Confident Monte Carlo: Rigorous Analysis of Guessing Curves for Probabilistic Password Models

Peiyuan Liu*
Computer Science Department
Purdue University
West Lafayette, IN USA
liu2039@purdue.edu

Jeremiah Blocki*
Computer Science Department
Purdue University
West Lafayette, IN USA
jblocki@purdue.edu

Wenjie Bai*
Computer Science Department
Purdue University
West Lafayette, IN USA
bai104@purdue.edu

2023 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-9336-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SP46215.2023.10179365

***Abstract*—In password security a defender would like to identify and warn users with weak passwords. Similarly, the defender may also want to predict what fraction of passwords would be cracked within B guesses as the attacker's guessing budget B varies from small (online attacker) to large (offline attacker). Towards each of these goals the defender would like to quickly estimate the guessing number for each user password pwd assuming that the attacker uses a password cracking model M i.e., how many password guesses will the attacker check before s/he cracks each user password pwd. Since naïve brute-force enumeration can be prohibitively expensive when the guessing number is very large, Dell'Amico and Filippone [1] developed an efficient Monte Carlo algorithm to estimate the guessing number of a given password pwd. While Dell'Amico and Filippone proved that their estimator is unbiased there is no guarantee that the Monte Carlo estimates are accurate nor does the method provide confidence ranges on the estimated guessing number or even indicate if/when there is a higher degree of uncertainty.**

Our contributions are as follows: First, we identify theoretical examples where, with high probability, Monte Carlo Strength estimation produces highly inaccurate estimates of individual guessing numbers as well as the entire guessing curve. Second, we introduce Confident Monte Carlo Strength Estimation as an extension of Dell'Amico and Filippone [1]. Given a password our estimator generates an upper and lower bound with the guarantee that, except with probability δ, the true guessing number lies within the given confidence range. Our techniques can also be used to characterize the attacker's guessing curve. In particular, given a probabilistic password cracking model M we can generate high confidence upper and lower bounds on the fraction of passwords that the attacker will crack as the guessing budget B varies.

***Index Terms*—Monte Carlo Estimation, Password Cracking Models, Concentration bounds**

I. Introduction

In addition to their offensive uses, Probabilistic Password Cracking Models have many defensive applications. One defensive application is to use the password cracking model to estimate the strength of a user's password during account registration so that we can warn users who attempt to register with a weak password that would be easy for an attacker to guess. For this application we would like to quickly determine how many guesses an attacker using password cracking model M would need to check before s/he cracks a particular user's password pwd. One way to determine the guessing number of a particular password pwd would simply be to enumerate all possible password guesses, ordered according to their probability under the model M, keeping track of the number of incorrect guesses which appear before the correct password pwd. However, naïve brute-force enumeration is often prohibitively expensive especially when the guessing number is very large e.g., $> 10^{15}$.

Dell'Amico and Filippone [1] developed a Monte Carlo algorithm to efficiently output an estimate of the guessing number of a given password pwd without resorting to brute-force enumeration. Their technique applies generically to any probabilistic password model M under the assumption that (1) the model M defines a distribution over passwords and we can efficiently sample from this distribution, and (2) given a particular password pwd we can quickly compute the probability that this password is generated by M. Dell'Amico and Filippone proved that their Monte Carlo estimator is unbiased i.e., the expected value of the estimate is equal to the actual guessing number. However, there is no absolute guarantee that the estimate is accurate. Their Monte Carlo estimator does not provide any statistical confidence intervals for the range of possible guessing numbers or even indicate if/when there is a high degree of uncertainty about the true guessing number.

In other defensive applications we may want to estimate the attacker's entire guessing curve. How many consecutive incorrect login attempts should be allowed before we lockdown an account? Will doubling the cost of the password hash function significantly reduce the fraction of passwords that an offline attacker will crack?

Formally, let $\lambda_{M,B,D}$ denote the fraction of passwords in a dataset D which would be cracked within the first B guesses generated by model M. Similarly, let $\lambda_{M,B}$ denote the probability that a fresh password sampled from an (unknown) distribution $\mathcal{P}$ over user passwords would be cracked within the first B guesses output by the model M. Characterizing the entire guessing curve $\lambda_{M,B,D}$ (or $\lambda_{M,B}$) as B ranges from small (online attack) to very large (offline attack) can help a defender set password policies. Thus, we would like to

This research was supported in part by the National Science Foundation under awards CCF #1910659 and CNS #2047272, a gift from Protocol Labs, and by a Purdue Big Ideas award.

*The authors are listed in reverse alphabetical order.

DOI 10.1109/SP46215.2023.00185

generate high confidence upper/lower bounds on the guessing curves $\lambda_{M,B,D}$ and/or $\lambda_{M,B}$.

However, when B is sufficiently large we cannot efficiently compute $\lambda_{M,B,D}$ since that would require enumeration of the top B passwords in the distribution M. As a heuristic approach we could use Monte Carlo strength estimation [1] to quickly estimate the guessing number of each password in D and then compute the fraction of passwords whose estimated guessing number is below B. While this heuristic approach has become popular in the password literature (e.g., [2]–[5]), it could yield poor estimates of $\lambda_{M,B,D}$ when the estimated guessing numbers are inaccurate. Indeed, in our empirical analysis we identify several instances where the heuristically estimated guessing curve is highly inaccurate. These problematic cases tend to arise when the guessing number B is very large. The bottom line is that there is no absolute guarantee that our heuristic estimate of $\lambda_{M,B,D}$ is accurate nor does this heuristic approach provide any statistical confidence intervals on the attacker's guessing curve.

In many settings we would like to characterize the guessing curve $\lambda_{M,B}$ using a relatively small number of samples from our unknown password distribution. For example, suppose that we conduct a user study to determine whether or not a particular password policy intervention, e.g., requiring users to select passwords with upper and lower case letters[1], effectively strengthens the distribution over user passwords. We can view the data collected from the user study as samples D_0 (control group) and D_1 (intervention group) from two different (unknown) password distributions $\mathcal{P}_0$ and $\mathcal{P}_1$ respectively. We would like to use these samples to draw statistical comparisons about the guessing curves before/after the policy intervention i.e., does the policy reduce the fraction of user passwords cracked by an offline attacker making $B = 10^{15}$ guesses per user. Observe that if we assume that the dataset S_i was sampled iid from distribution $\mathcal{P}_i$ that we have $\lambda_{M,B} = \mathbb{E}[\lambda_{M,B,D}]$. We could use the popular heuristic estimate for λ_{M,B,D_i} (described above) to estimate the guessing curves $\lambda_{M,B}$ for the two different distributions. However, we now have an additional source of estimation error due to sampling of the datasets D_0 and D_1 — in addition to the error from Monte Carlo strength estimation. One challenge is that the number of samples collected from the user study will typically be constrained by research budgets making it harder to ensure that the sampling error is small.

A. Contributions

First, we provide theoretical examples of models M and passwords pwd where regular Monte Carlo will (1) significantly overestimate the true guessing number of pwd with high probability, and (2) underestimate the guessing number by a factor of ≈ 2 with probability at least 0.3. We also consider the popular heuristic of using regular Monte Carlo strength estimation to estimate the attacker's guessing curve. We provide a (theoretical) example of a model M and a password distribution $\mathcal{P}$ such that (1) an attacker following model M will actually crack 0% of passwords within B guesses i.e., $\lambda_{M,B} = 0$ and $\lambda_{M,B,D} = 0$, but (2) the popular heuristic using regular Monte Carlo Strength estimation incorrectly predicts that an attacker will crack 100% of user passwords sampled from $\mathcal{P}$ within B guesses! We argue that such issues are inherent to *any* blackbox method for Monte Carlo strength estimation.

Second, we introduce several rigorous statistical techniques to upper/lower bound the guessing number of a password and show how these techniques can be extended to upper/lower bound an attacker's entire guessing curve. On a technical note our upper/lower bounds are derived using concentration inequalities such as Hoeffding and Chernoff as well as a strategic application of Markov's inequality. We call our new toolkit Confident Monte Carlo. In particular, given a password cracking model M, a particular password pwd and a confidence parameter δ Confident Monte Carlo will output an upper bound U and a lower bound L on the (unknown) true guessing number $G(pwd)$ with the guarantee that $\Pr[L \leq G(pwd)] \geq 1-\delta$ and $\Pr[U \geq G(pwd)] \geq 1-\delta$. We stress that this is distinct from the weaker task of finding of upper/lower bounds for a Monte Carlo estimate $\hat{G}(pwd)$ of $G(pwd)$ e.g., finding $L < U$ such that $\Pr[L \leq \hat{G}(pwd) \leq U] \geq 1-\delta$. This weaker goal could be easily accomplished by standard techniques for generating confidence intervals such as multiple sampling iterations. Confident Monte Carlo works under the exact same generic assumptions as regular Monte Carlo strength estimation.[2] Thus, whenever we can apply apply regular Monte Carlo strength estimation to estimate guessing numbers for a password cracking model M we can also apply our Confident Monte Carlo techniques to obtain confidence bounds for the estimated guessing number. Empirical analysis shows that our upper/lower bounds on the guessing number are usually quite close. Typically, we find that the estimates generated by regular Monte Carlo lie within our confidence range, but we also find that for many rare passwords the estimated guessing number generated by regular Monte Carlo is *demonstrably* inaccurate e.g., the estimate lies below our *lower bound*.

Third, we develop rigorous statistical techniques to upper/lower bound the attacker's entire guessing curve. In particular, given a dataset D we can generate curves $\lambda^{ub}_{M,B,D}$ (resp. $\lambda^{lb}_{M,B,D}$) such that with probability at least $1-\delta$ for all guessing budgets B we have $\lambda_{M,B,D} \leq \lambda^{ub}_{M,B,D}$ (resp. $\lambda_{M,B,D} \geq \lambda^{ub}_{M,B,D}$).

If we assume that our dataset D was sampled iid from the (unknown) password distribution $\mathcal{P}$ then we can apply McDi-

[1]There are many password interventions that one could consider. We could require passwords to include special characters or numbers (password composition policies). We could display a password strength meter during registration. We could warn users about the risks of picking weak passwords before registration. We could require users to participate in training activities or watch an instructional video about picking good passwords before registration.

[2]We assume that (1) the model M describes a distribution $\mathcal{M}$ over passwords and we can efficiently sample from this distribution, and (2) given a particular password pwd we can quickly compute the probability that this password is generated by M.

armid's inequality to argue that (whp) $|\lambda_{M,B,D} - \lambda_{M,B}| \leq \epsilon$. This allows us to confidently upper/lower bound the guessing curve $\lambda_{M,B}$ for our (unknown) user password distribution $\mathcal{P}$ using only a dataset D sampled from this distribution. This observation also provides a rigorous statistical framework for many natural tasks in password research (1) analyzing the impact of a password policy on the guessing curve given samples S_1 (resp. S_2) from the unknown distribution $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) representing the distribution of user passwords before (resp. after) the policy intervention, (2) comparing the performance of different password cracking models M and M' against an unknown password distribution $\mathcal{P}$.

Finally, we evaluate Confident Monte Carlo empirically using several large breached password datasets. We find that our upper/lower bounds on the guessing curve are typically very close and thus tightly bound the true values $\lambda_{M,B,D}$ or $\lambda_{M,B}$. We compare our upper/lower bounds with the popular heuristic estimate using Regular Monte Carlo strength estimation. On the positive side, our empirical experiments demonstrate that Regular Monte Carlo generates reasonable estimates of the guessing curve in most cases i.e., as long as the guessing budget B is not too large (e.g., $B \leq 10^{20}$) the estimated guessing curve is tightly sandwiched between our upper/lower bounds. Thus, Confident Monte Carlo can often be used as a tool to verify if the heuristic guessing curve generated by Regular Monte Carlo is accurate. On the negative side when the guessing budget B grows very large we find several examples where the Regular Monte Carlo guessing curve is *provably inaccurate* i.e., lies above our upper bound.

B. Related Work

Password Guessing Models. Offline password attacks have been a concern since the Unix system was devised [6]. Many sophisticated probabilistic password models have been proposed to generate password guesses for an online attacker such as Probabilistic Context-Free Grammars [7]–[9], Markov models [10]–[13], and neural networks [2]. Each of these probabilistic password models are compatible with regular Monte Carlo Strength estimation [1] and our Confident Monte Carlo techniques. Thus, we can apply our statistical techniques to derive high confidence upper/lower bounds on guessing numbers for each of these models. By contrast, heuristic (rule-based) tools such as Hashcat [14] and John the Ripper [15] are not compatible with Monte Carlo Strength Estimation. Liu et al. [3] developed tools to estimate guessing numbers for Hashcat and John the Ripper without resorting naïve brute-force enumeration.

Password Strength Estimation. Regular Monte Carlo strength estimation [1] has been widely used in the password research literature to understand the impact of culture/language on password strength [16], evaluate the impact of policy interventions such as password composition policies [2], [13], [17], develop password strength meters [18] and evaluate the effectiveness of key-stretching mechanisms against offline attacks [4]. However, to the best of our knowledge the problem of providing rigorous confidence intervals for the estimated guessing numbers has not been explored.[3] Blocki and Liu [19] recently focused on the problem of upper/lower bounding the guessing curve of a perfect knowledge attacker who *knows* the user password distribution $\mathcal{P}$. While this can be a useful goal, in practice it can also be useful to characterize the guessing curve of an attacker following a state of the art password cracking model M since a real world attacker will not have perfect knowledge of the user password distribution. We also note that the upper/lower bounds of Blocki and Liu [19] for the guessing curve of a perfect knowledge attacker rapidly diverge even for moderately large values of B e.g., $B = 10^7$. By contrast, we are able to obtain relatively tight upper/lower bounds on the guessing curve of an attacker using model M even when the guessing budget B is very large e.g., $B = 10^{24}$.

II. Background

In this section we introduce probabilistic password models, regular Monte Carlo Estimation and password guessing curves more formally. A summary of notations is found in Table III in Appendix F.

Probabilistic Password Guessing Model. In this work we assume that our attacker is untargeted and that the attacker uses a Probabilistic Password Model M to crack passwords. To apply regular Monte Carlo or Confident Monte Carlo we make several assumptions about the model M. First, we assume that the model M implicitly defines a distribution $\mathcal{M}$ over passwords and that M allows us to efficiently sample from the distribution $\mathcal{M}$. Second, given an arbitrary password pwd we assume that we can efficiently compute $p^M_{pwd} \doteq \Pr_{x \leftarrow \mathcal{M}}[x = pwd]$ i.e., likelihood of the password pwd according to our distribution $\mathcal{M}$. We note that these assumptions hold for most sophisticated password cracking models such as Probabilistic Context-Free Grammars [7]–[9], Markov models [10]–[13], and neural networks [2].

It will be convenient to let $pwd_1, pwd_2, \ldots$ denote the list of passwords in the support of our distribution $\mathcal{M}$ and to let $p^M_i \doteq p^M_{pwd_i}$ denote the probability of password i. It will also be convenient to assume that these passwords are ordered such that $p^M_1 \geq p^M_2 \geq \ldots$ i.e., so that an attacker using model M would check guesses in the order $pwd_1, pwd_2, \ldots$. We let $G(pwd)$ denote the number of guesses that an attacker, following model M, would need to attempt in order to crack the password pwd i.e., $G(pwd_i) = i$.

Given a probability value $q \in [0, 1]$ we would like to define $G(q)$ as the hypothetical guessing number for a password pwd with probability $p^M_{pwd} = q$. However, if there are multiple passwords in $\mathcal{M}$ with probability exactly q there will be multiple different values of the guessing number. To avoid ambiguity we instead define an exclusive bound

[3] Melicher et al. [2] mention that with at least one million samples typically they observe "95% confidence intervals of less than 10% of the value of the guess-number estimate" and "passwords for which the error exceeded 10% tended to be guessed only after more than 10^{18} guesses" in their experiments. However, the paper does not contain any details about these claimed confidence intervals or the methodology by which they were derived. We reached out the the authors to provide clarification, but received no response.

$G^{\texttt{EX}}(q) := |\{i : p_i^M > q\}|$ to count the number of passwords with probability *strictly greater than* q and an inclusive bound $G^{\texttt{IN}}(q) := |\{i : p_i^M \geq q\}|$ to count the number of passwords with probabilities *greater than or equal to* q. Observe that for a password pwd with probability p_{pwd}^M we have $G^{\texttt{EX}}(q)+1 \leq G(pwd) \leq G^{\texttt{IN}}(q)$. It will sometimes be convenient to write $G^{\texttt{EX}}(pwd) \doteq G^{\texttt{EX}}(p_{pwd}^M)$ or $G^{\texttt{IN}}(pwd) \doteq G^{\texttt{IN}}(p_{pwd}^M)$.

