# Structured Encryption and Dynamic Leakage Suppression

Marilyn George
Brown University

Seny Kamara
Brown University

Tarik Moataz
Aroki Systems

# Data Security and Privacy

https://www.metacompliance.com/blog/

# Data Security and Privacy

Facebook (2021)

533 million records

~$3.7 billion

https://www.metacompliance.com/blog/

# Data Security and Privacy

Facebook (2021)
533 million records
~$3.7 billion

Microsoft (2020)
250 million
~$1.8 billion

https://www.metacompliance.com/blog/

# Data Security and Privacy

Facebook (2021)
533 million records
~$3.7 billion

EasyJet
(2020)
9 million
~50m

Microsoft (2020)
250 million
~$1.8 billion

https://www.metacompliance.com/blog/

# Data Security and Privacy

Facebook (2021)
533 million records
~$3.7 billion

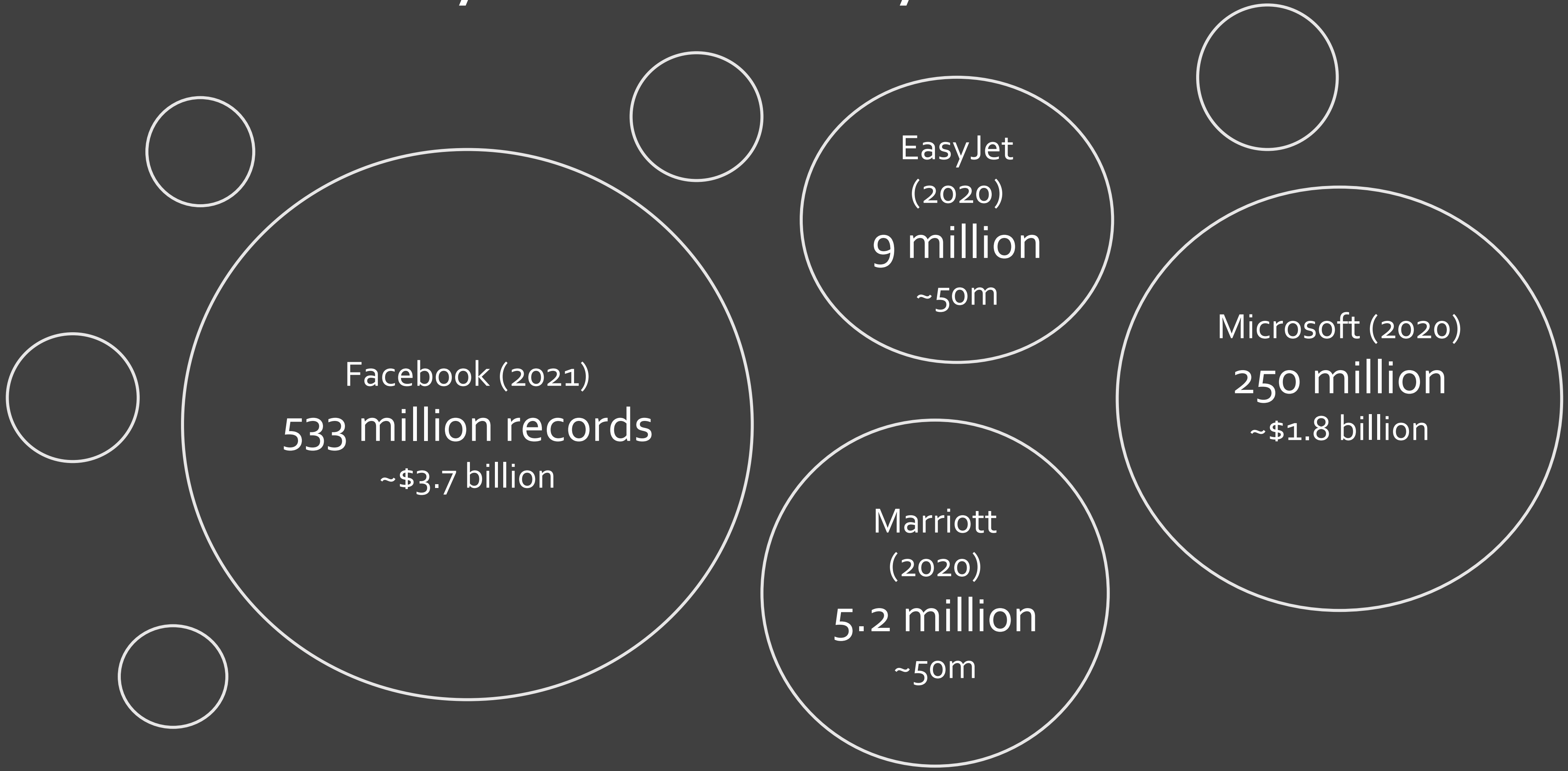
EasyJet
(2020)
9 million
~50m

Marriott
(2020)
5.2 million
~50m

Microsoft (2020)
250 million
~$1.8 billion

https://www.metacompliance.com/blog/

# Data Security and Privacy



Facebook (2021)
533 million records
~$3.7 billion

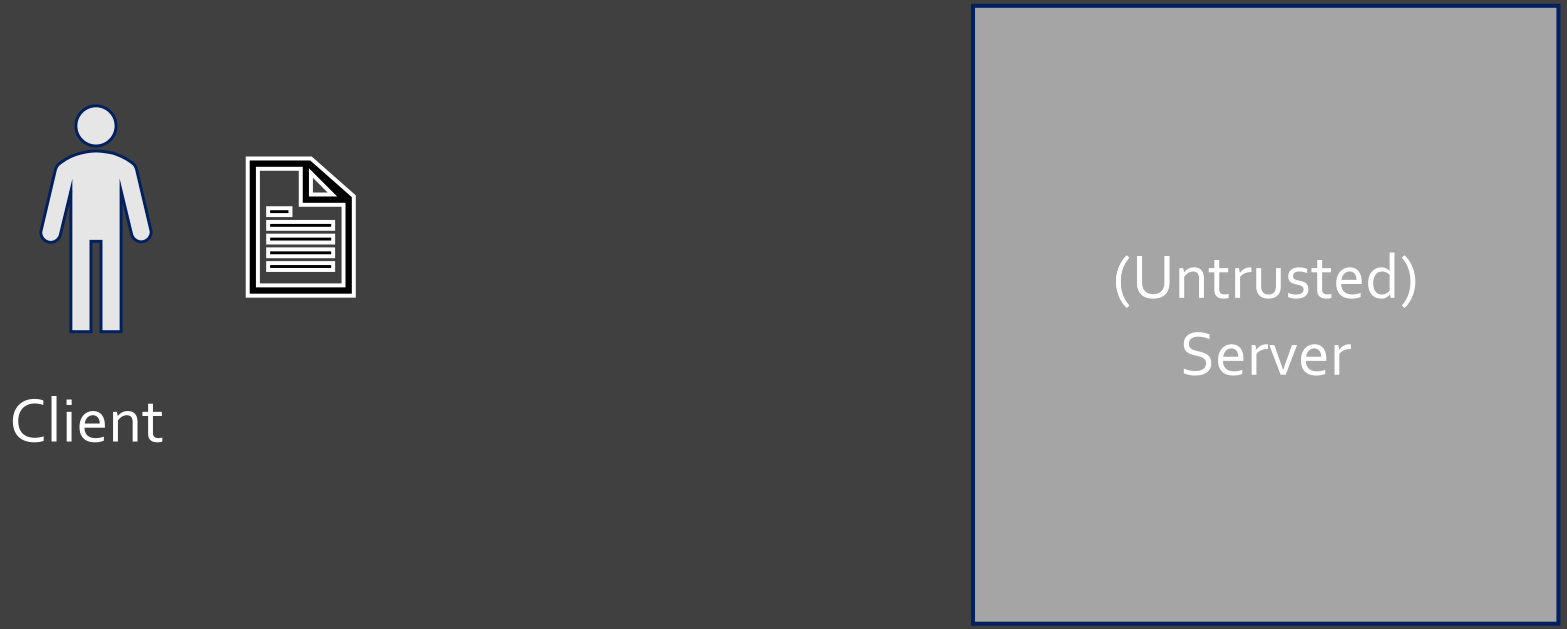
EasyJet
(2020)
9 million
~50m

Marriott
(2020)
5.2 million
~50m

Microsoft (2020)
250 million
~$1.8 billion

https://www.metacompliance.com/blog/

# Encryption



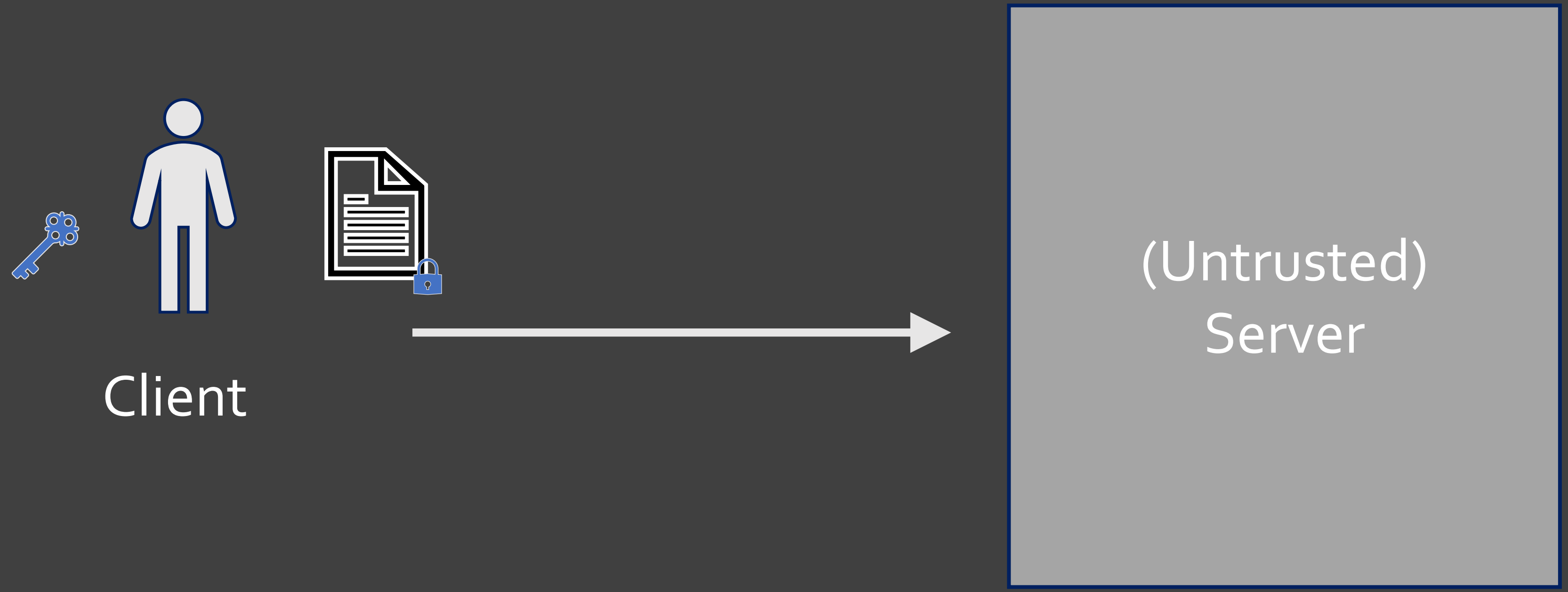
Client

(Untrusted)
Server

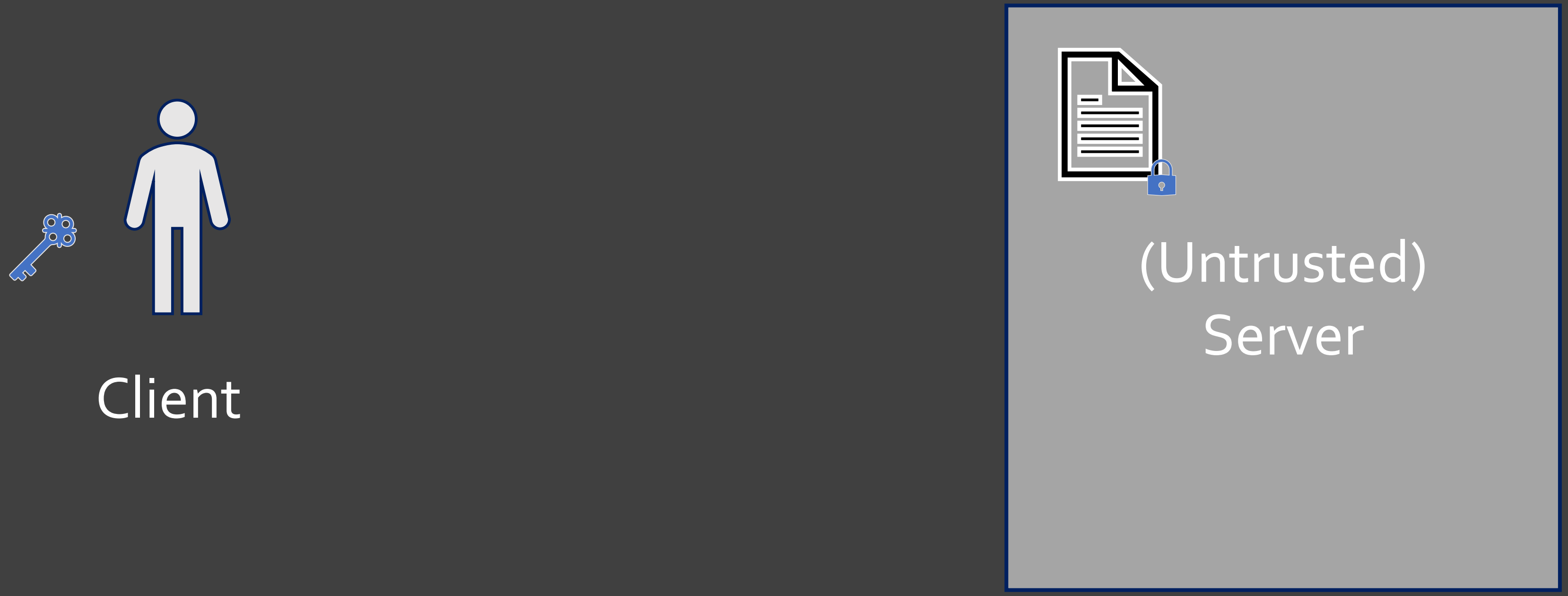
# Encryption

Client
(Untrusted)
Server

# Encryption



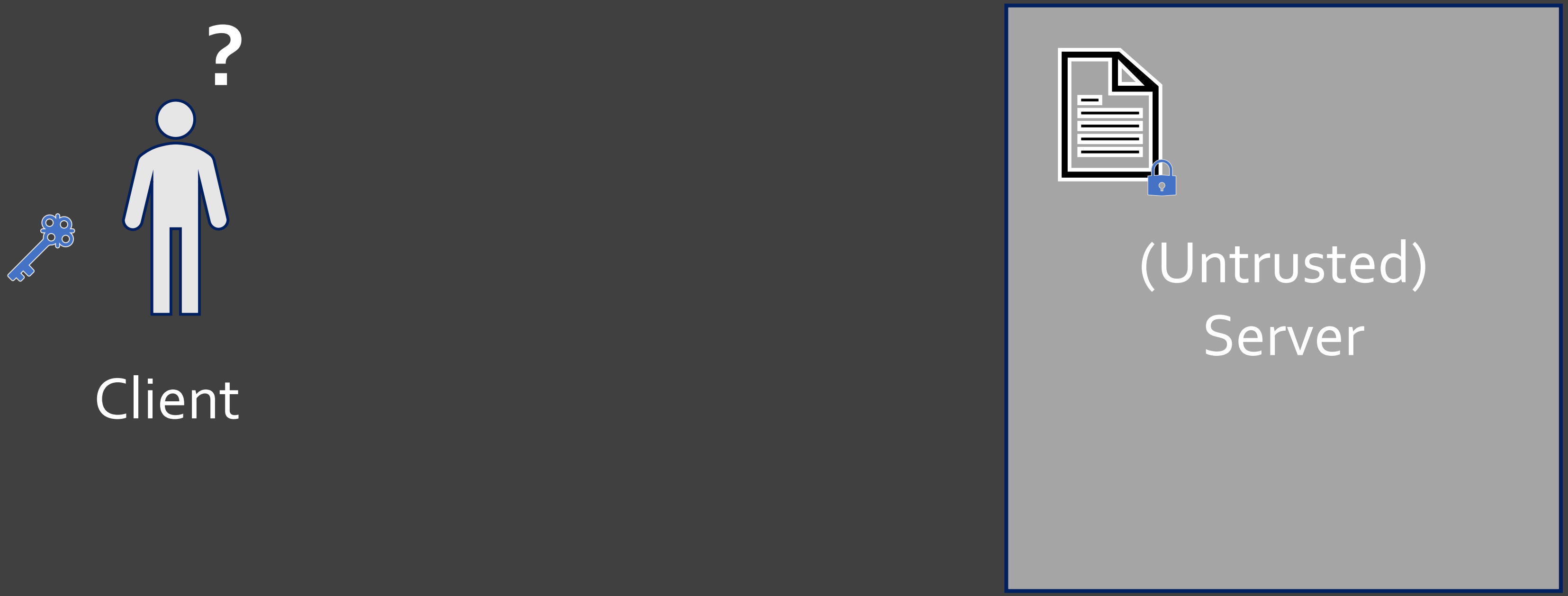
# Encrypted Search



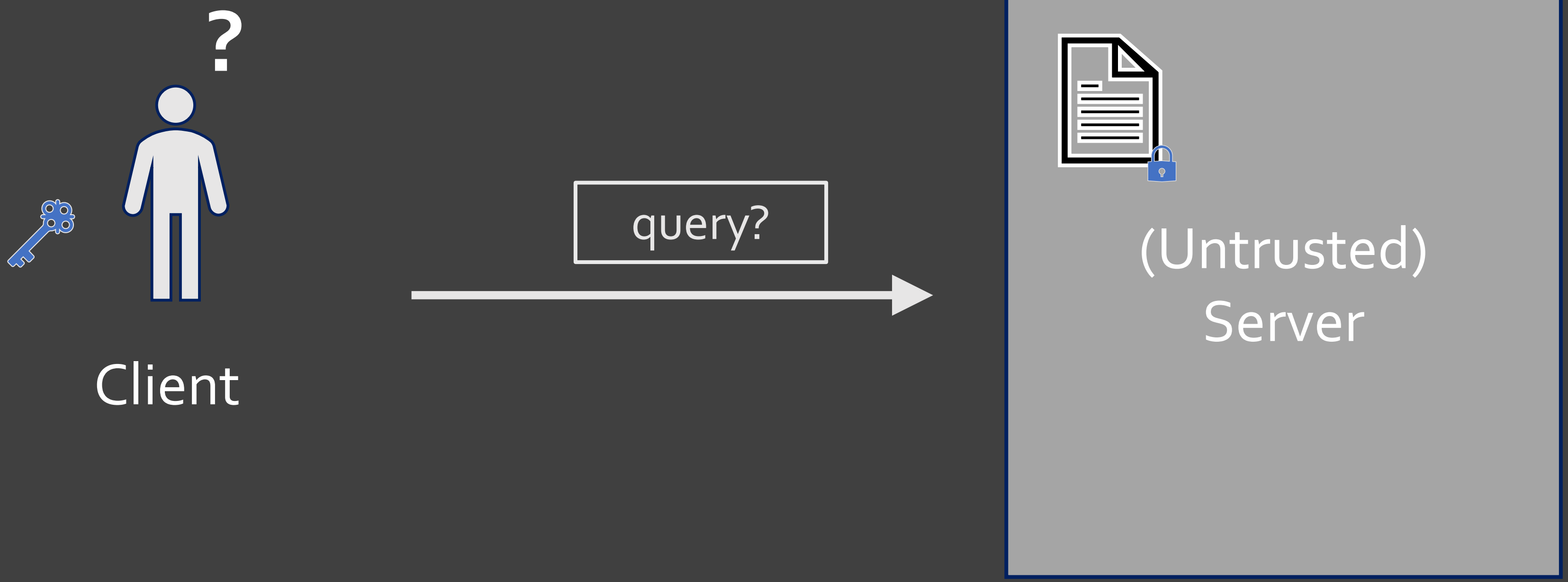
Client

(Untrusted) Server

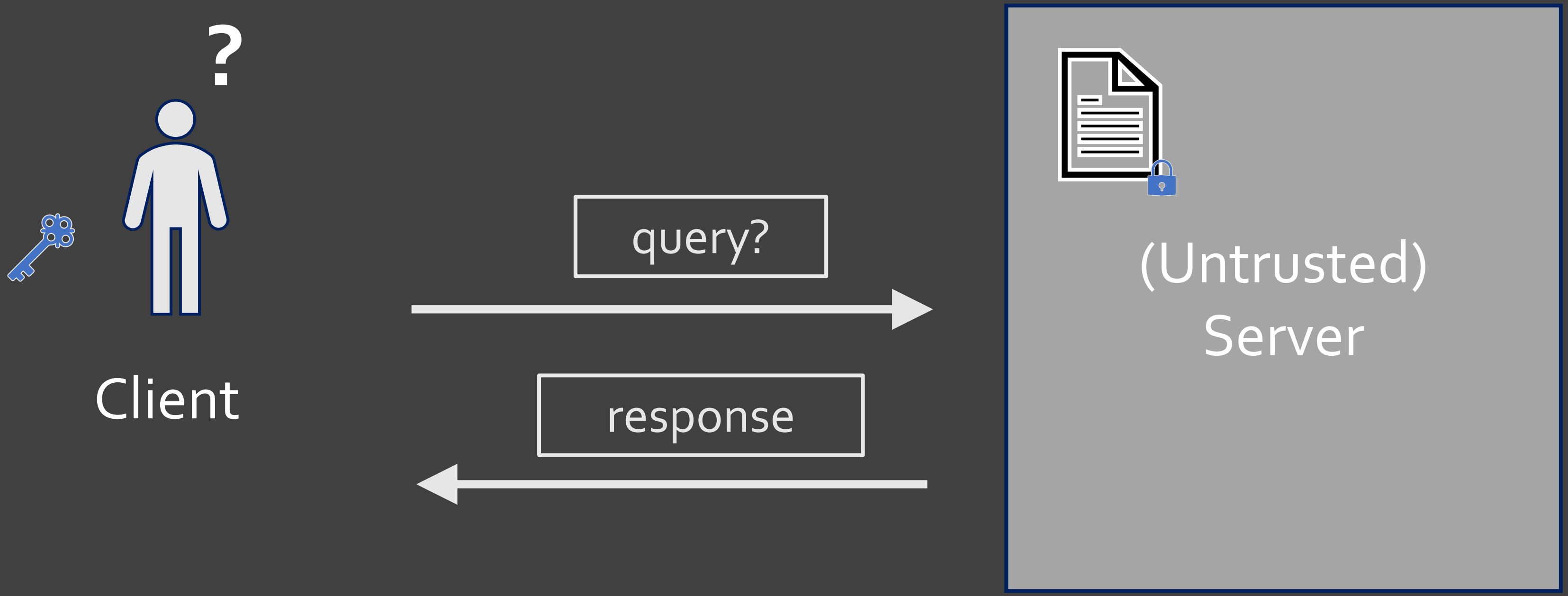
# Encrypted Search



# Encrypted Search



# Encrypted Search



?

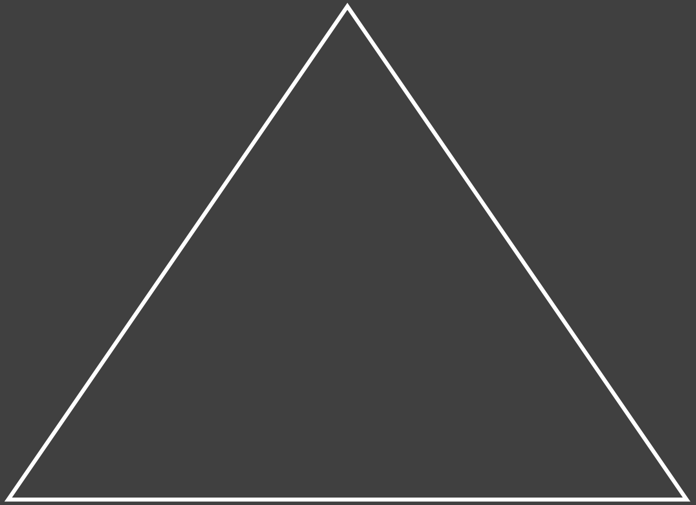
Client

query?

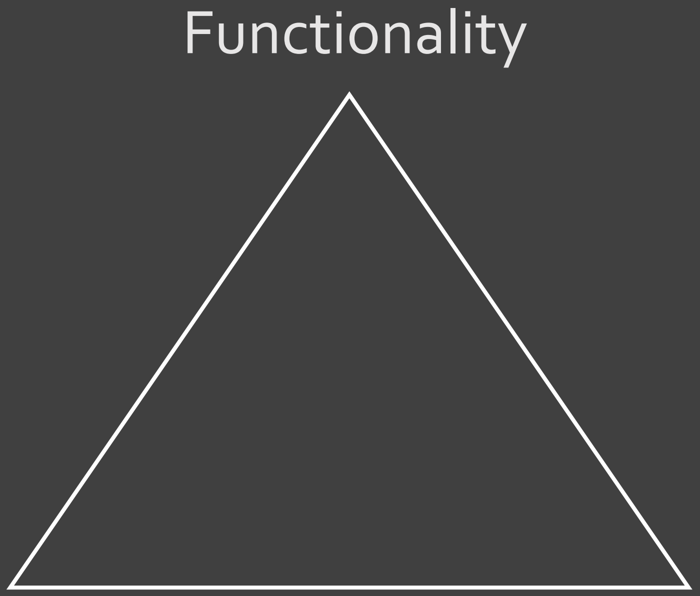
response

(Untrusted) Server

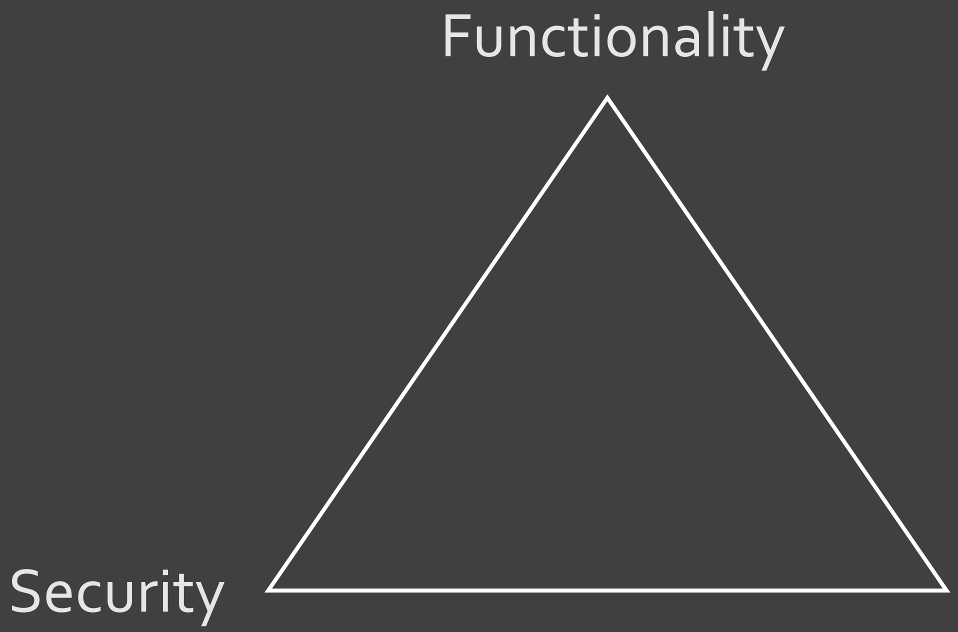
# Encrypted Search



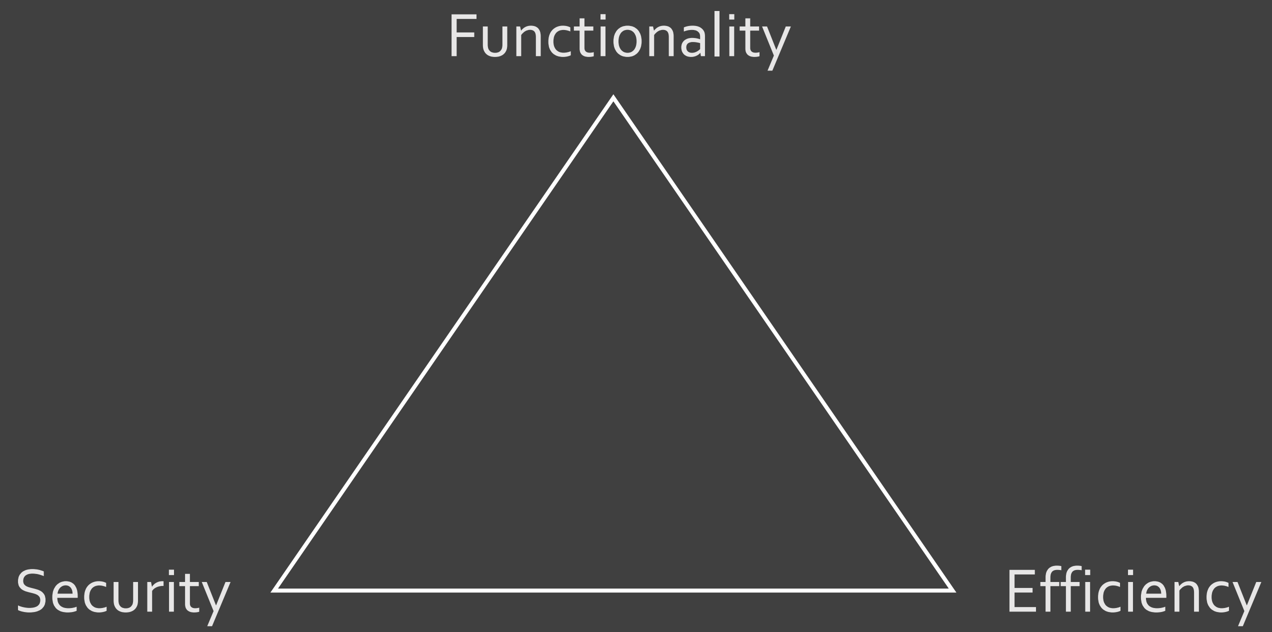
# Encrypted Search



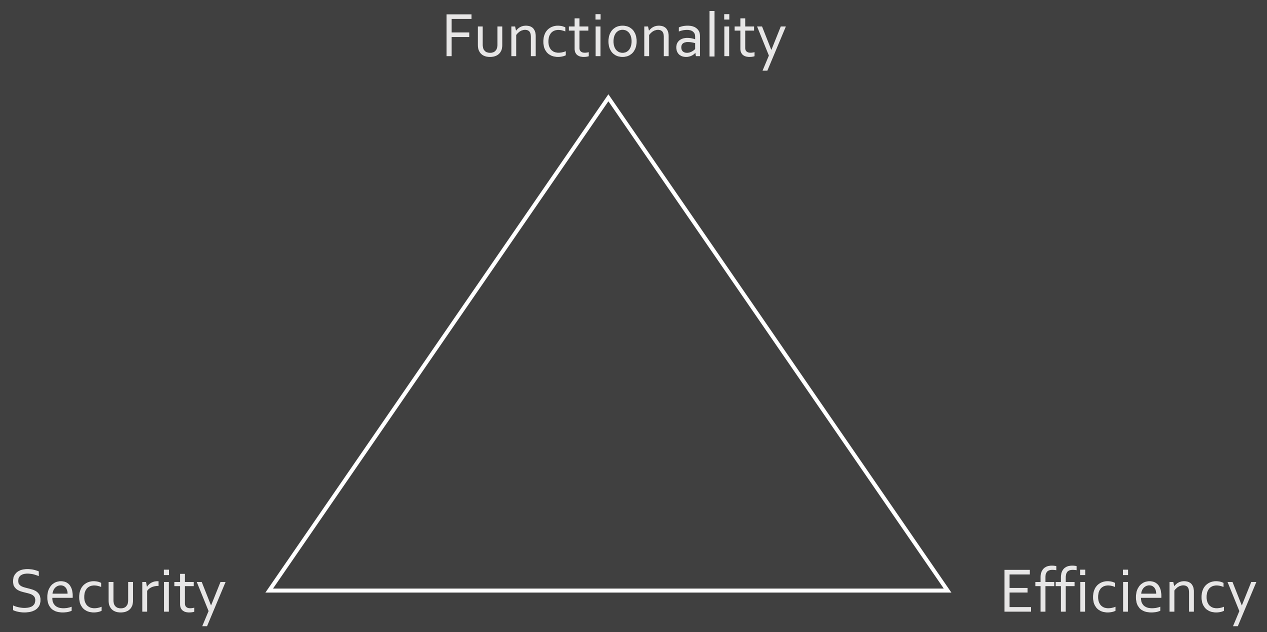
# Encrypted Search



# Encrypted Search

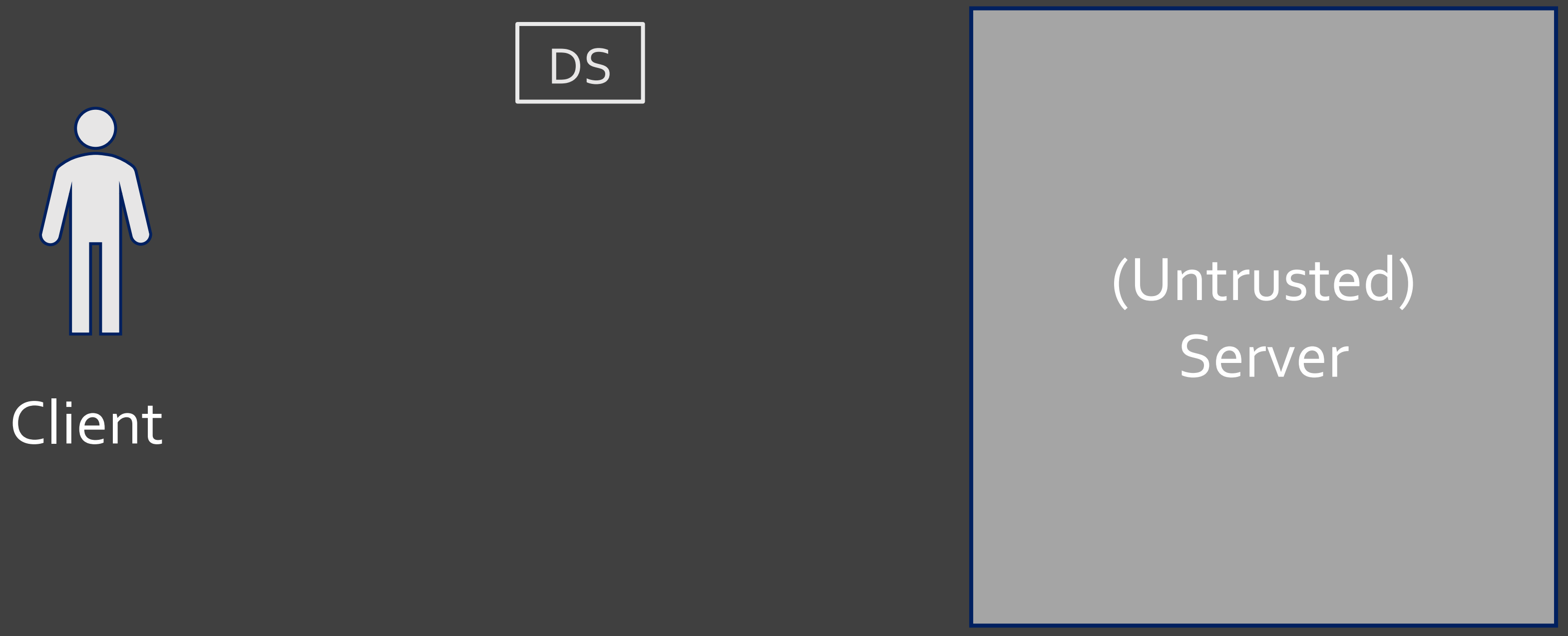


# Encrypted Search



FHE, PPE, STE …

# Structured Encryption [CK10]

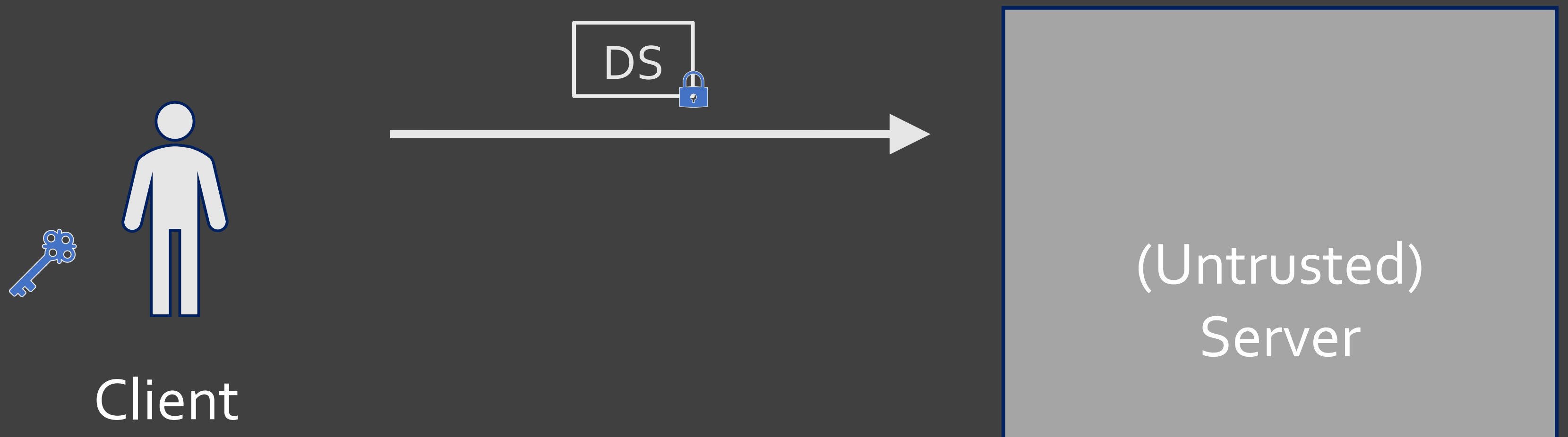


DS

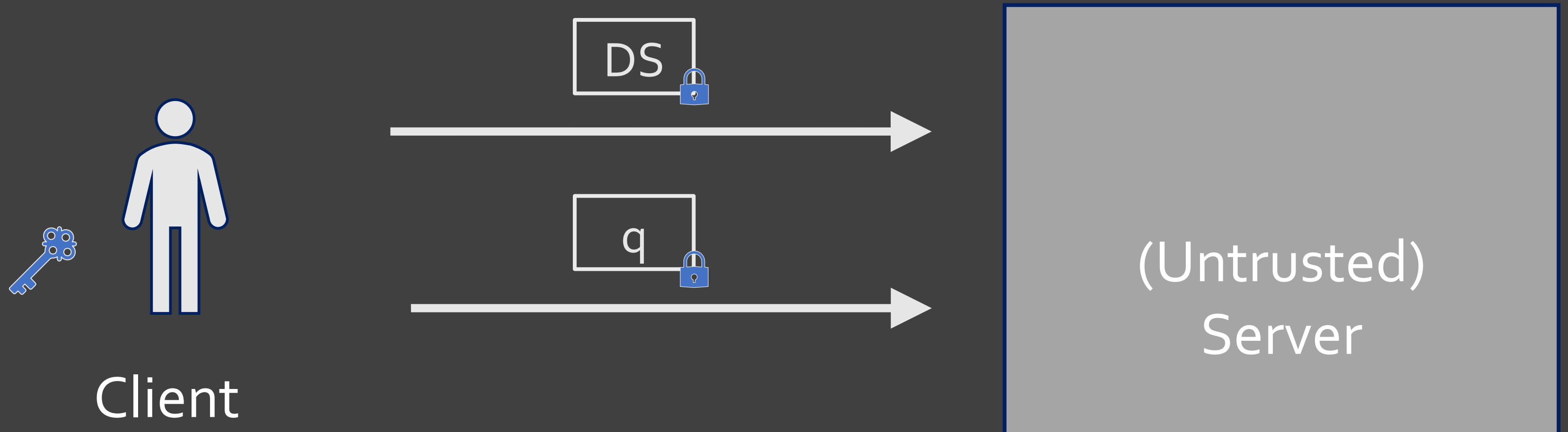
Client

(Untrusted)
Server

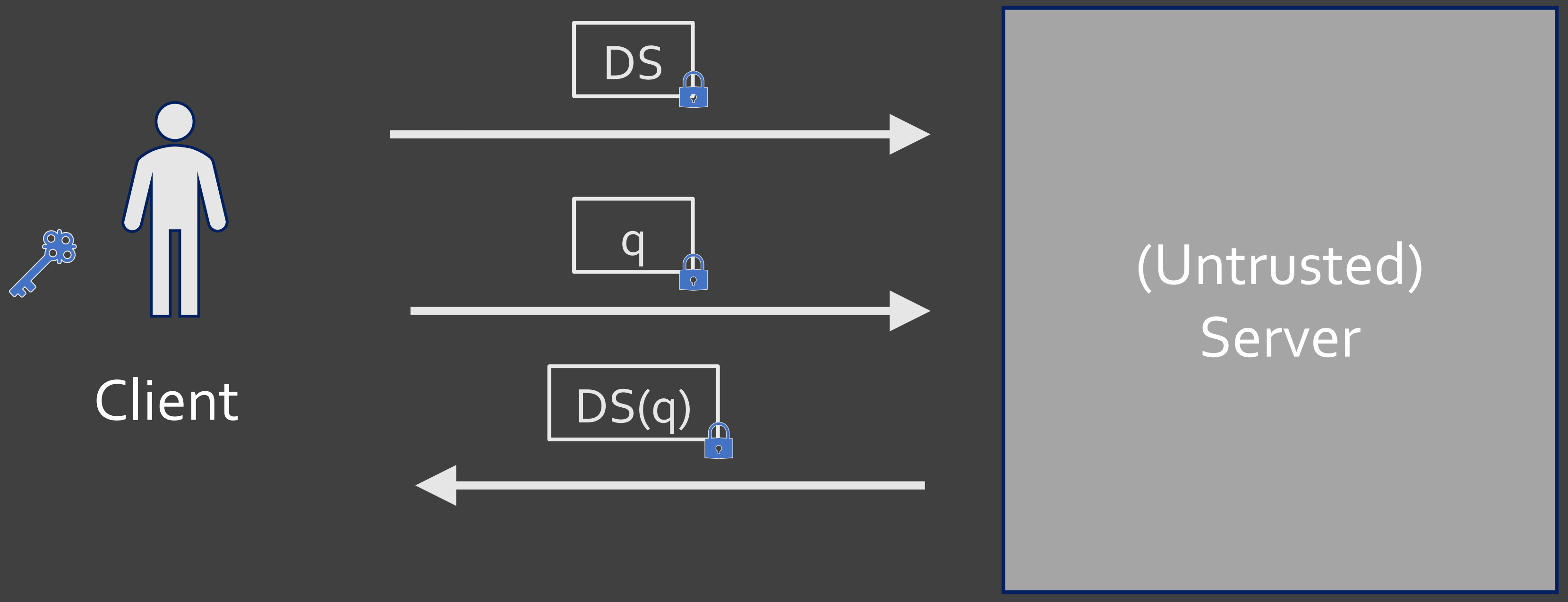
# Structured Encryption [CK10]



