Secure Multi-party Computation of Differentially Private Median

Jonas Böhler (SAP Security Research)
Florian Kerschbaum (University of Waterloo)

USENIX Security 2020

PUBLIC

THE BEST RUN SAP

Motivation & Preliminaries

Distributed Private Learning

Parties with **sensitive data** want to learn **statistics over joint data** while **preserving privacy**

- Real-world examples
  - *Differential Privacy*
    - browser settings, *Google* [EPK14]
    - website‘s resource consumption, *Apple* [A17]
    - telemetry data, *Microsoft* [DKY17]
  - *Secure Computation*
    - ad conversions, *Google & Mastercard* [B18]
    - tax fraud detection, *Estonian government & Sharemind* [BJSV15]
    - government studies, *Boston Women’s Workforce Council* [LJAIQVB18]

Our focus: **semi-honest users** computing **rank-based statistics**, especially the **median**

- with **high accuracy** even for **small number or users** (small data)
- and **strong privacy**, supporting **large domains**

© 2020 SAP SE or an SAP affiliate company. All rights reserved. I PUBLIC

Why rank-based statistics & median?

Rank of a value w.r.t. a data set D: *first* position in sorted data (zero-indexed)

Data set D	0	1	1	2	3	5	8	13	21
Rank	0	1		3	4	5	6	7	8

Rank-based statistics: versatile & robust

- min
- max
- in general, k^{th}-ranked element (p^{th}-percentile)
  - **median**
    - “typical value” in data
    - more **robust** to outliers than mean

Example: income in Medina, Washington
Population ≈3,000

- **Median Income** ≈ $186,000
- **Average Income** ≫$1,000,000,000
  - „outliers“ Jeff Bezos and Bill Gates

© 2020 SAP SE or an SAP affiliate company. All rights reserved. I PUBLIC

Why Differential Privacy (DP)?

We consider **private** distributed learning

- Median is one individual's value, no privacy

ϵ-DP is a strong **privacy guarantee**

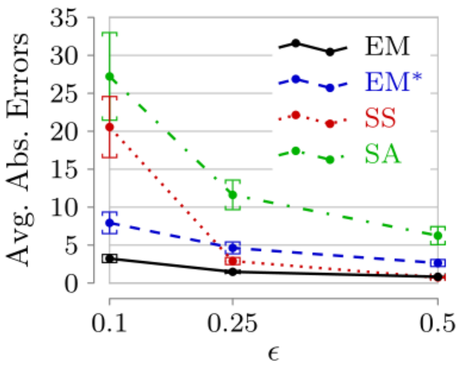
- bounds output differences if input changes in one record
  – small ϵ corresponds to high privacy

DP achieved by **additive noise** or **exponential mechanism** (EM) [MT07]

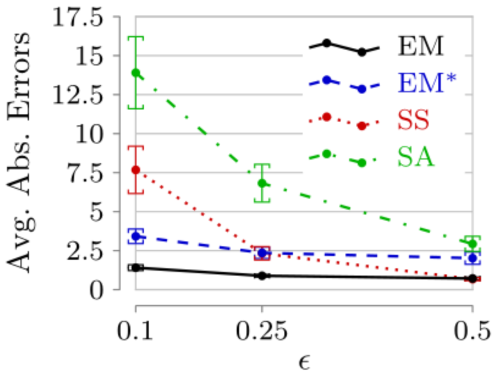
- EM outputs m from domain U w.r.t. data set $D \in U^n$ with probability $\propto \exp(\epsilon \cdot u(D, m))$
  – utility $u(D,\cdot)$ scores, e.g., closeness to **median**
- we use EM as it provides better accuracy for the median [LLSY16]

Accuracy: **(central) DP median solutions** (average absolute error of 100 runs)

- *With trusted server*
  – Exponential mechanism EM [MT07]
  – Smooth sensitivity SS [NRS07]
- *Without trusted server*
  – This work EM*
  – Sample-and-Aggregate SA [PL15]



