# Discrete Distribution Estimation under Local Privacy

Peter Kairouz, Keith Bonawitz, Daniel Ramage
Accepted to ICML 2016
Paper: go/private-distributions-paper
Slides: go/private-distributions-slides

# Distribution Estimation Under Local Privacy

# Private histograms

We need to understand **patterns across large groups** but **do not need to look at any individual**.

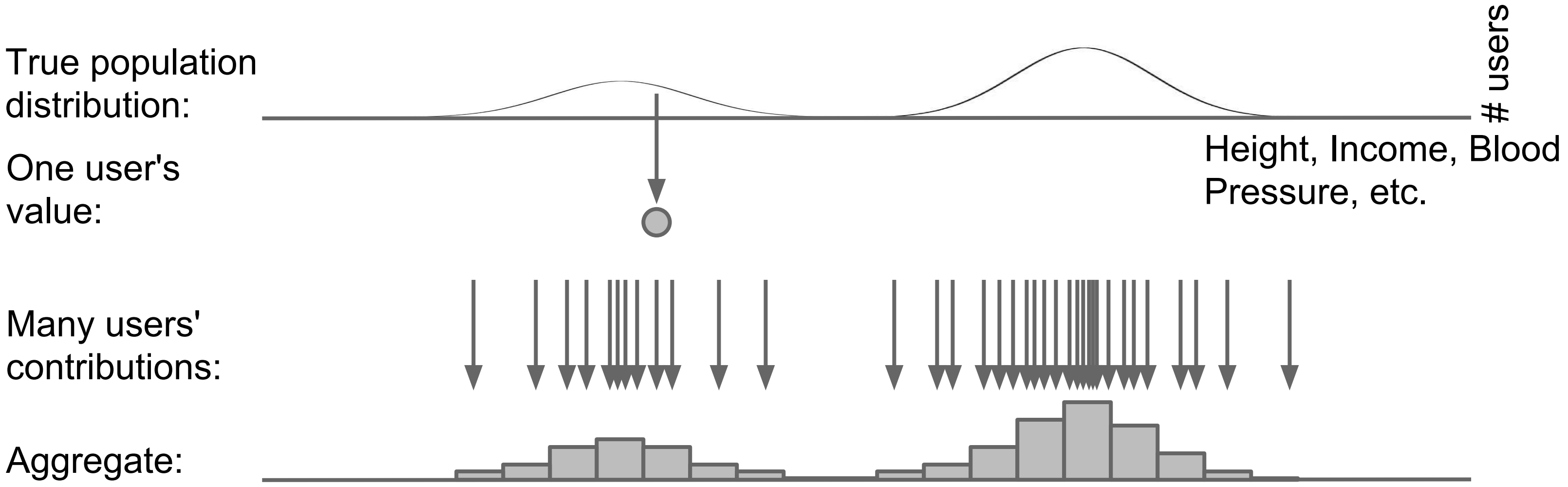
# Private histograms

We need to understand **patterns across large groups** but **do not need to look at any individual**.

**Differential Privacy:**

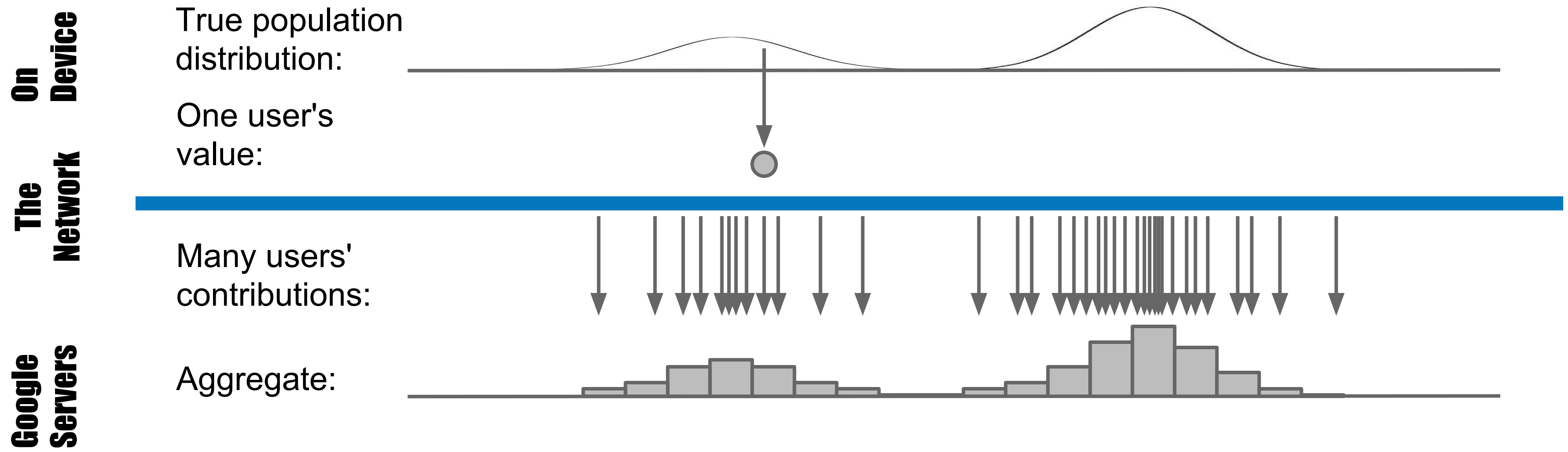
> Provably **limit the information gathered about individual** users by **carefully injecting noise**

# Private histogram intuition

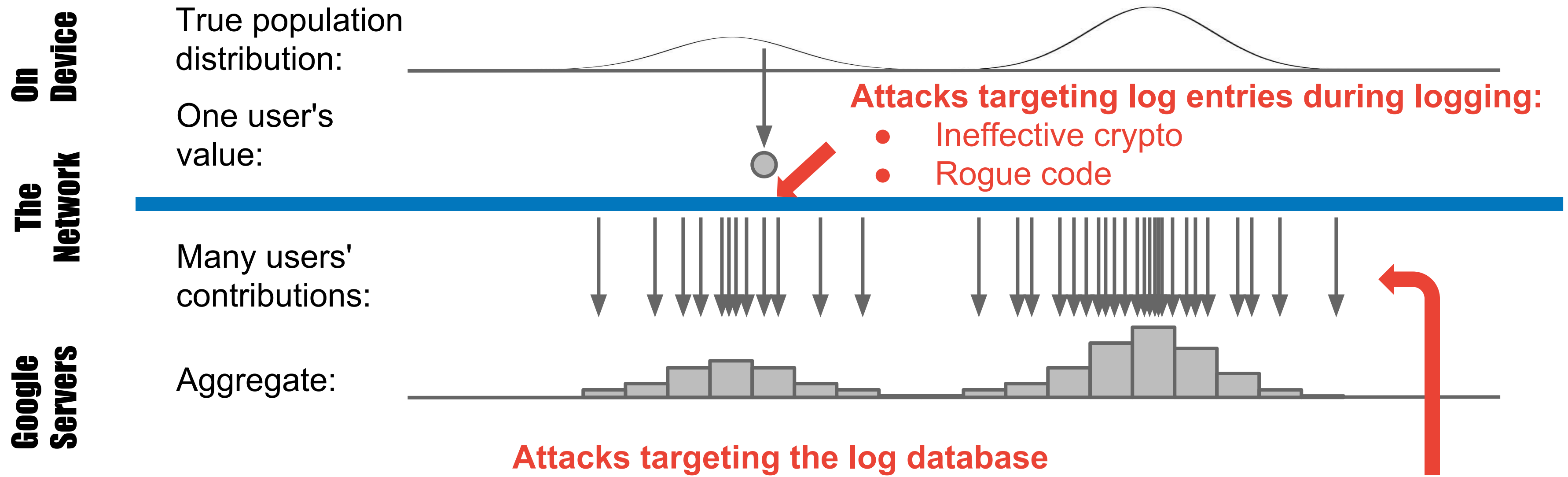



Google

# Private histogram intuition



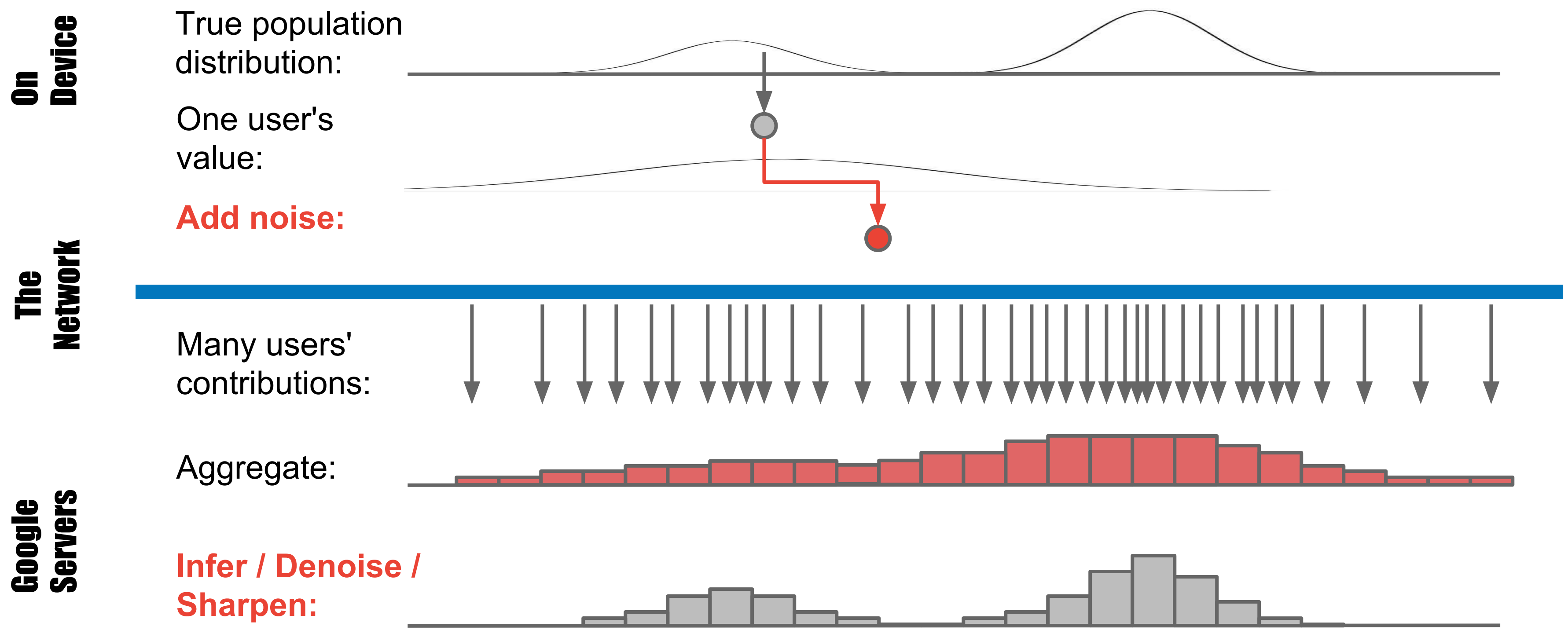
# Private histogram intuition



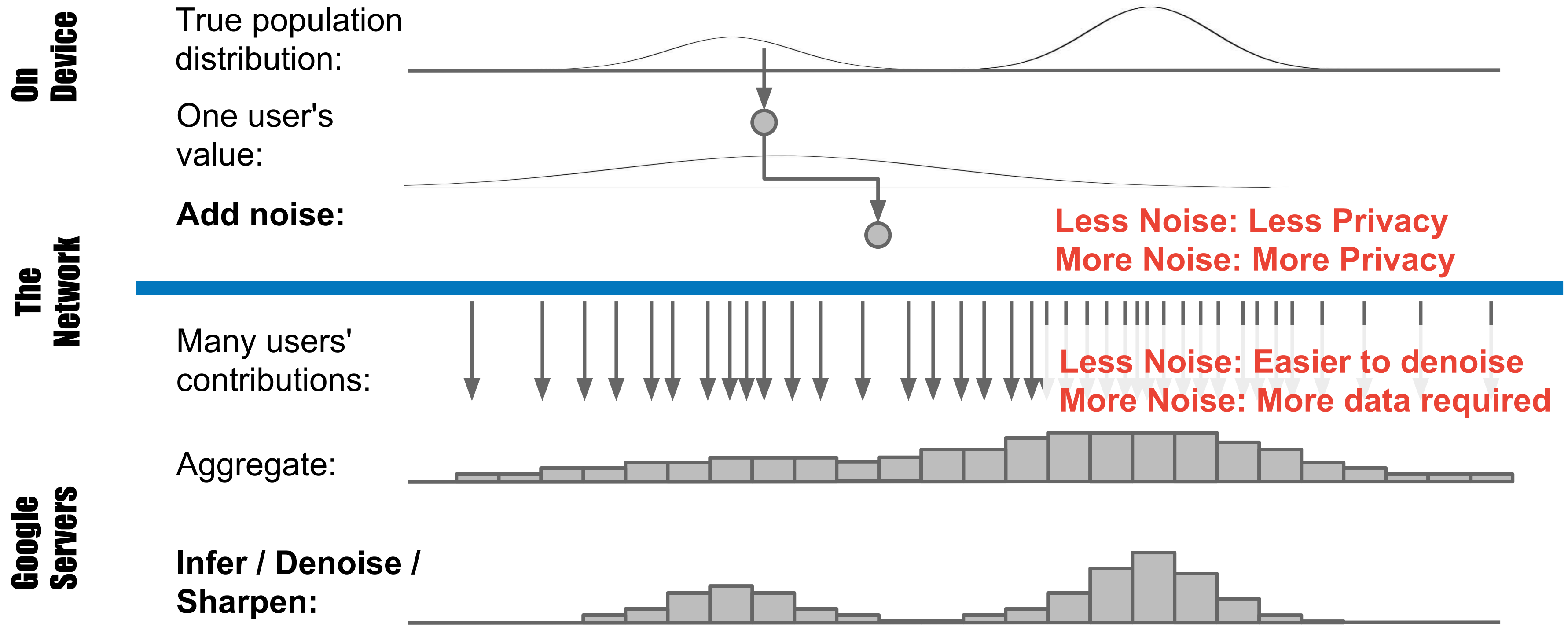

**Attacks targeting the log database**

- Accidental / Incidental
- Authorized user goes rogue
- Break in
- Government compulsion
- Change of ownership

