Low-power and Computing-free Privacy Design for IoT Systems

Hui Sun[†], Kyle Mooney[*], Mario Renteria-Pinon[*], Tingxiang Ji[†], Hritom Das[$], Na Gong[*] and Jianqing Liu[†]
[†]Department of CS, NC State University, Raleigh, USA 27606.
[*]Department of ECE, University of South Alabama, Mobile, USA 36688.
[$]Department of EECS, University of Tennessee, Knoxville, USA 37996.
[†]{hsun26, tji2, jliu96}@ncsu.edu [*]{kam1827, mrenteria, nagong}@southalabama.edu [$]hdas@utk.edu

***Abstract*—Most IoT devices operate in environments with limited bandwidth and power, performing real-time dedicated functions. The sensitive nature of data collected by these devices can compromise the privacy of individuals under surveillance. Unfortunately, protecting IoT data becomes challenging when handling complex, unstructured data such as video, or semi-structured data like URLs and partially labeled information. Furthermore, privacy protection methods that do not align with the IoT devices' real-time and low-power characteristics can be counterproductive. In this paper, we propose a memory-based approach to apply local differential privacy to semi-structured data on IoT devices. Our technique uses a low-power, processing-in-memory architecture to introduce noise to sensitive data during storage, bypassing CPU processing and conserving energy. This design also allows data curators to accurately derive statistical information from the noisy data prepared by IoT devices.**

***Index Terms*—local differential privacy, static random-access memory, low-power, hardware-software co-design.**

I. Introduction

The proliferation of IoT devices has led to the extensive collection of data for analytical purposes. IoT devices capture real-world, real-time data from everyday human activities through various sensors and cameras, elevating the importance of data privacy. One robust solution that has emerged is differential privacy, a mathematical framework designed to quantify and control the privacy loss associated with the release of statistical data. This concept, introduced by Dwork et al., [1] has revolutionized the way organizations approach data privacy, providing a systematic method to add random noise to the data in such a way that the privacy of individual data entries is maintained while still allowing for accurate aggregate data analysis. The adoption of differential privacy by major tech companies, such as Google [2] and Microsoft [3], highlights its significance and effectiveness in today's digital landscape, where data security and privacy are of utmost concern.

Building on the concept of differential privacy, a decentralized approach called Local Differential Privacy (LDP) further enhances privacy guarantees by adding noise to data before it leaves the user's device. Unfortunately, two major concerns arise when implementing LDP in small IoT devices: (1) the intrinsic characteristics of IoT devices, including limited network bandwidth and constrained CPU capabilities, pose significant challenges in implementing LDP in such a small device; and (2) although differential privacy preserves the statistical results of data elements, it often results in the loss of latent categorical information embedded within semi-structured data. For instance, URLs is a semi-structured data with explicit domain suffixes (e.g., .com, .org, and .edu) as the latent categorical information; while IP addresses sharing the same higher bits (i.e., the same network ID) is another example of semi-structured data with implicit latent categorical information. LDP designs for such semi-structured data should therefore be considerate to avoid losing essential categorical information while still maintaining robust data privacy, which poses technical challenges.

Recognizing the challenges posed by the intrinsic characteristics of IoT devices in implementing LDP, a recent study by Liu et al. [4] introduces `SRAM_DP`, an innovative static random access memory (SRAM) architecture to realize LDP by design. This design leverages the characteristic failure of SRAM cells at reduced voltages to introduce LDP noises to stored data. This approach, which embodies the "privacy by design" philosophy [5], is power saving and CPU-free in contrast to software-based LDP methods. In this paper, we aim to optimize `SRAM_DP` in two fronts.

