This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2019.2895955, IEEE Transactions on Information Forensics and Security

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

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

***Abstract*—In September 2017, McAffee Labs quarterly report [2] 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 queries (guesses) required of an adversary before a correct sequence, such as a password, is found in an optimal attack. Guesswork is a direct surrogate for time and computational effort of guessing a sequence from a set of sequences with associated likelihoods. We model the lack of synchronization by a worst-case optimization in which the queries made by multiple adversarial agents are received in the worst possible order for the adversary, 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.**

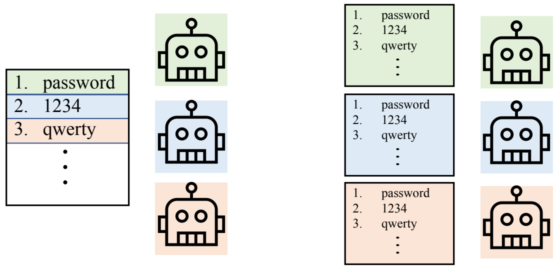
I. Introduction

From online banking [3] and bitcoin wallets [4], to secure shell (SSH), file transfer protocol (ftp), and telnet servers [5], and passing by governmental institutions [6], 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 uses 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 hard to detect legitimate users in the crowd of illegitimate attackers. These attacks come with a cost, namely, the attack is now distributed across thousands, sometimes 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 under 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 $\hat{X}_1, \hat{X}_2$, 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 is 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 ρ-th moment, i.e.

This paper was presented in part at the 2017 IEEE International Symposium on Information Theory [1]. Salman Salamatian, Wasim Huleihel, and Muriel Médard are with the department of Electrical Engineering and Computer Science, MIT, Cambridge MA (salmansa@mit.edu, wasimh@mit.edu, beirami@mit.edu, medard@mit.edu). A. Beirami was with the department of Electrical Engineering and Computer Science, MIT, Cambridge, MA. He is currently with EA Digital Platform – Data & AI, Electronic Arts, Redwood City, CA (ahmad.beirami@gmail.com). Asaf Cohen is with the department of Electrical Engineering, Ben-Gurion University of the Negev, Israel (coasaf@bgu.ac.il).



a) Synchronized attack

b) Asynchronized attack

Fig. 1: In a synchronized attack, the bots query from the password-list in a specified order. In the asynchronous attack, they do not know the order in which the queries will be sent. Our solution will consist at drawing guesses according to some distribution, instead of querying passwords one-by-one.

$\mathbb{E}[G^*(X)^\rho]$. 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 (we shall use adversary and agents interchangeably) coordinate their attack, 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, might affect 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 secrets that are long sequences drawn according to some types of distribution. This optimal guessing distribution is non-trivial, and, perhaps 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 of discussion, we have made the following assumptions on the password generation process, as well as on the brute-force attack itself.

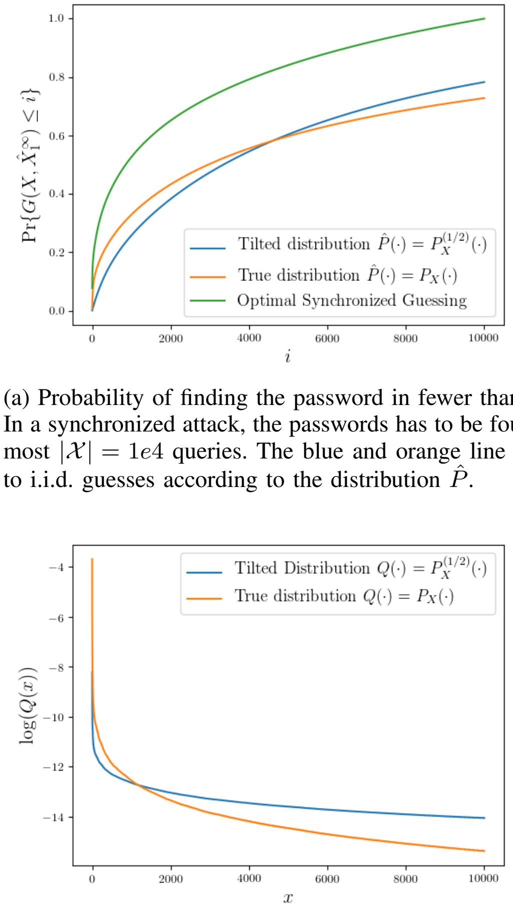
1) Passwords are assumed to be strings of given length n. Note that in some applications, the brute-force attack takes place on private key of some fixed size, in which case the length of the secret key is often known.
2) Passwords are assumed to be strings whose characters are generated i.i.d. from a distribution P_X.[1]
3) The common goal of the agents is to guess one given password, or string, a so called targeted attack. In practice, there might be multiple accounts which undergo attacks simultaneously.
4) The agents have no additional information about the users and may construct guesses based solely on P_X[2].

We believe that some of these assumptions could be relaxed and generalized using techniques from the literature, as discussed below. In addition, the i.i.d. setting, and the resulting asymptotic results, can be used as guidelines in designing systems even if the real system violates the memoryless assumption. For example, such results can be used to choose the minimal length of a password to secure a system. 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 [7] for a description of the dataset). In particular, we extracted the 10^4 most likely passwords from a subset of 10 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

[1] We briefly mention generalizations to passwords generated according to an irreducible stationary Markov Chain in Remark 1 in Section III.

[2] We also briefly discuss the presence of side-information during the attack, which an adversary might use to modify the distribution of the potential passwords at the end of Section III.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2019.2895955, IEEE Transactions on Information Forensics and Security



(a) Probability of finding the password in fewer than i queries. In a synchronized attack, the passwords has to be found after at most $|\mathcal{X}| = 1e4$ 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

one wishes to make only a small 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 guessing distribution which optimizes the average number of guesses increases the probability of querying the less likely passwords, as those passwords represent the main computational burden on the adversary when they occur.

