Super-Quadratic Quantum Speed-Ups and Guessing Many Likely Keys

Timo Glaser, Alexander May, and Julian Nowakowski*

Ruhr-University Bochum, Bochum, Germany
{timo.glaser,alex.may,julian.nowakowski}@rub.de

Abstract. We study the fundamental problem of guessing cryptographic keys, drawn from some non-uniform probability distribution $\mathcal{D}$, as e.g. in LPN, LWE or for passwords. The optimal classical algorithm enumerates keys in decreasing order of likelihood. The optimal quantum algorithm, due to Montanaro (2011), is a sophisticated Grover search.

We give the first tight analysis for Montanaro's algorithm, showing that its runtime is $2^{\mathrm{H}_{2/3}(\mathcal{D})/2}$, where $\mathrm{H}_\alpha(\cdot)$ denotes Renyi entropy with parameter α. Interestingly, this is a direct consequence of an information theoretic result called Arikan's Inequality (1996) – which has so far been missed in the cryptographic community – that tightly bounds the runtime of classical key guessing by $2^{\mathrm{H}_{1/2}(\mathcal{D})}$.

Since $\mathrm{H}_{2/3}(\mathcal{D}) < \mathrm{H}_{1/2}(\mathcal{D})$ for every non-uniform distribution $\mathcal{D}$, we thus obtain a *super-quadratic* quantum speed-up $s > 2$ over classical key guessing. To give some numerical examples, for the binomial distribution used in Kyber, and for a typical password distribution, we obtain quantum speed-up $s > 2.04$. For the n-fold Bernoulli distribution with parameter $p = 0.1$ as in LPN, we obtain $s > 2.27$. For small error LPN with $p = \Theta(n^{-1/2})$ as in Alekhnovich encryption, we even achieve *unbounded* quantum speedup $s = \Omega(n^{1/12})$.

As another main result, we provide the first thorough analysis of guessing in a multi-key setting. Specifically, we consider the task of attacking many keys sampled independently from some distribution $\mathcal{D}$, and aim to guess a fraction of them. For product distributions $\mathcal{D} = \chi^n$, we show that any constant fraction of keys can be guessed within $2^{\mathrm{H}(\mathcal{D})}$ classically and $2^{\mathrm{H}(\mathcal{D})/2}$ quantumly per key, where $\mathrm{H}(\chi)$ denotes Shannon entropy. In contrast, Arikan's Inequality implies that guessing a single key costs $2^{\mathrm{H}_{1/2}(\mathcal{D})}$ classically and $2^{\mathrm{H}_{2/3}(\mathcal{D})/2}$ quantumly. Since $\mathrm{H}(\mathcal{D}) < \mathrm{H}_{2/3}(\mathcal{D}) < \mathrm{H}_{1/2}(\mathcal{D})$, this shows that in a multi-key setting the guessing cost per key is substantially smaller than in a single-key setting, both classically and quantumly.

1 Introduction

Key guessing. The most fundamental problem in cryptanalysis is the *key guessing problem*. Formally, let $k \in \mathcal{K}$ be a cryptographic key sampled from

* This submission is https://eprint.iacr.org/2023/797.

some key space $\mathcal{K}$, and let $\mathbb{1}_k : \mathcal{K} \to \{0,1\}$ be an oracle with

$$\mathbb{1}_k(x) := \begin{cases} 1 & \text{if } x = k, \\ 0 & \text{else.} \end{cases}$$

The key guessing problem asks to find k, given oracle access to $\mathbb{1}_k(\cdot)$.

The optimal *classical* algorithm for key guessing is brute force search, i.e., iterating over the key space $\mathcal{K}$ in uniformly random order until one finds k. On expectation, brute force finds k after $\Theta(|\mathcal{K}|)$ steps. The optimal *quantum* algorithm is Grover search [Gro96], which achieves a square root speed-up over the classical algorithm, thus solving the key guessing problem in time $\Theta(\sqrt{|\mathcal{K}|})$.

Key guessing from a distribution. The situation changes if the attacker knows the probability distribution $\mathcal{D}$ of the key k. In that case, the optimal classical algorithm for key guessing is to query $\mathbb{1}_k(\cdot)$ on keys $x \in \mathcal{K}$ in decreasing order of probability. This approach has expected runtime

$$T_C(\mathcal{D}) := \sum_{i=1}^{|\mathcal{K}|} i \cdot p_i,$$

where p_i is the probability of the i-th likeliest key.

An information theoretic inequality by Arikan [Ari96] shows

$$\frac{2^{\mathrm{H}_{1/2}(\mathcal{D})}}{1 + \log |\mathcal{K}|} \leq T_C(\mathcal{D}) \leq 2^{\mathrm{H}_{1/2}(\mathcal{D})} \tag{1}$$

where $\mathrm{H}_{1/2}(\mathcal{D})$ is the *Rényi entropy* of the distribution $\mathcal{D}$ with parameter $\frac{1}{2}$. Notice, if $\mathcal{D}$ is the uniform distribution over $\mathcal{K}$, then $2^{\mathrm{H}_{1/2}(\mathcal{D})} = |\mathcal{K}|$. However, for non-uniform $\mathcal{D}$, we have $2^{\mathrm{H}_{1/2}(\mathcal{D})} < |\mathcal{K}|$. Hence, for $\mathcal{D}$ different from the uniform distribution, enumerating keys in decreasing order of likelihood yields a non-trivial speed-up.

For instance, suppose that $\mathcal{D}$ is the n-fold Bernoulli distribution $\mathrm{Ber}(p)^n$ with parameter $p \in (0,1)$. That is, $\mathcal{D}$ samples binary strings from $\{0,1\}^n$ and sets each bit independently to 1 with probability p. Then

$$|\mathcal{K}| = 2^n, \quad \text{and} \quad 2^{\mathrm{H}_{1/2}(\mathcal{D})} = 2^{\mathrm{H}_{1/2}(\mathrm{Ber}(p))n}.$$

As Figure 1 shows, we have $\mathrm{H}_{1/2}(\mathrm{Ber}(p)) < 1$ for every $p \neq \frac{1}{2}$. Hence, for every constant $p \neq \frac{1}{2}$, enumerating keys in decreasing order of likelihood improves over brute-force by an exponential factor

$$\frac{|\mathcal{K}|}{T_C(\mathcal{D})} = \widetilde{\Theta}\left(\frac{2^n}{2^{\mathrm{H}_{1/2}(\mathrm{Ber}(p))n}}\right) = 2^{\Omega(n)}.$$

On the other hand, if $p = \frac{1}{2}$, then $\mathrm{Ber}(p)^n$ is the uniform distribution over $\{0,1\}^n$, and we do not improve over brute force.

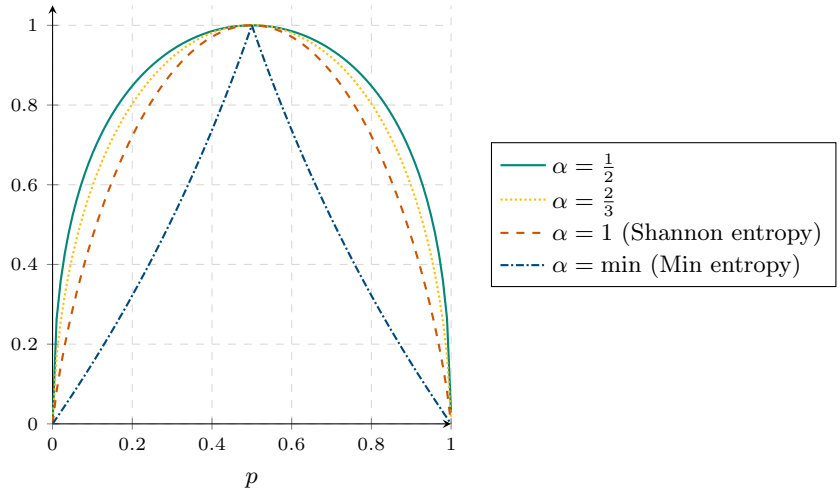


Fig. 1: Rènyi entropy $\mathrm{H}_\alpha(\mathrm{Ber}(p))$ of the Bernoulli distribution $\mathrm{Ber}(p)$ for various α.

