Multi-Central Differential Privacy

Thomas Steinke[*]

Abstract

Differential privacy is typically studied in the *central model* where a trusted "aggregator" holds the sensitive data of all the individuals and is responsible for protecting their privacy. A popular alternative is the *local model* in which the aggregator is untrusted and instead each individual is responsible for their own privacy. The decentralized privacy guarantee of the local model comes at a high price in statistical utility or computational complexity. Thus intermediate models such as the *shuffled model* and *pan privacy* have been studied in an attempt to attain the best of both worlds.

In this note, we propose an intermediate trust model for differential privacy, which we call the *multi-central model*. Here there are multiple aggregators and we only assume that they do not collude nefariously. This model relaxes the trust requirements of the central model while avoiding the price of the local model. We motivate this model and provide some simple and efficient algorithms for it. We argue that this model is a promising direction for further research.

1   Introduction

Differential privacy [DMNS06] is a formal standard for the protection of sensitive information about individuals in data analysis. Informally, differential privacy ensures that we cannot infer the attributes of any individual from the output of the algorithm. Differentially private algorithms randomly distort their output enough to obscure an individual's contribution – but, hopefully, not too much, so that the output remains useful.

It is simplest to assume that the differentially private algorithm is executed by a trusted party – the "aggregator" – that has access to all the data and only the final output is released. This standard setup is referred to as the *central model*.

The central model requires individuals to trust the aggregator to securely store and process their data, but such a universally trusted entity may not exist. Thus it is important to also consider privacy threats arising from the computation process, rather than just the final output.

There are a few approaches for avoiding the need for a trusted aggregator:

- *Local randomization:* Certain algorithms, like randomized response [War65], perform the distortion needed to guarantee differential privacy at the individual level, before the data is aggregated. Since the individuals (or, rather, their personal devices) perform the distortion themselves, they need not trust any other party. This is what is known as the *local model* of differential privacy [KLNRS08].

  The minimal trust requirements and scalable algorithms of local differential privacy make it very attractive in practice. Google [EPK14], Apple [App16], and Microsoft [DKY17] have all deployed locally differentially private systems with millions of users.

  Unfortunately, algorithms in the local model are necessarily less accurate than their counterparts in the central model [BNO08; KLNRS08; CSS12b; DJW18; Ull18; BNS18; DR18; JMNR19;

---

[*]IBM Research – Almaden .... `multicentral@thomas-steinke.net`

JMR20]. Essentially, each individual must randomly distort their own data as much as a trusted aggregator would distort the sum of everyone's data. The variances of all these distortions add up, resulting in much larger error being necessary in the local model than in the central model.

- *Cryptography:* Various works [DKMMN06; BNO08; SCRCS11; NH12; CSS12a; PL15; Bon+17; GX17; Kai+19] have employed variants of *secure multiparty computation (SMC)* (a.k.a. *secure function evaluation*) [Yao82; Yao86; GMW87; LP08] to simulate having a trusted aggregator while only making trust assumptions akin to the local model. These tools typically provide a computational differential privacy guarantee [MPRV09; MMPRTV10] – that is, we must make an assumption about the computational abilities of a potential adversary.

  However, these cryptographic protocols incur significant overheads. Building practical SMC systems remains a significant research challenge [PSSW09; HKE13; CMTB16]. At this stage, it is not feasible to scale SMC to thousands or millions of individuals who only have the computational and communication resources (and reliability) of a personal computer or smartphone.

- *Special assumptions:* Intermediate trust models between the central model and the local model have also been considered. These make some special-purpose assumption, with the goal of striking a better balance between realistic privacy and statistical utility.

  The *pan-private model* [DNPR10] assumes that the aggregator is trustworthy now, but may not be trustworthy in the future. It requires the aggregator to only store distorted values, instead of the raw data, to protect against future intrusions.

  The *shuffled model* [CSUZZ19; EFMRTT19] assumes that there are many individuals honestly participating in the protocol and that the aggregator receives their messages in an anonymous manner – that is, the adversary does not know which message originated from which individual.

We propose another intermediate trust model between the central and local models and argue that it attains the best of both worlds – a realistic trust model, accurate and practical algorithms, and, of course, a rigorous privacy guarantee.

1.1 Our Model: Multi-Central Differential Privacy

We now specify the setup of our model and the guarantee of multi-central differential privacy.

