Zero Knowledge Arguments for Verifiable Sampling

César Sabater
MAGNET Team
Inria Lille - Nord Europe
59650 Villeneuve d'Ascq, France
cesar.sabater@inria.fr

Jan Ramon
MAGNET Team
Inria Lille - Nord Europe
59650 Villeneuve d'Ascq, France
jan.ramon@inria.fr

Abstract

In privacy-preserving machine learning, it is less obvious to verify correct behavior of participants because they are not supposed to reveal their inputs in cleartext to other participants. It is hence important to make federated machine learning robust against data poisoning and related attacks. While input data can be related to a distributed ledger (blockchain), a less studied input is formed by the random sampling parties perform. In this paper, we describe strategies based on zero knowledge proofs to allow parties to prove they perform sampling (and other computations) correctly. We sketch a number of alternative ways to implement our idea and provide some preliminary experimental results.

1 Introduction

Privacy preserving machine learning studies the learning of models without revealing sensitive data, e.g., personal data of individuals. While a wide variety of techniques are emerging to accomplish that goal, they often assume that participating agents are honest-but-curious, i.e., they assume participants try to infer information from the data they see but honestly follow the prescribed protocol. Unfortunately, in practice agents may be tempted to contribute incorrect data, as they don't need to disclose their inputs to the computation and may benefit from a biased end-result. Such behavior is called data poisoning. The goal of such an attack could be to influence the process such that the resulting machine learning model is biased in some way. Examples of this can occur in different scenarios. Consider a group of store owners who collaboratively learn a model predicting in which products a customer may be interested. They use privacy-preserving machine learning not revealing any of their customer data nor any intermediate results. Now some store owners may be tempted to introduce bias in the model. Intelligently biasing the collected statistics may make the model poor, not signaling to other store owners that a certain customer may be interested in their products, while the cheating store owner is the only one knowing the bias and able to correct for it to obtain good predictions. As real-world examples, approaches have been reported to exploit social navigation services [31] or influence Google's algorithm to learn traffic density [34].

Zero-knowledge proofs (ZKPs) are cryptographic techniques that allow a party to prove statements such as logical and arithmetic relations over private values. Such zero-knowledge proofs can be used to prove that a party in a federated machine learning effort performs its computations correctly, without disclosing its inputs or outputs. So if we could assume that all inputs and random numbers an algorithm uses are correct, then if we are provided a zero-knowledge proof we can trust that the output (the machine learning model) is correct, even if we don't know the details of the computation.

The issue of verifying the correctness of input data is common and non-trivial. One important approach consists of letting parties commit to their input data to ensure that the same values are used when feeding the same variables as input to other algorithms. Letting parties commit to data has applications in numerous other domains, e.g., in financial systems and blockchains [4, 29]. An option to limit the effect of incorrect input is to require a ZKP that the input belongs to the correct domain or that the inputs are consistent with each other. Fully guaranteeing input correctness may not always

Preprint. Under review.

be possible, but similarly it is also not always possible in the non-private federated learning setting either.

Key remaining challenges include (1) a need for progress in zero-knowledge proofs, e.g., for mathematical relationships important for machine learning, (2) a need to guarantee that parties correctly draw random numbers. For example, machine learning algorithms often need statistical relations and probability distributions, e.g., the Poisson distribution or the error function. Also, differential privacy often involves drawing random numbers to serve as noise, e.g., from the Laplace distribution or the Gaussian distribution. If an adversary could decide the value of every number he is supposed to draw randomly he could already bias some models significantly.

Methods to privately compute elementary functions using Secure Multiparty Computation have been studied in [1, 3, 12, 24, 26] and used for machine learning in [23]. While these secure multi-party compuation approaches don't disclose data, they assume that parties are honest (e.g., some require the community to trust that a small set of servers don't collude). Proofs of correct computation of training of models have recently started to be studied in [18, 35]. However, these rely on the use of integer-friendly activation functions and don't provide complete privacy and security guarantees. To the best of our knowledge, zero-knowledge proofs only have been applied in machine learning in a limited way. Also, we are not aware of protocols that allow a party to prove it has correctly generated a random Gaussian number.

To bridge this gap, we work towards verifiably correct sampling, i.e., protocols to let agents draw random elements from probability distributions and prove to other agents that they did so correctly. While doing so, we also provide advances in verifiability of relations involving transcendental functions, which independently serve as a building block in robust machine learning.

