The science of guessing: analyzing an anonymized corpus of 70 million passwords

Joseph Bonneau
Computer Laboratory
University of Cambridge
jcb82@cl.cam.ac.uk

***Abstract*—We report on the largest corpus of user-chosen passwords ever studied, consisting of anonymized password histograms representing almost 70 million Yahoo! users, mitigating privacy concerns while enabling analysis of dozens of subpopulations based on demographic factors and site usage characteristics. This large data set motivates a thorough statistical treatment of estimating guessing difficulty by sampling from a secret distribution. In place of previously used metrics such as Shannon entropy and guessing entropy, which cannot be estimated with any realistically sized sample, we develop partial guessing metrics including a new variant of guesswork parameterized by an attacker's desired success rate. Our new metric is comparatively easy to approximate and directly relevant for security engineering. By comparing password distributions with a uniform distribution which would provide equivalent security against different forms of guessing attack, we estimate that passwords provide fewer than 10 bits of security against an online, trawling attack, and only about 20 bits of security against an optimal offline dictionary attack. We find surprisingly little variation in guessing difficulty; every identifiable group of users generated a comparably weak password distribution. Security motivations such as the registration of a payment card have no greater impact than demographic factors such as age and nationality. Even proactive efforts to nudge users towards better password choices with graphical feedback make little difference. More surprisingly, even seemingly distant language communities choose the same weak passwords and an attacker never gains more than a factor of 2 efficiency gain by switching from the globally optimal dictionary to a population-specific lists.**

***Keywords*-computer security; authentication; statistics; information theory; data mining;**

I. Introduction

Text passwords have dominated human-computer authentication since the 1960s [1] and been derided by security researchers ever since, with Multics evaluators singling passwords out as a weak point in the 1970s [2]. Though many password cracking studies have supported this claim [3]–[7], there is still no consensus on the actual level of security provided by passwords or even on the appropriate metric for measuring security. The security literature lacks sound methodology to answer elementary questions such as "do older users or younger users choose better passwords?" Of more concern for security engineers, it remains an open question the extent to which passwords are weak due to a lack of motivation or inherent user limitations.

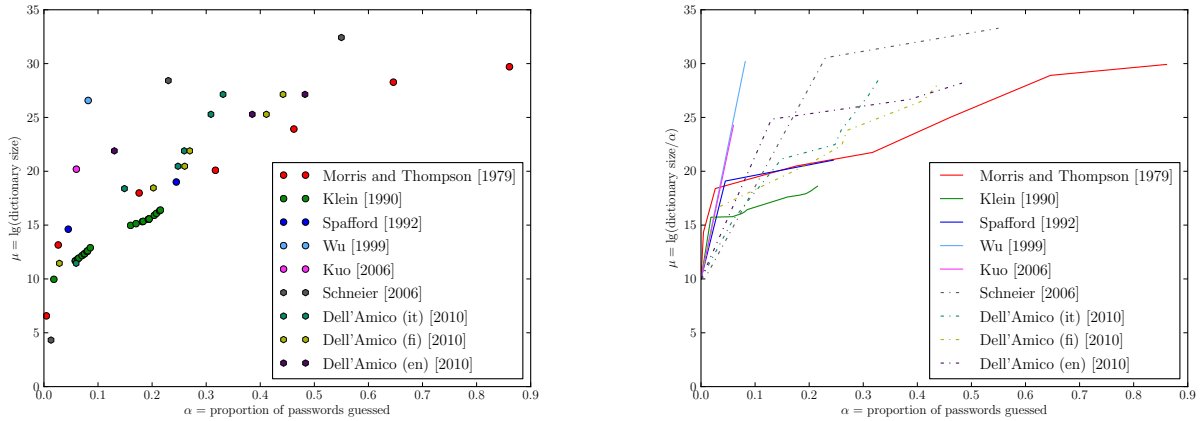
The mass deployment of passwords on the Internet may provide sufficient data to address these questions. So far, large-scale password data has arisen only from security breaches such as the leak of 32 M passwords from the gaming website RockYou in 2009 [7], [8]. Password corpora have typically been analyzed by simulating adversarial password cracking, leading to sophisticated cracking libraries but limited understanding of the underlying distribution of passwords (see Section II). Our goal is to bring the evaluation of large password data sets onto sound scientific footing by collecting a massive password data set legitimately and analyzing it in a mathematically rigorous manner.

This requires retiring traditional, inappropriate metrics such as Shannon entropy and guessing entropy which don't model realistic attackers and aren't approximable using sampled data. Our first contribution (Section III) is to formalize improved metrics for evaluating the guessing difficulty of a skewed distribution of secrets, such as passwords, introducing α-guesswork as a tunable metric which can effectively model different types of practical attack.

Our second contribution is a novel privacy-preserving approach to collecting a password distribution for statistical analysis (Section IV). By hashing each password at the time of collection with a secret key that is destroyed prior to our analysis, we preserve the password histogram exactly with no risk to user privacy.

Even with millions of passwords, sample size has surprisingly large effects on our calculations due to the large number of very infrequent passwords. Our third contribution (Section V) is to adapt techniques from computational linguistics to approximate guessing metrics using a random sample. Fortunately, the most important metrics are also the best-approximated by sampled data. We parametrically extend our approximation range by fitting a generalized inverse Gaussian-Poisson (Sichel) distribution to our data.

Our final contribution is to apply our research to a massive corpus representing nearly 70 M users, the largest ever collected, with the cooperation of Yahoo! (Section VI). We analyze the effects of many demographic factors, but the password distribution is remarkably stable and security estimates in the 10–20 bit range emerge across every subpopulation we considered. We conclude from our research (Section VII) that we are yet to see compelling evidence that motivated users can choose passwords which resist guessing by a capable attacker.



(a) Historical cracking efficiency, raw dictionary size

(b) Historical cracking efficiency, equivalent dictionary size

Figure 1. The size of cracking dictionaries is plotted logarithmically against the success rate achieved in Figure 1a. In Figure 1b, the dictionary sizes are adjusted to incorporate the inherent need for more guesses to crack more passwords. Circles and solid lines represent operating system user passwords, squares and dashed lines represent web passwords.

II. Historical evaluations of password security

It has long been of interest to analyze how secure passwords are against guessing attacks, dating at least to Morris and Thompson's seminal 1979 analysis of 3,000 passwords [3]. They performed a rudimentary dictionary attack using the system dictionary and all 6-character strings and recovered 84% of available passwords. They also reported some basic statistics such as password lengths (71% were 6 characters or fewer) and frequency of non-alphanumeric characters (14% of passwords). These two approaches, password cracking and semantic evaluation, have been the basis for dozens of studies in the thirty years since.

A. Cracking evaluation

The famous 1988 Morris worm propagated in part by guessing passwords using a 350-word password dictionary and several rules to modify passwords [9]. The publicity surrounding the worm motivated independent studies by Klein and Spafford which re-visited password guessing [4], [5]. Both studies broke 22–24% of passwords using more sophisticated dictionaries such as lists of names, sports teams, movies and so forth. Password cracking evolved rapidly in the years after these studies, with dedicated software tools like John the Ripper emerging in the 1990s which utilize *mangling* rules to turn a single password like "john" into variants like "John", "J0HN", and "nhoj." [10]. Research on mangling rules has continued to evolve; the current state of the art by Weir et al. [11] automatically generates mangling rules from a large training set of known passwords.

Later studies have often utilized these tools to perform dictionary attacks as a secondary goal, such as Wu's study of password cracking against Kerberos tickets in 1999 [12] and Kuo et al.'s study of mnemonic passwords in 2006 [13], which recovered 8% and 11% of passwords, respectively.

Recently, large-scale password leaks from compromised websites have provided a new source of data for cracking evaluations. For example, Schneier analyzed about 50,000 passwords obtained via phishing from MySpace in 2006 [6]. A more in-depth study was conducted by Dell'Amico et al., who studied the MySpace passwords as well as those of two other websites using a large variety of different dictionaries [7]. A very large data set of 32M passwords leaked from RockYou in 2009, which Weir et al. studied to examine the effects of password-composition rules on cracking efficiency [8].

Reported numbers on password cracking efficiency vary substantially between different studies, as shown in Figure 1. Most studies have broken 20–50% of accounts with dictionary sizes in the range of 2^{20}–2^{30}. All studies see diminishing returns for larger dictionaries. This is clear in Figure 1b, which adjusts dictionary sizes based on the percentage of passwords cracked so that the degree of upward slope reflects only decreasing efficiency. This concept will motivate our statistical guessing metrics in Section III-E.

There is little data on the efficiency of small dictionaries as most studies employ the largest dictionary they can process. Klein's study, which attempted to identify highly efficient sub-dictionaries, is a notable exception [4]. There is also little data on the size of dictionary required to break a large majority of passwords—only Morris and Thompson broke more than 50% of available passwords[1] and their results may be too dated to apply to modern passwords.

B. Semantic evaluations

In addition to cracking research, there have been many studies on the semantics of passwords with psychologists

[1] A 2007 study by Cazier and Medlin claimed to break 99% of passwords at an e-commerce website, but details of the dictionary weren't given [14].

year	study	length	% digits	% special
1989	Riddle et al. [15]	4.4	3.5	—
1992	Spafford [5]	6.8	31.7	14.8
1999	Wu [12]	7.5	25.7	4.1
1999	Zviran and Haga [18]	5.7	19.2	0.7
2006	Cazier and Medlin [14]	7.4	35.0	1.3
2009	*RockYou leak* [19]	7.9	54.0	3.7

Table I
COMMONLY ESTIMATED ATTRIBUTES OF PASSWORDS

and linguists being interested as well as computer security researchers. This approach can be difficult as it either requires user surveys, which may produce unrealistic password choices, or direct access to unhashed passwords, which carries privacy concerns. Riddle et al. performed linguistic analysis of 6,226 passwords in 1989, classifying them into categories such as names, dictionary words, or seemingly random strings [15]. Cazier et al. repeated this process in 2006 and found that hard-to-classify passwords were also the hardest to crack [14].

Password structure was formally modeled by Weir et al. [11] using a context-free grammar to model the probability of different constructions being chosen. Password creation has also been modeled as a character-by-character Markov process, first by Narayanan and Shmatikov [16] for password cracking and later by Castelluccia et al. [17] to train a pro-active password checker.

