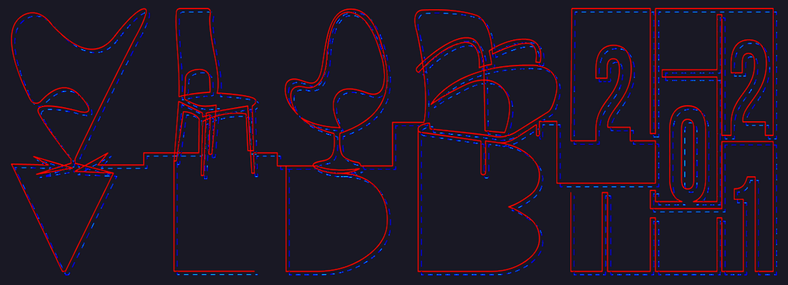


Frequency Estimation under Local Differential Privacy

[Experiments, Analysis and Benchmarks]

Authors: Graham Cormode, Samuel Maddock, Carsten Maple
University of Warwick

Presenter: Samuel Maddock

WARWICK
THE UNIVERSITY OF WARWICK

Differential Privacy (DP)

Key idea in all DP mechanisms is to add random noise to users data to ensure plausible deniability

Privacy Budget - ϵ

- Small epsilon (<1) guarantees more privacy
- Larger ϵ guarantee less privacy for individual

Focused on the local model **(LDP)**

- Clients perturb data locally and release to a central server
- Interested in privately estimating frequencies
- Industry deployments include Google's RAPPOR system and Apple's CMS/SFP

Definition 2.1 (Local differential privacy). The local randomizer $\mathcal{R}$ is ϵ-locally differentially private (ϵ-LDP) if for all pairs of inputs $x \in \mathcal{D}, x' \in \mathcal{D}$ and outputs $y \in \mathcal{Y}$ we have that $\frac{\Pr[\mathcal{R}(x)=y]}{\Pr[\mathcal{R}(x')=y]} \leq e^{\epsilon}$.

WARWICK
THE UNIVERSITY OF WARWICK

Outline

Main goal is to present an unbiased evaluation of frequency estimation and heavy hitter LDP methods

We consider a framework of four “layers”, analysing experimentally current state-of-the-art techniques

1 **Frequency Oracle (FOs)** + 2 **Domain Reduction** + 3 **Post-Processing** + 4 **Heavy Hitter Protocol**

WARWICK
THE UNIVERSITY OF WARWICK

Outline

Main goal is to present an unbiased evaluation of frequency estimation and heavy hitter LDP methods

We consider a framework of four “layers”, analysing experimentally current state-of-the-art techniques

1 **Frequency Oracle (FOs)** + 2 **Domain Reduction** + 3 **Post-Processing** + 4 **Heavy Hitter Protocol**

WARWICK
THE UNIVERSITY OF WARWICK

Outline

Main goal is to present an unbiased evaluation of frequency estimation and heavy hitter LDP methods

We consider a framework of four “layers”, analysing experimentally current state-of-the-art techniques

1 **Frequency Oracle (FOs)** + 2 **Domain Reduction** + 3 **Post-Processing** + 4 **Heavy Hitter Protocol**

$h_1(😊) = 2 \overset{FO}{\rightsquigarrow} 3$

$h_2(😊) = 4 \overset{FO}{\rightsquigarrow} 4$

$h_3(😊) = 1 \overset{FO}{\rightsquigarrow} 5$

$$\begin{pmatrix} ? & ? & +1 & ? & ? \\ ? & ? & ? & +1 & ? \\ ? & ? & ? & ? & +1 \end{pmatrix}$$

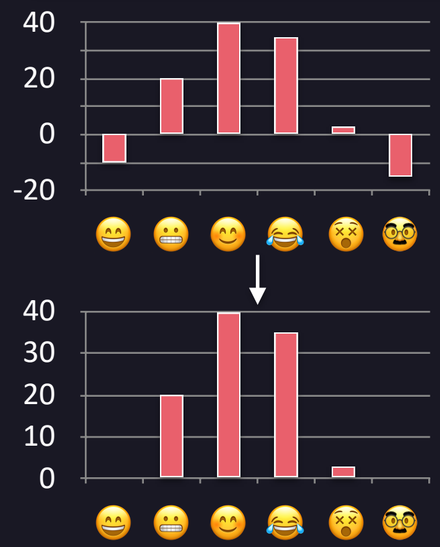
WARWICK
THE UNIVERSITY OF WARWICK

Outline

Main goal is to present an unbiased evaluation of frequency estimation and heavy hitter LDP methods

