计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(9): 2355−2370, 2026

计数器溢出可感知的非易失性存储器安全重加密延迟优化技术

延航宇[1] 陈仁海[1] 张 雨[2] 史文燕[1] 吴洪越[1] 冯志勇[1]
[1](天津大学智能与计算学部 天津 300350)
[2](清华大学计算机科学与技术系 北京 100084)
(yanhangyu@tju.edu.cn)

Counter Overflow-Aware Optimization Technique for Secure Delayed Re-Encryption in Non-Volatile Memory

Yan Hangyu[1], Chen Renhai[1], Zhang Yu[2], Shi Wenyan[1], Wu Hongyue[1], and Feng Zhiyong[1]
[1] (*College of Intelligence and Computing*, *Tianjin University*, *Tianjin* 300350)
[2] (*Department of Computer Science and Technology*, *Tsinghua University*, *Beijing* 100084)

Abstract Non-volatile memory (NVM), with the advantages of persistence after power failure, high density, and large capacity, has attracted extensive attention from both academia and industry. However, its persistence also makes NVM more vulnerable to security threats such as unauthorized access and data tampering. To ensure data confidentiality and integrity during storage and access, secure NVM systems generally employ encryption and integrity verification mechanisms. Nevertheless, directly applying these mechanisms to NVM may cause severe write amplification, resulting in increased write latency and energy consumption. To address this problem, we propose epoch-based re-encryption delay (ERED), including word-epoch-based re-encryption delay (W-ERED) and line-epoch-based re-encryption delay (L-ERED). Since the re-encryption frequency of a data line is usually dominated by its most frequently updated word, W-ERED introduces flag bits for local counters to identify frequently overflowing words. When a local counter overflows, only the overflowing word is re-encrypted with a new line counter, while non-overflowing words continue to use the old line counter, thereby avoiding unnecessary re-encryption operations. Based on this observation, L-ERED further exploits the fact that the re-encryption rate of a data page is mainly determined by the data line with the highest overflow frequency in that page. By introducing overflow flags for line counters, L-ERED ensures that only overflowing data lines are re-encrypted with the major counter upon overflow, thus significantly reducing the re-encryption granularity and overhead caused by line-counter overflow. Furthermore, to reduce write overhead and improve crash recovery capability, we propose an enhanced scheme, STAR+. STAR+ introduces a counter write-filtering mechanism that writes counters back to NVM only when the number of cache updates reaches a preset threshold, thereby reducing counter write operations. During crash recovery, STAR+ adopts the Osiris technique and uses shadow counters to record incremental updates. It reconstructs counter states by reading persisted counter values, replaying the logged increments, and verifying their correctness. In addition, STAR+ restores the integrity tree by updating the authentication values of leaf and parent nodes and recursively recalculating upper-level Hash values, thereby ensuring consistency between data and metadata. Experimental results show that, compared with SECRET, W-ERED reduces the average write bit-flip rate by 5%, write latency by 11%, and write energy by 6%,

收稿日期: 2025-06-09; **修回日期**: 2026-05-12
基金项目: 国家自然科学基金项目(62572343)
This work was supported by the National Natural Science Foundation of China (62572343).
通信作者: 陈仁海(renhai.chen@tju.edu.cn)

with only 8/512 additional storage overhead. In L-ERED, the number of re-encrypted lines is only 26.7% of that in BASE and 47.7% of that in RSR, with only 1/512 additional storage overhead. Moreover, compared with STAR, STAR+ reduces NVM writes by 9.8% and performance overhead by 3.1%, while achieving comparable recovery time.

Key words non-volatile memory; data encryption; counter mode encryption; write optimization; crash consistency

摘 要 非易失性存储器（non-volatile memory, NVM）因断电后数据持久保存、高密度和大容量等优势成为了研究热点，但其持久性也使其更易遭受未授权访问与数据篡改等安全威胁。安全NVM通常采用加密与完整性验证技术，但直接加密会引发写放大，导致写入延迟增加和能耗上升。为缓解该问题，提出计数器溢出可感知的安全重加密延迟优化技术ERED（epoch-based re-encryption delay），包括基于周期的字级重加密延迟（word-epoch based re-encryption delay, W-ERED）优化技术与基于周期的行级重加密延迟（line-epoch based re-encryption delay, L-ERED）优化技术。W-ERED通过为局部计数器设置溢出标志，在溢出时仅对溢出数据字使用新的行计数器加密，未溢出数据字仍使用旧行计数器，从而避免不必要的重加密。在此基础上，L-ERED通过为行计数器设置溢出标志，在溢出发生时仅对溢出数据行使用主计数器重加密，显著减小因行计数器溢出导致的重加密粒度。进一步提出STAR+以优化写入开销并增强崩溃恢复能力：引入计数器写过滤机制，仅当缓存中计数器更新达到阈值时写入NVM；恢复阶段结合Osiris与影子计数器重建计数器状态，并通过更新MAC与递归计算哈希恢复完整性树，确保数据与元数据一致性。实验结果表明，与智能加密节能（smartly encrypted energy efficient, SECRET）技术相比，W-ERED平均位翻转率降低5%、写入延迟降低11%、写入能耗降低6%，存储开销仅增加8/512；L-ERED重加密行数仅为BASE方案的26.7%和RSR方案的47.7%，额外存储开销为1/512；STAR+较STAR写入次数减少9.8%、性能开销降低3.1%，恢复时间基本持平。

关键词 非易失性存储器；数据加密；计数器模式加密；写入优化；崩溃一致性

中图法分类号 TP333.2

DOI: 10.7544/issn1000-1239.202550462 **CSTR:** 32373.14.issn1000-1239.202550462

随着信息技术的飞速发展和数据驱动应用的普及，存储技术正面临性能与安全性的双重挑战。非易失性存储器(non-volatile memory, NVM)以其断电后可持久保存数据的特性，成为下一代存储技术的有力竞争者。相比传统的动态随机存取存储器(dynamic RAM, DRAM)，NVM具有更高的存储密度、更低的静态功耗，并具备快速读写能力，因此在数据密集型计算系统中得到了广泛关注。然而，NVM的这些优越特性也带来了新的安全与可靠性问题，例如数据的持久性增加了恶意攻击的风险，而有限的写入寿命则限制了其长期可用性。

在存储系统中，安全性和性能往往难以兼顾。NVM面临的主要安全风险包括数据机密性和完整性威胁。计数器模式加密(counter mode encryption, CME)[1-2]由于解密效率高，常用于保障数据机密性。然而，CME在加密过程中引发的写放大问题显著增加了位翻转率、写入延迟和能耗，进而影响系统的性能和耐用性[3]。针对数据完整性的威胁，基于Merkle树[4]的完整性验证技术，如Bonsai Merkle树(Bonsai Merkle trees, BMT)[5]和SGX完整性树(SGX-style integrity tree, SIT)[6-8]，在保障数据一致性方面表现出色，但其高频的元数据更新在崩溃恢复场景下造成了较高的性能开销。此外，这些技术的引入加剧了存储资源消耗，进一步增加了能效管理的难度。

为了平衡NVM的安全性与性能，近年来，研究者提出了一系列优化方案[9]。例如，双计数器加密(dual counter encryption, DEUCE)[10]通过减少数据行级别的重加密操作，有效降低了写入开销；智能加密节能(smartly encrypted energy efficient, SECRET)技术[11]则结合零字检测和能量掩码技术，进一步优化了加密效率和能耗。尽管如此，现有技术仍难以有效缓解计数器溢出带来的性能瓶颈，尤其是在高强度读写负载下频繁触发重加密操作时，其性能劣化问题更为明显。此外，在安全NVM系统中，如何保证元数据的崩溃一致性[12]同样面临挑战，特别是在采用CME加密与完整性验证机制的情况下，该问题更加复杂。尽管Osiris[13]提出了一种适用于BMT和与CME组合架构的有限计数器回写策略，但此策略未能有效扩展至SIT与CME的组合体系。此外，Anubis[14]，Phoenix[15]，STAR[16]对SIT崩溃恢复方面取得了优化，但未能完

全解决 CME 和 SIT 组合系统中额外元数据写入的问题。因此，如何在提升加密方案有效性的同时，妥善解决元数据一致性问题，仍然是一个重要的研究课题。

为解决上述问题，本文提出了以下方案。

1）计数器溢出可感知的安全重加密延迟优化技术 ERED（epoch-based re-encryption delay）。该方案分为 2 个子方案：基于周期的字级重加密延迟（word-epoch based re-encryption delay, W-ERED）优化技术和基于周期的行级重加密延迟（line-epoch based re-encryption delay, L-ERED）优化技术。W-ERED 通过为数据字级的局部计数器设置溢出标志，在某一数据字的局部计数器发生溢出时，系统仅对该数据字进行重加密，而非对整行数据字进行重加密；在 W-ERED 基础上，我们进一步提出了 L-ERED, L-ERED 对行计数器设置溢出标志，当某行的行计数器发生溢出时，系统仅对该行进行重加密，而不会对整页的数据行进行加密。通过这种精细化管理，ERED 有效避免了不必要的全行或全页重加密，从而显著降低写放大效应与冗余写入开销，并大幅提升了加密效率及系统整体性能。

2）基于 Osiris 的 SIT 跟踪与恢复方案（STAR+）。该方案利用 SGX 完整性树的懒更新特性，通过将父节点的修改信息嵌入子节点 MAC 字段中的未使用位，实现了父子节点的一次性原子更新，显著减少了对完整性树的多次写入。同时，STAR+ 引入了计数器写过滤机制，仅在缓存中的计数器累计更新达到设定阈值时才将其写回 NVM。针对此延迟写策略可能引发的计数器值不一致问题，STAR+采用 Osiris 技术，通过分析持久化元数据和冗余信息恢复最新计数器值。在崩溃恢复阶段，STAR+利用影子计数器记录增量更新，结合持久化计数器值和日志增量，高效重建计数器状态，并通过验证确保正确性。此外，STAR+通过递归更新叶子节点和父节点的 MAC 值来重构完整性树，确保数据和元数据的机密性与一致性。相较于现有方案，STAR+在性能优化和恢复鲁棒性方面展现出显著优势，为 NVM 系统的高效安全存储提供了新的解决方案。

