SeqAss: 利用串行关联的缓存结构结构来实现能抵御冲突型缓存侧信道攻击的高性能末级缓存

宋 威

中国科学院信息工程研究所
网络空间安全防御全国重点实验室

第29届中国计算机系统研讨会 2025年12月27日

主要内容

缓存侧信道攻击是攻击者通过在受害机器上运行恶意程序（非接触式）从缓存中获取关键信息的一种信息泄露的攻击方式。近年来被大量采用。

缓存随机化是近年来提出的一种纯硬件的缓存防御方法，可以抵御冲突型的缓存侧信道攻击。

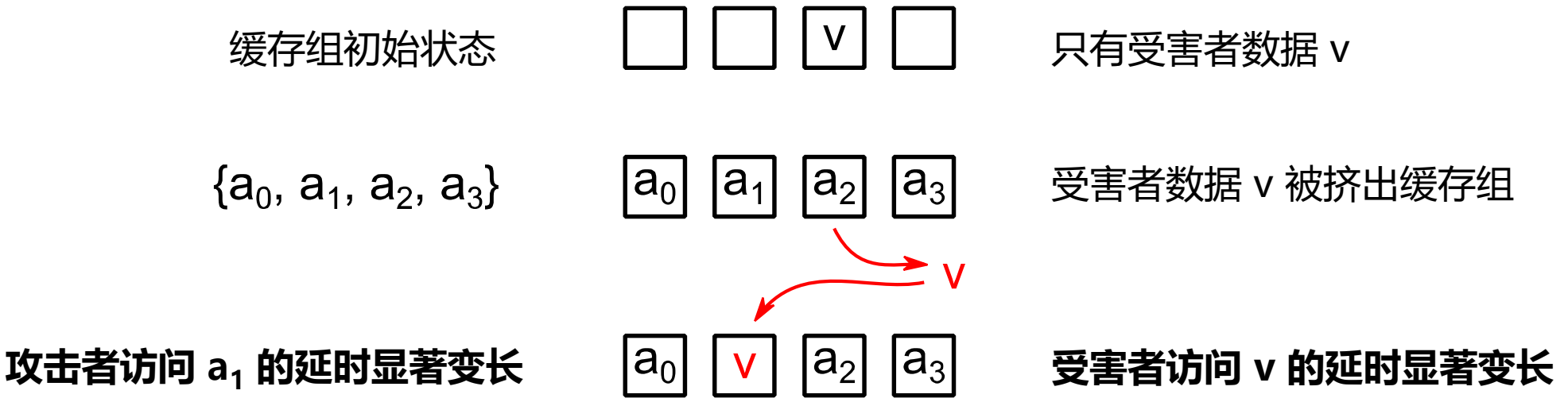
Mirage是当前最先进的随机化缓存结构。但是其元数据冗余供给并分离存储的架构导致了高达**22%**的面积代价和**21%**的附加功耗。

我们提出使用**串行关联**代替skewed缓存，利用**负载估计**代替元数据冗余供给，再引入**按需重定位**和**基于检测的延迟重映射**。所设计的**SeqAss缓存结构**实现了类似Mirage的安全强度，面积和功耗代价下降至**4%**，缓存缺失率下降**11%**。

已被IEEE S&P 2026录用。

Wei Song, Zhidong Wang, Jinchi Han, Da Xie, Hao Ma, and Peng Liu. **SeqAss: Using sequential associative caches to mitigate conflict-based cache attacks with reduced cache misses and performance overhead**. *IEEE Symposium on Security and Privacy (S&P)*, San Francisco, CA, United States, pp. 1371–1388, May 2026.

冲突型缓存侧信道攻击



- 正常情况下，受害者的目标数据 v 在缓存命中
- 攻击者通过访问一个驱逐集 $\{a_0, a_1, a_2, a_3\}$，强迫将 v 挤出缓存.
- 受害者再次访问 v 的延时显著变长，攻击者访问驱逐集的时间也变长。

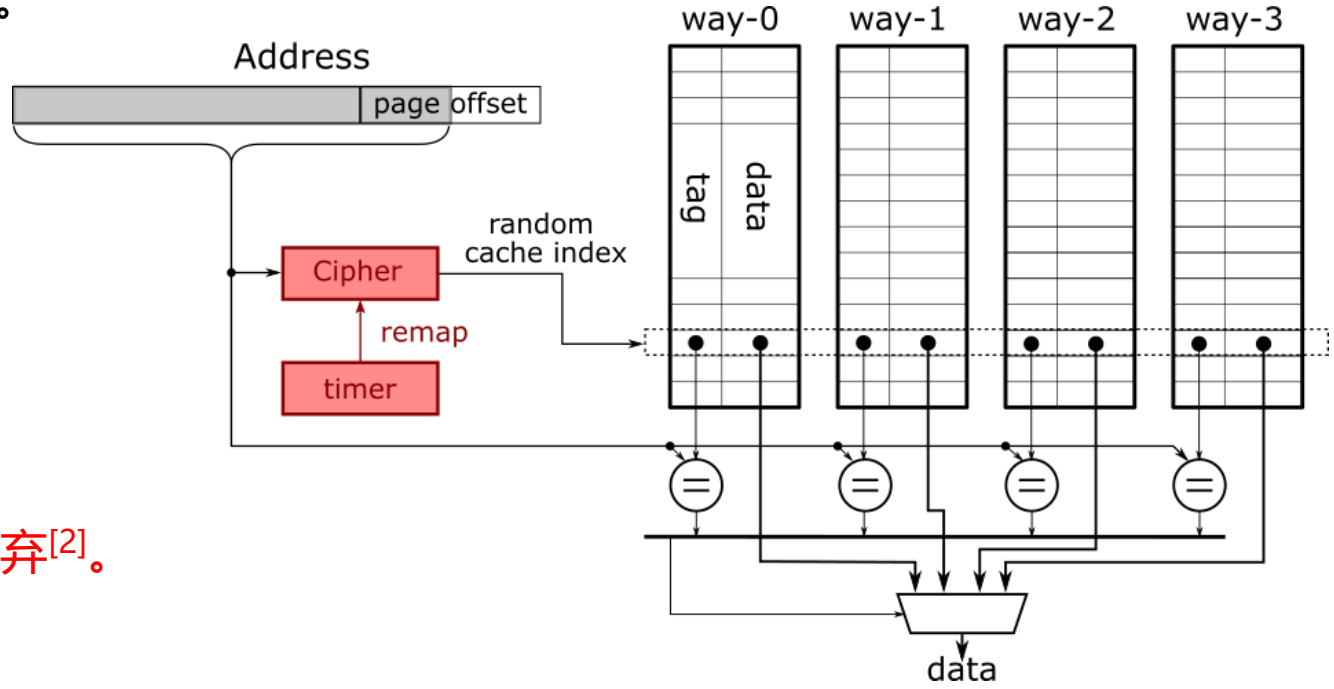
$\{a_0, a_1, a_2, a_3\}$ 和 v 在缓存中保存在同一个缓存组 (congruent)