We consider a framework of four “layers”, analysing experimentally current state-of-the-art techniques

1 **Frequency Oracle (FOs)** + 2 **Domain Reduction** + 3 **Post-Processing** + 4 **Heavy Hitter Protocol**



WARWICK
THE UNIVERSITY OF WARWICK

Outline

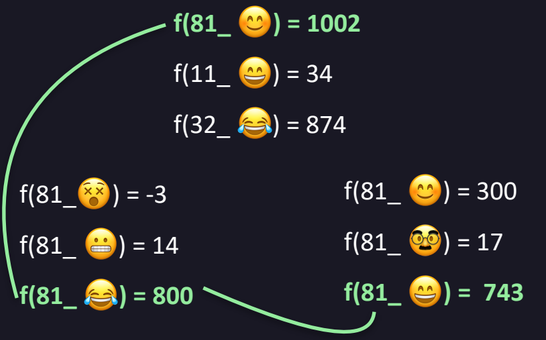
Main goal is to present an unbiased evaluation of frequency estimation and heavy hitter LDP methods

We consider a framework of four “layers”, analysing experimentally current state-of-the-art techniques

1 **Frequency Oracle (FOs)** + 2 **Domain Reduction** + 3 **Post-Processing** + 4 **Heavy Hitter Protocol**

String: 😊😂😄

h(😊😂😄) = 81



WARWICK
THE UNIVERSITY OF WARWICK

Experimental Setup

Consider n clients/users sending privatised data from a domain of size d

Interested in estimating frequencies over the domain:

- Mean Square Error (MSE) and k-MSE
- Precision, Recall and F1 score

Experiments were performed on both synthetic and real-world data:

- **Synthetic data:** Zipf distribution s=1.1 varying n and d depending on experiment

- **AOL Dataset:** Search query dataset containing clicked URLs
  - Each clicked URL to belong to a single user
  - Gives n=1,935,614 users with d=383,467 unique URLs

All experiments were implemented in Python 3.7.4
Code is publicly available at **https://github.com/Samuel-Maddock/pure-LDP**

Frequency Oracles

Direct Encoding (DE) - Basic extension of Randomised Response (RR) to more than 2 outputs

Unary Encoding (SUE, OUE) - Transform data via one-hot encoding, independently perturb entries

- **Symmetric Unary Encoding (SUE) [Erlingsson et al 2014]**
- **Optimised Unary Encoding (OUE) [Wang et al 2017]**

Local Hashing (BLH, OLH, FLH) - To remove the dependence on the domain size d, LH methods have users pick hash functions and perturb the hash of their data to satisfy LDP

- **Binary Local Hashing (BLH) [Bassily and Smith 2015]**
- **Optimised Local Hashing (OLH) [Wang et al 2017]**
- **Fast Local Hashing (FLH)** - Heuristic approach

FO	Variance bound	Decode time	Communication
DE	$O((e^\epsilon + d - 2)/(e^\epsilon - 1)^2)$	$O(n + d)$	$O(\log d)$
SUE	$O(e^{\epsilon/2}/(e^{\epsilon/2} - 1)^2)$	$O(nd)$	$O(d)$
OUE	$O(e^\epsilon/(e^\epsilon - 1)^2)$	$O(nd)$	$O(d)$
BLH	$O((e^\epsilon + 1)^2/(e^\epsilon - 1)^2)$	$O(nd)$	$O(\log d)$
OLH	$O(e^\epsilon/(e^\epsilon - 1)^2)$	$O(nd)$	$O(\log d)$
FLH	(heuristic)	$O(k'd)$	$O(\log d)$
HM	$O((e^\epsilon + 1)^2/(e^\epsilon - 1)^2)$	$O(n + d)$	$O(\log d)$
HR	$O(e^\epsilon/(e^\epsilon - 1)^2)$	$O(n + d)$	$O(\log d)$

Hadamard Encodings (HM, HR) - Use of the Hadamard Transform to speed up server-side aggregation