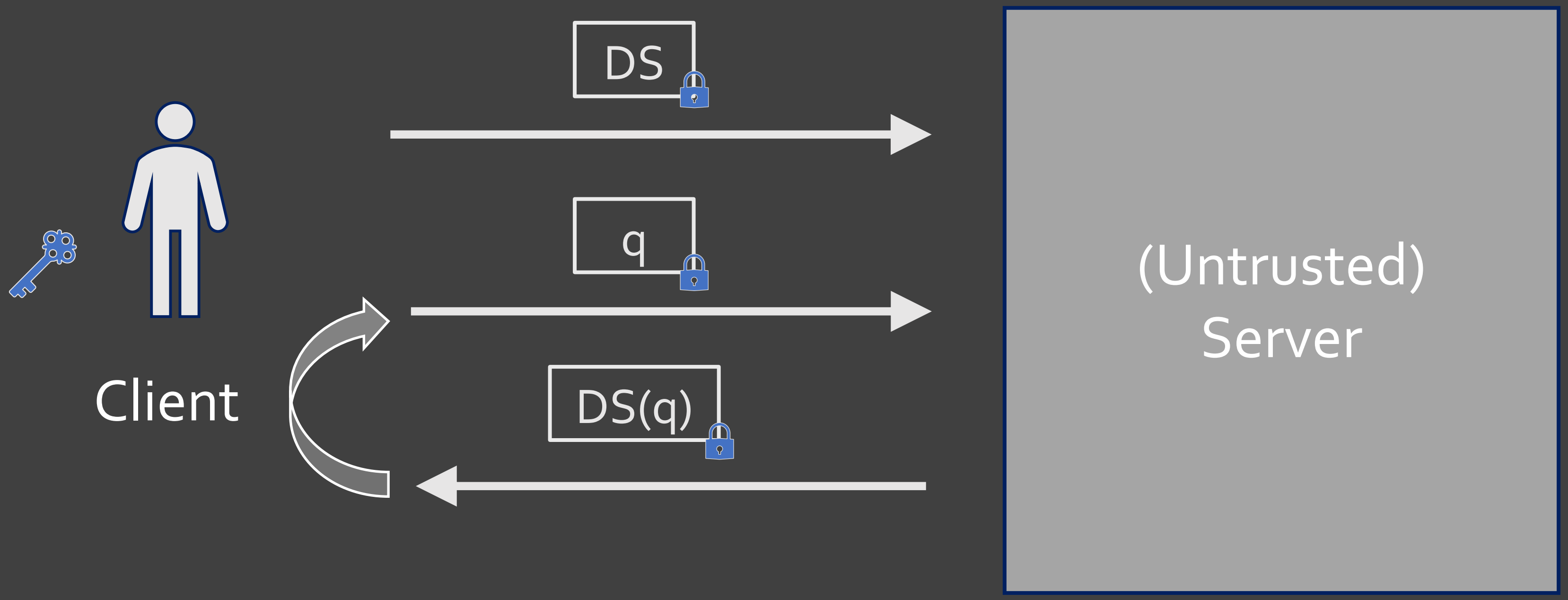
# Structured Encryption [CK10]



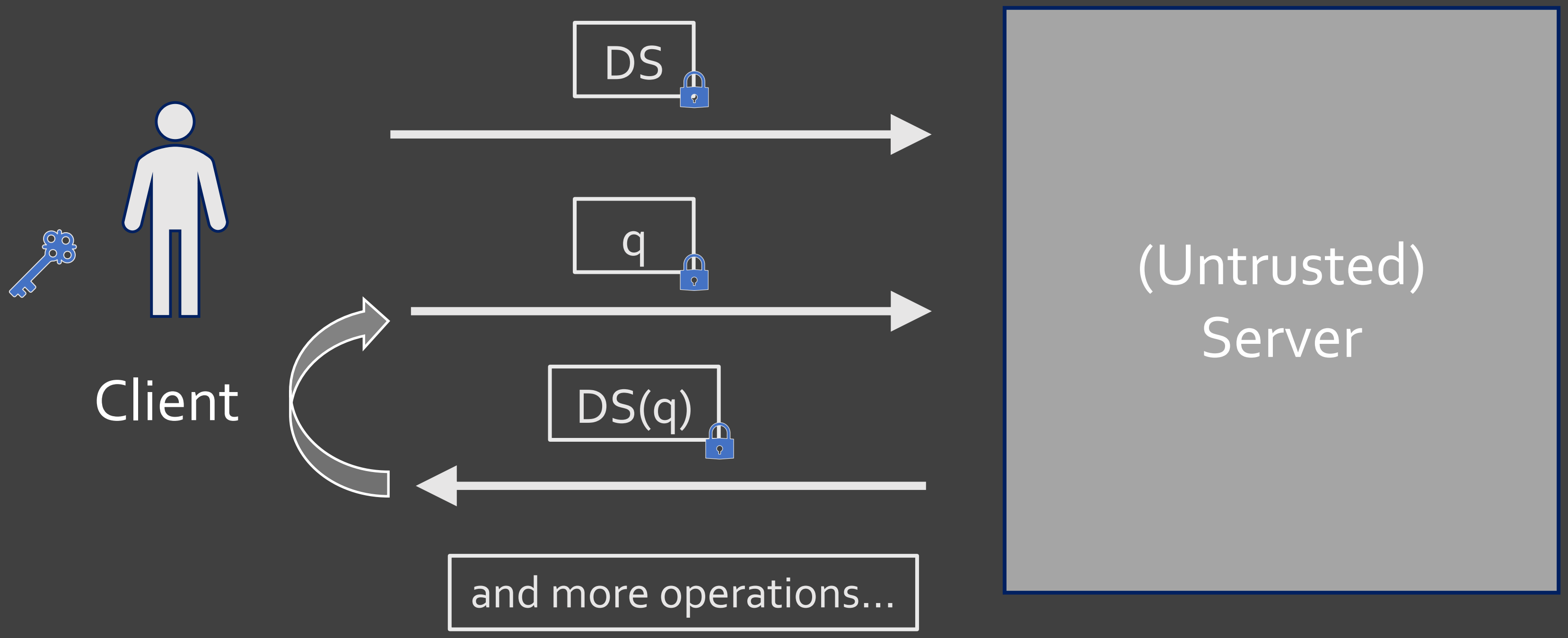
# Structured Encryption [CK10]



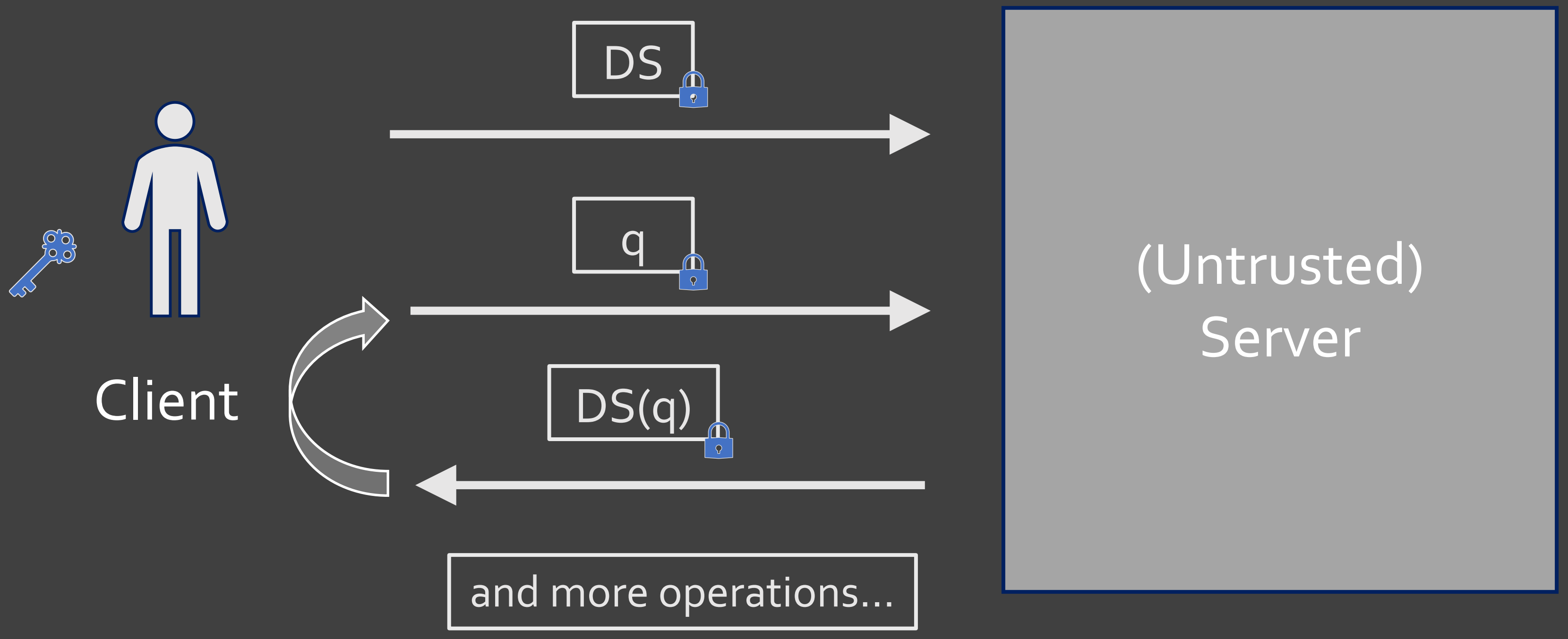
# Structured Encryption [CK10]



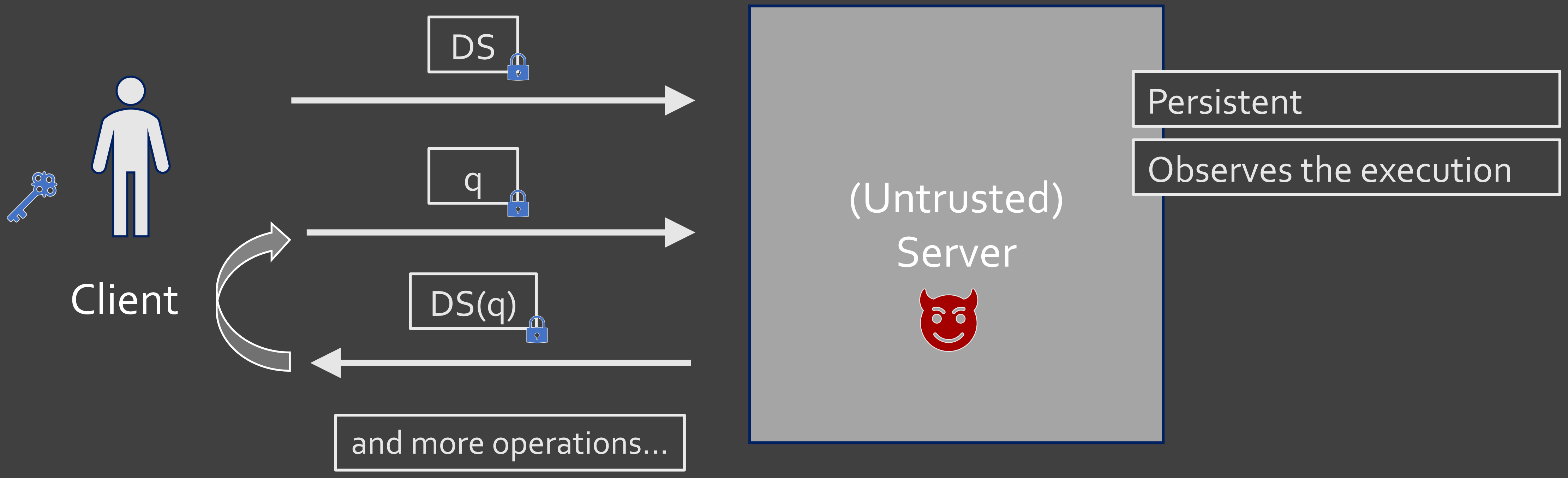
# Structured Encryption [CK10]



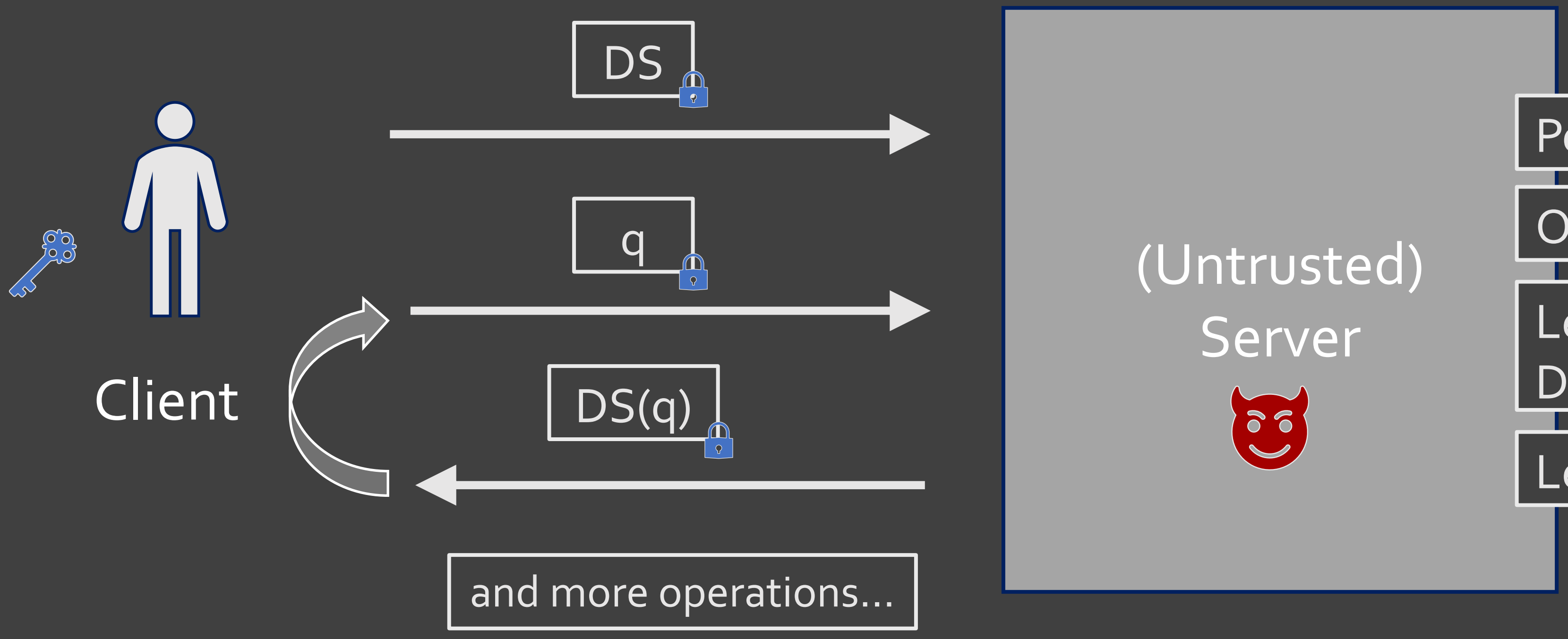
# Structured Encryption [CK10]



# Structured Encryption [CK10]



# Structured Encryption [CK10]



- Persistent
- Observes the execution
- Learns information about DS and q
- Leakage

# Studying Leakage

# Studying Leakage

How much information is leaked?
Leakage Quantification [JS19, JPS21 for PPE]

# Studying Leakage

Can leaked information be used?
Leakage Attacks [IKK12,…]

How much information is leaked?
Leakage Quantification [JS19, JPS21 for PPE]

# Studying Leakage

Can leakage be eliminated completely?
Leakage Suppression

Can leaked information be used?
Leakage Attacks [IKK12,…]

How much information is leaked?
Leakage Quantification [JS19, JPS21 for PPE]

# Studying Leakage

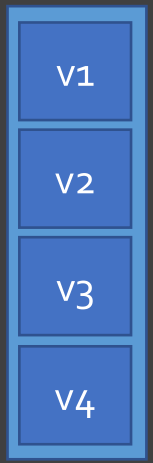
Can leakage be eliminated completely?
Leakage Suppression

Can leaked information be used?
Leakage Attacks [IKK12,…]

How much information is leaked?
Leakage Quantification [JS19, JPS21 for PPE]

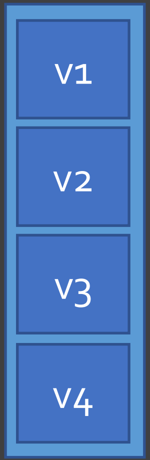
# Preliminaries: Data Structures

# Preliminaries: Data Structures



Array RAM
Read and Write

# Preliminaries: Data Structures



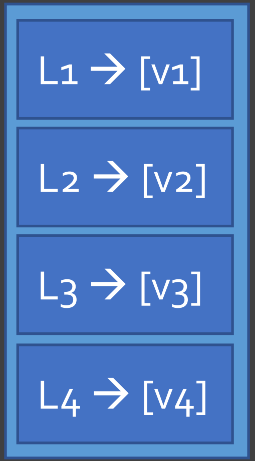
v1

v2

v3

v4

Array RAM
Read and Write



L1 → [v1]

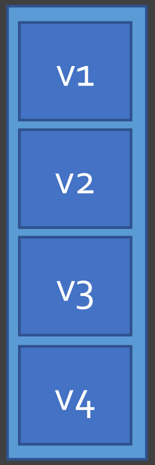
L2 → [v2]

L3 → [v3]

L4 → [v4]

Dictionary
Get and Put

# Preliminaries: Data Structures



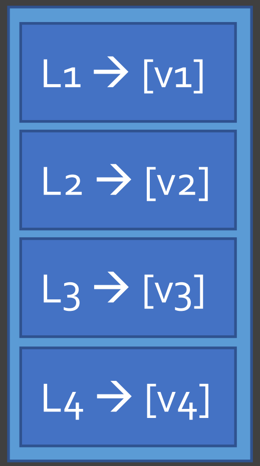
v1

v2

v3

v4

Array RAM
Read and Write



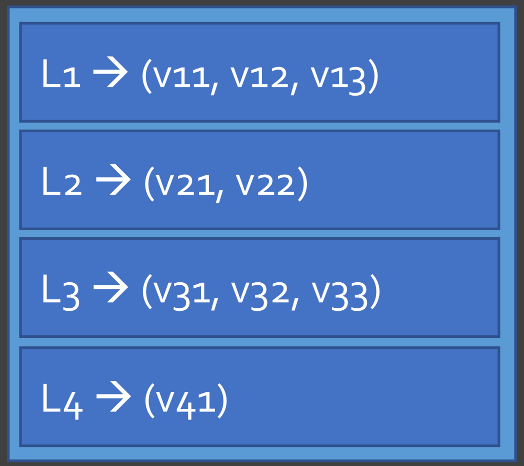
L1 → [v1]

L2 → [v2]

L3 → [v3]

L4 → [v4]

Dictionary
Get and Put



L1 → (v11, v12, v13)

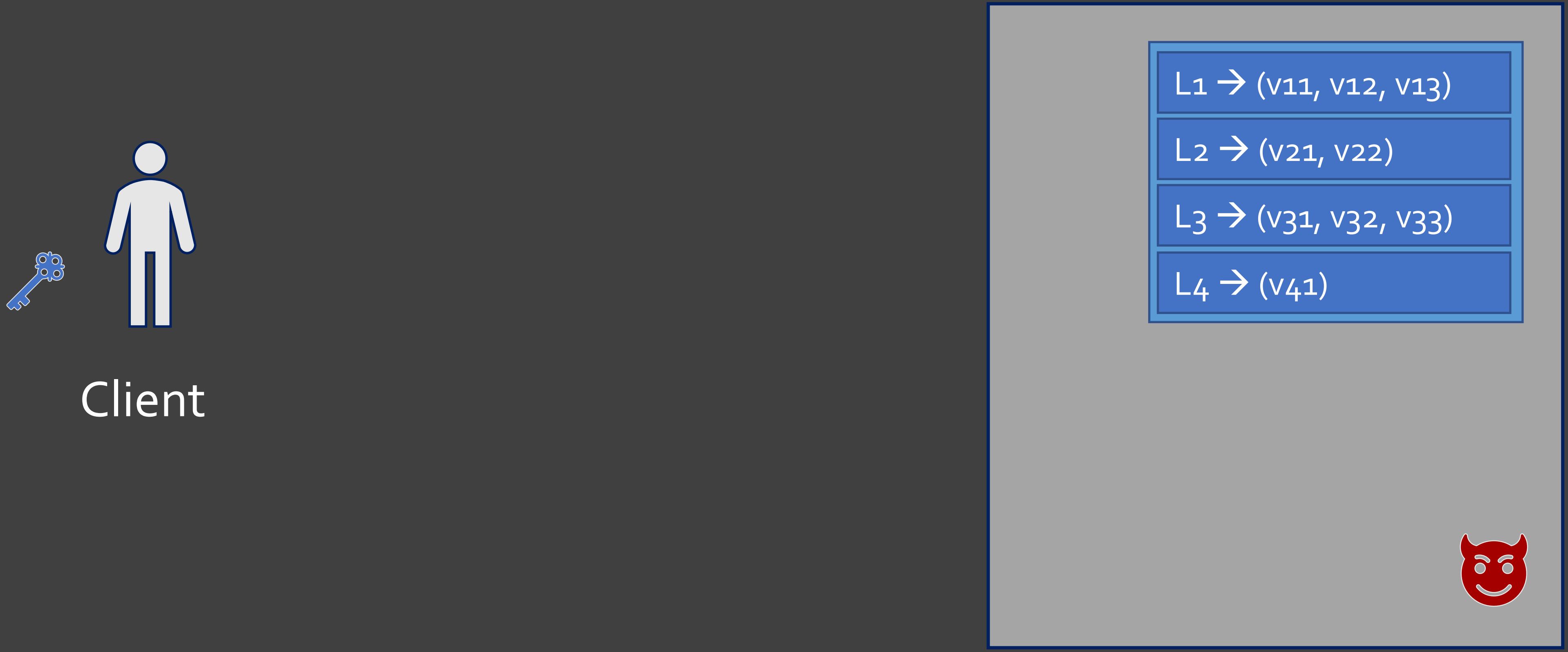
L2 → (v21, v22)

L3 → (v31, v32, v33)

L4 → (v41)

Multi-map
Get and Put

# Preliminaries: Leakage Patterns



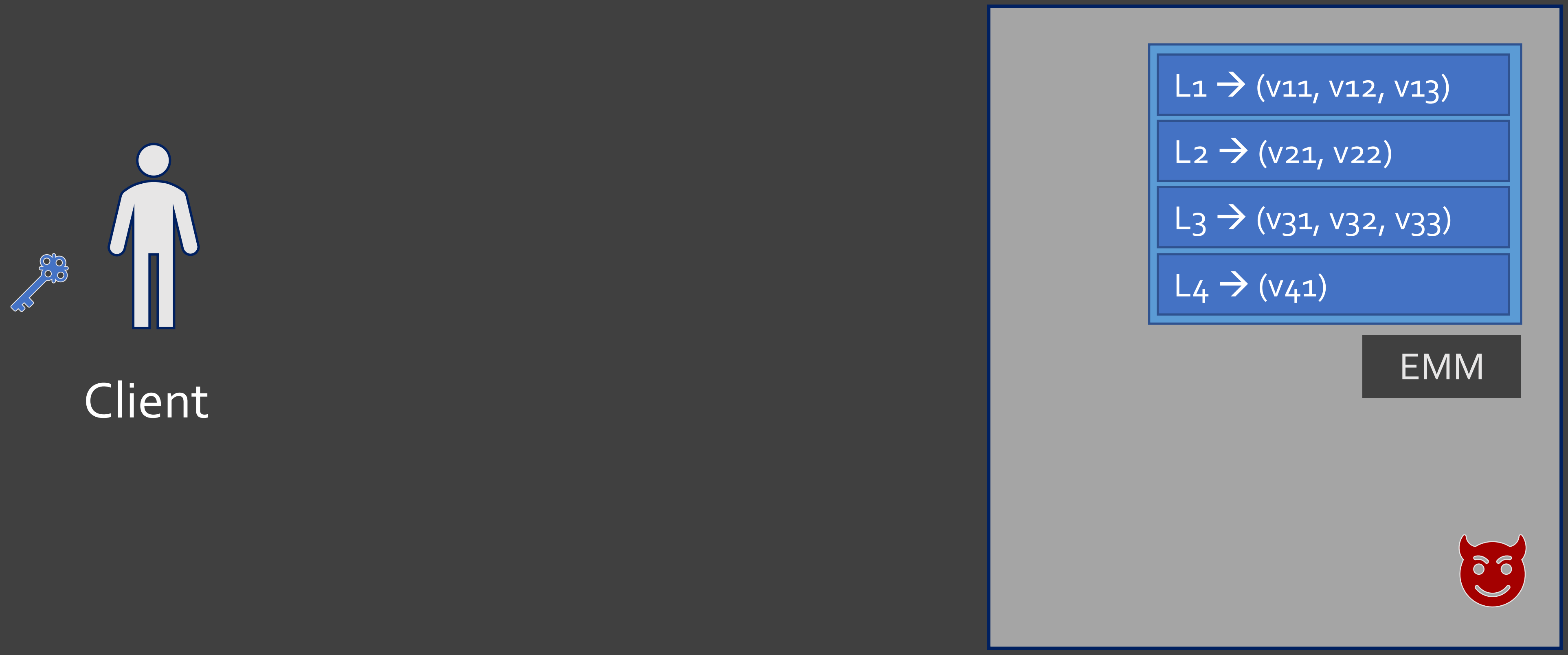
L1 → (v11, v12, v13)

L2 → (v21, v22)

L3 → (v31, v32, v33)

L4 → (v41)

# Preliminaries: Leakage Patterns



Client

L1 → (v11, v12, v13)

L2 → (v21, v22)

L3 → (v31, v32, v33)

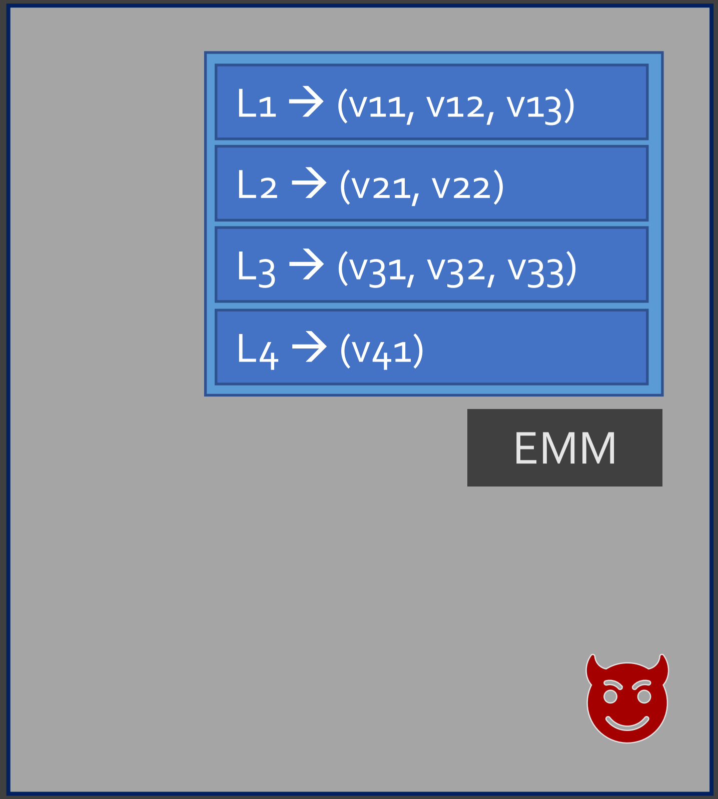
L4 → (v41)

EMM

# Preliminaries: Leakage Patterns

Client

Query equality:
Are two queries to the EMM on the same label?



L1 → (v11, v12, v13)

L2 → (v21, v22)

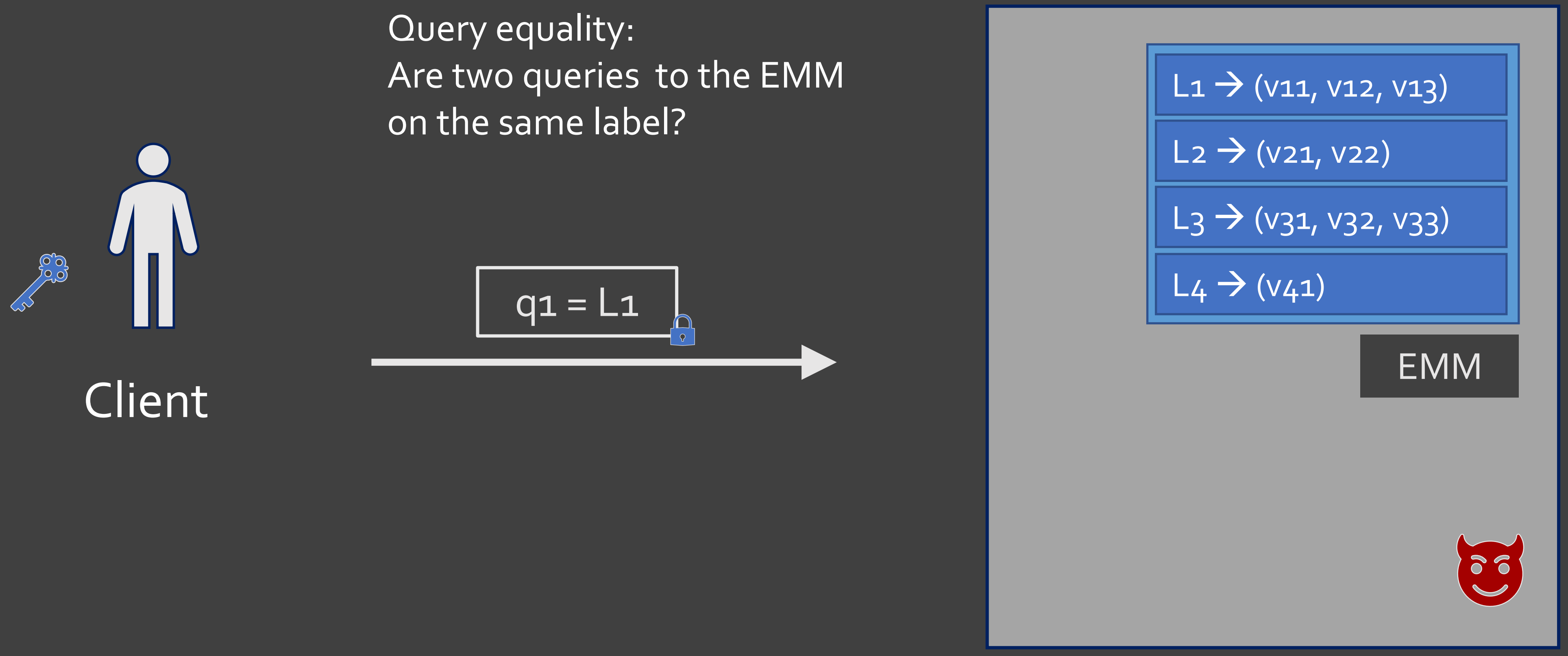
L3 → (v31, v32, v33)

L4 → (v41)

EMM

# Preliminaries: Leakage Patterns

Query equality:
Are two queries  to the EMM
on the same label?



Client

q1 = L1

L1 → (v11, v12, v13)

L2 → (v21, v22)

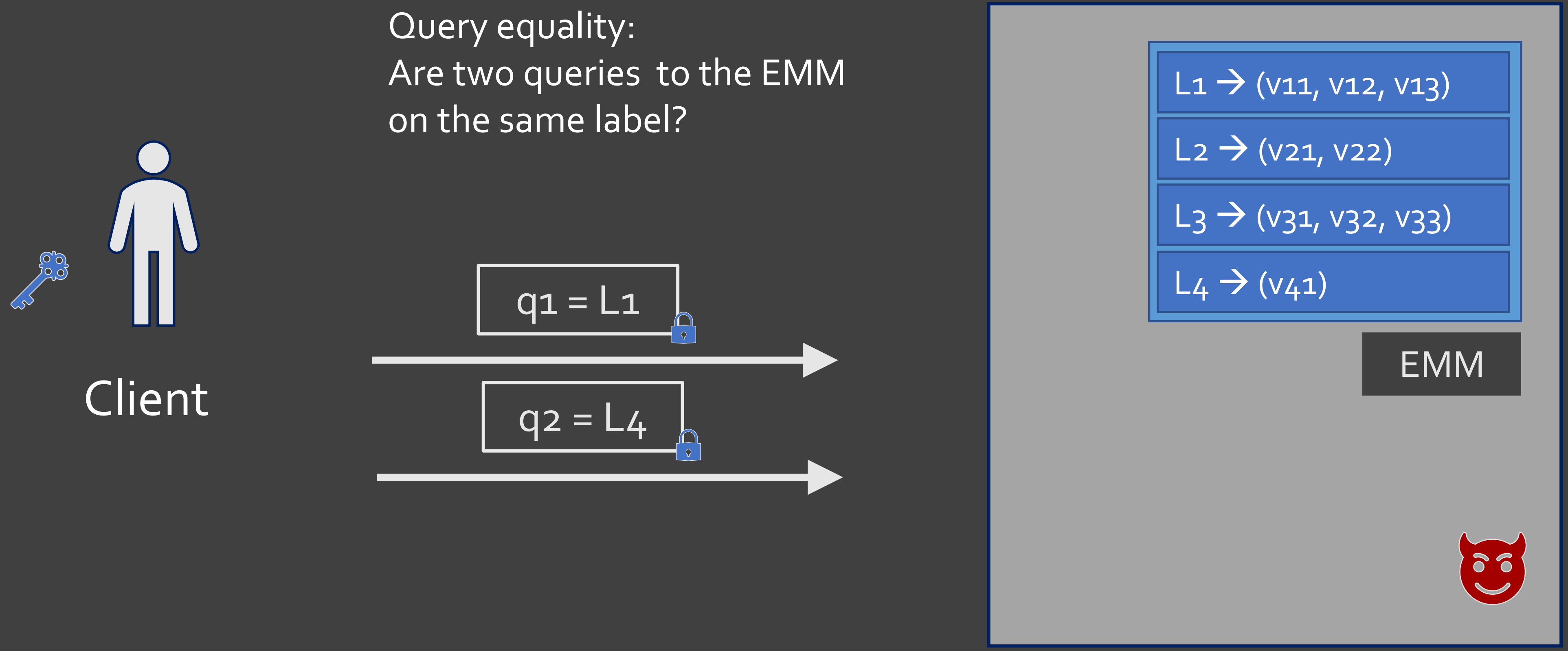
L3 → (v31, v32, v33)

L4 → (v41)

EMM

# Preliminaries: Leakage Patterns

Query equality:
Are two queries  to the EMM
on the same label?



Client

q1 = L1

q2 = L4

L1 → (v11, v12, v13)

L2 → (v21, v22)

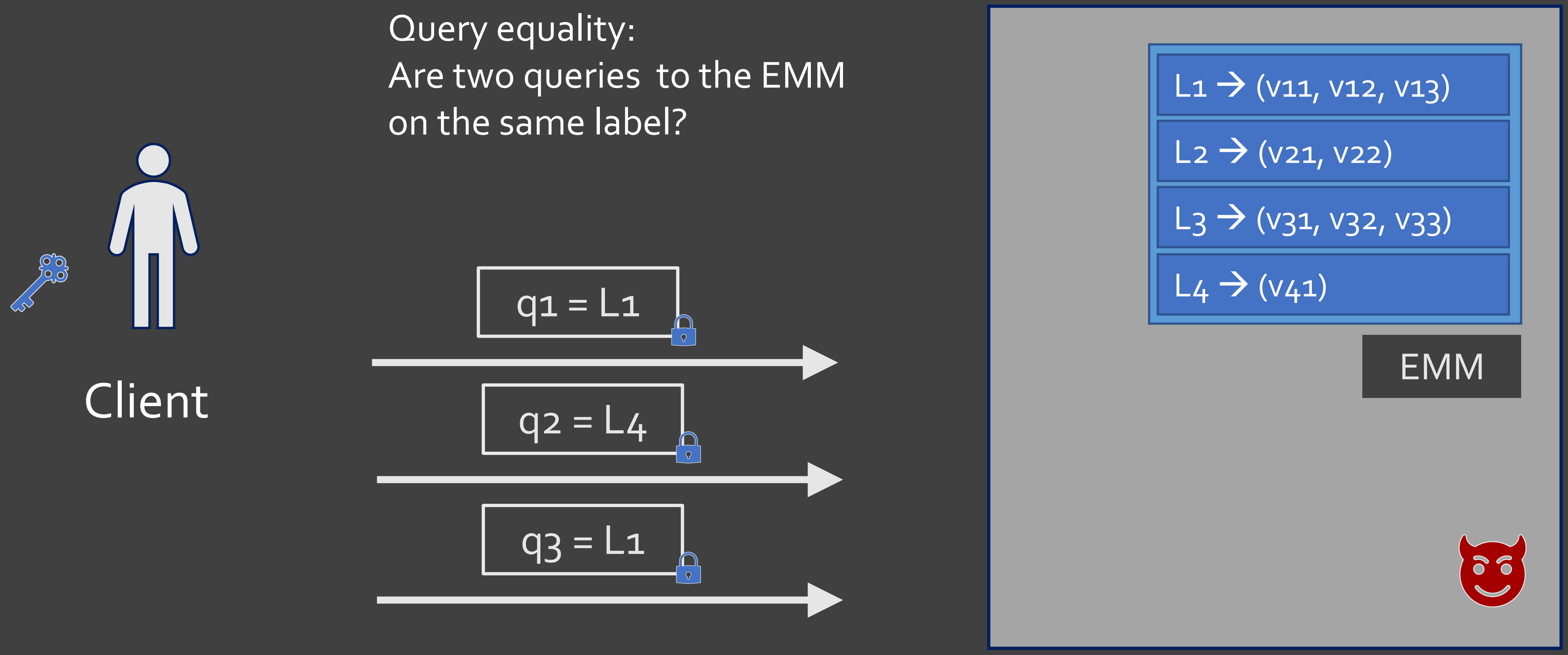
L3 → (v31, v32, v33)

L4 → (v41)

EMM

# Preliminaries: Leakage Patterns

Query equality:
Are two queries to the EMM
on the same label?



Client

q1 = L1

q2 = L4

q3 = L1

L1 → (v11, v12, v13)

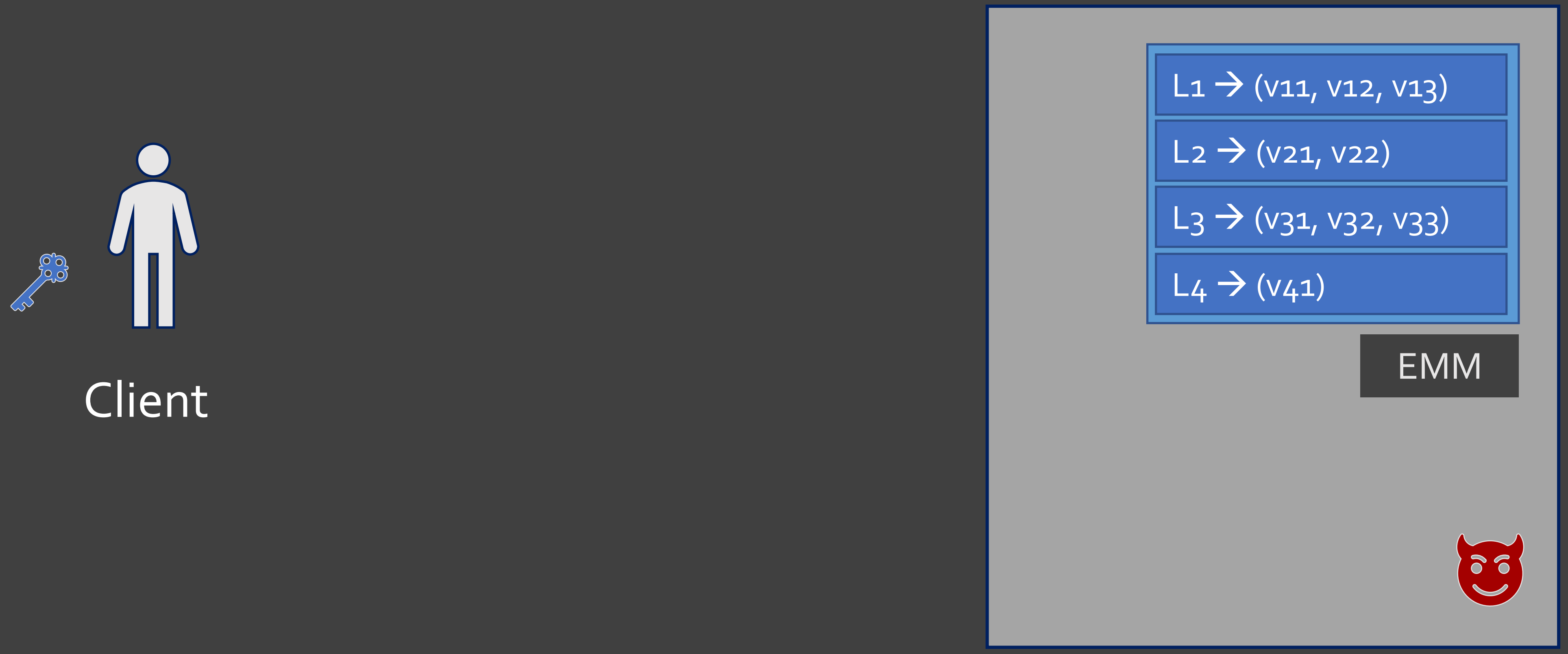
L2 → (v21, v22)

L3 → (v31, v32, v33)

L4 → (v41)

EMM

# Preliminaries: Leakage Patterns



Client

L1 → (v11, v12, v13)

L2 → (v21, v22)

L3 → (v31, v32, v33)

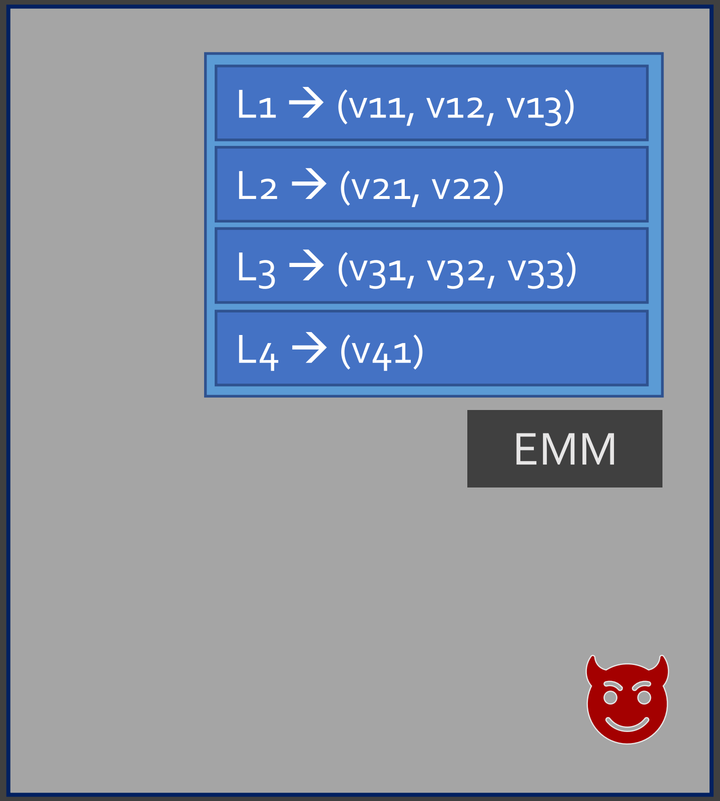
L4 → (v41)

EMM

# Preliminaries: Leakage Patterns

Volume:
How many values correspond to a query?