# **Private histogram intuition:** Add noise before logging



# **Private histogram intuition:** Add noise before logging



# Local Differential Privacy

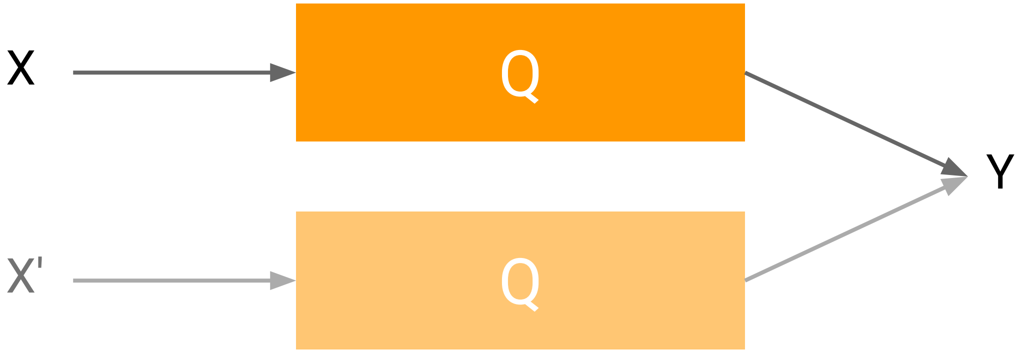
# Local Differential Privacy Definition

Let Q(Y | X) be a privatization mechanism.

# Local Differential Privacy Definition

Let $Q(Y \mid X)$ be a privatization mechanism.

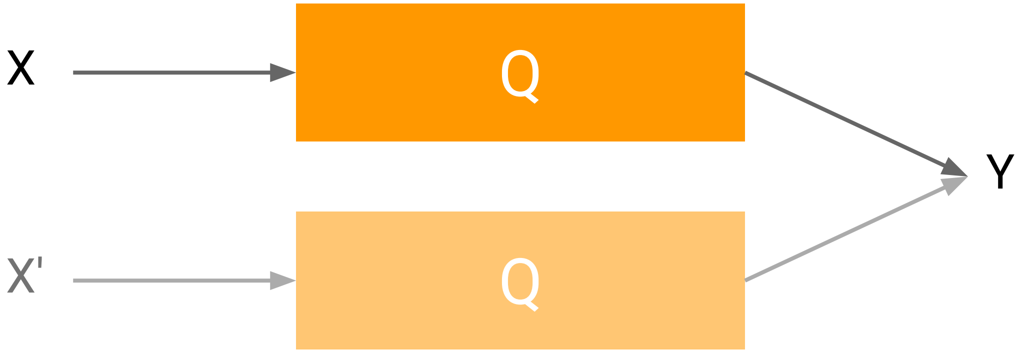
$Q$ is $\varepsilon$-locally differentially private if $Q(Y|X) \leq e^{\varepsilon} Q(Y|X')$ for all $X$, $X'$, $Y$



# Local Differential Privacy Definition

Let $Q(Y \mid X)$ be a privatization mechanism.

Q is ε-locally differentially private if $\frac{Q(Y|X)}{Q(Y|X')} \leq e^{\varepsilon}$ for all X, X', Y

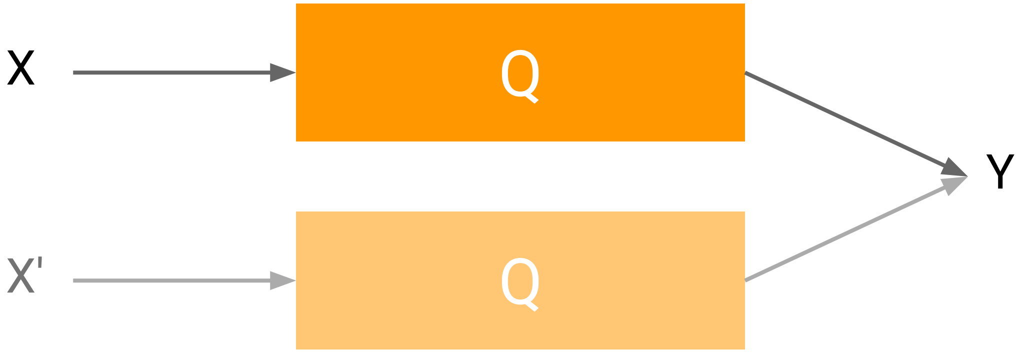


# Local Differential Privacy Definition

Let Q(Y | X) be a privatization mechanism.

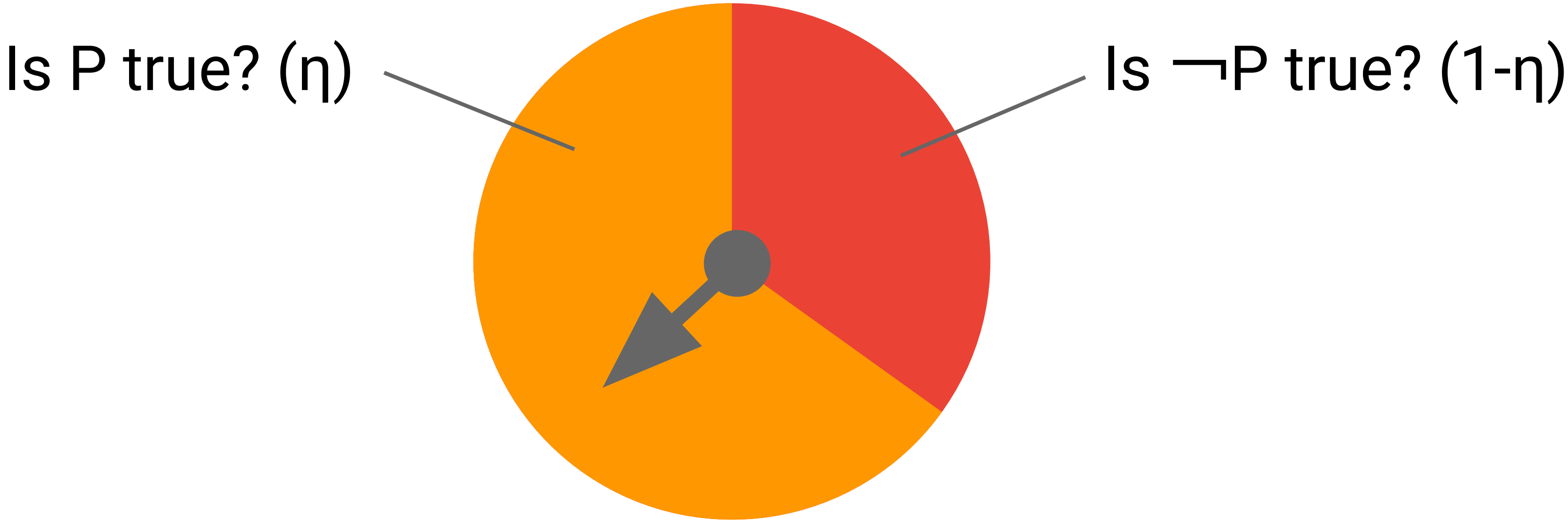
Q is ε-locally differentially private if $\frac{Q(Y|X)}{Q(Y|X')} \leq e^{\varepsilon}$ for all X, X', Y

Worst Case

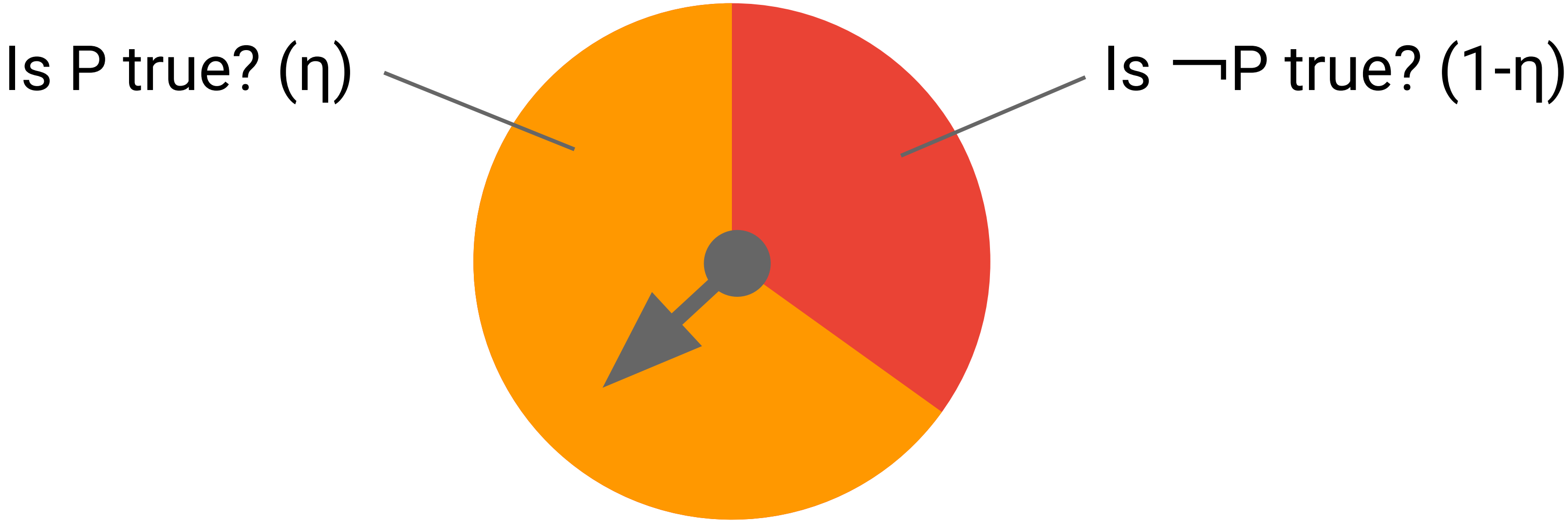


# Binary Alphabets

# Warner's Randomized Response, 1965 ("W-RR")



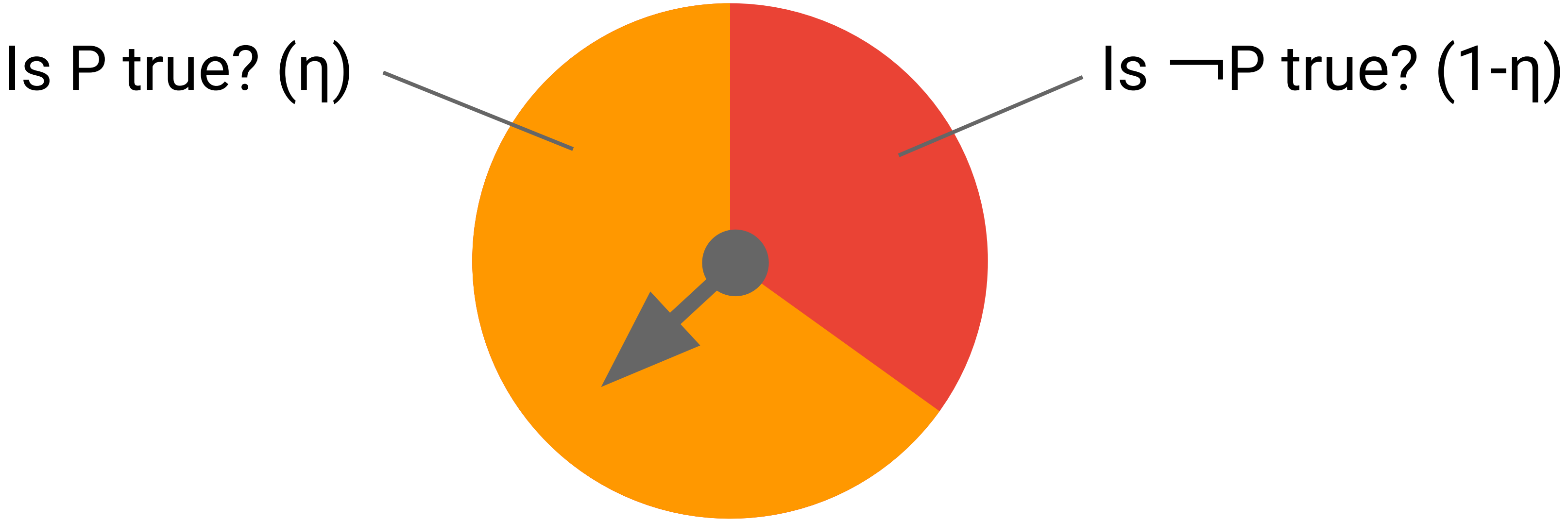
# Warner's Randomized Response, 1965 ("W-RR")