Somewhat curiously, Arikan's result seems to be widely unknown in the cryptographic community. Consider, for instance, the recent line of research on *hybrid attacks* on lattice-based crypto [IDF22, DP23, AS22, BM23]. Hybrid attacks try to recover a part of the secret key via key guessing, and the remaining part via lattice-based techniques. The cryptographic literature contains the following results for key guessing in hybrid attacks:

- Bernstein [Ber23] uses the loose upper bound $T_C(\mathcal{D}) \leq |\mathcal{K}|$ in the analysis of his hybrid attack. Since for any $\mathcal{D}$ different from the uniform distribution we have $2^{\mathrm{H}_{1/2}(\mathcal{D})} < |\mathcal{K}|$, Arikan's result shows that Bernstein *overestimates* the runtime of key guessing.
- The analysis of MATZOV [IDF22] claims (without proof) that $T_C(\mathcal{D}) = 2^{\mathrm{H}(\mathcal{D})}$, with $\mathrm{H}(\mathcal{D})$ the *Shannon entropy* of $\mathcal{D}$. Since for non-uniform $\mathcal{D}$ we have $2^{\mathrm{H}_{1/2}(\mathcal{D})} > 2^{\mathrm{H}(\mathcal{D})}$, it follows from Arikan's result that MATZOV *underestimate* the complexity of key guessing.
- Independently of Arikan's result, MATZOV's inaccuracy has been criticized by follow-up works: Ducas-Pulles [DP23] give counterexample distributions $\mathcal{D}$ for which MATZOV's claim is wrong. Albrecht-Shen [AS22] try to improve MATZOV's analysis by computing an upper bound on $T_C(\mathcal{D})$ with discrete Gaussian $\mathcal{D}$. Budroni-Mårtensson [BM23] give an efficient algorithm that, for certain distributions $\mathcal{D}$, can compute $T_C(\mathcal{D})$ numerically.

Quantum key guessing from a distribution. For the quantum setting, Montanaro [Mon11] gave an optimal algorithm that, given a distribution $\mathcal{D}$,

solves the key guessing problem in time

$$T_Q(\mathcal{D}) := \sum_{i=1}^{|\mathcal{K}|} \sqrt{i} \cdot p_i.$$

Let $T_C(\mathcal{D}) = T_Q(\mathcal{D})^s$ for some *quantum speed-up* s. For any distribution $\mathcal{D}$, we have

$$T_Q(\mathcal{D}) \leq \sqrt{T_C(\mathcal{D})}. \tag{2}$$

Hence, Montanaro's quantum algorithm obtains a quantum speed-up over the optimal classical algorithm of at least

$$s = \frac{\log T_C(\mathcal{D})}{\log T_Q(\mathcal{D})} \geq 2.$$

We call a speed-up with $s = 2$ *quadratic*, $s > 2$ *super-quadratic*, $s = \Theta(1)$ *polynomial*, $s = \omega(1)$ *super-polynomial*, and with $s = \Theta(n)$ *exponential*.

Montanaro showed that there exist probability distributions for which Equation (2) is strict, i.e., $T_Q(\mathcal{D}) < \sqrt{T_C(\mathcal{D})}$, achieving a super-quadratic speed-up. However, these distributions are rather artificial. For general distributions, the impact of Montanaro's algorithm seemed unclear so far. Especially, for distributions of cryptographic interest no super-quadratic speed-up $s > 2$ is known.

1.1 Our Contributions

Tight Analysis of Montanaro's Quantum Algorithm. We observe that Arikan's inequality [Ari96] is sufficiently powerful to not only cover classical but also quantum key guessing in the case of distributions $\mathcal{D}$ with finite support $\mathcal{K}$. Namely, a direct application of [Ari96] yields

$$\frac{2^{\frac{\mathrm{H}_{2/3}(\mathcal{D})}{2}}}{\sqrt{1 + \log_2 |\mathcal{K}|}} \leq T_Q(\mathcal{D}) \leq 2^{\frac{\mathrm{H}_{2/3}(\mathcal{D})}{2}}. \tag{3}$$

Combining Equation (1) and Equation (3), we obtain quantum speed-up

$$s := \frac{\log T_C(\mathcal{D})}{\log T_Q(\mathcal{D})} \geq 2 \cdot \frac{\mathrm{H}_{1/2}(\mathcal{D}) - \log(1 + \log |\mathcal{K}|)}{\mathrm{H}_{2/3}(\mathcal{D})}.$$

Thus, all distributions $\mathcal{D}$ with $\log |\mathcal{K}| = o\left(\mathrm{H}_{1/2}(\mathcal{D})\right)$, which e.g. holds for any product distribution $\mathcal{D} = \chi^n$ with finite support, admit quantum speed-up

$$s \geq 2 \cdot \frac{\mathrm{H}_{1/2}(\mathcal{D})}{\mathrm{H}_{2/3}(\mathcal{D})}(1 - o(1)).$$

Since $\mathrm{H}_{1/2}(\mathcal{D}) > \mathrm{H}_{2/3}(\mathcal{D})$ for all but the uniform distribution, this leads asymptotically to a *super-quadratic* quantum speed-up.

Super-Quadratic (and Beyond) Quantum Speed-Ups. We study several distributions that are motivated by cryptographic key choices, such as Bernoulli (LPN [BKW00, HB01, Ale03]), Ternary (NTRU [HPS98, HRSS17]), Binomial (Kyber [BDK+18]), and a Discrete Gaussian (LWE [Reg03]). For completeness, we also analyze common distributions that frequently appear in theory such as the Geometric and the Poisson distribution.

We show asymptotically super-quadratic quantum speed-ups for all these distributions, sometimes even *unbounded* speed-ups that grow as a function of n.

Multi-Key Guessing. Typically, a large-scale real-world adversary Eve does not target a *single* cryptographic key k, but rather a plethora m of collected keys $k_1, \ldots, k_m$ at once, with the goal of recovering a fraction cm of all keys for some $0 \leq c \leq 1$. Arikan's Inequality tightly bounds the required time for guessing a single key *with success probability 1* by *Rényi* entropy $\mathrm{H}_{1/2}(\mathcal{D})$ classically and $\mathrm{H}_{2/3}(\mathcal{D})$ quantumly. However, the inequality does not apply in the multi-key setting, where a success probability of roughly c per key suffices.

We show, for any constant c and keys drawn from a product distribution $\mathcal{D} = \chi^n$, that the classical runtime for guessing cm keys is upper bound by the *Shannon* entropy. More precisely, we introduce a novel algorithm that, for every constant $c < \frac{1}{2}$, recovers cm keys with amortized cost per key

$$\widetilde{\mathcal{O}}\left(2^{\mathrm{H}(\mathcal{D})}\right) = \widetilde{\mathcal{O}}\left(2^{\mathrm{H}(\chi)n}\right).$$

Furthermore, for $\frac{1}{2} \leq c < 1$, the runtime increases only by a subexponential factor $2^{\mathcal{O}(\sqrt{n})}$ to $2^{\mathrm{H}(\chi)n+\mathcal{O}(\sqrt{n})}$. The key idea here is to abort key guessing after a certain number of guesses to avoid spending too much on time on unlikely keys.

Scenario	$0 < c \leq 1$	Cost (Classic)	Cost (Quantum)
constant fraction c	$c < \frac{1}{2}$	$\widetilde{\mathcal{O}}(2^{\mathrm{H}(\chi)n})$ (Theorem 3.4)	$\widetilde{\mathcal{O}}(2^{\frac{\mathrm{H}(\chi)n}{2}})$ (Theorem 4.4)
	$\frac{1}{2} \leq c < 1$	$2^{\mathrm{H}(\chi)n+\widetilde{\mathcal{O}}(\sqrt{n})}$ (Theorem 3.4)	$2^{\frac{\mathrm{H}(\chi)n}{2}+\widetilde{\mathcal{O}}(\sqrt{n})}$ (Theorem 4.4)
all-keys, single key ($m = 1$)	$c = 1$	$\widetilde{\mathcal{O}}\left(2^{\mathrm{H}_{1/2}(\chi)n}\right)$ ([Ari96],Theorem 3.2)	$\widetilde{\mathcal{O}}\left(2^{\frac{\mathrm{H}_{2/3}(\chi)n}{2}}\right)$ (Theorem 4.2)

Table 1: Multi-key: guessing cost per key for cm out of m keys from $\mathcal{D} = \chi^n$.

