Discrete Distribution Estimation under Local Privacy

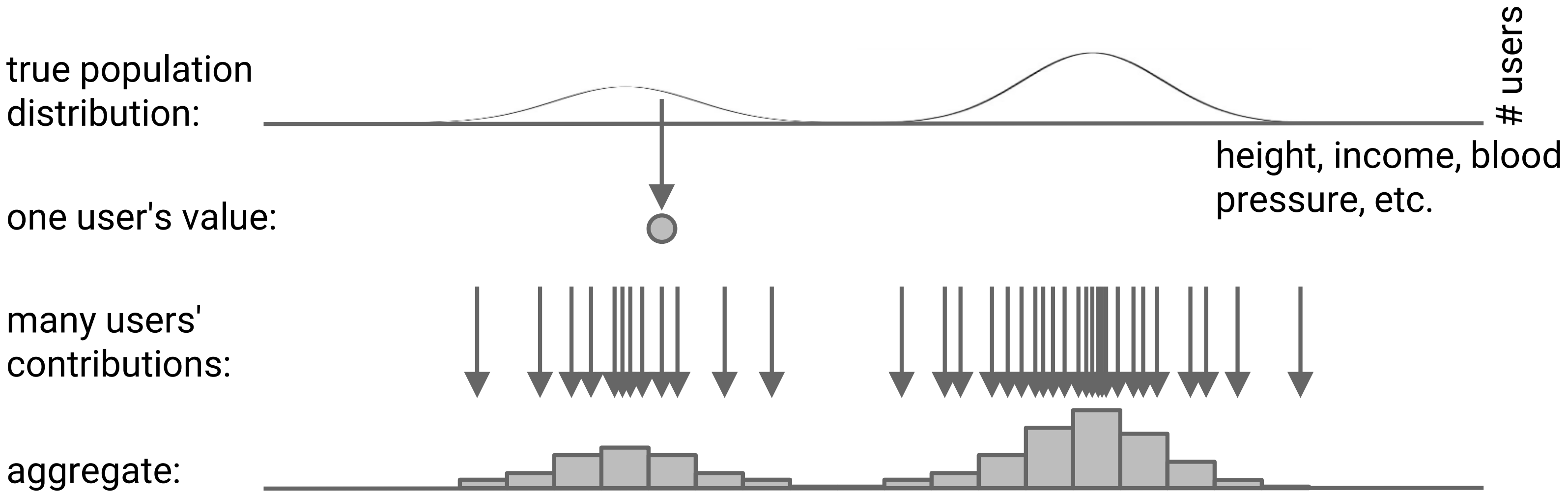
Peter Kairouz, Keith Bonawitz, Daniel Ramage

