Your Culture is in Your Password: An Analysis of a Demographically-diverse Password Dataset

Mashael AlSabah[a,*], Gabriele Oligeri[b,*], Ryan Riley[c,*]

[a] *Qatar Computing Research Institute, Hamad Bin Khalifa University*
[b] *Hamad Bin Khalifa University, College of Science and Engineering, Division of Information and Computing Technology*
[c] *Computer Science Program, Carnegie Mellon University in Qatar*

Abstract

A large number of studies on passwords make use of passwords leaked by attackers who compromised online services. Frequently, these leaks contain only the passwords themselves, or basic information such as usernames or email addresses. While metadata-rich leaks exist, they are often limited in the variety of demographics they cover.

In this work, we analyze a meta-data rich data leak from a Middle Eastern bank with a demographically-diverse user base. We provide an analysis of passwords created by groups of people of different cultural backgrounds, some of which are under-represented in existing data leaks, e.g., Arab, Filipino, Indian, and Pakistani.

The contributions provided by this work are many-fold. First, our results contribute to the existing body of knowledge regarding how users include personal information in their passwords. Second, we illustrate the differences that exist in how users from different cultural/linguistic backgrounds create passwords. Finally, we study the (empirical and theoretical) guessability of the dataset based on two attacker models, and show that a state of the art password strength estimator inflates the strength of passwords created by users from non-English speaking backgrounds. We improve its estimations by training it

*This work was performed while all three authors were at Qatar University

Email addresses: msalsabah@hbku.edu.qa (Mashael AlSabah), goligeri@hbku.edu.qa (Gabriele Oligeri), rileyrd@cmu.edu (Ryan Riley)

with contextually relevant information.

---

1. Introduction

Password-based authentication is the most widely deployed mechanism to protect users' accounts and personal information on web-based services around the world. Web services rely on passwords to authenticate and authorize millions of users daily to access their email, perform financial transactions, interact with government agencies, communicate confidential data, perform sensitive transactions, and more. There has been a significant amount of research regarding how users choose passwords as well as how secure their choices are [1, 2, 3, 4, 5, 6, 7, 8].

Much of the research on passwords makes use of passwords leaked by attackers who have compromised online services. Frequently these leaks contain only the passwords themselves, or also include a basic username or email address. While these sorts of leaks allow for thorough analysis of the passwords themselves, the lack of information about the password creators prevents researchers from gaining insight into how a user may incorporate their personal information into their password choice. In addition, it also limits an analysis of how a user's linguistic or cultural background impacts password choices. While more interesting datasets containing personal information do exist [9], such leaks are limited in that they cover limited demographic groups (i.e. Chinese, or advanced English users).

In April of 2016, a data leak was released on social media purported to be the customer database for a major Middle Eastern bank. The leak contained detailed information on about 400,000 of the bank's customers, including hashed passwords for the 100,000 of those customers who had configured online banking. While the existence of the leaked data represents a tremendous breach of user privacy, it also opens up a unique opportunity for analyzing and comparing passwords of users that come from different demographic and cultural backgrounds.

Despite its small size, this dataset has some unique properties that allows us to gain insights not found from analyzing prior leaks:

- The data contains rich meta-data about the users who created the passwords, such as names, phone numbers, emails, addresses, recovery questions and answers, and more.
- The data includes nationality information, allowing analysis based on cultural/linguistic backgrounds.
- The dataset represents banking passwords, which in theory users may care more about that typical leaks representing web forums or entertainment websites.

Ethical Considerations. There is an arms race between the security research community and attackers. Hackers and criminal groups benefit from studying publicly accessible corpuses of password lists to improve their attacks. Therefore, there is a pressing need for the security research community to improve the defences. Understanding and analyzing how different demographic groups choose their passwords will give important insights on the security weaknesses of such passwords, and will shed light on how to enhance the security of those users against password compromise attacks.

Given that this work involves leaked passwords and significant amounts of individually identifying personal information, we sought and obtained an Institutional Review Board (IRB) waiver. As part of receiving the IRB waiver, we agreed to:

1. Follow best practices with regards to storing the information in an encrypted format.
2. Not further disseminate the leaked information, which includes sharing or submitting passwords or password hashes to external services.
3. Not explicitly name the financial institution involved.
4. Not release any information that could be used to identify any individual included in the leak.

5. Not intentionally identify individuals within the leak. This includes initiating contact with individuals whose information is contained within the leak.

Throughout the paper, we discuss ethical issues that may arise due to to our analysis.

Contributions. To our knowledge, this is the first password study that analyses a password dataset that is both rich with personal information as well as demographically diverse, which allows us to get insights which were not possible in prior studies. This work offers the following contributions:

- We divide the leaked password datasets into 4 demographic groups, which we analyze extensively to provide insights into how different groups construct their passwords.
- We provide insights on the extent to which users from those groups incorporate personal information in their passwords and highlight the differences observed between the groups.
- We highlight the most popular security questions chosen by users.
- We study the guessability of the passwords using empirical and theoretical metrics, and show that a current, state-of-the-art password strength estimator, zxcvbn [10], overestimates the strength of passwords created by users from non-English speaking backgrounds. We improve its performance by training it with contextually relevant data.

Roadmap. The remainder of this paper is organized as follows. Section 2 describes the leaked dataset, and how we divide it into multiple smaller datasets based on demographics. Section 3 presents our extensive demographic-based analysis, and elaborates on how we quantify the amount of personal information in users' passwords. Section 4 presents two attack models and evaluates the guessability of the password datasets and shows how we can improve the empirical results. Section 5 provides a throughout discussion on our findings.

Finally, Section 6 summarizes previous work and compares it with our work, and Section 7 provides some concluding remarks.

2. Dataset Information

In April of 2016 a hacking group released a dataset on social media purported to contain the customer database for a large, Middle Eastern bank based in Qatar. Following the leak, the bank released a statement saying that they do not comment on reports circulated on social media, and assuring their customers that there was no financial impact.

2.1. Contents

The leaked data contains a variety of tables from the customer database for the victim bank, of which two tables are of interest in this study: The customer master database and the online banking user profile database.

The customer master database contains personal information for 403,870 customers. This includes information such as: Name, gender, date of birth, phone numbers (home, work, mobile, fax), address, email, city of birth, government ID number, and nationality. Nationality is stored using the numeric country identifiers listed in ISO 3166 [11]. Not all fields are filled in for every customer, but most are. The data set includes a mix of personal and corporate customers.

The online banking user profile database contains further information for 97,674 accounts that have been configured for online web or mobile banking. This includes information such as: Username, password hash, security question, and the hash of the answer to the security question.

2.2. Verifying the Information

Given the source of the leaked data, it is prudent to perform some sort of manual verification that the data is indeed legitimate. After the leak occurred, numerous bank customers contacted the authors to ask for assistance in determining if their information was contained in the leak. Of the 50 or so customers we talked to, all but one were included in the leak. The one who was not included was a recent customer, and as such we hypothesize that the leaked information was stolen prior to his joining. After seeing the information included about them, none of the customers pointed out any incorrect information. For this reason we believe the information in the leak to be genuine.

We also performed an informal analysis using population statistics for Qatar [12], as well as our estimated market share of the bank, leading us to the further conclusion that the leak contains information for the bank's entire customer database.

In summary, we believe the leak is genuine and contains information for all of the bank's customers.

2.3. Relevant Background and Context

In order to help explain the context of the data, we now provide a brief overview of the process a new customer goes through to create an account at the bank in question. The population of Qatar is unique when compared to most countries. It is a small nation that is experiencing rapid growth thanks to its large reserves of natural gas, and because of this most residents of the country are not citizens, instead they are expatriates. (Only about 12% of the population are citizens [12].)

All of these factors combined mean that most new customers to the bank are also new to the country. Typically, opening a bank account is one of the first things a new resident does. (Without a local bank account, she cannot receive her salary.) When opening the new account, the customer provides a copy of their passport or residence visa in order to prove identity and certain personal information such as nationality. Typically, the account is opened immediately and the customer is provided with a new debit card and PIN number. Later, at their leisure, customers may signup for online banking using a computer or an app on their phone. At this time, the user selects a username, password, security question, and security question answer. There are certain rules enforced on the password related to length and required character sets, but these rules have changed over time and as such the same rules have not been applied to all

Table 1: Password Recovery Rate by Nationality

Country	**Recovered Passwords**	**Total Passwords**	**Recovery Rate**
Qatar	18945	21281	89.02%
India	13071	17170	76.13%
Philippines	7122	8821	80.74%
Egypt	5754	7115	80.87%
United Kingdom	3036	3491	86.97%
Lebanon	2454	3101	79.14%
United States	2315	2790	82.97%
Pakistan	1909	2417	78.98%
Canada	1794	2119	84.66%
Jordan	1738	2098	82.84%
Syrian Arab Republic	1601	2048	78.17%

customers. However, it is reasonable to assume that passwords were required to be 8 characters or more and contain both letters and numbers.

2.4. Password Recovery

Both the passwords and the security question answers are hashed with unsalted MD5. In order to perform the analysis of the passwords, the password cracking tools John the Ripper [13] and Hashcat [14] were used on the hashed passwords over a period of about three months. Whenever possible, GPU acceleration with a single NVidia GTX 980 was used to speed-up the process. We applied a variety of dictionaries for multiple languages, a significant number of transformation rules (those included with HashCat and others taken from KoreLogic [15]), and applied a complete bruteforce for up to 8 characters.