Thus methodology for analyzing password structure has varied greatly, but a few basic data points like average length and types of characters used are typically reported, as summarized in Table I. The estimates vary so widely that it is difficult to infer much which is useful in systems design. The main trends are a tendency towards 6-8 characters of length and a strong dislike of non-alphanumeric characters in passwords.[2] Many studies have also attempted to determine the number of users which appear to be choosing random passwords, or at least passwords without any obvious meaning to a human examiner. Methodologies for estimating this vary as well, but most studies put it in the 10–40% range.

Elements of password structure, such length or the presence of digits, upper-case, or non-alphanumeric characters can be used to estimate the "strength" of a password, often measured in bits and often referred to imprecisely as "entropy".[3] This usage was cemented by the 2006 FIPS Electronic Authentication Guideline [20], which provided a "rough rule of thumb" for estimating entropy from password characteristics such as length and type of characters used. This standard has been used in several password studies with too few samples to compute statistics on the entire distribution [21]–[23]. More systematic formulas have been proposed, such as one by Shay et al. [22] which adds entropy from different elements of a password's structure.

C. Problems with previous approaches

Three decades of work on password guessing has produced sophisticated cracking tools and many disparate data points, but a number of methodological problems continue to limit scientific understanding of password security:

1) Comparability: Authors rarely report cracking results in a format which is straightforward to compare with previous benchmarks. To our knowledge, Figure 1 is the first comparison of different data points of dictionary size and success rate, though direct comparison is difficult since authors all report efficiency rates for different dictionary sizes. Password cracking tools only loosely attempt to guess passwords in decreasing order of likeliness, introducing imprecision into reported dictionary sizes. Worse, some studies report the running time of cracking software instead of dictionary size [14], [24], [25], making comparison difficult.

2) Repeatability: Precisely reproducing password cracking results is difficult. John the Ripper [10], used in most publications of the past decade, has been released in 21 different versions since 2001 and makes available 20 separate word lists for use (along with many proprietary ones), in addition to many configuration options. Other studies have used proprietary password-cracking software which isn't available to the research community [6], [14]. Thus nearly all studies use dictionaries varying in content and ordering, making it difficult to exactly re-create a published attack to compare its effectiveness against a new data set.

3) Evaluator dependency: Password-cracking results are inherently dependent on the appropriateness of the dictionary and mangling rules to the data set under study. Dell'Amico et al. [7] demonstrated this problem by applying language-specific dictionaries to data sets of passwords in different languages and seeing efficiency vary by 2–3 orders of magnitude. They also evaluated the same data set as Schneier three years earlier [6] and achieved two orders of magnitude better efficiency simply by choosing a better word list. Thus it is difficult to separate the effects of more-carefully chosen passwords from the use of a less appropriate dictionary. This is particularly challenging in data-slicing experiments [8], [23] which require simulating an equally good dictionary attack against each subpopulation.

4) Unsoundness: Estimating the entropy of a password distribution from structural characteristics is mathematically dubious, as we will demonstrate in Section III-D, and inherently requires making many assumptions about password selection. In practice, entropy estimates have performed poorly as predictors of empirical cracking difficulty [8], [23].

[2]It is often suggested that users avoid characters which require multiple keys to type, but this doesn't seem to have been formally established.

[3]This terminology is mathematically incorrect because entropy (see Sections III-A and III-B) measures a complete probability distribution, not a single event (password). The correct metric for a single event is *self-information* (or *surprisal*). This is perhaps disfavored because it is counter-intuitive: passwords should avoid including information like names or addresses, so high-information passwords sound weak.

III. Mathematical metrics of guessing difficulty

Due to the problems inherent to password cracking simulations or semantic evaluation, we advocate security metrics that rely only on the statistical distribution of passwords. While this approach requires large data sets, it eliminates bias from password-cracking software by always modeling a best-case attacker, allowing us to assess and compare the inherent security of a given distribution.

Mathematical notation: We denote a probability distribution with a calligraphic letter, such as $\mathcal{X}$. We use lower-case x to refer to a specific event in the distribution (an individual password). The probability of x is denoted p_x. Formally, a distribution is a set of events $x \in \mathcal{X}$, each with an associated probability $0 < p_x \leq 1$, such that $\sum p_x = 1$. We use N to denote the total number of possible events in $\mathcal{X}$.

We often refer to events by their index i, that is, their rank by probability in the distribution with the most probable having index 1 and the least probable having index N. We refer to the i^{th} most common event as x_i and call its probability p_i. Thus, the probabilities of the events in $\mathcal{X}$ form a monotonically decreasing sequence $p_1 \geq p_2 \geq \ldots \geq p_N$.

We denote an unknown variable as X, denoting $X \xleftarrow{\text{R}} \mathcal{X}$ if it is drawn at random from $\mathcal{X}$.

Guessing model: We model password selection as a random draw $X \xleftarrow{\text{R}} \mathcal{X}$ from an underlying password distribution $\mathcal{X}$. Though $\mathcal{X}$ will vary depending on the population of users, we assume that $\mathcal{X}$ is completely known to the attacker. Given a (possibly singleton) set of unknown passwords $\{X_1, X_2, \ldots X_k\}$, we wish to evaluate the efficiency of an attacker trying to identify the unknown passwords X_i given access to an oracle for queries of the form "is $X_i = x$?"

A. Shannon entropy

Intuitively, we may first think of the *Shannon entropy*:

$$H_1(\mathcal{X}) = \sum_{i=1}^{N} -p_i \lg p_i \tag{1}$$

as a measure of the "uncertainty" of X to an attacker. Introduced by Shannon in 1948 [26], entropy appears to have been ported from cryptographic literature into studies of passwords before being used in FIPS guidelines [20].

It has been demonstrated that H_1 is mathematically inappropriate as a measure guessing difficulty [27]–[30]. It in fact quantifies the average number of subset membership queries of the form "Is $X \in \mathcal{S}$?" for arbitrary subsets $\mathcal{S} \subseteq \mathcal{X}$ needed to identify X.[4] For an attacker who must guess individual passwords, Shannon entropy has no direct correlation to guessing difficulty.[5]

[4]The proof of this is a straightforward consequence of Shannon's source coding theorem [26]. Symbols $X \xleftarrow{\text{R}} \mathcal{X}$ can be encoded using a Huffman code with average bit length $\leq H_1(\mathcal{X}) + 1$, of which the adversary can learn one bit at a time with subset membership queries.

[5]H_1 has further been claimed to correlate poorly with password cracking difficulty [8], [23], though the estimates of H_1 used cannot be relied upon.

B. Rényi entropy and its variants

Rényi entropy H_n is a generalization of Shannon entropy [31] parametrized by a real number $n \geq 0$:[6]

$$H_n(\mathcal{X}) = \frac{1}{1-n} \lg \left(\sum_{i=1}^{N} p_i^n \right) \tag{2}$$

In the limit as $n \to 1$, Rényi entropy converges to Shannon entropy, which explains why Shannon entropy is denoted H_1. Note that H_n is a monotonically decreasing function of n. We are most interested in two special cases:

1) Hartley entropy H_0: For $n = 0$, Rényi entropy is:

$$H_0 = \lg N \tag{3}$$

Introduced prior to Shannon entropy [32], H_0 measures only the size of a distribution and ignores the probabilities.

2) Min-entropy H_∞: As $n \to \infty$, Rényi entropy is:

$$H_\infty = -\lg p_1 \tag{4}$$

This metric is only influenced by the probability of the most likely symbol in the distribution, hence the name. This is a useful worst-case security metric for human-chosen distributions, demonstrating security against an attacker who only guesses the most likely password before giving up. H_∞ is a lower bound for all other Rényi entropies and indeed all of the metrics we will define.

C. Guesswork

A more applicable metric is the expected number of guesses required to find X if the attacker proceeds in optimal order, known as *guesswork* or *guessing entropy* [27], [30]:

$$G(\mathcal{X}) = E\left[\#_{\text{guesses}}(X \xleftarrow{\text{R}} \mathcal{X}) \right] = \sum_{i=1}^{N} p_i \cdot i \tag{5}$$

Because G includes all probabilities in $\mathcal{X}$, it models an attacker who will exhaustively guess even exceedingly unlikely events which can produce absurd results. For example, in the RockYou data set over twenty users (more than 1 in 2^{21}) appear to use 128-bit pseudorandom hexadecimal strings as passwords. These passwords alone ensure that $G(\text{RockYou}) \geq 2^{106}$. Thus G provides little insight into practical attacks and furthermore is difficult to estimate from sampled data (see Section V).

D. Partial guessing metrics

Guesswork and entropy metrics fail to model the tendency of real-world attackers to cease guessing against the most difficult accounts. As discussed in Section II, cracking evaluations typically report the fraction of accounts broken by a given attack and explicitly look for weak subspaces of passwords to attack. Having many accounts to attack is an

[6]Rényi entropy is traditionally denoted H_α; we use H_n to avoid confusion with our primary use of α as a desired success rate.

important resource for a real attacker, as it enables a *partial guessing* attack which trades a lower proportion of accounts broken for increased guessing efficiency.

Formally, if Eve must sequentially guess each of k passwords drawn from $\mathcal{X}$, she will need $\sim k \cdot G(\mathcal{X})$ guesses on average. However, a second guesser Mallory willing to break only $\ell < k$ of the passwords can do much better with the optimal strategy of first guessing the most likely password for all k accounts, then the second-most likely value and so on. As ℓ decreases, Mallory's efficiency increases further as the attack can omit progressively more low-probability passwords. For large values of k and ℓ, Mallory will only need to guess the most popular β passwords such that $\sum_{i=1}^{\beta} p_i \geq \alpha$, where $\alpha = \frac{\ell}{k}$. There are several possible metrics for measuring guessing in this model:

1) β-success-rate: A very simple metric, first formally defined by Boztaş [29], measures the expected success for an attacker limited to β guesses per account:

$$\lambda_\beta(\mathcal{X}) = \sum_{i=1}^{\beta} p_i \tag{6}$$

2) α-work-factor: A related metric, first formalized by Pliam [28], evaluates the fixed number of guesses per account needed to break a desired proportion α of accounts.

$$\mu_\alpha(\mathcal{X}) = \min\left\{ j \middle| \sum_{i=1}^{j} p_i \geq \alpha \right\} \tag{7}$$

If $\mu_\alpha(\mathcal{X}) = n$, this tells us that an attacker must use an optimal dictionary of n entries to have a probability α of breaking an individual account, or equivalently to break an expected fraction α of many accounts.

