingerprint enhancement with dyadic scale-space,” *Pattern Recognition Letters*, vol. 25, no. 11, pp. 1273–1284, 2004.
- [51] J. Gu, J. Zhou, and D. Zhang, “A combination model for orientation field of fingerprints,” *Pattern Recognition*, vol. 37, no. 3, pp. 543–553, 2004.
- [52] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, “Handbook of fingerprint recognition,” *New York*, 2003.
- [53] H. Berghel, “Identity theft and financial fraud: Some strangeness in the proportions,” *Computer*.
- [54] J. Potts, *Computer Security: A Bibliography with Indexes*. Nova Science Pub Incorporated, 2002.
- [55] M. Support, “Microsoft corporation user profiles overview,” March 2010. [Online]. Available: http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/userprofile_overview.mspx?mfr=true
- [56] G. Carat, E. C. J. R. Centre, and I. for Prospective Technological Studies, *ePayment Systems database*. European Commission Joint Research Centre, 2002.
- [57] V. Oliveira and T. Silva, “The power of credit card numbers and long cvvs.” in *NSS*, P. Samarati, S. Foresti, J. Hu, and G. Livraga, Eds. IEEE, 2011, pp. 290–294.

- [Online]. Available: <http://dblp.uni-trier.de/db/conf/nss/nss2011.html#OliveiraS11>
- [58] S. Dharwadkar and N. Masood, "Next generation network," in *Consumer Electronics, 2007. ISCE 2007. IEEE International Symposium on*. IEEE, 2007, pp. 1–4.
- [59] C. R., "Integrated user profiles and service profiles," *Journal of Telecommunications Management*, p. 411, April 2008.
- [60] I. Nikolaidis, "Network security essentials: applications and standards," *Network, IEEE*, vol. 14, no. 2, pp. 6–6, 2000.
- [61] N. Zheng, A. Paloski, and H. Wang, "An efficient user verification system via mouse movements," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 139–150.
- [62] J. Stewart, M. Chapple, and D. Gibson, *CISSP: Certified Information Systems Security Professional Study Guide*. Sybex, 2004.
- [63] A. Ahmed and I. Traore, "Anomaly intrusion detection based on biometrics," in *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*. IEEE, 2005, pp. 452–453.

- [64] M. Brain, “How internet cookies work,” [Online]. Retrieved from the Internet, vol. 7, 2003.
- [65] T. Steindel, “A path toward user control of online profiling,” *Mich. Telecomm. Tech. L. Rev.*, vol. 17, pp. 459–491, 2011.
- [66] A. Gonsalves, “Company bypasses cookie-deleting consumers,” *Information Week (March 2005)*, 2009.
- [67] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. Hoofnagle, “Flash cookies and privacy,” *SSRN eLibrary*, 2009.
- [68] M. Abraham, C. Meierhoefer, and A. Lipsman, “The impact of cookie deletion on the accuracy of site-server and ad-server metrics: An empirical comscore study,” *Retrieved October*, vol. 14, p. 2009, 2007.
- [69] D. Shanmugapriya and G. Padmavathi, “A survey of biometric keystroke dynamics: Approaches, security and challenges,” *CoRR*, vol. abs/0910.0817. [Online]. Available: <http://dblp.uni-trier.de/db/journals/corr/corr0910.html#abs-0910-0817>
- [70] F. Bergadano, D. Gunetti, and C. Picardi, “User authentication through keystroke dynamics.” *ACM Trans. Inf. Syst. Secur.*, no. 4. [Online]. Available: <http://dblp.uni-trier.de/db/journals/tissec/tissec5.html/BergadanoGP02>

- [71] N. L. Clarke and S. Furnell, "Authenticating mobile phone users using keystroke analysis." *Int. J. Inf. Sec.*, vol. 6, no. 1, pp. 1–14, 2007. [Online]. Available: <http://dblp.uni-trier.de/db/journals/ijisec/ijisec6.html/ClarkeF07>
- [72] H. Lee and S. Cho, "Retraining a keystroke dynamics-based authenticator with impostor patterns." *Computers and Security*, vol. 26, no. 4, pp. 300–310, 2007. [Online]. Available: <http://dblp.uni-trier.de/db/journals/compsec/compsec26.html/LeeC07>
- [73] M. S. Obaidat and B. Sadoun, "Keystroke dynamics based authentication," in *In and quot;Biometrics. Personal Identification in Networked Society and quot;.* A.Jain, R.Bolle, S.Pankanti Eds. Kluwer Academic Publishers.
- [74] J. Wayman, "Error rate equations for the general biometric system," *Robotics and Automation Magazine, IEEE*, vol. 6, no. 1, pp. 35–48, 1999.
- [75] D. Hosseinzadeh, S. Krishnan, and A. Khademi, "Keystroke identification based on gaussian mixture models," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 3. IEEE, 2006, pp. III–III.

- [76] L. H.R. and W. W.Y., “Biologic verification based on pressure sensor keyboards and classifier fusion techniques,” *Consumer Electronics, IEEE Transactions on*, vol. 52, no. 3.
- [77] S. Sahoo, T. Choubisa *et al.*, “Multimodal biometric person authentication: A review,” *IETE Technical Review*, vol. 29, no. 1, p. 54, 2012.
- [78] A. Jain, R. Bolle, and S. Pankanti, *Biometrics: personal identification in networked society*. kluwer academic publishers.
- [79] L. Ern and G. Sulong, “Fingerprint classification approaches: An overview,” in *Signal Processing and its Applications, Sixth International, Symposium on. 2001*. IEEE.
- [80] P. S. Teh, A. B. J. Teoh, C. Tee, and T. S. Ong, “Keystroke dynamics in password authentication enhancement,” *Expert Syst. Appl.*, vol. 37, pp. 8618–8627, December 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.eswa.2010.06.097>
- [81] K. MAXION, R. KILLOURHY, “Keystroke biometrics with number pad input.” *International Conference on Dependable Systems and Network. IEEE.*, no. 12, pp. 101–210, 2010.

- [82] Y. Sheng, V. Phoha, and S. Rovnyak, "A parallel decision tree-based method for user authentication based on keystroke patterns," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 35, no. 4, pp. 826–833, 2005.
- [83] D. Gunetti and C. Picardi, "Keystroke analysis of free text." *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 3, pp. 312–347, 2005. [Online]. Available: <http://dblp.uni-trier.de/db/journals/tissec/tissec8.html/GunettiP05>
- [84] F. Monrose, M. K. Reiter, and S. Wetzal, "Password hardening based on keystroke dynamics." in *ACM Conference on Computer and Communications Security*, J. Motiwalla and G. Tsudik, Eds. ACM, 1999, pp. 73–82. [Online]. Available: <http://dblp.uni-trier.de/db/conf/ccs/ccs1999.html/MonroseRW99>
- [85] E. Yu and S. Cho, "Keystroke dynamics identity verification - its problems and practical solutions." *Computers and Security*, vol. 23, no. 5, pp. 428–440, 2004. [Online]. Available: <http://dblp.uni-trier.de/db/journals/compsec/compsec23.html/YuC04>
- [86] W. De Ru and J. Eloff, "Enhanced password authentication through fuzzy logic," *IEEE Expert*, vol. 12, no. 6, pp. 38–45, 1997.

- [87] F. Wong, A. Supian, A. Ismail, L. Kin, and O. Soon, "Enhanced user authentication through typing biometrics with artificial neural networks and k-nearest neighbor algorithm," in *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*, vol. 2. IEEE, 2001, pp. 911–915.
- [88] R. Maxion and K. Killourhy, "Keystroke biometrics with number-pad input," in *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*.
- [89] P. Teh, A. Teoh, C. Tee, and T. Ong, "Keystroke dynamics in password authentication enhancement," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8618–8627, 2010.
- [90] D. DSouza, "Typing dynamics biometric authentication," *Department of Information Technology and Electrical Engineering, University of Queensland. Bachelor of Engineering in the Division of Software Engineering*, 2002.
- [91] F. Monroe and A. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Systems*, vol. 16, no. 4.
- [92] H.-R. Lv and W.-Y. Wang, "Biologic verification based on pressure sensor keyboards and classifier fusion techniques," *Consumer Electronics, IEEE Transactions on*, no. 3, aug.

- [93] C. Mandia, K. Prosis and M. Pepe, *Incident response and computer forensic*. Hightstown, United States of America: McGraw-Hill Companies, Inc, 2003.
- [94] D. Senie and P. Ferguson, “Network ingress filtering: defeating denial of service attacks which employ ip source address spoofing,” *Network*, 1998.
- [95] S. Inc, *Security complete*. Sybex, 2001.
- [96] J. Zedlewski, S. Sobti, N. Garg, F. Zheng, A. Krishnamurthy, R. Wang *et al.*, “Modeling hard-disk power consumption,” in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, vol. 28, 2003, pp. 32–72.
- [97] R. P. R, *Corporate computer and network security*. New Jersey United States of America: Pearson Education, Inc, 2004.
- [98] C. Wang and H. Leung, “A private and efficient mobile payment protocol,” *Computational Intelligence and Security*.
- [99] J. Tellez and J. Sierra, “Anonymous payment in a client centric model for digital ecosystem,” *IEEE DEST*, pp. 422–427, 2007.