Regular Monte Carlo Estimation. To compute $G(pwd)$ (or $G^{\texttt{EX}}(q)$ or $G^{\texttt{IN}}(q)$) exactly a defender would need to enumerate all possible passwords in M whose probability is above a given threshold (p_{pwd}^M). This can be prohibitively expensive for the defender when the guessing number is large. Thus, Dell'Amico and Filippone [1] developed a Monte Carlo algorithm to efficiently estimate $G(pwd)$. More accurately, for any probability value q their algorithm produces an unbiased estimate of $G^{\texttt{EX}}(q)$ — we have $G(pwd) = G^{\texttt{EX}}(q)+1$ in whenever there is a unique password pwd with probability $p_{pwd}^M = q$. This regular Monte Carlo algorithm works as follows: (1) draw k iid samples from the distribution $\mathcal{M}$ i.e., $S \leftarrow \mathcal{M}^k$, (2) output the estimate $\hat{G}_S^{\texttt{EX}}(q) = \frac{1}{k}\sum_{x\in S, p_x^M > q} \frac{1}{p_x^M}$. In practice, Dell'Amico and Filippone [1] proposed that one could draw the sample S ahead of time and use this sample to obtain out strength estimate $\hat{G}_S^{\texttt{EX}}(pwd)$ for multiple different passwords.

Dell'Amico and Filippone [1] proved that for any probability parameter $q \in [0,1]$ the expectation of the estimation is equal to its true value, i.e., $\mathbb{E}(\hat{G}_S^{\texttt{EX}}(q)) = G^{\texttt{EX}}(q)$. They also argued that the variance $\mathrm{Var}(\hat{G}_S^{\texttt{EX}}(q)) = \frac{1}{k}(\sum_{i:p_i^M>q} \frac{1}{p_i^M} - G^{\texttt{EX}}(q))^2$ converges to 0 as the sample size k gets to infinite. However, in practice the sample size k is finite and can be very small compared to $\frac{1}{q}$.

Similarly, as a trivial extension of [1] one can define $\hat{G}_S^{\texttt{IN}}(q) := \frac{1}{k}\sum_{x\in S, p_x^M \geq q} \frac{1}{p_x^M}$ as an unbiased estimate for our inclusive term $G^{\texttt{IN}}(q)$.

Password Guessing Curve. Given a dataset D[4] we let $\lambda_{M,B,D} \doteq \frac{1}{|D|}|\{x \in D \ : \ G(x) \leq B\}|$ denote the fraction of passwords in D that would be cracked within B guesses by an untargeted attacker following model M. Similarly, we define $\lambda_{M,B} := \Pr_{y\leftarrow\mathcal{P}}[G(y) \leq B]$ to be the probability that a random password y sampled from $\mathcal{P}$ would be cracked within B guesses. We will typically assume that the user password distribution $\mathcal{P}$ is unknown, but that we are given a dataset D consisting of iid samples from $\mathcal{P}$. In this case we have $\lambda_{M,B} = \mathbb{E}[\lambda_{M,B,D}]$ where the randomness is taken over the selection of D from the unknown distribution $\mathcal{P}$.

For an online attacker B is usually small since an authentication service can lock out the user account after several failed login attempts. For an offline attacker B can be much larger since with the stolen (salted) cryptographic hash of the user's password an offline attacker can check as many passwords as s/he wants by comparing the (salted) cryptographic hash with the hashes of the top B guesses $pwd_1, \ldots, pwd_B$. An offline attacker is limited only by the resources s/he is willing to invest cracking and by the cost of repeatedly evaluating the password hash function.

[4] D may contain duplicated passwords since different users might select the same password.

III. Limitations of Regular Monte Carlo Strength Estimation

In this section we discuss the limitations of the regular Monte Carlo strength estimation [1]. We provide a theoretical example of password models where regular Monte Carlo will dramatically underestimate the guessing number with high probability, and another example of a password model where Monte Carlo will overestimate the guessing number by a factor of 2 with probability at least 0.3. Specifically, for any sample size k we define a model for which the estimation is inaccurate with a significant probability. We also provide a theoretical example of a password model and a password distribution where the regular Monte Carlo estimation has a large error on predicting the guessing curve. We further demonstrate that the above issue is inherent to any blackbox method for estimation given only a finite amount of samples without exploiting specific properties of the password guessing model. All examples of password models and distributions in this section are constructed for the purpose of illustration. We are *not* claiming that they are representative of distributions/models one would encounter in practice.

A. Error on Guessing Number

1) Underestimating the Guessing Number: We first provide an example of a password model where Monte Carlo will (whp) dramatically underestimate the true guessing number.

The Model/Distribution $\mathcal{M}$: Consider a model M which induces a distribution $\mathcal{M}$ over passwords $pwd_1, pwd_2, \ldots$ such that $\Pr_{x\leftarrow\mathcal{M}}[x = pwd_i] = 2^{-i}$.

Actual Guessing Number: The actual guessing number of each password pwd_i is $G(y) = i$.

Analysis of Monte Carlo Estimate: Suppose that we apply Monte Carlo with $k = |S| \geq 2^{10}$ iid samples $S \leftarrow \mathcal{M}^k$ from $\mathcal{M}$ to estimate the guessing number of a password. For any $i > 2\log(k)$ with high probability $(1-2^{-i+1})^k > 0.99$ our sample set S will contain *no passwords* with probability equal to or less than 2^{-i}. In this case for *any* $j \geq i$ our estimated guessing number for pwd_j is $\hat{G}_S^{\texttt{EX}}(pwd_j) = \frac{1}{k}\sum_{x\in S}\frac{1}{p_x^M}$ where $\frac{1}{k}\sum_{x\in S}\frac{1}{p_x^M} \leq \frac{2^{\log(k^2)}}{k} = k$. Thus, if $j \gg k$ Monte Carlo will *dramatically underestimate* the true guessing number with high probability.

The authors of [1] state that the variance of $\hat{G}_S^{\texttt{EX}}(pwd_j)$ approaches 0 as k increases. However, in the above example the variance for the jth most probable password pwd_j is $Var(\hat{G}_S^{\texttt{EX}}(pwd_j)) = \frac{1}{k}(\sum_{t=1}^{j-1}\frac{1}{2^{-t}} - (j-1)^2) = \frac{1}{k}(2^j - 1 - (j-1)^2)$. While the variance does decrease linearly with the sample size k it also increases exponentially with the true guessing number j.

2) Overestimating the Guessing Number: We now give an example where regular Monte Carlo estimation can *overestimate* the guessing number by a factor of ≈ 2 with non-negligible probability (e.g., 0.3).

The Model/Distribution $\mathcal{M}$: Fix the sample size $k \geq 5$ and consider a model M corresponding to a password distribution $\mathcal{M}$ where there are $n_1 = 2k-1$ passwords with probability $p_1 = \frac{1}{2k}$, $n_2 = \frac{k^2}{2} - 1$ passwords with probability $p_2 = \frac{1}{k^3}$, and $n_i = 1$ password with probability $p_i = \frac{1}{2^{i-2}k^3} < p_{i-1}$ for any $i \geq 3$. Observe that $\sum_{i \geq 1} n_i p_i = 1$.

Actual Guessing Number: The actual guessing number of the password with probability p_3 is $G(p_3) = n_1 + n_2 + 1 = 2k - 1 + \frac{k^2}{2}$.

Analysis of Monte Carlo Estimate: Suppose $|S| = k$ ($k \geq 5$) samples are generated to estimate the guessing numbers. Consider the event E that S contains exactly one password with probability p_2 and all the remaining $(k-1)$ samples have probability p_1 (i.e., none of the remaining samples in S have probability smaller than p_2). If the event E occurs then the estimated guessing number of password with probability p_3 is $\hat{G}_S^{\texttt{EX}}(p_3) + 1 = \frac{1}{k}((k-1)\frac{1}{p_1} + \frac{1}{p_2}) + 1 = 2k - 1 + k^2$ while the actual guessing number is $G(p_3) = n_1 + n_2 + 1 = 2k - 1 + \frac{1}{2}k^2$ i.e., we overestimate the guessing number by a factor $\frac{\hat{G}_S^{\texttt{EX}}(p_3)+1}{G(p_3)} \approx 2$. The probability of the event E is $n_2\binom{k}{1}p_2(n_1p_1)^{k-1} = (\frac{1}{2} - \frac{1}{k^2})(1 - \frac{1}{2k})^{k-1} > 0.3$.

3) Inherent Limitations of Any *Blackbox Guessing Number Estimate:* We now argue that *any* blackbox method for estimating the guessing number will have similar issues. In particular, for any sample size k and any blackbox strength estimation method we can find cases where the model will (whp) give us highly inaccurate strength estimates. As a concrete example consider the model M defined in Section III-A1 i.e., the model induces a distribution $\mathcal{M}$ over passwords $pwd_1, pwd_2, \ldots$ such that $\Pr_{x \leftarrow \mathcal{M}}[x = pwd_j] = 2^{-j}$ for each $j \geq 1$. Now let us define a model M' inducing a distribution $\mathcal{M}'$ similar to $\mathcal{M}$ except that we replace the i passwords $pwd_{i+1}, \ldots, pwd_{2i}$ with an exponentially large subset of $2^i - 1$ new passwords of probability 2^{-2i}. In particular, we have $\Pr_{x \leftarrow \mathcal{M}'}[x = pwd_j] = \Pr_{x \leftarrow \mathcal{M}}[x = pwd_j] = 2^{-j}$ for all $j \leq i$ and all $j \geq 2i+1$. In model M' the passwords $pwd_{i+1}, \ldots, pwd_{2i}$ are replaced with $2^i - 1$ new passwords $pw_1, \ldots, pw_{2^i-1}$. For each $j \leq 2^i - 1$ we set $\Pr_{x \leftarrow \mathcal{M}'}[x = pw_j] = 2^{-2i}$ so that $\sum_{j=1}^{2^i-1} \Pr_{x \leftarrow \mathcal{M}'}[x = pw_j] = 2^{-i}\left(1 - 2^{-i}\right) = \sum_{j=i+1}^{2i} \Pr_{x \leftarrow \mathcal{M}}[x = pw_j]$.

We now make several observations about the models M and M'. First, we note that if we draw $k = o\left(2^i\right)$ samples in a blackbox manner then we will not even be able to distinguish between the distributions $\mathcal{M}$ and $\mathcal{M}'$ (whp).[5] Second, we note that, when $j \geq 2i+1$, the actual guessing number for pwd_j is very different under model M and M' due to the presence/absence of the exponentially large set of passwords $\{pw_1, \ldots, pw_{2^i-1}\}$ which would be guessed before pwd_j. In particular, the actual guessing number number is j (resp. $2^i - 1 + j - i$) under model M (resp. M'). If we cannot even distinguish between distributions $\mathcal{M}$ and $\mathcal{M}'$ then, for $j \geq 2i+1$, we cannot rule out the possibility that the actual guessing number for pwd_j could be as small as j (model M) or as large as $2^i - 1 + j - i$ (model M'). Thus, providing a concrete strength estimate without providing any indication that there is a high degree of uncertainty would be particularly problematic in this example.

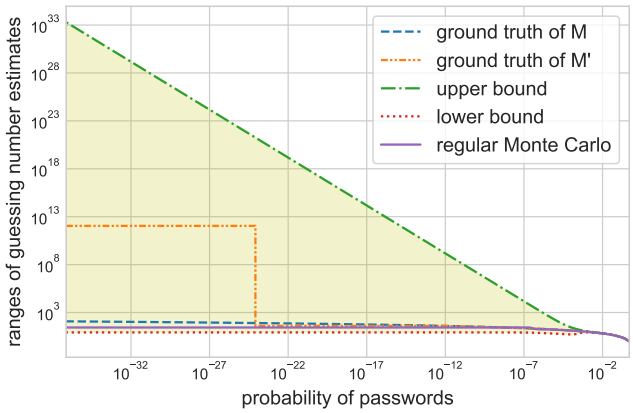


Fig. 1: Blackbox Limitations (Example)

We visualize these results in Figure 1 where we set $k = 2^{20}$ and $i = 2\log k = 40$. We also show the upper/lower bounds obtained using our Confident Monte Carlo approach ($\geq 98\%$ confidence) as well as the estimates generated by regular Monte Carlo. We note that the regular Monte Carlo estimates are reasonable accurate for model M, but highly inaccurate for the model M'. Because Confident Monte Carlo is also blackbox we cannot hope to achieve tight upper/lower bounds. In particular, observe that any accurate blackbox lower bound for the guessing number of pwd_{2i+1} *must* be at least $2i+1$ (ground truth under model M) and *at least* 2^i+i (ground truth under model M'). This is precisely what we observe in Figure 1 e.g., for passwords with probability 10^{-37} the gap between our upper/lower bounds is $\approx 10^{33}$. In particular, Confident Monte Carlo accurately indicates that we are highly uncertain about the real guessing number.

B. Error on Guessing Curve

1) Regular Monte Carlo Guessing Curves: Another application of Monte Carlo strength estimation is generating (an approximation of) the attacker's guessing curve. For example, we might like to estimate the probability $\lambda_{M,B}$ that an attacker can crack a random user's password (sampled from potentially unknown distribution $\mathcal{P}$) within B guesses using model M or, given a dataset D of user passwords we might want to estimate $\lambda_{M,B,D}$ — the fraction of passwords in D that the attacker using model M can crack within B guesses per user account. The standard (heuristic) way to approximate $\lambda_{M,B,D}$ and $\lambda_{M,B}$ is to simply compute $\hat{\lambda}_{M,B,D} := \frac{1}{|D|}|\{y \in D : \hat{G}^{\texttt{EX}}(y) \leq B\}|$. This heuristic approach has been widely adopted in the password literature (e.g., [2]–[5]). However, for any guessing budget B and sampling parameter k we can provide an example of a model M and a distribution $\mathcal{P}$

[5]To see this let E denote the event that we never sample a password y with probability $p_y < 2^{-i}$. Observe that if we condition on the event E that we only sample passwords from $pwd_1, \ldots, pwd_i$ then the two distributions $\mathcal{M}$ and $\mathcal{M}'$ are equivalent. Furthermore, by union-bounds we have $\Pr[\overline{E}] \leq k2^{-i} = o(1)$.

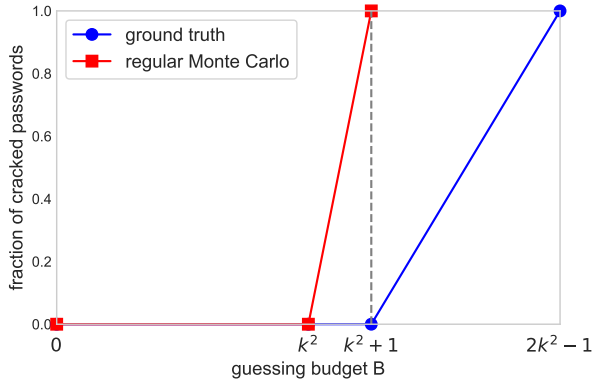


Fig. 2: Error on Monte Carlo Estimated Guessing Curve

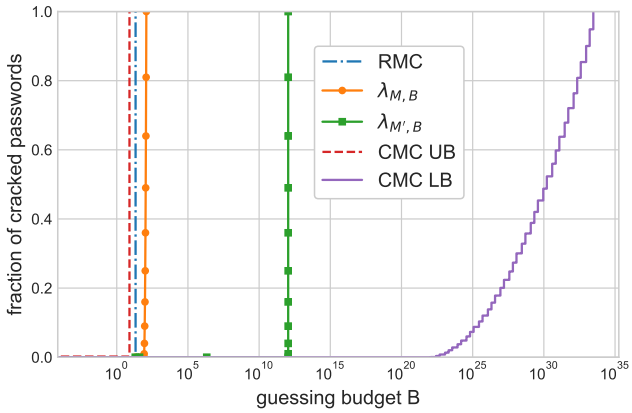


Fig. 3: Limitations: Blackbox Guessing Curve Estimate

over user passwords such that: (1) an attacker who makes B guesses will never crack any user password sampled from $\mathcal{P}$ i.e., $\lambda_{M,B} = 0$, and (2) with high probability $\hat{\lambda}_{M,B,D} = 1$ i.e., the widely adopted heuristic using regular Monte Carlo strength estimation predicts that the attacker cracks 100% of passwords within B guesses.