Client



L1 → (v11, v12, v13)

L2 → (v21, v22)

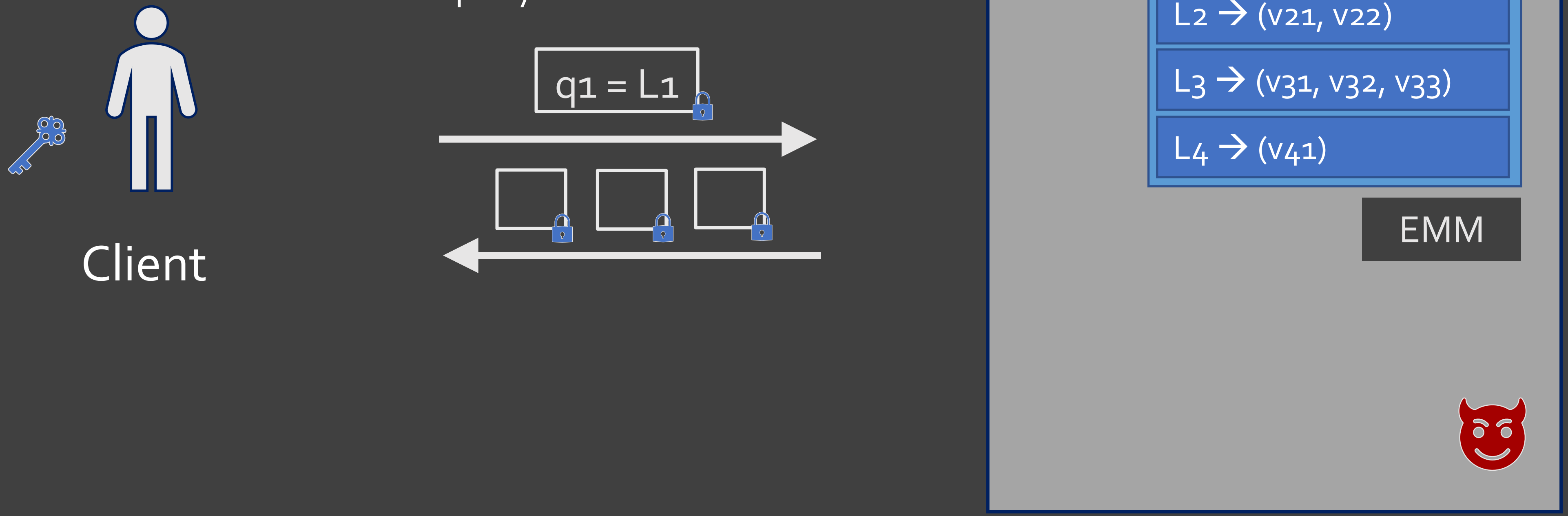
L3 → (v31, v32, v33)

L4 → (v41)

EMM

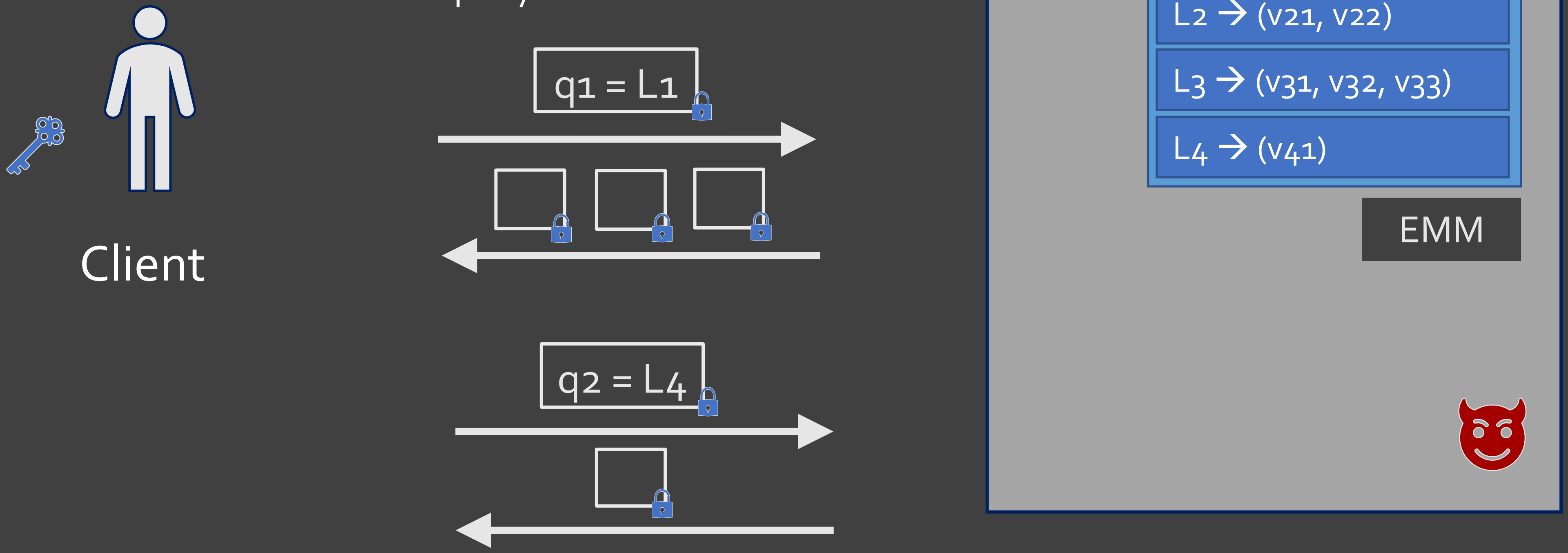
# Preliminaries: Leakage Patterns

Volume:
How many values correspond to a query?

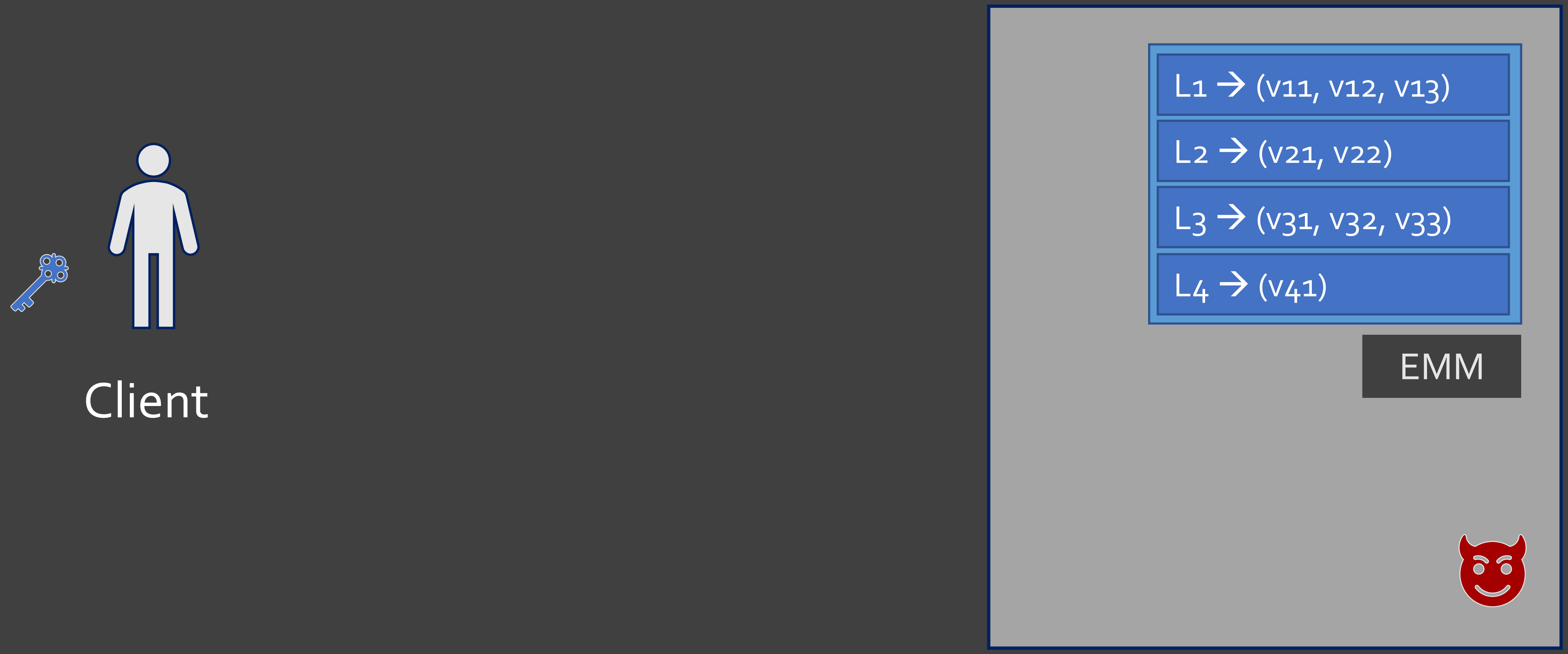


# Preliminaries: Leakage Patterns

Volume:
How many values correspond to a query?



# Preliminaries: Leakage Patterns



Client

L1 → (v11, v12, v13)

L2 → (v21, v22)

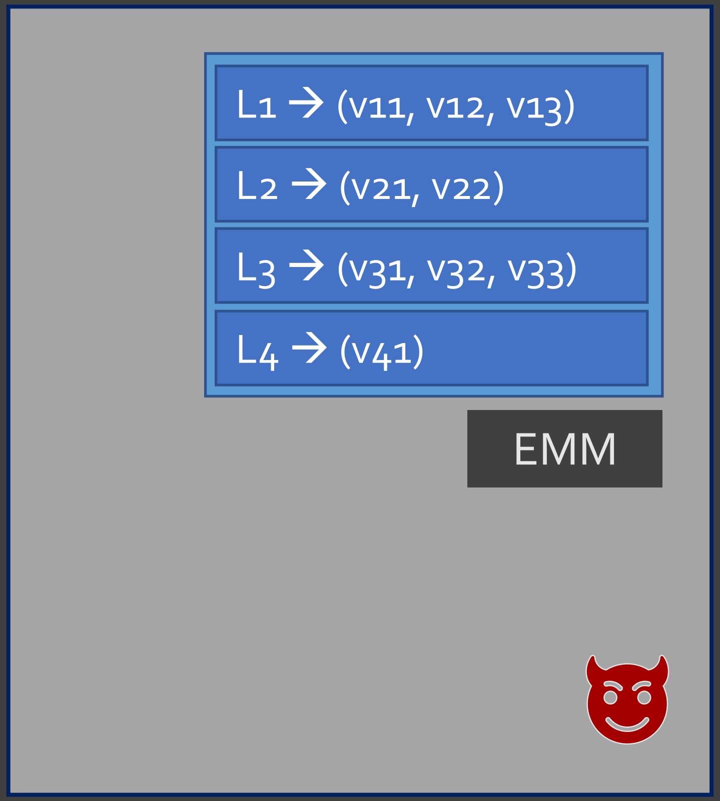
L3 → (v31, v32, v33)

L4 → (v41)

EMM

# Preliminaries: Leakage Patterns

Operation identity:
Which operation is the client running?



L1 → (v11, v12, v13)

L2 → (v21, v22)

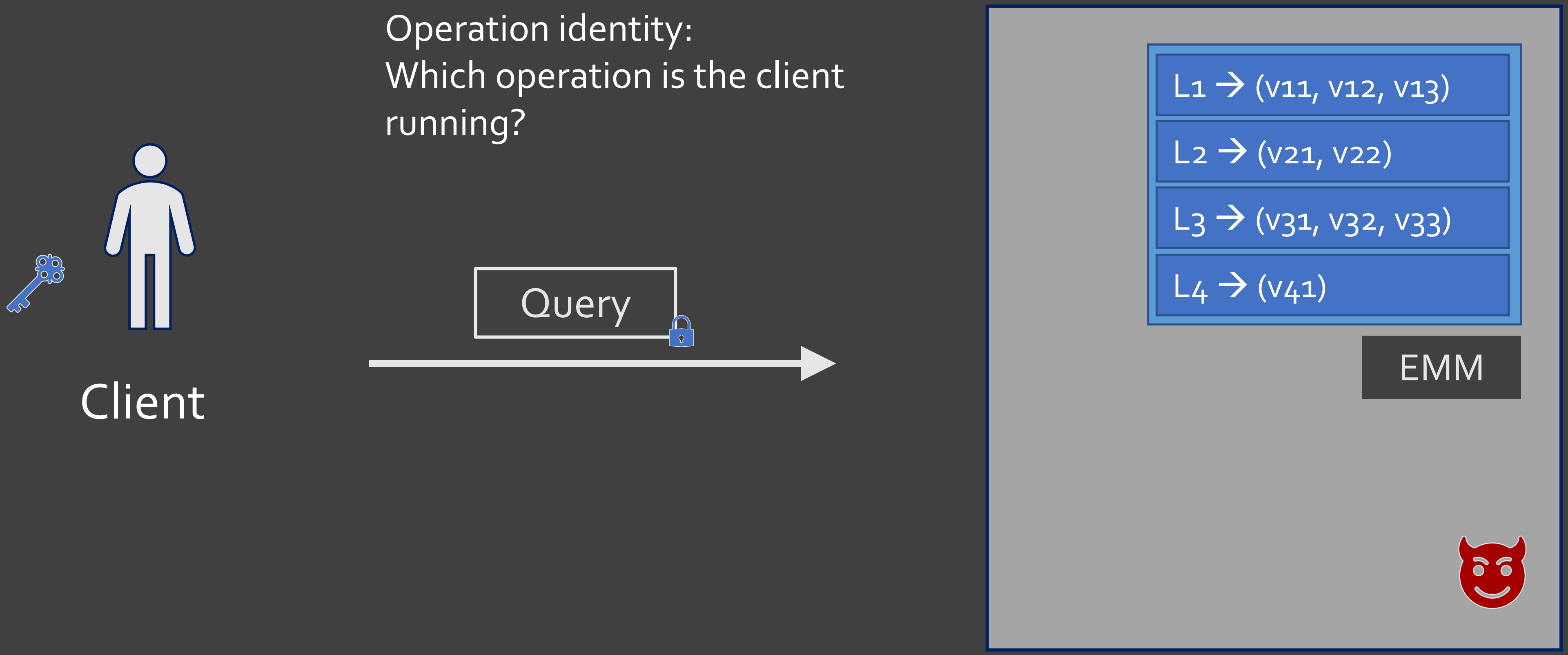
L3 → (v31, v32, v33)

L4 → (v41)

EMM

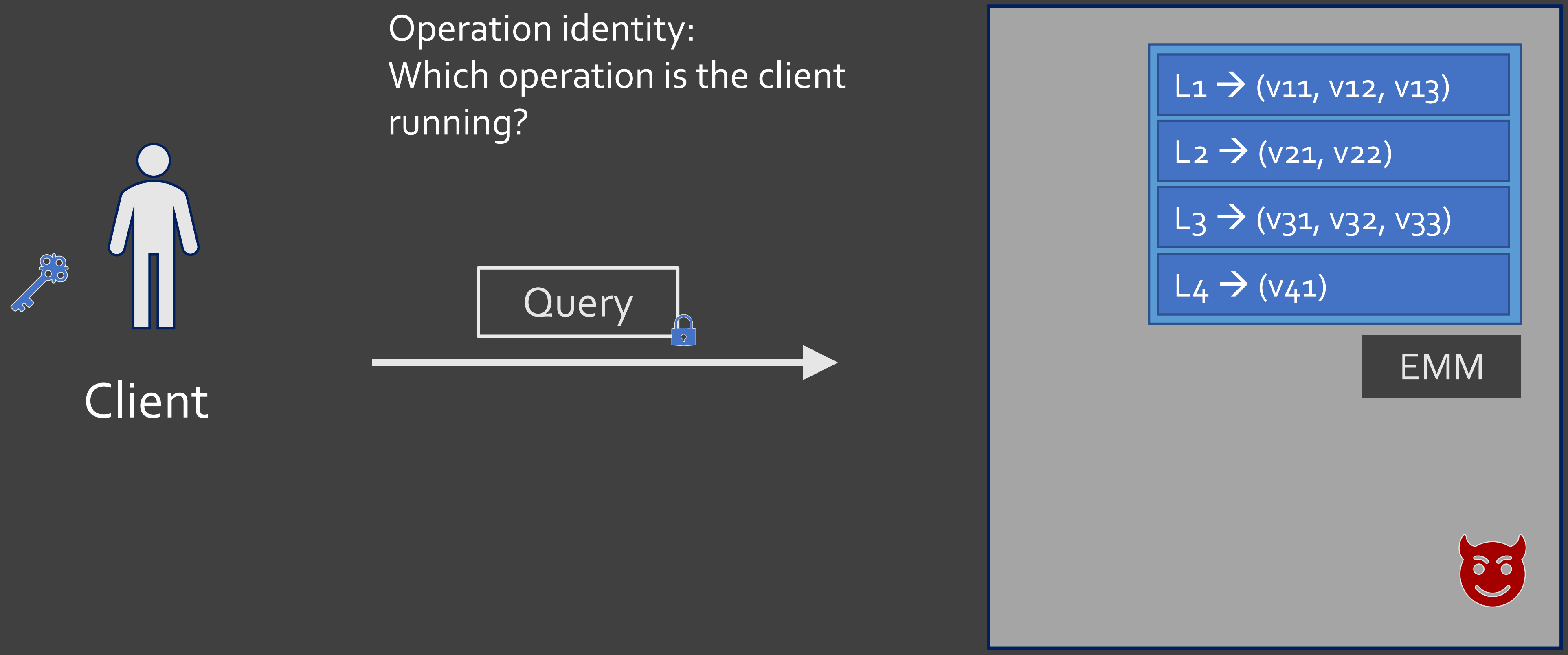
# Preliminaries: Leakage Patterns

Operation identity:
Which operation is the client running?



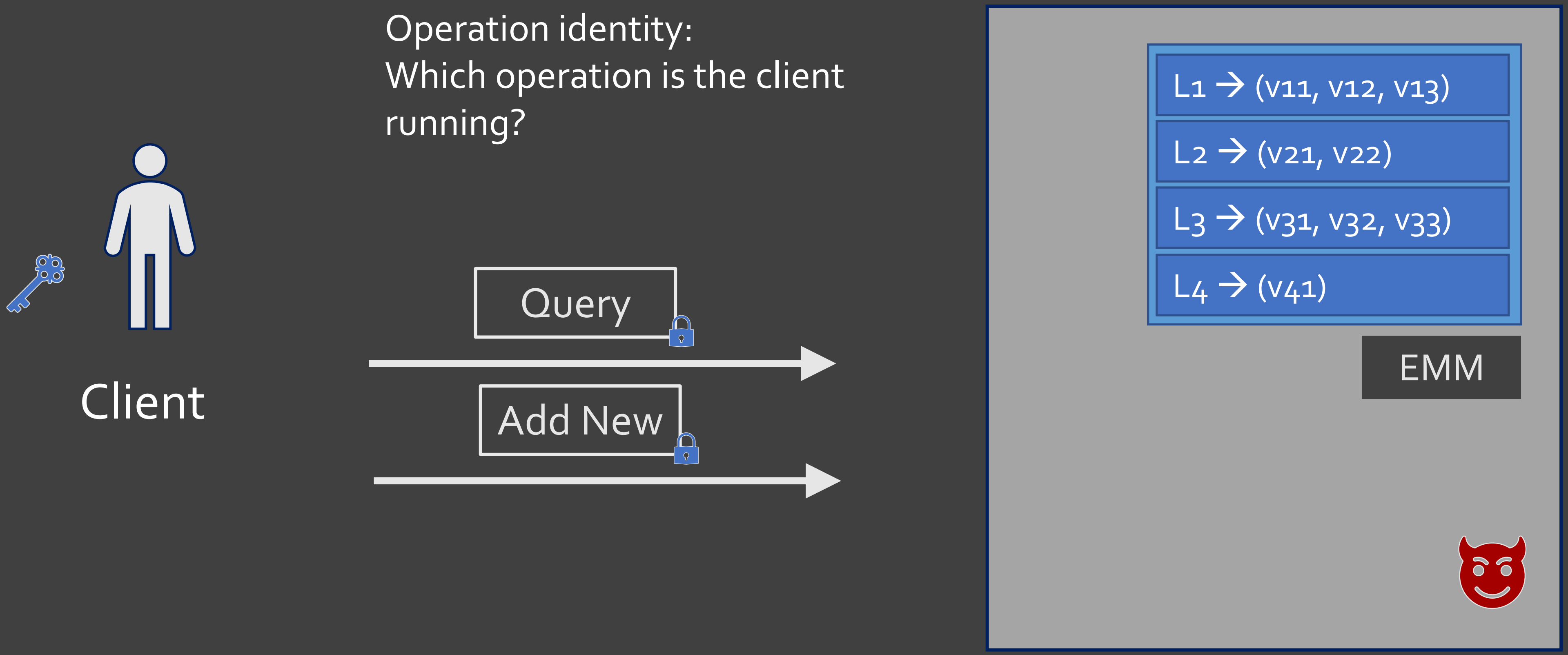
# Preliminaries: Leakage Patterns

Operation identity:
Which operation is the client running?



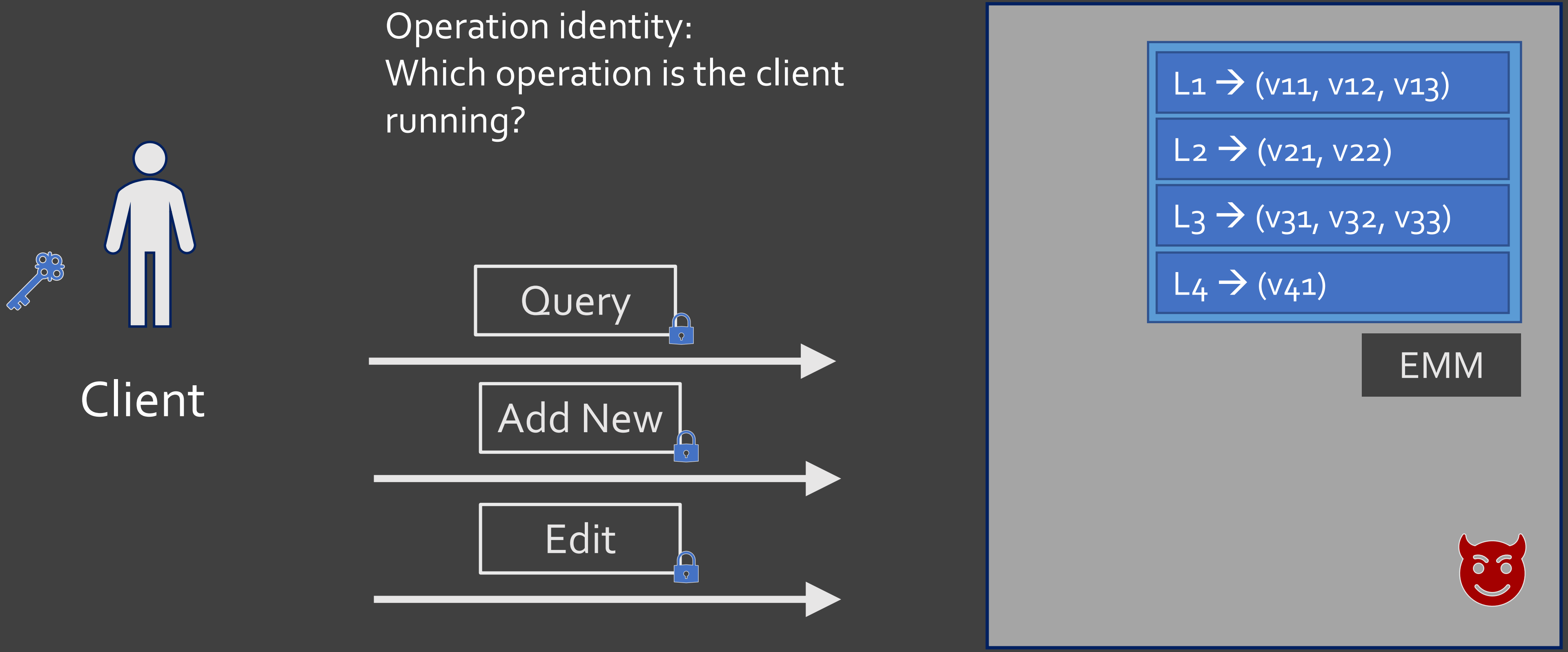
# Preliminaries: Leakage Patterns

Operation identity:
Which operation is the client running?



# Preliminaries: Leakage Patterns

Operation identity:
Which operation is the client running?



Client

Query

Add New

Edit

L1 → (v11, v12, v13)

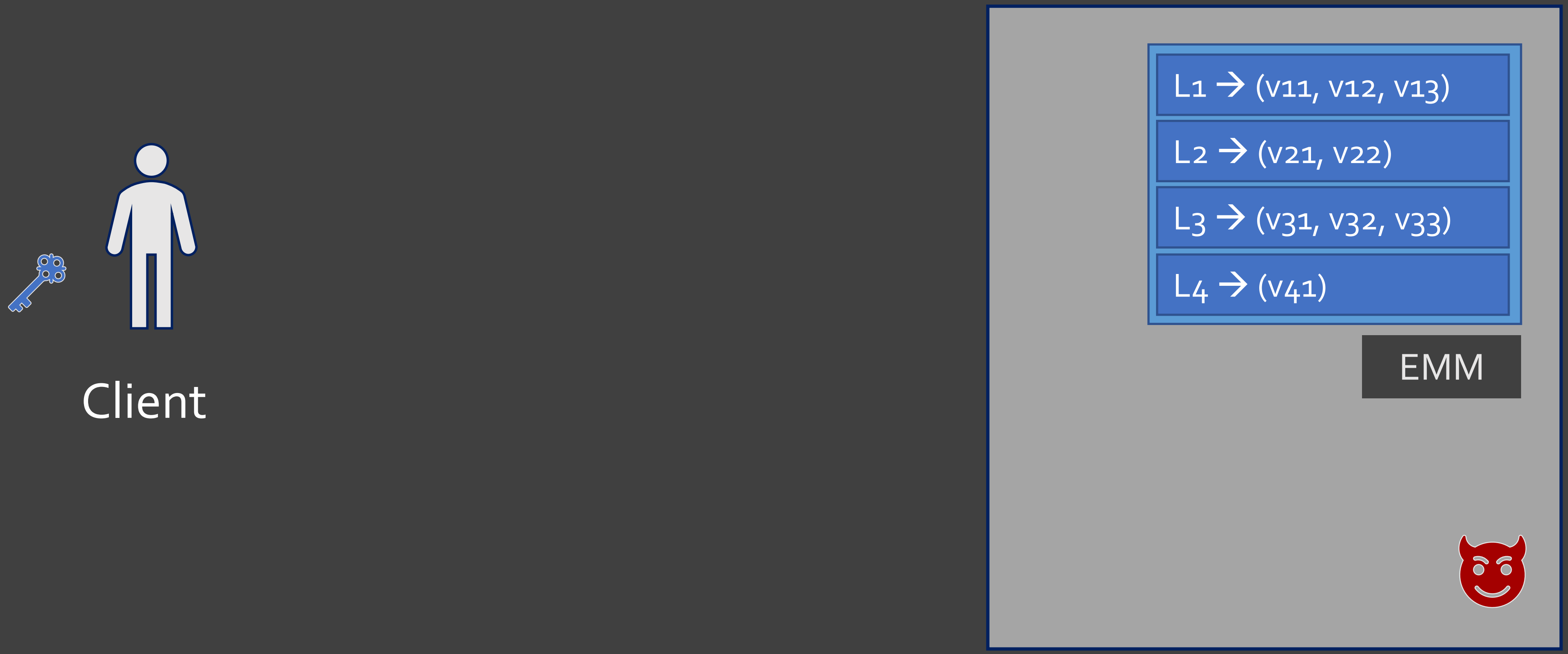
L2 → (v21, v22)

L3 → (v31, v32, v33)

L4 → (v41)

EMM

# Preliminaries: Leakage Patterns



Client

L1 → (v11, v12, v13)

L2 → (v21, v22)

L3 → (v31, v32, v33)

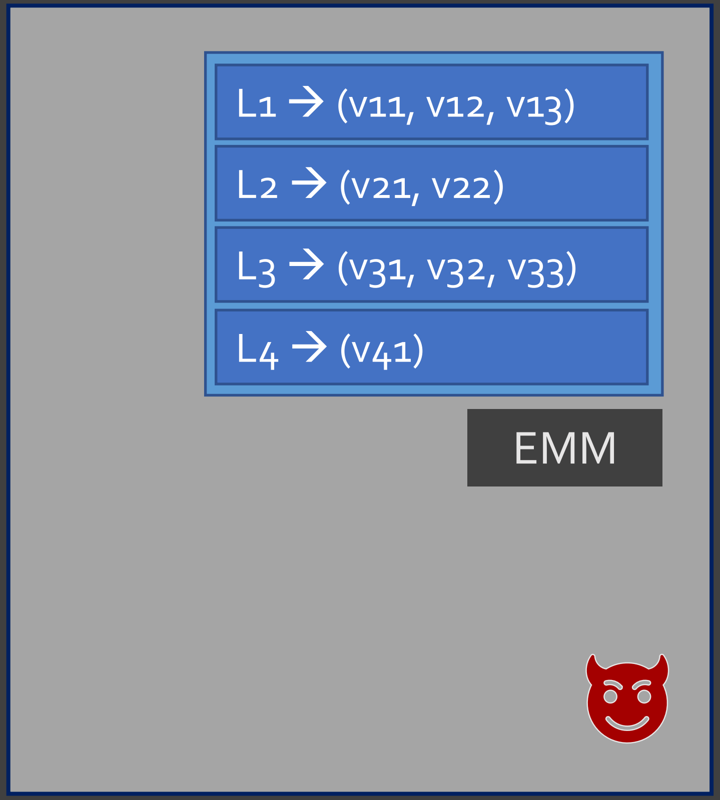
L4 → (v41)

EMM

# Preliminaries: Leakage Patterns

Client

Operation equality:
Are two operations on the same label?



L1 → (v11, v12, v13)

L2 → (v21, v22)

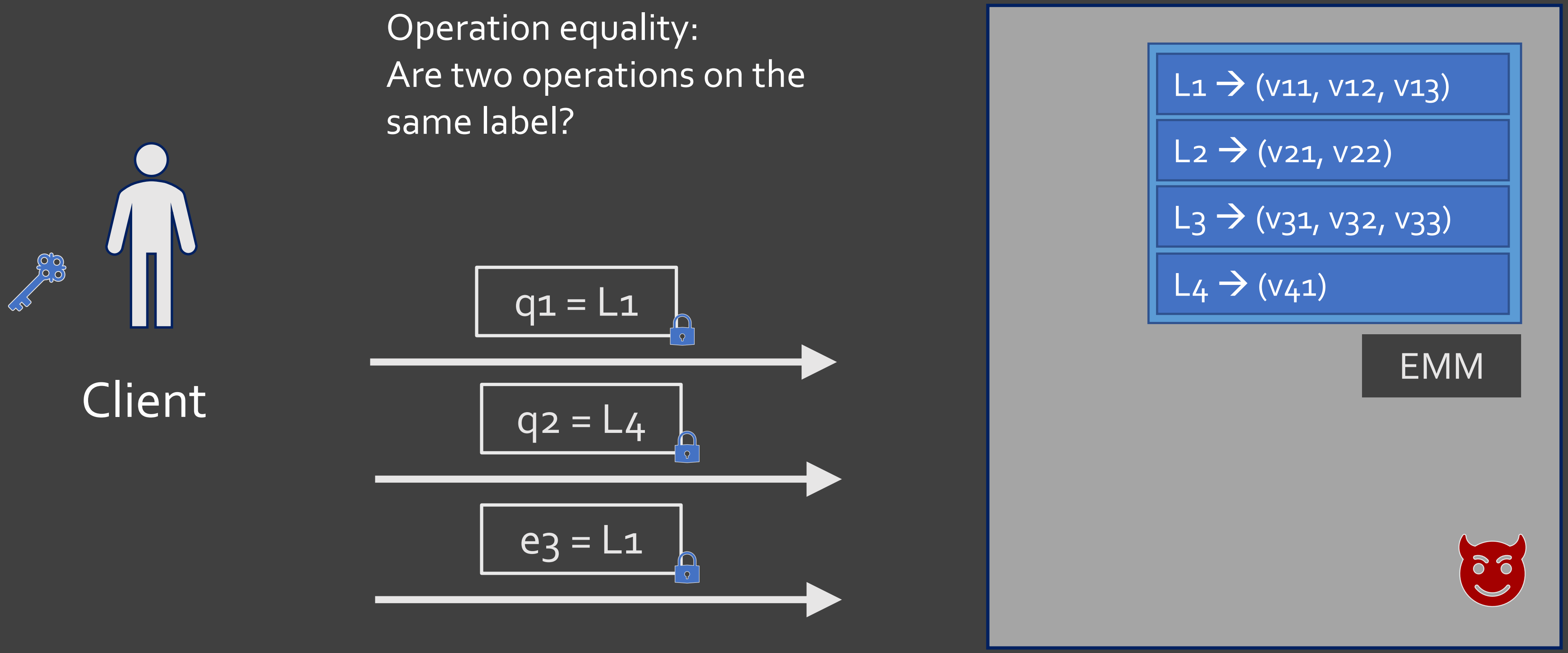
L3 → (v31, v32, v33)

L4 → (v41)

EMM

# Preliminaries: Leakage Patterns

Operation equality:
Are two operations on the same label?



# Leakage Suppression

(Untrusted) Server

😈

Can leakage be eliminated completely?

# Leakage Suppression

(Untrusted)
Server
😈

Can leakage be eliminated completely?

Query Equality Pattern:
Static Framework [KMO18]

# Leakage Suppression

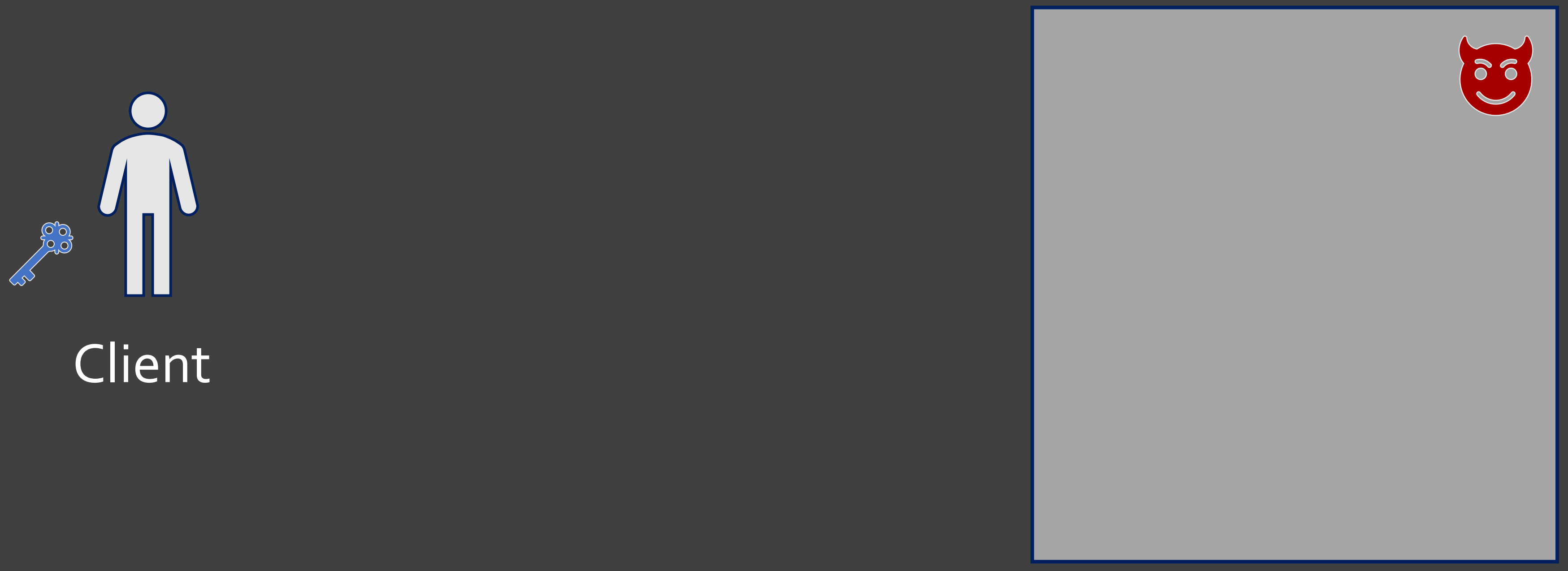
(Untrusted)
Server
😈

Can leakage be eliminated completely?

Query Equality Pattern:
Static Framework [KMO18]

Volume Pattern:
Computational Volume-Hiding [KM19]
Volume-Hiding via Hashing [PPYY19]