100% P → η say yes

0% P → (1-η) say yes

# Warner's Randomized Response, 1965 ("W-RR")




$\varepsilon$-locally differentially private, for $e^{\varepsilon} = \eta / (1-\eta)$

# Warner's Randomized Response, 1965 ("W-RR")

$Q_{W\text{-}RR}$:

$$Q_{\mathrm{WRR}} = \frac{1}{e^{\varepsilon}+1} \begin{cases} e^{\varepsilon} & \text{if } y = x, \\ 1 & \text{if } y \neq x. \end{cases}$$

# Warner's Randomized Response, 1965 ("W-RR")

$Q_{\text{W-RR}}$:

$$Q_{WRR} = \frac{1}{e^{\varepsilon}+1}\begin{bmatrix} e^{\varepsilon} & 1 \\ 1 & e^{\varepsilon} \end{bmatrix}$$

m=pQ

row vectors

p(x): distribution over inputs
m(y): distribution over outputs

# Warner's Randomized Response, 1965 ("W-RR")

$Q_{W\text{-}RR}$:

$$Q_{\mathrm{WRR}} = \frac{1}{e^{\varepsilon}+1}\begin{bmatrix} e^{\varepsilon} & 1 \\ 1 & e^{\varepsilon} \end{bmatrix}$$

**Decoding**: $m=pQ$, so $p_{est}=m_{obs}Q^{-1}$

$p(x)$: distribution over inputs
$m(y)$: distribution over outputs

# Warner's Randomized Response, 1965 ("W-RR")

$Q_{W\text{-}RR}$:

$$Q_{\mathrm{WRR}} = \frac{1}{e^{\varepsilon}+1}\begin{bmatrix} e^{\varepsilon} & 1 \\ 1 & e^{\varepsilon} \end{bmatrix}$$

$$\hat{p} = \left(\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1}\right)\frac{T}{n} - \frac{1}{e^{\varepsilon}-1}$$

p: probability of predicate P
T: number of "Yes" reports
n: number of reports (total)

W-RR offers
**optimal utility** for
binary alphabets.

# Warner's Randomized Response, 1965 ("W-RR")

**Theorem 2** *For all binary distributions* $\mathbf{p}$, *all loss functions* $\ell$, *and all privacy levels* $\varepsilon$, $\mathbf{Q}_{\mathrm{WRR}}$ *is the optimal solution to the private minimax distribution estimation problem*

# $k$-ary Alphabets

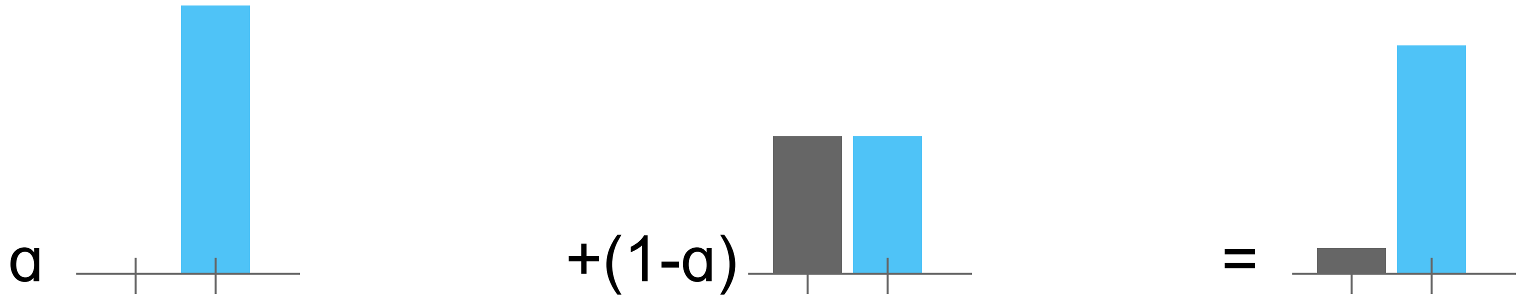
# Two different ways to extend to k-ary alphabets

1. k-RR modifies the mechanism
2. k-RAPPOR modifies the encoding

# k-RR modifies the mechanism

$Q_{W\text{-}RR}$ :

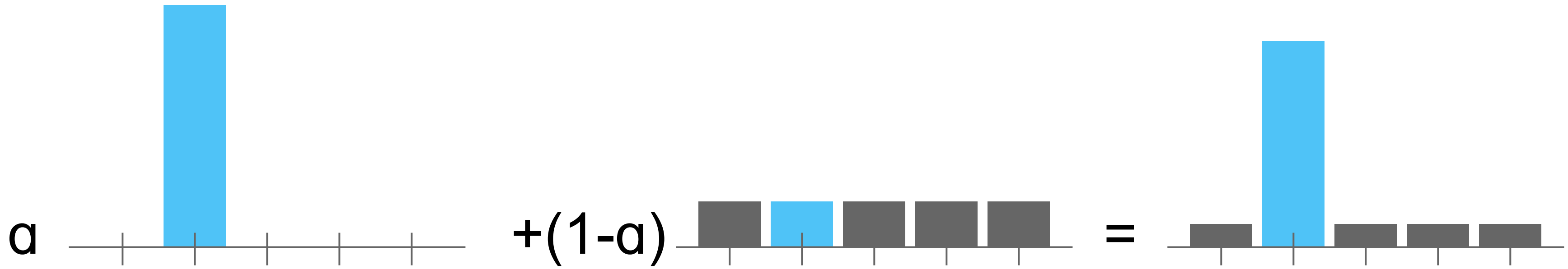
$$\boldsymbol{Q}_{\mathrm{WRR}} = \frac{1}{e^{\varepsilon}+1} \begin{cases} e^{\varepsilon} & \text{if } y = x, \\ 1 & \text{if } y \neq x. \end{cases}$$




$$a = \frac{e^{\varepsilon}-1}{e^{\varepsilon}+1}$$

# k-RR modifies the mechanism

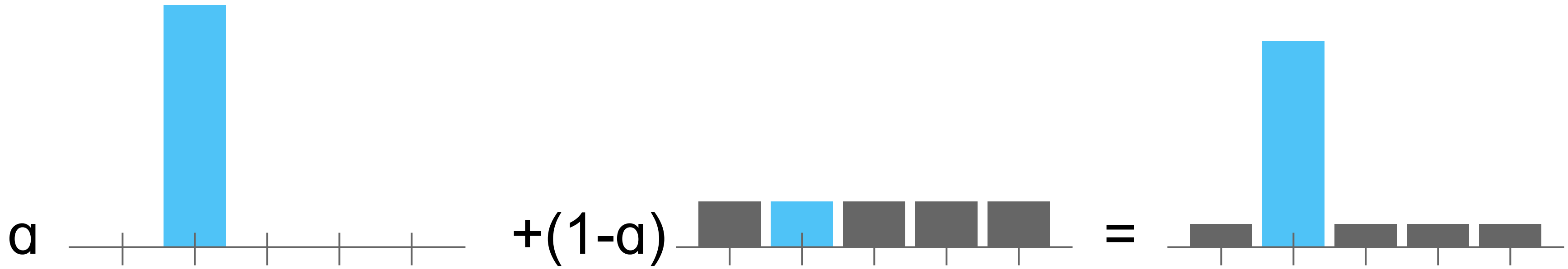
$Q_{k\text{-}RR}$:



# k-RR modifies the mechanism

$Q_{\text{k-RR}}$:

$$\boldsymbol{Q}_{\mathrm{KRR}}(y|x) = \frac{1}{k-1+e^{\varepsilon}} \begin{cases} e^{\varepsilon} & \text{if } y = x, \\ 1 & \text{if } y \neq x. \end{cases}$$



# k-RR modifies the mechanism

$Q_{k\text{-RR}}$ :

$$\boldsymbol{Q}_{\mathrm{KRR}}(y|x) = \frac{1}{k-1+e^{\varepsilon}} \begin{cases} e^{\varepsilon} & \text{if } y = x, \\ 1 & \text{if } y \neq x. \end{cases}$$

**Decoding**: $m=pQ$, so $p_{est}=m_{obs}Q^{-1}$

p(x): distribution over inputs
m(y): distribution over outputs

# k-RR modifies the mechanism

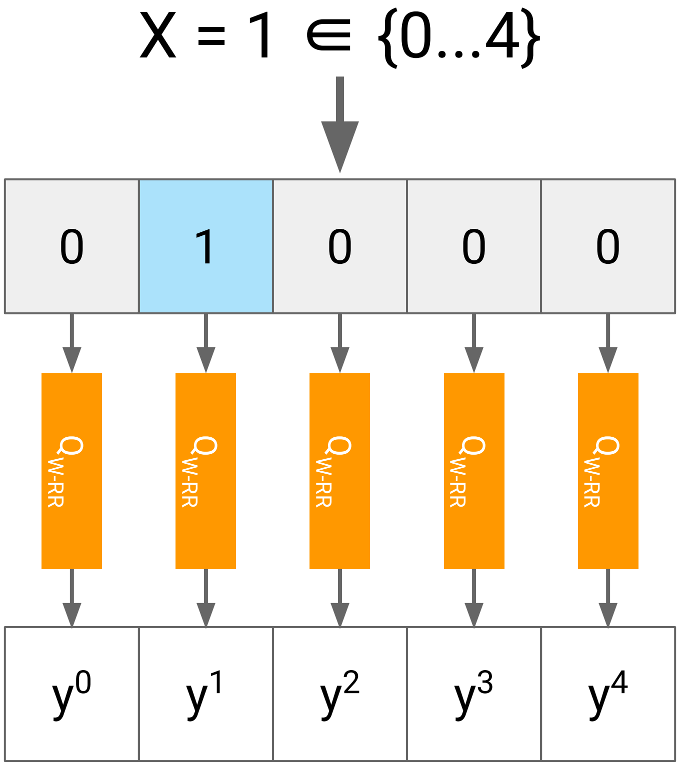
$Q_{\text{k-RR}}$:

$$Q_{\mathrm{KRR}}(y|x) = \frac{1}{k-1+e^{\varepsilon}} \begin{cases} e^{\varepsilon} & \text{if } y = x, \\ 1 & \text{if } y \neq x. \end{cases}$$

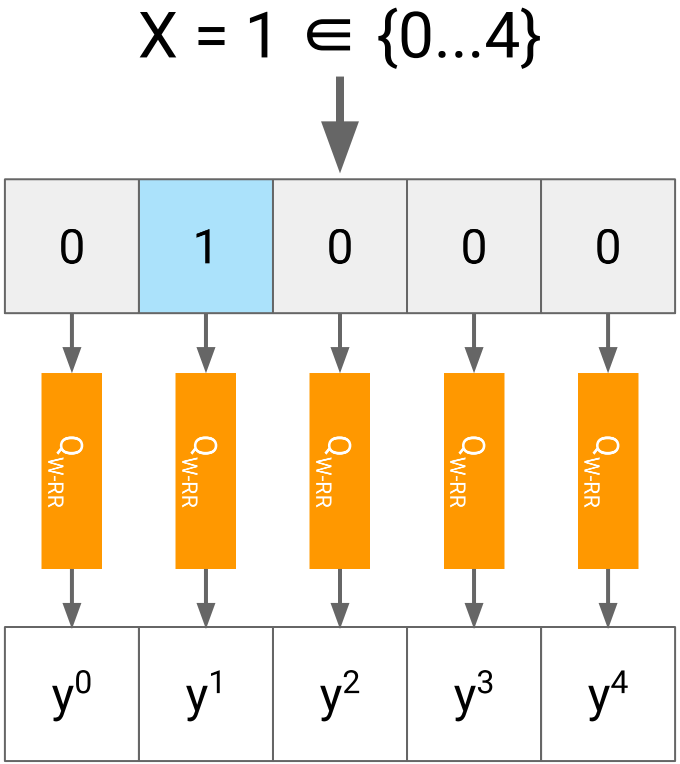
$$\begin{aligned} \hat{p} &= \hat{m} Q_{\mathrm{KRR}}^{-1} \\ &= \frac{e^{\varepsilon}+k-1}{e^{\varepsilon}-1}\hat{m} - \frac{1}{e^{\varepsilon}-1} \end{aligned}$$