In the end, we recovered passwords for 79,760 accounts, which is an 81.66% recovery rate overall. The recovery rates were not even distributed among nationalities, however. Table 1 shows the recovery rates for the 10 largest nationalities in the dataset. As can be seen, there is a roughly 10 percentage point difference between the best and worst recovery rates.

2.5. Demographics

Previous work observed that languages have a significant impact on passwords [2]. While there has been some work providing insights about Chinese passwords [4, 16, 16], and how they differ from English-based ones, little or no existing work (to the best of our knowledge) characterizes passwords formed by users speaking other languages (specifically the ones we have in our dataset). For this reason, we seek to divide our dataset into different groups based on their linguistic/cultural background, as determined by their recorded nationality. While we were not able to verify the accuracy of the nationality registered for each user ourselves, it is important to note that nationality is determined by the bank using the customer's passport when the account is initially opened.

We created the four different groupings found in Table 2:

- D1: Arabic speakers. This group contains individuals from 16 different countries where Arabic is the official language and the people can, generally, be considered Arab.
- D2: India and Pakistan. These two countries, despite being very diverse in terms of language and religion, share many common cultural traits and history, making them suitable to group together.
- D3: Philippines.
- D4: English speakers. This group contains individuals from five countries where English is the native language.

While linguistic/cultural-based grouping may seem rather simple, it actually has the following advantages. First, it allows us to understand the impact of language on passwords and observe patterns in password habits that are specific to some groups, as we show in Section 3. Second, language-based groupings allows us to test the hypothesis if current password strength meters provide reasonable results for non-English users. This is an important message to the security community and stakeholders in different countries which currently use off the shelf products.

These four demographic groups include 65,941 customers for whom we have

Table 2: Four Demographic Groupings

Grouping	Countries	Number of Passwords
D1	Algeria, Bahrain, Egypt, Iraq, Jordan, Kuwait, Lebanon, Morocco, Oman, Qatar, Saudi Arabia, State of Palestine, Sudan, Syrian Arab Republic, Tunisia, United Arab Emirates	35598
D2	India, Pakistan	14980
D3	Philippines	7122
D4	Australia, Canada, Ireland, New Zealand, United Kingdom, United States	8241

a cleartext password[1]. This covers users from 25 different countries out of the approximately 198 countries codes listed in the dataset. These 25 countries, however, contribute over 82% of the passwords.

There are a few important notes to be made about this demographic split. First, we do not claim that these groupings are optimal or ideal. Our decision to group in this way is born out of our hypothesis that, in general, linguistic/cultural groups create passwords differently, as has been also confirmed by previous research [2]. Second, this dataset does not contain a random sampling of users from the various nationalities. The non-Qatari customers of this bank are all expatriates living in Qatar, which biases the results. The average American living in Qatar, for example, is likely better educated than the average American in the USA due to the types of jobs than bring them to Qatar. Indians in Qatar are more likely to be from the regions of Kerala and Uttar Pradesh [12]. Similar biases exist for other nationalities. Third, separating people into linguistic, cultural, or religious groupings is an inherently error-prone process. For example, India is a large country containing a great many languages, cultures, and religions. In our case we grouped India and Pakistan together due to the

[1]From this point forward in the text, the reader should assume that all analyses are performed on this set of 65,941 customers, unless otherwise specified.

Table 3: Top 10 Passwords

D1	*D2*	*D3*	*D4*	RockYou	Xato
abcd1234 = 103 (0.29%)	abcd1234 = 20 (0.13%)	= 3 (0.04%)	qatar2013 = 6 (0.07%)	123456 = 290732 (0.89%)	123456 = 55893 (0.56%)
qwer1234 = 61 (0.17%)	1234abcd = 8 (0.05%)	= 2 (0.03%)	charlie1 = 5 (0.06%)	12345 = 79080 (0.24%)	password = 19580 (0.2%)
qatar123 = 43 (0.12%)	qatar123 = 8 (0.05%)	= 2 (0.03%)	abcde12345 = 4 (0.05%)	123456789 = 76794 (0.24%)	12345678 = 13582 (0.14%)
1234qwer = 39 (0.11%)	pakistan1 = 7 (0.05%)	= 2 (0.03%)	doha2011 = 4 (0.05%)	password = 59479 (0.18%)	qwerty = 13137 (0.13%)
asdf1234 = 37 (0.1%)	= 6 (0.04%)	= 2 (0.03%)	abcd1234 = 4 (0.05%)	iloveyou = 49960 (0.15%)	123456789 = 11696 (0.12%)
a1234567 = 34 (0.1%)	a1b2c3d4 = 6 (0.04%)	= 2 (0.03%)	deborah1 = 4 (0.05%)	princess = 33369 (0.1%)	12345 = 10938 (0.11%)
aaaa1111 = 25 (0.07%)	bismillah786 = 6 (0.04%)	= 2 (0.03%)	liverpool1 = 4 (0.05%)	1234567 = 21727 (0.07%)	1234 = 6432 (0.06%)
qatar2022 = 23 (0.06%)	asdf1234 = 6 (0.04%)	= 2 (0.03%)	allah111 = 3 (0.04%)	rockyou = 20918 (0.06%)	111111 = 5682 (0.06%)
1234abcd = 20 (0.06%)	india123 = 5 (0.03%)	= 2 (0.03%)	doha2012 = 3 (0.04%)	12345678 = 20554 (0.06%)	1234567 = 4796 (0.05%)
m1234567 = 19 (0.05%)	doha2010 = 5 (0.03%)	= 2 (0.03%)	paris123 = 3 (0.04%)	abc123 = 16648 (0.05%)	dragon = 3927 (0.04%)

Table 4: Top 10 Base Words

D1	*D2*	*D3*	*D4*	RockYou	Xato
qatar = 579 (1.63%)	qatar = 66 (0.44%)	june = 30 (0.42%)	qatar = 48 (0.58%)	password = 94488 (0.29%)	password = 23717 (0.24%)
ahmed = 237 (0.67%)	doha = 41 (0.27%)	qatar = 28 (0.39%)	doha = 38 (0.46%)	iloveyou = 71895 (0.22%)	qwerty = 19301 (0.19%)
mohd = 204 (0.57%)	abcd = 38 (0.25%)	march = 21 (0.29%)	london = 16 (0.19%)	love = 59671 (0.18%)	dragon = 6359 (0.06%)
abcd = 163 (0.46%)	pakistan = 27 (0.18%)	april = 19 (0.27%)	summer = 12 (0.15%)	princess = 58516 (0.18%)	alex = 5187 (0.05%)
sara = 157 (0.44%)	bismillah = 26 (0.17%)	july = 19 (0.27%)	liverpool = 12 (0.15%)	angel = 45775 (0.14%)	love = 5022 (0.05%)
hamad = 141 (0.4%)	jesus = 23 (0.15%)	august = 18 (0.25%)	omar = 12 (0.15%)	monkey = 33232 (0.1%)	monkey = 4869 (0.05%)
doha = 140 (0.39%)	khan = 22 (0.15%)	october = 17 (0.24%)	charlie = 11 (0.13%)	babygirl = 32043 (0.1%)	master = 4736 (0.05%)
nasser = 126 (0.35%)	india = 19 (0.13%)	nicole = 17 (0.24%)	canada = 11 (0.13%)	nicole = 31574 (0.1%)	shadow = 4560 (0.05%)
qwer = 125 (0.35%)	asdf = 18 (0.12%)	password = 17 (0.24%)	ahmed = 11 (0.13%)	soccer = 30530 (0.09%)	football = 4338 (0.04%)
khalid = 112 (0.31%)	sairam = 17 (0.11%)	december = 17 (0.24%)	alexander = 9 (0.11%)	rockyou = 27783 (0.09%)	michael = 4275 (0.04%)

similarities between Hindi and Urdu, but there are many sub groups within both countries that are not related to each other either linguistically or culturally.

Applying alternative grouping methodologies is an interesting topic for future work.

3. Demographic Based Analysis

In this section we present a variety of traditional password analyses, but break down the results using the demographic groups previously identified.

3.1. Common Passwords

In this section, we analyze information about common passwords, base words, and password length. In order to facilitate comparison with previous leaked passwords, we also show information for the Rock You [17] and Xato [18] datasets.

Ethical considerations. For ethical reasons, we hide the top passwords that can uniquely identify individuals. We only release top common passwords that contain common names, patterns, or locations. We observe that the top common passwords are similar to the ones released by previous analyses. We hide the *D3* passwords in particular because they contain individual names and birthdays. In order to comply with local IRB requirements, we also blinded one top password from *D2* because it contains the bank name.