3) α-guesswork: While λ_β and μ_α are closer to measuring real guessing attacks, both ignore the fact that a real attacker can stop early after successful guesses. While making up to μ_α guesses per account will enable breaking a fraction α of accounts, some will require fewer than μ_α guesses. We introduce a new metric to reflect the expected number of guesses per account to achieve a success rate α:

$$G_\alpha(\mathcal{X}) = (1 - \lambda_{\mu_\alpha}) \cdot \mu_\alpha + \sum_{i=1}^{\mu_\alpha} p_i \cdot i \tag{8}$$

We use λ_{μ_α} in place of α to round up to the proportion of passwords actually covered by μ_α guesses. Note that the traditional guesswork metric G is a special case G_1 of this metric with $\alpha = 1$. We could equivalently define G_β for an attacker limited to β guesses, but this is less useful as for small β the effect of stopping early is negligible.

E. Effective key-length metrics

While λ_β, μ_α and G_α are not measures of entropy, it is convenient to convert them into units of bits. This enables direct comparison of all metrics as a logarithmically scaled attacker workload which is intuitive to programmers and cryptographers. This can be thought of as an "effective key-length" as it represents the size of a randomly chosen cryptographic key which would give equivalent security.[7]

We convert each metric by calculating the logarithmic size of a discrete uniform distribution $\mathcal{U}_N$ with $p_i = \frac{1}{N}$ for all $1 \leq i \leq N$ which has the same value of the guessing metric. For β-success-rate, since we have $\lambda_\beta(\mathcal{U}_N) = \frac{\beta}{N}$ we say that another distribution $\mathcal{X}$ is equivalent with respect to λ_β to a uniform distribution of size $N = \frac{\beta}{\lambda_\beta(\mathcal{X})}$. We take the logarithm of this size to produce our effective key-length metric $\tilde{\lambda}_\beta$, using a tilde to denote the conversion to bits:

$$\tilde{\lambda}_\beta(\mathcal{X}) = \lg\left(\frac{\beta}{\lambda_\beta(\mathcal{X})}\right) \tag{9}$$

The conversion formula for α-work-factor is related:

$$\tilde{\mu}_\alpha(\mathcal{X}) = \lg\left(\frac{\mu_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}}\right) \tag{10}$$

Again, we use λ_{μ_α} in place of α in the denominator because μ_α increases as a step function as α increases. Without this correction, $\tilde{\mu}_\alpha$ would decrease over each range of α where μ_α is constant, giving a misleading over-estimate of security. Using λ_{μ_α} effectively rounds up to the next value of α which would require additional guesses to cover, ensuring that $\tilde{\mu}_\alpha$ is monotonically increasing.

To convert G_α, we consider that an attacker desiring to break a proportion α of accounts will average G_α guesses per account, or one successful guess per $\frac{G_\alpha}{\alpha}$ guesses. Against the uniform distribution $\mathcal{U}_N$, an attacker will break an account every $\frac{N+1}{2}$ guesses, giving us the formula:

$$\tilde{G}_\alpha(\mathcal{X}) = \lg[\frac{2 \cdot G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} - 1] + \lg\frac{1}{2 - \lambda_{\mu_\alpha}} \tag{11}$$

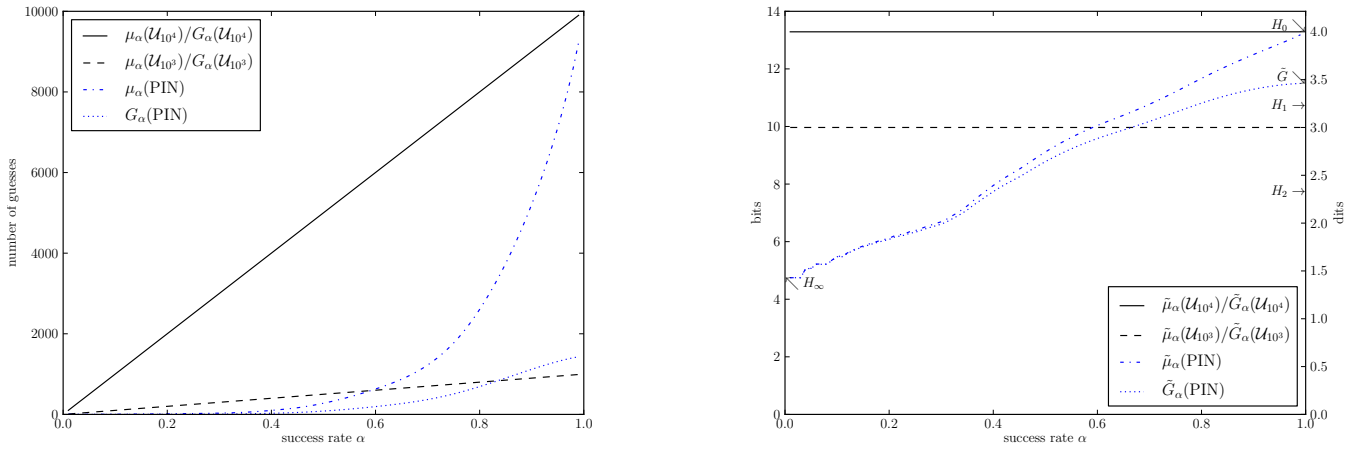
using the same correction for α as we did for $\tilde{\mu}_\alpha$ to achieve monotonicity, and the correction factor $\lg\frac{1}{2-\lambda_{\mu_\alpha}}$ to make the metric constant for a uniform distribution.

F. Relationship between metrics

We enumerate a few useful relationships between different metrics in Table II. Note that for a discrete uniform distribution $\mathcal{U}_N$, all of the metrics H_n, $\tilde{G}_\alpha$, $\tilde{\lambda}_\beta$ and $\tilde{\mu}_\alpha$ are equivalent. This validates the definitions and demonstrates why more complicated guessing metrics have rarely come up in cryptographic literature, as they provide no additional information for uniform distributions.

Massey proved that $\tilde{G}_1 \geq H_1 - 1$ [30], which is sometimes used to justify H_1 as a guessing metric. However, several negative results show that neither H_1 nor $\tilde{G}_1$ can provide any lower bound on partial guessing. Theorems proved by Pliam [28], Boztaş [29], and Bonneau [34] demonstrate an unbounded gap: for any desired success rate $\alpha < 1$, it is possible to construct a distribution $\mathcal{X}$ such that $\tilde{\mu}_\alpha(\mathcal{X}) +$

[7] Boztaş introduced the term effective key-length specifically to refer to $\tilde{\mu}_{0.5}$ [29]. We extend the notion here to all of our metrics.



(a) μ_α, G_α (number of guesses)

(b) μ_α, G_α (effective key-length)

Figure 2. Two ways of comparing the guessing difficulty of user-chosen 4-digit PINs [33] against uniform distributions of size 10,000 and 1,000 ($\mathcal{U}_{10^4}$ and $\mathcal{U}_{10^3}$, respectively). Fig. 2a plots the dictionary size μ_α needed to have a chance of success α as well as the expected number of guesses per account G_α. Fig. 2b converts both metrics into an effective key-length, enabling visual comparison across the entire range of α. Traditional single-point metrics H_0, H_1, H_2, H_∞ and $\tilde{G}$ are also marked for comparison. Note that $\tilde{\mu}_\alpha$ and $\tilde{G}_\alpha$ are horizontal lines for uniform distributions; an attacker gains no efficiency advantage from lowering his desired success rate α.

$m \leq H_1(\mathcal{X})$ and $\tilde{\mu}_\alpha(\mathcal{X}) + m \leq \tilde{G}_1(\mathcal{X})$ for any separation parameter m. Furthermore, for any $\alpha_1 < \alpha_2$ a distribution $\mathcal{X}$ can be found with $\tilde{\mu}_{\alpha_1}(\mathcal{X}) + m \leq \tilde{\mu}_{\alpha_2}(\mathcal{X})$ for any m. These results easily extend to $\tilde{G}_\alpha$ using the bounds listed in Table II and related results can be proved for $\tilde{\lambda}_\beta(\mathcal{X})$.

equivalences	
$\forall_n \quad H_n(\mathcal{U}_N) = \lg N$	all metrics equal for $\mathcal{U}$
$\forall_\beta \quad \tilde{\lambda}_\beta(\mathcal{U}_N) = \lg N$	all metrics equal for $\mathcal{U}$
$\forall_\alpha \quad \tilde{\mu}_\alpha(\mathcal{U}_N) = \lg N$	all metrics equal for $\mathcal{U}$
$\forall_\alpha \quad \tilde{G}_\alpha(\mathcal{U}_N) = \lg N$	all metrics equal for $\mathcal{U}$
$H_0 = \tilde{\mu}_1 = \tilde{\lambda}_N = \lg N$	metrics depending only on N
$H_\infty = \tilde{\mu}_{\alpha \leq p_1} = \tilde{\lambda}_1 = -\lg p_1$	metrics depending only on p_1
bounds	
$H_\infty \leq \tilde{G}_\alpha, \tilde{\mu}_\alpha, \tilde{\lambda}_\beta$	H_∞ is abs. lower bound
$\tilde{G}_\alpha, \tilde{\mu}_\alpha, \tilde{\lambda}_\beta \leq H_0$	H_0 is abs. upper bound
$\tilde{G}_\alpha \leq \tilde{\mu}_\alpha$	straightforward proof
$\tilde{G}_\alpha - \tilde{\mu}_\alpha \leq \lg(1-\alpha)$	straightforward proof
monotonicity	
$H_\infty \leq \ldots \leq H_1 \leq H_0$	H_n decreasing with n
$\tilde{\lambda}_\beta \leq \tilde{\lambda}_{\beta+\epsilon}$	$\tilde{\lambda}_\beta$ increasing with β
$\tilde{\mu}_\alpha \leq \tilde{\mu}_{\alpha+\epsilon}$	$\tilde{\mu}_\alpha$ increasing with α
$\tilde{G}_\alpha \leq \tilde{G}_{\alpha+\epsilon}$	$\tilde{G}_\alpha$ increasing with α

Table II
RELATIONS BETWEEN GUESSING METRICS

G. Application in practical security evaluation

For an online attacker we can use $\tilde{\lambda}_\beta$ with β equal to the guessing limits imposed by the system. There is no standard for β, with 10 guesses recommended by usability studies [35], 3 by FIPS guidelines [20], and a variety of values (often ∞) seen in practice [36]. Sophisticated rate-limiting schemes may allow a probabilistic number of guesses [37]. We consider $\tilde{\lambda}_{10}$ a reasonable benchmark for resistance to online guessing, though $\tilde{\lambda}_1 = H_\infty$ is a conservative choice as a lower bound for all metrics proposed.