Our algorithm admits for a quantum version with runtimes $\widetilde{\mathcal{O}}\left(2^{\frac{\mathrm{H}(\chi)n}{2}}\right)$ and $2^{\frac{\mathrm{H}(\chi)n}{2}+\mathcal{O}(\sqrt{n})}$, respectively. Since for any non-uniform χ we have $\mathrm{H}(\chi) < \mathrm{H}_{2/3}(\chi) < \mathrm{H}_{1/2}(\chi)$, we thus substantially improve over the runtimes $2^{\mathrm{H}_{1/2}(\chi)n}$ and $2^{\frac{\mathrm{H}_{2/3}(\chi)n}{2}}$ in the single-key setting. Our results are summarized in Table 1.

Organization of the paper. In Section 2 we introduce Arikan's result. Section 3 studies classical key guessing, both in the single-key and in the multi-key setting. In Section 4 we analyze Montanaro's algorithm for the single-key setting and provide our quantum algorithm for the multi-key setting. Section 5 studies cryptographically relevant distributions and their quantum speed-ups for the single-key setting.

2 Preliminaries

For a positive integer n, we set $[n] := \{1, \ldots, n\}$. All logarithms are base-2. For a cryptographic key k sampled from space key space $\mathcal{K}$, we define $\mathbb{1}_k : \mathcal{K} \to \{0, 1\}$ with $\mathbb{1}_k(x) = 1$ if and only if $x = k$. Throughout the paper, all probability distributions are discrete. We write $X \leftarrow \mathcal{D}$ to denote that a random variable X is drawn from some probability distribution $\mathcal{D}$. Expected value and variance of X are denoted by $\mathbb{E}[X]$ and $\mathrm{Var}[X]$, respectively. For a probability distribution $\mathcal{D}$ with support $\mathcal{K}$, its *probablity mass function* $P_{\mathcal{D}}$ is defined as

$$P_{\mathcal{D}} : \mathcal{K} \to (0, 1],\ k \mapsto \Pr_{X \leftarrow \mathcal{D}}[X = k].$$

Definition 2.1 (Entropy). *Let $\mathcal{D}$ be a probability distribution with support $\mathcal{K}$.*

1. *For $\alpha > 0$, $\alpha \neq 1$, the* Rényi entropy *of $\mathcal{D}$ is defined as*

$$\mathrm{H}_\alpha(\mathcal{D}) := \frac{1}{1-\alpha} \log \left(\sum_{x \in \mathcal{K}} P_{\mathcal{D}}(x)^\alpha \right).$$

2. *The* Shannon entropy *of $\mathcal{D}$ is defined as*

$$\mathrm{H}(\mathcal{D}) := \mathop{\mathbb{E}}_{X \leftarrow \mathcal{D}}[-\log(P_{\mathcal{D}}(X))] = -\sum_{x \in \mathcal{K}} P_{\mathcal{D}}(x) \log(P_{\mathcal{D}}(x)).$$

3. *The* min-entropy *of $\mathcal{D}$ is defined as*

$$\mathrm{H}_{\min}(\mathcal{D}) := \min_{x \in \mathcal{K}} -\log(P_{\mathcal{D}}(x)).$$

We extend the definition of $\mathrm{H}_\alpha(\mathcal{D})$ for $\alpha \in \{0, 1, \infty\}$ by setting

$$\mathrm{H}_\alpha(\mathcal{D}) = \lim_{\alpha' \to \alpha} \mathrm{H}_{\alpha'}(\mathcal{D})$$

and find

$$\mathrm{H}_0(\mathcal{D}) = \log|\mathcal{K}|, \quad \mathrm{H}_1(\mathcal{D}) = \mathrm{H}(\mathcal{D}), \quad \mathrm{H}_\infty(\mathcal{D}) = \mathrm{H}_{\min}(\mathcal{D}).$$

Note that for product distributions $\mathcal{D} = \chi^n$, we have $\mathrm{H}_\alpha(\mathcal{D}) = n\,\mathrm{H}_\alpha(\chi)$ for every α.

Lemma 2.2 (Arikan's Inequality [Ari96]). *Let $\mathcal{D}$ be a probability distribution. Let $p_1 \geq p_2 \geq \ldots$ denote the values assumed by the probability mass function $P_{\mathcal{D}}$. Then for every $\rho > 0$, it holds that*

$$\frac{2^{\rho\,\mathrm{H}_{\frac{1}{1+\rho}}(\mathcal{D})}}{(1+\log|\mathcal{K}|)^\rho} \leq \sum_i i^\rho \cdot p_i \leq 2^{\rho\,\mathrm{H}_{\frac{1}{1+\rho}}(\mathcal{D})}.$$

Lemma 2.3 (Berry-Esseen Theorem [Ber41, Ess45]). *Let $X_1, X_2, \ldots$ be a sequence of i.i.d. random variables with $\mathbb{E}[X_i] < \infty$, $0 < \mathrm{Var}[X_i] < \infty$ and $\mathbb{E}[|X_i|^3] < \infty$. Define $\mu := \mathbb{E}[X_i]$, $\sigma^2 := \mathrm{Var}[X_i]$ and $\overline{X_n} := \frac{1}{n}\sum_{i=1}^n X_i$. Then the distribution of $\sqrt{n}(\overline{X_n} - \mu)$ converges to a Gaussian distribution with mean 0 and variance σ^2 at rate $\mathcal{O}(1/\sqrt{n})$. That is, for every $t \in \mathbb{R}$ it holds that*

$$\Pr[\sqrt{n}(\overline{X_n} - \mu) \leq t] = \int_{-\infty}^{t} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \mathrm{d}x \pm \mathcal{O}\left(\frac{1}{\sqrt{n}}\right).$$

Lemma 2.4 (Grover's Algorithm [Gro96, Høy00, BHMT02]). *Let $|\Psi\rangle$ be a uniform superposition over some finite set $\mathcal{K}$, and let $\tau : \mathcal{K} \to \{0,1\}$ be a function, such that $\tau(x) = 1$ for at most one $x \in \mathcal{K}$. Given $|\Psi\rangle$ and oracle access to $\tau(\cdot)$, Grover's algorithm outputs $x \in \mathcal{K}$ with $\tau(x) = 1$, if it exists, and an error symbol $\perp$ otherwise. Grover's algorithm achieves this, using $\lceil \frac{\pi}{4}\sqrt{|\mathcal{K}|}\rceil + 1$ queries to τ.*

3 Classical Key Guessing

3.1 Single-Key Guessing

Definition 3.1 (Single-Key Guessing Problem). *Let $\mathcal{D}$ be a probability distribution with support $\mathcal{K}$. Let $k \leftarrow \mathcal{D}$. The* single-key guessing problem *is defined as follows. Given a description of $\mathcal{D}$ and oracle access to $\mathbb{1}_k(\cdot)$, the goal is to find the key k.*

The optimal strategy for solving the single-key guessing problem with success probability 1 is to enumerate all possible keys in decreasing order of probability, until the correct key is found. This strategy requires access to an efficient algorithm $\textsc{GetKey}_{\mathcal{D}}$, that on input $i \in \{1, \ldots, |\mathcal{K}|\}$ outputs the i-th most likely key. For the common cryptographic setting, where $\mathcal{D}$ is a product distribution $\mathcal{D} = \chi^n$ over some finite set $\mathcal{K} = A^n$, Budroni and Mårtenson [BM23] give an efficient instantiation of such an algorithm, that (after one initial pre-computation phase

Algorithm 1: KeyGuess

Input: Key guessing instance $(\mathcal{D}, \mathbb{1}_k)$, access to algorithm $\textsc{GetKey}_{\mathcal{D}}$
Output: Key k

```
i ← 0
repeat
    i ← i + 1
    x ← GetKey_D(i)
until 𝟙_k(x) = 1;
return x
```

running in time $\widetilde{\mathcal{O}}(n^{|A|-1}))$ outputs the i-th key in time $\mathcal{O}(|A| \cdot n)$. Throughout the paper, we assume black box access to $\textsc{GetKey}_{\mathcal{D}}$.

The optimal algorithm for the single-key guessing problem is given in Algorithm 1. As shown by Arikan [Ari96], the Rényi entropy tightly bounds the runtime T_{KG} of KeyGuess (Algorithm 1).

