2013

Exploiting Human Factors in User Authentication

Payas GUPTA
Singapore Management University, payas.gupta.2008@smu.edu.sg

Exploiting Human Factors in User Authentication

by

Payas Gupta

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

Dissertation Committee:

Debin GAO (Supervisor/Chair)
Assistant Professor of Information Systems
Singapore Management University

Xuhua DING (Co-Supervisor)
Associate Professor of Information Systems
Singapore Management University

Robert DENG Huijie
Professor of Information Systems
Singapore Management University

Zhenkai LIANG
Assistant Professor of Computer Science
National University of Singapore

Singapore Management University

2013

Exploiting Human Factors in User Authentication

Payas Gupta

Abstract

Our overarching issue in security is the human factor – and dealing with it is perhaps one of the biggest challenges we face today. Human factor is often described as the weakest part of a security system and users are often described as the weakest link in the security chain. In this thesis, we focus on two problems which are caused by human factors in user authentication and propose respective solutions. a) Secrecy information inference attack – publicly available information can be used to infer some secrecy information about the user. b) Coercion attack – where an attacker forces a user to handover his/her secret information such as account details and password.

In the secrecy information inference attack, an attacker can use publicly available data to infer secrecy information about a victim. We should be prudent in choosing any information as secrecy information in user authentication. In this work, we exploit public data extracted from Facebook to infer users' interests. Such interests can also found on their profile pages but such pages are often private. Our experiments conducted on over more than $34,000$ public pages collected from Facebook show that our inference technique can infer interests which are often hidden by users with moderate accuracy. Using the inferred interests, we also demonstrate a secrecy information inference attack to break a preference based backup authentication system Blue Moon^{TM}. To mitigate the effect of secrecy information inference attack, we propose a new authentication mechanism based on user's cellphone usage data which is often private. The system generates *memorable* and dynamic fingerprints which can be used to create authentication challenges. In particular, in this work, we explore if the generated behavioral fingerprints are *memorable*

enough to be remembered by end users to be used for authentication credentials. We demonstrate the application of memorable fingerprints by designing an authentication application on top of it. We conducted an extensive user study that involved collecting about one month of continuous usage data from 58 Symbian and Android smartphone users. Results show that the fingerprints generated are remembered by the user to some extent and that they were moderately secure against attacks even by family members and close friends.

The second problem which we focus in this thesis is human vulnerability to coercion attacks. In such attacks, the user is forcefully asked by an attacker to reveal the secret/key to gain access to the system. Most authentication mechanisms today are vulnerable to coercion attacks. We present a novel approach in generating cryptographic keys to fight against coercion attacks. Our technique incorporates a measure of user's emotional status using skin conductance (which changes when the user is under coercion) into the key generation process. A preliminary user study with 39 subjects was conducted which shows that our approach has moderate false acceptance and false rejection rates. Furthermore, to meet the demand of scalability and usability, many real-world authentication systems have adopted the idea of responsibility shifting, where a user's responsibility of authentication is shifted to another entity, usually in case of failure of the primary authentication method. In a responsibility shifting authentication scenario, a human helper who is involved in regaining access, is vulnerable to coercion attacks. In this work, we report our user study on 29 participants which investigates the helper's emotional status when being coerced to assist in an attack. Results show that the coercion causes involuntary skin conductance fluctuation on the helper, which indicates that he/she is nervous and stressed. The results from the two studies show that the skin conductance is a viable approach to fight against coercion attacks in user authentication.

Contents

List of Figures

List of Tables

Publications arising from this thesis

1. Payas Gupta, Swapna Gotipatti, Jing Jiang, and Debin Gao. Your love is public now: Questioning the use of personal information in authentication. In *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '13, New York, NY, USA, 2013. ACM (**Chapter 3**)

2. Payas Gupta, Tan Kiat Wee, Narayan Ramasubbu, David Lo, Debin Gao, and Rajesh Krishna Balan. Human: Creating memorable fingerprints of mobile users. In *Tenth Annual IEEE International Conference on Pervasive Computing and Communications, PerCom*. IEEE, 2012 (**part of Chapter 4**)

3. Payas Gupta, Kiat Wee Tan, Narayan Ramasubbu, David Lo, Debin Gao, and Rajesh Krishna Balan. Design and implementation of human memorable fingerprints. Technical Report SMU-SIS-13-100, Singapore Management University, Mar 2013 (**part of Chapter 4**)

4. Payas Gupta and Debin Gao. Fighting coercion attacks in key generation using skin conductance. In *Proceedings of the 19th USENIX conference on Security*, USENIX Security'10, Berkeley, CA, USA, 2010. USENIX Association (**Chapter 5**)

5. Payas Gupta, Xuhua Ding, and Debin Gao. Coercion resistance in authentication responsibility shifting. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, New York, NY, USA, 2012. ACM (**Chapter 6**)

Acknowledgements

To the casual observer, a doctoral dissertation may appear to be solitary work. However, to complete a project of this magnitude requires a network of support, and I am indebted to many people.

This thesis would not have been possible without the support of many people. Many thanks to my adviser, Debin Gao. I am very grateful for the opportunity to work with him and could not have imagined having a better mentor for my Ph.D study. Working with him has definitely sharpened my research ability and helped me grow as an individual and a professional. Perhaps the most important lesson I learned from him was how to effectively convey my thoughts and ideas in both written and spoken mediums. I admire his ability to balance research interests and personal pursuits.

Furthermore, I am very grateful to my committee members, Robert Deng, Xuhua Ding, and Zhenkai Liang, for their guidance, support, insightful comments and hard questions.

In addition, I would like to express my deepest appreciation to Steve Miller, Dean of School of Information Systems, Singapore Management University, whose enthusiasm for "quality research" and "big ideas" have significantly improved my work and inspired many new research directions.

Besides my advisor and committee members, I am very privileged to work with Rajesh Krishna Balan, Narayan Ramasubbu and David Lo. Their technical excellence and tremendous grasp of experimental issues had a great impact on me. Without them I could not have excelled in conducting user studies effectively.

I would like to express my sincere thanks to Living Analytics Research Center at Singapore Management University for giving me the opportunity to spend a year in Carnegie Mellon University as a research scholar. Furthermore, I am very grateful to Adrian Perrig, for his insightful comments in my studies, for many motivating discussions and guidance throughout my stay at CMU. The work done in collaboration with him has helped me to explore other areas.

Very special thanks to Ong Chew Hong. She helped me in all my user studies, answering to all sorts of questions and going out of her way to help me out in recruiting participants. None of the work in this thesis and my other research projects could have been possible without her support. She is a gem of a person. Above all, she made me feel a friend, which I appreciate from my heart. Besides her I would also like to thank Seow Pei Huan for her help.

During my stay at Singapore Management University and at Carnegie Mellon University, I have made many friends and they have been vital in making the Ph.D. process a fun and enriching experience. I want to thank Pawan Gupta, Varun Khanna, Salman Hamid, Aditya Maru, Vivek Desai, Roger Cherian, Yash Divadkar, Ankit Birla, Prem Prasoon, Darshan Santani, Husain Kagalwala, Manu Nahar, Sudhanshu Nahata, Meryl Gotlieb and Nancy Beatty for taking an extra mile to help me out in day to day life. In addition, I have been very privileged to get to know and to collaborate with many other great people who became friends over the last several years. I appreciate Swapna Gottipati, Yan Qiang, Kartik Muralidharan, Sougata Sen, Jun Han, Han Jin, Noi Sian and Tey Chee Meng for their friendship, collaboration and encouragement.

I would like to specially thank Varunika Goyal for her love and encouragement. And, thank you for your support when I have needed it the most. Thank you with all my heart!

Above all, I am thankful to my parents and my sister Kopal Gupta for their support and love who endured this long process with me.

To my loving parents, Kusum Gupta and Vijendra Gupta.

Chapter 1

Introduction

An integral part of computer security is user authentication, which seeks to confirm the identity of a user for the purpose of granting individual users access to their respective accounts. All security access methods are based on four fundamental pieces of information: something the user is, something the user has, something the user knows, and recently proposed someone the user knows [23]. If the user of the system can provide proof in some or all of these areas, he/she is admitted to the system. To protect the user and the communication between the user and the system, there are many security software solutions available. However, even using the very best software, which implements the most advanced technology and the most secure algorithms, cannot guarantee 100% security because the end users are humans and humans are gullible in understanding security concepts.

The human factor is the underlying reason why many attacks on computers and systems are successful. The human factor is often described as the weakest part of a security system and users are often described as the weakest link in the security chain. It has been noticed that attackers always try to exploit the weakest link. For example, researchers have shown how attackers can exploit human activity on public forums and online social networking websites to mine personal attributes (e.g. age, sex, sexual orientation) and sensitive information (e.g. answers to challenge questions such as mother's maiden name). Another human naiveness in understand-

ing security concepts is in dealing with passwords for different accounts. Having so many accounts, humans can no longer remember all the passwords resulting in duplication of passwords. Sometimes they either use simpler passwords which are not good enough to reliably defend against dictionary attacks or use stronger passwords which are too complicated to be remembered and write them down on a piece of paper. Though not the focus of the thesis, but to demonstrate a few more human factor exploitation in user authentication; attackers do not target on-line banking directly. Instead, they attack the bank's customers, using phishing techniques to trick them into giving away their credentials. Although widespread deployment of the Secure Sockets Layer (SSL) helps protect password authentication against passive eavesdropping attacks, it does little to help users resist more devious threats, such as phishing. Alternatively, an attacker can call the IT help desk, pretend to be a senior manager and gain access to confidential information. This is social engineering - exploiting human vulnerabilities rather than technical ones.

From this dissertation we would like to highlight that any information or an entity which can be used to exploit the vulnerability of a system is a valuable resource to the attacker. This information can be used to harm the user; being it side channel information, inferred information or the user himself/herself. Specifically, in this dissertation we formulate and propose solutions to two authentication problems relating to human factors. a) Secrecy information inference attack – where publicly available information can be used to infer secrecy information about the user. b) Coercion attack – where an attacker forces a user to handover his/her secret information such as account details and password.

Using personal and private information in generating challenges for authentication systems has been there for a long time e.g. in backup authentication (mother's maiden name) when the user forgets the login details of the primary account. In a backup authentication, a user's responsibility of authentication is shifted to another entity, usually in case of failure of the primary authentication method. In recent years, online social networking activity has increased a lot and because of the avail-

able platforms like Facebook more and more people are sharing private information online. People show their support to a number of things in public e.g. by clicking on the Facebook 'like' button. In this work, we first show how an attacker can exploit the public data extracted from Facebook to infer users' undisclosed interests on their personal profiles. We also show how this inferred information can be used to break a preference based backup authentication system and demonstrate why we should not use weak personal information e.g. "interests" in the generation of challenges. To resist the secret information inference attack, we design a system called HuMan (History based User Centric Memorable ApplicatioN). HuMan generates memorable and dynamic fingerprints from the user's cellphone usage data which can be remembered by the user and can be posed as a challenge during authentication.

However, all security mechanisms fails when the user is succumb to coercion attacks i.e. putting a gun on the user's head and coercing him/her e.g. to enter his/her bank account details. The user has no choice but to comply and reveal his/her secret. This is an extreme form of human factor exploitation. For that we propose to build a coercion resistant system. For a system to be coercion attack resistant, it is required that when the user is under coercion, he/she will have no way of generating the secret, or the secret generated will never be the same as the one generated when he/she is not being coerced. If this requirement is met, then an adversary would not apply any threat to him/her because the adversary understands that the user would not be able to generate the secret when he is threatened to do so. We demonstrate this attack under two scenarios (when the user is forced to reveal his own secret and when the user is forced to reveal someone else's secret) by conducting two separate user studies and hence propose to use emotional response (skin conductance in our case) as a parameter to fight against such attacks.

In the following sections, we individually highlight the attacks and demonstrate through various user studies which evince how different resources can be obtained to exploit human factors in user authentication. We also demonstrate the human factors which should not be used in creating authentication challenges.

1.1 Fighting against secrecy information inference attack in user authentication

In a secrecy information inference attack, user's secret information can be leaked from the publicly available data. Prior research shows that the information shared on the web with a limited set of users can still leak undisclosed privacy attributes, e.g., users' interests and even sexual orientation [149, 115]. Authors of [29] crawled Facebook users' personal profiles (which are often private) to infer users' undisclosed interests. Private data can only be obtained by either crawling user's personal profile on social networking sites or taking explicit permission from the user. However, getting access to the private data is not as easy as compared to the public data. Facebook, for example, has made all the fan pages public by default [44]. Access to the data of these pages can be conveniently obtained through Graph APIs [45]. Mining private information from public data is not easy mainly due to the large amount of noise contained in the heterogeneous pages, and the huge amount of unstructured data involved. In this work, we first demonstrate that this belief might not be true in certain aspect. In particular, we show how we can obtain data from Facebook and use it to infer users' interests that can usually be obtained only from their personal and often private profile pages. This information can be used in many ways including targeted spamming, showing ads without the consent of users, or even breaking into specific authentication systems. To demonstrate the security and privacy implication of this, we base our experiments on mining personal interests to break into Blue Moon^{TM} [81] introduced by RavenWhite as a backup authentication system to provide better security and usability. Our experiments conducted on over more than 34K public pages collected from Facebook and data from volunteers show that our inference technique can infer interests that are often hidden by users on their personal profile with moderate accuracy. We are able to disclose 22 interests of a user and find more than 80,097 users with at least 2 interests. From our findings, it is clear that we should be prudent in choosing the information to create

authentication challenges because attackers can use the publicly available interests data from Facebook to break into authentication systems like Blue Moon™.

Considering this, we propose to use an authentication system based on personal data which is resistant to secrecy information inference attack. The authentication challenges are generated dynamically and the user can still remember without requiring any extra effort. We built a system, called HuMan, which generates fingerprints from user's cellphone usage data. We explore if the generated behavioral fingerprints are memorable enough to be remembered by end users. The dynamicity and memorability of these fingerprints can also eradicate human factors like human memory interference, sharing of secrets etc. We evaluated the memorable fingerprints generated from this rich multi-context data by asking each user to answer various authentication questions generated from the fingerprints. We conducted an extensive user study that involved collecting about one month of continuous usage data from 58 Symbian and Android smartphone users. Results show that the fingerprints generated by HuMan are remembered by the user to some extent and that they were moderately secure against attacks even by the people who knows a lot of information about the user i.e. intimates and acquaintances.

1.2 Fighting against coercion attack in user authentication

Many techniques have been proposed for secure communication and authentication. Some of these techniques, e.g., those using biometrics [58, 116, 119, 120, 53], offer desirable security properties including ease of use, unforgettability, unforgeability (to some extent), high entropy and etc. However, most of these schemes are not resistant to coercion attacks in which the adversary uses physical force, e.g., wielding a gun, to coerce the trustee to comply [130]. When the user's life is threatened by an attacker, one would have to surrender the secret, and the system will be com-

promised despite all the security properties described above. This is an extreme form of exploitation of human factor in user authentication to gain access to the system. Specifically, we present a novel approach to protection against rubber hose cryptanalysis i.e. coercion attacks in generating cryptographic keys. For a cryptographic key generation technique to be coercion attack resistant, it is required that when the user is under coercion, he/she will have no way of generating the key, or the key generated will never be the same as the one generated when he/she is not being coerced. If this requirement is met, then an adversary would not apply any threat to him/her because the adversary understands that the user would not be able to generate the key when he is threatened to do so. We explore the incorporation of user's emotional status (through the measure of skin conductance) into the process of key generation to achieve coercion resistance. With 39 participants in our user study, we find that our technique enjoys moderate false acceptance rate of 3.2% and false rejection rate of 2.2% in key generation.

Furthermore, to meet the demand of scalability and usability, many real-world authentication systems have adopted the idea of responsibility shifting, explicitly or implicitly, where a user's responsibility of authentication is shifted to another entity, usually in case of failure of the primary authentication method. One example of explicit responsibility shifting is in the fourth-factor authentication whereby a user gets the crucial authentication assistance from a helper who takes over the responsibility [23]. Facebook also uses a similar authentication protocol which allows the user to recover his account's password by collecting vouch codes from his trusted friends [46]. There is also implicit responsibility shifting which might not seem as obvious. For instance, whenever suspicious activity is detected in a user account, the system administrator takes over the responsibility of revoking the attempted authentication. In the fourth-factor authentication system [23], subverting the helper allows the adversary to log in without capturing the password of the user. When the trustee to whom the responsibility has shifted is another computer system, we can use any standard security mechanism to protect it. However, when such a trustee

is a human being, protection becomes non-trivial because of the potential coercion attacks. We remark that it is unclear whether the same technique could help in protecting the trustee in our study. The difference between the trustee and a victim in general coercion attacks is subtle, yet critical in terms of security. No prior study has shown the effect on emotional status of trustee in this case and his skin conductance. Therefore, the crux of our work is to investigate whether the trustee's skin conductance also changes under coercion, and if any, whether the magnitude of change is large enough to be captured by the coercion resistance technique. We design and conduct a user study involving 29 university students to evaluate the trustee's emotional status in a simulated coercion attack. The results of our user study are positive with false acceptance rate of 3.1% and false rejection rate of 1.7%. This shows that the victim's skin conductance still changes under physical threats. The principles of our findings in this study are applicable to other authentication mechanisms as well.

The rest of this dissertation is organized as follows: Chapter 2 reviews the existing studies on authentication and human factors which can be exploited to gain illegitimate access to the system. In Chapter 3, we demonstrate a secrecy information inference attack on a preference based authentication system using the public data extracted from Facebook. In Chapter 4, we argue how we can use the private data from the user's cellphone to create authentication challenges which are memorable and resistant to secrecy information inference attack. We then present an extreme case of human factor exploitation i.e. coercion attack in Chapter 5 and propose a solution to fight against this attack in generating cryptographic key. In Chapter 6, we extend our work to verify if the solution presented in the previous chapter can be used to fight against coercion attack in authentication responsibility shifting. Finally, we conclude with future direction of the current research in Chapter 7.

Chapter 2

Literature review

Authentication has been studied by cryptographers, security engineers, human-computer interface designers, linguists, ethnographers, and others. This chapter will survey the diverse academic literature with particular focus on the security research motivating this dissertation. As every authentication mechanism requires some involvement of humans and humans are considered to be the weakest link in the security chain, therefore, in this chapter we discuss some of the prior works done in the area of human factors in authentication. Related work specifically to this dissertation has been described in individual chapters.

Memory interference and limits Passwords are by far the most used and most easily subverted method of personal authentication. The use of secret words to authenticate humans has ancient origins. It also appears in folklore, famously in the tale of Ali Baba and the forty thieves (first translated into English in 1785 [148]), with the protagonist using the phrase "open sesame" to unseal a magical cave. Ominously, Ali Baba's greedy older brother Qasim forgets this password during the course of the story with disastrous consequences.

If an organization institutes policies to ensure secure passwords (such as frequently changed alphanumeric upper/lower case combination of at least 10 characters) the inconvenience is so great that such a policy will be violated in an overwhelming number of cases. The use of alphanumeric usernames and passwords

is the most often used (and also the cheapest) method of computer authentication [102]. However, unfortunately human beings are limited in their information processing capabilities [33, 114]. People either use simple passwords that are easy to remember but easy to crack or use difficult passwords which are difficult to remember. According to [168], there are very few people who do not deviate from the best practices for password use. Users either use the same password all the time, or use relatively simple passwords; re-use their old password; write passwords down either on paper or store it in an electronic file without protecting it; share passwords with others, etc.

Many recommendation and techniques including using pictures instead of passwords [38, 35, 89], passface [25], pass-phrase [143] etc have been suggested in the past by taking into consideration of user's knowledge [2, 138, 173]. These schemes suffer from the same problem of memory interference; scalability is a major issue. For schemes like passphrases, usability studies of passphrases [97] have found them to be just as memorable as passwords, subject to an increased rate of typographical errors. Users may find it difficult to remember so many different pass-phrases for different accounts. Moreover, systems need to manage a database of a huge number of images, so that they can prevent guessing and DDOS attacks.

Sharing of secrets The argument — "if you don't have anything to hide you won't mind sharing passwords" is the chief weapon in the arsenal of the password sharers. We are always told not to share our passwords or bank account PINs with others, but the rule is harder to apply when it's your significant other who wants to check those party pictures in your Facebook account [37]. In a recent study [107], authors found that roughly one in three online teens (30%) reports sharing one of their passwords with a friend, boyfriend, or girlfriend. While passwords may be guarded closely by some youth, password sharing among peers can be a sign of trust and intimacy. Online girls are much more likely than online boys to share passwords with friends and significant others (38% vs. 23%), and older teens ages 14-17 are more likely to

do so than younger ones (36% vs. 17%). Looking more closely at older girls aged 14-17, nearly half (47%) admit to sharing passwords with friends or significant others. It has been found that spouses check their partners accounts without their permission which can lead to divorces later on [9]. This is not a technical problem. Its a social problem, however, the implications are huge.

Insider Attacks Insider attacks go beyond the hacking. Hackers, especially "terrorist hackers" or "cyberwar hackers" get lots of press. They do indeed pose a serious problem. However, the threat they pose pales before that posed by those closest to us: the insiders. It is no secret that companies spend a majority of their security budget on protecting from external attacks but, "one of the toughest and most insidious problems in information security, and indeed in security in general, is that of protecting against attacks from an insider." [17]. An insider can be an employee, or a student or other members of the organization. He/she can be someone duped or coerced by an outsider to perform actions on the other's behalf.

Shoulder surfing or Peeping tom In computer security, shoulder surfing refers to using direct observation techniques, such as looking over someone's shoulder, to get information. It is commonly used to obtain passwords, PINs, security codes, and similar data [103]. Current approaches in the effort of reducing shoulder surfing attacks typically also reduce the usability of the system; often requiring users to use security tokens like RSA security token, interacting with systems that do not provide direct feedback [135, 160] or requiring additional steps to prevent an observer from easily disambiguating the input to determine the password/PIN [25, 64, 135, 157, 160]. Previous gaze-based authentication methods [103, 75, 111] do not support traditional password schemes.

Some of the techniques [135] assume that the adversary is not able to capture the complete interaction between a user and the server. Such an assumption actually forms a secure channel between the user and the server, which transforms the secret

leakage problem to the protection of the secure channel. However, in many cases this may not hold where an adversary can deploy a hidden camera, key logger to capture the whole password entry process. To address such realistic concerns, recent efforts [7, 108, 157, 160, 170, 11, 137] have focused on the adversary model, where the adversary is allowed to record the complete interaction between the user and the server.

Phishing Phishing is a form of social engineering which attempts to acquire information (and sometimes, indirectly, money) such as usernames, passwords, and credit card details by masquerading as a trustworthy entity in an electronic communication. It was for a long time commonly held among security practitioners that the widespread deployment of SSL would eliminate phishing once consumers become aware of the risks and nature of phishing attacks. This, very clearly, has not been the case, as supported both by real life observations and by experiments [165]. It is crucial for the security practitioners and service providers to understant what consumers think and want. The lack of security knowledge [159] and education is typical among users. A recent study shows how computer users fall victims to phishing attacks based on a lack of understanding of how computer systems work, due to a lack of attention, and because of visual deception practiced by the phishers [39]. There has been numerous studies to know what a typical user reacts to when they browse to see their emails [87, 161, 77]. Previous studies have examined the extent to which users fall for phishing scams and whether users benefit from the information provided by anti-phishing tools. These studies have shown that most users are likely to fall for phishing scams, and that many users ignore warnings provided by anti-phishing tools [39, 41, 84, 165].

Physical security A recent attack by a tech journalist exposes vital security flaws in several customer service systems, most notably Apple and Amazon [76]. A security guard on a door may not increase the security of the whole system. It may

decrease the security of the system. A person can fool the security guard by producing a false official letter stating that he/she is a legitimate user of the system and he/she has an authority to enter the building. An attacker can put a gun on the security guard's head to force him/her to gain access to the doors [69]. There has been solutions to circumvent coercion attacks by using panic passwords [31]. In this thesis, we demonstrate the use of skin conductance to fight against coercion attacks.

Online social networking With the rise of online social networking in the last decade, there is more and more private information being shared with friends and other people. Prior research shows that the information shared with a limited set of users can leak undisclosed privacy attributes, e.g., users' interests and even sexual orientation [149, 115]. Authors of [29] crawled Facebook users' personal profiles to infer users' undisclosed interests. There are high security implications of this as we demonstrate in Chapter 3.

Chapter 3

Secrecy information leakage from public data

3.1 Introduction

With the rise of online social networks (OSNs), more and more private information of users is available on the web. It also forms a fertile ground for a variety of research efforts. The information shared on OSNs can be classified into two broad categories, *private* (shared with a limited set of users) and *public* (shared with the whole world). Prior research shows that the information shared with a limited set of users can leak undisclosed privacy attributes, e.g., users' interests and even sexual orientation [149, 115]. Authors of [29] crawled Facebook users' personal profiles to infer users' undisclosed interests. However, getting access to such information that is shared with a limited set of users is non-trivial as it is not available to public via any APIs. Moreover, OSNs are adopting ways to restrict crawling unless explicit permission is granted [134, 166].

In comparison to the private information available only to a limited set of users, public information is readily available. In many cases APIs are provided by OSNs for anyone to efficiently download such public data. Facebook, for example, has made all the fan pages public by default. Access to the data of these pages can be

conveniently obtained through Graph APIs [45]. It is generally believed that these public pages hardly contain users' private information, and mining private information from them is not easy mainly due to the large amount of noise contained in the heterogeneous pages, and the huge amount of unstructured data involved. For example, Facebook has little control on the titles and descriptions of fan pages; posts from users may contain text and multimedia content; users use a lot of short sentences and slang (e.g., "LOL", "LMAO", etc); off-topic discussions go on frequently (e.g., on a "Jazz" page, we found users discussing the latest soccer game). All this adds to the noise in public data on OSNs. Moreover, almost 15 percent of user-submitted content on large Facebook fan pages is spam [55]. Such noise and the huge amount of unstructured data to be processed usually makes mining interesting private information not practical.

In this work, we show that this belief might not be true in certain aspect. In particular, we show how we use *publicly available* data from Facebook to infer users' interests that are usually only on their personal profile pages. We make use of the graph APIs provided by Facebook to obtain public fan pages [44]. As these pages are public irrespective of the users' privacy settings, an attacker can grab the unique profile IDs of those who have interacted with the page. We show that by aggregating different interests of the users found across different pages, one could build users' interests profiles from the public data without gaining access to the personal profile pages of any of the users. This collective information can be used in many ways including targeted spamming, showing ads without the consent of users, or even breaking into specific authentication systems.

To demonstrate the security and privacy implication of this, we base our experiments on mining personal interests to break into Blue Moon™ [81] introduced by RavenWhite as a backup authentication system to provide better security and usability. From the dataset in our experiments involving 1.1 million different user IDs from 34,000 Facebook public pages, we detected 80,097 (6.89%) users with two or more interests. Out of these 80,097 users, there are 66 who have been found with

more than 8 interests, which is enough to break their corresponding Blue Moon accounts (if they have) with reasonable accuracy under certain assumptions. In one case, we were able to build a user profile with as many as 22 interests by mining the data we collected. We also present valuable lessons we learned in our experiments, among which the most notable one being that users' sentiment orientation might not be inclined towards the sentiment orientation of the page i.e. simply liking a page does not corroborate enough that the user is really interested in the page. Therefore we performed sentiment mining to find out the actual sentiment orientation of the user.

In summary, this work makes the following contributions.

- We use publicly available data on Facebook to infer users' privacy attributes (i.e., interests) and aggregate this information across different pages. This differs from prior research as we do not use user's personal data posted on their profile page (e.g., gender, current location, activities, interests, etc.).
- We find that liking a page does not corroborate a user's inclination towards a page or interest category. We performed an in-depth analysis (sentiment) using text mining to find the real sentiment orientation or polarity (like or dislike) of the user towards a page and an interest.
- We use Facebook's public Graph API [45] to obtain the public pages. Unlike crawling which is usually restricted in its usage by OSNs to a small number of partners, our method could be easily used by anyone with little restriction.
- We demonstrate the severe implication of this private information mining by showing that interests inferred from the public data can be used to exploit a previously proposed preference based authentication system. This suggests that one should be mindful in designing the challenges for the authentication system, in this case, information such as interests should not be used in creating authentication challenges as this information can easily be obtained in the era of Facebook.

The rest of the Chapter is structured as follows. We provide background and related work in Section 3.2 where we show some of the important prior work to abuse OSNs. We explain our technique to mine user interests from Facebook public pages in Section 3.3, and report experimental results in Section 3.4. We then discuss the limitations of our technique in Section 3.5.

3.2 Related work

In this section, we first discuss related work in obtaining a user's private information by abusing OSNs in general. After that, we discuss the more specific interests inference techniques in social networks.

3.2.1 Abusing OSN data

With the increasing popularity of OSNs, people start to find ways of abusing it, e.g., illegitimate use by spammers with ad deals. In this Chapter, we focus on the abuse in which a user's privacy attributes are inferred from information hosted on OSNs. In general, attackers could base their attacks on two types of data obtained in two ways.

One is to use restricted pages by crawling. Prior research shows that information on restricted pages (shared with a limited set of users) can leak undisclosed privacy attributes about the users [149, 115]. Existing techniques have demonstrated that private information can be crawled to obtain attributes like mother's maiden name, date of birth, hometown, first school attended to break into backup authentication mechanisms that are based on such privacy attributes [83]. Attackers can also correlate information from different OSNs to retrieve undisclosed attributes of the users [115]. Authors of [29] crawled users' personal profiles of Facebook to infer their undisclosed interests. [14] describes how an attacker could query popular social networks for registered e-mail addresses on a large scale and information from different social networks can be aggregated to launch sophisticated and targeted

attacks.

An important limitation to using restricted pages by crawling is that most OSNs restrict crawling to a small number of partners only. That is, crawling restricted pages is not a technique available to general attackers.

The other type of data to use is public pages. As compared to crawling restricted pages on OSNs, anyone can use a legitimate channel (usually by using public APIs provided by OSNs) to gather public pages. Although these public pages are more readily available for anyone to analyze, as pointed out in Section 3.1, it is generally believed that mining interesting private information from these public pages is difficult due to the noise in it and the huge amount of unstructured data to be analyzed. In this work, we show that mining users' otherwise undisclosed interests from public pages on OSNs is, in fact, practical.

There are strong security and privacy implications to such abuse of OSN data because the private information mined could potentially be used to break existing personal authentication systems, typically those that use challenge questions as a backup to the main authentication mechanism. In Table 3.1, we highlight noticeable differences between this work and prior research. Previous work has shown that OSN data and public databases can be used to infer or guess sensitive information about users [149, 115]. A number of incidents, e.g., in 2008 the Republican vice presidential nominee Sarah Palin's email account was compromised by an attacker who guessed her personal authentication question (where did you meet your spouse?) [24], in 2009 a vandal successfully guessed a Twitter executive's password and leaked the company's internal documents [34], have shown the severe damage such attacks could have. Personal authentication questions are usually a weaker link in authentication systems [131]. [131] shows that answers to predefined questions can be easily guessed or obtained from OSNs. Instead of specific attack incidents where one or two particular accounts are compromised, our work presented in this work shows an attack to the authentication system and evaluates the extent to which thousands of users of such a system could be attacked.

#	Work	Datasets source	Dataset type	Count	Inferred information	Collected information	Dataset gathering technique
1	Mislove et al. [115]	Rice University Facebook network	Personal profiles	6,156	Missing attribute on personal profile page	Attributes of others (friends etc.)	Crawling
		New Orleans Facebook network	Personal profiles	90,269			Crawling
2	Chaabane et al. [29]	Facebook	Personal profiles	104,000	Country, Age, Sex & Relationship status	Interests	Crawling
		Volunteers from Facebook	Personal profiles	200			Voluntarily provided
3	Lindamood et al. [109]	LiveJournal	Personal profiles	66,766	Nodes in the social graph	Friendship relations	Crawling
4	Goga et al. [63]	Twitter	Personal profiles	93,839	Correlating missing attributes	Profile and privacy information	Friend-finder
		Flicker		59,476			
		Yelp		24,176			
5	Avello et al. [61]	Twitter	Personal Profiles	4.98M	Sex, age, political theory, race	Personal details & tweets	Crawling
			Tweets	27.9M			
6	Zheleva et al. [171]	Flicker	Personal Profiles	9,179	Location	Information on profile	Crawling
		Facebook	Personal Profiles	1,598	Sex, political theory		Crawling
		Dogster	Dog Profiles	2,632	Breed category		Crawling
		BibSonomy	Personal Profiles	31,715	Spammer		N/A
7	*Our Work*	Facebook	*Public pages*	34,000	Interests	Public page's data	*API*
			Personal profiles	1.1M			

Table 3.1: Comparison with the related work

3.2.2 Interests mining from OSN data

This work focuses on personal interests mining because personal interest is one of the most popular choices used in challenge questions. Authors of [125] leverage on the friendship network to mine users' interests. [60] employs feature engineering to generate hand-crafted meta-descriptors as fingerprints for a user. However, such models alone may not derive the complete interest list of any user [169]. [101, 30] resort to collaborative filtering techniques to profile user interests by collaboratively uncovering user behaviors.

The "Like" function on OSN provides a more intuitive way of estimating user interests as compared to non-direct indicators such as user-service interactions. Clicking on the "Like"/"Dislike" button associated with an object usually indicates that (s)he is highly interested/disinterested in the object [169]. Recent approaches like LinkMiner [90] assumes that clicking the "like" button demonstrates the user's liking towards the object. In this Chapter, we show complications in using such an assumption on large datasets and propose solutions to it.

3.3 Interests inference

Although users understand that public pages are for everyone to view and should not contain sensitive or private information, these pages nevertheless reveal what users do and what users think. Therefore, it is probably not difficult to be convinced that such pages still contain private information, probably indirectly and to a limited degree, e.g., by reflecting what users like and dislike. This wok is not to argue this, but rather to investigate how practical it is to mine interesting personal interests from the large amount of unstructured data on public pages that contain a lot of noise.

To do this, we first introduce the data source on which our analysis is performed, i.e., the public pages on Facebook (see Section 3.3.1). Section 3.3.2 presents our methodology to fetch information from these public pages. Finally, we present our methodology to infer users' interests (i.e. likes and dislikes) for different categories

(e.g., music, cars, sports) in Section 3.3.3.

3.3.1 Facebook page layout

A Facebook Page is a public profile where users can talk (and comment/like) about a particular topic. As shown in Figure 3.1, it usually contains many attributes including title, page description, profile picture, wall posts, likes, etc. Any registered user can create a page and by default all Facebook pages are public. Please note that anyone can view the page, however, to interact with a page, a registered Facebook user must "like" it first by pressing the like button on the page. After liking the page, the user can post a message/link/photo/video which will appear on the wall of that page. Other users who have already liked that page can post comments on a post, like the post, etc.

Figure 3.1: A public Facebook Page

As an illustration, Figure 3.1 shows a Facebook page "LSU Football" with one post "LSU Tigers in the NFL – Week 10" from the user "LSU Football". 244 users

have liked that post; a user has commented on it and 4 users have liked it.

As shown in Figure 3.2, in general, a public page P is a collection of many attributes. First, there is a title and description t. Each page P may contain a number of posts p^1, p^2, etc. Each post p^i may have a few likes L_1^i, L_2^i, etc. and a few comments c_1^i, c_2^i, etc. Each comment c_j^i might have a few likes $l_{j,1}^i$, $l_{j,2}^i$, etc.

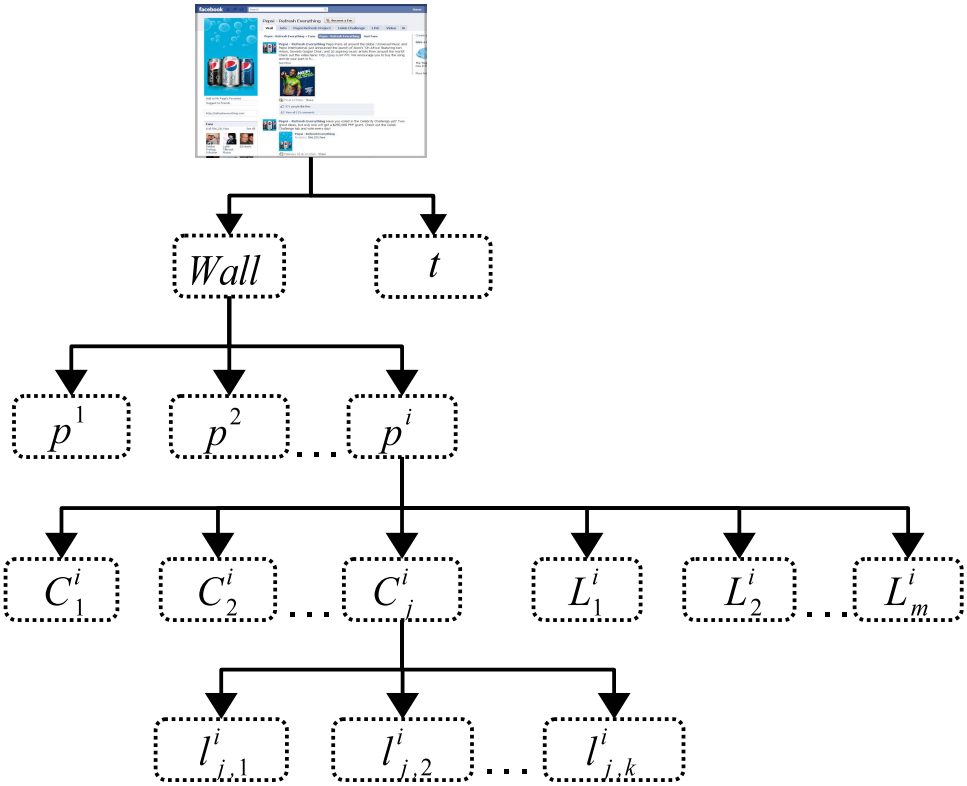


Figure 3.2: Structure of a Facebook page

3.3.2 Data collection

As discussed in Section 3.2, since we make use of public pages only, we can use Facebook's public Graph API to fetch data of any Facebook public page. Information that we manage to fetch for each page include its title and description t, all posts p^i, comments c_j^i and likes L_m^i of each post, as well as likes of each comment $l_{j,k}^i$.

