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

Wenjie Bai
Dept. of Computer Science
Purdue University
West Lafayette, USA
bai104@purdue.edu

Jeremiah Blocki
Dept. of Computer Science
Purdue University
West Lafayette, USA
jblocki@purdue.edu

Ben Harsha
Dept. of Computer Science
Purdue University
West Lafayette, USA
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 their hash value with the 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., the attacker's profit is given by the value of the cracked passwords *minus* the total guessing cost. 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.**

***Index Terms*—Bayesian Persuasion, Password Authentication, Stackelberg Game**

I. 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 (salted) cryptographic hash ($h_u = H(salt_u, pw_u)$) 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 [1], [2] 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 [3] 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 [4] 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

[1]The propose may be less counter-intuitive to those familiar with prior work in the area of Bayesian Persuasion [3].

[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 $h_u = H(salt_u, pw')$.

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.

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

a) 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.

b) 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 information signaling be used to protect passwords against rational attackers? If so, how can we compute the optimal signaling strategy?

A. 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 [1] 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! [5], [6], and LinkedIn [7]. 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 information signaling can be used to help deter online attacks when CAPTCHAs are used for throttling.

An additional advantage of our information 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.

II. Related Work

The human tendency to pick weaker passwords has been well documented e.g., [5]. Convincing users to select stronger passwords is a difficult task [8]–[13]. One line of research uses password strength meters to nudge users to select strong passwords [14]–[16] though a common finding is that users were not persuaded to select a stronger password [15], [16]. 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 [12], [17]–[19], and in some cases can even lead to users selecting weaker passwords [9], [20].

Offline password cracking attacks have been around for decades [21]. There is a large body of research on password cracking techniques. State of the art cracking methods employ methods like Probabilistic Context-Free Grammars [22]–[24], Markov models [25]–[28], and neural networks [29]. Further work [30] has described methods of retrieving guessing numbers from commonly used tools like Hashcat [31] and John the Ripper [32].

A good password hashing algorithm should be moderately expensive so that it is prohibitively expensive for an offline attacker to check billions of password guesses. Password BCRYPT [33] or PBKDF2 [34] attempt to increase guessing this by iterating the hash function some number of times. However, Blocki et al. [2] argued that hash iteration cannot adequately deter an offline attacker due to the existence of sophisticated ASICs (e.g., Bitcoin miners) which can compute the underling hash function trillions of times per second. Instead, they advocate for the use of Memory Hard Functions (MHF) for password hashing.

MHFs at their core require some large amount of memory to compute in addition to longer computation times. Candidate MHFs include SCRYPT [35], Balloon hashing [36], and Argon2 [37] (the winner of the Password Hashing Competition [38]). MHFs can be split into two distinct categories or modes of operation - data-independent MHFs (iMHFs) and data-dependent MHFs(dMHFs) (along with the hybrid idMHF, which runs in both modes). dMHFs like SCRYPT are maximally memory hard [39], although they have the issue of possible side-channel attacks. Closely related to the notion of memory hardness is that of depth-robustness - a property of directed acyclic graphs (DAG). Alwen and Blocki showed that a depth robust DAG is both necessary [40] and sufficient [41]

to construct a data-independent memory-hard function. Recent work has proposed candidate iMHF constructions that show resistance to currently-known attacks [42]. Harsha and Blocki introduced a memory-hard KDF which accepts the input passwords as a stream so that the hashing algorithm can perform extra computation while the user is typing the password [43].

Blocki and Datta [1] 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, information signaling does not require any changes to the password hashing algorithm.

Distributed hashing methods (e.g. [44]–[47]) offer a method to distribute storage and/or computation over multiple servers. Thus, an attacker who only breaches one server would not be able to mount an offline attack. Juels and Rivest proposed the inclusion of several false entries per user, with authentication attempts checked against an independent server to see if the correct entry was selected [48]. These "Honeyword" passwords serve as an alarm that an offline cracking attack is being attempted. Other methods of slowing down attackers include requiring some hard (for computers) problem to be solved after several failed authentication attempts (e.g. by using a CAPTCHA) [49]–[51]. An orthogonal line of research aims to protect users against online guessing attacks [52], [53].

A large body of research has focused on alternatives to text passwords. Alternatives have included one time passwords [54]–[56], challenge-response constructions [57], [58], hardware tokens [59], [60], and biometrics [61]–[63]. While all of these offer possible alternatives to traditional passwords it has been noted that none of these strategies outperforms passwords in all areas [64]. 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 [65].

Password strength signaling is closely related to the literature on Bayesian Persuasion. Kamenica and Gentzkow [3] 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. Alonso and Camara [66] studied the (sufficient) conditions under which a sender can benefit from persuasion. Dughmi et. al [67] and Hoefer et. al [68] study Bayesian Persuasion from an algorithmic standpoint in different contexts. There are a few prior results applying Bayesian Persuasion in security contexts, e.g., patrols [69], honeypots [70], 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. In the most general case it is computationally intractable to compute the sender's optimal strategy [67]. 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.

III. Preliminaries

A. Password Representation

We use $\mathbb{P}$ to denote the set of all passwords that a user might select and we 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.

1) 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.

2) 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 information 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 incentive to start cracking passwords with lower frequency.

3) Monte Carlo Password Distribution: Following [71] 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. See more details in section VIII.

4) Password Equivalence Set: It is often convenient to group passwords having (approxmiately) equal probability into an *equivalence set* es. Suppose there are N' equivalence sets, we typically have $N' \ll N$. Thus, an algorithm whose running time scales with n' is much faster than an algorithm whose running time scales with N, see Appendix A.

B. Differential Privacy and Count Sketches

As part of our information signaling, we need a way for the authentication server to estimate the strength of each user's passwords. We propose to do this with a (differentially private)

Count-Sketch data structure, which allows us to approximately determine how many users have selected each particular password. As a side-benefit the authentication server could also use the Count-Sketch data structure to identify/ban overly popular passwords [72] and to defend against online guessing attacks [52], [53]. We first introduce the notion of differential privacy.

1) ϵ-Differential Privacy: ϵ-Differential Privacy [73] is a mechanism that provides strong information-theoretic privacy guarantees for all individuals in a dataset. Formally, an algorithm $\mathcal{A}$ preserves ϵ-differential privacy iff for all datasets D and D' that differ by only one element and all subsets S of Range($\mathcal{A}$):

$$\Pr\left[\mathcal{A}(D) \in S\right] \leq \mathrm{e}^{\epsilon} \Pr\left[\mathcal{A}(D') \in S\right].$$

In our context, we can think of D (resp. D') as a password dataset which does (resp. does not) include our user u's password pw_u and we can think of $\mathcal{A}$ as a randomized algorithm that outputs a noisy count-sketch algorithm. Intuitively, differential privacy guarantees that an attacker cannot even tell if pw_u was included when the count-sketch was generated. In particular, (up to a small multiplicative factor e^{ϵ}) the attacker cannot tell the difference between $\mathcal{A}(D)$ and $\mathcal{A}(D')$ the count-sketch we sample when pw_u was (resp. was not) included. Thus, whatever the attacker hopes to know about u's from $\mathcal{A}(D)$ the attacker could have learned from $\mathcal{A}(D')$.

2) Count-sketch: A count sketch over some domain E is a probabilistic data structure that stores some information about the frequency of items seen in a stream of data — in our password context we will use the domain $E = \mathbb{P}$. A count-sketch functions as a table T with width w_s columns and depth d_s rows. Initially, $T[i,j] = 0$ for all $i \leq w_s$ and $j \leq d_s$. Each row is associated with a hash function $H_i : \mathbb{P} \to [w_s]$, with each of the hash functions used in the sketch being pairwise independent.