The separation results of Section III-F mean that for brute-force attacks we can't rely on any single value of α; each value provides information about a fundamentally different attack scenario. For a complete picture, we can consider $\tilde{\mu}_\alpha$ or $\tilde{G}_\alpha$ across all values of α. We can plot this as the *guessing curve* for a distribution, as seen in Figure 2.

For offline attacks, where an adversary is limited only by time and computing power, we might consider $\tilde{\mu}_\alpha$ or $\tilde{G}_\alpha$ for a standard value such as 0.5 as a benchmark ($\tilde{\mu}_{0.5}$ was originally suggested by [29]). While $\tilde{G}_\alpha$ more directly measures the efficiency of a guessing attack, $\tilde{\mu}_\alpha$ can be advantageous in practice because it is simpler to compute. In particular, it can be computed using previously published cracking results reported as "a dictionary of size μ compromised a fraction α of available accounts," as plotted in Figure 1b. Furthermore, the difference between the metrics is only significant for higher values of α; for $\alpha \leq 0.5$ the two will never differ by more than 1 bit (from the bound in Table II).

IV. PRIVACY-PRESERVING EXPERIMENTAL SETUP

By using statistical guessing metrics to evaluate passwords, we are freed from the need to access passwords in their original form. Users may be willing to provide passwords to researchers with ethics oversight [4], [23] but this approach does not scale and the validity of the collected passwords is questionable. In contrast, leaked data sets provide unquestionably valid data but there are ethical questions with using stolen password data and its availability shouldn't be relied on [38]. There is also no control over

the size or composition of leaked data sets. Thus far, for example, no leaked sources have included demographic data.

We addressed both problems with a novel experimental setup and explicit cooperation from Yahoo!, which maintains a single password system to authenticate users for its diverse suite of online services. Our experimental data collection was performed by a proxy server situated in front of live login servers. This is required as long-term password storage should include account-specific salting and iterated hashing which prevent constructing a histogram of common choices, just as they mitigate pre-computed dictionary attacks [39].

Our proxy server sees a stream of pairs $(u, \text{password}_u)$ for each user u logging in to any Yahoo! service. Our goal is to approximate distinct password distributions $\mathcal{X}_{f_i}$ for a series of demographic predicates f_i. Each predicate, such as "does this user have a webmail account?", will typically require a database query based on u. A simplistic solution would be for the proxy to emit a stream of tuples $(\mathbf{H}(\text{password}_u), f_1(u), f_2(u), \dots)$, removing user identifiers u to prevent trivial access to real accounts and using a cryptographic hash function $\mathbf{H}$ to mask the values of individual passwords.[8] There are two major problems to address:

A. Preventing password cracking

If a user u can be *re-identified* by the uniqueness of his or her demographic predicates [40], then the value $\mathbf{H}(\text{password}_u)$ could be used as an oracle to perform an offline dictionary attack. Such a re-identification attack was demonstrated on a data set of movie reviews superficially anonymized for research purposes [41] and would almost certainly be possible for most users given the number and detail of predicates we would like to study.

This risk can be effectively mitigated by prepending the same cryptographically random nonce r to each password prior to hashing. The proxy server must generate r at the beginning of the study and destroy it prior to making data available to researchers. By choosing r sufficiently long to prevent brute-force (128 bits is a conservative choice) and ensuring it is destroyed, $\mathbf{H}(r||\text{password}_u)$ is useless for an attacker attempting to recover password_u but the distribution of hash values will remain exactly isomorphic to the underlying distribution of passwords seen.

B. Preventing cross-account compromise

While including a nonce prevents offline search, an attacker performing large-scale re-identification can still identify sets of users which have a password in common. This decreases security for all users in a group which share a password, as an attacker may then gain access to all accounts in the group by recovering just one user's password by auxiliary means such as phishing, malware, or compromise of an external website for which the password was re-used. Solving this problem requires preventing re-identification by not emitting vectors of predicates for each user.

Instead, the proxy server maintains a histogram $\mathcal{H}_i$ of observed hash values for each predicate f_i. For each pair $(u, \text{password}_u)$ observed, the proxy server adds $\mathbf{H}(r||\text{password}_u)$ to each histogram $\mathcal{H}_i$ for which $f_i(u)$ is true. An additional list is stored of all previously seen hashed usernames $\mathbf{H}(r||u)$ to prevent double-counting users.

C. Deployment details

The collection code, consisting of a few dozens lines of Perl, was audited and r generated using a seed provided by a Yahoo! manager and machine-generated entropy. The experiment was approved by Yahoo!'s legal team as well as the responsible ethics committee at the University of Cambridge. We deployed our experiment on a random subset of Yahoo! servers for a 48 hour period from May 23–25, 2011, observing 69,301,337 unique users and constructing separate histograms for 328 different predicate functions. Of these, many did not achieve a sufficient sample size to be useful and were discarded.

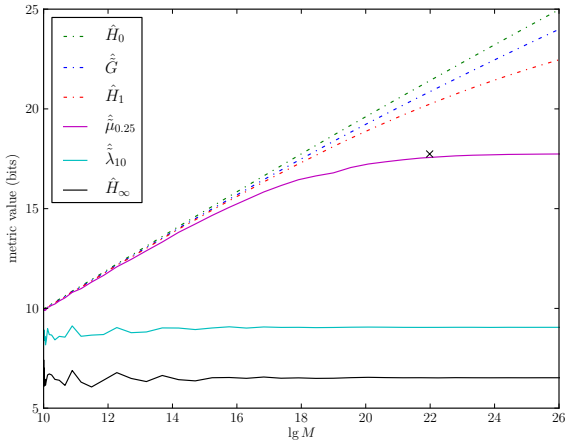


Figure 3. Changing estimates of guessing metrics with increasing sample size M. Estimates for H_∞ and $\tilde{\lambda}_{10}$ converge very quickly; estimates for $\tilde{\mu}_{0.25}$ converge around $M = 2^{22}$ (marked $\times$) as predicted in Section V-A. Estimates for H_0, H_1, and $\tilde{G}$ are not close to converging.

V. Effects of sample size

In our mathematical treatment of guessing difficulty, we assumed complete information is available about the underlying probability distribution of passwords $\mathcal{X}$. In practice, we will need to approximate $\mathcal{X}$ with empirical data.[9] We assume that we have M independent samples $X_1, \dots, X_M \xleftarrow{\text{R}} \mathcal{X}$ and we wish to calculate properties of $\mathcal{X}$.

The simplest approach is to compute metrics using the distribution of samples directly, which we denote $\hat{\mathcal{X}}$.[10] As

[8] Note that $\mathbf{H}$ cannot incorporate any user-specific salt—doing so would occlude the frequency of repeated passwords.

[9] It possible that an attacker knows the precise distribution of passwords in a given database, but typically in this case she or he would also know per-user passwords and would not be guessing statistically.

[10] We use the hat symbol $\hat{}$ for any metric estimated from sampled data.

shown in Figure 3, this approach produces substantial and systematic under-estimates of most metrics, most prominently $\hat{H}_0 = \lg \hat{N}$ which increases nearly continuously with increasing sample size M indicating that new passwords are still being seen often even at our massive sample size. The maximum-likelihood estimation of the growth rate $\frac{d\hat{N}}{dM}$ has been shown to be exactly $\frac{V(1,M)}{M}$, the proportion of passwords in the sample observed only once [42].[11] This can be seen because in in exactly $\frac{V(1,M)}{M}$ of all possible orderings that the sample may have been collected will the last observation have been a new item. For our full sample, $\frac{V(1,M)}{M} = 42.5\%$, indicating that a larger sample would continue to find many new passwords and hence larger estimates for H_0, H_1, G_1 etc. Similarly, for a random subsample of our data, many passwords will be missed and estimates of these metrics will decrease.

Interpreting hapax legomena is a fundamental problem in statistics and there are there are no known non-parametric techniques for estimating the true distribution size N [42]. This is a not merely a theoretical restriction; in the case of passwords determining that apparently pseudorandom passwords really are 128-bit random strings would require an utterly intractable sample size many times greater 2^{128}. Good-Turing techniques [43] aren't helpful for the distribution-wide statistics we are interested in; they can only estimate the cumulative probability of all unobserved events (the "missing mass") and provide damped maximum-likelihood estimates of the probability of individual events.

Fortunately, in practice we can usefully approximate our guessing metrics from reasonably-sized samples; though these estimations implicitly rely on assumptions about the underlying nature of the password distribution. As seen in Figure 3, partial guessing metrics which rely only on the more-frequent items in the distribution are the easiest to approximate, while those which rely on a summation over the entire distribution such as H_0, H_1 and $\tilde{\mu}_\alpha, \tilde{G}_\alpha$ for large values of α will be the most difficult.

A. The region of stability

We can reliably estimate p_i for events with observed frequency $f_i \gg 1$ due to the law of large numbers. Estimating H_∞ requires estimating only p_1, the probability of the most common password, which was 1.08% in our data set. Gaussian statistics can be used to estimate the standard error of the maximum-likelihood estimate $\hat{p}_i$:

$$\text{error}(\hat{p}_i) = \sqrt{\frac{p_i(1-p_i)}{M}} \cdot \frac{1}{p_i} \approx \sqrt{\frac{f_i}{M^2}} \cdot \frac{M}{f_i} = \frac{1}{\sqrt{f_i}}$$

For our data set, this gives a standard error of under 0.1 bit in $\hat{H}_\infty$ for $M \geq 2^{14}$. This argument extends to $\hat{\tilde{\lambda}}_\beta$ for small values of β and in practice we can measure resistance to online guessing with relatively modest sample size.

[11] Events observed only once in a sample are called *hapax legomena* in linguistics, Greek for "said only once."