A small difficulty we faced was the authentication needed to use the Graph API. To fetch pages from Facebook using Graph API, one requires an authentication code. This authentication code is generally provided to Facebook applications for

a limited amount of time. However, we did not build any Facebook application to obtain this authentication code. Instead, we created a php script to automatically login to Facebook and then parse the webpage returned at the following URL and look for the authentication code `https://developers.facebook.com/docs/reference/api/`.

There are limitations in using the public APIs only. For example, we were not able to obtain the IDs of those people who only liked the page and did not comment or posted anything on the page. We were also limited by the number of API calls we can make in a certain duration. At the time when the experiments were conducted, Facebook used to provide the list of users who liked a page, this feature is not supported anymore.

3.3.3 Automated profiling with attributes

In this subsection, we first present how we analyze an individual attribute on a Facebook page to figure out if a user has personal interests in the topic covered in that page. This might sound simple, as the user's interaction with the attributes of a page reveals some inclination towards that page. For example, if a user posts a positive message on the wall of a soccer page, it can be inferred that the user may be interested in soccer. For this we propose a technique SPM (**S**im**P**le **M**ining) to mine the information of users' interest (see Section 3.3.3). However, we also observed during our analysis that many users may "like" a page even though they are not interested in the corresponding topic, or if they have strong negative opinions on the topic. To solve this problem, in Section 3.3.3 we propose a more advanced technique called SOM (**S**entiment **O**riented **M**ining) to use sentiment analysis of the attributes to find the actual sentiment orientation of the users.

SPM: SimPle Mining

In SPM, we simply assume that a user's involvement in any of the attributes of P in whichever way indicates that the user has a same interest on the topic of the page, which can be inferred from t. For example, if a user likes a comment or adds a comment on a post on P, then we believe that the user is interested in P. If multiple users have interacted with P, we add all these users into the set $u_q^=$ which denotes the set of users who are interested in P about the interest q.

SPM is simple, but can easily introduce errors to $u_q^=$ because there could be a group of users of P (denoted $u_q^{\neq}$) who hold an opinion opposite to the focus of P. For example, a user who liked the cats page posted the following posts "I hate cats", "Lewis is a mad cat", "go doggies cats are crap". They "liked" the page not because they really like it, but simply because Facebook does not allow them to add a post until they "like" it. To minimize this noise, we perform sentiment mining to find out the inclination of all users towards that page. See the following section for details.

SOM: Sentiment Oriented Mining

Sentiment analysis is the task of identifying positive and negative opinions, emotions, and evaluations [162]. Sentiment analysis has been used in many fields where users have subjective agenda such as movie reviews [126]. Intuitively, the content of the posts/comments should be accounted in deriving the users' interest. Hence, the polarity of the sentiment information of the text aids in conforming the users' interest.

We first define two sets of attributes on a Facebook page P. $\mathcal{A}_{\mathcal{I}}$ = {t, p, c } is the set of independent attributes which consists of text, while $\mathcal{A}_{\mathcal{D}}$ = {L, l } is the set of dependent attributes which does not contain text. We separate these attributes into two groups because those that consist of text can go through a more thorough sentiment analysis on the text, while attributes of the other set are more dependent on the post or comment upon which the "like" was applied.

Sentiment analysis on $\mathcal{A}_{\mathcal{I}}$ We propose to use lexicon approach [80], which is one of the most popular methods used in sentiment analysis to detect the opinion bearing words. Lexicon approach concerns the use of lexical resources such as a dictionary of opinionated terms or opinion words. Collectively, they are called the opinion lexicon and are instrumental for sentiment analysis. Opinion words are the words that are commonly used to express positive (◐) or negative (◑) sentiments. For example: 'beautiful', 'wonderful', 'good', and 'amazing' are positive opinion words, and 'bad', 'poor', and 'terrible' are negative opinion words. Many opinion words are adjectives and adverbs. Sometimes, nouns such as 'rubbish', 'junk', and 'crap' and verbs such as 'hate' and 'like' also indicate opinions. Words which are neither positive nor negative are marked as neutral (⊖).

Several opinion lexicons are available and SentiWordNet [10] is one such resource containing opinion information on terms extracted from the WordNet database and made publicly available for research purposes. SentiWordNet is a lexical resource built on top of WordNet. WordNet [48] is a thesaurus containing descriptions of terms, and relationships between terms and part-of-speech (POS) types. For example "car" is a subtype of vehicle and car has same concept as automobile. Hence, a synset (a synonym set) in WordNet comprises of all the terms with the same concept, e.g., the synset is car, automobile.

SentiWordNet assigns three sentiment scores to each synset of WordNet: positivity, negativity, objectivity/neutral. The sentiscores are in the range of [0, 1] and sum up to 1 for each triplet. For example, in SentiWordNet, the sentiscore of the term "good" is (pos, neg, obj) = (0.875, 0.0, 0.125). For our experiments, the scores are approximated with labels/part-of-speech of term in the text or sentence. First, the text is tagged using a standard POS tagger. A standard POS Tagger [147] is a piece of software that reads text in some language and assigns parts of speech to each word, such as noun, verb, adjective, etc. Then the SentiWordNet is used to get the scores for each term in the text. As our sentiment analysis is domain independent, we choose the general lexicon method as compared to the corpus-based

method which is a domain dependent approach.

We now explain in detail how the sentiment is derived for the attributes in $\mathcal{A}_{\mathcal{I}}$. We call them independent attributes because their sentiment orientation is independent of the other attributes. For a particular attribute $a \in \mathcal{A}_{\mathcal{I}}$, we count the total number of words/phrases with positive sentiment (p_s) and that of words/phrases with negative sentiment (n_s) for all posts and comments he/she has, and use the term-counting method proposed by [151] to determine the sentiment orientation of a to be

$$\Psi(a) = \begin{cases} \CIRCLE, & \text{if } p_s > n_s \\ \CIRCLE, & \text{if } p_s < n_s \\ \ominus, & \text{otherwise} \end{cases}$$

Sentiment analysis of $\mathcal{A}_{\mathcal{D}}$ Attributes in $\mathcal{A}_{\mathcal{D}}$ does not contain text, but they also contribute to a user's sentiment orientation. We call them dependent attributes because their sentiment orientation is dependent on other attributes. For example, if a user u_1 has a post p^i with negative opinion, and user u_2 likes that post L^i_m, then both u_1 and u_2 share negative sentiment orientation on the topic. Similarly, if a user u_1 has a comment c^i_j with positive opinion, and user u_2 likes that comment $l^i_{j,k}$, then both u_1 and u_2 share positive sentiment orientation on the topic. That is,

$$\Psi\left(L^i_m\right) = \Psi\left(p^i\right)$$

and

$$\Psi\left(l^i_{j,k}\right) = \Psi\left(c^i_j\right)$$

Aggregating interests profiling from multiple attributes on multiple pages A user might have multiple posts, comments, and likes on a single Facebook page, and multiple Facebook pages might be about the same interest. Therefore, we have to aggregate the sentiment analysis results on multiple attributes from multiple pages

in order to figure out the sentiment orientation of the user on that interest.

Let $A = \{a_1, a_2, ..., a_k\}$ be the set of posts, comments, and likes of a user u on a page P about a particular interest q. For each $a \in A$, we compute the sentiment orientation. Then, the sentiment orientation of u towards P, S_P^u is ◐/◑ if the number of attributes with positive sentiment is greater/lesser than the number of attributes with negative sentiment respectively, otherwise $\ominus$. Aggregating all Facebook pages about q, sentiment orientation of u towards q, S_q^u is ◐/◑ if the number of pages with positive sentiment orientation is greater/lesser than the number of pages with the negative sentiment orientation respectively; otherwise $\ominus$. If the sentiment orientation of q and S_q^u is same, then, u is added to the set $u_q^=$ otherwise to $u_q^{\neq}$. That is,

$$u_q^= = \left\{ u \in U \,\middle|\, \left(\left(\Psi(q) = \text{◐} \,\&\&\, S_q^u \in \{\text{◐}, \ominus\} \right) \| \left(\Psi(q) = \text{◑} \,\&\&\, S_q^u = \text{◑} \right) \right) \right\}$$

$$u_q^{\neq} = \left\{ u \in U \,\middle|\, \left(\left(\Psi(q) = \text{◐} \,\&\&\, S_q^u = \text{◑} \right) \| \left(\Psi(q) = \text{◑} \,\&\&\, S_q^u \in \{\text{◐}, \ominus\} \right) \right) \right\}$$

3.4 Experimental results

To base our analysis on a concrete example, we focus on breaking Blue Moon™ [81], a backup authentication system which can be used by a user to reset his lost or forgotten credentials. For example, if a user forgets his password of an email account, he or she can use Blue Moon to reset the password. The idea is to use personal preferences as challenge questions for authentication. Figure 3.3 shows a screenshot of Blue Moon[1].

During enrollment, the user is asked to select 8 items which he likes and 8 items

[1]This image is taken from `http://www.ravenwhite.com/iforgotmypassword.html`.

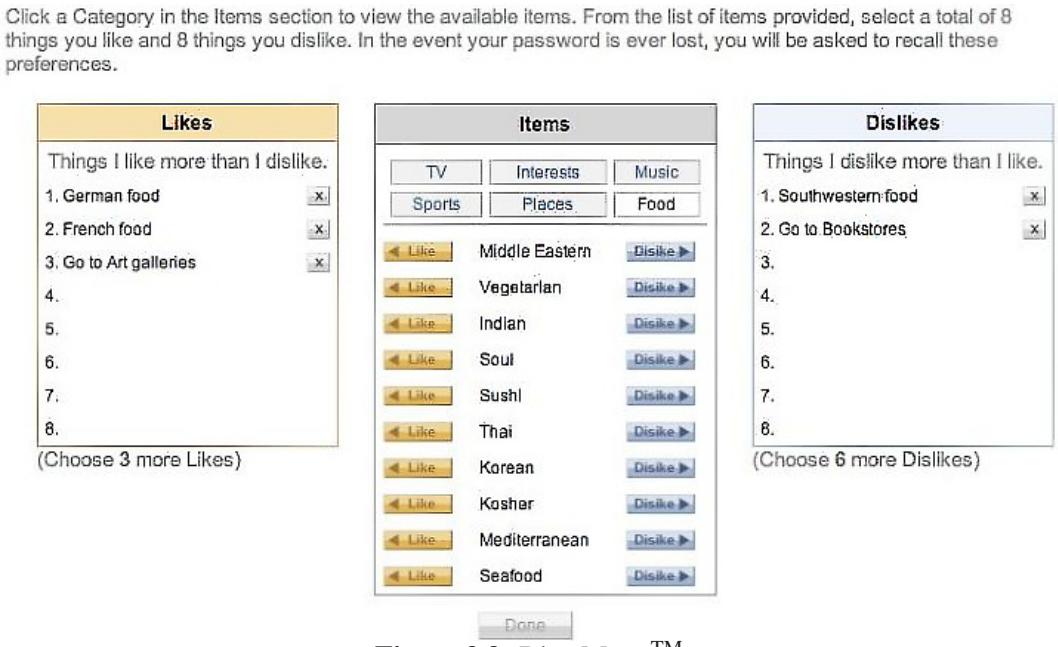


Figure 3.3: Blue Moon^{TM}

which he dislikes from a list of 76 common interests. During authentication, the user is presented with a set containing the chosen items in a randomized fashion. The user categorizes the items to like and dislike. A user is not required to pick all the interests correctly. Instead, the user just need to correctly categorize 8 items[2] to reset his password [86]. To make our analysis consistent and the evaluation comparable, in the rest of the chapter we assume that a user has to correctly categorize 8 items from the entire list of 76 interests which are shown in Table 3.2.

3.4.1 Dataset description

In order to attack the Blue Moon system, we assume a strategy taken by an attacker as follows. He first construct a set of interests Q from Table 3.2, and another set $Q^{'}$ containing the corresponding negated items like "I hate golf" and "I hate jazz". He then leverages the Facebook's public Graph API to 1) find all public Facebook pages related to $q \in \{Q \cup Q^{'}\}$; 2) fetch all attribute data of these pages; and 3) use the technique described in Section 3.3 to find $u_q^{=}$ and $u_q^{\neq}$ for all $q \in \{Q \cup Q^{'}\}$.

Note that this methodology does not cover all those pages which are semanti-

[2]The threshold where false acceptance rate and false rejection rate meets.

Sports	Music	Places	Interests	TV	Food
Aerobics	Instrumental	Garage sales	Cars	Watching Extreme sports	Soul food
Billiards	Symphony	Bookstores	Crafts	Documentaries	Indian food
Racing	Folk	Political events	Creative writing	Watching Auto racing	Korean food
Martial arts	Easy listening	Art galleries	Casino	Watching News	Kosher food
Baseball	Gospel	Raves	Painting	Watching Figure skating	Middle Eastern food
Soccer	Electronics	Antique stores	Gardening	Watching Diving	Southwestern food
Golf	Classical	Museums	Religion	Watching Baseball	French food
Running	Jazz	Flea markets	Politics	Watching Hockey	Seafood food
Yoga	Big Band	Libraries	Poetry	Watching Golf	Thai food
Skating	Reggae	The opera	Gaming	Game shows	Vegetarian food
Cycling	Show tunes	Politics	Reading comics	Watching Soccer	German food
Hockey	Heavy Metal		Cats	Watching Bowling	Mediterranean food
Pool			Gambling		
Motocross					
Basketball					
Football					

Table 3.2: User interests domain

cally related with the query term. For example, query term "cars" does not fetch pages of Mercedes, Hyundai or Porche which are indeed the pages of cars. [29] provides a solution to fetch these pages using semantic search with the help of an ontology build upon Wikipedia. We are sure that this could significantly increase accuracy of the attack, although that comes with a price of longer processing of a large set of pages. We leave this as a future work to increase the size of the corpus.

Table 3.3 summarizes the availability of the attributes with their counts. From 34K pages fetched for 152 categories we found 2.5 million posts, 7.5 million likes and 4.3 million comments on these posts, and 1.3 million likes on those comments.

Attribute	Count
Categories	152
Pages	34,738
Posts	2,538,987
Post likes	7,574,965
Comments	4,381,967
Comment Likes	1,361,361

Table 3.3: Number of attributes found

We apply both SPM and SOM mining techniques as discussed in Section 3.3.3 and 3.3.3 respectively to the dataset `PubProf`, and discuss the results in the next subsections.

3.4.2 Inferred interests using SPM

We found a total of $1,162,575$ unique users whose interests can be inferred from the pages analyzed. These users are the users who have either posted something on the pages, commented on the posts, or liked the posts/comments. Applying the SPM approach to our dataset, we detected $80,097$ users with 2 or more interests. This amounts to 6.89% of all the user IDs collected in our dataset. Figure 3.4 shows breakdown of users with different number of interests found. We were able to build a user profile with as many as 22 interests.

Note that although SPM might not be accurate in finding the true sentiment orientation of the user over an interest, the numbers presented here is not affected

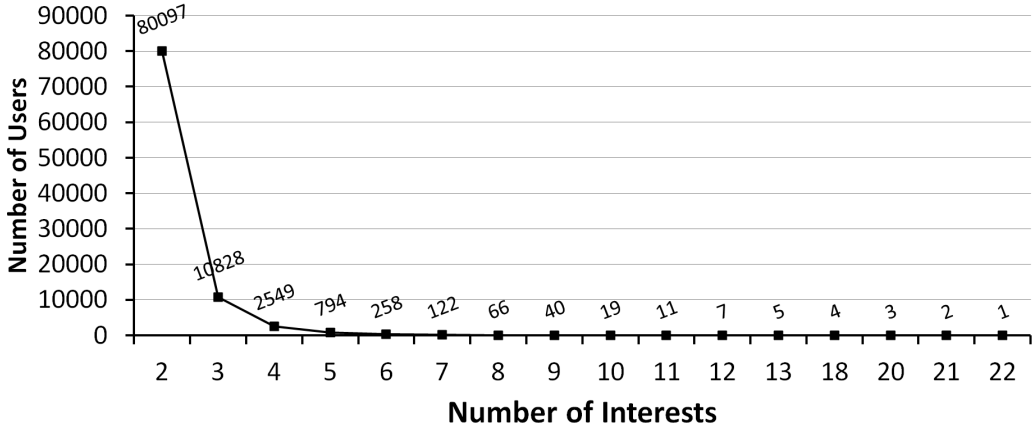


Figure 3.4: Inferred interests of the users using SPM

by this inaccuracy. That is, results presented in Figure 3.4 actually applies to SOM as well.

Results show that the number of users whose interests could be inferred from the public pages is significant, and this would have an important impact on the possibility of breaking into such users' Blue Moon account. For example, for those users found to have two interests, the search space for breaking into their Blue Moon account is reduced by about a factor of 3,000 ($^{76}C_2$). Please also note the dataset we use represents a tiny subset of the Facebook pages.

3.4.3 Inferred interests using SOM

In this section, we first investigate the inaccuracies when applying SPM on our data. As discussed in Section 3.3.3, these inaccuracies happen when users like a page but oppose to the topic in it. For example, if the page's title is "I hate Cats", we want to find those users who clicked on the like button on this page, however, actually like cats. Figure 3.5 shows the result of our sentiment analysis on all the comments, in particular, the number of negative comments different user posts. We can observe that about 10% of the comments posted are negative, meaning that the comment itself does not have the same sentiment orientation as that of the page. This suggests that more careful analysis and handling of the sentiments of posts and comments are important in order to find out the users' interests.

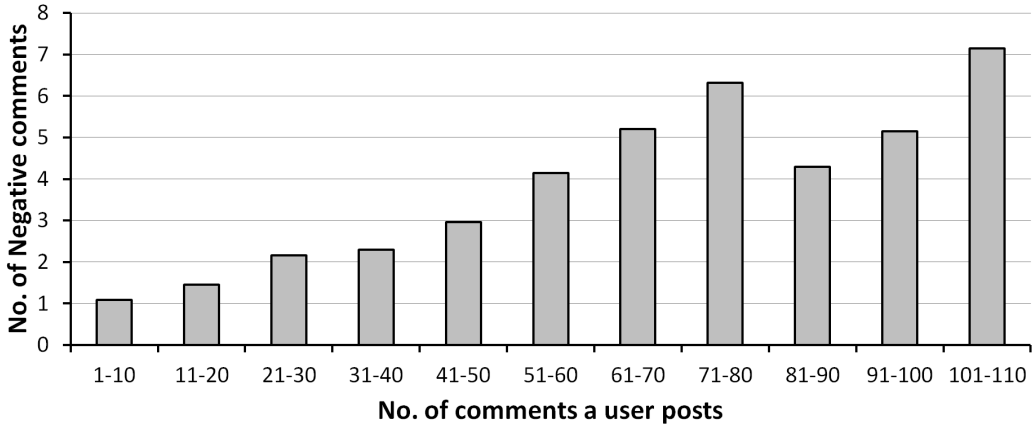


Figure 3.5: No. of negative comments posted by users

One interesting finding we also observe from Figure 3.5 is that users who comment a lot (more than 70 comments) tend to have a smaller percentage of negative comments. It is our future work to investigate whether the same is observed on a larger dataset.

To further investigate the result of our sentiment analysis, Table 3.4 shows 15 interests with the largest percentage of users who disagree with the corresponding topic of the page. $|u_q^=|$ is the total number of users whose sentiment orientation is same as that of the category, and $|u_q^{\neq}|$ is the total number of users whose sentiment orientation is opposite to that of the category. We see that although the inaccuracies from the SPM technique exist, most pages, especially those with a large number of users discussing, tend to have less than 10% of the users with negative sentiment orientation.

Another interesting observation is that most of the entries in Table 3.4 are of pages with a negative sentiment, i.e., q is "Hate xxx". We believe that it is because there are more people who want to voice out their disagreement with such pages than people who disagree with pages with a positive sentiment.

3.4.4 Comparing SPM and SOM

Table 3.5 shows the number of users found liking/disliking selected categories for both the SPM and SOM techniques. We only show a few categories with the largest

Category (q)	$\lvert u_q^{\neq} \rvert$ (%)	$\lvert u_q^{=} \rvert$ (%)	Total
Hate motocross	25.00	75.00	8
Hate skating	25.00	75.00	4
Hate heavy Metal	14.28	85.72	14
Hate poetry	12.50	87.50	8
Hate hockey	11.53	88.47	26
Watching baseball	11.12	88.88	9
Hate baseball	8.33	91.67	48
Hate basketball	5.52	94.48	181
Hate cats	5.39	94.61	260
Hate religion	5.36	94.64	56
Hate football	3.35	96.65	359
Raves	3.22	96.78	280
Gamble	3.08	96.92	195
Hate cars	3.08	96.92	130
Game shows	2.95	97.05	34

Table 3.4: The percentage of users whose sentiment orientation is inclined / not inclined towards the sentiment orientation of the page across all interests categories.

discrepancies due to space constraint. Note that these are accumulated categories, e.g., we combined all 12 musical categories like jazz, classical, etc.

Results show that although there are users who like the page while having a different sentiment orientation as shown in the previous subsection, these users are minorities, and that is why we do not see a large discrepancy between results from SPM and SOM. In this respect results here seem to be consistent with those presented in Table 3.4

Also, we observe that sports is the most popular category where 12.62% of users are inclined to the sports. These numbers could potentially be used to obtain the a priori probability for an unknown user having different interests, and subsequently used in attacking the Blue Moon system. We leave more detailed analysis on this our future work.

3.4.5 Errors in sentiment analysis

As text mining is prone to errors, in this section we evaluate the correctness of our SOM approach in detecting the correct sentiment orientation. We manually label

Main Category	#Users using SPM	#Users using SOM
Like Sports	146756	145141
Like Music	29597	29255
Like General	65354	64571
Like Entertainment	163	158
Like Food	4031	3990
Do not like Sports	1120	1079
Do not like Music	30	28
Do not like General	825	800

Table 3.5: Likes and dislikes

300 sentences randomly chosen from various categories including sports, music, religion, politics, cats, and food. An independent human annotator then labels each sentence to either ◐, ⊖ or ◑ depending on his/her understanding of the sentence. We then evaluate SentiWordNet's accuracy using precision metrics (automated labeling against manually annotated labels). This measure have been commonly used to evaluate the accuracy of various retrieval, classification, and mining algorithms. Precision refers to the proportion of true positives over the sum of the true positives and false positives. The sentiment mining technique provided an overall accuracy of 69.33%, see Table 3.6 where the diagonal figures represent the accurate labeling, while off diagonal figures represent false positives.

		Estimated/Sentiwordnet		
		◐	⊖	◑
Actual/Human	◐	**66.33**	17.35	16.33
	⊖	23.68	**63.16**	13.16
	◑	18.25	06.35	**73.02**

Table 3.6: Confusion matrix for sentiment (%)

Sentiment analysis failed for conjugate and multi sentences. For example, "*As far as intelligence goes cats have a different kind of intelligence than that of dogs. They can MANIPULATE their environment to SURVIVE can hunt on their own and...*" is labeled negative and "*really no 1 can say our governance in 9ja is understood... we are just driven here and there no prosperous direction... we really don't know... Im shot of words 4 my dear country*" is labeled as positive. Sentiment approach also failed for sarcastic statements like "*GREAT.. now I can not get*

ONLINE..." which has been labeled positive.

The inaccurate estimation of positive to negative or negative to positive labels has more impact on building the user profile. The neutral messages are overwritten by the page polarity and hence no impact on user profile.

3.4.6 Concentrated group with ground truth

We tried to get some ground truth to be compared with the results obtained. We chose 450 users from $u^{=}$ obtained using the SPM approach with the largest number of interests inferred (more than 4 in particular). Out of the 450 user profiles, 47 have either been deactivated or deleted. We manually sent Facebook friend requests and messages to the remaining 403 Facebook users to know more about their interests in certain categories. Due to privacy settings imposed by many users, we were only able to send 334 friend request (70 accepted) and 299 messages (15 replied back).

We expected those who accepted our friend request would reply to our messages sent; however, there were only 12 (out of 70) who replied to our message. 56 did not reply and we were not able to send message to 2 users because of the privacy settings imposed by them.

Majority users (212) neither accepted our friend request nor responded to our message sent. There were 2 participants who did not accepted our friend request but still replied to the message. Please refer to Table 3.7 for a summary of the responses we got.

		Disallow Message	Allow Message	
			No reply	Replied
Friend req. not allowed		52	16	1
Friend req. allowed	Not added	50	212	2
	Added	2	56	12

Table 3.7: Users category settings in `VolProf`

We take the responses of those 15 users who replied to our messages and compared them with the corresponding interests inferred using SPM. Figure 3.6 shows the percentage of the correctly inferred interests of these users. It shows that we

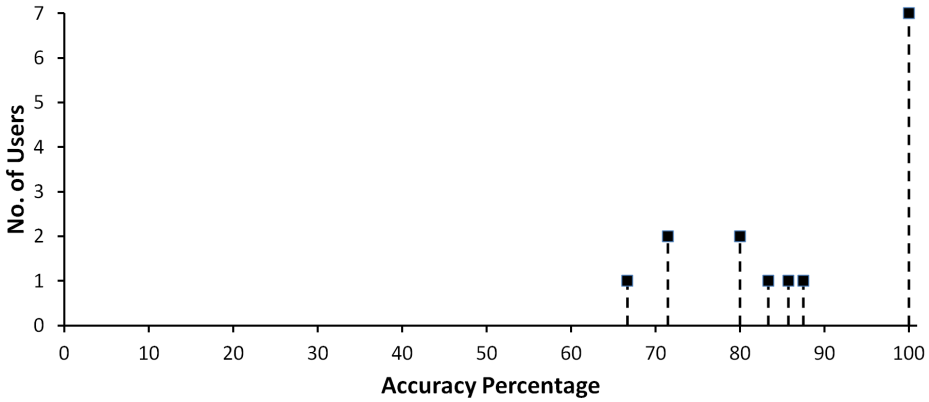


Figure 3.6: Users Interest from `VolProf`

were able to infer approximately half of the users' interests with 100% accuracy. Also, we can see from the figure that at least two-third of the interests are inferred correctly. Although we did not manage to get a larger pool of people with ground truth due to the manual work involved, available data seems to suggest that our technique provides reasonable accuracy in inferring users' interests from public pages on Facebook.

3.5 Discussion and limitations

There are many factors that could have contributed to inaccuracies in our analysis. In this section, we discuss some of them and also point out limitations in our results. One of the most important contribution to inaccuracies in our analysis is noise in the public pages on Facebook. This noise could come from advertisements, spamming in general, conversations in comments, and others.

Facebook is popular marketing media and some users are actually advertisers. For example, a person selling Nike shoes may post his ad in all the categories corresponding to sports. Our system might therefore believe that this user has an interest in sports. To have a sense on the noise level, we search for advertisements in selected categories by randomly choosing some posts and manually labeling them as advertisements. Table 3.8 shows the number of advertisements found in a number of posts for selected categories.

Category	No of posts scanned	% of Ads found
Sports	1,348	0.445
Food	2,312	0.346
Cats	2,312	0.216
Music	7,000	0.171
Politics	9,928	0.060

Table 3.8: Percentage of advertisement posts

It appears that the noise level in our dataset is very low, however, our manual process in finding advertisements might be error prone, too. Another step we took to minimize this error was to manually check whether the users inferred with more than eight interests are real users. Our simple manual checking revealed that except a few users whose accounts had been deactivated or deleted, all of them seem to be legitimate. We also filtered off users who posted same message in more than four pages.

We have seen many people get into their personal conversations on public pages that are not related to the topic of the page. Unfortunately we do not find a scalable way of filtering out such noise, and therefore it might have contributed to errors made.

There are also limitations in the techniques that we use. First, we use a context independent text mining algorithm. This limits our capability in analyzing the sentiment of certain pages, e.g., "Lets help the dogs in the streets and kick the cats." The sentiment scoring without context will fail to identify the user interest in this example. To solve this problem, we need additional scoring models that can handle the sentiment with the context.

Second, we only managed to obtain ground truth for a small set of users. We wish we could find a better approach to obtain the ground truth, but sending out message to a large number of users had one of our Facebook account suspended, and that was why we did not go further to target a larger group.

Last but not the least, manual work was involved in a number of steps in our experiments, including evaluation of the accuracy of sentiment analysis, spam de-

tection, etc. This manual work could potentially introduce errors into the evaluation.

To summarize, in this work we demonstrate the technique to mine secrecy information from the public data for Facebook and how it can be used to attack Blue Moon^{TM} authentication system. We present two mining based approaches to predict private undisclosed users' interests. Using only simple mining approach we extracted unobservable Interest topics by analyzing the corpus of Interests obtained via legitimate use of Graph API provided by Facebook. From our experiments, we were able to disclose 22 interests of a user and found more than 80,097 users with at least more than 2 interests. We also showed how this inferred information can be used to break preference based backup authentication system. We also demonstrated that simply liking a Facebook page does not imply the users' inclination towards that page. There exists many users who liked a Facebook page, however they have posted negative comments/posts on the page. We also found that categories like sports, music, food can be easily found on the social networking sites and should not be used in creating authentication challenges.

Chapter 4

Memorable fingerprints for authentication

4.1 Introduction

Profiling or fingerprinting human behavior has been widely used as a technique in providing context awareness [13], intrusion detection [167], etc. The uniqueness of such fingerprints for their collision resistance for authentication [133] and identification purposes [85] are important properties. Such fingerprints are usually stored on devices and are not memorable by the end users themselves. For example, users do not typically remember their DNA sequences or cryptographic keys generated from behavioral biometric [116] (which are forms of fingerprints). Since we are using users' private information which is not publicly available, it can resist to secrecy information inference attack demonstrated in Chapter 3.

However, there are many scenarios in which *memorable* fingerprints are desirable in profiling human behavior. One obvious application is in authenticating users who are not technically proficient. Memorable fingerprints are highly useful in these situations as they can be used to generate authentication questions that anyone can answer without memorizing or needing any physical device. Note that *memorable is usually more than memorizable*, i.e., a memorable fingerprint is one that can be

recalled and recognized by human users, but is not necessarily one that needs to be memorized. This helps in eliminating human factor issues like memory interference because of too many passwords. People tend to note down their passwords on the paper. If this kind of system exists then people do not need to remember their passwords and still are able to login and authenticate to the system. Another interesting way these memorable fingerprints could be used (when transformed to questions) is to determine how close two people are to each other. The better someone else (a friend/acquaintance) can answer these questions, probably the closer friend he/she is.

In addition, memorable fingerprints achieve their desired properties by capturing information and behaviors that are more noteworthy and significant from the user's perspective. Memorable behavioral fingerprints are especially useful for context-aware applications as it is usually difficult to understand which aspect of a context a particular user deems most important, especially when the context is derived from multiple data sources or the various aspects are conflicting from one another. For example, a user might have two sets of nighttime behaviors. One is to call a friend at 10 pm while the other is to play a mobile game while calling. Which of these behavioral fingerprints is more important to the user? A memorable fingerprint would be more significant to the user and should be given higher weight when providing context awareness.

Generating memorable fingerprints from behavioral data is nontrivial. Users typically do not remember details of regular past events especially when they are not asked to memorize them. There is no predefined criteria detailing the exact types of events or information that are more memorable. In this work, we present HuMan: **H**istory-based **U**ser Centric **M**emorable **A**pplicatio**N** for generating memorable fingerprints of cellphone users. In this work, we chose cellphones as the platform for generating memorable fingerprints because of the following reasons. First, cellphones are almost a human necessity nowadays [59]. Second, cellphones are primarily used by a single person, and are almost always with that person or

close by. Third, the huge amount of information such as phone calls, messaging, Internet browsing, instant messaging, game playing, and location data that can be collected from cellphones makes the memorable fingerprints attractive in various applications mentioned above.

HuMan consists of two modules: a data collection module that runs on the user's cellphone that monitors and records the events, e.g., SMSes sent and received, incoming/outgoing/missed calls, location, etc. The data mining module processes the collected records and generates the memorable fingerprint e.g. "When a call is made, the callee is Bill". To evaluate the memorability of fingerprints, we developed a simple mobile authentication application. In particular, we translated the fingerprints into questions with reasonable candidate answers (e.g., a question involving names would pick the other name choices from the participant's cellphone's contact list) used them as a challenge. Thus, this helps us to test the viability and usefulness of our fingerprints. The key features and contributions of this work are:

- **Memorable fingerprints**: HuMan is the first attempt to generate memorable fingerprints from the users' cellphone usage behavior. HuMan does not require a technically proficient user.

- **Multi-context data from cellphone usage**: HuMan generates fingerprints that are derived from data sources including call, SMS, email, calendar, application usage and browsing. We do this because 1) fingerprints of different users are usually different, and 2) adversaries can not memorize static rules to break the system.

- **Useful and effective in real-world applications**: We designed an authentication application based on HuMan to authenticate users and aims to achieve security properties (entropy) close to a 6-digit numeric PIN while providing desirable security features such as memorability. Please see section 4.6 for results.

- **Security protection**: We subject HuMan to a *difficult* security threat model where *intimates* (family members, close friends particularly those living with the participant) and *acquaintance* (casual friends, colleagues particularly those not spending a lot of time interacting with the participant) try to guess the fingerprints, and show that it provides *moderate* resistance to these threats. This is difficult as we expect family members and friends to be involved in a significant number of common activities such as calls and SMSes and are probably aware of a lot of user activities. In addition, family members might be able to infer events that they constantly observe, e.g., a fingerprint like "On Sunday mornings, you are likely to call Bill". We leave the scenario where an attacker has a complete log of the data communication from and to the cellphone as a future work.

These features and contributions were validated via a user study involving 58 participants on two phases on two different phone operating systems (Symbian S60 3rd Edition and Android v2.1 and above). By analyzing results from the user studies, we shed light on the characteristics of memorable fingerprints and how they can be generated. We also show what kind of fingerprints should be used and what should not be used in creating authentication challenges.

4.2 Related work

There has been previous work that tries to understand the behavior of cellphone users [36, 98, 78, 79, 18]. Unlike those studies, HuMan is the first system which uses cellphone usage data from multiple data sources to generate memorable profiles for the users. We now describe the differences from past studies in more detail.

Hong et al. studied the behavior of mobile data service users [78]. In our user study, we are also concerned with user behavior; however rather than investigating factors that affect their behavior, we would like to find memorable signatures that characterize their behavior. In another study, Hong et al. investigated models that

determined mobile Internet usage [79]. Our fingerprints could also be viewed as a collective model of user behavior. In addition, we are concerned with the behavior of individual cellphone users and the construction of memorable fingerprints. Belwal et al. [18] studied cellphone usage among university students in Oman via surveys, while we focus on actual usage measurements.

Using historical data to fingerprint user profiles and network profiles have been used extensively to detect illegal authorization, intrusion, etc, [22, 123, 127, 62]. For example, "Black-box" (or "gray-box") host-based intrusion detectors are trained with system-call traces of the program when processing intended inputs [49, 51, 56, 57, 141, 145]. Our system is different from these intrusion detectors in that HuMan tries to learn the usage patterns and generate memorable fingerprints, but does not try to detect deviations to these patterns. Our fingerprints also need to be memorable and dynamic; these characteristics are often ignored by intrusion detections systems.

Emerging human-centric authentication systems proposed by Nosseir et al. [124] make use of context data to pose questions to end users. Unlike these systems, HuMan does not restrict itself to one source of information and our fingerprint approach can be used for more than just authentication. In addition, HuMan's question generation is derived from the fingerprints and not from the raw data and finally, HuMan's evaluation uses a much harder attacker model involving intimates and acquaintances instead of strangers.

Traditional biometric authentication schemes such as face [66], fingerprint (Motorola Atrix) and voice also include low-level features extraction, data processing and detection phase. These schemes have become technically feasible on the mobile platforms [156], though suffers from well-known issues e.g. low-light, different facial orientation, image quality of the camera, surrounding noise, oily fingers, stained sensors etc [3]. The above mentioned schemes are also subjected to spoofing attacks e.g. using a picture, pre-recorded voice and fake fingerprint of the user which can result to high false acceptance rates [112]. HuMan uses a combination of features to generate fingerprints rather than relying on one source of information which can be

used for authentication purpose. Recently, there has been similar work in designing context-aware applications on the cellphones [40, 172, 12]. The focus is to understand users' context and explore any design implications on the device. In addition, substantial work has been done in measuring, quantifying and predicting user activities (walking, standing, etc.) using sensors such as accelerometers [20, 73, 67].