Table 5: Password lengths

D1	*D2*	*D3*	*D4*	RockYou	Xato
8 = 14436 (40.55%)	8 = 4999 (33.37%)	8 = 2216 (31.11%)	8 = 3493 (42.39%)	6 = 8497562 (26.06%)	8 = 2980856 (29.81%)
9 = 9476 (26.62%)	9 = 4048 (27.02%)	9 = 1827 (25.65%)	9 = 2142 (25.99%)	8 = 6504916 (19.95%)	6 = 2543979 (25.44%)
10 = 6719 (18.87%)	10 = 2874 (19.19%)	10 = 1440 (20.22%)	10 = 1404 (17.04%)	7 = 6284712 (19.28%)	7 = 1662856 (16.63%)
11 = 2588 (7.27%)	11 = 1617 (10.79%)	11 = 817 (11.47%)	11 = 659 (8.0%)	9 = 3938519 (12.08%)	9 = 680812 (6.81%)
12 = 1309 (3.68%)	12 = 830 (5.54%)	12 = 456 (6.4%)	12 = 318 (3.86%)	10 = 2943315 (9.03%)	5 = 494999 (4.95%)
13 = 556 (1.56%)	13 = 354 (2.36%)	13 = 198 (2.78%)	13 = 135 (1.64%)	5 = 1343832 (4.12%)	10 = 471284 (4.71%)
14 = 259 (0.73%)	14 = 154 (1.03%)	14 = 96 (1.35%)	14 = 55 (0.67%)	11 = 1151591 (3.53%)	4 = 345142 (3.45%)
15 = 101 (0.28%)	15 = 55 (0.37%)	15 = 44 (0.62%)	15 = 17 (0.21%)	12 = 677835 (2.08%)	11 = 263464 (2.64%)
16 = 50 (0.14%)	16 = 32 (0.21%)	16 = 9 (0.13%)	16 = 9 (0.11%)	13 = 423204 (1.3%)	12 = 190978 (1.91%)
7 = 48 (0.13%)	7 = 6 (0.04%)	17 = 7 (0.1%)	17 = 3 (0.04%)	14 = 276029 (0.85%)	13 = 135586 (1.36%)

Most frequent passwords. Table 3 shows the top ten passwords for each dataset. For *D1*, *D2*, and *D4*, one can see that the most repeated passwords consist mostly of keyboard and number sequences, or patterns. It is also important to note that, as a whole, there is actually very little password reuse within this dataset, especially when compared to Rock You and Xato. The most common password overall, *abcd1234* is used less than 200 times across all groups. We hypothesize this is due to a combination of the fact that users are more careful with financial passwords than others, and the fact that the bank enforced more strict password rules than the sources of previous leaks.

Most frequent base words. To compare these groups in terms of the most used tokens or patterns, we also extract the top ten base words, and present them in Table 4. In order to perform the above operation, we made use of *pipal*, a well-know tool for password analysis [19]. We observe that for datasets *D1*, *D2*, and *D4*, the common base word lists consist mainly of locations and names. One can see that "qatar" is the most common base word across three of the datasets, and the second most common for the *D3* dataset. The base word "doha", the capital of Qatar, is also very common in the *D1*, *D2*, and *D4* datasets. Overall, the base word "qatar", and "doha" occurred 863, 150, 48, and 122 times in *D1*, *D2*, *D3*, and *D4*, respectively. The bank name (omitted due to IRB) occurred 222, 106, 27, and 53 times in *D1*, *D2*, *D3*, and *D4*, respectively. As we show later in Section 4.1, such base words are not recognized as being common by state-of-the-art password meters, which results in inflated entropy results for passwords that contain those contextually common base words.

In addition to where they live, one can also observe that users in each demographic group in our datasets tend to use the location or country where they come from. For example, “london”, “liverpool”, “canada” are among the top base words for *D4*. Likewise, both “pakistan” and “india” topped the *D2* dataset.

One common trend observed in password choices for *D3* users is the use of months. Out the top ten base words for *D3*, seven are month names. Note that this observation does not hold for the other groups.

Password Length. Table 5 compares the password length across all datasets. For *D1*, *D2*, *D3*, and *D4*, most users create passwords of length 8, followed by a decent number from 9 to 12. However, more *D1* and *D4* users (nearly 40% and 42%, respectively) tend to favor passwords of length 8, while *D2* and *D3* users (33% and 31%, respectively) favor it less so. Although percentages of passwords of length 9, and 10 seemed similar across all groups, one can observe that *D2* and *D3* groups picked slightly more passwords of length 11.

While we include password length information for Rock You and Xato, it is important to note that this leak's password lengths should not be directly compared with them. Recall that the passwords in this leak were retrieved through brute-force hash cracking, meaning that the cracked passwords will be biased towards shorter (and hence easier to crack) passwords.

3.2. Password Character Composition

We also analyzed the composition of passwords in the dataset with respect to which characters they use. The results can be found in Table 6. The majority of the bank users used the S_D password structure which means the password started with a string (S), followed by one (or more) digit(s)(D), whereas string-only passwords dominated the selection of Rock You and Xato users. This also suggests that while the bank did enforce the usage of both letters and digits, it did not enforce special characters (S_P). Despite this lack of enforcement, it appears from our results that 9% of *D2* passwords contained special characters, whereas less than 5% of other groups used special characters.

We further investigated the password composition in terms of characters. Fig. 1 shows the percentage of passwords containing different character cate-

Table 6: Password Composition

D1	*D4*	*D2*	*D3*	RockYou	Xato
S_D: 77% (27725)	S_D: 75% (6226)	S_D: 72% (10907)	S_D: 81% (5798)	S: 44% (14446520)	S: 41.86%(4185427)
D_S: 8% (3081)	Other: 5% (484)	S_SP_D: 9% (1425)	D_S: 6% (448)	S_D: 30% (9833381)	S_D: 21.74% (2173420)
S_D_S: 3% (1391)	D_S: 5% (465)	D_S: 5% (790)	S_SP_D: 4% (291)	D: 15% (5194466)	D: 20.36% (2035149)
Other: 3% (1185)	S_D_S: 5% (454)	Other: 3% (577)	Other: 3% (222)	D_S: 2% (896003)	D_S: 5.5% (549643)
S_SP_D: 3% (1127)	S_D_SP: 2% (225)	S_D_SP: 3% (477)	S_D_S: 2% (163)	Other: 1% (631963)	Other: 5.24% (524366)
S_D_SP: 1% (383)	S_SP_D: 2% (191)	S_D_S: 2% (425)	S_D_SP: 1% (72)	S_D_S: 1% (597558)	S_D_S: 3.64% (363760)

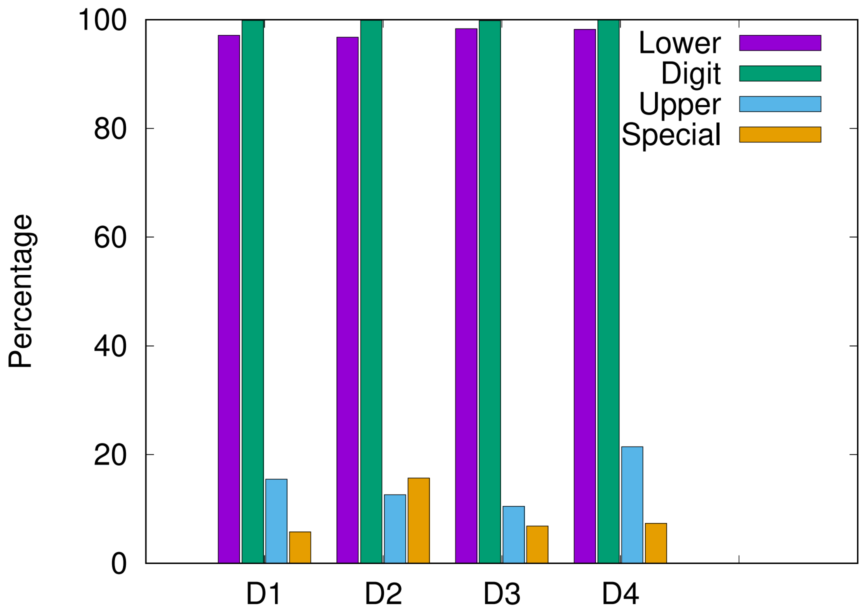


Figure 1: Percentage of passwords containing different class of characters: lower case, digits, upper case and special characters, respectively.

gories: lower case, digits, upper case, and special characters, respectively. First, we observe that lower case letters and digits are both heavily used. More than 96% of passwords in all groups contain at least some lower case letters, and more than 99% of passwords contain at least some digits. For those two metrics, there is also no significant distinction between the demographic groups. There is, however, a distinction when it comes to special characters. Group *D2* is almost twice as likely to use a special character as any of the other groups. Using the Fisher Exact test, we are able to statistically confirm that *D2* users utilize special characters more than other groups (p-value = 0.03 between *D2* and *D1*, and even smaller between *D2* and the other two groups).

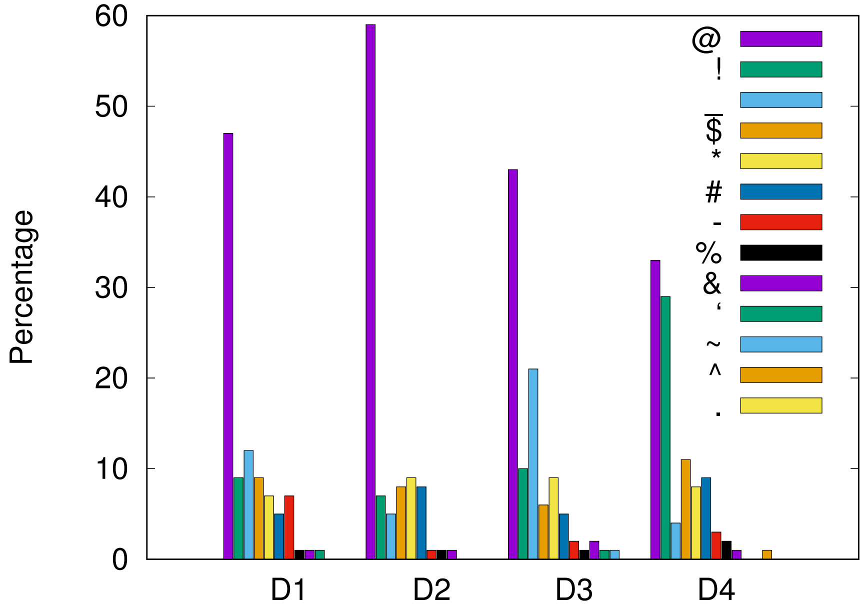


Figure 2: Normalized frequency of special characters

We subsequently broke down the analysis of the special characters category. Fig. 2 shows the normalized frequency of the special characters used in the passwords. We used the total number of special characters as the normalization factor. We observe that character “@” is the most common in all the groups. Beyond that, however, *D4* show a preference for “!” not seen in other groups, while *D3* prefers “_”. *D1* and *D2* don't show a significant preference for their next most popular character.