Model Distribution $\mathcal{M}$: Consider a distribution $\mathcal{M}$ generated by a cracking model M where there are $k^2 - 1$ passwords $pwd_1, \ldots, pwd_{k^2-1}$ with probability $\frac{1}{k^2}$. The remaining k^2 passwords $pwd_{k^2}, \ldots, pwd_{2k^2-1}$ in the support of our distribution satisfy the following properties: (1) $\sum_{i=k^2}^{2k^2-1} \Pr_{x \leftarrow \mathcal{M}}[x = pwd_i] = \frac{1}{k^2}$ and (2) $\Pr_{x \leftarrow \mathcal{M}}[x = pwd_i] > \Pr_{x \leftarrow \mathcal{M}}[x = pwd_{i+1}]$ for each $k^2 - 1 \leq i < 2k^2 - 1$ i.e., passwords are listed in descending order of probability. Let $B = k^2 + 1$, let $F = \{pwd_1, \ldots, pwd_B\}$ denote the most popular B passwords and let $L = \{pwd_{B+1}, \ldots, pwd_{k^2-1}\}$ denote the remaining passwords.

Actual Password Distribution $\mathcal{P}$: For the actual user password distribution we can consider any distribution $\mathcal{P}$ with support L i.e., $\Pr_{x \leftarrow \mathcal{P}}[x \in F] = 0$.

Actual Guessing Curve: Notice that the attacker's first B guesses will all be from the set F. By construction, the support of our user password distribution $\mathcal{P}$ does not include any password from F. Thus, for any dataset D of passwords sampled from $\mathcal{P}$ we will have $\lambda_{M,B,D} = 0$ and we also have $\lambda_{M,B} = \mathbb{E}[\lambda_{M,B,D}] = 0$.

Analysis of Monte Carlo Curve: Given a sample set S with size k from $\mathcal{M}$, except with probability *at most* $\frac{1}{k}$, all k samples are from $pwd_1, \ldots, pwd_{k^2-1}$. In this case the estimated guessing number for *any* password pwd_i with $k^2 - 1 \leq i \leq 2k^2 - 1$ is $\hat{G}_S^{\text{EX}}(pwd_i) + 1 = 1 + \frac{1}{k} \sum_{x \in S} k^2 = 1 + k^2 \leq B$. Thus, with probability at least $1 - \frac{1}{k}$, we have $\hat{\lambda}_{M,B,D} = 1$ meaning that this widely adopted heuristic incorrectly predicts that the attacker will crack 100% of passwords in our dataset D. We compare the (regular) Monte Carlo estimated guessing curve with the actual (ground truth) guessing curve in Figure 2.

2) Inherent Limitations of Any *Blackbox Guessing Curve Estimate:* Let us reconsider the models M and M' as defined in Section III-A3. We will define a user password distribution $\mathcal{P}$ such that (1) any *blackbox* method given samples from $\mathcal{P}$ in addition to samples from either $\mathcal{M}$ or $\mathcal{M}'$ *still* cannot distinguish between the models M and M' i.e., $\Pr_{x \leftarrow \mathcal{M}'}[x = pwd] = \Pr_{x \leftarrow \mathcal{M}'}[x = pwd]$ for all passwords pwd in the support of $\mathcal{P}$ and (2) the guessing curves $\lambda_{\mathcal{M}',B}$ and $\lambda_{\mathcal{M},B}$ are dramatically different.

User Password Distribution: Define the actual user password distribution $\mathcal{P}$ to be the uniform distribution over $\{pwd_{2i+1}, \ldots, pwd_{3i}\}$ i.e., for all $j \in [2i+1, 3i]$ we have $\Pr_{x \leftarrow \mathcal{P}}[x = pwd_j] = \frac{1}{i}$, $\Pr_{x \leftarrow \mathcal{P}}[x = pwd_j] = 0$ for $j \notin [2i+1, 3i]$ and $\Pr_{x \leftarrow \mathcal{P}}[x = pw_j] = 0$ for $j \leq 2^i - 1$. Recall that for all $j \in [2i+1, 3i]$ that we have $\Pr_{x \leftarrow \mathcal{M}'}[x = pwd_j] = 2^{-j} = \Pr_{x \leftarrow \mathcal{M}'}[x = pwd_j]$. Thus, it will not be possible to distinguish the models M and M' using samples from $\mathcal{P}$.

Actual Guessing Curves: Under model M' the guessing number for any password pwd_j in the support of $\mathcal{P}$ is *at least* $2^i + i$. Thus, $\lambda_{M',B} = 0$ for all $B \leq 2^i + i$. By contrast, under model M the guessing number for any password pwd_j in the support of $\mathcal{P}$ is *at most* $3i$ so we will have $\lambda_{M,B} = 1$ for all $B \geq 3i$.

We visualize these results in Figure 3 where we set $k = 2^{20}$ and $i = 2 \log k = 40$. The orange and green curves plot the ground truth guessing curves $\lambda_{M,B}$ and $\lambda_{M',B}$ for the models M and M' respectively. The blue curve shows the heuristic guessing curve estimate obtained using Regular Monte Carlo (RMC). While this curve is reasonably close to $\lambda_{M,B}$, it quite far from $\lambda_{M',B}$. Thus, if the real password cracking model is M' the RMC curve is highly inaccurate. Finally, the red (resp. purple) curves plot the best upper (resp. lower) bounds obtained using Confident Monte Carlo (CMC) — several rigorous statistical techniques we introduce later in the paper to upper/lower bound the guessing number and the guessing curve with high confidence (we used 99% confidence in Figure 3). We note that both Regular and Confident Monte Carlo utilize the model in a blackbox manner and cannot distinguish the distributions $\mathcal{M}$ or $\mathcal{M}'$. Thus, we do not label the CMC or RMC curves with the model M or M' since the results will be the same (whp). Observe that the curves $\lambda_{M,B}$ and $\lambda_{M',B}$

both lie between our upper/lower bounds. Thus, Confident Monte Carlo accurately indicates that there is a high degree of uncertainly about the true guessing curve.

IV. Theoretical Bounds on Guessing Number

In this section, we present two techniques to rigorously bound the actual guessing number $G(y)$ of an arbitrary password y using iid password samples randomly selected from a model distribution $\mathcal{M}$ generated by a password cracking model M. In particular, we can ensure that the actual guessing number $G(y)$ is sandwiched between our upper/lower bounds with high confidence (e.g., 99%) allowing us to quantify the uncertainty of guessing number estimates.

A. Upper/Lower Bounds Using Hoeffding's Inequality

In this section, we prove an upper bound and a lower bound on the actual guessing number $G(q)$ of a password with probability q using Hoeffding's inequality. The formal statement is shown below:

Theorem 1. *Given a set S with k iid password samples sampled from distribution $\mathcal{M}$, for any probability $q \in [0,1]$ and any parameter $\epsilon \geq 0$, we have:*

$$\Pr[G(q) \leq \hat{G}_S^{\texttt{IN}}(q) + \epsilon/q] \geq 1 - \exp\left(-2k\epsilon^2\right)$$
$$\Pr[G(q) \geq \hat{G}_S^{\texttt{EX}}(q) + 1 - \epsilon/q] \geq 1 - \exp\left(-2k\epsilon^2\right)$$

where the randomness is taken over $S \leftarrow \mathcal{M}^k$.

Due to space limitations the formal proof is deferred to Appendix E. Briefly, we observe that $\hat{G}_S^{\texttt{EX}}(q)$ can be viewed as the sum of k independent random variables $X_1^{\texttt{EX}}, \ldots, X_k^{\texttt{EX}}$ where $X_i^{\texttt{EX}} = 0$ if the ith password z from our sample has probability $p_z^M < q$ and $X_i^{\texttt{EX}} = 1/p_z^M$ otherwise. Since the random variable are independent we can apply Hoeffding's inequality to upper bound the probability that our estimate $\hat{G}_S^{\texttt{EX}}(q)$ is significantly smaller than its expectation $\mathbb{E}\left[\hat{G}_S^{\texttt{EX}}(q)\right] = G_S^{\texttt{EX}}(q)$ where $G(q) \geq G_S^{\texttt{EX}}(q)+1$. Similarly, we can apply Hoeffding's inequality to bound $\hat{G}_S^{\texttt{IN}}(q)$. In particular, for $\phi \in \{\texttt{EX}, \texttt{IN}\}$ we have $|\hat{G}_S^{\phi}(q) - G^{\phi}(q)| \leq \epsilon/q$ with any $\epsilon \geq 0$ as below:

$$\Pr[G^{\phi}(q) \leq \hat{G}_S^{\phi}(q) + \epsilon/q] \geq 1 - \exp\left(-2k\epsilon^2\right) \quad (1)$$
$$\Pr[G^{\phi}(q) \geq \hat{G}_S^{\phi}(q) - \epsilon/q] \geq 1 - \exp\left(-2k\epsilon^2\right) \quad (2)$$

Note that given a password y the upper/lower bounds in Theorem 1 differ by an additive factor of ϵ/p_y^M. Thus, we can obtain tight upper/lower bounds $\epsilon/p_y^M \ll G(y)$ although the bounds can diverge as p_y^M grows small i.e., when the password is particularly rare.

B. A Tighter Lower Bound For Rare Passwords

The upper and lower bounds in Section IV-A will diverge when the password is rare. In the worst case, the lower bound will be useless if $\hat{G}_S^{\texttt{EX}}(q) + 1 - \epsilon/q$ becomes a negative value. Is it possible to derive tighter bounds on the guessing numbers of rare passwords? In this section we further tighten the lower bound for rare passwords by applying Markov's inequality and taking the median estimates. In particular, for any password probability $q \in [0,1]$ we define $\hat{G}^{\phi}_{\mathbb{S},med}(q) = \text{median}\left(\{\hat{G}^{\phi}_{S_i}(q)\}_{1 \leq i \leq n}\right)$ where $\mathbb{S} = \{S_1, S_2, \ldots, S_n\}$ and each S_i contains k independent samples from our model. Fixing any parameter $\delta < \frac{1}{2}$ we show that (whp) $\delta \cdot \hat{G}^{\texttt{EX}}_{\mathbb{S},med}(q)$ lower bounds the true guessing number as stated in the following theorem:

Theorem 2. *For any password probability $q \in [0,1]$, and any parameters $0 \leq \delta \leq \frac{1}{2}$ and $0 \leq \epsilon \leq \frac{1}{2} - \delta$,*

$$\Pr[G(q) \geq \delta \cdot \hat{G}^{\texttt{EX}}_{\mathbb{S},med}(q) + 1] \geq 1 - \exp(-2n\epsilon^2)$$

where the randomness is taken over n sets of k Monte Carlo samples $\mathbb{S} = \{S_1, \cdots, S_n\}$ from model M.

Theorem 2 often allows us to tighten the lower bounds on the guessing number of rare passwords. Due to space limitations the formal proof is deferred to Appendix E. Intuitively, we can define an indicator random variable X_i^{ϕ} for $\phi \in \{\texttt{EX}, \texttt{IN}\}$ such that $X_i^{\phi} = 1$ if and only if $G^{\phi}(q) \geq \delta \cdot \hat{G}^{\phi}_{S_i}(q)$, i.e., if and only if the ith sample set S_i overestimates $G^{\phi}(q)$ by at most a factor of $1/\delta$, and $X_i^{\phi} = 0$ otherwise. As long as $\sum_{i=1}^n X_i^{\phi} \geq n/2$ we are guaranteed that the median estimate is not too bad and that $G(q) \geq \delta \cdot \hat{G}^{\texttt{EX}}_{\mathbb{S},med}(q) + 1$. Thus, it suffices to upper bound the probability that $\sum_{i=1}^n X_i^{\phi} \geq n/2$. We first apply Markov's inequality to show that $\Pr[X_i^{\phi} = 1] \geq 1 - \delta$. Finally, since each X_i^{ϕ} is independent we can apply concentration bounds to show that $\sum_{i=1}^n X_i^{\phi} \geq n/2$ with high probability. In particular, by Chernoff bounds for any $0 \leq \epsilon \leq \frac{1}{2} - \delta$ we have

$$\Pr[\hat{G}^{\phi}_{\mathbb{S},med}(q) \leq \frac{1}{\delta} G^{\phi}(q)] \geq \Pr[\sum_{i=1}^{n} X_i^{\phi} \geq \frac{n}{2}]$$
$$\geq \Pr[\sum_{i=1}^{n} X_i^{\phi} \geq n(1 - \delta - \epsilon)] \geq 1 - \exp(-2n\epsilon^2). \quad (3)$$

V. Theoretical Bounds on Guessing Curve

So far our focus has been on upper/lower bounding the guessing number of a particular user password against a cracking model M. However, in some defensive applications our goal will be to upper/lower bound the attacker's entire guessing curve e.g., to determine whether or not a password policy intervention results in a password distribution that is harder for the adversary to crack. More formally, in this section we develop techniques to upper/lower bound the curves $\lambda_{M,B,D}$ and $\lambda_{M,B}$ as the guessing budget B varies from small to large. Recall that $\lambda_{M,B,D}$ denotes the fraction of passwords in dataset D that would be cracked within B guesses, and that $\lambda_{M,B}$ denotes the probability that randomly sampled password would be cracked within B guesses.

Given a dataset D of independent samples from an *unknown* password distribution, our first observation is that the expected value of $\lambda_{M,B,D}$ (over the random selection of D) is simply $\lambda_{M,B}$ and, if D is large enough, the random variable $\lambda_{M,B,D}$ is tightly concentrated around its mean — see Theorem 3. Given this result our main task will be to develop high confidence

upper/lower bounds on $\lambda_{M,B,D}$ which will immediately yield high-confidence upper/lower bound for $\lambda_{M,B}$ as a corollary. Thus, we will focus primarily on bounding $\lambda_{M,B,D}$ in the remainder of this section. Theorem 3 follows directly from McDiarmid's inequality [20]. The formal proof is deferred to Appendix E.

Theorem 3. *For any guessing number $B \geq 0$ and any $0 \leq \epsilon \leq 1$, we have:*

$$\Pr[\lambda_{M,B} \geq \lambda_{M,B,D} - \epsilon] \geq 1 - \exp(-2|D|\epsilon^2), \quad \textit{and}$$
$$\Pr[\lambda_{M,B} \leq \lambda_{M,B,D} + \epsilon] \geq 1 - \exp(-2|D|\epsilon^2)$$

where the randomness is taken over the sample set $D \leftarrow \mathcal{P}^{|D|}$.

A. The General Framework

In this section, we propose a generalized framework for converting confident upper/lower bounds on guessing numbers into confident upper and lower bounds on $\lambda_{M,B,D}$ (and by extension $\lambda_{M,B}$). Suppose that $G(q)$ denotes the guessing number for a password pwd whose probability (according to our model) is q — for simplicity of exposition let us first suppose that there is only one such password with probability exactly q. Although $B = G(q)$ is unknown we observe that it is still possible compute the quantity $\lambda_{M,B,D}$ i.e., by computing the fraction of passwords in D (i.e., $pw \in D$) whose probability is $p_{pw}^M \geq q$. Unfortunately, this is still not sufficient to plot the curve $\lambda_{M,B,D}$ since we do not actually know the value of B. However, if we are given upper/lower bounds $L \leq B \leq U$ then we can use the value of $\lambda_{M,B,D}$ as an upper bound for $\lambda_{M,L,D}$ and as a lower bound for $\lambda_{M,U,D}$ since we know that $\lambda_{M,L,D} \leq \lambda_{M,B,D} \leq \lambda_{M,U,D}$.

Our key idea is to pick a sequence $q_1, q_2, \ldots, q_\ell$ of ℓ probability mesh points and obtain upper (resp. lower) bounds $U_1, \ldots, U_\ell$ (resp. $L_1, \ldots, L_\ell$) on the corresponding guessing numbers. Intuitively, as long as all of our upper (resp. lower) bounds are valid we can use them to lower (resp. upper) bound the guessing curve $\lambda_{M,B,D}$ at multiple points $B \in \{U_1, \ldots, U_\ell\}$ (resp. $B \in \{L_1, \ldots, L_\ell\}$).