Related Work: The problem of a cipher with a guessing wiretapper was considered in [8]. The problem of guessing subject to distortion and constrained Shannon entropy were investigated in [9] and [10], respectively. The above results have been generalized to ergodic Markov chains [11] and a wide range of stationary sources [12]. The problem of guessing under source uncertainty was investigated in [13]. The analysis of the guessing exponents, using large deviations theory, was considered in [14]. In [15] it was shown that the guesswork satisfies a large deviation property and the rate function was characterized. Guessing a sequence given an erased version of the sequence was studied in [16], where the interplay between the large-deviations of the erasure process, and of the sequence generation were characterized. A brute-force attack where adversaries are interested in multiple passwords is discussed in [17]. A distributed attack model based on password hints was proposed in [18] and evaluated under guesswork metrics, and a wiretap system under guessing guarantees was studied in [19]. A geometric characterization of the guesswork was established in [20] and expanded in [21]. Guesswork under an entropy budget was studied in [22]. Connections between guesswork and one-to-one coding were explored in [23]. Finally, applications of guesswork [24] to cryptographic guessing was studied in [24], where oblivious or memoryless guessers were studied. The results of [24] are non-asymptotic, but very much related to our setting, as optimal i.i.d. guessing strategies both in terms of number of guesses and in terms of probability of success are studied, and a distributed attack scenario is also envisaged.

The statistics of password generation were studied in [25-28]. Password frequencies have been shown to follow closely variants of the Zipf's law distribution. In particular, the so-called *CDF-Zipf's law* model introduced in [25, 26] is a modification of the Zipf's law which captures the frequencies of passwords, both for very frequent passwords, and the tails, as exhibited by the close empirical fit to multiple password datasets (see [26, 25, 27]). Note that an adversary can benefit greatly from the the non-uniformity of these distributions to design more powerful brute-force attacks. Indeed, Guesswork, and other related notions of security related to brute-force attacks are also studied in [25, 27, 28]. A special case of brute-force attack is given by *targeted attacks*, in which the adversary uses the personal information of an user in his guessing strategy, see e.g. [29]. Works such as [30, 31] empirically demonstrate the threat of these targeted attacks, as most users chose their passwords according to some personal information which an adversary might have easy access to (e.g. birthdays, names of family members, locations, or simply password reuse) .

Main Contributions: We define a min-max formulation that models a worst case asynchronous attack from the attacker's perspective, and show that a randomized strategy in which each guess is drawn i.i.d. from a certain distribution achieves the same asymptotic performance (in the length of the password sequence n) 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 guessing distribution. In fact, the optimal choice is a tilted distribution, where the tilt parameter is chosen depending on the moment of guesswork to be 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 again has 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 in n) 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.

Previous Publication: In a conference publication [1], we studied the problem of a memoryless guesser, and derived some of the technical results appearing in the present paper for binary sources, and integer moments $\rho > 0$. Although not directly related to botnets and asynchronous attacks, the technical results introduced in that paper are at the core of the analysis of an asynchronous distributed attack. In particular, Lemma 2, Theorem 2 and Theorem 3 are present, although restricted to binary sources and integer moments, in [1]. However, the connection to the asynchronous botnet problem is novel and made explicit here. In particular, the min-max formulation of Section III, along with Theorem 1, numerical results, and extensions of our previous results to general finite alphabets and non-integer moments distinguish this work from our previous publication. Finally, Corollary 2 which characterizes the loss in universality of distributed asynchronous attacks, is also novel.

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 $\mathbf{X}_n$ to designate the sequence of RVs $(X_1, \ldots, X_n)$, and may drop the subscript when the size of the sequence considered is clear from the context, e.g., $\mathbf{X}$. The set of all n-vectors with components taking values in a certain finite alphabet, will be denoted by the same alphabet superscripted by n, e.g., $\mathcal{X}^n$. Probability distributions will be denoted by the letters P and Q, with subscripts that denote the names of the random variables involved along with their conditioning, if applicable, following the customary notation rules in probability theory. For example, Q_{XY} stands for a generic joint distribution $\{Q_{XY}(x,y),\ x \in \mathcal{X},\ y \in \mathcal{Y}\}$, $P_{Y|X}$ denotes the matrix of single-letter transition probabilities, and so on.

The expectation operator will be denoted by $\mathbb{E}\left\{\cdot\right\}$, and when we wish to make the dependence on the underlying distribution Q clear, we denote it by $\mathbb{E}_Q\left\{\cdot\right\}$. The Kullback-Liebler (KL) divergence between two probability measures P and Q will be denoted by $D(P||Q)$. For entropies, it will be convenient to write explicitly the distributions, e.g. $H(P_X)$. When dealing with binary random variables we may use the short-hand notation $H(p)$, where it is understood that it refers to the Shannon entropy over a Bernouilli distribution parametrized by p. A similar notation will be used for divergences, e.g., $D(p_1\|p_2)$.

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

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 $\Pr\left\{\mathcal{E}\right\}$. For two sequences of positive numbers, $\{a_n\}$ and $\{b_n\}$, the notation $a_n \doteq b_n$ 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 $[x]_+ \triangleq \max\{0, x\}$.

Guessing Functions and Strategies: A (possibly randomized) guessing strategy is a sequence $\hat{X}_1^\infty \triangleq \{\hat{X}_k(P_X) : k \geq 1\}$, where $\hat{X}_k(P_X) \in \mathcal{X}$, is independent of the realization X but may depend on P_X. In other words, $\hat{X}_1^\infty$ is the list of guesses the attacker will use one after the other when trying to guess X. The corresponding guessing function, $G(X, \hat{X}_1^\infty)$, defined as

$$G(X, \hat{X}_1^\infty) \triangleq \inf\left\{k \geq 1 : \hat{X}_k(P_X) = X\right\}, \qquad (1)$$

represents the number of queries before reaching X. The ρ-th moment of the number of guesses is thus given by $\mathbb{E}[G(X, \hat{X}_1^\infty)^\rho]$, where the expectation is taken over the distribution P_X and the randomness inherent in the guessing strategy $\hat{X}_1^\infty$. The ρ-th moment *guesswork* of a source $X \sim P_X$ is given by

$$\min_{\hat{X}_1^\infty} \mathbb{E}\left[G(X, \hat{X}_1^\infty)^\rho\right], \qquad (2)$$

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.[3] It was shown in [32] 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. [4] The problem of bounding the expected number of guesses was investigated in [33]. Specifically, among other things, it was shown that for any $\rho \geq 0$, and any guessing function $G(\cdot)$,

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

[3] 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.

[4] Note that the optimal strategy $G^*(X)$ is deterministic, therefore the dependence on $\hat{X}_1^\infty$ is dropped in the notation.

On the contrary, the optimal guessing function, satisfies[5]

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

Finally, letting $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a sequence of independent and identically distributed (i.i.d.) random variables over a finite set, and letting $G^*(\mathbf{X})$ denote the optimal guessing function of a realization of $\mathbf{X}$, it was shown that [33, Proposition 5]

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

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

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

Note that the function $E_\rho(P_X)$ 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[6] with the length of the password n, the total number of queries remains the same[7].

