PII-PSM: A New Targeted Password Strength Meter Using Personally Identifiable Information

Qiying Dong[1], Ding Wang[1*], Yaosheng Shen[2], and Chunfu Jia[1]

[1] College of Cyber Science, Nankai University, Tianjin 300071, China
[2] School of ECE, Peking University Shenzhen Graduate School, Shenzhen 518055, China
wangding@nankai.edu.cn

Abstract. In recent years, unending breaches of users' personally identifiable information (PII) have become increasingly severe, making targeted password guessing using PII a practical threat. However, to our knowledge, most password strength meters (PSMs) only consider the traditional trawling password guessing threat, and no PSM has taken into account the more severe targeted guessing threat using PII (e.g., name, birthday, and phone number). To fill this gap, in this paper, we mainly focus on targeted password strength evaluation in the scenario where users' PII is available to the attacker. First, to capture more fine-grained password structures, we introduce the high-frequency substring as a new grammar tag into leading targeted password probabilistic models TarGuess-I and TarMarkov, and propose TarGuess-I-H and TarMarkov-H. Then, we weight and combine our two improved models to devise PII-PSM, *the first practical* targeted PSM resistant to common PII-accessible attackers. By using the weighted Spearman (WSpearman) metric recommended at CCS'18, we evaluate the accuracy of our PII-PSM and its counterparts (i.e., our TarGuess-I-H and TarMarkov-H, as well as two benchmarks of Optimal and Min-of-All). We conduct evaluation experiments on password datasets leaked from eight high-profile English and Chinese services. Results show that our PII-PSM is more accurate than TarGuess-I-H and TarMarkov-H, and is closer to Optimal and Min-of-All, with WSpearman differences of only 0.014∼0.023 and 0.012∼0.031, respectively. This establishes the accuracy of PII-PSM, facilitating to nudge users to select stronger passwords.

Keywords: Password authentication · Targeted guessing · Password strength meter · Personally identifiable information · Password probabilistic model.

1 Introduction

Identity authentication is the first line of defense to ensure information system security, and text passwords are the most widely used method [2]. The most common threat to password-based authentication is password guessing attacks, which can be divided into trawling attacks and targeted attacks based on the attacker's knowledge. By exploiting users' vulnerable behaviors (e.g., adopting popular passwords [1, 18] and keyboard patterns [23]), a trawling attacker performs indiscriminate password guessing on all user accounts to crack as many accounts as possible. In contrast, to guess a specific user's password, a targeted attacker takes advantage of the user's personal information to facilitate guessing. This is realistic because users tend to employ a variety of personal information (e.g., name, birthday, and old/sister passwords) when generating passwords [4, 9, 15, 22, 24].

In recent years, there have been numerous data breaches containing users' personal information. For example, the LinkedIn breach [14] leaks 700 million users' full names, phone numbers, physical addresses, email addresses, geolocation records, LinkedIn usernames and profile URLs, personal and professional experiences and backgrounds, genders, and other social media accounts and usernames; the Facebook breach [5] leaks 533 million users' full names, Facebook IDs, phone numbers, locations, birthdays, biographies, and email addresses; the Nitro PDF breach [7] leaks 77 million users' email addresses, full names, bcrypt hashed passwords, titles, company names, IP addresses, and other system-related information. This provides sufficient material for targeted guessing, making it a more severe and realistic threat than traditional trawling guessing [24].

To nudge users to select strong passwords, nearly every respectable web service provider has deployed password strength meters (PSMs). However, as far as we know, apart from PPSM [15], leading PSMs (e.g., [3, 6, 13, 21, 26]) only consider trawling guessing scenarios. These trawling PSMs do not include the user's personal information in password strength evaluation, and are thus unable to accurately measure password strength when facing real-world attacks. Besides, the targeted PPSM [15] relies on sister passwords from different sites in evaluating password strength, which is highly impractical due to two reasons: 1) The server generally does not hold the user's old (sister) passwords; 2) Sister passwords are not easily accessible [4, 24]. For example, Das et al. [4] analyzed 7.96 million accounts from different sites and found that only 152 (0.00191%) were successfully matched by email more than once; Wang et al. [24] analyzed 547.56 million accounts and found that less than 1.02% and 1.73% were successfully matched by email and username more than once. Comparatively, users often submit personally identifiable information (PII, such as name and phone number) when registering. Even if they do not submit, PII is easy to obtain (e.g., through social networks) by attackers. Thus, designing a PII-based PSM is urgent and necessary.

Contributions.

(1) **Two improved targeted password probabilistic models.** We analyze passwords from eight high-profile English and Chinese services, and find that high-frequency substrings (HFSs) can capture more fine-grained password structures than popular passwords. Thus, we introduce the HFS as a new grammar tag into leading targeted probabilistic models TarGuess-I [24] and TarMarkov [22], and propose the improved models TarGuess-I-H and TarMarkov-H.

(2) **A new targeted password strength meter (PSM).** We weight and combine our proposed TarGuess-I-H and TarMarkov-H to devise PII-PSM. It is *the first practical* targeted PSM resistant to common targeted attackers with personally identifiable information (PII), using the stochastic gradient descent approach to optimize the weights. In this way, the impact of randomly/manually setting the weights on PSM accuracy can be eliminated.

(3) **An extensive evaluation.** By using the weighted Spearman correlation coefficient (WSpearman) metric recommended by Golla et al. [8], we evaluate the accuracy of our PII-PSM and its counterparts (including our TarGuess-I-H and TarMarkov-H, as well as two benchmarks of Optimal and Min-of-All). We perform experiments on eight large-scale password datasets with different user languages and service types. Results show that our PII-PSM is more accurate than TarGuess-I-H and TarMarkov-H, and is closer to Optimal and Min-of-All, with WSpearman differences of only

0.014∼0.023 and 0.012∼0.031, respectively. This indicates the accuracy of PII-PSM, facilitating to help users set stronger passwords.

2 Preliminaries and related work

In this section, we introduce leading targeted password probabilistic models using personally identifiable information (PII), and elaborate on the preliminaries of targeted password strength meters (PSMs).

2.1 Targeted password probabilistic models

A password probabilistic model (e.g., [6,9,13,21,22,24]) can assign password construction probabilities in the password space. It can be used to construct probability-based PSMs and password guessing models. A targeted guessing attacker usually utilizes the target user's personal information to improve guessing efficiency. There are various types of personal information, such as PII (e.g., name, user name, and email address), and user identification information (e.g., users' old passwords and sister passwords from different sites) [24]. Targeted password probabilistic models can be categorized according to the personal information incorporated, e.g., Personal-PCFG [9], TarGuess-I [24], and TarMarkov [22] including PII; TarGuess-II [24], pass2path and PPSM [15], and Das et al.'s [4] including sister passwords; TarGuess-III [24] including both PII and sister passwords. In this paper, we mainly focus on the most basic yet realistic targeted guessing scenario that exploits users' PII.

Personal-PCFG. Based on the probabilistic context-free grammar (PCFG) password model [25], Li et al. [9] proposed a targeted model Personal-PCFG. It divides personal information into six categories (i.e., **U**ser name, **E**mail, **N**ame, **B**irthday, **P**hone number, and **I**D number) and combines PCFG grammar tags (i.e., **L**etter string, **D**igit string, and **S**ymbol string). Besides, it determines the password structure according to the type and length of strings and personal information. For example, the password `Li123!` for a user named `Hua Li` is converted to $N_2D_3S_1$. The rest of the training and password generation approaches for Personal-PCFG [9] are the same as PCFG [25].

However, Wang et al. [24] have shown that the above length-based PII matching approach of Personal-PCFG [9] is inaccurate to capture users' PII usage behaviors. For example, Personal-PCFG [9] transforms the passwords `Hua123` and `Liu456` of the users named `Hua Li` and `Kai Liu` into the same base structure N_3D_3 during the training phase, and the password `wang789` of the user named `Lei Wang` into N_4D_3. However, the user Hua Li uses her given name to build passwords, while users Kai Liu and Lei Wang use their family names to build passwords. Such inherently different user behaviors are misleadingly characterized in Personal-PCFG [9].

TarGuess-I. Almost at the same time, Wang et al. [24] proposed TarGuess-I, which is based on PCFG but uses a novel type-based PII matching method. For instance, TarGuess-I [24] transforms the password `Hua123` of the user named `Hua Li` into N_4D_3, and passwords `Liu456` and `wang789` (of users named `Kai Liu` and `Lei Wang`) into the same base structure N_3D_3. That is, TarGuess-I [24] uses the subscript n to represent the *sub-type* of a specific PII type (e.g., Name N and Birthday B), *not* the length of a specific PII type as in Personal-PCFG [9]. This well eliminates the misleading characterization of user behaviors in Personal-PCFG [9]. Taking N_n as an example, for a user named `Lei Wang`, N_1 stands for `leiwang`, N_2 for `lw`, and N_3 for `wang`. The grammar $\mathcal{G}_{TarGuess-I} = (\mathcal{S}, \mathcal{V}, \Sigma, \mathcal{R})$ is described as:

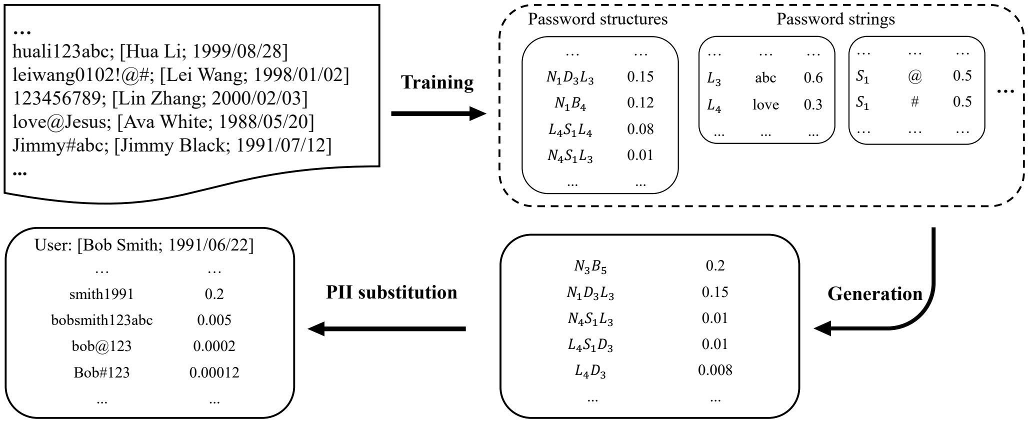


Fig. 1. An illustration of TarGuess-I [24].

1) $\mathcal{S} \in \mathcal{V}$ is the start symbol;
2) $\mathcal{V} = \{\mathcal{S}; L_n, D_n, S_n; N_n, B_n, U_n, E_n, I_n, T_n; \varepsilon\}$ is the set of grammar tags, where
   a) L_n, D_n, S_n are the grammar tags of basic PCFG [25], representing the letter, digit, and symbol strings of length n, respectively;
   b) $N_n, B_n, U_n, E_n, I_n, T_n$ are the grammar tags of TarGuess-I [24], representing the different forms of Name, Birthday, User name, Email, ID number, and Phone number distinguished by the number n;
   c) ε is the terminator;
3) Σ is the set of 94 printable ASCII characters;
4) $\mathcal{R}$ is a finite set of rules of the form $A \rightarrow \beta$, with $A \in \mathcal{V}$ and $\beta \in \mathcal{V} \cup \Sigma$.

Different from PCFG [25], when guessing the target user $user_A$'s password, TarGuess-I [24] does not directly generate the final passwords for guessing, but first generates PII-tags and then replaces them with $user_A$'s PII, as shown in Fig. 1. The experiment results of Wang et al. [24] showed that within 1,000 guesses, the guessing success rate of TarGuess-I is 37.11% higher than Personal-PCFG [9].

In 2020, Xie et al. [27,28] modified TarGuess-I by introducing grammar tags of popular passwords, keyboard patterns, and special strings. However, their experiments showed that these modifications only marginally improved the guessing success rate of the model (increased by less than 2.62% within 100 guesses). Moreover, the guessing success rate decreased on password datasets of certain service types (e.g., the train ticketing service 12306). Besides, they only used passwords leaked from Chinese services in experiments, without considering the impact of different user languages (e.g., English and Chinese) on model performance. Therefore, we adopt the original TarGuess-I [24] for evaluation and comparison in this paper.

TarMarkov. Unlike Personal-PCFG [9] and TarGuess-I [24], TarMarkov [22] is a sequence model that infers the next string state based on the current string state. Its grammar $\mathcal{G}_{TarMarkov} = (\mathcal{S}, \mathcal{V}, \mathcal{R})$ is described as below:

1) $\mathcal{S} \in \mathcal{V}$ is the start symbol;
2) $\mathcal{V} = \{\mathcal{S}; N_n, B_n, U_n, E_n, I_n, T_n; \Sigma; \varepsilon\}$ is the state set, where

a) $N_n, B_n, U_n, E_n, I_n, T_n$ have the same meaning as the corresponding grammar tags in TarGuess-I [24], except that they represent different states here;

b) Σ is the set of 94 printable ASCII characters;

c) ε is the terminator;

3) $\mathcal{R}$ is a finite set of markov state transition rules of the form $s_1 \rightarrow s_2$, with $s_1, s_2 \in \mathcal{V}^*$.

2.2 Targeted password strength meters

The above targeted password probabilistic models enable us to design targeted PSMs. Though academia has proposed a series of well-performed PSMs (e.g., [3, 6, 13, 21, 26]), the main focus is still trawling guessing scenarios, while paying little attention to the more threatening targeted guessing scenarios (especially when users' PII is available). Thus, we mainly focus on targeted PSMs using common PII.

Users' vulnerable behaviors. Users' password security/strength is intrinsically impacted by their vulnerable behaviors, mainly including [24]: (1) using popular passwords [1, 10], (2) password reuse [11, 12], and (3) using personal information [23]. Existing PSMs can prevent issue-1 and issue-2 well. For example, fuzzyPSM [21] can accurately capture users' password reuse behaviors and has a built-in base dictionary containing popular passwords. However, to the best of our knowledge, issue-3 has not been well addressed. This is because current practice using third-party corpora (e.g., common names and places) during training will result in PSM accuracy largely dependent on the corpus selection [23]. Besides, in a targeted guessing scenario where the attacker can obtain users' PII, the same password containing PII constructed by different users should be rated with different strengths. For instance, `Hua Li` and `Lei Wang` both select the password `Li123#`. The string `Li` is likely to be constructed by `Hua Li` using her family name, while for `Lei Wang` it may just be a random letter string. Thus, it is essential to propose a PSM that can accurately evaluate the strength of different users' passwords in targeted guessing.

Ideal targeted password strength meter. For the ideal PSM under trawling guessing scenarios, the formal definition given by Wang et al. [21] is as follows. For the function $M(\cdot)$ and password distribution $\mathcal{D}$, if

$$P_{\mathcal{D}}(pw_i) \geq P_{\mathcal{D}}(pw_j), \tag{1}$$

there is

$$\forall pw_i, pw_j \in \mathcal{D};\ M(pw_i) \geq M(pw_j). \tag{2}$$

Then, $M(\cdot)$ is called an ideal trawling PSM.

Analogously, targeted PSMs are adopted to evaluate the strength of the password pw in the password space under the given users' PII, so we give the formal definition of the ideal targeted PSM as follows. Suppose $user_A$ uses her own PII to construct passwords; for the function $M(\cdot)$ and password distribution $\mathcal{D}_{user_A}$, if

$$P_{\mathcal{D}_{user_A}}(pw_i) \geq P_{\mathcal{D}_{user_A}}(pw_j), \tag{3}$$

there is

$$\forall pw_i, pw_j \in \mathcal{D}_{user_A};\ M(pw_i) \geq M(pw_j). \tag{4}$$

Then, $M(\cdot)$ is called an ideal targeted PSM.

Table 1. Basic info about our eight password datasets (PII=personally identifiable information).

Dataset	Web service	Language	When leaked	Total passwords	With PII
Rootkit	Hacker forum	English	Feb. 2011	69,418	✓
12306	Train ticketing	Chinese	Dec. 2014	129,303	✓
Yahoo	Web portal	English	July 2012	453,491	
000webhost	Web hosting	English	Oct. 2015	15,299,907	
CSDN	Programmer	Chinese	Dec. 2011	6,428,632	
Dodonew	E-commerce	Chinese	Dec. 2011	16,283,140	
Rockyou	Forum	English	Dec. 2009	32,603,387	—[†]
Tianya	Forum	Chinese	Dec. 2011	29,513,716	—[†]

[†] We choose Rockyou and Tianya as base dictionaries of high-frequency substrings, so the users' PII contained in them is not considered.

Table 2. Basic info about our PII datasets (PII=personally identifiable information).

Dataset	Items num	Types of PII
PII-Rootkit	69,330	Email, User name, Name, Birthday
PII-12306	129,303	Email, User name, Name, Birthday, Phone number
PII-Yahoo	214	Email, User name, Name, Birthday
PII-000webhost	79,580	Email, User name, Name, Birthday
PII-CSDN	77,216	Email, User name, Name, Birthday, Phone number
PII-Dodonew	161,517	Email, User name, Name, Birthday, Phone number

3 Analysis of real password data

In this section, we analyze the characteristics of real-world leaked password data, and provide the basis for our improved targeted probabilistic models TarGuess-I-H and TarMarkov-H and our proposed targeted PII-PSM.

3.1 Our datasets and ethical considerations

Datasets. We analyze eight large-scale leaked password datasets and show basic information in Table 1. These datasets have different password strengths, languages, and service types, and have been widely used in password research (e.g., [8,13,16,21,23–26]). Referring to Wang et al.'s password data cleaning method [23], we first remove the junk information in the dataset, such as unnecessary headers, descriptions, footnotes, hash values, and strings containing symbols other than 94 printable ASCII characters and the space character. Besides, we remove those passwords longer than 30 for they are unlikely to be chosen by users but by password managers, while our concerned PSMs are designed to evaluate user-constructed passwords.

Two of our datasets, 12306 and Rootkit, contain certain types of users' PII (e.g., Email, User name, Name, Birthday, and Phone number). To make our targeted probabilistic models more extensible, we match the above two datasets containing users' PII with the remaining four datasets (i.e., Yahoo, 000webhost, CSDN, and Dodonew) through email, resulting in four datasets associated with PII (e.g., PII-Yahoo). The types of PII in each dataset and the number of passwords associated with PII are shown in Table 2.

Ethical considerations. Despite the fact that these password datasets are publicly available and widely used, passwords are highly private and sensitive. Thus, we still process them with caution. We only show aggregated statistics (like total passwords, top-10 HFSs%, and given name%) and treat each account as confidential, so that our use will not make attackers gain extra advantages in password guessing. Besides, we

Table 3. Top-10 popular passwords (left) and high-frequency substrings (right).[†]

Rank	English							
	Rootkit		Yahoo		000webhost		Rockyou	
1	123456	123456	123456	123456	abc123	abc123	123456	123456
2	password	password	password	101	123456a	123456a	12345	12345
3	rootkit	rootkit	welcome	ana	12qw23we	12qw23we	123456789	123456789
4	111111	111111	ninja	100	123abc	123abc	password	password
5	12345678	12345678	abc123	cat	a123456	a123456	iloveyou	iloveyou
6	qwerty	qwerty	123456789	red	123qwe	123qwe	princess	princess
7	123456789	123456789	12345678	star	secret666	secret	1234567	1234567
8	123123	123123	sunshine	dog	YfDbUfNjH10305070	asd	rockyou	rockyou
9	qwertyui	12345	princess	102	asd123	qwerty	12345678	12345678
10	12345	1234	qwerty	ard	qwerty123	YfDbUfNjH10305070[‡]	abc123	abc123
%	3.94	**5.38**	1.01	**1.93**	0.79	**1.35**	2.05	2.05

Rank	Chinese							
	12306		CSDN		Dodonew		Tianya	
1	123456	123456	123456789	123456789	123456	123456	123456	123456
2	a123456	a123456	12345678	12345678	a123456	a123456	111111	123
3	5201314	5201314	11111111	11111111	123456789	123456789	000000	111
4	123456a	123456a	dearbook	dearbook	111111	111111	123456789	12345678
5	111111	111111	00000000	00000000	5201314	520	123123	520
6	woaini1314	123123	123123123	123123123	123123	123	123321	321
7	123123	000000	1234567890	1234567890	a321654	a321654	5201314	123123
8	000000	woaini	88888888	88888888	12345	123123	12345678	666666
9	qq123456	qq123456	111111111	111111111	000000	000000	666666	111
10	1qaz2wsx	1qaz	147258369	147258369	123456a	1234	111222tianya	tianya
%	3.28	**3.78**	10.44	**10.44**	0.79	**1.75**	7.43	**16.33**

[†] A high-frequency substring in blue indicates that it is different from the popular password of the same rank in the same password dataset. In Chinese, the homophonic meaning of 5201314 is "I love you (520) forever (1314)". The Chinese pinyin woaini means "I love you".

[‡] The letter segment YfDbUfNjH can be mapped to a Russian word that means "navigator", and why it is so popular is beyond our comprehension.

process all our password-related data on computers not connected to the Internet, and delete sensitive info after finishing experiments. Furthermore, our use of these datasets is not only beneficial for research on targeted guessing and password strength evaluation, but also for security admins to protect user account security.

3.2 High-frequency substrings (HFSs) and popular passwords

When constructing passwords, users may adopt more common and fine-grained HFSs as password components than popular passwords [20]. To investigate this issue, we count top-10 HFSs (see Secs. 4.1 and 5.1 for detailed identification approaches and parameter settings) and popular passwords in our eight password datasets, and calculate the proportion of passwords containing them in the dataset. The results are shown in Table 3. It can be seen that 1.35%∼16.33% of the passwords contain top-10 HFSs, while only 0.79%∼7.43% contain top-10 popular passwords. That is, *HFSs are more common in users' passwords than popular passwords, indicating that users may prefer to utilize HFSs to construct passwords.* In addition, when constructing passwords, Chinese users prefer to use simple digit strings (e.g., 123456, 00000000, and 123123) and some strings with semantics (e.g., 5201314 and woaini related to "love"). In contrast, English users tend to use a combination of letter and digit strings (e.g., abc123 and qwerty123) and common English words/phrases (e.g., password, iloveyou, and cat).

Table 4. Percentages (%) of users constructing passwords with (left) and only with (right) their heterogeneous personal information, popular passwords, and high-frequency substrings (HFSs).†

Typical usages of PII (examples)	English						Chinese					
	PII-Rootkit (69,330)		PII-Yahoo (214)		PII-000webhost (2,950)		PII-12306 (129,303)		PII-CSDN (77,439)		PII-Dodonew (161,510)	
Top-10 popular passwords (`123456`)	2.45	2.14	0.06	0.02	0.79	0.47	1.56	1.01	9.32	8.42	4.61	2.18
Top-100 popular passwords	2.76	2.31	0.09	0.04	0.87	0.53	1.78	1.14	26.31	24.54	4.91	2.43
Top-10 HFSs (`123`, `abc`)	2.98	2.02	0.19	0.00	3.01	0.45	1.78	1.08	12.59	8.42	5.33	2.18
Top-100 HFSs	6.32	2.25	0.59	0.04	6.38	0.49	3.45	1.12	29.32	27.73	7.57	2.39
Full name (`hua li`)	1.38	0.75	2.34	1.87	2.44	1.32	5.02	1.13	4.85	1.81	4.68	0.82
Family name (`li`)	2.28	0.78	4.67	1.87	3.73	1.46	11.23	0.00	9.75	0.00	11.15	0.01
Given name (`hua`)	0.49	0.07	0.93	0.00	0.75	0.20	6.61	0.07	6.26	0.08	6.49	0.07
Abbr. full name (`lh`, `hl`, `hli`)	0.15	0.01	0.00	0.00	0.20	0.00	13.13	0.00	9.42	0.00	13.64	0.02
Full Birthday (`19980102`, `01021998`)	0.08	0.06	0.47	0.00	0.10	0.07	4.33	1.77	6.29	5.16	3.12	1.00
Year of birthday (`1982`)	0.75	0.01	1.40	0.00	1.12	0.00	10.78	0.00	11.37	0.00	8.92	0.00
Date of birthday (`0102`, `0201`)	0.44	0.01	0.47	0.00	0.58	0.00	10.03	0.00	11.84	0.00	8.32	0.00
Abbr. birthday (`199812`, `980102`)	0.10	0.05	0.00	0.00	0.20	0.14	3.31	1.12	2.89	1.45	2.37	0.59
User name strings (`neko_10`, `neko`)	2.91	0.86	4.01	1.40	2.20	1.32	3.57	1.22	0.91	0.67	2.61	1.71
Email strings (`loveu@exa`, `loveu`)	0.77	0.49	4.38	1.87	1.32	0.78	3.23	1.95	4.65	2.48	5.37	3.08
Phone strings (`123-4567-8900`)	—	—	—	—	—	—	0.07	0.01	0.50	0.45	0.11	0.11

† All decimals in the table are in "%". For instance, 2.45 in the upper left corner means that 2.45% of the 69,330 PII-Rootkit users employ top-10 popular passwords to build passwords; 2.14 means that 2.14% of these 69,330 PII-Rootkit users' passwords are just top-10 popular passwords.

Further, we extract top-10 and top-100 HFSs and popular passwords, respectively, and use them together with some PII-tags (e.g., name and email) to mark and analyze passwords. The results are shown in Table 4. The left column corresponding to each dataset in Table 4 is the proportion of passwords containing the tag, and the right column is the proportion of passwords that are exactly the tag. For example, if the tag content is `123456`, the counted passwords in the left column include `1234567` and `a123456`, and that in the right column only include `123456`. It can be seen that passwords with a specific PII-tag account for a considerable portion, the highest being 13.64%, showing that users' vulnerable behaviors of using PII to construct passwords are common.

Here we focus on HFS and popular password tags, and find that: 1) For the same dataset, the proportion of passwords containing top-10/top-100 HFSs (on the left column) is greater than that of top-10/top-100 popular passwords (in two columns), indicating that *HFSs can capture more fine-grained password characteristics than popular passwords*; 2) The proportion of passwords that are exactly the top-10/top-100 HFS-tags (on the right column) is close to the proportion of top-10/top-100 popular passwords (in two columns), indicating that some HFSs are directly used by users as passwords and play the role of popular passwords; 3) A larger scale of HFS-tags (e.g., from top-10 to top-100) can significantly cover more passwords and capture more password characteristics.

3.3 Password structure

To investigate how HFS-tags and popular password-tags characterize password structure, we convert the two types of tags into grammar tags of $\mathcal{G}_{TarGuess-I}$ and $\mathcal{G}_{TarMarkov}$. More specifically, we count the top-100 popular passwords and HFSs, labeled "P_n" meaning a popular password of length n and "H_n^i" meaning an HFS ranked i in those substrings of length n. Following the longest-prefix matching rule, we first match PII segments in a

Table 5. Top-10 password structures marked with popular password tags (P_n; on the left) and high-frequency substring tags (H_n^i; on the right) of each dataset, and proportions of password structures containing the two tags (P_n% and H_n^i%) in each dataset. (P_n=a popular password of length n, and H_n^i=a high-frequency substring ranked i in those substrings of length n)

Rank	English						Chinese					
	Rootkit		Yahoo		000webhost		12306		CSDN		Dodonew	
1	P_6	H_6^1	P_6	H_6^1	P_6	H_6^1	P_6	H_6^1	P_8	D_8	E_1	E_1
2	P_8	H_8^1	P_8	H_8^1	P_8	H_8^1	D_6	H_6^2	D_8	H_8^1	D_7	H_7^3
3	D_8	H_6^2	D_6	H_6^2	P_7	H_6^2	D_7	D_6	E_1	E_1	P_6	H_6^1
4	L_8	H_8^2	L_6	H_8^2	D_6	H_7^1	N_2D_6	D_7	B_1	B_1	D_6	H_6^2
5	P_7	H_7^1	L_8	L_8	D_8	H_8^2	U_1	H_7^1	D_9	D_9	D_8	D_6
6	N_2D_6	H_6^3	D_9	D_9	L_6	N_1D_6	D_8	U_1	N_2D_6	N_2D_6	N_2D_6	N_2D_6
7	D_5	$N_2H_6^1$	P_9	P_9	N_3D_1	U_1D_1	E_1	D_8	U_1	U_1	U_1D_7	U_1D_7
8	U_1D_1	N_2D_6	N_1D_1	H_9^1	N_4D_1	N_1D_1	N_2D_7	E_1	D_{11}	D_{11}	N_2D_7	N_2D_7
9	N_3D_1	U_1D_1	U_1D_1	N_1D_1	E_1D_3	N_3D_1	U_3	N_2D_7	N_2D_7	N_2D_7	U_1	U_1
10	N_4D_1	D_5	N_3D_1	H_8^3	D_{10}	N_1	U_2D_6	$N_2H_7^1$	D_10	H_10^1	U_2D_6	$U_2H_6^1$
P_n%	14.12		16.78		10.14		6.31		10.11		4.25	
H_n^i%	30.13		34.28		26.13		17.22		16.12		6.39	

password, then use the remaining segments to match P_n and H_n^i, and obtain password structures. We show the top-10 password structures and the proportions containing P_n and H_n^i in Table 5. We find that the top-10 password structures of these password datasets include a number of simple structures independent of PII-tags, such as single P_n, H_n^i, L_n, and D_n. Therefore, adding P_n and H_n^i tags to $\mathcal{G}_{TarGuess-I}$ and $\mathcal{G}_{TarMarkov}$ are likely to facilitate password probabilistic models to identify simple yet common strings in passwords more effectively, thereby helping to build more accurate PSMs.

What's more, H_n^i *can characterize more fine-grained password structures than* P_n. For example, after introducing H_n^i tags, the top-2 password structures of Rootkit are further refined into $P_6 \rightarrow H_6^1, H_6^2$ and $P_8 \rightarrow H_8^1, H_8^2$. Similarly, there are $P_9 \rightarrow H_9^1$; $L_8 \rightarrow H_8^1, H_8^2$; and $U_2D_6 \rightarrow U_2H_6^1$. This can help TarGuess-I [24] construct more HFSs rather than redundant segments when generating passwords. For TarMarkov [22], more refined and diverse password structures are helpful to well solve the long-standing issue of data sparsity. To sum up, we take H_n^i as a new grammar tag to improve the leading targeted password probabilistic models TarGuess-I [24] and TarMarkov [22].

4 Methodology

In this section, we first propose two new targeted password probabilistic models, TarGuess-I-H and TarMarkov-H, which can identify HFSs in users' passwords. Based on these two models, we devise a new targeted PSM called PII-PSM.

4.1 Improved password probabilistic models

To help construct accurate and practical targeted PSMs, we first need to devise well-performed password probabilistic models. Thus, we propose the improved TarGuess-I-H and TarMarkov-H as follows.

Our TarGuess-I-H. We introduce HFS as a new grammar tag into TarGuess-I [24], and propose a novel targeted password probabilistic model TarGuess-I-H. Its grammar $\mathcal{G}_{TarGuess-I-H} = (\mathcal{S}, \mathcal{V}, \Sigma, \mathcal{R})$ is described as below:

1) $\mathcal{S} \in \mathcal{V}$ is the start symbol;
2) $\mathcal{V} = \{\mathcal{S}; L_n, D_n, S_n; N_n, B_n, U_n, E_n, I_n, T_n; H_n^i; \varepsilon\}$ is the set of grammar tags, where
   a) L_n, D_n, S_n are the grammar tags of basic PCFG [25], representing the letter, digit, and symbol strings of length n, respectively;
   b) $N_n, B_n, U_n, E_n, I_n, T_n$ are the grammar tags of TarGuess-I [24], representing the different forms of Name, Birthday, User name, Email, ID number, and Phone number distinguished by the number n;
   c) H_n^i is proposed in this paper *for the first time*, representing the set of strings ranked i among those substrings of length n in descending order of frequency;
   d) ε is the terminator;
3) Σ is the set of 94 printable ASCII characters;
4) $\mathcal{R}$ is a finite set of rules of the form $A \to \beta$, with $A \in \mathcal{V}$ and $\beta \in \mathcal{V} \cup \Sigma$.

Our TarMarkov-H. TarMarkov [22] is a sequence model that infers the next string state based on the current string state. We introduce HFS as a new state into TarMarkov [22], and propose a novel targeted password probabilistic model TarMarkov-H. Its grammar $\mathcal{G}_{TarMarkov-H} = (\mathcal{S}, \mathcal{V}, \mathcal{R})$ is described as below:

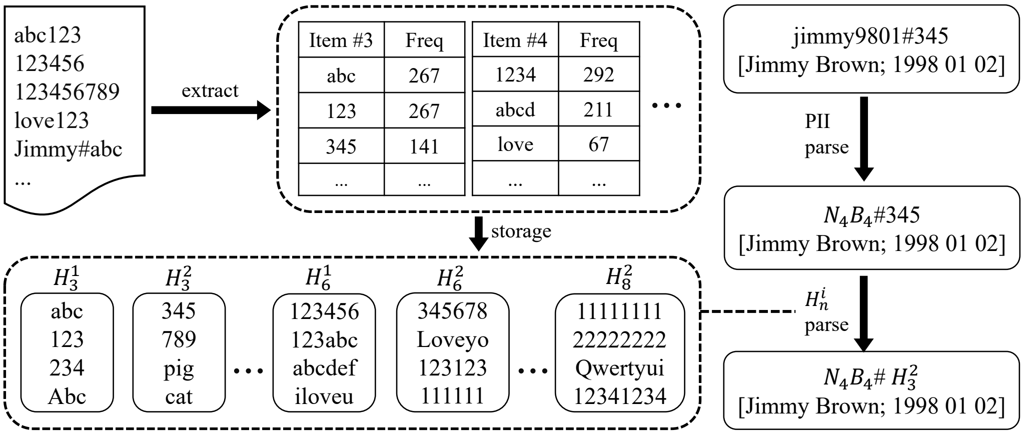
1) $\mathcal{S} \in \mathcal{V}$ is the start symbol;
2) $\mathcal{V} = \{\mathcal{S}; N_n, B_n, U_n, E_n, I_n, T_n; H_n^i; \Sigma; \varepsilon\}$ is the state set, where
   a) $N_n, B_n, U_n, E_n, I_n, T_n$ and H_n^i have the same meaning as the corresponding grammar tags in TarGuess-I-H, except that they represent different states in TarMarkov-H;
   b) Σ is the set of 94 printable ASCII characters;
   c) ε is the terminator;
3) $\mathcal{R}$ is a finite set of markov state transition rules of the form $s_1 \to s_2$, with $s_1, s_2 \in \mathcal{V}^*$.

High-frequency substrings (HFSs). In a password dataset, HFSs are password substrings with the frequency exceeding a certain threshold, and they can be identified by taking the following steps:

1) Record the count $C(p_s)$ of each password substring p_s with the length $n \geq 3$;
2) Set the threshold T_1 and delete the substrings with a count less than T_1;
3) Modify the substring count record as

$$C(p_s)^{new} = C(p_s)^{old} - \sum_{c \in \Sigma} [C(c + p_s)^{old} + C(p_s + c)^{old}], \tag{5}$$

   where $C(p_s)^{old}$ is the original count record of p_s, and $c + p_s$ and $p_s + c$ respectively mean that the character c is concatenated to the beginning and end of p_s;
4) Set the threshold T_2 and identify p_s as a HFS if $C(p_s)^{new} \geq T_2$;
5) Store HFSs with the same length n into the set H_n ($n \geq 3$), arrange them in descending order of count, and denote the set of substrings ranked i in H_n as H_n^i. The parsing process is shown in Fig. 2. Parameter setups of T_1, T_2, and n are detailed in Sec. 5.1.



Item #3	Freq
abc	267
123	267
345	141
...	...

Item #4	Freq
1234	292
abcd	211
love	67
...	...

Fig. 2. An illustration of H_n^i-tag processing. H_n^i denotes the high-frequency substring ranked i in those substrings of length n.

4.2 Our targeted PII-PSM

Our proposed password probabilistic models TarGuess-I-H and TarMarkov-H introduced above can be individually transformed into two targeted PSMs. Still, we combine these two models to construct a new targeted PSM called PII-PSM, because Dong et al. [6] found that: In online guessing (often guess number$<10^4$), PCFG-based password models usually outperform markov-based ones; on the contrary, in offline guessing (often guess number$>10^4$), markov-based ones usually outperform PCFG-based ones. Thus, taking into account PSM performance under both online and offline guessing, we construct our PII-PSM by weighing the strength scores of the PCFG-based TarGuess-I-H and markov-based TarMarkov-H. For a password pw, we denote the probabilities calculated by TarGuess-I-H and TarMarkov-H as p_1 and p_2, and the corresponding weights are α ($\alpha \in [0,1]$) and 1-α (detailed α setups are in Sec. 5.2). Then the strength score of pw evaluated by PII-PSM under targeted guessing scenarios can be denoted as

$$Final\ score_{pw} = \alpha \times (-\log_2 p_1) + (1-\alpha) \times (-\log_2 p_2). \tag{6}$$

Justification for PII-PSM. Under targeted guessing scenarios, to evaluate the strength of the password pw in the password space, it is ideal to obtain all of $user_A$'s personal data (e.g., all PII and all existing passwords), and compute $user_A$'s password distribution space as $P(pw|all\ user_A's\ personal\ data,\ public\ data)$. However, this is intrinsically/virtually impossible to obtain all of $user_A$'s personal data. Fortunately, $user_A$'s password distribution space can be approximated more accurately when $user_A$'s more personal data (e.g., common PII) is available. Accordingly, the password strength evaluation models under targeted guessing using PII hold that

$$\forall pw, user_A, user_B;$$

$$\forall P(pw|PII_{user_A},\ public\ data) \neq P(pw|PII_{user_B},\ public\ data). \tag{7}$$

That is, the probabilities of pw in $user_A$'s password space and $user_B$'s are different.

Table 6. Training and test settings for targeted password guessing and strength evaluation.

Exp#	Language	Training set	Test set	Auxiliary dataset
1	English	1/2 PII-Rootkit	1/2 PII-Rootkit	Rockyou
2		1/2 PII-Yahoo	1/2 PII-Yahoo (1/2 Yahoo†)	
3		1/2 PII-000webhost	1/2 PII-000webhost	
4	Chinese	1/2 PII-12306	1/2 PII-12306	Tianya
5		1/2 PII-CSDN	1/2 PII-CSDN	
6		1/2 PII-Dodonew	1/2 PII-Dodonew	

† Since PII-Yahoo size is small (only 214) and thus unable to evaluate targeted PSMs accurately, in targeted password strength evaluation, we randomly sample half of the passwords from Yahoo (226,731) as the test set. When an account in the test set lacks PII, PCFG-based and markov-based models degenerate into basic PCFG [25] and Markov [11].

It is worth noting that, when evaluating password strength, our PII-PSM first replaces the PII-related segments in pw with corresponding PII-tags, and obtains a new password form $pw_{PII-tag}$. For example, if a user's name, birthday, and password are `Li Wang`, `1998/08/18`, and `wang980818abc`, respectively, the converted $pw_{PII-tag}$ is N_3B_8abc. Since PII-PSM uses the same grammar rules for all users when calculating $pw_{PII-tag}$, there is

$$\forall pw_{PII-tag}, user_A, user_B;$$

$$P(pw_{PII-tag}|PII_{user_A},\ public\ data) = P(pw_{PII-tag}|PII_{user_B},\ public\ data). \qquad (8)$$

When given users' PII, pw is determined by $pw_{PII-tag}$, satisfying Eq. 7 under targeted guessing scenarios.

5 Experiments

In this section, we first experimentally quantify the improvement of our proposed TarGuess-I-H and TarMarkov-H over the basic TarGuess-I [24] and TarMarkov [22]. Then, we evaluate the accuracy of our PII-PSM and its counterparts (including our TarGuess-I-H and TarMarkov-H, as well as two benchmarks of Optimal and Min-of-All) using the weighted Spearman metric recommended in CCS'18 [8].

5.1 Validation of the improvements

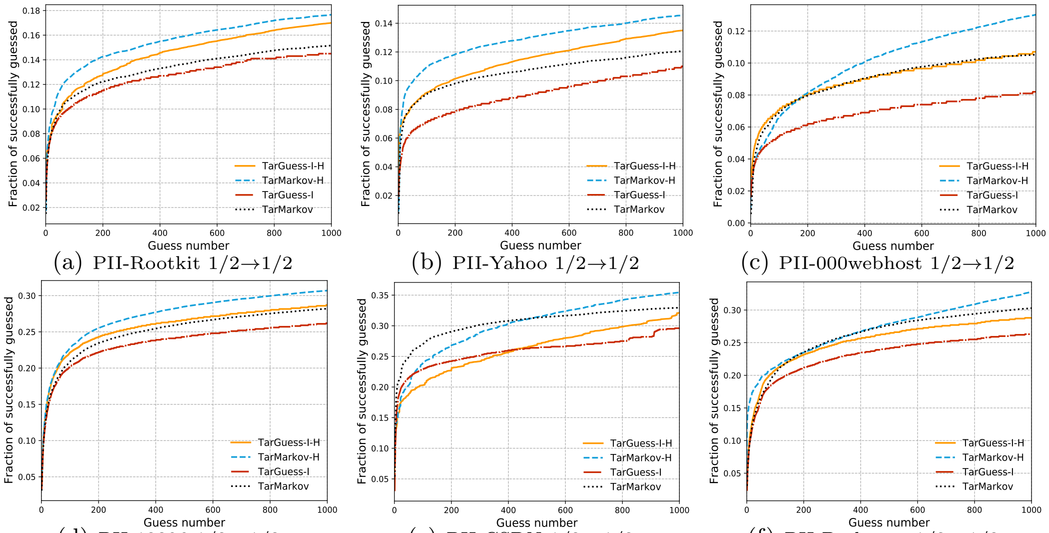
Many studies (e.g., [3, 6, 13, 15]) have shown that password probabilistic models with good guessing ability can be used to construct accurate and practical PSMs. Therefore, we first perform password guessing experiments to demonstrate that our TarGuess-I-H and TarMarkov-H are indeed significantly improved over the original TarGuess-I [24] and TarMarkov [22], and thus are likely to be used to build more accurate targeted PSMs.

Experiment setups. The user language, service type, and password policy are the three most influential factors on password security and strength in turn [23]. The closer the training set is to the passwords of the target site, the better [23]. Therefore, we sample the training and test sets from the same dataset, and show the experiment settings in Table 6. Taking Exp #1 as an example, we randomly divide PII-Rootkit into two equal-sized parts used for training and testing, respectively.

Since our TarGuess-I-H and TarMarkov-H have considered the impact of HFSs and introduced H_n^i tags, we need to select third-party auxiliary datasets to build the HFS dictionary. We use Rockyou and Tianya as auxiliary datasets for English and Chinese

Table 7. Settings of targeted password probabilistic models.

Model	L/D/S-tags	PII-tags	H_n^i-tag	Model order	Probability threshold
TarGuess-I [24]	✓	✓		—	10^{-6}
TarMarkov [22]		✓		3	10^{-6}
Our TarGuess-I-H	✓	✓	✓	—	10^{-6}
Our TarMarkov-H		✓	✓	3	10^{-6}



(a) PII-Rootkit 1/2→1/2 (b) PII-Yahoo 1/2→1/2 (c) PII-000webhost 1/2→1/2

(d) PII-12306 1/2→1/2 (e) PII-CSDN 1/2→1/2 (f) PII-Dodonew 1/2→1/2

Fig. 3. Experiment results of targeted guessing scenarios on six different datasets. Sub-figures (a) to (c) are on datasets from English sites, and (d) to (f) are on datasets from Chinese sites.

training sets, respectively, because the two low-strength datasets contain a large number of weak passwords [6,16], and have been widely used in leading password research (e.g., [6,11,13,15,16,24,25]) in recent years. Besides, to make our TarGuess-I-H and TarMarkov-H perform well, we have implemented multiple experiments with different HFS parameter configurations, and finally set the HFS thresholds T_1=500 and T_2=50, the HFS length $3 \leq n \leq 8$, and the HFS dictionary composed of top-100 HFSs. The settings of targeted password probabilistic models are shown in Table 7.

Experiment results. We show the experiment results in Fig. 3 and find that:

1) In Figs. 3(a)∼3(d), the performances are ordered as our TarMarkov-H, our TarGuess-I-H, TarMarkov [22], and TarGuess-I [24]. In Figs. 3(e) and 3(f), it is our TarMarkov-H, TarMarkov [22], our TarGuess-I-H, and TarGuess-I [24]. On average, our added H_n^i-tags make the performances of our TarMarkov-H and TarGuess-I-H higher than the basic TarMarkov [22] and TarGuess-I [24] by 1.72% and 3.11%, respectively. The reasons are as follows: (a) According to Sec. 3.2, users tend to use HFSs when constructing passwords. Thus, password models with H_n^i-tags can well identify HFSs in passwords during training, and can better learn users' password construction habits when generating passwords, thereby improving model performance. (b) According to Sec. 3.3, password models with H_n^i-tags can more accurately capture password structures, such as $L_8 \rightarrow H_8^1, H_8^2$, and thus reduce redundant segments when generating passwords. (c) Introducing H_n^i-tags can increase the variety of password structures. For example, in Exp #1 of Table 6, the extracted password structures increase from 53,168

to 76,133 (a 43.19% increase). In this way, our TarMarkov-H can mitigate the inherent data sparseness issue of markov-based password models.

2) Our TarGuess-I-H outperforms TarGuess-I [24] by 2.02%∼3.43% (relative increases are 8.33%∼15.22%) and our TarMarkov-H outperforms TarMarkov [22] by 1.45%∼2.63% (relative increases are 9.73%∼15.22%). This is because markov-based TarMarkov-H and TarMarkov [22] can generate more novel passwords than PCFG-based TarGuess-I-H and TarGuess-I [24]. In contrast, the performance of PCFG-based models is largely limited by password structures in the training set, especially when the training size is small.

3) In Figs. 3(e) (on PII-CSDN) and 3(f) (on PII-Dodonew), TarGuess-I-H and TarGuess-I [24] perform worse than TarMarkov [22]. A possible explanation is that, PCFG-based TarGuess-I-H and TarGuess-I [24] parse passwords from the segment level, and many passwords in PII-CSDN and PII-Dodonew contain digit strings [6] (marked as D_n). This causes the model to generate a large number of redundant password candidates when filling D_n in the password generation stage, thus reducing the performance. In contrast, markov-based TarMarkov [22] parses passwords from the character level, reducing generating redundant digit strings.

Summary. By adding HFS tags, our TarGuess-I-H and TarMarkov-H significantly outperform the basic TarGuess-I [24] and TarMarkov [22] in most cases, suggesting that our two models can be employed to build accurate targeted PSMs.

5.2 PSM accuracy evaluation

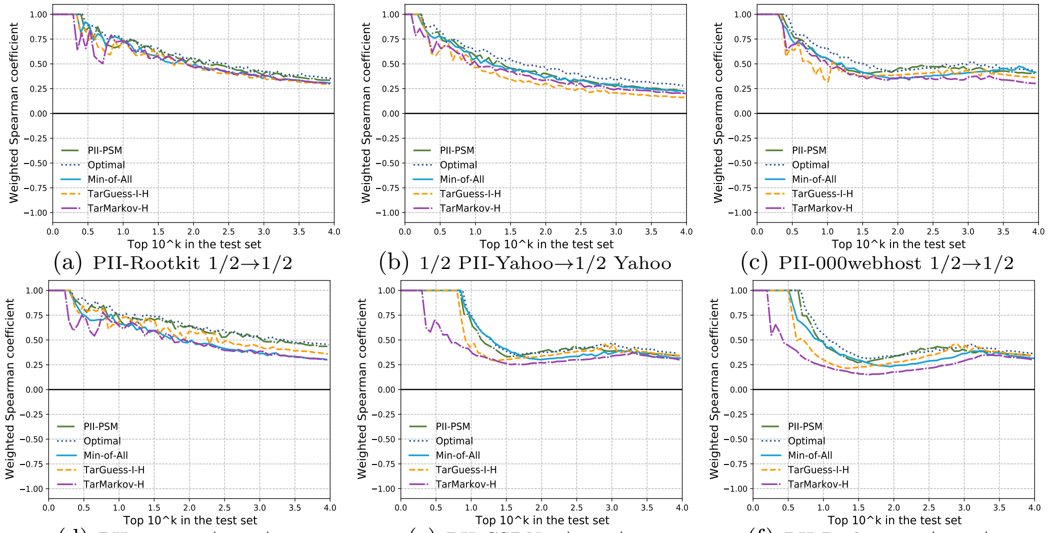
PSM accuracy evaluation metric. Accuracy is the most essential property of a PSM. Only PSMs with accurate strength feedback can indeed nudge users to choose stronger passwords [17,18]. In recent years, researchers have used various metrics (e.g., Spearman and Kendall correlation coefficients) to measure PSM accuracy [13,21,26]. At CCS'18, Golla et al. [8] tested 19 candidate metrics for evaluating PSM accuracy and selected the weighted Spearman correlation coefficient (WSpearman), because it is robust to monotonic transformations, disturbances, and quantization. Thus, inspired by Golla et al.'s work [8], we use WSpearman to evaluate PSM accuracy, calculated as

$$WSpearman(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=1}^{n}[w_i(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{\sum_{i=1}^{n}[w_i(x_i - \bar{x})^2]\sum_{i=1}^{n}[w_i(y_i - \bar{y})^2]}}, \tag{9}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the weighted rank vectors of the ideal PSM and the tested PSM, x_i and y_i are the members of $\mathbf{X}$ and $\mathbf{Y}$ ranked i (1≤i≤n) in descending order of frequency, $\bar{x}$ and $\bar{y}$ are the weighted means of $\mathbf{X}$ and $\mathbf{Y}$, and w_i is the password frequency ranked i in the test set. The higher the WSpearman value (in [-1,1]), the more accurate the PSM.

Experiment setups. TarGuess-I-H and TarMarkov-H in this section refer to targeted PSMs based on these two models. We show the experiment setups of targeted password strength evaluation in Table 6, and config the targeted PSMs and benchmarks for comparison and evaluation as follows:

- **Our TarGuess-I-H and TarMarkov-H.** The parameter settings of these two PSMs are shown in Table 7. The strength of the password pw is evaluated as $-\log_2 p$, where p is the construction probability of pw under the corresponding model.
- **Our PII-PSM.** The password strength evaluated by our PII-PSM is obtained by weighting the strengths output by TarGuess-I-H and TarMarkov-H with α and 1-α; see Eq. 6. α is initialized to a random value in [0,1], optimized by the stochastic gradient



(a) PII-Rootkit 1/2→1/2 (b) 1/2 PII-Yahoo→1/2 Yahoo (c) PII-000webhost 1/2→1/2

(d) PII-12306 1/2→1/2 (e) PII-CSDN 1/2→1/2 (f) PII-Dodonew 1/2→1/2

Fig. 4. Weighted Spearman correlation coefficient of our targeted PSMs. Sub-figures (a) to (c) are on datasets from English sites, and (d) to (f) are on datasets from Chinese sites.

descent (SGD) approach with batchsize=n (i.e., every n passwords in the training/testing set are split into a batch). In this way, the impact of randomly/manually setting the α value on PSM accuracy can be eliminated. We calculate WSpearman for each batch in the test set and the corresponding batch in the training set, and use it as a loss to penalize α until α reaches convergence. The convergent α and SGD parameter setups are shown in Table 8. What's more, since the training set is known to our PII-PSM, the optimization for α is feasible, which contributes to the practicality of PII-PSM.

• **Min-of-All.** Min-of-All is a PSM strength benchmark indicating a conservative approximation of password strength. It is proposed by Ur et al. [19] and is widely used in leading PSM research (e.g., [13, 26]). Regarding a password, Min-of-All takes the minimum value of the results of all evaluated PSMs as the password strength. In this paper, we also adopt Min-of-All as a PSM strength benchmark, which is calculated as the minimum evaluation results of our TarGuess-I-H, TarMarkov-H, and PII-PSM.

Table 8. Convergent α and SGD setups.

Exp#	α	SGD		
		Batchsize	Step length	$\Delta^{\dagger}$
1	0.623	50	[0.2,0.8]	10^{-5}
2	0.629	100	[0.5,1.0]	
3	0.642	50	[0.2,0.8]	
4	0.612	50	[0.2,0.8]	
5	0.601	50	[0.2,0.8]	
6	0.617	50	[0.2,0.8]	

† When $|\alpha_{new} - \alpha_{old}| \leq \Delta$, SGD stops optimizing and chooses α as the optimal value.

• **Our Optimal.** To indicate the optimal evaluation capability that practical PSMs can achieve, we propose a new PSM strength benchmark, Optimal. Regarding a password, Optimal takes the one closest to the real frequency rank among the results of all evaluated PSMs (e.g., our TarGuess-I-H, TarMarkov-H, and PII-PSM in this paper) as the password strength. Note that an Optimal PSM is unlikely to be deployed in real-world scenarios, because it can hardly know the real password rank. Nevertheless, since the password rank is known in our experiments, Optimal is effective as a PSM strength benchmark.

Experiment results. We show WSpearman of targeted PSMs in Fig. 4 and find that:

1) The Wspearman value of PII-PSM is higher than TarGuess-I-H and TarMarkov-H, and fluctuates more slightly within top-10^2 passwords. This is because our adopted SGD effectively optimizes the weights α and 1-α of TarGuess-I-H and TarMarkov-H that constitute PII-PSM, thus improving PII-PSM accuracy. Note that the convergence value of α in Table 8 is 0.601∼0.642 instead of around 0.5, indicating that TarGuess-I-H and TarMarkov-H have different effects/contributions to PII-PSM accuracy. A possible explanation is that, according to Fig. 4, TarMarkov-H generally fluctuates more strongly than TarGuess-I-H (especially within top-10^2 passwords), indicating that the former is less accurate than the latter in evaluating weak passwords. Thus, SGD will give the more accurate TarGuess-I-H a higher weight.

2) In Figs. 4(e) (on PII-CSDN) and 4(f) (on PII-Dodonew), the WSpearman value of all PSMs decreases rapidly in top-3∼top-10 (i.e., top-$10^{0.5}$∼top-$10^{1.0}$) passwords, and increases slowly from top-30 (i.e., top-$10^{1.5}$) until stable. The possible reason is that, in PII-CSDN and PII-Dodonew, the top-10 passwords account for a large proportion (8.33% and 7.91%), resulting in a more concentrated password probability distribution that can be accurately evaluated by PSMs. Thus, the WSpearman value is stable at 1 in top-10. While the followed passwords have a more uniform probability distribution and thus PSMs cannot accurately evaluate some of the passwords. As a result, the WSpearman value decreases significantly. With more passwords being evaluated, PSMs can more accurately capture password distribution characteristics, so the WSpearman value gradually stabilizes.

3) Compared to individual TarGuess-I-H and TarMarkov-H, PII-PSM is closer to the Optimal benchmark, and the WSpearman differences are only 0.014∼0.023. This suggests that PII-PSM has almost the optimal evaluation ability that the compared practical PSMs can achieve, and thus is more accurate. In addition, PII-PSM is also closer to the Min-of-All benchmark, and the WSpearman differences are only 0.012∼0.031. This indicates that PII-PSM evaluates password strength more strictly and conservatively, which may help nudge users to select stronger passwords.

Summary. Our PII-PSM obtained by combining and weighting TarGuess-I-H and TarMarkov-H is more accurate than both individual PSMs, and is closer to the PSM accuracy benchmarks Min-of-All and our Optimal. This suggests that a rational combination of multiple PSMs that perform well in different guessing scenarios (e.g., online and offline guessing) is helpful for designing accurate targeted PSMs.

6 Conclusion

We have introduced the high-frequency substring (HFS) as a new grammar tag into leading targeted password probabilistic models TarGuess-I [24] and TarMarkov [22], and proposed our improved models TarGuess-I-H and TarMarkov-H. Then, we weighted and combined our two models and, *for the first time*, devised a practical targeted password strength meter (PSM) called PII-PSM that exploits common personally identifiable information (PII; e.g., name and birthday). Extensive evaluation experiments show that our PII-PSM is more accurate than individual TarGuess-I-H and TarMarkov-H, and is closer to two benchmarks of Optimal and Min-of-All. What's more, eight large-scale password datasets across different user languages and service types indicate the practicality of our PII-PSM. We believe that our targeted probabilistic models and PII-PSM can shed light on both existing password practice and future password research.

Acknowledgment. The authors are grateful to the anonymous reviewers for their invaluable comments. Ding Wang is the corresponding author. This research was in part supported by the National Natural Science Foundation of China (Grant No.62172240), and by the Natural Science Foundation of Tianjin, China (Grant No.21JCZDJC00190).

References

1. Bonneau, J.: The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In: Proc. IEEE S&P 2012. pp. 538–552
2. Bonneau, J., Herley, C., van Oorschot, P., Stajano, F.: Passwords and the evolution of imperfect authentication. Commun. ACM **58**(7), 78–87 (2015)
3. Castelluccia, C., Dürmuth, M., Perito, D.: Adaptive password-strength meters from markov models. In: Proc. NDSS 2012. pp. 1–14
4. Das, A., Bonneau, J., Caesar, M., Borisov, N., Wang, X.: The tangled web of password reuse. In: Proc. NDSS 2014. pp. 1–15
5. Dellinger, A.: Personal data of 533 million facebook users leaks online (April 2021), `https://shorturl.at/dlHUV`
6. Dong, Q., Jia, C., Duan, F., Wang, D.: RLS-PSM: A robust and accurate password strength meter based on reuse, leet and separation. IEEE Trans. Inf. Forensics Secur. **16**, 4988–5002 (2021)
7. Gatlan, S.: Hacker leaks full database of 77 million nitro pdf user records (Jan 2021), `https://shorturl.at/fjwI5`
8. Golla, M., Dürmuth, M.: On the accuracy of password strength meters. In: Proc. ACM CCS 2018. pp. 1567–1582
9. Li, Y., Wang, H., Sun, K.: A study of personal information in human-chosen passwords and its security implications. In: Proc. INFOCOM 2016, pp. 1–9
10. Li, Z., Han, W., Xu, W.: A large-scale empirical analysis of chinese web passwords. In: Proc. USENIX SEC 2014. pp. 559–574
11. Ma, J., Yang, W., Luo, M., Li, N.: A study of probabilistic password models. In: Proc. IEEE S&P 2014. pp. 689–704
12. Mazurek, M.L., Komanduri, S., Vidas, T., Cranor, L.F., Kelley, P.G., Shay, R., Ur, B.: Measuring password guessability for an entire university. In: Proc. ACM CCS 2013. pp. 173–186
13. Melicher, W., Ur, B., Segreti, S.M., Komanduri, S., Bauer, L., Christin, N., Cranor, L.F.: Fast, lean, and accurate: Modeling password guessability using neural networks. In: Proc. USENIX SEC 2016. pp. 1–17
14. Morris, C.: Massive data leak exposes 700 million linkedin users' information (June 2021), `https://shorturl.at/mDGQ1`
15. Pal, B., Daniel, T., Chatterjee, R., Ristenpart, T.: Beyond credential stuffing: Password similarity models using neural networks. In: Proc. IEEE S&P 2019. pp. 417–434
16. Pasquini, D., Gangwal, A., Ateniese, G., Bernaschi, M., Conti, M.: Improving password guessing via representation learning. In: Proc. IEEE S&P 2021. pp. 1382–1399
17. Tan, J., Bauer, L., Christin, N., Cranor, L.F.: Practical recommendations for stronger, more usable passwords combining minimum-strength, minimum-length, and blocklist requirements. In: Proc. ACM CCS 2020. pp. 1407–1426
18. Ur, B., Kelley, P.G., Komanduri, S., Lee, J., Maass, M., Mazurek, M.L., Passaro, T., Shay, R., Vidas, T., Bauer, L., Christin, N., Cranor, L.F.: How does your password measure up? The effect of strength meters on password creation. In: Proc. USENIX SEC 2012. pp. 65–80
19. Ur, B., Segreti, S.M., Bauer, L., Christin, N., Cranor, L.F., Komanduri, S., Kurilova, D., Mazurek, M.L., Melicher, W., Shay, R.: Measuring real-world accuracies and biases in modeling password guessability. In: Proc. USENIX SEC 2015. pp. 463–481

20. Veras, R., Collins, C., Thorpe, J.: On the semantic patterns of passwords and their security impact. In: Proc. NDSS 2014. pp. 1–16
21. Wang, D., He, D., Cheng, H., Wang, P.: fuzzyPSM: A new password strength meter using fuzzy probabilistic context-free grammars. In: Proc. IEEE/IFIP DSN 2016. pp. 595–606
22. Wang, D., Wang, P.: The emperor's new password creation policies. In: Proc. ESORICS 2015, pp. 456–477
23. Wang, D., Wang, P., He, D., Tian, Y.: Birthday, name and bifacial-security: Understanding passwords of Chinese web users. In: Proc. USENIX SEC 2019. pp. 1537–1555
24. Wang, D., Zhang, Z., Wang, P., Yan, J., Huang, X.: Targeted online password guessing: An underestimated threat. In: Proc. ACM CCS 2016. pp. 1242–1254
25. Weir, M., Aggarwal, S., De Medeiros, B., Glodek, B.: Password cracking using probabilistic context-free grammars. In: Proc. IEEE S&P 2009. pp. 391–405
26. Wheeler, D.L.: zxcvbn: Low-budget password strength estimation. In: Proc. USENIX SEC 2016. pp. 157–173
27. Xie, Z., Zhang, M., Guo, Y., Li, Z., Wang, H.: Modified password guessing methods based on Targuess-I. Wirel. Commun. Mob. Comput. **2020**, 8837210:1–8837210:22 (2020)
28. Xie, Z., Zhang, M., Yin, A., Li, Z.: A new targeted password guessing model. In: Proc. ACISP 2020. LNCS, vol. 12248, pp. 350–368