2013

# Towards Secure and Usable Leakage-Resilient Password Entry

Qiang YAN
*Singapore Management University*, qiang.yan.2008@phdis.smu.edu.sg

## Citation

YAN, Qiang. Towards Secure and Usable Leakage-Resilient Password Entry. (2013).
**Available at:** https://ink.library.smu.edu.sg/etd_coll/91

# Towards Secure and Usable Leakage-Resilient Password Entry

by

**YAN Qiang**

Submitted to School of Information Systems in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Information Systems

**Dissertation Committee:**

Robert DENG Huijie (Supervisor / Chair)
Professor of Information Systems
Singapore Management University

Yingjiu LI (Co-supervisor)
Associate Professor of Information Systems
Singapore Management University

Debin GAO
Assistant Professor of Information Systems
Singapore Management University

Feng BAO
Director of Security and Privacy Lab
Huawei Technologies Co., Ltd.

Singapore Management University
2013

# Towards Secure and Usable Leakage-Resilient Password Entry

YAN Qiang

## Abstract

Password leakage is one of the most common security threats for pervasive password-based user authentication. The design of a secure and usable password entry against password leakage remains a challenge since twenty year ago when the first academic proposal attempted to address it. This dissertation focuses on investigating the difficulty in designing leakage-resilient password entry (LRPE) schemes and exploring the feasibility of constructing secure and usable LRPE schemes with the assistance of state-of-the-art technology.

The first work in this dissertation reveals the infeasibility of designing practical LRPE schemes in the absence of trusted devices by investigating the inherent tradeoff between security and usability in LRPE design. We start with demonstrating that most of the existing LRPE schemes without using trusted devices are subject to two types of generic attacks - brute force and statistical attacks, whose power has been underestimated in the literature. In order to defend against these two generic attacks, we introduce five design principles that are necessary to achieve leakage resilience in the absence of trusted devices. We show that these attacks cannot be effectively mitigated without significantly sacrificing the usability of LRPE schemes. To better understand the tradeoff between security and usability of LRPE schemes, we further propose a quantitative analysis framework on usability costs of password entry schemes based on experimental psychology. Our analysis shows that a secure LRPE scheme in practical settings always imposes a considerable amount of cognitive workload on its users, which indicates the inherent limitations of such schemes and in turn implies that an LRPE scheme has to incorporate certain trusted device in order to be both secure and usable.

Following the first work, we further explore the feasibility of designing practical LRPE schemes by analyzing the existing LRPE schemes that utilize trusted devices. We develop a broad set of design metrics which cover three aspects in evaluating LRPE schemes, including quantitative usability costs with specified security strength, built-in security, and universal accessibility. We apply these design metrics on existing LRPE schemes, revealing that all the schemes have limitations, which may explain why none of them are widely adopted. However, our further analysis indicates that it is possible to overcome these limitations by improving the design according to the proposed metrics.

Guided by these design metrics, we propose a secure and usable LRPE scheme leveraging on the touchscreen feature of mobile devices. These devices provide additional features such as touchscreen that are not available in the traditional settings, which makes it possible to achieve both security and usability objectives that are difficult to achieve in the past. Our scheme named CoverPad achieves leakage resilience while retaining most benefits of legacy passwords. The usability of CoverPad is evaluated with an extended user study which includes additional test conditions related to time pressure, distraction, and mental workload. These test conditions simulate common situations for a password entry scheme used on a daily basis, which have not been evaluated in the prior literature. The results of our user study show the impacts of these test conditions on user performance as well as the practicability of the proposed scheme.

This dissertation makes contributions on understanding and solving the problem of designing secure and usable LRPE schemes. The proposed design principles, design metrics, analysis and evaluation methodologies are applicable to not only LRPE schemes but also generic user authentication schemes, which provide useful insights for the field of user authentication research. The proposed scheme has been implemented as a prototype, which can be used to effectively defend against password leakage during password entry.

# Table of Contents

**1 Introduction 1**

1.1 Identify the Limitations . . . . . . . . . . . . . . . . . . . . . . . . 2

1.2 Explore the Feasibility . . . . . . . . . . . . . . . . . . . . . . . . 4

1.3 Construct a Practical Design . . . . . . . . . . . . . . . . . . . . . 5

1.4 Contributions and Organization . . . . . . . . . . . . . . . . . . . . 6

**2 Literature Review 8**

**3 On Limitations of Designing Leakage-Resilient Password Entry: Attacks, Principles and Usability 12**

3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

3.2 Definitions and Threat Model . . . . . . . . . . . . . . . . . . . . . 15

3.2.1 Leakage-Resilient Password Entry . . . . . . . . . . . . . . 15

3.2.2 Threat Model and Experimental Settings . . . . . . . . . . 17

3.3 Brute Force Attack and Its Defense Principles . . . . . . . . . . . . 19

3.3.1 Attack Strategy . . . . . . . . . . . . . . . . . . . . . . . 19

3.3.2 P1: Large Root Secret Space Principle . . . . . . . . . . . . 20

3.3.3 P2: Large Round Secret Space Principle . . . . . . . . . . . 22

3.4 Statistical Attack and Its Defense Principles . . . . . . . . . . . . . 24

3.4.1 Attack Strategy . . . . . . . . . . . . . . . . . . . . . . . 24

3.4.2 P3: Uniform Distributed Challenge Principle . . . . . . . . 26

3.4.3 P4: Large Decision Space or Indistinguishable Individual Principle . . . . . . . . . . . . . . . . . . . . . . . . . . 28
3.4.4 P5: Indistinguishable Correlation Principle . . . . . . . . . 31
3.5 Usability Costs of Defense Principles . . . . . . . . . . . . . . . 34
3.6 Quantitative Tradeoff Analysis . . . . . . . . . . . . . . . . . . . . . . 36
3.6.1 Atomic Cognitive Operations . . . . . . . . . . . . . . . . . 37
3.6.2 Quantitative Analysis Framework . . . . . . . . . . . . . . 40
3.6.3 High Security at Cost of Heavy Cognitive Demand . . . . . 42
3.7 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

**4 Usable Leakage-Resilient Password Entry: Challenges and Design Metrics 46**

4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
4.2 LRPE Problem Overview . . . . . . . . . . . . . . . . . . . . . . . 48
4.2.1 Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . 48
4.2.2 Threat Model . . . . . . . . . . . . . . . . . . . . . . . . . 49
4.2.3 Common Design Paradigms . . . . . . . . . . . . . . . . . . 51
4.2.4 Design Metrics Overview . . . . . . . . . . . . . . . . . . . 52
4.3 Relations between Security Strength and Usability Costs . . . . . . 53
4.3.1 Password Space and Memory Effort . . . . . . . . . . . . . 54
4.3.2 Leakage Resistance and Cognitive Workload . . . . . . . . 56
4.3.3 Effectiveness and Costs of Interaction Channels . . . . . . . 60
4.4 Built-in Security . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
4.4.1 Inconsistency with Personal Habits . . . . . . . . . . . . . 62
4.4.2 Violations of Social Norms . . . . . . . . . . . . . . . . . . 64
4.5 Universal Accessibility . . . . . . . . . . . . . . . . . . . . . . . . 65
4.5.1 Beneficiary Scope . . . . . . . . . . . . . . . . . . . . . . . 65
4.5.2 Device Availability . . . . . . . . . . . . . . . . . . . . . . 65
4.5.3 Environmental Adaptation . . . . . . . . . . . . . . . . . . 68

4.6 Using the Metrics: Evaluation of Existing LRPE Schemes . . . . . 68

4.6.1 Paradigm Level Analysis . . . . . . . . . . . . . . . . . . . . 71

4.6.2 Scheme Level Analysis . . . . . . . . . . . . . . . . . . . . . 73

4.7 Challenges behind the LRPE Problem . . . . . . . . . . . . . . . . 74

4.8 Implications and Limitations . . . . . . . . . . . . . . . . . . . . . . 77

4.8.1 Implication of a Practical LRPE Scheme . . . . . . . . . . . 77

4.8.2 Other Metrics . . . . . . . . . . . . . . . . . . . . . . . . . . 78

4.8.3 Limitations . . . . . . . . . . . . . . . . . . . . . . . . . . . 79

4.9 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81

**5 Designing Leakage-Resilient Password Entry on Touchscreen Mobile Devices 82**

5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82

5.2 Threat Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84

5.3 CoverPad Design . . . . . . . . . . . . . . . . . . . . . . . . . . . 87

5.3.1 Design Objectives . . . . . . . . . . . . . . . . . . . . . . . 87

5.3.2 Conceptual Design . . . . . . . . . . . . . . . . . . . . . . . 88

5.3.3 Implementation Variants . . . . . . . . . . . . . . . . . . . . 90

5.4 Security Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . 92

5.4.1 External Eavesdropping Attacks . . . . . . . . . . . . . . . . 92

5.4.2 Side-channel Attacks . . . . . . . . . . . . . . . . . . . . . . 93

5.5 Usability Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . 93

5.5.1 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . 93

5.5.2 Simulating Various Test Conditions . . . . . . . . . . . . . . 96

5.5.3 Learning Curve . . . . . . . . . . . . . . . . . . . . . . . . . 98

5.5.4 Experimental Results . . . . . . . . . . . . . . . . . . . . . . 99

5.5.5 Statistical Test Results . . . . . . . . . . . . . . . . . . . . . 108

5.5.6 Comparison with Legacy Passwords . . . . . . . . . . . . . . 109

5.6 Other Practical Issues and Limitations . . . . . . . . . . . . . . . . 111

5.6.1 Eavesdropping Attacks . . . . . . . . . . . . . . . . . . . . . . 111

5.6.2 Device Screen Size . . . . . . . . . . . . . . . . . . . . . . . 112

5.6.3 Limitations . . . . . . . . . . . . . . . . . . . . . . . . . . . 112

5.7 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113

**6 Dissertation Conclusion and Future Work 115**

6.1 Summary of Contribution . . . . . . . . . . . . . . . . . . . . . . . 115

6.2 Future Direction . . . . . . . . . . . . . . . . . . . . . . . . . . . 116

# List of Figures

3.1 Demonstration of a typical LRPE scheme . . . . . . . . . . . . . . . 16

3.2 The average number of valid candidates shrinks for the Undercover scheme. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

3.3 The average number of valid candidates shrinks for the PAS scheme. 24

3.4 Definition and example for decision path . . . . . . . . . . . . . . . . 26

3.5 The average false positive rate decreases for the high-complexity CAS scheme. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

3.6 Informal proof for the strength of multi-dimensional counting . . . . 31

3.7 The pair-based score distribution is distorted for the SecHCI scheme 33

4.1 Examples of LRPE schemes following common design paradigms . 51

4.2 Major design factors in LRPE schemes . . . . . . . . . . . . . . . . . 54

4.3 Password composition of the Undercover scheme . . . . . . . . . . 55

4.4 The usage of the PhoneLock scheme . . . . . . . . . . . . . . . . . . 60

4.5 User interface in the ShieldPin scheme . . . . . . . . . . . . . . . . . 64

4.6 The usage of the PressureGrid scheme . . . . . . . . . . . . . . . . . 66

4.7 Visual redundancy in the PAS scheme . . . . . . . . . . . . . . . . . 67

4.8 User interface of the HapticKeypad scheme . . . . . . . . . . . . . 69

4.9 SUR calculation for the CuePin scheme . . . . . . . . . . . . . . . . 73

4.10 A layered view of potential attacks against an LRPE scheme . . . . 75

5.1 Attack scenarios . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85

5.2 Conceptual design of CoverPad . . . . . . . . . . . . . . . . . . . . . 88

5.3 The hand-shielding gesture and its effectiveness . . . . . . . . . . . 89

5.4 Demonstration of three implementation variants . . . . . . . . . . . 90

5.5 Timing deviations and distributions for entering each password element . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94

5.6 Learning curve of CoverPad . . . . . . . . . . . . . . . . . . . . . . 99

5.7 Average login time, success rate, and edit distance under the normal condition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 101

5.8 Impact of time pressure . . . . . . . . . . . . . . . . . . . . . . . . 103

5.9 Impact of distraction . . . . . . . . . . . . . . . . . . . . . . . . . 104

5.10 Impact of mental workload . . . . . . . . . . . . . . . . . . . . . . 105

5.11 Accuracy rate of performing secondary tasks . . . . . . . . . . . . 107

5.12 Total number of times for each participant to press the "*show my password*" button . . . . . . . . . . . . . . . . . . . . . . . . . . 107

5.13 Perception of participants . . . . . . . . . . . . . . . . . . . . . . . 108

5.14 Conceptual demonstration on a small screen device . . . . . . . . . 113

# List of Tables

3.1 Tradeoff comparison of representative leakage-resilient password entry schemes for their default parameters. . . . . . . . . . . . . . . . 42

3.2 Detailed computation of cognitive workload for representative leakage-resilient password entry schemes . . . . . . . . . . . . . . . . . . . . 43

4.1 Evaluation and comparison of representative leakage-resilient password entry schemes . . . . . . . . . . . . . . . . . . . . . . . . . . 70

5.1 Short names for test conditions . . . . . . . . . . . . . . . . . . . . . 97

5.2 The results of statistical tests on login time . . . . . . . . . . . . . . . 109

5.3 Evidence for the ceiling effect in statistical tests on login accuracy . 109

5.4 Comparison between CoverPad and legacy passwords using LRPE design metrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110

5.5 Comparison between CoverPad and legacy passwords using usability-deployability-security metrics . . . . . . . . . . . . . . . . . . . . . 111

# Acknowledgments

I would like to thank Professor Steven MILLER, Professor Robert DENG, Associate Professor Yingjiu LI, Assistant Professor Debin GAO and Doctor Feng BAO for their guidance in completing my dissertation.

I also thank my friends HAN Jin and KOH Noi Sian for the research collaboration, their friendship, and their encouragement.

Finally, I would like to thank my parents, who are always supporting me and encouraging me with their best wishes.

# Dedication

I dedicate my dissertation work to my loving parents, DONG Yueqin and YAN Huizhong. Thanks, Mum and Daddy.

# List of Publications

## Conference Papers

**Q. Yan**, J. Han, Y. Li, J. Zhou, and R. H. Deng. Designing Leakage-Resilient Password Entry on Touchscreen Mobile Devices. In *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security*, China, 2013.

J. Han, S. M. Kywe, **Q. Yan**, F. Bao, R. H. Deng, D. Gao, Y. Li, and J. Zhou. Launching Generic Attacks on iOS with Approved Third-Party Applications. In *Proceedings of the 11th International Conference on Applied Cryptography and Network Security*, Canada, 2013.

Y. Li, Y. Li, **Q. Yan**, and R. H. Deng. Think Twice before You Share: Analyzing Privacy Leakage under Privacy Control in Online Social Network (short paper). In *Proceedings of the 7th International Conference on Network and System Security*, Spain, 2013.

J. Han, **Q. Yan**, D. Gao, J. Zhou, and R. H. Deng. Comparing Mobile Privacy Protection through Cross-Platform Applications. In *Proceedings of the 20th Annual Network & Distributed System Security Symposium*, USA, 2013.

**Q. Yan**, J. Han, Y. Li, and R. H. Deng. On Limitations of Designing Leakage-Resilient Password Systems: Attacks, Principles and Usability. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, USA, 2012.

J. Han, **Q. Yan**, D. Gao, and R. H. Deng. On Detection of Erratic Arguments. In *Proceedings of the 7th International ICST Conference on Security and Privacy in Communication Networks*, United Kingdom, 2011.

**Q. Yan**, J. Han, Y. Li, R. H. Deng, and T. Li. A Software-Based Root-of-Trust Primitive on Multicore Platforms. In *Proceedings of 6th ACM Symposium on Information, Computer and Communications Security*, Hong Kong, 2011.

**Q. Yan**, R. H. Deng, Z. Yan, Y. Li, and T. Li. Pseudonym-based RFID Discovery Service to Mitigate Unauthorized Tracking in Supply Chain Management. In *Proceedings of 2nd International Symposium on Data, Privacy and E-Commerce*, USA, 2010.

**Q. Yan**, Y. Li, T. Li, and R. H. Deng. Insights into Malware Detection and Prevention on Mobile Phones. In *Proceedings of International Conference on Security Technology*, Korea, 2009.

**Q. Yan**, Y. Li, T. Li, and R. H. Deng. A Comprehensive Study for RFID Malwares on Mobile Devices. In *the 5th Workshop on RFID Security*, Taiwan, 2009.

# Journal Papers

**Q. Yan**, Y. Li, and R. H. Deng. Anti-Tracking in RFID Discovery Service for Dynamic Supply Chain Systems. *International Journal of RFID Security and Cryptography*, Informatics Society, 1(1/2), 2012, pp. 25-35.

**Q. Yan**, R. H. Deng, Y. Li, and T. Li. On the Potential of Limitation-Oriented Malware Detection and Prevention on Mobile Phones. *International Journal of Security and Its Applications*, 4(1), 2010, pp. 21-30.

# Book Chapters

**Q. Yan**, Y. Li, and R. H. Deng. Malware Protection on RFID-Enabled Supply Chain Management Systems in the EPCglobal Network. *Advanced Security and Privacy for RFID Technologies*, chapter 10, IGI Global, USA, 2013.

# Chapter 1

# Introduction

The wide adoption of computing systems not only transforms many physical assets into virtual assets but also creates new assets that only exist in the virtual world. Preventing unauthorized access to these assets is one of the major themes of information security, where user authentication is the key mechanism to guarantee that only legitimate users can access protected assets. Passwords have been the most pervasive means for user authentication since the advent of computers. Compared to their alternatives, such as biometrics and smartcards, passwords are much easier and cheaper to create, update, and revoke. However, the use of passwords has intrinsic problems. Among them, password leakage is one of the most common security threats [49]. Password leakage can be caused by various attacks including malware, key logger, and hidden camera. The consequence of password leakage could be catastrophic, as password-based authentication has been widely used for financial services, social networks, and other valuable services.

The design of a secure and usable password entry against password leakage remains a challenge since twenty year ago when the first academic proposal [52] attempted to address it. The difficulty comes from the fact that passwords are widely used not only within organizations such as governments and companies, but also by every individual who uses a computing system. Therefore, unlike early security systems [60, 2] that are mainly designed to be operated by well-trained users with dedicated devices, a secure and usable leakage-resilient password entry (LRPE) scheme brings the following challenges: 1) The users may not have sufficient knowledge and skills due to cognitive limitations or other conditions; 2) The devices for system deployment may not implement all the required features due to manufacturing, management, or other costs. Both restrictions have to be properly addressed in an LRPE scheme intended for practical use.

This dissertation investigates the difficulty in designing LRPE schemes and exploring the feasibility of constructing secure and usable LRPE schemes with the assistance of state-of-the-art technology. We first identify the inherent limitations of designing LRPE schemes, then establish the key design metrics that affect the practicability of LRPE schemes, and finally develop a secure and usable LRPE scheme leveraging on the touchscreen feature of mobile devices. The details of these works are introduced as follows.

## 1.1 Identify the Limitations

The first work in this dissertation reveals the infeasibility of designing practical LRPE schemes in the absence of trusted devices by investigating the inherent tradeoff between security and usability in LRPE design. Compared to an LRPE scheme, legacy passwords that are used pervasively ask a user to directly input his entire plaintext password recalled from the user's memory, so that an observation of a single authentication session is sufficient to capture the password. In order to prevent password leakage during password entry, a user needs to input the password indirectly, which imposes an extra burden on the user and creates a tradeoff between security and usability. How to design a password entry scheme that minimizes password leakage and is still easy to use is the fundamental problem in LRPE design.

It was an interesting problem of designing a secure and usable LRPE scheme without using any trusted devices. The technical challenge behind this problem is to handle the capability asymmetry between user and adversary. An adversary may use a hidden camera or malicious software to record complete interactions between a user and his computer and then analyze the data with powerful machines. Many LRPE schemes [35, 48, 71, 72, 78, 10, 6] have been proposed to defend against this type of password leakage attacks without utilizing any trusted devices. However, as we will demonstrate later, all these existing proposals with acceptable usability are vulnerable to either or both types of generic attacks: brute force attack and statistical attack. We notice that these two generic attacks are different from other specific attacks [32, 47]. They cannot be easily defended without significantly sacrificing the scheme's usability, which implies inherent limitations of LRPE schemes without using trusted devices. In order to defend against these attacks, we introduced five design principles which should be followed to achieve leakage resilience. Using counterexamples, we show that an LRPE scheme can be easily broken when these principles are violated.

To further understand the tradeoff between security and usability in the design of LRPE schemes, we propose for the first time a quantitative analysis framework on usability costs of LRPE schemes. This framework decomposes the process of human-computer authentication into atomic cognitive operations. Performance data of average human-beings reported in psychology literatures [65, 27, 21, 70, 22, 55, 57, 19, 73, 74, 36, 17, 34] are used to estimate usability costs of existing LRPE schemes [35, 48, 71, 72, 78, 10, 6]. Our analysis results are consistent with the experimental results reported in the original literatures, while the hidden costs previously not addressed are identified. Our results show that a secure LRPE scheme in practical settings [35, 6] always leads to a considerable amount of cognitive workload, which explains why some of the existing LRPE schemes require extremely long authentication time and have high authentication error rate. This limitation has not been, and will not be easily solved in the design of LRPE schemes in the absence of trusted devices.

## 1.2 Explore the Feasibility

Under the limitations discovered in the first work, the second work in this dissertation explores the feasibility of designing practical LRPE schemes with the assistance of trusted devices. A trusted device forms a secure channel between user and server, which ensures that at least part of the authentication process should be invisible to an adversary so as to prevent password leakage while maintaining acceptable usability in realistic settings. However, despite of many prior efforts [44, 61, 23, 25, 42, 13, 12], there is still no practical and widely adopted solution today. This raises a question on the practicability of adopting a secure channel in password-based authentication.

In this study, we make the first attempt to systematically investigate the challenges of designing usable LRPE schemes even when a secure channel is available. We first formalize the authentication process of LRPE schemes and classify existing schemes into three common design paradigms. We then develop a broad set of design metrics, which cover three aspects in evaluating LRPE schemes, including quantitative usability costs with specified security strengths, built-in security, and universal accessibility. Unlike traditional evaluation metrics, the proposed metrics are designed to identify the potential limitations of an LRPE scheme in the design phase before carrying out user studies.

We apply our design metrics to existing LRPE schemes, which reveals and identifies their limitations. The major limitations include: 1) the requirement of an uncommon device feature, 2) the inoperability in certain common scenarios, and 3) the lack of trusted execution environment. This partially explains why none of these schemes are widely adopted nowadays. However, it does not necessarily imply that it is infeasible to design an LRPE scheme that is both secure and practical. Our further analysis indicates that it is possible to overcome these limitations by improving the design according to the proposed metrics.

## 1.3 Construct a Practical Design

Guided by the metrics developed in the second work, the third work in this dissertation proposes a secure and usable LRPE scheme leveraging on the touchscreen feature of mobile devices. These devices provide additional features such as touchscreen that are not available in the traditional settings, which makes it possible to achieve both security and usability objectives that are difficult to achieve in the past.

Our scheme named CoverPad achieves leakage resilience of password entry while retaining most benefits of legacy passwords. Leakage resilience is achieved by utilizing the gesture detection feature of touchscreen in forming a cover for user inputs. This cover is used to safely deliver hidden messages, which break the correlation between the underlying password and the interaction information observable to an adversary. From the other perspective, our scheme is also designed to retain the benefits provided by legacy passwords. This requirement is critical, as Bonneau et al. [15] concluded that any user authentication is unlikely to gain traction if it does not retain comparable benefits of legacy passwords. Our scheme approaches this requirement by involving only intuitive cognitive operations and requiring no extra devices in the design.

We implement three variants of CoverPad and evaluate them with an extended user study. This study includes additional test conditions related to time pressure, distraction, and mental workload. These test conditions simulate common situations for a daily-used password entry scheme, which have not been evaluated in the prior literature. We design new experiments to examine their influence based on previous work in psychology literature [40, 22, 38]. Experimental results show the influence of these conditions on user performance and the practicability of our proposed scheme.

## 1.4 Contributions and Organization

To summarize, the following contributions have been made in this dissertation:

- We analyze and demonstrate the effectiveness of two types of generic attacks, brute force and statistical attacks, against leakage-resilient password entry (LRPE) schemes. We propose two statistical attack techniques, probabilistic decision tree and multi-dimensional counting, and show their effectiveness against existing schemes. We introduce five principles that are necessary to mitigate brute force and statistical attacks. We use typical existing LRPE proposals as counterexamples to show that an adversary can easily obtain a user's password in the schemes violating our principles. We establish the first quantitative analysis framework on usability costs of the existing LRPE schemes. This framework utilizes the performance models of atomic cognitive operations in authentication to estimate usability costs. Our analysis result shows that there is a strong tradeoff between security and usability in the existing LRPE schemes. It implies that an unaided human may not be competent enough to effectively use a secure LRPE scheme in practical settings; in other words, it is inevitable to incorporate certain trusted device in LRPE design.

- We identify the challenges of designing usable LRPE schemes even with the presence of trusted devices, and classify existing LRPE schemes into three common design paradigms. We develop a broad set of design metrics for LRPE schemes, which defines quantitative relation between security and usability, and extends the scope of security and usability to include built-in security and universal accessibility. We apply the proposed metrics on existing LRPE schemes and reveal that all the schemes have limitations that could be further improved. Our analysis provides not only a systematic understanding on existing LRPE schemes, but also a useful guide for the future research in this area.

- We propose a secure and usable LRPE scheme named CoverPad to protect password entry on touchscreen mobile devices. It achieves leakage resilience and retains most benefits of legacy passwords by involving only intuitive cognitive operations and requiring no extra devices. We implement three variants of CoverPad to address different user preferences. Our user study shows the practicability of these variants. We extend user study methodology to examine the influence of various additional test conditions. Among these conditions, time pressure and mental workload are shown to have significant impacts on user performance. Therefore, it is recommended to include these conditions in the evaluation of user authentication schemes in the future.

The reminder of this dissertation is organized as follows: Chapter 2 is a literature review which examines closely related research on leakage-resilient password entry (LRPE). Chapter 3 investigates the limitations of designing LRPE schemes. Chapter 4 studies the feasibility of designing practical LRPE schemes by analyzing the existing LRPE schemes that utilize trusted devices and establishes the key design metrics that affect the practicability of LRPE schemes. Chapter 5 provides a secure and usable LRPE scheme leveraging on the touchscreen feature of mobile devices. Finally, Chapter 6 summarizes the contributions of this dissertation.

# Chapter 2

# Literature Review

As one of the most important security tools of modern society, password-based user authentication has been extensively investigated. We summarize the closely related research work from the following aspects: attacks, principles, tradeoff analysis, design metrics, and system designs for Leakage-Resilient Password Entry (LRPE).

Most of proposed LRPE schemes have been broken. The recent works on representative attack and analysis include: Golle and Wagner proposed the SAT attack [32] against the CAS schemes [71]; Li et al. demonstrated the brute-force attack [46] against the PAS scheme [10]; they later presented a Gaussian elimination-based algebraic attack [47] against the virtual password scheme [45]; Asghar et al. introduced a statistical attack [5] against the CHC scheme [72]; Dunphy et al. analyzed a replay-based shoulder surfing attack for recognition-based graphical password schemes under a weaker threat model [26]. Compared to them, our work [75] provides security analysis in a more generic setting, which presents two types of generic attacks that can be used to analyze any LRPE schemes. Furthermore, we introduce a new statistical attack, probabilistic decision tree, and a generalized version of existing statistical attacks, multi-dimensional counting. We analyze and re-examine the existing LRPE schemes with these new attack tools. Thereby, we discover the vulnerabilities of Undercover [61] and SecHCI [48] that have not been reported before. We notice that a recent work by Perkovic et al. [56] also identified the design flaw of Undercover independently.

Some other design principles have been proposed for LRPE schemes. Roth et al. [58] proposed the basic principle of using cognitive trapdoor game, where the knowledge of secret should not be directly revealed during password entry. Li and Shum [48] later suggested another three principles that require time-variant responses, randomness in challenges and responses, and indistinguishability against the statistical analysis. Our principles further extend the coverage by including the defense principles against brute force attack, and provide more concrete guidelines against two generic statistical attacks introduced in our work [75].

Until now it is still a challenge to provide a quantitative tradeoff analysis among multiple LRPE schemes [14]. As pointed out by Biddle et al. [14], the usability evaluation in prior research lacks consistency, which makes it is difficult to compare those results. Our quantitative analysis framework is the first attempt to provide a uniform usability measurement based on experimental psychology. Based on this framework and our security analysis, we discover that the tradeoff between security and usability is strong in the absence of trusted devices, which indicates the inherent limitation in the design of LRPE schemes. This limitation was first addressed by Hopper and Blum [35], where they hoped the future research could find out practical solutions for unaided humans that satisfy both security and usability requirements. Unfortunately, from our results, such solution may not exist (i.e. at least a partial secure channel formed by a trusted device has to be incorporated). Coskun and Herley [20] also reached a similar conclusion by analyzing the efficiency of brute force attack with regards to response entropy. Their conclusion is based on the assumption that a user has to make a large number of sequential binary decisions so as to increase response entropy. However, this assumption may not be valid as humans have a strong parallel processing capability when performing certain visual tasks (e.g. visual search).

To the best of our knowledge, our work also makes the first attempt to establish comprehensive design metrics for LRPE schemes, which include the relations between security and usability, built-in security, and universal accessibility. Unlike traditional evaluation metrics, these design metrics can be used to identify the potential limitations during the design phase before conducting user studies. The concept of built-in security is also mentioned in a recent field study by De Luca et al. [24]. Bonneau et al. [15] recently proposed a generic framework for evaluating user authentication proposals. Their framework introduced twenty-five usability, deployability and security benefits from users' perspective, which is different from our metrics developed from designers' perspective. Our metrics are more specific and quantitative, which aim to guide the design of practical LRPE schemes. We also introduce new metrics related to form factor, social norms, and pressure, which are not addressed in the existing works.

As indicated by our design metrics, it is not trivial to design a practical LRPE scheme even with the assistance of a partial secure channel. As the counterexamples [61, 25] shown in our work, an LRPE scheme may still leak secret information related to the password under its secure channel prerequisite. The establishment of partial secure channels may require the adoption of new user interface technologies such as touchscreen. This explains why most LRPE schemes [44, 61, 23, 25, 42, 13, 12] with partial secure channels were proposed in recent years. Among them, our scheme design [76] was mostly inspired by the concept of physical metaphor introduced in [42]. Our scheme distinguishes itself from prior work in the sense that it not only achieves leakage resilience but also retains most benefits of legacy passwords, while some of prior schemes [61, 25] are flawed in terms of security, and the others incur extra usability costs due to various reasons including: 1) using an uncommon device such as gaze tracker [44, 23], haptic motor [13], and large pressure-sensitive screen [42], 2) requiring an extra accessory device [12], and 3) inoperable in a non-stationary environment [13].