The formal framework is slightly complicated by the fact that we will occasionally have multiple passwords with the same probability q according to model M. However, we can deal with this concern by lower (resp. upper) bounding the exclusive (resp. inclusive) guessing numbers. Recall that for any $\phi = \{\texttt{EX}, \texttt{IN}\}$ equations (1), (2) and (3) in Section IV provide high confidence upper and lower bounds on $G^\phi(q)$. In general, for any probability $0 \leq q \leq 1$ we define $\mathsf{UB}_{G^\phi,|S|}(q, S)$ (resp. $\mathsf{LB}_{G^\phi,|S|}(q, S)$) to be an arbitrary upper (resp. lower) bound of $G^\phi(q)$ with error rate $\mathsf{ERR}(\mathsf{UB}_{G^\phi,|S|})$ (resp. $\mathsf{ERR}(\mathsf{LB}_{G^\phi,|S|})$), i.e., with randomness taken over the selection of sample set S from model M we have:

$$\Pr[G^\phi(q) \geq \mathsf{LB}_{G^\phi,|S|}(q, S)] \geq 1 - \mathsf{ERR}(\mathsf{LB}_{G^\phi,|S|}) \quad (4)$$
$$\Pr[G^\phi(q) \leq \mathsf{UB}_{G^\phi,|S|}(q, S)] \geq 1 - \mathsf{ERR}(\mathsf{UB}_{G^\phi,|S|}) \quad (5)$$

Formally, we define $\hat{\lambda}^{ub}_{M,B,D,S}$ and $\hat{\lambda}^{lb}_{M,B,D,S}$ as below:

$$\hat{\lambda}^{ub}_{M,B,D,S} := \min_{1 \leq i \leq \ell, B \leq \mathsf{LB}_{G^{\texttt{IN}},|S|}(q_i,S)} \left(\lambda_{M,G^{\texttt{IN}}(q_i),D}\right), \quad (6)$$

$$\hat{\lambda}^{lb}_{M,B,D,S} := \max_{1 \leq i \leq \ell, B \geq \mathsf{UB}_{G^{\texttt{EX}},|S|}(q_i,S)} \left(\lambda_{M,G^{\texttt{EX}}(q_i),D}\right). \quad (7)$$

For completeness, we set our upper (res. lower) bound $\hat{\lambda}^{ub}_{M,B,D,S} = 1$ (resp. $\hat{\lambda}^{lb}_{M,B,D,S} = 0$) if $B > \max_{1 \leq i \leq \ell} \{\mathsf{LB}_{G^{\texttt{IN}},|S|}(q_i, S)\}$ (resp. $B < \min_{1 \leq i \leq \ell} \{\mathsf{UB}_{G^{\texttt{EX}},|S|}(q_i, S)\}$).

Theorem 4 shows that $\hat{\lambda}^{ub}_{M,B,D,S}$ (resp. $\hat{\lambda}^{lb}_{M,B,D,S}$) upper (resp. lower) bounds the value of $\lambda_{M,B,D}$ with high confidence. Intuitively, the error terms $\ell \cdot \mathsf{ERR}(\mathsf{LB}_{G^{\texttt{IN}},|S|})$ and $\ell \cdot \mathsf{ERR}(\mathsf{UB}_{G^{\texttt{EX}},|S|})$ are obtained by taking union bounds.

Theorem 4. *Given a password dataset D containing iid samples from distribution $\mathcal{P}$ and a sequence of probability mesh points $Q = \{q_1, \cdots, q_\ell\}$, we have:*

$$\Pr\left[\forall B \geq 0, \lambda_{M,B,D} \leq \hat{\lambda}^{ub}_{M,B,D,S}\right] \geq 1 - \ell \cdot \mathsf{ERR}(\mathsf{LB}_{G^{\texttt{IN}},|S|})$$
$$\Pr\left[\forall B \geq 0, \lambda_{M,B,D} \geq \hat{\lambda}^{lb}_{M,B,D,S}\right] \geq 1 - \ell \cdot \mathsf{ERR}(\mathsf{UB}_{G^{\texttt{EX}},|S|})$$

where the randomness is taken over the sample set S from model M.

If we assume that our dataset D is sampled iid from our unknown password distribution we can apply Theorem 3 to upper/lower bound $\lambda_{M,B}$ as an immediate corollary of Theorem 4 — see Corollary 5.

Corollary 5. *Given a password distribution $\mathcal{P}$ and a sequence of probability mesh points $Q = \{q_1, \cdots, q_\ell\}$, for any guessing number $B > 0$ and any parameters $0 \leq \epsilon \leq 1$, we have:*

$$\Pr\left[\lambda_{M,B} \leq \hat{\lambda}^{ub}_{M,B,D,S} + \epsilon\right] \geq 1 - \ell \cdot \mathsf{ERR}(\mathsf{LB}_{G^{\texttt{IN}},|S|}) - \exp(-2|D|\epsilon^2)$$
$$\Pr\left[\lambda_{M,B} \geq \hat{\lambda}^{lb}_{M,B,D,S} - \epsilon\right] \geq 1 - \ell \cdot \mathsf{ERR}(\mathsf{UB}_{G^{\texttt{EX}},|S|}) - \exp(-2|D|\epsilon^2)$$

where the randomness is taken over the sample set S from model M and the dataset $D \leftarrow \mathcal{P}^{|D|}$.

Due to space limitations the formal proofs are deferred to the full version [21].

B. Concrete Bounds on Guessing Curves

We now derive concrete upper/lower bounds on the guessing curves ($\lambda_{M,B,D}$ and $\lambda_{M,B}$) by applying our general framework from Section V-A with the concrete upper/lower bounds on the guessing numbers $G^{\texttt{EX}}(q)$ and $G^{\texttt{IN}}(q)$ from Section IV. Due to space limitations, we present the concrete bounds on $\lambda_{M,B}$ here and defer the formal statements of concrete bounds on $\lambda_{M,B,D}$ to Appendix C.

First Concrete Upper/Lower Bound. We first apply Theorem 4 to the upper and lower bounds on $G^{\texttt{EX}}(q)$ and $G^{\texttt{IN}}(q)$ from

equations (1) and (2). In particular, we define $\hat{\lambda}^{ub1}_{M,B,D,S,\epsilon}$ and $\hat{\lambda}^{lb1}_{M,B,D,S,\epsilon}$ as the concrete upper/lower bounds on $\lambda_{M,B,D}$:

$$\hat{\lambda}^{ub1}_{M,B,D,S,\epsilon} := \min_{1\le i\le \ell, B\le \hat{G}^{\texttt{IN}}_S(q_i)-\epsilon/q_i} \left(\lambda_{M,G^{\texttt{IN}}(q_i),D}\right),$$

$$\hat{\lambda}^{lb1}_{M,B,D,S,\epsilon} := \max_{1\le i\le \ell, B\ge \hat{G}^{\texttt{EX}}_S(q_i)+\epsilon/q_i} \left(\lambda_{M,G^{\texttt{EX}}(q_i),D}\right).$$

For completeness, we set $\hat{\lambda}^{ub1}_{M,B,D,S,\epsilon} = 1$ (resp. $\hat{\lambda}^{lb1}_{M,B,D,S,\epsilon} = 0$) if $B > \max_{1\le i\le \ell}\{\hat{G}^{\texttt{IN}}_S(q_i) - \epsilon/q_i\}$ (resp. $B < \min_{1\le i\le \ell}\{\hat{G}^{\texttt{EX}}_S(q_i) + \epsilon/q_i\}$).

By Theorem 4 it follows that with probability at least $\ell \cdot \exp(-2k\epsilon^2)$ that $\hat{\lambda}^{ub1}_{M,B,D,S,\epsilon}$ (resp. $\hat{\lambda}^{lb1}_{M,B,D,S,\epsilon}$) is an upper (resp. lower) bound on $\lambda_{M,B,D}$ for every $B \ge 1$ where the randomness depends only on the selection the selection of k samples $|S| = k$ from our model M — see Theorem 12 in Appendix C for the formal statement. If we additionally assume that our dataset D was sampled iid from our unknown password distribution $\mathcal{P}$ then we can apply Theorem 3 (or Corollary 5) and use $\hat{\lambda}^{ub1}_{M,B,D,S,\epsilon}$ (resp. $\hat{\lambda}^{lb1}_{M,B,D,S,\epsilon}$ to upper (resp. lower) bound $\lambda_{M,B}$ for every $B \ge 1$. To ensure that we obtain high confidence bounds we include a small additional slack term (ϵ_2) to account for sampling error selecting our dataset D — see Theorem 6.

Theorem 6. *Given a password distribution $\mathcal{P}$ and a sequence of probability mesh points $Q = \{q_1, \cdots, q_\ell\}$, for any guessing number $B > 0$ and any parameters $0 \le \epsilon_1, \epsilon_2 \le 1$, we have:*

$$\Pr\left[\lambda_{M,B} \le \hat{\lambda}^{ub1}_{M,B,D,S,\epsilon_1} + \epsilon_2\right] \ge 1-\alpha$$

$$\Pr\left[\lambda_{M,B} \ge \hat{\lambda}^{lb1}_{M,B,D,S,\epsilon_1} - \epsilon_2\right] \ge 1-\alpha$$

where $\alpha = \ell\cdot\exp(-2k\epsilon_1^2) - \exp(-2|D|\epsilon_2^2)$ and the randomness is taken over the sample set $S \leftarrow \mathcal{M}^k$ with size k and the dataset $D \leftarrow \mathcal{P}^{|D|}$.

As long as the number of mesh points $\ell = |Q|$ is not too large and the sample size k (and $|D|$) is not too small, both the upper and lower bounds will hold with high probability.

Second Upper Bound. Recall that in equation (3) we derived a second guessing number lower bound using Markov's inequality and concentration bounds. This lower bound can be effective for rare passwords. We can use this lower bound on the guessing number to derive a second upper bound on the attacker's guessing curve. In particular, we define $\hat{\lambda}^{ub2}_{M,B,D,\mathbb{S},\delta}$ as another upper bound on $\lambda_{M,B,D}$:

$$\hat{\lambda}^{ub2}_{M,B,D,\mathbb{S},\delta} := \min_{1\le i\le \ell, B\le \delta\cdot\hat{G}^{\texttt{IN}}_{\mathbb{S},med}(q_i)} \left(\lambda_{M,G^{\texttt{IN}}(q_i),D}\right)$$

As before we set $\hat{\lambda}^{ub2}_{M,B,D,\mathbb{S},\delta} = 1$ whenever $B > \max_{1\le i\le \ell} \delta \cdot \hat{G}^{\texttt{IN}}_{\mathbb{S},med}(q_i)$. Applying Theorem 4 we can conclude that with high probability we have $\lambda_{M,B,D} \le \hat{\lambda}^{ub2}_{M,B,D,\mathbb{S},\delta}$ for all $B \ge 1$ — see Theorem 13 in Appendix C for the formal statement. If we additionally assume that the dataset D was sampled iid from our (unknown) password distribution $\mathcal{P}$ we can also upper bound $\lambda_{M,B}$ as in Theorem 7.

Theorem 7. *Given a password distribution $\mathcal{P}$ and a sequence of probability mesh points $Q = \{q_1, \cdots, q_\ell\}$, for any guessing number $B > 0$ and any parameters $0 < \delta \le \frac{1}{2}, 0 \le \epsilon_1 \le \frac{1}{2} - \delta, 0 \le \epsilon_2 \le 1$, we have:*

$$\Pr\left[\lambda_{M,B} \le \hat{\lambda}^{ub2}_{M,B,D,\mathbb{S},\delta} + \epsilon_2\right] \ge 1-\alpha$$

where $\alpha = \ell\cdot\exp(-2n\epsilon_1^2) - \exp(-2|D|\epsilon_2^2)$ and the randomness is taken over n Monte Carlo sample sets $\mathbb{S} = \{S_1, S_2, \ldots, S_n\}$ each of which contains k samples from model M and the dataset $D \leftarrow \mathcal{P}^{|D|}$.

C. A Trivial Upper Bound for Large Guessing Number

The previous section presents two upper bounds and one lower bound on $\lambda_{M,B,D}$ using a series of mesh points $q_1, \ldots, q_\ell$. However, when B gets large (i.e., $B > \max(\hat{G}_{med,S}(q_\ell) - \epsilon/q_\ell, \delta \cdot \hat{G}_{med,S}(q_\ell))$) we will run out of mesh points $q_1, \ldots, q_\ell$ and the two upper bounds will immediately jump to 1.

Note that if a password y is never outputted by a password cracking model M (i.e., $p^M_y = 0$), then the attacker using M will not be able to successfully guess this password in a dataset D. Denote $\hat{\lambda}^{ub3}_{M,D} := \frac{1}{|D|}|y \in D : p^M_y > 0|$ be the percentage of passwords that will be eventually guessed by model M. Trivially, we have $\lambda_{M,B,D} \le \hat{\lambda}^{ub3}_{M,D}$ for all finite guessing budgets $B \ge 0$ — see Theorem 14 in Appendix C for the formal theorem statement. We can also obtain the following upper bound on $\lambda_{M,B}$:

Theorem 8. *Given a password distribution $\mathcal{P}$ and a password cracking model M, for any parameters $0 \le \epsilon \le 1$, we have:*

$$\Pr\left[\forall B \ge 0, \lambda_{M,B} \le \hat{\lambda}^{ub3}_{M,D} + \epsilon\right] \ge 1 - \exp(-2|D|\epsilon^2)$$

where the randomness is taken over the dataset $D \leftarrow \mathcal{P}^{|D|}$.

Theorem 8 is derived by the observation that $\hat{\lambda}^{ub3}_{M,D}$ is concentrated on its expectation $\sum_{y\in\mathcal{P}} p^M_y$ which is the total probability mass of passwords in distribution $\mathcal{P}$ that will be guessed with non-zero probability in model M, and $\lambda_{M,B} \le \lambda_{M,\infty} = \sum_{y\in\mathcal{P}} p^M_y$. The formal proof is in Appendix E.

We remark that for the neural network models we consider we have $p^M_{pw} > 0$ for every password $pw \in D$ in our datasets. Thus, the upper bound $\hat{\lambda}^{ub3}_{M,D} = 1$ becomes trivial. However, as we will show in Section VI, for some other probabilistic models such as Markov Models and PCFG, over 20% and 40% of passwords in some of the datasets we consider had $p^M_{pwd} = 0$ indicating that these passwords will never be guessed by these particular cracking models. In these cases, our trivial upper bound $\hat{\lambda}^{ub3}_{M,D}$ can yield tighter bounds for large guessing number B.

D. Password Composition Policies

Some organizations impose restrictions (password composition policies) on the passwords that user's are allowed to select e.g., users may be required to include numbers, special symbols and/or capital letters. Even if our model M was trained entirely on passwords that are consistent with the policy $\mathbb{C}$ it is

still possible that some of the guesses generated by the model will be inconsistent with $\mathbb{C}$. In this case a trivial optimization for the attacker would be to simply filter out inconsistent guesses since they cannot appear in our password dataset D or in the support of our user password distribution $\mathcal{P}$. Intuitively, our bounds work by sampling $|S| = k$ passwords from our model M and then filtering to obtain $S' \subseteq S$ the subset of passwords which are consistent with our policy. We can then show that $\frac{1}{k}\sum_{z \in S', p_z^M > q} \frac{1}{p_z^M}$ is an unbiased estimate of the updated (exclusive) guessing number after filtering.

We show how our statistical techniques can be extended to provide confident upper/lower bounds on the guessing numbers *after* this filtering step. We can then apply our general framework from Section V-A to upper/lower bound the attacker's updated guessing curves $\lambda^{\mathbb{C}}_{M,B}$ and $\lambda^{\mathbb{C}}_{M,B,D}$ after filtering out password guesses that are inconsistent with $\mathbb{C}$— as before concentration bounds imply that $\lambda^{\mathbb{C}}_{M,B}$ and $\lambda^{\mathbb{C}}_{M,B,D}$ will be close (whp). See Appendix B for formal claims (Theorems 9, 10, 11) about our high confidence upper and lower bounds on $\lambda^{\mathbb{C}}_{M,B}$. Due to space limitations, we defer all the formal proofs and theorems of bounding guessing numbers and $\lambda^{\mathbb{C}}_{M,B,D}$ to the full version [21].

VI. Empirical Experiments

In this section we apply our statistical techniques to upper/lower bound the guessing numbers for user passwords and to upper/lower bound the attacker's guessing curve $\lambda_{M,B,D}$ and $\lambda_{M,B}$ as the guessing budget B varies from small to large.[6] To apply our statistical techniques we need to fix a password cracking model M and a password dataset D.

Password Cracking Models We consider 3 generative probabilistic models: Transformer neural network [22], 4-gram Markov model [10] and PCFG [7] (probabilistic context free grammar), each representing a different category of password cracking models. For Markov Models and PCFGs we use the same implementations as [23]. We chose Transformer as the representative of neural network because Transformer is the state-of-the-art machine learning model in learning sequential data. It is faster in training and sampling than RNN [2]; also, it was more efficient in guessing strong passwords in our local tests. The structure and hyperparameters in training Transformer model can be found in the Appendix A-A. We did not expend significant effort optimizing our password cracking models as our primary focus is demonstrating how our statistical techniques can be applied to obtain tight upper/lower bounds on guessing numbers and the attacker's guessing curve $\lambda_{M,B}$.