p(x): distribution over inputs
m(y): distribution over outputs

# k-RAPPOR modifies the encoding

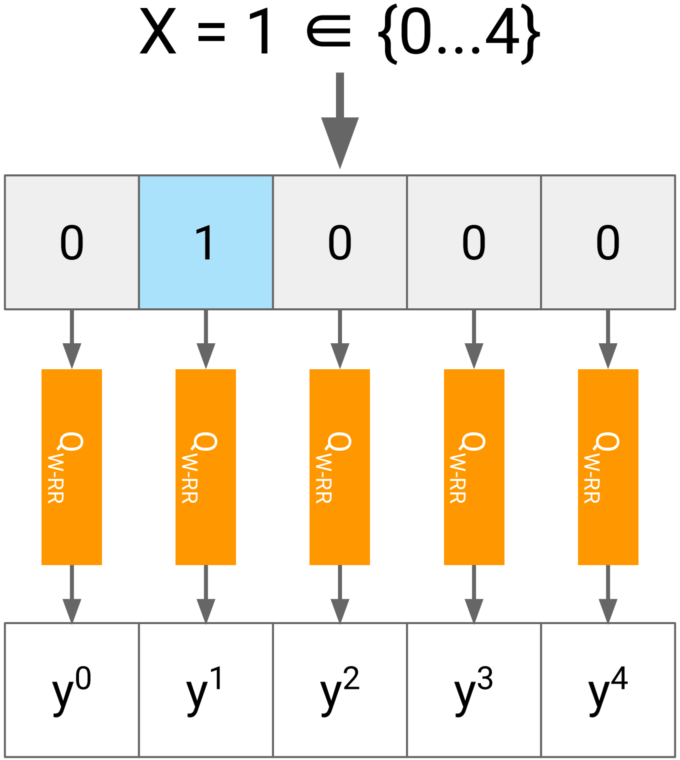


# k-RAPPOR modifies the encoding




$$Y^{(j)} = \begin{cases} \tilde{X}^{(j)} & \text{with probability } \dfrac{e^{\varepsilon/2}}{1+e^{\varepsilon/2}} \\ 1 - \tilde{X}^{(j)} & \text{with probability } \dfrac{1}{1+e^{\varepsilon/2}} \end{cases}$$

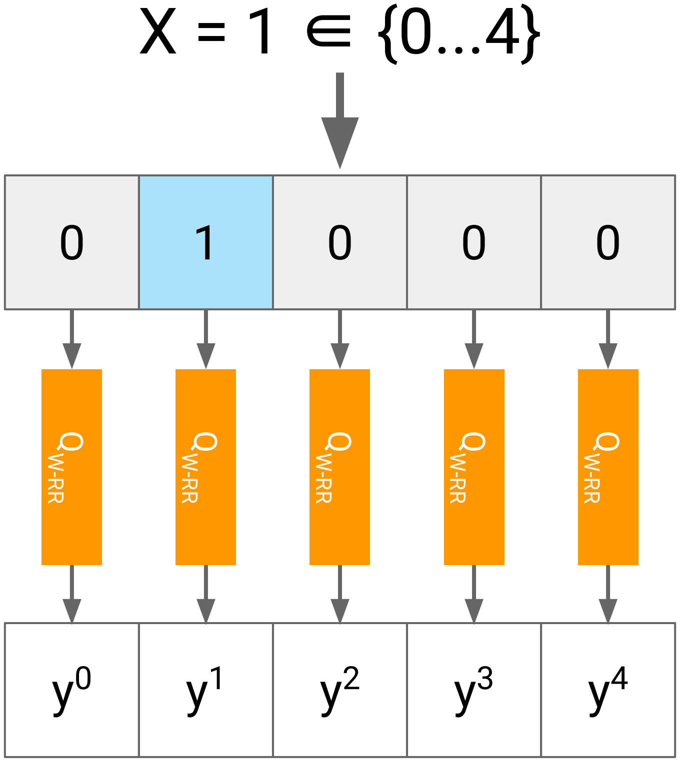
# k-RAPPOR modifies the encoding




2 bits different between any X, X'

$$Y^{(j)} = \begin{cases} \tilde{X}^{(j)} & \text{with probability } \dfrac{e^{\varepsilon/2}}{1+e^{\varepsilon/2}} \\ 1-\tilde{X}^{(j)} & \text{with probability } \dfrac{1}{1+e^{\varepsilon/2}} \end{cases}$$

# k-RAPPOR modifies the encoding




$$Y^{(j)} = \begin{cases} \tilde{X}^{(j)} & \text{with probability } \dfrac{e^{\varepsilon/2}}{1+e^{\varepsilon/2}} \\ 1-\tilde{X}^{(j)} & \text{with probability } \dfrac{1}{1+e^{\varepsilon/2}} \end{cases}$$

**Decode each bit independently:**

$$\hat{p}_j = \left(\frac{e^{\varepsilon/2}+1}{e^{\varepsilon/2}-1}\right)\frac{T_j}{n} - \frac{1}{e^{\varepsilon/2}-1}$$

$p_j$: probability of X=j

$T_j$: number of reports with $y^j$=1

N: number of reports (total)

# Utility (Bounds on Expected Loss)

| | $\mathbb{E}\,\|\|\hat{\boldsymbol{p}} - \boldsymbol{p}\|\|_1$ | $\mathbb{E}\,\|\|\hat{\boldsymbol{p}} - \boldsymbol{p}\|\|_2^2$ |
|---|---|---|
| **No Privatization** | $\sqrt{\frac{2(k-1)}{\pi n}}$ | $\frac{1-\frac{1}{k}}{n}$ |

# Utility (Bounds on Expected Loss)

| | $\mathbb{E}\,\|\|\hat{\boldsymbol{p}}-\boldsymbol{p}\|\|_1$ | $\mathbb{E}\,\|\|\hat{\boldsymbol{p}}-\boldsymbol{p}\|\|_2^2$ |
|---|---|---|
| **No Privatization** | $\sqrt{\frac{2(k-1)}{\pi n}}$ | $\frac{1-\frac{1}{k}}{n}$ |
| ***k*-RR** | $\left(\frac{e^\varepsilon+k-1}{e^\varepsilon-1}\right)\sqrt{\frac{2(k-1)}{\pi n}}$ | $\left(\frac{e^\varepsilon+k-1}{e^\varepsilon-1}\right)^2\frac{1-\frac{1}{k}}{n}$ |
| ***k*-RAPPOR** | $\sqrt{\frac{2(e^{\varepsilon/2}+k-1)(e^{\varepsilon/2}(k-1)+1)}{(e^{\varepsilon/2}-1)^2\pi n}}$ | $\frac{1-\frac{1}{k}}{n}\left(1+\frac{k^2e^{\varepsilon/2}}{(k-1)(e^{\varepsilon/2}-1)^2}\right)$ |

# Utility (Bounds on Expected Loss)

| | $\mathbb{E}\|\|\hat{p}-p\|\|_1$ | $\mathbb{E}\|\|\hat{p}-p\|\|_2^2$ | **Effective Samples** |
|---|---|---|---|
| **No Privatization** | $\boxed{\sqrt{\frac{2(k-1)}{\pi n}}}$ | $\boxed{\frac{1-\frac{1}{k}}{n}}$ | n |
| ***k*-RR** | $\left(\frac{e^\varepsilon+k-1}{e^\varepsilon-1}\right)\boxed{\sqrt{\frac{2(k-1)}{\pi n}}}$ | $\left(\frac{e^\varepsilon+k-1}{e^\varepsilon-1}\right)^2\boxed{\frac{1-\frac{1}{k}}{n}}$ | $n\left(\frac{e^\varepsilon-1}{e^\varepsilon+k-1}\right)^2$ |
| ***k*-RAPPOR** | $\sqrt{\frac{2(e^{\varepsilon/2}+k-1)(e^{\varepsilon/2}(k-1)+1)}{(e^{\varepsilon/2}-1)^2\pi n}}$ | $\boxed{\frac{1-\frac{1}{k}}{n}}\left(1+\frac{k^2e^{\varepsilon/2}}{(k-1)(e^{\varepsilon/2}-1)^2}\right)$ | $n\left(\frac{(k-1)(e^{\varepsilon/2}-1)^2}{(k-1)(e^{\varepsilon/2}-1)^2+k^2e^\varepsilon}\right)$ |

# Utility (Effective Samples)

| | General |
|---|---|
| **No Privatization** | n |
| ***k*-RR** | $n\left(\frac{e^{\varepsilon}-1}{e^{\varepsilon}+k-1}\right)^2$ |
| ***k*-RAPPOR** | $n\left(\frac{(k-1)(e^{\varepsilon/2}-1)^2}{(k-1)(e^{\varepsilon/2}-1)^2+k^2e^{\varepsilon}}\right)$ |

# Utility (Effective Samples)

| | $\varepsilon \approx \ln(k)$ (Low Privacy) | General |
|---|---|---|
| **No Privatization** | n | n |
| ***k*-RR** | n/4 | $n\left(\frac{e^{\varepsilon}-1}{e^{\varepsilon}+k-1}\right)^2$ |
| ***k*-RAPPOR** | n/sqrt(k) | $n\left(\frac{(k-1)(e^{\varepsilon/2}-1)^2}{(k-1)(e^{\varepsilon/2}-1)^2+k^2e^{\varepsilon}}\right)$ |

For k-ary alphabets:
**k-RR is order-optimal for low privacy** (and k-RAPPOR is sub-optimal)

# Utility (Effective Samples)

| | $\varepsilon \approx \ln(k)$ (Low Privacy) | Small $\varepsilon$ (High Privacy) | General |
|---|---|---|---|
| **No Privatization** | n | n | n |
| ***k*-RR** | n/4 | $n\varepsilon^2/k^2$ | $n\left(\frac{e^{\varepsilon}-1}{e^{\varepsilon}+k-1}\right)^2$ |
| ***k*-RAPPOR** | n/sqrt(k) | $n\varepsilon^2/4k$ | $n\left(\frac{(k-1)(e^{\varepsilon/2}-1)^2}{(k-1)(e^{\varepsilon/2}-1)^2+k^2e^{\varepsilon}}\right)$ |

For k-ary alphabets:
**k-RAPPOR is order-optimal for high privacy**
(and k-RR is sub-optimal)

# Constraining to the Simplex

Probability vectors sum to 1 and all elements are non-negative.

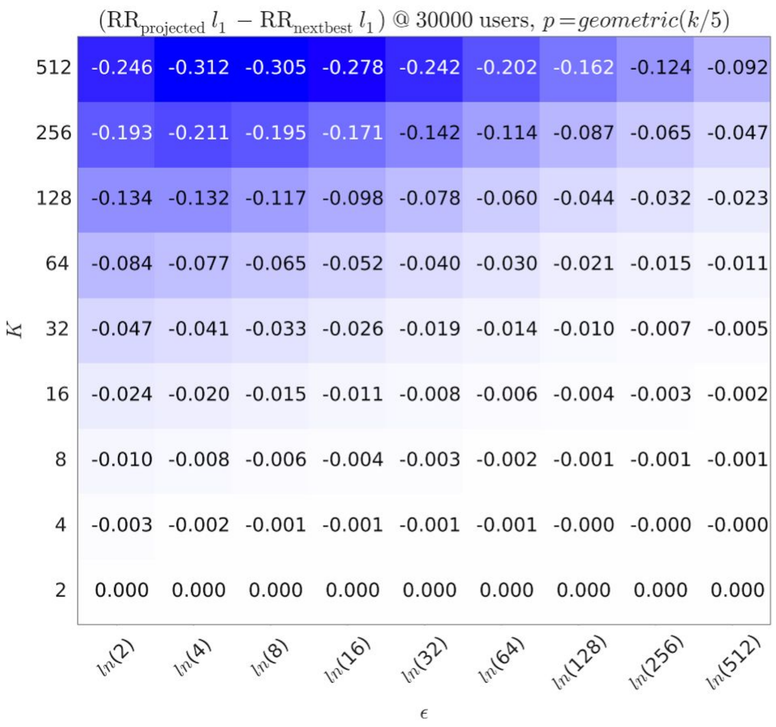
# Constraining to the Simplex

Probability vectors sum to 1 and all elements are non-negative.