100k credit card payments

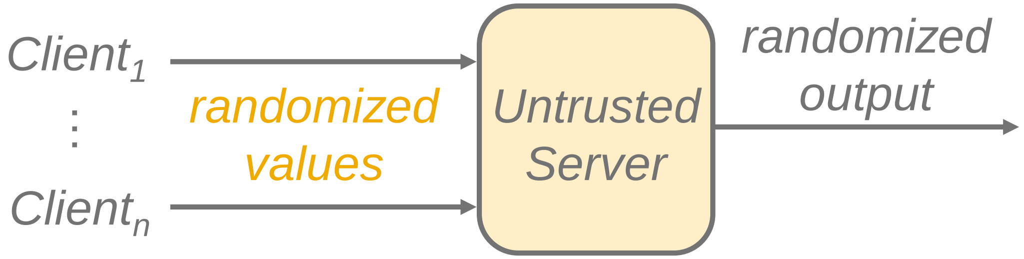


100k samples, normal distribution, $\sigma = 3$

© 2020 SAP SE or an SAP affiliate company. All rights reserved. I PUBLIC

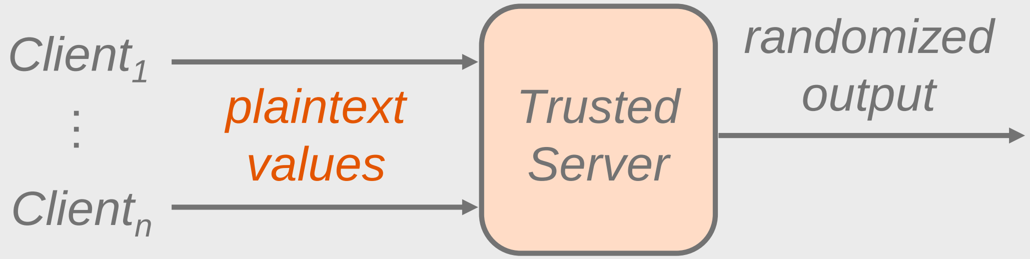
Why Secure Multi-party Computation (MPC)?

Local DP model



- ✓ no trusted server
- ✗ *requires large data for good accuracy*

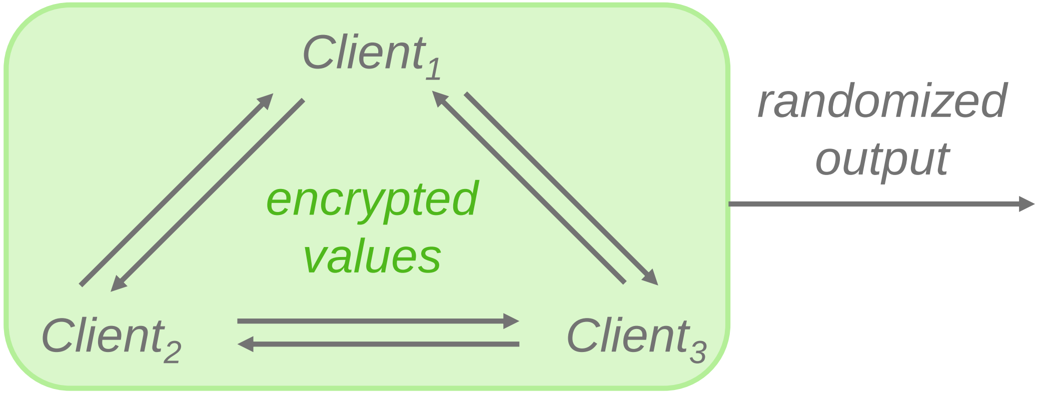
Central DP model



- ✓ high accuracy
- ✗ *requires trusted server*

MPC:

3+ parties jointly compute a function, without revealing inputs



- ✓ no trusted server
- ✓ high accuracy
- ? *inefficient*

© 2020 SAP SE or an SAP affiliate company. All rights reserved. I PUBLIC

Efficient MPC for DP Median via $\mathbf{EM}$

Challenges & Solutions

EM outputs domain element m with probability $\propto \exp\big(\epsilon \cdot u(D, m)\big)$

