Statistical metrics for individual password strength

Joseph Bonneau

jcb82@cl.cam.ac.uk
University of Cambridge

Abstract. We propose several possible metrics for measuring the strength of an individual password or any other secret drawn from a known, skewed distribution. In contrast to previous ad hoc approaches which rely on textual properties of passwords, we consider the problem without any knowledge of password structure. This enables rating the strength of a password given a large sample distribution without assuming anything about password semantics. We compare the results of our generic metrics against those of the NIST metrics and other previous "entropy-based" metrics for a large password dataset, which suggest over-fitting in previous metrics.

1 Introduction

It is often desirable to estimate the resistance to guessing provided by a specific password. This can be useful both for research purposes to compare password choices between two groups with insufficient data to construct full distributions [9,11,14]. It can also be used to make a proactive password checker which indicates to a user the strength of a particular password during enrolment. [1]. Current practice usually relies on rules-of-thumb based on entropy estimates of English such as those standardised by NIST [3]. These have been argued to be inaccurate estimators of a password's vulnerability to cracking attacks [15].

Instead, we advocate using probability estimates based on an approximation of the distribution $\mathcal{X}$ which a passord x was drawn from. We'll call such a measure the *estimated strength* and denote it as $S_{\mathcal{X}}(x)$. It is important to recognise that any estimated strength is only as accurate as our approximation for $\mathcal{X}$ is accurate for a target population. For example, the password tequiero (which roughly translates to iloveyou) is much weaker within a distribution of passwords chosen by Spanish speakers than by English speakers.[1] We'll assume initially that the purposes of this paper that the population-wide distribution of passwords $\mathcal{X}$ is completely known before exploring the implications of relaxing this assumption.

[1] This problem was illustrated by the research of Dell'Amico et al. [7], who found vastly different guessing efficiency based on the language of a password dictionary and the users who chose passwords in a dataset.

2 Desired properties

We can describe two important properties that we would like for a strength metric:

1. **Consistency for uniform distributions**: For a discrete uniform distribution $\mathcal{U}_N$ in which each of N events is equally likely, such as randomly-chosen 64-bit keys, we want any strength metric to indicate that each value has 2^N bits of strength. Specifically:

$$\forall_{x \in \mathcal{U}_N} \quad S_{\mathcal{U}_N}(x) = \lg N \tag{1}$$

2. **Monotonicity**: A strength metric should rate any event more weakly than events which are less common in the underlying distribution $\mathcal{X}$:

$$\forall_{x,x' \in \mathcal{X}} \quad p_x \geq p_{x'} \iff S_{\mathcal{X}}(x) \leq S_{\mathcal{X}}(x') \tag{2}$$

3 Strength metrics

We formalise several purely statistical techniques for estimate a password's strength in this way, which require no general assumptions about human tendencies in password selection. In all cases, we convert our strength estimates into units of bits to satisfy our desired consistency property.

3.1 Probability metric

The simplest approach is to take the estimated probability p_x from $\mathcal{X}$ and estimate the size of a uniform distribution in which all elements have probability p_x:

$$S^{\mathrm{P}}_{\mathcal{X}}(x) = -\lg p_x \tag{3}$$

This is, in fact, the classic definition of the *self-information* of x (also called the *surprisal*), which is the underlying basis for Shannon entropy H_1 [13]. It is easy to see that for a randomly drawn value $x \xleftarrow{\mathrm{R}} \mathcal{X}$, the expected value of this metric is:

$$E\left[S^{\mathrm{P}}_{\mathcal{X}}(x) \middle| x \xleftarrow{\mathrm{R}} \mathcal{X}\right] = \sum_{i=1}^{|\mathcal{X}|} p_x \cdot -\lg p_x = H_1(\mathcal{X}) \tag{4}$$

Previous metrics which attempt to measure the probability of a password, for example using Markov models [6,12,4] or probabilistic context-free grammars [16], can be seen as attempts to approximate $S^{\mathrm{P}}_{\mathcal{X}}(x)$.

Applied to guessing, this model measures the (logarithmic) expected number of guesses before x is tried by an attacker who is guessing random values $x \xleftarrow{\mathrm{R}} \mathcal{X}$ with no memory. This attack model might apply if an attacker is guessing randomly according to the distribution to evade an intrusion-detection system. For optimal sequential guessing attacks however, this metric has no direct relation.

