Why Botnets Work: Distributed Brute-Force Attacks Need No Synchronization

Salman Salamatian, Wasim Huleihel, Ahmad Beirami, Asaf Cohen, Muriel Médard

Abstract

In September 2017, McAfee Labs quarterly report [1] estimated that brute force attacks represent 20% of total network attacks, making them the most prevalent type of attack ex-aequo with browser based vulnerabilities. These attacks have sometimes catastrophic consequences, and understanding their fundamental limits may play an important role in the risk assessment of password-secured systems, and in the design of better security protocols. While some solutions exist to prevent online brute-force attacks that arise from one single IP address, attacks performed by botnets are more challenging. In this paper, we analyze these distributed attacks by using a simplified model. Our aim is to understand the impact of distribution and asynchronization on the overall computational effort necessary to breach a system. Our result is based on Guesswork, a measure of the number of password queries (guesses) before the correct one is found in an optimal attack, which is a direct surrogate for the time and the computational effort. We model the lack of synchronization by a worst-case optimization in which the queries are received in the worst possible order, resulting in a min-max formulation. We show that even without synchronization and for sequences of growing length, the asymptotic optimal performance is achievable by using randomized guesses drawn from an appropriate distribution. Therefore, randomization is key for distributed asynchronous attacks. In other words, asynchronous guessers can asymptotically perform brute-force attacks as efficiently as synchronized guessers.

I. INTRODUCTION

From online banking [2] and bitcoin wallets [3], to secure shell (SSH), file transfer protocol(ftp), and telnet servers [4], and passing by governmental institutions [5], brute-force attacks have shown to be one of the major threats to network security. Despite the computational burden on the attacker, brute-force attacks are prevalent. This can be explained through multiple points of view. First, passwords are often weaker than what they ought to be, meaning that attackers can hope to find the correct password well before they query a significant portion of the possible password strings. Next, attacks through huge networks of compromised computers (botnets) are now more common, giving access to significant computational resources for the attacker. More critically, these botnets help to disguise the attack by distributing it. Indeed, a main solution to the threat of online brute-force attacks is to setup a system that detects and prevents too many queries from any one user, as determined by IP addresses. As such, an attacker which would use only a single IP address, would be limited to a fixed number of guesses. In recent years however, this defense was circumvented by using massive botnets, each bot querying potential passwords. In this situation, it is extremely hard to detect legitimate users in the crowd of illegitimate attackers. However, these attacks come with a cost, namely, the attack is now distributed across thousands, if not millions of computers, each with limited computational power and synchronization tools.

As a first step to understand the impact of synchronization, we put forth a simplified mathematical model for passwords and brute-force attacks. We believe that the intuition gained from this model is informative and helpful in assessing the security of systems against brute-force attacks. In particular, we study Guesswork, a measure of the number of password queries (guesses) that an adversary would have to perform before finding the correct one. Guesswork is best explained through the following simple game: Alice selects a secret discrete random variable $X$ taking values in a finite set $\mathcal{X}$, and distributed according to $P_X$. Then, Bob, who does not see the realization of $X$ but does know $P_X$, presents to Alice a successive sequence of guesses $\hat{X}_1, \hat{X}_2, \ldots$ and so on. For each guess $\hat{X}_i$, Alice checks whether it is the correct symbol $X$. If the answer is affirmative, Alice says “yes”, and the game ends. Otherwise, the game continues, and Alice examines subsequent guesses. This game has a simple interpretation in the context of security. Consider a setup where a system is protected using a password $X$, that Alice draws at random from a distribution $P_X$ (or drawn by nature and revealed to Alice, as it happens in several important password-protection tools which generate passwords, e.g. iCloud keychain). An adversary, Bob, wishes to breach the system by performing a brute-force attack, or, in other words, by guessing the password $X$. The brute-force attack on the system would consist of, first, producing a list of all possible password strings $\mathcal{X}$ ordered from most, to least likely with respect to $P_X$, and then exhausting the list of passwords one by one until successfully guessing the correct password. In order to understand the security of such a system under these attacks, it is necessary to evaluate the computational effort required by Bob to breach the system. To achieve this, it is reasonable to quantify the number of queries before the correct password is found, which we...
shall denote by \( G^*(X) \), and in particular, its \( \rho \)-th moment, i.e. \( \mathbb{E}[G^*(X)^\rho] \). The number of queries is a direct surrogate for the computational effort that Bob must accomplish, and the lower this quantity, the more vulnerable the system is to brute-force attacks.

If multiple adversarial agents coordinate their attack on the secret string, the system will be compromised as soon as any of them succeeds. Moreover, the individual computational effort of each adversary is reduced, while the total number of queries remains the same. Indeed, an optimal strategy here would consist of having each agent query the most-likely password that has not been queried by any of the other agents. Since this strategy reduces to querying as a group from the optimal list, the average number of queries completed by each agent is thus reduced by a factor of the number of agents, with respect to the case where a single agent queries alone. This requires the agents to be able to synchronize their queries, that is, there must be a knowledge of an ordering in which the agents make guesses. However, in many practical scenarios the adversarial agents are completely distributed and have limited communication with each other. One prime example is botnets, in which agents are often oblivious to the actions taken by other agents, and may have limited access to shared memory or synchronization tools. Owing to constraints of the physical computers in which these bots run, the speed, latency, and reliability of these agents is heterogeneous — thus perfect synchronization is unlikely. Note that even if a central agent distributes lists of possible guesses to the bots, such that the lists form a partition of all guesses, making sure no guess is repeated, the lack of synchronization may still render the process sub-optimal. We illustrate an example of synchronized and asynchronous attack in Fig 1. At one extreme, a complete lack of synchronization can be modeled by a worst-case optimization, in which the guesses of each agent come in the worst possible order. The goal of this paper is to study how much the lack of synchronization, as described above, is detrimental to the overall number of queries that are made until the game ends. We discuss why deterministic strategies cannot perform well in this paradigm, while on the other hand, a simple randomized strategy in which all the guesses are drawn i.i.d. from a certain distribution asymptotically achieves the same optimal performance of a synchronous attack when guessing infinite sequences. This optimal guessing distribution is non-trivial, and, surprisingly it is not the original password generating distribution \( P_X \). It is a tilted distribution from \( P_X \), where the tilt exponent depends on the moment of guesswork of interest. In other words, distributed and asynchronous agents can adopt a strategy for which the asymptotic number of total queries sent before a system breach is optimal, regardless of the ordering in which these queries are received, but this distribution is only optimal for a given moment of guesswork, and not optimal universally across all moments.