There are m aggregators $A_1, \cdots, A_m$ and n individuals with private data $x_1, \cdots, x_n$. The individuals will communicate with the aggregators and the aggregators will communicate with one another and, finally, produce some public output.

Our trust assumption is that the aggregators do not collude nefariously. That is, we assume that at least one of the aggregators is trustworthy, but we do not know which one. More precisely, under multi-central differential privacy, the adversary chooses $m-1$ aggregators and $n-1$ individuals to control. The remaining individual to be protected and remaining trustworthy aggregator faithfully follow the protocol. The view of the adversary consists of all the information available to the aggregators it controls plus the final output of the trustworthy aggregator. It excludes only the internal information of the trustworthy aggregator and the individual to be protected. Differential privacy for the remaining individual's data must be provided with respect to this view. That is, the protocol is computationally (ε, δ)-differentially private in the multi-central model if, for all trustworthy aggregators $j \in [m]$ and all individuals $i \in [n]$ to be protected and all datasets $x_1, \cdots, x_n, x_i'$ and all efficiently computable S,

$$\mathbb{P}\left[\mathsf{View}_{A_1, \cdots, A_{j-1}, \mathsf{output}(A_j), A_{j+1}, \cdots, A_m}(x_1, \cdots, x_n) \in S\right] \leq e^\varepsilon \cdot \mathbb{P}\left[\mathsf{View}_{A_1, \cdots, \mathsf{output}(A_j), \cdots, A_m}(x_1, \cdots, x_i', \cdots, x_n) \in S\right] + \delta.$$

We remark that the number of aggregators m can be viewed as a "trust parameter." The larger m is, the less reliant we are on trusting any individual aggregator. Setting $m = 1$ simply recovers the usual central model, while $m = n$ is analogous to the local model. The statistical or computational cost of privacy also increases with m.

Note that this model does not guarantee utility in the presence of adversaries; utility may be more easily compromised by a single malicious party than privacy.

1.2 Motivation

We envisage the m aggregators as being a handful of large data centers and the n individuals as millions of people with smartphones or personal computers and all communication happening via the internet using public-key cryptography. The final output can be freely shared.

We now suggest several scenarios in which our trust model is potentially realistic (or, at least, may be more realistic than the central model).

- *Insider/physical threats:* A major concern when individuals allow their data to be stored and processed in a data center is that an employee or someone who has gained physical access to the servers will obtain their sensitive data.

  The multi-central model protects against such insider threats, as long as no employee is granted access to systems in all of the data centers. Likewise, a physical attacker would need to break into all of the data centers to compromise the multi-central model.

- *Multiple jurisdictions:* Privacy laws are becoming increasingly complex and this is compounded by the fact that data often crosses international borders. For example, the European Union's General Data Protection Regulation (EU GDPR) restricts the transfer of private data abroad.

  The multi-central model may help address some of these international legal challenges. If a data center (and the relevant management) is placed in each applicable jurisdiction, then the data could be protected under the laws of all of these jurisdictions. Bypassing multi-central differential privacy and accessing the raw data in a legal fashion would presumably require adhering to the privacy laws of all jurisdictions. Users may have increased confidence in the privacy protections knowing that, in the event of a privacy breach, they potentially have legal recourse in multiple courts.

- *Trust as a Service:* A trusted organization can be contracted to serve as one of the aggregators. Thereby users need only trust that this organization follows the terms of its contract. In essence, this organization is providing a "trust service". Such a service would be useful for companies that are not themselves fully trusted by users or have incentives to breach privacy.

- *Collaborating Rivals:* The multi-central model provides a framework for companies that are usually fierce rivals to collaborate and, in the process, benefit the privacy of their users. In this framework, the companies would each take the role of one aggregator and their user bases would be combined for the purpose of data collection. Thereby the companies gain better statistical accuracy from the larger combined user base. Furthermore, users' privacy is protected by the rivalry between companies. That is, the multi-central model protects the user data, as long as we trust the companies to not collude to violate privacy.

- *Honest but curious:* The multi-central model protects against overly curious aggregators – since each aggregator only sees the data in an indecipherable form, they are not tempted to explore it.