实验结果表明，ERED 和 STAR+方案具有显著的性能与安全性。相比现有优化方案 SECRET，本文提出的 W-ERED 方案在位翻转率上降低约 5%，写操作延迟下降约 11%，写入能耗降低约 6%，在此基础上，仅需 8/512 的额外存储。在 L-ERED 机制中，重加密的行数仅为传统方案的 26.7%，同时仅增加了 1/512 的存储开销。此外，STAR+方案与 STAR 相比，NVM 写入次数减少 9.8%，性能开销降低 3.1%，但恢复时间几乎持平。以上成果表明，本文提出的方案在提升 NVM 的安全性和耐用性方面具有显著效果，还为未来高效 NVM 系统的设计提供了有力支持。

1 相关工作

NVM 凭借其高速读写、数据持久以及低功耗等显著优势，正逐步成为传统存储体系的有力补充。然而，NVM 断电后数据依然存留的特性，使其易于沦为攻击者的目标。为保障 NVM 的安全性，研究者提出了一系列安全加密措施，但对 NVM 进行直接加密常导致严重的写放大问题，显著影响其性能与寿命。为应对这一挑战，研究人员已从数据加密、完整性保障等多个维度探索并提出了一系列安全机制及优化方案。为了全面深入地理解 NVM 系统中的安全设计要点，本文将系统地梳理并分析相关研究成果。

1.1 攻击模型

目前的安全计算系统通常将可信边界限定在处理器内部，而芯片外部组件（如内存总线）则可能存在安全漏洞，易被攻击者利用。在此背景下，针对 NVM 的攻击主要可归纳为机密性破坏与完整性破坏 2 类。机密性破坏旨在在不被察觉的情况下窃取存储内容，常见方式包括监听处理器与 NVM 之间的总线传输以获取明文数据，或通过物理方式访问 NVM 模块并直接提取其中信息。由于 NVM 具备非易失性，攻击者即使在系统断电后也能提取存储数据，这无疑显著增加了物理访问攻击的风险，并进一步加剧了机密性破坏攻击的危害。完整性破坏是指对内存数据进行未授权修改的行为，例如通过伪造、重放或修改操作破坏存储内容的正确性，进而影响系统运行的可信性。为抵御上述威胁，NVM 系统通常采用数据加密技术以保护存储内容的机密性，并辅以完整性验证机制以维护数据的一致性与可信性。

本研究主要集中于机密性破坏（如总线窥探攻击和物理访问攻击）与完整性破坏（如篡改、拼接和重放攻击）2 类攻击，其他类型的攻击不在本文的研究范围内。

1.2 加密技术

为了应对 NVM 在加密过程中面临的性能和能效挑战，研究者提出了多种优化方案，以提升加密效率、降低加密开销，并保障数据安全性。其中，CME、DEUCE 以及 SECRET 是 3 种具有代表性的方案，尤其在高效、低能耗存储的需求下，展现出了良好的性

能和可扩展性。

CME 是一种将分组密码转化为流密码的加密模式, 其基本原理如图 1 所示。该模式广泛应用于保护 NVM 中的数据安全。CME 通过为每个数据分组生成唯一的计数器, 并加密该计数器以生成一次性密码本(one-time pad, OTP), 然后将 OTP 与明文逐位异或以生成密文。相比直接对数据使用高级加密标准(Advanced Encryption Standard, AES)加密, CME 仅对计数器进行加密, 从而提高了加密效率, 同时利用并行处理隐藏解密延迟。

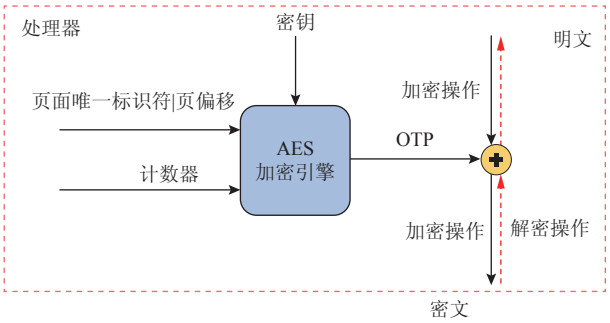


Fig. 1 Counter mode encryption

图 1 计数器模式加密

在组合计数器加密模式中, 计数器由 64b 主计数器和多个 7b 子计数器组成。主计数器标识物理页面, 与密钥和子计数器一起生成 OTP, 确保加密数据的唯一性。子计数器管理数据块的版本号, 避免密钥流重复。当子计数器溢出时, 需重加密整个页面以防重放攻击, 然而这样会导致写放大效应, 显著增加系统的延迟和能耗, 成为影响其性能的关键因素之一。

为解决这一问题, Young 等人[10]提出了 DEUCE 方法。DEUCE 通过引入双计数器架构追踪存储行内被修改的部分, 仅重加密实际更改的数据, 而未修改的数据保持原有加密状态, 从而避免整行数据的无效加密。通过这种双计数器设计, DEUCE 有效减少了不必要的加密操作, 降低了系统开销, 同时保持了较高的安全性。此外, 为防止长期加密状态不变带来的安全隐患, DEUCE 定期执行整行数据的重加密。总体而言, DEUCE 在安全性与系统性能之间实现了有效平衡, 为存储加密提供了一种可扩展的解决方案。

为进一步降低 NVM 加密过程中的能耗, Swami 等人[11]提出了 SECRET 方法。SECRET 采用字级粒度的加密策略, 每个缓存行内的字均独立分配计数器, 仅对发生变化的字重新生成密钥流并加密, 从而避免整行冗余操作。此外, SECRET 利用实际工作负载中零字的高出现率, 通过设置零字标志位跳过未变零字的加密计算, 减少了位翻转率和计算开销。为进一步优化写入能耗, SECRET 引入基于 XOR 的能量掩码技术, 通过对密文执行 XOR 操作, 将原本高能耗的位状态转换为低能耗的位状态, 大幅降低了将密文写入 NVM 时的能量消耗。

近年来, 随着 NVM 应用场景的不断扩展, 研究者致力于在提升系统安全性的同时, 降低加密机制带来的写放大与实现开销。在体系结构与持久化层面, 针对日益复杂的 NVM 安全威胁, 近期工作提出了恢复感知的可靠性与安全性协同设计方案[17]; 同时, 也有研究从系统级角度探讨安全机制与持久化模型(如 ADR/eADR)之间的协同设计[18], 以减少冗余写入和恢复开销。在算法与隐私保护层面, 为了防御侧信道攻击并隐藏访问模式, 一些最新研究引入了概率性缓存或异构内存调度机制, 对 ORAM 等隐私保护方案的访问延迟进行了优化[19-20]; 此外, 针对 AI 驱动的嵌入式终端, 也有研究揭示了 NVM 存内计算面临的新型侧信道漏洞, 并提出了相应的硬件防御策略[21]。现有研究从不同层级对安全 NVM 系统进行了探索, 但其关注重点多集中于系统级协同、AI 安全或高开销的隐私保护算法。为此, 本文从微架构层面切入, 着重解决计数器模式加密中由溢出引发的冗余重加密问题。我们在不引入复杂系统支持的前提下, 提出了一种能有效抑制写放大、并兼顾性能与能效的优化方案。

1.3 完整性验证技术

为了确保 NVM 系统中的数据安全性, 加密和完整性验证通常被结合使用。加密技术(如计数器加密)用于保护数据的机密性, 防止未经授权的访问。然而, 加密本身并不能完全防止数据的篡改, 即使数据被加密, 恶意用户仍然可能通过篡改加密数据或干扰存储过程威胁数据的完整性。因此, 完整性验证成为确保数据未被篡改的必要补充。

Merkle 树是一种经典的完整性保护技术, 广泛用于防御数据篡改和重放攻击。其结构以二叉哈希树的形式组织, 叶子节点存储数据块的哈希值, 非叶子节点存储其子节点的哈希值。树的根节点的哈希值代表整棵树的数据完整性, 任何数据篡改都会引起对应哈希值的变化, 并进一步影响根节点的哈希值。系统在验证数据完整性时, 仅需检查从数据块到根节点的路径, 而无需验证整个数据集。这种方式不仅提高了完整性验证效率, 还能快速检测数据篡改。然而, 在实际应用中, Merkle 树的元数据更新较为频繁, 尤其在崩溃恢复场景下, 可能带来额外的

性能开销。为此，Rogers 等人[5]提出了 BMT 结构，以减少 Merkle 树的存储与计算开销。

BMT 采用了一种针对计数器的精简 Merkle 树策略，其结构如图 2 所示。与传统完整 Merkle 树不同，它仅为存储系统中的计数器部分构建验证结构。由于数据块已通过计数器模式加密，BMT 将 Merkle 树的功能聚焦于防止计数器篡改，利用计数器占用空间远小于数据块的特点，大幅降低存储开销和缓存占用。为增强安全性，BMT 为每个数据块关联一个由数据内容和计数器值计算的消息认证码 MAC，当计数器递增时，旧数据及其 MAC 若被重放，会因不匹配而被检测并阻止，从而有效防范篡改和重放攻击。此外，BMT 通过优化 L2 缓存利用率减少缓存污染，提高了缓存中数据块的比例，进而提升系统性能。然而，在某些高并发或安全性要求更高的场景中，BMT 的单一保护机制和串行更新方式可能面临局限。