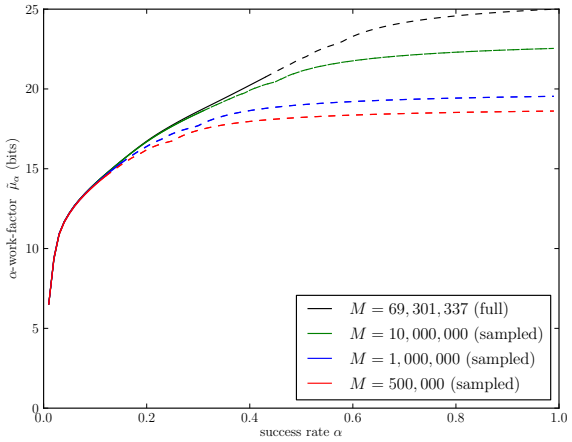


Figure 4. Estimated guessing curves with reduced sample size M. Subsamples were computed randomly without replacement, to simulate having stopped the collection experiment earlier. After the maximum confidence point α_6; there are two (almost indistinguishable) dashed plots representing the 1st and 99th percentiles from 1,000 random samples.

Reasoning about the error in $\hat{\tilde{\mu}}_\alpha$ and $\hat{\tilde{G}}_\alpha$ for values of α which represent realistic brute-force attacks is more difficult. Fortunately, we observe that for our password data set the *number* of events $V(f, M)$ which occur f times in a sample of size M is very consistent for small f and provides a reasonable estimate of the number of events with probability $\frac{f-0.5}{M} \leq p \leq \frac{f+0.5}{M}$ in our full data set.[12]

This enables a useful heuristic that $\tilde{\mu}_\alpha$ and $\tilde{G}_\alpha$ will be well approximated when α is small enough to only rely on events occurring greater than some small frequency f. Calling α_f the cumulative estimated probability of all events occurring at least f times, we took 1,000 random samples of our corpus with $M = 2^{19}$ and observed the following values in the 1st and 99th percentiles:

f	6	7	8
α_f	0.162–0.163	0.153–0.154	0.145–0.146
$\tilde{\mu}_{\alpha_f} - \hat{\tilde{\mu}}_{\alpha_f}$	0.157–0.180	0.125–0.148	0.103–0.127
$\tilde{G}_{\alpha_f} - \hat{\tilde{G}}_{\alpha_f}$	0.155–0.176	0.123–0.146	0.101–0.126

We observed very similar values for larger values of M. Thus, we will use $\hat{\tilde{\mu}}_\alpha, \hat{\tilde{G}}_\alpha$ directly for $\alpha \leq \alpha_6$ for random subsamples of our data. The utility of this heuristic is seen in Figure 3, where it accurately predicts the point at which $\tilde{\mu}_{0.25}$ stabilizes, and in Figure 4, where it marks the point below which $\tilde{\mu}_\alpha$ is inaccurate for varying M.

[12] $V(f, M)$ will almost always overestimate this value because more low-probability events will be randomly over-represented than the converse.

B. Parametric extension of our approximations

Estimating $\tilde{\mu}_\alpha$ and $\tilde{G}_\alpha$ for higher α requires directly assuming a model for the underlying password distribution. Passwords have been conjectured to follow a *power-law distribution*[13] where:

$$\Pr\left[p(x) > y\right] \propto y^{1-a} \tag{12}$$

Unfortunately, using a power-law distribution is problematic for two reasons. First, estimates for the scale parameter a are known to decrease significantly with sample size [42]. Using maximum-likelihood fitting techniques [44] for our observed count data we get the following estimates:

M	69M	10M	1M	100k
$\hat{a}$	2.99	3.23	3.70	4.21

A second problem is this model fits our observed, integer counts. To correctly estimate $\tilde{\mu}_\alpha$ from samples, we need to model the presence of passwords for which $p_i \cdot M < 1$. Power law distributions require assuming a non-zero minimum password probability a-priori [44], which we have no meaningful way of doing.

Instead we need a model $\psi(p)$ for the distribution of password probabilities, an approach taken by linguists for modeling word frequencies [45]. We model the probability of observing a password k times using a mixture-model: first we draw a password probability p randomly according to the probability density function $\psi(p)$, then we draw from a Poisson distribution with expectation $p \cdot M$ to model the number of times we observe this password:

$$\Pr[k \text{ obs.}] = \frac{\int_0^1 \frac{(p\cdot M)^k \cdot e^{-p\cdot M}}{k!}\psi(p)\mathrm{d}p}{1 - \int_0^1 e^{-p\cdot M}\psi(p)\mathrm{d}p} \tag{13}$$

The numerator integrates the possibility of seeing a password with probability p exactly k times, weighted by the probability $\psi(p)$ of a password having probability p. The denominator corrects for the probability of not observing a password at all. This formulation allows us to take a set of counts from a sample $\{f_1, f_2, \ldots\}$ and find the parameters for $\psi(p)$ which maximize the likelihood of our observations:

$$\text{Likelihood} = \prod_{i=1}^{\hat{N}} \Pr\left[f_i \text{ obs.}\right] \tag{14}$$

This model has been effectively applied to word frequencies using the *generalized inverse-Gaussian distribution*:[14]

$$\psi(p|b,c,g) = \frac{2^{g-1}p^{g-1}e^{\frac{p}{c}-\frac{b^2 c}{4p}}}{(bc)^g \cdot K_g(b)} \tag{15}$$

where K_g is the modified Bessel function of the second kind.

[13]Power-law distributions are also called Pareto or Zipfian distributions, which can all be shown to be equivalent formulations [42].

[14]The combined generalized inverse-Gaussian-Poisson model which we adopt is also called the Sichel distribution after its initial use by Sichel in 1975 to model word frequencies [46].

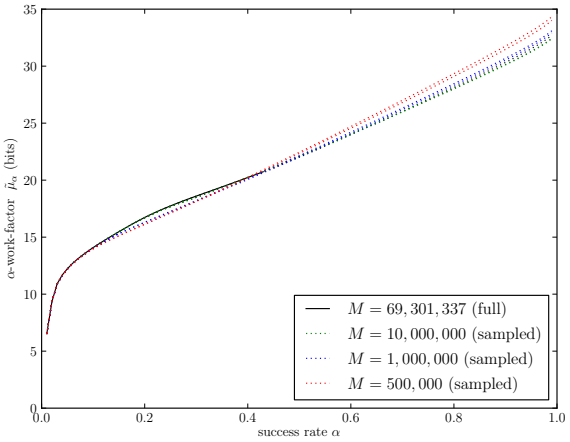


Figure 5. Extrapolated estimates for $\tilde{\mu}_\alpha$ using the generalized inverse Gaussian-Poisson distribution. Compared to naive estimates (Figure 4) the effect of sample size are mitigated. Each plot shows the 99% confidence interval from 1,000 random subsamples. Error from lack of fit of the model dwarfs error due to the randomness of each sample.

The generalized inverse-Gaussian is useful because it blends both power-law $\left(p^{g-1}\right)$ and exponential $\left(e^{\frac{p}{c}-\frac{b^2 c}{4p}}\right)$ behavior and produces a well-formed probability distribution. By plugging Equation 15 into Equation 13 for ψ and solving the integral, we obtain:

$$\Pr[k|b,c,g] = \frac{(\frac{1}{2}\cdot\frac{bcn}{\sqrt{1+cn}})^r \cdot K_{r+g}(b\sqrt{1+cn})}{r!\left((1+cn)^{\frac{g}{2}}K_g(b) - K_g(b\sqrt{1+cn})\right)} \tag{16}$$

Though unwieldy, we can compute Equation 14 using Equation 15 for different parameters of b, c, g. Fortunately, for $b > 0, c > 0, g < 0$ there is only one maximum of this function [45], which enables approximation of the maximum-likelihood fit efficiently by gradient descent.

We can use this model to produce an extrapolated distribution, removing all observed passwords with $f_i < 6$ to leave the well-approximated region of the distribution unchanged and adding synthetic passwords according to our estimated model $\psi(p)$. This is achieved by dividing the region $\left(0, \frac{6}{M}\right)$ into discrete bins, with increasingly small bins near the value p^+ which maximizes $\psi(p^+)$. Into each bin (p_j, p_{j+1}) we insert $\hat{N} \cdot \int_{p_j}^{p_{j+1}} \psi(p)\mathrm{d}p$ events of observed frequency $\frac{p_j + p_{j+1}}{2\cdot M}$. We then normalize the probability of all synthetic events by multiplying the correction factor $\frac{1}{\alpha_{f\geq 6}} \cdot \int_{\frac{6}{M}}^1 \psi(p)\mathrm{d}p$ to leave the head of the distribution intact.

Figure 5 plots the 1st and 99th percentile of $\tilde{\mu}_\alpha$ for extrapolations of random subsamples of our data. We use $\tilde{\mu}_\alpha$ because it is strictly less-well approximated than $\tilde{G}_\alpha$, which is weighted slightly more towards well-approximated events in the distribution. Some key values are:

	69M	10M	1M	500k
$\hat{\tilde{\mu}}_{0.25}$	17.74	17.67–17.67	17.24–17.25	17.07–17.17
$\hat{\tilde{\mu}}_{0.5}$	22.01	22.09–22.11	22.06–22.11	22.28–22.48
$\hat{\tilde{\mu}}_{0.75}$	27.07	26.98-27.01	27.25–27.35	27.02–27.89

Our estimates are biased towards under-correction for lower values of α and over-biased for higher values. Still, even with a 500k sample the estimates for $\tilde{\mu}_\alpha$ agree to within 1 bit for all values of α.

C. Limitations and estimating confidence

We can not conclude that the underlying probability distribution of passwords is completely modeled. Indeed, using a Kolmogorov-Smirnov test we can reject with very high confidence ($p > 0.99$) the hypothesis that our sample was drawn from the modeled distribution.

Our goal is to accurately compare statistics for differently-sized subsamples of our data. Doing so using our empirical precision estimates directly is accurate only under the assumption that two different subpopulations have each chosen a distribution of passwords which our model fits equally well.[15] If some definable population of user-generated passwords form a very different underlying distribution (for example, uniform or exponential), then our model might produce much more variable estimates. When analyzing our data in Section VI we thus make a weaker claim only that different demographic subsamples of users are significantly different from the global population of users if our extrapolation produces estimates which are outside the 1st or 99th percentile of estimates observed for similarly-sized random samples as listed in this section.