In particular, as a first contribution we extend a classic technique for computing trigonometric and exponential functions, CORDIC [33], towards a zero knowledge proof strategy for relationships containing such functions. Our second contribution is a protocol to collaboratively and provably correctly sample a random number from a Gaussian distribution. This protocol is based on the Box-Müller method. This has a direct application in the Gaussian Mechanism used in Differential Privacy. Our approximations achieve a precision of n fractional digits with $O(n^2)$ computations. Finally, in a third contribution we adopt an argument of database lookup, which can be used to prove correct sampling from arbitrary statistical distributions and to prove relationships more general than those covered by our first contribution. This strategy requires, for a database of size M, $O(M)$ preprocessing (probability distribution description) and a constant cost per lookup (sample). We first describe the problem, then sketch our ongoing work and conclude with ideas for future work.

2 Problem Statement

The notion of differential privacy [14] has become the gold standard to measure the extent to which an algorithm is private. In many applications, one wants to learn a model with training data of multiple parties, each of which doesn't want to reveal their own sensitive data to others. The simplest and most secure strategy to learn in a privacy-preserving way is to use Local Differential Privacy (LDP) [13, 20, 21, 22], where first all parties add noise to their own data before sharing it. A clear disadvantage is the large amount of noise (and the resulting poor accuracy of the learned model). In an other popular strategy, known as central differential privacy (CDP), all parties send their data to a trusted curator after which the latter computes the model and adds noise to it. While the amount of noise needed here is much smaller (and the model more accurate), the main objection against this approach is that one should be able to fully trust the curator. Various methods have been proposed to achieve an accuracy comparable to CDP without the need to trust a curator. These are usually based on encryption or shuffling [2, 8, 16, 17] to avoid that parties can see sensitive data of other parties or can learn which data belongs to which party. A major disadvantage shared by these methods (and for that matter also by LDP) is that parties can to some extent poison their contributed data or the noise they add to it without being detected. There exist a range of cryptographic tools to mitigate this new problem. For example, commitments [30] allow one to commit to a piece of data without revealing it. This technique is also used in blockchains for publicly registering information without revealing it. While a value underlying a commitment could still be wrong, committing implies that in all business a party does the same value must be used. As lying consistently over all data over a long time is clearly more difficult, it decreases the opportunity for data poisoning.

We study the less explored question of verifiable random number generation. Suppose we have an encrypted optimal model parameter $E(\theta)$ and that we still need to add noise to it before it can be

decrypted and published. For the simplicity of explanation we will assume that θ is a real number, but our approach can easily be generalized to the case that θ is a vector of real numbers. A common approach exploits partial homomorphic encryption: an encryption E is partially homomorphic if for all x_1 and x_2, $D(E(x_1) \otimes E(x_2)) = x_1 + x_2$ for the decryption function D and operation $\otimes$ in ciphertext space. Then, some party can sample noise η from some distribution $\mathcal{D}$ and compute $E(\theta + \eta) = E(\theta) \otimes E(\eta)$ after which decryption of the final result $E(\theta + \eta)$ can proceed. The main question now is how to ensure that the noise η can be kept secret while proving to all participating parties that η is correctly sampled from $\mathcal{D}$.

In our setting we assume that at least one party is honest but curious, and the others can collude. We will focus on procedures to let a party draw an element from a distribution without revealing it while proving that the sample was correctly drawn. We won't require that the drawing party itself doesn't learn the drawn element. In summary:

Problem statement. Given n parties $P_1, P_2 \dots P_n$, and a probability distribution $\mathcal{D}$, we want to find for a generic party P_i with $i \in \{1, \dots n\}$ a distributed algorithm A such that after its executions (i) a number $E(\eta)$ has been published, (ii) only P_i can infer the value of η and (iii) all parties are convinced that η was drawn from $\mathcal{D}$.

The problem is reasonably straightforward if $\mathcal{D}$ is the uniform distribution over the interval $[0, L)$ for some $L > 0$. Let E_i be a homomorphic encryption function for which only P_i has the corresponding decryption key. Let each party P_j generate a random number r_j uniformly over $[0, L)$. Let each party P_j, publish $E_i(r_j)$. Then, P_i can compute $r = \sum_j r_j \bmod L$, publish $E_i(r)$ and prove correct computation by the homomorphic property $E_i(\sum_j r_j) = \otimes_j E_i(r_j)$ and a zero knowledge proof for the modulo operation. It is easy to see that r is a uniformly sampled random number secret to all parties except P_i, if at least one party sampled a truly uniformly drawn number.