- **Large domains?**
  - divide domain in subranges, iteratively select subrange with highest utility
    - running time sublinear in domain size
- **Distributed data?**
  - use decomposable utility functions: $u(\text{JointData}, \cdot) = \sum_i u'(\text{ClientData}_i, \cdot)$
    - examples: counts, **ranks**, mode, convex optimization (empirical risk minimization)
- **Costly secure exponentiation?**
  - leverage decomposability, let $u = \sum_i u'(\text{ClientData}_i, \cdot)$ and compute:

$\boldsymbol{\epsilon}$	$\mathbf{exp}(\boldsymbol{\epsilon u})$
$\ln(2)$	2^u
$\ln(2)/2^d$, integer $d \geq 1$	$2^{\lfloor u/2^d \rfloor} \cdot 2^{(u \bmod 2^d)/2^d}$
$\in \mathbb{R}$	$\prod_i \exp\big(\epsilon \cdot u'(\text{ClientData}_i, \cdot)\big)$

© 2020 SAP SE or an SAP affiliate company. All rights reserved. ǀ PUBLIC

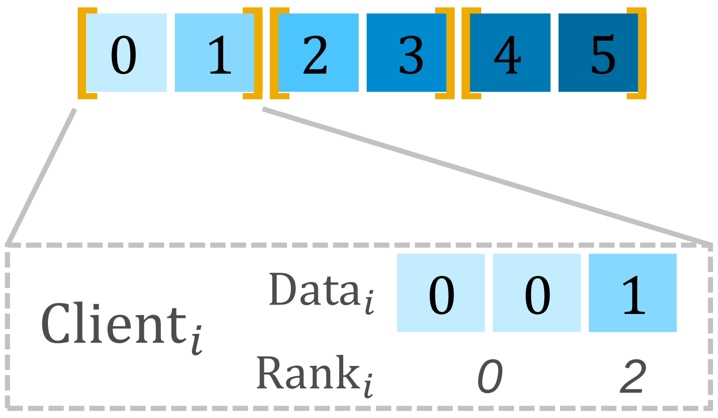
Step by Step

Divide data domain into subranges

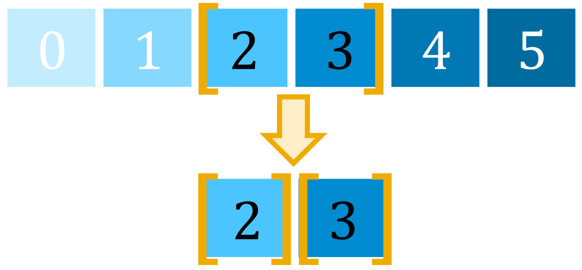
Repeat until subranges are small:

- **Evaluate**
  – compute local results (utility or weight) per subrange
    - *utility*: rank of subrange endpoints relative to median's rank $\frac{|\text{JointData}|}{2}$
    - *weight*: $\exp\left(\epsilon \cdot u'(\text{Data}_i, \cdot)\right)$
- **Combine**:
  – combine local results into global weights
- **Select**:
  – output a subrange based on its weights
- Divide selected subrange into subranges for next iteration

Output random element from last subrange



$$\text{Rank}_{\text{Joint}}(\cdot) = \sum_i \text{Rank}_i(\cdot)$$



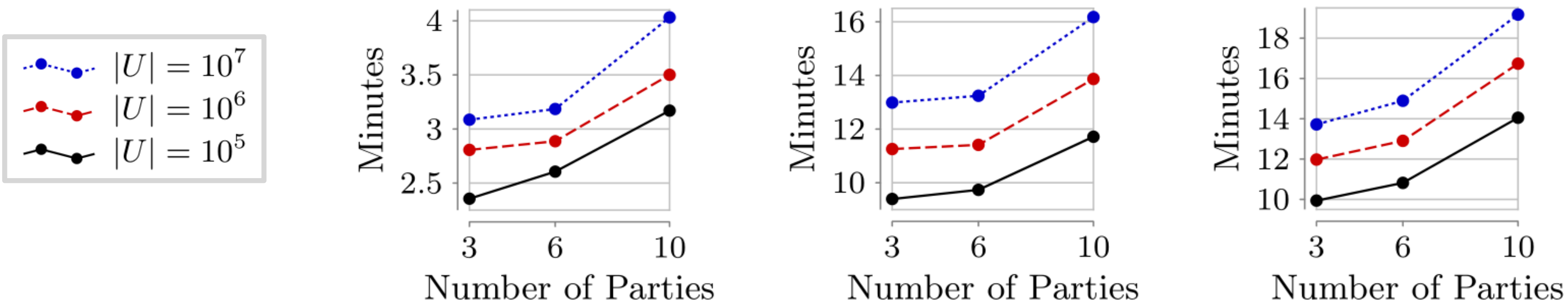
© 2020 SAP SE or an SAP affiliate company. All rights reserved. I PUBLIC

