Dedup Est Machina

Memory Deduplication as an Advanced Exploitation Vector

Erik Bosman, Kaveh Razavi, Herbert Bos and Cristiano Giuffrida

VU VRIJE UNIVERSITEIT AMSTERDAM

Deduplication
(software side-channel)
+
Rowhammer
(hardware bug)

Deduplication
(software side-channel)
+
Rowhammer
(hardware bug)

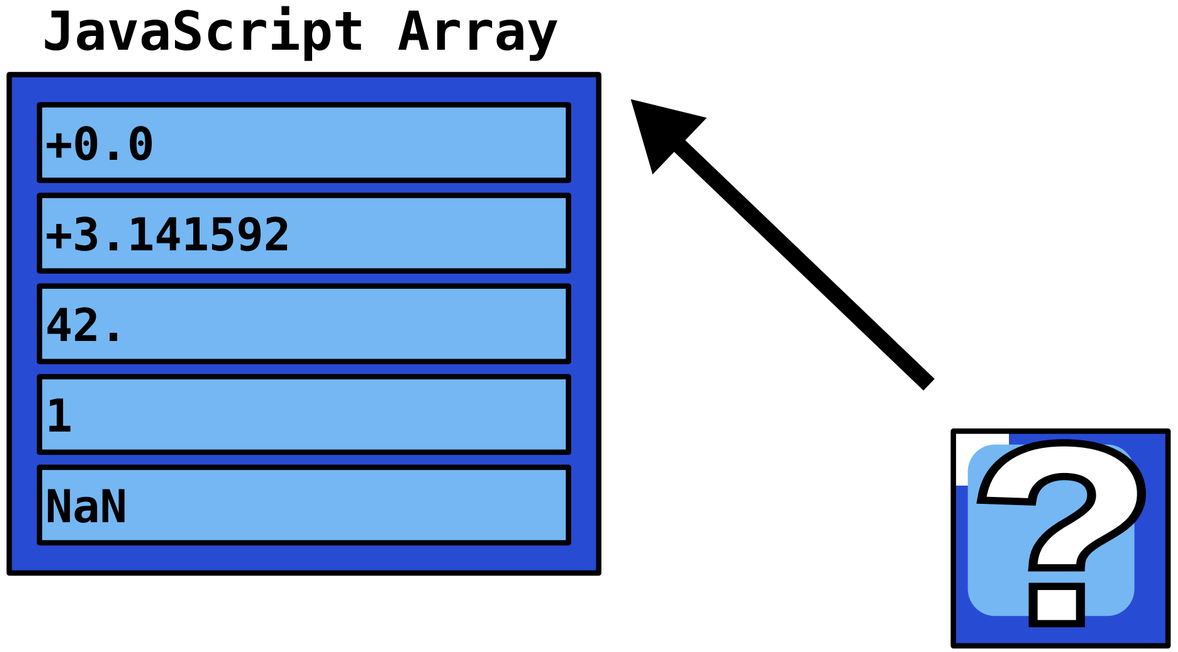
Exploit MS Edge without software bugs
(from JavaScript)