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 $\mathbf{X}$, an i.i.d. sequence of length n generated from P_X, be the sequence to be guessed, and let $\{\hat{\mathbf{X}}_k^{(a)} : k \geq 1\}$ be the strategy of agent $a \in \mathcal{A}$, where $\mathcal{A}$ is a, possibly infinite, countable set. Again, we shall be interested in the regime where $|\mathcal{A}|$ grows at least exponentially fast with n, and the goal is to characterize number of queries made in total. We let the permutation $\pi : \mathbb{N}^+ \to \mathcal{A} \times \mathbb{N}^+$ denote the ordering in which the queries are received, i.e., $\pi(i) = (a_i, k_i)$ means that the i-th query received is $\hat{X}_{k_i}^{(a_i)}$. Denote by Π the set of all such possible orderings. Under an ordering π, Alice receives the sequence of queries $\pi(\hat{\mathbf{X}}_1^\infty) \triangleq \{\hat{\mathbf{X}}_{k_i}^{(a_i)} : i \geq 1\}$. Note that this permutation allows reordering of guesses of a given agent $a \in \mathcal{A}$ which may be received in any arbitrary order. For some fixed strategies $\{\hat{\mathbf{X}}_k^{(a)} : k \geq 1\}$, the worst ordering in terms of guesswork is thus given by

$$\sup_{\pi\in\Pi} \mathbb{E}\left\{G(\mathbf{X}, \pi(\hat{\mathbf{X}}_1^\infty))^\rho\right\}. \tag{7}$$

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

$$\inf_{\{\hat{X}_k^{(a)}, k\geq 1\} \text{ for } a\in\mathcal{A}} \; \sup_{\pi\in\Pi} \; \mathbb{E}\left\{G(\mathbf{X}, \pi(\hat{\mathbf{X}}_1^\infty))^\rho\right\}. \tag{8}$$

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 $\mathbf{X}_n$ an i.i.d. sequence according to P_X, and $\{\hat{\mathbf{X}}_k^{(t)}, k \geq 1\}$ sequences of guesses which are independent over $a \in \mathcal{A}$, we have the following

$$\begin{aligned}
&\lim_{n\to\infty} \frac{1}{n} \log \left(\inf_{\{\hat{\mathbf{X}}_k^{(t)} : k\geq 1\}} \; \sup_{\pi\in\Pi} \; \mathbb{E}\left\{G(\mathbf{X}_n, \pi(\hat{\mathbf{X}}_1^\infty))^\rho\right\} \right) \\
&\quad = \lim_{n\to\infty} \frac{1}{n} \log \mathbb{E}\left\{G^*(\mathbf{X}_n)^\rho\right\} \\
&\quad = \rho \cdot H_{\frac{1}{1+\rho}}(X).
\end{aligned} \tag{9}$$

Note that guesswork measures the *total number of guesses made by the agents.* Thus it is clear that with full synchronization among the agents this value will not depend on $|\mathcal{A}|$. In a sense, dependence on $|\mathcal{A}|$ 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 $|\mathcal{A}|$ and is similar to the fully synchronous case. The above result and (5) show that *synchronization is not necessary to achieve the asymptotic optimal guessing 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 $\mathbb{E}\left\{G^*(\mathbf{X})\right\}$ upper bounds (8).

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

$$\inf_{\{\hat{X}_k^{(t)}, k\geq 1\} \text{ for } a\in\mathcal{A}} \; \sup_{\pi\in\Pi} \; \mathbb{E}\left\{G(\mathbf{X}, \pi(\hat{\mathbf{X}}_1^\infty))^\rho\right\} \geq \mathbb{E}\left\{G^*(\mathbf{X})\right\}. \tag{10}$$

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, that is $X_1^{(a)}$ is the most likely symbol for all

[5] An improved bound by a factor of 2 was reported in [34].

[6] 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 $\lceil |\mathcal{X}|^n/k \rceil$ agents to guarantee that a password of length n will be found.

[7] Note that in this work we use the total number of queries as the main metric for computational effort, as opposed to e.g. [24] where the average number of guesses *per agent* is characterized.

$a \in \mathcal{A}$, $X_2^{(a)}$ 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 $|\mathcal{A}|$, 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 $|\mathcal{A}|$, even though it cannot grow beyond $|\mathcal{X}|^n$, as every unique password is queried at most once. In particular, if $|\mathcal{A}| = |\mathcal{X}|^n$, then the worst-case analysis achieves its upper-bound. Note that these observations are not only an artifact of the worst-case analysis, but rather a consequence 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 $\{\hat{X}_k^{(a)}, k \geq 1\}$ be an i.i.d. process with respect to $\hat{P}(\cdot)$, for all $a \in \mathcal{A}$. For a given $\rho > 0$, we define the quantity

$$V_\rho(X, \hat{X}_1^\infty) \triangleq \binom{G(X, \hat{X}_1^\infty) + \rho - 1}{\rho}, \tag{11}$$

where $\binom{x}{y}$ is the generalized binomial coefficient defined in terms of the Gamma function $\Gamma(\cdot)$, i.e.

$$\binom{x}{y} = \frac{\Gamma(x+1)}{\Gamma(y+1)\Gamma(x-y+1)}. \tag{12}$$

In particular, $V_1(X, \hat{X}_1^\infty) = G(X, X_1^\infty)$. The motivation for this definition of $V_\rho(X, \hat{X}_1^\infty)$ will be made clear in the proof of Lemma 2, where it allows us to compute a particular infinite sum neatly. Note that for large $G(X, \hat{X}_1^\infty)$ and fixed integer ρ, Stirling's approximation of the binomial coefficient directly gives $V_\rho(X, \hat{X}_1^\infty) \approx G(X, \hat{X}_1^\infty)^\rho/\rho!$, therefore $V_\rho(X, \hat{X}_1^\infty)$ approximates the behavior of the guesswork moment $G(X, \hat{X}_1^\infty)^\rho$, up to some factor.

We are interested in the following optimization problem

$$\mathbb{E}\{V_\rho^*(X, \hat{X}_1^\infty)\} \triangleq \inf_{\hat{P} \in \mathcal{P}} \mathbb{E}\{V_\rho(X, \hat{X}_1^\infty)\}, \tag{13}$$

where $\mathcal{P}$ is the probability simplex and $\{\hat{X}_k : k \geq 1\}$ is generated i.i.d. from $\hat{P}$. We let $\hat{P}_\rho^*$ designate the minimizer. The following Lemma is the main ingredient in proving an achievability and thus Theorem 1.

