A Unified View of Frequency Estimation and their Attacks on Local Differential Privacy

Al Mehdi Saadat Chowdhury, Dhaval Pankaj Tanna, Deepak Vellanki, Chirag Manjeshwar
School of Computing and Augmented Intelligence
Arizona State University

Outline

Introduction:
- Differential Privacy
- Local Differential Privacy
- Pure LDP Framework
- Attack Problem

Frequency Estimation Techniques:
- RAPPOR
- K Randomized Response (kRR)
- Optimized Unary Encoding (OUE)
- Optimized Local Hashing (OLH)

Attacks:
- General Attack Formulation
- Attacking kRR
- Attacking OUE
- Attacking OLH

Evaluation:
- Comparison between Estimators
- Gain from Attacks
- Impacts of Parameters on Attacks

Conclusion:

Introductory Concepts

Differential Privacy

Dataset, $\mathcal{D}$	
Features	$x_1, x_2, \dots, x_n$
Labels	$y_1, y_2, \dots, y_m$

Subset of Labels	
S	y_1, y_4, y_{11}

$$\Pr\left[\mathrm{M}\left(\begin{array}{c} \boldsymbol{x} \\ \hline sample_1 \\ sample_2 \\ sample_3 \\ sample_4 \\ sample_5 \end{array} \right) \in \mathrm{S} \right] \leq e^{\epsilon} \Pr\left[\mathrm{M}\left(\begin{array}{c} \boldsymbol{y} \\ \hline sample_1 \\ sample_2 \\ sample_3 \\ sample_4 \\ sample_{11} \end{array} \right) \in \mathrm{S} \right] + \delta$$

Local Differential Privacy

$$\Pr\left[\mathrm{M}\left(\begin{array}{|c|}\hline x \\ \hline sample_1 \\ \hline\end{array}\right) \in \mathrm{y}\right] \leq e^{\epsilon} \Pr\left[\mathrm{M}\left(\begin{array}{|c|}\hline y \\ \hline sample_{11} \\ \hline\end{array}\right) \in \mathrm{y}\right]$$

Protocols for Local Differential Privacy

Encode

Rice
Bean
Soda

Rice	Bean	Eggs	Soda	Oreo	Taco
1	1	0	1	0	0

Perturb

1	1	0	1	0	0

1	0	0	1	1	0

Aggregate

1	0	0	1	1	0
1	0	1	1	0	1
0	0	0	1	1	0

2	0	1	3	2	1

Frequency Estimation Problem

Pure Differential Privacy

- A unified framework --- Can be used to define all frequency protocols.

- Defined based on two fixed probabilities p^* and q^*

$$\Pr[PE(v_1) \in \{y \,|v_1 \in Support(y)\}] = p^*$$
$$\Pr[PE(v_2) \in \{y \,|v_1 \in Support(y)\}] = q^*$$

- Based on this framework, we will describe 4 protocols – RAPPOR, kRR, OUE, OLH.

Attacking Estimators – Problem Formulation

- Assume: $n\ real\ users$
- Inject: $m\ fake\ users$
- To increase frequency of r items: $T = \{t_1, t_2, \dots, t_r\}$

- Goal:

$$\Delta\tilde{f}_t = \tilde{f}_{t,after} - \tilde{f}_{t,before} \qquad \forall t \in T$$

$$\max_{Y} \sum_{\{t \in \mathrm{T}\}} \mathbb{E}\left[\Delta\tilde{f}_t\right]$$

Foundational Work

Google's RAPPOR

Encode

$\mathcal{H} = \{H_1, H_2, \ldots, H_m\}$

0	1	0	0	0	0

$B_0[i] = 1 \quad if\ \exists H \in \mathcal{H}, s.t., H(v) = i$

Perturb

0	1	0	0	0	0

0	1	0	0	1	0

$$\Pr[B_1[i]] = 1 = \begin{cases} p = 1 - \frac{f}{2} & if\ B_0[i] = 1 \\ q = \frac{f}{2} & if\ B_0[i] = 0 \end{cases}$$