Needless to say, the multi-central model does not account for all possible threats. For example, it is likely that the aggregators will share software or a hardware provider; if this common software or hardware is compromised, then so too is the multi-central model. However, even local differential privacy is vulnerable to compromised software or hardware at the user end. Additionally, if all the aggregators belong to the same organization, then the leadership of that organization could presumably violate the multi-central privacy protections. This is why it is preferrable to have the aggregators belong to distinct organizations.

1.3 Related Work

We begin by noting that our proposed trust model is not novel. It has been widely studied, for example, in *private information retrieval* [CGKS95] and *secret sharing* [Sha79; Bla79]. Indeed, we seek to build upon these lines of work.

Furthermore, combining this trust model with differential privacy has been previously proposed [CGB17]. However, to the best of our knowledge, this has not been seriously investigated. The purpose of this note is to begin such an investigation and to encourage further work.

The multi-central trust model is, of course, closely related to the central, local, and shuffled models. We briefly remark about how these compare. Clearly, any algorithm for the local model can be run in the multi-central model and any algorithm in the multi-central model can be run in the central model. This establishes the multi-central model as an intermediate trust model.

Also, the shuffled model can be simulated in the multi-central model – the aggregators must simply perform the shuffling of messages. Thus our model subsumes the functionality of the shuffled model.

Using secure multiparty computation, it is possible to simulate the central model in the multi-central model (with a computational differential privacy guarantee). Unfortunately, secure multiparty computation is not yet practical for general-purpose computations. Nonetheless, this establishes that there is no inherent ceiling to the potential of algorithms the multi-central model model. The only limitation to the multi-central model is our ability to devise and execute algorithms and the limits that would be present even in the central model.

In the rest of this note, we explore some algorithms for the multi-central model. We believe these demonstrate that the multi-central model warrants further (empirical) exploration.

2 Algorithms

Having introduced and motivated the multi-central model, we now turn to demonstrating its utility by proposing several simple, accurate, and efficient private algorithms.

2.1 Counting Queries

Counting queries are a particularly simple and fundamental task. Namely, given $q : \mathcal{X} \to \{0, 1\}$, the task is to privately approximate $\frac{1}{n} \sum_{i=1}^{n} q(x_i)$, where the private data of the individuals is $x_1, \cdots, x_n \in \mathcal{X}$. Our algorithm for this task is also simple and is based on a *linear secret sharing scheme* [Sha79; Bla79].

Choose a modulus $p > n$. Each individual $i \in [n]$ generates independent uniformly random values $v_i^1, \cdots, v_i^{m-1} \in \mathbb{Z}_p$ and sets $v_i^m = q(x_i) - \sum_{j=1}^{m-1} v_i^j \mod p$. Then v_i^j is transmitted to A_j (using public-key cryptography) for each $j \in [m]$.

If the aggregators where to combine all their values, they can recover $q(x_i) = \sum_{j=1}^{m} v_i^j$. However, any $m-1$ aggregators see only uniformly random values and have no information about $q(x_i)$. This

is the security property of the secret sharing scheme and the basis of the multi-central differential privacy guarantee.

In the protocol, each aggregator $j \in [m]$ computes $v^j = \sum_{i=1}^{n} v_i^j$ and outputs a noisy sample $V^j \leftarrow \mathcal{N}(v^j, \sigma^2)$ from the continuous or discrete Gaussian [CKS20]. The addition of noise ensures multi-central $(\rho = 1/2\sigma^2)$-concentrated differential privacy [BS16], which entails multi-central $(\varepsilon = 1/2\sigma^2 + \sqrt{2\log(1/\delta)/\sigma^2}, \delta)$-differential privacy [DKMMN06] for all $\delta > 0$.

The final output is $\frac{\sum_{j=1}^{m} V^j \mod p}{n}$, which is distributed as $\mathcal{N}\left(\frac{1}{n}\sum_{i=1}^{n} q(x_i), \frac{m \cdot \sigma^2}{n^2}\right) \mod \frac{p}{n}$. Assuming $p \gg n$ and $m \cdot \sigma^2 \ll n^2$, this value will be close to the desired value. The accuracy guarantee degrades as m gets larger. We note that $m = 1$ recovers the accuracy guarantee obtainable in the central model and $m = n$ recovers the accuracy guarantee obtainable in the local model.

This algorithm is also very efficient. Each individual must transmit $O(\log p) = O(\log n)$ bits per aggregator in one round. The main computational task is encrypting the messages for transmission. The aggregators must simply decrypt the values and add them up along with some noise.