驱逐集：一个包含足够多congruent地址的集合，访问它可以将目标地址 v 驱逐出缓存。

地址映射随机化：随机化缓存的基础技术

代表设计：**CEASER**[1]

- 地址到缓存组的映射由加密模块（Cipher）随机化。
- 周期性重映射：修改加密密钥并更新缓存映射。



防御作用

- 页内偏移不能用（时间代价 x64）。
- 寻找攻击目标依靠缓存组扫描（时间代价 x100~1000）。

遗留问题

- 每次重映射会造成40%到60%的缓存数据被遗弃[2]。
- 攻击者能够在重映射周期内找到驱逐集 [2]！

[1] M. K. Qureshi. "CEASER: Mitigating conflict-based cache attacks via encrypted-address and remapping". MICRO 2018.
[2] W. Song, B. Li, Z. Xue, et al. "Randomized last level caches are still vulnerable to cache side channel attacks! But we can fix it." S&P 2021.

缓存随机插入：降低Congruent地址的有效性

代表设计：**CEASER-S**[1], **ScatterCache**[2], **PhatomCache**[3]

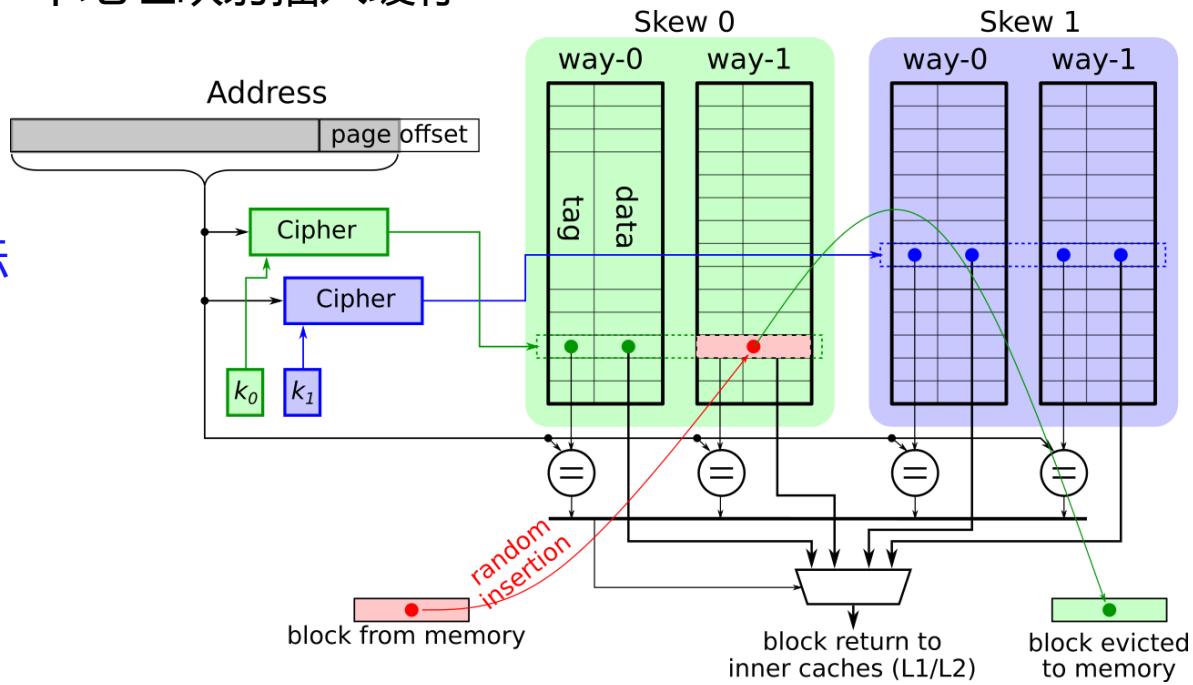
- 准备多个地址映射模块，在插入缓存块时，随机选择一个地址映射插入缓存
- 一般基于skewed缓存结构

防御作用

- 当使用S个映射模块，访问congruent地址可造成目标缓存组冲突的概率下降至$1/S^2$

遗留问题

- 攻击者仍然能找到足够大的驱逐集 [4]。
- 不能阻止持续攻击[5]。



[1] M. K. Qureshi, "New attacks and defense for encrypted-address cache." ISCA 2019.
[2] M. Werner, T. Unterluggauer, L. Giner, et al. "ScatterCache: Thwarting cache attacks via cache set randomization." USENIX Security 2019.
[3] Q. Tan, Z. Zeng, K. Bu, and K. Ren, "PhantomCache: Obfuscating Cache Conflicts with Localized Randomization." NDSS 2021.
[4] W. Song, B. Li, Z. Xue, et al. "Randomized last level caches are still vulnerable to cache side channel attacks! But we can fix it." S&P 2021.
[5] T. Bourgeat, J. Drean, Y. Yang, et al. "CaSA: End-to-end Quantitative Security Analysis of Randomly Mapped Caches." MICRO 2020.