VI. Analysis of Yahoo! data

A. External comparison

We first compare our collected data to several known data sets. To the author's knowledge, there have been two large-scale leaks of password data suitable for statistical analysis:[16] the 2009 RockYou leak and a 2011 leak of roughly 500k passwords from the gaming website Battlefield Heroes.[17] Guessing metrics for these distributions and our collected data are listed in Table III. All three distributions, despite being taken from substantially different populations, agree to within 1 bit for estimates of online attacks ($\hat{H}_\infty$ and $\hat{\tilde{\lambda}}_{10}$), and within 2 bits for offline attacks ($\hat{\tilde{G}}_{0.25}$ and $\hat{\tilde{G}}_{0.5}$).

We plot the guessing curve for our collected data in Figure 6 along with that of the RockYou distribution. We also include guessing curves for two distributions from non-password-based authentication schemes: a distribution of four-digit unlock codes used for an iPhone application leaked in 2011 [33] and the distribution of surnames (the most common category of answer to personal knowledge questions) from a large-scale crawl of Facebook [34]. Within our plot we add estimated data points from cracking experiments. We include both the password-cracking experiments discussed in Section II-A and cracking attempts on two graphical schemes: a 2004 study of user choice in a face-based graphical PIN scheme [47] and a 2005 study of user-selected image points in the PassPoints scheme [48]. Note that due to our use of published cracking results, we are restricted to using $\tilde{\mu}_\alpha$ instead of $\tilde{G}_\alpha$.

	M	$\hat{H}_\infty$	$\hat{\tilde{\lambda}}_{10}$	$\hat{\tilde{G}}_{0.25}$	$\hat{\tilde{G}}_{0.5}$
Yahoo! (2011)	69301337	6.5	9.1	17.6	21.6
RockYou (2009)	32603388	6.8	8.9	15.9	19.8
Battlefield Heroes (2011)	548774	7.7	9.8	16.5	20.0

Table III
Comparison of Yahoo! data with leaked data sets

The guessing curve shows how close the distribution of passwords at both Yahoo! and RockYou are compared to other authentication schemes. Both password distributions have a much sharper increase for very low success-rate attackers than the surname or PIN distributions do, meaning passwords are particularly vulnerable to a trawling attacker who only makes a few attempts at a large number of accounts. However, passwords have comparatively high α-work-factor against brute-force attackers. The 1990 cracking study by Klein provided estimates very close to the optimal attack for our observed data, suggesting that passwords have changed only marginally since then.

B. Comparing subpopulations

Of the 328 subpopulations for which we compiled separate distributions, we summarize the most interesting which gathered a sufficient number of samples in Table IV. All of our sub-distributions had similar guessing metrics: the range of H_∞ was 5.0–9.1 bits and for $\tilde{\lambda}_{10}$ from 7.5–10.9 bits, just over one decimal order of magnitude in variation. Variation in $\tilde{G}_{0.5}$ was substantially larger, with the weakest population having an estimated 17.0 bits and the strongest 26.6 (nearly three decimal orders of magnitude).

Thus, while there is no "good" population of users which isn't generally vulnerable to guessing attacks, there is still variation which is strongly detectable within the limits of our sampling confidence: our estimates of H_∞ and $\tilde{\lambda}_{10}$ are all accurate to within at least 0.1 bit based on our calculations in Section V-A, while our extrapolation of $\tilde{G}_\alpha$ allows us to identify many groups which are statistically different from the overall population as discussed in Section V-C.

Demographically, users' reported gender had a small but

[15] Supporting this assumption, we find that our model produces similarly accurate estimates for subsamples of the RockYou distribution, the only other large password data set to which we have access.

[16] A prominent 2010 leak revealed nearly 1M passwords from the blogging site Gawker, but these were salted via the Unix `crypt()` function, preventing full analysis of the distribution.

[17] The Battlefield Heroes passwords were hashed with MD5, but without any salt, making analysis of the distribution possible.

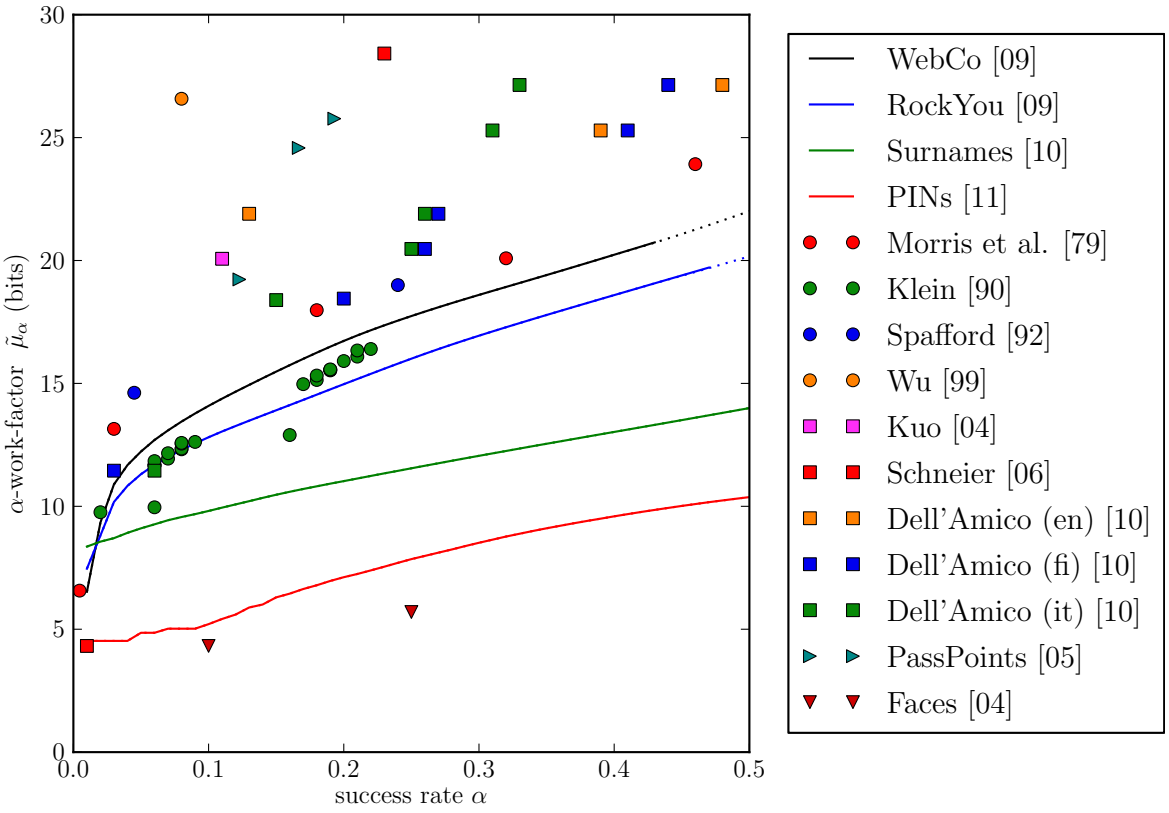


Figure 6. Guessing curve for Yahoo! passwords compared with previously published data sets and cracking evaluations.

split effect, with male-chosen passwords being slightly more vulnerable to online attack and slightly stronger against offline attack. There is a general trend towards better password selection with users' age, particularly against online attacks, where password strength increases smoothly across different age groups by about a bit between the youngest users and the oldest users. Far more substantial were the effects of language: passwords chosen by Indonesian-speaking users were amongst the weakest subpopulations identified with $H_\infty = 5.5$. In contrast, German and Korean-speaking users provided relatively strong passwords.

Users' account history also illustrates several interesting trends. There is a clear trend towards stronger passwords amongst users who actively change their password, with users who have changed passwords 5 or more times being one of the strongest groups.[18] There is a weaker trend towards stronger passwords amongst users who have completed an email-based password recovery. However, users who have had their password reset manually after reporting their account compromised do not choose better passwords than average users.[19] Users who log in infrequently, judging by the time of previous login before observation in our experiment, choose slightly better passwords. A much stronger trend is that users who have recently logged in from multiple locations choose relatively strong passwords.[20]

There is a weak trend towards improvement over time, with more recent accounts having slightly stronger passwords. Of particular interest to the security usability research community, however, a change in the default login form at Yahoo! appears to have had little effect. While Yahoo! has employed many slightly different login forms across its different services, we can compare users who initially enrolled using each of two standard forms: one of which has no minimum length requirement and no guidance on password selection, and the other with a 6 character minimum and a graphical indicator of password strength. This change made almost no difference in security against online guessing, and increased the offline metrics by only 1 bit.

Finally, we can observe variation between users who have

[18] As these password changes were voluntary, this trend doesn't relate mandatory password change policies, particularly as many users choose predictably related passwords when forced [49].

[19] A tempting interpretation is that user choice in passwords does not play a significant role in the risk of account compromise, though this is not clearly supported since we can only observe the post-compromise strength.

[20] Yahoo! maintains a list of recent login locations for each user for abuse detection purposes.