- [100] M. Rajalingam, S. Alomari, and P. Sumari, "Prevention of phishing attacks based on discriminative key point features of webpages," *International Journal of Computer Science and Security (IJCSS)*, vol. 6, no. 1, p. 527, 2012.
- [101] K. Patowary. (2009) How to interpret hard disk model numbers. [Online]. Available: <http://www.instantfundas.com/2009/02/how-to-interpret-hard-disk-model.html>
- [102] Microsoft Corporation, "Microsoft office activation/registration privacy statement." [Online]. Available: <http://o.ce.microsoft.com/en-us/help/HP010069531033.aspx>
- [103] M. Corporation, "Microsoft office activation/registration privacy statement," *Microsoft office*, January 2010. [Online]. Available: <http://office.microsoft.com/en-us/help/HP010069531033.aspx>
- [104] S. Malik, *Network Security Principles and Practices*. 800 East 96th Street Indianapolis, USA: Cisco Press logo of Cisco System, Inc, 2003.
- [105] C. Mongoho, "Mac based wireless authentication with ias," *Techre public*. [Online]. Available: [http://techrepublic.com.com/5208-7343-0.html?forumID=102&threadID=226120&start=0&tag=content;leftColm,](http://techrepublic.com.com/5208-7343-0.html?forumID=102&threadID=226120&start=0&tag=content;leftColm)

- [106] J. Pang, B. Greenstein, R. Gummadi, S. Srinivasan, and D. Wetherall, “802. 11 user fingerprinting,” in *International Conference on Mobile Computing and Networking: Proceedings of the 13 th annual ACM international conference on Mobile computing and networking*, vol. 9, no. 14, 2007, pp. 99–110.
- [107] I. Graham and W. Joseph, “Authenticating public access networking,” in *Proceedings of the 30th annual ACM SIGUCCS conference on User services*. ACM.
- [108] O. Corre, I. Fodil, V. Ksinant, and G. Pujolle, “An architecture for access network management with policies (an-pbm),” in *Management of Multimedia Networks and Services*, ser. Lecture Notes in Computer Science, A. Marshall and N. Agoulmine, Eds. Springer Berlin Heidelberg, 2003, vol. 2839, pp. 328–340. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-39404-4_25
- [109] D. Fye, “Evolution of wlan roaming services,” in *CDG WLAN Technical Forum, Dallas, Texas*, vol. 2, 2003.
- [110] F. A. and B. Levy, “Network security device which performs mac address translation without affecting the ip address,” uS Patent 5,757,924.
- [111] P. Hamalainen, M. Hannikainen, M. Niemi, and T. Hamalainen, “Performance evaluation of secure remote

- password protocol,” in *Circuits and Systems, 2002 IEEE International Symposium on*, vol. 3. IEEE, 2002, pp. III–29.
- [112] T. Wu, “The secure remote password protocol,” in *Internet Society Symposium on Network and Distributed System Security*, 1998.
- [113] A. Menezes, P. Van Oorschot, and S. Vanstone, *Handbook of applied cryptography*. CRC, 1996.
- [114] A. Ahmed, “Security monitoring through human computer interaction devices,” Ph.D. dissertation, UNIVERSITY OF VICTORIA.
- [115] E. Ferreira, D. Milori, E. Ferreira, R. Da Silva, and L. Martin-Neto, “Artificial neural network for cu quantitative determination in soil using a portable laser induced breakdown spectroscopy system,” *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. 63, no. 10, pp. 1216–1220, 2008.
- [116] E. Chudler, “A computer in your head?” *Odyssey Magazine*, vol. 10, pp. 6–7, 2001.
- [117] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.
- [118] M. Dubin, *How the brain works*. Blackwell Publishing, 2002.

- [119] B. Ripley, *Pattern recognition and neural networks*. Cambridge university press, 2008.
- [120] K. Shihab, “A backpropagation neural network for computer network security,” *Journal of Computer Science*, vol. 2, no. 9, pp. 710–715, 2006.
- [121] M. Singh, “Password based a generalize robust security system design using neural network,” *Arxiv preprint arXiv:0910.1838*, 2009.
- [122] N. Harun, W. Woo, and S. Dlay, “Performance of keystroke biometrics authentication system using artificial neural network (ann) and distance classifier method,” in *Computer and Communication Engineering (ICCCE), 2010 International Conference on*. IEEE, 2010, pp. 1–6.
- [123] M. Negnevitsky, *Artificial intelligence: a guide to intelligent systems*. Addison-Wesley Longman, 2005.
- [124] J. Mena, *Investigative data mining for security and criminal detection*. Butterworth-Heinemann, 2003.
- [125] R. Richardson, “Csi computer crime and security survey,” *Computer Security Institute*, vol. 1, pp. 1–30, 2008.
- [126] M. Sammany, M. Sharawi, M. El-Beltagy, and I. Saroit, “Artificial neural networks architecture for intrusion detection systems and classification of attacks,” in *fifth international conference-INFO*, 2007, pp. 24–26.

- [127] R. Kemmerer and G. Vigna, “Intrusion detection: a brief history and overview,” *Computer*.
- [128] J. Saltzer and M. Schroeder, “The protection of information in computer systems,” *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, 1975.
- [129] J. Bonifácio Jr, A. Cansian, A. De Carvalho, and E. Moreira, “Neural networks applied in intrusion detection systems,” in *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, vol. 1. IEEE, 1998, pp. 205–210.
- [130] P. Helman and G. Liepins, “Statistical foundations of audit trail analysis for the detection of computer misuse,” *Software Engineering, IEEE Transactions on*, vol. 19, no. 9, pp. 886–901, 1993.
- [131] J. Cannady, “Artificial neural networks for misuse detection,” in *National information systems security conference*, 1998.
- [132] R. E. Crossler, A. C. Johnston, P. B. Lowry, Q. Hu, M. Warkentin, and R. Baskerville, “Future directions for behavioral information security research,” *Computers and Security*, 2012.

- [133] H. J. A. AL-Najjar, “A value sensitive design investigation of privacy enhancing tools in web browsers,” 2012.
- [134] M. C. A. Kioon, Z. Wang, and S. D. Das, “Security analysis of md5 algorithm in password storage,” 2013.
- [135] J. Shirazi, *Java performance tuning*. O’Reilly Media, Incorporated, 2003.
- [136] W. Diffie and M. Hellman, “New directions in cryptography,” *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644–654, 1976.
- [137] A. K. Jain, A. Ross, and S. Pankanti, “Biometrics: a tool for information security,” *Information Forensics and Security, IEEE Transactions on*, vol. 1, no. 2, pp. 125–143, 2006.
- [138] A. K. Jain, A. Ross, and S. Prabhakar, “An introduction to biometric recognition,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 1, pp. 4–20, 2004.
- [139] A. A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of multibiometrics*. Springer.
- [140] G. Böhm, R. Muhr, and R. Jaenicke, “Quantitative analysis of protein far uv circular dichroism spectra by neural networks,” *Protein engineering*, vol. 5, no. 3, pp. 191–195, 1992.

- [141] L. Hansen and P. Salamon, "Neural network ensembles," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 10, pp. 993–1001, 1990.
- [142] T. Kurban and E. Beşdok, "A comparison of rbf neural network training algorithms for inertial sensor based terrain classification," *Sensors*, vol. 9, no. 8, pp. 6312–6329, 2009.
- [143] E. Stergiopoulou and N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 8.
- [144] D. West, "Neural network credit scoring models," *Computers and Operations Research*, vol. 27, no. 11, pp. 1131–1152, 2000.
- [145] I. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of microbiological methods*.
- [146] G. Carpenter and S. Grossberg, "The art of adaptive pattern recognition by a self-organizing neural network," *Computer*, vol. 21, no. 3, pp. 77–88, 1988.
- [147] I. Basheer, "Selection of methodology for neural network modeling of constitutive hystereses behavior of soils,"

Computer-Aided Civil and Infrastructure Engineering,
vol. 15, no. 6, pp. 445–463, 2000.

Appendices

A. Collecting HW information code.

B. Full HAUP code.

C. Data base architecture.

D. Matlab Files to Run Neural Network Functions.

E. Published papers.

Collecting hardware information code

1. Collecting Media Access Control serial number.
 2. Collecting Motherboard serial number.
 3. Collecting Hard disk serial number.
-

```
import java.io.*; import java.net.*; import java.util.*;

import java.util.regex.*;

import java.awt.Graphics;

import javax.swing.JApplet;

import java.io.File;

import java.io.FileWriter;

import java.io.BufferedReader;

import java.io.InputStreamReader;

public class GetMac extends JApplet

public static void main(String[] args)

throws IOException

public String getMacAddress() throws IOException

//1. Collecting Media Access Control serial number

String macAddress = null; //String NodeType = null;

String command = "ipconfig /all";

Process pid = Runtime.getRuntime().exec(command);

BufferedReader in = new BufferedReader(new InputStreamReader(
    pid.getInputStream()));
```



```
while (true)

String line = in.readLine();

if (line == null)

break;

Pattern p = Pattern.compile(".*Physical Address.*: (.*)");

Matcher m = p.matcher(line);

if (m.matches())

macAddress = m.group(1);

break;

in.close();

return macAddress;

public static String getMotherboardSN()

String result = "";

//2. Collecting Motherboard serial number

try

File file = File.createTempFile("realhowto", ".vbs");

file.deleteOnExit();

FileWriter fw = new java.io.FileWriter(file);
```

```
String vbs =  
  
"Set objWMIService = GetObject("winmgmts:.root cimv2")"  
  
+"Set    colItems    =    objWMIService.ExecQuery"  
  
+"("Select    *    from    Win32_BaseBoard")"  
  
+ "For Each objItem in colItems n"  
  
+ " Wscript.Echo objItem.SerialNumber "  
  
+ " exit for ' do the first cpu only! "  
  
+ "Next ";  
  
fw.write(vbs);  
  
fw.close();  
  
Process p = Runtime.getRuntime().exec("cscript //NoLogo " + file.getPath());  
  
BufferedReader input = new BufferedReader (new InputStreamReader(p.getInputStream()));  
  
String line;  
  
while ((line = input.readLine()) != null)  
  
result += line;  
  
input.close();
```

```
catch(Exception e)

e.printStackTrace();

return result.trim();

public static String getSerialNumber(String drive)

String result = "";
```

