Password Strength Signaling: A Counter-Intuitive Defense Against Password Cracking

Wenjie Bai[1], Jeremiah Blocki[1], and Ben Harsha[1]

Department of Computer Science,
Purdue University, West Lafayette IN, USA
{bai104,jblock,bharsha}@purdue.edu

Abstract. We introduce password strength signaling as a potential defense against password cracking. Recent breaches have exposed billions of user passwords to the dangerous threat of offline password cracking attacks. An offline attacker can quickly check millions (or sometimes billions/trillions) of password guesses by comparing a candidate password's hash value with a stolen hash from a breached authentication server. The attacker is limited only by the resources he is willing to invest. We explore the feasibility of applying ideas from Bayesian Persuasion to password authentication. Our key idea is to have the authentication server store a (noisy) signal about the strength of each user password for an offline attacker to find. Surprisingly, we show that the noise distribution for the signal can often be tuned so that a rational (profit-maximizing) attacker will crack *fewer* passwords. The signaling scheme exploits the fact that password cracking is not a zero-sum game i.e., it is possible for an attacker to increase their profit in a way that also reduces the number of cracked passwords. Thus, a well-defined signaling strategy will encourage the attacker to reduce his guessing costs by cracking fewer passwords. We use an evolutionary algorithm to compute the optimal signaling scheme for the defender. We evaluate our mechanism on several password datasets and show that it can reduce the total number of cracked passwords by up to 12% (resp. 5%) of all users in defending against offline (resp. online) attacks. While the results of our empirical analysis are positive we stress that we view the current solution as a proof-of-concept as there are important societal concerns that would need to be considered before adopting our password strength signaling solution.

Keywords: Bayesian Persuasion, Password Authentication, Stackelberg Game.

1 Introduction

In the last decade, large scale data-breaches have exposed billions of user passwords to the dangerous threat of offline password cracking. An offline attacker who has obtained the salt and cryptographic hash $((h_u, salt_u) =$

$(H(salt_u, pw_u), salt_u))$ of a user u's password (pw_u) can attempt to crack the password by comparing this hash value with the hashes of likely password guesses i.e., by checking if $h'_u = H(salt_u, pw')$ for each pw'. The attacker can check as many guesses as he wants offline — without interacting with the authentication server. The only limit is the resources that the attacker is willing to invest in trying to crack the password. A rational password cracker [9,12] will choose the number of guesses that maximizes his utility.

Password hashing serves as a last line of defense against an offline password attacker. A good password hash function H should be moderately expensive to compute so that it becomes prohibitively expensive to check millions or billions of password guesses. However, we cannot make H too expensive to compute as the honest authentication server needs to evaluate H every time a user authenticates. In this paper, we explore a highly counter-intuitive[1] defense against rational attackers which does not impact hashing costs: password strength signaling! In particular, we apply Bayesian Persuasion [30] to password authentication. Specifically, we propose to have the authentication server store a (noisy) signal sig_u which is correlated with the strength of the user's password.

Traditionally, an authentication server stores the tuple $(u, salt_u, h_u)$ for each user u where $salt_u$ is a random salt value and $h_u = H(salt_u, pw_u)$ is the salted hash. We propose to have the authentication server instead store the tuple $(u, salt_u, sig_u, h_u)$, where the (noisy) signal sig_u is sampled based on the strength of the user's password pw_u. The signal sig_u is simply recorded for an offline attacker to find if the authentication server is breached. In fact, the authentication server never even uses sig_u when the user u authenticates[2]. The attacker will only use the signal sig_u if it is beneficial — at minimum the attacker could always choose to ignore the signal.

It is natural, but incorrect, to imagine that password cracking is a zero-sum game i.e., the attacker's gain is directly proportional to the defender's loss. In a zero-sum game there would be no benefit from information signaling [59] e.g., in a zero-sum game like rock-paper-scissors there is no benefit to leaking information about your action. However, we stress that password cracking is *not* a zero-sum game. The defender's (the sender of strength signal) utility is inversely proportional to the fraction of user passwords that are cracked. By contrast, it is possible that the attacker's utility is marginal even when he cracks a password i.e., when guessing costs offset the reward. In particular, the attacker's utility is given by the (expected) value of all of the cracked passwords minus his (expected) guessing costs. Thus, it is possible that password strength signaling would persuade the attacker to crack fewer passwords to reduce guessing costs. Indeed, we show that the signal distribution can be tuned so that a rational (profit-maximizing) attacker will crack *fewer* passwords.

[1] The propose may be less counter-intuitive to those familiar with prior work in the area of Bayesian Persuasion [30].

[2] If a user u attempts to login with password pw' the authentication server will lookup $salt_u$ and h_u and accept pw' if and only if $h_u = H(salt_u, pw')$.

To provide some intuition of why password strength signaling might be beneficial, we give two examples.

Example 1 Suppose that we add a signal $sig_u = 1$ to indicate that user u's password pw_u is uncrackable (e.g., the entropy of the password is over 60-bits) and we add the signal $sig_u = 0$ otherwise. In this case, the attacker will simply choose to ignore accounts with $sig_u = 1$ to reduce his total guessing cost. However, the number of cracked user passwords stays unchanged.

Example 2 Suppose that we modify the signaling scheme above so that even when the user's password pw_u is *not* deemed to be uncrackable we still signal $sig_u = 1$ with probability ϵ and $sig_u = 0$ otherwise. If the user's password is uncrackable we always signal $sig_u = 1$. Assuming that ϵ is not too large a rational attacker might still choose to ignore any account with $sig_u = 1$ i.e., the attacker's expected reward will decrease slightly, but the attacker's guessing costs will also be reduced. In this example, the fraction of cracked user passwords is reduced by up to ϵ i.e., any lucky user u with $sig_u = 1$ will not have their password cracked.

In this work, we explore the following questions: Can password strength signaling be used to protect passwords against rational attackers? If so, how can we compute the optimal signaling strategy?

1.1 Contributions

We introduce password information signaling as a novel, counter-intuitive, defense against rational password attackers. We adapt a Stackelberg game-theoretic model of Blocki and Datta [9] to characterize the behavior of a rational password adversary and the optimal signaling strategy for an authentication server (defender). We analyze the performance of password information signaling using several large password datasets: Bfield, Brazzers, Clixsense, CSDN, Neopets, 000webhost, RockYou, Yahoo! [10, 14], and LinkedIn [8]. We analyze our mechanism both in the idealistic setting, where the defender has perfect knowledge of the user password distribution $\mathcal{P}$ and the attacker's value v for each cracked password, as well as in a more realistic setting where the defender only is given approximations of $\mathcal{P}$ and v. In our experiments, we analyze the fraction $x_{sig}(v)$ (resp. $x_{no-sig}(v)$) of passwords that a rational attacker would crack if the authentication server uses (resp. does not use) password information signaling. We find that the reduction in the number of cracked passwords can be substantial e.g., $x_{no-sig}(v) - x_{sig}(v) \approx 8\%$ under empirical distribution and 13% under Monte Carlo distribution. We also show that password strength signaling can be used to help deter online attacks when CAPTCHAs are used for throttling.

An additional advantage of our password strength signaling method is that it is independent of the password hashing method and requires no additional hashing work. Implementation involves some determination of which signal to attach to a certain account, but beyond that, any future authentication attempts are handled exactly as they were before i.e. the signal information is ignored.