Aggregate

- Using Linear Regression
- Using LASSO Regression

Limitations of RAPPOR

- Use of Bloom filter reduced communication cost

However,

- Accuracy decreased significantly
- Computation cost of the aggregation step increased significantly

SOTA Frequency Estimators

K Randomized Response

Encode

A	B	C	D	E	F

A	B	C	D	E	F

Perturb

A	B	C	D	E	F

A	D	C	C	F	E

$$\Pr[PE(v) = i] = \begin{cases} p = \dfrac{e^{\epsilon}}{e^{\epsilon} + d - 1} & if\ i = v \\ q = \dfrac{1}{e^{\epsilon} + d - 1} & otherwise \end{cases}$$

Aggregate

$$\tilde{f}_v = \frac{\frac{1}{n}\sum_{\{i=1\}}^{n} \mathbb{I}_{\{Support(y_i)\}}(v) - q^*}{p^* - q^*}$$

For kRR,

$$Support(y_i) = \{y\}$$

Optimal Unary Encoding

Encode

Item

$\downarrow$

0	0	1	0	0	0

Perturb

0	0	1	0	0	0

$\downarrow$

0	0	1	0	1	0

$$\Pr[PE(v) = 1] = \begin{cases} p = \frac{1}{2} & if\ i = v \\ q = \frac{1}{e^{\epsilon} + 1} & otherwise \end{cases}$$

Aggregate

$$\tilde{f}_v = \frac{\frac{1}{n}\sum_{\{i=1\}}^{n} \mathbb{I}_{\{Support(y_i)\}}(v) - q^*}{p^* - q^*}$$

For OUE,

$$Support(y_i) = \{v \mid v \in [d]\ and\ y_v = 1\}$$

Optimal Local Hashing

Encode

$\mathcal{H} = \{H_1, H_2, \dots, H_m\}$

H_3

0	0	1	0	0	0

Perturb

0	0	1	0	0	0

0	0	0	0	0	0

$$\Pr[y = \langle H, x \rangle] = \begin{cases} p = \dfrac{e^{\epsilon}}{e^{\epsilon} + d - 1} & if\ x = i \\ q = \dfrac{1}{e^{\epsilon} + d - 1} & otherwise \end{cases}$$

Aggregate

$$\tilde{f}_v = \frac{\frac{1}{n} \sum_{\{i=1\}}^{n} \mathbb{I}_{\{Support(y_i)\}}(v) - q^*}{p^* - q^*}$$

