NIST Lightweight Cryptography Workshop 2022

Low-Latency Crypto: An Emerging Paradigm of Lightweight Cryptography

Santosh Ghosh, Research Scientist, Intel Labs, Intel Corporation, US

Notices & Disclaimers

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

No product or component can be absolutely secure. Your costs and results may vary. Results have been estimated or simulated.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

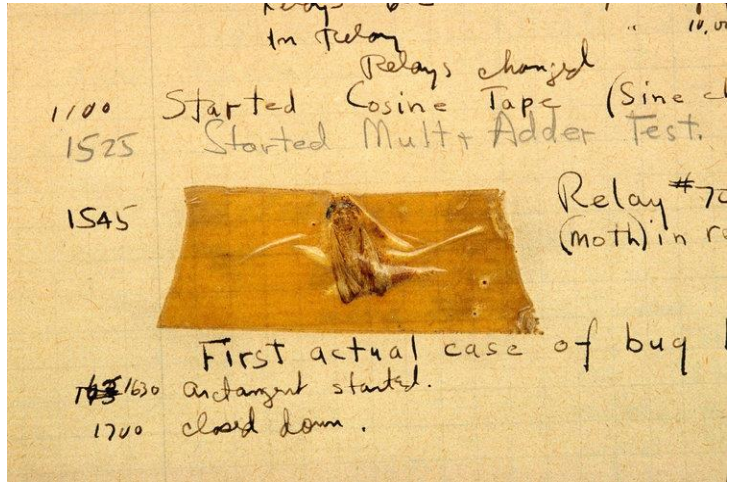
These materials are provided “as is.” Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Outline

- New application of Lightweight Crypto (LWC)
  - SW bugs, vulnerabilities and memory safety
  - Cryptographic Capability Computing (C^3)
  - Low-latency – a new paradigm of LWC
  - Small block size – another new direction of symmetric key encryption
- Latency estimation and evaluation
- Open problems

Memory Safety Issues and Traditional Protections

- Bugs, Vulnerabilities and Exploits are as old as computing



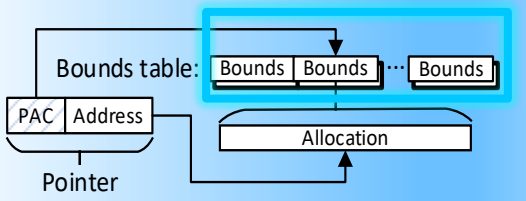
Grace Hopper's operational logbook for the Harvard Mark II computer – back in 1944 [1]

- Microsoft shows ~70% of vulnerabilities are due to memory safety violations [2]

CHERI 128-bit fat pointers [3]:

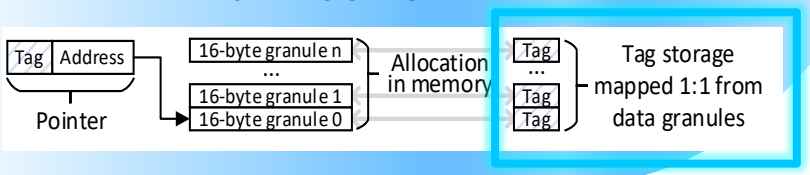
Intel® MPX multi-level bounds tables [4]:

Always-On Memory Safety (AOS) [5]:



Metadata Performance HW/SW changes

ARM® Memory Tagging Extension (MTE) [6]:*



[1] Grace Hopper bug image and text from https://www.businessinsider.com/harvard-mark-i-grace-hopper-bug-2015-7
[2] Matt Miller, 2019, Trends, Challenges, and Strategic Shifts in the Software Vulnerability Mitigation Landscape. https://www.youtube.com/watch?v=PjbGojjnBZQ
[3] Robert NM Watson et al., 2015, Cheri: A hybrid capability-system architecture for scalable software compartmentalization. In IEEE Symposium on Security and Privacy
[4] Oleksii Oleksenko et al., 2018, Intel MPX Explained: A Cross-Layer Analysis of the Intel MPX System Stack
[5] Y. Kim, J. Lee and H. Kim, 2020, Hardware-based Always-On Heap Memory Safety
[6] Kostya Serebryany, 2019, ARM Memory Tagging Extension and How It Improves C/C++ Memory Safety