Lemma 2 For any $\rho \geq 1$,

$$\log \mathbb{E}\{V_\rho^*(X, \hat{X}_1^\infty)\} = \rho \cdot H_{\frac{1}{1+\rho}}(X), \tag{14}$$

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

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

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 [35, 36, 24]. 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 $\hat{P}_X$ yields the exact same list as if done according to P_X, see e.g. [21]. 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

$$\log \mathbb{E}\{G(X, \hat{X}_1^\infty)\} = \log |\mathcal{X}|,$$

which could be much worse than $\log \mathbb{E}\{V_1^*(X, \hat{X}_1^\infty)\} = H_{1/2}(X)$. 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. [37, 38]). 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 of Lemma 2: First, note that given X, $G(X, \hat{X}_1^\infty)$ is a geometric random variable, and for $k \geq 1$,

$$\Pr\{G(X, \hat{X}_1^\infty) = k\} = \sum_{x \in \mathcal{X}} P_X(x)(1 - \hat{P}(x))^{k-1}\hat{P}(x).$$

Then, for any $\rho > 0$, we have

$$\mathbb{E}\{V_\rho(X, \hat{X}_1^\infty)\} = \sum_{m=1}^{\infty} \binom{m+\rho-1}{m-1} \Pr\{G(X, \hat{X}_1^\infty) = m\}$$

$$= \sum_{x \in \mathcal{X}} P_X(x)\hat{P}(x) \sum_{m=1}^{\infty} \binom{m+\rho-1}{m-1} (1 - \hat{P}(x))^{m-1}.$$

In the following, we calculate the second summation term in the r.h.s. of the last equality. This is equivalent to calculating

$$\sum_{m=1}^{\infty} \binom{m+\rho-1}{\rho} y^{m-1}.$$

Note that, using the identity $\Gamma(x+1) = x\Gamma(x)$ recursively, we get that

$$\frac{\Gamma(m+\rho)}{\Gamma(\rho+1)} = (m+\rho-1) \cdot (m+\rho-2) \cdots (\rho+1)$$

$$
\begin{aligned}
&= (-1)^{m-1}(-\rho-1)\cdot(-\rho-2)\cdots(-\rho-m+1)\\
&= (-1)^{m-1}\frac{\Gamma(-\rho)}{\Gamma(-\rho-m+1)}, \qquad (16)
\end{aligned}
$$

which yields $\binom{m+\rho-1}{\rho} = (-1)^{m-1}\binom{-\rho-1}{m-1}$, and together with the change of variable $k = m-1$ we obtain

$$
\begin{aligned}
\sum_{m=1}^{\infty}\binom{m+\rho-1}{m-1}y^{m-1} &= \sum_{k=0}^{\infty}\binom{-\rho-1}{k}(-y)^k \qquad (17)\\
&= (1-y)^{-\rho-1}, \qquad (18)
\end{aligned}
$$

where the last equality follows from the binomial formula. Thus,

$$
\begin{aligned}
\mathbb{E}\{V_\rho(X,\hat{X}_1^\infty)\} &= \sum_{x\in\mathcal{X}} P_X(x)\hat{P}(x)\frac{1}{\hat{P}(x)^{1+\rho}}\\
&= \sum_{x\in\mathcal{X}} \frac{P_X(x)}{\hat{P}(x)^\rho}. \qquad (19)
\end{aligned}
$$

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 λ is a Lagrange multiplier, and thus,

$$
\hat{P}^*(x) = \frac{P_X(x)^{\frac{1}{1+\rho}}}{\sum_{x'\in\mathcal{X}} P_X(x')^{\frac{1}{1+\rho}}}.
$$

On substituting this optimal distribution in (19) we finally get

$$
\mathbb{E}\{V_\rho^*(X,\hat{X}_1^\infty)\} = \sum_{x\in\mathcal{X}}\frac{P_X(x)}{\hat{P}^*(x)^\rho} = \left(\sum_{x\in\mathcal{X}} P_X(x)^{\frac{1}{1+\rho}}\right)^{1+\rho},
$$

as claimed. ■

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

Corollary 1 Let $\mathbf{X}$ be a sequence of length n generated i.i.d. from P_X. Then, we have,

$$
\lim_{n\to\infty}\frac{1}{n}\log\mathbb{E}\{V_\rho^*(\mathbf{X},\hat{\mathbf{X}}_1^\infty)\} = \rho\cdot H_{\frac{1}{1+\rho}}(X). \qquad (20)
$$

Proof: Treating $\mathbf{X}$ as a random vector, a direct application of Lemma 2 yields

$$
\begin{aligned}
\log\mathbb{E}\{V_\rho^*(\mathbf{X},\hat{\mathbf{X}}_1^\infty)\} &= \rho\cdot H_{\frac{1}{1+\rho}}(\mathbf{X})\\
&= \left(\sum_{\mathbf{x}\in\mathcal{X}^n} P_{\mathbf{X}}(\mathbf{x})^{\frac{1}{1+\rho}}\right)^{1+\rho}.
\end{aligned}
$$

The desired result follows by the additivity of the Rényi entropy. ■

Note that when $\boldsymbol{X}$ is generated i.i.d., tilting the marginal distributions and drawing symbols i.i.d., or tilting the entire product distribution result in the same optimal distribution.

Remark 1 We note that the result above can be generalized to passwords $\boldsymbol{X}$ which are generated according to an irreducible stationary Markov Chain. More precisely, let $U = (U_{ab})$ and γ_a, for $a,b\in\mathcal{X}$, be the stochastic matrix and stationary distribution of the Markov chain, respectively, so that

$$
\Pr\{\mathbf{X} = (x_1\ldots x_n)\} = \gamma_{x_1}\prod_{i=1}^{n-1}U_{x_ix_{i+1}} \qquad (21)
$$

Then, it was shown in [11] that

$$
\lim_{n\to\infty}\log\mathbb{E}\{G^*(\boldsymbol{X})^\rho\} = \frac{1}{1+\rho}\log\lambda, \qquad (22)
$$

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

$$
\sum_{a\in\mathcal{X}} l_a = 1, \quad \sum_{a\in\mathcal{X}} l_aW_{ab} = \lambda l_b, \quad \sum_{b\in\mathcal{X}} r_bW_{ab} = \lambda r_a. \qquad (23)
$$

Analogously to the result of Corollary 1, it can be shown that generating guesses $\hat{\mathbf{X}}$ according to a Markov Chain with entries $W_{ab}r_b/(\lambda r_a)$ achieves the asymptotic performance in (21). A proof of this fact is outside the scope of this paper, but follows from steps outlined in [11] along with the proof of Lemma 2.