3.3. The Use of Names

To get an insight on how often users use names, we seek to identify passwords that contain names. To this end, we downloaded *nameDB*, a name database containing nearly 129,000 first and last names. The database is based on the US census data, so most names are English-based (though derived from various origins). That said, we observed that it also contains a substantial number of Middle Eastern names, some of which come in various spellings. For example, variants of “Mohamed”, such as “Mohammad”, “Mohamed”, and “Mohammed” all appear in the dataset. Similarly, the database also contains English-based names that have different spellings. We augmented the database with names extracted from our dataset.

While official Arabic datasets exist, we were not able to use them since those datasets are stored in Arabic, and direct conversion of Arabic text to Roman characters does not adhere to well formed rules. For example, the name "Abdulrahim" in Arabic has more than 1000 potential spellings in Roman characters, despite the fact that only a few variations are commonly used. While our names dataset contained a few variants for some names, we believe that the most common variants suffice since less common variants are less likely to be used.

We stored these names in a trie. For each password in the datasets, we removed numbers and special characters and searched for whole string matches (ignoring case since it is irrelevant) whose lengths are greater than or equal to δ, in the trie. In our analysis, we choose to set δ to 4 and 5. This means that we only consider a password substring as a name if the substring match is greater than δ (4 and 5). While our approach dismisses the detection of names of length 3 (e.g. Ali or Ian), we favor reducing false positives (strings of length 3 that look like names but are not), over increasing true positives (names of length 3). Fig. 3 shows the percentage of passwords that contain names when δ is 4 and 5.

When δ is 4, four out of the five datasets show that more than 30% of passwords contain names. The percentage is above 25% on average for all datasets (except the Xato dataset). Interestingly, we observe that the *D2* dataset passwords seems to exceed other datasets (close to 45% when $\delta = 4$) in terms of the percentage of names used. However, one can not conclude that the corresponding demographic is more likely to use names, as the name detection percentage may be an artifact of the databases we used to construct the trie, which may bias the results among the different demographics. This experimental limitation does not impact the insight that a substantial percentage of passwords are based on names. In fact, the percentages shown in the figure represent a lower bound on how often names exist in passwords.

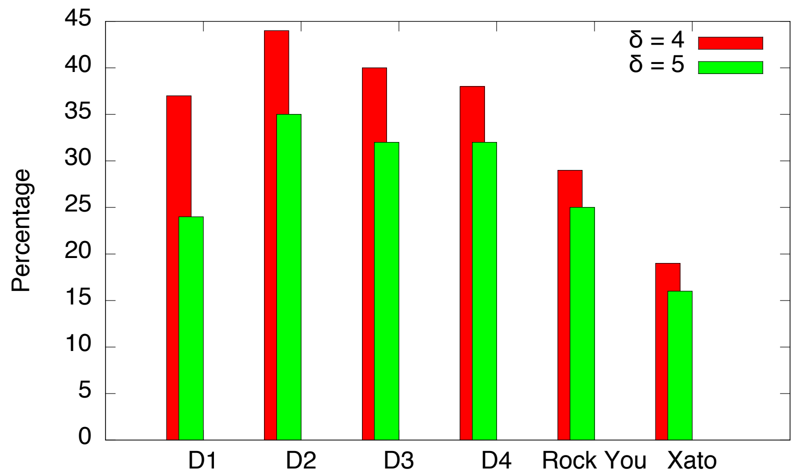


Figure 3: Percentage of the passwords that contain a name of length greater than or equal to δ

Comparison to previous work. Contrary to a previous study [20], which showed that only 14% of passwords contained names, our observations indicate that a substantial percentage of passwords contain names. Another study analyzing a leaked dataset of Chinese users [16] found that 22% of users use their *own* names in their passwords. Recall that our results search for *any* name matched.

3.4. Keyboard Walks

It has been previously observed that users tend to use keyboard walks to construct their passwords. Several common keyboard walks including "abcd", "qwerty", and "asdf" made it to top ten passwords and base words of Section 3.1. In this section, we seek to understand how often keyboard walks are used among different demographics.

We used an open source keyboard walk checker [21] to identify passwords that are based on keyboard patterns. The tool is fed with a QWERTY keyboard graph containing the keyboard adjacencies. For each password, the tool checks if every two consecutive letters are adjacent according to the input graph. A password is considered a keyboard walk if at least l consecutive letters are

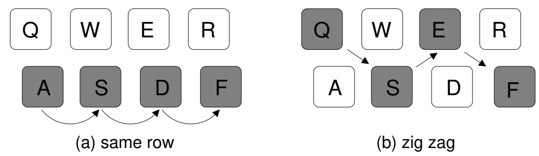


Figure 4: Commonly used keyboard walks to type passwords: (a) same row, (b) zig zag and (c) snake.

adjacent.

We also augmented the tool to classify walks into *same row*, *zig zag*, and *snake* walks. Those categories were defined by Li *et al.* [4]. A walk is considered a *same row* category if l adjacent characters are on the same row (e.g. asdf). A walk is considered *zig zag* if all l consecutive characters are adjacent but no two consecutive characters are on the same row (e.g. qsef). Finally, a snake pattern is a walk that is neither a *same row*, nor a *zig zag* (e.g. qwdf). Fig. 4 illustrates the three different keyboard walks considered in this work.

Fig. 5 shows the total percentage of walks, with the breakdown of the walk categories, for each demographic. Passwords in the *D1* and *D2* datasets exceed other datasets in the percentage of walks used at 12.5% and 7.8%, respectively. Across all datasets, the *same row* keyboard pattern dominated the choice of users, whereas the *zig zag* pattern showed minimal usage with at most 1%.

Comparison to previous work. Li *et al.* [4] compare the use of keyboard walks in passwords for Chinese and English users, and find that 8% of Chinese passwords are composed according to keyboard patterns, compared to less than 3% for English users.

3.5. Quantifying Personal Data Based Leakage

We now seek to understand how users across all datasets use their own personal information to compose their passwords. In particular, we analyze how often users use their own names and birth dates in their passwords. Such personal information can be easily obtained by an attacker in the case of a targeted attack or an insider attacker (e.g. bank employee).

To quantify how susceptible passwords are against targeted attacks, we devise a new metric, PL_i, which is the percentage of password leakage of password

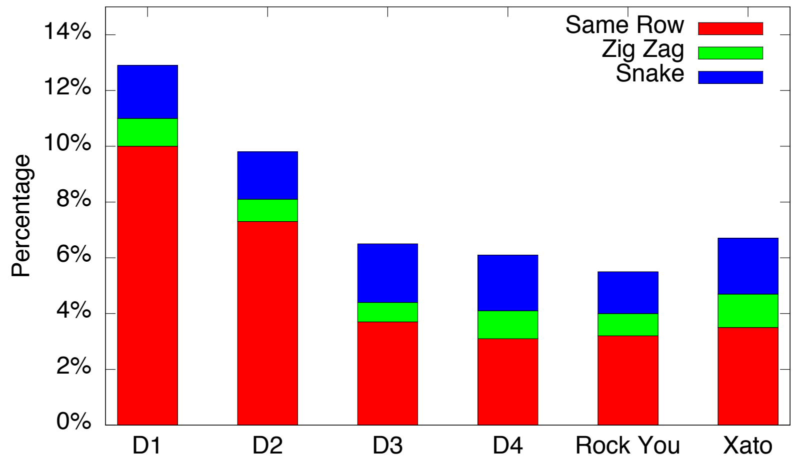


Figure 5: Percentage of passwords that are based on keyboard walks.

pw_i given the target user's name and birth year. We compute PL_i as follows,

$$PL_i = \frac{length(LCS(pw_i, name)) + match(pw_i, year)}{length(pw_i)} \tag{1}$$

$$match(password, year) = \begin{cases} 4 & \text{if 4-digit year exists} \\ 2 & \text{if 2-digit year exists} \\ 0 & \text{no year exists} \end{cases}$$

where $LCS(password, name)$ is the longest common substring between a user's password and name that is greater than a certain threshold, which we set to 3.[2] The function $match(password, year)$ returns 4, or 2 if a user's four-digit, or the least significant two-digit, birth year, respectively exists in the password. If neither exists, it returns 0. For example, if a birth date of a user is 18-January-1980, then $match(password, year)$ returns 4 if "1980" is a substring in the password, 2 if only "80" exists in the password, or 0 if neither exists in the password.

Fig. 6 depicts the distribution of PL_i for all datasets. Among the four

[2]We ignore letter case in substring matching.