Distribution estimation

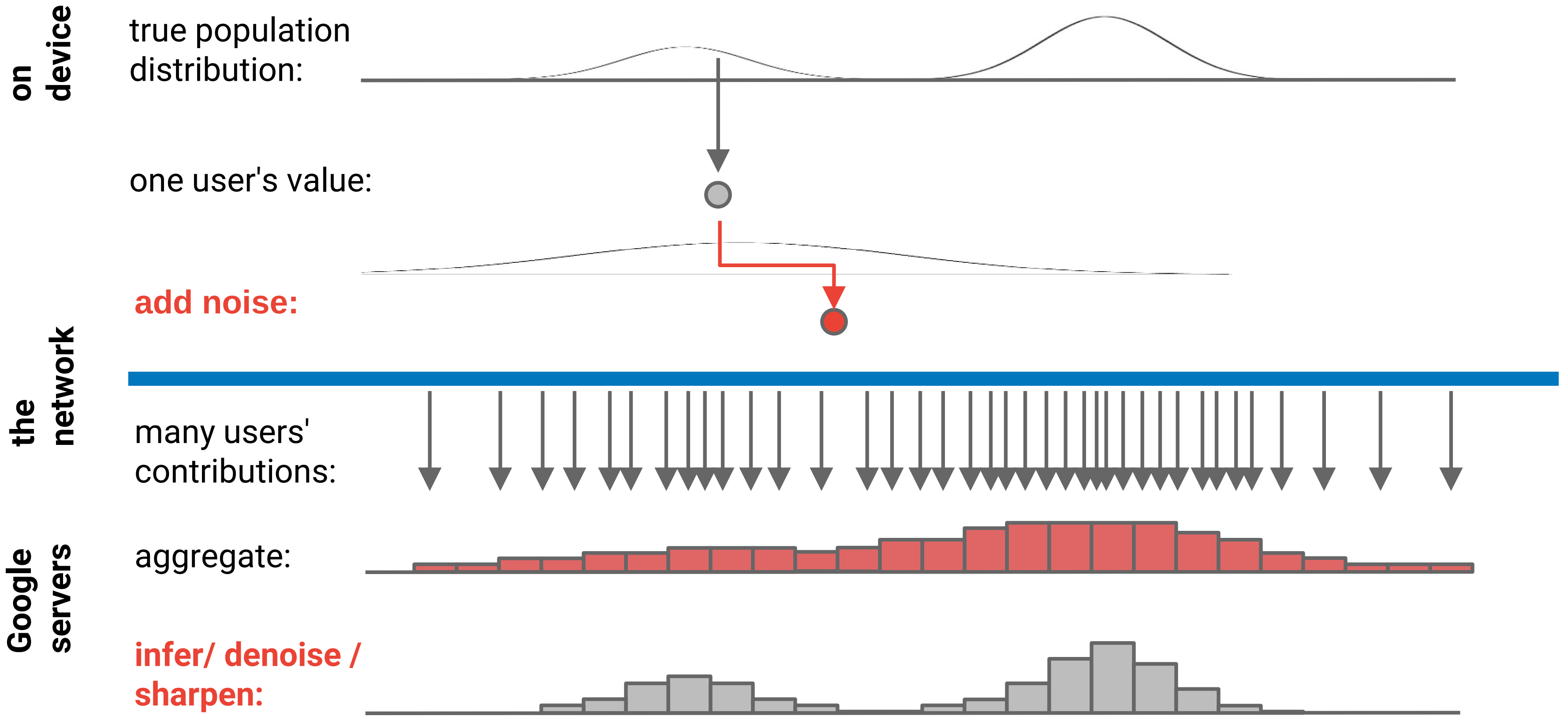
We need to understand **patterns across large groups** but **do not need to look at any individual**.