1) From the hardware design perspective, we propose a new SRAM design using custom design 6T array with multiple supply voltages (Vdd) to support LDP, greatly reducing silicon area overhead compared with `SRAM_DP`. Specifically, 6T columns with a lower Vdd are used to inject noise and columns with a higher Vdd are used to store (re-)shuffle pattern information. As compared to the state-of-the art, the proposed new memory design saves implementation cost, while enabling runtime adaptation and full flexibility of wide failure range for LDP.
2) From the software design perspective, we develop a new encoding method, which enables IoT devices to flexibly allocate their privacy budgets between the data and the categorical information embedded within it, all under a fixed total privacy budget. Furthermore, our design enables data curators to accurately gather data statistics, notably the categorical information. This ensures that even after data has been perturbed for privacy, it still try to retains a close relationship with other data where they

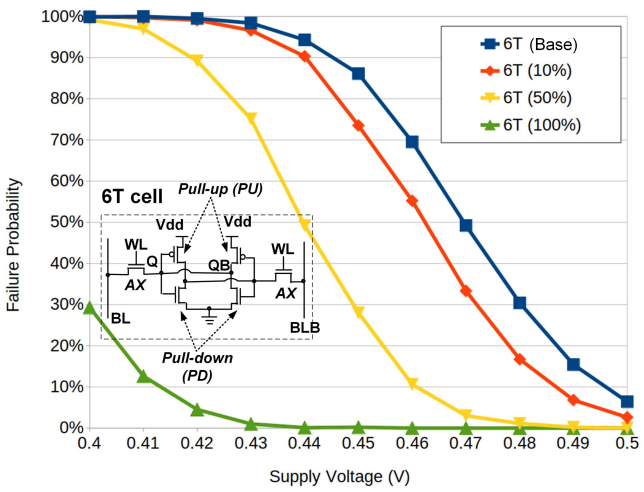


Fig. 1: Failure characteristics of 130nm 6T cells.

share similar characteristics. This balance is essential for maintaining data integrity and relevance, crucial elements for the aforementioned fields.

II. Privacy by Design in SRAM Memory

A. Preliminary Knowledge

SRAM has been the workhorse for embedded memory design to all electronics devices [6]. In the past decade, low-power SRAM designs through reducing supply voltage have been widely investigated. Among different designs, 6T is the most widely used SRAM cell structure due to its super area efficiency, particularly for IoT devices. However, as technology continues to scale due to the process variation, the failure probability of 6T cells grows significantly as voltage scales down. Fig. 1 shows the failure rates of four different sized 6T cells using the 130nm CMOS technology from SkyWater technology. All the simulations performed with NgSpice software. In our analysis, 1,000 Monte Carlo simulations were conducted to obtain the failure rates at different voltages at its worst process corner (fast NMOS and slow PMOS for read operations) [7]. The variations in multiple process parameters including transistor gate oxide thickness, threshold voltage, and sub-threshold voltage offset, have been considered based on the model provided by the fab. Also, to study the impact of transistor sizing on the failure probability of 6T, the sizing of the transistor ratios was varied based on the base cell. Specifically, the sizing of the base 6T cell was designed as follows: access transistors (AX) = 0.42um/0.15um, pull-up (PU) = 0.55um/0.15um, and pull-down (PD) = 1.26um/0.15um; based on it, a 10%, 50%, and 100% increase in each device's width-length ratio was applied. It can be seen from Fig. 1 that the failure rate of a specific cell grows as voltage is reduced. Also, at the same voltage, the failure rate decreases as the size of 6T increases.

B. Principle of SRAM_DP

Consider that IoT devices collect and locally store sensitive semi-structured data, which implicitly or explicitly contains category information. This category information can naturally sort the data into different groups, forming a dataset $D = \bigcup_{k=1}^{K} D_k$ where D_k's are the non-overlapping subsets of D curating all the data belonging to the k^{th} group, i.e., D_k=$\{B_j | 1 \leq j \leq |D_k|\}$. Any data can be encoded into a binary string, defined as $B = \{b_1, ...b_m, b_{m+1}, ...b_{m+n}\}$. In our design, the first m bits, referred to as "label bits", explicitly encode the categorical information; while the subsequent n bits, referred to as "data bits", encode the normal data information. For the label bits, we apply ordinal encoding which requires $m = \lceil \log_2(K) \rceil$ bits for the full representation of K groups. While for the data bits, they are encoded using the number of 1s. That is to say, for the data belonging to the same group, they are encoded to carry the same number of 1s but differing to each other in bit locations. The motivation of this design is to let the same categorical data embeds their group information for the subsequent data recovery by the data curator. With this encoding scheme, the length of data bits n will depend on the maximum number of data elements across all groups; specifically, n must pick

$$\min \left\{ n \in \mathbb{Z}^+ : \binom{n}{\lceil n/2 \rceil} \geq \max_{1 \leq k \leq K} |D_k| \right\} \tag{1}$$