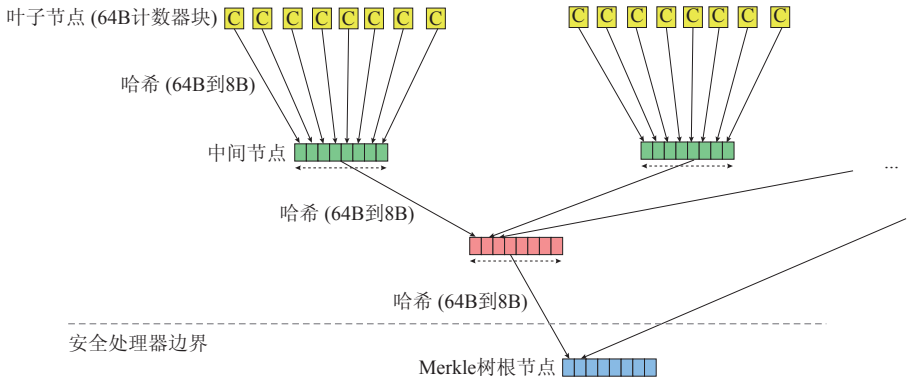


Fig. 2 Bonsai Merkle trees

图 2 Bonsai Merkle 树

相比之下，SIT 采用统一且规整的树结构，为计数器模式加密提供高效的完整性保护。在 SIT 设计中，每个节点固定大小(通常为 64B)，包含多个计数器和一个用于数据完整性验证的消息认证码。为增强安全性，SIT 引入双重 MAC 机制，其结构如图 3 所示，通过额外的校验保护计数器免受篡改，并支持并行更新以提升高并发环境下的数据修改效率。根据应用需求，节点内计数器数量可灵活调整，使 SIT 适应不同安全级别和数据流量。然而，SIT 的固定结构和高并发更新可能导致较高的存储和计算开销，这一问题在内存操作密集型的应用场景中尤为突出。在 SGX 计算中，每次页面置换涉及操作系统调用和树更新，可能消耗高达 40 000 个 CPU 周期。为解决 SIT 在 SGX 环境中因页面置换带来的性能瓶颈，Taassori 等人[22]提出了改进算法 VAULT，通过引入可变节点度与扁平化树结构降低遍历深度，并结合加密叶子节点、共享 MAC 及数据压缩技术以减少存储与重置开销。尽管 VAULT 在性能与空间效率方面取得了显著改进，其结构复杂性也相应增加。

在此基础上，近年来针对安全 NVM 场景下完整性树的更新效率与崩溃一致性问题，研究者从不同系统层级提出了多种改进方案。例如：部分工作面向资源受限的工业级嵌入式 NVM，提出了低开销的轻量级完整性保护机制[23]；针对系统崩溃后的数据一致性挑战，也有研究提出了面向完整性树根节点的原子性更新策略[24]；此外，近期工作还从系统恢复效率角度提出了基于 Steins 算法的快速恢复方案，以缩短系统重启时间窗口[25]。上述研究表明，在安全 NVM 系统中，如何在一致性保障、恢复效率与系统实现复杂度之间取得平衡，仍是该方向面临的重要挑战。然而，现有方法多聚焦于恢复机制或根节点更新优化，对高频写入场景下父子节点更新特性的开销影响讨论仍不充分。基于此，本文进一步关注 SGX 完整性树在高频写入条件下的父子节点更新行为，并提出相应的优化机制。

2 动　　机

在 NVM 中，高效管理写操作是提升系统性能和延长设备寿命的关键挑战。相比读操作，写操作不仅具有更高的延迟和能耗，还会加速设备磨损，显著缩短使用寿命。因此，减少不必要的写操作对于优化 NVM 性能至关重要。研究表明，存储系统中的写操作通常仅涉及少量位修改，而非整个数据块的更新。为应对这一问题，已有优化方案如数据比较写(data-comparison write, DCW)[26]和翻转写入(flip-n-write, FNW)[27]，通过减少写入数据量，有效降低了写操作的性能开销。

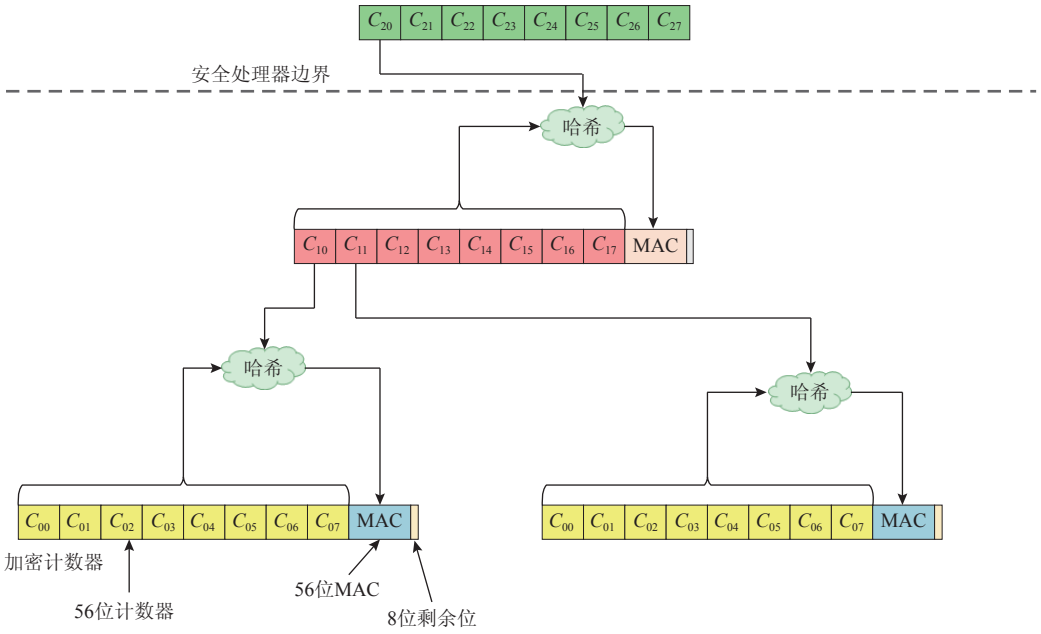


Fig. 3 SGX integrity tree

图 3 SGX 完整性树

引入加密技术后, 写操作的复杂性显著增加。加密过程不仅带来了额外的数据处理开销, 还会提高位翻转率, 进一步加剧了写操作的能耗和延迟问题。SECRET 等字级加密技术尝试通过仅加密发生修改的数据字来降低位翻转率, 但对于写操作中频繁针对特定字修改的程序, 其效果有限。在 SECRET 及组合计数器加密机制下, 局部计数器溢出会引发行级重加密; 当行计数器递增至 127 后, 还将进一步触发主计数器更新并导致整页数据重加密, 从而造成系统短暂阻塞。为量化该现象对实际运行的影响, 本文选取计数器增长速度较快的 5 个典型应用(gobmk, hammer, leslie3d, mcf, milc)进行长期跟踪统计。通过对计数器递增趋势进行建模分析, 计算相邻 2 次行计数器溢出的时间间隔。结果如表 1 所示, 在 8b 行计数器配置下, 这些应用的溢出时间间隔平均约为 0.2 s。

Table 1 Estimated Time Interval for Counter Overflow

表 1 预估的计数器溢出时间间隔

应用程序	8b 计数器溢出时间间隔/s	16b 计数器溢出时间间隔/min	32b 计数器溢出时间间隔/天	64b 计数器溢出时间间隔/千年
mcf	0.2	1	48	592 417
leslie3d	0.2	1	50	602 316
gobmk	0.2	0.8	36	453 279
milc	870	1.5	60	621 256
hammer	0.2	1	43	590 101
平均值	0.24	1.06	47.4	571 873.8

针对加密系统中频繁出现的计数器溢出问题, 本文提出了基于 SECRET 的计数器溢出可感知的安全重加密延迟优化技术——ERED。通过在数据字和数据行中设置溢出标志, 系统能够精确追踪并识别频繁溢出的计数器, 从而减少不必要的重加密操作和位翻转率。这一方案通过局部处理溢出的数据字或数据行, 有效减少了大范围数据块的重加密, 优化了写操作效率。其中, W-ERED 方案通过对单个数据字的溢出进行加密, 而 L-ERED 则在此基础上进一步优化, 确保只有溢出行被重加密。该方案在保证数据安全性和完整性的同时, 显著提升了系统性能和能效, 降低了性能开销, 适应了高效加密需求。

此外, 尽管 Osiris 系统能够提供崩溃一致性保障, 但其设计难以适用于结合 SIT 和 CME 加密的系统。为此, 本文提出了基于 Osiris 的 SIT 跟踪与恢复(STAR+)方案, 通过减少元数据写入次数和优化崩溃恢复过程, 进一步提升了 NVM 写操作的效率, 并为安全存储系统提供可靠的崩溃一致性保障。综上, 本文通过 ERED 与 STAR+方案, 有效减少了 NVM 系统的写操作开销, 同时提升了安全性与耐用性, 为高效、安全的 NVM 设计提供了新的优化策略。

3 系统设计与实现

3.1 ERED

为了减少位翻转率和重加密频率, 本文提出了

计数器溢出可感知的安全重加密延迟优化技术——ERED，它包含 2 种子技术，即 W-ERED 和 L-ERED。该方案集成了局部和行级计数器溢出检测单元、零字检测单元以及能量掩码单元，以优化加密过程中数据的处理方式。

在数据写入过程中，首先检测数据从最后一级缓存(LLC)驱逐后对应计数器是否发生了溢出。若发生溢出，AES 加密引擎为每个数据字生成 OTP。零字检测单元识别明文中的零字，并跳过这些数据，避免不必要的写入。对于剩余的非零字，系统将其与对应的 OTP 进行异或运算，生成新的密文。最终，在数据写入 NVM 之前，能量掩码单元将数据从高能态转换为低能态，以降低写入能耗。系统的整体架构如图 4 所示。