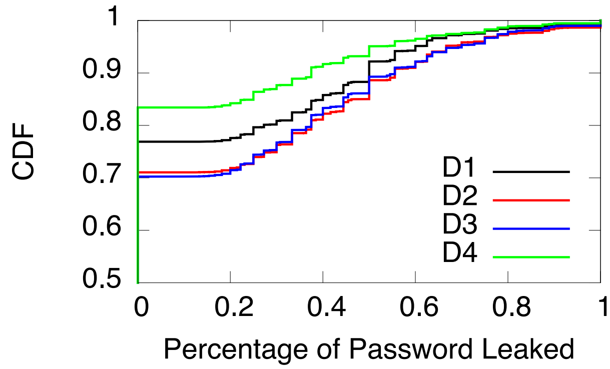


Figure 6: The distribution of the percentage of password leakage of password pw_i (PL_i) for all datasets.

demographics, *D2* and *D3* users are more likely to leak password characters using their name and birth year. Around the last quartile of *D2* and *D3* users leaked more than 40% of their passwords by composing passwords that contain their names and birth years. Although users from *D4* seemed the least likely (among our dataset) to rely on their names or birth years, a substantial fraction of users (10%) also leaked more than 40% of their passwords. Users from *D1* fit in between with 20% of the users leaking more than 40% of their passwords.

Fig. 7 provides an alternative depiction of the data. It depicts the percentage of users who (1) used their name with LCS greater than or equal to the threshold of 3, (2) used 4-digit birth year, (3) used 2-digit birth year, (4) used both LCS ≥ 3 and 2-digit birth year, and (5) used both LCS ≥ 3 and 4-digit birth year. More than 25% of *D1*, *D2*, and *D3* groups had an LCS ≥ 3, whereas only 15% of *D4* had an LCS ≥ 3. Also, more than 5% of users from all groups used 2-digit birth year. Overall, English users seem to rely less on their names or dates to compose their passwords.

Comparison to previous work. Wang *et al.* [9] also observed that English users tend to use less Personally Identifiable Information (PII) in composing their passwords, though their analysis covers a set of security-savvy (data leak from a hacker forum) English users. They don't discuss the percentage of password leaked due to personal information.

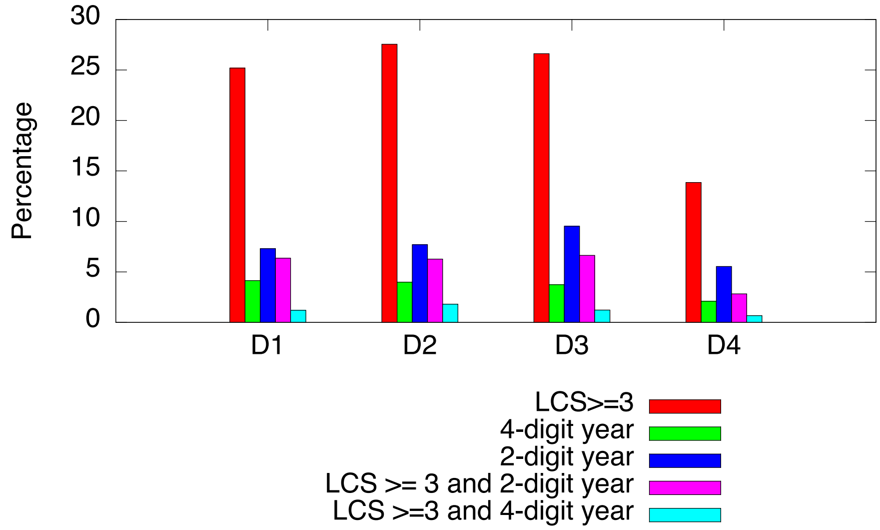


Figure 7: The percentage of users who (1) used their name with LCS (Longest Common Substring) greater than or equal to the threshold of 3, (2) used 4-digit birth year, (3) used 2-digit birth year, (4) used both LCS ≥ 3 and 2-digit birth year, and (5) used both LCS ≥ 3 and 4-digit birth year.

3.6. Phone Number

Of the 79,760 accounts for which we have passwords, 42,650 of them have at least one registered phone number. These phone numbers could be a mobile phone, home phone, work phone, or fax. Many accounts contain more than one phone number. We analyzed these 42,650 accounts in order to determine how many of those users include their phone number as a part of their password. A user is considered to have used their phone number in their password if at least one of their phone numbers can be found, in its entirety[3], within their password.

Overall, 3.9% of the users for whom we have a phone number include their phone number as part of their password. This was not evenly distributed across our demographic groupings, however. Fig. 8 shows the results broken down by demographic. The *D1* and *D2* groups were significantly more likely (χ^2 test,

[3]This is not strictly true. The leaked data contains phone numbers that are eight digits, however prior to 2010 all phone numbers in Qatar were only seven digits. Because of this, we also accepted matches based on the older seven digit scheme.

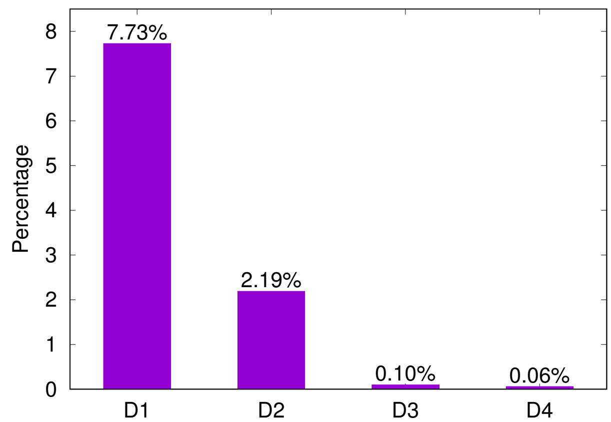


Figure 8: Percentage of users whose password contains at least one of their registered phone numbers

p-value < 0.001) than the other two groups to use their phone number as part of their password. This is higher than has been found in previous work analyzing passwords from Chinese users. Li *et al.* [4] find that 2.7% of passwords contain users' registered phone numbers. Wang *et al.* [9] observe less than 0.5% of password contain a registered phone number.

For the users who used their phone number in their password, we also studied what percentage of their password is comprised of it. A CDF of these results for *D1* and *D2* can be found in Fig. 9. As can be seen in the figure, over 50% of users in *D2* and 70% of users in *D1* add 2 or fewer characters to their phone number when converting it into a password. This demonstrates that, in general, if a user includes their phone number in their password, they do not include many other characters in order to increase complexity.

3.7. Security Questions

While not strictly a part of how users create their passwords, the dataset also contains security questions for 97,086 users. There is one security question per user, and it was chosen by the user at the time of account creation. This provides a unique opportunity to see what types of security questions users choose.

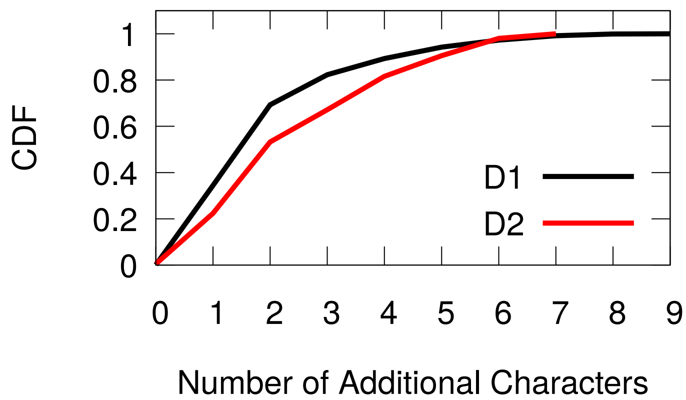


Figure 9: The distribution of the percentage of password leakage by users who use their phone number in their password

Table 7: Top 10 Security Questions

Question	**Frequency**
What is my name?	7.55% (7326)
What is my mother's name?	4.37% (4242)
What is my birthday?	3.94% (3827)
What is my son's name?	2.69% (2616)
What is my nickname?	2.43% (2363)
What is my mother's maiden name?	2.19% (2124)
What is my father's name?	1.92% (1868)
What is my wife's name?	1.79% (1739)
What kind of car do I drive?	1.71% (1658)
What is my daughter's name?	1.64% (1588)

We performed a manual analysis of all the security questions by sorting and grouping them according to the type of question. A manual, rather than automated, analysis was used due to the variety of ways users ask the same question. For example: "mama name", "mom's name", and "mother name" all refer to questions asking about the name of the user's mother. Misspellings (such as "mather naem") were also present.

While the majority of security questions were written in English (which is somewhat surprising given the demographic of users in the data-set), questions in other languages were translated to English whenever possible using either a speaker of that language or Google Translate.