Deduplication

- leak heap & code pointer addresses

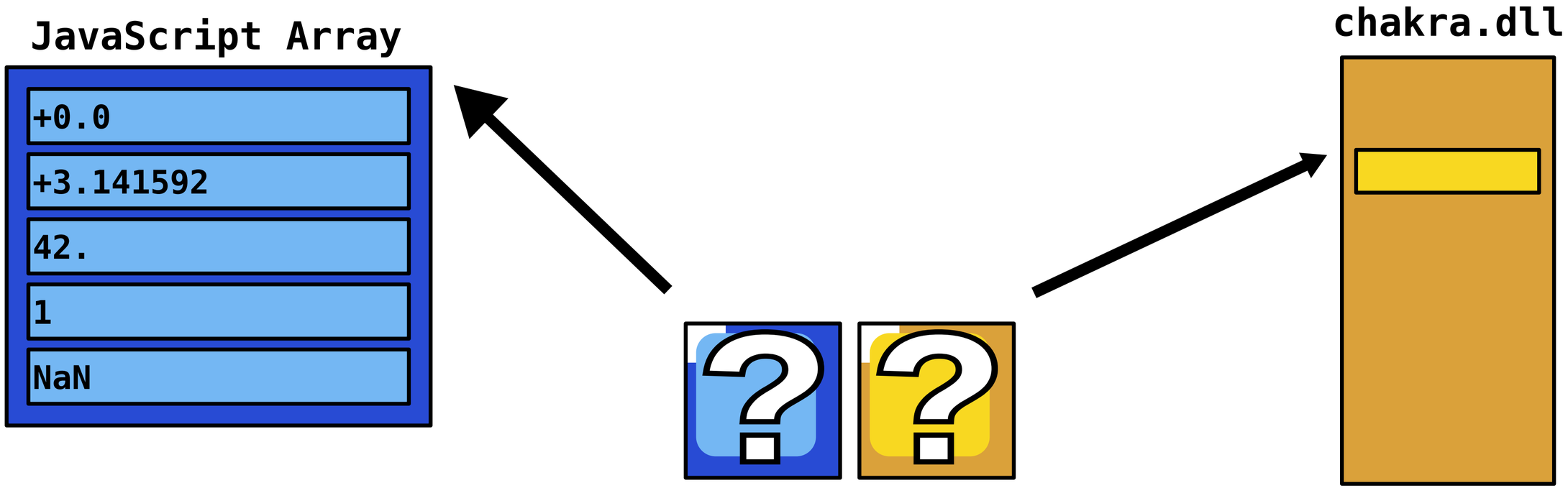
Deduplication

- leak heap & code pointer addresses



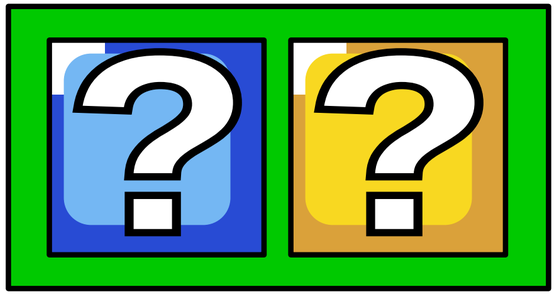
Deduplication

- leak heap & code pointer addresses



Deduplication

- leak heap & code pointer addresses
- create a fake object

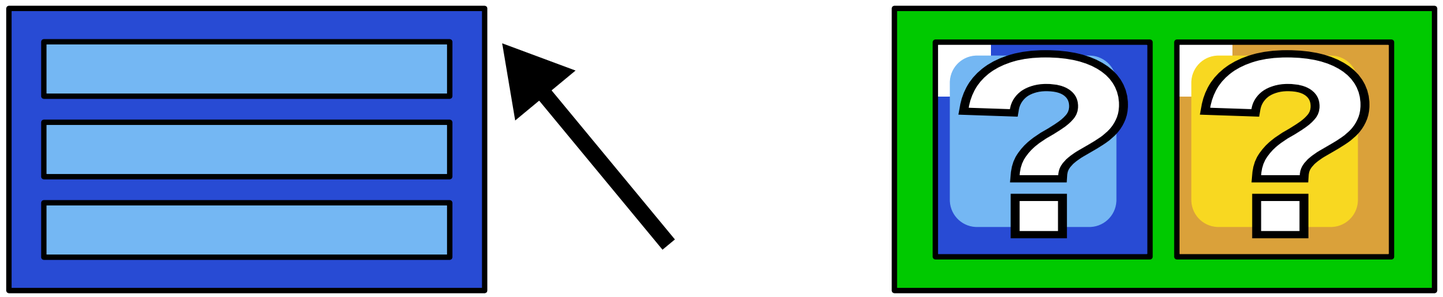


Deduplication

- leak heap & code pointer addresses
- create a fake object

Rowhammer

- create reference to our fake object

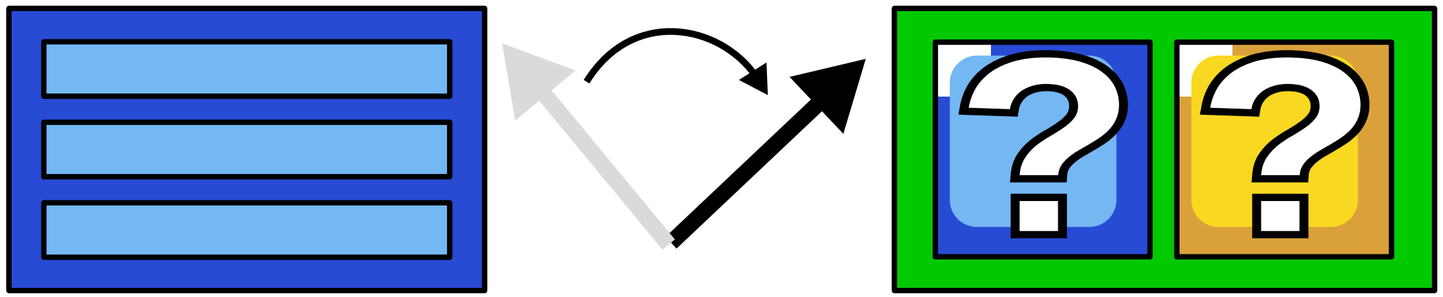


Deduplication

- leak heap & code pointer addresses
- create a fake object

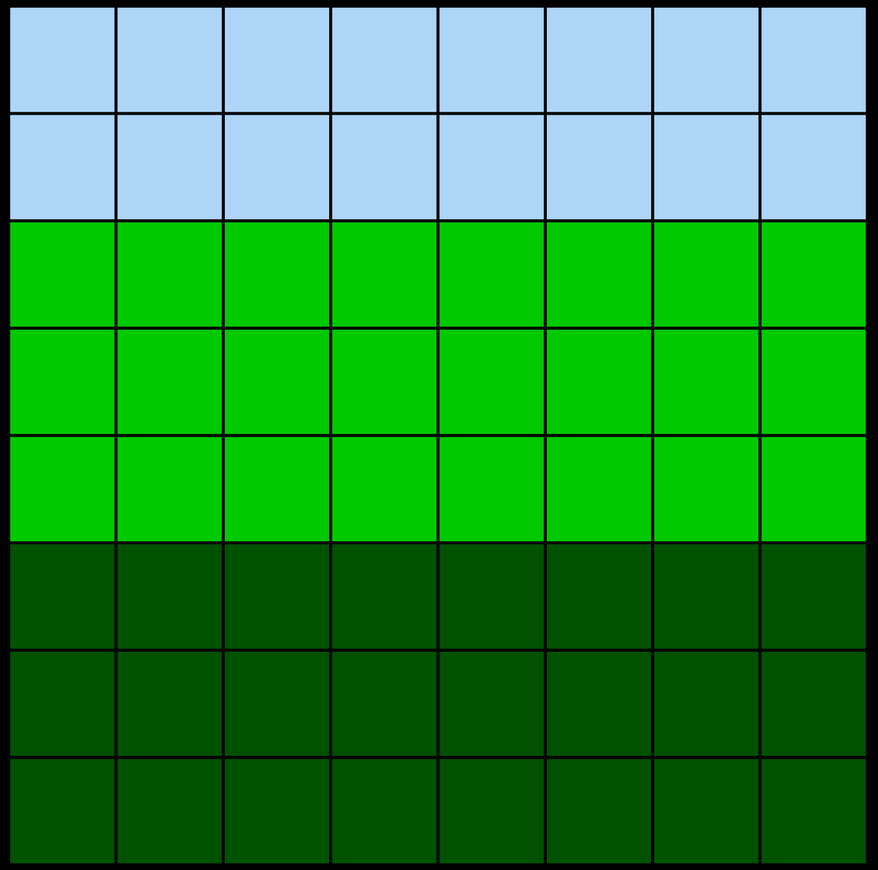
Rowhammer

- create reference to our fake object

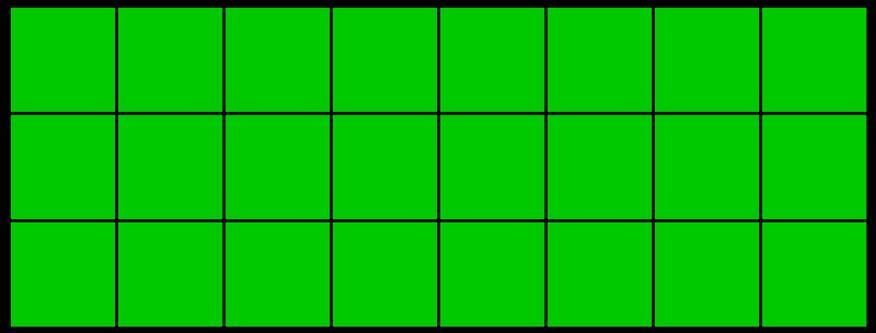


deduplication side-channel attack

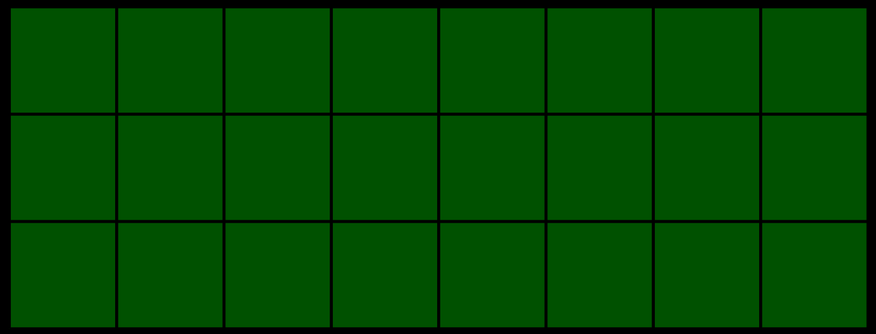
physical memory



attacker process

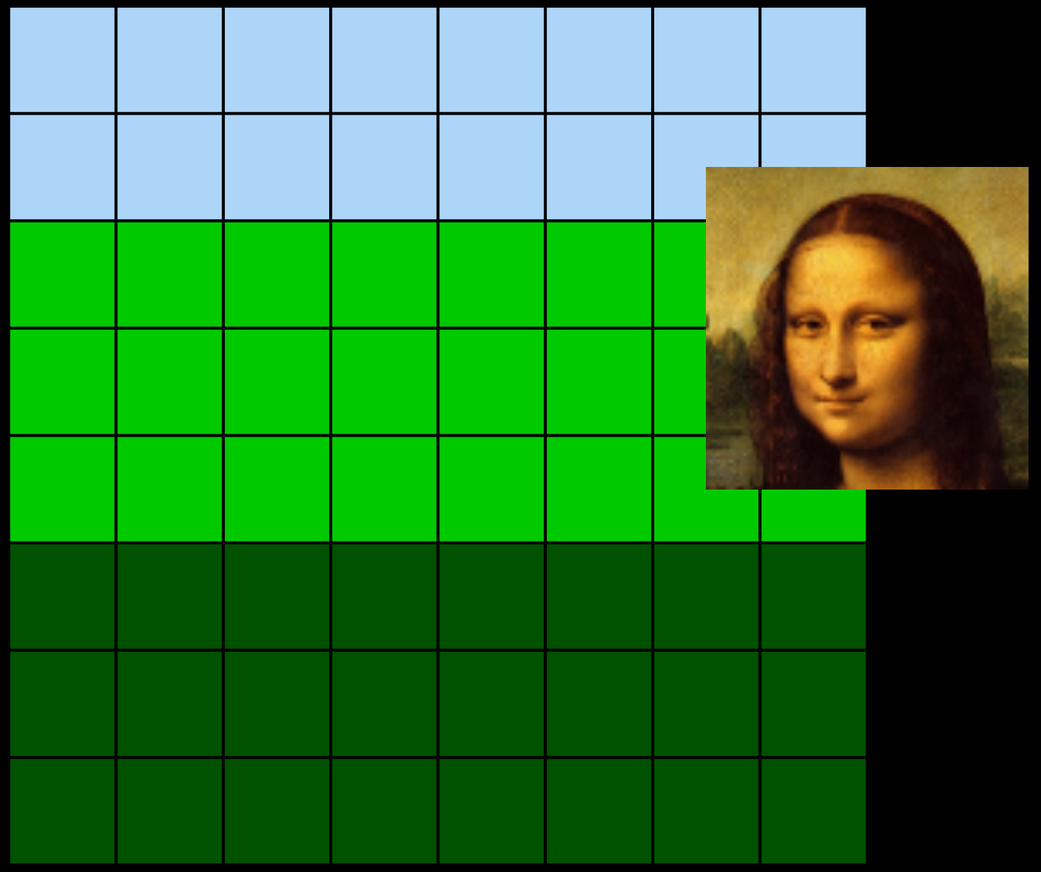


victim process

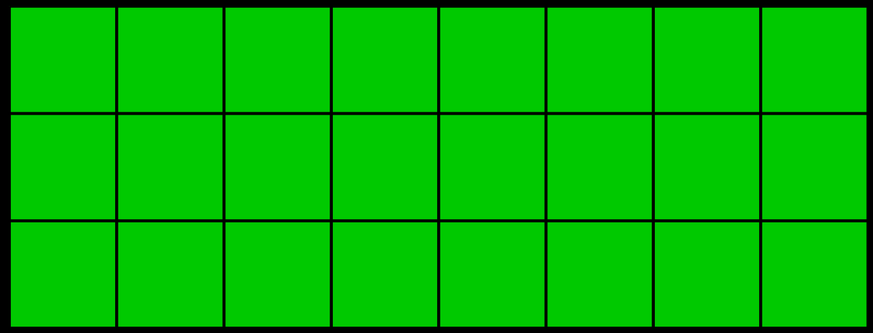


deduplication side-channel attack

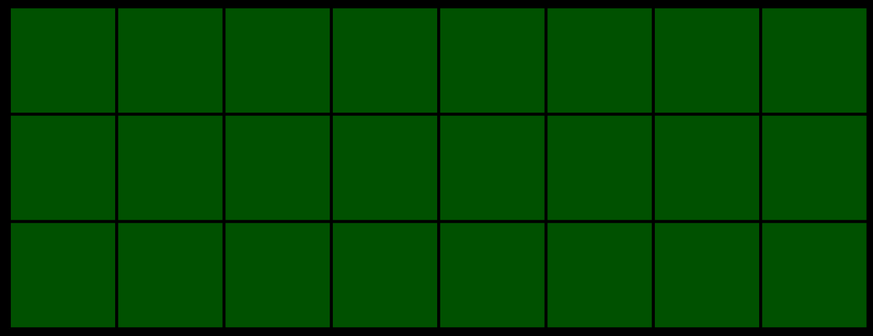
physical memory



attacker process

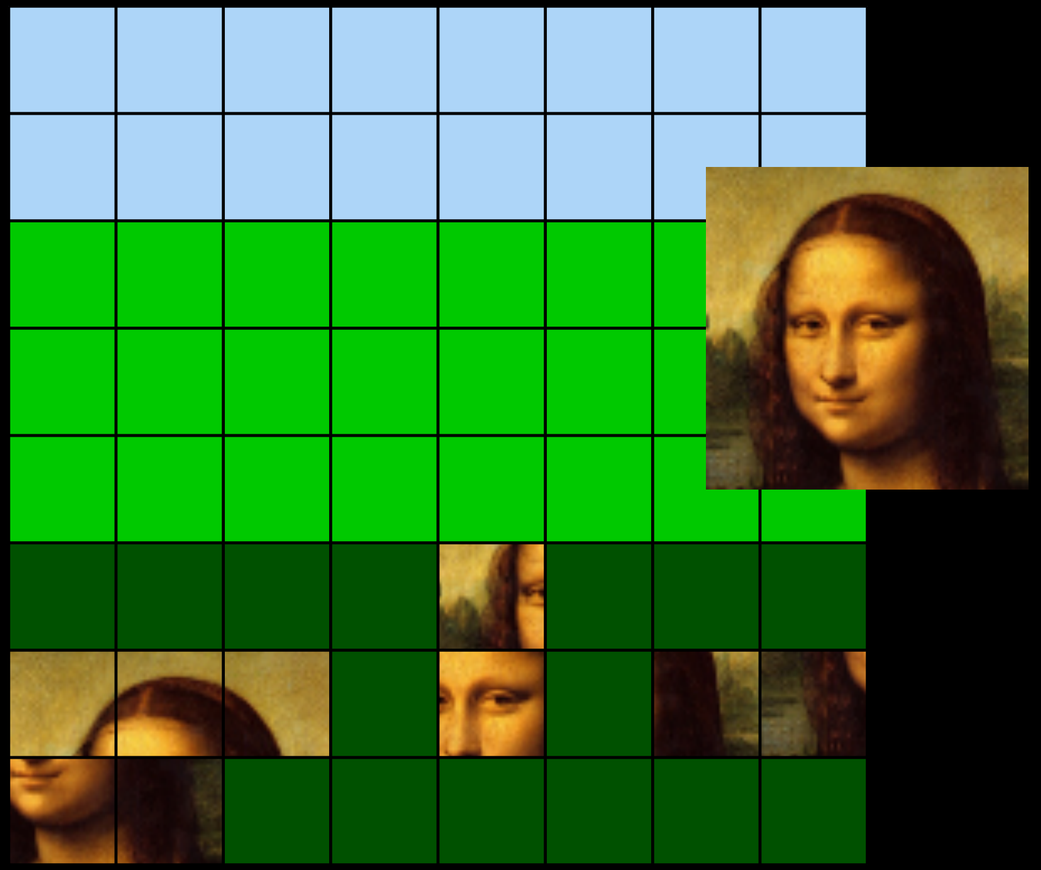


victim process

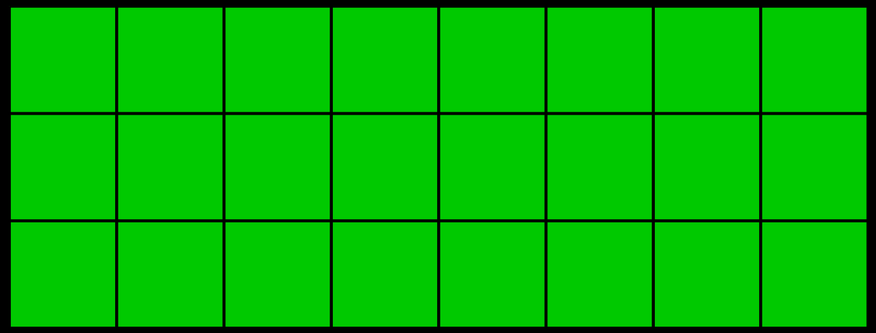


deduplication side-channel attack

physical memory



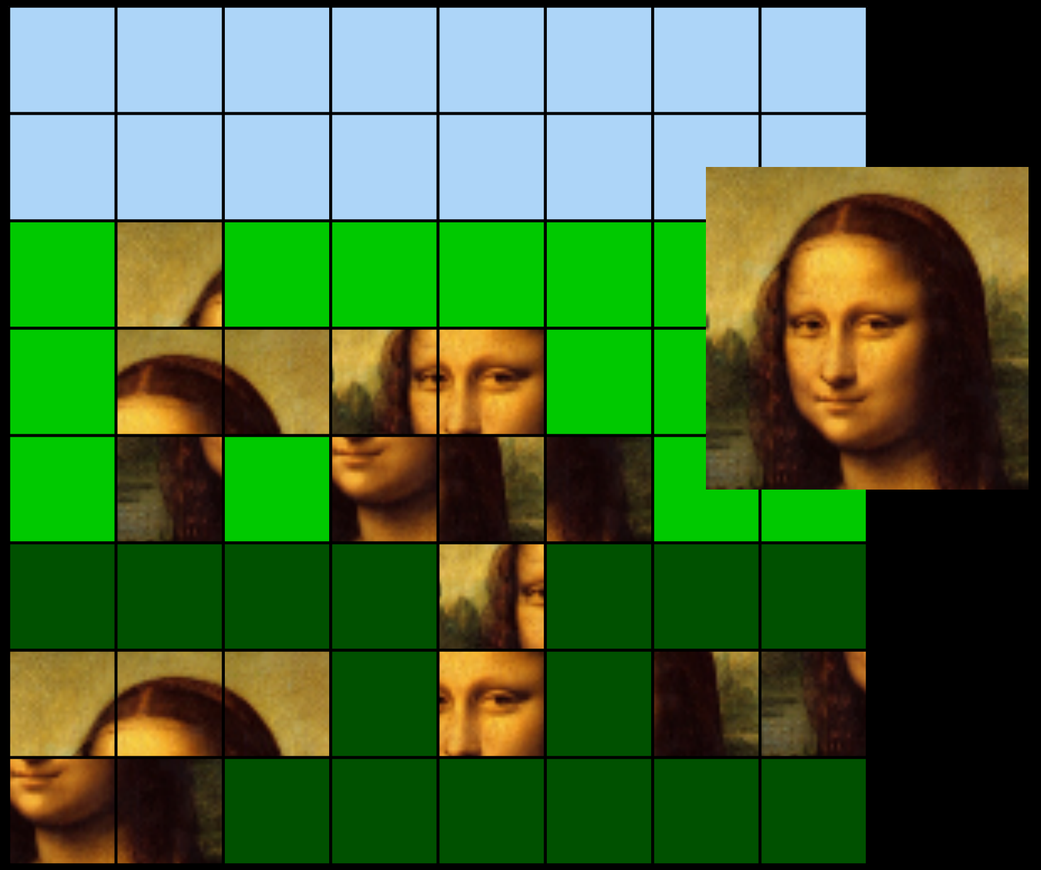
attacker process



victim process

deduplication side-channel attack

physical memory

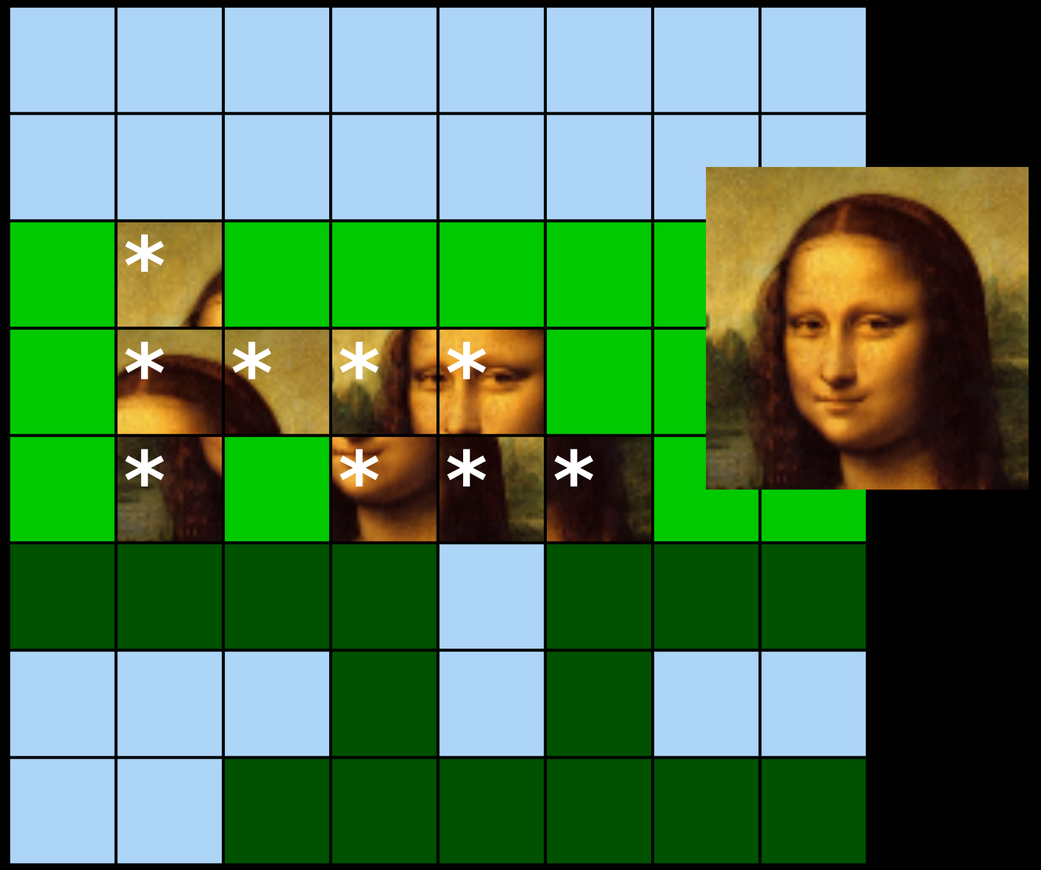


attacker process

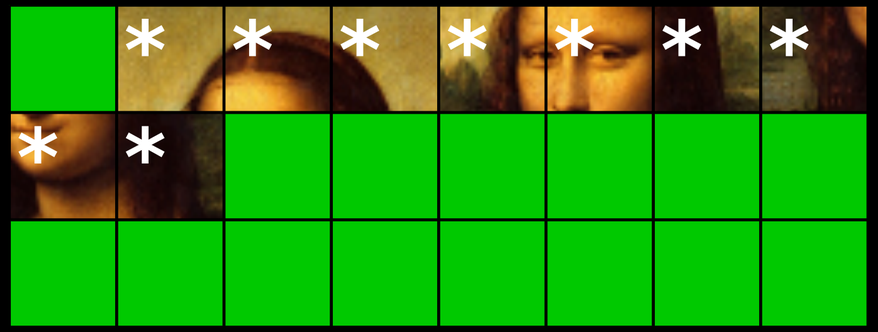
victim process

deduplication side-channel attack

physical memory



attacker process



victim process

deduplication side-channel attack

normal write

deduplication side-channel attack

normal write

deduplication side-channel attack

normal write

copy on write (due to deduplication)

deduplication side-channel attack

normal write

copy on write (due to deduplication)

Can we generalize this to leaking arbitrary data, like an ASLR pointer or a password?

Challenge 1:

The secret we want to leak does not span an entire page.

turning a secret into a page

secret

turning a secret into a page

secret

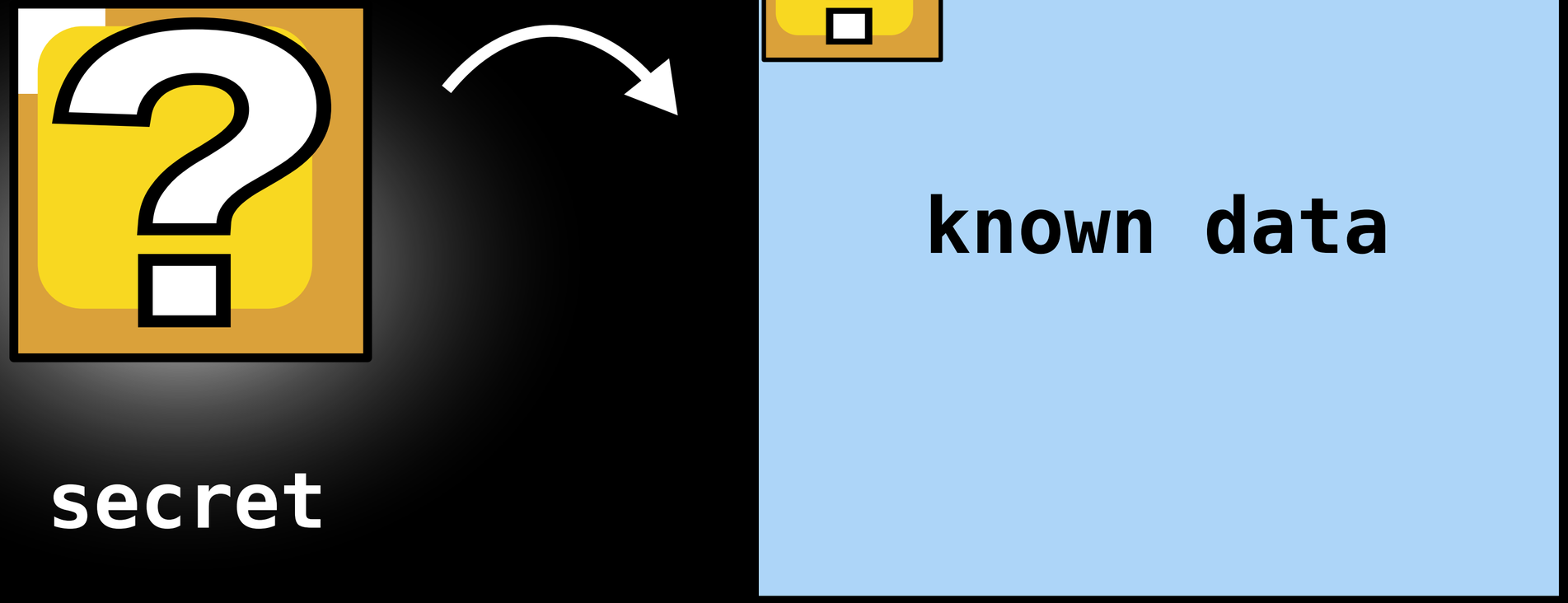
known data

secret page

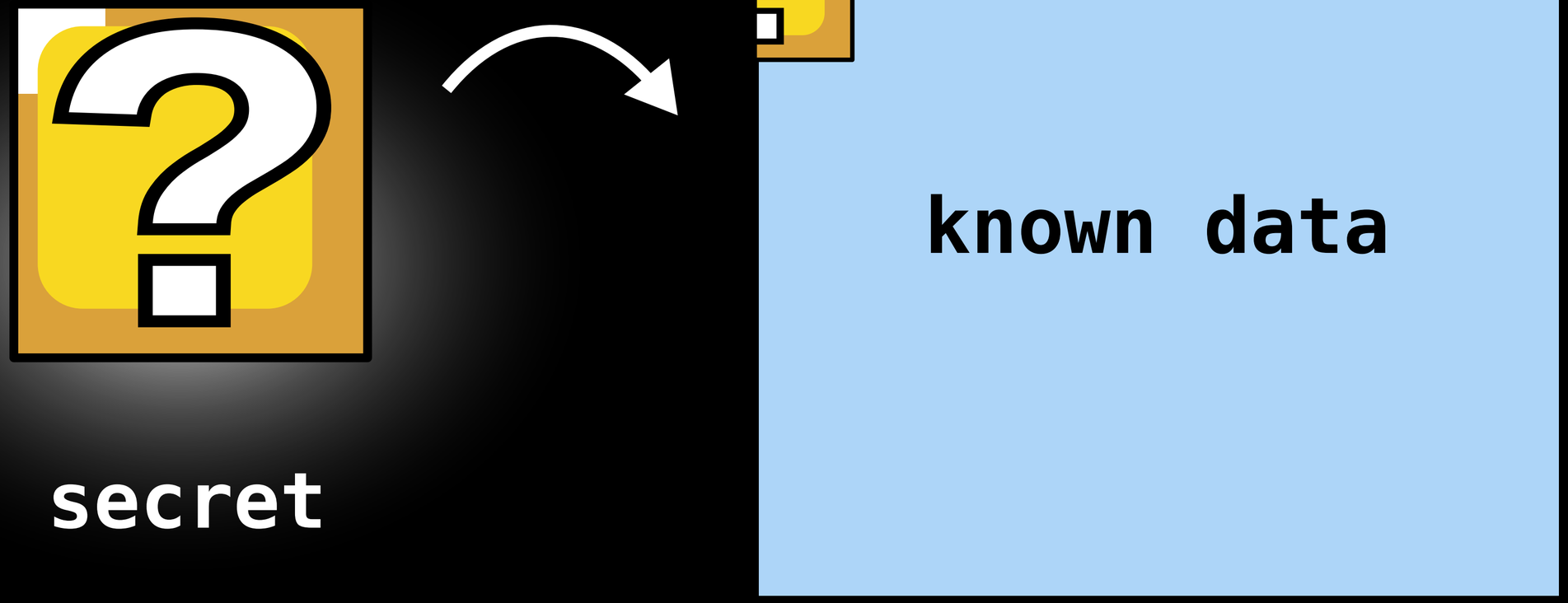
Challenge 2:

The secret we want to leak has too much entropy to leak all at once.