For the sake of simplicity, we have made the following assumptions on the password generation process, as well as on the brute-force attack itself.

1) Password are assumed to be of a fixed length \( n \). Note that in some applications, the brute-force attack takes place on private key of some fixed size, in which case the length of the secret key is often known.
2) Passwords are assumed to be generated i.i.d. from a distribution \( P_X \).\(^1\)
3) The goals of the adversaries is to guess one given password. In practice, there might be multiple accounts which undergo attacks simultaneously.
4) The adversaries have no additional information about the users and make their guessing based solely on \( P_X \).

We believe that some of these assumptions could be relaxed and generalized using techniques from the literature, as discussed below. Despite these assumptions, the insights gained from the model we study shed light on the robustness of brute-force attacks to asynchronization. To illustrate this claim, we have shown our results on an extract of the Adobe Leaked password dataset (see [6] for a description of the dataset). In particular, we extracted the \( 10^4 \) most likely passwords from a subset of 10

\(^1\)We briefly mention generalizations to passwords generated according to an irreducible stationary Markov Chain in Remark 1 in Section III.
(a) Probability of finding the password in fewer than \( i \) queries. In a synchronized attack, the passwords has to be found after at most \( |X| = 1 \times 10^4 \) queries. The blue and orange line correspond to i.i.d. guesses according to the distribution \( \hat{P} \).

(b) Log-probability mass function. Notice how the tilted distribution gives more weight to less likely symbols, as they correspond to the symbol which are the most costly for password guessing.

Fig. 2: Experiments on a subset of Adobe Leak password data (only \( 10^4 \) unique passwords kept). Despite the heavy tail of the distribution, a randomized strategy with some tilt improves the log expected number of guesses from 9.2 when using the naive distribution, to 8.8 when using the optimal tilt.

millions passwords in the data, and restricted our study to those passwords. We investigate the guesswork when the correct password is drawn according to the distribution \( P_X \) as computed on this restricted sample of the data. We show in Figure 2 the performance of a randomized strategy when using the optimal guessing distribution versus the naive distribution \( P_X \), both in terms of expected number of guesses and in terms of probability of making less than a fixed number of guesses. Note that the true distribution \( P_X \) performs well if one wish to make only a limited number of guesses, but eventually takes longer to reach a high probability. This is due to less frequent passwords, which are barely ever queried if guesses are drawn according to \( P_X \). The optimal distribution attributes more weight to these less likely symbols.

Related Work: The problem of a cipher with a guessing wiretapper was considered in [7]. The problem of guessing subject to distortion and constrained Shannon entropy were investigated in [8] and [9], respectively. The above results have been generalized to ergodic Markov chains [10] and a wide range of stationary sources [11]. The problem of guessing under source uncertainty was investigated in [12]. The analysis of the guessing exponents, using large deviations theory, was considered in [13]. In [14] it was shown that the guesswork satisfies a large deviation property and the rate function was characterized. They also provided an approximation to the distribution of guesswork using the large deviation property. Guesswork under erasures was studied in [15]. A brute-force attack where adversaries are interested in multiple passwords is discussed in [16]. A distributed attack model based on password hints was proposed in [17] and evaluated under guesswork metrics, and a wiretap system under guessing guarantees was studied in [18]. Finally, a geometric characterization of the guesswork was established in [19] and expanded in [20].

Main Contributions: We define a min-max formulation that models a worst case asynchronous attack, and show that a randomized strategy in which each guess is drawn i.i.d. from a certain distribution achieves the same asymptotic performance as an optimal synchronized attack. This optimal distribution is non-trivial; performing guesses according to the distribution from which the password was generated yields a strategy that is exponentially worse than the optimal distribution. In fact, the optimal choice is a tilted distribution, where the tilt parameter is chosen depending on the moment of guesswork which is optimized. We also discuss optimal strategies when the benchmark is to maximize the probability of success of an attack with a fixed number of overall queries, and show that an i.i.d. guessing strategy has again optimal performance asymptotically. The optimal distribution is again a tilted distribution, where the tilt depends on the number of queries allowed. Together these results indicate that there is no loss in performance (asymptotically) when performing an asynchronous attack.

The paper is organized as follows. In Section II we establish some notation and provide a brief background on the guessing problem. We discuss the impact of synchronization under the number of guesses in Section III and then under the probability of a system breach with a fixed number of queries in Section IV.

II. NOTATION AND BACKGROUND

Throughout this paper, scalar random variables (RVs) will be denoted by capital letters, their sample values will be denoted by the respective lower case letters, and their alphabets will be denoted by the respective calligraphic letters, e.g. \( X, x, \) and \( \mathcal{X} \), respectively. We also use the notation \( X_n \) to designate the sequence of RVs \( (X_1, \ldots, X_n) \), and may drop the subscript
A (possibly randomized) guessing strategy is a sequence $\log(D)$ denoted by $\hat{\rho}$. Specifically, among other things, it was shown that for any constraint on the set of possible guessing strategies, the optimal guessing strategy is obtained by ordering the symbols in $X$, where the expectation is taken over the distribution $\hat{\rho}$. A similar notation will be used for divergences, e.g., $D(p_1||p_2)$.

For a given vector $x_n$, let $P_{x_n}$ denote the empirical distribution, that is, the vector $\{\hat{P}_{x_n}(x), x \in \mathcal{X}\}$, where $\hat{P}_{x_n}(x)$ is the relative frequency of the letter $x$ in $x_n$. Let $T(\hat{P}_X)$ denote the type class associated with $\hat{P}_X$, that is, the set of all sequences $x_n$ for which $\hat{P}_{x_n} = P_X$.

The cardinality of a finite set $\mathcal{A}$ will be denoted by $|\mathcal{A}|$, its complement will be denoted by $\mathcal{A}^c$. The probability of an event $\mathcal{E}$ will be denoted by $\Pr\{\mathcal{E}\}$. For two sequences of positive numbers, $\{a_n\}$ and $\{b_n\}$, the notation $a_n = b_n$ means that $\{a_n\}$ and $\{b_n\}$ are of the same exponential order, i.e., $n^{-1}\log a_n/b_n \to 0$ as $n \to \infty$, where logarithms are defined with respect to (w.r.t.) the natural basis, that is, $\log(\cdot) = \ln(\cdot)$. Finally, for a real number $x$, we denote $[x]^+ = \max(0,x)$.