The problem becomes more challenging when $\mathcal{D}$ is not the uniform distribution, but is a normal distribution or a Laplace distribution as would be needed in the Gaussian Mechanism or Laplace Mechanism of differential privacy [15]. An approach for arbitrary distributions is known as the inversion method, which consists of sampling uniformly from the $(0, 1)$ interval and applying the inverse of the cumulative distribution function (CDF). Performing this in a verifiable way requires a closed form for the inverse CDF, or a sufficiently good and efficient approximation schema. In case $\mathcal{D}$ is the Gaussian distribution, a number of additional specialized methods are available:

- The Central Limit Theorem (CLT) approach, which consists of sampling repeatedly from a uniform distribution and computing the average, which is simple but requires $O(1/\Delta^2)$ time for a root mean squared error Δ.
- The Box-Müller method [6], that can obtain two Gaussian numbers from two uniform samples by the application of a closed form formula, but involves the computation of a square root, trigonometric functions and a logarithm.
- Rejection sampling methods, such as the polar version of Box-Müller [25] or the Ziggurat Method [27] are efficient and highly accurate. While the former avoids the computation of trigonometric functions and leads to an efficient verifiable implementation, the latter is uses several conditional branches which are expensive to prove in zero knowledge and requires an external method for sampling in the tails of the distribution.
- The inversion method for Gaussians involve the approximation of the inverse error function erf^{-1}, which can be done with rational functions or Taylor polynomials.
- The recursive method of Wallace [32] is very popular for its efficiency, but requires as input a vector of already generated Gaussian samples to generate an output vector of the same size. Furthermore, samples from input and output vectors are correlated, which deteriorates the statistical quality.

Generating uniformly distributed random numbers is cheap, and most above ideas for sampling from other distributions use uniform sampling as a building block, together with various ways to postprocess the resulting sample. Therefore we will discuss in the next section ways for P_i to prove to other agents that the performed postprocessing is correct.

3 Methodology Towards Verifiable Sampling

Cryptographic Tools. To prove properties over secret committed values, we use Zero Knowledge Proofs [19]. They allow a party to prove statements without revealing extra information. Here, we consider the combination of Σ-protocols [9] and the strong Fiat-Shamir heuristic [5] for its simplicity

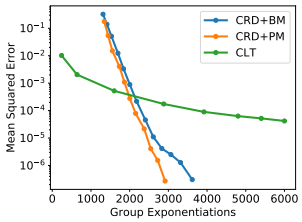

Figure 1: The required GEx for one sample against the MSE, for CRD+BM , CRD+PM and CLT.

and modularity. They can be used to prove arithmetic relations between private values [10] and and disjunctions and conjunctions of these [11]. To efficiently prove relations involving trigonometric, exponential, logarithmic and/or square root functions in fixed precision, we use CORDIC and its numerical approximation methods which only require simple operations. As opposed to some alternatives (see [28] for a comprehensive treatment) CORDIC [33] is an iterative method only using additions, bit-shifts and a few multiplications. It suffices to provide a Σ-protocol proving its correct execution. A naive implementation costs $O(n^2)$.

Statistical Distributions. As outlined in Section 2, proving correct sampling from a uniform distribution is relatively easy and we can leverage it towards verifiable sampling from other distributions. First, we provide an argument that we correctly draw an $y \sim \mathcal{N}(0,1)$. One strategy is to use the Box-Müller method [6], which defines in closed form a function $f : (0,1)^2 \to \mathbb{R}^2$ such that, if x and y are uniformly distributed, then the components of $f(x,y)$ follow a Gaussian distribution. f only requires one evaluation of $\sin$, $\cos$, $\ln$ and square root functions, for which we can use the CORDIC ZKP. Alternatively, the Polar Method[25], optimizes Box-Müller to avoid the computation of trigonometric functions using rejection sampling. We measured the amount of group exponentiations (GEx) needed to prove a Gaussian sample using the CORDIC combined with Box Müller (CRD+BM) and Polar Method (CRD+PM) approaches and compared them to an implementation using the CLT approach for several parameter values. For each setting we drew 10^7 samples and measured the Mean Squared Error (MSE) from the ideal Gaussian CDF, see Figure 1.

Now we consider the same ZKP but for an arbitrary statistical distribution $\mathcal{D}$. A general method to generate samples is by inversion of the cumulative probability function $f_{\mathcal{D}}$, which can be approximated via *table lookups*. It consist on publicly precomputing a sufficiently large set of points of $f_{\mathcal{D}}^{-1}$ and later retrieve them evaluate the function. The main challenge is then to prove the correct retrieval of an evaluation from its private index. For that, we use a well known Σ-protocol to prove set membership [7], i.e. $x \in S$ where x is private and S is public. If S is a set of table entries, this can be easily adapted to the proof of retrieval we need. The preprocessing of the technique is exponential in the required precision, but each sample requires a constant cost. Additionally, the preprocessing step can be reused by other parties.