On the other hand, the procedure of applying random transformations on a fixed password used in our scheme design is a classic idea to prevent password leakage. But it is not easy to be realized in a *human-friendly* manner without the new user interface technologies, which are only available on modern computing devices. These new technologies give our scheme advantages when compared to recently patented schemes. Take GridCode [30] as an example, which asks users to memorize extra secrets (besides the passwords) in order to perform the transformations specified in its scheme design, while our scheme does not have such requirement. Another advantage of our scheme is that each character of the password uses a different hidden transformation during an authentication attempt, while GridCode uses the same transformation for all the characters in the password. If a hidden transformation in GridCode is disclosed, the entire password will be exposed. However, if a hidden transformation in our scheme is disclosed, only the single character associated with the transformation will be exposed. These two fundamental differences show both security and usability advantages of our scheme [76].

Other prior research related to password-based user authentication can be found in a recent survey paper [14], which summarized the development of new password entry schemes in the past decade.

# Chapter 3

# On Limitations of Designing Leakage-Resilient Password Entry: Attacks, Principles and Usability

## 3.1 Introduction

This chapter reveals the infeasibility of designing practical leakage-resilient password entry (LRPE) schemes in the absence of trusted devices by investigating the inherent tradeoff between security and usability in LRPE design. Compared to an LRPE scheme, legacy passwords that are used pervasively ask a user to directly input his entire plaintext password recalled from the user's memory, so that an observation of a single authentication session is sufficient to capture the password. In order to prevent password leakage during password entry, a user needs to input the password indirectly, which imposes an extra burden on the user and creates a *trade-off* between security and usability. How to design a password entry scheme that *minimizes password leakage and is still easy to use* is the fundamental problem in LRPE design.

An ideal LRPE scheme allows a user to generate a *one-time password* (OTP) for each authentication session based on an easy-to-remember password. This can be easily achieved when a secure channel is available between user and authentication service. The secure channel blinds the adversary by decoupling a user input from the underlying password, when the message delivered over the secure channel is not revealed to the adversary. However, the prerequisite of a secure channel may be infeasible or introduces other vulnerabilities in practical settings. For example, when the secure channel is formed by a trusted device such as secure token or mobile phone, that device is subject to theft or loss. This motivates the existing research on usable and secure LRPE schemes with only the support of human cognitive capabilities [52, 35, 58, 48, 71, 72, 78, 10, 61, 6]. A few representative schemes include Convex Hull Click (CHC) [72], Cognitive Authentication Scheme (CAS) [71], and Predicate-based Authentication Service (PAS) [10].

The difficulty in designing an LRPE scheme stems from the *capability asymmetry* between user and strong adversary. A strong adversary may use a hidden camera or malicious software to record complete interactions between a user and his computer and then analyze the data with powerful machines. Many LRPE schemes [35, 48, 71, 72, 78, 10, 61, 6] have been proposed to defend against this type of password leakage attacks. However, as we will demonstrate later in this work, all the existing proposals with acceptable usability are vulnerable to either or both types of generic attacks: brute force attack and statistical attack.

Brute force attack is a pruning process for the entire candidate password set, whose strength has often being underestimated in prior research. Our experiments show that brute force attack is able to recover the passwords of certain existing LRPE schemes from a small number of observations of authentication sessions. Statistical attack, on the other hand, represents a learning process to extract a user's password due to statistical significance of the password. We introduce two types of statistical attack, probabilistic decision tree and multi-dimensional counting. Rigorous experiments are conducted to show the effectiveness of these two attacks in breaking existing schemes.

We note that these two generic attacks are different from other specific attacks that have been systematically studied in the literature, including SAT [32] and Gaussian elimination [47]. SAT attacks can be efficiently prevented by asking a user to select only one of the correct responses while multiple correct responses can be derived from each challenge, since this would increase the size of the SAT expression exponentially with the number of observations. On the other hand, Gaussian elimination-based algebraic attacks can be efficiently prevented by using a nonlinear response function [48] or introducing noises from user's intentional mistakes [35]. Unlike these specific attacks, brute force and statistical attacks cannot be easily defended without significantly sacrificing the scheme's usability, which implies inherent limitations of LRPE schemes without using trusted devices. In order to defend against these attacks, we introduced five design principles which should be followed to achieve leakage resilience. Using counterexamples, we show that an LRPE scheme can be easily broken when these principles are violated.

To further understand the tradeoff between security and usability in the design of LRPE schemes, we propose for the first time a quantitative analysis framework on usability costs of LRPE schemes. This framework decomposes the process of human-computer authentication into atomic cognitive operations. Performance data of average human-beings reported in psychology literatures [65, 27, 21, 70, 22, 55, 57, 19, 73, 74, 36, 17, 34] are used to estimate usability costs of existing LRPE schemes [35, 48, 71, 72, 78, 10, 61, 6]. Our analysis results are consistent with the experimental results reported in the original literatures, while the hidden costs previously not addressed are identified. Our results show that a secure LRPE scheme in practical settings [35, 6] always leads to a considerable amount of cognitive workload, which explains why some of the existing LRPE schemes require extremely long authentication time and have high authentication error rate. This limitation has not been, and will not be easily solved in LRPE design in the absence of trusted devices.

In a nutshell, the contributions of this work are three-fold:

- We analyze and demonstrate the effectiveness of two types of generic attacks, brute force and statistical attacks, against LRPE schemes. We propose two statistical attack techniques, probabilistic decision tree and multi-dimensional counting, and show their effectiveness against existing schemes.
- We introduce five principles that are necessary to mitigate brute force and statistical attacks. We use typical existing LRPE proposals as counterexamples to show that an adversary can easily obtain a user's password in the schemes violating our principles.
- We establish the first quantitative analysis framework on usability costs of the existing LRPE schemes. This framework utilizes the performance models of atomic cognitive operations in authentication to estimate usability costs. Our analysis result shows that there is a strong tradeoff between security and usability in the existing LRPE schemes. It implies that an unaided human may not be competent enough to effectively use a secure LRPE scheme in practical settings; in other words, it is inevitable to incorporate certain trusted device in LRPE design.

## 3.2 Definitions and Threat Model

In this section, we introduce related notions and our threat model. We focus on the fundamental problem of designing LRPE schemes for unaided humans, i.e. *when a secure channel or trusted device is unavailable*. We exclude the LRPE schemes using secure channel or trusted device in this work unless explicitly mentioned.

### 3.2.1 Leakage-Resilient Password Entry

An LRPE scheme is essentially a challenge-response protocol between human and computer (as demonstrated in Figure 3.1). We refer to human as *user*, and computer as *server*. During registration, a user and a server agree on a *root secret*, usually referred to as a **password**. The user later uses the root secret to generate *responses* to *challenges* issued by the server to prove his identity. Unlike legacy passwords, a response in LRPE is an obfuscated message derived from the root secret, rather than the plaintext of the root secret itself. Considering the limited cognitive capabilities of unaided humans, a usable obfuscation function $F$ is usually a many-to-one mapping from a large candidate set to a small answer set. The small size of the answer set increases the success rate of *guessing attack* where an adversary attempts to pass the authentication by randomly picking an answer from the answer set. For this reason, an *authentication session* of LRPE often requires executing multiple rounds of the challenge-response procedure in order to reach an expected authentication strength $D$ (specifically, the resistance against random guessing, e.g. $D = 10^{-6}$ for 6-digit PIN), where each round is referred to as an *authentication round*. We use $d$ to denote the average success rate of guessing attack per authentication round. Given $d$ and $D$, the minimum number $m$ of authentication rounds for an authentication session is $\lceil \log_d D \rceil$.

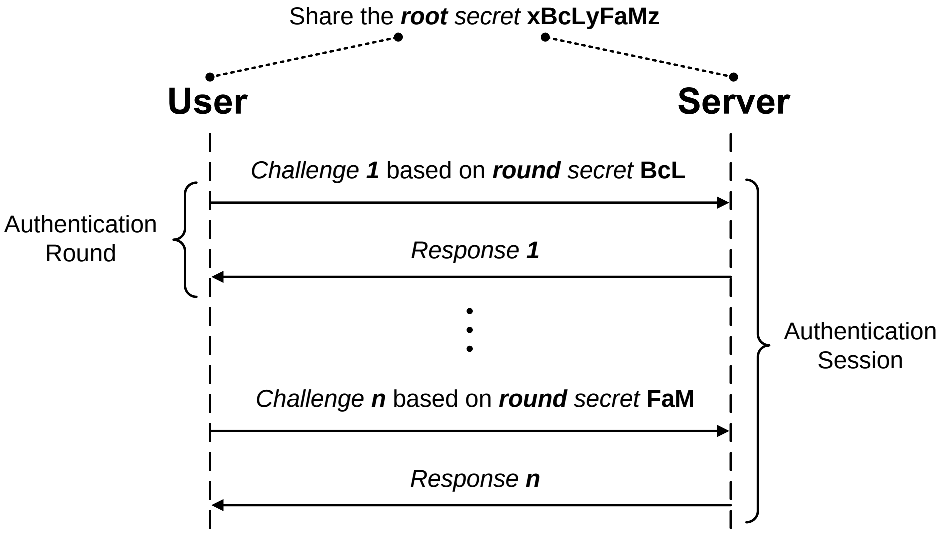

Share the ***root*** *secret* **xBcLyFaMz**
**User**
**Server**
*Challenge **1*** based on ***round*** *secret* **BcL**
Authentication Round
*Response **1***
Authentication Session
*Challenge **n*** based on ***round*** *secret* **FaM**
*Response **n***

Figure 3.1: Demonstration of a typical LRPE scheme

To imbue the server with a high flexibility in challenge generation, the *k-out-of-n paradigm* [35] has been adopted for secret agreement in most existing LRPE schemes [35, 48, 71, 72, 78, 61, 6]. In this paradigm, the root secret consists of $k$

independent elements randomly drawn from a pool of $n$ elements. An element can be an image, a text character, or any symbol in a notational scheme. The set of $k$ secret elements is called the *secret set* (and forms the root secret of the user), and the complementary set is called the *decoy set*. The server knows the secret set chosen by the user, and uses a subset or all of these $k$ elements to generate the challenge in each round. We refer to the chosen portion of the root secret for an authentication round as a *round secret*.

Based on the above notions, the common system parameters of the most existing LRPE schemes [35, 48, 71, 72, 78, 10, 61, 6] can be described by a tuple $(D, k, n, d, w, s)$, where $D$ is the expected authentication strength of an authentication session, $k$ is the number of secret elements drawn from an alphabet of $n$ candidate elements, $d$ is the average success rate of guessing attack in a single round, $w$ is the average window size which is the number of elements appearing on the screen for an authentication round, and $s$ is the average length of user's decision path which is the number of decisions that a user has to make before producing the correct response for an authentication round. The total round number $m$ can be derived from $D$ and $d$. The parameters $m$, $w$, and $s$ are required for usability evaluation. More details will be given in Sections 3.5 and 3.6.

### 3.2.2 Threat Model and Experimental Settings

There are two types of passive adversary models for password leakage attacks used in prior research. The weaker passive adversary model (e.g. *cognitive shoulder-surfing* [58]) assumes that the adversary is not able to capture the complete interaction between a user and the server [58]. Such an assumption actually forms a secure channel between user and server, which transforms the password leakage problem to the protection of the secure channel. However, this assumption may not hold for a prepared adversary who deploys a *hidden camera, key logger, or phishing web site* to capture the whole password entry process. To address such realistic concerns, recent efforts [48, 71, 72, 78, 10, 61, 6] have focused on the strong passive adversary model, where the adversary is allowed to record the complete interaction between the user and the server.

In the strong passive adversary model, password leakage during human-computer authentication is unavoidable. The user's response is based on his knowledge of the password, which distinguishes it from a random choice as required for the authentication purpose. This difference leaks information about the password. After recording a sufficient number of authentication rounds, the adversary may use any reasonable computation resources to analyze and recover the underlying password. The research problem under such a threat model is to lower the rate of password leakage while maintaining acceptable usability for unaided humans.

In this work, we consider both brute force attack and statistical attack under this strong passive adversary model. The security strength of an LRPE scheme is defined as *the resistance against these two generic attacks given the same success rate of random guessing* (i.e. the same authentication strength for a legitimate user). We will use simulation experiments to evaluate the security strength of existing schemes, whose process is summarized as follow: 1) Generate a random password as the root secret (i.e. the password); 2) Generate a challenge for an authentication round; 3) Generate a response based on the password and the underlying scheme design; 4) Analyze the collected challenge-response pairs after each authentication round assuming that the adversary has full knowledge of the scheme design except the password; 5) Repeat steps 2, 3, and 4 until the exact password is recovered. The final findings shown in the following sections are the average results of 20 runs for each scheme.

## 3.3 Brute Force Attack and Its Defense Principles

### 3.3.1 Attack Strategy

Brute force attack is a general pruning-based learning process, where the adversary keeps removing irrelevant candidates when more and more cues are available. Its procedure can be described as follows: 1) List all possible candidates for the secret in the target scheme; 2) For each independent observation of a challenge-response round, check the validity of each candidate in the current candidate set by running the verification algorithm used by the server, and remove invalid candidates from the candidate set; 3) Repeat the above step until the size of candidate set reaches a small threshold.

The above procedure shows that the efficiency of brute force attack in the leakage resilience setting is *design-independent*, and is only limited by the size of the candidate set. We introduce two statements to further describe the power of brute force attack. These statements apply not only to root secret, but also to round secrets when the adversary is able to reliably group the observations for individual round secret.

**Statement 1:** *The verification algorithm used in brute force attack for candidate verification is at least as efficient as the verification algorithm used by server for response verification.*

The proof is trivial as the verification process for candidate pruning is essentially the same as the verification process for the server to check correct response. It is also possible for the adversary to design a more efficient algorithm if there are correlations between candidates.

**Statement 2:** *The average shrinking rate for the size of valid candidate set is the same as one minus the average success rate of guessing attack.*

The average success rate of guessing attack is defined as the probability of generating correct response by randomly picking a candidate from the candidate set. This is an equivalent definition of average shrinking rate of the valid candidate set. Given $X$ as the size of the candidate set, and $d$ as the average success rate of guessing attack, the average number of rounds to recover the exact secret is $m = \lceil \log_{1/d} X \rceil$, assuming that each candidate is independent of each other. If each candidate is not independent, the average number of rounds to recover the exact secret will be smaller than $m$. This statement can be used to estimate the average success rate of guessing attack, $d = X^{-\frac{1}{m}}$, when the precise analysis is difficult to perform (see later examples). The statement also explains why most password entry schemes [58] reveal the entire secret after one or two authentication sessions recorded by the adversary, as their expected success rates of guessing attack are sufficiently low so that the whole candidate set rapidly collapse to the exact secret. This implies that, when brute force attack is feasible, enhancing strength against guessing attack is strictly at the cost of sacrificing leakage resilience.

### 3.3.2 P1: Large Root Secret Space Principle

**Principle 1:** *An LRPE scheme with password leakage should have a large candidate set for the* **root secret**.

The first principle requires a large password space as the basic defense against brute force attack, where large means that it is computational infeasible for the adversary to enumerate all candidates in a practical setting (the same meaning of *large* will be used in the following discussion). This principle seems trivial but actually not, as the necessity of involving a large password space depends on whether an LRPE scheme has password leakage under a given threat model, which is not straightforward to decide. In general, there are three possible leakage sources in an LRPE scheme: *the response alone*, *the challenge-response pair*, and *the challenge alone*. Among them, the last source has not been well recognized. We use Undercover [61] as a counterexample to show that password leakage could happen even when a secure channel is present.

Undercover is a typical scheme based on the $k$-out-of-$n$ paradigm. During registration, a user is assigned $k$ images as his secret from a pool of $n$ images. In each authentication round, the user is asked to recognize if there is a secret image from $w$ candidate images and report the position of that image if the secret image is shown in the current window; otherwise the user reports the position of the "none" symbol. Before the user reports the position, a haptics-based secure channel is deployed to map the real position to a random position decided by the hidden message delivered via the secure channel.

The hidden mapping blinds the adversary from learning any information from the response. The authors suggested a small password space is sufficient so that the default parameters are $k = 5$, $n = 28$, and $w = 4 + 1$ (i.e. four images and a "none" symbol). The number of candidate root secrets is $C_{28}^{5} = 98280$. However, this scheme does not prevent the challenge alone from becoming a source of leakage. *In Undercover, there is at most one secret image among the $w$ candidate images for each authentication round. This implies a candidate of the root secret is invalid if two images in this candidate appeared in an authentication round.* Since it has a small candidate space, we can use brute force to recover the secret with the information from the challenge alone. Figure 3.2 shows how the size of the candidates shrinks as the number of observed authentication rounds increases. On average, 53.06 rounds (6 sessions) are sufficient to recover the exact secret, and the size of the candidate set can be reduced to less than 10 after 43.55 rounds (5 sessions). This result shows that *a secure channel alone is not sufficient to prevent password leakage.*

The same problem also appears in the Convex Hull Click (CHC) scheme [72], where the default parameters are $k = 5, n = 112, w = 83$. The size of the candidate set for its root secret is $C_{112}^{5} = 1.34 \times 10^{8}$. In our simulation, we are able to recover the exact secret within 12.28 rounds (2 sessions). Another interesting finding for CHC is that we can now estimate the average success rate of guessing attack from the results of brute force attack, though a precise analysis is difficult [72]. According

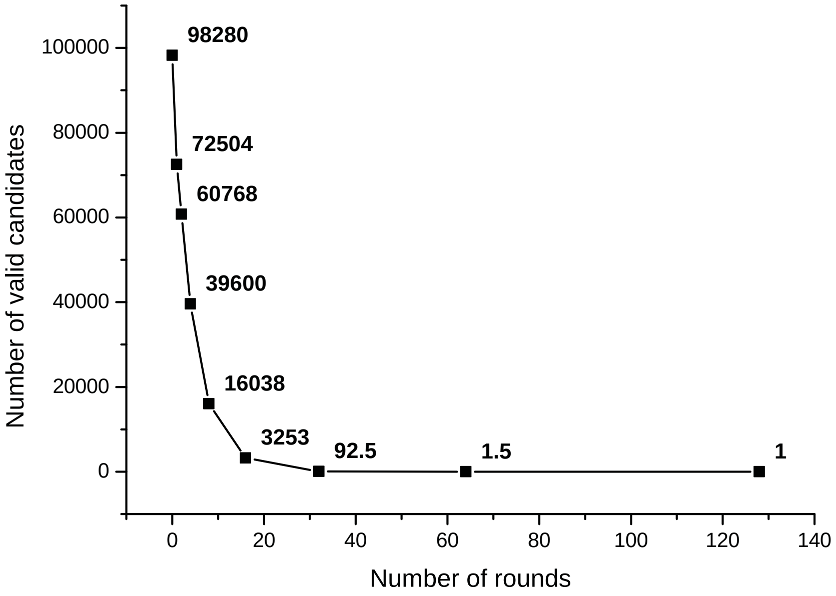


Figure 3.2: The average number of valid candidates shrinks for the Undercover scheme.

to *Statement 2*, the average success rate is $21.78\% = (C_{112}^{5})^{-\frac{1}{12.28}}$. This technique can also be applied to other complex LRPE schemes to determine their security strength when the other analysis techniques are infeasible.

### 3.3.3 P2: Large Round Secret Space Principle

**Principle 2:** *An LRPE scheme with password leakage should have a large candidate set for the* **round secret**.

This principle emphasizes that a large candidate set for the *root secret* is necessary but not sufficient to defend against brute force attack. The large candidate set for the root secret can be *broken down* based on the attack to the round secrets. We use Predicate-based Authentication Services (PAS) [10] as a counterexample to show that a round secret with a small candidate set can be easily recovered and later used to reveal the root secret.

During registration of PAS, a user is asked to remember $p$ secret pairs, each of which includes a secret position and a secret word. At the beginning of each authentication session, the server prompts for an integer index $I$. Then the user uses

$I$ to calculate $p$ predicates as follows: For each pair, the corresponding predicate is the secret position and a secret character. The secret character is the $x$th character in the secret word (1-based indexing), where $x = 1 + ((I - 1) \mod len)$, and $len$ is the length of the secret word. For example, given two secret pairs (⟨2,3⟩, sente), (⟨4,1⟩, logig) and $I = 15$, the predicates are (⟨2,3⟩, e) and (⟨4,1⟩, g), where $x = 5 = 1 + ((15 - 1) \mod 5)$, and the secret position $\langle a, b \rangle$ means "at row $a$ and column $b$". Given these $p$ predicates, the user examines the cells at secret positions in $l$ challenge tables to check whether a secret character is present in its corresponding cell. It yields an answer vector that consists of $p \cdot l$ "present" or "absent" answers with a candidate space of $2^{pl}$. This vector is then used to lookup another response table, which provides a many-to-one mapping from $2^{pl}$ elements to $2^l$ elements. Finally, the user inputs one of those $2^l$ elements indexed by the answer vector to finish an authentication round.

The above many-to-one mapping is used in PAS to confuse the adversary. However, when the round secret only has a small candidate set, many mappings will have the same pre-image and the effective mapping space collapses to the candidate set of the round secret. In PAS, the size of the candidate set for the round secret is $422500 = (25 \times 26)^2$ for the default parameters, where $p = 2$, and there are 25 cells in each challenge table and 26 possible letters for the secret character. It is not difficult to use brute force to recover the round secret of PAS. Figure 3.3 shows the shrinking of the candidate set size as the number of observed authentication rounds increases. On average, 9.4 rounds are sufficient to recover the exact round secret (1 session). Since all the predicates generated from the same secret pair share the same secret position, after recovering the first round secret, it is easy for the adversary to recover the other round secrets and finally the root secret. A similar attack technique has been used in [46]. The same problem also appears in the S3PAS scheme [78], which is a variant of the CHC scheme [72]. In our experiments, we are able to discover the exact root secret in 8 sessions.

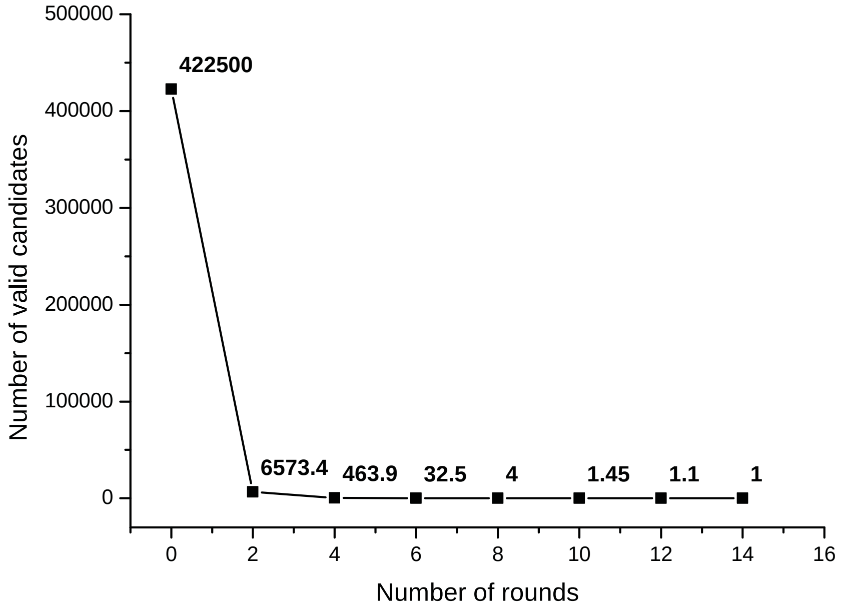


Figure 3.3: The average number of valid candidates shrinks for the PAS scheme.

## 3.4 Statistical Attack and Its Defense Principles

### 3.4.1 Attack Strategy

Statistical attack is an accumulation-based learning process, where an adversary gradually increases its confidence on relevant targets when more and more cues are available. Compared to brute force attack, statistical attack has fewer limitations as it can be applied to schemes with a large password space. Recall that a user response is statistically biased towards his knowledge of the secret. Theoretically there exists a specific statistical attack for any password entry scheme. The efficiency of statistical attack is *design-dependent* and varies with different schemes and different analysis techniques. Here we introduce two general statistical analysis techniques that are able to efficiently extract the root secret of most existing schemes.

The first technique is *probabilistic decision tree*. It works efficiently for the existing schemes based on simple challenges [71, 72, 78, 10]. The procedure is described as follows: 1) Create a score table for each possible individual element or affordable-sized element group in the alphabet of the root secret, where *affordable* means computational feasible to maintain. We refer to a score table whose entry contains $t$ individual elements as a *t-element score table*. 2) For each independent observation of a challenge-response pair, the adversary enumerates every *consistent* decision path that leads to the current response. Each possible decision path is assigned a probability calculated based on the uniform distribution. For the $k$-out-of-$n$ paradigm, the probability is $p1 = k/n$ for a decision event in which the corresponding individual element belongs to the secret set, and $p0 = 1 - p1$ for the complementary event. For the example decision path $X$ given in Figure 3.4, its probability is $p(X) = p1 \cdot (p0 \cdot p1)$. After enumerating all consistent decision paths, the adversary sums up the probabilities of these paths and uses the sum $p_c$ to normalize the probability $p(X)$ for each decision path to its conditional probability $p(X|C) = p(X)/p_c$. The conditional probability represents the probability that a decision path is the path chosen by the user when the current response $C$ is observed. After the normalization, the adversary updates the score table using $p(X|C)$. For an entry that appears in a consistent decision path $X$, its score will be added by $p(X|C)$ if the corresponding event is that the entry belongs to the secret set, otherwise its score will be deducted by $p(X|C)$. 3) Repeat the above step until the number of entries with different score levels reaches a threshold (e.g. finding out $k$ entries with the highest/lowest scores when each entry represents a single element).

The second technique is *counting-based statistical analysis*. The basic idea is to simply maintain a counting table for the occurrences of elements. Multiple counting tables can be maintained simultaneously according to different response groups. The procedure proceeds as follows: 1) Create $l$ counting tables for $l$ response groups. The adversary creates a counting table for each possible response if affordable. “Any response” is still a useful response group if the secret elements appear more or less frequently than the decoy elements *in the challenge*. An entry in a counting table can be an individual element or affordable-sized element group. We refer to a counting table whose entry contains $t$ individual elements as a *t-element counting table*. When $t \geq 2$, we call this type of statistical analysis as

A *decision path* is an emulation of the user's decision process that consists of multiple decision nodes. Each *decision node* represents a decision event decided by the membership relation of a corresponding entry in the score table, whether or not it belongs to the secret set.

Consider a scheme which shows a four-element window $\langle S_1{:}1,\ S_2{:}2,\ S_3{:}1,\ D_1{:}1\rangle$ and asks the user to report the sum of the numbers associated with the *first* and *last* secret elements displayed in the window, where $S_i{:}x$ represents a secret element associated with number $x$, and $D_i{:}y$ represents a decoy element associated with number $y$. Since the correct response for this challenge is 2 by adding the numbers associated with the first and third elements, its decision path is $X = \langle S_1{:}1\rangle|\langle D_1{:}1;\ S_3{:}1\rangle$. There are two segments in this decision path. The first segment implies that $S_1$ is a secret element, and the second segment implies that $D_1$ is a decoy element and $S_3$ is a secret element. There usually exist other decision paths leading to the same response, such as $\langle S_1{:}1\rangle|\langle D_1{:}1\rangle$.

Figure 3.4: Definition and example for decision path

*multi-dimensional counting*. 2) For each independent observation of a challenge-response pair, the adversary first decides which counting table is updated according to the observed response. Then each entry in the chosen counting table is incremented by the number of occurrences of the corresponding individual element or element group. If the group of "any response" is used, its counting table is always updated for each observation. 3) Repeat the above step until the number of entries with different score levels reaches a threshold (e.g. finding out $k$ entries with the highest/lowest scores when each entry represents a single element). The score for an entry is a weighted sum of the count values for the same entry in different tables. The weight function is dependent on the specific target scheme and the response grouping strategy.

### 3.4.2 P3: Uniform Distributed Challenge Principle

**Principle 3:** *An LRPE scheme with password leakage should make the distribution of the elements in each* **challenge** *as uniformly distributed as possible.*

This principle requires that an LRPE scheme should be able to generate the challenges without knowing the secret[1]. For example, if there is a *structural requirement* in the challenge generation, password leakage is very likely to happen. Non-uniformly distributed elements in a challenge leave cues for the adversary to recover the secret even without knowing the response. Undercover [61] is a typical counterexample to show password leakage from biased challenges.

Undercover ensures that the distribution for each image is unbiased by showing every candidate image exactly once for each authentication session. However, its 2-dimensional distribution is biased in each authentication round, as secret-secret pairs cannot appear in the challenge (at most 1 secret image appearing). We use *2-element counting table* to recover the secret from the challenge. For each pair of candidate images, the count value is zero only if both of them belong to the secret set after a sufficient number of observations. On average, it is sufficient to recover the exact secret within 172.7 rounds (20 sessions), and recover $80\%$ secret elements (five secret images in total) after 126.9 rounds (15 sessions).

The same problem also appears in the CHC scheme [72] and in the low-complexity CAS scheme [71]. Both of them require that at least $k$ secret elements appear in the challenge window, while the challenge window only holds a subset of candidate elements. These structural requirements make the distribution of the elements in each challenge deviate from the uniform distribution. Under default parameters, we are able to recover the exact root secret within 18.18 rounds (2 sessions) for CHC. For the low-complexity CAS scheme, we can recover the exact root secret (i.e. 60 independent secret images) within 2087.2 rounds (105 sessions), and recover $90\%$ secret elements within 870.4 rounds (44 sessions).

The above discussion shows that the consequence of the distribution bias caused by structural requirements in the challenge is subtle to identify and has not been well recognized. In order to prevent leakage from biased challenges, the distribution of the elements in each challenge should be indistinguishable from the uniform distri-

[1] Even if server knows the secret, the secret (or its alternative form, e.g. hash value) should be only used to verify the response.

bution. If a structural requirement is *compulsory* in an LRPE scheme (e.g. at least $k$ secret elements being displayed) but the element distribution in each challenge is not uniform when the challenge window only shows a subset of candidate elements, the scheme should display *all* the candidate elements in each challenge.

### 3.4.3 P4: Large Decision Space or Indistinguishable Individual Principle

**Principle 4:** *An LRPE scheme with password leakage should make each individual element* **indistinguishable** *in the probabilistic decision tree if the candidate set for decision paths is* **enumerable***.*

This principle is critical to limit the feasibility of probabilistic decision tree attack. The power of probabilistic decision tree stems from its emulation of all possible decision processes leading to the observed response. The emulation creates a tight binding between each challenge and its response, from which the adversary is able to extract the subtle statistical difference during the user's decision if individual elements are distinguishable on consistent decision paths. It is not easy to make each individual element indistinguishable, especially when weight or order information is used in the challenge design. We use the high-complexity CAS scheme [71] as a counterexample to show how probabilistic decision tree efficiently discovers the root secret even when a number of decision paths lead to the same answer.