全局间接置换：阻止攻击者寻找Congruent地址

代表设计：**Chameleon**[1]

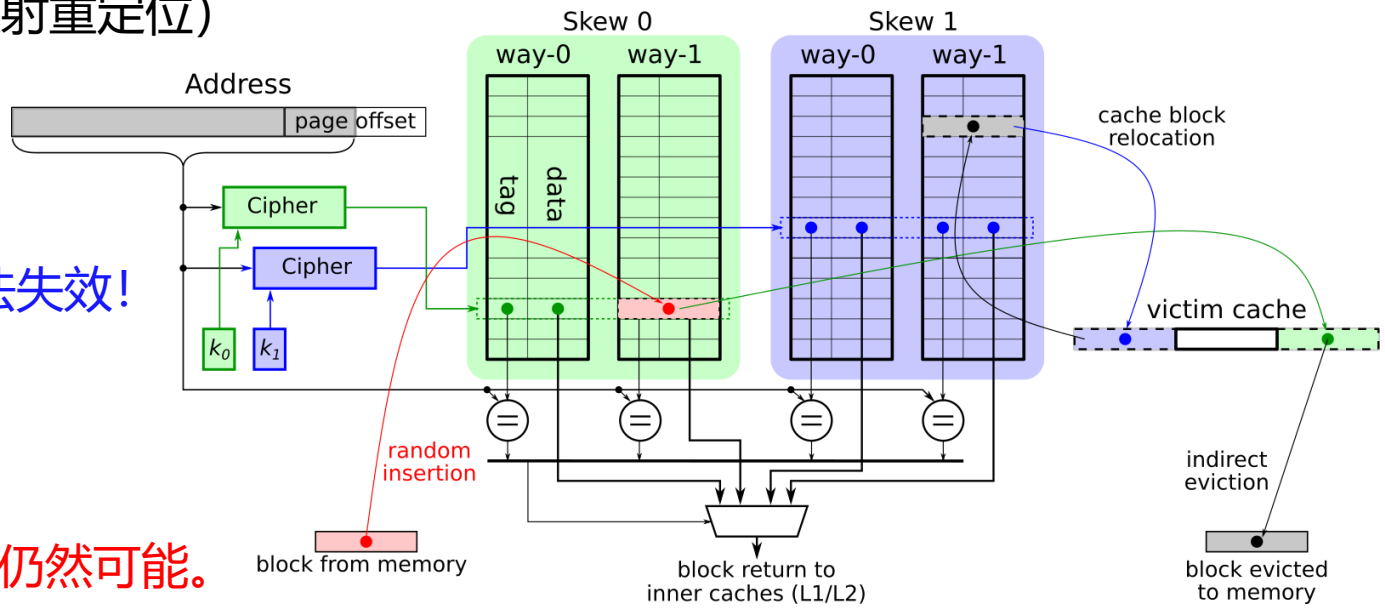
- 当缓存块被挤出缓存，将其放入VC，从VC中挤出另外一个缓存块。
- VC中的缓存块被重新插入缓存（利用另外的映射重定位）

防御作用

- 攻击者挤出的缓存块和造成冲突的缓存块没有congruence关系，绝大部分的驱逐集寻找算法失效！

遗留问题

- Write+Write算法不依赖于这种关系[2]。
- 如果有足够多的congruent地址，Evict+Time仍然可能。
- 重定位造成了高达70%的动态功耗代价。



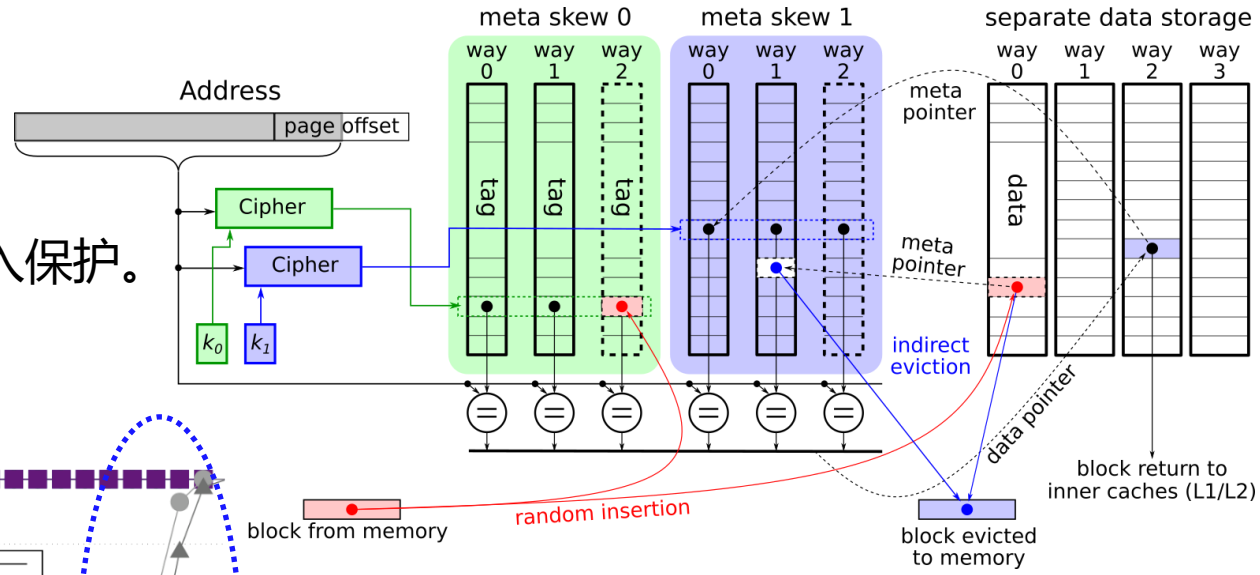
[1] T. Unterluggauer, A. Harris, S. Constable, et al. "Chameleon Cache: Approximating Fully Associative Caches with Random Replacement to Prevent Contention-Based Cache Attacks," SEED 2022.