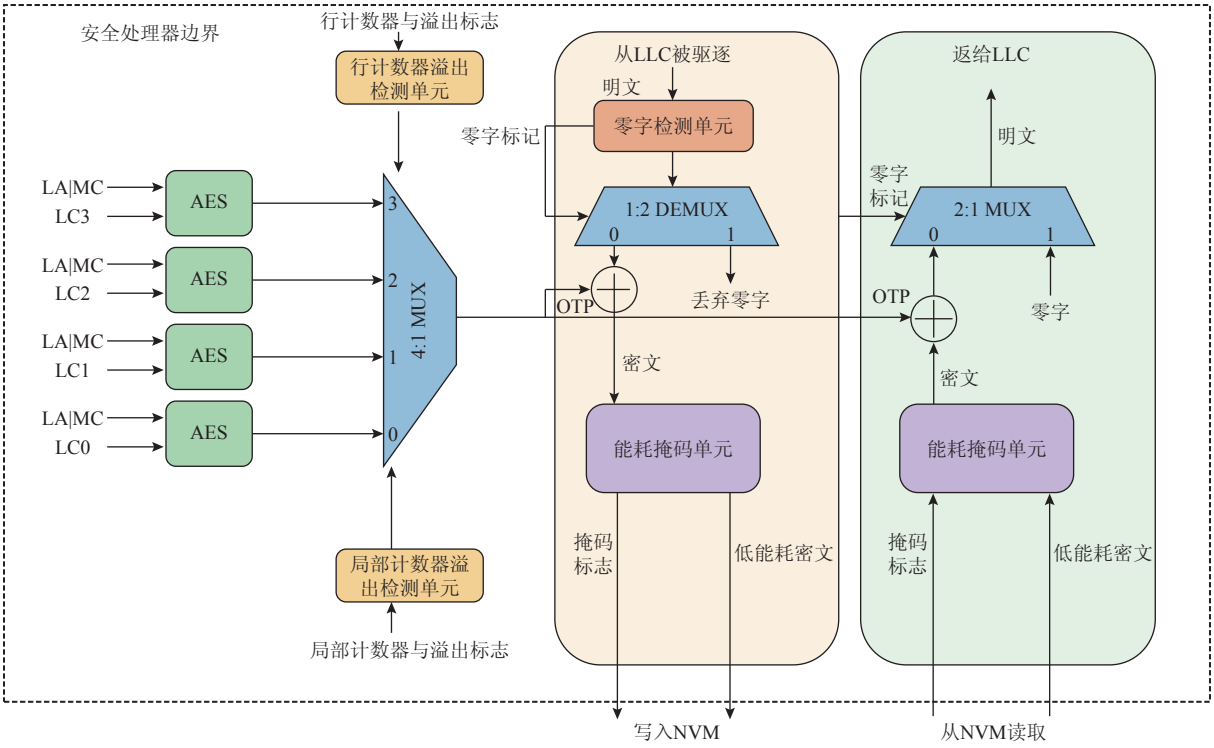


Fig. 4 Overall architecture of ERED

图 4 ERED 整体架构图

图 4 中“从 NVM 读取”所示过程表示数据读出过程，与写入过程相逆。首先，系统解密元数据，并执行局部和行级计数器溢出检测。随后，通过 XOR 运算恢复明文，并将标记为 0 的数据直接拼接至明文，避免冗余解密操作。这一设计不仅提升了加密处理效率，同时显著降低了系统能耗，优化了 NVM 的整体性能。

3.1.1 W-ERED

研究表明[28]，数据行的重加密频率通常由更新最频繁的字决定。然而，许多字的更新频率较低，甚至保持不变。尽管这些字很少发生修改，它们仍可能会因局部计数器的频繁溢出而被重加密，从而增加不必要的计算开销。为降低计数器溢出带来的重加密开销，本文提出 W-ERED，该方案对计数器状态进行监测，当检测到某些数据字对应的局部计数器发生溢出时，仅为这些数据字分配新的行计数器并完成相应的重加密操作；而未发生溢出的数据字则继续沿用原有行计数器进行加密处理。与此同时，旧计数器信息可通过对新计数器实施掩码运算进行恢复，从而在保证功能正确性的同时降低额外的存储与计算开销。

如图 5 所示，W-ERED 方案在保留 SECRET 方案字级重加密能力的同时，对局部计数器溢出情况进行感知与区分，有效降低写操作过程中产生的位翻转率。为了指示局部计数器是否溢出，每个字分配了一个 1b 的溢出标志。需要注意的是，这些计数器属于逻辑计数器，无需额外的物理存储空间。

在写操作过程中，W-ERED 方案根据局部计数器的溢出状态对密文进行选择性更新，从而降低位翻转率。以 1 次写入过程为例：首次写入时，仅第 1 个数据字发生非零变化，第 2 个数据字未被修改，而第 3 个数据字被更新为 0 值。相应地，发生变化的 2 个

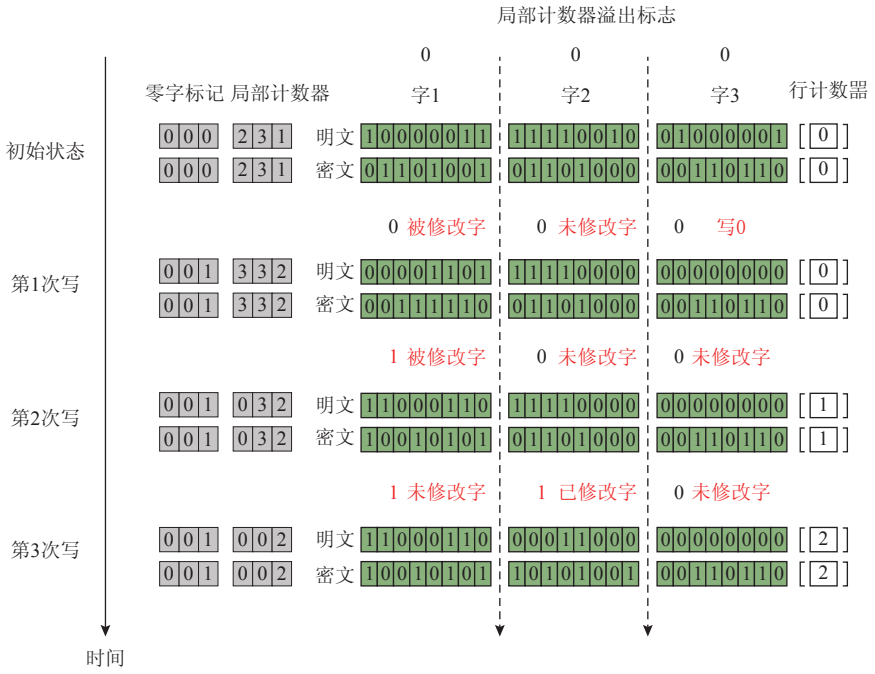


Fig. 5 Working principle of W-ERED

图 5 W-ERED 工作原理

数据字的局部计数器分别递增 1，其中第 3 个数据字的零字标志位被置为 1。在第 2 次写入时，第 1 个字再次被修改，导致该字的局部计数器达到最大值并发生溢出。在这种情况下，行计数器加 1，该字的溢出标志设置为 1，并将第 1 个字的局部计数器更新为 0。传统的 SECRET 方案需要使用新的计数器重加密该字所在的整个行；而 W-ERED 方案通过为局部计数器溢出的字设置溢出标志，仅使用新的计数器对该字进行加密。对于第 2 个字和第 3 个字，由于计数器没有溢出，它们仍然使用旧的行计数器（通过掩盖行计数器的最低 2 位来得到）进行加密，因此在 NVM 中保持密文不变。最后，在第 3 次操作中，第 2 个字被修改，导致其局部计数器溢出，此时行计数器加 1，该字的溢出标志设置为 1，并使用新的行计数器对第 1 个字和第 2 个字进行加密。通过这种方式，W-ERED 方案显著减少了位翻转率，提高了写操作的效率。

3.1.2 L-ERED

为了解决行计数器频繁溢出导致的性能开销，本文提出 L-ERED。该方案聚焦于页面中溢出最频繁的行，并仅选择性地重加密这些行，从而减少因行计数器溢出所需的重加密数据量，有效降低系统停顿时间，减少计算开销，并提升存储性能。

如图 6 所示，L-ERED 方案为每个行计数器引入溢出标志，用于指示该计数器在当前周期是否溢出，初始时数据块 D0，D1 对应的行计数器为 126 和 127。在第 1 次写入 D0 时，行计数器 C0 递增至 127，未发生溢出。随后，在第 2 次写入操作中，C0 递增至 128，发生溢出。在传统方案中，该溢出会触发页 0 的主计数器递增，并重加密页 0 内的所有数据行，同时重置所有行计数器为 0。相比之下，L-ERED 方案采用更高效的策略：仅设置 C0 的溢出标志设置为 1，重置 C0 为 0，并仅使用更新后的主计数器重加密 D0，避免对未溢出的数据行执行冗余加密。在第 3 次写入 D1 时，行计数器 C1 发生溢出，L-ERED 方案将 C1 的溢出标志设置为 1，主计数器增加至 2，并将 C1 和 C0 都重置为 0。通过选择性地重加密受溢出影响的具体数据行，L-ERED 显著减少了重加密整个页面的需求，从而优化存储系统性能，降低因计数器溢出导致的计算和存储开销。

3.1.3 安全性分析

ERED 方案的底层采用了 DCW 技术，其安全性基础已在 DEUCE 与 SECRET 等工作中得到了验证。计数器模式加密的安全性在于 OTP 不被重用。在 ERED 中，每次对数据字进行写入修改时，系统均强制使用递增后的计数器（局部计数器或行计数器）生成全新的 OTP，并对该数据字进行重加密。因此，在确保 OTP 唯一性的前提下，ERED 的加密策略并未引入新的机密性风险，其安全强度与标准计数器模式加密保持一致。此外，针对 ERED 引入的额外元数据（即溢出标志），其安全性也得到了保障。SECRET



Fig. 6 Working principle of L-ERED

图 6 L-ERED 工作原理

方案中的零字标志位与能量掩码标志均随密文存储于 NVM, 且已被证明不会泄露明文信息。同理, ERED 新增的溢出标志仅用于指示局部计数器是否发生溢出, 即使攻击者能够读取存储在 NVM 中的明文溢出标志, 其仅能获知哪些数据字或数据行曾发生过计数器溢出, 而无法利用该信息恢复原始明文数据或诱导 OTP 重用。在计数器模式加密系统中, 加密安全性依赖于 OTP 的不重用, 而非计数器或辅助状态信息的保密性。因此, ERED 引入的溢出标志不会破坏系统的机密性。