Distribution estimation

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

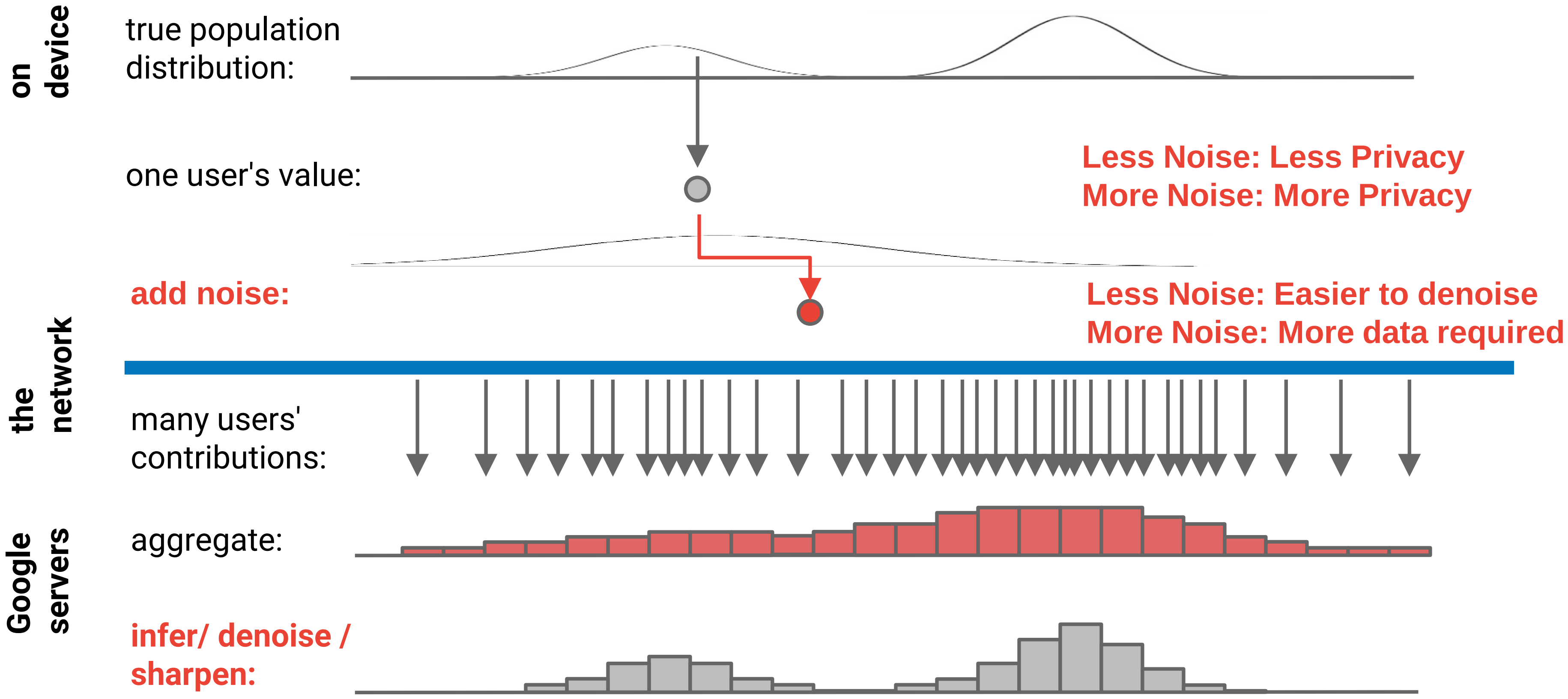


Private distribution estimation: add noise before logging



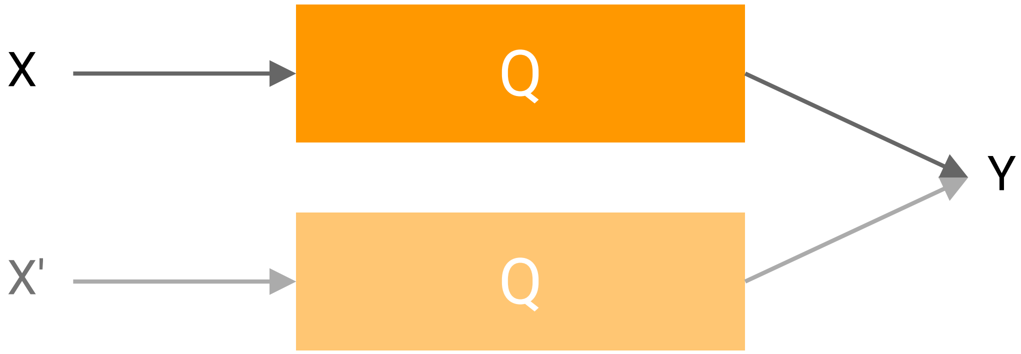
Google

Private distribution estimation: add noise before logging



Local Differential Privacy

Local differential privacy [Dwork 2006, Duchi et. al. 2012]



- If **true answer is x, say y with probability**: $Q(Y = y|X = x)$

- Q is locally differentially private if:

$$e^{-\varepsilon} \leq \frac{Q(Y=y|X=x)}{Q(Y=y|X=x')} \leq e^{\varepsilon}$$

Privacy utility tradeoff

$$\min_{Q} \min_{\hat{P}} \max_{P} \mathbb{E} d(P, \hat{P}(Q))$$

$$\text{subject to } Q \text{ locally differentially private}$$

Privacy utility tradeoff

$$\min_{Q} \min_{\hat{P}} \max_{P} \mathbb{E} d(P, \hat{P}(Q))$$

$$\text{subject to } Q \text{ locally differentially private}$$

What privacy mechanisms achieve the fundamental privacy-utility tradeoff for various privacy levels and alphabet sizes?