[2] J. P. Thoma and T. Güneysu, "Write Me and I'll Tell You Secrets – Write-After-Write Effects On Intel CPUs." RAID 2020.

全相联缓存：阻止由Congruent地址引发的特定冲突

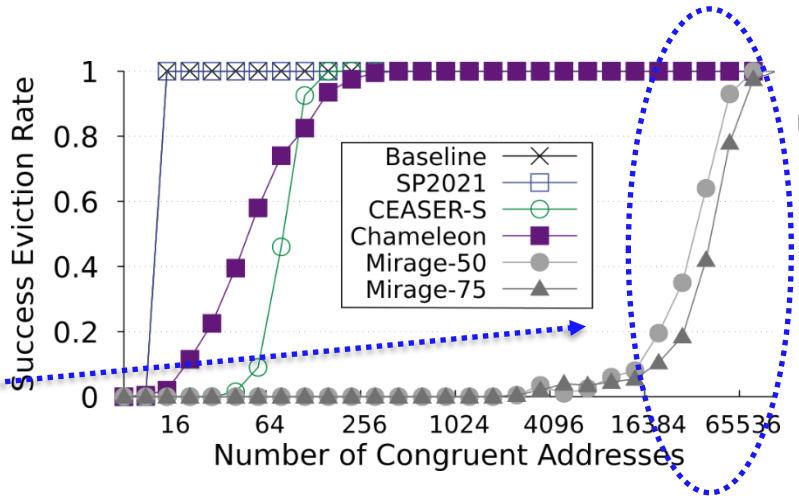
代表设计：**Mirage**[1]

- 数据和元数据分离，缓存块在数据存储中完全随机插入，位置由元数据记录。
- 冗余供给元数据存储空间，利用负载均衡的随机插入保护。



防御作用

- 无法攻击数据存储，congruent关系失效。
- 负载均衡的随机插入让元数据存储难以被攻击。



遗留问题

- 数据和元数据分离、以及元数据冗余供给造成了**高达22%和21%的面积和功耗代价。**
- 强制使用随机置换算法。

[1] G. Saileshwar and M. Qureshi, "MIRAGE: Mitigating Conflict-Based Cache Attacks with a Practical Fully-Associative Design," USENIX Security 2021.

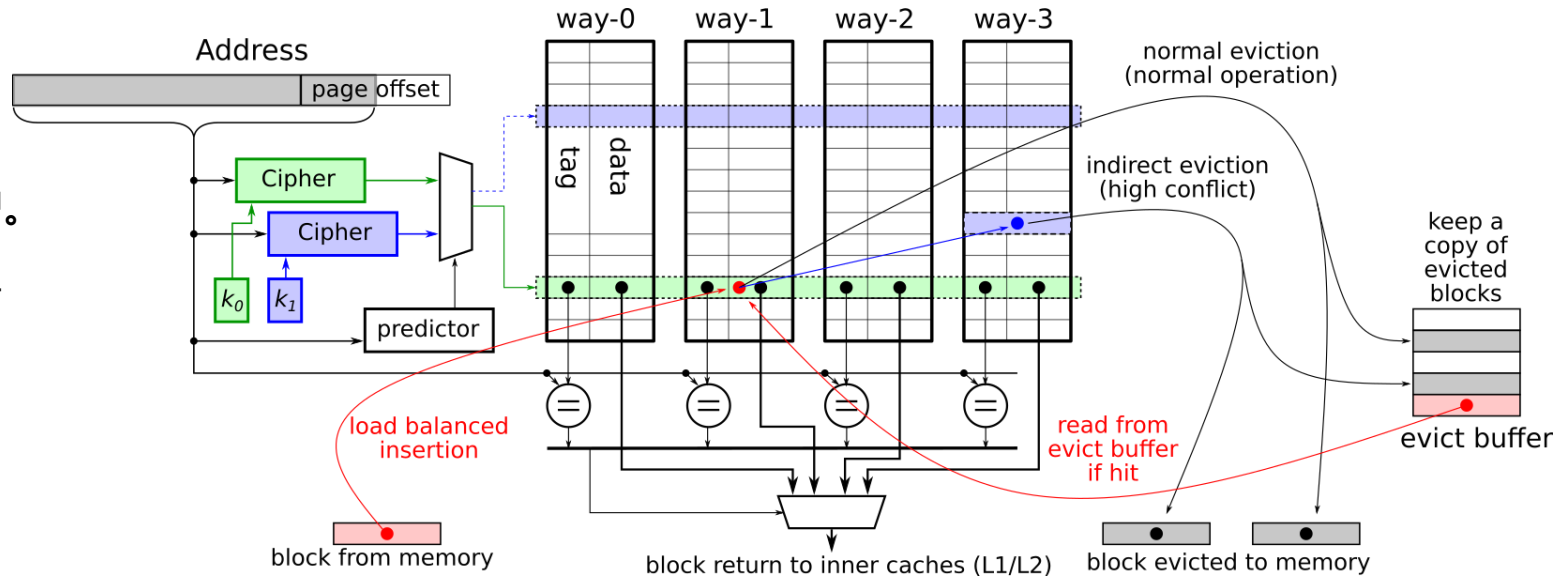
我们需要在不降低安全性的同时显著降低性能代价！

现状描述	问题描述
● **Mirage是当前最先进的随机缓存结构：** □ 数据的全相联存储阻止任何形式的攻击 □ 元数据存储成为了弱点，但是受到了负载均衡的保护	**元数据的安全性决定了Mirage的安全性。已经足够安全！**
● **数据的全相联存储**要求**数据和元数据分离**	双向指针索引，增大面积和功耗。
● 元数据存储的**负载均衡**要求**元数据冗余供给**	增大50%到75%的元数据存储面积，增大功耗，降低访存速度。
● **数据的全相联存储**强制**随机置换算法**	对于访存密集型程序，缓存命中率下降5%~20%。

