

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

**Kaveh Bashiri**

Bundesamt für Sicherheit in der Informationstechnik (BSI)

**Timo Glaser**

Ruhr University Bochum

**Alexander May**

Ruhr University Bochum

**Julian Nowakowski**

Ruhr University Bochum → Centrum Wiskunde & Informatica

# Key Guessing Problem

**Given:** oracle access to some function  $f_k : K \rightarrow \{0,1\}$  with

$$f_k(x) = \begin{cases} 1, & x = k, \\ 0, & \text{else.} \end{cases}$$

**Find:** key  $k$ .

# Key Guessing Problem

**Given:** oracle access to some function  $f_k : K \rightarrow \{0,1\}$  with

$$f_k(x) = \begin{cases} 1, & x = k, \\ 0, & \text{else.} \end{cases}$$

**Find:** key  $k$ .

**Optimal classical algorithm:**

▶ Iterate over key space  $K$  in random order.

▶ Expected runtime  $= \frac{|K| + 1}{2}$ .

**Optimal quantum algorithm:**

▶ Grover search.

▶ Expected runtime  $\approx |K|^{1/2}$ .

# Key Guessing Problem

**Given:** oracle access to some function  $f_k : K \rightarrow \{0,1\}$  with

$$f_k(x) = \begin{cases} 1, & x = k, \\ 0, & \text{else.} \end{cases}$$

**Find:** key  $k$ .

Optimal cla

What if we know the probability distribution  $D$  of  $k$ ?

► Iterate over key space  $K$  in random order.

► Grover search.

► Expected runtime  $= \frac{|K| + 1}{2}$ .

► Expected runtime  $\approx |K|^{1/2}$ .

# Key Guessing Problem

**Given:** oracle access to some function  $f_k : K \rightarrow \{0,1\}$  with

$$f_k(x) = \begin{cases} 1, & x = k, \\ 0, & \text{else.} \end{cases}$$

**Find:** key  $k$ .

Optimal cla

What if we know the probability distribution  $D$  of  $k$ ?

► Iterate over key space  $K$  in random order.

Grover search.

► Expected

**Main result:**

**Super-quadratic** quantum speed-up for every non-uniform  $D$ .

# Key Guessing from a Distribution $D$

**Optimal classical algorithm:**

- ▶ Iterate over key space  $K$  in decreasing order of likelihood.

# Key Guessing from a Distribution $D$

## Optimal classical algorithm:

- ▶ Iterate over key space  $K$  in decreasing order of likelihood.
- ▶ Expected runtime:

$$\sum_{i=1}^{|K|} i \cdot p_i,$$

where  $p_i$  is the probability of the  $i$ -th most likely key.

# Key Guessing from a Distribution $D$

## Optimal classical algorithm:

- ▶ Iterate over key space  $K$  in decreasing order of likelihood.
- ▶ Expected runtime:

$$\sum_{i=1}^{|K|} i \cdot p_i,$$

where  $p_i$  is the probability of the  $i$ -th most likely key.

## Optimal quantum algorithm:

- ▶ **Geometric** Grover Search (Montanaro, TQC'11)
- ▶ Expected runtime:

$$\sum_{i=1}^{|K|} \sqrt{i} \cdot p_i$$

# Key Guessing from a Distribution $D$

## Optimal classical algorithm:

- ▶ Iterate over key space  $K$  in decreasing order of likelihood
- ▶ Expected runtime:

$$\sum_{i=1}^{|K|} i \cdot p_i,$$

where  $p_i$  is the probability of the  $i$ -th most likely key

## Optimal quantum algorithm:

- ▶ **Geometric** Grover Search
- ▶ Expected runtime:

### Improved Estimation of Key Enumeration with Applications to Solving LWE

Alessandro Budroni<sup>1</sup> and Erik Mårtensson<sup>2,3</sup>  
 Cryptography Research Center, Technology Innovation Institute, Abu Dhabi, UAE<sup>1</sup>  
 alessandro.budroni@tii.ae  
 Selmer Center, Department of Informatics, University of Bergen, Norway<sup>2</sup>  
 Department of Electrical and Information Technology, Lund University, Sweden<sup>3</sup>  
 erik.martensson@{uib.no,eit.lth.se}

Asymptotics of hybrid primal lattice attacks

Report on the Security of LWE:  
Improved Dual Lattice Attack

ferent success  
of probability  
range, perhaps

Does the Dual-Sieve Attack on  
Learning with Errors even Work?

TZOV\*†

$\chi_s$ ) guesses,

Quantum Augmented Dual Attack

erlands  
ne Netherlands

be assumed that  
leads to guessing  
 $H(X)$  denotes the  
ion for this claim,

$\mathcal{D}_{\mathbb{Z}^n, \sigma}$ )

# Key Guessing from a Distribution $D$

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 42, NO. 1, JANUARY 1996

99

## An Inequality on Guessing and its Application to Sequential Decoding

Erdal Arikan, *Senior Member, IEEE*

### Improved Estimation of Key Enumeration with Applications to Solving LWE

Alessandro Budroni<sup>1</sup> and Erik Mårtensson<sup>2,3</sup>  
 Cryptography Research Center, Technology Innovation Institute, Abu Dhabi, UAE<sup>1</sup>  
 alessandro.budroni@tii.ae  
 Selmer Center, Department of Informatics, University of Bergen, Norway<sup>2</sup>  
 Department of Electrical and Information Technology, Lund University, Sweden<sup>3</sup>  
 erik.martensson@{uib.no,eit.lth.se}

Netherlands  
 the Netherlands

be assumed that  
 leads to guessing  
 $H(X)$  denotes the  
 ion for this claim,

$\mathcal{D}_{\mathbb{Z}^n, \sigma}$

# Key Guessing from a Distribution $D$

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 42, NO. 1, JANUARY 1996

99

## An Inequality on Guessing and its Application to Sequential Decoding

Erdal Arikan, *Senior Member, IEEE*

$$\sum_{i=1}^{|K|} i \cdot p_i \leq 2^{H_{1/2}(D)}$$

Rényi entropy with parameter 1/2

# Key Guessing from a Distribution $D$

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 42, NO. 1, JANUARY 1996

99

## An Inequality on Guessing and its Application to Sequential Decoding

Erdal Arikan, *Senior Member, IEEE*

$$\frac{2^{H_{1/2}(D)}}{1 + \log |K|} \leq \sum_{i=1}^{|K|} i \cdot p_i \leq 2^{H_{1/2}(D)}$$

Rényi entropy with parameter 1/2

# Rényi Entropy

For a distribution  $D$  with support  $K$  and probabilities  $p_1, \dots, p_{|K|}$ , we define

$$H_\alpha(D) := \frac{1}{1-\alpha} \log \left( \sum_{i=1}^{|K|} p_i^\alpha \right).$$

# Rényi Entropy

For a distribution  $D$  with support  $K$  and probabilities  $p_1, \dots, p_{|K|}$ , we define

$$H_\alpha(D) := \frac{1}{1-\alpha} \log \left( \sum_{i=1}^{|K|} p_i^\alpha \right).$$