- **Hadamard Mechanism (HM)**
- **Hadamard Response (HR) [Acharya et al 2018]**

WARWICK
THE UNIVERSITY OF WARWICK

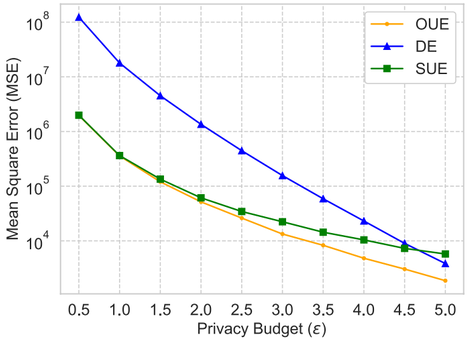
Frequency Oracles

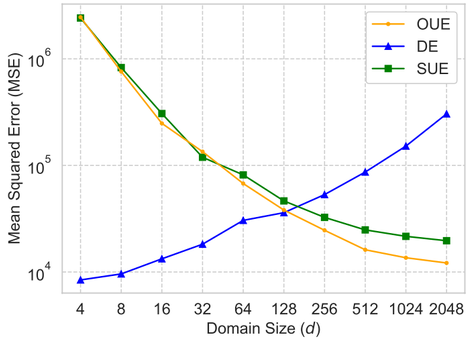
Comparison of Oracles

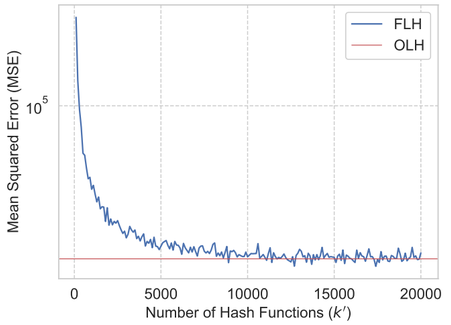
Unfortunately too many experiments to go through all here

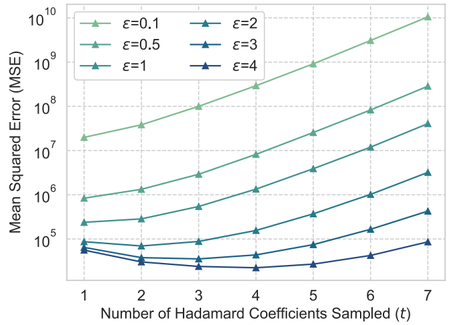
To begin we proceeded in a systematic fashion:

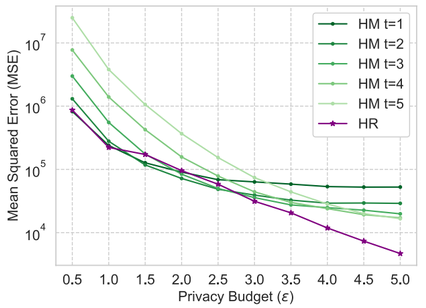
- UE vs DE
- Local Hashing Methods
- Hadamard Methods
- **Comparison of best oracles**