**Guessing Functions and Strategies:** A (possibly randomized) guessing strategy is a sequence $\hat{X}^\infty = \{\hat{X}_k(P_X) : k \geq 1\}$, where $\hat{X}_k(P_X) \in \mathcal{X}$, is independent of the realization $X$ but may depend on $P_X$. In other words, $\hat{X}^\infty$ is the list of guesses the attacker will use one after the other when trying to guess $X$. The corresponding guessing function, $G(X, \hat{X}^\infty)$, defined as

$$G(X, \hat{X}^\infty) = \inf \left\{ k \geq 1 : \hat{X}_k(P_X) = X \right\}, \quad (1)$$

represents the number of queries before reaching $X$. The $\rho$-th moment of the number of guesses is thus given by $\mathbb{E}[G(X, \hat{X}^\infty)^\rho]$, where the expectation is taken over the distribution $P_X$ and the randomness inherent in the guessing strategy $\hat{X}^\infty$. The $\rho$-th moment *guesswork* of a source $X \sim P_X$ is given by

$$\min_{\hat{X}^\infty} \mathbb{E}\left[G(X, \hat{X}^\infty)^\rho\right], \quad (2)$$

where the minimization is over all guessing strategies. In particular, the first moment, i.e. $\rho = 1$, corresponds to the average number of guesses that an adversary would have to perform before guessing the correct $X$.\(^2\) It was shown in [21] that, without any constraint on the set of possible guessing strategies, the optimal guessing strategy is obtained by ordering the symbols in $\mathcal{X}$ by decreasing order of $P_X$-probabilities, with ties broken arbitrarily, resulting in a deterministic strategy $\{\hat{x}_k(P_X) : k \geq 1\}$. The resulting guessing function, denoted by $G^*(X)$, represents the position of $X$ in the optimal list, i.e. the list of symbols ordered from most likely to least likely.\(^3\) The problem of bounding the expected number of guesses was investigated in [22]. Specifically, among other things, it was shown that for any $\rho \geq 0$, and any guessing function $G(\cdot)$,

$$\mathbb{E}[G(X)^\rho] \geq (1 + \log |\mathcal{X}|)^{-\rho} \sum_{x \in \mathcal{X}} P_X(x) \log P_X(x)^{1+\rho} . \quad (3)$$

On the contrary, the optimal guessing function, satisfies\(^4\)

$$\mathbb{E}[G^*(X)^\rho] \leq \left[ \sum_{x \in \mathcal{X}} P_X(x) \log P_X(x)^{1+\rho} \right] . \quad (4)$$

Finally, letting $X = (X_1, X_2, \ldots, X_n)$ be a sequence of independent and identically distributed (i.i.d.) random variables over a finite set, and letting $G^*(X)$ denote the optimal guessing function of a realization of $X$, it was shown that [22, Proposition 5]

$$E_\rho(P_X) \triangleq \lim_{n \to \infty} \frac{1}{n} \log \mathbb{E}[G^*(X)^\rho] = \rho \cdot H_{1+\rho}(X_1) , \quad (5)$$

where $H_{\alpha}(X)$ is the Rényi entropy of order $\alpha$ ($\alpha > 0$, $\alpha \neq 1$), defined as

$$H_{\alpha}(X) \triangleq \frac{1}{1-\alpha} \log \left[ \sum_{x \in \mathcal{X}} P_X(x)^\alpha \right] . \quad (6)$$

\(^2\)Although the most relevant moment to consider for practical purposes is the expectation, i.e. $\rho = 1$, we consider here a more general quantity as to be consistent with existing literature on guesswork.

\(^3\)Note that the optimal strategy $G^*(X)$ is deterministic, therefore the dependence on $\hat{X}^\infty$ is dropped in the notation.

\(^4\)An improved bound by a factor of 2 was reported in [23].
Note that the function $E_r(P_X)$ simply quantifies the exponential growth of the guesswork, as $n \to \infty$. We note that (5) gives an asymptotic operational characterization/meaning to Rényi entropy of order $0 \leq \alpha \leq 1$.

### III. Asynchronous Brute-Force Attack

In this section, we discuss synchronization when multiple agents aim to breach a secured system. Recall that we say that distributed agents are synchronized if they know in which order every agent’s queries will be received by Alice. In this case, they can query from the optimal list as a group, i.e., the first query received is the most likely symbol, etc. In other words, full synchronization means they can all share a single (optimal) list, and a pointer to this list advancing after each new guess. As a result, the total number of queries sent is the same as the optimal single agent guesswork, namely, (5) is achieved, while the individual computational burden on each agent is reduced since the queries are divided among agents. Further, even if the number of adversaries grows exponentially\(^5\) with the length of the password $n$, the total number of queries remains the same.

Instead, if agents do not know in which order the queries are delivered, they must adopt a strategy which performs well under any such ordering. In particular, we shall adopt a worst-case approach in which the goal is to minimize the number of queries in the worst ordering. Specifically, let $X$, an i.i.d. sequence of length $n$ generated from $P_X$, be the sequence to be guessed, and let $\{X_k^{(i)} : k \geq 1\}$ be the strategy of agent $t \in T$, where $T$ is possibly an infinite countable set. Again, we shall be interested in the regime where $|T|$ grows at least exponentially fast with $n$, and the goal is to characterize number of queries made in total. We let the permutation $\pi : \mathbb{N}^+ \to T \times \mathbb{N}^+$ denote the ordering in which the queries are received, i.e., $\pi(i) = (t_i, k_i)$ means that the $i$-th query received is $X_{k_i}^{(t_i)}$. Denote by $\Pi$ the set of all such possible orderings. Under an ordering $\pi$, Alice receives the sequence of queries $\pi(X_{1}^\infty) \triangleq \{X_{k_i}^{(t_i)} : i \geq 1\}$. Note that this permutation allows reordering of guesses of a given agent $t \in T$ which may be received in any arbitrary order. For some fixed strategies $\{X_k^{(i)} : k \geq 1\}$, the worst ordering in terms of guesswork is thus given by

$$\sup_{\pi \in \Pi} \mathbb{E}\left\{G(X, \pi(X_{1}^\infty))^\rho\right\}. \quad (7)$$

The goal of the agents is to minimize the worst-case number of queries, or, in other words, solve the min-max problem

$$\inf_{\{X_k^{(i)} : k \geq 1\} \text{ for } t \in T} \sup_{\pi \in \Pi} \mathbb{E}\left\{G(X, \pi(X_{1}^\infty))^\rho\right\}. \quad (8)$$

The main result of this section, presented below, characterizes the asymptotic exponent of (8), as $n \to \infty$. The proof of this result, along with the associated lemmas, are given after some discussion.