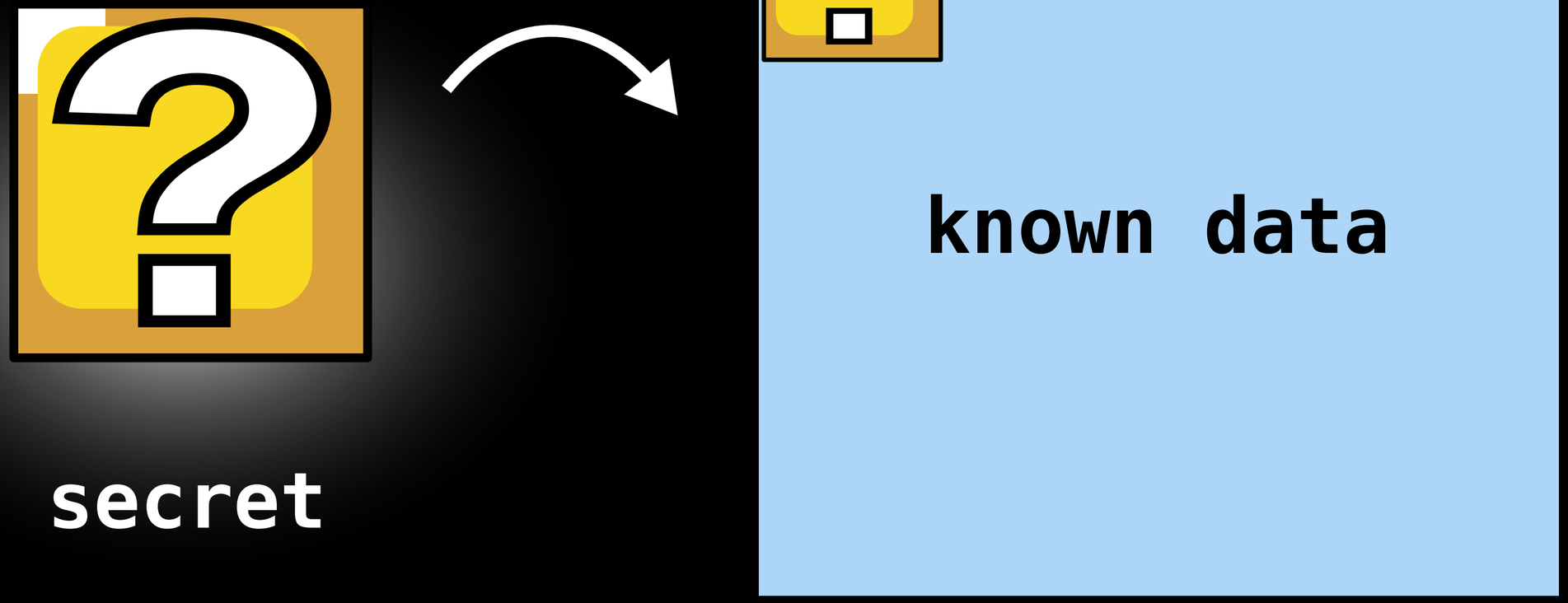
primitive #1: alignment probing



primitive #1: alignment probing

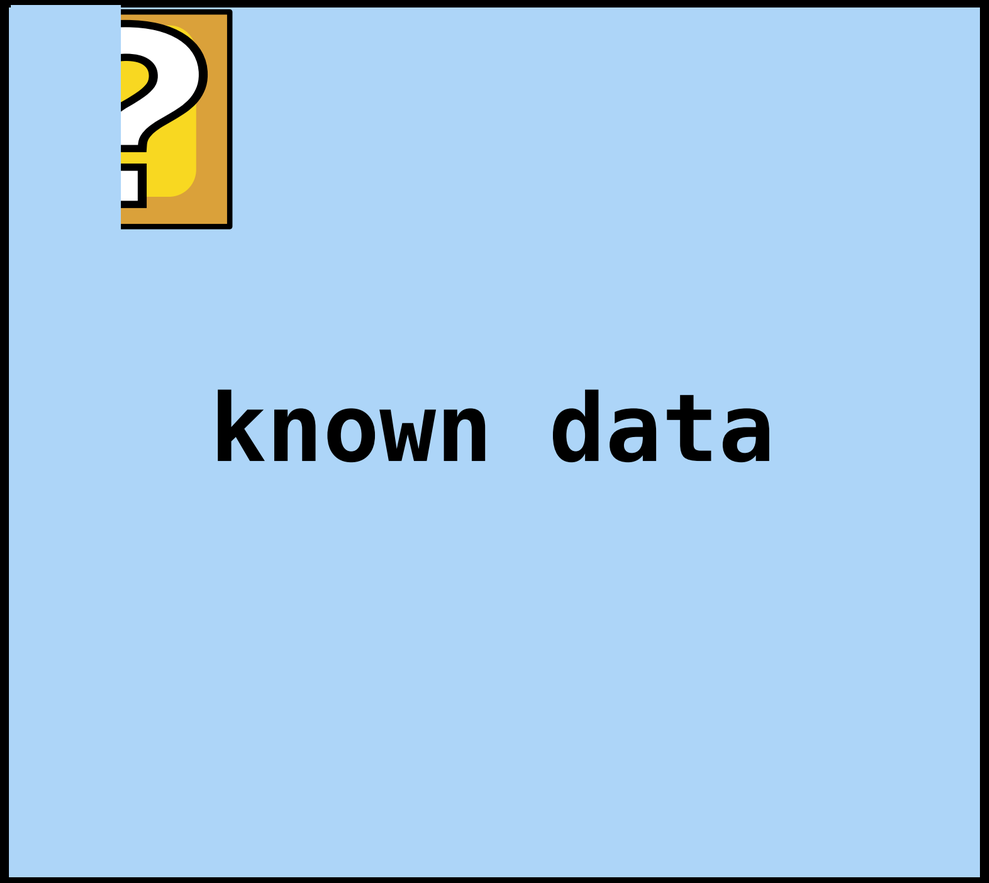


primitive #2: partial reuse



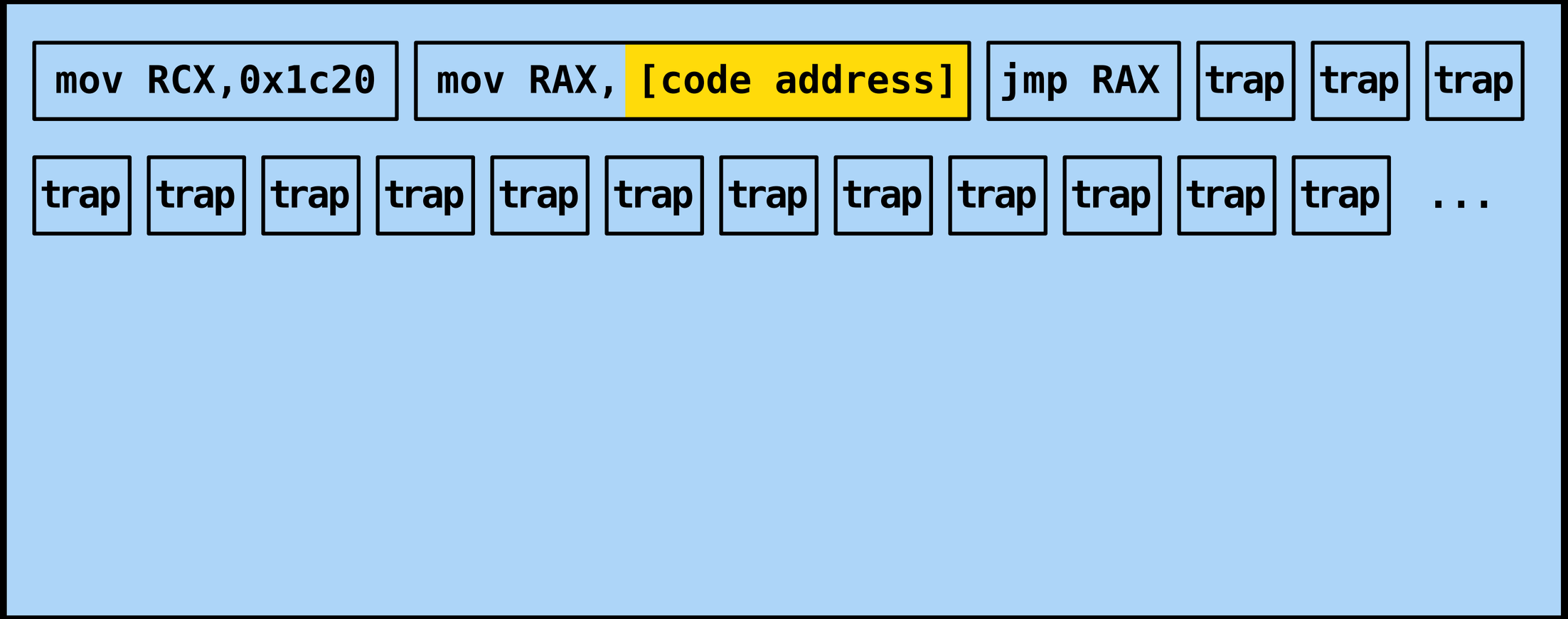
primitive #2: partial reuse

secret

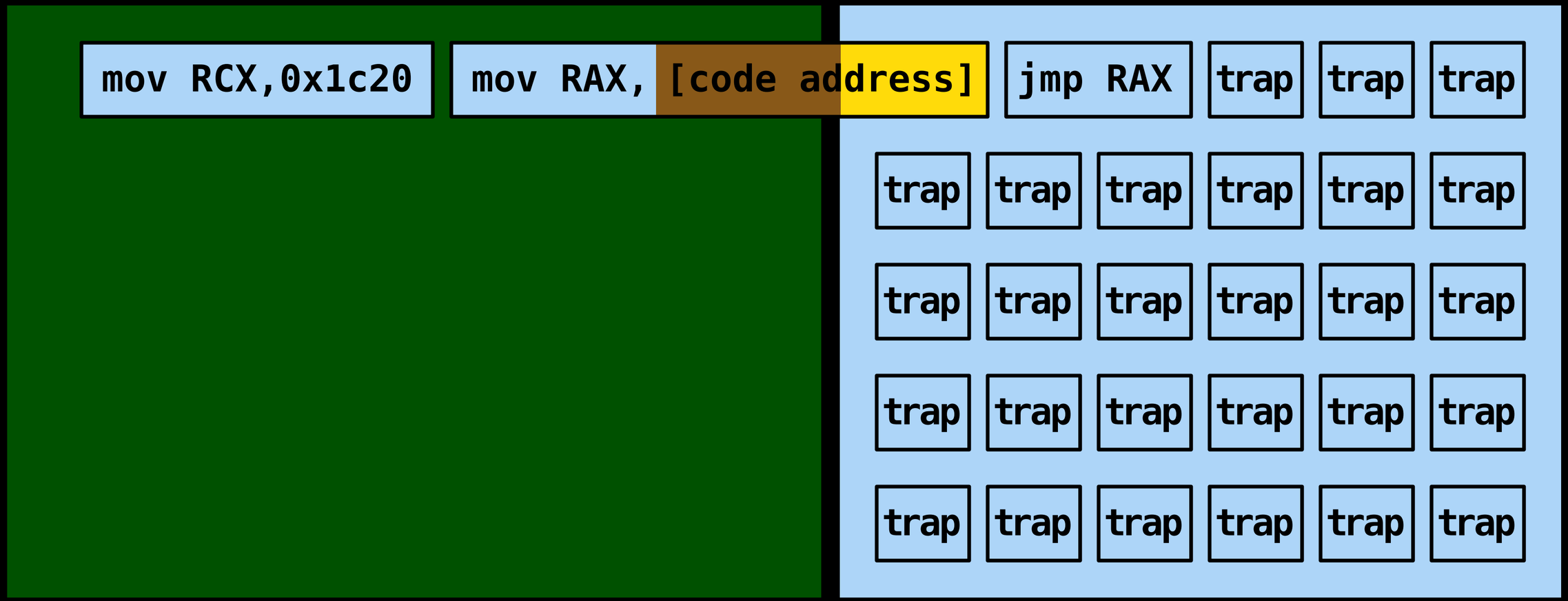


secret page

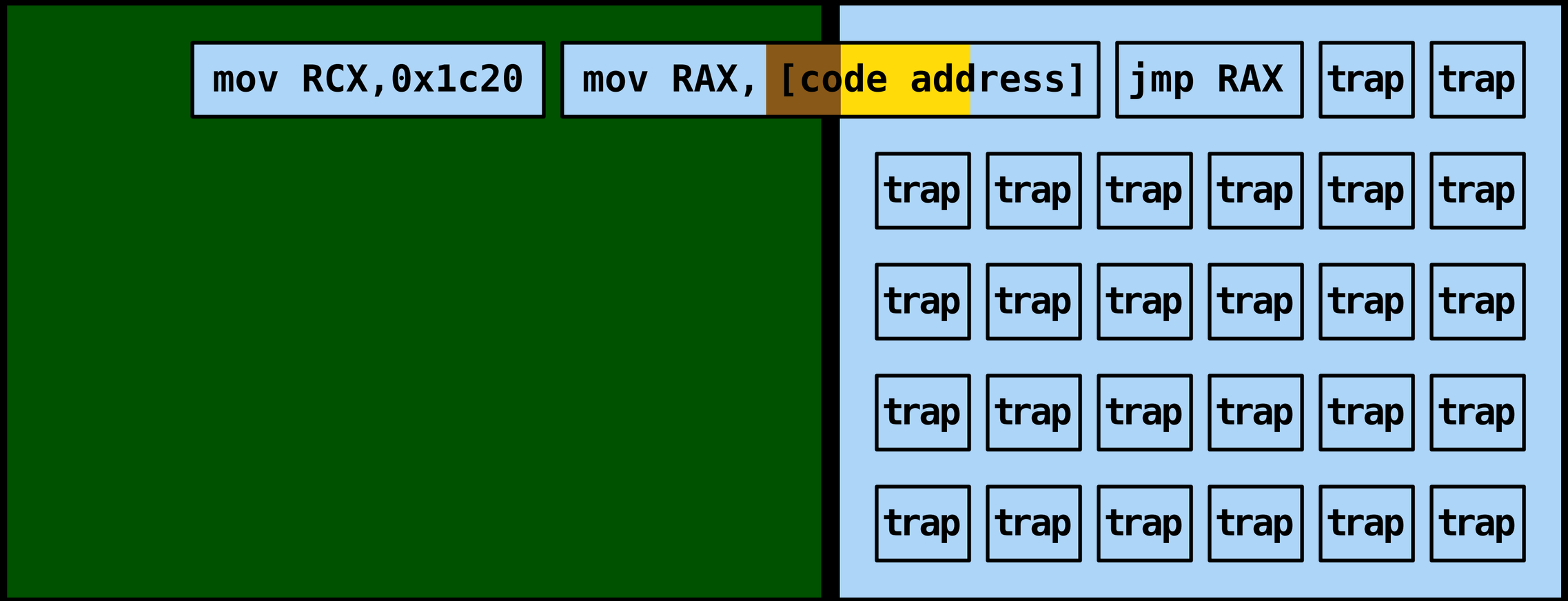
JIT function epilogue (MS Edge)



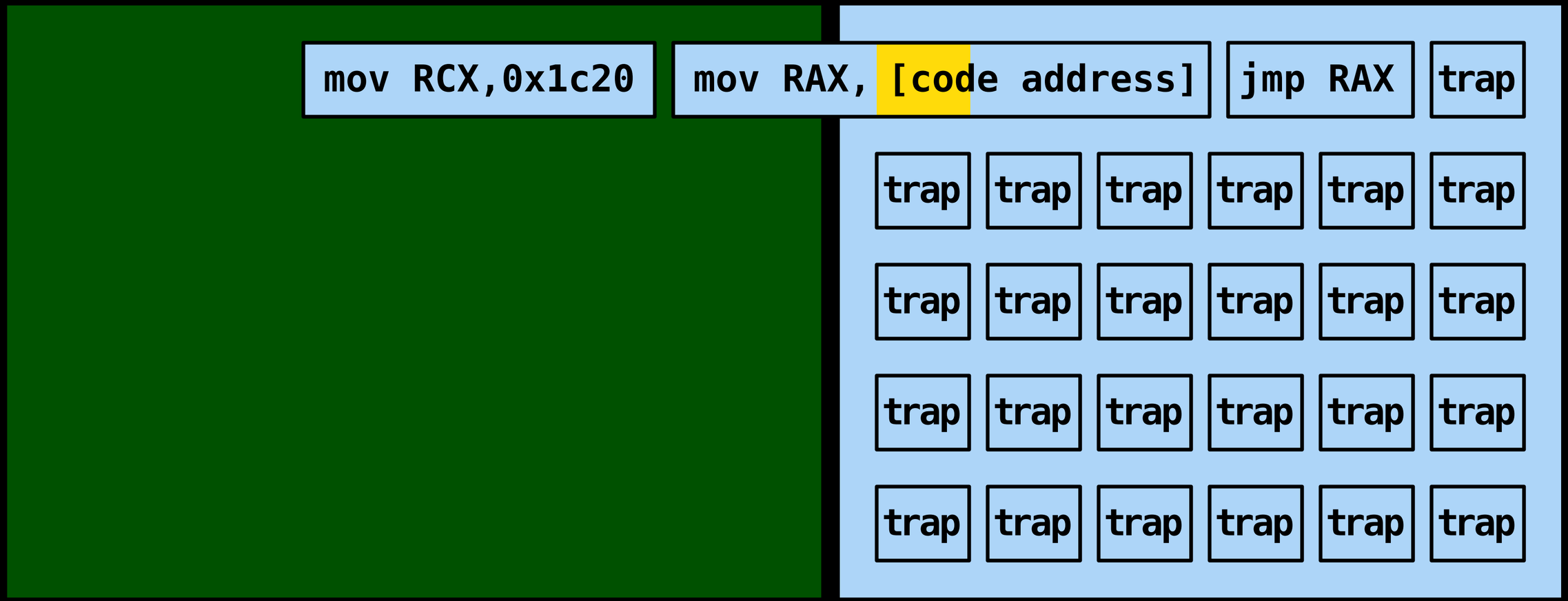
JIT function epilogue (MS Edge)



JIT function epilogue (MS Edge)



JIT function epilogue (MS Edge)



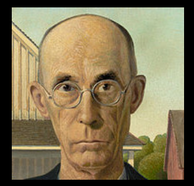
But what if we can't leak part of the secret?