In the encoding process, for each binary string in group D_k, we randomly select a_k bits, i.e., a_k is the number of 1s, from the "data bits" and set these bits to '1', while the remaining bits are set to '0'. The value a_k depends on the number of data elements in group D_k, i.e., $|D_k|$. Consequently, a_k for any $1 \leq k \leq K$ should pick

$$\min \left\{ a_k \in \mathbb{Z}^+ : \binom{n}{a_k} \geq |D_k| \right\}, \ 1 \leq k \leq K. \tag{2}$$

This implies that the number of bit-location permutations using a_k bits out of n bits is sufficient to uniquely encode $|D_k|$ data elements. Note that while the data within the same group can be completely differentiated by the n "data bits", the data from different groups could be encoded into the same "data bits" but such collision can be resolved when combined with the m "label bits".

Following the above encoding principle, an IoT device will encode its collected data into a bit string and store it in the device's SRAM cells $C = \{c_1, ..., c_m, c_{m+1}, ..., c_{m+n}\}$. Upon reporting their information to a data curator, the binary string read out as $O = \{o_1, ..., o_m, o_{m+1}, ..., o_{m+n}\}$ from a IoT device's SRAM which has been subjected to LDP noise.

By changing the voltage v of a IoT device's SRAM, the failure rates of SRAM cells can be adjusted. The mathematical outcome of this step for a bit is

$$o_i = \begin{cases} 0 & \text{with probability } \frac{1}{2} f_i \\ 1 & \text{with probability } \frac{1}{2} f_i \\ b_i & \text{with probability } 1 - f_i \end{cases}$$

Specifically, a bit $b_i \in B$ will maintain its original value with a probability of $1 - f_i$ and will be in an uncertain state with a probability of f_i. This fits perfectly with the randomized response mechanism [8] mechanism, originally designed to enable the collection of sensitive data while maintaining the privacy of individual respondents, and gradually becomes a classical implementation of LDP. In reality, the data curator will carefully choose f_i to meet the analysts' requirements. Here, we define that "label bits" have the same probability f' while the "data bits" have the same probability f''. Given a privacy budget $\epsilon = \epsilon_1 + \epsilon_2$, where ϵ_1 denotes the privacy budget for "label bits", and ϵ_2 denotes the privacy budget for "data bits". Our mechanism will divide it to two parts for implicit information encoding and data itself, respectively.

After receiving the encoding map and privacy budget allocation, IoT devices store the data in SRAM and adjust its Vdd's accordingly to inject noise into failed cell positions when reporting to server.

C. Hardware Architecture of SRAM_DP

The proposed novel SRAM memory architecture shown in Fig. 2 is based on the design by Liu et al. [4]. Here, different supply voltages are used for additional noise injection to the data stored in the SRAM_DP by changing the voltage on the cells in different columns. As shown in Fig. 2, the cells in different columns can be powered by different supply voltages and each voltage can be determined by the target failure rate. Similar to [4], the memory peripherals, such as row decoder, data shuffler and re-shuffler, and random generator, will use nominal supply voltage to support read and write operations. As a result, the proposed architecture enables full flexibility for noise injection using different memory columns. It is important to note that, as another major advantage as compared to [4], the proposed memory can adapt noise injection by adjusting the supply voltages in real time, which has potential to support different applications using the same memory chip.

III. Performance Evaluation

A. Evaluation Metrics

Categorization Success Rate (CSR): In scenarios where analysts prioritize category statistics over the distribution of the data itself, the objective is to ensure that even if the data has been perturbed for privacy, it still has a higher probability of being accurately categorized into a group with the same categorical information after recovery. This concept is known as 'categorization success'. Therefore, we define the CSR as the probability that data, once subjected to noise addition and subsequently recovered by the data curator, is accurately categorized into the correct label group after recovery.

Mean Square Error (MSE): Specifically, we use MSE to evaluate the fidelity of the reconstructed normalized frequency distribution of dataset elements compared to the original distribution. This metric provides a clear indication of how closely the reconstructed data mirrors the original data in terms of element frequency, offering a straightforward and effective measure of reconstruction accuracy. The lower the MSE, the closer the reconstructed distribution is to the original, indicating a more accurate approximation

B. Input Data

We sampled 10,000 times from our candidate dataset Ω, which consists of 50 data elements, each uniquely labeled with one of 8 different labels. Sampling followed three common real-world distributions: exponential($\lambda = 0.2041$), Gaussian($\mu = 24.5$, $\sigma = 5$), Zipf($s = 1.5$). This process resulted in three datasets, each containing 10,000 9-bit binary strings as input data.