4.3 Architecture of HuMan

We propose and implement HuMan to generate memorable fingerprints from cellphone usage. HuMan comprises of two modules; a data collection module and a fingerprint generation module (see Figure 4.1). The data collection module runs in the background on cellphones and unobtrusively logs all interesting user events. To produce fingerprints with rich entropy, HuMan collects a wide range of information on call, SMS, application, browsing, etc. This forms the base of HuMan with hundreds of thousands of data entries.

The fingerprint generation module resides above the data collection module (see Figure 4.1) and consists of three sub-modules. The *rule mining sub-module* uses data mining techniques to process the raw data and generates thousands of rules which are machine-recognizable with high confidence and support. These rules are further processed by the *template fitting sub-module* to combine and transform machine recognizable rules into hundreds of human-memorable rules using certain template instances. The last sub-module performs further filtering to obtain tens of memorable fingerprints that are user specific.

In the rest of this section, we provide details of the data collection and fingerprint generation modules of HuMan.

4.3.1 Data collection

The data collection module (logger) runs unobtrusively in the background of the cellphone and captures a wide range of high level application events that result from

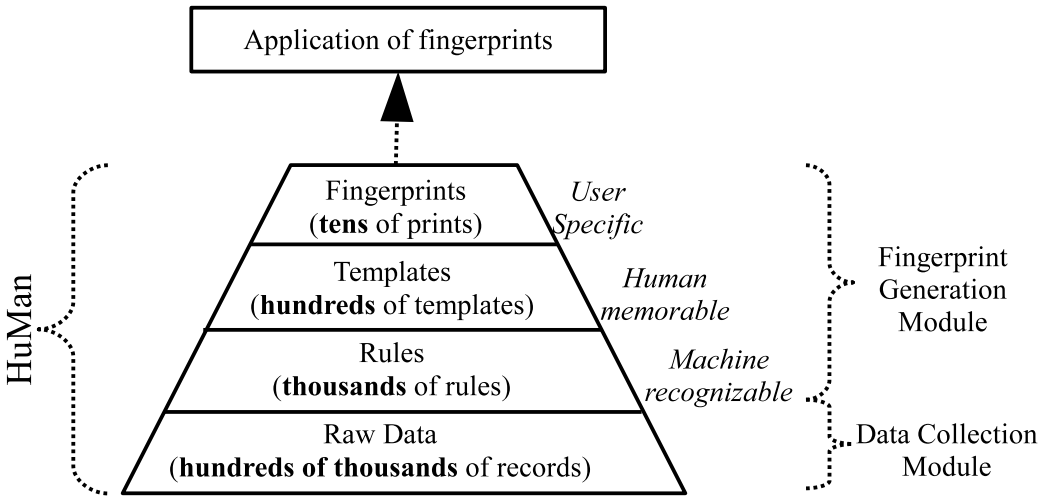


Figure 4.1: Architecture of HuMan

user-phone interactions directly and indirectly. Table 4.1 summarizes the different types of events logged in our implementation on Symbian (v3.0, 3.1 and 3.2) and Android (v2.1 and above) OSes. An additional watchdog program was installed to monitor the execution of the logger and to restart it in case of unintended shutdown.

Implementation and Challenges on Symbian

The Symbian based logger (5506 lines of Carbide C++ code) was developed for phones running the 3rd Generation Symbian S60 OS. As shown in Figure 4.2, the logger operates on the Symbian stack, drawing on the lower level API from the generic OS, communication and connectivity services.

The development of the logger on the Symbian platform encountered some challenges. For example, we were limited by the available APIs and are therefore unable to obtain touch events or keystrokes from touch screen phones. We were also limited to using cell tower IDs to capture user location. In addition, due to lack of a geocoding translation service, we translated the positioning information to just three possibilities that could be easily identify from the cell tower ID information; namely, home, university and *rest*.

Events	Symbian	Android	Description
Phone profile state	●	●	Phone states: on/off/charging/etc.
Phone and Messaging	●	●	Receive/Send SMS/MMS messages. Making/Receiving phone calls.
Calendar events	●	●	Items on the user's calendar (time, location, notes, etc).
Application events	●	●	Application states such as installed/started events.
Cell tower id	●	●	Changes to the GSM cell tower ids (Track user's location in Symbian).
Location	●	●	User's location. Every 5 min on Android, either GPS or WiFi/CellTower positioning.
Access of Media Files	○	●	Audio, Video and Images. Time, name, duration, artist are captured.
Radio states events	●	●	WiFi, bluetooth states: On/Off.
Contact list events	●	●	Contact changes add/delete.
Usability Events	○	◑	User touch events on standard Android UI.
Home Screen Placement	○	◑	Placement of shortcuts on the home screen/s. Widget unavailable.
Email	○	◑	Email events: View/Compose etc. Email and Gmail clients. Excluding content
IM Data	○	●	Chat time, user ids. IM clients: MSN, Ebuddy, Meebo, Gtalk.
Browser history	○	◑	Browser history.
Network Data	○	◑	Network data in the TCPDump log.
System settings	○	●	Change events on phone settings. Volume/ringtone etc.
Search Data	○	◑	Search query. Only Global and App searches. Map search unavailable

Table 4.1: Events logged by HuMan on Symbian and Android; ●-Yes, ○-No, ◑-Partial

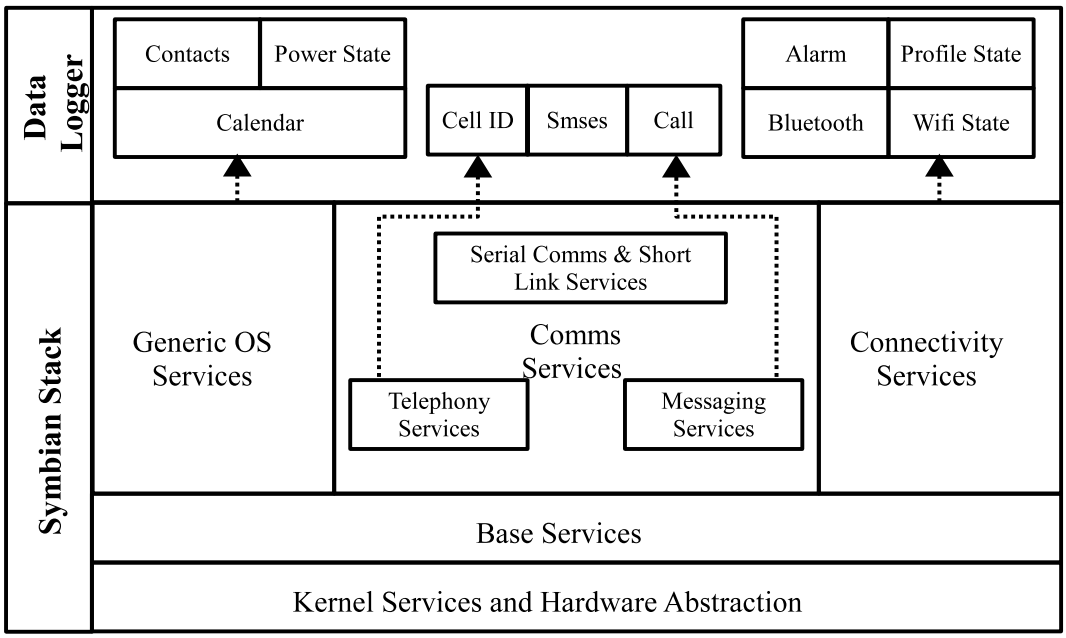


Figure 4.2: Symbian data logger

Implementation and Challenges on Android

The Android platform offers a much richer set of applications and APIs that allowed us to log many additional types of events. The Android logger (a system application) was built in Java (11,935 lines of code) for Android OS 2.1 and above.

The data collection module on Android makes use of the supported API and the root access. The supported API enables data collection from the user-phone interaction while the root access extracts data beyond the API limitations. As shown in Figure 4.3, the data logger operates on top of the Android stack and pulls the required data described in Table 4.1 using the application framework APIs. It also obtains root privileges in order to obtain additional information, e.g. Emails, Gtalk, Dolphin Browser history, etc.

Root access is needed to overcome three main difficulties encountered during the logging of the data. First, the APIs were limited in the type of data being exposed. Second, the APIs were not designed for building applications that actively log user activities. Lastly, callbacks of many events were not available. Root access is done with the "hotplug" method that exploited an Android vulnerability to install an accessible modified root shell, with the user's consent [19].

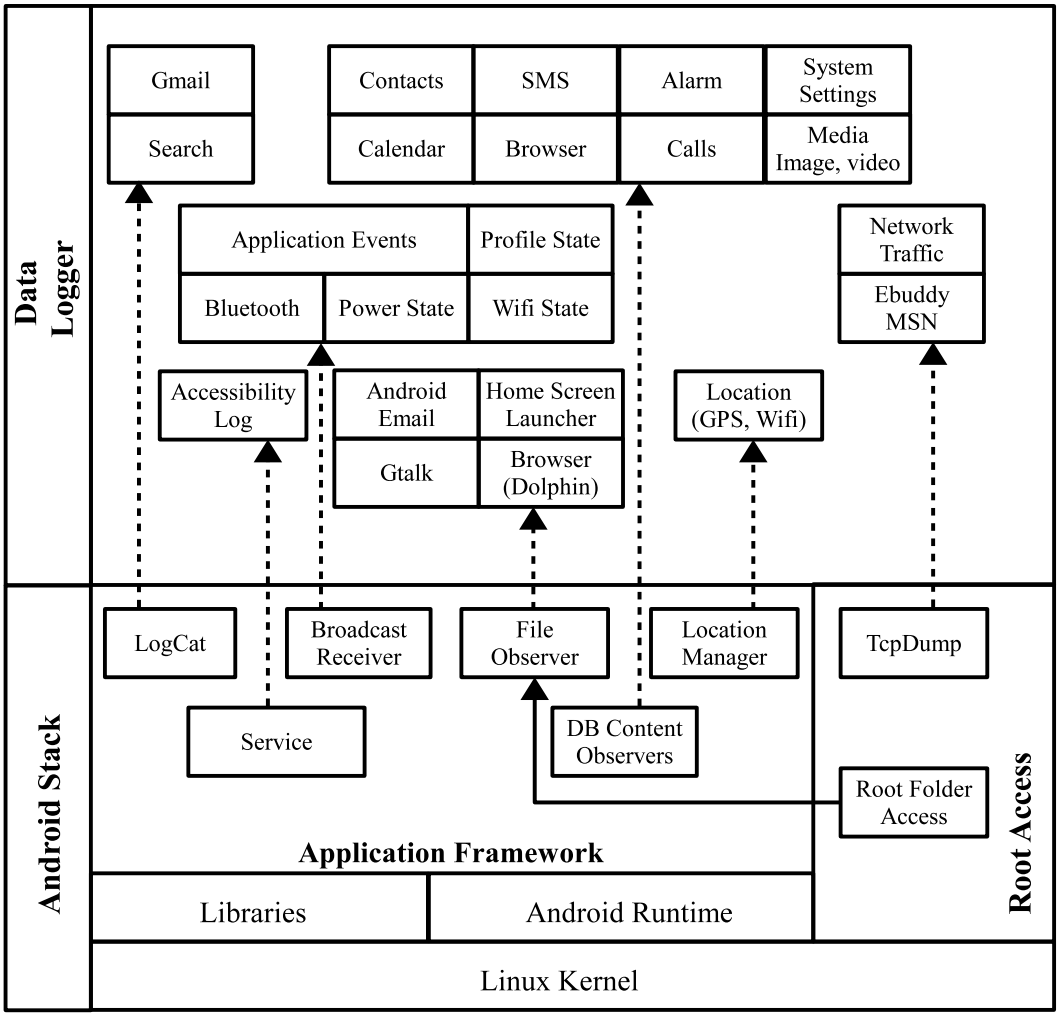


Figure 4.3: Android data logger

Memory and Battery Usage

The Symbian logger has an installed base of 66KB on Nokia N79. In idle periods (with logger running), the power consumption is approximately 0.6% per hour. Memory consumption is about 280KB. When there are a few calls and SMS the power consumption is about 1.6% per hour. There is little change in memory consumption though. Typical size of the log over a six-week collection period is around 6 MB.

For the Android platform, the logger has an installed base of 85KB on Nexus one. In the idle periods (with logger running), the power consumption is about 1% per hour with memory consumption of about 160MB. With a constant usage of browser, application installation, social network activity, Gmail, camera and Google maps, power consumption is on average 12.5% drop per hour, with an average memory consumption of 162MB. Typical size of the log over a four-week collection period is around 25MB.

We notice that the battery consumption on Android is higher. This is likely due to the large amount of information logged including all types of events. We plan on reducing the logging activity when users are not using their phones to reduce the power consumption.

4.3.2 Fingerprint generation

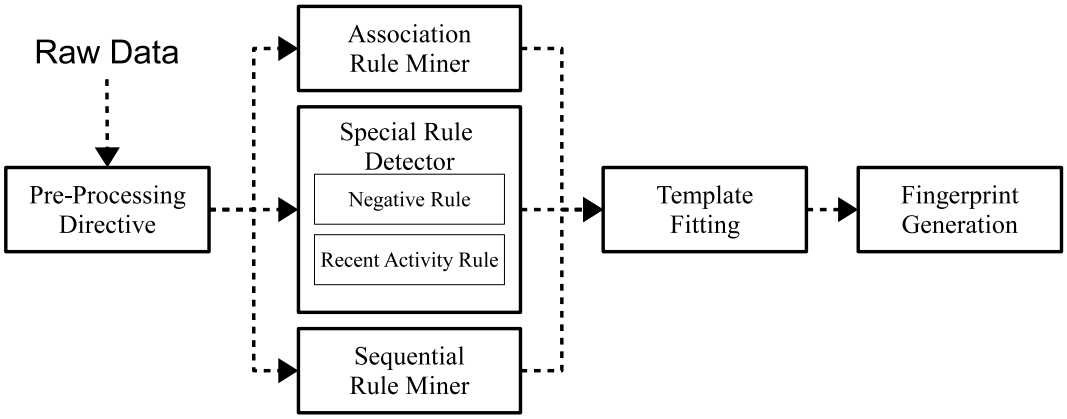


Figure 4.4: Fingerprint generation from raw events

Figure 4.4 shows the sub-modules used by HuMan to generate memorable fingerprints. The raw events are first preprocessed into the right formats for the data mining sub-module, which consists of standard association rule mining [5], sequential pattern mining [110, 144] and a special rule detector module (see Section 4.3.2). A template-fitting sub-module then transforms the rules learned from data mining into a form that is more likely to be memorable (see Section 4.3.2). Finally, a submodule filters them based on our design criteria to obtain memorable fingerprints (Section 4.3.2). Currently, in our design this submodule is *semi-automatic*, however we provide some insights of how the full automation on this module is possible. We walk-through the fingerprint generation procedure using an example shown in Figure 4.5.

Pre-processing Directive

The data logged by our data collection module is in the form of raw events, e.g.,

• [Event:Call] [CallType:Outgoing] [To:Bill] [StartTime: 8-12pm] [EndTime:8-15pm]

• [Event:Set Alarm] [SetTime:09-45am] [AlarmTime:7am]

i.e. "call made to Bill at 8:12 pm and ended at 8:15 pm", or, "set an alarm at 9:45am for 7am" which are hardly memorable by human beings. To overcome this we process the raw data to transform the fields like exact times (e.g. 8:12pm) to more generic values (e.g. night), exact coordinates (e.g. 32.008076,23.48877) to an area (e.g. downtown), day of the week to weekdays/weekends etc. Figure 4.5 (step 1) shows the transformation of a set of 5 such events.

Machine-recognizable rules

In this section, we describe the rules generation process using user's raw cellphone usage data. One approach could be to enumerate all instances of events and find those that are discriminative. However, this would be slow e.g. even for a simple pattern "$< event - 1 >$ is before $< event - 2 >$" has many possible instances with concrete events replacing "$< event - 1 >$" and "$< event - 2 >$". We use

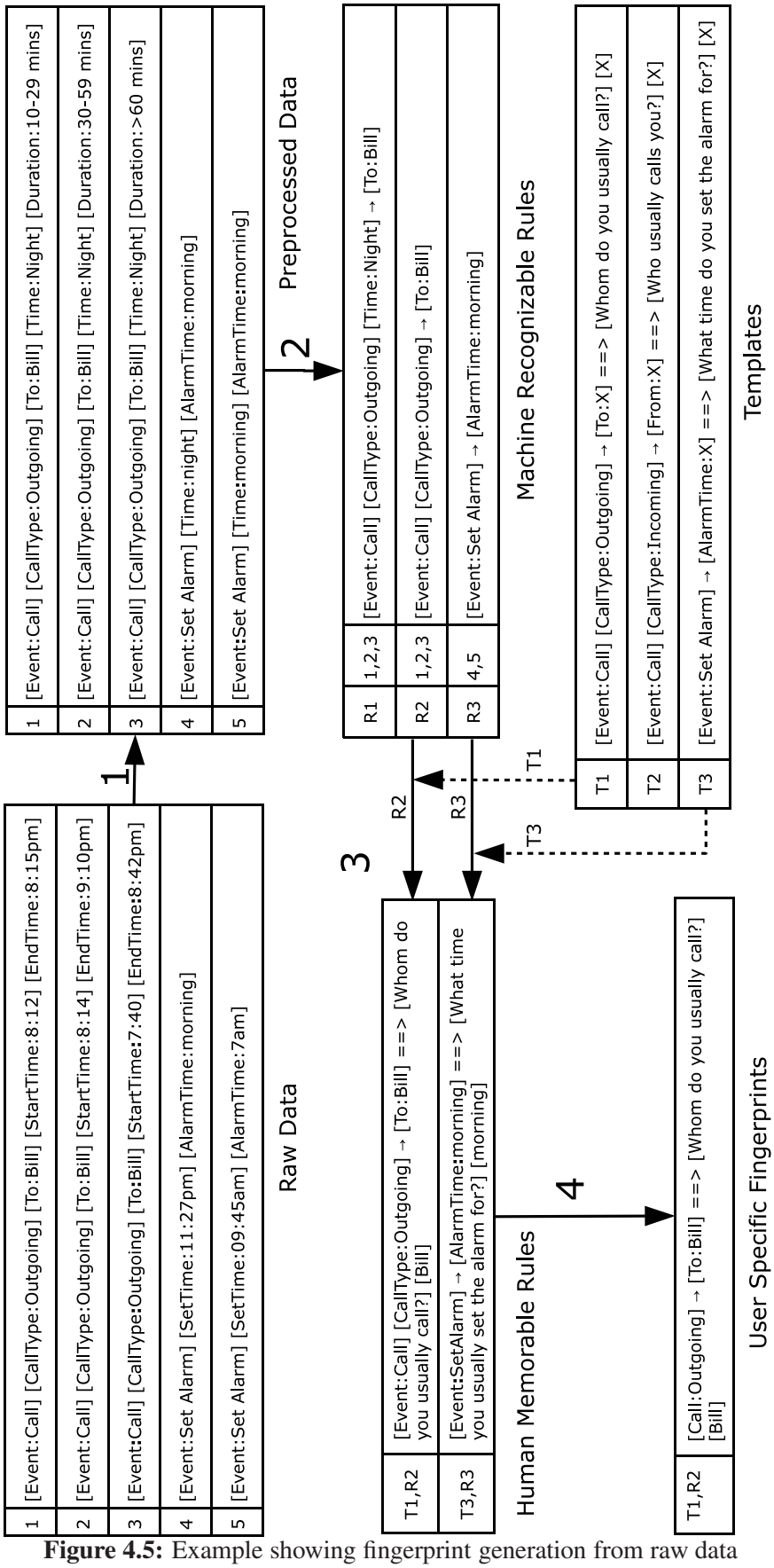


Figure 4.5: Example showing fingerprint generation from raw data

data mining technique to generate machine recognizable rules. This helps us to find which of the instances of the rules are discriminative for a particular user efficiently.

An association rule describes a set of events that happen as the consequence of another set of events. HuMan treats cellphone related events, e.g., a user makes a phone call, as a transaction, and considers various features of this event, e.g., the phone number called, the duration of the call, the location from where the call is made, the time/day when the call is made, etc., as items in the transaction. The preprocessed data is then analyzed to form rules, e.g., the first three events from preprocessed data in step 2 of Figure 4.5 form the rules $R1$ and $R2$

• [Event:Call] [CallType:Outgoing] [Time:Night] $\rightarrow$ [To:Bill]

• [Event:Call] [CallType:Outgoing] $\rightarrow$ [To:Bill]

respectively i.e. "Whenever there is an outgoing call at night, the callee is always Bill", or, "Whenever there is an outgoing call, the callee is always Bill".

A sequential rule describes two sets of events that happen one after another in a sequence. We split the history of cellphone events into a set of sequences by employing a windowing approach. Events separated by not more than 180 seconds are grouped into the same window. We put an event into a new window if the current window already contains more than 30 events. Each window corresponds to a sequence of events. We can consider longer events too. However, the computational requirement would be higher for longer events. Thus it is a trade-off of accuracy versus efficiency. Some existing studies also limit a window to be of a particular size [113, 28, 105, 54]. This database of sequences is then analyzed to form rules like "Whenever Jack calls Bill on Sunday, he calls David right after it".

In the data-mining algorithms, *support* captures the number of times a rule is observed in a dataset. We use low relative support thresholds (.05% to 4% of the size of the dataset) as the dataset is diverse. *Confidence* captures the likelihood of the rule's pre-condition to be followed by its post-condition. We use a confidence threshold >50% to remove many spurious rules.

Human-memorable rules

The data mining techniques help us learn tens of thousands of machine-recognizable rules with high confidence and support. However, not all these rules are necessarily memorable by human. Hence, we fit the machine-recognizable rules to hundreds of *templates* and filter out the ones that are less likely to be memorable.

Thus, to build our templates to transform machine rules into a human memorable format, we first developed heuristics to rank the memorability of various types of information. We surveyed people to learn these memorability heuristics.

1. We noticed that people could remember communication (e.g., SMSes, calls etc.) and application-based events (e.g., Apps, the action performed etc.) better.

2. We also observed from our survey data that information corresponding to events that directly resulted from human-phone interactions were more easily memorable.

3. An interesting survey finding was that many people tend to remember *negative* rules and rules about *recent activities* well. Negative rules correspond to events that had never occurred. For example, "you have never called X" and "you never sent a message to Y".

4. Recent activity rules ignore the rule's support in favor of the time when the rule last occurred. For example, "application-X was the last one installed", "The last change of the alarm clock setting was on day-Y".

We used these heuristics to create our set of templates, which are in the form of rules with placeholders indicating information that can be easily memorable. Following are some examples of templates as shown in Figure 4.5,

• [Event:Call] [CallType:Outgoing] $\rightarrow$ [To:X] $\Rightarrow$ [Whom do you usually call?] [X]

• [Event:Call] [CallType:Incoming] $\rightarrow$ [From:X] $\Rightarrow$ [Who usually calls you?] [X]

The above two templates can be derived as "Whenever there is an outgoing call, the callee is always X", or, "Whenever there is an incoming call, the caller is always X", where X is the placeholder indicating information that can be easily memorable (e.g. Bill). Each template has two parts separated by '$\Rightarrow$', where the *antecedent* denotes the rules and *consequent* denotes the English translation of the rule. Please note, *consequent* can differ depending on the use case and the application of the fingerprints. In our case we used authentication as an application therefore, we used the questions as English translation.

The entire template generation is a one time process and is not user specific. These templates can be continuously updated when necessary to improve the memorability. Currently, we have 202 templates in HuMan. Note that the templates generated are *human-memorable*, but not *user-specific*. Fitting the machine-recognizable rules with the templates (an automatic process) is as easy as matching the placeholders (i.e. [X]) in the templates with corresponding information in the rules. For example (see Figure 4.5), template $T1$ and rule $R2$ are matched to form the human-memorable rule $[T1, R2]$, similarly, template $T3$ and rule $R3$ are matched to form $[T3, R3]$. The structure of $T1$ matches with $R2$, thus X is replaced by $Bill$ in both antecedent and consequent. Please note, there are a large number of non-template matching machine-recognizable rules that are filtered out automatically as they are not likely to be memorable.

User-specific fingerprints

The generated template fitted rules which are human-memorable may not be user-specific e.g. (as shown in Figure 4.5) rule $[T3, R3]$,

[Event:SetAlarm] $\rightarrow$ [AlarmTime:morning] $\Rightarrow$ [What time do you usually set the alarm for?] [morning]

says "the alarm is set for morning" could apply equally to multiple users. We thus *manually* filter these rules to pick the most user-specific rules. In future, this can be automated by filtering out the common fingerprints generated across majority of the

users.

We also filter out conflicting and redundant rules at this stage as they reduce the entropy. For example, "On Sunday, when a call is made, the callee is Bill" and "When a call is made, the callee is Bill" overlap and should not appear together in the same fingerprint. If one fingerprint is a subset of the other, then only the one which has higher support is included in the final set of fingerprint. Finally, the small set of the most memorable template rules that do not overlap are combined to form the final fingerprint. We are still learning the best way to pick good rules when forming the final fingerprint. We describe some of the lessons, regarding fingerprint selection, we learned from our user studies in the next few sections.

4.4 Evaluation methodology

To evaluate HuMan, we installed our logger on the cellphones of participants for a period of 6 and 4 weeks for the Symbian and Android studies, respectively, to collect the raw data. To evaluate the memorability of fingerprints, we used them as an authentication mechanism. In particular, we translated the fingerprints into questions with reasonable candidate answers (e.g., a question involving names would pick the other name choices from the participant's cellphone's contact list). Figure 4.6 shows the authentication program's User Interface.

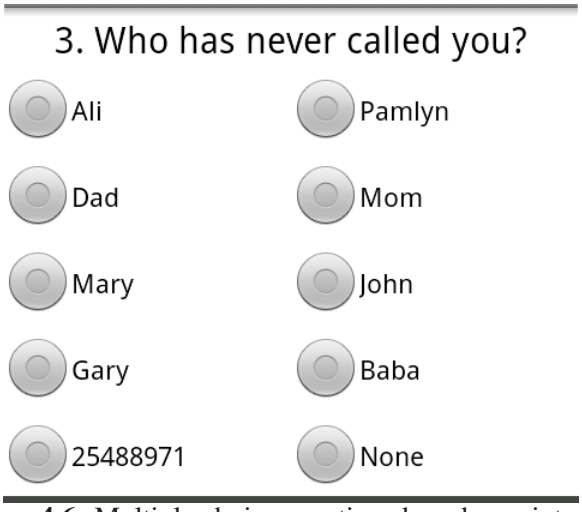


Figure 4.6: Multiple choice questions based user interface

Immediately after the data collection period, the participants were asked to come in for a series of tasks conducted in a lab environment. They were also asked to bring along two persons, an *intimate* (a close friend or family member that knows them very well) and an *acquaintance* (a casual friend that they socialize with on a regular basis). We paid USD $32 and $8 to each participant and persons (the participant brought in), respectively, for participating in the study. The ability of participants to answer these questions correctly gave us insights into the memorability of the fingerprints. The intimate and the acquaintance separately and independently answered the same set of questions. Intimate's and acquaintance's answers gave us insights whether fingerprints are actually resistant to attacks by people who know users the best. We conducted two user studies on two different platforms, i.e. Symbian and Android. Figure 4.7 shows these two studies (with two phases on Symbian and one phase on Android) and the improvements made to HuMan after each phase in chronological order.

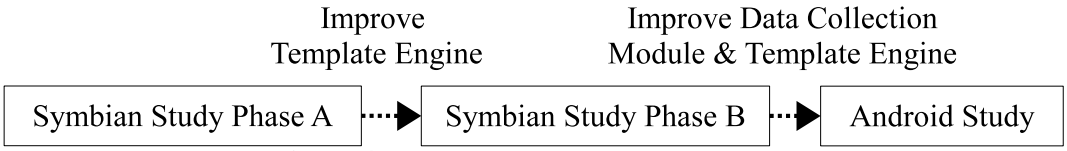


Figure 4.7: User studies design phase model

The first study was conducted on Symbian. We found that most of our Symbian users limited their cellphone usage to phone calls and SMSes, which might not provide enough entropy to generate good memorable fingerprints. This was a key reason why we performed the next study on Android, as we believed that Android users would exhibit much richer sets of activities, including applications like IM, Emails.

The feedback from the iterative design also helped HuMan to improve the ability to extract the *human-memorable* and *user-specific* rules. In particular, after Phase A of the Symbian study (see Figure 4.7), we improved HuMan's template generation engine. Similarly, the Android study was conducted with a version of HuMan that incorporated all the lessons learned from the entire Symbian study. We discuss the

lessons we learned regarding fingerprint generation from these studies at the end of the chapter.

4.5 Symbian study

In the Symbian study, two phases were conducted, Symbian phase A and Symbian phase B. We first discuss the study setup and results of phase A, then present the lessons we learned as well as improvements made to HuMan as a result, and finally discuss the results of phase B.

4.5.1 Participant selection

We solicited for participants from the undergraduate population at our university. The only requirement we had was that each participant be able to bring himself or herself, an intimate, and an acquaintance to the lab study after 6 weeks of data collection.

4.5.2 Experimental settings

We divided the experiments into two task sets: the base case experiments done by everyone (the participant, the intimate and acquaintance) and a set of specific experiments designed to investigate and validate specific parts of HuMan. Each task was performed on the phone and had the same question structure.

Task set 1 — Baseline Tests. As we adopted the authentication mechanism as the form of fingerprint verification, we decided to design the question-answering mechanism to have a similar entropy of a standard 6-digit pin (a password space of 10^6). Please note that, question/answers are not completely random, meaning that some answers have higher a-priori probabilities. For example, generally speaking a person is more likely to call someone with the same last name (likely a family member) in the evening time. Therefore, the corresponding answer has a higher

chance to be the correct answer. HuMan is designed to be comparable but the true entropy depends on the particular answers, (please see Section 4.5.2 Task set 2, for the details on how the answers are chosen). We designed three variants (6, 7, and 8 questions each with 10, 8, and 6 choices, respectively) of HuMan to test user preference on fewer questions with more choices versus more questions with fewer choices. The tradeoff is that with questions with fewer choices, answering each individual question might be easier, but the overall process takes longer. Conversely, having less questions with more choices might take less time overall, but answering each question is harder. We could not use less than 6 question with 10 choices as that would provide lower security guarantees than a 6-digit PIN. We cannot use free form question because it is harder for the machine to infer and verify (if the answer is correct or not). The participant's intimate and acquaintance were both asked to do the same baseline tests.

Task set 2 — Detailed Tests. In this set of experiments, the participants and the intimate/acquaintance were asked to do two different tasks that investigated specific portions of HuMan in greater details. We fixed the number of choices to 8 for each question to allow us to compare results across all participants.

1 **Incorrect Choices**: We varied the way that we picked the incorrect choices presented to the user for each question to understand the impact of the answer selection process on HuMan. We selected the incorrect choices using three algorithms:

   i **100% related** — incorrect choices with support and/or confidence *close* to that of the correct answer. For example, if the question asked is "Who do you call the most?", then the incorrect choices are from the pool of contacts whom the user has frequently called.

   ii **100% unrelated** — incorrect choices with support and/or confidence *not close* to that of the correct answer. For example, if the question asked is "Who do you call the most?", then the incorrect choices are from the pool of contacts whom the user has never called.

iii **50% unrelated - 50% related** — a 50-50 mix of the above two.

2 **Support/Confidence**: We changed the confidence and support thresholds of the fingerprints to understand if there is a threshold value below which participants were unable to reliably answer the questions. In addition, we also wanted to understand if participants were better in *low*-support— *high*-confidence questions (rare unique events).

The total number of questions in the different set of experiment were 18 (6 for each option) for "Choices" and 6 for "Support/Confidence". We randomly separated the participants into 3 equal, non-overlapping groups and assigned a different set of experiments to each.

4.5.3 Symbian study phase A

Our participants were a mix of students from technical and non-technical majors. In total we had 31 participants (10 male, 21 female) and their corresponding intimates and acquaintance from the undergraduate population at our university. 21 of the intimates spent between 4-8 hours per day with the participant while the remaining 10 intimates lived with the participant. Among the 31 acquaintances, 19 spent around 1-4 hours per day with the participant, while the rest saw the participant almost daily but did not really interact with him/her.

Results of the baseline tests We found no statistical difference (using t-test analysis for gender, technical qualification, etc.) in the accuracy of answers in all the 10-, 8-, and 6-choices variants. Therefore, we aggregated results from all three variants together in subsequent analysis. We evaluated the accuracy in terms of false rejection rate (FRR, when the participant was not able to login) and false acceptance rate (FAR, when intimates/acquaintances were able to login) for different threshold values (see Figure 4.8). The threshold is the percentage of questions a user/attacker needs to correctly answer to authenticate to the system e.g. if the threshold is 50% then the user only needs to answer 3 out of 6 questions correctly to authenticate

to the system. If the threshold is reduced, there will be less false non-matches but more false accepts. Correspondingly, a higher threshold will reduce the FAR but increase the FRR. Unfortunately, we found that the threshold where FAR and FRR meet (approximately 42%) is quite low (approximately 33%).

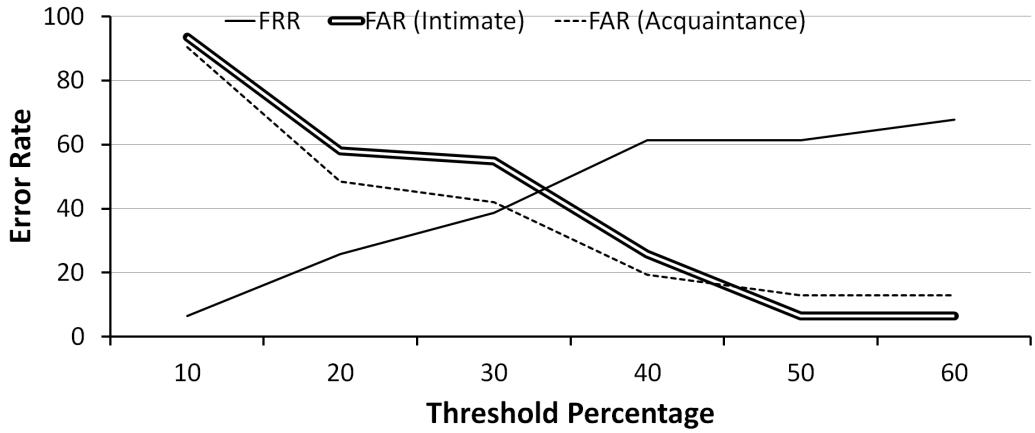


Figure 4.8: Symbian phase A- false acceptance & false rejection rates

To understand the reasons, we performed an in-depth analysis on the types of questions asked and categorized them into 4 categories depending on the focus of the question — *who* (questions about a person), *what* (about an activity), *when* (about time), and *where* (about a location).

We first calculate the accuracy of the participants, intimates, and acquaintances for all four categories. Figure 4.9 shows the results of this per-question analysis for the baseline tests.

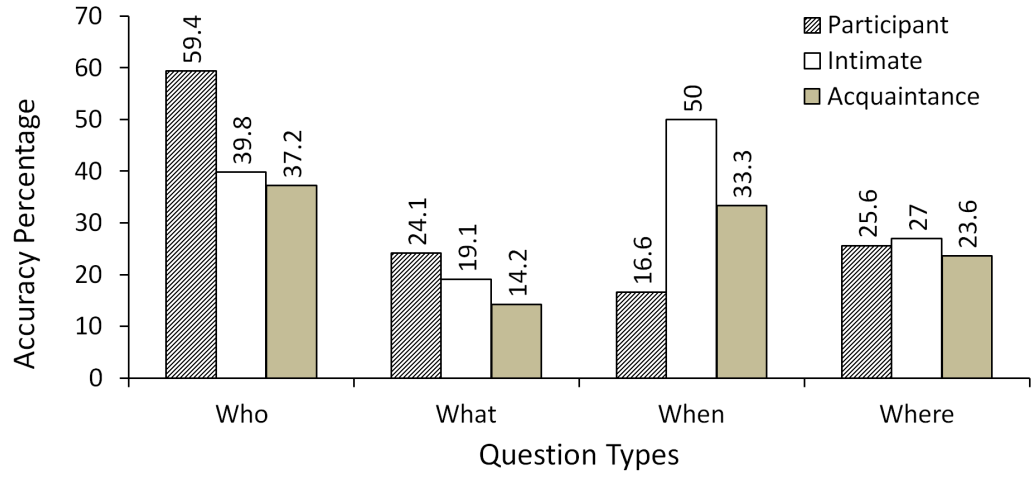


Figure 4.9: Effect of different types of questions (Symbian)

Who: For example, "who do you call the most?" This appeared to be the most reliable type of question that demonstrated an accuracy advantage of the participant over potential attackers. This could be caused by people being more sensitive and more capable of answering who-type questions. We therefore believe that adding more *who*-type of questions would improve the overall accuracy of our system.