Password Datasets We consider six breached password datasets in our experiments: Bfield (0.54m user accounts), Brazzers (0.93m), Clixsense (2.2m), CSDN (6.4m), Neopets (68.3m), 000webhost (15.3m). When we analyze the guessing curve $\lambda_{M,D}$ we will assume that each dataset D represents $|D|$ independent samples from an (unknown) probability distribution over user passwords — this unknown password distribution may be different at different sites. We remark that when analyzing the guessing curve $\lambda_{M,B,D}$ we do not need to make any assumptions about how the dataset D was sampled. However, we argue that the independent samples assumption is reasonable for the datasets we consider. Blocki and Liu [19] developed a linear programming technique which can detect when a dataset is blatantly inconsistent with our assumption about iid samples, e.g., if many user registered for two accounts with the same password or if a large fraction of the dataset was duplicated. Their technique rejects the LinkedIn frequency corpus, which contained far more passwords than unique e-mail addresses. By contrast, all of the datasets that we consider passed the consistency checks from [19].

Ethical Considerations The usage of password datasets which contain stolen passwords that were subsequently leaked on the internet raises important ethical considerations. Our usage of the datasets does not pose any additional risk to users as the datasets are already publicly available. We do not crack any new passwords as part of our analysis nor do we attempt to deanonymize the datasets by linking passwords to particular user accounts. Furthermore, we believe that the statistical techniques developed in this paper may benefit users by helping defenders to pick informed password policies.

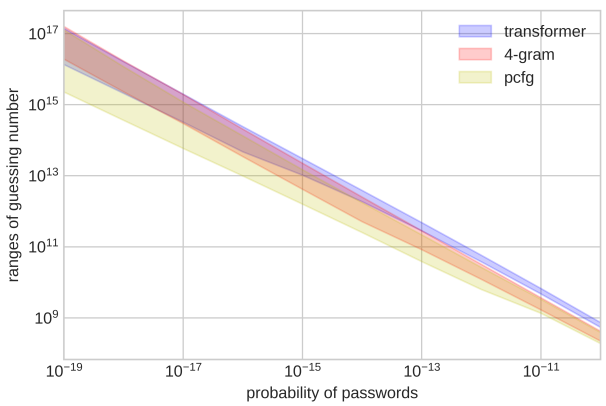
A. Bounding Guessing Number with Confidence

We begin by using Theorems 1 and 2 to upper and lower bound the guessing numbers for specific passwords. When applying Theorem 1 we set the number of samples $k = 206848$ [7] and $\epsilon = 0.005$ to obtain confidence $> 99\%$ that each upper/lower bound is correct. Similarly, when applying Theorem 2 we set $n = 186$, $k = 5120$, $\delta = 0.333$ and $\epsilon = 0.167$ to obtain 99% confidence that each upper bound is correct. Since we have two separate lower bounds on the guessing number we will take the maximum of the lower bounds obtained from Theorems 1 and 2 — union bounds imply that the maximum lower bound will be correct with probability at least 98%. Figure 4a plots our upper/lower bounds for the guessing number as the probability q ranges from 10^{-19} to 10^{-12}. We consider three models: PCFG, Markov and Transformer each trained on the Bfield dataset (due to space limitations we omit similar plots for models trained on other datasets). As we can see, the distance between the upper/lower bounds increases as the password probability probability decreases. For example, consider the PCFG model, for passwords with probability $q = 10^{-19}$ our upper/lower bound on the guessing number range from 2.3×10^{15} to 1.3×10^{17}. By contrast, when $q = 10^{-11}$ the range is $[1.9 \times 10^8, 3.9 \times 10^8]$. Estimating the guessing numbers for strong passwords is particularly error prone. Thus, it is important to consider the confidence range of guessing numbers when measuring password strength/resistance to brute-force guessing attacks.

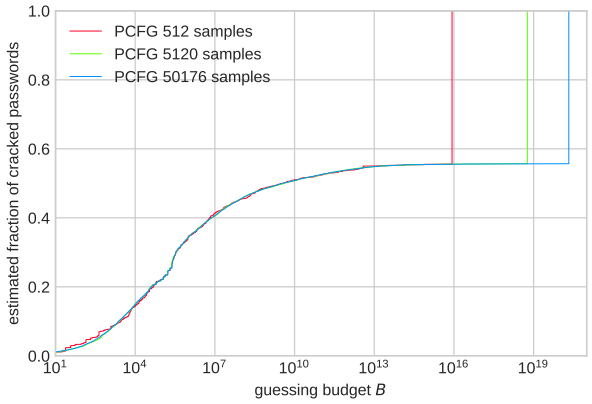
Limitation of Regular Monte Carlo Strength Estimation Figure 4b shows what happens if we apply regular Monte

[6] Our source code is publicly available at the Github repository https://github.com/ConfidentMonteCarlo/ConfidentMonteCarlo.

[7] Throughout the experiments we set number of samples a multiple of 512 since sample generation using transformer is computed by GPU in parallel with batch size 512.



(a) Ranges of Guessing Number vs Password Probability



(b) Regular Monte Carlo Estimation with Varying Sample sizes

Fig. 4: Limitation of Regular Monte Carlo Estimation

Carlo estimation as a heuristic to estimate the attacker's guessing curve. We train our PCFG model using the Bfield dataset withholding $25,000$ passwords D_{test} for testing. For the purpose of comparison we run regular Monte Carlo strength estimation with three different sampling parameters $k \in \{512, 5120, 50176\}$. The figure plots the guessing budget B vs. the estimated fraction of cracked passwords i.e., the fraction of passwords in D_{test} whose estimated guessing number is less than B. As we noted previously our guessing number estimates become less and less certain as q decreases so intuitively we might expect the estimated guessing curves to be less accurate when the guessing budget B is large. Indeed this is what we observe. In reality the PCFG model does not perform well i.e., for 43% of the passwords x in D_{test} we have $p_x^M = 0$ meaning that this particular password will *never* be guessed and the curve $\lambda_{M,B,D_{test}}$ will plateau before 57%.[8] However, we observe that each of the estimated guessing curves suddenly spikes to 100%. In particular, given any set S of $|S| = k$ passwords sampled from the our model M we can define $B_S \doteq \frac{1}{k}\sum_x \frac{1}{p_x^M}$. Observe that the estimated guessing number for *any* password y is $\frac{1}{k}\sum_{x:p_x^M \geq p_y^M} \frac{1}{p_x^M} \leq B_S$ even if $p_y^M = 0$. Thus, for sample S any $B \geq B_S$ regular Monte Carlo will incorrectly estimate that $\lambda_{M,B,D} = 1$. The spikes appear at different points for $k \in \{512, 5120, 50176\}$ since the upper bound B_S increases with $k = |S|$ as there are more opportunities to sample rare passwords in M. This example demonstrates that regular Monte Carlo can produce inaccurate results on real datasets and further motivates the need to derive upper/lower bounds on the attacker's guessing curve which hold with high probability.

B. Confident Guessing Curves

We now turn our attention to the problem of upper/lower bounding the attacker's guessing curve using Theorem 6, 7, 8, 12, 13 and 14.

[8]The problem of PCFG plateauing has been observed in prior work e.g., see [13]. Our focus is on how the plateau impacts Monte Carlo guessing curve estimates.

Experimental Setup: For each password dataset $D_{original}$ we first perform train-test split to obtain D_{train} and D_{test} with $|D_{test}| = 25,000$. All 3 probabilistic models are trained with D_{train}, then we use D_{test} to upper/lower bound the attacker's guessing curve.

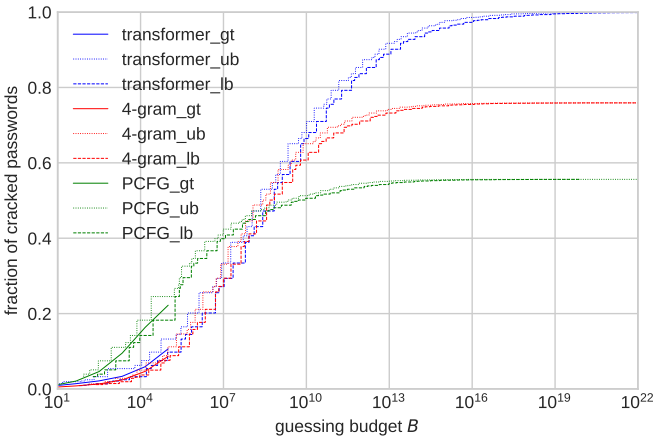
To apply Theorem 6 and 7 we need to define a set of probability mesh points $Q = \{q_1, \cdots, q_\ell\}$. In particular, we fix probability mesh points to be $j \times 10^{-4-i}$ for $i \in [1, 25]$ and $j \in \{0.25, 0.5, 0.75, 1\}$ for a total of $\ell = 25 \cdot 4 = 100$ mesh points. In Theorem 6 and 7 there are two sources of confidence loss. The term $\ell \cdot \exp(2k\epsilon_1^2)$ (resp. $\ell \cdot \exp(-2n\epsilon_1^2)$) in Theorem 6 (resp. 7) upper bounds the guessing number error associated with *any* point in our probability mesh. The term $\exp(-2|D_{test}|\epsilon_2^2)$ accounts for confidence loss due to sampling error from our unknown password distribution.

The parameter setting of Theorem 12, 13 and 14 is identical with that of Theorem 6, 7 and 8, respectively.

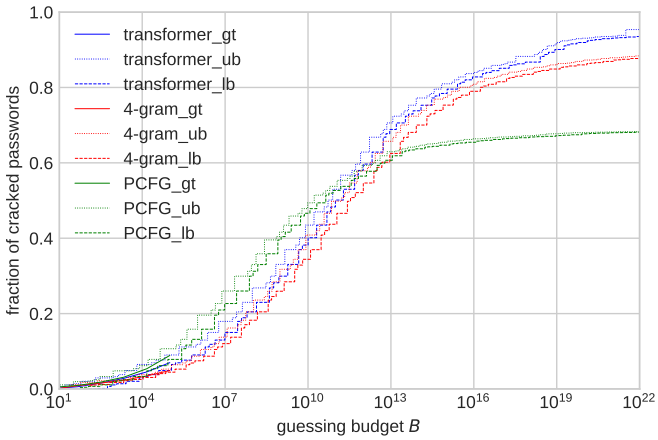
In our experiments we instantiate the parameters k, ϵ_1, ϵ_2, n and $|D_{test}|$ to ensure that the total probability of failure for each bound is at most 0.01. More specifically, we set $|D_{test}| = 25,000$ and $\epsilon_2 = 0.01$ in both Theorem 6 and 7. We set $\epsilon_1 = 0.005$ and $k = 206848$ in Theorem 6; we set $\epsilon_1 = 0.167$, $k = 5120$, $n = 186$ and $\delta = 0.333$ in Theorem 7. Similarly, in Theorem 8 we set $\epsilon = 0.0096$ and $|D_{test}| = 25,000$ to ensure that, except with probability 0.01, $\hat{\lambda}^{ub3}_{M,D} + \epsilon$ is an upper bound on $\lambda_{M,B}$ *for all* guessing budgets $B \geq 0$.

Because we have multiple techniques to generate upper bounds it will often make sense to consider the best upper/lower bound. Thus, we define $\hat{\lambda}^{ub}_{M,B} = \min\left\{\hat{\lambda}^{ub1}_{M,B,D,S,\epsilon_1} + \epsilon_2, \hat{\lambda}^{ub2}_{M,B,D,\mathbb{S},\delta} + \epsilon_2, \hat{\lambda}^{ub3}_{M,D} + \epsilon, 1\right\}$ and $\hat{\lambda}^{ub}_{M,B,D} = \min\left\{\hat{\lambda}^{ub1}_{M,B,D,S,\epsilon_1}, \hat{\lambda}^{ub2}_{M,B,D,\mathbb{S},\delta}, \hat{\lambda}^{ub3}_{M,D}, 1\right\}$ which are best upper bounds for $\lambda_{M,B}$ and $\lambda_{M,B,D}$, respectively. Applying union bounds the probability that the curve $\hat{\lambda}^{ub}_{M,B}$ (resp. $\hat{\lambda}^{ub}_{M,B,D}$) is not a valid upper bound for $\lambda_{M,B}$ (resp. $\lambda_{M,B,D}$) is at most 0.03 (resp. 0.02[9]). For notional

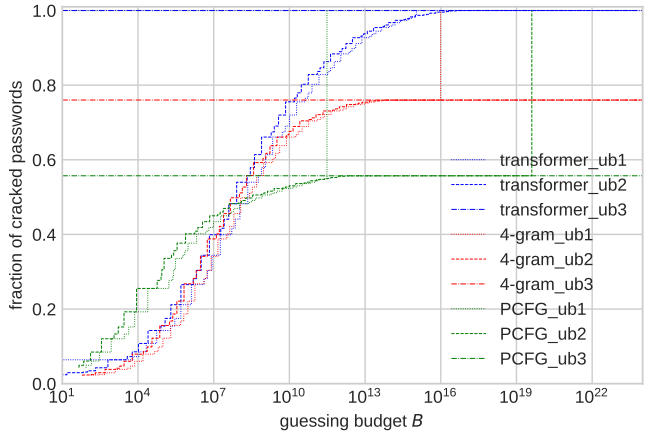
[9]For any $B > 0$ we have $\hat{\lambda}^{ub3}_{M,D} \geq \lambda_{M,B,D}$ with probability 1.



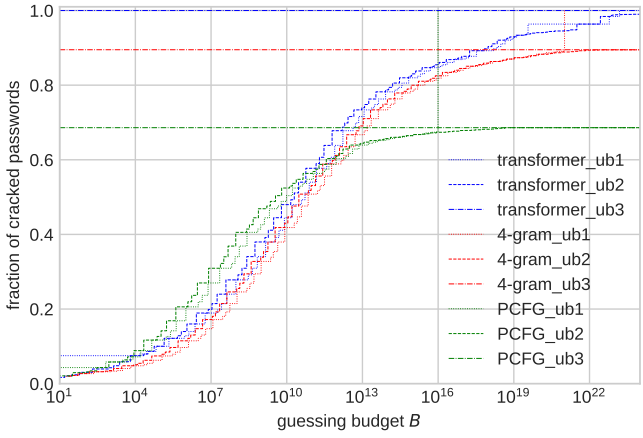
(a) Upper/Lower Bounds on $\lambda_{M,B,D}$ (Bfield)



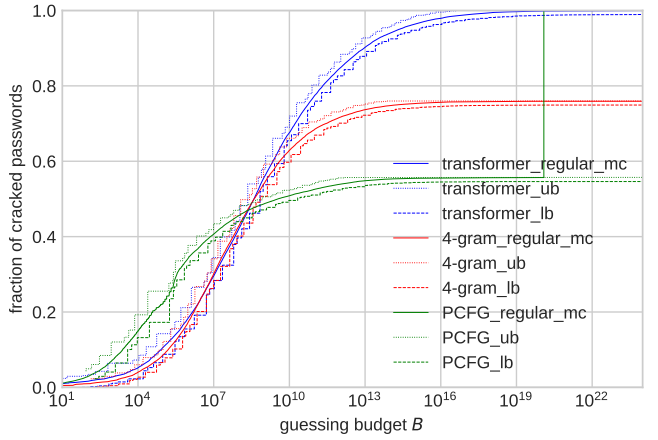
(b) Upper/Lower Bounds on $\lambda_{M,B,D}$ (000Webhost)



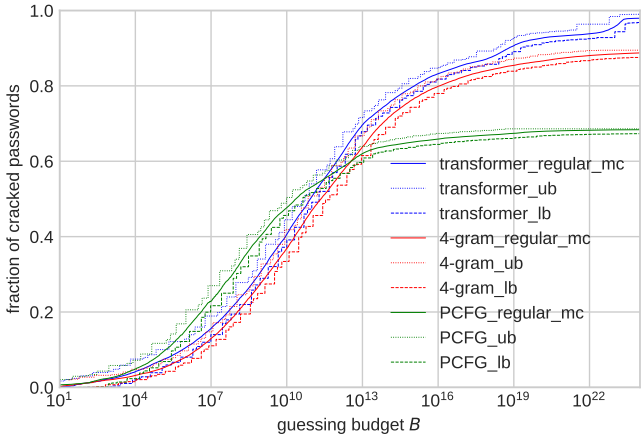
(c) Three Upper Bounds on $\lambda_{M,B}$ (Bfield)



(d) Three Upper Bounds on $\lambda_{M,B}$ (000Webhost)



(e) Upper/Lower Bounds on $\lambda_{M,B}$ (Bfield)



(f) Upper/Lower Bounds on $\lambda_{M,B}$ (000Webhost)

Fig. 5: Upper/Lower Bounds on Guessing Curves

convenience we also define $\hat{\lambda}^{lb}_{M,B} = \hat{\lambda}^{lb1}_{M,B,D,S,\epsilon_1} - \epsilon_2$ and $\hat{\lambda}^{lb}_{M,B,D} = \hat{\lambda}^{lb1}_{M,B,D,S,\epsilon_1}$. In figure legends we use `modelname_ub` to denote $\hat{\lambda}^{ub}_{M,B}$ or $\hat{\lambda}^{ub}_{M,B,D}$ contingent on the figure caption. The same case applies to lower bound legend `modelname_lb`. Figure 5 plots our upper/lower bounds, the regular Monte Carlo estimation, and the ground truth of password guessing curves for the Bfield and 000webhost datasets. Due to space limitations the remaining plots for the Brazzers, Clixsense, CSDN, and Neopets datasets are deferred to Appendix D — see Figure 7 and the full version [21].