3. Collecting Hard disk serial number

```
try

File file = File.createTempFile("realhowto",".vbs");

file.deleteOnExit();

FileWriter fw = new java.io.FileWriter(file);

String vbs = "Set objFSO = CreateObject( "Scripting.FileSystemObject
")"

+"Set colDrives = objFSO.Drives "

+"Set objDrive = colDrives.item( "" + drive + " )"

+"Wscript.Echo objDrive.SerialNumber"; // see note

fw.write(vbs);

fw.close();

Process p = Runtime.getRuntime().exec("cscript //NoL-
ogo " + file.getPath());
```

```
BufferedReader input =  
    new BufferedReader  
    (new InputStreamReader(p.getInputStream()));  
String line;  
while ((line = input.readLine()) != null)  
    result += line;  
input.close();  
catch(Exception e)  
    e.printStackTrace();  
return result.trim();
```

Full HAUP code

1. Username and password checker code.
 - 2- Observing User hardware.
 3. Observing user keystroke behaviour code.
 4. Recognising user pattern and behaviour code.
 5. Presenting User hardware and behaviour code.
-

The attached CD includes the HAUP code and Java files. These files contain the main classes to be run in the server and user's machine. Some of these class are to collect the hardware information (See appendix (A)) and another class to analyse user hardware and behaviour. Following list are the Java classes including the aim of each class:

1- PhDprogressservicesLog: This java class is implemented in users computer to perform the log-in procedure. This file check the username and password validity, read user hardware and observe user behaviour in typing username-password keys. Inaddition, This file analyse user hardware and behaviour to recognise the relationships between previous and current user hardware. This file analyse user behaviour based on the hardware usage.

2- GetMac: this java class is reading users hardware information(MAC address, HDD, BIOS)

3- KeystrokeTest: This java file represents user behaviour in graph to show user behaviour.

4- LoginTimePlot: This java file represents how often user log-in time is during the hours day.

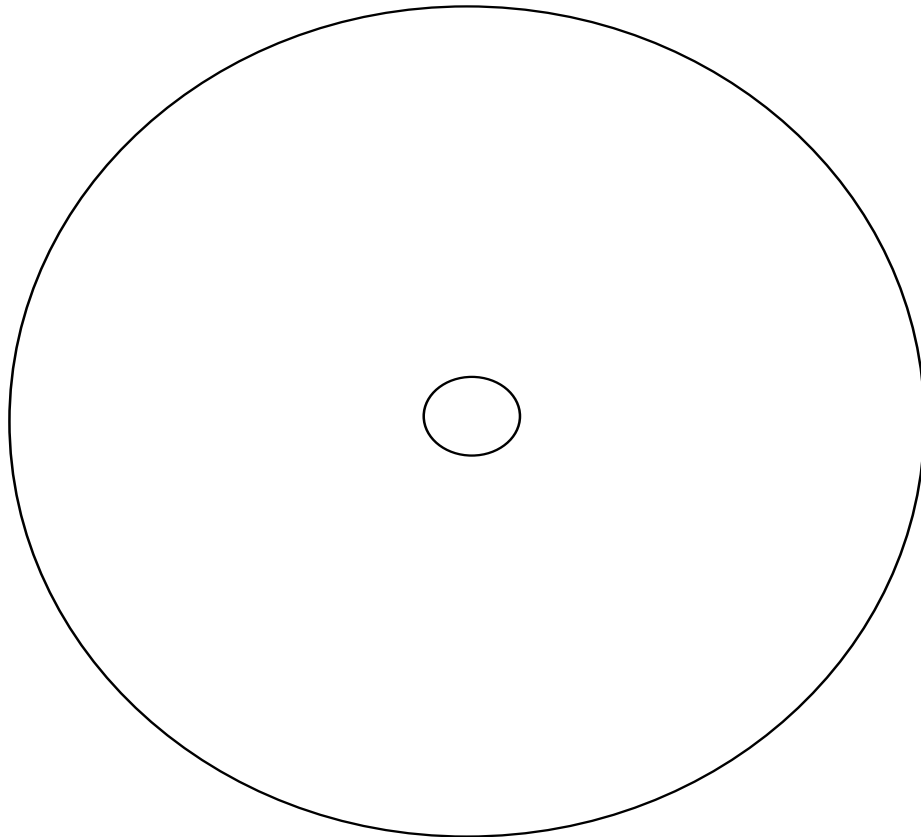
5- LogInDB: This file check the username and password validity.

6- PasswordBehaviour: This file represents user behaviour in typing the password keys.

7- PlotTest: this file show the domain of user pattern when username and password are typed. In addition, this file shows user behaviour in current successful log-in attempt.

8- SaveInDB: this file have the saving procedure to save user hardware and behaviour every successful log-in attempt.

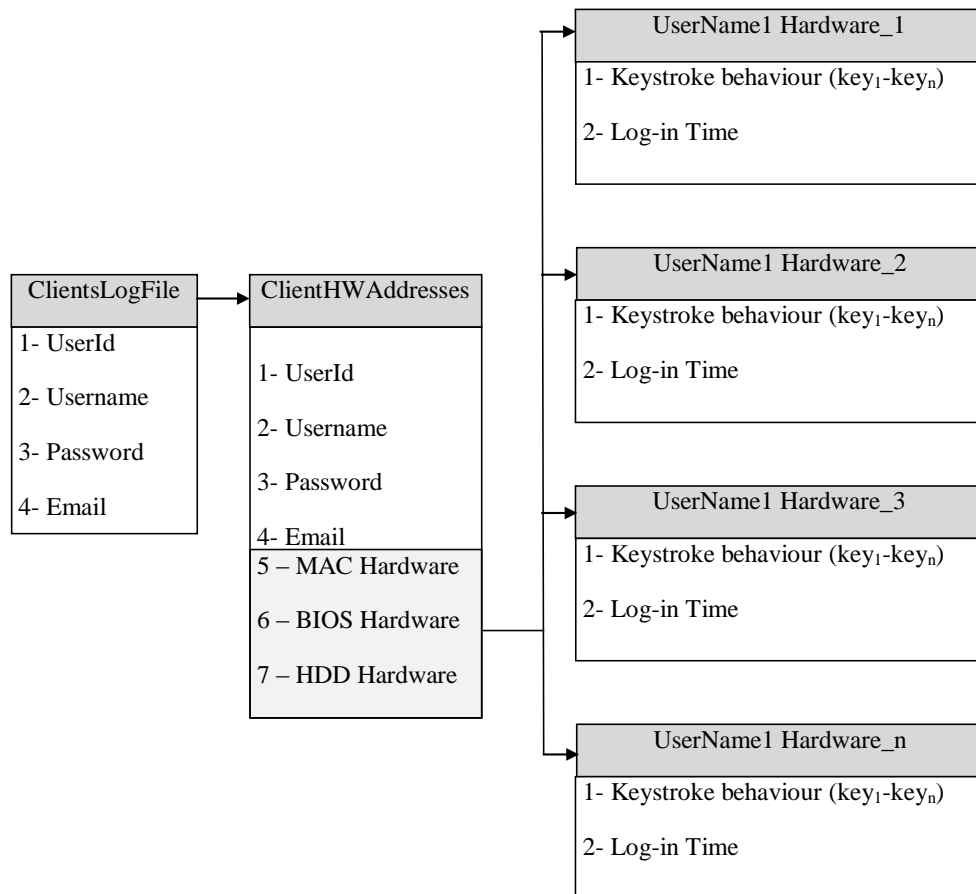
9- OverAllTrustFram: This file represents the overall level of trust for the user log-in attempt.



Data base architecture.

1. Log-in file “ClientsLogFile”.
 - 2- Log file including Users hardware information “ClientHWAddresses”.
 3. User behaviour tables based on hardware for analysis, e.g., “abdulgader-00-1C-C0-6D-6E-AA-128640614-BQJO8280076R”.
-

HAUP prototype database is built by Microsoft Access. This database contain three types of tables to authenticate and profile the user. Firstly, "ClientsLogFile" table will check if the username and password are valid or not. If valid the second table "ClientHWAddresses" checks if the user have used the current hardware information or not. Following figure shows the database structure.



If the user used current hardware before then the system will analyse user hardware and behaviour form the available data, e.g., “abdulgader-00-1C-C0-6D-6E-AA-128640614-BQJO8280076R”. However if the user is using new hardware then the system will build new table for new hardware and behaviour. These tables are:

1. Log-in table “ClientsLogFile”.

In this table ”ClientsLogFile” the system checks the user name and password validity. This table will contain the following fields:

ClientsLogFile	
Field Name	Data Type
CounterAcc	AutoNumber
UserName	Text
Password	Text
UserEmail	Text
LastLog	Text

The Log-in table has the main log-in information. For example, following figure show the user log information.

Custom		ClientsLogFile				
		CounterAcc	UserName	Password	UserEmail	LastLog
asd		2	username	password	myemail@mydomain.cc	0
bbbbbb		3	jadi	password	amr.mh2006@yahoo.coi	0
ClientHWAddresses		32	essam888	password	essam888@gmail.com	0
ClientsLogFile		33	andy	password	abn@dmu.ac.uk	0
emad		37	ssomdah	password	ssomdah@yahoo.com	0
essam888		38	snooh	password	sa.nooh@yahoo.com	0
Hosam		39	p0800238x	password	p0800238x@myemail.dr	0
jadi		51	adeeb	adeeb	adeeb	0
kingmy		52	aaaaaa	aaaaaa	aaaaaa	0
nasa		53	bbbbbb	bbbbbb	bbbbbb	0
p0800238x		54	nasa	nasa	nasa	0
salah		55	abdul	password	aa_alharbi@hotmail.cor	0
salamro		56	salamro	password	sulaiman9949@hotmail.	0
snooh		58	abdulgader	password	abdulgader@gmail.com	0
ssomdah		59	4400773	password	a_barnawi2000@yahoo.	0
username		60	yusefzahrani	password	yosef58@hotmail.com	0
yusefzahrani		62	asd	password	al_asd99@yahoo.com	0
4400773-00-1C-C0-6D-6E-AA-12864...		63	salah	123	SZAAMOUT@HOTMAIL.C	0
4400773-00-1C-C0-E6-38-4C		64	AlHosam	UQqu1402	h.h.alhakam@gmail.con	0
		65	Hosam	password	h.h.alhakam@gmail.con	0
		66	kingmy	kingmy	kingmy	0
		*(New)	0	0	0	0

2- Log file including Users HW information “ClientHWAddresses”.