Theorem 3.2 ([Ari96]). *Let $\mathcal{D}$ be a distribution with support $\mathcal{K}$. On input of a single-key guessing instance $(\mathcal{D}, \mathbb{1}_k)$,* KeyGuess *outputs the correct key k in expected time*

$$\frac{2^{\mathrm{H}_{1/2}(\mathcal{D})}}{1 + \log|\mathcal{K}|} \leq \mathbb{E}[T_{\mathsf{KG}}] \leq 2^{\mathrm{H}_{1/2}(\mathcal{D})}.$$

Proof. Let p_i denote the probability of the i-th likeliest key drawn from $\mathcal{D}$. Then $\Pr[T_{\mathsf{KG}} = i] = p_i$, and thus $\mathbb{E}[T_{\mathsf{KG}}] = \sum_i i \cdot p_i$. By applying Arikan's inequality (Lemma 2.2), the desired statement follows. □

3.2 Multi-Key Guessing

The *multi-key guessing problem* asks to find a fraction cm out of m keys. More precisely, the problem is defined as follows.

Definition 3.3 (Multi-Key Guessing Problem). *Let $\mathcal{D}$ be a probability distribution with support $\mathcal{K}$. Let $(k_1, \ldots, k_m) \leftarrow \mathcal{D}^m$. The* multi-key guessing problem *with parameter $0 < c < 1$ is defined as follows. Given a description of $\mathcal{D}$ and oracle access to $\mathbb{1}_{k_1}(\cdot), \ldots, \mathbb{1}_{k_m}(\cdot)$, the goal is to find a tuple $(k'_1, \ldots, k'_m) \in (\mathcal{K} \cup \{\perp\})^m$ such that $k'_i = k_i$ for at least cm keys k_i.*

Before we can give our algorithm for solving the *multi-key* guessing problem, we have to introduce with AbortedKeyGuess (Algorithm 2) another algorithm for the *single-key* guessing problem. As its name suggests, AbortedKeyGuess is a variant of KeyGuess that aborts if it does not find the key k within the first t trials, where t is some runtime parameter.

Using AbortedKeyGuess, we can now introduce our algorithm MultiKeyGuess (Algorithm 3) for solving the multi-key guessing problem. Given a multi-key guessing instance $(\mathcal{D}, \mathbb{1}_{k_1}, \ldots, \mathbb{1}_{k_m})$, MultiKeyGuess runs AbortedKeyGuess repeatedly on each single-key instance $(\mathcal{D}, \mathbb{1}_{k_j})$ with exponentially increasing runtime parameter $t = 2, 4, 8, \ldots$ until it recovers at least cm keys k_j.

Algorithm 2: AbortedKeyGuess

Input: Key guessing instance $(\mathcal{D}, \mathbb{1}_k)$, runtime parameter t, access to algorithm $\textsc{GetKey}_{\mathcal{D}}$

Output: Key k or error symbol $\perp$.

$i \leftarrow 0$
repeat
  $i \leftarrow i + 1$
  $x \leftarrow \textsc{GetKey}_{\mathcal{D}}(i)$
until $\mathbb{1}_k(x) = 1$ or $j > t$;
if $\mathbb{1}_{k_j}(x) = 1$ **then return** x ;
else return $\perp$;

Algorithm 3: MultiKeyGuess

Input: Multi-key guessing instance $(\mathcal{D}, \mathbb{1}_{k_1}, \ldots, \mathbb{1}_{k_m})$, parameter c, access to algorithm $\textsc{GetKey}_{\mathcal{D}}$

Output: $(k'_1, \ldots, k'_m) \in (\mathcal{K} \cup \{\perp\})^m$ with $k'_i = k_i$ for at least cm keys k_i.

$(k'_1, \ldots, k'_m) \leftarrow (\perp, \ldots, \perp)$
$t \leftarrow 1$
repeat
  $t \leftarrow 2t$
  for $j \in [m]$ **do**
    $k'_j \leftarrow \textsc{AbortedKeyGuess}((\mathcal{D}, \mathbb{1}_{k_j}), t)$
  end
until $k'_j \neq \perp$ for at least cm keys k'_j;
return $(k'_1, \ldots, k'_m)$

Our first major result is the following theorem, which shows that for product distributions $\mathcal{D} = \chi^n$ the *Shannon* entropy bounds the runtime of MultiKeyGuess. Compared to the single-key setting (Theorem 3.2), the runtime thus improves by a factor at least

$$\frac{2^{\mathrm{H}(\chi)n}}{2^{\mathrm{H}_{1/2}(\chi)n}}.$$

For every non-uniform χ, this is exponential in n.

Theorem 3.4. *Let $\mathcal{D}$ be a product distribution $\mathcal{D} = \chi^n$, where χ has constant support. Let $(\mathcal{D}, \mathbb{1}_{k_1}, \ldots, \mathbb{1}_{k_m})$ be a multi-key guessing instance with constant parameter $0 < c < 1$.*

1. *If $c < \frac{1}{2}$, then with probability at least $1 - e^{-\Omega(m)}$,* MultiKeyGuess *solves $(\mathcal{D}, \mathbb{1}_{k_1}, \ldots, \mathbb{1}_{k_m})$ with amortized cost per key*

$$T_{\mathsf{MKG}} = \widetilde{\mathcal{O}}\left(2^{\mathrm{H}(\chi)n}\right).$$

2. *For $\frac{1}{2} \leq c < 1$, the amortized cost increases at most by a subexponential factor $2^{\mathcal{O}(\sqrt{n})}$ to $2^{\mathrm{H}(\chi)n + \mathcal{O}(\sqrt{n})}$.*

3.3 Proof for Theorem 3.4

To be able to prove Theorem 3.4, we have to introduce the following novel definition.

Definition 3.5 (Core Set). *Let χ be a probability distribution with support A. The* core set $\mathcal{C}_\chi^{n,\delta}$ *of χ with parameters $n \in \mathbb{N}$ and $\delta \geq 0$ is defined as*

$$\mathcal{C}_\chi^{n,\delta} := \left\{ a \in A^n \mid P_{\chi^n}(a) \geq 2^{-\mathrm{H}(\chi)n - \delta n} \right\}.$$

Note that our core set contains the *typical set*

$$\mathcal{T}_\chi^{n,\delta} := \left\{ a \in A^n \mid 2^{-\mathrm{H}(\chi)n + \delta n} \geq P_{\chi^n}(a) \geq 2^{-\mathrm{H}(\chi)n - \delta n} \right\} \subseteq \mathcal{C}_\chi^{n,\delta},$$

which is often used in information theory, see [CT06, Chapter 3]. An important property of the typical set is that, for every *constant* $\delta > 0$ and $n \to \infty$, the probability mass $P_{\chi^n}(\mathcal{T}_\chi^{n,\delta})$ of the typical set converges to 1. Our core set allows for the following more fine-grained statement.

Lemma 3.6. *Let χ be a probability distribution with finite support. For every $\delta \geq 0$, it holds that*

$$\Pr_{k \leftarrow \chi^n}\left[k \in \mathcal{C}_\chi^{n,\delta}\right] \geq \frac{1}{2} \pm \mathcal{O}\left(n^{-\frac{1}{2}}\right). \tag{4}$$

Furthermore, for every $\varepsilon \in (0,1)$, there exists $\delta = \Theta\left(\sqrt{\frac{\ln(\varepsilon^{-1})}{n}}\right)$ such that

$$\Pr_{k \leftarrow \chi^n}\left[k \in \mathcal{C}_\chi^{n,\delta}\right] \geq 1 - \varepsilon. \tag{5}$$

Proof. Let $k = (k_1, \ldots, k_n) \leftarrow \chi^n$, and let A denote the support of χ. If χ is the uniform distribution on A, then $\mathrm{H}(\chi) = \log(|A|)$, and therefore

$$P_{\chi^n}(a) = |A|^{-n} = 2^{-\mathrm{H}(\chi)n},$$

for every $a \in A^n$. Hence, for uniform χ, every $a \in A^n$ lies in $\mathcal{C}_\chi^{n,\delta}$ for arbitrary $\delta \geq 0$, and we have $\Pr\left[k \in \mathcal{T}_\chi^{n,\delta}\right] = 1$. In particular, both Equations (4) and (5) hold.