birthday paradox

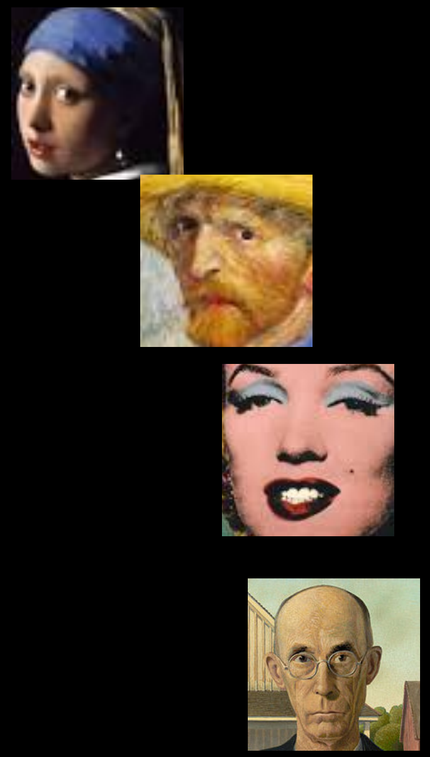
birthday paradox

birthday paradox

birthday paradox

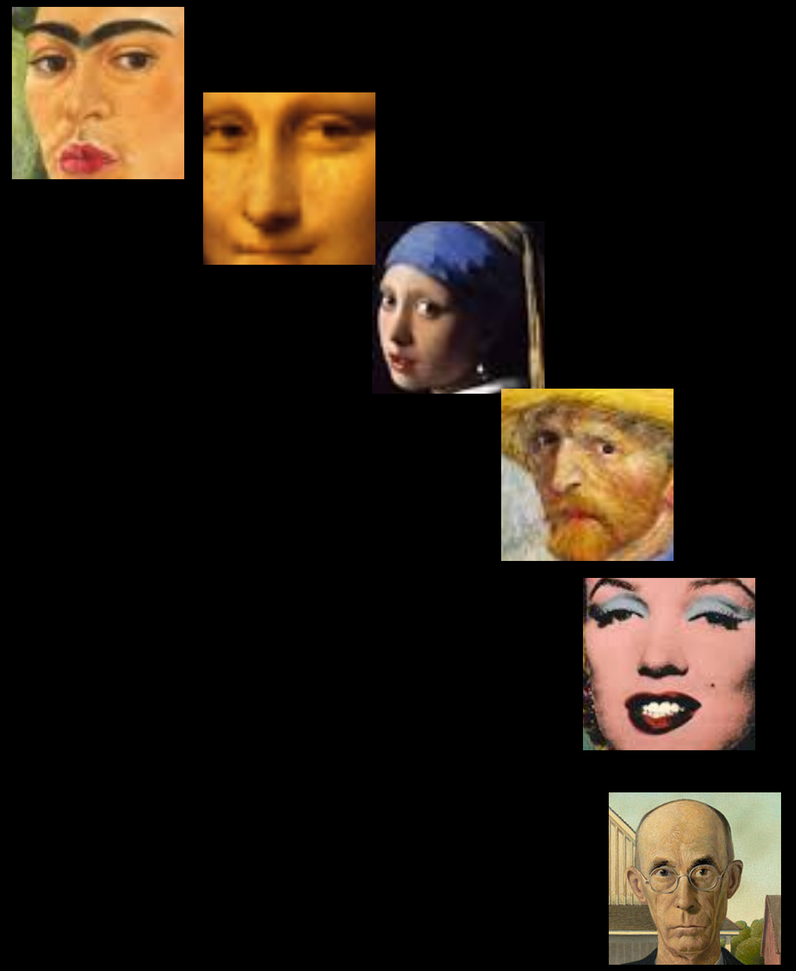


birthday paradox

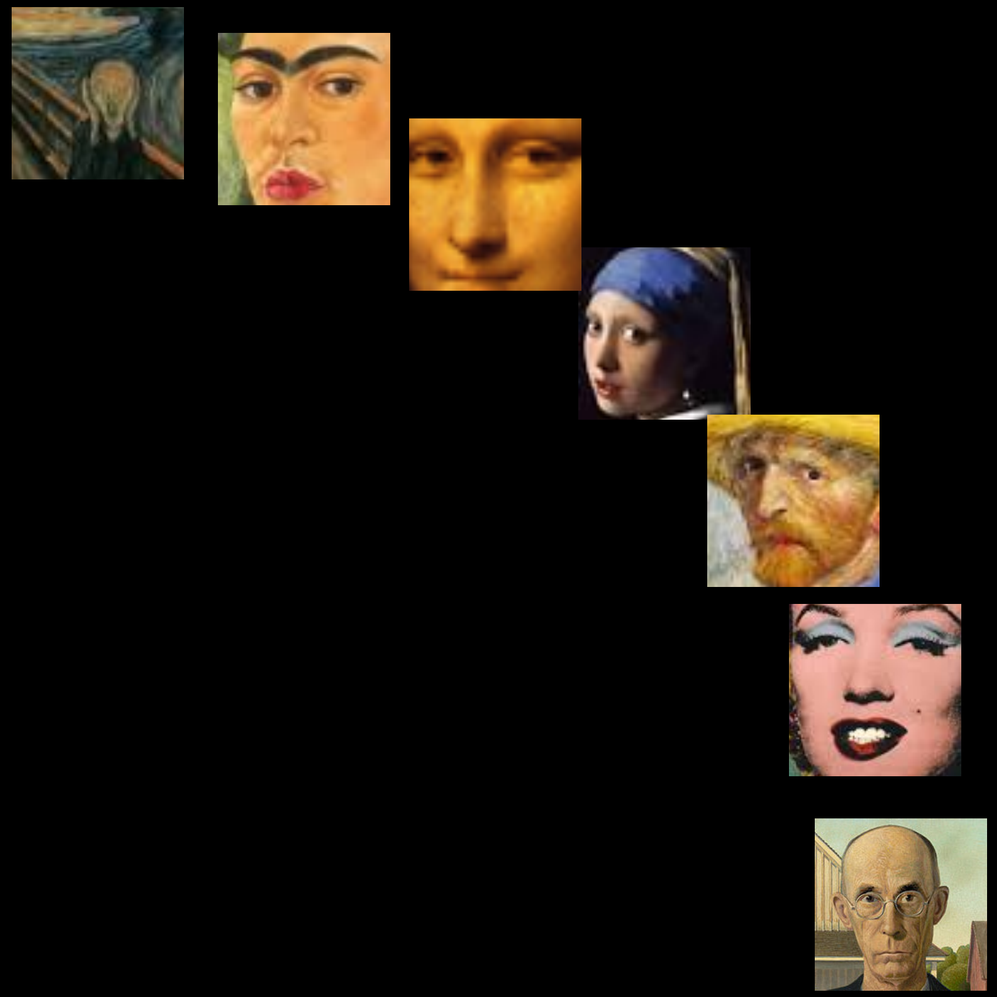


birthday paradox

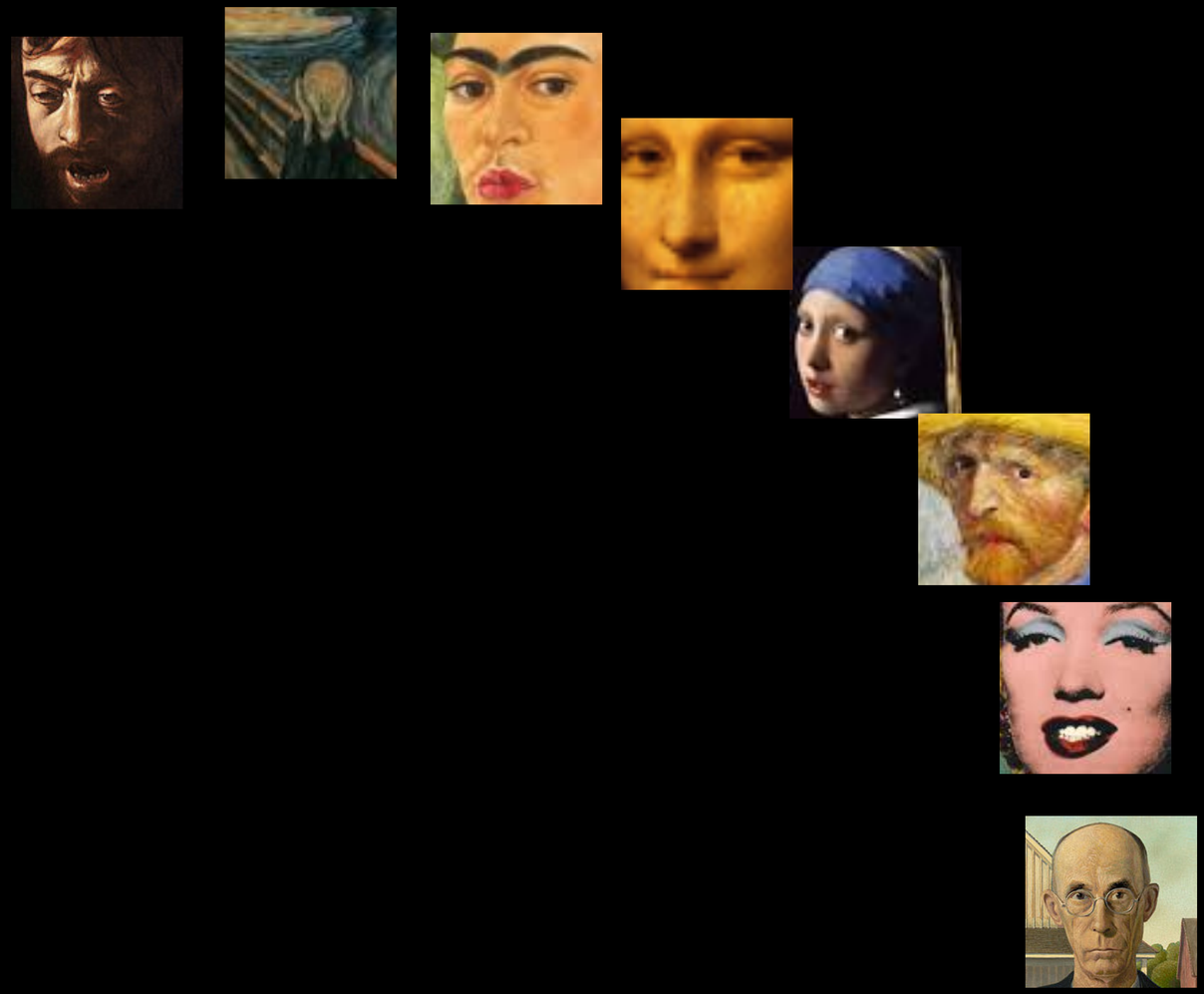
birthday paradox



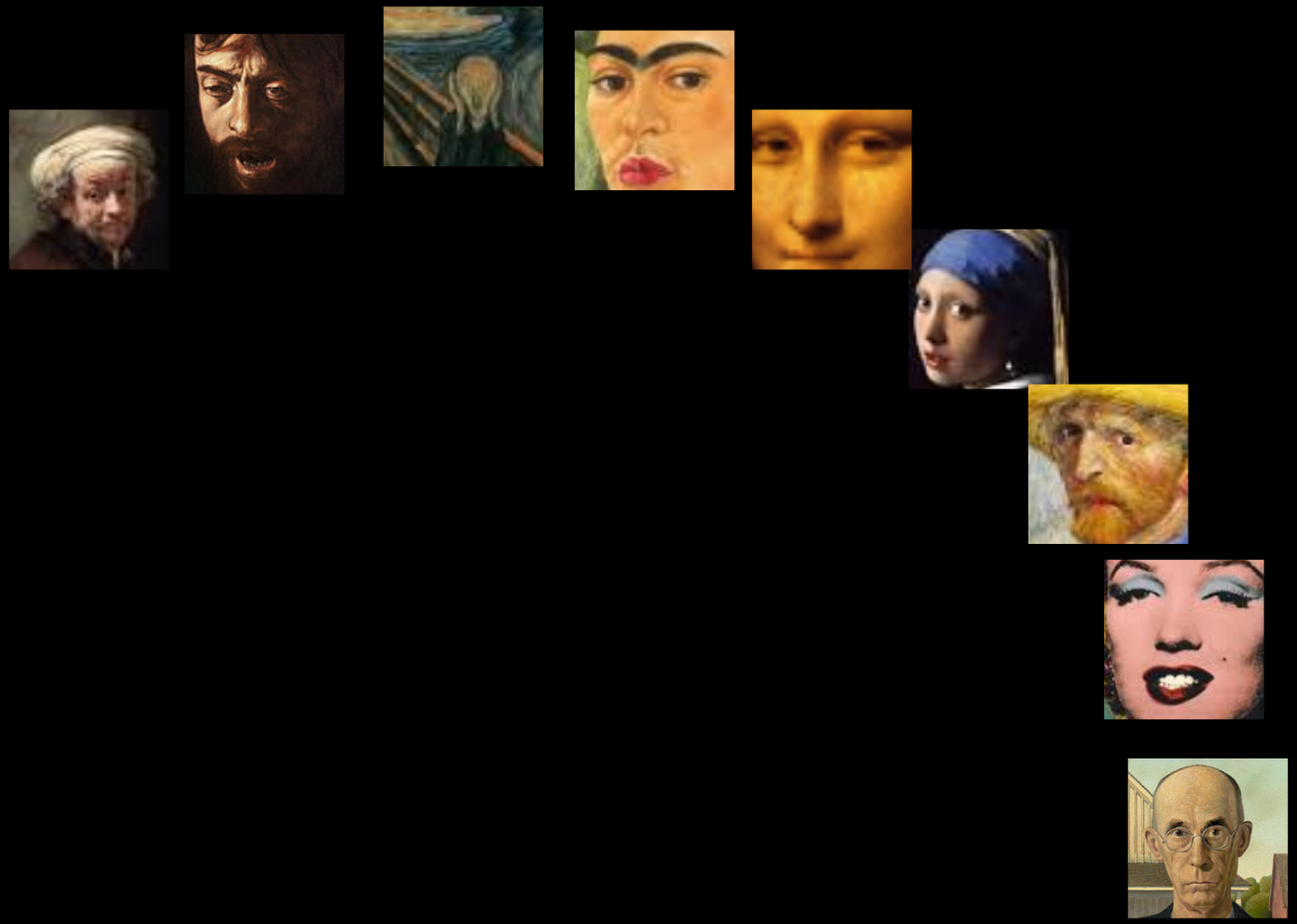
birthday paradox



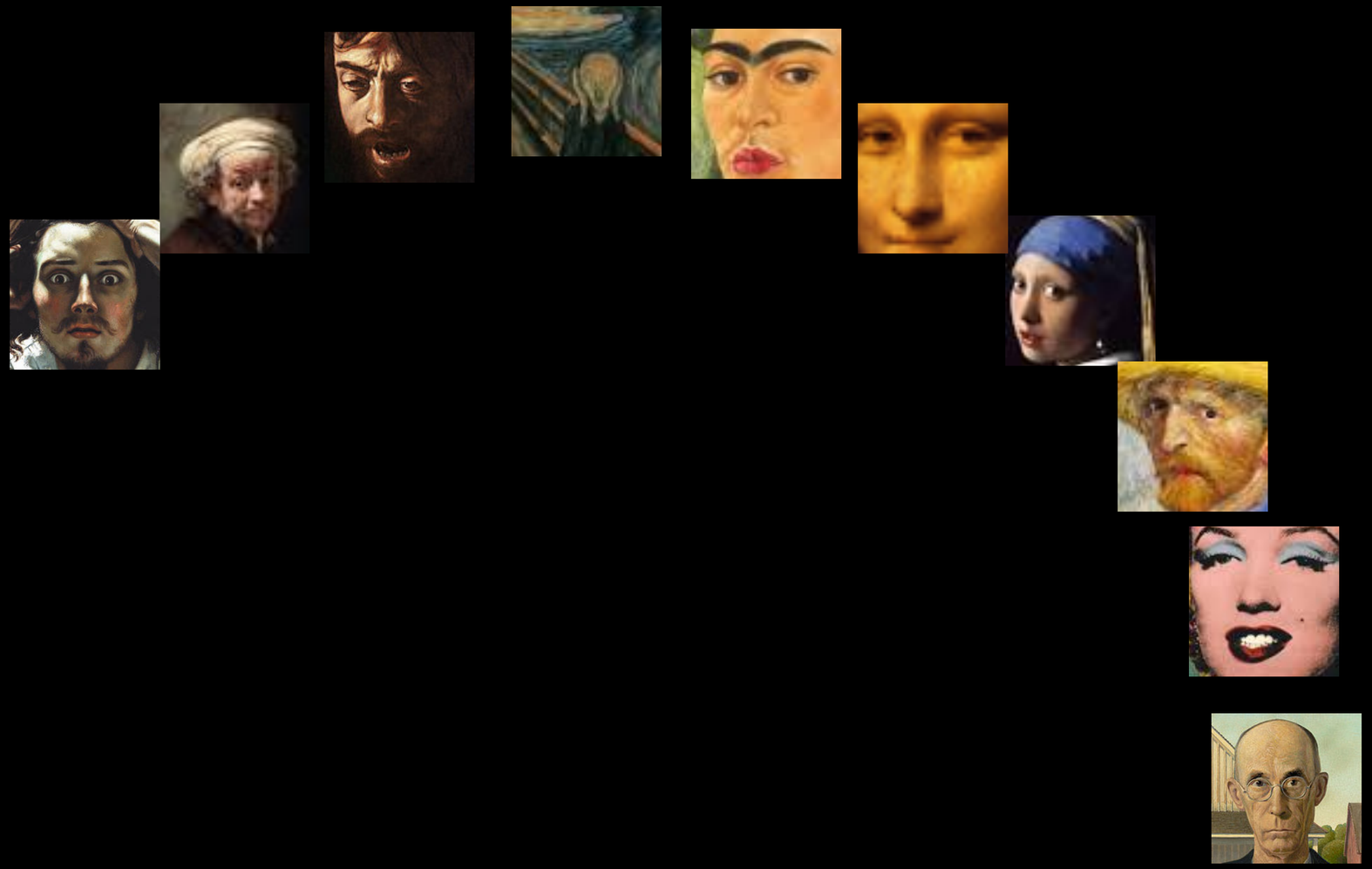
birthday paradox



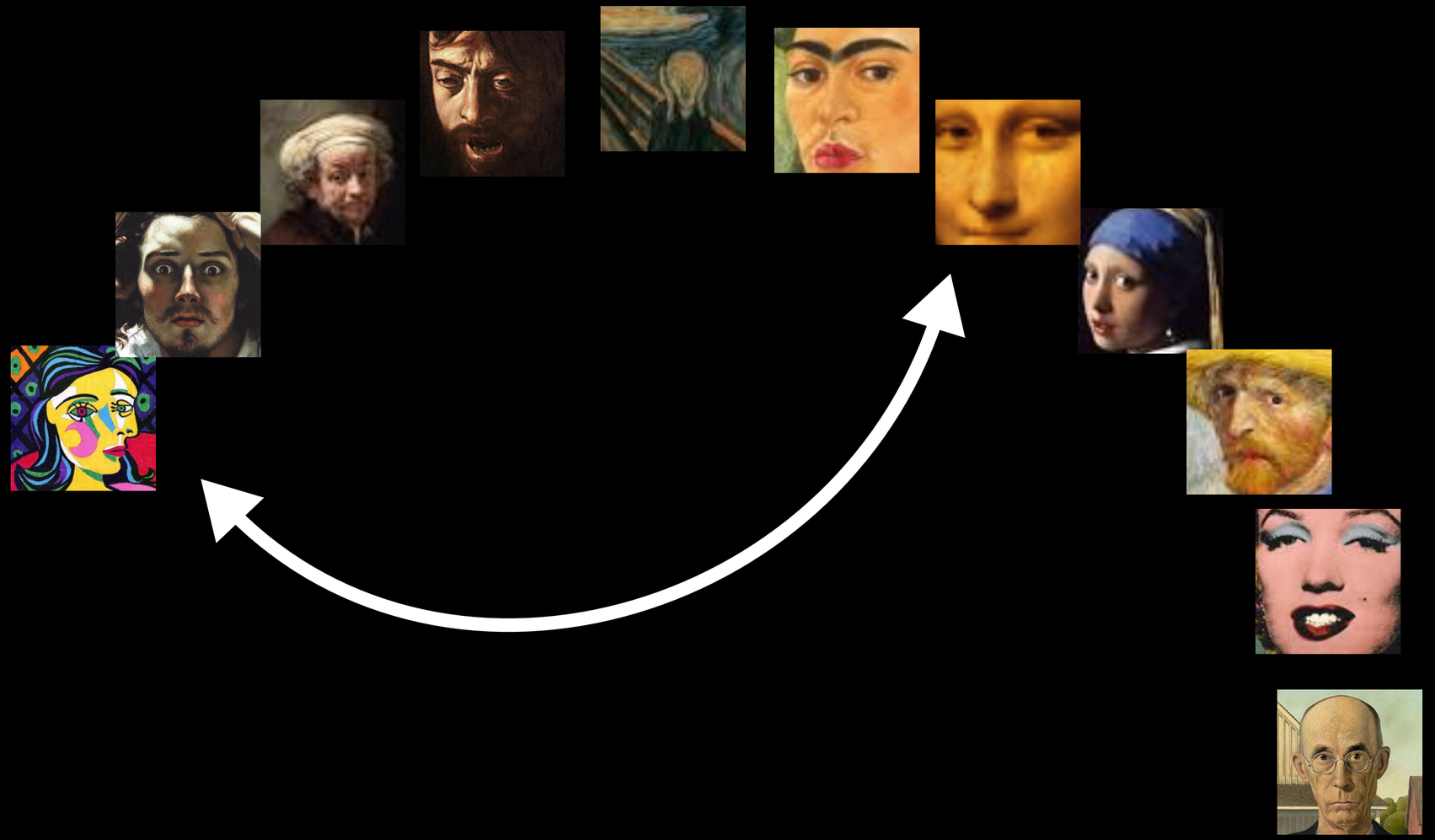
birthday paradox



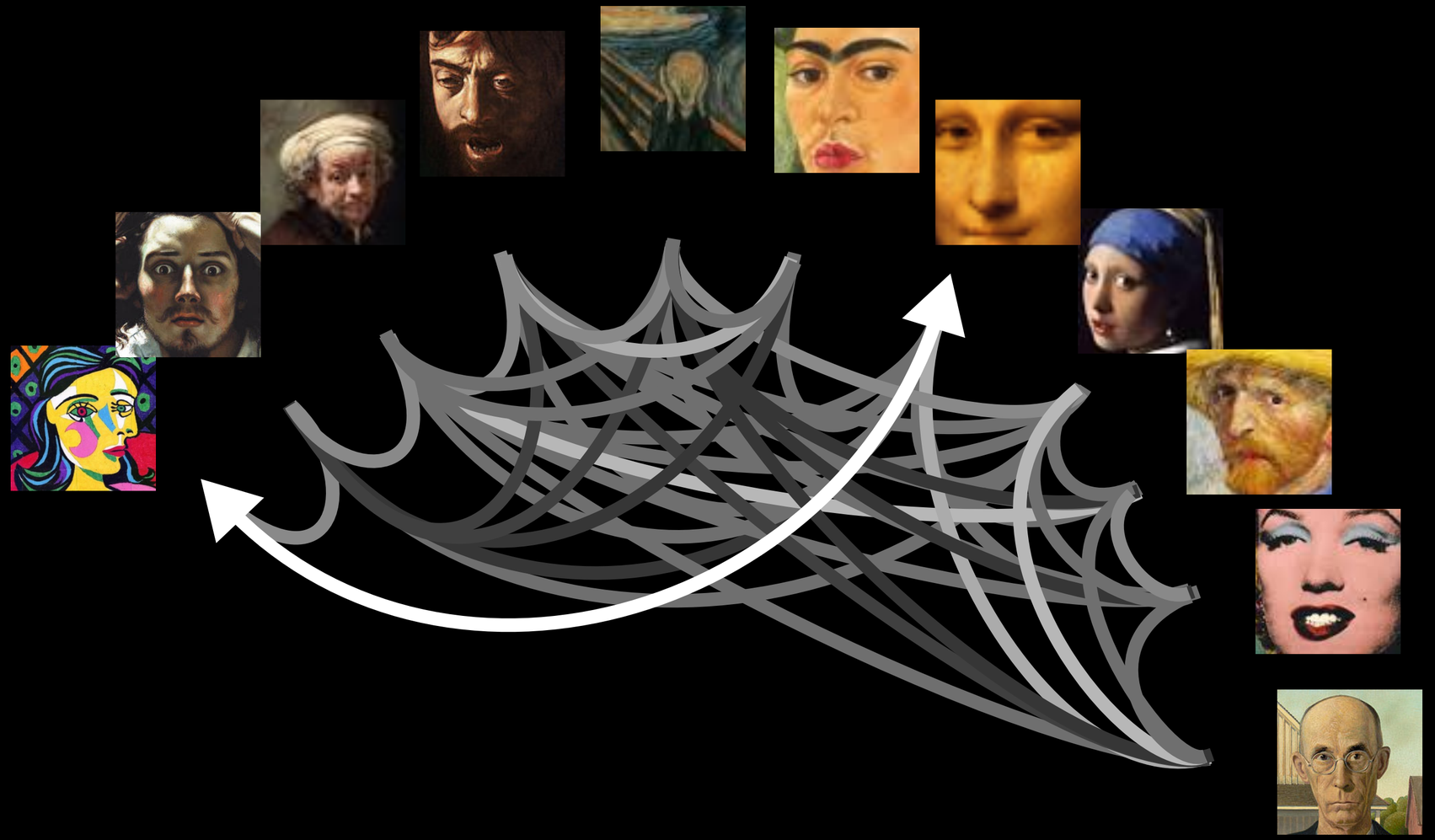
birthday paradox



birthday paradox

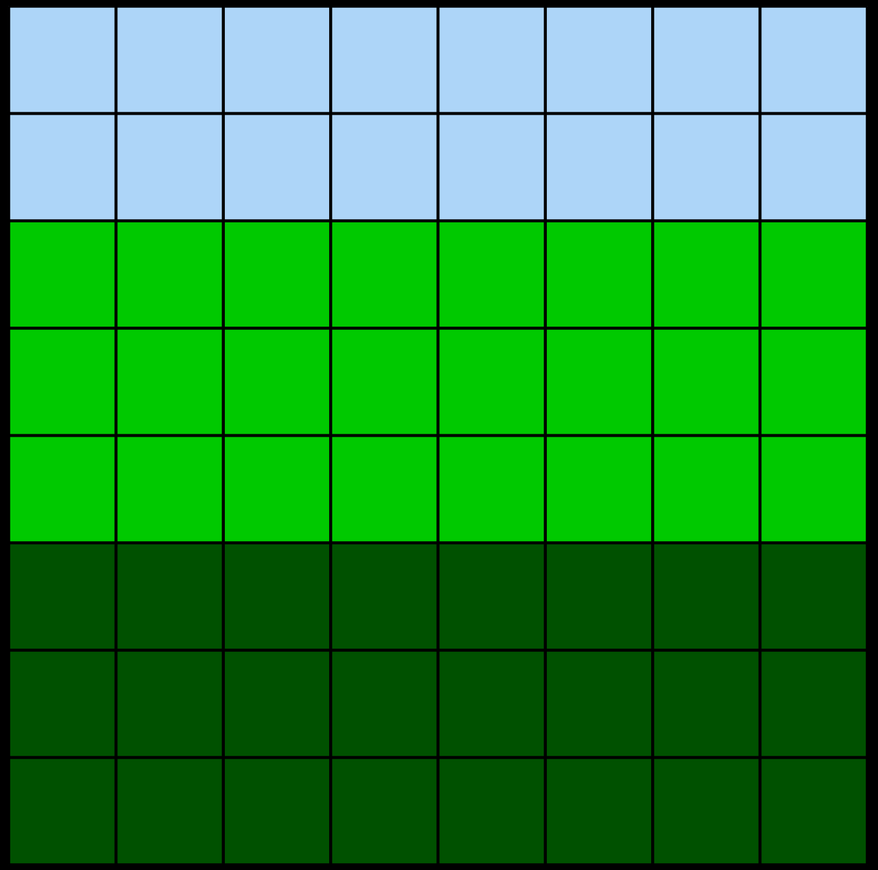


birthday paradox

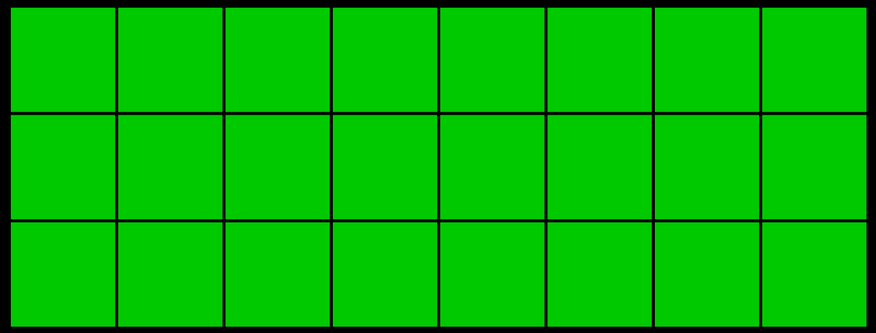


primitive #3: birthday heapspray

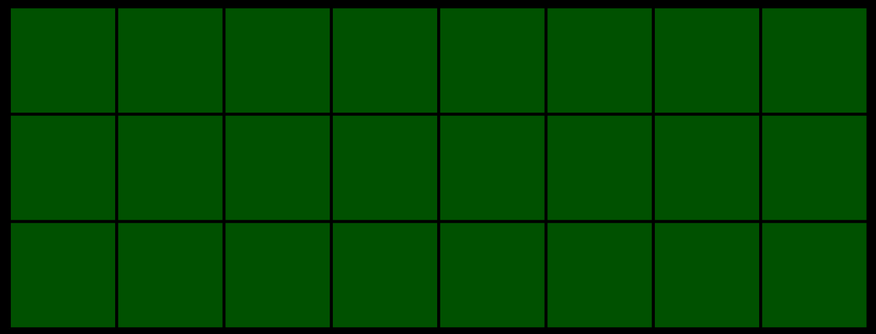