3.2 Index metric

To estimate the strength of an individual event against an optimal guessing attack, the only relevant fact is the index i_x of x, that is, the number of items in $\mathcal{X}$ of greater probability than p_x. We can convert this to an effective key-length by considering the size of a uniform distribution which would, on average, require i_x guesses. This gives us a strength metric of:

$$S^{\mathrm{I}}_{\mathcal{X}}(x) = \lg\left(2 \cdot i_x - 1\right) \tag{5}$$

Intuitively, this metric is related to real-world attempts to measure the strength of an individual password by estimating when it would be guessed by password-cracking software [10].

A practical problem with this metric is that many events may have the same estimated probability. If we break ties arbitrarily then in the case of the uniform distribution $\mathcal{U}_N$ the formula won't give $\lg N$ for all events. In fact, it won't even give $\lg N$ on average for all events, but instead $\approx \lg N - (\lg e - 1)$ (proved in Appendix B).

A better approach for a sequence of j equiprobable events $x_i \cdots x_{i+j}$ where $p_i = \ldots = p_{i+j}$ is to assign index $\frac{i+j}{2}$ to all events when computing the index strength metric. This is equivalent to assuming an attacker will choose randomly given a set of remaining candidates with similar probabilities and it does give a value of $\lg N$ to all events in the uniform distribution.

This is slightly unsatisfactory however, as it means for a distribution $\mathcal{X} \approx \mathcal{U}_N$ which is "close" to uniform but with a definable ordering of the events, the average strength will appear to jump down by $(\lg e - 1) \approx 0.44$ bits.

A second problem is that the most probable event in $\mathcal{X}$ is assigned a strength of $\lg 1 = 0$. To satisfy our consistency requirement is a necessary limitation, since for $\mathcal{U}_1$ we must assign the single possible event a strength of $\lg 1 = 0$.

3.3 Partial guessing metric

The probability metric doesn't model a sequential guessing attack, while the index approach has a number of peculiarities. A better approach is to consider the minimum amount of work done per account by an optimal partial guessing attack which will compromise accounts using x. The α-guesswork $\tilde{G}_\alpha$, defined in [2], reflects exactly the (logarithmic) expected amount of work per account required of an attacker desiring to break a proportion α of accounts. We provide the full definition for computing $\tilde{G}_\alpha$ in Appendix A.

For example, if a user chooses the password encryption, an optimal attacker performing a sequential attack against the RockYou distribution will have broken 51.8% of accounts before guessing encryption. Thus, a user choosing the password encryption can expect to be safe from attackers who aren't aiming to compromise at least this many accounts, which takes $\tilde{G}_{0.518}$ work on average per account. We can turn this into a strength metric as follows:

$$S^{\mathrm{G}}_{\mathcal{X}}(x') = \tilde{G}_{\alpha_x}(\mathcal{X}) : \alpha_x = \sum_{i=1}^{i_x} p_i \tag{6}$$

Because $\tilde{G}_\alpha(\mathcal{U}_N) = \lg N$ for all α, the consistency property is trivially satisfied. Moreover, for "close" distributions $\mathcal{X} \approx \mathcal{U}_N$ where $|p_i - \frac{1}{N}| < \varepsilon$ for all i, we'll have $S^{\mathrm{G}}_{\mathcal{X}}(x_i) \to \lg N$ for all i as $\varepsilon \to 0$, unlike for S^{I} where strength will vary as long as there is any defined ordering.

As with $S^{\mathrm{I}}_{\mathcal{X}}$ though, we encounter the problem of of ordering for events with statistically indistinguishable probabilities. We'll apply the same tweak and give each event in a sequence of equiprobable events $x_i \cdots x_{i+j}$ the index $\frac{i+j}{2}$.

4 Estimation from a sample

Just like for distribution-wide metrics [2], there are several issues when estimating strength metrics using an approximation for $\mathcal{X}$ obtained from a sample.

4.1 Estimation for unseen events

All of the above metrics are undefined for a previously unobserved event $x' \notin \mathcal{X}$. This is a result of our assumption that we have perfect information about $\mathcal{X}$. If not, we inherently need to rely on some approximation. As a consequence of the monotonicity requirement, $S(x' \notin \mathcal{X})$ must be $\geq \max(S(x \in \mathcal{X}))$. The naive formula for $S^{\mathrm{P}}_{\mathcal{X}}(x')$ assigns a strength estimate of ∞ though, which is not desirable.