We conclude by discussing several societal and ethical issues that would need to be addressed before password strength signaling is used. While password

strength signaling decreases the total number of compromised accounts, there may be a few users whose accounts are cracked *because* they were assigned an "unlucky" signal. One possible solution might be to allow users to opt-in (resp. opt-out). Another approach might try to constrain the solution space to ensure that there are no "unlucky" users.

1.2 Related Work

The human tendency to pick weaker passwords has been well documented e.g., [14]. Convincing users to select stronger passwords is a difficult task [16,28,33,47–49]. One line of research uses password strength meters to nudge users to select strong passwords [17, 32, 52] though a common finding is that users were not persuaded to select a stronger password [17, 52]. Another approach is to require users to follow stringent guidelines when they create their password. However it has been shown that these methods also suffer from usability issues [3, 24, 28, 50], and in some cases can even lead to users selecting weaker passwords [13, 33].

Offline password cracking attacks have been around for decades [38]. There is a large body of research on password cracking techniques. State of the art cracking methods employ methods like Probabilistic Context-Free Grammars [31, 55, 58], Markov models [19, 20, 36, 53], and neural networks [37]. Further work [35] has described methods of retrieving guessing numbers from commonly used tools like Hashcat [1] and John the Ripper [23].

Blocki and Datta [9] used a Stackelberg game to model the behavior of a rational (profit-motivated) attacker against a cost-asymmetric secure hashing (CASH) scheme. However, the CASH mechanism is not easily integrated with modern memory-hard functions. By contrast, password strength signaling does not require any changes to the password hashing algorithm.

A large body of research has focused on alternatives to text passwords. Alternatives have included one time passwords [25, 34, 41], challenge-response constructions [21, 29], hardware tokens [40, 46], and biometrics [4, 22, 45]. While all of these offer possible alternatives to traditional passwords it has been noted that none of these strategies outperforms passwords in all areas [15]. Furthermore, it has been noted that despite the shortcomings of passwords they remain the dominant method of authentication even today, and research should acknowledge this fact and seek to better understand traditional password use [27].

Password strength signaling is closely related to the literature on Bayesian Persuasion. Kamenica and Gentzkow [30] first introduced the notion of Bayesian Persuasion where a person (sender) chooses a signal to reveal to a receiver in an attempt to convince the receiver to take an action that positively impacts the welfare of both parties. There are a few prior results applying Bayesian Persuasion in security contexts, e.g., patrols [18], honeypots [43], with the sender (resp. receiver) playing the roles of defender (resp. attacker). To the best of our knowledge Bayesian Persuasion has never been applied in the context of password authentication. Most prior works use linear programming to find (or approximate) the sender's optimal signaling strategy. We stress that there are several unique challenges in the context of password authentication: (1) the

action space of the receiver (attacker) is exponential in the size of (the support of) the password distribution, and (2) the sender's objective function is non-linear.

2 Preliminaries

We use $\mathbb{P}$ to denote the set of all passwords that a user might select and use $\mathcal{P}$ to denote a distribution over user-selected passwords i.e., a new user will select the password $pw \in \mathbb{P}$ with probability $\Pr_{x\sim\mathcal{P}}[x = pw]$ — we typically write $\Pr[pw]$ for notational simplicity.

Password Datasets Given a set of N users $\mathcal{U} = \{u_1, \ldots, u_N\}$ the corresponding password dataset D_u is given by the multiset $D_u = \{pw_{u_1}, \ldots, pw_{u_N}\}$ where pw_{u_i} denotes the password selected by user u_i. Fixing a password dataset D we let f_i denote the number of users who selected the ith most popular password in the dataset. We note that that $f_1 \geq f_2 \geq \ldots$ and that $\sum_i f_i = N$ gives the total number N of users in the original dataset.

Empirical Password Distribution Viewing our dataset D as N independent samples from the (unknown) distribution $\mathcal{P}$, we use f_i/N as an empirical estimate of the probability of the ith most common password pw_i and $D_f = (f_1, f_2, \ldots)$ as the corresponding frequency list. In addition, $\mathcal{D}_e$ is used to denoted the corresponding empirical distribution i.e., $\Pr_{x\sim\mathcal{D}_e}[x = pw_i] = f_i/N$. Because the real distribution $\mathcal{P}$ is unknown we will typically work with the empirical distribution $\mathcal{D}_e$. We remark that when $f_i \gg 1$ the empirical estimate will be close to the actual distribution i.e., $\Pr[pw_i] \approx f_i/N$, but when f_i is small the empirical estimate will likely diverge from the true probability value. Thus, while the empirical distribution is useful to analyze the performance of password strength signaling when the password value v is small, this analysis will be less accurate for larger values of v i.e., once the rational attacker has an incentive to start cracking passwords with lower frequency.

Monte Carlo Password Distribution Following [5] we also use the Monte Carlo Password Distribution $\mathcal{D}_m$ to evaluate the performance of our password signaling mechanism when v is large. The Monte Carlo distributions is derived by subsampling passwords from our dataset D, generating guessing numbers from state-of-the-art password cracking models, and fitting a distribution to the resulting guessing curve. Due to the length limits, we omit discussion and experiment results for Monte Carlo Password Distribution. See more details in the full version of this paper [6].

3 Strength Signaling and Password Storage

In this section, we overview our basic signaling mechanism deferring until later how to optimally tune the parameters of the mechanism to minimize the number of cracked passwords.

3.1 Account Creation and Signaling

When users create their accounts they provide a user name u and password pw_u. First, the server runs canonical password storage procedure—randomly selecting a salt value $salt_u$ and calculating the hash value $h_u = H(salt_u, pw_u)$. Next, the server calculates the (estimated) strength $str_u \leftarrow \mathsf{getStrength}(pw_u)$ of password pw_u and samples the signal $sig_u \xleftarrow{\$} \mathsf{getSignal}(st_u)$. Finally, the server stores the tuple $(u, salt_u, sig_u, h_u)$ — later if the user u attempts to login with a password pw' the authentication server will accept pw' if and only if $h_u = H(salt_u, pw')$. The account creation process is formally presented in Algorithm 1.

Algorithm 1 Signaling during Account Creation

Input: u, pw_u, L, d

1: $salt_u \xleftarrow{\$} \{0,1\}^L$
2: $h_u \leftarrow H(salt_u, pw_u)$
3: $str_u \leftarrow \mathsf{getStrength}(pw_u)$
4: $sig_u \xleftarrow{\$} \mathsf{getSignal}(str_u)$
5: $\mathsf{StoreRecord}(u, salt_u, sig_u, h_u)$

A traditional password hashing solution would simply store the tuple $(u, salt_u, h_u)$ i.e., excluding the signal sig_u. Our mechanism requires two additionally subroutines $\mathsf{getStrength}()$ and $\mathsf{getSignal}()$ to generate this signal. The first algorithm is deterministic. It takes the user's password pw_u as input and outputs str_u — (an estimate of) the password strength. The second randomized algorithm takes the (estimated) strength parameter str_u and outputs a signal sig_u. The whole signaling algorithm is the composition of these two subroutines i.e., $\mathcal{A} = \mathsf{getSignal}(\mathsf{getStrength}(pw))$. We use $s_{i,j}$ to denote the probability of observing the signal $sig_u = j$ given that the estimated strength level was $str_u = i$. Thus, $\mathsf{getSignal}()$ can be encoded using a signaling matrix $\mathbf{S}$ of dimension $a \times b$,

$$\begin{bmatrix} s_{0,0} & s_{0,1} & \cdots & s_{0,b-1} \\ s_{1,0} & s_{1,1} & \cdots & s_{1,b-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{a-1,0} & s_{a-1,1} & \cdots & s_{a-1,b-1} \end{bmatrix},$$

where a is the number of strength levels that passwords can be labeled, b is the number of signals the server can generate and $\mathbf{S}[i,j] = s_{i,j}$.