The top 10 security questions, as well as how many users use each question,

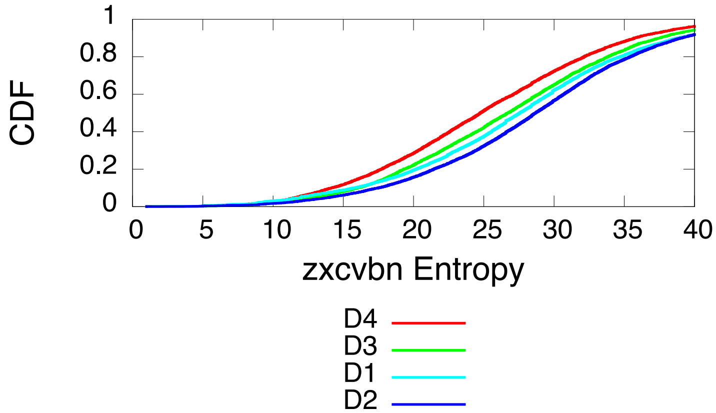
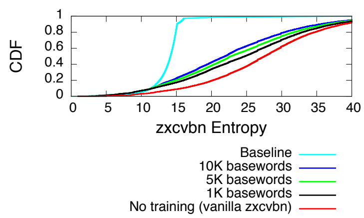
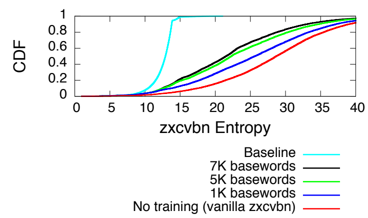


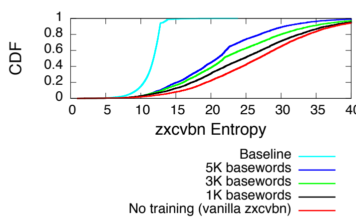
Figure 10: Empirical entropy estimation using *zxcvbn* [10].



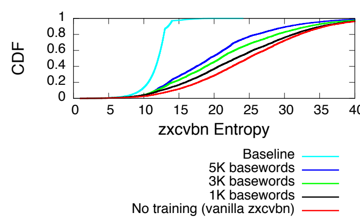
(a) D1



(b) D2



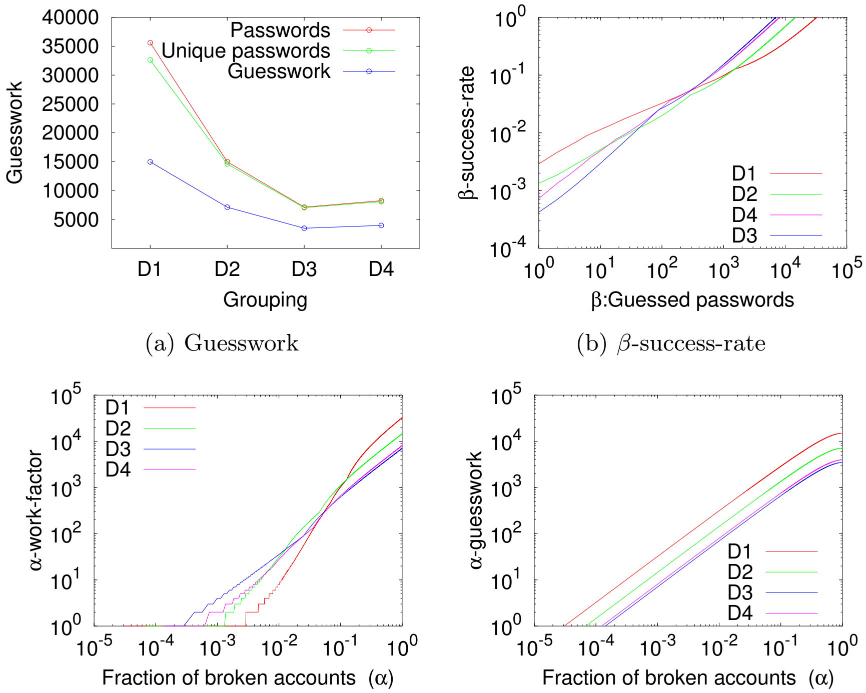
(c) D3



(d) D4

Figure 11: zxcvbn Entropy distribution comparison between the four data sets for different training sizes.

can be found in Table 7. As can be seen, names are the most common type of security question, with the user's own name being the most common. As one would suspect, none of the top security questions are about information that is actually a secret. It is very likely that a targeted attack could find the information on social media. These results further support the idea that security questions, even when chosen by the user, are not secure in the face of a targeted



(a) Guesswork

(b) β-success-rate

(c) α-work-factor

(d) α-guesswork

Figure 12: Analysis of the guessability of the datasets using theoretical metrics

attack.

While we did not study the security question answers in this work (a more thorough analysis of security questions and their answers is left for future work), we did compare the hashed passwords and hashed security question answers and determined that for 17 users they are identical. For only two of those users, however, was the security question some variation of "What is my my password?".

4. Adversarial models and guessing metrics

In this section, we consider two adversarial models: $\mathcal{A}_1$ and $\mathcal{A}_2$, and we evaluate their effectiveness against the four groups using various guessing metrics.

- $\mathcal{A}_1$. This is the most studied model in the literature where the attacker aims to perform password cracking using dictionary attacks, or brute-force cracking. The attacker resorts to external datasets (training sets) to recover a set of the unknown cryptographically hashed passwords. To evaluate the guessability of our datasets against this attacker model, we use *zxcvbn*[10], a state-of-the-art password strength estimator to calculate the entropy in bits.

- $\mathcal{A}_2$. We assume the adversary has the whole password dataset, and given a set of unknown passwords randomly drawn from the known dataset, we want to evaluate the efficiency of the adversary on trying to uniquely identify the set of the unknown passwords. This adversary model consider only password's frequencies and not how passwords are composed.

$\mathcal{A}_2$ model refers to an adversary that, by systematically trying the passwords from the most to the less frequent, measures the guessability of the remaining passwords. $\mathcal{A}_2$ evaluates only password frequencies without analyzing how passwords are made.

4.1. $\mathcal{A}_1$ adversary model

One shortcoming of relying on non-parametric statistics, presented in Section 4.2, to evaluate the guessability of passwords, is that the analysis can vary greatly with the sample size [3]. In this section, we seek to validate our theoretical results and augment them with empirical analysis. To that end, we use *zxcvbn*, an open source password strength estimator, which has been experiencing a growing industry adoption (e.g. Dropbox, Wordpress, Kaspersky Labs, etc) [10, 22]. In a recent large scale evaluation study of password strength meters [22], the authors recommend companies to adopt *zxcvbn* due to its effectiveness.[4]

[4]Due to our IRB restrictions we are unable to use Carnegie Mellon University's Password Guessing Service (PGS) [23], which is considered the gold standard in password strength estimation. The reason for this limitation is that PGS requires us to submit the passwords to be evaluated to their service, and we are not permitted to do so. However, *zxcvbn* has shown accurate performance when compared with PGS [10].

zxcvbn estimates a password's strength in three phases. First, a matching step is performed to find a set of S overlapping substrings in the password. Identifying such matches depends on dictionaries (of common names, passwords, keyboard walks, patterns, etc) that *zxcvbn* builds based on frequency. Next, a scoring step assigns a guess attempt estimation to each match independently.Finally, the last step is to search for the sequence S' of non-overlapping adjacent matches drawn from S such that S' fully covers the password and minimizes a total guess attempt score.

Fig. 10 compares the entropy in bits of the four groups using *zxcvbn.* Interestingly, while *D4* users showed less tendency to rely on personal information in composing their passwords, they showed slightly lower entropy distributions–24 bits compared to 28 bits at the median for the *D2* group.[5]

Ideally, from the perspective of a targeted attack, *D4* users are expected to show higher entropy (more secure) since their passwords contain less personal information. However, this is not the case because *zxcvbn* is based mainly on English dictionaries and base words. This again highlights a key shortcoming of entropy-based tools–they fail to consider contextualized data, which are readily available in the case of a targeted attack. Using such tools in different parts of the Middle East, or Asia, for example, would not provide realistic entropy results, and may provide users with a false sense of security.

To improve the results of *zxcvbn*, we extracted the top base words from each group using *pipal.* Fig. 11 shows, for each of the four groups, the entropy distribution results when *zxcvbn* is fed with increasing sizes of base word sets. To help put the results in perspective, we obtained baseline entropy distribution for each group by feeding *zxcvbn* with the respective complete passwords dataset.

As expected, providing *zxcvbn* with more contextualized base words significantly improves its performance as it starts to provide more realistic entropy results. For *D1*, when *zxcvbn* is provided with 10K base words (less than 30%

[5]Kolmogorov-Smirnov (K-S) test can not confirm that the samples of *D4* and other groups come from the same distribution.

of *D1*), the median entropy drops from 27 to 21 bits. The reason is that *zxcvbn* is able to perform better sequence matching. For example, before training with contextualized base words, a password containing the two substrings "qatar" and "doha" concatenated with a two-digit number gets an entropy score of 40 bits since the tool does not recognize those two strings. However, after training, the same password gets an entropy score of 14 bits, which is a more realistic score given the context.[6]

4.2. $\mathcal{A}_2$ adversary model

In the following, we refer to the password dataset as

$$W : \{W_{D1}, W_{D2}, W_{D3}, W_{D4}\}$$

where W_g refers to passwords belonging to the $g^{th} \in \{\mathrm{D1}, \mathrm{D2}, \mathrm{D3}, \mathrm{D4}\}$ group. Moreover, we refer to the frequency of the i^{th} password w_i as p_i, where $0 < p_i \leq 1$, such that $\sum p_i = 1$. We also assume the frequencies of the passwords $\{w_1, \ldots, w_N\}$ form a monotonically decreasing sequence, i.e., $p_1 \geq p_2 \geq \ldots \geq p_N$, where N is the dataset cardinality. We stress that the results from the following metrics should be carefully considered and weighted since the grouping break-down inevitably affects the sample dimension.