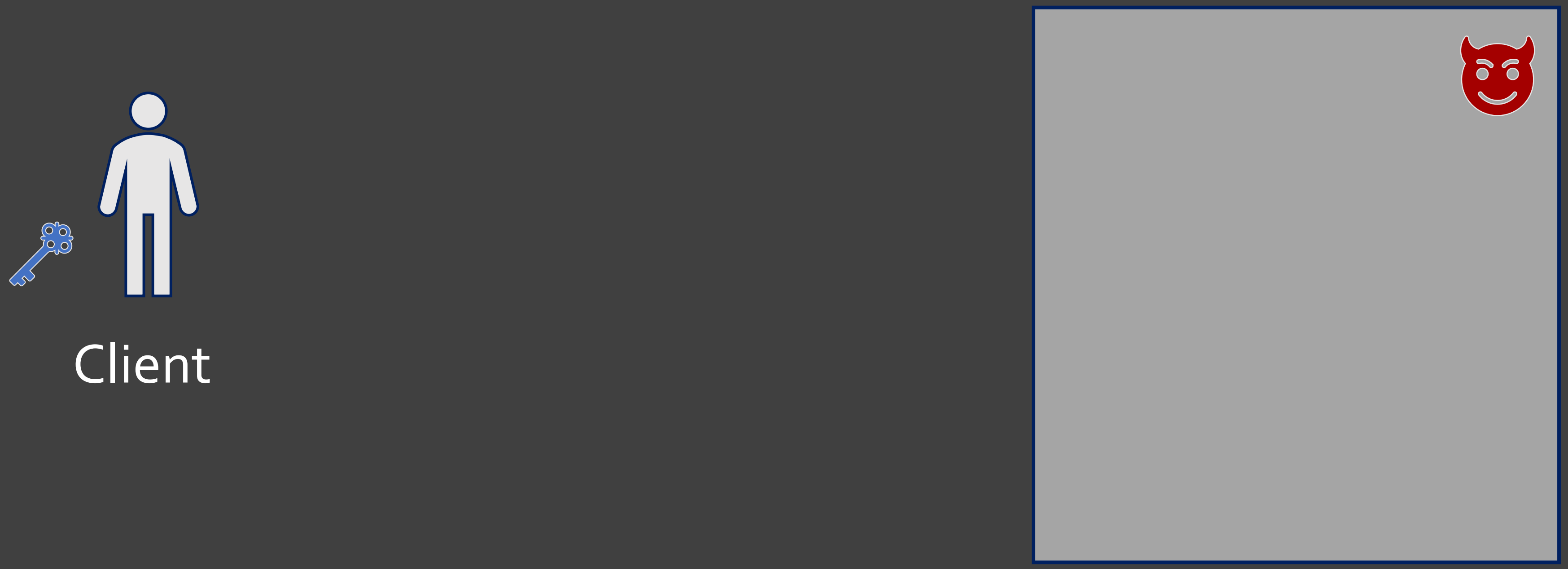
# Query Equality Suppression



Client

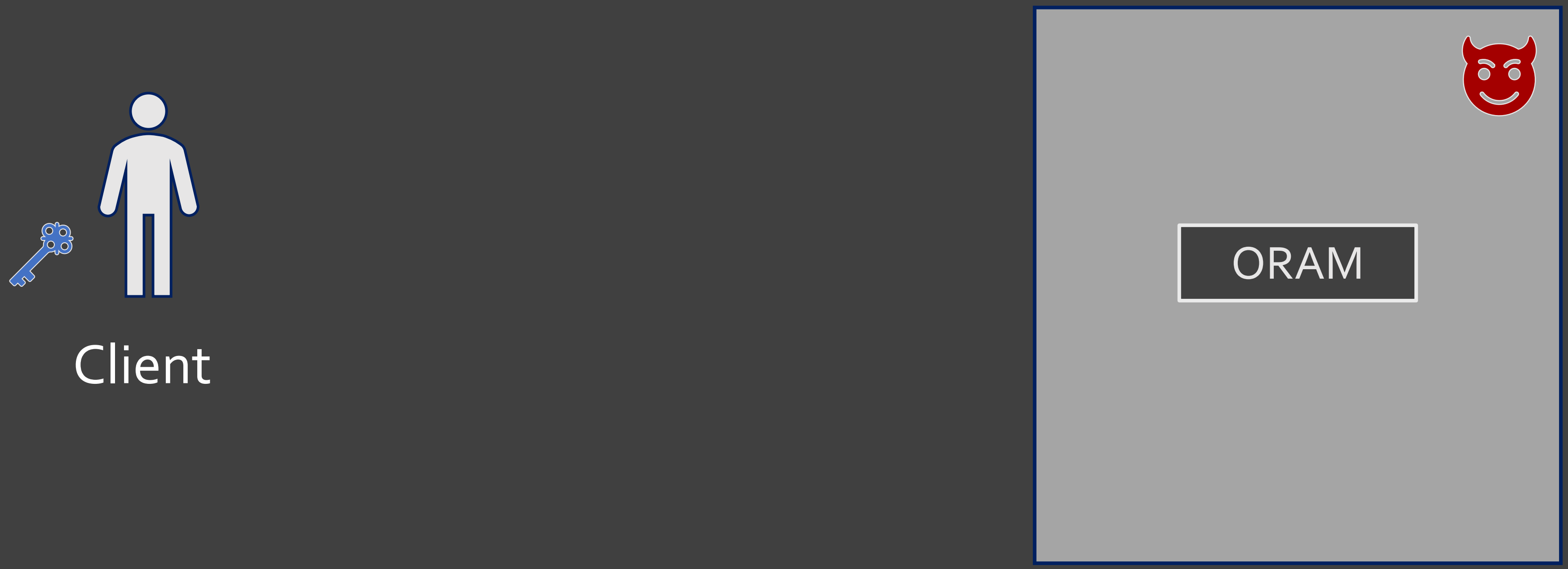
# Query Equality Suppression

- Black-box ORAM simulation



# Query Equality Suppression

- Black-box ORAM simulation

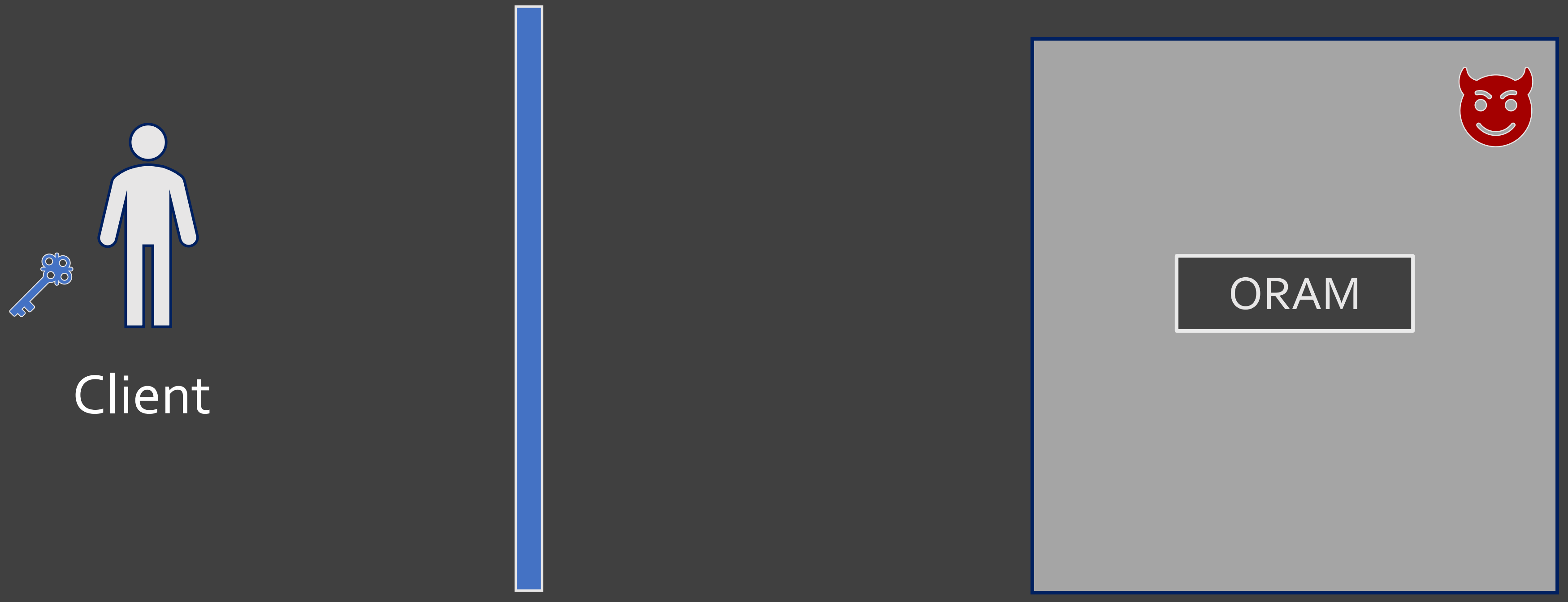


Client

ORAM

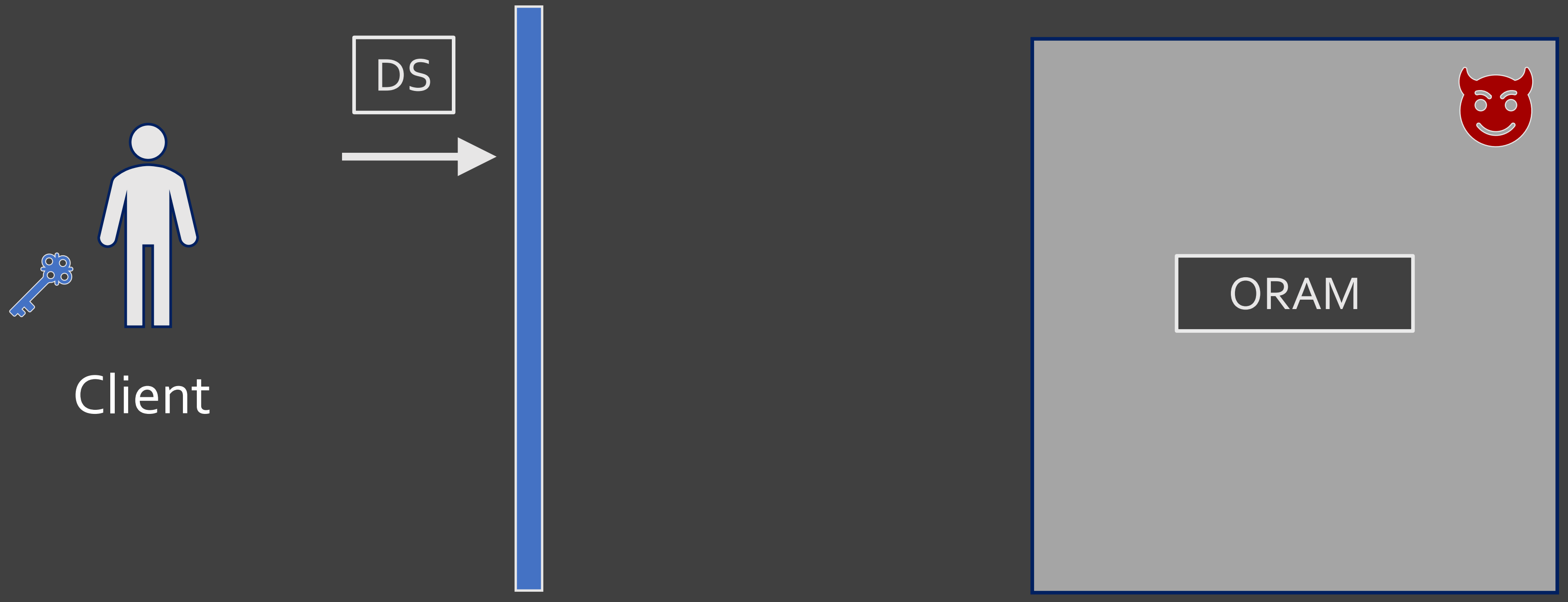
# Query Equality Suppression

- Black-box ORAM simulation



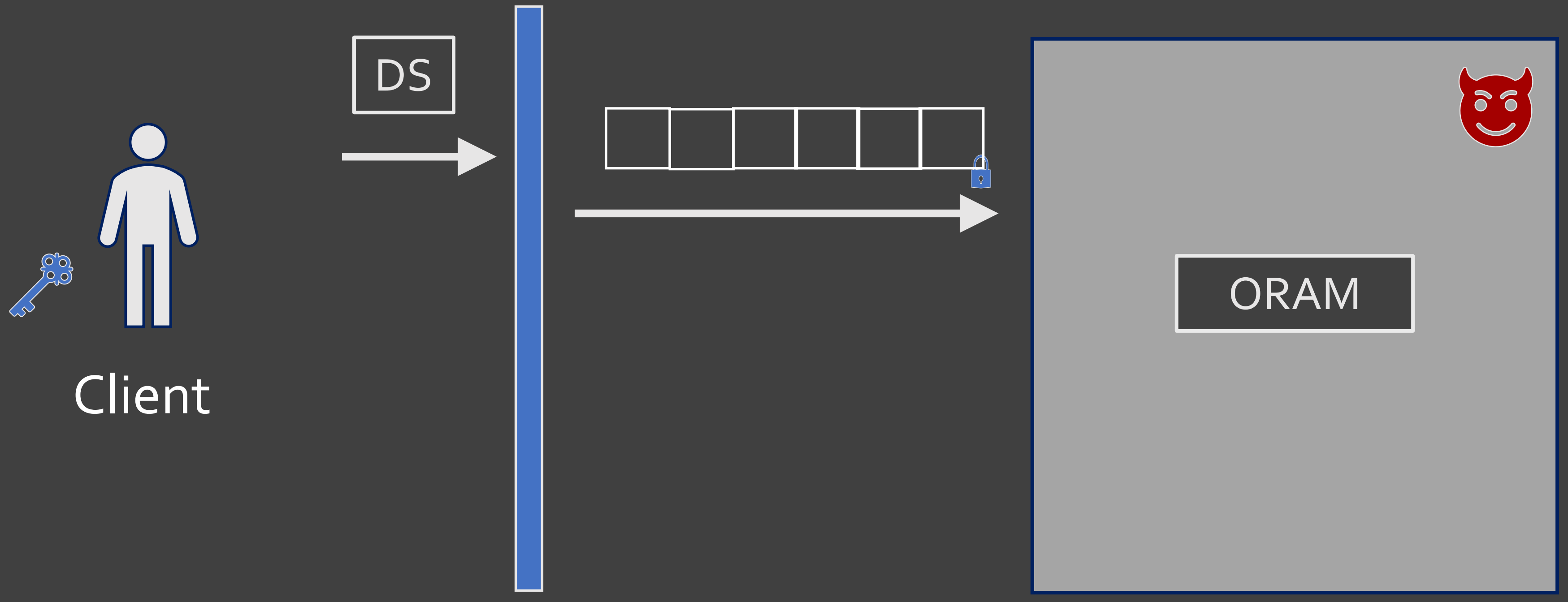
# Query Equality Suppression

- Black-box ORAM simulation



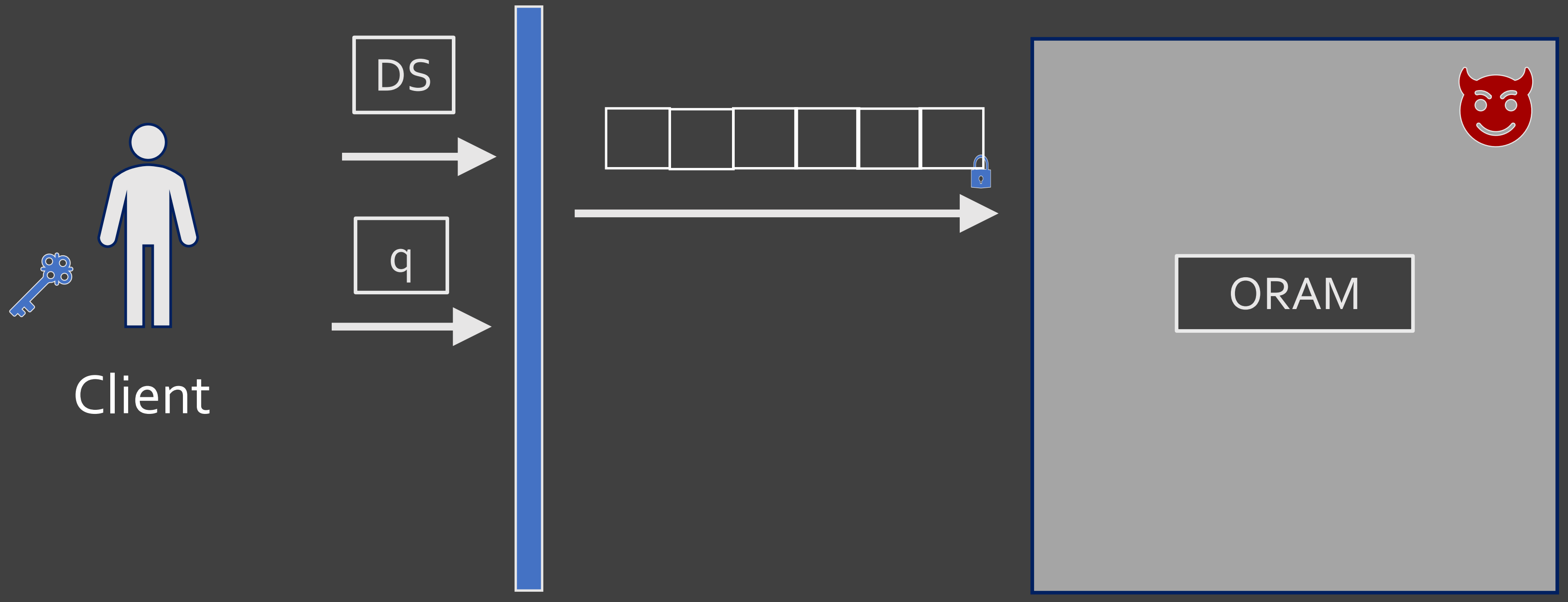
# Query Equality Suppression

- Black-box ORAM simulation



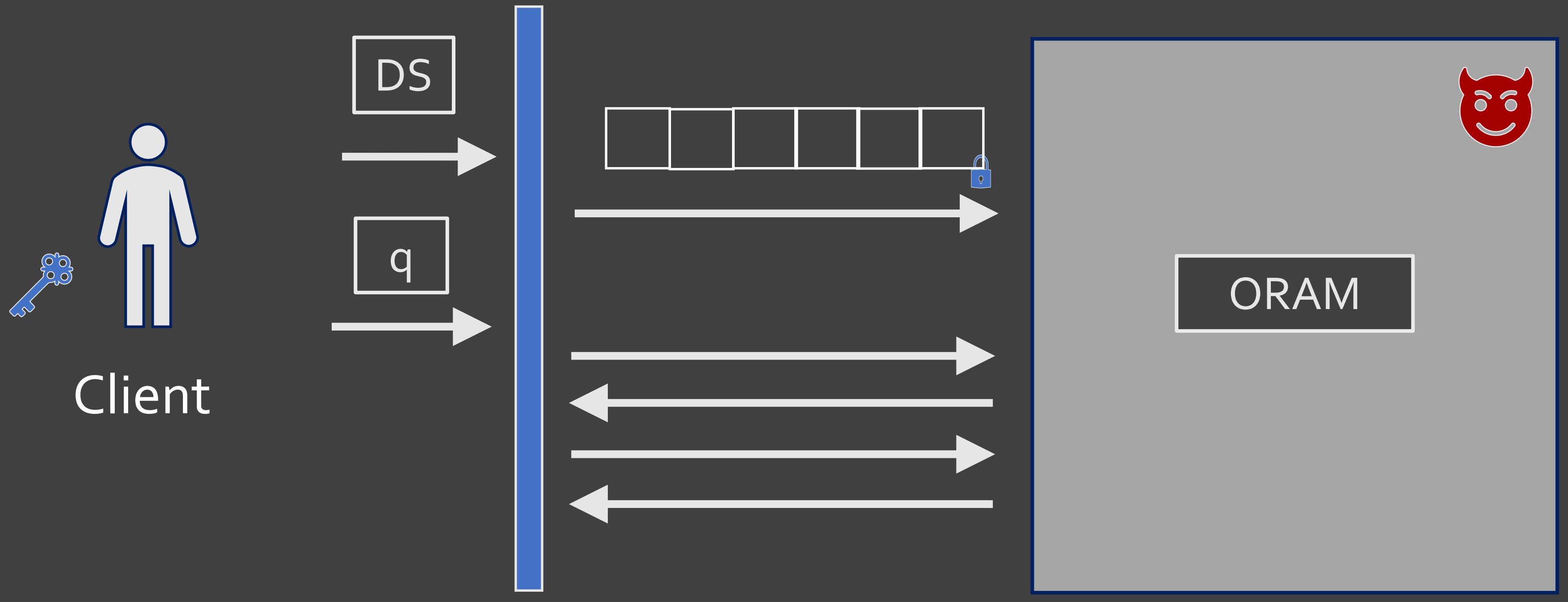
# Query Equality Suppression

- Black-box ORAM simulation



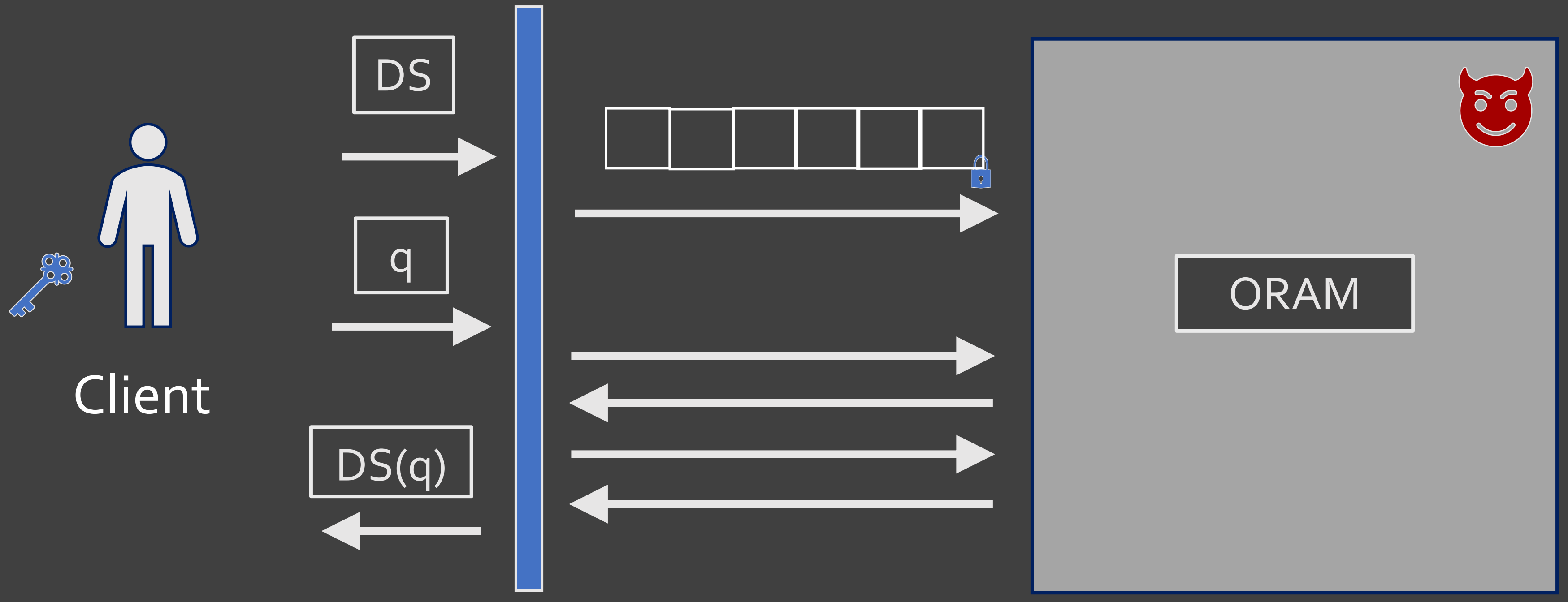
# Query Equality Suppression

- Black-box ORAM simulation



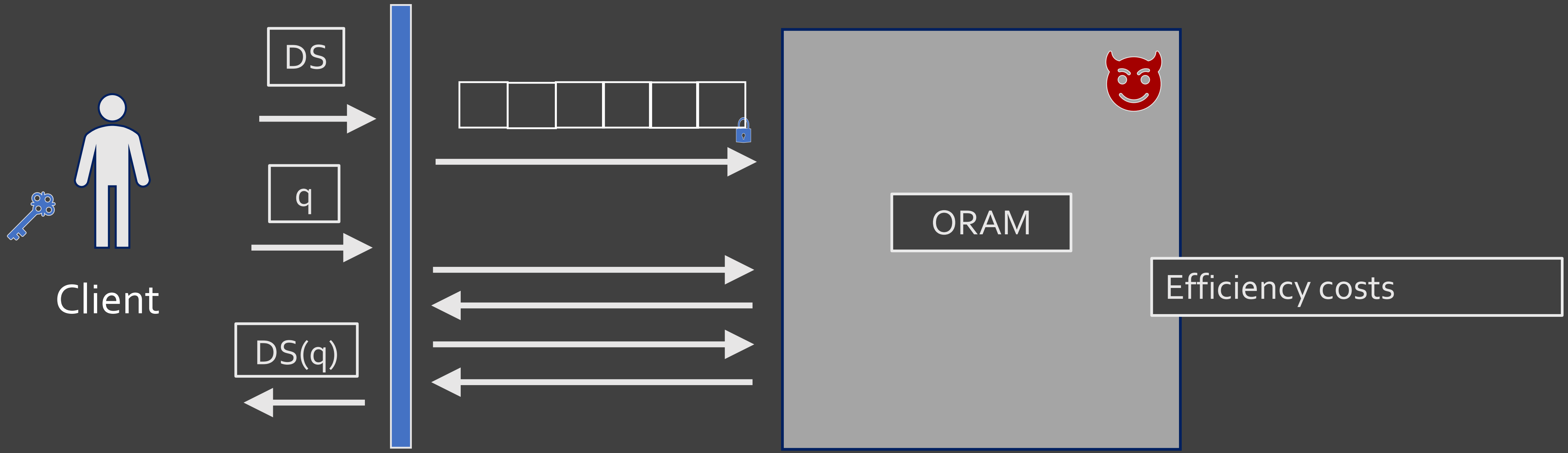
# Query Equality Suppression

- Black-box ORAM simulation



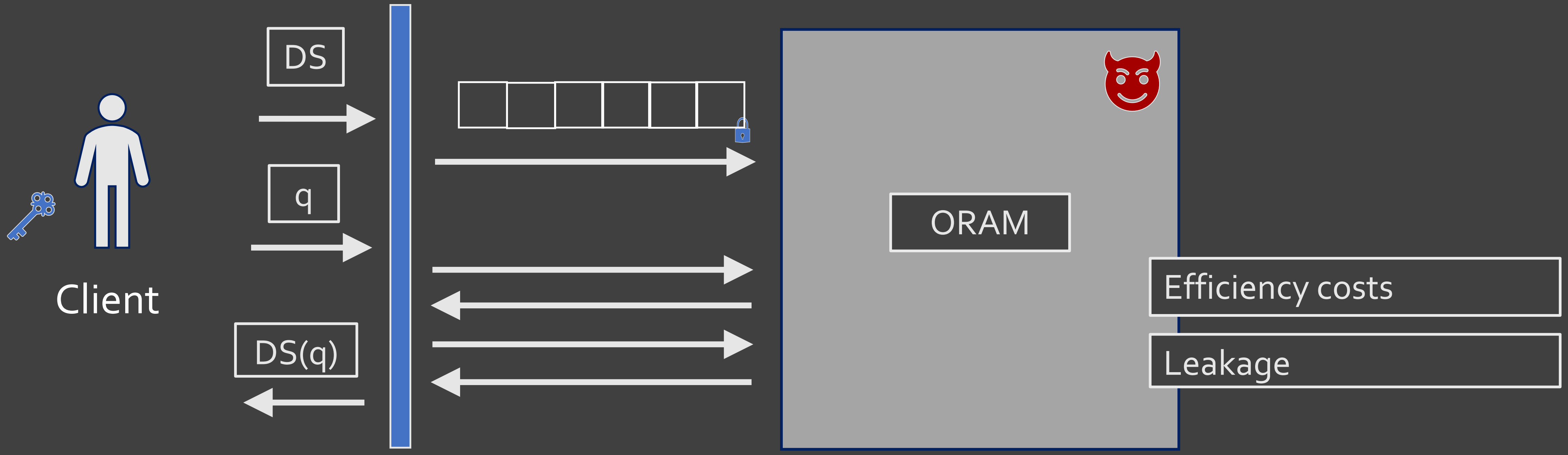
# Query Equality Suppression

- Black-box ORAM simulation



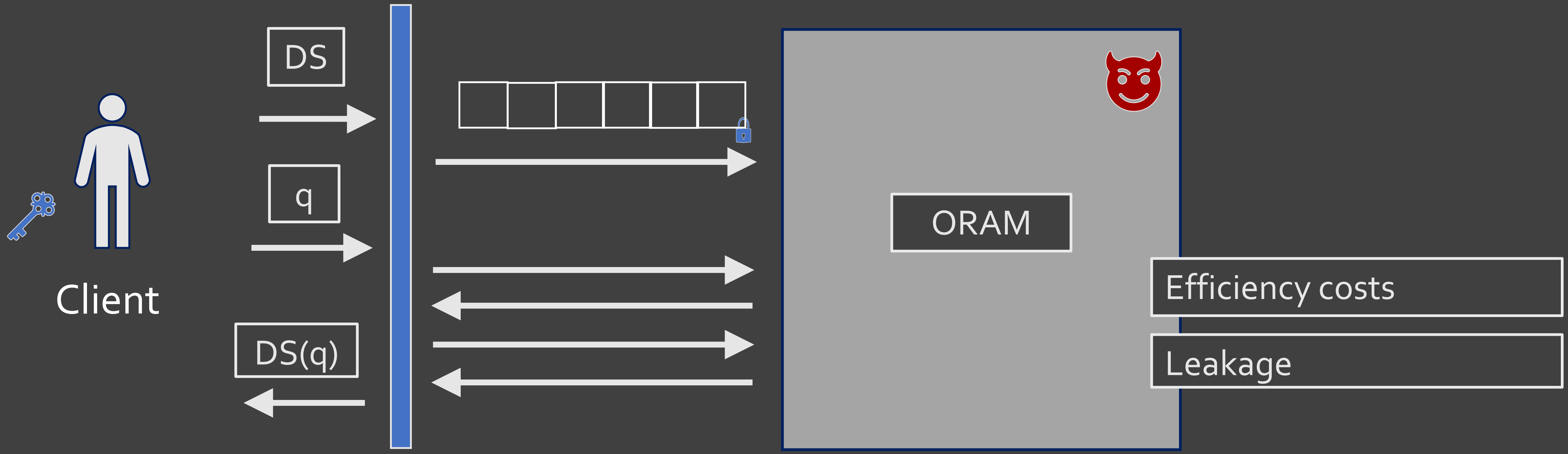
# Query Equality Suppression

- Black-box ORAM simulation



# Query Equality Suppression

- Black-box ORAM simulation



- Custom-made Oblivious Data Structures [WNL+14]

Can we suppress query equality for general data structures more efficiently?

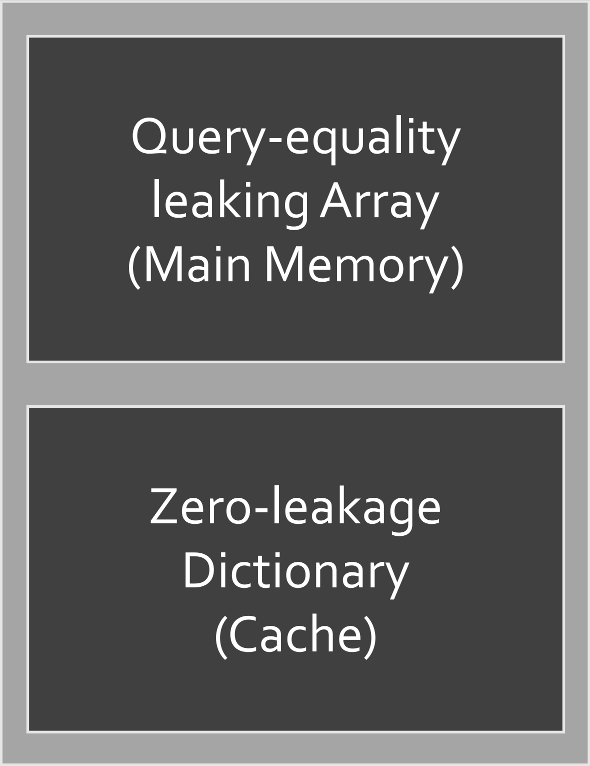
# Query Equality Suppression Framework [KMO18]

# Query Equality Suppression Framework [KMO18]

Square-root ORAM [GO96]

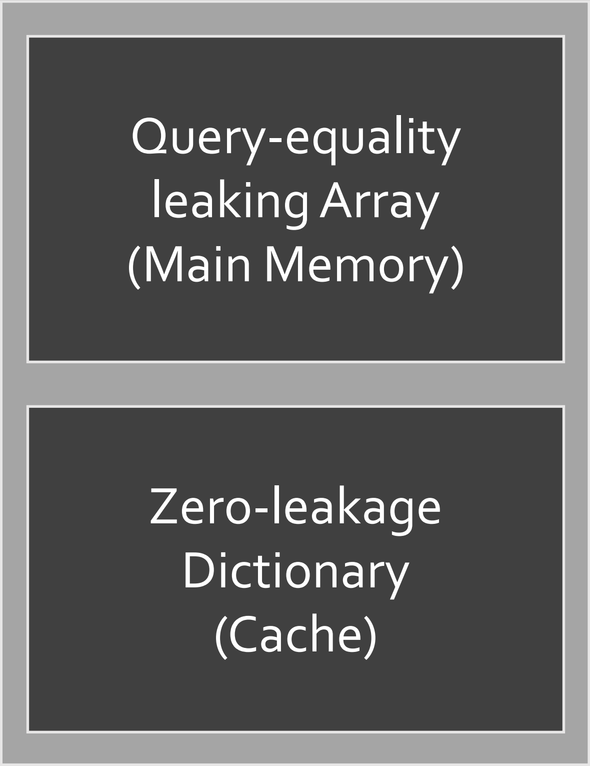
# Query Equality Suppression Framework [KMO18]

## Square-root ORAM [GO96]


Query-equality leaking Array (Main Memory)
Zero-leakage Dictionary (Cache)

# Query Equality Suppression Framework [KMO18]

Square-root ORAM [GO96]



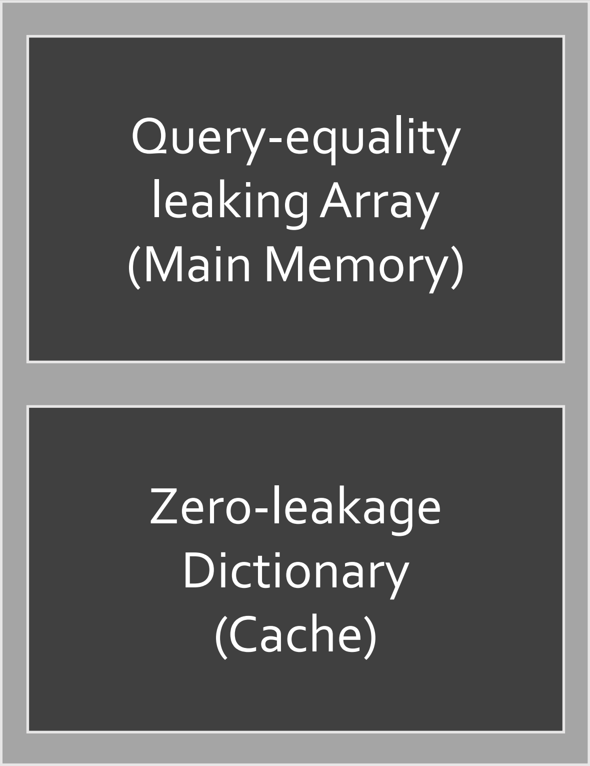
Query-equality leaking Array (Main Memory)

Zero-leakage Dictionary (Cache)

Can be viewed as leakage suppression

# Query Equality Suppression Framework [KMO18]

Square-root ORAM [GO96]

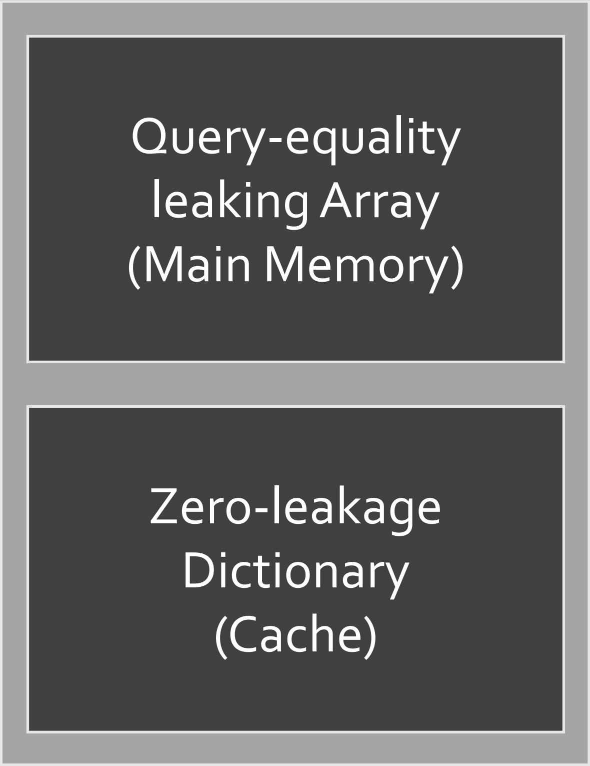

Query-equality leaking Array (Main Memory)
Zero-leakage Dictionary (Cache)

Can be viewed as leakage suppression

Can be generalized to more complex data structures and STE schemes

# Query Equality Suppression Framework [KMO18]

## Square-root ORAM [GO96]



Query-equality leaking Array (Main Memory)

Zero-leakage Dictionary (Cache)

Can be viewed as leakage suppression

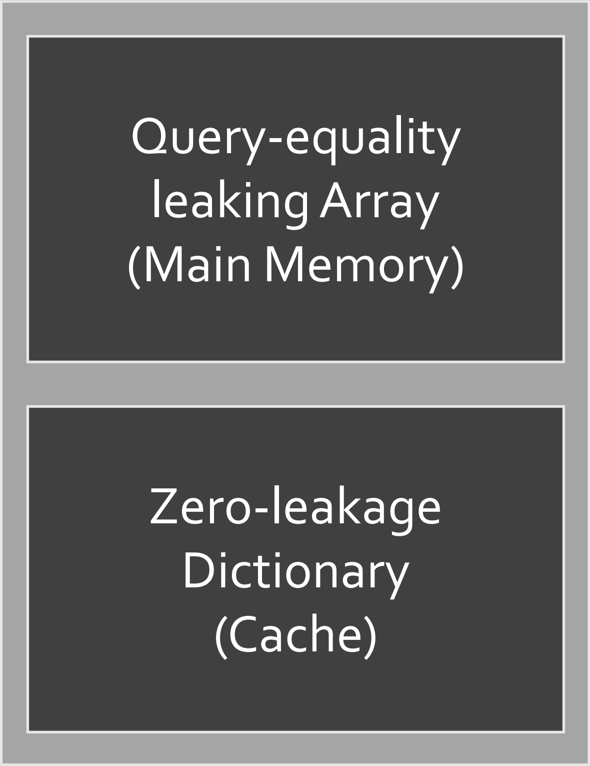
More efficient than black-box ORAM simulation

Can be generalized to more complex data structures and STE schemes

As efficient as custom-made oblivious data structures

# Query Equality Suppression Framework [KMO18]

## Square-root ORAM [GO96]



Query-equality leaking Array (Main Memory)

Zero-leakage Dictionary (Cache)

Can be viewed as leakage suppression

More efficient than black-box ORAM simulation

Can be generalized to more complex data structures and STE schemes

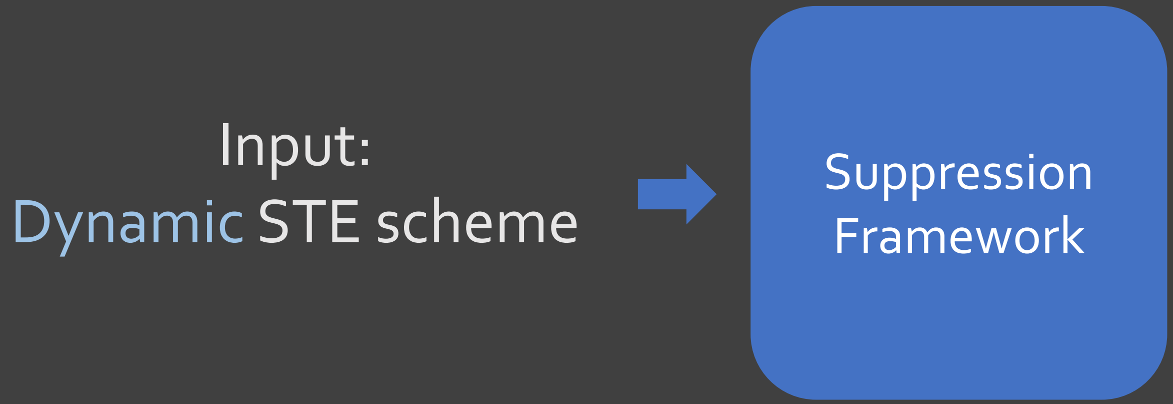
As efficient as custom-made oblivious data structures

The framework only produces static schemes

# Query Equality Suppression Framework [KMO18]

Suppression Framework

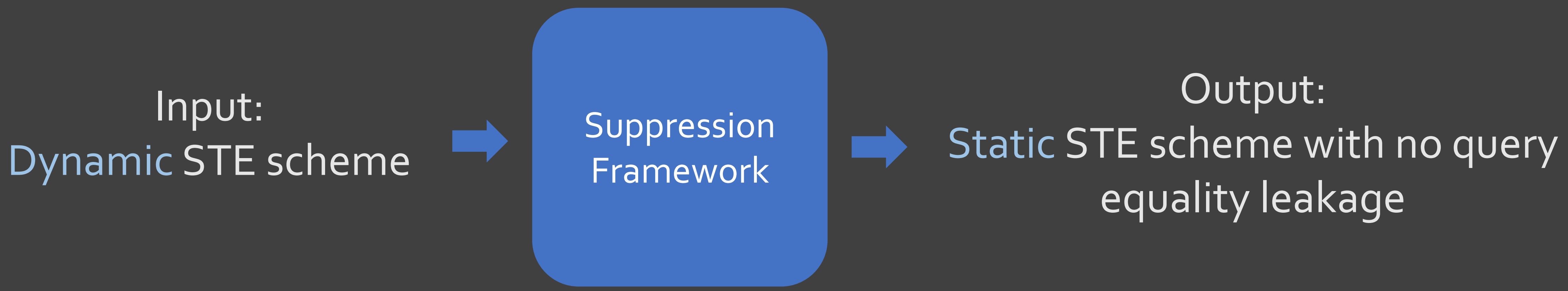
# Query Equality Suppression Framework [KMO18]



Input:
Dynamic STE scheme

Suppression Framework

# Query Equality Suppression Framework [KMO18]



Input:
Dynamic STE scheme

Suppression Framework

Output:
Static STE scheme with no query equality leakage

Can query equality leakage be suppressed in the dynamic setting?

Is it possible to create a dynamic query equality suppression framework?

# Dynamic Leakage Suppression: Challenges

# Dynamic Leakage Suppression: Challenges

Operation equality across all dynamic operations (Add, Edit, Delete) must be suppressed

# Dynamic Leakage Suppression: Challenges

Operation equality across all dynamic operations (Add, Edit, Delete) must be suppressed

Correlated leakage in the dynamic setting

# Dynamic Leakage Suppression: Challenges

Operation equality across all dynamic operations (Add, Edit, Delete) must be suppressed

Correlated leakage in the dynamic setting

# Dynamic Leakage Suppression: Challenges

Operation equality across all dynamic operations (Add, Edit, Delete) must be suppressed

Correlated leakage in the dynamic setting

- Operation identity leakage
- Volume leakage

# Dynamic Leakage Suppression: Challenges

Operation equality across all dynamic operations (Add, Edit, Delete) must be suppressed

Correlated leakage in the dynamic setting

- Operation identity leakage
- Volume leakage
    - Input volume-hiding schemes, volume leakage already suppressed

# Dynamic Leakage Suppression: Challenges

Operation equality across all dynamic operations (Add, Edit, Delete) must be suppressed

Correlated leakage in the dynamic setting

- Operation identity leakage
- Volume leakage
  - Input volume-hiding schemes, volume leakage already suppressed
  - Many volume-hiding schemes have limited dynamicity and must be ‘upgraded’ using our framework

# The Dynamic Suppression Framework

# The Dynamic Suppression Framework

# The Dynamic Suppression Framework

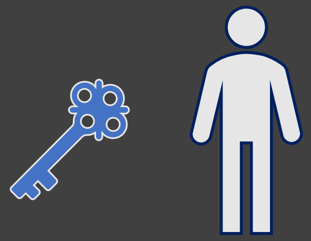
L1 → (v11, v12, v13)

L2 → (v21, v22)

L3 → (v31, v32, v33)

L4 → (v41)

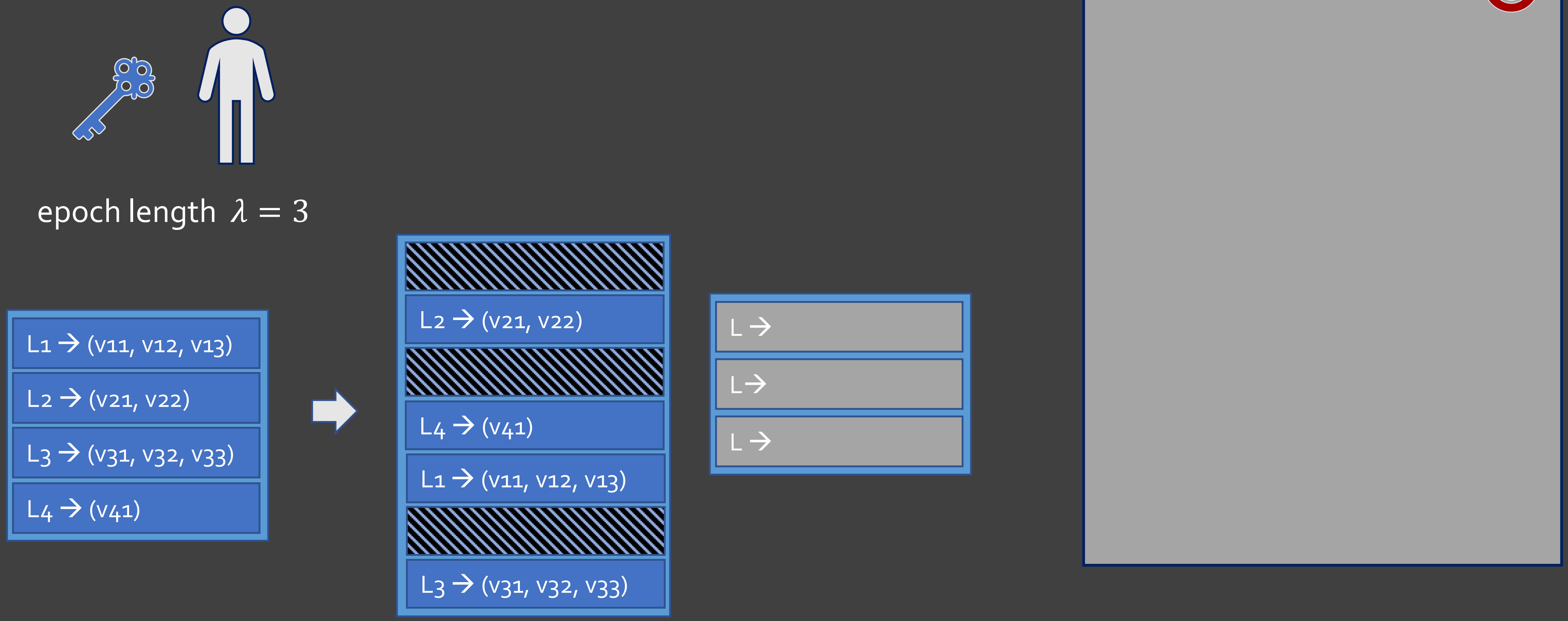
# The Dynamic Suppression Framework



epoch length $\lambda = 3$

- L1 → (v11, v12, v13)
- L2 → (v21, v22)
- L3 → (v31, v32, v33)
- L4 → (v41)

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

L1 → (v11, v12, v13)

L2 → (v21, v22)

L3 → (v31, v32, v33)

L4 → (v41)

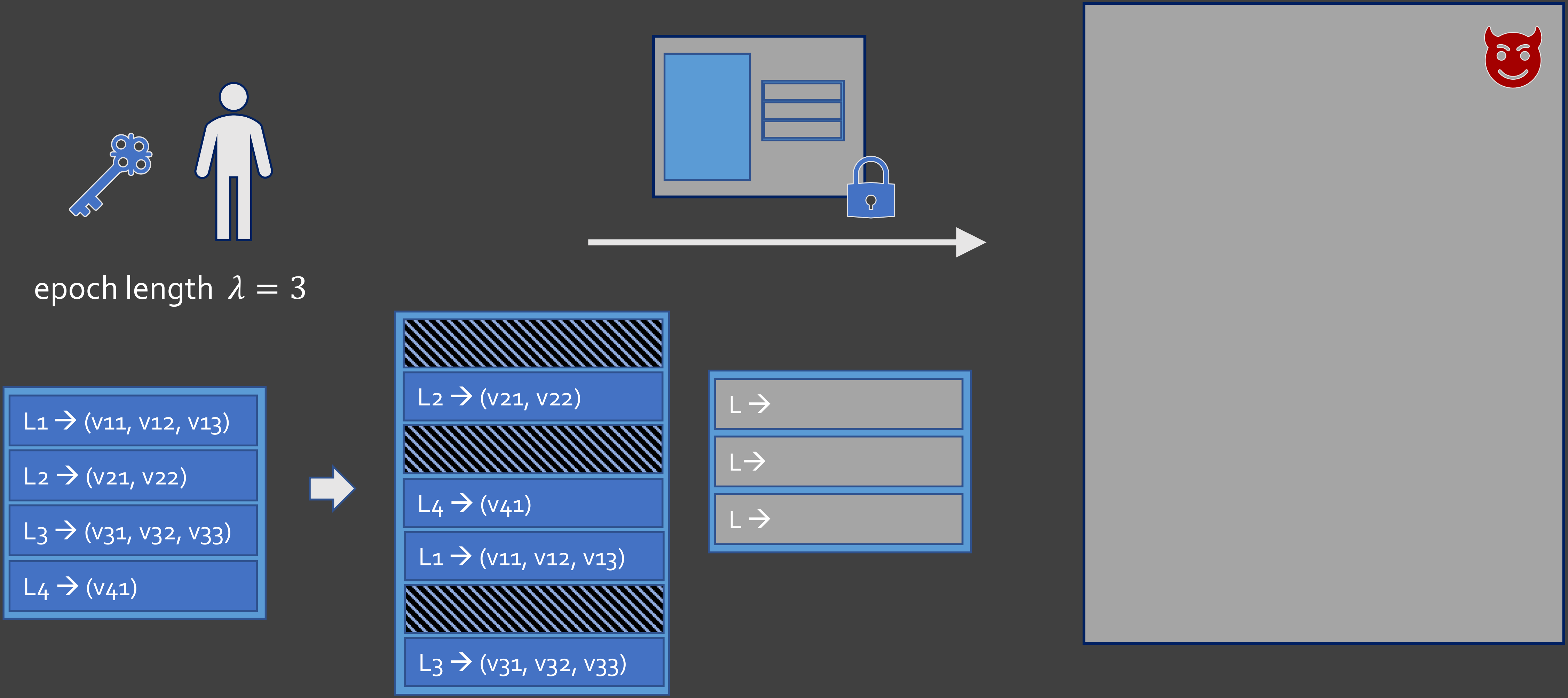
L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

L3 → (v31, v32, v33)

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

L1 → (v11, v12, v13)

L2 → (v21, v22)

L3 → (v31, v32, v33)

L4 → (v41)

▨

L2 → (v21, v22)

▨

L4 → (v41)

L1 → (v11, v12, v13)

▨

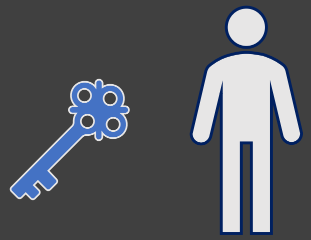
L3 → (v31, v32, v33)