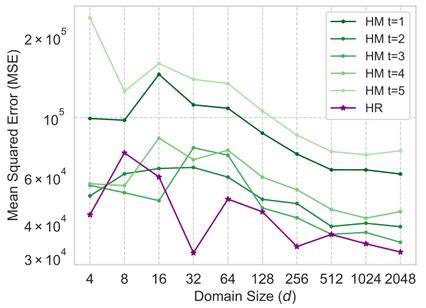










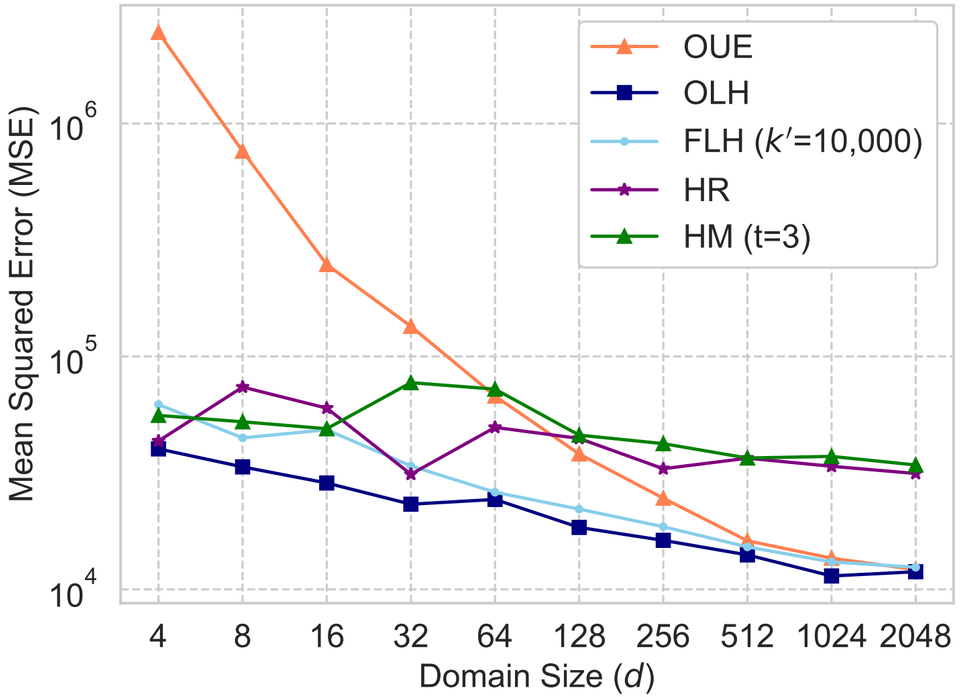


WARWICK
THE UNIVERSITY OF WARWICK

Frequency Oracles

Comparison of Oracles

- Experiments inline with theory -OLH (dark blue) achieving the smallest MSE
- Heuristic FLH (light blue) has near identical performance as d increases
- FLH also has substantial speed-ups (10x) in server aggregation
- Clear grouping between Hadamard methods and the local hashing/OUE methods



WARWICK
THE UNIVERSITY OF WARWICK

Domain Reduction

Consider three types that can be combined with any FO:

1. **Bloom Filters:** **[Erlingsson et al 2014]** Uses a set of hash functions to encode input. Perturbation applied to satisfy LDP. Server decoding slightly more complex

2. **Count-Min (CM) Sketches** **[Apple 2017]** Randomly choose one of r hash functions that map data to {1,…,c }, perturb the hash. Server constructs an r x c noisy sketch matrix to form final estimates

3. **Count Sketches (CS)** **[Bassily et al 2017]** Similar to CM but uses two sets of hash functions

We consider **Minimum, Mean and Median** estimation methods for sketches

In the full paper, we run experiments detailing the effects of sketch sizes, comparison of estimation methods and Bloom Filter vs Sketching

WARWICK
THE UNIVERSITY OF WARWICK

Domain Reduction

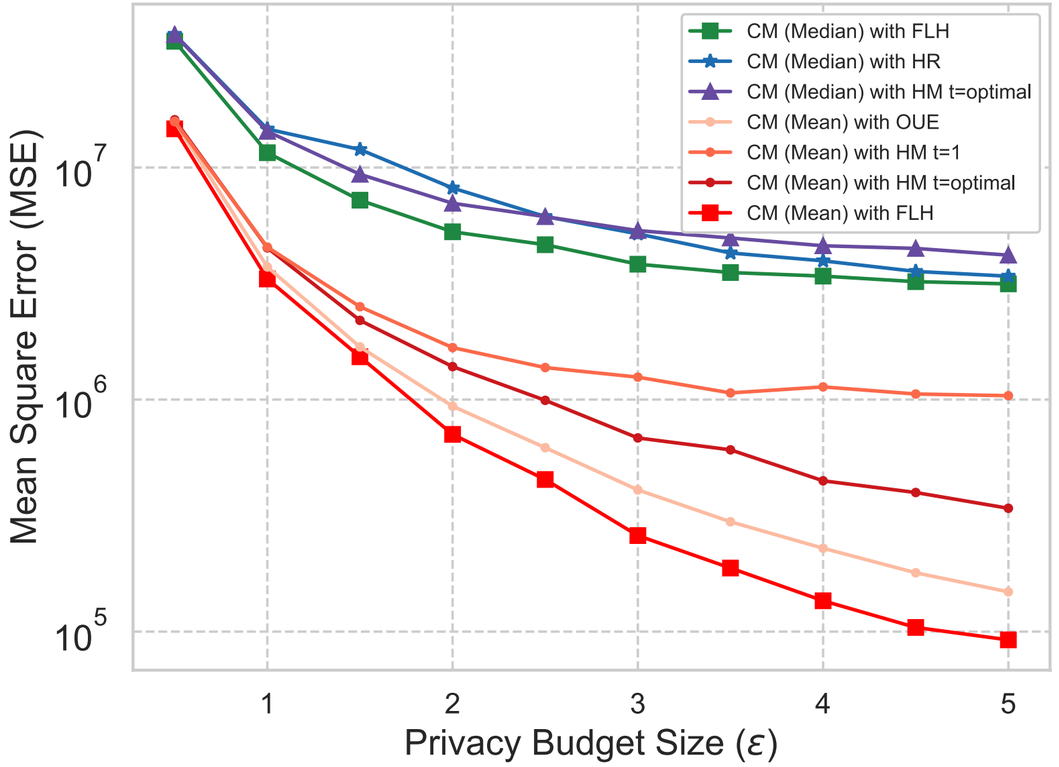
Sketch + Oracle Combinations

Popular LDP implementations can easily be expressed in the framework:

- **Google's RAPPOR** = Bloom Filter + SUE
- **Apple's CMS** = CM Mean + S/OUE
- **Apple's HCMS** = CM Mean + HM (t=1) for HCMS

Experiment Results:

- Grouping of mean and median techniques
- Sketching with (F)LH performs best and beats Apple's CMS
- Sampling more Hadamard coefficients gives improvements over Apple's HCMS
- Overall, clear improvements over Apple's deployments



WARWICK
THE UNIVERSITY OF WARWICK

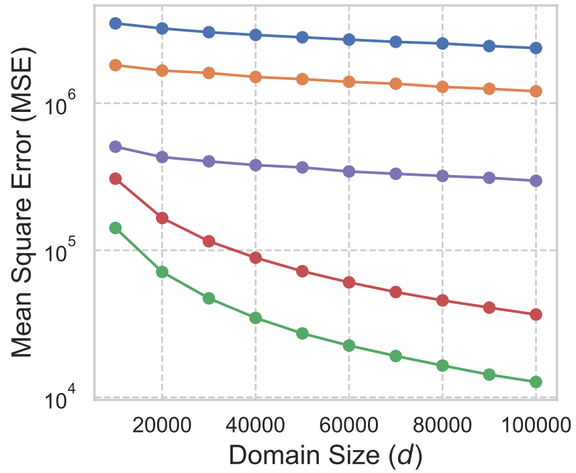
Post-Processing

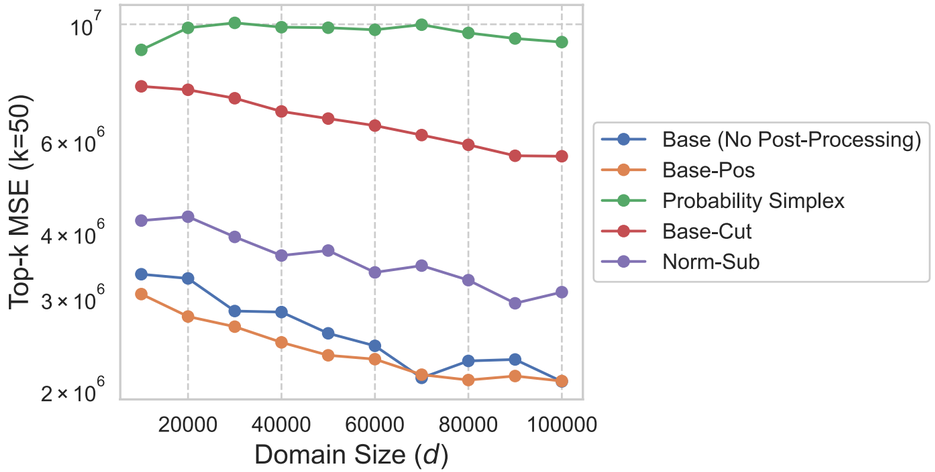
[Wang et al 2020] present an experimental comparison of post-processing techniques

Post-processing affects both MSE and k-MSE separately

General conclusions:

- Rounding negative values to zero always improves MSE
- Adding a normalisation constant to ensure frequency estimates sum up to n achieves best balance of MSE/k-MSE
- Further experiments were performed with sketching, conclusions generally the same





WARWICK
THE UNIVERSITY OF WARWICK

Heavy Hitters

Compare and contrast the most popular methods:

- **Sequence Fragment Puzzle (SFP) [Apple 2017]**
  - Frequent strings in a population will also have frequent substrings
  - Heavy hitters are ‘reconstructed’ by privatising substrings and joining them together based on the hash of the overall string and estimated frequencies

- **Hierarchical Search Methods -**Prefixes (instead of substrings) are used to build up HHs

  - Prefix Extending Methods (PEM) **[Wang et al 2019]** - Randomly sample length of prefix

  - TreeHistogram (TH) **[Bassily et al 2017]** - Split privacy budget in half and privatise the whole string and also a randomly chosen prefix (of a fixed length)