physical memory



attacker memory

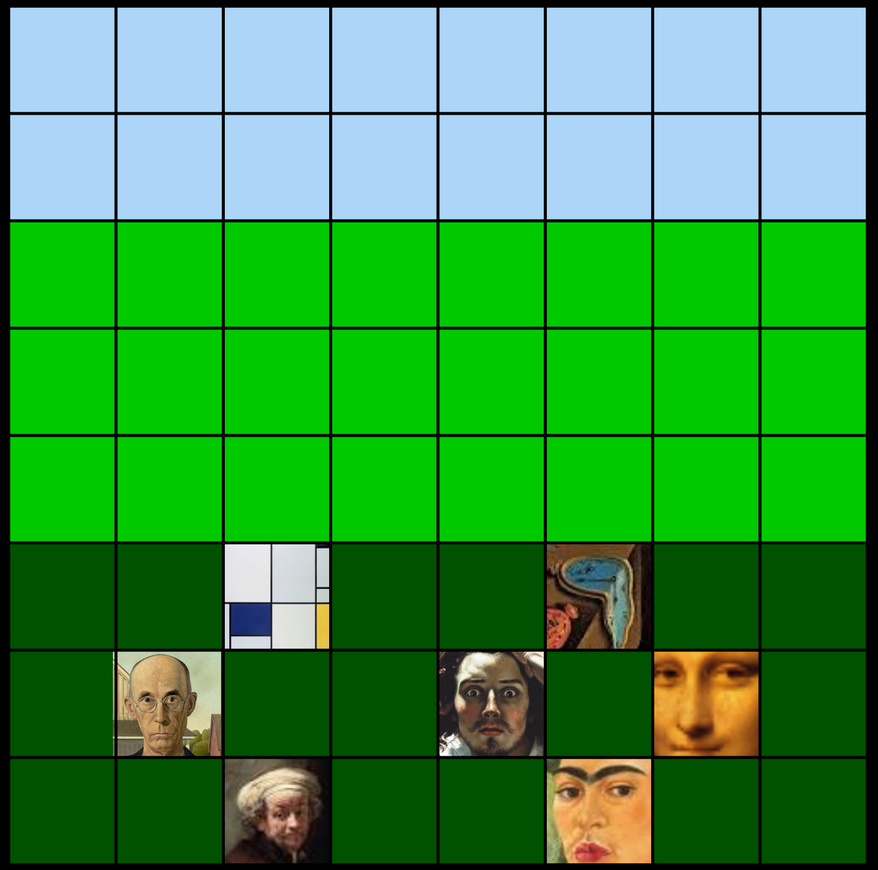


victim memory

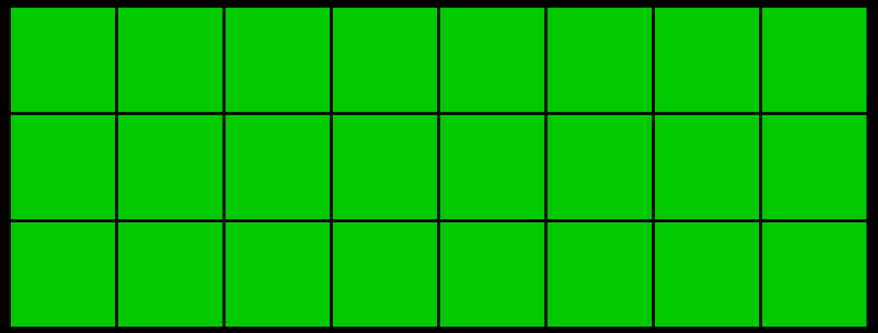


primitive #3: birthday heapspray

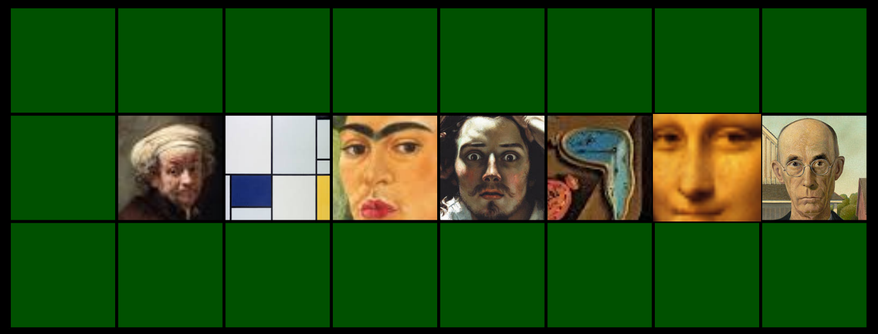
physical memory



attacker memory

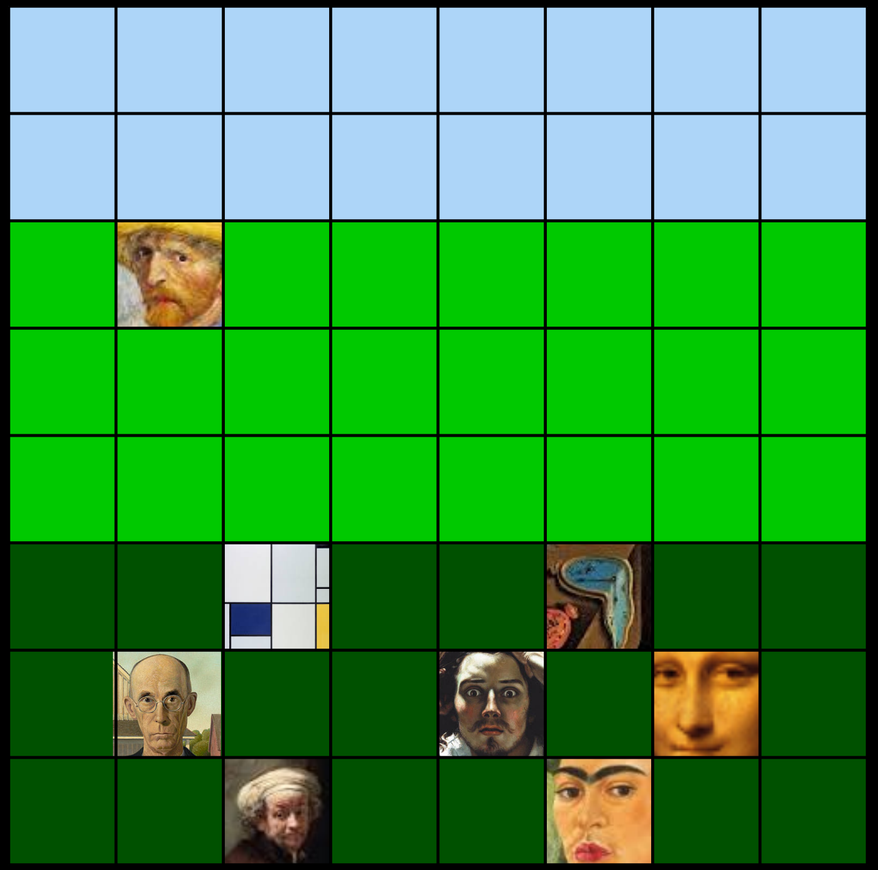


victim memory

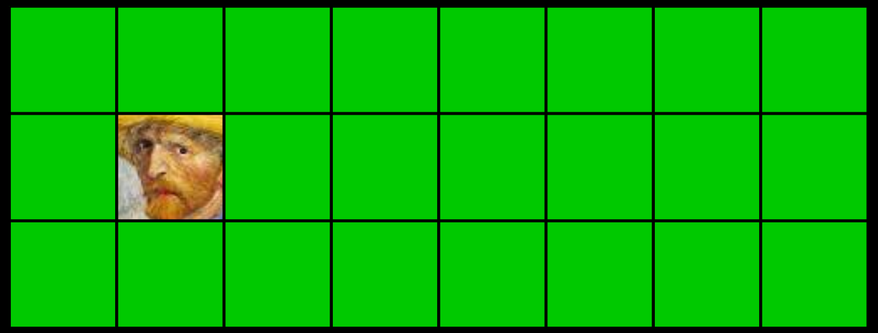


primitive #3: birthday heapspray

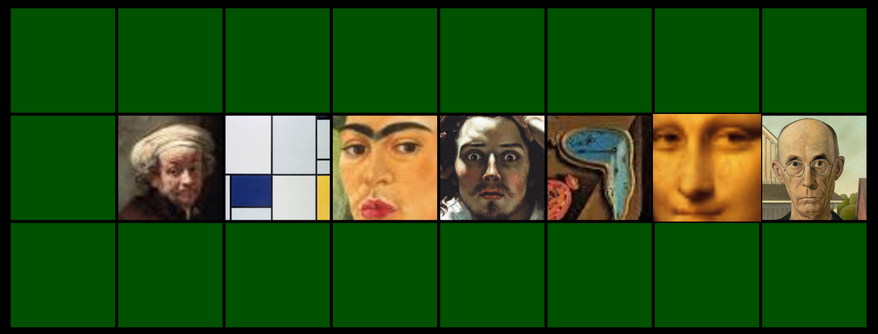
physical memory



attacker memory

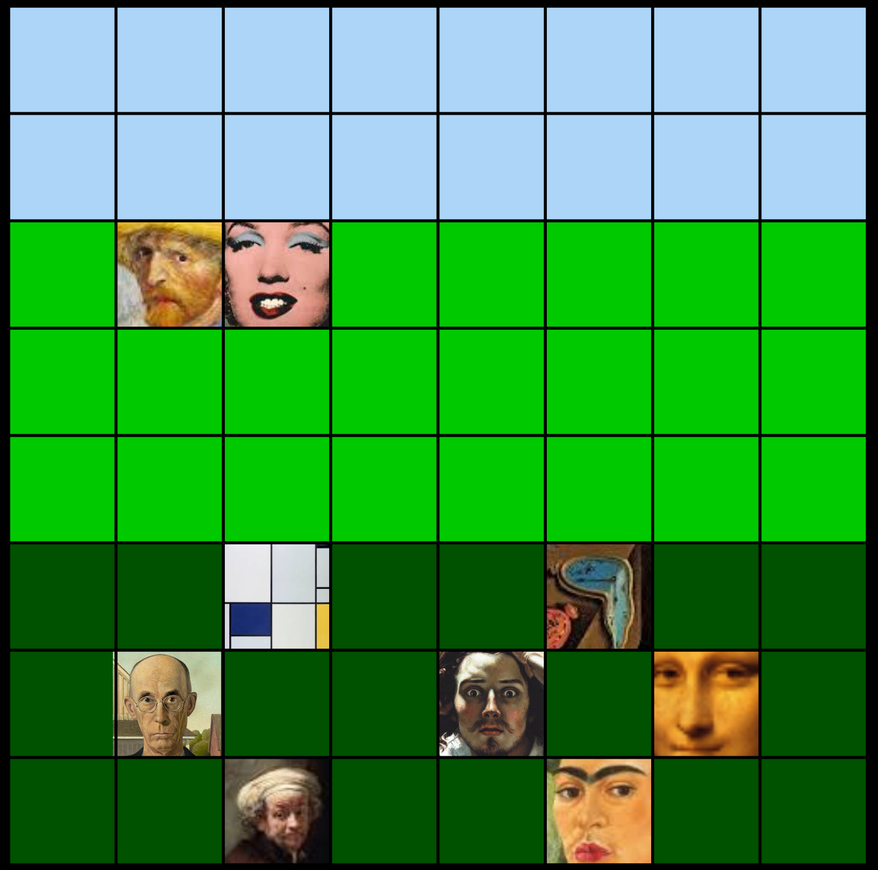


victim memory

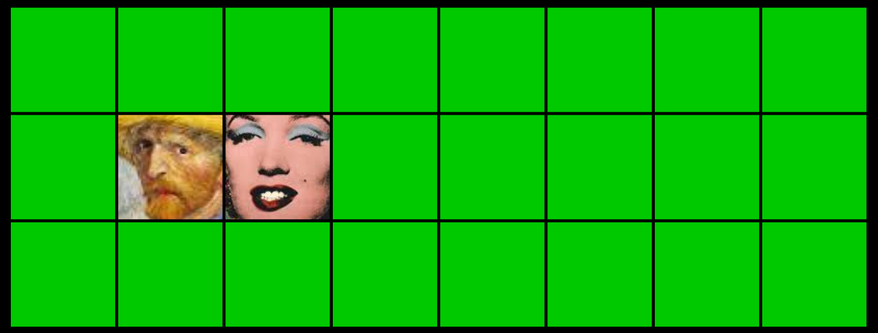


primitive #3: birthday heapspray

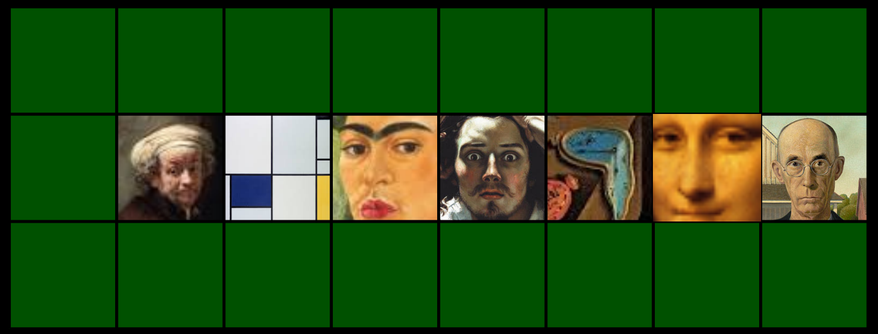
physical memory



attacker memory

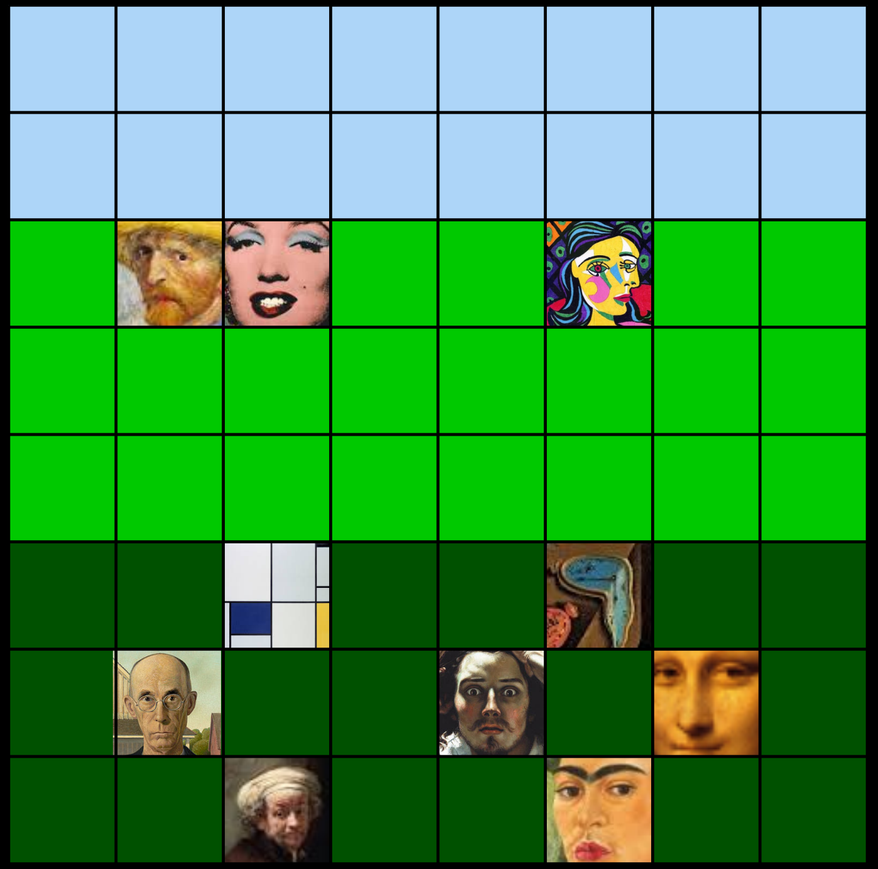


