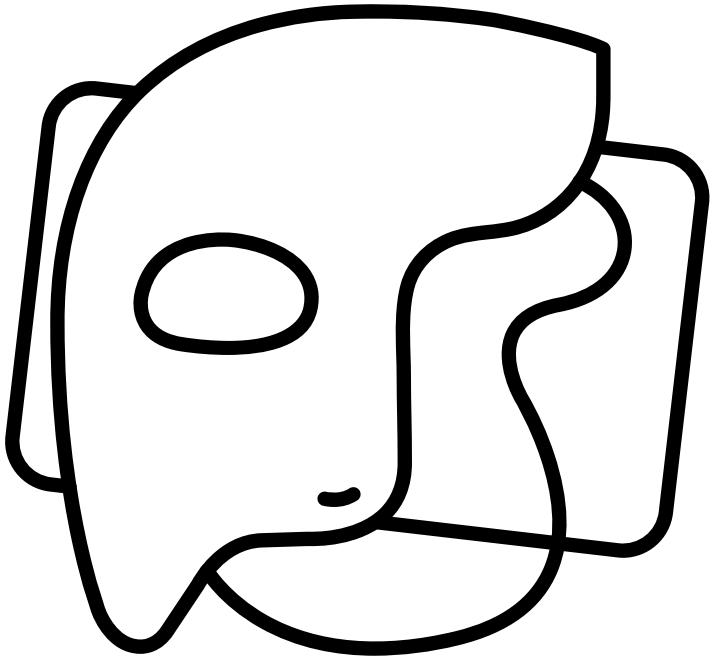


Securing Resource Constrained Processors with Name Confusion

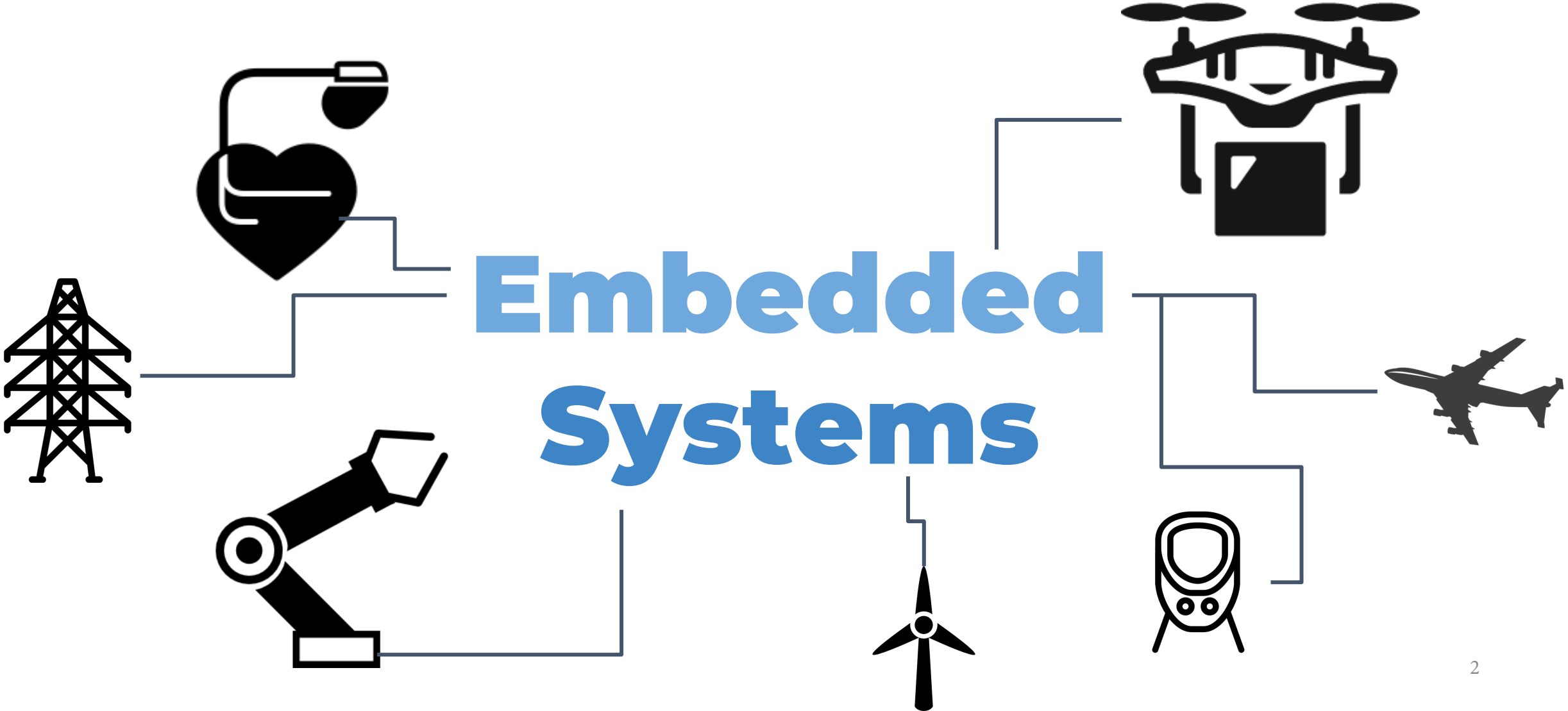
Mohamed Tarek Ibn Ziad, Miguel A. Arroyo, Evgeny Manzhosov, Vasileios P. Kemerlis, and Simha Sethumadhavan

CS@CU COMPUTER SCIENCE

BROWN

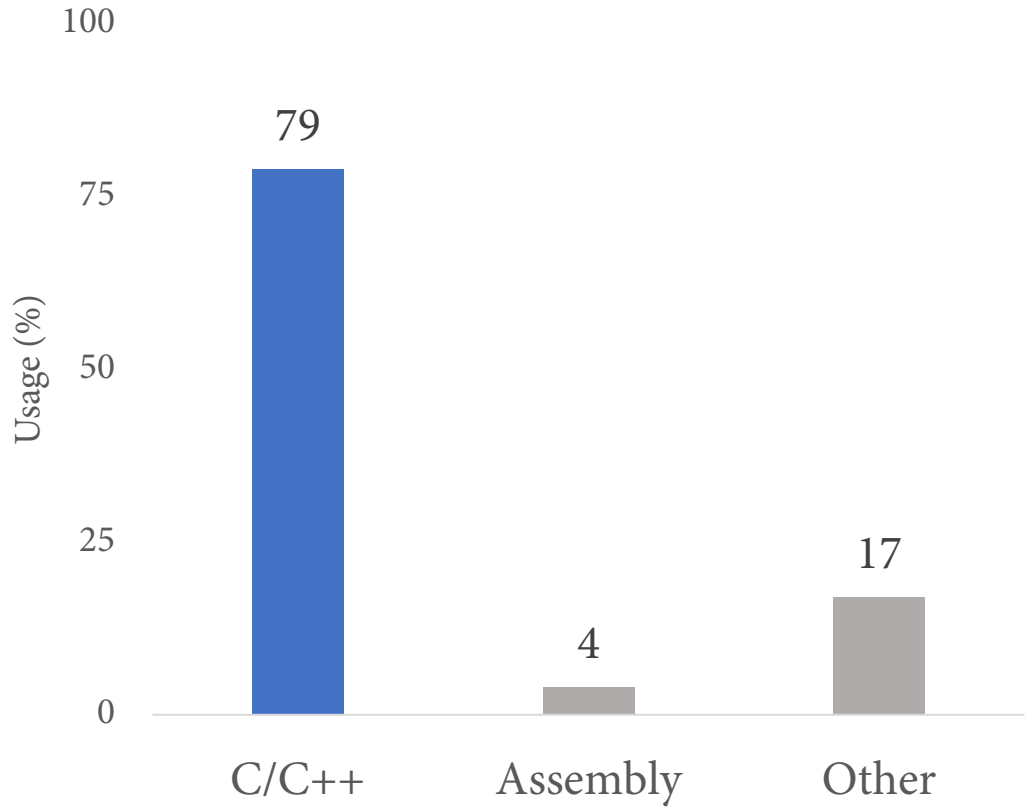
Columbia University
Brown University
09/21/2021

Embedded systems are everywhere!



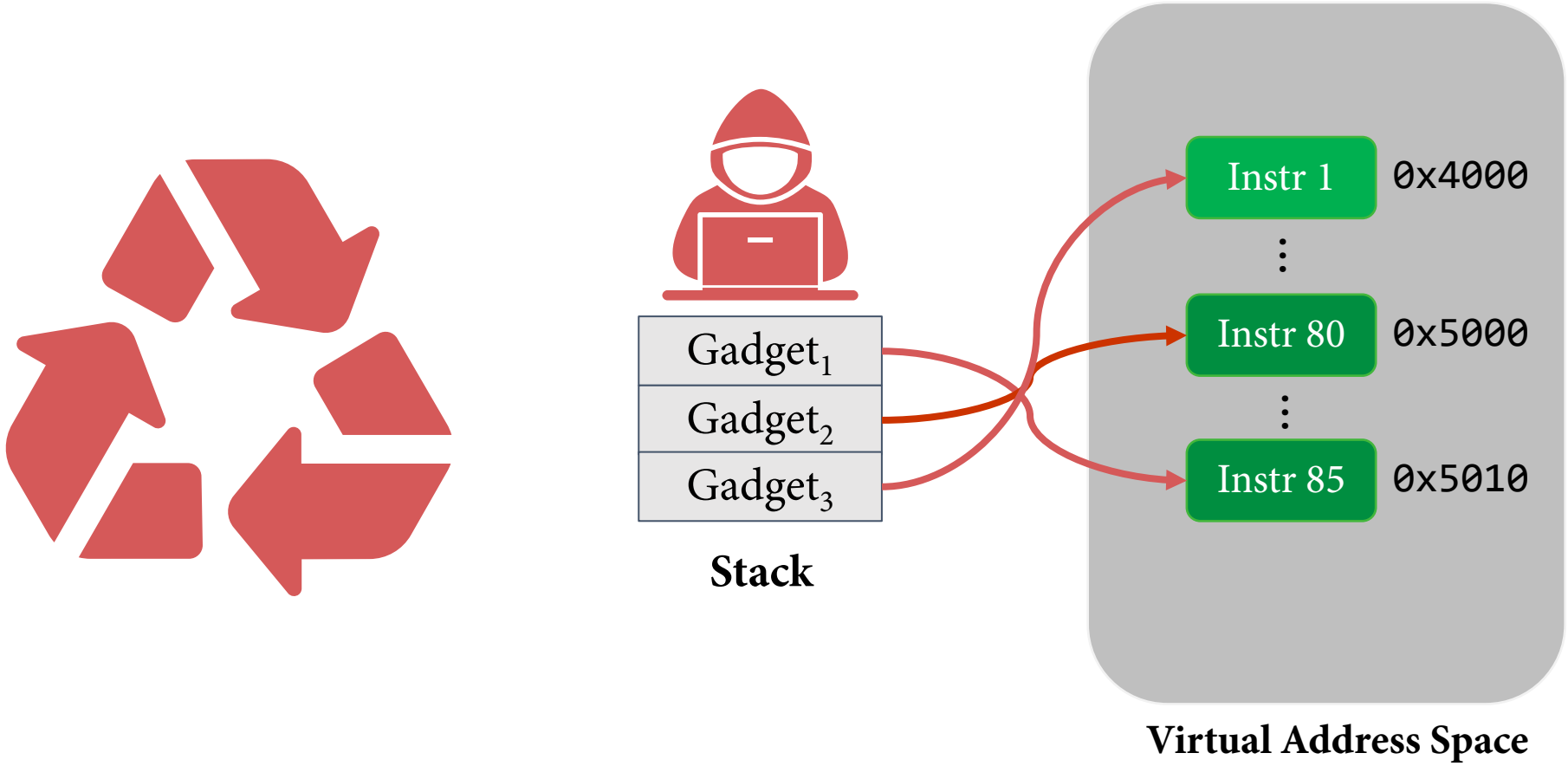
Why focus on software threats?

CPSs are predominantely written in memory unsafe languages.



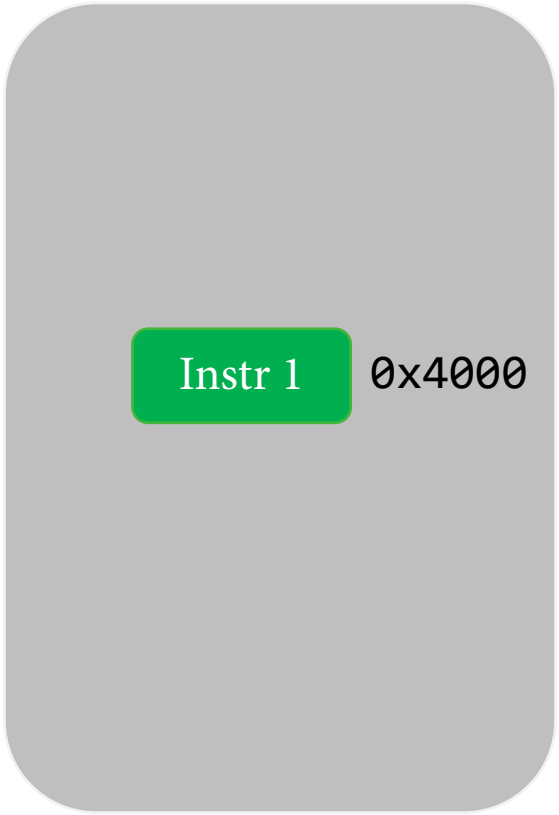
Source: *Embedded Insights Survey 2019*

Code Reuse Attacks



Traditional Architecture

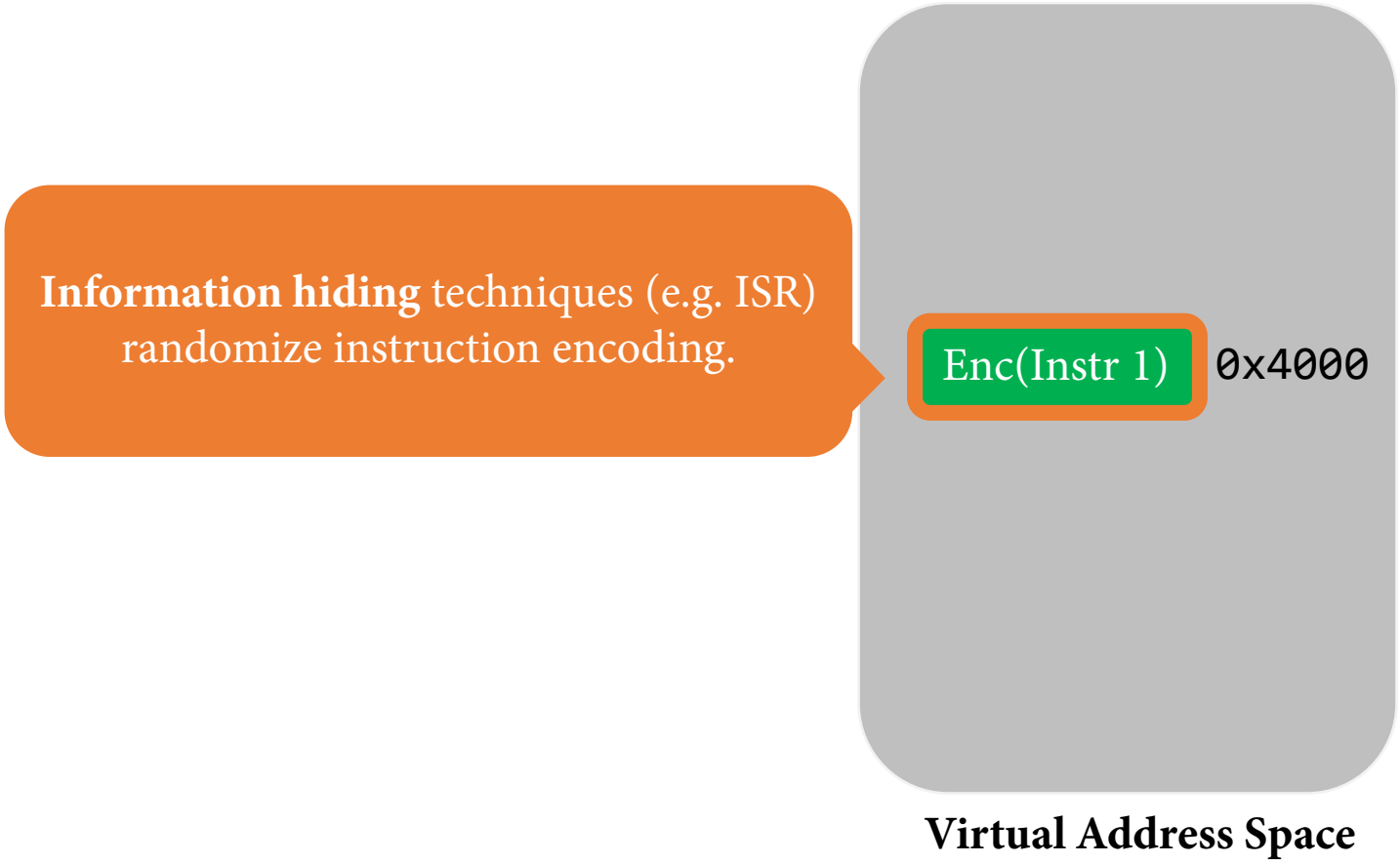
An instruction has a single name (or address).