It remains to prove Equations (4) and (5) for non-uniform χ. By definition of $\mathcal{C}_\chi^{n,\delta}$ it holds that

$$\Pr\left[k \in \mathcal{C}_\chi^{n,\delta}\right] = \Pr\left[\prod_{i=1}^{n} P(k_i) \geq 2^{-\mathrm{H}(\chi)n - \delta n}\right].$$

Let $X_i := -\log P(X_i)$. We set $\overline{X_n} := \frac{1}{n}\sum_{i=1}^{n} X_i$, and rewrite the above probability as

$$\Pr\left[k \in \mathcal{C}_\chi^{n,\delta}\right] = \Pr\left[-\sum_{i=1}^{n} X_i \geq -\mathrm{H}(\chi)n - \delta n\right] = \Pr\left[\overline{X_n} - \mathrm{H}(\chi) \leq \delta\right].$$

We now make three important observations:

1. By definition of Shannon entropy, $\mathbb{E}[X_i] = \mathrm{H}(\chi) < \infty$.
2. Since χ is not uniform, X_i is not constant and thus $\mathrm{Var}[X_i] > 0$.
3. Since χ has finite support, both $\mathrm{Var}[X_i]$ and $\mathbb{E}[|X_i|^3]$ are finite.

By the Berry-Esseen Theorem (Lemma 2.3), the distribution of $\sqrt{n}(\overline{X_n} - \mathrm{H}(\chi))$ thus converges at rate $\mathcal{O}(1/\sqrt{n})$ to a Gaussian distribution with mean 0 and variance $\sigma^2 := \mathrm{Var}[X_i]$. Hence,

$$\begin{aligned}
\Pr\left[k \in \mathcal{C}_\chi^{n,\delta}\right] &= \Pr\left[\overline{X_n} - \mathrm{H}(\chi) \leq \delta\right] = \Pr\left[\sqrt{n}(\overline{X_n} - \mathrm{H}(\chi)) \leq \delta\sqrt{n}\right] \\
&= \int_{-\infty}^{\delta\sqrt{n}} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \mathrm{d}x \pm \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\
&\geq \int_{-\infty}^{0} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \mathrm{d}x \pm \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \\
&= \frac{1}{2} \pm \mathcal{O}\left(\frac{1}{\sqrt{n}}\right),
\end{aligned}$$

which proves Equation (4).

In order to prove Equation (5), simply apply Hoeffding's inequality to conclude that

$$\Pr\left[k \in \mathcal{C}_\chi^{n,\delta}\right] = \Pr\left[\overline{X_n} - \mathrm{H}(\chi) \leq \delta\right] \geq 1 - \exp\left(-\Theta\left(\delta^2 n\right)\right).$$

□

Lemma 3.7. *Let χ be a probability distribution. For every $\delta \geq 0$, it holds that*

$$|\mathcal{C}_\chi^{n,\delta}| \leq 2^{\mathrm{H}(\chi)n + \delta n}.$$

Proof. Our proof is analogous to the proof of [CT06, Theorem 3.1.2], which gives the same upper bound on the size of the typical set $\mathcal{T}_\chi^{n,\delta}$. By definition of $\mathcal{C}_\chi^{n,\delta}$, we have

$$\begin{aligned}
1 &= \sum_{a \in A^n} P_{\chi^n}(a) \geq \sum_{a \in \mathcal{C}_\chi^{n,\delta}} P_{\chi^n}(a) \geq \sum_{a \in \mathcal{C}_\chi^{n,\delta}} 2^{-\mathrm{H}(\chi)n - \delta n} \\
&= |\mathcal{C}_\chi^{n,\delta}| \cdot 2^{-\mathrm{H}(\chi)n - \delta n}.
\end{aligned}$$

Multiplying the above inequality by $2^{\mathrm{H}(\chi)n + \delta n}$, the lemma follows. □

Intuitively, Lemmas 3.6 and 3.7 together show that the $2^{\mathrm{H}(\chi)n}$ most likely keys make up a large fraction of the probability mass of χ^n. Using this observation, we are now ready to prove Theorem 3.4, which we restate below for better readability.

Theorem 3.4. *Let $\mathcal{D}$ be a product distribution $\mathcal{D} = \chi^n$, where χ has constant support. Let $(\mathcal{D}, \mathbb{1}_{k_1}, \ldots, \mathbb{1}_{k_m})$ be a multi-key guessing instance with constant parameter $0 < c < 1$.*

1. *If $c < \frac{1}{2}$, then with probability at least $1 - e^{-\Omega(m)}$,* MultiKeyGuess *solves $(\mathcal{D}, \mathbb{1}_{k_1}, \ldots, \mathbb{1}_{k_m})$ with amortized cost per key*

$$T_{\mathsf{MKG}} = \widetilde{\mathcal{O}}\left(2^{\mathrm{H}(\chi)n}\right).$$

2. *For $\frac{1}{2} \leq c < 1$, the amortized cost increases at most by a subexponential factor $2^{\mathcal{O}(\sqrt{n})}$ to $2^{\mathrm{H}(\chi)n + \mathcal{O}(\sqrt{n})}$.*

Proof. For $\alpha \in \mathbb{N}$, let

$$\mathcal{M}(\alpha) := \{\textsc{GetKey}_{\mathcal{D}}(i) \mid i \in \{1, 2, \ldots, 2^\alpha\}\}$$

denote the set of the 2^α most likely keys. MultiKeyGuess queries the oracles $\mathbb{1}_{k_1}, \ldots, \mathbb{1}_{k_m}$ on all elements of $\mathcal{M}(1), \mathcal{M}(2), \ldots$ until it reaches some $\mathcal{M}(\alpha_{\max})$ that contains at least cm keys k_i. The runtime of MultiKeyGuess is thus

$$\sum_{\alpha=1}^{\alpha_{\max}} m \cdot 2^\alpha \leq \alpha_{\max} \cdot m \cdot 2^{\alpha_{\max}} = \widetilde{\mathcal{O}}\left(m \cdot 2^{\alpha_{\max}}\right).$$

In other words, the amortized cost per key is $T_{\mathsf{MKG}} = \widetilde{\mathcal{O}}\left(2^{\alpha_{\max}}\right)$. Hence, to prove the theorem, it suffices to prove the following statements:

1. If $c < \frac{1}{2}$ and n is sufficiently large, then

$$\Pr\left[\alpha_{\max} \leq \lceil \mathrm{H}(\chi)n \rceil\right] \geq 1 - e^{-\Omega(m)}. \tag{6}$$

2. If $\frac{1}{2} \leq c < 1$, then

$$\Pr\left[\alpha_{\max} \leq \mathrm{H}(\mathcal{D})n + \Theta(\sqrt{n})\right] \geq 1 - e^{-\Omega(m)}. \tag{7}$$

For $\delta \geq 0$ and $j \in [m]$, let $X_{j,\delta} \in \{0, 1\}$ denote an indicator variable with

$$X_{j,\delta} = 1 \iff k_j \in \mathcal{M}(\lceil \mathrm{H}(\chi)n + \delta n \rceil).$$

Then

$$\Pr\left[\alpha_{\max} \leq \lceil \mathrm{H}(\chi)n + \delta n \rceil\right] = \Pr\left[\sum_{j=1}^{m} X_{j,\delta} \geq cm\right]. \tag{8}$$

From the definition of $\mathcal{C}_\chi^{n,\delta}$ and Lemma 3.7 it follows that $\mathcal{C}_\chi^{n,\delta} \subseteq \mathcal{M}(\lceil \mathrm{H}(\chi)n + \delta n \rceil)$. Hence, for $\mu_\delta := \mathbb{E}[X_{j,\delta}]$, we have

$$\mu_\delta = \Pr[X_{j,\delta} = 1] = \Pr\left[k_j \in \mathcal{M}\left(\alpha_\delta\right)\right] \geq \Pr\left[k_i \in \mathcal{C}_\chi^{n,\delta}\right]. \tag{9}$$

We now prove the result for $c < \frac{1}{2}$, i.e., Equation (6). If $c < \frac{1}{2}$, then there exists some constant $c' \in (c, \frac{1}{2})$. By Equation (9) and Lemma 3.6 (specifically Equation (4)) we have, for sufficiently large n, that

$$\mu_0 \geq \Pr\left[k_i \in \mathcal{C}_\chi^{n,0}\right] \geq c'.$$

Let $\varepsilon := 1 - \frac{c}{c'} \in (0,1)$. Then $cm = (1-\varepsilon)dm \leq (1-\varepsilon)\mu_0 m$. Hence, by Equation (8) and a Chernoff bound, we have

$$\Pr\left[\alpha_{\max} \leq \lceil \mathrm{H}(\chi)n \rceil\right] \geq \Pr\left[\sum_{j=1}^{m} X_{j,\delta} \geq (1-\varepsilon)\mu_0 m\right] \geq 1 - e^{-\Omega(m)},$$

which proves Equation (6).