What: For example, "what action do you usually perform with Bob?" There appears to be a relatively small difference between accuracy of the participants and the potential attackers. By investigating deeper, we found that some of the choices were misleading. For example, our questions differentiated deleting "sent SMS" events from deleting "received SMS" events, whereas the participants could only remember that they deleted an SMS. We therefore modified some templates used for what-type questions to make the choices clearer in phase B.

When: For example, "at what time of the day you would most likely use WiFi?" This type of questions has a negative overall impact as family intimates were able to answer them with even higher accuracy than the participant. Intuitively, this is possible when intimates spend a lot of time with the participant. We thus decided to use fewer *when*-questions in phase B.

Where: For example, "Where do you usually charge your phone?" The where-type questions did not perform well and we discovered that the accuracy for this type of questions was high for the intimates. Due to this unreliable variation, we decided to use fewer where-type questions in phase B.

Results of the detailed tests By carefully analyzing the two detailed question sets (different ways of picking incorrect choices and different support/confidence thresholds), we found that the threshold settings of confidence and support did not have strong impact on the accuracy of answering the generated questions. We therefore turned our attention to the various ways of picking choices for each question.

Intuitively, different ways of picking the possible choices for each question should have an impact on the accuracy. For example, it should be easier to an-

swer a question "who did you call last night" if all, but one, of the choices were people you do not know. Surprisingly, the results from this test showed very small accuracy differences for the participant when different methods for choosing the incorrect options were used. However, there was a slightly larger gap between the accuracy of the participant and the intimate when using the 100%-unrelated setting (see Figure 4.10). This result suggests that some (although relatively small) security advantage can be obtained if we favor the 100%-unrelated setting when picking choices for each question.

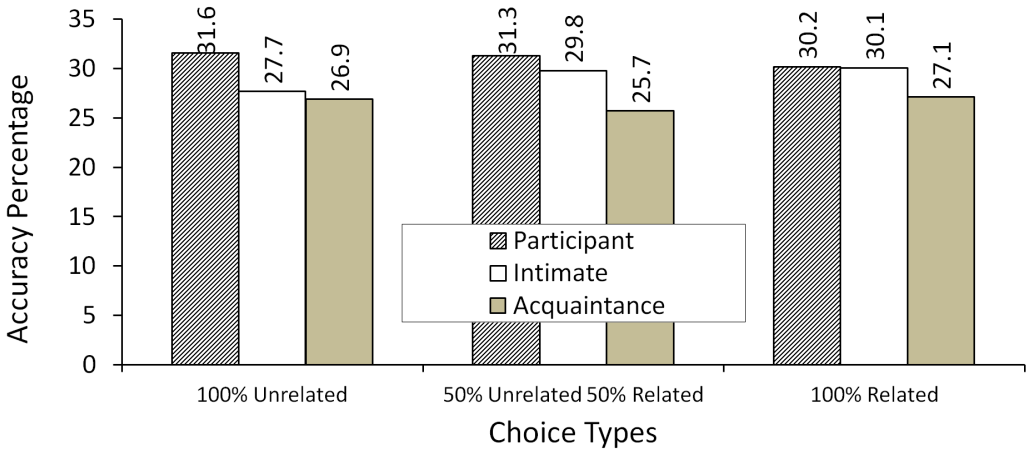


Figure 4.10: Effect of different incorrect choice picking method (Symbian phase A)

4.5.4 Symbian study phase B

Based on the result and analysis of Phase A, we made a few modifications to the template and fingerprint generation subsystem, namely to 1) favor more *who*- and *what*- type questions; 2) modify templates of *what*-type questions to improve clarity; 3) favor 100%-unrelated choices for each question. With these changes, we performed a second user study with 14 new participants (9 male and 5 female) and their corresponding intimates and acquaintances. There were 10 out of 14 participants from non-technical background. 5 of the intimates spent between 4-8 hours per day with the participant while the remaining 9 intimates lived with the participant. Among the 14 acquaintances, 4 spent around 1-4 hours per day with the

participant, while the rest saw the participant almost daily but did not really interact with him/her.

Figure 4.11 shows the FAR and FRR of our improved system. The results are promising, as the threshold is now increased to approximately 57% where FAR and FRR meet (approximately 38% for intimates and approximately 17% for acquaintances). The improved system thus has a clear security advantage over the previous one (Symbian Phase A, see Figure 4.8). However, still the system is not usable with high FAR and FRR.

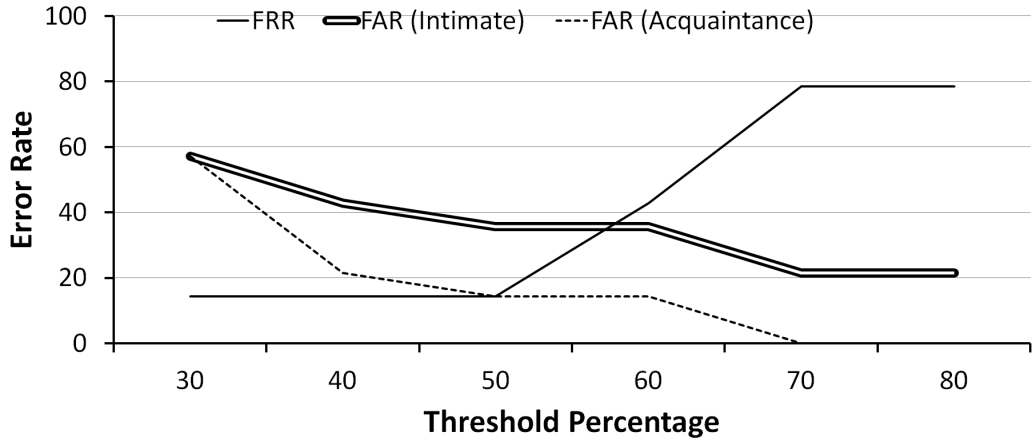


Figure 4.11: Symbian phase B - false acceptance & false rejection rates

We believe that the boost in the accuracy is largely due to the changes we made regarding the preferential choosing of certain types of question over others (as described above). We confirmed this by comparing the question-type distribution between the improved and the original system. Figure 4.12 shows this comparison. The larger percentage of who-type questions in the improved system is a significant cause of the overall accuracy improvement.

4.6 Android study

We found that our participants' Symbian usage behavior was limited to calls and SMSes. Unlike Symbian, Android provides a richer set of multi-context data. In this study, we investigated if better fingerprints could be generated from the richer

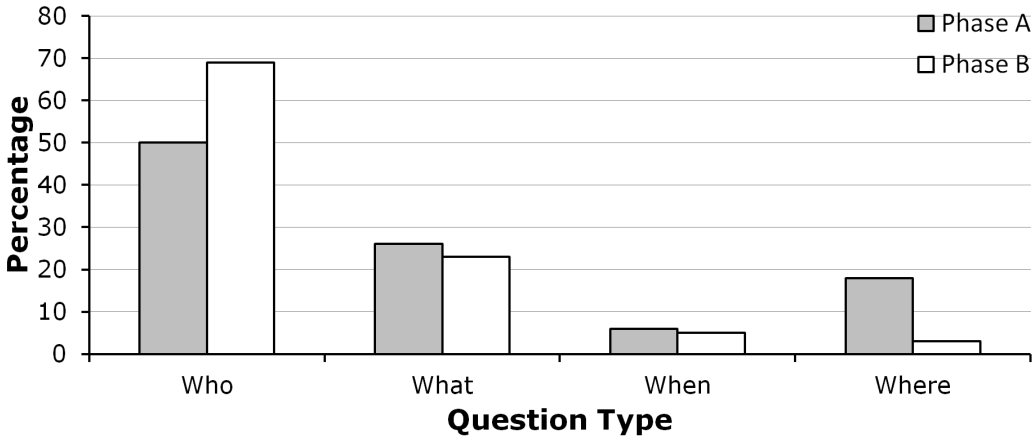


Figure 4.12: Symbian - Comparing the breakdown of type of questions asked between Symbian phase A and Symbian phase B

data-set. This study was similar to the Symbian study with the following notable differences:

1. *Data Mining Thresholds*. Similar to the Symbian study, we used low minimum support thresholds (1% and 0.5% for sequential rule mining and association rule mining, respectively) as the dataset was diverse. However, we used higher minimum confidence thresholds (80% and 70% for sequential rule mining and association rule mining, respectively) to further remove spurious rules.

2. *Choice of Fingerprints*. Based on the lessons learned from the Symbian study, we only incorporated *who* and *what* questions in our fingerprints. In the Symbian study, participants performed better for these types of questions (see Figure 4.9).

3. *User Interface Experiments*. We also introduced two new user interfaces and evaluated their effectiveness. For each question, the incorrect choices were 100% unrelated to the correct answer. This was again due to the lessons learned from our Symbian study.

4.6.1 Participant selection

In addition to undergraduates from our university, we also included working adults. In total, we had 13 participants (9 male and 4 female) out of which 9 were undergraduates (age between 19 and 25) and 4 (age between 24 and 33) were working professionals. We installed a data collector for one month. After one month, the participant, along with an intimate and acquaintance, conducted our in-lab study. 11 of the intimates spent between 4-8 hours per day with the participant while the remaining 2 intimates lived with the participant Among the 13 acquaintances, 4 spent around 1-4 hours per day with the participant, while the rest saw the participant almost daily but did not really interact with him/her.

4.6.2 Experimental settings

In the Android study, the participants, intimates, and acquaintances did the same experiments, which consisted of two tasks. All task were conducted on a Google Nexus One provided by us.

Task set 1 — Baseline Tests. We asked users to answer multiple-choice questions with the following characteristics: We asked 6 questions with 10 choices each, to achieve the same security strength as a 6-digit pin. We did not consider other options as our Symbian study showed that there were no significant differences when 6, 8, or 10 choices were used.

Task set 2 — Additional Tests. As android provides a much richer and nicer touch screen based interface, therefore in this task set, we investigated the use of spatial fingerprints on two interactive User Interfaces, as an alternative to multiple choice questions. This was not possible in Symbian.

Launcher User Interface Users keep shortcuts that they use frequently on their cellphone's home screen panel. We generated fingerprints pertaining to the location of these shortcuts by identifying applications which were launched from the home

screen panel and whose position had not changed during the data collection period. We then formed fingerprints such as "application x is located at position y in the home screen" for frequently used applications.

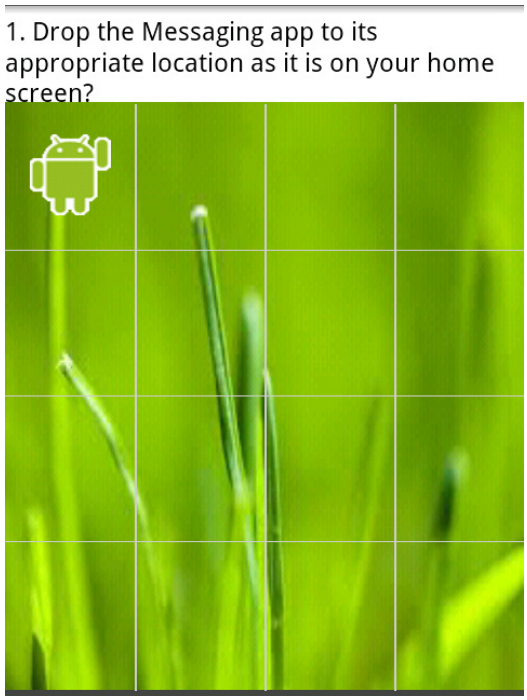
We presented these fingerprints as questions through a launcher User Interface (see Figure 4.13(a)) that asked users to drag the provided application icon to the right spot on a 4x4 sample home panel grid.

Map User Interface This interface investigated the use of a graphical user interface to answer *where*-type fingerprints (see Figure 4.13(b)). The user interface presented the participant with a question together with a map of our city divided into 20 segmented zones. The user had to select the correct zone to answer the question. Users could rotate the phone in landscape mode to view the full map, and use gestures like pinching to zoom in and out and flicking to move around the map to choose the correct zone.

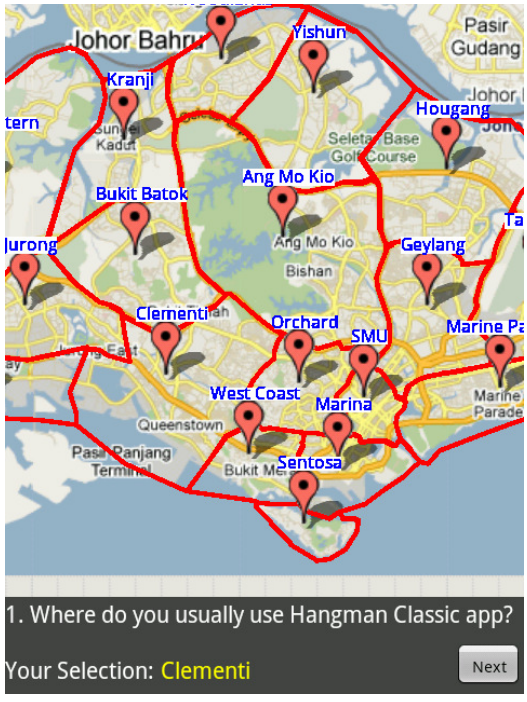
4.6.3 Results

Figure 4.14 shows the FAR and FRR of the test. This is a big improvement over the Symbian results. We were able to increase the threshold to 61.8% (entropy similar to a 4 digit pin) while decreasing both the FAR and FRR to approximately 15.3%. The improved accuracy was due to the changes in the user study design as well as the richness of the android multi-context data-set. One possible reason why intimates and acquaintances are still able to answer many questions correctly could be because they can observe a person and thus know a lot of details about the person peculiar habits and characteristics.

We found that the improved accuracy is because of the changes made to the user study and the multi-context data-set. More information on multi-context dataset can be found in the appendix A. However, additional experiments need to be done to explore this area of generating memorable fingerprints and this work is one such step towards achieving this goal.



(a) Launcher



(b) Map

Figure 4.13: User interface variants used in Android user study

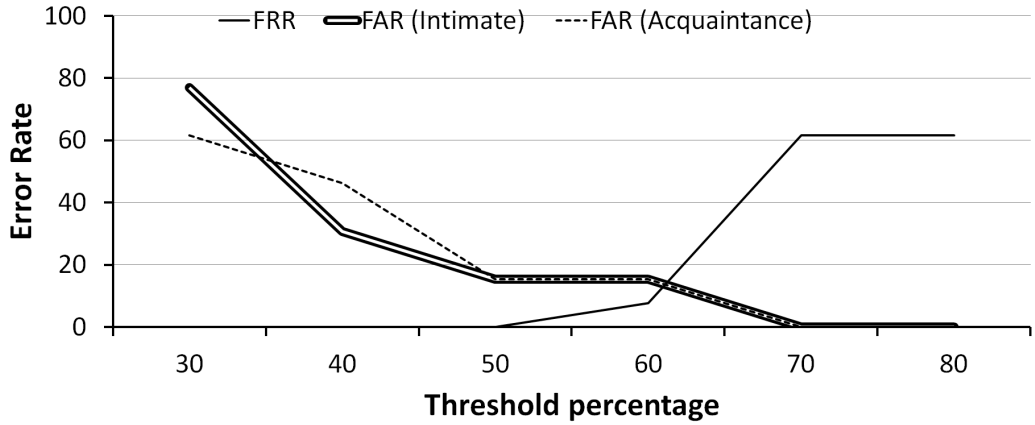


Figure 4.14: Android - false acceptance & false rejection rates

The accuracy of the launcher UI was 57% for participants, compared to 7% and 0% for intimates and acquaintances respectively. This large gap between the participant's and their intimate and acquaintance's results suggests that launcher data could be considered to generate good fingerprints and left as a future work.

The accuracy of the map UI was 38% for participants, 38% for intimates, and 41% for acquaintances. Hence, even with richer location data, the *where*-type fingerprints do not perform well — indicating, perhaps, that *where*-type rules are not ideal for generating fingerprints.

4.7 Discussion

In this section, we discuss some lessons that we have learned in the generation of memorable fingerprints (see Section 4.7.1). HuMan uses users' personal and private information in generating fingerprints from the cellphone usage thereby leading to privacy concerns. We discuss some of the key issues and how HuMan can mitigate this to a certain extent (see Section 4.7.3). We discuss some of the limitations of the user study in Section 4.7.4.

4.7.1 Characteristics of memorable fingerprints

Our user studies were a great learning process to allow us to understand the characteristics of memorable fingerprint.

Broad Range of Events Necessary Symbian users hardly used their phones for anything but SMSes and calls, from which very few memorable signatures could be constructed. On the other hand, Android provided more event types including applications, emails, which allowed us to recover more memorable fingerprints.

Find the Most Confident Events Behavioral fingerprints are attached with statistical notions of support and confidence (or likelihood of the rule's pre-condition being followed by its post-condition). The Symbian study used rules with more than 50% confidence whereas the Android study used rules with more than 70% confidence. As the participants in the Android study performed better than those in the Symbian study, it suggests that good fingerprints have higher confidence.

Use the Most Memorable Templates We quickly realized that certain events are more memorable than others. We categorize our templates based on the type of information they contained, i.e., "who", "what", etc. In our experiments, we consistently found that the templates containing "what" and "who" types were more memorable. We also found that certain special types of fingerprints performed well, e.g., those representing negative rules. We also found that "when" and "who" types were the least memorable and should not be used in generating authentication challenges.

4.7.2 Strength of fingerprints

In this section, we discuss the secrecy information inference attack on the fingerprints generated from the user's cellphone usage data and describe how HuMan is resistant to such attacks.

Not all secrecy information i.e. fingerprints have the same measure of strength. Some information may leave the smartphone while either uploading geo-tagged photos to Facebook or public check-ins on Foursquare. Such activities will reveal user's location to the public. Moreover, it could be possible that a human observer might be continuously keeping a note of the information entered by the user e.g. to whom the SMS has been sent, where the call has been made from, the duration of the call etc. Such information used in the generation of fingerprints/authentication challenges become susceptible to secrecy information inference attack and should be avoided. We believe, if the information which can be inferred from the public sources should not be used in generating challenges and should be flagged as weak fingerprints.

HuMan generates fingerprints based on multiple attributes unlike Blue Moon^{TM} which only uses the likes/dislikes of the user to create authentication challenges. For example a challenge generated from HuMan is in the form of "Whom do you usually call from your home?" and response could be "David". As we can notice that there are multiple attributes attached with this challenge like the activity 'call', the callee 'David' and the location 'home'. A single location information leak during the activity period may not tell much about the actual activity performed. This implies that the fingerprints generated from HuMan are more resistant to secrecy information inference attack. Moreover, we have not observed during our user study any source of information which is being posted in public and also found in our fingerprints generated from HuMan. However, we leave this as a future work to explore more.

4.7.3 Security and privacy issues

HuMan collects a lot of personal information, therefore, there exists some security and privacy issues in generating memorable fingerprints from cellphone usage data.

1. To minimize the privacy risk as much as we can, HuMan masks out as much

critical information as possible. For example, the content of SMS and email messages are *not logged*.

2. During authentication, each question is followed by some choices. Even without knowing anything about the user an attacker can learn something about her while looking at the question and different choices. To mitigate this attack, we always provide a choice "None" as one of the available possible choice. This can provide some protection against not so sophisticated attackers.

3. HuMan collects a lot of personal and private information to generate memorable fingerprints, however, the current model is not designed to provide any protection against attackers who has the access to the raw data stored on the physical device or provider-level attacks where service provider has access to most of the information.

4.7.4 Limitations

Through our exploration with the fingerprints, we believe that our user study provides a good test on the memorability of fingerprints generated by HuMan and we also note the limitations discovered through this exploration.

- *Trade-off between Power/Performance.* There was an inevitable minor issue on Android with regard to the tradeoff between the slight lag in performance and power drain due to the increase logging of more data. Some participants expressed unhappiness and we plan on modifying the logger to be more adaptive to user behavior minimizing this trade-off.

- *Number of templates.* We manually crafted the templates and thus the number of templates are currently limited. We plan on adding more in the future.

- *Small number of placeholders.* (typically 1-3) in the templates. With more complex templates, we could generate more context sensitive fingerprints.

- *Authentication.* In authentication scenarios where a system requires frequent authentication, the current version of HuMan may not be the best fit because of moderate accuracy and the time to answer one question (9 seconds on average) as compared to 8.46 seconds on average to enter a 6 digit PIN (based on our tests). However, some authentication scenarios where HuMan is suitable which requires less frequent authentication are 1) when the phone is lost and needs to be locked remotely. 2) to access private and sensitive data. 3) to unlock or change the SIM card and 4) to access the systems folder of the phone.

- *Universality to all users.* Although HuMan was designed to be useful for all types of users regardless of their technical proficiency and literacy levels, our user studies showed that the extent to which this is true depended on the user's cellphone usage pattern (more diverse data usage generated better fingerprints).

4.7.5 Further comments

- *Size of the participant population.* Scholars suggest how one can choose an appropriate participants' sample size [122, 92]. According to the prior research and the studies in HCI field, we believe our participant pool is enough for some preliminary analysis.

- *Adaption to changing lifestyles.* Our system could adapt to changing lifestyles of users as the training could be performed periodically, e.g., once a week. Though only the frequent patterns will constitute the fingerprints.

- *Shoulder surfing attacks.* HuMan is not designed to be resilient against shoulder surfing attacks.

Chapter 5

Coercion attack in biometric key generation

5.1 Introduction

Many techniques have been proposed to generate strong cryptographic keys for secure communication and authentication. Some of these techniques, e.g., those using biometrics [58, 116, 119, 120, 53], offer desirable security properties including ease of use, unforgettability, unforgeability (to some extent), high entropy and etc. However, most of these schemes are not resistant to coercion attacks in which the user is forcefully asked by an attacker to reveal the key [130]. When the user's life is threatened by an attacker, one would have to surrender the key, and the system will be compromised despite all the security properties described above. This is an example of an extreme form of human factor exploitation to gain access to system. In this Chapter, we present a novel approach to protection against coercion attacks in generating keys.

For a cryptographic key generation technique to be coercion attack resistant, it is required that when the user is under coercion, he/she will have no way of generating the key, or the key generated will never be the same as the one generated when he/she is not being coerced. If this requirement is met, then an adversary would not

apply any threat to him/her because the adversary understands that the user would not be able to generate the key when he is threatened to do so. Here we assume that the coercion resistance property is publicly known to everyone, including the attackers; otherwise it might lead to a dangerous situation for the user, a problem we do not address in this work.

To show how desirable it is to have a coercion-resistant cryptographic key generation technique, here we list a few scenarios in which such a technique could be useful:

- Bank's vault and safe: According to statistics released by the FBI [82], there were $1,094$ reported robberies (out of which 58 cases were of vault/safe robberies) of commercial banks between July 1, 2009 and September 30, 2009 totaling more than $9.4 million. If such systems are used to fight against these attacks, then managers will never be forced to open the vault.

- Cockpit doors on airliners: The hijackers of the September 11, 2001 use the fueled aircraft as a missile to destroy ground targets. If the cockpit doors on airliners are well equipped with coercion resisted techniques, then hijackers can never force a flight attendant to open the door.

- Secret/capability holders in a war: secret and capability holders would not be forced to reveal the secret or use the capability.

In this work, we explore the incorporation of user's emotional status (through the measure of skin conductance) into the process of key generation to achieve coercion resistance. We demonstrate this possibility by incorporating skin conductance into a previously proposed key generation technique using biometrics [116] (see Figure 5.1).

Incorporating skin conductance information into key generation is nontrivial. First, the fact that a change in a user's emotional status leads to changes in a user's skin conductance does not necessarily mean that our proposed technique is coercion resistant. If known patterns exist in such changes, an attacker might be able to guess

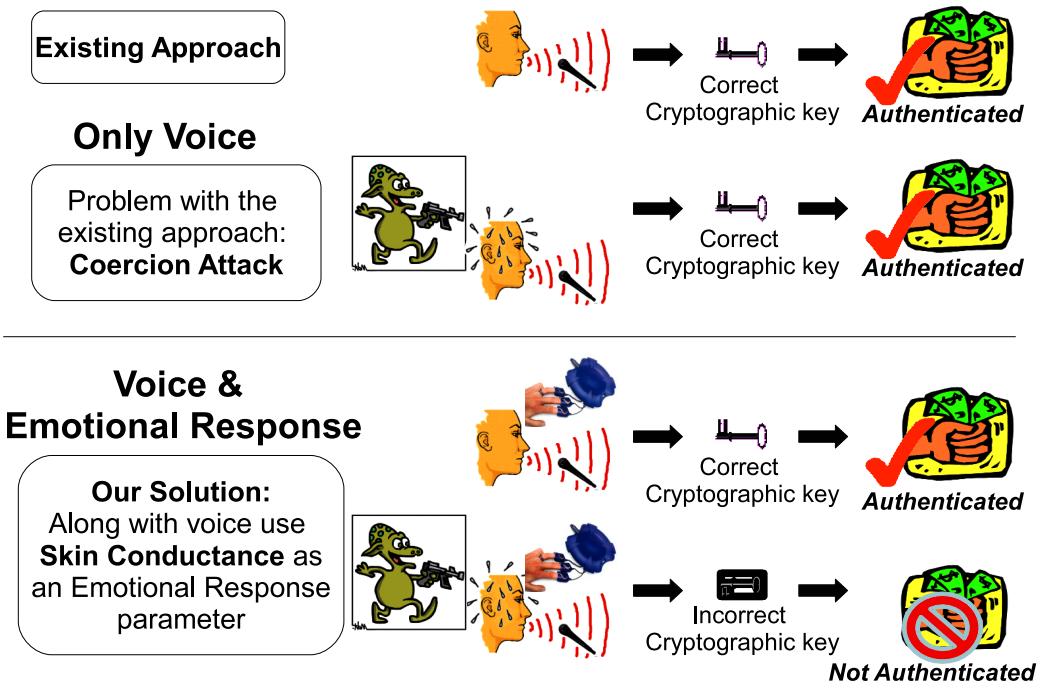


Figure 5.1: Coercion attacks in key generation

the skin conductance of the user when he is not nervous by, e.g., flipping a few bits of the feature key (see Section 5.4) generated from the skin conductance of the user when he is nervous. We analyze this attack and its consequences, and show that the reduction in password space is small.

Second, we hope that the key generation algorithm will take in the least amount of user specific information except the live data collected when it is used. This is because the key generation algorithm might be executed from the client's machine, and the inputs to the algorithm could potentially be retrieved by the attacker during a coercion attack. However, when dealing with biometrics data, removing such user specific information from the inputs of the algorithm is not plausible, as different people have different sets of consistent and inconsistent biometric features. The algorithm would have too high false rejection rates without this additional user specific information. We propose using only user-specific feature lookup tables which contain valid key shares or garbage. We also analyze conceivable attacks that result from our proposal.

Third, it is nontrivial how a user study can be performed to evaluate our technique. We need to collect biometric data corresponding to different emotional states

of real human beings. Efforts in this area are more demanding than traditional efforts to get pattern recognition data [129]. To analyze the effectiveness of our proposal, we perform a user study to see how one's skin conductance changes when he/she is being coerced. This is used to evaluate the false acceptance and false rejection rates of our model, and to analyze the attacker's strategy in guessing the cryptographic key. With 39 participants in our user study, we find that our technique enjoys moderate false acceptance and false rejection rates in key generation. Furthermore, we find that the reduction in the password space for an informed attacker is small.

The rest of the Chapter is organized as follows. We discuss the related work in Section 5.2. Background knowledge about the chosen biometrics and fingerprint are discussed in Section 5.3. In Section 5.4, we present the details of our approach in key generation using skin conductance and voice. The user study and results are presented in Sections 5.5 and Section 5.6 respectively. We discuss some of the advanced attacks and limitations in Section 5.7.

5.2 Related work

In this section, we review some of the techniques and methodologies used to generate cryptographic keys from biometrics and some previous work on the emotion recognition schemes using physiological signals.

Many key generation techniques from biometrics, e.g., voice, iris, face, fingerprints, keystroke dynamics, and etc., have been proposed in the last decade [58, 116, 119, 120, 53]. The pioneer work in cryptographic key generation from behavioral biometrics uses keystroke dynamics of a user while typing the password [117]. The features of interest are the duration of keystrokes and the latency between each pair of keystrokes. The generated cryptographic key is called the hardened password. However the password generated is not very long and is susceptible to brute-force attacks [117]. Another method using secret sharing was proposed to generate the

biometric key from voice [116]. The distinguishing biometric features are selected based on the separation between the authentic and the imposter data, and then binarized by some thresholds. However, this method is not resistant to coercion attacks (which our proposed model trying to target), as the attacker can force the user to speak out the password in a normal way. We will discuss key generation approach from voice in more detail in the formal framework of our model (see Section 5.3).

Another work on key generation from voice uses phonemes instead of words, as it is possible to generate larger keys with shorter sequences [58]. Using the information of the voice model and the phoneme information of the segments, a set of features are created to train an SVM (Support Vector Machine) that could generate a cryptographic key. False-positives and entropy of the system were not demonstrated, which does not give a clear picture of the security of the scheme.

There are many risk and security concerns over biometric systems [130, 132, 153]. Some of the threat models include fake biometrics at the sensor, tampering with the stored templates, coercion attacks. Biometrics liveness detection is proposed to thwart fake biometrics attacks, e.g., by using perspiration in the skin [1] or blood flow [104]. However, no previous work has been proposed to resist coercion attacks in generating cryptographic keys using biometrics. There have been suggestions like panic alarm or duress code to fight against coercion attacks, but they are different from what we are proposing here because in previous schemes users *choose* not to generate the key but to send a signal to authorities without catching the adversary's attention, whereas in our scheme we require that users simply will not be able to generate the key. It is clear that our scheme offers much stronger security properties.

Previous work also shows that emotion recognition using physiological signals, affects from speech, and facial expressions have various success rates between 60% and 98% [129]. Although many techniques have been proposed for emotion recognition [129, 99, 121, 100], none has looked into the incorporation of emotional status into key generation as what we propose in this work.

5.3 Background

In this section, we present some background knowledge of voice and skin conductance, and discuss why in future an addition of fingerprint in our model would be better as an authentication measure for the protection against coercion attack. We also discuss the reasons for the selection of these features and the advantages over others in terms of acceptability, feasibility and usability.

5.3.1 Why skin conductance?

An emotion is a mental and physiological state associated with a wide variety of feelings, thoughts, and behavior. Emotions are subjective experiences, often associated with mood, temperament, personality, and disposition [42]. This emotional behavioral change is the key component in our model in fighting against coercion attack. Several physiological peripheral activities have been found to be related to emotional processing of situations. Many physiological parameters were studied for emotion recognition, e.g., heart beat rate [6] (HR), skin conductance [106] (SC), EMG (Electromyography) signals, ECG (Electrocardiography) signals, body temperature, BVP (Blood Volume Pulse) signals, and etc., among which HR and SC are especially attractive due to their strong association with behavioral activation system (BAS) and behavioral inhibition system (BIS) respectively [52].

SC is the change in the electrical properties of an individual person's skin caused by an interaction between environmental events and the individual psychological state. Human skin is a good conductor of electricity and when subject to a weak electrical current, a change in the skin conductance level occurs [158]. We chose SC over HR for the following reasons.

1. The skin conductance is one of the fastest responding measures of stress response [74]. It is one of the most robust and non-invasive physiological measures of autonomic nervous system activity [26]. Researchers have linked skin conductance response to stress and autonomic nervous system arousal [142].

2. The change in HR not only accounts for stress but for many other reasons, including jogging or doing some heavy work load. SC, on the other hand, has been shown to be a promising measure in experimental studies [140] for its reliability.

3. According to [154], HR is also impacted when stress levels rise but the shifts take a bit of time to happen and by the time the changes are noticeable the triggering stimulus is long past, whereas SC responses are rapid and easy to measure.

4. HR is not suitable to our model due to prevailing feasibility issues. HR can be measured using an Electrocardiogram (ECG) machine or a stethoscope. Using an ECG machine is impractical because it is very cumbersome due to many (at least three) electrodes required and installation costs [21]. Stethoscope is not good either because different placements of the stethoscope could lead to high FTC rate (failure to capture rate) [128].

5. Using SC has an extra advantage as it can be measured simultaneously while fingerprints are being scanned. This ensures that SC is measured from the authentic person (more on this in the coming subsection). The wide acceptance of finger scanning [85, 152] also suggest that SC measurement would have the potential to gain user acceptance.

There are some limitations of using skin conductance as with any other biometric. Some skin lotions can be used to manipulate the skin conductance level. In a test done by [136], the usage of specific solutions produced significant increase in skin water content, and was indicated by increase in skin conductance level. According to the product after the application of the cream by EncoSkin, skin moisture level can be significantly increased which can be monitored by skin conductance [43].

5.3.2 Why voice?

Voice has been used previously to generate cryptographic keys [58, 116]. Voice as a biometric is desirable for generating keys for two important reasons. First, it is the most familiar way of communication, which makes it ideal for many applications. Second, voice is a dynamic biometric and is not static like iris or fingerprint. A user can have different keys for different accounts by just changing the password (what to pronounce) or the vocalization of the same password (how to pronounce) to generate different cryptographic keys. In an event of key compromise a new cryptographic key can be easily generated. Note that voice has a potential disadvantage when used in fighting against coercion, namely that the attacker may blame the user for intentionally pronouncing the wrong password. We demonstrated our technique with voice; however, our scheme is not limited to using voice, other biometric can be used as well.

5.3.3 Why fingerprint?

A potential threat to our biometric system is to use spoken password from the genuine user *(under stress)* and SC responses from another person *(normal emotional state)*. To ensure that SC is not unforgeable, one can make use of a device to collect fingerprint and skin conductance of the user at the same time so that the fingerprint of the user can be checked and mapped to his/her skin conductance signal. However, we did not demonstrate how to use this as a measure in our proposed model as this is not the contribution of this work and is left for the future work.

5.4 Key generation from voice and skin conductance

In order to show how skin conductance can be used to fight against coercion attacks in cryptographic key generation, in this section, we present the details of a cryptographic key generation technique using voice and skin conductance. Note

the criteria behind choosing skin conductance and voice in Section 5.3. Other biometrics in lieu of voice could be used as well. Our way of using voice is similar (with some differences) to an earlier proposal of generating cryptographic keys using voice [116]. Table 5.1 shows some notations used in the rest of this Chapter.

General Notations		**Notations related to Spoken Password**	
$\mathcal{K}$	cryptographic key	V	Voice
C	a set of centroids	N_V	# samples in V during training
c	a centroid in C	f_V	frame vector
m	$m = m_V + m_{SC}$	ϕ_V	feature descriptor
		n	number of frames
		T_V	lookup table generated using V
		m_V	total bits in a feature descriptor of V
		b_V	feature key using V
		s	number of segments
		R	segment vector

Notations related to Skin Conductance	
SC	Skin Conductance
N_{SC}	# samples in SC during training
ϕ_{SC}	feature descriptor
f_{SC}	vector containing sampled values of SC
ℓ	number of frames
T_{SC}	lookup table generated using SC
m_{SC}	total bits in a feature descriptor of SC
b_{SC}	feature key using SC

Table 5.1: Notations

5.4.1 An overview

Figure 5.2: Input devices

Inputs to our model include the voice captured when the user utters the password into the microphone and the skin conductance measured. Figure 5.2 shows the input devices used in our experimental setup. Output of our model is a cryptographic key generated.

In the first phase (Figure 5.3 (a)–(h)), features extracted from the spoken password are used to generate a sequence of frames $f_V(1), \ldots, f_V(n)$ (5.3 (c)), from which an optimal segmentation of s segments (component sounds) (5.3 (f)). The segmentation obtained are then mapped to the feature descriptor using a random α_V plane (5.3 (g)). Furthermore, features are also extracted from the SC sample and the corresponding feature descriptors are computed (5.3 (h)). These feature descriptors should be "sufficiently similar" for the same user and "sufficiently different" for different users. By the end of the first phase, we have feature descriptors for both voice and SC signal.

In the second phase (Figure 5.3 (i)–(l)), we perform lookup table generation and cryptographic key reconstruction. A total of N_V samples from voice and N_{SC} samples from SC are used to generate lookup tables T_V and T_{SC}. In cryptographic key reconstruction, feature keys are generated from the spoken password (m_V bits) and SC (m_{SC} bits). The two lookup tables generated and the features keys are then used to generate the cryptographic key.

In the next two subsections, we will present these two phases in more detail.