victim memory

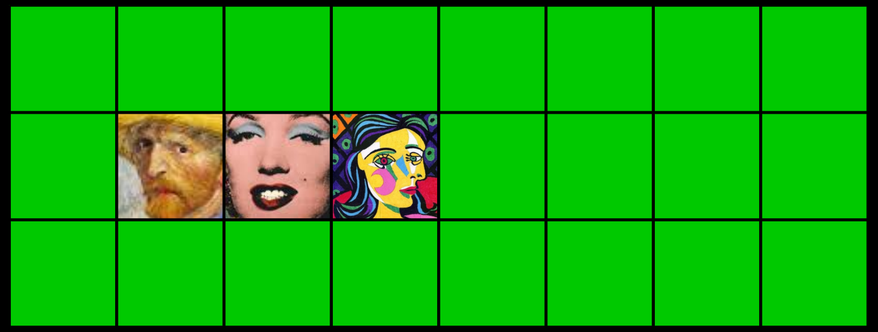


primitive #3: birthday heapspray

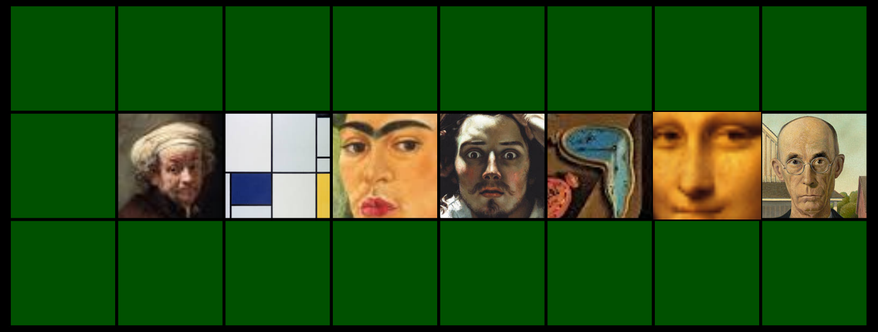
physical memory



attacker memory

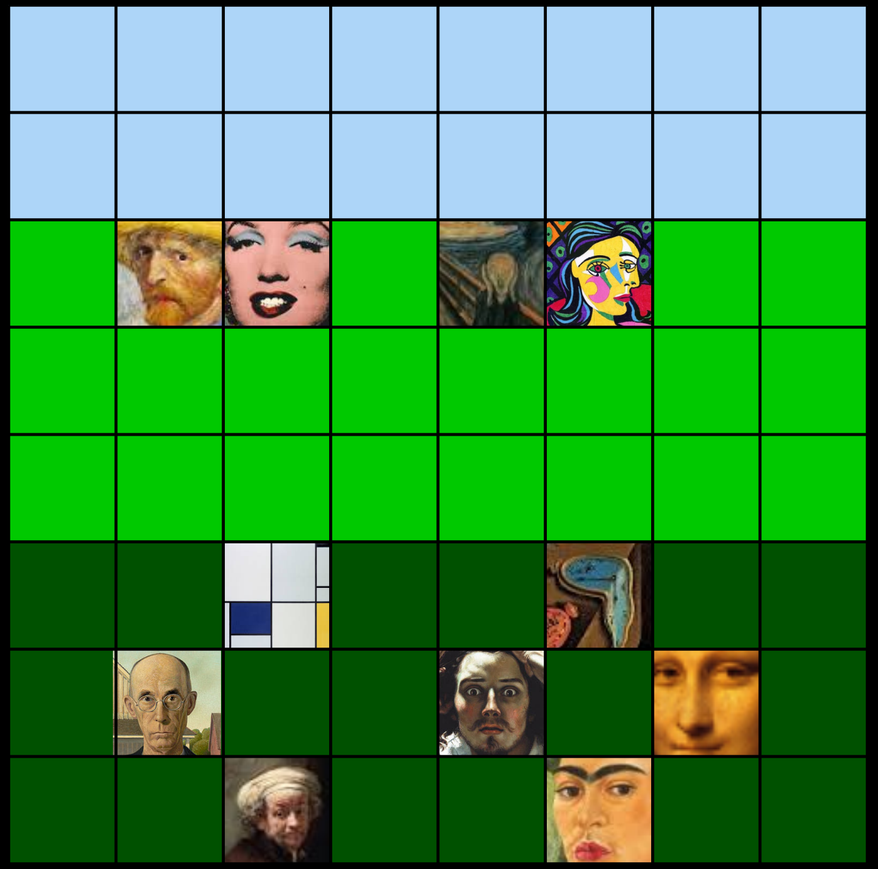


victim memory

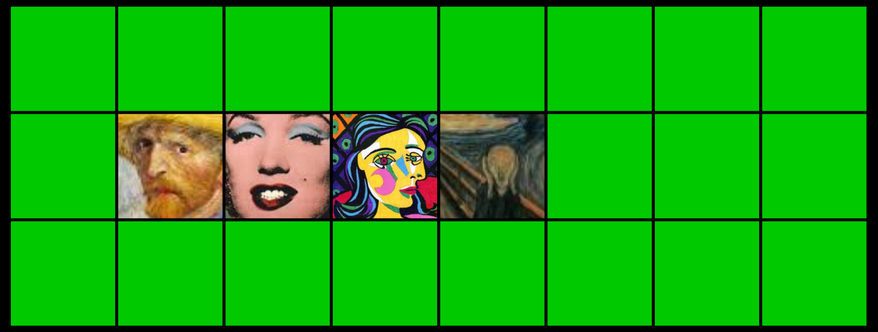


primitive #3: birthday heapspray

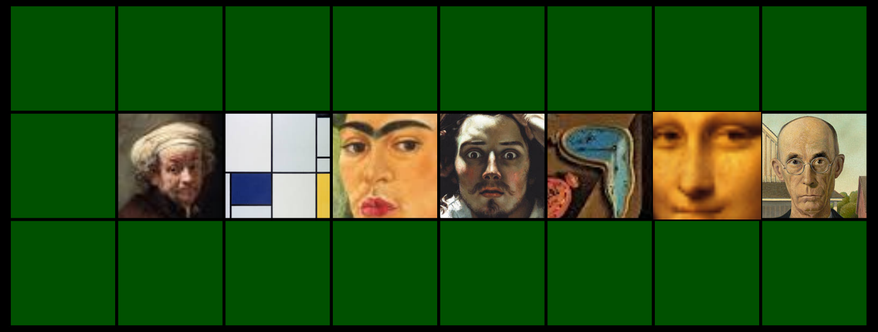
physical memory



attacker memory

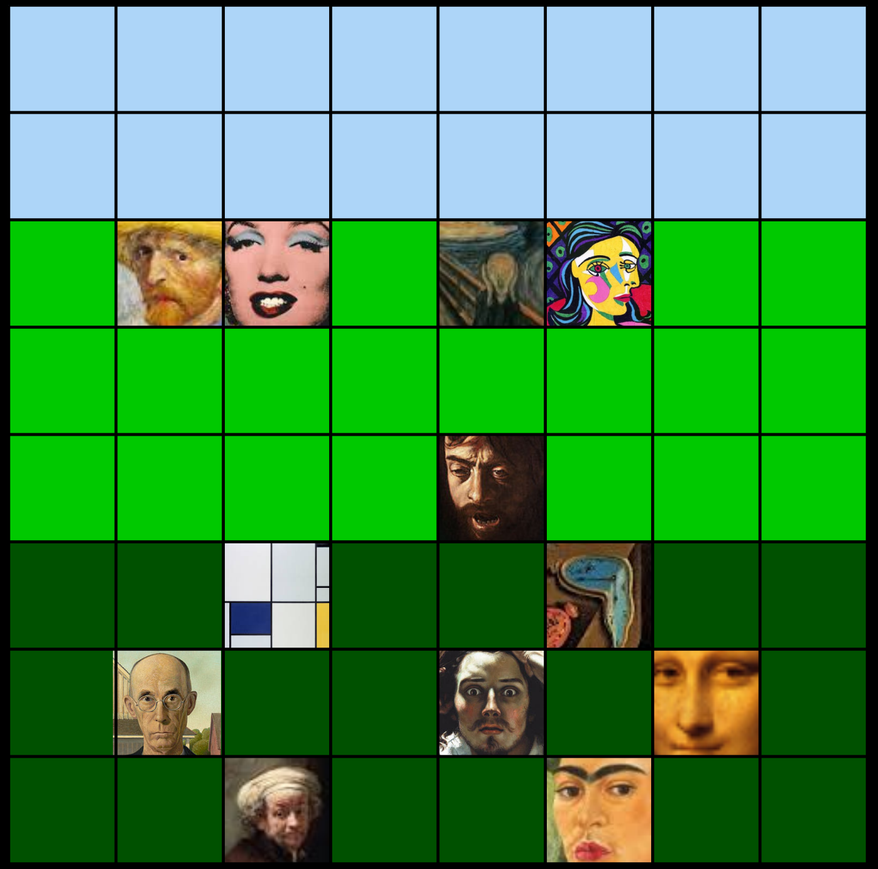


victim memory

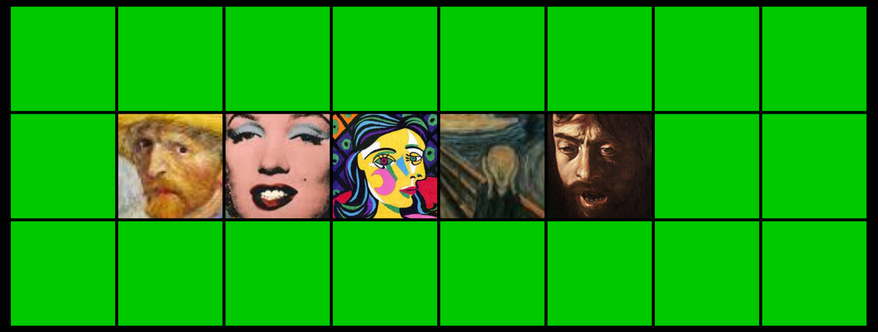


primitive #3: birthday heapspray

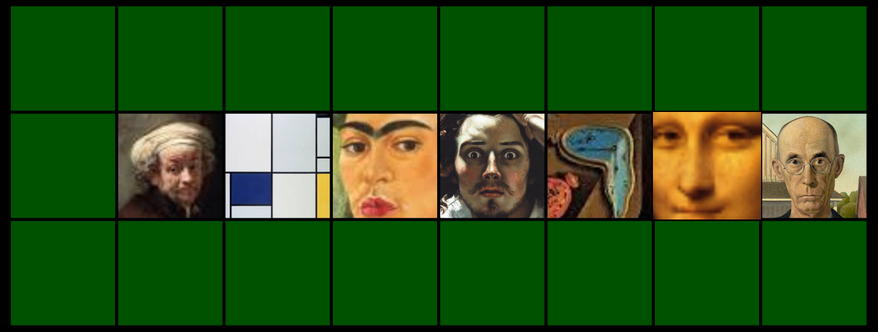
physical memory



attacker memory

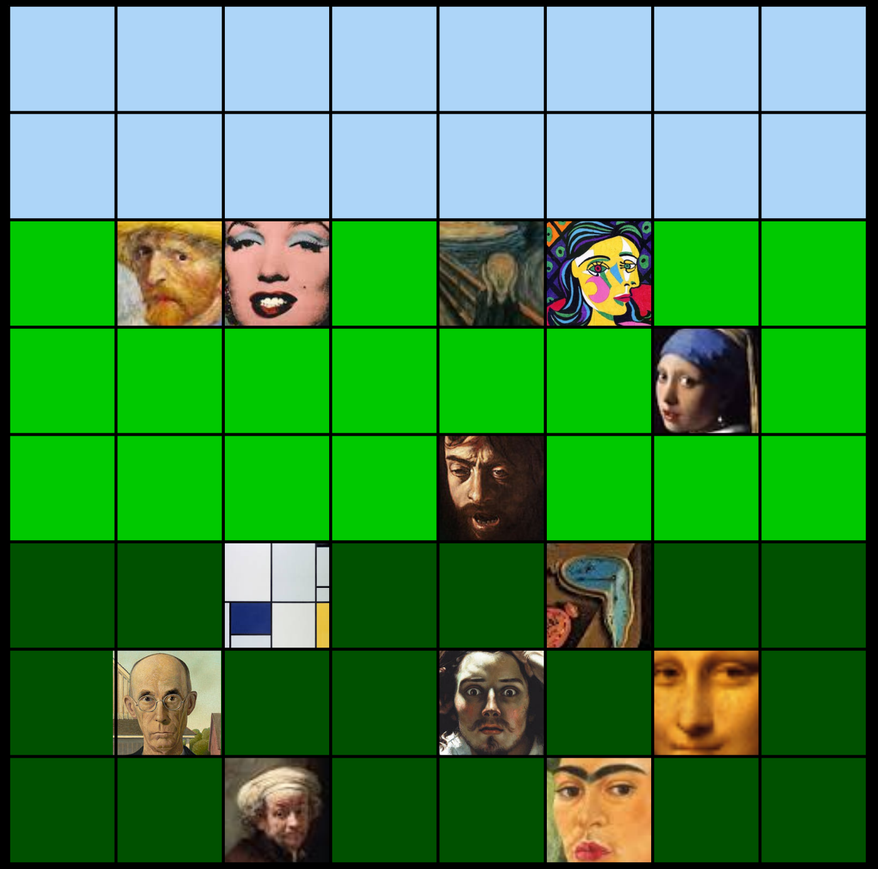


victim memory

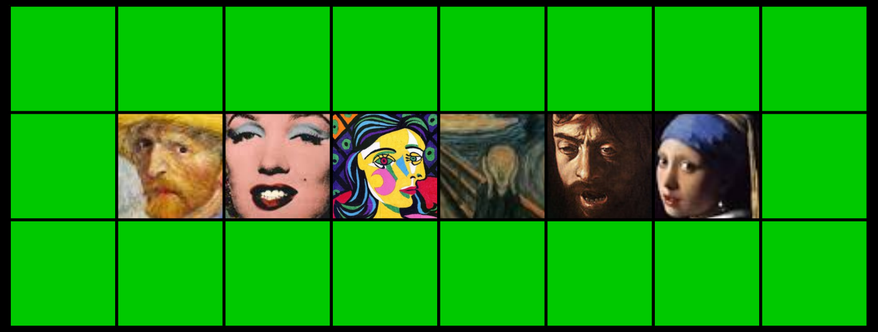