It remains to prove the result for $c \in [\frac{1}{2}, 1)$, i.e., Equation (7). If $c \in [\frac{1}{2}, 1)$, then there exists some constant $c' \in (c, 1)$. By Equation (9) and Lemma 3.6 (specifically Equation (5)), there exists $\delta = \Theta\left(\sqrt{n}\right)$ such that

$$\mu_\delta \geq c'.$$

Now using the same argument as above, we obtain

$$\Pr\left[\alpha_{\max} \leq \mathrm{H}(\chi)n + \Theta\left(\sqrt{n}\right)\right] \geq 1 - e^{-\Omega(m)},$$

which proves Equation (7). □

4 Quantum Key Guessing

4.1 A Tight Analysis for Montanaro's Single-Key Algorithm

Montanaro [Mon11] proved the following result for the quantum complexity of the single-key guessing problem.

Theorem 4.1 (Propositions 2.1 and 2.4 in [Mon11]). *There is a quantum algorithm that solves single-key guessing instances $(\mathcal{D}, \mathbb{1}_k)$ in expected time at most*

$$e\pi \sum_i \sqrt{i} p_i,$$

where $p_1 \geq p_2 \geq \ldots$ are the values assumed by the probability mass function $P_\mathcal{D}$. Up to constant factors, this is the optimal runtime for solving the single-key guessing problem quantumly.

By Cauchy-Schwarz, the quantum complexity $T_{\mathsf{QKG}} := \sum_i \sqrt{i} p_i$ of single-key guessing is upper bound by

$$T_{\mathsf{QKG}} = \sum_i \sqrt{i} p_i = \sum_i \sqrt{i}\sqrt{p_i}\sqrt{p_i} \leq \sqrt{\sum_i i p_i}\sqrt{\sum_i p_i} = \sqrt{\sum_i i p_i}.$$

Hence, for any distribution $\mathcal{D}$, Montanaro's algorithm achieves *at least* a quadratic speed-up over the optimal classical algorithm KeyGuess, which has runtime $\sum_i p_i$. However, it seemed unclear so far, whether the algorithm achieves a *super-quadratic* square-root for cryptographically relevant distributions. As the following novel theorem shows, this is indeed the case.

Theorem 4.2 (Montanaro Runtime). *Let $\mathcal{D}$ be a distribution $\mathcal{D}$ with support $\mathcal{K}$. On input of a single-key guessing instance $(\mathcal{D}, \mathbb{1}_k)$, Montanaro's algorithm outputs the correct key k in expected time*

$$\frac{2^{\frac{\mathrm{H}_{2/3}(\mathcal{D})}{2}}}{\sqrt{1+\log|\mathcal{K}|}} \leq \mathbb{E}[T_{\mathsf{QKG}}] \leq 2^{\frac{\mathrm{H}_{2/3}(\mathcal{D})}{2}}.$$

Proof. Applying Arikan's inequality (Lemma 2.2) with $\rho = \frac{1}{2}$ to Theorem 4.1 immediately gives the bounds on $\mathbb{E}[T_{\mathsf{QKG}}]$. □

Theorem 4.3 (Super-Quadratic Speed-Up). *Let $\mathcal{D}$ be a distribution $\mathcal{D}$ with support $\mathcal{K}$. On input of a single-key guessing instance $(\mathcal{D}, \mathbb{1}_k)$, Montanaro's algorithm achieves over the optimal classical algorithm quantum speed-up*

$$s \geq 2 \cdot \frac{\mathrm{H}_{1/2}(\mathcal{D}) - \log(1+\log|\mathcal{K}|)}{\mathrm{H}_{2/3}(\mathcal{D})}.$$

For a product distribution $\mathcal{D} = \chi^n$ with finite support we obtain

$$s \geq 2 \cdot \frac{\mathrm{H}_{1/2}(\chi)}{\mathrm{H}_{2/3}(\chi)} \left(1 - \mathcal{O}\left(\frac{\log n}{n}\right)\right).$$

Proof. Let $T_C(\mathcal{D})$ and $T_Q(\mathcal{D})$ be the expected runtimes of the optimal classic KeyGuess algorithm and Montanaro's quantum algorithm, respectively. Using Theorems 3.2 and 4.2, we obtain quantum speed-up

$$s = \frac{\log(T_C(\mathcal{D}))}{\log(T_Q(\mathcal{D}))} \geq 2 \cdot \frac{\mathrm{H}_{1/2}(\mathcal{D}) - \log(1+\log|\mathcal{K}|)}{\mathrm{H}_{2/3}(\mathcal{D})}.$$

For a product distribution $\mathcal{D} = \chi^n$ with finite support we may rewrite the speed-up as

$$s \geq 2 \cdot \frac{n\,\mathrm{H}_{1/2}(\chi) - \log(\mathcal{O}(n))}{n\,\mathrm{H}_{2/3}(\chi)} = 2 \cdot \frac{\mathrm{H}_{1/2}(\chi)}{\mathrm{H}_{2/3}(\chi)} \left(1 - \mathcal{O}\left(\frac{\log n}{n}\right)\right).$$

□

Recall that for all χ different from the uniform distribution, we have $\mathrm{H}_{1/2}(\chi) > \mathrm{H}_{2/3}(\chi)$. Hence, Theorem 4.3 shows that Montanaro's algorithm asymptotically achieves a super-quadratic speed-up for any non-uniform χ.

4.2 Quantum Multi-Key Guessing

In Section 3, we showed that the amortized cost for guessing many keys is $\widetilde{\mathcal{O}}\left(2^{\mathrm{H}(\chi)n}\right)$, and thus significantly below the cost $\widetilde{\mathcal{O}}\left(2^{\mathrm{H}_{\frac{1}{2}}(\chi)n}\right)$ of guessing a single key. In this section, we show a similar speed-up for the quantum setting. To this end, we introduce the algorithm QuantumMultiKeyGuess (Algorithm 4), which is the quantum analogue of MultiKeyGuess (Algorithm 3).

Algorithm 4: QuantumMultiKeyGuess

Input: Multi-key guessing instance $(\mathcal{D}, \mathbb{1}_{k_1}, \ldots, \mathbb{1}_{k_m})$, parameter c, access to algorithm $\textsc{GetKey}_{\mathcal{D}}$

Output: $(k'_1, \ldots, k'_m) \in (\mathcal{K} \cup \{\bot\})^m$ with $k'_i = k_i$ for at least cm keys k_i.

$(k'_1, \ldots, k'_m) \leftarrow (\bot, \ldots, \bot)$
$t \leftarrow 1$
repeat
  $t \leftarrow 2t$
  for $j \in [m]$ **do**
    Initialize superposition $|\Psi\rangle \leftarrow 1/\sqrt{t} \cdot \sum_{i=1}^{t} |i\rangle$.
    Run Grover's algorithm on $|\Psi\rangle$ with oracle $\mathbb{1}_{k_j}(\textsc{GetKey}_{\mathcal{D}}(\cdot))$.
    If Grover's algorithm did not return $\bot$, apply $\textsc{GetKey}_{\mathcal{D}}$ to the result and set k'_j to the resulting key.
  end
until $k'_j \neq \bot$ for at least cm keys k'_j;
return $(k'_1, \ldots, k'_m)$

Recall that our classical algorithm MultiKeyGuess uses AbortedKeyGuess to query the oracles $\mathbb{1}_{k_j}$ on the t most likely keys, for some exponentially increasing parameter t. Analogously, our quantum algorithm runs Grover's algorithm on the t most likely keys. As a a result, we obtain a square-root speed-up over MultiKeyGuess's runtime from Theorem 3.4. In particular, we have the following theorem.