C. Recovery Algorithm

The Expectation-Maximization (EM) algorithm is used to estimate parameters in probabilistic models. In our context, it is suitable for identifying the most likely candidate B^* that, after noise has been added by our designed SRAM, corresponds to the observed binary string O.

D. Privacy Analysis

Liu et al. [4] have proven that the design of SRAM_DP, which incorporates noise addition in hardware, satisfies LDP. When a specific voltage v is applied to the cells, the relationship between cell failure rate and voltage is illustrated in Fig. 1. It is evident that the failure rate determines whether a bit $b_i \in B$ can retain its original value. The probability of a bit maintaining its original value is given by:

$$P(o_i = b_i) = 1 - p_{v,c_i}.$$

Here, p_{v,c_i} represents the probability of cell c_i fail at a given voltage v. By utilizing the characteristic failure rate of the chip's cells, we adjust the voltages applied to these cells to meet analysts' requirements.

To meet a given privacy budget, ϵ, we first divide the budget into ϵ_1 and ϵ_2 for the data elements and the embedded information within the data, respectively, as specified by the data curator. For any given bit b_i, with a failure rate f_i, this bit contributes $\ln(\frac{1-\frac{1}{2}f_i}{\frac{1}{2}f_i})$ to ϵ, as per [2]. For simplicity, we assume that the first m bits each have the same bit failure rate f' and the last n bits each have the same bit failure rate f''. Therefore, the relationship is established as $\epsilon = \epsilon_1 + \epsilon_2 = m \cdot \ln(\frac{1-\frac{1}{2}f'}{\frac{1}{2}f'}) + n \cdot \ln(\frac{1-\frac{1}{2}f''}{\frac{1}{2}f''})$ where $\epsilon_1 = m \cdot \ln(\frac{1-\frac{1}{2}f'}{\frac{1}{2}f'})$ and $\epsilon_2 = n \cdot \ln(\frac{1-\frac{1}{2}f''}{\frac{1}{2}f''})$ as shown in Fig. 3.

E. Simulation Results and Analysis

Simulation Setup: To simulate the entire procedure of SRAM_DP and statistical re-construction, we use PyCharm Community Edition 2023.2.1 to implement the procedure in section II-B, and MATLAB_R2024a for the statistical recovery. Specifically, after constructing the encoding map based on the candidate dataset, we sample 10,000 times following a specific distribution and encode those 10,000 elements into binary strings. Given a specific privacy budget ϵ, we construct an independent and identically distributed (i.i.d.) Bernoulli failure rate vector to mimic the cell fail in hardware. We keep