5.4.2 Phase I: Feature descriptors derivation

Feature descriptors from voice

In the last six decades, speech recognition and speaker recognition have advanced a lot [27]. A speaker recognition system usually has three modules: feature extraction, pattern matching and decision making, among which feature extraction is especially important to our research as it estimates a set of features from the speech signal that represent the speaker-specific information. These features should be

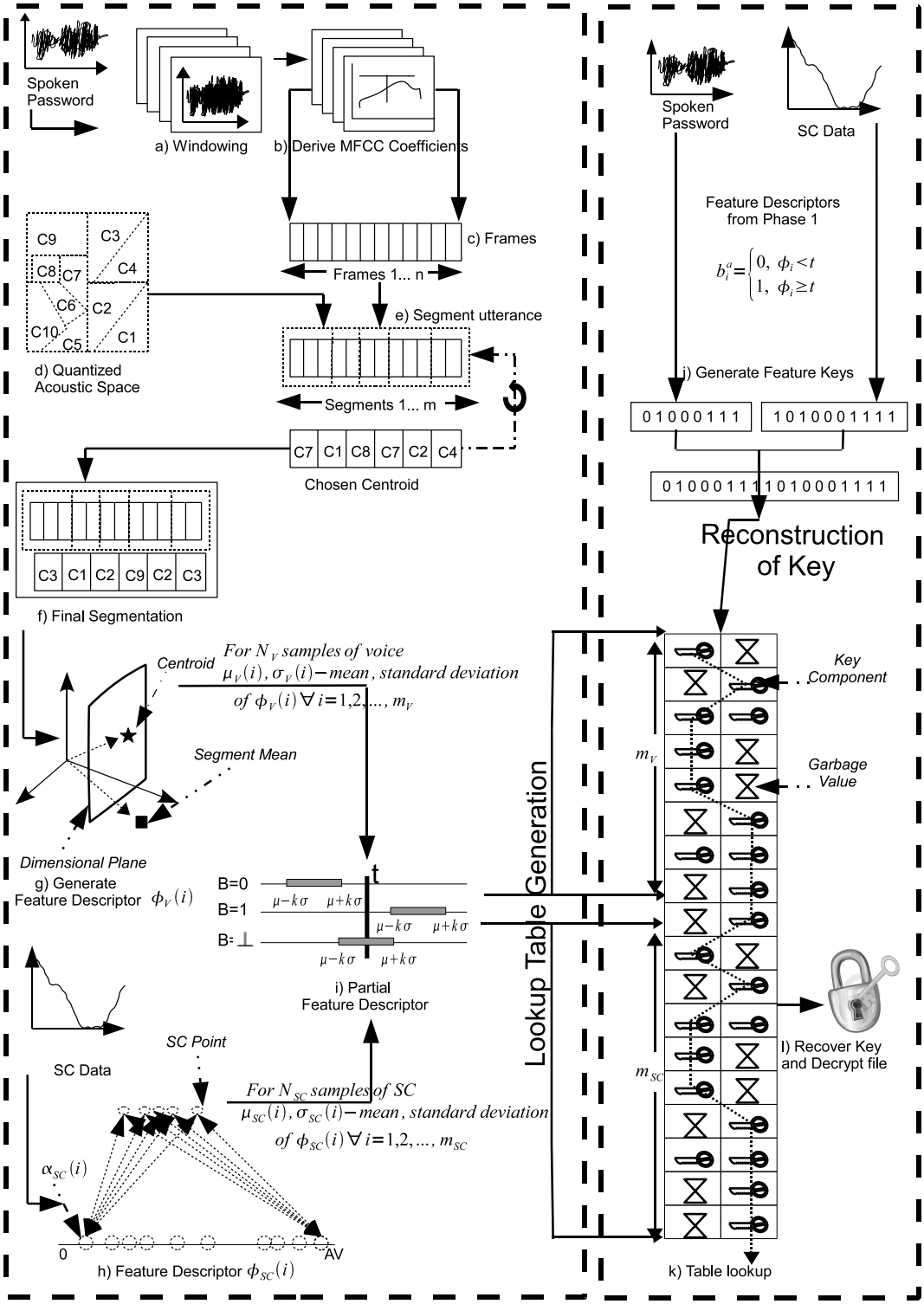


Figure 5.3: Design overview, refer to Section 5.4.2 for detailed description

consistent for each speaker and should not change over time. The way we extract these features and derive the feature descriptors is very similar to the previous approach [116], except that we use the Mel-frequency Cepstral Coefficients (MFCCs) instead of linear cepstrum [116]. MFCC has advantages over linear cepstrum that the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the linear cepstrum [47].

Associating centroids to the acoustic model We convert the raw speech signal into a sequence of acoustic feature vectors in terms of the Mel-frequency Cepstral Coefficients (MFCCs) [155]. In the next paragraph we provide a short description on the extraction of MFCC (see Figure 5.4).

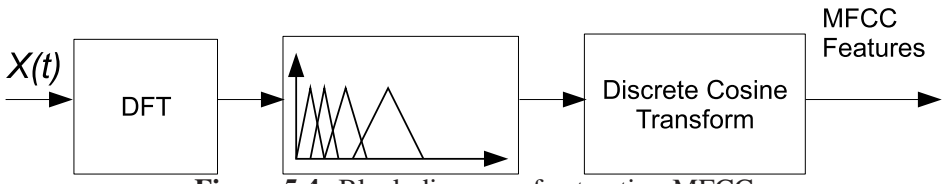


Figure 5.4: Block diagram of extracting MFCC

The voice signal is first divided into blocks of 20 to 30 msec (see Figure 5.3(a)), and Discrete Fourier Transform (DFT) is performed to obtain the frequency representation of each block. The neighboring frequencies in each block are grouped into bins of overlapping triangular bands of equal bandwidth. These bins are equally spaced on a Mel-scale instead of a normal scale as the lower frequencies are perceptually more important than the higher frequencies. The content of each band is now summed and the logarithmic of each sum is computed. To see this effect in time domain, Discrete Cosine Transform is applied to yield a "spectrum like" representation $\psi(t)$ that collectively make up an ***MFC***, and $\psi(1), \ldots \psi(12)$ are called MFCC, where higher order coefficients are discarded. This vector is called a frame (f_V).

We run a sliding window of 30 msec over an utterance to obtain blocks 10 msec apart from one another, and extract the MFCC, $\langle \psi(1), \ldots \psi(12) \rangle$, for each block

(see Figure 5.3(b)). n frames are obtained from utterance of the password (see Figure 5.3(c)). An acoustic model of vectors from a speaker-independent and text-independent database of voice signals is obtained, from which vector quantization is used to partition the acoustic model into clusters (see Figure 5.3(d)). A multivariate normal distribution for each cluster is generated, where each cluster is parameterized by the vector c of a component-wise means (called a centroid) and the covariance matrix Σ for the vectors in the cluster. The density function for this distribution is

$$P(c \mid x) = \frac{1}{(2\pi)^{\delta/2}\sqrt{det(\Sigma)}} e^{-(x-c)^T \Sigma^{-1}(x-c)/2}$$

where δ is the dimension of the vectors. We denote the set of centroids as C.

Segmentation of frames After getting the centroids from a database of speaker-independent voice signals, we try to obtain the transcription, i.e., the starts and ends, of the phonemes of an individual user's utterance.

To do this, we perform segmentation on the spoken password. Let $f_V(1), \ldots f_V(n)$ be the sequence of frames from the utterance, and $F(R_1), \ldots F(R_s)$ be the sequence of s segments (s is a constant and same for all users), where $F(R_i)$ is the i^{th} segment containing the sequence of frames $f_V(j), \ldots f_V(j')$ such that, $1 \leq j \leq j' \leq n$. Intuitively, each $F(R_i)$ corresponds to one "component sound" of the user's utterance.

We did this with an iterative approach (see algorithm 1). Ranges $R_1, \ldots, R_s$ are first initialized to be equally long. We then calculate the matching centroid c for a segment F(R), i.e., the one for which the likelihood of F(R) w.r.t. c is maximum. Dynamic programming is then used to determine a new segmentation for that frame sequence. This process is repeated until an optimal segmentation is obtained, which is mapped to the feature descriptor (see Figure 5.3(e,f)).

Feature descriptor Having derived a segmentation for a spoken password, we next define the feature descriptor (ϕ_V) of this segmentation that is typically the

Algorithm 1 Spoken password segmentation

Segmentation $(f_V(1), \ldots, f_V(n), s)$

1: Score$^{'} \longleftarrow 0$
2: **for** $i = 1$ to s **do**
3: $\quad R_i \longleftarrow \left(\left\lfloor \frac{(i-1) \times n}{s} \right\rfloor, \left\lfloor \frac{i \times n}{s} \right\rfloor \right)$
4: **end for**
5: **repeat**
6: $\quad$ Score $\longleftarrow$ Score$^{'}$
7: $\quad$ **for** $i = 1$ to s **do**
8: $\quad\quad$ **while** $\forall c \in C$ **do**
9: $\quad\quad\quad L(F(R_i)|c) \longleftarrow \prod_{j \in R_i} (f_V(\text{j})|c)$
10: $\quad\quad$ **end while**
11: $\quad\quad c(R_i) \longleftarrow arg \max_{c \in C} \{L(F(R)|c)\}$
12: $\quad$ **end for**
13: $\quad$ let $\bigcup_{i=1}^{s} R_i^{'} \longleftarrow [1, n]$
14: $\quad$ Score$^{'} \longleftarrow \prod_{i=1}^{s} L(F(R_i^{'}|c(R_i)))$
15: $\quad R_i \longleftarrow R_i^{'}$
16: **until** Score$^{'}$ - Score < Δ

same when the same user speaks out the same utterance. To do this, we use a fixed vector α_V, and define the i^{th} bit of the feature descriptor as (see Figure 5.3(g))

$$\phi_V(i) = \alpha_V.(\mu_V(R_i) - c(R_i)), \quad \forall \quad 1 \leq i \leq s$$

That is, we normalize $\mu_V(R_i)$ with $c(R_i)$ and let ϕ_V(i) be the linear combination of components in it as specified by α_V. This process results in a feature descriptor (ϕ_V), where N_V feature descriptors are then generated from N_V voice samples and used to generate a lookup table T_V (in Phase II).

Feature descriptor from skin conductance

When some external or internal stimuli occur that makes a person stressed, the skin becomes a better conductor of electricity. This conductance can be measured between two points on the body (e.g., two fingers) and the level of electrical conductance is called skin conductance. Since we want to detect changes in the emotional

status of a person, we record skin conductance over a time period.

SC signal was measured with our device and sampled at a frequency of 30 samples per second. Let $f_{SC}(1), \ldots, f_{SC}(\ell)$ denote the sampled values obtained from the SC signal. We model the feature values into a feature descriptor (ϕ_{SC}) in a similar way as we did in the processing of voice. We choose a random vector α_{SC}= $[\alpha_{SC}(1), \alpha_{SC}(2), \ldots, \alpha_{SC}(m_{SC})]$ (m_{SC} is a constant), and use the Euclidean distance between all the points of the α_{SC} vector and f_{SC} to compute the distance measure M and henceforth the feature descriptor (ϕ_{SC}).

$$M(i,j) = \alpha_{SC}(i) \times f_{SC}(j) \quad \forall \quad 1 \leq i \leq m_{SC}, 1 \leq j \leq \ell$$

ϕ_{SC} is the mean of all the distance measures for each $\alpha_{SC}(i)$ values (see Figure 5.3(h)), i.e.,

$$\phi_{SC}(i) = \frac{1}{\ell} \sum_{j=1}^{\ell} M(i,j) \quad \forall \quad 1 \leq i \leq m_{SC}$$

Note that the upper bound of $\alpha_{SC}(i)$ needs to be carefully chosen to maintain a good entropy on the feature descriptor of different people. Also note that we do not store skin conductance information directly but rather the feature descriptor generated from the distance measure is stored (same as in the case of voice). N_{SC} feature descriptors are derived from N_{SC} SC samples and then are used to generate a lookup table T_{SC} (in Phase II).

5.4.3 Phase II: Lookup table and cryptographic key generation

We explain how we obtained the feature descriptors from voice and skin conductance in the previous subsection. Here, we will explain how we constructed lookup tables (training of the model) and obtained the cryptographic keys from the tables (usage of the model). The basic idea is that each entry of the lookup tables contains a share of the correct key or some garbage value, and the feature descriptor is used

to determine the corresponding entry from the lookup table. In the end, the shares from the lookup tables are used to reconstruct the key.

Lookup table generation

Intuitively, if a feature descriptor is the same as the one recorded previously (i.e., in training), then the system should choose the correct key share from the lookup table, or the garbage otherwise. In order to tolerate some small deviation of a user's utterance and skin conductance, we calculate the mean (μ_{ϕ_V}(i), $\mu_{\phi_{SC}}$(i)) and standard deviation (σ_{ϕ_V}(i), $\sigma_{\phi_{SC}}$(i)) of each feature descriptor over N_V, N_{SC} training samples, and define the partial feature descriptors B_V, B_{SC} as

$$B_V(i) = \begin{cases} 0, \text{ if } \mu_{\phi_V}(\text{i}) + k\sigma_{\phi_V}(\text{i}) < t_V \\ 1, \text{ if } \mu_{\phi_V}(\text{i}) - k\sigma_{\phi_V}(\text{i}) > t_V \qquad \forall\, 1 \le i \le m_V \\ \perp, \text{ otherwise} \end{cases}$$

$$B_{SC}(i) = \begin{cases} 0, \text{ if } \mu_{\phi_{SC}}(\text{i}) + k\sigma_{\phi_{SC}}(\text{i}) < t_{SC} \\ 1, \text{ if } \mu_{\phi_{SC}}(\text{i}) - k\sigma_{\phi_{SC}}(\text{i}) > t_{SC} \quad \forall\, 1 \le i \le m_{SC} \\ \perp, \text{ otherwise} \end{cases}$$

for some threshold t_V and t_{SC} respectively (see Figure 5.3(j)). This phase is the training phase in our model. Here k is a parameter to acquire a tradeoff between security and usability. With the increase in value of k, the user has better chance to generate the key successfully, but will hamper the security of the scheme. More precisely, the increase in the value of k will increase the false acceptance rate and decrease the false rejection rate (as shown in our results in the evaluation Section 5.6).

The idea of defining the partial feature descriptor in this way is illustrated in Figure 5.5 (where the set $\{B, \mu, \sigma, t\}$ is replaced by $\{B_V, \mu_{\phi_V}, \sigma_{\phi_V}, t_V\}$ for voice and $\{B_{SC}, \mu_{\phi_{SC}}, \sigma_{\phi_{SC}}, t_{SC}\}$ for skin conductance). If the i^{th} feature descriptor is

consistently same i.e. $\mu(i) + k\sigma(i) < t$ (the first case in Figure 5.5), then there is a high probability that the value of the i^{th} feature descriptor will be less than *t* during key reconstruction. Therefore, we can let the cell $T(i, 0)$ of the lookup table contain a valid share of the key (and let $T(i, 1)$ contain random bits). If the i^{th} feature descriptor is consistently different, i.e. the value of the feature descriptor is unreliable (when compared to the threshold t as in the third case in Figure 5.5), we let both $T(i, 0)$ and $T(i, 1)$ contain valid shares (typically different). Unlike [116], lookup tables are not encrypted (for discussion on this, see section 5.4.4).

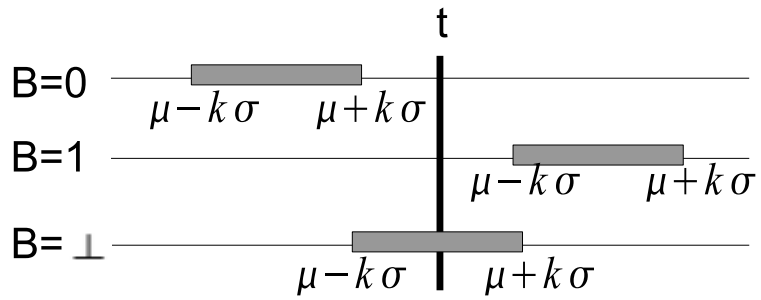


Figure 5.5: Definition of partial descriptor

Having valid shares in both $T(i, 0)$ and $T(i, 1)$ leads to different key shares used and consequently different keys being generated, which might not be desirable in systems that require a unique key. To solve this problem, a random cryptographic key $\mathcal{K}$ (unique for each user) is first generated, which is then encrypted with all possible valid keys (K_{H_i}) that can be derived from <$T_V \| T_{SC}$>. The key generation template therefore comprises of key $\mathcal{K}$ encrypted with $Z = |K_{H_i}|$ derived keys and the lookup tables <$T_V \| T_{SC}$>. Thus, the template =

$$\left\langle \langle T_V \mid T_{SC} \rangle, \left\langle E_{K_{H_1}}(\mathcal{K} \| \hat{r}), E_{K_{H_2}}(\mathcal{K} \| \hat{r}), \ldots, E_{K_{H_Z}}(\mathcal{K} \| \hat{r}) \right\rangle \right\rangle,$$

where $E_{\mathcal{K}_{H_i}}(msg)$ is a publicly known encryption algorithm and $\hat{r}$ is a unique string associated to each user which helps us to determine whether the decryption is correct or not in section 5.4.3.

Cryptographic key reconstruction

When a user tries to reconstruct the cryptographic key, he/she first presents his/her spoken password and the skin conductance. The model collect this information, extracts the features and generates the feature descriptors for both voice and the SC. Corresponding shares from the lookup tables are chosen based on the feature descriptors.

$$b_V(i) = \begin{cases} 0 & \text{if } \phi_V(i) < t_V \\ 1 & \text{otherwise} \end{cases} \quad \forall \quad 1 \leq i \leq m_V$$

$$b_{SC}(i) = \begin{cases} 0 & \text{if } \phi_{SC}(i) < t_{SC} \\ 1 & \text{otherwise} \end{cases} \quad \forall \quad 1 \leq i \leq m_{SC}$$

For example, if the feature descriptor $\phi_{SC}(i)$ is less than the threshold t_{SC}, then $b_{SC}(i) = 0$ and $T_{SC}(i, 0)$ is chosen from T_{SC} as a key share; otherwise $b_{SC}(i) = 1$ and $T_{SC}(i, 1)$ is chosen (see Figure 5.3(i)). b_V and b_{SC} are the feature keys and are obtained from voice and SC respectively.

A key K' is derived by concatenating the key shares (see Figure 5.3(k)). This derived key is then used to decrypt the $|K_{H_i}|$ encrypted keys stored in the template. If the decryption succeeds (by matching the released $\hat{r}$ and the stored $\hat{r}$), then the key $\mathcal{K}$ is released.

$$\mathcal{K}_D = \begin{cases} D_{K'}(E_{K_{H_i}}(\mathcal{K} \| \hat{r})), & if \quad K' = K_{H_i} \\ Random, & if \quad K' \neq K_{H_i} \end{cases}$$

where, $D_{K'}(msg)$ is a publicly known decryption algorithm.

5.4.4 Discussions

While we try to use the consistency of voice and skin conductance to generate the correct key only when it is the genuine user in the normal emotional state, the inconsistency of voice and skin conductance poses challenges too. Voice produced and skin conductance measured of the genuine user in a non-stressed emotional status might change due to tiredness, illness, noise, and etc.

We used an error correction technique, in particular, hamming distance, to improve the usability of the scheme. ${}^{m}C_{d}$ different keys are derived from any freshly generated key K' obtained from the feature descriptors and T (similar to the one derived in section 5.4.3), which are d distance away from the derived key K'. All of these ${}^{m}C_{d}$ keys are then used to decrypt the encrypted keys before giving any negative answer to the user. If the decryption succeeds then the key $\mathcal{K}$ is released. For example, if $d = 2$ and length of the key is m, then ${}^{m}C_{2}$ different keys are derived. Thus, $|K_{H_i}| \times^{m} C_2$ decryptions are performed in attempting to recover $\mathcal{K}$.

Another issue concerns the privacy of the biometric data used. Ballard et al. propose using randomized biometric templates protected with low-entropy passwords to provide strong biometric privacy [15]. One can use this in conjunction with our model to provide both coercion resistance and biometric privacy. However, it is unclear whether the use of low-entropy passwords may have a negative impact on coercion resistance since, intuitively, an attacker may blame the user for providing the wrong low-entropy password in a coercion (similar problem discussed in section 5.3.2). We leave this as future work to develop a solution that satisfies both requirements.

5.5 Experimental Setup

We presented our design in generating a cryptographic key using voice and skin conductance in Section 5.4. It is important to test it out with real human beings to evaluate its performance. However, this is difficult as we need to find a way to make

the participants feel stressed or nervous. It is clear that we cannot actually coerce them to do something by, e.g., putting a gun over their heads. Nevertheless, we performed case studies to induce stress on the participants and measure their voice and skin conductance. (IRB approval was obtained from our university before the user study.) We present the experimental setup in this section and the evaluation results and discussion in the next section.

5.5.1 Demographics

Since we were going to induce stress on the participants, we decided to concentrate on the younger generation (undergraduate and graduate students in the age from 18 to 30). We had altogether 43 participants, from which 4 participants detached the sensors from their fingers when they were nervous during the experiment. Therefore, we successfully performed our experiments on 39 participants, out of which 22 were male and 17 were female.

5.5.2 Experimental settings

Participants were asked to sit in a small office where the overhead fluorescent lights were turned off and a dim red incandescent lamp was turned on to reduce the possible electrical interference with the monitoring equipments. The room was air conditioned to approximately 72°F and humidity level was generally dry. This is done in accordance to the variation of skin conductance in different environmental conditions [140].

Skin conductance sensors[1] were attached to the three middle fingers of the participant to record SC (shown in Figure 5.2). The participant was also asked to keep her left hand (with sensors attached) as still as possible to avoid interference from the sensors. Fake heart rate tags were tied to the wrist, which gave an illusion of monitoring the heart rate.

[1] We use a physiological data acquisition device called Lightstone from WildDivine [146].

Initially, there was an incomplete disclosure regarding the purpose and the steps of the study in order to ensure that the participant's responses will not be affected by her knowledge of the research.

5.5.3 Procedure

We ran two experiments (e1 and e2). Each experiment consisted of two parts, where the first parts (e1n and e2n) were conducted when the participants were in a normal (calm) condition, and the second parts (e1s and e2s) were conducted when the participants were stressed.

We ran experiment e1n by

- showing nice (geographical) pictures one after another and short phrases (the spoken password embedded) which are related to the pictures, and asking the participant to read them out;
- showing fake visual heartbeats at a normal rate at the bottom of the screen and correspondingly playing heartbeats sound.

In order to capture the emotional responses in the stress scenario in e1s,

- a frightening horror movie was played, replacing the nice pictures;
- the rate of the heartbeats were gradually increased to induce more stress on the participant;
- the participant was asked to read out some short phrases at the end of each horror scene (rather than along with the video) to avoid distraction.

Similar studies [118, 91] have been performed previously to measure the stress level in users.

In e2, we went a bit further to induce more stress on the participant. Figure 5.6 shows the change in skin conductance in response to different events in e2. During e2, the participant was asked to type a few sentences (e.g., "Work is much more

fun than fun") shown to her in a fixed period of time. She was also warned (prior to the experiment) not to press the "ALT" key on the keyboard, as it would cause the computer program to crash and all data would be lost (event A). We then left the participant alone in the room to continue typing (event B). We configured the computer to restart after 3 minutes irrespective of whether the participant actually touched the "ALT" key or not. The computer would then boot from a USB drive into MS-DOS and display some error messages (event C). This completes the first part of e2, i.e., e2n.

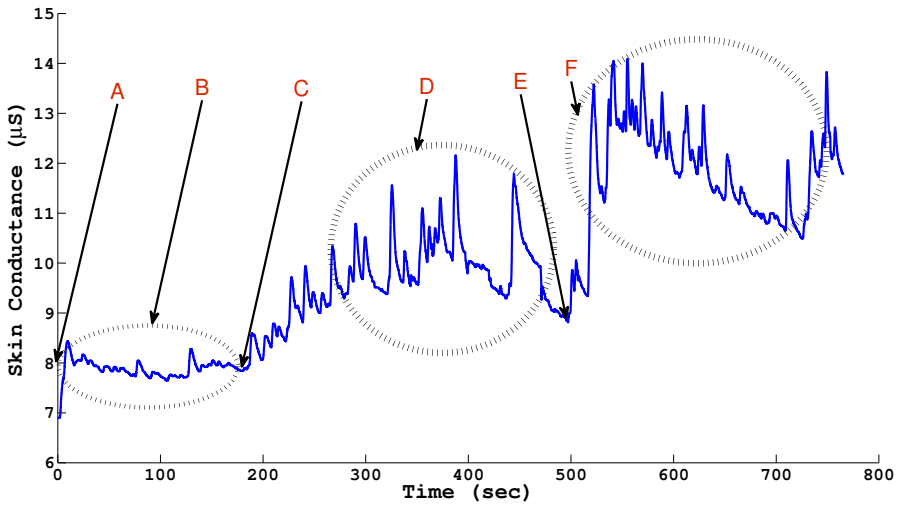


Figure 5.6: Change of skin conductance in e2

Stress started to develop at this point in time as the participant believed that she had pressed the "ALT" key which caused data loss on the computer (event D). We purposely left the participant alone so that stress could develop further and she could not get immediate help to resolve the "problem". After that, the researcher entered the room and examined the keyboard and the computer (event E) and then accused the participant of her negligent act of pressing the "ALT" key (event F). This turned out to be successful in making the participant stressed as we observed that many participants were nervous at this point in time. Some kept saying "sorry"; some tried very hard to fix the "problem", and some started calling for help. There were also voluntary confession statements from the participants, e.g., "I hit the ALT key by mistake in place of typing the 'X' key", "It was a mistake from my side.".

5.5.4 Discussion

In this section, we discuss the difference of the emotional state of a user in real life and in our user study, and limitations of our experiment.

1. Training of the system

    - Real life: the user is in a (controlled) environment specified by our system, in which the stress level is low. This allows us to generate the lookup table for that particular user with the normal skin conductance level.
    - User study: the user is in exactly the (controlled) environment specified by our system, i.e., when watching a relaxation movie.

2. Trying to generate the cryptographic key; no coercion

    - Real life: a user could be in various emotional states, including being happy, sad, angry, etc.
    - User study: same as in training when the user is watching a relaxation movie. In this work, we only try to analyze how our system performs when users are calm and relaxed. It remains future work to analyze how it works when the user is in other emotional states. We do expect the false rejection rate to rise when the user is in other emotional states.

3. Trying to generate the cryptographic key; in coercion

    - Real life: a user can be forced/coerced in many different ways, e.g., a gun to the head, or a knife under the throat, etc.
    - User study: watching a horror movie and being forced to plead guilty (having damaged a notebook computer). We tried our best to approximate the real-life scenarios, but there is a limit we could go when doing this to real human beings (e.g., IRB restriction). However, we believe

that what we did is a clever way of studying human behavior when being coerced.

Discussions above highlight some limitations of our scheme, e.g., we have not tested how it reacts to other emotional status (happy, sad, angry, etc.) and how skin conductance may change naturally (due to oily fingers, etc.). There are two other important limitations in the present study. First, our study does not test the repeatability of using our scheme, i.e., we did not ask the participants to come back and try again. The second limitation comes with the over-controlled environment, e.g., quiet office (because of the use of voice), controlled temperature and humidity [32](because of the use of skin conductance), and etc. It remains further work to test our scheme in different settings.

5.6 Evaluation

In this section, we analyze the data collected in our user study. We first describe how we partition the data into different groups (e.g., for training and test purposes), see Section 5.6.1. We then present a series of analysis on the false acceptance and false rejection rates (Section 5.6.2). Finally we show the change in the password space where an attacker has perfect knowledge of our design and the content stored.

5.6.1 Training and testing datasets

We have collected voice and skin conductance signals for 39 participants. For each participant, we have collected many samples of the signals when the participant is either calm or stressed. Table 5.2 shows the number of samples we collected in each experiment for each participant. Voice signals are typically 2 to 3 seconds long, while skin conductance signals are about 10 seconds long to avoid fluctuations.

Figure 5.7 shows how we obtain dataset to

- split original sample sets $\{\nu_{\mathsf{e1}n}^{\mathsf{full}}, \omega_{\mathsf{e1}n}^{\mathsf{full}}, \omega_{\mathsf{e2}n}^{\mathsf{full}}\}$ into two equal halves $\{\nu_{\mathsf{e1}n}^{\mathsf{train}},$

Feature		e1n	e1s	e2n	e2s
Voice	# of samples	26	5	0	0
	Notation	$\nu^{\mathsf{full}}_{\mathsf{e1}n}$	$\nu^{\mathsf{full}}_{\mathsf{e1}s}$	-	-
SC	# of samples	26	60	18	60-80
	Notation	$\omega^{\mathsf{full}}_{\mathsf{e1}n}$	$\omega^{\mathsf{full}}_{\mathsf{e1}s}$	$\omega^{\mathsf{full}}_{\mathsf{e2}n}$	$\omega^{\mathsf{full}}_{\mathsf{e2}s}$

Table 5.2: Number of samples collected for each participant

$\omega^{\mathsf{train}}_{\mathsf{e1}n}$, $\omega^{\mathsf{train}}_{\mathsf{e2}n}$} and {$\nu^{\mathsf{test}}_{\mathsf{e1}n}$, $\omega^{\mathsf{test}}_{\mathsf{e1}n}$, $\omega^{\mathsf{test}}_{\mathsf{e2}n}$} to obtain datasets for training and testing (see the half circles);

- combine different voice samples and skin conductance samples to create new datasets to test our system (see circles in the middle column). {$\nu^{\mathsf{train}}_{\mathsf{e1}n}$ & $\omega^{\mathsf{train}}_{\mathsf{e1}n}$}, {$\nu^{\mathsf{test}}_{\mathsf{e1}n}$ & $\omega^{\mathsf{test}}_{\mathsf{e1}n}$}, {$\nu^{\mathsf{train}}_{\mathsf{e1}n}$ & $\omega^{\mathsf{train}}_{\mathsf{e2}n}$}, {$\nu^{\mathsf{test}}_{\mathsf{e1}n}$ & $\omega^{\mathsf{test}}_{\mathsf{e2}n}$} are combined to create {$\xi^{\mathsf{train}}_{\mathsf{e1}n}$}, {$\xi^{\mathsf{test}}_{\mathsf{e1}n}$}, {$\xi^{\mathsf{train}}_{\mathsf{e2}n}$}, {$\xi^{\mathsf{test}}_{\mathsf{e2}n}$} respectively.
- to obtain the stress dataset {$\nu^{\mathsf{full}}_{\mathsf{e1}s}$ & $\omega^{\mathsf{full}}_{\mathsf{e1}s}$}, {$\nu^{\mathsf{full}}_{\mathsf{e1}s}$ & $\omega^{\mathsf{full}}_{\mathsf{e2}s}$} are combined to create {$\xi^{\mathsf{full}}_{\mathsf{e1}s}$}, {$\xi^{\mathsf{full}}_{\mathsf{e2}s}$} respectively.

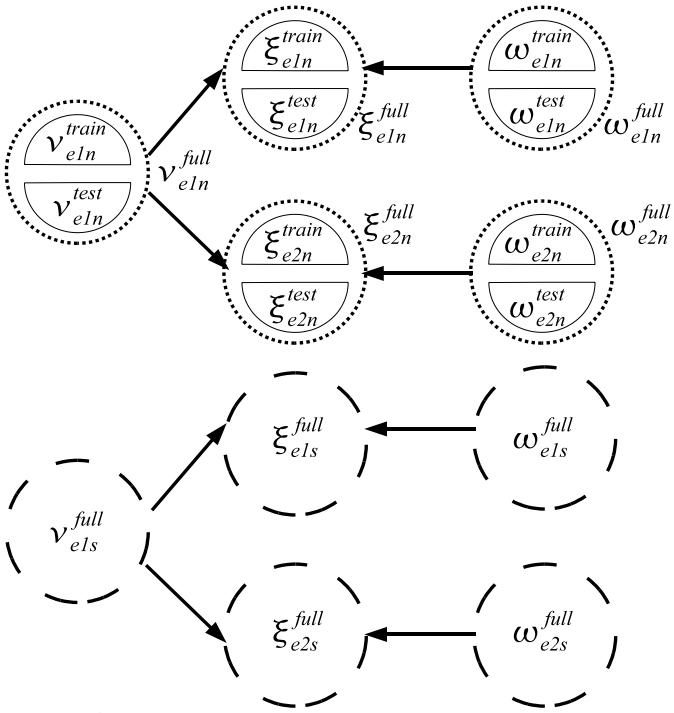


Figure 5.7: Splitting and combining datasets

Note that the voice and skin conductance samples that are combined together might not have been captured at exactly the same time. We allow a time gap because

an attacker might record the voice of the victim to be used in conjunction with the skin conductance of the victim at a slightly different time. Both samples were captured in the same part of the experiment, though, i.e., both from e1s or both from e2s.

5.6.2 Accuracy of our model

The false rejection rate of our system is defined as the percentage of failed login attempts by a legitimate user with her cryptographic key generated, averaged over all users in a population A. Similarly, the false acceptance rate is defined as the percentage of failed detection of attempts by illegitimate users or legitimate users in a stressful situation, averaged over all users in a population A.

Voice samples only We first evaluate the voice samples we collected in our experiments. The purpose is to check out the false acceptance and false rejection rates, in an event if only voice samples are used to generate cryptographic keys. The system is trained with $\nu_{\mathsf{e1n}}^{\mathsf{train}}$ of user u_i, and is tested against $\nu_{\mathsf{e1n}}^{\mathsf{full}}$ of user u_j where $i \neq j, \forall$ $j \in A$ to calculate the false acceptance rates; and against $\nu_{\mathsf{e1n}}^{\mathsf{test}}$ of user u_i to calculate the false rejection rates. Results are averaged on all users in A. We try different random α_V vectors and choose the one that yields the smallest sum of the false acceptance and false rejection rates. We try different settings of the hamming distance parameter d, and find that 2 gives a reasonable tradeoff between false acceptance and false rejection rates. The false acceptance and false rejection rates for different values of k are plotted in Figure 5.8.

Figure 5.8 shows that we manage to get a comparable accuracy with the previous work [116] in terms of the false rejection rate. False acceptance rate was not reported in [116].

Skin conductance only Next, we evaluate the skin conductance samples to see how well they reflect the change in the participants' emotional status. We show the

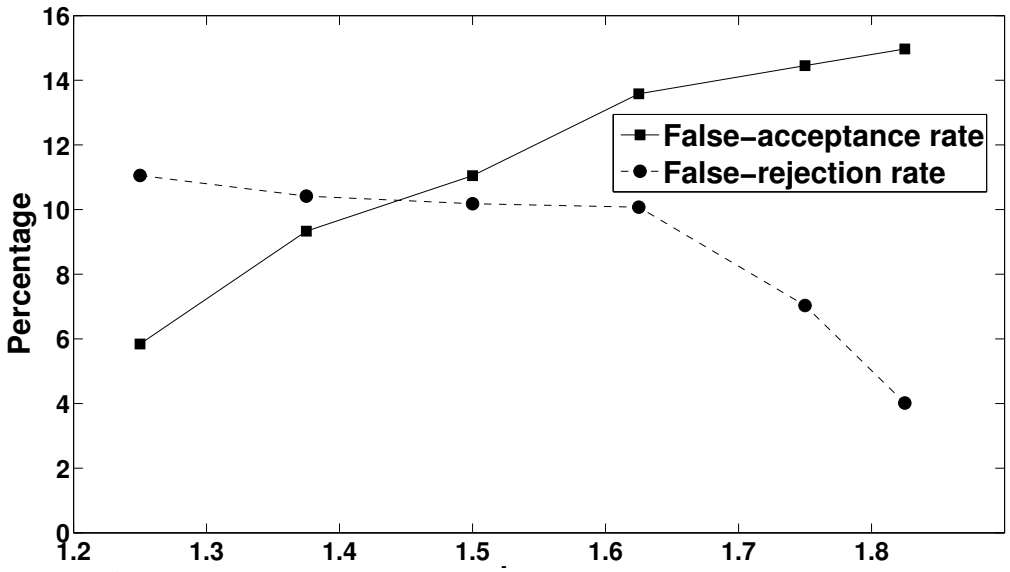
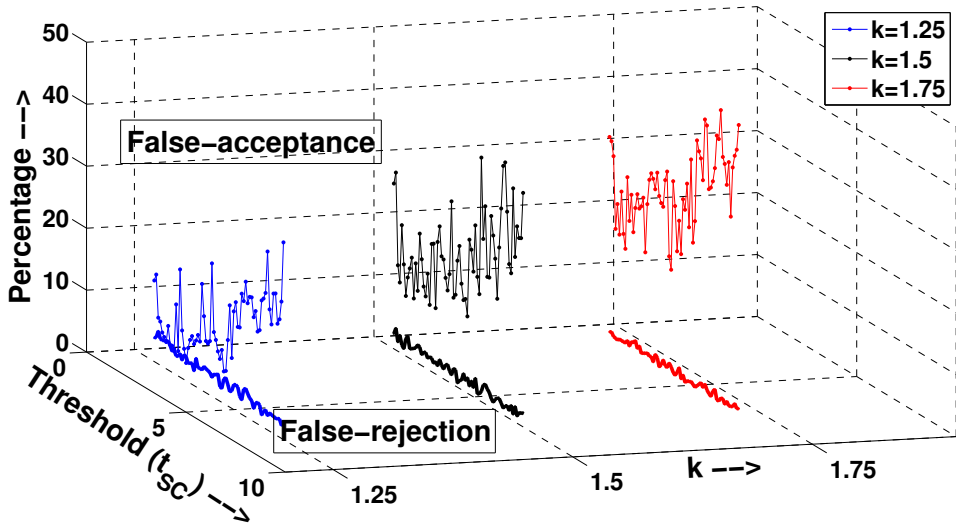


Figure 5.8: False acceptance and false rejection rates for spoken passwords

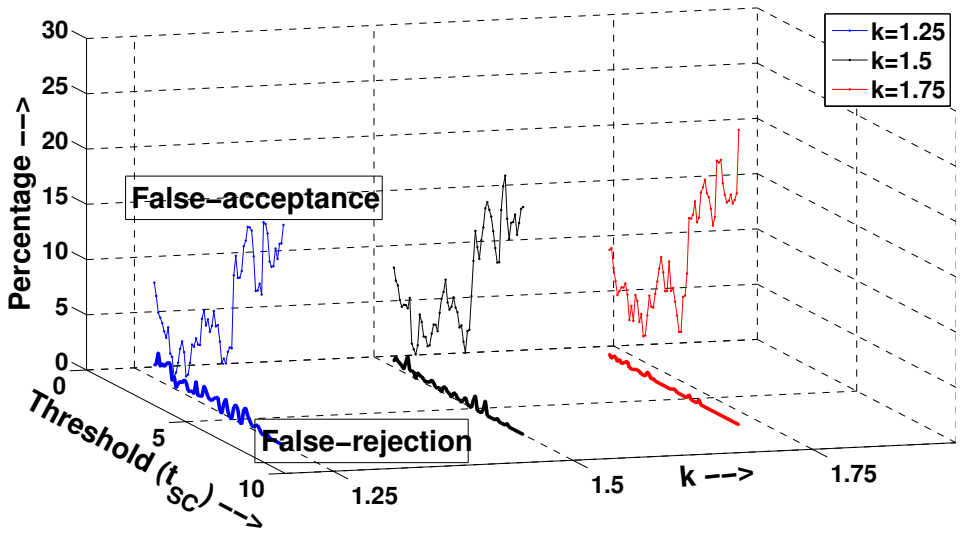
results in Figure 5.9(a) and Figure 5.9(b) for experiment e1 and e2, respectively. The different color lines denotes different 'k' values in Figure 5.9 and Figure 5.10. The system is trained with $\omega_{\mathsf{e1n}}^{\mathsf{train}}$ (and $\omega_{\mathsf{e2n}}^{\mathsf{train}}$, respectively) of user u_i, and is tested against the stressed full data set, $\omega_{\mathsf{e1s}}^{\mathsf{full}}$ (and $\omega_{\mathsf{e2s}}^{\mathsf{full}}$, respectively) of the same user u_i to calculate the false acceptance rates; or against the normal test data set, $\omega_{\mathsf{e1n}}^{\mathsf{test}}$ (and $\omega_{\mathsf{e2n}}^{\mathsf{test}}$, respectively) of the same user u_i to calculate the false rejection rates. Results are averaged over all users in A.