Remark 2 In the standard guessing problem [33] 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, (or oblivious) attacker [35, 36, 1, 24].

We are now ready to prove Theorem 1.

Proof of Theorem 1: We start by noting that letting $\{\hat{\mathbf{X}}_k^{(t)} : k\geq 1\}$ 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 $\{\mathbf{X}_k^{(t)} : k\geq 1\}$ be an i.i.d. process distributed according to $\hat{P}^*$ for all $t\in T$. Then, it is evident that $\pi(\hat{\mathbf{X}}_1^\infty)$ 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 ρ. Indeed, the larger ρ, 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.

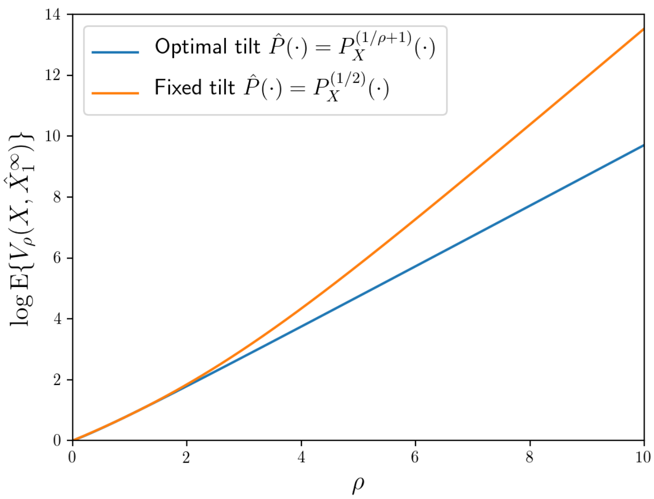


Fig. 3: This plots compares the performance of the randomized strategy as a function of the moment ρ. We compare the optimal strategy which depends on ρ, against a fixed tilted distribution ($\gamma = 1$ in Corollary 2), when $X \sim \mathrm{Ber}(1/5)$.

We do so by tilting P_X by $1/(1+\rho)$. Nevertheless, the optimal distribution, and thus guessing strategy, will change as a function of the guesswork moment ρ 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 ρ. This loss of universality is exploited 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 Fix $\gamma > 0$, and let $\{\hat{X}_k : k \geq 1\}$ be an i.i.d. process generated according to $\hat{P}_\gamma^*(x)$. Then:

$$\log \mathbb{E}\{V_\rho(X, \hat{X}_1^\infty)\} = \frac{\rho}{1+\gamma} H_{\frac{\gamma-\rho+1}{1+\gamma}}(X) + \frac{\gamma \cdot \rho}{1+\gamma} H_{\frac{1}{1+\gamma}}(X) \tag{24}$$

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

Remark 3 (Zipf's distribution) We emphasize that Lemma 2 is a non-asymptotic result. As such, it can be readily used in the context of passwords generated according to a Zipf's law distribution of parameter s for some $s \geq 0$ (also known as PDF-Zipf model [25]), i.e.,

$$P_X(i) \triangleq \frac{1}{H_{m,s}} \cdot \frac{1}{i^s} \tag{25}$$

where $i = 1, \ldots, m$, and $H_{m,s}$ is the *generalized* harmonic number defined as $H_{m,s} = \sum_{j=1}^{m} \frac{1}{j^s}$. As pointed out in the introduction, this family of distribution has been shown in the literature to be useful in modeling password distributions, where the parameter s is dataset dependent. We refer to [25, 26] for more details about the relevance of the Zipf's law in this setting. Under this distribution, applying Lemma 2, we obtain that the optimal i.i.d. guessing strategy is to generate guesses according to a Zipf's law of parameter $s/(\rho+1)$. Further, we get that

$$\log \mathbb{E}\left\{V_\rho^*(X, \hat{X}_1^\infty)\right\} = (1+\rho)\log H_{m,\frac{s}{1+\rho}} - \log H_{m,s}. \tag{26}$$

Note that this is worse than the optimal synchronized strategy which achieves $\log H_{m,(s-\rho)} - \log H_{m,s}$, for $s \geq \rho$, but can perform much better than picking the sub-optimal i.i.d. guessing distribution $P_{\hat{X}} = P_X$, which gives $\log m$. Note that a similar result would hold for the so-called CDF-Zipf's law in [25], i.e., when $P_X(i) = Ci^s - C(i-1)^s$, for some normalizing constant C and parameter $0 \geq s \leq 1$. Namely, it is easy to show that the resulting optimal i.i.d. strategy is then according to the distribution $\hat{P}_\rho^*(i) = C'(i^s - (i-1)^s)^{\frac{1}{1+\rho}}$, where C' is once again a normalizing constant.

Remark 4 (Targeted Attacks) Lemma 2 can also be 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}. This models targeted attacks [29] where an adversary makes use of the additional information he possess about an user (e.g. personal information, previously compromised passwords), as modeled by the side-information Y, to make guesses. Note that, as there are various kinds of side-information Y (e.g., sister password, gender), each of which has a different role in impacting password creation, how to systematically employ such side-information Y is subtle. We refer readers to [29] for a more precise treatment of targeted attacks, and the change in performance that results from them. 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 $G(X, \hat{X}_1^\infty|Y) \triangleq \inf\{k \geq 1 : \hat{X}_k(Y) = X\}$. Then, following the proof of Theorem 2 we can show that the optimal guessing distribution is

$$\hat{P}_{X|Y}^*(x|y) = \frac{P_{X|Y}(x|y)^{\frac{1}{1+\rho}}}{\sum_{x' \in \mathcal{X}} P_{X|Y}(x'|y)^{\frac{1}{1+\rho}}} \tag{27}$$

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

$$\log \mathbb{E}\{V_\rho^*(X, \hat{X}_1^\infty|Y)\} = \rho \cdot H_{\frac{1}{1+\rho}}(X|Y), \tag{28}$$

where $H_\alpha(X|Y)$ is the conditional Rényi entropy of order α, and $V_\rho^*(X, \hat{X}_1^\infty|Y)$ is defined as in (13) but with $G(X, \hat{X}_1^\infty)$ replaced by $G(X, \hat{X}_1^\infty|Y)$. This demonstrates that targeted attacks can also be performed in a distributed way by employing i.i.d. guesses from the distribution $P_{X|Y}(\cdot|Y)$. Note that this assumes that all distributed agents have access to the same side-information Y. A setting in which this does not hold true, i.e. agents may use different side-information Y_i, is outside the scope of this paper, but was studied in [39]. In particular, [39] compares two mechanisms, one in which the agents do not share their side-information and attempt to breach the system independently, and one in which all the side-information is pooled.