The high-complexity CAS scheme is another typical scheme based on the $k$-out-of-$n$ paradigm. During registration, a user is assigned $k = 30$ images as his secret from a pool of $n = 80$ images. In each authentication round, a challenge is an $8 \times 10$ grid consists of all the images, one image for each cell. The user is asked to mentally compute a path starting from the cell in the upper-left corner. The computation rule is described as follows: Initially the current cell is the cell in the upper-left corner. If the image in the current cell belongs to the secret set, move down by one cell, otherwise move right by one cell; if the next moving position is out of the grid, it is referred to as an *exit position*. The path computation ends with an exit position. The user reports the answer associated with that exit position to finish an authentication round. The answer is an integer from $[0, 3]$, and is randomly assigned to each exit position. Since the same answer is assigned to multiple exit positions (i.e. 4 answers assigned to 18 exit positions), the adversary cannot easily tell which the exact exit position is. For each exit position, there are also many possible paths leading to it, which further increases the difficulty for the adversary.

Since the default parameters are large ($k = 30$, $n = 80$), brute force attack is infeasible for this scheme. The scheme also follows *Principle 3* to display all the candidate images in each challenge so that the adversary cannot extract the secret only by analyzing the challenges. However, each individual element is distinguishable in this scheme during the decision process, as each element has different impact on the transition of decision paths. One can use probabilistic decision tree to recover the secret from the observations of challenge-response pairs.

Each possible path leading to the observed response forms a decision path in the probabilistic decision tree. The probability of a decision path is decided by the movements on this path. For example, a path $X$ = $\langle$*DOWN*, *RIGHT*, *RIGHT*, *DOWN*$\rangle$ means the first and the fourth images belong to the secret set, while the second and third images do not. The probability $p(X)$ is $p1 \cdot p0 \cdot p0 \cdot p1$, where $p1 = k/n$ and $p0 = 1 - p1$. Initially, we create a *1-element score table*. Given a response with the answer $i$, we enumerate all consistent decision paths leading to this answer, and update the score table according to the conditional probability $p(X|\text{response} = i)$.

For an $8 \times 10$ grid specified by the default parameters, there are 43758 possible decision paths in total, with average path length of 14.5539. For each candidate image, its score is at a significantly high level if it belongs to the secret set after a sufficient number of observations. Figure 3.5 shows the false positive rate decreasing along with the increasing number of observed authentication rounds. On average, it is sufficient to discover the exact secret within 640.8 rounds (65 sessions), and discover 90% secret elements after 264.7 rounds (27 sessions). Although the required number of session observations is larger, it is still possible for the adversary to collect them using a key logger, and such security strength is achieved only when the user is able to remember 30 independent secret images.

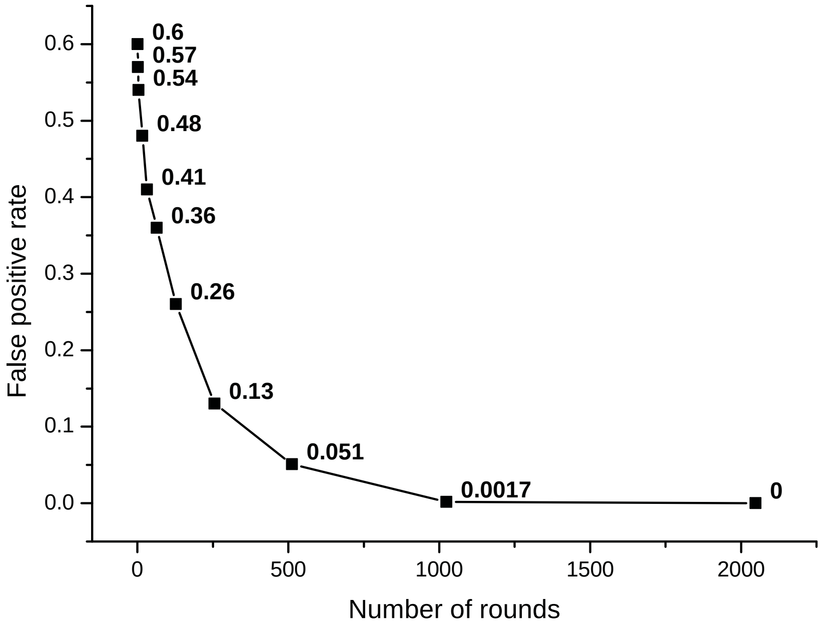


Figure 3.5: The average false positive rate decreases for the high-complexity CAS scheme.

Probabilistic decision tree can also be applied to the low-complexity CAS scheme [71], the CHC scheme [72], the S3PAS scheme [78], and the PAS scheme [10]. All of them are based on simple challenges with an enumerable candidate space for decision paths and the individual element has different impact on the transition of decision paths.

From these counterexamples, we can see that *it is necessary to increase the number of candidate decision paths if it is infeasible to make each individual element indistinguishable in the probabilistic decision tree.* The only known designs that satisfy this indistinguishability requirement are the counting-based schemes [35, 48]. In those schemes, there is no order or weight information associated with each candidate element, which usually distinguishes the elements in decision paths. The user is asked to count their secret elements appearing in the challenge. The final response is based on the count value. For these schemes, probabilistic decision tree attack does not apply, but they may still subject to counting-based statistical analysis attack.

### 3.4.4 P5: Indistinguishable Correlation Principle

**Principle 5:** *An LRPE scheme with password leakage should minimize the statistical difference in low-dimensional correlations among each possible* **response**.

This principle is complementary to *Principle 4* to limit the efficiency of counting-based statistical analysis. Although counting-based statistical analysis is straightforward, it cannot be completely prevented without a secure channel, as the user's response is always statistically biased towards his knowledge of the secret.

In the extreme case, the adversary is able to maintain a counting table to hold every candidate for the root secret, and update the table according to every available observation. Using these counting tables, the statistical difference caused by the knowledge of the secret is always identifiable even when the user is asked to make intentional mistakes at a predefined probability only known by the server (see informal proof in Figure 3.6). In this sense, the counting-based statistical analysis is more powerful than brute force attack if sufficient resources are available to the adversary.

*Proof.* Assuming the user makes mistakes in the responses with a fixed error probability $\rho$, the average success rate of guessing attack on the "correct" response for each authentication round is $d$, the number of candidate root secrets is $N$, the adversary cannot distinguish the true secret only when the equation $\frac{1-\rho}{(1-\rho)(Nd-1)+\rho\cdot Nd} = \frac{1}{N-1}$ holds, which means the decoys get the same count value as that of the secret. Solving the equation gives $\rho = 1 - d$. Therefore, the user should make the correct response with probability $1 - \rho = d$. This implies that the user's decision process is similar to a random guessing, which defeats the purpose of the authentication. □

Figure 3.6: Informal proof for the strength of multi-dimensional counting

In reality, the resources available to the adversary are not unbounded. The cost of maintaining *t-element counting tables* is $O(n^t)$, which increases exponentially with the number of elements $t$ contained in a table entry, where $n$ is the number of total individual elements. If the adversary fails to maintain a high-dimensional counting table, the correlation information in these tables is safe from the adversary. However, it is still possible for the adversary to exploit the low-dimensional correlation to recover the secret. We use SecHCI [48] as a counterexample to show how it works while brute force and probabilistic decision tree are infeasible.

During registration of SecHCI, a user is assigned $k$ icons as his secret from a pool of $n$ icons. In each authentication round, the challenge is a window consisting of $w$ icons. The user is asked to count how many secret icons appearing in the window. After getting the count value $x$, the user calculates $r = \lfloor (x \mod 4)/2 \rfloor$. The final response $r$ is either 0 or 1. The challenge is designed so that each individual candidate has the same probability to appear in the window for either response. Hence, it is impossible for the adversary to extract useful information based on *1-element statistical analysis*.

Since the default parameters are large, $k = 14$, $n = 140$, brute force attack is not applicable. Also because it is a counting-based scheme, it is not subject to probabilistic decision tree attack according to *Principle 4*. However, 2-dimensional counting attack is still applicable. Compared to decoy icons, there are $0.599$ more pairs on average among secret icons for response 0, and $0.599$ less pairs on average among secret icons for response 1. So we can use two *2-element counting tables* to recover its secret, one table for each response. We update the count value for each pair displayed in each challenge and each response. The score for each entry is calculated as the value difference between these two tables. For each pair of candidate icons, the score is at a significantly high level if both of them belong to the secret set after a sufficient number of observations. Figure 3.7 shows the pair-based score distribution after 20000 authentication rounds, from which the secret icons can be easily distinguished. On average, it is sufficient to recover the exact secret with 14219.4 rounds (711 sessions), and recover $90\%$ secret elements after 10799.8 rounds (540 sessions). Since SecHCI follows most of our principles, these numbers are much larger than the schemes we analyzed previously, but it is still far less secure than it is claimed to be [48]. Its security strength is achieved by imposing a high cognitive workload where the user is asked to correctly examine 600 icons (30 icons per round × 20 rounds) one by one for each authentication session.

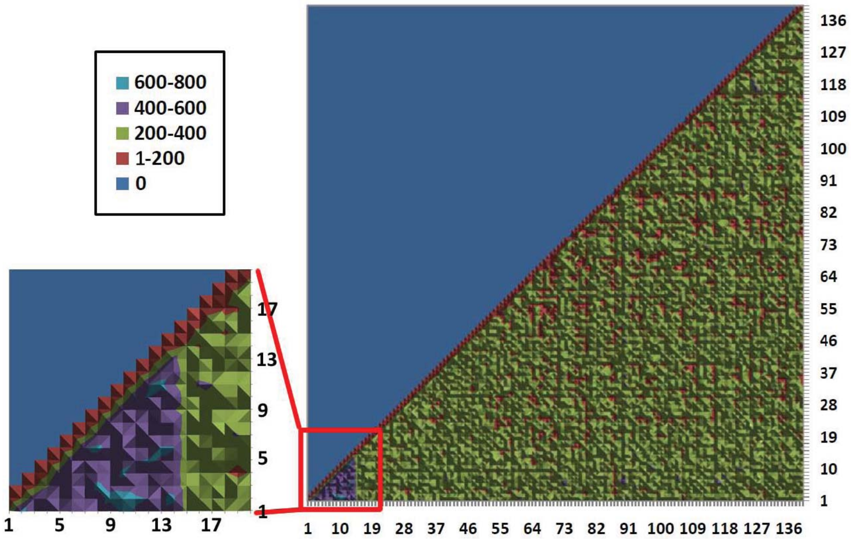


Figure 3.7: The pair-based score distribution is distorted for the SecHCI scheme. The first 14 elements are the secret icons, whose pair-based scores are distinguishable from the scores of other icons.

The password leakage on pair-based statistics for SecHCI can be fixed by changing its response function from $r = \lfloor (x \mod 4)/2 \rfloor$ to $r = x \mod 2$, where $x$ is the number of secret icons in the challenge window, but this fix will make SecHCI subjects to algebraic attack based on Gaussian elimination [48]. This is also the original motivation of the scheme to use its current function. To further defend against this algebraic attack, a user has to produce incorrect answers with a fixed error probability to create noises as suggested in [35]. This certainly further decreases the scheme's usability.

Another design limitation on counting-based scheme is that the response function cannot be in the form of $r = x \mod q$, where $q$ is an integer larger than 2. In our simulation experiments, we discover that pair-based statistical difference in

Counting-based LRPE schemes appears when $q$ is larger than 2, and increases with the value of $|r - w \cdot k/n|$, where $r$ is the response value, $w$ is the window size, $k$ is the number of secret elements, and $n$ is the total number of elements. This can be explained as follows: For a response, if the expected number of secret elements in a window is less than the expected number $w \cdot k/n$ derived from the uniform distribution, the number of pairs among secret elements is also less than the expected number $C^2_{wk/n}$, and the number of pairs among decoy elements is larger than the expected number derived from the uniform distribution, and vice versa. The adversary is then able to distinguish the secret elements from the other elements by grouping the observations of different responses. Such attack restricts a counting-based scheme from using a larger $q$ and thus reducing the number of rounds of an authentication session without using a more complex response function.

## 3.5 Usability Costs of Defense Principles

In this section, we provide a qualitative analysis for usability costs of our defense principles. We show the relation and tradeoff among the constraints imposed by our principles and the requirements on human capabilities. This section aims to provide a high level understanding of the quantitative tradeoff analysis to be presented in the next section.

As defined in Section 3.2, the common parameters of an LRPE scheme is a tuple ($D$, $k$, $n$, $d$, $w$, $s$). All of the parameters except $D$ (the expected authentication strength) are affected by our principles. The principles related to brute force attack mainly dictates the memory demand for the secret, and the principles related to statistical attack mainly increase the computation workload for each authentication session. Their impacts are also interrelated.

Principles 1 and 2 require a large candidate set for the root secret and the round secret. This implies that either $k$ increases or $n$ increases. An increase in $k$ requires the user to memorize more elements as his secret. An increase in $n$ will not raise the memory demand, but will increase statistical significance of the secret in the whole candidate set, which indirectly increases the computation workload as analyzed later. Principle 2 also directly raises the computation workload, as it indicates a challenge is not safe against brute force attack if it can be solved by using a small number of possible secret elements. In order to increase the candidate space of the round secret, the round secret must be either randomly selected from the root secret [48, 71, 72] or use all elements in the root secret [35, 6]. The former choice requires the user to recognize the current displayed secret elements that change in every round; the latter requires the user to recall a large number of secret elements that would be difficult when $k$ is large. Finally, more elements appearing in a challenge means more computation workload to aggregate them into the correct response. This demands much more effort compared to using a fixed short round secret in legacy passwords.

Principles 3, 4, and 5 have more impact on ($d$, $w$, $s$). Principle 3 requires that the elements in the challenge should be uniformly drawn from the candidate set. Due to previous requirements of large secret space and our preference of minimizing the memory demand for the secret, the value of $k$ is to be small and the value of $n$ is to be large. The consequence of this is that the average number of secret elements displayed in a challenge window, $w \cdot k/n$, cannot be large enough if the window size $w$ is not large. This restricts the number of possible responses to a small value, which raises the success rate $d$ of guessing attack and increases the round number required to achieve an expected authentication strength $D$. On the other hand, if the window size is large, the LRPE scheme is limited only for large screen devices and it also increases the difficulty for the user to examine the elements in the challenge window. Regardless of the window size, this principle imposes increased computation workload and the error rate for the user. Principles 4 and 5 further rule out most schemes based on simple challenges. Principle 4 states if a leakage-resistant challenge design is not complex enough to aggregate a large number of secret elements into a response, it leads to a counting problem. Principle 5 further states that only 0 and 1 can be safely used as the response for a counting problem if the modular operation is the only operation used to generate the final response. Hence, the three possible choices for a challenge are: 1) a complex challenge using many secret elements - the round number will be small but the challenge will be very difficult for the user to respond (the average length $s$ of decision paths significantly increased); 2) a counting-based challenge using the modular operation - the round number will be large and the challenge will be relatively easier to respond; and 3) a counting-based challenge using a specially designed response function that has a large number of possible responses and satisfies the correlation indistinguishability condition; however, it will be a challenge to design such a function with acceptable usability. All of the three choices impose a considerable burden on the user.

## 3.6 Quantitative Tradeoff Analysis

In this section, we establish a quantitative analysis framework for evaluating the usability cost of typical existing LRPE schemes. This framework decomposes the process of human-computer authentication into atomic cognitive operations in psychology. There are four types of atomic cognitive operations commonly used: single/parallel recognition, free/cued recall, single-target/multi-target visual search and simple cognitive arithmetic. Their performance models characterize the relations between experiment parameters and reaction time of an average human, which are used to evaluate the cognitive workload for typical existing LRPE schemes. The results in this section provide quantitative assessment of the tradeoff between security and usability of LRPE schemes. According to conventions in psychology literature, we will refer user as *subject* in this section.

### 3.6.1 Atomic Cognitive Operations

**(Single/Parallel) Recognition**

Recognition is the process to correctly judge whether a presented item have been encountered before. Recognition can be considered as a matching process of comparing presented items with those stored in memory. The reaction time of a recognition operation depends on the number of items which a subject memorizes. The item set in the subject's memory is referred to as a *positive set*. For single item recognition, that is, only one item is shown to the subject each time, one of the most well-known recognition models [65] evaluates the reaction time as $RT = 0.3964 + 0.0383 \cdot k$, where $k$ is the size of the positive set. When multiple items are present simultaneously, the subject is able to perform recognition in parallel. According to the working memory capacity theory [27, 21, 70], the maximum number of parallel recognition channels is limited to 4 for an average subject. The reaction time of recognizing $x$ items displayed simultaneously can be estimated as $RT = (0.3964 + 0.0383 \cdot k) \cdot \lceil x/4 \rceil$.

Recognition is a common operation in LRPE, which is used by the subject to judge whether an element appearing in the challenge belongs to the positive set. The high-complexity CAS scheme [71] is an example for single item recognition, where the subject is asked to recognize an image in the current position before deciding which image will be recognized in the next move. The low-complexity CAS scheme [71] and SecHCI [48] are examples of parallel recognition. In the low-complexity CAS scheme, the subject needs to find out the first and the last secret image appearing in a window consisting of 20 images; while in SecHCI, the subject needs to identify all his secret images among 30 candidate images.

**(Free/Cued) Recall**

Recall is the other principal method of memory retrieval [8], which is defined as reproducing the stimulus items. Compared to recognition, the recall process is much slower [22, 55]. The common interpretation of this is that recall is associated with greater resource costs than recognition [22]. Recall might be carried out as a slow process of serial search while recognition as a fast process of parallel retrieval [55].

Free recall and cued recall are two basic recall types. In free recall, the subject is given a list of items to remember and then is tested by recalling them in any order [57]. In cued recall, the subject is given a list of items with cues to remember, and cues are given in the test. Cues act as guides to what the person is supposed to remember. For example, given “a body of water”, the phrase is the cue for the word “pond” [22]. Many psychological experiments have shown that the reaction time of free recall increases *exponential* as the size of positive set increases [57, 69]. In contrast, the reaction time for cued recall is much shorter and only increases *linearly* [22, 55].

Some LRPE schemes require subjects to recall all his secret items during the authentication. The LPN scheme [35] and the APW scheme [6] are two examples, where the subject has to recall all the secret items and their corresponding locations in order to read the challenge digit associated with each secret item. These recall processes should be classified as free recall as cues are not presented. However, no experimental data have been provided in psychology literatures for a large positive set consisted of 15 items required by these schemes, while the common size for a positive set is 8 for free recall. Since it is difficult to decide whether the exponential trend still holds when the positive set is large, we use the reaction time of cued recall as a conservative estimation for free recall used in those schemes. According to the experimental results in [55, 19], the formula for the reaction time of cued recall is $RT = (0.3964 + 0.0383 \cdot \varphi \cdot \gamma \cdot k)$, where $\varphi$ is the ratio of cued recall compared to single item recognition ($\varphi = 1.969$ in [55]), while $\gamma$ is the additional penalty if subjects are required to simultaneously recalling the position of an item ($\gamma = 1.317$ in [19]).

### (Single-target/Multi-target) Visual Search

Visual search is a perceptual task that involves an active scan of the visual environment for particular targets among other distractors. The measure of the involvement of attention in visual search is often manifested as a slope of the response time function over the number of items displayed (referred to as *window size*) [73]. For single-target visual search, searching a single target among a set of items, its reaction time is believed to be linear as the window size increases [74, 73] and can be estimated as $RT = 0.583 + 0.0529 \cdot w$ [74], where $w$ is the window size. For multi-target visual search, the reaction time is accelerated instead of increasing linearly as the number of targets increases in a fixed-sized window [36].

Visual search is usually used in LRPE schemes based on simple challenges. PAS [10] and CHC [72] are examples of using single-target visual search and multi-target visual search, respectively. In PAS, the subject is asked to scan a table cell containing 13 random letters to check whether a secret letter is present or not. In CHC, the subject needs to locate 3 secret elements in a window to form a triangle. According to the results from [36], the reaction time of 3-targets visual search in CHC is approximately 1.8 times longer than that of single-target visual search in the same window.

### Simple Cognitive Arithmetic

Simple cognitive arithmetic is a mental task to solve simple problems involving basic arithmetic operations (e.g., $3 + 4$, $7 - 3$, $3 \times 4$, $12 \div 3$). The simple arithmetic problems can be further divided into three subsets, small, large and zero-and-one problems [17]. For both addition and multiplication, small problems are defined as those with the product of two operands smaller than or equal to 25, and large problems are defined as those with the product of two operands larger than 25. The small and large problems in subtraction and division are defined on the basis of the inverse relationships between addition and subtraction and between multiplication and division. Zero-and-one problem is defined as involving 0 or 1 as an operand or answer. The common instances of zero-and-one problems include counting, exclusive-or, and mod 2. As reported in the experiments of [17], the average reaction time is 0.773 seconds for small addition, 0.959 seconds for small division, 0.924 seconds for large addition, and 0.738 seconds for zero-and-one problems.

Simple cognitive arithmetic is usually used in LRPE schemes based on algebra problems. The counting-based schemes [35, 48] are examples, where the subject is asked to count the number of secret icons appearing in the challenge, and use the count value to calculate a response based on a simple algebraic function.

### 3.6.2 Quantitative Analysis Framework

There are two components in our quantitative analysis framework, *Cognitive Workload* (C) and *Memory Demand* (M). Cognitive workload is measured by the total reaction time required by the involved cognitive operations. Long reaction time for each *authentication round* implies that it is difficult for the subject to answer each challenge and the overall error rate is also high. Long reaction time for each *authentication session* implies that the overall cognitive workload is high and the involvement of attention and patience is also high. Memory demand is measured by the number of elements that must be memorized by the subject, which is the prerequisite of any password entry scheme. Since this prerequisite process is independent from the authentication process, we consider it as a separate component. Since the precise relation between overall error rate and total reaction time is difficult to measure in controlled psychology experiments, our framework provides *lower bound* estimation for the usability of a human-computer authentication scheme. The detailed calculation for both components is described as follows.

For cognitive workload, the cost for each authentication round is the sum of average reaction time for all involved atomic cognitive operations. This cost represents the average thinking time of a subject required to answer a challenge. A typical authentication round consists of at least a memory retrieval operation and a simple arithmetic operation. For the graphic-based scheme, visual search is also common. According to the working memory capability theory [57, 21, 70, 69], the average reaction time is not shortened by repetitive rehearsal, when the subject has to maintain more than $4(\pm 1)$ items in his working memory. The rehearsal only improves the accuracy, which represents an inherent limitation of human capabilities. This limitation is also applied to other non-memory operations such as visual search when the item positions are shuffled in each challenge [73]. Overall, the cognitive workload of an authentication session is calculated as the product of the cognitive workload of an authentication round and the round number when the number of the secret items is larger than 5. For the schemes [10, 72] with no more than 5 secret items, we only count once for their memory retrieval operations, assuming that the secret will not be flushed out due to the limitation of working memory capacity.

Besides the reaction time, other usability measurements for cognitive workload (such as user frustration level, concentration load, and motivational effort) are usually collected from standardized testing questionnaires. However, these measurements are susceptible to many implementation and environmental factors, such as screen size, graphic or text-based interface design, and the education background of subjects. In contrast, the influence of those unstable factors has been minimized in more than a century's development of experimental psychology. So the advantage of using performance models of atomic cognitive operations is that they are *implementation-independent*. This property is necessary for a fair comparison between different LRPE designs. Consequently, our estimation of cognitive workload is very consistent with the time costs reported in the original papers [48, 71, 72, 10].

For memory demand, the cost for each scheme is a ratio $k/\lambda_{\text{op}}$ between the number of secret items, $k$, and the *accuracy rate* of corresponding memory retrieval operation within a fixed memorization time, $\lambda_{\text{op}}$. Since recognition is much easier than recall [34, 57, 69, 55, 22], it is necessary to distinguish the difficulty for different memory retrieval operations. According to [34], $\lambda_{\text{op}}$ is $29.6\%$ for recall and 84.8% for recognition. A better estimation for the memory demand could be the minimum time for the subject to remember all the secrets. However, the lower bound of memorization time is difficult to measure in experimental psychology, as the subject may not realize the precise time point when he just remembers all the secrets. An unconfident subject may take more time to rehearsal than that actually required. Other memory factors, like password interference and recall accuracy over extended periods, may also be considered but are not integrated in our current analysis framework.

Finally, an overall score, HP (standing for *Human Power*), is calculated as the product of cognitive workload score HP(C) and memory demand score HP(M). This score (HP) indicates the expected human capability requirement for a human-computer authentication scheme.

### 3.6.3 High Security at Cost of Heavy Cognitive Demand

Table 3.1 shows the security strength and HP for the representative LRPE schemes based on our quantitative analysis framework.

| | **k** | **n** | Win size | Password space | Guess Rate /round | No. of rounds /login | Reported Time /round(sec) | HP (C) /round (sec) | HP (C) /login (sec) | **HP (M)** | HP Total =M×C ($\times 10^2$) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LPN[35] | 15 | 200 | 200 | $1.463 \times 10^{22}$ | 0.50 | 20 | **23.71** | **33.423** | 668.45 | 50.68 | **338.74** |
| APW[6] | 16 | 200 | 200 | $8.369 \times 10^{24}$ | 0.10 | 6 | **35.50** | **57.928** | 347.57 | 54.05 | **187.87** |
| CAS Low[71] | 60 | 240 | 20 | $2.433 \times 10^{57}$ | 0.50 | 20 | **5.00** | **6.073** | 121.46 | 70.75 | **85.94** |
| CAS High[71] | 30 | 80 | 80 | $8.871 \times 10^{21}$ | 0.25 | 10 | **20.00** | **22.099** | 220.99 | 35.38 | **78.18** |
| SecHCI[48] | 14 | 140 | 30 | $6.510 \times 10^{18}$ | 0.50 | 20 | **9.00** | **10.638** | 212.76 | 16.51 | **35.13** |
| CHC[72] | 5 | 112 | 83 | $1.341 \times 10^{8}$ | 0.22 | 10 | **10.97** | **9.326** | 93.26 | 16.89 | **15.75** |
| PAS[10] | 4 | N/A | 13 | $4.225 \times 10^{5}$ | 0.25 | 10 | **8.37** | **6.837** | 68.37 | 13.51 | **9.24** |

Table 3.1: Tradeoff comparison of representative leakage-resilient password entry schemes for their default parameters.

Those schemes are listed in the descend order of their HP. All the schemes use their default parameter values except that the round number is adjusted to make the successful rate of random guessing to reach the same level (i.e. the authentication strength of 6-digit PIN). This adjustment is necessary to make a fair comparison as they now have the same strength to defend against an adversary without prior knowledge. The other two points in this table which need explanation are about PAS [10] and CHC [72]. In PAS, we consider the root secret for each authentication session as the predicates instead of the complete secret pairs, due to that the same predicates are used for all the rounds in an authentication. The predicates are the actual root secret of each authentication session. In CHC, the expected successful rate of guessing attack is not reported in the original paper. We estimate it based on *Statement 2*, which is $21.78\%$ derived from our simulation results. The detailed computation of the cognitive workload for those schemes is given in Table 3.2.

| | **Atomic Cognitive Operations** | **Calculation of HP (C) per round** |
|---|---|---|
| LPN[35] | Cued-recall with position, counting, mod | $(0.3964 + 0.0383 \cdot k \cdot \varphi \cdot \gamma) \cdot k + (k/2 - 1) \cdot \alpha_0 + 1 \cdot \alpha_0$ |
| APW[6] | Cued-recall with position, large addition, mod | $((0.3694 + 0.0383 \cdot k \cdot \varphi \cdot \gamma) + 1 \cdot \alpha_3 + 1 \cdot \alpha_0) \cdot k$ |
| CAS Low[71] | Parallel recognition, xor | $(0.3694 + 0.0383 \cdot k) \cdot \lceil 7.4038/4 \rceil + 1 \cdot \alpha_0$ |
| CAS High[71] | Recognition | $(0.3694 + 0.0383 \cdot k) \cdot 14.5539$ |
| SecHCI[48] | Parallel Recognition, counting, mod, small division | $(0.3694 + 0.0383 \cdot k) \cdot (\lceil 30/4 \rceil) + 2 \cdot \alpha_0 + 1 \cdot \alpha_0 + 1 \cdot \alpha_2$ |
| CHC[72] | Cued-recall, Multi-target visual search (3-based) | $((0.3694 + 0.0383 \cdot k \cdot \varphi) \cdot 5/10) + (0.583 + 0.0529 \cdot 83) \cdot 1.8$ |
| PAS[10] | Cued-recall, single-target visual search, small addition | $(0.3694 + 0.0383 \cdot 2 \cdot \varphi) \cdot 4/10 + (0.583 + 0.0529 \cdot 13) \cdot 4 + 2 \cdot \alpha_1$ |

Table 3.2: Detailed computation of cognitive workload for representative leakage-resilient password entry schemes. $\alpha_0 = 0.738$, $\alpha_1 = 0.773$, $\alpha_2 = 0.959$, $\alpha_3 = 0.924$ are the average reaction time for arithmetic problems involving 0 or 1, small addition, small division, and large addition correspondingly. $\varphi = 1.969$ is the ratio of cued recall compared to single item recognition, while $\gamma = 1.317$ is the additional penalty caused by simultaneously recalling the position of an item. For CAS Low and High, $7.4038$ and $14.5539$ are the average lengths of their decision paths, respectively.

The column "HP(C)/round" in this table shows the cognitive workload required to solve the challenge in each authentication round. It shows the average thinking time. All of them except LPN [35] and APW [6] are very close to the average time cost reported in the original literatures [48, 71, 72, 10]. For LPN, there is no report on a controlled user study. The scheme is implemented as a public web page, to which the subjects can freely access and get a reward for each successful login. There is no evidence showing that the subjects were asked to memorize their root secret (which are 15 secret positions), and then recall them in each authentication round. Thus, the average time cost reported for each round is very likely to be underestimated, as the recall operations are probably replaced by directly reading their written-down secrets. For APW, its time cost is directly estimated based on the results of LPN (with no actual user study conducted), which implies it could also be underestimated.

This table shows three tiers in these representative schemes. From bottom to top, the schemes in an upper tier have better security against password leakage at the cost of lower usability. The schemes at the bottom are PAS [10] and CHC [72], which are susceptible to both brute force and statistical attacks. When moving to the middle tier (consisting of CAS [71] and SecHCI [48]), the memory demand increases to make brute force attack infeasible. However, they are still susceptible to statistical attack as the simple challenge used in these schemes is not sufficient to hide the statistical significance of the secret. More cognitive workload is required to mix the secret items with the other items. The top tier consists of LPN [35] and APW [6], which follow all of our design principles. They are immune to both brute force and statistical attacks in practical settings, but impose significantly high usability cost.

There is an interesting finding when looking at the two schemes in the top tier. In our quantitative analysis framework, LPN has a higher HP score but a smaller password space compared to APW. This is because our security measurement is limited to brute force and two generic statistical attacks. It is still possible to find out other more efficient attacks that lower the security strength of APW. The tradeoff relation under our quantitative analysis framework may not strictly follow the order of HP, as it is always feasible to design a scheme with a lower usability for a given security strength. But it is required that the human capability should reach a *lower bound* so as to achieve a high security strength.