This table is the main table for HAUP information. This table contains user’s HW and behaviour.

Field Name	Data Type
Key0	Number
Res0	Number
Key1	Number
Res1	Number
Key2	Number
Res2	Number
Key3	Number
Res3	Number
Key4	Number
Res4	Number
Key5	Number
Res5	Number
Key6	Number
Res6	Number
Key7	Number
Res7	Number
Key8	Number
Res8	Number
Key9	Number
Res9	Number
Key10	Number
Res10	Number
Key11	Number
Res11	Number
Key12	Number
Res12	Number
Key13	Number
Res13	Number
Key14	Number
Res14	Number
Key15	Number
Res15	Number
Key16	Number
Res16	Number
Key17	Number
Res17	Number
Key18	Number
Res18	Number
Key19	Number
Res19	Number
Key20	Number
Res20	Number
Key21	Number
Res21	Number
Key22	Number
Res22	Number
Key23	Number
Res23	Number
Key24	Number
Res24	Number
Key25	Number
Res25	Number
Key26	Number
Res26	Number
Key27	Number
Res27	Number
Key28	Number
Res28	Number
Key29	Number
Res29	Number
Key30	Number
MacAddress	Text
StorageMedia	Text
BIOSNo	Text
LogInTime	Number
UserName	Text
Password	Text
UserEmail	Text
LastLog	Text
LogInDate	Number
CurrentDayOfWeek	Number
MillisecondTotalRhythm	Number

Following tables shows example of data when is stored in (ClientHWAddresses) table.

Custom	ClientHWAddresses	Key0	Res0	Key1	Res1	Key2	Res2	Key3	Res3	Key4	Res4	Key5	Res5	Key6	Res6	Key7	Res7
asd		7	0	78	218	78	188	78	125	78	547	125	187	78	266	109	18
bbbbbb		47	0	78	47	78	156	62	110	78	453	109	94	78	63	78	20
ClientHWAddresses		79	0	110	140	62	125	79	109	63	671	141	78	78	15	94	42
ClientLogFile		62	0	109	110	78	141	78	109	78	516	15	172	62	0	94	32
emad		47	0	94	172	78	187	63	125	93	422	109	110	63	47	93	25
essam888		78	0	79	125	62	250	78	125	63	500	62	125	94	141	94	26
Hosam		94	0	78	63	78	203	109	94	62	610	78	63	78	390	93	20
jadi		47	0	78	156	78	156	94	94	78	453	109	63	63	156	109	25
kingmy		78	0	94	141	78	203	78	109	78	563	15	172	78	0	94	28
nasa		94	0	78	156	47	172	78	109	78	438	94	78	62	0	94	54
p0800238x		93	0	78	141	63	203	63	109	63	843	94	62	78	47	79	28
salah		93	0	78	141	63	203	63	109	63	843	94	62	78	47	79	28
salamro		62	0	79	453	63	218	78	109	78	438	94	78	78	31	79	28
snooh		78	0	94	125	63	187	94	109	109	594	93	63	62	16	78	23
ssomdah		94	0	94	62	62	203	78	110	63	437	109	94	93	16	78	26
username		47	0	78	156	62	219	94	172	62	547	109	94	78	344	63	21
yusefzahrani		63	0	94	172	78	203	63	125	63	484	109	125	62	297	94	31
4400773-00-1C-C0-6D-6E-AA-12864...		63	0	94	140	94	234	79	203	63	968	94	187	62	172	79	25
4400773-00-1C-C0-E6-38-4C		63	0	94	312	78	250	78	125	78	469	94	94	63	46	78	54
		109	0	109	109	79	203	46	125	78	516	125	63	78	31	93	26
		78	0	78	250	62	172	63	125	63	468	94	62	78	16	78	26

Custom	ClientHWAddresses	Key14	Res14	Key15	Res15	Key16	Res16	Key17	Res17	Key18	Res18	Key19	Res19	MacAddress	StorageMedia	BIOSNo
asd		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
bbbbbb		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
ClientHWAddresses		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
ClientLogFile		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
emad		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
essam888		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
Hosam		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
jadi		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
kingmy		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
nasa		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
p0800238x		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
salah		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
salamro		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
snooh		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
ssomdah		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
username		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
yusefzahrani		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
4400773-00-1C-C0-6D-6E-AA-12864...		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
4400773-00-1C-C0-E6-38-4C		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
4400773-Non		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
abdul-00-1C-C0-6D-6E-AA-1286406...		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
abdul-00-1C-C0-E6-38-4C		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW
abdulgader-00-1C-C0-6D-6E-AA-12...		0	0	0	0	0	0	0	0	0	0	0	0	0 00-1C-C0-E6-38-4C	-1670299433	AZCB927006JW

3. User behaviour tables based on hardware for analysis, e.g., “abdulgader-00-1C-C0-6D-6E-AA-128640614-BQJO8280076R”.

This table stores users behaviour in particular hardware by naming the table by users username and HW information. Following figure shows the data fields to contain user’s keystroke behaviour.

	Key16	Key17	Key18	Key19	Key20	Key21	Key22	Key23	MacAddress	StorageMedia	BIOSN
salamro	62	78	32	47	47	63	79	78	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
snooh	62	78	32	47	47	63	79	78	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
ssomdah	47	47	47	31	31	47	63	47	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
username	47	47	47	31	31	47	63	47	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
yusefzahrani	62	63	31	47	47	62	62	78	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
4400773-00-1C-C0-6D-6E-AA-128640614-BQJO8280076R	47	79	32	47	62	47	47	78	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
4400773-00-1C-C0-E6-38-4C	63	79	78	47	47	47	78	62	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
4400773-Non	47	47	47	31	47	78	47	46	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
abdui-00-1C-C0-6D-6E-AA-128640614-BQJO8280076R	63	47	32	46	62	78	47	31	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
abdui-00-1C-C0-E6-38-4C	63	62	47	47	63	47	47	78	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
abdulgader-00-1C-C0-6D-6E-AA-128640614-BQJO8280076R	63	62	63	78	63	46	47	62	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
abdulgader-00-1C-C0-E6-38-4C	47	78	31	62	78	47	63	31	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
abdulgader-Non	46	47	31	62	63	47	63	47	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
abdui-Non	78	78	47	46	62	78	47	46	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
andy-00-1C-C0-6D-6E-AA-128640614-BQJO8280076R	47	78	63	47	62	78	63	46	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
andy-Non	63	62	62	62	47	47	47	63	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
andy-PWD-00-1C-C0-6D-6E-AA-128640614-BQJO8280076R	47	79	47	47	62	63	47	63	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
asd-00-1C-C0-6D-6E-AA-128640614-BQJO8280076R	47	63	47	78	63	63	47	63	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
asd-00-1C-C0-E6-38-4C	47	78	32	63	78	63	47	63	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
asd-Non	62	63	47	78	78	47	47	63	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
Hosam-00-1C-C0-6D-6E-AA-128640614-BQJO8280076R	46	47	47	78	63	47	47	78	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
Hosam-00-1C-C0-E6-38-4C	47	79	32	47	62	47	47	78	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
Hosami-Non	47	63	47	47	47	47	78	94	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
	63	79	78	47	47	47	78	62	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
	47	47	47	31	47	78	47	46	00-1C-C0-6D-6E-AA	-128640614	BQJO8280
	63	47	32	46	62	78	47	31	00-1C-C0-6D-6E-AA	-128640614	BQJO8280

Matlab Files to Run Neural Network Functions

- 1- runClassifier.m
- 2- getCrossValidationMatrix.m
- 3- getTestIndex.m

Neural Network Function to Test User
behaviour

```

function [ ] = runClassifier( input, target)    %extract test data
    testInd = getTestIndex( target );
    testInput = input([testInd],:);
    testTarget = target([testInd],:);    %the remaining will be training data
    trainingInd = setdiff(1:size(target), testInd);
    input = input(trainingInd, :);
    target = target(trainingInd, :);

    hiddenLayersSize = [1];
    cvMatrix = getCrossValidationMatrix(target);

    for cv=1:10
        validationSetInd = cvMatrix(cv,:);
        trainingSetInd = setdiff(1:length(target), cvMatrix(cv,:));

        net = feedforwardnet(hiddenLayersSize);
        net.inputs{1}.processFcns = {'removeconstantrows', 'mapminmax'};
        net.outputs{2}.processFcns = {'removeconstantrows'};
        net.divideFcn = 'divideint';
        net.divideMode = 'sample'; % Divide up every sample
        net.divideParam.trainRatio = 90/100;
        net.divideParam.valRatio = 10/100;
        net.divideParam.testRatio = 0/100;
        net.trainFcn = 'trainrp';
        net.performFcn = 'mse';
        net.layers{1}.transferFcn = 'tansig'; %for hidden layer
        %'logsig',softmax', 'tansig';
        net.layers{2}.transferFcn = 'softmax'; %for output layer

        net.trainParam.showWindow = false;
        %net.trainParam.showCommandLine = true;

        [net,tr] = train(net,input(trainingSetInd,:),target(trainingSetInd,:));

        output = net(input(validationSetInd(find(validationSetInd ~= 0)),:));
        performance = perform(net,target(validationSetInd(find(validationSetInd ~= 0)),:),output);
        %performance=1;

        if(cv == 1)
            cvPerformance.worst=performance;
            cvPerformance.best=performance;
            cvPerformance.average=performance;

            network=net;
            trainingRecord=tr;
        else
            if(performance > cvPerformance.worst)
                cvPerformance.worst = performance;
            end

            if(performance < cvPerformance.best)
                cvPerformance.best=performance;
                network=net;
                trainingRecord=tr;
            end

            if(performance > 3.363105e+000)

                %tr.best_perf
                %tr.best_vperf
            end

            cvPerformance.average = cvPerformance.average + performance;
        end
    end
    cvPerformance.average = cvPerformance.average/10;