2.2 Heavy Hitters

Identifying heavy hitters is a well-studied and important task in differential privacy, particularly in the local model [MS06; HKR12; EPK14; BS15; App16; QYYKXR16; BST17; BNS18]. The key task is to be able to privately estimate the frequency within the dataset of any element. (Sophisticated reductions on top of such a frequency estimator can then be used to find the frequent elements.)

We can efficiently construct a frequency estimator for multi-central differential privacy using the previous algorithm for statistical queries combined with linear sketching, such as the Johnson-Lindenstrauss transform. In particular, our algorithm produces a sketch subject to multi-central $(\rho = \frac{1}{2}\varepsilon^2)$-concentrated differential privacy [BS16] which can be used to estimate the frequency of any element with additive relative error $\alpha = O\left(\frac{\sqrt{m\log(1/\beta)}}{\varepsilon n} + \sqrt{\frac{\log(1/\beta)}{\ell}}\right)$, where β is the allowed failure probability and ℓ is the size of the sketch. Each individual must send $O(\ell \cdot \log n)$ bits to each aggregator. (The algorithm also requires $O(\log|\mathcal{X}| \cdot \log(1/\beta))$ bits of shared randomness for the sketch [CW09, Thm. 2.2], where $\mathcal{X}$ is the data universe.) Multiple queries can be answered using the same sketch.

For $m = O(1)$, this provides utility comparable to the central model.

2.3 Computationally-Secure Secret Sharing & Decision Trees

The previous two algorithms use information-theoretic secret sharing. However, we can reduce the communication in some settings by using computational *function secret sharing*. Boyle et al. [BGI15; BGI16] (see also [WYGVZ17; Wan18]) show how to secret share any family of queries represented by a decision tree of size ℓ with secret shares of size $O(\lambda \cdot \ell)$, where λ is the security parameter (i.e., key length). These constructions are very efficient and only use symmetric-key cryptography (specifically, pseudorandom generators), rather than more exotic cryptographic primitives.

This yields an efficient algorithm in the multi-central model for threshold queries [DNPR10; BNSV15] over domain $\mathcal{X} \subset \mathbb{R}$, which can be computed by a decision tree of size $\ell = O(\log|\mathcal{X}|)$. This also yields an alternative frequency estimation algorithm for heavy hitters in which the share size and, hence, communication per aggregator is $O(\lambda \cdot \min\{\log|\mathcal{X}|, \log(1/\alpha\beta)\})$ bits, rather than $O(\log(n) \cdot \log(1/\beta)/\alpha^2)$ as above. This is a significant improvement in the high accuracy (small α) regime.

More generally, function secret sharing can be used to significantly reduce communication in the

multi-central model. Answers to a structured family of queries (e.g., thresholds) can be compressed. However, compression is possible even in unstructured settings: For example, suppose we need to answer k arbitrary counting queries. If we use the information-theoretic secret sharing algorithm above k times, then the communication will grow linearly in k, which could be prohibitive. However, it is possible to instead have each individual pick one out of the k queries at random and only answer that one. Function secret sharing allows the individual to transmit this single answer to the aggregators without the aggregators learning which query was answered and, crucially, the communication grows only logarithmically in k. Effectively, the individual sends a vector of answers where at most one entry is nonzero and this can be compressed. This approach introduces some statistical noise due to sampling, but, depending on the parameters, should still provide sufficiently accurate answers. In terms of privacy, each query answer should be privatized as before, but, since the adversary does not know which query the individual chose to answer, their privacy is amplified [CSUZZ19; EFMRTT19] and the final guarantee is comparable to that of the less efficient algorithm.

2.4 Secure Multiparty Computation & Selection

A general-purpose tool for the multi-central model is secure multiparty computation (SMC). The individuals can use a simple secret sharing scheme to send their data to the aggregators, who can then use SMC to compute the desired differentially private output as if they were a single trusted aggregator. This protocol is very efficient for the individuals and allows a rich set of analyses to be run, but may be computationally expensive for the aggregators (although not as expensive as including all individuals in the SMC protocol).