利用串行关联搭建随机化缓存：SeqAss缓存

主要思路

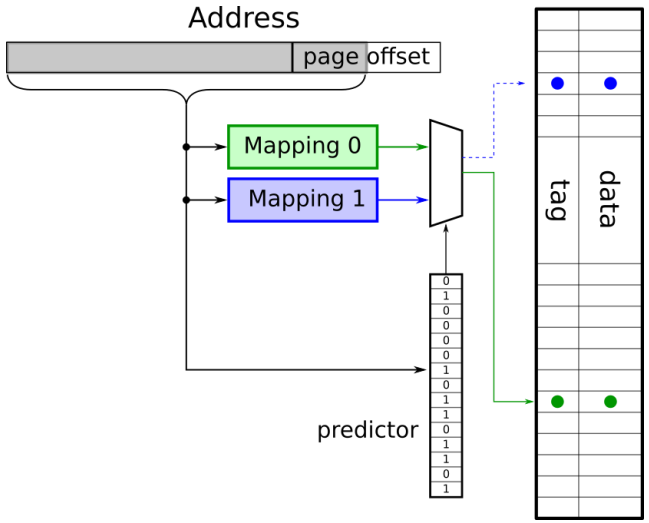
- **放弃全相联**，代价太大，过度保护。
- 将元数据和数据一同保护，沿用传统的**多路组相联**。
- **负载均衡的随机插入是关键**，利用**串行关联**和**负载估计计数器**实现。



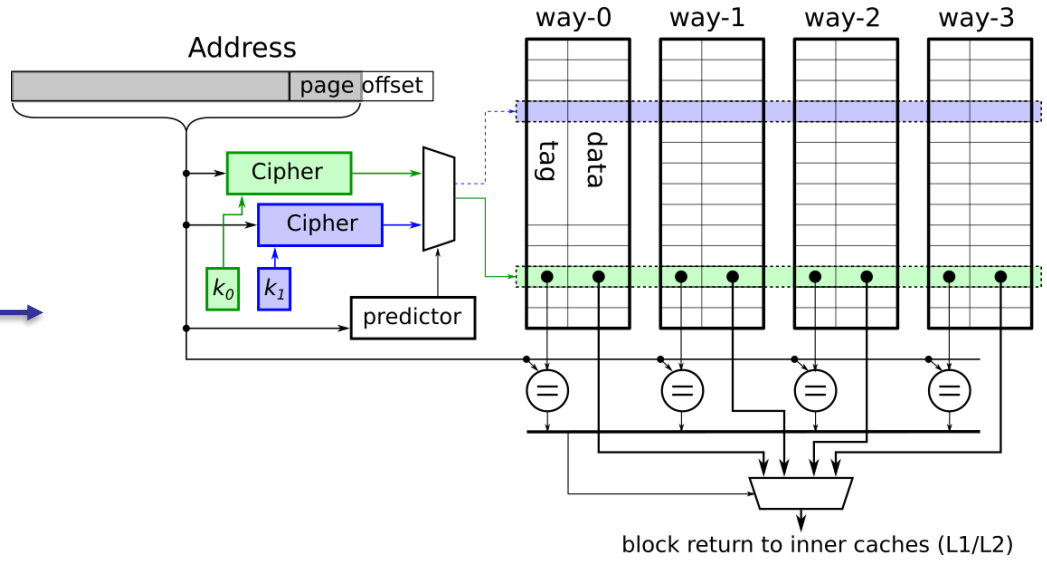
防御技术	Mirage的实现方式	SeqAss的实现方式
全相联的数据存储	数据与元数据分离	不采用，沿用传统的**多路组相联**
缓存块的随机插入	全局随机置换	不采用，使用修改的**LRU置换算法**
元数据的随机插入	Skewed缓存构成的元数据存储	使用**串行关联结构**
负载均衡的随机插入	元数据50%~75%冗余供给	使用单独的**负载估计计数器**