The above results provide quantitative evidence for the inherent limitations in the design of LRPE. They indicate the incompetence of human cognitive capabilities in using secure LRPE schemes without a secure channel in practical settings. This may also explain why the problem is still open since its first proposal [52] twenty years ago.

## 3.7 Discussion

In this work, we provided a comprehensive analysis for the inherent tradeoff between security and usability in designing a leakage-resilient password entry (LRPE) schemes. We analyzed the impacts of two types of generic attacks, brute force and statistical attacks, on the existing schemes designed for unaided humans. Unlike the specific attacks proposed before (such as SAT [32] and Gaussian elimination [47]), these two generic attacks, as demonstrated in our work, cannot be mitigated without involving considerable demand on human capabilities. We introduced five principles that are necessary to achieve leakage resilience when a secure channel is unavailable. Usability costs for these principles were analyzed. Our findings indicate that either high memory demand or high cognitive workload is unavoidable in the design of secure LRPE schemes for unaided humans. To further understand the tradeoff between security and usability, we established the first quantitative analysis framework on usability costs. Our result shows that there is a strong tradeoff between security and usability, indicating that an unaided human may not be competent enough to use a secure LRPE scheme in practical settings.

We remark that our quantitative analysis framework is still in its preliminary stage. We would like to point out two limitations in our current work: 1) Since the cognitive workload is not totally independent with the memory demand, it is possible to improve the overall score calculation instead of using the product operation (i.e. HP= M×C); 2) Error rate is currently not included in our analysis framework as it is difficult for experimental psychology to provide the general relation between thinking time and error rate. Certain approximation can be added to improve the precision of this framework in the future.

# Chapter 4

# Usable Leakage-Resilient Password Entry: Challenges and Design Metrics

## 4.1 Introduction

Under the limitations discovered in our first work, this chapter explores the feasibility of designing practical leakage-resilient password entry (LRPE) schemes with the assistance of trusted devices. One possible design based on trusted devices is to ask the users to transcribe the *one-time passwords* (OTPs) generated by *tamper-resistant* hardware tokens [59]. However, the applicability of this technique is limited due to the considerable costs of manufacturing, distributing, and managing hardware tokens for service providers, and the costs of carrying hardware tokens for users. As a result, most user accounts in the cyberspace are *not* protected by hardware-based OTPs. Moreover, hardware-based OTP has its own vulnerabilities such as subjecting to theft [51, 16]. In order to prevent such vulnerabilities, a hardware-based OTP is usually used together with a password, which is still subject to password leakage attacks.

These limitations motivates the researchers to explore the alternative design based on trusted devices, that is, using a trusted device to forms a *secure channel* between user and server. This secure channel ensures that at least part of the authentication process should be invisible to an adversary so as to prevent password leakage while maintaining acceptable usability in realistic settings. However, despite of many prior efforts [44, 61, 23, 25, 42, 13, 12], there is still no practical and widely adopted solution today. This raises a question on the practicability of adopting a secure channel in password-based authentication.

In this work, we make the first attempt to systematically investigate the challenges of designing usable LRPE schemes even when a secure channel is available. We first formalize the authentication process of LRPE schemes and classify existing schemes into three common design paradigms. We then develop a broad set of design metrics, which cover three aspects in evaluating LRPE schemes, including *quantitative* usability costs with specified security strengths, *built-in security*, and *universal accessibility*. Unlike traditional evaluation metrics, the proposed metrics are designed to identify the potential limitations of an LRPE scheme in the design phase before carrying out user studies. These metrics can also be used to dissect a scheme design into individual design elements, which facilitates a more precise and fair comparison among the classified schemes.

We apply our design metrics to existing LRPE schemes, which reveals and identifies their limitations. The major limitations include: 1) the requirement of an uncommon device feature, 2) the inoperability in certain common scenarios, and 3) the lack of trusted execution environment. This partially explains why none of these schemes are widely adopted nowadays. However, it does not necessarily imply that it is infeasible to design an LRPE scheme that is both secure and practical. Our further analysis indicates that it is possible to overcome these limitations by improving the design according to the proposed metrics.

To summarize, the contribution of this work is three-fold:

- We identify the challenges of designing LRPE schemes and classify existing

LRPE schemes into three common design paradigms.

- We develop a broad set of design metrics for LRPE schemes, which defines quantitative relation between security and usability, and extends the scope of security and usability to include built-in security and universal accessibility.
- We apply the proposed metrics on existing LRPE schemes and reveal that all the schemes have limitations that could be further improved. Our analysis provides not only a systematic understanding on existing LRPE schemes, but also a useful guide for the future research in this area.

## 4.2 LRPE Problem Overview

In this section, we define the problem of leakage-resilient password entry (LRPE) and describe its threat model. We also summarize the common design paradigms of existing LRPE schemes. At last, we provide an overview of our design metrics for the evaluation of LRPE schemes.

### 4.2.1 Definitions

In general, an LRPE scheme allows a human *user* to be authenticated to a (local or remote) computer *server* in a secure manner. During registration, user and server agree on a *password*, where each element contained in the password is referred to as a *password element*. A password element can be an image, a text character, or any symbol in a notational scheme. The user later uses his knowledge of the password to generate *responses* to *challenges* issued by the server to prove his identity. This process is referred to as *password entry*. In the case of legacy passwords, the user directly enters his plaintext password so that the adversary may capture the password via various attacks including malware, key logger, and hidden camera. *Password leakage* is the threat that a user's password is directly disclosed or indirectly inferred. The purpose of an LRPE scheme is to establish a *leakage-resilient* environment in order to mitigate or prevent password leakage during password entry.

An *authentication session* of a typical LRPE scheme requires executing multiple rounds of a challenge-response procedure in order to reach an expected *authentication strength* (e.g., $10^{-6}$ resistance against random guessing for 6-digit PIN). A *round secret* is a portion of the password which is used for an *authentication round.*

An authentication scheme is *not* considered as an LRPE scheme if a user only *transcribes* the response generated by a tamper-resistant device [59]. Such a scheme addresses a different problem which verifies a user to be the person who possesses the device and is usually used together with legacy passwords or biometrics to mitigate the risk of unauthorized access to the device, which may still be subject to the password leakage threat addressed in the LRPE problem.

### 4.2.2 Threat Model

Various potential attacks need to be addressed in the design of LRPE schemes. An adversary may use malware, key logger, or other sophisticated mechanisms to capture *messages* delivered between user and server such that the underlying password can be inferred. Prior proposals on LRPE schemes can be categorized according to whether or not a secure channel is used in the authentication process. There are quite a few LRPE schemes in the literature which are designed solely based on human cognitive capabilities without using any secure channel [35, 48, 71, 72, 10]. However, all those schemes have failed to be both secure and usable [32, 46, 5, 56, 75]. It is shown in [75] that an LRPE scheme must rely on the existence of certain secure channel to achieve both security and usability.

Although it could be difficult to establish a standard secure channel that protecting all the messages delivered between user and server, it is possible for an LRPE scheme to utilize a partial secure channel. The requirement of a *partial* secure channel is weaker than a standard secure channel, as it only requires that a portion of messages delivered between user and server be invisible to an adversary. For example, the use of a partial secure channel may ensure that the leakage resistance of an LRPE scheme is preserved even after allowing an adversary to observe most messages during password entry as long as certain critical messages are not disclosed. A partial secure channel is usually *unidirectional* either from server to user or from user to server.

In the presence of a partial secure channel, it is possible to achieve the optimal security objective, *no password leakage* during password entry. No password leakage with a partial secure channel means that if the portion of messages protected by the partial secure channel are not disclosed, a secure LRPE scheme should provide the same leakage resilience as *one-time pad* [54], where the most efficient attacks for an adversary to learn the password are online dictionary attacks. This study focuses on LRPE schemes using such a partial secure channel and excludes LRPE schemes without using any form of secure channel unless explicitly mentioned.

There are LRPE schemes in the literature based on weak threat models, where the requirements of secure channels are not precisely specified. An example of such schemes considers the threat of cognitive shoulder-surfing [58, 26], where the adversary is assumed be not able to observe the entire password entry due to his cognitive limitations; however, it is not clear what is the *exact* part of password entry that is invisible to the adversary. We exclude these schemes in our discussion.

In addition to the attacks mentioned at the beginning of this subsection that happen during password entry, password leakage may also be caused by other types of attacks, such as social engineering, phishing or even non-technical attacks such as dumpster diving [49]. Although their mitigation technologies such as secure URL checker and spam filter have become standard components of modern computer systems, some of these attacks may not be completely preventable by technical solutions alone and are orthogonal to the password entry problem. Another example is the database reading attack, where the adversary intrudes into the back-end databases to compromise all user passwords. These attacks are out of the scope of this work.

### 4.2.3 Common Design Paradigms

As analyzed in the previous subsection, it is necessary for a practical LRPE scheme to use a partial secure channel. The key idea of an LRPE scheme with a partial secure channel is to hide certain messages during password entry from the adversary. These *hidden messages* break the *correlation* between the password and the information observable to the adversary so that the adversary will not be able to infer the password. The three common paradigms are described as follows (see Figure 4.1):

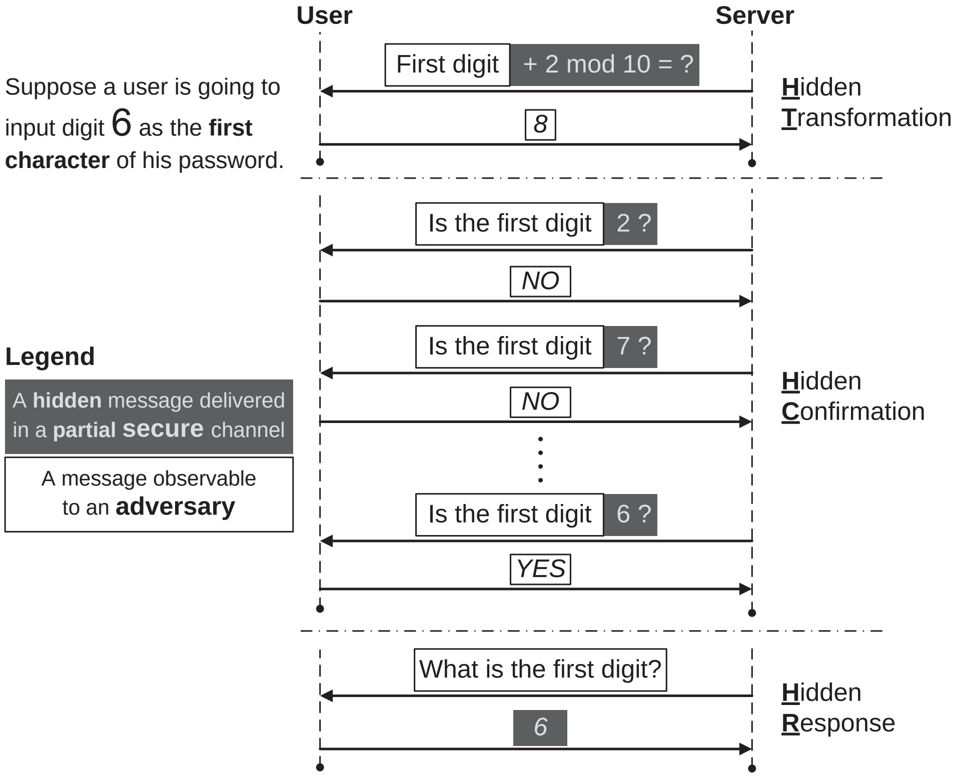


Figure 4.1: Examples of LRPE schemes following three common design paradigms. For a message contains both white and grey boxes, only the message in the white part is observable to an adversary, while the message in the grey part is delivered via a partial secure channel. Take "*First digit* + 2 mod 10 = ?" as an example. An adversary knows a user's next input is related to the *first digit* in the password, but he does not know the hidden transformation, which is "+ 2 mod 10". Thus, even if the adversary observes the answer 8, he is not able to infer the entered password element, which is 6.

1. **Hidden transformation (HT):** Hidden transformations are delivered via a

partial secure channel from server to user. A hidden transformation is a part of a challenge, which transforms a password element into another form that is not *correlated* with the original password element if the question itself is not disclosed. The correct response for each password element is calculated according to both the corresponding password element and the hidden transformation.

2. **Hidden confirmation (HC):** Hidden binary questions are delivered via a partial secure channel from server to user. These questions enumerate all possible candidate elements in the password alphabet in a random order. A user answers *Yes/No* to the question depends on whether the password element he wants to input appears in the question. Multiple confirmation questions are usually required for inputting a single password element.

3. **Hidden response (HR):** The *entire* response is delivered over a partial secure channel from user to server. An instance following this paradigm is a keypad fully covered by a glove. If the vision channel from the outside of the glove is the only way that the adversary can observe the password entry, the glove forms a partial secure channel that hides all the user inputs from the adversary.

The first two paradigms take the strategy of hiding the challenges, while the last paradigm hides the responses. Referring to Figure 4.1 again, intuitively, a scheme in these paradigms will be secure as long as the messages (shown in the grey boxes) delivered via the corresponding partial secure channel are not disclosed. The detailed characteristics of these paradigms will be analyzed together with the corresponding design metrics introduced in the following sections.

### 4.2.4 Design Metrics Overview

Our design metrics characterize and assess the major design factors in LRPE schemes, which are organized in three different aspects and introduced in the following three sections, respectively. The first aspect is related to two key security features of an LRPE scheme, password space and leakage resistance. A large password space is an essential requirement for any secure password scheme. The size of password space generally increases at the expense of users' memory effort. The second security feature, leakage resistance, relates to both the cognitive workload and the resistance property of interaction channels. Higher leakage resistance of an LRPE scheme usually implies either a higher cognitive workload or a higher deployment cost of interaction channels. The first aspect of our metrics addresses the quantitative relations between these two security features and their associated costs. As shown in Figure 4.2, these design factors are marked with section numbers **4.3.1**, **4.3.2**, and **4.3.3**. Our research in this aspect gives the lower bound condition for usability costs to achieve a given security strength, but it does not provide guarantee to ensure that the security strength is not affected by user behavior. This problem is addressed in the second aspect of our metrics, build-in security, as shown in the design factors marked with section numbers **4.4.1** and **4.4.2** in Figure 4.2. The last aspect extends the scope of usability to universal accessibility. We examine how the variety and availability of users' capabilities, devices, and other environmental factors influence the practicability of LRPE schemes, as shown in the design factors marked with section numbers **4.5.1**, **4.5.2**, and **4.5.3** in Figure 4.2.

## 4.3 Relations between Security Strength and Usability Costs

The relation between security and usability is not necessarily a strict tradeoff. It is possible to improve security without sacrificing usability if it does not reach the usability lower bound for a given security strength. We discuss the usability costs associated with the key security features of LRPE schemes below.

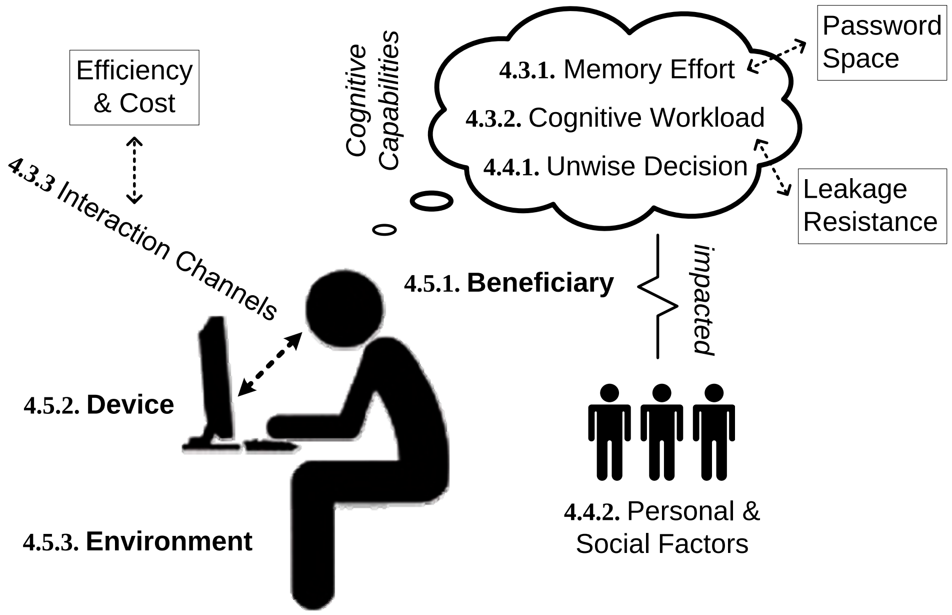


Figure 4.2: Major design factors in LRPE schemes

### 4.3.1 Password Space and Memory Effort

A large password space is an important security feature against brute force attacks, where *large* means it is computational infeasible for an adversary to enumerate all possible candidates in a practical setting. Given an alphabet with $n$ elements and a password length $k$, the password space reaches the maximum size $n^k$ if each password is a random ordered sequence that allows duplicate elements. Otherwise the password space will be smaller than $n^k$, which means the security strength against brute force attacks can be improved for the usability cost of memorizing a $k$-length password. A *space-memory ratio* metrics is defined to characterize the relation between a password space and users' memory effort.

**Space-Memory Ratio (SMR):** *Given an alphabet with* n *elements and a password length* k*, a space-memory ratio is the size of the password space divided by password length* k.

The maximum value of SMR is $n^k/k$. Given a specific resistance against brute force attacks (i.e. a specific size of the password space), a higher value of SMR means the less memory effort for users. We use the Undercover scheme [61] as an example for SMR calculation. A password in this scheme consists of 5 distinct images to achieve the authentication strength of 4-digit PIN. According to its design, the size of its password space is $C_n^k$ for memorizing $k$ distinct images from a pool of $n$ images. Thus, its SMR value is $C_n^k/k$.

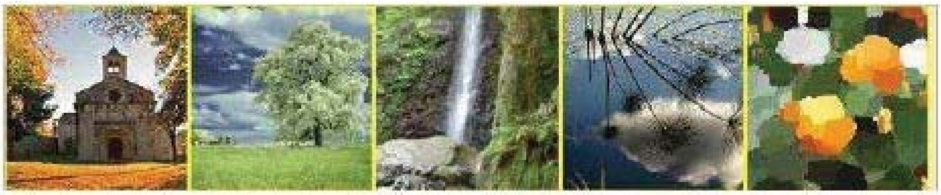


Figure 4.3: Password composition of the Undercover scheme [61]

The definition of SMR does not distinguish two primary memory retrieval operations – recall and recognition[1]. Given the same password length, recall is usually slower than recognition [22] when the elements in the password are independent such that the user has to memorize them individually. If the user is able to find out the logical relation between these elements and memorize them as chunks, the difference between recall and recognition is no longer significant. This effect is called *chunking* in psychology [31]. It is also possible for the user to perform a faster recall than recognition if a simple logical relation exists in the password required by recall but does not exist in another password required by recognition. For example, if 2047 is the room number of a user's apartment, it could be easier for him to recall it as his 4-digit PIN, compared with recognizing 5 distinct images shown in Figure 4.3. Therefore, recognition is not always an easier choice compared to recall. Likewise, SMR does not distinguish the types of elements, e.g. text characters or images, as their difference cannot be deterministically characterized. Despite these differences, a shorter password generally implies less memory effort.

On the other hand, a longer password does not necessarily imply high guessing-resistance, but it is the essential requirement for a larger password space. A recent research [41] shows that a longer password without any other composition restric-

[1] Memory recall and recognition are the two principal methods of memory retrieval [8]. Recall is defined as reproducing the stimulus items; Recognition is the process to correctly judge whether a presented item have been encountered before.

tion tends to have a higher guessing-resistance in real life compared to a shorter password with a complex composition policy. Therefore, the password space calculated based on the password length also provides a reasonable estimation for the effective resistance against brute force attacks.

### 4.3.2 Leakage Resistance and Cognitive Workload

No password leakage is the major security objective of LRPE schemes. We now discuss the requirements to achieve no password leakage for the three common paradigms given in Section 4.2.3.

In the hidden transformation (HT) paradigm, an adversary should not be able to learn any information from the following five leakage sources: the challenge alone, the response alone, the hint alone, the challenge-response pair, and the challenge-hint pair. Here a *hint* means a hidden transformation, which is a part of a challenge that assists the user to calculate the final response. For example, a hint could be a transformation like "*plus 2 mod 10*" as illustrated in Figure 4.1. Note that it is possible for the adversary to access the hint by attempting to use the LRPE scheme as the scheme always shows the hint during authentication. It is thus required that the hint should not be embedded with any knowledge about the underlying password.

We derive the following necessary conditions for an LRPE scheme in the HT paradigm to achieve no password leakage:

**No-Leakage Conditions**: *Let $X_i^t = \{x_{i1}, \ldots, x_{it}\}$ be a set of $t$ elements that may appear in a challenge, $Pr(X_i^t|c_m)$ be the probability that all the elements in $X_i^t$ appear together in a challenge $c_m$, $Pr(h_i|c_m)$ be the probability that a hint $h_i$ appears with a challenge $c_m$, and $Pr(r_i|c_m)$ be the probability that a response $r_i$ appears with a challenge $c_m$, the following rules must be satisfied so as to achieve no password leakage.*

1. **Uniform-distribution rule**[2]: *For any $i$, $j$, $m$, $t \in \mathbb{Z}^+$, $Pr(X_i^t|c_m) =$*

[2] When $t = 1$, this rule means each single element is uniformly distributed in the challenge.

$Pr(X_j^t|c_m)$, $Pr(h_i|c_m) = Pr(h_j|c_m)$, *and* $Pr(r_i|c_m) = Pr(r_j|c_m)$

2. **Zero-correlation rule**: *For any* $i$, $j$, $k$, $m$, $t \in \mathbb{Z}^+$, $Pr(X_i^t|c_m, r_k) = Pr(X_j^t|c_m, r_k)$ *and* $Pr(X_i^t|c_m, h_k) = Pr(X_j^t|c_m, h_k)$

The uniform-distribution rule ensures that the challenge alone, the response alone, and the hint alone do not leak any information related to the underlying password. The zero-correlation rule further prevents password leakage from the challenge-response pair and the challenge-hint pair. The only remaining source is the correlation between responses and hints, which is protected by the partial secure channel from the server to the user.

LRPE schemes in the other two paradigms, hidden confirmation (HC) and hidden response (HR), are always able to achieve no password leakage if the corresponding partial secure channels are not compromised. Specifically, for the HR paradigm, the entire response should be delivered via the partial secure channel.

We use the VibraPass scheme [25] as a counterexample to show how password leakage happens when no-leakage conditions are not satisfied. This scheme is in the HT paradigm, which utilizes the vibration function of an extra mobile phone to construct a secure haptic channel. A password in this scheme consists of $k$ text characters. To enter a $k$-length password, extra $l$ lie rounds are added. In each lie round, a user is asked to input a random character instead of the next correct character that he is supposed to input. The user knows the current round is a lie round when he feels his mobile phone vibrating. The total round number is $k + l$, where the positions of $l$ lie rounds are randomized in all $k + l$ rounds. For example, given a password 1234 and a round sequence "$0, 1, 0, 0, 1, 0$" where 1 means a lie round and 0 means a normal round, the user should input "$1, x, 2, 3, y, 4$" to pass the authentication, where $x$ and $y$ are two random digits. This scheme does not satisfy the **uniform-distribution** rule. The underlying password characters *must* appear in the response every time while the other fake characters may not. So

When $t = 2$, this rule means each pair of elements also uniformly appears in the challenge.

the condition $Pr(r_i|c_m) = Pr(r_j|c_m)$ does not hold, where $r_i$ and $r_j$ are a user's responses of inputting individual characters. An adversary can simply maintain a counting table to count the frequencies of individual characters appearing in the responses, and then infer the password by reconstructing the order among the most frequent characters.

Since no password leakage is achievable, we do not consider other weaker security objectives in the following discussion but focus on measuring the cognitive workload to accomplish this objective. A recent work [75] provides a quantitative analysis framework for the usability costs of an LRPE scheme *without* a partial secure channel. The framework divides the authentication procedure of an LRPE scheme into a sequence of atomic cognitive operations that can be quantitatively measured based on the results from experimental psychology. We adapt this methodology and define a *cognitive operation list* metric to estimate a user's cognitive workload.

**Cognitive Operation List (COL):** *Given an LRPE scheme that requires a user to solve* $\mathrm{m}$ *challenges where each of them consists of* $\mathrm{s}$ *atomic cognitive operations, a cognitive operation list contains all* $\mathrm{s} * \mathrm{m}$ *atomic cognitive operations involved in a successful authentication.*

Since the amount of mental effort for the same atomic cognitive operation may vary with different users, this metric only enumerates all the operations. It is difficult to derive a consistent metric that calculates a more quantitative value based on this list. For example, a metric that merely calculates the total number of operations may not be useful as the overall cognitive workload of a shorter COL could be higher than the workload of a longer COL. Nonetheless, for the schemes using similar types of cognitive operations, a COL with fewer operations generally implies a lower cognitive workload. For the HT paradigm, the minimal value of $s$ is 3, which involves one operation of reading the hint from the partial secure channel, one memory retrieval operation for the round secret, and one simple cognitive calculation for response obfuscation. For the HC paradigm, the minimal value of $s$ is $1 + n/2$, where 1 is for the recall of the round secret and $n$ is the size of password alphabet. $n/2$ is the expected number of recognitions that are required to confirm the correct round secret from a random ordered list. For the HR paradigm, the minimal value of $s$ is 2, which involves one recall of the round secret and one response by operating the partial secure channel. Among these three paradigms, the HR paradigm may have the shortest COL, which also has the highest requirement on the availability of partial secure channels as analyzed in Section 4.6. Compared to the HC paradigm, the HT paradigm has an advantage to support a large alphabet under the same usability cost, which makes it easier to scale up to a large password space.

We use the PhoneLock scheme [12] as an example for COL enumeration. This scheme is in the HC paradigm, where a headset is utilized to construct a secure acoustic channel (see Figure 4.4). A password in this scheme is a PIN. When a user presses on a specific cue region on the ring, the headset will play a random spoken number picked from 0 to 9. If the user hears a number he wants to input, he selects that number by pressing the circle in the center, and then finishes the current authentication round. The mappings between spoken numbers and cue region positions are reshuffled in each round and remain fixed within each round. According to its design, in each round, 1 recall for the round secret (i.e. the next number a user is supposed to input) is required and $n/2$ cue regions on average have to be explored in order to find the current round secret from an $n$-sized alphabet. Thus, the COL of this scheme contains $1 + n/2$ operations described above for each round, and these operations repeat $m$ times for a successful authentication session.

Since the length of COL also depends on the round number $m$ (i.e. the number of challenges), it is possible to further reduce the cognitive workload if we can reduce $m$ for a given authentication strength. Such issue will be addressed by another metric, *screen utility rate*, which is introduced in Section 4.5.

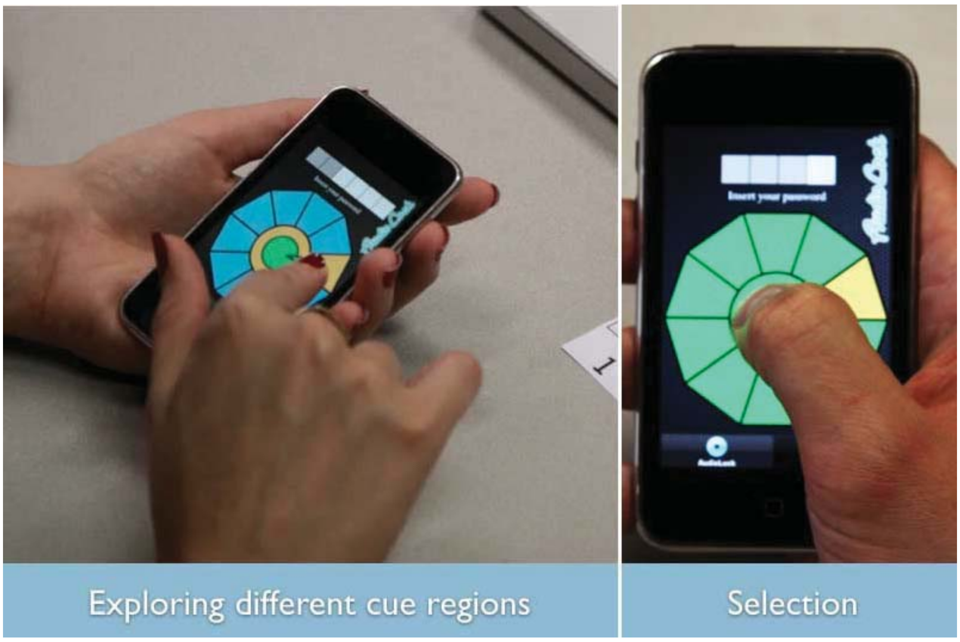


Figure 4.4: The usage of the PhoneLock scheme [12]

### 4.3.3 Effectiveness and Costs of Interaction Channels

The choice of interaction channels between user and user interface may have a significant influence on the efficiency of the authentication process of LRPE schemes. From users' perspective, three types of interaction channels are used in a typical LRPE scheme, which include public input channels for challenge, partial secure channels for a user either to receive inputs or to provide outputs, and public output channels for response. A good interaction channel for an LRPE scheme should satisfy two requirements: 1) it has a high bandwidth for efficient message delivery, and 2) it has high reliability and minimum demand on human capabilities so that human beings can use it easily in various environments. In addition, a channel is further required to be difficult for the adversary to compromise if it is used as a partial secure channel.

Existing user interfaces require that a user gets inputs from vision [42, 44, 23, 61, 25], acoustics [12], or haptics [13], and provide outputs via acoustics or motion [42, 44, 23, 61, 25, 12, 13]. For the input channel, evidences from psychology show that vision is the fastest channel to reliably collect information for nonblind users. This phenomenon is called as *visual dominance*. In perception and information processing, vision has been shown to dominate over acoustics [18] and haptics [29]. For the output channel, motion is shown to be a more reliable and faster channel compared to the acoustics channel as average human beings have better control over body, especially hand, than sound [66] and it has better resistance against environmental noises. Among all the possible motions, clicking [42, 61, 25, 12, 13] is the simplest which only requires the user to move one finger without a high precision control as required by other motions like shaking in a specific way. Hence, the optimal choice for interaction channels in a general sense is vision for input, and clicking for output. Any other choice for interaction channels may be considered *low efficiency* unless they are designed for specific application scenarios.

In realistic settings, an interaction between user and user interface may be captured through multiple *leakage* channels from an adversary's perspective. For example, a clicking action on a keypad may be intercepted from the vision channel (where the adversary installs a hidden camera) and from the haptic channel (where the adversary installs a sensitive haptic board above the original keypad). Therefore, a secure LRPE scheme should protect all these channels that may potentially cause password leakage. Considering an LRPE scheme that uses a glove to protect password entry, since an adversary can steal the password by watching the password entry or installing an external key logger, the mere act of blocking the vision channel by a glove is not sufficient.