Virtual Address Space

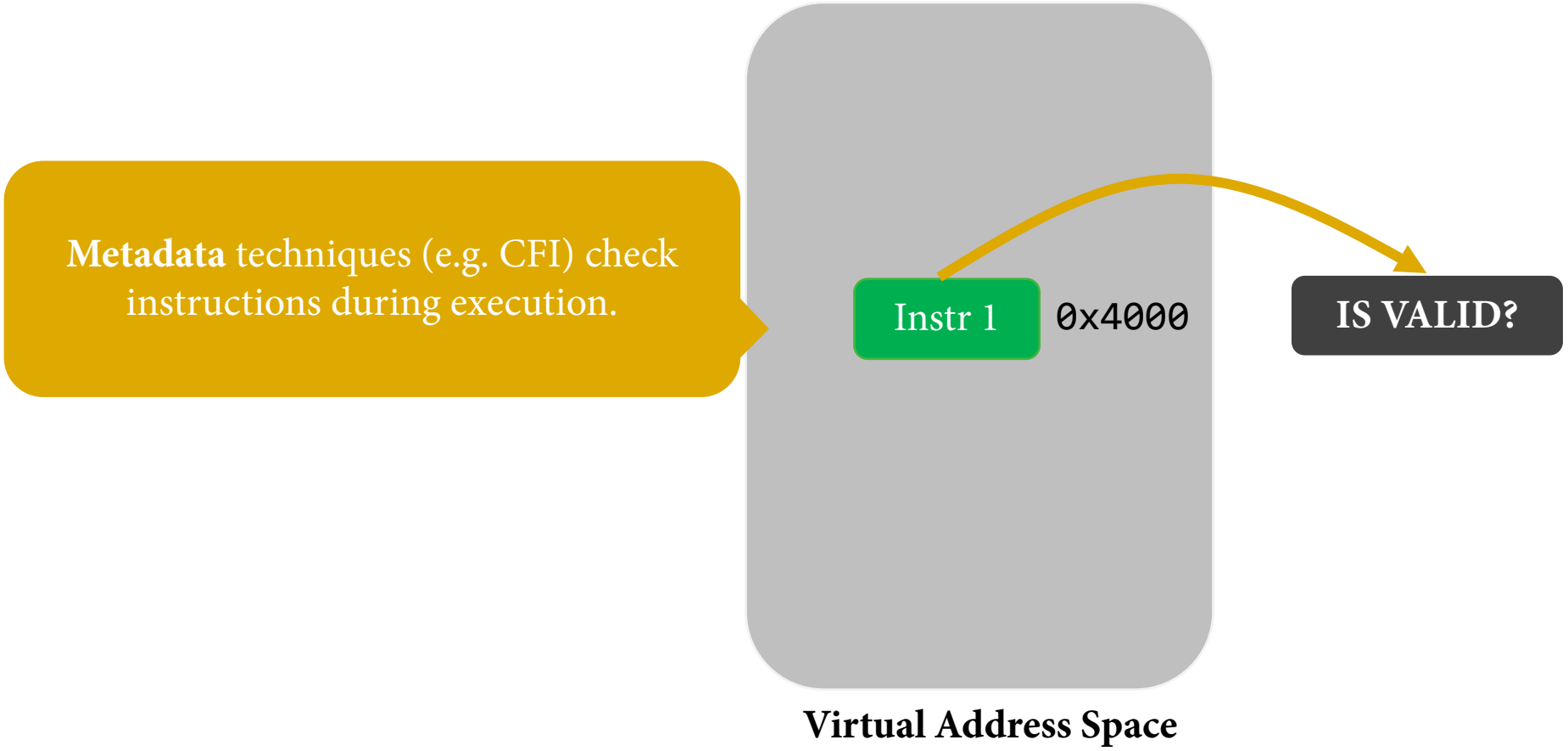
Traditional Architecture

An instruction has a single name (or address).



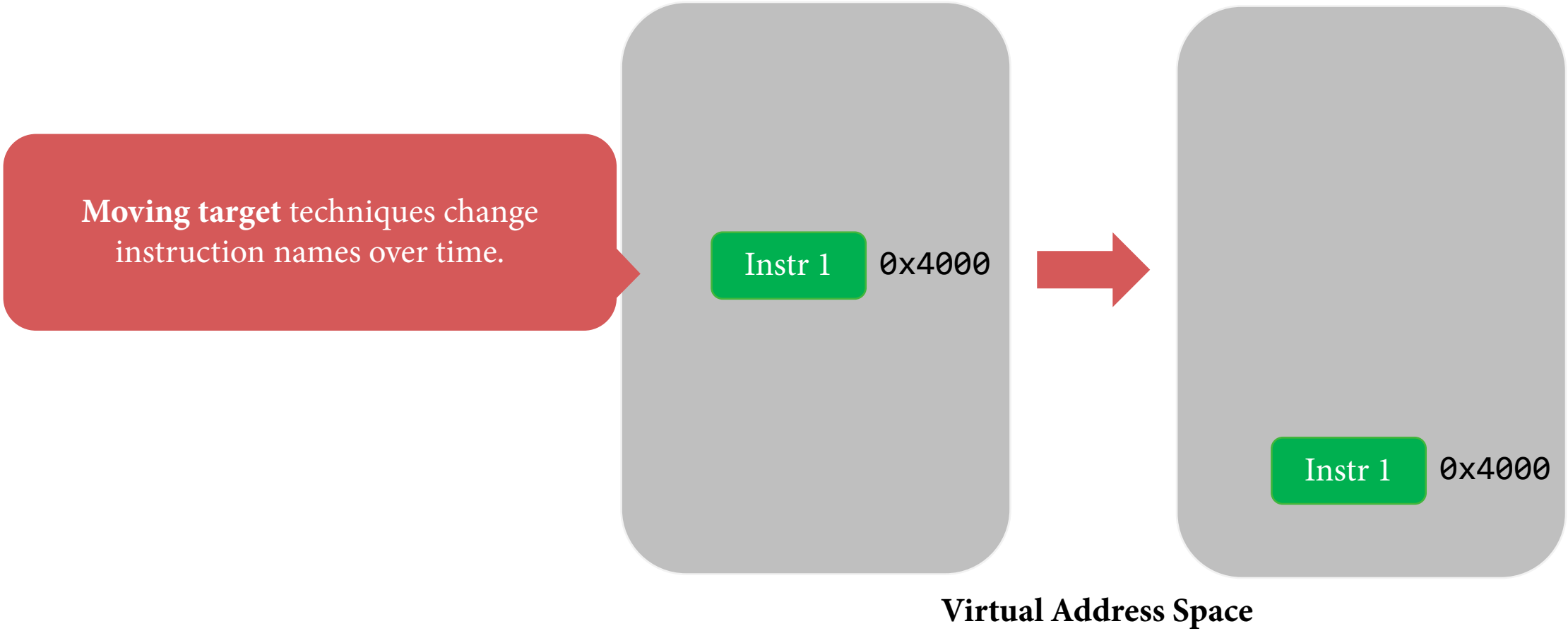
Traditional Architecture

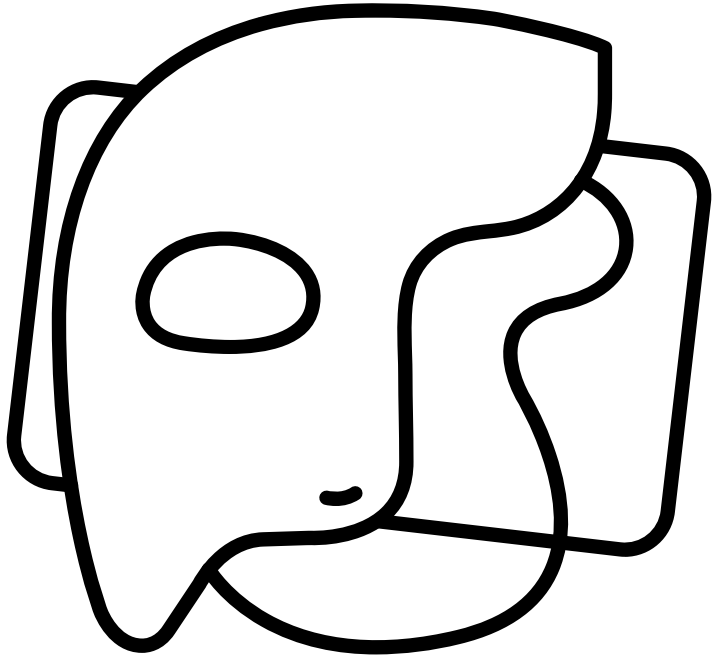
An instruction has a single name (or address).



Traditional Architecture

An instruction has a single name (or address).



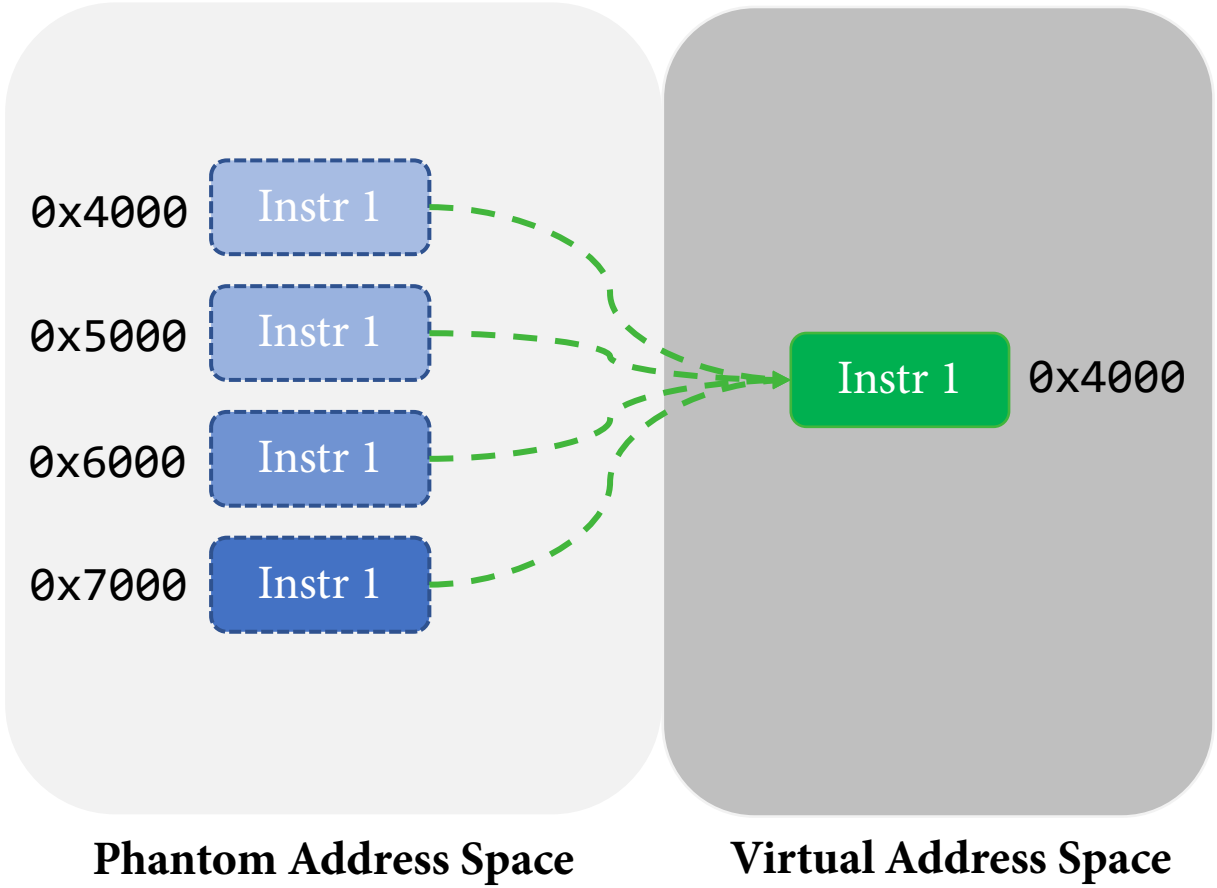


Phantom Name System

A Name Confusion Architecture

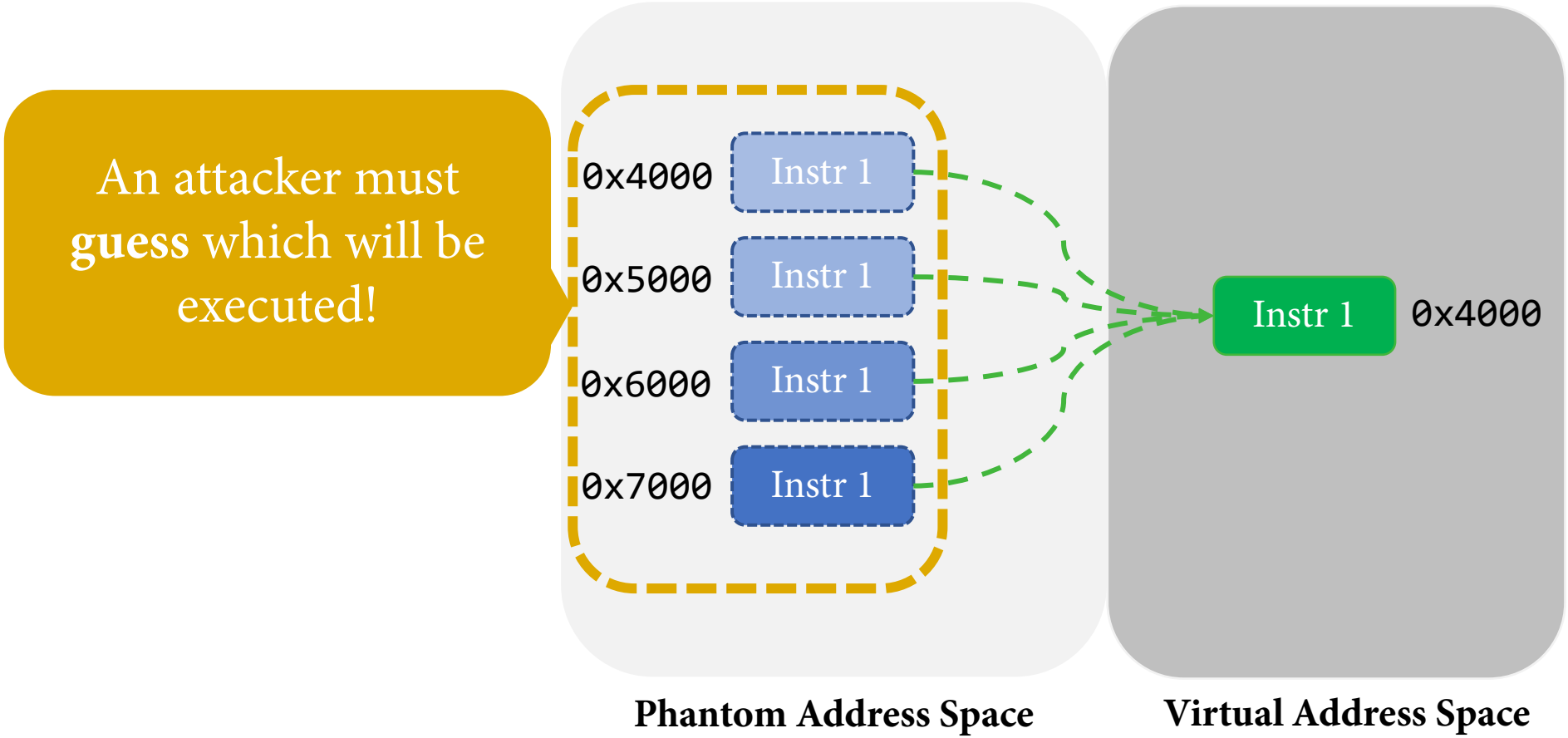
Name Confusion Architecture

Multiple phantom addresses alias to an instruction.