Root-cause of Memory-Safety Violations

Execution pipeline does not have full context about the instruction

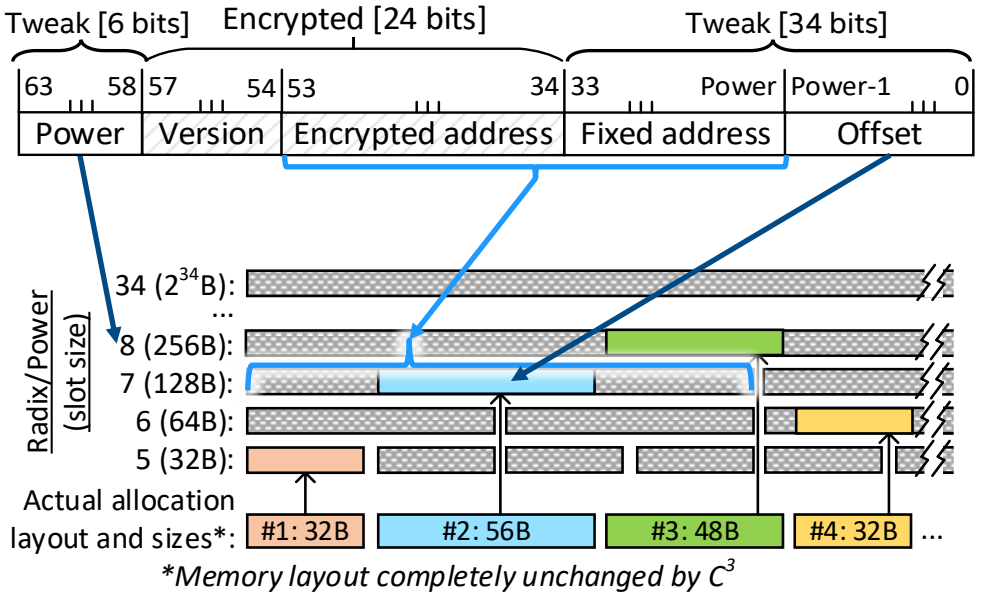
Mitigation Approaches:

- Add context – Metadata based traditional techniques
- Provide a cryptographic barrier to exploit the vulnerability – emerging!!

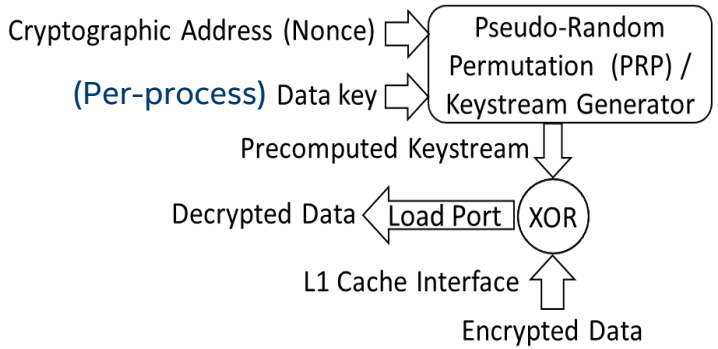
Cryptographic Capability Computing (C^3)

- ☑ Stateless, no metadata, no memory layout changes, negligible overheads
- ☑ Addresses a wide variety of memory safety vulnerabilities

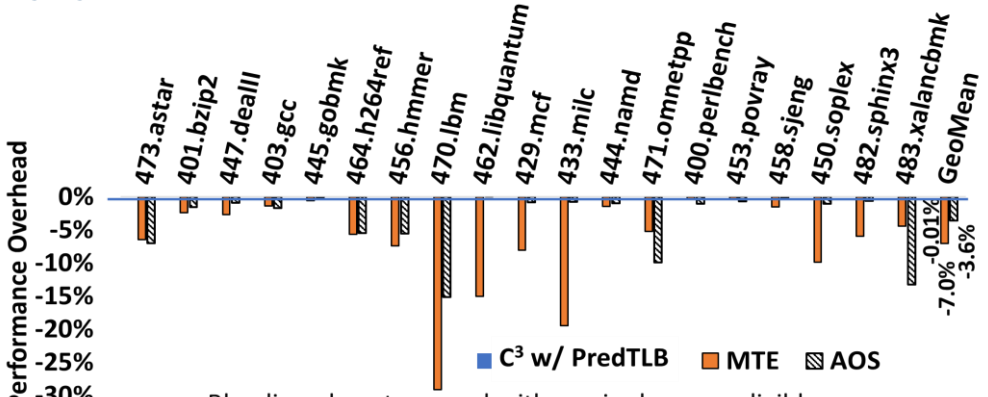
Unique, unforgeable (within cryptographic bounds) Cryptographic Address for each allocation:



Unique Cryptographic Address ⇒ Cryptographically isolated allocations



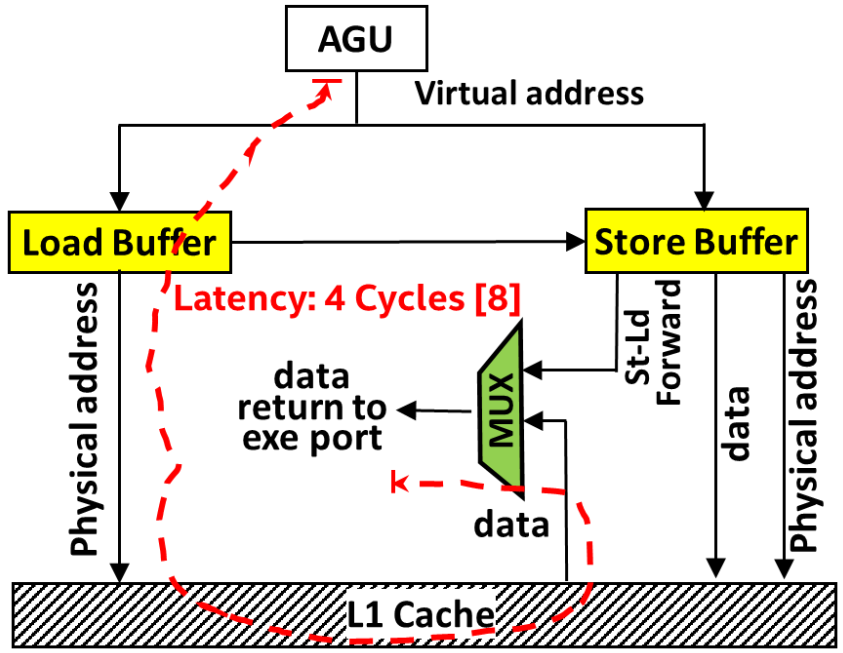
Negligible performance overhead



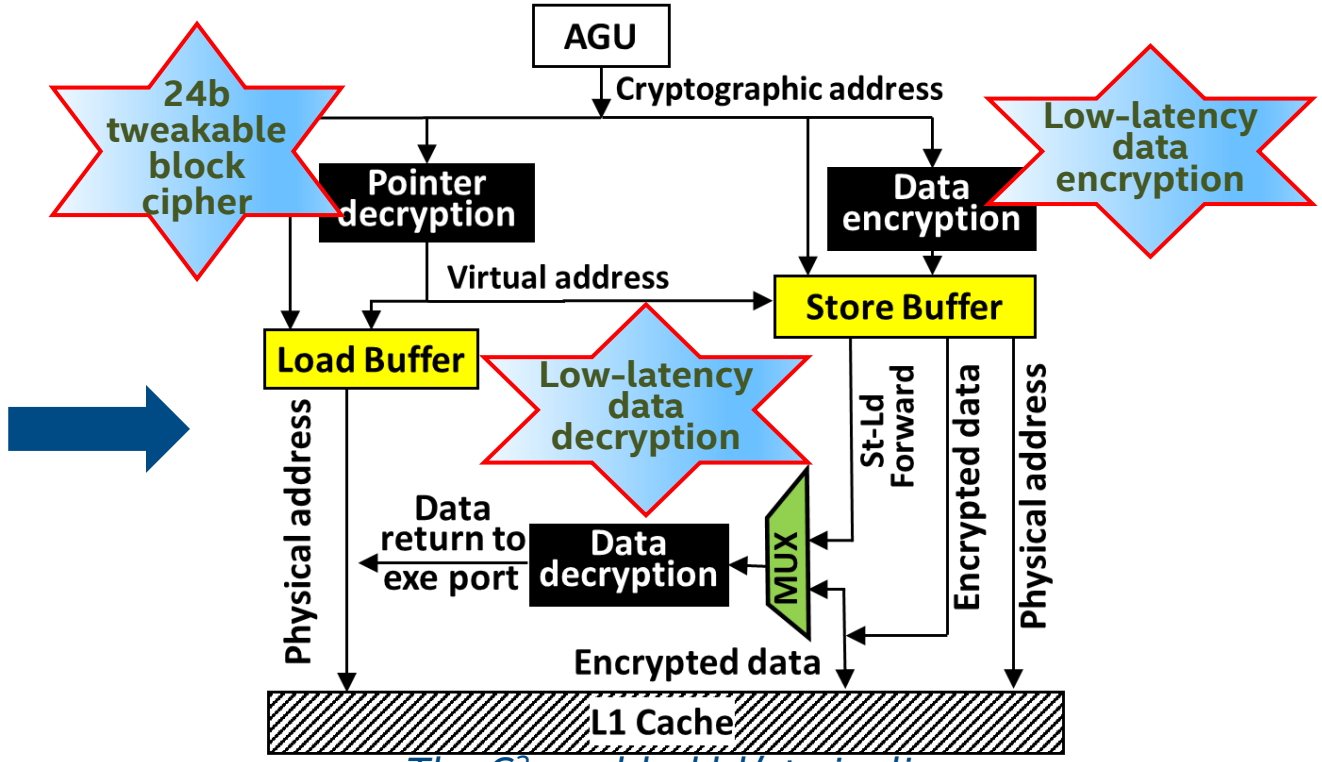
Blue line almost merged with x-axis shows negligible C^3 w/ PredTLB overheads for all workloads

[7] Michael LeMay, Joydeep Rakshit, Sergej Deutsch, David M. Durham, Santosh Ghosh, Anant Nori, Jayesh Gaur, Andrew Weiler, Salmin Sultana, Karanvir Grewal, and Sreenivas Subramoney. *Cryptographic