**Special cases:**

▶  $H_0(D) = \log(|K|)$

👉 Max entropy

▶  $H_1(D) = - \sum_i p_i \log(p_i)$

👉 Shannon entropy

▶  $H_\infty(D) = \min_i -\log(p_i)$

👉 Min entropy

# Rényi Entropy

For a distribution  $D$  with support  $K$  and probabilities  $p_1, \dots, p_{|K|}$ , we define

$$H_\alpha(D) := \frac{1}{1-\alpha} \log \left( \sum_{i=1}^{|K|} p_i^\alpha \right).$$

**Special cases:**

►  $H_0(D) = \log(|K|)$

👉 Max entropy

►  $H_1(D) = - \sum_i p_i \log(p_i)$

👉 Shannon entropy

►  $H_\infty(D) = \min_i -\log(p_i)$

👉 Min entropy

**Brute force:**  $2^{H_0(D)}$

**Guessing in order of likelihood:**  $2^{H_{1/2}(D)}$

# Rényi Entropy

For a distribution  $D$  with support  $K$  and probabilities  $p_1, \dots, p_{|K|}$ , we define

$$H_\alpha(D) := \frac{1}{1-\alpha} \log \left( \sum_{i=1}^{|K|} p_i^\alpha \right).$$

## Special cases:

▶  $H_0(D) = \log(|K|)$

👉 Max entropy

▶  $H_1(D) = - \sum_i p_i \log(p_i)$

👉 Shannon entropy

▶  $H_\infty(D) = \min_i -\log(p_i)$

👉 Min entropy

**Brute force:**  $2^{H_0(D)}$

**Guessing in order of likelihood:**  $2^{H_{1/2}(D)}$

## Properties:

▶  $D$  uniform

$\implies H_\alpha(D) = H_\beta(D)$ , for all  $\alpha, \beta \in [0, \infty]$

▶  $D$  non-uniform

$\implies H_\alpha(D)$  decreases as  $\alpha$  increases

▶ Product distributions  $D = \chi^n$

$\implies H_\alpha(D) = n \cdot H_\alpha(\chi)$

# Key Guessing from a Distribution $D$

Classical runtime:

$$\sum_{i=1}^{|K|} i \cdot p_i$$

Quantum runtime:

$$\sum_{i=1}^{|K|} \sqrt{i} \cdot p_i$$

# Key Guessing from a Distribution $D$

Classical runtime:

$$\sum_{i=1}^{|K|} i \cdot p_i \approx 2^{H_{1/2}(D)}$$

Quantum runtime:

$$\sum_{i=1}^{|K|} \sqrt{i} \cdot p_i$$

# Key Guessing from a Distribution $D$

Classical runtime:

$$\sum_{i=1}^{|K|} i \cdot p_i \approx 2^{H_{1/2}(D)}$$

Quantum runtime:

$$\sum_{i=1}^{|K|} \sqrt{i} \cdot p_i$$

**Arikan, 1996:**

For every  $\rho \geq 0$ , we have  $\sum_{i=1}^{|K|} i^\rho \cdot p_i \approx 2^{\rho H_{1/(1+\rho)}(D)}$

# Key Guessing from a Distribution $D$

Classical runtime:

$$\sum_{i=1}^{|K|} i \cdot p_i \approx 2^{H_{1/2}(D)}$$


Quantum runtime:

$$\sum_{i=1}^{|K|} \sqrt{i} \cdot p_i \approx 2^{H_{2/3}(D)/2}$$

**Arikan, 1996:**

For every  $\rho \geq 0$ , we have  $\sum_{i=1}^{|K|} i^\rho \cdot p_i \approx 2^{\rho H_{1/(1+\rho)}(D)}$

$\rho = 1/2$



# Key Guessing from a Distribution $D$

Classical runtime:

$$\sum_{i=1}^{|K|} i \cdot p_i \approx 2^{H_{1/2}(D)}$$

Quantum runtime:

$$\sum_{i=1}^{|K|} \sqrt{i} \cdot p_i \approx 2^{H_{2/3}(D)/2}$$

**Arikan, 1996:**

For every  $\rho \geq 0$ , we have  $\sum_{i=1}^{|K|} i^\rho \cdot p_i \approx 2^{\rho H_{1/(1+\rho)}(D)}$

$\rho = 1/2$

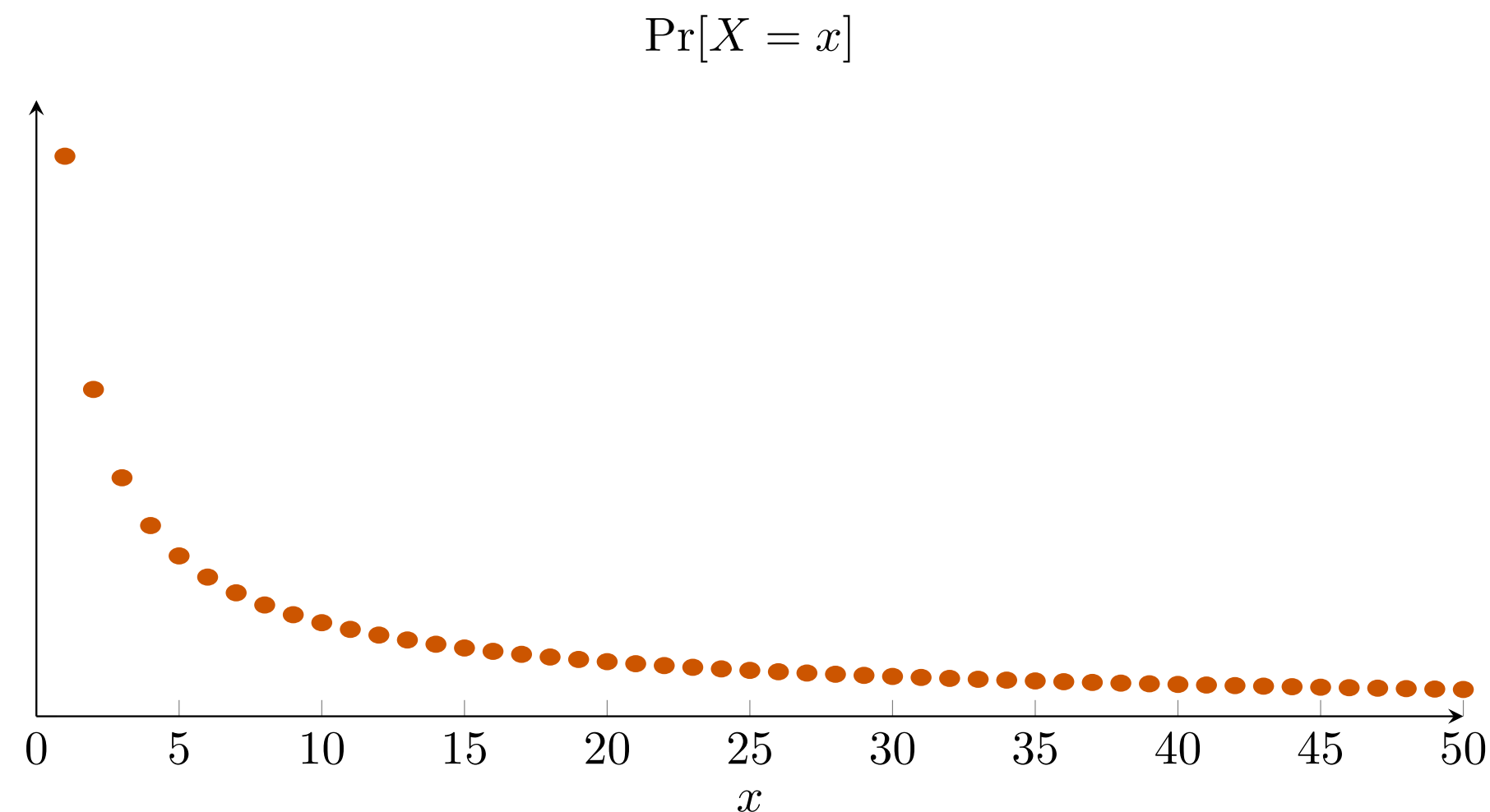
**Quantum speed-up:**  $2^{\frac{H_{1/2}(D)}{H_{2/3}(D)}} > 2$