3.2 STAR+

3.2.1 STAR+ 机制设计

在安全 NVM 系统中, 崩溃一致性保障至关重要, 为此, 本文提出了 STAR+方案, 该方案旨在提升系统恢复能力的同时, 最大程度降低性能损耗并优化写入效率。图 7 展示了 STAR+方案的核心组成部分, 该方案通过一系列优化设计, 确保在提供高安全性与数据一致性的同时, 不会显著影响系统的整体性能。

STAR+方案包含计数器 MAC 融合机制、缓存树、位图行以及计数器写过滤机制 4 个核心组件。这些组件协同工作, 确保系统在崩溃或断电后能够快速恢复数据, 提升系统的写操作效率, 并显著减轻传统崩溃一致性保障机制带来的性能负担。

首先, STAR+ 采用懒更新策略优化 SIT 的更新过程。通过计数器与 MAC 的融合设计, 父节点的更新被延迟并合并到子节点中完成, 使 1 次写操作即

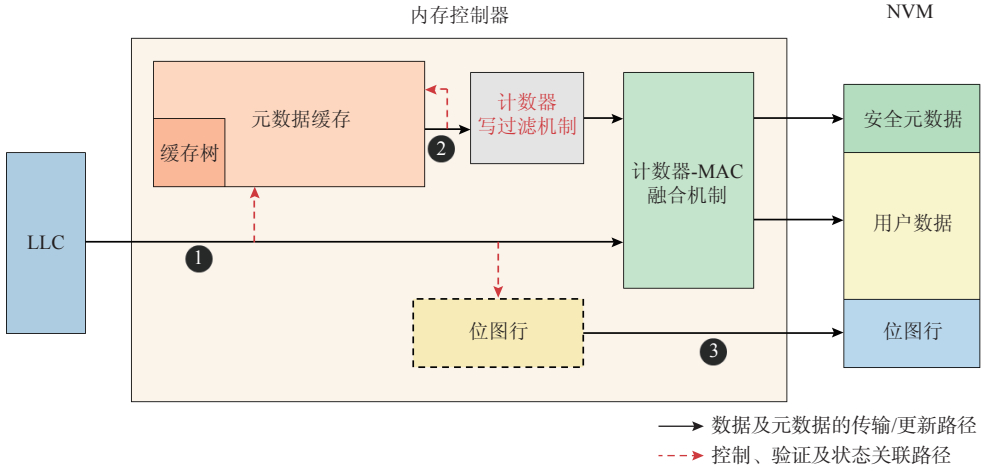


Fig. 7 Overall structure of STAR+

图 7 STAR+ 整体结构

可同步更新父子节点并完成持久化，减少了额外的元数据写入开销，从而提升整体更新效率。

为保证上述父子节点合并更新机制在异常情况下的一致性，STAR+ 在设计中结合了缓存行粒度写回特性与崩溃恢复机制进行保障。具体而言，父节点的修改信息被嵌入存储在子节点 MAC 字段的未使用位中，并随子节点作为缓存行粒度的原子写回单元一并写回 NVM，从而避免持久化过程中父子节点更新不完整而导致的状态不一致问题。在系统异常或崩溃场景下，STAR+ 结合位图行记录的安全元数据地址以及 Osiris 机制对计数器状态进行恢复，通过重建计数器并重新计算对应的 MAC 值，恢复完整性树的一致状态。此外，完整性验证机制能够进一步识别异常修改并阻止错误数据被系统接受，从而保证崩溃场景下的数据一致性。

同时，STAR+ 通过位图行记录历史安全元数据地址，加速崩溃恢复过程中关键元数据的定位，从而进一步缩短系统恢复时间。

相较于 STAR 方案，STAR+在叶子节点写回策略上进行了调整。传统设计在叶子节点发生自然驱逐时，会将对应数据写回至 NVM，而 STAR+则引入基于计数器的条件控制机制，对写回行为进行门控管理。具体而言，只有当节点计数值满足 $counter\%N == 0$ 时，相关数据才会被持久化至 NVM，从而减少非必要的写回更新。由于叶子节点通常是元数据缓存更新的主要目标，该优化进一步降低了存储系统的写入开销，提高了整体性能和耐用性。

通过这些优化，STAR+不仅有效解决了 NVM 系统中的崩溃一致性问题，还显著提高了写入效率，降低了能耗和性能损耗，为安全 NVM 系统提供了一种高效且可靠的解决方案。

图 8 展示了 STAR+方案中从 LLC 到 NVM 的数据写入流程。当脏数据块被驱逐时，内存控制器首先计算该数据块的错误校正码(ECC)并请求相应的加密计数器。如果计数器已存在于缓存中，则直接更新；否则，从 NVM 读取该计数器，并采用 Osiris 技术进行恢复。缓存树负责验证数据的正确性，确保加密与恢复过程中无误，而位图行则用于跟踪脏元数据，优化恢复时间。叶子计数器与中间节点的驱逐决定了数据是否写回 NVM。父节点采用懒更新策略，其更新后的 MAC 值直接存储在父节点中。随后，AES 加密引擎使用计数器、地址和密钥生成 OTP，并通过 XOR 运算将 OTP 与数据结合，生成密文后写回 NVM。

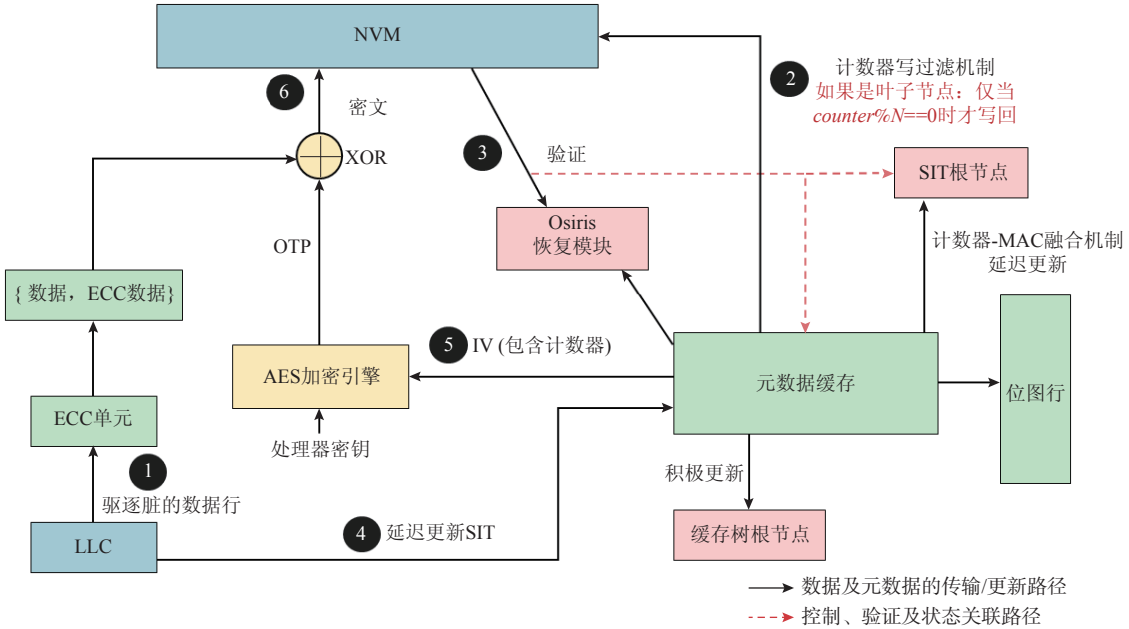


Fig. 8 Schematic diagram of STAR+ write operation

图 8 STAR+ 写操作示意图

在系统崩溃时，芯片上的 16 个位图行通过 ADR 技术刷新回 NVM。STAR+通过扫描位图行识别非零行，并根据位图位定位 NVM 中旧的安全元数据的物理位置。旧元数据包括加密计数器和 SIT 中间节点。加密计数器通过 Osiris 技术恢复，而 SIT 中间节点的新值则通过推导获得：新值的高位来自 NVM 中的旧值，低 10 b 字段来自子节点的新值。恢复缺失的 SIT 节点计数器后，系统会重新计算每个节点的

MAC 值。通过重建缓存树并与崩溃前保存的根节点进行比较，确保数据恢复的正确性。如果恢复结果不一致，则可能表明系统遭遇攻击。

3.2.2 安全性影响分析

STAR+方案结合了懒更新与日志恢复机制，无论是在系统恢复阶段还是正常运行阶段，均具备对篡改或重放攻击的检测能力。首先，针对与恢复相关的元数据（如旧节点计数器的高位信息、位图行等），若攻击者对其进行篡改或重放，将破坏恢复阶段重建的元数据缓存完整性，从而产生错误的 MAC 值。由此，恢复后的缓存树根哈希将与芯片内非易失性寄存器中存储的参考根哈希不匹配。一旦检测到这种不匹配，系统将判定恢复失败并拒绝接受当前的不安全状态，从而阻断攻击。其次，对于不参与恢复过程的元数据，尽管其在恢复阶段可能无法被立即验证，但在系统恢复完成后的正常运行过程中，通过访问 SIT 根节点或缓存中的完整性验证路径，系统仍能检测到任何潜在的篡改或重放行为。值得注意的是，STAR+采用了“懒更新”策略，即在脏叶子节点被驱逐时不立即写回 NVM，而是依赖 Osiris 机制在后续恢复其最新值。由于被丢弃的叶子节点并未触发父节点的更新，旧的叶子节点版本与旧的父节点版本在完整性树中依然保持一致，因此这种策略不会削弱完整性保护。