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 denoted by J. The goal of the adversary is then to maximize her probability of success within this fixed number of queries, both in the synchronized case [33], as well as the asynchronous case. For synchronous guessers, the probability of success associated with the optimal strategy is given by

$$\mathrm{P}_{c,J}^{\text{synchr}} = \sum_{x \in \mathcal{L}} P_X(x),$$

where $\mathcal{L}$ designates the set of the J most likely elements according to P_X. For asynchronous guessers, one strategy consists in generating guesses $\hat{X}$ i.i.d. from a distribution $P_{\hat{X}}$, as was done in the previous section. This setting was precisely studied in [24, Theorem 6], where the optimal guessing distribution $P_{\hat{X}}$ was characterized as a function of the password distribution P_X and of the number of guesses J. Instead, in this work, we focus on the scenario of guessing n-length i.i.d. sequences, and we assume the adversaries make $J = \lceil \mathcal{X}^{n\alpha} \rceil$ 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 [33] as well as the asynchronous case.

We start with synchronized guessers, and define the exponential rate of $\mathrm{P}_{c,J}^{\text{synchr}}$ as

$$E_{c,\alpha}^{\text{synchr}} \triangleq \liminf_{n\to\infty} -\frac{1}{n} \log \mathrm{P}_{c,J}^{\text{synchr}} \tag{29}$$

$$= \liminf_{n\to\infty} -\frac{1}{n} \log \sum_{\mathbf{x} \in \mathcal{L}} P_{\mathbf{X}}(\mathbf{x}), \tag{30}$$

where again $\mathcal{L}$ represents the set of the J most likely elements distributed according this time to the product distribution $P_{\mathbf{X}}$. The following result is an immediate application of the large deviation principle of Guesswork, shown in [15].

Theorem 2 (Theorem 3 in [15]) For any $\alpha \in [0,1]$,

$$E_{c,\alpha}^{\text{synchr}} = \min_{Q_X \in \mathcal{Q}(\alpha)} D(Q_X \| P_X), \tag{31}$$

where $\mathcal{Q}(\alpha)$ is defined as:

$$\mathcal{Q}(\alpha) = \left\{ Q_X : D(Q_X \| P_X) + H(Q_X) < D(Q_X^* \| P_X) + H(Q_X^*) \right\}, \tag{32}$$

with Q_X^* being the solution of the optimization problem:

$$\begin{aligned} \underset{Q_X}{\text{minimize}} \quad & D(Q_X \| P_X) + H(Q_X) \\ \text{subject to} \quad & H(Q_X) \geq \alpha \end{aligned} \tag{33}$$

In particular, if $\alpha > H(P_X)$, then $E_{c,\alpha}^{\text{synchr}} = 0$.

Note that the average number of guesses, roughly $2^{nH_{1/2}(X)}$, is much larger than the required list size that drives $\mathrm{P}_{c,J}^{\text{synchr}}$ to one (exponentially). This great difference comes from the way atypical events are treated in each optimization. In the case of guesswork, an exponential price is payed for atypical events, since the number of queries will be exponential. For probability of error however, the scenario is closer to regular source coding in which the impact of atypical events is sub-exponential, meaning that the optimized quantity will necessarily be related to the typical events. Consider now the asynchronous case, and let $\{\hat{X}_k : k \geq 1\}$ be once again i.i.d. with distribution $P_{\hat{X}}$. In this case the probability of success is defined as

$$\mathrm{P}_{c,J}^{\text{asynchr}} \triangleq \Pr\left\{ G(\mathbf{X}, \hat{\mathbf{X}}_1^\infty) \leq J \right\}. \tag{34}$$

One can verify that

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

Finally we define

$$E_{c,\alpha}^{\text{asynchr}} \triangleq \liminf_{n\to\infty} -\frac{1}{n} \log \mathrm{P}_{c,J}^{\text{asynchr}}. \tag{35}$$

While, in principle, the distribution $P_{\hat{\mathbf{X}}}$ can be optimized to maximize the probability of success, we will assume that this distribution is simply given by the tilted distribution of $P_{\mathbf{X}}$, namely, for some $\beta \geq 0$, and any $\mathbf{x} \in \mathcal{X}^n$,

$$P_{\hat{X}}^{(\beta)}(x) \triangleq \frac{P_X(x)^\beta}{\sum_{x \in \mathcal{X}} P_X(x)^\beta}. \tag{36}$$

We motivate this choice by the results of the previous subsection, which showed that these tilted distributions were optimal in terms of the number of guesses. We have the following result.

Theorem 3 For any $\alpha, \beta \geq 0$,

$$E_{c,\alpha}^{\text{asynchr}}(\beta) = \min_{Q_X \in \mathcal{Q}(\alpha)} \Big\{ D(Q_X \| P_X) + \Big[D(Q_X \| P_{\hat{X}}^{(\beta)}) + H(Q_X) - \alpha \Big]_+ \tag{37}$$

where $[x]_+ \triangleq \max\{x, 0\}$.

Using Theorem 3, we obtain the following immediate result.

Corollary 3

$$\min_{\beta \geq 0} E_{c,\alpha}^{\text{asynchr}} = \min_{Q_X \in \mathcal{Q}(\alpha)} D(Q_X \| P_X) \tag{38}$$

$$= E_{c,\alpha}^{\text{synchr}}. \tag{39}$$

Corollary 3 essentially proves that the tilted family is asymptotically optimal, and that there exist a unique optimal tilt β for each size list $J = \lceil \mathcal{X}^{n\alpha} \rceil$. It follows from this that even though the optimization (31) is over a set of distributions $\mathcal{Q}(\alpha)$, the solution is always a tilted distribution $P_X^{(\beta)}$ for some $\beta \geq 0$ which depends on α.

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

$$\min_{\beta \geq 0} E_{c,\alpha}^{\text{asynchr}} \leq [H(P_X) - \alpha]_+ = 0. \tag{40}$$

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

$$\min_{\beta\geq 0} E_{c,\alpha}^{\text{asynchr}} \geq D(Q_X^*\|P_X), \tag{41}$$

where Q_X^* is the solution of the optimization

$$\begin{aligned} \underset{Q_X}{\text{minimize}} \quad & D(Q_X\|P_X) + H(Q_X) \\ \text{subject to} \quad & H(Q_X) \geq \alpha. \end{aligned} \tag{42}$$

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

$$\min_{\beta\geq 0} E_{c,\alpha}^{\text{asynchr}} \leq D(Q_X^*\|P_X) + \min_{\beta\geq 0} \left[D(Q_X^*\|P_{\hat{X}}^{(\beta)})\right]_+ .$$

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

$$\min_{\beta\geq 0} E_{c,\alpha}^{\text{asynchr}} \leq D(Q_X^*\|P_X). \tag{43}$$

The result follows from combining (41) and (43). ■

We next provide the proofs of Theorems 2 and 3.

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

$$\frac{1}{n}\log \hat{P}_{X^n}(x^n) = -D(\hat{P}_{\mathbf{x}_n}||\bar{p}^\beta) - H(\hat{P}_{\mathbf{x}_n}) \tag{44}$$

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

$$\begin{aligned} \mathrm{P}_{c,J}^{\text{asynchr}} &= \sum_{x^n\in\mathcal{X}^n} P_{X^n}(x^n)\left[1-(1-\hat{P}_{\mathbf{x}_n})^J\right] \\ &= \sum_{x^n\in\mathcal{X}^n} 2^{-n\left(D(\hat{P}_{\mathbf{x}_n}||p)+H(\hat{P}_{\mathbf{x}_n})\right)} \\ &\qquad \times \left[1-(1-2^{-n(D(\hat{P}_{\mathbf{x}_n}||\bar{p}^\beta)+H(\hat{P}_{\mathbf{x}_n}))})^J\right]. \end{aligned}$$

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

$$\begin{aligned} \mathrm{P}_{c,J}^{\text{asynchr}} &= \sum_{q\in\mathcal{Q}_{n,n}} |T(q)|\, 2^{-n(D(q||p)+H(q))} \\ &\qquad \times \left[1-(1-2^{-n(D(q||\bar{p}^\beta)+H(q))})^J\right] \\ &\doteq \sum_{q\in\mathcal{Q}_{n,n}} 2^{nH(q)} 2^{-n(D(q||p)+H(q))} 2^{-n\left[D(q||\bar{p}^\beta)+H(q)-\alpha\right]_+} \\ &\doteq \max_{q\in[0,1]} 2^{-n\left[D(q||p)+\left[D(q||\bar{p}^\beta)+H(q)-\alpha\right]_+\right]} \end{aligned}$$

where the fourth equation follows from the fact that (see, e.g., [40, 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

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

■

Together, Lemma 2 and Corollary 3 imply that i.i.d. guesses can perform optimally, both in terms of the expected number of guesses, and in terms of the probability of success. Note that, analogous to Lemma 2, the optimal distribution in Corollary 3 depends on the parameter α. As a result, asynchronous guessers can perform brute-force attacks as efficiently as synchronized guessers asymptotically, at the expense of universality. Finally, it should be emphasized that the optimality of the tilted distribution is a by-product of the asymptotic treatment. Indeed, the results of [24] show that the optimal distribution in the non-asymptotic regime is not a tilted distribution of P_X, but rather a more involved functional of the password distribution. As such, our result does not follow from [24] in a straightforward way.

Remark 5 (Probability of failure) The above results characterized the probability of success of an adversary. In particular we demonstrated that a list size J which is large enough (i.e., such that $\alpha > H(P_X)$) will have an exponent of success probability equal to 1, both under asynchronous and synchronous attacks. Note that this result can be strengthened by looking at the complementary probability of failure $P_{f,J}^{\text{synchr}}$ and $P_{f,J}^{\text{asynchr}}$. Again, in the i.i.d. setting, using essentially the same tools as for the probability of success, one can show that the exponents of the probability of failure for both synchronous and asynchronous attacks are the same, equal to 1 when $\alpha < H(P_X)$, and decreasing as α grows. Similarly, the optimal guessing distribution for asynchronous guessers is a tilted distribution, where the tilt depends on the size of the list.

Remark 6 (J-Guesswork) We briefly mention J-Guesswork, a related notion of computational security which was introduced in [28] (denoted α-Guesswork). While the usual Guesswork captures the average number of guesses necessary for a system breach, the average J-Guesswork, denoted by $\mathbb{E}[G_J(X)]$, captures the average number of guesses for an adversary which performs no-more than J queries, where J is picked to guarantee a certain probability of success. As such, when $J = \mathcal{X}^n$, the J-Guesswork reduces to $\mathbb{E}[G(X)]$. We can rewrite the average J-Guesswork, as a sum of two terms, i.e.

$$J \times \mathbb{P}(G(X) > J) + \sum_{i=1}^{J} i \cdot P_X(i), \tag{45}$$

where the first term corresponds to the case where the attacker is unsuccessful and stops at J guesses, and the second terms captures his average number of guesses otherwise. In the asymptotic regime where we look at passwords generated from the product distribution $P_{\mathbf{X}}$, and letting $J = \lceil|\mathcal{X}|^{n\alpha}\rceil$, for $\alpha > H(P_X)$, it follows from the remark above that the probability $\mathbb{P}(G(X) > J)$ goes to zero with an exponent $D(P_X^{(\beta)}\|P_X)$ for some unique $\beta \geq 0$, as long as J is large enough (i.e. $\alpha > H(P_X)$). It is then easy to prove that, when

$\alpha > H(P_X)$, the average J-Guesswork takes exponent

$$\lim_{n\to\infty} \frac{1}{n} \log \mathbb{E}[G_\beta(\mathbf{X})] = \max\{\alpha - D(P_X^{(\beta)} \| P_X), H_{1/2}(P_X)\}. \tag{46}$$

When J is too small, i.e. when $\alpha < H(P_X)$, then with high probability $G(X) > J$, and therefore the exponent is dominated by J itself, that is $\lim_{n\to\infty} \frac{1}{n} \log \mathbb{E}[G_\beta(\mathbf{X})] = \alpha$. Note that these result hold true in an asynchronous setting as well. Indeed, picking guesses i.i.d. from a distribution $P_{\hat{\mathbf{X}}}$ such that it is equal to the tilted distribution which achieves the maximum in (46) gives the same exponent of J-Guesswork. Therefore, i.i.d. guesses perform asymptotically optimally with respect to J-Guesswork as well.