For OLH,

$Support(y_i)$
$= \{v \mid v \in [d]\ and\ H(v) = x\}$

Maximum Gain Attacks on Frequency Estimators

Frequency Estimators Under Attack

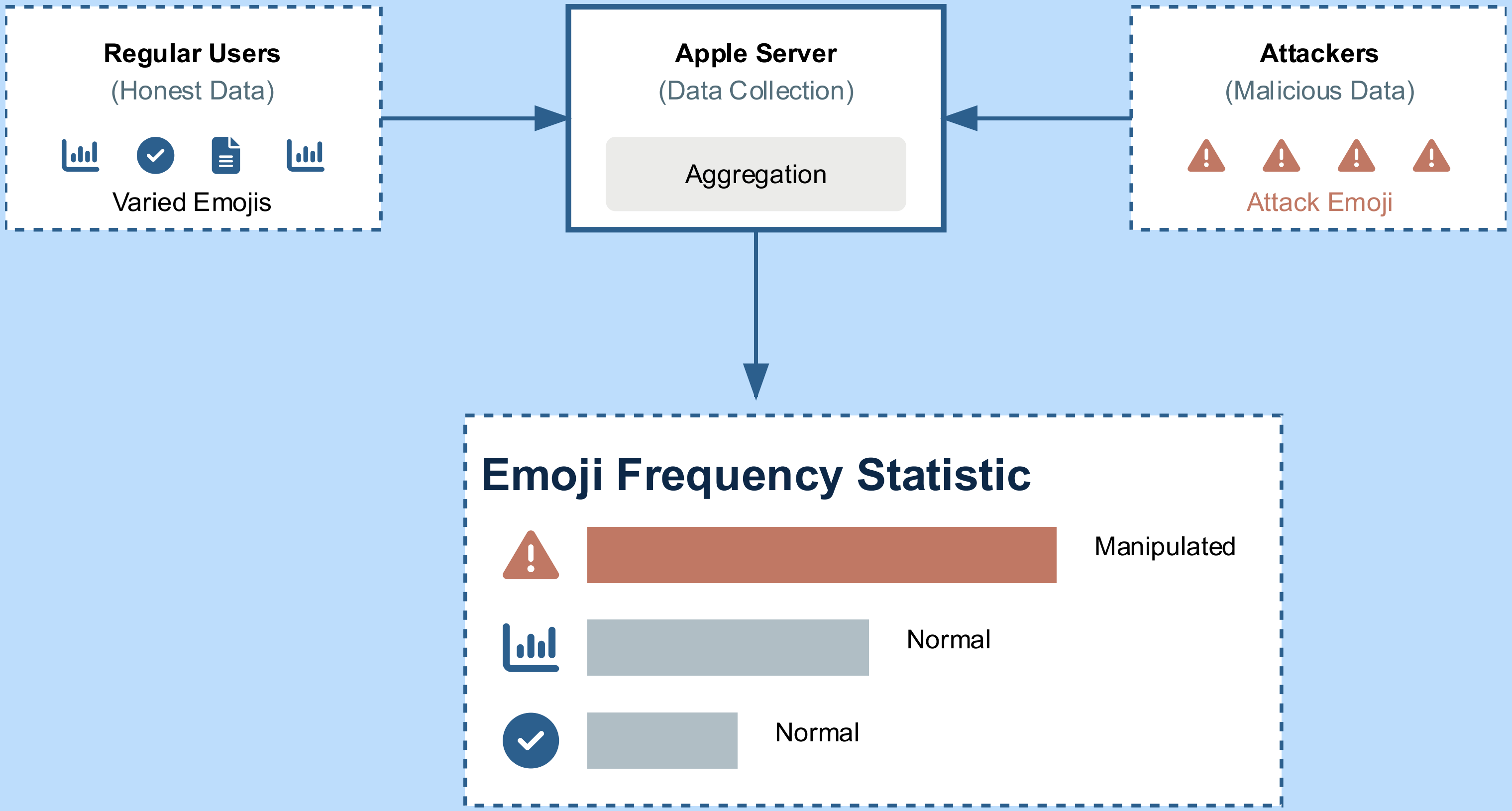


Illustration: Apple introduced LDP to estimate popular emojis in 2016

Frequency Gain from an Attack

General Formula:

$$G = \frac{\sum_{\{i=n+1\}}^{\{n+m\}} \mathbb{E}\left[\mathbb{I}_{Support(y_i)}(t)\right]}{(n+m)(p^* - q^*)} - \frac{m \sum_{\{i=1\}}^{\{n\}} \mathbb{E}\left[\mathbb{I}_{Support(y_i)}(t)\right]}{n(n+m)(p^* - q^*)}$$

(Accounts for Fake Users' Impact)

(Accounts for Genuine Users' Impact Dilution)

