FEATURE ARTICLE

Privacy by Memory Design: Visions and Open Problems

Jianqing Liu, *North Carolina State University, Raleigh, NC, 27606, USA*
Na Gong, *University of South Alabama, Mobile, AL, 26688, USA*

The threat to data privacy has never been more alarming than it is today. Among existing privacy-enhancing technologies, differential privacy (DP) is widely accepted as the de facto standard for privacy preservation. Yet, the software-based implementation of DP mechanisms is neither friendly for lightweight devices nor secure against side-channel attacks. In this article, we propose a first-of-its-kind design regime that realizes DP in hardware memories. The salient feature of this novel design lies in its transformation of the notorious memory noises at subnominal voltages into the desired DP noises, thereby achieving power savings and privacy preservation simultaneously: a "win-win" outcome. We demonstrate the feasibility of this design regime using a 1-Kb memory prototype based on 45-nm technology. For future prospects, a research road map that contains open research problems is delineated for the broad research community.

Today, the collection of sensitive data is immense, and it poses a serious threat to our people and society. Among existing privacy-enhancing technologies (PETs), differential privacy (DP) has been widely embraced for privacy preservation since its formal inception by Dwork et al.[1] in 2006. The distinctive property of DP resides in its rigorous assurance of individual's data privacy while preserving the general statistical characteristics of the data, i.e., harmonizing data privacy and usability.

The flourishing research efforts in the past decade have nearly stretched DP to its maximum potential, yet, we observe that the implementation aspect of DP has seldom been highlighted. This is largely due to the prevailing presumption that a software-based algorithm can easily and reliably add a secret number (i.e., noise) sampled from a probability distribution to the true value. However, software-based realization of DP mechanisms bear many issues. 1) The sampling and adding procedures in software are converted to floating-point arithmetic in a device's operating system (OS), which may deviate markedly from the DP's mathematical abstraction due to the rounding rules and compounding errors in floating-point arithmetic. This issue has reportedly bred side-channel attacks that undermine the privacy-preserving promise of DP, as evidenced by Mironov[2]. 2) The DP noise sampling process requires function calls in the high-layer protocol stack, which are easily supported by legacy OSs such as iPhone operating system but may not stand for "slim" OSs in lightweight devices such as embedded sensors. These slim OSs are mostly vendor- and application-specific, which also makes the realization of DP mechanisms difficult to scale. 3) The arithmetical processes involve a significant number of CPU calls and memory accesses that could be resource consuming. Despite the relatively negligible cost for one-time DP randomization, it will be costly for real-time applications when these computations repeat and the overhead compounds.

The solution to the aforementioned problems necessitates a software-agnostic approach that is ideally more primitive and secure, less resource demanding, and highly scalable. After exploring all the design vectors, we determined that hardware memories, ubiquitous in all contemporary electronic devices, hold the key to our vision. Of course, leveraging hardware primitives for security and privacy designs is not a novel concept by any means. Yet, existing techniques are mostly not in situ and thus cannot truly achieve the vision of "privacy by design." To use the popular static random-access memory (SRAM)-based physical unclonable functions as an example, the generated randomness

0272-1732 © 2024 IEEE
Digital Object Identifier 10.1109/MM.2023.3337094
Date of publication 28 November 2023; date of current version 12 February 2024.

based on the unique responses of an SRAM chip is used as an input for other processes like encryption in the CPU. In this article, our vision is to achieve DP solely by memory, namely, privacy by memory design (PbMD). In practical implementations, PbMD will be deployed as a dedicated memory enclave that can naturally perturb data without resorting to other hardware (let alone software) components.

To realize this vision, our proposed technique in this article draws inspiration from the field of low-power memory design: a well-established area of research in the very large-scale integration (VLSI) community. However, this area has been primarily explored from a different perspective among VLSI researchers. To acquaint readers with the necessary background knowledge, we first introduce the principles and state of the art of low-power memory design. As one of the most effective techniques for low-power design, voltage downscaling can reduce the memories' power consumption because of the strong dependency of dynamic and leakage power consumption on supply voltage.[3] In the meantime, as voltage scales down, volatile memories like SRAMs are susceptible to cell failures due to significant process variation and the aging effect. There is an obvious tradeoff between power savings and reliability. In light of this, existing VLSI researches focus mainly on eliminating data errors when scaling down the supply voltage. The solutions usually include utilizing complex error-correcting codes (ECCs) and/or adopting upsized cells (e.g., larger 6-T or more than 6-T cells). Unfortunately, those designs come with significant overhead (e.g., 2× silicon-area overhead[4]), which are not sufficient to satisfy the storage needs of different electronic devices, particularly constrained or ultra-constrained end devices (see Figure 1).

AMONG EXISTING PETS, DP IS WIDELY ACCEPTED AS THE DE FACTO *STANDARD FOR PRIVACY PRESERVATION.*

Can we harness memory cell failures at subnominal voltages as a source of noise for data protection? If the answer proves affirmative, the benefits would be threefold: we can achieve PbMD, save power, and incur no additional chip overhead. In the remainder of this article, we first prepare the reader with the necessary preliminary knowledge of DP (the "DP Basics" section) and memory failure (the "Memory Failure Characteristics"

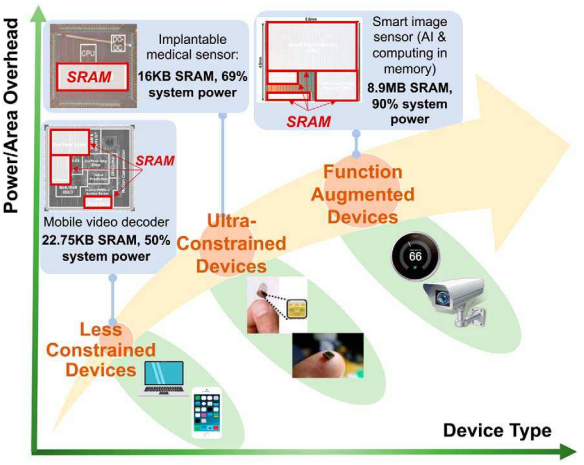

FIGURE 1. SRAM in different types of devices.

section). Then, we introduce a novel memory architecture that controls memory failures in compliance with the DP notion (the "Novel Failure-Tunable Low-Power Memory Architecture" section). Subsequently, a proof-of-concept memory chip is presented, which demonstrates the feasibility of our proposal (the "Proof-of-Concept Study: LDP by SRAM Design" section). Its engineering use cases and scientific significance are highlighted thereafter (the "PbMD Use Cases and Scientific Impact" section). We also explore the open problems associated with this emerging technology and outline a corresponding research road map that tackles them (the "Open Research Problems and Potential Solutions" section). We firmly believe that this article holds significant implications for both the cybersecurity and VLSI communities as it marks the pioneering interdisciplinary effort to achieve mutual benefits.

DP BASICS

Among existing PETs, DP is widely accepted as the *de facto* standard for privacy preservation. The development of DP stems from the need to protect an individual's sensitive data when they are collected into an aggregated database and later published as statistics to serve the public interest. Examples of such services include the U.S. census and U.S. election votes. Unfortunately, most of the techniques prior to DP are deficient because adversaries can still infer an individual's sensitive data from a statistical database by creating a series of targeted queries and remembering and correlating the results with other public databases (e.g., privacy leakage from the anonymous Netflix database in 2006).

DP was formally introduced in 2006 to mathematically define the privacy loss (byε) associated with any data release drawn from a statistical database. Intuitively, this is done by making changes to the true statistical release such that the perturbed one is not overly dependent on the data of any one individual. For this reason, the released statistics cannot be used to infer much about any individual. Formally, ε-differential privacy (ε-DP) specifies that for any two arbitrary databases D_1 and D_2 that differ in one record, a randomization technique $\mathcal{M}$ offers ε-DP if

$$\Pr[\mathcal{M}(D_1) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D_2) \in S]$$

for all output $S \subseteq \mathrm{Range}(\mathcal{M})$. $\mathcal{M}$ is commonly instantiated by the Laplace mechanism that is used to sample and add a random number from a Laplacian probability distribution with zero mean and variance dictated by ε and sensitivity—the significance of one record's influence on the statistical release.

Since its inception in 2006, DP has evolved into various kinds to cater to the needs of specific privacy-preserving scenarios. This includes the introduction of the (ε, δ)-differential privacy relaxation for less noise addition, where the local DP (LDP) notion protects local data against an untrusted data curator and many others. Specifically, we elaborate on how LDP works as it will facilitate the understanding of our proof-of-concept study in the "Proof-of-Concept Study: LDP by SRAM Design" section.

LDP can guarantee *indistinguishability* between any two arbitrary data records v_1 and v_2.[5] This is done by using the *random response* (RR) technique as $\mathcal{M}$. The general working principle of RR is *controlled bit flip*. Specifically, for a private binary value $x \in \{0,1\}$, RR follows a 2×2 matrix to perturb x, that is, $p_{sv} = p[y = s|x = v]$ $(s, v \in \{0,1\})$ as the probability of the output being s when the input is v. The RR that satisfies ε-LDP follows

$$p_{00} = p_{11} = \frac{e^{\varepsilon}}{1+e^{\varepsilon}} \text{ and } p_{01} = p_{10} = \frac{1}{1+e^{\varepsilon}}.$$

MEMORY FAILURE CHARACTERISTICS

In the past decade, low-power memory designs have been widely investigated in the literature. Reducing supply voltage enhances power efficiency, while memory failure probability significantly increases due to its growing sensitivity to process variations at a lower voltage. Specifically, SRAMs demonstrate the following three important failure characteristics at low voltages.

First, the failure probability of a cell monotonely decreases with respect to the increase of its silicon area. Our previous research revealed that in many real design applications, the failure probability function (Q) of SRAM can be fitted using its silicon area (S) as $Q = e^{aS+b}$, where a and b are constants for a specific manufacturing technology.[4] Accordingly, existing low-power memory designs usually adopt larger 6-T or more than 6-T cells to minimize or avoid the memory failures as voltages are reduced, at the expense of silicon-area overhead.

Second, any off-the-shelf memory has a property that is "fixed output upon cell failure." In design time (before a memory chip is fabricated), any data bit stored in a failed cell is considered ambiguous to determine, so the readout could be either zero or one. However, after a memory chip is fabricated, its failed bits will always be read out as the same value. For example, at low voltages, the failed bits of Cypress's commercial memory chip (CY62146GN) consistently generate 1 s.[6]

THE GENERAL WORKING PRINCIPLE OF RR IS CONTROLLED BIT FLIP.

Third, the SRAM cell failure exhibits a "fault inclusion" property. That is to say, the cells that fail at voltage v_1 will certainly fail at a lower voltage v_2, where $v_2 < v_1$. This property is commonly utilized to maintain lightweight fault maps for runtime supply voltage adaptation.[7] However, it is problematic for privacy design due to the correlation of noises.

In this article, considering all those important failure characteristics, we switch from the traditional high-overhead, "failure-avoiding" low-power memory design to a new "failure-embracing" design paradigm. To this end, a novel failure-tunable memory architecture is presented in the next section.

NOVEL FAILURE-TUNABLE LOW-POWER MEMORY ARCHITECTURE

The proposed failure-tunable low-power memory architecture is shown in Figure 2. The proposed memory is capable of adjusting the memory failure characteristics (e.g., failure probability and failure positions), which are enabled by custom memory design during the design time and by the added dynamic failure knobs that are tunable during runtime. As shown in Figure 2, in addition to the conventional memory components that support read/write operations (e.g., decoders, drivers, and precharge circuits), the proposed memory has two

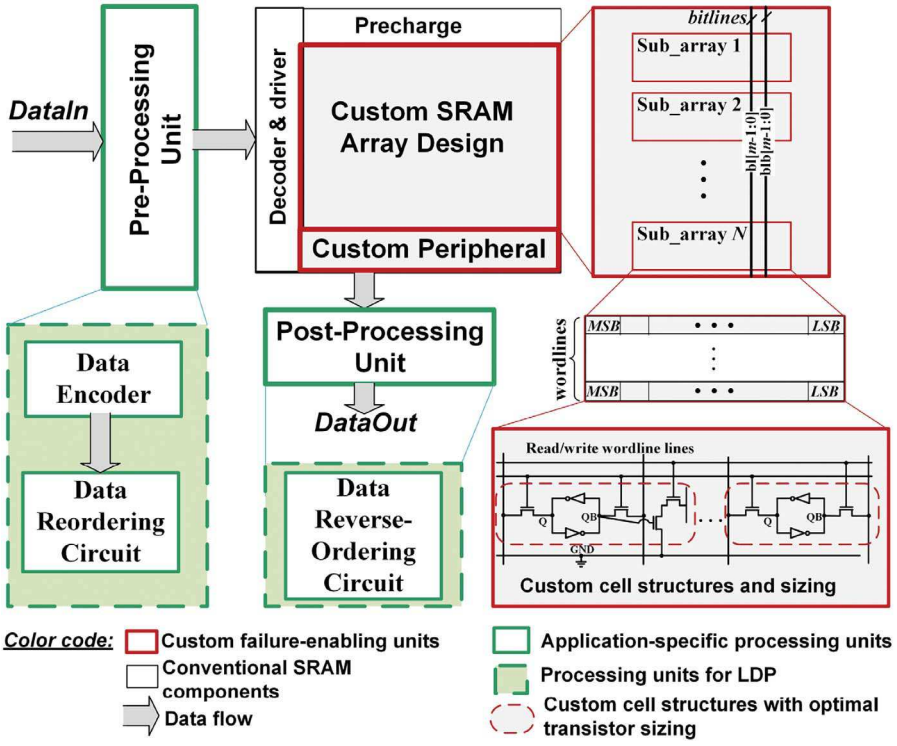

FIGURE 2. Proposed memory architecture. LSB: least significant bit; GND: ground.

main design components that support failure adaptation: 1) custom failure-enabling units, including custom SRAM array and peripheral circuits, and 2) application-specific processing units, as detailed next.

Custom Failure-Enabling Units

During the design process, the SRAM array and peripheral circuits are custom designed and optimized under different design constraints to enable the dynamic failure knobs and failure adaption. Specifically, the custom SRAM cell array design is achieved by three key design steps: 1) cell structure design, 2) device size optimization, and 3) effective cell integration. First, depending on the target failure characteristics and device application scenarios, SRAM cell structures with different number of transistors will be designed and optimized for each data bit. As discussed in the "Memory Failure Characteristics" section, larger 6-T or more than 6-T cells are able to reduce the failure probability as compared to smaller 6-T cells. However, for cells with the same structure, transistor size will further influence their failure probability and thus needs to be optimized for a specific application. For example, we studied size-dependent 6-T failure characteristics and concluded that increasing the access transistors in a 6-T cell gives rise to a lower failure probability.[8] After the structure and size of each cell are identified, developing an effective layout integration scheme is critical to reduce layout-area overhead. Typically, it needs more design efforts and a higher implementation cost to integrate cells with different structures as compared to cells with the same structure but different device size. To avoid the time-consuming and laborious cell custom design process, in our earlier work[4] we developed optimization models using nonlinear programs and integer linear programs. Different memory designs, such as alternative SRAM cells and transistor sizing techniques, are considered in our models, which can be used as a standard cell custom design and optimization modeling process for the proposed failure-tunable memory.

Also, the peripheral circuits may need a careful design process that supports the proposed memory, such as wordline or bitline voltage boosting and ECCs. In contrast to an SRAM array design, peripheral circuits offer runtime agility and adaptation of SRAM failures. Among existing designs, the most effective techniques for runtime SRAM failure adaptation are voltage scaling, bit truncation, and ECCs. Specifically, to support voltage scaling, on-chip voltage converters or additional input pin(s) are the key peripheral components.

Designing an ECC circuit that can protect different data bits is another technique that can be used to enable runtime failure adaptation. As an example, in our recent work[6] we designed an ECC circuit that achieved three SRAM failure levels caused by hamming code-74 (ECC74), hamming code-1511 (ECC1511), and no ECC.

Application-Specific Processing Units

Another key component of the proposed memory is application-specific processing units that preprocess or postprocess data (or both), which is determined by the specific application scenarios. The preprocessing unit may consist of data encoders and data reordering circuits. The purpose of a data encoder is to convert data of arbitrary types (e.g., categorical and structured) into numerical ones for memory storage. Despite encoders like one-hot encoders, which are realized by software, we propose implementing a lightweight lookup table (LUT) during design time for data encoding. In addition, the data reordering circuit is added to permutate the positions of data bits. The purpose of data reordering is to manipulate where data bits are stored in the memory so as to have fine-grained control of a data bit's failure probability. This is especially useful for data-aware applications. For instance, with the data reordering circuit, one can shift the most significant bit (MSB) to a memory cell with the least failure probability to protect the fidelity of original data. Moreover, the permutation can also be probabilistic depending on the application scenario. Our case study in the "Proof-of-Concept Study: LDP by SRAM Design" section is a good example. In practice, the data recording circuit can be implemented using multiplexers (MUXs) and a random-number generator.

When the data are read out from the memory, a postprocessing unit reverses the data back to their original modality. A similar MUX-based module can be used for data reverse-ordering circuits. Note that the processing units in Figure 2 are a general representation. Other peripheral modules, like a denoising unit, can be included for domain-specific applications and optimized performance.

PROOF-OF-CONCEPT STUDY: LDP BY SRAM DESIGN

For a proof-of-concept study, we designed a 10-Kb SRAM memory with a layout of eight memory banks, each of which has 128 words $\times$ 10 bits. A wordline with 10 SRAM bitcells is shown in Figure 3. The memory circuit was implemented using Cadence Virtuoso based

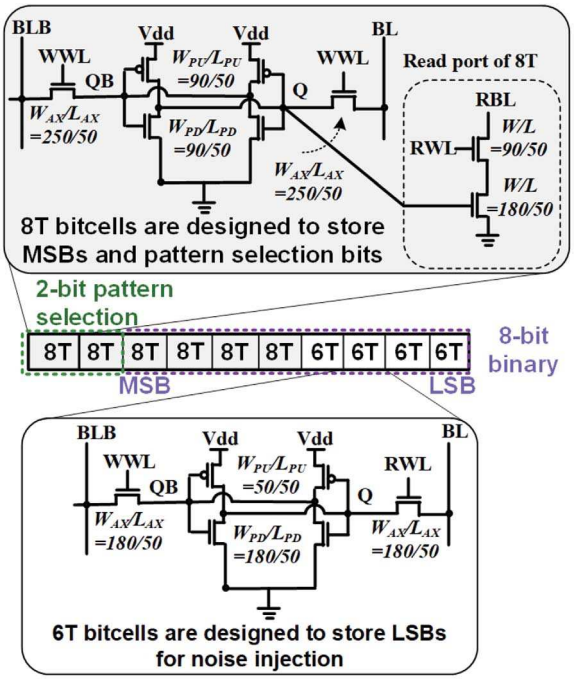

FIGURE 3. Memory cell design for LDP. LSBs: least significant bits.

on a 45-nm CMOS technology, and its nominal supply voltage is 1 V. In our analysis, 100,000 HSPICE Monte Carlo simulations were performed in the worst process corner to obtain the failure rates of cells. In addition, we assume that the host device generates numerical data of 8-bit length (i.e., of decimal values 0–255) following a Gaussian distribution with $\mu = 125$ and $\sigma = 20$. This specific experiment setup is applicable to many low-end Internet of Things (IoT) sensors with small SRAM memories. Our objective is to achieve LDP by this SRAM memory.

To render LDP noises by this customized memory, our design consists of the following four key steps:

1) *Data reordering*: Four permutation patterns, π_1-$[0,1,2,3,4,5,6,7]$, π_2-$[0,1,2,3,5,4,7,6]$, π_3-$[0,1,2,3,6,7,4,5]$, and π_4-$[0,1,2,3,7,6,5,4]$, are calculated offline and then stored as an LUT in the memory peripheral. Data reordering is achieved by a circuit that contains four 4-to-1 MUXs, for which every permutation pattern is supported by an MUX. The output of a random-number generator is used as the selection signal for an MUX.
2) *Memory storage*: As shown in Figure 3, the selected permutation pattern, 2-bit information, is stored in the memory's two leading cells,

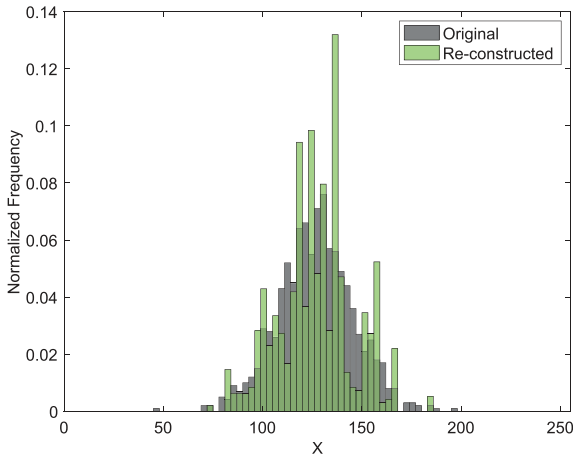

FIGURE 4. Statistics reconstruction from the memory-perturbed data.

while the 8-bit binary data are stored in the subsequent eight cells. The noise injection process is done by enabling a subnominal voltage, which leads to cell failures across the memory.

3) *Noise injection*: When the data are to be read out, each readout bit is subject to noise adding. This is accomplished by connecting MUXs to sense amplifiers of conventional SRAM.
4) *Data reverse-ordering*: The memory implements a similar MUX-based data reverse-ordering circuit. It reads the permutation pattern that was recorded in the memory's two leading cells in step 1 and reverts the eight binary bits to their original positions.

The rationale for the aforementioned design principle is as follows. The data reordering in step 1 is used to achieve fine-grained control over the failure probability of each bit position for high utility preservation as LDP is notoriously known for its high distortion to the true data value. In our design, the MSBs [resp. the least significant bit (LSB)] are shifted to the cell with the least (resp. highest) failure probability. As shown in Figure 3, the heterogeneous failure probability across MSBs and LSBs is achieved by hybrid memory cells with different cell structures and device sizes, for which a large-size cell (8 T) is less vulnerable to failures than a small-size cell (6 T) under the same subnominal voltage.

The memory storage in step 2 is used to let memory cells manipulate the bits stored therein. The noise injection in step 3 is used to address the fault inclusion property issue that the bits stored in failed memory cells will always be read out as 0 or 1 s. To retain randomness, during the readout process, we inject random noises (zero or one) into the failed cell positions. The data reverse-ordering in step 4 is used to restore the readout bits to their original positions.

After the customized memory introduces noises to the sensitive data, we analyze whether the data curator can still extract useful statistics from the noisy data. For such an evaluation, we let the customized memory write in, perturb, and then read out 1000 8-bit data records. The expectation–maximization (EM) algorithm is adopted to reconstruct the original statistics. Specifically, the EM algorithm is made aware of the memory failure probability (in our experiment, supply voltage is set to 0.5 V and the corresponding ϵ is 1.49) for the likelihood calculation, one of the key steps in the EM algorithm. When the algorithm converges, the results in Figure 4 reveal that the reconstructed histograms strongly resemble the original ones. Furthermore, the reconstructed mean and variance values are within a 3% error margin of their true values.

In addition, we draw a comparison to an existing software-based LDP mechanism to demonstrate the advantages of our memory-based LDP design. Specifically, we adopt the IBM Diffprvlib[9] toolbox, in which the *diffprivlib.mechanisms.Binary* class is called to add LDP noises to the same 8-bit binary data. The program is compiled and run in the VS Code IDE in a Macintosh OS v12.3 computer with an Apple M1 chip with 39-W standard power and a 32-GB memory of 12-W standard power. By using the *psutil* tool, we obtain 8.71% CPU usage and 0.287% memory usage for the 8-bit binary randomization process, which accounts for 7.61×10^{-5} s

TABLE 1. System overhead and comparison (bold ones are better).

System Metrics	Baseline	PbMD	IBM Diffprivlib
Chip peripheral overhead (number of transistors)	66,000	67,623 (+2.459%)	**+0%**
Latency (ns)	0.84	**1.02 (+21.4%)**	7.61×10^4 (+9.06 $\times 10^6$%)
Power consumption (mW)	3.713	**0.5724 (−88.58%)** when $\epsilon = 1.49$	3.43×10^3 (+922.78%)

of runtime and consumes roughly $39 \times 8.71\% + 12 \times 0.287\% = 3.43$-W power. Moreover, we consider a standard memory (45-nm CMOS and eight memory banks; each bank has 128 words $\times$ 10 bits) without any customization as the baseline for comparison. As shown in Table 1, despite a minor increase in chip peripheral overhead due to added circuits for voltage control and bit manipulation, our memory-based LDP significantly outperforms IBM Diffprvlib in terms of system responsiveness (measured in latency) and power savings.

PbMD USE CASES AND SCIENTIFIC IMPACT

The PbMD regime has a transformative impact on several research disciplines and many engineering systems. From a cybersecurity perspective, PbMD removes reliance on possibly uncensored software, which could be provided by malicious vendors while putting root trust on the hardware during the chip fabrication process. This methodology will greatly simplify security scrutiny throughout the device's lifecycle. On the other hand, PbMD relaxes the constraints of traditional memory designs, notably ECC and power supply modules. The chip-area overhead attributed to ECCs can now be eliminated and power consumption can also be reduced.

The practical use cases of PbMD are also immense. To use the aforementioned prototype as an example, the LDP SRAM chip can be used by resource-constrained real-time devices that are battery limited, built around a slim OS lacking the support of high-level code libraries, and that collect sensitive data. Such devices could be health monitoring sensors and surveillance cameras, as shown in Figure 1. The LDP SRAM chip, potentially implemented as a secure memory enclave segregated from normal SRAMs, can protect the devices' collected data in situ without exposing vulnerable interfaces, while prolonging the devices' battery life. Moreover, following the general architecture in the "Novel Failure-Tunable Low-Power Memory Architecture" section, other PbMD chips can be instantiated, customized, and adopted by many engineering systems, such as cloud servers.

OPEN RESEARCH PROBLEMS AND POTENTIAL SOLUTIONS

Although the developed prototype demonstrates the feasibility of achieving LDP by SRAM, the scope of PbMD and the open research problems extend far beyond. On the one hand, the notion of DP encompasses various specific models designed to address diverse privacy threats, including LDP, central DP, and hybrid DP. The associated (randomization) mechanisms for these models are also markedly different. For example, central DP can be achieved by adding noises from the Laplace, Gaussian, or binomial distributions; whereas hybrid DP is typically accomplished through data shuffling. On the other hand, SRAM is not the only memory type that exhibits cell volatility. DRAM and other emerging memory technologies may provide alternative design possibilities.

Collectively, it is worthwhile to investigate how a specific memory technology can effectively realize a particular DP model. This embodies a wide range of research subtopics in hardware, software, and their intersections. In the following sections, we outline a list of open research problems along with their potential solutions. The list is by no means exhaustive, but it aims to encourage broad investigations from different disciplines.

Hardware Perspective

Wide-Range and Fine-Grained Memory Noise Tuning

The runtime control of memory failure probability in a wide range and fine granularity is the cornerstone to controlling the scale of data randomization and thus the privacy level. Although voltage scaling is a common tuning knob, memories' sensitivity to voltage variation is nonlinear and attributable to many factors. A very tiny reduction of supply voltage at the lower subnominal voltage region could possibly lead to "sudden memory death" (i.e., memory cells failing altogether). To overcome the deficiency of voltage scaling, one viable approach is to use heterogeneous memory cells of various structures and transistor sizes. In our previous study,[3] the cell failure probability of a larger 6-T memory cell changes in a much narrower range, yet with finer granularity than that of a smaller 6-T memory cell

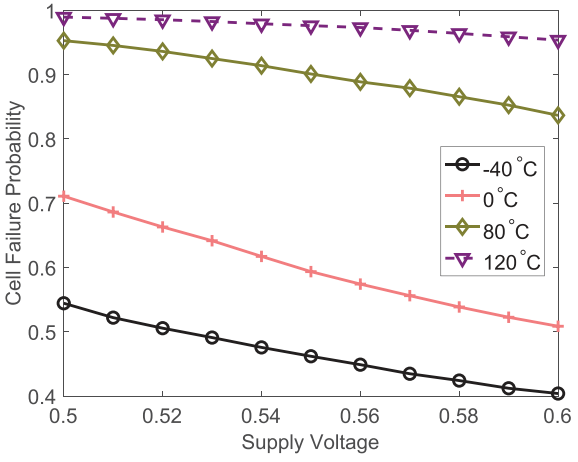

FIGURE 5. Cell failures under various temperatures.

(with a smaller transistor size). By combining the distinct merits of different memory cells, a hybrid memory can select the appropriate cells to incur noises for any specific supply voltage region.

Reliability Assurance Under Changing Operation Conditions

The behavior of the designed memory may vary under different operational conditions, which can be attributed to various factors such as hardware aging effects and ambient temperatures. For example, our previous study revealed that when subjected to the same subnominal voltages, older SRAM chips are more susceptible to cell failures than newer ones.[10] Furthermore, the simulation results depicted in Figure 5 demonstrate that the prototype discussed in the "Proof-of-Concept Study: LDP by SRAM Design" section exhibits distinct cell failure probabilities when operating at different temperatures. The impact of temperature-dependent cell failure in the context of DP is not trivial. Our calculations indicate that a $\pm 1\%$ drift in cell probability can lead to as much as ± 0.08 variations in ϵ for our prototype. Hardware reliability is thus a critical concern as it could undermine the system's operational consistency and lead to detrimental consequences. For example, a user may be misled into having a false sense of privacy protection. Moreover, an adversary can manipulate the operational conditions to their advantage.

Unfortunately, the law of physics does not permit lifetime reliability of memory hardware, but the designer should at least be informed of the hardware variances and make runtime adaptation whenever needed. A possible solution is to perform the power-on self-test (POST) on the memory hardware and extract its most-to-date memory failure characteristics. Then, the supply voltage can be adapted to achieve a target cell failure probability. Yet, an open research problem is when and how often the POST is performed, provided that the POST requires a tedious system reboot leading to undesired system downtime. For low-end IoT devices with active-sleep cycles, the POST could be carried out when a device is awakened from its sleep mode.

Technology-Dependent Memory Design

Our analysis in previous sections focuses on SRAM, which has been the workhorse for embedded memory technology for several decades. However, the continuous downscaling of CMOS technology becomes increasingly challenging. Recently, researchers have made great efforts to search for feasible alternatives, such as embedded DRAM[4] and emerging nonvolatile memory technologies (e.g., memristors).[11] In terms of dynamic RAM (DRAM), conventional DRAM design schemes, including commercial memories, are implemented based on the worst-case refresh cycle, which is determined by the leakiest cell in the DRAM array. However, refresh operations have an adverse effect on the DRAM's overall energy efficiency and performance. Energy efficiency declines due to the expensive, periodic activation of individual rows during the refresh process. The existing literature also highlights the importance of the DRAM's refresh rate. For instance, Liu et al.[12] predict that due to the ever-increasing capacity of DRAM, refresh power will become the most dominant power component. Therefore, the refresh period will be an effective technology-dependent dynamic failure knob that implements failure-tunable DRAMs. Also, the proposed failure-tunable memory can be implemented using emerging technologies. For example, as a promising memory technology candidate for ultra-constrained devices with artificial intelligence and computing in memory, the failure characteristics of memristors is largely determined by their device properties, such as nonlinearity, device-to-device variation, cycle-to-cycle variation, maximum conductance variation, and minimum conductance variation. Accordingly, memristor-based failure-tunable memristor-based memory can be implemented by adapting the voltage pulses in runtime.[11]

Software Perspective

Data Encoder/Decoder Design

The data stored in memory are in a binary format. General data types such as characters and structured data may lose semantics after being transformed into binary. It is very likely that a minor memory noise may completely destroy the usability of the original data. Therefore, a new data encoding/decoding algorithm is sorely needed to support utility-preserving PbMD. In essence, an ideal encoder should be *space friendly* and *semantics preserving*, but many popular encoders such as one-hot encoders, unary encoders, and Bloom filters are, unfortunately, not good solution candidates. Challenging requirements thus call for innovative algorithmic designs from the broader community. From our perspective, the *similarity-preserving min-hash encoder* is worthy of investigation as it sorts the structured data according to their morphological similarities, and then proceeds with dictionary encoding to convert the sorted data into compact integer/binary values.

Memory-Aware Denoising Methods

DP mechanisms achieve privacy protection at the expense of drop in data utility. In existing DP research

works, retaining high data utility through "denoising" techniques is an indispensable considering factor. Denoising does not compromise privacy level because DP provides resistance to any postprocessing algorithms after perturbation. In our PbMD regime, although we have largely discussed how to add memory noises, it is equally important to investigate how to preserve high data utility. Depending on the stage of the data's lifecycle, denoising can be performed at the source, immediately after adding memory noises; at the destination, after receiving the noisy data; or both. Our prototype in the "Proof-of-Concept Study: LDP by SRAM Design" section followed the latter scheme and applied the EM algorithm and regression analysis to recover useful statistics from the noisy data. The denoising techniques in our prototype can be further optimized by making the destination aware of runtime cell failure probabilities. On the other hand, as the source holds the true data, denoising at the source can have greater design freedom.

Data-Aware Reconfigurable Memory

This line of research is not a standalone hardware or software research, but rather a co-design regime. Specifically, the memory should be made cognizant of the nature of its stored data or the application in general. For a single datum, we know that adding noises to MSBs alters their original value more significantly than doing so to LSBs. Although among several data, their significance to the overall application is different (e.g., a border pixel is less important than a center one, or a face is more important than the background in a video). That is to say, data have a discriminatory nature and the proposed memory architecture should be made aware. This poses a design challenge about how to automatically reconfigure the memory when different data stream in.

CONCLUSION

In this article, we proposed a new design regime called *PbMD*. This idea is underpinned by a novel memory architecture that is primitive, generic, and reconfigurable and opens the door to many customized designs. We demonstrated a prototype based on this memory architecture that achieved LDP on a customized SRAM chip. Beyond this case study, we explored the research problems associated with this new design regime and shed light on their potential solutions. Looking to the future, we anticipate that this interdisciplinary research direction will invite a broad exploration of systems and memory, security and privacy, and their intersections.

ACKNOWLEDGMENT

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

REFERENCES

1. C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.*, Berlin, Germany: Springer-Verlag, 2006, pp. 265–284.
2. I. Mironov, "On significance of the least significant bits for differential privacy," in *Proc. ACM Conf. Comput. Commun. Secur.*, New York, NY, USA: ACM, 2012, pp. 650–661, doi: 10.1145/2382196.2382264.
3. Y. Xu, H. Das, and N. Gong, "Application-aware quality-energy optimization: Mathematical models enabled simultaneous quality and energy-sensitive optimal memory design," *IEEE Trans. Sustain. Comput.*, vol. 6, no. 4, pp. 559–571, Oct./Dec. 2021, doi: 10.1109/TSUSC.2020.2999882.
4. Y. Xu, H. Das, Y. Gong, and N. Gong, "On mathematical models of optimal video memory design," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 1, pp. 256–266, Jan. 2020, doi: 10.1109/TCSVT.2018.2890383.
5. J. Liu, C. Zhang, and Y. Fang, "EPIC: A differential privacy framework to defend smart homes against internet traffic analysis," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1206–1217, Apr. 2018, doi: 10.1109/JIOT.2018.2799820.
6. H. Das, A. A. Haidous, S. C. Smith, and N. Gong, "Flexible low-cost power-efficient video memory with ECC-adaptation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 10, pp. 1693–1706, Oct. 2021, doi: 10.1109/TVLSI.2021.3098533.
7. M. Gottscho, A. BanaiyanMofrad, N. Dutt, A. Nicolau, and P. Gupta, "DPCS: Dynamic power/capacity scaling for SRAM caches in the nanoscale era," *ACM Trans. Archit. Code Optim.*, vol. 12, no. 3, pp. 1–26, 2015, doi: 10.1145/2792982.
8. N. Gong, S. A. Pourbakhsh, X. Chen, X. Wang, D. Chen, and J. Wang, "SPIDER: Sizing-priority-based application-driven memory for mobile video

applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 9, pp. 2625–2634, Sep. 2017, doi: 10.1109/TVLSI.2017.2715002.

9. N. Holohan, S. Braghin, P. M. Aonghusa, and K. Levacher, "Diffprivlib: The IBM differential privacy library," 2019, *arXiv:1907.02444*.
10. N. Gong, S. Jiang, A. Challapalli, M. Panesar, and R. Sridhar, "Variation-and-aging aware low power embedded SRAM for multimedia applications," in *Proc. IEEE Int. SOC Conf.*, Piscataway, NJ, USA: IEEE Press, 2012, pp. 21–26.
11. 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 Things J.*, vol. 8, no. 12, pp. 9672–9682, Jun. 2021, doi: 10.1109/JIOT.2020.3023623.
12. J. Liu, B. Jaiyen, R. Veras, and O. Mutlu, "RAIDR: Retention-aware intelligent DRAM refresh," *ACM SIGARCH Comput. Archit. News*, vol. 40, no. 3, pp. 1–12, Jun. 2012, doi: 10.1145/2366231.2337161.

JIANQING LIU is an assistant professor in the Department of Computer Science, North Carolina State University, Raleigh, NC, 27606, USA. His research interests include wireless networks, security and privacy, and quantum information. Liu received his Ph.D. degree from the University of Florida. He is a Member of IEEE. Contact him at jliu96@ncsu.edu.

NA GONG is a professor in the Department of Electrical and Computer Engineering, University of South Alabama, Mobile, AL, 26688, USA. Her research interests include power-efficient computing circuits and systems, memory optimization, and neuromorphic computing. Gong received her Ph.D. degree in computer science and engineering from the State University of New York. She is a Member of IEEE. Contact her at nagong@southalabama.edu.