4 Conclusion and Future Work

We have presented novel methods combining zero knowledge proofs and strategies to compute or sample from various functions to prove correct fixed precision calculation and correct sampling from Gaussian and arbitrary distributions.

Multiple directions of future work remain. We mainly plan to work on decreasing the cost of sampling. It would be interesting to investigate how intensively parties need to collaborate before it becomes affordable to use the inverse error function lookup-table approach described at the end of Section 3. We also want to experimentally compare with other numerical strategies which may offer a faster convergence in terms of ZKP cost, which may be different from the cost when using plaintext. We could also try to reduce the cost for larger amounts of random numbers by only randomly drawing a seed and then let parties prove they correctly use a (sufficiently affordable) random number generator.

Acknowledgements. We thank Andreas Peter for its useful feedback and suggestions.

References

[1] ALY, A., AND SMART, N. P. Benchmarking Privacy Preserving Scientific Operations. In *Applied Cryptography and Network Security* (Cham, 2019), R. H. Deng, V. Gauthier-Umaña, M. Ochoa, and M. Yung, Eds., Lecture Notes in Computer Science, Springer International Publishing, pp. 509–529.

[2] BALLE, B., BELL, J., GASCÓN, A., AND NISSIM, K. The Privacy Blanket of the Shuffle Model. In *Advances in Cryptology – CRYPTO 2019* (Cham, 2019), A. Boldyreva and D. Micciancio, Eds., Lecture Notes in Computer Science, Springer International Publishing, pp. 638–667.

[3] BAYATBABOLGHANI, F., BLANTON, M., ALIASGARI, M., AND GOODRICH, M. Secure fingerprint alignment and matching protocols. *arXiv preprint arXiv:1702.03379* (2017).

[4] BEN SASSON, E., CHIESA, A., GARMAN, C., GREEN, M., MIERS, I., TROMER, E., AND VIRZA, M. Zerocash: Decentralized Anonymous Payments from Bitcoin. In *2014 IEEE Symposium on Security and Privacy* (May 2014), pp. 459–474. ISSN: 2375-1207.

[5] BERNHARD, D., PEREIRA, O., AND WARINSCHI, B. How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In *Advances in Cryptology – ASIACRYPT 2012* (Berlin, Heidelberg, 2012), X. Wang and K. Sako, Eds., Lecture Notes in Computer Science, Springer, pp. 626–643.

[6] BOX, G. E. P., AND MULLER, M. E. A Note on the Generation of Random Normal Deviates. *Annals of Mathematical Statistics 29*, 2 (June 1958), 610–611. Publisher: Institute of Mathematical Statistics.

[7] CAMENISCH, J., CHAABOUNI, R., AND SHELAT, A. Efficient Protocols for Set Membership and Range Proofs. In *Advances in Cryptology - ASIACRYPT 2008* (Berlin, Heidelberg, 2008), J. Pieprzyk, Ed., Lecture Notes in Computer Science, Springer, pp. 234–252.

[8] CHEU, A., SMITH, A., ULLMAN, J., ZEBER, D., AND ZHILYAEV, M. Distributed Differential Privacy via Shuffling. In *Advances in Cryptology – EUROCRYPT 2019* (Cham, 2019), Y. Ishai and V. Rijmen, Eds., Lecture Notes in Computer Science, Springer International Publishing, pp. 375–403.

[9] CRAMER, R. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, University of Amsterdam, Jan. 1997.

[10] CRAMER, R., AND DAMGÅRD, I. Zero-knowledge proofs for finite field arithmetic, or: Can zero-knowledge be for free? In *Advances in Cryptology — CRYPTO '98* (Berlin, Heidelberg, 1998), H. Krawczyk, Ed., Lecture Notes in Computer Science, Springer, pp. 424–441.

[11] CRAMER, R., DAMGÅRD, I., AND SCHOENMAKERS, B. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *Advances in Cryptology — CRYPTO '94* (Berlin, Heidelberg, 1994), Y. G. Desmedt, Ed., Lecture Notes in Computer Science, Springer, pp. 174–187.

[12] DIMITROV, V., KERIK, L., KRIPS, T., RANDMETS, J., AND WILLEMSON, J. Alternative Implementations of Secure Real Numbers. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, Oct. 2016), CCS '16, Association for Computing Machinery, pp. 553–564.

[13] DUCHI, J. C., JORDAN, M. I., AND WAINWRIGHT, M. J. Local Privacy and Statistical Minimax Rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science* (Oct. 2013), pp. 429–438. ISSN: 0272-5428.

[14] DWORK, C. Differential Privacy. In *ICALP* (2006).

