Towards Usable End-user Authentication

Mohammad Tanviruzzaman
Marquette University

TOWARDS USABLE END-USER AUTHENTICATION

by

Mohammad Tanviruzzaman

A Dissertation submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy

Milwaukee, Wisconsin
August 2014

ABSTRACT

TOWARDS USABLE END-USER AUTHENTICATION

Mohammad Tanviruzzaman

Marquette University, 2014

Authentication is the process of validating the identity of an entity, e.g., a person, a machine, etc.; the entity usually provides a proof of identity in order to be authenticated. When the entity—to be authenticated—is a human, the authentication process is called end-user authentication. Making an end-user authentication usable entails making it easy for a human to obtain, manage, and input the proof of identity in a secure manner. In machine-to-machine authentication, both ends have comparable memory and computational power to securely carry out the authentication process using cryptographic primitives and protocols. On the contrary, as a human has limited memory and computational power, in end-user authentication, cryptography is of little use. Although password based end-user authentication has many well-known security and usability problems, it is the de facto standard. Almost half a century of research effort has produced a multitude of end-user authentication methods more sophisticated than passwords; yet, none has come close to replacing passwords.

In this dissertation, taking advantage of the built-in sensing capability of smartphones, we propose an end-user authentication framework for smartphones, called ePet, which does not require any active participation from the user most of the times; thus the proposed framework is highly usable. Using data collected from subjects, we validate a part of the authentication framework for the Android platform. For web authentication, in this dissertation, we propose a novel password creation interface, which helps a user remember a newly created password with more confidence—by allowing her to perform various memory tasks built upon her new password. Declarative and motor memory help the user remember and efficiently input a password. From a within-subjects study we show that declarative memory is sufficient for passwords; motor memory mostly facilitate the input process and thus the memory tasks have been designed to help cement the declarative memory for a newly created password. This dissertation concludes with an evaluation of the increased usability of the proposed interface through a between-subjects study.

ACKNOWLEDGEMENTS

Mohammad Tanviruzzaman

I would like to thank my committee members for their valuable suggestions and guidance. I am deeply indebted to my supervisor, Dr. Sheikh Iqbal Ahamed, without whose selfless time and care, I would not have survived this long and arduous journey, called Ph.D. I would like to pay my gratitude to Dr. Stephen Merrill; beside learning a lot in his courses, I found a way to regain belief in myself; and to me that was an invaluable gift. My parents' and my brothers' love and their belief in me have always been a source of courage for me. My wife Rizwana Rizia, whom I met in my Ph.D. life, has been my constant companion. And, my two-year old son, Angshu—to whom and to whose (imagined) future-self alike, I often hold delightful conversations—was my greatest inspiration towards the completion.

Contents

List of Tables **vii**

List of Figures **viii**

1 Introduction **1**

1.1 Motivation for Usable End-User Authentication for Smartphones . . 1

1.2 Motivation for Usable End-User Authentication for the Web 3

1.3 Major Contributions 4

1.3.1 Design, Development, and Evaluation of ePet 4

1.3.2 Effect of the Declarative and the Motor Memory on Password-Input 4

1.3.3 Design, Development, and Evaluation of Password Creation Web-Interface 5

1.4 Dissertation Organization 5

1.5 Publications 6

1.5.1 Publications on ePet 6

1.5.2 Works That will be Benefited by ePet 6

1.5.3 Publications on Password Creation Interface 7

1.5.4 Works That will be Benefited by Password Creation Interface 7

2 Related Work **9**

2.1 Something the user knows 10

2.1.1 Conscious Knowledge 10

2.1.2 Subconscious Knowledge 18

2.2 Something the User Has 18

2.3 Something the User Is 19

2.4 Something the User Does 19

2.5 Someone Who Knows the User 20

2.6 Unorthodox Input Methods 21

2.7 Fall-back Authentication 28
2.8 Password Manager 29
2.9 Graded Security 31
2.10 Implicit Authentication 31
2.11 Placing the Work in Context 31

3 Usable End-user Authentication for Smartphones 33

3.1 Proposed Authentication Framework 34
3.2 A Subset of the Framework 40
3.2.1 Recognizing Walking Pattern 40
3.2.2 Building Location Trace 42
3.2.3 Authentication Using Gait Pattern and Location Cues 43
3.3 Experimental Procedure 43
3.3.1 Pattern of Walking 44
3.3.2 Trace of Visited Places 45
3.3.3 Battery Charge 46
3.3.4 Ideal Authentication System for Smartphones 46
3.4 Potential Issues 47
3.4.1 Battery Charge 48
3.4.2 On-body Placement 48
3.4.3 Psychology 49
3.4.4 Pace 49
3.5 Discussion 49

4 A Closer Look into the Password Input Process 51

4.1 Human Memory 51
4.2 Conceptual Model for Password Input 53
4.2.1 Look-up Coordinates 53
4.2.2 Erroneous Finger Movements 55
4.3 Study Procedure 55
4.3.1 Old password, Familiar Keyboard-Layout 56
4.3.2 Old Password, Familiar Keyboard-Layout, with Count-Down 56

4.3.3 Old Password, Unfamiliar Keyboard-Layout 56
4.3.4 Old Password, Unfamiliar Keyboard-Layout, with Count-Down 57
4.3.5 Four Conditions with a New Password 57
4.3.6 Time Spent in Password Input 59
4.4 Results 61
4.4.1 Users' Opinions 63
4.5 Related Works 65
4.6 Discussion 66

5 Usable End-User Authentication for the Web 67

5.1 Password Creation Interface 67
5.2 Motivation Behind the Proposed Password Creation Interface 67
5.3 Description of the Proposed Password Creation Interface 69
5.3.1 Distractor Task 69
5.3.2 Memory Task-1 71
5.3.3 Memory Task-2 73
5.3.4 Memory Task-3 75
5.3.5 Memory Task-4 76
5.4 Security Issues 78
5.5 Experimental Procedure 78
5.6 Tools Used 79
5.7 Information Collected 79
5.8 Results 79
5.8.1 Number of Passwords 80
5.8.2 Password Strength 81
5.8.3 Success Rate 88
5.8.4 Time of the Earliest Significant Failure 90
5.8.5 Confusion in Successful Attempts 91
5.8.6 Error in Failed Login Attempts 94
5.9 Discussion 98

6 Conclusions and Future Work **100**

6.1 Research Achievements 100

6.1.1 Usable End-User Authentication for Smartphones 100

6.1.2 A Closer Look into the Password Input Process 101

6.1.3 Usable End-User Authentication for the Web 101

6.2 Summary of Contributions 102

6.3 Future Research Directions 102

6.3.1 Usable End-User Authentication for Smartphones 103

6.3.2 A Closer Look into the Password Input Process 104

6.3.3 Usable End-User Authentication for the Web 104

Bibliography **107**

List of Tables

3.1 Smartphone-based gait recognition 42
3.2 EER (%) at chosen thresholds 45
3.3 Ideal authentication for smartphones 48

4.1 The eight conditions, in which a user had to login 58
4.2 The order in which the eight conditions were performed 59
4.3 Tasks in Eight Conditions 59
4.4 Comparing time spent in a pair of conditions 60
4.5 Reasons for the relations between median times in a pair of conditions 61
4.6 Information about old password and typing speed 61
4.7 Comparison of pairs of conditions 63
4.8 Challenges with an unfamiliar layout 64

5.1 Overall information 80
5.2 No. of passwords per user 80
5.3 Overall valid information 80
5.4 Information about composite symbols 85
5.5 Password strength 88
5.6 Strengths of first and second passwords 88
5.7 Success rate per user 89
5.8 Time (min.) of the earliest significant failure 91
5.9 Time-sepration between distinct successful login attempts (in hrs) . 92
5.10 $E_{\alpha_u^-}$ according to equation 5.3 95
5.11 $\mathcal{E}_{\alpha_u^-}$ according to equation 5.4 98

List of Figures

2-1 Authentication methods 30

3-1 Periodicity in gait data 40
3-2 DTW Matching of two sine waves 41
3-3 Correlation: DTW score vs. other factors 44
3-4 Timing of location and gait computations 46
3-5 Battery charge over time 47

4-1 Organization of human memory 52
4-2 Information flow in human memory 52
4-3 Password input process 53
4-4 Example table look-up 54
4-5 Login-time in different conditions 62

5-1 Creating a Gmail account 68
5-2 Correctly perform the addition or subtraction 69
5-3 In the middle of a distractor task 70
5-4 Distribution of the number of difficult arithmetic operations 72
5-5 Select correct username and password 72
5-6 Select correct characters from the drop-down lists 73
5-7 10-sided die [159] 73
5-8 User's effort for the second memory task 74
5-9 Select correct characters from the drop-down lists 76
5-10 Distribution of number of inversions for various password lengths . . 77
5-11 Select correct characters from the drop-down lists 77
5-12 Database schema 79
5-13 Cumulative distribution functions 86
5-14 Composite mass 86
5-15 Entropies of composites and bigrams 88

5-16 Password strengths 89
5-17 Success rates of two groups over time 90
5-18 Time-separation between successful attempts 92
5-19 Violin plot of confusion 93
5-20 $E_{\alpha_u^-}$ according to equation 5.3 95
5-21 Time-separation between a failure and the previous success 96
5-22 Including time in error 97

Chapter 1

Introduction

Computers, connected to the Internet, are essential for the proper functioning of our everyday life; as a result, security of computer systems is vital. The global market for security technology and services is forecast to grow to more than $86 billion by 2016 [160]. Generally, computer security has at least three aspects to it: physical security, technological security, and security policies; all three aspects must be taken care of, to provide overall security of a computer system [1].

The focus of this dissertation is on end-user authentication, which belongs to the technological aspect of security. End-user authentication takes place between a human and a computer; thus, the asymmetry of memory and computational power between the two sides come into play. We attempt at making the authentication process more usable, i.e., easier for the human, without sacrificing security too much in two specific scenarios: when a user is trying to unlock her smartphone and when she is authenticating herself to a website. For the two scenarios, in the next two sections, we give the motivation behind our work through two realistic, but otherwise fictional setups.

1.1 Motivation for Usable End-User Authentication for Smartphones

Mary never found a need for setting a locking mechanism on her Android smartphone. She left her phone unattended several times before, but nothing harmful happened. For example, once she went home, forgetfully leaving her phone in a coffee shop; but, when she returned to the shop later, the shop owner—a long time acquaintance of her—gave the phone back to her and everything on the phone was in proper order. However, just a while ago, some very personal photos on her smartphone—which she never intended to share—have been published to a social network that she regularly accesses from her phone. In retrospect, she now realizes, when in a hurry, she forgetfully left the phone for a short while in a pub, which she visited with her friends, someone must have picked it up and uploaded those photos

to the always-connected social network from her phone.

Mary has become more cautious now, and she has set a password on her smartphone. Initially, the password she chose was 12 characters long, with captial letters, digits, and special symbols in it; but, not only it took time and effort to memorize the password, it was also very difficult to correctly input the password on the phone's keypad; now, she has shortened the password to 6 characters, yet it has proven very tiresome to input it again and again throughout the day. Mary does not want to leave her phone unprotected, but the difficulty with the password is driving her crazy.

On finding how difficult it is to use a password on a smartphone, Mary has set a lock-pattern on her smartphone, which looks like a "⊔"; she believes, it is easy to use and it makes the phone secure. But the other day, she found out that her friend, Tom, somehow knew about Mary's date on the next Friday. Mary never told anyone about the date. When she asked Tom how he knew about her future date, Tom told that it was on her phone's calendar and it was quite easy to unlock the phone; for, even a casual glance from a distance at her lock-pattern was enough to reproduce it.

On learning how easy it is for anyone to get around the lock- pattern, Mary has now switched to the face and voice based locking mechanism on her smartphone, which she thinks is more secure and easy-to-use than a password or a pattern. Even though the voice based locking mechanism does not seem to work at a noisy place, the face-based locki seems to work fine for her. Right now, she is attending a presentation; the light is dim; she is also waiting for an urgent email; unfortunately she cannot unlock her phone to check emails; because, the phone cannot recognize her face in such low light. She cannot go out of the room now and she is stuck.

In this dissertation, we propose and evaluate an authentication framework for smartphones, called ePet, that is built around the metaphor of how a pet—on its own—recognizes its owner; most of the times ePet does not require the user to participate actively in the authentication process and yet provides sufficient security. Thus, with the ePet on her smartphone, Mary may continue using her phone to check emails while attending the presentation without even realizing that her smart-

phone has recognized Mary on its own; if someone else other than Mary tried to use the phone, it would have locked itself. Because the authentication process is not visible anymore, Mary may not feel as secure as she would with a traditional locking mechanism; but, we believe, the pet metaphor helps in communicating the idea quickly to a user.

1.2 Motivation for Usable End-User Authentication for the Web

Jill is an employee of a company called Mahut, and she is trying to place a large order of some goods on behalf of her company from the website of an e-commerce company named Haati. In order to prevent a malicious competitor from stealing its business by creating a similar-looking website, Haati stipulates using Secure Sockets Layer or SSL protocol so that Haati's website authenticates itself to Jill's web browser [161]. Jill in turn authenticates herself to Haati's website by entering her username and password. Matt is another employee of Mahut, who does auditing and accounting. Matt cannot place an order on behalf of his company, but he can see how many orders have been placed. So, Mahut's website authorizes Jill to place an order after she logs in, whereas it does not allow Matt to do so even if he can login.

Haati does not want its competitors to be able to see the details of the orders that its customers make; in other words, it wants to keep its business confidential; using SSL ensures that all communications between, say Jill and Haati, are encrypted [161]. Haati's competitor can alter the amount of Jill's order, so Jill might gets frustrated and switch to one of its competitors. SSL protocol uses Message Authentication Code or MAC [162] to ensure that a communication cannot get altered without being noticed (message integrity).

One of the competitors can try to make Haati's website unavailable by launching a Denial-of-Service or DoS attack [163]; so, Haati takes measures against such possibility using firewalls [164], over-provisioning [165], upstream defense [166], etc. Everytime Jill or another customer places an order to Haati's website; the website makes a log entry for that order to facilitate accountability.

Jill's web browser and Haati's website runs a non-repudiation protocol [167], so that for each order placed, Neither Jill nor Haati can deny the order or make false

claims about it; e.g., in the absence of such a protocol, Haati can claim that Jill placed twice as much as she actually did.

We see that even if Haati authenticates itself to Jill's web-browser and authorization, confidentiality, message integrity, availability, accountability, and non-repudiation, all, are correctly implemented, still if someone gets hold of Jill's password, the chain of security breaks down. A strong password is a long and random sequence of symbols drawn from the symbols available on a keyboard; but, it is harder for a user to memorize and remember correctly [2]. In this thesis, we propose and evaluate a novel password creation interface that helps a user to memorize a newly created password; as, a user will, with our interface in place, now be more confident about her memory of a newly created password, we hope that she will be motivated to create a strong password.

1.3 Major Contributions

Here we briefly state the contributions of the dissertation:

1.3.1 Design, Development, and Evaluation of ePet

People do critical tasks on their smartphones and over time sensitive data can gather on the phone. A security-conscious user may setup a locking mechanism on her phone, e.g., password or pass-pattern; but, unlocking the smartphone again and again throughout the day can frustrate her. To mitigate this problem, we propose and evaluate ePet, which senses a subset of the traits of the user in a cost-saving manner and makes the authentication decision on its own relieving the user from the burden of active participation most of the time.

1.3.2 Effect of the Declarative and the Motor Memory on Password-Input

After many logins using a particular password, the declarative memory [168] for a password is augmented with a motor memory [134] in a way that a user may not be fully conscious of each individual character of her password while inputting it; the improved muscle-activations of her fingers facilitate the input process for that particular password. Through a within-subjects study involving eight conditions, we

investigate on the impotance of motor memory in inputting a password correctly. Our empirical results show that the declarative memory for a password is sufficient for a user to be able to input a password successfully, albeit with increased time and error; the role of motor memory is auxiliary, for it increases efficiency of the input process but does not prevent against failure of declarative memory.

1.3.3 Design, Development, and Evaluation of Password Creation Web-Interface

In a traditional password creation interface, a user creates a password and confirms it by retyping once; otherwise, remembering the new password is solely the user's burden; she might not remember it a couple of minutes later. The lack of confidence about the memory of a new password might inspire the user to choose a weak password or to reuse a well remembered password that she uses for another website, both of which weaken security. In order to alleviate the problem, we propose and evaluate a novel password creation interface, which guides the user through a memorizing drill built upon her new password. As a result, after successful completion of the drill, a user is more confident about the memory of her password. Instead of requiring the user to type in her password many times—which might help build motor memory—we focus on remembering the ordered contents of the password; in other words, we help cement the declarative memory for the password.

1.4 Dissertation Organization

The rest of the thesis is organized as follows:

- In chapter 2, we discuss the state of art of end-user authentication both for smartphones and for the Web and attempt at placing our approaches in the overall classification.

- In chapter 3, we elaborate on the ePet authentication framework, then we provide pseudocode for one possible realization of a subset of the framework. Next, we describe the algorithms that we used in our implementation of the subset of ePet on Android platform. Based on the data collected from users, an evaluation of the effectiveness of the implemented subset of the authentication

framework, its battery drainage, and users' opinions follow.

- In chapter 4, we take a closer look into the password input process and provide a conceptual model for the password input process. Then we provide details of the within-subjects study that was carried out to learn about the effect of motor memory on successful password input. We conclude the chapter stating the results and a brief discussion of the findings in the light of the proposed conceptual model.

- In chapter 5, we propose a novel password creation web-interface and discuss its design. Then we go on discussing the findings from the data we have collected from a semester-long between-subjects study, with a goal to evaluate the usability gains from using our proposed password creation web-interface.

- In chapter 6, we conclude the dissertation summarizing the research achievements and outlining directions for future research.

1.5 Publications

1.5.1 Publications on ePet

- **Mohammad Tanviruzzaman**, Sheikh Iqbal Ahamed, Chowdhury Sharif Hasan, and Casey O'brien. 2009. ePet: when cellular phone learns to recognize its owner. In Proceedings of the 2nd ACM workshop on Assurable and usable security configuration (SafeConfig '09). ACM, New York, NY, USA, 13–18. DOI=10.1145/1655062.1655066 http://doi.acm.org/10.1145/1655062.1655066

- **Mohammad Tanviruzzaman** and Sheikh Iqbal Ahamed. 2014. Your Phone Knows You: Almost Transparent Authentication for Smartphones. (*Accepted in the The 38th Annual International Computers, Software & Applications Conference.*)

1.5.2 Works That will be Benefited by ePet

- **Mohammad Tanviruzzaman**, Rizwana Rizia, Sheikh Iqbal Ahamed, and Roger Smith. 2012. Americans with Disabilities Act—Compliance Assess-

ment Toolkit on Smartphone. In Proceedings of the 2012 IEEE 36th Annual Computer Software and Applications Conference (COMPSAC '12). IEEE Computer Society, Washington, DC, USA, 442–451. DOI=10.1109/COMPSAC.2012.65http://dx.doi.org/10.1109/COMPSAC.2012.65

- Rizwana Rizia, **Mohammad Tanviruzzaman**, and Sheikh Iqbal Ahamed. 2012. KnockAround: Location Based Service via Social Knowledge. In Proceedings of the 2012 IEEE 36th Annual Computer Software and Applications Conference (COMPSAC '12). IEEE Computer Society, Washington, DC, USA, 623–631. DOI=10.1109/COMPSAC.2012.88http://dx.doi.org/10.1109/COMPSAC.2012.88

- **Mohammad Tanviruzzaman**, Casey Obrien, Rizwana Rizia, Sheikh Iqbal Ahamed, and Roger O. Smith. 2011. iFactotum: Sensor-Rich iPhone as a Versatile Tool. In proceedings of the 3rd International Symposium on Quality of Life Technology (isQOLT 2011), Toronto, Canada, 2011.

- Chowdhury S. Hasan, Sheikh I. Ahamed, and **Mohammad Tanviruzzaman**. 2009. A Privacy Enhancing Approach for Identity Inference Protection in Location-Based Services. In Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference - Volume 01 (COMPSAC '09), Vol. 1. IEEE Computer Society, Washington, DC, USA, 1–10. DOI=10.1109/COMPSAC.2009.11http://dx.doi.org/10.1109/COMPSAC.2009.11

1.5.3 Publications on Password Creation Interface

- **Mohammad Tanviruzzaman**, Praveen Madiraju, Sheikh Ahamed, and Rizwana Rizia. 2014. Coping with Passwords: Approaches for Smartphones and the Web. (*Submitted to the ACM Transactions on Computer-Human Interaction.*)

1.5.4 Works That will be Benefited by Password Creation Interface

- Rezwan Islam, Sheikh I. Ahamed, Chowdhury S. Hasan, and **Mohammad Tanviruzzaman**. 2009. Towards Universal Access to Home Monitoring for

Assisted Living Environment. In Proceedings of the 5th International Conference on Universal Access in Human-Computer Interaction (HCI 2009), San Diego, CA, USA, July 19–July 24, 2009.

Chapter 2

Related Work

"Open Sesame," uttered by Ali Baba in the story "Ali Baba and Forty Thieves" is one of the oldest passwords; it was simple, user-friendly, and efficient. The story also reveals some problems with passwords, like: Ali Baba stole the password by over-hearing the thieves and Ali Baba's brother, Cassim, forgot the exact words and could not get out of the cave and eventually got killed by the thieves [4]. Nearly half a century of research on overcoming the faults of a password while retaining its benefits has not seen great success and passwords remain to be the principal method of end-user authentication [5]. In this chapter we discuss research works related to end-user authentication with a view to placing our work in context.

End-user authentication usually consists of five components [3]:

1. A set $\mathcal{A}$ of authentication information, which the user provides to the system to prove her identity.

2. A set $\mathcal{C}$ of complementary information—usually derived from the authentication information—that the system stores and based on it, the validation is performed.

3. A set $\mathcal{F}$ of complementary functions that compute the complementary information from the authentication information: for $f \in \mathcal{F}$, $f : \mathcal{A} \mapsto \mathcal{C}$.

4. A set $\mathcal{L}$ of authentication functions that verify identity: for $l \in \mathcal{L}$, $l : \mathcal{A} \times \mathcal{C} \mapsto \{ \textbf{true}, \textbf{false}\}$.

5. A set $\mathcal{S}$ of selection functions that enable the user to create or alter the authentication and complementary information.

Depending on what information constitutes the set $\mathcal{A}$, first we shall organize the related works in five categories: something the user knows, something the user has, something the user is, something the user does, and someone who knows the user.

2.1 Something the user knows

2.1.1 Conscious Knowledge

Password

Alphanumeric password is the de facto standard of end-user authentication on the Web. The strength of a password is determined by the amount of time an attacker needs to guess it; for that matter, a strong password is a long string of characters, where each character is randomly drawn from a sizable alphabet. For example, if a password has N characters in it and each character is uniformly drawn from an alphabet of size A, then an attacker has to try $\frac{A^N}{2}$ guesses, on average, to break it. A popular metric for comparing strength of passwords is the Shannon's entropy [12]. However, in reality users prefer more memorable passwords over less memorable ones, which perturbs the password-selection probability and works in favor of the attacker. One way around this problem is to randomly generate a password for the user. But forcing a difficult-to-remember password upon the user may result in the user writing it down in an insecure place, e.g., on a post-it [13]. Another more user friendly alternative is to let the user chose her password, but from a restricted domain. In this vein, reactive password checking periodically runs a password cracker and if it finds a password to be easily guessable, it notifies the corresponding user. Proactive password checking works when the user is trying to select a new password; if it finds that the chosen password does not meet the minimum requirements, it rejects the password and the user then tries with another choice [2]. The problem remains that if the rules of a password checker is too restrictive, it is actually shrinking the password space significantly and making it possible for an attacker to perform brute-force search. On the other hand, a user may still find it difficult to remember the password accepted by the system and she may write it down.

Password Composition Policies

Policies mandate how a password should be composed. The idea is to force the user into creating a strong password. There are three types of policies: explicit, implicit, and external. Explicit policies give the user a set of rules upfront and the user then

creates a password that meets the criteria. Implicit policies react to user-chosen passwords by accepting or rejecting them. External policies improve the security by controlling the password fully (proposes a generated password) or partially (modifies a base password).

Forget et al. [6] propose an explicit policy where ideas from Persuasive Technology are utilized to influence users in creating more secure passwords. The user chooses an initial password (minimum of 8 characters) and then the system either replaces some of the characters in the password at random positions with randomly chosen characters or the system inserts a few randomly chosen characters into the user-chosen password at random positions. The user then can shuffle the resulting password to find one that she thinks is memorable enough. Weir et al. [7] propose an implicit policy where a cracking algorithm learns: the frequency of people's usage of certain words in their passwords, how people mangle cases, the basic structure of passwords, the probability of digits and special symbols, etc.; and based on the acquired information, the algorithm then constructs a probabilistic context-free grammar, which models how people create passwords. When a user submits a password of her choosing, the algorithm computes how likely it is that the password came from the grammar and using a preselected threshold can either accept or reject it. Shay et al. [8] conducted a paper-based survey among 470 students of Carnegie Mellon University on their feeling and attitude towards the, then newly adopted, password composition policy at CMU, which was stricter than the previous one. They found that the users find new requirements annoying but believe that the requirements provide better security; some users struggle to comply with the new requirements; the users are more likely to share and reuse than to write down; the users tend to modify old passwords to create new ones; over time the likelihood of sharing passwords increases, and use of dictionary words and names are the most common strategies to create passwords. Using Amazon's Mechanical Turk, from a user-study among 5000 users, Komanduri et al. [9] found that long and simple passwords obtain better trade-off between usability and security than short and more complex passwords. Kim and Huh [10] found that the stricter the policy, generally, the more secure the PINs, the more the remembrance is difficult. All the policies

they explored mandate expiration: a user has to change her password periodically, so that the attacker has less time to inflict damage. Zhang et al. [11] devise an algorithm that—through a set of primitive transforms—guesses new passwords from old expired ones, with 41% accuracy and within seconds; and for 17% of the passwords, five online guesses are sufficient for cracking. Accordingly, they propose to eliminate password expiration altogether. Another aspect common to many password policies is lock-out or rate-limiting to limit the number of guesses an attacker can try online.

That the policies usually vary a lot across different websites exacerbates the already daunting problem of remembering passwords; and no memory aids are provided in the policies to make remembering passwords easier for the user. Consequently, the user chooses the password that is the easiest to remember and barely meets the criteria, but presumably a weak one; or from exasperation of being rejected, do not care to remember the password and frequently retrieves to fallback authentication, or just writes the password down and keeps the note in easily accessible but unprotected places.

Passphrase

To help a user remember a strong password, passphrases were introduced; where a user chooses a long phrase of words and from each word of the phrase, she then chooses one or more characters—possibly with some transformation, like leet, applied to the chosen characters. The motivation for using passphrases is, the meaningfulness of the words in a phrase will help the user remember it, yet the random choice of a phrase along with the random choice of the characters from the words accompanied by a transformation would make the resulting password strong.

Kuo et al. [14] examined if it were possible to build a dictionary to crack mnemonic phrase-based passwords. They built a 400000- entry dictionary of phrases gathered by screen-scraping various sites, like: advertising slogans, childrens nursery rhymes and songs, movie quotes, famous quotations, song lyrics, etc. With the dictionary at hand, the authors were able to crack 4% of mnemonic passwords whereas with a standard dictionary with 1.2 million entries they were able to crack 11% of control passwords. The user- generated mnemonic passwords were more resistant to

brute-force attacks than control passwords. The authors concluded that mnemonic passwords could become more vulnerable in the future and should not be treated as a final solution. Shay et al. [15] found that system-assigned passphrases and passwords were forgotten at similar rates, led to similar level of difficulty and annoyed the users equally, and the users frequently wrote down both. However, the users needed longer time to type in passphrases and error- correction appeared to be a necessity for counteracting input mistakes.

Graphical Password

From research on human memory, it is known that if someone learns concepts by viewing pictures, she is likely to recall those more easily than if she learned them by viewing their written word-forms; the name for this phenomenon is the "picture superiority effect" [16]. Taking inspiration from the picture superiority effect, researchers have proposed various graphical password schemes of three types: pure recall, cued-recall, and recognition [17].

Wiedenbeck et al. [18] propose PassPoints, a cued recall based graphical password system, where during registration the user chooses a set of points on an image; and later, at the time of authentication, she needs to click on those chosen points in the correct order to prove her identity. They found that the user has trouble remembering the points when the tolerance region around a point is small, e.g. 10×10 pixels. In a PassPoints system—based on the idea that the number of focal points in an image is usually limited—how likely it is that a point will be chosen by the user, has been modeled in [19]. For two images, the accuracy of the models prediction were 80% and 71%, which shows that some images are better at thwarting automated attacks. From user-studies on PassPoints, Chiasson et al. [20] found that a user can remember her password points even if the tolerance region is small. They also found that for password-points in an image, the users select easy-to-remember points, geometric patterns, color patterns, and personally meaningful items in an image. In another study, Chiasson et al. [21] found that in the short-term, PassPoints passwords are more robust than text passwords against multiple password interference. However, after two weeks, recall of the textual passwords and the graphical

passwords were not statistically different. Jermyn et al. [22] propose Draw-a- Secret (DAS), where the user draws a free-form picture on an $N \times N$ grid and this drawing serves as her authentication-secret. The secret is encoded as an ordered sequence of cells that the user crosses while drawing it. Two secrets are identical, even if they do not look exactly the same, if their encodings are the same; thus there is some leeway in reconstructing the secret. In [23] Dunphy and Yan show that the users' choices for DAS have some predictable characteristics, e.g. symmetry; and using background images with DAS they were able to decrease those predictable characteristics of DAS while increasing its memorability. Three variations of DAS to foil shoulder-surfing attacks—have been proposed in [24]. "Decoy Strokes" produces spurious strokes while the user is drawing her secret picture. "Disappearing Strokes" produces the effect of writing on a fluid surface where a stroke starts disappearing after a while of having been drawn. "Line Shaking" gives the attacker even less time to look at a stroke than in Disappearing Strokes; because a stroke starts disappearing even while it is being drawn. Through user-studies they found that among the three variations, Disappearing Strokes is the best trade-off between security and usability. Through a user-study Komanduri and Hutchins [25] found that both character and picture passwords of very high entropy were easily forgotten and they found that serial ordering was the main cause of failure; thus picture-password systems that do not require ordered input may produce memorable, high-entropy passwords. They found in the study that the users often replicate the same errors in their login attempts which cut down the effective number of guesses a user is allowed while authenticating. Hayashi et al. [26] propose a graphical password scheme based on the human ability to recognize a degraded version of a previously seen image. During registration the user chooses p images from a set of n images ($n > p$) as her portfolio and she gets a chance to see the distorted version of the p images that belong to her portfolio. During authentication she has to correctly choose the p images from the set of n images where all n images are distorted. They found that too much distortion makes it hard for even the legitimate user to recognize her portfolio images, whereas too little distortion allows the attacker to perceive a lot about an image; and the attacker then can apply social engineering

attacks to learn what the likely portfolio images of a user are. A similar idea has been explored in [27] where the user needs to recognize degraded versions of their chosen images. The authors notice that only one bit of entropy can be extracted from each image, making the authentication time too long to be practical and they conclude that the approach may be suitable for password recovery, rather than for regular authentication. Hayashi et al. [28] shows that their idea of degrading images for increased security [26] can indeed defend against educated guesses even when the attacker knows the target well. Passfaces [29] is a commercial graphical password system where the user chooses p images as her portfolio and during authentication she is presented with p consecutive screens, each having a 3×3 grid of 9 images and one of the 9 images belongs to the users portfolio; the user needs to identify all p of her portfolio images in the p screens. Dunphy et al. [30] examined if Passfaces password can be described verbally to others and they found that Passfaces passwords can be made more secure against verbal description if portfolio images are presented with visual and verbally grouped decoys. A comprehensive study of multiple password interference with Passfaces passwords revealed that participants who used four distinct passwords a week were ten times more likely to completely fail to authenticate than participants who used a single password a week [31]. Khot et al. [32] proposed a variation where each of the users portfolio images has a set of tags associated with it and during authentication, the system instead of presenting the exact image from the portfolio presents some image that has the same tag from Google search. Mostly to thwart shoulder-surfing, for public terminals, De Luca et al. [33] propose a gaze-based graphical password system where the user needs to draw a shape by connecting points from a 3×4 grid of points using her eye-gaze. Locimetric eye-gaze based method proposed by Forget et al. [34] utilizes gaze-clicks on specific points of one or more images. Bulling et al. [35] removes visually appealing parts of an image through automated image processing so that the user chooses a gaze-based password which is not obvious. Dunphy et al. [36] look into intersection attacks, shoulder-surfing attacks, and importance of image processing for Passfaces password on mobile devices. To prevent intersection attacks, they propose to use a decoy image portfolio in addition to a key image portfolio; to prevent image quality

based attacks, they propose using key and decoy images from the same source; and they propose to automatically filter out visually similar images or less memorable images. Wright et al. [37] tried to bring in the better memorability of recognition based graphical passwords into textual passwords; but, they did not find any statistically significant difference in memorability between the recall and recognition based systems. Furthermore, recognition based system took longer to authenticate.