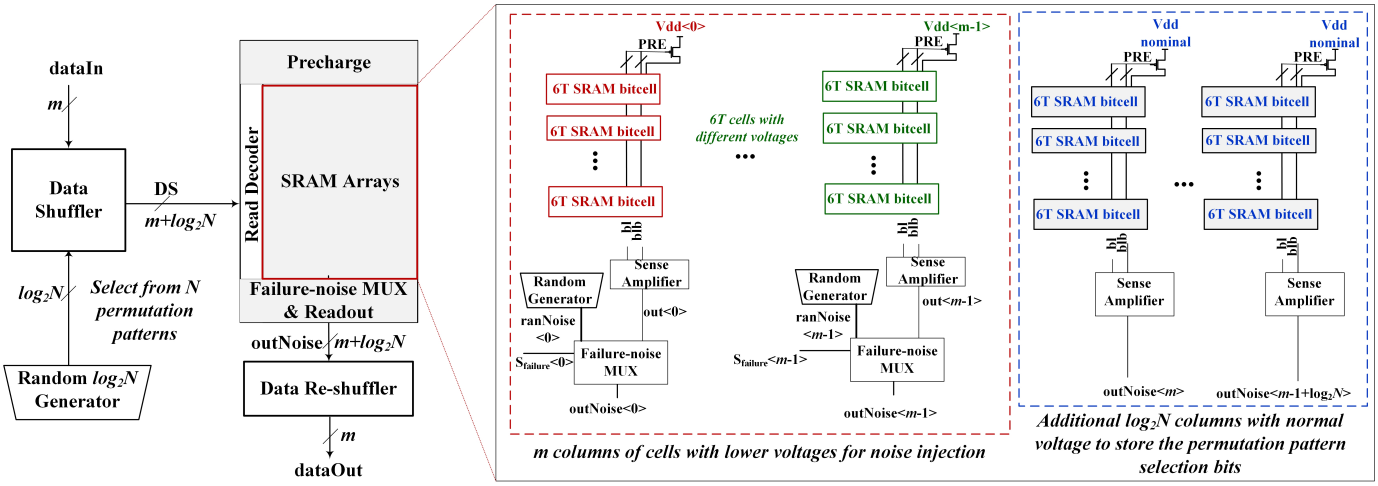


Fig. 2: Hardware Architecture of the proposed SRAM_DP with 6T cells. Each column of 6T cells can be powered by its own supply voltage support noise injection or pattern selection bits storage.

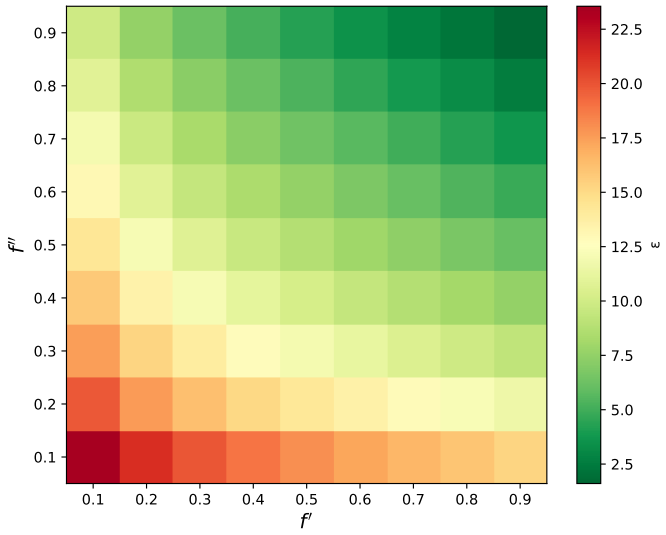


Fig. 3: Relationship between f', f'' and ϵ.

the unchanged bits as they are and replace the changed bits with random 0s or 1s to report them to the data curator. For the statistical reconstruction phase, we use an EM algorithm.

CSR Analysis: The straightforward method for encoding a non-numeric dataset is through binary encoding, which we refer to as the baseline here. This method requires at least $\log_2 N$ bits to encode a dataset with N elements. For our baseline method, N is 6. We compare our encoding method with this baseline, and the results are shown in TABLE I. Given an $\epsilon = 9$, the baseline requires 6 bits, with each bit having a failure rate of 0.36485 while our method requires 8 bits, with each bit having a failure rate of 0.49000. Apparently, our method outperforms the baseline method in terms of CSR, with a similar MSE of the recovery distribution. We further elucidate, later in this section, how the flexible failure rate of our method contributes to these CSR enhancements under a consistent ϵ compared to a uniform bit failure rate.

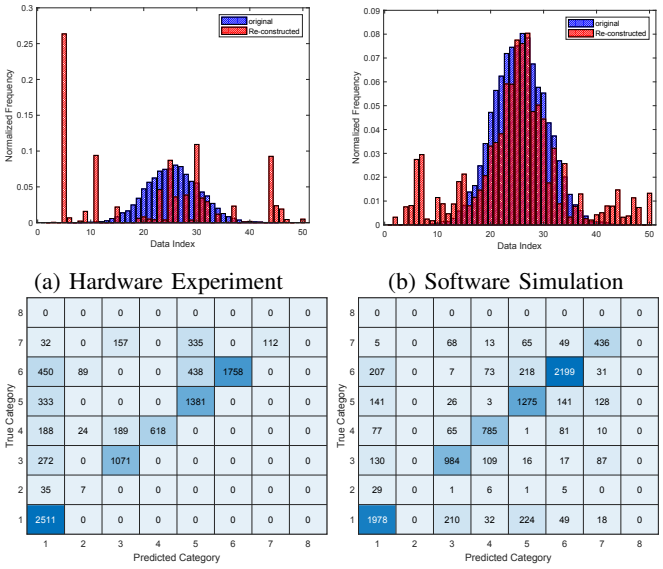
TABLE I: CSR/MSE of different failure rate patterns.