Note that the false acceptance and false rejection rates are higher for e1 in Figure 5.9(a). We believe, this is because of the reason that the intensity of some of the horror videos was not very high, which did not result in a noticeable change in the skin conductance for many users.

We can observe the tradeoff of various settings of k and the threshold from these figures. In general, this shows that whenever a user is under stress, her skin conductance can be used to differentiate between the two emotional state with good accuracy. For example in e2, when $k = 1.25$ and $t_{SC} = 2.1$, we obtained a false acceptance rate of 3.2% and a false rejection rate of 2.2% (see Figure 5.9(b)). If we increase the value of k from 1.25 to 1.75 in both Figures 5.9(a) and 5.9(b), we could see a decrease in the false rejection rates (increasing usability) and increase in the false acceptance rates (compromising with the security). We used the hamming



(a) e1



(b) e2

Figure 5.9: False acceptance and false rejection rates for skin conductance

distance parameter $d = 2$ in our setting.

Voice combined with skin conductance Voice and skin conductance samples are combined as shown in Figure 5.7 to obtain the samples needed in this evaluation. We first train the system with $\xi_{\mathsf{e2n}}^{\mathsf{train}}$, and then evaluate the system against three different datasets to evaluate the false acceptance and false rejection rates.

a $\xi_{\mathsf{e2n}}^{\mathsf{full}}$ of user u_j where $i \neq j, \forall\, j \in A$: when a different person tries to generate the key (Figure 5.10(a));

b $\xi_{\mathsf{e2s}}^{\mathsf{full}}$ of user u_i: when the same user tries to generate the key when she is being coerced (Figure 5.10(b));

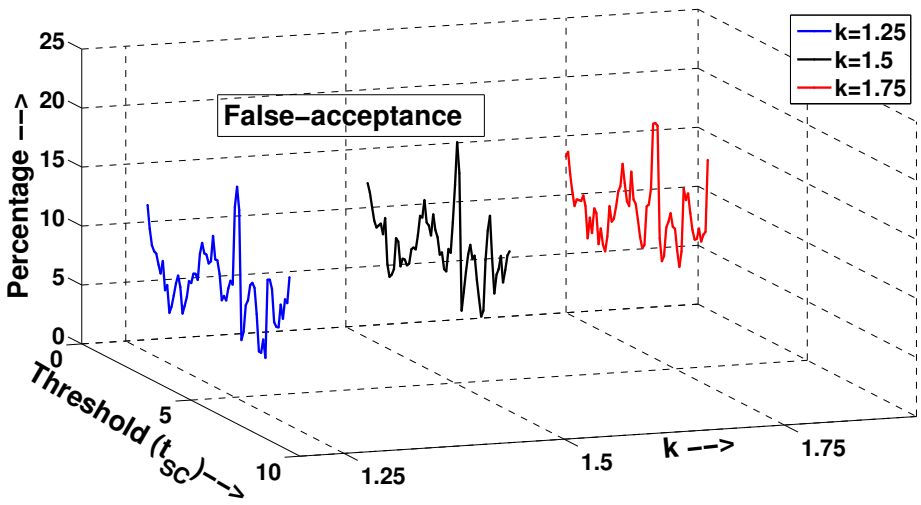
c $\xi_{\mathsf{e2n}}^{\mathsf{test}}$ of user u_i: when the same user tries to generate the key when she is not being coerced (Figure 5.10(c)).

We evaluate the false acceptance rates in the first two cases and the false rejection rates in the third case. Results are averaged over all users in A. We use a hamming distance parameter $d = 4$, and show the results in Figure 5.10.

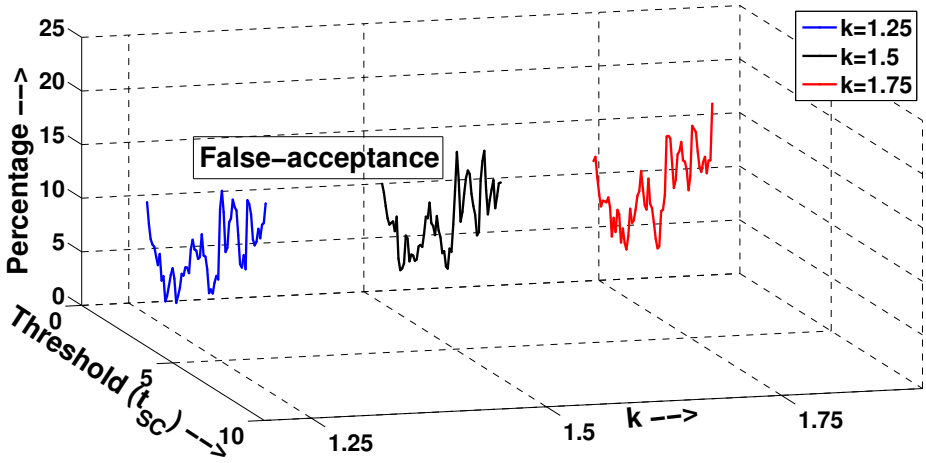
These results show that generating cryptographic keys from voice and skin conductance is effective in fighting coercion attacks, as we observe false acceptance rates between 6% to 15% for $1 \leq t_{SC} \leq 4$, which can also rise up to 22% for $t_{SC} \geq 5$. False rejection rates are between 0% and 4.5% for all values of t_{SC}. Further efforts are needed to reduce the false acceptance and false rejection rates. Same as in the previous subsection, if we increase the value of k from 1.25 to 1.75, we could see a decrease in the false rejection rates and increase in the false acceptance rates.

5.7 Discussion and limitations

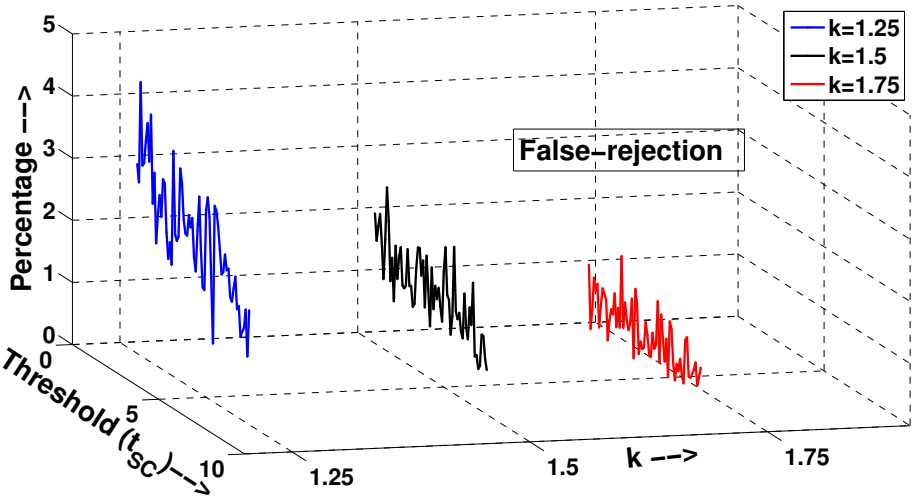
In this section we discuss some of the advanced attack on the model (see Section 5.7.1 and the limitations of our work (see Section 5.7.2).



(a) False acceptance against $\xi_{\text{e2n}}^{\text{full}}$ of user u_j $i \neq j$



(b) False acceptance against $\xi_{\text{e2s}}^{\text{full}}$ of user u_i



(c) False rejection against $\xi_{\text{e2n}}^{\text{test}}$ of user u_i

Figure 5.10: False acceptance and false rejection rates for voice combined with skin conductance

5.7.1 Change in the password space

In this section, we discuss more advanced attacks on our system (if implemented) beside forcing the victim to obtain her spoken password and skin conductance. If such system is implemented, then we need to approximate the entropy in the worst case of these advanced attacks, in which the attacker makes use of the group information about the skin conductance and information stored in the key generation module.

The group information about skin conductance refers to the patterns observed in the change in the users' feature key generated from the skin conductance (b_{SC}) when they are coerced. An attacker could use this information to selectively modify the victims skin conductance feature key in order to improve the probability of generating the correct key. To know how we obtained the feature key (b_{SC}) for SC, see section 5.4.

Although we do not store any biometric information of the user directly on the device (see discussions in Section 5.4), we still need to store the lookup tables (T_V and T_{SC}) which are derived from the user specific data (e.g., feature descriptors). Although this table can be encrypted with a user password as discussed in previous work [116], however we try not to rely the security of our model on the secrecy of this table because we are dealing with coercion attacks. In the rest of this subsection, we assume that an attacker has perfect knowledge in both the group information about skin conductance and the lookup tables. We want to approximate the guessing entropy, i.e., the reduction in the password space for this more powerful attacker.

More precisely, we assume in the worst case that an attacker has access to

- the lookup tables T_V and T_{SC};
- the recorded spoken password of the user and the corresponding feature key $\{b_V(i)\}$;
- the recorded skin conductance when the user is stressed and the corresponding feature key $\{b^S_{SC}(i)\}$;

- the database D which contains the mapping of the SC feature keys when users are normal ($\{b_{SC}^N(i)\}$) to the scenario when they are stressed ($\{b_{SC}^S(i)\}$) for all users in a population A.

A sample database D for such mapping of SC is shown in Table 5.3 for $|A|$ users. Each row in the table is a record of the feature key of a user when she is normal and stressed, and the last column shows the index of the feature keys that had changed from b_{SC}^N to b_{SC}^S.

#	b_{SC}^N	b_{SC}^S	Flipped bits' position
1	011011011011	001101110011	2,4,5,7,9
2	010010010111	010100110110	4,5,7,12
⋮	⋮	⋮	⋮
$\|A\|$	010101001100	111111100110	1,3,5,7,9,11

Table 5.3: A sample database D

The attacker's strategy would be to analyze D to learn patterns in which people's feature keys $\{b_{SC}^N\}$ changes to $\{b_{SC}^S\}$, e.g., whenever the i-th index of the feature key changes, the j-th one will change too.

These patterns can be easily learned by applying a well studied technique called association rule mining [4]. The attacker can then use these patterns to reduce the password space. Here, we use a simple example to demonstrate the idea.

We first represent the password space by a sequence of 0's (the corresponding index in $\{b_{SC}^S\}$ will definitely not change when a user's emotional status changes), 1's (the corresponding index in $\{b_{SC}^S\}$ will definitely change), and $*$'s (don't know), e.g., $[1, *, *]$ represents a password space in which only the first index of $\{b_{SC}^S\}$ will change, and therefore the password space is $2^2 = 4$. When the attacker makes use of a pattern learned, e.g., "the change of the first index of $\{b_{SC}^S\}$ implies the change of the second one", he can convert the password space from $[1, *, *]$ to $[1, 1, *]$, since the second index of the $\{b_{SC}^S\}$ will definitely change, too. With this, the password space reduces to $2^1 = 2$.

We present the detailed algorithm with an example in estimating this reduction in the password space in the Appendix B.

We constructed the database D with the skin conductance samples collected in our user study, mine all association rules, and then use the above algorithm to find out the change in the password space. Figure 5.11 shows the results for different settings of the threshold and minimum confidence in the association rule mining.

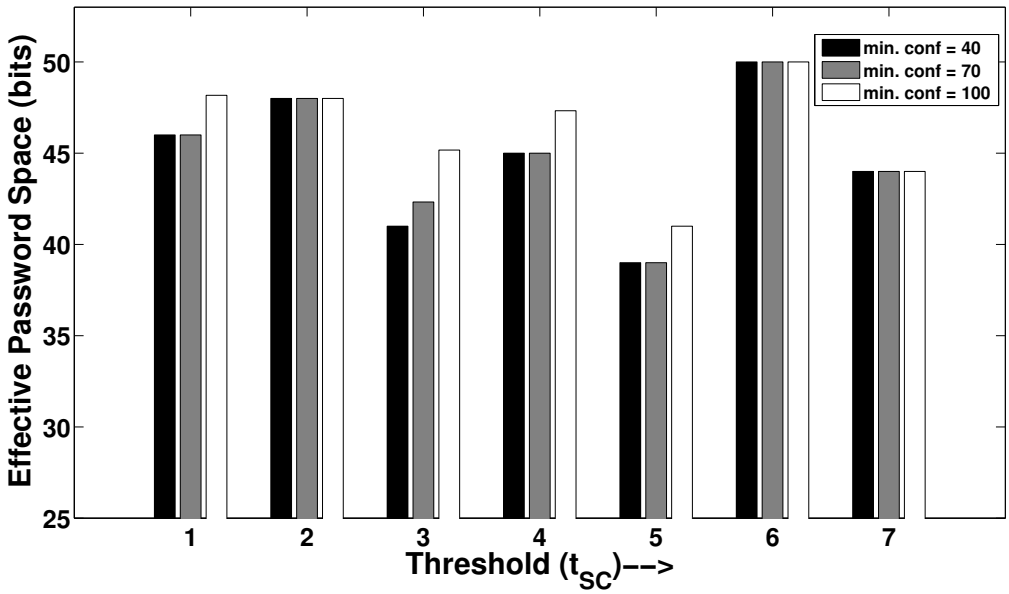


Figure 5.11: Password space reduction

k is set to 1.25 in this experiment, and the minimum support is set to 30%. Note that the original password space is $2^{m_{SC}} = 2^{50}$. Although in the worst case the effective number of bits to represent the password space reduces by roughly 20%, many settings of the threshold value result in only 10% reduction.

Another way to attack our system is to make the user take a sedative to relieve his/her anxiety before capturing SC. The attacker can then use this skin conductance to generate the key. We are trying to collaborate with medical practitioners and researchers to see the correlation between the two skin conductances, one under normal condition without taking any sedative and the other under coercion and having taken the sedative. For now this remains as a future work.

5.7.2 Limitations and summary

Note that guessing entropy and guessing distance [16] might provide deeper insight in the security of our model. We leave it as our future work. In terms of feasibility, in future we will also like to see in some possibilities of building the system (may be a mobile device) with all three: voice, skin conductance and fingerprint extraction mechanism to authenticate to the system. Furthermore, we would like to look into other emotional responses like happy, joy, anger, sad etc., to make the claim of using SC in fighting coercion attacks stronger. This work does not study the repeatability of the key using the proposed scheme and is left as a future work.

To summarize in this work, we demonstrated how an attacker can exploit human factor by succumbing him to coercion attack. To circumvent this attack we presented a novel approach for fighting against coercion attacks in generating cryptographic keys using *skin conductance* (SC) of a person. In coercion attack, the attacker forces a user to grant him access to the system. SC was used to determine the person's overall arousal state i.e. (emotional status). The change in the emotional status of a person results in different keys. We discussed the reasons of adopting SC as an emotional response parameter and why it was preferred over other physiological signals like Electrocardiography, Electromyography, Heart Rate, respiration, skin temperature etc. In this work, we have chosen skin conductance along with voice in generating cryptographic keys; however, one can choose any other biometric for e.g. iris, fingerprint, face etc. in lieu of voice. Cryptographic key is generated using lookup table method as discussed in [116].

We conducted two experiments in our user study and have shown some interesting results. The proposed model was tested with 39 user's voice and skin conductance data to compute the false acceptance and false rejection rate. Furthermore our results showed that the cryptographic key generated in two different scenarios are different for the same person. This bolsters our heuristic to use skin conductance for fighting against coercion attacks. As both skin conductance and voice are not

static biometrics, in some cases we obtained high false rejections. We evaluated the security of the proposed model in terms of entropy and several threat models and discussed how difficult it is for an attacker, in an event when she has full information about the key generation module; the skin conductance of the victim in the stressful scenario; and the group information about the skin conductance.

Chapter 6

Coercion attack in authentication responsibility shifting

6.1 Introduction

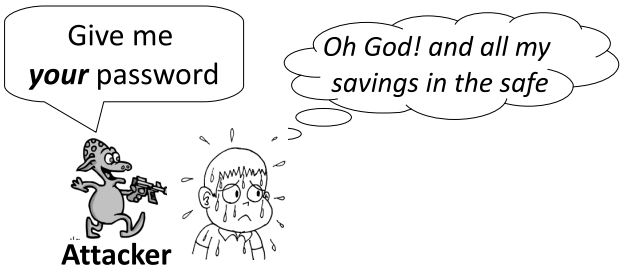
In Chapter 5, we proposed a solution on how to use skin conductance to fight against coercion attacks when the user is forced to reveal his own secret. However, to meet the demand of scalability and usability, many real-world authentication systems have adopted the idea of responsibility shifting, explicitly or implicitly, where a user's responsibility of authentication is shifted to another entity, usually in case of failure of the primary authentication method. One example of explicit responsibility shifting is in the fourth-factor authentication [23] whereby a user gets the crucial authentication assistance from a helper[1] who takes over the responsibility. Facebook also uses a similar authentication protocol which allows the user to recover his account's password by collecting vouch codes from his trusted friends [46]. There is also implicit responsibility shifting which might not seem as obvious. For instance, whenever suspicious activity is detected in an user account, the system administrator takes over the responsibility of revoking the attempted authentication.

Responsibility shifting does *not* enhance the security of the authentication. In-

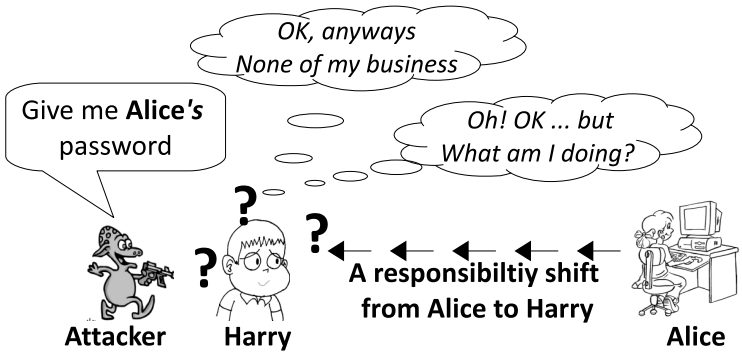
[1]The helper is said to be the fourth factor as someone the user knows.

stead, it entangles with the authentication scenario and may weaken its security. A system that relies on alternate email addresses for password recovery is only as secure as whoever managing those alternate email accounts. The provider of the alternate email account could become the victim of an attack in an attempt to break the authentication in the primary system. In the fourth-factor authentication system [23], subverting the helper allows the adversary to log in without capturing the password of the user.

When the trustee to whom the responsibility has shifted is another computer system, we can use any standard security mechanism to protect it. However, when such a trustee is a human being, protection becomes non-trivial because of the potential *coercion attacks*. To the best of our knowledge, this is the first work to study the security of human trustees under coercion attacks in a responsibility shifting in authentication.



(a) When forced to reveal his own password



(b) When forced to reveal someone else's password

Figure 6.1: Coercion attack in different scenarios

Our previous work in Chapter 5, rely on the fact that the victim's skin-conductance (an emotional response parameter [100]) changes involuntarily upon

coercion, resulting in incorrect authentication credentials. We remark that it is unclear whether the same techniques could help in protecting the trustee in our study. The difference between the trustee and a victim in general coercion attacks is subtle, yet critical in terms of security, see Figure 6.1.

The victim shown in Figure 6.1(a) (and studied in Chapter 5) is coerced to reveal *her own* credential. The consequences include the victim's account being broken into, and her valuable being stolen. It is therefore naturally believed (and experimentally verified) that the victim becomes nervous under such an attack. In contrast, Harry, the trustee considered in this Chapter (see Figure 6.1(b)), is coerced to provide *Alice's credential*, direct consequence of which does not inflict any harm on himself. No prior study has shown the effect on emotional status of Harry in this case and his skin conductance. Therefore, the crux of our work is to investigate whether the trustee's skin conductance also changes under coercion, and if any, whether the magnitude of change is large enough to be captured by the coercion resistance technique.

To put our study into a concrete example, we focus on the fourth-factor authentication [23], a recent proposal on shifting responsibility to help backup authentication. We first provide an overview of the fourth-factor authentication protocol and discuss in detail the potential coercion attack on it. As the main contribution, we then design and conduct a user study involving 29 university students to evaluate the trustee's emotional status in a simulated coercion attack. The results of our user study are positive in the sense that the victim's skin conductance still changes under physical threats. Although there exist several forms of responsibility shifting, our focus is on the fourth-factor authentication scheme because it facilitates rigorous analysis. The principles of our findings in this study are applicable to other authentication mechanisms. Having shown that skin conductance could be used to detect coercion attacks on the helper in the fourth-factor authentication scenario, we further propose a modified protocol that is coercion resistant in protecting the helper.

The rest of the Chapter is structured as follows. Section 6.2 discusses the related work on different responsibility shift schemes. We provide an overview of the fourth factor authentication protocol and its vulnerability to coercion attacks in Section 6.3. We describe our user study in Section 6.4, and report the results in Section 6.5.

6.2 Related work

In this section we review some of the techniques which involve implicit/explicit responsibility shifting and some previous work on emotion recognition. To the best of our knowledge, this work is the first work on stress detection under the context of responsibility shifting. As explained in Section 6.1, an explicit responsibility shift occurs when a user fails to reproduce her credential where an implicit shift occurs in case when there is some suspicious activity in the account etc. In both cases the entity to which the responsibility is shifted can be either "human" or a "computer system".

Role based access control [50] is one such example where the responsibility is implicitly shifted to the system admin (human) to suspend the account suspicious activities are detected. Recently Twitter revoked automatic access to those third party apps abusing its APIs for users tweet collection [150]. This is also an example of implicit responsibility shift whereby the responsibility is shifted to a computer system checking whether the number of API calls are exceeding the limit or not.

There have been many proposals on explicit shift of responsibility when the user fails to generate her credentials. A lot of work has been done in securely shifting the responsibility to another entity(computer systems) while maintaining usability. Alternate email addresses can be used to reset the password of the primary email-id in the case of password loss [164]. Personal knowledge based questions are another backup authentication mechanism and the most commonly used [8, 94, 93]. Another alternative is the preference based backup authentication mechanism proposed in [86, 88]. Google already has added a 2-step verification layer on top

of Google Apps where a mobile phone can be used to receive a verification code via text message or phone call to reset/recover the password [65]. As shown earlier in the Chapter, the fourth factor authentication [23] is one example of explicitly shifting the responsibility to a human being. Another work in this line of research is [139]. Facebook has added a security feature similar to fourth-factor authentication where a user can recover his account by collecting the codes from 3 of his trusted friends [46]. Authentication schemes involving responsibility shifting are always vulnerable to coercion attacks as long as the trustee is a human being.

6.3 Fourth-factor authentication and coercion attacks

As discussed in Section 6.1, fourth-factor authentication [23] is a typical example of responsibility shifting. In this section, we first provide an overview of the protocol used in the fourth-factor authentication (see Section 6.3.1), and then discuss a potential coercion attack when responsibility shifting takes place (see Section 6.3.2). Table 6.1 shows some notations used in the rest of this Chapter.

6.3.1 Fourth-factor authentication protocol

In fourth-factor authentication, a trustee (Harry) to an account holder (Alice) is another registered user of the system who can authenticate himself successfully and is usually a person who knows Alice, e.g., a work colleague. He can verify Alice's identity via any social means, e.g., by recognizing Alice's face or voice over the phone, when the responsibility to authenticate Alice is shifted to him. Here we provide an overview of the fourth-factor authentication system which consists of the authentication server (AS), Alice (u) who needs help in her authentication, and Harry (H) to whom the responsibility to authenticate is shifted.

Enrollment: u provides AS with a list of members $\mathsf{L_u}$ to whom a responsibility to

PARTIES INVOLVED	
u	User or Asker (Alice)
H	Helper (Harry)
AS	Authentication Server
CRP	Coercion Resistance Provider
FOURTH-FACTOR AUTHENTICATION	
H ☞ u	H wants to vouch for u
P	Password
TK	Token
C	code sent from u to H
$\mathsf{L_u}$	list of u's helpers
VC	vouch code
$\mathcal{K}_u$	secret key shared between u and AS
$\mathcal{K}_\mathsf{H}$	secret key shared between H and AS
COERCION RESISTANT FOURTH-FACTOR AUTHENTICATION (ADDITIONAL)	
SID	session ID
LT	lifetime
$\mathcal{K}_{H-\mathsf{CRP}}$	session key for H and CRP provided by AS
$\mathcal{K}_{SC}$	key generated using SC
$\mathcal{K}$	key used to decrypt the encrypted VC
$\mathcal{T}_\mathsf{CRP}$	A ticket provided by AS for CRP

Table 6.1: Notations

authenticate can be shifted in case of emergency authentication.

Responsibility shifting: In case u loses her hardware token TK (but has password P), she shifts the responsibility to H to authenticate herself. Figure 6.2 details each individual step indicated by the numbers and each step's corresponding message exchange indicated by letters.

1. **Partial authentication:** u initiates the authentication process by contacting AS (a), and then encrypts the challenge N_{S1} she receives from AS (b) with her secret key $\mathcal{K}_u = hash(\mathsf{P})$ and sends it back together with a new challenge N_u (c). This step ends with AS verifying N_{S1} and providing u a temporary code *C* which can be used by H (to be chosen in the next step) (d).

2. **Shifting responsibility:** u chooses a helper H from $\mathsf{L_u}$, contacts him and passes the code *C* to him. H verifies the identity of u (by recognizing her face or voice).

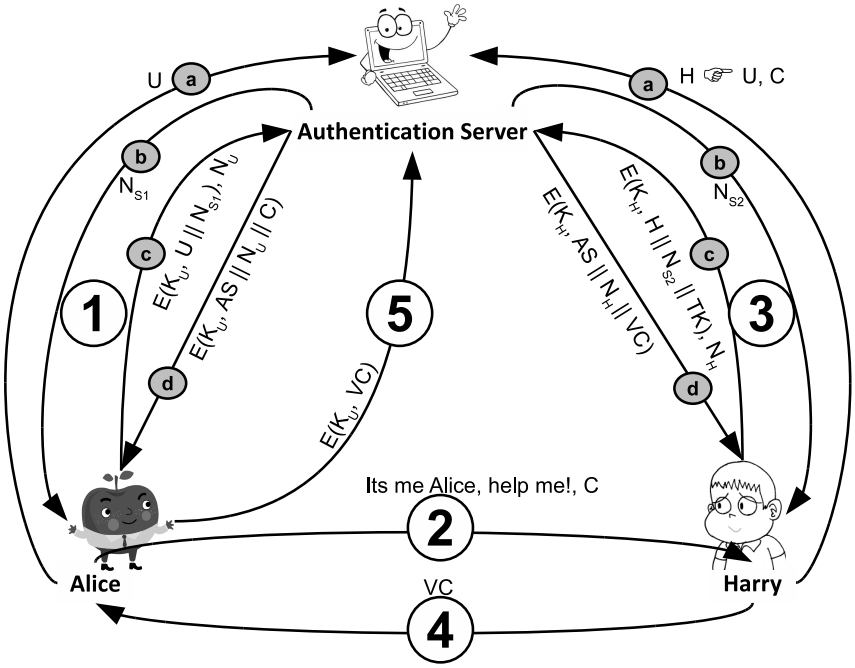


Figure 6.2: Fourth-factor authentication protocol

3. **Authentication of** H **to** AS**:** H presents C and the information H ☞ u (a), authenticates himself to AS by encrypting a challenge N_{S2} from AS (b,c), and obtains a vouch code VC for u (d).

4. H **provides** VC **to** u**:** H passes VC to u using the same means (over the phone or face-to-face).

5. u **presents** VC **to** AS**:** u encrypts VC with $\mathcal{K}_u$ and sends it to AS. AS verifies VC and authenticates u.

This completes the fourth-factor authentication involving a responsibility shifting.

6.3.2 Potential coercion attacks

Note that the responsibility shifting extends the trust base to authenticate Alice from one person (the owner of the account, i.e., Alice) to two persons (Alice and Harry). In Section 6.3.1, Harry together with "half of Alice" (who only has P and loses her hardware token) manage to authenticate Alice to the system. The attacker who has stolen the other half of Alice (the hardware token) could potentially use the same protocol to impersonate Alice if he gets the help from Harry (e.g., by coercing

Harry). This responsibility shifting enables the attacker to extend his coercion target from Alice (who could be an important person heavily armed) to any registered helper (who could be much easier to coerce). Therefore, from Alice's perspective, assigning a helper could potentially make her account less secure. From Harry's perspective, by agreeing to be the helper of Alice, he might run into the risk of attracting coercion attacks on himself due to the new capability he has on Alice's account.

We reiterate that such a coercion attack exists in any responsibility shifting to authenticate in general, e.g., Facebook trust based authentication [46], although in this Chapter we use fourth-factor authentication as a concrete example for better explanation.

6.4 User study

In a coercion attack, the adversary uses physical force, e.g., wielding a gun, to force the victim to comply. When the victim's life is threatened, she would have no choice but to follow what she is ordered to do. Therefore, a critical element to fight against coercion attacks is victim's involuntariness, i.e., defenses must *disable* the victim to perform what the adversary orders her to do.

As discussed in Section 6.1, the previous scenario is substantially different from responsibility shifting discussed in this chapter, where the coercion victim (Harry) is forced to reveal *someone else's* credential (VC for Alice) instead of his own. This raises an important question as whether the requirement of victim's involuntariness still holds here, i.e., whether Harry will be nervous or stressed (which leads to involuntary change of his skin conductance and a different cryptographic key) under such a coercion.

We answer this question by designing and conducting a user study. Obviously, we cannot "really" coerce the participants in our study, but have to mimic a scenario that is close enough while passing our Institutional Review Board (IRB)'s

evaluation. In this section, we first discuss the difficulties and complexity involved in designing this user study. We then explain the participant demographics and the experimental procedure. Results of the user study are shown in Section 6.5.

6.4.1 Difficulties and complexity

The challenge of this user study is to mimic the context of responsibility shifting. For Harry to take over the responsibility from Alice in an authentication, he needs to know her well so that he is able to verify her identity by recognizing her face or her voice. Therefore, one approach of the user study would be to ask two participants (probably friends) to come together. However, this poses a concern as we need to coerce Harry to reveal some personal/privacy information of Alice. Such coercion might lead to a negative impact on the participants' friendship, and is therefore not desirable (would not pass IRB evaluation).

We propose another strategy whereby one participant plays the role of Harry with two conductors (researchers) playing the role of Alice and the adversary ($\mathcal{M}$) respectively. Such a setting eliminates the concern of breaking the friendship of the participants, but would need to satisfy the following criteria.

1. Harry (the participant) should hold some secret of Alice (a researcher) which $\mathcal{M}$ (another researcher) doesn't know (or Harry believes that $\mathcal{M}$ doesn't know).

2. Harry should know this secret before $\mathcal{M}$ tries to coerce him to reveal the secret.

3. Harry should believe that if this secret is leaked to $\mathcal{M}$, then there will be some severe consequences on Alice or on Alice's personal/private data.

Moreover, another difficulty to overcome is to find the right balance between the research requirement of applying sufficient pressure on the participant so as to

mimic a coercion attack, and the human rights requirement of no physical or mental harm to the participants.

6.4.2 Participants and initial setup

Considering the stress on the participants, we decided to concentrate on the younger generation (undergraduate and graduate students in the age from 18 to 30). We have altogether 30 participants, from which one participant was not able to understand the story presented during the user study. Therefore, we have only successfully performed our experiments on 29 participants, out of which 14 were male and 15 were female. Participants were compensated with $20 (equivalent currency) for their participation in the study.

We used the skin conductance device (similar to the one used in Chapter 5) to monitor the skin conductance response SC of the participant.

Initially, there was an incomplete disclosure regarding the purpose and the steps of the study in order to ensure that the participants' responses are not affected by the knowledge of the research.

The user study was carried out in a relatively small room with two laptop computers for Alice and Harry to use. Although Harry was informed that both are Alice's personal and work computers (see Phase-I in Section 6.4.3), we denote these two computers as Alice's computer and Harry's computer in the rest of this Chapter for the sake of clarity. Alice's computer was used to capture the skin conductance of Harry, and Harry's computer was the vehicle for the responsibility shifting as well as coercion attacks (see the detailed procedures below). We developed a small program running on Alice's smartphone which can lock Harry's computer remotely. Alice carried the smartphone in her pocket and used it to lock Harry's computer without being noticed by Harry.

6.4.3 Experimental procedure

The user study is divided in four phases.

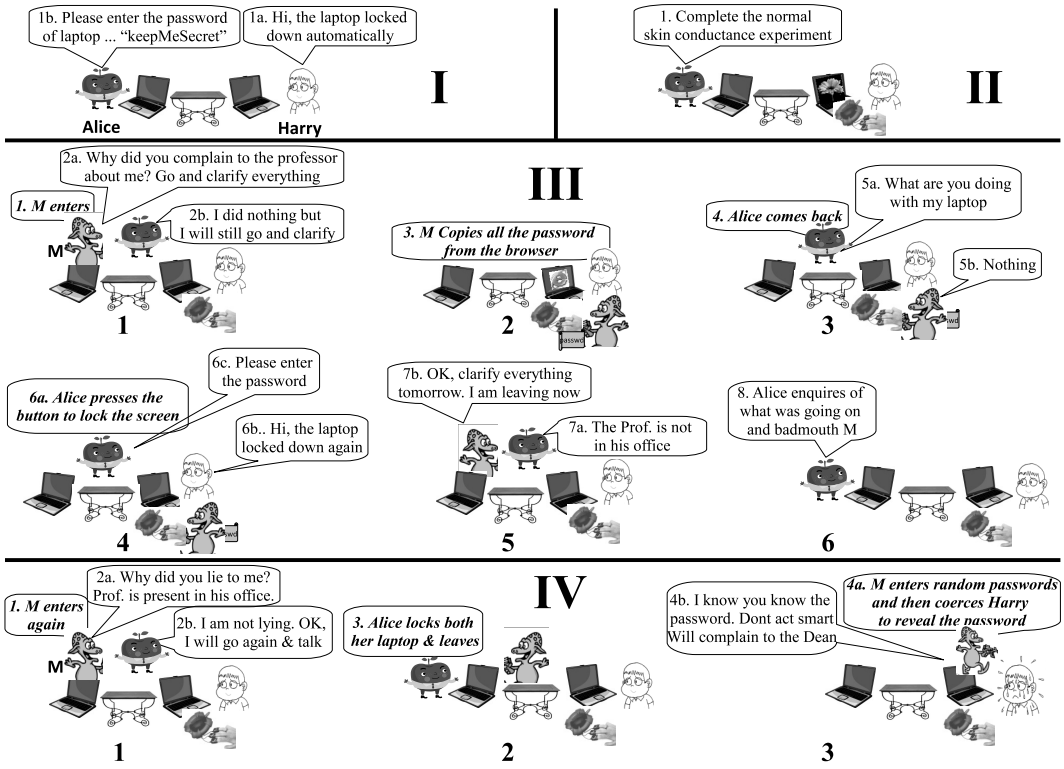


Figure 6.3: Four phases and their component steps/conversation during the user study

Phase I. Passing the secret to Harry

Aim — This is to satisfy criteria 1 and 2 discussed in Section 6.4.1. A secret of Alice is passed to Harry while making Harry believe that $\mathcal{M}$ knows nothing about the secret.