for every non-uniform  $D$

# Example: Zipf Distribution (Passwords)

## Definition:

- ▶ Fix parameters  $N$  and  $t$ .
- ▶  $\Pr[X = x] \propto \frac{1}{x^t}$  for  $x = 1, \dots, N$ .
- ▶ Leaked LinkedIn passwords (approximately) follow Zipf distribution with  $N = 2^{27.2\dots}$  and  $t = 0.777$ .



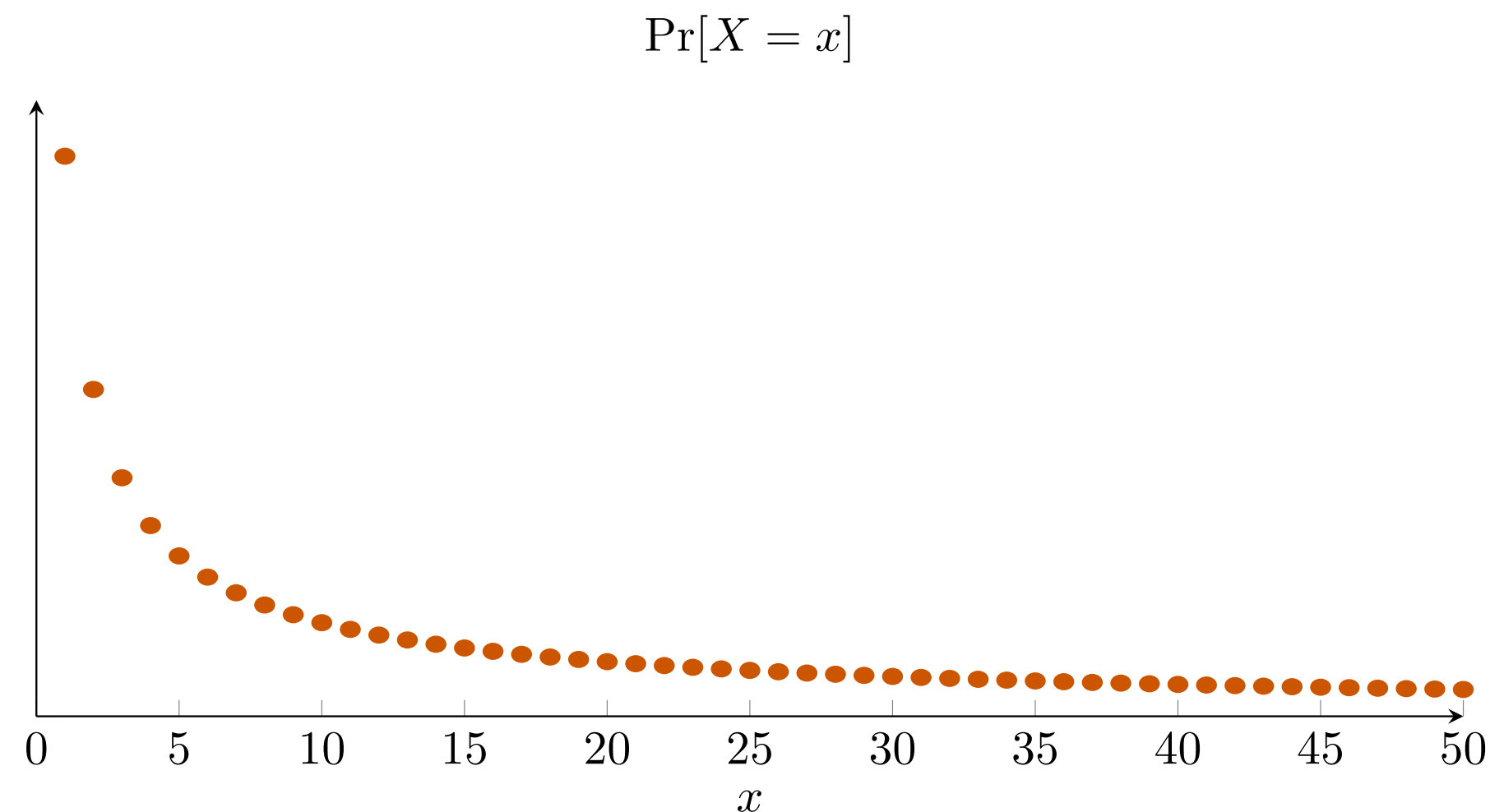
# Example: Zipf Distribution (Passwords)

## Definition:

- ▶ Fix parameters  $N$  and  $t$ .
- ▶  $\Pr[X = x] \propto \frac{1}{x^t}$  for  $x = 1, \dots, N$ .
- ▶ Leaked LinkedIn passwords (approximately) follow Zipf distribution with  $N = 2^{27.2\dots}$  and  $t = 0.777$ .

## Classical guessing:

- ▶  $2^{H_{1/2}(D)} = 2^{26.5\dots}$



# Example: Zipf Distribution (Passwords)

## Definition:

- ▶ Fix parameters  $N$  and  $t$ .
- ▶  $\Pr[X = x] \propto \frac{1}{x^t}$  for  $x = 1, \dots, N$ .
- ▶ Leaked LinkedIn passwords (approximately) follow Zipf distribution with  $N = 2^{27.2\dots}$  and  $t = 0.777$ .