4 实验与结果

本节对 ERED 和 STAR+方案的性能进行评估，并与最新的加密与完整性验证技术进行对比分析。

4.1 实验环境与特定配置

评估采用基于工作负载的模拟器，其配置如表 2 所示。模拟器模拟了一个 8 核 x86 处理器，配备 32 KB L1 数据缓存并采用 VAULT 树作为 SIT 树。系统配备 512 KB 模拟元数据缓存，以 16 GB 相变内存（PCM）[29-30] 作为存储介质。测试基准选取 SPEC CPU2006[31] 测试套件中的内存密集型应用。其中，ERED 方案的测试采用 11 个基准测试程序，STAR+方案的测试采用 9 个基准测试程序，以全面评估不同方案在相应工作负载下的性能表现。实验重点考察以下性能指标：位翻转率、写入延迟、平均重加密的行数、写入次数以及性能开销。通过对 ERED 与 STAR+方案在这些指标上的性能数据进行分析，评估其在优化存储系统效率和降低性能开销方面的有效性。

Table 2 Configuration Parameters of Non-Volatile Storage System

表 2 NVM 系统的配置参数

处理器	
配置项	参数设置
CPU	4 核单发射顺序 CMP，主频为 2 GHz
L1 缓存	私有缓存，32 KB，2 路组相联，LRU 替换策略，块大小 64 B，访问延迟 2 周期
L2 缓存	共享缓存，1 MB，8 路组相联，LRU 替换策略，块大小 64 B，访问延迟 20 周期
模拟 PCM 内存系统	
配置项	参数设置
容量	16 GB
延迟	读：75 ns，写：150 ns
安全参数	
配置项	参数设置
安全元数据缓存	容量 512 KB，8 路组相联，块大小 64 B，访问延迟 2 周期，位于 MC（内存控制器）中
SIT	9 层结构，8 路分支，块大小 64 B
缓存树	4 层结构，8 路分支，块大小 64 B
位图行	容量 1 KB，共 16 行，位于 MC（内存控制器）中
多层索引	容量 4 MB，位于 NVM 中
持久性限制	4

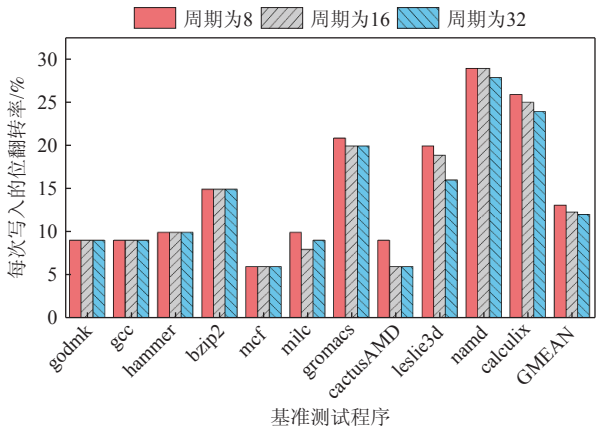


Fig. 9 Impact of epoch size on bit flip rate

图 9 周期大小对位翻转率的影响

4.2 ERED 结果分析

4.2.1 epoch 影响

图 9 给出了不同周期参数下重加密开销的变化情况。随着周期值由 8 逐步提升至 16 和 32，单次写入过程中产生的位翻转率分别由 13.3% 下降至 12.3%，并进一步降低至 12.1%，整体呈现随周期增大而缓慢下降的趋势。然而，从数值差异来看，各周期配置之间的变化幅度较为有限，最大差值不足 1%。因此，本文将周期参数默认设置为 32，已在性能收益

与安全保障之间取得平衡。

4.2.2 W-ERED 实验结果分析

1)位翻转率

如图 10 所示，不同的 NVM 安全保护方案在位翻转率方面表现各异。平均而言，DEUCE 方案将位翻转率降低至 23.2%，而 SECRET 通过消除不必要的写回操作，进一步将位翻转率降至 17%。然而，在某些应用中，特定字的频繁修改导致局部计数器频繁溢出，传统方案需要重加密整个数据行，从而增加了位翻转率。

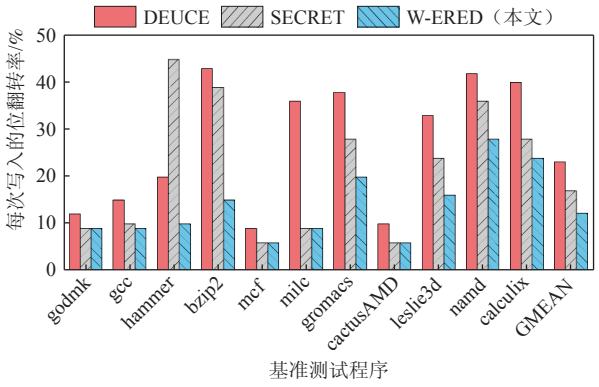


Fig. 10 Write bit flip rate of different schemes

图 10 不同方案的写入位翻转率

W-ERED 通过识别并跟踪计数器溢出状态，仅加密局部计数器溢出的字，有效规避了 SECRET 方案的局限性。同时，该方案在自然驱逐过程中消除了对叶子节点的写回，进一步优化了性能。最终，W-ERED 方案将位翻转率降至 12.1%，比 SECRET 降低了 5%，显著提升了 NVM 的存储效率和写入能耗优化能力。

2)写入延迟

如图 11 所示，不同的 NVM 安全保护方案在写入延迟方面表现各异。DEUCE 方案通过字级 DCW 技术，将写入延迟降低 19%。SECRET 方案则进一步结合字级 DCW、零字标志和能量掩码技术，实现 39% 的延迟降低。然而，在某些应用(如 bzip2)中，由于局部计数器频繁溢出，写入延迟显著增加。

W-ERED 通过选择性地仅对发生局部计数器溢出的数据字执行重加密，避免了溢出时触发整行重加密所带来的冗余加密更新，从而减少了写入过程中不必要的加密处理开销，降低写入延迟。最终，W-ERED 方案实现了平均 50% 的延迟降低，比 SECRET 方案额外减少 11%，显著提升了 NVM 的存储性能与能效。

上述结果表明，W-ERED 在写入延迟方面具有最为显著的改进效果。鉴于其对写入关键路径的直接

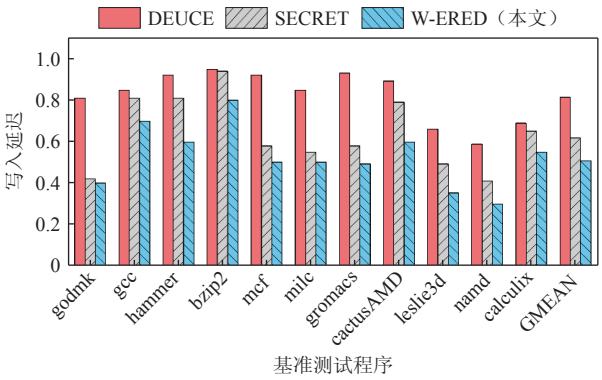


Fig. 11 Write latency of different schemes

图 11 不同方案的写入延迟

优化作用，本文在写入延迟分析中重点讨论 W-ERED 的表现。具体而言，写入延迟主要取决于写路径中重加密操作的粒度以及额外写回的数据量。W-ERED 通过字级选择性重加密机制，直接减少单次写操作中不必要的重加密更新，对写入关键路径具有最直接且普遍的影响。相比之下，L-ERED 主要针对行计数器溢出事件触发的额外重加密开销进行优化，通过减少溢出情况下需要重加密的行数来有效降低写放大。由于该方案主要优化溢出场景下的重加密行数削减，对常规写操作的平均写入延迟影响相对有限，因此其收益更适合通过“平均重加密行数”等指标进行刻画。而 STAR+方案侧重于优化元数据写回策略与崩溃恢复流程，旨在提升系统级的一致性保障效率，对正常运行阶段写入延迟影响不显著。因此本文在写入延迟部分以 W-ERED 作为代表性方案进行重点分析，其结果体现了写路径优化带来的主要延迟收益。

3)写入能耗分析

如图 12 所示，本文对 DEUCE, SECRET, W-ERED 三种方案的写入能耗进行了对比分析。实验结果表明，W-ERED 通过避免局部计数器溢出触发的整行

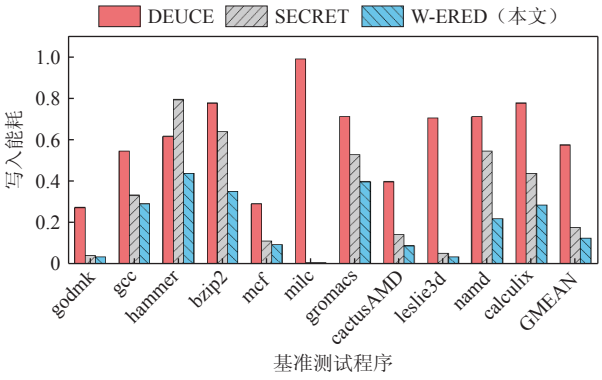


Fig. 12 Write energy consumption of different schemes

图 12 不同方案的写入能耗

重加密，有效减少了冗余写回操作，从而在各类工作负载下均表现出更低的写入能耗。综合各工作负载的统计结果表明，与 SECRET 相比，W-ERED 的写入能耗平均降低约 6%，验证了其在写入效率与能效方面的优势。

4.2.3 L-ERED 实验结果分析

1）重加密行数

L-ERED 是在 W-ERED 基础上进一步扩展得到的优化机制，其核心目标是降低行计数器溢出时触发的大规模重加密开销。图 13 对 3 种行计数器溢出处理策略在溢出事件发生时的平均重加密行数进行了比较，所有方案均构建于 W-ERED 框架之上。在传统 BASE 方案中，一旦行计数器溢出，系统需要暂停当前执行流程，对该计数器所属页面中的全部数据行完成重加密操作后才能继续运行，因此会带来较高的阻塞开销。RSR 方案则通过引入 RSR 寄存器对重加密进度进行管理：对于已经驻留在缓存中的数据行，仅对其状态进行标记，而不立即执行重加密，从而在一定程度上缓解阻塞问题。