We should therefore smooth our calculation of strength metrics by using some estimate $p(x') > 0$ even when $x' \notin \mathcal{X}$. Good-Turing techniques [8] do not address this problem as they provide an estimate for the total probability of seeing a new event, but not the probability of a specific new event. A conservative approach is to add x' to $\mathcal{X}$ with a probability $\frac{1}{N+1}$, on the basis that if it is seen in practice in a distribution we're assuming is close to our reference $\mathcal{X}$, then we can treat x' as one additional observation about $\mathcal{X}$. This is analogous to the common heuristic of "add-one smoothing" in word frequency estimation [8]. This correction gives an estimate of $S^{\mathrm{P}}_{\mathcal{X}}(x) = \lg(N+1)$ for unobserved events.

For the index metric, this correction produces the smoothed estimate $S^{\mathrm{I}}_{\mathcal{X}}(x') = \lg 2N + 1$, an increase of roughly 1 bit, due to the aforementioned instability for a distribution $\mathcal{X} \approx \mathcal{U}_N$. For the partial guessing strength metric we have $S^{\mathrm{G}}_{\mathcal{X}}(x') \approx \tilde{G}_1(\mathcal{X})$, representing that guessing an unseen value requires at least an attacker willing to guess all known values.

All of these estimates are somewhat unsatisfactory because they don't allow us to distinguish between the estimated security of a new observation `encryption1` compared to `e5703572ae3c`, the latter of which intuitively seems much more difficult to guess. Solving this problem inherently relies on semantic evaluation of the newly-seen event, which is out of scope.

4.2 Stability of metrics

All of the proposed metrics will produce variable results for low-frequency events in a sample. The logarithmic nature of the estimators damps this problem to a large extent: if the hapax legomenon password sapo26 occurred two more times in the RockYou data set, tripling its observed frequency, its strength estimate would decrease by only 1.59, 2.22 and 2.55 bits for $S^{\mathrm{P}}_{\mathrm{RY}}$, $S^{\mathrm{I}}_{\mathrm{RY}}$, and $S^{\mathrm{G}}_{\mathrm{RY}}$, respectively.

It is straightforward to establish bounds on the worst case error when changing an event's observed probability from $p \to p' = p + \Delta p$. For S^{P}, the estimate can change from $\lg p$ to $\lg p'$, a difference of at most $\mathrm{abs}\left(\lg \frac{p'}{p}\right)$ bits.

For the index metric the worst-case scenario is that changing from $p \to p' = p + \Delta p$ changes the index by N, the total number of events in the distribution, if all other events have probability $p \leq p^* \leq p + \Delta p$. In this case, the maximum number of events in the distribution is $N = \frac{1}{\min(p,p')}$. This gives a worst-case change of $\lg(2N-1) - \lg 0 \approx \lg \frac{2}{\min(p,p')}$.

The worst-case change for S^{G} occurs in the same situation but is just $\lg N - \lg \frac{1}{p'} = \lg \frac{1}{p} - \lg \frac{1}{p'} = \mathrm{abs}\left(\lg \frac{p'}{p}\right)$, exactly as the case was for S^{P}. This is an attractive property of S^{G}: it offers worst-case stability equivalent to S^{P} while having a better connection to real guessing attacks.

However, in the special case where $\mathcal{X}$ is a Zipf distribution For a Zipf distribution each event's probability is roughly proportional to the inverse of its index in the distribution raised to a constant s:

$$p_x \propto \left(\frac{1}{i_x}\right)^s$$

Thus, if an event's probability increases by a constant factor k, its index should decrease by a factor of $k^{\frac{-1}{s}}$. This will decrease S^{P} by $\lg k$ bits and decrease S^{I} by $\frac{\lg k}{s}$ bits. For the classic Zipf distribution with $s \approx 1$, this means that changing an event's probability by k will affect $S^{\mathrm{I}}_{\mathcal{X}}$ and $S^{\mathrm{P}}_{\mathcal{X}}$ by exactly the same amount. While we reject the hypothesis that passwords are produced by a simple Zipfian distribution [5], this is a rough justification for why we don't expect $S^{\mathrm{I}}_{\mathcal{X}}$ to be highly unstable in practice.