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 sufficently 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. [41] or [42]). Generalizations of the standard Guesswork problem to this setting have been studied (see [17]), 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 (Maynooth University) for many fruitful discussions and for providing the Adobe leaked password dataset used to generate Fig. 2.

References

[1] W. Huleihel, S. Salamatian, and M. Médard, “Guessing with limited memory,” in *Information Theory (ISIT), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 2253–2257.

[2] “McAfee Labs Threat Report,” https://www.mcafee.com/ca/resources/reports/rp-quarterly-threats-sept-2017.pdf, 2017.

[3] K. J. Hole, V. Moen, and T. Tjostheim, “Case study: Online banking security,” *IEEE Security & Privacy*, vol. 4, no. 2, pp. 14–20, 2006.

[4] “Bitcoin Wallets under siege from ’Large Collider’ Attack,” http://fortune.com/2017/04/15/bitcoin-collider/.

[5] J. Owens and J. 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.

[6] “’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/.

[7] “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/, 2013.

[8] 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.

[9] E. Arikan and N. Merhav, “Guessing subject to distortion,” *IEEE Trans. on Inf. Theory*, vol. 44, no. 3, pp. 1041–1056, May 1998.

[10] 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.

[11] D. Sullivan and W. G. Sullivan, “Guesswork and entropy,” *IEEE Trans. on Inf. Theory*, vol. 50, no. 3, pp. 525–526, Mar. 2004.

[12] 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.

[13] R. Sundaresan, “Guessing under source uncertainty,” *IEEE Trans. on Inf. Theory*, vol. 53, no. 1, pp. 525–526, Jan. 2007.

[14] 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.

[15] 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.

[16] 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.

[17] ——, “Multi-user guesswork and brute force security,” *IEEE Transactions on Information Theory*, vol. 61, no. 12, pp. 6876–6886, 2015.

[18] A. Bracher, E. Hof, and A. Lapidoth, “Guessing attacks on distributed-storage systems,” *arXiv preprint arXiv:1701.01981*, 2017.

[19] 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.

[20] 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.

[21] A. Beirami, R. Calderbank, M. Christiansen, K. Duffy, and M. Médard, "A characterization of guesswork on swiftly tilting curves," *IEEE Transactions on Information Theory*, pp. 1–1, 2018.

[22] A. Rezaee, A. Beirami, A. Makhdoumi, M. Médard, and K. Duffy, "Guesswork subject to a total entropy budget," in *Communication, Control, and Computing (Allerton), 2017 55th Annual Allerton Conference on*. IEEE, 2017, pp. 1008–1015.

[23] O. Kosut and L. Sankar, "Asymptotics and non-asymptotics for universal fixed-to-variable source coding," *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 3757–3772, June 2017.

[24] S. Boztas, "On rényi entropies and their applications to guessing attacks in cryptography," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 97, no. 12, pp. 2542–2548, 2014.

[25] D. Wang and P. Wang, "On the implications of zipfs law in passwords," in *European Symposium on Research in Computer Security*. Springer, 2016, pp. 111–131.

[26] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipfs law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[27] J. Blocki, B. Harsha, and S. Zhou, "On the economics of offline password cracking," *IEEE Security and Privacy (to appear)*, 2018.

[28] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," in *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 2012, pp. 538–552.

[29] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted online password guessing: An underestimated threat," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. ACM, 2016, pp. 1242–1254.

[30] D. Wang, H. Cheng, P. Wang, J. Yan, and X. Huang, "A security analysis of honeywords." NDSS, 2018.

[31] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse." in *NDSS*, vol. 14, 2014, pp. 23–26.

[32] J. L. Massey, "Guessing and entropy," in *1994 IEEE International Symposium on Information Theory Proceedings*, 1994, p. 204.

[33] E. Arikan, "An inequality on guessing and its application to sequential decoding," *IEEE Trans. on Inf. Theory*, vol. 42, no. 1, pp. 99–105, Jan. 1996.

[34] S. Boztaş, "Comments on: An inequality on guessing and its application to sequential decoding," *IEEE Trans. on Inf. Theory*, vol. 43, no. 6, pp. 2062–2063, Nov. 1997.

[35] ——, "Oblivious distributed guessing," in *2012 IEEE International Symposium on Information Theory Proceedings*, Jul. 2012, pp. 2162–2165.

[36] M. J. Hanawal and R. Sundaresan, "Randomised attacks on passwords," in *DRDO-IISc Programme on Advanced Research in Mathematical Engineering*, Feb. 2010.

[37] L. L. Campbell, "A coding theorem and rényi's entropy," *Information and control*, vol. 8, no. 4, pp. 423–429, 1965.

[38] A. C. Blumer and R. J. McEliece, "The rényi redundancy of generalized huffman codes," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1242–1249, 1988.

[39] S. Salamatian, A. Beirami, A. Cohen, and M. Médard, "Centralized vs decentralized multi-agent guesswork," in *Information Theory (ISIT), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 2258–2262.

[40] A. Somekh-Baruch and N. Merhav, "Achievable error exponents for the private fingerprinting game," *IEEE Trans. on Inf. Theory*, vol. 53, no. 5, pp. 1827–1838, May 2007.

[41] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982. [Online]. Available: http://doi.acm.org/10.1145/357172.357176

[42] R. J. Perlman, "Network layer protocols with byzantine robustness," Ph.D. dissertation, Massachusetts Institute of Technology, 1988.

Appendix A
Additional Lemmas

The following lemma relates the position of a sequence $\mathbf{x}_n$ in the optimal list, with the type of that sequence.

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

$$\mathcal{Q}(\alpha) = \{Q_X : D(Q_X \| P_X) + H(Q_X) < D(Q_X^* \| P_X) + H(Q_X^*)\}\,, \tag{A.1}$$

with Q_X^* being the solution of the optimization problem:

$$\begin{aligned} \underset{Q_X}{\text{minimize}} \quad & D(Q_X \| P_X) + H(Q_X) \\ \text{subject to} \quad & H(Q_X) \geq \alpha \end{aligned} \,. \tag{A.2}$$

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

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

An application of the method of types gives that the left-hand side evaluates to $2^{n \sup_{Q_X \in \mathcal{Q}(\alpha)} H(Q_X)}$, meaning that $\sup_{Q_X \in \mathcal{Q}(\alpha)} H(Q_X) = \alpha$. 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. ■