Binary alphabets

Warner's randomized response [Warner 1965]

"Have you ever used illegal drugs?"

answer **truthfully** w.p. $\frac{e^{\varepsilon}}{e^{\varepsilon}+1}$

lie w.p. $\frac{1}{e^{\varepsilon}+1}$

W-RR offers
optimal utility for
binary alphabets.

k-ary Alphabets

Two different ways to extend to k-ary alphabets

1. Modify the mechanism
2. Modify the encoding

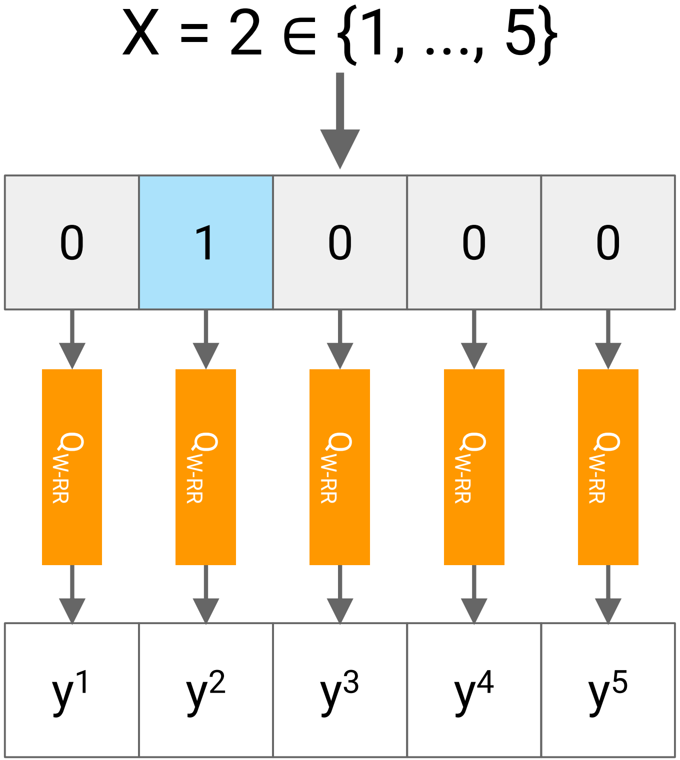
k-RR modifies the mechanism [Kairouz et. al. 2014]

answer **truthfully** w.p. $\dfrac{e^{\varepsilon}}{e^{\varepsilon}+k-1}$

lie w.p. $\dfrac{k-1}{e^{\varepsilon}+k-1}$

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

k-RAPPOR modifies the encoding [Erlingsson et. al. 2014]



2 bits different between any X, X'

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

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

Utility (sample complexity)

	ε ≈ ln(k) (Low Privacy)	Small ε (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**	$n/\sqrt{k}$	$n\varepsilon^2/4k$	$n\left(\frac{(k-1)^2(e^{\varepsilon/2}-1)^2}{(k-1)(e^{\varepsilon/2}-1)^2 + k^2 e^\varepsilon}\right)$

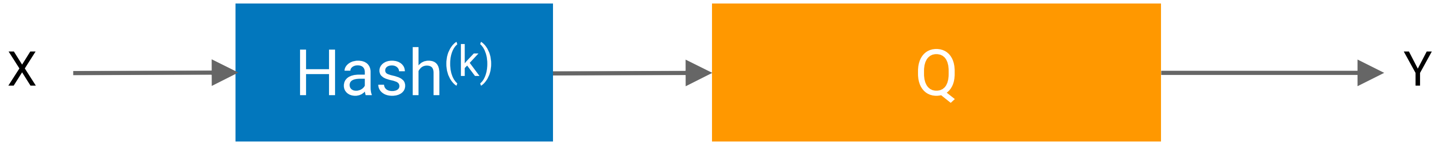
Open alphabets

Open alphabets

1. What if we don't know the set of input symbols ahead of time?
2. Can we avoid penalties for having large k?

Hashing (Sketches)