The practicality of the SMC will depend on the task being run. One important task is selection – given a dataset x and a set of (counting) queries $q_1, q_2, \cdots, q_k$, we wish to privately output some approximation to $\underset{\ell \in [k]}{\mathsf{argmax}}\ q_\ell(x)$. The classic algorithm for selection is the exponential mechanism [MT07]. An alternative is the report-noisy-max algorithm [DR14], which outputs $\underset{\ell \in [k]}{\mathsf{argmax}}\ q_\ell(x) + \xi_\ell$, where $\xi_1, \cdots, \xi_k$ are independent Laplace noise scaled to the global sensitivity of the queries. We can implement a variant of report-noisy-max in the multi-central model using fairly limited SMC as follows.

First, every individual $i \in [n]$ computes secret shares $v_{i,j,\ell} \in \mathbb{Z}_p$ such that $\sum_{j \in [m]} v_{i,j,\ell} = q_\ell(x_i)$ mod p for all $\ell \in [k]$. Each $v_{i,j,\ell}$ is then sent to aggregator j. Each aggregator $j \in [m]$ computes $\hat{v}_{j,\ell} := \sum_{i \in [n]} v_{i,j,\ell} + \xi_{j,\ell} \mod p$, where $\xi_{j,\ell}$ is (discrete) Laplace noise [CKS20]. Finally, SMC is used to compute

$$\ell^* := \underset{\ell \in [k]}{\mathsf{argmax}} \left(\sum_{j \in [m]} \hat{v}_{j,\ell} \mod p \right)$$

and the value ℓ^* is released. This is differentially private as long as the SMC only reveals the argmax and the adversary does not know the noise added by at least one of the aggregators. The utility is also comparable to that of the exponential mechanism, as long as the number of aggregators m is small. In terms of computation, the SMC must only compute the modular sum and the argmax. There is hope that this special-purpose task can be performed practically, but further investigation is needed.

We remark that efficient selection is believed to *not* be possible in the shuffled model [CSUZZ19; GGKPV19]. Thus this suggests a possible advantage of the multi-central model. Selection is an important tool, and is a critical subroutine in more complex systems [HR10]. Implementing selection

will allow other things to be built on top of it.

3 Conclusion and Further Work

This note motivates the study of the multi-central model of differential privacy and proposes several basic algorithms for the model. We hope that this motivates and serves as a basis for further exploration, including experimental evaluations.

We hope to develop more algorithms, guarantee accuracy in the presence of faults (using tools like *verifiable secret sharing* [CGMA85; CGB17]), and also explore other trust models.

References

[App16] Apple. *Differential Privacy Overview.* http://www.apple.com/newsroom/2016/06/apple-previews-ios-https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf. 2016.

[BGI15] E. Boyle, N. Gilboa, and Y. Ishai. "Function secret sharing". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques.* Springer. 2015, pp. 337–367.

[BGI16] E. Boyle, N. Gilboa, and Y. Ishai. "Function secret sharing: Improvements and extensions". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* ACM. 2016, pp. 1292–1303.

[Bla79] G. R. Blakley. "Safeguarding cryptographic keys". In: *Proceedings of the national computer conference.* Vol. 48. 1979, pp. 313–317.

[BNO08] A. Beimel, K. Nissim, and E. Omri. "Distributed Private Data Analysis: Simultaneously Solving How and What". In: *Proceedings of the 28th Annual Conference on Cryptology: Advances in Cryptology.* CRYPTO. 2008. URL: http://dx.doi.org/10.1007/978-3-540-85174-5_25.

[BNS18] M. Bun, J. Nelson, and U. Stemmer. "Heavy hitters and the structure of local privacy". In: *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems.* ACM. 2018, pp. 435–447.

[BNSV15] M. Bun, K. Nissim, U. Stemmer, and S. Vadhan. "Differentially private release and learning of threshold functions". In: *FOCS.* IEEE. 2015.

[Bon+17] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. "Practical secure aggregation for privacy-preserving machine learning". In: *CCS.* 2017.

[BS15] R. Bassily and A. Smith. "Local, private, efficient protocols for succinct histograms". In: *Proceedings of the forty-seventh annual ACM symposium on Theory of computing.* ACM. 2015, pp. 127–135.

[BS16] M. Bun and T. Steinke. "Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds". In: *TCC.* https://arxiv.org/abs/1605.02065. 2016.