5 Example estimates for individual passwords

Example values for the proposed strength metrics are given in Table 1 for passwords in the RockYou data set. Overall, the differences between S^{P}, S^{I}, and S^{G} are moderate with the exception of very common passwords, which receive significantly lower strength estimates by S^{I}. For much of the distribution, S^{G} provides estimates in between those of S^{P} and S^{I}, until the region of less-common passwords for which S^{G} is lower as it incorporates an attacker's ability to stop early upon success.

The fact that $S^{\mathrm{I}} < S^{\mathrm{P}}$ holds in every row is not a coincidence. Because the elements are ordered by probability, an event's index will always be lower in a

x	$\lg(i_x)$	f_x	$S^{\mathrm{P}}_{\mathrm{RY}}$	$S^{\mathrm{I}}_{\mathrm{RY}}$	$S^{\mathrm{G}}_{\mathrm{RY}}$	S^{NIST}
123456	0	290729	6.81	0.00	6.81	14.0
12345	1	79076	8.69	1.58	7.46	12.0
password	2	59462	9.10	2.81	8.01	18.0
rockyou	3	20901	10.61	3.91	8.68	16.0
jessica	4	14103	11.17	4.95	9.42	16.0
butterfly	5	10560	11.59	5.98	10.08	19.5
charlie	6	7735	12.04	6.99	10.71	16.0
diamond	7	5167	12.62	7.99	11.30	16.0
freedom	8	3505	13.18	9.00	11.88	16.0
letmein	9	2134	13.90	10.00	12.48	16.0
bethany	10	1321	14.59	11.00	13.09	16.0
lovers1	11	739	15.43	12.00	13.74	22.0
samanta	12	389	16.35	13.00	14.42	16.0
123456p	13	207	17.27	14.00	15.13	22.0
diving	14	111	18.16	15.00	15.87	14.0
flower23	15	63	18.98	16.00	16.62	24.0
scotty2hotty	16	34	19.87	17.02	17.38	30.0
lilballa	17	18	20.79	18.01	18.13	18.0
robbies	18	9	21.79	19.06	18.93	16.0
DANELLE	19	5	22.64	19.96	19.62	22.0
antanddeck06	20	3	23.37	20.84	20.30	30.0
babies8	21	2	23.96	21.78	21.00	22.0
sapo26	22	1	24.96	24.00	22.44	20.0
jcb82	23	0	23.77	24.00	22.65	18.0

Table 1. Example strength estimates for a selection of passwords from the RockYou data set. The estimator S^{NIST} is calculated using the NIST entropy estimation formula [3].

skewed distribution than in a uniform distribution with identical events, so we will always have $S^{\mathrm{I}} \leq S^{\mathrm{P}}$.

The entropy estimation formula proposed by NIST [3] is shown for comparison as S^{NIST} (though note that S^{NIST} doesn't meet either of our desired mathematical criteria for a strength metric). It fails for a few passwords which demonstrate the challenges of semantic evaluation: both scotty2hotty and antanddeck06 score highly by S^{NIST} for being long and including digits. Neither is particularly strong, however: scotty2hotty is a professional wrestler, while antanddeck06 is based on the name of a British comedy show. In contrast sapo26 is much shorter and rated 10 bits lower by S^{NIST}, but doesn't have a well-known real-world meaning.

Because we listed passwords in order of exponentially increasing index, we can test the Zipfian relationship on the difference between S^{P} and S^{I} using the data by comparing the ratio of differences for successive passwords x_2, x_1 in

Table 1:

$$s \approx \frac{S^{\mathrm{P}}_{\mathrm{RY}}(x_{i+1}) - S^{\mathrm{P}}_{\mathrm{RY}}(x_i)}{S^{\mathrm{I}}_{\mathrm{RY}}(x_{i+1}) - S^{\mathrm{I}}_{\mathrm{RY}}(x_i)}$$

For successive rows of the table, we get estimates for s ranging from 0.34 to 1.37. The average estimate, however, is $s = 0.76$, almost identical to the estimate we would get by computing s using only the first and last row of the table. Using the equivalence between the power-law and Zipf formulation that $a = 1 + \frac{1}{s}$, we would estimate $a = 2.31$ given $s = 0.76$. This is not a sound way of computing a Zipfian fit for the data set s in general, but the fact that it is plausible supports our hypothesis that S^{I} will be stable for realistic distributions which follow a (very rough) power-law approximation.