L →

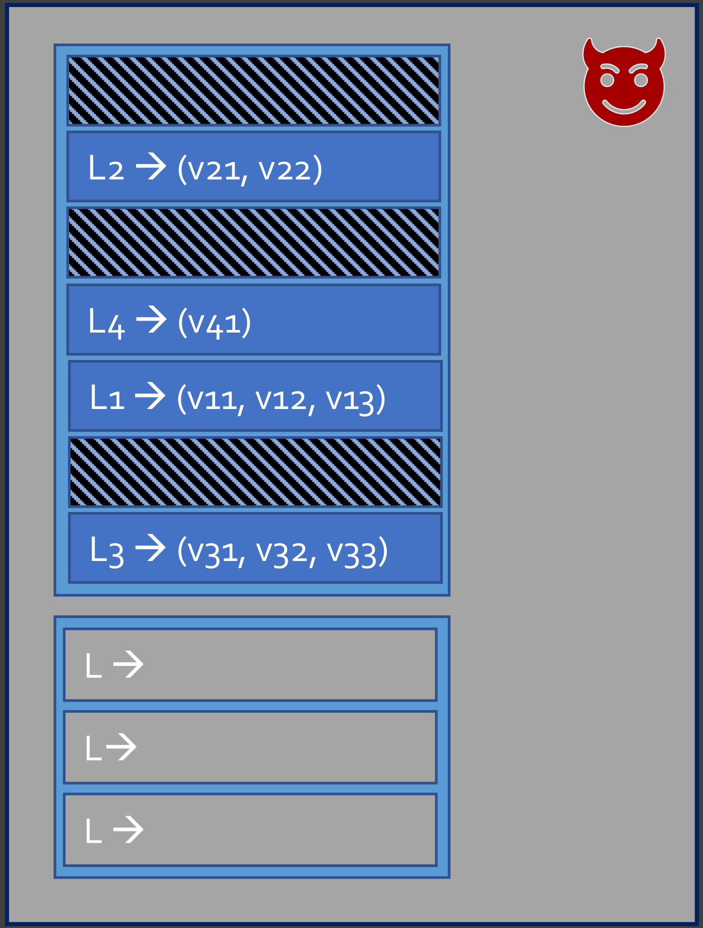
L →

L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$



L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

L3 → (v31, v32, v33)

L →

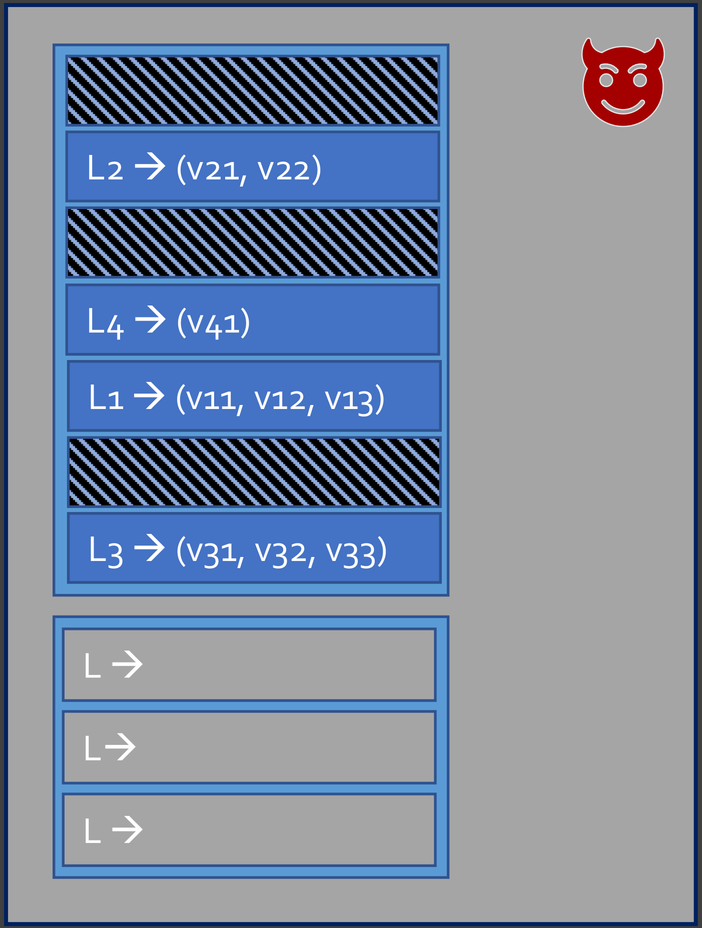
L →

L →

# The Dynamic Suppression Framework

epoch length $\lambda = 3$

Query L1



L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

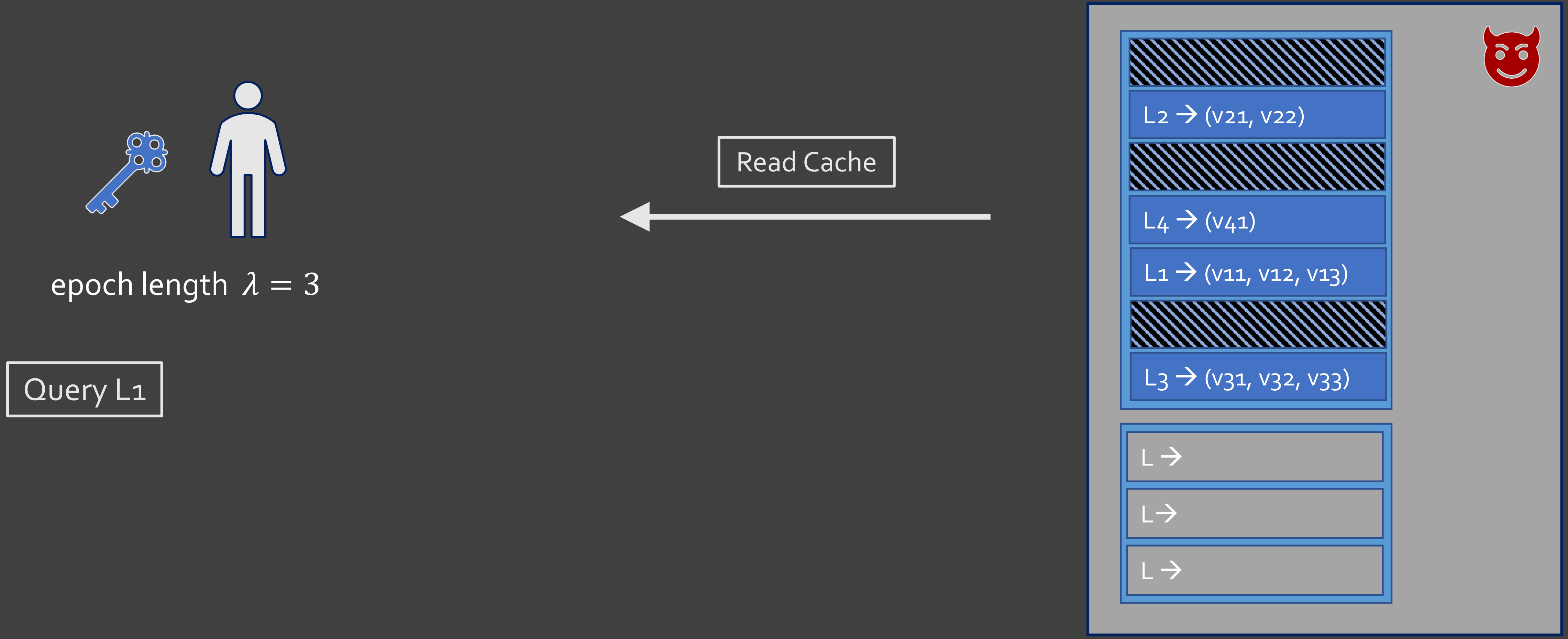
L3 → (v31, v32, v33)

L →

L→

L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

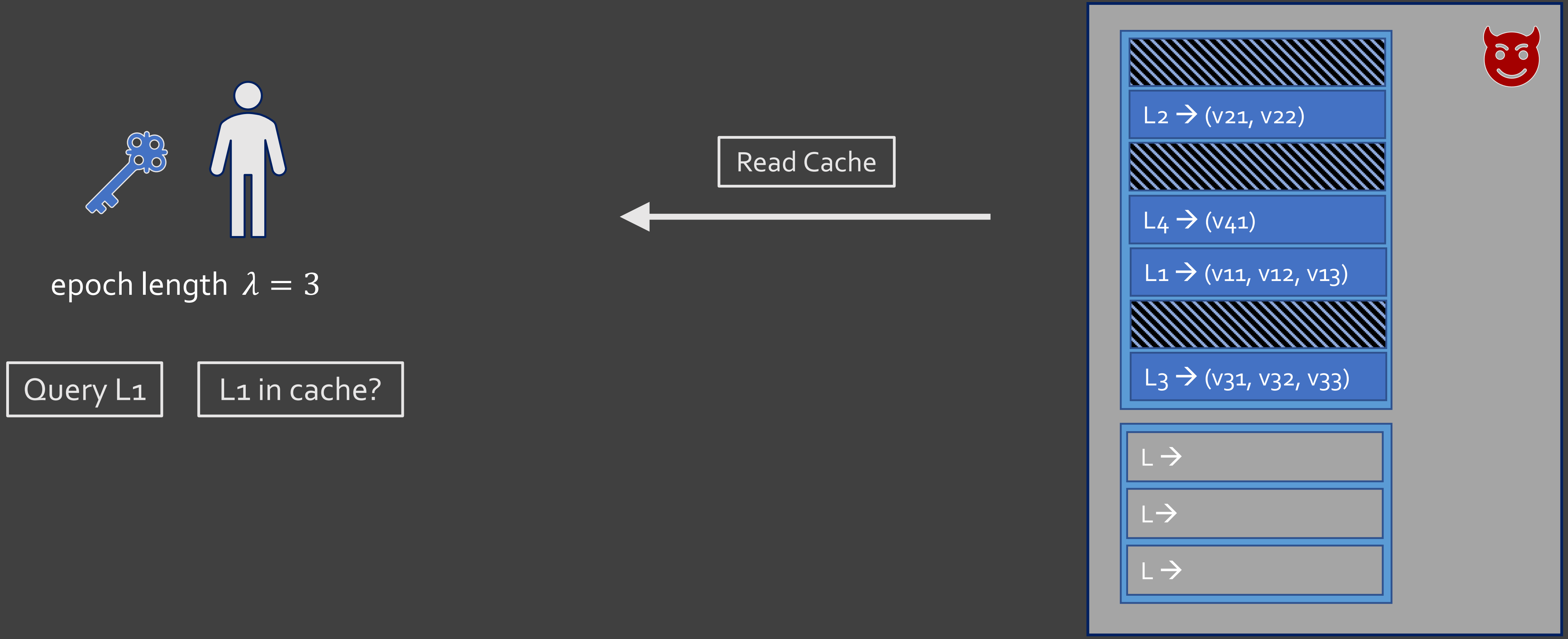
Query L1

Read Cache

| |
|---|
| ▨ |
| L2 → (v21, v22) |
| ▨ |
| L4 → (v41) |
| L1 → (v11, v12, v13) |
| ▨ |
| L3 → (v31, v32, v33) |

| |
|---|
| L → |
| L→ |
| L → |

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

L1 in cache?

Read Cache

L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

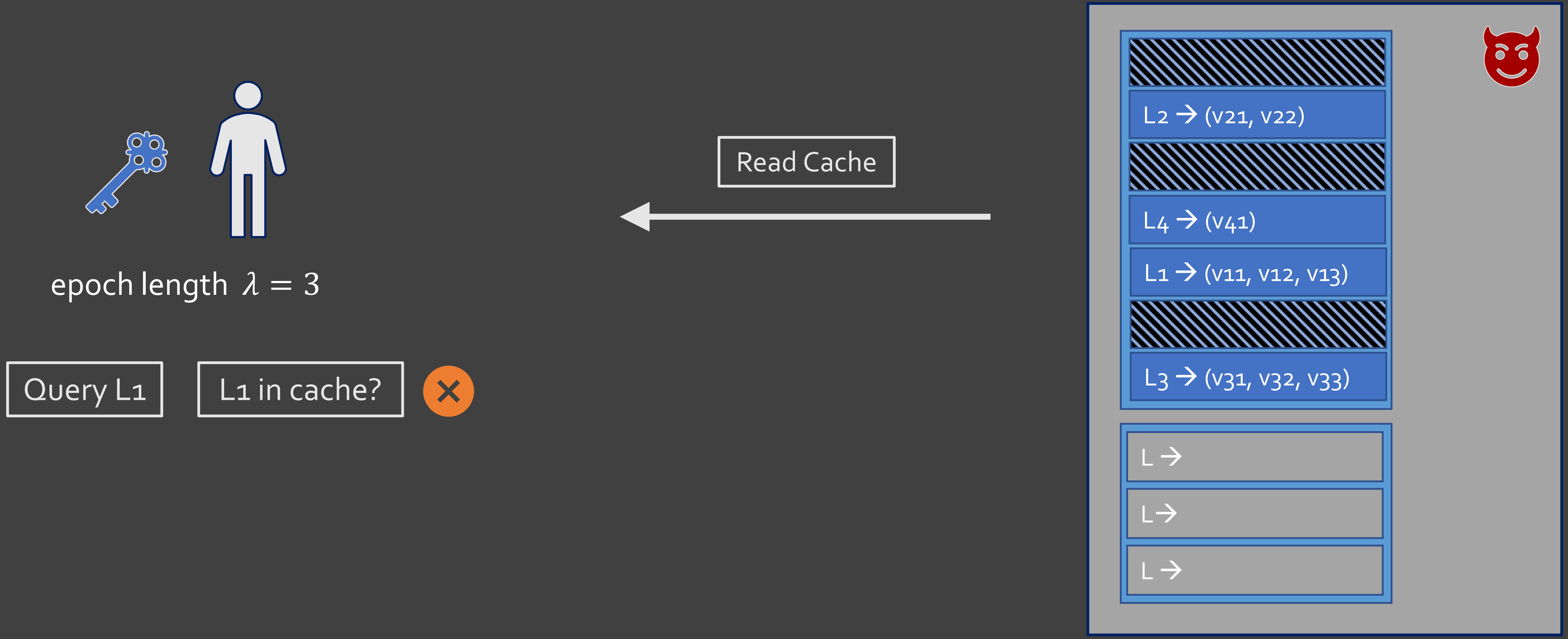
L3 → (v31, v32, v33)

L →

L →

L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

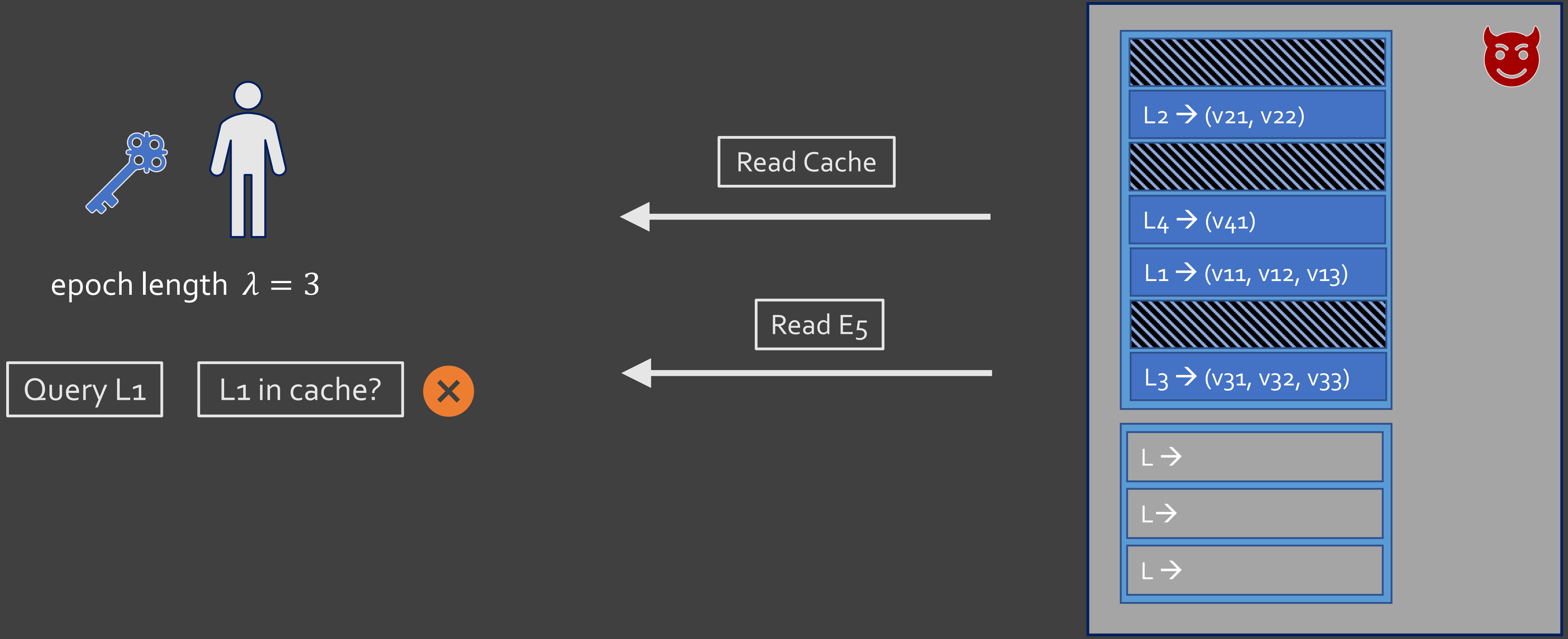
L1 in cache? ✗

Read Cache

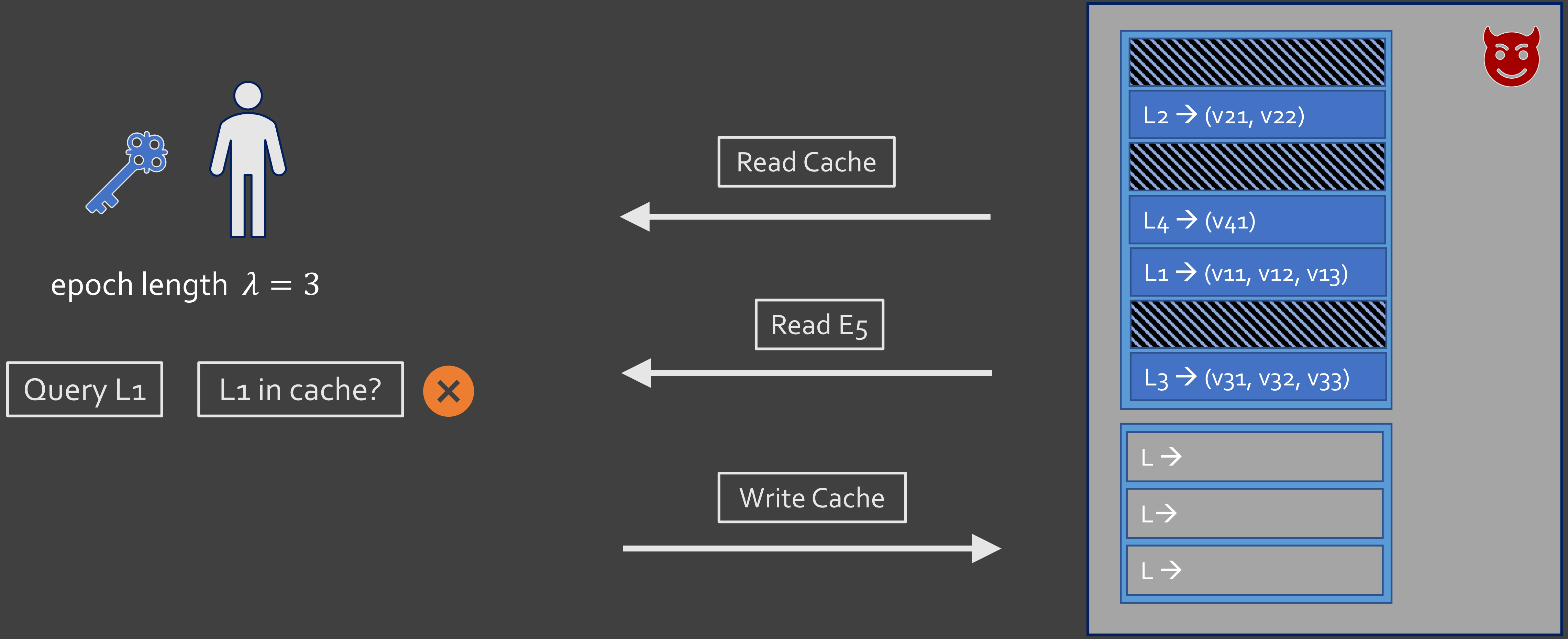
| |
|---|
| ▨ |
| L2 → (v21, v22) |
| ▨ |
| L4 → (v41) |
| L1 → (v11, v12, v13) |
| ▨ |
| L3 → (v31, v32, v33) |

| |
|---|
| L → |
| L→ |
| L → |

# The Dynamic Suppression Framework



# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

L1 in cache?

Read Cache

Read E5

Write Cache

L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

L3 → (v31, v32, v33)

L →

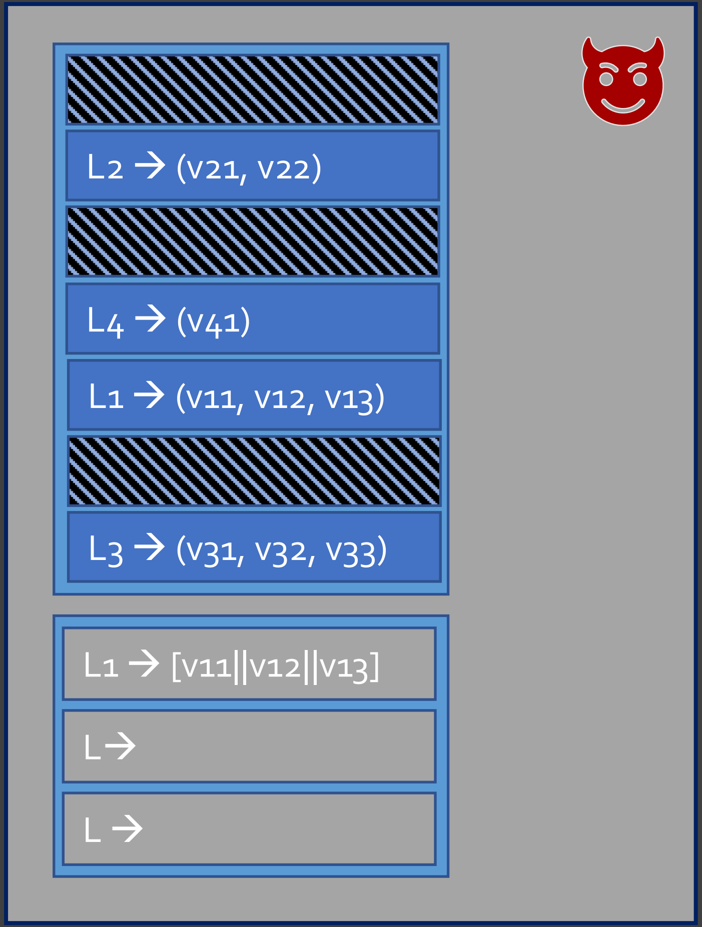
L →

L →

# The Dynamic Suppression Framework

epoch length $\lambda = 3$

Query L1



L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

L →

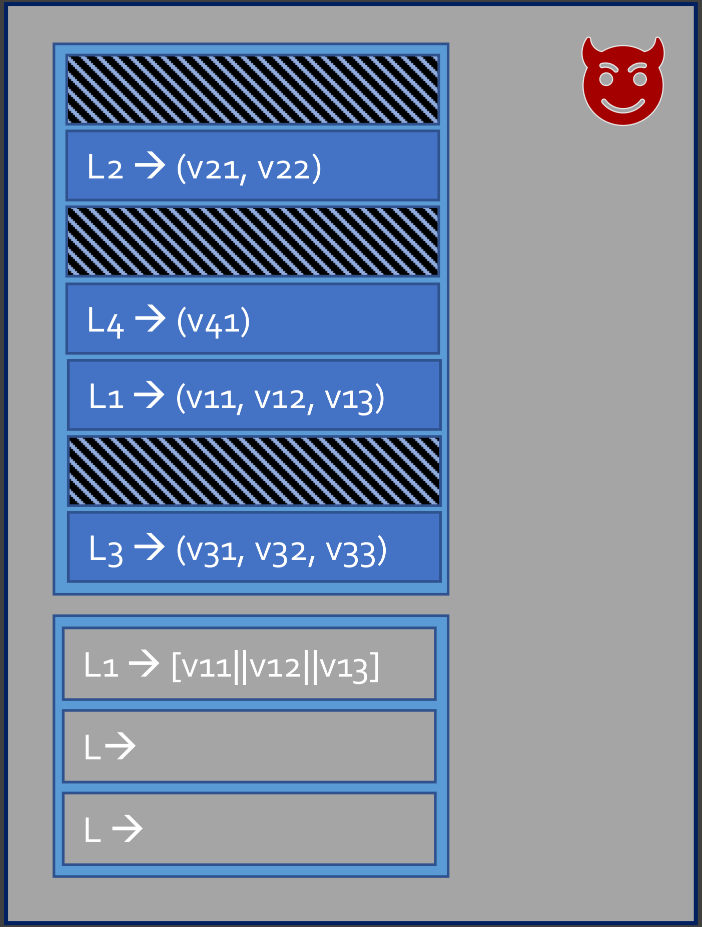
L →

# The Dynamic Suppression Framework

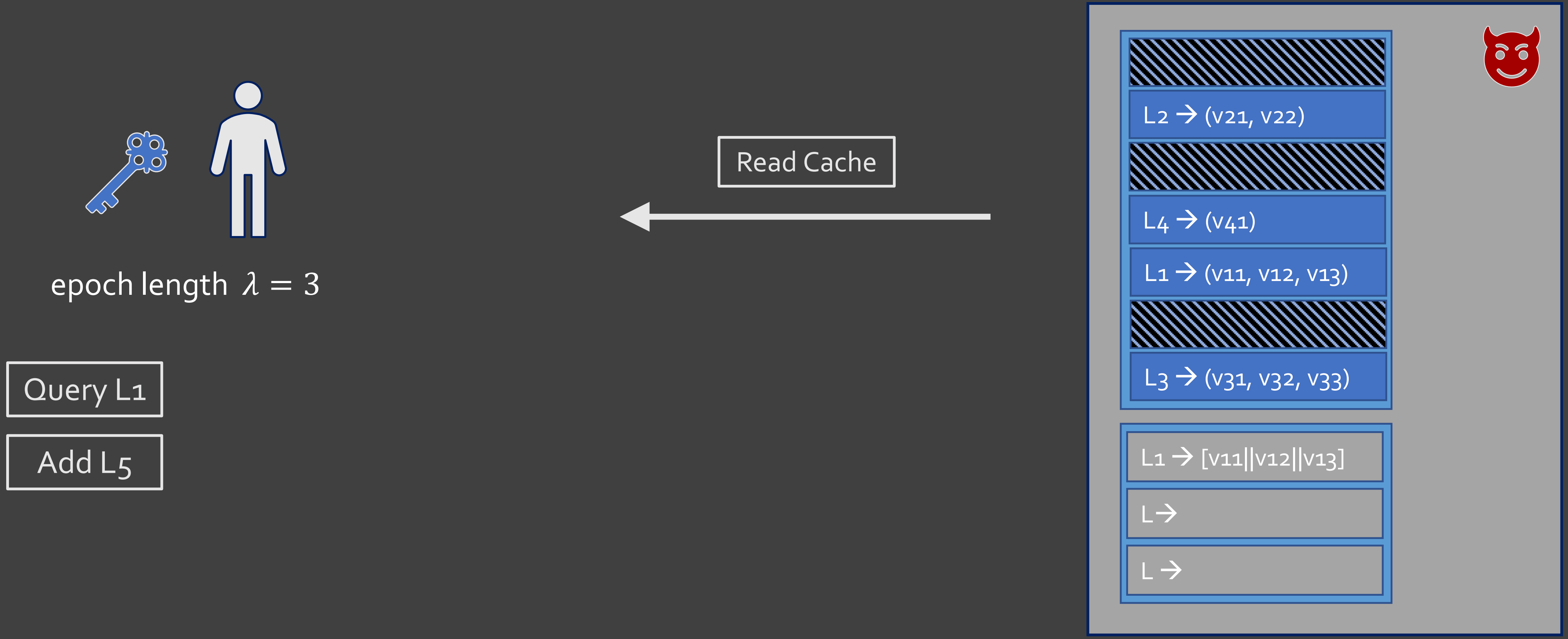
epoch length $\lambda = 3$

Query L1

Add L5


L2 → (v21, v22)
L4 → (v41)
L1 → (v11, v12, v13)
L3 → (v31, v32, v33)
L1 → [v11||v12||v13]
L →
L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5

Read Cache

▨

L2 → (v21, v22)

▨

L4 → (v41)

L1 → (v11, v12, v13)

▨

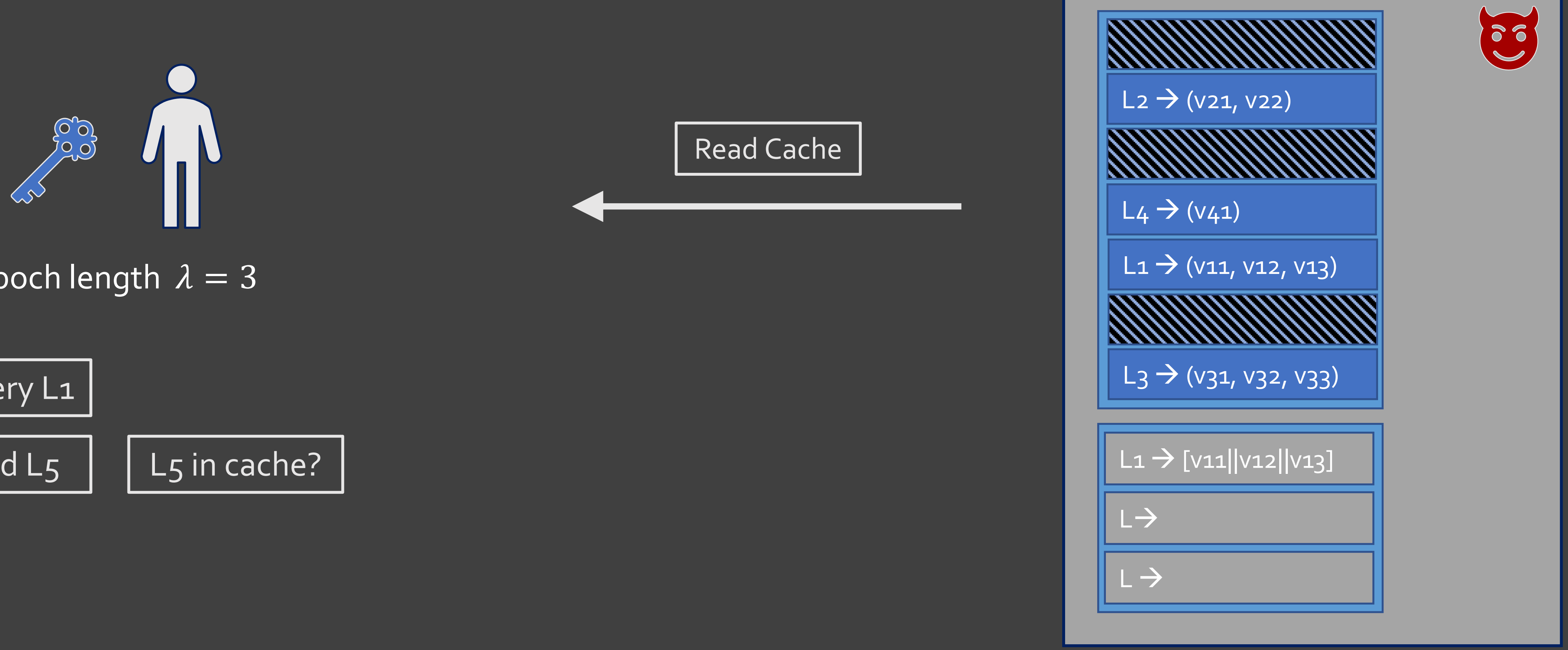
L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

L →

L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5

L5 in cache?

Read Cache

L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

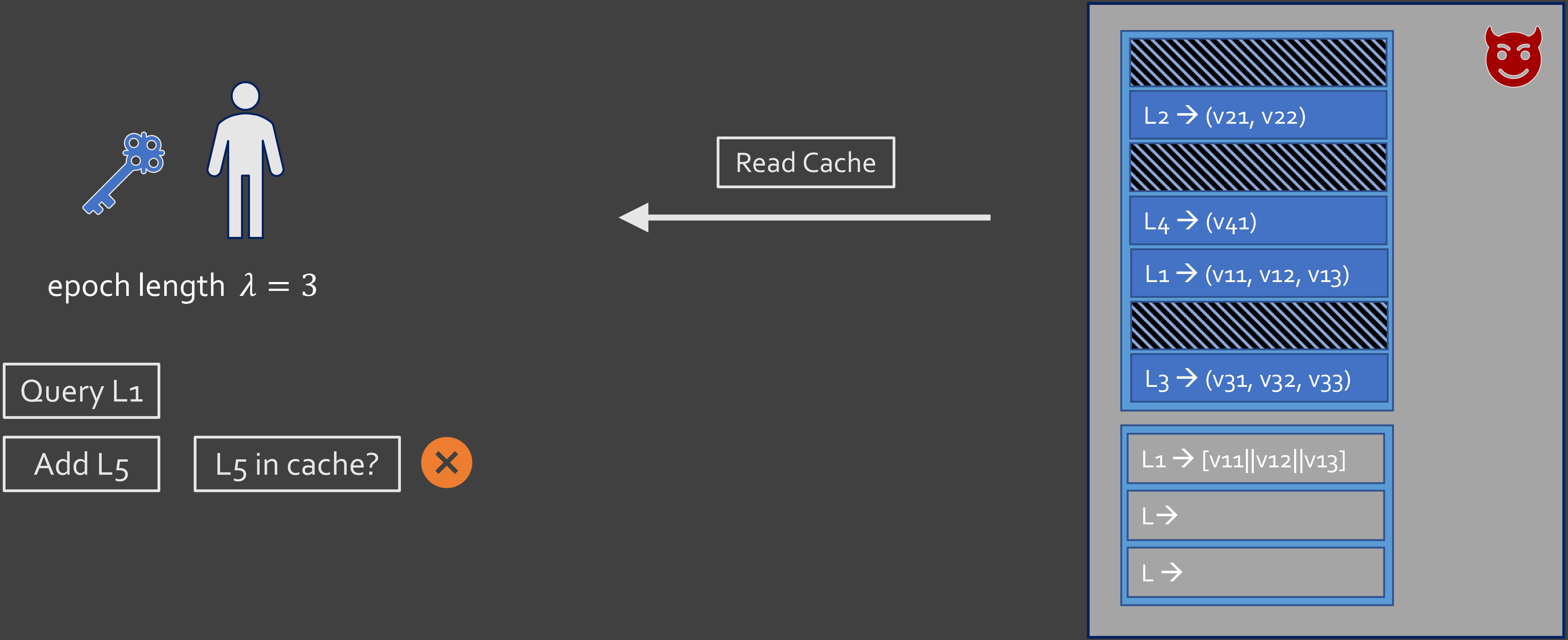
L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

L →

L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5

L5 in cache? ✗

Read Cache

L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

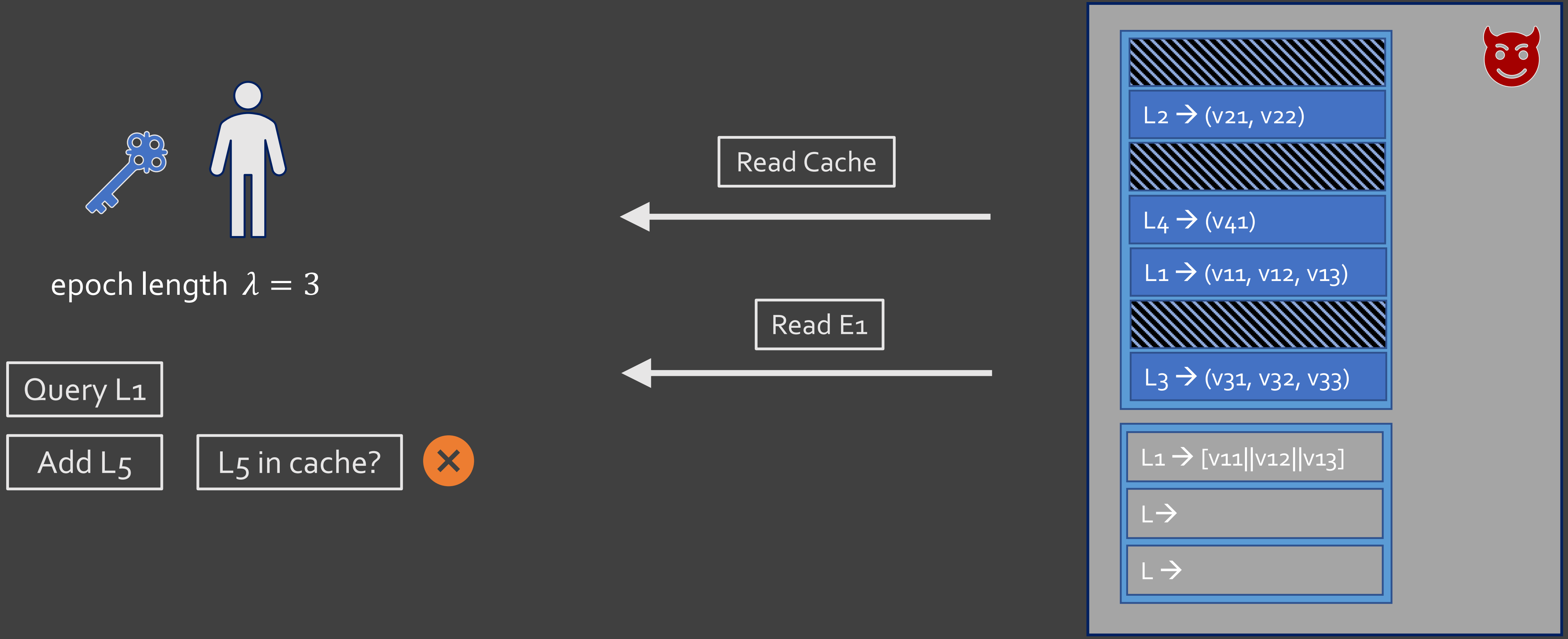
L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

L →

L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5

L5 in cache? ✕

Read Cache

Read E1

L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

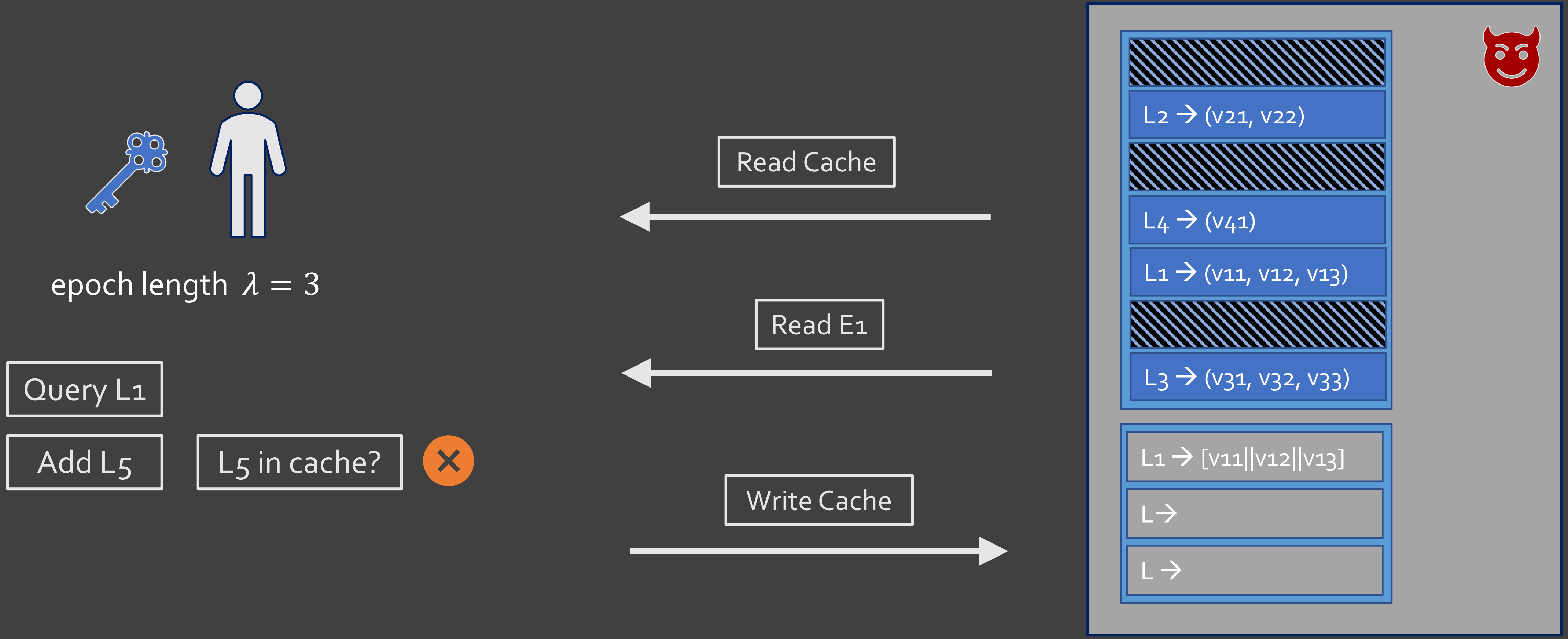
L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

L →

L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5

L5 in cache? ✗

Read Cache

Read E1

Write Cache

L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

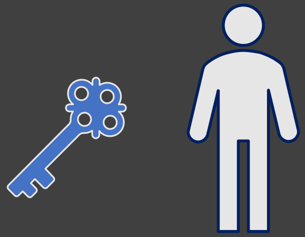
L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

L →

L →

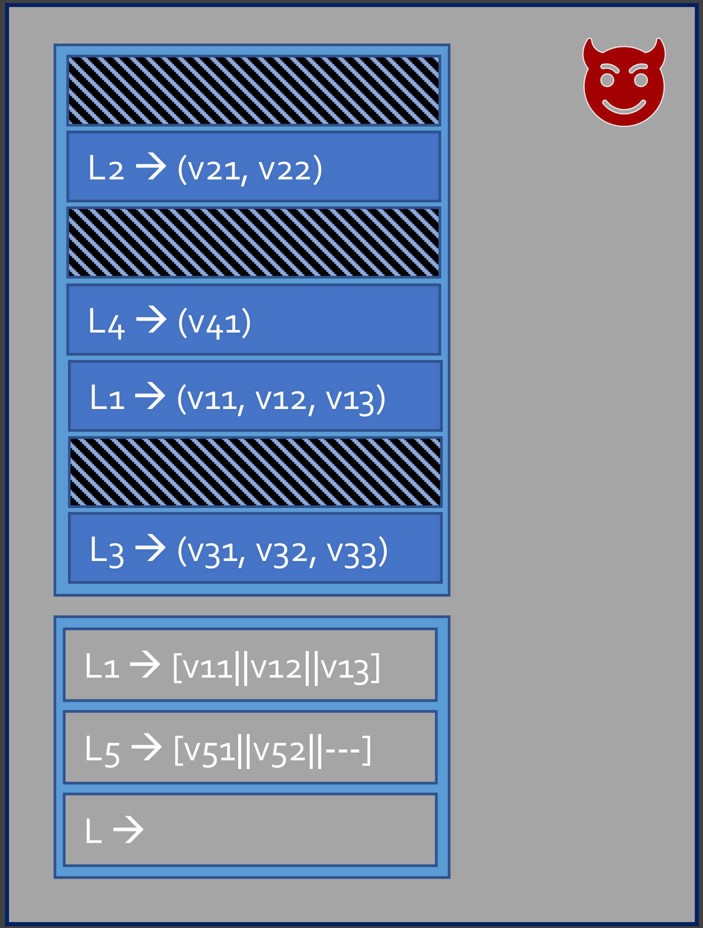
# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5