Failure Rate Patterns	CSR (%)	MSE (10^{-5})	CSR Δ%	MSE $\Delta(10^{-5})$
baseline, Exp	46.48%	2.6384	-	-
baseline, Gauss	52.01%	0.9515	-	-
baseline, Zipf	69.21%	3.4249	-	-
our method, Exp	55.24%	2.2099	+8.76%	-0.4285
our method, Gauss	54.03%	1.9139	+2.02%	+0.9624
our method, Zipf	72.30%	2.3304	+3.09%	-1.0945

Data Re-construction: Fig. 5 shows the CSR and MSE results. The results of $\epsilon = 3$ and 6 truncated because the ϵ values are so small so that, even if we allocated the entire privacy budget to 'label bits', achieving the required failure rate is not feasible. From Fig. 5a to Fig. 5c, we can observe that a higher ϵ results in a higher CSR. Our method performs relatively better with Zipf distributions, which simulate real-world datasets where a few elements with the same label occur extremely frequently. From Fig. 5d to Fig. 5f, we observe that a higher ϵ results in a lower MSE which means we get a better histogram frequency recovery. This situation is expected because a higher ϵ implies weaker privacy protection. Typically, in the industrial sector, ϵ is chosen to be on the order of a few tens. Comparing the results between CSR and MSE, we find that a higher CSR typically comes at the expense of MSE.

F. Experiment Results and Analysis

After comparing the results from the software simulation and the hardware experiment, as shown in Fig. 4a and 4b, we found significant differences in the MSE. The MSE for the hardware experiment is 239.13×10^{-5}, whereas for the software simulation, it is 13.59×10^{-5}, as depicted in Fig. 4c and 4d. The hardware results are notably less favorable, likely



(a) Hardware Experiment (b) Software Simulation

(c) Hardware Experiment (d) Software Simulation

Fig. 4: Comparative result of hardware and software.

due to discrepancies in failure rate patterns between the actual hardware and the simulation. Improvements are needed in technology to accurately control voltage and achieve effective random noise addition. Despite this, the Correct Success Rate (CSR) for both the hardware experiment and software simulation are closely matched at 76.57% and 75.58%, respectively. This demonstrates our success in maintaining similarity with other data in the same category, even after data perturbation for privacy.

G. Hardware Results

Layout design: Fig. 6 shows the layout design of 6T SRAM with 32 words x 10 bits using the Skywater 130nm technology. Three supply voltages are used for the implementation of different failure rates: $Vdd_{nominal}$ = 1.8V is applied to bit cell columns related to the pattern selection bits, Vdd_1 = 486mV is administered to the bit cells related to the 3 MSBs, while Vdd_2 = 455mV is used for the remaining 5 LSBs. In our layout design process, in order to avoid the silicon area overhead to integrate columns with different supply voltages, the SRAM 6T cells are laid out on a mirrored fashion and therefore cells in the same column can share the same supply voltage interconnect, but cells in the save row (word) can use different supply voltages. Accordingly, due to the column distribution of the bit cells, no additional area is introduced by the use of these 3 different voltage supply lines. Since the proposed memory enables efficient integration of 6T cells without the need of 8T cells, it can enable significant silicon area reduction as compared to the state-of-the art [5].

Power consumption: Two power consumption measurements are conducted based on post-layout simulations including parasitic extraction. First, only the nominal voltage is used as the baseline design (i.e. $Vdd_{nominal}$ = Vdd_1 = Vdd_2 = 1.8V), and next a test case using the voltages mentioned above (i.e., Vdd_1 = 486mV and Vdd_2 = 455 mV) to enable the target failure rates to support DP. For each test case, the average power consumption is measured to a random word with the clock period for each operation as 100ns. Specifically, the selected word is initialized to the binary value "0110100101", and then "1000001111" is written to the same word, followed by an immediate read operation from the same word. Therefore, all read/write memory operations are equally included (i.e., reading '0' and '1', and writing '0' to '0', '0' to '1', 1' to '0', and '1' to '1') on all bit-lines corresponding to Vdd_1 and Vdd_2. The results show that the average power consumption in the baseline design is $214\mu W$, while the average power consumption when introducing noise is $93\mu W$, providing a 56.54% power savings.