MGA tries to maximize this gain

Maximum Gain Attack: Overview

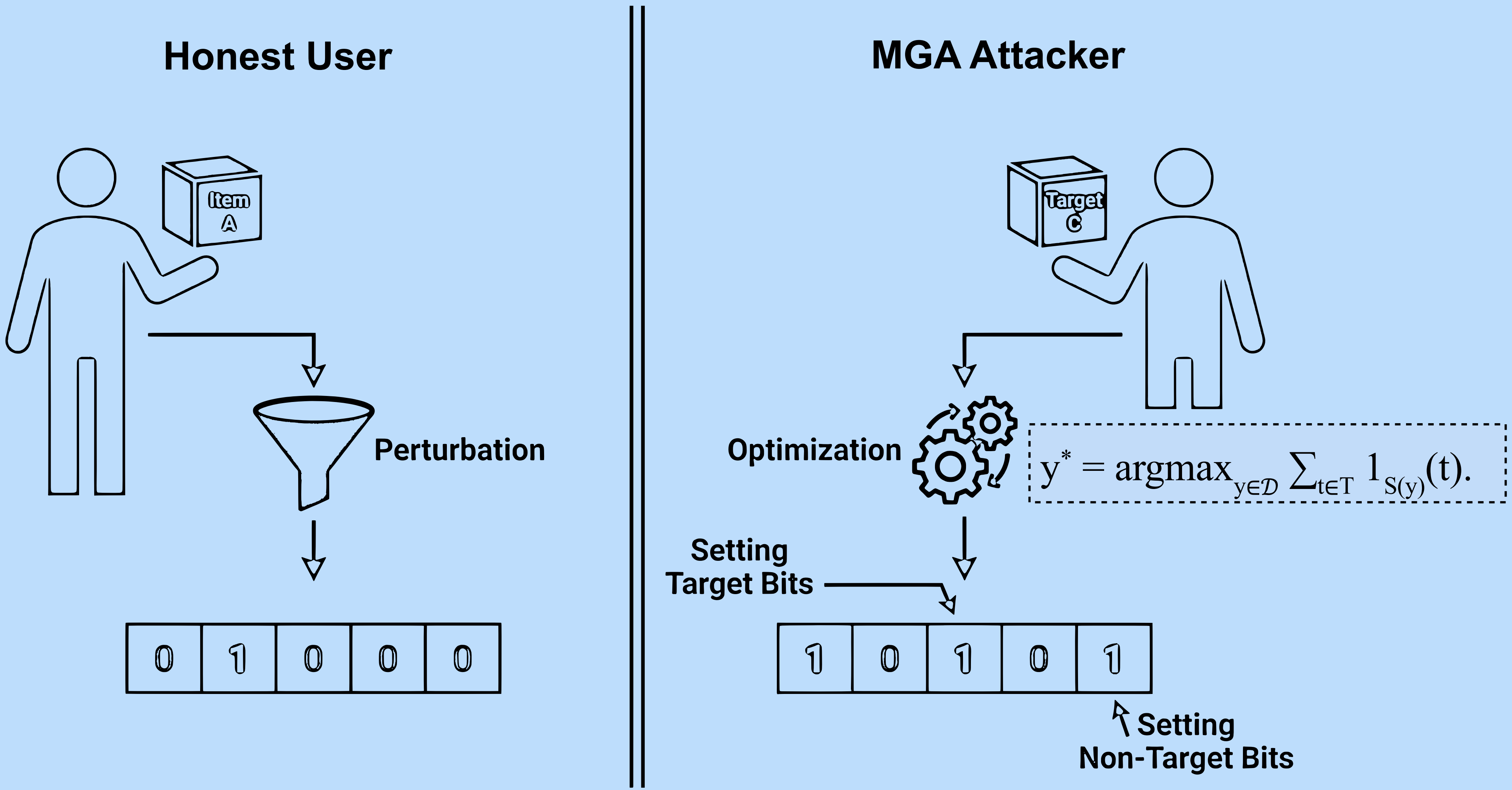


Illustration: MGA attack on the OUE protocol

Attacking kRR

- The support set is:

$$\sum_{\{t \in T\}} \mathbb{I}_{Support(y_i)}(t) = 1$$

- Gain becomes:

$$G = \frac{m}{(n+m)(p^* - q^*)} - c$$

- Plugging in p^* and q^* gives:

$$G = \beta(1 - f_T) + \frac{\beta(d-r)}{e^\epsilon - 1}$$

Attacking OUE

- The support set is:

$$\sum_{\{t \in T\}} \mathbb{I}_{Support(y_i)}(t) = r$$

- Gain becomes:

$$G = \frac{rm}{(n+m)(p^* - q^*)} - c$$

- Plugging in p^* and q^* gives:

$$G = \beta(2r - f_T) + \frac{2\beta r}{e^{\epsilon} - 1}$$

Attacking OLH

- The support set is:

$$\sum_{\{t \in T\}} \mathbb{I}_{Support(y_i)}(t) = r$$

- Gain becomes:

$$G = \frac{rm}{(n+m)(p^* - q^*)} - c$$

- Plugging in p^* and q^* gives:

$$G = \beta(2r - f_T) + \frac{2\beta r}{e^{\epsilon} - 1}$$

Evaluation

Estimators: Utility vs. Security

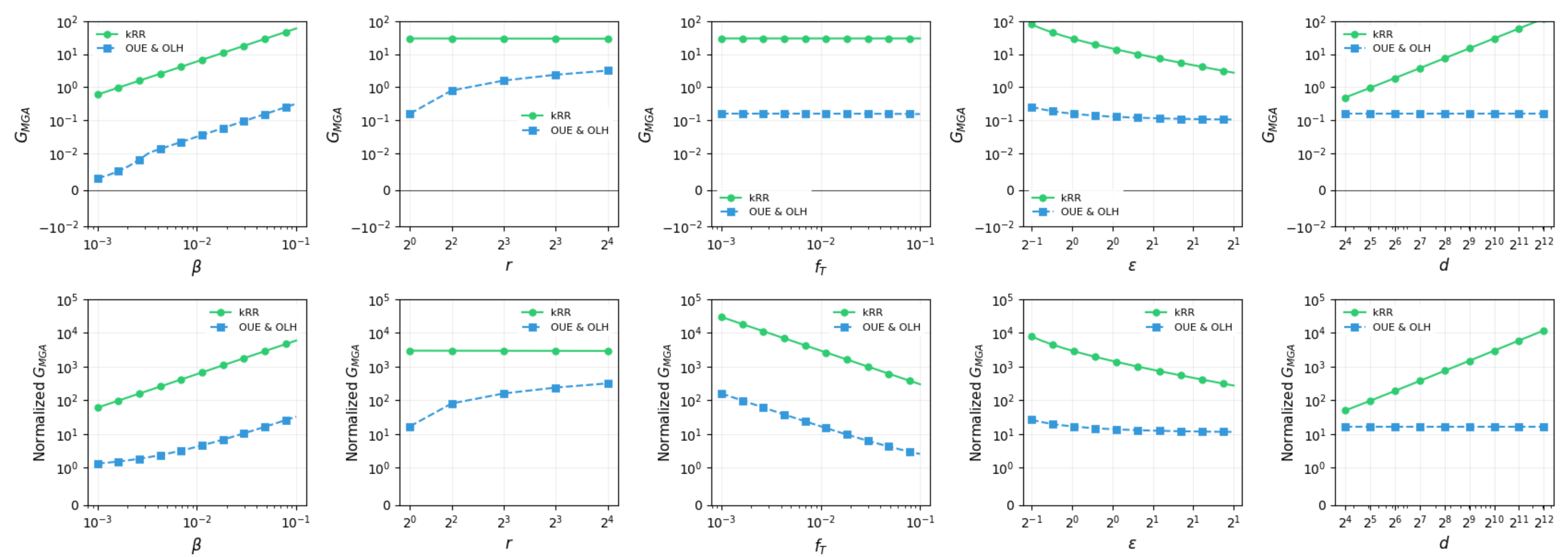
	kRR	OUE	OLH
Communication Cost	$\theta(\log d)$	$\theta(d)$	$\theta(\log n)$
Variance	$n \cdot \frac{d-2+e^{\epsilon}}{(e^{\epsilon}-1)^2}$	$n \cdot \frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}$	$n \cdot \frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}$
Gain of MGA	$\beta(1-f_T)+\frac{\beta(d-r)}{e^{\epsilon}-1}$	$\beta(2r-f_T)+\frac{2\beta r}{e^{\epsilon}-1}$	$\beta(2r-f_T)+\frac{2\beta r}{e^{\epsilon}-1}$

n = number of real users
m = number of fake users
$\beta = \frac{m}{n+m}$ = fraction of fake users

d = domain size
T = set of target items
$r = |T|$

$$f_T = \sum_{\{t \in T\}} f_t$$

MGA Comparison



Conclusion

Summary and Open Questions

- Different Frequency Estimation Protocols under LDP
- Attacks on Estimators
  - kRR performs poorly when the domain size increases
  - OUE, OLH perform poorly when the target items increases
- Open Question:
  - Large domain, Large target items
  - How to handle complex data types
  - Utility – Privacy trade-offs

Thank You