Theorem 4.4. *Let $\mathcal{D}$ be a product distribution $\mathcal{D} = \chi^n$, where χ has constant support. Let $(\mathcal{D}, \mathbb{1}_{k_1}, \ldots, \mathbb{1}_{k_m})$ be a multi-key guessing instance with constant parameter $0 < c < 1$.*

1. *If $c < \frac{1}{2}$, then with probability at least $1 - e^{-\Omega(m)}$, QuantumMultiKeyGuess solves $(\mathcal{D}, \mathbb{1}_{k_1}, \ldots, \mathbb{1}_{k_m})$ with amortized cost per key*

$$T_{\mathsf{QMKG}} = \widetilde{\mathcal{O}}\left(2^{\frac{\mathrm{H}(\chi)n}{2}}\right).$$

2. *For $\frac{1}{2} \leq c < 1$, the amortized cost increases at most by a subexponential factor $2^{\mathcal{O}(\sqrt{n})}$ to $2^{\frac{\mathrm{H}(\chi)n}{2} + \mathcal{O}(\sqrt{n})}$.*

5 Quantum Speed-Ups for Various Distributions

Using Theorem 4.3, we compute the quantum speed-up s of Montanaro's quantum algorithm over classical key guessing for several distributions of interest.

Bernoulli. The Bernoulli distribution $\chi = \mathrm{Ber}(p)$ with $X \sim \mathrm{Ber}(p)$ satisfies

$$\Pr[X = 1] = p \text{ and } \Pr[X = 0] = 1 - p.$$

Consider keys sampled from $\mathcal{D} = \chi^n$ as in LPN [BKW00]. The results of applying Theorem 4.3 are depicted in Figure 2.

Notice that $\mathrm{Ber}(\frac{1}{2})$ is the uniform distribution over $\{0,1\}^n$. For any $p \neq \frac{1}{2}$, we obtain super-quadratic quantum speed-ups s. As a numerical example, we have $s > 2.27$ for $p = 0.1$. Notice that LPN typically uses small constant p [HB01]. LPN-based public key encryption [Ale03], requires a smaller choice of $p = \Theta(n^{-\frac{1}{2}})$. In this small noise LPN regime, we obtain a *super-polynomial* quantum speed-up $s = \Omega(n^{\frac{1}{12}})$.

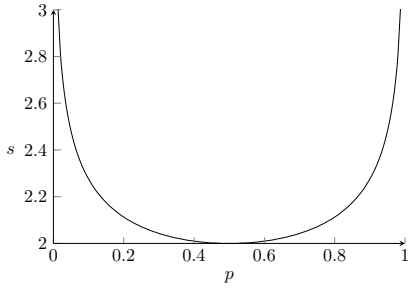


Fig. 2: Bernoulli $\mathrm{Ber}(p)$ Speed-Up

Lemma 5.1 (Small Noise LPN Quantum Speed-Up). *On input of a single-key guessing instance $(Ber(p)^n, \mathbb{1}_k)$ with $p \leq \frac{1}{2}$, Montanaro's algorithm asymptotically achieves a quantum speed-up over classical key guessing* KeyGuess *satisfying*

$$s \geq \frac{1}{3} p^{-\frac{1}{6}}.$$

Thus, for $p = \Theta(n^{-\frac{1}{2}})$ Montanaro's algorithm achieves speed-up $s = \Omega(n^{\frac{1}{12}})$.

Proof. According to Theorem 3.2, the run time of KeyGuess is tightly bounded by $2^{H_{1/2}(\mathrm{Ber}(p))n}$. For $p \leq \frac{1}{2}$ we obtain

$$H_{\frac{1}{2}}(\mathrm{Ber}(p))n = 2\log\left(p^{\frac{1}{2}} + (1-p)^{\frac{1}{2}}\right) n \geq 2\log\left(1 + \frac{p^{\frac{1}{2}}}{2}\right) n \geq \frac{1}{2}\log(e) p^{\frac{1}{2}} n.$$

By Theorem 4.2 the runtime of Montanaro's algorithm is upper bounded by $2^{\frac{1}{2}H_{2/3}(\mathrm{Ber}(p))n}$, where

$$H_{2/3}(\mathrm{Ber}(p))n = 3\log\left(p^{\frac{2}{3}} + (1-p)^{\frac{2}{3}}\right) n \leq 3\log\left(1 + p^{\frac{2}{3}}\right) n \leq 3\log(e) p^{\frac{2}{3}} n.$$

Therefore, we achieve quantum speed-up

$$s = 2 \cdot \frac{H_{\frac{1}{2}}(\mathrm{Ber}(p))n}{H_{\frac{2}{3}}(\mathrm{Ber}(p))n} \geq \frac{1}{3} p^{-\frac{1}{6}}.$$

□

Ternary. We define the Ternary distribution $\chi = \mathcal{T}(p)$, where $X \sim \mathcal{T}(p)$ satisfies

$$\Pr[X = (-1)] = \Pr[X = 1] = \frac{p}{2} \text{ and } \Pr[X = 0] = 1 - p.$$

Consider keys sampled from $\mathcal{D} = \chi^n$ as in NTRU-type schemes [HPS98, HRSS17]. The results of applying our quantum speed-up theorem (Theorem 4.3) are provided in Figure 3.

Notice that for $p = \frac{2}{3}$ we obtain the uniform distribution over $\{-1, 0, 1\}^n$, and for $p = 1$ the uniform distribution over $\{-1, 1\}^n$. For any $p \notin \{\frac{2}{3}, 1\}$, we obtain super-quadratic quantum speed-ups s. As a numerical example, we have $s \approx 2.4$ for $p = 0.1$. A typical NTRU choice is $\mathcal{T}(3/8)$ with $s \approx 2.06$. Again, the speed-up s grows to infinity s for $p \to 0$.

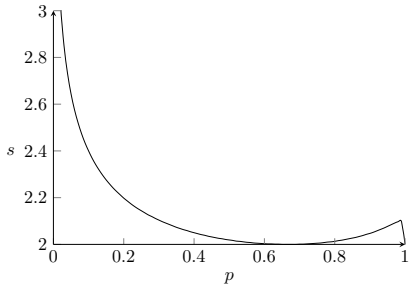


Fig. 3: Ternary $\mathcal{T}(p)$ Speed-Up

Discrete Gaussian. We define a discrete Gaussian distribution $\chi = \mathcal{D}_{\mathcal{S},\sigma}$ with support S and standard deviation σ, where $X \sim \mathcal{D}_{\mathcal{S},\sigma}$ satisfies for all $i \in S$

$$\Pr[X = i] = \frac{1}{c(S)} \cdot e^{\frac{-i^2}{2\sigma^2}} \text{ with } c(S) = \sum_{j \in S} e^{\frac{-j^2}{2\sigma^2}}.$$

Consider keys sampled from $\mathcal{D} = (\mathcal{D}_{\mathcal{S},\sigma})^n$ as in typical LWE schemes [Reg03]. The choice $S = \mathbb{Z}$ results in infinite support for the discrete Gaussian. In order to apply our quantum speed-up theorem (Theorem 4.3) we choose a discrete Gaussian $\chi = \mathcal{D}_{\{-100,\ldots,100\},\sigma}$ with finite support. The results are provided in Figure 4.

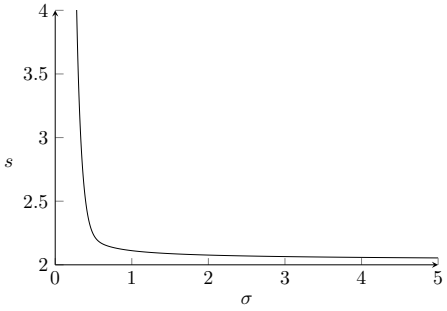


Fig. 4: Gauss $\mathcal{D}_{\{-100,\ldots,100\},\sigma}$ Speed-Up

For $\sigma = 1$, we obtain $s > 2.11$. For large standard deviation σ the quantum speed-up converges to 2, since the discrete Gaussian approaches the uniform distribution. For small σ, the quantum speed-up s becomes arbitrary large.