To insert an element $pw \in \mathbb{P}$ into the count sketch we update $T[i, H_i(pw)] \leftarrow T[i, H_i(pw)] + 1$ for each $i \leq d_s$ [3]. To estimate the frequency of pw we would output $f\left(T[1, H_1(pw)], \ldots, T[d_s, H_{d_s}(pw)]\right)$ for some function $f : \mathbb{N}^{d_s} \to \mathbb{N}$. In our experiments we instantiate a Count-Mean-Min Sketch where $f = \texttt{median}\left\{T[i, H_i(pw)] - \frac{\#total - T[i,H_i(pw)]}{d_w - 1} : i = 1, \ldots, d_s\right\}$ ($\#total$ is the total number of elements being inserted) so that bias is subtracted from overall estimate. Other options are available too, e.g., $f = \texttt{min}$ (Count-Min), $f = \texttt{mean}$ (Count-Mean-Sketch) and $f = \texttt{median}$ (Count-Median) [4].

Oserve that adding a password only alters the value of $T[i,j]$ at d_s locations. Thus, to preserve ϵ-differential privacy we can initialize each cell $T[i,j]$ by adding Laplace noise with scaling parameter d_s/ϵ [74].

[3]In some instantiations of count sketch we would instead set $T[i, H_i(pw)] \leftarrow T[i, H_i(pw)] + G_i(pw)$ where the hash function $G_i : \mathbb{P} \to \{-1, 1\}$

[4]Count-Median Sketch uses a different insersion method

C. Other Notation

Given a permutation π over all allowable passwords $\mathbb{P}$ we let $\lambda(\pi, B) := \sum_{i=1}^{B} \Pr\left[pw_i^{\pi}\right]$ denote the probability that a randomly sampled password $pw \in \mathbb{P}$ would be cracked by an attacker who checks the first B guesses according to the order π — here pw_i^{π} is the ith password in the sequence π. Given an randomized algorithm $\mathcal{A}$ and a random string r we use $y \leftarrow \mathcal{A}(x; r)$ to denote the output when we run $\mathcal{A}$ with input x fixing the outcome of the random coins to be r. We use $y \xleftarrow{\$} \mathcal{A}(x)$ to denote a random sample drawn by sampling the random coins r uniformly at random. Given a randomized (signaling) algorithm $\mathcal{A} : \mathbb{P} \to [0, b-1]$ (where b is the total number of signals) we define the conditional probability $\Pr[pw \mid y] := \Pr_{x \sim \mathcal{P}, r}[x = pw \mid y = \mathcal{A}(pw)]$ and

$$\lambda(\pi, B; y) := \sum_{i=1}^{B} \Pr[pw_i^{\pi} \mid y] \ .$$

We remark that $\Pr[pw \mid y]$ can be evaluated using Bayes Law given knowledge of the signaling algorithm $\mathcal{A}(x)$.

IV. Information 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.

A. Account Creation and Signaling

When users create their accounts they provide a user name u and password pw_u. First, the server runs the 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, getSignal() can be encoded using a signaling matrix $\mathbf{S}$ of dimension $a \times b$, i.e.,

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

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 for some signaling matrices (e.g., if $\mathbf{S}[i,0] = 1$ for all i [5]) 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 getSignal() is replaced with a constant/null function. getStrength() is password strength oracle that outputs the actual/estimated strength of a password. We discuss ways that getStrength() could be implemented in Section VIII. For now, we omit the implementation details of strength oracle getStrength() for sake of readability.

B. Generating Signals

We use $[a] = 0, 1, \ldots, a-1$ (resp. $[b] = 0, 1, \ldots, b-1$) to denote the range of getStrength() (resp. 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 $\Pr[\mathsf{getSignal}(pw_u) = j \mid \mathsf{getStrength}(pw_u) = i] = \mathbf{S}[i,j] = s_{i,j}$.

a) 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

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

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.

[5]The index of matrix elements start from 0

C. Delayed Signaling

In some instances, the authentication server might implement the password strength oracle getStrength() by training a (differentially private) Count-Sketch based on the user-selected passwords $pw_u \sim \mathcal{P}$. In this case, 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 $(u, salt_u, sig_u = \perp, h_u)$. During the next (successful) login with the password pw_u the server can update $sig_u = \mathsf{getSignal}(\mathsf{getStrength}(pw_u))$.

V. Adversary Model

We adapt the economic model of [1] 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.

a) 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 [75] and [76] 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 [6].

A. Adversary Utility: No Signaling

We first discuss how a rational adversary would behave when is 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

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

[6]The energy cost of transferring 1GB of memory between RAM and cache is approximately $0.3J$ on an [77], 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 [37], [78] 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.

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.

B. 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

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

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.,

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

VI. 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.

A. 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

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

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 information signaling,

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

where

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

B. 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 [71] to further speed up the computation of the attacker's optimal strategy π^*_y and B^*_y — see discussion in the appendix.

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,

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

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

C. Optimal Signaling Strategy

Once the function $\mathsf{getStrength}()$ 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 $P^s_{adv}(v, k, \mathbf{S})$ denote the expected adversary success rate with information signaling when playing with his/her optimal strategy, then

$$P^s_{adv}(v, k, \mathbf{S}) = \sum_{y \in SL} \Pr[Sig = y] \lambda(\pi^*_y, B^*_y; \mathbf{S}, y), \tag{8}$$

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 $P^s_{adv}(v,k,\mathbf{S})$, therefore we define

$$U^s_{ser}\left(v,k,\{\mathbf{S},(\vec{\pi}^*,\vec{B}^*)\}\right) = -P^s_{adv}(v,k,\mathbf{S}). \quad (9)$$

Our focus of this paper is to find the optimal signaling strategy, namely, the signaling matrix $\mathbf{S}^*$ such that $\mathbf{S}^* = \arg\min_{\mathbf{S}} P^s_{adv}(v,k,\mathbf{S})$. 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,

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

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

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

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 $\mathbf{S}$.

VII. Theoretical Example

Having presented our Stackelberg Game model for information signaling we now give an (admittedly contrived) example of a password distribution where information signaling can dramatically reduce the percentage of cracked passwords. We assume that the attacker has value $v = 2k+\epsilon$ for each cracked password where the cost of each password guess is k and $\epsilon > 0$ is a small constant.

a) Password Distribution: Suppose that $\mathbb{P} = \{\mathrm{pw}_i\}_{i\geq 1}$ and that each password pw_i has probability 2^{-i} i.e., $\Pr_{pw\sim\mathcal{P}}[pw = i] = 2^{-i}$. The weakest password pw_1 would be selected with probability $1/2$.

b) Optimal Attacker Strategy without Signaling: By checking passwords in descending order of probability (the checking sequence is π) the adversary has an expected cost

$$C(k,\pi,B) = k\sum_{i=1}^{B} i \times 2^{-i} + 2^{-B} \times B \times k = 2k(1-2^{-B}).$$

The expression above is equivalent to equation (2), the attacker succeeds in ith guess with probability 2^{-i} at the cost of ik; the attacker fails after making B guess with probability 2^{-B} at the cost of Bk. The the expected gain is

$$R(v,k,\pi,B) = v\sum_{i=1}^{B} i2^{-i} = v\left(1-2^{-B}\right).$$

Therefore the expected utility is

$$U_{adv}(v,k,\pi,B) = R(v,k,B) - C(k,\pi,B) = (v-2k)\left(1-2^{-B}\right).$$

A profit-motivated adversary is interested in calculating $B^* = \operatorname{argmax}_B U_{adv}(v,k,\pi,B)$. With our sample distribution we have

$$B^* = \begin{cases} 0, & \text{if } v < 2k, \\ \infty & \text{if } v \geq 2k. \end{cases}$$

Since we assume that $v = 2k+\epsilon > 2k$ the attackers optimal strategy is $B^* = \infty$ meaning that 100% of passwords will be cracked.

c) Signaling Strategy: Suppose that $\mathsf{getStrength}$ is define such that $\mathsf{getStrength}(pw_1) = 0$ and $\mathsf{getStrength}(pw_i) = 1$ for each $i > 1$. Intuitively, the strength level is 0 if and only if we sampled the weakest password from the distribution. Now suppose that we select our signaling matrix

$$\mathbf{S} = \begin{bmatrix} 1/2 & 1/2 \\ 0 & 1 \end{bmatrix},$$

such that $\Pr[Sig = 0 \mid pw = pw_1] = \frac{1}{2} = \Pr[Sig = 1 \mid pw = pw_1]$ and $\Pr[Sig = 1 \mid pw \neq pw_1] = 1$.

d) Optimal Attacker Strategy with Information Signaling: We now analyze the behavior of a rational attacker under signaling when given this signal matrix and password distribution assuming that the attacker's value $v = 2k+\epsilon$ is the same.