**Theorem 1** For $X$ an i.i.d. sequence according to $P_X$, and $\{X_k^{(i)} : k \geq 1\}$ sequences of guesses which are independent over $t \in T$, we have the following

$$\lim_{n \to \infty} \frac{1}{n} \log \left( \inf_{\{X_k^{(i)} : k \geq 1\}} \sup_{\pi \in \Pi} \mathbb{E}\left\{G(X, \pi(X_{1}^\infty))^\rho\right\} \right) = \sup_{\rho} \frac{1}{H_{\frac{1}{1+\rho}}(X)} \mathbb{E}\left\{G^*(X)^\rho\right\} = \rho \cdot H_{\frac{1}{1+\rho}}(X). \quad (9)$$

Note that guesswork measures the total number of guesses made by the agents. Thus it is clear that with full synchronization between the agents this value will not depend on $|T|$. In a sense, dependence on $|T|$ for a certain scheme would indicate a lack of synchronization, as it would suggest that queries are repeated by the agents. Surprisingly, Theorem 1 states that even under a worst-case assumption, there exist a strategy under which the guesswork does not depend on $|T|$ and is similar to the fully synchronous case. The above result and (5) show that synchronization is not necessary to achieve the asymptotic optimal performance. This can be equivalently formulated by an achievability strategy, and a converse. The converse result is trivial, as the performance of the synchronized strategy $\mathbb{E}\{G^*(X)\}$ upper bounds (8).

**Lemma 1 (Converse)** For any strategy $\hat{X}_{1}^\infty$,

$$\inf_{\{\hat{X}_k^{(i)} : k \geq 1\} \text{ for } t \in T} \sup_{\pi \in \Pi} \mathbb{E}\left\{G(X, \pi(\hat{X}_{1}^\infty))^\rho\right\} \geq \mathbb{E}\{G^*(X)\}. \quad (10)$$

We now turn to finding an appropriate strategy which would match this converse bound. Let us first examine a naive solution to this problem. Consider the strategy which consists in letting each agent construct the optimal list and query it individually,

\(^5\)Note that in practice, the number of agents usually needs to grow since most secured systems include a mechanism which blocks IP addresses after a given number of password attempts. Thus, if a single agent can only make $k$ queries, there must be at least $\lceil |X|^\alpha / k \rceil$ agents to guarantee that a password of length $n$ will be found.
that is $X_1^{(t)}$ is the most likely symbol for all $t \in T$, $X^{(t)}$ the second most likely symbol, etc. It is easy to see that (7) would evaluate to a quantity which grows with the number of agents $|T|$. Indeed, many queries are duplicated, and thus the overall number of queries grows with $|T|$, without even reducing the computational burden on each adversary since they all must query the same password strings. Note that this remains true if one considers a less stringent worst-case analysis, by for example, letting the guesses of each of the agent to be consistent among themselves, i.e. the permutation does not change the relative order of the guesses of each agent.

If instead the agents agree on a partition of the guesses before the attack, in a way such that no two guesses are repeated, then the correct password is queried by one unique agent. Again, it is easy to see that the worst-case analysis yields a quantity which grows with $|T|$, even though it cannot grow beyond $|\mathcal{X}|^n$, as every unique password is queried at most once. In particular, if $|T| = |\mathcal{X}|^n$, then the worst-case analysis achieves its upper-bound. Note that these observations are not only an artifact of the deterministic nature of the queries.

This motivates us to study randomized strategies. In particular, we consider guesses, which are randomly and independently drawn according to a specific distribution, independent from each other, and identically distributed. We then study this optimal distribution in terms of the expected moments of guesswork. Consider first a scalar $X \in \mathcal{X}$, generated from $P_X$. We let $\{\hat{X}_k^{(t)}, k \geq 1\}$ be an i.i.d. process with respect to a certain distribution, independently, and without a list, then using the original distribution is strictly sub-optimal, i.e. the permutation does not change the relative order of the guesses of each agent.

For any integer $\rho \geq 1$,

$$\log \mathbb{E}\{V_{\rho}^{*}(X, \hat{X}^{\infty})\} = \rho \cdot H_{\frac{1}{\alpha + \rho}}(X),$$

and for any $x \in \mathcal{X}$,

$$\hat{P}_{\rho}^{*}(x) = \frac{P_X(x)^{\frac{1}{\alpha + \rho}}}{\sum_{x^\prime \in \mathcal{X}} P_X(x^\prime)^{\frac{1}{\alpha + \rho}}}. \tag{15}$$

Before providing the proof of Lemma 2 we briefly discuss our result. First, we note that contrary to (5), the above result provides an exact operational meaning for Rényi entropy $H_{\alpha}(X)$ of order $\alpha > 0$. It should be mentioned here that a similar interpretation for $H_{1/2}(X)$ was reported in [24, 25]. Also, we see that the optimal guessing distribution (15) is simply the tilted distribution of $P_X$ of order $1/(1 + \rho)$. It should be emphasized that, since the function $f(x) = x^{1/(1 + \rho)}$ is monotone, creating an optimal list according to $P_X$ yields the exact same list as if done according to $P_X$. However, the list of guesses chosen i.i.d. according to $\hat{P}_X$ will be different from the one if guesses are made i.i.d. according to $P_X$. Indeed, letting $\hat{P}(x) = P_X(x)$ gives