[BST17] R. Bassily, U. Stemmer, and A. G. Thakurta. "Practical locally private heavy hitters". In: *Advances in Neural Information Processing Systems.* 2017, pp. 2288–2296.

[CGB17] H. Corrigan-Gibbs and D. Boneh. "Prio: Private, Robust, and Scalable Computation of Aggregate Statistics." In: *NSDI.* 2017.

[CGKS95] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. "Private information retrieval". In: *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on.* IEEE. 1995, pp. 41–50.

[CGMA85] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. "Verifiable secret sharing and achieving simultaneity in the presence of faults". In: *FOCS.* 1985.

[CKS20]   C. Canonne, G. Kamath, and T. Steinke. "The Discrete Gaussian for Differential Privacy". In: *arXiv preprint arXiv:2004.00010* (2020).

[CMTB16]   H. Carter, B. Mood, P. Traynor, and K. Butler. "Secure outsourced garbled circuit evaluation for mobile devices". In: *Journal of Computer Security* 24.2 (2016), pp. 137–180.

[CSS12a]   T.-H. H. Chan, E. Shi, and D. Song. "Privacy-preserving stream aggregation with fault tolerance". In: *International Conference on Financial Cryptography and Data Security*. 2012.

[CSS12b]   T. H. Chan, E. Shi, and D. Song. "Optimal lower bound for differentially private multi-party aggregation". In: *European Symposium on Algorithms*. Springer. 2012, pp. 277–288.

[CSUZZ19]   A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev. "Distributed differential privacy via shuffling". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2019, pp. 375–403.

[CW09]   K. L. Clarkson and D. P. Woodruff. "Numerical linear algebra in the streaming model". In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM. 2009, pp. 205–214.

[DJW18]   J. C. Duchi, M. I. Jordan, and M. J. Wainwright. "Minimax Optimal Procedures for Locally Private Estimation". In: *J. Amer. Statist. Assoc.* 113.521 (2018), pp. 182–201. ISSN: 0162-1459. URL: https://doi.org/10.1080/01621459.2017.1389735.

[DKMMN06]   C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. "Our Data, Ourselves: Privacy Via Distributed Noise Generation". In: *EUROCRYPT*. 2006.

[DKY17]   B. Ding, J. Kulkarni, and S. Yekhanin. "Collecting Telemetry Data Privately". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 3571–3580. URL: http://papers.nips.cc/paper/6948-collecting-telemetry-data-privately.pd

[DMNS06]   C. Dwork, F. McSherry, K. Nissim, and A. Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *TCC*. http://repository.cmu.edu/jpc/vol7/iss3/2. 2006.

[DNPR10]   C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. "Differential privacy under continual observation". In: *STOC*. 2010.

[DR14]   C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy*. Vol. 9. 3-4. Foundations and Trends® in Theoretical Computer Science, 2014.

[DR18]   J. C. Duchi and F. Ruan. "The Right Complexity Measure in Locally Private Estimation: It is not the Fisher Information". In: *arXiv preprint arXiv:1806.05756* (2018).

[EFMRTT19]   Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta. "Amplification by shuffling: From local to central differential privacy via anonymity". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2019, pp. 2468–2479.

[EPK14]   U. Erlingsson, V. Pihur, and A. Korolova. "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS. New York, NY, USA: ACM, 2014. URL: http://doi.acm.org/10.1145/2660267.2660348.

[GGKPV19]   B. Ghazi, N. Golowich, R. Kumar, R. Pagh, and A. Velingker. *On the Power of Multiple Anonymous Messages*. 2019. arXiv: 1908.11358 [cs.CR].

[GMW87]   O. Goldreich, S. Micali, and A. Wigderson. "How to play any mental game". In: *Proceedings of the nineteenth annual ACM Symposium on Theory of Computing (STOC)*. ACM. 1987, pp. 218–229.

[GX17]   S. Goryczka and L. Xiong. "A comprehensive comparison of multiparty secure additions with differential privacy". In: *IEEE transactions on dependable and secure computing* (2017). URL: http://www.mathcs.emory.edu/aims/pub/goryczka15tdsc.pdf.

[HKE13]   Y. Huang, J. Katz, and D. Evans. "Efficient Secure Two-Party Computation Using Symmetric Cut-and-Choose". In: *Advances in Cryptology – CRYPTO 2013*. Ed. by R. Canetti and J. A. Garay. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 18–35.