primitive #3: birthday heapspray

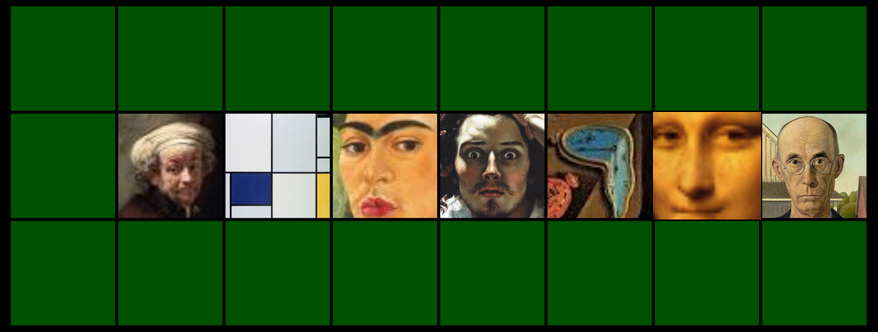
physical memory



attacker memory

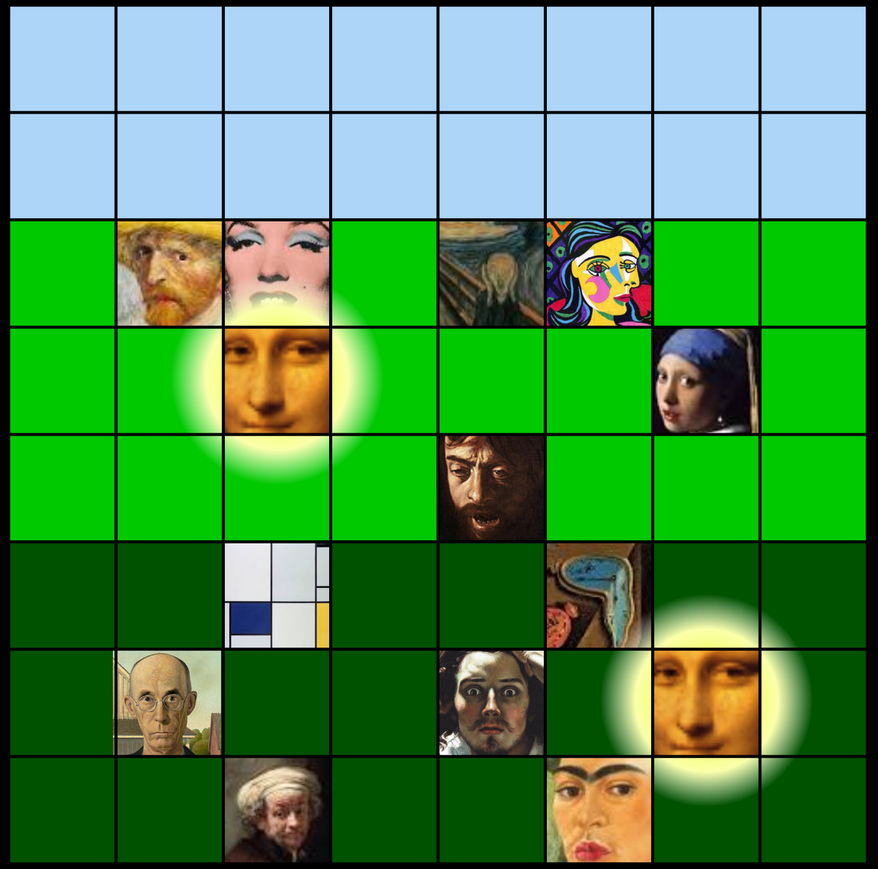


victim memory

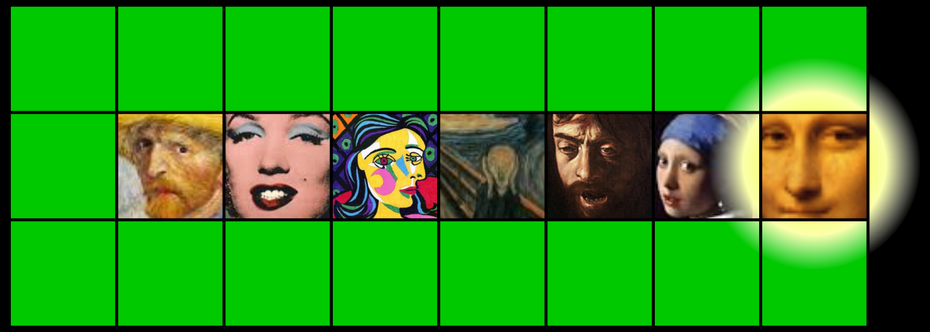


primitive #3: birthday heapspray

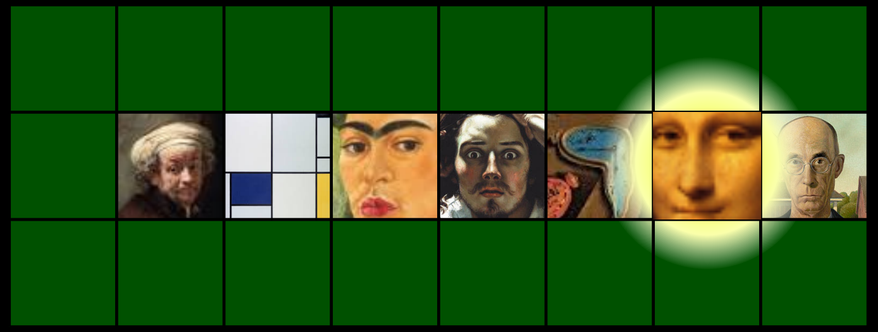
physical memory



attacker memory



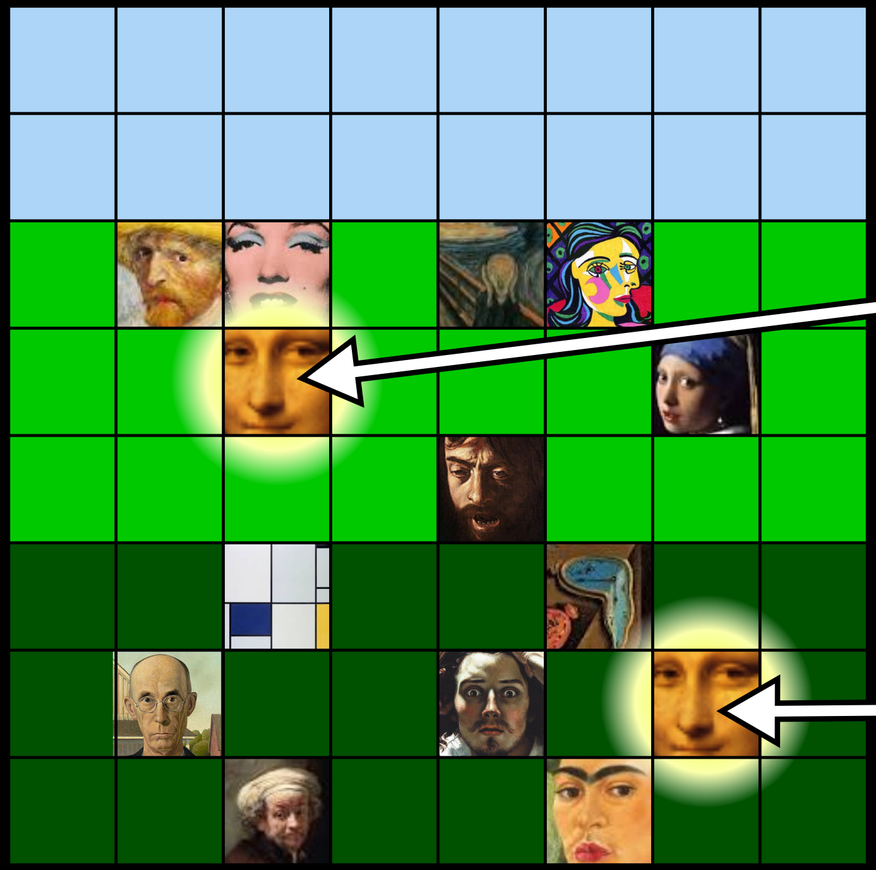
victim memory



primitive #3: birthday heapspray

physical memory

attacker memory



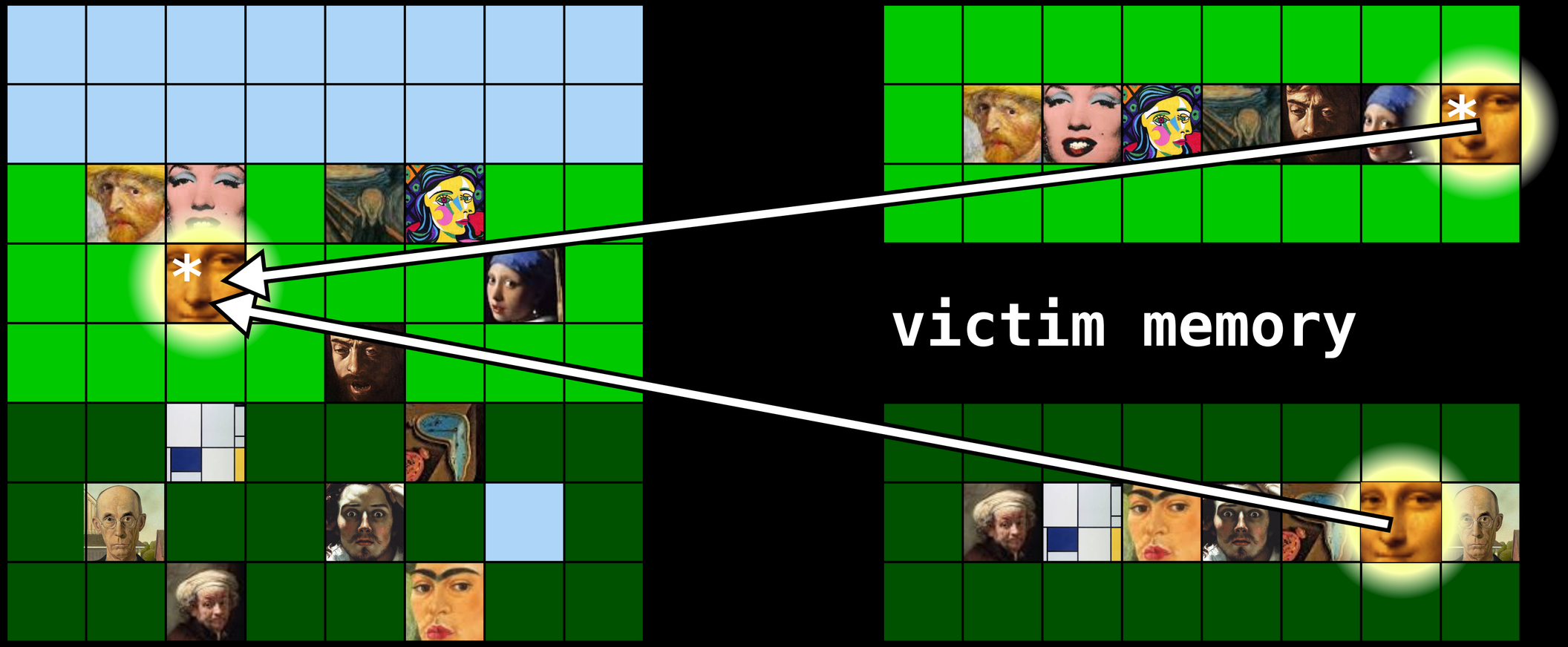
victim memory

primitive #3: birthday heapspray

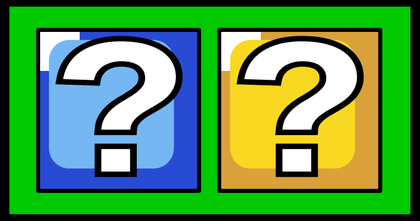
physical memory

attacker memory

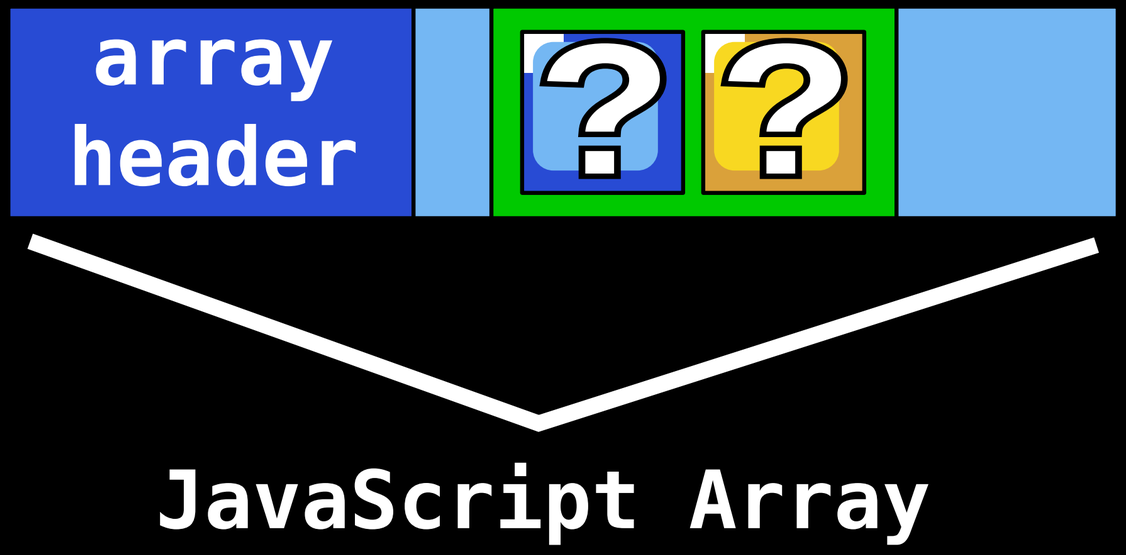
victim memory



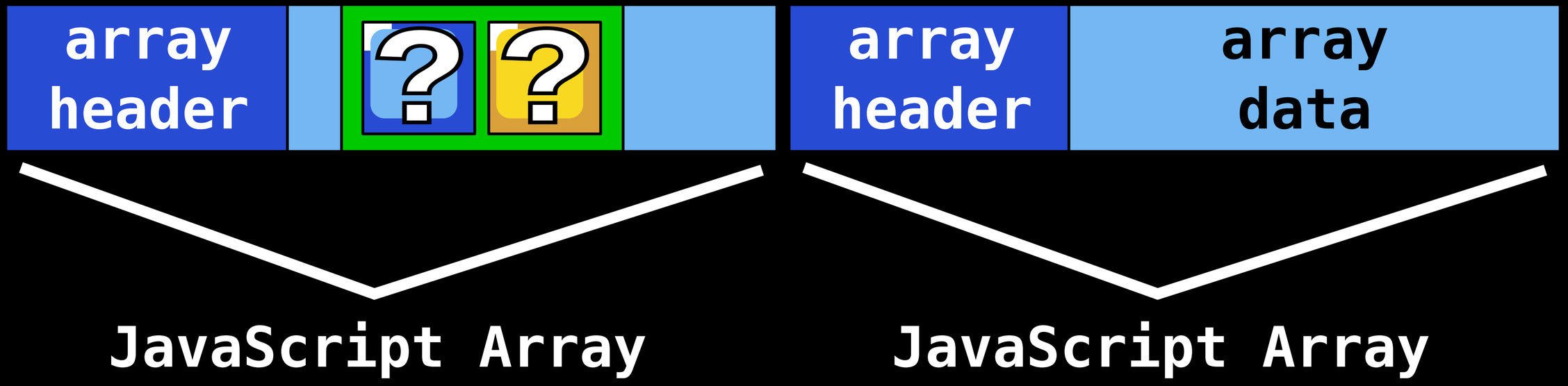
pointer pivotting