WARWICK
THE UNIVERSITY OF WARWICK

Heavy Hitters

Experiments

Goal to recover top-10 most frequent URLs in AOL dataset, results sorted in order of F1 score

Combinations Used:

- **Top-T:** T=10, T=20 for using top-10 or top-20 prefixes
- **Frequency Oracles:** OUE, FLH
- **Domain Reduction:** CM Mean and CM Median over 3 different sketch sizes (32,1024), (128,1024), (1024,2048)

Key Results:

- PEM dominates
- FLH and OUE performance generally similar, but FLH has better communication cost
- Good performance at smaller sketch sizes than those proposed in **[Apple, 2017]**

HH Protocols on AOL dataset, n=1,935,614, d=383,4667, ϵ=3

	Heavy Hitter	Sketch Method	Frequency Oracle	Parameters	Precision	Recall	F1 Score
1	PEM	CM (Median)	FLH	T=20, r=1024, c=2048	0.822	0.74	**0.779** (0.052)
2	PEM	CM (Median)	FLH	T=20, r=128, c=1024	0.789	0.74	0.763 (0.051)
3	PEM	CM (Mean)	OUE	T=10, r=1024, c=2048	0.626	**0.96**	0.757 (0.057)
4	PEM	CM (Mean)	OUE	T=20, r=32, c=1024	0.782	0.72	0.749 (0.025)
5	PEM	CM (Median)	FLH	T=10, r=1024, c=2048	0.758	0.74	0.748 (0.061)
6	PEM	CM (Mean)	OUE	T=20, r=128, c=1024	0.786	0.70	0.738 (0.035)
7	PEM	CM (Mean)	OUE	T=10, r=32, c=1024	0.782	0.70	0.738 (0.025)
8	PEM	CM (Median)	FLH	T=20, r=32, c=1024	0.792	0.68	0.731 (0.036)
9	PEM	CM (Mean)	OUE	T=20, r=1024, c=2048	0.766	0.70	0.73 (0.029)
10	PEM	CM (Median)	FLH	T=10, r=32, c=1024	0.744	0.70	0.721 (0.055)

WARWICK
THE UNIVERSITY OF WARWICK

Key Takeaways

1. Frequency Oracles: In small domains LH methods are the best with OLH dominating, heuristic variant FLH achieves faster aggregation with little loss in accuracy. For larger domains HR is as accurate but magnitudes faster than LH

2. Domain Size Reduction: CM sketches are preferable with mean estimation performing best. Accurate results are possible using much smaller sketch sizes than previously published i.e in **[Apple 2017]**

3. Post-processing: Should always be used, more sophisticated methods can improve performance when d large/have prior knowledge

4. Heavy Hitters: Hierarchical encoding methods particularly PEM dominates other choices. Use of FLH oracle with smaller sketches show improvements over current implementations (i.e Apples)

References

WARWICK
THE UNIVERSITY OF WARWICK

- **[Erlingsson et al 2014]** Erlingsson, Ú., Pihur, V. and Korolova, A., 2014, November. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security* (pp. 1054-1067).

- **[Wang et al 2017]** Wang, T., Blocki, J., Li, N. and Jha, S., 2017. Locally differentially private protocols for frequency estimation. In *26th {USENIX} Security Symposium ({USENIX} Security 17)* (pp. 729-745).

- **[Bassily and Smith 2015]** Bassily, R. and Smith, A., 2015, June. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing* (pp. 127-135).

- **[Acharya et al 2018]** Acharya, J., Sun, Z. and Zhang, H., 2019, April. Hadamard response: Estimating distributions privately, efficiently, and with little communication. In *The 22nd International Conference on Artificial Intelligence and Statistics* (pp. 1120-1129). PMLR.

- **[Apple 2017]** Apple Privacy Team**, https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf**

- **[Wang et al 2019]** Wang, T., Li, N. and Jha, S., 2019. Locally differentially private heavy hitter identification. *IEEE Transactions on Dependable and Secure Computing*.

- **[Bassily et al 2017]** Bassily, R., Nissim, K., Stemmer, U. and Thakurta, A., 2017. Practical locally private heavy hitters. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)

- **[Wang et al 2020]** Wang, T., Lopuhaä-Zwakenberg, M., Li, Z., Skoric, B. and Li, N., 2019. Locally differentially private frequency estimation with consistency. In *NDSS 2020*