Guessing model. We assume the adversary knows the whole password dataset, i.e., $W : \{w_1, \ldots, w_N\}$, and given a set of unknown passwords randomly drown from W, i.e., $X : \{x_1, \ldots, x_M\}$, with $M \leq N$, we want to evaluate the efficiency of the adversary on trying to uniquely identify the set X.

One of the first metrics to estimate the guessing robustness of a password is the *guesswork* or *guessing entropy* $G(W)$ defined as it follows:

$$G(W) = \sum_{i=1}^{N} p_i \cdot i$$

The guesswork estimates the expected number of guesses required to find an unknown password $X = \{x_1\}$ proceeding in optimal order, i.e., exhaustively

[6]Note that the baseline entropy is 14 bits at the median for *D1*. This means that at the median, the size of the attacker search space is 2^{14}, which roughly half the size of *D1*.

trying all the passwords from the most likely w_1 to the most unlikely w_N.

Fig. 12a shows for each of the group g, with $g \in \{\mathrm{D1}, \mathrm{D2}, \mathrm{D3}, \mathrm{D4}\}$, the number of passwords, the number of unique passwords and the guesswork. We observe that $G(D1) >> G(D2) > G(D3) \approx G(D4)$; indeed, guesswork measures the expected number of guesses before succeeding, taking into account all the instances of the unique passwords. Group *D1* is the most robust to guessing attacks given the high number of unique passwords (difference between the red and the green line in Fig. 12a compared to the group size (Table 2). Since $G(W)$ involves an exhaustive search over all the passwords, i.e., even the most unlikely one such as w_N, it cannot be considered an efficient way to guess an unknown password. Generally speaking, the above method requests an average of $M \cdot G(W)$ guesses when the number of passwords unknown to the adversary are M, i.e., $X : \{x_1, \ldots, x_M\}$. Nevertheless, a smarter adversary might do much better with the optimal strategy of first guessing the most likely password w_1 for all unknown passwords and then move to the second one w_2, and so on. By considering this model there are several guessing measures.

The first metric we evaluate is the *β-success-rate*, yielding:

$$\lambda_\beta(W_g) = \sum_{i=1}^{\beta} p_i$$

$\lambda_\beta(W_g) \in [0, 1]$ measures the expected success for an attacker limited to $\beta \in [1, \ldots, N]$ password guesses.

Fig. 12b shows the β-success-rate as function of β, i.e., the number of guessed passwords, for each of the considered group. We observe that for low values of guessed passwords, i.e., $\beta << 100$, the four groups behave significantly different, i.e., $\lambda_\beta(W_{D1}) > \lambda_\beta(W_{D2}) \approx \lambda_\beta(W_{D4}) > \lambda_\beta(W_{D3})$. In particular, group *D1* is less robust to guessing compared to group *D3*, while group *D2* and group *D4* behave approximately in the same way. Indeed, group *D1* presents more replicated passwords than the others—recall Fig. 12a and in particular the difference between number of passwords and unique passwords. For large values of guessed passwords, i.e., $\beta >> 100$, the different cardinalities of the datasets affect the results and group *D1* turns out to be the most robust. Indeed, for large values of guessed passwords, the adversary does not gain significant advantage when the probabilities of unlikely passwords are very small.

We also consider the *α-work-factor*, yielding:

$$\mu_\alpha(X) = \min\left\{ j \;\middle|\; \sum_{i=1}^{j} p_i \geq \alpha \right\}$$

$\mu_\alpha(X)$ represents the minimum number of guesses an adversary has to perform to break a desired fraction of the dataset. Fig. 12c shows the α-work-factor for the 4 groups as a function of the fraction of the guessed passwords $\alpha \in [0, 1]$. $\mu_\alpha(X)$ measures also the minimum number of guesses to experience a probability α to break an individual password adopting an optimal dictionary as already introduced before. Fig. 12c confirms the results already highlighted by the β-success-rate: for small fractions of the compromised dataset ($\alpha < 5 \cdot 10^{-2}$), the group 3 is the most robust to password guessing while group *D1* is the weakest one. As also highlighted before, we observe a break-even point at $\beta \approx 200$ (Fig. 12b) and $\alpha \approx 5 \cdot 10^{-2}$ (Fig. 12c) for which the guessing robustness changes when unlikely probabilities are taken into account.

We observe that the above statistics do not capture the fact that a real attacker can stop early after successful guesses. Indeed, while μ_α measures the fixed number of required guesses, in the following we evaluate the *α-guesswork*, which in turn, it estimates the average number of guesses per account to achieve a success rate α, yielding:

$$G_\alpha(W) = (1 - \lambda_{\mu_\alpha}) \cdot \mu_\alpha + \sum_{i=1}^{\mu_\alpha} p_i \cdot i$$

$G_\alpha(\mathcal{W})$ represents a compromise for an adversary spending $\sum_{i=1}^{\mu_\alpha} p_i \cdot i$ guesses for the passwords in its dictionary, while only μ_α guesses for the passwords outside of its dictionary. It is also interesting to observe that $G_{\alpha=1}(W) = G(W)$: the α-guess-work becomes the guesswork when $\alpha = 1$, i.e., when the adversary guess all the passwords exhaustively in the dataset. Fig. 12d shows the α-guess-work for the four groups as function of the fraction of guessed passwords α. Recalling that the above statistics consider the passwords ordered according to their frequencies, group *D1* is overall the most robust to guessing attacks given the large amount of unique passwords.

5. Discussion

We will now provide some additional discussion regarding this study and its results.

5.1. Lessons Learned

The analysis of the dataset confirms common insecure practices adopted by people while choosing passwords even in the case when passwords are used to protect more sensitive assets such as an online bank account. As already observed in many other data leaks, people tend to make passwords from common base words. Our results confirm that these are biased by culture and location, e.g., *qatar*, *ahmed*, and *doha* are the most common basewords in our dataset. We confirmed that users tend to include in their password the following information: names, keyboard walks, birth-dates and phone numbers. Our analysis shows that more than 25% of the passwords contain names and more than 4% contain either keyboard walks or phone numbers. Our analysis also confirms that even the security question is strictly related to information that can be easily recovered by a straightforward social networking website analysis; indeed, more than 10% of the questions are related to a user's name, mother's name, or birthday. Finally, we observe that password composition rules are consistent among our grouping but slightly different from other datasets. While passwords from the Rock You and Xato datasets are mainly constituted by strings, our passwords are (for the vast majority) constituted by strings followed by digits.

Our dataset is particularly interesting because it comes with rich meta-data that can be exploited to study how people from different cultures are choosing passwords. Indeed, we have very interesting and unprecedented peculiarities: Indians and Pakistanis are more prone to include names in their passwords while people coming from Philippines tend to include months' names in their password. It is also interesting to observe that while the "@" character is the most used among all the groups, English speakers next favor "!" while Filipinos prefer "_". The usage of keyboard patterns such as the *snake* and the *zig zag* is consistent throughout the groups and with the other password leaks that we have analyzed (Rock You and Xato); nevertheless, we observe a larger number of patterns using the *Same Row* for the Arabic group (10%). Finally, we observe that Arabic people include their mobile phone number in the passwords more frequently than the other groups, i.e., three times more than the second group in our grouping.

While confirming insecure practices in choosing passwords, our analysis highlights several correlations between how passwords are generated and the user's country of origin. These information might be used for several purposes, e.g., they might be implemented in online password strength estimators to force users to choose more unpredictable passwords.

5.2. Guessability analysis

In order to evaluate the guessability of the passwords in our dataset, we used both theoretical and empirical metrics by considering two different adversarial models, i.e., $\mathcal{A}_1$ and $\mathcal{A}_2$ (see Sections 4.1 and 4.2). While the former involves well-known metrics from previous works [3][6], the latter involves metrics extracted from the *zxcvbn* tool[10][22]. For both the adversarial models, we observe that the English group (*D4*) performs worse than the other ones, i.e., the guessability is higher. Indeed, by considering the theoretical metrics, such as the α-guesswork (Fig. 12d), we recall that a lower α-guesswork is required to guess a fixed fraction of the accounts.

Moreover, recalling the entropy estimation provided by *zxcvbn* in Figure. 10, we observe that the English group (red line) is characterized by less entropy than the other groups despite the observation that English users tend to use less personal information in their passwords than other groups. This was surprising because one would expect other groups to have lower entropy. This surprising result can be explained by recalling the fact that *zxcvbn* resorts to an internal English-based dictionary to estimate the password guessability, and therefore, commonly used words such as "qatar" or "doha" are not recognized and the corresponding passwords end up with an inflated entropy. This leads us to the general idea that password meters could be biased (due to the lack of culturally relevant training) and should take into account the local context such as social factors, languages and the culture.

5.3. Limitations

Our dataset is constituted by less than 66k passwords which is significantly less than the other datasets considered in our analysis (Rock You and Xato). Moreover, while we acknowledge that the grouping process reduced even more the sets' cardinality, it gave us the unprecedented opportunity to perform analysis and comparisons that have not been carried out before. Our work is the first to characterize passwords based on the 3 groups we identified. Despite the small cardinality, our analysis identified many observations and insights that were not known before. Another novelty is that our dataset is unique in the sense that it is rich in personal information (significantly more than other data sets studied before) which allows us to understand how users in these groups use their personal information to select their passwords for sensitive application such a banking. We also observe a few other limitations in our analysis: (i) About 18% of password hashes from original leak remain uncracked, and these are likely the most complex passwords; (ii) Some Arabic names can be spelled in several ways and this eventually may have affected our results; (iii) Password composition analysis (Table 4) takes into account only names of people, while in some cases we observed other attributes such as pets' names.