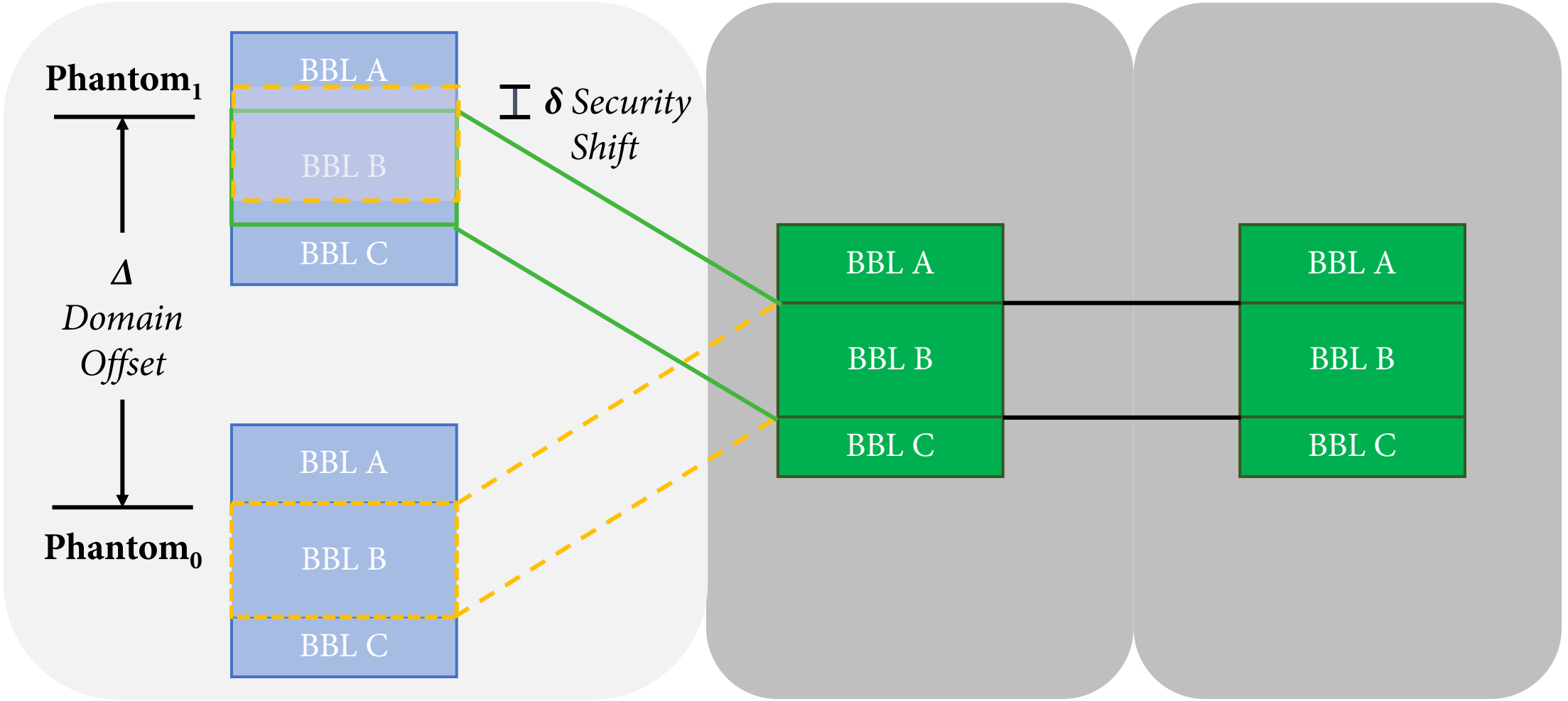
Name Confusion Architecture

Multiple phantom addresses alias to an instruction.



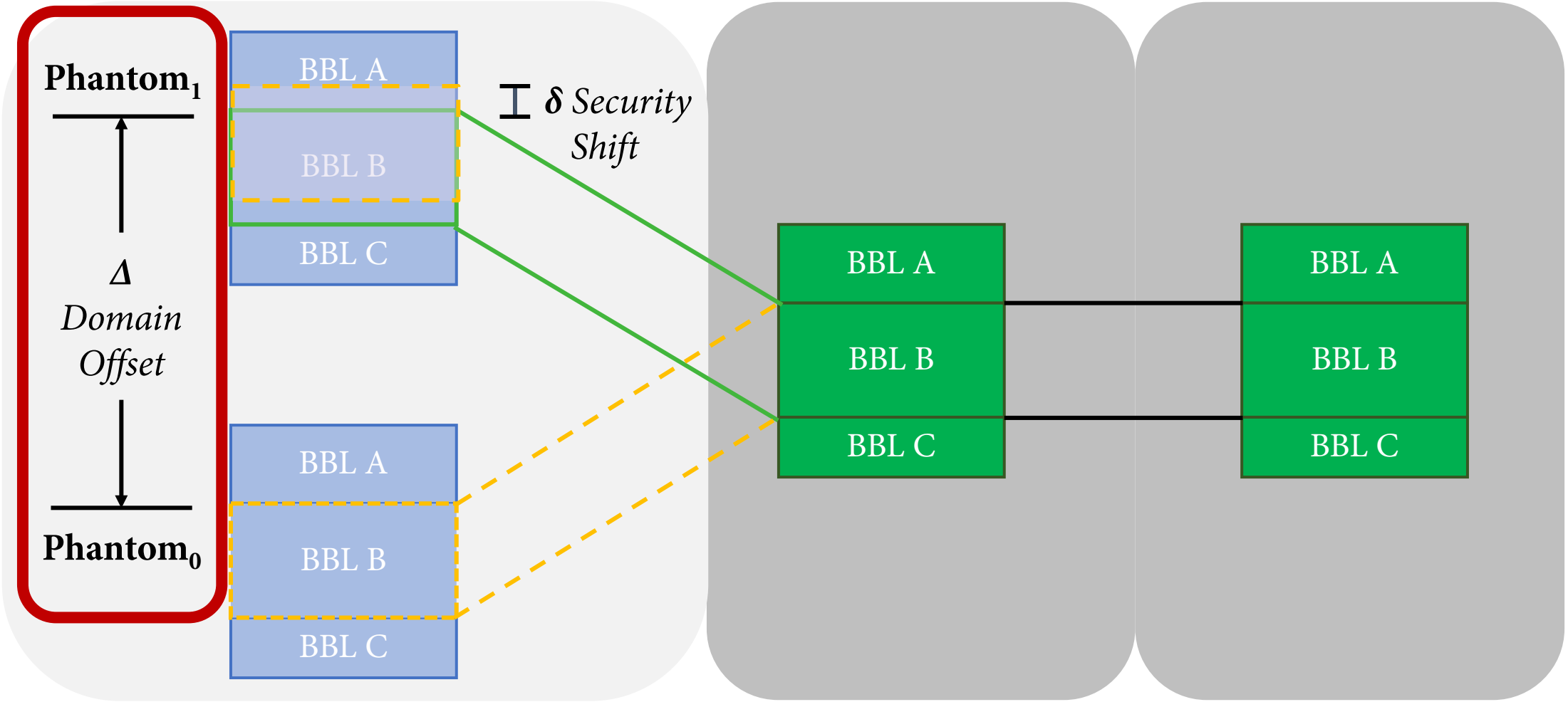
How are phantoms constructed?

Phantoms are logically displaced relative to the original program.



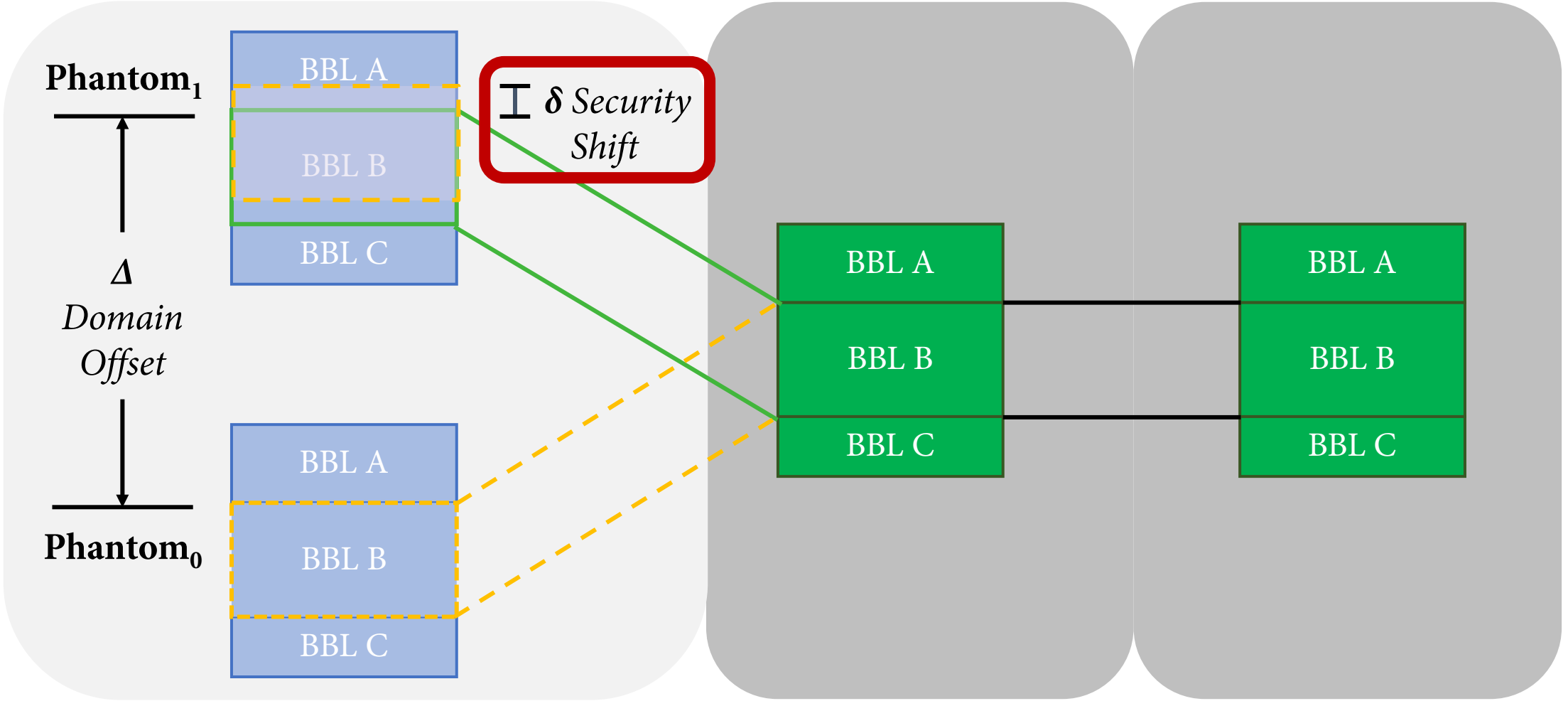
How are phantoms constructed?

Phantoms are logically displaced relative to the original program.



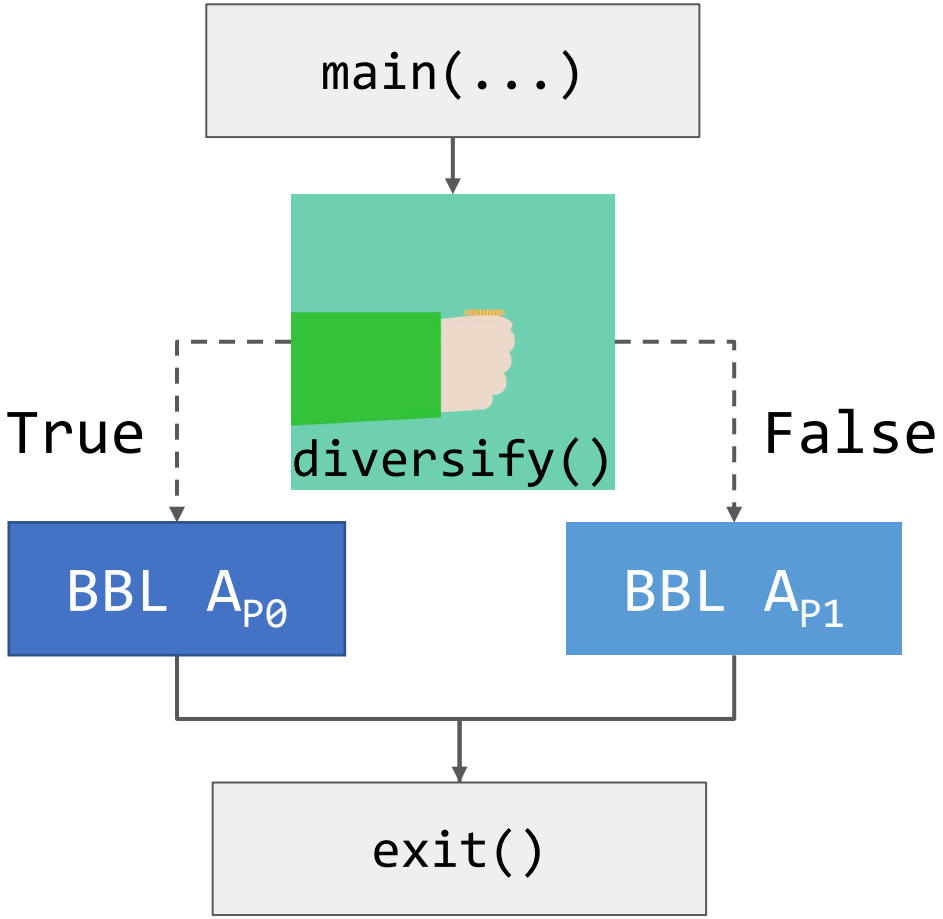
How are phantoms constructed?

Phantoms are logically displaced relative to the original program.



How does PNS diversify execution?