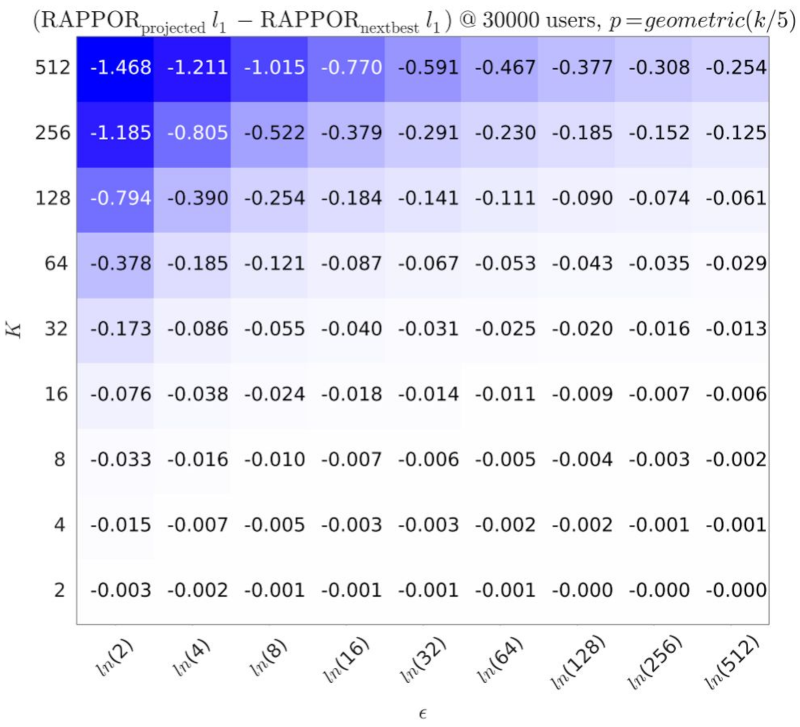
1. Do nothing.
2. Truncate and renormalize.
3. Project onto the nearest point on the simplex.
4. Something else creative (e.g. a different decoder)

# Constraining to the Simplex

## *k*-RR




## *k*-RAPPOR



For skewed distributions, the **projected estimator offers the best utility**.

# Open Alphabets

# Open Alphabets

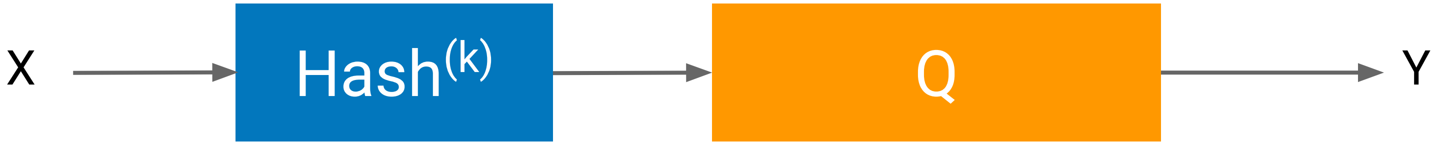
- What if we don't know the set of input symbols ahead of time?
- Can we want to avoid penalties for having large $k$?

# Hashing (Sketches)

**Instead of encoding x directly...**

# Hashing (Sketches)

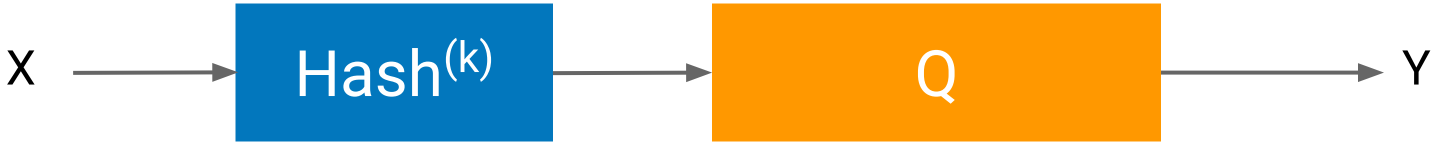
Instead of encoding x directly, **we encode hash(x) mod k.**



# Hashing (Sketches)

Instead of encoding x directly, we encode hash(x) mod k.

**But what about collisions?**

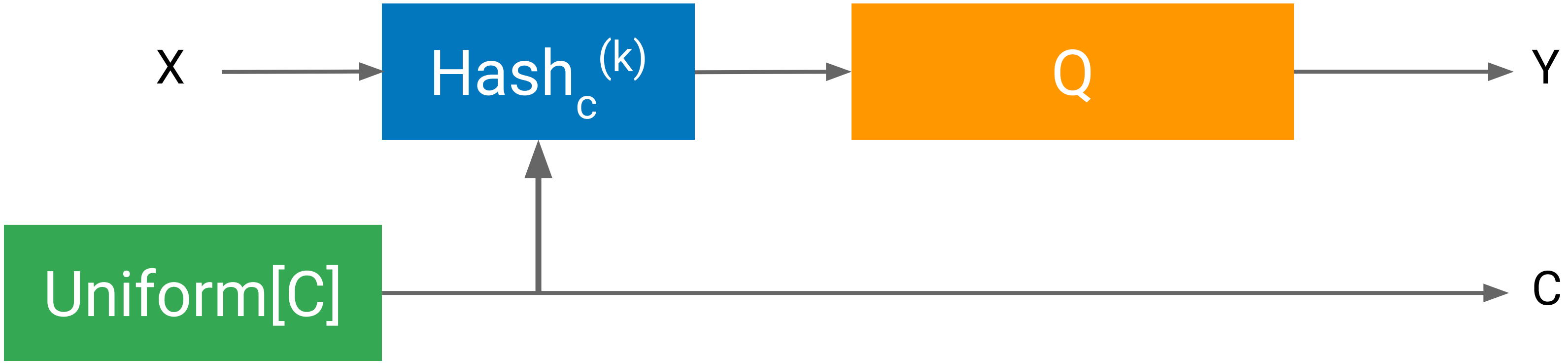


# Hashing (Sketches)

Instead of encoding x directly, we encode hash(x) mod k.

But what about collisions?

**Multiple Hash Functions → Independent Views (Sketches)**

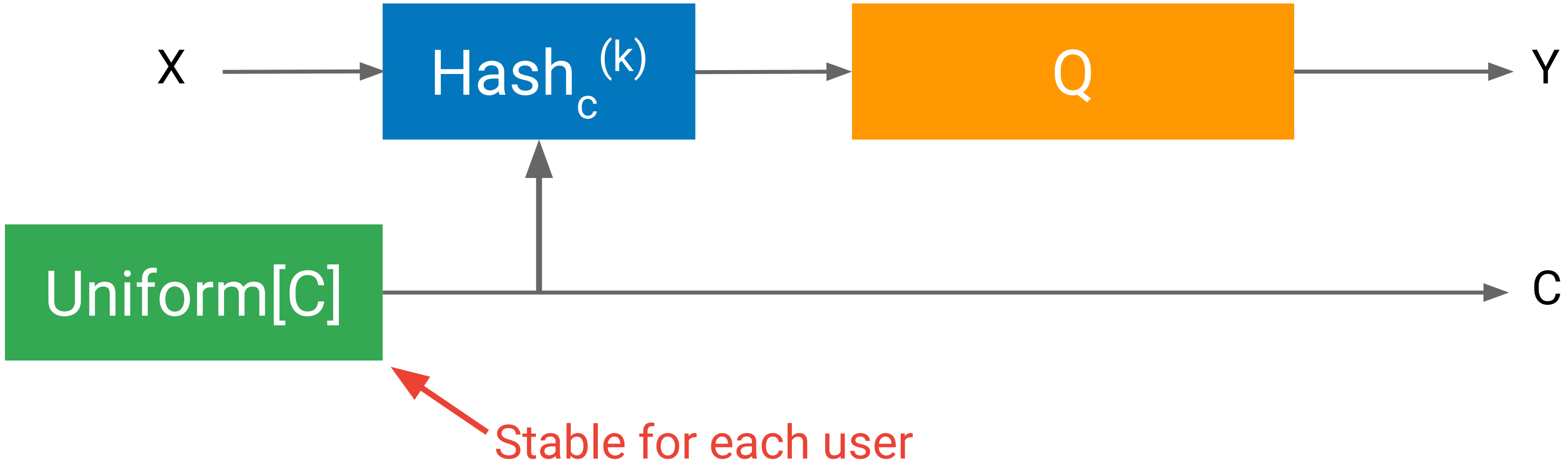


# Hashing (Sketches)

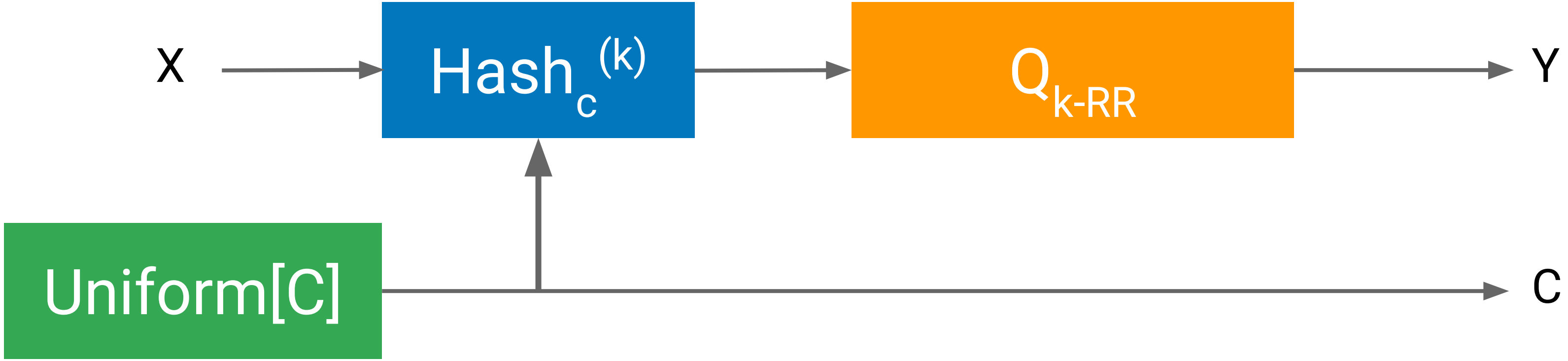
Instead of encoding x directly, we encode hash(x) mod k.

But what about collisions?

**Multiple Hash Functions → Independent Views (Sketches)**



# O-RR

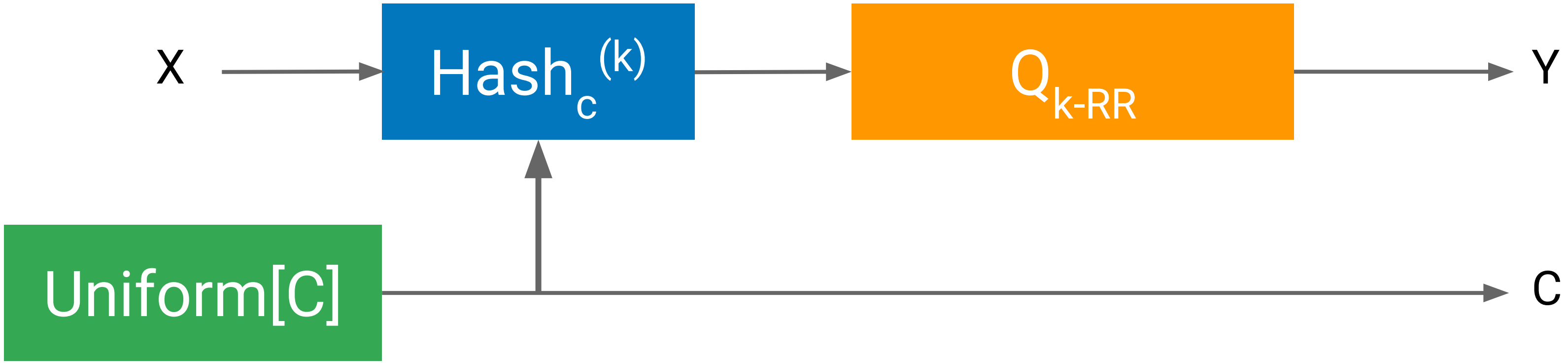


# O-RR

$$Q_{\mathrm{ORR}} = \frac{1}{C}\frac{1}{e^{\varepsilon}+k-1}\left(\mathbf{1}+(e^{\varepsilon}-1)\boldsymbol{H}\right)$$

where:

$$\boldsymbol{H}(y, c|s) = \mathbb{1}_{\{\mathrm{HASH}_c^{(k)}(s)=y\}}$$



# O-RR

$$\boldsymbol{Q}_{\mathrm{ORR}} = \frac{1}{C}\frac{1}{e^{\varepsilon}+k-1}\left(\mathbf{1}+(e^{\varepsilon}-1)\boldsymbol{H}\right)$$

where:

$$\boldsymbol{H}(y, c|s) = \mathbb{1}_{\{\mathrm{HASH}_c^{(k)}(s)=y\}}$$

**Decoding**:

$$\hat{\mathbf{p}}_{\mathrm{ORR}}\boldsymbol{H} = \frac{1}{e^{\varepsilon}-1}\left(C(e^{\varepsilon}+k-1)\hat{\boldsymbol{m}}-\mathbf{1}\right)$$

p(s): distribution over inputs
m(y): distribution over outputs

# O-RR

$$\boldsymbol{Q}_{\mathrm{ORR}} = \frac{1}{C}\frac{1}{e^{\varepsilon}+k-1}\left(\mathbf{1}+(e^{\varepsilon}-1)\boldsymbol{H}\right)$$

where:

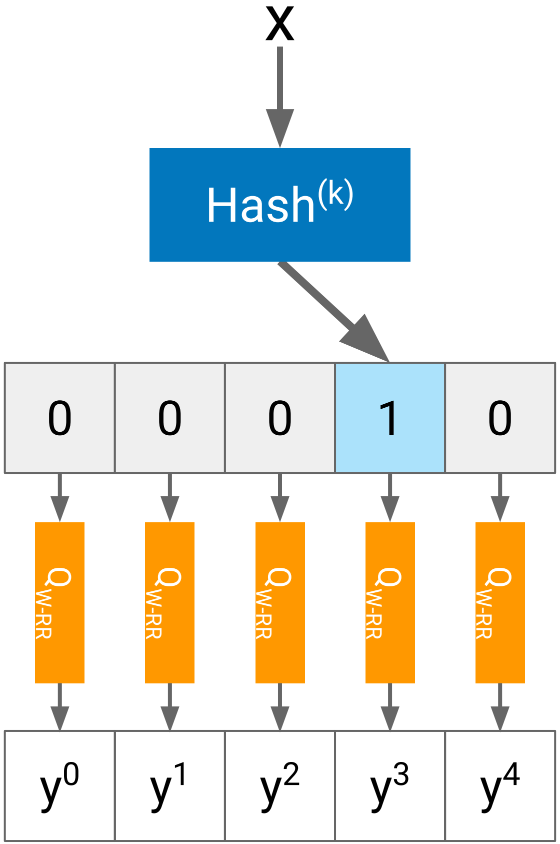
$$\boldsymbol{H}(y, c|s) = \mathbb{1}_{\{\mathrm{HASH}_c^{(k)}(s)=y\}}$$

**Decoding**: (H not invertible: solve via least squares)

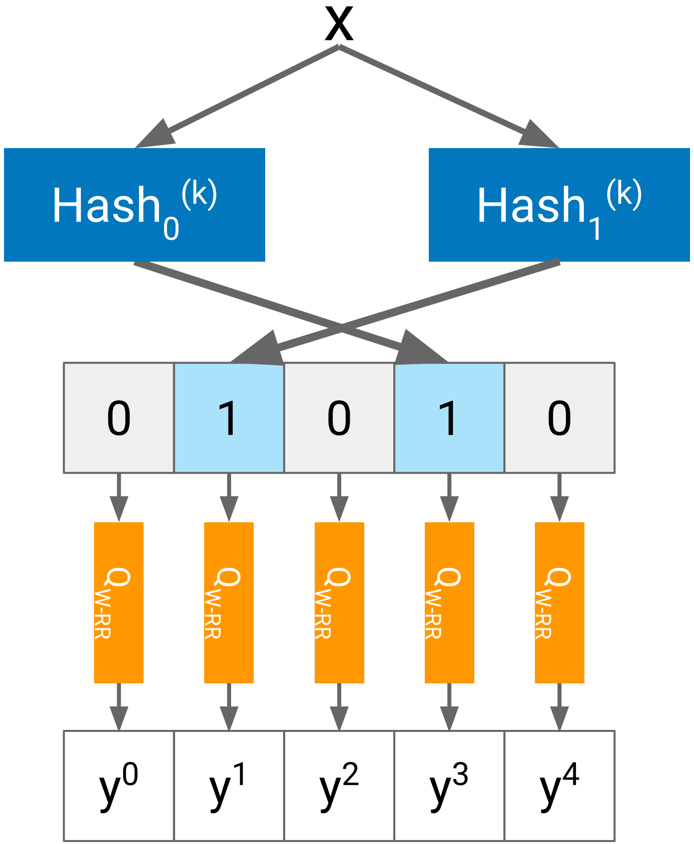
$$\hat{\boldsymbol{p}}_{\mathrm{ORR}}\boldsymbol{H} = \frac{1}{e^{\varepsilon}-1}\left(C(e^{\varepsilon}+k-1)\hat{\boldsymbol{m}}-\mathbf{1}\right)$$

p(s): distribution over inputs
m(y): distribution over outputs

# O-RAPPOR

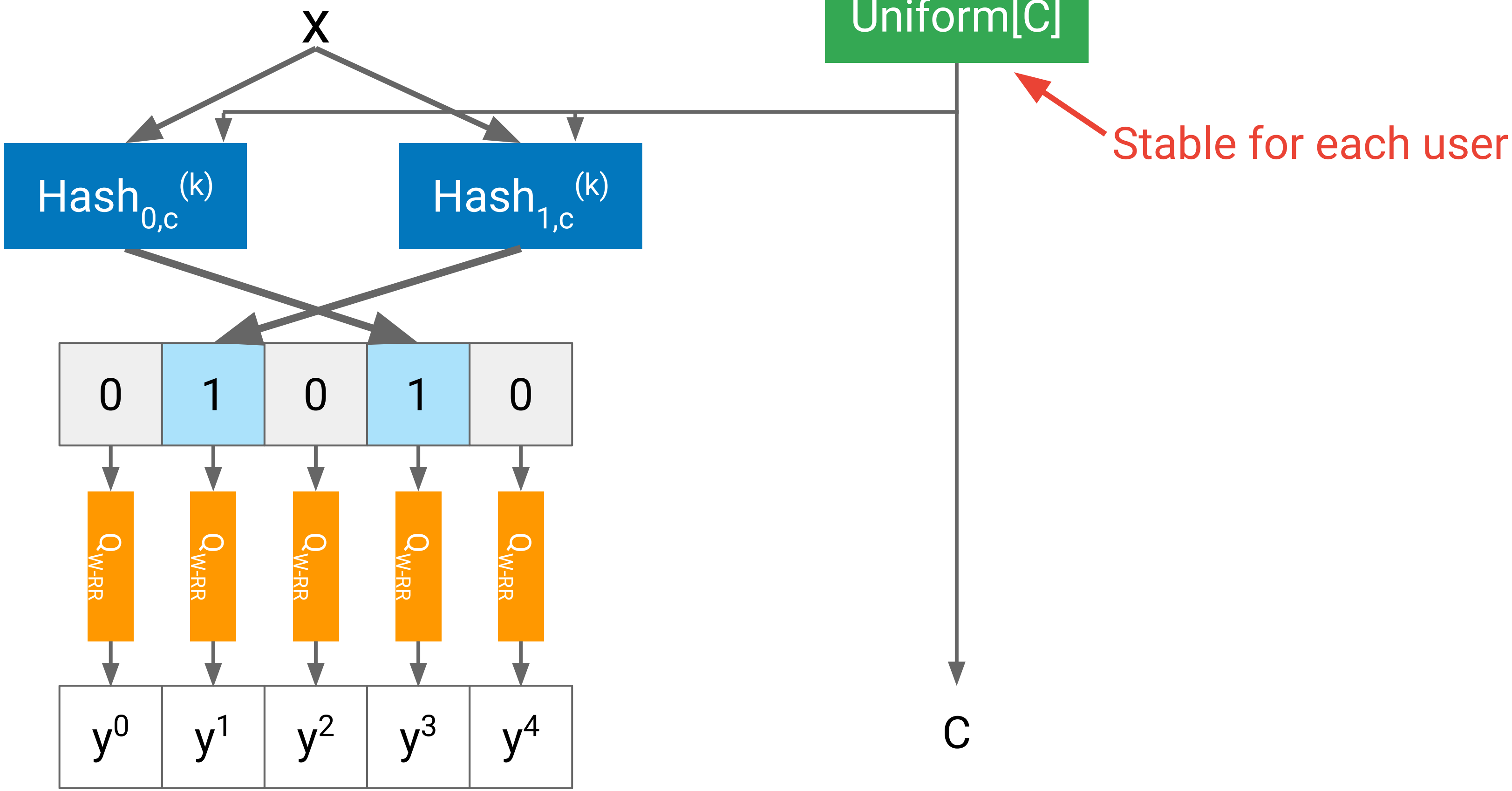


# O-RAPPOR



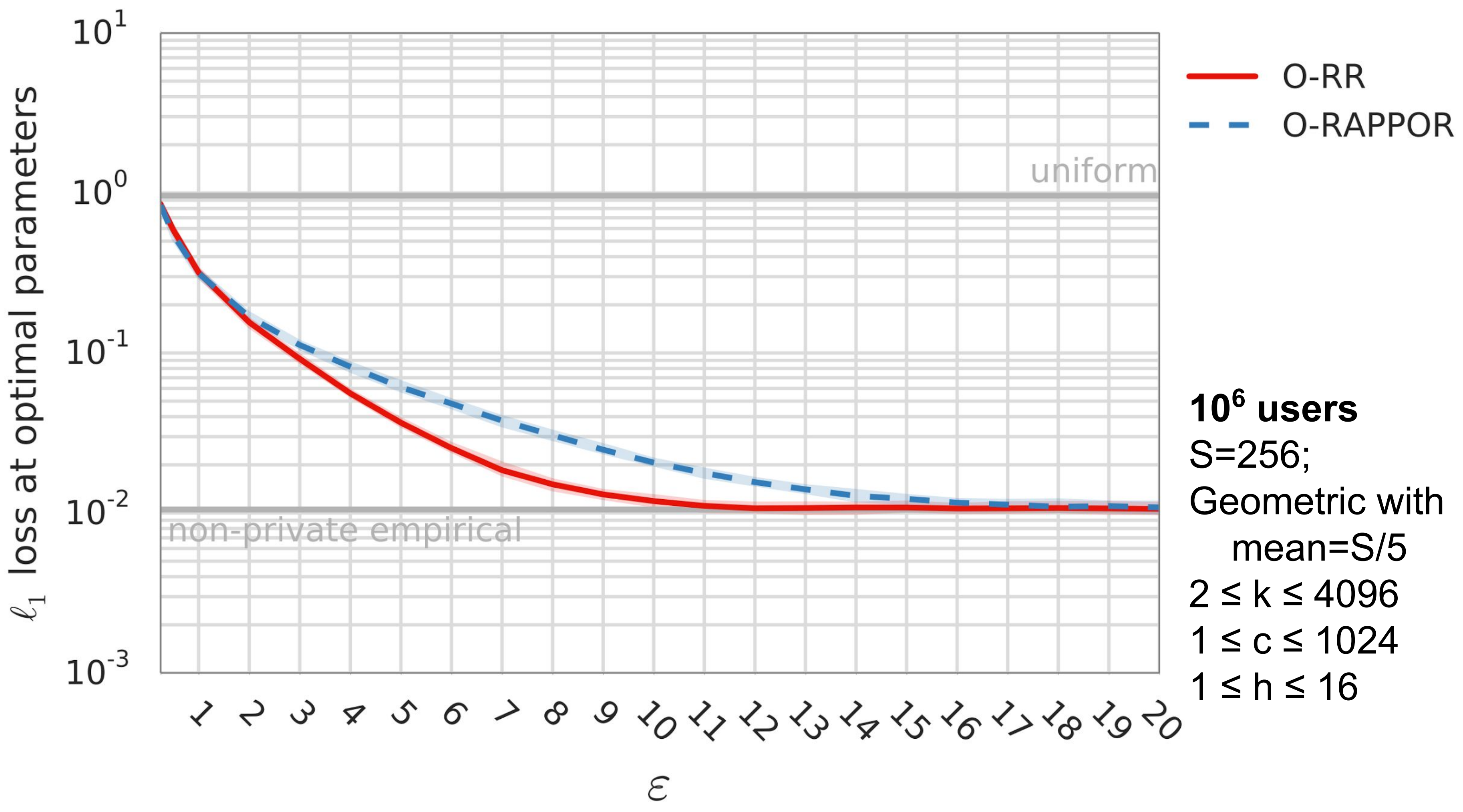

More bits in output: Bloom filter!