6 Application to small data sets

A second application of strenth metrics is to estimate the average strength given only a very small sample for which distribution-wide statistics [2] can't be computed. This method can only be accurate for data sets which are approximately drawn from the same population as the base distribution, though this limitation is equally true of evaluation by password cracking [10] or semantic evaluation [14].

If we interpret the small set of passwords as a sample from some larger distribution, we need to reason about the expected value of each strength metric. We've already shown in Equation 4 that $E\left[S^{\mathrm{P}}_{\mathcal{X}}(x) \middle| x \xleftarrow{\mathrm{R}} \mathcal{X}\right] = H_1(\mathcal{X})$. The expected value of S^{I} was too complicated to compute directly even for a uniform distribution. Similarly, the expected value of S^{G} is:

$$E\left[S^{\mathrm{G}}_{\mathcal{X}}(x) \middle| x \xleftarrow{\mathrm{R}} \mathcal{X}\right] = \int_0^1 \tilde{G}_\alpha(\mathcal{X}) \mathrm{d}\alpha \tag{7}$$

which doesn't appear to admit a simple analytic formula. Instead, we can only compute $E\left[S^{\mathrm{I}}_{\mathcal{X}}(x) \middle| x \xleftarrow{\mathrm{R}} \mathcal{X}\right]$ and $E\left[S^{\mathrm{G}}_{\mathcal{X}}(x) \middle| x \xleftarrow{\mathrm{R}} \mathcal{X}\right]$ directly for our reference distribution $\mathcal{X}$ and use this as a benchmark for comparison against a smaller distribution.

In Table 2 a variety of small password data sets for which cleartext passwords are available are evaluated using the RockYou data set as a baseline. None of the statistical metrics are obviously superior, though S^{G} is typically in between the values produced by the other two.

The NIST formula produces more plausible results when averaged than for individual passwords, correctly ranking the 2011 Twitter blacklist as much weaker than the other lists (though not as weak as the statistical estimates). The NIST formula also plausibly rates the foreign-language Hebrew data set lower than the statistical estimates, as it doesn't assume the passwords are in English like using RockYou as a baseline implicitly does.

In the myBart data set about two-thirds of users retained site-assigned random passwords. This set was rated highly by all of the metrics, being recognised

Dataset	M	% seen	$S^{\mathrm{P}}_{\mathrm{RY}}$	$S^{\mathrm{I}}_{\mathrm{RY}}$	$S^{\mathrm{G}}_{\mathrm{RY}}$	S^{NIST}
RockYou (baseline)	—	100.0%	21.15	18.79	18.75	19.82
small password sets						
70yx (sampled)	1000	34.0%	22.28	21.24	21.52	20.21
Fox	369	68.8%	20.95	18.99	19.33	19.28
Hebrew	1307	50.3%	21.25	19.63	20.14	17.46
Hotmail	11576	57.6%	21.82	20.29	20.43	18.21
myBart	2007	19.0%	22.93	22.37	22.54	23.53
MySpace	50546	59.5%	21.64	20.02	20.19	22.53
NATO-Books	11822	50.9%	21.66	20.17	20.47	19.35
Sony-BMG	41024	61.3%	20.93	19.10	19.53	19.87
malware dictionaries						
Conficker	190	96.8%	16.99	13.60	15.07	16.51
Morris	445	94.4%	18.62	15.68	16.56	15.27
blacklists						
Twitter-2010	404	7.9%	23.16	22.86	23.02	15.30
Twitter-2011	429	99.8%	15.11	11.31	13.46	15.27

Table 2. Average strength estimates for small lists of leaked passwords. The NIST entropy estimation formula [3] is listed as S^{NIST}.

inadvertently by the NIST formula because the site-assigned passwords always contained a number.