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

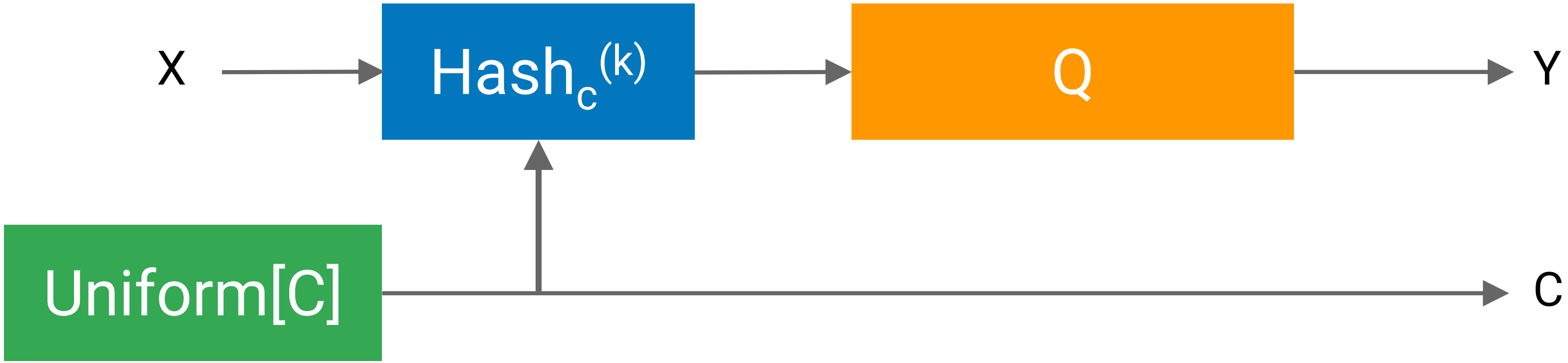


Hashing (Sketches)

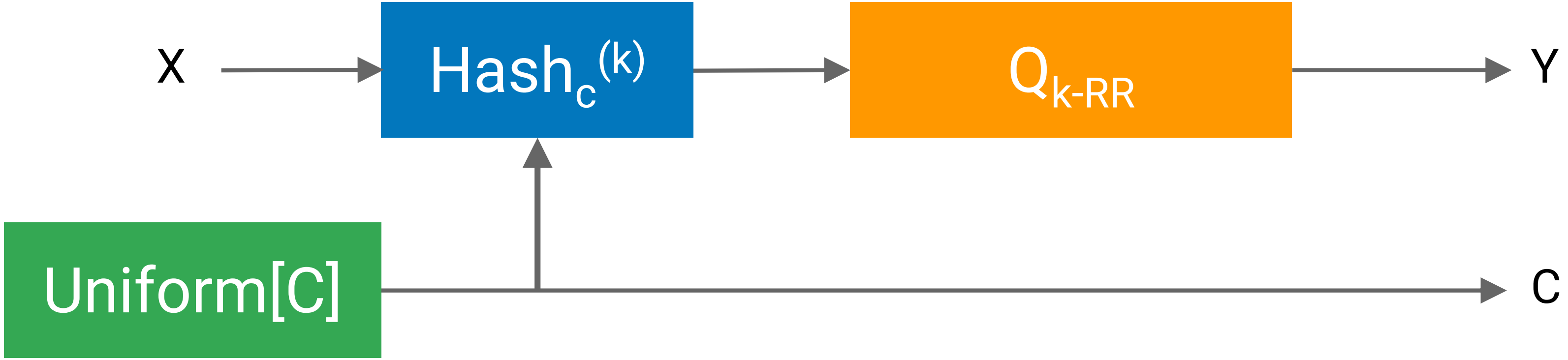
Instead of encoding x directly, we encode hash(x) mod k.