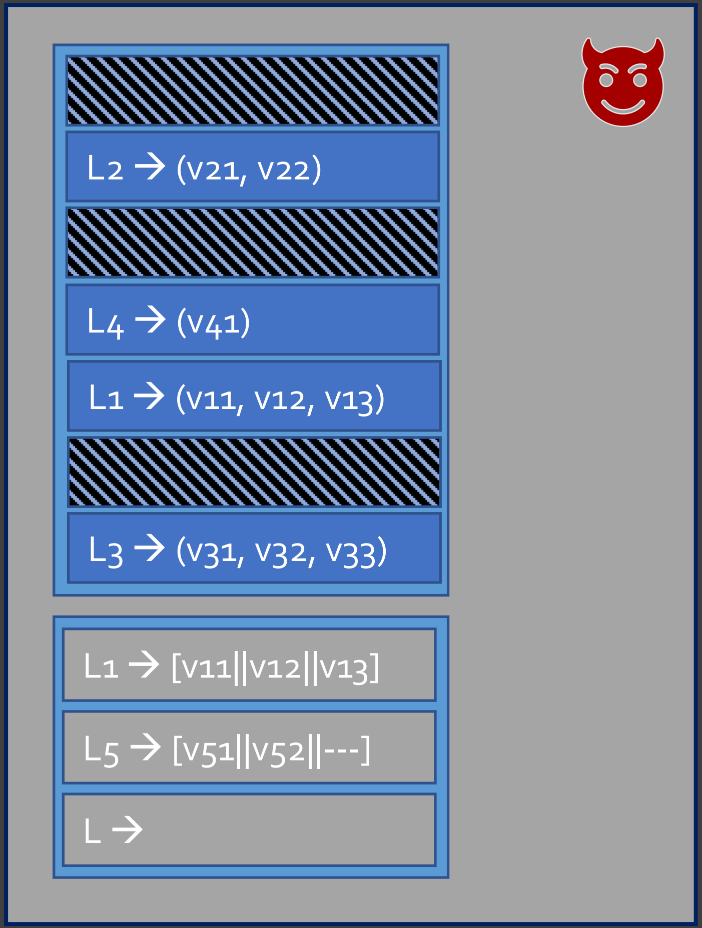
# The Dynamic Suppression Framework

epoch length $\lambda = 3$

Query L1

Add L5

Edit L1



▨

L2 → (v21, v22)

▨

L4 → (v41)

L1 → (v11, v12, v13)

▨

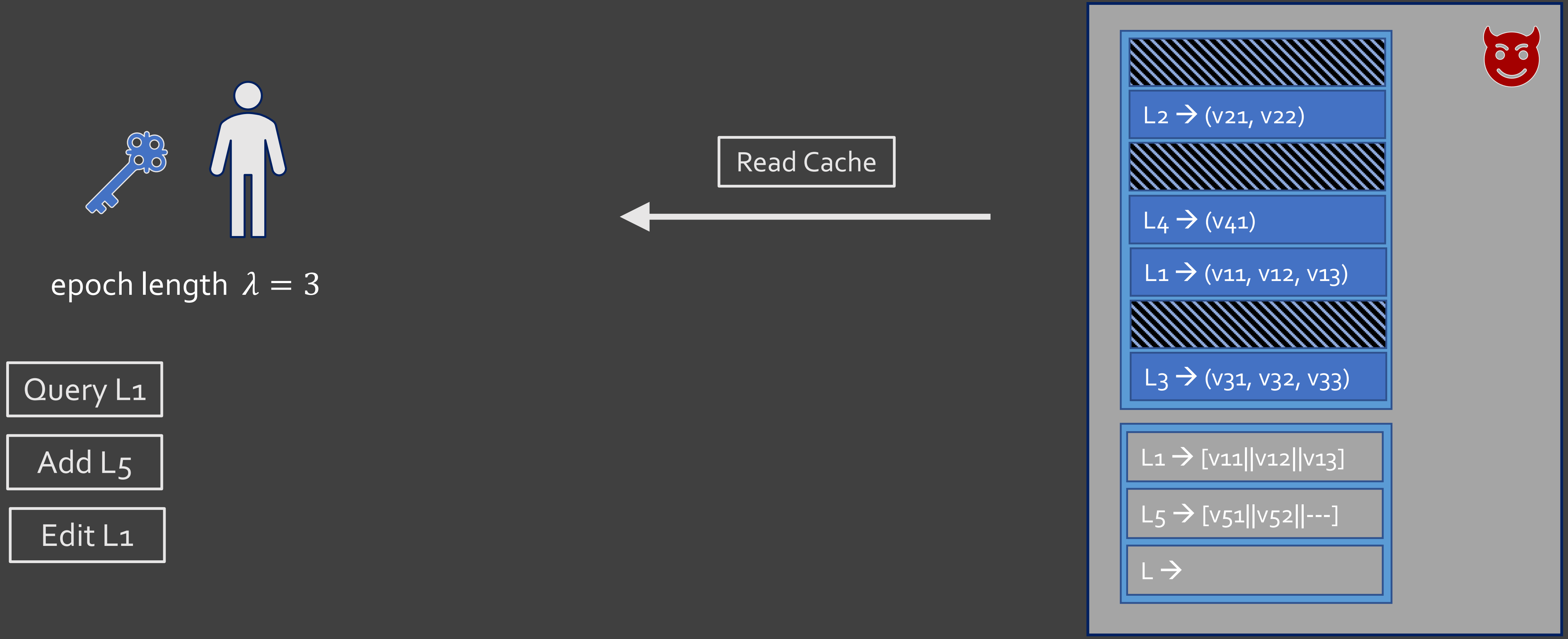
L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

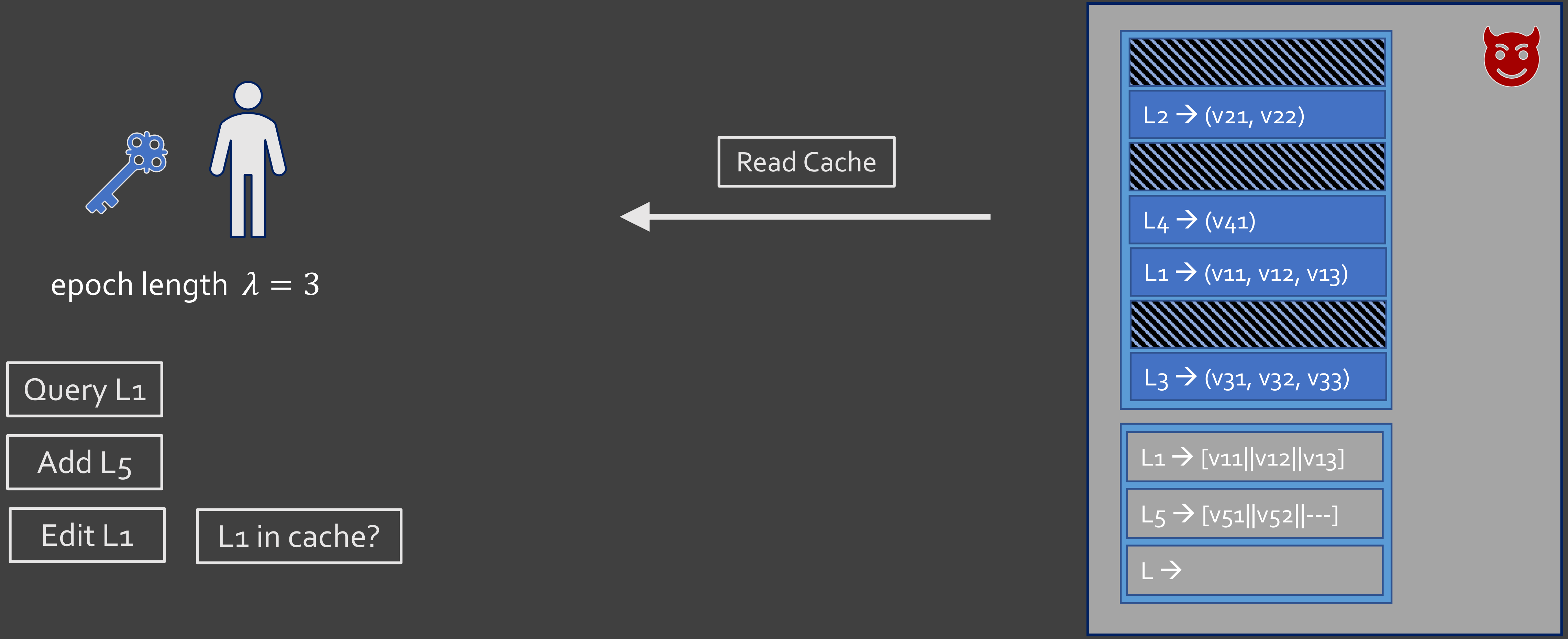
L5 → [v51||v52||---]

L →

# The Dynamic Suppression Framework



# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5

Edit L1

L1 in cache?

Read Cache

L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

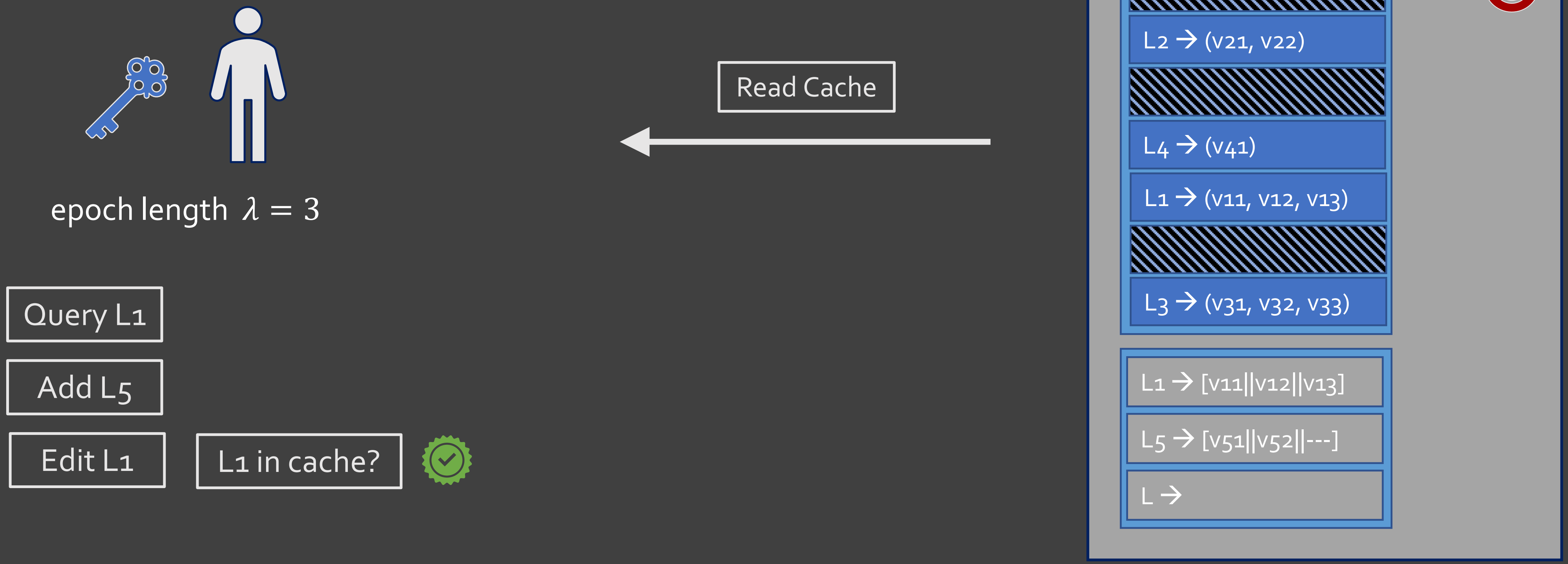
L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

L5 → [v51||v52||---]

L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5

Edit L1

L1 in cache? ✓

Read Cache

L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

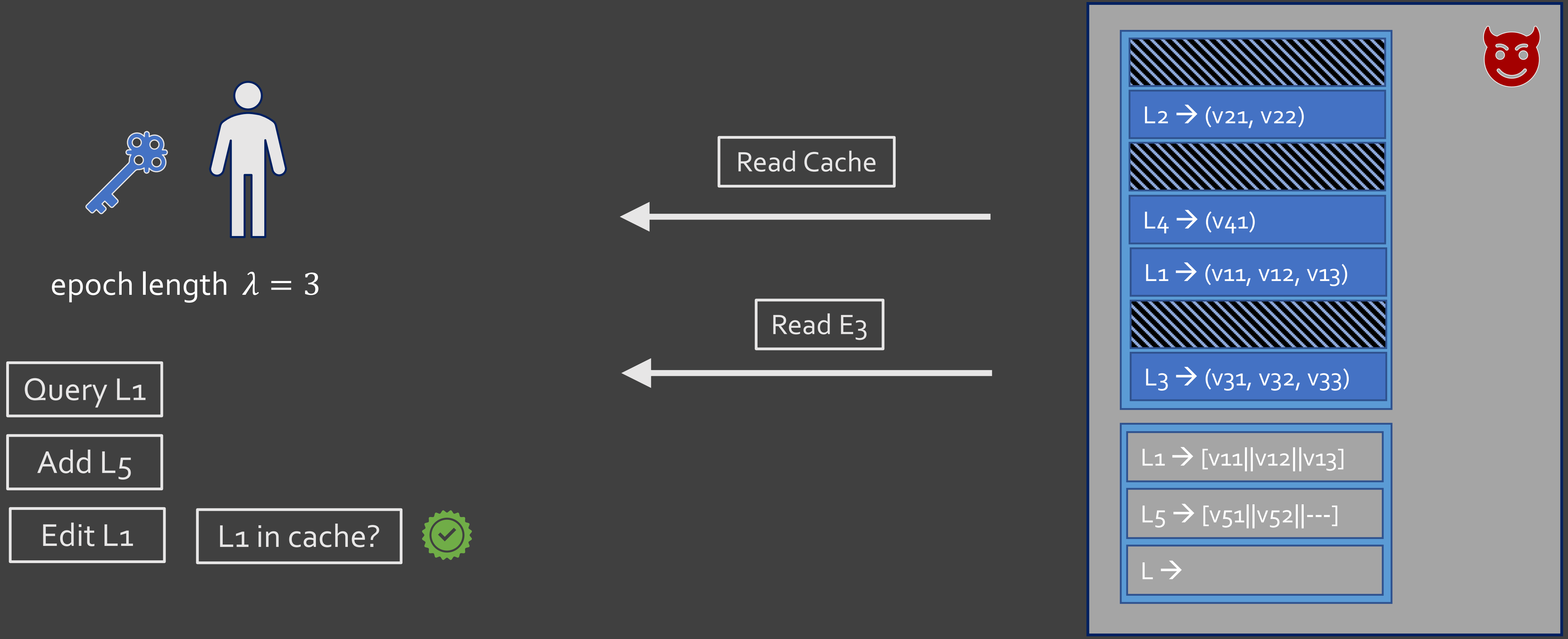
L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

L5 → [v51||v52||---]

L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5

Edit L1

L1 in cache? ✓

Read Cache

Read E3

▨

L2 → (v21, v22)

▨

L4 → (v41)

L1 → (v11, v12, v13)

▨

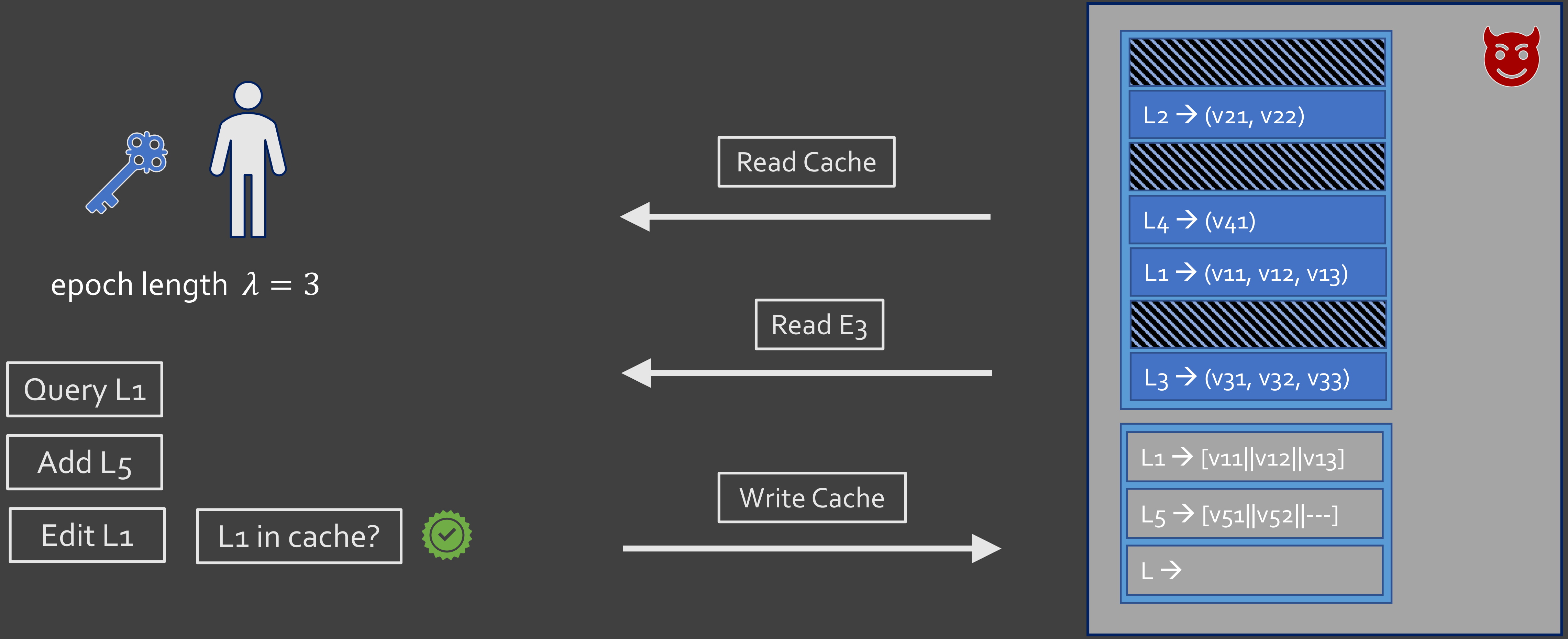
L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

L5 → [v51||v52||---]

L →

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5

Edit L1

L1 in cache? ✓

Read Cache

Read E3

Write Cache

▨

L2 → (v21, v22)

▨

L4 → (v41)

L1 → (v11, v12, v13)

▨

L3 → (v31, v32, v33)

L1 → [v11||v12||v13]

L5 → [v51||v52||---]

L →

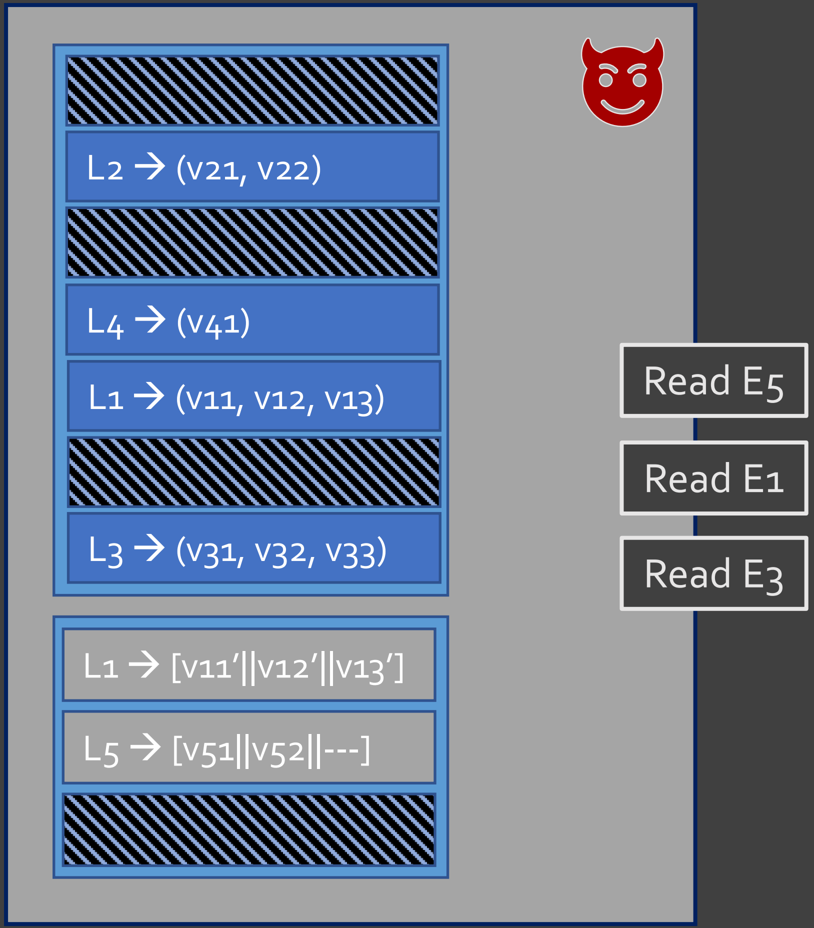
# The Dynamic Suppression Framework

epoch length $\lambda = 3$

Query L1

Add L5

Edit L1



L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

L3 → (v31, v32, v33)

L1 → [v11'||v12'||v13']

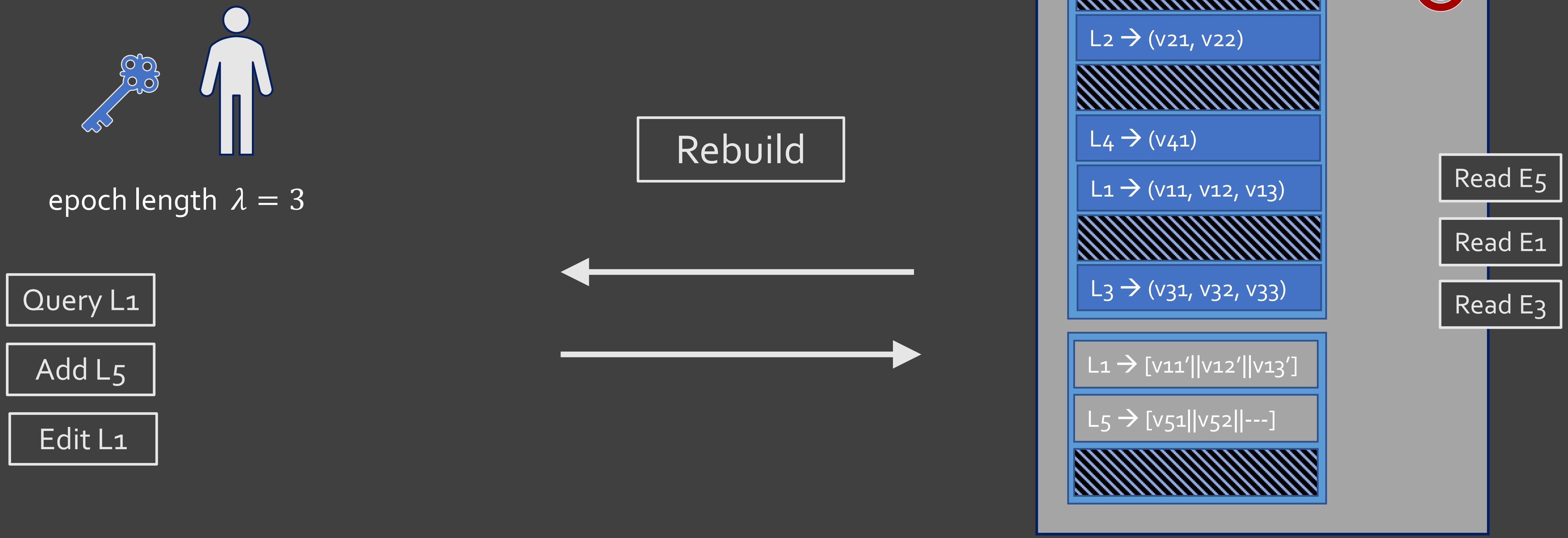
L5 → [v51||v52||---]

Read E5

Read E1

Read E3

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Query L1

Add L5

Edit L1

Rebuild

L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

L3 → (v31, v32, v33)

L1 → [v11’||v12’||v13’]

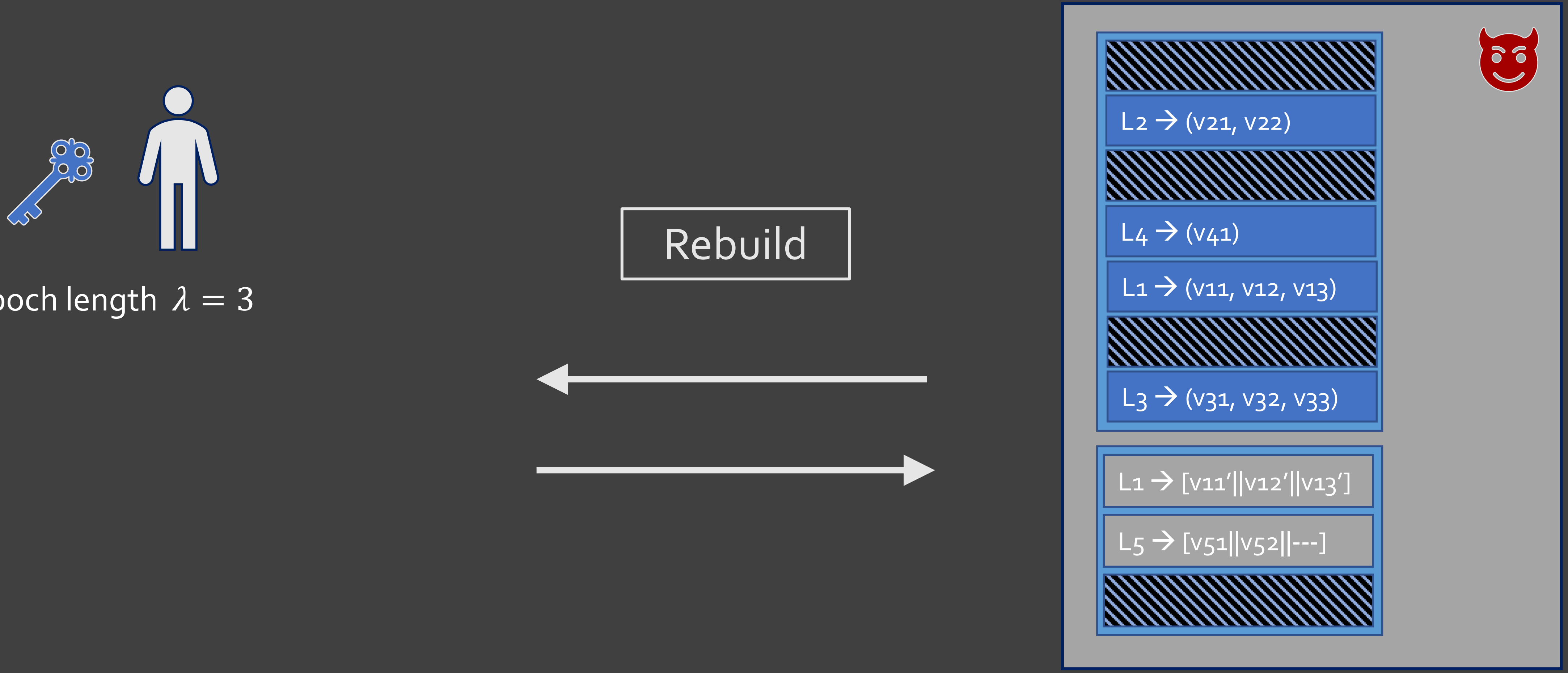
L5 → [v51||v52||---]

Read E5

Read E1

Read E3

# The Dynamic Suppression Framework



epoch length $\lambda = 3$

Rebuild

L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

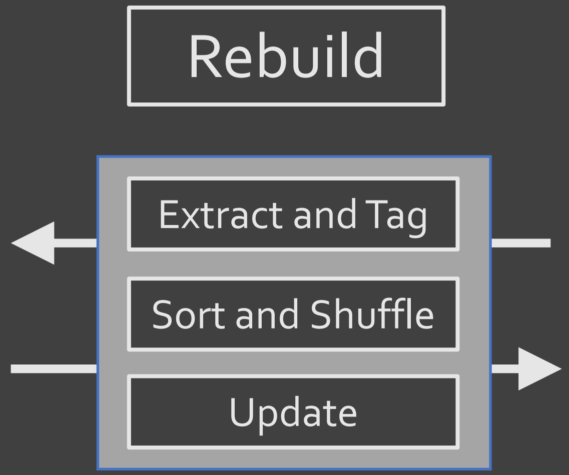
L3 → (v31, v32, v33)

L1 → [v11′||v12′||v13′]

L5 → [v51||v52||---]

# The Dynamic Suppression Framework

epoch length $\lambda = 3$

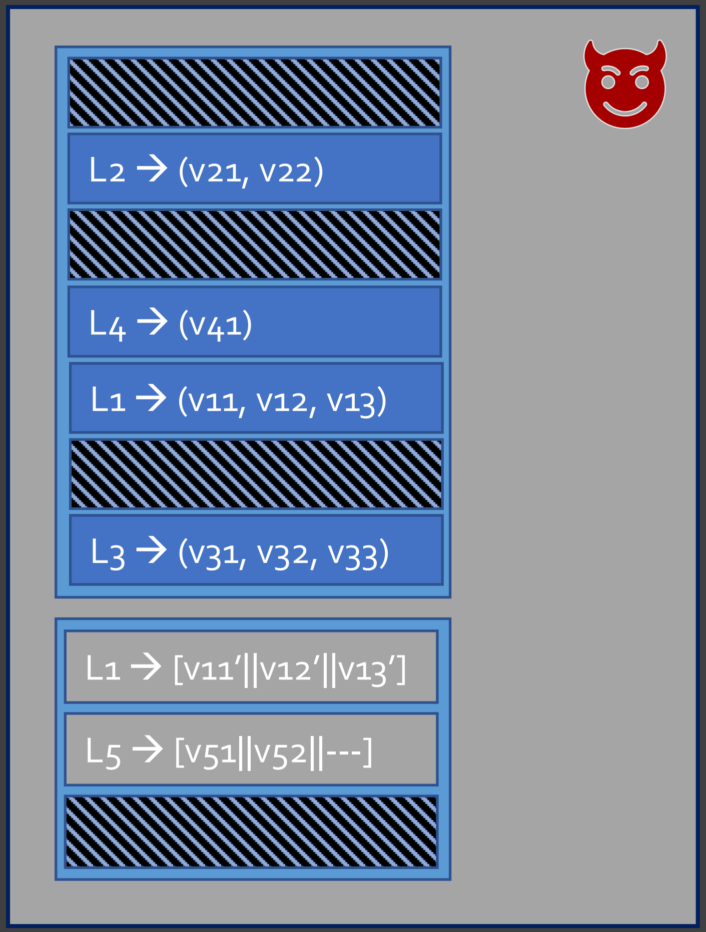


Rebuild

Extract and Tag

Sort and Shuffle

Update



L2 → (v21, v22)

L4 → (v41)

L1 → (v11, v12, v13)

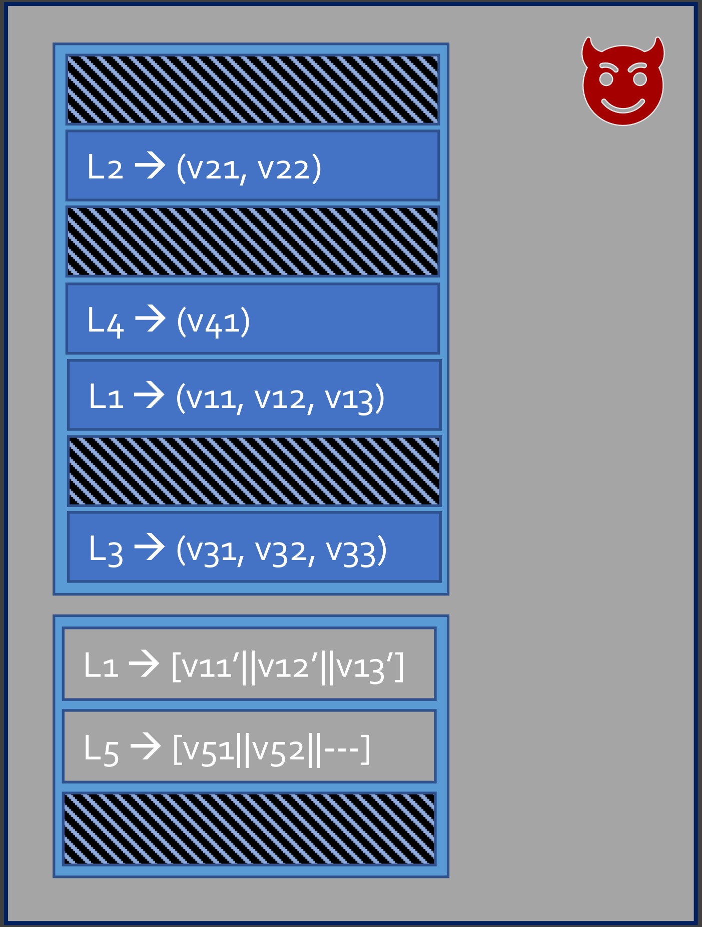
L3 → (v31, v32, v33)

L1 → [v11′||v12′||v13′]

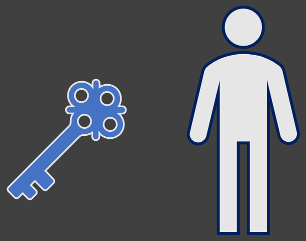
L5 → [v51||v52||---]

# Rebuilding: Extract and Tag

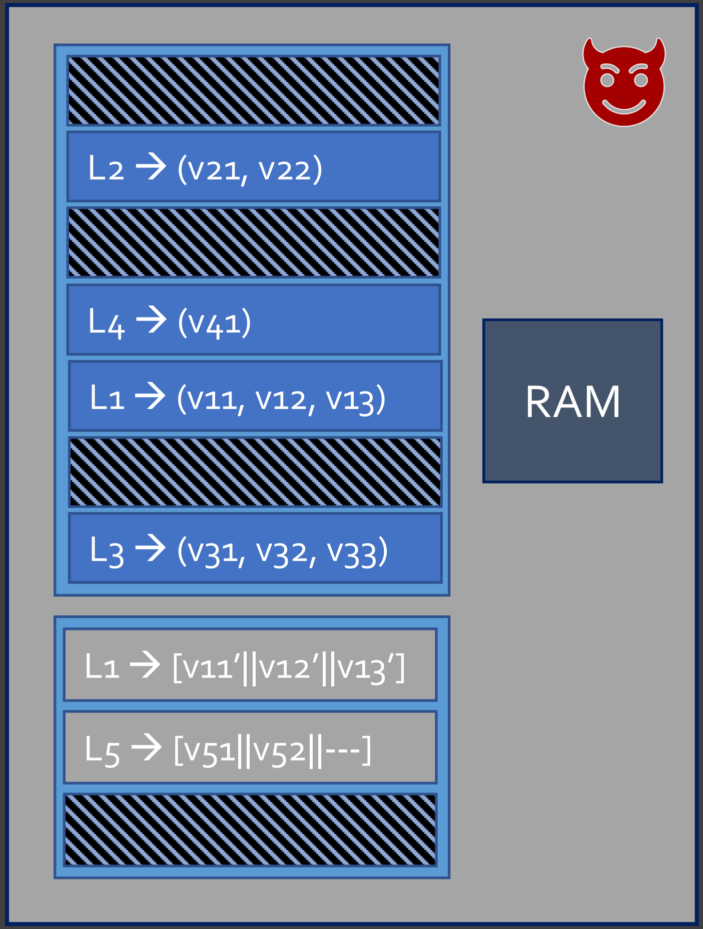
epoch length $\lambda = 3$



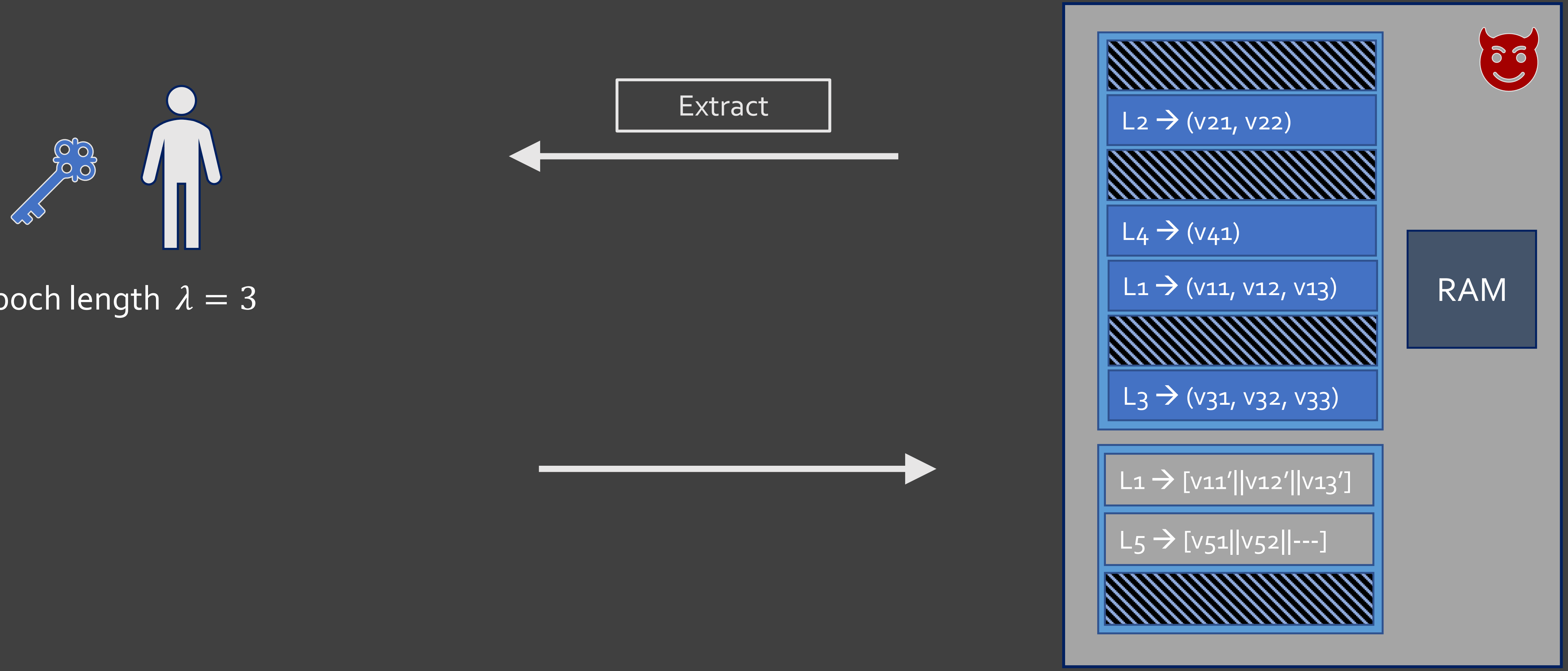
# Rebuilding: Extract and Tag



epoch length $\lambda = 3$

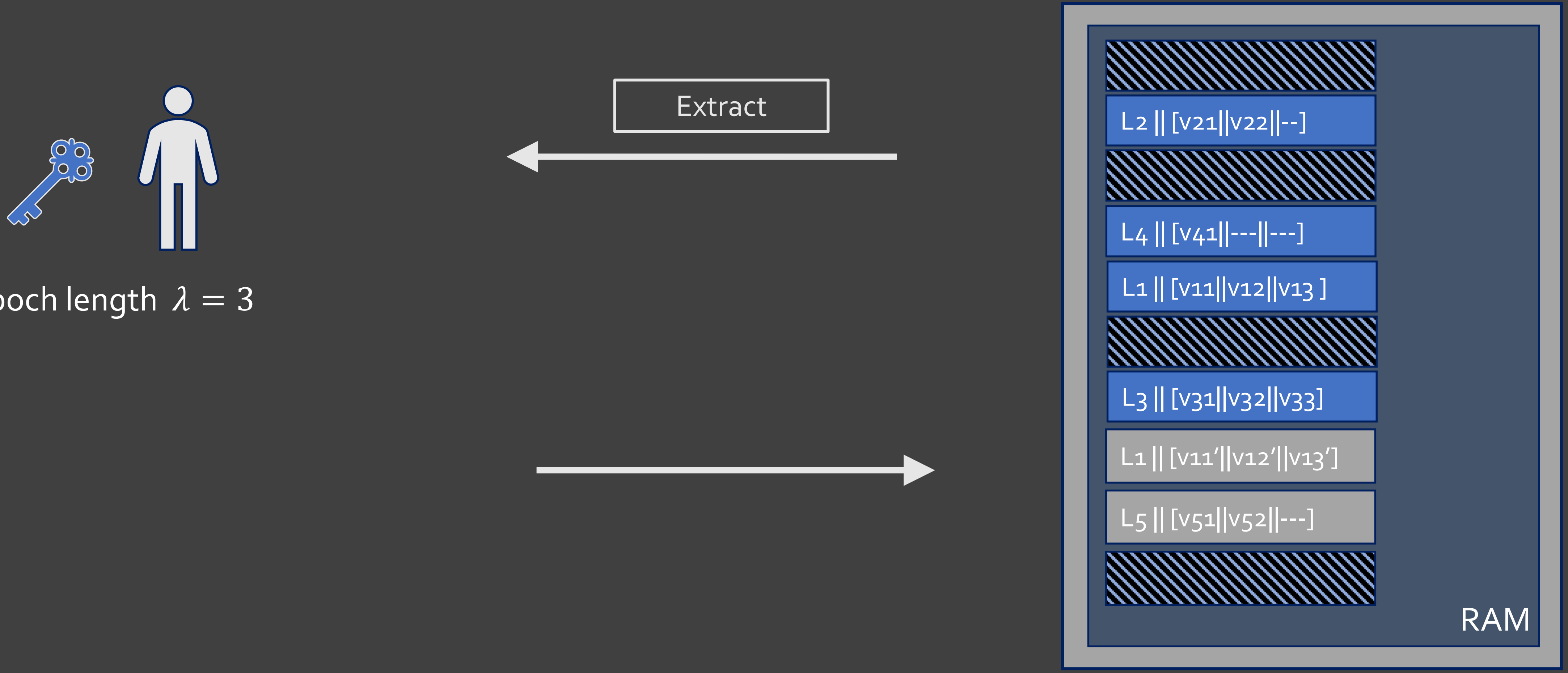


# Rebuilding: Extract and Tag



epoch length $\lambda = 3$

# Rebuilding: Extract and Tag



Extract

epoch length $\lambda = 3$

L2 || [v21||v22||--]

L4 || [v41||---||---]

L1 || [v11||v12||v13 ]

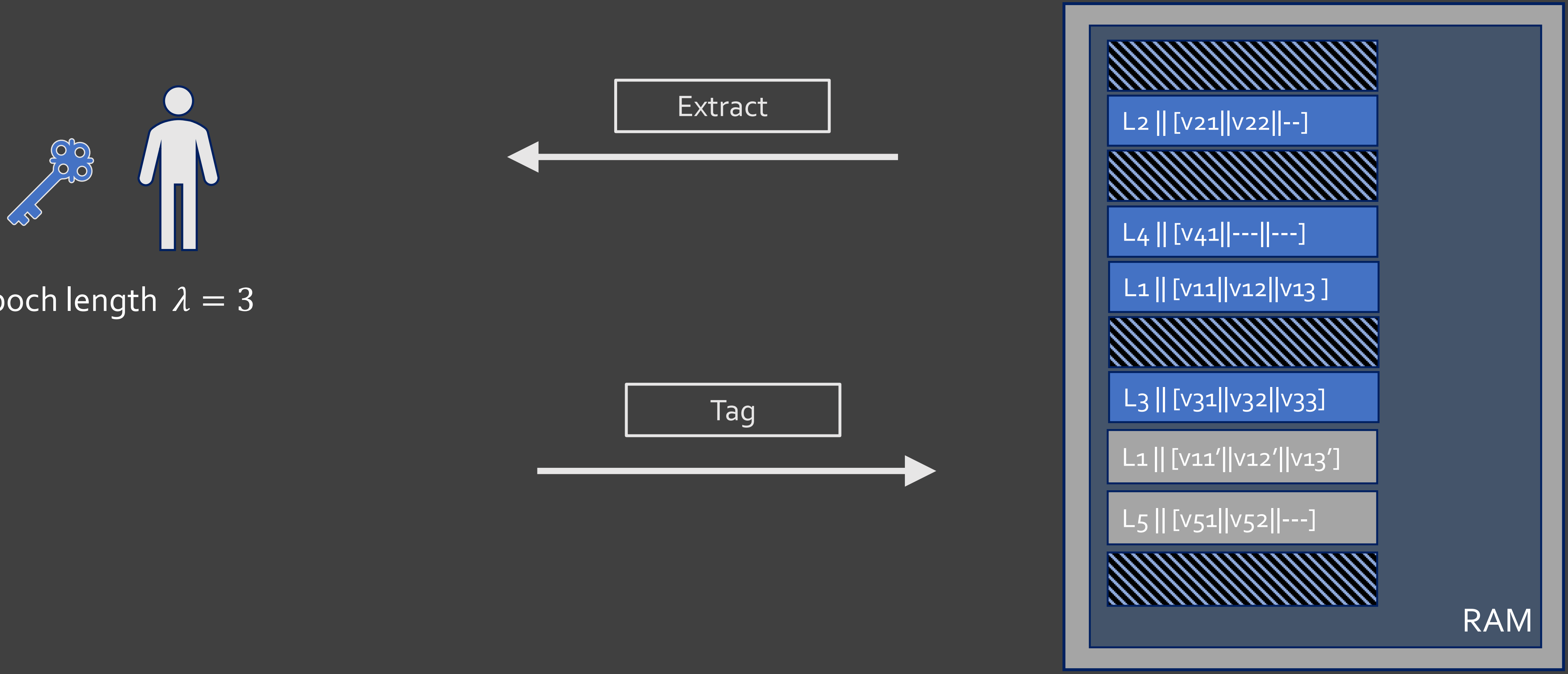
L3 || [v31||v32||v33]