We first note that attacker observes the signal $Sig = 0$ we know for sure that the user selected the most common password as $\Pr[pw = pw_1 \mid Sig = 0] = 1$ so as long as $v \geq k$ the attacker will crack the password.

Next we consider the case that the attacker observes the signal $Sig = 1$. We have the following posterior probabilities for each of the passwords in the distribution:

$$\Pr[pw = pw_1 \mid Sig = 1] = \frac{0.5 \times 0.5}{0.75} = \frac{1}{3},$$
$$\Pr[pw = pw_i, i > 1 \mid Sig = 1] = \frac{1 \times 2^{-i}}{0.75} = \frac{4 \times 2^{-i}}{3}.$$

Now we compute the attacker's expected costs conditioned on $Sig = 1$.

$$\begin{aligned} C(k,\pi,B;\mathbf{S},1) &= k\left(\frac{1}{3} + \frac{4}{3}\sum_{i=2}^{B} i * 2^{-i}\right) + kB\left(\frac{2}{3} - \frac{4}{3}\left(\sum_{i=2}^{B} 2^{-i}\right)\right) \\ &= k\left(\frac{7}{3} - \frac{2^{3-B}}{3}\right), \end{aligned}$$

The expected gain of the attacker is

$$R(v,k,\pi,B;\mathbf{S},1) = v\left(\frac{1}{3} + \frac{4}{3}\sum_{i=2}^{B} 2^{-i}\right) = v\left(1 - \frac{2^{2-B}}{3}\right),$$

Thus, the attacker's utility is given by:

$$\begin{aligned} U_{adv}(v,k,\pi,B;\mathbf{S},1) &= R(v,k,\pi,B;\mathbf{S},1) - C(k,\pi,B;\mathbf{S},1) \\ &= v\left(1 - \frac{2^{2-B}}{3}\right) - k\left(\frac{7}{3} - \frac{2^{3-B}}{3}\right). \end{aligned}$$

Assuming that $\epsilon < \frac{1}{3}k$ the attacker will have negative utility $U_{adv}(2k+\epsilon, k, \pi, B; \mathbf{S}, 1) < 0$ whenever $B > 1$. Thus, when the signal is $Sig = 1$ the optimal attacker strategy is to select $B^* = 0$ (i.e., don't attack) to ensure zero utility. In particular, the attacker cracks the password if and only if $Sig = 0$ which happens with probability $1 - \Pr[Sig = 1] = 0.25$ since $\Pr[Sig = 1] = \Pr[pw = pw_1]\Pr[Sig = 1 \mid pw = pw_1] + \Pr[pw \neq pw_1]\Pr[Sig = 1 \mid pw \neq pw_1] = \frac{3}{4}$. Thus, the attacker will only crack 25% of passwords when $v = 2k + \epsilon$[7].

e) Discussion: In our example an attacker with value $v = 2k + \epsilon$ cracks 100% of passwords when we don't use information signaling. However, if our information signaling mechanism (above) were deployed, the attacker will only crack 25% of passwords — a reduction of 75%! Given this (contrived) example it is natural to ask whether or not information signaling produces similar results for more realistic password distributions. We explore this question in the next sections.

VIII. Experimental Design

We now describe our empirical experiments to evaluate the performance of information 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 generate a 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.

In the remainder of this section we describe how the oracle $\mathsf{getStrength}()$ is implemented in different experiments, the password distribution(s) derived from empirical password datasets and how we implement $\mathsf{genSigMat}()$.

A. 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 privacy [6] and other privacy-preserving measures [5]. All the other datasets come from server breaches.

a) Empirical Distribution: For all 9 datasets we can derive an empirical password distribution $\mathcal{D}_e$ where $\Pr_{pw \sim \mathcal{D}_e}[pw_i] = f_i/N$. Here, N is the number of users in the dataset and f_i is the number of occurences of pw_i in the dataset. We remark that for datasets like Yahoo! and LinkedIn where the datasets only include frequencies f_i without the original plaintext password we can derive a distribution simply by generating unique strings for each password. The empirical distribution is useful to analyze the performance of information 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 incentive to start cracking passwords with lower frequency. Following an approach taken in [71], we use Good-Turing frequency estimation to identify and highlight regions of uncertainty where the CDF for the empirical distribution might significantly diverge from the real password distribution. To simulate an attacker with imperfect knowledge of the distribution we train a differentially private Count-Mean-Min-Sketch. In turn, the Count-Sketch is used to derive a distribution $\mathcal{D}_{train}$, to implement $\mathsf{getStrength}()$ and to generate the signaling matrix $\mathbf{S}^* \leftarrow \mathsf{genSigMat}(v, k, a, b, \mathcal{D}_{train})$ (see details below).

b) Monte Carlo Distribution: To derive the Monte Carlo password distribution from a dataset we follow a process from [71]. In particular, we subsample passwords $D_s \subseteq D$ from the dataset and derive guessing numbers $\#guessing_m(pw)$ for each $pw \in D_s$. Here, $\#guessing_m(pw)$ denotes the number of guesses needed to crack pw with a state of the art password cracking model m e.g., Probabilistic Context-Free Grammars [22]–[24], Markov models [25]–[28], and neural networks [29]. We used the password guessing service [28] to generate the guessing numbers for each dataset. We then fit our distribution to the guessing curve i.e., fixing thresholds $t_0 = 0 < t_1 < t_2 \ldots$ we assign any password pw with $t_{i-1} < \min_m\{\#guessing_m(pw)\} \leq t_i$ to have probability $\frac{g_i}{|D_s|(t_i - t_{i-1})}$ where g_i counts the number of sampled passwords in D_s with guessing number between t_{i-1} and t_i. Intuitively, the Monte Carlo distribution $\mathcal{D}_m$ models password distribution from the attacker's perspective. One drawback is that the distribution would change if the attacker were to develop an improved password cracking model.

We extract Monte Carlo distribution from 6 datasets (Bfield, Brazzers, Clixsense, CSDN, Neopets, 000webhost) for which we have plain text passwords so that we can query Password Guessing Service [28] about password guessing numbers. In the imperfect knowledge setting we repeated the process above twice for each dataset with disjoint sub-samples to derive two distributions $\mathcal{D}_{train}$ and D_{eval}.

B. 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

[7] If the attacker's value was increased to $v = 4k$ then the attacker would crack 100% of passwords since we would have $U_{adv}(4k, k, \pi, B; \mathbf{S}, 1) = k\left(\frac{5}{3} - \frac{2^{3-B}}{3}\right)$ which is maximized at $B^* = \infty$. In this case the defender would want to look for a different password signaling matrix.

into a Count-Mean-Min-Sketch initialized with Laplace noise to ensure differential privacy.

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 fix the width d_w (resp. depth d_s) of our count sketch to be $d_w = 10^8$ (resp. $d_s = 10$) and add Laplace Noise with scaling factor $b = d_s/\epsilon_{pri} = 5$ to preserve $\epsilon_{pri} = 2$-differential privacy. Since we were not optimizing for space we set the number of columns d_w to be large to minimize the probability of hash collisions and increase the accuracy of frequency estimation. Each cell is encoded by an 4-byte int type so the total size of the sketch is 4 GB.

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 getStrength() (see details below). We then use $\mathcal{D}_{train}$ to derive frequency thresholds for getStrength() and to generate the signaling matrix $\mathbf{S}^* = \mathsf{genSigMat}(v, k, a, b, \mathcal{D}_{train})$. Finally we evaluate results on the original empirical distribution $\mathcal{D}_e$ for the original dataset D i.e., $P^s_{adv} = \mathsf{evaluate}(v, k, a, b, \mathbf{S}^*, \mathcal{D}_e)$.