## Classical guessing:

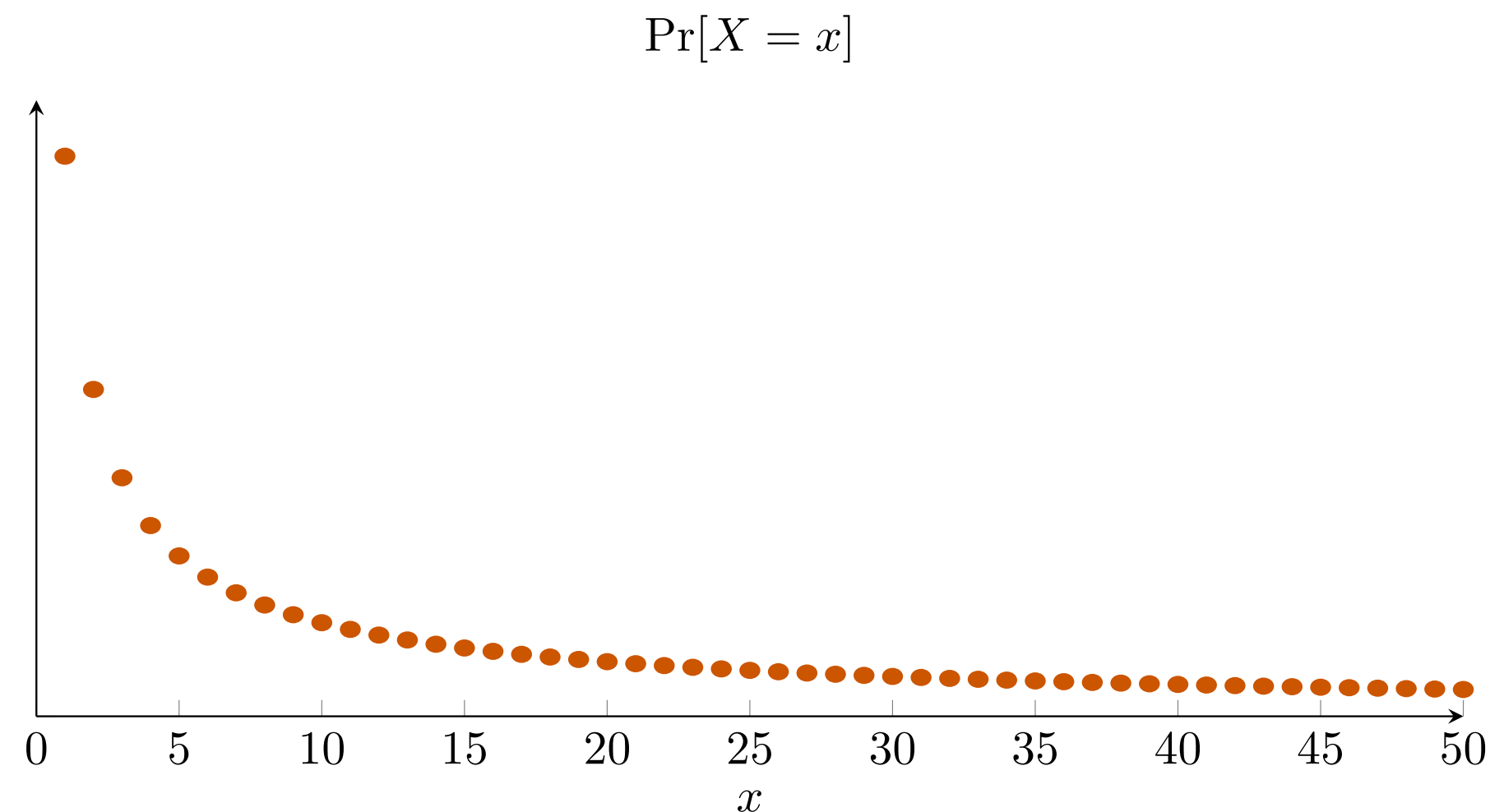
- ▶  $2^{H_{1/2}(D)} = 2^{26.5\dots}$

## Quantum guessing:

- ▶  $2^{H_{2/3}(D)/2} = 2^{13.1\dots}$

## Speed-up:

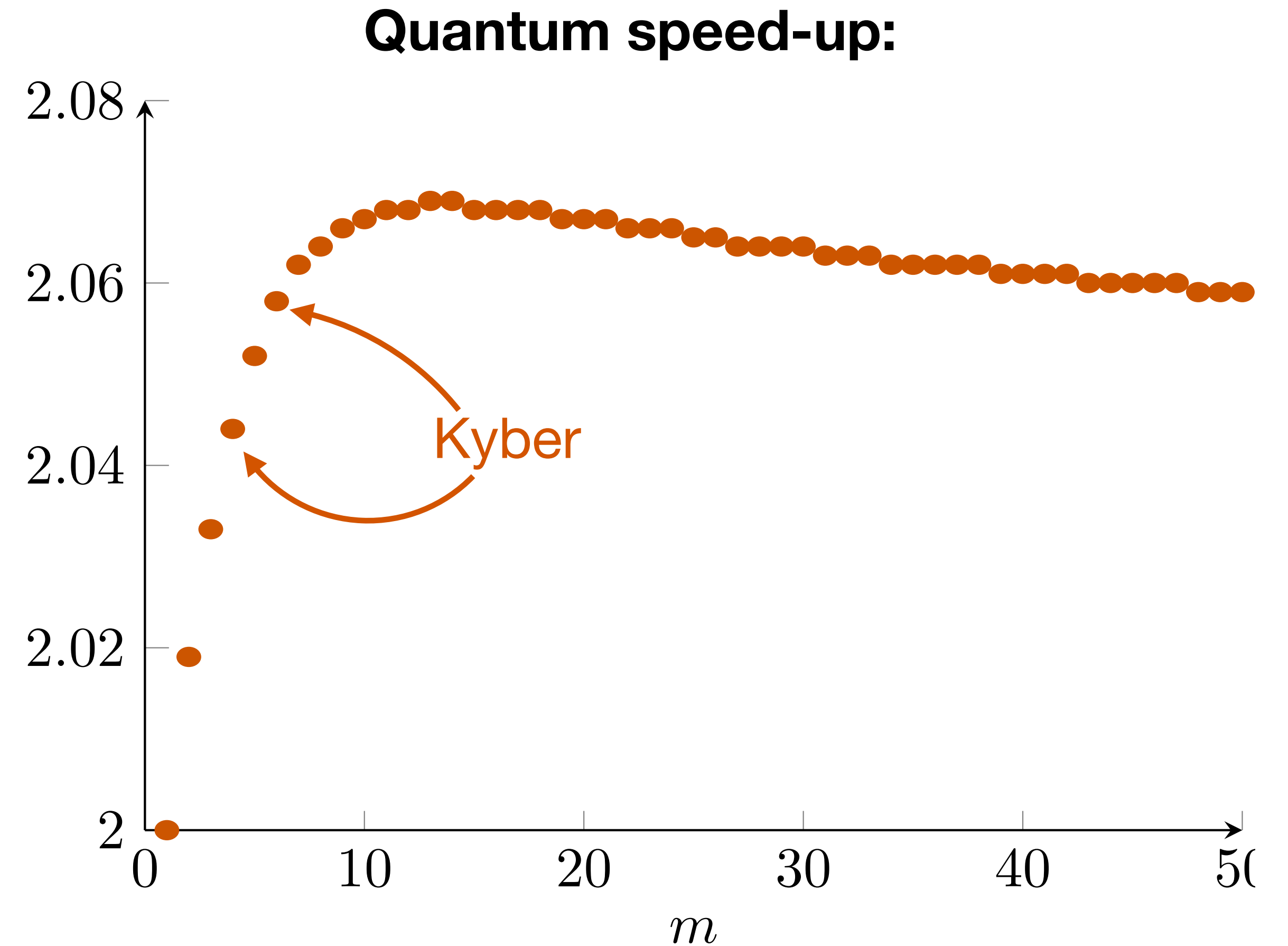
- ▶ 2.03



# Example: Centered Binomial (Kyber)

## Definition:

- ▶ Sample  $(X_1, \dots, X_m) \leftarrow \{0,1\}^m$  uniformly.
- ▶ Output  $X_1 + \dots + X_m - \frac{m}{2}$ .