Evaluation

Running time in WAN

- **WAN** with 100 ms latency, 100 Mbits/s bandwidth (AWS regions Frankfurt, Ohio)
  – **LAN** running time: $10 - 60$ seconds
- 10^6 clients with one value each using $3, 6, 10$ computation parties
  – computation parties (t2.medium instances) already received client inputs
- iterate until last subrange has size 1
  – $\lceil \log_{10}|U| \rceil \in \{5,6,7\}$ iterations
- Evaluated 3 different weight computations w.r.t. ϵ

$\epsilon_{\text{iteration}} = \ln(2)$ $\qquad$ $\epsilon_{\text{iteration}} = \frac{\ln(2)}{4}$ $\qquad$ $\epsilon_{\text{iteration}} \in \mathbb{R}$



© 2020 SAP SE or an SAP affiliate company. All rights reserved. ǀ PUBLIC

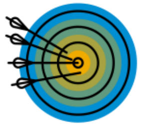
Conclusion

Conclusion

Existing DP median solutions with good accuracy require either

- large data (local model)
- trusted third party (central model)
- small domain (MPC)

Our contributions are



- **high accuracy** even for small data and low ϵ
  – MPC of exponential mechanism

- **efficient MPC protocol**
  – decomposable utility functions
  – independent of data size

- supporting **large domains**
  – using subranges

© 2020 SAP SE or an SAP affiliate company. All rights reserved. I PUBLIC

Thank you.

Contact information:

jonas.boehler@sap.com

THE BEST RUN SAP®

References

[A17] Apple's Differential Privacy Team. "Learning with privacy at scale," 2017, https://machinelearning.apple.com/research/learning-with-privacy-at-scale

[B18] https://www.bloomberg.com/news/articles/2018-08-30/google-and-mastercard-cut-a-secret-ad-deal-to-trackretail-sales

[BJSV15] D. Bogdanov, M. Jõemets, S. Siim, and M. Vaht, "How the estonian tax and customs board evaluated a tax fraud detection system based on secure multi-party computation," FC 2015

[BKKRST16] D. Bogdanov, L. Kamm, B. Kubo, R. Rebane, V. Sokk, and R. Talviste, "Students and taxes: a privacy-preserving study using secure computation," PETS 2016.

[DKY17] B. Ding, J. Kulkarni, and S. Yekhanin. "Collecting telemetry data privately," NIPS, 2017

[EPK14] Ú. Erlingsson, V. Pihur, and A. Korolova. "Rappor: Randomized aggregatable privacy-preserving ordinal response," CCS, 2014.

[LJAIQVB18] A. Lapets, F. Jansen, K.D. Albab, R. Issa, L. Qin, M. Varia and A. Bestavros. "Accessible Privacy Preserving Web-Based Data Analysis for Assessing and Addressing Economic Inequalities," COMPASS 2018.

[LLSY16] N. Li, M. Lyu, D. Su, and W. Yang. "Differential privacy: From theory to practice," Synthesis Lectures on Information Security, Privacy, & Trust, 2016.

[MT07] F. McSherry and K. Talwar, "Mechanism design via differential privacy," FOCS 2007.

[NRS07] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," STOC, 2007.

[PL15] M. Pettai and P. Laud. "Combining differential privacy and secure multiparty computation," ACSAC, 2015.

© 2020 SAP SE or an SAP affiliate company. All rights reserved. ǀ PUBLIC