C. Implementing getStrength()

Given a distribution $\mathcal{D}$ and a frequency oracle $\mathcal{O}$ which outputs $f(pw)$ in the perfect knowledge setting and an estimate of frequency $\hat{f}(pw)$ in the imperfect knowledge setting, we can specify getStrength() by selecting thresholds $x_1 > \ldots > x_{a-1} > x_a = 1$. In particular, if $x_{i+1} \leq \mathcal{O}(pw) < x_i$ then $\mathsf{getStrength}(pw) = i$ and if $\mathcal{O}(pw) \geq x_1$ then $\mathsf{getStrength}(pw) = 0$. Let $Y_i \doteq \Pr_{pw \sim \mathcal{D}}[x_i \leq \mathsf{getStrength}(pw) < x_{i-1}]$ for $i > 1$ and $Y_1 = \Pr_{pw \sim \mathcal{D}}[\mathsf{getStrength}(pw) > x_1]$. We fix the thresholds $x_1 \geq \ldots \geq x_{a-1}$ to (approximately) balance the probability mass of each strength level i.e., to ensure that $Y_i \approx Y_j$. In imperfect (resp. perfect) knowledge settings we use $\mathcal{D} = \mathcal{D}_{train}$ (resp. $\mathcal{D} = \mathcal{D}_{eval}$) to select the thresholds.

D. 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 our experiments we will treat f as a black box and use derivative-free optimization methods to find good signaling matrices $\mathbf{S}^*$.

In our experiment, we choose BITmask Evolution OPTimization (BITEOPT) algorithm [79] to compute the quasi-optimal signaling matrix $\mathbf{S}^*$. BITEOPT is a free open-source stochastic non-linear bound-constrained derivative-free optimization method (heuristic or strategy). BiteOpt took 2nd place (1st by sum of ranks) in BBComp2018-1OBJ-expensive competition track [80].

In each experiment we use 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 $k \leq C_{max}$. We refer to the procedure as $\mathbf{S}^* \leftarrow \mathsf{genSigMat}(v, k, a, b, \mathcal{D}_1)$.

IX. Empirical Analysis

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

A. Password Signaling against Offline Attacks

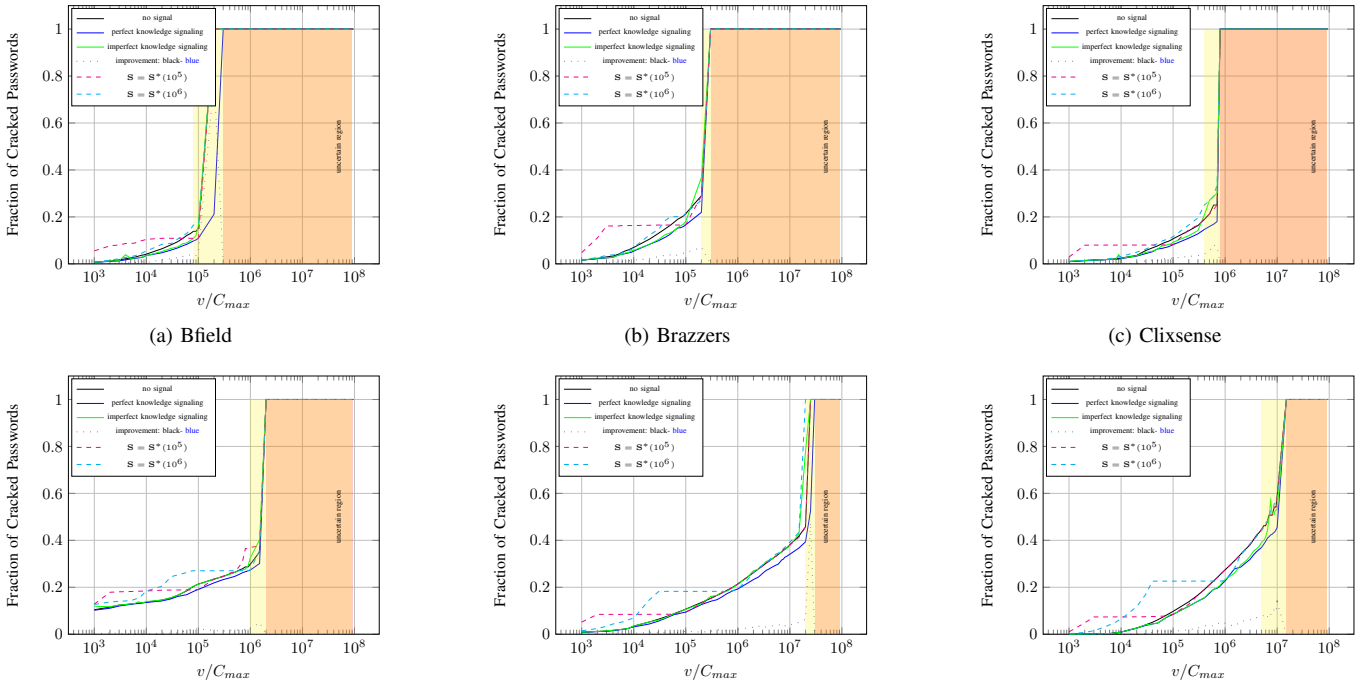
We consider four scenarios using the empirical/Monte Carlo distribution in a setting where the defender has perfect/imperfect knowledge of the distribution.

1) 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 $v/C_{max} \in \{i \times 10^j : 1 \leq i \leq 9, 3 \leq j \leq 7\} \cup \{(i + 0.5) \times 10^j : 1 \leq i \leq 9, 6 \leq j \leq 7\}$. 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 6 datasets — additional plots can be found in Fig 5 in the Appendix.

We follow the approach of [71], 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.

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 affect affect when v/C_{max} is large. This is because the noisy distribution will be less accurate for stronger passwords that were sampled only once.



(a) Bfield (b) Brazzers (c) Clixsense

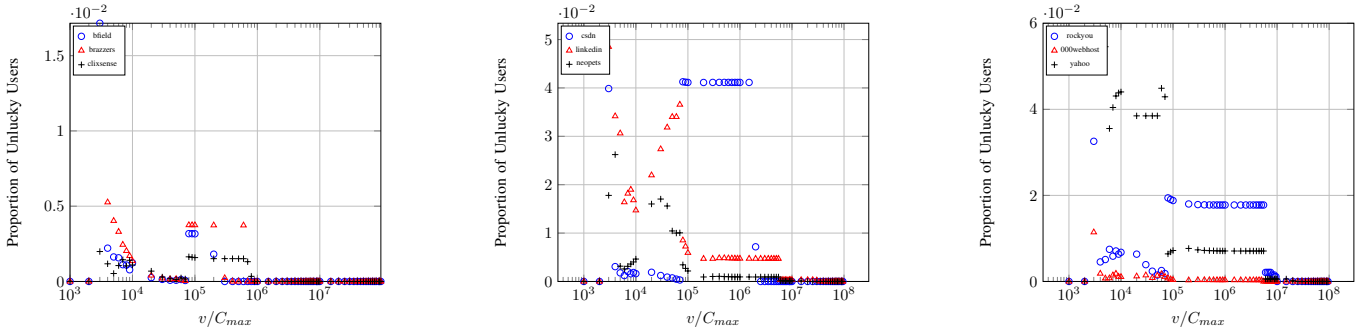
(d) CSDN (e) Linkedin (f) Neopets

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 diverges from the real distribution $E \geq 0.1$ (resp. $E \geq 0.01$).

a) Which accounts are cracked?: As Fig 1 demonstrates information 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 information 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 information 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 $E[X_u] \approx 0.03\%$ and $E[L_u] \approx 6\%$ for LinkedIn. In all instances the net advantage $E[L_u] - E[X_u]$ remains positive. We remark that the reduction in cracked passwords does not come from persuading the attacker to crack weak passwords though the attacker might shift his attention. The shift of attacker's attention is directionless, not necessarily towards weaker passwords. The contribution of attention shift to reduction in cracked passwords is very small since the passwords ordering of posterior distribution upon receiving a signal is very close to that of prior distribution, which means the attacker cracks passwords (almost) in the same order whether given the signal or not. Strength signaling works mainly because the attacker would like to save cost by making less futile guesses.

b) 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. 1f 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.