# O-RAPPOR

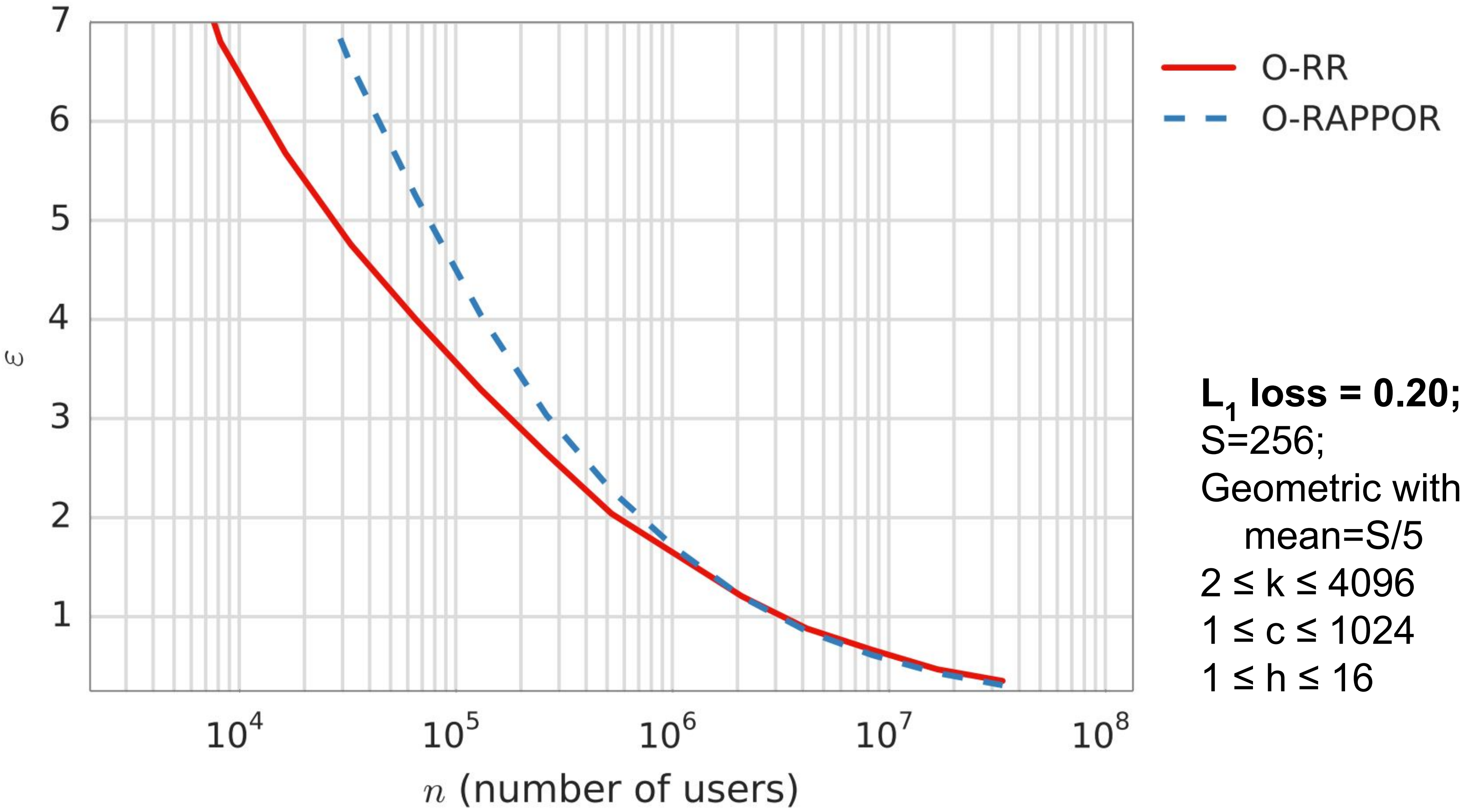


x
Uniform[C]
Stable for each user
$Hash_{0,c}^{(k)}$
$Hash_{1,c}^{(k)}$
0
1
0
1
0
$Q_{W-RR}$
$Q_{W-RR}$
$Q_{W-RR}$
$Q_{W-RR}$
$Q_{W-RR}$
$y^0$
$y^1$
$y^2$
$y^3$
$y^4$
C

# Empirical Comparison



O-RR
O-RAPPOR
uniform
non-private empirical
$\ell_1$ loss at optimal parameters
$\varepsilon$
$10^1$
$10^0$
$10^{-1}$
$10^{-2}$
$10^{-3}$
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
**$10^6$ users**
S=256;
Geometric with
mean=S/5
2 ≤ k ≤ 4096
1 ≤ c ≤ 1024
1 ≤ h ≤ 16



O-RR
O-RAPPOR
ε
$n$ (number of users)
$10^4$
$10^5$
$10^6$
$10^7$
$10^8$
1
2
3
4
5
6
7
**$L_1$ loss = 0.20;**
S=256;
Geometric with
mean=S/5
2 ≤ k ≤ 4096
1 ≤ c ≤ 1024
1 ≤ h ≤ 16

**O-RR meets or exceeds utility of O-RAPPOR** over wide range of privacy settings.

# Closed Alphabets, revisited

# Minimal Perfect Hash Functions

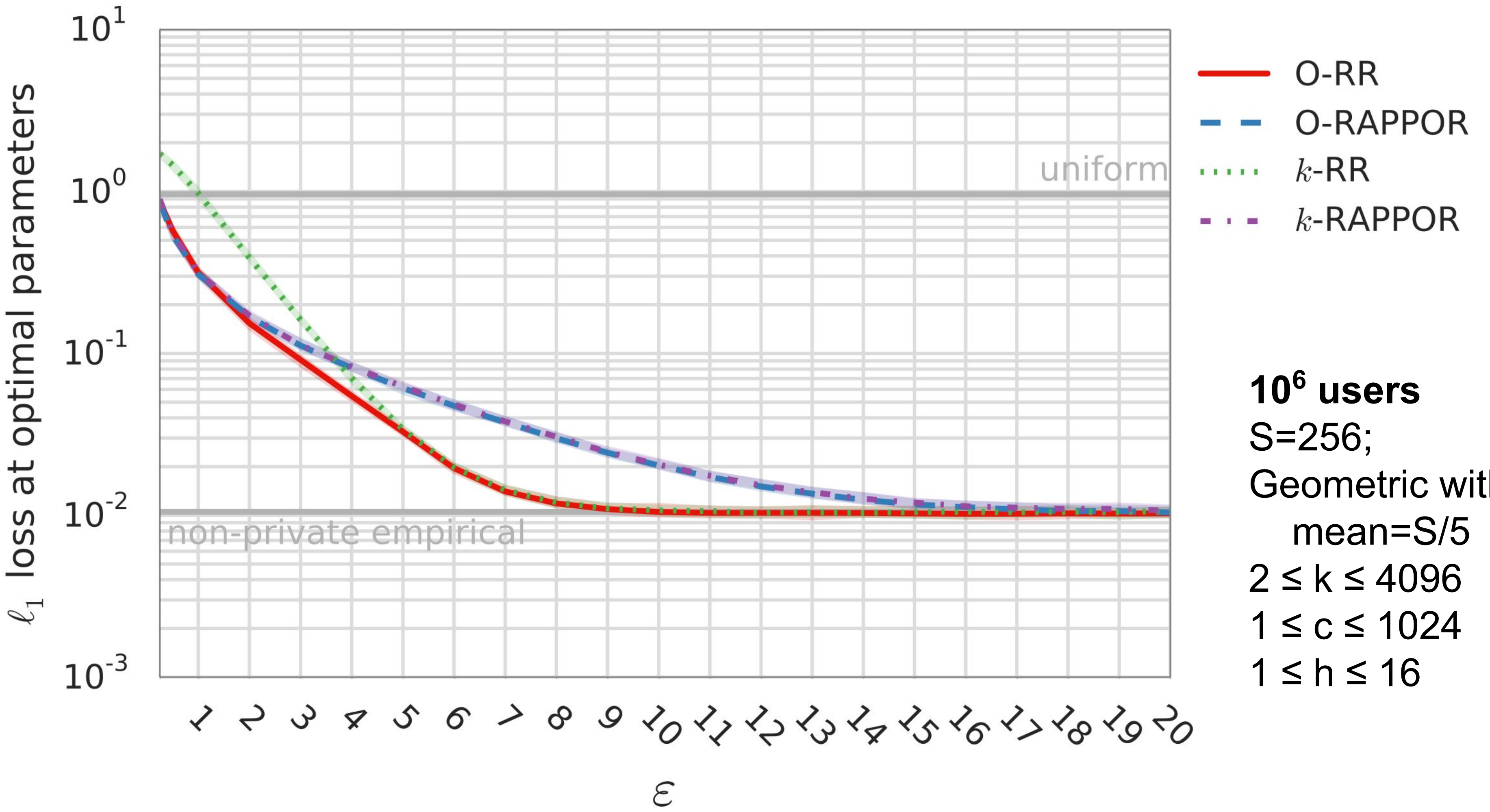
A **Minimal Perfect Hash Function** maps m keys to m consecutive integers.

If the m keys are the same set of consecutive integers, this is just a permutation.

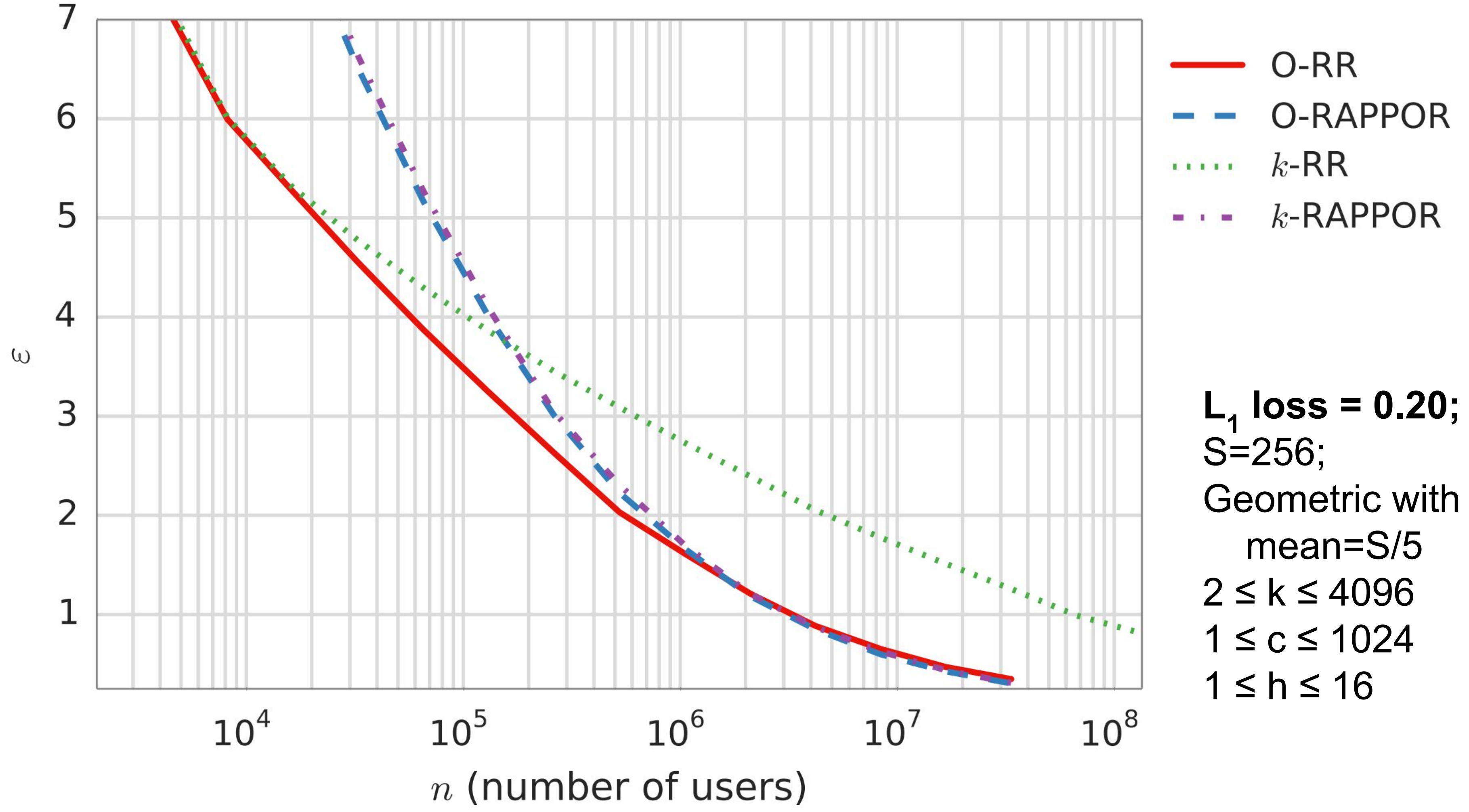
# Minimal Perfect Hash Functions

**For Closed Sets:** Modify O-RR and O-RAPPOR to use Minimal Perfect Hash Functions.

Note that with C=1 and h=1, we recover k-RR and k-RAPPOR (modulo a permutation of the output symbols).