串行关联的多路组相联缓存

原有串行关联结构



引入多路组相联缓存



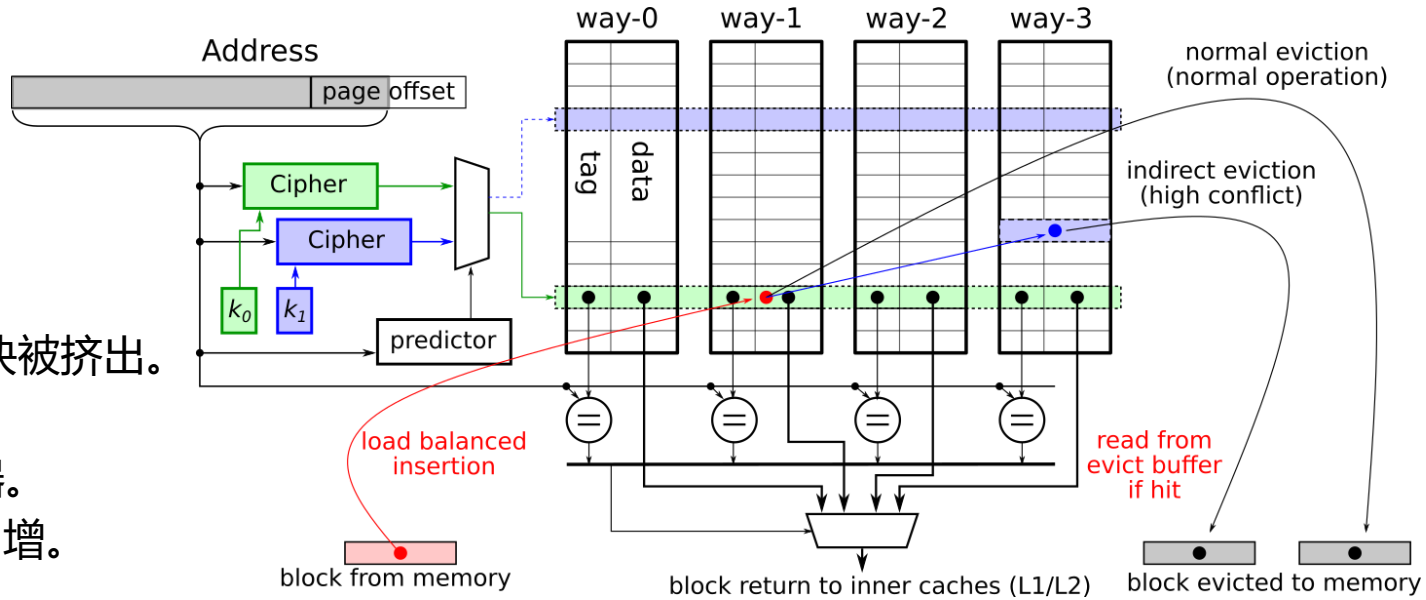
- 利用多个cipher为每个缓存块提供多个缓存组，随机选择。
- 缓存保持多路组相联结构。
- 在放访问时按序串行检查所有可能的缓存组。（缓存访问时间边长）
- 通过引入预测器（predictor），每个缓存块对应有1bit的cipher选择预测，90%的缓存首次检查命中。

以0.1周期的附加访问延迟，获得skewed缓存类似的效果，对LRU等置换算法没有影响！

基于负载估计计数器的负载均衡和重定位

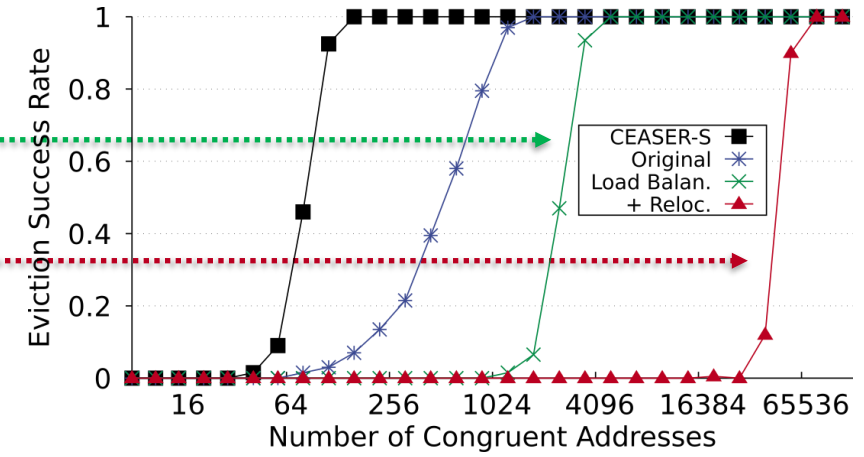
仅有随机插入是远远不够的，负载均衡将进一步增加攻击者引发特定缓存组冲突的难度！

利用负载计数器估计负载，避免元数据冗余。
被攻击的目标缓存组的负载较高。
当缓存组负载较高时，启动重定位，避免缓存块被挤出。



- 为每一个缓存组添加一个3-bit的负载计数器。
  - 每发生一次冲突，发生冲突的计数器自增。
  - 冲突同时触发全局的计数器衰减。
  - 计数器的值稳定在1左右。