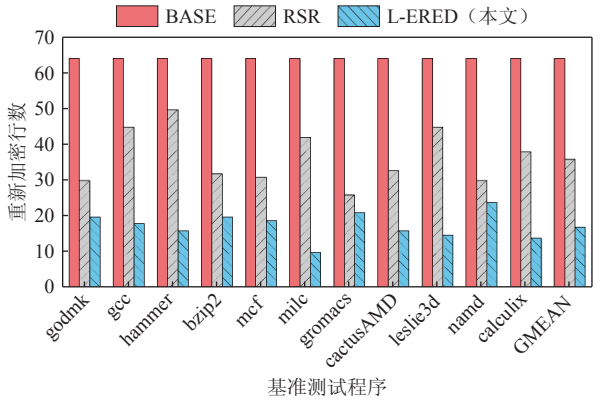


Fig. 13 The number of re-encryption lines of different schemes

图 13 不同方案的重加密行数

与上述 2 种策略不同，L-ERED 通过在行计数器层面引入溢出标志，仅对实际发生溢出的数据行执行重加密，而对未触发溢出的数据行继续保持原有加密状态。该按行选择性的处理方式显著缩小了重加密范围。

实验结果表明，在行计数器溢出场景下，L-ERED 的平均重加密行数分别仅为 BASE 方案的 26.7% 以及 RSR 方案的 47.7%。相比之下，所需重加密的数据规模大幅下降，从而有效降低计算负担，并提升 NVM 系统的整体加密效率与写入性能。

4.2.4 ERED 的存储与实现开销

基于 SECRET 架构，W-ERED 对数据行内 8B 数据单元的局部计数器状态进行感知，以支持按字粒度的重加密调控。为实现该机制，W-ERED 为每个数据字引入 1 b 的溢出标志，用于指示该字在当前周期内是否发生过局部计数器溢出。在本文采用的默认配置下，缓存行大小为 64 B、字粒度为 8 B，因此每个缓存行包含 8 个字，相应地，每个缓存行额外引入 8 b 的元数据，其存储开销为 8/512，约为 1.56%。该溢出标志属于轻量级安全元数据，其组织方式与 SECRET 中已有的零字标志类似，无需引入独立的元数据存储结构或额外的存储访问操作。

在计算开销方面，W-ERED 的改进主要体现在对字级重加密行为的精细化控制。相较于 SECRET 在局部计数器溢出时需要触发整行重加密，W-ERED 通过溢出感知机制将重加密粒度缩小到“仅溢出字”，从而减少冗余的 OTP 生成与异或加密操作次数，降低整体加密计算开销。其新增操作仅包括局部计数器溢出检测、溢出标志维护以及基于标志的计数器选择逻辑，这些功能均可由简单比较器、1 b 状态寄存器和多路选择器等基础硬件逻辑实现，不涉及额外的 AES 运算。由于上述控制逻辑可在 OTP 生成过程中并行执行，因此不会显著增加写入路径上的计算负担。

在此基础上，L-ERED 进一步对行计数器溢出状态进行感知，通过仅对溢出的数据行执行重加密，避免整页重加密带来的冗余加密更新，从而降低整体计算开销。为实现该机制，L-ERED 仅需为每个数据行维护 1b 的行级溢出标志，因此额外存储开销为 1/512，且仅引入极小的控制逻辑开销。

4.2.5 性能与开销变化原因分析

如 4.2.4 节所述，本文所提方法仅引入了少量溢出标志位与轻量级控制逻辑，其附加开销为常数级。此外，溢出检测与计数器选择操作可在密钥流生成阶段同步完成，不增加额外的 AES 运算次数，因此不会带来显著的加密处理延迟。

实验结果中性能提升且系统总体开销下降的主要原因在于：相比新增的轻量级控制成本，ERED 能够在计数器溢出场景下避免现有方案触发整行或整页重加密，从而减少冗余的加密更新与写入操作。在 NVM 存储体系中，写操作具有较高的延迟与能耗代价，且写入更新量的增加会进一步导致位翻转率上升和性能波动。本文通过溢出感知与选择性重加密机制，将加密更新粒度限定在实际发生溢出的数据单元范围内，尤其在内存密集型工作负载中能够有效缓解写放大效应，并最终体现为写入延迟、位翻

转率及能耗的同步下降。

4.3 STAR+结果分析

4.3.1 写入次数

STAR+方案的核心优势在于通过消除叶子节点的NVM写操作，并结合Osiris技术恢复被丢弃的叶子节点内容，从而显著降低了写回操作的开销。为全面评估其效果，我们对比了WB(无恢复能力的基准方案)、Anubis、Phoenix、STAR以及STAR+五种方案的写入，具体结果如图14所示。相较于基准WB方案，Anubis因其元数据跟踪机制实现可恢复性，导致写入次数额外增加了87%。Phoenix在此基础上进行了优化，通过改进叶子节点跟踪策略并利用Osiris恢复，将此额外写入开销有效控制在15%。随后，STAR方案进一步消除了针对元数据缓存的额外写入，但其位图管理机制仍带来了6.8%的写入次数增长。最终，STAR+凭借其对叶子节点写回操作的优化以及Osiris恢复，表现最为突出：其NVM写入次数不仅比STAR方案显著减少了9.8%，甚至比基准WB方案还要低3.0%，证明了其在减少写放大方面的卓越效率。

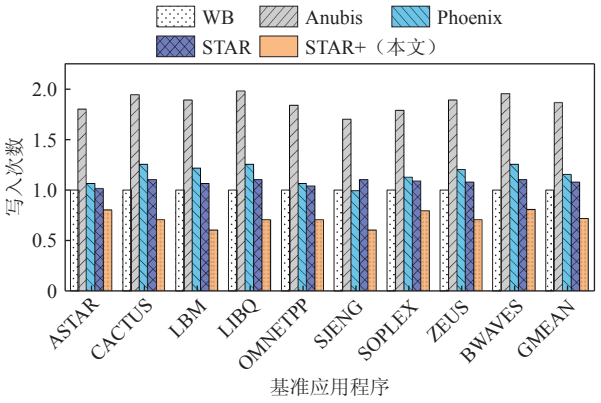


Fig. 14 Write frequency comparison of different schemes

图14 不同方案的写入次数对比

4.3.2 STAR+性能开销

图15展示了5种方案的性能开销。相较于基准WB方案，Anubis为保障SIT的崩溃后恢复能力，引入了8.6%的额外性能开销。Phoenix通过其改进的脏叶子节点地址跟踪及Osiris恢复机制，成功将额外性能开销降至3.2%。STAR技术则利用在子节点中保存父节点修改的策略来确保可恢复性，使额外性能开销降至1.2%。在此基础上，STAR+对叶子节点写回策略进行了优化(同样依托Osiris恢复机制)，实现了最佳的性能表现。其整体性能开销不仅低于STAR方案3.1%，甚至超越了无恢复能力的基准WB方案，达到了1.9%的净性能提升(即负1.9%的开销)。这充分证明了STAR+在提供强大恢复能力的同时，能够以极低开销高效运行。

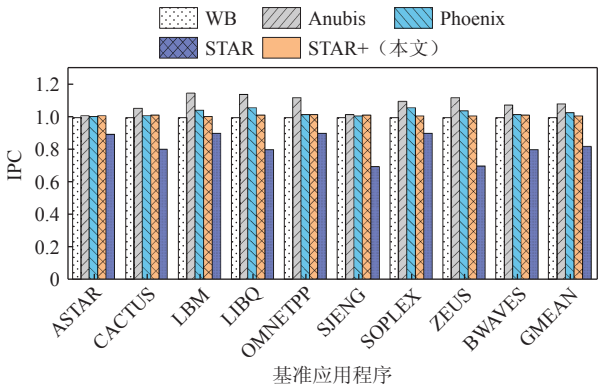


Fig. 15 Comparison of IPC among different schemes

图15 不同方案的IPC对比

5 结 论

本文针对现有安全NVM写优化技术的局限性，提出了一系列改进方案，以提升系统的性能、耐用性和数据安全性。首先，本文提出计数器溢出可感知的安全重加密延迟优化技术，即ERED，该方案分为2个子技术：W-ERED和L-ERED。W-ERED加密方案通过追踪局部计数器的溢出状态，仅对溢出的字重加密，避免对未溢出计数器的重复加密。该方案有效减少了不必要的写操作和加密计算开销，相比传统加密方法，显著降低了位翻转率，优化了能耗并减少了写入延迟。

其次，本文提出L-ERED方案，聚焦于降低行级别的加密负担。L-ERED通过智能识别行计数器的溢出状态，选择性地重加密受影响的数据行，避免了全页面重加密的高计算成本，从而显著减少系统运行时的停顿时间和能耗。

此外，为解决SIT频繁更新尤其是叶子节点的高频写入的问题，本文提出了STAR+方案。该方案通过引入计数器写过滤机制，显著减少了元数据写入次数，提升了系统整体性能。同时，STAR+保障了崩溃一致性，并实现安全元数据的快速恢复，进一步增强了系统的可靠性和稳定性。

实验结果表明，ERED和STAR+方案在减少写入错误、降低能耗和延迟方面表现优越，显著提升了系统性能。特别是在安全元数据管理和崩溃恢复的关键场景中，这些优化措施展现出了卓越的效果，为构建高效、安全的NVM系统提供了有效的技术支持。

针对更高等级的安全需求，后续研究可考虑将本文提出的ERED、STAR+机制与ORAM类存储模式（如PCD-ORAM[19]，HM-ORAM[20]）进行正交叠加，以进一步增强系统抵御访问模式侧信道攻击的能力。ORAM通过隐藏逻辑访问与物理位置之间的映射关系，能与本文在数据机密性、完整性及崩溃一致性方面的优化形成功能互补。在具体实现中，需要实现认证元数据与ORAM逻辑块标识之间的映射对齐，并对ORAM关键元数据更新引入可恢复的持久化约束。考虑到ORAM带来的路径搬移开销与写放大效应，未来将进一步评估不同ORAM方案下的整体性能-安全权衡，并开展系统级实现与验证。