**$10^6$ users**
S=256;
Geometric with
mean=S/5
$2 \leq k \leq 4096$
$1 \leq c \leq 1024$
$1 \leq h \leq 16$

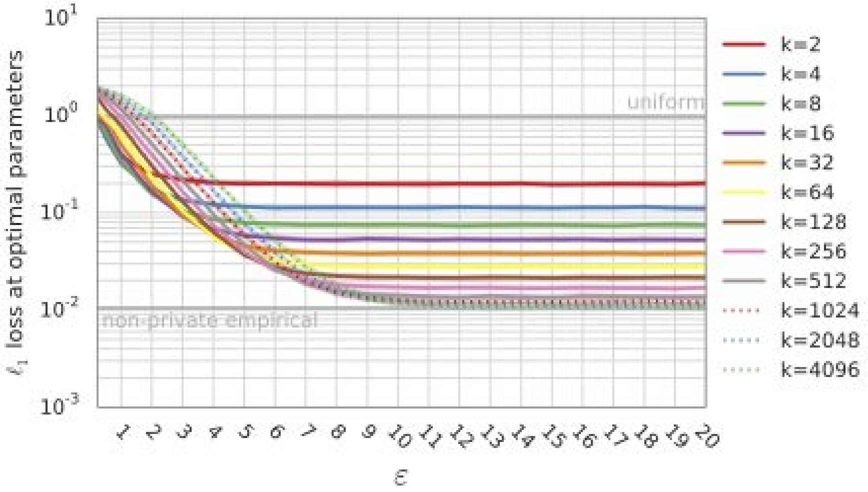


O-RR
O-RAPPOR
$k$-RR
$k$-RAPPOR
$\varepsilon$
$n$ (number of users)
$10^4$
$10^5$
$10^6$
$10^7$
$10^8$
1
2
3
4
5
6
7
**$L_1$ loss = 0.20;**
S=256;
Geometric with mean=S/5
2 ≤ k ≤ 4096
1 ≤ c ≤ 1024
1 ≤ h ≤ 16

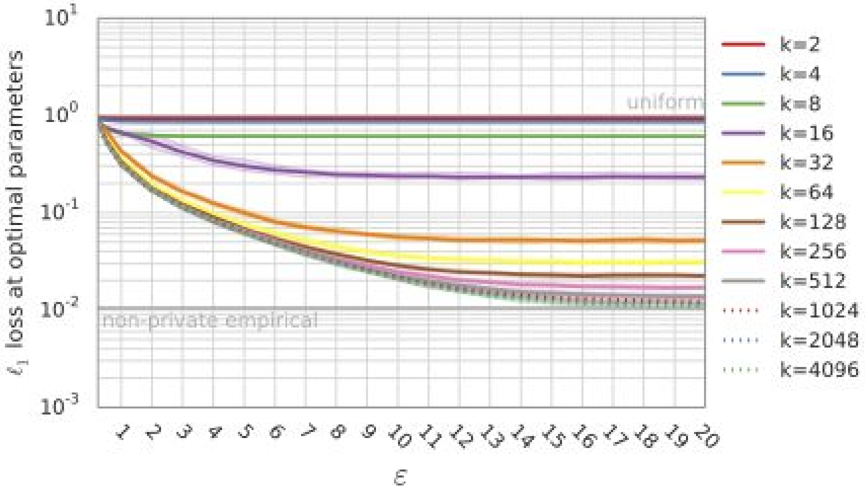
**O-RR meets or exceeds utility of O-RAPPOR** over wide range of privacy settings (for k-ary alphabets)

# Understanding Parameters

# **Open Set Decoding:** Output Alphabet Size

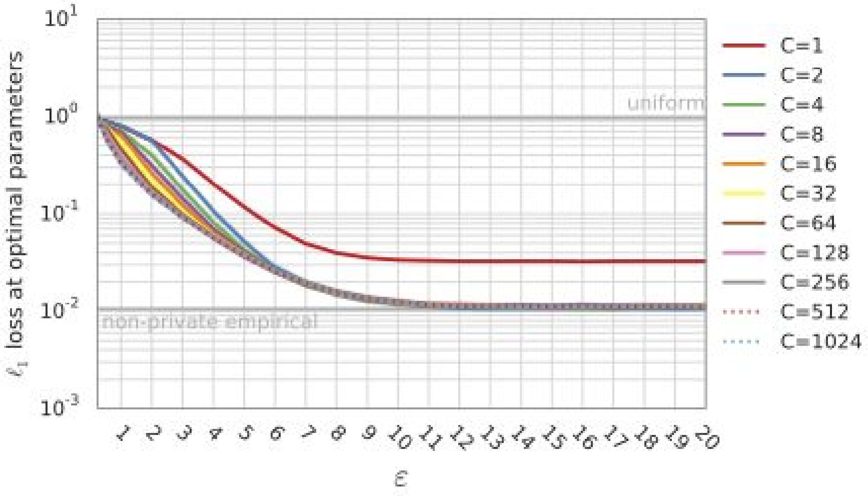



(a) O-RR varying $k$

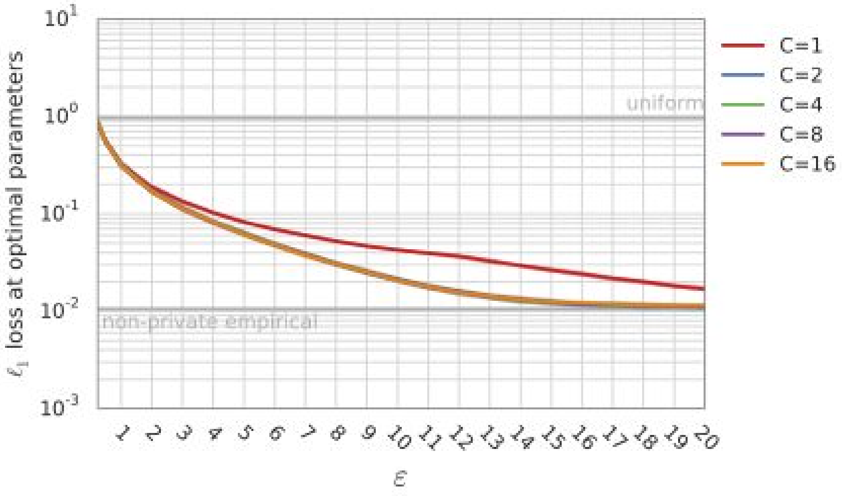



(b) O-RAPPOR varying $k$

# **Open Set Decoding:** # Cohorts

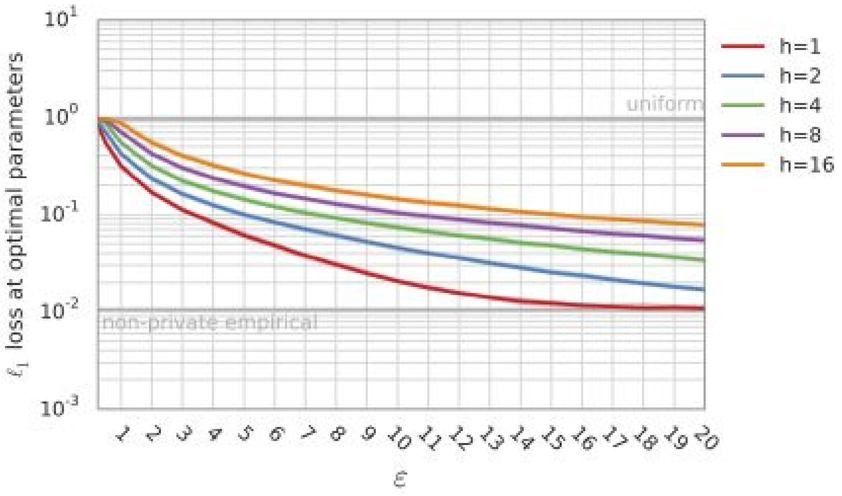



(c) O-RR varying $C$




(d) O-RAPPOR varying $C$

# **Open Set Decoding:** # Hashes in Bloom Filter

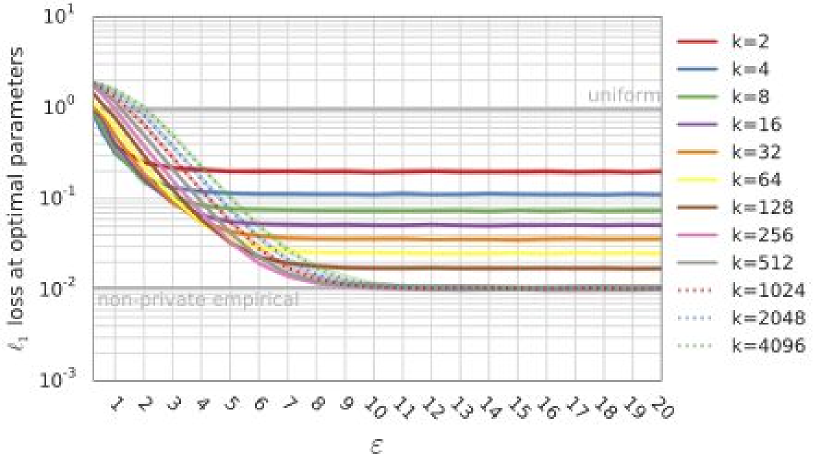



(e) O-RAPPOR varying $h$

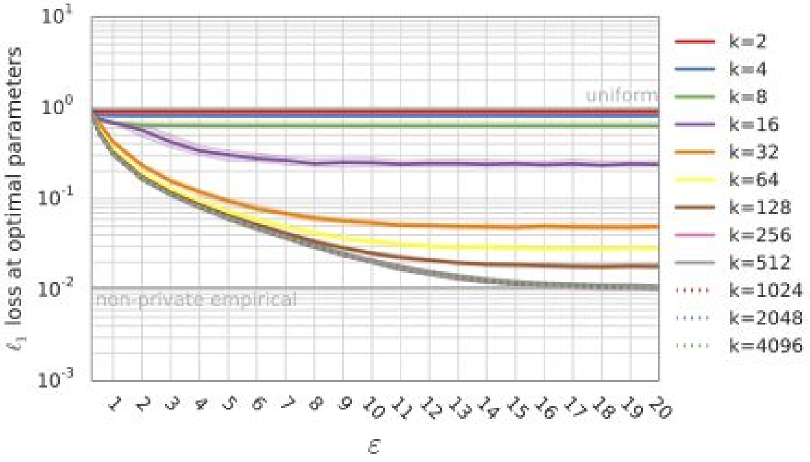
**O-RR (open):**
Alphabet size should match expected input size.
Cohorts matter more for high privacy, but always $\geq 2$.

**O-RAPPOR (open)**:
Bloom Filters don't help.
Use 2 cohorts and make the alphabet large.

# **Closed Set Decoding:** Output Alphabet Size

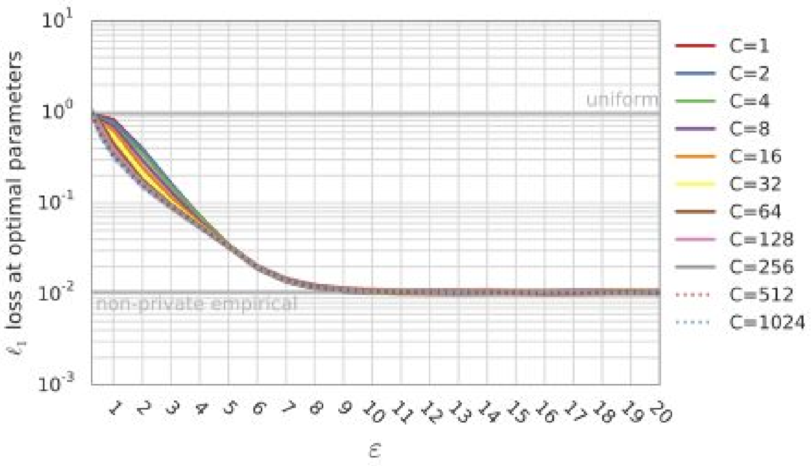



(a) O-RR varying $k$

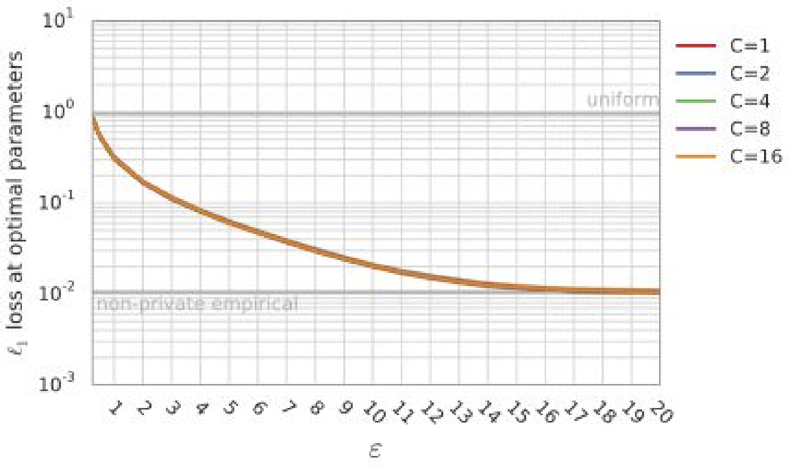



(b) O-RAPPOR varying $k$

# **Closed Set Decoding:** # Cohorts

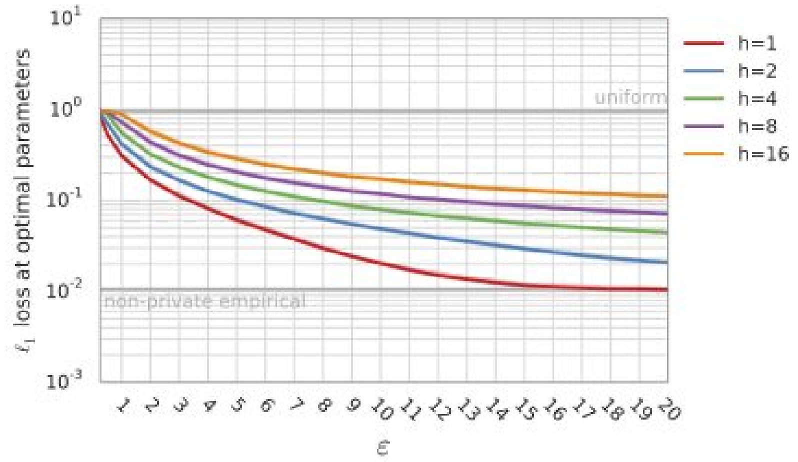



(c) O-RR varying $C$




(d) O-RAPPOR varying $C$

# **Closed Set Decoding:** # Hashes in Bloom Filter




(e) O-RAPPOR varying $h$

**O-RR (closed):**
Alphabet size should match expected input size.
Cohorts matter for high privacy.

**O-RAPPOR (closed)**: Bloom Filters and Cohorts don't help. Just use k-RAPPOR and make the alphabet large.