Upper/Lower Bounds on $\lambda_{M,B,D}$. Figure 5a and 5b plot our best upper bound $\hat{\lambda}^{ub}_{M,B,D}$ (resp. best lower bound $\hat{\lambda}^{lb}_{M,B,D}$) as the guessing budget B varies for each password model M. We additionally plot the ground truth[10] $\lambda_{M,B,D}$ for $B \leq 10^6$ — denoted by `modelname_gt` in figure legends.

Discussion. Consider Figure 5a (Bfield) as an example. We find the the upper and lower bounds are reasonably close to each other. Furthermore, when $B \leq 10^6$ we note that $\lambda_{M,B,D}$ (the ground truth of fraction of cracked passwords against dataset D) is sandwiched between our upper/lower bounds for all three models M. We can also use our confident guessing curves to draw rigorous statistical comparisons between the three cracking models. In general, if we find that $\hat{\lambda}^{lb}_{M_1,B,D} > \hat{\lambda}^{ub}_{M_2,B,D}$ then this supports the hypothesis that model M_1 outperforms model M_2 with guessing budget B against dataset D. Notice that whenever $B \geq 10^{11}$ the lower bound for Transformer is strictly higher than the upper bound of 4-gram Markov model. This supports the hypothesis that Transformers outperforms 4-gram for larger guessing budgets. This observation is consistent with prior work (e.g., [2]), but by using Confident Monte Carlo we can increase our confidence that this observation is correct and not the result of inaccurate estimates for the guessing curve.

Comparing Our Upper Bounds on $\lambda_{M,B}$. For the sake of comparison Figure 5c and 5d plot our three upper bounds on $\lambda_{M,B}$ separately. In the legend `modelname_ub1` denotes the upper bound $\min\{1, \hat{\lambda}^{ub1}_{M,B,D,S,\epsilon_1} + \epsilon_2\}$, `modelname_ub2` denotes the upper bound $\min\{1, \hat{\lambda}^{ub2}_{M,B,D,\mathbb{S},\delta} + \epsilon_2\}$ and `modelname_ub3` denotes the upper bound $\min\{1, \hat{\lambda}^{ub3}_{M,D} + \epsilon\}$. In Figure 5c and 5d we occasionally observe spiking behavior for the first two upperbounds (ub1 and ub2). We note that for an upper bound this behavior is not problematic for two reasons. First, even if an upper bound spikes to 1.0 the upper bound continues to be accurate i.e., the interpretation is simply that the current statistical approach does not rule out the *possibility* that the attackers cracks 100% of passwords. By contrast, for regular Monte Carlo strength estimation if the guessing curve spikes to 1 this represents a (likely incorrect) *prediction* that the attacker will crack 100% of passwords. Second, in all of the plots from Figure 5c and 5d we find that the first two upper bounds ub1 and ub2 approach the third upper bound ub3 (a straight line) *before* we observe the spiking behavior. Thus, we expect to obtain reasonably tight upper bounds by considering best of the three bounds i.e., $\hat{\lambda}^{ub}_{M,B}$.

Confident Bounds on $\lambda_{M,B}$. Figure 5e and 5f compare our best upper/lower bounds $\hat{\lambda}^{ub}_{M,B}$ and $\hat{\lambda}^{lb}_{M,B}$ with the guessing curve obtained from regular Monte Carlo Strength estimation — denoted by $\lambda^{MC}_{M,B}$.[11] The number of samples used to compute $\lambda^{MC}_{M,B}$ is 10240, a multiple of 512 that is closest to sample size 10000 which is recommended in [1].

Discussion. Given that Monte Carlo Strength estimation has been widely used in password research (e.g., [2], [4], [13], [16]–[18]) it is natural to ask when regular Monte Carlo estimates are (in)accurate. If $\lambda^{MC}_{M,B} < \hat{\lambda}^{lb}_{M,B}$ (resp. $\lambda^{MC}_{M,B} > \hat{\lambda}^{ub}_{M,B}$) then we can confidently conclude that regular Monte Carlo Strength Estimation is underestimating (resp. overestimating) the true guessing curve. We observed that regular Monte Carlo tends to overestimate the true guessing curve when the guessing budget B is sufficiently large. As an example, consider Figure 5e (Bfield) using PCFG as a our probabilistic password model. When $B \approx 1.2 \times 10^{20}$, $\lambda^{MC}_{M,B}$ suddenly jumps to 1 whereas $\hat{\lambda}^{ub}_{M,B} = 0.56$. Thus, we can confidently conclude that regular Monte Carlo Strength estimation significantly overestimates the fraction of passwords cracked by PCFG when $B \geq 1.2 \times 10^{20}$. On the positive side we consistently found that $\hat{\lambda}^{lb}_{M,B} \leq \lambda^{MC}_{M,B} \leq \hat{\lambda}^{ub}_{M,B}$ as long as the guessing budget B is not too large indicating that the Monte Carlo estimate $\lambda^{MC}_{M,B}$ is *plausibly accurate*. Furthermore, if we have $\hat{\lambda}^{lb}_{M,B} \leq \lambda^{MC}_{M,B} \leq \hat{\lambda}^{ub}_{M,B}$ and the difference $\hat{\lambda}^{ub}_{M,B} - \hat{\lambda}^{lb}_{M,B}$ is sufficiently small (e.g., < 0.05) then we can confidently conclude that $\lambda^{MC}_{M,B}$ is an accurate estimation. For most guessing budgets $1.7 \times 10^5 \leq B \leq 1.2 \times 10^{20}$ we have $\hat{\lambda}^{ub}_{M,B} - \hat{\lambda}^{lb}_{M,B} < 0.05$ allowing us to confidently conclude that $\lambda^{MC}_{M,B}$ is an accurate estimate for $\lambda_{M,B}$. Thus, our findings indicate that regular Monte Carlo may be a reasonable heuristic whenever the guessing budget B is not too large — with the caveat that one would still need to use Confident Monte Carlo to be fully confident that the heuristic Monte Carlo guessing curve is still accurate for each new password model/distribution or dataset.

We can also apply our results to compare the distributions from different datasets. For example, we compare Bfield and 000webhost by fixing the password probabilistic model M to be transformer and guessing budget to be $B = 10^{12}$, then we consider bounds for fraction of cracked passwords. We have $\hat{\lambda}^{lb}_{M,B} = 0.84$ for the Bfield distribution and $\hat{\lambda}^{ub}_{M,B} = 0.63$ for the 000webhost distribution. Thus, we can confidently conclude that the 000webhost distribution is more resistant to attacks by an attacker using the Transformer Cracking Model with guessing budget $B = 10^{12}$.

[10] We generated a dictionary of the top 1 million popular passwords in each model-defined distribution $\mathcal{M}$ by brute-force and use the dictionary to crack passwords in D_{test}.

[11] Note that the regular Monte Carlo estimate $\lambda^{MC}_{M,B}$ will also depend on the test dataset D (sampled from the unknown password distribution) and samples S (sampled from the model M) in addition to the model M and guessing budget B. We omit S and D from the subscript to simplify notation.

C. Small Password Datasets

In this subsection we apply Confident Monte Carlo to analyze small password datasets D motivated by applications to password user studies where the number of users $|D|$ may be constrained by research budgets e.g., 20000 participants in [2], 4509 participants in [18], 5000 participants in [24]. Typically, the dataset D is further partitioned into disjoint sets D_0 (control group) and $D_1, D_2, \ldots$ (treatment groups) and we would like to apply hypothesis testing to determine whether or not a particular intervention (e.g., requiring user's to include numbers, capital letters and/or special symbols) results in passwords that are harder for an attacker to crack. More specifically, fixing a model M and a guessing budget B we would like to test the hypothesis that $\mathbb{E}[\lambda_{M,B,D_0}] < \mathbb{E}[\lambda_{M,B,D_1}]$ when D_0 and D_1 are sampled from different distributions $\mathcal{P}_0$ (control) and $\mathcal{P}_1$ (treatment).

If we knew λ_{M,B,D_0} and λ_{M,B,D_1} then we could apply standard hypothesis tests, but unfortunately when B is very large we cannot compute λ_{M,B,D_0} or λ_{M,B,D_1} exactly. However, we can obtain tight upper/lower bounds $\hat{\lambda}^{ub}_{M,B,D_i}$ and $\hat{\lambda}^{lb}_{M,B,D_i}$ since the number of samples from our model M is not constrained by user study size. Now to test the hypothesis $\mathbb{E}[\lambda_{M,B,D_0}] < \mathbb{E}[\lambda_{M,B,D_1}]$ we can apply standard hypothesis tests under the (pessimistic) assumption that $\lambda_{M,B,D_0} = \hat{\lambda}^{ub}_{M,B,D_0}$ and $\lambda_{M,B,D_1} = \hat{\lambda}^{lb}_{M,B,D_1}$. As long as the upper/lower bounds are close $\hat{\lambda}^{ub}_{M,B,D_i} \approx \hat{\lambda}^{lb}_{M,B,D_i}$ the impact on statistical power will be minimal.

When reporting confidence levels we need to account for errors which might occur if our upper/lower bounds are inaccurate (i.e., $\lambda_{M,B,D_i} > \hat{\lambda}^{ub}_{M,B,D_i}$ or $\lambda_{M,B,D_i} < \hat{\lambda}^{lb}_{M,B,D_i}$) in addition to standard sampling error over the selection of D_i from $\mathcal{P}_i$. Thus, when we apply hypothesis testing and obtain a "p"-value p' our final "p"-value would be $p = p' + \delta_1$ where δ_1 denotes the probability of the event that either one of our upper/lower bounds for λ_{M,B,D_i} was incorrect. In our analysis below we ensure that $\Pr[\lambda_{M,B,D_i} > \hat{\lambda}^{ub}_{M,B,D_i} \vee \lambda_{M,B,D_i} < \hat{\lambda}^{lb}_{M,B,D_i}] \leq \delta_1 = 0.01$.

We provide a concrete example in Figure 6. To simulate a small user study we subsampled datasets D_0 (000webhost) and D_1 (Bfield) of size $|D_0| = |D_1| = 500$. Figure 6 plots the upper bound $\hat{\lambda}^{ub}_{M,B,D_i}$ and lower bound $\hat{\lambda}^{lb}_{M,B,D_i}$ for both datasets where the model M is a Transformer model trained on the Neopets dataset using the same parameters in section VI-B. In Table I we also provide 96% binomial confidence intervals for $B = 10^{13}$ under the assumption that (1) $\lambda_{M,B,D_i} = \hat{\lambda}^{ub}_{M,B,D_i}$ and (2) under the assumption that $\lambda_{M,B,D_i} = \hat{\lambda}^{lb}_{M,B,D_i}$. We can combine both confidence intervals to obtain a new 95% confidence interval. In this case we observe that the binomial confidence intervals are disjoint and that $\hat{\lambda}^{ub}_{M,B,D_0} < \hat{\lambda}^{lb}_{M,B,D_1}$ where D_0 denotes 000webhost and D_1 denotes Bfield. Thus, we accept the hypothesis that the 000webhost password distribution provide stronger resistance against an password attacker with a guessing budget $B = 10^{13}$ using model M.

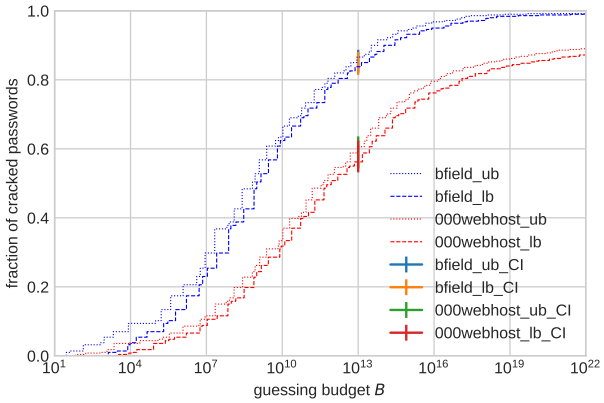


Fig. 6: Upper/Lower Bound on $\lambda_{M,B,D}$ and the Confidence Interval at $B = 10^{13}$

TABLE I: Overall 95% Binomial Confidence Interval When Guessing Budget $B = 10^{13}$

	Bfield	000webhost
ub	(0.821, 0.887)	(0.544, 0.636)
lb	(0.814, 0.882)	(0.532, 0.624)
overall	(0.814, 0.887)	(0.532, 0.636)

VII. Conclusion

In this paper, we provided theoretical and empirical evidence that regular Monte Carlo will sometimes yield inaccurate estimations of the guessing number of a password and of the attacker's guessing curve. We extend the regular Monte Carlo method by developing rigorous statistical techniques to confidently upper/lower bound the guessing number of a password. We also showed how to use our Confident Monte Carlo framework to provide high confidence upper/lower bounds on the attacker's guessing curve. Our rigorous statistical framework allows us to evaluate the impact of a password policy interventions (e.g., password composition policies) on password strength, rigorously compare the performance of different cracking models and characterize the resistance of a password dataset/distribution to an offline password attacker.

Acknowledgment

We would like to thank anonymous reviewers for constructive feedback which helped us to improve the paper.

References

[1] M. Dell'Amico and M. Filippone, "Monte Carlo strength evaluation: Fast and reliable password checking," in *ACM CCS 2015*, I. Ray, N. Li, and C. Kruegel, Eds. ACM Press, Oct. 2015, pp. 158–169.

[2] 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. USENIX Association, Aug. 2016, pp. 175–191.

[3] 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*. IEEE Computer Society Press, May 2019, pp. 380–397.

[4] W. Bai and J. Blocki, "Dahash: distribution aware tuning of password hashing costs," in *International Conference on Financial Cryptography and Data Security*. Springer, 2021, pp. 382–405.

[5] D. Pasquini, M. Cianfriglia, G. Ateniese, and M. Bernaschi, "Reducing bias in modeling real-world password strength via deep learning and dynamic dictionaries," in *USENIX Security 2021*, M. Bailey and R. Greenstadt, Eds. USENIX Association, Aug. 2021, pp. 821–838.

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

[7] 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*. IEEE Computer Society Press, May 2009, pp. 391–405.

[8] 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*. IEEE Computer Society Press, May 2012, pp. 523–537.

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

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

[11] 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.

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

[13] 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. USENIX Association, Aug. 2015, pp. 463–481.

[14] "Hashcat: advanced password recovery," https://hashcat.net/hashcat/.

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

[16] D. Wang, P. Wang, D. He, and Y. Tian, "Birthday, name and bifacial-security: Understanding passwords of chinese web users," in *USENIX Security 2019*, N. Heninger and P. Traynor, Eds. USENIX Association, Aug. 2019, pp. 1537–1555.

[17] 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, "Designing password policies for strength and usability," *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 4, may 2016. [Online]. Available: https://doi.org/10.1145/2891411

[18] B. Ur, F. Alfieri, M. Aung, L. Bauer, N. Christin, J. Colnago, L. F. Cranor, H. Dixon, P. Emami Naeini, H. Habib *et al.*, "Design and evaluation of a data-driven password meter," in *Proceedings of the 2017 chi conference on human factors in computing systems*, 2017, pp. 3775–3786.

[19] J. Blocki and P. Liu, "Towards a rigorous statistical analysis of empirical password datasets," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023.

[20] C. McDiarmid *et al.*, "On the method of bounded differences," *Surveys in combinatorics*, vol. 141, no. 1, pp. 148–188, 1989.