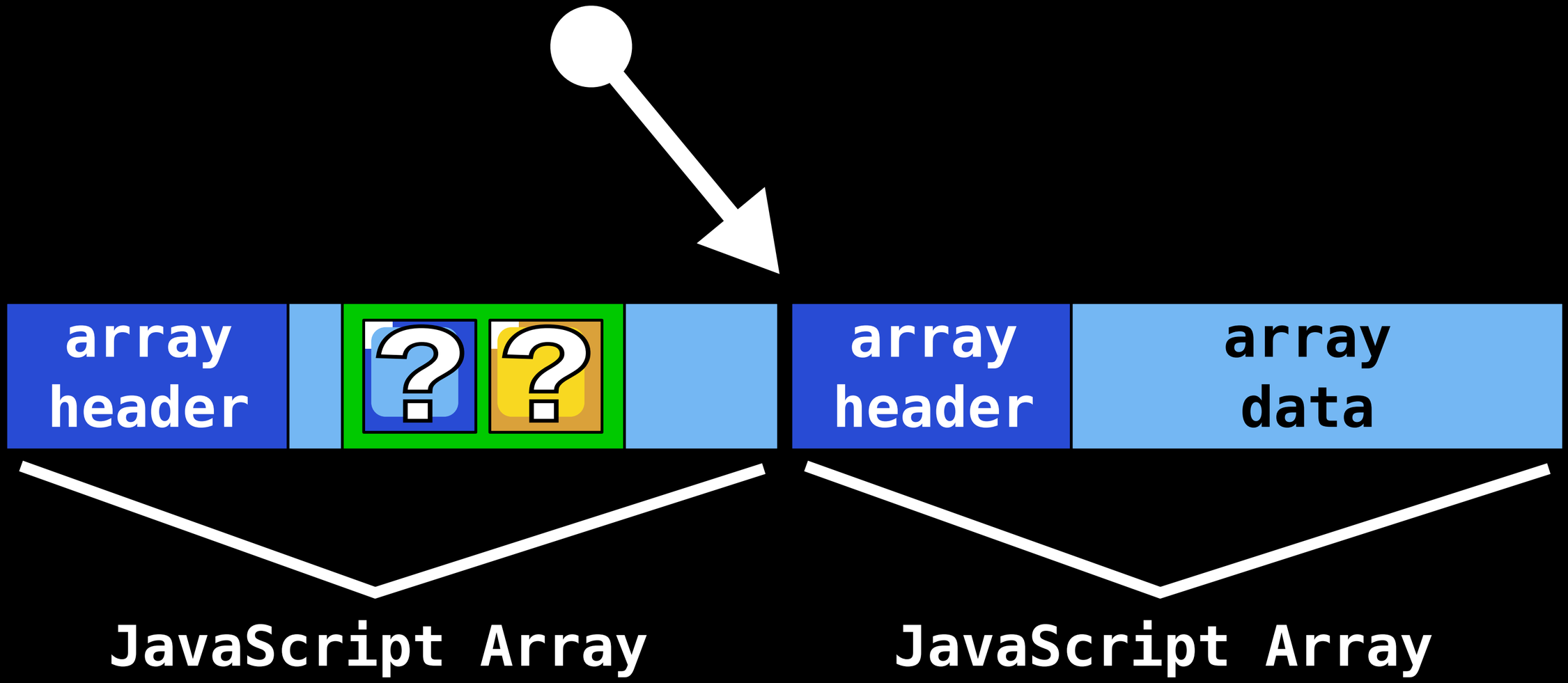
pointer pivotting



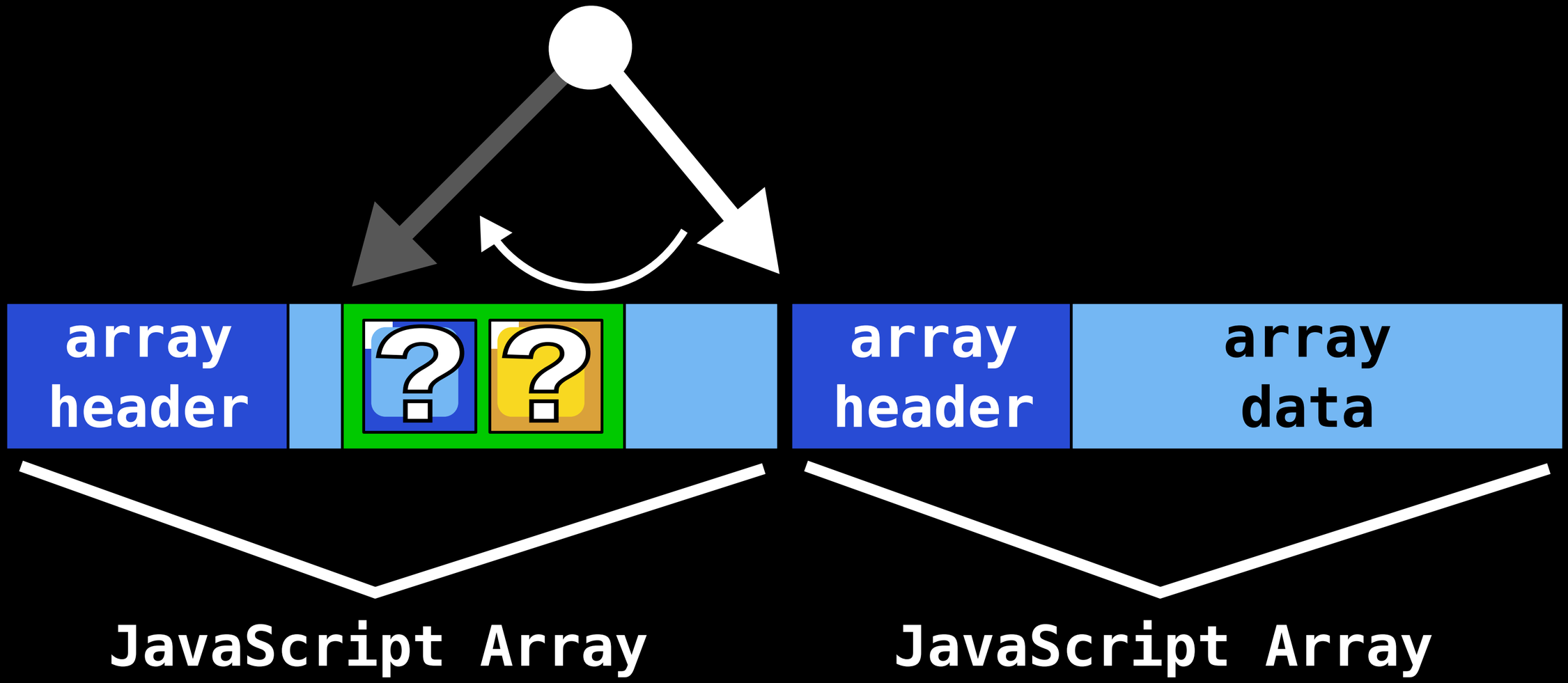
pointer pivotting



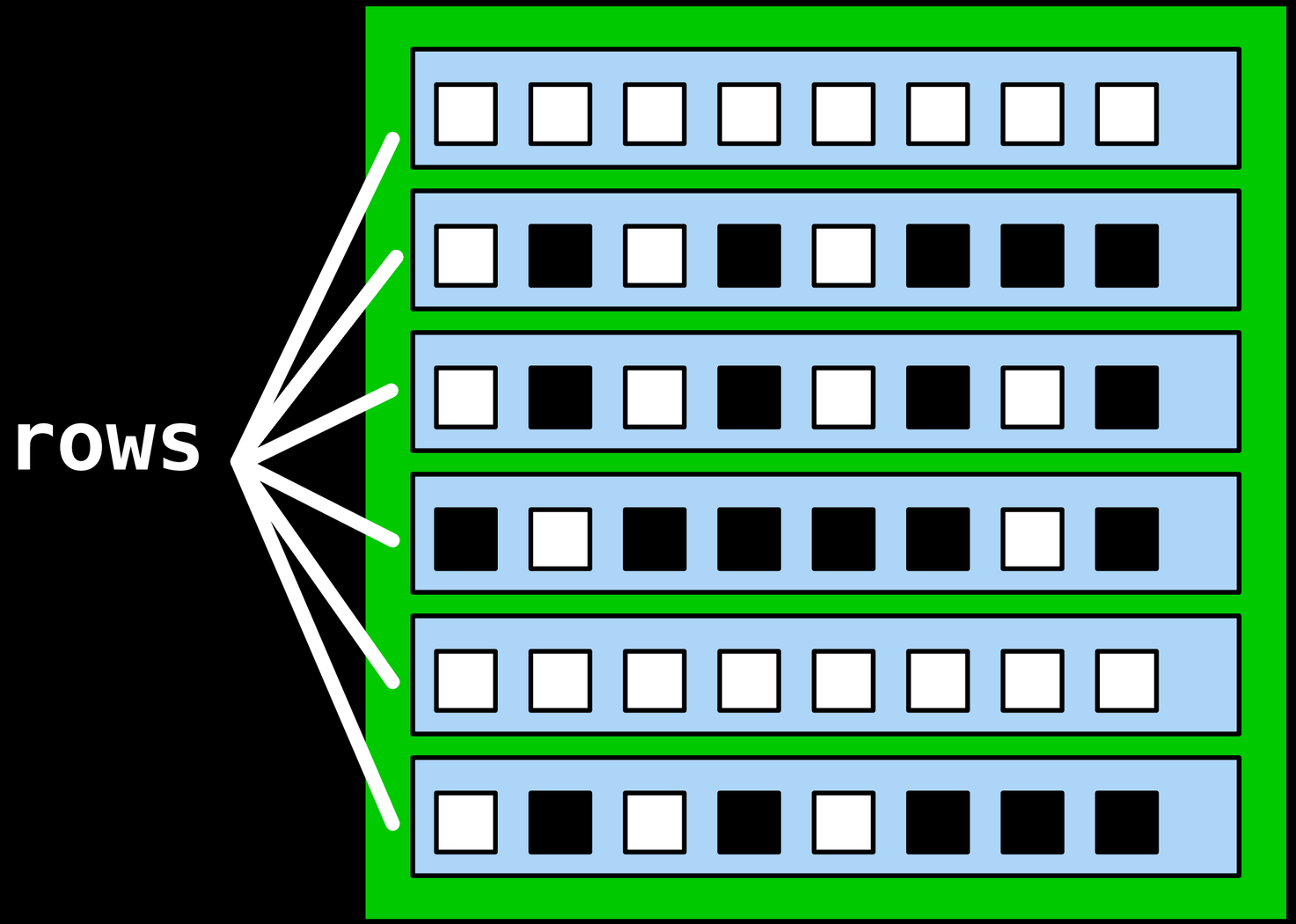
pointer pivotting



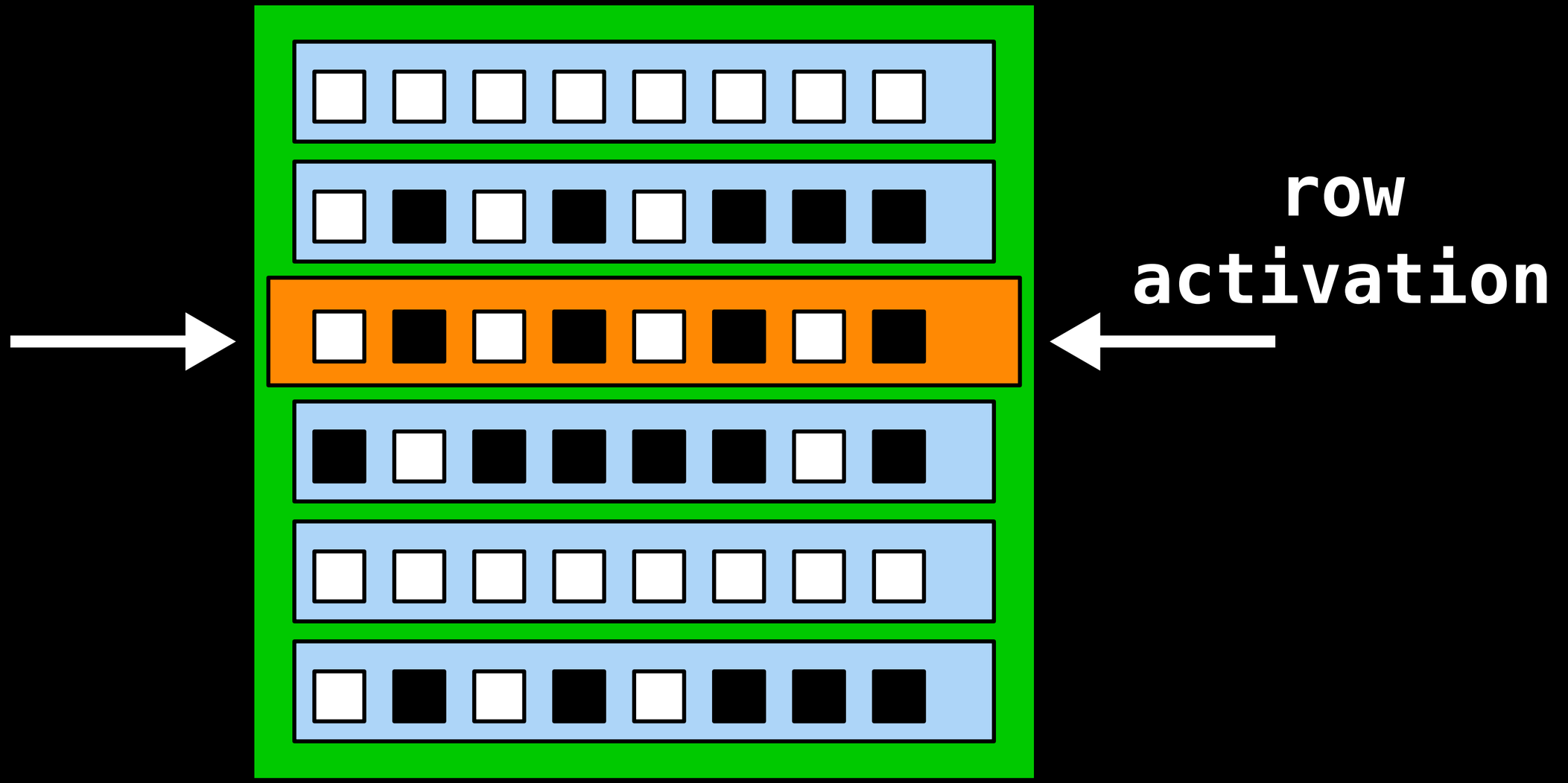
pointer pivotting



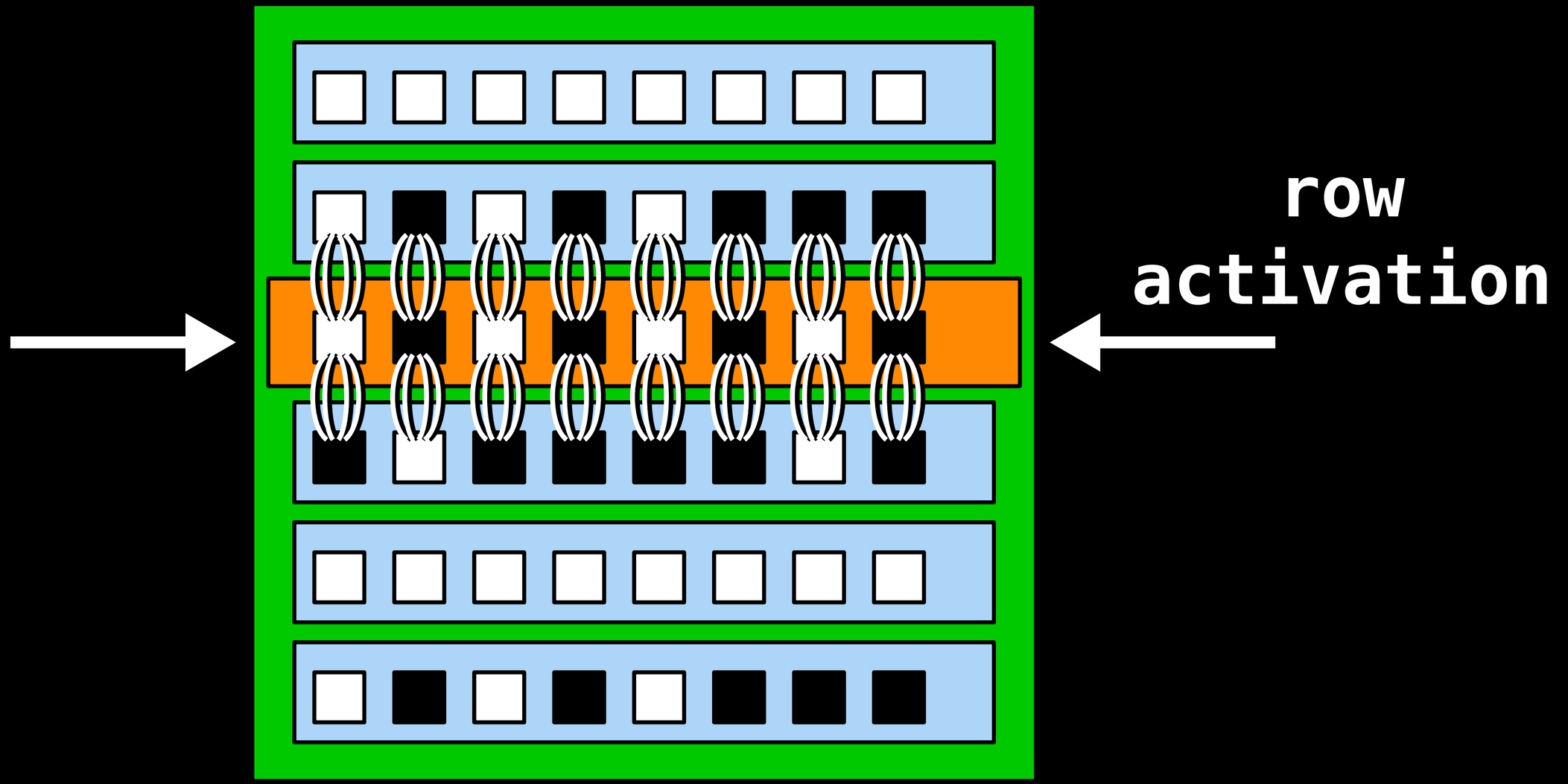
rowhammer attack



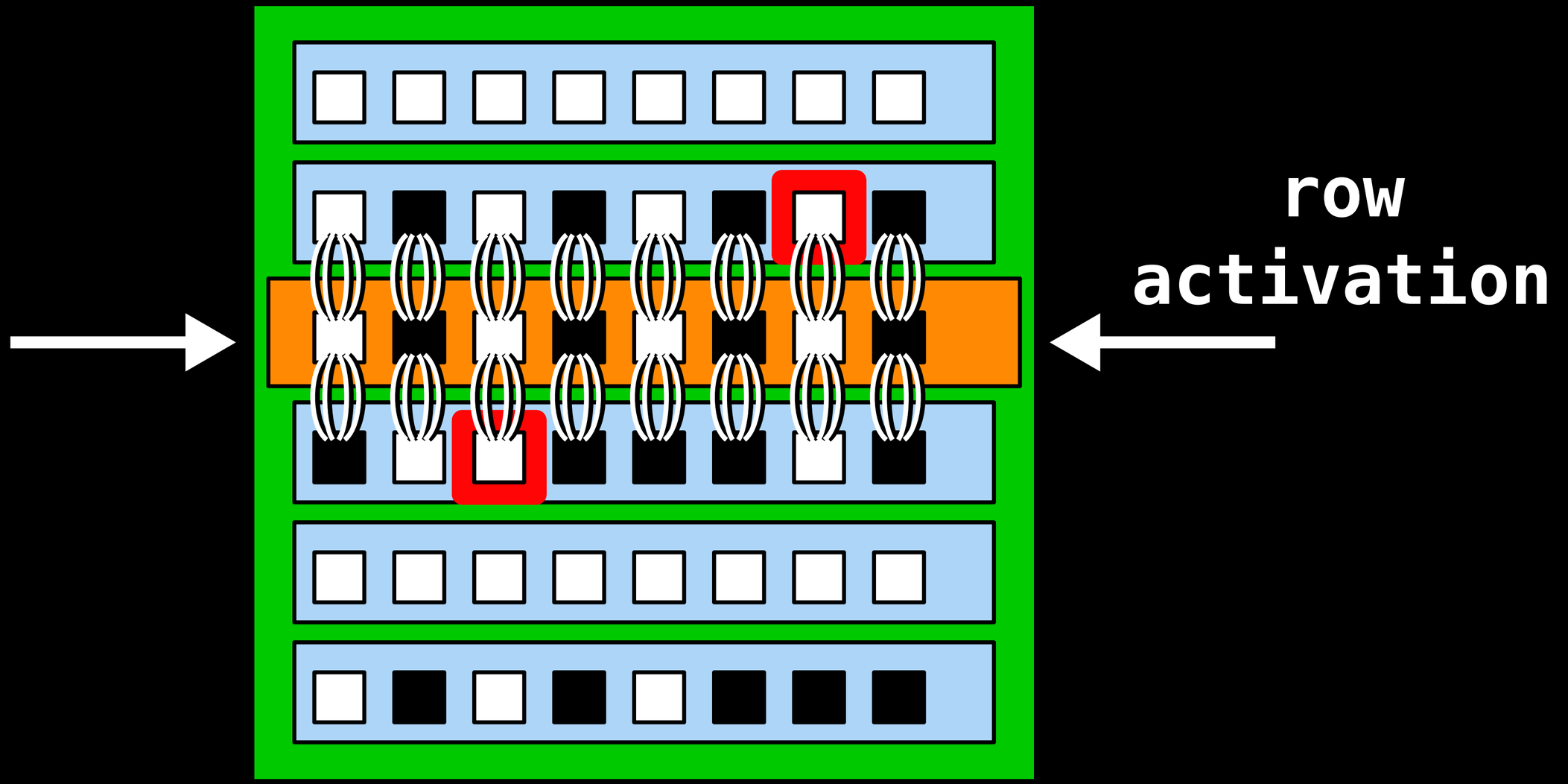
rowhammer attack



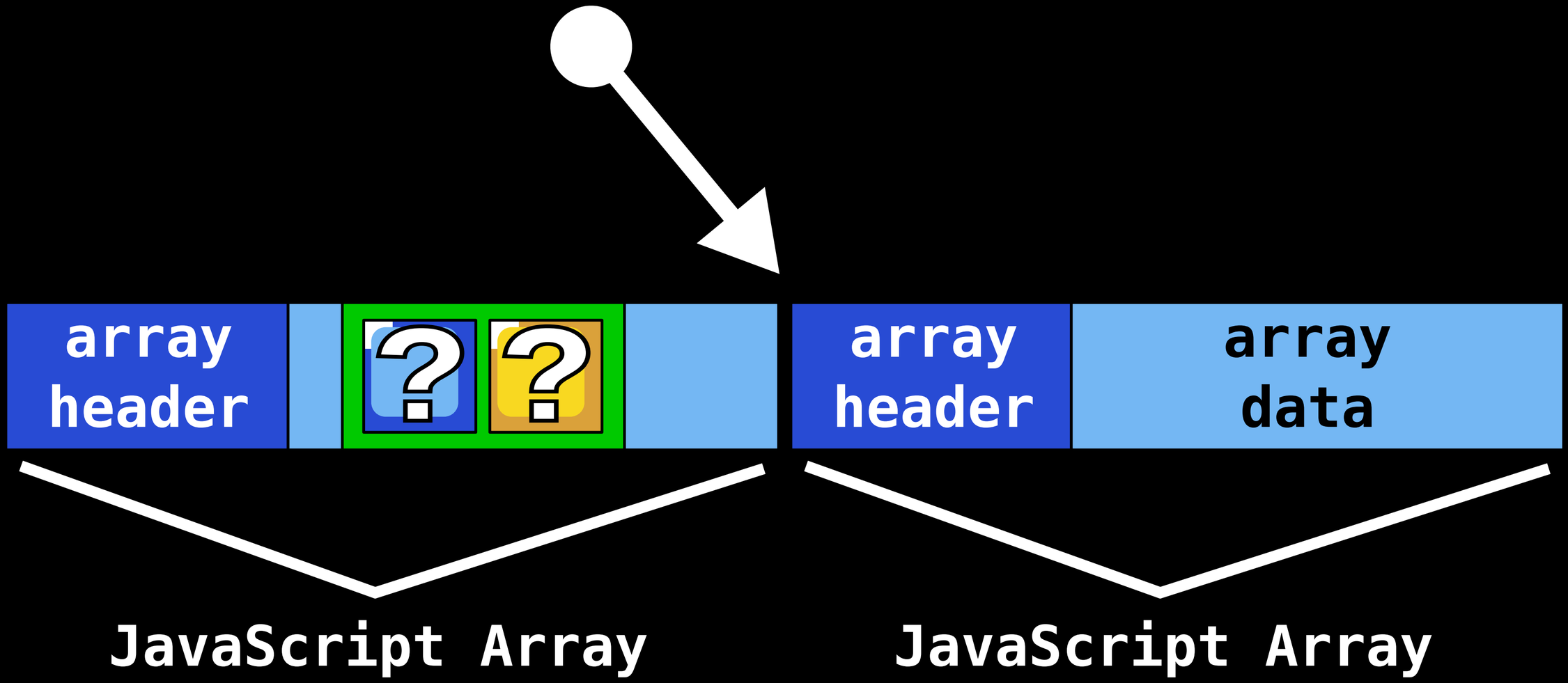
rowhammer attack



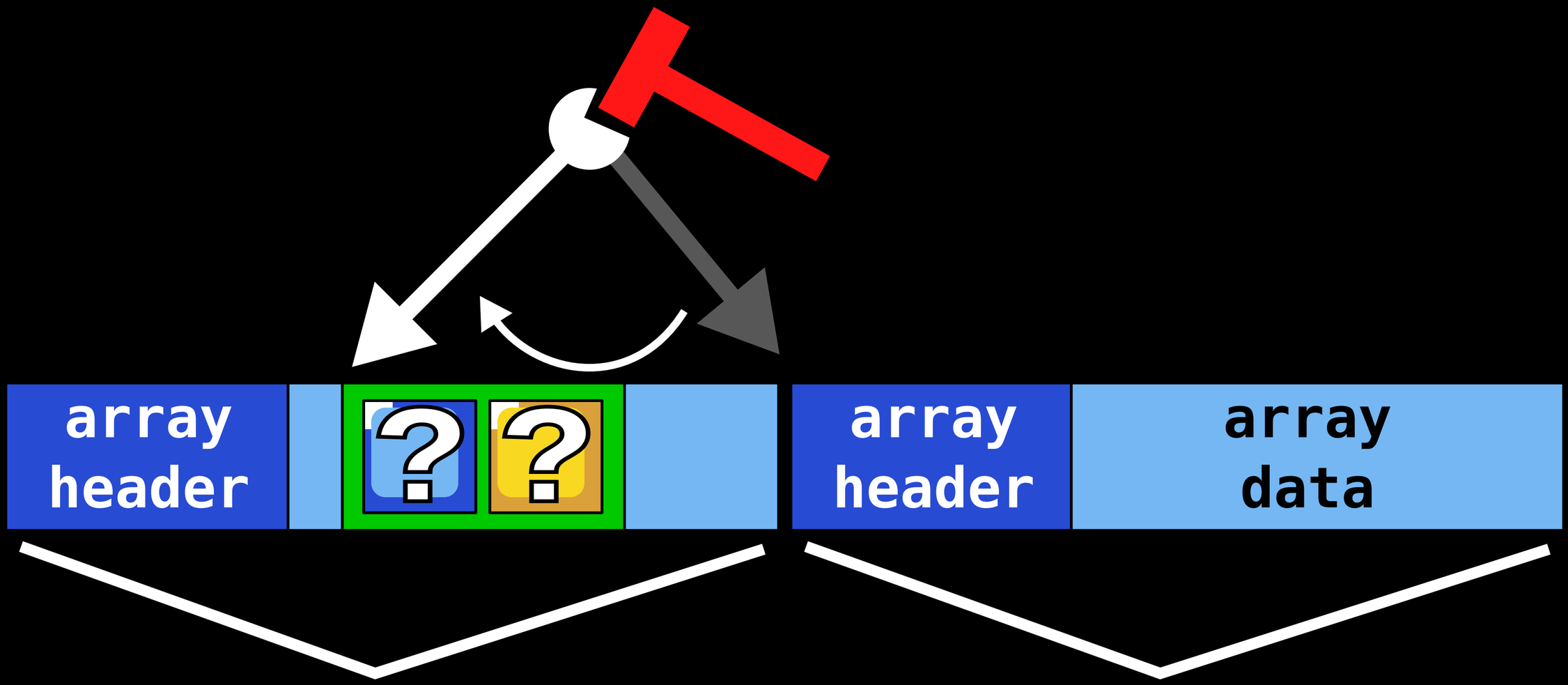
rowhammer attack



pointer pivotting



pointer pivotting



in short:

- Memory deduplication is a more powerful side-channel than previously thought.

- Reliable browser exploitation using rowhammer in JavaScript is possible.

Any questions?