```



```

display('Worst performance');
cvPerformance.worst
display('Best performance');
cvPerformance.best
display('Average performance');
cvPerformance.average

testSize= length(testTarget);
testTarget = testTarget';
testInput = testInput';

testResultsFile = 'testResultsFile.txt';
fid = fopen(testResultsFile,'w');
fprintf(fid,'%s \t\t\t\t %s \t\t %s \r\n\r\n','Input', 'Target Output', 'Actual Output');

correct=0;

for k=1:testSize
    [testOutput] = network(testInput(:, k));

    if(testTarget(1 , k) > testTarget(2 , k))

        if(testOutput(1 , 1) > testOutput(2 , 1))
            fprintf(fid,'%s \t| ', sprintf('%d ',testInput(:, k)));
            fprintf(fid,'%s \t| ', sprintf('%d ',testTarget(:, k)));
            fprintf(fid,'%s | ',sprintf('%f ',testOutput));
            fprintf(fid,' %s \r\n\r\n', 'Correct');
            correct=correct+1;
        else

            fprintf(fid,'%s \t| ', sprintf('%d ',testInput(:, k)));
            fprintf(fid,'%s \t| ', sprintf('%d ',testTarget(:, k)));
            fprintf(fid,'%s | ',sprintf('%f ',testOutput));
            fprintf(fid,' %s \r\n\r\n', 'Not Correct');

        end

    elseif(testTarget(1 , k) < testTarget(2 , k))

        if(testOutput(1 , 1) < testOutput(2 , 1))
            fprintf(fid,'%s \t| ', sprintf('%d ',testInput(:, k)));
            fprintf(fid,'%s \t| ', sprintf('%d ',testTarget(:, k)));
            fprintf(fid,'%s | ',sprintf('%f ',testOutput));
            fprintf(fid,' %s \r\n\r\n', 'Correct');
            correct=correct+1;
        else

            fprintf(fid,'%s \t| ', sprintf('%d ',testInput(:, k)));
            fprintf(fid,'%s \t| ', sprintf('%d ',testTarget(:, k)));
            fprintf(fid,'%s | ',sprintf('%f ',testOutput));
            fprintf(fid,' %s \r\n\r\n', 'Not Correct');

        end

    end
end
fclose(fid);
TestResults = (correct/testSize)*100;

display(TestResults);

plotperform(trainingRecord);

end

```

```
function [ cvMatrix ] = getCrossValidationMatrix( target )
%UNTITLED9 Summary of this function goes here
% Detailed explanation goes here

targetValues= unique(target,'rows');
if(length(targetValues) > 2 & length(targetValues) < 2)
    display('This function accepts only matrixes with two target classes.')
    cvMatrix = zeros(1,1);
else

    classOneIndx=find(ismember(target, targetValues(1,:), 'rows'));
    classTwoIndx=find(ismember(target, targetValues(2,:), 'rows'));

    numOfClassOnes = length(classOneIndx);
    numOfClassTwos = length(classTwoIndx);

    cvClassOnes = numOfClassOnes * 0.1;
    cvClassTwos = numOfClassTwos * 0.1;

    cvMatrix = zeros(10,ceil(cvClassOnes+cvClassTwos));

    for cvFold=1:10
        counter=0;
        %f: first class
        for f=cvFold:10:numOfClassOnes
            counter=counter+1;
            cvMatrix(cvFold,counter)=classOneIndx(f);
        end

        %s: second class
        for s=cvFold:10:numOfClassTwos
            counter=counter+1;
            cvMatrix(cvFold,counter)=classTwoIndx(s);
        end
    end
end
end
```

```
function [ testInd ] = getTestIndex( target )
%UNTITLED9 Summary of this function goes here
% Detailed explanation goes here

targetValues= unique(target,'rows');
if(length(targetValues) > 2 & length(targetValues) < 2)
    display('This function accepts only matrixes with two target
classes.')
    testInd = zeros(1,1);
else

    classOneIndx=find(ismember(target, targetValues(1,:), 'rows'));
    classTwoIndx=find(ismember(target, targetValues(2,:), 'rows'));

    numOfClassOnes = length(classOneIndx);
    numOfClassTwos = length(classTwoIndx);

    cvClassOnes = numOfClassOnes * 0.05;
    cvClassTwos = numOfClassTwos * 0.05;

    testInd = zeros(1,ceil(cvClassOnes+cvClassTwos));

    counter=0;
    %f: first class
    for f=1:10:numOfClassOnes
        counter=counter+1;
        testInd(1,counter)=classOneIndx(f);
    end

    %s: second class
    for s=1:10:numOfClassTwos
        counter=counter+1;
        testInd(1,counter)=classTwoIndx(s);
    end
end
end
```

Published paper

Multi-Factor Authentication Using Hardware
Information and User Profiling Techniques

Multi-Factor Authentication Using Hardware Information and User Profiling Techniques

Adeeb Alnajjar and Helge Janicke
 Software Technology Research Laboratory, Faculty of Technology,
 De Montfort University, Leicester, UK
 Email: P08041453@myemail.dmu.ac.uk, heljanic@dmu.ac.uk

March 15, 2012

Abstract

This paper presents a multi-factor authentication approach that extends traditional username-password authentication with hardware and user behaviour profiling techniques. The aim of the approach is to improve the reliability of authentication by computing trust and confidence scores against user profiles. Based on the level of trust, the access control mechanisms may then choose to (un-)lock certain functions or even classify the access as an attack and redirect the user to a honey-pot to gather additional information about the attacker that can be used for a trace-back. The novelty of the approach is that it observes the correlation between users' behaviours and their hardware usage as implicit verification procedures to discriminate the usage of the user-name and password entry.

Keywords: Authentication, Profiling, Multi Factor Authentication, Keystroke Recognition.

1 Introduction

In this paper, we present a simple password mechanism that is augmented with additional profiling techniques to create a form of multi-factor authentication. Using password keys in authentication alone is not reliable due to the users inability to keep them confidential; in addition passwords are often prone to dictionary or rainbow-table attacks as well as the ease with which social engineering techniques can obtain passwords. To address some of these issues our approach integrates with the traditional password authentication by using *Hardware Manufacture Serial Part Numbers (HMSPNs)* to consider the user environment. This approach can be easily integrated in existing password based authentication schemes. Additional factors that are considered in the authentication process are the users' behaviour in providing the user-name and password and the user-profile in using a variety of hardware. Both factors do not require the user to memorise or otherwise keep additional secret information.

Three widely accepted authentication principles base the identification of a user on a) something the user has, b) something the user knows or c) something the user is or does. Multi-factor Authentication Mechanisms employ various techniques, often drawing on several of the above principles to establish a user's identity. For example the credit card payment system (Kumar et al. 2008) with biometric authentication proposes to employ fingerprint verification with a credit card in a multi factor authentication scheme, combining principles a) the card and c) the fingerprint.

However, such an approach would require the installation of additional equipment, thus increasing the cost. The use of additional devices such as fingerprint readers typically also adds to the time taken for authentication which affects the user acceptability for the system. Given that fingerprints can be spoofed with relative ease (Ihmaidi et al. 2006) the overall gain in security is questionable. Indeed most current approaches to multi-factor authentication (Naji et al. 2011, Trevathan et al. 2009) are typically expensive and difficult to deploy and directly affect the usability of the system, as they prolong the authentication process.

The approach presented in this paper avoids the impact of the additional authentication procedures on usability and does not require extra devices to be deployed to end-users. The key novelty of the presented approach is that it integrates profiling information with established user-name/password authentication and can be used to discriminate valid use of password credentials against misuse by an attacker, without complicating the authentication process or incurring large extra costs.

This paper is organized as follows. Section 2 reviews related work of authentication techniques, *HMSPNs* usage in access control and tracking approaches. Section 3 illustrates our authentication approach and the main system

activity. Next, the paper provides a sample analysis scenario using our approach to profile hardware and user activity. After that, the paper provides our system architecture and implements a prototype to show a case study. Finally, the paper evaluates the initial results of our technique and presents the conclusion of the paper including achievements and future work.

2 Related Work

Naji et al. (2011) enhance the security of an access control system using handwritten signature. Their system employs the static and dynamic features of the signature to make a decision about the identity of the signature through a combination of matching statistical models to analyse them. Handwritten signature processing and extracting their features is time consuming and requires dedicated hardware at the user-end.

Card readers are an additional level of hardware security is using one-time password (OTP). The chip on the client “user” card generates the OTP, with the caveat that the account is rendered inaccessible if the card is lost or stolen. This additional challenge-response mechanism over a separate channel removes the need for security questions to confirm transactions and helps preventing fraud. However, this mechanisms requires additional accessories and increases deployment cost (Ravi et al. 2004). With the ubiquity of mobile phones, sending SMS text or voice messages that include one-time password (OTP) is in effect extending the card-reader approach. Here the mobile phone is considered a secure channel, albeit with the increasing connectivity of smart-phones this cannot be considered as independent as the original card-reader. Whilst this approach reduces the cost in deploying readers it adds additional costs on the extra communication channel and requires these channels to be accessible to the user (Zomai & Jsang 2010).

Hardware has been used to facilitate authentication for a long time. The idea is that users register devices (e.g. based on their MAC address) so that the devices are authenticated rather than their users. Examples of devices are storage media drivers such as hard disc drives HDDs. Each storage media has a unique *HMSPN* as an identifier product that can be used in profiling (Patowary 2009). This *HMSPNs* are already actively used for identification, albeit they can be modified at a firm-ware level and thus are susceptible to spoofing, e.g. Microsoft products send product and hardware identifiers during the activation process (Microsoft Corporation 2010). These hardware information provide the opportunity to profile the users’ computing environment.

Based on the hypothesis that different people type in uniquely different typing measure. There are many basic methods (Shanmugapriya & Padmavathi 2009, Attila M 2007, Bergadano et al. 2002, Clarke & Furnell 2007, Yu & Cho 2004, Lee & Cho 2007) used to analyse keystroke typing.

Keystroke dynamics can be used as behavioural biometrics for users. It is an analysing technique for users typing behaviour when keyboard input is monitored (Obaidat & Sadoun 1999). However, if keystroke is not combined with particular keystrokes keys such as the password, it is insufficient to be an objective authentication factor (Teh et al. 2010). The keystroke approach is mostly characterised by the error rates in these following precision cases based on False Acceptance Rates (FAR), False Rejection Rates (FRR) and Equal Error Rates (EER) (Monrose & Rubin 2000).

Statistical (Bergadano et al. 2002) and *neural network* (Gunetti & Picardi 2005) techniques are the main two analysing keystroke approaches. Additionally, there are some combinations of both approaches (Monrose et al. 1999, Clarke & Furnell 2007). Statistical approaches compare a reference set of typing characteristic of specific user with test set of typing characteristic of the same user. Neural Networks use historical data that come from first usage, and then uses this data model to expect the result of new test or classify a new observation (Yu & Cho 2004, Lee & Cho 2007).

Some drawbacks have been exposed by other research (Lv & Wang 2006) that inhibits keystroke from real word applications. One research experiment provided the possibility of using modified keyboards that were based on a pressure sensor to recognize users keystroke (Lv & Wang 2006). This method requires specific keyboards that thus adding again additional cost to the user. To reduce the environment factor that may affect user behaviour in keystroke, Maxion & Killourhy (2010) explored a number pad input using a single finger. They tried to discriminate users typing style, FAR and FRR scope suggests a low level of surety that authentication using keystroke biometrics might be possible in this particular environment.

3 Our Approach

Our approach combines hardware identification with key-stroke biometrics, yielding a multi-factor authentication approach in which user biometrics can be correlated with the hard-ware that is used during the login process. The analysis of user-typing patterns on particular hardware by monitoring the keyboard inputs can visualize the significant pattern difference between the users. This correlation is reducing the FAR and FRR rates and allows the approach to be deployed throughout heterogeneous systems which are comprised of various hardware interfaces.

The key contribution of our approach is to improve the login-procedure by determining the level of trust of the user without additional cost or making the deployment of the solution overly complex. Thus, the key objective of our approach is developing a novel technique for the analysis of *HMSPNs* properties and patterns that are captured in the computational model. After that, an approach is developed for modelling the dynamic behaviour of the user. Then, user profiles based on analyzing and modelling users' behaviour to develop a new technique for the analysis of Internet services based on these profiles is formulated.

Hardware parts have a particular history in *HMSPNs* usage. Some computer hardware parts have not changed and have been used by the manufacturer for a long time. Therefore, every computer device has a history tracking over the time of its *Life cycle*. Thus, each computer hardware part has a particular track of usage from manufacture phase to destruction. First, if a user has been dealing with a device for every log in procedure for access control applications for a long time, this user will be more familiar with this hardware and has a particular behaviour when using it. Therefore, the user has a particular pattern scope that will be used with this hardware. Consequently, if the number of users of a particular hardware is increased, our authentication approach has to recognize the way these users behave when using this hardware, even if they use the same user-name and password. Of course, the sharing of accounts is bad practice, but still commonly encountered in both domestic and corporate environments over which the service provider has little influence. For example in Figure 1 Bob and Colin used John's hardware, however they have different behaviours in dealing with same hardware. Consequently, our approach has to find the different attribution of users' behaviour when they use the same hardware and the same user-name and password. Ultimately, our authentication technique maps user *environment* hardware in order to demonstrate the user behaviour in previous pattern usage in particular hardware.

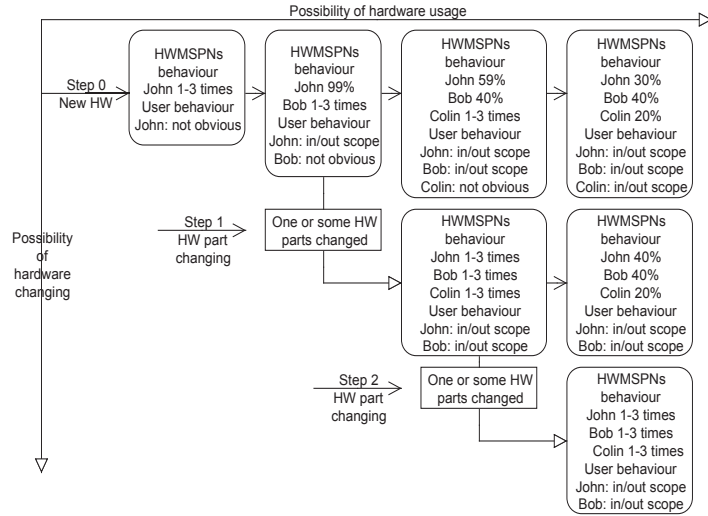


Figure 1: Hardware and users behaviour *Life Cycle*

The hardware life cycle in Figure 1 explains conceptually the hardware usage that supports the learning of user behaviour depending on a particular hardware configuration. However, the hardware parts may change over the time, resulting at configurations that are distinct to previous login attempts by their users. One example is the use of a tablet. E.g. the login may be typed on the touch screen or (after attaching the tablet to a docking station) through a physical keyboard. These changes in hardware configurations affect user profiling. "Step 1" and "Step 2" in Figure 1 reflect changing the hardware parts which change user environment. Therefore, the system has to recognise hardware changing and compare user's hardware at every login.

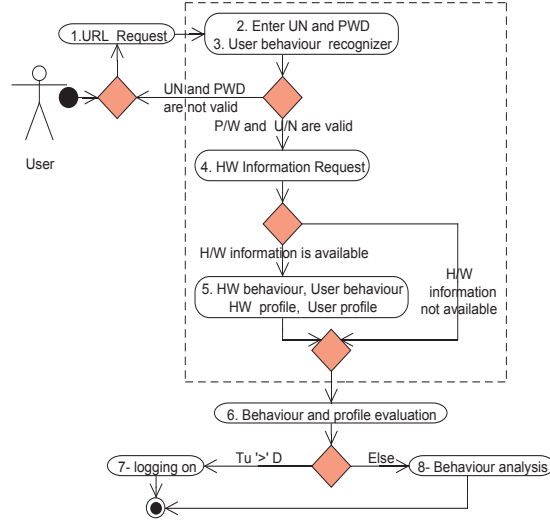


Figure 2: System Overview

3.1 System Overview

Our authentication system uses two components in the login procedure. Whilst the user u is typing his/her username and password, our first component captures the *current user behaviour* (b_u) by calculating the keystroke (both key-press and release) speed when username and password are typed. The second component collects the HMSPNs, which make up the user's current hardware configuration (c_u). As the user or other security software installed on the client machine can prevent the gathering of hardware information, we consider this to be optional information. However, if this information is not provided it has detrimental effects on the accuracy of our mechanism, as the hardware profiling information is coupled with the selection of the user-profile for keystroke recognition. If the user provides access to the hardware profile, the system begins to analyse and compare the current hardware configuration (c_u) with the established profile of that user (\bar{c}_u) to determine their similarity. If the user has used the current hardware before, the system computes the similarity between the current keystroke behaviour of the user (b_u) and the behaviour that has been recorded against this hardware configuration previously (\bar{b}_{u,c_u}). If the current hardware configuration is not known, the component will try to match b_u against all known keystroke behaviours for that user $\bar{b}_{u,*}$ indiscriminate of the hardware configuration, which obviously reduces the effectiveness of this mechanism.

Given that the username and password checks are successfully passed, the system will compute out of the similarity between the hardware configuration and their profiles, and the associated keystroke behaviour similarity to their profiles two levels of trust. If only keystroke information is available, only one level of trust is being used in the following.

Given that usernames and passwords are not very secure, the hardware similarity test reflects the idea that hardware that has been previously used by the same user increases the likelihood of the user being genuine, as this rules out attacks in which passwords have been observed by shoulder surfing or rainbow table attacks. In addition, uncharacteristic use of hardware, e.g. the use of a company PC that has regularly been used during office-hours for 6 month and from which now an access is taking place at 2am in the night, is flagged up by a low trust-level in the hardware.

Similarly the key-stroke behaviour is evaluated, linked against the used hardware configuration (c_u) if available. The system will authenticate normally if the username and password are correct and a threshold in both levels of trust is passed. If the user-name and password do not match, the authentication is considered failed. If the username and password are correct, and only a low level of trust is established based on the hardware or keystroke



Figure 3: Hardware history

behaviour the system can be configured to adapt to the level of trust. E.g. the authentication can be failed; the user can be authenticated with reduced privileges such as only being able to view his account details; the system can increase the threshold for an intrusion detection system that identifies fraudulent activity based on the transactions that are undertaken or even redirect the user to a honey pot trapping system to explore if the user is a hacker using a spoofed user-name and password. In an e-banking context, this could e.g. mean to delay the transactions and attempt to contact the user via a different channel such as email or phone. Figure 2 shows the basic steps in the system operation.

3.2 System Activities

Our technique depends on the matching of the current hardware configuration c_u against the users previous hardware behaviour \bar{c}_u and the associated user behaviour b_u against the previous user behaviour \bar{b}_u as part of the login procedure.

On the client side, the login prompt performs three data-collection functions. Firstly the username and password is collected in the traditional way. Secondly the keystroke behaviour of the user is gathered during the typing of the username and password. Functions like autocompletion and provision for copy & paste are turn off, as they would effectively disable the recognition of the keystroke behaviour. Thirdly the login prompt will attempt to collect the hardware configuration from the user's operating system. This may require the user to whitelist the login software or the server address from which the login prompt is loaded.

On the server side the authentication module will first check the username and password hash against the stored credentials. If this is successful, the additional two components *hardware recogniser* and *keystroke recogniser* are invoked to further qualify the login request, thus providing additional scrutiny.

3.2.1 Hardware Recogniser

The hardware trust is computed by the *hardware recogniser*, which matches the current configuration against previously used hardware configurations for the same user based on the parts' serial numbers. This process takes into account the previous usage patterns of the user over time and also considers other aspects such as concurrent usage of the same hardware configuration or hardware parts in different login processes, which e.g. could indicate a spoofing attack. Essentially there are three key results that can be generated by this component:

1. Trust level based on usage of hardware configuration
2. Known configuration for use in behaviour recognition (or matching configuration)
3. Cross login analysis for attack detection.

The trust level is computed against the history of previous login-attempts and their associated hardware configurations \bar{c}_u which is essentially drawn from the sequence of previous successful login attempts by this user.

Figure 3 shows a simplified example. Every node on the timeline represents a successful login by the user in question. The used hardware configuration is depicted by the shape of the node, eg. the empty circle could be the user's office machine, the square a mobile device, the filled circle a user's home computer. The first step is that the hardware is checked whether it has been used before, ie. it is known to the system, which is important for the keystroke recogniser in subsequent checks. This establishes a baseline trust for the access in case the hardware is known.

Secondly the access is viewed in the context of the other accesses (left neighbours), the time and the day of the access. We chose metrics based on time of day and day in week as these constitute the majority of repetitions we have encountered. We currently do not support more complex analysis of these events in our prototype, but envision the use of neural networks or support vector machines to establish a behaviour baseline against which the check can be performed. Based on the "fit" of the hardware configuration used in the login the trust level is adjusted.

Thirdly, the hardware recogniser maintains a cache of recent and current login activities over the entire user-base. If there is a current login from the same hardware configuration or configurations that share particular hardware components there is a chance that one of the logins is fraudulent and based on spoofed hardware information. It is known that some hardware manufacturers fail to provide unique serial numbers for their components. For the

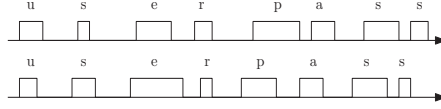


Figure 4: Keystroke patterns

Table 1: Keystroke profile \bar{b}_{u,c_u} against hardware configuration c_u

#	1	2	3	4	5	6	7	8	9	10	11
u↓	10	8	9	11	15	8	10	8	11	6	12
u↑	10	8	9	11	15	8	10	8	11	6	12
s↓	6	5	7	8	9	6	7	6	8	5	8
s↑	15	10	10	12	20	12	11	10	12	12	10

known cases we have a black-list of manufacturer ids which are excluded from this analysis step. A collision here reduces the trust level established by the *hardware recogniser*.

3.2.2 Keystroke Recogniser

The keystroke recogniser takes the current keystroke pattern entered by the user (b_u) and matches it against the previous recorded keystroke behaviour of that user using that hardware (\bar{b}_{u,c_u}).

The keystroke pattern is characterised by the press and release times of the keys that are used in entering the username and password and is gathered on the client side. Figure 4 gives an example of such a pattern.

Our current prototype only considers the press and release times as a proof of concept and does not use other correlations between subsequent keypress events that may be further improving the accuracy. As the contribution of this paper is not a novel keystroke recognition scheme, but the integration of multiple approaches this mechanism can be replaced with more sophisticated techniques such as specific keystroke recognition (Shanmugapriya & Padmavathi 2009).

We currently build a trust-metrics based on whether the current keystroke pattern fits the users profile information, where the profile is created based on the previous user inputs. For example with respect to Figure 4 the first keyevent is the time the letter “u” is pressed. Previous logins e.g. recorded the times in Table 1 which forms the user profile, depicted in Figure 5. Currently the system looks at the variance of the data and the percentile into which the current keystroke pattern falls with respect to each of the keypress and release events and computes an accumulated trust level over all events contained in the keystroke pattern. In comparison to e.g. specific keystroke recognition (Obaidat & Sadoun 1999) this is a very simple approach which we plan to refine in the future.

3.3 System analysis

Our technique depends on the matching of the current hardware configuration c_u against the users previous hardware behaviour \bar{c}_u and the associated user behaviour b_u against the previous user behaviour \bar{b}_u as part of the login procedure. On the client side, the login prompt performs three data-collection functions. Firstly the user-name and

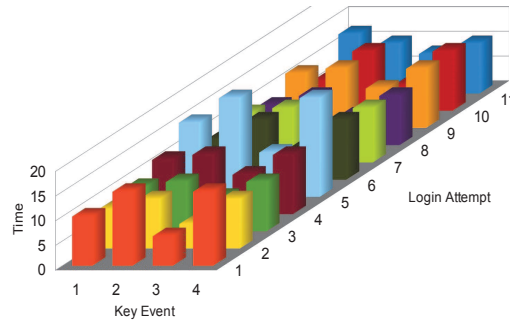


Figure 5: Keystroke Profile

password is collected in the traditional way. Secondly the keystroke behaviour of the user is gathered during the typing of the user-name and password. Functions like autocompletion and provision for copy & paste are turn off, as they would effectively disable the recognition of the keystroke behaviour. Thirdly the login prompt will attempt to collect the hardware configuration from the user's operating system. This may require the user to whitelist the login software or the server address from which the login prompt is loaded.

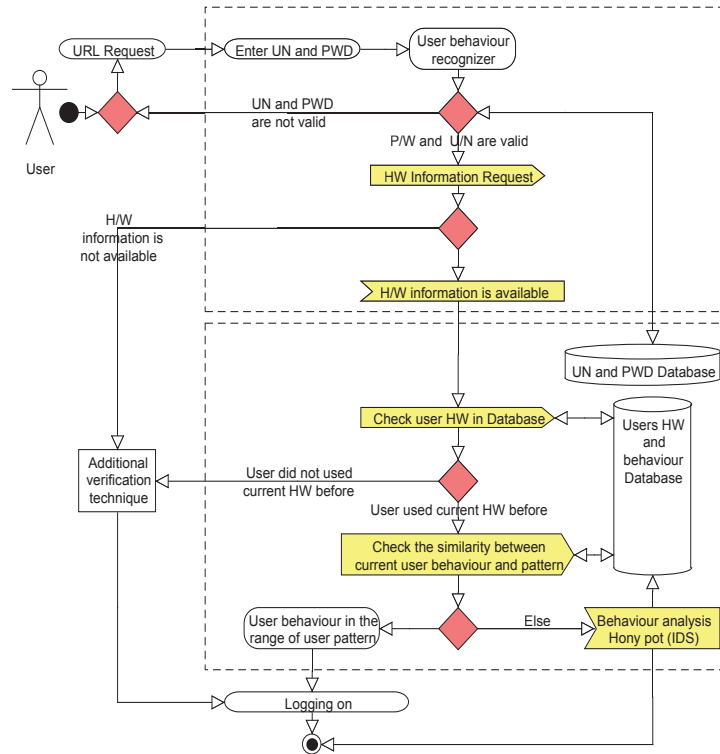


Figure 6: Flow chart

4 Case-Study

We developed a simple Java application to apply our approach in the login process as an implicit login procedure. Every log in, our system captures user behaviour using a keystroke function to calculate users typing speed and response time among the keys of the user-name and password. The user-name and password contains characters and number. Then, when the user typed his/her valid user-name and password the system collects three parts of *HMSPNs*. These parts are the BIOS device number, MAC address number and the hard disk drive number. After that, the system recognizes if the user used current hardware before and if and to what extend the hardware was used by other users. Figure 7 shows the percentage of hardware usage and user pattern stamp by determining how the current user behaviour is related to previous usage patterns.

In this case study, system improves the ability of observe the levels of trust to reflect the different b_u when the user uses different hardware. In this scenario, the user performed 200 succeeded log in using *username* and *password* as key to log. However, the user used two devices representing two different hardware environments.

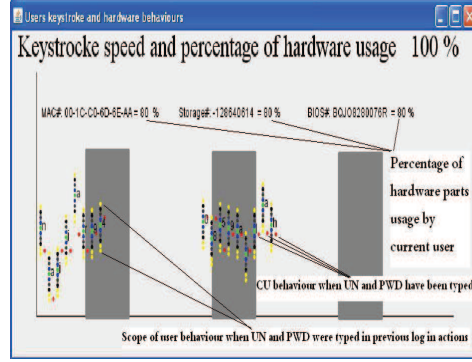


Figure 7: $HMSPNs$ usage c_u and profile \bar{b}_{u,c_u} against keystroke pattern b_u .