$$\log \mathbb{E}\{G(X, \hat{X}^{\infty})\} = \log |\mathcal{X}|,$$

which could be much worse than $\log \mathbb{E}\{V_{1}^{*}(X, \hat{X}^{\infty})\} = H_{1/2}(X)$. Namely, when one is allowed only to guess passwordds according to a certain distribution, independently, and without a list, then using the original distribution is strictly sub-optimal, and the tilted distribution should be used. This result is related to similar results from the source-coding literature in which a tilted distribution also appears as the solution of an optimization where longer codewords are penalized exponentially (see e.g. [26, 27]). Finally, note that the result is not asymptotic. In particular, the randomized strategy can be used over an alphabet $\mathcal{X}$ where each $x \in \mathcal{X}$ corresponds to a password. This result is thus relevant to dictionary attacks, where queries are drawn according to a dictionary of possible passwords, and suggests that distributed dictionary attacks should use a guessing distribution which is a tilted version of the true distribution.
Proof 1 (Proof of Lemma 2) First, note that given $X$, $G(X, \hat{X}_n^\infty)$ is a geometric random variable, and for $k \geq 1$,
\[
\Pr\{G(X, \hat{X}_n^\infty) = k\} = \sum_{x \in \mathcal{X}} P_X(x)(1 - \hat{P}(x))^{k-1} \hat{P}(x).
\]
Then, for any $\rho > 0$, we have
\[
\mathbb{E}\{V_\rho(X, \hat{X}_n^\infty)\} = \sum_{m=1}^{\infty} \binom{m + \rho - 1}{m - 1} \Pr\{G(X, \hat{X}_n^\infty) = m\}
= \sum_{x \in \mathcal{X}} P_X(x) \hat{P}(x) \sum_{m=1}^{\infty} \binom{m + \rho - 1}{m - 1} (1 - \hat{P}(x))^{m-1}.
\]
In the following, we calculate the second summation term in the r.h.s. of the last equality. This is equivalent to calculating
\[
\sum_{m=1}^{\infty} \binom{m + \rho - 1}{m - 1} \rho^{m-1}.
\]
Note that, using the identity $\Gamma(x + 1) = x\Gamma(x)$ recursively, we get that
\[
\frac{\Gamma(m + \rho)}{\Gamma(\rho + 1)} = (m + \rho - 1) \cdot (m + \rho - 2) \cdots (\rho + 1)
= (-1)^{m-1}(-\rho - 1) \cdot (-\rho - 2) \cdots (-\rho - m + 1)
= (-1)^{m-1} \frac{\Gamma(-\rho)}{\Gamma(-\rho - m + 1)},
\]
which yields \(\binom{m + \rho - 1}{m - 1}\) = \((-1)^{m-1}(-\rho - 1)\), and together with the change of variable $k = m - 1$ we obtain
\[
\sum_{m=1}^{\infty} \binom{m + \rho - 1}{m - 1} \rho^{m-1} = \sum_{k=0}^{\infty} \binom{-\rho - 1}{k} (-y)^k
= (1 - y)^{-\rho-1},
\]
where the last equality follows from the binomial formula. Thus,
\[
\mathbb{E}\{V_\rho(X, \hat{X}_n^\infty)\} = \sum_{x \in \mathcal{X}} P_X(x) \hat{P}(x) \frac{1}{\hat{P}(x)^{1+\rho}}
= \sum_{x \in \mathcal{X}} \frac{P_X(x)}{\hat{P}(x)^{\rho}}.
\]
Next, we minimize the last expression with respect to $\hat{P} \in \mathcal{P}$. To this end, since (19) is convex in $\hat{P}$, $\hat{P}^*$ is given by the solution of (for $x \in \mathcal{X}$)
\[
-\rho \cdot \frac{P_X(x)}{\hat{P}^*(x)^{\rho+1}} + \lambda = 0,
\]
where $\lambda$ is a Lagrange multiplier, and thus,
\[
\hat{P}^*(x) = \frac{P_X(x)^{\frac{1}{\rho+1}}}{\sum_{x' \in \mathcal{X}} P_X(x')^{\frac{1}{\rho+1}}}.
\]
On substituting this optimal distribution in (19) we finally get
\[
\mathbb{E}\{V_\rho^*(X, \hat{X}_n^\infty)\} = \sum_{x \in \mathcal{X}} \frac{P_X(x)}{\hat{P}^*(x)^{\rho}} = \left(\sum_{x \in \mathcal{X}} P_X(x)^{\frac{1}{\rho}}\right)^{1+\rho},
\]
as claimed.

The previous lemma applies to a scalar RV $X$, but can be easily extended to sequences $X_n$, as shown in the following corollary.

Corollary 1 Let $X$ be a sequence of length $n$ generated i.i.d. from $P_X$. Then, we have,
\[
\lim_{n \to \infty} \frac{1}{n} \log \mathbb{E}\{V_\rho^*(X, \hat{X}_n^\infty)\} = \rho \cdot H^{\frac{1}{\rho+1}}(X).
\]
Remark 1 We note that the result above can be generalized to passwords which are generated according to an irreducible stationary Markov Chain. More precisely, let $U$ be the stochastic matrix and stationary distribution of the Markov chain, respectively, so that

$$
\Pr \{ X = (x_1 \ldots x_n) \} = \gamma x_1 \prod_{i=1}^{n-1} U_{x_i, x_{i+1}}
$$

(21)

Then, it was shown in [10] that

$$
\lim_{n \to \infty} \log \mathbb{E} \{ G^*(X)^{\rho} \} = \frac{1}{1 + \rho} \log \lambda,
$$

(22)

where $\lambda$ is the Perron-Frobenius eigenvalue of the matrix with entries $V = (U_{ab}^{1/1+\rho})$ for $a, b \in \mathcal{X}$. Further, let $\{v_a\}$ and $\{w_a\}$ be the left and right eigenvectors of $V$ associated with $\lambda$, that is

$$
\sum_{a \in \mathcal{X}} v_a = 1, \quad \sum_{a \in \mathcal{X}} v_a V_{ab} = \lambda v_b, \quad \sum_{b \in \mathcal{X}} w_b V_{ab} = \lambda w_a.
$$

(23)

Analogously to the result of Corollary 1, it can be shown that generating guesses $\hat{X}$ according to a Markov Chain with entries $V_{ab}^{ab} / (\lambda w_a)$ achieves the asymptotic performance in (21). A proof of this fact is outside the scope of this paper, but follows from steps outlined in [10] along with the proof of Lemma 2.