The largest difference in the rankings occurred for the Hotmail and MySpace data sets, which produced indistinguishable statistical estimates but differed by over 4 bits by the NIST formula. Examining the passwords, it appears that a good portion of the MySpace data was collected under a policy mandating non-alphabetic characters: password1 is the most popular password, over twice as popular as password, and most of the other top passwords include a number. Popular numeric passwords such as 123456 appear to have been banned under some of the collection rules, as they are less common than variants like 123456a. The Hotmail data set, on the other hand, appears to have had no restrictions. Because the NIST formula awards a constant 6 points to passwords with a mix of numbers and letters, the MySpace complexity policy significantly raises S^{NIST}. However, the statistical estimators suggest these passwords may not actually be much stronger by this policy as a large number of users simply append a digit (usually a 1 or 0) to a weak password. In this case, statistical strength metrics are less influenced by the effects of complexity requirements.

The NIST formula also struggled to recognise datasets of explicitly weak passwords. For example, it considers the Conficker password dictionary to contain stronger passwords than the the Morris password dictionary, though the Conficker list is a more modern, better attack dictionary (containing much weaker passwords). Similarly, the two versions of the Twitter blacklist are rated similarly by the NIST formula, but the statistical metrics identify the 2011 version as a vast improvement (again in that it contains weaker passwords).

References

1. Matt Bishop and Daniel V. Klein. Improving System Security via Proactive Password Checking. *Computers & Security*, 14(3):233–249, 1995.
2. Joseph Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *SP '12: Proceedings of the 2012 IEEE Symposium on Security and Privacy*, 2012.
3. William E. Burr, Donna F. Dodson, and W. Timothy Polk. Electronic Authentication Guideline. *NIST Special Publication 800-63*, 2006.
4. Claude Castelluccia, Markus Dürmuth, and Daniele Perito. Adaptive Password-Strength Meters from Markov Models. In *NDSS '12: Proceedings of the Network and Distributed System Security Symposium*, 2012.
5. Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-Law Distributions in Empirical Data. *SIAM Review*, 51:661–703, 2009.
6. Chris Davies and Chris Ganesan. BApasswd: A New Proactive Password Checker. In *Proceedings of the 16th National Computer Security Conference*, 1993.
7. Matteo Dell'Amico, Pietro Michiardi, and Yves Roudier. Password Strength: An Empirical Analysis. In *INFOCOM'10: Proceedings of the 29th Conference on Information Communications*, pages 983–991. IEEE, 2010.
8. William A. Gale and Geoffrey Sampson. Good-Turing Frequency Estimation Without Tears. *Journal of Quantitative Linguistics*, 2(3):217–237, 1995.
9. Mike Just and David Aspinall. Personal Choice and Challenge Questions: A Security and Usability Assessment. In *SOUPS '09: Proceedings of the 5th Symposium on Usable Privacy and Security*, 2009.
10. Patrick Gage Kelley, Saranga Komanduri, Michelle L. Mazurek, Rich Shay, Tim Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. Technical Report CMU-CyLab-11-008, Carnegie Mellon University, 2011.
11. Patrick Gage Kelley, Michelle L. Mazurek, Richard Shay, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Saranga Komanduri, and Serge Egelman. Of Passwords and People: Measuring the Effect of Password-Composition Policies. In *CHI '11: Proceedings of the 29th ACM SIGCHI Conference on Human Factors in Computing Systems*, 2011.
12. Arvind Narayanan and Vitaly Shmatikov. Fast Dictionary Attacks on Passwords Using Time-Space Tradeoff. In *CCS '05: Proceedings of the 12th ACM Conference on Computer and Communications Security*, pages 364–372. ACM, 2005.
13. Claude E. Shannon. A Mathematical Theory of Communication. In *Bell System Technical Journal*, volume 7, pages 379–423, 1948.
14. Richard Shay, Saranga Komanduri, Patrick Gage Kelley, Pedro Giovanni Leon, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Encountering Stronger Password Requirements: User Attitudes and Behaviors. In *SOUPS '10: Proceedings of the 6th Symposium on Usable Privacy and Security*. ACM, 2010.
15. Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords. In *CCS '10: Proceedings of the 17th ACM Conference on Computer and Communications Security*, pages 162–175. ACM, 2010.
16. Matt Weir, Sudhir Aggarwal, Breno de Medeiros, and Bill Glodek. Password Cracking Using Probabilistic Context-Free Grammars. In *SP '09: Proceedings of the 2009 IEEE Symposium on Security and Privacy*, pages 391–405. IEEE, 2009.