Binomial. The Binomial distribution $\mathrm{Bin}(m, \frac{1}{2})$ with $X \sim \mathrm{Bin}(m, \frac{1}{2})$ satisfies

$$\Pr[X = i] = \frac{\binom{m}{i}}{2^n} \text{ for } i = 0, \ldots, m.$$

The Binomial distribution, usually centered around 0, is used as an efficient replacement of a discrete Gaussian distribution in LWE-based schemes. The distribution is easy to sample and leads to compact keys with a small support. As an example, Kyber [BDK+18] uses $\mathrm{Bin}(5, \frac{1}{2})$ and $\mathrm{Bin}(7, \frac{1}{2})$. The results of applying Theorem 4.3 to $\mathrm{Bin}(m, \frac{1}{2})$ are shown in Figure 5. For the Kyber choices $m = 5$ and 7 Montanaro's algorithm achieves quantum speed-ups of $s > 2.05$ and $s > 2.06$, respectively.

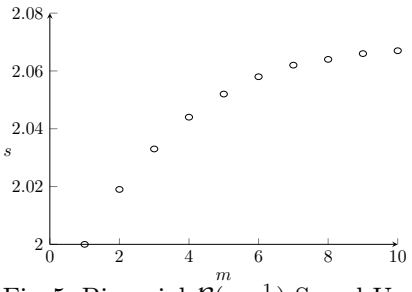


Fig. 5: Binomial $\mathcal{B}(m, \frac{1}{2})$ Speed-Up

For all $m > 1$, one obtains superquadratic speed-ups. For arbitrary large m, we still have $s \approx 2.04$.

Zipf. The Zipf distribution $\chi = \mathcal{Z}(N, t)$ with $X \sim \mathcal{Z}(N, t)$ satisfies for all $i = 1, \ldots, N$

$$\Pr[X = i] = \frac{1}{c(N,t)} \cdot i^{-t} \text{ with } c(N,t) = \sum_{j=1}^{n} j^{-t}.$$

Zipf distributions empirically appear in password database leaks and approximate the observed password distribution. For instance, the well-known LinkedIn database leak with $N = 1.6 \cdot 10^8$ passwords can be modelled via $\mathcal{Z}(N, 0.777)$ [MM12]. Notice that in contrast to all other distributions analyzed in this section, the Zipf distribution is not a product distribution.

Currently, the best known quantum password guessing [DGM+21] achieves a quantum speed-up of (almost) $s = 2$ for any Zipf parameter t. An application of Theorem 4.3 results in Figure 6.

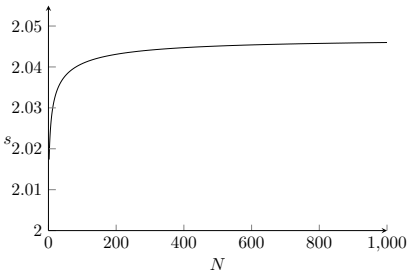
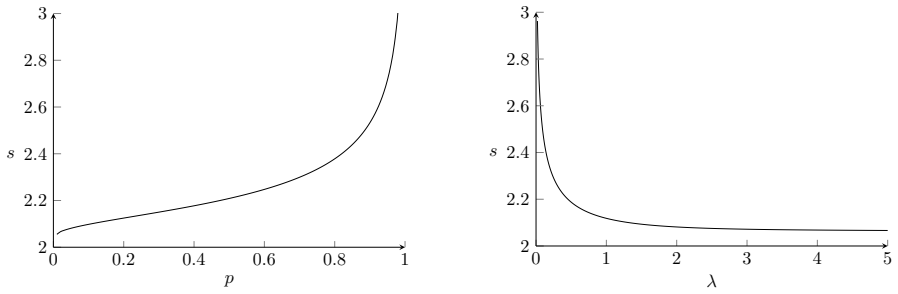


Fig. 6: Zipf $\mathcal{Z}(N, 0.777)$ Speed-Up

For $\mathcal{Z}(N, 0.777)$, our analysis improves over [DGM+21] to quantum speed-up $s > 2.04$ for password guessing.

The following two probability distributions, Geometric and Poisson, have to the best of our knowledge not been used for cryptographic key choices. However, due to their wide range of applications in theoretical computer science quantum speed-ups for these distributions might be of independent interest. In order to apply Theorem 4.3, we define both distributions with finite support.



(a) Geo$(1000, p)$ Speed-Up (b) Poi$(1000, \lambda)$ Speed-Up

Fig. 7: Quantum Speed-ups for Geometric and Poisson distribution

Geometric. The Geometric distribution $\chi = \mathrm{Geo}(N, p)$ with $X \sim \mathrm{Geo}(N, p)$ satisfies for $i = 0, \ldots, N$

$$\Pr[X = i] = \frac{1}{c(N,p)} \cdot (1-p)^i \text{ with } c(N,p) = \sum_{j=0}^{N} (1-p)^j.$$

Let us consider keys sampled from $(\mathrm{Geo}(1000, p))^n$, the quantum speed-ups are depicted in Figure 7a. We observe that we obtain super-quadratic speed-ups for all p. When p converges to 1, our speed-up becomes unbounded.

Poisson. The Poisson distribution $\chi = \mathrm{Poi}(N, \lambda)$ with $X \sim \mathrm{Poi}(N, \lambda)$ satisfies for all $i = 0, \ldots, N$

$$\Pr[X = i] = \frac{1}{c(N,\lambda)} \cdot \frac{\lambda^k}{k!} \text{ with } c(N,\lambda) = \sum_{j=0}^{N} \frac{\lambda^j}{j!}.$$

The quantum speed-ups for $\mathrm{Poi}(1000, \lambda)$ are depicted in Figure 7b, with unbounded speed-up for $\lambda \to 0$.

References

Ale03. Michael Alekhnovich. More on average case vs approximation complexity. In *44th FOCS*, pages 298–307. IEEE Computer Society Press, October 2003.

Ari96. Erdal Arikan. An inequality on guessing and its application to sequential decoding. *IEEE Transactions on Information Theory*, 42(1):99–105, 1996.

AS22. Martin R. Albrecht and Yixin Shen. Quantum augmented dual attack. Cryptology ePrint Archive, Report 2022/656, 2022.

BDK+18. Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

Ber41. Andrew C Berry. The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the american mathematical society*, 49(1):122–136, 1941.

Ber23. Daniel J. Bernstein. Asymptotics of hybrid primal lattice attacks. Cryptology ePrint Archive, Report 2023/1892, 2023.

BHMT02. Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

BKW00. Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *32nd ACM STOC*, pages 435–440. ACM Press, May 2000.

BM23. Alessandro Budroni and Erik Mårtensson. Improved estimation of key enumeration with applications to solving lwe. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 495–500, 2023.

CT06. Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, second edition, 2006.

DGM+21. Markus Dürmuth, Maximilian Golla, Philipp Markert, Alexander May, and Lars Schlieper. Towards quantum large-scale password guessing on real-world distributions. In Mauro Conti, Marc Stevens, and Stephan Krenn, editors, *CANS 21*, volume 13099 of *LNCS*, pages 412–431. Springer, Cham, December 2021.

DP23. Léo Ducas and Ludo N. Pulles. Does the dual-sieve attack on learning with errors even work? In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 37–69. Springer, Cham, August 2023.

Ess45. Carl-Gustav Esseen. Fourier analysis of distribution functions. a mathematical study of the laplace-gaussian law. 1945.

Gro96. Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219. ACM, 1996.

HB01. Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 52–66. Springer, Berlin, Heidelberg, December 2001.

Høy00. Peter Høyer. Arbitrary phases in quantum amplitude amplification. *Physical Review A*, 62(5):052304, 2000.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *LNCS*, pages 267–288. Springer, June 1998.

HRSS17. Andreas Hülsing, Joost Rijneveld, John M. Schanck, and Peter Schwabe. High-speed key encapsulation from NTRU. In Wieland Fischer and Naofumi Homma, editors, *CHES 2017*, volume 10529 of *LNCS*, pages 232–252. Springer, Cham, September 2017.

IDF22. MATZOV IDF. Report on the security of lwe:improved dual lattice attack, 2022. https://zenodo.org/record/6412487#.ZCrT7-xBxqs.

MM12. David Malone and Kevin Maher. Investigating the distribution of password choices. In *Proceedings of the 21st international conference on World Wide Web*, pages 301–310, 2012.

Mon11. Ashley Montanaro. Quantum search with advice. In *Theory of Quantum Computation, Communication, and Cryptography: 5th Conference, TQC*

2010, Leeds, UK, April 13-15, 2010, Revised Selected Papers 5, pages 77–93. Springer, 2011.

Reg03. Oded Regev. New lattice based cryptographic constructions. In *35th ACM STOC*, pages 407–416. ACM Press, June 2003.