black hat BRIEFINGS
DECEMBER 10-11, 2025
EXCEL LONDON / UNITED KINGDOM

Fifty Dollars to Root the Cloud

Low-Cost Memory Interposer Attacks on Confidential Computing

Jesse De Meulemeester & Jo Van Bulck

Collaboration with David Oswald, Luca Wilke, Thomas Eisenbarth, Ingrid Verbauwhede

Cybersecurity 101 / Cloud Security / Cloud Security Breaches

Top 10 Cloud Security Breaches in 2024

Updated

The biggest data breach fines, penalties, and settlements so far

Feature

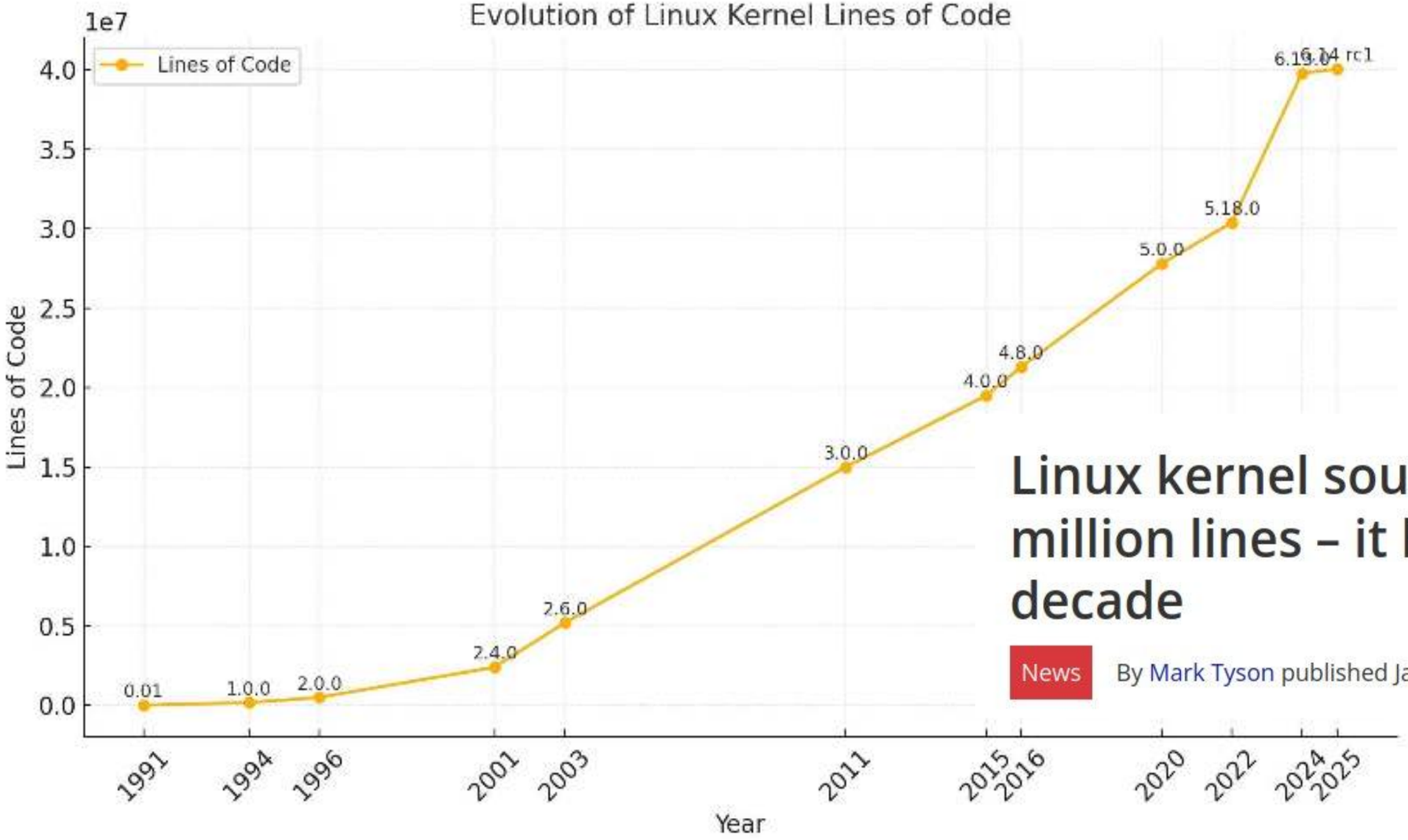
Jan 8, 2025 • 17 mins

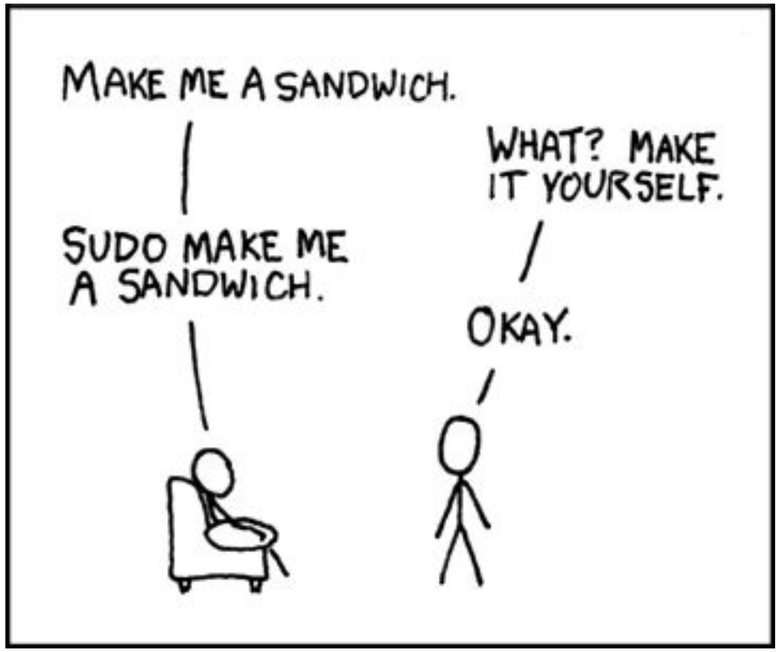
Security Spotlight, News

Oracle Cloud Breach Compromises 6 Million Records, Threatening 140,000 Businesses

https://itsfoss.gitlab.io/post/linux-kernel-source-code-surpasses-40-million-lines/

Root Cause: Privileged Software Complexity?

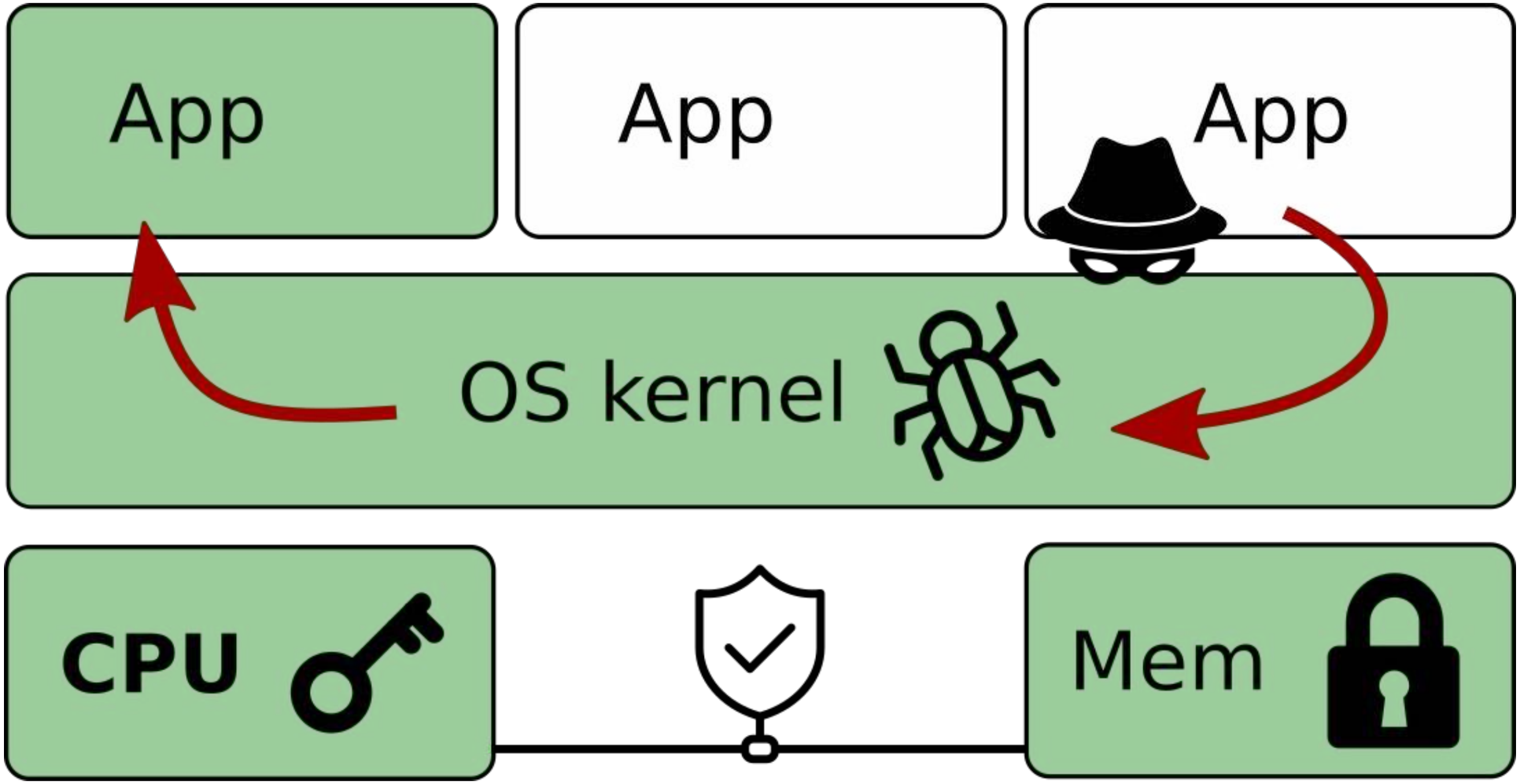




Linux kernel source expands beyond 40 million lines – it has doubled in size in a decade

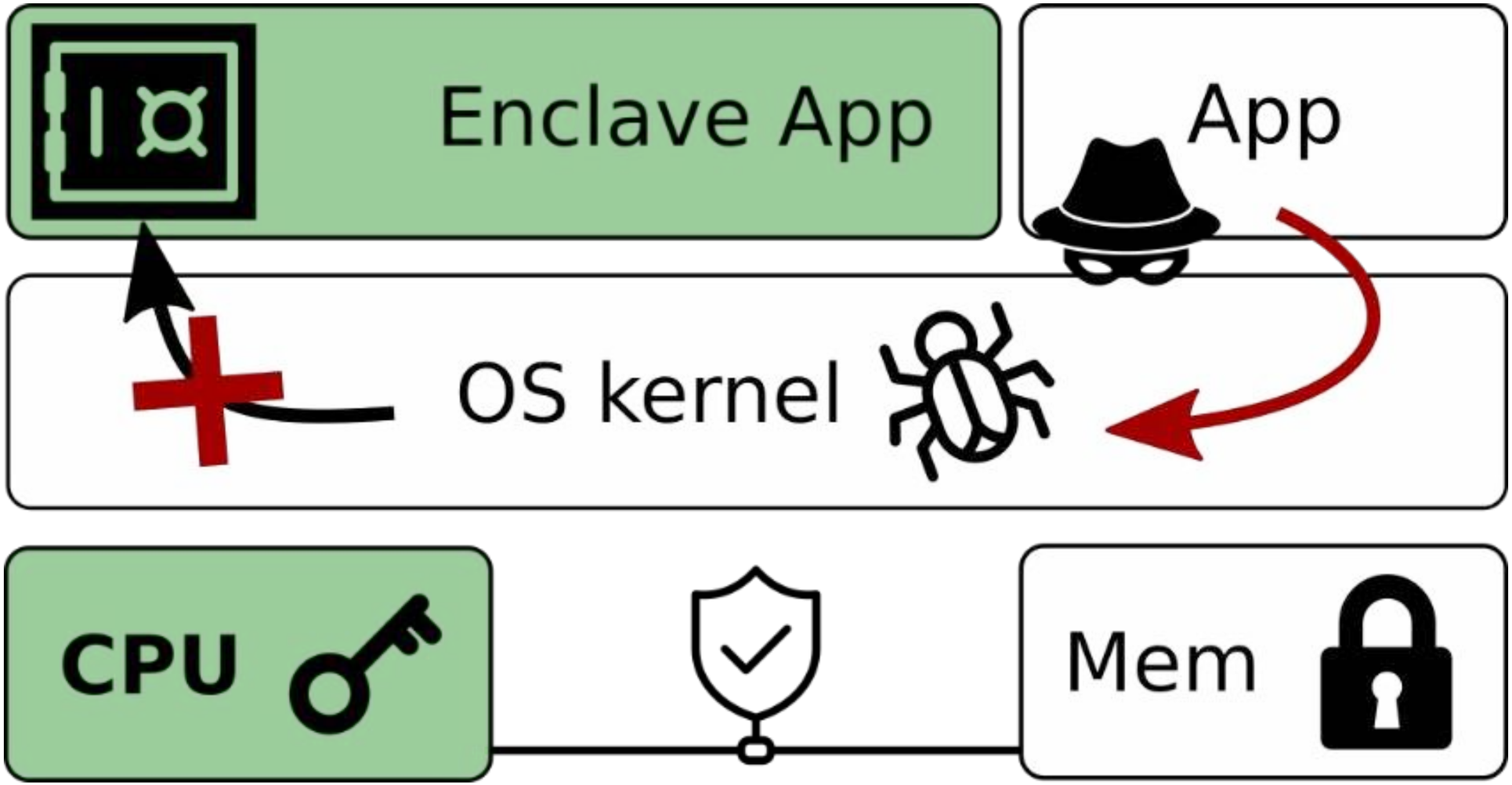
News By Mark Tyson published January 26, 2025

The Case for Confidential Computing



Traditional **layered designs:** Large trusted computing base

The Case for Confidential Computing



Trusted execution: Hardware-level isolation and attestation

The Rise of Confidential Computing CPU Technology

- 2004: ARM TrustZone
- 2015: **Intel Software Guard Extensions (SGX)**
- 2016: AMD Secure Encrypted Virtualization (SEV)
- 2018: IBM Protected Execution Facility (PEF)
- 2020: AMD SEV with Secure Nested Paging (SEV-SNP)
- 2022: Intel Trust Domain Extensions (TDX)
- 2023: ARM Confidential Compute Architecture (CCA)
- 2024: NVIDIA Confidential Computing

TEEs are here to stay...

The Rise of Confidential Computing CPU Technology

DARKREADING

Cloud Providers Throw Their Weight Behind Confidential Computing

New technologies designed into processors allow enterprises to leverage cloud advantages while meeting privacy regulations.

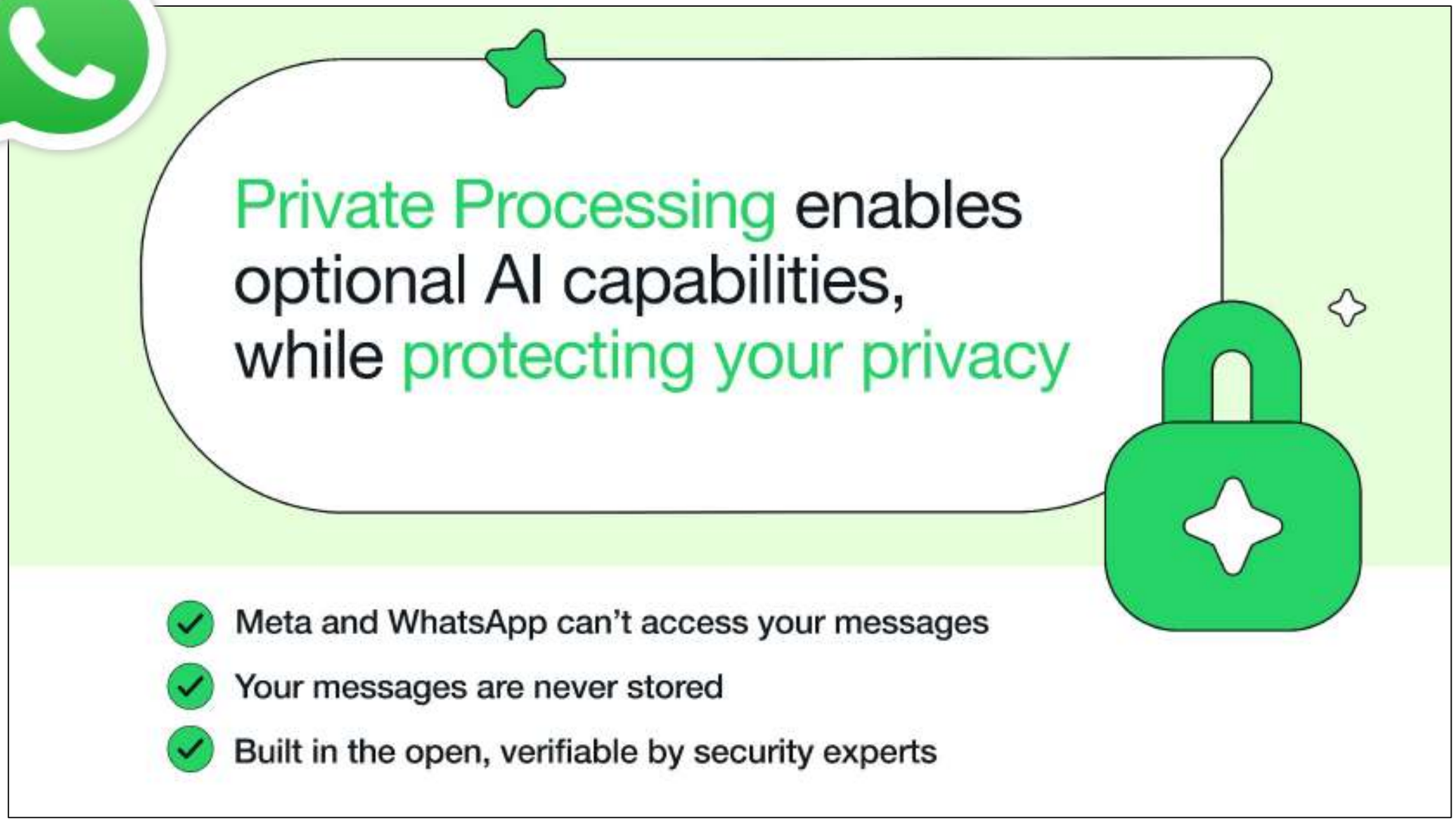
https://www.darkreading.com/cloud-security/cloud-providers-are-putting-more-weight-in-confidential-computing

Confidential Computing Today, Computing Tomorrow

In the near future,
“confidential computing”
*will just be “computing.”**

* **Mark Russinovich,** **CTO**
Microsoft Azure

Example: WhatsApp Private Processing Confidential AI



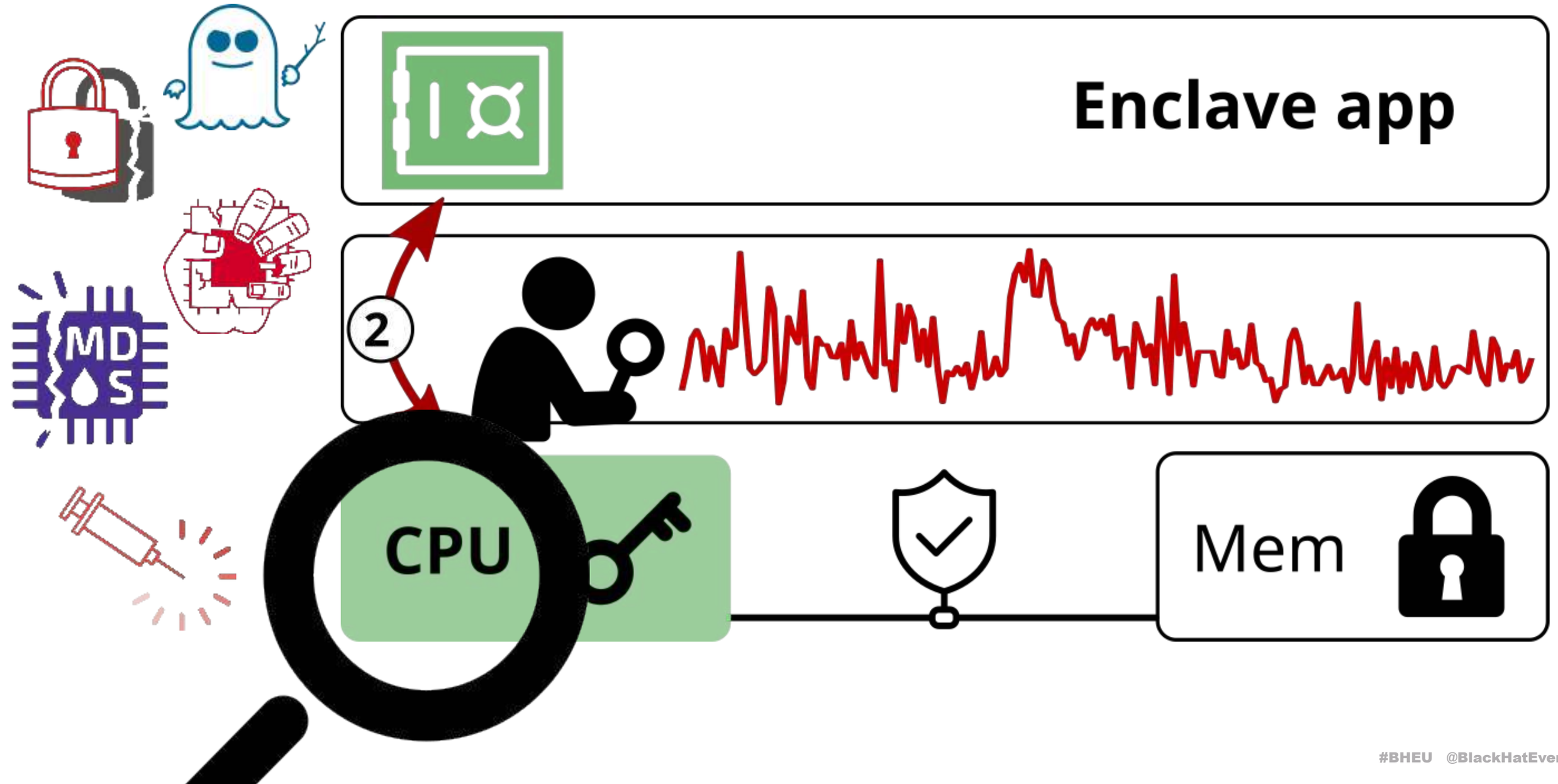
https://engineering.fb.com/2025/04/29/security/whatsapp-private-processing-ai-tools/

Confidential Computing: The Weakest Link?

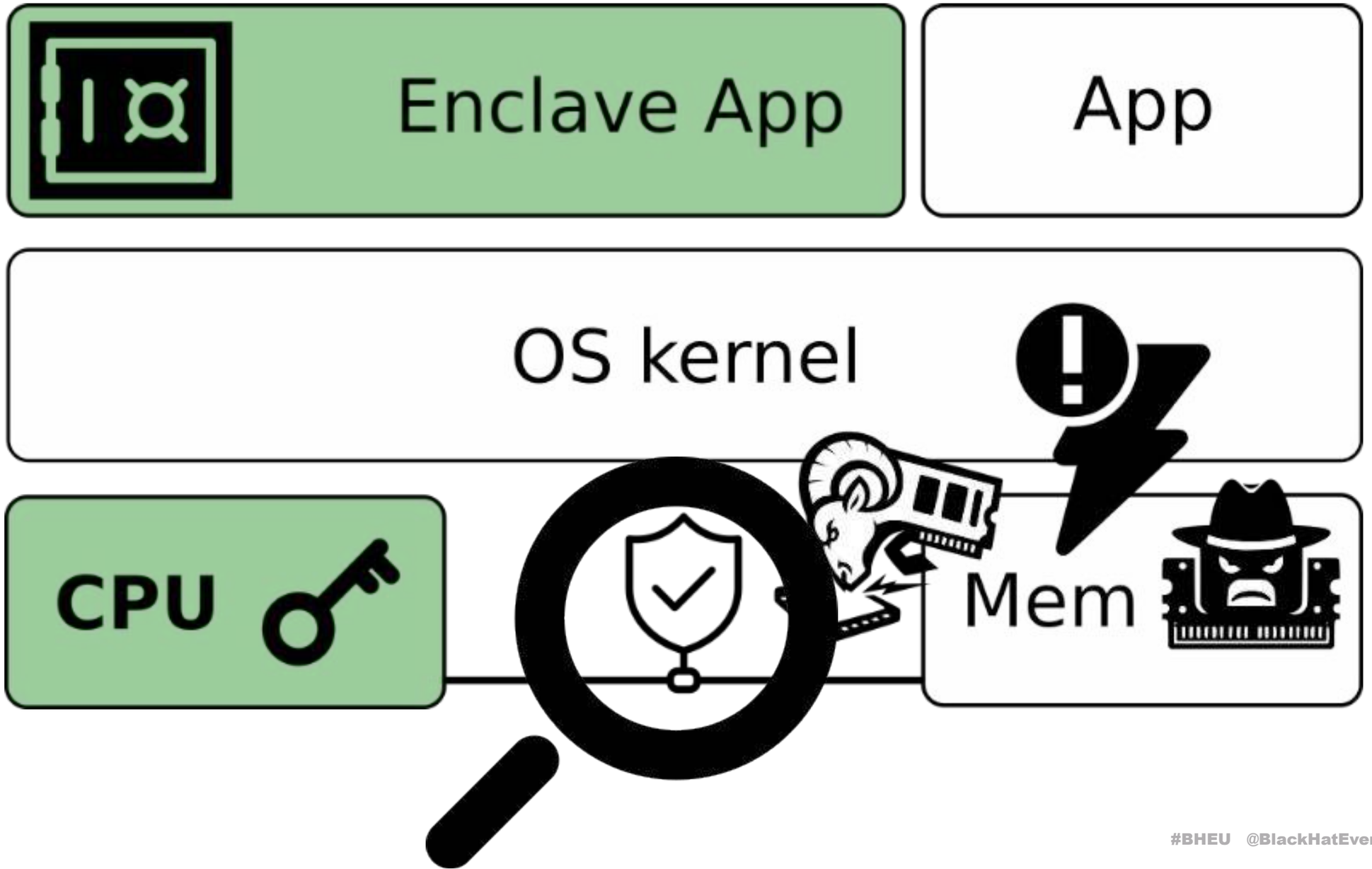
Recap: Hardware-Based Confidential Computing

Confidential computing = CPU isolation + memory encryption

Leaky Processors: SW-Based Microarchitectural Attacks



Today: Hardware Attacks on Encrypted Memory

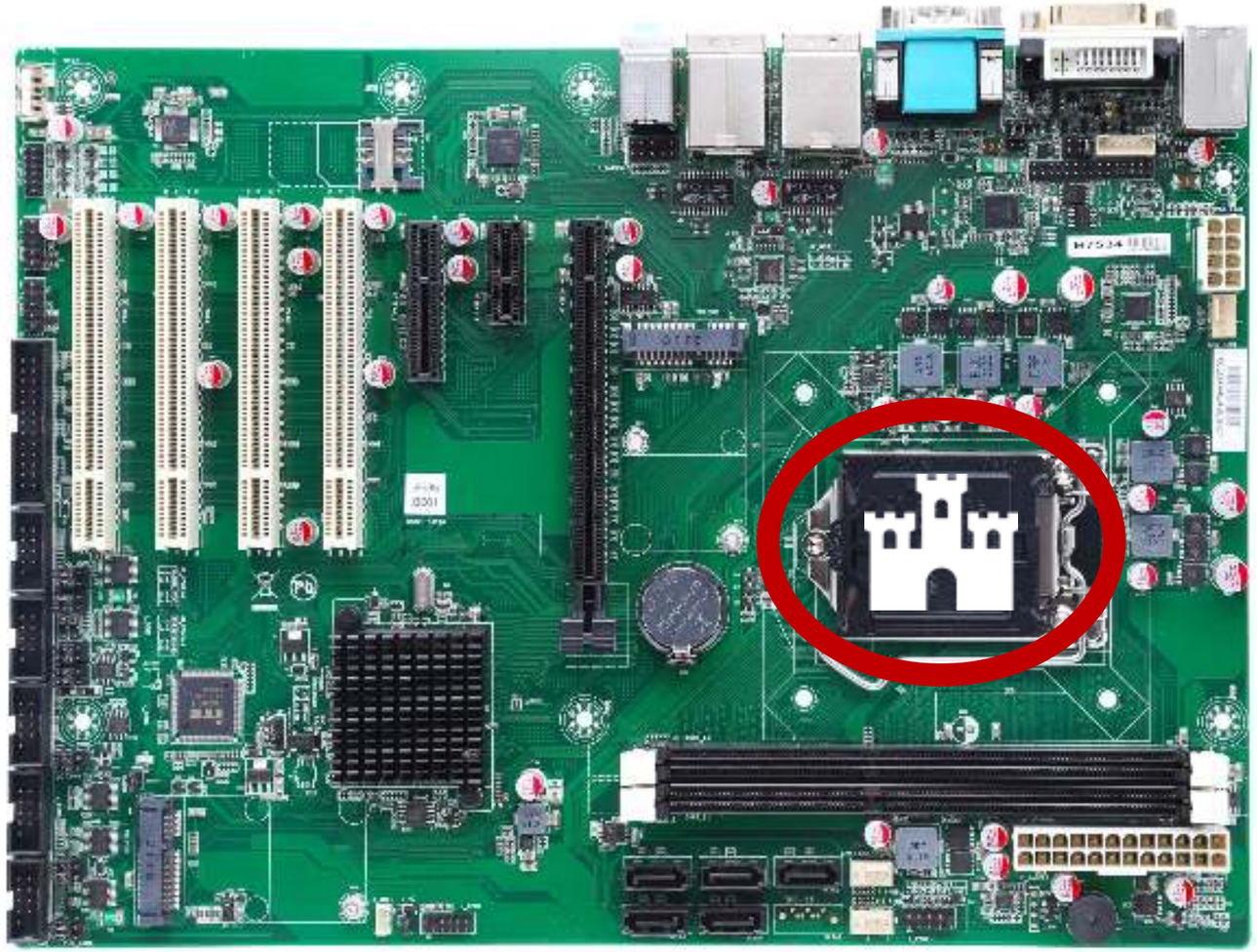


Road Map

1. **Modern memory encryption designs and where to find them**
2. BadRAM: What if your DRAM lies to you?
3. Battering Ram: Low-cost physical interposer attacks
4. Conclusions and takeaways

Confidential Computing: Trust Boundary

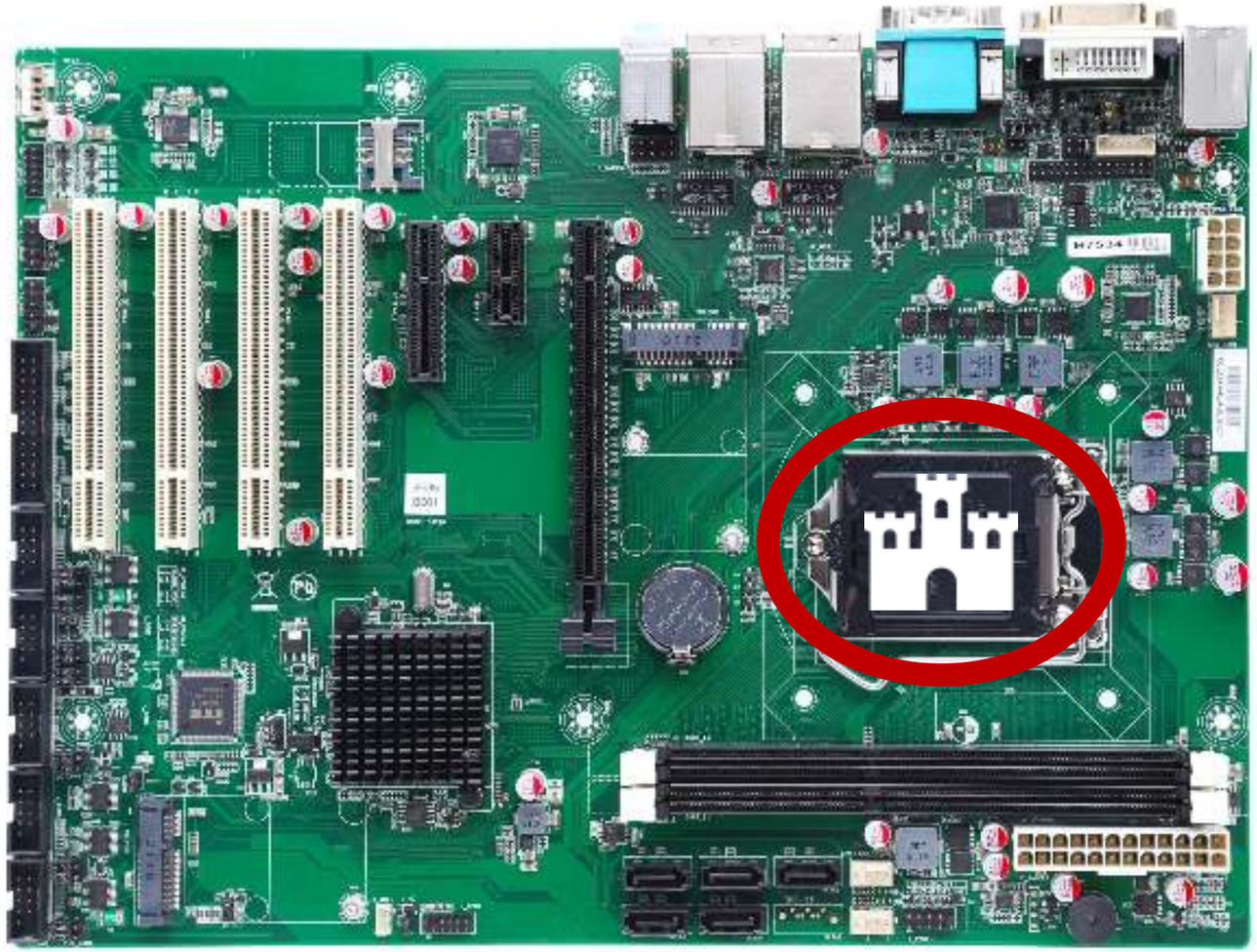
- **CPU package** = trust boundary



Confidential Computing: Trust Boundary

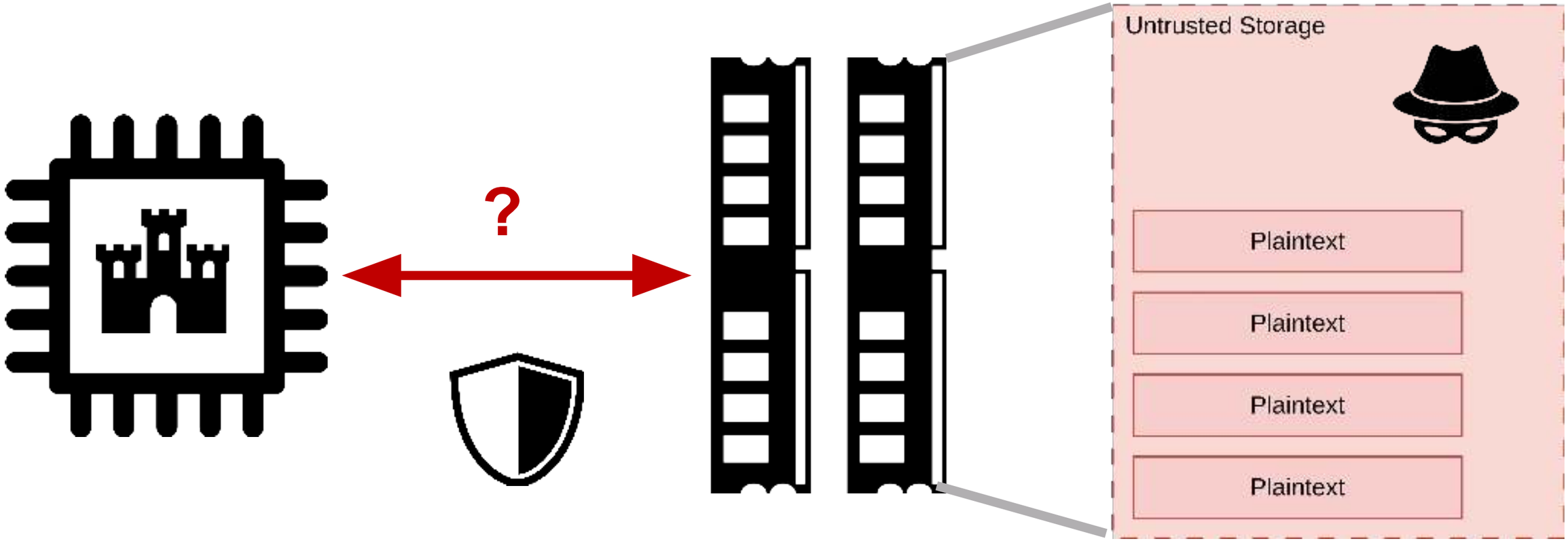
- **CPU package** = trust boundary
- **Memory encryption** to protect against <u>physical access:</u>

1. Rogue cloud provider employees
2. Supply-chain adversaries
3. Local law enforcement



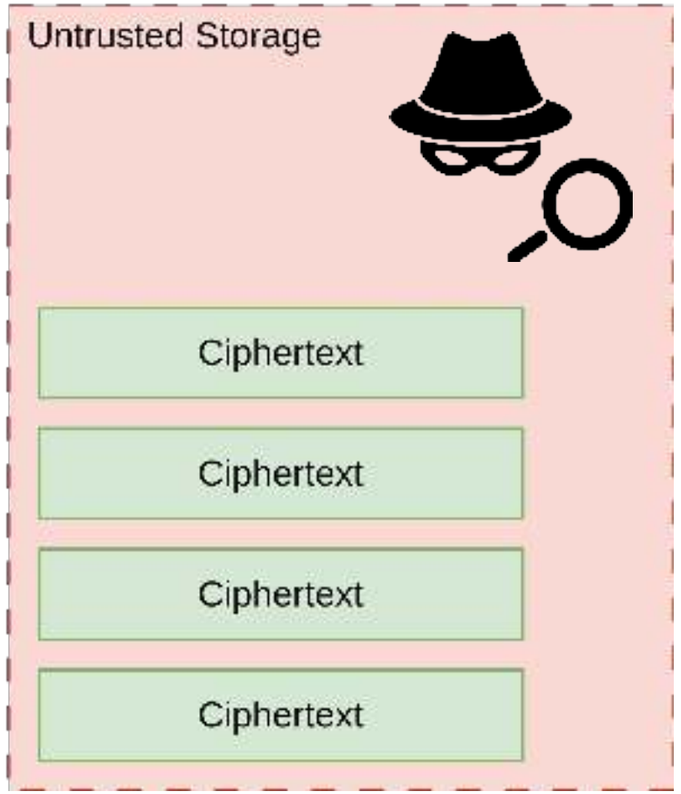
Memory Encryption

Idea: Treat attacker-controlled DRAM as untrusted storage



Memory Encryption: Confidentiality

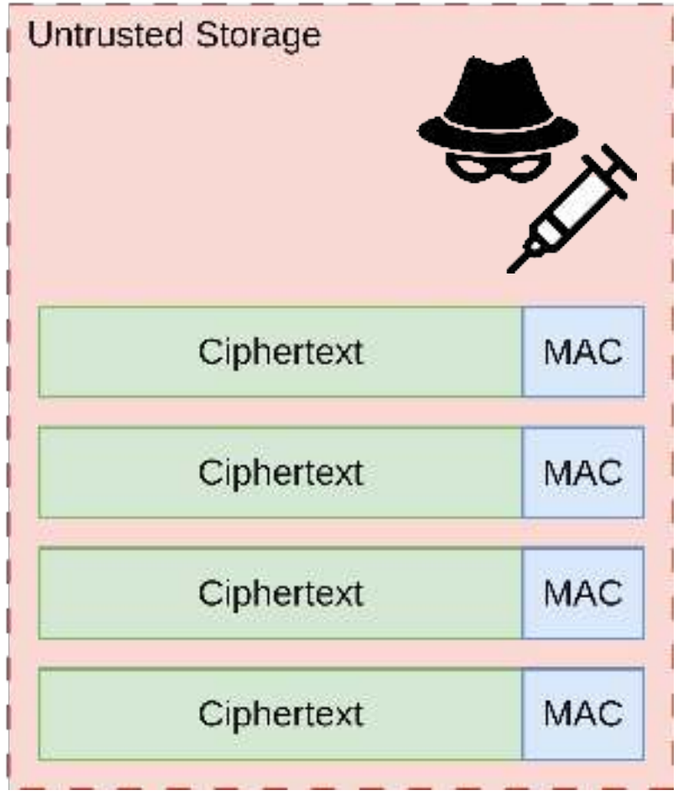
Confidentiality



Memory Encryption: Confidentiality + Integrity

Confidentiality

Integrity

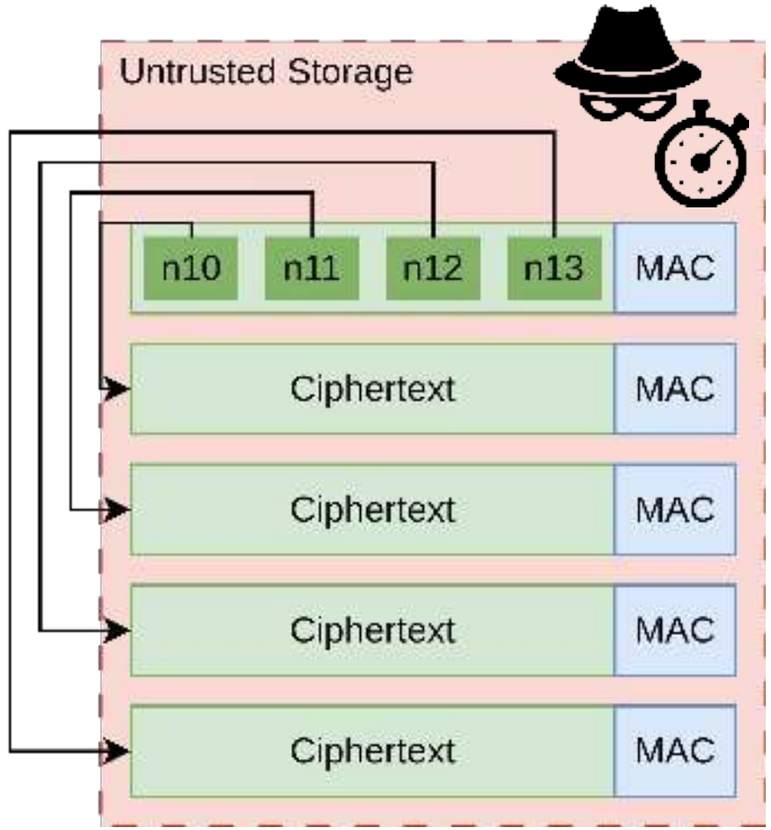


Memory Encryption: Conf. + Integrity + Freshness

Confidentiality

Integrity

Freshness

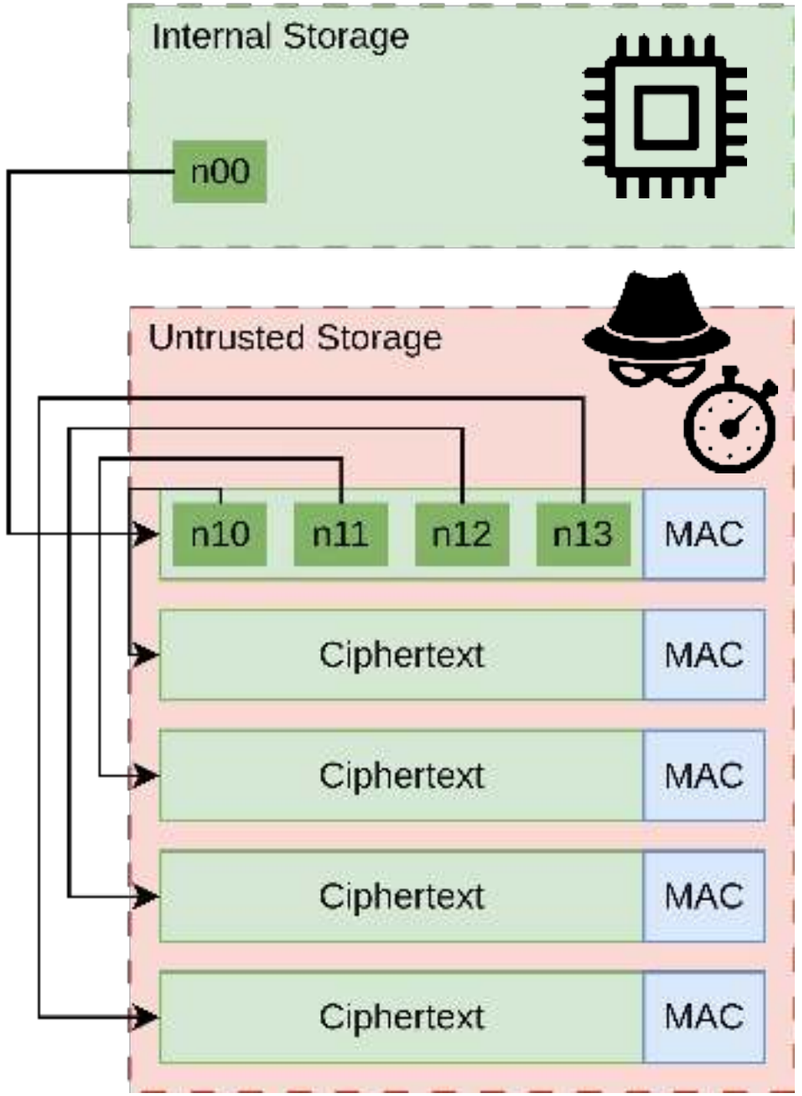


Memory Encryption: Conf. + Integrity + Freshness

Confidentiality

Integrity

Freshness



A Brief History of Commercial Memory Encryption

intel

SGX

2015

Confidentiality

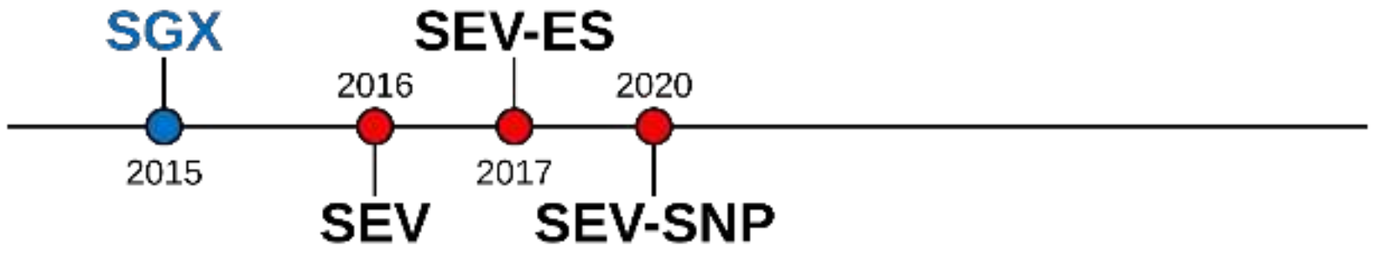
Integrity

Freshness

Scalable

A Brief History of Commercial Memory Encryption

intel



AMD

Confidentiality	🔒	🔒	🔒	🔒
Integrity	🛡️	✗	✗	✗
Freshness	🔄	✗	✗	✗
Scalable	✗	↗	↗	↗
	🤩	🤔	🤔	🤔

A Brief History of Commercial Memory Encryption

CLOUD | OPERATIONS & MANAGEMENT | NEWS

Why Google Cloud Turned to AMD to Solve for Runtime Encryption

AMD's latest server chips enabled better scalability, less lag, and more memory than Intel SGX, the cloud provider said.

Maria Korolov
July 21, 2020

5 Min Read

https://www.datacenterknowledge.com/cloud/why-google-cloud-turned-to-amd-to-solve-for-runtime-encryption

A Brief History of Commercial Memory Encryption

PUTTING ON A BRAVE FACE

Intel promises Full Memory Encryption in upcoming CPUs

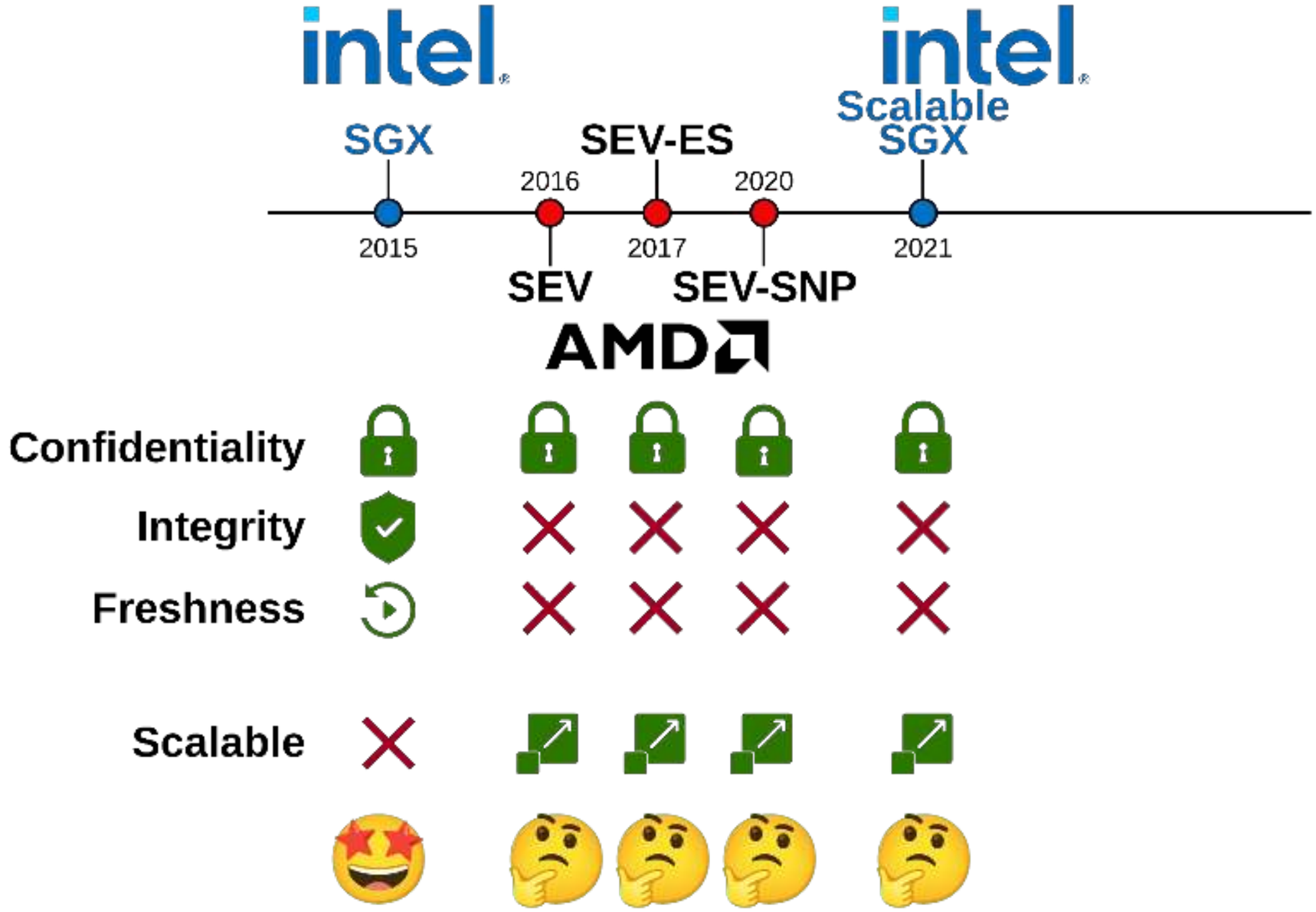
Intel's security plans sound a lot like "we're going to catch up to AMD."

JIM SALTER – FEB 26, 2020 8:29 PM | 120

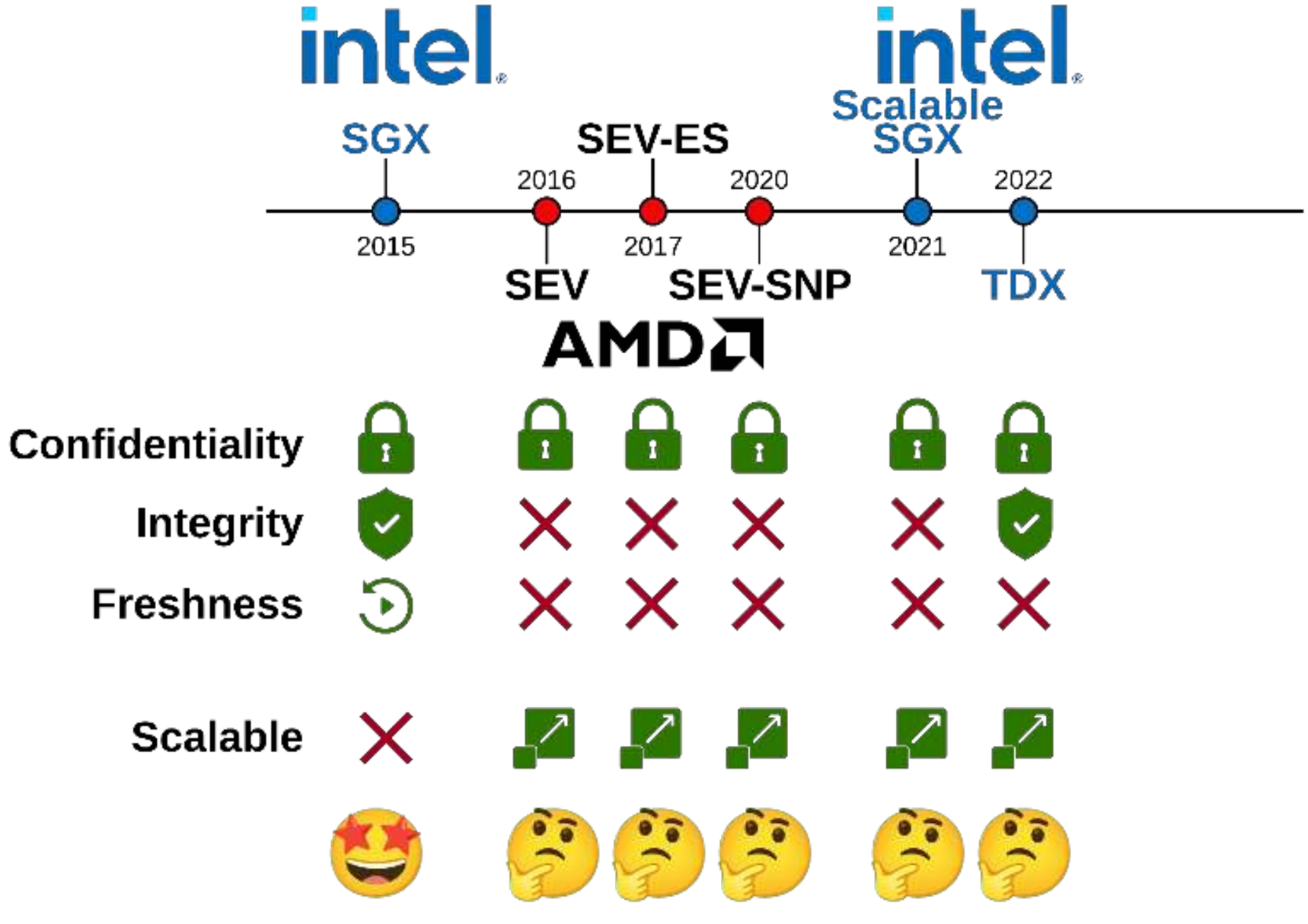
-> Intel Security Architecture and Technology Director John Sell provided an overview of Intel's mission to provide common security capabilities across all architectures. Credit: Intel Corporation

https://arstechnica.com/gadgets/2020/02/intel-promises-full-memory-encryption-in-upcoming-cpus/

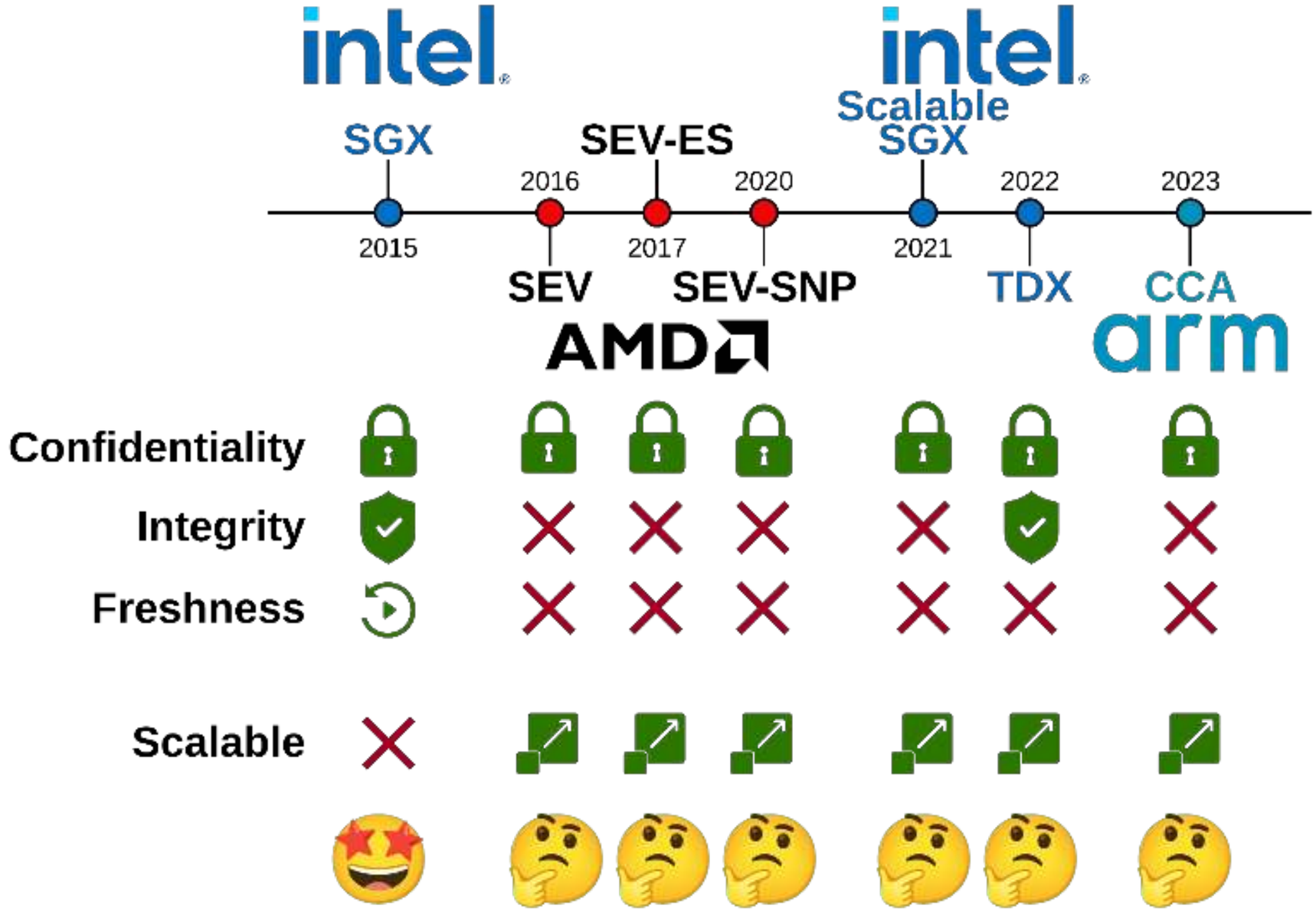
A Brief History of Commercial Memory Encryption



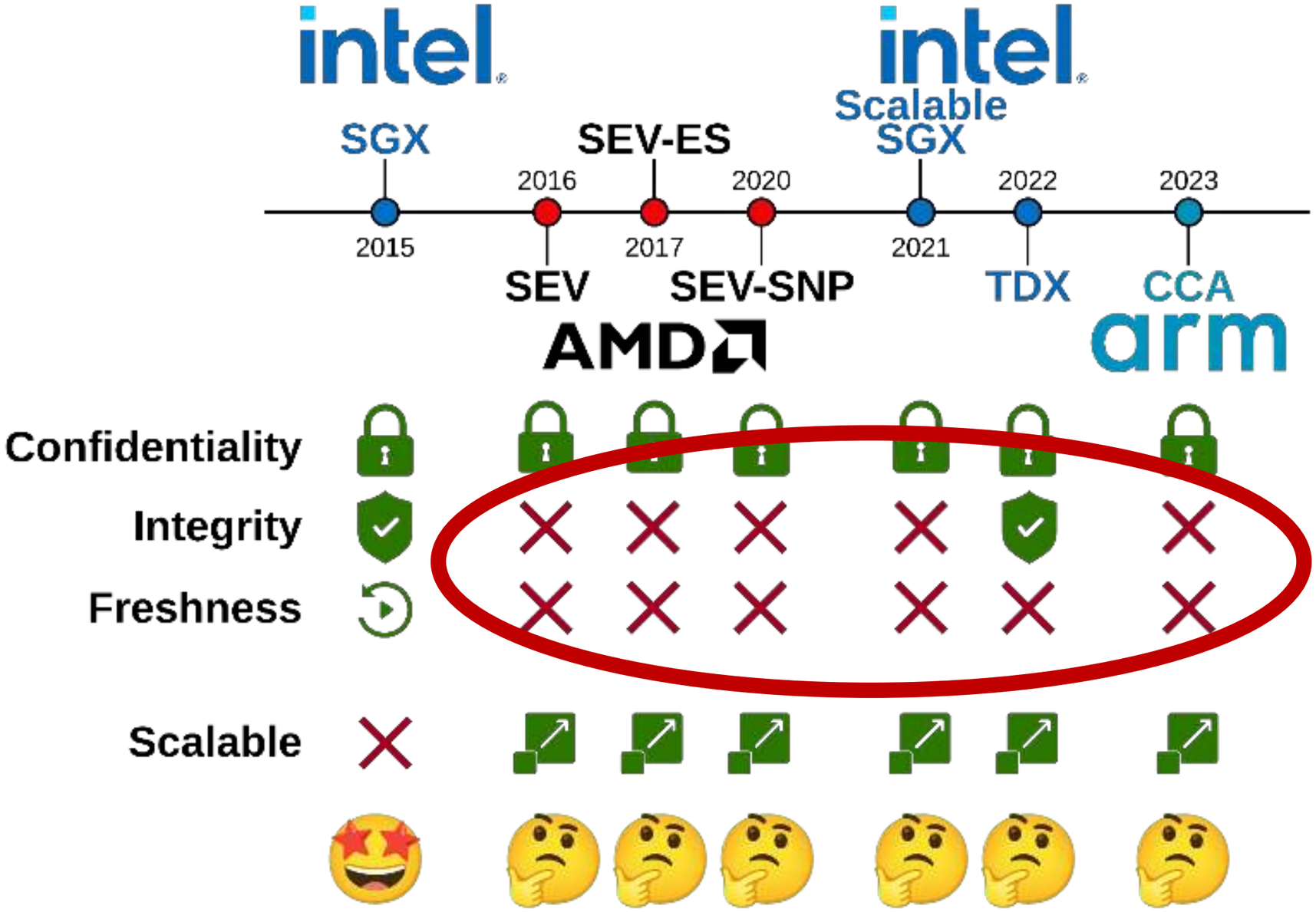
A Brief History of Commercial Memory Encryption

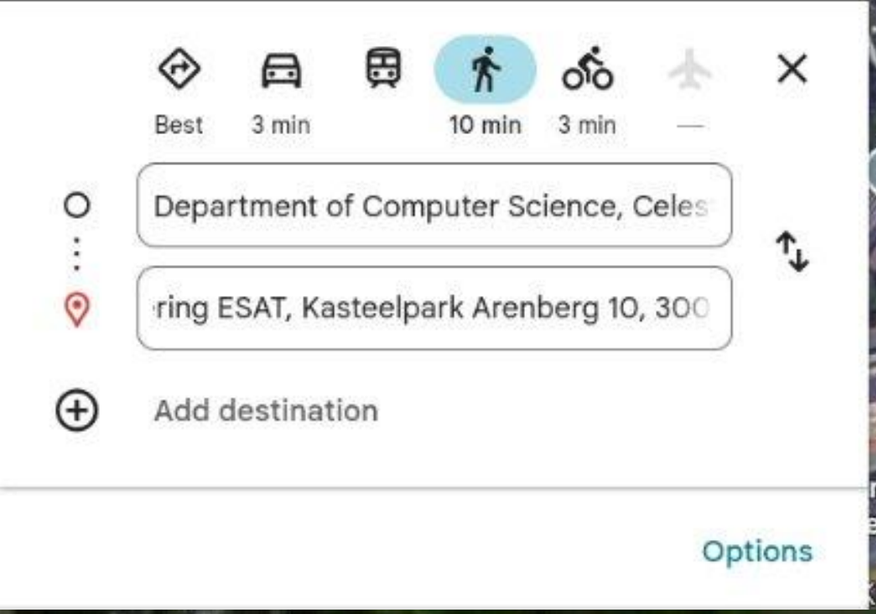


A Brief History of Commercial Memory Encryption



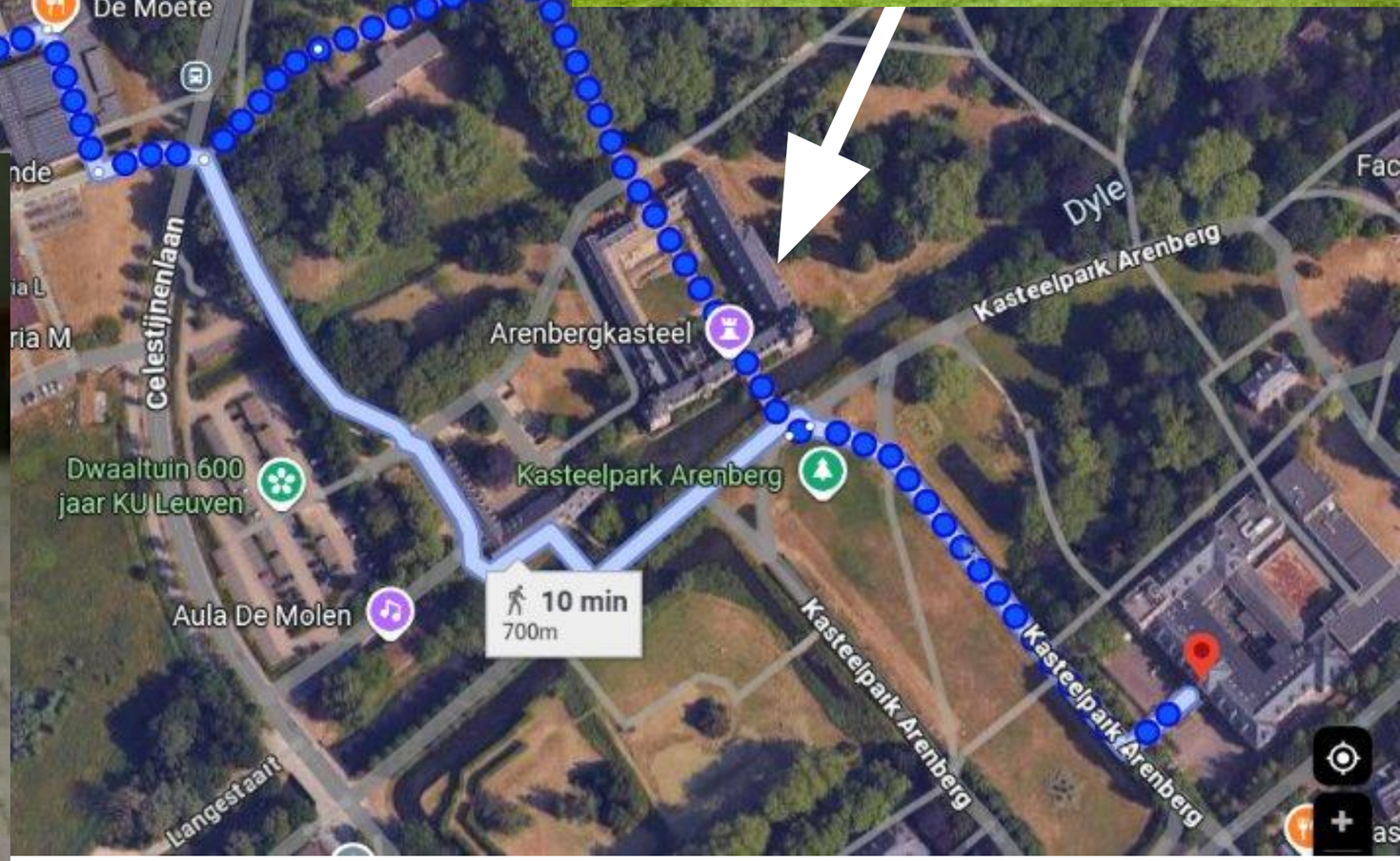
A Brief History of Commercial Memory Encryption




Best
3 min
10 min
3 min
Department of Computer Science, Celes
ring ESAT, Kasteelpark Arenberg 10, 300
Add destination
Options


Search along the route...
Restaurants
Kapeldreef
Kapeldreef
Celestijnenl
De Moete
KU Leuven

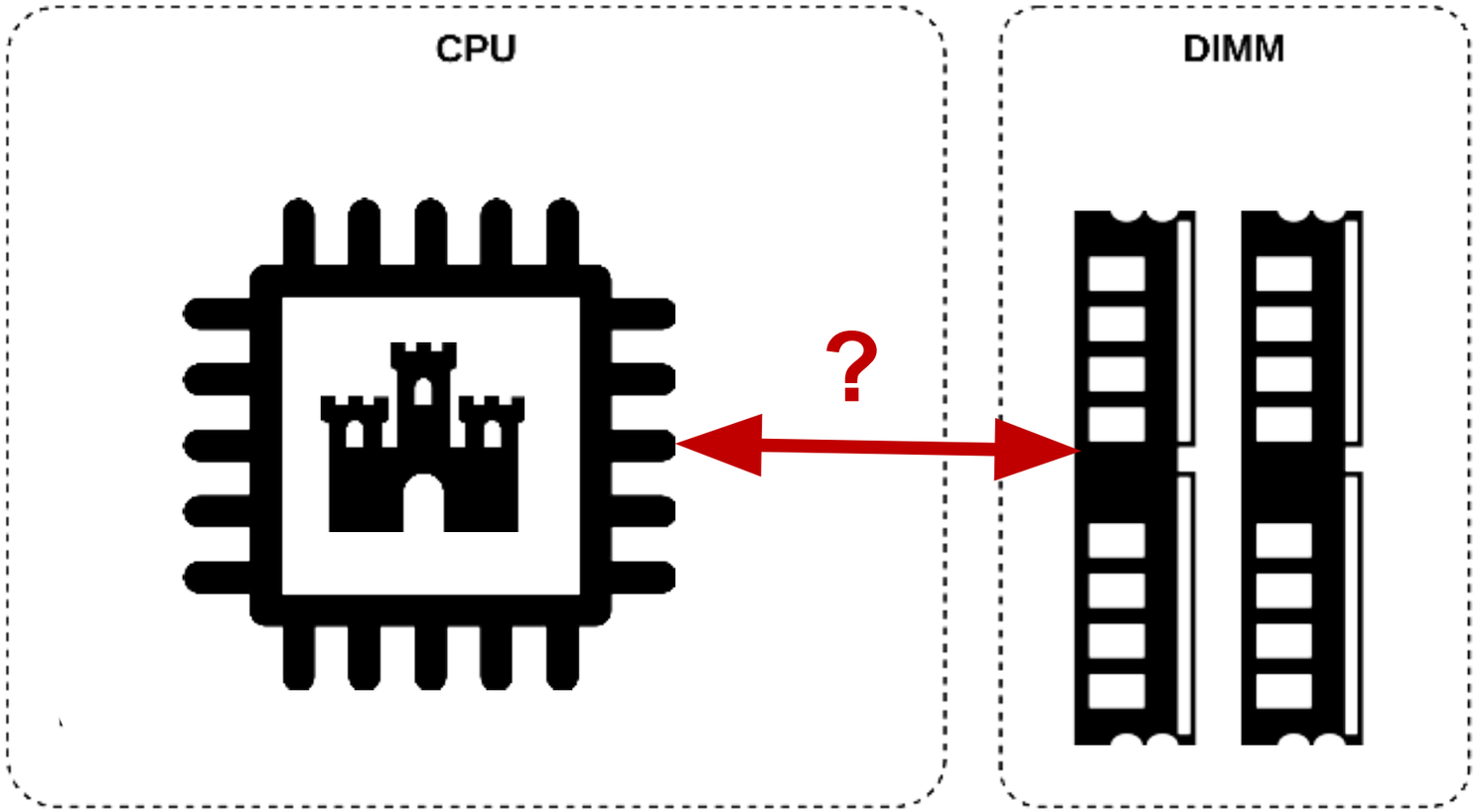

I HAVE...
AN EVIL PLAN
quickmeme.com


Celestijnenlaan
Arenbergkasteel
Dyle
Kasteelpark Arenberg
Dwaaltuin 600 jaar KU Leuven
Kasteelpark Arenberg
Aula De Molen
10 min
700m
Kasteelpark Arenberg
Kasteelpark Arenberg
Langestaart

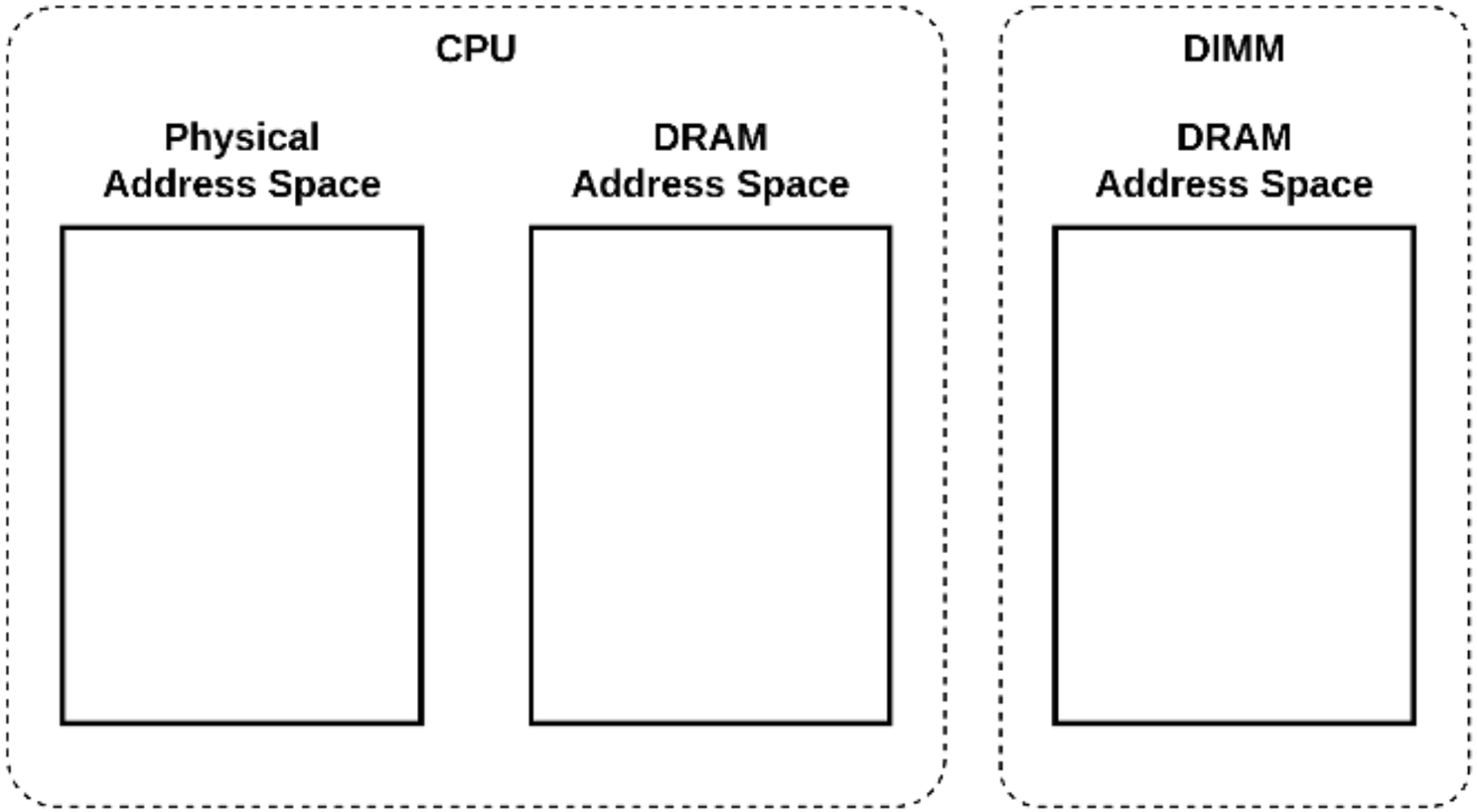
Road Map

1. Modern memory encryption designs and where to find them
2. **BadRAM: What if your DRAM lies to you?**
3. Battering Ram: Low-cost physical interposer attacks
4. Conclusions and takeaways

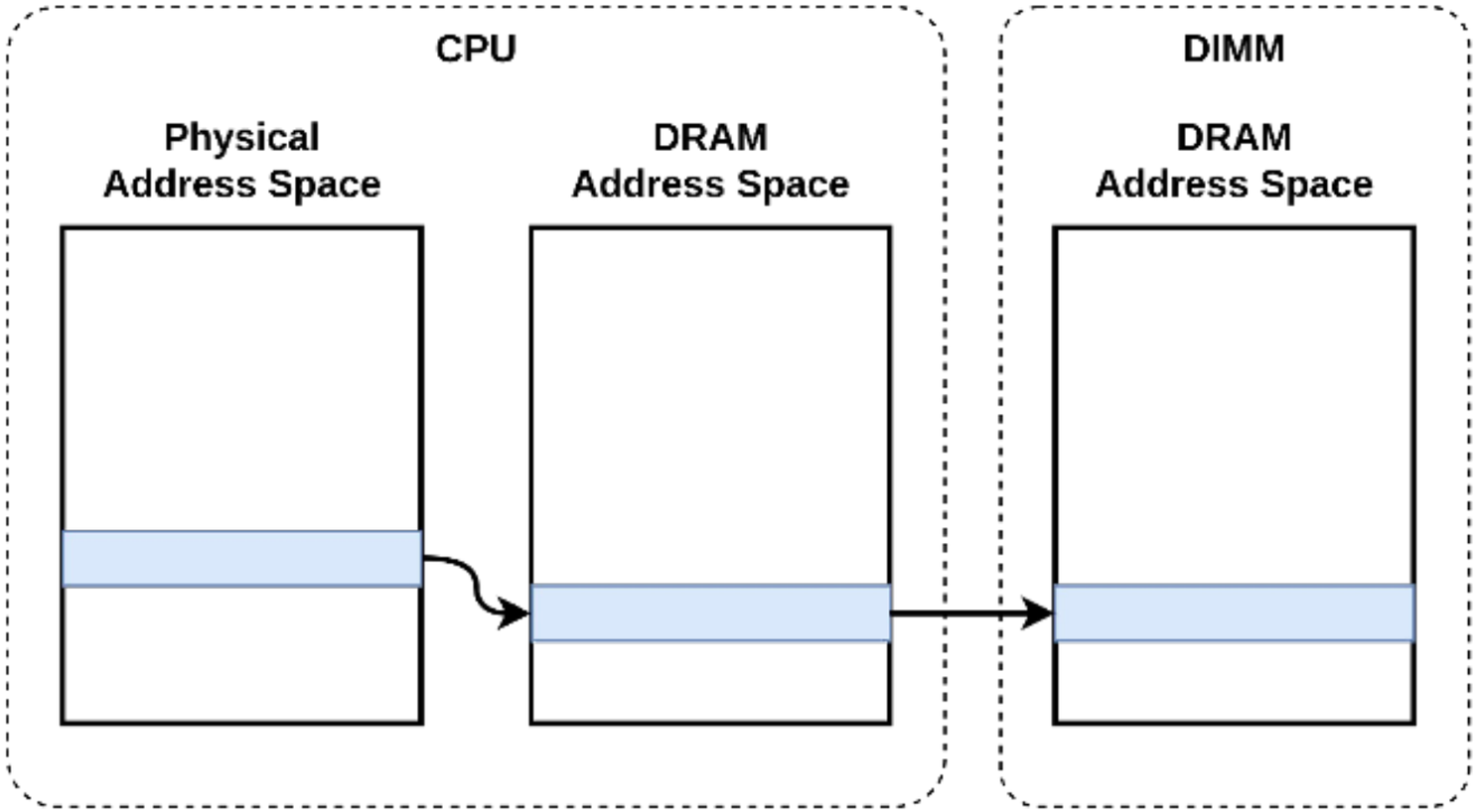
DRAM Addressing 101



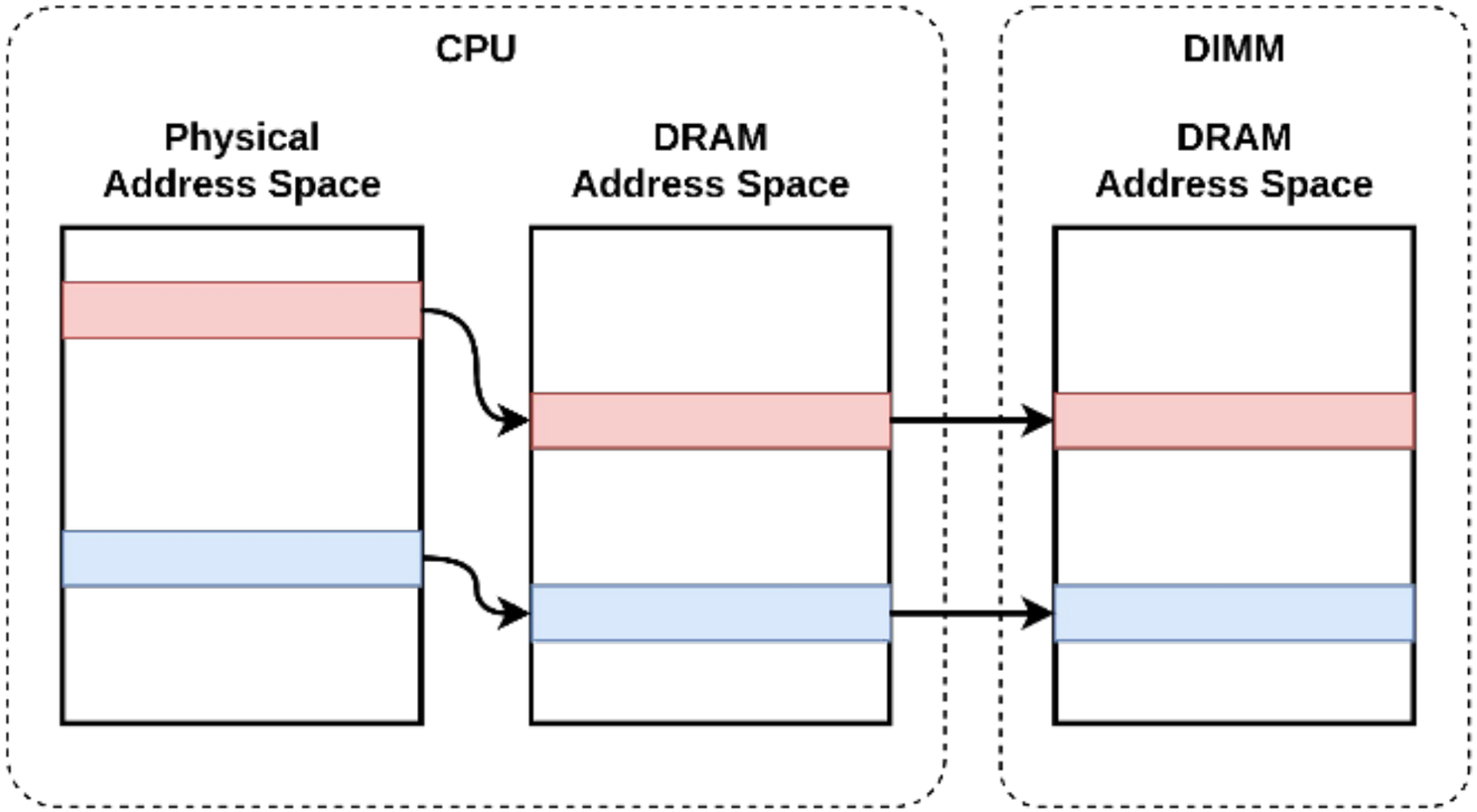
DRAM Addressing 101



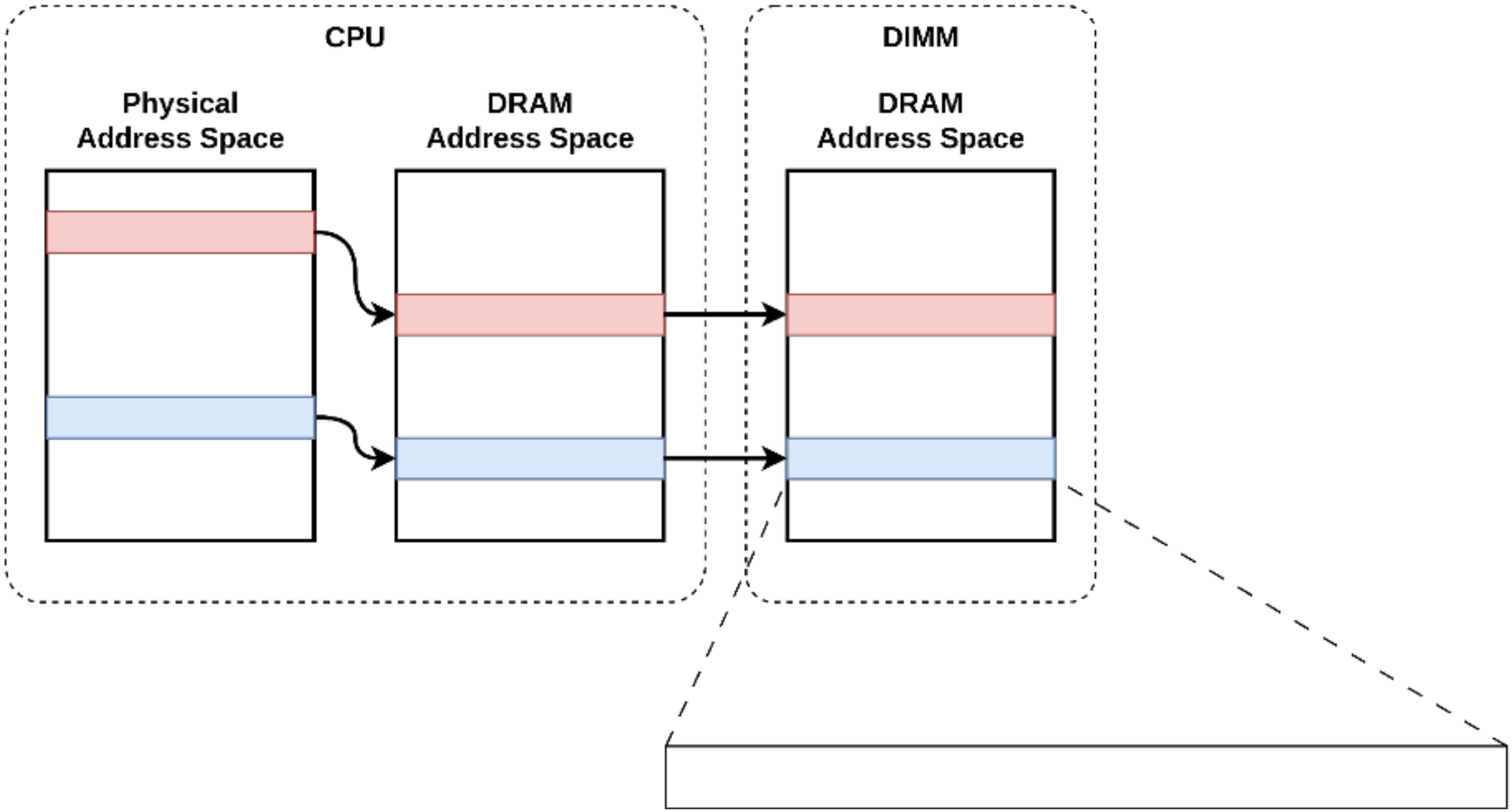
DRAM Addressing 101



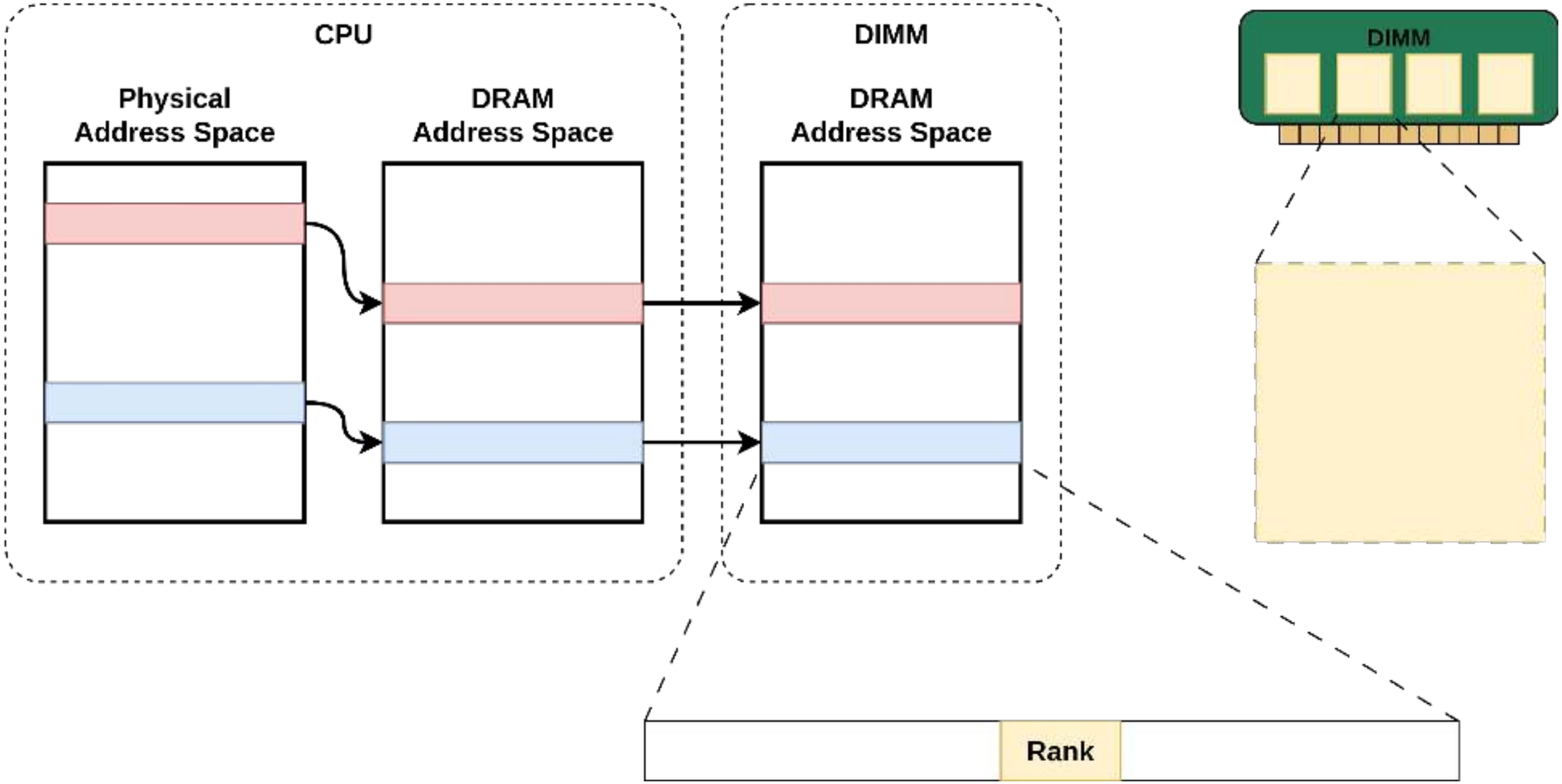
DRAM Addressing 101



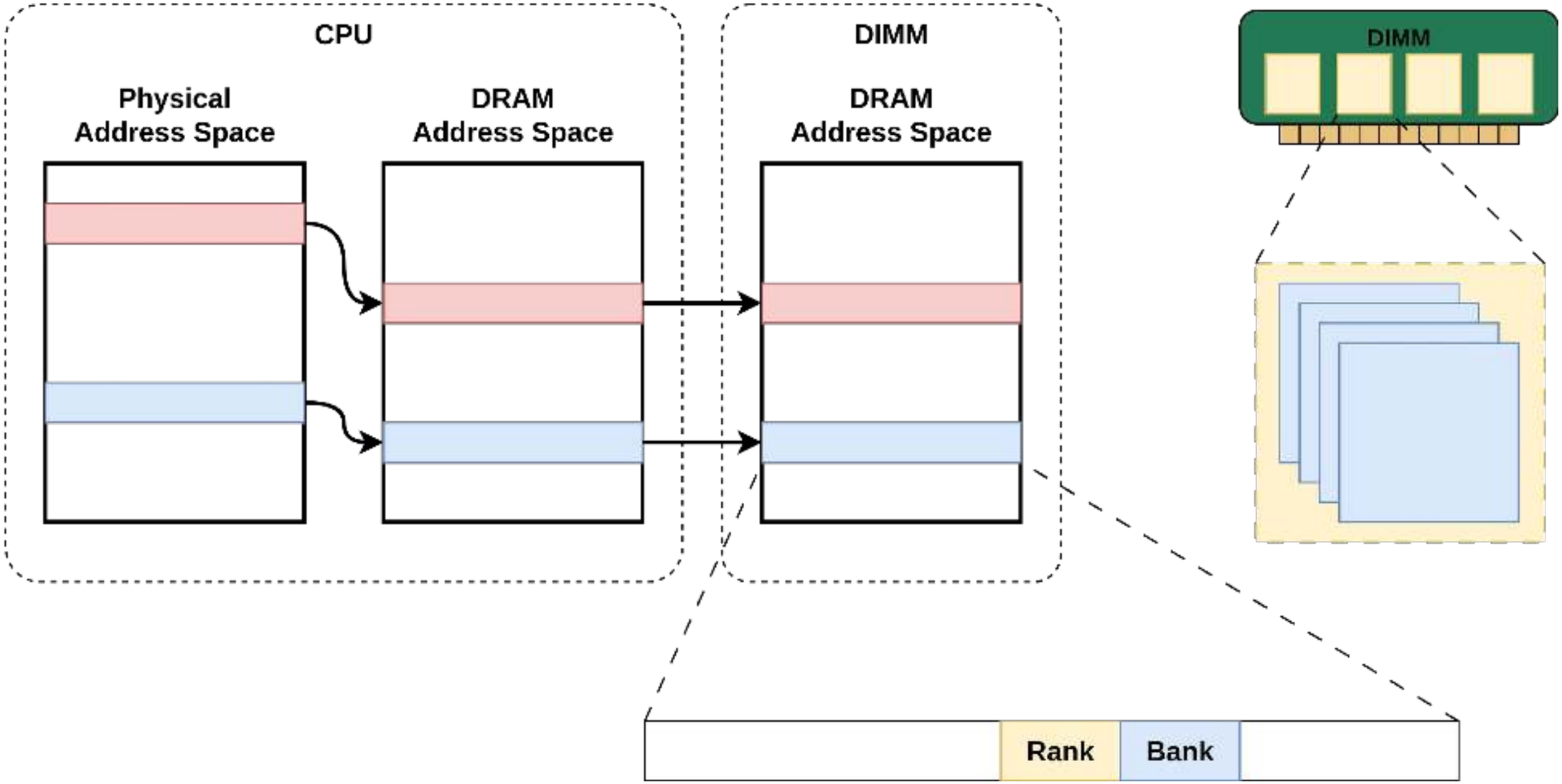
DRAM Addressing 101



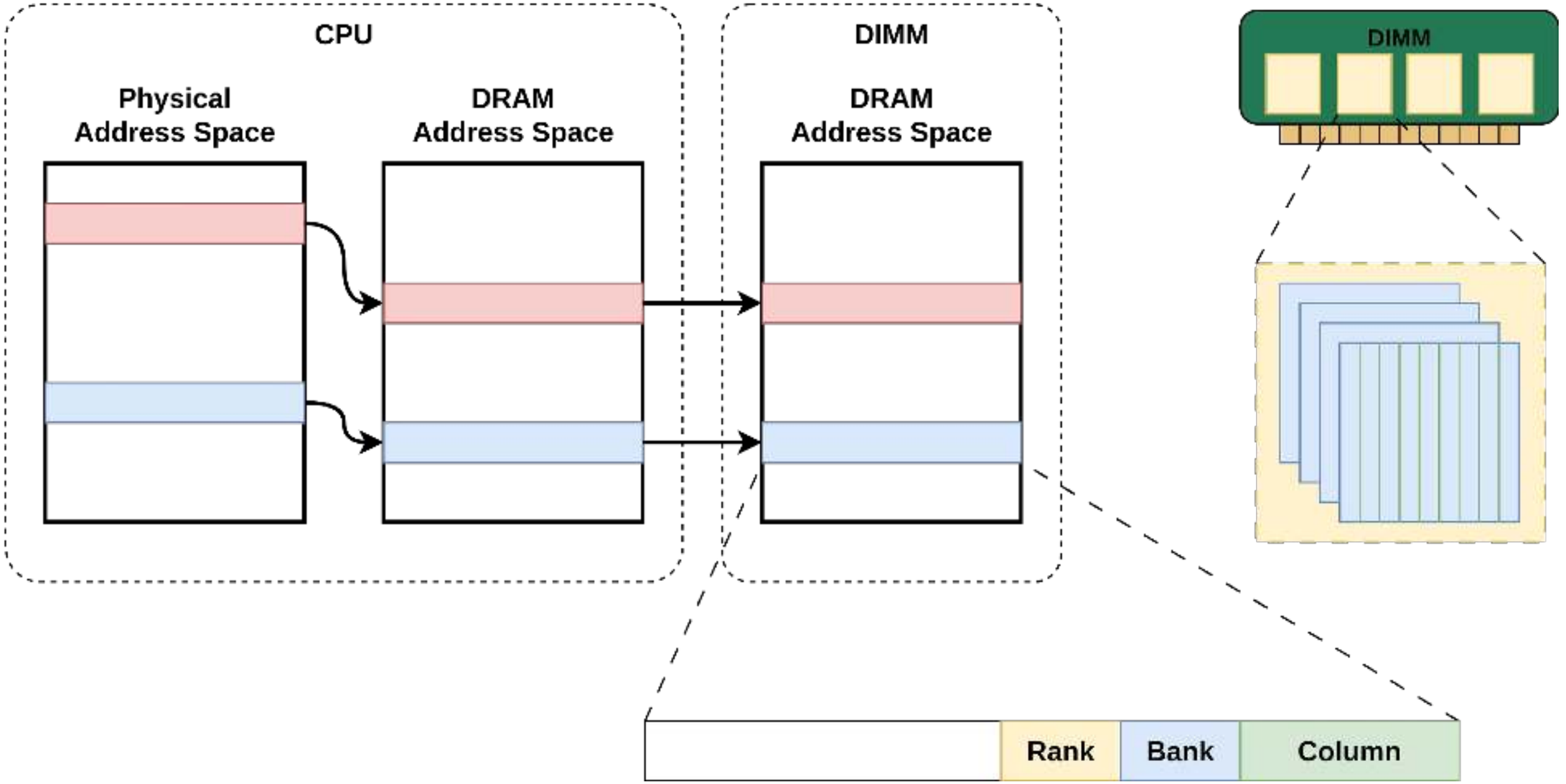
DRAM Addressing 101



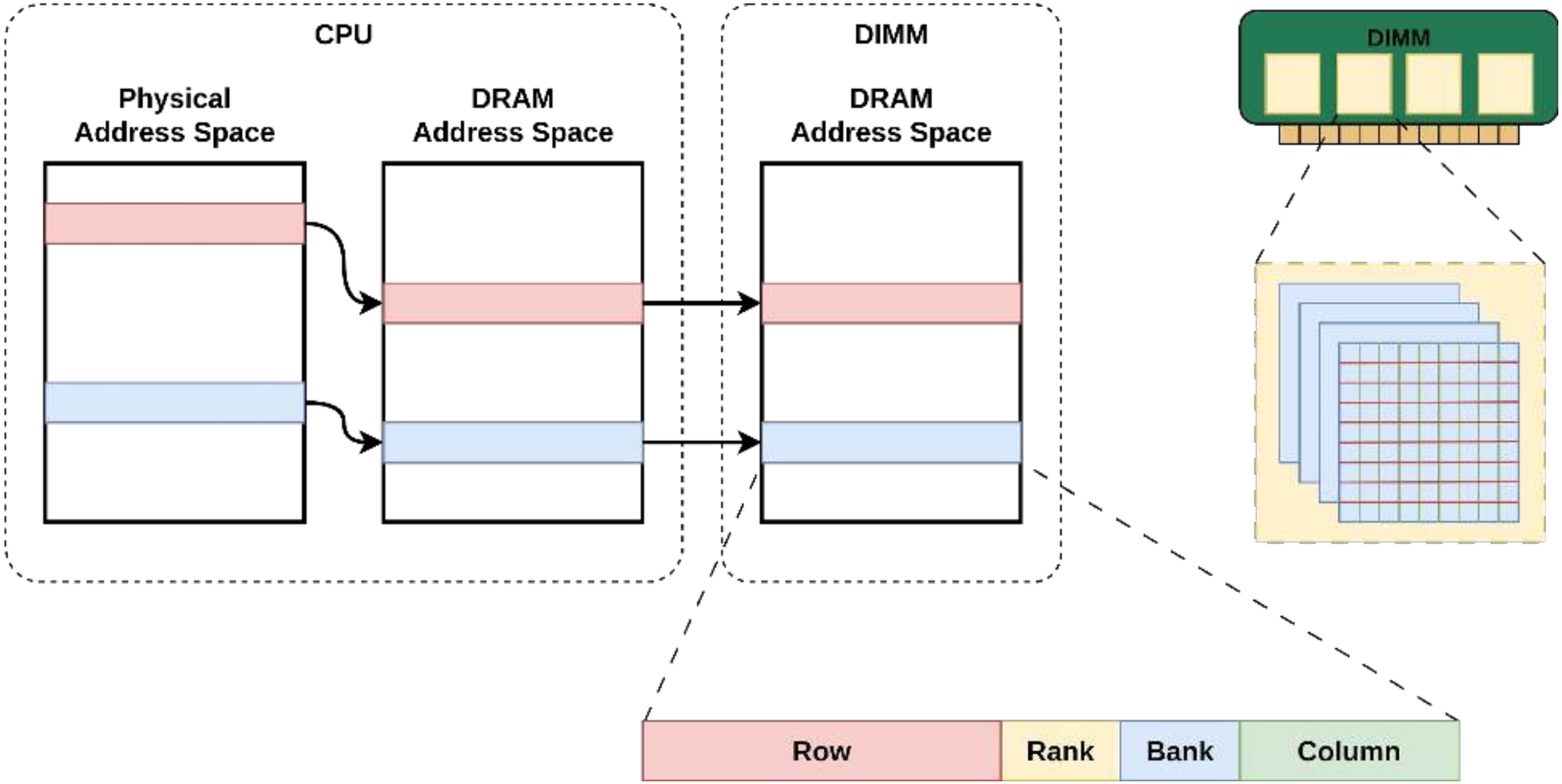
DRAM Addressing 101



DRAM Addressing 101



DRAM Addressing 101



Serial Presence Detect

- EEPROM
- Contains DIMM configuration
  - Size
  - Speed
  - Manufacturer

Serial Presence Detect

- EEPROM
- Contains DIMM configuration
  - Size
  - Speed
  - Manufacturer

Serial Presence Detect

- EEP
- Co
  -
  -
  -

Serial Presence Detect

- EEPROM
- Contains DIMM configuration
  - Size
  - Speed
  - Manufacturer

0	0x000	Number of Serial PD Bytes Written / SPD Device Size	1, 2
1	0x001	SPD Revision	
2	0x002	Key Byte / DRAM Device Type	
3	0x003	Key Byte / Module Type	
4	0x004	SDRAM Density and Banks	3
5	0x005	SDRAM Addressing	3
6	0x006	Primary SDRAM Package Type	3
7	0x007	SDRAM Optional Features	3
8	0x008	SDRAM Thermal and Refresh Options	3
9	0x009	Other SDRAM Optional Features	3
10	0x00A	Secondary SDRAM Package Type	3
11	0x00B	Module Nominal Voltage, VDD	3
12	0x00C	Module Organization	
13	0x00D	Module Memory Bus Width	
14	0x00E	Module Thermal Sensor	
15	0x00F	Extended module type	
16	0x010	Reserved -- must be coded as 0x00	
17	0x011	Timebases	
18	0x012	SDRAM Minimum Cycle Time (t_{CKAVG}min)	3
19	0x013	SDRAM Maximum Cycle Time (t_{CKAVG}max)	3

Serial Presence Detect

- EEPROM
- Contains DIMM configuration
  - Size
  - Speed
  - Manufacturer

0	0x000	Number of Serial PD Bytes Written / SPD Device Size	1, 2
1	0x001	SPD Revision	
2	0x002	Key Byte / DRAM Device Type	
3	0x003	Key Byte / Module Type	
4	0x004	SDRAM Density and Banks	3
5	0x005	SDRAM Addressing	3
6	0x006	Primary SDRAM Package Type	3
7	0x007	SDRAM Optional Features	3
8	0x008	SDRAM Thermal and Refresh Options	3
9	0x009	Other SDRAM Optional Features	3
10	0x00A	Secondary SDRAM Package Type	3
11	0x00B	Module Nominal Voltage, VDD	3
12	0x00C	Module Organization	
13	0x00D	Module Memory Bus Width	
14	0x00E	Module Thermal Sensor	
15	0x00F	Extended module type	
16	0x010	Reserved -- must be coded as 0x00	
17	0x011	Timebases	
18	0x012	SDRAM Minimum Cycle Time (t_{CKAVG}min)	3
19	0x013	SDRAM Maximum Cycle Time (t_{CKAVG}max)	3

Serial Presence Detect

- EEPROM
- Contains DIMM configuration
  - Size
  - Speed
  - Manufacturer

0	0x000	Number of Serial PD Bytes Written / SPD Device Size	1, 2
1	0x001	SPD Revision	
2	0x002	Key Byte / DRAM Device Type	
3	0x003	Key Byte / Module Type	
4	0x004	SDRAM Density and Banks	3
5	0x005	SDRAM Addressing	3
6	0x006	Primary SDRAM Package Type	3

Byte 5 (0x005): SDRAM Addressing

This byte describes the row addressing and the column addressing in the SDRAM device. Bits 2~0 encode the number of column address bits, and bits 5~3 encode the number of row address bits. These values come from the DDR4 SDRAM data sheet.

Bits 7~6	Bits 5~3	Bits 2~0
Reserved	**Row Address Bits**	**Column Address Bits**
Reserved; must be coded as 00	Bit [5, 4, 3] : 000 = 12 001 = 13 010 = 14 011 = 15 100 = 16 101 = 17 110 = 18 All others reserved	Bit [2, 1, 0] : 000 = 9 001 = 10 010 = 11 011 = 12 All others reserved

17	0x011	Timebases	
18	0x012	SDRAM Minimum Cycle Time (t_{CKAVG}min)	3
19	0x013	SDRAM Maximum Cycle Time (t_{CKAVG}max)	3

Serial Presence Detect

- EEPROM
- Contains DIMM configuration
  - Size
  - Speed
  - Manufacturer

0	0x000	Number of Serial PD Bytes Written / SPD Device Size	1, 2
1	0x001	SPD Revision	
2	0x002	Key Byte / DRAM Device Type	
3	0x003	Key Byte / Module Type	
4	0x004	SDRAM Density and Banks	3
5	0x005	SDRAM Addressing	3
6	0x006	Primary SDRAM Package Type	3

Byte 5 (0x005): SDRAM Addressing

This byte describes the row addressing and the column addressing in the SDRAM device. Bits 2~0 encode the number of column address bits, and bits 5~3 encode the number of row address bits. These values come from the DDR4 SDRAM data sheet.

Bits 7~6	Bits 5~3	Bits 2~0
Reserved	Row Address Bits	**Column Address Bits**
Reserved; must be coded as 00	Bit [5, 4, 3] : 000 = 12 001 = 13 010 = 14 011 = 15 100 = 16 101 = 17 110 = 18 All others reserved	Bit [2, 1, 0] : 000 = 9 001 = 10 010 = 11 011 = 12 All others reserved

17	0x011	Timebases	
18	0x012	SDRAM Minimum Cycle Time (t_{CKAVG}min)	3
19	0x013	SDRAM Maximum Cycle Time (t_{CKAVG}max)	3

Serial Presence Detect

- EEPROM
- Contains DIMM configuration
  - Size
  - Speed
  - Manufacturer

- ***What if we overwrite this data?***

0	0x000	Number of Serial PD Bytes Written / SPD Device Size	1, 2
1	0x001	SPD Revision	
2	0x002	Key Byte / DRAM Device Type	
3	0x003	Key Byte / Module Type	
4	0x004	SDRAM Density and Banks	3
5	0x005	SDRAM Addressing	3
6	0x006	Primary SDRAM Package Type	3

Byte 5 (0x005): SDRAM Addressing

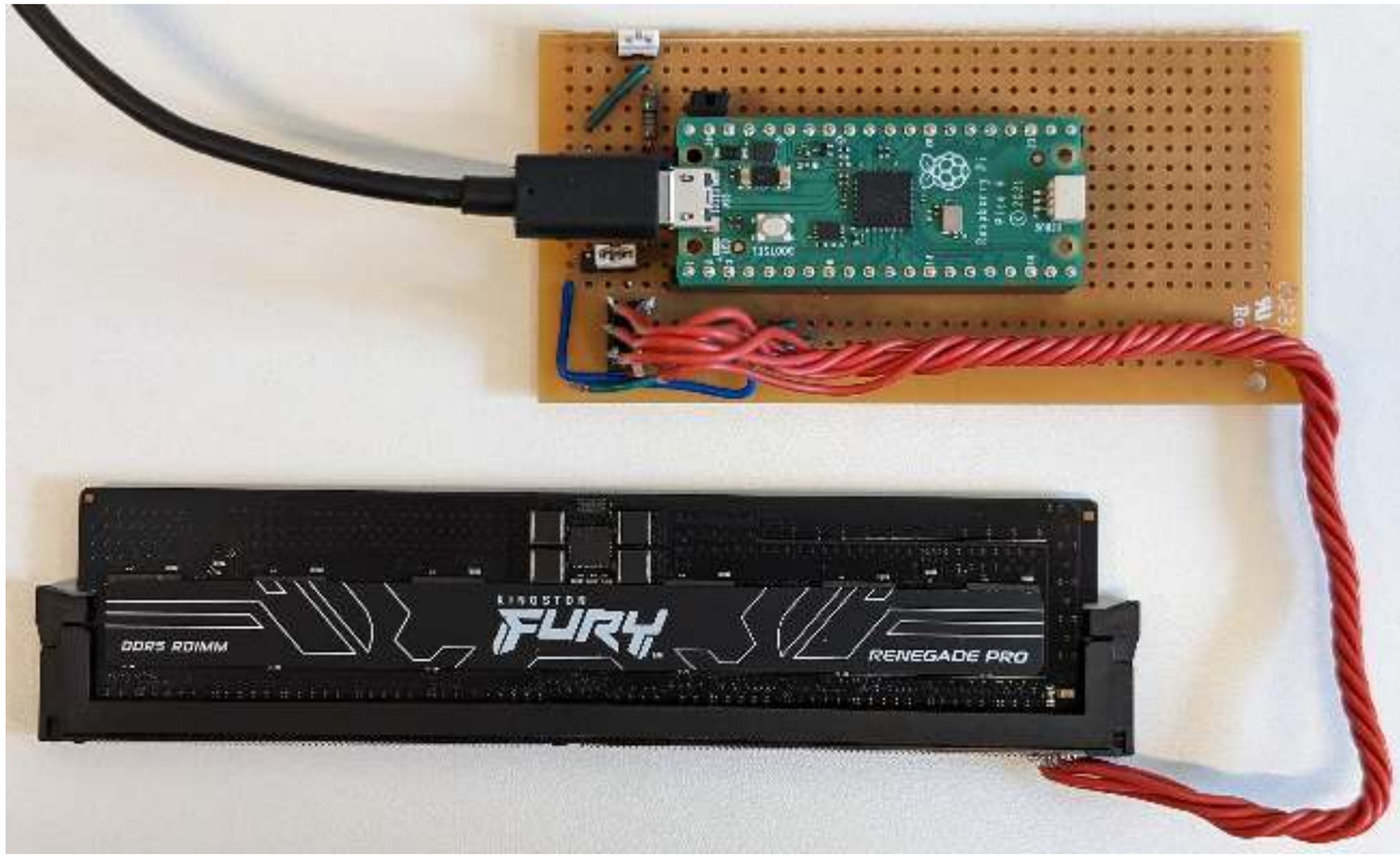
This byte describes the row addressing and the column addressing in the SDRAM device. Bits 2~0 encode the number of column address bits, and bits 5~3 encode the number of row address bits. These values come from the DDR4 SDRAM data sheet.

Bits 7~6	Bits 5~3	Bits 2~0
Reserved	Row Address Bits	**Column Address Bits**
Reserved; must be coded as 00	Bit [5, 4, 3] : 000 = 12 001 = 13 010 = 14 011 = 15 100 = 16 101 = 17 110 = 18 All others reserved	Bit [2, 1, 0] : 000 = 9 001 = 10 010 = 11 011 = 12 All others reserved

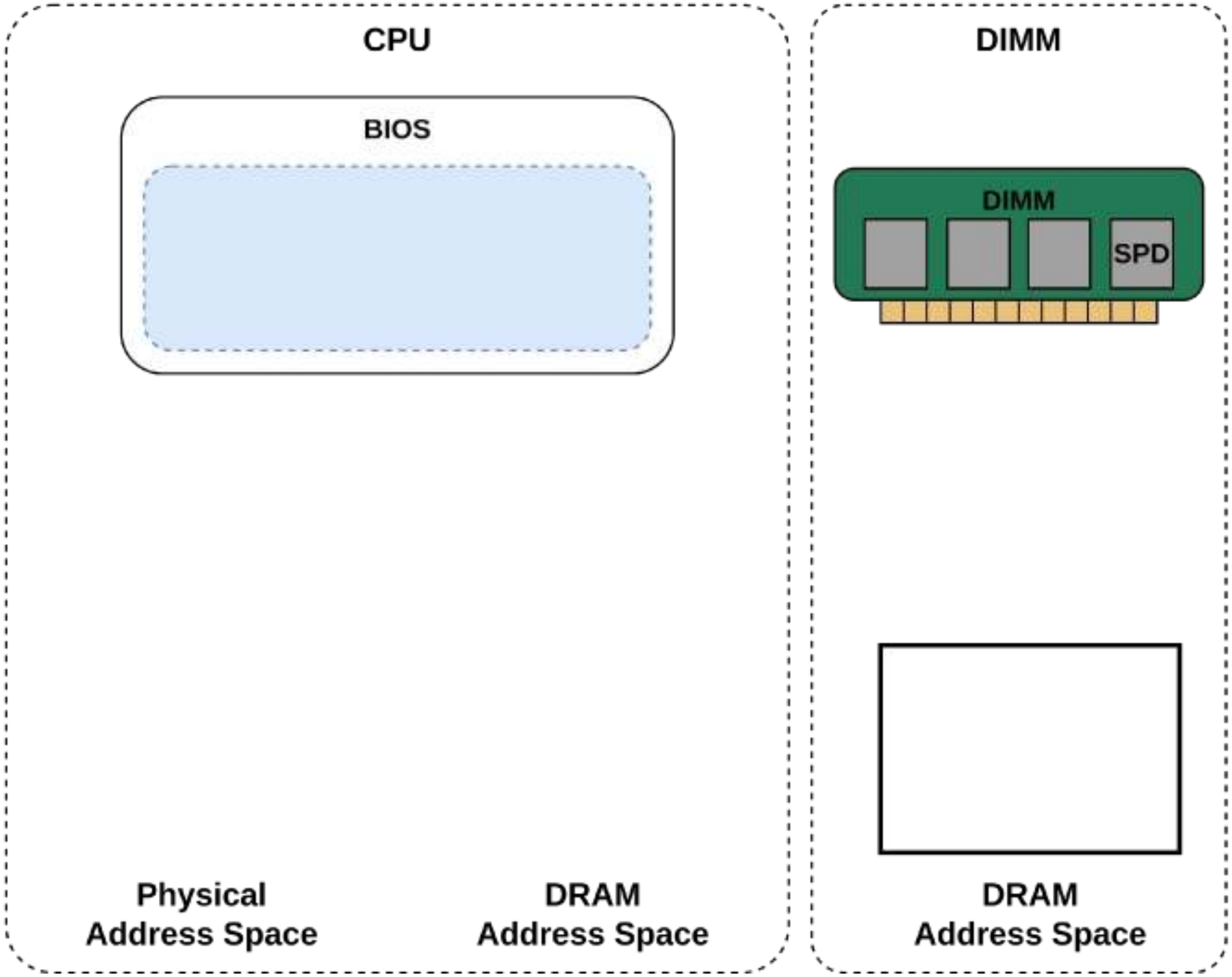
17	0x011	Timebases	
18	0x012	SDRAM Minimum Cycle Time (t_{CKAVG}min)	3
19	0x013	SDRAM Maximum Cycle Time (t_{CKAVG}max)	3

Serial Presence Detect

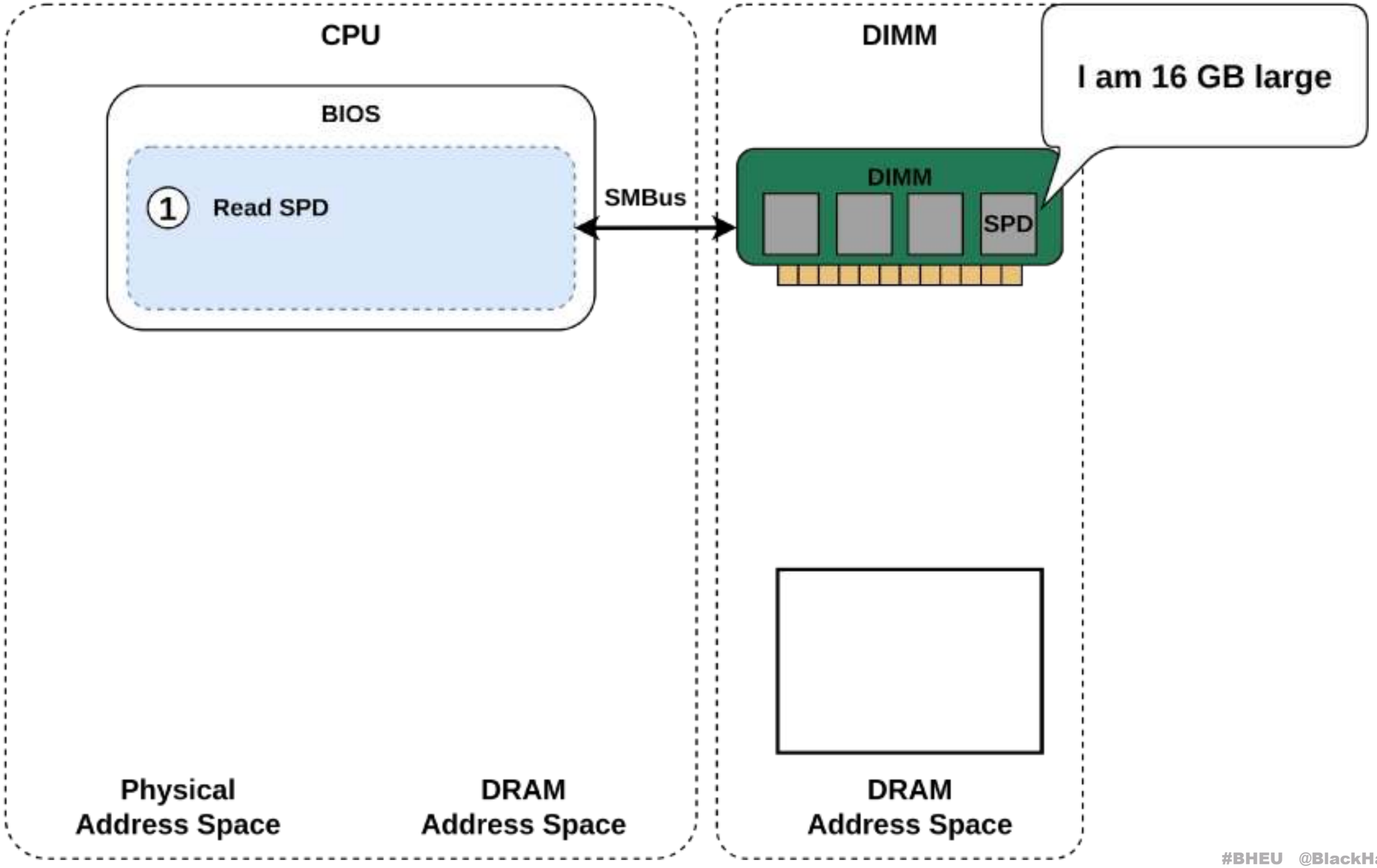
- I²C pins exposed on DIMM
- Trivial to unlock and overwrite
  - **Total cost: ~10$**



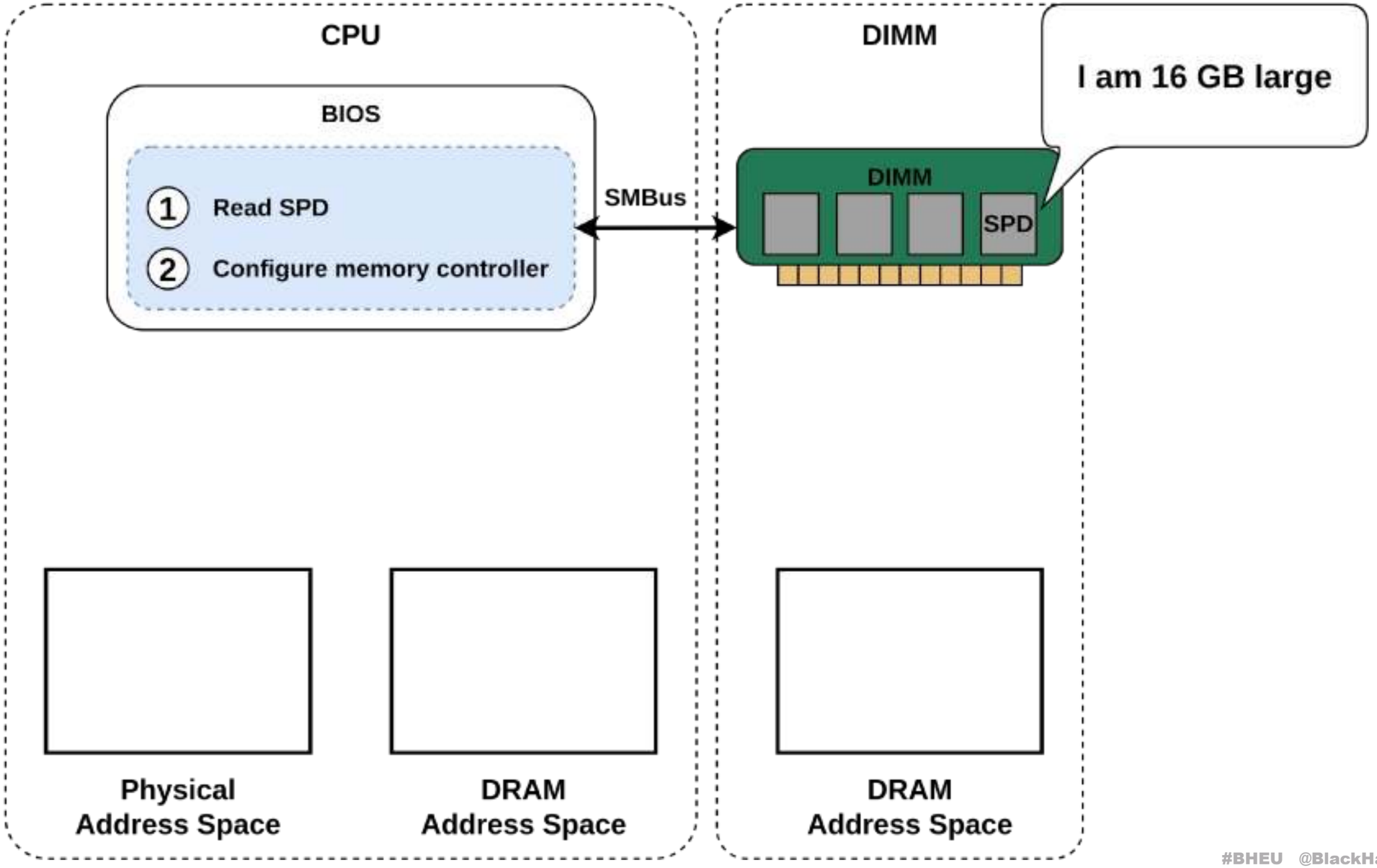
Introducing Aliases



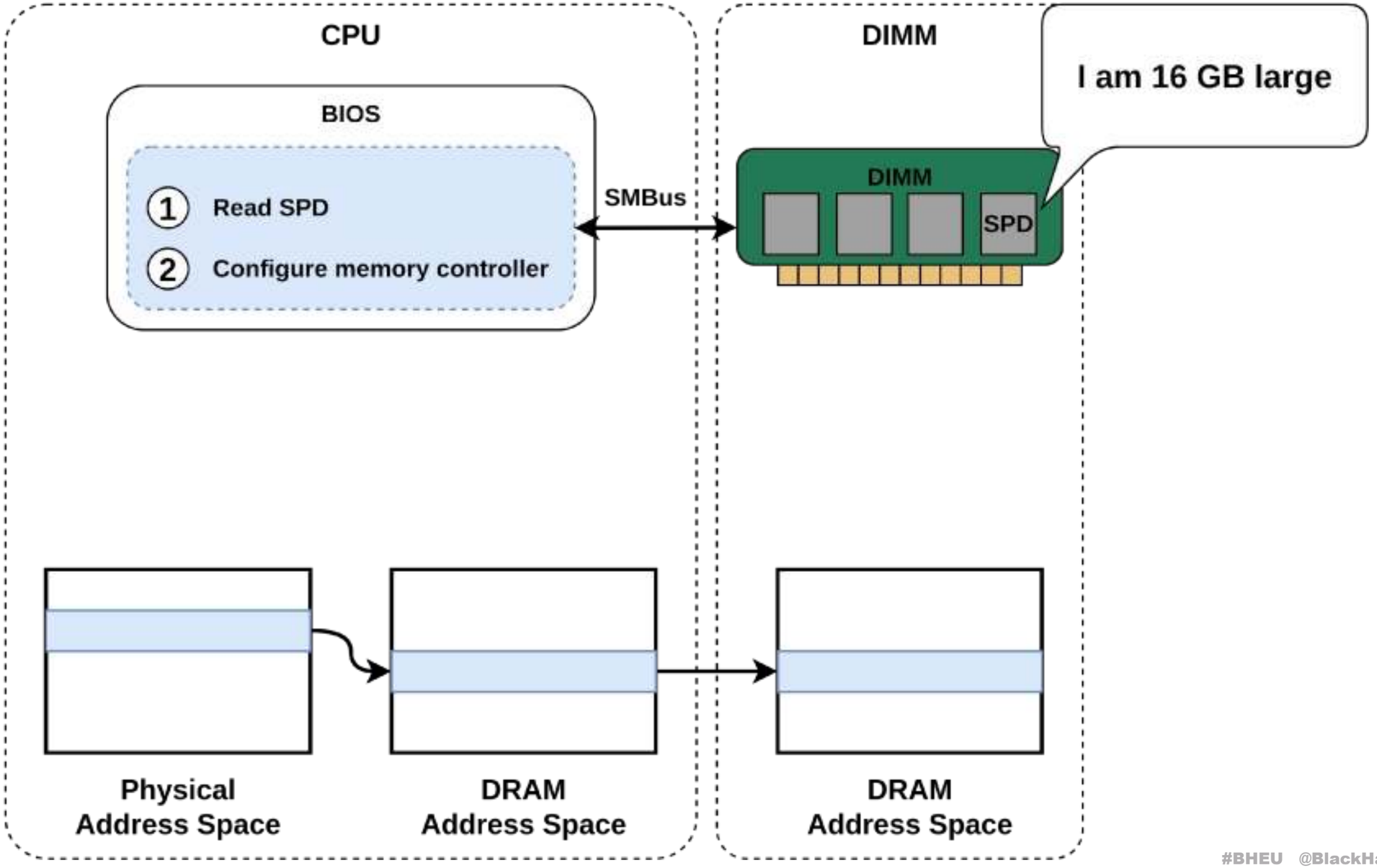
Introducing Aliases



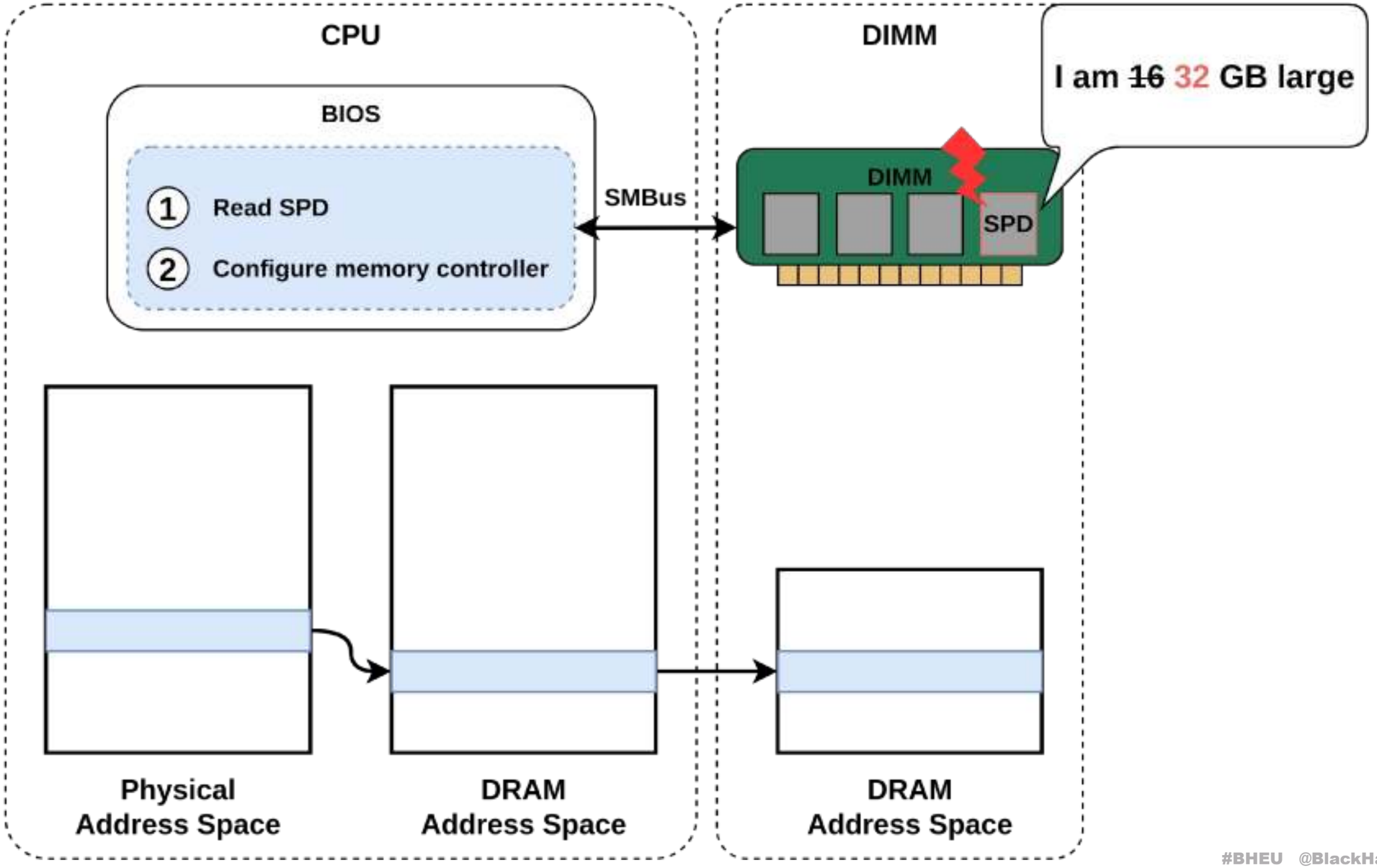
Introducing Aliases



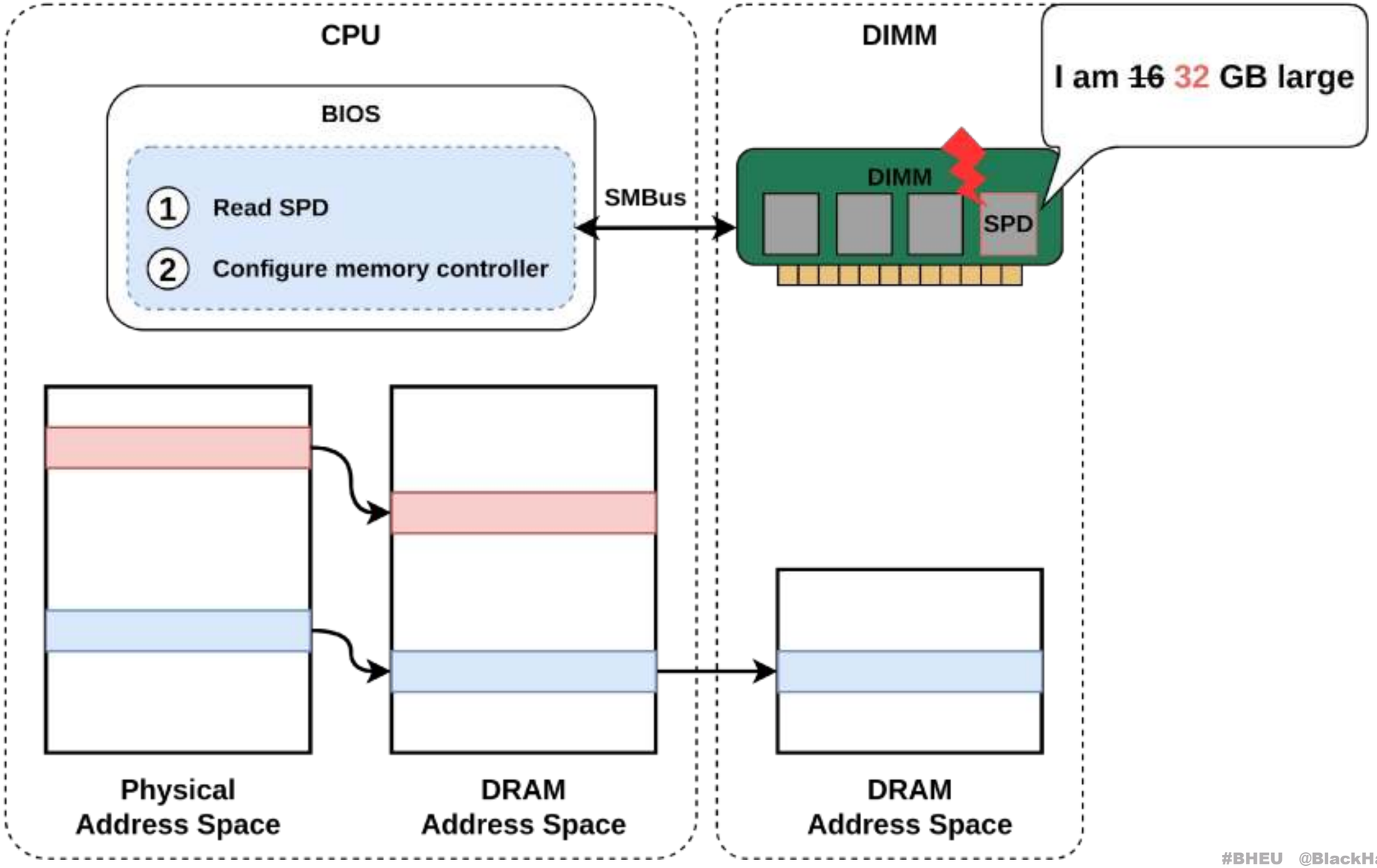
Introducing Aliases



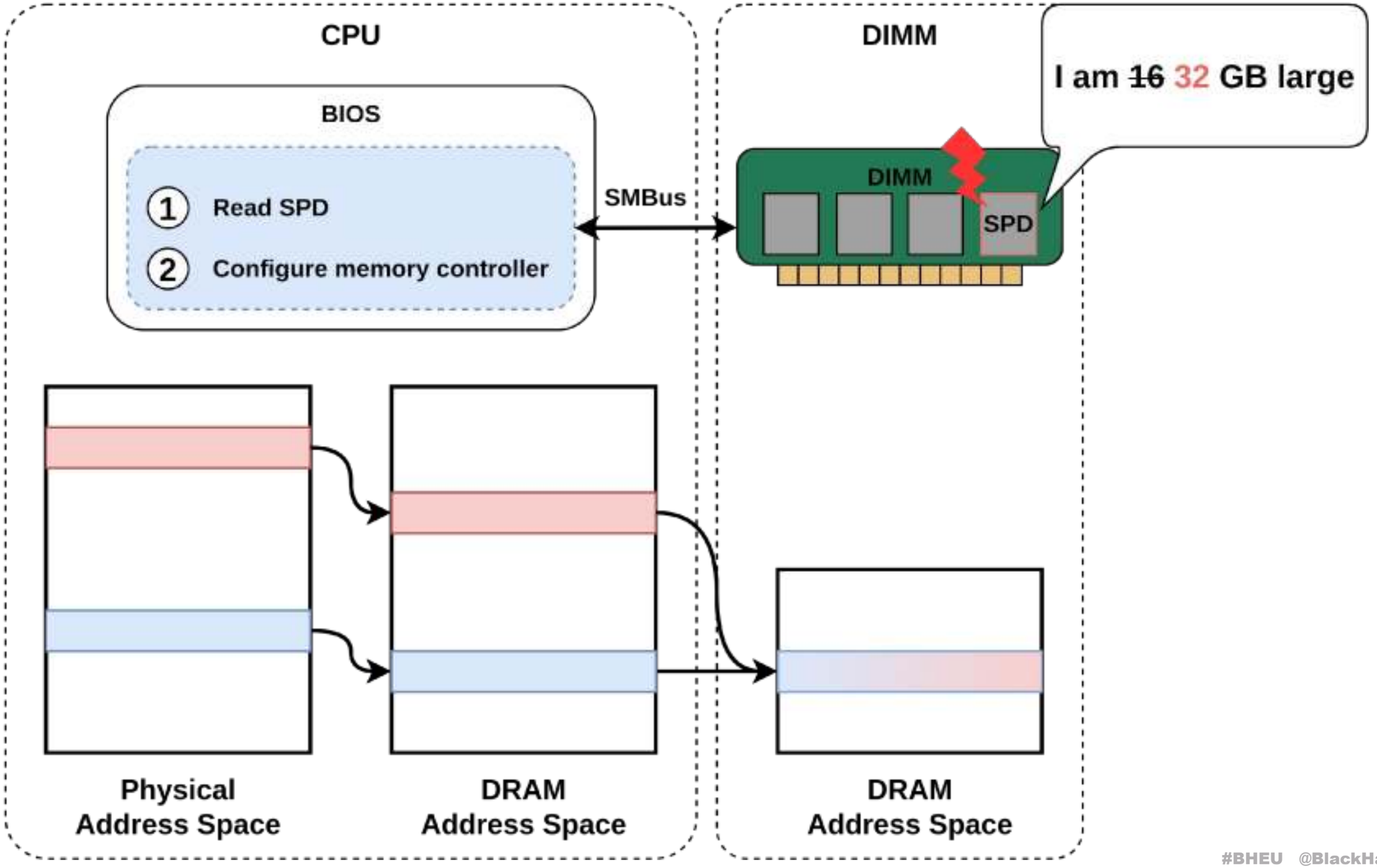
Introducing Aliases



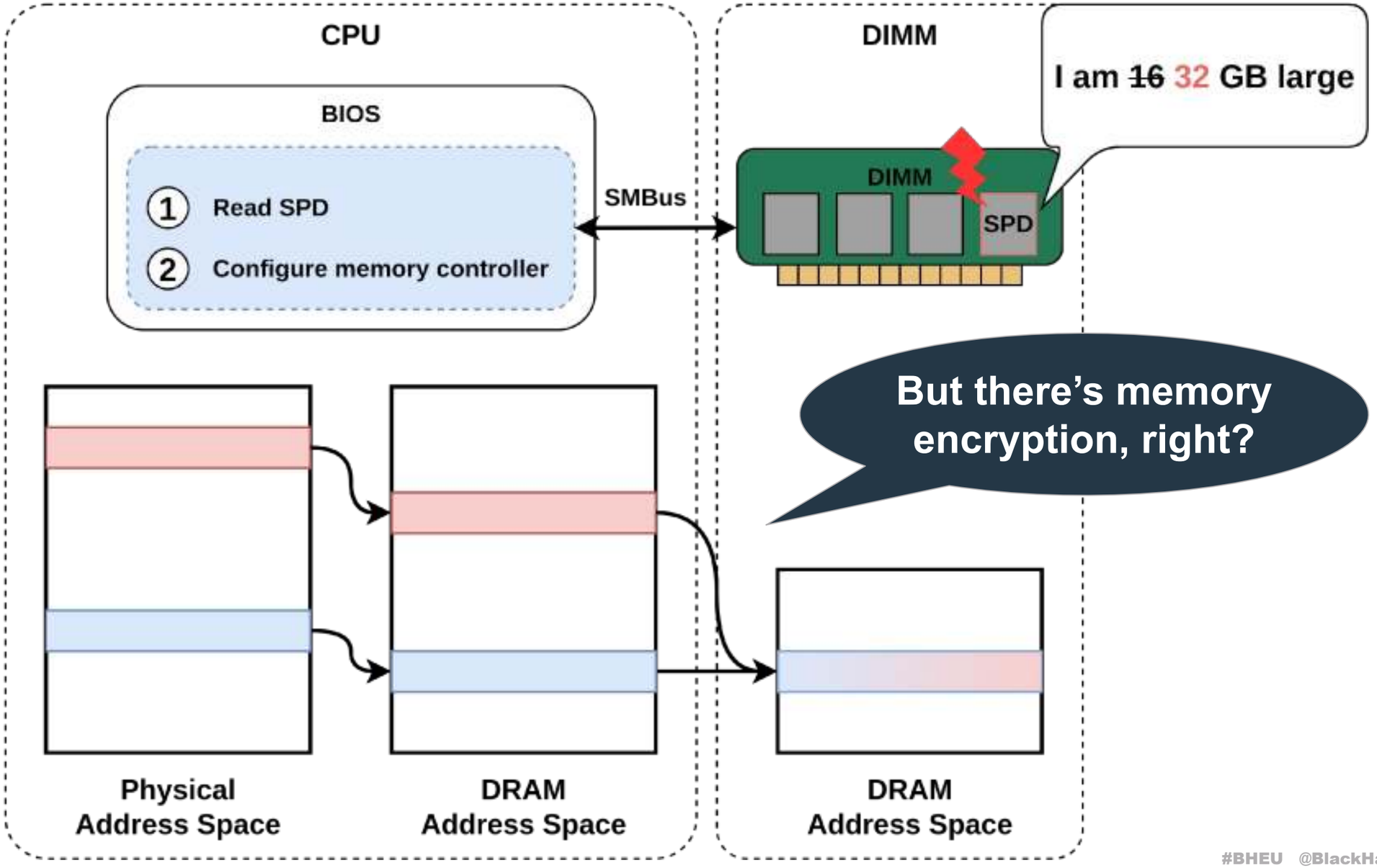
Introducing Aliases



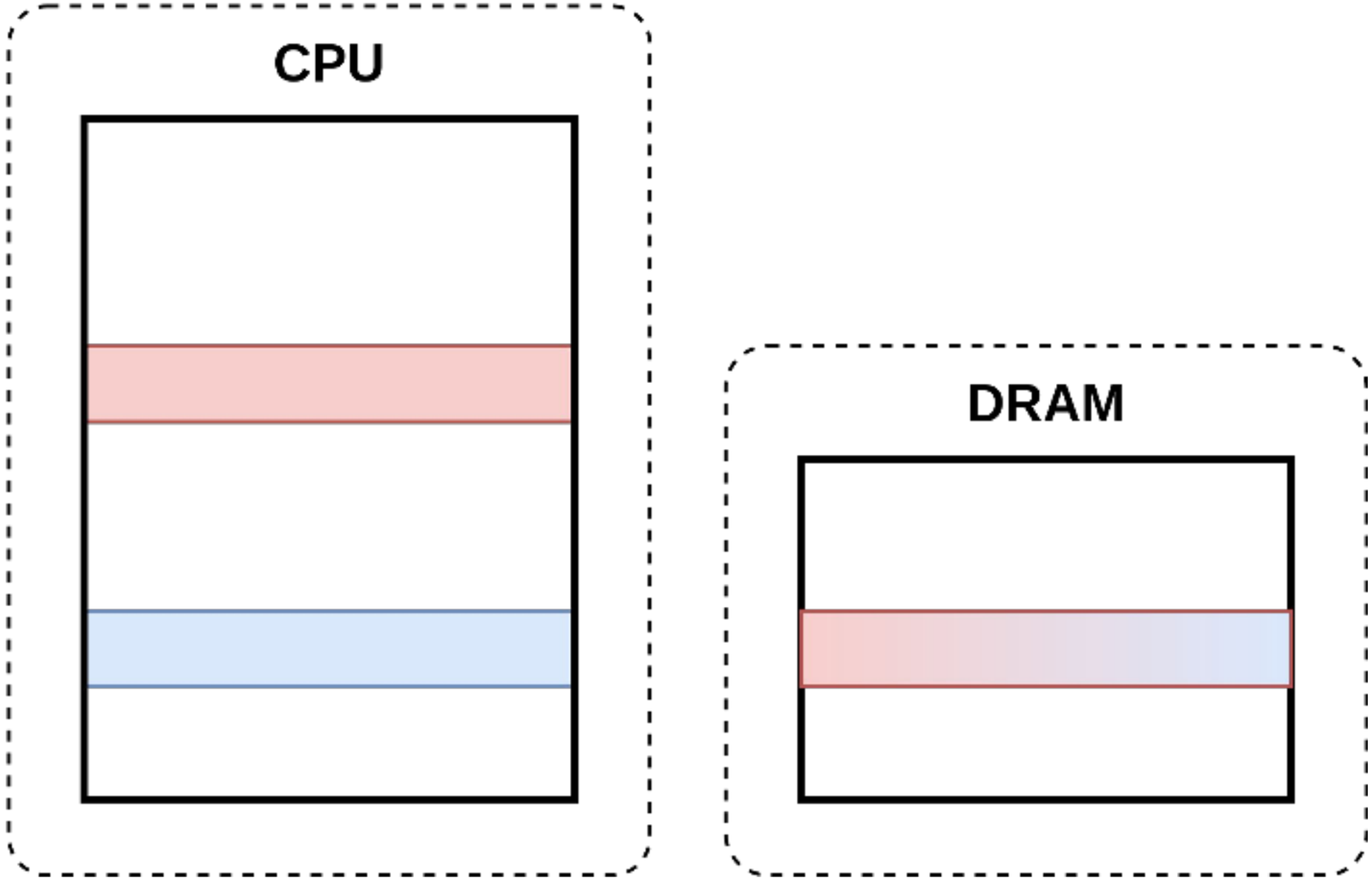
Introducing Aliases



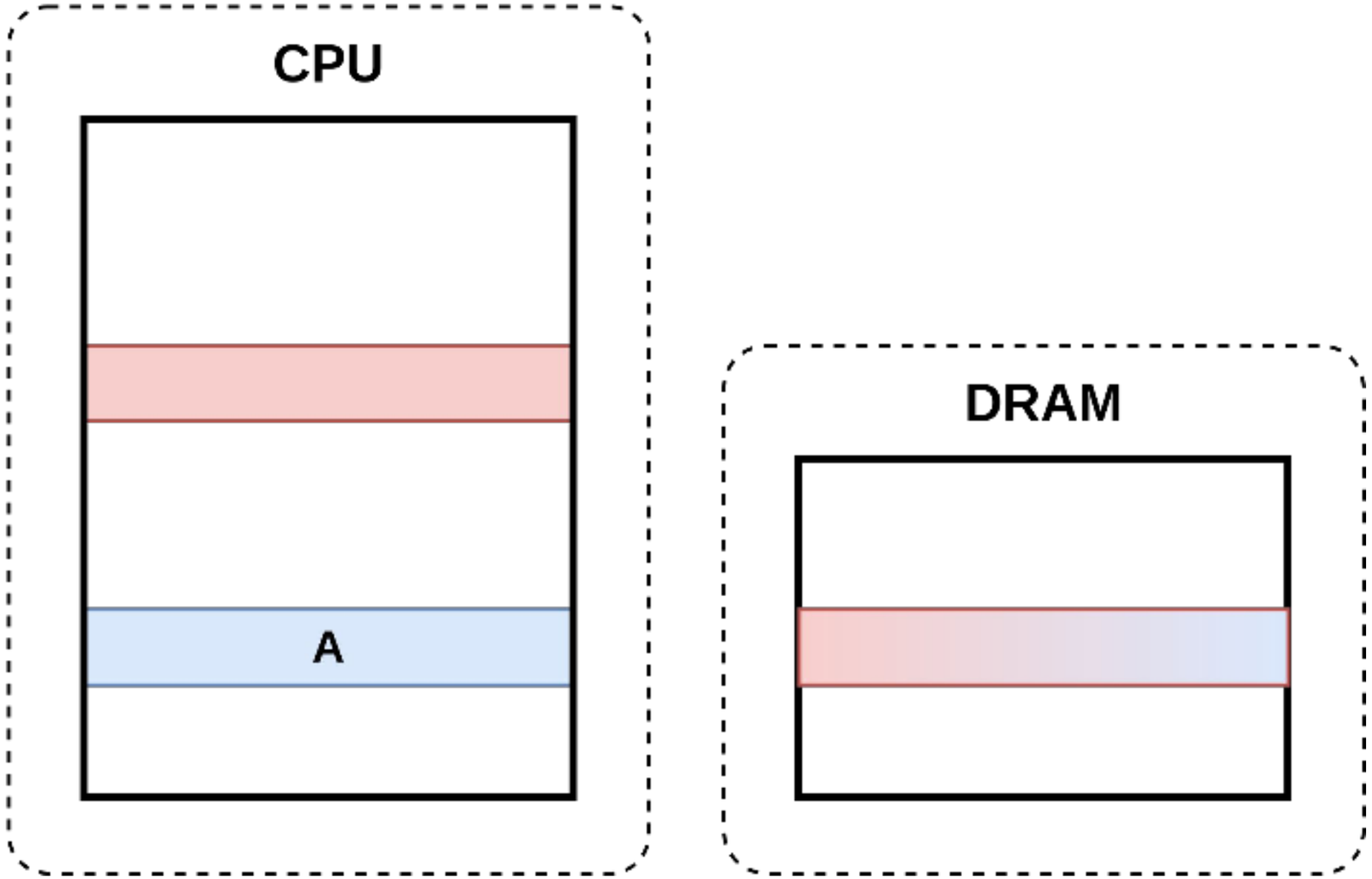
Introducing Aliases



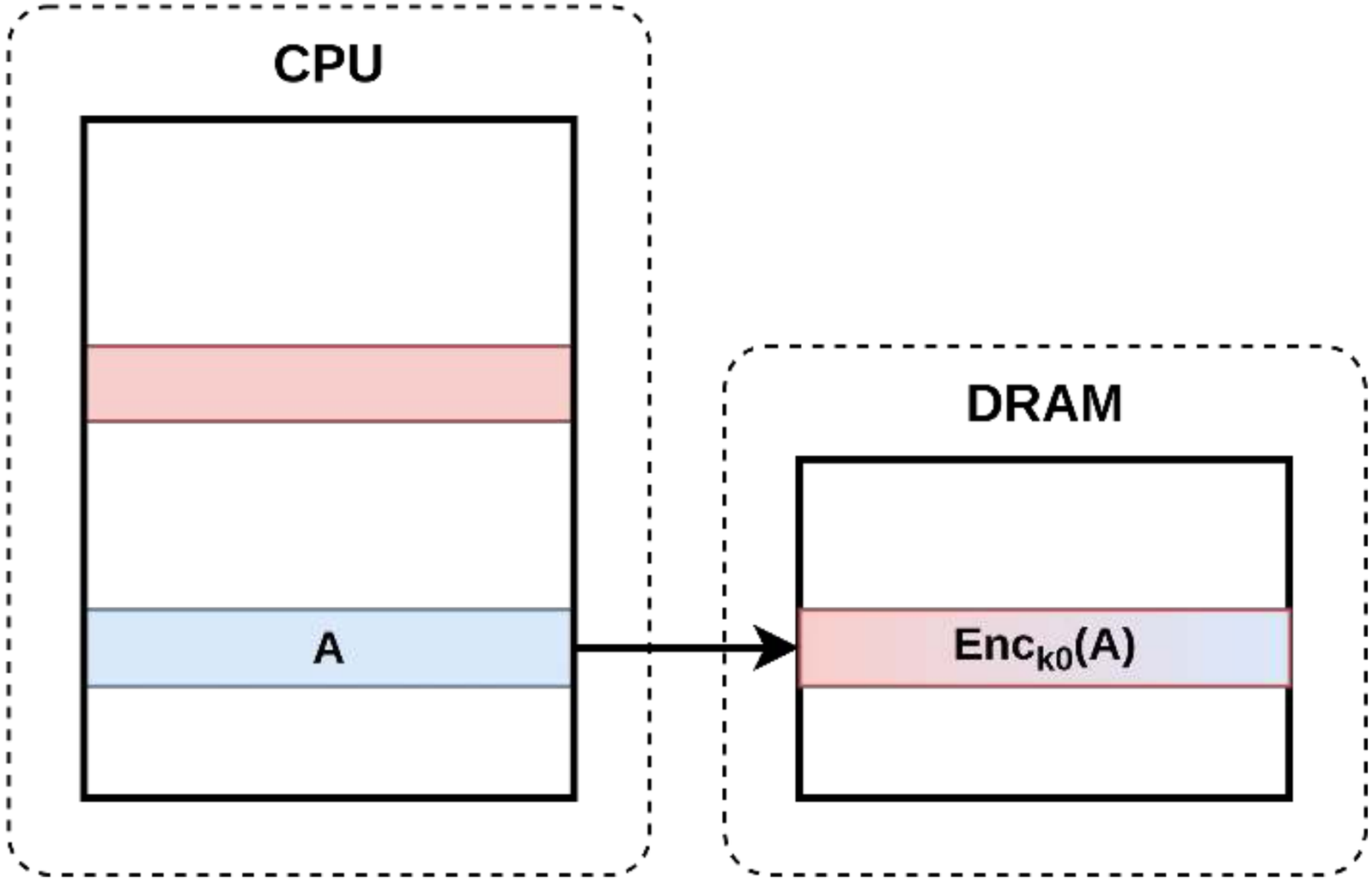
Introducing Aliases – Static Memory Encryption



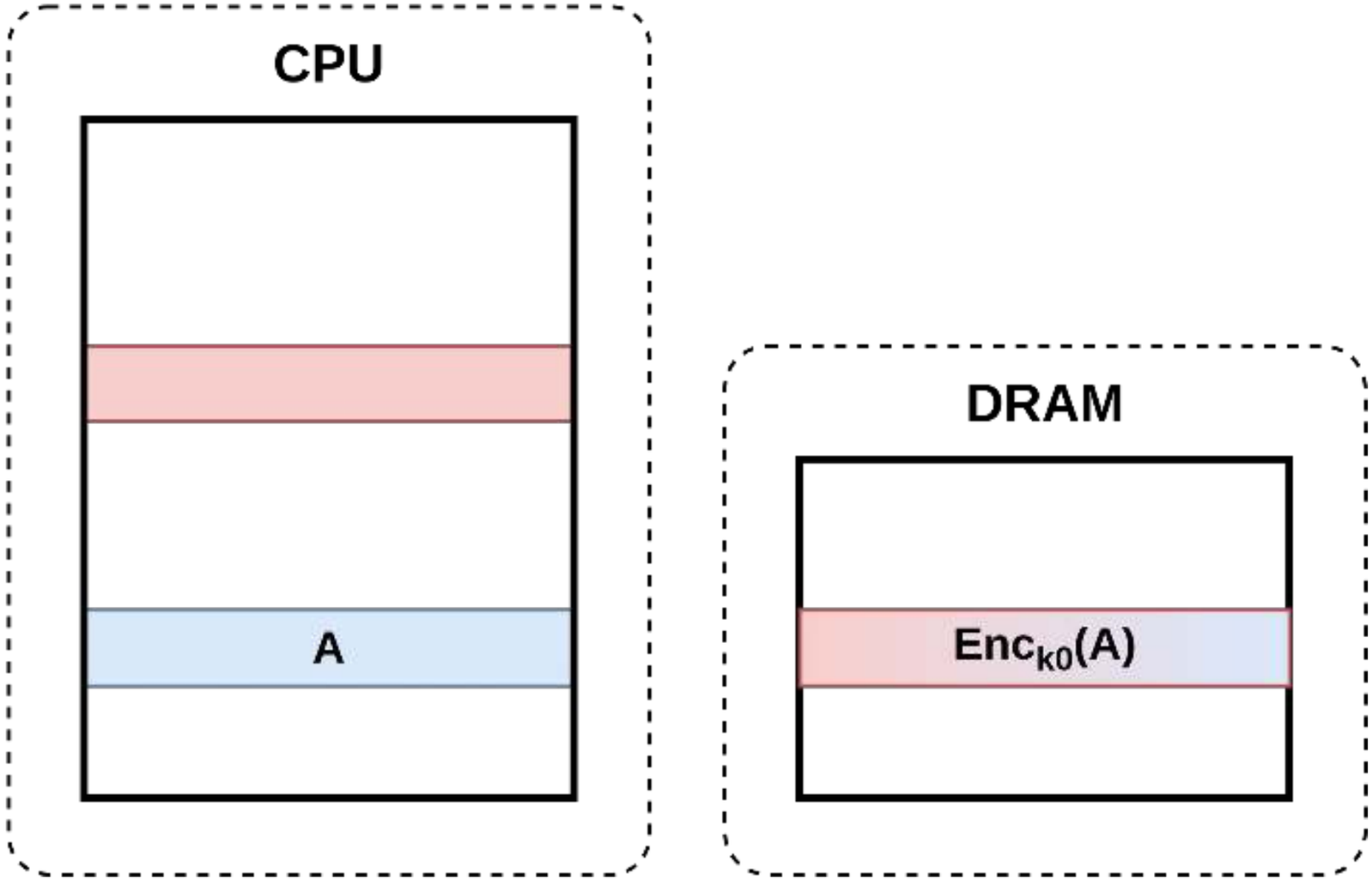
Introducing Aliases – Static Memory Encryption



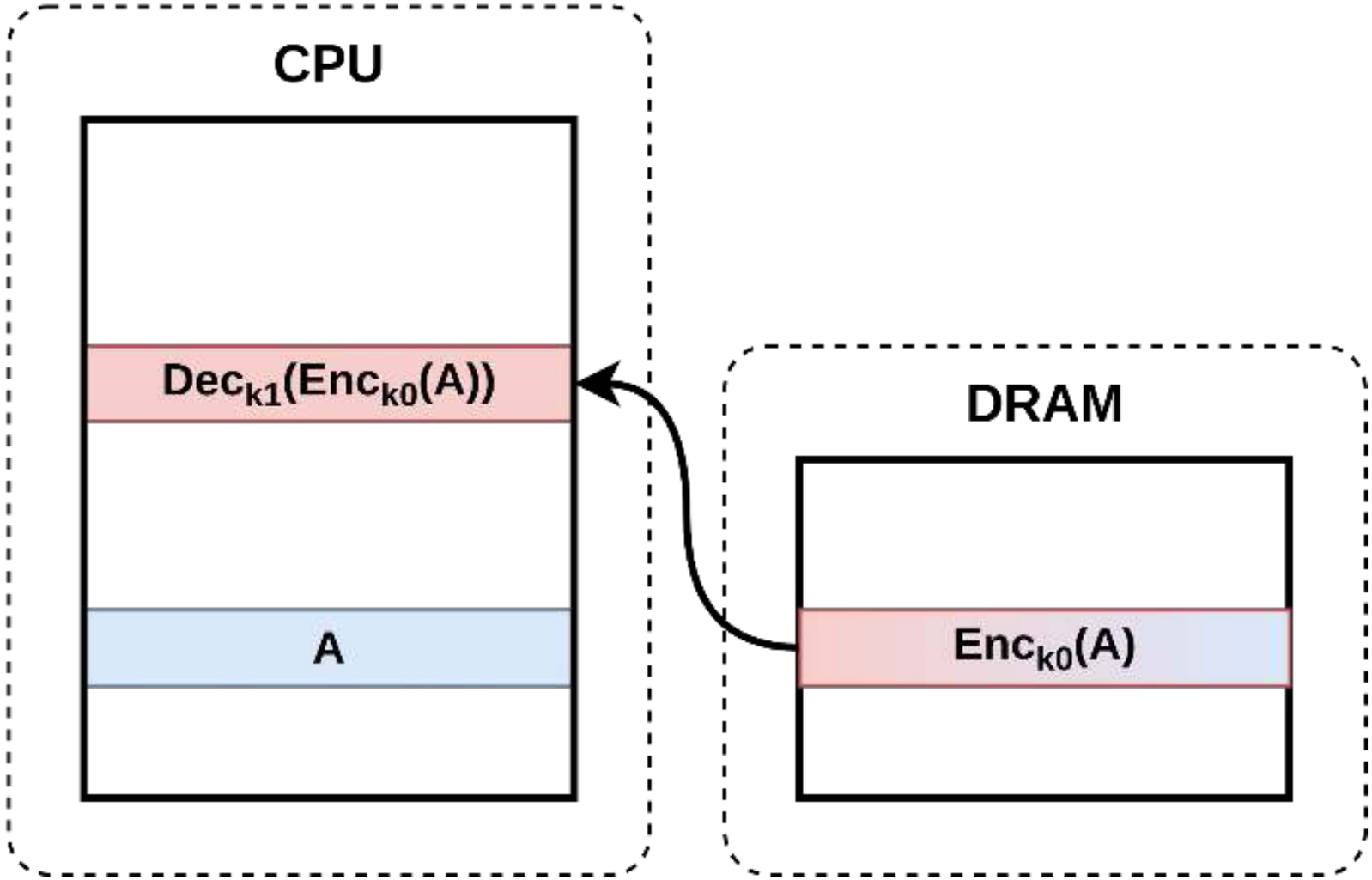
Introducing Aliases – Static Memory Encryption



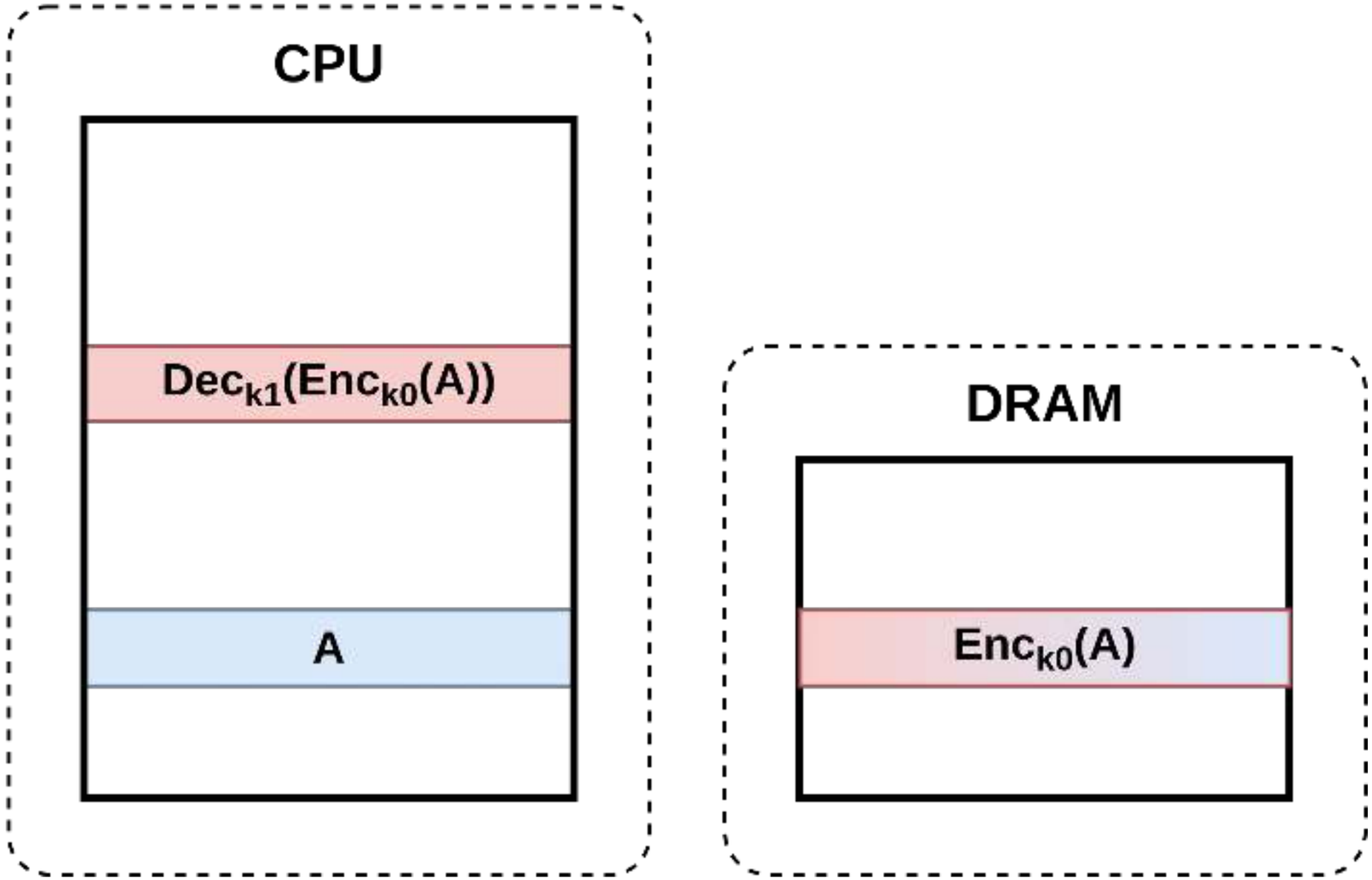
Introducing Aliases – Static Memory Encryption



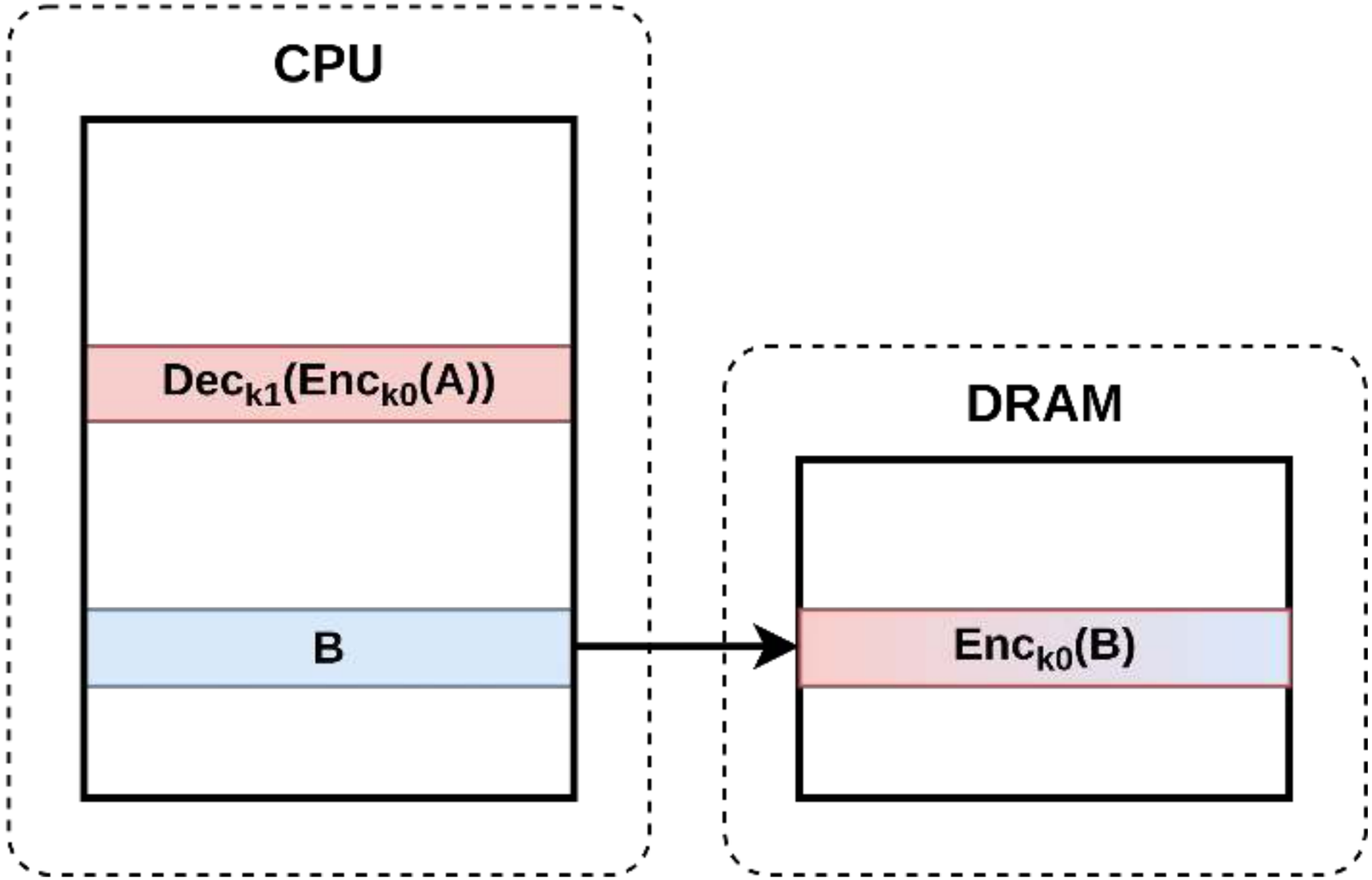
Introducing Aliases – Static Memory Encryption



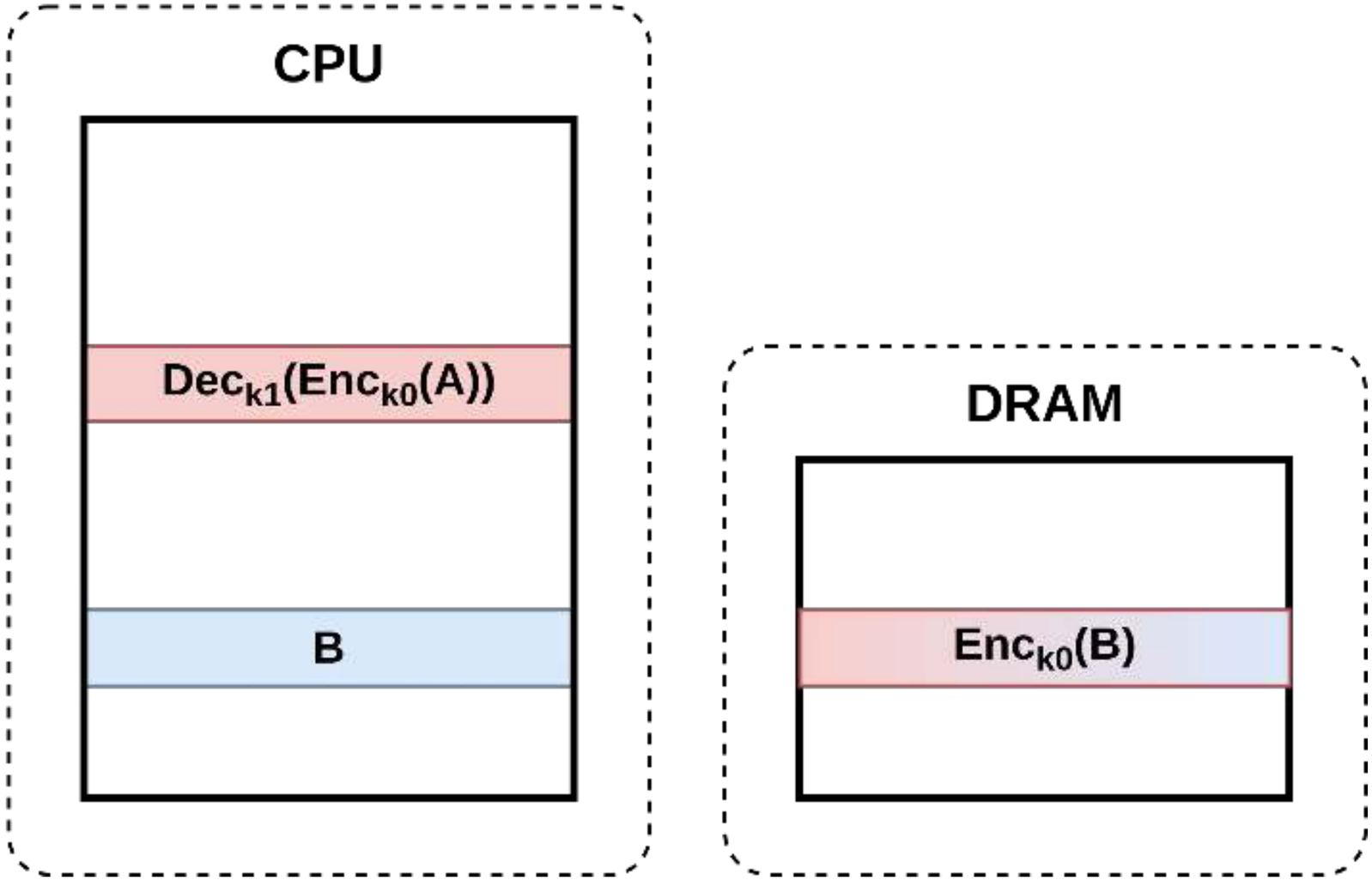
Introducing Aliases – Static Memory Encryption



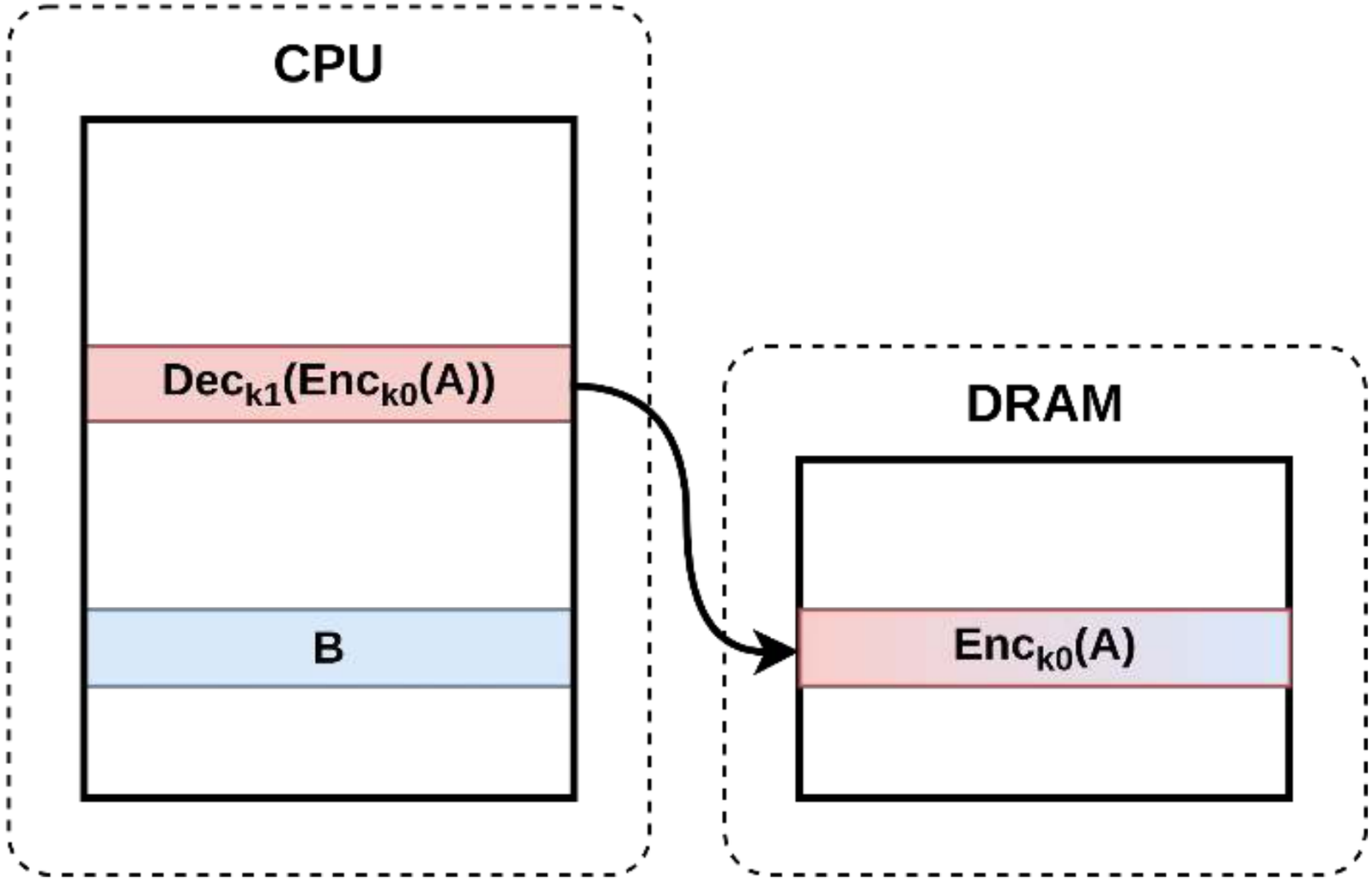
Introducing Aliases – Static Memory Encryption



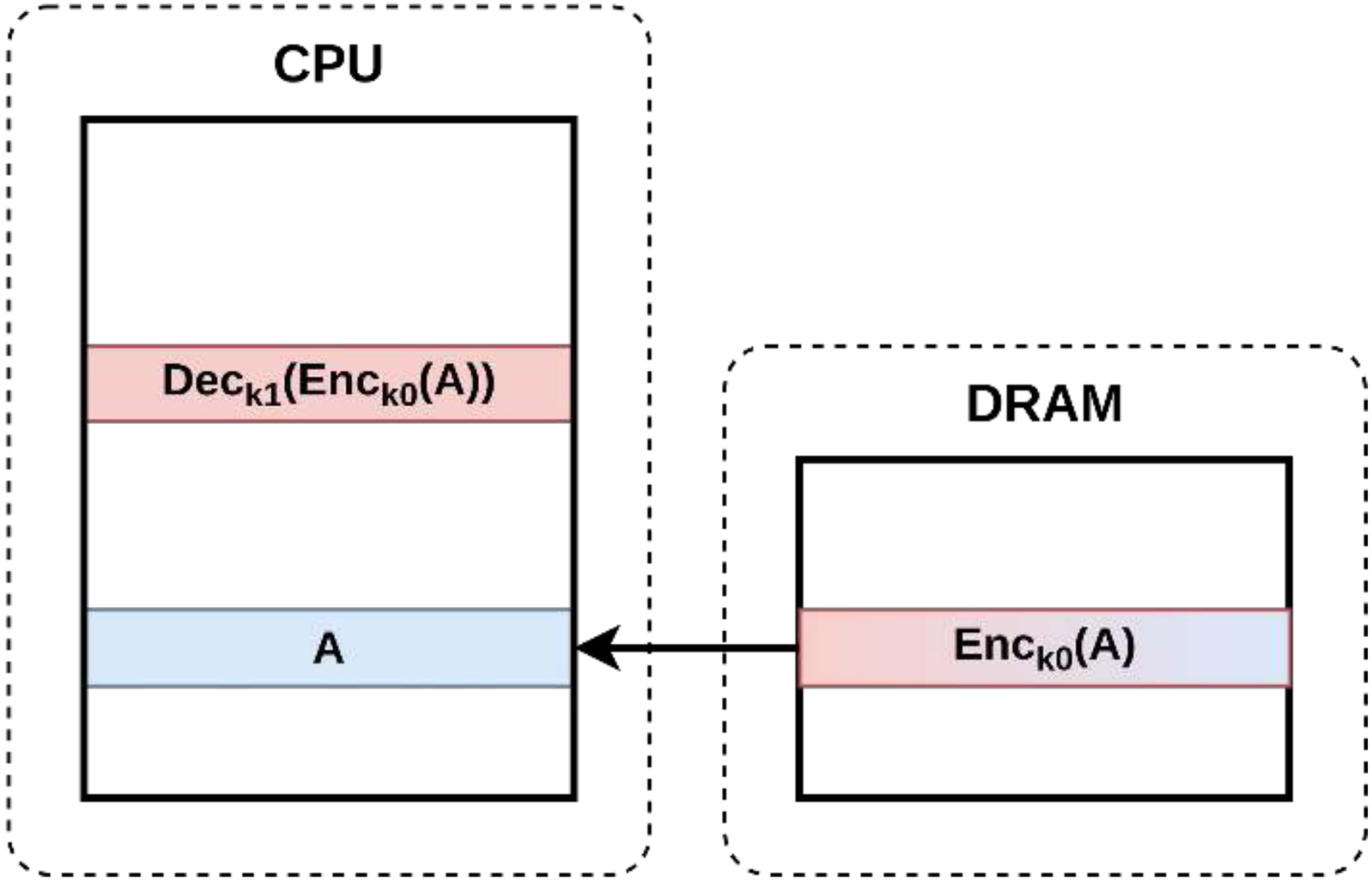
Introducing Aliases – Static Memory Encryption



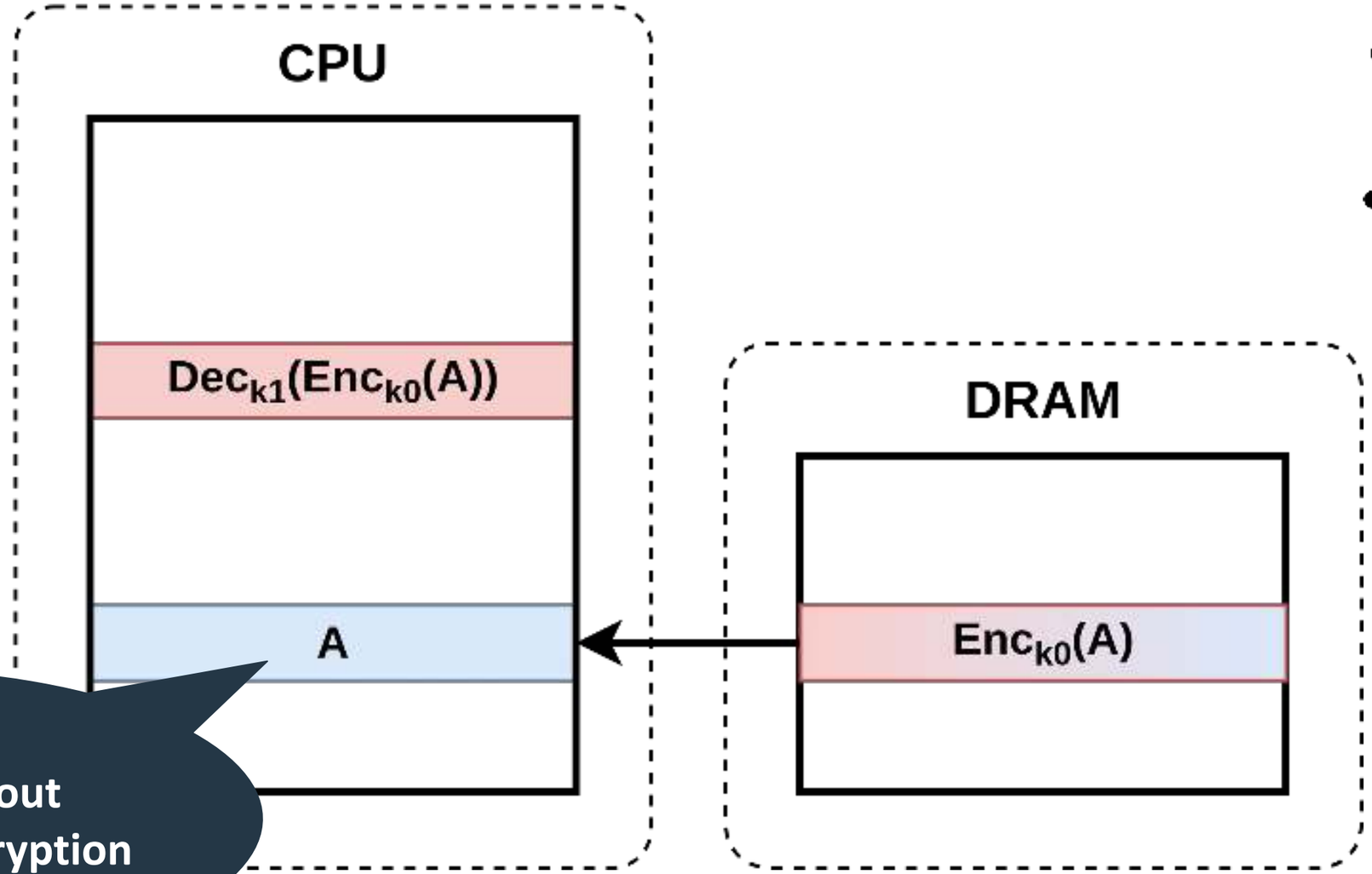
Introducing Aliases – Static Memory Encryption



Introducing Aliases – Static Memory Encryption



Introducing Aliases – Static Memory Encryption



Replay cancels out static memory encryption

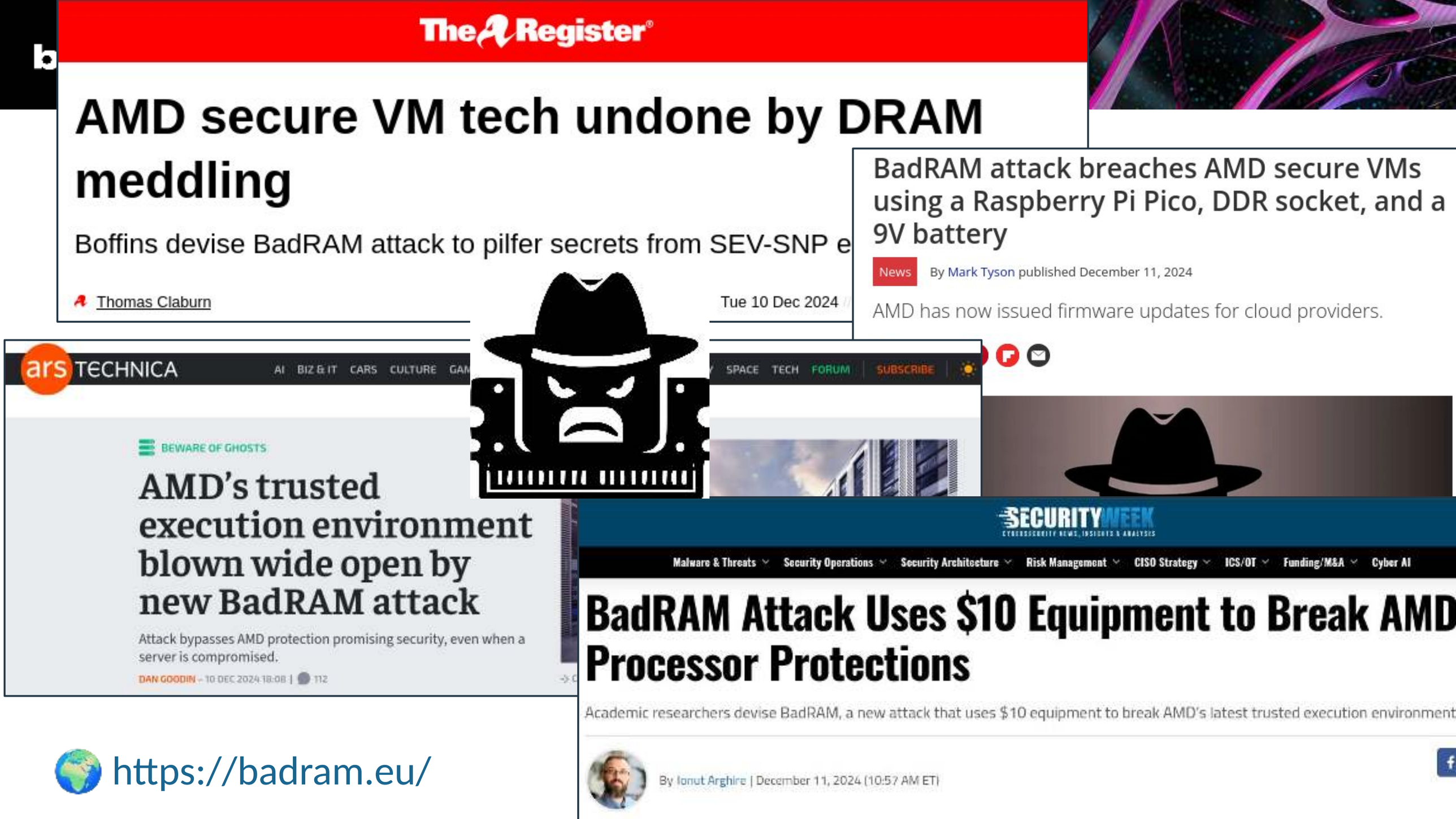

Demo
Replaying encrypted memory

***** AMD SEV-SNP Victim VM *****

```
Initialized 64-byte memory buffer
 ==> Virtual address:  0x7f485c6e7000
 ==> Physical address: 0x26f5e000

Buffer initialized to zero:
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Capture the ciphertext, then press enter
```

***** Privileged Software Attacker *****

```
Translating victim address
 ==> Guest physical address: 0x26f5e000
 ==> Host physical address:  0x18415e000

Calculating memory alias
 ==> Original address: 0x18415e000
 ==> Alias address:    0x58415e000

Press Enter to capture ciphertext
```



https://badram.eu/

Vendor Response: Boot-Time Firmware Mitigations

Undermining Integrity Features of SEV-SNP with Memory Aliasing

AMD ID: AMD-SB-3015
Potential Impact: Loss of Integrity
Severity: Medium

Summary

A team of researchers has reported to AMD that it may be possible to modify serial presence detect (SPD) metadata to make an attached memory module appear larger than it is, potentially allowing an attacker to overwrite physical memory.

Guest Attestation Report [Attestation method for Guest VM]

ATTESTATION_REPORT Structure PLATFORM_INFO field in Byte offset 0h bit 5 contains indication that the mitigation has been applied and confirmed.

Byte Offset	Bits	Name	Description
00h	63:6	-	Reserved.
	5	ALIAS_CHECK_COMPLETE	Indicates that alias detection has completed since the last system reset and there are no aliasing addresses. Resets to 0.

https://www.amd.com/en/resources/product-security/bulletin/amd-sb-3015.html


You don't trust me?

Understanding Memory Aliasing Firmware Mitigations

Idea: Scan for memory aliases at boot time

RISC-V implementation:

- Naive detection: 0.753 *s*


```
for each protected page A do:
  write_mem(A, marker)
  for each page B != A do:
    if read_mem(B) == marker:
      terminate system
    end if
  end for
end for
```

Understanding Memory Aliasing Firmware Mitigations

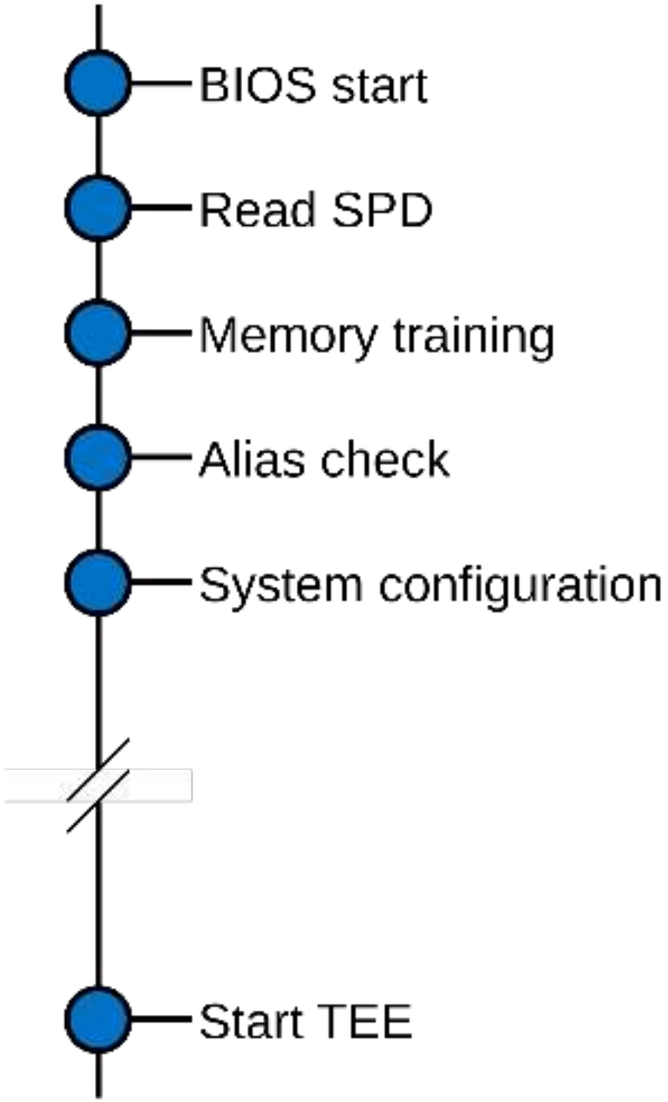
Idea: Scan for memory aliases at boot time

RISC-V implementation:

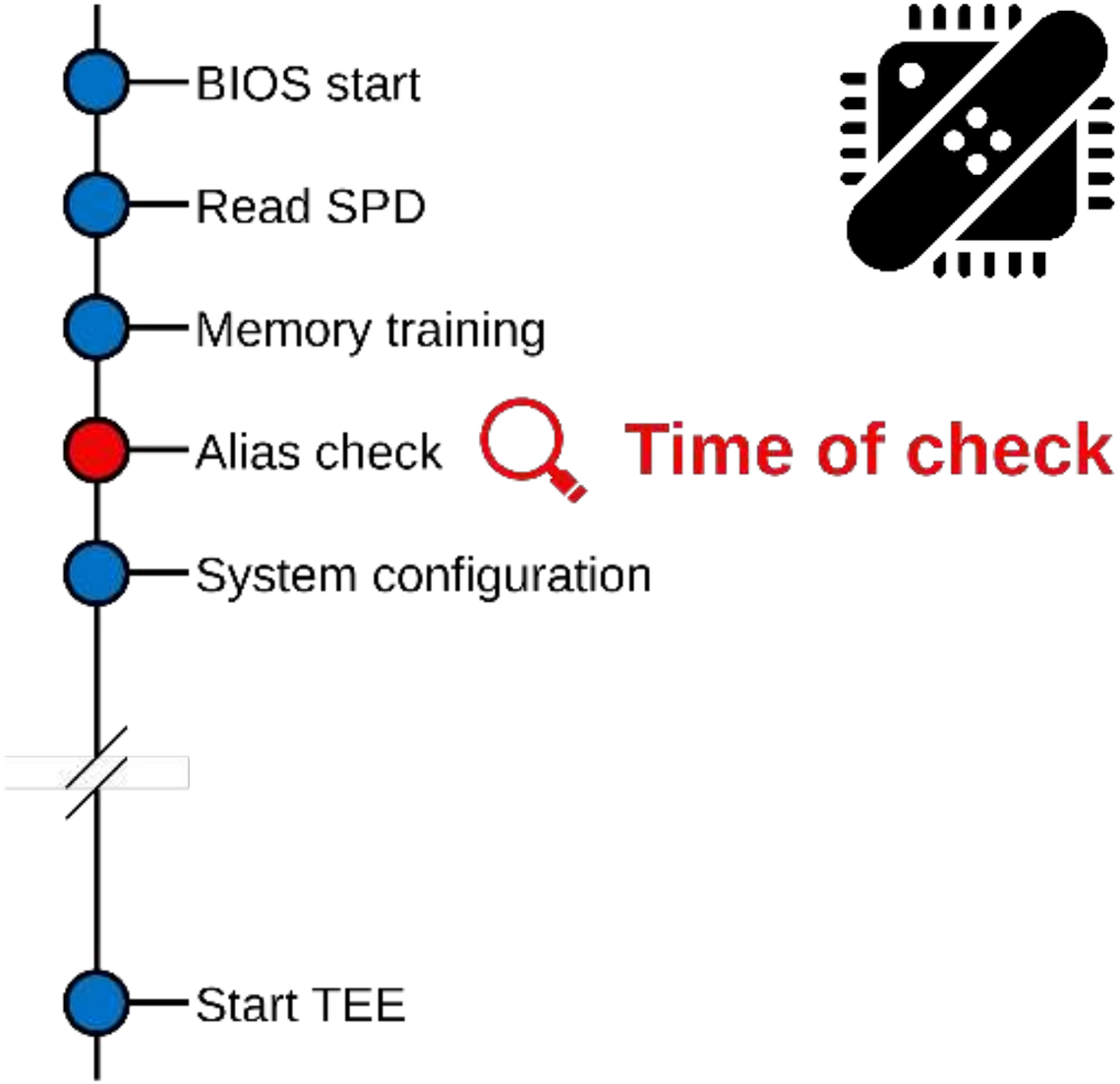
- Naive detection: 0.753 *s*
- Opt. detection: 5.687 *µs*


```
for each protected page A do:
  write_mem(A, marker)
  for each DRAM addr bit i do:
    B = A ^ i
    if read_mem(B) == marker:
      terminate system
    end if
  end for
end for
```

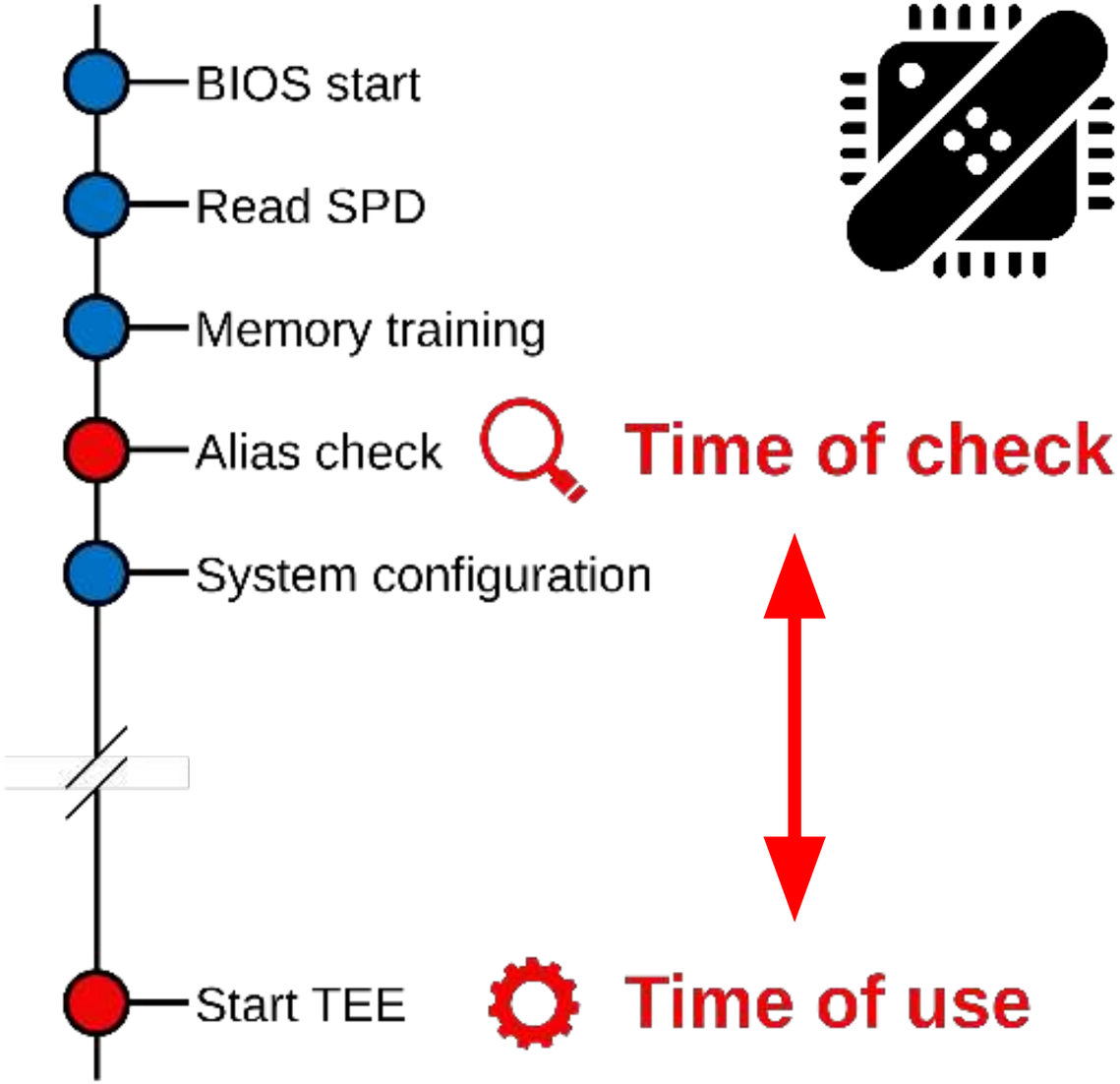
Idea: Bypassing Boot-Time Aliasing Checking?



Idea: Bypassing Boot-Time Aliasing Checking?

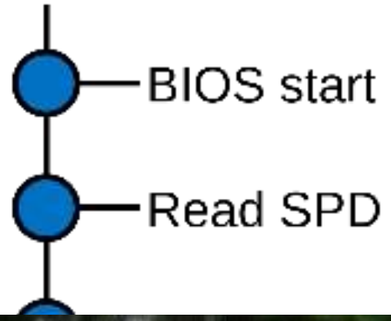


Idea: Bypassing Boot-Time Aliasing Checking?



What if we can introduce aliases at runtime (post-boot)?

Idea: Bypassing Boot-Time Aliasing Checking?



What if we can introduce aliases at runtime (post-boot)?

Interfering at Runtime: Commercial DRAM Interposers?






Add Inquiry Basket to Compare

Genuine New MW-Keysight U4972A DDR4 Protocol Debugging and Analysis Solution Logic Analyzers Factory Wholesale Price

US$782,016.00
1 Set (MOQ)

Send Inquiry | Chat Now

Product Details

Customization:	Available
After-sales Service:	12 Months
Warranty:	12 Months

Shenzhen Leading International Trading Co., Ltd. >

Gold Member Since 2024

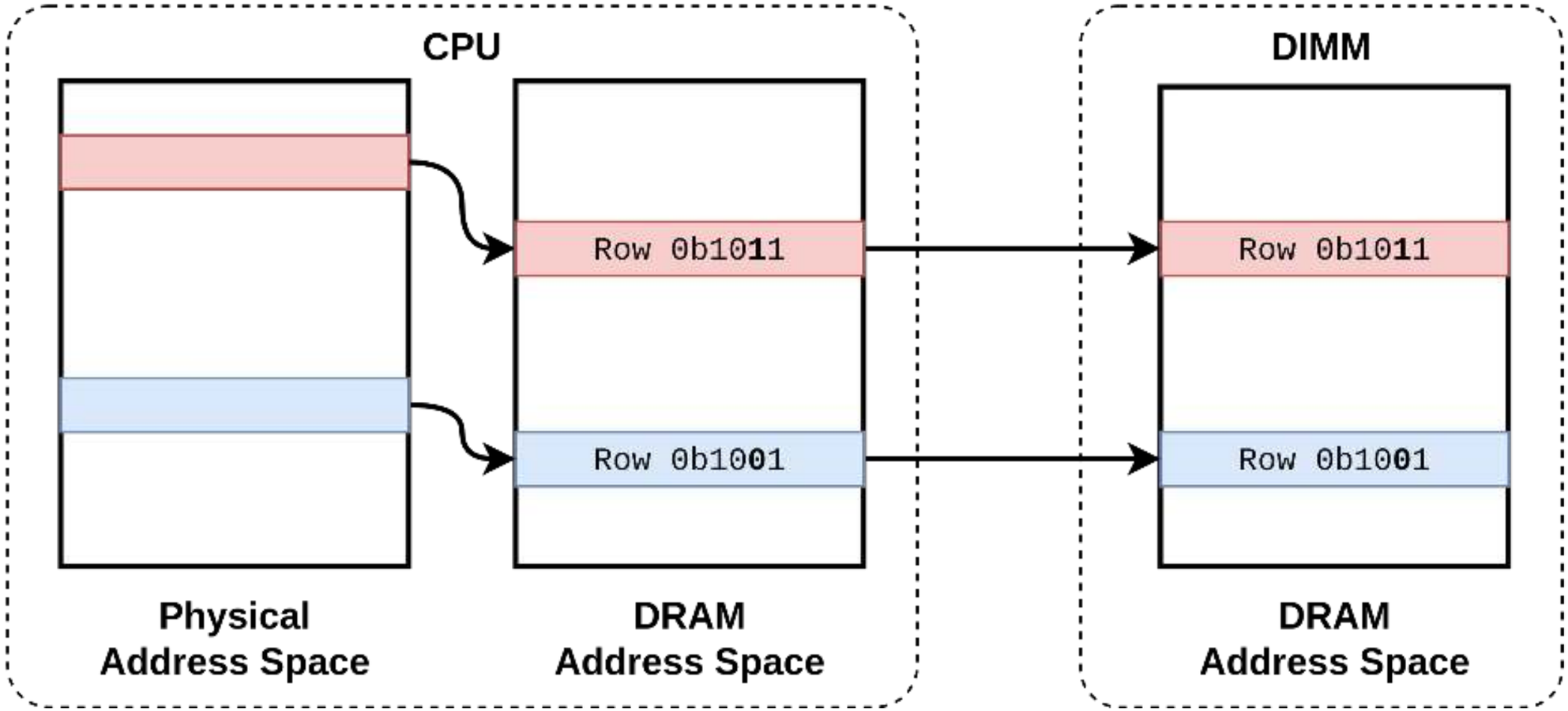
Audited Supplier

Tampering with Addressing at Runtime

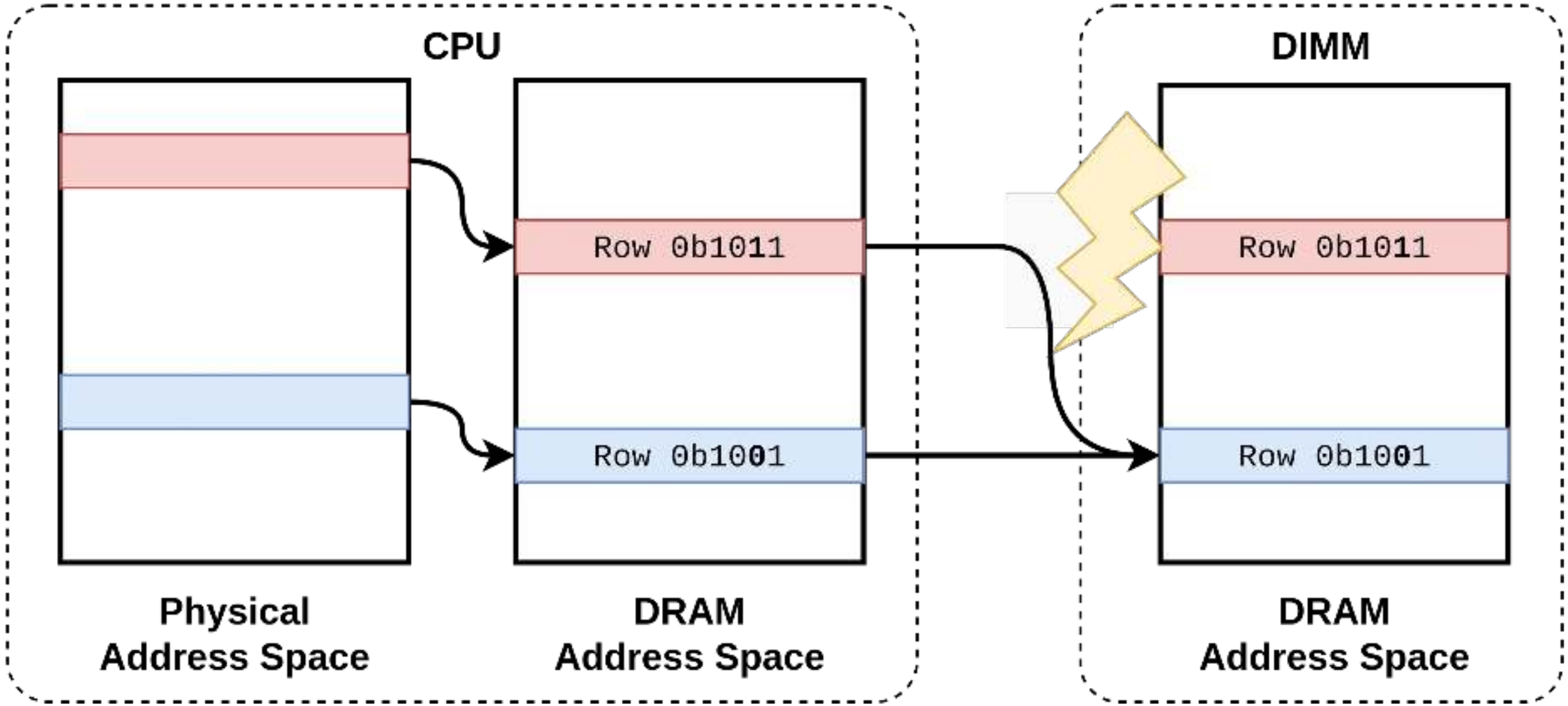
Road Map

1. Modern memory encryption designs and where to find them
2. BadRAM: What if your DRAM lies to you?
3. **Battering Ram: Low-cost physical interposer attacks**
4. Conclusions and takeaways

Tampering with Addressing at Runtime



Tampering with Addressing at Runtime



Tampering with Addressing at Runtime

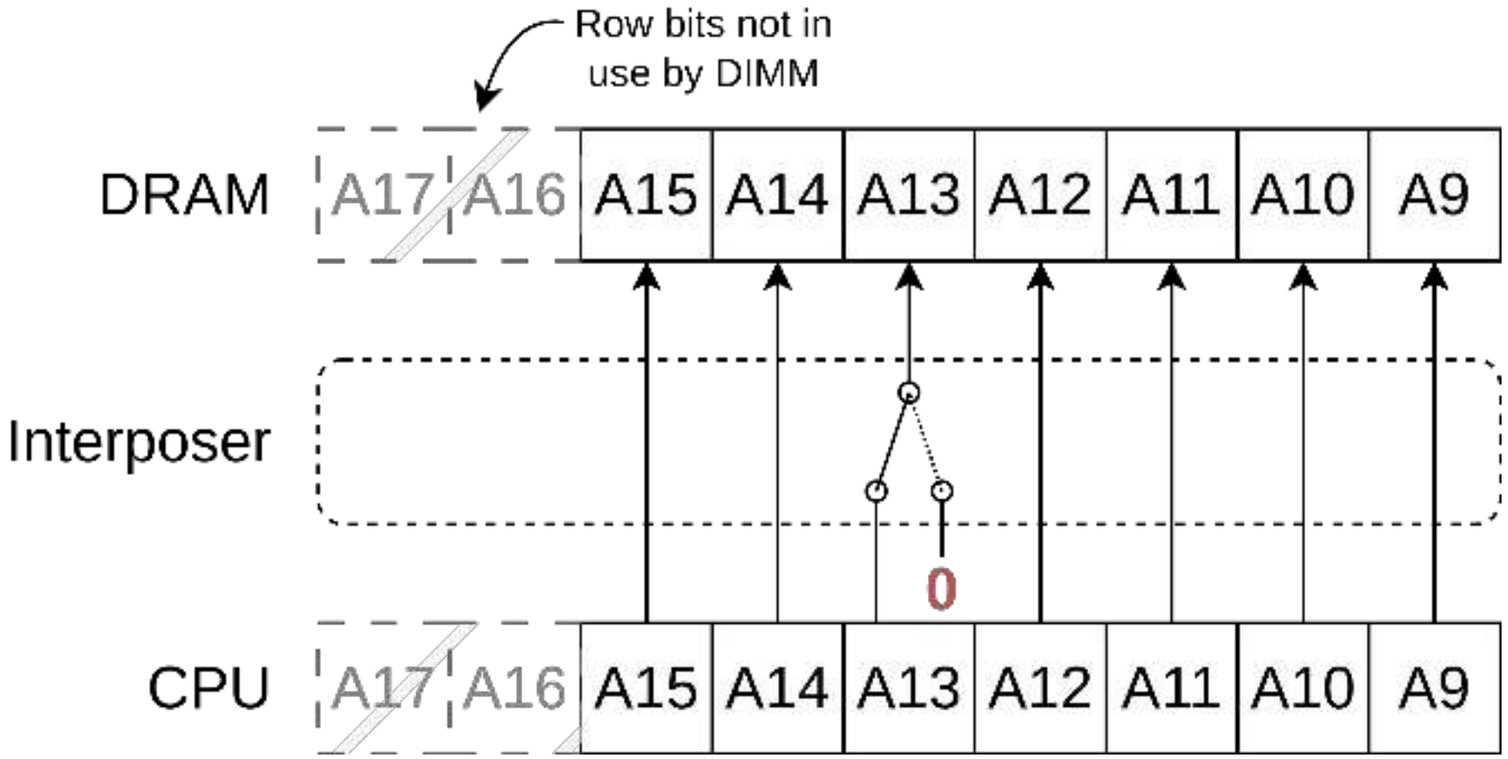
Tampering with Addressing at Runtime

	$\overline{CS}$	$\overline{ACT}$	BG	BA	CID	A17	A16	A15	A14	A13	A12	A11	A10	A9-0
ACT	L	L	Bank		CID	Row								
MRS	L	H	Reg		V	RFU	L	L	L	OP				
REF	L	H	V	V	CID	V	L	L	H	V	V	V	V	V
PRE	L	H	Bank		CID	V	L	H	L	V	V	V	L	V
PREA	L	H	V	V	CID	V	L	H	L	V	V	V	H	V
RFU	L	H	RFU				L	H	H	RFU				
WR	L	H	Bank		CID	V	H	L	L	V	$\overline{BC}$	V	AP	CA
RD	L	H	Bank		CID	V	H	L	H	V	$\overline{BC}$	V	AP	CA
ZQCL	L	H	V	V	V	V	H	H	L	V	V	V	H	V
ZQCS	L	H	V	V	V	V	H	H	L	V	V	V	L	V
NOP	L	H	V	V	V	V	H	H	H	V	V	V	V	V
DES	H	X	X	X	X	X	X	X	X	X	X	X	X	X

Tampering with Addressing at Runtime

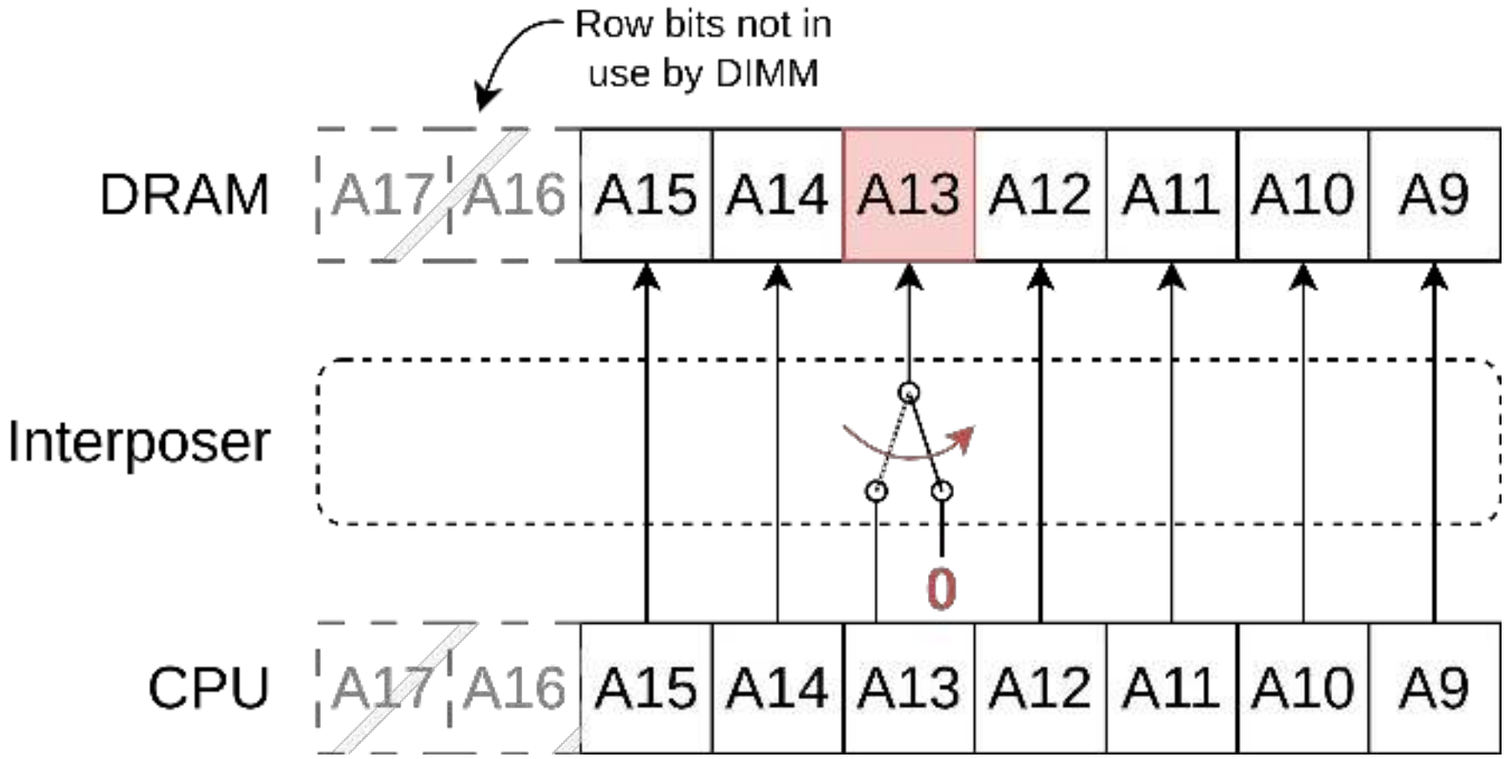
	$\overline{CS}$	$\overline{ACT}$	BG	BA	CID	A17	A16	A15	A14	A13	A12	A11	A10	A9-0
ACT	L	L	Bank		CID					Row				
MRS	L	H	Reg		V	RFU	L	L	L			OP		
REF	L	H	V	V	CID	V	L	L	H	V	V	V	V	V
PRE	L	H	Bank		CID	V	L	H	L	V	V	V	L	V
PREA	L	H	V	V	CID	V	L	H	L	V	V	V	H	V
RFU	L	H	RFU				L	H	H			RFU		
WR	L	H	Bank		CID	V	H	L	L	V	$\overline{BC}$	V	AP	CA
RD	L	H	Bank		CID	V	H	L	H	V	$\overline{BC}$	V	AP	CA
ZQCL	L	H	V	V	V	V	H	H	L	V	V	V	H	V
ZQCS	L	H	V	V	V	V	H	H	L	V	V	V	L	V
NOP	L	H	V	V	V	V	H	H	H	V	V	V	V	V
DES	H	X	X	X	X	X	X	X	X	X	X	X	X	X

Tampering with Addressing at Runtime



- Boot time: **Inactive**
  - Passes checks

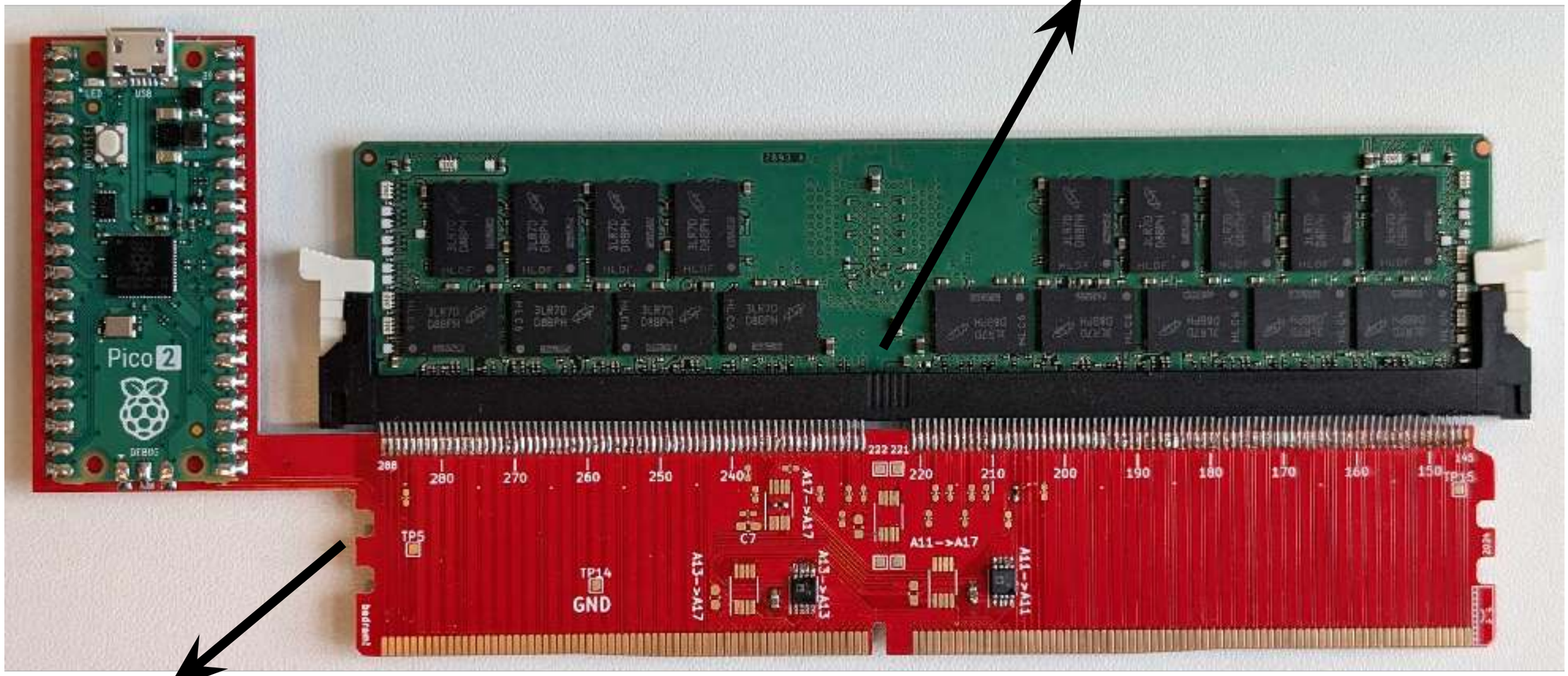
Tampering with Addressing at Runtime



- Boot time: **Inactive**
  - Passes checks
- Attack: **Active**
  - Dynamically switch to GND

Tampering with Addressing at Runtime

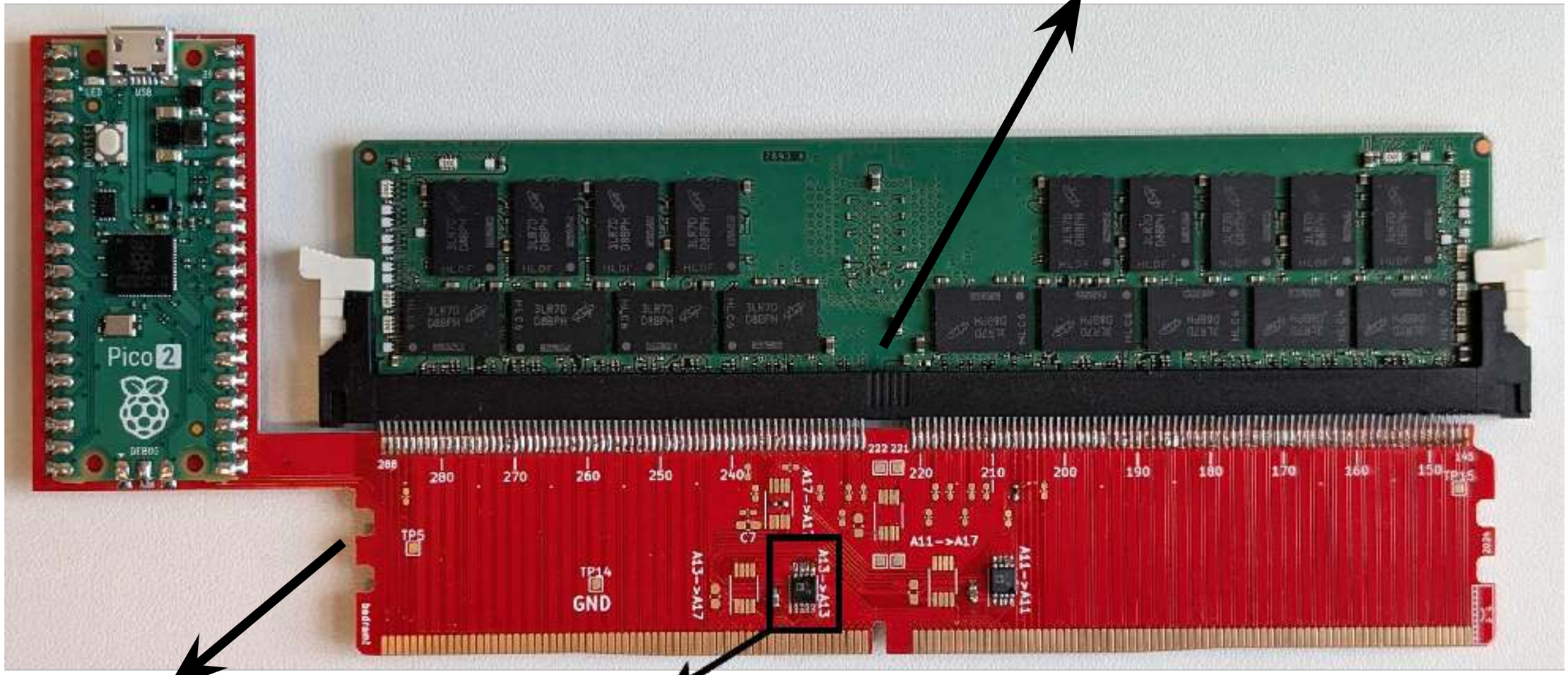
DDR4 connector: $16



Custom PCB: $18

Tampering with Addressing at Runtime

DDR4 connector: $16



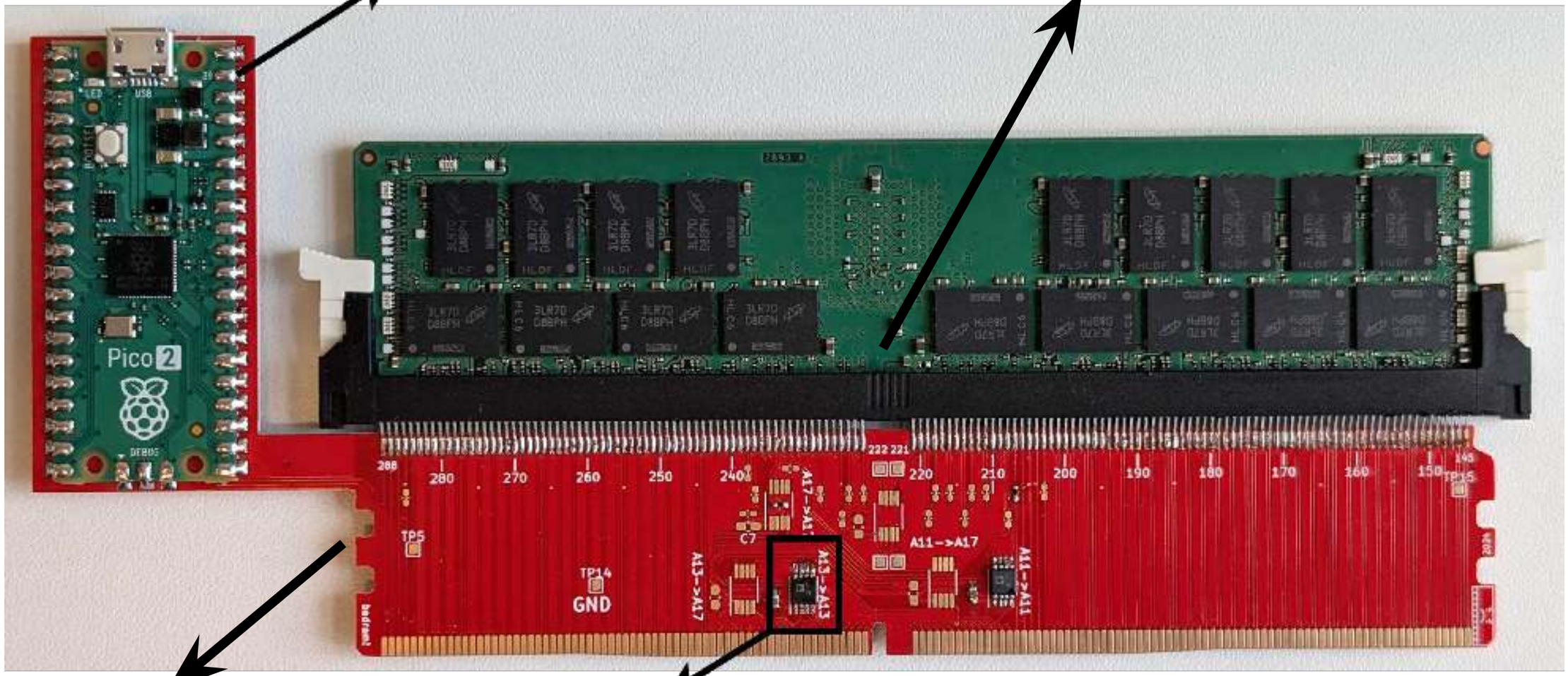
Custom PCB: $18

Analog Switch (ADG902) $4

Tampering with Addressing at Runtime

Microcontroller (RPi Pico) $4

DDR4 connector: $16



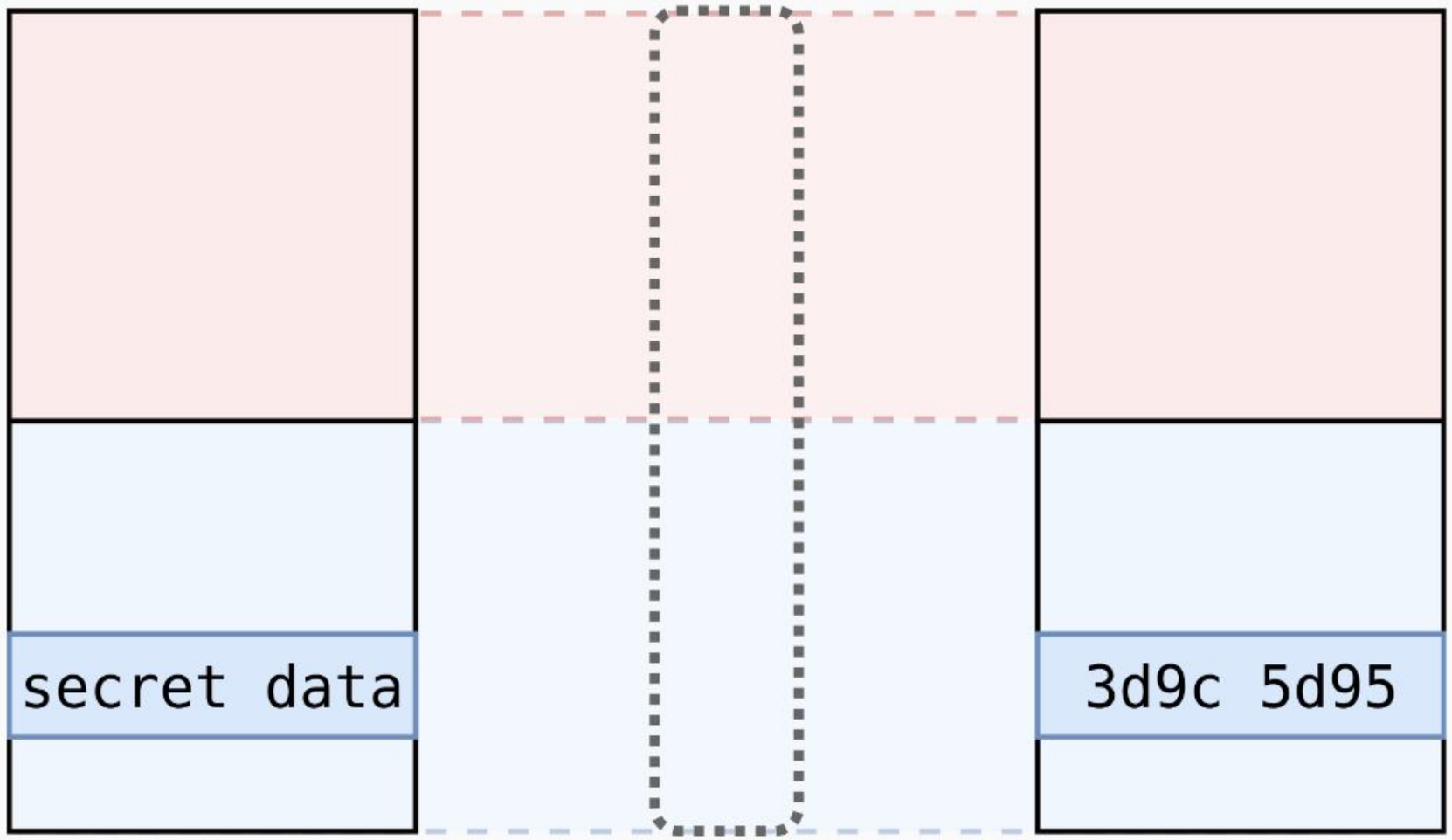
Custom PCB: $18

Analog Switch (ADG902) $4

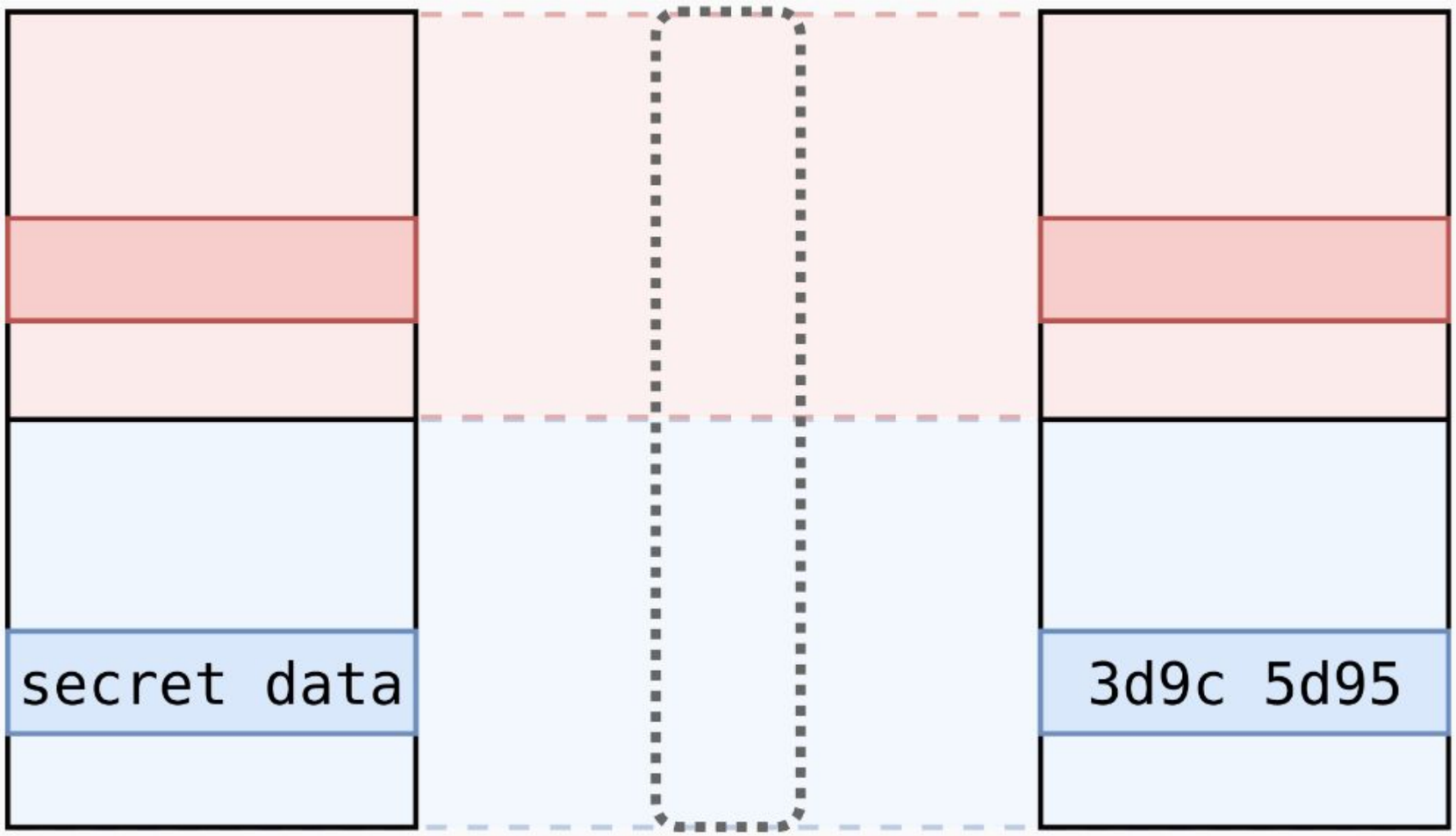
Tampering with Addressing at Runtime

- **Stability**
  - Works at any DDR4 frequency
- **Feasibility**
  - Deterministic memory aliasing
  - Entirely invisible to CPU
- **Limitations**
  - Unstable memory regions
  - Memory interleaving complicates attacks

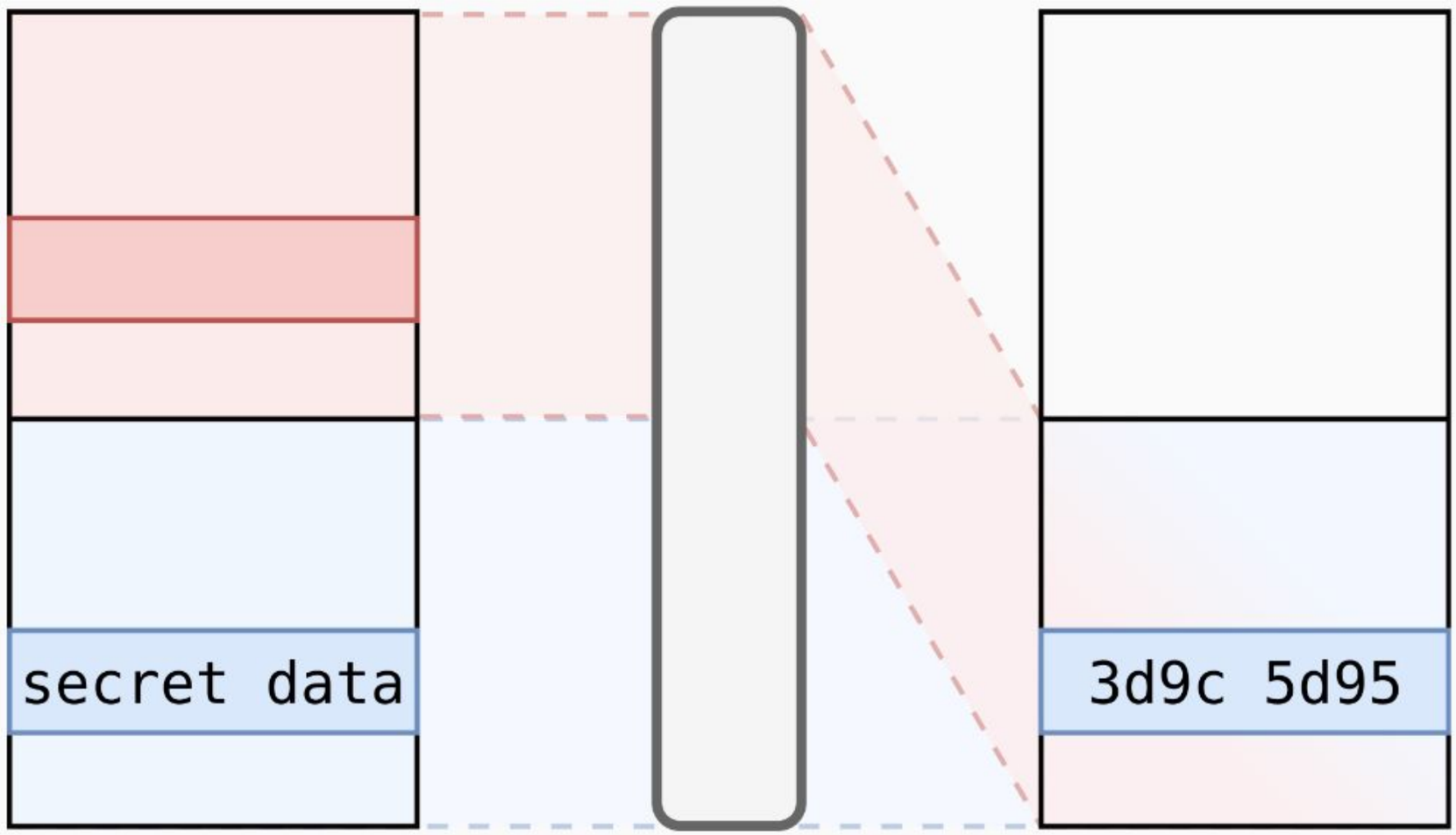
Scalable SGX Plaintext Access



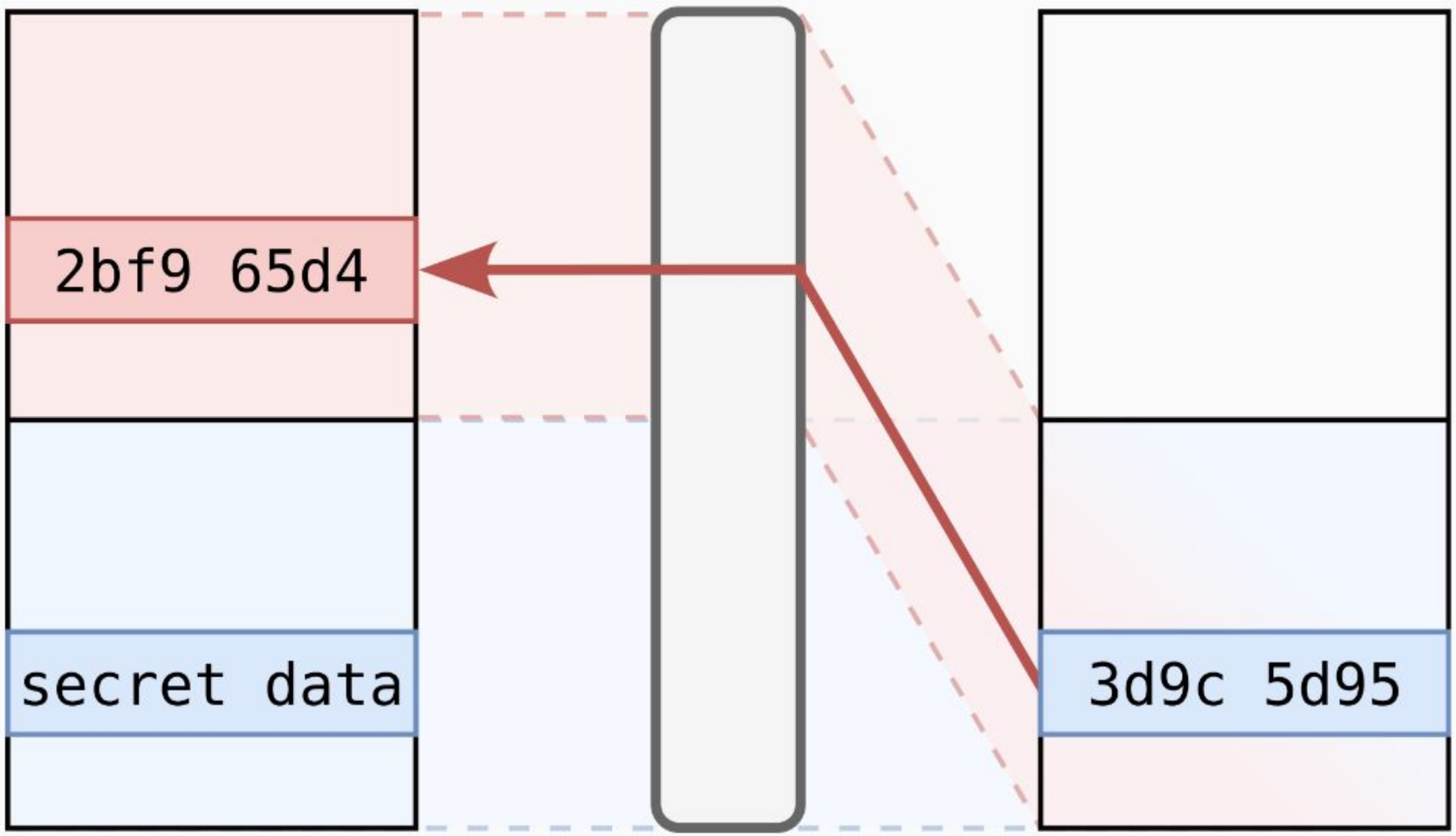
Scalable SGX Plaintext Access



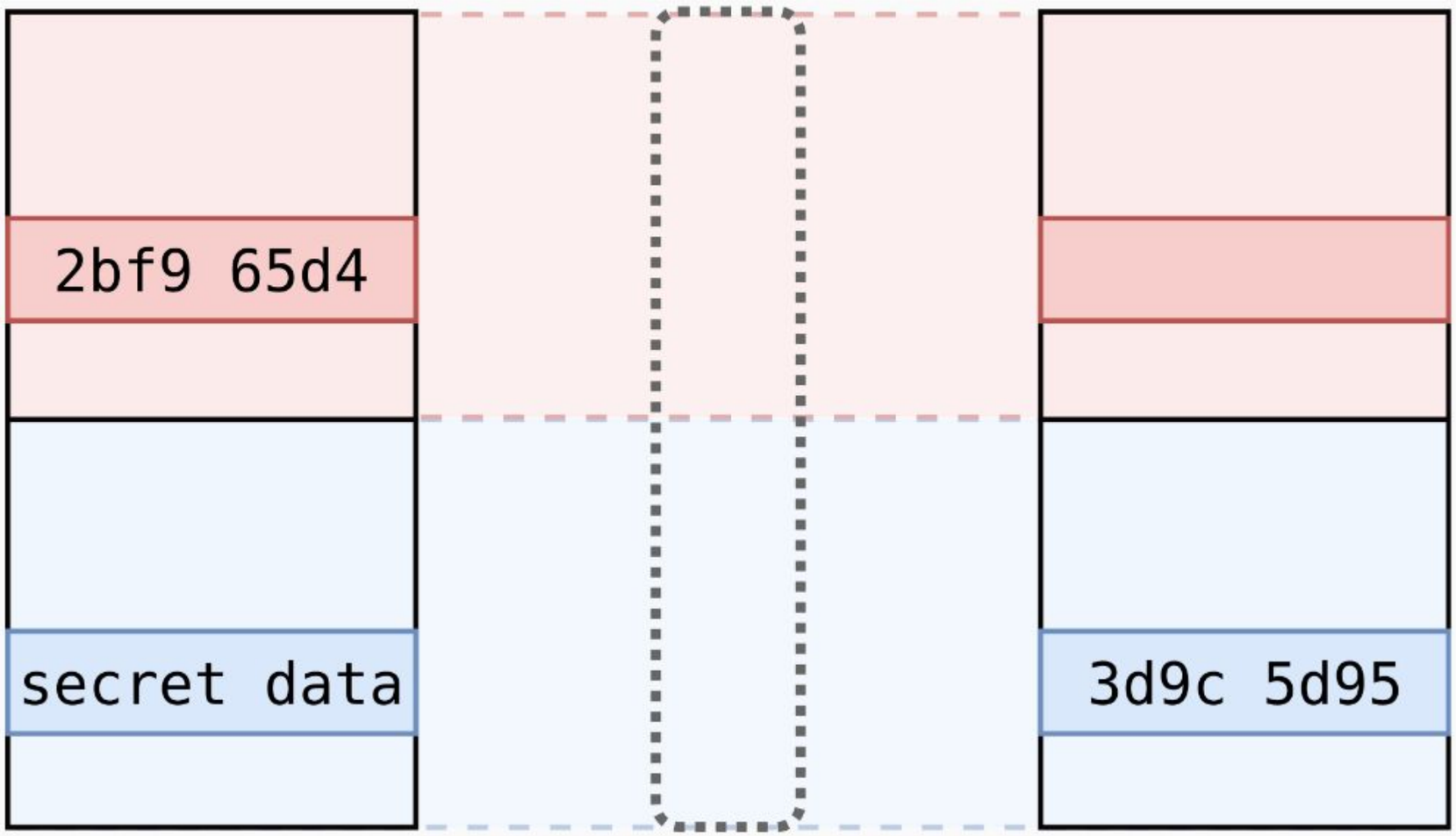
Scalable SGX Plaintext Access



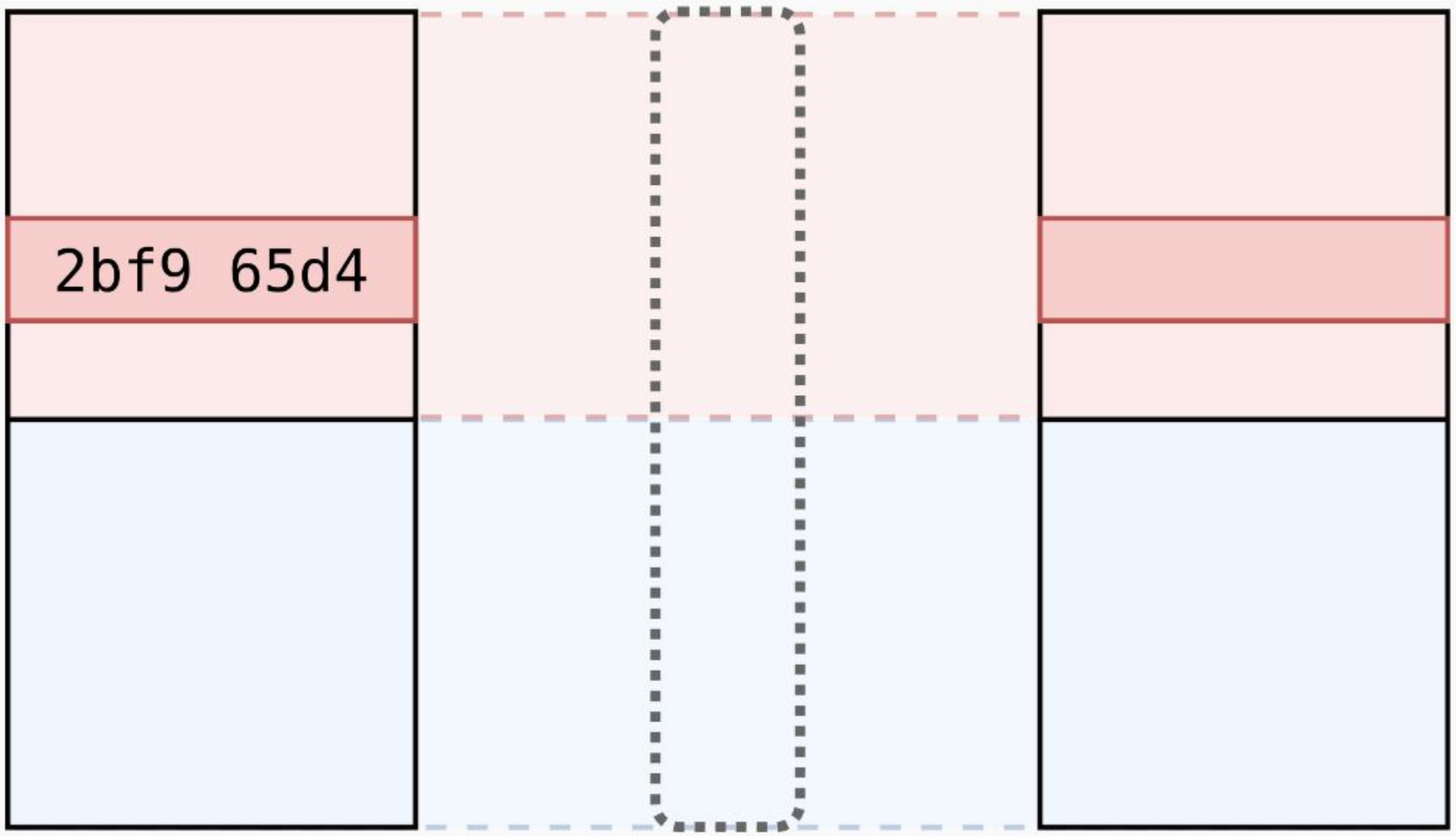
Scalable SGX Plaintext Access



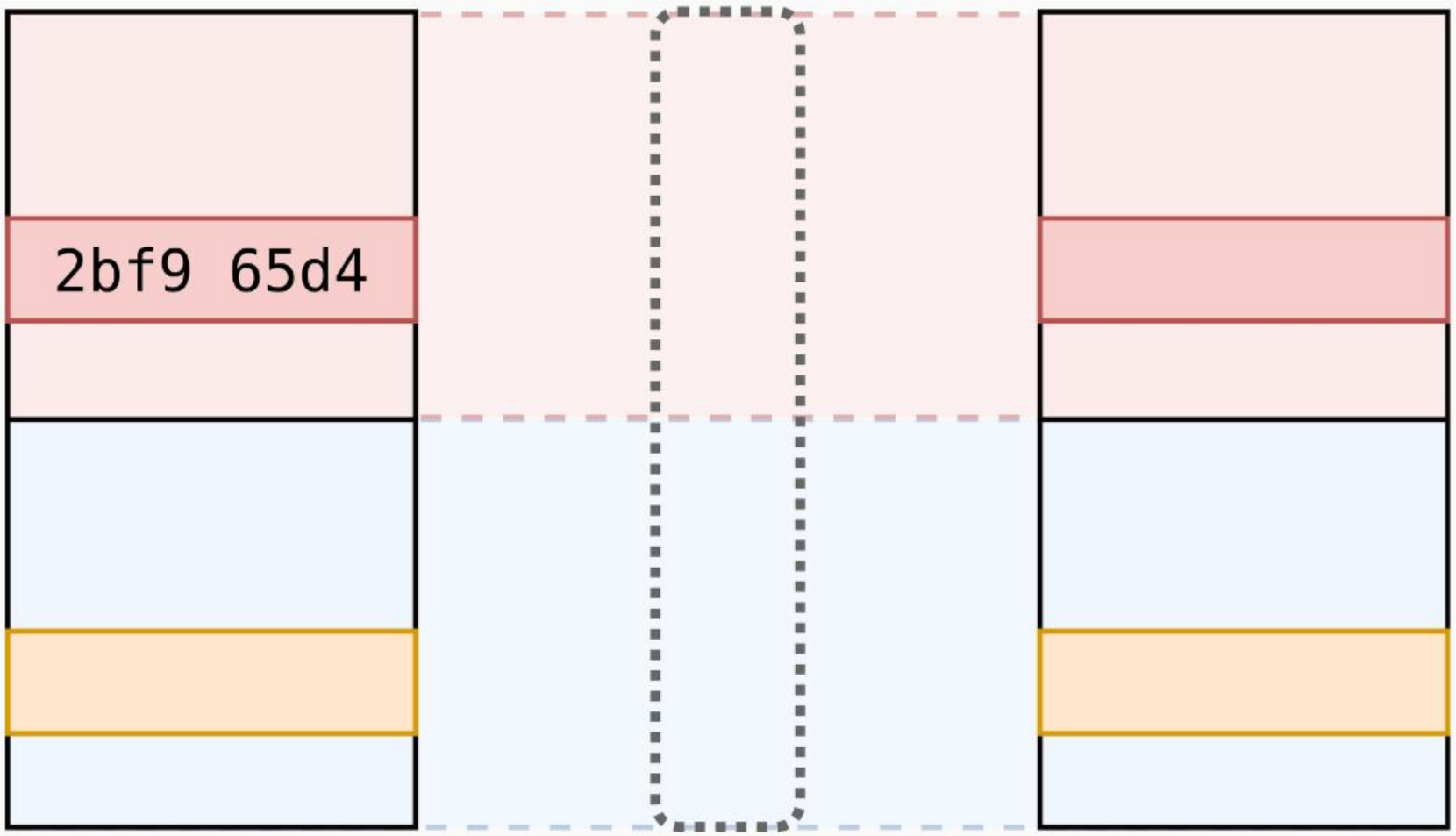
Scalable SGX Plaintext Access



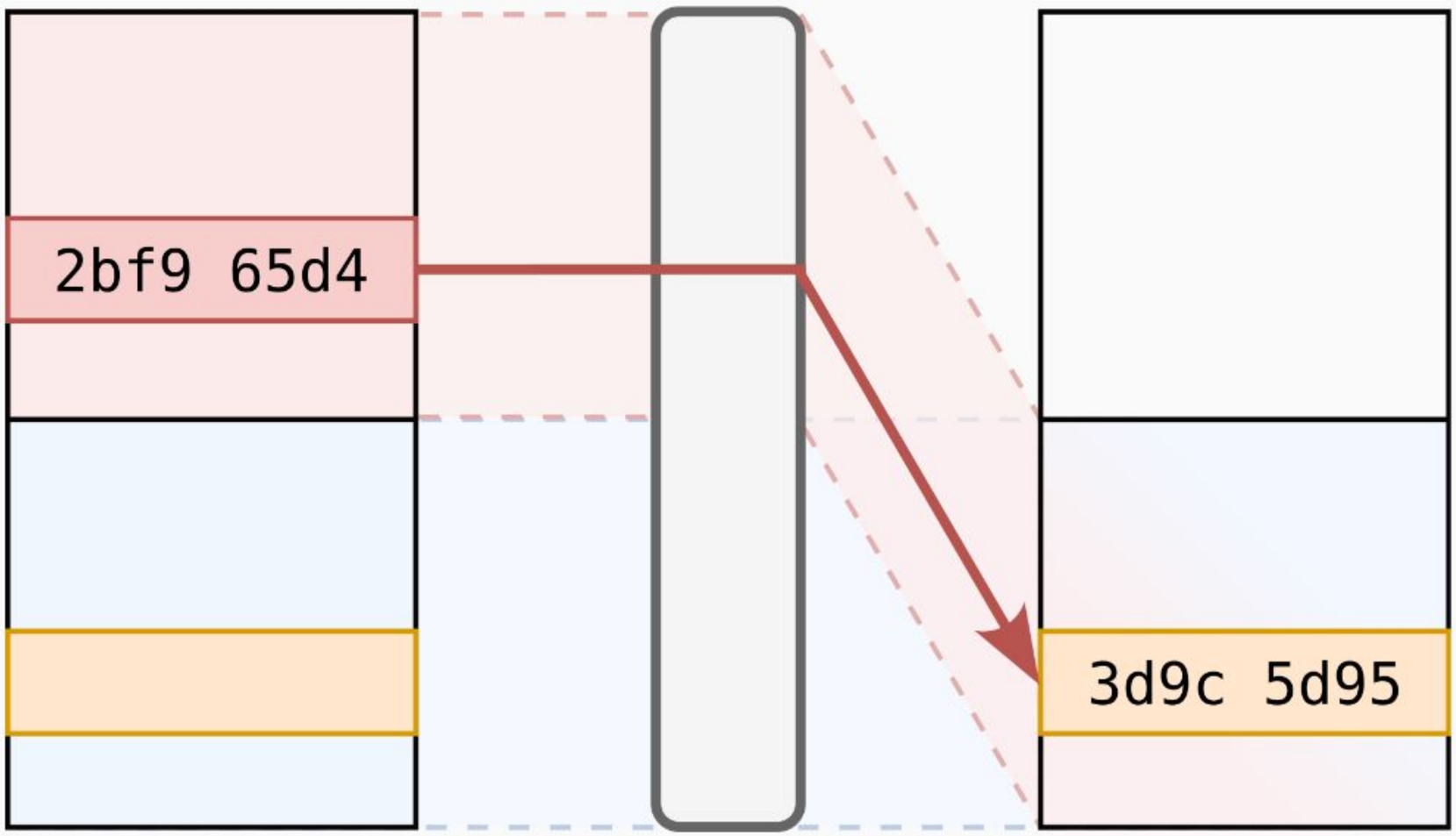
Scalable SGX Plaintext Access



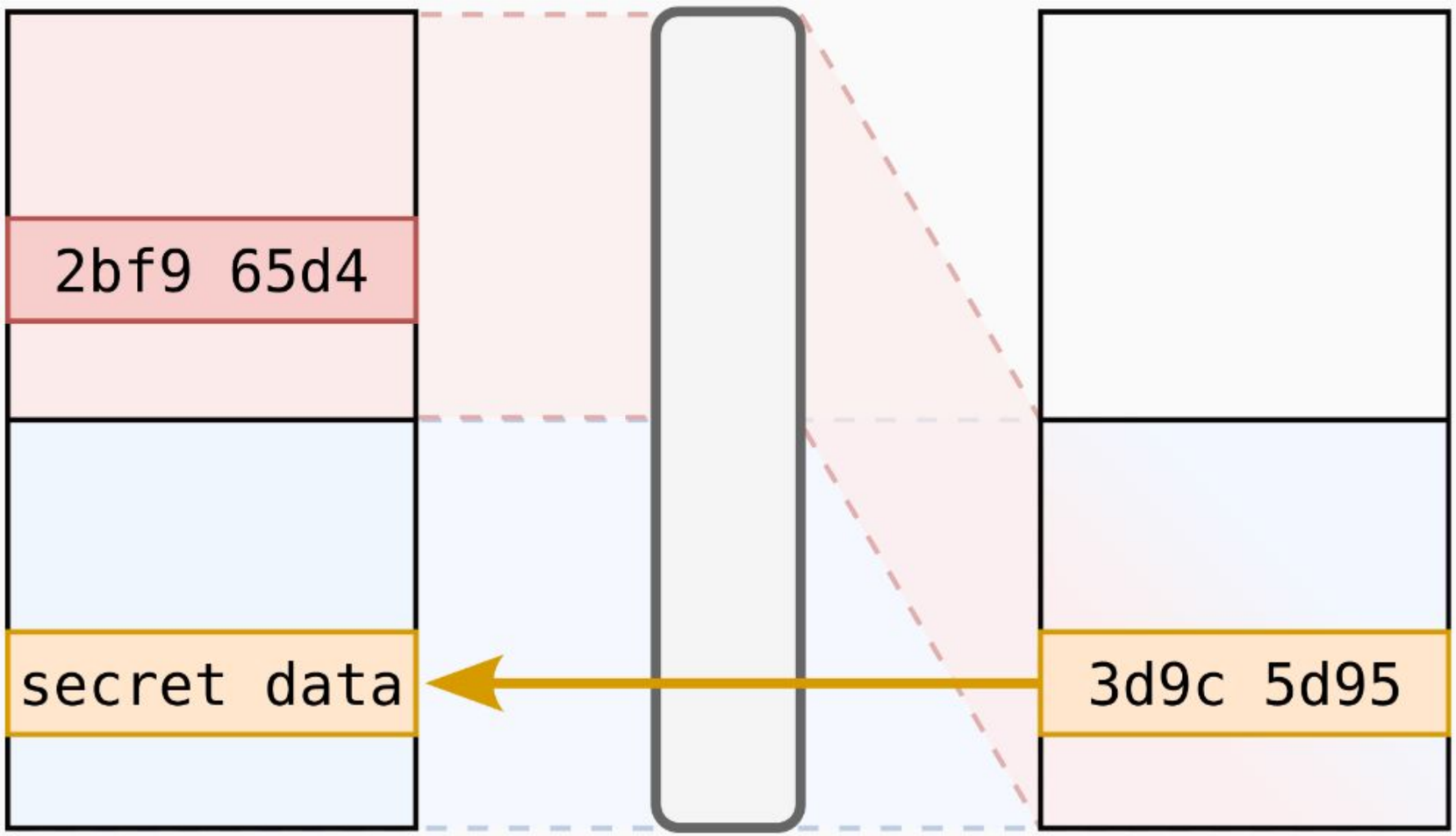
Scalable SGX Plaintext Access



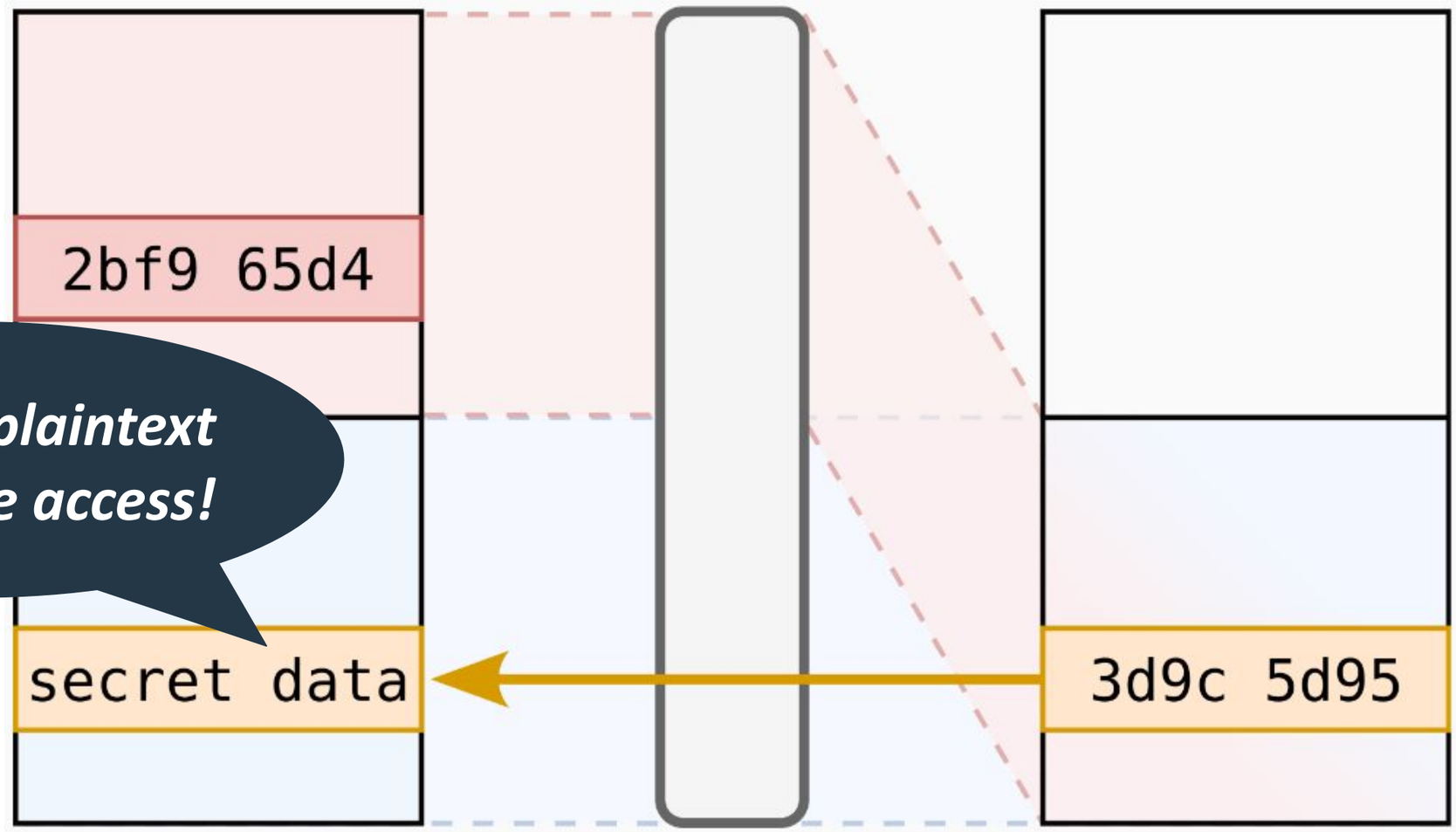
Scalable SGX Plaintext Access



Scalable SGX Plaintext Access

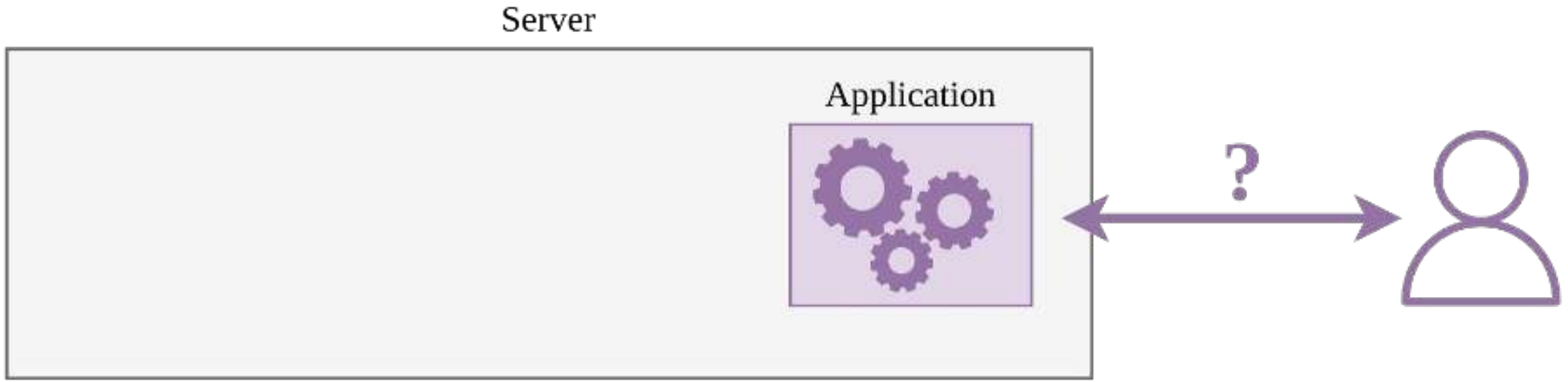


Scalable SGX Plaintext Access

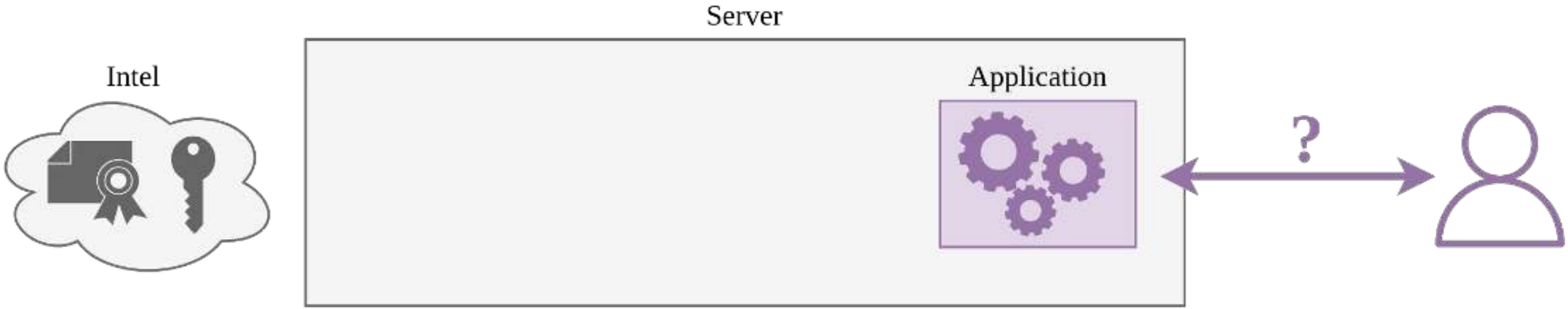


Arbitrary plaintext read/write access!

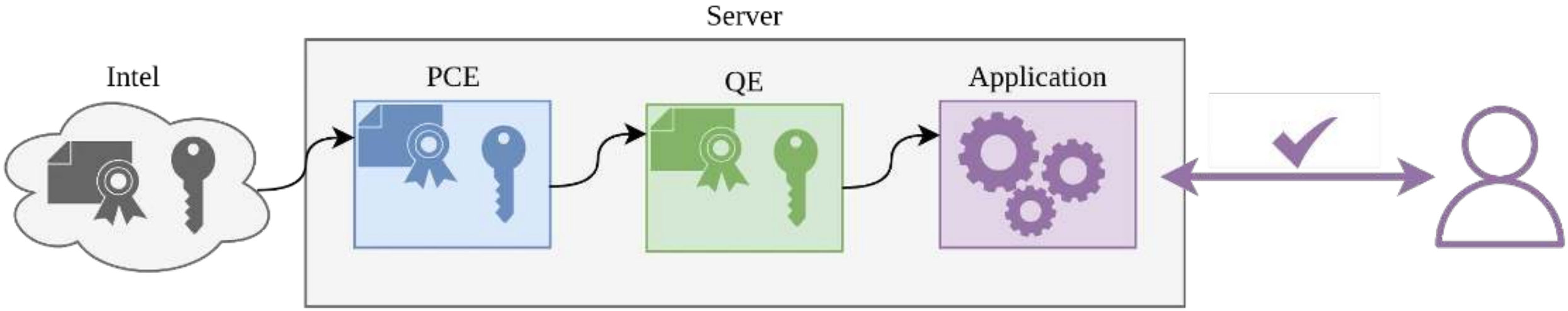
Extracting Scalable SGX Secrets



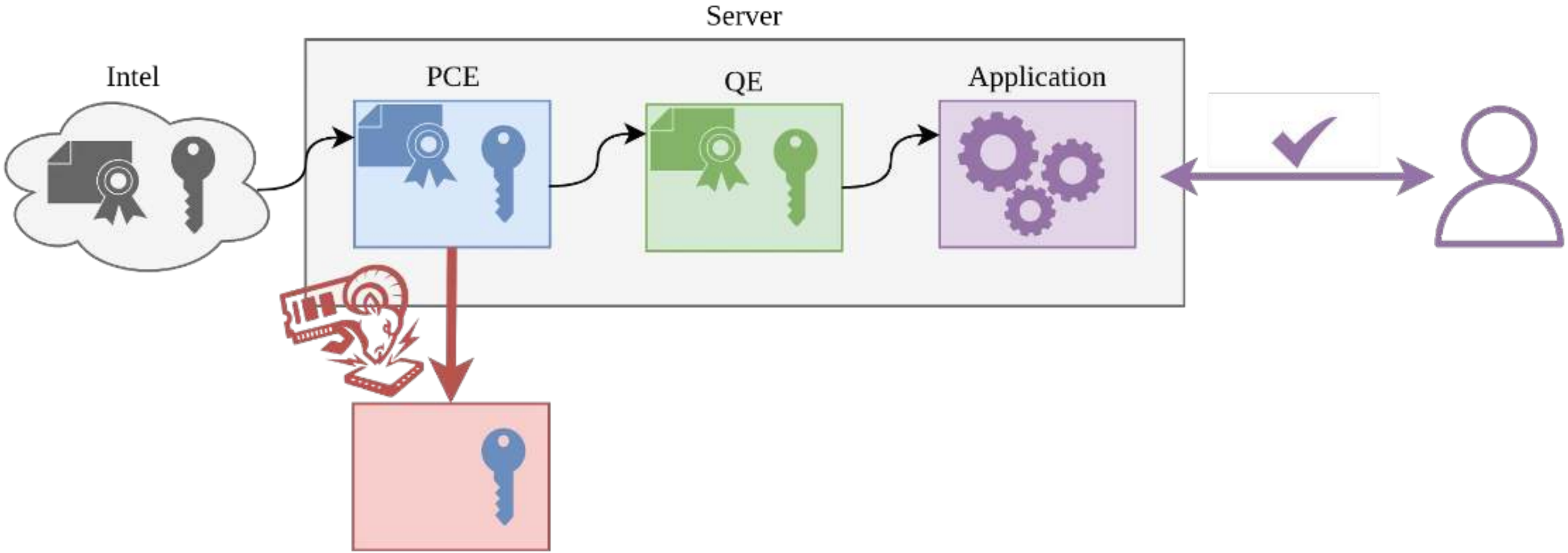
Extracting Scalable SGX Secrets



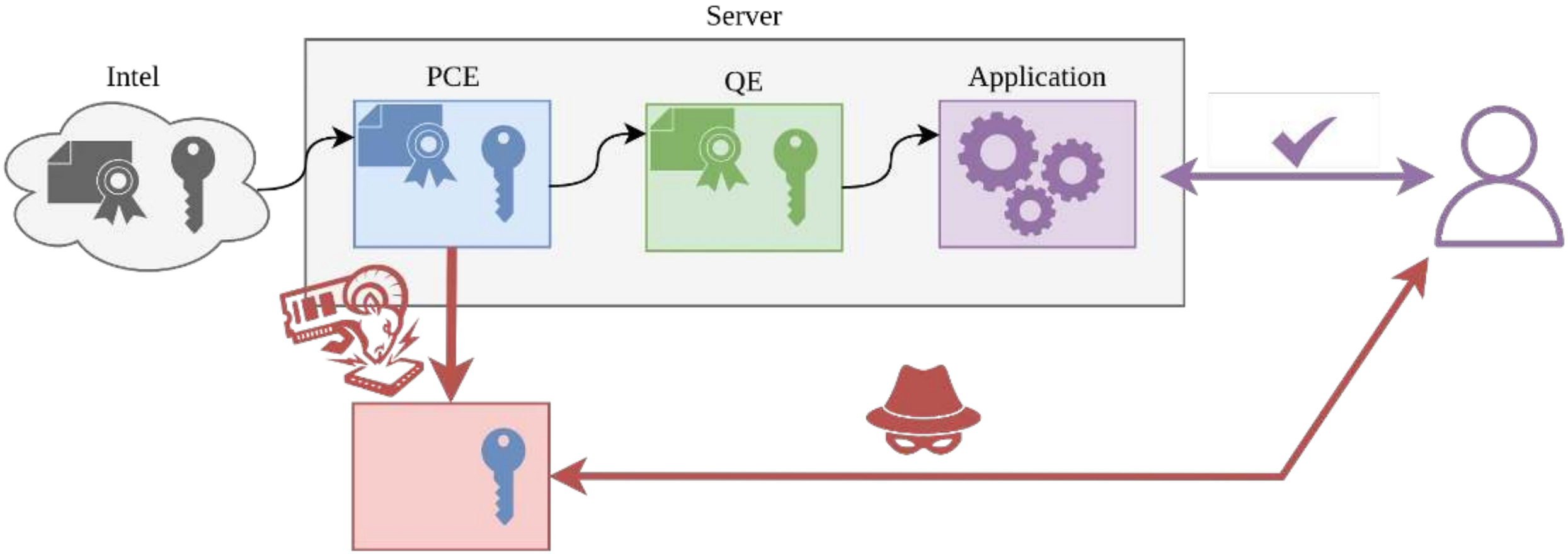
Extracting Scalable SGX Secrets



Extracting Scalable SGX Secrets



Extracting Scalable SGX Secrets




Demo
Arbitrary plaintext access on Intel Scalable SGX

tryagain@tryagain-X12: /hotstringram/scalable-ogv-attacks/pcr/$

black BRIEF

ars TECHNICA AI BIZ & IT CARS CULTURE GAMING HEALTH POLICY SCIENCE SECURITY SPACE TECH FORUM SUBSCRIBE SIGN IN

Intel and AMD trusted enclaves, a foundation for network security, fall to physical attacks

The chipmakers say physical attacks aren't in the threat model. Many users didn't get the memo.

DAN GOODIN – 30 SEPT 2025 22:25 | 67

Cheap Hardware Module Bypasses AMD, Intel Memory Encryption

Researchers built an inexpensive device that circumvents chipmakers' confidential computing protections and reveals weaknesses in scalable memory encryption.

CLOUD SECURITY

Rob Wright, Senior News Direc
November 25, 2025

Battering RAM Attack Breaks Intel and AMD Security Tech With $50 Device

Intel and AMD say the research is not in scope of their threat model because the attack requires physical access to a device.

https://batteringram.eu/

SEV-SNP Physical Memory Aliasing

AMD ID: AMD-SB-3024
Potential Impact: N/A

Summary

Researchers have reported a method for privileged attackers with physical access to a motherboard to potentially compromise confidentiality and integrity of AMD Secure Encrypted Virtualization – Secure Nesting Paging (SEV-SNP) guests.

AMD does not plan to release any mitigations in response to this report because the reported exploit is outside the scope of the published threat model for SEV-SNP, as detailed in Table 1 of the AMD SEV-SNP technical paper.

https://www.amd.com/en/resources/product-security/bulletin/amd-sb-3024.html

More Information on Encrypted Memory Frameworks for Intel Confidential Computing

ID	Updated	Version	
865767	10/27/2025	1.0	Public

In the *Battering RAM* paper, researchers from KU Leuven and University of Birmingham developed a custom interposer to actively alias memory and gain arbitrary read/write access into Intel SGX-protected memory.

Both research teams assume a physical adversary has direct access to the hardware with a memory bus interposer. Both methods can then be used to attack Intel SGX-protected assets, including Intel SGX attestation keys. In a separate disclosure to Intel, Fortanix provided a potential attack that requires a replay-capable physical interposer. Such attacks are outside the scope of the boundary of protection offered by Advanced Encryption Standard-XEX-based Tweaked Codebook Mode with Ciphertext Stealing (AES-XTS) based memory encryption, as originally stated in the 2021 Intel publication Supporting Intel® SGX on Multi-socket Platforms. As it provides limited confidentiality protection, and no integrity or anti-replay protection against attackers with physical capabilities, Intel does not plan to issue a CVE.

https://www.intel.com/content/www/us/en/developer/articles/news/more-information-encrypted-memory-frameworks.html

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Search...

About ANSSI | Regulation | Scientific standing | Digital Risk Management | Incident Response

Accueil > Liste Des Publications > Technical Position Paper on Confidential Computing

Technical Position Paper on Confidential Computing

In this position paper, ANSSI outlines its views on Confidential Computing. It recalls the attack models that Confidential Computing purports addressing, its main security mechanisms and their current limitations. It also provides guidelines to Cloud Service Providers and other companies developing security pro

As mentionned before, Confidential Computing is often presented by commercial providers as a solution to run remote workloads with the same level of confidentiality and integrity as a local setup, *i.e.* resistant to a physical attack. However, **physical attacks are explicitly out-of-scope of the security target defined by hardware vendors**. This means in particular that if a user is concerned about a cloud-provider conducting targeted attacks, instead of relying on a Confidential Computing approach they need to switch to a cloud-provider they trust, *i.e.* with strong counterparts or control capabilities, or use their own hardware with physical security protection measures. Likewise, the security of Confidential Computing assumes an uncompromised Manufacturer TCB: manufacturer and supply-chain attackers, including state-level ones, are thus explicitly out-of-scope.

https://cyber.gouv.fr/en/publications/technical-position-paper-confidential-computing

$10M Idea: Confidential Computing in Space…

SpaceComputer - 天机 | sp/acc ✓
@SpaceComputerIO

🛰️ Total raise: $10M 🛰️

The seed round turns breakthrough research into a reality → live constellation of satellites in orbit.

Leads → @Maven11Capital & @lattice_fund

Participants → @Superscrypt, @etherealvc, @arbitrum, @starship_vc, @nascent, @hash3xyz, @thebbfund, @ambergroup_io, @cmsholdings, @BitscaleCapital, @HashKey_Capital, @Offchain, @bodhi_ventures, @ChorusOne, @imTokenOfficial, @MoonrockCapital, @Wise3Ventures, @encodeclub and others.

Angels → @WarcMeinstein, @JasonYanowitz, @ameensol, @Not3Lau_Capital, @will__price, @joonian, @crypto_han, @melynx and others.

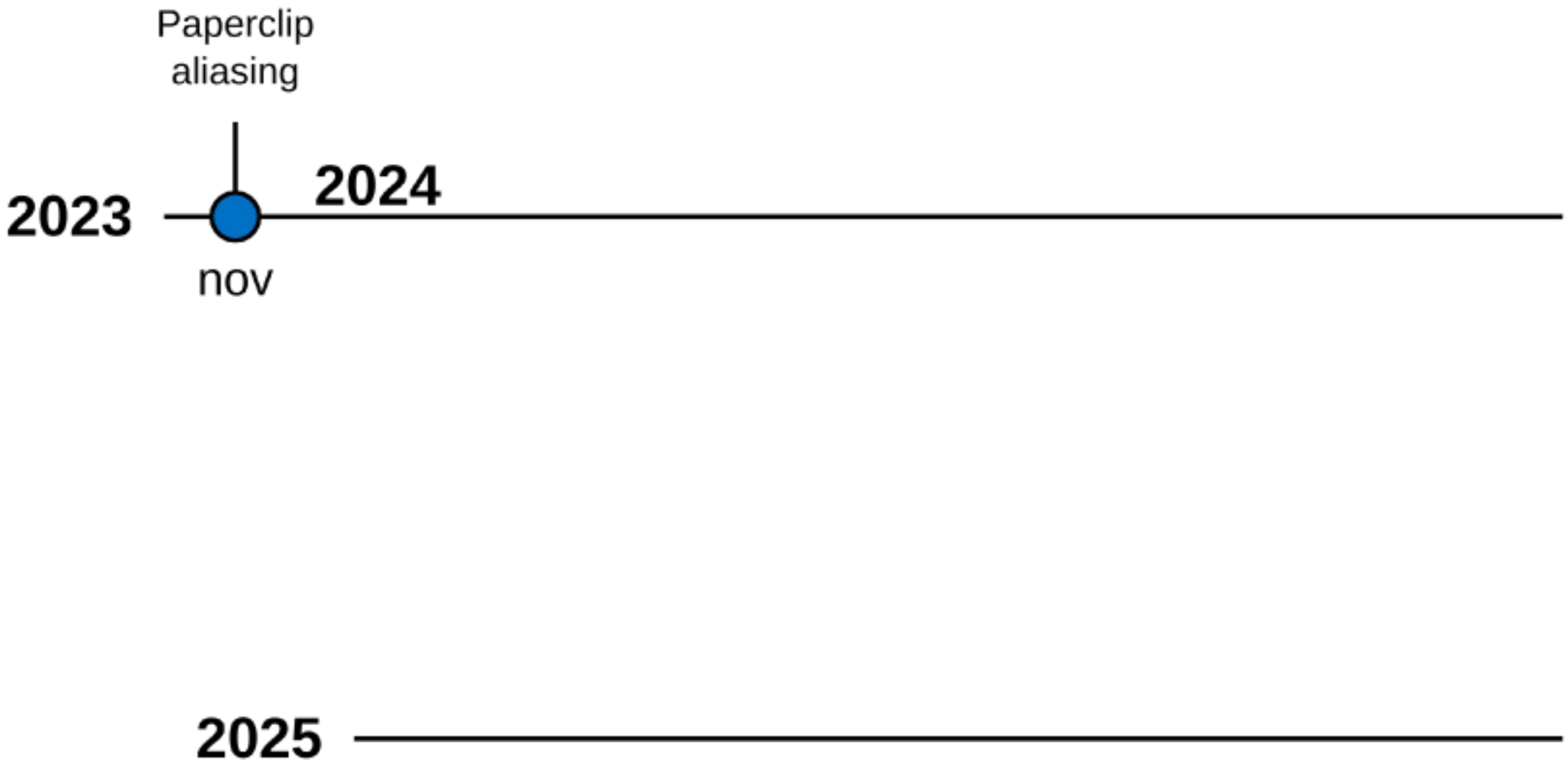
> **The Defiant** ✓ @DefiantNews · Nov 26
> EXCLUSIVE: @SpaceComputerIO has raised $10M in a seed round led by @Maven11Capital and @lattice_fund to run secure computing for blockchains from space.

Timeline

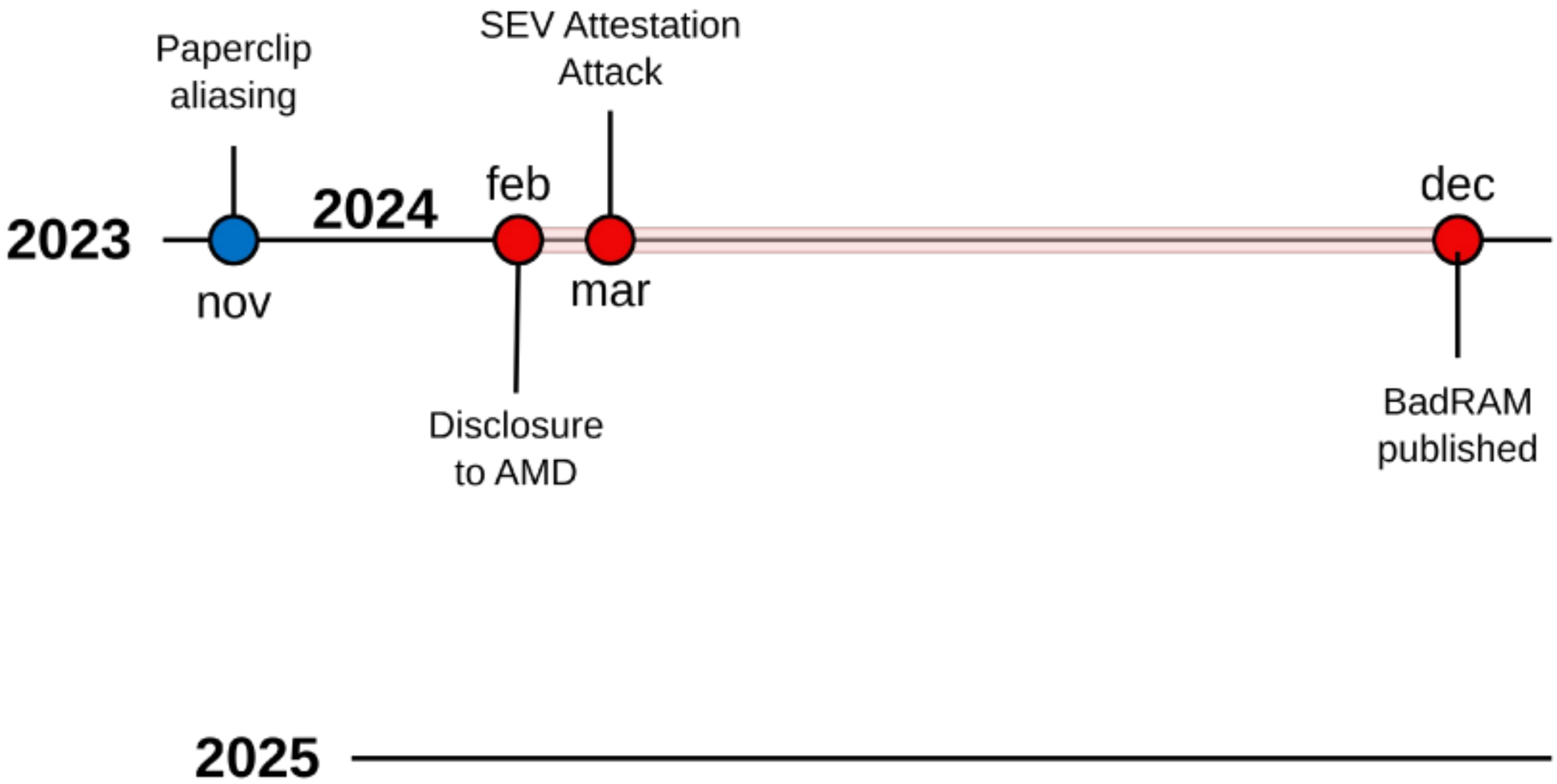
2023 — 2024

2025

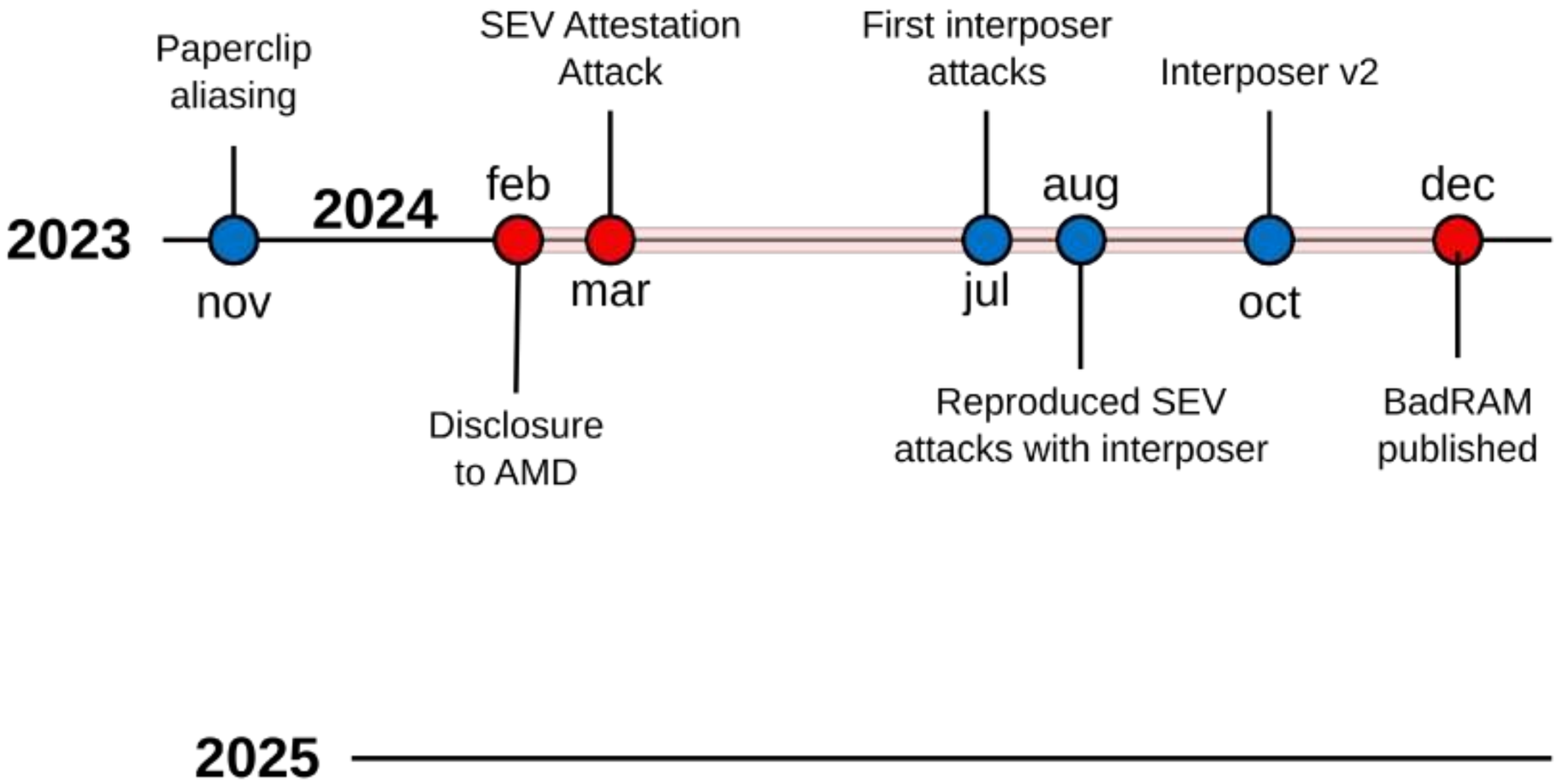
Timeline



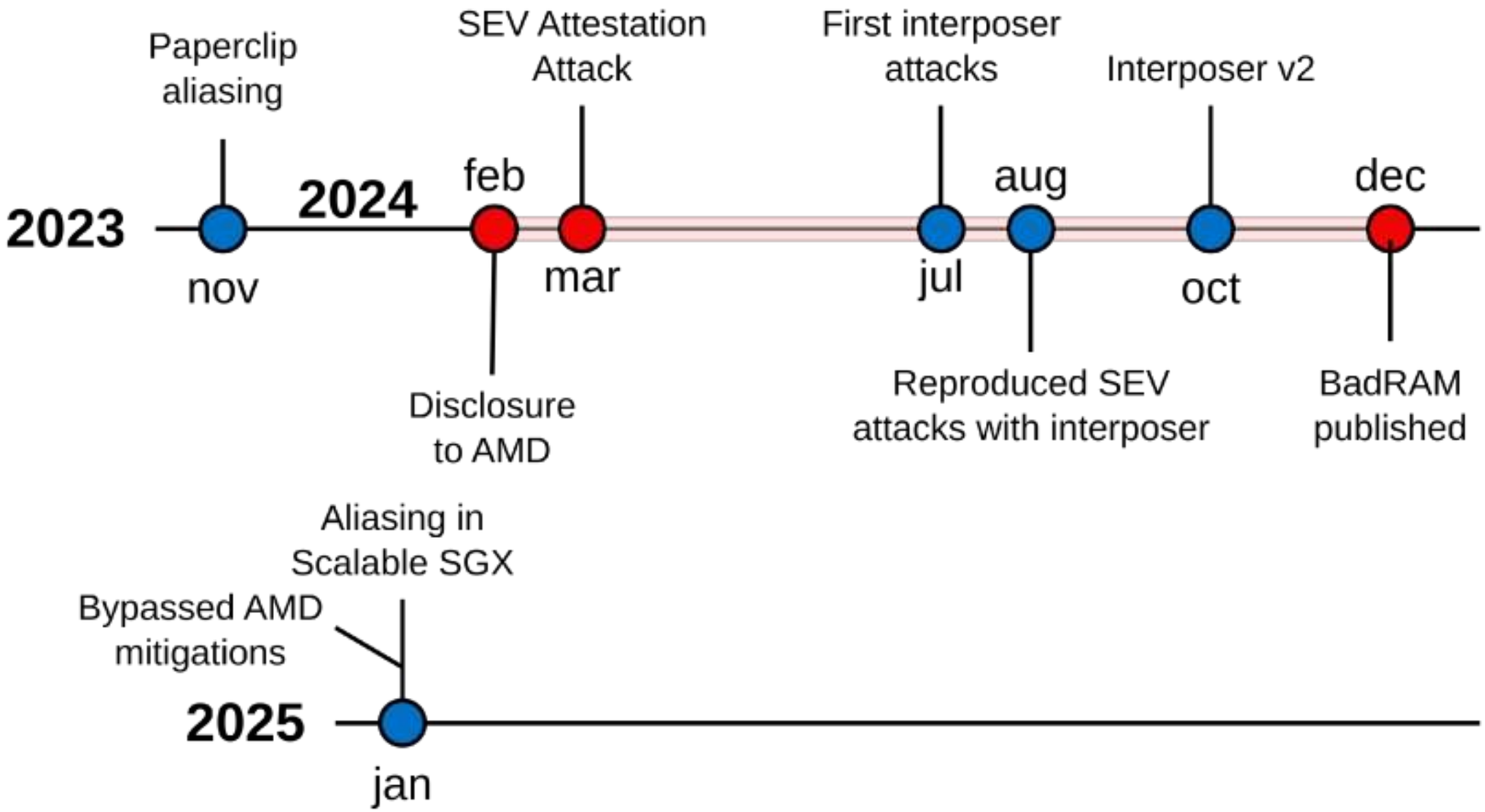
Timeline


Paperclip aliasing
SEV Attestation Attack
2023
2024
feb
dec
nov
mar
Disclosure to AMD
BadRAM published
2025

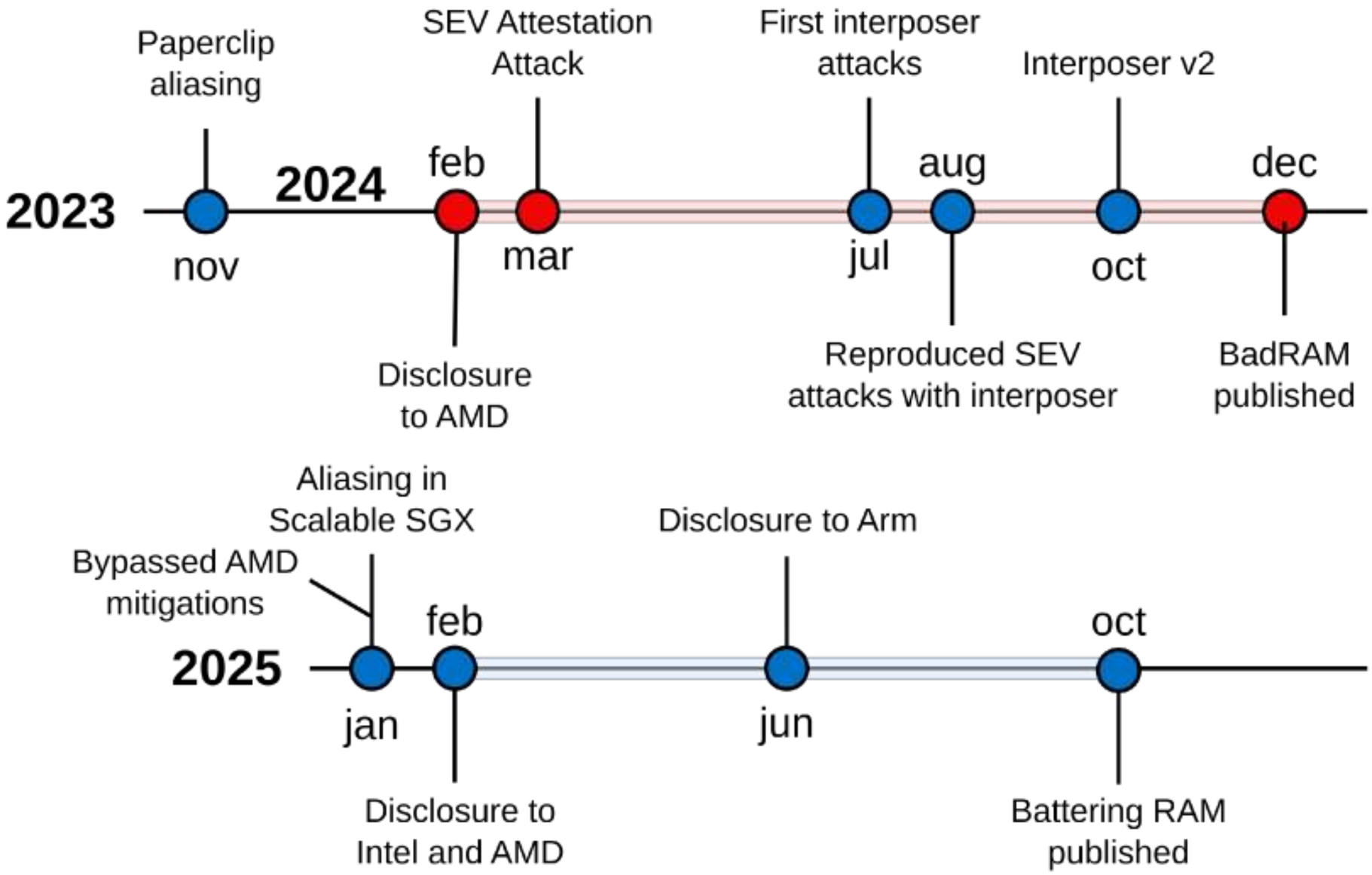
Timeline


Paperclip aliasing
SEV Attestation Attack
First interposer attacks
Interposer v2
2023
2024
feb
aug
dec
nov
mar
jul
oct
Disclosure to AMD
Reproduced SEV attacks with interposer
BadRAM published
2025

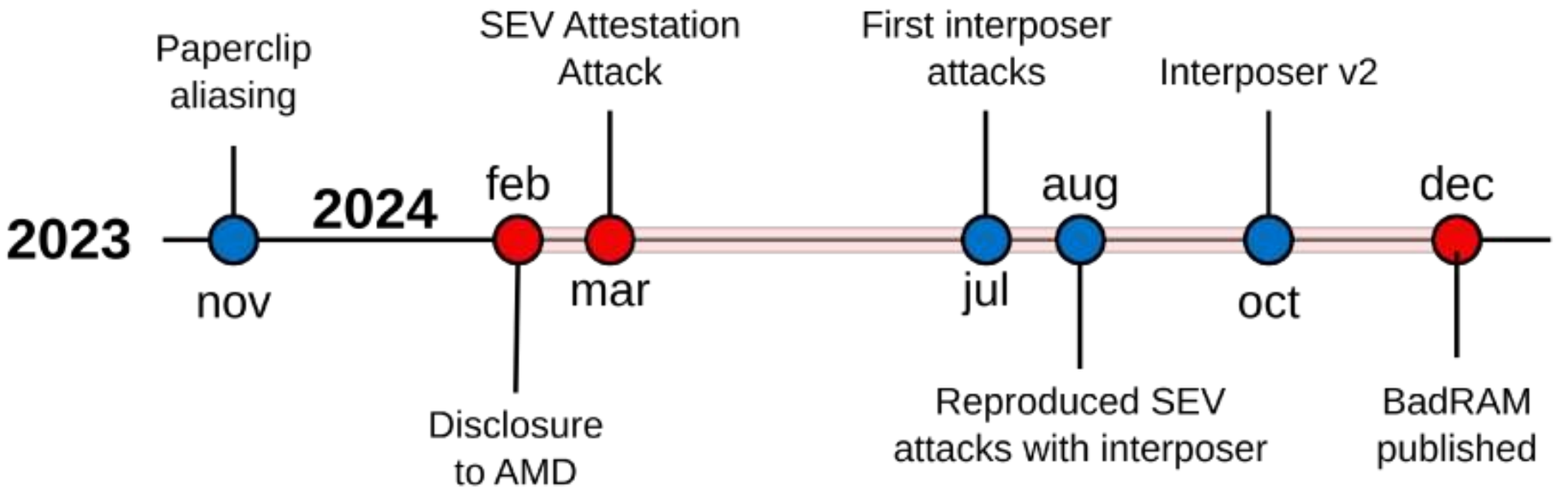
Timeline

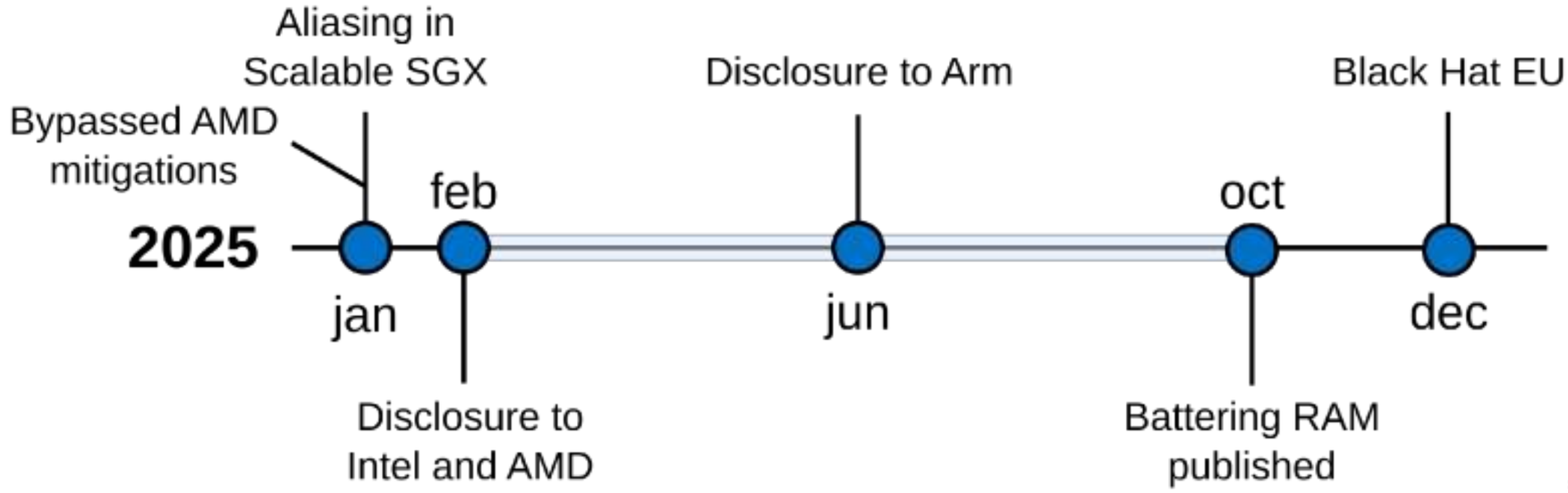

Paperclip aliasing
SEV Attestation Attack
First interposer attacks
Interposer v2
2023
2024
feb
aug
dec
nov
mar
jul
oct
Disclosure to AMD
Reproduced SEV attacks with interposer
BadRAM published
Aliasing in Scalable SGX
Bypassed AMD mitigations
2025
jan

Timeline


Paperclip aliasing
SEV Attestation Attack
First interposer attacks
Interposer v2
2023
nov
2024
feb
mar
jul
aug
oct
dec
Disclosure to AMD
Reproduced SEV attacks with interposer
BadRAM published
Aliasing in Scalable SGX
Disclosure to Arm
Bypassed AMD mitigations
2025
jan
feb
jun
oct
Disclosure to Intel and AMD
Battering RAM published

Timeline


Paperclip aliasing
SEV Attestation Attack
First interposer attacks
Interposer v2
2023
nov
2024
feb
mar
jul
aug
oct
dec
Disclosure to AMD
Reproduced SEV attacks with interposer
BadRAM published


Aliasing in Scalable SGX
Disclosure to Arm
Black Hat EU
Bypassed AMD mitigations
2025
jan
feb
jun
oct
dec
Disclosure to Intel and AMD
Battering RAM published

Road Map

1. Modern memory encryption designs and where to find them
2. BadRAM: What if your DRAM lies to you?
3. Battering Ram: Low-cost physical interposer attacks
4. **Conclusions and takeaways**

Lessons Learned

1. **Confidential computing** is here to stay

2. Challenge your **attacker models**

3. **Hardware attacks** are practical

Lessons Learned

https://batteringram.eu/

1. **Confidential computing** is here to stay

2. Challenge your **attacker models**

3. **Hardware attacks** are practical

Thank you! Questions?


black hat
BRIEFINGS
DECEMBER 10-11, 2025
EXCEL LONDON / UNITED KINGDOM
#BHEU @BlackHatEvents