[21] "Confident monte carlo: Rigorous analysis of guessing curves for probabilistic password models (full version)," https://github.com/ConfidentMonteCarlo/ConfidentMonteCarlo/blob/main/Rigorous_Analysis_of_Guessing_Curves_for_Password_Cracking_Models.pdf, accessed: 2023-04-07.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[23] M. Dell'Amico. (2017) Implementation of monte carlo estimation. https://github.com/matteodellamico/montecarlopwd.

[24] 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 *Proceedings of the sigchi conference on human factors in computing systems*, 2011, pp. 2595–2604.

TABLE II: The percentage of filtered passwords

Bfield	Brazzers	Clixsense	CSDN	Neopets	000Webhost
0.155%	0.009%	1.317%	0.445%	2.128%	2.410%

APPENDIX A
EXPERIMENT DETAILS

A. Transformer neural network

Our Transformer neural network is composed of a classic encoder-decoder structure. The encoder contains 16 identical layers stacked upon each other. Each layer has 2 sub-layers. The first is a multi-head self-attention mechanism with 16 heads, and the second is a simple, position-wise fully connected feed-forward network. The decoder is a basic linear layer. In addition, we set the embedding size to be 128 and the size of hidden layers to be 1024.

B. Data Preprocessing

For Transformer neural network, we restrict the alphabet set to be the union of 95 printable ASCII characters and 2 special characters — $\mathsf{start}/\perp$ denoting the start/end of a password text string, respectively. Also, we fix the maximum length of passwords to be 16. Thus, passwords in D_{train} containing non-ASCII characters or having length larger than 16 are filtered out, and the remaining passwords are prepend/append with $\mathsf{start}/\perp$. The percentage of filtered passwords are shown in Table II. Passwords in D_{test} are untouched in evaluation, passwords containing characters not in the alphabet set are assigned probability 0, which implies not being cracked under any circumstances. For 4-gram Markov model, we adopted implementation from [23] which only performs prepending/appending without any smoothing techniques. We did not apply any smoothing techniques to make sure the probabilities of all allowable strings add up to 1, i.e., the model strictly defines a probability distribution. How heuristics like smoothing techniques affect rigorous analysis of password probabilistic models remains a open question.

APPENDIX B
BOUNDING GUESSING CURVE UNDER PASSWORD COMPOSITION POLICIES

For any password pwd we use $\mathtt{Allowed}^{\mathbb{C}}(pwd)$ to describe the password policy $\mathbb{C}$ an attacker would follow when making guesses. We set $\mathtt{Allowed}^{\mathbb{C}}(pwd) = 1$ if and only if password pwd satisfies the policy $\mathbb{C}$ (e.g., passwords must be at least 8 characters long). Similar to $G^{\mathtt{EX}}(\cdot)$ and $G^{\mathtt{IN}}(\cdot)$, we define $G^{\mathbb{C},\mathtt{EX}}(q) = |\{z \in \mathcal{M} : p_z^M > q \wedge \mathtt{Allowed}^{\mathbb{C}}(z) = 1\}|$ and $G^{\mathbb{C},\mathtt{IN}}(q) = |\{z \in \mathcal{M} : p_z^M \geq q \wedge \mathtt{Allowed}^{\mathbb{C}}(z) = 1\}|$ such that $G^{\mathbb{C},\mathtt{EX}}(q)+1$ and $G^{\mathbb{C},\mathtt{IN}}(q)$ are the smallest and largest possible guessing number of a password with probability q in M under policy $\mathbb{C}$, i.e., $G^{\mathbb{C},\mathtt{EX}}(q)+1 \leq G^{\mathbb{C}}(q) \leq G^{\mathbb{C},\mathtt{IN}}(q)$. Given a set S of k iid samples randomly selected from model distribution $\mathcal{M}$, we let $\hat{G}_S^{\mathbb{C},\mathtt{EX}}(q) = \frac{1}{k}\sum_{z\in S, p_z^M > q, \mathtt{Allowed}^{\mathbb{C}}(z)=1} \frac{1}{p_z^M}$ be the regular Monte Carlo estimate of $G^{\mathbb{C},\mathtt{EX}}(q)$, and $\hat{G}_S^{\mathbb{C},\mathtt{IN}}(q) = \frac{1}{k}\sum_{z\in S, p_z^M \geq q, \mathtt{Allowed}^{\mathbb{C}}(z)=1} \frac{1}{p_z^M}$ be the regular Monte Carlo estimate of $G^{\mathbb{C},\mathtt{IN}}(q)$. For $\phi = \{\mathtt{EX}, \mathtt{IN}\}$, we

denote $\hat{G}^{\mathbb{C},\phi}_{\mathbb{S},med}(q) = \text{median}\left(\{\hat{G}^{\mathbb{C},\phi}_{S_i}(q)\}_{1\le i\le n}\right)$ where $\mathbb{S} = \{S_1, S_2, \ldots, S_n\}$ and each S_i contains k independent samples from model M.

In this section, we state the theorems that upper and lower bound the guessing curve of an attacker who follows given password policies $\mathbb{C}$. Due to space limitations we defer the formal proof to the full version [21]. We define $\lambda^{\mathbb{C}}_{M,B,D} = \frac{1}{|D|}|y \in D : G^{\mathbb{C}}(y) \ge B|$ to be the guessing curve of set D against an attacker with model M under password policy $\mathbb{C}$, i.e., the percentage of passwords in D cracked by making the top B most probable guesses outputted by model M satisfying password policy $\mathbb{C}$. Define $\lambda^{\mathbb{C}}_{M,B} = \Pr_{y\leftarrow\mathcal{P}}[G^{\mathbb{C}}(y) \le B]$ to be the probability of a password from some unknown distribution $\mathcal{P}$ cracked by an attacker making the top B guesses of model M satisfying password policy $\mathbb{C}$.

First Upper/Lower Bound. We define $\hat{\lambda}^{\mathbb{C},ub1}_{M,B,D,S,\epsilon}$ and $\hat{\lambda}^{\mathbb{C},lb1}_{M,B,D,S,\epsilon}$ below to be the first upper and lower bounds on $\lambda^{\mathbb{C}}_{M,B,D}$:

$$\hat{\lambda}^{\mathbb{C},ub1}_{M,B,D,S,\epsilon} := \min_{1\le i\le \ell, B\le \hat{G}^{\mathbb{C},\mathtt{IN}}_S(q_i)-\epsilon/q_i} \left(\lambda^{\mathbb{C}}_{M,G^{\mathbb{C}}(q_i),D}\right),$$
$$\hat{\lambda}^{\mathbb{C},lb1}_{M,B,D,S,\epsilon} := \max_{1\le i\le \ell, B\ge \hat{G}^{\mathbb{C},\mathtt{EX}}_S(q_i)+\epsilon/q_i} \left(\lambda^{\mathbb{C}}_{M,G^{\mathbb{C}}(q_i),D}\right).$$

For completeness, we set $\hat{\lambda}^{\mathbb{C},ub1}_{M,B,D,S,\epsilon} = 1$ (resp. $\hat{\lambda}^{\mathbb{C},lb1}_{M,B,D,S,\epsilon} = 0$) if $B > \max_{1\le i\le \ell}\{\hat{G}^{\mathbb{C},\mathtt{IN}}_S(q_i) - \epsilon/q_i\}$ (resp. $B < \min_{1\le i\le \ell}\{\hat{G}^{\mathbb{C},\mathtt{EX}}_S(q_i)+\epsilon/q_i\}$). Then our first upper and lower bounds on $\lambda^{\mathbb{C}}_{M,B}$ is shown below:

Theorem 9. *Given a password distribution $\mathcal{P}$, a sequence of probability mesh points $Q = \{q_1, \cdots, q_\ell\}$ and a password policy $\mathbb{C}$, for any guessing number $B \ge 0$ and any parameters $0 \le \epsilon_1, \epsilon_2 \le 1$, we have:*

$$\Pr\left[\lambda^{\mathbb{C}}_{M,B} \le \hat{\lambda}^{\mathbb{C},ub1}_{M,B,D,S,\epsilon_1} + \epsilon_2\right] \ge 1-\alpha$$
$$\Pr\left[\lambda^{\mathbb{C}}_{M,B} \ge \hat{\lambda}^{\mathbb{C},lb1}_{M,B,D,S,\epsilon_1} - \epsilon_2\right] \ge 1-\alpha$$

where $\alpha = \ell\cdot\exp(-2k\epsilon_1^2) - \exp(-2|D|\epsilon_2^2)$ and the randomness is taken over the sample set $S \leftarrow \mathcal{M}^k$ with size k and the dataset $D \leftarrow \mathcal{P}^{|D|}$.

Second Upper Bound. We define $\hat{\lambda}^{\mathbb{C},ub2}_{M,B,D,S,\delta}$ below to be our second bound on $\lambda^{\mathbb{C}}_{M,B,D}$:

$$\hat{\lambda}^{\mathbb{C},ub2}_{M,B,D,S,\delta} := \min_{1\le i\le \ell, B\le \delta\cdot\hat{G}^{\mathbb{C},\mathtt{IN}}_{\mathbb{S},med}(q)} \left(\lambda^{\mathbb{C}}_{M,G^{\mathbb{C}}(q_i),D}\right)$$

For completeness, we set $\hat{\lambda}^{\mathbb{C},ub2}_{M,B,D,S,\epsilon} = 1$ if $B > \max_{1\le i\le \ell}\{\delta\cdot \hat{G}^{\mathbb{C},\mathtt{IN}}_{\mathbb{S},med}(q)\}$. Then our second upper bound on $\lambda^{\mathbb{C}}_{M,B}$ is shown below:

Theorem 10. *Given a password distribution $\mathcal{P}$, a sequence of probability mesh points $Q = \{q_1, \cdots, q_\ell\}$ and a password policy $\mathbb{C}$, for any guessing number $B \ge 0$ and any parameters $0 < \delta \le \frac{1}{2}, 0 \le \epsilon_1 \le \frac{1}{2} - \delta, 0 \le \epsilon_2 \le 1$, we have:*

$$\Pr\left[\lambda^{\mathbb{C}}_{M,B} \le \hat{\lambda}^{\mathbb{C},ub2}_{M,B,D,\mathbb{S},\delta} + \epsilon_2\right] \ge 1-\alpha$$

where $\alpha = \ell\cdot\exp(-2n\epsilon_1^2) - \exp(-2|D|\epsilon_2^2)$ the randomness is taken over n Monte Carlo sample sets $\mathbb{S} = \{S_1, S_2, \ldots, S_n\}$ each of which contains k samples from model M and the dataset $D \leftarrow \mathcal{P}^{|D|}$.

Third Upper Bound. We denote $\hat{\lambda}^{\mathbb{C},ub3}_{M,D} := \frac{1}{|D|}|y \in D : p^M_y > 0 \wedge \mathtt{Allowed}^{\mathbb{C}}(y) = 1|$ to be the percentage of passwords that will be eventually guessed by model M restricted by policy $\mathbb{C}$. We have $\lambda^{\mathbb{C}}_{M,B,D} \le \hat{\lambda}^{\mathbb{C},ub3}_{M,D}$ for any guessing number $B > 0$ due to the fact that passwords with zero probability in M will never be guessed by M. Then we have the following trivial upper bound on $\lambda^{\mathbb{C}}_{M,B}$:

Theorem 11. *Given a password distribution $\mathcal{P}$, a password cracking model M and a password policy $\mathbb{C}$, for any parameters $0 \le \epsilon \le 1$, we have:*

$$\Pr\left[\forall B \ge 0, \lambda^{\mathbb{C}}_{M,B} \le \hat{\lambda}^{\mathbb{C},ub3}_{M,D} + \epsilon\right] \ge 1 - \exp(-2|D|\epsilon^2)$$

where the randomness is taken over the dataset $D \leftarrow \mathcal{P}^{|D|}$.

Appendix C
Bounding Guessing Curve $\lambda_{M,B,D}$

In this section we present the theorems of bounding $\lambda_{M,B,D}$.

First, by applying Theorem 4 to the upper and lower bounds on $G^{\mathtt{EX}}(q)$ and $G^{\mathtt{IN}}(q)$ from equations (1) and (2) we have our first upper and lower bounds on $\lambda_{M,B,D}$:

Theorem 12. *Given a password dataset D containing iid samples from distribution $\mathcal{P}$ and a sequence of probability mesh points $Q = \{q_1, \cdots, q_\ell\}$, for any parameter $0 \le \epsilon \le 1$, we have:*

$$\Pr\left[\forall B \ge 0, \lambda_{M,B,D} \le \hat{\lambda}^{ub1}_{M,B,D,S,\epsilon}\right] \ge 1 - \ell\cdot\exp(-2k\epsilon^2)$$
$$\Pr\left[\forall B \ge 0, \lambda_{M,B,D} \ge \hat{\lambda}^{lb1}_{M,B,D,S,\epsilon}\right] \ge 1 - \ell\cdot\exp(-2k\epsilon^2)$$

where the randomness is taken over the sample set $S \leftarrow \mathcal{M}^k$ with size k.

Second, by applying Theorem 4 to the lower bound on $G^{\mathtt{IN}}(q)$ from equation (3) we have our second upper bound on $\lambda_{M,B,D}$:

Theorem 13. *Given a password dataset D containing iid samples from distribution $\mathcal{P}$ and a sequence of probability mesh points $Q = \{q_1, \cdots, q_\ell\}$, for any parameters $0 < \delta \le \frac{1}{2}, 0 \le \epsilon \le \frac{1}{2} - \delta$, we have:*

$$\Pr\left[\forall B \ge 0, \lambda_{M,B,D} \le \hat{\lambda}^{ub2}_{M,B,D,\mathcal{S},\delta}\right] \ge 1 - \ell\cdot\exp(-2n\epsilon^2)$$

where the randomness is taken over n Monte Carlo sample sets $\mathbb{S} = \{S_1, S_2, \ldots, S_n\}$ each of which contains k samples from model M.

Third, we show the formal statement of the trivial upper bound on $\lambda_{M,B,D}$ proposed in Section V-C as below:

Theorem 14. *Given a password cracking model M, for any guessing number $B \geq 0$, $\lambda_{M,B,D} \leq \hat{\lambda}^{ub3}_{M,D}$.*

Appendix D
Additional Plots

Figure 7 shows the additional plots (Brazzers and CSDN) for experiments described in Section VI. Due to space limitations the plots for Clixsense and Neopets are deferred to the full version [21].

Appendix E
Missing Proofs

Reminder of Theorem 3. *For any guessing number $B \geq 0$ and any $0 \leq \epsilon \leq 1$, we have:*

$$\Pr[\lambda_{M,B} \geq \lambda_{M,B,D} - \epsilon] \geq 1 - \exp(-2|D|\epsilon^2), \quad \textit{and}$$
$$\Pr[\lambda_{M,B} \leq \lambda_{M,B,D} + \epsilon] \geq 1 - \exp(-2|D|\epsilon^2)$$

where the randomness is taken over the sample set $D \leftarrow \mathcal{P}^{|D|}$.

Proof of Theorem 3. Consider $\lambda_{M,B,D}$ to be a function of the $|D|$ samples in D. For any two sets with the same number of iid samples from $\mathcal{P}$ $D = \{d_1, \cdots, d_i, \cdots, d_{|D|}\}$ and $D' = \{d_1, \cdots, d'_i, \cdots, d_N\}$ that only differs on the one sample d_i and d'_i, the difference of $\lambda_{M,B,D}$ and $\lambda_{M,B,D'}$ is at most $1/|D|$, i.e., $|\lambda_{M,B,D} - \lambda_{M,B,D'}| \leq 1/|D|$. Therefore, using McDiarmid's inequality [20] we have:

$$\Pr[\lambda_{M,B} \geq \lambda_{M,B,D} - \epsilon] \geq 1 - \exp(-2|D|\epsilon)$$
$$\Pr[\lambda_{M,B} \leq \lambda_{M,B,D} + \epsilon] \geq 1 - \exp(-2|D|\epsilon)$$

□

Reminder of Theorem 8. *Given a password distribution $\mathcal{P}$ and a password cracking model M, for any parameters $0 \leq \epsilon \leq 1$, we have:*

$$\Pr\left[\forall B \geq 0, \lambda_{M,B} \leq \hat{\lambda}^{ub3}_{M,D} + \epsilon\right] \geq 1 - \exp(-2|D|\epsilon^2)$$

where the randomness is taken over the dataset $D \leftarrow \mathcal{P}^{|D|}$.