We remark that if $\mathbf{S}[i,0] = 1$ for all i [3] then the actual signal sig_u is *uncorrelated* with the password pw_u. In this case our mechanism is equivalent to the traditional (salted) password storage mechanism where $\mathsf{getSignal}()$ is replaced with a constant/null function. $\mathsf{getStrength}()$ is password strength oracle that outputs the actual/estimated strength of a password. We discuss ways that $\mathsf{getStrength}()$ could be implemented in full version of this paper [6].

[3] The index of matrix elements start from 0

3.2 Generating Signals

We use $[a] = \{0, 1, \ldots, a-1\}$ (resp. $[b] = \{0, 1, \ldots, b-1\}$) to denote the range of $\mathsf{getStrength}()$ (resp. $\mathsf{getSignal}()$). For example, if $[a] = \{0, 1, 2\}$ then 0 would correspond to weak passwords, 2 would correspond to strong passwords and 1 would correspond to medium strength passwords. To generate signal for pw_u, the server first invokes subroutine $\mathsf{getStrength}(pw_u)$ to get strength level $str_u = i \in [a]$ of pw_u, then signals $sig_u = j \in [b]$ with probability $\Pr[\mathsf{getSignal}(pw_u) = j \mid \mathsf{getStrength}(pw_u) = i] = \mathbf{S}[i, j] = s_{i,j}$.

Bayesian Update. An attacker who breaks into the authentication server will be able to observe the signal sig_u and $\mathbf{S}$. After observing the signal $sig_u = y$ and $\mathbf{S}$ the attacker can perform a Bayesian update. In particular, given any password $pw \in \mathbb{P}$ with strength $i = \mathsf{getStrength}(pw)$ we have

$$\begin{aligned} \Pr[pw \mid y] &= \frac{\Pr[pw]\mathbf{S}[i, y]}{\sum_{pw' \in \mathbb{P}} \Pr[\mathsf{getSignal}(\mathsf{getStrength}(pw'))] \cdot \Pr[pw']} \\ &= \frac{\Pr[pw]\mathbf{S}[i, y]}{\sum_{i' \in [a]} \Pr_{pw' \sim \mathcal{P}}[\mathsf{getStrength}(pw') = i'] \cdot \mathbf{S}[i', y]} \end{aligned} \tag{1}$$

If the attacker knew the original password distribution $\mathcal{P}$ then s/he can update posterior distribution $\mathcal{P}_y$ with $\Pr_{x \sim \mathcal{P}_y}[x = pw] := \Pr[pw \mid y]$. We extend our notation, let $\lambda(\pi, B; y) = \sum_{i=1}^{B} \Pr[pw_i^\pi \mid y]$ where pw_i^π is the ith password in the ordering π. Intuitively, $\lambda(\pi, B; y)$ is the conditional probability of cracking the user's password by checking the first B guesses in permutation π after observing signal y.

3.3 Delayed Signaling

In some instances, the authentication server might implement the password strength oracle $\mathsf{getStrength}()$ by training a (differentially private) Count-Sketch based on the user-selected passwords $pw_u \sim \mathcal{P}$, detailed discussion about use of count-sketch in password strength signaling can be found in full version of this paper [6]. The strength estimation will not be accurate until a larger number N of users have registered. In this case, the authentication server may want to delay signaling until after the Count-Sketch has been initialized. In particular, the authentication server will store the tuple $(u, salt_u, sig_u = \perp, h_u)$ when users first register their accounts. After the count-sketch has been initialized, the server can update $sig_u = \mathsf{getSignal}(\mathsf{getStrength}(pw_u))$ upon users' next successful login.

4 Adversary Model

We adapt the economic model of [9] to capture the behavior of a rational attacker. We also make several assumptions: (1) there is a value v_u for each password pw_u that the attacker cracks; (2) the attacker is untargeted and that the value $v_u = v$

for each user $u \in U$; (3) by Kerckhoffs's principle, the password distribution $\mathcal{P}$ and the signaling matrix are known to the attacker.

Value/Cost Estimates One can derive a range of estimates for v based on black market studies e.g., Symantec reported that passwords generally sell for \$4—\$30 [26] and [51] reported that Yahoo! e-mail passwords sold for $\approx$ \$1. Similarly, we assume that the attacker pays a cost k each time he evaluates the hash function H to check a password guess. We remark that one can estimate $k \approx \$1 \times 10^{-7}$ if we use a memory-hard function [4].

4.1 Adversary Utility: No Signaling

We first discuss how a rational adversary would behave when no signal is available (traditional hashing). We defer the discussion of how the adversary would update his strategy after observing a signal y to the next section. In the no-signaling case, the attacker's strategy (π, B) is given by an ordering π over passwords $\mathbb{P}$ and a threshold B. Intuitively, this means that the attacker will check the first B guesses in π and then give up. The expected reward for the attacker is given by the simple formula $v \times \lambda(\pi, B)$, i.e., the probability that the password is cracked times the value v. Similarly, the expected guessing cost of the attacker is

$$C(k, \pi, B) = k \sum_{i=1}^{B} (1 - \lambda(\pi, i-1)), \tag{2}$$

Intuitively, $(1 - \lambda(\pi, i-1))$ denotes the probability that the adversary actually has to check the ith password guess at cost k. With probability $\lambda(\pi, i-1)$ the attacker will find the password in the first $i-1$ guesses and will not have to check the ith password guess pw_i^{π}. Specially, we define $\lambda(\pi, 0) = 0$. The adversary's expected utility is the difference of expected gain and expected cost, namely,

$$U_{adv}(v, k, \pi, B) = v \cdot \lambda(\pi, B) - C(k, \pi, B). \tag{3}$$

Sometimes we omit parameters in the parenthesis and just write U_{adv} for short when the v, k and B are clear from context.

4.2 Optimal Attacker Strategy: No Signaling

A rational adversary would choose $(\pi^*, B^*) \in \arg\max U_{adv}(v, k, \pi, B)$. It is easy to verify that the optimal ordering π^* is always to check passwords in descending order of probability. The probability that a random user's account is cracked is

$$P_{adv} = \lambda(\pi^*, B^*). \tag{4}$$

[4] The energy cost of transferring 1GB of memory between RAM and cache is approximately $0.3J$ [44], which translates to an energy cost of $\approx \$3 \times 10^{-8}$ per evaluation. Similarly, if we assume that our MHF can be evaluated in 1 second [7,11] then evaluating the hash function 6.3×10^7 times will tie up a 1GB RAM chip for 2 years. If it costs \$5 to rent a 1GB RAM chip for 2 years (equivalently purchase the RAM chip which lasts for 2 years for \$5) then the capital cost is $\approx \$8 \times 10^{-8}$. Thus, our total cost would be around $\$10^{-7}$ per password guess.

We remark that in practice $\arg\max U_{adv}(v, k, \pi, B)$ usually returns a singleton set (π^*, B^*). If instead the set contains multiple strategies then we break ties adversarially i.e.,

$$P_{adv} = \max_{(\pi^*, B^*) \in \arg\max U_{adv}(v,k,\pi,B)} \lambda(\pi^*, B^*).$$

5 Information Signaling as a Stackelberg Game

We model the interaction between the authentication server (leader) and the adversary (follower) as a two-stage Stackelberg game. In a Stackelberg game, the leader moves first and then the follower may select its action after observing the action of the leader.