2) Monte Carlo Distribution: We use the Monte Carlo distribution to evaluate information signaling when v/C_{max} is large. In particular, we subsample 25k passwords from each datast for which we have plain text passwords (excluding Yahoo! and LinkedIn) and obtain guessing numbers from the Password Guessing Service. Then we split our 25k subsamples in half to obtain two guessing curves and we extract two Monte Carlo distributions $\mathcal{D}_{train}$ and $\mathcal{D}_{eval}$ from these curves (see details in the last section). In the perfect knowledge setting the signaling matrix is both optimized and tested on $\mathcal{D}_{eval}$ i.e., $\mathbf{S}^* = \mathsf{genSigMat}(v, k, a, b, \mathcal{D}_{eval})$, $P^s_{adv} = \mathsf{evaluate}(v, k, a, b, \mathbf{S}^*, \mathcal{D}_{eval})$. In the imperfect knowledge setting the signaling matrix is tuned on $\mathcal{D}_{train}$ while the



(a) bfield, brazzers, clixsense (b) csdn, LinkedIn, neopets (c) RockYou, 000webhost, Yahoo!

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

attacker's success rate is evaluated on $\mathcal{D}_{eval}$. One advantage of simulating Monte Carlo distribution is that it allows us to evaluate the performance of information signaling against state of the art password cracking models when the v/C_{max} is large. We consider $v/C_{max} \in \{i * 10^j : 1 \leq i \leq 9, 5 \leq j \leq 10\}$ in performance evaluation for Monte Carlo distribution. As before we set $a = 11$ and $b = 3$ so that the signaling matrix is in dimension of 11×3. We present our results in Fig. 3.

Fig. 3 shows that information signaling can significantly reduce the number of cracked passwords. In particular, for the neopets dataset when $v/C_{max} = 6 \times 10^7$ the number of cracked passwords is reduced from 52.2% to 40% (resp. 43.8%) when the defender has perfect (resp. imperfect) knowledge of the distribution. The green curve (signaling with imperfect knowledge) generally lies between the black curve (no signaling) and the blue curve (signaling with perfect information) though we occasionally find points where the green curve lies slightly above the black curve.

B. Password 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 [81], or provide some other proof of work (PoW), after each incorrect login attempt [82]. One advantage of this approach is that a malicious attacker cannot lockout an honest user by repeatedly submitting incorrect passwords [83]. 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, information 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 [84]) 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 information 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. 4. Due to space limitations the results for 6 datasets are in Fig. 6 in the appendix.

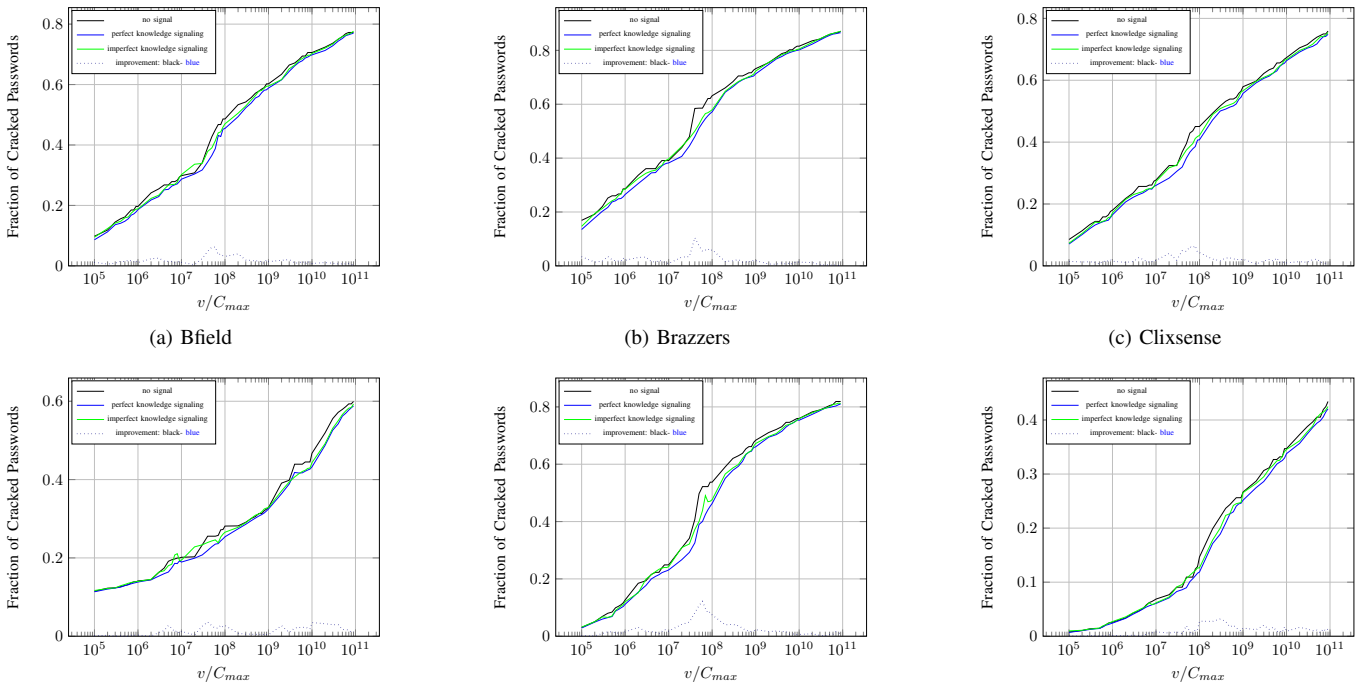
Our results demonstrate that information signaling can be an effective defense against online attackers as well. For example, in Fig. 4b, when $v/C_{max} = 9 \times 10^4$, our mechanism reduces the fraction of cracked passwords from 20.4% to just 15.3%. Similar, observations hold true for other datasets.

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. 4b). 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) Implementing Password Signaling: One naive way to implement password signaling in an online would simply be to explicitly send back the signal noisy signal sig_u in response to any incorrect login attempt. As an alternative we propose a solution where users with a weaker signal sig_u are throttled more aggressively. For example, if sig_u indicates that the password is strong then it might be reasonable to allow for 10 consecutive incorrect login attempts before throttling the account by requiring the user to solve a CAPTCHA challenge before every login attempt. On the other hand if the signal sig_u indicates that the password is weak the server might begin throttling after just 3 incorrect login attempts. The attacker can indirectly infer the signal sig_u by measuring how many login attempts s/he gets before throttling begins. This solution might also provide motivation for users to pick stronger passwords.

C. Discussion

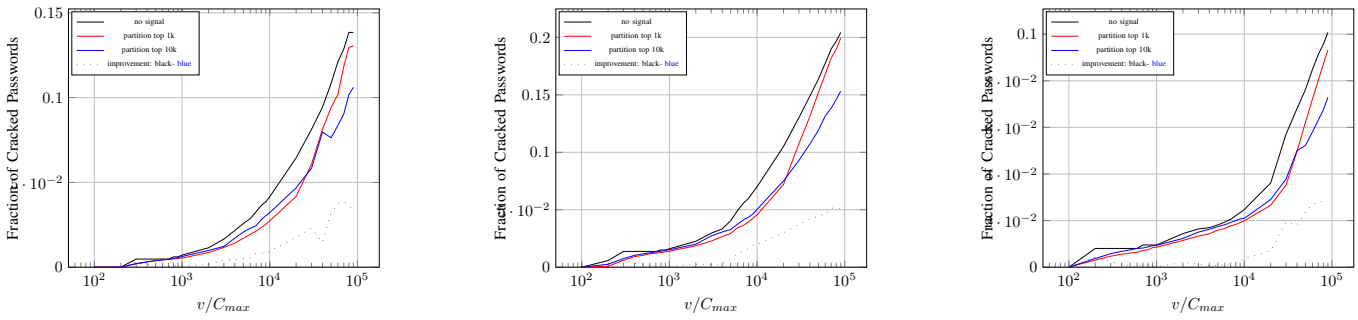
While our experimental results are positive, we stress that there are several questions that would need to be addressed



(a) Bfield (b) Brazzers (c) Clixsense

(d) CSDN (e) Neopets (f) 000webhost

Fig. 3. Adversary Success Rate vs v/k for Monte Carlo Distributions



(a) Bfield (b) Brazzers (c) Clixsense

Fig. 4. Adversary Success Rate vs v/C_{max} in Defense of Online Attacks

before we recommend deploying information signaling to protect against offline attacks.