L1 || [v11’||v12’||v13’]

L5 || [v51||v52||---]

RAM

# Rebuilding: Extract and Tag



epoch length $\lambda = 3$

Extract

Tag

L2 || [v21||v22||--]

L4 || [v41||---||---]

L1 || [v11||v12||v13 ]

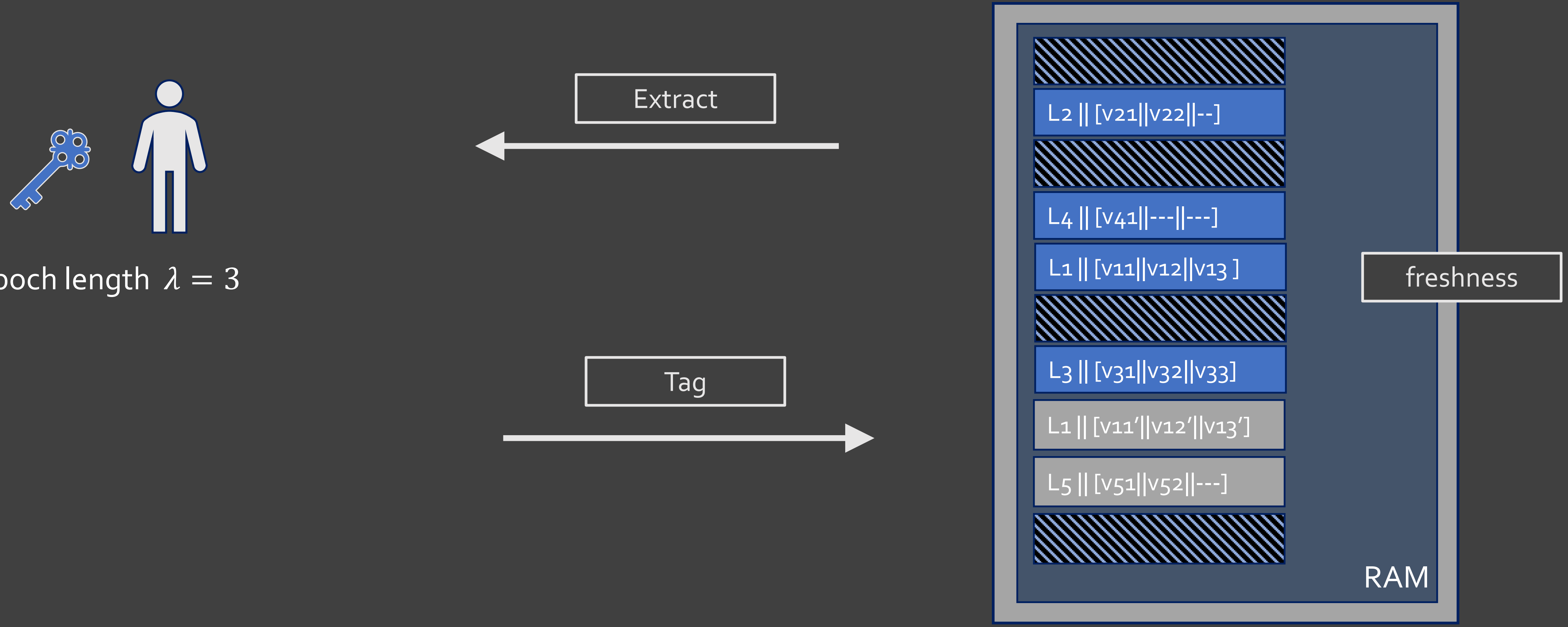
L3 || [v31||v32||v33]

L1 || [v11'||v12'||v13']

L5 || [v51||v52||---]

RAM

# Rebuilding: Extract and Tag



epoch length $\lambda = 3$

Extract

Tag

L2 || [v21||v22||--]

L4 || [v41||---||---]

L1 || [v11||v12||v13 ]

L3 || [v31||v32||v33]

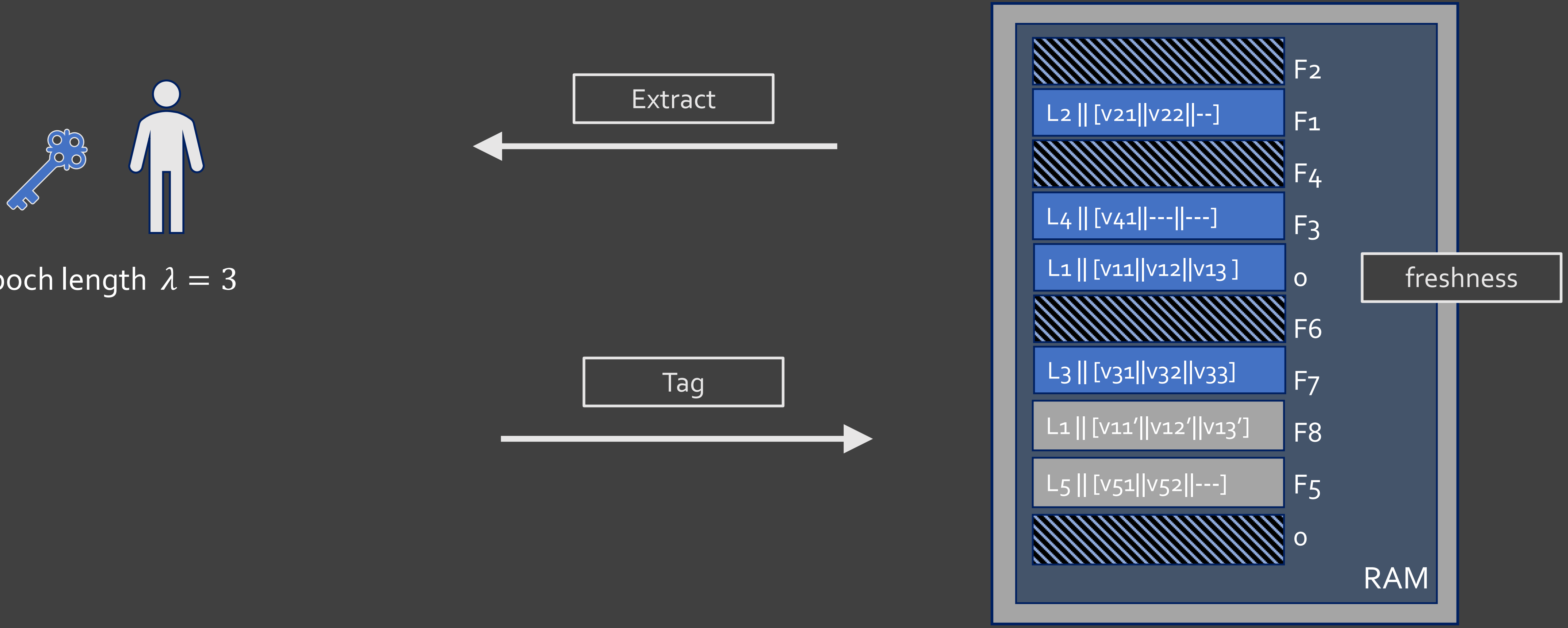
L1 || [v11’||v12’||v13’]

L5 || [v51||v52||---]

freshness

RAM

# Rebuilding: Extract and Tag



epoch length $\lambda = 3$

Extract

Tag

▨ F2

L2 || [v21||v22||--] F1

▨ F4

L4 || [v41||---||---] F3

L1 || [v11||v12||v13 ] 0

▨ F6

L3 || [v31||v32||v33] F7

L1 || [v11′||v12′||v13′] F8

L5 || [v51||v52||---] F5

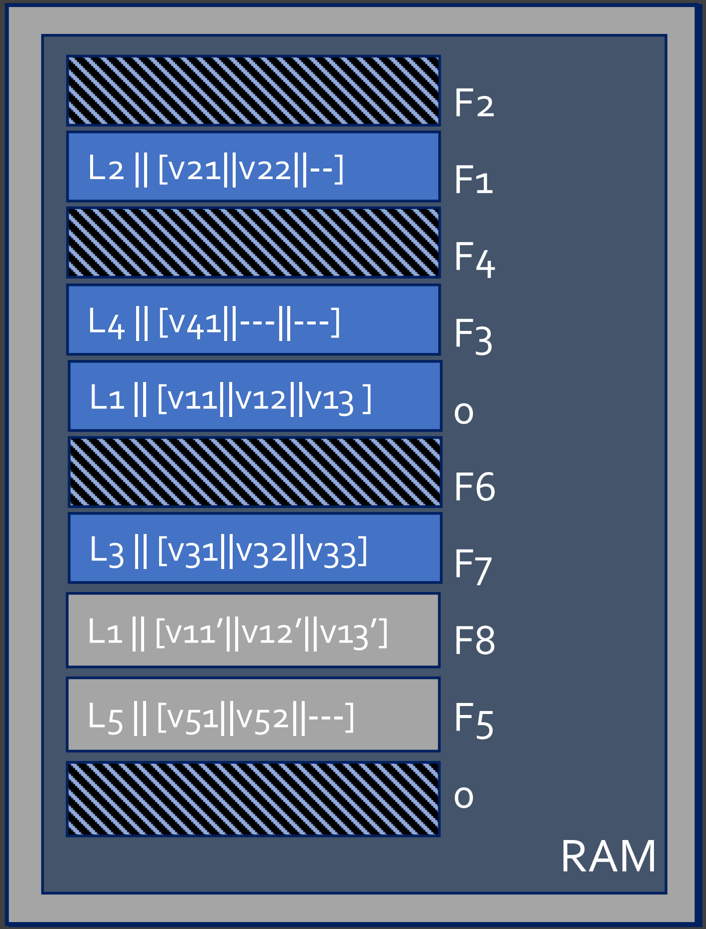
▨ 0

freshness

RAM

# Rebuilding: Sort and Shuffle

epoch length $\lambda = 3$



F2

L2 || [v21||v22||--] F1

F4

L4 || [v41||---||---] F3

L1 || [v11||v12||v13 ] 0

F6

L3 || [v31||v32||v33] F7

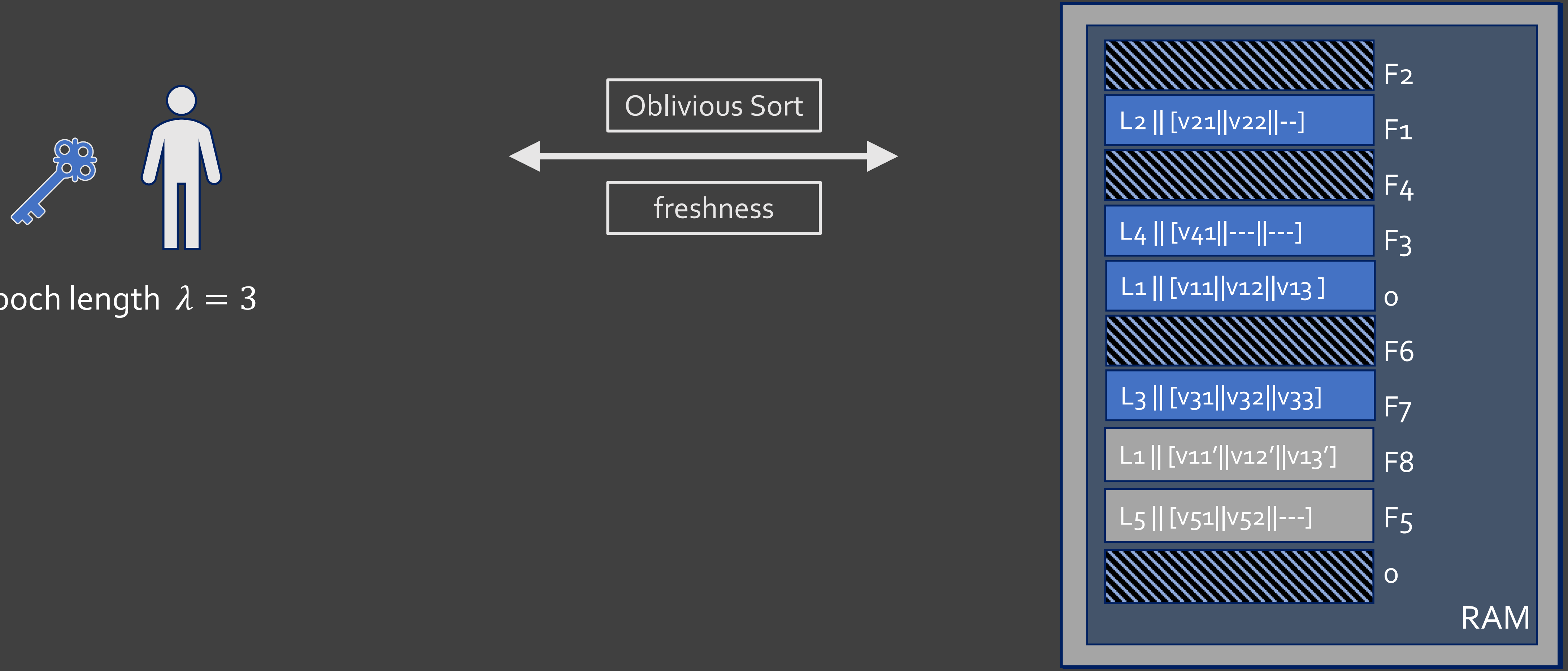
L1 || [v11'||v12'||v13'] F8

L5 || [v51||v52||---] F5

0

RAM

# Rebuilding: Sort and Shuffle



epoch length $\lambda = 3$

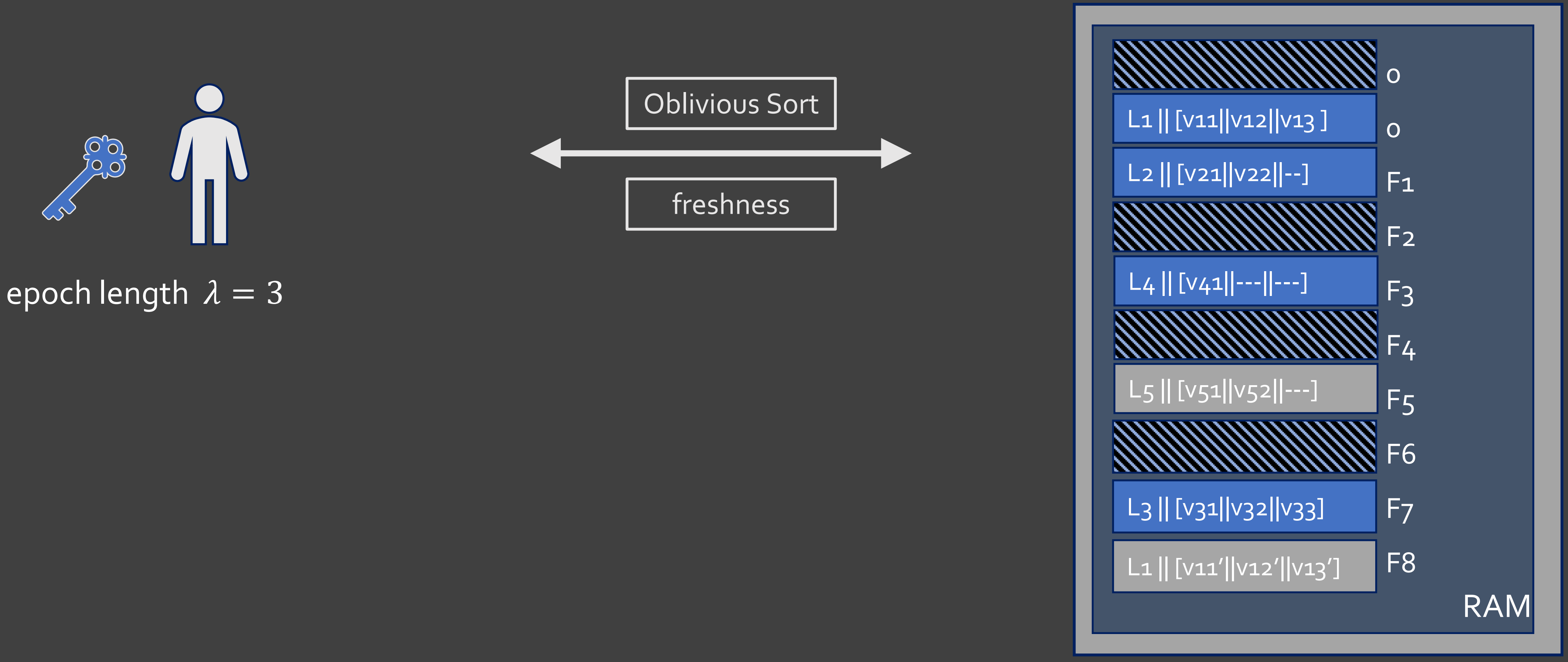
Oblivious Sort

freshness

| | |
|---|---|
| ▨ | F2 |
| L2 \|\| [v21\|\|v22\|\|--] | F1 |
| ▨ | F4 |
| L4 \|\| [v41\|\|---\|\|---] | F3 |
| L1 \|\| [v11\|\|v12\|\|v13 ] | 0 |
| ▨ | F6 |
| L3 \|\| [v31\|\|v32\|\|v33] | F7 |
| L1 \|\| [v11'\|\|v12'\|\|v13'] | F8 |
| L5 \|\| [v51\|\|v52\|\|---] | F5 |
| ▨ | 0 |

RAM

# Rebuilding: Sort and Shuffle



epoch length $\lambda = 3$

Oblivious Sort

freshness

▨ 0

L1 || [v11||v12||v13 ] 0

L2 || [v21||v22||--] F1

▨ F2

L4 || [v41||---||---] F3

▨ F4

L5 || [v51||v52||---] F5

▨ F6

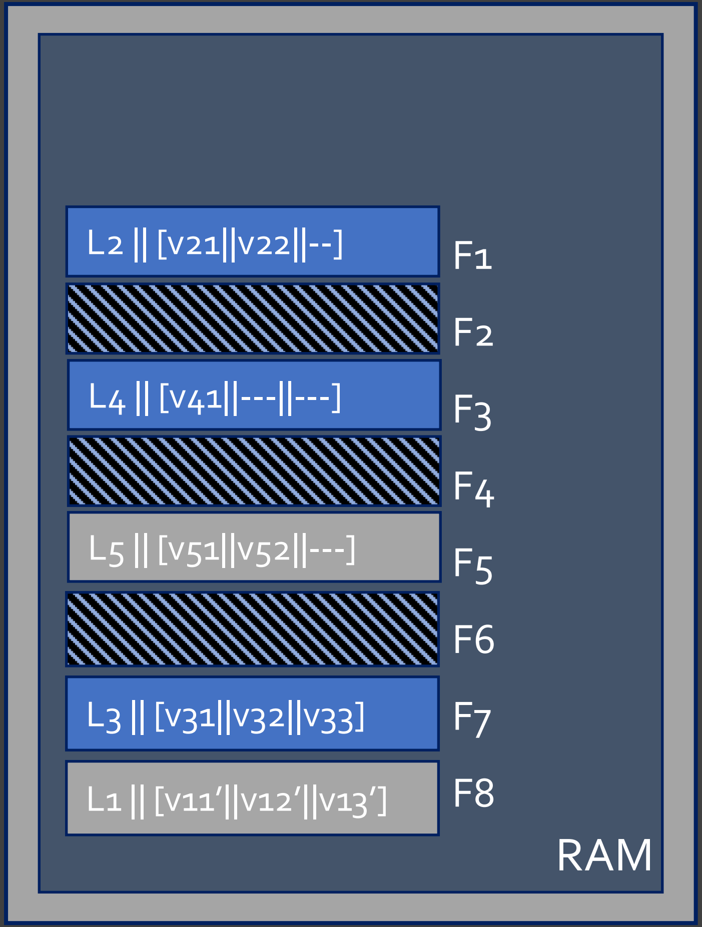
L3 || [v31||v32||v33] F7

L1 || [v11'||v12'||v13'] F8

RAM

# Rebuilding: Update

epoch length $\lambda = 3$



L2 || [v21||v22||--] F1

▨ F2

L4 || [v41||---||---] F3

▨ F4

L5 || [v51||v52||---] F5

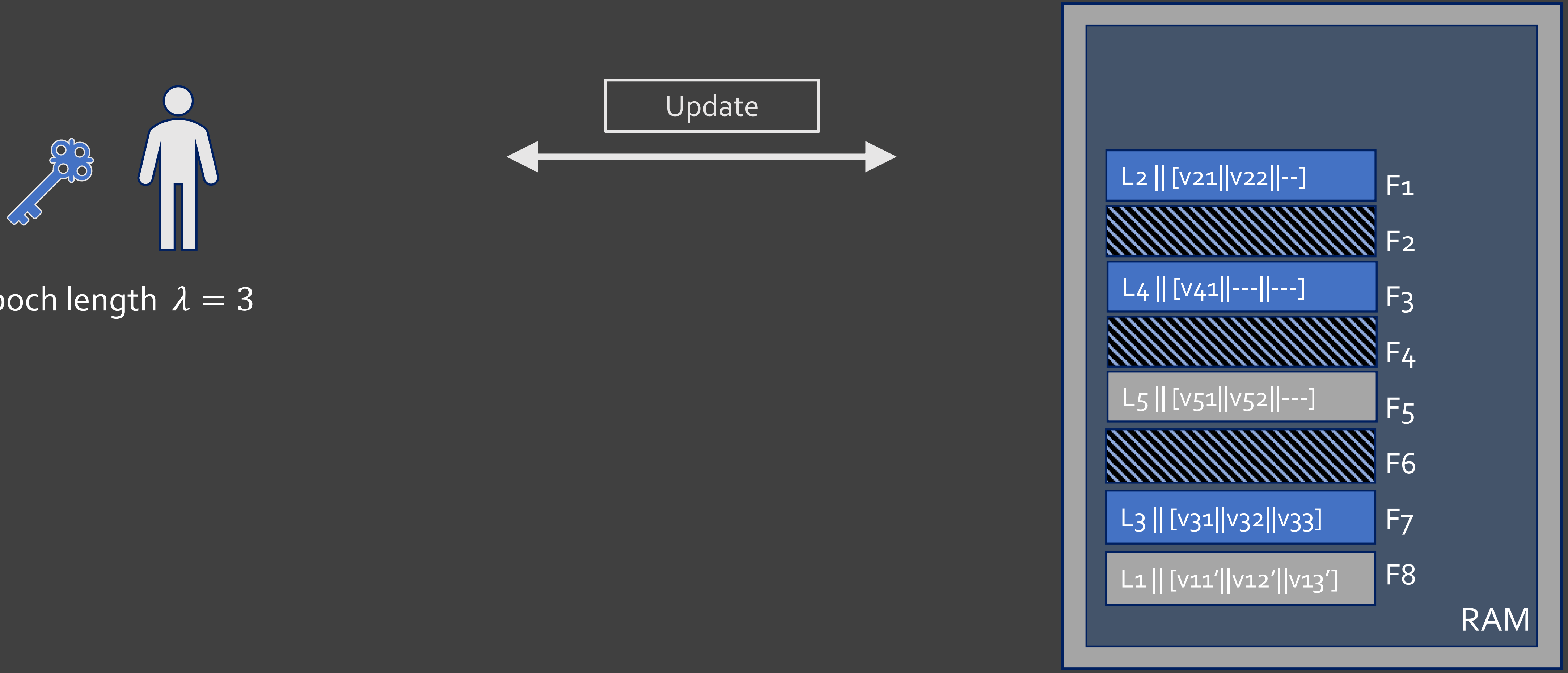
▨ F6

L3 || [v31||v32||v33] F7

L1 || [v11'||v12'||v13'] F8

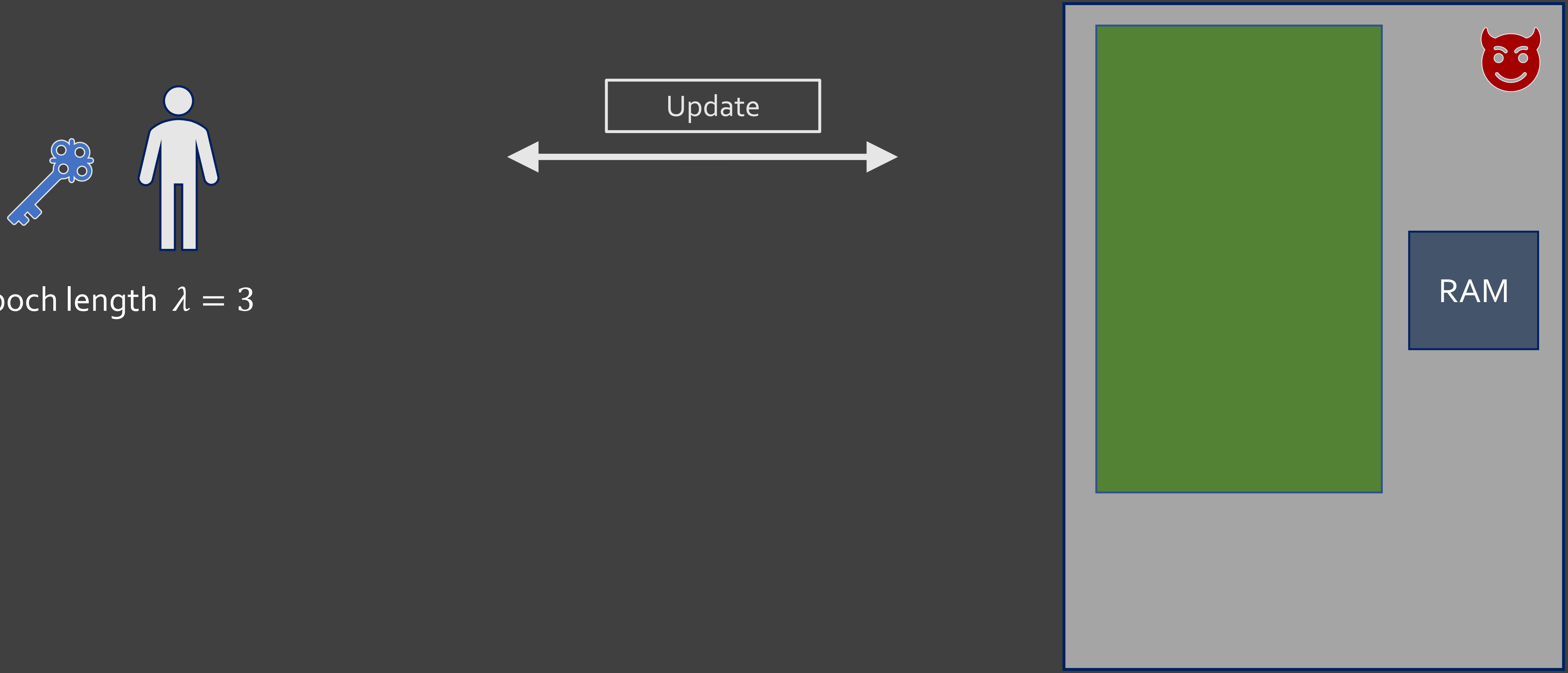
RAM

# Rebuilding: Update



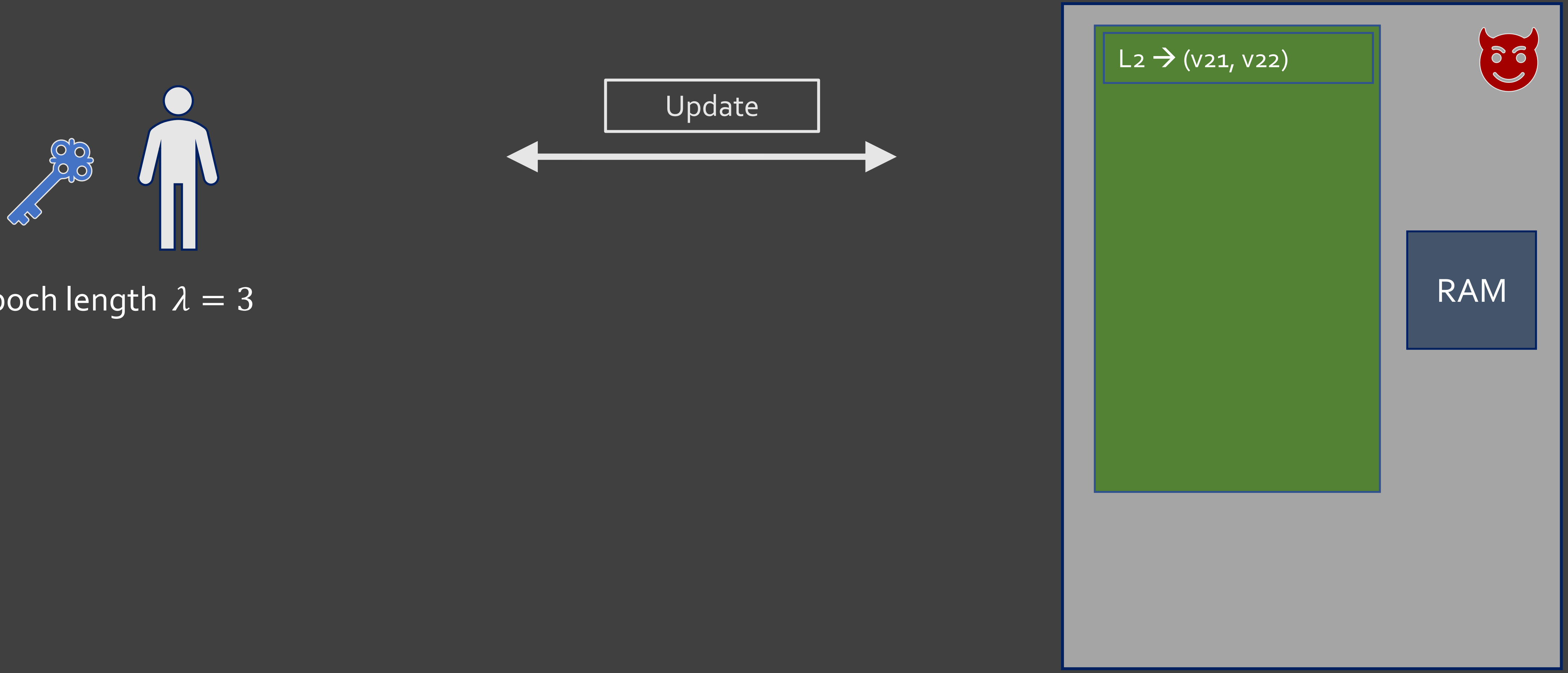
epoch length $\lambda = 3$

# Rebuilding: Update



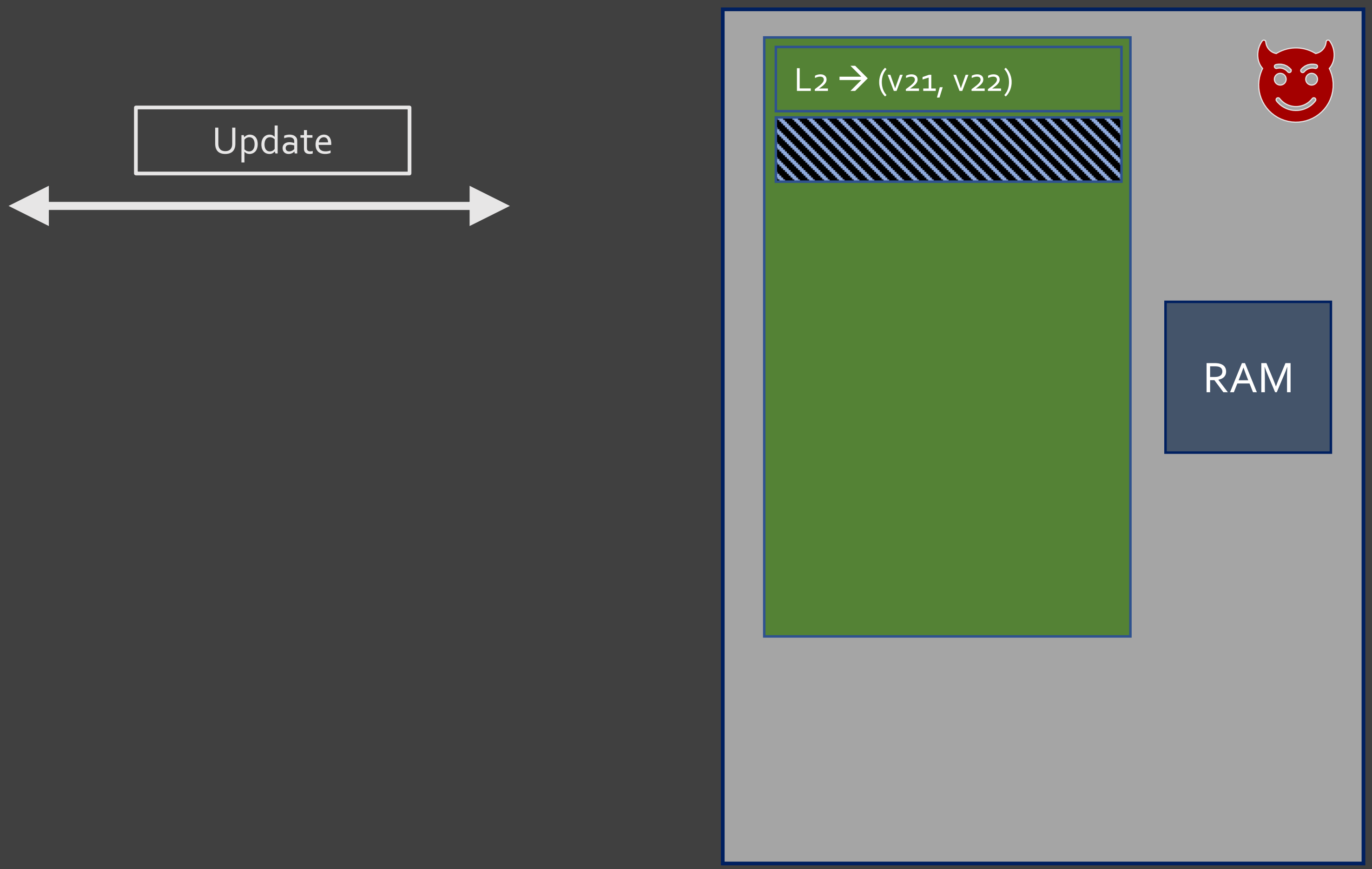
epoch length $\lambda = 3$

# Rebuilding: Update



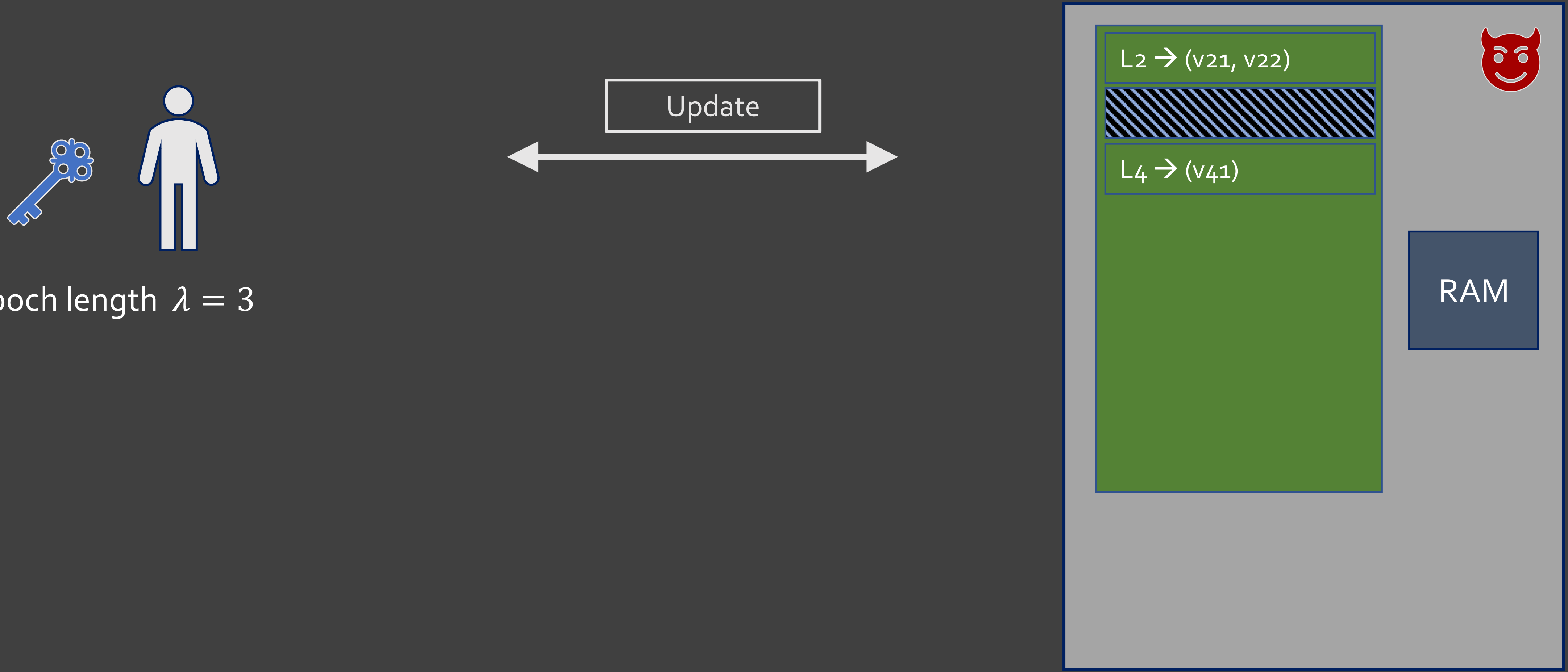
epoch length $\lambda = 3$

# Rebuilding: Update



epoch length $\lambda = 3$

# Rebuilding: Update



Update

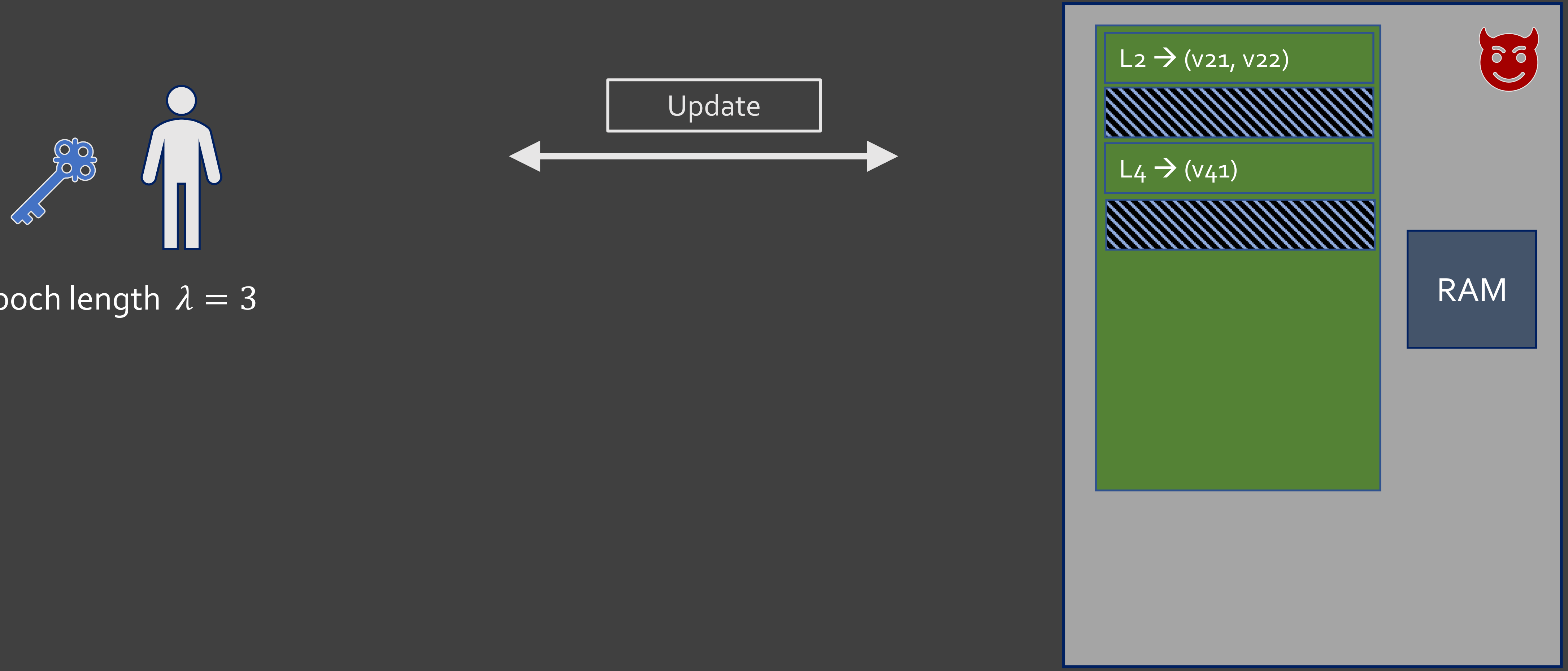
L2 → (v21, v22)

L4 → (v41)