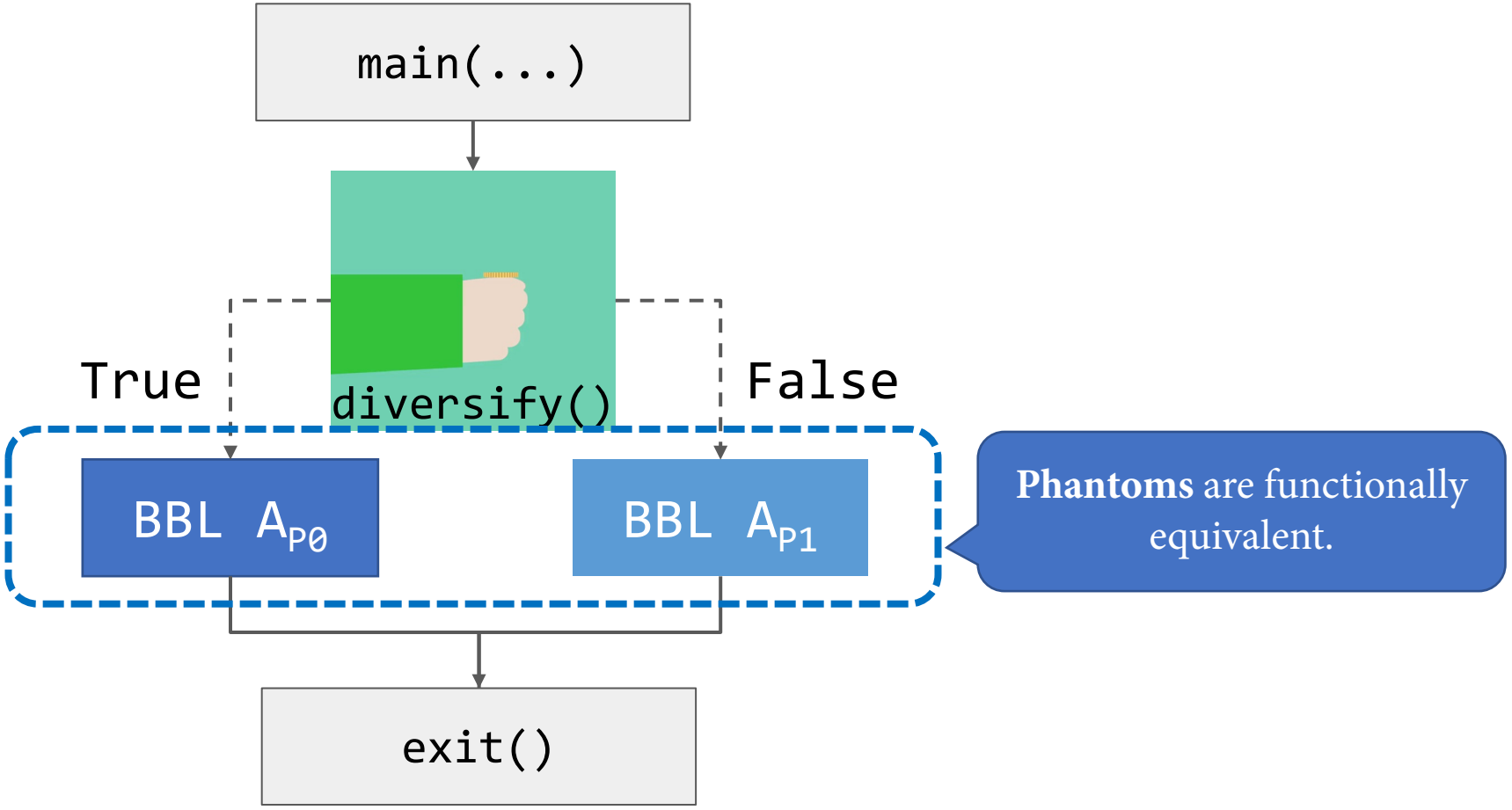
It diversifies the *path* of execution at every basic block.



Program Control Flow Graph

How does PNS diversify execution?

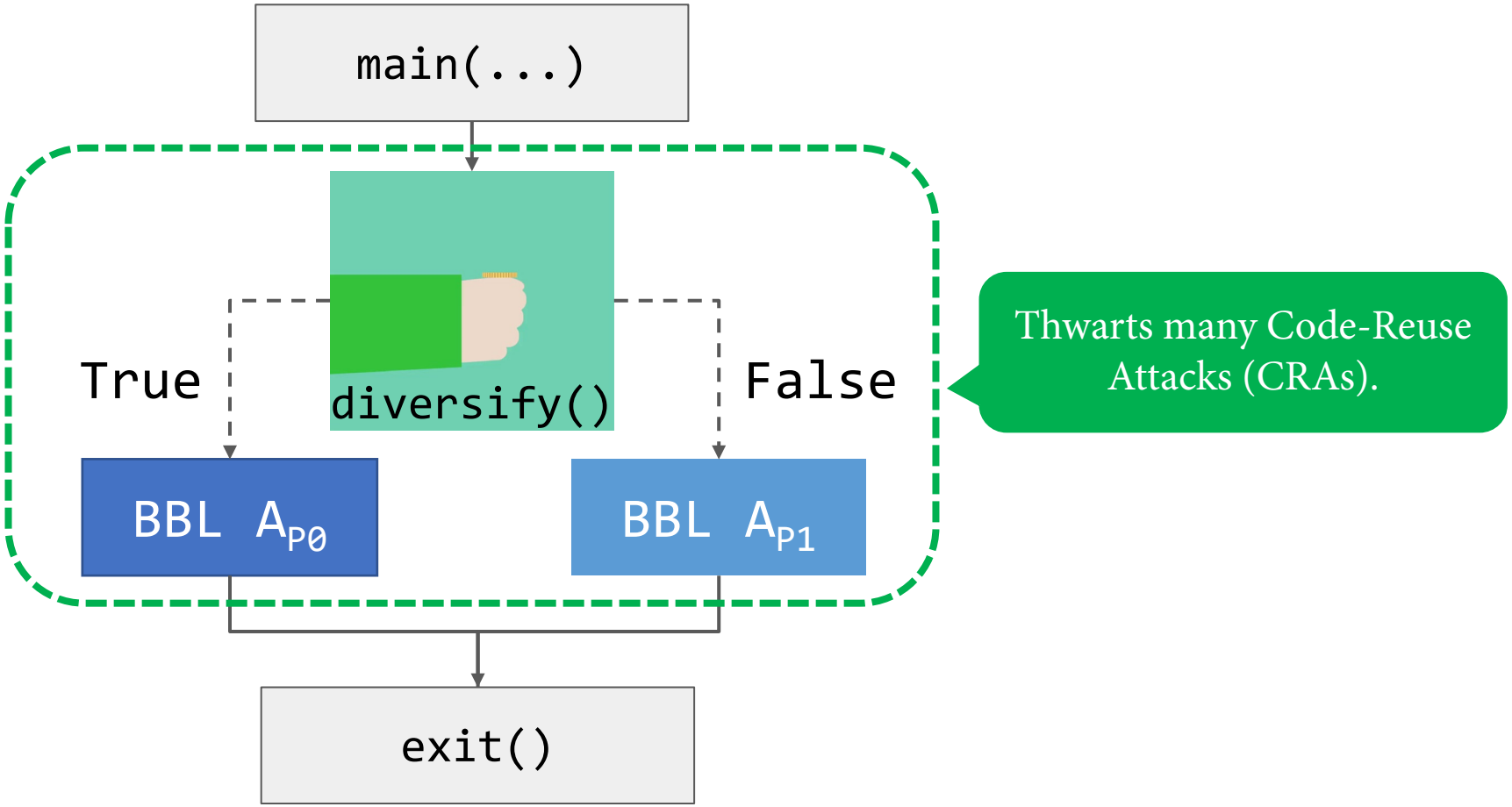
It diversifies the *path* of execution at every basic block.



Program Control Flow Graph

How does PNS diversify execution?

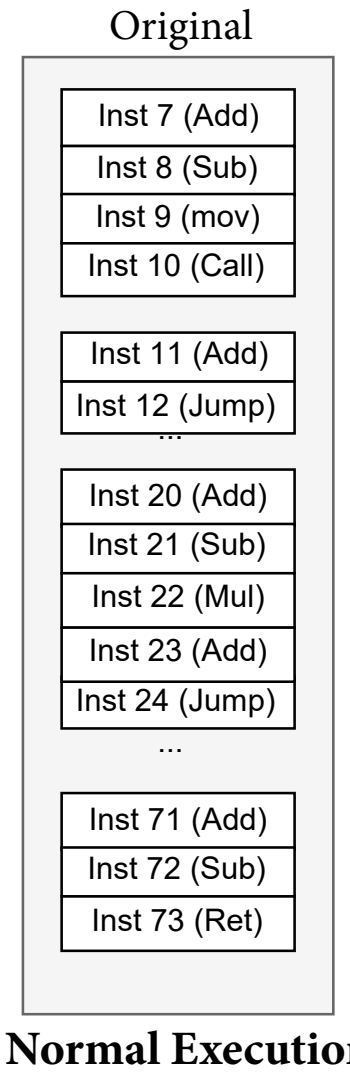
It diversifies the *path* of execution at every basic block.



Program Control Flow Graph

How does PNS protect against CRAs?

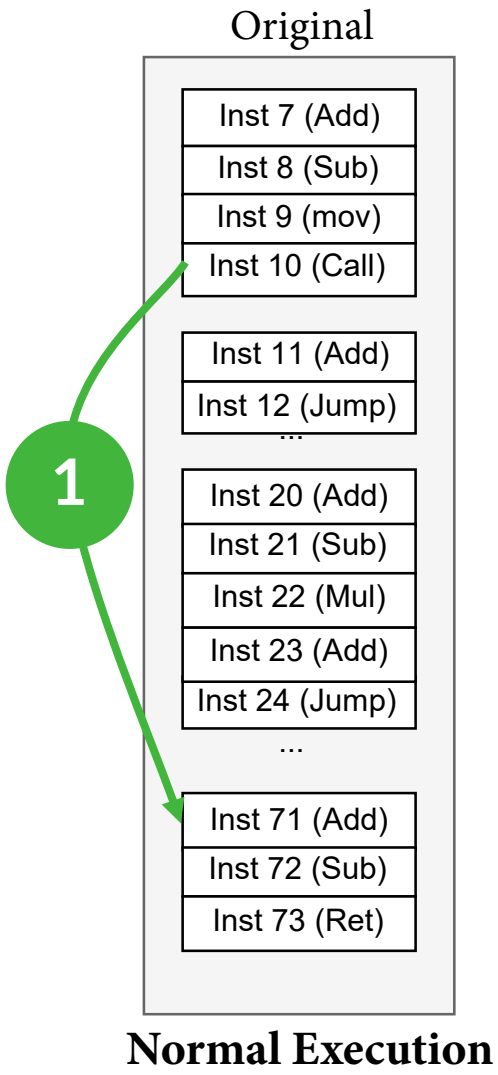
Phantoms force an adversary to guess the execution path.



Normal Execution

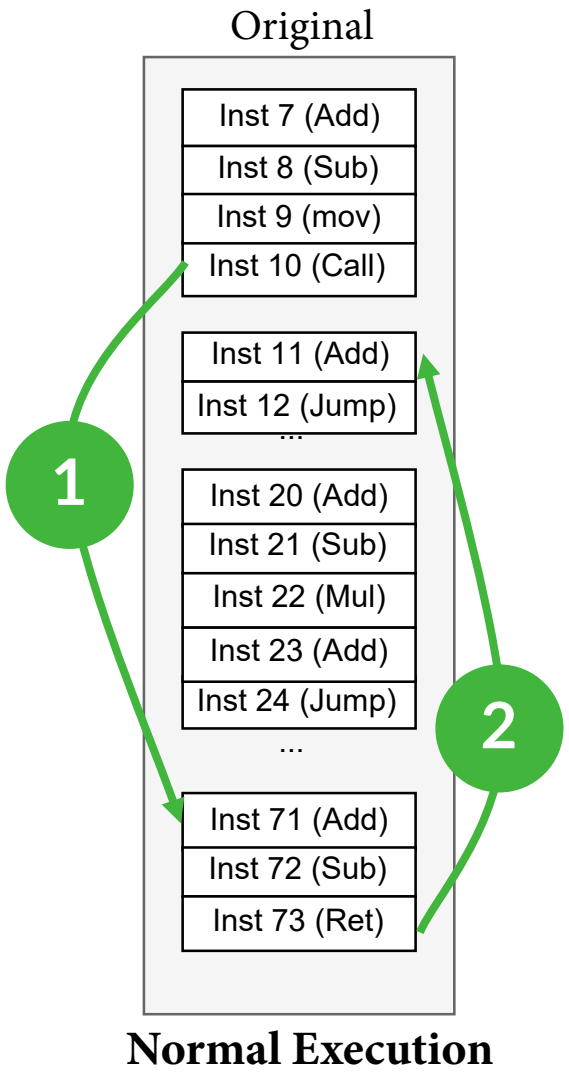
How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



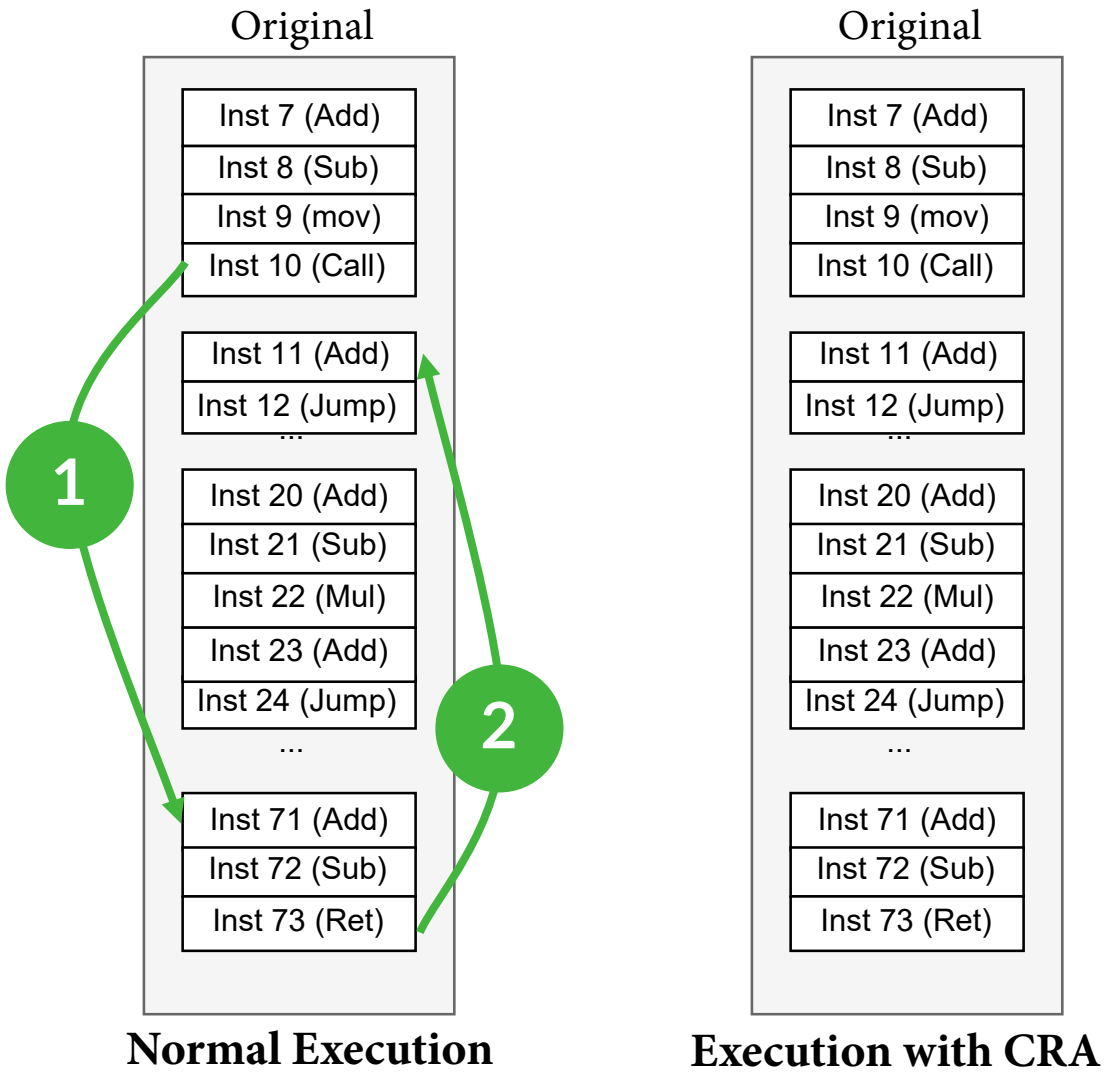
How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



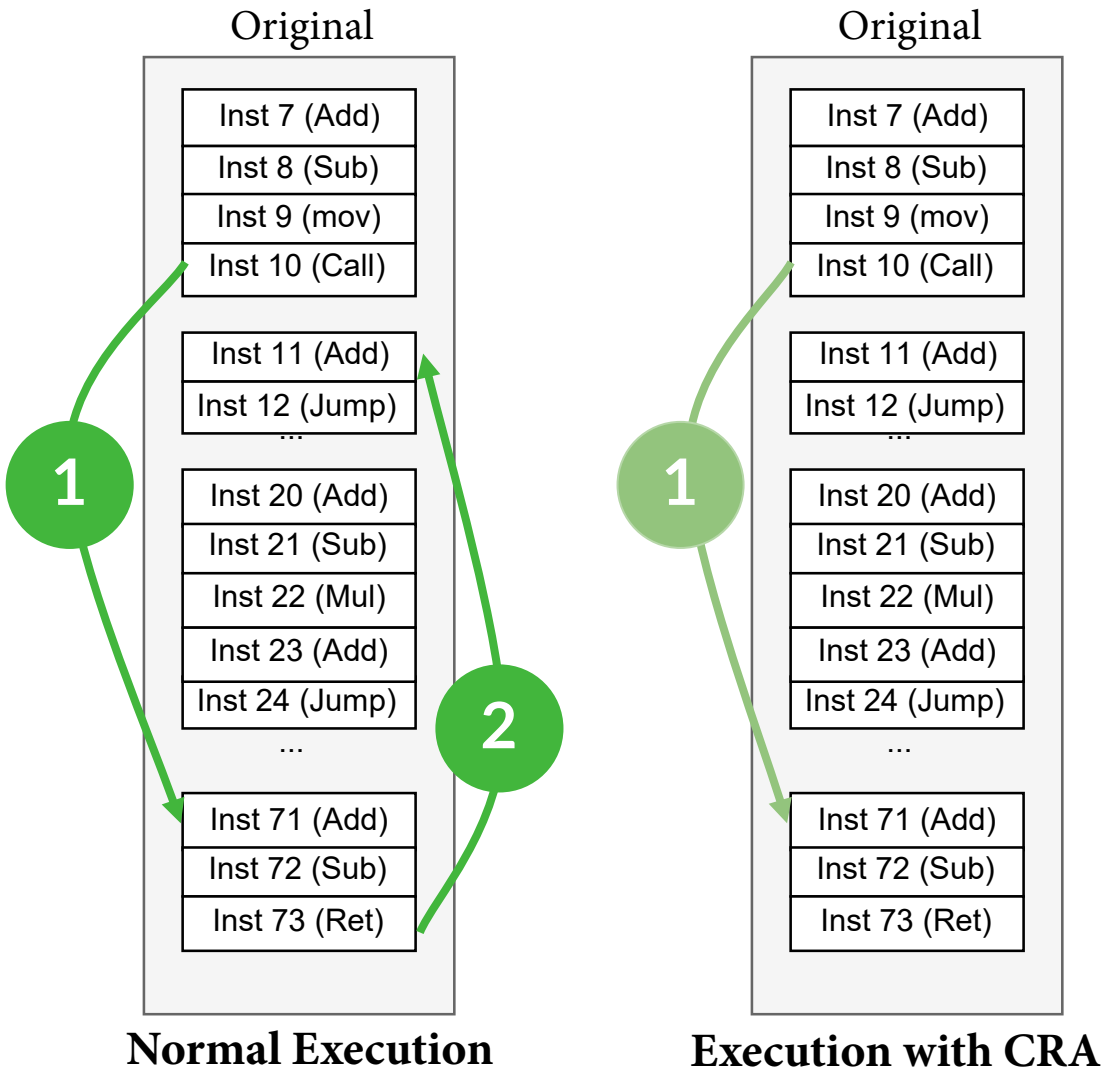
How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



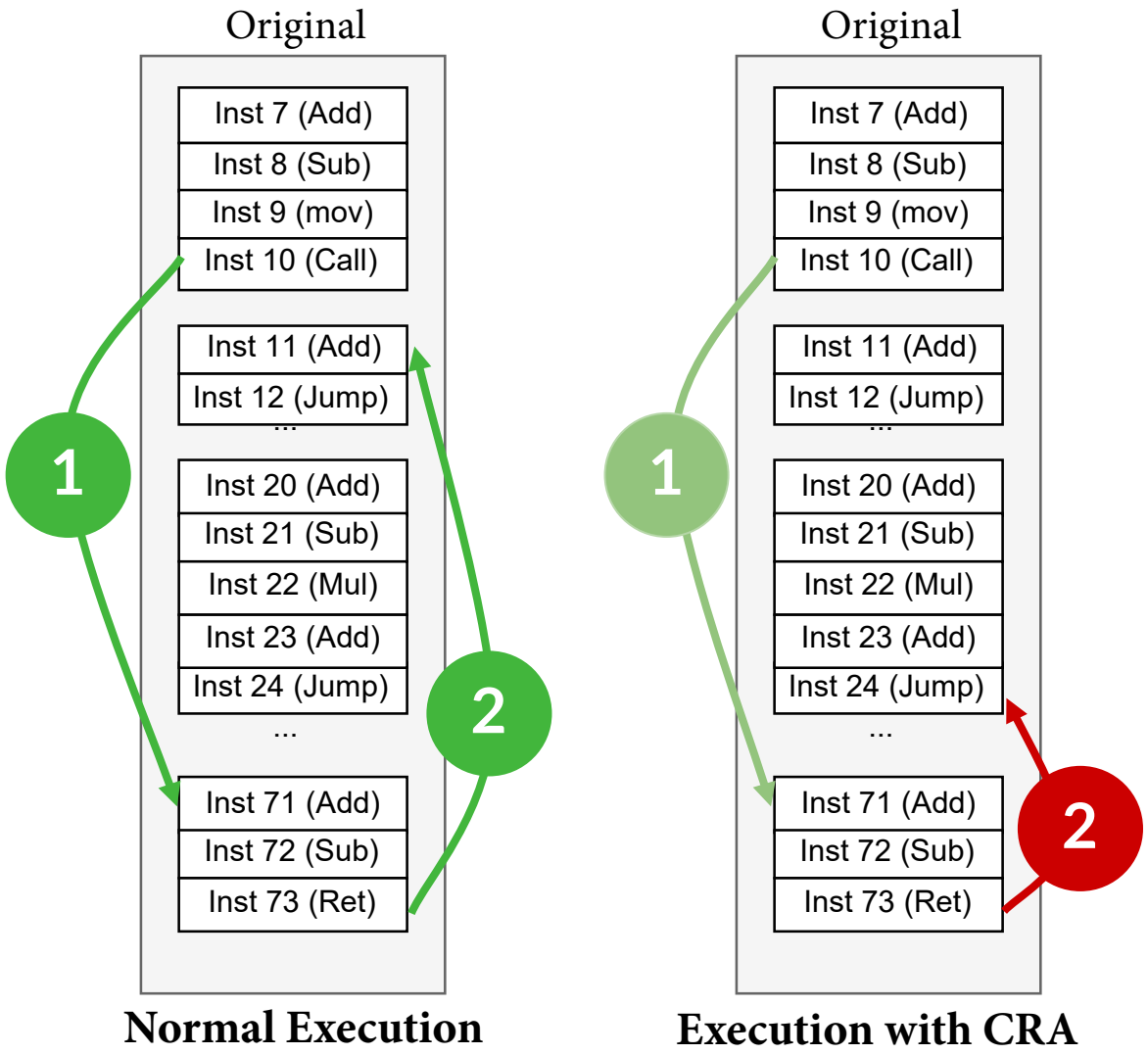
How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



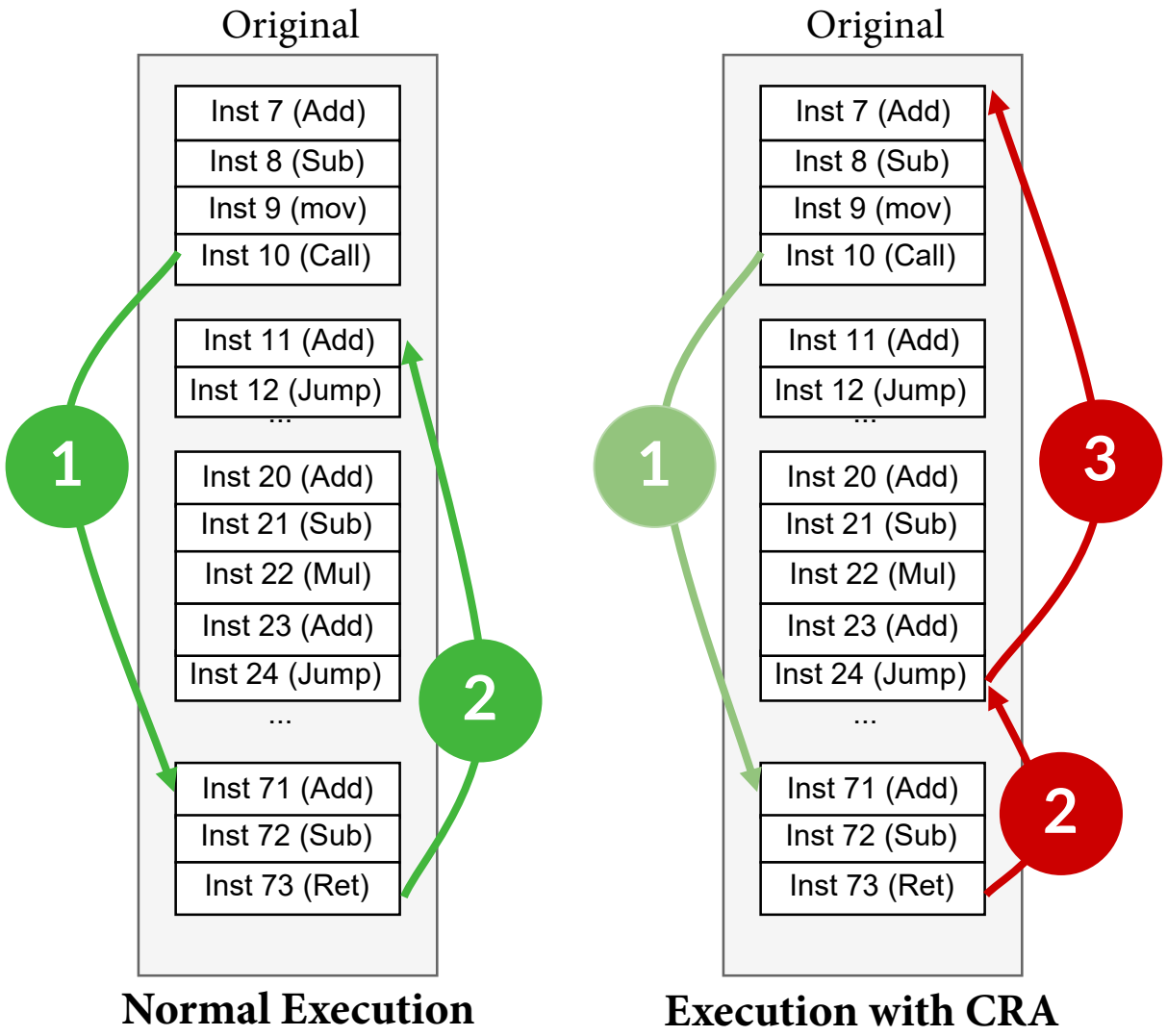
How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



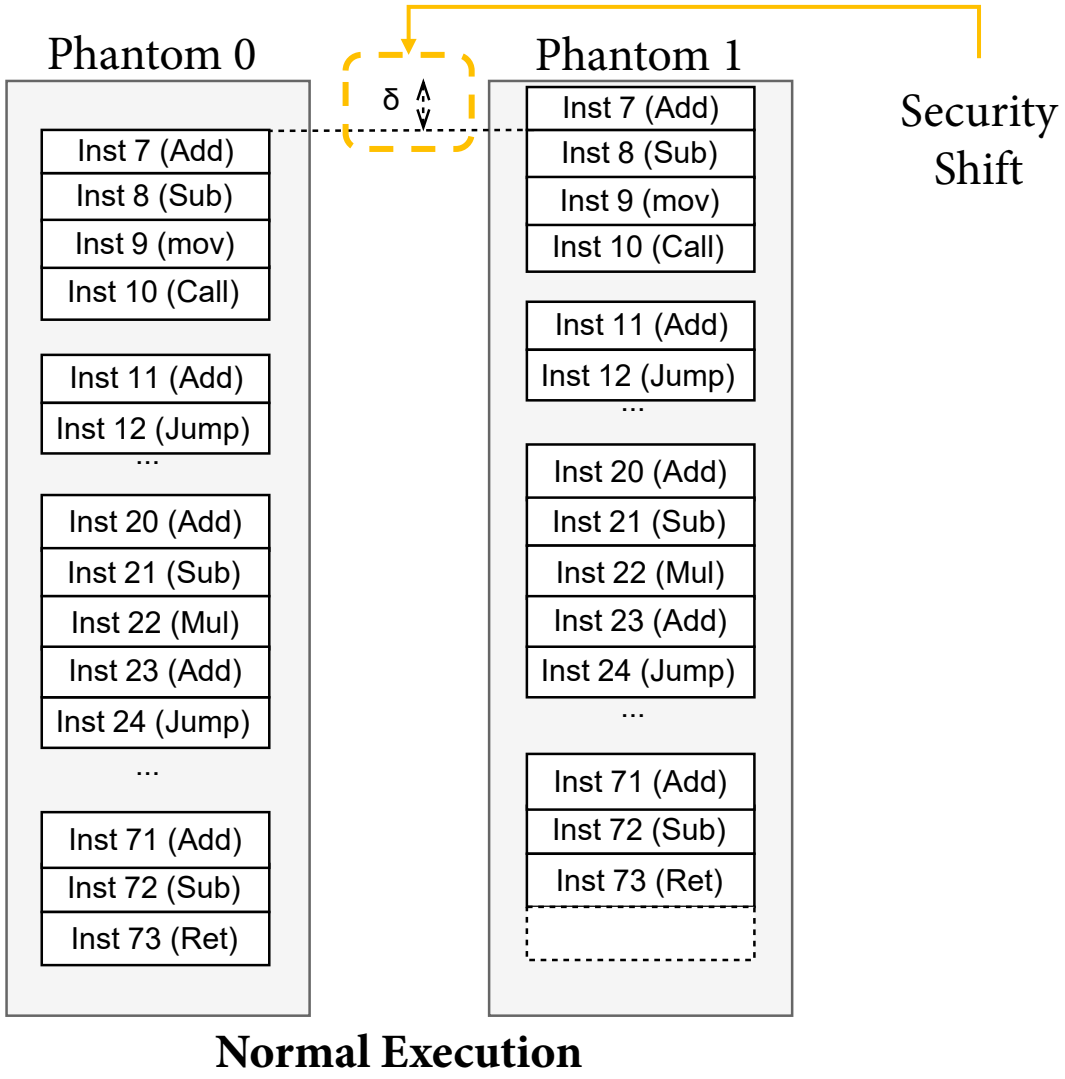
How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



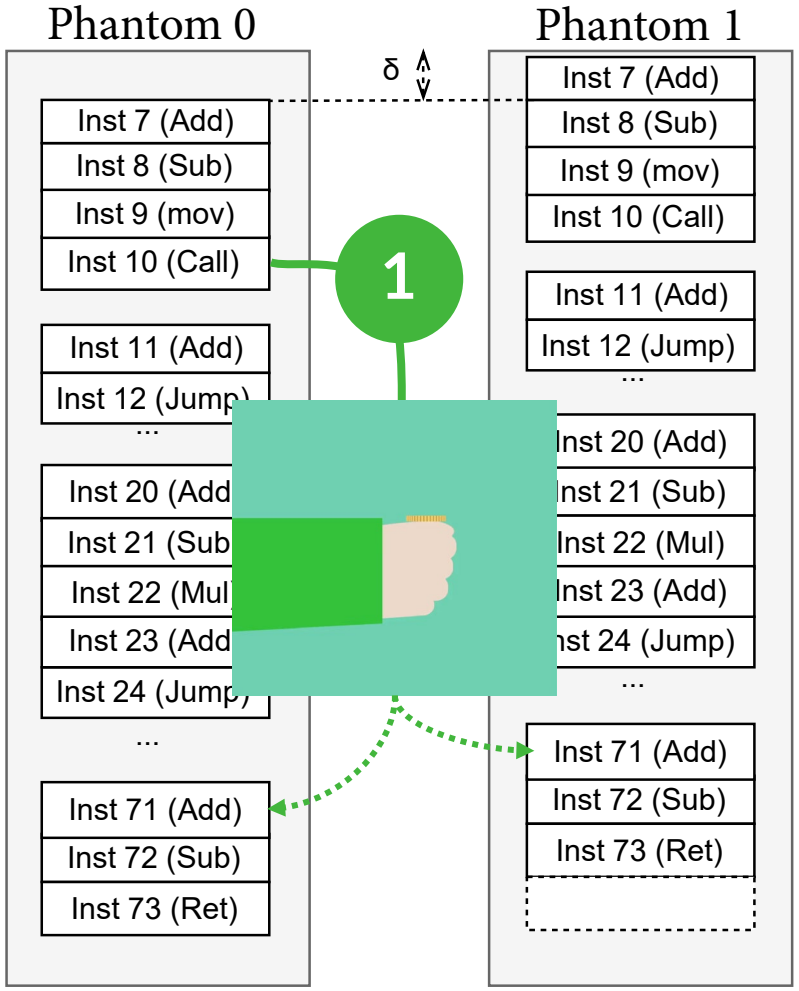
How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



How does PNS protect against CRAs?

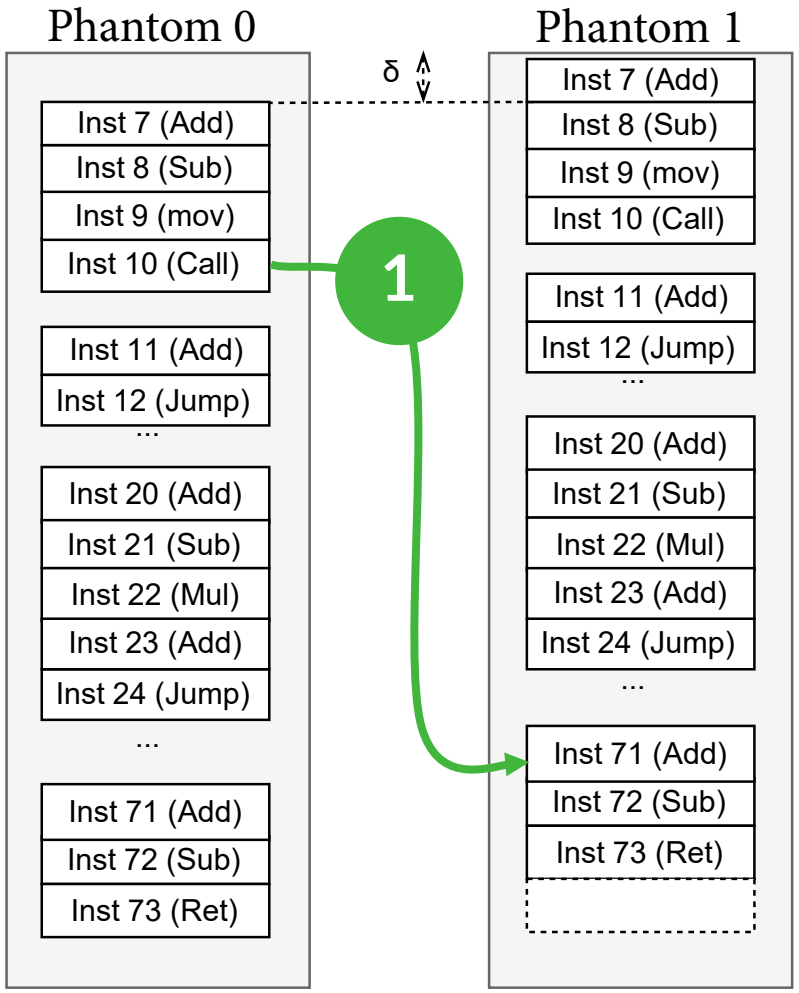
Phantoms force an adversary to guess the execution path.



Normal Execution

How does PNS protect against CRAs?

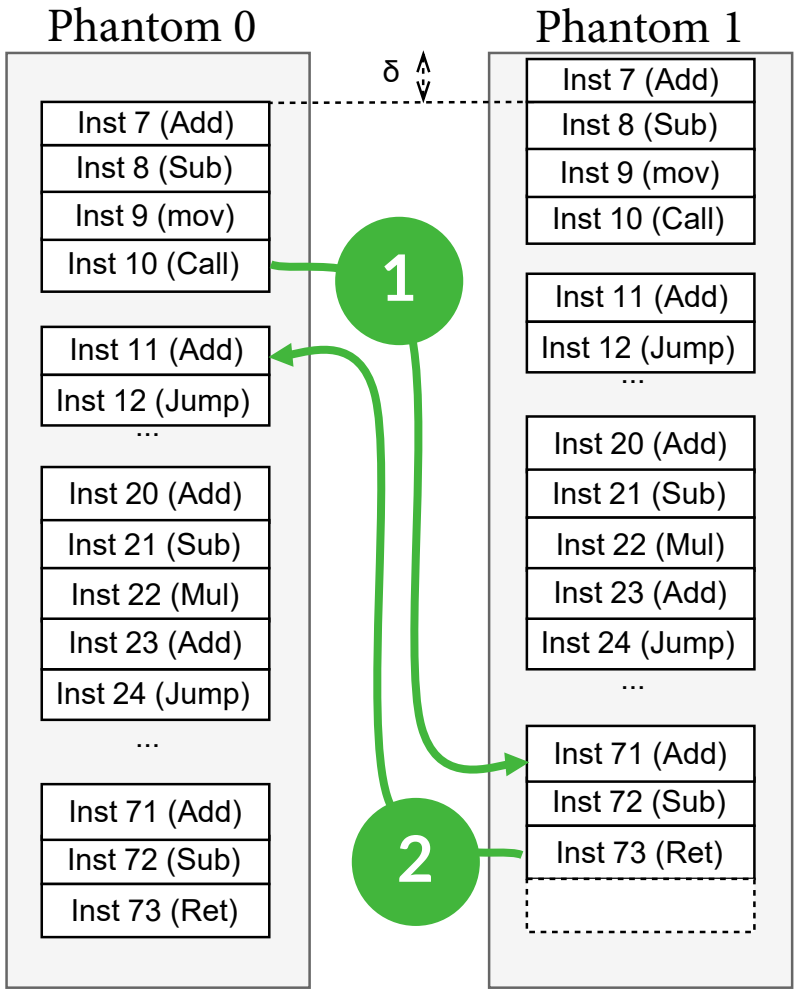
Phantoms force an adversary to guess the execution path.



Normal Execution

How does PNS protect against CRAs?

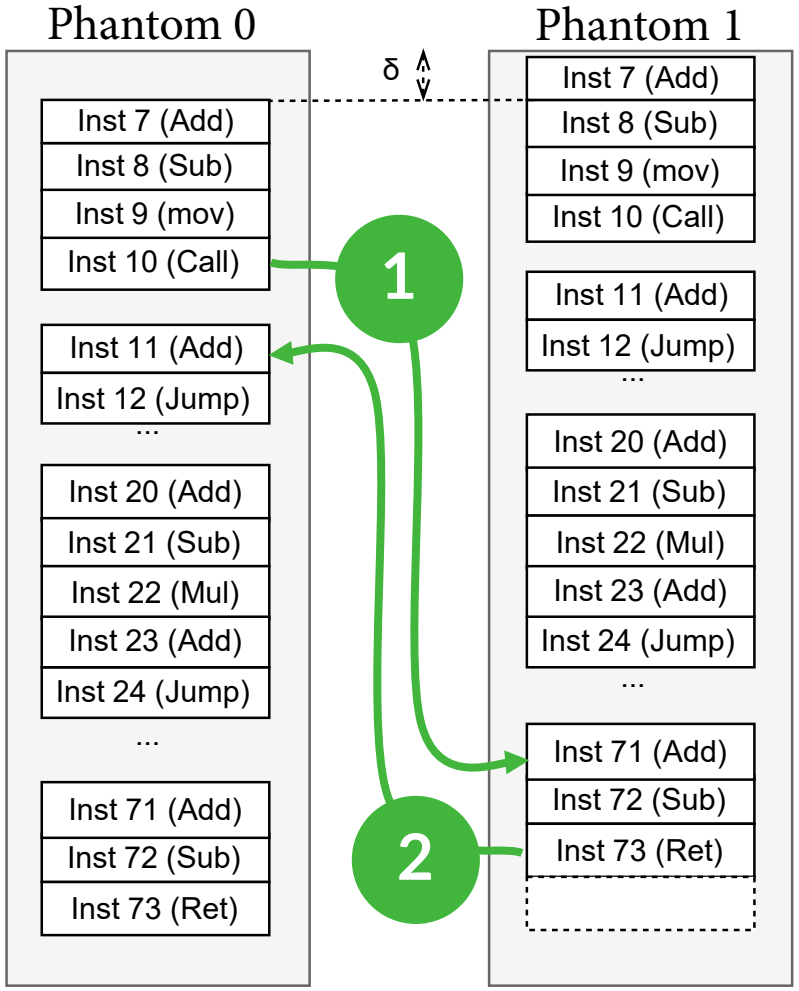
Phantoms force an adversary to guess the execution path.



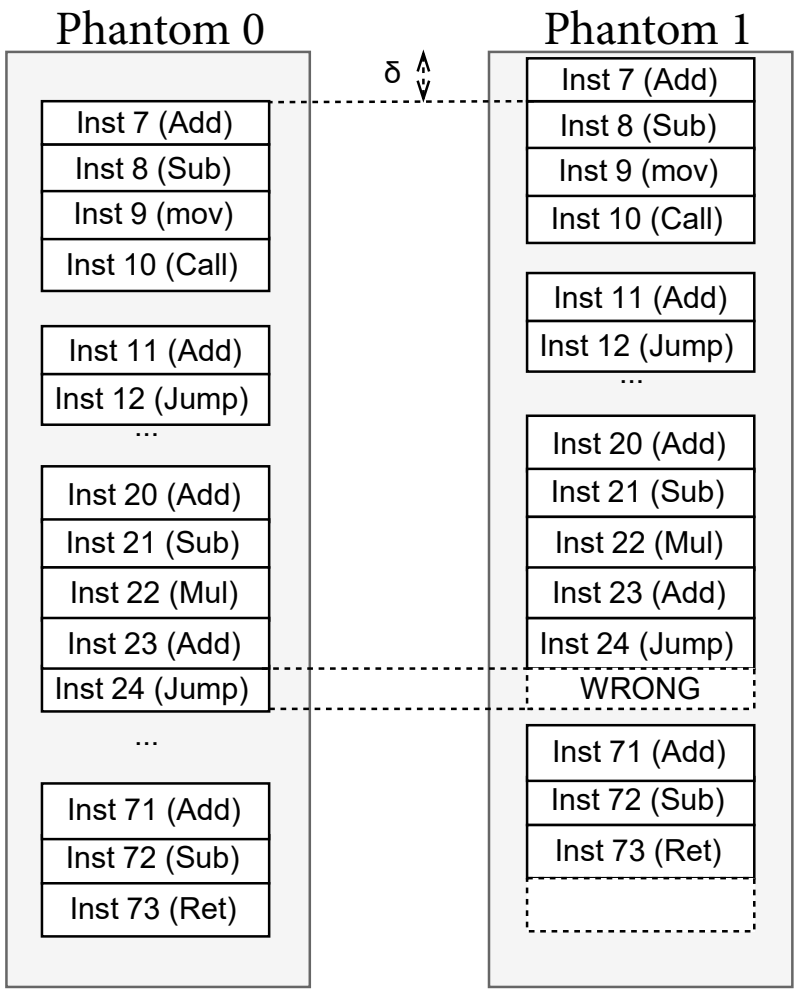
Normal Execution

How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



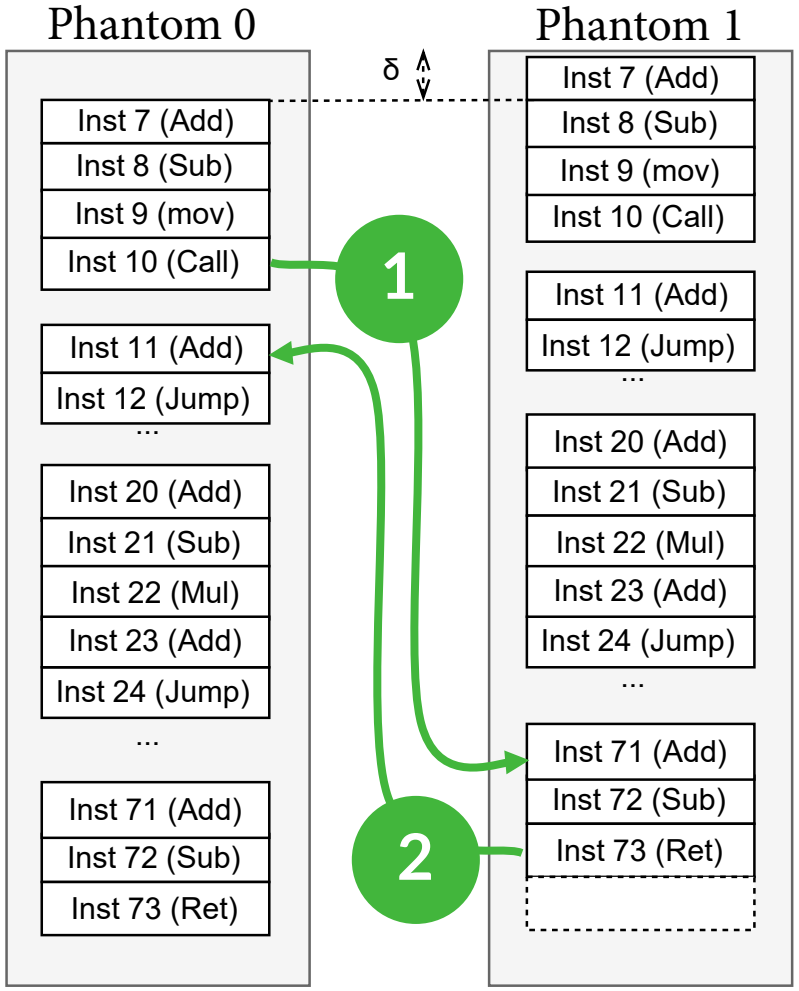
Normal Execution



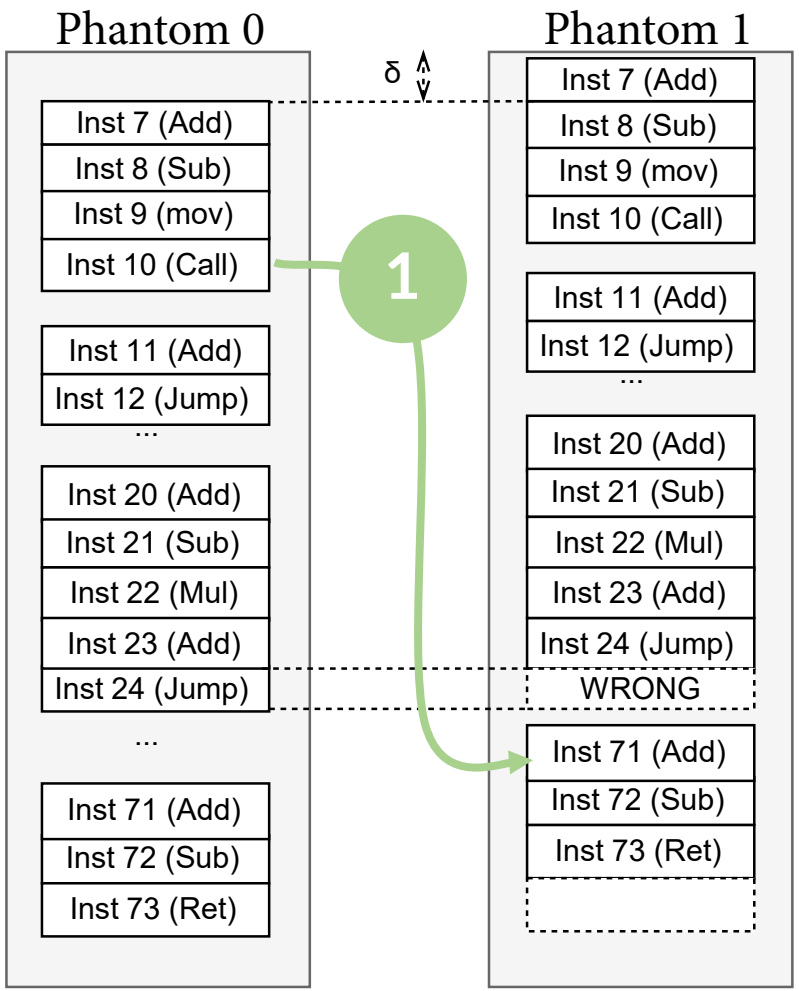
Execution with CRA

How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



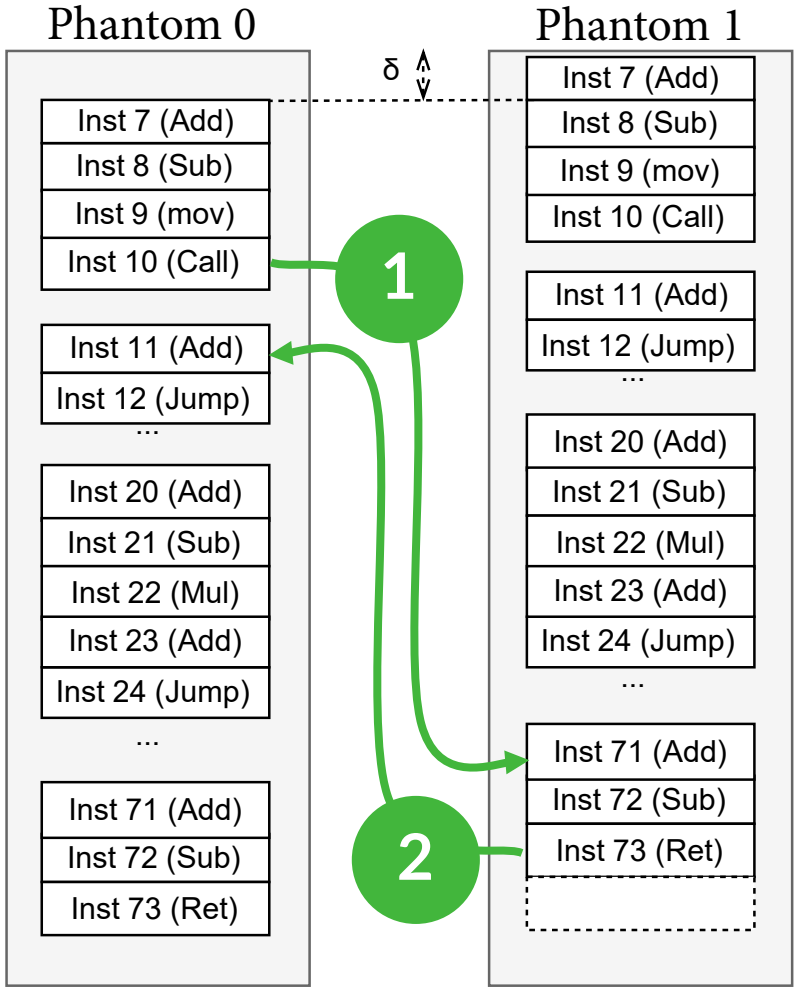
Normal Execution



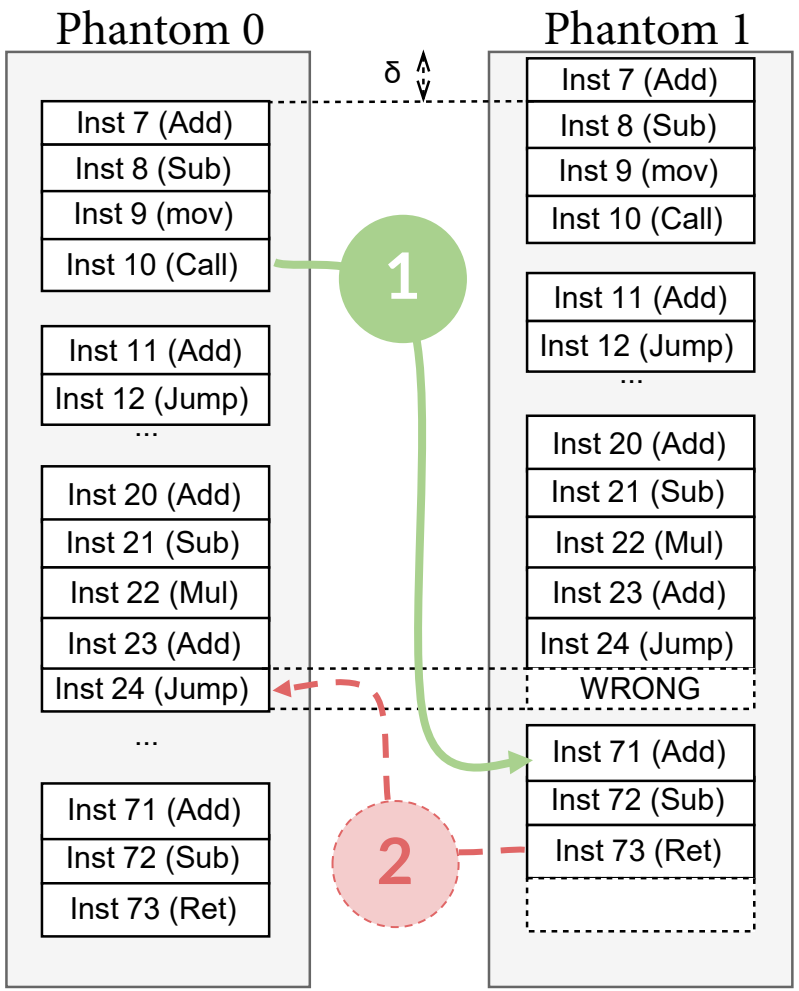
Execution with CRA

How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



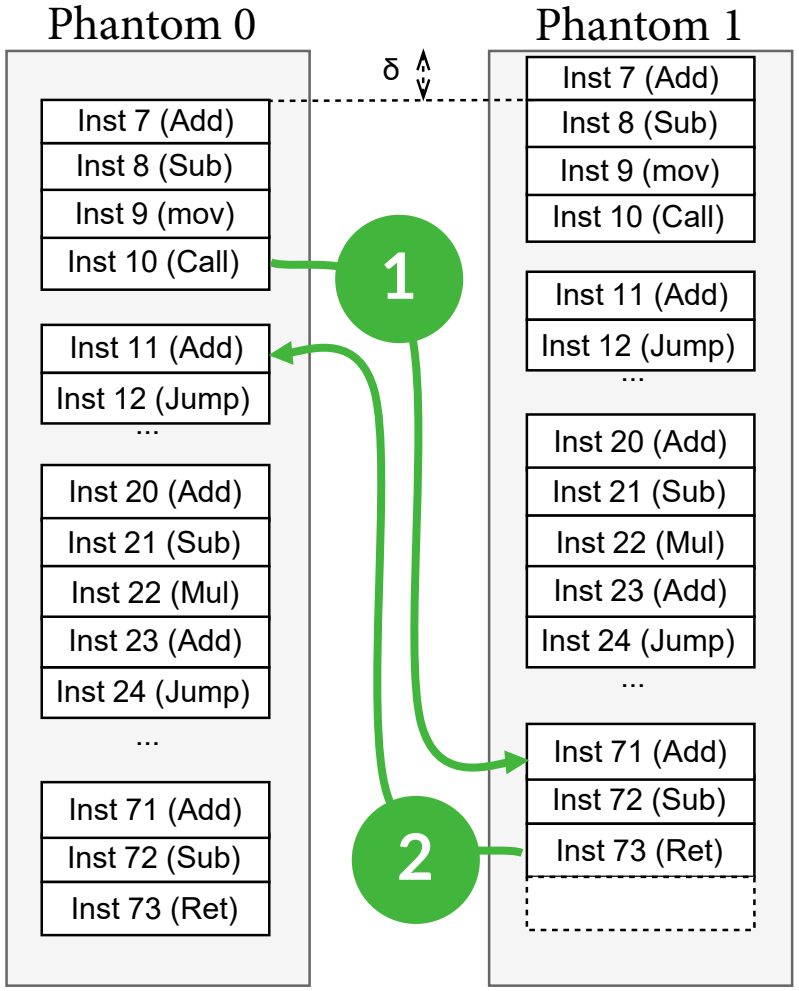
Normal Execution



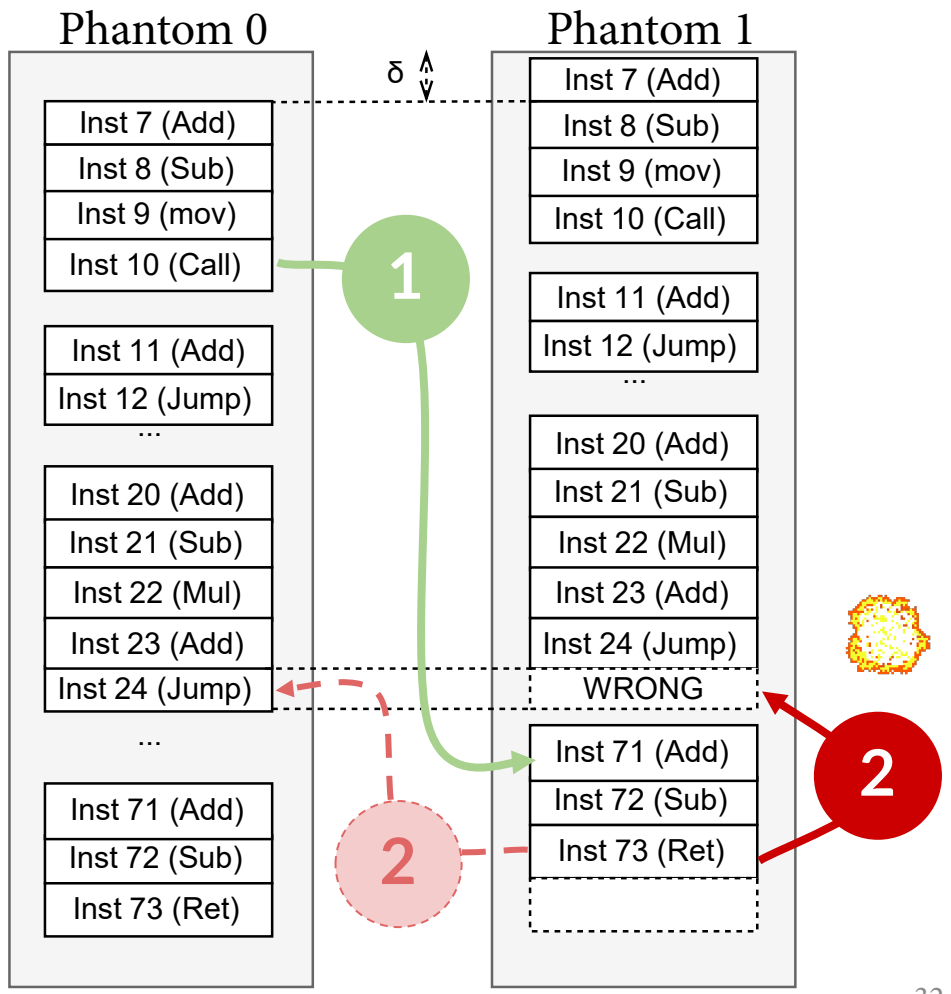
Execution with CRA

How does PNS protect against CRAs?

Phantoms force an adversary to guess the execution path.



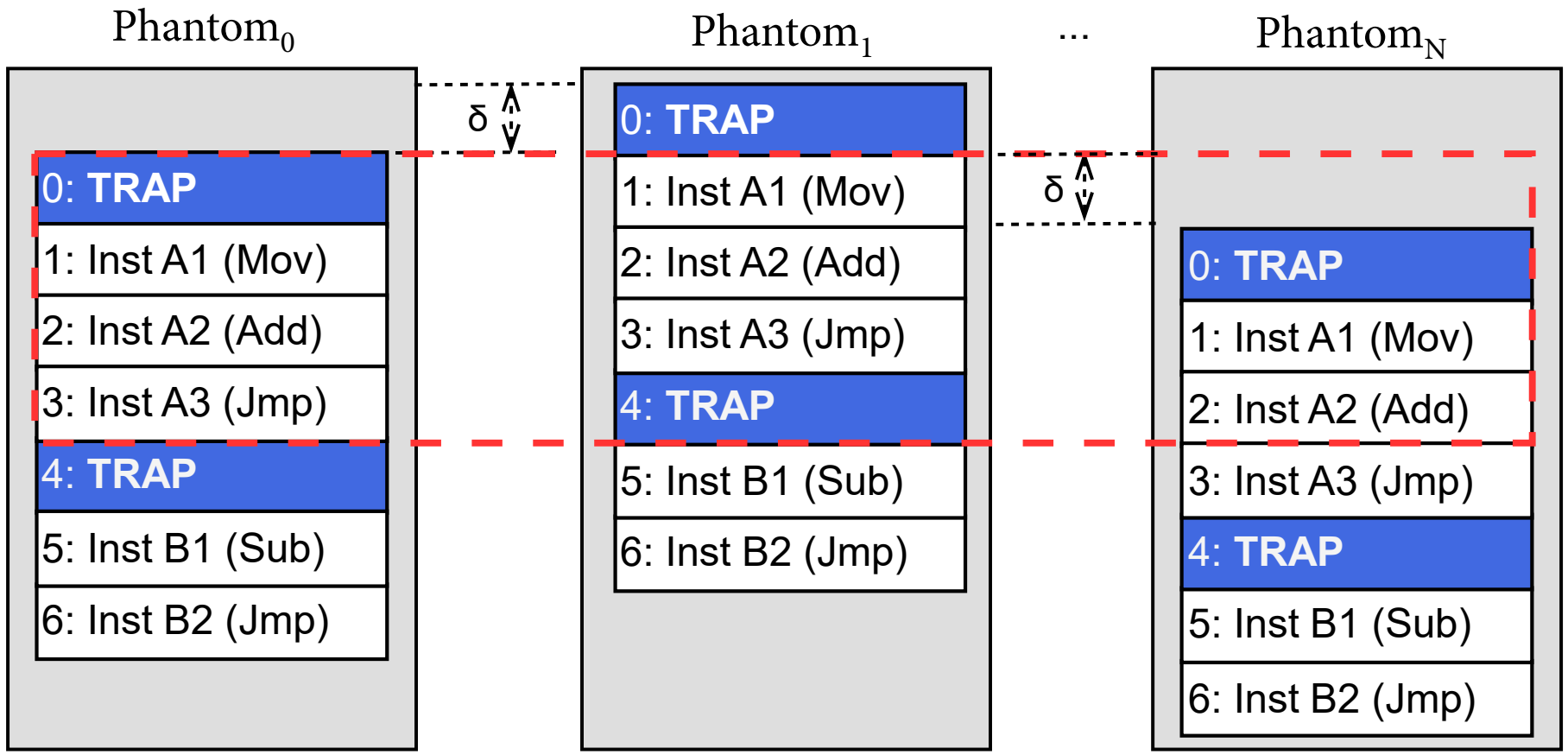
Normal Execution



Execution with CRA

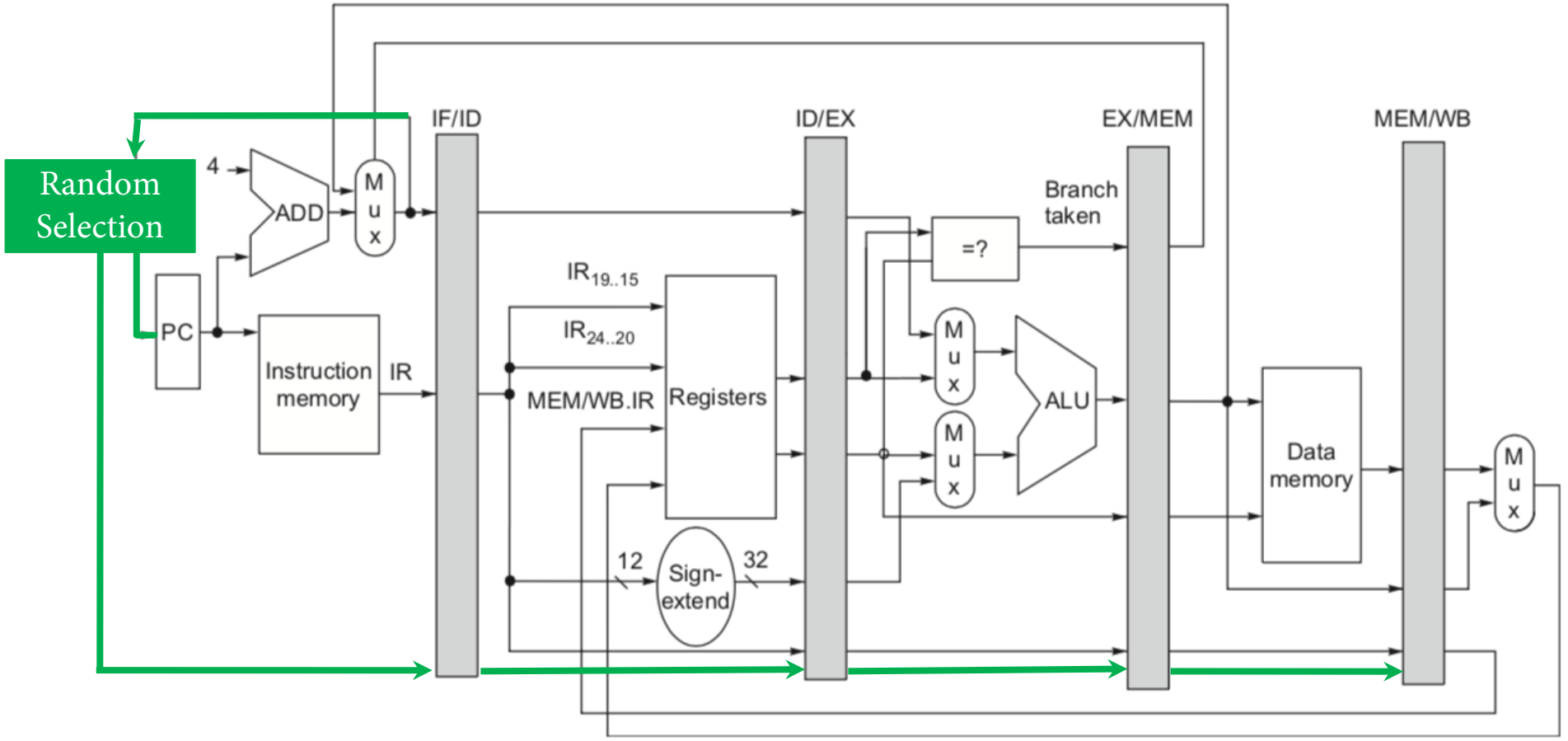
How does PNS precisely trap an attacker?

Code is instrumented with special instructions to throw an exception.



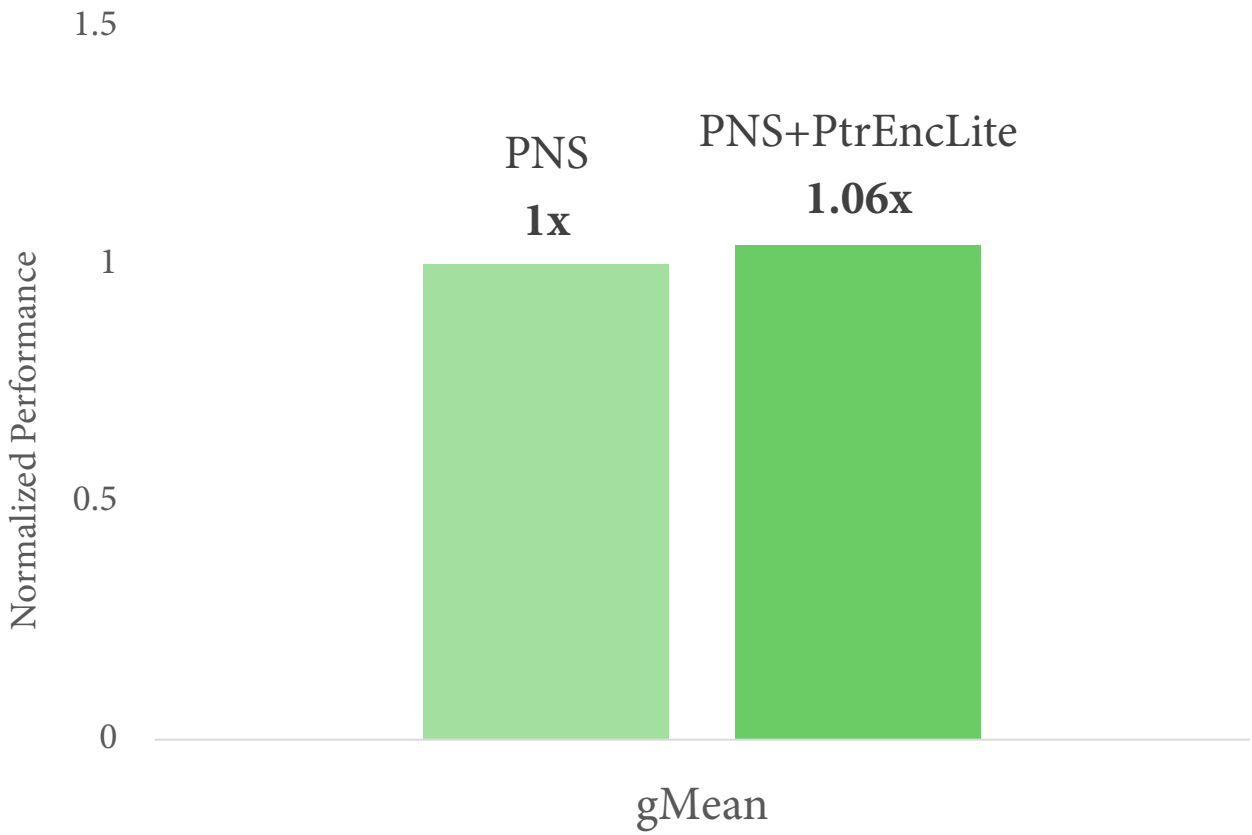
How is PNS implemented?

We do minimal changes to the processor frontend.



How was PNS evaluated?

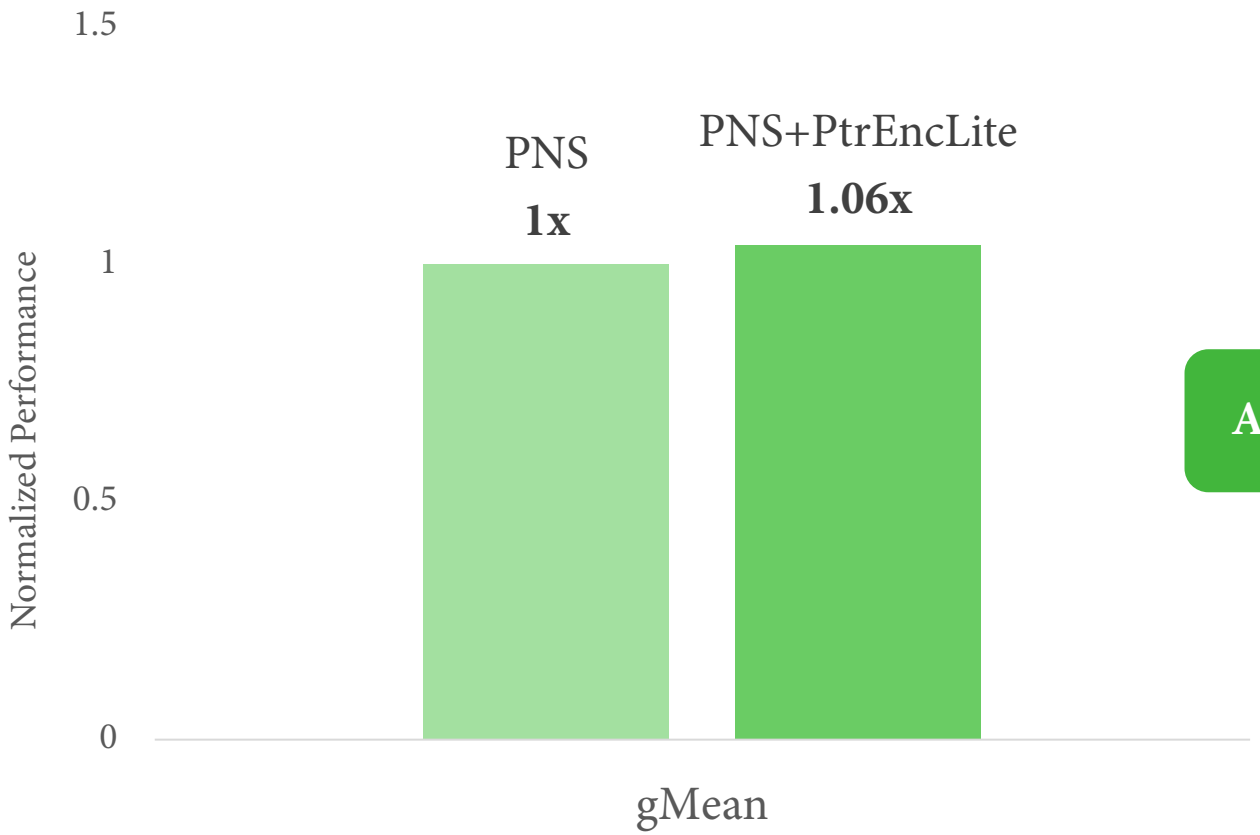
We used the gem5 architectural simulator to validate correctness & performance.



SPEC CPU2017

How was PNS evaluated?

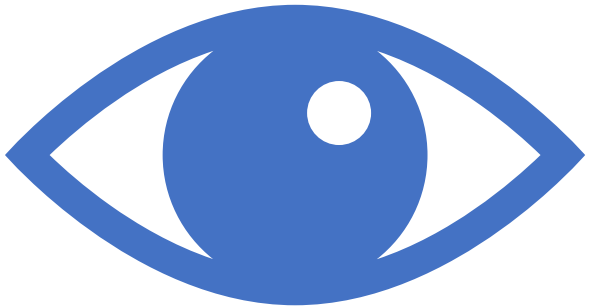
We used the gem5 architectural simulator to validate correctness & performance.



Average slowdown < 6%!

SPEC CPU2017

Limitations



Repeated Observation Attack

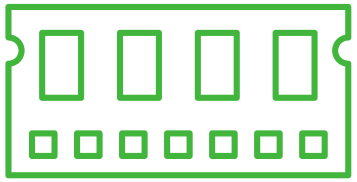
Running the same binary on a non-protected system can leak the security shift of a return address.

Why is PNS well suited for constrained devices?

It brings efficient protection with minimal cost.

Minimal Performance Impact

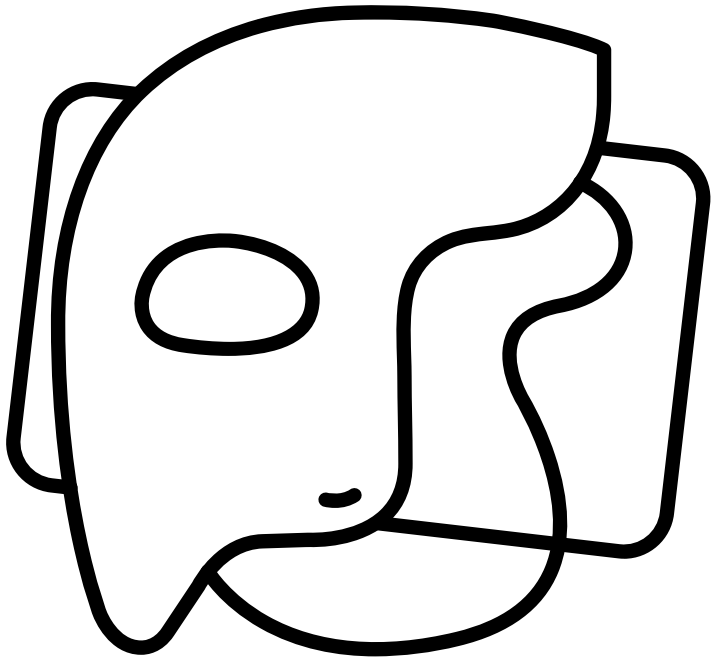
PNS has minimal impact on workload execution.



Memory Savings

PNS cuts down on resource duplication associated with aliasing multiple instructions.

Thanks for listening!



Phantom Name System

Find the paper here!

https://arxiv.org/pdf/1911.02038.pdf