- Can we accurately predict the value to cost ratio v/C_{max}? Our results suggest that information signaling is useful even when our estimates deviate by a factor of 2. However, if our estimates are wildly off then information signaling could be harmful.
- While information 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 user's to opt in/out of information 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 (eq 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 [85].

X. Conclusions

We introduced 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 information signaling on 9 password datasets. When testing on the empirical (resp. Monte Carlo) password distribution distribution we find that information signaling reduces the number of passwords that would have been cracked by up to 8% (resp. 12%). Additionally, we find that information 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 by NSF grant number 1755708 and Rolls-Royce through a Doctoral Fellowship.

References

[1] J. Blocki and A. Datta, "CASH: A cost asymmetric secure hash algorithm for optimal password protection," in *IEEE 29th Computer Security Foundations Symposium*, pp. 371–386, 2016.

[2] J. Blocki, B. Harsha, and S. Zhou, "On the economics of offline password cracking," in *2018 IEEE Symposium on Security and Privacy*, pp. 853–871, IEEE Computer Society Press, May 2018.

[3] E. Kamenica and M. Gentzkow, "Bayesian persuasion," *American Economic Review*, vol. 101, pp. 2590–2615, October 2011.

[4] H. Xu and R. Freeman, "Signaling in bayesian stackelberg games," in *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems*, 2016.

[5] J. Bonneau, "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.

[6] J. Blocki, A. Datta, and J. Bonneau, "Differentially private password frequency lists," in *NDSS 2016*, The Internet Society, Feb. 2016.

[7] J. Blocki and B. Harsha, "Linkedin password frequency corpus," 2019.

[8] J. Campbell, W. Ma, and D. Kleeman, "Impact of restrictive composition policy on user password choices," *Behaviour & Information Technology*, vol. 30, no. 3, pp. 379–388, 2011.

[9] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman, "Of passwords and people: measuring the effect of password-composition policies," in *CHI*, pp. 2595–2604, 2011.

[10] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor, "Encountering stronger password requirements: user attitudes and behaviors," in *Proceedings of the Sixth Symposium on Usable Privacy and Security*, SOUPS '10, (New York, NY, USA), pp. 2:1–2:20, ACM, 2010.

[11] J. M. Stanton, K. R. Stam, P. Mastrangelo, and J. Jolton, "Analysis of end user security behaviors," *Comput. Secur.*, vol. 24, pp. 124–133, Mar. 2005.

[12] P. G. Inglesant and M. A. Sasse, "The true cost of unusable password policies: Password use in the wild," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, (New York, NY, USA), pp. 383–392, ACM, 2010.

[13] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor, "Can long passwords be secure and usable?," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, (New York, NY, USA), pp. 2927–2936, ACM, 2014.

[14] S. Komanduri, R. Shay, L. F. Cranor, C. Herley, and S. Schechter, "Telepathwords: Preventing weak passwords by reading users' minds," in *23rd USENIX Security Symposium (USENIX Security 14)*, (San Diego, CA), pp. 591–606, USENIX Association, Aug. 2014.

[15] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor, "How does your password measure up? the effect of strength meters on password creation," in *Proceedings of USENIX Security Symposium*, 2012.

[16] X. Carnavalet and M. Mannan, "From very weak to very strong: Analyzing password-strength meters," in *NDSS 2014*, The Internet Society, Feb. 2014.

[17] M. Steves, D. Chisnell, A. Sasse, K. Krol, M. Theofanos, and H. Wald, "Report: Authentication diary study," Tech. Rep. NISTIR 7983, National Institute of Standards and Technology (NIST), 2014.

[18] D. Florêncio, C. Herley, and P. C. Van Oorschot, "An administrator's guide to Internet password research," in *Proceedings of the 28th USENIX Conference on Large Installation System Administration*, LISA'14, pp. 35–52, 2014.

[19] A. Adams and M. A. Sasse, "Users are not the enemy," *Communications of the ACM*, vol. 42, no. 12, pp. 40–46, 1999.

[20] J. Blocki, S. Komanduri, A. Procaccia, and O. Sheffet, "Optimizing password composition policies," in *Proceedings of the fourteenth ACM conference on Electronic commerce*, pp. 105–122, ACM, 2013.

[21] R. Morris and K. Thompson, "Password security: A case history," *Communications of the ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[22] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in *2009 IEEE Symposium on Security and Privacy*, pp. 391–405, IEEE Computer Society Press, May 2009.

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

[24] R. Veras, C. Collins, and J. Thorpe, "On semantic patterns of passwords and their security impact," in *NDSS 2014*, The Internet Society, Feb. 2014.

[25] C. Castelluccia, M. Dürmuth, and D. Perito, "Adaptive password-strength meters from Markov models," in *NDSS 2012*, The Internet Society, Feb. 2012.

[26] C. Castelluccia, A. Chaabane, M. Dürmuth, and D. Perito, "When privacy meets security: Leveraging personal information for password cracking," *arXiv preprint arXiv:1304.6584*, 2013.

[27] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," in *2014 IEEE Symposium on Security and Privacy*, pp. 689–704, IEEE Computer Society Press, May 2014.

[28] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher, and R. Shay, "Measuring real-world accuracies and biases in modeling password guessability," in *USENIX Security 2015* (J. Jung and T. Holz, eds.), pp. 463–481, USENIX Association, Aug. 2015.

[29] W. Melicher, B. Ur, S. M. Segreti, S. Komanduri, L. Bauer, N. Christin, and L. F. Cranor, "Fast, lean, and accurate: Modeling password guessability using neural networks," in *USENIX Security 2016* (T. Holz and S. Savage, eds.), pp. 175–191, USENIX Association, Aug. 2016.

[30] E. Liu, A. Nakanishi, M. Golla, D. Cash, and B. Ur, "Reasoning analytically about password-cracking software," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 380–397, IEEE, 2019.

[31] "Hashcast: advanced password recovery."

[32] S. Designer, "John the ripper password cracker," 2006.

[33] N. Provos and D. Mazieres, "Bcrypt algorithm," USENIX, 1999.

[34] B. Kaliski, "Pkcs# 5: Password-based cryptography specification version 2.0," 2000.

[35] C. Percival, "Stronger key derivation via sequential memory-hard functions," in *BSDCan 2009*, 2009.

[36] D. Boneh, H. Corrigan-Gibbs, and S. E. Schechter, "Balloon hashing: A memory-hard function providing provable protection against sequential attacks," in *ASIACRYPT 2016, Part I* (J. H. Cheon and T. Takagi, eds.), vol. 10031 of *LNCS*, pp. 220–248, Springer, Heidelberg, Dec. 2016.

[37] A. Biryukov, D. Dinu, and D. Khovratovich, "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.

[38] "Password hashing competition." https://password-hashing.net/.

[39] J. Alwen, B. Chen, K. Pietrzak, L. Reyzin, and S. Tessaro, "Scrypt is maximally memory-hard," in *EUROCRYPT 2017, Part III* (J.-S. Coron

and J. B. Nielsen, eds.), vol. 10212 of *LNCS*, pp. 33–62, Springer, Heidelberg, Apr. / May 2017.

[40] J. Alwen and J. Blocki, "Efficiently computing data-independent memory-hard functions," in *CRYPTO 2016, Part II* (M. Robshaw and J. Katz, eds.), vol. 9815 of *LNCS*, pp. 241–271, Springer, Heidelberg, Aug. 2016.

[41] J. Alwen, J. Blocki, and K. Pietrzak, "Depth-robust graphs and their cumulative memory complexity," in *EUROCRYPT 2017, Part III* (J.-S. Coron and J. B. Nielsen, eds.), vol. 10212 of *LNCS*, pp. 3–32, Springer, Heidelberg, Apr. / May 2017.

[42] J. Blocki, B. Harsha, S. Kang, S. Lee, L. Xing, and S. Zhou, "Data-independent memory hard functions: New attacks and stronger constructions," in *Annual International Cryptology Conference*, pp. 573–607, Springer, 2019.