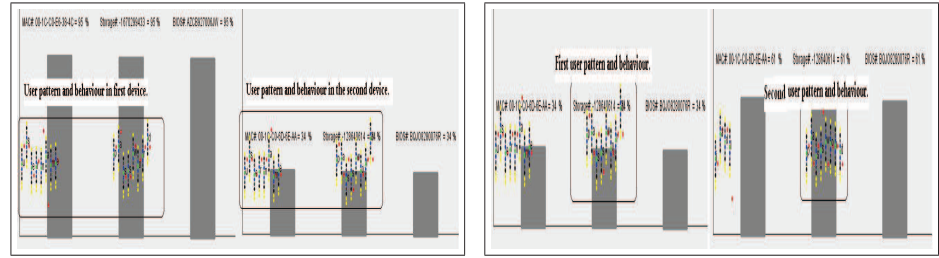


Figure 8: LEFT: $HMSPNs$ usage c_u and profile \bar{b}_{u,c_u} against one user uses two hardware. RIGHT: $HMSPNs$ usage c_u and profile \bar{b}_{u,c_u} against two users use same login information (password) in one hardware

In the second scenario, two users used same hardware and a shared password for 100 successful log in attempts. The system recognised the effect of the hardware in user keystroke behaviour. In addition, the system compared between the users depending on their familiarity with the hardware. This recognition comes from the hardware trust.

4.1 Trust

For all login attempts that provided the correct username and hardware we computed the hardware trust based on the hardware configuration that was used in the login attempt against the previously encountered hardware. We computed the trust-level based on precedences, ie. if the hardware was encountered previously we assigned a baseline trust of 40% for previously encountered hardware. Based on whether there was a precedent of that hardware being used on that day in the week, within that hour of the day and after the use of the previously used hardware configuration, we added additional 20% as these occurrences increased our confidence. If the hardware configuration (or part thereof) was used concurrently in another login process we subtracted 60% from the trust-level.

For all three hardware configurations that were used in the case-study, we recorded 100 keystroke patterns to build up the profile. The trust was computed by calculating the deviation from the mean for each key-event (key-press and release) of the profile against the standard deviation as a percentage value. The overall keystroke trust was then computed as the mean of the individual percentage values.

We overall set relatively low thresholds for both trust levels, and proceeded with the authentication when both trust levels exceeded 70%. If only one of the trust-levels exceeded the threshold, an additional verification question was asked from the user. If this was answered correctly the authentication was considered successful. If both trust levels fell below the threshold, the login attempt was considered unsuccessful and the user was returned to the login prompt. We considered a maximum of three unsuccessful login attempts before the account was blocked.

The recorded profile information was only updated after a successful login attempt. This means that even if behaviour or hardware usage changed over time the system was able to adapt, in most cases via the provision of an additional security question. We did not yet integrate actual honeypots into our system or linked it to the access

control system.

5 Conclusion & Future Work

The availability of hardware information can enhance authentication mechanisms. The work presented in this paper shows that by capturing a wide range of statistics it is possible to perform an analysis of hardware and user behaviour. In this paper we considered keystroke as a biometrics. By combining password based authentication with hardware profiling and keystroke recognition we provided a multi-factor authentication scheme that does not require additional devices to be deployed and adds little cost to the deployment of the authentication system.

The paper reviewed related work on authentication approaches and their limitation as a motivation for this approach. We then presented our approach and showed how the additional data can be collected on the client side and what data needs to be collected. We then described in detail the server-side and the functioning of the hardware-recogniser and the keystroke recogniser and how their interaction improves the accuracy of keystroke recognition as a more specific profile can be maintained depending on the hardware that is used.

We implemented our prototype system using basic profiling techniques for the analysis and presented a trust-model that takes into account the hardware usage and the user behaviour when entering his/her username and password. The prototype is of course a proof of concept that shows that the techniques can be combined and that their combination yields a positive influence on the accuracy of the detection. In the future we will refine the individual techniques and adopt e.g. keystroke recognition approaches that have been presented in (Obaidat & Sadoun 1999). We provided a java-based prototype implementation of our authentication system and presented a small case-study as a proof of concept for our work.

In the future we will refine the profiling techniques used in our authentication framework and are looking at implementing techniques based on neural networks or support vector machines. We also investigate the use of the profile information in attack attribution, as the hardware profiles can provide indication about (fraudulent) users. In addition, we will look at geo-spatial information and its integration in the hardware recogniser. The idea is that successive logins from different geographical areas are not plausible and can indicate fraudulent activity. In this line of investigation we will also actively deploy honeypots to further identify behavioural traits of the user. This information can then be used twofolds: a) to provide additional attribution information about the attacker; b) to retrospectively authorise the actions performed if the user is deemed to be genuine.

References

- Attila M, Zoltn B, L. C. (2007), 'Strengthening passwords by keystroke dynamics', *IEEE* .
www.knt.vein.hu
- Bergadano, F., Gunetti, D. & Picardi, C. (2002), 'User authentication through keystroke dynamics.', *ACM Trans. Inf. Syst. Secur.* **5**(4), 367–397.
<http://dblp.uni-trier.de/db/journals/tissec/tissec5.html/BergadanoGP02>
- Clarke, N. L. & Furnell, S. (2007), 'Authenticating mobile phone users using keystroke analysis.', *Int. J. Inf. Sec.* **6**(1), 1–14.
<http://dblp.uni-trier.de/db/journals/ijisec/ijisec6.html/ClarkeF07>
- Gunetti, D. & Picardi, C. (2005), 'Keystroke analysis of free text.', *ACM Trans. Inf. Syst. Secur.* **8**(3), 312–347.
<http://dblp.uni-trier.de/db/journals/tissec/tissec8.html/GunettiP05>
- Ihmaidi, H.-D., Al-Jaber, A. & Hudaib, A. (2006), Securing online shopping using biometric personal authentication and steganography, in 'Information and Communication Technologies, 2006. ICTTA '06. 2nd', Vol. 1, pp. 233–238.
- Kumar, D., Ryu, Y. & Kwon, D. (2008), A survey on biometric fingerprints: The cardless payment system, in 'Biometrics and Security Technologies, 2008. ISBAST 2008. International Symposium on', pp. 1–6.
- Lee, H. & Cho, S. (2007), 'Retraining a keystroke dynamics-based authenticator with impostor patterns.', *Computers and Security* **26**(4), 300–310.
<http://dblp.uni-trier.de/db/journals/compsec/compsec26.html/LeeC07>
- Lv, H.-R. & Wang, W.-Y. (2006), 'Biologic verification based on pressure sensor keyboards and classifier fusion techniques', *Consumer Electronics, IEEE Transactions on* **52**(3), 1057–1063.

- Maxion, R. & Killourhy, K. (2010), Keystroke biometrics with number-pad input, in 'Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on', pp. 201–210.
- Microsoft Corporation (2010), 'Microsoft office activation/registration privacy statement'.
<http://o\ce.microsoft.com/en-us/help/HP010069531033.aspx>
- Monrose, F., Reiter, M. K. & Wetzel, S. (1999), Password hardening based on keystroke dynamics., in J. Motiwalla & G. Tsudik, eds, 'ACM Conference on Computer and Communications Security', ACM, pp. 73–82.
<http://dblp.uni-trier.de/db/conf/ccs/ccs1999.html/MonroseRW99>
- Monrose, F. & Rubin, A. D. (2000), 'Keystroke dynamics as a biometric for authentication', *Future Gener. Comput. Syst.* **16**, 351–359.
<http://dl.acm.org/citation.cfm?id=338350.338359>
- Naji, A. W., Housain, A. S., Zaidan, B. B., Zaidan, A. A. & Hameed, S. A. (2011), 'Security improvement of credit card online purchasing system', *Scientific Research and Essays* **6**(16), 3357–3370.
- Obaidat, M. S. & Sadoun, B. (1999), Keystroke dynamics based authentication, in 'In "Biometrics. Personal Identification in Networked Society". A.Jain, R.Bolle, S.Pankanti (Eds', Kluwer Academic Publishers, pp. 213–229.
- Patowary, K. (2009), 'How to interpret hard disk model numbers'.
<http://www.instantfundas.com/2009/02/how-to-interpret-hard-disk-model.html>
- Ravi, S., Kocher, P. C., Lee, R. B., McGraw, G. & Raghunathan, A. (2004), Security as a new dimension in embedded system design., in S. Malik, L. Fix & A. B. Kahng, eds, 'DAC', ACM, pp. 753–760.
<http://dblp.uni-trier.de/db/conf/dac/dac2004.html/RaviKLMR04>
- Shanmugapriya, D. & Padmavathi, G. (2009), 'A survey of biometric keystroke dynamics: Approaches, security and challenges', *CoRR abs/0910.0817*.
<http://dblp.uni-trier.de/db/journals/corr/corr0910.htmlabs-0910-0817>
- Teh, P. S., Teoh, A. B. J., Tee, C. & Ong, T. S. (2010), 'Keystroke dynamics in password authentication enhancement', *Expert Syst. Appl.* **37**, 8618–8627.
<http://dx.doi.org/10.1016/j.eswa.2010.06.097>
- Trevathan, J., McCabe, A. & Read, W. (2009), Online payments using handwritten signature verification., in S. Latifi, ed., 'ITNG', IEEE Computer Society, pp. 901–907.
<http://dblp.uni-trier.de/db/conf/itng/itng2009.html/TrevathanMR09>
- Yu, E. & Cho, S. (2004), 'Keystroke dynamics identity verification - its problems and practical solutions.', *Computers and Security* **23**(5), 428–440.
<http://dblp.uni-trier.de/db/journals/compsec/compsec23.html/YuC04>
- Zomai, M. A. & Jsang, A. (2010), The mobile phone as a multi otp device using trusted computing., in Y. Xiang, P. Samarati, J. Hu, W. Zhou & A.-R. Sadeghi, eds, 'NSS', IEEE Computer Society, pp. 75–82.
<http://dblp.uni-trier.de/db/conf/nss/nss2010.html/ZomaiJ10>