In summary, in pure-recall type graphical passwords, a user chooses a shape on a grid by joining various grid-cells in some order; and during authentication she has to reproduce the shape. This type of graphical passwords are hardest to remember for a user, so users typically chooses predictable and easily memorable shapes. In cued recall type graphical passwords, the user selects some points on an image with rich background in a particular order; and during authentication, she has to select those points in that order. A problem with cued recall type password is the presence of hotspots; users like to choose points from certain parts of the background image which reduces the password space. Among the three types of graphical passwords, recognition based password is the most memorable kind. Here a user is presented with a set of grids, each grid containing multiple pictures; the user selects one picture from each grid and the set of chosen pictures form her password; during authentication the grid of pictures are again presented to the user, probably each grid is shuffled, and the user has to choose the correct picture from each grid. Human memory for faces is near limitless; so a recognition based graphical password using human faces as the grid-pictures is very usable; but, the problem is, humans tend to choose faces from the similar ethnic groups and other predictable choices kick in. In order to mitigate the predictability of user's choice, degraded versions of the actual images are put on the grid, taking advantage of the fact that once a person has seen the actual picture, she can easily recognize the degraded version.

Graphical passwords are specially suitable for devices with touch screens, like: smartphones and tablets. On Android smartphones, pass-pattern, a pure- recall type graphical password, is a popular locking mechanism. However, smudge attack, which attempts to find out the pattern from the oily resides left on the touch-screen is a significant threat for the pattern based scheme [38]. If the grid is randomly shuffled,

the probability of success of a smudge attack can be decreased, but hurting the usability benefits [39]. All the graphical passwords increase the success of shoulder-surfing attack, where the attacker discerns the password by looking while the user is inputting it.

Cognitive Games

In order to thwart the shoulder-surfing attack [40], various cognitive games have been proposed. Roth et al. [41] propose two variants of a cognitive trapdoor game to defend against shoulder surfing attack on PIN entry. First variant defends against human observer and is based on the premise that human short term memory is limited. For each digit of the PIN, the user has to play several rounds. During each round, the 10 digits (0-9) on the PIN pad are divided into two groups and shown in two colors: black and white. Depending on white or black, to which group the current PIN digit belongs, the user clicks one of two buttons. The verifier, e.g. the ATM machine, knows which digit the user has input by taking intersection of the rounds. There were two sub-variants: immediate oracle choice and delayed oracle choice. The second variant, to some extent, defends against video recording of the entire PIN entry process; recording of multiple sessions can still reveal the PIN. In this variant, the intersection of the rounds does not reveal a unique digit. In comparison to traditional 4 -digit PIN entry, it takes about 10 times longer to log in using these methods. Lab-based user studies show that perceived security was higher, but usability of the methods were rated lower in comparison to the traditional PIN entry method. The proposed methods also caused more fatigue. The users seem to accept the oracle methods despite them being slow and strenuous, may be, they would not accept it so well in real life where the PIN entry is a secondary task.

Weinshall [42] proposed a cognitive authentication scheme where the user answers a query about a set of images where a subset belongs to the user's secret image portfolio. The user has to answer multiple such queries. If there are Q queries and each query has P possible answers, then an attacker making guesses at random has a probability $\frac{1}{P} \cdot Q$ of success. The number of queries to be used depends on the required security; but in general the more we want of the security, the longer it takes

to authenticate: even up to three minutes. However, Golle and Wagner [43] showed that Weinshall's scheme can be broken within seconds using SAT solver.

In summary, in the authentication schemes based on cognitive games, the user and the system shares a secret and the user reveals her secret knowledge a little at a time over multiple interactions; as a result, by observing a single interaction between a user and the system, it becomes difficult for the attacker to know the secret. In addition to being significantly lengthy, stressful, and error-prone, the scheme is vulnerable if the attacker can observe multiple login sessions.

2.1.2 Subconscious Knowledge

No matter how secure an end-user authentication scheme is, an attacker can always get hold of the user and coerce her to divulge the secret. Although the cost over benefit would prevent a rational attacker from doing so in general; there are some cases, like with a key government personnel who knows crucial state secrets, the benefit could prevail over the cost. For such situations, it is desirable—though counter-intuitive—that the user does not explicitly know the secret.

The proposed solution depends on the idea of implicit memory, like the memory of riding a bicycle; even though a person cannot tell somebody exactly how she rides a bicycle, when needed, she can ride one. Bojinov et al. [44] design a cognitive game where the user reacts to a series of stimuli during training. The series contains a covert repeating sequence which the user learns implicitly. During authentication the knowledge of the embedded repeating sequence is assessed by comparing the performance rate (percent correct) between the times when the cues follow the trained sequence and the times when the cues follow an untrained sequence; the training period, however, can be quite long: 30–60 minutes.

2.2 Something the User Has

Security tokens are used for "something you have" based approach to authentication. The user has to carry the security token with her to be able to authenticate. However, a security token frees the user from remembering complicated passwords; a token can store arbitrarily complex secrets, and the authentication process can be made

transparent to the user. For example, a user can tap her personal `YubiKey Neo` against a mobile device when she needs to unlock her phone [45]. But in case the user forgets to bring the token with her or she loses the token, there is usually no other immediate way to authenticate [12].

2.3 Something the User Is

This method of authentication refers to physiological biometrics of a user. From face-geometry to palm-vein geometry, various physiological traits of the user have been employed for authentication. In [48] acoustic properties of the outer flap of the user's ear and ear canal, were the basis for authentication. Force field transformation method proved more resilient to ear occlusion from wearing a hat or a scarf [49]. Face [50] and fingerprint [51] are popular physiological biometrics. Corneal reflections and iris-geometry give more accuracy as biometrics [54, 55]. Voice recognition gives moderate accuracy and is available on many smartphones [60]. Teeth geometry enjoys more invariance than most physiological biometrics [58] and vein geometry [59] is a relatively less-known option.

2.4 Something the User Does

Behavioral biometrics represent this form of authentication and it includes pattern of arm-sweep, where the owner unlocks the phone by grasping and shaking it or she may walk while holding the phone in her hand; acceleration sensor embedded in the phone is used to decide whether the shaking or arm- sweep pattern matches that of the owner [46, 47]. Haptics pattern is decided from velocity, force, torque, and angular orientation of the stylus; or finger pressure, movement characteristics of center of palm and fingertips [52]. Pattern of hip movement involves attaching acceleration sensor on the hip and detecting how the hip moves while the user is walking [53]. For keystroke dynamics, usually the user is required to type in a fixed string and then her typing pattern may be decided from interstroke latencies, hold time, error rate, etc. [56, 57]. Predefined gestures or the hand-movement a user makes while making a call have been utilized for authentication [72, 73]. A 3D

accelerometer or a smartphone with a built-in accelerometer—placed at ankle, hip, waist, trouser pocket, or arm—has been used to recognize a user's gait pattern [76].

Biometrics frees the user from remembering secrets and from the burden of remembering to always carry a token. But biometrics can also be unreliable: manual workers may lack sharp fingerprint; some fingerprint scanners do not work if the finger is unclean; face changes due to make-up, facial hair, glasses, weight-changes, or skin color changes from sun exposure; cold can change voice; hand geometry changes from injury, swelling, weight-changes, arthrosis, etc. [62] In order to mitigate the reliability problem with biometrics, a combination of biometrics called multimodal biometrics have been employed for authentication: voice and ear [61]; signature and speaker verification [63]; gait and voice [64]; teeth and voice [65]; face and voice [68]; face, teeth, and voice [66, 67]. Error rates may go up when combination of biometrics methods, that require active participation from the users, are used [75].

In a biometrics based authentication system, there are eight places where an attack can be mounted: spoofing attack, replay attack, substitution attack, tampering, masquerade attack, Trojan horse attack, overriding yes or no response, and insufficient accuracy [70] It is hard to revoke a biometrics because it is intrinsic to the user. The stability inherent to biometrics, provides a masquerader long time to design an attack after she has gotten hold of the master template; e.g., using neural networks, fingerprint images were reverse engineered from the master template stored as minutiae points [69]. A protected biometrics template should be non-invertible and non-linkable; there should also be a liveness detection technique [71]. People can feel uncomfortable when biometrics data are saved on a remote server [74].

2.5 Someone Who Knows the User

Somebody the user knows has been proposed as a fallback authentication mechanism in [77]. The user has a list of helpers and she can ask a helper to vouch for him during fallback authentication. The regular authentication involves token and PIN. If the user loses the token, she can receive a vouch-code from one of her helpers over telephone or in face-to- face meeting. Schechter et al. [78] found that the users forget

who they selected as their trustees and phone-based attacks by close acquaintances often succeed. Email based attacks did not fare well, because the trustees suspected something fishy in them. The participants took several hours at the least to acquire enough trustee-codes to pass the fallback authentication, which should not pose too great a problem; because the need for fallback occurs rarely.

2.6 Unorthodox Input Methods

A number of password input mechanisms–gaze-based, haptic, gesture-based, vibration-based, multi-touch, touch pattern, force-based, etc.—have been proposed, mostly to defend against shoulder-surfing attack.

In [79] users enter their password using their eye-gaze on a 1280×1024 screen at 96 dpi equipped with `Tobii 1750` eye tracker. Two methods of gaze-entry were explored: trigger based and dwell based. In trigger based entry, the user presses a specific key e.g. the space bar each time she has focused on a character. Time between consecutive presses may leak information. In dwell based entry, the user fixes her gaze on a particular password-character for a while before moving onto finding the next. Error was higher for the trigger based method, may be because; it was hard for the user to time and to coordinate between pressing a button and precisely fixating her gaze, which might be accounted for algorithmically utilizing the history of the user's gaze-path. Lab-based study on 18 users (50% male) of average 21 years of age reveals that the gaze-based password entry takes 5 times longer (average 10 seconds) than usual keyboard- based input. However, more than 80% users expressed their preference for gaze -based input in a public setting. Error was not significantly higher. Eyeglasses did not hamper the eye tracker. `QWERTY` layout proved faster than alphabetic layout, presumably due to user's prior experience with the former. Over time, the users may develop muscle memory in their eyes and become more accurate and faster in eye-gazing. The method does not produce any mouse or keyboard event and thus it is more resistant to keyloggers. The entropy of the gaze-based password (authors call EyePassword) can be increased further by taking into account the user-specific gaze-path and dwell time patterns.

Sasamato et al. [80] proposed an authentication scheme based on the assumption

that multi-sensory processing (dissociation and recombination of different sensory inputs) capability of humans is good. A prototype bank- terminal was built requiring 4-digit PIN-level security. The user's PIN consists of 5 images (called user's portfolio) of her choosing. In order to input her PIN, the user is presented with a series of 7 screens. Each screen consists of 5 images, 0 or 1 of which is from the user's portfolio. A trackball capable of 5 distinct movements (roll up, right, down, left and vibration) remains under the user's palm. For each screen the trackball makes one of the 5 possible movements and the user then responds by pushing one of the 5 buttons. Say, the third image from the left belongs to the user's portfolio, then the user should have pushed the third button from the left, but here, the trackball's movement acts like a random key with which the user xor's the "right" answer and inputs the result, like a one-time pad. The attacker fails to see the trackball's movement and thus has no idea which one among the presented 5 images belongs to user's portfolio. Because, $\binom{7}{5} \cdot 4^5$ is larger than 10^4, the method needs 7 screens. More images per screen would have reduced the number of screens (equivalently number of user interactions) but that would have complicated the instrument and also the user's cognitive load would be higher. The authors performed usability and security analysis based on a lab study with 38 participants (4 students and the rest government employee). The median login time is 32 seconds, 15 participants suggested that login time should be below 15 seconds. Overall failure rate was 26% and the younger users were quicker and more accurate. The authors noted that comparison of a novel system with the traditional PIN entry is difficult due to the extended experience the users have with the PIN system. There were two video cameras in the room. Many users failed to cover the trackball. Some points to the map, some moves their hands to better feel the trackball's movement, and some spoke out loud. All these leak information. 9 out of the 38 users leaked information in some way. Some of the users saw the experiment as an exam and probably behaved unrealistically. The bank-terminal could be a fake and the whole process would then fail. Very high quality sound capture, e.g., with a parabolic microphone may leak information due to variation in the instrument's (motor's) noise. All legacy deployments have to be modified significantly. Interestingly, the paper is devoid of

any statistical test. Social engineering attacks might be possible due to the user's self-selection of portfolio images. Moreover, two attacks on Undercover have been reported [81], analyzed, and evaluated. First attack utilizes non-uniformity of human behavior. The button layout corresponding to "Upward rotation" of the trackball was $\langle 1, 2, 3, 4, 5 \rangle$ which allowed the user to respond quickly, as they did not have to mentally account for a rotation in the layout. This time difference in response was exploited successfully. The authors proposed to modify the layouts so that each of them requires the same amount of mental effort for the user to respond. They also found that if the 5 pass-images have already been shown and there remain one or two screens to show, the user knows by that time that the remaining screens will not have a pass-image and thus responds more quickly. They modified the system so that the last screen always had a pass-image in it. The modified system has lower entropy but still it is higher than 4-digit PIN. The second was an intersection attack. Each public challenge (each screen) in Undercover contains at most one pass-image. From the attacker's point of view, if more than two images in a candidate 5-image password appear on a screen she can immediately discard the password as invalid, reducing the effective password space.

Liu et al. [82] propose a gesture based user identification and authentication method. uWave is the underlying gesture recognition system and it requires a single tri-axis accelerometer. Users select their own gestures and training with a personalized gesture can be done with a single sample. A user study with 25 participants was conducted over one month. For identification 98 % accuracy was achieved. Without visual disclosure the equal error rate was 3 % for authentication. With visual disclosure (likely for gestures), false positive increases to 10%, implying that gesture based authentication can be used only when mild security is needed or some other methods must be employed alongside. For user-identification (e.g., multiple accounts on TV), not surprisingly, it was found out that selection constraints improve accuracy as the complexity constraints eliminate simple gestures that can be easily confused with each other and the rejection procedure guarantees enough difference between gesture templates; also template-replacement at regular intervals, improves accuracy. Compared to text ID, users did not find it more difficult to remember or to perform

gestures. For user-authentication (password gestures), users chose highly symbolic gestures such as regular shapes, letters in their native languages. Users also chose gestures having personal meaning; because, it helped them remember complicated gestures. These patterns, in user-choices, make gesture passwords vulnerable to dictionary or social engineering attacks. Moreover, sharing gesture passwords with others is easy.

In "VibraPass," the mobile phone's vibration capability has been utilized in securing PIN entry [83]. The authors propose to add noise to the entered PIN or password to combat observation attacks. The noise here is "lies" communicated over Bluetooth to the user as vibrations of her mobile phone (in pocket). For example, if a user's PIN is 9362 and the lie sequence is $\langle 0, 1, 0, 0 \rangle$, the user may enter $9\,\underline{5}\,6\,2$, where 5 is the "lied" digit. A user study with 24 participants (average age 23 years, 8 females) reveals that error rates increase with higher lie overhead and longer password. Interaction time increases with higher lie-overheads and longer passwords. To simulate attack, two video cameras and two microphones were setup and the study was performed in a quiet environment. Out of 749 successful authentication sessions, with a lie-overhead of 0% breaking-success was perfect and with a lie-overhead of 30%, 50%, or 100%, only 32.5% could be broken. Two main causes of successful attacks were: audible vibrations due to keys in the users' pockets and "bad lies." Bad lies include repeated pressing of the same key, confused waiting before pressing, and using characters as PINs. Lie-overhead of 30% seems to be a good trade-off between security and usability, having error rates and interaction time close to those of PIN/password. High quality sound recording, intersection attack from observation of multiple sessions (authors argue that compromised ATMs are replaced very quickly and thus multiple sessions are unlikely to be observed), compromised Bluetooth module, and timing attacks based on non-uniform user behavior (like [81]) could degrade security of VibraPass. From usability standpoint, the user may not have a pocket to hide the phone and she may fail to perceive a vibration.

Bianchi [84] devised a haptic keypad to defend against shoulder-surfing attacks. The keypad has three buttons, each can vibrate with 1 Hz, 2 Hz, and continuously. These three kinds of vibration are called tactons. A user's PIN is a sequence of

tactons, e.g., $\langle 2, \ldots, 1 \rangle$. The system assigns randomized tactons to the three buttons and the user needs to push the button with the correct tacton. Two modes, called normal and hybrid, were tested. The normal mode having 6 digits was found to perform better with a median authentication time of 22.2 seconds. However, it has a small PIN space of $3^6 = 729$. User study was conducted with 12 participants having mean age of 29. Security hinges on the assumption that the observer will not be able to infer which button has which tacton, but high quality sound recording may render the assumption incorrect.

A number of authentication schemes suitable for tabletops have been explored in [85]. Tabletops pose a unique problem for authentication because many people (e.g., friends) can be around the table and thus PIN or password entry is inherently vulnerable to observation attack. "ShieldPIN" allows user to use her one hand to create a crescent shaped shield and the other hand then inputs the PIN on a soft PIN-pad. The shield may not cover all directions and thus some people around the table may see the entered PIN. "SlotPIN" has 10 rows and 4 columns of digits. Except for the first column, the user can rotate the column like a wheel so that one of the 10 rows contains her PIN. As there are 9 other rows, the observer does not know which row is the PIN. Nevertheless, "SlotPIN" is vulnerable to intersection attacks, where the attacker gathers information from multiple sessions and the row that remains constant over all sessions must be the PIN. "CuePIN" is a mixure of the previous two—the user makes a shield with her one hand and a letter between A and J appears inside of her shield. Then the user rotates the column to put the current PIN digit in the letter's row. He has to do that for each digit of her PIN. Because one letter is much easier to shield than an entire PIN- pad, CuePIN is harder to break; even so, it leaks information and an intersection attack over multiple sessions can break it. "Color-Rings" is similar to the Convex Hull Click [86]. The user has some circles and she has to place the circle so that her portfolio image is inside that circle. Since there are multiple circles and there are multiple images per circle, an attacker cannot know the PIN from a single session, but intersection attack is still viable. "PressureGrid" is an enhancement to the PassFaces system where the user does not directly clicks on her portfolio image, but she chooses the x and the

y coordinate of the image in a grid of images (3×3) with her two hands. Now she uses three fingers per hand and which finger is actually selecting a coordinate of the image depends on the pressure applied by a finger. Hopefully the observer does not see the pressure difference across the fingers and she does not know which image was selected. Average duration of a login using PressureGrid was 10.8 seconds.

In [87], the idea is to defend against observation attacks on PIN input at ATM terminals while keeping time to login comparable to standard PIN method. The user needs to interact with 4 screens to input 4 digits of her ColorPIN. Each screen contains 9 digits arranged in a 3×3 grid and each digit has 3 letters in 3 different colors just below it. On a screen there are 9 letters in total, which means a letter occurs at 3 random positions on the screen having distinct colors. In addition to the PIN, the user needs to remember a color sequence, which means her ColorPIN consists of a PIN and a sequence of colors. To enter a digit, the user chooses a letter below the digit having the right color (which she remembers). The space of ColorPIN is $\frac{27^4}{3^4} = 9^4$. If an attacker records the keypad entries only, she does not know your PIN. On the other hand, if she records both screen as well as the keypad, she may perform an intersection attack. If the attacker knew your PIN is your birth year and if she could see one entry process, she still has 81 choices, which makes ColorPIN more robust. But the cognitive load can be high, because the user has to look at the screen to find out the appropriate letter and then she has to input that on the keypad. The legacy ATM deployments have to change. A lab-based user-study was performed with 24 volunteers (18 males) having average age of 28 years. Average login time was found to be 13.33 seconds, which decreased to 3.5 seconds after 4 authentication sessions, implying that with experience a user could enter her ColorPIN quickly. Error rate was not significantly high. Only 2 out of 38 ColorPIN sessions were compromised by single session observing which is significantly better than regular PIN. In those two cases, the user actually pointed to the digit on the screen with her other hand so that she can input the correct letter on the keypad.

Vulnerability of an authentication scheme for magnetometer-equipped smart phone has been explored in [88]. The user produces a signature around the device with a magnetic token, stylus, or finger ring and that can be sensed in 3D by

the phone's magnetometer. Dynamic Time Warping (DTW) is used to match the current signature against the stored template. The magnetic signature was recoded using 4 cameras from four angles: front, rear, left, and right. 22 participants tried to forge the signature by observing the video-recording. With a matching threshold of 1.67 the number of successful attacks is zero while authentication of legitimate users was always successful.

For blind people, a smartphone PIN entry method utilizing multi-touch enabled screen–called PassChords—has been proposed to defend against aural and visual attacks [89]. The user places her four fingers on the screen to mark reference points. Then she touches with one or more fingers to enter one digit. The PassChords algorithm uses Maximum Likelihood to determine which fingers touched the screen. No audio or video feedback is provided, only vibration feedback is given after referencing. Proprioception removes need for further feedbacks. Theoretical entropy of PassChords is $4 \cdot \log_2(15) = 15.6$ bits, because any combination of four fingers can be touching the screen. Effective entropy is lower because some finger combinations are more likely than others due to the physiology of the hand, e.g., simultaneously tapping the middle and pinky fingers is more difficult than tapping the index finger. A user study with 16 legally blind, smartphone using user with average age of 51 years was conducted. The mean authentication time for PassChords was 2.67 seconds whereas VoiceOverPIN tool 7.52 seconds on an average. Index finger was most frequent (66.5%) and the most common length was 3 taps. Empirical first-order entropy was 12.6 bits for PassChords and was 12.7 bits for VoiceOverPIN. PassChords may be vulnerable to shoulder-surfing attack on finger movements. In [90], the user of a smartphone puts her finger on the screen and without moving the finger pushes in one of four directions: up, down, left, and right, to enter one digit of her passcode. If there are 4 digits in her passcode, the passcode space is $4^4 = 256$ which is lot less than that for a regular PIN. However, the motivation was to defend against observation attack. `Tekscan Flexiforce Model A201` was used as a sensing unit. A user study with six participants revealed that the users would like to use less force and more directions, e.g., diagonal.

In [91] the user uses a permanent magnet shaped like a ring or pen to "write"

her signature in air around the free space of her smart phone. Authentication is based on the temporal variation in the magnetic field sensed by the magnetometer of the phone. Training phase receives and stores the user's signature and in the authentication phase Dynamic Time Warping is used to match the current signature against the template. The user has to click a button to signal that she is about to input her signature. Authors claim that 3D signature is hard to copy and it allows for more options for possible signatures than its 2D counterpart. Observing the signature or even recording it is very easy and it should be tested how hard it is to recreate the 3D signature for an attacker.

A password entry method for touch screen enabled mobile phones have been proposed in [92], where authentication is based on user's pass shape and also on a set of behavioral biometrics (exhibited while entering the pass shape): how hard the finger presses, area of the finger touching the screen, x and y coordinates, and time pressed between two different points. Dynamic Time Warping algorithm was used to compare current set of biometrics with the template. Even if the attacker knows the pass shape, she is unlikely to reproduce the set of biometrics with enough accuracy. A user study with 31 participants with average age of 27 years reveals that overall accuracy was 77% with maximum warp distance as the threshold of comparison. The authors note that a dynamic template which evolves over time could increase accuracy. An attack was defined as a comparison between a target user and all other users which is not what happens in a real attack where the attacker observes the users input process and then tries to impersonate.

2.7 Fall-back Authentication

When a user forgets her password for an account, she falls back to password- recovery mechanisms, e.g., a temporary password is sent to her primary email account or to her smartphone, or she must answer some preset security questions, or she can request a number of preselected social acquaintances in a secure way (e.g., in person or over voice-call) to vouch for her identity.

Rabkin [93] analyzes over 200 personal security questions used by 20 banking or brokerage websites for fallback authentication and concludes that the questions

provide, at best, weak security. Availability of personal information on the Internet weakens security of personal knowledge based questions. The author, as a remedy, advises to allow the user to come up with her own security questions; advises the user to answer approximately correctly, to use questions with ephemeral answers (e.g., about recent browsing history) or preference-based answers. He also suggests that the system should guide the users in choosing secure questions. The attack on personal security questions is recognized as an information retrieval exercise and the author suggests making the exercise harder through embedding multimedia contents into questions, e.g., asking questions about a user- selected image. Schechter et al. [94] looked into the personal security questions used for password recovery by AOL, Google, Microsoft, and Yahoo! They explored the security and memorability of the questions and found that untrusted acquaintances could correctly guess 17% answers, users forgot 20% answers within six months, and 13% answers were one of the five most popular answers.

2.8 Password Manager

It relieves the user from the burden of remembering numerous passwords by storing them; all she needs to do is to remember one strong password to unlock the password manager itself [95]. Password managers can be built into browsers and their auto fill-up option makes it unnecessary for the user to type in her password every time she logs into a site a very convenient feature. But the browser based password manager is confined to a particular machine. Web-based password managers can be accessed from any machine. Password managers generally create a single point of failure so that an attacker can concentrate her effort on cracking the master password of the password manager; once she cracks the master password, she gets all passwords of the user. Figure 2-1 shows the overall classification of various methods for end-user authentication.

For smartphones and tablets, depending on the granularity level of access that a successful login ensues and how much user involvement an authentication scheme requires, there are two broad categories of authentication schemes: graded security and implicit authentication.

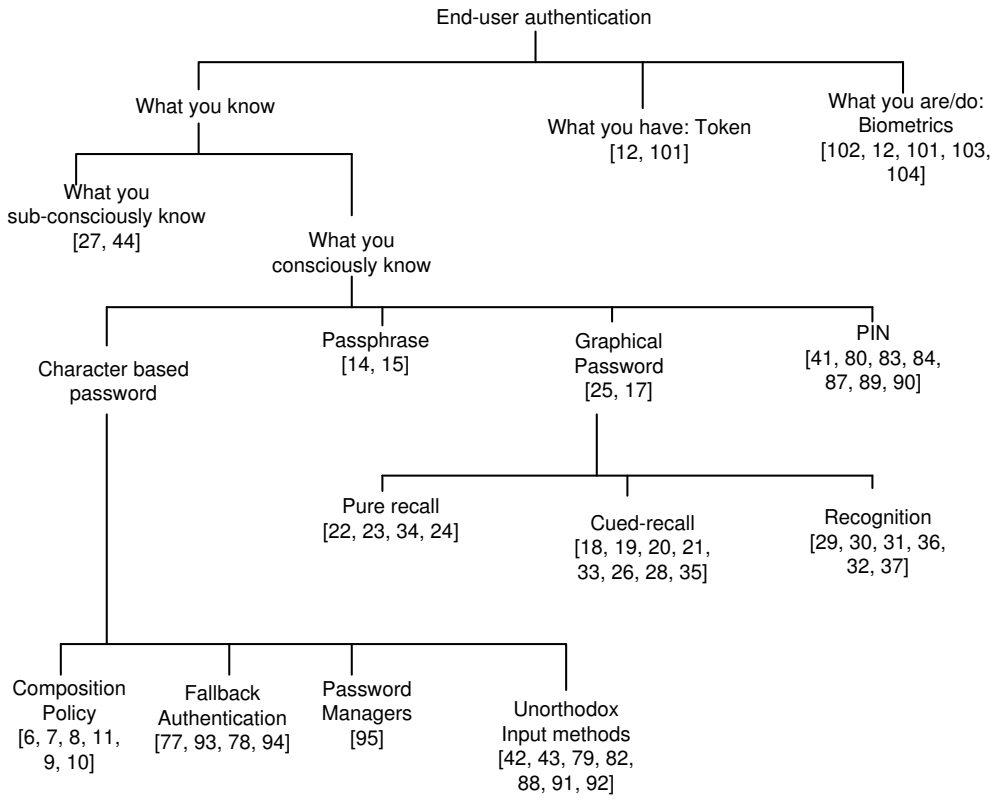


Figure 2-1: Authentication methods

2.9 Graded Security

In contrast to the traditional "one shoe fits all" approach to security, graded security is the concept of providing different levels of security to different applications and data, e.g., SMS application is secured with a simple visual code.

A focus group discussion was conducted in [96] with 19 participants on different authentication methods, like: fingerprint with swipe sensor on laptop, 2D gestures on the touchpad of a laptop, 3D gestures like `Wii` but using mockup, iris or face using phone's camera, keystokes on laptop keyboard, points on a picture, and speaker recognition. Fingerpint was found to have significant lead both in terms of security and usability. Through an interview of 20 users who had a smartphone and a tablet, Hayashi et al. [97] found that all-or-nothing access control was a poor fit for users' preferences, both on smartphones and on tablets. The participants wanted roughly half of the applications to be available even when the device was locked. The users showed positive attitude towards face and voice biometrics. Several parents who had young children expressed the concern that children could accidentally delete important information from the device; so, a simple way to let the children access games was a preference for those parents—may be with a simple shared PIN.

2.10 Implicit Authentication

User's identity is transparently inferred in implicit authentication schemes. In [98], the probability of the user being at a certain place determines how much confidence to put on the identity of the user. In [99] a learning algorithm, given the user's past behavior in terms of her usage of emails, calls, SMSs, GPS coordinates, contacts, etc., assuming all tasks to be independent, outputs a model of the user and the model is then used for implicit authentication.