	M	$\hat{H}_\infty$	$\hat{\tilde{\lambda}}_{10}$	$\hat{\tilde{G}}_{0.25}$	$\hat{\tilde{G}}_{0.5}$
all passwords	**69301337**	**6.5**	**9.1**	**17.6**	**21.6**
gender (self-reported)					
female	30545765	6.9	9.3	*17.2*	21.1
male	38624554	6.3	8.8	17.7	*21.8*
age (self-reported)					
13–24	18199547	6.3	8.7	16.7	20.9
25–34	22380694	6.2	8.8	*17.1*	21.2
35–44	12983954	6.8	9.4	*17.4*	21.3
45–54	8075887	7.3	9.8	*17.3*	21.3
$\geq$ 55	7110689	7.5	9.8	*17.3*	21.4
language preference					
Chinese	1564364	*6.5*	8.6	*17.3*	*22.0*
German	1127474	7.4	9.7	15.8	19.7
English	55805764	6.5	*9.0*	*17.4*	21.5
French	2084219	6.9	*9.0*	14.8	18.6
Indonesian	1061540	5.5	7.9	14.3	17.0
Italian	811133	6.8	*9.0*	14.5	18.0
Korean	530759	7.5	9.5	18.1	22.7
Portuguese	2060256	6.5	*9.0*	15.6	18.8
Spanish	3065901	6.6	*9.1*	15.6	19.7
tenure of account					
$\leq$ 1 y	5182527	6.9	*9.1*	18.0	22.5
1–2 years	5182527	6.9	*9.1*	18.0	22.5
2–3 years	12261556	6.2	8.6	17.7	*21.8*
3–4 years	10332348	6.2	8.8	*17.5*	21.6
4–5 years	9290840	6.1	8.8	*17.2*	21.2
$\geq$ 5 years	29104856	6.8	9.3	*17.2*	21.2
password requirements at registration					
none	20434875	6.6	9.2	16.8	20.7
6 char. minimum	13332334	6.5	*9.0*	*17.6*	*21.6*
last recorded login					
$<$ 30 days	32627777	*6.5*	*9.0*	*17.5*	21.5
$<$ 90 days	55777259	6.5	*9.0*	*17.5*	21.5
$>$ 90 days	8212643	7.0	9.5	17.7	*21.9*
number of login locations					
1	16447906	6.0	8.6	*17.1*	21.1
$\geq$2	52853431	6.7	*9.2*	17.7	*21.7*
$\geq$ 10	17146723	7.3	9.7	18.3	22.6
number of password changes					
none	52117133	6.2	8.8	*17.1*	20.9
1	9608164	8.3	10.4	18.8	23.2
$>$1	7576040	8.6	10.7	19.5	24.2
$\geq$5	930035	9.1	10.9	19.7	25.9
number of password resets (forgotten password)					
none	61805038	6.4	8.9	*17.3*	21.3
1	4378667	8.2	10.5	19.2	23.8
$>$1	3117632	8.7	10.8	19.7	24.6
$\geq$5	387469	8.7	10.6	19.9	26.6
amount of data stored with Yahoo!					
1st quartile	9830792	5.6	8.2	*17.3*	21.5
2nd quartile	20702119	6.3	8.8	*17.5*	21.5
3rd quartile	21307618	6.8	9.3	*17.5*	21.4
4th quartile	17447029	7.6	10.0	17.8	*22.0*
usage of different Yahoo! features					
media sharing	5976663	7.7	10.1	18.0	*22.3*
retail	2139160	8.8	10.5	16.8	21.4
webmail	15965774	6.3	8.8	*17.4*	21.2
chat	37337890	6.2	8.7	*17.1*	21.2
social networking	14204900	7.1	9.6	17.7	*21.8*
mobile access	20676566	6.7	9.3	*17.1*	21.1
Android client	1359713	8.3	10.3	*17.3*	21.5
iPhone client	6222547	8.1	10.1	*17.6*	*21.6*
RIM client	3843404	7.6	10.0	*17.2*	21.1

Note: Estimates in italics are not significantly different from the aggregate population of users, as discussed in Section V-C.

Table IV
GUESSING STATISTICS FOR VARIOUS GROUPS OF YAHOO! USERS.

actively used different Yahoo! services. Users who have used Yahoo!'s online retail platform (which means they have stored a payment card) do choose very weak passwords with lower frequency, with $\tilde{\lambda}_{10}$ increasing by about 2 bits. However, the distribution is indistinguishable from average users against offline attack. A similar phenomenon occurs for users of some other features, such as media sharing or dedicated smartphone clients for Android, Blackberry, or iOS, which see slightly better security against online attacks but are indistinguishable otherwise. Other popular features, such as webmail, chat, and social networking, saw slightly fewer weak passwords than normal, but again were indistinguishable against offline attacks.

One other interesting categorization is the amount of data that users have stored with Yahoo!. While this is a very rough proxy for how active user accounts have been, there is a clear trend that users with a large amount of stored data choose better passwords.

C. Effects of dictionary specificity

While we have focused so far only on comparing the shape of distributions, it is also interesting to compare their content to examine the extent to which an inappropriate cracking dictionary might slow down attackers (or skew the conclusions of academic studies). To do this, we can simulate a guessing attack on one distribution by guessing passwords in the order they appear in another distribution, instead of an optimal attack. We tested the top 1,000 passwords in each subpopulation, comparing $\hat{\lambda}_{1000}$ for an attack with the optimal dictionary to a sub-optimal one. A simple example is to compare male and female-chosen passwords:

		dictionary	
		♀	♂
target	♀	**7.8%**	6.8%
	♂	6.3%	**7.1%**

There is a 10–15% loss in efficiency if an attacker uses the optimal male dictionary against female-chosen passwords, or vice-versa. This is small enough that we may conclude real-world attackers are unlikely to tailor their guessing approach based on the gender distribution of their target users. In general, using an inappropriate dictionary has surprisingly little impact on guessing efficiency, at least for an attacker with a desired success rate $\alpha < 10\%$, which we tested to stay in the well-approximated region given our data. In Table V we compare the efficiency loss when using a password dictionary from users of different languages, perhaps the most inappropriate dictionaries possible. Surprisingly, the worst efficiency loss observed is only a factor of 4.8, when using an optimal Vietnamese-language password dictionary against French speakers' passwords.

We also observe in Table V that simply using the global list of most popular passwords performs very well against

		dictionary												
		Chinese	German	Greek	English	French	Indonesian	Italian	Korean	Portuguese	Spanish	Vietnamese	global	minimax
target	Chinese	**4.4%**	1.9%	2.7%	2.4%	1.7%	2.0%	2.0%	2.9%	1.8%	1.7%	2.0%	2.9%	2.7%
	German	2.0%	**6.5%**	2.1%	3.3%	2.9%	2.2%	2.8%	1.6%	2.1%	2.6%	1.6%	3.5%	3.4%
	Greek	9.3%	7.7%	**13.4%**	8.4%	7.4%	8.1%	8.0%	8.0%	7.7%	7.8%	7.7%	8.6%	8.9%
	English	4.4%	4.6%	3.9%	**8.0%**	4.3%	4.5%	4.3%	3.4%	3.5%	4.2%	3.5%	7.9%	7.7%
	French	2.7%	4.0%	2.9%	4.2%	**10.0%**	2.9%	3.2%	2.2%	3.1%	3.4%	2.1%	5.0%	4.9%
	Indonesian	6.7%	6.3%	6.5%	8.7%	6.3%	**14.9%**	6.2%	5.8%	6.0%	6.2%	5.9%	9.3%	9.6%
	Italian	4.0%	6.0%	4.6%	6.3%	5.3%	4.6%	**14.6%**	3.3%	5.7%	6.8%	3.2%	7.2%	7.1%
	Korean	3.7%	2.0%	3.0%	2.6%	1.8%	2.3%	2.0%	**5.8%**	2.4%	1.9%	2.2%	2.8%	3.0%
	Portuguese	3.9%	3.9%	4.0%	4.3%	3.8%	3.9%	4.4%	3.5%	**11.1%**	5.8%	2.9%	5.1%	5.3%
	Spanish	3.6%	5.0%	4.0%	5.6%	4.6%	4.1%	6.1%	3.1%	6.3%	**12.1%**	2.9%	6.9%	7.0%
	Vietnamese	7.0%	5.7%	6.2%	7.7%	5.8%	6.3%	5.7%	6.0%	5.8%	5.5%	**14.3%**	7.8%	8.3%

Table V
LANGUAGE DEPENDENCY OF PASSWORD GUESSING. EACH CELL INDICATES THE SUCCESS RATE OF A GUESSING ATTACK WITH 1000 ATTEMPTS USING A DICTIONARY OPTIMAL FOR USERS REGISTERED AT YAHOO! WITH DIFFERENT PREFERRED LANGUAGES.

most subsets. The greatest efficiency loss for any subset when using the global list is only 2.2, for Portuguese language passwords. We can improve this slightly further by constructing a special dictionary to be effective against all subsets. We do this by repeatedly choosing the password for which the lowest popularity in any subset is maximal and call it the "minimax" dictionary, also seen in Table V. This dictionary performs very similarly to the global dictionary, reducing the maximum efficiency loss to a factor 2.1, also for Portuguese language passwords.

Digging into our data we find "global passwords" which are popular across all subgroups we observed. The single most popular password we observed, for example, occurred with probability at least 0.14% in every subpopulation. Some overall popular passwords were very rare in certain subpopulations. For example, the third most common password, with overall probability 0.1%, occurred nearly 100 times less frequently in some subpopulations. However, there were eight passwords which occurred with probability at least 0.01% in every subpopulation. Without access to the raw passwords, we can only speculate that these are numeric passwords as these are popular[21] and internationalize well.

Despite the existence of globally popular passwords, however, we still conclude that dictionary specificity can have surprisingly large results. For example, the following table shows efficiency losses of up to 25% from dictionaries tailored to people from different English-speaking countries:

		dictionary				
		us	uk	ca	au	global
target	us	**8.2%**	6.6%	7.4%	7.2%	8.1%
	uk	5.4%	**6.9%**	5.5%	5.6%	5.5%
	ca	8.8%	7.9%	**9.9%**	8.7%	8.8%
	au	7.4%	7.2%	7.6%	**8.8%**	7.5%

[21]Within the RockYou data set, `123456` was the most popular password and 5 other number-only passwords were amongst the top ten.

We observe comparable efficiency losses based on age:

		dictionary				
		13–20	21–34	35–54	55+	global
target	13–20	**8.4%**	7.8%	7.1%	6.5%	7.9%
	21–34	7.3%	**7.9%**	7.3%	6.7%	7.8%
	35–54	5.4%	5.8%	**6.4%**	6.1%	6.2%
	55+	5.4%	5.8%	6.8%	**7.3%**	6.5%

We even observe efficiency losses based on service usage:

		dictionary				
		retail	chat	media	mail	global
target	retail	**7.0%**	5.6%	6.6%	5.6%	6.0%
	chat	6.9%	**8.4%**	7.8%	8.3%	8.3%
	media	5.7%	5.6%	**6.0%**	5.6%	5.8%
	mail	6.7%	8.0%	7.5%	**8.2%**	8.1%

VII. CONCLUDING REMARKS

By establishing sound metrics and rigorously analyzing the largest password corpus to date, we hope to have contributed both tools and numbers of lasting significance.

As a rule of thumb for security engineers, passwords provide roughly equivalent security to 10-bit random strings against an optimal online attacker trying a few popular guesses for large list of accounts. In other words, an attacker who can manage 10 guesses per account, typically within the realm of rate-limiting mechanisms, will compromise around 1% of accounts, just as they would against random 10-bit strings. Against an optimal attacker performing unrestricted brute force and wanting to break half of all available accounts, passwords appear to be roughly equivalent to 20-bit random strings. This means that no practical amount of iterated hashing can prevent an adversary from breaking a large number of accounts given the opportunity for offline

search. An important caveat is that these passwords were chosen with very few restrictions—a stricter password selection policy might produce distributions with significantly higher resistance to guessing.