But what about collisions?

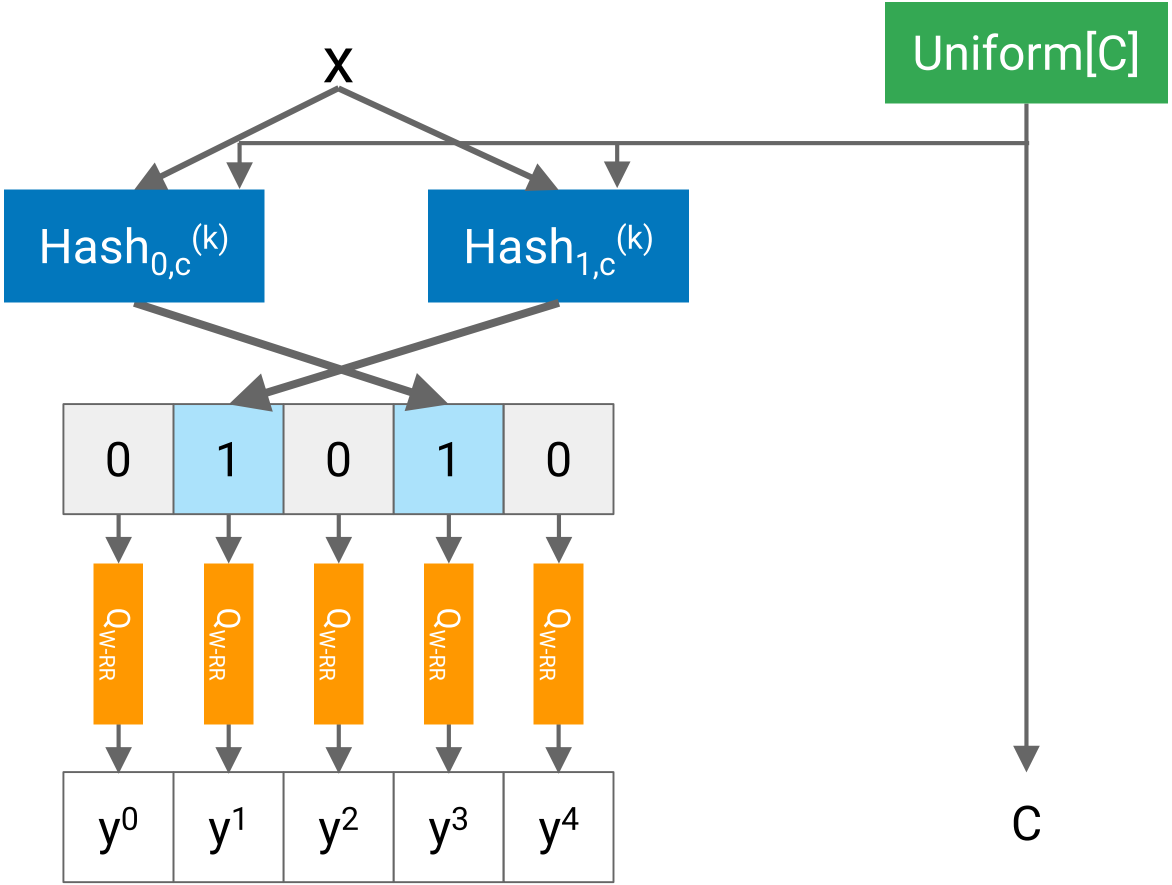
Multiple Hash Functions → Independent Views (Sketches)