作者贡献声明：延航宇负责方案整体设计并撰写、修改论文；陈仁海提出研究方向，并指导论文修改；张雨、史文燕负责部分算法思路和实验方法；吴洪越、冯志勇提出指导意见并修改论文。

参 考 文 献

[1] Diffie W, Hellman M E. Privacy and authentication: An introduction to cryptography[J]. Proceedings of the IEEE, 1979, 67(3): 397−427

[2] Lipmaa H, Rogaway P, Wagner D. CTR-mode encryption[C/OL]// Proc of the 1st NIST Workshop on Modes of Operation. 2000: 1−4. [2024-10-15]. https://www.cs.ucdavis.edu/~rogaway/papers/ctr.pdf

[3] Zhang Ming, Hua Yu, Liu Lurong, et al. A write-optimized re-computation scheme for non-volatile memory[J]. Journal of Computer Research and Development, 2020, 57(2): 243−256 (in Chinese)
(张铭, 华宇, 刘璐荣, 等. 面向非易失内存写优化的重计算方法[J]. 计算机研究与发展, 2020, 57(2): 243−256)

[4] Yu Ge, Nie Tiezheng, Li Xiaohua, et al. The challenge and prospect of distributed data management techniques in blockchain systems[J]. Chinese Journal of Computers, 2021, 44(1): 28−54 (in Chinese)
(于戈, 聂铁铮, 李晓华, 等. 区块链系统中的分布式数据管理技术——挑战与展望[J]. 计算机学报, 2021, 44(1): 28−54)

[5] Rogers B, Chhabra S, Prvulovic M, et al. Using address independent seed encryption and bonsai Merkle trees to make secure processors OS- and performance-friendly[C]//Proc of the 40th Annual IEEE/ACM Int Symp on Microarchitecture. Los Alamitos, CA: IEEE Computer Society, 2007: 183−196

[6] Schunter M. Intel software guard extensions: Introduction and open research challenges[C]//Proc of the 2nd Int Workshop on Software Protection. New York: ACM, 2016: 1−1

[7] Hoekstra M, Lal R, Pappachan P, et al. Using innovative instructions to create trustworthy software solutions[C]//Proc of the 2nd Int Workshop on Hardware and Architectural Support for Security and Privacy. New York: ACM, 2013: 11−18

[8] Fei Shufan, Yan Zheng, Ding Wenxiu, et al. Security vulnerabilities of SGX and countermeasures: A survey[J]. ACM Computing Surveys, 2021, 54(6): 1−36

[9] Wei Xueliang, Yang Mingshun, Feng Dan, et al. Coordinated metadata management for secure persistent memory[J]. Journal of Computer Research and Development, 2022, 59(11): 2437−2450 (in Chinese)
(魏学亮, 杨明顺, 冯丹, 等. 面向安全持久性内存的元数据协同管理方法[J]. 计算机研究与发展, 2022, 59(11): 2437−2450)

[10] Young V, Nair P J, Qureshi M K. DEUCE: Write-efficient encryption for non-volatile memories[C]//Proc of the 20th Int Conf on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2015: 33−44

[11] Swami S, Rakshit J, Mohanram K. SECRET: Smartly EnCRypted energy efficient non-volatile memories[C]//Proc of the 53rd Annual Design Automation Conf. New York: ACM, 2016: 166: 1−166: 6

[12] Zhu He, Hua Qiangsheng, Jin Hai, et al. LMSA: A high performance dynamic graph processing data structure for non-volatile memory systems[J]. Chinese Journal of Computers, 2022, 45(7): 1446−1461 (in Chinese)
(祝贺, 华强胜, 金海, 等. LMSA: NVM环境下高性能动态图处理数据结构[J]. 计算机学报, 2022, 45(7): 1446−1461)

[13] Ye Mao, Hughes C, Awad A. Osiris: A low-cost mechanism to enable restoration of secure non-volatile memories[C]//Proc of the 51st Annual IEEE/ACM Int Symp on Microarchitecture. Los Alamitos, CA: IEEE Computer Society, 2018: 403−415

[14] Zubair K A, Awad A. Anubis: Ultra-low overhead and recovery time for secure non-volatile memories[C]//Proc of the 46th Int Symp on Computer Architecture. New York: ACM, 2019: 157−168

[15] Al-Wadi M, Zubair K A, Mohaisen D, et al. Phoenix: Towards ultra-low overhead, recoverable, and persistently secure NVM[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(2): 1049−1063

[16] Huang Jianming, Hua Yu. A write-friendly and fast-recovery scheme for security metadata in non-volatile memories[C]//Proc of the 27th IEEE Int Symp on High-Performance Computer Architecture. Los Alamitos, CA: IEEE Computer Society, 2021: 359−370

[17] Zubair K A, Abdullah R, Mohaisen D, et al. RC-NVM: Recovery-aware reliability-security co-design for non-volatile memories[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 21(4): 1817−1830

[18] Freij A, Zhou Huiyang, Solihin Y. SecPB: Architectures for secure non-volatile memory with battery-backed persist buffers[C]//Proc of the 29th IEEE Int Symp on High-Performance Computer Architecture. Piscataway, NJ: IEEE, 2023: 677−690

[19] Wang Yi, Wang Zhencheng, Chen Weixuan, et al. PCD-ORAM: A path-aware and cross-layer design to enhance data locality in oblivious RAM[J/OL]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2026, 45(6): 2898−2911

[20] Liu Gang, Xiao Zheng, Li Kenli, et al. HM-ORAM: A lightweight crash-consistent ORAM framework on hybrid memory system[J]. ACM Transactions on Storage, 2025, 21(2): 1−28

[21] Goswami B, Moorthii C J, Bansal H, et al. Investigation of security vulnerabilities in NVM-based persistent TinyML hardware[J]. IEEE Embedded Systems Letters, 2024, 17(3): 204−209

[22] Taassori M, Shafiee A, Balasubramonian R. VAULT: Reducing paging overheads in SGX with efficient integrity verification structures[C]//Proc of the 23rd Int Conf on Architectural Support for Programming Languages and Operating Systems. New York: ACM, 2018: 665−678

[23] Guo Zilong, Chang Liang, Liu Jianwei, et al. SENTRY: Lifetime secure technique for industry embedded non-volatile random-access memory[J]. Journal of Computer Science and Technology, 2025, 40(6): 1530−1545

[24] Huang Jianming, Hua Yu. Root crash consistency of SGX-style integrity trees in secure non-volatile memory systems[C]//Proc of the 29th IEEE Int Symp on High-Performance Computer Architecture. Piscataway, NJ: IEEE, 2023: 152−164

[25] Shi Yujie, Hua Yu, Huang Jianming. A high-performance and fast-recovery scheme for secure non-volatile memory systems[C]//Proc of the 26th IEEE Int Conf on Cluster Computing. Piscataway, NJ: IEEE, 2024: 451−463

[26] Zhou Ping, Zhao Bo, Yang Jun, et al. A durable and energy efficient main memory using phase change memory technology[C]//Proc of the 36th Int Symp on Computer Architecture. New York: ACM, 2009: 14−23

[27] Cho S, Lee H. Flip-N-Write: A simple deterministic technique to improve PRAM write performance, energy and endurance[C]//Proc of the 42nd Annual IEEE/ACM Int Symp on Microarchitecture. Los Alamitos, CA: IEEE Computer Society, 2009: 347−357

[28] Kong Jingfei, Zhou Huiyang. Improving privacy and lifetime of PCM-based main memory[C]//Proc of the 2010 IEEE/IFIP Int Conf on Dependable Systems and Networks. Piscataway, NJ: IEEE, 2010: 333−342

[29] Raoux S, Burr G W, Breitwisch M J, et al. Phase-change random access memory: A scalable technology[J]. IBM Journal of Research and Development, 2008, 52(4/5): 465−479

[30] Zhao Peng, Zhu Longyun. A scheme for protecting confidentiality of non-volatile main memory based on phase-change memory[J]. Chinese Journal of Computers, 2011, 34(11): 2114−2120 (in Chinese)
(赵鹏, 朱龙云. 基于相变存储器的非易失内存数据机密性保护[J]. 计算机学报, 2011, 34(11): 2114−2120)

[31] Henning J L. SPEC CPU2006 benchmark descriptions[J]. ACM SIGARCH Computer Architecture News, 2006, 34(4): 1−17



Yan Hangyu, born in 1999. Master. His main research interests include computer storage, artificial intelligence, and algorithm hardware acceleration.

延航宇, 1999 年生。硕士。主要研究方向为计算机存储、人工智能、算法硬件加速。



Chen Renhai, born in 1987. PhD, associate professor. His main research interests include computer architecture, embedded systems, machine learning, artificial intelligence, communication, and transmission.

陈仁海, 1987 年生。博士，副教授。主要研究方向为计算机体系结构、嵌入式系统、机器学习、人工智能、通信、传输。



Zhang Yu, born in 2000. Master. His main research interests include computer storage and algorithm hardware acceleration.

张　雨, 2000 年生。硕士。主要研究方向为计算机存储、算法硬件加速。



Shi Wenyan, born in 1991. Master. Her main research interests include time-series database, distributed storage, and computer architecture.

史文燕, 1991 年生。硕士。主要研究方向为时序数据库、分布式存储、计算机体系结构。



Wu Hongyue, born in 1988. PhD, associate professor. His main research interests include service computing, mobile edge computing, and mobile cloud computing.

吴洪越, 1988 年生。博士，副教授。主要研究方向为服务计算、移动边缘计算、移动云计算。



Feng Zhiyong, born in 1965. PhD, professor. His main research interests include software engineering, artificial intelligence, service computing, knowledge engineering, and computer cognition.

冯志勇, 1965 年生。博士，教授。主要研究方向为软件工程、人工智能、服务计算、知识工程、计算机认知。