- **负载均衡：在插入时，选择负载低的缓存组。**
- **重定位：如果被插入的缓存组为高负载（计数器值>1），触发重定位，导致不相关的缓存块被逐出。**

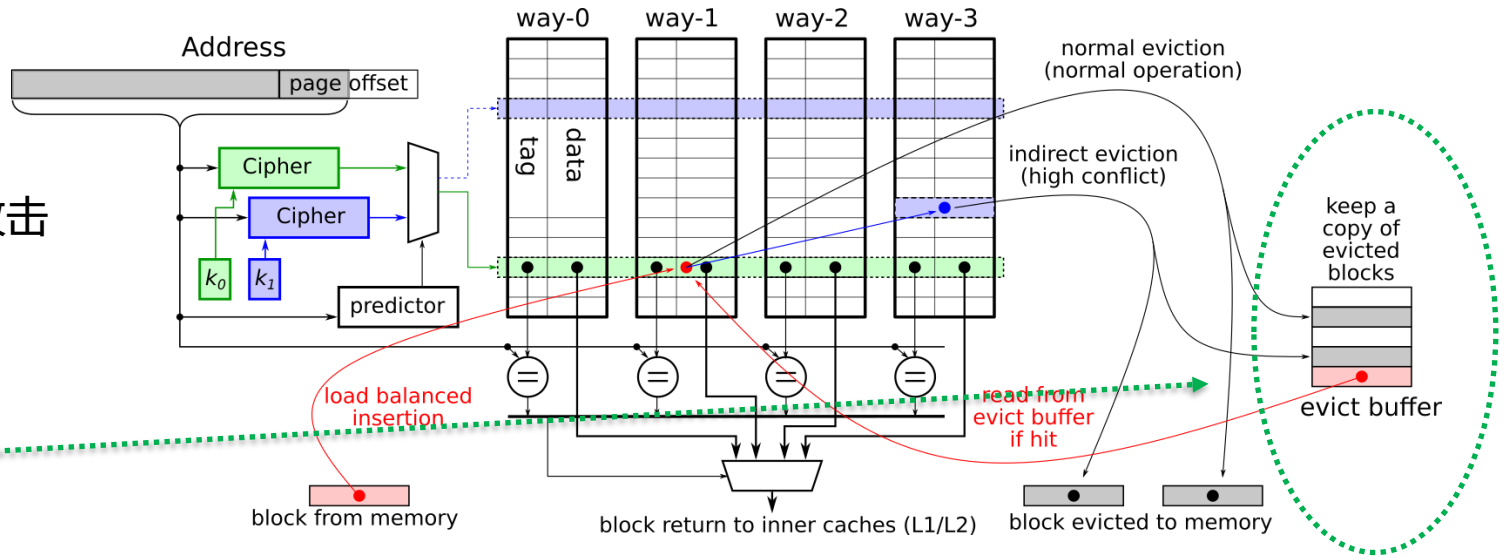


基于乒乓访问模式的攻击检测和延迟重映射

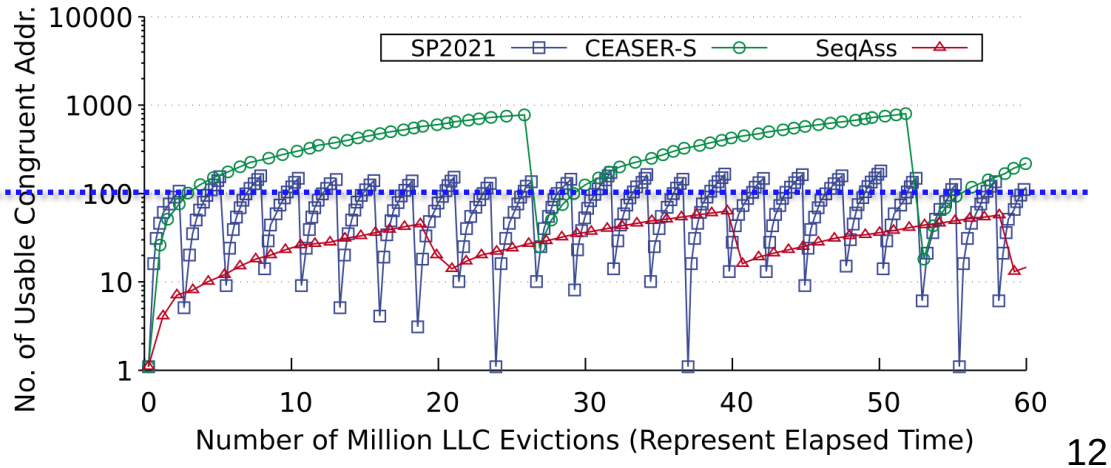
攻击者已经很难发起侧信道攻击，
驱逐目标地址需要3万多congruent地址！

但是攻击者可能通过congruent地址和持续攻击
逆向随机映射函数。

仅存的驱逐集寻找算法会留下乒乓访问模式，
通过一个16路全相联的驱逐缓冲，
SeqAss缓存可以检测该乒乓模式，
当足够的乒乓访问发生后，**触发延迟重映射。**



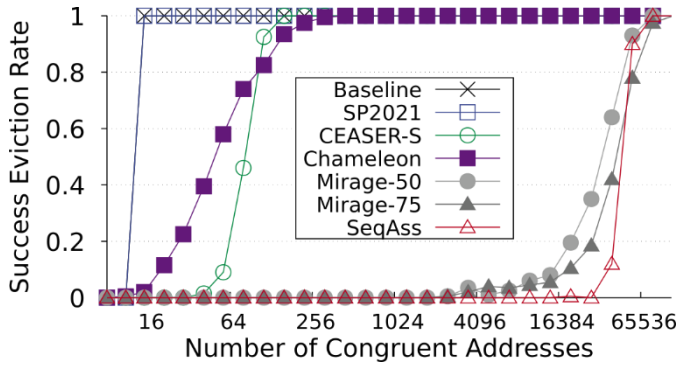
在开启攻击检测和延迟重映射后，攻击者可以获得的
有效congruent地址数恒定小于100。



安全性评估

Structure	CT rate	CT evictions	CT-prefetch rate	CT-prefetch evictions	CTPP rate	CTPP evictions	PPP rate	PPP evictions
Baseline	**100%**	**264K**	**100%**	**16.5K**	**100%**	**549K**	**8.7%**	**443K**
SP2021	**100%**	**263K**	**100%**	**16.8K**	**100%**	**531K**	**7.9%**	**443K**
CEASER-S	**100%**	**259K**	**100%**	**30.5K**	0.4%	1.07M	0.4%	480K
Chameleon	0%	265K	0%	105K	0%	584K	0%	453K
Mirage	0%	253K	0%	247K	0%	609K	0%	469K
SeqAss	**86.1%**	**345K**	**85.7%**	**344K**	0%	604K	0%	455K

虽然CT和CT-prefetch算法仍然可以获得congruent地址，但是非常缓慢，**平均访问34万个地址获得1个congruent地址。**其乒乓模式会触发延迟重映射，从而限制**攻击者总共获得的可用地址数小于100。**



SeqAss的安全强度类似于Mirage，需要3万多个congruent地址才能驱逐目标地址。

Structure	No. of Addresses	Preci-sion	Recall	F1 Score	Possibility
Baseline	16	100%	100%	1	Definitely.
SP2021	16	100%	100%	1	Definitely.
CEASER-S	67	100%	100%	1	Definitely.
Chameleon	14	70%	1.4%	0.027	Unlikely.
Mirage	512	0%	0%	0	Unlikely.
SeqAss	512	0%	0%	0	Unlikely.