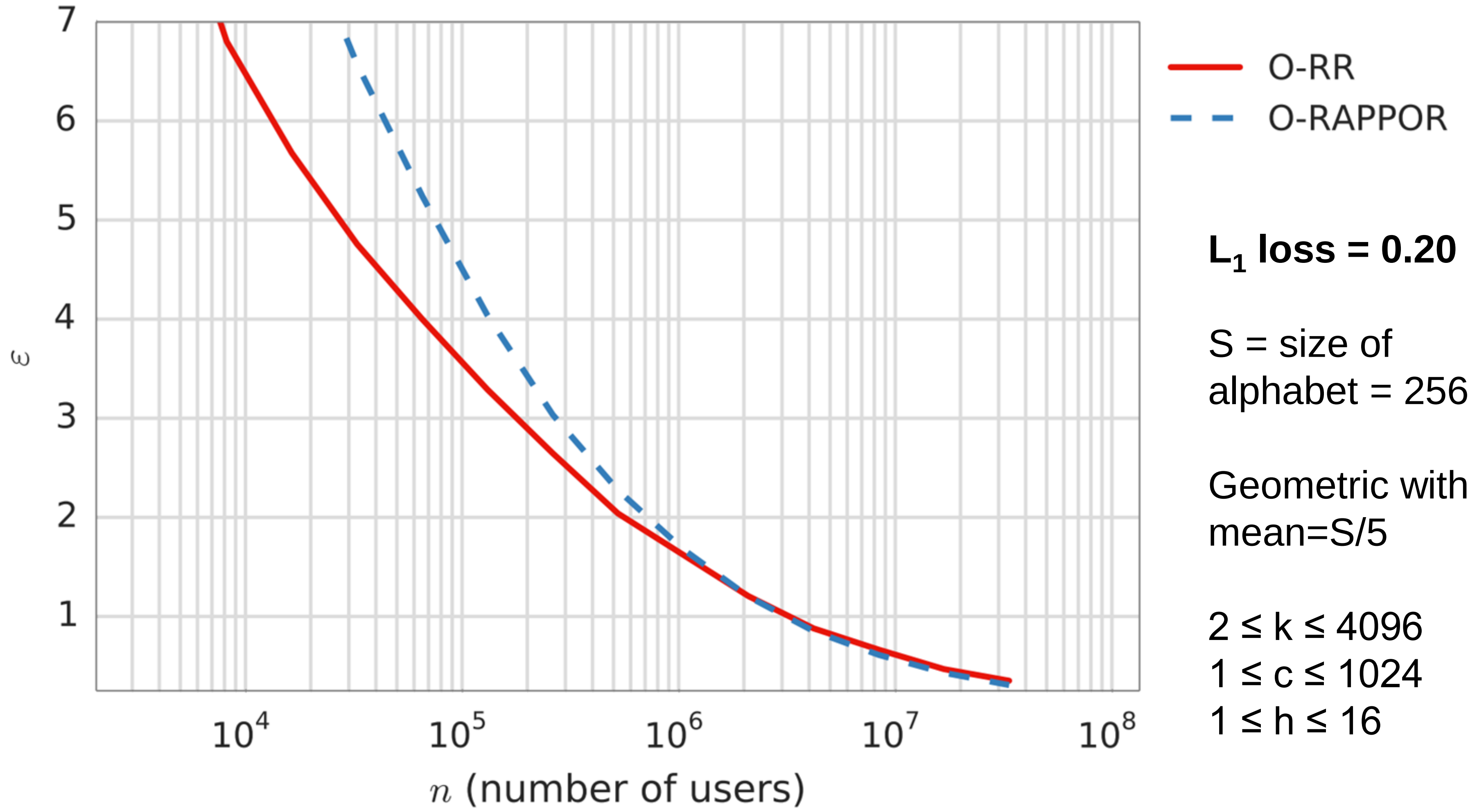


O-RR



O-RAPPOR [Erlingsson et. al. 2014]





O-RR
O-RAPPOR
ε
n (number of users)
L1 loss = 0.20
S = size of alphabet = 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.