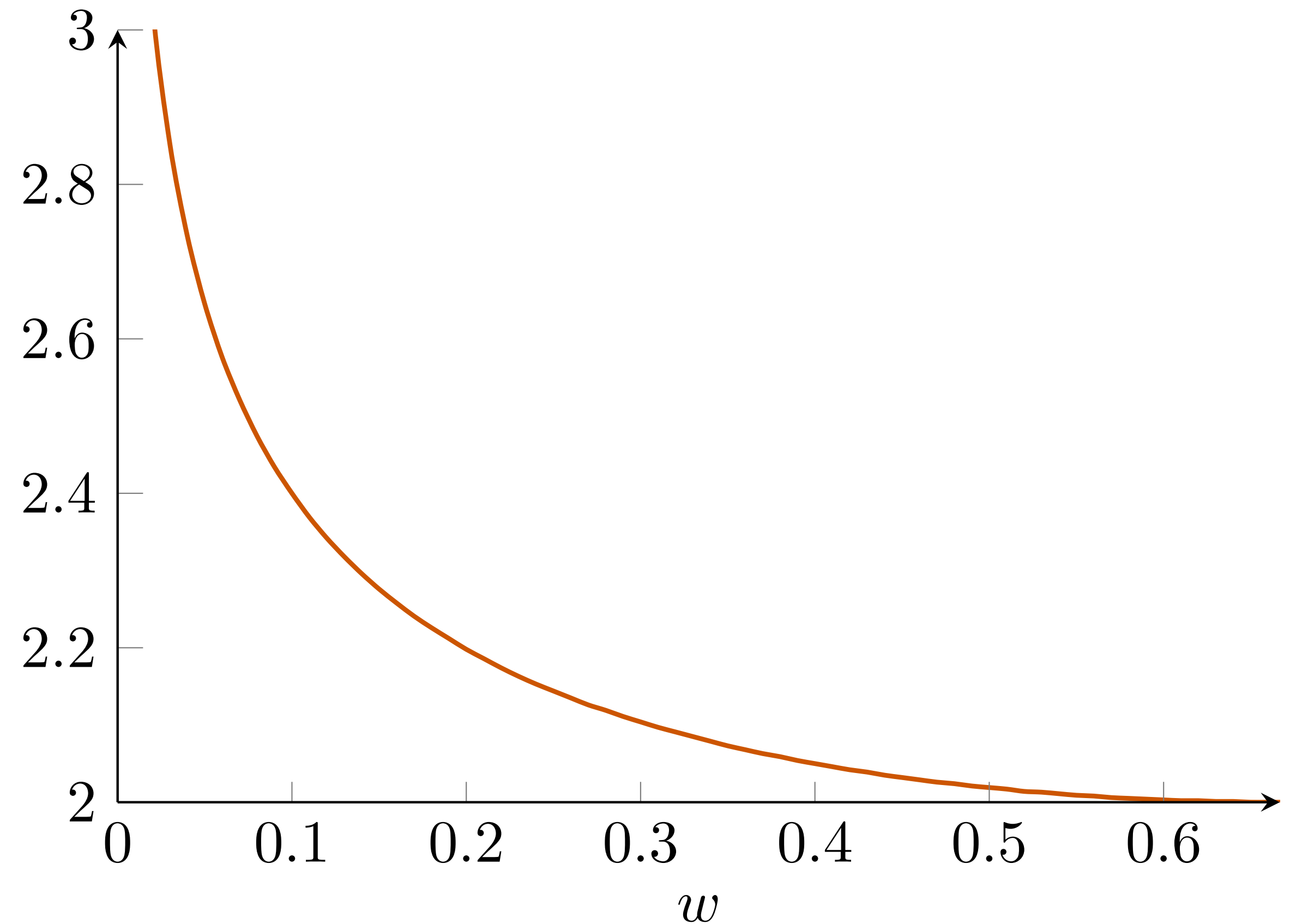


# Example: Weighted Ternary (FHE)

## Definition:

- ▶  $\Pr[X = 1] = w/2$        $\rightarrow w \in (0, 2/3]$
- ▶  $\Pr[X = -1] = w/2$
- ▶  $\Pr[X = 0] = 1 - w$
  
- ▶ Used for sampling ternary  $n$ -dimensional vectors of expected Hamming weight  $wn$ .

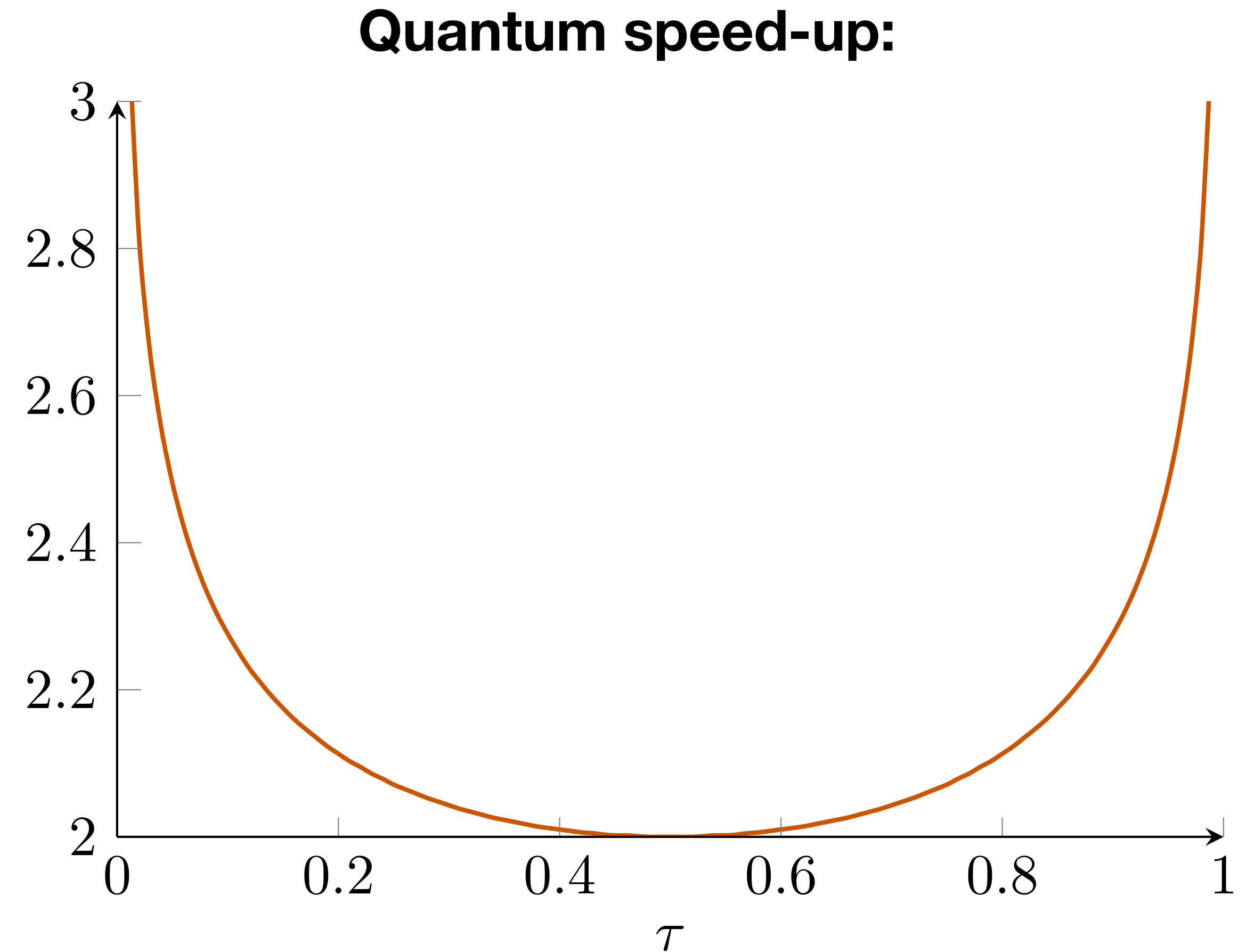
Quantum speed-up:



# Example: Bernoulli Distribution (LPN)

## Definition:

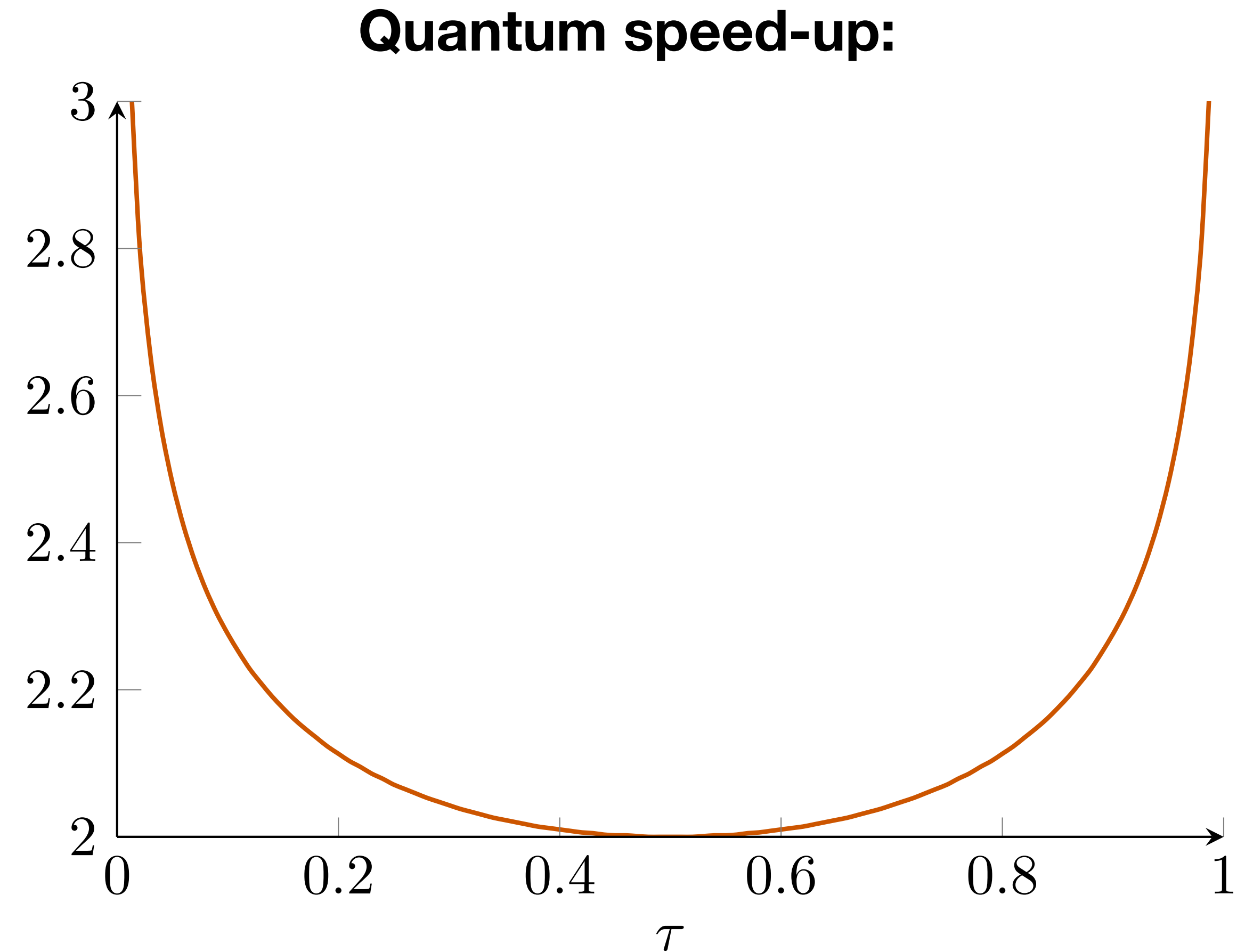
- ▶  $\Pr[X = 1] = \tau$
- ▶  $\Pr[X = 0] = 1 - \tau$



# Example: Bernoulli Distribution (LPN)

## Definition:

- ▶  $\Pr[X = 1] = \tau$
- ▶  $\Pr[X = 0] = 1 - \tau$
  
- ▶ LPN-based encryption uses  $n$ -fold Bernoulli with  $\tau = n^{-1/2}$ .
- ▶ Quantum speed-up:  $\approx n^{1/12}$ .



# Guessing One Out of $m$ Keys

- ▶ Real world protocols typically involve many keys.
- ▶ Successfully guessing only one of them can already be a devastating attack.

# Guessing One Out of $m$ Keys

- ▶ Real world protocols typically involve many keys.
- ▶ Successfully guessing only one of them can already be a devastating attack.
- ▶ We prove the following **upper bounds**:

$$2^{H_{1-1/(m+1)}(D)} \quad \sqrt{2^{H_{1-1/(2m+1)}(D)}}$$

**classically**

**quantumly**

# Guessing One Out of $m$ Keys

- ▶ Real world protocols typically involve many keys.
- ▶ Successfully guessing only one of them can already be a devastating attack.
- ▶ We prove the following **upper bounds**:

$$2^{H_{1-1/(m+1)}(D)} \quad \sqrt{2^{H_{1-1/(2m+1)}(D)}}$$

classically

quantumly

$m$

1

$\infty$

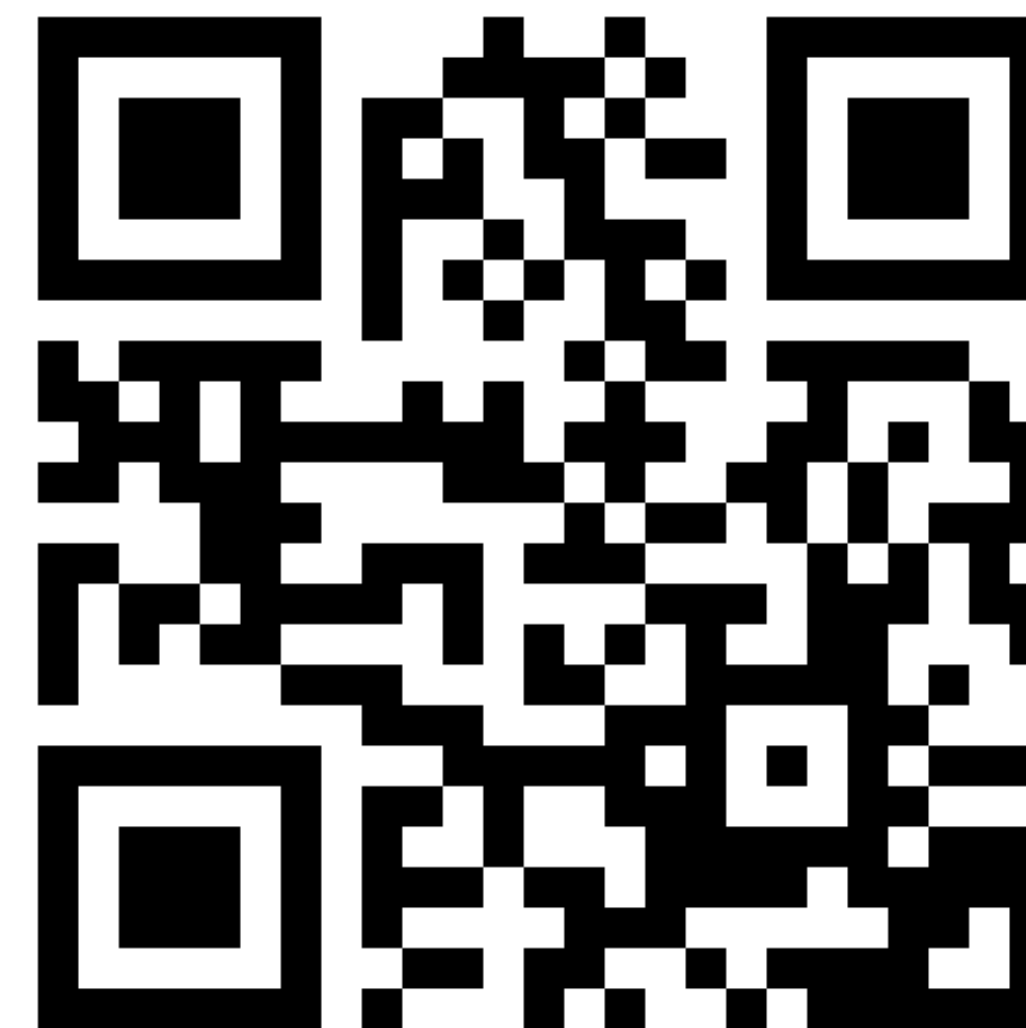
---

classically	$2^{H_{1/2}(D)}$	$2^{H_1(D)}$
-------------	------------------	--------------

quantumly	$\sqrt{2^{H_{2/3}(D)}}$	$\sqrt{2^{H_1(D)}}$
-----------	-------------------------	---------------------

# Summary

	Single Key	One-out-of-m
classically	$H_{1/2}(D)$	$H_{1-\frac{1}{m+1}}(D)$
quantumly	$H_{2/3}(D)/2$	$H_{1-\frac{1}{2m+1}}(D)/2$



<https://ia.cr/2023/797>

## More in the paper:

- ▶ For product distributions, we succeed with probability  $\gtrsim 50\%$  after
  - ▶  $2^{H_1(D)}$  steps classically
  - ▶  $2^{H_1(D)/2}$  steps quantumly.



# Likely Keys

## Example:

- ▶ LPN subkey drawn from  $n$ -fold Bernoulli distribution with  $n = 15$ ,  $\tau = 0.2$ .
- ▶ Let  $T$  be RV for number of required guesses.
- ▶ Arikan:  $\mathbb{E}[T] \approx 2^{H_{1/2}(D)} = 6476.64\dots$

# Likely Keys

## Example:

- ▶ LPN subkey drawn from  $n$ -fold Bernoulli distribution with  $n = 15$ ,  $\tau = 0.2$ .
- ▶ Let  $T$  be RV for number of required guesses.
- ▶ Arikan:  $\mathbb{E}[T] \approx 2^{H_{1/2}(D)} = 6476.64\dots$

Does  $T$  concentrate around  $\mathbb{E}[T]$ ?

- ▶ Markov's inequality:  $T \gg \mathbb{E}[T]$  is unlikely.
- ▶ What about  $T \ll \mathbb{E}[T]$ ?