Remark 2 In the standard guessing problem [22] Alice tries to guess $X$ using her knowledge of $P_X$. It is assumed that there are no constraints on the memory of Alice, namely, for each new guess, Alice knows her previous guesses, and thus she can adapt her new guess accordingly (i.e., she will not guess again a previous incorrect guess). The setting we consider here is equivalent to one in which Alice cannot keep track of her guesses, but still knows the distribution $P_X$. It should be clear that in this case all that Alice can do is to present a sequence of i.i.d. guesses $\hat{X}_1, \hat{X}_2, \ldots$, drawn from some distribution $\hat{P}(\cdot)$, which shall be optimized in some sense. Lemma 2 can be equivalently interpreted as the performance of a memoryless attacker [24, 25, 28].

We are now ready to prove Theorem 1.

Proof 3 (Proof of Theorem 1) We start by noting that letting $\{\hat{X}^{(t)}_k : k \geq 1\}$ be an i.i.d. process distributed according to $\hat{P}^*$ (as defined in Lemma 2) gives an upper bound on (8). We prove that two bounds match asymptotically, by showing that the exponent of the upper-bound is equal to $\rho \cdot H_{1/(\rho+1)}(X)$. Indeed, let $\{X_k^{(t)} : k \geq 1\}$ be an i.i.d. process distributed according to $\hat{P}^*$ for all $t \in T$. Then, it is evident that $\pi(X_k^{\infty})$ is also an i.i.d. process distributed according to $\hat{P}^*$, for any permutation $\pi \in \Pi$. An application of Corollary 1 concludes the proof.

Note that the optimal distribution from Lemma 2 depends on the moment $\rho$. Indeed, the larger $\rho$, the more we are penalized for passwords which are less frequent (which increase the work significantly). Therefore, the optimal strategy gives extra weight to less frequent symbols as to make sure that they are more likely to be chosen than what their probability suggests. We do so by raising $P_X$ to a power $1/1+\rho$. Nevertheless, the optimal distribution, and thus guessing strategy, will change as a function of the guesswork moment $\rho$ of interest. This contrasts with the synchronous case, in which the optimal strategy consisting of querying the sequences from most likely to least likely is optimal universally for all moments $\rho$. This loss of universality can be quantified in the following corollary, which characterizes the loss in using a distribution optimized for a moment $\rho > 0$, when measured in terms of a moment $\gamma \neq \rho$, and is illustrated for a binary source in Figure 3.

Corollary 2 Let $\{\hat{X}_k : k \geq 1\}$ be an i.i.d. process generated according to $\hat{P}^*_\gamma(x)$. Then:

$$
\log \mathbb{E} \{ V^*_\rho(X, \hat{X}_k^{\infty}) \} = \frac{\rho}{1+\gamma} H_{\frac{1}{1+\gamma}}(X) + \frac{\gamma}{1+\gamma} \cdot \frac{\rho}{1+\gamma} H_{\frac{1}{1+\gamma}}(X).
$$

(24)
Fig. 3: This plots compares the performance of the randomized strategy as a function of the moment $\rho$. We compare the optimal strategy which depends on $\rho$, against a fixed tilted distribution ($\gamma = 1$ in Corollary 2), when $X \sim \text{Ber}(1/5)$.

**Proof 4**  The proof follows by plugging $\hat{P}(\cdot) = \hat{P}^*_\gamma(\cdot)$ into (19).

Lemma 2 can also be easily generalized to the case of availability of some side information $Y$ which is correlated with $X$. That is, $(X,Y)$ is now a pair of random variables with joint distribution $P_{XY}$. Then, assume that the guesser generates a sequence of guesses $\hat{X}_1, \hat{X}_2, \ldots$ which are i.i.d. given $Y$, and distributed according to $\hat{P}_{X|Y}(\cdot|\cdot)$. As before, we define $G(X, \hat{X}_\infty|Y) \triangleq \inf\{k \geq 1 : \hat{X}_k(Y) = X\}$. Then, following the proof of Theorem 2 we can show that the optimal guessing distribution is

$$
\hat{P}^*_X(x|y) = \frac{P_{X|Y}(x|y) + \rho}{\sum_{x' \in \mathcal{X}} P_{X|Y}(x'|y) + \rho}
$$

for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, and

$$
\log \mathbb{E}\{V^*_\rho(X, \hat{X}_\infty|Y)\} = \rho \cdot H_{\frac{1}{1+\rho}}(X|Y),
$$

where $H_{\alpha}(X|Y)$ is the conditional R´enyi entropy of order $\alpha$, and $V^*_\rho(X, \hat{X}_\infty|Y)$ is defined as in (13) but with $G(X, \hat{X}_\infty|Y)$ replaced by $G(X, \hat{X}_\infty|Y)$.

**IV. CONSTRAINTS ON THE NUMBER OF GUESSES**

In Section III, we considered the case in which guesses are made until the correct sequence is found. In this section, we consider the case where adversaries can use only a fixed number of guesses. More precisely, we focus on the scenario of guessing $n$-length i.i.d. sequences, and we assume the adversaries make $J = \lceil X^{\alpha \gamma} \rceil$ total guesses. We analyze the success probability in guessing the correct sequence and derive expressions which are exponentially tight as a function of $n$. We consider both the synchronized case [22] as well as the asynchronous case.

We start with synchronized guessers. Let $\mathcal{L}$ designate the set consisting of the $J$ most likely sequences according to $P_X(\cdot)$. The probability of success associated with the optimal guessing strategy is given by

$$
P_{\text{synchr}}^{\text{c},J} = \sum_{x \in \mathcal{L}} P_X(x).
$$

Also, we define the exponential rate of $P_{\text{synchr}}^{\text{c},J}$ as

$$
E_{\text{c},\alpha}^{\text{synchr}} \triangleq \liminf_{n \to \infty} -\frac{1}{n} \log P_{\text{synchr}}^{\text{c},J}.
$$

The following result is an immediate application of the large deviation principle of Guesswork, shown in [14].
Corollary 3 essentially proves that the tilted family is asymptotically optimal, and that there exist a unique optimal tilt $\beta$ for each size list $J = \lceil X^{nH}\rceil$.