Proof of Theorem 8. Let $\sum_{y\in\mathcal{P}} p_y^M$ be the total probability mass of passwords in distribution $\mathcal{P}$ that will be guessed with non-zero probability in model M. Then we have $\lambda_{M,B} \leq \sum_{y\in\mathcal{P}} p_y^M$ for any $B \geq 0$, since passwords with zero probability in M will never be guessed. Let $X_1, \ldots, X_{|D|}$ be $|D|$ random variables where $X_i = 1$ if the ith sample in D has non-zero probability in M, and $X_i = 0$ otherwise. Then $\sum_{i=1}^{|D|} X_i = |D|\hat{\lambda}^{ub3}_{M,D}$ is the number of passwords in D that will eventually be guessed by model M. Note that $\hat{\lambda}^{ub3}_{M,D} = \frac{1}{|D|}\mathrm{E}(\sum_{i=1}^{|D|} X_i) = \sum_{y\in\mathcal{P}} p_y^M$. Using Chernoff bound, for any $0 \leq \epsilon \leq 1$ we have:

$$\Pr[\sum_{y\in\mathcal{P}} p_y^M \leq \hat{\lambda}^{ub3}_{M,D} + \epsilon] \geq 1 - \exp(-2|D|\epsilon^2).$$

Since $\forall B \geq 0, \lambda_{M,B} \leq \sum_{y\in\mathcal{P}} p_y^M$, we have:

$$\Pr\left[\forall B \geq 0, \lambda_{M,B} \leq \hat{\lambda}^{ub3}_{M,D} + \epsilon\right] \geq 1 - \exp(-2|D|\epsilon^2).$$

□

Reminder of Theorem 1. *Given a set S with k iid password samples sampled from distribution $\mathcal{M}$, for any probability $q \in [0,1]$ and any parameter $\epsilon \geq 0$, we have:*

$$\Pr[G(q) \leq \hat{G}^{\texttt{IN}}_S(q) + \epsilon/q] \geq 1 - \exp\left(-2k\epsilon^2\right)$$
$$\Pr[G(q) \geq \hat{G}^{\texttt{EX}}_S(q) + 1 - \epsilon/q] \geq 1 - \exp\left(-2k\epsilon^2\right)$$

where the randomness is taken over $S \leftarrow \mathcal{M}^k$.

Proof of Theorem 1. Given a password probability q and a set S with k samples randomly sampled from distribution $\mathcal{M}$, consider k independent random variables $X_1^\phi, \ldots, X_k^\phi$ where for any $1 \leq i \leq k$ and $\phi \in \{\texttt{EX}, \texttt{IN}\}$ we define:

$$X_i^{\texttt{EX}} := \begin{cases} \frac{1}{p_z^M} & \text{if the } i\text{th sampled password is } z \text{ and } p_z^M > q; \\ 0 & \text{otherwise.} \end{cases}$$

$$X_i^{\texttt{IN}} := \begin{cases} \frac{1}{p_z^M} & \text{if the } i\text{th sampled password is } z \text{ and } p_z^M \geq q; \\ 0 & \text{otherwise.} \end{cases}$$

Then we have $0 \leq X_i^\phi \leq \frac{1}{q}$ and the expectation of X_i^ϕ is $G^\phi(q)$ as shown below:

$$\mathbb{E}(X_i^{\texttt{EX}}) = \sum_{z\in\mathcal{M}, p_z^M > q} p_z^M \cdot \frac{1}{p_z^M} = |\{z \in \mathcal{M} : p_z^M > q\}| = G^{\texttt{EX}}(q)$$

$$\mathbb{E}(X_i^{\texttt{IN}}) = \sum_{z\in\mathcal{M}, p_z^M \geq q} p_z^M \cdot \frac{1}{p_z^M} = |\{z \in \mathcal{M} : p_z^M \geq q\}| = G^{\texttt{IN}}(q)$$

Observe that $\hat{G}^\phi_S(q) = \frac{1}{k}\sum_{i=1}^k X_i^\phi$ and $\mathbb{E}(\frac{1}{k}\sum_{i=1}^k X_i^\phi) = \frac{1}{k}\sum_{i=1}^k \mathbb{E}(X_i^\phi) = G^\phi(q)$. Using Hoeffding's inequality we can bound $|\hat{G}^\phi_S(q) - G^\phi(q)|$ with any $t \geq 0$ as below:

$$\Pr[G^\phi(q) - \hat{G}^\phi_S(q) \leq t] = \Pr[\mathbb{E}(\frac{1}{k}\sum_{i=1}^k X_i^\phi) - \frac{1}{k}\sum_{i=1}^k X_i^\phi \leq t]$$
$$\geq 1 - \exp\left(\frac{-2t^2k^2}{\sum_{i=1}^k (\frac{1}{q} - 0)^2}\right) = 1 - \exp\left(-2kt^2q^2\right);$$
$$\Pr[G^\phi(q) - \hat{G}^\phi_S(q) \geq -t] = \Pr[\mathbb{E}(\frac{1}{k}\sum_{i=1}^k X_i^\phi) - \frac{1}{k}\sum_{i=1}^k X_i^\phi \geq -t]$$
$$\geq 1 - \exp\left(\frac{-2t^2k^2}{\sum_{i=1}^k (\frac{1}{q} - 0)^2}\right) = 1 - \exp\left(-2kt^2q^2\right).$$

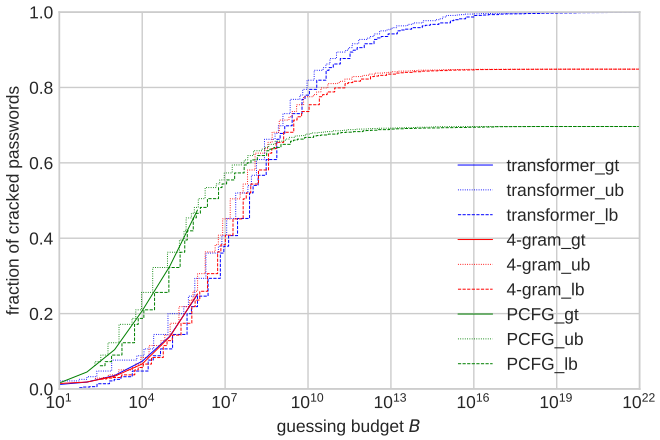
Setting $t = \epsilon/q$, then for any $\epsilon \geq 0$ we have:

$$\Pr[G^\phi(q) \leq \hat{G}^\phi_S(q) + \epsilon/q] \geq 1 - \exp\left(-2k\epsilon^2\right)$$
$$\Pr[G^\phi(q) \geq \hat{G}^\phi_S(q) - \epsilon/q] \geq 1 - \exp\left(-2k\epsilon^2\right)$$

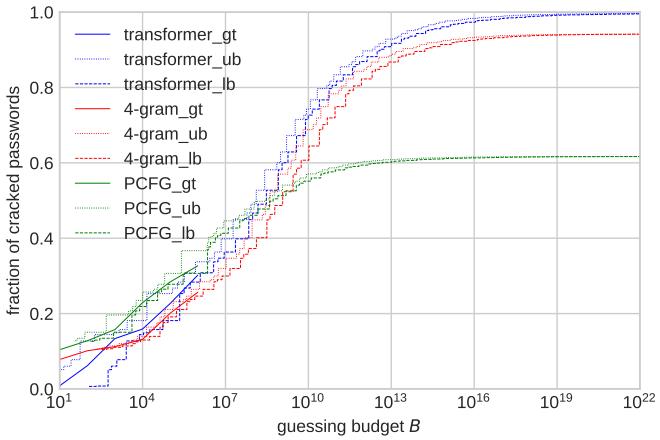
Note that the actual guessing number $G(q)$ in between $G^{\texttt{EX}}(q) + 1$ and $G^{\texttt{IN}}(q)$, i.e., $G^{\texttt{EX}}(q) + 1 \leq G(q) \leq G^{\texttt{IN}}(q)$. Therefore, by directly applying the upper bound of $G^{\texttt{IN}}(q)$ and the lower bound of $G^{\texttt{EX}}(q)$ above, we can rigorously bound $G(q)$ with high confidence as stated in this theorem. □

Reminder of Theorem 2. *For any password probability $q \in [0,1]$, and any parameters $0 \leq \delta \leq \frac{1}{2}$ and $0 \leq \epsilon \leq \frac{1}{2} - \delta$,*

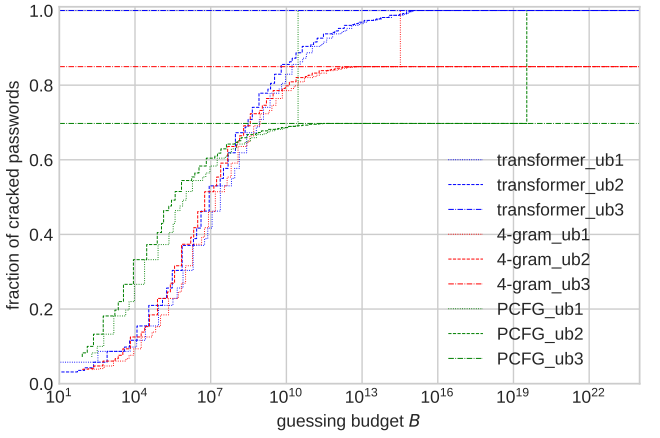
$$\Pr[G(q) \geq \delta \cdot \hat{G}^{\texttt{EX}}_{\mathbb{S},med}(q) + 1] \geq 1 - \exp(-2n\epsilon^2)$$



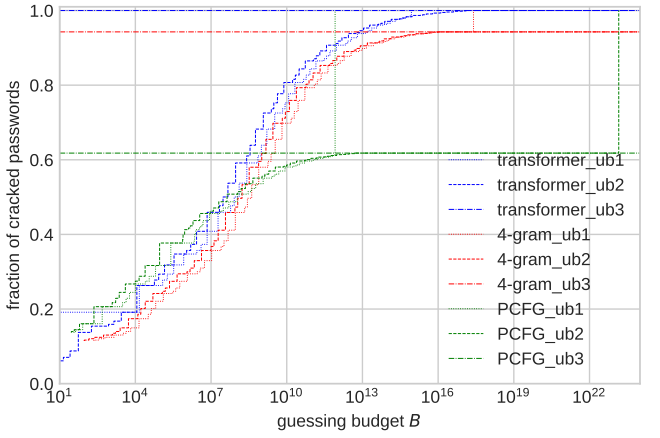
(a) Upper/Lower Bounds on $\lambda_{M,B,D}$ (Brazzers)



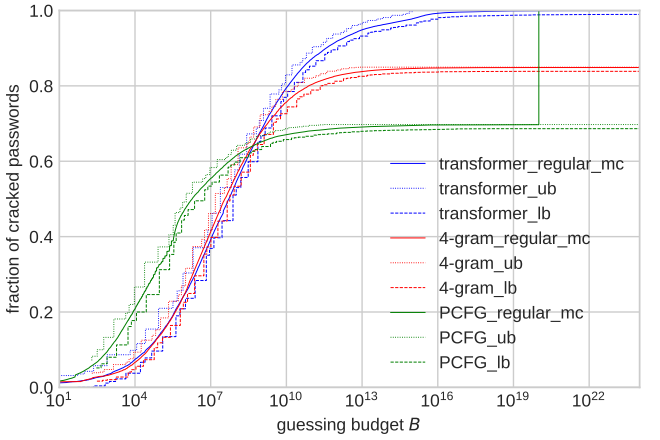
(b) Upper/Lower Bounds on $\lambda_{M,B,D}$ (CSDN)



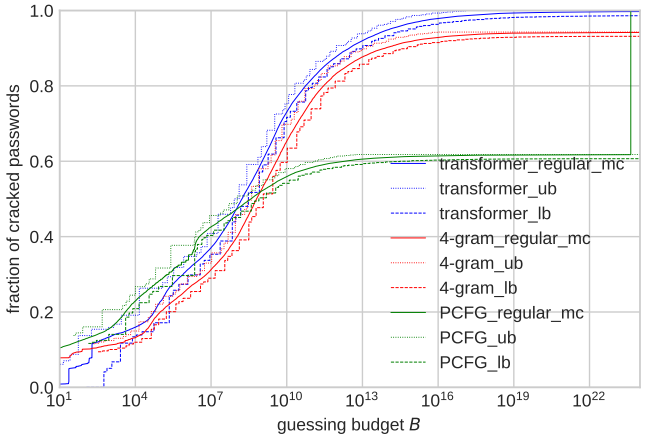
(c) Three Upper Bounds on $\lambda_{M,B}$ (Brazzers)



(d) Three Upper Bounds on $\lambda_{M,B}$ (CSDN)



(e) Upper/Lower Bounds on $\lambda_{M,B}$ (Brazzers)



(f) Upper/Lower Bounds on $\lambda_{M,B}$ (CSDN)

Fig. 7: Confident Bounds for Additional Datasets

where the randomness is taken over n sets of k Monte Carlo samples $\mathbb{S} = \{S_1, \cdots, S_n\}$ from model M.

Proof of Theorem 2. We start with lower bounding $G^{\phi}(q)$ for $\phi \in \{\texttt{EX}, \texttt{IN}\}$ by directly applying Markov's inequality. Recall that $\mathbb{E}(\hat{G}^{\phi}_{S}(q))_{S \leftarrow \mathcal{M}^{|S|}} = G^{\phi}(q)$. Using Markov's inequality, for any $0 < \delta \leq 1$ we have:

$$\Pr[\hat{G}^{\phi}_{S}(q) \leq G^{\phi}(q)/\delta] \geq 1 - \delta. \tag{8}$$

Let X^{ϕ}_{i} be the indicator variable corresponding to the ith execution of the regular Monte Carlo estimation with randomly selected sample set S_i. $X^{\phi}_{i} = 1$ if and only if the ith estimation $\hat{G}^{\phi}_{S_i}(q) \leq G^{\phi}(q)/\delta$; otherwise, $X^{\phi}_{i} = 0$. Note that $\mathbb{E}(X^{\phi}_{i}) \geq 1 - \delta$ according to equation (8). Using Chernoff bound we have:

$$\Pr[\sum_{i=1}^{n} X^{\phi}_{i} \leq \frac{n}{2}] \leq \Pr[\sum_{i=1}^{n} X^{\phi}_{i} \leq n(1 - \delta - \epsilon)] \leq \exp(-2n\epsilon^2)$$

where we set $\delta \leq \frac{1}{2}$ and $\epsilon \leq \frac{1}{2} - \delta$. Then we have:

$$\Pr[\hat{G}^{\phi}_{\mathbb{S},med}(q) \leq \frac{G^{\phi}(q)}{\delta}] \geq \Pr[\sum_{i=1}^{n} I_i \geq \frac{n}{2}] \geq 1 - \exp(-2n\epsilon^2)$$

Recall $G(q) \geq G^{\texttt{EX}}(q) + 1$. By directly applying the lower bound of $G^{\texttt{EX}}(q)$ above on $G(q)$ this theorem is proved.

□

APPENDIX F
NOTATION TABLE

Table III lists the common notations that are used in this paper.

TABLE III: Notation Table

M	A probabilistic password guessing model
$\mathcal{M}$	Password distribution generated by model M
p^{M}_{pwd}	Probability of password pwd outputted by model M
$G(pwd)$	Number of guesses that an attacker using model M needs to check in order to crack the password pwd
$G(q)$	Guessing number of a password pwd with probability $p^{M}_{pwd} = q$
$G^{\texttt{EX}}(q)$	Number of passwords with probability *strictly greater than* q
$G^{\texttt{IN}}(q)$	Number of passwords with probability *greater than or equal to* q
ϕ	A variable representing `EX` or `IN`
k	Number of Monte Carlo iid samples in S from $\mathcal{M}$
S	Sample set that has k iid samples from $\mathcal{M}$
$\hat{G}^{\phi}_{S}(q)$	Monte Carlo estimation of $G^{\phi}(q)$ for $\phi \in \{\texttt{EX}, \texttt{IN}\}$
N	Number of repetitions of Monte Carlo estimation
$\mathbb{S}$	Set of N Monte Carlo sample sets $S_1, \ldots, S_N$
$\hat{G}^{\phi}_{\mathbb{S},med}(q)$	Median estimated guessing number among N estimated guessing numbers $\hat{G}^{\phi}_{S_1}(q), \ldots, \hat{G}^{\phi}_{S_N}(q)$
B	Number of guesses checked by an attacker
D	Password dataset an attacker wants to crack
$\lambda_{M,B,D}$	Fraction of passwords in D that can be cracked within B guesses by an attacker using model M
$\mathcal{P}$	Real password distribution that D is sampled from.
$\lambda_{M,B}$	The probability that a random password pwd sampled from $\mathcal{P}$ would be cracked within B guesses
$\mathbb{C}$	Password composition policy