Procedure — At the start of the experiment(see Figure 6.3-I), Harry is greeted by Alice in the room. Alice informs Harry that both computers are hers (personal and work use), and nicely asks Harry not to delete or modify any existing data. After Harry settles down in front of one computer, Alice remotely locks it with her smartphone, and tells him to use password "keepMeSecret" to unlock it. This password becomes the secret Harry knows about Alice and $\mathcal{M}$ will later coerce him to reveal it.

Note that the secret in our study is passed from Alice to Harry directly. This is different from the real world responsibility shifting where the secret is usually passed from an authentication server or another entity. We remark that this would not have changed the results of the user study, as long as the third criterion stated in Section 6.4.1 is satisfied.

Phase II. Gathering normal skin conductance data

Aim — We need to capture the skin conductance response level when Harry is calm to set a baseline (normal emotional state) before coercing him

Procedure — We play a video by showing pleasant (geographical) pictures with soothing music when capturing Harry's skin conductance (see Figure 6.3-II).

Phase III. Portraying $\mathcal{M}$ as a bad guy

Aim — Since we cannot really coerce Harry with, e.g., a gun pointing to his head, we mimic the coercion in a way that is acceptable to the IRB. The simulated coercion has two steps. First, we make Harry believe that $\mathcal{M}$ is a bad guy, and secondly, $\mathcal{M}$ will "coerce" Harry to do something inappropriate (i.e., revealing Alice's secret in Phase IV of the user study).

To make the attack scenario appear real for Harry, we also make an impression in front of Harry that $\mathcal{M}$ is aware of the fact that Harry knows Alice's secret (the password that unlocks Harry's computer in Phase I). This mimics the context of coercion attack in responsibility shifting that $\mathcal{M}$ knows that Harry has taken over the responsibility of Alice's account.

Procedure — as shown in Figure 6.3-III.

1. $\mathcal{M}$ walks into the room and asks Alice (in a slightly rude manner) to leave the room. Alice then walks out.

2. $\mathcal{M}$ walks to Harry's laptop, opens the password manager of the web browser and starts writing down the passwords on a piece of paper. $\mathcal{M}$ makes sure

that Harry observes what he is doing on the laptop.

3. In a short while, Alice returns and $\mathcal{M}$ acts like he is in the situation of embarrassment (idiom: "caught with pants down"). $\mathcal{M}$ immediately closes the password manager.

4. Alice presses the button on her smartphone to lock Harry's laptop (without being noticed by Harry), and then asks Harry to enter the password to unlock it (without speaking out the password). All these take place when $\mathcal{M}$ is in the room.

5. $\mathcal{M}$ behaves rudely while talking to Alice and subsequently leaves the room.

6. Alice explains to Harry that $\mathcal{M}$ is her classmate, and inquires what $\mathcal{M}$ has done during her absence. No matter whether Harry mentions the details or not, Alice badmouths $\mathcal{M}$, which further convinces Harry that $\mathcal{M}$ is really a bad guy.

Phase IV. Coercing Harry

Aim — This is to capture Harry's skin conductance response when $\mathcal{M}$ coerces him to reveal Alice's secret.

Procedure — as shown in Figure 6.3-IV.

1. $\mathcal{M}$ enters the room again and rudely demands that Alice leave the room.

2. This time, Alice walks to Harry's computer and manually locks the screen before leaving the room.

3. After Alice leaves, $\mathcal{M}$ walks over to Harry's computer and starts guessing the password. After a few trials, $\mathcal{M}$ verbally "coerces" Harry to reveal or enter the password. Sentences used by $\mathcal{M}$ include "I will complain to my professor and he will take strict actions against you", "Don't act smart, I know that you know the password".

Toward the end of the user study, we explain to the participants the real motivation of the study and provide a questionnaire to find out their experience during the whole study. Note that Harry's skin conductance is continuously measured throughout the study.

Figure 6.4 shows one of our participants' skin conductance recorded in Phase II, III, and IV.

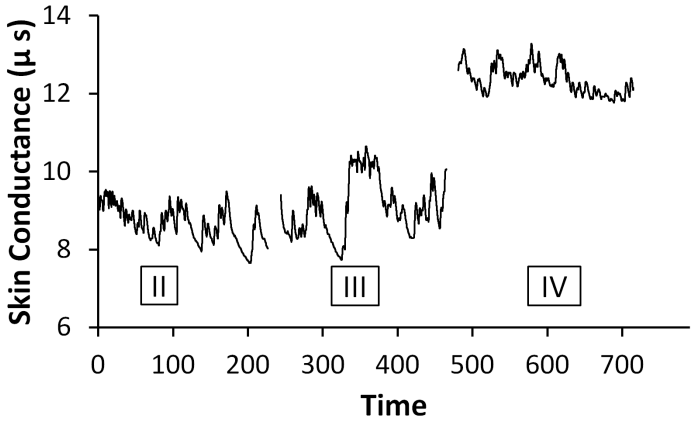


Figure 6.4: Skin conductance response of one participant

6.4.4 Discussions

The user study seems complicated, but every single step is indispensable to achieve the aims as explained above. The design of the user study has gone through many revisions, thanks to the repeated rejections from our IRB and their detailed feedback and recommendations.

Similar studies have been performed previously to measure the stress level in users [118, 91, 69] and to induce people to internalize blame for outcomes they did not produce [96]. Many false confessions are mentioned in [95, 163] that were elicited through the use of torture, threats, and promises.

6.5 Evaluation

We present the results of the user study and our interpretation of the results in this section. As discussed in Section 6.1, there is a subtle yet important difference between the coercion received by someone in a non-responsibility-shifting scenario as shown in Chapter 5 and Harry in our user study. The difference is whether the victim is coerced to reveal her own secret (or the secret that protects her own valuables) or someone else's secret. Therefore, we first analyze what participants felt when they were being coerced to reveal Harry's laptop's password. Building upon that, we then state our hypotheses and based on approach proposed in Chapter 5 we analyze how many participants were actually nervous and stressed. Here, we assume that Harry might be using such a system to protect Alice's secret he has, and evaluate the false acceptance rate and false rejection rate of the system. After that, we analyze the participants' responses to the questionnaire to have a better understanding of the collected skin conductance data. The participants' responses to the questionnaire are noted on a 1–5 Likert scale: strongly agree (●), somewhat agree (◐), neutral (⊖), somewhat disagree (◑) and strongly disagree (○). Finally we discuss the design and some of the limitations of our user study.

6.5.1 Did Harry feel nervous and stressed?

We first review the participants' questionnaire responses to check whether they *felt* nervous and stressed during the coercion. According to the results obtained for our 29 participants, 86% of the participants felt nervous and stressed, and the rest feeling neutral. This has two important implications. First, our user study design is largely a success, in the sense that we have achieved the goal of mimicking coercion on the participants. Second, it seems that most people do feel nervous and stressed even when coerced to reveal someone else's secret, which is the main question our user study seeks to answer. Four out of the 29 participants did reveal the password of Harry's computer, whose comments include the following when inquired.

- "I was intimidated and gave in the password";
- "I was not comfortable when the bad guy was forcing me to enter the password";
- "It was not my password and data";
- "Alice can always change her passwords later on".

Comments from those who did not reveal the password include "it is not ethical to give away someone else's secret information to other", "it is not a good idea to get involved in someone else's personal conflicts", "I was not sure of the kind of personal data residing in the researcher's (Alice) laptop".

6.5.2 Was Harry really nervous and stressed?

Skin conductance has been shown in many previous studies to be a reliable indication of one's emotional status [142]. If participants actually feel nervous in a responsibility-shifting scenario, we envision that one could build a coercion-resistant system using skin conductance. To better understand the extent to which such a system could be successful, we evaluate its accuracy in detecting coercions.

We first state our two hypotheses that

- **Hypothesis 1:** The trustee whom the authentication responsibility shifts to becomes nervous and stressed upon a coercion attack;
- **Hypothesis 2:** What the participants have experienced in the user study presented in Section 6.4 and what the trustee would experience in a coercion attack in the fourth-factor authentication follow the same distribution.

We simulate the execution of the system built upon a previous proposed coercion-resistant system (see Chapter 5 for details) and evaluate our two hypotheses stated with the skin conductance data captured during our user study. We then evaluate its accuracy in terms of false acceptance rate (a correct cryptographic key

generated when Harry is coerced) and false rejection rate (an incorrect key generated when Harry is calm). We define a user as calm/nervous if the key generated during authentication does/does not match against the key generated during enrollment.

The system is trained with 10 out of 26 SC samples (randomly chosen with a duration of 10 seconds) captured during Phase II (when Harry is calm, see Section 6.4.3), and is tested with the remaining 16 SC samples in Phase II (to calculate the false rejection rate) as well as all SC samples in Phase IV (to calculate the false acceptance rate). Figure 6.5 shows the results with three different settings of k (k is used to tolerate some errors in the skin conductance response) and several different settings of t_{SC} (t_{SC} is a threshold value); see Chapter 5 for details.

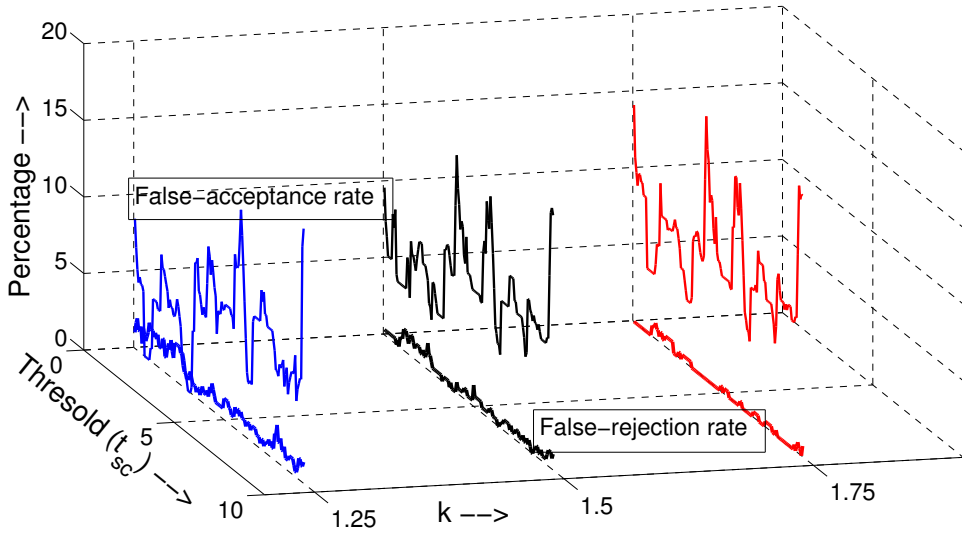


Figure 6.5: False acceptance and false rejection rates

We observe relatively low false acceptance and false rejection rates of our system built under our hypotheses. For example, when $k = 1.25$ and $t_{SC} = 3.1$, we obtained a false acceptance rate of 3.1% and a false rejection rate of 1.7%, which are comparable to those originally obtained in a non-responsibility-shifting scenario (see Chapter 5) (false acceptance rate of 3.2% to 3.1% and false rejection rate of 2.2% to 1.7%). This, in general, shows that Harry was nervous and stressed when coerced to reveal Alice's secret, and the combination of our two hypotheses are good explanations to the data observed during the user study. We also found from

the skin conductance of those 4 participants who revealed the password of Alice's laptop during coercion were all nervous and stressed.

A closer look at Figure 6.5 shows that the false acceptance rates are higher than the false rejection rates. One possible explanation to this is that some participants were not nervous for the whole period of Phase IV of the user study. The "coercion" applied to Harry in our user study is not as severe as a real-world coercion attack, which leads to inaccuracy in our hypothesis 2 and an increase to the false acceptance rate.

6.5.3 Perception v/s reality

We have reported how the participants felt in the Section 6.5.1. It is possible that participants' perception may differ from reality. Table 6.2 shows the comparison between the participants' perceptions (from questionnaire see Section 6.5.1) with the reality (from skin conductance in phase IV see Section 6.5.2). Note that we use the setting of $k = 1.25$, $t_{SC} = 3.1$ where both false alarm and miss rates converge.

Questionnaire (perceived)			Skin conductance (reality)	
			Nervous	Calm
Nervous when being coerced to reveal the password	●	8	8	0
	◐	17	16	1
	⊖	4	4	0
	◑	0	0	0
	○	0	0	0

Table 6.2: Perception v/s reality during coercion

We notice that among the eight participants who *strongly* felt that they were nervous, all had their perception matched with the reality. Among the 17 participants who *somewhat* agreed that they were nervous, however, there was one whose skin conductance did not indicate a change in emotional status. This participant's final comments showed that he was angry rather than nervous. This subtle difference in one's emotion is interesting in that it reveals some limitations in using skin conductance in such a security setting.

Even more interestingly, four participants mentioned that they were neither nervous nor calm; however, their SC data showed that all four of them were actually nervous. Although we do not have other evidence to further analyze these four cases (e.g., we do not have other objective ways of evaluating their emotional status), it does appear that skin conductance could potentially help detect one's emotional changes even before they start feeling the change themselves, an interesting property that deserves further study.

6.5.4 Personal v/s someone else's secret

In this subsection, we focus specifically on Hypothesis 1 to see how the difference between being coerced to reveal one's personal secret and being coerced to reveal someone else's secret could have affected its validity. Note that this is also part of the main question we aim to answer.

We have presented to every participant the following two statements and asked for their responses. Results are shown in Table 6.3.

S-1. In the real world, you feel *nervous* when being coerced to reveal *someone else's* secret information (e.g., email account password).

S-2. In the real world, you feel *nervous* when being coerced to reveal your *own* secret information (e.g. email account password).

		Revealing someone else's secret				
		●	◐	⊖	◑	○
Revealing your own secret	●	6	12	2	0	0
	◐	1	4	0	1	0
	⊖	0	1	1	0	0
	◑	0	0	0	1	0
	○	0	0	0	0	0

Table 6.3: Nervous when being coerced to reveal secret information?

From Table 6.3, we notice that the number of participants above the diagonal (highlighted) are higher (those feeling more nervous when revealing their own secret) as compared to that below the diagonal (those who feel more nervous when

revealing someone else's secret). The result seems to follow common sense, however, note that

- they are based on the perception from the participants, and might not be exactly the same as the reality (see discussions in Section 6.5.3);
- even if these results match with the actual emotional status of the participants it can still be protected by a coercion-resistant system using skin conductance (see Section 6.5.2), as long as the extent to which users feel nervous when being coerced to reveal someone else's secret does not fall below certain threshold.

To get an idea of this point, we perform some simple analysis on the skin conductance captured in this study and that captured as shown in Chapter 5. We found that in our user study the change in the SC data is actually higher (μ=5.18, σ=2.58) as compared to (μ=1.86, σ=1.28).

We warn readers from drawing more than what it deserves from such a simple analysis. First, the two studies are quite different, and a direct comparison of the skin conductance captured does not have a strong basis. Second, although changes in skin conductance have been shown to be a reliable indicator of emotional status [142], it has not been shown that the value of skin conductance reflects the extent to which the user feels nervous. That said, we believe that our simple analysis could be viewed as an evidence that skin conductance does change when they are coerced to perform involuntarily, regardless the ownership of the secret.

6.5.5 Deceptions and observations

In this subsection, we further evaluate Hypothesis 2 in view of the many deceptions used in our user study. A detailed analysis of these deceptions may enable us to explain the non-zero false alarm rate of our system.

Table 6.4 summarizes the six important deceptive events we have in the user study. We specifically query all participants about their reactions to these events,

Event	Deception		Success?	Observation
When participant arrived	Description	The laptop in front of Harry was Alice's laptop.	Yes	Almost all participants felt that the computer was Alice's personal one and is not a setup.
	Perception	● 18 ◐ 10 ⊖ 1 ◑ 0 ○ 0		
When Harry's computer was first locked	Description	Password told to Harry by Alice was an accident and not part of the study.	Yes	25 participants were convinced that it was an accident and not a setup.
	Perception	● 19 ◐ 6 ⊖ 3 ◑ 0 ○ 1		
When Harry's computer was locked again	Description	There was some problem with the computer.	Partial	Only some participants believed that the laptop was faulty. Others felt suspicious.
	Perception	● 9 ◐ 8 ⊖ 6 ◑ 6 ○ 0		
When $\mathcal{M}$ first entered the room	Description	$\mathcal{M}$ behaved rudely in front of the participant.	Yes	Most believed that $\mathcal{M}$ was a bad guy. Others felt suspicious, e.g., that $\mathcal{M}$ would have knocked before coming.
	Perception	● 12 ◐ 13 ⊖ 3 ◑ 1 ○ 0		
When $\mathcal{M}$ coerced the participant	Description	Revealing the password was going to compromise the security of Alice's accounts.	Yes	Almost all participants understood the security of the passwords, except one who believed that it can be changed easily.
	Perception	● 19 ◐ 7 ⊖ 2 ◑ 0 ○ 1		
	Description	$\mathcal{M}$ was going to complain to the professor.	Partial	Coercion was not strong for them to reveal the password (only 11 agreed); however most were nervous.
	Perception	● 1 ◐ 10 ⊖ 10 ◑ 3 ○ 5		

Table 6.4: Participants' perception towards various deceptions used

results of which are shown in Table 6.4 as well. We realize that four out of are convincing enough.

However, the other two events appear suspicious to some participants. One is when the computer was locked for the second time (the third row in Table 6.4). There were 6 participants who question its genuineness and another 6 participants were neutral in their responses. Most of the participants felt that Alice could have simply used another computer for the study rather than using the faulty one. However, the SC data of these 12 (6+6) participants indicate that all were nervous during coercion. So, although this event seems suspicious to a portion of the participants, it does not contribute to the false alarm rate of the system.

The other event is when the attacker coerces the participant (last row in Table 6.4). Some participants felt that the attacker will not complain to the professor as they have not done anything wrong. Even if attacker complains, they could easily defend themselves. Some felt that the coercion/reason was not strong enough for them to reveal Alice's password. However, all of them were actually nervous during coercion according to their SC data.

In summary, part of the deceptions are not fully satisfactory. However, there is no evidence showing that they render the user study unsuccessful or have significantly contributed to the false alarm rates.

6.5.6 Design of our user study

We continue to evaluate the validity of Hypothesis 2, now from the perspective of the design of our user study. Specifically, we evaluate the differences between coercion in our study and that in a real-world fourth-factor authentication.

Relationship between Alice and Harry In the fourth-factor authentication protocol, Alice and Harry are close friends or colleagues (strong bonding), while in our user study Alice and Harry did not know each other (weak bonding) until the study begins. Such a relatively weak bonding between Alice and Harry in our study

might make Harry feel not nervous when being coerced to reveal Alice's password. In other words, measuring skin conductance might lead to better protection against coercion in a real-world fourth-factor authentication, when compared to that observed in our user study.

Consequences of revealing the secret One may argue that in fourth-factor authentication, the secret (vouchcode) alone cannot be used to get access to Alice's account and therefore has less severe consequences. This could make Harry in our study feel relatively more stressed because of the severe consequences as the password alone allows $\mathcal{M}$ to invade Alice's account. However, if one assumes that the attacker in fourth-factor authentication has obtained Alice's second-factor authentication token before the coercion attack, such a difference would not exist.

How secret was given to Harry In the fourth-factor authentication protocol, the secret (vouchcode) is passed to Harry by the server. However, the secret (the password) is passed to Harry by Alice in our study. The participant might believe that Alice's password sharing with him implies that it could be shared with others, too. Although this remaining a possibility, we do not see any evidence from our study that supports it. All participants understand the consequence of revealing Alice's password, including the four who did reveal it under coercion.

Degree of coerciveness Threatening to complain to a professor is definitely not as coercive as putting a gun over one's head. This could make Harry in our study feel less nervous or stressful. This implies that our results about skin conductance could be conservative in the sense that a real life coercion is more likely to cause the victim more nervous.

6.5.7 Limitations of our user study

There are two main limitations of our user study. First, Alice is played by a female member of our research team in our user study. Since people in general show compassion towards female gender, our results could be biased. Second, as the user study is an act, many unforeseen events did take place. The actual scenarios were not always consistent throughout the user study across different participants.

6.6 Coercion resistant fourth-factor authentication

Our user study shows that a trustee to whom responsibility has shifted would become nervous and stressed when being coerced, which would be reflected on a change in his/her skin conductance. This leads us to believe that skin conductance could be used to provide coercion resistance to authentication protocols with a shift of responsibility. Having formalized the fourth-factor authentication protocol in Section 6.3.1, we design a modified version of the protocol with coercion resistance using skin conductance in this section.

We introduce a trusted third-party Coercion Resistance Provider (CRP) to provide coercion resistance property to fourth-factor authentication. Following the well-known *separation of duty* principle, we consider CRP and AS as two separate entities. Note that combining them into one entity requires minimal changes to the following protocol. The details of the protocol are described below, resembling the Kerberos protocol. Note that the notations are defined in Table 6.1.

Enrollment It is the same as the Enrollment procedure in Section 6.3.1, except that Harry (H) needs to register himself to the CRP. We use the approach proposed in previous chapter to generate a cryptographic key from H's skin conductance and use it to encrypt a secret stored with CRP. If at the time of authentication H is able to re-generate the original key and decrypt the stored secret correctly, then H is not under coercion. CRP maintains a skin-conductance template for each user as

explained previously.

Authentication When H tries to vouch for u, H needs to prove to CRP that he is not being coerced, besides authenticating himself to AS. To do that, we modify step 3 (see Figure 6.2) of the fourth-factor authentication protocol. Figure 6.6 shows these modified/additional steps (3-d to 3-f).

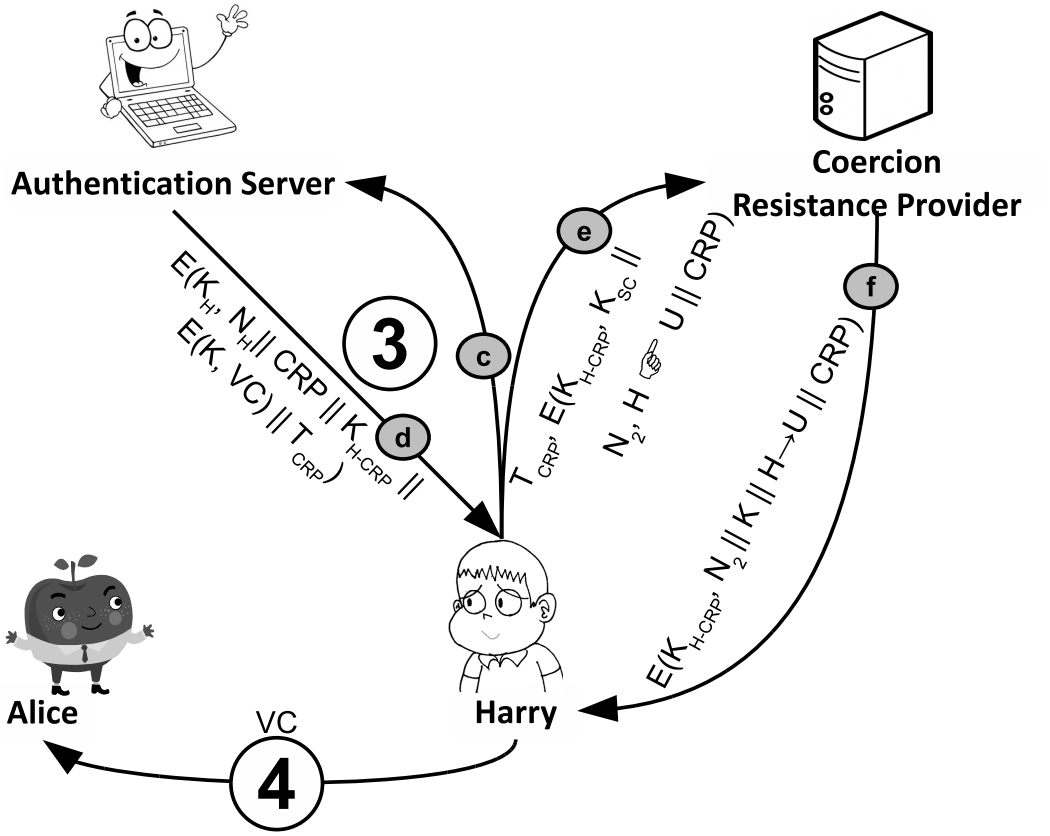


Figure 6.6: Coercion Resistant Fourth-factor authentication

3(d) **Server sends Ticket to Helper:** Rather than encrypting VC with $\mathcal{K}_{\mathsf{H}}$ (see Section 6.3.1), AS encrypts VC with a key $\mathcal{K}$. This key $\mathcal{K}$ is encrypted with CRP's key ($\mathcal{K}_{\mathsf{CRP}}$) to generate a ticket ($\mathcal{T}_{\mathsf{CRP}}$). This ensures that H cannot obtain VC without proving to CRP that he is not being coerced to do so. $\mathcal{T}_{\mathsf{CRP}}$ contains a session key ($\mathcal{K}_{H-\mathsf{CRP}}$) shared between H and CRP provided by AS, a key ($\mathcal{K}$) to decrypt the encrypted VC, lifetime of this message (LT), session-id (SID) and the information (H ☞ u) which states that H vouches for u. Formally,

$$\mathsf{AS} \longrightarrow \mathsf{H}: \ \mathrm{E}(\mathcal{K}_{\mathsf{H}}, N_{\mathsf{H}} \parallel \mathsf{CRP} \parallel \mathcal{K}_{H-\mathsf{CRP}} \parallel \mathrm{E}(\mathcal{K}, \mathsf{VC}) \parallel \mathcal{T}_{\mathsf{CRP}})$$

where, $\mathcal{T}_{\mathsf{CRP}} = \mathrm{E}(\mathcal{K}_{\mathsf{CRP}}, \mathcal{K}_{H-\mathsf{CRP}} \parallel \mathsf{H} \text{☞} u \parallel \mathsf{SID} \parallel \mathcal{K} \parallel \mathsf{LT})$

3(e) **Helper contacts Coercion Resistance Provider:** H decrypts the obtained message with his secret key ($\mathcal{K}_{\mathsf{H}}$), and retrieves $\mathcal{T}_{\mathsf{CRP}}$ and $\mathcal{K}_{H-\mathsf{CRP}}$. In order to prove to CRP that he is not under coercion, he generates a key $\mathcal{K}_{SC}$ using his skin conductance response. The generated $\mathcal{K}_{SC}$, the freshly generated nonce (N_2) and the information (H ☞ u) are all encrypted with $\mathcal{K}_{H-\mathsf{CRP}}$ and sent to CRP. $\mathcal{T}_{\mathsf{CRP}}$ is also forwarded to prove that he has already been authenticated by AS. Formally,

$$\mathsf{H} \longrightarrow \mathsf{CRP}: \ \mathcal{T}_{\mathsf{CRP}}, \mathrm{E}(\mathcal{K}_{H-\mathsf{CRP}}, \mathcal{K}_{SC} \parallel N_2 \parallel \mathsf{H} \text{☞} u \parallel \mathsf{CRP})$$

3(f) **Helper Obtains Key to Decrypt Vouchcode:** CRP decrypts the message from H by using $\mathcal{K}_{H-\mathsf{CRP}}$ and obtains $\mathcal{K}_{SC}$. $\mathcal{K}_{SC}$ is then used to decrypt all the stored encrypted values. If the decryption succeeds (by matching the released $\mathcal{B}$ and the stored $\mathcal{B}$), then CRP is ensured that H is not under coercion. It then encrypts key $\mathcal{K}$ with $\mathcal{K}_{H-\mathsf{CRP}}$ and sends the result to H. H obtains $\mathcal{K}$ and decrypts the encrypted VC earlier provided by AS. Formally,

$$\mathsf{CRP} \longrightarrow \mathsf{H}: \ \mathrm{E}(\mathcal{K}_{H-\mathsf{CRP}}, N_2 \parallel \mathcal{K} \parallel \mathsf{CRP} \parallel \mathsf{H} \text{☞} u)$$

6.7 Discussion

To summarize, in this work, we demonstrate another evidence of exploiting human factor to gain access though the system and proposed a solution to circumvent those attacks. We study the security of human-trustee based authentication responsibility shifting, in particular, under coercion attacks. Our intensive user study shows that most trustees demonstrate nervousness when being forced to reveal others' secret, which can be captured by their involuntary skin conductance changes. We retrofit the fourth factor authentication system to develop a coercion-resistant systems for responsibility shifting in authentication.

Chapter 7

Conclusions and perspectives

7.1 Summary of contribution and future work

User authentication is vulnerable to attacks exploiting human factors, e.g. secrecy information inference and coercion attacks; and information from user's mobile device and skin conductance can help to fight against those attacks. In this thesis, we demonstrated aforementioned attacks and proposed respective solutions.

We demonstrated severe implications of secrecy information mining by showing that interests inferred from public data (obtained using Graph API from Facebook) can be used to exploit a previously proposed preference based authentication system (Blue Moon^{TM}). This differs from prior research as we do not use users' personal data posted on their profile pages (e.g., gender, current location, activities, interests, etc.). From our experiments, we were able to disclose 22 interests of a user and found more than 80,097 users with at least 2 interests. We showed that there exists many users who liked a Facebook page, however they post negative comments on the page. In future, we would like to improve our mining approach to infer more secrecy attributes apart from interests.

Human memory interference is a major problem with today's authentication mechanisms. We designed HuMan, that uses cellphone usage patterns to generate memorable fingerprints which can be used as authentication challenges and are

resistant to secret information inference attacks as well. We subjected HuMan to a difficult security threat model where users' intimates and acquaintances try to guess the fingerprints. This was validated via a user study involving 58 participants on two phases on two different phone operating systems (Symbian and Android). We found that HuMan is moderately secure against the threat model proposed. Moving forward, we plan to continue our research into human-centric approaches in generating quality fingerprints in a number of ways: 1) developing a better fingerprint generation module using both better data mining algorithms as well as improved template generation engines; and 2) testing HuMan with a broader and more diverse set of users. As a future work, we would like to work on a recommendation system for backup authentication systems where all the public/private profiles of a user can be used to determine his/her weak and strong fingerprints. Organizations like call centers or staff at help desks can use these fingerprints to decide what challenges should (not) be used in determining the identity of users in backup authentication systems.

The second problem which we focused in this thesis is human vulnerability to coercion attacks. We proposed a novel approach for fighting against coercion attacks in generating cryptographic keys using skin conductance of a person. We conducted two experiments in our user study and have shown some interesting results. The proposed model was tested with 39 user's voice and skin conductance data to compute the false acceptance rate and false rejection rate. Furthermore, our results showed that the cryptographic key generated in two different scenarios are different for the same person. This bolsters our heuristic to use skin conductance for fighting against coercion attacks. We also study the security aspects of human-trustee based authentication responsibility shifting, in particular, under coercion attacks. Our intensive user study showed that most trustees demonstrate nervousness when being forced to reveal others' secret, which can be captured by their involuntary skin conductance changes. We then design and develop a coercion-resistant system for responsibility shifting in authentication. It remains unclear how to com-

pare one person's skin conductance fluctuation under different coercion scenarios. Since the victim of coercion attacks may not be the same as the adversary's ultimate target, we conjecture that the victim may produce involuntary responses in different ways for different scenarios. In terms of feasibility, in future we will also like to see in some possibilities of building the system (may be a mobile device) with all three: voice, skin conductance and fingerprint extraction mechanism to authenticate to the system. Furthermore, we would like to look into other emotional responses like happiness, anger, sadness etc., to make the claim of using skin conductance in fighting coercion attacks stronger. We did not study the repeatability of the key using the proposed scheme and is left as a future work.

7.2 Future perspective

In future, most devices will posses the capability of connecting to the Internet and communicating with each other. Everything will be connected and communicating to everything else, uninterruptedly. Today, we humans leave our behavioral fingerprints in day to day life while using these devices/instruments e.g. smartphones, biometric devices like fingerprint reader, digital media like TV or music players, computers, household items like microwaves or washing machines, air conditioners, motor vehicles or cars, electrical equipments like lamps etc. Microsoft's surface touch table has already started to replace our old and traditional concept of table. People interact with these devices in day to day life knowingly or unknowingly. "What", "when", "where" and "how" these devices are used by a particular human being constitute the behavioral fingerprint of that person. I envision that authentication based on users' behavior can play a big role in the design of a secure system. If a user knows his/her own behavior then he/she does not need to remember extra stuff to login in to her email account or unlocking the phone using pin numbers. Behavioral fingerprinting can add a dimension to authentication challenges.

Bibliography

[1] Aditya Abhyankar and Stephanie Schuckers. Integrating a wavelet based perspiration liveness check with fingerprint recognition. *Pattern Recognition*, 42(3):452–464, 2009.

[2] Anne Adams and Martina Angela Sasse. Users are not the enemy. *Commun. ACM*, 42(12):40–46, December 1999.

[3] Yael Adini, Yael Moses, and Shimon Ullman. Face recognition: the problem of compensating for changes in illumination direction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):721 –732, jul 1997.

[4] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM.

[5] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 487–499, 1994.

[6] Jenni Anttonen and Veikko Surakka. Emotions and heart rate while sitting on a chair. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 491–499, New York, NY, USA, 2005. ACM.

[7] Hassan Jameel Asghar, Josef Pieprzyk, and Huaxiong 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*, ACNS'10, pages 349–366, Berlin, Heidelberg, 2010. Springer-Verlag.

[8] Farzganeh Asgharpour and M. Jakobsson. Adaptive challenge questions algorithm in password reset/recovery. Presented at First International Workshop on Security for Spontaneous Interaction (IWIISI), 2007.

[9] Family Law Attorney. Maintaining privacy and starting a separate life during divorce. `http://www.michaelarobbins.com/Practice-Areas/Maintaining-Privacy-and-Starting-a-Separate-Life-During-Divorce.shtml`.

[10] Andrea Esuli Stefano Baccianella and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[11] Xiaole Bai, Wenjun Gu, Sriram Chellappan, Xun Wang, Dong Xuan, and Bin Ma. Pas: Predicate-based authentication services against powerful passive adversaries. In *Proceedings of the 2008 Annual Computer Security Applications Conference*, ACSAC '08, pages 433–442, Washington, DC, USA, 2008. IEEE Computer Society.

[12] Lynne Baillie, Lee Morton, Gillian MacLellan, and Gemma Ryde. Designing a mobile application to capture everyday activity. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '09, pages 82:1–82:2, New York, NY, USA, 2009. ACM.

[13] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context aware systems. *Int. J. Ad Hoc Ubiquitous Comput.*, 2:263–277, June 2007.

[14] Marco Balduzzi, Christian Platzer, Thorsten Holz, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. Abusing social networks for automated user profiling. In *Proceedings of the 13th international conference on Recent advances in intrusion detection*, RAID'10, pages 422–441, Berlin, Heidelberg, 2010. Springer-Verlag.

[15] Lucas Ballard, Seny Kamara, Fabian Monrose, and Michael K. Reiter. Towards practical biometric key generation with randomized biometric templates. In *CCS '08: Proceedings of the 15th ACM conference on Computer and communications security*, pages 235–244, New York, NY, USA, 2008. ACM.

[16] Lucas Ballard, Seny Kamara, and Michael K. Reiter. The practical subtleties of biometric key generation. In *SS'08: Proceedings of the 17th conference on Security symposium*, pages 61–74, Berkeley, CA, USA, 2008. USENIX Association.

[17] Gilles Barthe, Anupam Datta, and Sandro Etalle, editors. *Formal Aspects of Security and Trust - 8th International Workshop, FAST 2011, Leuven, Belgium, September 12-14, 2011. Revised Selected Papers*, volume 7140 of *Lecture Notes in Computer Science*. Springer, 2012.

[18] Rakesh Belwal and Shweta Belwal. Mobile phone usage behavior of university students in oman. In *Proceedings of the 2009 International Conference on New Trends in Information and Service Science*, pages 954–962, Washington, DC, USA, 2009. IEEE Computer Society.

[19] Benn. *Android Root Source Code*, September 2010. `http://intrepidusgroup.com/insight/2010/09/android-root-source-code-looking-at-the-c-skills`.

[20] Gerald Bieber, Philipp Koldrack, Christopher Sablowski, Christian Peter, and Bodo Urban. Mobile physical activity recognition of stand-up and sit-down transitions for user behavior analysis. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '10, pages 50:1–50:5, New York, NY, USA, 2010. ACM.

[21] Lena Biel, Ola Pettersson, Lennart Philipson, , and Peter Wide. ECG analysis: a new approach in human identification. *Instrumentation and Measurement, IEEE Transactions on*, 50(3):808–812, Jun 2001.

[22] Abhijit Bose, Xin Hu, Kang G. Shin, and Taejoon Park. Behavioral detection of malware on mobile handsets. In *Proceeding of the 6th international conference on Mobile systems, applications, and services*, MobiSys '08, pages 225–238, New York, NY, USA, 2008. ACM.

[23] John Brainard, Ari Juels, Ronald L. Rivest, Michael Szydlo, and Moti Yung. Fourth-factor authentication: somebody you know. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS '06, pages 168–178, New York, NY, USA, 2006. ACM.

[24] Ted Bridis. Hacker impersonated palin, stole e-mail password. `http://www.usatoday.com/news/politics/2008-09-17-152224562_x.htm`.

[25] Sacha Brostoff and M. Angela Sasse. Are passfaces 1 more usable than passwords? a field trial investigation.

[26] John T. Cacioppo and Louis G. Tassinary. Inferring psychological significance from physiological signals. *American Psychologist*, 45(1):16–28, Jan 1990.

[27] J. P. Campbell. Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997.

[28] Gemma Casas-Garriga. Discovering unbounded episodes in sequential data. In Nada Lavrac, Dragan Gamberger, Ljupco Todorovski, and Hendrik Blockeel, editors, *Knowledge Discovery in Databases: PKDD 2003*, volume 2838 of *Lecture Notes in Computer Science*, pages 83–94. Springer Berlin / Heidelberg, 2003.

[29] Abdelberi Chaabane, Gergely Acs, and Mohamed A. Kaafar. You are what you like! information leakage through users' interests. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, February 2012.

[30] Ye Chen, Dmitry Pavlov, and John F. Canny. Large-scale behavioral targeting. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 209–218, New York, NY, USA, 2009. ACM.

[31] Jeremy Clark and Urs Hengartner. Panic passwords: authenticating under duress. In *Proceedings of the 3rd conference on Hot topics in security*, HOTSEC'08, pages 8:1–8:6, Berkeley, CA, USA, 2008. USENIX Association.

[32] Jack Ervin Conklin. Three factors affecting the general level of electrical skin-resistance. *The American Journal of Psychology*, 64(1):78–86, Jan 1951.

[33] Nelson Cowan, Candice C. Morey, Zhijian Chen, Amanda L. Gilchrist, and J. Scott Saults. Theory and measurement of working memory capacity limits. In Brian H. Ross, editor, *The Psychology of Learning and Motivation*, volume 49, pages 49 – 104. Academic Press, 2008.

[34] Nik Cubrilovic. The anatomy of the twitter attack. `http://techcrunch.com/2009/07/19/the-anatomy-of-the-twitter-attack/`.

[35] Darren Davis, Fabian Monrose, and Michael K. Reiter. On user choice in graphical password schemes. In *Proceedings of the 13th conference on USENIX Security Symposium - Volume 13*, SSYM'04, pages 11–11, Berkeley, CA, USA, 2004. USENIX Association.

[36] Marco de Sá, Luís Carriço, Luís Duarte, and Tiago Reis. A mixed-fidelity prototyping tool for mobile devices. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 225–232, New York, NY, USA, 2008. ACM.

[37] Cynthia Dermody. Sharing passwords with your husband is risky business. http://thestir.cafemom.com/technology/131914/sharing_passwords_with_your_husband.

[38] Rachna Dhamija and Adrian Perrig. Déjà vu: a user study using images for authentication. In *Proceedings of the 9th conference on USENIX Security Symposium - Volume 9*, SSYM'00, pages 4–4, Berkeley, CA, USA, 2000. USENIX Association.

[39] Rachna Dhamija, J. D. Tygar, and Marti Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 581–590, New York, NY, USA, 2006. ACM.

[40] Trinh Minh Tri Do, Jan Blom, and Daniel Gatica-Perez. Smartphone usage in the wild: a large-scale analysis of applications and context. In *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, pages 353–360, New York, NY, USA, 2011. ACM.

[41] Julie S. Downs, Mandy B. Holbrook, and Lorrie Faith Cranor. Decision strategies and susceptibility to phishing. In *Proceedings of the second symposium on Usable privacy and security*, SOUPS '06, pages 79–90, New York, NY, USA, 2006. ACM.

[42] P Ekman. *Basic Emotions*, volume 476 of *Handbook of Cognition and Emotion*. T. Dalgleish and M. Power, John Wiley & Sons Ltd. Sussex UK, 1999.

[43] EncoSkin. Encoll's collagen technology, 2010. http://www.encoll.com/SkinCare_and_Dietary_Products.htm.

[44] Facebook. Facebook pages. http://www.facebook.com/directory/pages/.

[45] Facebook. Graph api. https://www.developers.facebook.com/docs/reference/api/.

[46] Facebook. What are trusted friends? security. http://www.facebook.com/help/?faq=119897751441086.

[47] Zheng Fang, Zhang Guoliang, and Song Zhanjiang. Comparison of different implementations of mfcc. *J. Comput. Sci. Technol.*, 16(6):582–589, 2001.

[48] C. Fellbaum. Wordnet: An electronic lexical database. http://wordnet.princeton.edu/.

[49] Henry Hanping Feng, Oleg M. Kolesnikov, Prahlad Fogla, Wenke Lee, and Weibo Gong. Anomaly detection using call stack information. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, SP '03, pages 62–, Washington, DC, USA, 2003. IEEE Computer Society.

[50] David Ferraiolo and Richard Kuhn. Role-based access control. In *In 15th NIST-NCSC National Computer Security Conference*, pages 554–563, 1992.

[51] Stephanie Forrest, Steven A. Hofmeyr, Anil Somayaji, and Thomas A. Longstaff. A sense of self for unix processes. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, SP '96, pages 120–, Washington, DC, USA, 1996. IEEE Computer Society.

[52] Don C. Fowles. The three arousal model: Implications of gray's two-factor learning theory for heart rate, electrodermal activity, and psychopathy. *Psychophysiology*, 17(2):87–104, 1980.

[53] M. Freire-Santos, J. Fierrez-Aguilar, and J. Ortega-Garcia. Cryptographic key generation using handwritten signature. *Biometric Technologies for Human Identification III, Processing of SPIE*, 6202, 2006.

[54] Mark Gabel and Zhendong Su. Online inference and enforcement of temporal properties. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1*, ICSE '10, pages 15–24, New York, NY, USA, 2010. ACM.

[55] Liz Gannes. Mole-whacking: Vendor says spam is growing on facebook fan pages. `http://allthingsd.com/20110414/mole-whacking-vendor-says-spam-is-growing-on-facebook-fan-pages/`.

[56] Debin Gao, Michael K. Reiter, and Dawn Song. Gray-box extraction of execution graph for anomaly detection. In *Proceedings of the 11th ACM Conference on Computer & Communication Security (CCS)*, 2004.

[57] Debin Gao, Michael K. Reiter, and Dawn Song. On gray-box program tracking for anomaly detection. In *Proceedings of the 13th USENIX Security Symposium*, 2004.

[58] L.P. García Perera, J.A. Nolazco Flores, and C. Mex Perera. Cryptographic-speech-key generation architecture improvements. In *IbPRIA05*, page II:579, 2005.

[59] Gartner Newsroom. *Worldwide Mobile Phone Sales Grew 35 in Third Quarter 2010*, November 2010. `http://www.gartner.com/it/page.jsp?id=1466313`.

[60] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. The adaptive web. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, chapter User profiles for personalized information access, pages 54–89. Springer-Verlag, Berlin, Heidelberg, 2007.

[61] Daniel Gayo Avello. All liaisons are dangerous when all your friends are known to us. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, HT '11, pages 171–180, New York, NY, USA, 2011. ACM.

[62] Anup K. Ghosh, Aaron Schwartzbard, and Michael Schatz. Learning program behavior profiles for intrusion detection. In *Proceedings of the Workshop on Intrusion Detection and Network Monitoring*, pages 51–62, Berkeley, CA, USA, 1999. USENIX Association.

[63] Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, and Renata Teixeira. On exploiting innocuous user activity for correlating accounts acrossăsocial network sites & twitter & personal profiles. Technical report, International Computer Science Institute, 2012.

[64] Philippe Golle and David Wagner. Cryptanalysis of a cognitive authentication scheme (extended abstract). In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, pages 66–70, Washington, DC, USA, 2007. IEEE Computer Society.

[65] Google. 2-step verification. `http://www.google.com/support/a/bin/answer.py?answer=175197/`.

[66] Google. *Google Android Face Unlock*. `http://www.android.com/about/ice-cream-sandwich/`.

[67] Leonard H. Grokop, Anthony Sarah, Chris Brunner, Vidya Narayanan, and Sanjiv Nanda. Activity and device position recognition in mobile devices. In *Proceedings of the 13th international conference on Ubiquitous computing*, UbiComp '11, pages 591–592, New York, NY, USA, 2011. ACM.

[68] Payas Gupta, Xuhua Ding, and Debin Gao. Coercion resistance in authentication responsibility shifting. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, New York, NY, USA, 2012. ACM.

[69] Payas Gupta and Debin Gao. Fighting coercion attacks in key generation using skin conductance. In *Proceedings of the 19th USENIX conference on Security*, USENIX Security'10, Berkeley, CA, USA, 2010. USENIX Association.

[70] Payas Gupta, Swapna Gotipatti, Jing Jiang, and Debin Gao. Your love is public now: Questioning the use of personal information in authentication. In *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '13, New York, NY, USA, 2013. ACM.

[71] Payas Gupta, Kiat Wee Tan, Narayan Ramasubbu, David Lo, Debin Gao, and Rajesh Krishna Balan. Design and implementation of human memorable fingerprints. Technical Report SMU-SIS-13-100, Singapore Management University, Mar 2013.

[72] Payas Gupta, Tan Kiat Wee, Narayan Ramasubbu, David Lo, Debin Gao, and Rajesh Krishna Balan. Human: Creating memorable fingerprints of mobile users. In *Tenth Annual IEEE International Conference on Pervasive Computing and Communications, PerCom*. IEEE, 2012.

[73] Norbert Györbíró, Ákos Fábián, and Gergely Hományi. An activity recognition system for mobile phones. *Mob. Netw. Appl.*, 14:82–91, February 2009.

[74] M. Helander. Applicability of drivers' electrodermal response to the design of the traffic environment. *Journal of Applied Psychology*, 63(4):481–488, 1978.

[75] Bogdan Hoanca and Kenrick Mock. Secure graphical password system for high traffic public areas. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, ETRA '06, pages 35–35, New York, NY, USA, 2006. ACM.

[76] Mat Honan. How apple and amazon security flaws led to my epic hacking. `http://www.wired.com/gadgetlab/2012/08/apple-amazon-mat-honan-hacking/all/`.

[77] Jason Hong. The state of phishing attacks. *Commun. ACM*, 55(1):74–81, January 2012.

[78] Se-Joon Hong, James Y. L. Thong, Jae-Yun Moon, and Kar-Yan Tam. Understanding the behavior of mobile data services consumers. *Information Systems Frontiers*, 10:431–445, September 2008.

[79] Se-Joon Hong, James Y. L. Thong, and Kar Yan Tam. Understanding continued information technology usage behavior: a comparison of three models in the context of mobile internet. *Decis. Support Syst.*, 42:1819–1834, December 2006.

[80] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD*, pages 168–177, 2004.

[81] RavenWhite Inc. The blue moon authentication system. `http://www.ravenwhite.com/iforgotmypassword.html`.

[82] FEDERAL BUREAU OF INVESTIGATION. Bank crime statistics (bcs) federal insured financial institutions july 1, 2009 Ű september 30, 2009. `http://www.fbi.gov/publications/bcs/bcs2009/bank_crime_2009q3.htm`.

[83] Danesh Irani, Steve Webb, Kang Li, and Calton Pu. Modeling unintended personal-information leakage from multiple online social networks. *IEEE Internet Computing*, 15(3):13–19, May 2011.

[84] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Commun. ACM*, 50(10):94–100, October 2007.

[85] Anil K. Jain, Arun Ross, and Shankar Pankanti. Biometrics: a tool for information security. *IEEE Transactions on Information Forensics and Security*, 1(2):125–143, 2006.

[86] Markus Jakobsson, Erik Stolterman, Susanne Wetzel, and Liu Yang. Love and authentication. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 197–200, New York, NY, USA, 2008. ACM.

[87] Markus Jakobsson, Alex Tsow, Ankur Shah, Eli Blevis, and Youn-Kyung Lim. What instills trust? a qualitative study of phishing. In *Proceedings of the 11th International Conference on Financial cryptography and 1st International conference on Usable Security*, FC'07/USEC'07, pages 356–361, Berlin, Heidelberg, 2007. Springer-Verlag.

[88] Markus Jakobsson, Liu Yang, and Susanne Wetzel. Quantifying the security of preference-based authentication. In *Proceedings of the 4th ACM workshop on Digital identity management*, DIM '08, pages 61–70, New York, NY, USA, 2008. ACM.

[89] Ian Jermyn, Alain Mayer, Fabian Monrose, Michael K. Reiter, and Aviel D. Rubin. The design and analysis of graphical passwords. In *Proceedings of the 8th conference on USENIX Security Symposium - Volume 8*, SSYM'99, pages 1–1, Berkeley, CA, USA, 1999. USENIX Association.

[90] Xin Jin, Chi Wang, Jiebo Luo, Xiao Yu, and Jiawei Han. Likeminer: a system for mining the power of 'like' in social media networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 753–756, New York, NY, USA, 2011. ACM.

[91] Kareem J. Johnson and Barbara L. Fredrickson. We all look the same to me: Positive emotions eliminate the own-race bias in face recognition. *Psychological Science*, 16:875–881, 2005.

[92] Lewis JR. Sample sizes for usability studies: additional considerations. In *Human Factors*, volume 36, pages 368–378, Boca Raton, Florida 33429-1328, Jun 1994. International Business Machines, Inc.

[93] Mike Just. Designing and evaluating challenge-question systems. *Security & Privacy Magazine, IEEE*, 2(5):32–39, 2004.

[94] Mike Just and David Aspinall. Personal choice and challenge questions: a security and usability assessment. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pages 8:1–8:11, New York, NY, USA, 2009. ACM.

[95] Saul Kassin and Lawrence S. Wrightsman. *The psychology of evidence and trial procedure*, chapter Confession evidence, pages 67 – 94. Beverly Hills: Sage, 1985.

[96] Saul M. Kassin and Katherine L. Kiechel. The social psychology of false confessions: Compliance, internalization, and confabulation. *Psychological Science*, 7(3):pp. 125–128, 1996.

[97] Mark Keith, Benjamin Shao, and Paul John Steinbart. The usability of passphrases for authentication: An empirical field study. *Int. J. Hum.-Comput. Stud.*, 65(1):17–28, January 2007.

[98] Byoungsoo Kim, Minnseok Choi, and Ingoo Han. User behaviors toward mobile data services: The role of perceived fee and prior experience. *Expert Syst. Appl.*, 36:8528–8536, May 2009.

[99] Jonghwa Kim. Bimodal emotion recognition using speech and physiological changes. In *In M. Grimm, K. Kroschel (Ed.), Robust Speech Recognition and Understanding*, pages 265–280. I-Tech Education and Publishing, Vienna, Austria, 2007.

[100] K. H. Kim1, S. W. Bang, and S. R. Kim. Emotion recognition system using short-term monitoring of physiological signals. *Medical and Biological Engineering and Computing*, 42(3):419–427, May 2004.

[101] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems, 2009.

[102] Ilker Korkmaz and Mehmet Emin Dalkilic. The weak and the strong password preferences: a case study on turkish users. In *Proceedings of the 3rd international conference on Security of information and networks*, SIN '10, pages 56–61, New York, NY, USA, 2010. ACM.

[103] Manu Kumar, Tal Garfinkel, Dan Boneh, and Terry Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd symposium on Usable privacy and security*, SOUPS '07, pages 13–19, New York, NY, USA, 2007. ACM.

[104] Philip Dean Lapsley, Jonathan Alexander Lee, David Ferrin Pare, Jr., and Ned Hoffman. Anti-fraud biometric scanner that accurately detects blood flow. US Patent # 5737439, 1998.

[105] Srivatsan Laxman, P. S. Sastry, and K. P. Unnikrishnan. A fast algorithm for finding frequent episodes in event streams. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 410–419, New York, NY, USA, 2007. ACM.

[106] C. K. Lee, S. K. Yoo, Yoonj Park, Namhyun Kim, Keesam Jeong, and Byungchae Lee. Using neural network to recognize human emotions from heart rate variability and skin resistance. In *27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005*, pages 5523–5525, 2005.

[107] Amanda Lenhart, Mary Madden, Aaron Smith, Kristen Purcell, Kathryn Zickuhr, and Lee Rainie. Teens, kindness and cruelty on social network sites. `http://pewinternet.org/~/media//Files/Reports/2011/PIP_Teens_Kindness_Cruelty_SNS_Report_Nov_2011_FINAL_110711.pdf`.

[108] Shujun Li and Heung-Yeung Shum. Secure human-computer identification (interface) systems against peeping attacks: Sechci, 2005.

[109] Jack Lindamood, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. Inferring private information using social network data. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 1145–1146, New York, NY, USA, 2009. ACM.

[110] David Lo, Siau-Cheng Khoo, and Limsoon Wong. Non-redundant sequential rules – theory and algorithm. *Information Systems*, 34:438–453, 2009.

[111] Anthony J. Maeder, Clinton B. Fookes, and Sridha Sridharan. Gaze based user authentication for personal computer applications. In *International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 727–730, Hong Kong, China, October 2004. IEEE.

[112] Hirokazu Manabe, Yuta Yamakawa, Tomohiro Sasamoto, and Ryoichi Sasaki. Security evaluation of biometrics authentications for cellular phones. *Intelligent Information Hiding and Multimedia Signal Processing, International Conference on*, 0:34–39, 2009.

[113] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov.*, 1(3):259–289, January 1997.

[114] George Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information, 1956. One of the 100 most influential papers in cognitive science: http://cogsci.umn.edu/millennium/final.html.

[115] Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 251–260, New York, NY, USA, 2010. ACM.

[116] Fabian Monrose, Michael K. Reiter, Qi Li, and Susanne Wetzel. Cryptographic key generation from voice(extended abstract). In *SP '01: Proceedings of the 2001 IEEE Symposium on Security and Privacy*, page 202, Washington, DC, USA, 2001. IEEE Computer Society.

[117] Fabian Monrose, Michael K. Reiter, and Susanne Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.

[118] I Nachson and B Feldman. Psychological stress evaluator - validity study. *Crime and Social Deviance*, 7(2):65–81, 1979.

[119] Karthik Nandakumar, Anil K. Jain, and Sharath Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Transactions on Information Forensics and Security*, 2(4):744–757, 2007.

[120] Karthik Nandakumar, Abhishek Nagar, and Anil K. Jain. Hardening fingerprint fuzzy vault using password. In *ICB07*, pages 927–937. Springer Berlin / Heidelberg, 2007.

[121] Fatma Nasoz, Kaye Alvarez, L. Lisetti, and Neal Finkelstein. Emotion recognition from physiological signals using wireless sensors for presence technologies. *Cognition, Technology and Work*, 6(1):4–14, 2004.

[122] Jakob Nielsen and Thomas K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, CHI '93, pages 206–213, New York, NY, USA, 1993. ACM.

[123] Peng Ning, Sushil Jajodia, and Xiaoyang Sean Wang. Abstraction-based intrusion detection in distributed environments. *ACM Trans. Inf. Syst. Secur.*, 4:407–452, November 2001.

[124] Ann Nosseir and Sotirios Terzis. A study in authentication via electronic personal history questions. In *ICEIS (5)*, pages 63–70, 2010.

[125] Jeremiah Owyang. The many challenges of social network sites, 2008.

[126] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[127] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51:3448–3470, August 2007.

[128] Koksoon Phua, Jianfeng Chen, Tran Huy Dat, and Louis Shue. Heart sound as a biometric. *Pattern Recogn.*, 41(3):906–919, 2008.

[129] Rosalind W. Picard, Elias Vyzas, and Jennifer Healey. Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE Transaction Pattern Analysis Matching Intelligence*, 23(10):1175–1191, 2001.

[130] Salil Prabhakar, Sharath Pankanti, and Anil K. Jain. Biometric recognition: Security and privacy concerns. *IEEE Security and Privacy*, 1(2):33–42, 2003.

[131] Ariel Rabkin. Personal knowledge questions for fallback authentication: security questions in the era of facebook. In *Proceedings of the 4th symposium on Usable privacy and security*, SOUPS '08, pages 13–23, New York, NY, USA, 2008. ACM.

[132] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. An analysis of minutiae matching strength. In *AVBPA '01: Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 223–228, London, UK, 2001. Springer-Verlag.

[133] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Syst. J.*, 40:614–634, March 2001.

[134] Robots.txt. A standard for robot exclusion. `http://www.robotstxt.org/orig.html`.

[135] Volker Roth, Kai Richter, and Rene Freidinger. A pin-entry method resilient against shoulder surfing. In *Proceedings of the 11th ACM conference on Computer and communications security*, CCS '04, pages 236–245, New York, NY, USA, 2004. ACM.

[136] Kirk Sakai and Timothy W. Quick. Moisturizing skin preparation, July 1988.

[137] Hirokazu Sasamoto, Nicolas Christin, and Eiji Hayashi. Undercover: authentication usable in front of prying eyes. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 183–192, New York, NY, USA, 2008. ACM.

[138] M. A. Sasse, S. Brostoff, and D. Weirich. Transforming the 'weakest link' – a human/computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3):122–131, July 2001.

[139] Stuart Schechter, Serge Egelman, and Robert W. Reeder. It's not what you know, but who you know: a social approach to last-resort authentication. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 1983–1992, New York, NY, USA, 2009. ACM.

[140] Stefan Schmidt and Harald Walach. Electrodermal activity (eda) - state of the art measurement and techniques for parapsychological purposes. *Journal of Parapsychology*, 64:139 – 163, June 2000.

[141] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni. A fast automaton-based method for detecting anomalous program behaviors. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, pages 144–155, May 2001.

[142] Hans Selye. *The Stress of Life*, chapter 1-7. McGraw-Hill, 1956.

[143] Yishay Spector and Jacob Ginzberg. Pass-sentence -Ůa new approach to computer code. *Comput. Secur.*, 13(2):145–160, April 1994.

[144] Myra Spiliopoulou. Managing interesting rules in sequence mining. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases*, 1999.

[145] K. Tan and R. Maxion. "Why 6?"– Defining the operational limits of stide, an anomaly-based intrusion detector. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 188–201, May 2002.

[146] Journey to the Wild Divine. *Skin conductance aquisition device, Lightstone*. `http://www.wilddivine.com/`.

[147] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[148] Antoine Galland (translator). *The Arabian Nights Entertainments*, volume 17. Harrison and Company, 1785.

[149] Alexia Tsotsis. Hacker proves facebook's public data is public. http://techcrunch.com/2010/07/28/hacker-proves-facebooks-public-data-is-public/.

[150] Alexia Tsotsis. Twitter revokes automatic 3rd party dm access, gives users more details on app permissions. TechCrunch. http://techcrunch.com/2011/05/18/twitter-revokes-automatic-3rd-party-dm-access-gives-users-more-details-on-app-permissions/.

[151] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[152] UIDIA. Unique identification authority of india card project-india. http://www.uidaicards.com/.

[153] U. Uludag and A.K. Jain. Attacks on biometric systems: a case study in fingerprints. In *Proc. SPIE-EI 2004, Security, Seganography and Watermarking of Multimedia Contents VI*, pages 622 – 633, 2004.

[154] Diana Valentine. Skin conductance one of the fastest ways to test stress, 2009. http://www.articlesbase.com/health-articles/skin-conductance-one-of-the-fastest-ways-to-test-stress-1464442.html, [Online; accessed 16-November-2009].

[155] Alex Waibel and Kai-Fu Lee, editors. *Readings in speech recognition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[156] Jun Wang, Lijun Yin, Xiaozhou Wei, and Yi Sun. 3d facial expression recognition based on primitive surface feature distribution. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:1399–1406, 2006.

[157] Daphna Weinshall. Cognitive authentication schemes safe against spyware (short paper). In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, SP '06, pages 295–300, Washington, DC, USA, 2006. IEEE Computer Society.

[158] Joyce H. D. M. Westerink, Egon L. van den Broek, Marleen H. Schut, Jan van Herk, and Kees Tuinenbreije. *Computing Emotion Awareness Through Galvanic Skin Response and Facial Electromyography*, volume 8 of *Philips Research Book Series*. Springer Netherlands, New York, December 2007.

[159] Alma Whitten and J. D. Tygar. Why johnny can't encrypt: a usability evaluation of pgp 5.0. In *Proceedings of the 8th conference on USENIX Security Symposium - Volume 8*, SSYM'99, pages 14–14, Berkeley, CA, USA, 1999. USENIX Association.

[160] Susan Wiedenbeck, Jim Waters, Leonardo Sobrado, and Jean-Camille Birget. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '06, pages 177–184, New York, NY, USA, 2006. ACM.

[161] Jason Wiese, Patrick Gage Kelley, Lorrie Faith Cranor, Laura Dabbish, Jason I. Hong, and John Zimmerman. Are you close with me? are you nearby?: investigating social groups, closeness, and willingness to share. In *Proceedings of the 13th international conference on Ubiquitous computing*, UbiComp '11, pages 197–206, New York, NY, USA, 2011. ACM.

[162] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433, 2009.

[163] Lawrence S. Wrightsman and Saul Kassin. *Confessions in the Courtroom*. SAGE Publications, Inc, 1993.

[164] Min Wu, Simson Garfinkel, and Rob Miller. Secure web authentication with mobile phones. In *DIMACS Workshop on Usable Privacy and Security Software*, 2004.

[165] Min Wu, Robert C. Miller, and Simson L. Garfinkel. Do security toolbars actually prevent phishing attacks? In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 601–610, New York, NY, USA, 2006. ACM.

[166] Yahoo. Rate limiting for yahoo! search web services. `http://developer.yahoo.com/search/rate.html`.

[167] Roman V. Yampolskiy. Human computer interaction based intrusion detection. *Information Technology: New Generations, Third International Conference on*, 0:837–842, 2007.

[168] Jeff Yan, Alan Blackwell, Ross Anderson, and Alasdair Grant. Password memorability and security: Empirical results. *IEEE Security and Privacy*, 2(5):25–31, September 2004.

[169] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 537–546, New York, NY, USA, 2011. ACM.

[170] Huanyu Zhao and Xiaolin Li. S3pas: A scalable shoulder-surfing resistant textual-graphical password authentication scheme. In *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, volume 2, pages 467 –472, may 2007.

[171] Elena Zheleva and Lise Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 531–540, New York, NY, USA, 2009. ACM.

[172] Konglin Zhu, Pan Hui, Yang Chen, Xiaoming Fu, and Wenzhong Li. Exploring user social behaviors in mobile social applications. In *Proceedings of the 4th Workshop on Social Network Systems*, SNS '11, pages 3:1–3:6, New York, NY, USA, 2011. ACM.

[173] Moshe Zviran and William J. Haga. Cognitive passwords: the key to easy access control. *Comput. Secur.*, 9(9):723–736, January 1991.

Appendices

Appendix A

Cellphone usage patterns

Understanding cellphone usage is not a main contribution of this work. Nonetheless, we present the usage statistics obtained by our month long data collection process in the hope that they will be useful to other researchers and projects.

Categories	Android		Symbian	
	μ	σ	μ	σ
Total, Per User, for the entire Data Collection Period				
Total days of data logged	30.2	3.9	49.3	2.6
Contacts in phonebook	501.2	342.0	322.4	182.2
Total installed applications	161.3	51.6	34.0	13.3
Total Images files in the media folder	157.5	115.5	–	–
Total Audio files in the media folder	358.7	382.3	–	–
Per Day, Per User				
SMSes (sent/received)	16.93	14.31	42.08	31.51
Calls (incoming/outgoing/missed)	6.28	3.13	10.11	5.13
Emails (sent/received)	9.1	9.1	–	–
Chats (individual chat messages)	7.4	13.2	–	–
New applications installed	0.9	0.6	–	–
New image files added to media folder	1.6	2.5	–	–
New audio files added to media folder	4.7	7.2	–	–
Connection attempts to WiFi network	1.5	1.1	2.3	7.6
Power on (restarting the phone)	2.0	1.1	1.6	1.1
Switch on Bluetooth	0.0	0.1	0.1	0.1
Websites visited using the browser	9.2	8.6	–	–
Searches performed	1.5	0.6	–	–

Table A.1: Participants' data usage

Table A.1 shows the usage patterns of all the participants in our user study,

divided into Symbian and Android columns. For each column, the mean (μ) and standard deviation (σ) values are provided.

First, we observe that, on average, data was collected for 30 days from Android participants, and 49 days for Symbian participants. We then notice that even though Android participants had more contacts (501 to 322 on average) than Symbian participants, they sent far less SMSes (16.93 versus 42.08 daily) and calls (6.28 versus 10.11 daily) than Symbian participants. This disparity is because Android users have access to other communication channels on their phones, such as instant messaging, emails, Skype. Moreover the total number of applications installed was higher on Android (161.3 to 34 on average) than Symbian.

Application Category	**Average No. of Apps installed on phone (per user)**
Tools	14.69
Games	13.23
Lifestyle	10.38
Utilities	4.00
Social	4.00
News	3.85
Reference	3.31
Productivity	2.85
Widget	2.54
Map	2.46
Internet	1.31
Video	1.15
Comms	0.69
Finance	0.31

Table A.2: Applications usage breakdown for Android

Table A.2 shows the types of applications installed on the Android phones. It provides the average number of applications, of each category type, installed on the phones of the 13 Android participants. The categories were obtained from the Android Marketplace. The two tables reiterate the common wisdom that modern smartphones are used heavily for consuming media content. For example, Android users have more media content on their phones along with numerous game and lifestyle applications.

Appendix B

Guessing entropy for skin conductance

Let R be the set of rules the attacker can use to reduce the password space from S to S'. So, for a rule R_i

$$antecedent(\mathsf{A}) \Rightarrow consequent(\mathsf{C})$$

such that, $\mathsf{A}=[y_1, \ldots, y_{E_a}]$ and $\mathsf{C}=[z_1, \ldots, z_{E_c}]$, where E_a are the elements in the antecedent and E_c in consequent. The process of calculating the new password space from a given one is shown in algorithm 2. S' indicates a lower bound for the password space which shows the minimum number of combinations an attacker needs to guess if he has a full knowledge of the mappings in the database.

Let Ψ denote the candidate set and Φ be the large itemset, Ψ^I and Φ^I are the two dimensional vectors derived from the rules $R_1, \ldots, R_I$. Each item (Ψ^I_J) in a Ψ^I is a vector of the form $[x_1, x_2, \ldots, x_{m_{SC}}]$, $\forall\ 0 \leq J \leq L$, where $x_i \in (0, 1, *)$ and $L = |\Psi^I|$. Similarly, each item (Φ^I_J) in a Φ^I is also vector of the form $[x_1, x_2, \ldots, x_{m_{SC}}]$, $\forall\ 0 \leq J \leq L$, where $x_i \in (0, 1, *)$ and $L = |\Phi^I|$.

$*$ denotes *don't care* and can be assigned 0 or 1. The set of rules R obtained are passed to the algorithm 2 to generate S'. Φ^0_1 is initialized to $[* * * * * * * \ldots *]$ and $\mathsf{S} = 2^{m_{SC}}$. Below is the short description of the functions used in the algorithm.

Algorithm 2 Reduced Password Space for SC

PasswdSpace (R)

1: $\Phi_1^0 \longleftarrow [*, *, *, *, *, *, *, *, \ldots, *]$
2: $\mathsf{S} \longleftarrow 2^{m_{SC}}$
3: **for** I = 1 to $|R|$ **do**
4: $\quad \mathrm{L} \longleftarrow \mathbf{length}(\Phi^{I-1})$
5: $\quad \Psi^I \longleftarrow$ NULL
6: $\quad$ **for** J = 1 to L **do**
7: $\quad\quad$ **if any** $\left((\Phi_{J,y_1}^{I-1}, \Phi_{J,y_2}^{I-1}, \ldots, \Phi_{J,y_{E_a}}^{I-1}) == *\right)$ **then**
8: $\quad\quad\quad \Psi^I \longleftarrow \Psi^I \bigcup \mathbf{split}(\Phi_J^{I-1})$
9: $\quad\quad$ **else**
10: $\quad\quad\quad \Psi^I \longleftarrow \Psi^I \bigcup \Phi^{I-1}$
11: $\quad\quad$ **end if**
12: $\quad$ **end for**
13: $\quad \Psi^I \longleftarrow \mathbf{unique}\left(\Psi^I\right)$
14: $\quad$ cnt $\longleftarrow 1$
15: $\quad \mathrm{L} \longleftarrow \mathbf{length}(\Psi^I)$
16: $\quad$ **for** J = 1 to L **do**
17: $\quad\quad$ **if** $\left(\prod_{p=1}^{E_a} \Psi_{J,y_p}^I == 1 \& \prod_{q=1}^{E_c} \Psi_{J,z_q}^I == 0\right)$ **then**
18: $\quad\quad\quad \mathbf{delete}\left(\Psi_J^I\right)$
19: $\quad\quad$ **else**
20: $\quad\quad\quad \Phi_{cnt}^I \longleftarrow \Psi_J^I$
21: $\quad\quad\quad$ cnt++
22: $\quad\quad$ **end if**
23: $\quad$ **end for**
24: **end for**
25: $\mathsf{S}' \longleftarrow \Phi^{|R|}$

- $\mathbf{length}(\Psi^I)$ - gives the total vectors in the candidate set Ψ^I i.e. $|\Psi^I|$.
- $\mathbf{any}(\Phi^J_I(y_1, y_2, \ldots, y_{E_a}) == *)$ - a boolean function

$$= \begin{cases} \mathbf{1}, & (\Phi^I_{J,y_1} == *) \vee \ldots \vee (\Phi^I_{J,y_{Ea}}) == *) \\ \mathbf{0}, & \text{else} \end{cases}$$

- $\mathbf{split}(\Phi^I_J)$ - this function generates a new candidate set Ψ^I from a large itemset Φ^{I-1} based on a rule R_I. It generates the vectors for Ψ^I *s.t.*
  - Mark y_i, if $(\Phi^{I-1}_{J,y_i} == *)$, $\forall\ y_i \in [y_1, \ldots, y_{E_a}]$.
  - Generate all possible combination of the marked bits; which implies if total number of marked bits are mb then total possible combinations are 2^{mb}. For e.g. if Φ^I_J = [***1*1] and the rule R_I is $1 \Rightarrow 2$, then the result is $([11*1*1]\ [10*1*1]\ [01*1*1]\ [00*1*1])$
- $\mathbf{unique}(\Psi^I)$ - gives the unique vectors from Ψ^I.
- $\mathbf{delete}(\Psi^I_J)$ - delete Ψ^I_J from the candidate set Ψ^I.

During the Candidate Itemset Generation, a $*$ in the large itemset triggers a $split$; 1 and 0 indicates *do nothing*. However during the Large Itemset Generation a 1 in a candidate itemset triggers *add 1*; 0 indicates *do nothing*. During the whole procedure, each time one rule is used and the sets which does not comply with that rule are omitted to create the new set. The final password space is calculated by computing the total number of vectors which can be generated using $\Phi^{|R|}$, where $\Phi^{|R|}$ is the final large itemset generated from the rules $R_1, \ldots, R_{|R|}$.

An example shown in Table B.1 with *5* elements, to how to generate the candidate itemset and the large itemset from *3* rules. The total number of guesses which an attacker needs to make is *14* which implies the effective number of bits in the new password space are *4*; original was *5*.

R		Candidate Itemset		Large Itemset
Initialization			Φ_1^0	$*****$
$\mathbf{R_1}$ $1 \Rightarrow 3$	Ψ_1^1	$1****$	Φ_1^1	$1*1**$
	Ψ_2^1	$0****$	Φ_2^1	$0****$
$\mathbf{R_2}$ $(1,2) \Rightarrow 5$	Ψ_1^2	$101**$	Φ_1^2	$101**$
	Ψ_1^2	$111**$	Φ_1^2	$111*1$
	Ψ_2^2	$01***$	Φ_2^2	$01***$
	Ψ_3^2	$00***$	Φ_3^2	$00***$
$\mathbf{R_3}$ $5 \Rightarrow 1$	Ψ_1^3	$101*0$	Φ_1^3	$101*0$
	Ψ_2^3	$101*1$	Φ_2^3	$101*1$
	Ψ_3^3	$111*1$	Φ_3^3	$111*1$
	Ψ_4^3	$01**1$	Φ_4^3	$01**0$
	Ψ_5^3	$01**0$	Φ_5^3	$00**0$
	Ψ_6^3	$00**1$		
	Ψ_7^3	$00**0$		

Table B.1: Generating candidate set and large itemset