Prime+Probe攻击在SeqAss和Mirage缓存上完全失效，攻击的可见度为0。

性能评估：面积消耗

Structure	meta bits per set	cache sets	total meta bits	meta area (mm^2)	bits per block	total data bits	data area (mm^2)	total bits	overhead in bits	total area (mm^2)	overhead in ASIC area
Baseline	16x(28+11)	16K	10.2M	2.267	512	134.2M	28.30	144.7M	—	30.63	—
SP2021	16x(42+12)	16K	14.2M	3.287	512	134.2M	28.30	148.9M	2.93%	31.74	3.62%
CEASER-S	8x(42+12)	32K	14.2M	3.287	512	134.2M	28.30	148.6M	2.73%	31.65	3.33%
Chameleon	8x(42+11)	32K	13.9M	3.216	512	134.2M	28.30	148.1M	2.37%	31.52	2.91%
Mirage-50	12x(42+29)	32K	27.9M	7.175	531	139.2M	29.67	167.1M	15.5%	36.85	20.3%
Mirage-75	14x(42+29)	32K	32.6M	10.92	531	139.2M	29.67	171.8M	18.7%	40.59	32.5%
SeqAss	16x(42+12)+8	16K	14.3M	3.323	512	134.2M	28.30	149.1M	3.04%	31.77	3.72%

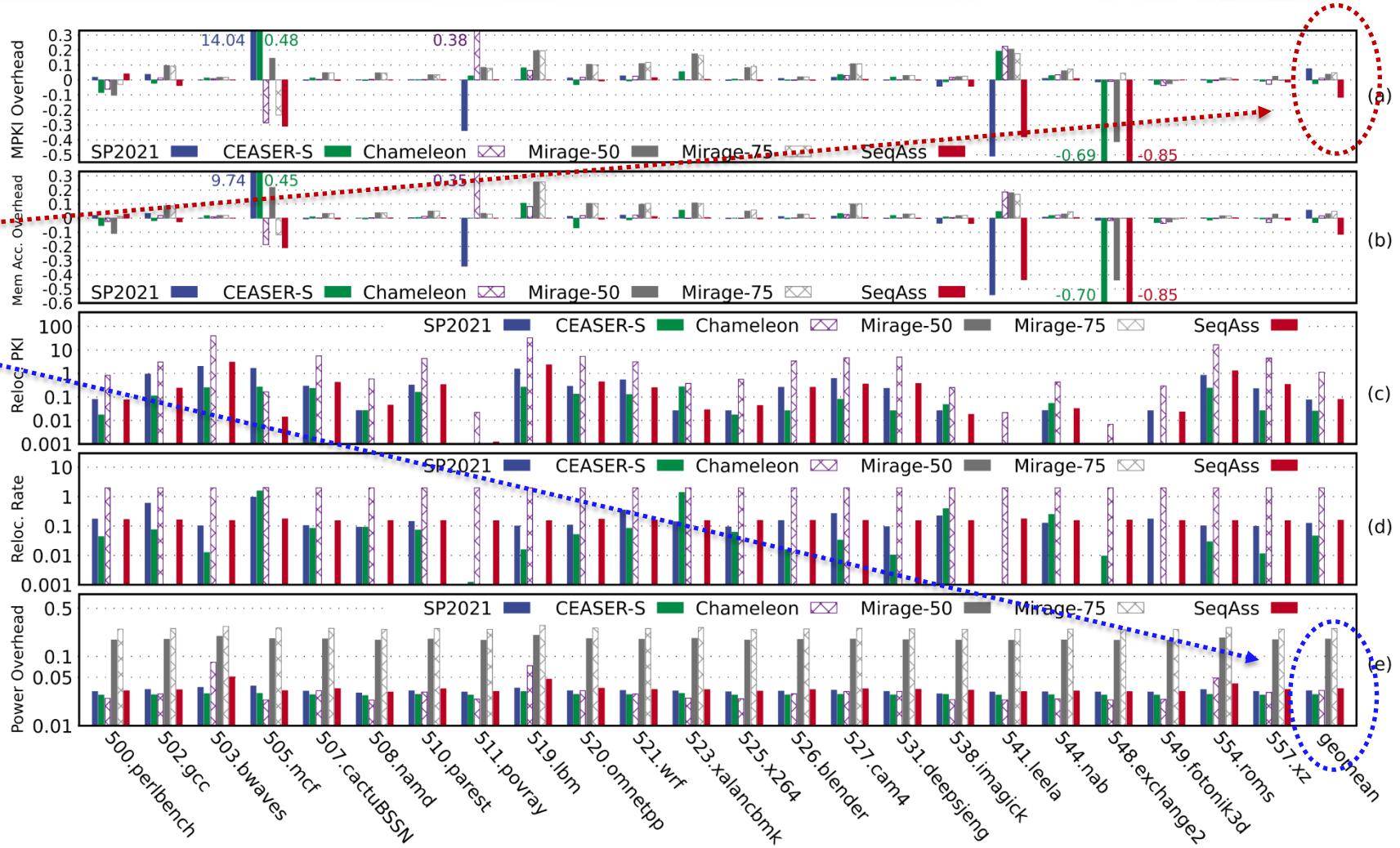
SeqAss的面积代价仅为3.72%，远小于Mirage-50的20.3%和Mirage-75的32.5%。

性能评估：运行时性能

MPKI下降了11.4%!

整体功耗损失3.43%。

运行519.lbm时的附加动态功耗损失26.1%，明显小于Mirage的61%和Chameleon的75%。



技术总结

- Mirage虽然用全相联保护了数据存储，**其元数据存储仍然使用skewed结构，成为了相对弱点。**
- 是否能找到congruent地址也许并不是关键，**关键在于阻止攻击者利用他们发起特定冲突。**
- **基于随机插入的负载均衡时非常强大的防御措施，配合缓存块的重定位能挫败所有的攻击。**
- **Skewed缓存不是实现随机插入的唯一方式**，串行关联同样可以，还维持了传统多路组相联结构。
- **元数据冗余供给并不是实现负载均衡的唯一方式。**一个3-bit的负载计数器达到了同样的效果。
- **缓存随机化不一定导致性能下降**，SeqAss的防御措施同时提高了缓存关联性，缓存缺失率下降11.4%。
- 不要只关注总体功耗的代价，**缓存随机化对动态功耗的影响更为明显。**

谢谢！其他的细节请阅读原文。

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS