Not So Secure TSC

Co-location Detection on AMD SEV-SNP Confidential Virtual Machines

Jonas Juffinger, **Sudheendra Raghav Neela**, Daniel Gruss

23rd International Conference on Applied Cryptography and Network Security

isec.tugraz.at

AMD SEV-SNP’s SecureTSC

Jonas Juffinger & Sudheendra Raghav Neela

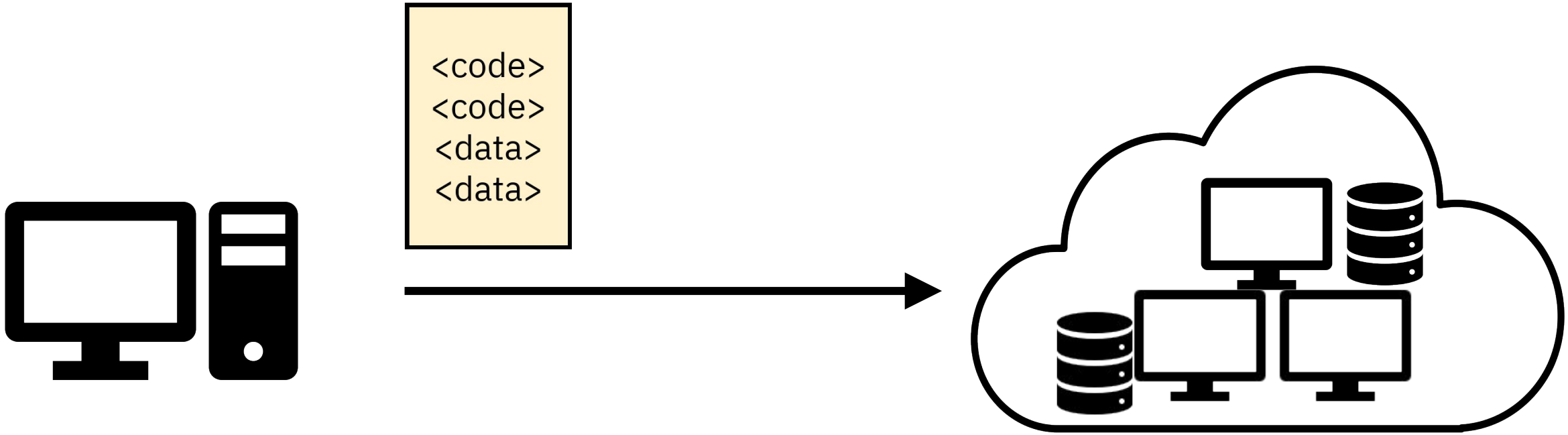

AMD
AMD
SEV-SNP's SecureTSC
Secure
Encrypted
Virtualization

Virtualization

- AMD SVM (or AMD-V)
- Intel VT-x

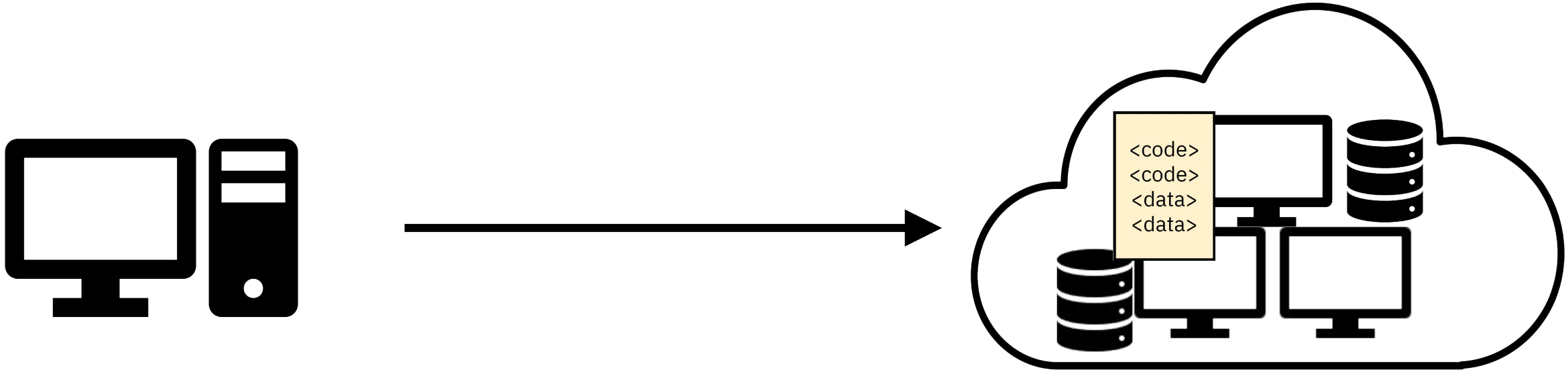
Virtualization

- AMD SVM (or AMD-V)
- Intel VT-x

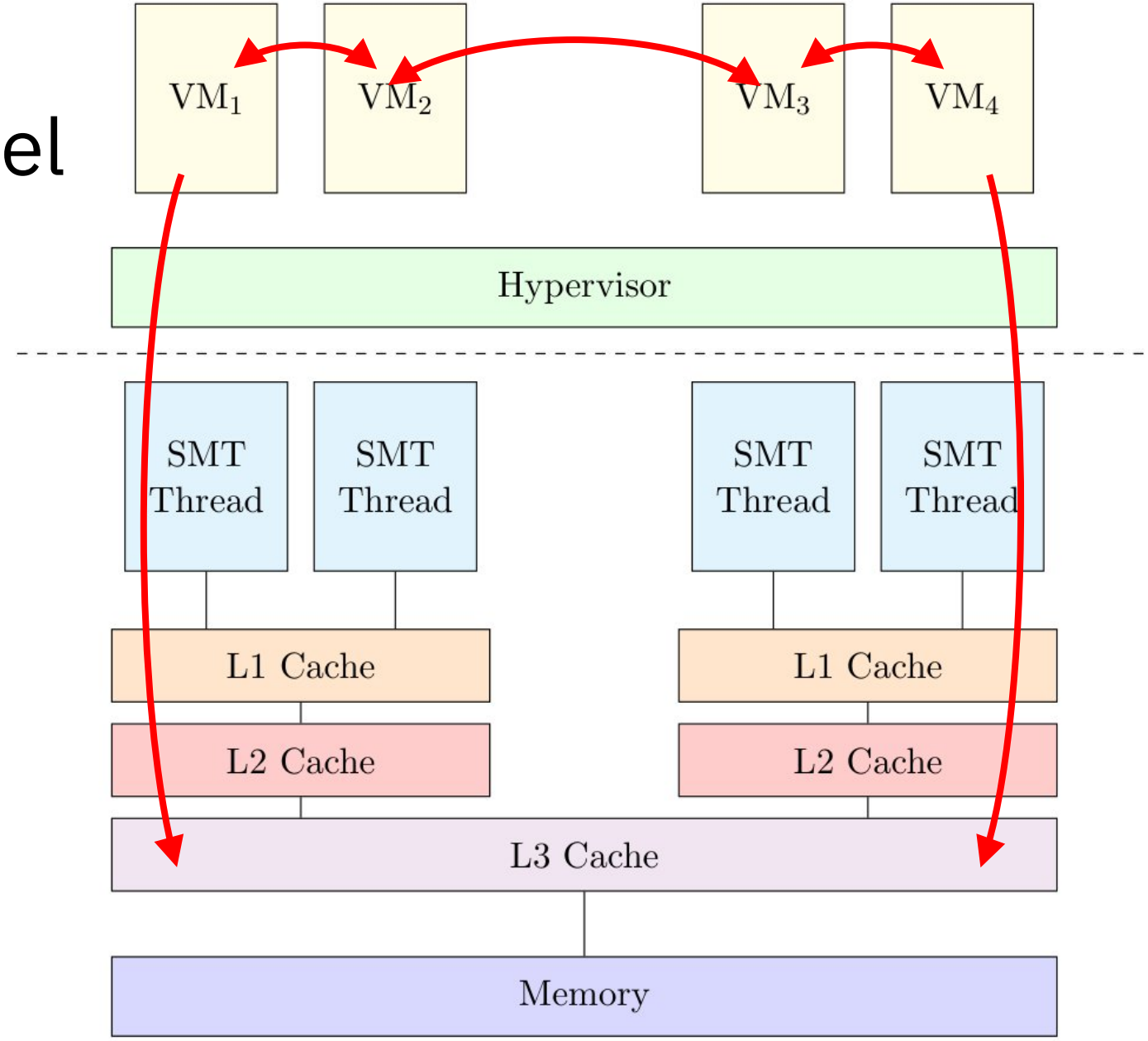


Virtualization

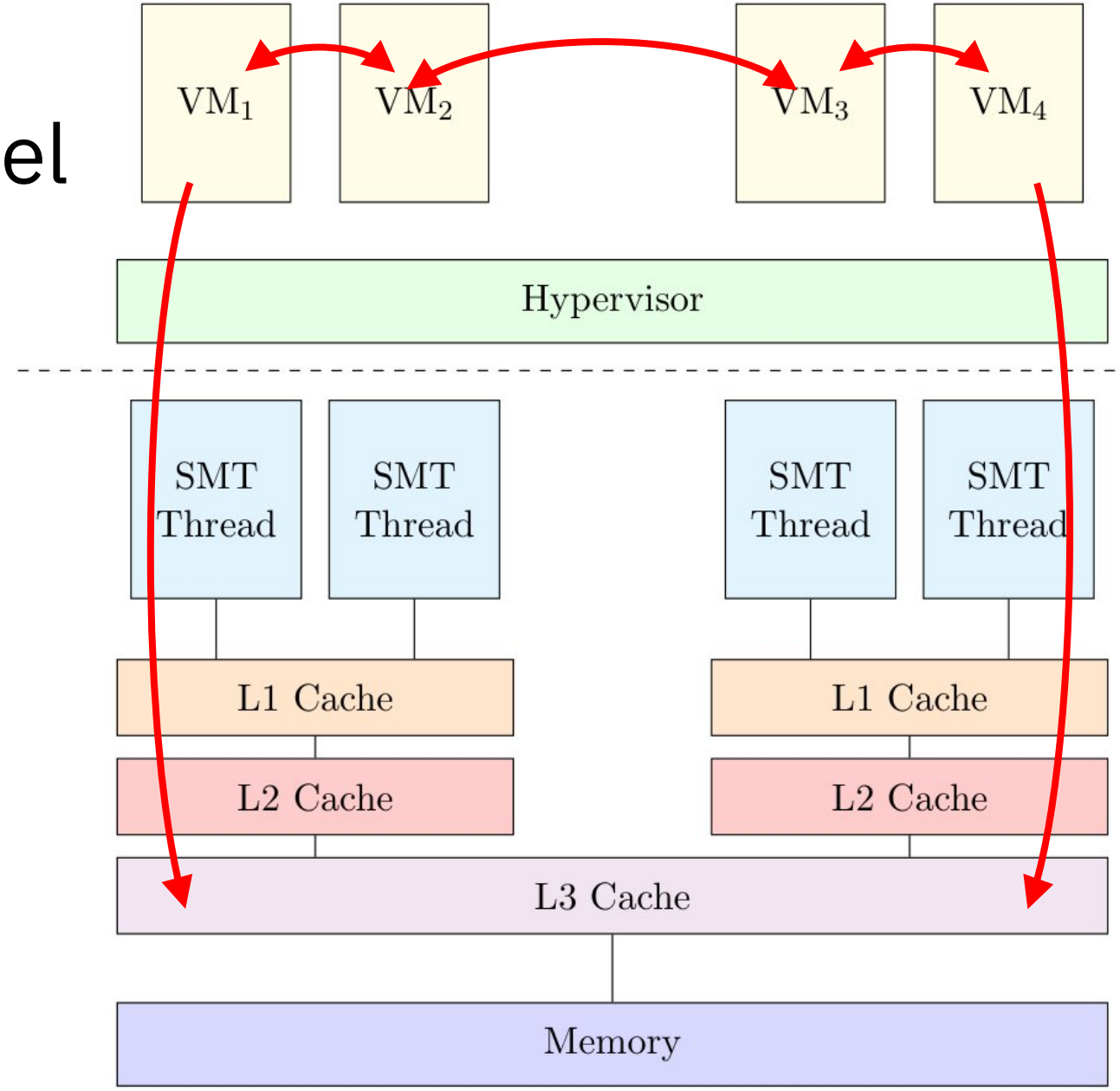
- AMD SVM (or AMD-V)
- Intel VT-x



Cross VM Side-Channel Attacks



Cross VM Side-Channel Attacks



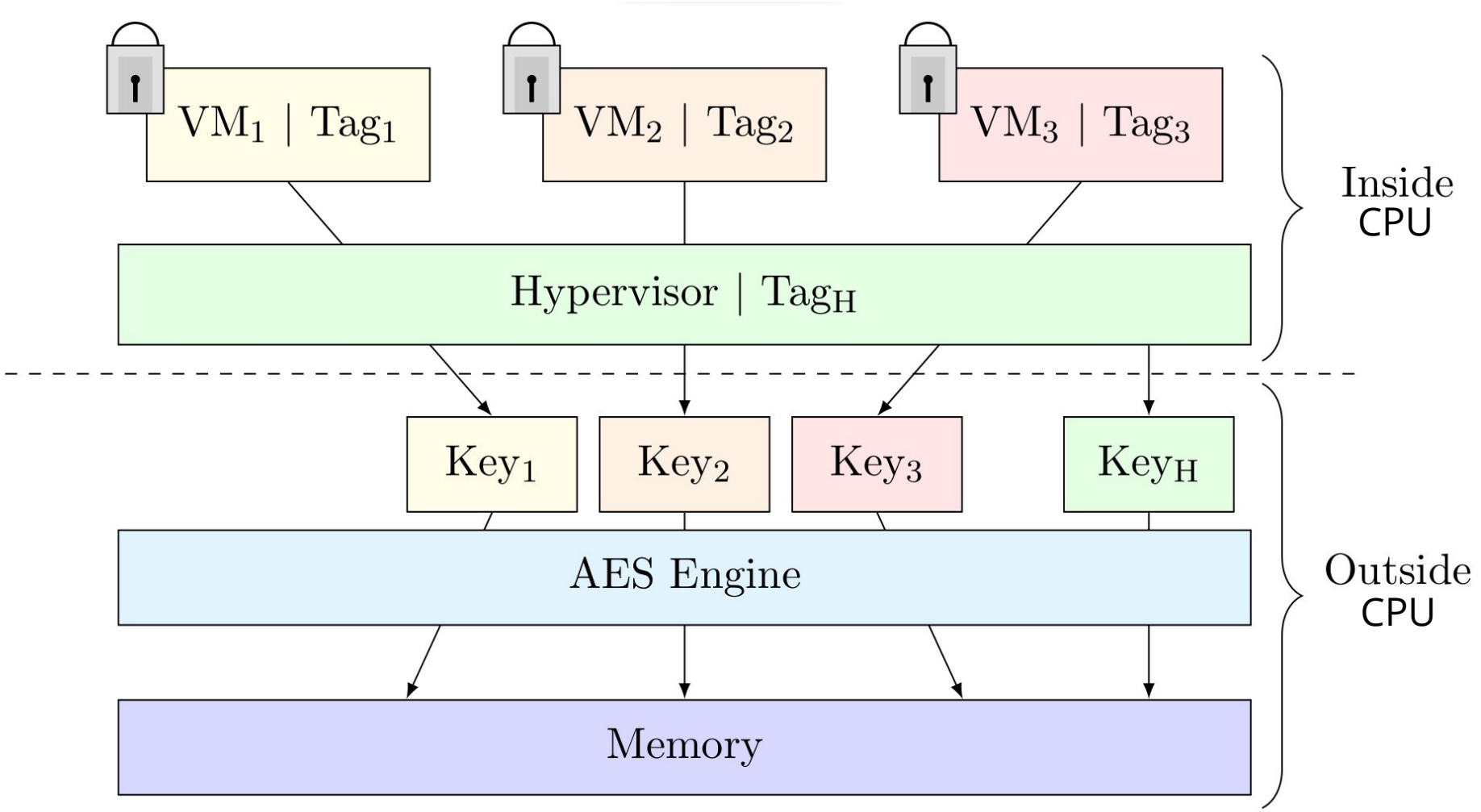
Prerequisite: Co-located Virtual Machines

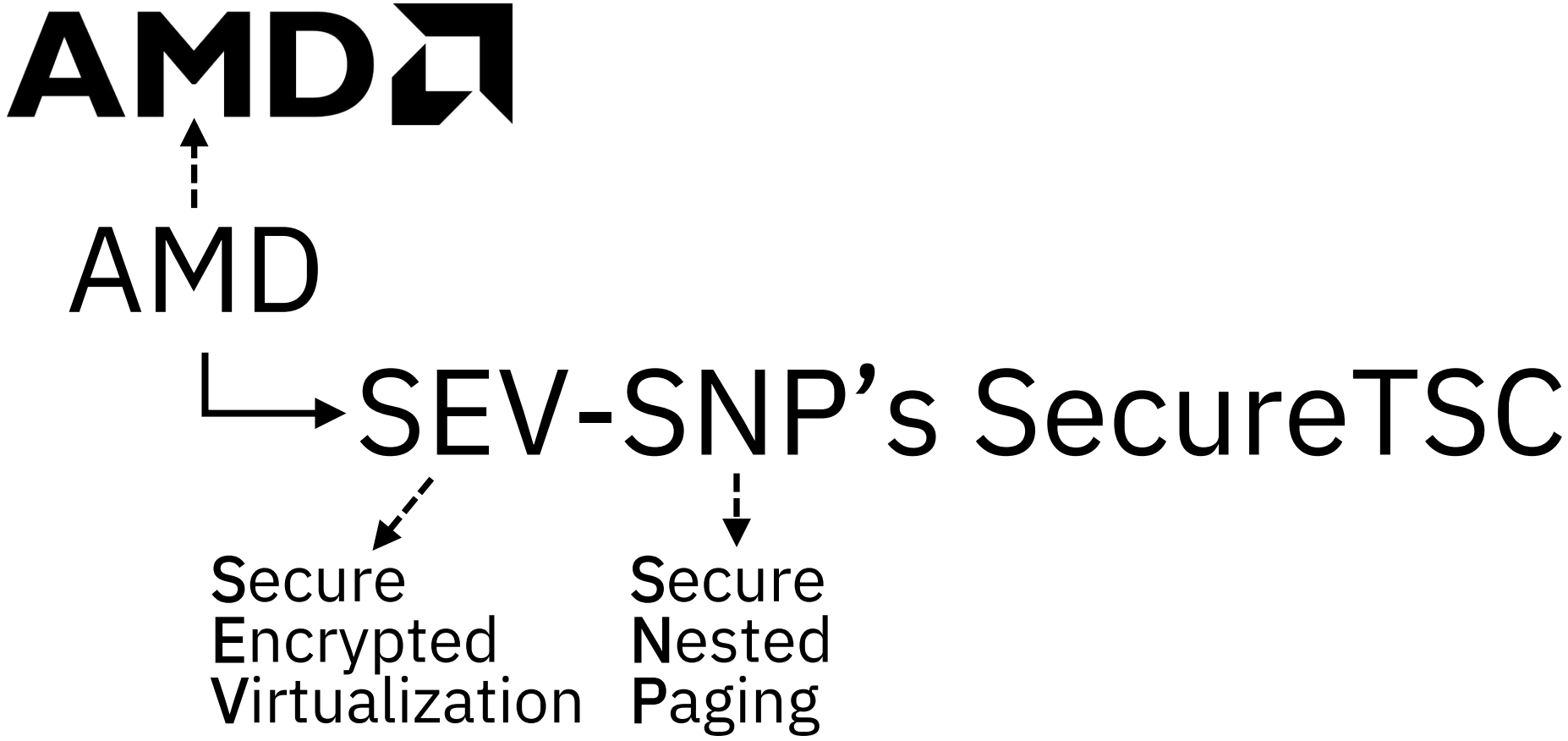
Confidential Computing

- De-coupling customers from the cloud provider.
- Instead, rely on the hardware being authentic (proved via attestation)


AMD
AMD
SEV-SNP's SecureTSC
Secure
Encrypted
Virtualization

AMD Secure Encrypted Virtualization (SEV)



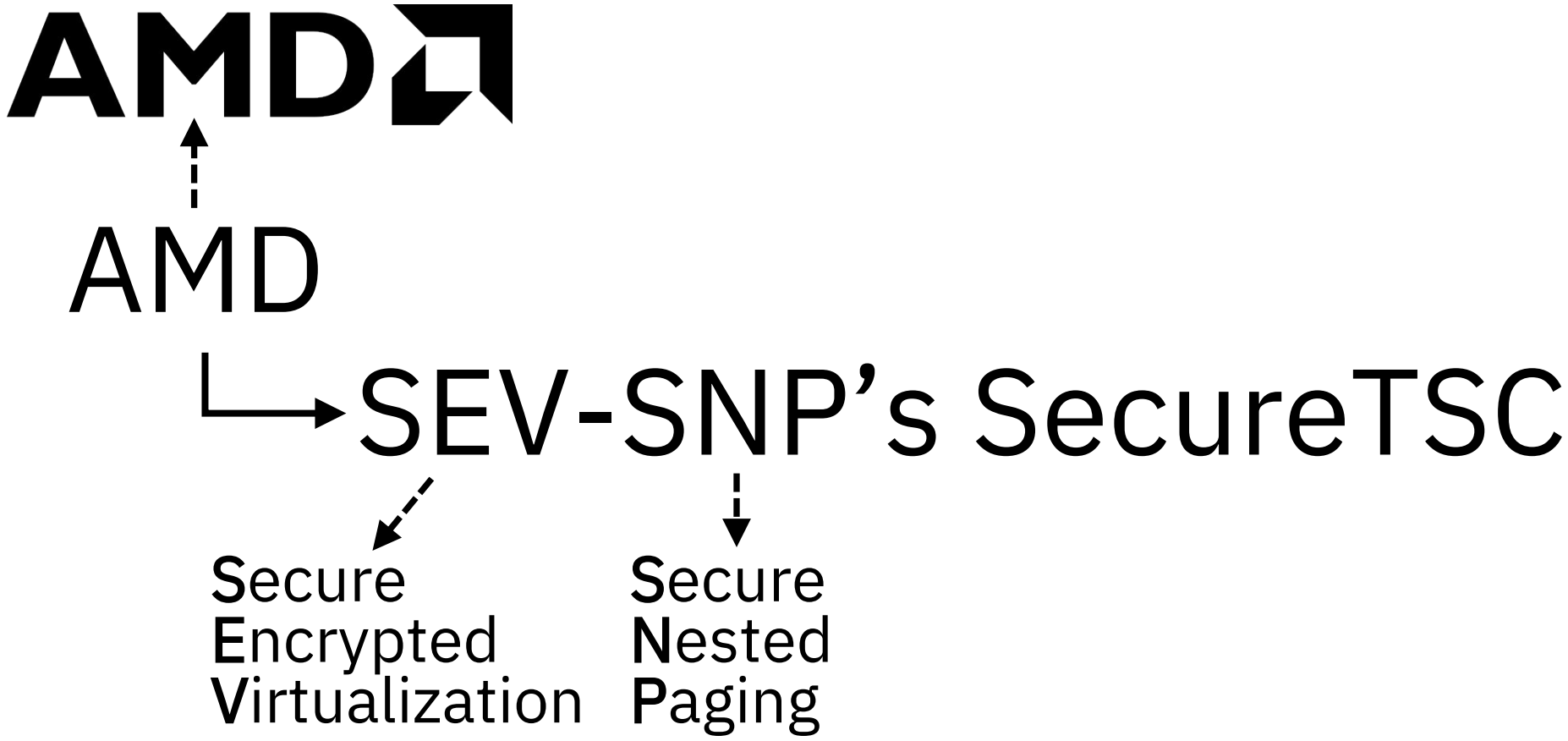

AMD
AMD
SEV-SNP's SecureTSC
Secure
Encrypted
Virtualization
Secure
Nested
Paging

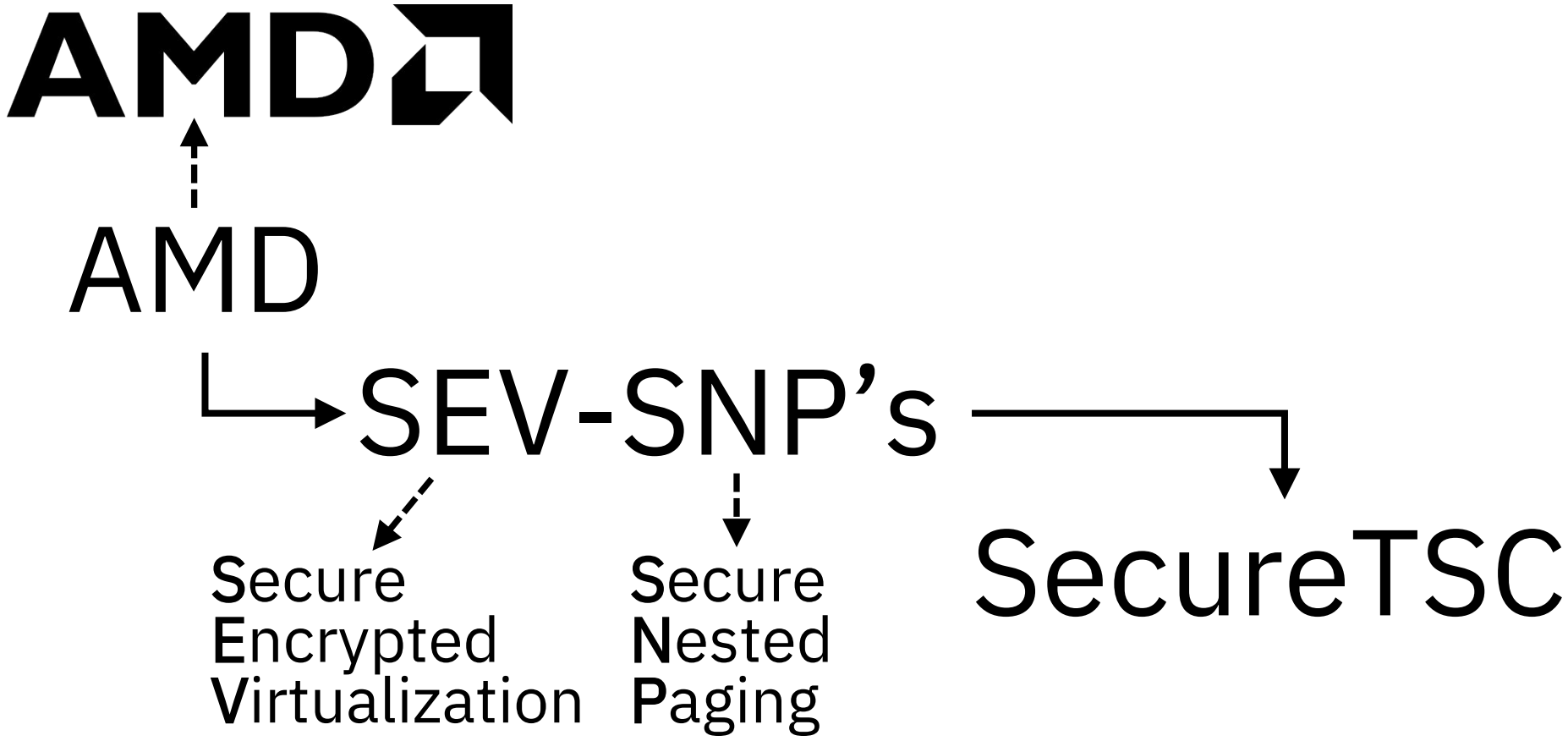
AMD SEV Secure Nested Paging (SEV-SNP)

- Memory enforcement checks and protection
- Prevents a malicious hypervisor from attacking memory
- New features:
  - SecureTSC
  - Trusted `CPUID` data
  - Flexible VM attestation
  - Disallowing Instruction Based Sampling
  - Obfuscating VM Save State Area registers

AMD SEV Secure Nested Paging (SEV-SNP)

- Memory enforcement checks and protection
- Prevents a malicious hypervisor from attacking memory
- New features:
  - **SecureTSC**
  - Trusted `CPUID` data
  - Flexible VM attestation
  - Disallowing Instruction Based Sampling
  - Obfuscating VM Save State Area registers


AMD
AMD
SEV-SNP's SecureTSC
Secure
Encrypted
Virtualization
Secure
Nested
Paging


AMD
AMD
SEV-SNP's
Secure
Encrypted
Virtualization
Secure
Nested
Paging
SecureTSC

The Time-Stamp Counter (TSC)

- Model-Specific Register (MSR):
  - high-resolution
  - monotonically-increasing
  - performance-monitoring

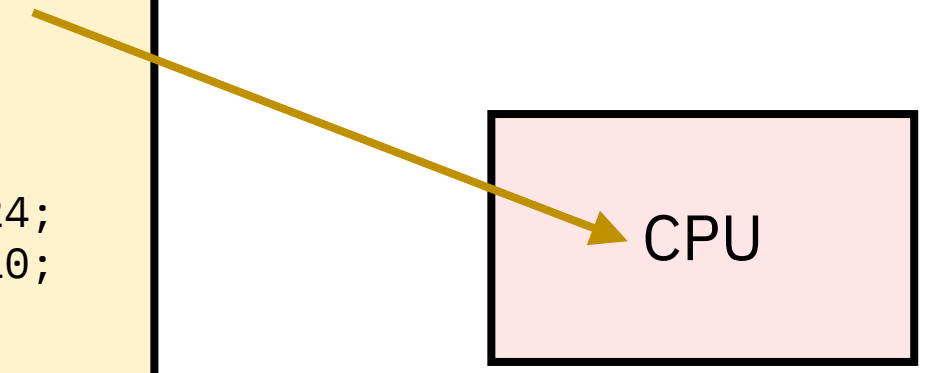
- `RDMSR, WRMSR`
- `RDTSC`
- `RDTSCP`

- On Linux:
  - `gettimeofday`
  - `clock_gettime`

The TSC is Interceptable

Virtual Machine

```
XOR RAX, RAX;
XOR RCX, RCX;
XOR RDX, RDX;

MOV RAX, 0x0124;
MOV RDX, 0x4210;

MUL RDX;
MOV RCX, RDX;

RDTSC;

MOV RAX, RDI;
SAL RAX, 32;
OR  RAX, RDX;
```



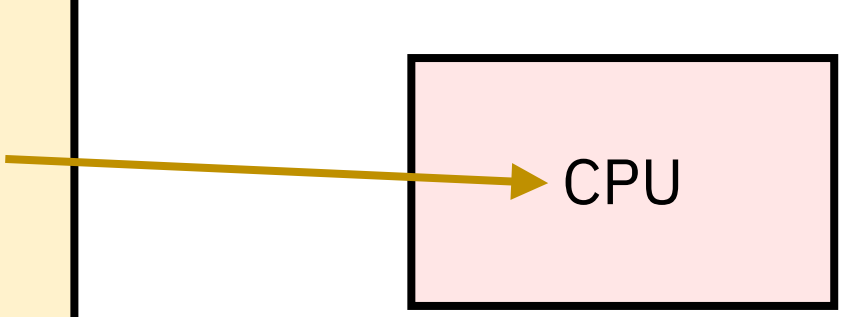
VMCB

- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL
- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

The TSC is Interceptable

Virtual Machine

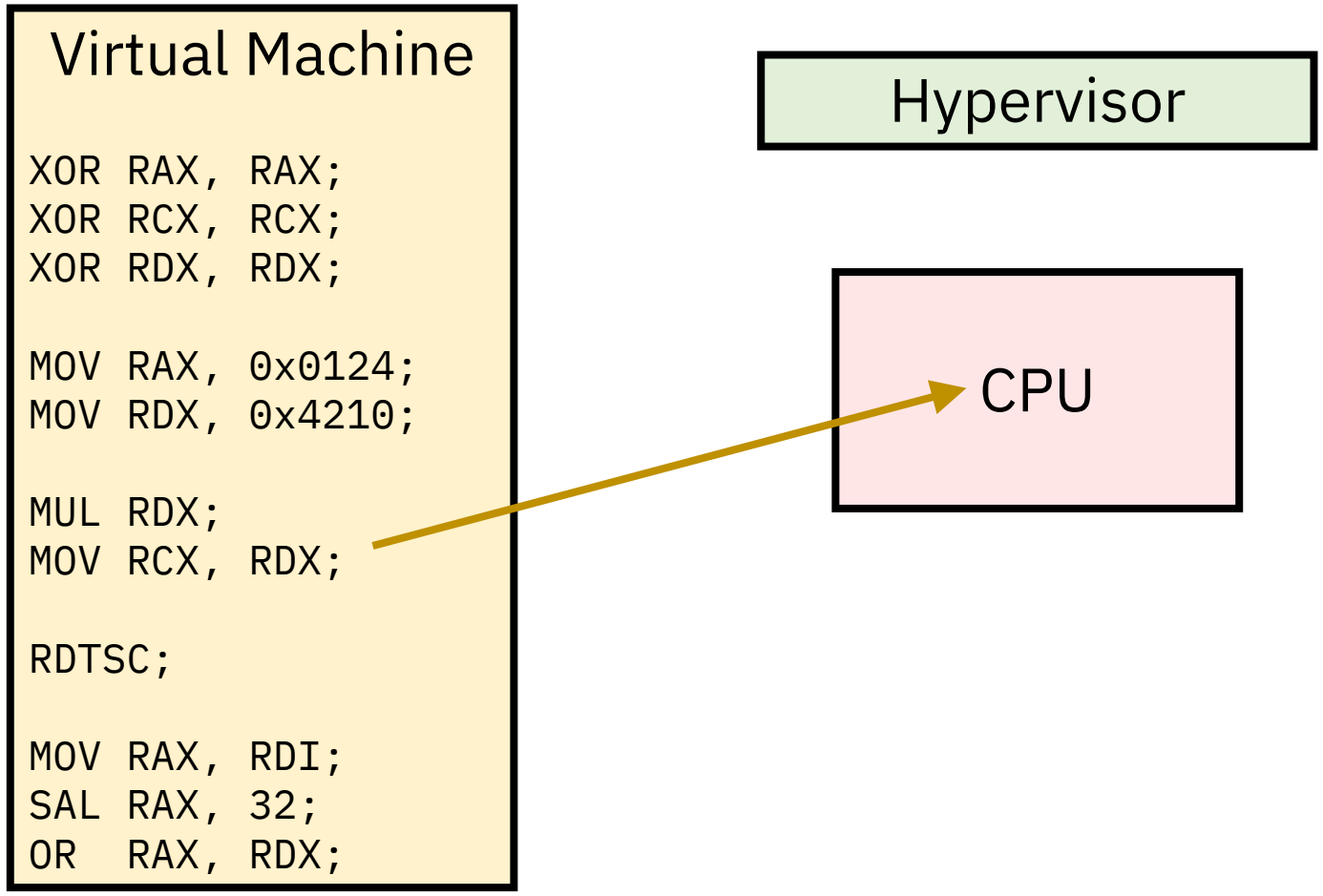
```
XOR RAX, RAX;
XOR RCX, RCX;
XOR RDX, RDX;

MOV RAX, 0x0124;
MOV RDX, 0x4210;

MUL RDX;
MOV RCX, RDX;

RDTSC;

MOV RAX, RDI;
SAL RAX, 32;
OR  RAX, RDX;
```



VMCB

- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL
- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

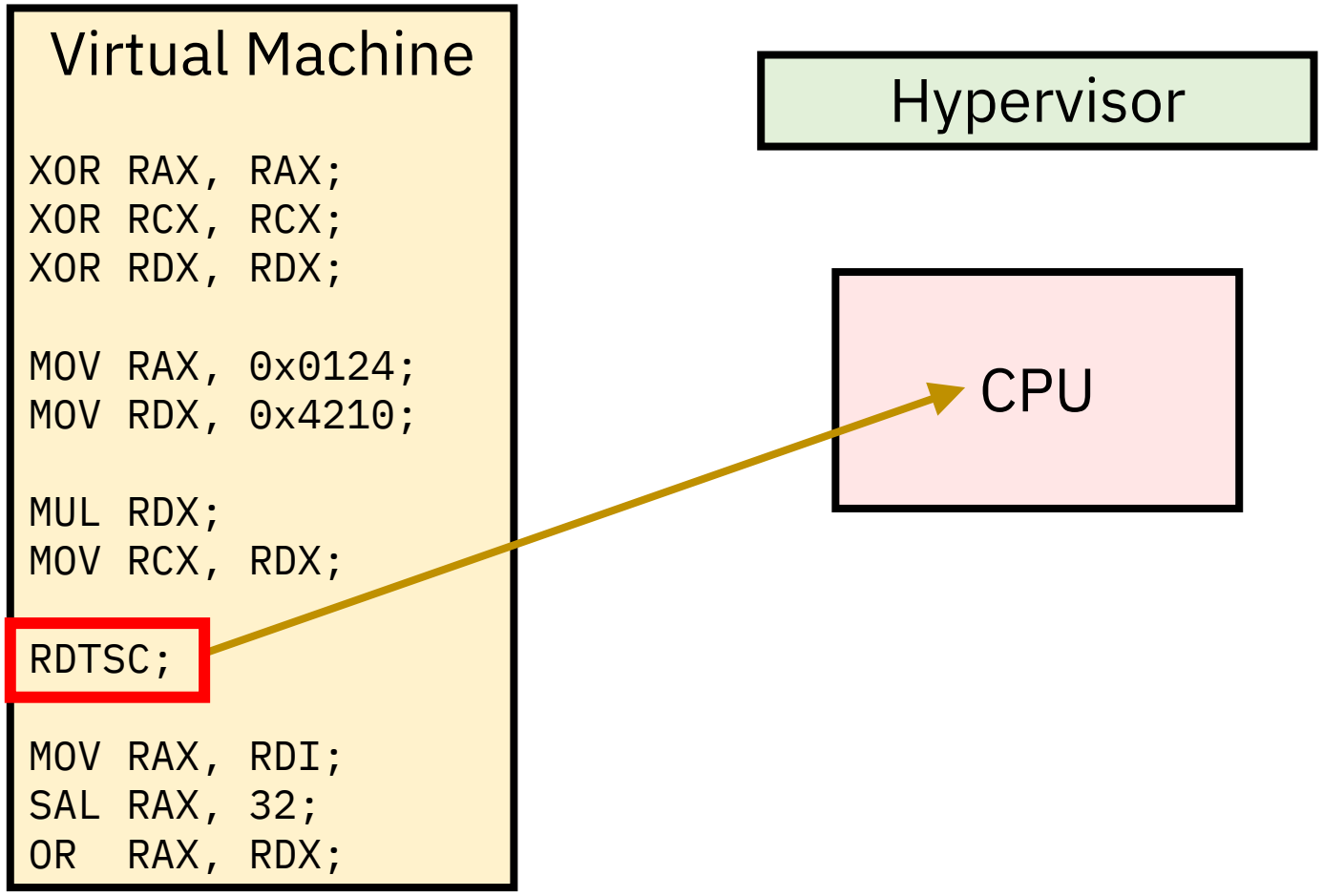
The TSC is Interceptable



VMCB

- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL
- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

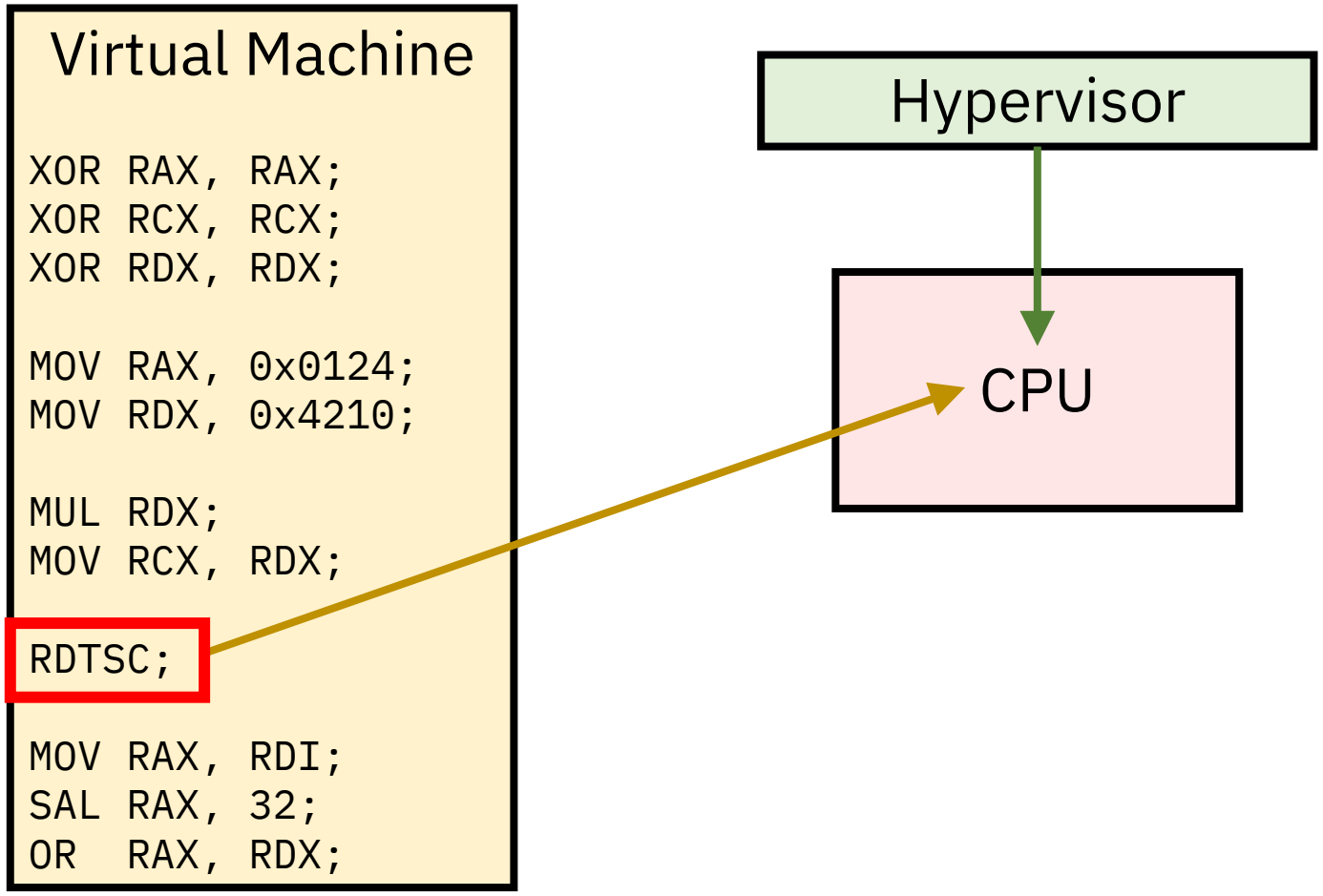
The TSC is Interceptable



VMCB

- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL
- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

The TSC is Interceptable



VMCB

- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL
- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

SecureTSC

- Secure method for SEV-SNP VMs to access the Time-Stamp Counter.
- The hypervisor’s TSC and VM’s SecureTSC are independent.

SystemTSC

Two parameters about SecureTSC (known by the VM):

- SecureTSC Scale
- SecureTSC Offset

$$SecureTSC = TSC\_Scale \times SystemTSC + TSC\_Offset$$

SystemTSC

Two parameters about SecureTSC (known by the VM):

- SecureTSC Scale
- SecureTSC Offset

$$SecureTSC = TSC\_Scale \times SystemTSC + TSC\_Offset$$

$$SystemTSC = \frac{SecureTSC - TSC\_Offset}{TSC\_Scale}$$

SystemTSC

Two parameters about SecureTSC (known by the VM):

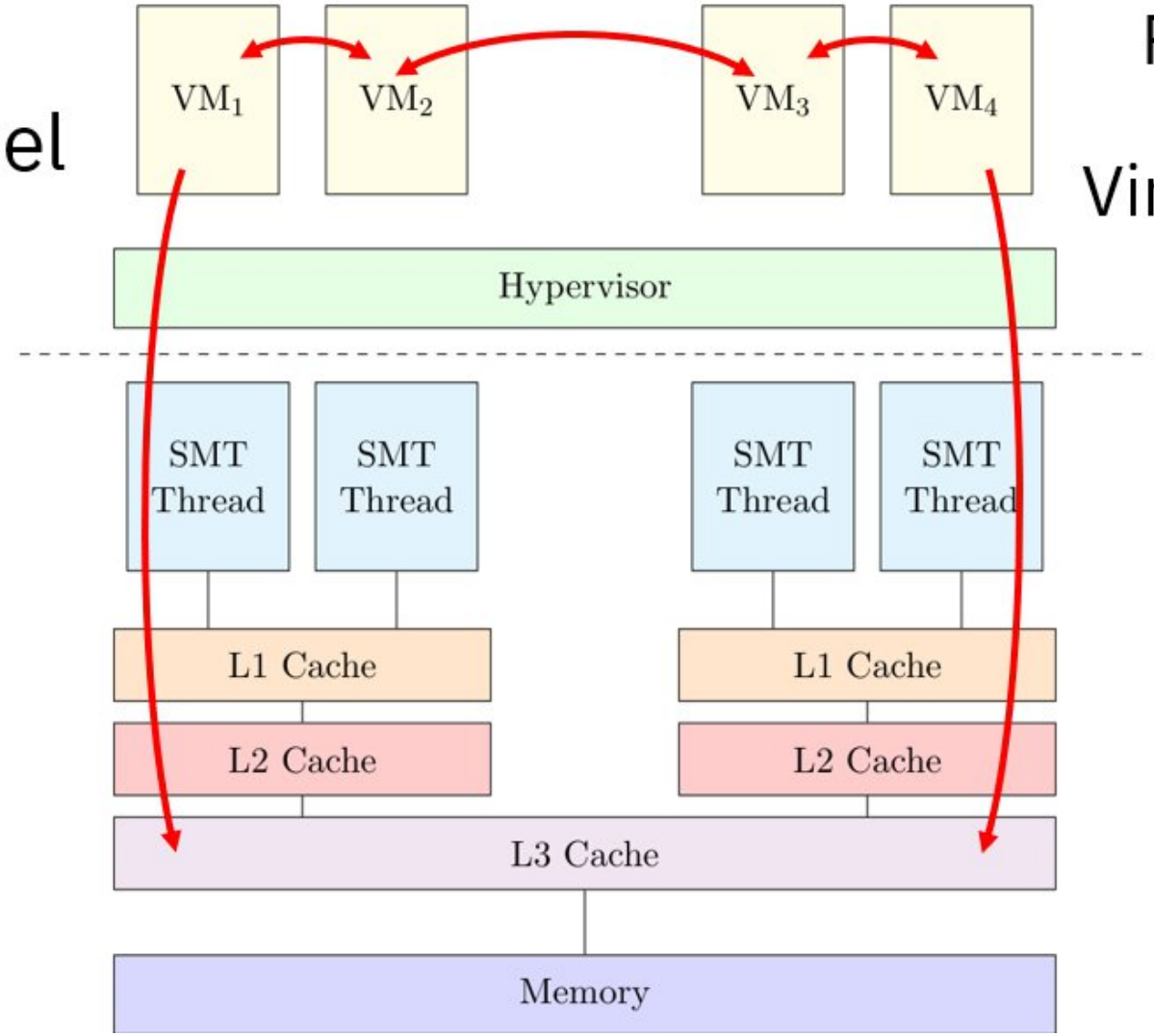
- SecureTSC Scale
- SecureTSC Offset

$$SecureTSC = TSC\_Scale \times SystemTSC + TSC\_Offset$$

$$SystemTSC = \frac{SecureTSC - TSC\_Offset}{TSC\_Scale}$$

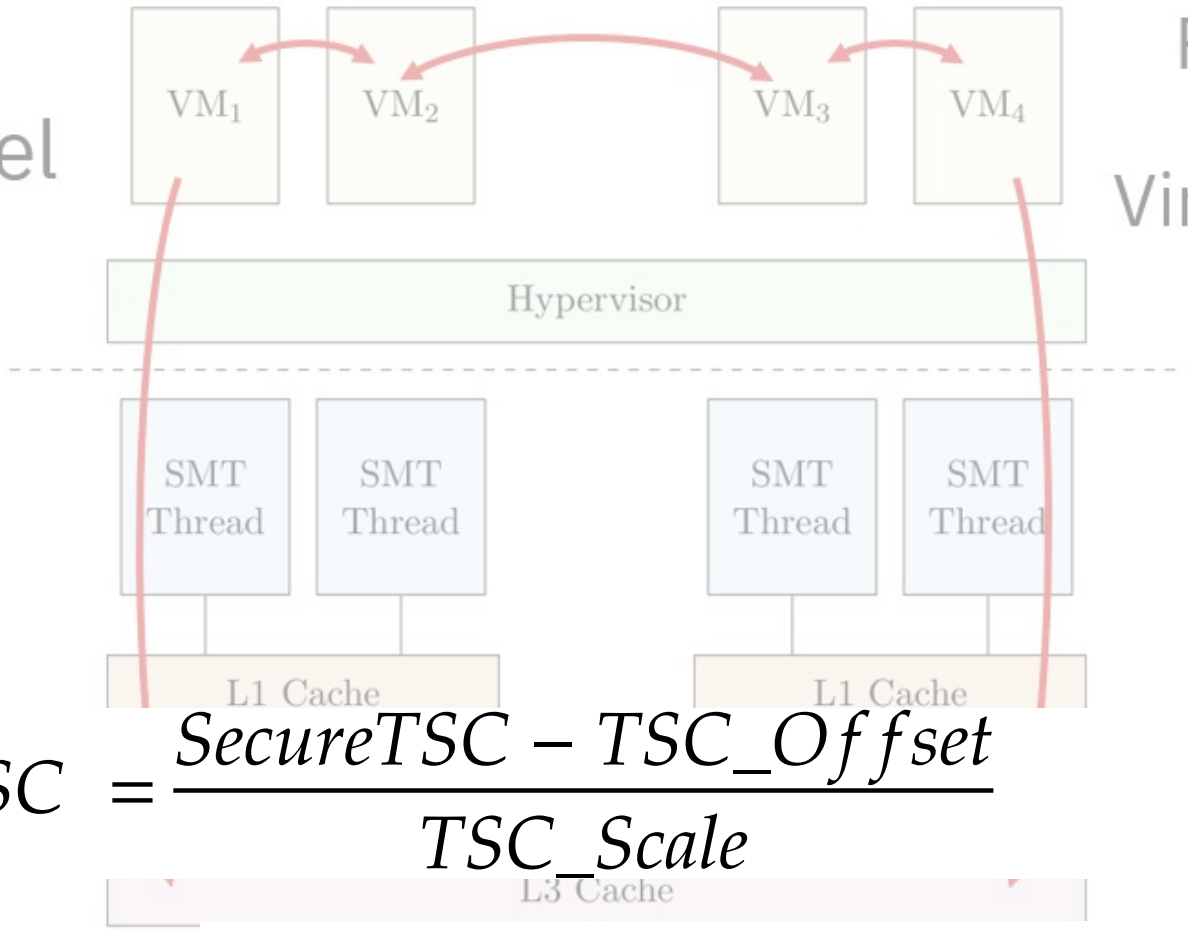
Same for every SEV-SNP VM on the **same** physical machine!

Cross VM Side-Channel Attacks



Prerequisite: Co-located Virtual Machines

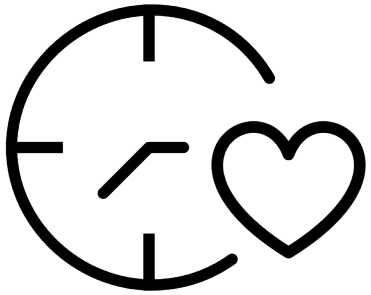
Cross VM Side-Channel Attacks



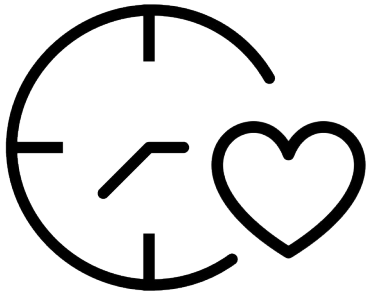
Prerequisite: Co-located Virtual Machines

$$SystemTSC = \frac{SecureTSC - TSC\_Offset}{TSC\_Scale}$$

Same for every SEV-SNP VM on the **same** physical machine!



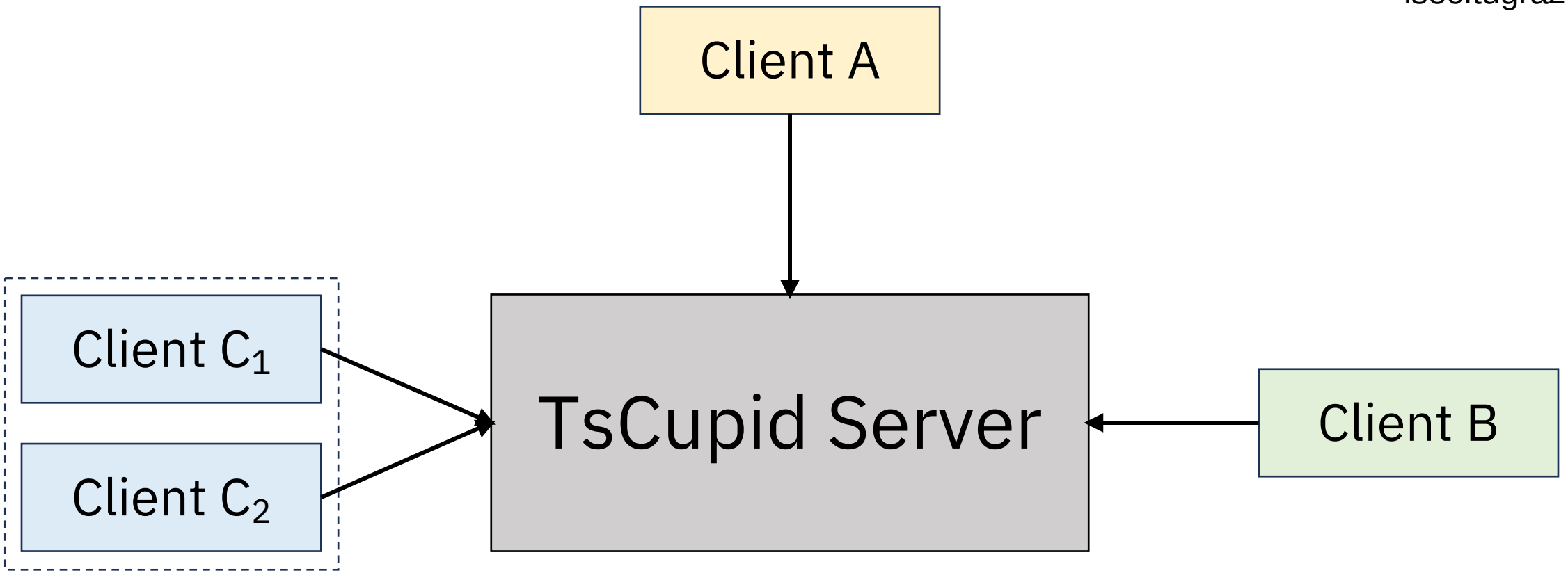
TsCupid

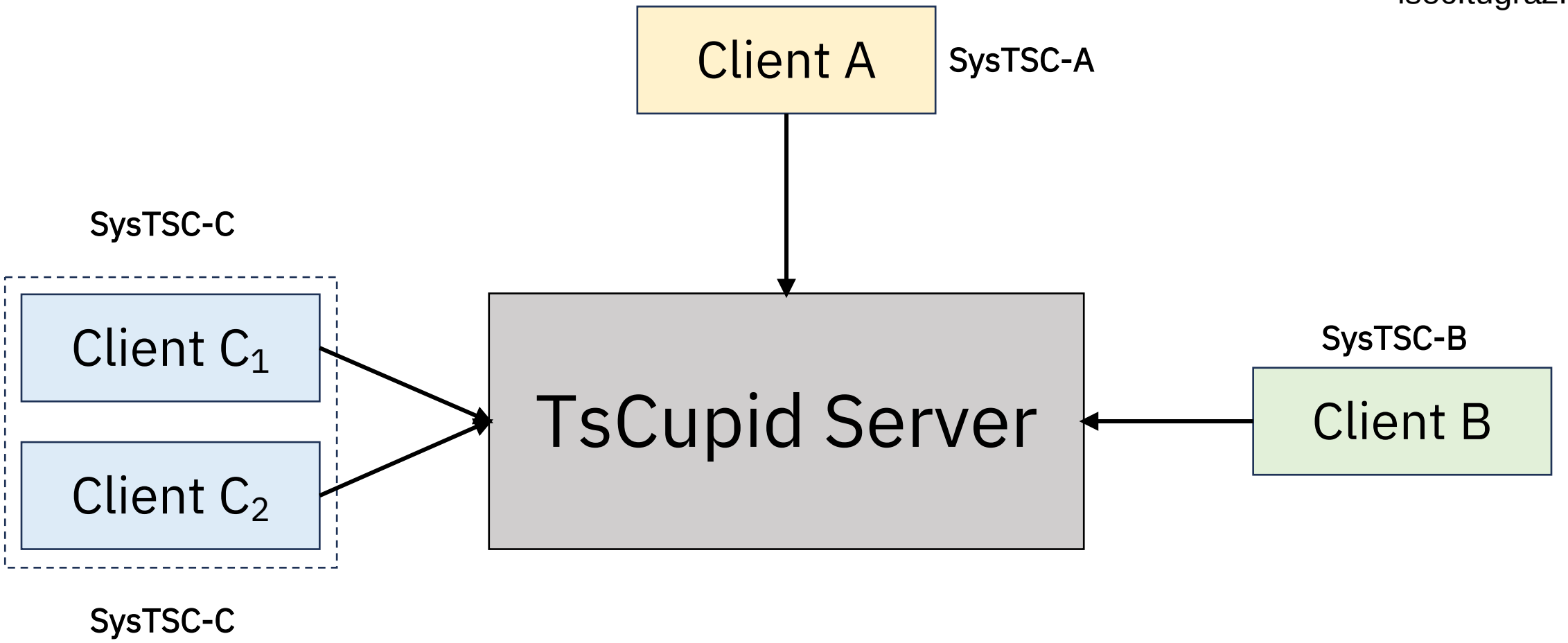


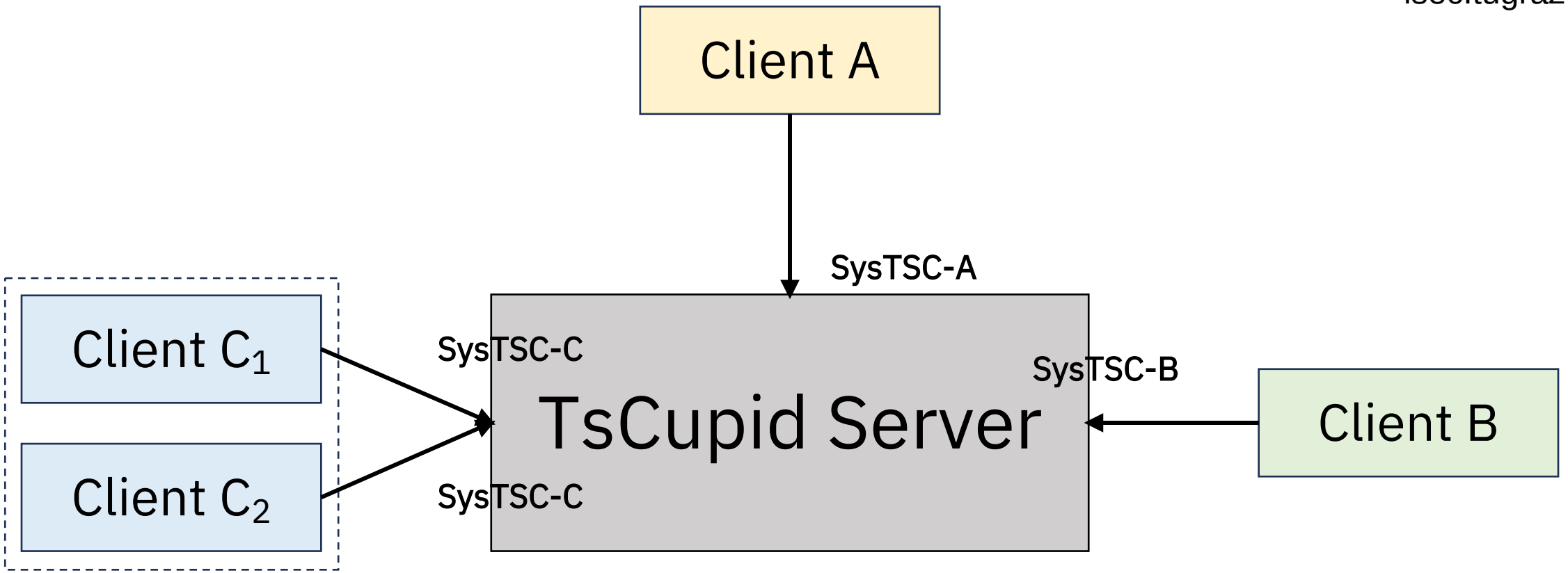
TsCupid

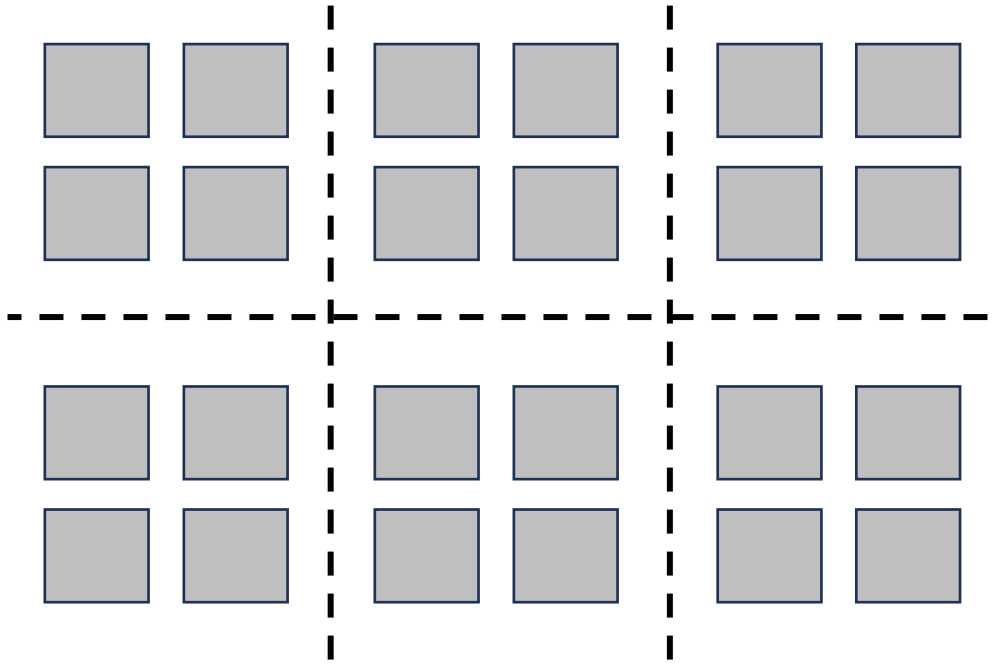
A Network Protocol for Co-location Detection

Jonas Juffinger & Sudheendra Raghav Neela

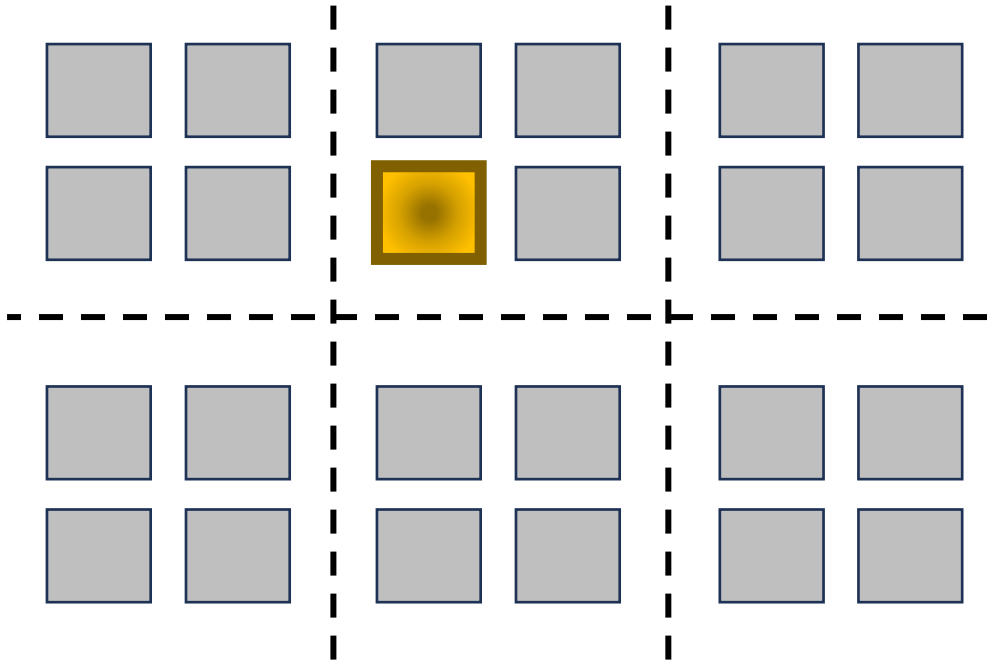

Client A
Client C_1
Client C_2
TsCupid Server
Client B


Client A
SysTSC-A
SysTSC-C
Client C_1
Client C_2
SysTSC-C
TsCupid Server
SysTSC-B
Client B

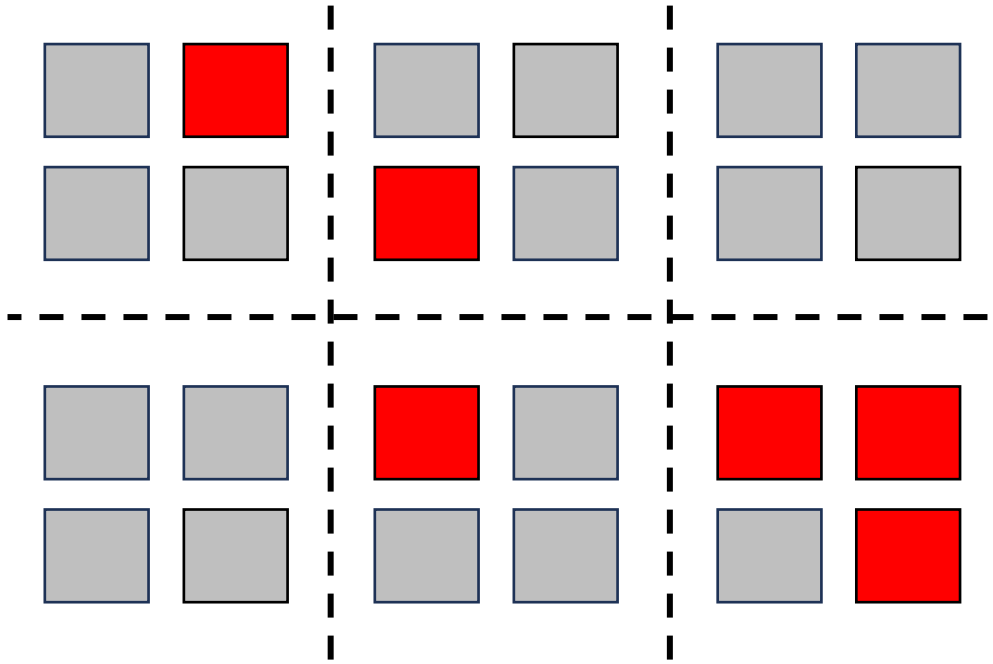

Client A
SysTSC-A
Client C_1
Client C_2
SysTSC-C
SysTSC-C
SysTSC-B
TsCupid Server
Client B



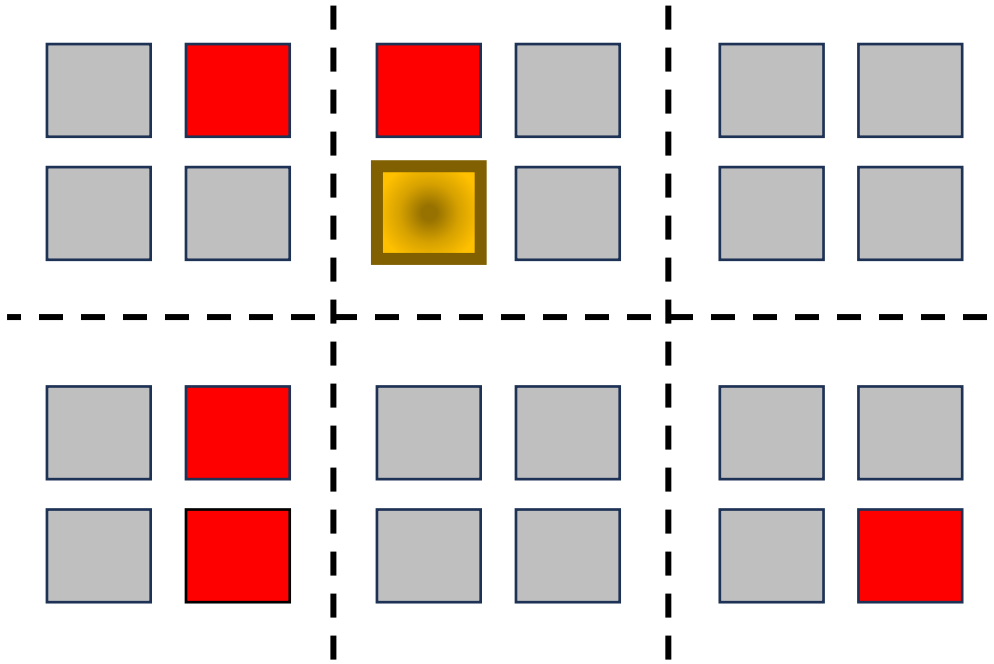
Co-locating Attackers



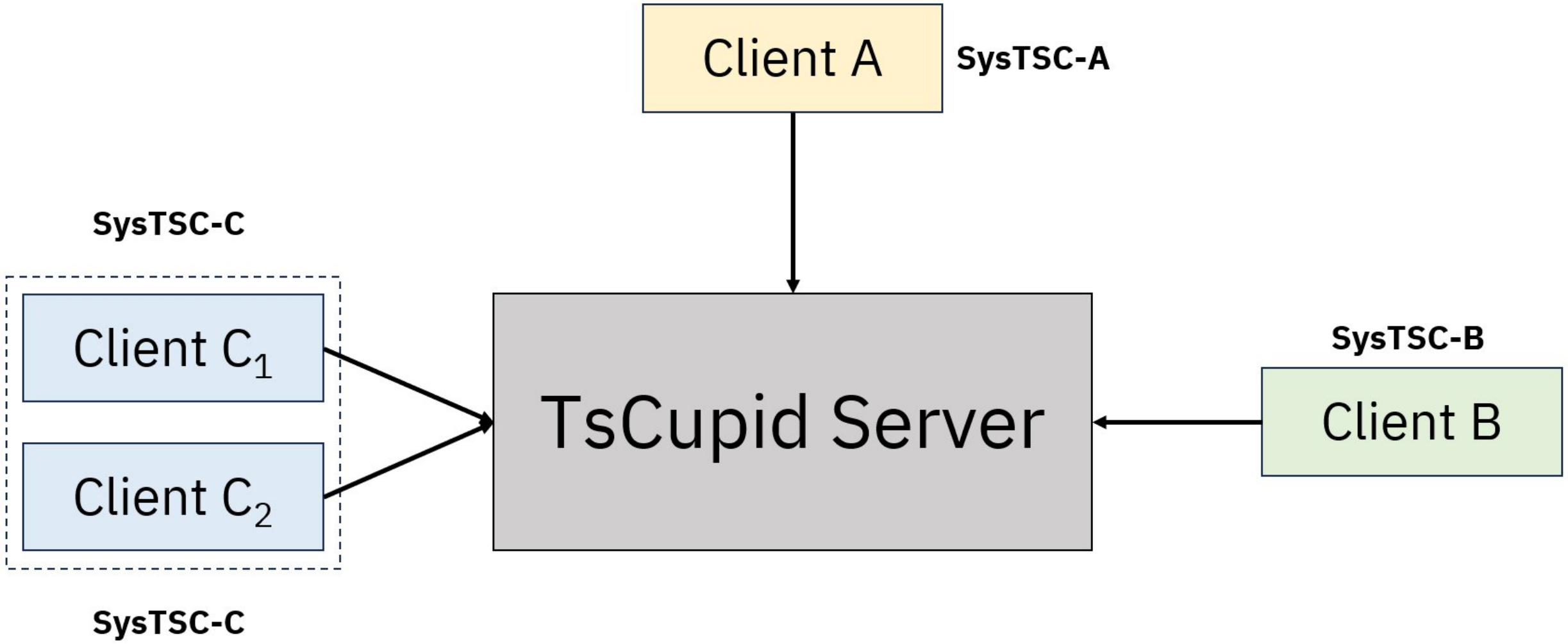
Targeted Guest



Co-locating Attackers

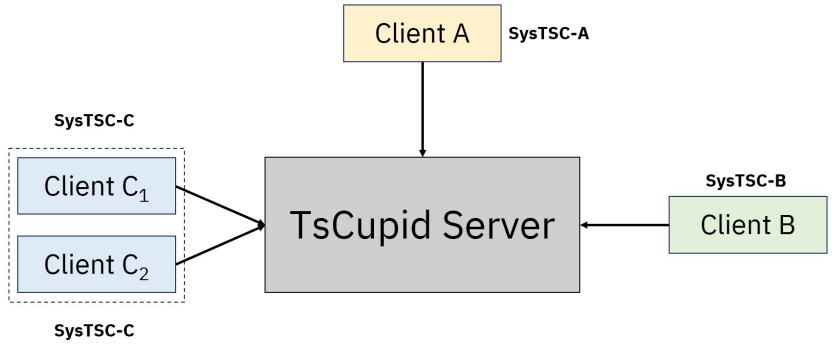


Targeted Guest

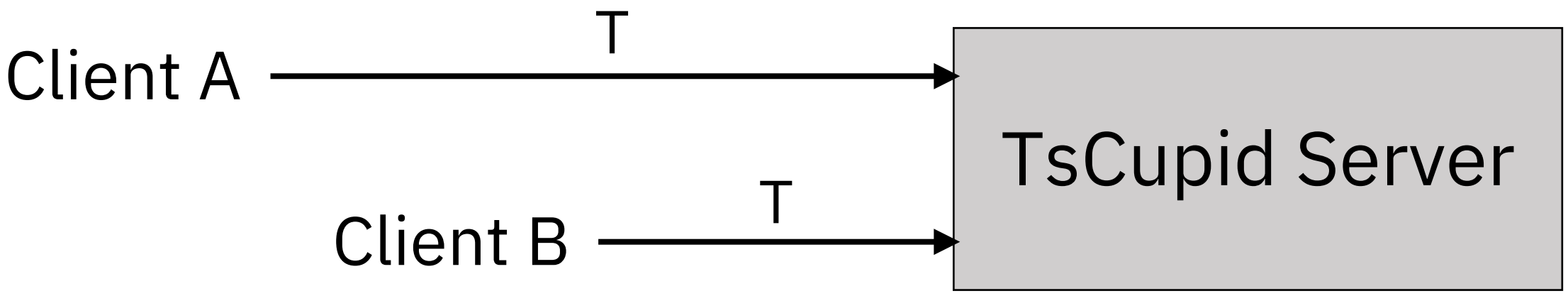

Client A
SysTSC-A
SysTSC-C
Client C_1
Client C_2
TsCupid Server
SysTSC-B
Client B
SysTSC-C

Three Challenges

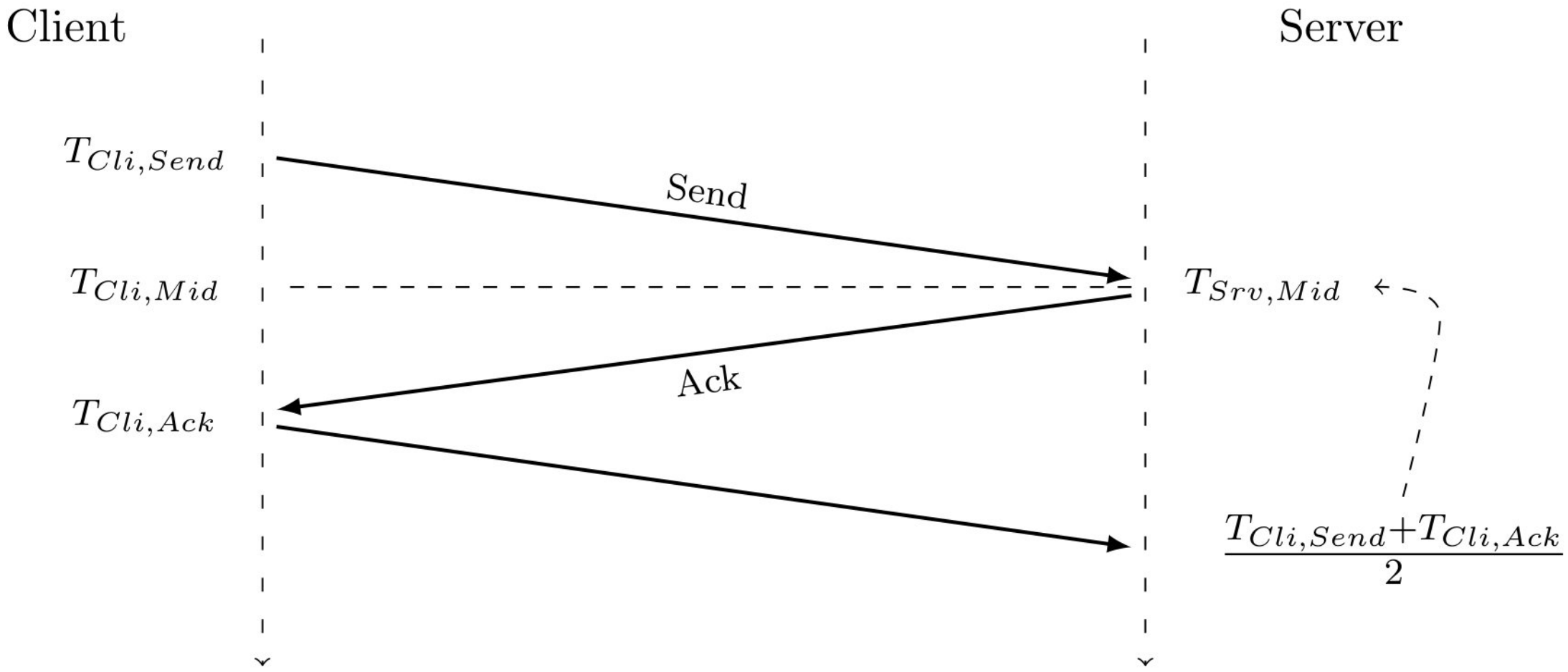
1. Network Latency
2. Staleness
3. Co-location Determination



1. Network Latency

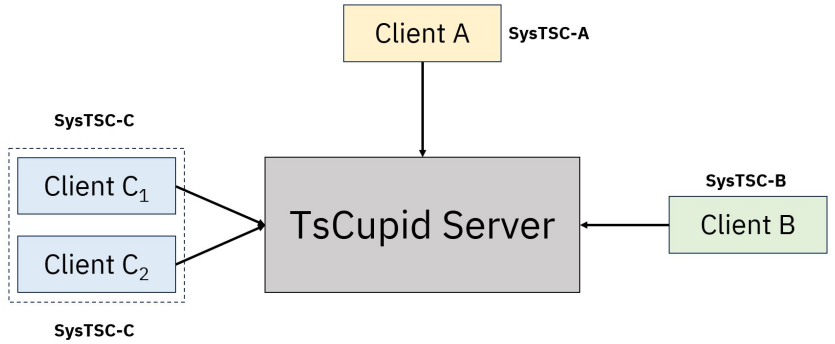


1. Network Latency

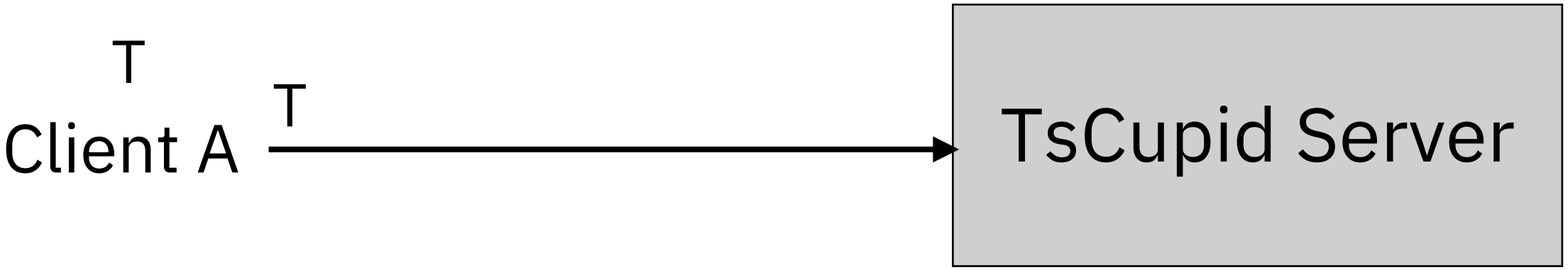


Three Challenges

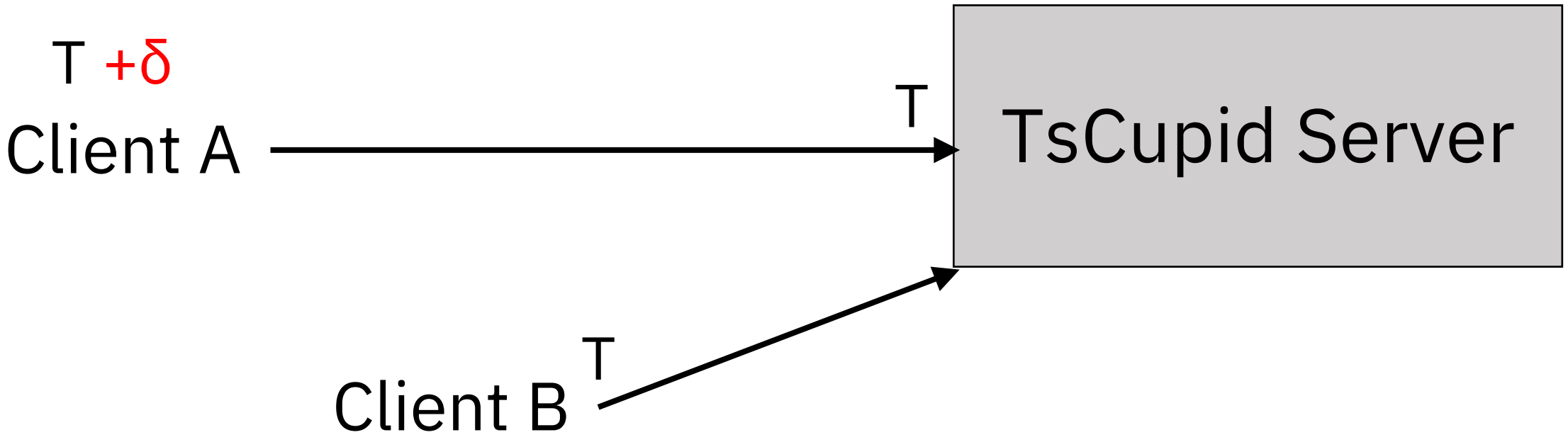
1. Network Latency
2. Staleness
3. Co-location Determination



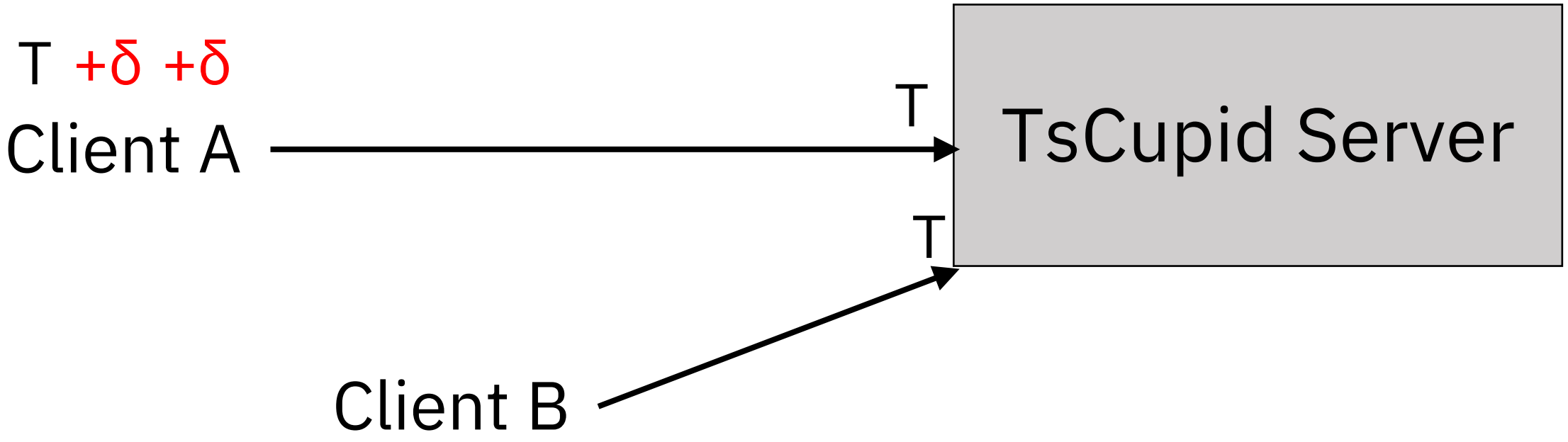
2. Staleness



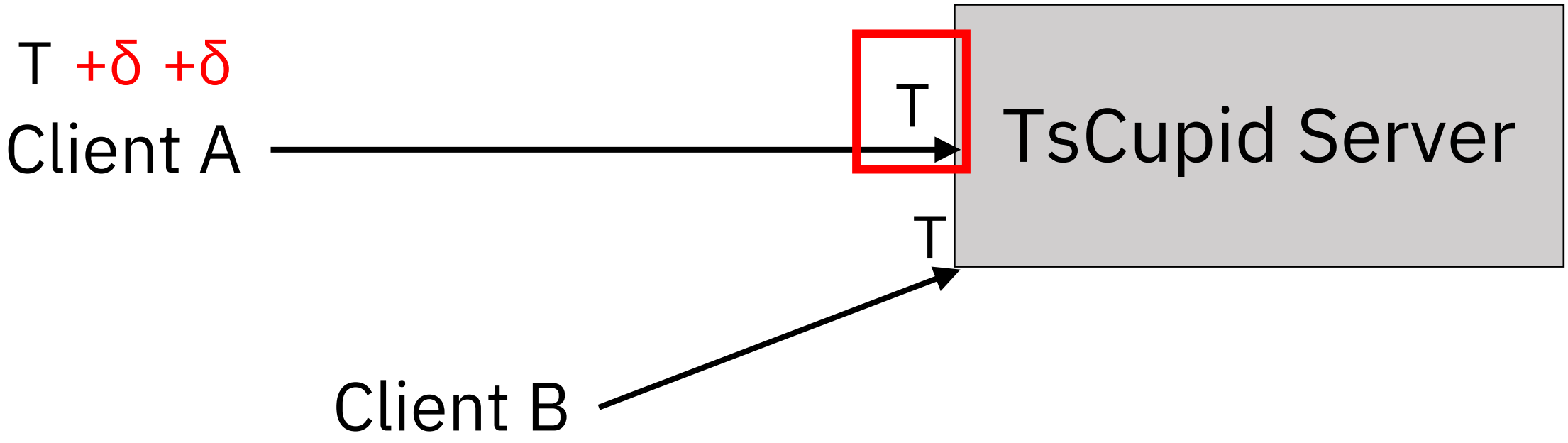
2. Staleness



2. Staleness



2. Staleness



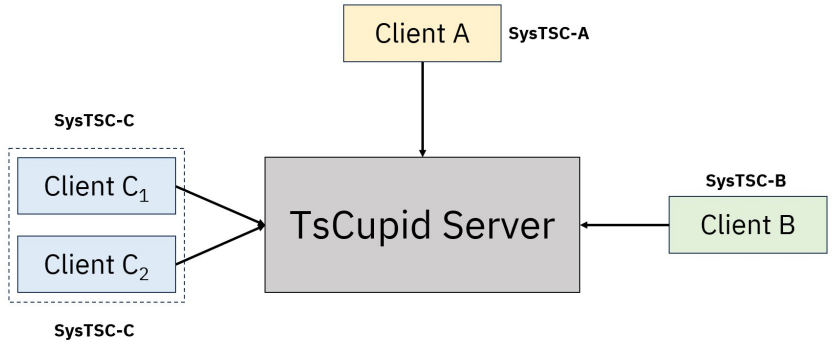
2. Staleness

Normalized Timestamp

$$T2'_A = T1_A + [\frac{Freq_A}{Freq_{Srv}} \times (T2_{Srv} - T1_{Srv})]$$

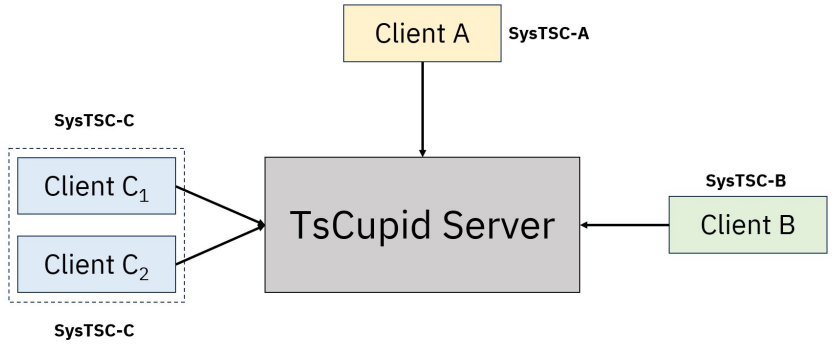
Three Challenges

1. Network Latency
2. Staleness
3. Co-location Determination



Three Challenges

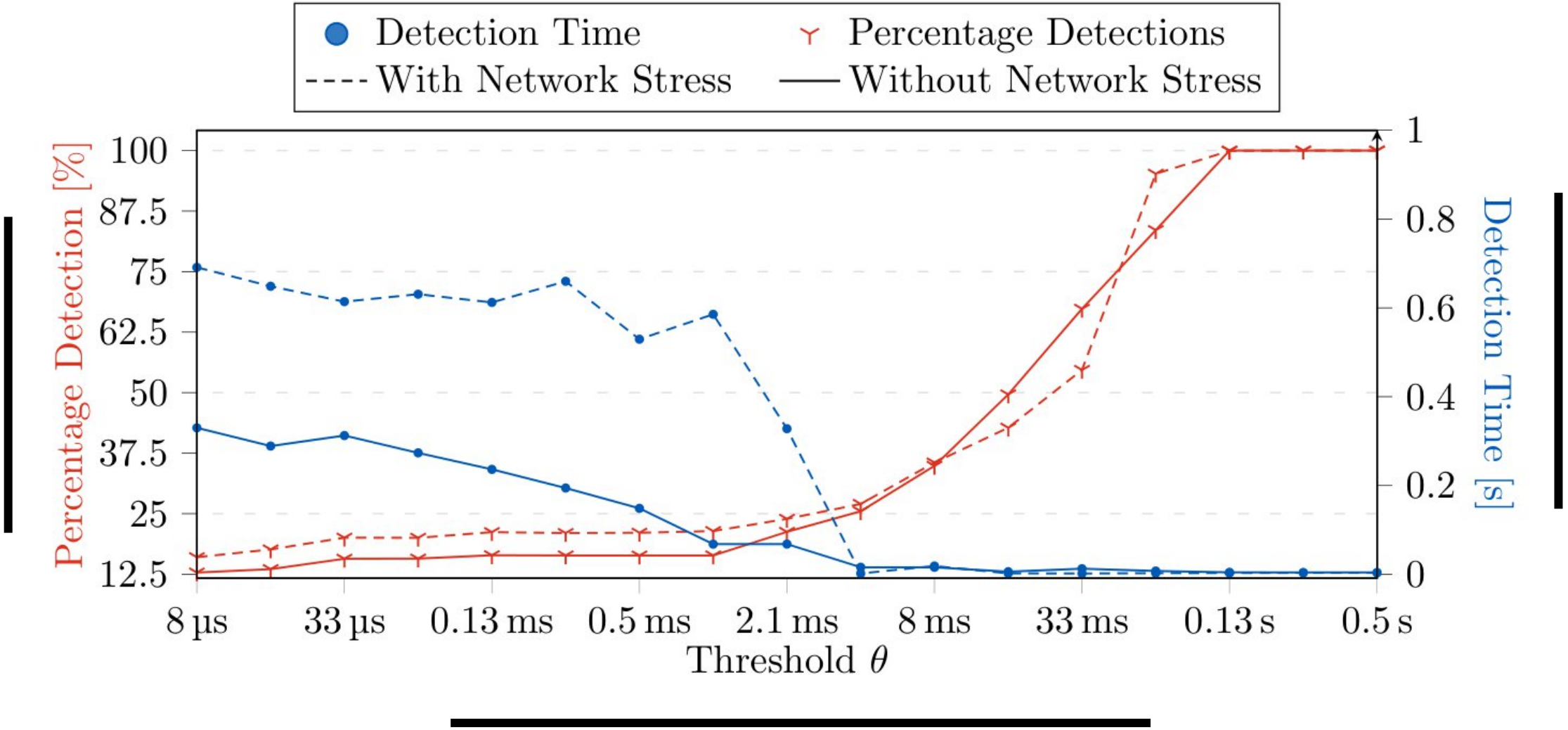
1. Network Latency
2. Staleness
3. Co-location Determination

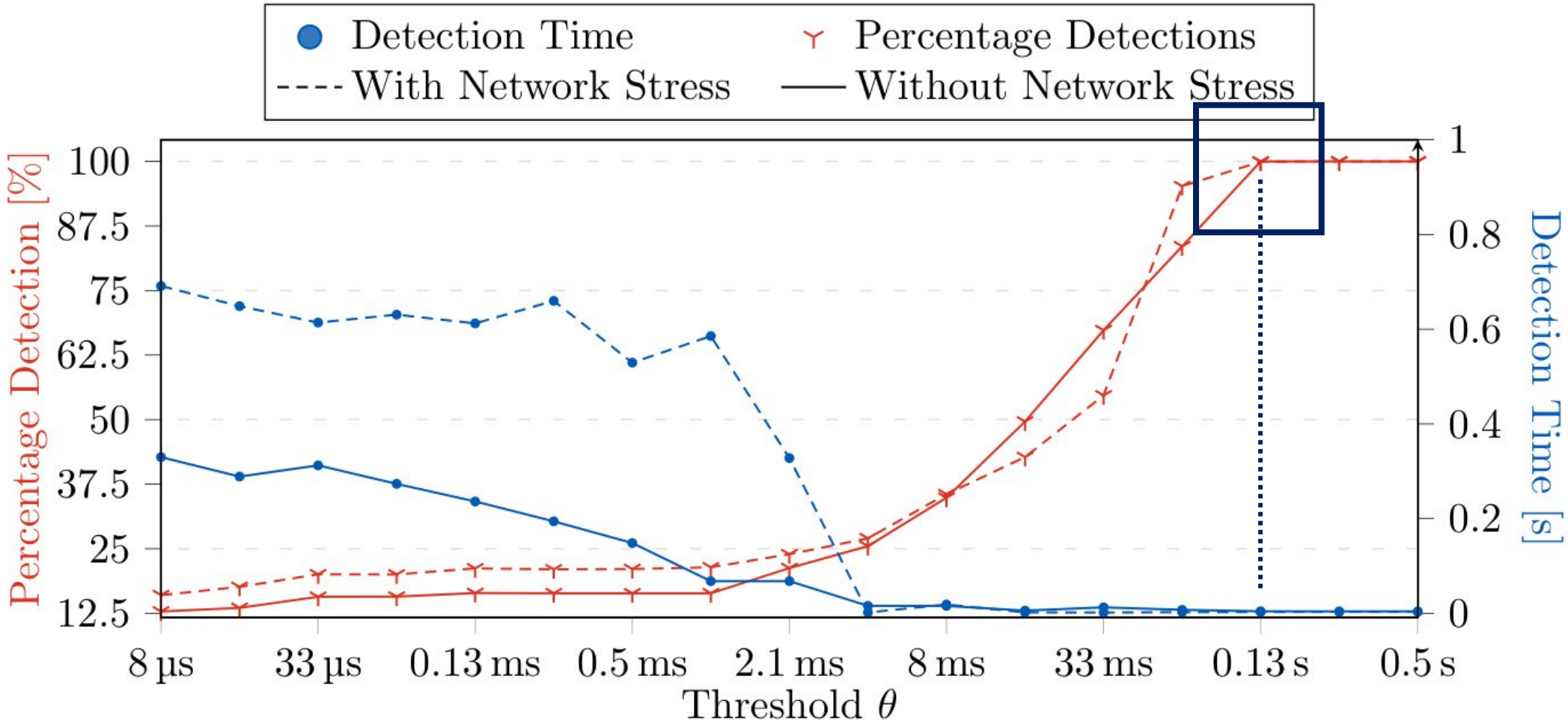


$$| T_A - T_B | < \theta$$

Experimental Setup

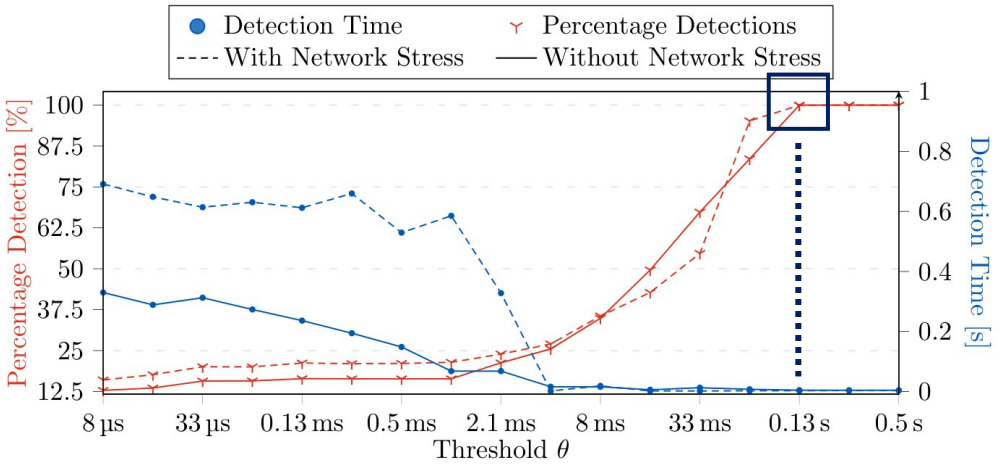
- Local AMD EPYC 7313P
- 16 co-located SEV-SNP VMs
- 17 thresholds θ for co-location determination from 8µs to 0.5s
- 100 experiments with and without network stress
- 3400 experiments in total


Detection Time
Percentage Detections
With Network Stress
Without Network Stress
Percentage Detection [%]
100
87.5
75
62.5
50
37.5
25
12.5
1
0.8
0.6
0.4
0.2
0
Detection Time [s]
8 µs
33 µs
0.13 ms
0.5 ms
2.1 ms
8 ms
33 ms
0.13 s
0.5 s
Threshold θ


Detection Time
Percentage Detections
With Network Stress
Without Network Stress
Percentage Detection [%]
100
87.5
75
62.5
50
37.5
25
12.5
1
0.8
0.6
0.4
0.2
0
Detection Time [s]
8 µs
33 µs
0.13 ms
0.5 ms
2.1 ms
8 ms
33 ms
0.13 s
0.5 s
Threshold θ

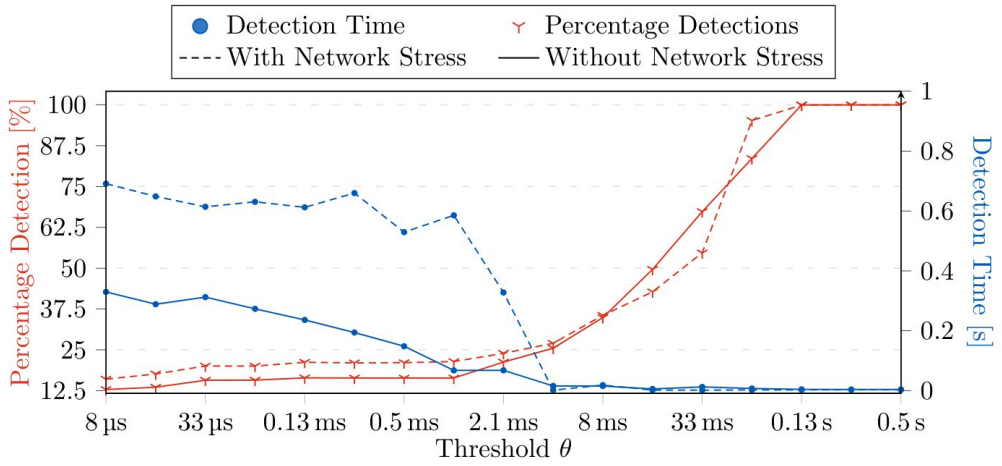
TsCupid Evaluation

- Threshold for co-location determination θ = 0.13s
- 100% detection rate
- 4ms on average to detect



TsCupid Evaluation

- Threshold for co-location determination θ = 0.13s
- 100% detection rate
- 4ms on average to detect

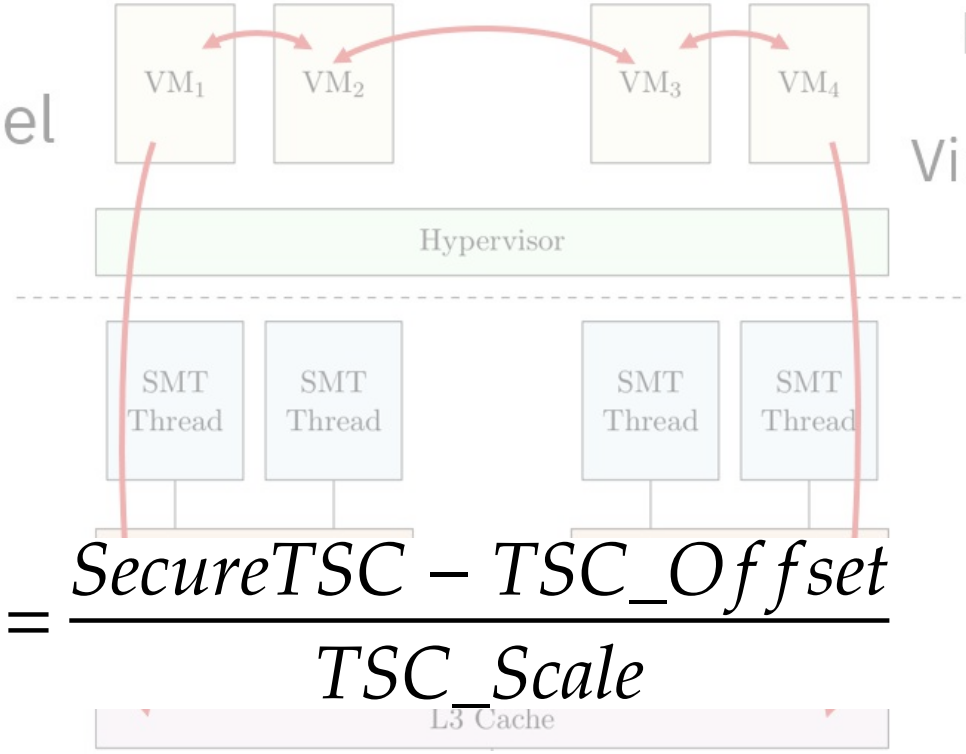


Much shorter than power-on delay options on server motherboards (usually ~1s)

How Do We Fix This?

Mitigation

Cross VM Side-Channel Attacks



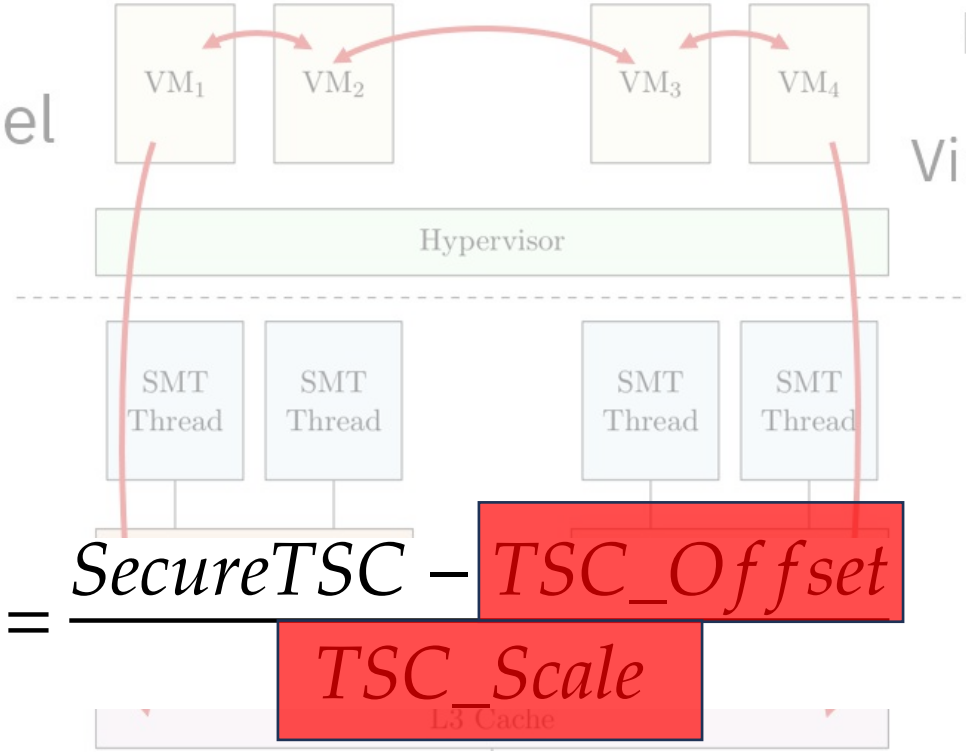
Prerequisite: Co-located Virtual Machines

$$SystemTSC = \frac{SecureTSC - TSC\_Offset}{TSC\_Scale}$$

Same for every SEV-SNP VM on the same physical machine!

Mitigation

Cross VM Side-Channel Attacks



Prerequisite: Co-located Virtual Machines

$$SystemTSC = \frac{SecureTSC - TSC\_Offset}{TSC\_Scale}$$

Same for every SEV-SNP VM on the same physical machine!

Mitigation

Two parameters about SecureTSC:

- SecureTSC Scale
- SecureTSC Offset

Used for Live Migration

$$SystemTSC = \frac{SecureTSC - TSC\_Offset}{TSC\_Scale}$$

Same for every SEV-SNP VM on the same physical machine!

Responsible Disclosure

- From March - April 2024, we reported this issue
- AMD acknowledged our findings, but stated that co-location falls outside their current threat model for SEV-SNP VMs and that they will not fix it.

Conclusion

- AMD added SecureTSC without thinking about potential side channels
- This is a pretty common problem
- Please include side channels into your threat models

Acknowledgments

This research was made possible by generous funding from:

Funded by the European Union

fssec erc
European Research Council
Established by the European Commission

SPyCoDe FWF
Der Wissenschaftsfonds.

NeRAM FWF
Der Wissenschaftsfonds.

DFG Deutsche Forschungsgemeinschaft

Red Hat

Google

intel

Supported in part by the European Research Council(ERC project FSSec 101076409) and the Austrian Science Fund (FWF SFB project SPyCoDe 10.55776/F85 and FWF project NeRAM I6054). Additional funding was provided by generous gifts from Red Hat, Google, and Intel. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding parties.

Not So Secure TSC

Co-location Detection on AMD SEV-SNP Confidential Virtual Machines

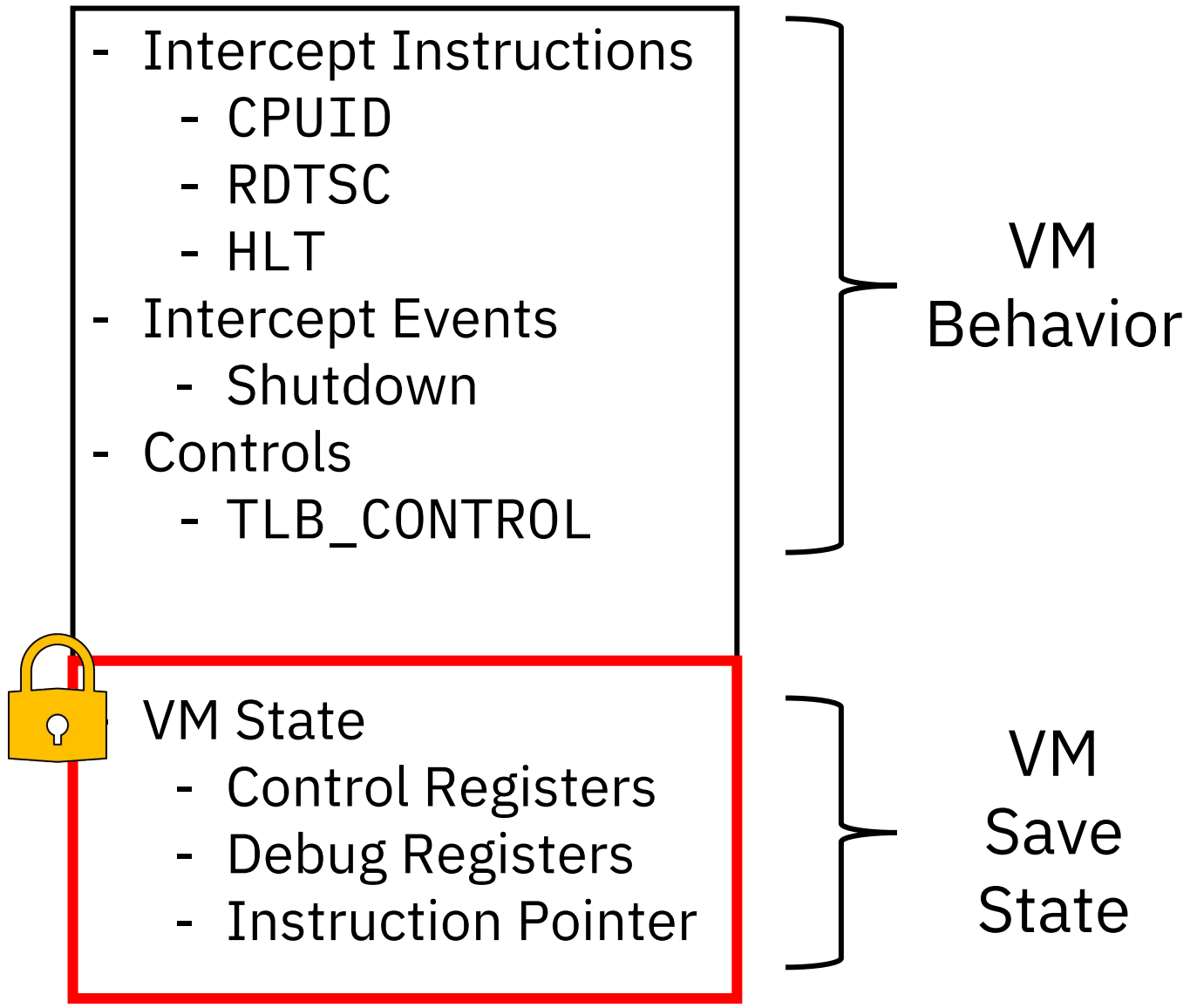
Jonas Juffinger, **Sudheendra Raghav Neela**, Daniel Gruss

jonasjuffinger.com snee.la gruss.cc

23rd International Conference on Applied Cryptography and Network Security

> isec.tugraz.at

AMD SEV with Encrypted State (SEV-ES)



Like Clockwork: A Systematic Analysis of AMD SEV-SNP's SecureTSC

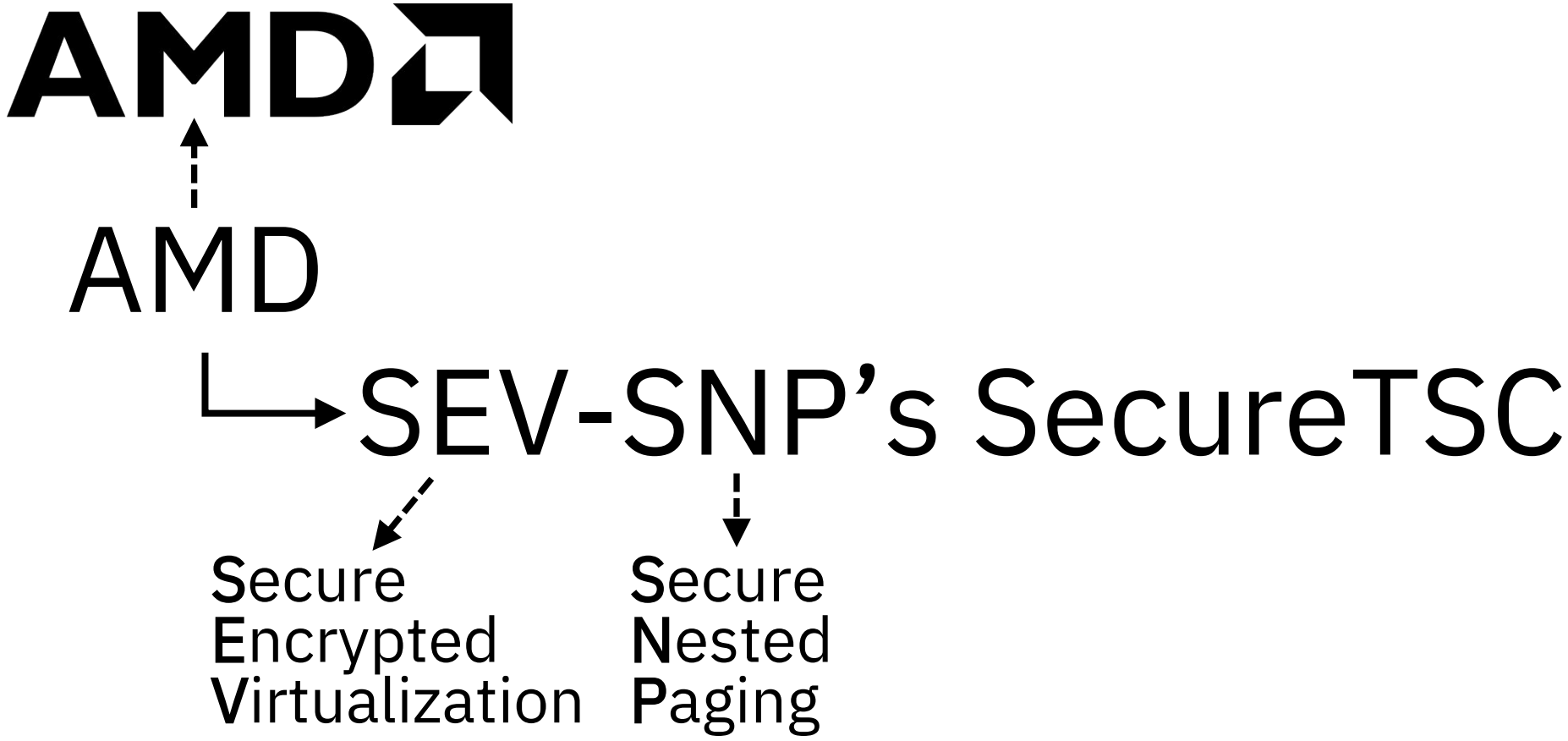
Sudheendra Raghav Neela

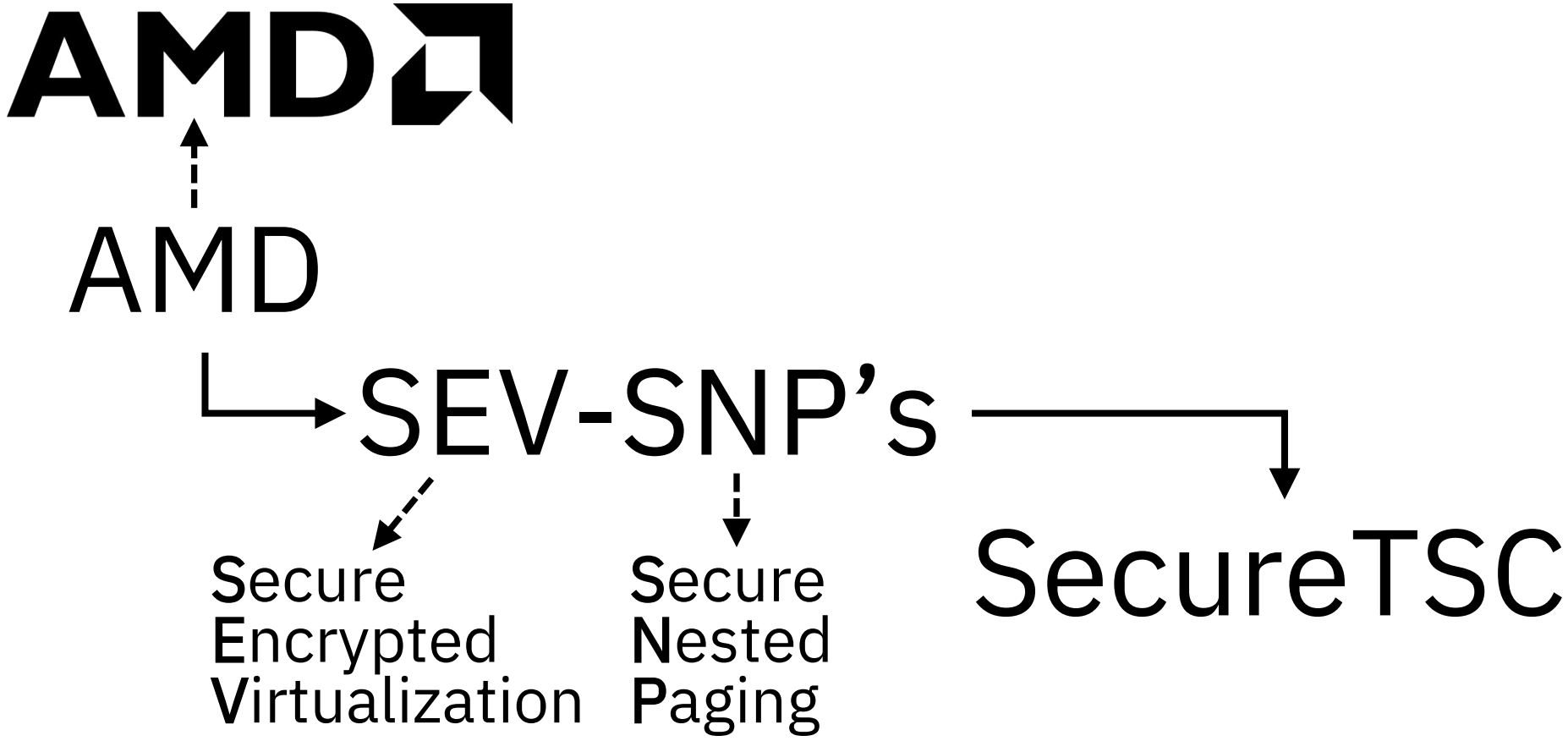
Advisors:
Jonas Juffinger, Daniel Gruss

12.12.24

AMD SEV-SNP’s SecureTSC


AMD
AMD
SEV-SNP's SecureTSC
Secure
Encrypted
Virtualization

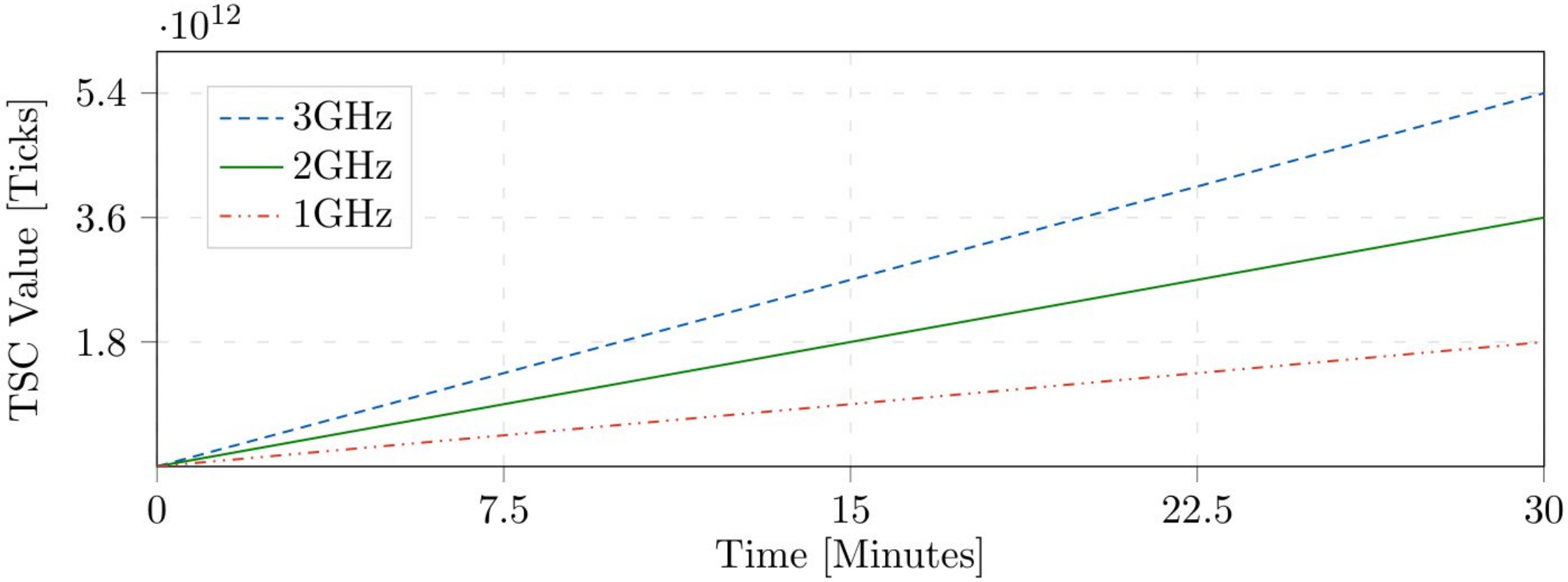

AMD
AMD
SEV-SNP's SecureTSC
Secure
Encrypted
Virtualization
Secure
Nested
Paging

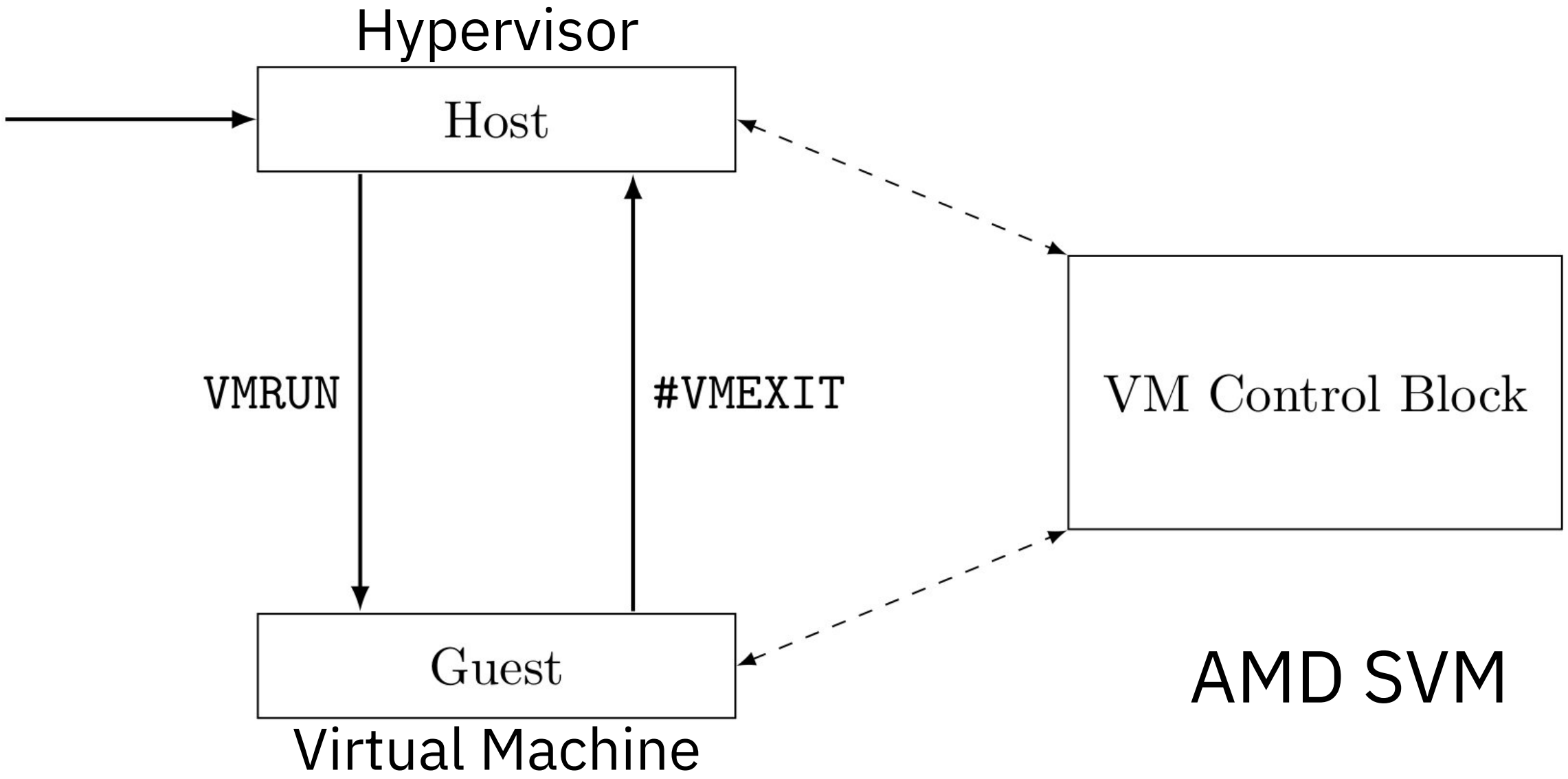

AMD
AMD
SEV-SNP's
Secure
Encrypted
Virtualization
Secure
Nested
Paging
SecureTSC

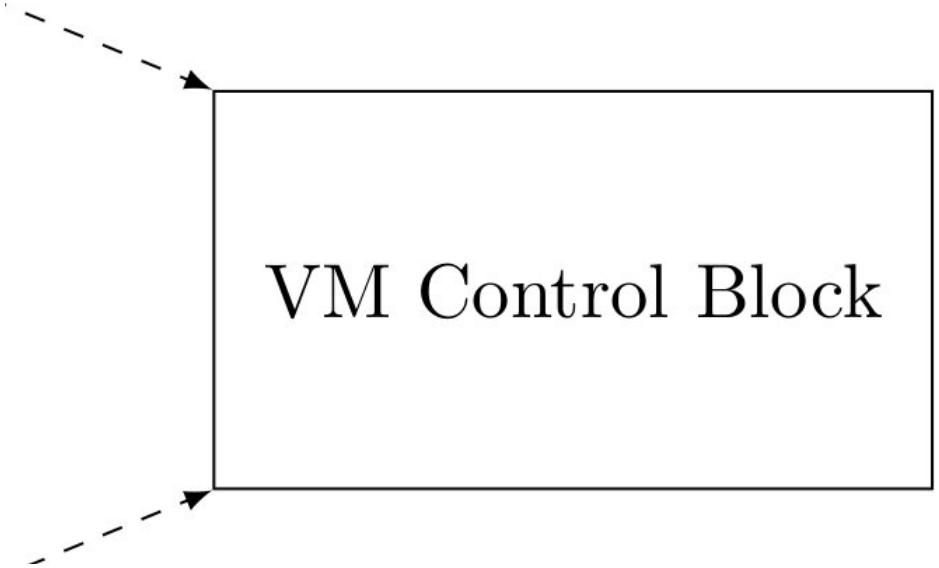
SecureTSC

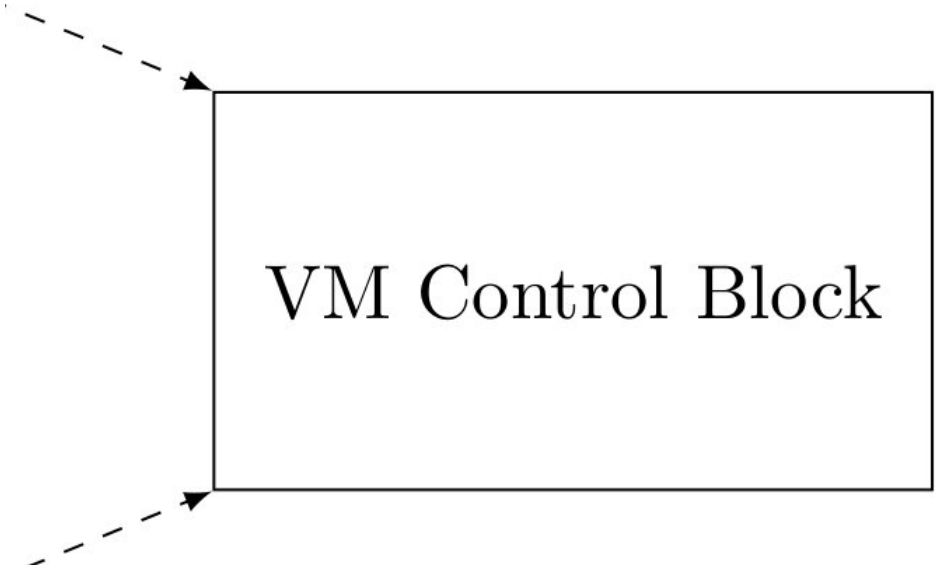
- Secure method for SEV-SNP VMs to access the Time-Stamp Counter.
- The hypervisor's TSC and VM's SecureTSC is independent.
- The hypervisor cannot influence the guest's SecureTSC, except through the Desired TSC Frequency

Hypervisor-Set Desired TSC Frequency



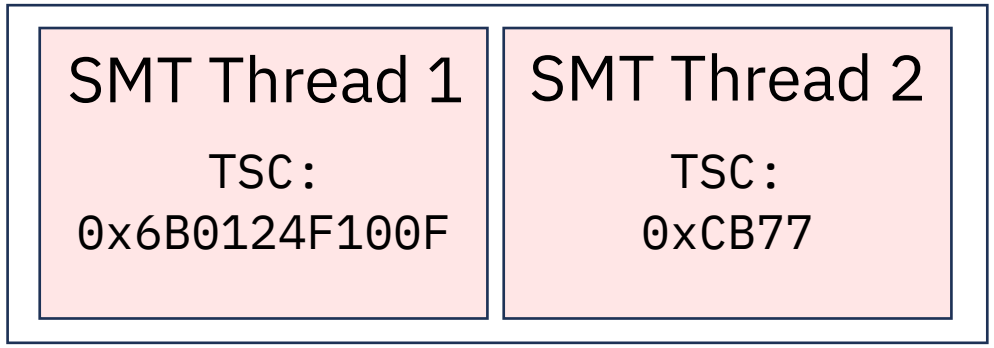

Hypervisor
Host
VMRUN
#VMEXIT
VM Control Block
Guest
Virtual Machine
AMD SVM


VM Control Block

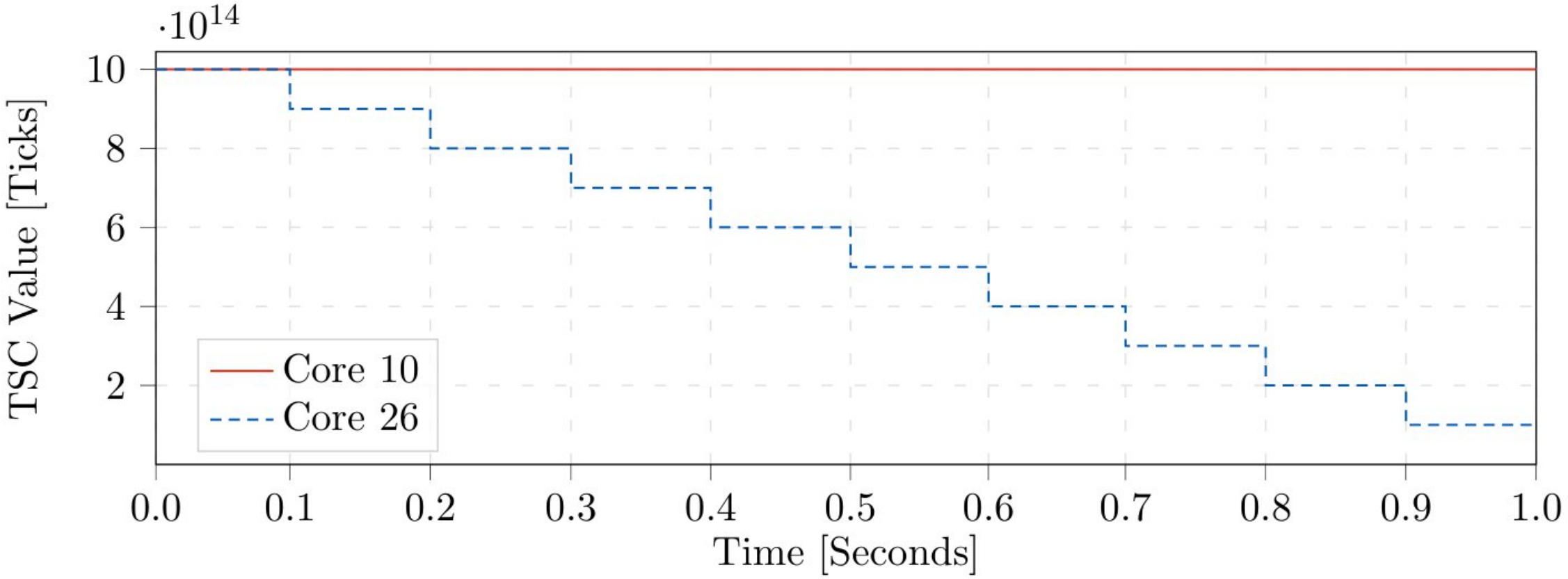


- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL
- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

The Time-Stamp Counter (TSC)



The Time-Stamp Counter (TSC)



Virtual Machine

```
XOR RAX, RAX;
XOR RCX, RCX;
XOR RDX, RDX;

MOV RAX, 0x0124;
MOV RDX, 0x4210;

MUL RDX;
MOV RCX, RDX;

RDTSC;

MOV RAX, RDI;
SAL RAX, 32;
OR  RAX, RDX;
```

Hypervisor

CPU

- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL

- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

Virtual Machine

```
XOR RAX, RAX;
XOR RCX, RCX;
XOR RDX, RDX;

MOV RAX, 0x0124;
MOV RDX, 0x4210;

MUL RDX;
MOV RCX, RDX;

RDTSC;

MOV RAX, RDI;
SAL RAX, 32;
OR  RAX, RDX;
```

Hypervisor

CPU

VM Behavior

- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL

- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

Virtual Machine

```
XOR RAX, RAX;
XOR RCX, RCX;
XOR RDX, RDX;

MOV RAX, 0x0124;
MOV RDX, 0x4210;

MUL RDX;
MOV RCX, RDX;

RDTSC;

MOV RAX, RDI;
SAL RAX, 32;
OR  RAX, RDX;
```

Hypervisor

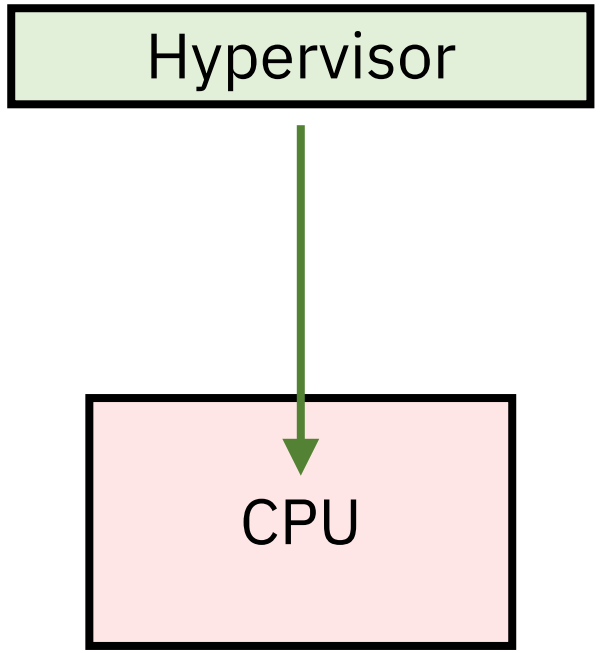
CPU

- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL

- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

Virtual Machine

```
XOR RAX, RAX;
XOR RCX, RCX;
XOR RDX, RDX;

MOV RAX, 0x0124;
MOV RDX, 0x4210;

MUL RDX;
MOV RCX, RDX;

RDTSC;

MOV RAX, RDI;
SAL RAX, 32;
OR  RAX, RDX;
```



- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL

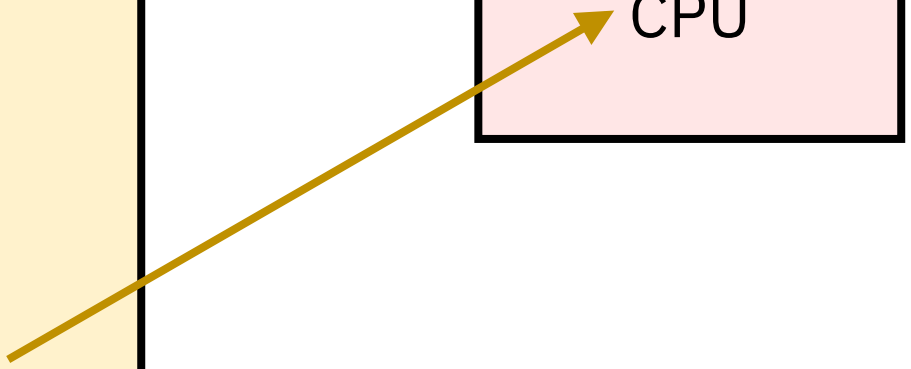
- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

Virtual Machine

```
XOR RAX, RAX;
XOR RCX, RCX;
XOR RDX, RDX;

MOV RAX, 0x0124;
MOV RDX, 0x4210;

MUL RDX;
MOV RCX, RDX;

RDTSC;

MOV RAX, RDI;
SAL RAX, 32;
OR  RAX, RDX;
```

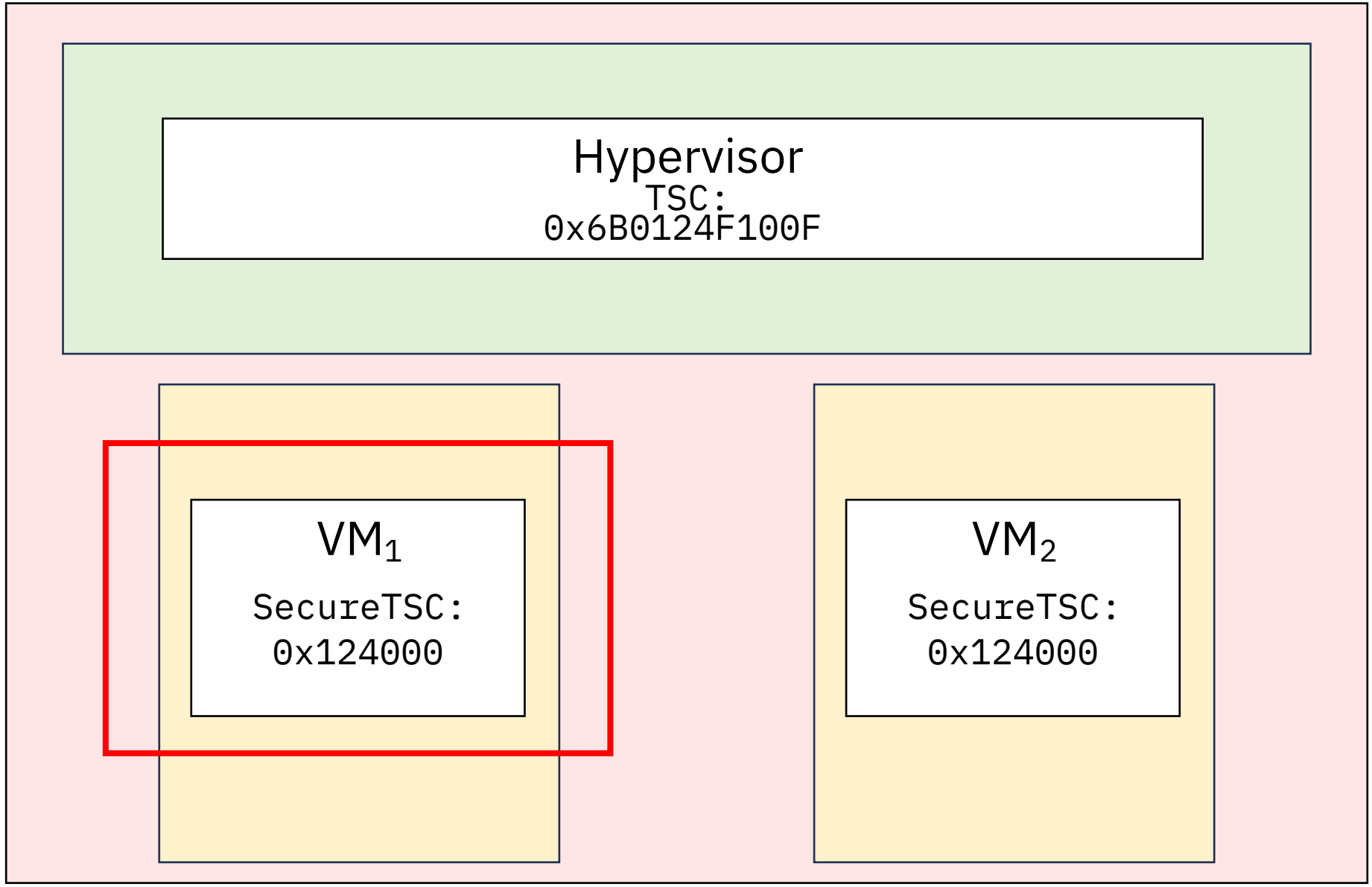
Hypervisor

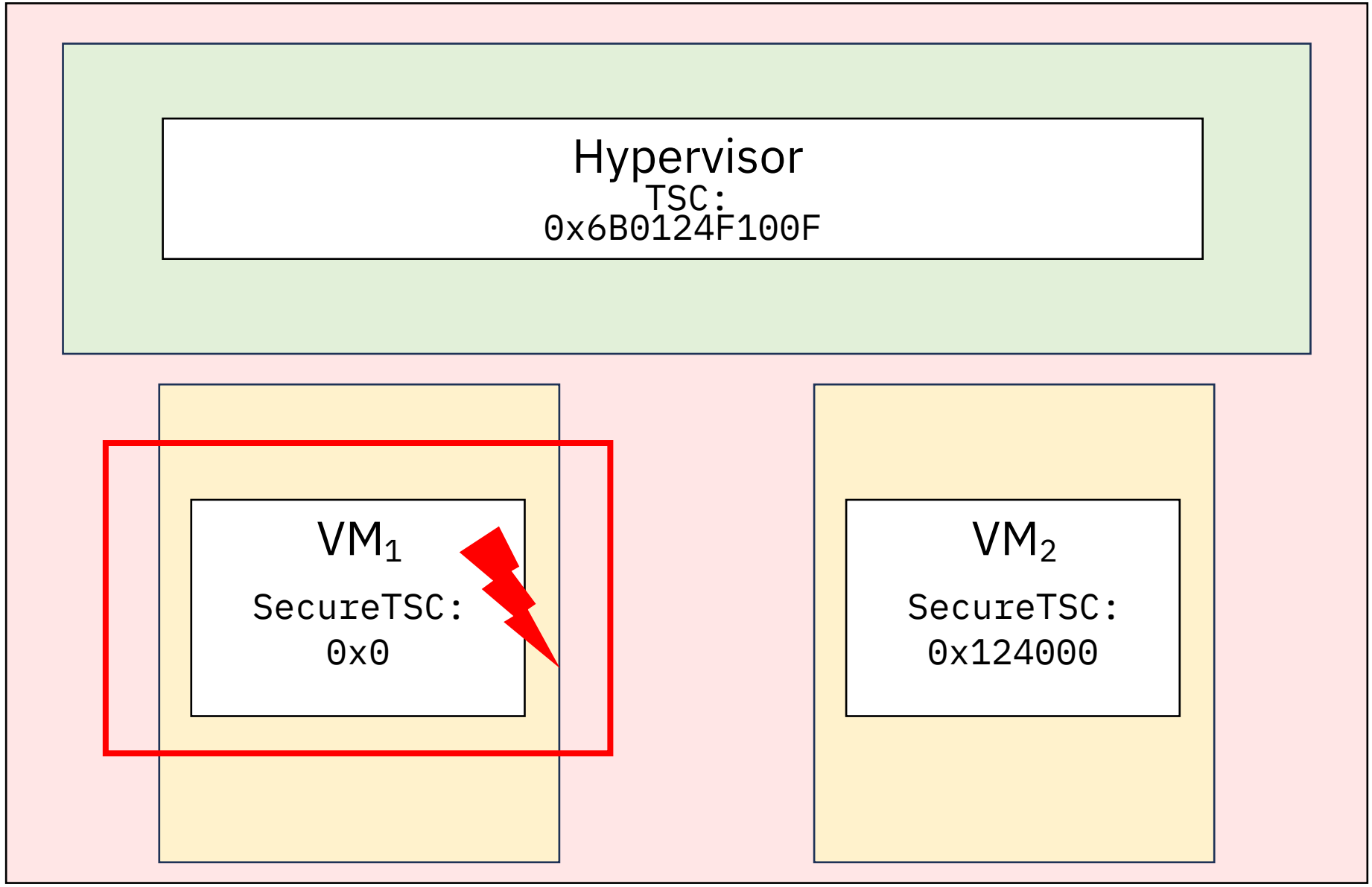
CPU

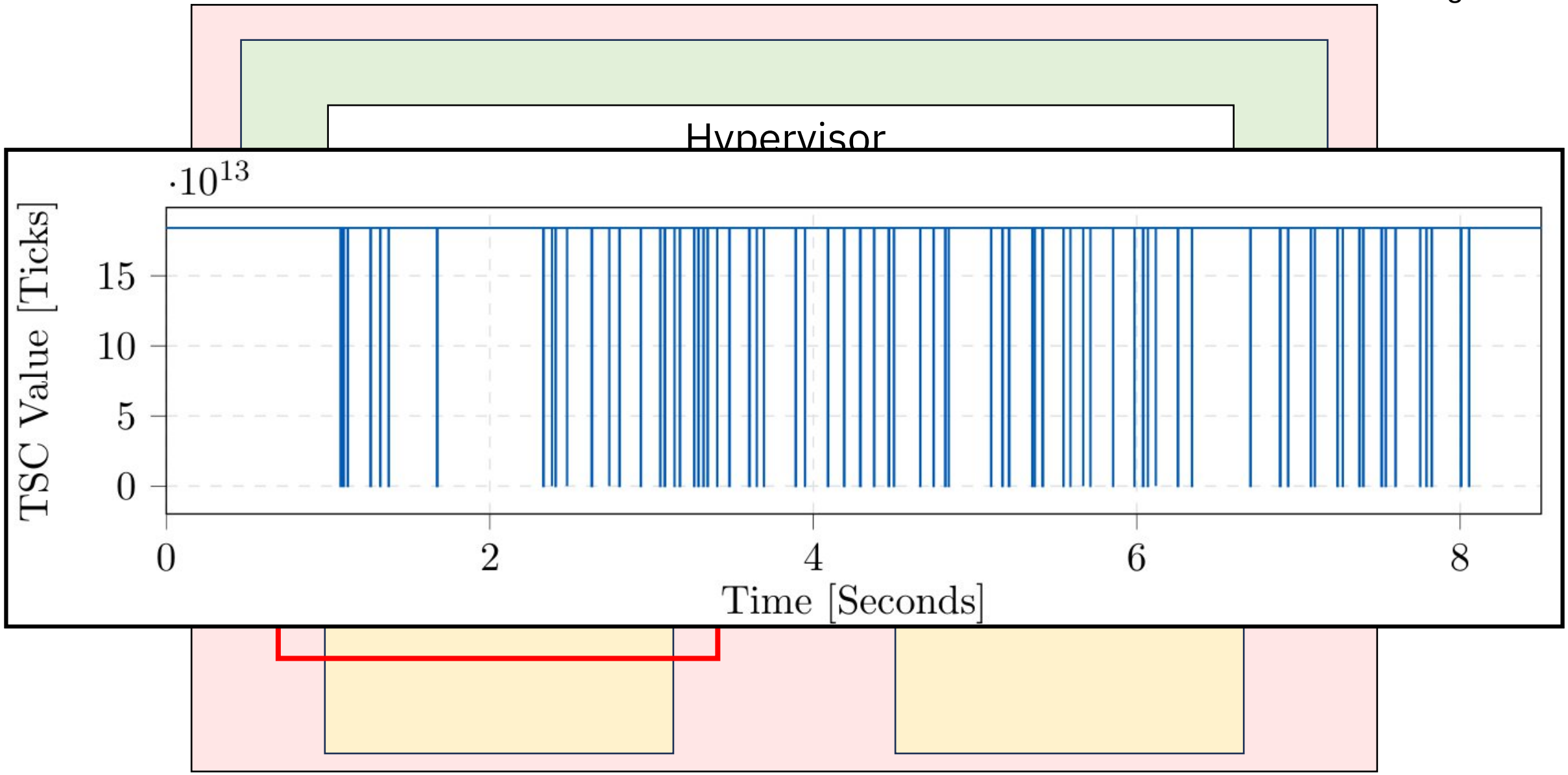


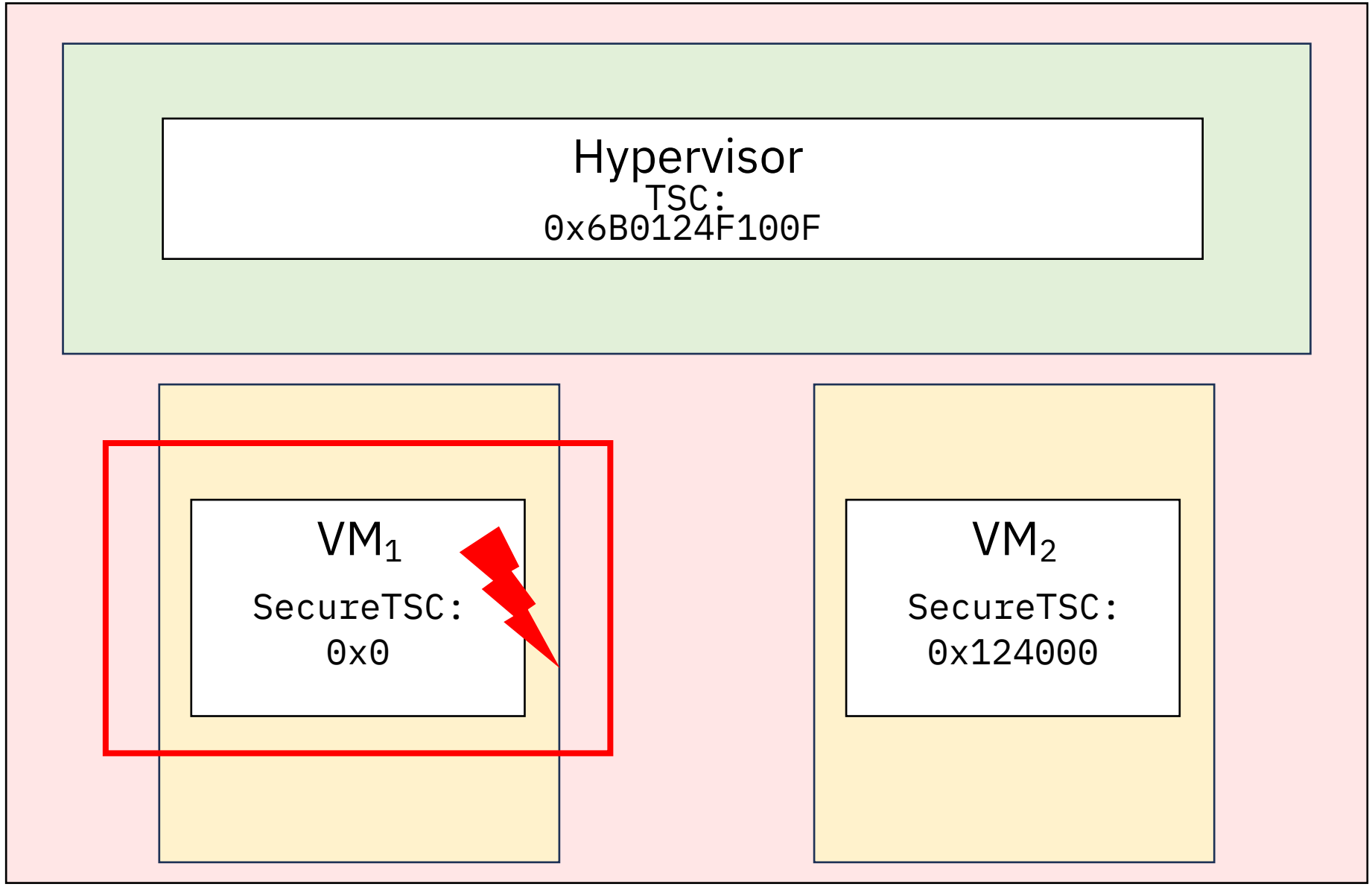
- Intercept Instructions
  - CPUID
  - RDTSC
  - HLT
- Intercept Events
  - Shutdown
- Controls
  - TLB_CONTROL

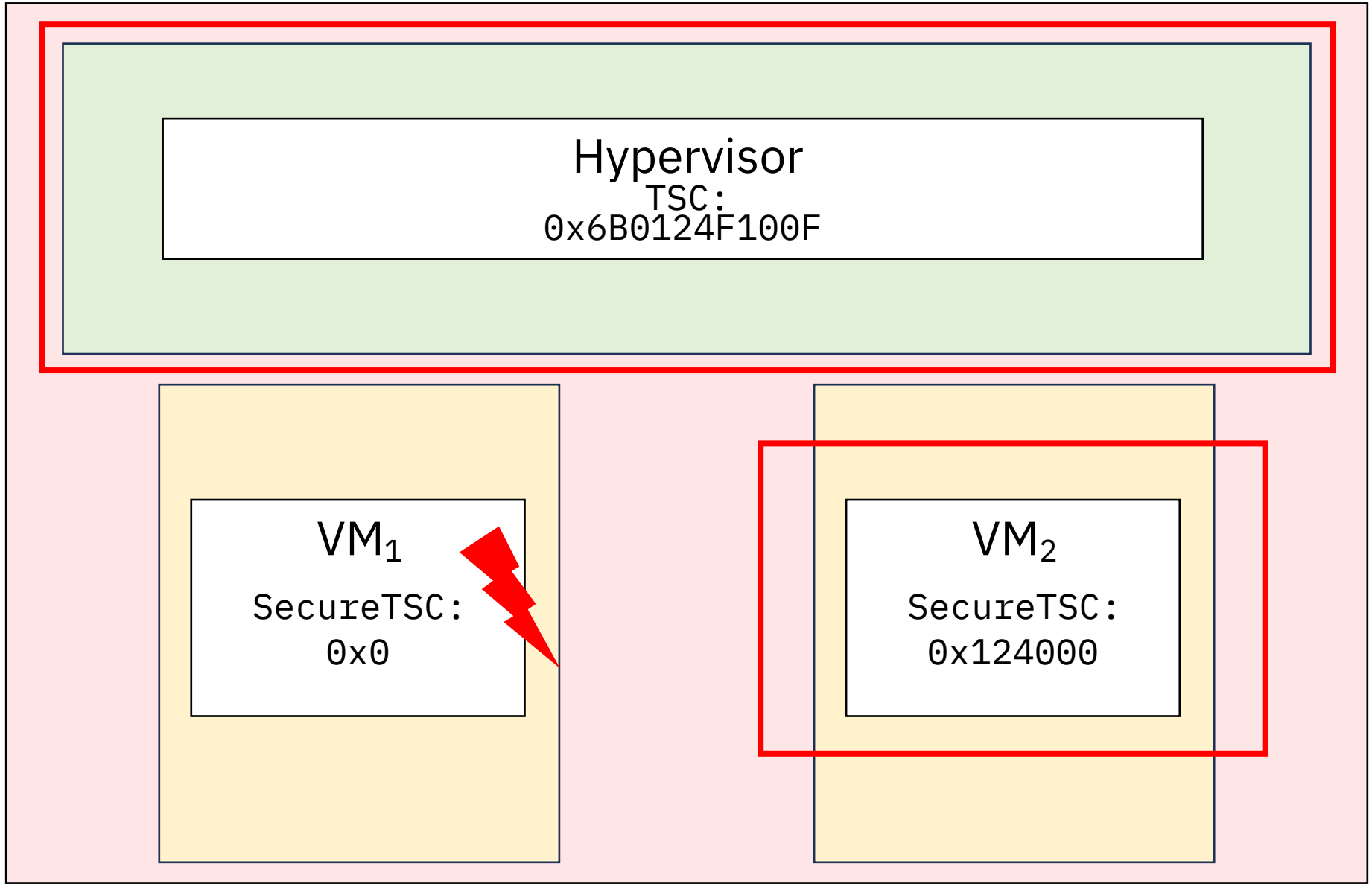
- VM State
  - Control Registers
  - Debug Registers
  - Instruction Pointer

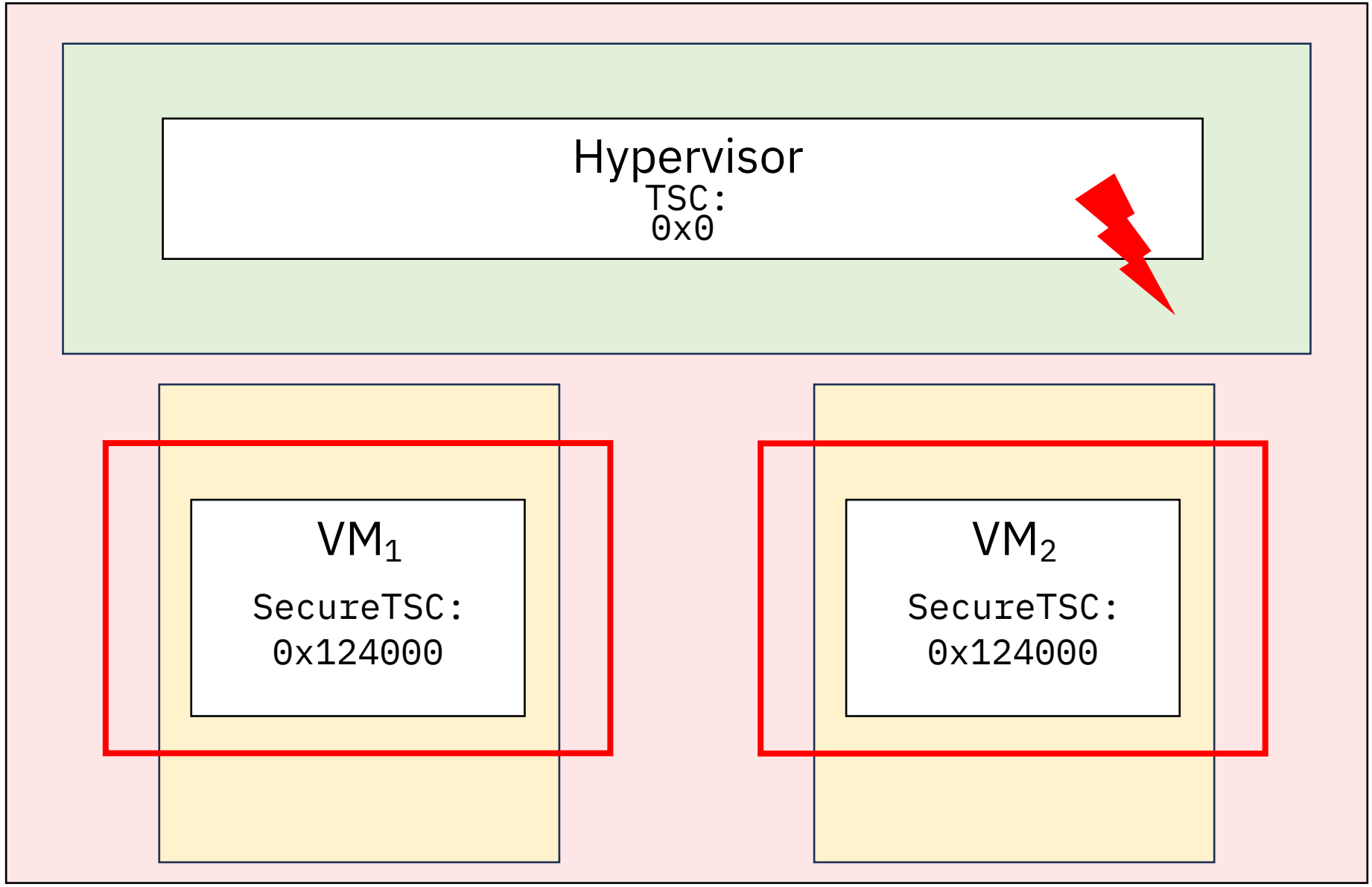

Hypervisor
TSC:
0x6B0124F100F
VM1
SecureTSC:
0x124000
VM2
SecureTSC:
0x124000


Hypervisor
TSC:
0x6B0124F100F
VM1
SecureTSC:
0x0
VM2
SecureTSC:
0x124000


Hypervisor
$\cdot 10^{13}$
TSC Value [Ticks]
15
10
5
0
0
2
4
6
8
Time [Seconds]


Hypervisor
TSC:
0x6B0124F100F
VM1
SecureTSC:
0x0
VM2
SecureTSC:
0x124000

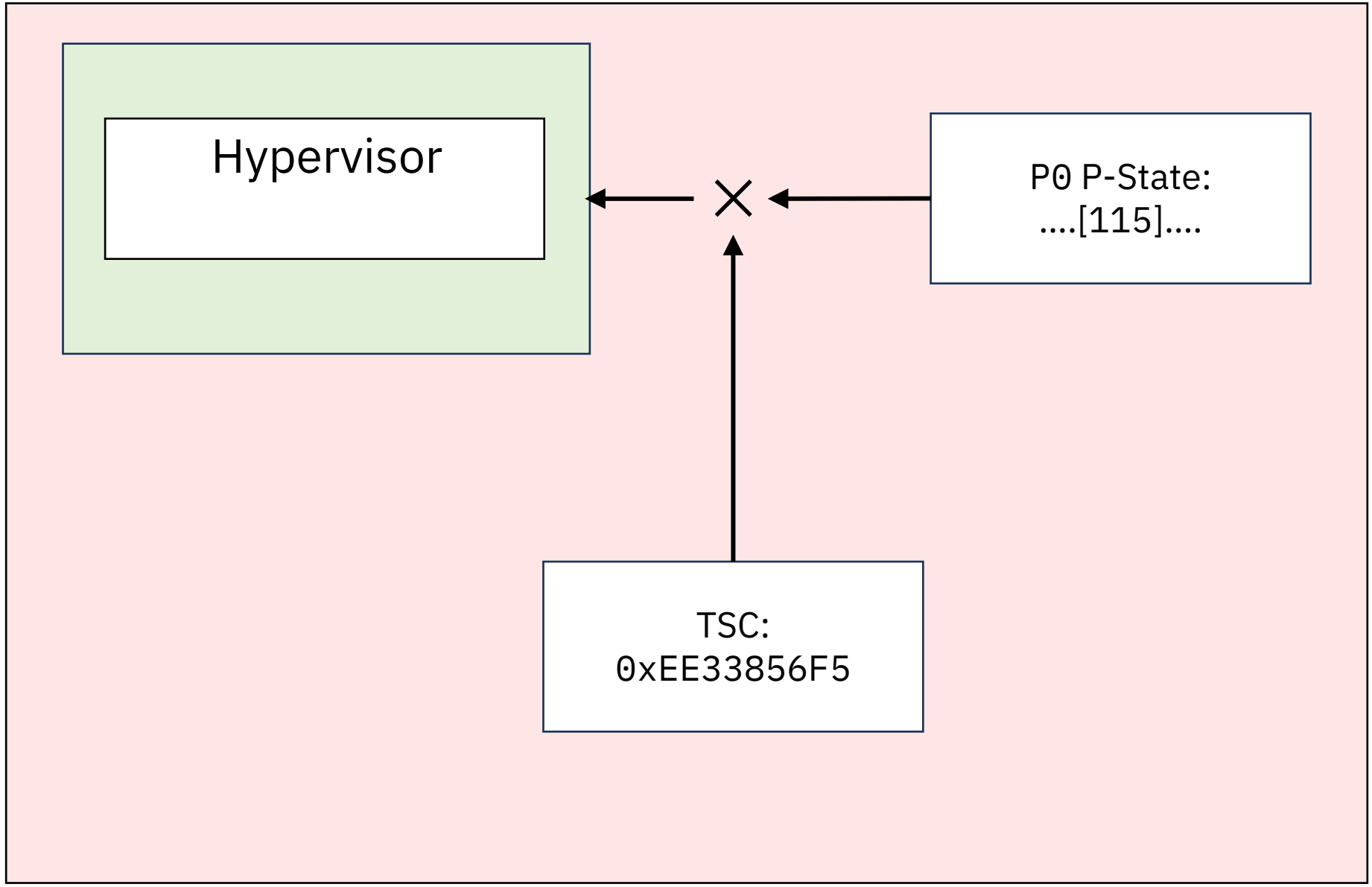

Hypervisor
TSC:
0x6B0124F100F
VM1
SecureTSC:
0x0
VM2
SecureTSC:
0x124000

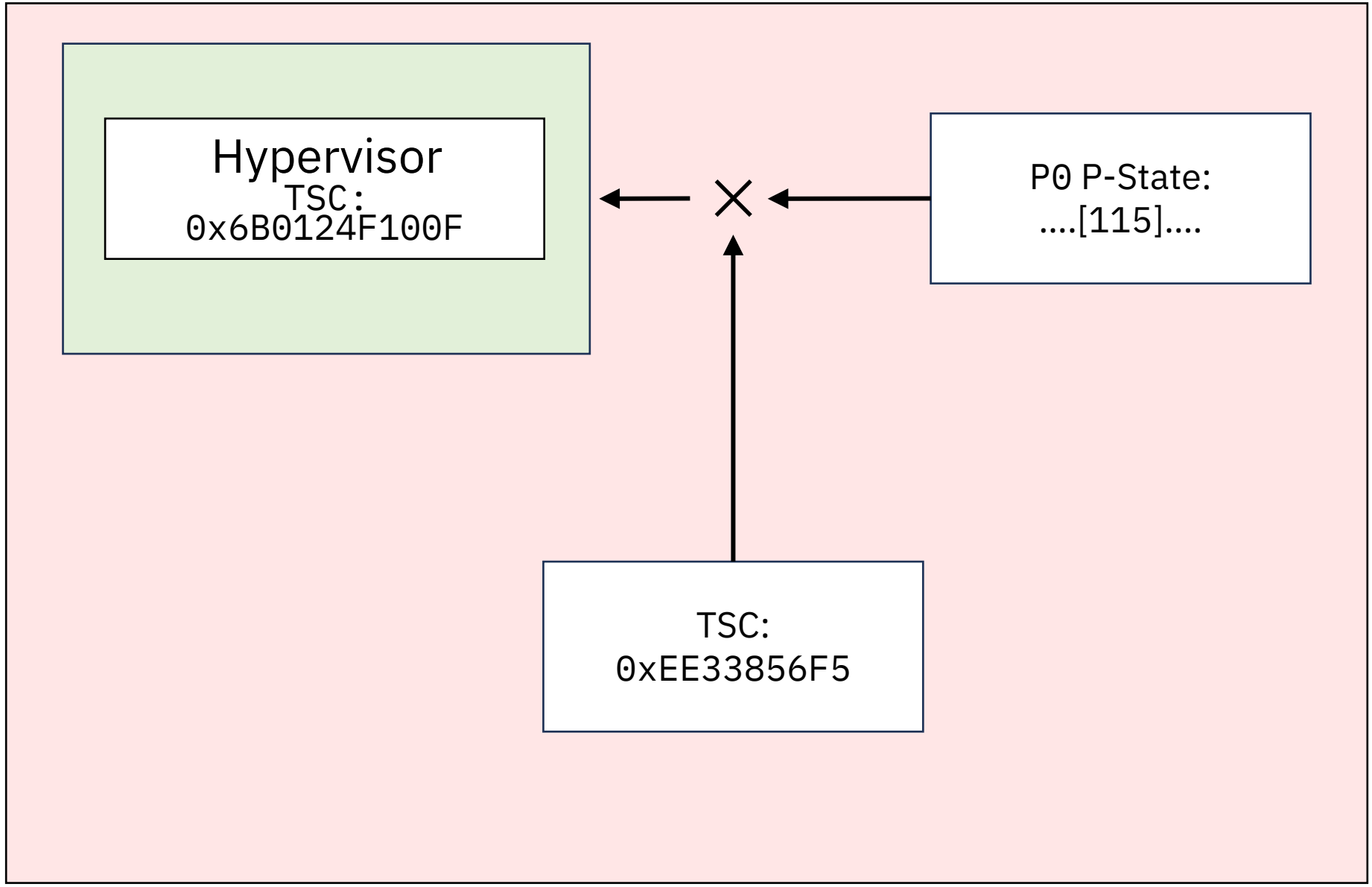

Hypervisor
TSC:
0x0
VM1
SecureTSC:
0x124000
VM2
SecureTSC:
0x124000

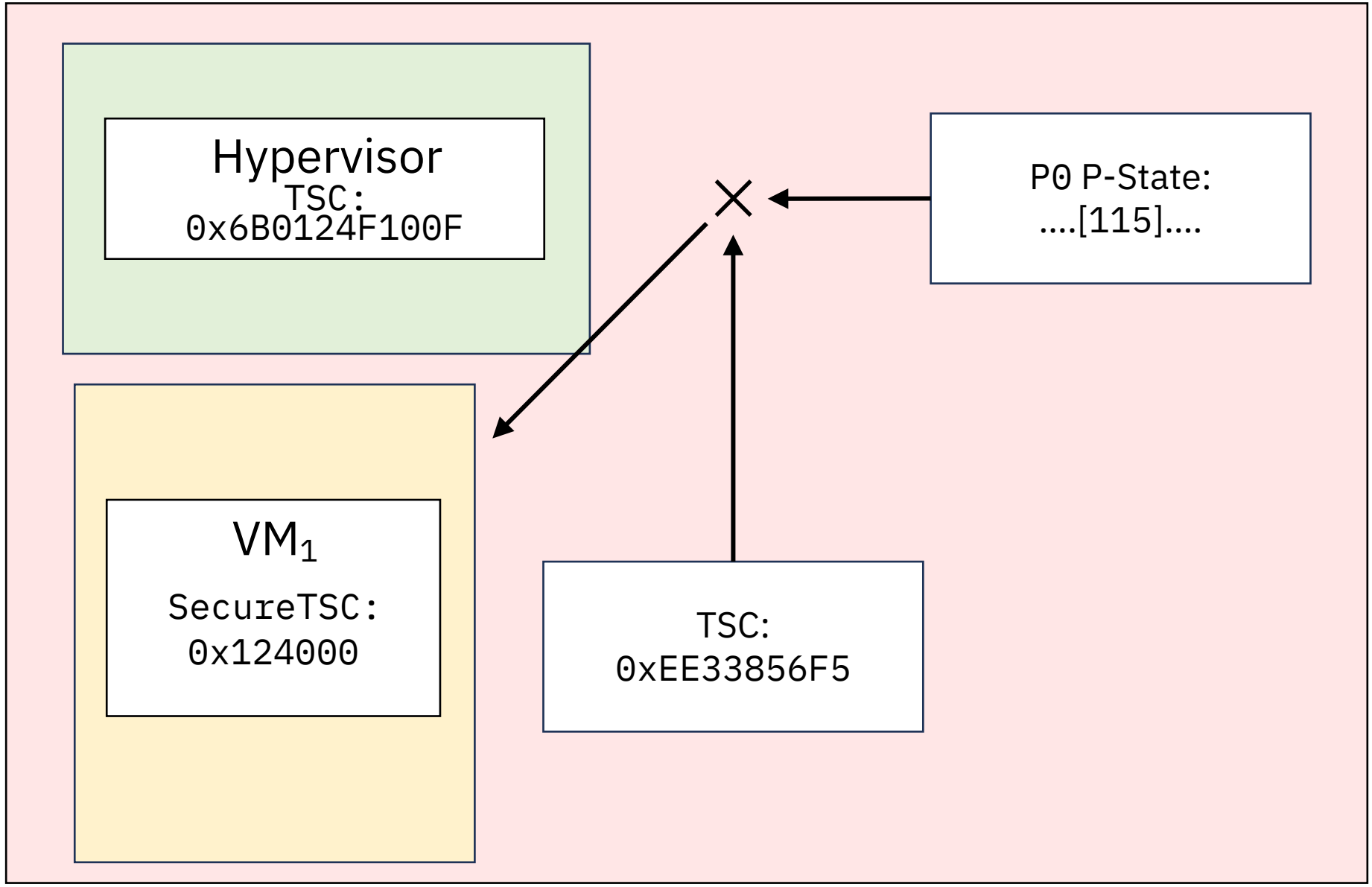
SecureTSC

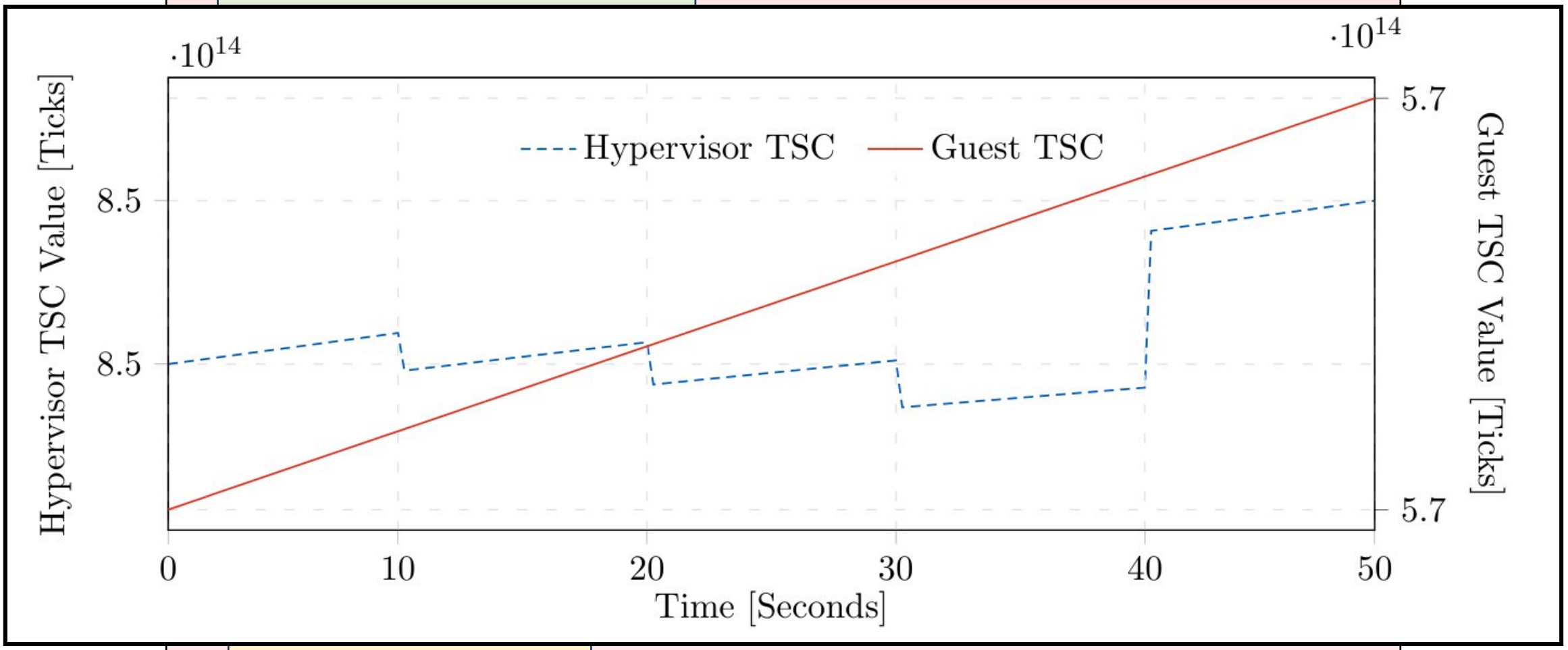
- Secure method for SEV-SNP VMs to access the Time-Stamp Counter.
- “Guests that run with SecureTSC enabled are not expected to perform writes to the TSC MSR. If such a write occurs, subsequent TSC values read are undefined.”
- “The P0 frequency value is not used…”

- AMD APM 15.36.18 Secure TSC


Hypervisor
P0 P-State:
....[115]....
TSC:
0xEE33856F5


Hypervisor
TSC:
0x6B0124F100F
P0 P-State:
....[115]....
TSC:
0xEE33856F5


Hypervisor
TSC:
0x6B0124F100F
P0 P-State:
....[115]....
VM1
SecureTSC:
0x124000
TSC:
0xEE33856F5


$\cdot 10^{14}$
$\cdot 10^{14}$
Hypervisor TSC Value [Ticks]
Guest TSC Value [Ticks]
Hypervisor TSC
Guest TSC
8.5
8.5
5.7
5.7
0
10
20
30
40
50
Time [Seconds]

Resetting SecureTSC

- Rebooting the Computer
- AMD Secure Processor
  - Firmware-Upgrade

Resetting SecureTSC

- Rebooting the Computer
- AMD Secure Processor
  - Firmware-Upgrade

Two parameters about SecureTSC:

- SecureTSC Scale
- SecureTSC Offset

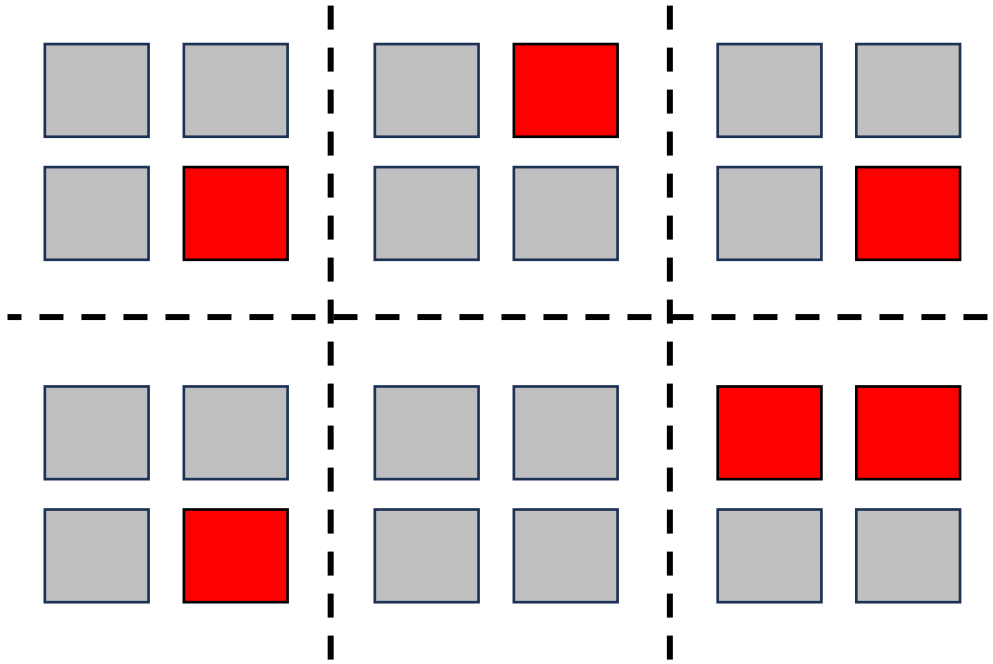
Resetting SecureTSC

- Rebooting the Computer
- AMD Secure Processor
  - Firmware-Upgrade

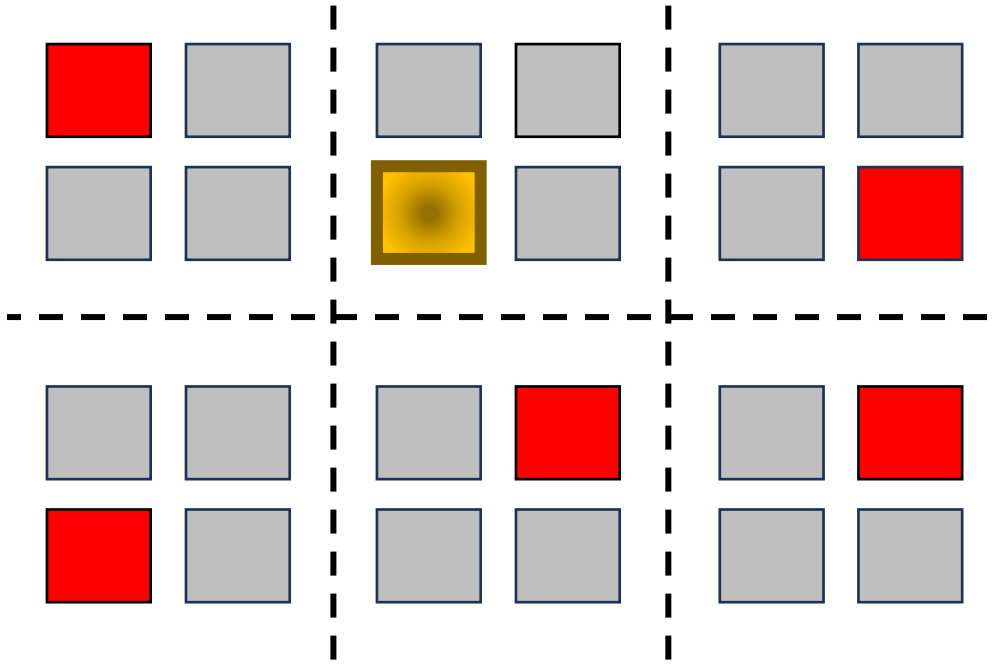
Two parameters about SecureTSC:

- SecureTSC Scale
- SecureTSC Offset

Used for Live Migration



Co-locating Attackers



Targeted Guest