Since a user's interaction channel may correspond to multiple leakage channels, all these leakage channels should be transformed into partial secure channels by the design of LRPE schemes. Although it is difficult to judge the leakage resistance of these partial secure channels in a general setting, the risk of password leakage can be reduced if a fewer number of partial secure channels are involved in a scheme design. The list of partial secure channels required by an LRPE scheme provides an estimation for the reliability of the scheme. We consider a scheme having *higher* *reliability* if it involves fewer partial secure channels and its partial secure channels have higher resistance against password leakage. In Section 4.5, we will discuss other factors related to interaction channels, including 1) the human capability requirements for operating an interaction channel, and 2) the availability of required device features.

## 4.4 Built-in Security

Built-in security requires that the security strength of an LRPE scheme should not rely on user behavior. If a scheme requires a user to perform an optional action to achieve its security strength, this security strength is unreliable as the user may not act appropriately due to the inconsistency with personal habits and the sensitivity on violations of social norms.

### 4.4.1 Inconsistency with Personal Habits

Most users do not have the habit of thinking of security first. Security mechanism such as user authentication is usually a minor task for users. What users care most is obtaining services after authentication [24].

There are two common inconsistencies between users' habits and security design. The first one is *impatience*, which means a user may not perform any optional actions which he is supposed to perform. Some common optional actions such as reading a manual, and checking the integrity of input device may make users impatient. A typical example is Error-Correcting-Challenge [35], which is the only existing scheme that is designed to defend against an *active* adversary. The adversary is allowed to arbitrarily manipulate the environment for password entry, such as modifying a challenge issued by a legitimate server. This scheme requires a user to verify the integrity of a challenge by solving linear equations before answering the challenge. A challenge in this scheme consists of $w \times h$ squares, where each square contains $10 \times 10$ digits. The digits in each square

are generated by a formula $L(x, y) = ax + by + c \mod 10$, where $L(x, y)$ is the digital value at location $(x, y)$. $a$, $b$, and $c$ are three random digits drawn uniformly for the current square. A user is asked to test the linearity of these digits in every square by choosing a random point $(x, y)$ and a random offset $r$ and checking whether $L(x, y) = L(x + r, y) - L(r, y) + L(0, y) \mod 10$ or $L(x, y) = L(x, y + r) - L(x, r) + L(x, 0) \mod 10$. The user will answer the challenge only after the challenge passed a sufficient number of linearity tests on all $100wh$ digits. During this process, a user may become impatient due to the high cognitive workload.

The second inconsistency is about users' inability of *generating random numbers* [7], where certain LRPE schemes rely on users to make random choices. For example, the LPN scheme [35] asks a user to calculate the responses by two different algorithms $A$ and $B$, respectively. Given a challenge, with probability $\eta$, the user randomly picks algorithm $A$; otherwise he uses algorithm $B$. The user passes the authentication if the ratio of correct responses generated by algorithm $A$ is not smaller than $\eta$. The leakage resistance of this scheme relies on the randomness in users' choices between two algorithms, which may be significantly undermined if a user always follows a fixed pattern to choose these two algorithms. However, it is usually difficult for average users to make such "random" choices specified by the scheme design.

User education will alleviate the problem to some extent, but the outcome is uncertain. A user may still make mistakes or be overconfident. Any LRPE scheme with high reliability in security should not rely mainly on user education. It is necessary to convert optional actions into compulsory actions if they are critical to secure LRPE schemes.

### 4.4.2 Violations of Social Norms

Social norms are also common concerns impeding a user from performing certain optional protection actions. A recent field study on ATM usage [24] found that a user is not willing to shield a keypad if he is accompanied by his friends. The user may think that a shielding gesture would be misinterpreted as a sign of distrust to his friends. This situation is more likely to happen among users who have an intimate relation with each other. Social norms may vary with different cultures, but their impact on LRPE schemes is similar, which may prevent users from performing certain optional protection actions required by LRPE schemes. Hence, a secure LRPE scheme should make necessary actions mandatory so as to achieve security objectives. This is also a solution to avoid potential misinterpretation on social norms.

The ShieldPin scheme [42] is an example that addresses this issue. The keypad in this scheme appears only when the protective gesture is being detected by the touchscreen (see Figure 4.5). Once a user raises his hand from the touchscreen, the keypad will immediately disappear so that password entry is always protected by the required action.

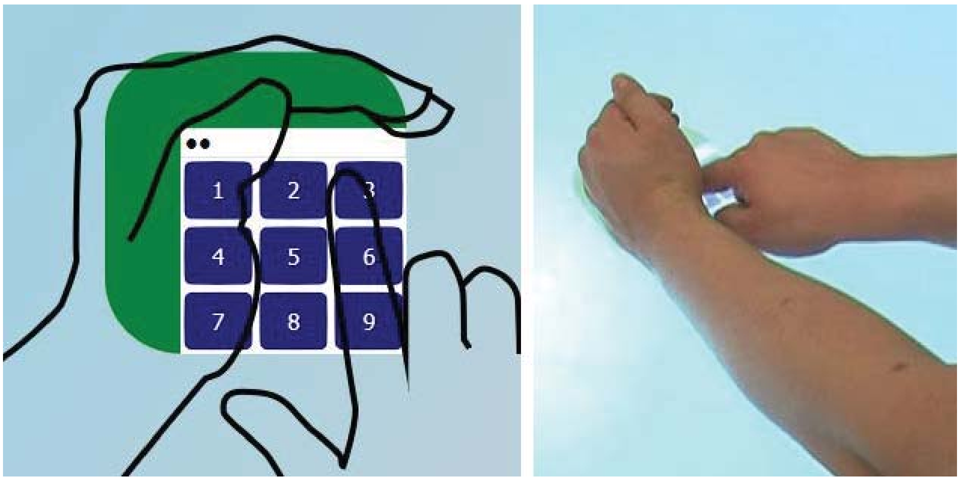


Figure 4.5: User interface in the ShieldPin scheme [42]

## 4.5 Universal Accessibility

Universal accessibility is intended to benefit the majority of users in the design of LRPE schemes. Specifically, it requires a scheme to be accessible even in a non-ideal environment such as situations when a user is not able to use all his capabilities or when environmental noise is high. Traditional laboratory user study that only considers ideal environment for unhampered users may not be sufficient to fully evaluate the usability of LRPE schemes in practice. We discuss three general aspects of universal accessibility below.

### 4.5.1 Beneficiary Scope

Beneficiary scope specifies who has the capabilities to use an LRPE scheme. The success of legacy passwords is largely attributed to its wide beneficiary scope, as it imposes minimum requirement on human capabilities in a general sense. Anyone who can see and move a single finger can use legacy passwords. A narrower beneficiary scope means some current users of legacy passwords cannot use the LRPE scheme. A practical LRPE scheme should attempt to preserve a similar beneficiary scope. Any LRPE scheme that requires extra human capabilities may not be appealing to the majority.

For example, the PressureGrid scheme [42] requires precise cooperation of multiple fingers (see Figure 4.6). To select a specific cell, a user is asked to apply additional pressure on one specific finger per hand. This operation could be difficult especially for elders, children, and those with physical (not *cognitive*) disability such as a person who loses one of his fingers.

### 4.5.2 Device Availability

Any LRPE scheme runs with at least one device, where the user uses a system protected by the LRPE scheme. This device is referred to as the *primary device*. Some existing LRPE schemes [12, 25] also require an extra device to form a partial secure

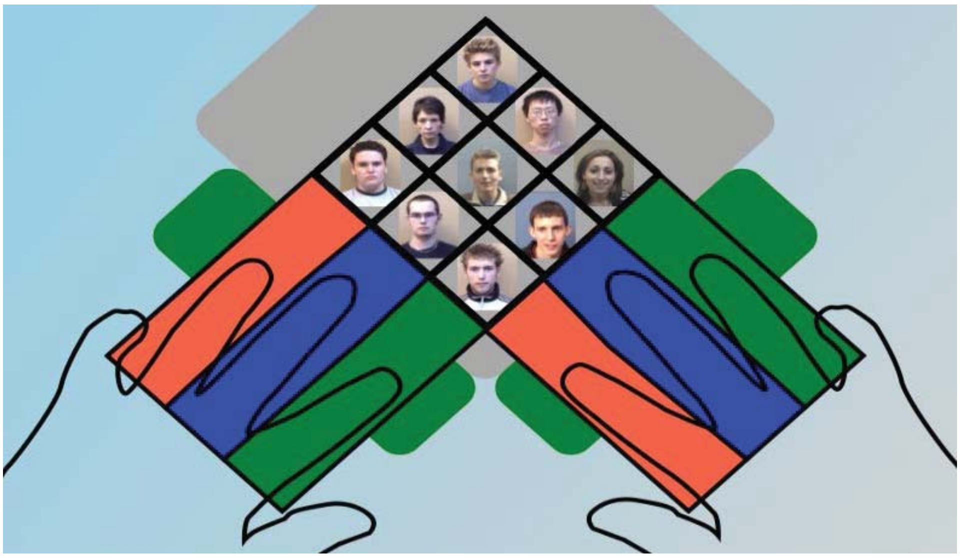


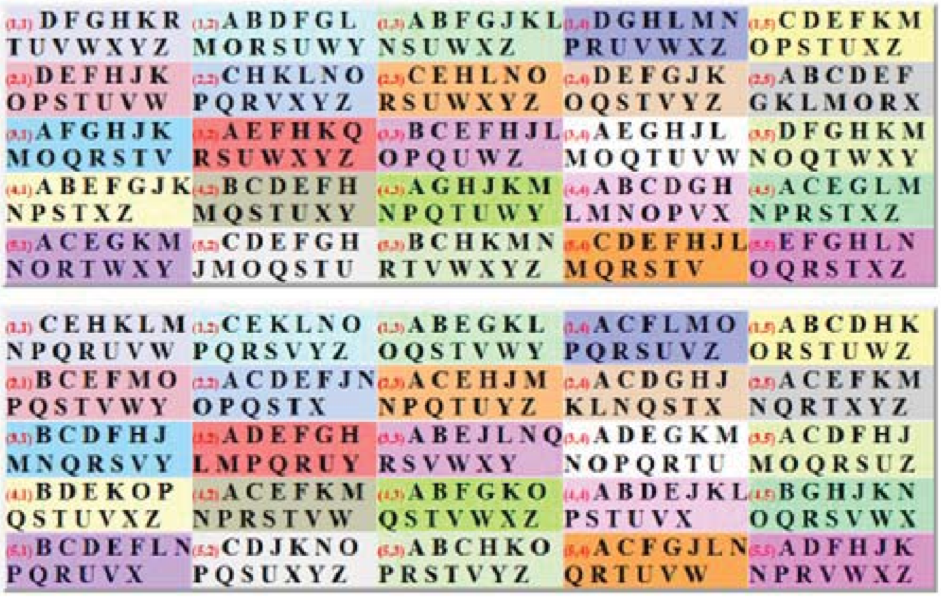
Figure 4.6: The usage of the PressureGrid scheme [42]

channel, which is referred to as the *secondary device*. The use of secondary device lowers device availability, even if the device is free of charge. This is because the secondary device must be carried by users and it subjects to extra risks such as theft, which in turn may cause security or accessibility problems. A good design should avoid the use of secondary device and focus on reusing the existing features of the primary device. Since device features *evolve* with time, it is possible to support more advanced security properties when the required features become available.

Even if a primary device is equipped with sufficient features to support an LRPE scheme, it usually has its own functional limitations. For example, the most popular mobile devices such as smartphones and tablets are usually equipped with a small screen. Therefore, one may not expect a primary device be equipped with a large screen like desktop computers. We define *screen utility rate* as an important indicator for the requirement on the primary device.

**Screen Utility Rate (SUR):** *Given a screen with* $\mathbf{N}$ *cells for displaying individual elements in a challenge of an LRPE scheme, which has a probability of* $\rho$ *for the adversary to use random guessing to find the correct response on average, the screen utility rate is* $\frac{1/\rho}{\mathbf{N}}$. *Control elements such as finish button and backspace button are not counted in* $\mathbf{N}$.

The maximum value is $1.0$ when $\rho = 1/N$. A higher value of SUR indicates a lower requirement on the screen size for achieving the specified authentication strength after each challenge. This metric characterizes how efficient an LRPE scheme is able to achieve certain security strength with a fixed-sized screen. A high value of SUR results in a small number of rounds in authentication, which makes it easier to adapt to small-screen devices. Although a larger value of SUR may not necessarily implies a better usability, but it address the design restriction from the screen size, which is also directly related to the form factor of the device. This metric raises the awareness of reducing unnecessary visual redundancy in the LRPE schemes as illustrated by the counterexample of the PAS scheme [10][3] (see Figure 4.7).



| | | | | |
|---|---|---|---|---|
| (1,1) D F G H K R<br>T U V W X Y Z | (1,2) A B D F G L<br>M O R S U W Y | (1,3) A B F G J K L<br>N S U W X Z | (1,4) D G H L M N<br>P R U V W X Z | (1,5) C D E F K M<br>O P S T U X Z |
| (2,1) D E F H J K<br>O P S T U V W | (2,2) C H K L N O<br>P Q R V X Y Z | (2,3) C E H L N O<br>R S U W X Y Z | (2,4) D E F G J K<br>O Q S T V Y Z | (2,5) A B C D E F<br>G K L M O R X |
| (3,1) A F G H J K<br>M O Q R S T V | (3,2) A E F H K Q<br>R S U W X Y Z | (3,3) B C E F H J L<br>O P Q U W Z | (3,4) A E G H J L<br>M O Q T U V W | (3,5) D F G H K M<br>N O Q T W X Y |
| (4,1) A B E F G J K<br>N P S T X Z | (4,2) B C D E F H<br>M Q S T U X Y | (4,3) A G H J K M<br>N P Q T U W Y | (4,4) A B C D G H<br>L M N O P V X | (4,5) A C E G L M<br>N P R S T X Z |
| (5,1) A C E G K M<br>N O R T W X Y | (5,2) C D E F G H<br>J M O Q S T U | (5,3) B C H K M N<br>R T V W X Y Z | (5,4) C D E F H J L<br>M Q R S T V | (5,5) E F G H L N<br>O Q R S T X Z |

| | | | | |
|---|---|---|---|---|
| (1,1) C E H K L M<br>N P Q R U V W | (1,2) C E K L N O<br>P Q R S V Y Z | (1,3) A B E G K L<br>O Q S T V W Y | (1,4) A C F L M O<br>P Q R S U V Z | (1,5) A B C D H K<br>O R S T U W Z |
| (2,1) B C E F M O<br>P Q S T V W Y | (2,2) A C D E F J N<br>O P Q S T X | (2,3) A C E H J M<br>N P Q T U Y Z | (2,4) A C D G H J<br>K L N Q S T X | (2,5) A C E F K M<br>N Q R T X Y Z |
| (3,1) B C D F H J<br>M N Q R S V Y | (3,2) A D E F G H<br>L M P Q R U Y | (3,3) A B E J L N Q<br>R S V W X Y | (3,4) A D E G K M<br>N O P Q R T U | (3,5) A C D F H J<br>M O Q R S U Z |
| (4,1) B D E K O P<br>Q S T U V X Z | (4,2) A C E F K M<br>N P R S T V W | (4,3) A B F G K O<br>Q S T V W X Z | (4,4) A B D E J K L<br>P S T U V X | (1,5) B G H J K N<br>O Q R S V W X |
| (5,1) B C D E F L N<br>P Q R U V X | (5,2) C D J K N O<br>P Q S U X Y Z | (5,3) A B C H K O<br>P R S T V Y Z | (5,4) A C F G J L N<br>Q R T U V W | (5,5) A D F H J K<br>N P R V W X Z |

13 X 5 X 5 X 2 = 650 displayed letters
0.25 probability of guessing the correct response from
4 possible answers

**SUR** = (1/0.25)/650 = 1/162.5 = 0.0062

Figure 4.7: Visual redundancy in the PAS scheme [10]

[3] Since the PAS scheme [10] does not use a partial secure channel, it is only used as a counterexample here, but will not be included in the analysis in Section 4.6.

### 4.5.3 Environmental Adaptation

Laboratory user study is usually conducted in a quiet room and the user is given sufficient time to perform a single task in each test. However, this may not be the case in daily usage. Users may act differently when they do not have peace in mind or stay in a quiet room. Below we summarize common environmental metrics which affect users' perception of security. 1) *Impact of time pressure*: a user tends to act hastily under time pressure, which may lead to mistakes. 2) *Impact of distraction*: unexpected distraction interferes with a user's mind when answering challenges. 3) *Impact of mental workload*: mental workload consistently interferes with a user's mind during answering challenges. 4) *Impact of environmental noise*: environmental noise may render certain interaction channels such as acoustics and haptics imprecise or even unusable. An example of haptics-based user interfaces illustrated in Figure 4.8 requires a stationary environment for a user to precisely feel haptic inputs. 5) *Impact of hampered capability*: a user's capability may be hampered even if he is not handicapped. For example, a user may only use one hand in authentication when he uses the other hand to carry a bag. These environmental metrics are important in the evaluation of LRPE schemes so as to obtain credible results in real-world scenarios. Among these metrics, only the last two metrics can be measured in the design phase, which are used to evaluate existing schemes. The first three metrics will be discussed in Section 5.6.

## 4.6 Using the Metrics: Evaluation of Existing LRPE Schemes

In this section, we apply our design metrics to representative LRPE schemes [44, 61, 23, 25, 42, 13, 12] that attempt to establish a partial secure channel between user and server. Table 4.1 shows the results of our analysis.

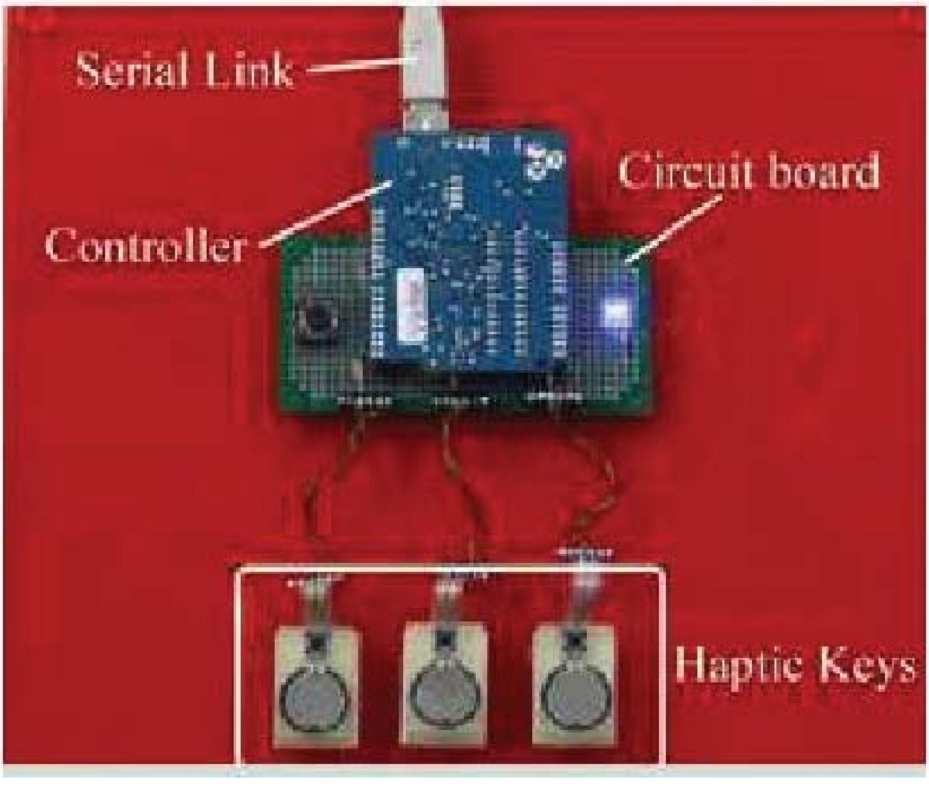


Figure 4.8: User interface of the HapticKeypad scheme [13]

Table 4.1: Evaluation and comparison of representative leakage-resilient password entry schemes

| | **Paradigm** | **No-leakage** | **Space-memory ratio** | **Cognitive operation list** | **Interaction channels** | **Partial secure channels** | **Required human capabilities** | **Extra device** | **Device feature** | **Screen utility rate** | **Environmental adaption** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ShieldPIN [42] | HR | Yes | $n^k/k$ | Recall + shield [$2m$] | Vision (I)<br>Click (O) | Vision, **Haptics** | Shielding gesture | No | Touch-screen | 1.0 | Fine |
| PressureGrid [42] | HR | Yes | $n^k/k$ | Recall + click with pressure control [$2m$] | Vision (I)<br>Click (O) | Haptics, **Vision** | Move multiple fingers | No | Pressure sensitive screen | 1.0 | Unusable in shaking |
| Gaze Tracking [44, 23] | HR | Yes | $n^k/k$ | Recall + move gaze point [$2m$] | Vision (I)<br>Gaze (O) | Vision | Gaze movement | No | Gaze tracker | 1.0 | Unstable in shaking |
| Haptic Keypad and Wheel [13] | HC | Yes | $n^k/k$ | Recall + $n/2$ recognition [$(1+n/2)m$] | Haptics (I)<br>Click (O) | Haptics | Haptics recognition | No | Haptic motor | 1.0 | Unusable in shaking |
| PhoneLock, SpinLock [12] | HC | Yes | $n^k/k$ | Recall + $n/2$ recognition [$(1+n/2)m$] | Acoustics(I)<br>Click (O) | Acoustics | Acoustics recognition | One headset | Sound player | 1.0 | Fine |
| Undercover [61] | HT | No | $C_n^k/k$ | Recognition + feeling + lookup [$3m$] | Vision (I)<br>Click (O) | Haptics | Haptics recognition | No | Haptic motor | 1.0 | Unusable in shaking |
| VibraPass [25] | HT | No | $n^k/k$ | $m$ recall + $(m+l)$feeling + $(m+l)$generate a real or fake input | Vision (I)<br>Click (O) | Haptics | Haptics recognition | One phone | Phone vibration | 1.0 | Unusable in shaking |
| CuePin [42] | HT | Yes | $n^k/k$ | Recall + shield + lookup [$3m$] | Vision (I)<br>Click (O) | Vision | Shielding gesture | No | Touch-screen | 0.5 | Fine |

In Table 4.1, $m$ is the round number required to achieve a specific authentication strength. The value of $m$ is decided by the screen utility rate (SUR). Given a fixed-size screen, a larger value of SUR generally implies a smaller value of $m$. The "*cognitive operation list*" column only lists the cognitive operations for one challenge, and these operations are repeated in all $m$ challenges. The total number of operations for a successful authentication session is given in the square brackets at the end of the operation list. There is an exception in this column for the VibraPass scheme [25]. For this scheme, the full operation list is given, and the total round number is $m + l$, where $m$ is the number of rounds for inputting all the characters in the password and $l$ is the number of extra lie rounds for confusing the adversary. More details about this scheme can be found in the example described in Section 4.3.2. In the "*interaction channels*" column, letter *I* or *O* in the parentheses indicates a channel is used by a user to get inputs or to provide outputs respectively. The "*extra device*" column describes whether it requires a secondary device for user authentication. The "*device feature*" column shows the special features required by the primary device and the secondary device if it exists. "*Fine*" in the "*environmental adaption*" column means there are no foreseeable extra environmental restrictions on scheme usage compared to legacy passwords. "*Unusable/unstable in shaking*" means the effectiveness of a scheme is significantly impacted if it is used in a non-stationary environment. The table does not include the metrics related to built-in security, as all these schemes do not require users to perform any optional actions.

### 4.6.1 Paradigm Level Analysis

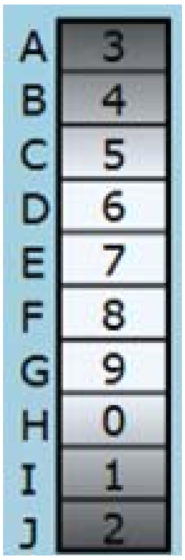
The schemes [42, 44, 23] in the hidden response (HR) paradigm require a special device feature such as gaze tracker, or a special gesture such as hand shielding to hide authentication responses. There are several limitations for the practicability of these schemes, including: 1) the special device feature required may not be universally available; 2) it may require extra human capability to operate the device; 3) the special device feature may not be operable in certain environments; 4) the gesture may not be able to protect all the leakage channels supposed to be protected. For example, the hand shielding gesture used in the ShieldPIN scheme [42] merely blocks the adversary's vision channel. However, it is still possible for the adversary to exploit the haptic channel that is not protected by the gesture. Although an external key logger for a touchscreen has not been observed in the wild, it is technically feasible to implement it like other hardware key loggers [64]. Considering that the thickness of the touchscreen in Samsung Galaxy S3 is just 1.1mm [3], a user may not be able to notice such difference if an extra hardware "touch" logger is installed above a normal touchscreen. So an extra partial secure haptic channel needs to be added to protect *click* operations for the ShieldPIN scheme [42], though this channel is not addressed in the original design. For a similar reason, an extra secure vision channel is added for the PressureGrid scheme [42] in Table 4.1.

The schemes [13, 12] in the hidden confirmation (HC) paradigm rely on the leakage resistance property of acoustic or haptic channel. It is relatively easy to protect such a channel. However, an inherent limitation on the usability of these schemes is that the cognitive workload increases linearly with the size of password alphabet, as it requires users to enumerate a list of randomly-ordered questions and confirm whether a password element appears in an enumerated question.

Compared to the other two paradigms, the hidden transformation (HT) paradigm has an advantage in that the partial secure channel delivers only a transformation (referred to as *hint*) for each challenge. The hint alone does not leak any information about the password, which maps a fixed round secret to a random response. If the hint is not revealed together with the corresponding response, it is impossible for an adversary to derive any valuable information about the password. The usability cost of this paradigm is more scalable compared to the HC paradigm, as the cognitive workload of each challenge is asymptotically constant for arbitrary-sized password alphabet.

### 4.6.2 Scheme Level Analysis

In Table 4.1, we use *grey* color to indicate *noticeable* costs discovered in the existing schemes. These costs could be reduced for a scheme utilizing the same design elements (e.g. interaction channels, device features, etc). For example, the SUR value is $0.5$ for the CuePin scheme [42] as illustrated in Figure 4.9, which implies this scheme requires doubled screen space compared to another scheme whose SUR value is $1.0$ if they are using the same size for individual visual elements. This change does not necessarily imply that a scheme with a larger SUR has a better usability, but it does imply that it will be easier to adapt such a scheme into small-screen devices like smartphones which are perceived to be the most pervasive computing devices in the near future.



10 X 2 = 20 displayed letters and digits
0.1 probability of guessing the correct response from 10 possible answers

**SUR** = (1/0.1)/20 = 1/2 = 0.5

Figure 4.9: SUR calculation for the CuePin scheme [42]

The results show that *all* of these schemes have grey fields, which indicates they can be improved in different aspects. Their major limitations can be summarized as 1) requiring an uncommon device feature, or 2) inoperable in certain common scenarios. We also notice that a few LRPE schemes still have password leakage though they are equipped with partial secure channels. For example, two schemes in the HT paradigm do not satisfy the no-leakage conditions given in Section 4.3.2, due to non-uniform distribution of challenges [61] or responses [25].

Table 4.1 does not highlight all the differences among these schemes, as some of them may not be directly comparable. These differences include those given in metrics named "*cognitive operation list*", "*interaction channels*", "*partial secure channels*", and "*required human capabilities*". Since different users may have different preferences and skills to perform these operations, and the effectiveness of these interaction channels also depends on the application scenarios, it is hard to claim that one scheme is absolutely better than another from the aspects characterized by those metrics. In particular, for the *cognitive operation list* (COL) metrics, although the schemes in the HC paradigm have a longer COL compared to the schemes in the HT paradigm, the actual authentication time of using the HC paradigm may be shorter. It is because recognition used in the HC paradigm could be faster compared to mental arithmetic used in the HT paradigm especially for a password with a small alphabet such as digital PINs. So it could be more appropriate to use these metrics to classify the schemes according to their paradigms and other design elements, and then compare the schemes within the same category.

Note that the purpose of the above analysis is not to identify the *best* scheme. Since these schemes are designed against different attacks, it may not be fair to directly compare them, though all these attacks are within the scope of the LRPE problem. We emphasize that our design metrics are best used to dissect the design of existing LRPE schemes so that we can better understand the underlying design decisions and identify potential limitations.

## 4.7 Challenges behind the LRPE Problem

The reliability of a partial secure channel is the key issue that affects the practicability of an LRPE scheme. It may not be easy to address all the common attacks in the LRPE problem. Figure 4.10 summarizes major potential attacks causing password leakage during password entry.

This figure divides intermediate components into three layers which deliver *mes-*

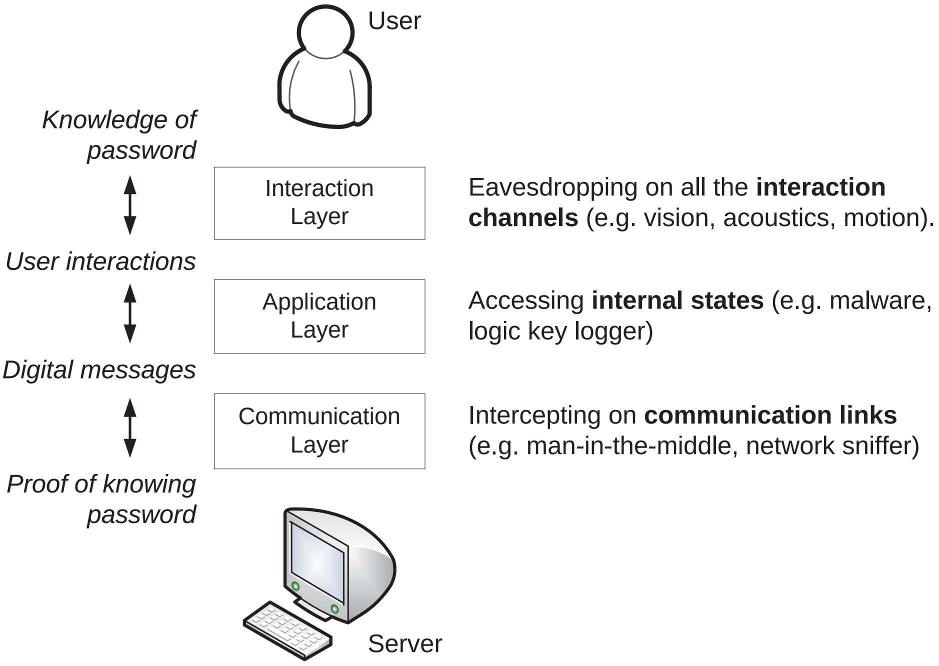

User
Knowledge of password
Interaction Layer
Eavesdropping on all the **interaction channels** (e.g. vision, acoustics, motion).
User interactions
Application Layer
Accessing **internal states** (e.g. malware, logic key logger)
Digital messages
Communication Layer
Intercepting on **communication links** (e.g. man-in-the-middle, network sniffer)
Proof of knowing password
Server

Figure 4.10: A layered view of potential attacks against an LRPE scheme

*sages* between user and server. The top layer is the *interaction layer* between user and user interface on a computing device. The messages delivered in this layer are subject to eavesdropping on all interaction channels, including vision, acoustics, and motion. For example, physical key logger is a typical attack in the interaction layer, which targets at the motion channel (i.e. recording the key sequence pressed by the user). The middle layer is the *application layer* that translates user interaction into digital messages and delivers these messages between user interface and communication layer. The major attacks in this layer are malware and logic key logger that may intercept plaintext passwords in computer memory. The *communication layer* is the bottom layer that delivers messages between application layer and (remote or local) server via a network connection or a local data bus. The messages delivered in this layer may be captured by man-in-the-middle attacks. All these attacks need to be properly addressed in the design of a practical LRPE scheme; otherwise the scheme will be vulnerable.

Most existing LRPE schemes [44, 61, 23, 25, 42, 13, 12] focus on designing the protection mechanisms on the interaction layer, while the attacks on the other two layers, such as malware and logic key logger, are usually not directly addressed in their scheme designs. However, it is actually feasible to effectively protect against these attacks with state-of-the-art technologies nowadays.

We discuss these technologies starting from the communication layer at the bottom. If an LRPE scheme involves a remote server, the attacks of *network eavesdropping* can be effectively prevented with Transport Layer Security (TLS) as long as it is properly implemented [28]. If an LRPE scheme is designed for unmanaged devices like public computer kiosks, trusted computing technologies [67] can be used to protect messages stored in computer memory and deliver them safely via a local data bus. Trusted computing can also be used to establish a trusted execution environment [53, 4] in the upper application layer. Such a trusted execution environment forms a sandbox, which prevents other applications including *malware* from accessing the messages stored in the memory space of the protected application that provides the user interface for password entry.

A recent technique called *remote view controller* implemented on iOS 6 [11] further separates the sensitive interface that receives user's password input from the application that asks the user to prove his identity. For example, when a user launches an email composer from a third party app on an iPhone 5, the user interface of the email composer is actually provided by another system service. This interface (referred to as *remote view*) cannot be controlled by the third party app once it launched, as it runs as a separate process. This technique enables privilege separation even within an application logic, which protects password related messages from a *logic key logger* implanted within the application as long as the integrity of the sensitive interface is not compromised.

All the above technologies provide feasible solutions to ensure messages can be safely delivered between user and server. Since their applicability is not dependent on user interaction, they can be easily adapted in any LRPE design and provide a foundation to establish the required partial secure channel.

We note that it will take time for the above technologies to become pervasive in computing devices. Lack of such **trusted execution environment** is a *major* obstacle for adopting LRPE schemes on the global scale. However, it is not necessary to have all these technologies in place to adopt LRPE schemes in certain settings. For example, if an LRPE scheme is designed for well managed devices such as ATM machines, protection of the interaction layer could be sufficient if application software, operating system, and network have already been secured with dedicated solutions.

Last but not least, we remark that it is always possible for an adversary to exploit subtle side channels such as that used in the brainwave analysis via a brain-computer interface [50]. It may not be feasible or practical to completely prevent these attacks exploiting inevitable human behavior patterns during password entry.

## 4.8 Implications and Limitations

In this section, we discuss the implication of solving the LRPE problem, other related metrics, and the limitations of the design metrics proposed in this study.

### 4.8.1 Implication of a Practical LRPE Scheme

The success of a practical LRPE scheme requires a leakage-resilient environment across all three layers between user and server and against various attacks including malware, key logger, and hidden camera. Thus the conditions required by this environment should be satisfied in reasonable real-world settings, instead of remaining as assumptions. As we analyzed in the last section, it is feasible to establish such an environment with state-of-the-art technologies [67, 53, 4, 11]. With the widespread deployment of these technologies, we expect to see increased use of LRPE schemes.

A good LRPE design may not even require a user to carry extra physical devices as seen in the CuePin scheme [42]. All the operations can be completed by using only the primary device that the user has to carry anyway to access services after user authentication. An LRPE scheme designed in this way will become much more scalable as it can apply to many application scenarios where it is not affordable or convenient to use hardware-based OTPs. For example, a user no longer needs to worry about password leakage caused by the ATM skimming attack [43] when using a physical ATM. If an LRPE scheme uses only commodity computing devices and incurs reasonably low usability costs, it may even have the potential to replace legacy passwords. On the other hand, hardware-based OTPs will still serve as a second factor in user authentication for high value services; they complement the protection by further mitigating other threats such as social engineering. By combining LRPE schemes and hardware-based OTPs, the cost of attacks to user authentication could be significantly increased.

### 4.8.2 Other Metrics

There are other metrics which are not directly related to password leakage during password entry, but they are still important for practical purposes. Secure password storage is one example. All existing usable LRPE schemes without using any form of secure channel [35, 48, 71, 72, 10] store users' passwords in cleartext; otherwise, the challenges in authentication cannot be generated as specified in their design. There is no such restriction if a partial secure channel is available.

On the other hand, although our metrics cover all the major design aspects of LRPE schemes that can be deterministically characterized in the design phase, the designer may still need to investigate the following nondeterministic aspects. 1) *Choice between text and graphic symbols:* although some psychology evidence [22] shows that it tends to be easier for users to remember graphic symbols, this advantage can be significantly undermined due to the chunking effect [31] when users can find a logical relation that helps memorizing a text password. 2) *Recall and recognition:* they cannot be deterministically characterized due to similar reasons caused by the chunking effect. 3) *Password interference:* it is caused by the inherent limitation of human memory capability explained by the interference theory [68], where memorizing a password may affect the user's memory of another password. 4) *Types of cognitive operations*: since most operations related to a partial secure channel are designed to operate an uncommon device, it is difficult to find respective performance models to precisely characterize the cognitive workload for each type of cognitive operation [75]. 5) *User perception-related metrics:* metrics likes user frustration level, concentration level, and motivational effort are susceptible to many implementation and environmental factors. Besides the above metrics, traditional evaluation metrics also include login error rate, time to login, learning curve, etc. It is necessary to examine all these metrics to ensure the completeness of scheme evaluation.

### 4.8.3 Limitations

The design metrics we propose in this work are still in the preliminary stage. The relations between some design metrics and the actual usability costs remain hypotheses, though the rationales behind these metrics are straightforward. Future work may be required to systematically verify their validity by conducting user study on real systems. However, the usability is a *vague* notion, which includes many interrelated aspects and different users may have completely different perceptions on how usable a system is. So it is always possible to miss certain factors that may significantly influence the actual usability. Furthermore, the results of usability evaluation may even heavily depend on the selection of subjects. People in different cultures have different advantage and disadvantage in using such systems [17]. Even for the same system and the same subject, the results could be significantly different depending on whether the subject is properly motivated and learns to use the system by self-learning or well-designed training. Many other factors also have to be controlled in order to produce a reasonable comparison among existing systems. Proper assessment of usability is indeed challenging, which probably explains why there does not exist any prior work on *large scale* quantitative analysis of user authentication systems.

As indicated by Bonneau et al. [15], it is important to provide consistent *qualitative* metrics when it is not feasible to derive consistent *quantitative* metrics. Our design metrics follow this philosophy but with a twist. We attempt to go deeper to dissect the system into individual design elements. This provides a fine-grained context for designers to better understand underlying design elements and their relations, which may further facilitate a fair comparison among systems with a similar internal structure. Although we are not able to provide more quantitative values for all the design metrics as the relations between these metrics and their impacts on usability may not be linear, the list of related design elements required by these metrics can still help the designers to examine the necessity of each design element and identify the limitations in the early stage.

Our metrics also complement traditional evaluation metrics by expanding the scope of security and usability evaluation beyond the traditional laboratory user study. This may remind designers of the important missing factors, such as the impacts of *time pressure*, *distraction*, and *mental workload*. Although they are not classified as design metrics as they need to be evaluated in a user study, they are common situations for an LRPE schemes used on daily basis. Thus it could be better for the future study to include these metrics so that the usability evaluation would be more precise. An example result could be "*usable even under pressure level 3.0*" instead of simply "*usable*", if we are able to answer the question how to define the standardized pressure condition. We believe this is the right direction to produce more credible results for the evaluation of security systems involving human interaction. We hope our efforts on the design metrics for the LRPE schemes can provide an insight for the future development of design metrics for other security systems.

## 4.9 Discussion

In this work, we made the first attempt to provide a comprehensive understanding for the leakage-resilient password entry (LRPE) problem, which addresses the challenges of designing practical LRPE schemes. We proposed a broad set of metrics to evaluate LRPE schemes from different perspectives, including security-usability relations, built-in security, and universal accessibility. These metrics were designed to identify the potential limitations before conducting user studies. They were applied to existing LRPE schemes, which reveals that their major limitations include 1) requiring an uncommon device feature, 2) inoperable in certain common scenarios, and 3) lack of trusted execution environment. Our analysis further showed that it is possible to overcome these limitations by improving the design according to the proposed metrics. We expect these design metrics be used to guide the design of LRPE schemes in future research.

# Chapter 5

# Designing Leakage-Resilient Password Entry on Touchscreen Mobile Devices

## 5.1 Introduction

Guided by the metrics developed in our second work, this chapter proposes a secure and usable LRPE scheme leveraging on the touchscreen feature of mobile devices. Mobile devices are becoming essential tools in modern life, which seamlessly connect human beings to the cyberspace. A user can now use a smartphone or tablet to access not only general informative services but also sensitive services such as mobile banking and corporate services. In order to prevent unauthorized access to these services, user authentication is required to verify the identity of a user. Among existing user authentication mechanisms, passwords are still the most pervasive due to their significant advantage in usability over other alternatives such as smartcards and biometrics [49]. However, password-based user authentication has intrinsic weakness in password leakage, which may lead to financial loss or corporate data disclosure. This threat could be more serious in scenarios when mobile devices are involved, as mobile devices are widely used in public places.

Password leakage during password entry is a classic problem in password-based authentication. Most prior research [35, 48, 71, 72, 10, 44, 61, 23, 42] on this problem focuses on desktop computers, where specific restrictions on mobile devices are usually not addressed. These restrictions mainly include: 1) a mobile device usually has a smaller screen size than a desktop computer; 2) a mobile device needs to be operable in non-stationary environments such as on public transit. On the other hand, mobile devices provide additional features such as touchscreen, which may not be available in traditional settings. These new features can be utilized to support advanced security properties that were difficult to achieve before.

In this work, we propose a *concise yet effective* authentication scheme named CoverPad, which is designed for password entry on touchscreen mobile devices. CoverPad improves *leakage resilience* of password entry while *retaining most benefits* of legacy passwords. Leakage resilience is achieved by utilizing the gesture detection feature of touchscreen in forming a *cover* for user inputs. This cover is used to safely deliver hidden messages, which break the correlation between the underlying password and the interaction information observable to an adversary. From the other perspective, our scheme is also designed to retain the benefits provided by legacy passwords. This requirement is critical, as Bonneau et al. [15] conclude that any user authentication is unlikely to gain traction if it does not retain comparable benefits of legacy passwords. Our scheme approaches this requirement by involving only intuitive cognitive operations and requiring no extra devices in the design.

We implement three variants of CoverPad and evaluate them with an extended user study. This study includes additional test conditions related to *time pressure*, *distraction*, and *mental workload*. These test conditions simulate common situations for a daily-used password entry scheme, which have not been evaluated in the prior literature. We design new experiments to examine their influence based on previous work in psychology literature [40, 22, 38]. Experimental results show the influence of these conditions on user performance and the practicability of our proposed scheme.

The contributions of this work are summarized as follows.

- We propose CoverPad to protect password entry on touchscreen mobile devices. It achieves leakage resilience and retains most benefits of legacy passwords by involving only intuitive cognitive operations and requiring no extra devices.
- We implement three variants of CoverPad to address different user preferences. Our user study shows the practicability of these variants.
- We extend user study methodology to examine the influence of various additional test conditions. Among these conditions, time pressure and mental workload are shown to have significant impacts on user performance. Therefore, it is recommended to include these conditions in the evaluation of user authentication schemes in the future.

## 5.2 Threat Model

Passwords are the most pervasive user authentication that allows a human *user* to be authenticated to a (local or remote) computer *server*. *Password leakage* is a threat that a user's password is directly disclosed or indirectly inferred. It usually happens during *password entry*, when a user inputs his password in order to prove his identity. In the case of legacy passwords, a user directly enters his plaintext password so that the password may be captured via various eavesdropping attacks including key logger, hidden camera, and malware. We classify these attacks into two types, *external* or *internal*, according to whether an adversary can access the internal states of a device for password entry, such as device memory.

An external eavesdropping attack is an attack exploiting a leakage channel outside a device. This type of attacks includes *vision*-based eavesdropping such as hidden camera, *haptics*-based eavesdropping such as physical key logger, and *acoustics*-based eavesdropping such as tone analysis. Compared to traditional scenarios involving only desktop computers, an adversary has more opportunities to launch an external eavesdropping attack against mobile devices, as mobile devices are widely used in public places. In a crowded area, an adversary may observe password entry in a close distance without being noticed (see Figure 5.1).

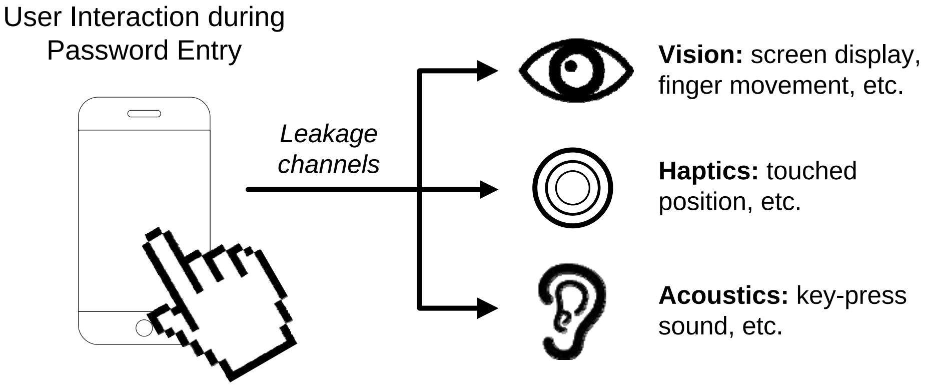

User Interaction during Password Entry
Leakage channels
Vision: screen display, finger movement, etc.
Haptics: touched position, etc.
Acoustics: key-press sound, etc.

Figure 5.1: Attack scenarios

For vision-based attacks, an adversary may infer the actual password by observing the movement of fingers even without direct line-of-sight on the screen display. This capability is significantly enhanced with emerging augmented-reality accessory like Google Glass [33], which is a small wearable glass transferring real-time video captured by a tiny camera to a server and displaying the analyzed results received from the server.

Haptics-based attacks are most likely to happen when users use public mobile devices. Mobile devices, such as iPad, have been used as public computer kiosks as observed in museums, restaurants, and hotels [39, 37, 77]. In addition, many existing kiosks are also equipped with touchscreen similar to mobile devices. This provides an incentive for an adversary to install a physical "touch" logger. Although such touch logger has not been observed in the wild, it is technically feasible to implement as other physical key loggers [64]. Considering that the thickness of touchscreen in Samsung Galaxy S3 is just 1.1mm [3], it may not be noticeable to users if an extra physical touch logger is installed on a normal touchscreen.

The effectiveness of acoustics-based attacks depends on whether user actions can be distinguished by their tone patterns. For example, different tones are played when a user dials different numbers on an old-style phone. Due to environmental noises, acoustics-based attacks are usually not as effective as vision-based attacks and haptics-based attacks.

The other type of attacks that cause password leakage is the internal eavesdropping attack. Such attacks exploit a leakage channel inside a device, where an adversary is allowed to access the internal states such as reading device memory. This type of attacks include *logic key logger*, *malware*, and *network eavesdropping*, which are common to all password-based user authentication schemes. Like most prior research [44, 61, 23, 25, 42, 13, 12], our scheme design does not address these attacks for the following reasons: 1) Existing solutions [53, 4, 11, 67] such as application sandbox are available to effectively defend against these attacks, though it takes time for them to replace legacy vulnerable systems; 2) these solutions are independent on user interaction during password entry so that they can be adapted to any user authentication schemes. Compared to external eavesdropping attacks, the threat from internal eavesdropping attacks can be effectively mitigated if a user uses a computer system that is properly updated and configured [28], while it is not easy to defend against external eavesdropping attacks as they are caused by *inevitable* exposure of human interaction during password entry. These external eavesdropping attacks impose realistic threats leading to password leakage. We will thus focus on external eavesdropping attacks in our scheme design.

Besides the above attacks which happen during password entry, password leakage may also be caused by other types of attacks including social engineering and phishing [49]. Although their mitigation technologies such as secure URL checker and spam filter have been widely deployed in modern computer systems, some of these attacks may not be completely preventable by technical solutions alone. Another example is the database reading attack, where the back-end databases are intruded so that all user passwords are compromised. Since these attacks are orthogonal to the password entry problem, they are out of the scope of this work.

## 5.3 CoverPad Design

In this section, we present the design of CoverPad. First, we describe our design objectives from both security and usability perspectives. Then, we introduce the conceptual design of CoverPad. Lastly, we present three variants in implementing CoverPad.

### 5.3.1 Design Objectives

CoverPad is designed to improve leakage resilience of password entry while retaining most benefits of legacy passwords. We describe our design objectives as follows.

First, in terms of security, a scheme should minimize password leakage during password entry under realistic settings. To achieve this objective, a user should 1) input obfuscated response derived from his password, and/or 2) input his password in a secure channel. A recent study [75] shows strong evidence on the infeasibility of using obfuscated response solely based on human cognitive capabilities. Therefore, it is necessary to rely on certain secure channel to achieve this security objective. However, a standard secure channel may be difficult to establish in practice, which requires to protect all messages delivered between user and server. Therefore, we choose a hybrid solution in our scheme design. With the assistance of simple obfuscation, the requirement on a secure channel can be significantly reduced, as only a few critical messages need to be protected. Such channel is referred to as *partial* secure channel.

In the presence of a partial secure channel, it is possible to achieve the optimal security objective – *no password leakage*. As long as the partial secure channel is not compromised, CoverPad provides the same leakage resilience as *one-time pad* [54], where the most efficient attacks for an adversary to learn the password are online dictionary attacks. We will show how this security objective is achieved in our scheme in the following sections.

Second, in terms of usability, a scheme should preserve the benefits of legacy passwords in order to gain traction [15]. The major benefits of legacy passwords include no extra devices required, and only intuitive cognitive operations performed. We further consider additional restrictions on mobile devices including that 1) a mobile device usually has a *smaller* screen size compared to a desktop computer; 2) a mobile device needs to be operable in a *non-stationary* environment such as on public transit. So we minimize the number of visual elements that are displayed simultaneously on the screen, and also simplify the involved operations to make them suitable in a non-stationary environment.

### 5.3.2 Conceptual Design

The conceptual design of CoverPad is shown in Figure 5.2, where a hidden transformation $T_i(\cdot)$ is a random mapping $\Omega \to \Omega$, where $\Omega$ is the set of all individual elements contained in the password alphabet.

**Setup:**
*A server and a user agree on a $k$-length password $pwd = (a_1, a_2, \ldots, a_k)$, where a* ***password element*** *$a_i = pwd[i]$ belongs to an alphabet with size $w$. It is allowed that $a_i = a_j$, for $i \neq j$.*

**Password Entry:**
*For each $i$ from $[1, k]$:*

Step 1: *The touchscreen shows a keypad with all the elements in the alphabet.*

Step 2: *The user is asked to perform a hand-shielding gesture to read the hidden transformation $T_i(\cdot)$ protected by the hand-shielding gesture. $T_i(\cdot)$ will immediately disappear if the gesture is no longer detected.*

Step 3: *The user clicks on response element $e_i$, where $e_i = T_i(a_i) = (a_i + r_i \mod w)$, where $r_i$ is a random number drawn from a uniform distribution. A new random number $r_i$ is generated for each round $i$. The hand-shielding gesture is not required for this step.*

Figure 5.2: Conceptual design of CoverPad

An example of using CoverPad is given as follows. Suppose a user has a $k$-length password. At the beginning of password entry, the user performs the hand-shielding gesture to view the current hidden transformation $T_1$ for the first character

$a_1$ in his password. Then, he applies $T_1$ to $a_1$ and enters the transformed response $e_1$. This procedure repeats for each password element $a_i$. During the whole password entry, $T_i$ disappears immediately once the gesture is not being detected. A user can always view $T_i$ by performing the gesture again before inputting $e_i$.

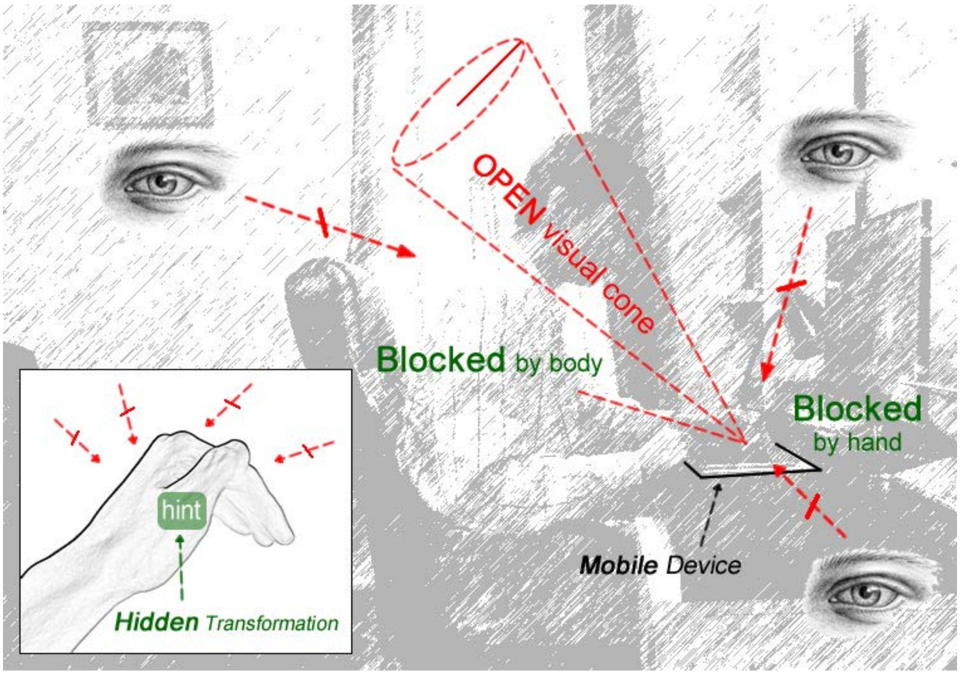


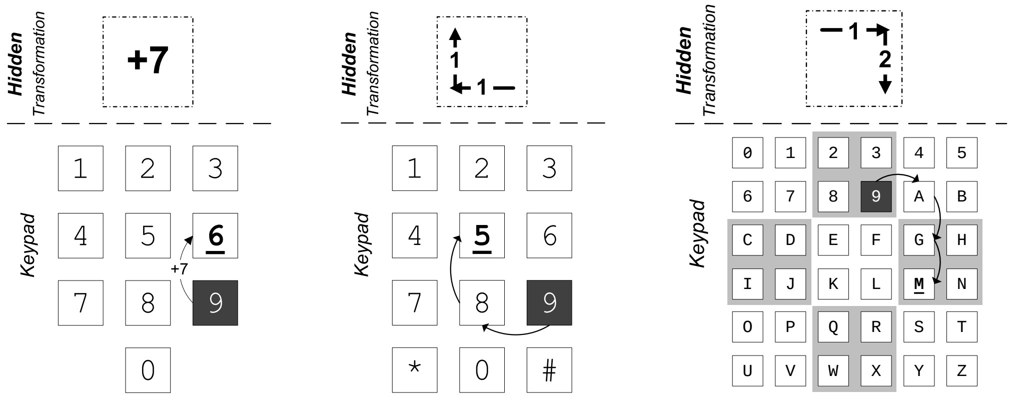
Figure 5.3: The hand-shielding gesture and its effectiveness

Figure 5.3 shows how to correctly perform a hand-shielding gesture. This gesture restricts the vision channel to a small visual cone. This visual cone is not accessible to an adversary unless the adversary's eyes are close enough to the user's head, which makes the adversary easily exposed. A hidden camera near the line of sight may help capture the hidden transformation. However, it needs to be adjusted according to the user's height and current position, which may lead to user's awareness. On the other hand, the observable responses for the same password element are uniformly randomized. Thus, CoverPad is also immune to haptics-based eavesdropping. Further analysis is provided in the next section.

Therefore, it is difficult to compromise the partial secure channel formed by the hand-shielding gesture from external eavesdropping attacks in practice, though the use of this gesture is simple. If the protective gesture is not being detected by the touchscreen, the hidden transformation will not be displayed such that the hidden transformation is always protected under the required gesture. Note that a hidden transformation alone does not leak any information about the password. As long as the hidden transformation is not revealed together with the corresponding response, observed interaction provides no valuable information for an adversary to infer the actual password. A proof about this security property will be given in Section 5.4.

### 5.3.3 Implementation Variants

We provide three variants of CoverPad that implement different features tailored for users with various skill sets, which are described and illustrated as follows (see Figure 5.4).



(a) NumPad-Add (b) NumPad-Shift (c) LetterPad-Shift

Figure 5.4: Demonstration of three implementation variants

#### NumPad-Add

In NumPad-Add, the alphabet of password consists of digits 0 to 9 only. The hidden transformation is performed by *adding* a random digit to the current password element and then $\mod 10$ if the sum is larger than 9, where the value of the random digit ranges from 0 to 9. For example, the correct response for the first round is $6 = (9 + 7) \mod 10$ given password $\underline{9}34567$ and the hidden message '*plus 7*'.

### NumPad-Shift

In NumPad-Shift, the alphabet of password consists of digits 0 to 9 only. The hidden transformation is performed by *shifting* the location of the current password element by $X$-offset and $Y$-offset, where the offset values are randomly taken from $\{-1, 0, 1\}$ for $X$-offset, and $\{-1, 0, 1, 2\}$ for $Y$-offset. For a $3 \times 4$ keypad design shown in Figure 5.4(b), the transformed response for $a_i$ is calculated as $pad[x(a_i) + \Delta x \mod 3][y(a_i) + \Delta y \mod 4]$, where $\Delta x$ is the $X$-offset, $\Delta y$ is the $Y$-offset, and $x(a_i)$ is the $X$-index of $a_i$, and $y(a_i)$ is the $Y$-index of $a_i$. For example, the correct response for the first round is 5 if the password is $\underline{9}34567$ and the hidden message is '*move left by 1 step and move up by 1 step*'.

Note that two extra keys $*$ and $\#$ are added to the keypad; otherwise, the distribution of hidden transformations is not uniform on the keypad layout. The proof for the necessity of these two keys is given as follows. Assuming $*$ and $\#$ keys are removed, the keypad now contains only 10 keys for digits 0 to 9. To provide a full transformation from a secret key to a random key, the minimum value set is $\{-1, 0, 1\}$ for $X$-offsets and $\{-1, 0, 1, 2\}$ for $Y$-offsets. There are twelve combinations between $X$-offsets and $Y$-offsets, but only ten keys on the keypad. If the offset values are drawn from a uniform distribution, certain response keys for a given password element would have a higher frequency compared to others (it is similar as placing twelve balls in ten buckets in a deterministic way). The exact distribution of response keys is decided by the underlying password element, thus it discloses valuable information about the password. From the other perspective, if response keys are drawn from a uniform distribution, the offset values will not be uniformly distributed due to similar reason. Therefore, it is necessary to add these two extra keys to the NumPad-Shift keypad.

**LetterPad-Shift**

In LetterPad-Shift, the alphabet of password consists of letters $a$ to $z$ and digits 0 to 9 (36 elements in total). The hidden transformation is the same as NumPad-Shift. The offset values are randomly taken from $\{-2, -1, 0, 1, 2, 3\}$ for both $X$-offset and $Y$-offset for a $6 \times 6$ keypad design. The transformed response for $a_i$ is calculated as $pad[x(a_i) + \Delta x \mod 6][y(a_i) + \Delta y \mod 6]$ in a similar way as for NumPad-Shift. A background grid is added to ease the calculation of shifting, as shown in Figure 5.4(c).

## 5.4 Security Analysis

### 5.4.1 External Eavesdropping Attacks

Common external eavesdropping attacks leading to password leakage may exploit vision, haptics, or acoustics channel as analyzed in Section 5.2. For CoverPad, an adversary using these attacks can observe *at most* a complete response key sequence pressed by a user, while the hidden transformation is protected by our design. From this key sequence, the adversary knows the $i$-th pressed key is decided by the $i$-th element in the password. However, the adversary cannot further infer what the $i$-th password element is, as proved as follows.

**Proof:** Given a pressed key $e_i$, and two password elements $a_x$ and $a_y$ in a $w$-sized password alphabet, let $Pr(e_i|a_x)$ and $Pr(e_i|a_y)$ be the probabilities for $e_i$ being pressed when the underlying password element are $a_x$ and $a_y$, respectively. We have $Pr(e_i|a_x) = Pr(e_i = a_x + r_i \mod w) = Pr(r_i = e_i - a_x \mod w) = Pr(r_i = C \mod w) = 1/w = Pr(e_i|a_y)$ for any $i$, $x$, and $y$, where $C$ is a constant integer randomly drawn from a uniform distribution. Therefore, a sequence of pressed keys observed by an adversary is equivalent to a random sequence, which is similar to a ciphertext generated by a one-time pad. □

In a partial secure channel where the hidden transformation is protected by the hand-shielding gesture, our scheme achieves no password leakage. As long as the hidden transformation is not disclosed together with the corresponding response, an adversary cannot infer any information about the underlying password (except password length) even after an infinite number of observations.

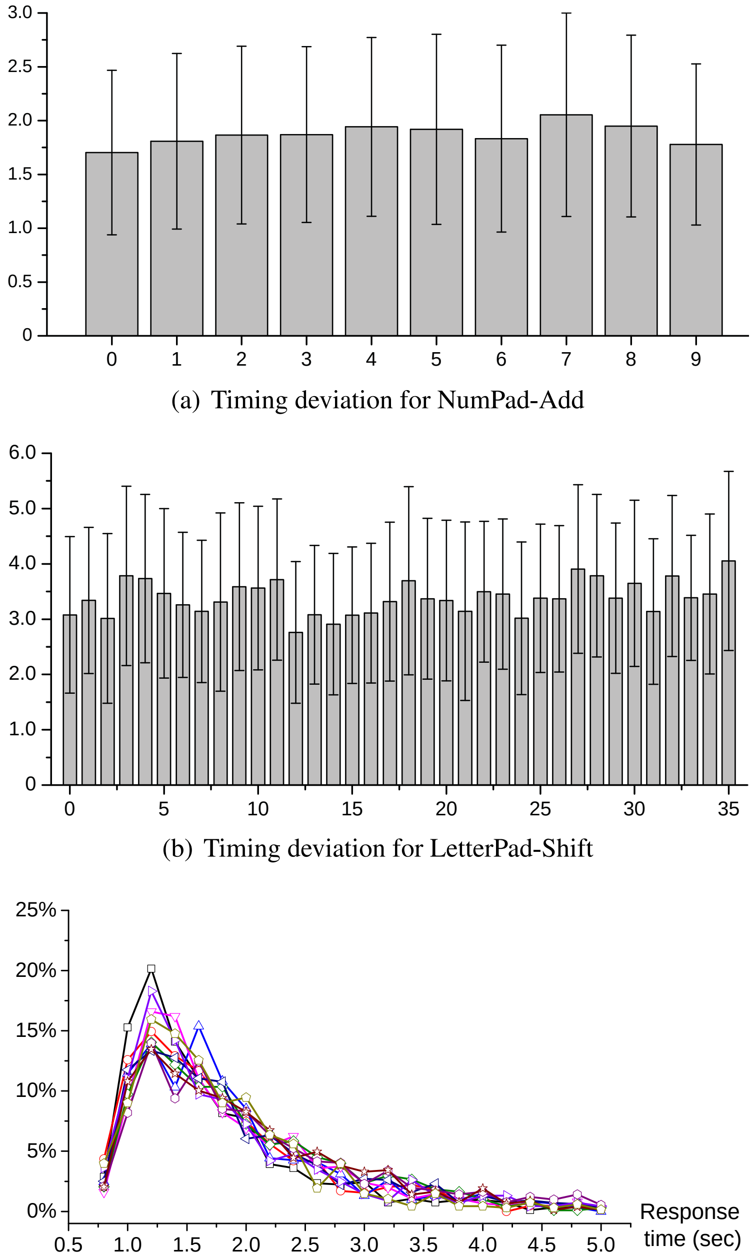
### 5.4.2 Side-channel Attacks

In reality, it is possible for an adversary to exploit subtle side-channels to collect password information during password entry. These attacks are not usually considered in common threat models [35, 48, 71, 72, 10, 44, 61, 23, 25, 42, 13, 12]. A typical side-channel attack is timing analysis [63], which analyzes the patterns in the response time of entering individual password elements. The preliminary results of our scheme against timing analysis are given in Figure 5.5. For the timing deviation shown in Figure 5.5(a) and 5.5(b), each bar with $x$-value $i$ represents the average response time for entering the transformed responses for a specific password element $i$. For the timing distribution shown in Figure 5.5(c), each line in the figure represents the distribution of the response time for entering the transformed responses for a specific password element. These results show the range and the distribution of the response time for entering different password elements are almost overlapped. This indicates that timing analysis is not a major concern for our scheme, though it is difficult to completely prevent such attacks due to inevitable human behavior patterns during password entry. Detailed analysis on side channel attacks is out of the scope of this work.

## 5.5 Usability Evaluation

### 5.5.1 Methodology

The participants in our user study are recruited from undergraduate students in our university. There are 61 participants in total, 30 male and 31 female, with age range



(a) Timing deviation for NumPad-Add

(b) Timing deviation for LetterPad-Shift

(c) Timing Distribution for NumPad-Add

Figure 5.5: Timing deviations and distributions for entering each password element. The results of NumPad-Shift are similar to the results of NumPad-Add shown in these figures.

between 20 and 25. These participants come from five different departments, in which 42 of them have a social science or business related background, and the remaining have a computer science or information technology related background. Each participant is paid with 10 dollars as compensation for their time. We establish a *ranking* system from which a participant can see a *performance score* representing how well the participant performs compared to other participants. This ranking system provides a moderate level of motivation for the participants to do their best in tests. A numerical identifier is assigned to each participant in order to protect user privacy.

The user study is conducted in a quiet room. The experiments use a within-objects design. Each participant is asked to use all three variants as three *test groups*. These variants are implemented on Apple iPad, which are referred to as *schemes* in this section. The order of the schemes is randomized to avoid the learning effect that affects the performance for a specific scheme. For each test group, a user is required to memorize a *randomly generated* password in the beginning. The password strength is set to be equivalent to 6-digit PIN, where the password length is 4 for LetterPad-Shift, and 6 for both NumPad-Shift and NumPad-Add. The same password will be used for the same test group and a "*show my password*" button is provided in case a participant forgot the password. The participants learn how to use a scheme by an interactive step-by-step tutorial. The participants are required to go through the whole tutorial for the first scheme appearing in the tests, and they may skip the tutorial for the second and third schemes after learning the basic scheme design. In the end of each tutorial, there is a short pretest for the participant to exercise. If a participant fails to pass the pretest, the researchers will provide help to ensure that the participant understands how to use the scheme before the tests start.

In each test group, there are six tests simulating additional *test conditions* that evaluate the influence of time pressure, distraction, and mental workload. The details of these test conditions are described in the next subsection. The order of these tests is also randomized in order to avoid the learning effect.

All three test groups consist of 18 tests in total. To avoid the participants from feeling exhausted and bored, each test is designed to be short and can be finished within one or two minutes. The participants are given a short break after each test group. At the end of the user study, the participants are given a questionnaire using 5-point Likert scale to collect their perception on the schemes. The whole user study takes $35 \sim 50$ minutes to complete.

### 5.5.2 Simulating Various Test Conditions

In order to simulate various test conditions related to time pressure, distraction, and mental workload, we introduce two extra experimental tools, timer and secondary task. A *timer* is used to create time pressure by showing a participant how much time is left for the current test condition. It is implemented as a progress bar whose length increases every second with a countdown text field showing how many seconds are left. *Secondary tasks* are used to simulate unexpected distraction and persistent mental workload. We use CRT (*choice reaction time*) tasks as secondary tasks, which is a standard technology in experimental psychology [40, 22, 38]. CRT tasks usually work as secondary tasks that occupy the central executive[1] in human brain when evaluating the performance of a primary task in the presence of a secondary task. CRT tasks require participants to give distinct responses for each possible stimulus. In our implementation, the participants are asked to press the correct button among $N$ buttons, where the correct button should have the same color as the stimulus. For example, if the stimulus shows a red button, a participant should press the red button among $N$ buttons with different colors. We use $N = 2$ for tests in the distraction condition as the major focus is to unexpectedly disrupt password entry with a CRT task. We use $N = 8$ for tests in the mental workload condition so as to create a considerable mental workload, which is the same as in the classic Jensen Box setting [40].

Based on the above experimental tools, we simulate six test conditions for each test group by combining the two modes and three statuses. Two modes related to a timer are described as follows:

- **Relaxed mode:** A participant is asked to minimize the error rate in a fixed number of login attempts where time is not considered in performance score

[1] The central executive is a control system that mediates attention and regulation of processes occurring in working memory [9].

calculation. The number of login attempts is 5 for no-extra-task status and 3 for distraction and mental workload statuses.

- **Timed mode:** A participant is asked to perform as many successful logins as possible within 1 minute, where both time and accuracy are considered in performance score calculation. The countdown of a timer creates time pressure.

Three statuses related to secondary tasks are described as follows:

- **No-extra-task status:** A participant is asked to perform the login task only.
- **Distraction status:** A simple CRT task may appear with $1/3$ probability each time when a participant presses a response key. This task is used to create unexpected distractions during password entry.
- **Mental workload status:** A relatively complex CRT task appears every time when a participant presses a response key. This task is used to create continuing mental workload during password entry.

Among six conditions, we referred to the combination of *relaxed* mode and *no-extra-task* status as the **normal condition**, which is the common condition usually tested in prior work [35, 48, 71, 72, 10, 44, 61, 23, 25, 42, 13, 12]. The short names for the other five conditions are given in Table 5.1.

| **Short name** | **Full specification** |
|---|---|
| normal | *relaxed* mode + *no-extra-task* status |
| timed | *timed* mode + *no-extra-task* status |
| distraction | *relaxed* mode + *distraction* status |
| distraction+timed | *timed* mode + *distraction* status |
| mental workload | *relaxed* mode + *mental workload* status |
| mental workload+timed | *timed* mode + *mental workload* status |

Table 5.1: Short names for test conditions

The hypotheses related to these test conditions are described as follows.

**(H1)** *Compared to the normal condition,* ***login time*** *will be significantly shorter when* ***time pressure*** *is present.*

**(H2)** *Compared to the normal condition,* ***login accuracy*** *will be significantly lower when* ***time pressure*** *is present.*

**(H3)** *Compared to the normal condition,* ***login time*** *will be significantly longer when unexpected* ***distraction*** *is present.*

**(H4)** *Compared to the normal condition,* ***login accuracy*** *will be significantly lower when unexpected* ***distraction*** *is present.*

**(H5)** *Compared to the normal condition,* ***login time*** *will be significantly longer when persistent* ***mental workload*** *is present.*

**(H6)** *Compared to the normal condition,* ***login accuracy*** *will be significantly lower when persistent* ***mental workload*** *is present.*

**(H7)** *Compared to a condition in* ***relaxed*** *mode with* ***secondary tasks, login time*** *will be significantly shorter for its counterpart in* ***timed*** *mode.*

**(H8)** *Compared to a condition in* ***relaxed*** *mode with* ***secondary tasks, login accuracy*** *will be significantly lower for its counterpart in* ***timed*** *mode.*

### 5.5.3 Learning Curve

Although our scheme design involves intuitive operations only, it requires a different process for password entry compared to legacy passwords. While we expect the participants can learn this process with the tutorial and pretests, we observed that some participants were impatient to read all instructions and keep pressing the next button. These participants proceeded to the evaluation stage before they fully understand our scheme design.

Figure 5.6 compares user performance under the normal condition for different positions where a scheme appears in the study. These results show the user performance in terms of login time and login success rates is significantly worse when the tested scheme is the first scheme which a participant encountered in the user study. But the differences on user performance are not significant if a scheme is encoun-

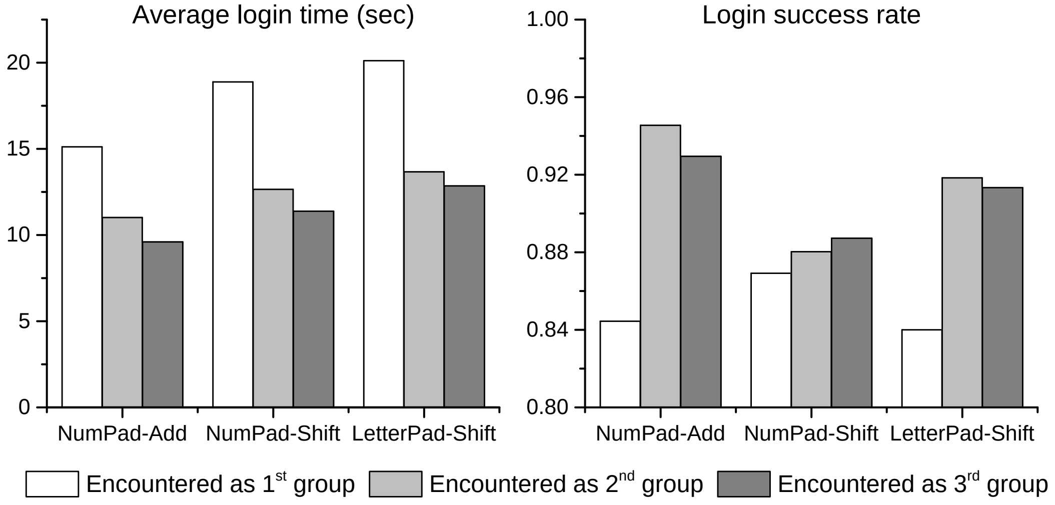


Figure 5.6: Learning curve of CoverPad

tered as the second or third test group, as all our schemes are similar due to the fact that they are based on the same conceptual design. As shown in the learning curve in Figure 5.6, most participants get familiar with our scheme design after the first test group. Therefore, we consider the first test group as part of the learning process, and use the performance data collected from the second and third test groups only in the following analysis.

### 5.5.4 Experimental Results

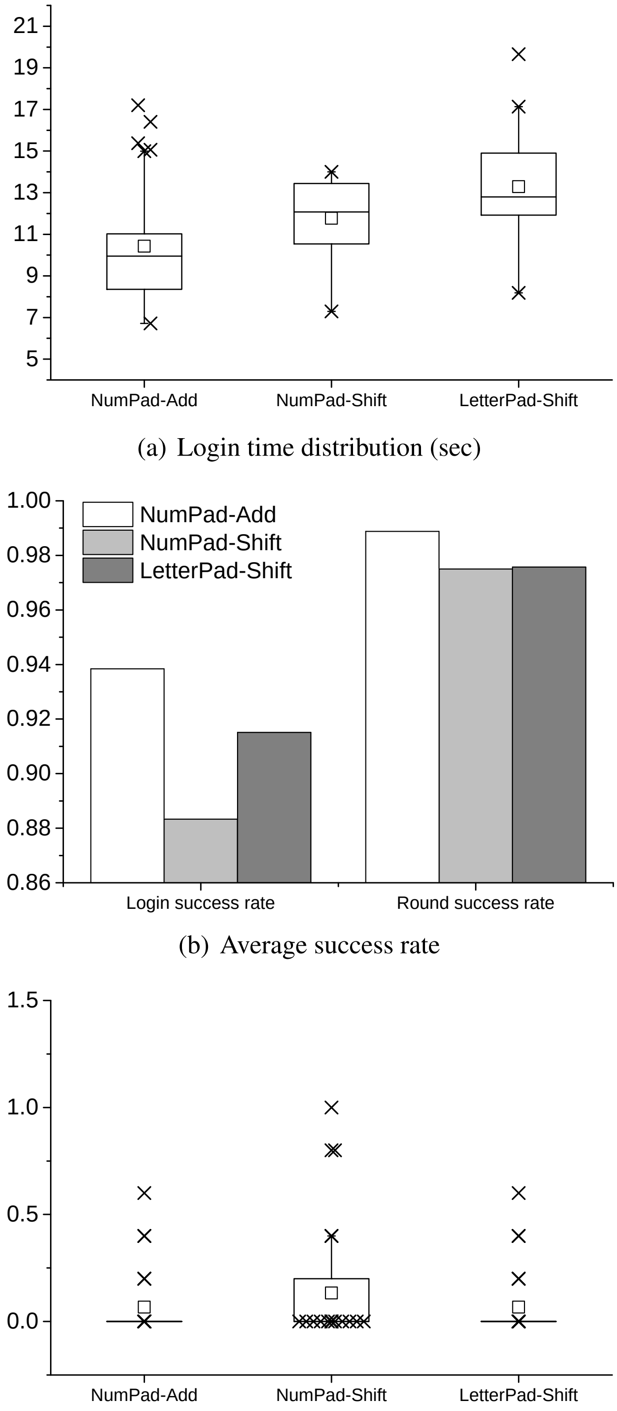
We measure user performance with the following metrics: average login time, login success rates, round success rates, and average edit distances. A *round* success rate is the average success rate for a user to correctly input one password element by applying a hidden transformation. An *edit distance* is the minimum number of insertions, deletions, substitutions, and adjacent transpositions required to transform an input string into the correct password string so that an *average* edit distance is the average value of edit distances calculated from all login attempts of a user under a test condition. Among these metrics, login success rates, round success rates, and average edit distances are used to evaluate *login accuracy*.

We use the following statistical tools to test the significance of our experimental results, where a significance level of $\alpha = .05$ is used. For each comparison, we run an *omnibus* test across all test conditions for each scheme. Since all our performance data are quantitative, we use *Kruskal-Wallis* (KW) test for omnibus tests, which is an analogue of ANOVA but does not require normality. If the omnibus test indicates significance, we further use *Mann-Whitney* (MW) U test to perform pair-wise comparisons so as to identify specific pairs with significant differences. The detailed results of our statistical tests are given in Section 5.5.5.

### Performance under Normal Condition

In the normal condition, a participant is only asked to perform login tasks without any time pressure or secondary tasks. It corresponds to the combination of relaxed mode and no-extra-task status, which is used as a *baseline* in our tests.

Figure 5.7(a) shows the average time for a successful login attempt in the normal condition. For all the three schemes, most participants are able to finish the login within 13 seconds. Figure 5.7(b) and 5.7(c) show the corresponding login accuracy. Since our experiment limits the number of login attempts to 5 in order to prevent the participants from feeling exhausted or bored, even a single mistake would take the login success rate down to $80\%$. Our results indicate that most participants make *at most* one mistake when they use our schemes for the first time after a short training. This is shown by $97.5\%$ average round success rate and $0.13$ average edit distance in the worst case. Particularly, for the distribution of average edit distance of NumPad-Shift, 27 participants among 40 samples (after removing the experimental data when NumPad-Shift appears as the first test group) has an average edit distance of zero (i.e. no mistakes during all tests under the test condition), which are shown as a cluster of *outliers* at the bottom of the box chart. The login accuracy is expected to increase after the participants get more familiar with the schemes.



(a) Login time distribution (sec)

(b) Average success rate

(c) Edit distance distribution

Figure 5.7: Average login time, success rate, and edit distance under the normal condition

### Influence of Time Pressure

Figure 5.8 shows the impact of time pressure without any secondary tasks. The results show that the participants behave much hastily in the presence of time pressure.

The average time for a successful login attempt becomes shorter and the login accuracy is decreased. The statistical tests show the difference in login time is significant ($p$ =.017 for NumPad-Add and $p$ <.001 for LetterPad-Shift) but the difference in login accuracy is not. Therefore, **H1** is supported while **H2** is not.

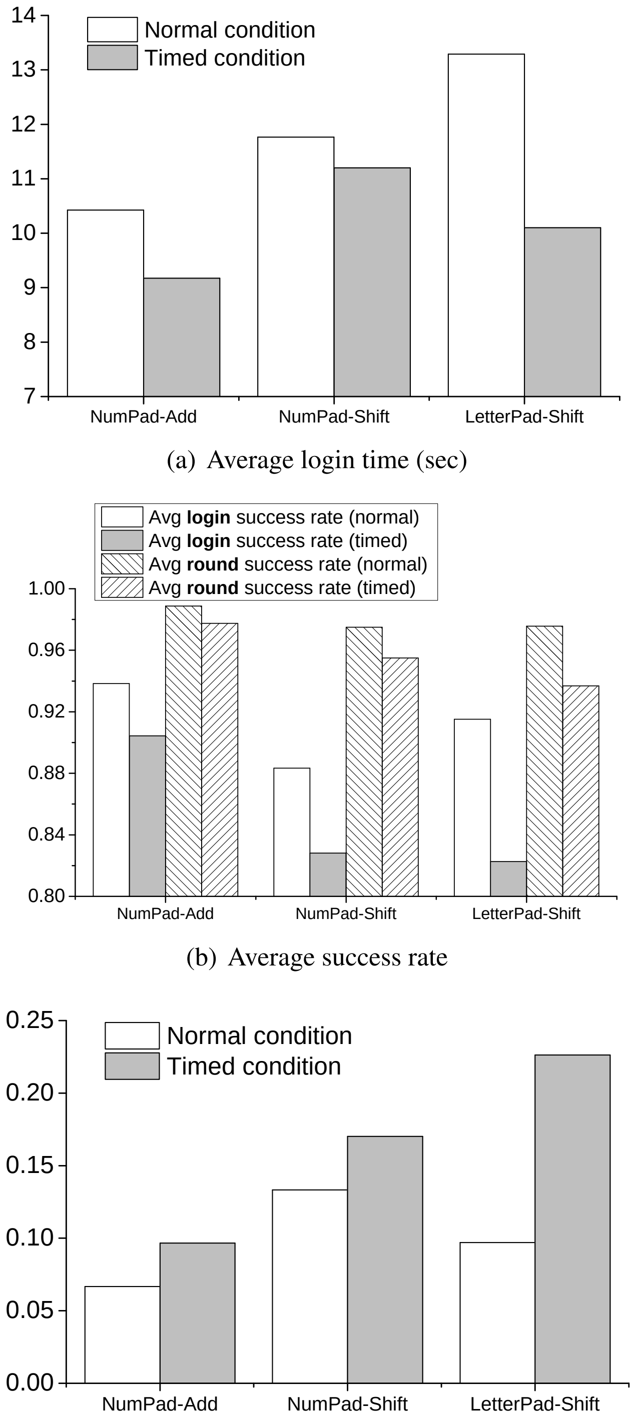
The insignificant results in login accuracy are due to the *ceiling* effect [1], which implies the tests are not sufficiently difficult to distinguish the influence of different test conditions. This effect could be caused by our scheme design, which is not difficult for the participants to use so that the majority of the participants did not make any mistakes during all the tests. This effect will be later discussed in Section 5.5.5. However, even without statistical significance, we still observe the average results of login accuracy become worse for all three tested schemes. Considering the simple design of our schemes, this indicates that time pressure may have a larger influence on the login accuracy of a more complex scheme.

### Influence of Distraction

Figure 5.9 shows the impact of distraction without time pressure. Many participants made a mistake when they saw a distraction task for the first time (however, NumPad-Shift is an exception). For NumPad-Add and LetterPad-Shift shown in Figure 5.9(b), the round success rate returns to a comparable level as the normal condition, after the first time the distraction task appears. This indicates that the distraction task is no longer a surprise for the participants. However, even after the participants get familiar with the distraction tasks, compared to the normal condition, the success rate is still lower, the average edit distance is larger, and the average login time is longer. But the statistical tests show these differences are not significant. Therefore, **H3** and **H4** are not supported in our experiments.

### Influence of Mental Workload

Figure 5.10 shows the impact of mental workload without time pressure. The average login time becomes significantly longer with mental workload ($p$ =.003 for



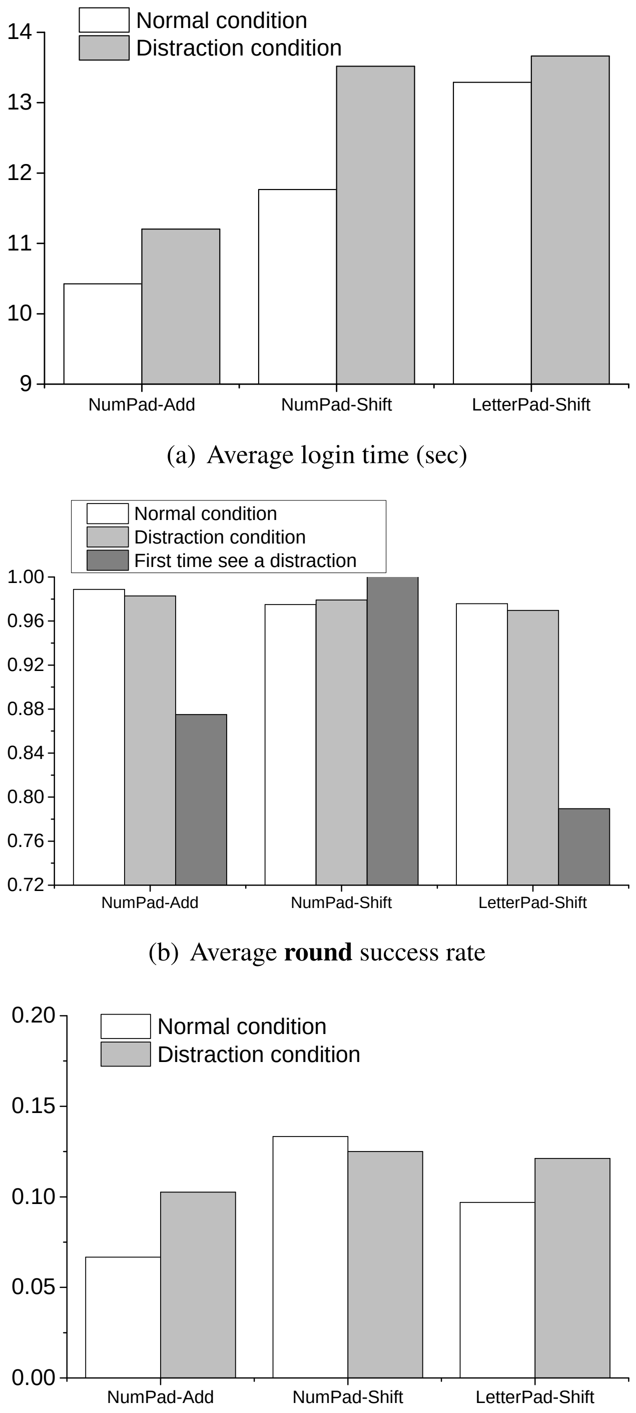
(a) Average login time (sec)

(b) Average success rate

(c) Average edit distance

Figure 5.8: Impact of time pressure

NumPad-Add) due to context switch in users' mind between password inputs and secondary CRT tasks. An extra startup time is required to release the central executive after each CRT task. Our experiment simulates the case when users cannot get rid of other thoughts during password entry. The actual effect of mental work-



(a) Average login time (sec)

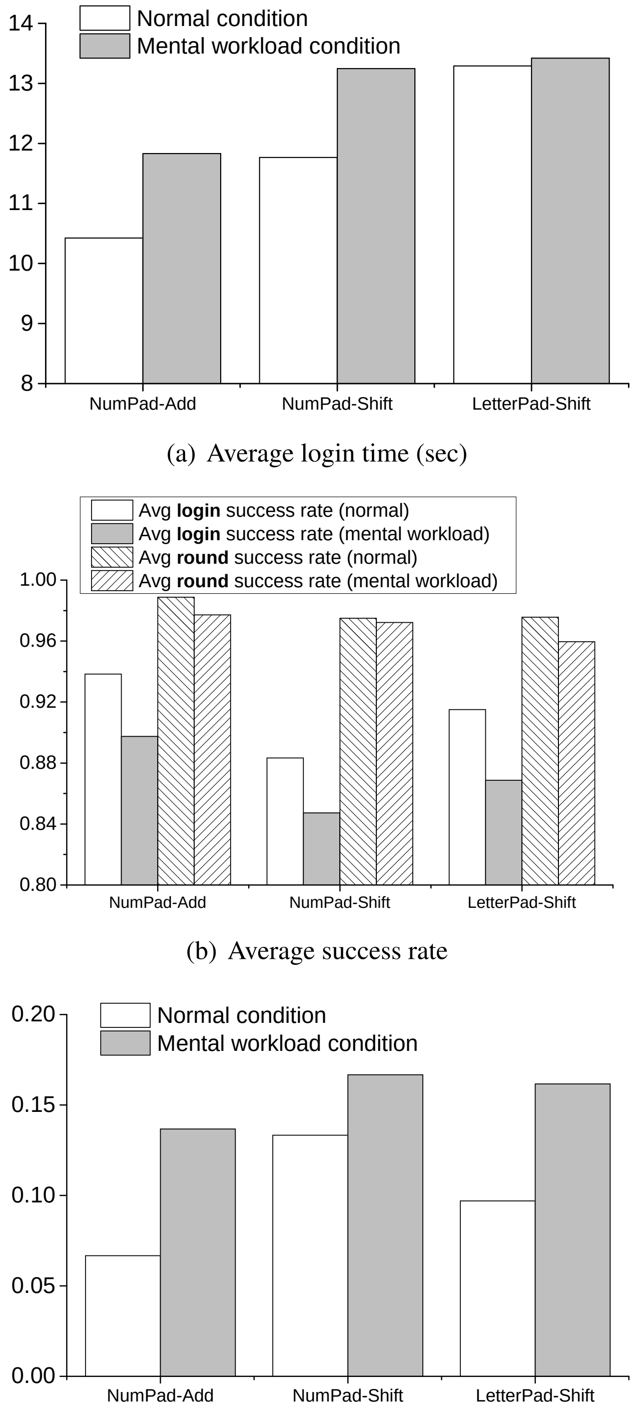
(b) Average **round** success rate

(c) Average edit distance

Figure 5.9: Impact of distraction

load depends on the status of users' mind. The impact may be elevated when the actual mental workload is higher than our CRT tasks. On the other hand, the login accuracy is lower compared to the normal condition but the difference is not significant due to the same ceiling effect mentioned in Section 5.5.4. Therefore,

**H5** is supported and **H6** is not. These results show that persistent mental workload significantly slows the process of password entry for our schemes.



(a) Average login time (sec)

(b) Average success rate

(c) Average edit distance

Figure 5.10: Impact of mental workload

### Performance under Combined Conditions

We also examine the overall impact when distraction or mental workload appears together with time pressure. As expected, compared to their counterparts without time pressure, the average login time becomes shorter (from 10.3 seconds to 11.7 seconds on average), the login success rate becomes even lower (from $81.3\%$ to $87.5\%$), and the average edit distance becomes larger (from 0.151 to 0.243). The statistical tests show the difference in login time is significant ($p =$.009 for NumPad-Add, $p =$.019 for NumPad-Shift, and $p <$.001 for LetterPad-Shift) and the difference in login accuracy is still not significant due to the ceiling effect explained in Section 5.5.4. Therefore, **H7** is supported but **H8** is not. These results show time pressure is still an effective stimulus to speed password entry even in the presence of secondary tasks.

### Effectiveness of Secondary Tasks

Figure 5.11 shows the distribution of the accuracy rate which represents the percentage of secondary tasks being correctly performed by a participant under certain test condition. The overall average accuracy rate is $98.3\%$ across all these test conditions. It implies that the participants did pay attention to these tasks, as they were told that the performance of these tasks also contributes to their scores in the ranking system. Therefore, these CRT tasks work as intended in disturbing participants' mind during password entry.

### Memory Interference by Mental Calculation

Figure 5.12 shows how frequently a participant presses the "*show my password*" button during all tests in a test group. Note that the participants are not allowed to write down their assigned passwords, but they can always click that button in case they forgot their passwords. The overall average value for the total number of times to press the "*show my password*" button is only 0.31 across all three test

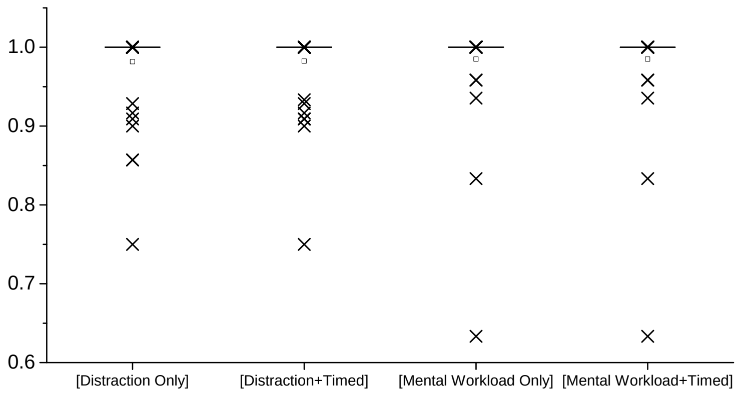


Figure 5.11: Accuracy rate of performing secondary tasks

groups. As shown in Figure 5.12, most users did not use this button during the tests. This implies that the mental calculation involved in the hidden transformation of our schemes does not pose a significant interference on participants' capability of recalling their passwords.

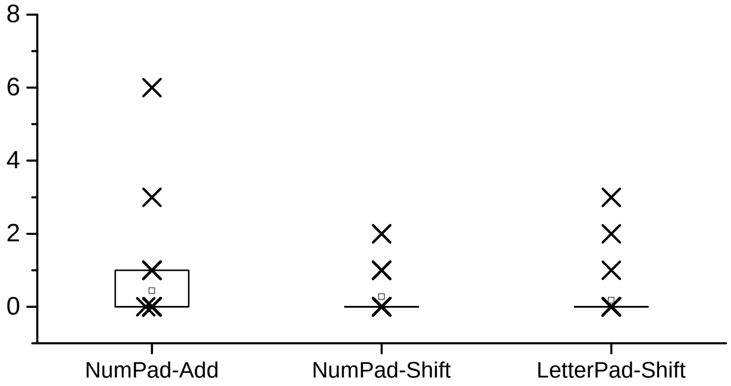
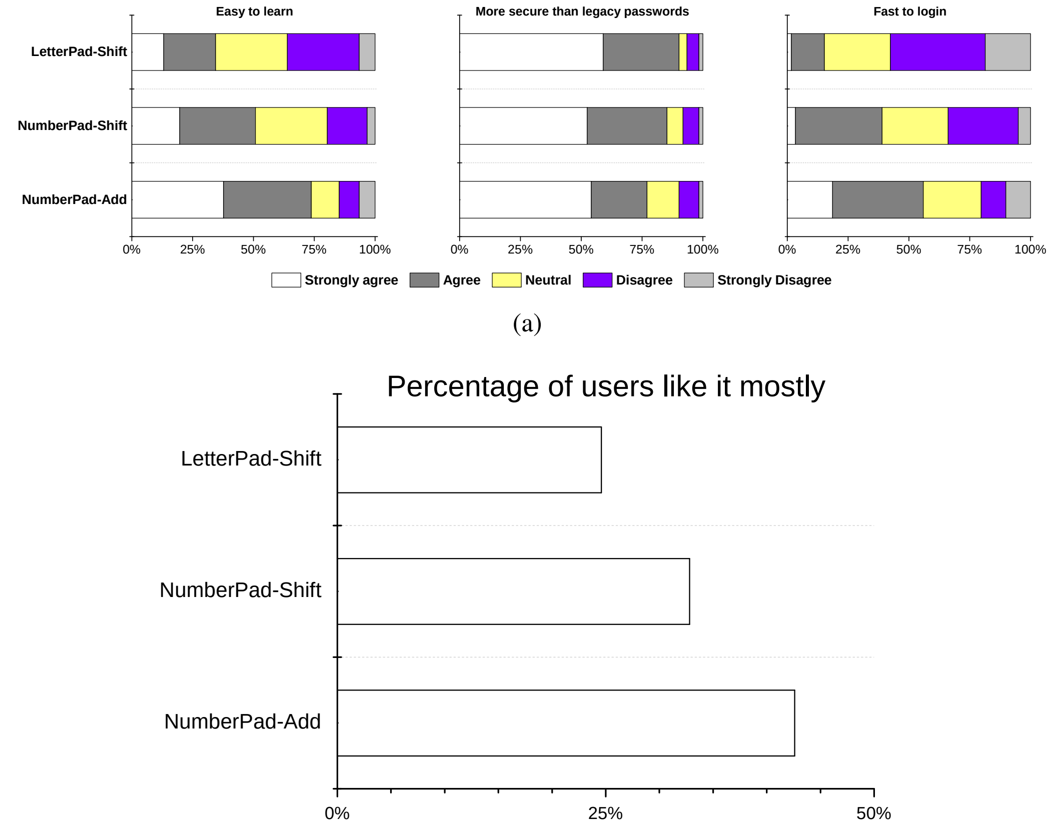


Figure 5.12: Total number of times for each participant to press the "*show my password*" button

### User Perception

Figure 5.13 shows the perception of participants collected from questionnaires. The results indicate that the participants generally feel that our schemes are secure and easy to use. While NumPad-Add is the most popular, the other two schemes also have their favorite users.



(a)

(b)

Figure 5.13: Perception of participants

### 5.5.5 Statistical Test Results

Table 5.2 shows the results of statistical tests on login time. All pairwise tests are Mann-Whitney U, where the statistically significant results are marked with ★. These results indicate that the same test condition may have different impact on the login time of different schemes.

The results of statistical tests on login accuracy are not shown as none of them indicate significance. This is caused by the ceiling effect, which can be observed from the data shown in Table 5.3. Even in the worst case, $50.0\%$ participants did not make any mistakes during all tests in the test condition, which implies our tests are not sufficiently difficult to distinguish these test conditions regarding their influence on the login accuracy of our schemes. This could be caused by the simple design of our schemes such that they are easy to use even in the presence of time pressure, distraction, and mental workload. However, it does not necessarily imply that these

**Average login time** of NumPad-Add - *omnibus* KW $\chi^2_5$=32.423, $p$ <.001

| | | |
|---|---|---|
| normal (10.4) | timed (9.2) | U=551, $p$ =.017 ★ |
| | distraction (11.2) | U=679, $p$ =.184 |
| | distraction+timed (10.3) | U=878, $p$ =.989 |
| | mental workload (11.8) | U=515, $p$ =.003 ★ |
| | mental workload+timed (10.7) | U=696, $p$ =.319 |
| distraction (11.2) | distraction+timed (10.3) | U=718, $p$ =.107 |
| mental workload (11.8) | mental workload+timed (10.7) | U=558, $p$ =.009 ★ |

**Average login time** of NumPad-Shift - *omnibus* KW $\chi^2_5$=11.965, $p$ =.034

| | | |
|---|---|---|
| normal (11.7) | timed (11.2) | U=666, $p$ =.199 |
| | distraction (13.5) | U=645, $p$ =.137 |
| | distraction+timed (11.7) | U=727, $p$ =.485 |
| | mental workload (13.3) | U=655, $p$ =.164 |
| | mental workload+timed (11.4) | U=644, $p$ =.135 |
| distraction (13.5) | distraction+timed (11.7) | U=565, $p$ =.024 ★ |
| mental workload (13.3) | mental workload+timed (11.4) | U=555, $p$ =.019 ★ |

**Average login time** of LetterPad-Shift - *omnibus* KW $\chi^2_5$=49.252, $p$ <.001

| | | |
|---|---|---|
| normal (13.2) | timed (10.1) | U=294, $p$ <.001 ★ |
| | distraction (13.6) | U=774, $p$ =.667 |
| | distraction+timed (11.0) | U=413, $p$ <.001 ★ |
| | mental workload (13.4) | U=653, $p$ =.116 |
| | mental workload+timed (11.5) | U=472, $p$ =.002 ★ |
| distraction (13.6) | distraction+timed (11.0) | U=422, $p$ <.001 ★ |
| mental workload (13.4) | mental workload+timed (11.5) | U=631, $p$ =.075 |

Table 5.2: The results of statistical tests on login time (sec)

factors will not significantly influence the login accuracy of other user authentication schemes. Since the average results of login accuracy are observed to be worse due to the presence of these factors in our tests, we expect they would have a more significant influence on other schemes with higher complexity.

| | **NumPad-Add** | **NumPad-Shift** | **LetterPad-Shift** |
|---|---|---|---|
| normal | 82.9% | 67.5% | 75.6% |
| timed | 78.0% | 62.5% | 53.7% |
| distraction | 80.5% | 70.0% | 63.4% |
| distraction+timed | 70.7% | 55.0% | 58.5% |
| mental workload | 75.6% | 57.5% | 65.9% |
| mental workload+timed | 65.9% | 50.0% | 51.2% |

Table 5.3: Evidence for the ceiling effect in statistical tests on login accuracy. Each cell in this table shows the percentages of the participants who did not make any mistakes in a test condition.

### 5.5.6 Comparison with Legacy Passwords

Table 5.4 compares CoverPad with legacy passwords based on the design metrics developed in Chapter 4.

|  | **CoverPad Schemes** | **Legacy Passwords** |
|---|---|---|
| Built-in security | **Yes** | No |
| Human capability | Shield, recall, simple arithmetic or geometric operation, click | Recall, click |
| Device availability | Keypad on a touchscreen | Any keypad |
| Environmental adaptation | Single hand with a support | Single hand with a support |
| Space-memory ratio | $n^k/k$ | $n^k/k$ |
| Screen utility rate | 1.0 | 1.0 |
| No-leakage | **Yes** | No |
| Cognitive operation list | 1 recall + 1 hint reading + 1 transforming $(3m)$ | 1 recall $(1m)$ |
| Channel effectiveness | Vision input + click-based output | Vision input + click-based output |
| Password storage | Ciphertext | Ciphertext |

Table 5.4: Comparison between CoverPad and legacy passwords using LRPE design metrics

Table 5.5 further gives a comparison based on the *usability-deployability-security* metrics proposed in [15], where a metric is not shown if neither our schemes nor legacy passwords offer corresponding benefit. We have the following observations in comparison. 1) Our schemes are rated as not *mature* since they are just proposed and have not been widely deployed. 2) Our schemes are not *server-compatible*, as most current servers support only static and replayable passwords, which could be changed in the near future. 3) Our schemes are *quasi-resilient-to-internal-observation* in a sense that any key logger or malware which fails to capture the hidden transformation causes no password leakage. Overall, these tables show that our schemes significantly improve the security strength while retaining most benefits of legacy passwords.

|  | Nothing-to-Carry | Easy-to-Learn | Efficient-to-Use | Infrequent-Errors | Easy-Recovery-from-Loss | Accessible | Negligible-Cost-per-User | Server-Compatible | Browser-Compatible | Mature | Non-Proprietary | Resilient-to-Physical-Observation | Resilient-to-Targeted-Impersonation | Resilient-to-Internal-Observation | Resilient-to-Theft | No-Trusted-Third-Party | Requiring-Explicit-Consent | Unlinkable |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CoverPad Schemes** | ● | ● | ○ | ○ | ● | ● | ● |  | ● |  | ● | ● | ○ | ○ | ● | ● | ● | ● |
| **Legacy Passwords** | ● | ● | ● | ○ | ● | ● | ● | ● | ● | ● | ● |  | ○ |  | ● | ● | ● | ● |

Table 5.5: Comparison between CoverPad and legacy passwords using usability-deployability-security metrics [15]. ● = offer the benefit, ○ = almost offer the benefit, *no circle* = does not offer the benefit

## 5.6 Other Practical Issues and Limitations

### 5.6.1 Eavesdropping Attacks

Eavesdropping attacks such as vision-based eavesdropping may require the physical presence of an adversary, which limits the scale of their threat. However, the scale of attacks is not the only factor that determines the impact of attacks, which is also decided by the severity of potential losses. If a victim is an important person in a company, password leakage may lead to disclosing sensitive corporate data, which would provide sufficient incentives to an adversary. While internal attacks such as malware and logic key logger could be prevented by properly updating and configuring the computing system [53, 4, 11, 67] used by the victim, it is difficult to effectively mitigate the threat of external eavesdropping attacks due to inevitable exposure of human-computer interaction during the entry of legacy passwords. This threat becomes more serious in scenarios when a mobile device is used in public places.

Nevertheless, the threat of external eavesdropping attacks can be effectively mitigated with CoverPad. Besides enhanced security features, our scheme retains most benefits of legacy passwords and can be implemented on *commodity* devices. Our scheme is not only applicable to mobile devices but also other devices equipped with touchscreen. For example, many ATM machines have been deployed with touchscreen. Our scheme can be deployed on these machines to mitigate the threat of the ATM skimming attack [43].

### 5.6.2 Device Screen Size

Although we implement our scheme on Apple iPad, it could be easily adapted to other screen sizes, as illustrated in Figure 5.14. For a mobile phone with a small touchscreen like Apple iPhone, a user can use a hand $A$ to perform the hand-shielding gesture, and use the other hand $B$ to hold the phone. The thumb on hand $B$ can be used to press the response keys. For a mobile phone with a larger touchscreen like Samsumg Galaxy Note II, a user may not be able to click all the keys with the thumb of hand $B$ that holds the device. To deal with this situation, he only needs to use one hand $A$ to perform the hand-shielding gesture and key pressing sequentially. Once the user raises his hand before pressing a key, the hidden transformation immediately disappears because the gesture is no longer detected by the touchscreen. Meantime, the user does not need to worry about whether the actual keys pressed or the finger movements during key pressing may be observed by an adversary, as the sequence of pressed keys alone does not leak any information about the underlying password as analyzed in Section 5.4.

### 5.6.3 Limitations

Ecological validity is a challenging issue in any user study. Like most prior research [35, 48, 44, 25, 42], our experiments engage only university students. These participants are younger and more educated compared to the general population.

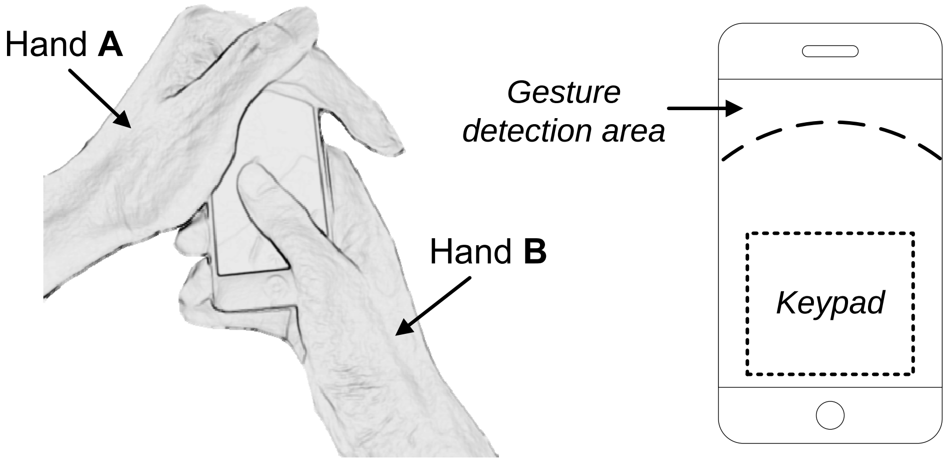


Figure 5.14: Conceptual demonstration on a small screen device

Therefore, usability evaluation may vary with other populations. Our experiments are also restricted by the sample size, which may affect the results of statistical tests. Typical examples are the insignificant results on the login accuracy of our schemes. Moreover, our user study does not include experiments on memory effects (e.g. forgetting). Since our scheme uses the same alphabet and password composition as legacy passwords, the users may use the same coping strategies to help themselves to memorize the passwords in our scheme. The impact of memory effects on the user performance would be similar to legacy passwords as shown in the prior literature [24, 62].

## 5.7 Discussion

In this work, we proposed a leakage-resilient password entry scheme leveraging on the touchscreen feature of mobile devices. It achieves leakage resilience while preserving most benefits of legacy passwords. Three variants of this scheme were implemented. The practicability of our scheme was evaluated in an extended user study that incorporates new experiments to examine the influence of additional test conditions related to time pressure, distraction, and mental workload. These conditions were tested for the first time in the evaluation of user authentication schemes. Among these conditions, time pressure and mental workload were shown to have significant impacts on user performance. Therefore, we suggest including these conditions in the evaluation of user authentication schemes in the future research.

# Chapter 6

# Dissertation Conclusion and Future Work

## 6.1 Summary of Contribution

This dissertation makes contributions on understanding and solving the problem of designing secure and usable leakage-resilient password entry (LRPE) schemes.

Our first work provided a comprehensive analysis for the inherent limitations in LRPE design. We analyzed the impacts of two types of generic attacks, brute force and statistical attacks, on the existing schemes designed for unaided humans when a trusted device is unavailable. We introduced five principles that are necessary to achieve leakage resilience. Usability costs for these principles were analyzed. Our findings indicate that either high memory demand or high cognitive workload is unavoidable in the design of secure LRPE schemes without utilizing trusted devices. To further understand the tradeoff between security and usability, we established the first quantitative analysis framework on usability costs. Our result shows that there is a strong tradeoff between security and usability, indicating that an unaided human may not be competent enough to use a secure LRPE scheme in practical settings. This work has been published in the Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS 2012) [75], and won the distinguished paper award.

In the second work, we made the first attempt to provide a comprehensive understanding for the challenges of designing practical LRPE schemes even in the presence of trusted devices. We proposed a broad set of design metrics to evaluate LRPE schemes from different perspectives, including security-usability relations, built-in security, and universal accessibility. These metrics were designed to identify the potential limitations before conducting user studies. They were applied to existing LRPE schemes to reveals their limitations. Our analysis further showed that it is possible to overcome these limitations by improving the design according to the proposed metrics. This work has been submitted to a security journal at the time when this dissertation was submitted.

Finally, we proposed a leakage-resilient password entry scheme leveraging on the touchscreen feature of mobile devices. It achieves leakage resilience while preserving most benefits of legacy passwords. Three variants of this scheme were implemented. The practicability of our scheme was evaluated in an extended user study that incorporates new experiments to examine the influence of additional test conditions. These conditions were tested for the first time in the evaluation of user authentication schemes. Among these conditions, time pressure and mental workload were shown to have significant impacts on user performance. Therefore, we suggest including these conditions in the evaluation of user authentication schemes in the future research. This work has been published in the Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security (AsiaCCS 2013) [76].

## 6.2 Future Direction

Designing a more secure but still usable user authentication scheme is the holy grail of the research on user authentication. After many failures [44, 61, 23, 25, 42, 13, 12] in this quest of replacing legacy passwords, more and more tradeoffs in the design of such schemes are identified. The discovery of these tradeoffs reveals the crucial truth that it may not be possible to design a perfect user authentication scheme that has *all desired* security properties without sacrificing usability [75, 20]. The underlying reason is that human beings are lack of necessary capabilities to perform calculation and memory operations required by a theoretically secure scheme. Since the imperfection seems inevitable, how to find a better balance between security and usability in a scheme design will remain a challenging research problem until the fundamental limitation caused by the incompetence of human beings is resolved.

Theoretically, there are two possible approaches to overcome this limitation. The first approach is to enhance nature human capabilities to a level that is sufficient to perform the required operations. The recent development of genetic engineering sheds light on this approach, where future generations with genetic modification will be more intelligent so that they can overcome the capability barrier required by a theoretically secure scheme. However, this may not be realized in our generation. Therefore, we may need to pursue the second approach, relying on extra devices to *complement* human capabilities. As shown in Chapter 5 of this dissertation, new mobile devices with touchscreen feature can be used to design a practical leakage-resilient password entry scheme [76]. Correspondingly, other devices can be used to design user authentication schemes with other security properties. The number of possible choices of such devices will keep increasing as technology evolves. For example, the recent development on wearable computing devices indicates that device-assisted user authentication would be a promising direction, as these wearable devices provide seamless user experience for human-computer interaction.

Besides the design problem of user authentication schemes, how to *objectively* evaluate user authentication schemes has also been a challenge for a long time. More and more researchers are now aware of the importance of user studies and report user performance with real user data instead of mere arguments. But the evaluation methodology of user authentication schemes is still in its preliminary stage.

Lack of consistency is common in the existing literature [44, 61, 23, 25, 42, 13, 12], where it is difficult to compare the evaluation results from different work and certain evaluations are not well controlled so that it is even difficult to reproduce the same result [15]. Therefore, the standardization of the evaluation methodology is an important work that would advance the field of user authentication. Chapter 4 in this dissertation made an initial step by developing the design metrics for leakage-resilient password entry. The proposed metrics could be used as a reference to define more general metrics for other user authentication schemes. This standardization process is much more complicated than it appears. Since different schemes may have different design objectives, it is difficult to define unified criteria for all kinds of schemes. Furthermore, the evaluation methodology needs to include not only usability but also security, as unintended user behaviors may significantly compromise a security property that can only be achieved when a user performs an expected action [24]. Many other factors need to be considered, which makes it an interesting and challenging problem for future research.

# Bibliography

[1] Ceiling effect. http://en.wikipedia.org/wiki/Ceiling_effect.

[2] R. Anderson. Why cryptosystems fail. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 215–227, 1993.

[3] Androidcommunity. Samsung galaxy siii display specs. http://androidcommunity.com/samsung-galaxy-siii-display-specs-edge-out-iphone-5-20121002/.

[4] Apple. Mac os x. www.apple.com/osx/.

[5] H. J. Asghar, S. Li, J. Pieprzyk, and H. Wang. Cryptanalysis of the convex hull click human identification protocol. In *Proceedings of the 13th international conference on Information security*, pages 24–30, 2010.

[6] H. J. Asghar, J. Pieprzyk, and H. Wang. A new human identification protocol and coppersmith's baby-step giant-step algorithm. In *Proceedings of the 8th international conference on Applied cryptography and network security*, pages 349–366, 2010.

[7] R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, 1974.

[8] A. D. Baddeley. *The Essential Handbook of Memory Disorders for Clinicians*, chapter 1, pages 1–13. John Wiley & Sons, 2004.

[9] A. D. Baddeley and G. Hitch. Working memory. *The psychology of learning and motivation*, 8:47–89, 1974.

[10] X. Bai, W. Gu, S. Chellappan, X. Wang, D. Xuan, and B. Ma. Pas: Predicate-based authentication services against powerful passive adversaries. In *Proceedings of the 2008 Annual Computer Security Applications Conference*, pages 433–442, 2008.

[11] O. Begemann. Remote view controllers in ios 6. http://oleb.net/blog/2012/10/remote-view-controllers-in-ios-6.

[12] A. Bianchi, I. Oakley, V. Kostakos, and D. S. Kwon. The phone lock: audio and haptic shoulder-surfing resistant pin entry methods for mobile devices. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 197–200, 2011.

[13] A. Bianchi, I. Oakley, and D.-S. Kwon. Obfuscating authentication through haptics, sound and light. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, pages 1105–1110, 2011.

[14] R. Biddle, S. Chiasson, and P. C. van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys*, 44(4), 2012.

[15] J. Bonneau, C. Herley, P. van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Proceedings of IEEE Symposium on Security and Privacy 2012*, 2012.

[16] P. Bright. RSA finally comes clean: SecurID is compromised, 2011. arstechnica.com/security/news/2011/06/rsa-finally-comes-clean-securid-is-compromised.ars.

[17] J. I. D. Campbell and Q. Xue. Cognitive arithmetic across cultures. *Journal of Experimental Psychology: General*, 130(2):299–315, 2001.

[18] F. B. Colavita. Human sensory dominance. *Attention, Perception, & Psychophysics*, 16(2):409–412, 1974.

[19] L. Corbina and J. Marquer. Effect of a simple experimental control: The recall constraint in sternberg's memory scanning task. *European Journal of Cognitive Psychology*, 20(5):913–935, 2008.

[20] B. Coskun and C. Herley. Can "something you know" be saved? In *Proceedings of the 11th international conference on Information Security*, pages 421–440, 2008.

[21] N. Cowan. The magical number 4 in short-term memory: a reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114, 2001.

[22] F. I. Craik and J. M. McDowd. Age differences in recall and recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13(3):474–479, 1987.

[23] A. De Luca, M. Denzel, and H. Hussmann. Look into my eyes!: can you guess my password? In *Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 7:1–7:12, 2009.

[24] A. De Luca, M. Langheinrich, and H. Hussmann. Towards understanding atm security: a field study of real world atm use. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, 2010.

[25] A. De Luca, E. von Zezschwitz, and H. Husmann. Vibrapass: secure authentication based on shared lies. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 913–916, 2009.

[26] P. Dunphy, A. P. Heiner, and N. Asokan. A closer look at recognition-based graphical passwords on mobile devices. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, pages 3:1–3:12, 2010.

[27] D. L. Fisher. Central capacity limits in consistent mapping, visual search tasks: Four channels or more? *Cognitive Psychology*, 16(4):449–484, 1984.

[28] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov. The most dangerous code in the world: validating ssl certificates in non-browser software. In *Proceedings of the 19th ACM Conference on Computer and Communications Security*, pages 38–49, 2012.

[29] J. J. Gibson. Adaptation, after-effect, and contrast in the perception of curved lines. *Journal of Experimental Psychology*, 20(6):553–569, 1937.

[30] L. Ginzburg, P. Sitar, and G. K. Flanagin. User authentication system and method. US Patent 7,725,712, SyferLock Technology Corporation, 2010.

[31] F. Gobet, P. C. R. Lane, S. Croker, P. C. H. Cheng, G. Jones, I. Oliver, and J. M. Pine. Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5(6):236–243, 2001.

[32] P. Golle and D. Wagner. Cryptanalysis of a cognitive authentication scheme (extended abstract). In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 66–70, 2007.

[33] Google. Google glass. http://plus.google.com/+projectglass.

[34] R. M. Hogan and W. Kintsch. Differential effects of study and test trials on long-term recognition and recall. *Journal of Verbal Learning and Verbal Behavior*, 10(5):562–567, 1971.

[35] N. J. Hopper and M. Blum. Secure human identification protocols. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, pages 52–66, 2001.

[36] T. S. Horowitz and J. M. Wolfe. Search for multiple targets: Remember the targets, forget the search. *Perception & Psychophysics*, 63(2):272–285, 2001.

[37] H. B. Hotel. ipad - free for every hotel guest. http://www.hollmann-beletage.at/en/ipad.

[38] I. Imbo and A. Vandierendonck. The role of phonological and executive working memory resources in simple arithmetic strategies. *European Journal Of Cognitive Psychology*, 19(6):910–933, 2007.

[39] A. Imran. ipads can now be used as public kiosks. http://www.redmondpie.com/ipad-public-kiosks-video/.

[40] A. R. Jensen. Process differences and individual differences in some cognitive tasks. *Intelligence*, 11(2):107–136, 1987.

[41] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez. Guess again (and again and again) measuring password strength by simulating password-cracking algorithms. In *Proceedings of IEEE Symposium on Security and Privacy 2012*, 2012.

[42] D. Kim, P. Dunphy, P. Briggs, J. Hook, J. W. Nicholson, J. Nicholson, and P. Olivier. Multi-touch authentication on tabletops. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1093–1102, 2010.

[43] Krebs. Would you have spotted the fraud? http://krebsonsecurity.com/2010/01/would-you-have-spotted-the-fraud.

[44] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd symposium on Usable privacy and security*, pages 13–19, 2007.

[45] M. Lei, Y. Xiao, S. Vrbsky, C.-C. Li, and L. Liu. A virtual password scheme to protect passwords. In *Proceedings of IEEE International Conference on Communications*, pages 1536–1540, 2008.

[46] S. Li, H. Asghar, J. Pieprzyk, A.-R. Sadeghi, R. Schmitz, and H. Wang. On the security of pas (predicate-based authentication service). In *Proceedings of the 2009 Annual Computer Security Applications Conference*, pages 209–218, 2009.

[47] S. Li, S. A. Khayam, A.-R. Sadeghi, and R. Schmitz. Breaking randomized linear generation functions based virtual password system. In *Proceedings of the 2010 IEEE International Conference on Communications*, pages 23–27, 2010.

[48] S. Li and H. yeung Shum. Secure human-computer identification (interface) systems against peeping attacks: SecHCI. In *Cryptology ePrint Archive, Report 2005/268*, 2005.

[49] J. Long and J. Wiles. *No Tech Hacking: A Guide to Social Engineering, Dumpster Diving, and Shoulder Surfing*. Syngress, 2008.

[50] I. Martinovic, D. Daviesy, M. Franky, D. Peritoy, T. Rosz, and D. Song. On the feasibility of side-channel attacks with brain-computer interfaces. In *Proceedings of the 21st USENIX Security Symposium*, 2012.

[51] T. Matsumoto. Gummy and conductive silicone rubber fingers. In *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, pages 574–576, 2002.

[52] T. Matsumoto and H. Imai. Human identification through insecure channel. In *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, pages 409–421, 1991.

[53] Microsoft. Windows 8. windows.microsoft.com.

[54] F. Miller. *Telegraphic code to insure privacy and secrecy in the transmission of telegrams*. C.M. Cornwell, 1882.

[55] P. A. Nobel and R. M. Shiffrin. Retrieval processes in recognition and cued recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27(2):384–413, 2001.

[56] T. Perkovic, A. Mumtaz, Y. Javed, S. Li, S. A. Khayam, and M. Cagalj. Breaking undercover: Exploiting design flaws and nonuniform human behavior. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, 2011.

[57] D. Rohrer and J. Wixted. An analysis of latency and interresponse time in free recall. *Memory. & Cognition*, 22(5):511–524, 1994.

[58] V. Roth, K. Richter, and R. Freidinger. A pin-entry method resilient against shoulder surfing. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 236–245, 2004.

[59] RSA. RSA SecurID two-factor authentication, 2011. www.rsa.com/products/securid/sb/10695_SIDTFA_SB_0210.pdf.

[60] J. H. Saltzer and M. D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, 1975.

[61] H. Sasamoto, N. Christin, and E. Hayashi. Undercover: authentication usable in front of prying eyes. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 183–192, 2008.

[62] R. Shay, P. G. Kelley, S. Komanduri, M. L. Mazurek, B. Ur, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor. Correct horse battery staple: exploring the usability of system-assigned passphrases. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, 2012.

[63] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on ssh. In *Proceedings of the 10th USENIX Security Symposium*, 2001.

[64] Spycop. Hardware keylogger detection. http://spycop.com/keyloggerremoval.htm.

[65] S. Sternberg. Memory-scanning: Mental processes revealed by reaction-time experiments. *American Scientist*, 57:421–457, 1969.

[66] L. J. Stifelman, B. Arons, C. Schmandt, and E. A. Hulteen. Voicenotes: a speech interface for a hand-held voice notetaker. In *Proceedings of the INTERCHI '93 conference on Human factors in computing systems*, pages 179–186, 1993.

[67] TCG. Trusted computing group. www.trustedcomputinggroup.org.

[68] T. D. Tomlinson, D. E. Huber, C. A. Rieth, and E. J. Davelaarc. An interference account of cue-independent forgetting in the no-think paradigm. *Proceedings of the National Academy of Sciences*, 106(37):15588–15593, 2009.

[69] N. Unsworth and R. W. Engle. The nature of individual differences in working memory capacity: Active maintenance in primary memory and controlled search from secondary memory. *Psychological Review*, 114(1):104–132, 2007.

[70] E. Vogel and M. Machizawa. Neural activity predicts individual differences in visual working memory capacity. *Nature*, 428(6984):748–751, 2004.

[71] D. Weinshall. Cognitive authentication schemes safe against spyware (short paper). In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pages 295–300, 2006.

[72] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proceedings of the working conference on Advanced visual interfaces*, pages 177–184, 2006.

[73] G. F. Woodman and M. M. Chun. The role of working memory and long-term memory in visual search. *Visual Cognition*, 14(4-8):808–830, 2006.

[74] G. F. Woodman and S. J. Luck. Visual search is slowed when visuospatial working memory is occupied. *Psychonomic Bulletin and Review*, 11(2):269–274, 2004.

[75] Q. Yan, J. Han, Y. Li, and R. H. Deng. On limitations of designing leakage-resilient password systems: Attacks, principles and usability. In *Proceedings of the 19th Annual Network and Distributed System Security Symposium*, 2012.

[76] Q. Yan, J. Han, Y. Li, J. Zhou, and R. H. Deng. Designing leakage-resilient password entry on touchscreen mobile devices. In *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security*, 2013.

[77] ZDNet. More ipad love: Now hotels offer ipad to customers. http://www.zdnet.com/blog/apple/more-ipad-love-now-hotels-offer-ipad-to-customers/6850.

[78] H. Zhao and X. Li. S3pas: A scalable shoulder-surfing resistant textual-graphical password authentication scheme. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops - Volume 02*, pages 467–472, 2007.