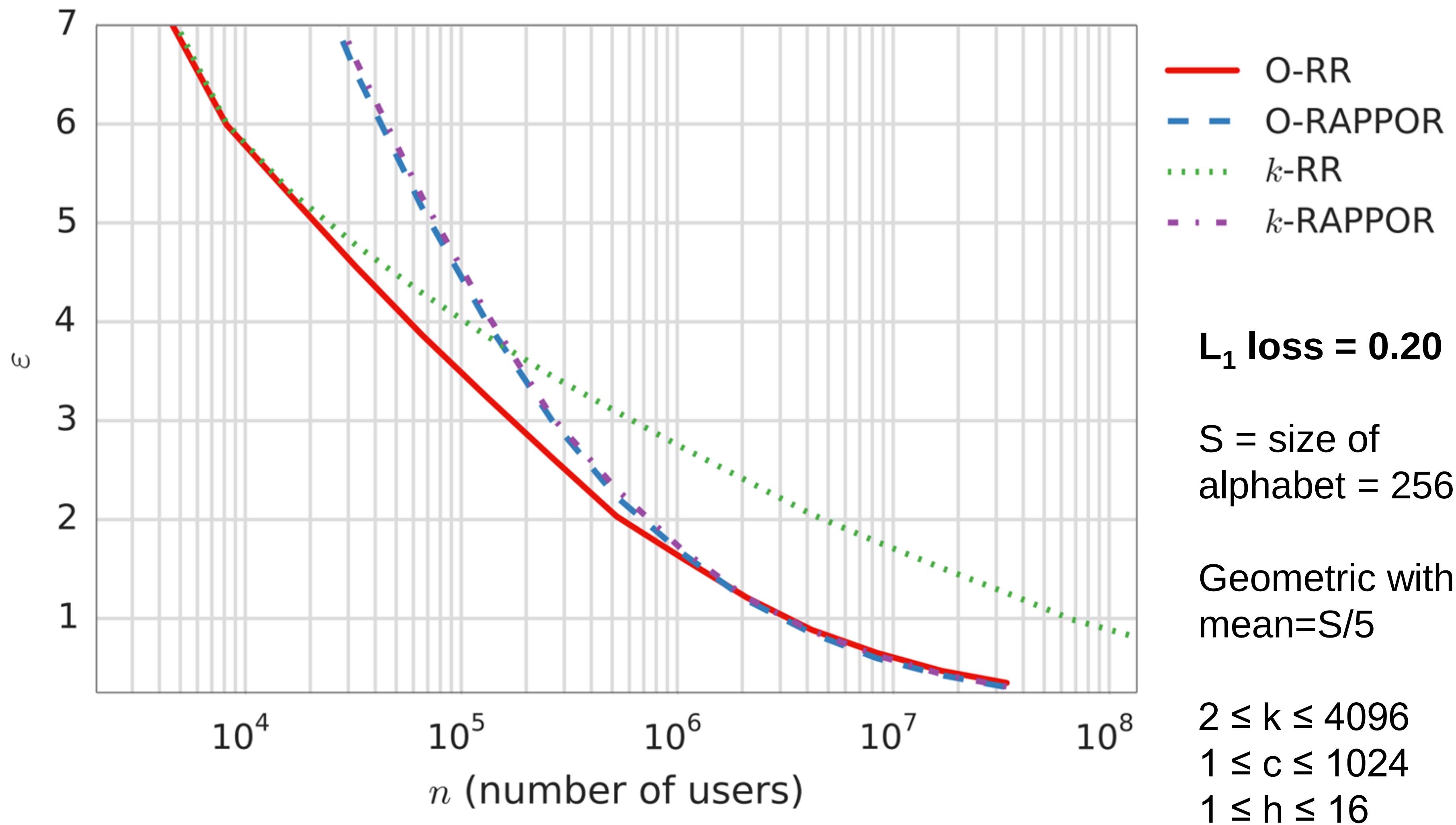
Closed Alphabets, revisited

Minimal perfect hash functions

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

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).



L_1 loss = 0.20

S = size of alphabet = 256

Geometric with mean=S/5

$2 \le k \le 4096$
$1 \le c \le 1024$
$1 \le h \le 16$

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

Thank you!