[43] B. Harsha and J. Blocki, "Just in time hashing," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 368–383, IEEE, 2018.

[44] A. Everspaugh, R. Chatterjee, S. Scott, A. Juels, and T. Ristenpart, "The pythia PRF service," in *USENIX Security 2015* (J. Jung and T. Holz, eds.), pp. 547–562, USENIX Association, Aug. 2015.

[45] J. Camenisch, A. Lysyanskaya, and G. Neven, "Practical yet universally composable two-server password-authenticated secret sharing," in *ACM CCS 2012* (T. Yu, G. Danezis, and V. D. Gligor, eds.), pp. 525–536, ACM Press, Oct. 2012.

[46] R. W. F. Lai, C. Egger, D. Schröder, and S. S. M. Chow, "Phoenix: Rebirth of a cryptographic password-hardening service," in *USENIX Security 2017* (E. Kirda and T. Ristenpart, eds.), pp. 899–916, USENIX Association, Aug. 2017.

[47] J. G. Brainard, A. Juels, B. Kaliski, and M. Szydlo, "A new two-server approach for authentication with short secrets," in *USENIX Security 2003*, USENIX Association, Aug. 2003.

[48] A. Juels and R. L. Rivest, "Honeywords: making password-cracking detectable," in *ACM CCS 2013* (A.-R. Sadeghi, V. D. Gligor, and M. Yung, eds.), pp. 145–160, ACM Press, Nov. 2013.

[49] R. Canetti, S. Halevi, and M. Steiner, "Mitigating dictionary attacks on password-protected local storage," in *CRYPTO 2006* (C. Dwork, ed.), vol. 4117 of *LNCS*, pp. 160–179, Springer, Heidelberg, Aug. 2006.

[50] J. Blocki, M. Blum, and A. Datta, "Gotcha password hackers!," in *Proceedings of the 2013 ACM workshop on Artificial intelligence and security*, pp. 25–34, ACM, 2013.

[51] J. Blocki and H.-S. Zhou, "Designing proof of human-work puzzles for cryptocurrency and beyond," in *TCC 2016-B, Part II* (M. Hirt and A. D. Smith, eds.), vol. 9986 of *LNCS*, pp. 517–546, Springer, Heidelberg, Oct. / Nov. 2016.

[52] Y. Tian, C. Herley, and S. Schechter, "Stopguessing: Using guessed passwords to thwart online guessing," in *Proc. IEEE European Symp. Security and Privacy (EuroS&P 2019)*, pp. 17–19, 2019.

[53] J. Blocki and W. Zhang, "Dalock: Distribution aware password throttling," *arXiv preprint arXiv:2005.09039*, 2020.

[54] D. A. F. Florêncio and C. Herley, "One-time password access to any server without changing the server," in *ISC 2008* (T.-C. Wu, C.-L. Lei, V. Rijmen, and D.-T. Lee, eds.), vol. 5222 of *LNCS*, pp. 401–420, Springer, Heidelberg, Sept. 2008.

[55] A. Pashalidis and C. J. Mitchell, "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.

[56] M. Kuhn, "Otpw—a one-time password login package," 1998.

[57] S. Chiasson, P. C. van Oorschot, and R. Biddle, "Graphical password authentication using cued click points," in *ESORICS 2007* (J. Biskup and J. López, eds.), vol. 4734 of *LNCS*, pp. 359–374, Springer, Heidelberg, Sept. 2007.

[58] R. Jhawar, P. Inglesant, N. Courtois, and M. A. Sasse, "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.

[59] RSA, "Rsa securid® 6100 usb token," 2003.

[60] B. Parno, C. Kuo, and A. Perrig, "Phoolproof phishing prevention," in *FC 2006* (G. Di Crescenzo and A. Rubin, eds.), vol. 4107 of *LNCS*, pp. 1–19, Springer, Heidelberg, Feb. / Mar. 2006.

[61] A. Ross, J. Shah, and A. K. Jain, "From template to image: Reconstructing fingerprints from minutiae points," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 4, pp. 544–560, 2007.

[62] J. Daugman, "How iris recognition works," in *The essential guide to image processing*, pp. 715–739, Elsevier, 2009.

[63] P. S. Aleksic and A. K. Katsaggelos, "Audio-visual biometrics," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 2025–2044, 2006.

[64] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "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.

[65] C. Herley and P. Van Oorschot, "A research agenda acknowledging the persistence of passwords," *IEEE Security & Privacy*, vol. 10, no. 1, pp. 28–36, 2011.

[66] R. Alonso and O. Câmara, "Bayesian persuasion with heterogeneous priors," *Journal of Economic Theory*, vol. 165, pp. 672–706, 2016.

[67] S. Dughmi and H. Xu, "Algorithmic bayesian persuasion," *SIAM Journal on Computing*, vol. 0, no. 0, pp. STOC16–68–STOC16–97, 0.

[68] M. Hoefer, P. Manurangsi, and A. Psomas, "Algorithmic persuasion with evidence," 2020.

[69] T. E. Carroll and D. Grosu, "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.

[70] Z. Rabinovich, A. X. Jiang, M. Jain, and H. Xu, "Information disclosure as a means to security," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, (Richland, SC), p. 645–653, International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[71] W. Bai and J. Blocki, "Dahash: Distribution aware tuning of password hashing costs," in *Financial Cryptography and Data Security*, Springer International Publishing, 2021.

[72] S. Schechter, C. Herley, and M. Mitzenmacher, "Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks," in *Proceedings of the 5th USENIX conference on Hot topics in security*, pp. 1–8, USENIX Association, 2010.

[73] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC 2006* (S. Halevi and T. Rabin, eds.), vol. 3876 of *LNCS*, pp. 265–284, Springer, Heidelberg, Mar. 2006.

[74] G. Cormode, C. Procopiuc, D. Srivastava, and T. T. Tran, "Differentially private summaries for sparse data," in *Proceedings of the 15th International Conference on Database Theory*, pp. 299–311, 2012.

[75] M. Fossi, E. Johnson, D. Turner, T. Mack, J. Blackbird, D. McKinney, M. K. Low, T. Adams, M. P. Laucht, and J. Gough, "Symantec report on the underground economy," November 2008. Retrieved 1/8/2013.

[76] M. Stockley, "What your hacked account is worth on the dark web," Aug 2016.

[77] L. Ren and S. Devadas, "Bandwidth hard functions for ASIC resistance," in *TCC 2017, Part I* (Y. Kalai and L. Reyzin, eds.), vol. 10677 of *LNCS*, pp. 466–492, Springer, Heidelberg, Nov. 2017.

[78] J. Blocki, B. Harsha, S. Kang, S. Lee, L. Xing, and S. Zhou, "Data-independent memory hard functions: New attacks and stronger constructions." Cryptology ePrint Archive, Report 2018/944, 2018. https://eprint.iacr.org/2018/944.

[79] A. Vaneev, "BITEOPT - Derivative-free optimization method." Available at https://github.com/avaneev/biteopt, 2021. C++ source code, with description and examples.

[80] "Black box optimization competition." https://www.ini.rub.de/PEOPLE/glasmtbl/projects/bbcomp/index.html.

[81] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Using hard AI problems for security," in *EUROCRYPT 2003* (E. Biham, ed.), vol. 2656 of *LNCS*, pp. 294–311, Springer, Heidelberg, May 2003.

[82] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks," in *ACM CCS 2002* (V. Atluri, ed.), pp. 161–170, ACM Press, Nov. 2002.

[83] "Hackers find new way to bilk eBay users - CNET," 2019.

[84] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage, "Re: CAPTCHAs-understanding CAPTCHA-solving services in an economic context," in *USENIX Security 2010*, pp. 435–462, USENIX Association, Aug. 2010.

[85] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted online password guessing: An underestimated threat," in *ACM CCS 2016* (E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, eds.), pp. 1242–1254, ACM Press, Oct. 2016.

[86] J. Blocki and A. Datta, "CASH: A cost asymmetric secure hash algorithm for optimal password protection," in *CSF 2016Computer Security Foundations Symposium* (M. Hicks and B. Köpf, eds.), pp. 371–386, IEEE Computer Society Press, 2016.