[HKR12]   J. Hsu, S. Khanna, and A. Roth. "Distributed private heavy hitters". In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2012, pp. 461–472.

[HR10]   M. Hardt and G. N. Rothblum. "A multiplicative weights mechanism for privacy-preserving data analysis". In: *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE. 2010, pp. 61–70.

[JMNR19]   M. Joseph, J. Mao, S. Neel, and A. Roth. "The role of interactivity in local differential privacy". In: *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2019, pp. 94–105.

[JMR20]   M. Joseph, J. Mao, and A. Roth. "Exponential separations in local differential privacy". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2020, pp. 515–527.

[Kai+19]   P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. "Advances and open problems in federated learning". In: *arXiv preprint arXiv:1912.04977* (2019).

[KLNRS08]   S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. D. Smith. "What Can We Learn Privately?" In: *2008 49th Annual IEEE Symposium on Foundations of Computer Science(FOCS)*. Vol. 00. Oct. 2008, pp. 531–540. URL: doi.ieeecomputersociety.org/10.1109/FOCS

[LP08]   Y. Lindell and B. Pinkas. "Secure Multiparty Computation for Privacy-Preserving Data Mining". In: (2008). URL: https://eprint.iacr.org/2008/197.pdf.

[MMPRTV10]   A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. P. Vadhan. "The Limits of Two-Party Differential Privacy". In: *FOCS*. 2010.

[MPRV09]   I. Mironov, O. Pandey, O. Reingold, and S. Vadhan. "Computational Differential Privacy". In: *Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*. CRYPTO. 2009. URL: http://dx.doi.org/10.1007/978-3-642-03356-8_8.

[MS06]   N. Mishra and M. Sandler. "Privacy via pseudorandom sketches". In: *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 2006.

[MT07]   F. McSherry and K. Talwar. "Mechanism Design via Differential Privacy". In: *FOCS*. 2007.

[NH12]   A. Narayan and A. Haeberlen. "DJoin: Differentially Private Join Queries over Distributed Databases." In: *OSDI*. 2012.

[PL15]   M. Pettai and P. Laud. "Combining Differential Privacy and Secure Multiparty Computation". In: *Proceedings of the 31st Annual Computer Security Applications Conference*. ACSAC. ACM, 2015. URL: http://doi.acm.org/10.1145/2818000.2818027.

[PSSW09]   B. Pinkas, T. Schneider, N. P. Smart, and S. C. Williams. "Secure Two-Party Computation Is Practical". In: *Advances in Cryptology – ASIACRYPT 2009*. Ed. by M. Matsui. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 250–267.

[QYYKXR16]   Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren. "Heavy hitter estimation over set-valued data with local differential privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2016, pp. 192–203.

[SCRCS11]   E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song. "Privacy-preserving aggregation of time-series data". In: *NDSS*. 2011.

[Sha79]   A. Shamir. "How to Share a Secret". In: *Commun. ACM* 22.11 (Nov. 1979), pp. 612–613. ISSN: 0001-0782. URL: `http://doi.acm.org/10.1145/359168.359176`.

[Ull18]   J. Ullman. "Tight Lower Bounds for Locally Differentially Private Selection". In: *arXiv preprint arXiv:1802.02638* (2018).

[Wan18]   F. Wang. *Function Secret Sharing (FSS) Library.* `https://github.com/frankw2/libfss`. 2018.

[War65]   S. L. Warner. "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias". In: *Journal of the American Statistical Association* 60.309 (1965). PMID: 12261830, pp. 63–69. eprint: `http://www.tandfonline.com/doi/pdf/10.1080/01621459.1965.10480775`. URL: `http://www.tandfonline.com/doi/abs/10.1080/01621459.1965.10480775`.

[WYGVZ17]   F. Wang, C. Yun, S. Goldwasser, V. Vaikuntanathan, and M. Zaharia. "Splinter: Practical Private Queries on Public Data." In: *NSDI*. 2017, pp. 299–313.

[Yao82]   A. C. Yao. "Protocols for secure computations". In: *FOCS*. 1982. URL: `doi.ieeecomputersociety.org/10.`

[Yao86]   A. C.-C. Yao. "How to generate and exchange secrets". In: *Foundations of Computer Science, 1986., 27th Annual Symposium on*. IEEE. 1986, pp. 162–167.