6. Related work

Evaluating password strength has been an active area of research with early contributions in the 70s [24] and later on [25, 26, 27]. The literature mostly uses the entropy as a metric to quantify the strength of passwords. More recently, Joseph Bonneau [3] observes that accepted metrics such as the guesswork and entropy fail to model the tendency of real world attackers to cease guessing against the most difficult accounts. He introduces the concepts of partial guessing metrics such as β-success-rate, α-work-factor and α-guesswork. We use these metrics in our theoretical analysis.

Much of our understanding of the security of passwords comes from analyzing leaked or collected password datasets. Passwords from an entire university have been analyzed in [28]. Interesting insights about how Chinese users compose their passwords have also been noted [2, 4]. Other studies have also been conducted to understand password habits of users [1]. Das *et al.* [5] study multiple leaked datasets of passwords to understand password reuse. Recently, Ur *et al.* [7] investigate the relationship between users' perceptions of the strength passwords and their actual strength.

Metadata-rich leaked datasets enable more interesting analysis. Castelluccia *et al.* [29] highlight how additional personal information about a user helps in speeding up password guessing, and propose a new password cracker based on Markov models. More recently, Wang *et al.* [9] study the use of personal information in datasets of Chinese users' passwords, and in a dataset of security-savvy English users. They also propose guessing algorithms that exploit users' personal information. Li *et al.* [16] also study the use of personal information in Chinese datasets, and also proposes a Probabilistic Context-Free Grammars (PCFG) method that considers symbols linked to personal information in password structures. While those studies examine the use of personal information mostly in Chinese passwords, our study is based on passwords produced by more diverse demographics. Bonneau and Xu [30] perform a high-level user behaviour study, mainly related to character choices, through datasets of passwords created by English, Chinese, Hebrew and Spanish speakers. Unlike this work, the authors do not investigate the use of personal information in the examined datasets, which were (with few exceptions) considered small and limited.

To improve guessability attacks, Veras *et al.* [31] propose a natural language processing algorithm to segment, classify, and generalize semantic categories from passwords. Ur *et al.* [23] measure the effectiveness of several guessing attacks. In particular, they found that updating the guessing strategy dynamically during the cracking significantly outperforms the fully automated strategy. Password composition policies and their robustness against dictionary attacks have been studied in [32]. Our work is complimentary since it provides insights that can potentially improve guessability attacks. At the same time, our work also highlights the importance of training password strength estimators such as *zxcvbn* with contextual relevant information.

Authors in [33] demonstrate that as long as passwords remain human-memorable, they are vulnerable to smart-dictionary attacks even when the space of potential passwords is large. Indeed, authors adopted standard Markov modeling techniques from natural language processing and proved that they can be used to dramatically reduce the size of the password space to be searched. Moreover, they proposed an algorithm for efficient enumeration of the remaining password space. Our contribution confirms the weaknesses of human-memorable passwords, and shed light on some recurrent patterns that depend on the origin country.

7. Conclusions

This work presents an analysis of a demographically-diverse password dataset which is rich in meta data. This allows us to get insights on how users from different groups incorporate their personal information (names, birth days, and phone numbers) into their passwords and show the extent to which they do so. Our results show that users from different demographical groups tend to show differences in how they form their passwords, and we quantify those differences for four groups: Arabs, Indian and Pakistani, Filipinos, and English speakers. In particular, we carried out a detailed comparative analysis among the groups and how each of the groups tend to incorporate personal information.

Our work also sheds light on the importance of training password meters with contextually relevant information before deployment. Existing password meters are based on common base words, and patterns observed from previous data leaks of mainly English speaking users. As a result, they can not provide accurate results of entropy estimation. We experimentally observe how a state-of-the-art password strength estimator (*zxcvbn*) actually overestimates the passwords generated by people with a cultural background different from English. We are able to improve its estimation by training it with relevant base words.

References

[1] D. Florencio, C. Herley, A large-scale study of web password habits, in: Proceedings of the 16th International Conference on World Wide Web, WWW '07, ACM, New York, NY, USA, 2007, pp. 657–666.

[2] W. Han, Z. Li, L. Yuan, W. Xu, Regional patterns and vulnerability analysis of chinese web passwords, IEEE Trans. Information Forensics and Security 11 (2) (2016) 258–272.

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

[4] Z. Li, W. Han, W. Xu, A large-scale empirical analysis of chinese web passwords, in: 23rd USENIX Security Symposium (USENIX Security 14), USENIX Association, San Diego, CA, 2014, pp. 559–574.

[5] A. Das, J. Bonneau, M. Caesar, N. Borisov, X. Wang, The tangled web of password reuse, in: 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014, 2014.

[6] J. Bonneau, C. Herley, P. C. van Oorschot, F. Stajano, Passwords and the evolution of imperfect authentication, Commun. ACM 58 (7) (2015) 78–87.

[7] B. Ur, J. Bees, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, Do users' perceptions of password security match reality?, in: Proceedings of

the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, ACM, New York, NY, USA, 2016, pp. 3748–3760.

[8] J. Blocki, A. Datta, J. Bonneau, Differentially private password frequency lists, IACR Cryptology ePrint Archive 2016 (2016) 153.

[9] D. Wang, Z. Zhang, P. Wang, J. Yan, X. Huang, Targeted online password guessing: An underestimated threat, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, ACM, New York, NY, USA, 2016, pp. 1242–1254.

[10] D. L. Wheeler, zxcvbn: Low-budget password strength estimation, in: 25th USENIX Security Symposium (USENIX Security 16), USENIX Association, Austin, TX, 2016, pp. 157–173.

[11] International Organization for Standardization, Country Codes - ISO 3166, http://www.iso.org/iso/home/standards/country_codes.htm (1993).

[12] J. Snoj, Population of Qatar by nationality - 2017 report, http://priyadsouza.com/population-of-qatar-by-nationality-in-2017/ (Online; accessed 17/05/2017).

[13] Openwall, John the Ripper password cracker, http://www.openwall.com/john/ (Online; accessed 15/05/2017).

[14] J. Steube, Hashcat Advanced Password Recovery, https://hashcat.net/hashcat/ (Online; accessed 15/05/2017).

[15] KoreLogic, Crack Me If You Can - DEFCON 2010, http://contest-2010.korelogic.com/rules-hashcat.html (Online; accessed 15/05/2017).

[16] Y. Li, H. Wang, K. Sun, A study of personal information in human-chosen passwords and its security implications, in: 35th Annual IEEE International Conference on Computer Communications, INFOCOM 2016, San Francisco, CA, USA, April 10-14, 2016, 2016, pp. 1–9.

[17] Daniel Miessler, SecLists, https://github.com/danielmiessler/SecLists/tree/master/Passwords (Online; accessed 15/05/2017).

[18] Mark Burnett, Today I Am Releasing Ten Million Passwords, https://xato.net/passwords/ten-million-passwords/ (Online; accessed 15/05/2017).

[19] R. Wood, Pipal, Password Analyzer, https://digi.ninja/projects/pipal.php (Online; accessed 15/05/2017).

[20] T. Hunt, The Science of Password Selection, https://www.troyhunt.com/science-of-password-selection (Online; accessed 15/05/2017).

[21] R. Kelley, Keyboard-Walk-Generators, https://github.com/Rich5/Keyboard-Walk-Generators (Online; accessed 15/05/2017).

[22] X. D. C. D. Carnavalet, M. Mannan, A large-scale evaluation of high-impact password strength meters, ACM Trans. Inf. Syst. Secur. 18 (1) (2015) 1:1–1:32.

[23] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher, R. Shay, Measuring real-world accuracies and biases in modeling password guessability, in: 24th USENIX Security Symposium (USENIX Security 15), USENIX Association, Washington, D.C., 2015, pp. 463–481.

[24] R. Morris, K. Thompson, Password security: A case history, Commun. ACM 22 (11) (1979) 594–597.

[25] J. Yan, A. Blackwell, R. Anderson, A. Grant, Password memorability and security: Empirical results, IEEE Security and Privacy 2 (5) (2004) 25–31.

[26] D. V. Klein, Foiling the cracker: A survey of, and improvements to, password security, in: Proceedings of the 2nd USENIX Security Workshop, 1990, pp. 5–14.

[27] E. H. Spafford, Observing reusable password choices, in: In Proceedings of the 3rd Security Symposium. Usenix, 1992, pp. 299–312.

[28] M. L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, P. G. Kelley, R. Shay, B. Ur, Measuring password guessability for an entire university, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS '13, ACM, New York, NY, USA, 2013, pp. 173–186.

[29] C. Castelluccia, C. Abdelberi, M. Dürmuth, D. Perito, When privacy meets security: Leveraging personal information for password cracking, CoRR abs/1304.6584.

[30] J. Bonneau, R. Xu, Character encoding issues for web passwords, in: Web 2.0 Security & Privacy, 2012.

[31] R. Veras, C. Collins, J. Thorpe, On semantic patterns of passwords and their security impact, in: 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014, 2014.

[32] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, J. Lopez, Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms, in: 2012 IEEE Symposium on Security and Privacy, 2012, pp. 523–537.

[33] A. Narayanan, V. Shmatikov, Fast dictionary attacks on passwords using time-space tradeoff, in: Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS '05, ACM, New York, NY, USA, 2005, pp. 364–372.