IV. Related Work

In the realm of differential privacy, recent research primarily concentrates on developing cross-disciplinary DP strategies tailored for a broad range of applications [9], [10], [11]. Despite the rich theoretical foundation of DP, practical implementations of these methods remain poorly explored. Cynthia Dwork, the pioneer of differential privacy, appeals for more attention to DP implementations in her paper [12].

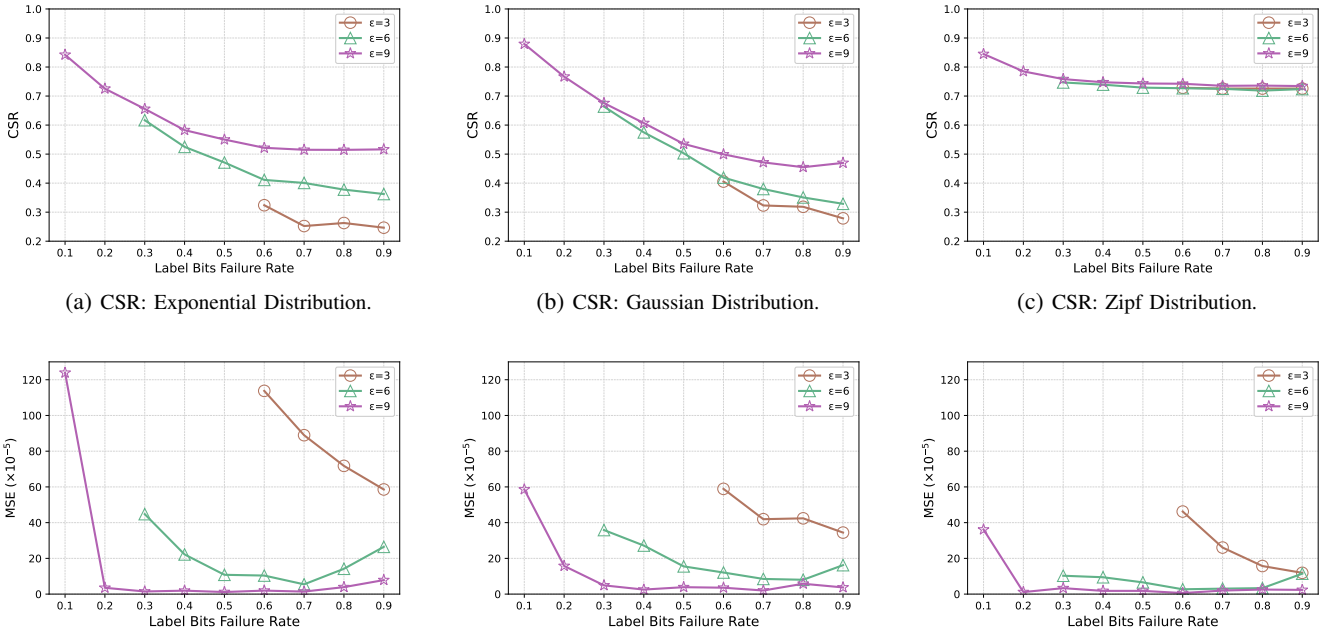
Fortunately, in recent years, a small number of researchers have begun exploring the use of specific hardware characteristics to implement DP. For instance, Yang et al. [13] developed a method to inject DP Gaussian noise into deep learning model training by reducing the supply voltage, thereby inducing SRAM bit errors. Similarly, Fu et al. [14] harnessed the natural Gaussian noise arising from imperfections in memristor operations to construct a differentially private deep learning model. Additionally, [4] designed `SRAM_DP`, which fully implements true randomness in hardware and is less susceptible to manipulations by attackers who might control the software-based randomizer, as discussed in [15]. These approaches showcase innovative strategies for harnessing inherent hardware properties to enhance privacy in IoT devices.