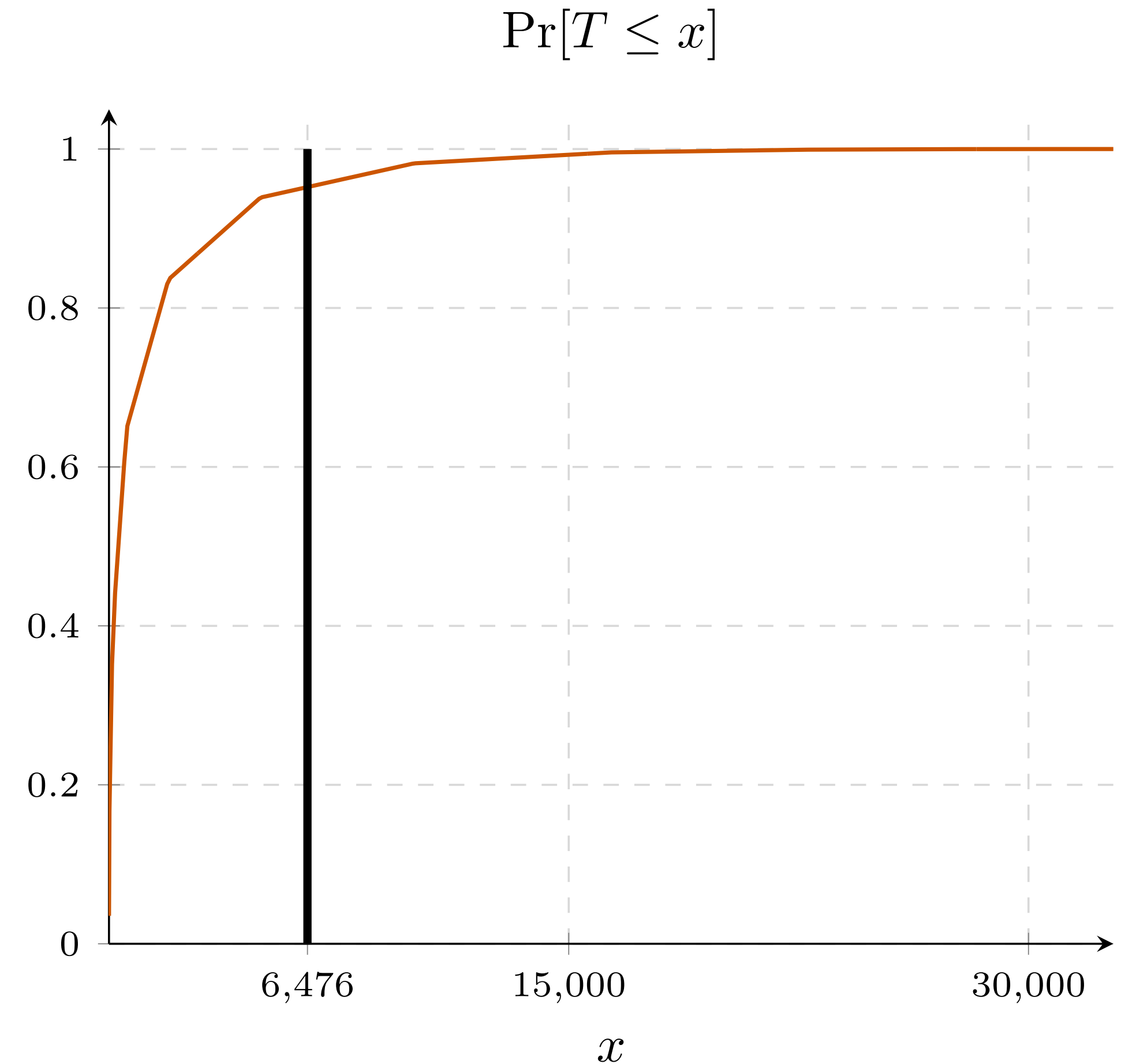
# Likely Keys

## Example:

- ▶ LPN subkey drawn from  $n$ -fold Bernoulli distribution with  $n = 15$ ,  $\tau = 0.2$ .
- ▶ Let  $T$  be RV for number of required guesses.
- ▶ Arikan:  $\mathbb{E}[T] \approx 2^{H_{1/2}(D)} = 6476.64\dots$

Does  $T$  concentrate around  $\mathbb{E}[T]$ ?

- ▶ Markov's inequality:  $T \gg \mathbb{E}[T]$  is unlikely.
- ▶ What about  $T \ll \mathbb{E}[T]$ ?



# Likely Keys

## Example:

- ▶ LPN subkey drawn from  $n$ -fold Bernoulli distribution with  $n = 15$ ,  $\tau = 0.2$ .
- ▶ Let  $T$  be RV for number of required guesses.
- ▶ Arikan:  $\mathbb{E}[T] \approx 2^{H_{1/2}(D)} = 6476.64\dots$

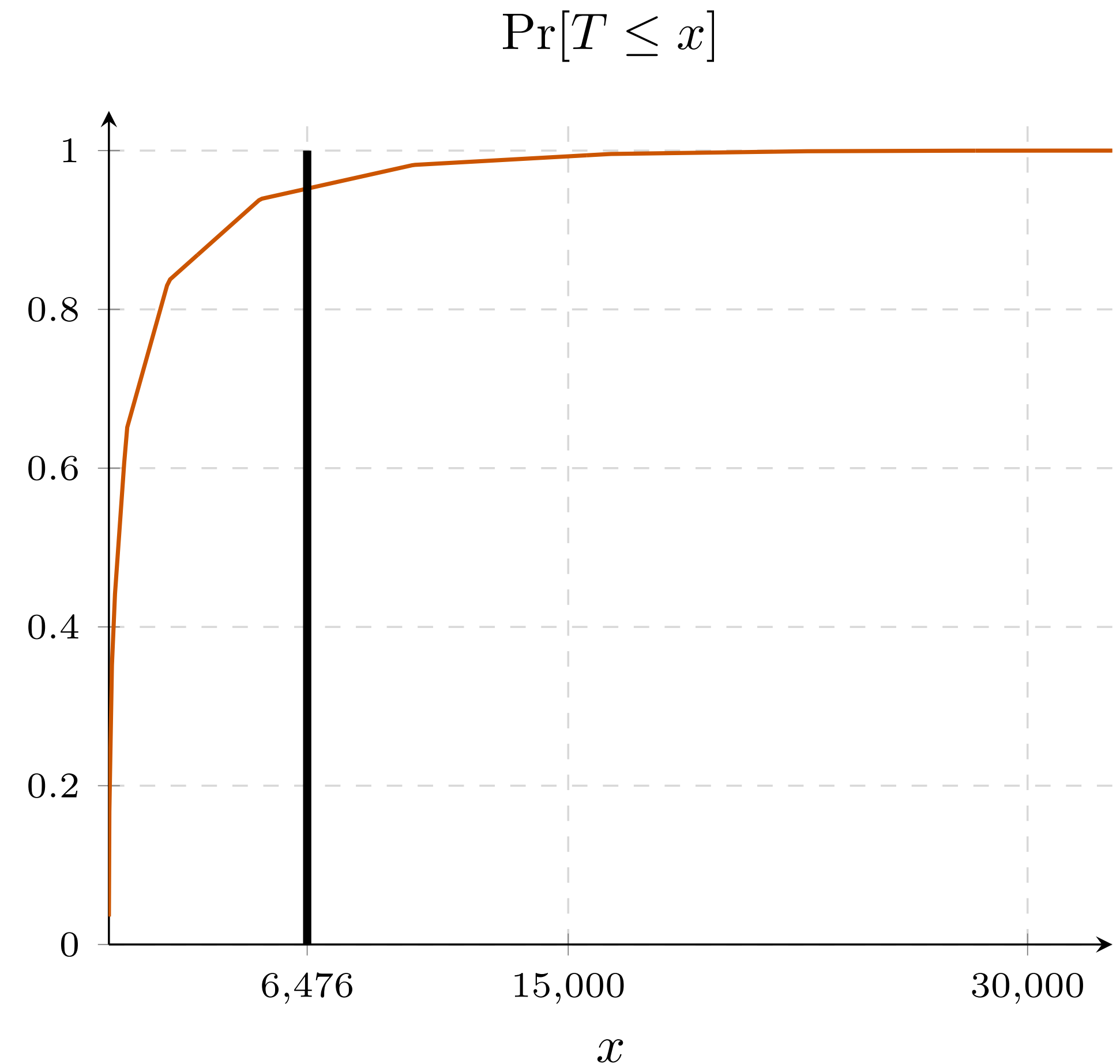
Does  $T$  concentrate around  $\mathbb{E}[T]$ ?

- ▶ Markov's inequality:  $T \gg \mathbb{E}[T]$  is unlikely.
- ▶ What about  $T \ll \mathbb{E}[T]$ ?



**32,768** possible keys.

**307** most likely keys carry 50% of probability mass.



# Likely Keys

**Theorem:**

For keys drawn from a product distribution  $D = \chi^n$ , the  $2^{H_1(D)}$  most likely keys carry at least

$$\frac{1}{2} \pm O(n^{-1/2})$$

of the probability mass.

# Likely Keys

## Theorem:

For keys drawn from a product distribution  $D = \chi^n$ , the  $2^{H_1(D)}$  most likely keys carry at least

$$\frac{1}{2} \pm O(n^{-1/2})$$

of the probability mass.

$$2^{H_0(D)} > 2^{H_{1/2}(D)} > 2^{H_1(D)}$$

↑  
#keys
↑  
expected time  
for key guessing
↑  
#likely keys

# Likely Keys

## Theorem:

For keys drawn from a product distribution  $D = \chi^n$ , the  $2^{H_1(D)}$  most likely keys carry at least

$$\frac{1}{2} \pm O(n^{-1/2})$$

of the probability mass.