APPENDIX

COMPRESSED PASSWORD DISTRIBUTIONS

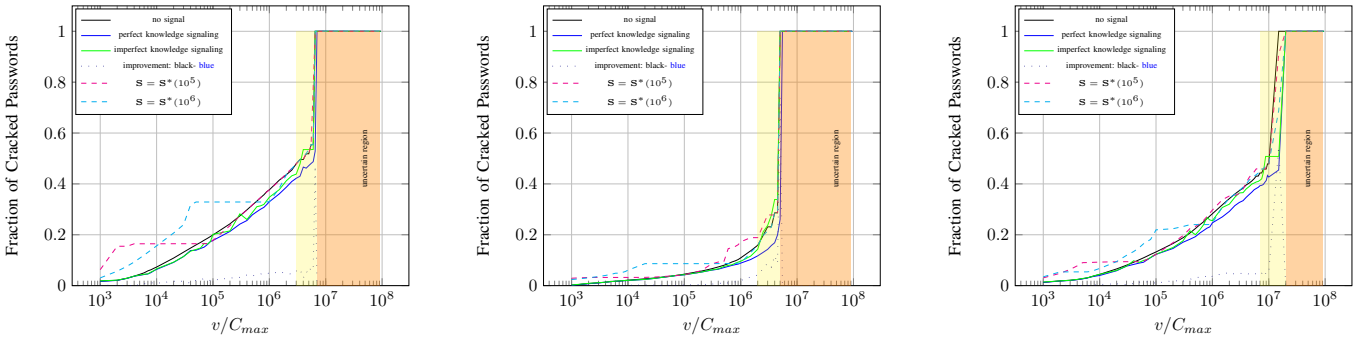
Given a distribution over N passwords with corresponding probabilities $p_1 \geq p_2 \geq \ldots p_N$ we can often encode the probabilities in compressed form by grouping passwords with equal probability. Viewed in this way we can encode the distribution as a sequence of $N' \leq N$ tuples $(p_1, c_1), \ldots, (p_{N'}, c'_N)$ where $p_1 > p_2 > \ldots > p_{N'}$ and c_i denotes the number of passwords with probability exactly p_i in the distribution. In all of the distributions we analyze we have $N' \ll N$ e.g., for the RockYou empirical we have $N \geq 3.2 \times 10^7$ while $N' \leq 2.3 \times 10^3$. Thus, it is desirable to ensure that our optimization algorithms scale with N' instead of N. Similar observations were used by Blocki and Datta [86] and Bai and Blocki [71].

Bai and Blocki [71, Lemma 1] showed that a rational adversary never *splits* equivalence sets i.e., if $\Pr[pw_i] = \Pr[pw_j]$ then the attacker will either check both passwords or neither. Thus, a rational attacker's optimal strategy will be to check the B^* most popular passwords where B^* is guaranteed to be in the set $\{0, c_1, c_1 + c_2, \ldots, \sum_{i=1}^{N'} c_i\}$. When the distribution is compact this substantially narrows down the search space in comparison to a brute-force search over all possible choices of $B^* \in [N]$.

We observe that if the original password distribution has a compact representation $(p_1, c_1), \ldots, (p_{N'}, c'_N)$ with N' equivalence sets then, after observing the password signaling $y \in [b]$, the posterior distribution has a compact representation with *at most* aN' equivalence sets where the dimension of the signaling matrix is $a \times b$ i.e., $\mathbf{S} \in \mathbb{R}^{a \times b}$. To see this notice that we can partition all passwords pw in equivalence set i into a groups based on the value $\mathsf{getStrength}(pw) \in [a]$. If $\Pr[pw] = \Pr[pw']$ and $\mathsf{getStrength}(pw) = \mathsf{getStrength}(\mathsf{pw})$ then the posterior probabilities are also equal i.e., $\Pr[pw|y] = \Pr[pw'|y]$.

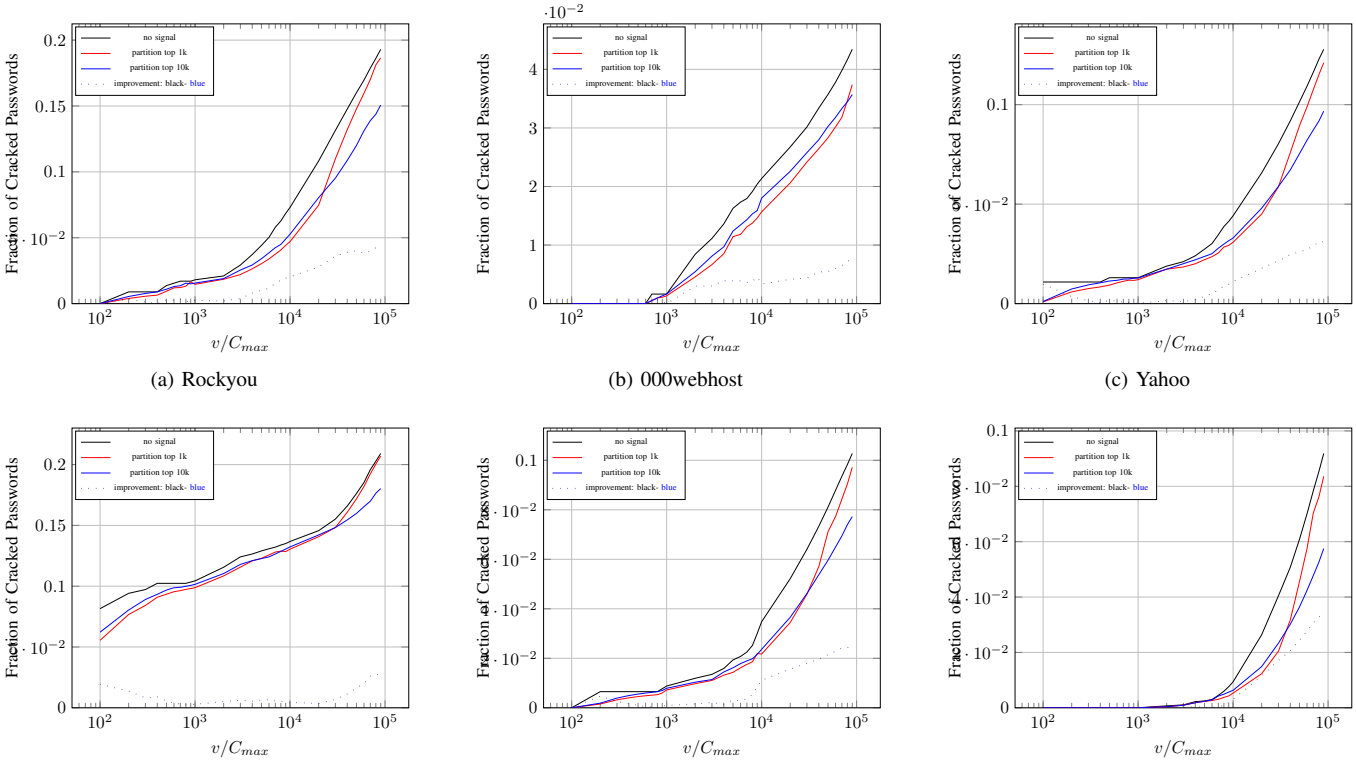
Thus, given a signaling matrix $\mathbf{S}$ and a signal $y \in [b]$ and we can compute the adversaries optimal response (π_b^*, B_y^*) to the signal y by 1) computing the compact representation of the posterior distribution, and 2) checking all aN' possible values of B_y^* to find the budget that maximizes the attacker's expected utility conditioning on the signal y. After preprocessing the original dataset the first step only requires time $O(aN' \log aN')$ for each new signaling matrix $\mathbf{S}$ and $y \in [b]$.

EXTRA PLOTS



(a) Rockyou (b) 000webhost (c) Yahoo

Fig. 5. 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 diverges from the real distribution $E \geq 0.1$ (resp. $E \geq 0.01$).



(a) Rockyou (b) 000webhost (c) Yahoo

(d) CSDN (e) Linkedin (f) Neopets

Fig. 6. Adversary Success Rate vs v/C_{max} in Defense of Online Attacks