capability computing*. In MICRO '21: 54th Annual IEEE/ACM International Symposium on Microarchitecture, 2021

Cryptographic Challenges for C^3



Existing ld/st pipeline



The C^3 enabled ld/st pipeline

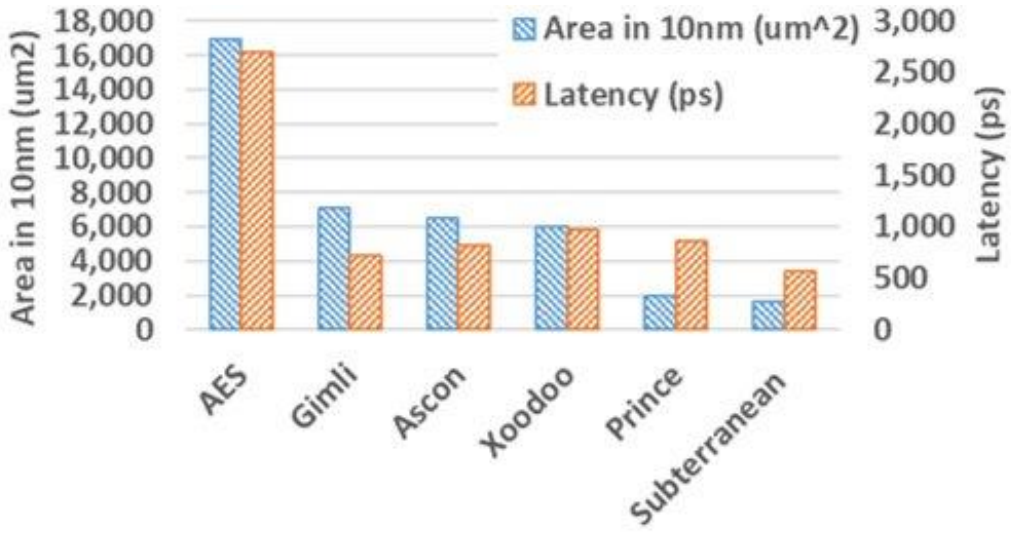
- Low-latency (2-3 cycles @4GHz) tweakable blockcipher with small block size – No candidate so far!
- Data decryption with 3 cycles @4GHz latency – Existing NIST standards require >5x!
- Area overhead impacts energy budgets