In our setting the action of the defender is to commit to a signaling matrix $\mathbf{S}$ as well as the implementation of $\mathsf{getStrength}()$ which maps passwords to strength levels. The attacker responds by selecting a cracking strategy $(\vec{\pi}, \vec{B}) = \{(\pi_0, B_0), \ldots, (\pi_{b-1}, B_{b-1})\}$. Intuitively, this strategy means that whenever the attacker observes a signal y he will check the top B_y guesses according to the ordering π_y.

5.1 Attacker Utility

If the attacker checks the top B_y guesses according to the order π_y then the attacker will crack the password with probability $\lambda(\pi_y, B_y; y)$. Recall that $\lambda(\pi_y, B_y; y)$ denotes the probability of the first B_y passwords in π_y according to the posterior distribution $\mathcal{P}_y$ obtained by applying Bayes Law after observing a signal y. Extrapolating from no signal case, the expected utility of adversary conditioned on observing the signal y is

$$U_{adv}(v, k, \pi_y, B_y; \mathbf{S}, y) = v \cdot \lambda(\pi_y, B_y; y) - \sum_{i=1}^{B_y} k \cdot (1 - \lambda(\pi_y, i-1; y)), \qquad (5)$$

where B_y and π_y are now both functions of the signal y. Intuitively, $(1 - \lambda(\pi_y, i-1; y))$ denotes the probability that the attacker has to pay cost k to make the ith guess. We use $U^s_{adv}\left(v, k, \{\mathbf{S}, (\vec{\pi}, \vec{B})\}\right)$ to denote the expected utility of the adversary with password strength signaling,

$$U^s_{adv}\left(v, k, \{\mathbf{S}, (\vec{\pi}, \vec{B})\}\right) = \sum_{y \in [b]} \Pr[Sig = y] U_{adv}(v, k, \pi_y, B_y; \mathbf{S}, y)\ , \qquad (6)$$

where

$$Pr[Sig = y] = \sum_{i \in [b]} \Pr_{pw \sim \mathcal{P}}[\mathsf{getStrength}(pw) = i] \cdot S[i, y]\ .$$

5.2 Optimal Attacker Strategy

Now we discuss how to find the optimal strategy $(\vec{\pi}^*, \vec{B}^*)$. Since the attacker's strategies in reponse to different signals are independent. It suffices to find $(\pi_y^*, B_y^*) \in \arg\max_{B_y,\pi_y} U_{adv}(v, k, \pi_y, B_y; y)$ for each signal y. We first remark that the adversary can obtain the optimal checking sequence π_y^* for pw_u associated with signal y by sorting all $pw \in \mathcal{P}$ in descending order of posterior probability according to the posterior distribution $\mathcal{P}_y$.

Given the optimal guessing order π_y^*, the adversary can determine the optimal budget B_y^* for signal y such that $B_y^* = \arg\max_{B_y} U_{adv}(v, k, \pi_y^*, B_y; y)$. Each of the password distributions we analyze has a compact representation allowing us to apply techniques from [5] to further speed up the computation of the attacker's optimal strategy π_y^* and B_y^*.

We observe that an adversary who sets $\pi_y = \pi$ and $B_y = B$ for all $y \in [b]$ is effectively ignoring the signal and is equivalent to an adversary in the no signal case. Thus,

$$\max_{\vec{\pi},\vec{B}} U_{adv}^s \left(v, k, \{\mathbf{S}, (\vec{\pi}, \vec{B})\}\right) \geq \max_{\pi,B} U_{adv}(v, k, \pi, B),\ \forall \mathbf{S}, \tag{7}$$

implying that adversary's expected utility will never decrease by adapting its strategy according to the signal.

5.3 Optimal Signaling Strategy

Once the function getStrength() is fixed we want to find the optimal signaling matrix $\mathbf{S}$. We begin by introducing the defender's utility function. Intuitively, the defender wants to minimize the total number of cracked passwords.

Let $P_{adv}^s(v, k, \mathbf{S})$ denote the expected adversary success rate with password strength signaling when playing with his/her optimal strategy, then

$$P_{adv}^s(v, k, \mathbf{S}) = \sum_{y \in SL} \Pr[Sig = y] \lambda(\pi_y^*, B_y^*; \mathbf{S}, y), \tag{8}$$

where (π_y^*, B_y^*) is the optimal strategy of the adversary when receiving signal y, namely,

$$(\pi_y^*, B_y^*) = \arg\max_{\pi_y, B_y} U_{adv}(v, k, \pi_y, B_y; \mathbf{S}, y).$$

If $\arg\max_{\pi_y,B_y} U_{adv}(v, k, \pi_y, B_y; y)$ returns a set, we break ties adversarially.

The objective of the server is to minimize $P_{adv}^s(v, k, \mathbf{S})$, therefore we define

$$U_{ser}^s \left(v, k, \{\mathbf{S}, (\vec{\pi}^*, \vec{B}^*)\}\right) = -P_{adv}^s(v, k, \mathbf{S}). \tag{9}$$

Our focus of this paper is to find the optimal signaling strategy, namely, the signaling matrix $\mathbf{S}^*$ such that $\mathbf{S}^* = \arg\min_{\mathbf{S}} P_{adv}^s(v, k, \mathbf{S})$. Finding the optimal signaling matrix $\mathbf{S}^*$ is equivalent to solving the mixed strategy Subgame Perfect Equilibrium (SPE) of the Stackelberg game. At SPE no player has the incentive to derivate from his/her strategy. Namely,

$$\begin{cases} U^s_{ser}\left(v,k,\{\mathbf{S}^*,(\vec{\pi}^*,\vec{B}^*)\}\right) \geq U^s_{ser}\left(v,k,\{\mathbf{S},(\vec{\pi}^*,\vec{B}^*)\}\right), \forall \mathbf{S}, \\ U^s_{adv}\left(v,k,\{\mathbf{S}^*,(\vec{\pi}^*,\vec{B}^*)\}\right) \geq U^s_{adv}\left(v,k,\{\mathbf{S}^*,(\vec{\pi},\vec{B})\}\right), \forall (\vec{\pi},\vec{B}). \end{cases} \tag{10}$$

Notice that a signaling matrix of dimension $a \times b$ can be fully specified by $a(b-1)$ variables since the elements in each row sum up to 1. Fixing v and k, we define $f : \mathbb{R}^{a(b-1)} \to \mathbb{R}$ to be the map from $\mathbf{S}$ to $P^s_{adv}(v,k,\mathbf{S})$. Then we can formulate the optimization problem as

$$\begin{aligned} \min_{\mathbf{S}} \quad & f(s_{0,0},\dots s_{0,(b-2)},\dots,s_{(a-1),0},s_{(a-1),(b-2)}) \\ \text{s.t.} \quad & 0 \leq s_{i,j} \leq 1,\ \forall 0 \leq i \leq a-1,\ 0 \leq j \leq b-2, \\ & \sum_{j=0}^{b-2} s_{i,j} \leq 1,\ \forall 0 \leq i \leq a-1. \end{aligned} \tag{11}$$

The feasible region is a $a(b-1)$-dimensional probability simplex. Notice that in 2-D ($a = b = 2$), the second constraint would be equivalent to the first constraint. In our experiments we will treat f as a black box and use derivative-free optimization methods to find good signaling matrices $\mathbf{S}^*$.

6 Experimental Design

We now describe our empirical experiments to evaluate the performance of password strength signaling. Fixing the parameters v, k, a, b, a password distribution $\mathcal{D}$ and the strength oracle $\mathsf{getStrength}(\cdot)$ we define a procedure $\mathbf{S}^* \leftarrow \mathsf{genSigMat}(v,k,a,b,\mathcal{D})$ which uses derivate-free optimization to solve the optimization problem defined in equation (11) and find a good signaling matrix $\mathbf{S}^*$ of dimension $a \times b$. Similarly, given a signaling matrix $\mathbf{S}^*$ we define a procedure $\mathsf{evaluate}(v,k,a,b,\mathbf{S}^*,\mathcal{D})$ which returns the percentage of passwords that a rational adversary will crack given that the value of a cracked password is v, the cost of checking each password is k. To simulate settings where the defender has imperfect knowledge of the password distribution we use different distributions $\mathcal{D}_1$ (training) and $\mathcal{D}_2$ (evaluation) to generate the signaling matrix $\mathbf{S}^* \leftarrow \mathsf{genSigMat}(v,k,a,b,\mathcal{D}_1)$ and evaluate the success rate of a rational attacker $\mathsf{evaluate}(v,k,a,b,\mathbf{S}^*,\mathcal{D}_2)$. We can also set $\mathcal{D}_1 = \mathcal{D}_2$ to evaluate our mechanism under the idealized setting in which defender has perfect knowledge of the distribution.

Password Distribution We evaluate the performance of our information signaling mechanism using 9 password datasets: Bfield (0.54 million), Brazzers ($N = 0.93$ million), Clixsense (2.2 million), CSDN (6.4 million), LinkedIn (174 million), Neopets (68.3 million), RockYou (32.6 million), 000webhost (153 million) and Yahoo! (69.3 million). The Yahoo! frequency corpus ($N \approx 7 \times 10^7$) was collected and released with permission from Yahoo! using differential pri-

vacy [10] and other privacy-preserving measures [14]. All the other datasets come from server breaches.

Differentially Private Count-Sketch. When using the empirical distribution $\mathcal{D}_e$ for evaluation we evaluate the performance of an imperfect knowledge defender who trains a differentially private Count-Mean-Min-Sketch. As users register their accounts, the server can feed passwords into a Count-Mean-Min-Sketch initialized with Laplace noise to ensure differential privacy (we briefly introduce count sketch and discuss the use of it to guarantee differential privacy in the full version of this paper [6]). After the Count-Sketch has been trained, the server can query the sketch about the estimated frequency for new users' passwords. Thus we can obtain a differentially private password frequency list D^{dp}.

When working with empirical distributions in an imperfect knowledge setting we split the original dataset D in half to obtain D_1 and D_2. Our noise-initialized Count-Mean-Min-Sketch is trained with D_1. We then use this count sketch along with D_2 to extract a noisy distribution $\mathcal{D}_{train}$. In particular, for every $pw \in D_2$ we query the the count sketch to get $\tilde{f}_{pw}$, a noisy estimate of the frequency of pw in D_2 and set $\Pr_{\mathcal{D}_{train}}[pw] \doteq \frac{\tilde{f}_{pw}}{\sum_{w \in D_2} \tilde{f}_w}$. We also use the Count-Mean-Min Sketch as a frequency oracle in our implementation of $\mathsf{getStrength}()$. $\mathcal{D}_{train}$ is used to derive frequency thresholds for $\mathsf{getStrength}()$ and to generate the signaling matrix $\mathbf{S}^* \leftarrow \mathsf{genSigMat}(v, k, a, b, \mathcal{D}_{train})$. Finally we evaluate results on the original empirical distribution $\mathcal{D}_e$ for the original dataset D i.e., $P^s_{adv} \leftarrow \mathsf{evaluate}(v, k, a, b, \mathbf{S}^*, \mathcal{D}_e)$.

Derivative-Free Optimization. Given a value v and hash cost k we want to find a signaling matrix which optimizes the defenders utility. Recall that this is equivalent to minimizing the function $f(\mathbf{S}) = \mathsf{evaluate}(v, k, a, b, \mathbf{S}, \mathcal{D})$ subject to the constraints that $\mathbf{S}$ is a valid signaling matrix.

In experiment we will treat f as a black box and use BITmask Evolution OPTimization [54] (BITEOPT) with 10^4 iterations to generate signaling matrix $\mathbf{S}^*$ for each different v/C_{max} ratio, where C_{max} is server's maximum authentication cost satisfying $k \leq C_{max}$.

7 Empirical Analysis

We describe the results of our experiments. In the first batch of experiments, we evaluate the performance of password strength signaling against an offline and an online attacker where the ratio v/C_{max} is typically much smaller.

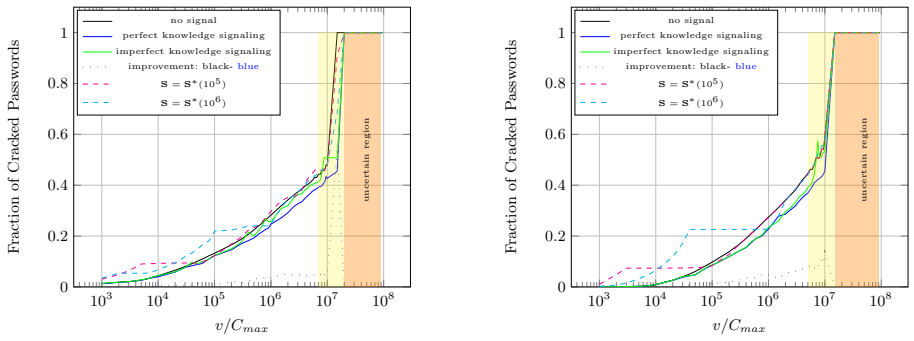
7.1 Password Strength Signaling against Offline Attacks

We consider four scenarios using the empirical/Monte Carlo distribution in a setting where the defender has perfect/imperfect knowledge of the distribution.

Empirical Distribution From each password dataset we derived an empirical distribution $\mathcal{D}_e$ and set $\mathcal{D}_{eval} = \mathcal{D}_e$. In the perfect knowledge setting we also set $\mathcal{D}_{train} = \mathcal{D}_e$ while in the imperfect knowledge setting we used a Count-Min-Mean Sketch to derive $\mathcal{D}_{train}$ (see details in the previous section).

We fix dimension of signaling matrix to be 11 by 3 (the server issues 3 signals for 11 password strength levels) and compute attacker's success rate for different value-to-cost ratios $v/C_{max} \in \{i \times 10^j : 1 \le i \le 9,\ 3 \le j \le 7\} \cup \{(i+0.5) \times 10^j : 1 \le i \le 9,\ 6 \le j \le 7\}$. In particular, for each value-to-cost ratio v/C_{max} we run $\mathbf{S}^* \leftarrow \mathsf{genSigMat}(v, k, a, b, \mathcal{D}_e)$ to generate a signaling matrix and then run $\mathsf{evaluate}(v, k, a, b, \mathbf{S}^*, \mathcal{D}_e)$ to get the attacker's success rate. The same experiment is repeated for all 9 password datasets. We plot the attacker's success rate vs. v/C_{max} in Fig. 1. Due to space limitations Fig. 1 only shows results for 2 datasets — additional plots can be found in full version of this paper [6].

We follow the approach of [5], highlighting the uncertain regions of the plot where the cumulative density function of the empirical distribution might diverge from the real distribution. In particular, the red (resp. yellow) region indicates $E > 0.1$ (resp. $E > 0.01$) where E can be interpreted as an upper bound on the difference between the two CDFs.



(a) Yahoo (b) Neopets

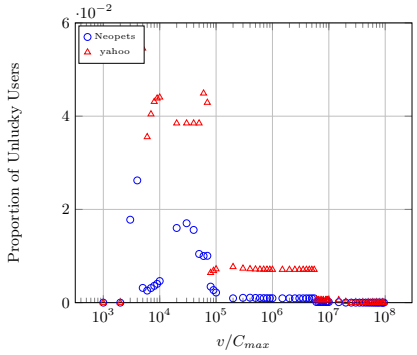
Fig. 1. Adversary Success Rate vs v/C_{max} for Empirical Distributions
the red (resp. yellow) shaded areas denote unconfident regions where the the empirical distribution might diverge from the real distribution $E \ge 0.1$ (resp. $E \ge 0.01$).

Fig. 1 demonstrates that information signaling reduces the fraction of cracked passwords. The mechanism performs best when the defender has perfect knowledge of the distribution (blue curve), but even with imperfect knowledge, there is still a large advantage. For example, for the Neopets dataset when $v/C_{max} = 5 \times 10^6$ the percentage of cracked passwords is reduced from 44.6% to 36.9% (resp. 39.1%) when the defender has perfect (resp. imperfect) knowledge of the password distribution. Similar results hold for other datasets. The green curve (signaling with imperfect knowledge) curve generally lies in between the black curve (no signaling) and the blue curve (signaling with perfect knowledge), but sometimes has an adverse effect when v/C_{max} is large. This is because the noisy

distribution will be less accurate for stronger passwords that were sampled only once.

We also use guessing numbers generated by state-of-the-art password cracking models (neural network, Markov model, PCFG) to fit a distribution, which we call Monte Carlo distribution. Monte Carlo distributions are useful in evaluating the performance of strength signaling when v/C_{max} is large. Experiments show that the reduction in the percentage of cracked passwords is up to 12% for Neopets, results can be found in the full version of this paper [6].

Which accounts are cracked? As Fig 1 demonstrates password strength signaling can substantially reduce the overall fraction of cracked passwords i.e., many previously cracked passwords are now protected. It is natural to ask whether there are any unlucky users u whose password is cracked after signaling *even though* their account was safe before signaling. Let X_u (resp. L_u) denote the event that user u is unlucky (resp. lucky) i.e., a rational attacker would originally not crack pw_u, but after password strength signaling the account is cracked. We measure $E[X_u]$ and $E[L_u]$ (See Fig. 2) for various v/C_{max} values under each dataset. Generally, we find that the fraction of unlucky users $E[X_u]$ is small in most cases e.g. ≤ 0.04. For example, when $v/k = 2 \times 10^7$ we have that $E[X_u] \approx 0.09\%$ and $E[L_u] \approx 5\%$ for Neopets. In all instances the net advantage $E[L_u] - E[X_u]$ remains positive.



(a) Yahoo!, Neopets

Fig. 2. Proportion of Unlucky Users for Various Datasets ($E[X_u]$)

We remark that the reduction in cracked passwords does not necessarily come from persuading the attacker to crack weak passwords, but rather through the attacker shifting his attention towards certain signals. A utility-maximizing attacker will be interested in passwords whose signals suggest the attacker will not need to spend as much effort to crack them. However, because of the noisy nature of the signaling scheme, this is only similar to, but not quite the same, as attacking only the weakest passwords in a set. Some weak passwords may be "saved" when they are signaled as being in a higher strength category than their

true strength merits. By contrast, without signaling we expect that a rational attacker will crack all of the weak passwords.

Robustness We also evaluated the robustness of the signaling matrix when the defender's estimate of the ratio v/C_{max} is inaccurate. In particular, for each dataset we generated the signaling matrix $\mathbf{S}(10^5)$ (resp. $\mathbf{S}(10^6)$) which was optimized with respect to the ratio $v/C_{max} = 10^5$ (resp. $v/C_{max} = 10^6$) and evaluated the performance of both signaling matrices against an attacker with different v/C_{max} ratios. We find that password signaling is tolerant even if our estimate of v/k is off by a small multiplicative constant factor e.g., 2. For example, in Fig. 1b the signaling matrix $\mathbf{S}(10^6)$ outperforms the no-signaling case even when the real v/C_{max} ratio is as large as 2×10^6. In the "downhill" direction, even if the estimation of v/k deviates from its true value up to 5×10^5 at anchor point 10^6 it is still advantageous for the server to deploy password signaling.

7.2 Password Strength Signaling against Online Attacks

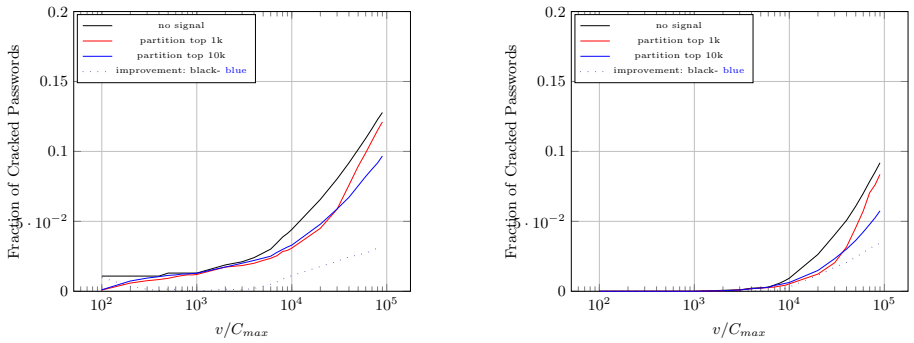
We can extend the experiment from password signaling with perfect knowledge to an online attack scenario. One common way to throttle online attackers is to require the attacker to solve a CAPTCHA challenge [56], or provide some other proof of work (PoW), after each incorrect login attempt [42]. One advantage of this approach is that a malicious attacker cannot lockout an honest user by repeatedly submitting incorrect passwords [2]. However, the solution also allows an attacker to continue trying to crack the password as long as s/he is willing to continue paying the cost to solve the CAPTCHA/PoW challenges. Thus, password strength signaling could be a useful tool to mitigate the risk of online attacks.

When modeling a rational online password we will assume that $v/C_{max} \leq 10^5$ since the cost to pay a human to solve a CAPTCHA challenge (e.g., $\$10^{-3}$ to 10^2 [39]) is typically much larger than the cost to evaluate a memory-hard cryptographic hash function (e.g., $\$10^{-7}$). Since $v/C_{max} \leq 10^5$ we use the empirical distribution to evaluate the performance of signaling against an online attacker. In the previous subsection, we found that the uncertain regions of the curve started when $v/C_{max} \gg 10^5$ so the empirical distribution is guaranteed to closely match the real one.

Since an online attacker will be primarily focused on the most common passwords (e.g., top 10^3 to 10^4) we modify getStrength() accordingly. We consider two modifications of getStrength() which split passwords in the top 10^3 (resp. 10^4) passwords into 11 strength levels. By contrast, our prior implementation of getStrength() would have placed most of the top 10^3 passwords in the bottom two strength levels. As before we fix the signaling matrix dimension to be 11×3. Our results are shown in Fig. 3. Plots for other datasets can be found in the full version of this paper [6].