2.11 Placing the Work in Context

Our proposed framework for end-user authentication for smartphones, ePet, utilizes ideas from graded security, implicit authentication, and multimodal biometrics based approaches; and with the pet metaphor—all three work in harmony. With ePet, a

user can choose different security settings for different applications (graded security) and the phone authenticates the user without requiring her active participation (implicit authentication) most of the times. The actual authentication process in ePet is based on the concept of traits, which are, in essence, various combinations of biometrics of a user.

The password-creation interface that we propose for the Web, on the other hand, is an attempt at coping with the memory-related issues of a password.

Chapter 3

Usable End-user Authentication for Smartphones

Worldwide there are around one billion smartphone users and this number is expected to reach 1.75 billion by the end of 2014 [106]. Nowadays, beside making calls, people are using smartphones for a wide range of tasks; e.g., social networking, mobile banking, mobile health-care [107, 108]. As a result, personal and sensitive data are increasingly being accessed via or stored on smartphones.

Often times, it is the data on the smartphone, rather than the device itself, is more valued by the users. If a smartphone is lost, the user may acquire a new smartphone with similar or even better features; but, the data that had gathered over time may be irrecoverable, or at the hand of wrong persons can bring great harm. The incident of a lost phone is not rare: everyday around $7 million worth of phones are lost worldwide [109]. Given, an average consumer claims that she will lose about $37000 worth of digital-assets, if a mobile device is lost or stolen and the average value of a lost item is about $176, the loss in device cost and in digital-assets cost total to more than $1.5 billion a day [111, 110]. If the cost due to the data falling in the wrong hands is even a small fraction of the total cost, that is presumably more than $7 million a day. Naturally, securing the data on smartphones is crucial more than ever.

> "Some show that nice sagacity of smell,
> And read with such discernment in the port
> And figure of the man, his secret aim
> That oft we owe our safety to a skill
> We could not teach and must despair to learn."
>
> (Cowper[112])

As Cowper so eloquently puts it, a pet, specially a dog, discerns a lot about its owner: her smell, the way she walks, her face, the way she caresses, her voice, and many other subtleties go into making the mental model that the pet has of its owner.

A pet recognizes its owner from a stranger using this mental model; for a stranger, the data a pet receives through its senses do not quite fit the model; and this act of recognizing happens without the owner being conscious about it. We propose to utilize this observation—about how a pet recognizes its owner from a stranger without the owner being conscious about it—to come up with an application-level end-user authentication framework for smartphones that, beside being effective, will cause minimal stress and distraction to the user, encouraging her to use authentication applications built around our framework for her phone's security. The pet metaphor works well for smartphones, because of three reasons: one, smartphones stay near its owner everyday for a substantial period of time; second, smartphones have various sensors to sense the surroundings along with its owner's traits, and third, the owner gets attached to her phone over time. The authentication framework proposed in this paper, shares many similarities with the multimodal biometrics approach; but, with a crucial difference: our proposed framework is adaptive; it tries to find an optimal trade-off between security, usability, and frugality by finding a mix of biometric features that will provide enough confidence using minimal resource (e.g., battery power) and user involvement. We also believe, the pet metaphor is suggestive enough for the user to quickly grasp the basic idea of how the authentication system works, relieving him of the likely uneasiness associated with the invisibility of the authentication process.

3.1 Proposed Authentication Framework

A smartphone has a set of sensors, S; associated with each sensor s is a set of sampling rates, Σ. A trait, τ, is a distinguishing characteristic of the owner, which can be computed from the data received from a subset of S. A template, θ, is a set of complex values, which represents a trait. An authentication criterion, α can be computed from a subset of the set of traits, Υ; an authentication decision depends on authentication criteria. Formally, we can define the authentication framework (AF) as a tuple.

Definition 3.1. *$AF = (S, \Sigma, \Upsilon, \omega, F, \Gamma, \kappa, D, \Pi, \Xi, \Psi, P, \Delta)$, where,*

$$S : \textit{Set of sensors;}$$

$$\Sigma : \textit{Set of sampling rates;}$$

$$\Upsilon : \textit{Set of traits;}$$

$$\Omega : \{\omega \mid \omega : (S \times \Sigma \times \mathbb{R}) \mapsto \mathbb{R}^n\};$$

$$F : \{f \mid f : \mathbb{R}^n \mapsto \mathbb{R}^m\};$$

$$\Gamma : \{\gamma \mid \gamma : \mathbb{R}^m \mapsto \mathbb{C}^p\};$$

$$P : \{p \mid p : \Omega \times \Upsilon \mapsto \{0, 1\}\};$$

$$\kappa : (\Omega \times F \times \Sigma \times P) \mapsto \mathbb{R}^2;$$

$$D : \textit{Database of templates;}$$

$$\Pi : \{\pi \mid \pi : (\mathbb{C}^p \times D) \mapsto \mathbb{R}\};$$

$$\Xi : \textit{Set of security settings;}$$

$$\Psi : \{\psi \mid \psi : (\Xi \times \mathbb{C}^p) \mapsto \mathbb{R}\};$$

$$\Delta : \{\delta \mid \delta : (\mathbb{C}^u \times \mathbb{C}^v \times \eta) \mapsto \{0, 1\}\}$$

Here, Ω is a set of data collecting functions and $|\Omega| = |S|$; e.g., if the trait is gait and we use accelerometer as the sensor, then on the Android platform, `onSensorChanged(SesnorEvent)` would be a data collecting function [113]. For a particular sensor and for a particular sampling rate σ, over a time interval of τ, an ω collects data of size $\sigma\tau$. The size of the set of traits, $|\Upsilon| \leq 2^{|S|} - 1$. Each $f \in F$ is associated with one sensor and it preprocesses the output of the corresponding ω; e.g., an accelerometer on the Android platform does not provide readings at a precisely regular interval and the preprocessing function can interpolate the received sensor-data to have a precise frequency. Template generating functions constitute Γ; e.g., accelerometer data corresponding to one step of a person's gait might be used as the template. A person may be sitting, so the accelerometer sensor data cannot at that instant be used to determine her gait pattern; thus not all sensors, always have useful data. A low-cost probing function, $p \in P$ decides whether a

sensor can be useful in computing a particular template. How costly—in terms of battery drainage and usability—a quadruple consisting of a probing function, a data collection function, a preprocessing function, and a template generating function, is determined by κ. A database containing the templates of each trait across a varied population is represented by D. Given a template of the owner, $\pi \in \Pi$ determines how much a owner is unique among the population with respect to the corresponding trait; e.g., Equal Error Rate for gait. Security settings chosen by the owner is represented by Ξ, a totally ordered set, she may assign different levels of security to different applications on her phone [96]. For each template (representing one trait) a $\psi \in \Psi$ translates a threshold value to be used by δ based on the stored template and the user's preferred security setting. The matcher function δ, given two instances of a template and a threshold value, decides whether the two instances belong to the same person.

Claim 3.1. *The possible number of templates is upper-bounded by* $\left\lceil \frac{|\Gamma|}{|\Upsilon|} \right\rceil |\Sigma|\,|S|\,2^{|S|-1}$.

Proof. From the definition of traits, $\Upsilon = \mathcal{P}(S) \setminus \emptyset$. If $X \in \mathcal{P}(S)$ and $x \in X$, then each x leads to at least one possible template. So, there are at least $\sum_{k=1}^{|S|} k\binom{|S|}{k}$ or $|S|\,2^{|S|-1}$ possible templates. Each x corresponds to a subset of Σ and for a particular sampling rate there is at least one template generating function; thus, to get the possible number of templates, we need to multiply $|S|\,2^{|S|-1}$ by the factor $\left\lceil |\Gamma|/|\Upsilon| \right\rceil |\Sigma|$. □

The authentication framework, AF, works in three phases: a database creation phase, a training phase and an authentication phase. In the database creation phase, a subset of the possible templates for an anonymous sample population of size n is generated and that information constitutes D. For a specific smartphone model, the database creation phase is needed once. In the training phase, a subset of the possible templates of the owner are generated; assuming, as long as the training phase is not over, the phone bearer is always its owner. Once the smartphone has been trained, it is ready to periodically authenticate the owner without requiring her to be actively involved. The false positive rates and the false negative rates in AF is never zero, but it can be made low enough for a particular security setting. The templates

generated in the training phase is sorted twice with a stable sorting algorithm: first in increasing order of discriminatory capability and then in increasing order of cost; this sorted template list $\langle t_i \rangle$, is then passed on to the authentication phase. From the owner-provided security setting the authentication phase finds an index k in the sorted template list, whenceforth every template can identify the owner with at least as much confidence as the present security setting requires. Periodically, in the authentication phase, using template t_k and a $\delta \in \Delta$, it is decided if the phone-bearer is the owner. The low-cost $p \in P$ can be invoked beforehand, to decide if a higher-indexed template from $\langle t_i \rangle$ needs to be selected due to the fact that one or more sensors corresponding to t_k may not be currently useful.

Claim 3.2. *If a user has a new smartphone having one or more new sensors, then AF can adapt.*

Proof. We shall argue through induction. If the smartphone has a single sensor, it can define at most one trait of the owner. In that case, AF will always use this single trait for authentication, if it can provide the requested level of security; otherwise the user has to rely on a fall-back locking mechanism for her smartphone. Assuming the previous smartphone had k sensors and AF was working on it, for the new smartphone with an additional sensor, one of four cases can occur:

1. The sensor can form a new trait by itself.
2. A new trait can be defined with the new sensor along with some previously available sensors in it.
3. The new sensor can be absorbed into one or more previously available trait(s).
4. The new sensor can lie outside of the authentication framework.

For the first three cases, D needs to be augmented and a training phase follows gathering the owner's templates corresponding to the new trait(s); thenceforth AF is seamless once again. From the user's point of view, she is developing a rapport with her new phone. □

In Algorithm 1 and 2 we outline one possible implementation of the training and authentication phases for AF in some details, where discriminatory capability of a template is computed from the equal error rate (EER). The EER indicates the point on a Receiver Operating Characteristic (ROC) curve, where the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR); EER can be thought of a reasonable trade-off between security (FAR) and usability (FRR) [103]. Thus, between two traits, the one with a lower EER is more reliable and usable with respect to authentication.

Algorithm 1 Training Phase

1: $factor \leftarrow 0$
2: **for** each $trait$ **in** $Traits$ **do**
3: $\quad factor \leftarrow factor + trait.EER^{-1}$
4: **end for**
5: **for** each $trait$ **in** $Traits$ **do**
6: $\quad trait.normalizedScore = \frac{trait.score - m}{M - m}$
7: $\quad trait.weight = \frac{factor^{-1}}{trait.EER}$
8: **end for**
9: **for** each $subset$ **of** $Traits$ **do**
10: $\quad subset.score \leftarrow 0$
11: $\quad$ **for** each $trait$ **in** $subset$ **do**
12: $\quad\quad subset.score \leftarrow subset.score + trait.weight \times trait.normalizedScore$
13: $\quad$ **end for**
14: $\quad subset.threshold \leftarrow ComputeThresholdAtEER()$
15: $\quad subset.confidence \leftarrow 1 - FAR(threshold)$
16: **end for**

In Algorithm 1, lines 1–8 compute the weight for each trait so that for each trait the weight is inversely proportional to its error [116]. Inside the **for** loop on line 9, we compute the score for each subset of traits in lines 10–13, each trait in a particular subset contributes according to its weight so that a more reliable trait contributes more score to its container subset. On line 14 we compute the threshold score at Equal Error Rate (EER) for a subset and on line 15 we compute how much confidence to put on a particular subset of traits: the more False Acceptance Rate, a subset has at the chosen threshold, the less confidence we put on it while authenticating.

In Algorithm 2, on lines 1–7 we compute b_s, which is a bit-string having a bit corresponding to each sensor and b_s has its i-th bit set to 1 if the i-th sensor is

Algorithm 2 Authentication Phase

```
1: mask ← 1
2: for each sensor in Sensors do
3:    if sensor.Probe() = true then
4:       b_s ← b_s | mask
5:    end if
6:    mask ← mask ≪ 1
7: end for
8: for each trait in Traits do
9:    if (trait.b_s | b_s) ⊕ b_s = 0 then
10:      T_a ← T_a ∪ {trait}
11:   end if
12: end for
13: minCost ← ∞
14: MinCostSufficientSubset ← ∅
15: for each subset of T_a do
16:   if subset.confidence ≥ threshold then
17:      if subset.cost < minCost then
18:         minCost ← subset.cost
19:         MinCostSufficientSubset ← subset
20:      end if
21:   end if
22: end for
23: for each trait in MinCostSufficientSubset do
24:   data ← trait.GatherData()
25:   template ← trait.CreateTemplate(data)
26:   isOwner ← Match(template, t)
27:   if isOwner = false then
28:      Lock()
29:      return
30:   end if
31: end for
```

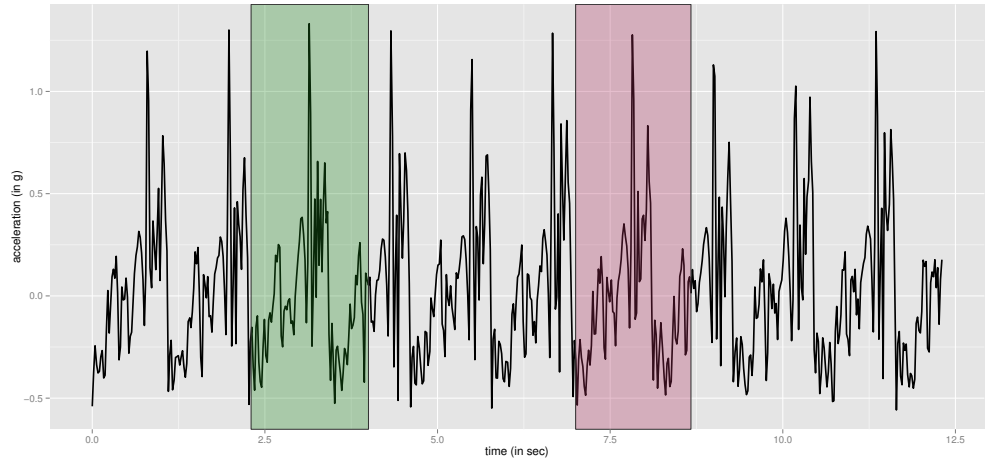


Figure 3-1: Periodicity in gait data

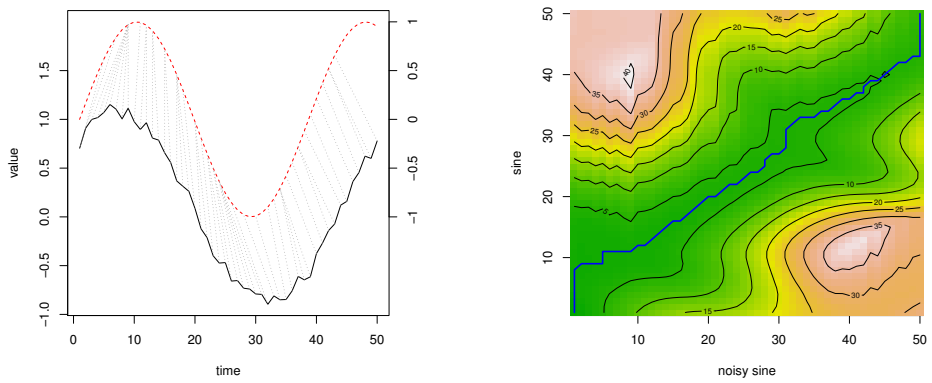
currently active. On lines 8–12, we determine which traits have all their sensors active right now; and these active traits make up the set T_a. Next, on lines 14–22, we compute a minimal cost subset of T_a that is able to provide enough confidence about the authenticity of the user. The remaining lines of Algorithm 2, authenticates the user using the traits in the chosen minimal cost subset.

3.2 A Subset of the Framework

We explore a subset of our proposed framework, which utilizes gait and location traces of the owner: consecutively from accelerometer and GPS data. The basic idea is that whenever the phone finds itself in a new location it increases the threshold for acceptance mimicking the heightened alertness of a pet in an unknown environment; the phone acts defensively by increasing False Rejection Rate (FRR) to decrease False Acceptance Rate (FAR). Below we describe the details of how to detect gait pattern, gather location trace, and authenticate using gait pattern in the presence of location cues.

3.2.1 Recognizing Walking Pattern

Walking is a periodic activity and the periodicity is reflected in accelerometer data of a smartphone [105]. The repeating patterns in the accelerometer data of a smartphone are similar for a particular person, while they differ across different individu-



(a) Nonlinear warping

(b) Terrain of matching cost

Figure 3-2: DTW Matching of two sine waves

als; hence, gait pattern can serve in identifying the owner. A step is the natural unit for gait pattern; but, detecting the portion of accelerometer data corresponding to a single step is a challenging task. Instead, we consider a portion of accelerometer data, which contains at least one step and by periodicity this portion also repeats across the entire gait data. In figure 3-1, the highlighted portion in green has been arbitrarily chosen, yet we see it repeated in the maroon region. We use this semi-arbitrary segment of data as the template and Dynamic Time Warping (DTW) algorithm works as the matching algorithm. A person may walk at different speeds and thus her gait pattern can be stretched or contracted along the time axis. The motivation for using DTW is that while matching, DTW warps two time series in a nonlinear fashion in order to cope with time deformations and different speeds.

Figure 3-2(a) shows exemplary nonlinear mapping between two sine waves of slightly different frequencies and one of them is noisy. For two time-series X and Y of length M and N, there is an $M \times N$ cost matrix, where cost between a pair of points is defined by a cost metric; while matching gait segments, we use Euclidean distance as the cost metric in DTW. In figure 3-2(b), a contour plot of such a cost matrix is shown; using dynamic programming, DTW efficiently finds an optimal alignment (shown as a blue line) between X and Y that runs along a low-cost valley of the cost matrix. Using the training data obtained from the users, we compute

FAR, FRR, and Equal Error Rate (EER) for gait pattern. Following Algorithm 1, we compute the threshold value at EER, which is used to make decision during authentication.

Table 3.1: Smartphone-based gait recognition

	Dev.	Pos.	Detect Step?	Features	Match. algo.	# users	Best EER	Rec. rate	Bat. drai.
[125]	Android Phone	N/A	No	Avg. accl., σ, Avg. abs. diff., Resultant Time bet. peaks Binned dist.	WEKA: J48, NN	36	N/A	82.2%	N/A
[126]	Google G1	Hip	Yes	Min DTW cycle	DTW	51	20%	N/A	N/A
[127]	Moto Droid	Pocket	No	Wavelet Coeff.	kNN	2	N/A	90%	N/A
[128]	Moto Milestone	Hip	No	Diff., Bin, RMS, Cross-corr., MFCC, BFCC	HMM, SVM, kNN	36	8.24% (kNN)	N/A	N/A
[124]	Android Phone	Pocket	No	Wavelet Trans. Spect.	SVM	36	N/A	99.4%	N/A
Ours	**Google Nexus S**	**Pocket**	**No**	**Segment**	**DTW**	**13**	**10%**	**N/A**	**10%**

In Fig. 4, for all users, correlation between DTW similarity score and weight, height, and weight-to-height ratio has been shown. None of the variables: weight, height, and weight-to-height ratio consistently had a significant correlation with DTW similarity scores. As a result, for an attacker, only having similar weight, height, or weight-to-height ratio does not guarantee success mimicking a user's gait; unless the attacker puts effort in studying a user's walking pattern [121].

3.2.2 Building Location Trace

At a new place, a pet might initially remain more alert, which helps it to explore the place, while keeping the risk low. Likewise, when the authentication system (AS) finds itself in a place that is absent in the list of its familiar places, it increases the gait matching threshold score to emulate hightened alertness. Location is not exactly an authentication criterion; AS does not use location cue to decide whether the phone-bearer is the owner; instead it helps regulate the level of confidence to

put in other authentication criteria, like: gait pattern.

During the training phase, each day AS periodically collects GPS coordinates and at the end of each day the owner inputs how many places she visited on that day. In order to cluster the GPS coordinates, AS uses k-means algorithm where the owner provides k [122]. After a week of refining the clusters corresponding to the visited places, AS saves the list of cluster-centers along with a radius for each cluster. During the authentication phase, AS periodically collects GPS coordinates and checks whether the place belongs to one of the saved clusters; if it does not find the place in its list, it increases the matching threshold, otherwise the location does not affect the threshold.

3.2.3 Authentication Using Gait Pattern and Location Cues

The authentication system (AS) periodically tries to authenticate the user using her gait pattern at a predefined threshold. It periodically also determines if the current place is a familiar one, i.e., already visited by the user. In case AS finds itself at a familiar place, nothing changes; it continues authenticating the user using her gait pattern. But, if AS finds itself at a new place, it increases the gait matching threshold; so that, it only let the phone bearer in, only when it is confident enough that she actually is the owner. In this situation, if the phone rejects the user incorrectly, the user may choose the fall-back authentication method to login and then train AS about the new place. We expect this to happen rarely for the owner after the initial training period; because, by that time the places she usually visits should be familiar to AS.

3.3 Experimental Procedure

In this section, we describe how we have collected gait and location data using a Google Nexus S smartphone that has a built-in 3-axis accelerometer and GPS receiver module. As our experiments involve human subjects, we obtained approval from the Institutional Review Board (IRB) of Marquette university. We also provide evaluation of our proposed algorithms using the collected data.

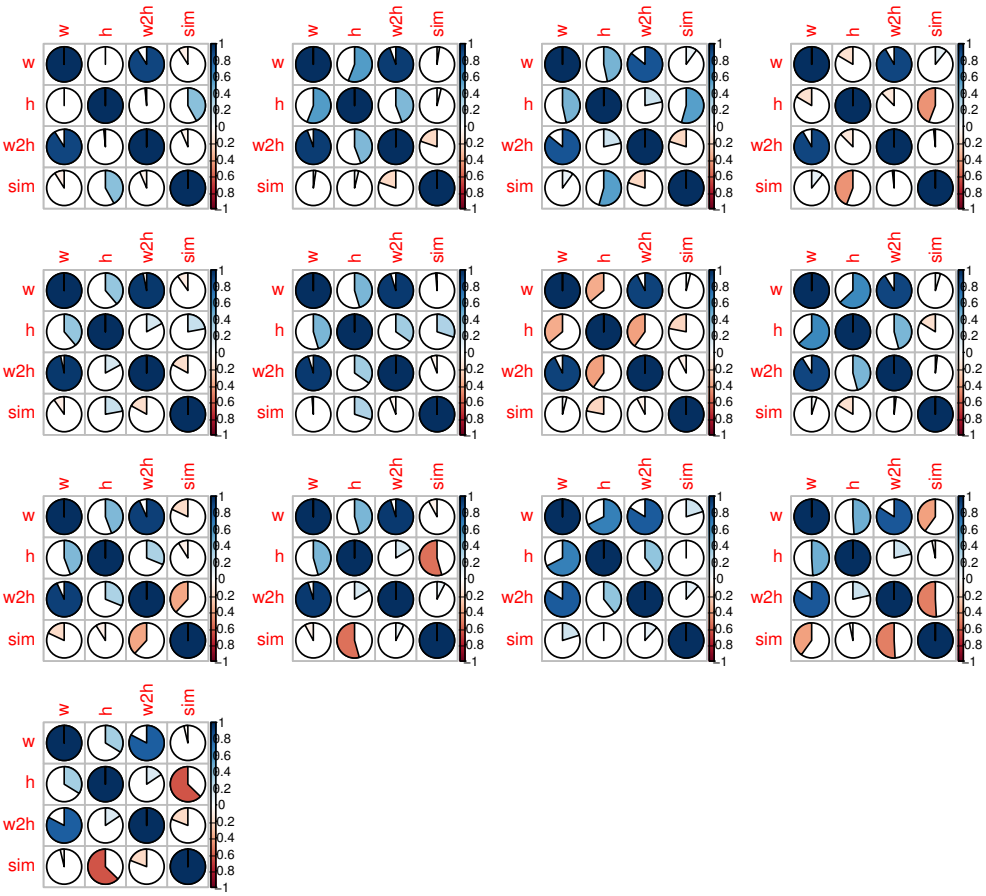


Figure 3-3: Correlation between DTW score and weight, height, or weight-to-height ratio

3.3.1 Pattern of Walking

We have collected walking data from 13 users whose age-range is between 18 and 34 years having a mean of 26 years and a standard deviation of 4.14 years; all of them are members of Ubicomp lab at Marquette university; every participant owned a smartphone and was familiar with using smartphone applications. For each user, we have collected walking data in two sessions separated by one week. Each user walked on a level surface, indoors, and at a normal speed with the phone in their right pant-pocket; they walked about 29 meters in each session.

We divided the walking data for each user and for each session into 5 chunks; thus for each user we have 4 chunks of data to compute genuine scores and 60 chunks of data to compute impostor scores—per session. Before computing DTW scores,

we standardized the data by subtracting the mean from it. While calculating FAR and FRR, in order to match weights of genuine and impostor scores, we linearly interpolate the genuine distribution. Table 3.2 shows EER's of the 13 users along with the threshold matching scores; and it is evident that gait pattern of a user has limited discriminative capacity. For a user whose EER is high, Algorithm 1 frugally incorporates other biometrics in the set of authentication criteria to increase confidence; whereas, for a user whose EER is low, considering only gait pattern may be sufficient; in this case Algorithm 1 chooses the singleton containing the gait pattern for authentication—in case that is the most frugal choice. In Table 3.1 we provide a comparison between other gait-based authentication for smartphones with our implementation.

Table 3.2: EER (%) at chosen thresholds

First day	After a week
16	14
14	14
20	20
14	16
16	14
18	14
14	18
14	14
16	18
16	10
14	14
16	16
16	14

3.3.2 Trace of Visited Places

One of the authors used the authentication system (AS) on a smartphone, to collect GPS coordinates for two weeks. From the coordinates of the first week, using k-means algorithm, we build a list of places. The number of clusters we used for the k-means is 5, which was determined by the number of distinct places the phone-bearer visited during that time. So the list of familiar places after the first week includes 5 pairs, each pair containing the coordinates of the cluster center and an associated radius. Using the GPS coordinates collected during the second week,

we evaluate how accurately AS can recognize a new place. The accuracy was 98%, which means that only after a week of location training, AS was able to recongnize a familiar place with good accuracy. It may be due to the sedentary life style of the author who carried AS, that after only a week the number of new places AS finds is so small; but we expect that most of the places a user regularly visits will be covered within a few weeks.

3.3.3 Battery Charge

In Fig. 3-4, we see that gait data was collected over 5 seconds every 5 minutes and thus a gait matching occurs every 5 minutes. If the gait template size is n and the collected accelerometer data has size m, then DTW matching occurs $m - n + 1$ times every 5 minutes. In DTW matching, as we used a window size of 12 and as we need only the matching score, we needed to maintain only about $2m$ floating point values. There are atmost $12n(m - n + 1)$ computations for the gait matching every 5 minutes. In our experiment n was about 120 and m was twice the size of n; thus overall complexity of gait matching, in our case, was $\Theta(n^2)$.

Figure 3-4: Timing of location and gait computations

In Fig. 3-5, the battery charge consumption over time is shown. We see that doing gait computations periodically does not drain battery much, it is similar to the base rate of battery charge consumption. However periodically obtaining GPS data and checking whether the current place is familiar, drain battery charge significantly.

3.3.4 Ideal Authentication System for Smartphones

When the study ended, each subject was asked what would be the ideal authentication system for smartphones. From Table 3.3 we see that an ideal authentication

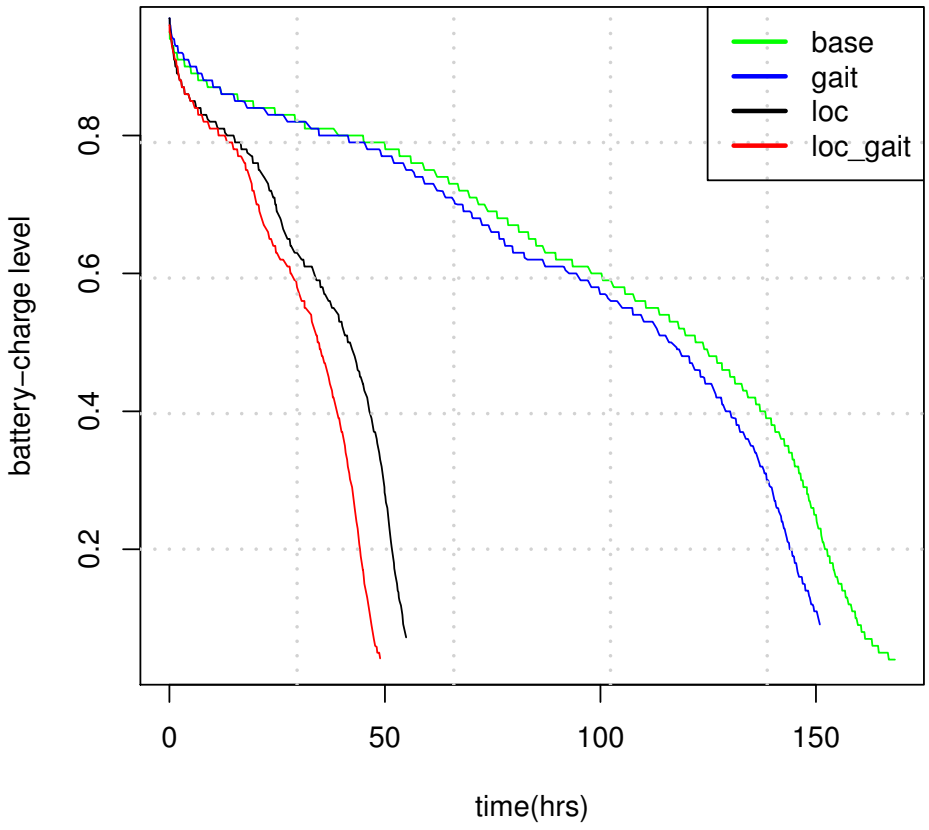


Figure 3-5: Battery charge over time

system, according to users' opinions, should not require one to remember secrets, should not take long to perform, should be secure, and best if it is fully transparent. Interestingly, ePet has all the properties the users wanted from an ideal authentication system for smartphones. And while we explained the idea behind ePet to the users, it was quite easy to make them understand using the pet metaphor.

3.4 Potential Issues

Here we briefly discuss, some of the issues that need to be addressed to make continuous and transparent authentication like ours, practical.

Table 3.3: Ideal authentication for smartphones

User	Reply
1	"Fingerprint biometrics"
2	"Multistage: biometrics and password"
3	"It should: enhance security, enable faster input, and require least user-involvement"
4	"Something that could recognize me. Therefore, not needed that I provide some authentication."
5	"Should be: strong, relatively easy to remember, and quick"
6	"Different layers of authentication"
7	"Voice-based password"
8	"Biometrics"
9	"I don't need to remember anything, it can be long like within 10 seconds, but should be strong, unbreakable"
10	N/A
11	"Voice, face, and touch authentication"
12	"A combination of voice, image, and other features if available."
13	"Didn't think much of it. But I think voice and fingerprint combination will work better."

3.4.1 Battery Charge

Our framework requires collection of sensor data periodically, which can drain battery charge quickly. Fortunately, "responsive sleeping" is a promising idea that has the potential to significantly reduce battery drainage due to continuous sensing on smartphones using heterogeneous multi-processors [129]. Apple M7 coprocessor has already been used to offload the collecting and processing of sensor data from the main central processing unit (CPU) on iPhone 5S [117]. For the GPS sensing, optimal scheduling strategies exist, that can significantly extend battery life without hurting the location accuracy much [130, 118].

3.4.2 On-body Placement

As all traits are computed from sensor data, it is important for the algorithms for computing traits be aware of where on the user's body the phone is presently kept; for example, the gait pattern may change if a the phone is held in hand instead of keeping it in the pocket. Fujinami et al. [123] looked into five different phone-storing positions: front pocket of trouser, back pocket of trouser, side pocket of jacket, chest pocket, and hanging around the neck; with 10 users, 5-minutes of walking data collected using a Google NexusOne phone for each storing position per user, and 29

features—they were able to detect the storing position with 72.3% accuracy using artificial neural network. Whether the accuracy they obtained is adequate for the purpose of authentication needs to be explored.

3.4.3 Psychology

Users may feel uncomfortable with the idea of invisible and continuous authentication using their personal data, like: gait pattern, location trace, etc. [119] In our framework, we propose to store the trait data locally on the phone in a secure storage like: Secure Enclave [120]; and all the computations are done on the phone. The pet metaphor, we believe, will help users to perceive that their personal data is not being shared with third-parties.

3.4.4 Pace

Gait pattern may vary depending on the pace of a user's movement. In [124] normal and fast walking were investigated with Android phones. With accelerometer and gyroscope data they explored three methods: a linear support vector machine (SVM), continuous wavelet transform time frequency spectrogram analysis, and cyclostationarity analysis; but between normal and fast speed the best verification rate was 61% at 0.1% FAR. Making gait matching contingent upon pace may increase accuracy while comparing between different speeds; e.g., storing owner's templates corresponding to different speeds of walking and then for intermediate speeds a template might be generated through interpolation, in keeping with the spirit of [114].

3.5 Discussion

We have proposed an authentication framework for smartphones that require minimal user involvement and that is thrifty about battery power. The framework mimics how a pet might recognize its owner from a stranger on its own. Continuous sensing and computing, is a major issue in our framework, which can drain a phone's battery quickly. We have shown that not all types of sensing take a large toll on battery charge, e.g., gait matching has moderate cost whereas GPS can drain battery quickly. We have indicated how these limitations can be mitigated. The pet

metaphor can help the user perceive that her personal data remains secured on the phone. Adding more traits to our current implementation is an obvious future goal. The estimate of the number of possible templates suggested in Claim 1, indicates that the detailed implementation of the framework may require a large one-time effort in building the database D. The sample size with which we have conducted the experiments is small and probably biased; we hope to perform the experiments with larger and more diverse samples in the future. Determining how a user might easily and effectively set the security level she wants, is an important future work. In Algorithms 1 and 2, we have not incorporated the usability cost while choosing the minimal cost subset for authentication; ranking the traits according to usability and then defining a generalized cost metric would also need to be considered in the future. Location is not a true trait, but a context; how to incorporate contexts in our proposed framework is another interesting future task. Exploring the effects of adding an avatar of a pet on the phone, that shows positive expressions when it recognizes its owner and shows signs of distress when the cost for authentication is high may be another interesting future pursuit; e.g., the user might be motivated towards training the authentication system more.

Chapter 4

A Closer Look into the Password Input Process

In this chapter we shall look into the password input process more closely with a goal to justifying that for a successful login (with an old password,) the declarative memory for the password plays the principal role; the role of the motor memory is to increase the efficiency of the input process—by increasing the typing speed and by decreasing the number of typing errors. In other words, for an old password, one can successfully login even if her motor memory is of little help, whereas significant disruption to the declarative memory has an incapacitating effect on the login success. First, we shall take a brief tour to the organization of human memory.

4.1 Human Memory

Human memory is an alliance of mainly three systems: sensory registers, short-term store, and long-term store [131]. Information can flow into the sensory registers even if a person is not paying attention to it; whereas, for the information to flow into the short-term store, attention is required; and for the long-term store one needs to rehearse. It is not possible to maintain information in the sensory registers for long—they are ever effervescent; whereas, it is possible to maintain information in the short-term store, but one needs to rehearse and pay attention continuously; and only after repetitions and organization, the information will pass onto the long-term store [132]. In Figure 4-1, we show the interrelations between various aspects of human memory [131, 135]. Long-term memory (resides in long-term store) is of two types: declarative memory and procedural memory. Declarative memory can be consciously recalled, e.g., facts and knowledge and it is of two types: episodic memory and semantic memory. Episodic memory refers to the memory of the events related to a person's own life, e.g., remembering going to a dentist a week ago. In contrast, semantic memory refers to memories about general factual knowledge—independent of personal experience, e.g. verbal memory [131]. By the declarative memory for a password, we refer to the semantic memory for the password. The other type of long-

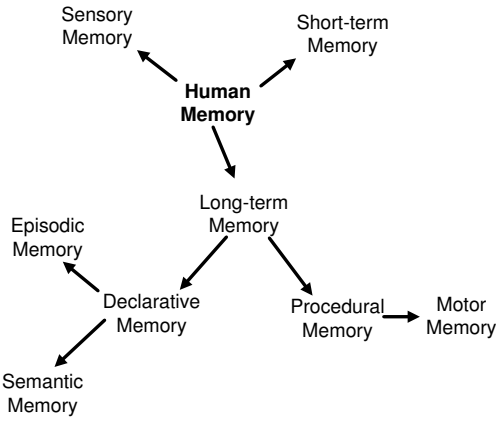


Figure 4-1: Organization of human memory

term memory, namely procedural memory, refers to the memory for the performance of particular types of actions, which one is usually unable to recall consciously; rather, when needed this kind of memory is automatically retrieved for utilization. Motor memory is a form of procedural memory, where through repetitions of a particular task, a long-term memory develops; this long-term memory is implicitly expressed through smoother movements and a reliance of the motor output on an internal model that anticipates the force requirements of the task, which is then reflected in the muscle activations [133, 134].

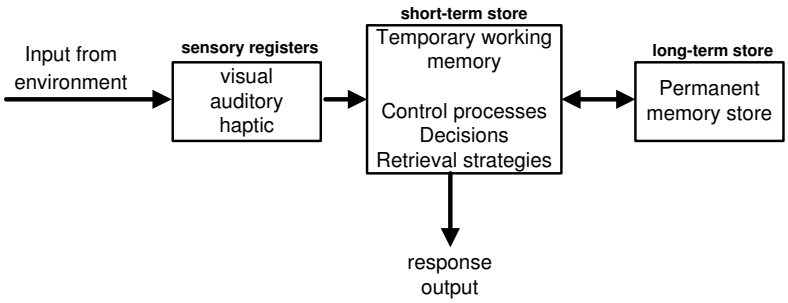


Figure 4-2: Information flow in human memory

In Figure 4-2 the Atkinson-Shiffrin model of information flow in human memory (as described in [131]) has been shown; we see that the short-term store (STS) works as a bridge between the long-term store (LTS) and the environment—before one can output an item from her long-term memory, she has to load that item into her short-term store first.

4.2 Conceptual Model for Password Input

Figure 4-3 shows the steps involved in inputting a password. In the first step, a password-symbol is fetched from the Long Term Store (LTS) to the Short Term Store (STS) (Figure 4-2). Then, in the second step, the decision is made about which finger to move and to which (approximate) key-coordinates; then, in the final step the finger is moved to the key and the key gets pressed. We believe that the declarative memory of a password is involved in the first step and the motor memory for a password, if any, comes into play at the second step. Our idea is to selectively inhibit the first and the second steps and to observe how the time for typing in a password gets affected. Before discussing the study, we elaborate on the "look up coordinates" part of the second step (② in Figure 4-3) of the password input process.

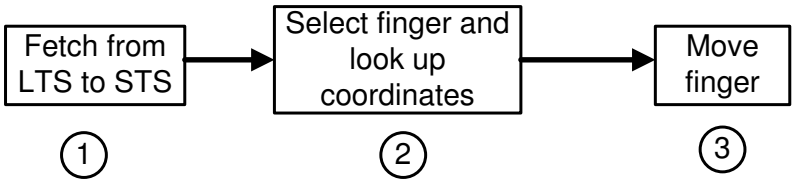


Figure 4-3: Password input process

4.2.1 Look-up Coordinates

If the layout of a keyboard is familiar to the user, over time a look-up table gets created; whenever a symbol is fetched to the STS, a table look-up gives the approximate coordinates of the position to which a finger should be moved to press the key corresponding to the fetched symbol. However, if a password is old and the layout of the keyboard is familiar, therefore the user has typed in the password many times on the keyboard, the look-up table for coordinates get modified. After practice, a password contains chunks of symbols instead of individual symbols and a look-up occurs only for the first symbol of a chunk; because, the coordinates of the rest of the symbols—belonging to the chunk—have already been saved in the look-up table along with the coordinates of the first symbol of a chunk. In our opinion, the time saved due to the reduction in the number of look-ups is what we see as the motor

memory for a password. On the other hand, if the keyboard layout is unfamiliar, the look-up table of coordinates does not help and the user must fall back to visual search for the key.

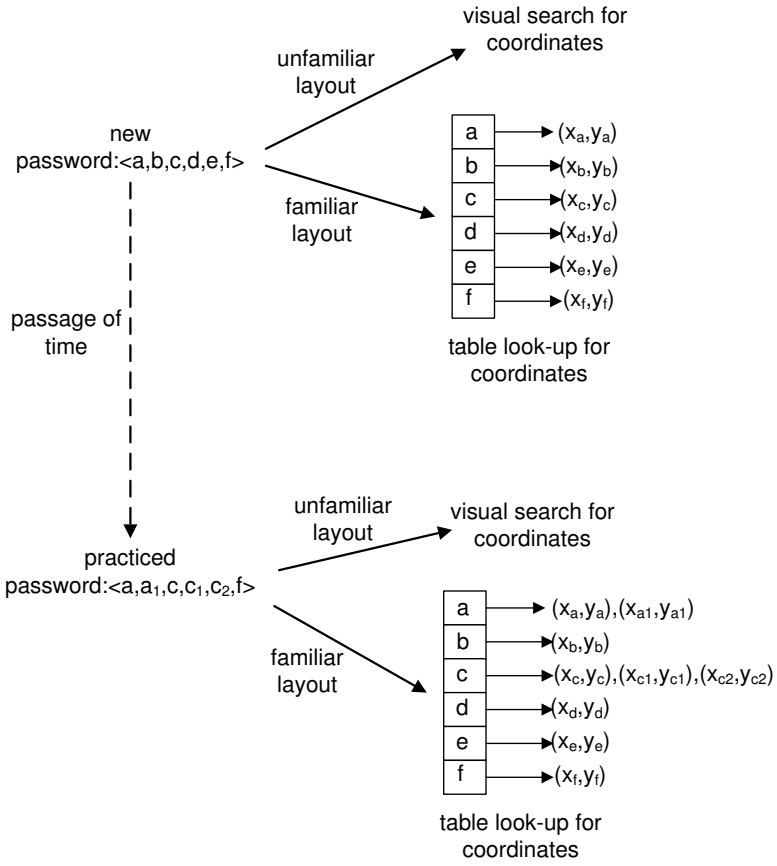


Figure 4-4: Example table look-up

For example, with reference to Figure 4-4, let $\langle \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f} \rangle$ be a new password created by a user. In order to login using the password $\langle \mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f} \rangle$, on a keyboard with an unfamiliar layout, the user has to visually search for each of the six symbols (unless there happens to be a chunk from some old information.) But if the keyboard layout is familiar, there is already a look-up table of key-coordinates, which maps each symbol to the approximate coordinates for the corresponding key on the keyboard; so, now she does not have to visually search the correct key, but to look it's coordinates up in the table. Consequently, there will be six look-ups and near-zero visual search for a new password on a keyboard with a familiar layout. However, with the passage of time, when the user has had practice with the password, it might

look like: $\langle \mathsf{a}, \mathsf{a}_1, \mathsf{c}, \mathsf{c}_1, \mathsf{c}_2, \mathsf{f} \rangle$, where a, a_1 belong to the same chunk; $\mathsf{c}, \mathsf{c}_1, \mathsf{c}_2$ belong to another chunk; and f is a chunk by itself. Note that, content-wise, $\mathsf{a}_1 = \mathsf{b}$, $\mathsf{c}_1 = \mathsf{d}$, and $\mathsf{c}_2 = \mathsf{e}$; the "chunks" are meaningful only in the context of how the password-symbols are accessed. Then, if the user has to type the practiced password on a keyboard with a familiar layout, only three look-ups are required: for a, c, and f respectively. The look-up for a will give coordinates for the keys corresponding to both a and a_1; the look-up for c will give coordinates for the keys corresponding to c, c_1, and c_2; and f will have its personal look-up. The time saved from not having to look up for a_2, c_1, and c_2, in our opinion, makes up the increased efficiency due to the motor memory.

4.2.2 Erroneous Finger Movements

The time it takes to input a password on a keyboard with an unfamiliar layout may further increase due to the occasional erroneous movements of the fingers—when the user momentarily forgets that the keyboard has an unfamiliar layout. At those times, the fingers move to the coordinates returned from the automatic look-ups in the table of coordinates, only to find out that the key-coordinates are incorrect [141]. The more a user is used to a particular keyboard-layout, the more often this type of errors should occur with a keyboard with an unfamiliar layout. If a password has motor memory associated with it, the probability of erroneous finger movements increases; because, a habitual routine look-up in the table of coordinates now returns not just one pair of coordinates, but all the pairs of coordinates belonging to a chunk.

4.3 Study Procedure

As the study involves human subjects, we acquired approval from the Institutional Review Board (IRB) of Marquette University. Seven subjects (2 females, 5 males), in the age-range of 23–35 years, participated in the study. All subjects were graduate students in the Computational Sciences program and had familiarity with password input process using a keyboard. In order to find out the relative importance of the roles played by the declarative memory and the motor memory for a password in performing a successful login, we recorded the time it takes for a user to login with

an old password and a new password in four different conditions, which accounts for eight conditions in total. Below we give the details of the conditions.

4.3.1 Old password, Familiar Keyboard-Layout

The subject was asked to login to the university email account using a `QWERTY` keyboard. The subject typed in her user-name first; after that, using an `ACCUSPLIT Pro Survivor-A601X` stopwatch, the experimenter recorded the time it took for the user to correctly type in the password; the stopwatch had a precision of 0.01 seconds [143]. The user started typing in the password when the experimenter uttered "start", and the stopwatch was stopped as soon as the experimenter perceived that the login screen had changed to the "starting" screen—the recorded time includes the time spent in failed logins.

4.3.2 Old Password, Familiar Keyboard-Layout, with Count-Down

The subject was asked to login to the university email account using a `QWERTY` keyboard. The subject typed in her user-name first; after that, using an `ACCUSPLIT Pro Survivor-A601X` stopwatch, the experimenter recorded the time it took for the user to correctly type in the password; the stopwatch had a precision of 0.01 seconds [143]. The user started typing in the password when the experimenter uttered "start", and the stopwatch was stopped as soon as the experimenter perceived that the login screen had changed to the "starting" screen—the recorded time includes the time spent in failed logins. While logging in, the user counted down by 2 starting at 97; if the user reached 1 and still was not able to log in, the count-down resumed from 97.

4.3.3 Old Password, Unfamiliar Keyboard-Layout

The subject was asked to login into the university email account using an alphabetic keyboard, specifically the `A2Z Alphabetic & Standard Layout Combination Keyboard`; the keyboard was switched to the alphabetic mode before the experiment. The subject typed in their user-name first; after that, using an `ACCUSPLIT Pro Survivor-A601X` stopwatch, the experimenter recorded the time it took for the user to correctly type

in the password; the stopwatch had a precision of 0.01 seconds [143]. The user started typing in the password when the experimenter uttered "start", and the stopwatch was stopped as soon as the experimenter perceived that the login screen had changed to the "starting" screen—the recorded time includes the time spent in failed logins.

4.3.4 Old Password, Unfamiliar Keyboard-Layout, with Count-Down

The subject was asked to login to the university email account using an alphabetic keyboard, specifically using the `A2Z Alphabetic & Standard Layout Combination Keyboard`; the keyboard was switched to the alphabetic mode before the experiment. The subject typed in her user-name first; after that, using an `ACCUSPLIT Pro Survivor-A601X` stopwatch, the experimenter recorded the time it took for the user to correctly type in the password; the stopwatch had a precision of 0.01 seconds [143]. The user started typing in the password when the experimenter uttered "start", and the stopwatch was stopped as soon as the experimenter perceived that the login screen had changed to the "starting" screen—the recorded time includes the time spent in failed logins. While logging in, the user counted down by 2 starting at 97; if the user reached 1 and still was not able to log in, the count-down resumed from 97.

4.3.5 Four Conditions with a New Password

The user was given a user-name and a password for a test website. The user-name was the user's first name in lower-case letters. The password was 10-characters long, was pronounceable with three syllables in it; and it contained one capital letter, one special symbol, and one digit in it; the other seven characters were lower-case letters. The digit and the special symbol in the password were "l33t" substitutions [142]. The user was given time to memorize the user-name and the password; when she felt confident that she had memorized the user-name and the password, she was given the following distractor task.

In the distractor task, the user was given a stack of five cards; on each card—from the second to the forth—there was a random sequence of 4 digits (each digit from 0–8); the purpose of the topmost card was to hide the random sequence on

the first card. When the experimenter uttered "start", the user removed the top card, looked at the first 4-digit random sequence, read it out loud, and looked away from the card. After waiting for two ticks from a metronome, which was set to tick every second, the user spelled out the random sequence with each digit increased by 1 while still looking away from the card. For example, if the 4-digit sequence on a card was "0571", then the user read it out loud and looked away from the card. After waiting for two ticks from the metronome, she spelled out "1682" while looking away from the card. The same process was repeated for the remaining three cards [145]. We used a software metronome called "Mobile Metronome" on a `Samsung Galaxy S4` smartphone [144]. When the distractor task was over, the user was asked to recall the password she was given before the distractor task; if she failed to recall the password, she was given the password again for memorizing and the process commenced again. The distractor task acted as a "rehearsal-prevention" task—to ensure that the user was not rehearsing the password under her breath [146]. At the end of the distractor task (lasting about 1 minute), if the user successfully recalled the password, it was assumed that she was able to register the password to her long-term memory; for, memory tested after one or two minutes behaves similarly to memory tested after one or two days, or even years [131].

Once the user had memorized the user-name and the password provided by the experimenter, she was asked to login to the test website in four conditions similar to those with her old password: with `QWERTY` keyboard, with `QWERTY` keyboard while counting down, with alphabetic keyboard, and finally with alphabetic keyboard while counting down. The eight study conditions are described in Table 4.1. In order

Table 4.1: The eight conditions, in which a user had to login

Index	Description
1	Old password, QWERTY keyboard
2	Old password, QWERTY keyboard while counting down
3	Old password, alphabetic keyboard
4	Old password, alphabetic keyboard while counting down
5	New password, QWERTY keyboard
6	New password, QWERTY keyboard while counting down
7	New password, alphabetic keyboard
8	New password, alphabetic keyboard while counting down

to keep the "learning effect" across the eight conditions minimized, we randomly shuffled the order of the conditions for each subject [147]. The order in which the subjects performed the conditions are shown in table 4.2.

Table 4.2: The order in which the eight conditions were performed

User	Order of Conditions
1	$\langle 1, 2, 3, 4, 5, 6, 7, 8 \rangle$
2	$\langle 6, 1, 8, 2, 7, 4, 3, 5 \rangle$
3	$\langle 7, 1, 5, 6, 3, 2, 4, 8 \rangle$
4	$\langle 8, 1, 6, 5, 3, 4, 7, 2 \rangle$
5	$\langle 2, 7, 1, 5, 3, 6, 8, 4 \rangle$
6	$\langle 5, 4, 2, 1, 8, 3, 6, 7 \rangle$
7	$\langle 1, 3, 6, 7, 8, 2, 5, 4 \rangle$

4.3.6 Time Spent in Password Input

During the password-input process, with reference to Figure 4-3, steps ② and ③ take the larger portion of time, while step ① is relatively fast. So, a disruption in step ② should increase the total time of the input process more than a disruption in step ①; nevertheless, as step ① must be completed in order to initiate step ②, according to the subjective experience of a user, a disruption in step ① should pose more difficulty. Table 4.3 shows the various time-consuming tasks—according to our

Table 4.3: Tasks in Eight Conditions

Index	Relevant Tasks that Consume Time
1	Fetch from LTS, look-up (chunked coordinates), and move finger
2	Fetch from LTS (↑↑), look-up (chunked coordinates), and move finger
3	Fetch from LTS, look-up (search visually)(↑↑), and move finger (correct erroneous finger movements)(↑↑)
4	Fetch from LTS (↑↑), look-up (search visually) (↑↑), and move finger (correct erroneous finger movements)(↑↑)
5	Fetch from LTS(↑), look-up (individual coordinates)(↑), and move finger
6	Fetch from LTS (↑↑↑), look-up (individual coordinates)(↑), and move finger
7	Fetch from LTS(↑), look-up (search visually)(↑↑), and move finger (correct erroneous finger movements)(↑)
8	Fetch from LTS (↑↑↑), look-up (search visually)(↑↑), and move finger(correct erroneous finger movements)(↑)

proposed model—that are involved in inputting a password in the eight conditions described in Table 4.1. There are three types of tasks: fetch from LTS, look-up coordinates, and move finger. Each of these three tasks has zero, one, or more upward arrows associated with it—indicating an increase in time from the baseline. For example, the baseline-time for the task "fetch from LTS," is in condition 1, while in condition 5 this task has one upward arrow associated with it—indicating that fetching is slower for a new password than it is for an old password. Table 4.4

Table 4.4: Comparing time spent in a pair of conditions

Condition Index → ↓	1	2	3	4	5	6	7	8
1	=	<	<	<	<	<	<	<
2		=	<	<	=?	<	<	<
3			=	<	>	<?	=?	<
4				=	>	>?	>	<
5					=	<	<	<
6						=	>?	<
7							=	<
8								=

shows what we should expect, according to Table 4.4, if we compare the time spent between a pair of conditions. A cell (x, y) has a "=" if the time spent in condition x is approximately equal to the time spent in condition y; it has a "<", if the time spent in condition x is less than the time spent in condition y; it has a ">" if the time spent in condition x is greater than the time spent in condition y; relation for $<$ holds; and it has a "<?", ">?", or "=?" if the relation is not apparent from our model. The reasons, why we should expect the relations shown in table 4.4 are given in table 4.5; we do not provide reasons for the pair of conditions where the relation is evident, e.g., any pair containing condition 1, because the time for condition 1 is the baseline for all other conditions. We note that if a cell (x, y) has a ">" in it, then the cell (y, x) has a "<" in it and vice versa; sp, in Table 4.4, only the diagonal and the upper triangular part is shown.

Table 4.5: Reasons for the relations between median times in a pair of conditions

Cell	Prediction	Reason
(2,3)	$<$	As fetch from LTS is robust for an old password, visual search dominates
(2,5)	$=?$	Fetch has higher time in 2, but saves time from chunked coordinates
(3,6)	$<?$	Highly disrupted fetch for a new password is more time-consuming than a visual search
(3,7)	$=?$	Fetch takes more time for a new password, but erroneous finger movements are higher for an old password
(4,6)	$>?$	Disruption in fetch for an old password along with a visual search dominate a highly disrupted fetch for a new password
(6,7)	$>?$	Highly disrupted fetch for a new password is more time-consuming than a visual search

Table 4.6: Information about old password and typing speed

User	Length	Approx. Number of Times Typed	Typing Speed (wpm)
1	14	900	32
2	19	480	70
3	10	72	48
4	14	40	53
5	13	1800	20
6	10	120	49
7	10	900	43

4.4 Results

Table 4.6 shows the length of the password (old password) that a user used for Marquette University Email account and her typing speed. Users took the typing test [148] three times and the mean adjusted speed in word per minute (wpm) is shown in the table. The recorded time for a condition includes the reaction time (≈ 0.15 seconds) of a subject to start typing after she hears the experimenter uttering "start" and the reaction time (≈ 0.19 seconds) of the experimenter when he stops the stop-watch as soon as he sees the "starting" screen, we subtracted a total of 0.34 seconds from the total time for each condition [149]. The lengths of the old password and the password a user was given for the test website were not equal in length; so, in order to compare between different conditions we normalized the total time spent typing a password by its length. From figure 4-5, it is clear that for the old password the largest time spent was in condition 4 and the least time spent was in condition 1; similarly for the new password (for the test site) condition 5 was the quickest and

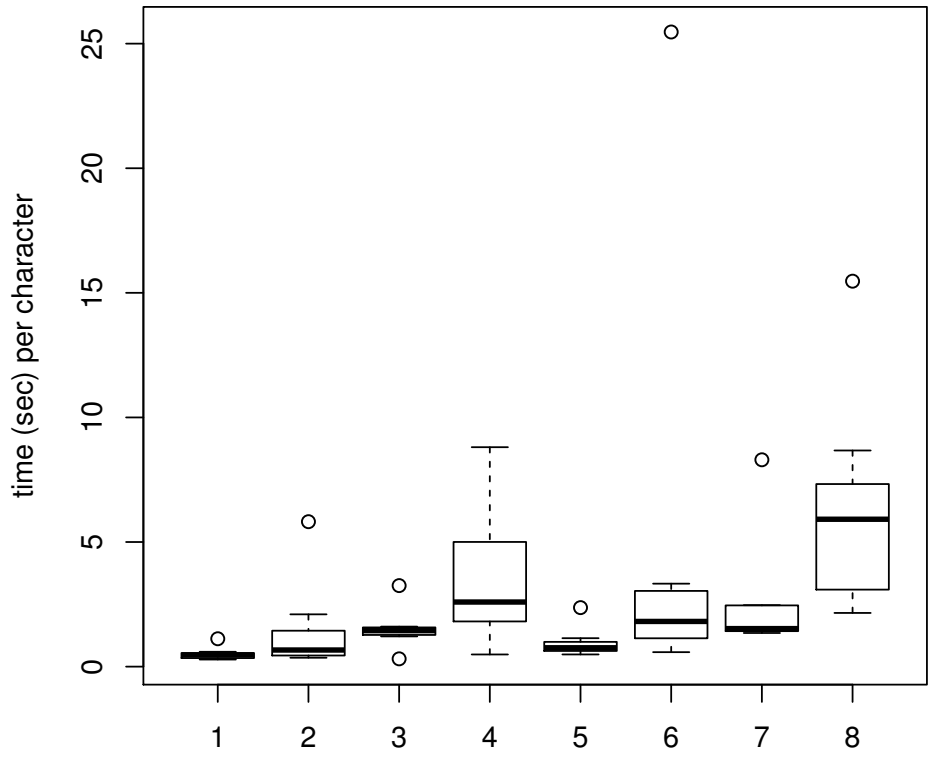


Figure 4-5: Box-and-Whisker plot of the time (per character) spent in eight conditions

condition 8 was the slowest. That the time spent in condition 8 was higher than the time spent in condition 4 and the time spent in condition 7 was higher than the time spent in condition 3, indicates that for the new password counting down had a greater disruptive effect; because, we think, the declarative memory is less robust (therefore step 1 is less robust) for the new password. We also note that the time spent in condition 5 was a little higher than it was for condition 1; which shows that motor memory did help increase the efficiency of the password input, but not significantly so. A cell (x, y) in Table 4.7 shows the difference between the median

Table 4.7: Difference in median time (sec.) per character between pairs of conditions

Condition Index → ↓	1	2	3	4	5	6	7	8
1	=	−0.20	−0.97	−2.12	−0.29	−1.35	−1.06	−5.45
2		=	−0.78	−1.93	−0.09	−1.15	−0.86	−5.25
3			=	−1.15	0.69	−0.37	−0.08	−4.47
4				=	1.84	0.78	1.07	−3.32
5					=	−1.06	−0.77	−5.16
6						=	0.29	−4.10
7							=	−4.39
8								=

time per character in condition x and the median time per character in condition y. We see that Table 4.7 matches well with the qualitative predictions from our model as shown in Table 4.4.

4.4.1 Users' Opinions

At the end of the study, each user was given a questionnaire; the goal of the questions was to learn about the users' subjective experience of the conditions; below we give the questions along with the users' replies:

Have you ever used a different layout than `QWERTY` for a keyboard?

The users only used keyboards with `QWERTY` layout. So, `A2Z Alphabetic & Standard Layout Combination Keyboard` forced the users to visually search for the desired key.

What was challenging about login-ing while counting backwards?

Multitasking was the reason that the users thought made the login-ing while counting backwards condition challenging; one user expressed that he had more difficulty remembering the new password while counting backwards than he had for the old password. So, in general counting backwards was able to keep the STS busy and it was more difficult to fetch from LTS for a new password.

What was challenging about login-ing using a keyboard with an unfamiliar layout?

Table 4.8: Challenges with an unfamiliar layout

User	Reply
1	"Finding the keys"
2	"Muscle memory for initial password didn't match"
3	"Have to find the buttons"
4	"Difficult to find letters (with respect to finger)"
5	"For `QWERTY` automatically my finger goes at the right place but for `A2Z` it is hard to find the right key"
6	"In subconscious I was looking for the keys in the `QWERTY` places"
7	"Difficult to find keys"

Users did have to search for the desired keys visually on the `A2Z` keyboard. From the replies of users 2, 5, and 6, it appears that they had the experience of erroneous finger movements.

Which one was more challenging: counting backwards or unfamiliar layout?

Even though falling back to visual search for keys on the `A2Z` keyboard increased login time more than the time increased due to counting backwards, the predominant subjective evaluation of the users was that counting backwards was more challenging among the two.

Do you think it is a good idea to allow a user to test herself if she has really remembered a password she has just created for a new account?

Users did think that such a test would help one to memorize and remember a new password better.

What type of test, if yo can successfully complete, will make you confident that you have remembered the new password?

Choosing one's password from among similar-looking decoy passwords and letting one perform mock logins after short delays were the replies we got from the users.

4.5 Related Works

Here we discuss some of the research works on motor skill learning related to the model that we have proposed for the password input process.

Willingham et al. [136] propose that when motor sequences are learned implicitly, they are learned in terms of response-locations coded in egocentric space; i.e., with respect to one's body-parts. In [137] it was proposed that the Central Nervous System plans reaching movements in extrinsic space, with target and hand location initially coded as vectors with respect to fixation that are then subtracted to produce an intended movement vector in a hand-centered coordinate system. The transformation of this vector into motor commands depends on the maps or the internal models. Wolpert et al. [138] mentioned, among other possibilities, lookup table as a possible representation for the internal model in the Central Nervous System. For Serial Reaction Time tasks, Koch and Hoffmann [139] found that relational patterns in keystroke sequences (in spatial dimension) helps in motor learning through facilitating the formation of chunks. Consistent insertion of pauses also improved learning for weakly patterned keystroke sequences; however, pauses hampers learning if the keystroke sequence has strong relational patterns, and insertion of random pauses did not help either. They also found that within a chunk, response was quicker and errors were few; but for the first element of a chunk the response was slower and the errors were more. Even in the absence of any relational or statistical structure, visuomotor sequences were learned as chunks which acted as single memory units; chunks were different across different people for the same sequence, and even when the sequence was shuffled preserving the learned chunks, savings in time were observed [140]. In a Serial Reaction Time task, if the sequence is unexpectedly removed and replaced with random trials, the participant initially continues to inappropri-

ately play out the sequence [141], which is somewhat similar to the erroneous finger movements in our model.

4.6 Discussion

Effect of motor memory on the time it takes to correctly type in a password is, according to our model, most vivid when we contrast condition 1 with condition 2; and we see that motor memory might have reduced the required (median) time, and the reduction—according to our data—has an upper bound of 29 centi-seconds per password-character and for say, a password of length 13, it would save at most ≈ 3.77 seconds per login. However, according to our model, motor memory is auxiliary to the input process; the user must perform step 1 accurately in order to initiate the efficiency gained from motor memory. For example, for two of the users, counting backwards (condition 2 and condition 6) had a near-incapacitating effect; they had to pause the counting significantly in order to make any progress in typing in the password; this never happened for conditions 3 and 7. Besides, motor memory for a password takes more time to develop than the declarative memory for a password. Predominantly, the failed attempts in our study occured in the conditions involving counting backwards task; though, the total login time increased more when a user had to visually search for a key with an unfamiliar layout than the increase in time due to counting backwards, users experienced more difficulty in the later condition. As a result, if memory tests are to be given in order to increase a user's confidence that she will remember a newly created password better, we think the tests should aim for bolstering a user's declarative memory of her password.

Chapter 5

Usable End-User Authentication for the Web

5.1 Password Creation Interface

Traditionally when a user wants to create an account on the web, she has to choose a username, a password, and has to provide some information about herself, e.g., her date of birth. Usually there are some restrictions on the types of passwords she can choose; and when a user chooses a password that conforms with the rules, she is asked to confirm the password by typing it in a second time. Fig. 5-1 shows the portion of a Gmail account creation interface where a user creates and confirms her password. After a user has successfully created an account, it is implicitly assumed that she has remembered her newly created password; albeit, the a user is advised to choose a long, hard-to-guess, random string containing lower and upper case letters, digits, and special characters and she is further advised not to reuse a password from another of her web accounts. As a result, a user might have already forgotten her password after a few minutes of creating the account; or she might reuse one of her favorite passwords, or she can use a slight variation of one of her well-memorized old passwords, all of which weaken security.

5.2 Motivation Behind the Proposed Password Creation Interface

The interface we propose, provides a user with the opportunity for testing if she has actually remembered a newly created password. Since a user usually adopts the security-leaking strategies related to a password, only to cope with the burden it creates on her memory, we hope that if a user leaves an account creation interface with the confidence that she will remember the password, she will be motivated to choose a strong password and will be less motivated to reuse passwords from other accounts or to write the password down [13].

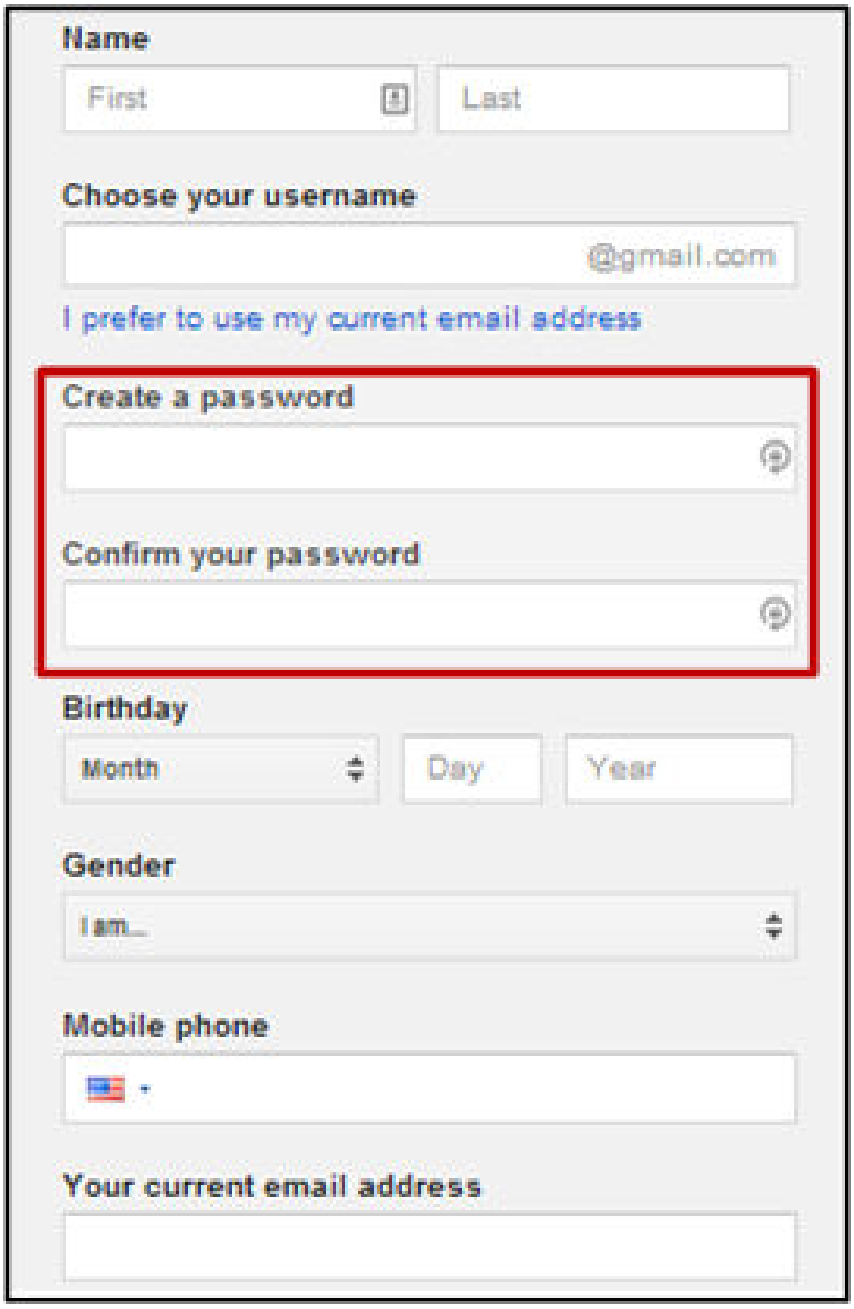


Figure 5-1: Creating a Gmail account

5.3 Description of the Proposed Password Creation Interface

Other than a button called, "Test," our proposed interface is similar to a traditional account creation interface on the web. Before submitting the account information along with the chosen password, a user may test her memory of the new password by clicking on the "Test" button. On clicking the "Test" button, a user is presented with a series of memory tasks built around her new password; the more tasks the user successfully completes, the more likely it is that she has remembered the new password. From memory research, it is known that if a person continues rehearsing a new information, she can maintain it in her short-term memory yet not be able to register the information to her long-term memory; in that case, she can answer questions about the new information but forget the information shortly afterwards. So, in our proposed interface, between any two memory tasks there is a distractor task, which acts as a "rehearsal prevention task" [146]. When a user clicks the "Test" button, the sequence of tasks she is presented with is: distractor task, memory task-1, distractor task, memory task-2, distractor task, memory task-3, distractor task, memory task-4. Next we describe the distractor task and the four memory tasks.

5.3.1 Distractor Task

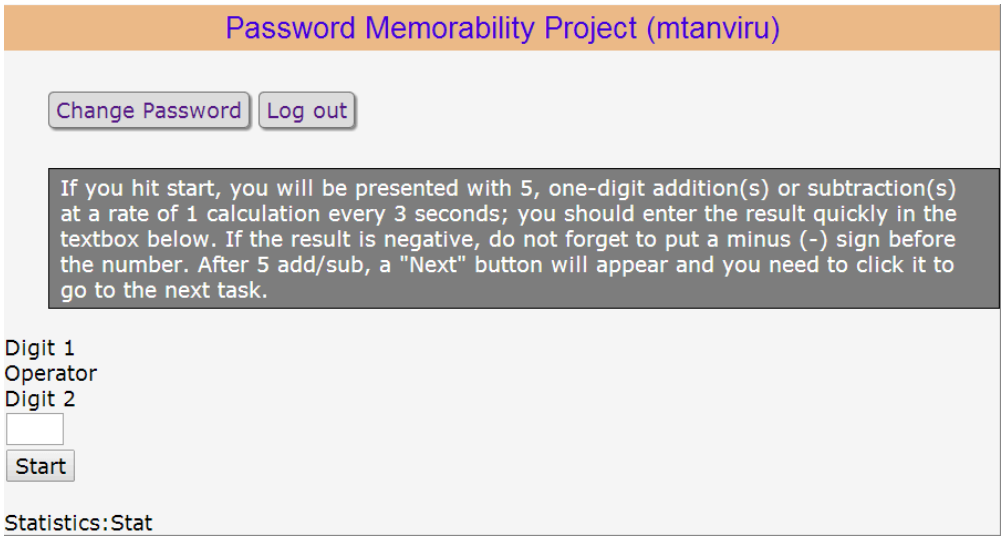


Figure 5-2: Correctly perform the addition or subtraction

Figure 5-2 shows the interface for the distractor task. A user is presented with

five consecutive additions or subtractions between two one-digit numbers in quick succession (one operation about every three seconds); she has to perform the calculations timely and correctly and how many correct calculations she has done so far is shown at the bottom. The motivation for the distractor task is to keep the user's short-term memory occupied, so that it becomes difficult for her to continue rehearsing the new password under her breath. Figure 5-3 shows a distractor task in action.

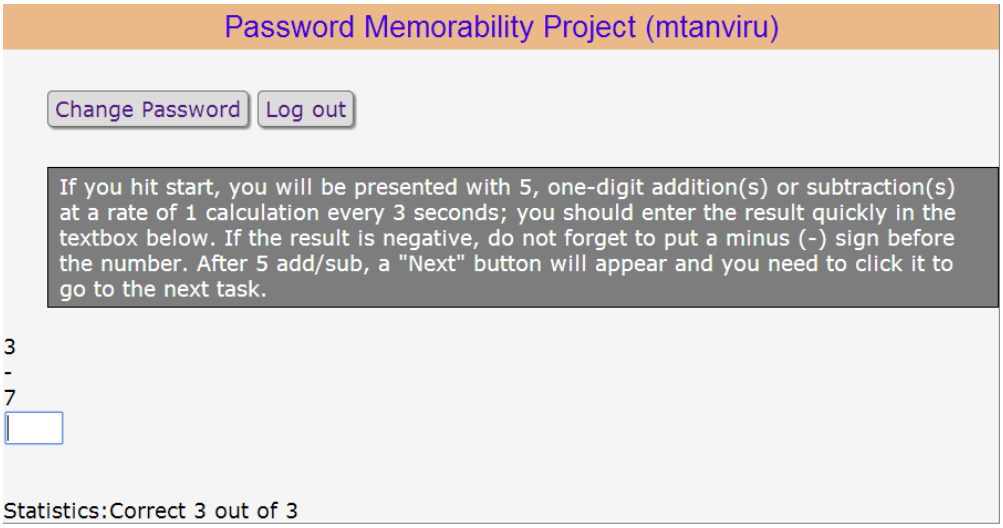


Figure 5-3: In the middle of a distractor task

Algorithm 3 Distractor Task

1: **for** $i \leftarrow 1$ **to** N **do**
2: $m \leftarrow \text{RandomInt}(1, 9)$
3: operator $\leftarrow$ ' + '
4: **if** $\mathcal{U}_{\mathbb{R}} \leq 0.5$ **then**
5: operator $\leftarrow$ ' − '
6: **end if**
7: $n \leftarrow \text{RandomInt}(1, 9)$
8: $\text{Prompt}(m, \text{operator}, n, t)$
9: **end for**

Algorithm 3 shows the pseudocode for a distractor task. On line 1, N denotes how many additions or subtractions a user needs to do in each distractor task; for our experiment $N = 5$. The "RandomInt" function returns an integer between 1 and 9, inclusive. On line 4, $\mathcal{U}_{\mathbb{R}}$ is the uniform random variable. On line 8, the "Prompt" function presents to the user the addition or subtraction involving m, operator, and

n for t seconds and in our experiments $t = 3$.

Although 3 seconds proved to be adequate for an addition or a subtraction between two digits and inputting the result, when the first digit is less than the second digit and the operation is a subtraction, it took users longer than the other operations. If we denote E as the event that the user is prompted with a subtraction of a digit from a smaller digit then E has a binomial distribution. Let us assume $\Pr(E) = p$. Then the distribution is

$$\mathsf{PDF}_E(k) = \binom{M}{k} p^k (1-p)^{M-k} \tag{5.1}$$

Here M is the total number of operations presented to the user. There are $\frac{8 \cdot 9}{2} = 36$ choices out of $9 \cdot 9 = 81$ pairs of (m, n) for which $m < n$. Let $E_<$ be the event that $m < n$ and $E_{\text{`}-\text{'}}$ be the event that the chosen operation is a subtraction. Now $\Pr(E) = \Pr(E_<) \cap \Pr(E_{\text{`}-\text{'}})$. As $E_<$ and $E_{\text{`}-\text{'}}$ are independent events, we have $\Pr(E) = \Pr(E_<) \cdot \Pr(E_{\text{`}-\text{'}}) = \frac{36}{81} \cdot \frac{1}{2} \approx 0.22$. In our experiment, we have four distractor tasks and each distractor task has 5 additions or subtractions; thus $M = 4 \cdot 5 = 20$. So, for our experiment Equation 5.1 takes the form:

$$\mathsf{PDF}_E(k) \approx \binom{20}{k} (0.22)^k (0.78)^{20-k} \tag{5.2}$$

Figure 5-4 shows that E usually occurs 3, 4, or 5 times among the 20 operations that the user has to perform in the four distractor tasks; thus, the user should not face too much of cognitive fatigue.

5.3.2 Memory Task-1

Figure 5-5 shows the interface for the first memory task; here, the correct username ("mtanviru") and the correct password ("B10dG@Jeiw") is buried among similar-looking decoys. A user has to choose the correct username from the left column and the correct password from the right column. The goal of this task is to test if the user has correctly associated her username with her new password in her memory. Traditional account creation interface does not allow a user to test for her username; we believe, it is important that the user correctly remembers not only her password

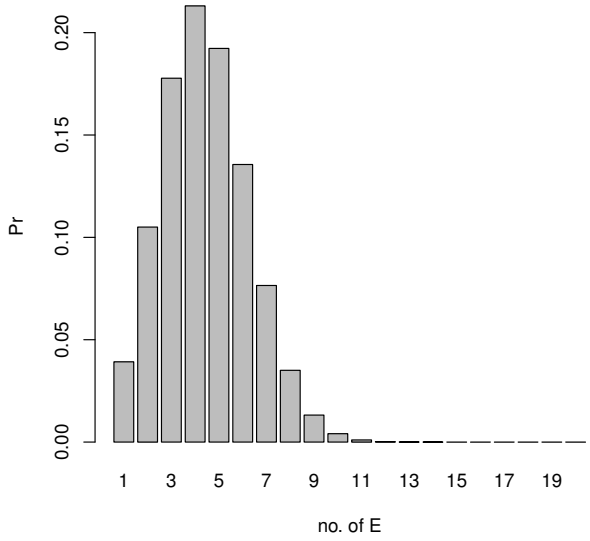


Figure 5-4: Distribution of the number of difficult arithmetic operations

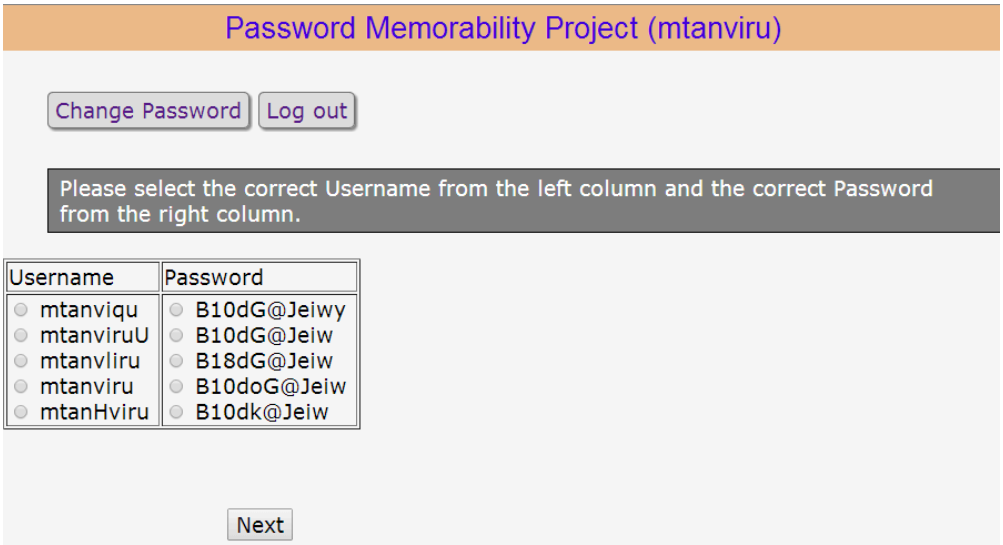


Figure 5-5: Select correct username and password

but also which username is associated with the correct password.

5.3.3 Memory Task-2

After successfully completing the first memory task and the following distractor task, a user is presented with the second memory task, which is shown in Figure 5-6. If a

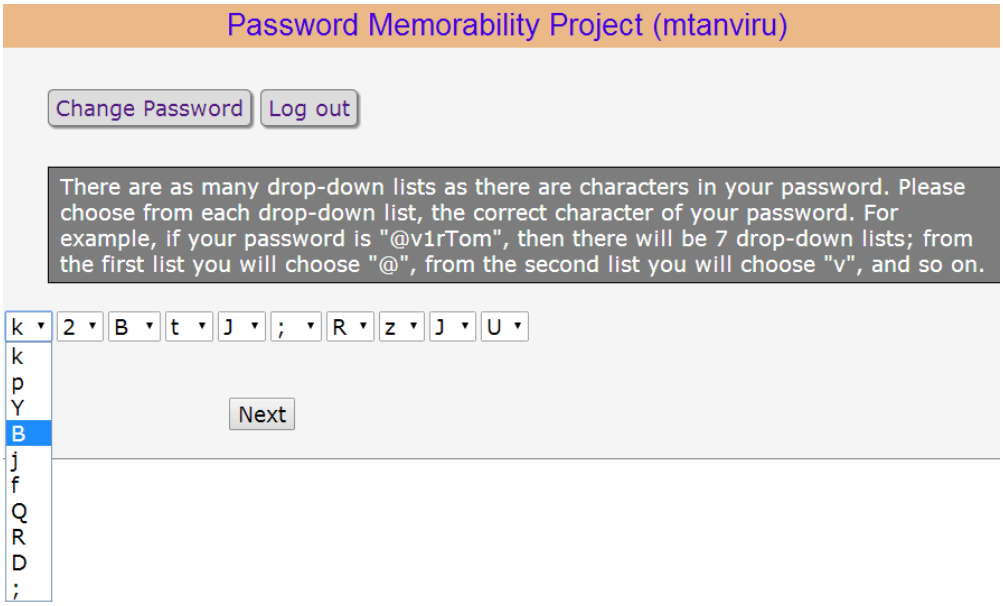
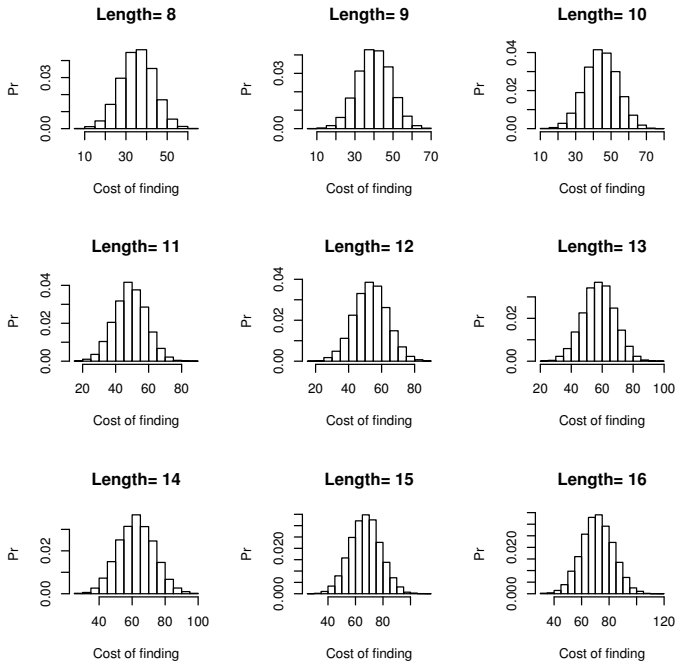


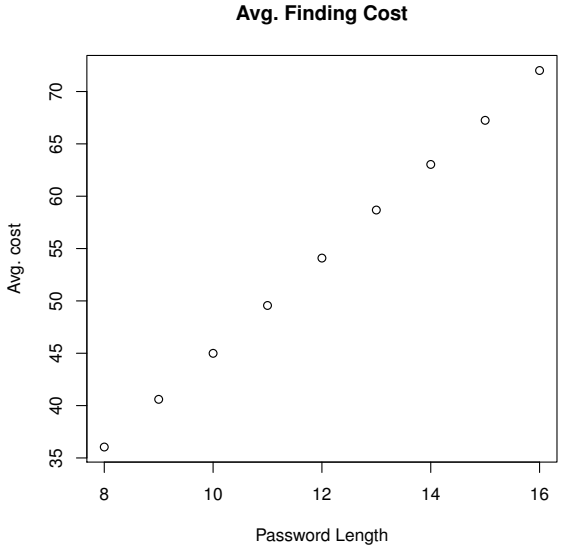
Figure 5-6: Select correct characters from the drop-down lists

user has 10 characters in her password, there will be 10 drop-down lists, each list will contain the correct character for the corresponding position of the password buried among random decoy characters. The goal of this memory task is to test if the user correctly recognizes the characters of her password.

Figure 5-7: 10-sided die [159]



(a) Distribution of user's Effort



(b) Average effort for various lengths

Figure 5-8: User's effort for the second memory task

Algorithm 4 Second Memory Task: Drop-down Lists

```
1: for every character c in Password do
2:   count ← 0
3:   List ← ()
4:   while count < nDecoys do
5:     char ← RandomCharacter(Σ)
6:     if char ≠ c and char ∉ List then
7:       List_count ← char
8:       count ← count + 1
9:     end if
10:  end while
11:  index ← RandInt(0, 9)
12:  Insert(index, char, List)
13:  LoadAsDropdown(List)
14: end for
```

Algorithm 4 shows the pseudocode for creating the drop-down lists, in each of which is buried one character of the password among 9 distinct decoy characters drawn from an alphabet, Σ.

In order to quantify how much effort a user may need to put to complete this memory task, let us assume that the user scans a drop-down list from top to bottom in search of the desired character and she needs equal effort to decide for any decoy character; then her effort for one drop-down list may be quantified by the position where the correct character is in. Thus for a password with n characters in it, a user's total effort for this memory task assumes the probability distribution of the sum of n, 10-sided die like the one shown in Figure 5-7. From Figure 5-8, we see that the average effort a user needs to perform memory task-2 increases linearly with the length of her password.

5.3.4 Memory Task-3

After successfully completing the second memory task and the following distractor task, a user has to perform the third memory task, which tests if the user correctly remembers the order of the characters in her password. Figure 5-9 shows the interface for this task. The characters in the password have been shuffled randomly and the user has to input the indices corresponding to the characters in an order that builds up her password. In the example of Figure 5-9, the password being "B10dG@Jeiw",

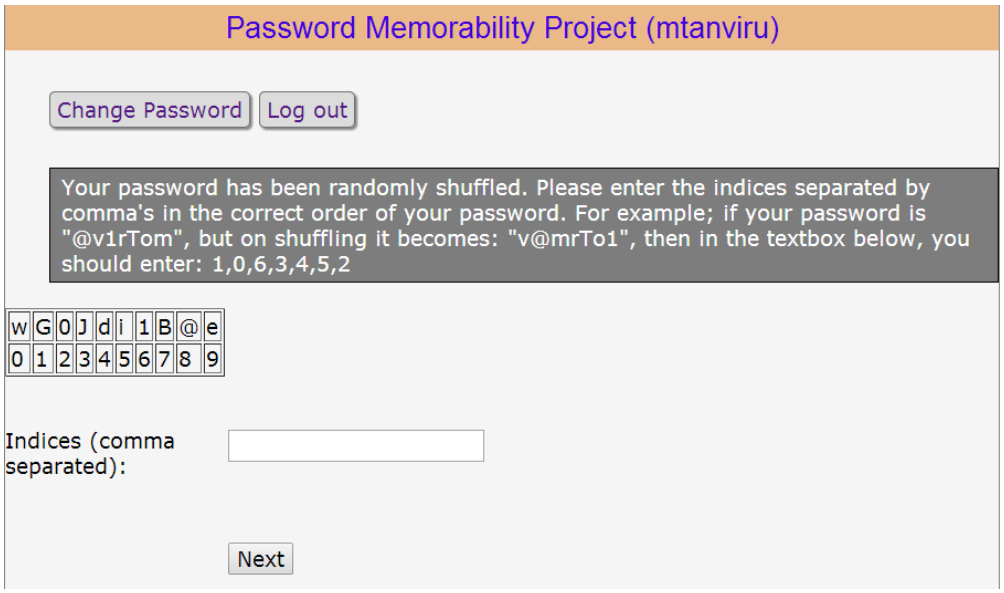


Figure 5-9: Select correct characters from the drop-down lists

the correct input would be "7,6,2,4,1,8,3,9,5,0".

If we assume that a password does not have any character repeated in it, then how much effort a user has to put in performing the Memory Task-3 may be quantified by counting how many inversions occur in the shuffled password with respect to her original (newly created) password. Let $\mathsf{S}[1 \ldots \mathsf{n}]$ be the shuffled version of the password $\mathsf{A}[1 \ldots \mathsf{n}]$. For $i < j$ and $x > y$, if $\mathsf{A}[\mathsf{x}] = \mathsf{B}[\mathsf{i}]$ and $\mathsf{A}[\mathsf{y}] = \mathsf{B}[\mathsf{j}]$, then we call the pair $\langle i, j \rangle$ an inversion [150]. The minimum and the maximum number of inversions are: 0 and $\binom{n}{2}$. If the shuffling is done via multiple independent swaps between randomly chosen pairs of characters, we should not expect the number of inversions to be too high or too low; for, that would correspond to many occurrences of a low probability ($\binom{n}{2}^{-1}$) event; rather, the number of inversions that lie in the middle of the spectrum should have high probability. Figure 5-10 shows the distribution of the number of inversions giving support to our expectation.

5.3.5 Memory Task-4

The final memory task prompts the user to input her new password in reverse order. Figure 5-11 shows the interface and the correct password being "B10dG@Jeiw," the correct input for this task would be "wieJ@Gd01B".

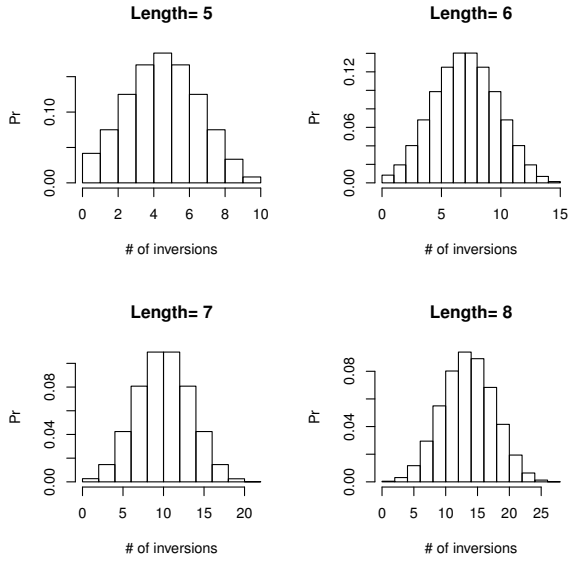


Figure 5-10: Distribution of number of inversions for various password lengths

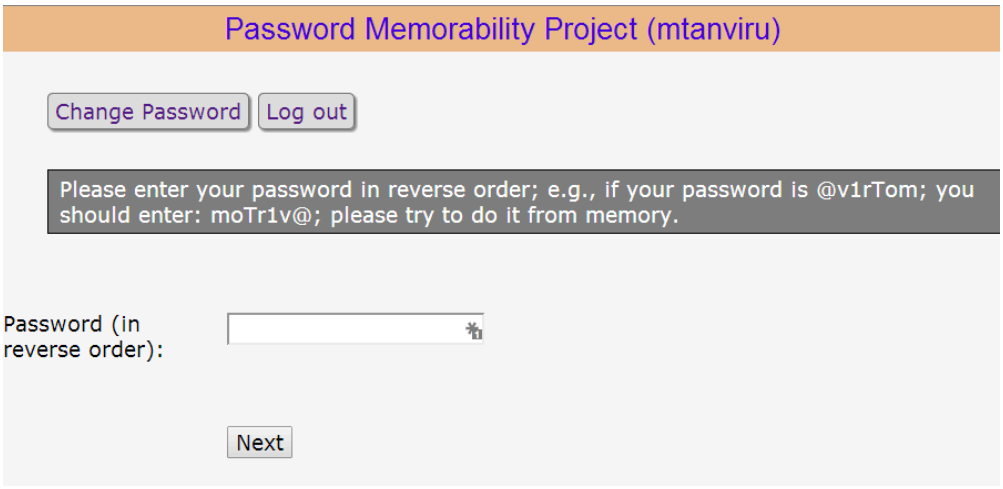


Figure 5-11: Select correct characters from the drop-down lists

By the time the user has successfully completed all four memory tasks, we believe that her declarative memory for her password, i.e., the memory of contents and their order, have been more permanent; additionally she will be more confident about her memory of the password, which may in turn motivate her to choose a strong password.

5.4 Security Issues

All the memory tasks and the distractor tasks, reside on a single `PHP` page; using JavaScript we hide or show pertinent parts of the page. When the user has finished testing her memory for the password, she submits the account information to the server similarly to a traditional interface. So, for our interface, even if it appears that the password is on the clear, it actually never leaves the user's machine as plain text.

A graver security concern with our interface is that it makes the success of a shoulder-surfing attack more probable; because, while the user is performing various memory tasks, the password is shown on the screen several times. We stipulate that the user tests her memory of a password only if she is confident enough that the surroundings is safe, for example at home, as was advised in [6].

5.5 Experimental Procedure

As the study involves human subjects, we acquired approval from the Institutional Review Board of Marquette University. We recruited 21 participants from a class on "Object-Oriented Software Design." In order to motivate the subjects to remember their account credentials, we posted the weekly assignments on our website; so that the students had to login to the website to access the materials related to the assignments in a regular fashion over the whole semester. About half of the participants were randomly assigned to the memory task version of the web interface, while the other half created the account and logged in to our website similarly to any traditional website. Henceforth we refer to the group of users who completed the memory tasks as the "Memory Task Group" (convetionally the experimental

group) and the latter group as the "Traditional Group" (conventionally the control group). The subjects completed a questionnaire at the end of the study.

5.6 Tools Used

We used `MySQL`, `PHP`, and `JavaScript` to build the website and to record the data. For data preprocessing we used `PHP` and for data analysis we used `R` and `Java`.

5.7 Information Collected

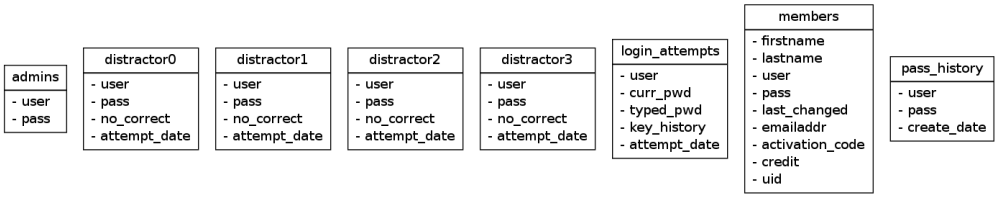


Figure 5-12: Database schema

In the study version of the web interface, we have collected some data about the login behavior of the subjects with a goal to compare between the Memory Task Group and the Traditional Group with respect to their memory for passwords. In Figure 5-12, the database schema has been shown. The table `admins` has the credentials of the administrators of the website, who could upload assignments. For each of the four distractor tasks, there is a corresponding table which records, among other information, how many of the 5 additions/subtractions a user got correct. The table `login_attempts` is crucial, because it records detailed information about each login attempt; specifically in the field `key_history` we record all the keys that the user pressed for a particular login attempt along with how long (in milliseconds) it took to press each of those keys. In the table `pass_history` we record information about all the created passwords and the `members` table keeps track of the users.

5.8 Results

Table 5.1 shows the total number of users, passwords, and login attempts in our study.

Table 5.1: Overall information

Group	No. of users	No. of passwords	No. of attempts
Memory Task	11	15	72
Traditional	10	18	141
Total	21	33	213

5.8.1 Number of Passwords

The first comparison criterion we explore is the number of passwords per user in the two groups. The motivation is that if user A created more passwords than user B, it probably indicates that user A was having more difficulty remembering the passwords than user B. Table 5.2 shows the result.

Table 5.2: No. of passwords per user

Group	Mean	Median	Standard Deviation
Memory Task	1.36	1	0.25
Traditional	1.80	1	1.07

From Table 5.2, we see that the mean number of passwords was higher for the Traditional Group than it was for the Memory Task Group; but, the difference was not significant ($t(12.787) = 1.2112, p = 0.1239$). However, the variance was significantly lower for the Memory Task Group. Both groups passed the Shapiro-Wilk test for normality [151]: for the Memory Task Group we got $W = 0.6245, p = 5.146 \times 10^{-5}$ and for the Traditional Group we got: $W = 0.6405, p = 0.0001687$. Consequently, to compare variance we used F-test [152] with the alternative hypothesis that Memory Task Group had lower variance; and at 95% confidence interval we got: $F(10, 9) = 0.2386, p = 0.01775$.

Table 5.3: Overall valid information

Group	No. of users	No. of passwords	No. of attempts
Memory Task	10	13	60
Traditional	10	15	101
Total	20(1 $\downarrow$)	28(5 $\downarrow$)	161(52 $\downarrow$)

In the `key_history` field of the `login_attempt` table, we noticed that some successful login attempts (typed password matched correct password) did not have enough key strokes, which we think happened because of the browser's auto-completion

and/or auto-fill features. In order to purge such suspect login attempts we use the following definitions:

Definition 5.1. *Valid key. A valid key is one among the* 63 *keys, that produces visible output, on a* `QWERTY` *keyboard.*

The 63 keys include: ␣`‘0123456789-=abcdefghijklmnopqrstuvwxyz[]\;’,./` and the keys on the number pad (if any). We note that using other keys in combination with these 63 keys, one could generate more symbols, e.g., `Shift`+`1` may produce `!`; however, as we are here interested only in the number of password-symbols, counting the 63 keys serves our purpose. We counted the 63 keys from the corresponding key-codes of the onkeydown events in JavaScript for the four web-browsers: Chrome, Firefox, Safari, and IE; the key-codes for the 63 keys and the key-codes for other usual keys, like: `Shift`, `Tab`, `Backspace`, `Delete`, etc., do not overlap accross the four browsers that we considered.

Definition 5.2. *Valid login attempt. A valid login attempt has at least* 3 *valid keys in its corresponding* `key_history` *field.*

Definition 5.3. *Valid password. A valid password has at least* 1 *valid login attempt associated with it.*

Definition 5.4. *Valid user. A valid user has at least* 1 *valid password.*

In the light of the above definitions, after getting rid of the invalid login attempts, passwords, and users, the remainder of the data is shown in Table 5.3. It is somewhat more problematic to define a valid failed attempt than it is to define a valid successful attempt; we decided upon Definition 1 through trial and error. After cleansing, the remaining login attempts had associated with them at least as many keys as 46% of the correct password length. Henceforth, our results are derived from the data in Table 5.3 where in the row “Total” each entry has associated with it the number of reduction in parentheses.

5.8.2 Password Strength

Now, we compare the strengths of the passwords chosen by the users of the two groups, which is important for our study, because it provides a context in which to

interpret the other results. If among the two groups, one has stronger passwords, we think the users of that group are performing a more difficult task remembering the passwords than the other group. We define the symbols that a password can contain as follows:

Definition 5.5. *Simple Symbol. A simple symbol is one of the 7-bit* `ASCII` *characters, except for "Delete," whose* `ASCII` *character-code is 127.*

We note that using combinations of multiple keys on the keyboard, a user could produce single characters that lie outside of the 7-bit `ASCII` character-set, like: ©; but, neither the passwords chosen by the users in our study, nor the corpus of real-life leaked passwords that we consulted, did contain a character that lies outside of the 7-bit `ASCII` character-set; so, Definition 5.5 suffices for our purpose [157].

Definition 5.6. *Composite symbol. A composite symbol is a meaningful English word having at least two characters in it.*

We consider the alphabet Σ from which the symbols for a password can be chosen to consist of two disjoint subsets: Σ_s and Σ_c, where Σ_s contains simple symbols and Σ_c contains composite symbols. We compute the strength of a password in two steps: first, we compute the strength while considering only composite symbols; by the end of the first step the password is stripped of the composite symbols in it; in the second step, we compute the strength of the remainder of the password while considering only simple symbols. For both steps we use Shannon's entropy to compute strength [12]. Then the sum of the Shannon's entropies of the two steps is the final strength of the password. For example, for a password $\pi = \langle s_1, s_2, c_3, s_4, s_5, c_6 \rangle$, the strength σ_1 computed in the first step, the strength σ_2 computed in the second step, and the final strength σ are as follows:

$$\sigma_1 = -p_{c_3} \log (p_{c_3}) - p_{c_6} \log (p_{c_6})$$

$$\sigma_2 = -p_{s_1} \log (p_{s_1}) - p_{s_2} \log (p_{s_2}) - p_{s_4} \log (p_{s_4}) - p_{s_5} \log (p_{s_5})$$

$$\sigma = \sigma_1 + \sigma_2$$

Here, $p_{s_1}, p_{s_2}, \ldots p_{c_6}$ are the symbol-probabilities. Given a dictionary of English

words $\mathcal{D}$ and a set of revealed unique real-life passwords $\mathcal{P}$, the probability distribution function for the composite symbols are obtained empirically following Algorithm 5, which outputs $\mathcal{H}_c$, that contains a mapping of each composite symbol c to its probability and a stripped down version of $\mathcal{P}$, in which no password contains a composite symbol in it.

Algorithm 5 Computing p.d.f. of composite symbols

```
 1: H_c ← ∅
 2: for each password ℘ ∈ P do
 3:     ℘' ← ℘
 4:     n ← |℘'|
 5:     for i ← 0 to n − 1 do
 6:         for j ← n down to i + 1 do
 7:             chunk ← ℘'[i ... (j − 1)]
 8:             if chunk ∈ D then
 9:                 if chunk ∉ H_c then
10:                     H_c ← H_c ∪ {⟨chunk, 1⟩}
11:                 else
12:                     H_c[chunk] ← H_c[chunk] + 1
13:                 end if
14:                 first_half ← ℘'[0 ... (i − 1)]
15:                 second_half ← 1_(j≠n) ℘'[j ... (n − 1)]
16:                 ℘' ← concat(first_half, second_half)
17:                 n ← |℘'|
18:                 break
19:             end if
20:         end for
21:     end for
22:     P[℘] ← ℘'
23: end for
24: m ← Σ_{c∈H_c} H_c[c]
25: for each composite symbol c ∈ H_c do
26:     H_c[c] ← H_c[c] / m
27: end for
28: return(H_c, P)
```

In Algorithm 5, on lines 5–25, for each password we find out the composite symbols contained in it and when one is found we take note of it in $\mathcal{H}_s$ along with how many times we have seen the composite symbol in question and then remove it from the password. After the password has been stripped of all the composite symbols contained in it, we save back the remaining skeleton in $\mathcal{P}$ on line 24. On lines 26–29, we transform the counts for the composite symbols in $\mathcal{H}_s$ into probability. In

the worst case, as Algorithm 5 considers all possible pairs of start and end indices in a password, for each password $\wp$ the algorithm runs $\mathcal{O}\left(\binom{|\wp|}{2}\right)$ times; so in total the worst case complexity for Algorithm 5 is $\mathcal{O}\left(|\mathcal{P}|\binom{|\wp|}{2}\right)$.

Given the output $(\mathcal{H}_c, \mathcal{P})$ from Algorithm 5 and the alphabet of simple symbols Σ_s as inputs, Algorithm 6 computes the probability distribution function for symbol symbols and outputs $\mathcal{H}_s$, which contains a mapping from each simple symbol to its probability.

Algorithm 6 Computing p.d.f. of simple symbols

1: $\mathcal{H}_s \leftarrow \varnothing$
2: **for** each simple symbol $s \in \Sigma_s$ **do**
3: $\quad \mathcal{H}_s \leftarrow \mathcal{H}_s \cup \{\langle s, 1\rangle\}$
4: $\quad$ **for** each skeleton password $\wp \in \mathcal{P}$ **do**
5: $\quad\quad$ **for** each simple symbol $s' \in \wp$ **do**
6: $\quad\quad\quad \mathcal{H}_s[s'] \leftarrow \mathcal{H}_s[s'] + 1$
7: $\quad\quad$ **end for**
8: $\quad$ **end for**
9: $\quad \mathsf{n} \leftarrow \sum_{s' \in \mathcal{H}_s} \mathcal{H}_s[s']$
10: $\quad$ **for** each simple symbol $s' \in \mathcal{H}_s$ **do**
11: $\quad\quad \mathcal{H}_s[s'] \leftarrow \frac{\mathcal{H}_s[s']}{\mathsf{n}}$
12: $\quad$ **end for**
13: **end for**
14: **return** $\mathcal{H}_s$

With $\mathcal{H}_c$ and $\mathcal{H}_s$, as outputs from Algorithm 5 and Algorithm 6, at hand, we can compute the strength of password (in two steps) following Algorithm 7. The **for** loop on line 3 in Algorithm 7 is similar to the **for** loop on line 5 in Algorithm 5; on line 8 we compute the Shannon's entropy.

For the English dictionary $\mathcal{D}$ in Algorithm 5 we added the most common 20 PIN's [156] to the dictionary in [155] containing about a quarter of a million English words. For $\mathcal{P}$ we used the freely available "passwords.txt" [157], which contains about 2.2 million real-life passwords. As a pre-processing step we converted each word in $\mathcal{D}$ and each password in $\mathcal{P}$ to lower-case. As a result, $\mathcal{H}_c$ and $\mathcal{H}_s$ contained only lower-case letters. While giving a password as input to Algorithm 7 we converted it to lowercase first. In Figure 5-13 we see that the cumulative distribution functions for both the composite and simple symbols in real-life passwords are far from a uniform

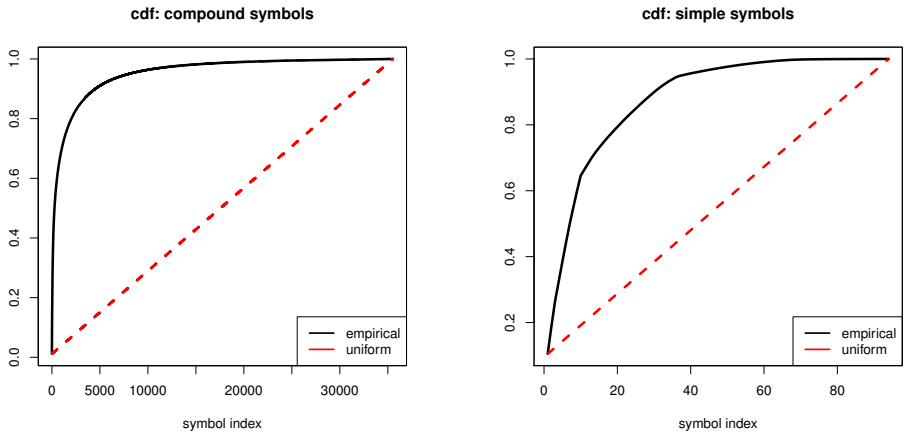
Algorithm 7 Computing strength σ of a password π

1: $\sigma \leftarrow 0$
2: $\mathsf{n} \leftarrow |\pi|$
3: **for** $i \leftarrow 0$ **to** $\mathsf{n} - 1$ **do**
4: **for** $j \leftarrow \mathsf{n}$ **down to** $i + 1$ **do**
5: $\mathsf{chunk} \leftarrow \wp'[i \ldots (j-1)]$
6: **if** $\mathsf{chunk} \in \mathcal{H}_c$ **then**
7: $p_s \leftarrow \mathcal{H}_c[\mathsf{chunk}]$
8: $\sigma \leftarrow \sigma + (-p_s \cdot \log_2(p_s))$
9: $\mathsf{first\_half} \leftarrow \pi[0 \ldots (i-1)]$
10: $\mathsf{second\_half} \leftarrow \mathbb{1}_{(j \neq \mathsf{n})}\pi[j \ldots (\mathsf{n}-1)]$
11: $\pi \leftarrow \mathrm{concat}(\mathsf{first\_half}, \mathsf{second\_half})$
12: $\mathsf{n} \leftarrow |\pi|$
13: **break**
14: **end if**
15: **end for**
16: **end for**
17: **for** each simple symbol $s \in \pi$ **do**
18: $p_c \leftarrow \mathcal{H}_s[s]$
19: $\sigma \leftarrow \sigma + (-p_c \cdot \log_2(p_c))$
20: **end for**
21: **return** σ

distribution. From Figure 5-14, we see that the number of characters contained in composite symbols for both groups are equal. The number of passwords (n), total number of characters in passwords (N), number of composites (m), number of characters contained in composite symbols (M), and the fraction of password characters covered by the composite symbols (f) found by Algorithm 7 in the passwords chosen by the Memory Task Group and the Traditional Group are shown in Table 5.4; once again, we see that the passwords in the Traditional Group contained more composite symbols. Even after eliminating the composite symbols on lines 3–16 in Algorithm 7, meaningful parts remained in the skeleton passwords and most of those were words written with "l33t" substitutions. Incorporating l33t substitutions in Algorithm 5 and 7 increased the computational complexity substantially; we thus leave it as a future work.

Table 5.4: Information about composite symbols

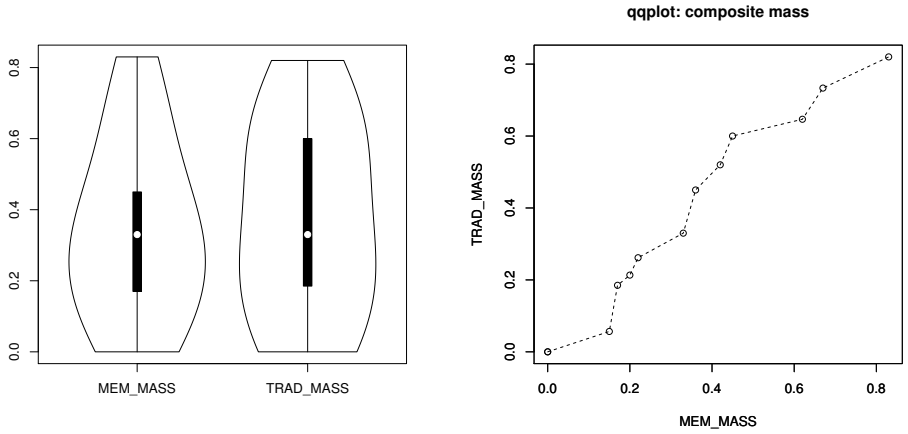
Group	n	N	m	M	$f = \frac{M}{N}$
Memory Task	13	141	13	50	$\approx 35.4\%$
Traditional	15	159	18	63	$\approx 39.6\%$



(a) c.d.f. of composite symbols

(b) c.d.f. of simple symbols

Figure 5-13: Empirically determined c.d.f. for composite and simple symbols



(a) Distribution

(b) Q-Q Plot

Figure 5-14: Fraction of password-length covered by composite symbols

Before discussing the strengths of the chosen passwords of the Memory Task Group and the Traditional Group, we investigate: between two passwords say, $\pi_1 = \langle s_1, s_2, c_1, s6\rangle$ and $\pi_2 = \langle s_1, s_2, s_3, s_4, s_6, \rangle$ of equal length, how often Algorithm 7 assigns more strength to π_2. In other words, between two equal length passwords, one (π_1) contains one or more composite symbols and the other (π_2) containing only simple symbols; and, all the simple symbols contained in π_1 are also present in π_2, we ask whether Algorithm 7 assigns more strength to π_2. We start with the following definition:

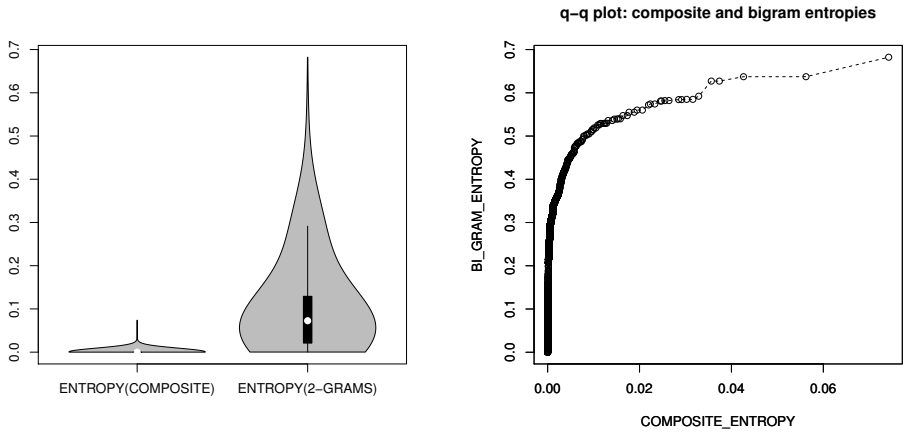
Definition 5.7. *Opportunity Entropies. Opportunity entropies of a composite symbol—of length n—is the set of entropies of all n-grams composed of only simple symbols and none of those n-grams is a composite symbol.*

We argue that most of the times, Algorithm 7 attributes higher strength to π_2, which is desirable; because, intuitively a composite symbol is easier to remember than a equally long n-gram which is not a composite symbol. Our argument is based on the following claim:

Claim 5.1. *Most of the opportunity entropies of almost all composite symbols, are higher than the entropies of the composite symbols themselves.*

Proof. The violin plot in Figure 5-15 shows that most of the 2-grams, which are not composite symbols themselves, have higher entropies than the entropies of almost all composite symbols (of length 2 or more). In fact, 6-th quantile of the 2-grams have higher entropy than about 99% of the composite symbols. As entropy is additive, for all n-grams where $n > 2$, the situation can only be be better. □

The Q-Q plot in Figure 5-16 shows that the strength of the passwords chosen by the Memory Task Group were higher; but, the median strength of the passwords chosen by the Memory Task Group was not significantly higher than the median strength of the passwords chosen by the Traditional Group (Wilcoxon: $W = 110, p = 0.1018$). There was a reason why, for a user in the Memory Task Group, the passwords after the first one could be stronger: since such a user—after creating her account—discovers that she belongs to the Memory Task Group and thus she gets to practice



(a) Violin plot

(b) Q-Q Plot

Figure 5-15: Entropy of composite symbols and entropy of 2-grams

with her password through the Memory Tasks, she might have chosen stronger passwords from the second time onwards. However, from Table 5.6 we see that the three users in the Memory Task Group, who used more than one passwords, did not consistently chose a stronger password as their second one. Consequently, we attribute the higher strength of passwords in the Memory Task Group to chance.

Table 5.5: Password strength

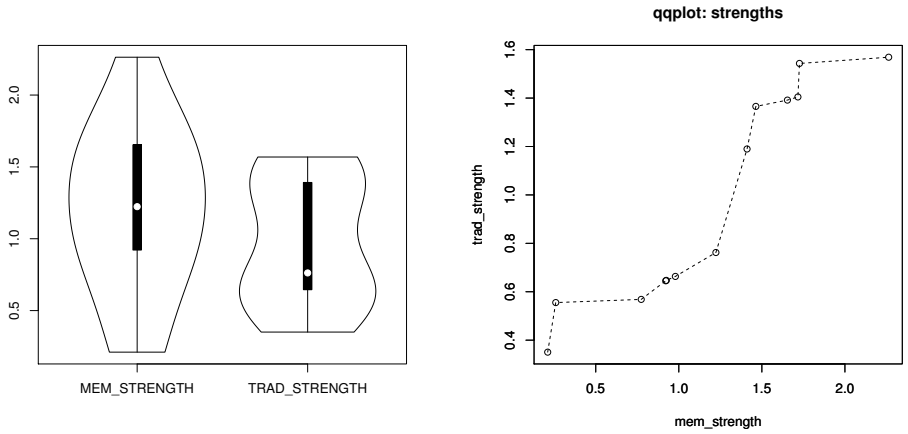
Group	Mean	Median	Standard Deviation
Memory Task	1.19	1.22	0.59
Traditional	0.97	0.76	0.44

Table 5.6: Strength of first and second passwords

User	First Password	Second Password
User 1	2.26	1.22
User 2	1.41	1.65
User 3	1.73	1.72

5.8.3 Success Rate

For a user, we define success rate to be the ratio between the number of successful login attempts and the number of total attempts. However, two or more successful attempts in quick succession or for that matter, two or more failed attempts in quick succession should be considered as a single attempt from the memory point of view.



(a) Violin plot (b) Q-Q Plot

Figure 5-16: Password strengths

We require that two consecutive successful or two consecutive failed attempts must have at least one minute of separation between them. But, if a successful attempt follows a failed attempt within one minute, we consider it to be a recovery and thus we count the latter (successful) but discount the earlier failed attempt.

Table 5.7: Success rate per user

Group	Mean	Median	Standard Deviation
Memory Task	0.91	1	0.13
Traditional	0.87	1	0.23

From Table 5.7, we see that the mean success rate was higher for the Memory Task Group, though it was not significant ($t(14.483) = 0.4402, p = 0.3331$); variance in success rates was lower for the Memory Task Group; but, the significance was marginal ($F(8, 9) = 0.3203, p = 0.0618$).

Correlation Between Password Strength and Success Rate

Using Spearman's rank correlation coefficient [153], we found a small negative correlation (-0.13) between password strength and success rate; though the correlation was not significant: $S = 6183.305, p = 0.2335$.

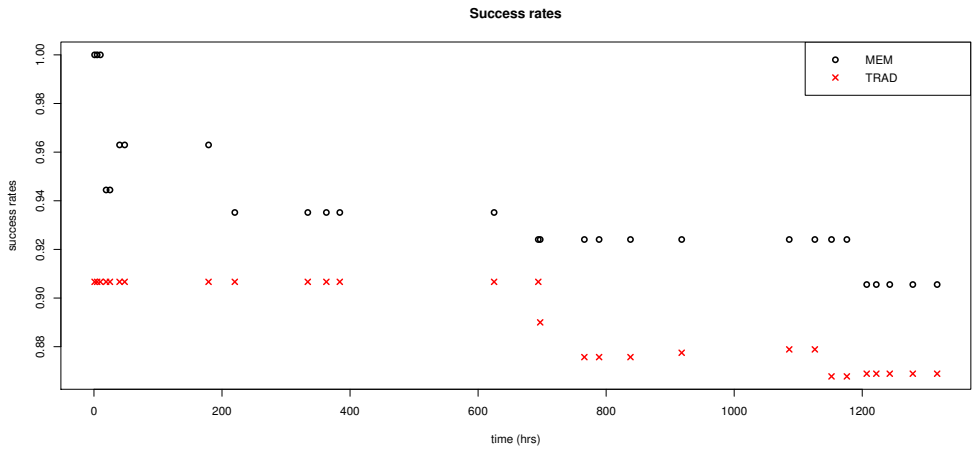


Figure 5-17: Success rates of two groups over time

Success Rate over Time

Figure 5-17 shows the success rates of the two groups at those points in time when both groups made at least one more login attempt than the previous time-point. In the figure we see that for both the Memory Task Group and the Traditional Group, success rates fell over time. Initially the Memory Task Group had much higher success rate than the Traditional Group, but the difference got smaller over time. We think, since the users of the Memory Task Group get an initial boost in memory, their success rates are higher than those of the users in the Traditional Group, but over time the effect of that one-time boost in memory wears off.

5.8.4 Time of the Earliest Significant Failure

Let a user u has a set of passwords Π. Then, for a password $\pi \in \Pi$, if the i-th login attempt fails, we denote it by α_i^-. For a password $\pi \in \Pi$, the date-time of creation is denoted by $\tau(\pi)$ and the date-time of the i-th failed attempt is denoted by $\tau\left(\alpha_i^-\right)$. Then the time of the earliest significant failure for a password π is defined as,

$$\phi(\pi) = \min_{\substack{\langle \alpha_i^-, \alpha_{i+1}^-, \alpha_{i+2}^- \rangle \\ \tau\left(\alpha_{i+2}^-\right) - \tau\left(\alpha_i^-\right) < \epsilon}} \left\{ \tau\left(\alpha_i^-\right) - \tau(\pi) \right\}$$

Table 5.8: Time (min.) of the earliest significant failure

(a) Memory Task Group

User	$\lvert\Pi\rvert$	$\{\phi(\pi)\}$	$\Phi(u)$
u_1	1	$\{\infty\}$	∞
u_2	1	$\{\infty\}$	∞
u_3	1	$\{\infty\}$	∞
u_4	2	$\{13169.87, \infty\}$	**13169.87** $\approx$ 9 days
u_5	2	$\{\infty, \infty\}$	∞
u_6	2	$\{41598.82, \infty\}$	**41598.82** $\approx$ 29 days
u_7	1	$\{\infty\}$	∞
u_8	1	$\{\infty\}$	∞
u_9	1	$\{\infty\}$	∞
u_{10}	1	$\{\infty\}$	∞

(b) Traditional Group

User	$\lvert\Pi\rvert$	$\{\phi(\pi)\}$	$\Phi(u)$
u_1	1	$\{\infty\}$	∞
u_2	1	$\{\infty\}$	∞
u_3	1	$\{\infty\}$	∞
u_4	3	$\{42874.92, 0.5, \infty\}$	**0.5**
u_5	3	$\{\infty, \infty, 8.27\}$	**8.27**
u_6	1	$\{\infty\}$	∞
u_7	1	$\{\infty\}$	∞
u_8	2	$\{\infty, \infty\}$	∞
u_9	1	$\{\infty\}$	∞
u_{10}	1	$\{\infty\}$	∞

For a user u, then, the time of the earliest significant failure is defined as,

$$\Phi(u) = \min_{\pi \in \Pi} \{\phi(\pi)\}$$

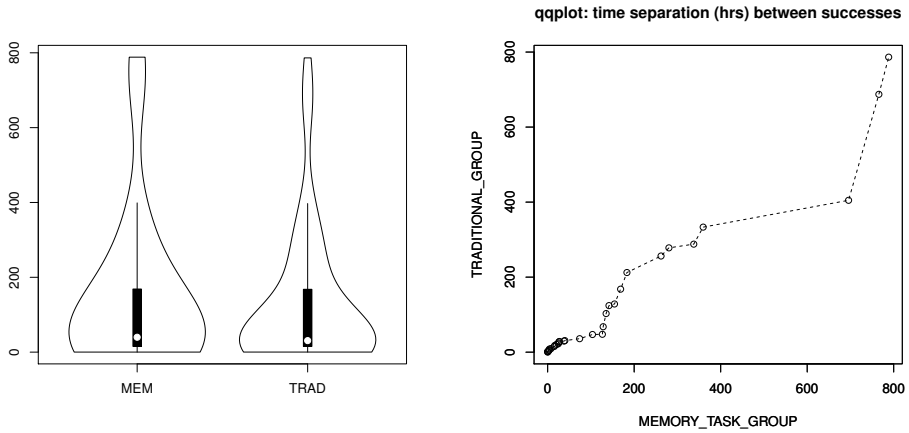
We set $\epsilon = 1$ minute and the result is shown in Table 5.8. For two users of the Traditional Group, significant failure happened almost immediately after the creation of an account; but, for the Memory Task Group the failures happened at least a week after the creation of an account. In our opinion, as a user of the Memory Task Group practice with her password just after creating her account, she gets a memory boost and she is motivated to ponder upon her new password; as a result, if a significant failure occurs, it should occur much later. For our dataset, one interesting consequence of setting $\epsilon = 1$ minute, was that when a user faced a significant failure with a password, she changed the password, which indicates that she forgot it. So, for our dataset, it appears that if a user (of either group) faces a significant failure with a password, then she has forgotten the password.

5.8.5 Confusion in Successful Attempts

We consider two successful attempts as distinct, only if there is a gap of at least one minute between them. Table 5.9 shows information about the time-separation of distinct successful login attempts.

Table 5.9: Time-sepration between distinct successful login attempts (in hrs)

Group	Mean	Median	Standard Deviation	Max.	Min.
Memory Task	150.28	39.76	217.72	788.37	0.12
Traditional	125.59	30.8	184.27	786.41	0.2



(a) Violin plot

(b) Q-Q Plot

Figure 5-18: Time-separation between successful attempts

The violin plot of the time-separation of the distinct successful attempts in Figure 5-18 indicates that time-spacings between successful login attempts of both groups are similar; a Kolmogorov-Smirnov test [154] gives: $D = 0.1319, p = 0.7754$.

Let the set of all distinct successful login attempts of a user u be denoted by $\mathcal{A}_u^+$. Let the key strokes made by a user u while performing a distinct successful login attempt $\alpha_u^+ \in \mathcal{A}_u^+$ be denoted by $\kappa_{\alpha_u^+}$ with $\left|\kappa_{\alpha_u^+}\right|$ strokes in it and the correct password at that attempt be denoted by $\pi_{\alpha_u^+}$ with $\left|\pi_{\alpha_u^+}\right|$ symbols in it. Then for a user u, the confusion in successful login attempts $\mathcal{C}_u$ is defined as:

$$\mathcal{C}_u = \frac{1}{\left|\mathcal{A}_u^+\right|} \sum_{\alpha_u^+ \in \mathcal{A}_u^+} \frac{1}{\left|\pi_{\alpha_u^+}\right|} \left\lfloor \frac{\left|\kappa_{\alpha_u^+}\right| - \left|\pi_{\alpha_u^+}\right|}{2} \right\rfloor$$

We note that $\left|\kappa_{\alpha_u^+}\right| - \left|\pi_{\alpha_u^+}\right| \geq 0$ and as every extra key that the user inputs has to be undone with say a backspace, the term under the summation is the number of extra keys the user has typed relative to the actual length of the password; and it also ignores a trailing "Enter" key. Figure 5-19 shows the distribution of C_u of

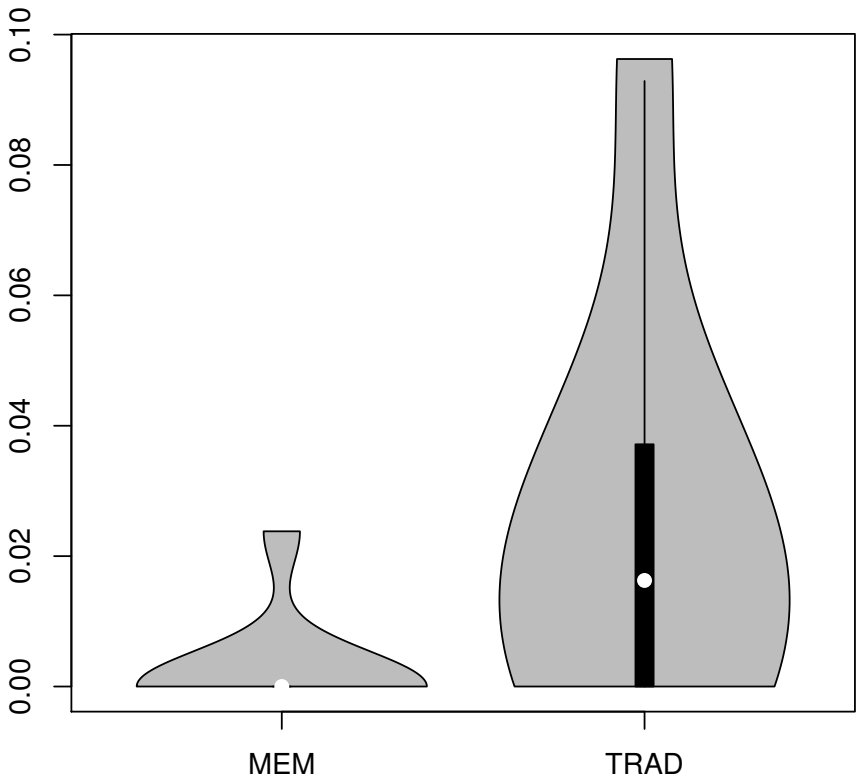


Figure 5-19: Violin plot of confusion

the Memory Task Group (MEM) and the Traditional Group (TRAD). The median C_u for the Memory Task Group was significantly lower than that of the Traditional Group: $t(7.742) = -1.9298, p = 0.04548$.

Correlation Between Password Strength and Confusion

Using Spearman's Rank Correlation Coefficient, we found a small negative correlation (-0.18) between confusion and password strength; the correlation, though, was not significant: $S = 1822, p = 0.21$.

5.8.6 Error in Failed Login Attempts

We consider two failed login attempts to be the same if the time-gap between them is less than 1 minute. In order to measure the amount of error for a failed attempt we use Damerau–Levenshtein distance between the typed password and the correct password [158]. The Levenshtein distance between two strings x and y is given by $\lambda_{x,y}(|x|, |y|)$ where,

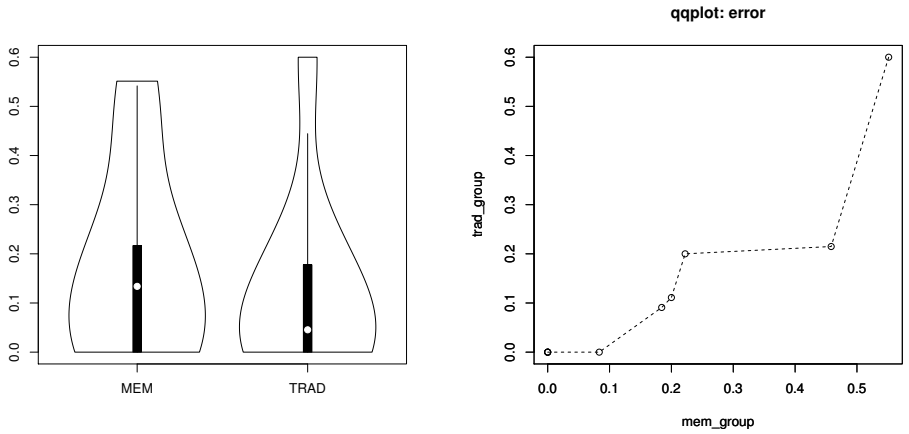
$$\lambda_{x,y}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \lambda_{x,y}(i-1,j) + 1 \\ \lambda_{x,y}(i,j-1) + 1 \\ \lambda_{x,y}(i-1,j-1) + \mathbb{1}_{x_i \neq y_j} \end{cases} & \text{otherwise} \end{cases}$$

Damerau–Levenshtein distance is then given by $\Delta_{x,y}(|x|, |y|)$ where,

$$\Delta_{x,y}(i,j) = \begin{cases} \min \begin{cases} \lambda_{x,y}(i,j) \\ \lambda_{x,y}(i-2,j-2) + \mathbb{1}_{x_i \neq y_j} \end{cases} & \text{if } i > 1, j > 1, x_i = y_{j-1}, x_{i-1} = y_j, \\ \lambda_{x,y}(i,j) & \text{otherwise} \end{cases}$$

The Damerau-Levenshtein distance between two strings is a count of the minimum number of operations needed to transform one string into the other, where an operation can be: a deletion, an insertion, a substitution of a single character, or a transposition of two adjacent characters. For example, if $x =$"people" and $y =$"pepole", then $\lambda_{x,y}(6,6) = 2$, whereas $\Delta_{x,y}(6,6) = 1$.

Let $\mathcal{A}_u^-$ be the set of failed login attempts of a user u, and for a failed login



(a) Violin plot

(b) Q-Q Plot

Figure 5-20: $E_{\alpha_u^-}$ according to equation 5.3

attempt $\alpha_u^- \in \mathcal{A}_u^-$, let the correct password be $\pi^*_{\alpha_u^-}$ and let the password entered by the user be denoted by $\pi_{\alpha_u^-}$. Then the error, $E_{\alpha_u^-}$ for a user u is defined as,

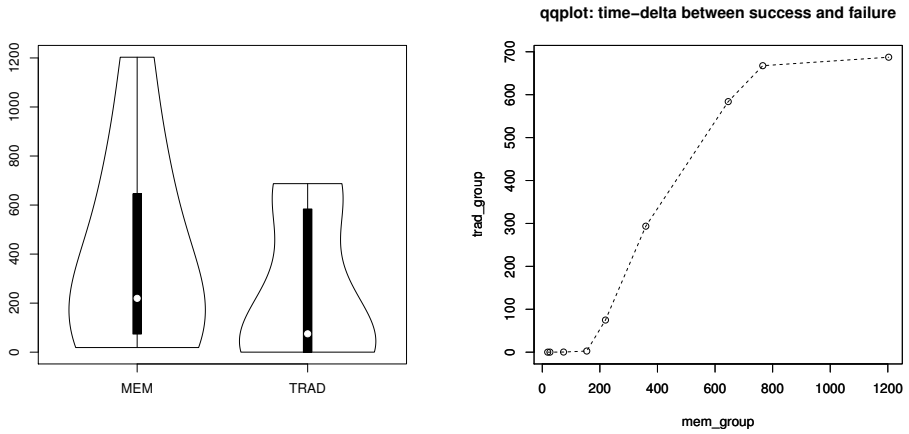
$$E_{\alpha_u^-} = \frac{1}{\left|\mathcal{A}_u^-\right|} \sum_{\alpha_u^- \in \mathcal{A}_u^-} \frac{1}{|\pi^*_{\alpha_u^-}|} \Delta_{\pi_{\alpha_u^-}, \pi^*_{\alpha_u^-}} \left(|\pi_{\alpha_u^-}|, |\pi^*_{\alpha_u^-}| \right) \tag{5.3}$$

Table 5.10: $E_{\alpha_u^-}$ according to equation 5.3

Group	Mean	Median	Standard Deviation
Memory Task	0.17	0.13	0.20
Traditional	0.12	0.05	0.19

From Table 5.10 and the Figure 5-20, we see that the users of the Memory Task Group had a higher $E_{\alpha_u^-}$. In order to investigate why the Memory Task Group, having a higher success rate and a lower confusion, made larger $E_{\alpha_u^-}$, we looked into the distribution of the time-separation between a failed login attempt and the preceding successful login attempt.

From Figure 5-21, we see that the distributions of the time-separations between failed login attempts and the preceding successful login attempts for the two groups are not similar; for the Memory Task Group, the failed login attempts were more distant from the last successful login attempts. As a result, the time-agnostic definition of error in Equation 5.3 was an incorrect way to compare errors between the



(a) Violin plot

(b) Q-Q Plot

Figure 5-21: Time-separation between a failure and the previous success

two groups. We modify Equation 5.3 as follows,

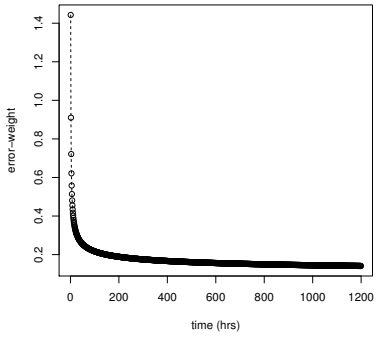
$$\mathcal{E}_{\alpha_u^-} = \frac{1}{\left|\mathcal{A}_u^-\right|} \sum_{\alpha_u^- \in \mathcal{A}_u^-} \frac{1}{|\pi^*_{\alpha_u^-}|} \Delta_{\pi_{\alpha_u^-}, \pi^*_{\alpha_u^-}} \left(|\pi_{\alpha_u^-}|, |\pi^*_{\alpha_u^-}|\right) \cdot \frac{1}{f\left(\tau_{\alpha_u^-}, \tau_{\alpha_u^+}\right)} \tag{5.4}$$

In Equation 5.4 $f\left(\tau_{\alpha_u^-}, \tau_{\alpha_u^+}\right)$ is a non-decreasing function of the difference between the date-time of a failed login attempt $\tau_{\alpha_u^-}$ and the date-time of its preceding successful login attempt, $\tau_{\alpha_u^+}$. Though the simplest choice for $f(\cdot)$ would be to take the difference between its parameters, we adopt the following definition instead:

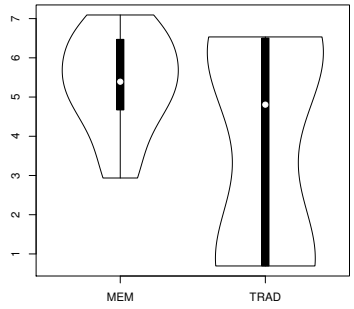
$$f\left(\tau_{\alpha_u^-}, \tau_{\alpha_u^+}\right) = \log_\beta \left(\tau_{\alpha_u^-} - \tau_{\alpha_u^+}\right) \tag{5.5}$$

The motivation behind using 5.5 is that for two errors of equal magnitude—one recent and one distant—we should give more weight to the recent error and two very distant errors should not have too much difference in weights, the user is expected to make an error if the login attempt is too far from her last successful login attempt. We used $\beta = \mathbf{e} = 2.718\ldots$; but, ideally β should be determined for each user. In Figure 5-22 we see the repercussion of using Equation 5.4 for comparing errors between two groups.

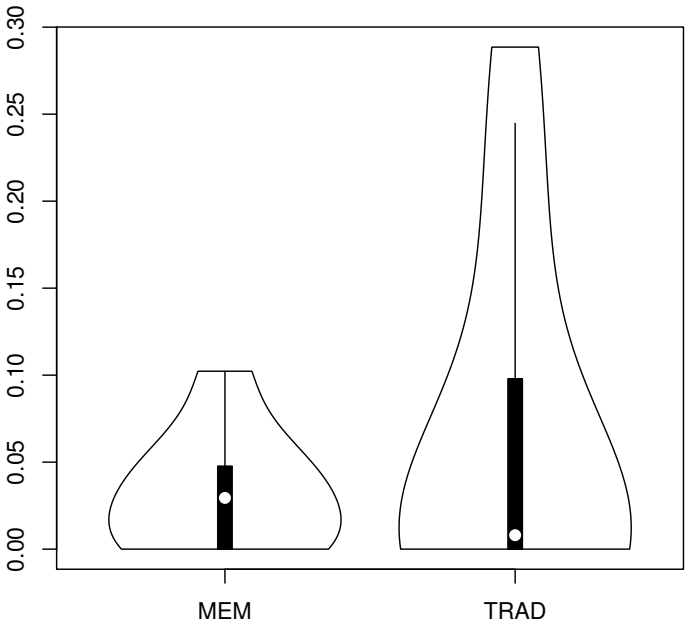
From Table 5.11 we see that the mean value of $(\mathcal{E}_{\alpha_u^-})$ for the Memory Task Group



(a) Weights according to equation 5.5



(b) Log-weighted time-separation



(c) Distribution of $\mathcal{E}_{\alpha_u^-}$

Figure 5-22: Including time in error

Table 5.11: $\mathcal{E}_{\alpha_u^-}$ according to equation 5.4

Group	Mean	Median	Standard Deviation
Memory Task	0.03	0.02	0.03
Traditional	0.06	0.01	0.10

was lower, though not significantly so ($t(10.978) = -0.9946, p = 0.17$).

Correlation Between $\mathcal{E}_{\alpha_u^-}$ and Password Strength

Using Spearman's Rank Correlation Coefficient, we got a small positive correlation (0.07) between $\mathcal{E}_{\alpha_u^-}$ and password strength; though the correlation was not significant: $S = 5084.97, p = 0.36$.

5.9 Discussion

The goal of our interface was to give the user an opportunity to test her memory of a newly created password. We expected that after successfully completing the four memory tasks interleaved between the distractor tasks, a user will feel more confident that she will remember the password better; additionally, the memory tasks will help cement a user's memory of her password and therefore the number of forgotten passwords will be reduced and we thought if a user gets to test her memory for a password in a systematic way, she will eventually be motivated to create stronger passwords than those she usually creates.

In order to compare the password-memorability between the two groups: one using our interface and the other following the traditional password creation ritual, we have defined several metrics, such as: number of passwords per user, success rate, time of the earliest significant failure, confusion, and error in failed attempts. For every metric, even with stronger passwords, the Memory Task Group appears to have performed better than the Traditional Group; although in most cases the significance was low. We believe the reason for the significance being low is: for both groups, the passwords chosen had significant number of composite symbols in them; as a result for some user in both groups; the passwords for them, thus, did not pose much challenge to the memory. As a result, for those users it was equivalent whether they were in the Memory Task Group or in the Traditional Group which

made the effective sample size lower.

That the difference between the success rates of the Memory Task Group and the Traditional Group decreased over time, suggests that completing the memory tasks gave users a memory boost. The time of earleist significant failure was much lower for the Memory Task Group suggesting our interface was able to reduce immediate forgetting and possibly thus it was able to lessen some of the embarrassment a user feels when she forgets a password in a short while after creating it. Lower confusion in successful login attempts indicates that the Memory Task Group recalled their password more accurately. In general, across all the metrics, the variance in the Memory Task Group was much lower than that of the Traditional Group; we think it happened because the memory tasks made every user in the Memory Task Group ponder about their password in a systematic way; thus, the difference between a more cautious and a less cautious user was smaller. As passwords are part of life to many people, usually a person has a strategy of her own for remembering a new password. An adverse effect of our interface could have been to make a user give up her long practiced personal strategy and to start depending on the memory tasks. But in our study, that the Memory Task Group, did at least as well as the Traditional Group in every metric suggests that the users of the Memory Task Group may have considered the tasks as complimentary to their own strategy.

Chapter 6

Conclusions and Future Work

In this chapter, we summarize the major contributions of this dissertation and outline some avenues for future research.

6.1 Research Achievements

Making end-user authentication both usable and secure at the same time, has been a research topic for almost half a century; yet, at present there is no comparable alternative to passwords. Instead of aiming to find a silver bullet that would solve the problems of end-user authentication in every respect, our approach was to address the problem differently for different scenarios. Specifically, in this dissertation, two approaches have been proposed and evaluated: one for smartphones and the other for the Web.

6.1.1 Usable End-User Authentication for Smartphones

- Taking advantage of the built-in sesnors of a smartphone, we have proposed a transparent end-user authentication framework, which— most of the times— would authenticate the user without her active involvement and thus it would free the user from the burden of remembering secrets and the burden of participating actively in the authentication process many times everyday.

- We have implemented an authentication system using gait pattern and location trace for the Android platform based on the proposed framework. Based on the data gathered from users, we have shown that the authentication system is technically feasible.

- We have identified several issues, like: on-body placement, pace, battery drainage etc., that must be addressed for any transparent end-user authentication system for smartphones.

- The metaphor of a pet has proved useful in communicating—how the authentication system works—to a user; moreover, it might help allay a user's anxiety due to the convert nature of the authentication process. For example, a user may intuitively know that the information about her traits is gathered and used by the authentication system, but never leaves the phone.

6.1.2 A Closer Look into the Password Input Process

- We have designed a within-subjects study to delineate the relative importance of the declarative and the motor memory for a password in performing a successful login. Based on the results from the study, we have outlined the reasons for choosing the memory tasks that help cement the declarative memory for a newly created password over the tasks that aim at bolstering the motor memory for a password.

- We have proposed a conceptual model for the password input process from the perspective of human memory and have shown that the predictions of the model agree with the data obtained from the study.

- The empirical results justify the conclusion that the motor memory for a password does not work independently; rather, it facilitates the password input process by reducing the time required for login.

6.1.3 Usable End-User Authentication for the Web

- We have designed a password-creation web interface that lets a user test if she has memorized a newly created password.

- Through a semester-long between-subjects study we have shown that our password creation interface provides a memory boost to its users and is thus able to reduce the number of immediate forgetting of passwords.

- We have defined several metrics to compare between the experimental group and the control group, like: success rate, time of the earliest significant failure, confusion in successful login attempts and error in failed login attempt.

- We also have found that it is important to interpret the results of the comparison in the context of password-strength chosen by the users of the two groups; because, if one group has passwords with higher strength, they are performing a more difficult task remembering them. Accordingly, we modified the definition of password-strength to incorporate the memorability aspect of a password; as a result, with our definition, if a password has a higher strength, it is usually more difficult to remember.

6.2 Summary of Contributions

Because it usually does not provide any direct value to the user, end-user authentication is considered to be a necessary but secondary task. As a result, users are not motivated to use the authentication methods that provide high security but are difficult to use. On the other hand, security requires a holistic approach—weakness at any point can render the whole chain of security useless. Our proposed authentication framework for smartphones will be able to relieve a user of the burden of authentication most of the times and thus a user will be better able to focus on her primary tasks. For web authentication, our proposed password-creation interface is able to give a user more confidence in her memory of a newly created password, which in turn may motivate her to choose stronger passwords. Together, the transparent authentication framework for smartphones and the password-creation web-interface, proposed in this dissertation, make the overall security ecosystem healthier.

In the long run, a smartphone, which is able to recognize the user on its own might work as a universal intermediary between the user and other computer systems; secrets will then be shared between other systems and the phone and as the phone knows the user and other systems know the phone, the user will at last be free from the burden of authentication once and for all.

6.3 Future Research Directions

Below we discuss some of the possible avenues for future reseach.

6.3.1 Usable End-User Authentication for Smartphones

- In our implementation we focused on gait pattern; augmenting the set of traits would make the authentication framework more interesting—it would then be able to choose between options for achieving the desired level of security at a low cost.

- Conducting the experiments with a larger and more diverse sample is an obvious next step.

- How to incorporate context, e.g., location cues into the formal framework, is an issue that needs to be addressed.

- Another important future direction would be to explore how a user might easily and effectively set the security level she wants.

- How accurately the different traits perform in identifying a user for different on-body placements of the phone, is another future vista that needs exploring.

- Transparent authentication systems, by definiton, are continuous in nature; thus, battery drainage is a crucial issue. We have found out that the continuous authentication using the gait-pattern of a user consumes only 10% more battery charge than the base-level consumption; whereas, the continuous GPS-sensing (for location cue) has a much larger negative effect on the battery life. If we want to add more traits and thus more types of continuous sensing to the authentication system, low-cost probe-sensing will need to be explored.

- How to incorporate user-preferences towards various traits, to be used for authentication, into the cost metric is another useful direction for future research.

- With our authentication framework, the more the user trains the authentication system, the more accurately the system will be able to authenticate the user. Whether representing the authentication system using an avatar of a pet—that shows positive expressions and gestures when it has recognized the user and that shows signs of distress when the authentication process is

costly—motivates the user towards training it more frequently, is an interesting question that we wish to investigate in the future.

6.3.2 A Closer Look into the Password Input Process

- The sample size in our experiment was small; performing the experiments with a larger and more diverse sample is a logical next step.
- In our experiments, we only collected the time it takes to type in a password. More information, like: how many failed login occurs in the conditions, what types of errors are made in different conditions, etc. can shed more light on the issue of motor memory for a password.
- The `A2Z` alphabetic keyboard used in our experiments, though unfamiliar to the user, did have a predictable layout—it was alphabetic and the digits-row was similar to a `QWERTY` layout; using a fully random layout might have been better at delineating the difference between the conditions.
- The distractor task of counting backwards by two, was able to make the fetching from long-term store more difficult, but the pace at which users counted varied; it would have been more revealing if the task were more consistent across users; e.g., counting-down to the regular ticks from a metronome.
- Although the motor memory for a newly created password needs significantly more time to develop than the declarative memory for the same password, there may be a schedule at which—if the user practices typing in her password—the motor memory develops more efficiently; in [135] we find some such hints and we wish to explore it in the future.

6.3.3 Usable End-User Authentication for the Web

- Because in order to login, a user needs to ultimately type in a password, alongside the four memory tasks in our experiment, mock logins—which enables more transfer-appropriate processing—interleaved with distractor tasks could have made the memory boost more effective [146]. Additionally, finding

more interesting and effective memory tasks is a general direction for future exploring.

- A larger and more diverse sample size and stronger passwords would have made the difference between the experimental and the control group significant.
- There are some hints in [132] on how to sort the memory tasks according to their effectiveness at helping a user remember a newly created password and in the future, we want to explore if such sorting can be effectively done for the memory tasks. Such a sorting would enable us to recruit the more effective tasks earlier in the procession of memory tasks that a user would perform and thus we shall be able to tax on a user's patience more judiciously.
- From the study, it is apparent that with our interface the one-time practice with a new password gives a memory-boost to its users; but, the salutary effects of that boost wears off over time; it is an interesting future work to explore if giving the users such memory-boosts at regular intervals would be able to maintain the login success rate high. Whether the memory tasks remain effective in providing memory boosts in case of multiple passwords interference is another interesting research question.
- A goal of our password-creation interface is to motivate a user towards creating stronger passwords by making her confident about her memory of a newly created password through the memory tasks. In the future, from a longitudinal study, we wish to find out if, in the long-term, a user is actually being motivated towards creating stronger passwords from using our password-creation interface.
- In our study, the users were young; memory problems are more pervasive among the elderly population; thus, whether our password-creation interface has a more beneficial effect for elderly people is a work we want to pursue in the future.
- From memory research [169, 170] it is known that not all times of a day is equally good for memorizing information—usually afternoons are better than

mornings; whether creating a password using our interface and performing the memory tasks at a suitable time of day, help the user remember the password significantly better is another of our future works.

- Users of our password-creation interface—if prompted to change passwords frequently—may find our interface less attractive due to the reduced return of investment with respect to the perceived effort they are putting in remembering a new password; finding out whether that is the case needs further exploring. In any case, it has been shown that forcing users to change their passwords frequently—contrary to the common belief—does not help increase security significantly [11].

Bibliography

[1] Daswani, N., Kern, C., and Kesavan, A.: Foundations of Security: What Every Programmer Needs to Know. Apress (2007)

[2] Stallings, W. and Brown, L.: Computer Security: Principles and Practice. Prentice Hall (2007)

[3] Bishop, M.: Computer Security: Art and Science. Addison-Wesley (2002)

[4] Bianchi, A., Oakley, I., and Kwon., D. 2012. Open Sesame: Design Guidelines for Invisible Passwords. Computer, vol.45, no.4, pp.58–65, April 2012, doi:10.1109/MC.2012.109

[5] Bonneau, J., Herley, C., van Oorschot, P. C., and Stajano, F. 2012. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP '12). IEEE Computer Society, Washington, DC, USA, 553–567. DOI=10.1109/SP.2012.44 `http://dx.doi.org/10.1109/SP.2012.44`

[6] Forget, A., Chiasson, S., van Oorschot, P.C., Biddle, R.: Improving text passwords through persuasion. In Proceedings of the 4th symposium on Usable privacy and security (SOUPS '08). ACM, New York, NY, USA, 1–12. doi:10.1145/1408664.1408666

[7] Weir, M., Aggarwal, S., Collins, M., Stern, H.: Testing metrics for password creation policies by attacking large sets of revealed passwords. In Proceedings of the 17th ACM conference on Computer and communications security (CCS '10). ACM, New York, NY, USA, 162–175. doi:10.1145/1866307.1866327

[8] Shay, R., Komanduri, S., Kelley, P.G., Leon, P.G., Mazurek, M.L., Bauer, L., Christin, N., Cranor, L.F.: Encountering stronger password requirements: user attitudes and behaviors. In Proceedings of the Sixth Symposium on Usable Privacy and Security (SOUPS '10). ACM, New York, NY, USA, , Article 2 , 20 pages. doi:10.1145/1837110.1837113

[9] Komanduri, S., Shay, R., Kelley, P.G., Mazurek, M.L., Bauer, L., Christin, N., Cranor, L.F., Egelman, S.: Of passwords and people: measuring the effect of password-composition policies. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 2595–2604. doi:10.1145/1978942.1979321

[10] Kim, H., Huh, J.H.: PIN selection policies: Are they really effective?, Computers & Security, Volume 31, Issue 4, June 2012, Pages 484–496, ISSN 0167-4048, doi:10.1016/j.cose.2012.02.003

[11] Zhang, Y., Monrose, F., Reiter, M.K.: The security of modern password expiration: an algorithmic framework and empirical analysis. In Proceedings of the 17th ACM conference on Computer and communications security (CCS '10). ACM, New York, NY, USA, 176–186. doi:10.1145/1866307.1866328

[12] Smith, R. E.: Authentication: From Passwords to Public Keys. Addison-Wesley (2002)

[13] Adams, A. and Sasse, M., A. 1999. Users are not the enemy. Commun. ACM 42, 12 (December 1999), 40–46. DOI=10.1145/322796.322806 http://doi.acm.org/10.1145/322796.322806

[14] Kuo, C., Romanosky, S., Cranor, L.F.: Human selection of mnemonic phrase-based passwords. In Proceedings of the second symposium on Usable privacy and security (SOUPS '06). ACM, New York, NY, USA, 67–78. doi:10.1145/1143120.1143129

[15] Shay, R., Kelley, P.G., Komanduri, S., Mazurek, M.L., Ur, B., Vidas, T., Bauer, L., Christin, N., Cranor, L.F.: 2012. Correct horse battery staple: exploring the usability of system-assigned passphrases. In Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS '12). ACM, New York, NY, USA, Article 7, 20 pages. doi:10.1145/2335356.2335366

[16] Kirkpatrick, E., A.: 1894. An experimental study of memory. Psychological Review, Vol 1(6), November 1984, 602–609

[17] Biddle, R., Chiasson, S., Van Oorschot, P.C.: Graphical passwords: Learning from the first twelve years. ACM Comput. Surv. 44, 4, Article 19 (September 2012), 41 pages. doi:10.1145/2333112.2333114

[18] Wiedenbeck, S., Waters, J., Birget, J-C., Brodskiy, A., Memon, N.: Authentication using graphical passwords: effects of tolerance and image choice. In Proceedings of the 2005 symposium on Usable privacy and security (SOUPS '05). ACM, New York, NY, USA, 1–12. doi:10.1145/1073001.1073002

[19] Dirik, A.E., Memon, N., Birget, J-C.: Modeling user choice in the PassPoints graphical password scheme. In Proceedings of the 3rd symposium on Usable privacy and security (SOUPS '07). ACM, New York, NY, USA, 20–28. doi:10.1145/1280680.1280684

[20] Chiasson, S., Biddle, R., van Oorschot, P.C.: A second look at the usability of click-based graphical passwords. In Proceedings of the 3rd symposium on Usable privacy and security (SOUPS '07). ACM, New York, NY, USA, 1–12. doi:10.1145/1280680.1280682

[21] Chiasson, S., Forget, A., Stobert, E., van Oorschot, P.C., Biddle, R.: Multiple password interference in text passwords and click-based graphical passwords. In Proceedings of the 16th ACM conference on Computer and communications security (CCS '09). ACM, New York, NY, USA, 500–511. doi:10.1145/1653662.1653722

[22] Jermyn, I., Mayer, A., Monrose, F., Reiter, M.K., Rubin, A.D.: The design and analysis of graphical passwords. In Proceedings of the 8th conference on USENIX Security Symposium - Volume 8 (SSYM'99), Vol. 8. USENIX Association, Berkeley, CA, USA, 1-1

[23] Dunphy, P., Yan, J.: Do background images improve "draw a secret" graphical passwords?. In Proceedings of the 14th ACM conference on Computer and communications security (CCS '07). ACM, New York, NY, USA, 36–47. doi:10.1145/1315245.1315252

[24] Zakaria, N.H., Griffiths, D., Brostoff, S., Yan, J.: Shoulder surfing defense for recall-based graphical passwords. In Proceedings of the Seventh Symposium on Usable Privacy and Security (SOUPS '11). ACM, New York, NY, USA, , Article 6 , 12 pages. doi:10.1145/2078827.2078835

[25] Komanduri, S., Hutchings, D.R.: Order and entropy in picture passwords. In Proceedings of graphics interface 2008 (GI '08). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 115–122

[26] Hayashi, E., Dhamija, R., Christin, N., Perrig, A.: Use Your Illusion: secure authentication usable anywhere. In Proceedings of the 4th symposium on Usable privacy and security (SOUPS '08). ACM, New York, NY, USA, 35–45. doi:10.1145/1408664.1408670

[27] Denning, T., Bowers, K., van Dijk, M., Juels, A.: Exploring implicit memory for painless password recovery. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 2615–2618. doi:10.1145/1978942.1979323

[28] Hayashi, E., Hong, J., Christin, N.: Security through a different kind of obscurity: evaluating distortion in graphical authentication schemes. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 2055–2064. doi:10.1145/1978942.1979242

[29] Passfaces Corporation. `http://www.realuser.com/` (accessed April 27, 2014)

[30] Dunphy, P., Nicholson, J., Olivier, P.: Securing passfaces for description. In Proceedings of the 4th symposium on Usable privacy and security (SOUPS '08). ACM, New York, NY, USA, 24–35. doi:10.1145/1408664.1408668

[31] Everitt, K.M., Bragin, T., Fogarty, J., Kohno, T.: A comprehensive study of frequency, interference, and training of multiple graphical passwords. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09). ACM, New York, NY, USA, 889–898. doi:10.1145/1518701.1518837

[32] Khot, R.A., Srinathan, K., Kumaraguru, P.: MARASIM: a novel jigsaw based authentication scheme using tagging. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 2605–2614. doi:10.1145/1978942.1979322

[33] De Luca, A., Denzel, M., Hussmann, H.: Look into my eyes!: can you guess my password?. In Proceedings of the 5th Symposium on Usable Privacy and Security (SOUPS '09). ACM, New York, NY, USA, , Article 7 , 12 pages. doi:10.1145/1572532.1572542

[34] Forget, A., Chiasson, S., Biddle, R.: Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 1107–1110. doi:10.1145/1753326.1753491

[35] Bulling, A., Alt, F., Schmidt, A.: Increasing the security of gaze-based cued-recall graphical passwords using saliency masks. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 3011–3020. doi:10.1145/2207676.2208712

[36] Dunphy, P., Heiner, A.P., Asokan, N.: A closer look at recognition-based graphical passwords on mobile devices. In Proceedings of the Sixth Symposium on Usable Privacy and Security (SOUPS '10). ACM, New York, NY, USA, Article 3 , 12 pages. doi:10.1145/1837110.1837114

[37] Wright, N., Patrick, A.S., Biddle, R.: Do you see your password? : applying recognition to textual passwords. In Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS '12). ACM, New York, NY, USA, Article 8, 14 pages. doi:10.1145/2335356.2335367

[38] Aviv, A. J., Gibson, K., Mossop E., Blaze M., Smith, J. M.: Smudge attacks on smartphone touch screens. WOOT, 1–7 (2010)

[39] Zezschwitz, E.v., Koslow, A., Luca, A.D., Hussmann, H.: Making graphic-based authentication secure against smudge attacks. In Proceedings of the

2013 international conference on Intelligent user interfaces (IUI '13). ACM, New York, NY, USA, 277–286. doi:10.1145/2449396.2449432

[40] Wikipedia contributors, "Shoulder surfing (computer security)," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Shoulder_surfing_(computer_security)#cite_note-1 (accessed April 27, 2014)

[41] Roth, V., Richter, K., and Freidinger, R. 2004. A PIN-entry method resilient against shoulder surfing. In Proceedings of the 11th ACM conference on Computer and communications security (CCS '04). ACM, New York, NY, USA, 236–245. DOI=10.1145/1030083.1030116 http://doi.acm.org/10.1145/1030083.1030116

[42] Weinshall, D. 2006. Cognitive Authentication Schemes Safe Against Spyware (Short Paper). In Proceedings of the 2006 IEEE Symposium on Security and Privacy (SP '06). IEEE Computer Society, Washington, DC, USA, 295–300. DOI=10.1109/SP.2006.10 http://dx.doi.org/10.1109/SP.2006.10

[43] Philippe Golle and David Wagner. 2007. Cryptanalysis of a Cognitive Authentication Scheme (Extended Abstract). In Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP '07). IEEE Computer Society, Washington, DC, USA, 66–70. DOI=10.1109/SP.2007.13 http://dx.doi.org/10.1109/SP.2007.13

[44] Bojinov, H., Sanchez, D., Reber, P., Boneh, D., Lincoln, P.: Neuroscience meets cryptography: designing crypto primitives secure against rubber hose attacks. In Proceedings of the 21st USENIX conference on Security symposium (Security'12). USENIX Association, Berkeley, CA, USA, 33-33

[45] A Technological Assault on the Password. http://www.technologyreview.com/news/523371/ces-2014-a-technological-assault-on-the-password/ (accessed April 27, 2014)

[46] Okumura, F., Kubota, A., Hatori, Y., Matsuo, K., Hashimoto, M., Koike, A.: A Study on Biometric Authentication based on Arm Sweep Action with

Acceleration Sensor. Int. Symp. on Intell. Signal Process. and Commun. (2006). doi: 10.1109/ISPACS.2006.364871

[47] Gafurov, D., Snekkkenes, E.: Arm Swing as a Weak Biometric for Unobtrusive User Authentication. In: Proceedings of the 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '08). IEEE Computer Society, Washington, DC, USA, 1080–1087. doi:10.1109/IIH-MSP.2008.47 http://dx.doi.org/10.1109/IIH-MSP.2008.47

[48] Akkermans, A. H. M., Kevenaar, T. A. M., Schobben, W. E.: Acoustic Ear Recognition for Person Identification. In: Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AUTOID '05). IEEE Computer Society, Washington, DC, USA, 219–223. doi:10.1109/AUTOID.2005.11 http://dx.doi.org/10.1109/AUTOID.2005.11

[49] Yuan, L., Mu, Z., Xu, Z.: Using ear biometrics for personal recognition. In: Proceedings of the 2005 international conference on Advances in Biometric Person Authentication (IWBRS'05), Stan Z. Li, Zhenan Sun, Tieniu Tan, Sharath Pankanti, and Gérard Chollet (Eds.). Springer-Verlag, Berlin, Heidelberg, 221–228. doi:10.1007/11569947_28 http://dx.doi.org/10.1007/11569947_28

[50] Tresadern, P. A., Ionita, M. C., Cootes, T. F.: Real-Time Facial Feature Tracking on a Mobile Device. Int. J. Comput. Vision 96, 3 (February 2012), 280–289. doi:10.1007/s11263-011-0464-9 http://dx.doi.org/10.1007/s11263-011-0464-9

[51] Han, F., Hu, J., Alkhathami, M., Xi K.: Compatibility of photographed images with touch-based fingerprint verification software. Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on , vol., no., pp.1034,1039, 21–23 June 2011 doi: 10.1109/ICIEA.2011.5975739

[52] Sae-Bae, N., Ahmed, K., Isbister, K., Memon, N.: Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In Proc:

SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 977–986. doi:10.1145/2207676.2208543 http://doi.acm.org/10.1145/2207676.2208543

[53] Gafurov, D., Bours, P.: Improved Hip-Based Individual Recognition Using Wearable Motion Recording Sensor. FGIT-SecTech/DRBC 2010: 179–186. (2010)

[54] Park, K., R., Park, H., Kang, B. J., Lee, E. C., Jeong, D. S.: A study on iris localization and recognition on mobile phones. EURASIP J. Adv. Signal Process 2008, pages. doi:10.1155/2008/281943 http://dx.doi.org/10.1155/2008/281943

[55] Kang, B. J., Park, K. R.: A new multi-unit iris authentication based on quality assessment and score level fusion for mobile phones. Mach. Vision Appl. 21, 4 (June 2010), 541–553. doi:10.1007/s00138-009-0184-0 http://dx.doi.org/10.1007/s00138-009-0184-0

[56] Chang, T., Tsai C., Lin, J.: A graphical-based password keystroke dynamic authentication system for touch screen handheld mobile devices. J. Syst. Softw. 85, 5 (May 2012), 1157–1165. doi:10.1016/j.jss.2011.12.044 http://dx.doi.org/10.1016/j.jss.2011.12.044

[57] Campisi, P., Maiorana, E., Lo Bosco, M., Neri, A.: User authentication using keystroke dynamics for cellular phones. IET Signal Processing, vol. 3, no. 4, pp. 333–341. doi:10.1049/iet-spr.2008.0171

[58] Kim, D., Shin, J., Hong, K.: Teeth recognition based on multiple attempts in mobile device. J. Netw. Comput. Appl. 33, 3 (May 2010), 283–292 (2010). doi:10.1016/j.jnca.2009.12.016

[59] Wang, D., Li, J., Memik, G.: User identification based on finger-vein patterns for consumer electronics devices. IEEE Trans. on Consum. Electron. 56, 2 (May 2010), 799–804. doi:10.1109/TCE.2010.5506004 http://dx.doi.org/10.1109/TCE.2010.5506004

[60] R. C. Johnson, Walter J. Scheirer, Terrance E. Boult: Secure voice based authentication for mobile devices: Vaulted Voice Verification. CoRR abs/1212.0042 (2012).

[61] Iwano, K., Hirose, T., Kamibayashi, E., Furui, S.: Audio-visual person authentication using speech and ear images. In Proc. of Workshop on Multimodal User Authentication (2003).

[62] Faundez-Zanuy, M.: Data fusion in biometrics. Aerospace and Electronic Systems Magazine, IEEE , vol.20, no.1, pp.34,38, Jan. 2005 doi: 10.1109/MAES.2005.1396793

[63] Garcia-Salicetti, S., Mellakh, M. A., Allano, L., Dorizzi, B.: Multimodal biometric score fusion: The mean rule vs. support vector classifiers. In Proc. 13th European Signal Processing Conference, Sep. 4–8, 2005, Antalya, Turkey.

[64] Vildjiounaite, E., Mkel, S., Lindholm, M., Riihimki, R., Kyllnen, V., Mntyjrvi, J., Ailisto, H.: Unobtrusive multimodal biometrics for ensuring privacy and information security with personal devices. In Proc. of the 4th international conference on Pervasive Computing (PERVASIVE'06), Kenneth P. Fishkin, Bernt Schiele, Paddy Nixon, and Aaron Quigley (Eds.). Springer-Verlag, Berlin, Heidelberg, 187–201. doi:10.1007/11748625_12

[65] Kim, D., Hong, K.: Multimodal biometric authentication using teeth image and voice in mobile environment. IEEE Trans. on Consum. Electron. 54, 4 (November 2008), 1790–1797. doi:10.1109/TCE.2008.4711236

[66] Kim, D., Chung, K., Hong, K.: Person authentication using face, teeth and voice modalities for mobile device security. IEEE Trans. on Consum. Electron. 56, 4 (November 2010), 2678–2685. doi:10.1109/TCE.2010.5681156

[67] Shi, W., Yang, J., Jiang, Y., Yang, F., Xiong, Y.: SenGuard: Passive user identification on smartphones using multiple sensors. WiMob 2011: 141–148 doi: 10.1109/WiMOB.2011.6085412

[68] Tresadern, P., Cootes, T.F., Poh, N., Matejka, P., Hadid, A., Levy, C., McCool, C., Marcel, S.: Mobile Biometrics: Combined Face and Voice Verification for a Mobile Platform. Pervasive Computing, IEEE , vol.12, no.1, pp.79,87, Jan.-Mar. 2013 doi: 10.1109/MPRV.2012.54

[69] Hill, C. J.: Risk of Masquerade Arising from the Storage of Biometrics. B.Sc. Thesis, Australian National University, Australia, 2001.

[70] Jain, A.K., Ross, A., Uludag, U.: Biometric template security: Challenges and solutions. In Proceedings of European Signal Processing Conference (EUSIPCO), 2005.

[71] Nagar, A.: Biometric Template Security. Ph.D. Thesis, Michigan State University, USA, 2012.

[72] Casanova, J.G., vila, C.S., Sierra, A. de S., del Pozo, G.B.: Score optimization and template updating in a biometric technique for authentication in mobiles based on gestures. Journal of Systems and Software 84(11): 2013–2021 (2011) doi:10.1016/j.jss.2011.05.059

[73] Conti, M., Zachia-Zlatea, I., Crispo, B.: Mind how you answer me!: transparently authenticating the user of a smartphone when answering or placing a call. In Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS '11). ACM, New York, NY, USA, 249–259. doi:10.1145/1966913.1966945

[74] Giarmi, S., Magnusson, H.: Investigation of user acceptance for biometric verification/identification methods in mobile units. Master thesis, Stockholm University/Royal Institute of Technology, Sweden (2002).

[75] Trewin, S., Swart, C., Koved, L., Martino, J., Singh, K., Ben-David, S.: Biometric authentication on a mobile device: a study of user effort, error and task disruption. In Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC '12). ACM, New York, NY, USA, 159–168. doi:10.1145/2420950.2420976

[76] Derawi, M.O.: Accelerometer-Based Gait Analysis, A survey. Norwegian Information Security Conference, Nov 23–24, 2010, pp. 33–44.

[77] Brainard, J., Juels, A., Rivest, R. L., Szydlo, M., and Yung, M. 2006. Fourth-factor authentication: somebody you know. In Proceedings of the 13th ACM conference on Computer and communications security (CCS '06). ACM, New York, NY, USA, 168–178. DOI=10.1145/1180405.1180427 http://doi.acm.org/10.1145/1180405.1180427

[78] Schechter, S., Egelman, S., and Reeder, R. W. 2009. It's not what you know, but who you know: a social approach to last-resort authentication. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09). ACM, New York, NY, USA, 1983–1992. DOI=10.1145/1518701.1519003 http://doi.acm.org/10.1145/1518701.1519003

[79] Kumar, M., Garfinkel, T., Boneh, D., Winograd, T.: Reducing shoulder-surfing by using gaze-based password entry. In Proceedings of the 3rd symposium on Usable privacy and security (SOUPS '07). ACM, New York, NY, USA, 13–19. doi:10.1145/1280680.1280683

[80] Sasamoto, H., Christin, N., Hayashi, E.: Undercover: authentication usable in front of prying eyes. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08). ACM, New York, NY, USA, 183–192. doi:10.1145/1357054.1357085

[81] Perković, T., Li, S., Mumtaz, A., Khayam, S., A., Javed, Y., and Čagalj, M. 2011. Breaking undercover: exploiting design flaws and nonuniform human behavior. InProceedings of the Seventh Symposium on Usable Privacy and Security (SOUPS '11). ACM, New York, NY, USA, , Article 5 , 15 pages. DOI=10.1145/2078827.2078834 http://doi.acm.org/10.1145/2078827.2078834

[82] Liu, J., Zhong, L., Wickramasuriya, J., Vasudevan, V.: User evaluation of lightweight user authentication with a single tri-axis accelerometer. In Pro-

ceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services(MobileHCI '09). ACM, New York, NY, USA, , Article 15 , 10 pages. doi:10.1145/1613858.1613878

[83] De Luca, A., von Zezschwitz, E., Humann, H.: Vibrapass: secure authentication based on shared lies. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09). ACM, New York, NY, USA, 913–916. doi:10.1145/1518701.1518840

[84] Bianchi, A., Oakley, I., Kwon, D.S.: The secure haptic keypad: a tactile password system. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 1089–1092. doi:10.1145/1753326.1753488

[85] Kim, D., Dunphy, P., Briggs, P., Hook, J., Nicholson, J., W., Nicholson, J., and Olivier, P. 2010. Multi-touch authentication on tabletops. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 1093–1102. DOI=10.1145/1753326.1753489 http://doi.acm.org/10.1145/1753326.1753489

[86] Wiedenbeck, S., Waters, J., Sobrado, L., and Birget J. 2006. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In Proceedings of the working conference on Advanced visual interfaces (AVI '06). ACM, New York, NY, USA, 177–184. DOI=10.1145/1133265.1133303 http://doi.acm.org/10.1145/1133265.1133303

[87] De Luca, A., Hertzschuch, K., Hussmann, H.: ColorPIN: securing PIN entry through indirect input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 1103–1106. doi:10.1145/1753326.1753490

[88] Shirazi, A.S., Moghadam, P., Ketabdar, H., Schmidt, A.: Assessing the vulnerability of magnetic gestural authentication to video-based shoulder surfing attacks. In Proceedings of the SIGCHI Conference on Human Factors

in Computing Systems(CHI '12). ACM, New York, NY, USA, 2045–2048. doi:10.1145/2207676.2208352

[89] Azenkot, S., Rector, K., Ladner, R., Wobbrock, J.: PassChords: secure multi-touch authentication for blind people. In Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '12). ACM, New York, NY, USA, 159–166. doi:10.1145/2384916.2384945

[90] Lee, H., Lee, B., Park, J.: Force code: a new interaction technique using tangential force input. In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion (MobileHCI '12). ACM, New York, NY, USA, 77–82. doi:10.1145/2371664.2371680

[91] Ketabdar, H., Moghadam, P., Naderi, B., Roshandel, M.: Magnetic signatures in air for mobile devices. In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion (MobileHCI '12). ACM, New York, NY, USA, 185–188. doi:10.1145/2371664.2371705

[92] De Luca, A., Hang, A., Brudy, F., Lindner, C., Hussmann, H.: Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems(CHI '12). ACM, New York, NY, USA, 987–996. doi:10.1145/2207676.2208544

[93] Rabkin, A. 2008. Personal knowledge questions for fallback authentication: security questions in the era of Facebook. In Proceedings of the 4th symposium on Usable privacy and security (SOUPS '08). ACM, New York, NY, USA, 13–23. DOI=10.1145/1408664.1408667 http://doi.acm.org/10.1145/1408664.1408667

[94] Schechter, S., Brush, A. J. B., and Egelman, S. 2009. It's No Secret. Measuring the Security and Reliability of Authentication via Secret Questions. In Proceedings of the 2009 30th IEEE Symposium on Security and Pri-

vacy (SP '09). IEEE Computer Society, Washington, DC, USA, 375–390. DOI=10.1109/SP.2009.11 http://dx.doi.org/10.1109/SP.2009.11

[95] National Institute of Science and Technology NIST Special Publication 800 118: Guide to Enterprise Password Management (Draft): Recommendations of the National Institute of Standards and Technology (2009); http://csrc.nist.gov/publications/drafts/800-118/draft-sp800-118.pdf

[96] Sieger, H., Kirschnick, N., and Möller, S. 2010. User preferences for biometric authentication methods and graded security on mobile phones. In Proceedings of the Sixth Symposium on Usable Privacy and Security (SOUPS '10), July 14–16, 2010.

[97] Hayashi, E., Riva, O., Strauss, K., Brush, A. J. B., and Schechter, S. 2012. Goldilocks and the two mobile devices: going beyond all-or-nothing access to a device's applications. In Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS '12). ACM, New York, NY, USA, Article 2, 11 pages. DOI=10.1145/2335356.2335359 http://doi.acm.org/10.1145/2335356.2335359

[98] Hulsebosch, R. J., Bargh, M. S., Lenzini, G., Ebben, P. W. G., Iacob, S. M.: Context sensitive adaptive authentication. In: Proc. of the 2nd European conference on Smart sensing and context (EuroSSC'07), Gerd Kortuem, Joe Finney, Rodger Lea, and Vasughi Sundramoorthy (Eds.). Springer-Verlag, Berlin, Heidelberg, 93–109.

[99] Jakobsson, M., Shi, E., Golle, P., Chow, E.: Implicit authentication for mobile devices. In: Proc. of the 4th USENIX conference on Hot topics in security (HotSec'09). USENIX Association, Berkeley, CA, USA, 9–9.

[100] Roth, V., Richter, K., Freidinger, R.: A PIN-entry method resilient against shoulder surfing. In Proceedings of the 11th ACM conference on Computer and communications security (CCS '04). ACM, New York, NY, USA, 236–245. doi:10.1145/1030083.1030116

[101] Clarke, N.: Transparent User Authentication: Biometrics, RFID and Behavioral Profiling. Springer. (2011)

[102] Cranor, L., Garfinkel, S.: Security and Usability. O'Reilly Media, Inc. (2005)

[103] Jain, A., Ross, A., Nandakumar, K.: Introduction to Biometrics. Springer. (2011)

[104] Panjwani, S., Cutrell, E.: Usably secure, low-cost authentication for mobile banking. In Proceedings of the Sixth Symposium on Usable Privacy and Security (SOUPS '10). ACM, New York, NY, USA, Article 4 , 12 pages. doi:10.1145/1837110.1837116

[105] Tanviruzzaman, M., Ahamed, S. I., Hasan, C. S., O'brien, C.: ePet: When cellular phone learns to recognize its owner. SafeConfig (2009). doi: 10.1145/1655062.1655066

[106] Smart Phone Users, By The Numbers. http://visual.ly. (Accessed 22 April 2013)

[107] Mobile and Money. http://www.iab.net/media/file/iab-inmobi-viggle\%20mobile\%20financial\%20services-final.pdf. (Accessed 22 April 2013.)

[108] Mobile Health Trends for 2012. http://manhattanresearch.com/Images---Files/Data-Snapshots/Mobile-Health-Trends-for-2012.aspx (Accessed 22 April 2013.)

[109] How often and where are most smartphones lost. http://www.redmondpie.com/how-often-and-where-are-most-smartphones-lost-infographic/ (accessed February 6, 2014)

[110] Lost and Found. http://mozy.com/about/news/reports/lost-and-found (accessed May 6, 2014)

[111] Lost smartphone can cost you $37000. http://www.pcworld.com/article/240647/lost_smartphone_could_cost_you_37000.html (accessed May 6, 2014)

[112] Lorenz, K.: Man Meets Dog. Routledge, London and Newyork (2002)

[113] Sensors Overview. http://developer.android.com/guide/topics/sensors/sensors_overview.html (accessed May 6, 2014)

[114] Tanviruzzaman, M., Obrien, C., Rizia, R., Ahamed, S., I., and Smith, R., O. 2011. iFactotum: Sensor-Rich iPhone as a Versatile Tool. In Proceedings of the 3rd International Symposium on Quality of Life Technology (isQOLT 2011), Toronto, Canada, 2011.

[115] How often and where are most smartphones lost. http://www.redmondpie.com/how-often-and-where-are-most-smartphones-lost-infographic/ (accessed February 6, 2014)

[116] Indovina, M., Uludag, U., Snelick, R., Mink, A., Jain, A.: Multimodal Biometric Authentication Methods: A COTS Approach. In Proceedings of the Workshop on Multimodal User Authentication (MMUA '03).

[117] Wikipedia contributors, "Apple M7," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Apple_M7 (accessed April 27, 2014)

[118] Paek, J., Kim, J., Govindan, R.: Energy-efficient rate-adaptive GPS-based positioning for smartphones. In Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys '10). ACM, New York, NY, USA, 299–314. DOI=10.1145/1814433.1814463 http://doi.acm.org/10.1145/1814433.1814463

[119] Kerr, I., Barrigar, J., Burkell, J., Black, K.: Soft Surveillance, Hard Consent, http://www.idtrail.org/files/SoftSurveillanceHardConsent.pdf (accessed January 24, 2014)

[120] "Secure Enclave," http://support.apple.com/kb/HT5949 (accessed January 24, 2014)

[121] Gafurov, D., Snekkenes, E., Bours, P.: Spoof Attacks on Gait Authentication System. Trans. Info. For. Sec. 2, 3 (September 2007), 491–502. doi:10.1109/TIFS.2007.902030

[122] Cornillon, P., Guyader, A., Husson, F., Jegou, N., Josse, J., Kloareg, M., Matzner-Lober, E., and Rouviere, L. R for Statistics. CRC Press (2012)

[123] Fujinami, K., Kouchi, S., Xue, Y.: Design and Implementation of an On-body Placement-Aware Smartphone. In Proceedings of the 2012 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW '12). IEEE Computer Society, Washington, DC, USA, 69–74. doi:10.1109/ICDCSW.2012.52

[124] Juefei-Xu, F.; Bhagavatula, C.; Jaech, A.; Prasad, U.; Savvides, M.: Gait-ID on the move: Pace independent human identification using cell phone accelerometer dynamics. Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on, pp.8,15, 23–27 Sept. 2012 doi: 10.1109/BTAS.2012.6374552

[125] Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Cell phone-based biometric identification. Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on, pp.1,7, 27–29 Sept. 2010 doi: 10.1109/BTAS.2010.5634532

[126] Mohammad Omar Derawi, Claudia Nickel, Patrick Bours, and Christoph Busch. 2010. Unobtrusive User-Authentication on Mobile Phones Using Biometric Gait Recognition. In Proceedings of the 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '10). IEEE Computer Society, Washington, DC, USA, 306–311. doi:10.1109/IIHMSP.2010.83

[127] Nickel, C., Busch, C., Rangarajan, S., Mobius, M.: Using Hidden Markov Models for accelerometer-based biometric gait recognition. Signal Processing and its Applications (CSPA), 2011 IEEE 7th International Colloquium on, pp.58,63, 4–6 March 2011 doi: 10.1109/CSPA.2011.5759842

[128] Nickel, C., Wirtl, T., Busch, C.: Authentication of Smartphone Users Based on the Way They Walk Using k-NN Algorithm. Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on, pp.16–20, 18–20 July 2012 doi: 10.1109/IIH-MSP.2012.11

[129] Priyantha, B., Lymberopoulos, D., Liu, J.: EERS: Energy Efficient Responsive Sleeping on Mobile Phones. In Proc. of PhoneSense 2010, Zurich, pp. 1–5, Nov. 2010.

[130] Lu, H., Yang, J., Liu, Z., Lane, N.D., Choudhury, T., Campbell, A.T.: The Jigsaw continuous sensing engine for mobile phone applications. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10). ACM, New York, NY, USA, 71–84. doi:10.1145/1869983.1869992

[131] Baddeley, A.: Your Memory: A User's Guide. Firefly Books (2004)

[132] Craik, F. I. M. and Lockhart, S.: Levels of Processing: A Framework for Memory Research. Journal of Verbal Learning & Verbal Behavior, Vol. 11(6) (December 1972), pp. 671–684

[133] Willingham, D. B., Nissen, M. J., and Bullemer, P. 1989. On the Development of Procedural Knowledge. Jouranl of Experimental Psychology: Learning, Memory, and Cognition. 1989, Nov; 15(6):1047 – 1060.

[134] Shadnehr, R. and Holcomb, H. H. 1997. Neural Correlates of Motor Memory Consolidation. Science, 8 August 1997: Vol. 277 no. 5327 pp. 821 – 825 DOI: 10.1126/science.277.5327.821

[135] Karni, A., Meyer, G., Rey-Hipolito, C., Jezzard, P., Adams, M., M., Turner, R., and Ungerleider, L. G. 1998. The Acquisition of Skilled Motor Performance: Fast and Slow Experience-driven Changes in Primary Motor Cortex. In Proceesings of National Academy of Sciences of the USA, Vol. 95, pp. 861–868, February 1998.

[136] Willingham, D. B., Wells, L. A., Farrell, J. M., and Stemwedel, M. 2000. Implicit motor sequence learning is represented in response locations. Memory & Cognition, 2000, 28(3), 366–375

[137] Krakauer, J. W. and Shadmehr, R. 2006. Consolidation of motor memory. Trends Neuroscience, Jan 2006; 29(1): 58–64

[138] Wolpert, D. M., Ghahramani, Z., and Flanagan, J. R. 2001. Perspectives and problems in motor learning. Trends in Cognitive Sciences , Volume 5 , Issue 11 , 487 – 494

[139] Koch, I. and Hoffmann, J. 2000. Patterns, chunks, and hierarchies in serial reaction-time tasks. Psychological Research, March 2000, Vol. 63, Issue 1, pp 22–35

[140] Sakai, K., Kitaguchi, K., and Hikosaka, O. 2003. Chunking during human visuomotor sequence learning. Experimental Brain Research (2003) 152:229–242; DOI 10.1007/s00221-003-1548-8

[141] Robertson, E. M. 2007. The Serial Reaction Time Task: Implicit Motor Skill Learning? The Journal of Neuroscience, September 19, 2007 · 27(38):1007310075

[142] Wikipedia contributors, "Leet," Wikipedia, The Free Encyclopedia, `http://en.wikipedia.org/wiki/Leet` (accessed April 27, 2014)

[143] Accusplit. `http://www.accusplit.com/ProductDetail.aspx?ProductId=37` (Accessed on May 21, 2014)

[144] Mobile Metronome. `https://play.google.com/store/apps/details?id=gabriel.metronome` (accessed April 27, 2014)

[145] Kahneman, D.: Thinking, Fast and Slow. Farrar, Straus and Giroux, New York (2011)

[146] Tulving, E. and Craik, F. I. M. The Oxford Handbook of Memory, Oxford University Press (2000)

[147] Lazar, J., Feng, J. H., and Hochheiser, H.: Research Methods in Human-Computer Interaction, Wiley (2010)

[148] Typing Speed Test. http://www.typingtest.com/test.html?minutes=1&textfile=aesop.txt&getfocus=1&start.x=58&start.y=30

[149] Kosinski, R. J. 2013. A Literature Review on Reaction Time. http://biae.clemson.edu/bpc/bp/Lab/110/reaction.htm#Type\%20of\%20Stimulus (accessed April 28, 2014)

[150] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. Introduction to Algorithms. The MIT Press, 2009.

[151] Shapiro-Wilk Normality Test. http://stat.ethz.ch/R-manual/R-devel/library/stats/html/shapiro.test.html (accessed April 28, 2014)

[152] F Test to Compare Two Variances. http://stat.ethz.ch/R-manual/R-patched/library/stats/html/var.test.html (accessed April 28, 2014)

[153] Correlation, Variance and Covariance (Matrices). http://stat.ethz.ch/R-manual/R-patched/library/stats/html/cor.html (accessed May 12, 2014)

[154] Kolmogorov-Smirnov Tests. http://stat.ethz.ch/R-manual/R-patched/library/stats/html/ks.test.html (accessed May 12, 2014)

[155] Text file with words. https://github.com/eneko/data-repository/blob/master/data/words.txt (accessed April 28, 2014)

[156] PIN analysis. http://www.datagenetics.com/blog/september32012/ (accessed April 28, 2014)

[157] Text file with passwords. http://dazzlepod.com/uniqpass/

[158] Gonzalo Navarro. 2001. A guided tour to approximate string matching. ACM Comput. Surv. 33, 1 (March 2001), 31–88. DOI=10.1145/375360.375365 http://doi.acm.org/10.1145/375360.375365

[159] Wikipedia contributors, "Dice," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Dice (accessed April 28, 2014)

[160] Growth of security market in size. http://www.gartner.com/newsroom/id/2512215 (accessed April 26, 2014)

[161] Wikipedia contributors, "Transport Layer Security," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Secure_Sockets_Layer (accessed April 27, 2014)

[162] Wikipedia contributors, "Message authentication code," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Message_authentication_code (accessed April 27, 2014)

[163] Wikipedia contributors, "Denial-of-service attack," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Denial-of-service_attack (accessed April 27, 2014)

[164] Wikipedia contributors, "Firewall (computing)," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Firewall_(computing) (accessed April 27, 2014)

[165] Wikipedia contributors, "Quality of service," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Quality_of_service#Over-provisioning (accessed April 27, 2014)

[166] Defending against denial of service. https://s3.amazonaws.com/access.3cdn.net/7ba8fcbc60c8271c1c_fdm6i2vsa.pdf (accessed April 27, 2014)

[167] Non-Repudiation in the Digital Environment. http://firstmonday.org/ojs/index.php/fm/article/view/778/687 (accessed April 27, 2014)

[168] Wikipedia contributors, "Declarative memory," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Declarative_memory (accessed April 27, 2014)

[169] Barbosa, F. F. and Albuquerque, F. S. 2008. Effect of time-of-day of training on explicit memory. Time-of-day of training and memory. Brazilian Journal of Medical and Biological Research (2008) 41: 477–481; ISSN 0100-879X

[170] Gerstner, J. R. and Yin, J. C. P. 2010. Circadian rhythms and memory formation. Nature Reviews, Neuroscience; 2010 Aug; 11(8): 577–588. DOI: 10.1038/nrn2881.