[8] https://www.anandtech.com/show/11544/intel-skylake-ep-vs-amd-epyc-7000-cpu-battle-of-the-decade/13

Analysis of NIST LWC Primitives and Beyond

NIST LWC Finalists	Underlying Primitive	State Size	Critical Path of the Premitive		
			Optimal Logic Levels /Round	# Rounds	Total Logic Levels / Primitive
ASCON	ASCON	320	5	12	60
Elephant	Spongent, Keccak[200]	160, 200	5, 7	80, 18	400, 126
GIFT-COFB	GIFT	128	9	40	360
Grain-128AEAD	Grain-128a	256	6	128	768
ISAP	ASCON, Keccak	320, 400	5,7	25, 48	125, 336
PHOTON-Beetle	PHOTON256	256	8	12	96
Romulus	Skinny	128	12	40	480
SPARKLE (SCHWAE MM and ESCH)	Sparkle	256, 384, 512	26	10, 11, 12	260 and more
TinyJambu	NFSR	128	3	1024*	96
Xoodyak	Xoodoo	384	5	12	60
AES	AES	256	20	10	200

Estimated latency of the primitives used in the finalist algorithms



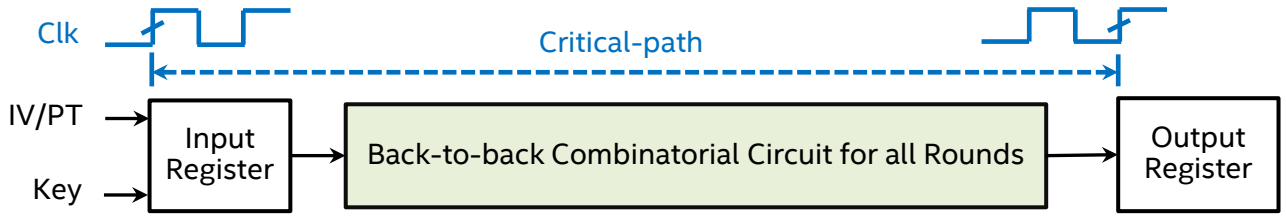
Latency and Area for the most promising lightweight primitive's vs AES

- Multiple primitives in the finalist algorithms have longer latency than AES-128
- ASCON and Xoodoo provide attractive latency and area – can fit within 4-5 cycles @4GHz
- Subterranean provides the lowest latency with affordable area overhead for C^3

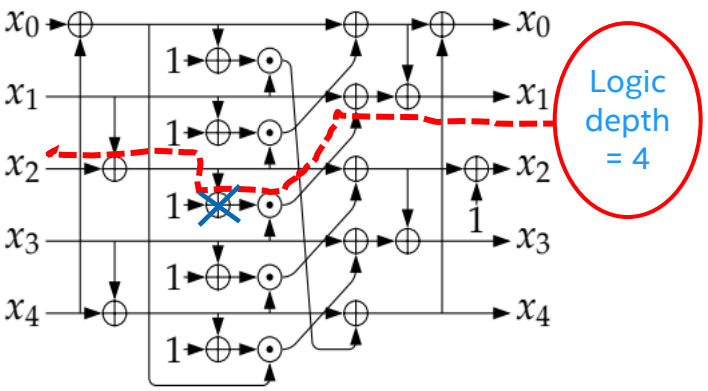
Latency and Area of a Primitive

Estimation must be integrated from the early stage in cipher design

- Number of two input logic depths
  - Latency of a uniform-round based primitive = number-of-rounds × latency-of-a-single-round



Latency measurement of a primitive



Counting gates in the longest path

- Smaller internal chunks vs larger
  - Area complexity of 4b S-box vs 8b: $\frac{256}{4}2^4$ vs $\frac{256}{8}2^8$
- Balance between complexity of a single round and number of rounds

Open Problems and Updates

- Small block-size tweakable blockcipher with very low-latency
- Low-latency low-area data encryption/decryption

- Intel launched Crypto Frontiers Research Center

Questions …