Our results demonstrate that password strength signaling can be an effective defense against online attackers as well. For example, in Fig. 3a, when $v/C_{max} =$

9×10^4, our mechanism reduces the fraction of cracked passwords from 12.7% to just 9.6%. Similarly, observations hold true for other datasets. We observe that the red curve (partitioning the top 10^3 passwords into 11 strength levels) performs better than the blue curve (partitioning the top 10^3 passwords into 11 strength levels) when v/k is small e.g., $v/C_{max} < 2 \times 10^4$ in Fig. 3a). The blue curve performs better when v/C_{max} is larger. Intuitively, this is because we want to have a fine-grained partition for the weaker (top 10^3) passwords that the adversary might target when v/C_{max} is small.



(a) Yahoo (b) Neopets

Fig. 3. Adversary Success Rate vs v/C_{max} in Defense of Online Attacks

7.3 Discussion

- While password strength signaling reduced the total number of cracked passwords a few unlucky users might be harmed i.e., instead of being deterred the unlucky signal helps the rational attacker to crack a password that they would not otherwise have cracked. The usage of password signaling raises important ethical and societal questions. How would users react to such a solution knowing that they could be one of the unlucky users? One possible way to address these concerns would be to allow users to opt-in/out of password strength signaling. However, each user u would need to make this decision without observing their signal. Otherwise, the decision to opt-in/out might be strongly correlated with the signal allowing the attacker to perform another Bayesian update. Another possible way to address these concerns would be to modify the objective function (equation (11)) to penalize solutions with unlucky users.
- Can we analyze the behavior of rational targeted attackers? We only consider an untargeted attacker. In some settings, an attacker might place a higher value on some passwords e.g., celebrity accounts. Can we predict how a targeted attacker would behave if the value v_u varied from user to user? Similarly, a targeted adversary could exploit demographic and/or biographical knowledge to improve password guessing attacks e.g., see [57].

8 Conclusions

We introduce password strength signaling as a novel, yet counter-intuitive defense against rational password attackers. We use Stackelberg game to model the interaction between the defender and attacker, and present an algorithm for the server to optimize its signaling matrix. We ran experiments to empirically evaluate the effectiveness of password strength signaling on 9 password datasets. When testing on the empirical (resp. Monte Carlo) password distribution distribution we find that password strength signaling reduces the number of passwords that would have been cracked by up to 8% (resp. 12%). Additionally, we find that password strength signaling can help to dissuade an online attacker by saving 5% of all user accounts. We view our positive experimental results as a proof of concept which motivates further exploration of password strength signaling.

Acknowledgement

This work was supported the National Science Foundation under NSF CAREER Award CNS-2047272 and under NSF Award 1755708 and by Rolls-Royce through a Doctoral Fellowship.

References

1. Hashcast: advanced password recovery, `https://hashcat.net/hashcat/`
2. Hackers find new way to bilk eBay users - CNET (2019), `https://www.cnet.com/news/hackers-find-new-way-to-bilk-ebay-users/`
3. Adams, A., Sasse, M.A.: Users are not the enemy. Communications of the ACM **42**(12), 40–46 (1999)
4. Aleksic, P.S., Katsaggelos, A.K.: Audio-visual biometrics. Proceedings of the IEEE **94**(11), 2025–2044 (2006)
5. Bai, W., Blocki, J.: Dahash: Distribution aware tuning of password hashing costs. In: Financial Cryptography and Data Security. Springer International Publishing (2021)
6. Bai, W., Blocki, J., Harsha, B.: Password strength signaling: A counter-intuitive defense against password cracking (2021)
7. Biryukov, A., Dinu, D., Khovratovich, D.: Argon2: new generation of memory-hard functions for password hashing and other applications. In: Security and Privacy (EuroS&P), 2016 IEEE European Symposium on. pp. 292–302. IEEE (2016)
8. Blocki, J., Harsha, B.: Linkedin password frequency corpus (2019), `\url{https://figshare.com/articles/linkedin\_files\_zip/7350287}`
9. Blocki, J., Datta, A.: CASH: A cost asymmetric secure hash algorithm for optimal password protection. In: IEEE 29th Computer Security Foundations Symposium. pp. 371–386 (2016)
10. Blocki, J., Datta, A., Bonneau, J.: Differentially private password frequency lists. In: NDSS 2016. The Internet Society (Feb 2016)
11. Blocki, J., Harsha, B., Kang, S., Lee, S., Xing, L., Zhou, S.: Data-independent memory hard functions: New attacks and stronger constructions. Cryptology ePrint Archive, Report 2018/944 (2018), `https://eprint.iacr.org/2018/944`

12. Blocki, J., Harsha, B., Zhou, S.: On the economics of offline password cracking. In: 2018 IEEE Symposium on Security and Privacy. pp. 853–871. IEEE Computer Society Press (May 2018). https://doi.org/10.1109/SP.2018.00009
13. Blocki, J., Komanduri, S., Procaccia, A., Sheffet, O.: Optimizing password composition policies. In: Proceedings of the fourteenth ACM conference on Electronic commerce. pp. 105–122. ACM (2013)
14. Bonneau, J.: The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In: 2012 IEEE Symposium on Security and Privacy. pp. 538–552. IEEE Computer Society Press (May 2012). https://doi.org/10.1109/SP.2012.49
15. Bonneau, J., Herley, C., van Oorschot, P.C., Stajano, F.: The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In: 2012 IEEE Symposium on Security and Privacy. pp. 553–567. IEEE Computer Society Press (May 2012). https://doi.org/10.1109/SP.2012.44
16. Campbell, J., Ma, W., Kleeman, D.: Impact of restrictive composition policy on user password choices. Behaviour & Information Technology **30**(3), 379–388 (2011)
17. Carnavalet, X., Mannan, M.: From very weak to very strong: Analyzing password-strength meters. In: NDSS 2014. The Internet Society (Feb 2014)
18. Carroll, T.E., Grosu, D.: A game theoretic investigation of deception in network security. In: 2009 Proceedings of 18th International Conference on Computer Communications and Networks. pp. 1–6 (2009). https://doi.org/10.1109/ICCCN.2009.5235344
19. Castelluccia, C., Chaabane, A., Dürmuth, M., Perito, D.: When privacy meets security: Leveraging personal information for password cracking. arXiv preprint arXiv:1304.6584 (2013)
20. Castelluccia, C., Dürmuth, M., Perito, D.: Adaptive password-strength meters from Markov models. In: NDSS 2012. The Internet Society (Feb 2012)
21. Chiasson, S., van Oorschot, P.C., Biddle, R.: Graphical password authentication using cued click points. In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 359–374. Springer, Heidelberg (Sep 2007). https://doi.org/10.1007/978-3-540-74835-9_24
22. Daugman, J.: How iris recognition works. In: The essential guide to image processing, pp. 715–739. Elsevier (2009)
23. Designer, S.: John the ripper password cracker (2006)
24. Florêncio, D., Herley, C., Van Oorschot, P.C.: An administrator's guide to Internet password research. In: Proceedings of the 28th USENIX Conference on Large Installation System Administration. pp. 35–52. LISA'14 (2014)
25. Florêncio, D.A.F., Herley, C.: One-time password access to any server without changing the server. In: Wu, T.C., Lei, C.L., Rijmen, V., Lee, D.T. (eds.) ISC 2008. LNCS, vol. 5222, pp. 401–420. Springer, Heidelberg (Sep 2008)
26. Fossi, M., Johnson, E., Turner, D., Mack, T., Blackbird, J., McKinney, D., Low, M.K., Adams, T., Laucht, M.P., Gough, J.: Symantec report on the underground economy (November 2008), retrieved 1/8/2013.
27. Herley, C., Van Oorschot, P.: A research agenda acknowledging the persistence of passwords. IEEE Security & Privacy **10**(1), 28–36 (2011)
28. Inglesant, P.G., Sasse, M.A.: The true cost of unusable password policies: Password use in the wild. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 383–392. CHI '10, ACM, New York, NY, USA (2010). https://doi.org/10.1145/1753326.1753384, http://doi.acm.org/10.1145/1753326.1753384

29. Jhawar, R., Inglesant, P., Courtois, N., Sasse, M.A.: Make mine a quadruple: Strengthening the security of graphical one-time pin authentication. In: 2011 5th International Conference on Network and System Security. pp. 81–88. IEEE (2011)
30. Kamenica, E., Gentzkow, M.: Bayesian persuasion. American Economic Review **101**(6), 2590–2615 (October 2011). https://doi.org/10.1257/aer.101.6.2590, https://www.aeaweb.org/articles?id=10.1257/aer.101.6.2590
31. Kelley, P.G., Komanduri, S., Mazurek, M.L., Shay, R., Vidas, T., Bauer, L., Christin, N., Cranor, L.F., Lopez, J.: Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In: 2012 IEEE Symposium on Security and Privacy. pp. 523–537. IEEE Computer Society Press (May 2012). https://doi.org/10.1109/SP.2012.38
32. Komanduri, S., Shay, R., Cranor, L.F., Herley, C., Schechter, S.: Telepathwords: Preventing weak passwords by reading users' minds. In: 23rd USENIX Security Symposium (USENIX Security 14). pp. 591–606. USENIX Association, San Diego, CA (Aug 2014), https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/komanduri
33. Komanduri, S., Shay, R., Kelley, P.G., Mazurek, M.L., Bauer, L., Christin, N., Cranor, L.F., Egelman, S.: Of passwords and people: measuring the effect of password-composition policies. In: CHI. pp. 2595–2604 (2011), http://dl.acm.org/citation.cfm?id=1979321
34. Kuhn, M.: Otpw—a one-time password login package (1998)
35. Liu, E., Nakanishi, A., Golla, M., Cash, D., Ur, B.: Reasoning analytically about password-cracking software. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 380–397. IEEE (2019)
36. Ma, J., Yang, W., Luo, M., Li, N.: A study of probabilistic password models. In: 2014 IEEE Symposium on Security and Privacy. pp. 689–704. IEEE Computer Society Press (May 2014). https://doi.org/10.1109/SP.2014.50
37. 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: Holz, T., Savage, S. (eds.) USENIX Security 2016. pp. 175–191. USENIX Association (Aug 2016)
38. Morris, R., Thompson, K.: Password security: A case history. Communications of the ACM **22**(11), 594–597 (1979)
39. Motoyama, M., Levchenko, K., Kanich, C., McCoy, D., Voelker, G.M., Savage, S.: Re: CAPTCHAs-understanding CAPTCHA-solving services in an economic context. In: USENIX Security 2010. pp. 435–462. USENIX Association (Aug 2010)
40. Parno, B., Kuo, C., Perrig, A.: Phoolproof phishing prevention. In: Di Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 1–19. Springer, Heidelberg (Feb / Mar 2006)
41. Pashalidis, A., Mitchell, C.J.: Impostor: A single sign-on system for use from untrusted devices. In: IEEE Global Telecommunications Conference, 2004. GLOBECOM'04. vol. 4, pp. 2191–2195. IEEE (2004)
42. Pinkas, B., Sander, T.: Securing passwords against dictionary attacks. In: Atluri, V. (ed.) ACM CCS 2002. pp. 161–170. ACM Press (Nov 2002). https://doi.org/10.1145/586110.586133
43. Rabinovich, Z., Jiang, A.X., Jain, M., Xu, H.: Information disclosure as a means to security. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. p. 645–653. AAMAS '15, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2015)

44. Ren, L., Devadas, S.: Bandwidth hard functions for ASIC resistance. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 466–492. Springer, Heidelberg (Nov 2017). https://doi.org/10.1007/978-3-319-70500-2_16
45. Ross, A., Shah, J., Jain, A.K.: From template to image: Reconstructing fingerprints from minutiae points. IEEE transactions on pattern analysis and machine intelligence **29**(4), 544–560 (2007)
46. RSA: Rsa securid® 6100 usb token (2003)
47. Shay, R., Komanduri, S., Durity, A.L., Huh, P.S., Mazurek, M.L., Segreti, S.M., Ur, B., Bauer, L., Christin, N., Cranor, L.F.: Can long passwords be secure and usable? In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2927–2936. CHI '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2556288.2557377, http://doi.acm.org/10.1145/2556288.2557377
48. Shay, R., Komanduri, S., Kelley, P.G., Leon, P.G., Mazurek, M.L., Bauer, L., Christin, N., Cranor, L.F.: Encountering stronger password requirements: user attitudes and behaviors. In: Proceedings of the Sixth Symposium on Usable Privacy and Security. pp. 2:1–2:20. SOUPS '10, ACM, New York, NY, USA (2010). https://doi.org/10.1145/1837110.1837113, http://doi.acm.org/10.1145/1837110.1837113
49. Stanton, J.M., Stam, K.R., Mastrangelo, P., Jolton, J.: Analysis of end user security behaviors. Comput. Secur. **24**(2), 124–133 (Mar 2005)
50. Steves, M., Chisnell, D., Sasse, A., Krol, K., Theofanos, M., Wald, H.: Report: Authentication diary study. Tech. Rep. NISTIR 7983, National Institute of Standards and Technology (NIST) (2014)
51. Stockley, M.: What your hacked account is worth on the dark web (Aug 2016), https://nakedsecurity.sophos.com/2016/08/09/what-your-hacked-account-is-worth-on-the-dark-web/
52. Ur, B., Kelley, P.G., Komanduri, S., Lee, J., Maass, M., Mazurek, M., 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: Proceedings of USENIX Security Symposium (2012)
53. 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: Jung, J., Holz, T. (eds.) USENIX Security 2015. pp. 463–481. USENIX Association (Aug 2015)
54. Vaneev, A.: BITEOPT - Derivative-free optimization method. Available at https://github.com/avaneev/biteopt (2021), c++ source code, with description and examples
55. Veras, R., Collins, C., Thorpe, J.: On semantic patterns of passwords and their security impact. In: NDSS 2014. The Internet Society (Feb 2014)
56. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: CAPTCHA: Using hard AI problems for security. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 294–311. Springer, Heidelberg (May 2003). https://doi.org/10.1007/3-540-39200-9_18
57. Wang, D., Zhang, Z., Wang, P., Yan, J., Huang, X.: Targeted online password guessing: An underestimated threat. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 2016. pp. 1242–1254. ACM Press (Oct 2016). https://doi.org/10.1145/2976749.2978339
58. Weir, M., Aggarwal, S., de Medeiros, B., Glodek, B.: Password cracking using probabilistic context-free grammars. In: 2009 IEEE Symposium on Security and

Privacy. pp. 391–405. IEEE Computer Society Press (May 2009). https://doi.org/10.1109/SP.2009.8
59. Xu, H., Freeman, R.: Signaling in bayesian stackelberg games. In: Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (2016)