Still, these numbers represents a minimal benchmark which any serious password replacement scheme should aim to decisively clear. The enormous gap between a real password distribution and the theoretical space of passwords shows why research proposals involving human-chosen secrets should estimate security using metrics like $\tilde{\mu}_\alpha$ and $\tilde{G}_\alpha$ to model partial guessing attacks. Where possible, comparison to past empirical estimates of guessing attacks should be provided, as we have done with Figures 1 and 6.

The most troubling finding of our study is how little password distributions seem to vary, with all populations of users we were able to isolate producing similar skewed distributions with effective security varying by no more than a few bits. Factors increasing security motivation like registering a payment card only seem to nudge users away from the weakest passwords, and a limited natural experiment on actively encouraging stronger passwords seems to have made little difference. Passwords have been argued to be "secure enough" for the web with users rationally choosing weak passwords for accounts of little importance [50], but these results may undermine this explanation as user choice does not vary greatly with changing security concerns as would be expected if weak passwords arose primarily due to user apathy. This may indicate an underlying problem with passwords that users aren't willing or able to manage how difficult their passwords are to guess.

Acknowledgments

This research would not have been possible without the gracious cooperation and support of many people at Yahoo!, in particular Henry Watts, my mentor, Elizabeth Zwicky who provided extensive help collecting and analyzing data, as well as Ram Marti, Clarence Chung, and Christopher Harris who helped set up data collection experiments. I would also like to thank my supervisor Ross Anderson, the paper's shepherd Arvind Narayanan, as well as Paul van Oorschot, Richard Clayton, Andrew Lewis, Cormac Herley, Saar Drimer, Markus Kuhn and Bruce Christianson for helpful comments and discussions about password statistics. My research is funded by the Gates Cambridge Trust.

References

[1] M. V. Wilkes, *Time-sharing computer systems*. New York: Elsevier, 1968.

[2] J. H. Saltzer, "Protection and the Control of Information Sharing in Multics," *Commun. ACM*, vol. 17, pp. 388–402, 1974.

[3] R. Morris and K. Thompson, "Password Security: A Case History," *Commun. ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[4] D. Klein, "Foiling the Cracker: A Survey of, and Improvements to, Password Security," in *Proceedings of the 2nd USENIX Security Workshop*, 1990, pp. 5–14.

[5] E. Spafford, "Observations on Reusable Password Choices," in *Proceedings of the 3rd USENIX Security Workshop*, 1992.

[6] B. Schneier, "Real-World Passwords," December 2006. [Online]. Available: www.schneier.com/blog/archives/2006/12/realworld_passw.html

[7] M. Dell'Amico, P. Michiardi, and Y. Roudier, "Password Strength: An Empirical Analysis," in *INFOCOM'10: Proceedings of the 29th Conference on Information Communications*. IEEE, 2010, pp. 983–991.

[8] M. Weir, S. Aggarwal, M. Collins, and H. 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*. ACM, 2010, pp. 162–175.

[9] D. Seeley, "Password Cracking: A Game of Wits," *Commun. ACM*, vol. 32, pp. 700–703, 1989.

[10] "John the Ripper," http://www.openwall.com/john/.

[11] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password Cracking Using Probabilistic Context-Free Grammars," in *SP '09: Proceedings of the 30th IEEE Symposium on Security and Privacy*. IEEE, 2009, pp. 391–405.

[12] T. Wu, "A Real-World Analysis of Kerberos Password Security," in *NDSS '99: Proceedings of the 1999 Network and Distributed System Security Symposium*, 1999.

[13] C. Kuo, S. Romanosky, and L. F. Cranor, "Human Selection of Mnemonic Phrase-based Passwords," in *SOUPS '06: Proceedings of the 2nd Symposium on Usable Privacy and Security*. ACM, 2006, pp. 67–78.

[14] J. A. Cazier and B. D. Medlin, "Password Security: An Empirical Investigation into E-Commerce Passwords and Their Crack Times." *Information Systems Security*, vol. 15, no. 6, pp. 45–55, 2006.

[15] B. L. Riddle, M. S. Miron, and J. A. Semo, "Passwords in use in a university timesharing environment," *Computers and Security*, vol. 8, no. 7, pp. 569–578, 1989.

[16] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *CCS '05: Proceedings of the 12th ACM Conference on Computer and Communications Security*. ACM, 2005, pp. 364–372.

[17] C. Castelluccia, M. Dürmuth, and D. Perito, "Adaptive Password-Strength Meters from Markov Models," *NDSS '12: Proceedings of the Network and Distributed System Security Symposium*, 2012.

[18] M. Zviran and W. J. Haga, "Password security: an empirical study," *Journal of Management Information Systems*, vol. 15, no. 4, pp. 161–185, 1999.

[19] M. M. Devillers, "Analyzing Password Strength," Radboud University Nijmegen, Tech. Rep., 2010.

[20] W. E. Burr, D. F. Dodson, and W. T. Polk, "Electronic Authentication Guideline," *NIST Special Publication 800-63*, 2006.

[21] D. Florêncio and C. Herley, "A large-scale study of web password habits," in *WWW '07: Proceedings of the 16th International Conference on the World Wide Web*. ACM, 2007, pp. 657–666.

[22] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor, "Encountering Stronger Password Requirements: User Attitudes and Behaviors," in *SOUPS '10: Proceedings of the 6th Symposium on Usable Privacy and Security*. ACM, 2010.

[23] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez, "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms," Carnegie Mellon University, Tech. Rep. CMU-CyLab-11-008, 2011.

[24] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password Memorability and Security: Empirical Results," *IEEE Security and Privacy Magazine*, vol. 2, no. 5, pp. 25–34, 2004.

[25] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in *CCS '09: Proceedings of the 16th ACM Conference on Computer and Communications Security*. ACM, 2009, pp. 635–647.

[26] C. E. Shannon, "A Mathematical Theory of Communication," in *Bell System Technical Journal*, vol. 7, 1948, pp. 379–423.

[27] C. Cachin, "Entropy measures and unconditional security in cryptography," Ph.D. dissertation, ETH Zürich, 1997.

[28] J. O. Pliam, "On the Incomparability of Entropy and Marginal Guesswork in Brute-Force Attacks," in *Progress in Cryptology-INDOCRYPT 2000*, 2000.

[29] S. Boztas, "Entropies, Guessing, and Cryptography," Department of Mathematics, Royal Melbourne Institute of Technology, Tech. Rep. 6, 1999.

[30] J. L. Massey, "Guessing and Entropy," in *Proceedings of the 1994 IEEE International Symposium on Information Theory*, 1994, p. 204.

[31] A. Rényi, "On measures of information and entropy," *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, pp. 547–561, 1961.

[32] R. V. Hartley, "Transmission of Information," *Bell System Technical Journal*, vol. 7, no. 3, pp. 535–563, 1928.

[33] J. Bonneau, S. Preibusch, and R. Anderson, "A birthday present every eleven wallets? The security of customer-chosen banking PINs," *FC '12: The 16th International Conference on Financial Cryptography and Data Security*, 2012.

[34] J. Bonneau, M. Just, and G. Matthews, "What's in a name? Evaluating statistical attacks against personal knowledge questions," *FC '10: The 14th International Conference on Financial Cryptography and Data Security*, 2010.

[35] S. Brostoff and A. Sasse, ""Ten strikes and you're out": Increasing the number of login attempts can improve password usability," in *Proceedings of CHI 2003 Workshop on HCI and Security Systems*. John Wiley, 2003.

[36] J. Bonneau and S. Preibusch, "The password thicket: technical and market failures in human authentication on the web," *WEIS '10: Proceedings of the 9th Workshop on the Economics of Information Security*, 2010.

[37] M. Alsaleh, M. Mannan, and P. van Oorschot, "Revisiting Defenses Against Large-Scale Online Password Guessing Attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 128–141, 2012.

[38] S. Egelman, J. Bonneau, S. Chiasson, D. Dittrich, and S. Schechter, "Its Not Stealing If You Need It: On the ethics of performing research using public data of illicit origin (panel discussion)," *WECSR '12: The 3rd Workshop on Ethics in Computer Security Research*, 2012.

[39] B. Kaliski, *RFC 2898: PKCS #5: Password-Based Cryptography Specification Version 2.0*, IETF, 2000.

[40] D. E. Denning and P. J. Denning, "The tracker: a threat to statistical database security," *ACM Transactions on Database Systems*, vol. 4, pp. 76–96, 1979.

[41] A. Narayanan and V. Shmatikov, "How To Break Anonymity of the Netflix Prize Dataset," *eprint arXiv:cs/0610105*, 2006.

[42] H. R. Baayen, *Word Frequency Distributions*, ser. Text, Speech and Language Technology. Springer, 2001.

[43] W. A. Gale, "Good-Turing smoothing without tears," *Journal of Quantitative Linguistics*, vol. 2, 1995.

[44] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-Law Distributions in Empirical Data," *SIAM Rev.*, vol. 51, pp. 661–703, 2009.

[45] M. Font, X. Puig, and J. Ginebra, "A Bayesian analysis of frequency count data," *Journal of Statistical Computation and Simulation*, 2011.

[46] H. Sichel, "On a distribution law for word frequencies," *Journal of the American Statistical Association*, 1975.

[47] D. Davis, F. Monrose, and M. K. Reiter, "On User Choice in Graphical Password Schemes," in *Proceedings of the 13th USENIX Security Symposium*, 2004.

[48] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, "PassPoints: design and longitudinal evaluation of a graphical password system," *International Journal of Human-Computer Studies*, vol. 63, pp. 102–127, 2005.

[49] Y. Zhang, F. Monrose, and M. K. Reiter, "The security of modern password expiration: an algorithmic framework and empirical analysis," in *CCS '10: Proceedings of the 17th ACM Conference on Computer and Communications Security*. ACM, 2010, pp. 176–186.

[50] C. Herley, P. van Oorschot, and A. S. Patrick, "Passwords: If We're So Smart, Why Are We Still Using Them?" *FC '09: The 13th International Conference on Financial Cryptography and Data Security*, 2009.