V. Conclusion

In this work, we have developed a new `SRAM_DP` design using 6T cells with multiple supply voltages, which enables lower implementation cost and higher flexibility as compared to the state-of-the art. We have also introduced a new encoding method that supports LDP for semi-structured data and allows data curators to flexibly adjust privacy allocations to meet various analytical needs. The analytical and experimental results confirmed the LDP conformation, accuracy in statistics recovery, and system power saving.

VI. Acknowledgement

The work by J. Liu was supported in part by the National Science Foundation under grants ECCS-2312738 and CNS-2247273. The work by N. Gong was supported in part by the National Science Foundation under grants CNS-2211215 and OIA-2218046.



(a) CSR: Exponential Distribution. (b) CSR: Gaussian Distribution. (c) CSR: Zipf Distribution.

(d) MSE: Exponential Distribution. (e) MSE: Gaussian Distribution. (f) MSE: Zipf Distribution.

Fig. 5: Given three different privacy budget $\epsilon = \{3, 6, 9\}$, the CSR and MSE are plotted based on different privacy allocation strategies, which reflect on the failure rate patterns of 'label bits'. We enumerate all possible label failure rates with a step size of 0.1; accordingly, the failure rates of 'data bits' will change.

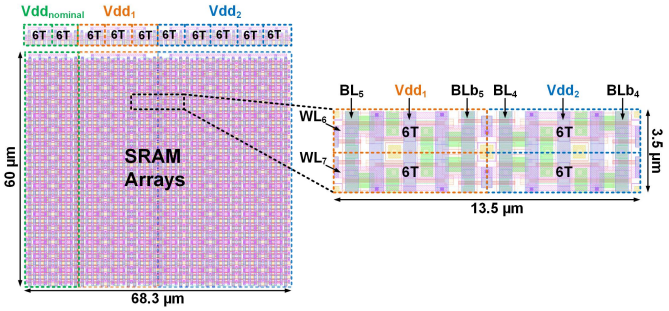


Fig. 6: Custom layout of the proposed SRAM array for DP. Different voltage supplies per bit column are used for noise injection. In each 6T cell: BL_i is bit-line, BLb_i is bit-line bar, WL_j is word-line, where subscripts i and j represent the column and row position, respectively.

References

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[2] Ú. 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*. ACM, 2014, pp. 1054–1067.

[3] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[4] J. Liu, N. Gong, and H. Das, "Two birds with one stone: Differential privacy by low-power sram memory," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–14, 2024.

[5] J. Liu and N. Gong, "Privacy by memory design: Visions and open problems," *IEEE Micro*, vol. 44, no. 1, pp. 49–58, 2024.

[6] Y. Xu, H. Das, Y. Gong, and N. Gong, "On mathematical models of optimal video memory design," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 256–266, 2019.

[7] J. Edstrom, Y. Gong, D. Chen, J. Wang, and N. Gong, "Data-driven intelligent efficient synaptic storage for deep learning," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 12, pp. 1412–1416, 2017.

[8] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, 1965.

[9] J. Liu, C. Zhang, B. Lorenzo, and Y. Fang, "Dpavatar: A real-time location protection framework for incumbent users in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 552–565, 2019.

[10] X. Pei, X. Deng, S. Tian, J. Liu, and K. Xue, "Privacy-enhanced graph neural network for decentralized local graphs," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1614–1629, 2023.

[11] N. Gai, K. Xue, B. Zhu, J. Yang, J. Liu, and D. He, "An efficient data aggregation scheme with local differential privacy in smart grid," *Digital Communications and Networks*, vol. 8, no. 3, pp. 333–342, 2022.

[12] C. Dwork, N. Kohli, and D. Mulligan, "Differential privacy in practice: Expose your epsilons!" *Journal of Privacy and Confidentiality*, vol. 9, no. 2, Oct. 2019.

[13] L. Yang and B. Murmann, "Approximate sram for energy-efficient, privacy-preserving convolutional neural networks," in *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2017, pp. 689–694.

[14] J. Fu, Z. Liao, J. Liu, S. C. Smith, and J. Wang, "Memristor-based variation-enabled differentially private learning systems for edge computing in iot," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9672–9682, 2020.

[15] X. Li, N. Li, W. Sun, N. Z. Gong, and H. Li, "Fine-grained poisoning attack to local differential privacy protocols for mean and variance estimation," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 1739–1756.