A Definition of α-guesswork G_α

Taken directly from the derivation in [2], the definition of $\tilde{G}_\alpha$ requires several parts. The α-work-factor μ_α reflects the required size μ of a dictionary needed to have a cumulative probability α of success in an optimal guessing attack:

$$\mu_\alpha(\mathcal{X}) = \min\left\{ \mu \middle| \sum_{i=1}^{\mu} p_i \geq \alpha \right\} \tag{8}$$

This metric doesn't account for the average number of guesses per account, which will be lower since the attacker is able to stop early after correct guesses. The α-guesswork G_α reflects the average number of guesses per account:

$$G_\alpha(\mathcal{X}) = (1 - \lceil\!\lceil \alpha \rceil\!\rceil) \cdot \mu_\alpha(\mathcal{X}) + \sum_{i=1}^{\mu_\alpha(\mathcal{X})} p_i \cdot i \tag{9}$$

Note that a rounded-up $\lceil\!\lceil \alpha \rceil\!\rceil$ is used to reflect that the actual probability of success after $\mu_a lpha$ guesses may be more than α:

$$\lceil\!\lceil \alpha \rceil\!\rceil = \sum_{i=1}^{\mu_\alpha(\mathcal{X})} p_i \tag{10}$$

Finally, G_α is converted to bits by finding the size of a uniform distribution which would have an equivalent value of G_α and taking a logarithm:

$$\tilde{G}_\alpha(\mathcal{X}) = \lg\left[\frac{2 \cdot G_\alpha(\mathcal{X})}{\lceil\!\lceil \alpha \rceil\!\rceil} - 1\right] - \lg(2 - \lceil\!\lceil \alpha \rceil\!\rceil) \tag{11}$$

B Proof of expected sum for naive index strength metric $S^{\mathrm{I}}(x)$ for the uniform distribution

As claimed in Section 3.2, using the definition from Equation 5:

$$S^{\mathrm{I}}_{\mathcal{X}}(x) = \lg\left(2 \cdot i_x - 1\right)$$

and randomly assigning an ordering to the uniform distribution does not produce an expected value of $\lg N$, but $\approx \lg N - (\lg e - 1)$.

Proof. We first take the expectation:

$$\begin{aligned} E\left[S^{\mathrm{I}}_{\mathcal{X}}(x)\middle|\, x \xleftarrow{\mathrm{R}} \mathcal{U}_N\right] &= \sum_{i=1}^{N} \frac{1}{N} \cdot \lg\left(2 \cdot i - 1\right) \\ &= \frac{1}{N} \cdot \left(\lg 1 + \lg 3 + \lg 5 + \ldots + \lg(2N-1)\right) \\ &= \frac{1}{N} \cdot \lg(1 \cdot 3 \cdot 5 \cdot \ldots \cdot (2N-1)) \\ &= \frac{1}{N} \cdot \lg\left(\frac{(2N)!}{2 \cdot 4 \cdot 6 \cdot \ldots \cdot 2N}\right) \\ &= \frac{1}{N} \cdot \lg\left(\frac{(2N)!}{2^N \cdot N!}\right) \end{aligned}$$

We can use Stirling's approximation $\ln N! \sim N \ln N - N$, converting the base to get $\lg N! \sim N \lg N - N \lg e$:

$$\begin{aligned} E\left[S^{\mathrm{I}}_{\mathcal{X}}(x)\middle|\, x \xleftarrow{\mathrm{R}} \mathcal{U}_N\right] &= \frac{1}{N} \cdot \lg\left(\frac{(2N)!}{2^N \cdot N!}\right) \\ &= \frac{1}{N} \cdot \left(\lg(2N)! - \lg N! - \lg 2^N\right) \\ &\approx \frac{1}{N} \cdot \left(2N \lg 2N - 2N \lg e - N \lg N + N \lg e - N\right) \\ &= \frac{1}{N} \cdot \left(2N \lg N + 2N - 2N \lg e - N \lg N + N \lg e - N\right) \\ &= \frac{1}{N} \cdot \left(N \lg N + N - N \lg e\right) \\ &= \lg N - (\lg e - 1) \end{aligned}$$