Proof (Proof of Corollary 3) By definition, $\min_{\beta \geq 0} E_{c,\alpha}^{\text{asynchr}} \geq 0$. Then, for $\alpha \geq H(P_X)$, we see from Theorem 3 that by taking $Q_X = P_X$ and $\beta = 1$, we have

$$\min_{\beta \geq 0} E_{c,\alpha}^{\text{asynchr}} \leq [H(P_X) - \alpha]_+ = 0.$$
For $\alpha < H(P_X)$, we first note that by definition $\min_{\beta \geq 0} E_{c,\alpha}^{\text{asynchr}} \geq E_{c,\alpha}^{\text{synchr}}$. Hence, due to Theorem 2 and Lemma 3 in the appendix we may conclude that

$$\min_{\beta \geq 0} E_{c,\alpha}^{\text{asynchr}} \geq D(Q_X^* \| P_X),$$

where $Q_X^*$ is the solution of the optimization

$$\begin{align*}
\text{minimize} & \quad D(Q_X \| P_X) + H(Q_X) \\
\text{subject to} & \quad H(Q_X) \geq \alpha.
\end{align*}$$

(38)

On the other hand, by taking $Q_X = Q_X^*$, we have

$$\min_{\beta \geq 0} E_{c,\alpha}^{\text{asynchr}} \leq D(Q_X^* \| P_X) + \min_{\beta \geq 0} \left[ \sum_{x \in Q} D(Q_X \| P_X) \right].$$

It is a simple exercise to verify that $Q_X^*$ is a tilted distribution, i.e. there exist a $\tilde{\beta}$ such that $Q^*(x) = \frac{Q_X(x)^{\beta}}{\sum_{x'} Q_X(x')^{\beta}}$. Letting $\beta = \tilde{\beta}$ gives

$$\min_{\beta \geq 0} E_{c,\alpha}^{\text{asynchr}} \leq D(Q_X^* \| P_X).$$

(40)

The result follows from combining (38) and (40).

We next provide the proofs of Theorems 2 and 3.

**Proof 6 (Proof of Theorem 3)** For simplicity of presentation, we prove the theorem for binary sequences, i.e. $X = \{0, 1\}$, and assume that $1/2 \geq p \triangleq P_X(0)$. For any given sequence $x^n \in X^n$,

$$\frac{1}{n} \log \hat{P}_{X^n}(x^n) = -D(\hat{P}_{X^n} \| \bar{p}^\beta) - H(\hat{P}_{X^n}),$$

(41)

where $\hat{P}_{X^n}$ is the empirical measure of a given sequence $x^n$, and $\bar{p}^\beta = \frac{p^\beta}{p^\beta + (1-p)^\beta}$. Then,

$$P_{c,J}^{\text{asynchr}} = \sum_{x^n \in X^n} P_{X^n}(x^n) \left[ 1 - (1 - \hat{P}_{X^n})^J \right]$$

$$= \sum_{x^n \in X^n} 2^{-n(D(\hat{P}_{X^n} \| p) + H(\hat{P}_{X^n}))} \times \left[ 1 - (1 - 2^{-n(D(\hat{P}_{X^n} \| \bar{p}^\beta) + H(\hat{P}_{X^n}))})^J \right].$$

Letting $Q_n$ denote the set of possible types, i.e. $Q_n \triangleq \{0, 1/n, 2/n, \ldots, n/n\}$ we obtain,

$$P_{c,J}^{\text{asynchr}} = \sum_{q \in Q_n} |T(q)| 2^{-n[D(q) \| p] + H(q)}$$

$$\times \left[ 1 - (1 - 2^{-n[D(q) \| \bar{p}^\beta] + H(q)})^J \right]$$

$$\leq \sum_{q \in Q_n} 2^n H(q) 2^{-n[D(q) \| p] + H(q)} \times 2^{-n[D(q) \| \bar{p}^\beta] + H(q) - \alpha}$$

$$= \max_{q \in [0, 1]} 2^{-n[D(q) \| p] + [D(q) \| \bar{p}^\beta] + H(q) - \alpha}$$

where the fourth equation follows from the fact that (see, e.g., [29, Lemma 1]) if $a \in [0, 1]$, then $\frac{1}{2} \min \{ 1, aM \} \leq 1 - (1 - a)^M \leq \min \{ 1, aM \}$. Thus, we have shown that

$$E_{c,\alpha}^{\text{asynchr}} = \min_{q \in [0, 1]} \left\{ D(q) \| p + [D(q) \| \bar{p}^\beta] + H(q) - \alpha \right\}.$$
V. Conclusion

In this paper, we have studied the impact of synchronization on brute-force attacks. We showed that despite a lack of synchronization, and considering a worst-case ordering of the guesses, a randomized guessing strategy allows to achieve the optimal asymptotic performance, both in terms of average number of guesses, and in terms of probability of success after a given number of steps. As such, a solution which prevents repeated queries from a single IP is not enough, and in fact does not guarantee security against even completely asynchronous adversaries. This highlights the importance of password selection, as increasing the guesswork is the key to a secure password-based system.

The insights from these randomized strategies also applies to a single attacker who attempts to breach a system which is likely to be attacked by many other sources of attack. Against such as system, the attacker’s strategy is analogous to one of a bot in a botnet. Indeed, since the system is likely to have been targeted by other attacks, the attacker might not want to follow his list in a deterministic way as to avoid repeating guesses from the other attackers. Using a randomized strategy does not hurt the performance asymptotically, but can prevent these repeated guesses.

A natural next step is to consider a distributed brute-force attack which aim at breaching any of $V$ password-secured accounts, rather than being aimed towards a single account. In this case, the computational effort will depend on the number of accounts which are under attack. More precisely, a brute-force attack directed against the accounts of $V$ members might be deemed successful as soon as $U$ of those accounts are compromised for $U \leq V$, regardless of which $U$ are compromised. The case where $U = 1$ corresponds to a classical brute-force attack directed at a multi-user system, while letting $U \geq 2$ models attacks on some distributed storage system, in which, because of the redundancy, some but not all of the servers should be compromised to access content. Additionally, once a system is compromised through sufficiently many accounts, it may be much harder to reliably detect or counteract the actions of the attacker, e.g., in the case of a Byzantine attack (c.f. [30] or [31]). Generalizations of the standard Guesswork problem to this setting have been studied (see [16]), and establish the gain that arises from considering more accounts, especially when $U$ is much smaller than $V$. However, the optimal strategies in this case rely on a round-robin approach — assuming the password generation process for all users is identical. More precisely, one should make password guesses to each account in turns, first making a guess for the first account, then the second, and so-on, until eventually successfully guessing the passwords of $U$ of the $V$ accounts. Generalizing such attacks to a distributed asynchronous case is of interest, and the subject of some future work.

ACKNOWLEDGMENT

The authors are very grateful to Ken Duffy for many fruitful discussions and for providing the Adobe leaked password dataset used to generate Fig. 2.

REFERENCES

[1] “McAfee Labs Threat Report,” https://www.mcafee.com/ca/resources/reports/p-quarterly-threats-sept-2017.pdf, 2017.
[2] K. J. Hole, V. Moen, and T. Tjøstheim, “Case study: Online banking security,” IEEE Security & Privacy, vol. 4, no. 2, pp. 14–20, 2006.
[3] “Bitcoin Wallets under siege from ‘Large Collider’ Attack,” http://fortune.com/2017/04/15/bitcoin-collider/.
[4] J. Owens and I. Matthews, “A study of passwords and methods used in brute-force ssh attacks,” in USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2008.
[5] “‘Brute force’ cyber attack on Parliament compromised up to 90 email accounts,” http://www.telegraph.co.uk/news/2017/06/25/brute-force-cyber-attack-parliament-compromised-90-email-accounts/.
[6] “Anatomy of a password disaster Adobes giant-sized cryptographic blunder,” https://nakedsecurity.sophos.com/2013/11/04/anatomy-of-a-password-disaster-adobes-giant-sized-cryptographic-blunder/.
[7] N. Merhav and E. Arikan, “The shannon cipher system with a guessing wiretapper,” IEEE Trans. on Inf. Theory, vol. 45, no. 6, pp. 1860–1866, Sep. 1999.
[8] E. Arikan and N. Merhav, “Guessing subject to distortion,” IEEE Trans. on Inf. Theory, vol. 44, no. 3, pp. 1041–1056, May 1998.
[9] A. Beirami, R. Calderbank, K. Duffy, and M. Médard, “Quantifying computational security subject to source constraints, guesswork and inscrutability,” in 2015 IEEE International Symposium on Information Theory Proceedings, Jun. 2015.
[10] D. Sullivan and W. G. Sullivan, “Guesswork and entropy,” IEEE Trans. on Inf. Theory, vol. 50, no. 3, pp. 525–526, Mar. 2004.
[11] C. E. Pfister and W. G. Sullivan, “Rényi entropy, guesswork moments, and large deviations,” IEEE Trans. on Inf. Theory, vol. 50, no. 11, pp. 2794–2800, Nov. 2004.
[12] R. Sundaresan, “Guessing under source uncertainty,” IEEE Trans. on Inf. Theory, vol. 53, no. 1, pp. 525–526, Jan. 2007.
[13] M. K. Hanawal and R. Sundaresan, “Guessing revisited: A large deviations approach,” IEEE Trans. on Inf. Theory, vol. 57, no. 1, pp. 70–78, Jan. 2011.
[14] M. M. Christiansen and K. R. Duffy, “Guesswork, large deviations, and Shannon entropy,” IEEE Trans. on Inf. Theory, vol. 59, no. 2, pp. 796–802, Feb. 2013.
[15] M. M. Christiansen, K. R. Duffy, F. du Pin Calmon, and M. Médard, “Guessing a password over a wireless channel (on the effect of noise non-uniformity),” in Signals, Systems and Computers, 2013 Asilomar Conference on. IEEE, 2013, pp. 51–55.
[16] ——, “Multi-user guesswork and brute force security,” IEEE Transactions on Information Theory, vol. 61, no. 12, pp. 6876–6886, 2015.
[17] A. Bracher, E. Hof, and A. Lapidoth, “Guessing attacks on distributed-storage systems,” arXiv preprint arXiv:1701.01981, 2017.
[18] N. Merhav and E. Arikan, “The shannon cipher system with a guessing wiretapper,” IEEE Transactions on Information Theory, vol. 45, no. 6, pp. 1860–1866, 1999.
[19] A. Beirami, R. Calderbank, M. Christiansen, K. Duffy, A. Makhdoumi, and M. Médard, “A geometric perspective on guesswork,” in 53rd Annual Allerton Conference (Allerton), Oct. 2015.
[20] A. Beirami, R. Calderbank, M. Christiansen, K. Duffy, and M. Médard, “A characterization of guesswork on swiftly tilting curves,” arXiv preprint arXiv:1801.09021, 2018.
APPENDIX A

ADDITIONAL LEMMAS

The following lemma relate the position of a sequence $x_n$ in the optimal list, with the type of that sequence.

**Lemma 3** Let $x_n$ be a i.i.d. generated sequence, and consider the position of $x_n$ in the optimal list according to $P_X$, i.e. $G^*(x)$. For a given $\alpha$, we have that $G^*(x) < \lceil |X|^\alpha \rceil$ if and only if the sequence $x$ satisfy $\hat{P}_x \in \mathbb{Q}(\alpha)$, where

$$Q(\alpha) = \{Q_X : D(Q_X \parallel P_X) + H(Q_X) < D(Q_X^\ast \parallel P_X) + H(Q_X^\ast)\} ,$$  
(A.1)

with $Q_X^\ast$ being the solution of the optimization problem:

$$\min_{Q_X} \quad D(Q_X \parallel P_X) + H(Q_X)$$  
subject to \quad $H(Q_X) \geq \alpha$ \quad (A.2)

**Proof** Recall that $P_X(x) = \exp \{-n \left( D(\hat{P}_x \parallel P_X) + H(\hat{P}_x) \right) \}$, and that the size of the type set $T(\hat{P}_x) \doteq 2^{nH(\hat{P}_x)}$. Let $\mathbb{Q}(\alpha)$ be the set of types of the sequences that are in the first $\lambda^\alpha n$ position in the list optimal list. Then, by definition of $\mathbb{Q}(\alpha)$:

$$\sum_{Q_X \in \mathbb{Q}(\alpha)} 2^{nH(Q_X)} \doteq 2^{n\alpha}$$  
(A.3)

An application of the method of types gives that the left-hand side evaluates to $2^n \sup_{Q_X \in \mathbb{Q}(\alpha)} H(Q_X)$, meaning that $\sup_{Q_X \in \mathbb{Q}(\alpha)} H(Q_X) = \alpha$. Thus, the threshold probability is given by the type that solves (A.2), and any type that has lower probability must appears before in the list.