[15] DWORK, C., AND ROTH, A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science 9*, 3-4 (2014), 211–407.

[16] ERLINGSSON, L., FELDMAN, V., MIRONOV, I., RAGHUNATHAN, A., TALWAR, K., AND THAKURTA, A. Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity. In *Proceedings of the 2019 Annual ACM-SIAM Symposium on Discrete Algorithms*, Proceedings. Society for Industrial and Applied Mathematics, Jan. 2019, pp. 2468–2479.

[17] GHAZI, B., PAGH, R., AND VELINGKER, A. Scalable and Differentially Private Distributed Aggregation in the Shuffled Model. *arXiv:1906.08320 [cs, stat]* (Dec. 2019). arXiv: 1906.08320.

[18] GHODSI, Z., GU, T., AND GARG, S. SafetyNets: verifiable execution of deep neural networks on an untrusted cloud. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, Dec. 2017), NIPS'17, Curran Associates Inc., pp. 4675–4684.

[19] GOLDWASSER, S., MICALI, S., AND RACKOFF, C. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing 18*, 1 (Feb. 1989), 186–208. Publisher: Society for Industrial and Applied Mathematics.

[20] KAIROUZ, P., OH, S., AND VISWANATH, P. Secure Multi-party Differential Privacy. In *Advances in Neural Information Processing Systems* (2015), pp. 1999–2007.

[21] KAIROUZ, P., OH, S., AND VISWANATH, P. Extremal Mechanisms for Local Differential Privacy. *Journal of Machine Learning Research 17* (2016), 1–51.

[22] KASIVISWANATHAN, S. P., LEE, H. K., NISSIM, K., RASKHODNIKOVA, S., AND SMITH, A. What Can We Learn Privately? *SIAM Journal on Computing 40*, 3 (Jan. 2011), 793–826. Publisher: Society for Industrial and Applied Mathematics.

[23] KELLER, M., AND SUN, K. Effectiveness of MPC-friendly Softmax Replacement. *arXiv:2011.11202 [cs]* (Nov. 2020). arXiv: 2011.11202.

[24] KERIK, L., LAUD, P., AND RANDMETS, J. Optimizing MPC for robust and scalable integer and floating-point arithmetic. In *International Conference on Financial Cryptography and Data Security* (2016), Springer, pp. 271–287.

[25] KNOP, R. Remark on algorithm 334 [G5]: normal random deviates. *Communications of the ACM 12*, 5 (1969), 281. Publisher: ACM New York, NY, USA.

[26] LIEDEL, M. Secure Distributed Computation of the Square Root and Applications. In *Information Security Practice and Experience* (Berlin, Heidelberg, 2012), M. D. Ryan, B. Smyth, and G. Wang, Eds., Lecture Notes in Computer Science, Springer, pp. 277–288.

[27] MARSAGLIA, G., AND TSANG, W. W. The ziggurat method for generating random variables. *Journal of statistical software 5*, 8 (2000), 1–7.

[28] MULLER, J.-M. *Elementary Functions: Algorithms and Implementation*, 3 ed. Birkhäuser Basel, 2016.

[29] NARULA, N., VASQUEZ, W., AND VIRZA, M. zkledger: Privacy-preserving auditing for distributed ledgers. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)* (Renton, WA, Apr. 2018), USENIX Association, pp. 65–80.

[30] PEDERSEN, T. P. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Advances in Cryptology — CRYPTO '91* (Berlin, Heidelberg, 1992), J. Feigenbaum, Ed., Lecture Notes in Computer Science, Springer, pp. 129–140.

[31] SINAI, M. B., PARTUSH, N., YADID, S., AND YAHAV, E. Exploiting social navigation. *Arxiv 1410.0151* (2014).

[32] WALLACE, C. S. Fast pseudorandom generators for normal and exponential variates. *ACM Transactions on Mathematical Software 22*, 1 (Mar. 1996), 119–127.

[33] WALTHER, J. S. A unified algorithm for elementary functions. In *Proceedings of the May 18-20, 1971, spring joint computer conference* (New York, NY, USA, May 1971), AFIPS '71 (Spring), Association for Computing Machinery, pp. 379–385.

[34] WECKERT, S. Google maps hacks. `http://www.simonweckert.com/googlemapshacks.html`.

[35] ZHAO, L., WANG, Q., WANG, C., LI, Q., SHEN, C., LIN, X., HU, S., AND DU, M. VeriML: Enabling Integrity Assurances and Fair Payments for Machine Learning as a Service. *arXiv:1909.06961 [cs]* (Sept. 2019). arXiv: 1909.06961.