RAM

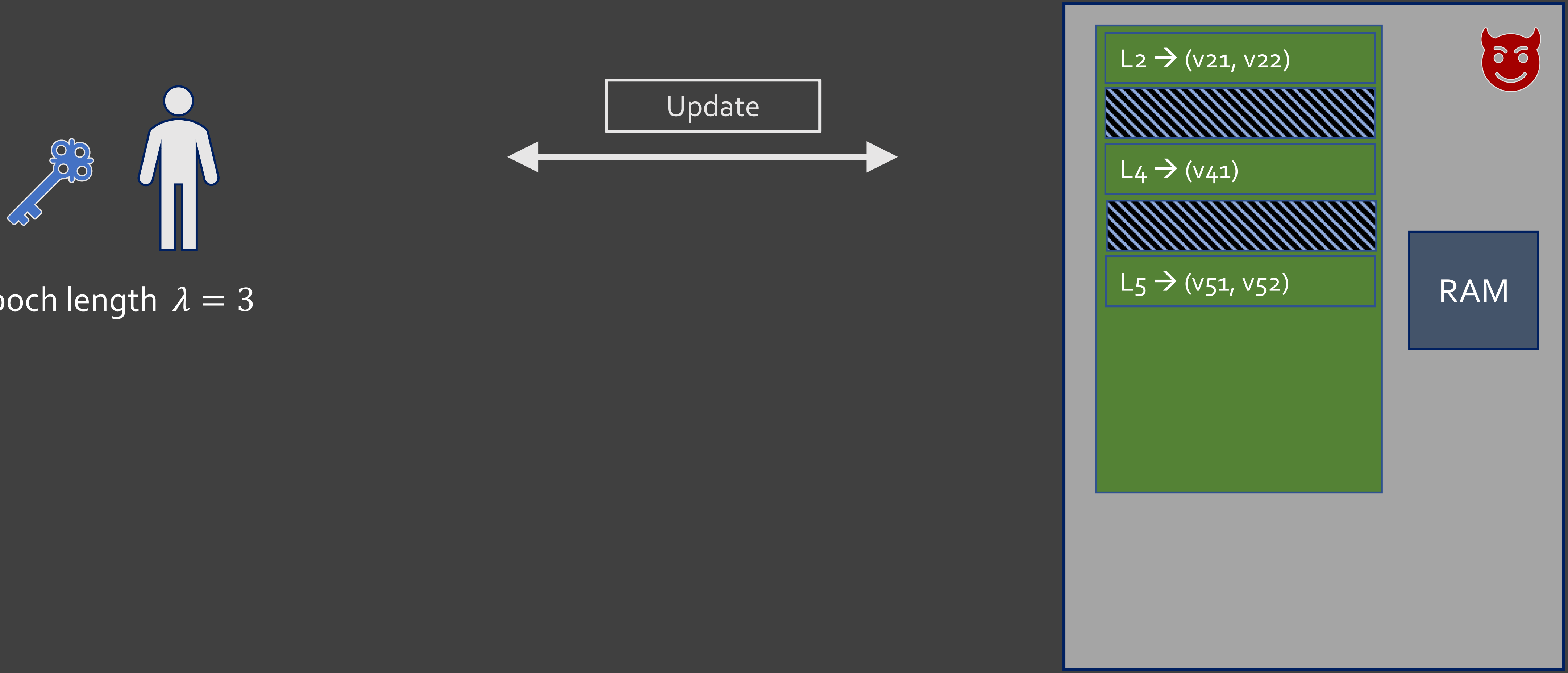
epoch length $\lambda = 3$

# Rebuilding: Update



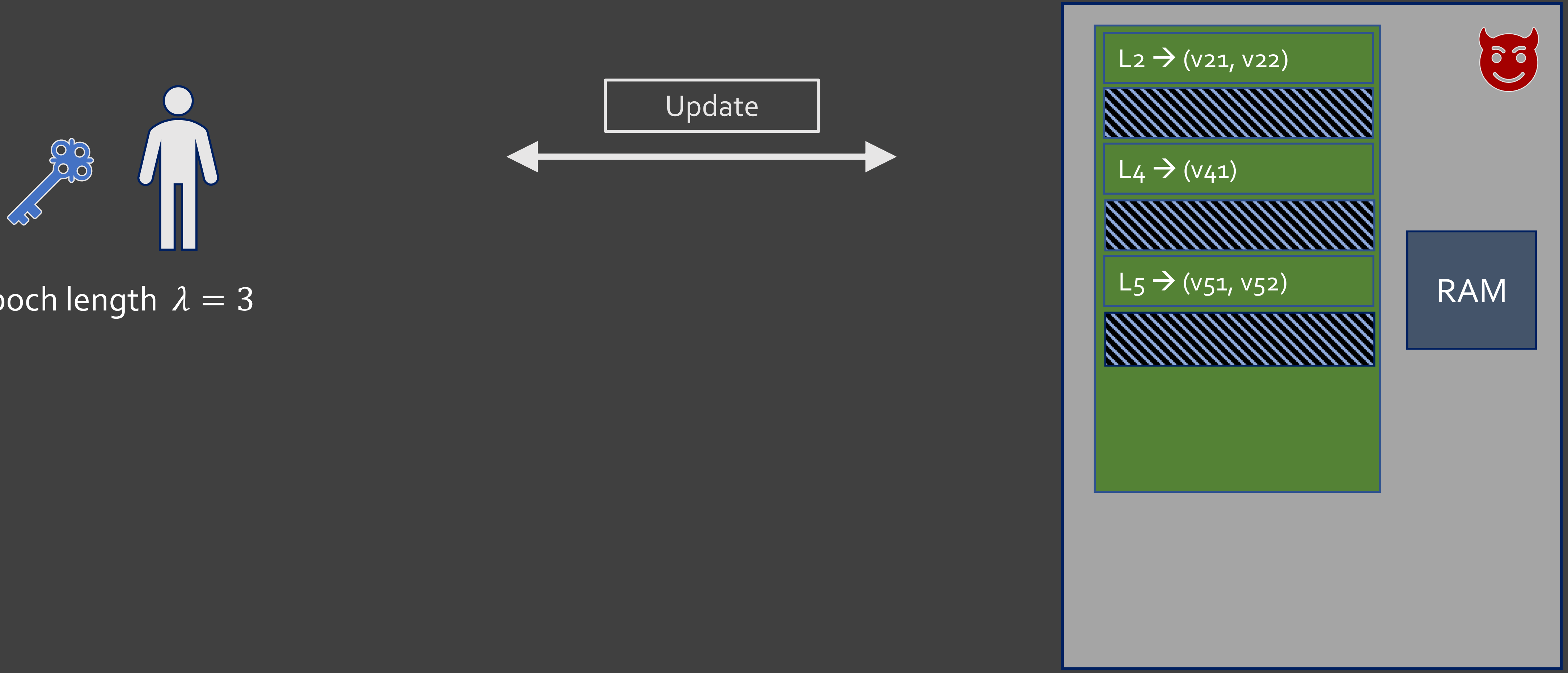
epoch length $\lambda = 3$

# Rebuilding: Update



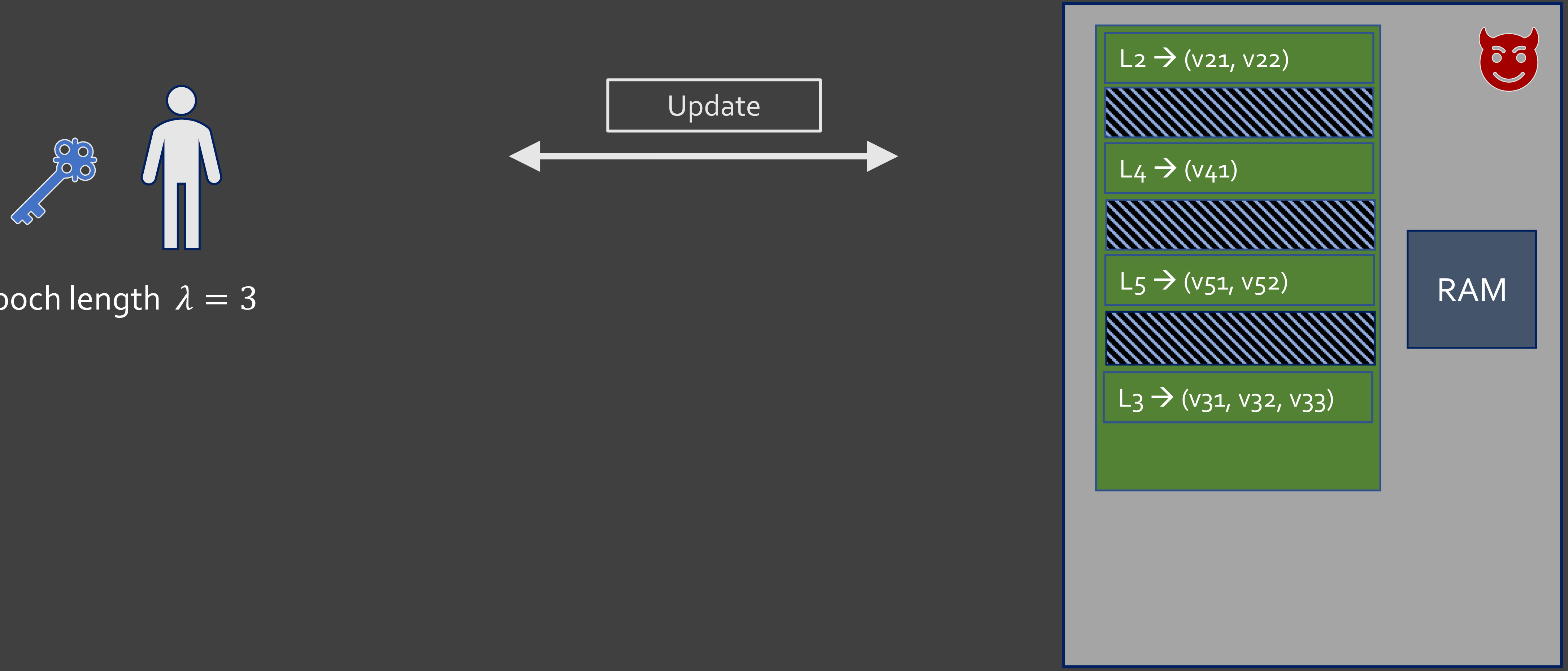
epoch length $\lambda = 3$

# Rebuilding: Update



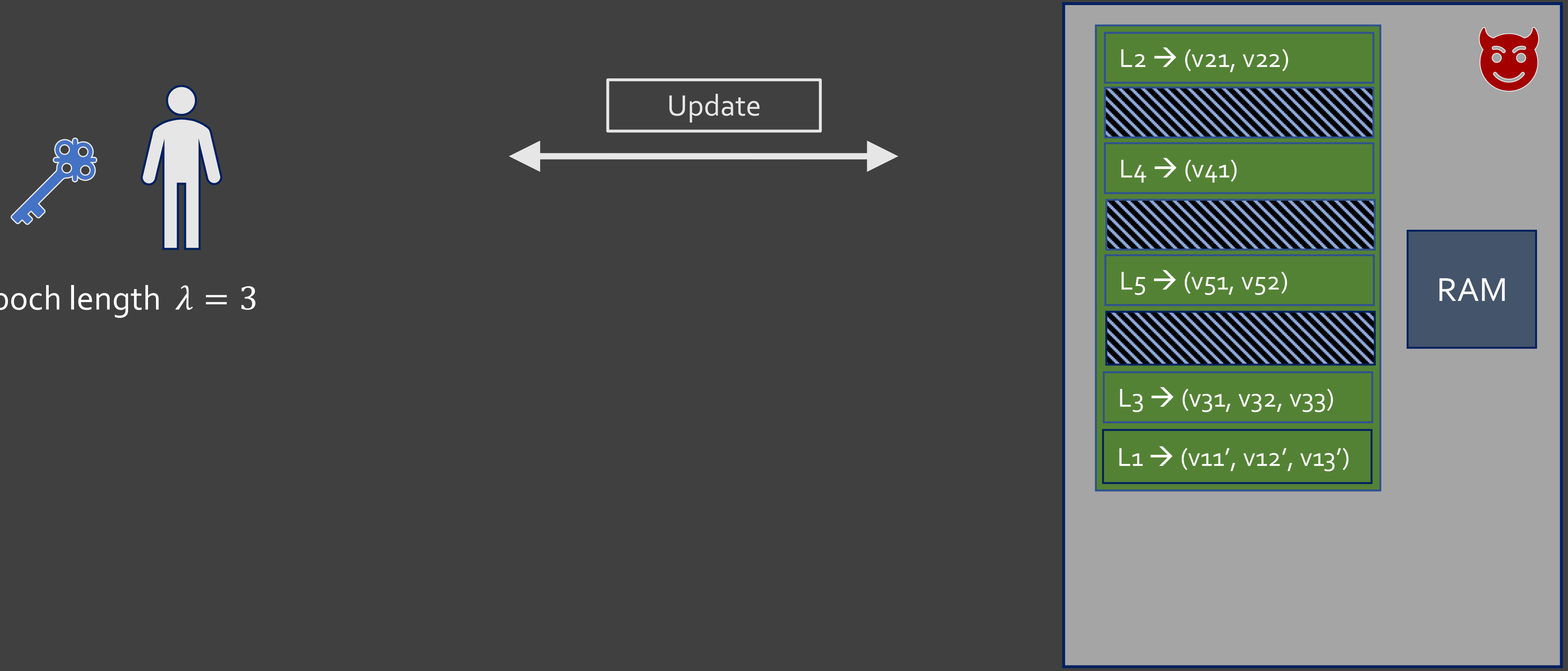
epoch length $\lambda = 3$

# Rebuilding: Update

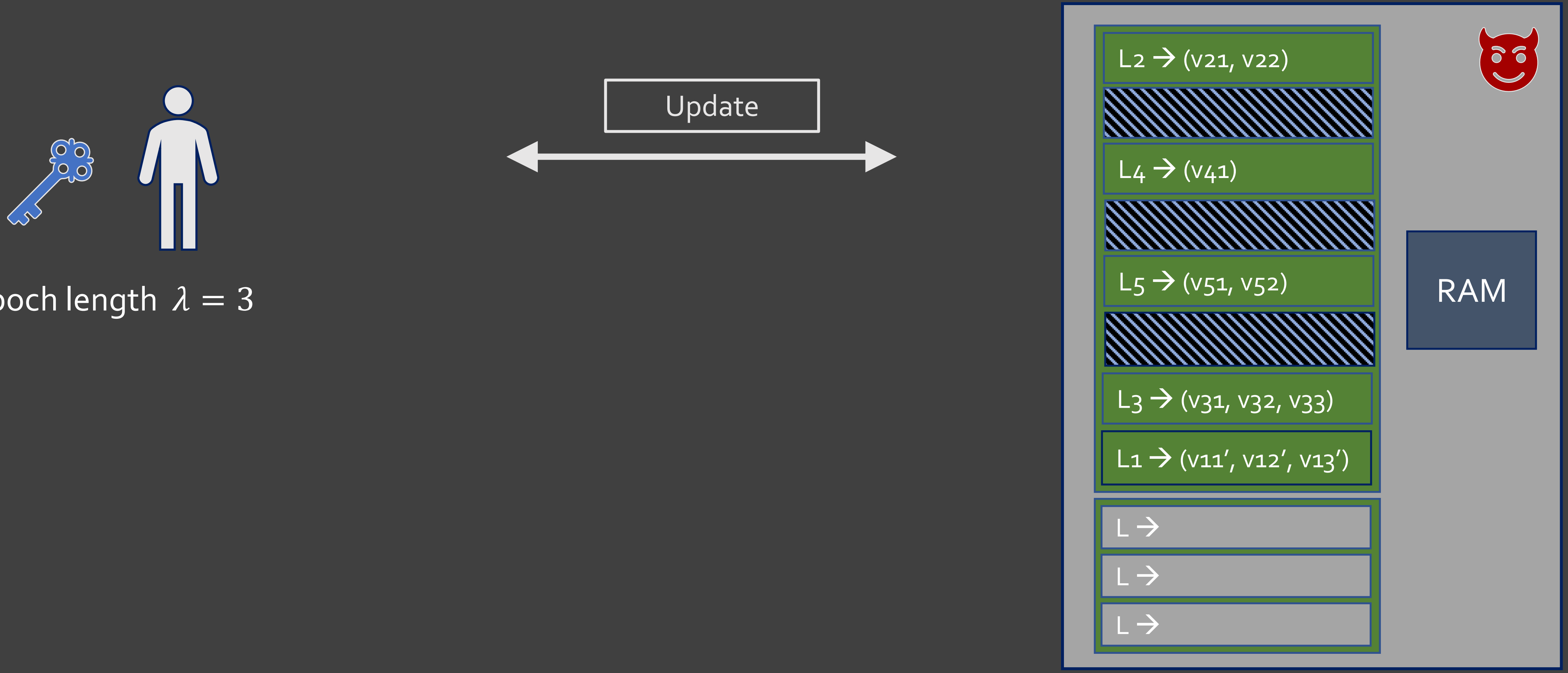


epoch length $\lambda = 3$

# Rebuilding: Update

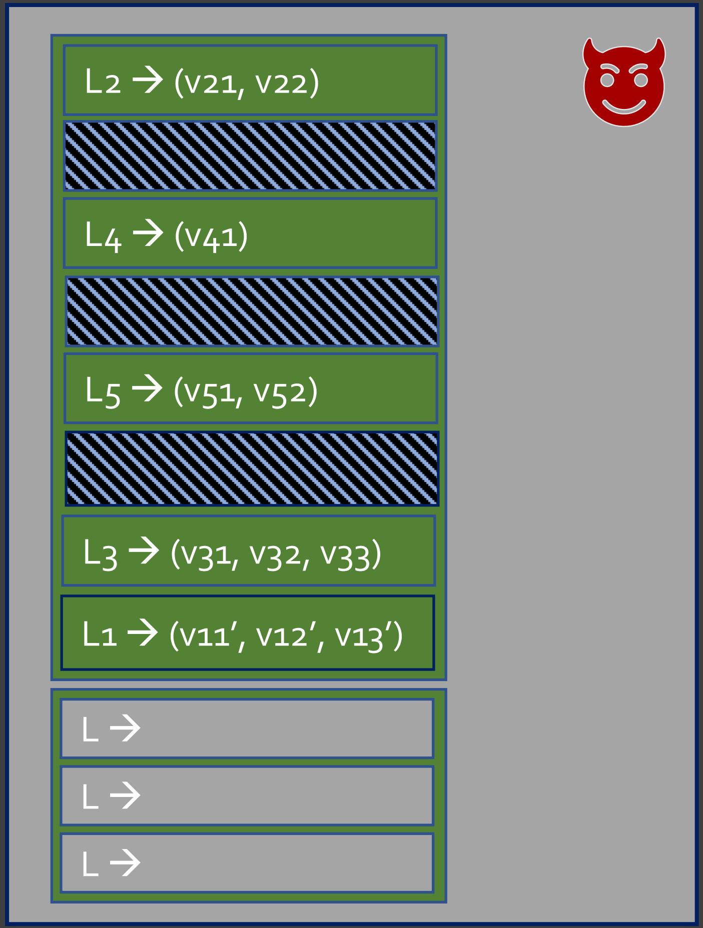


# Rebuilding: Update



# Rebuilding: Update

epoch length $\lambda = 3$



L2 → (v21, v22)

L4 → (v41)

L5 → (v51, v52)

L3 → (v31, v32, v33)

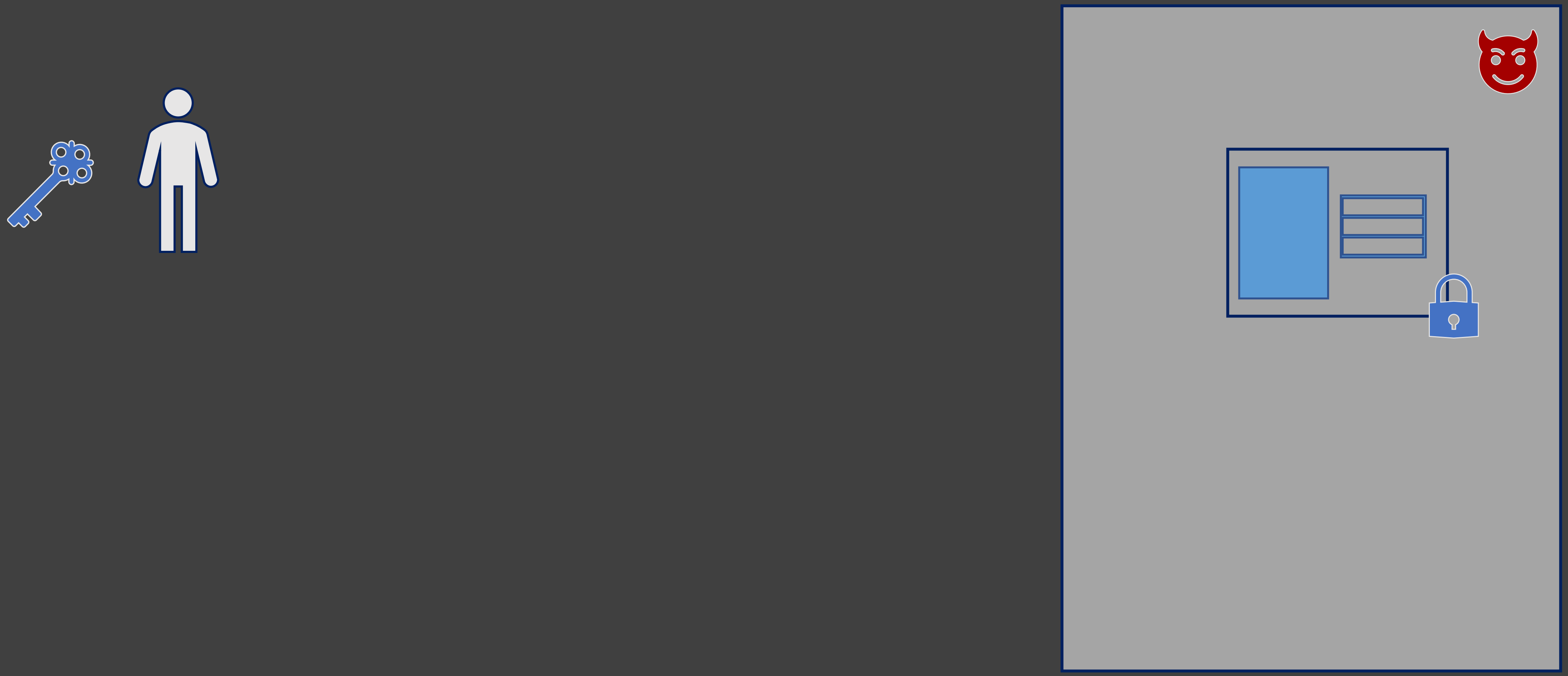
L1 → (v11', v12', v13')

L →

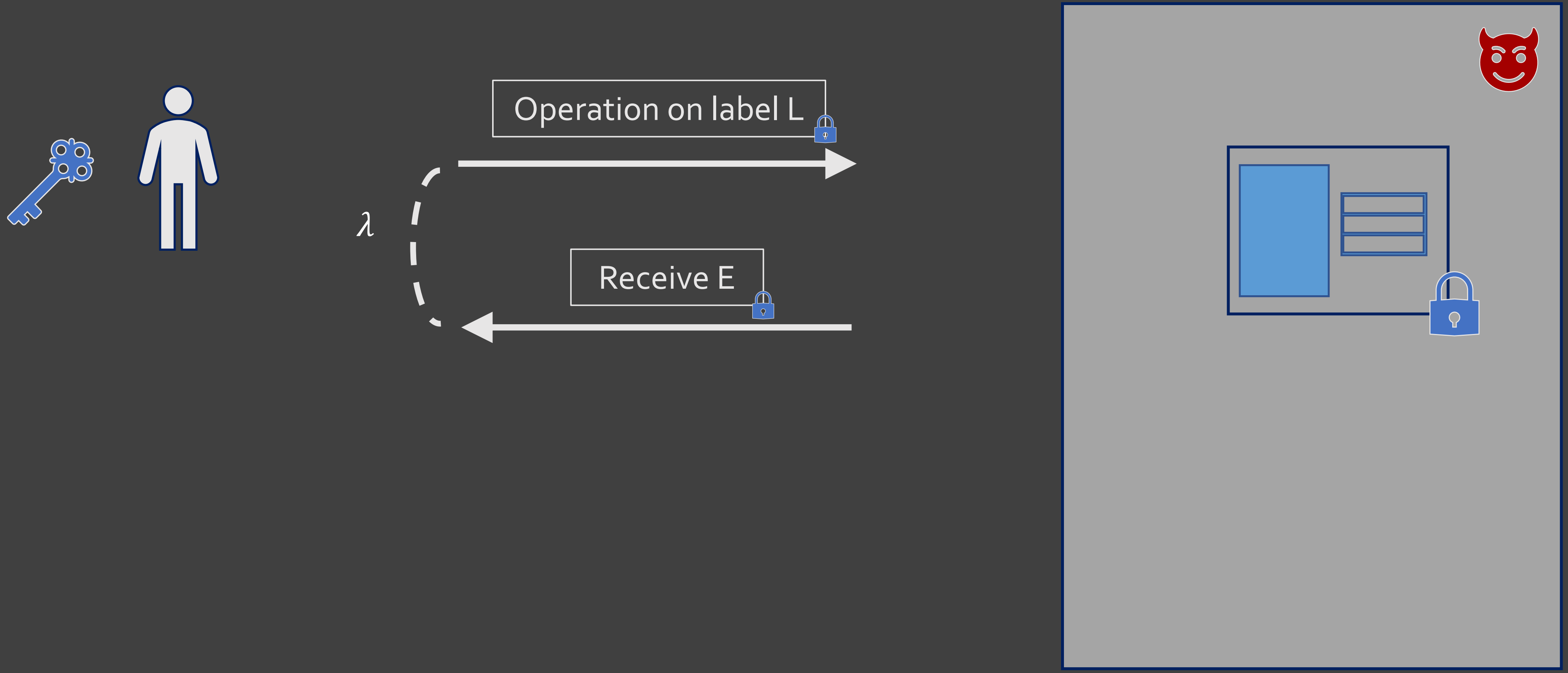
L →

L →

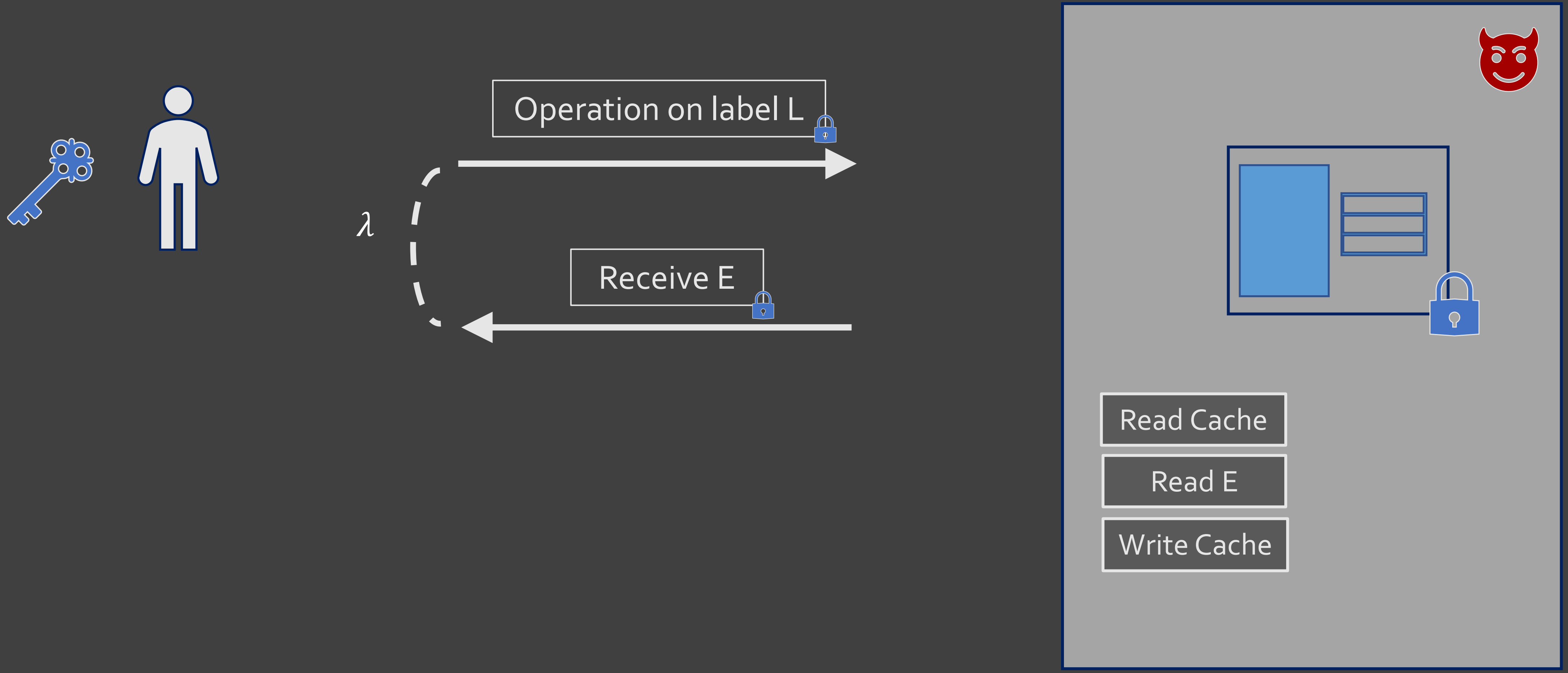
# Dynamic Framework in Action



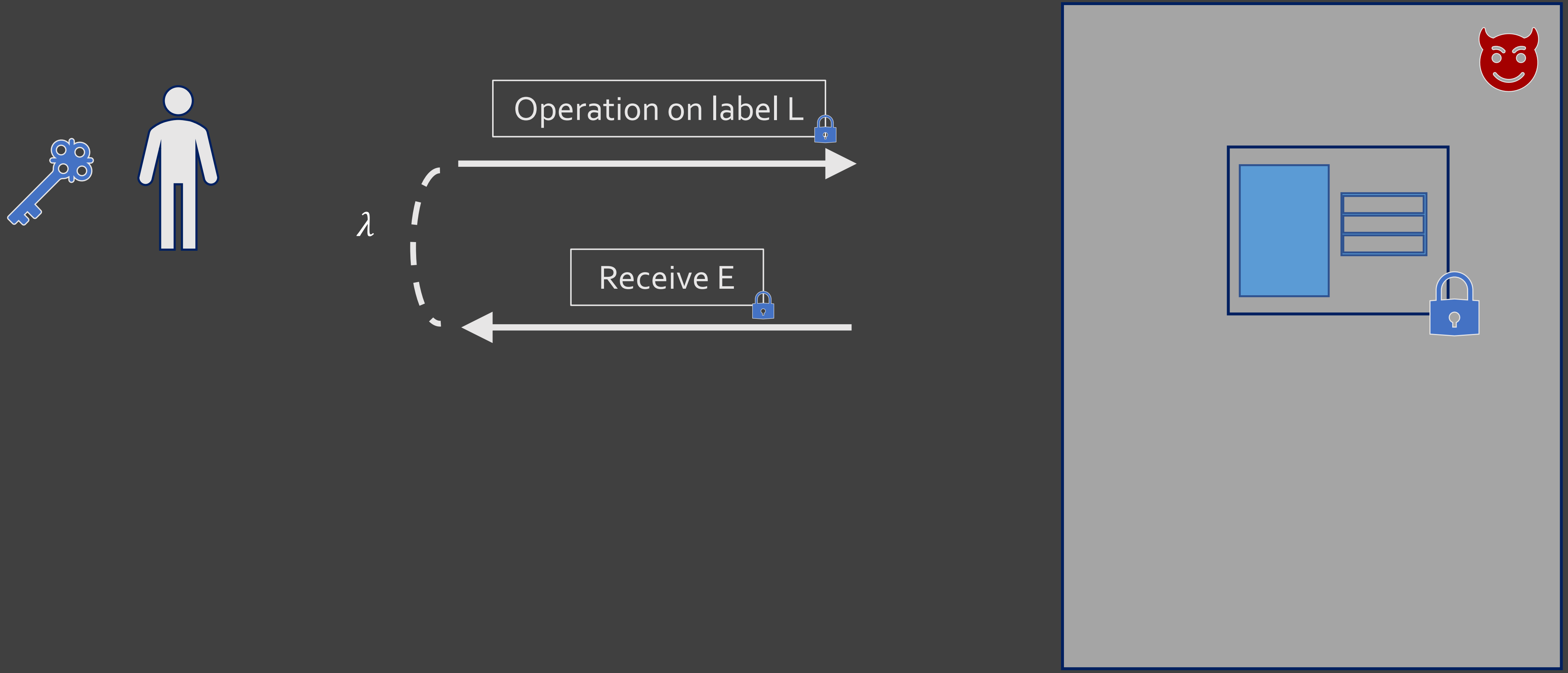
# Dynamic Framework in Action



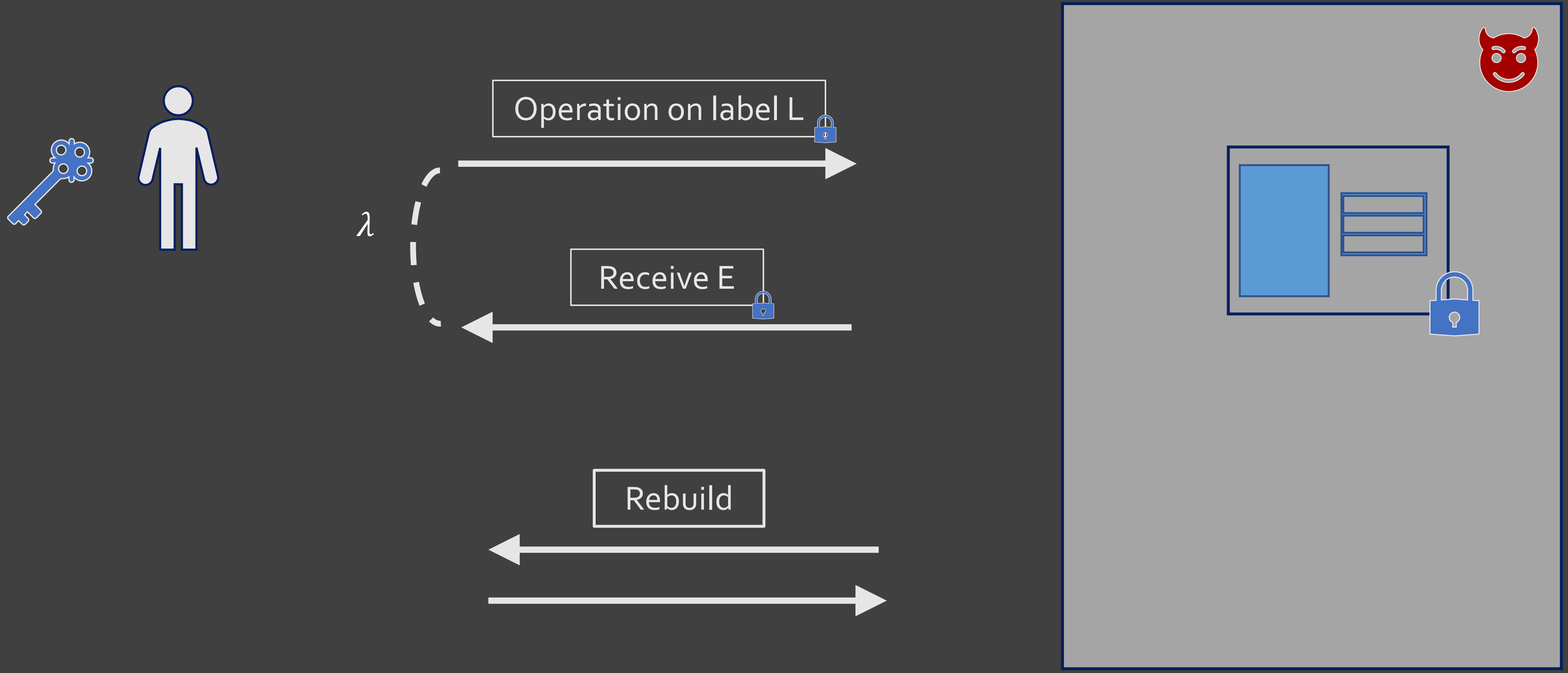
# Dynamic Framework in Action



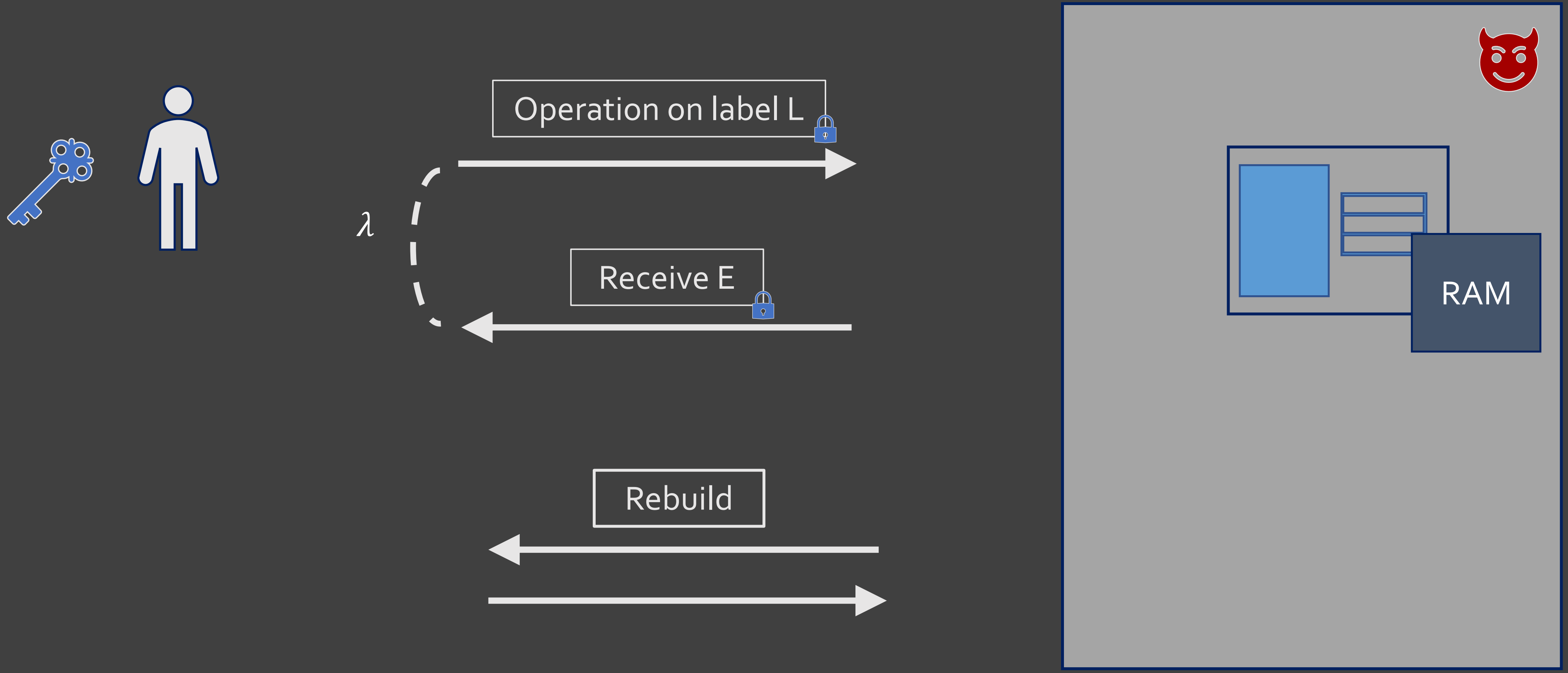
# Dynamic Framework in Action



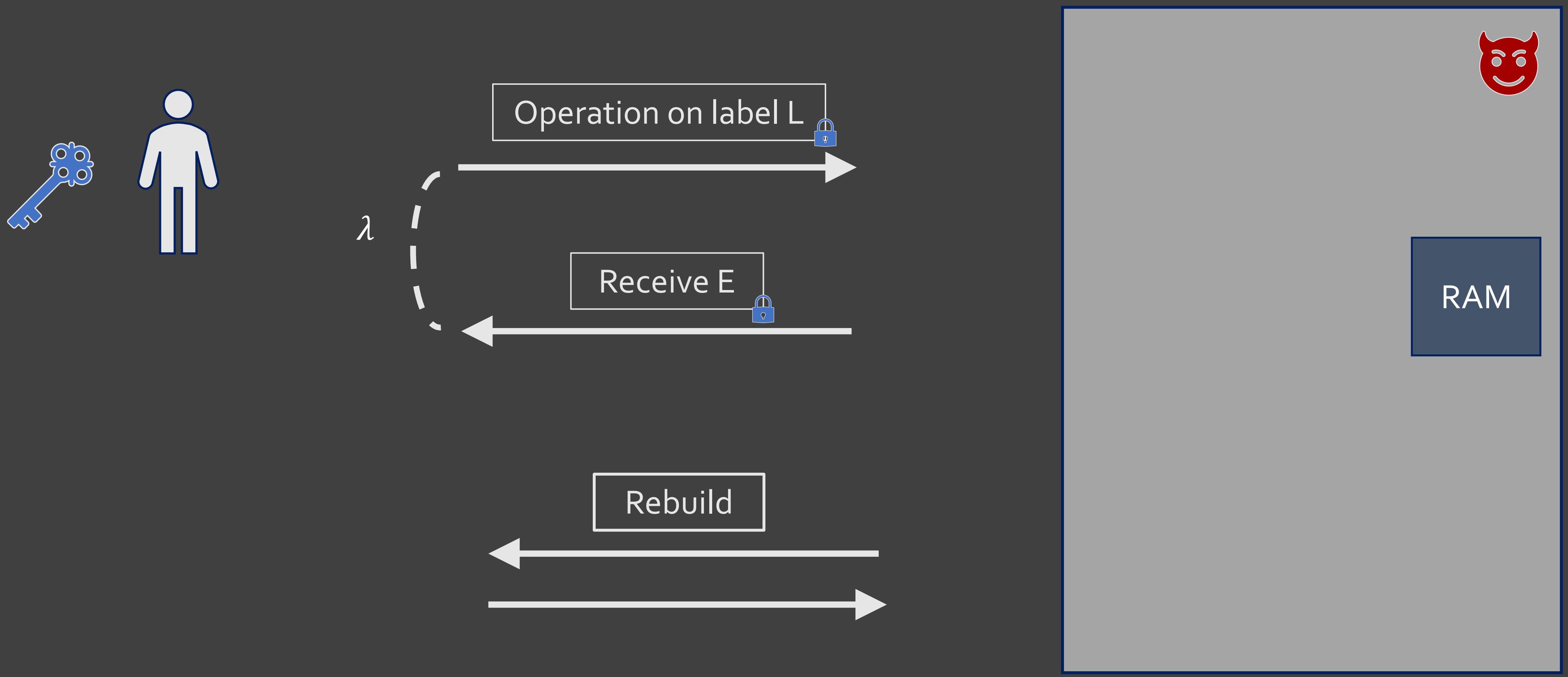
# Dynamic Framework in Action



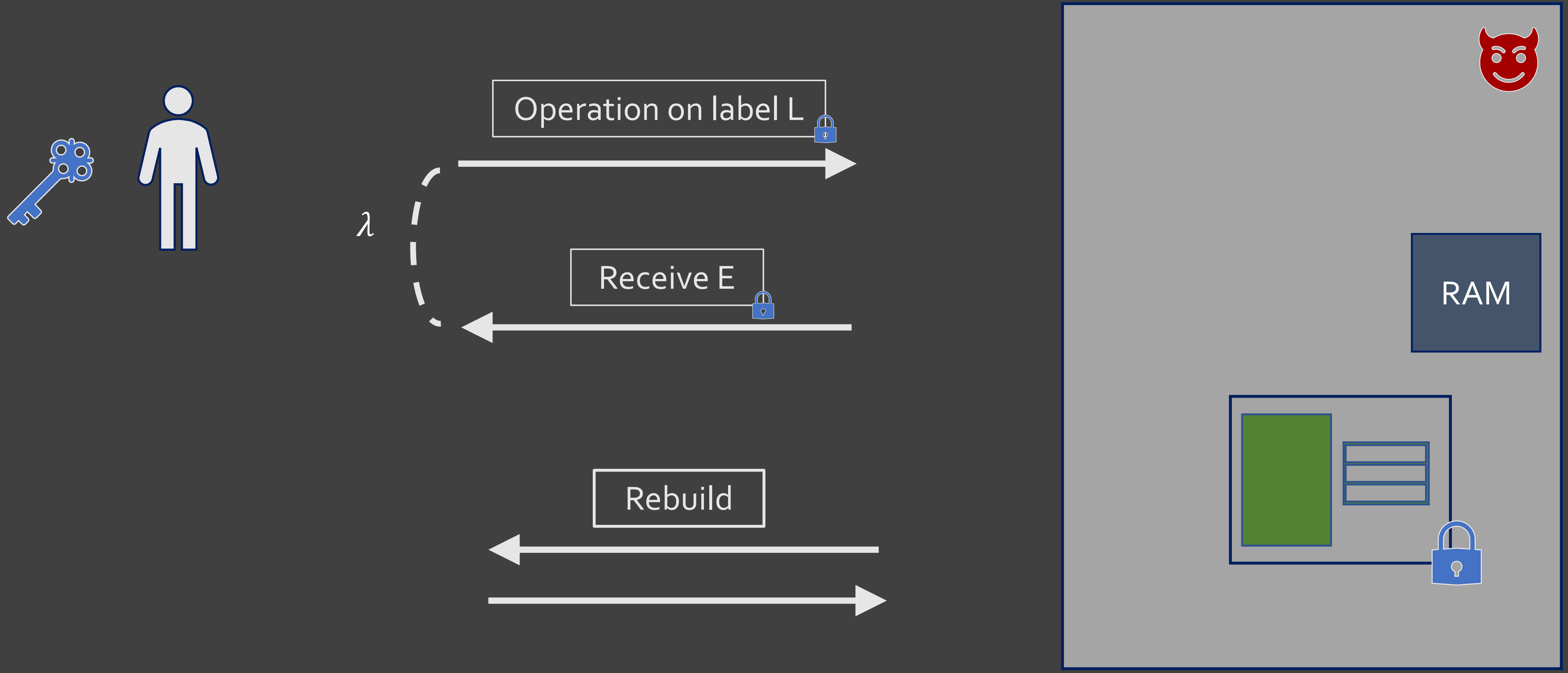
# Dynamic Framework in Action



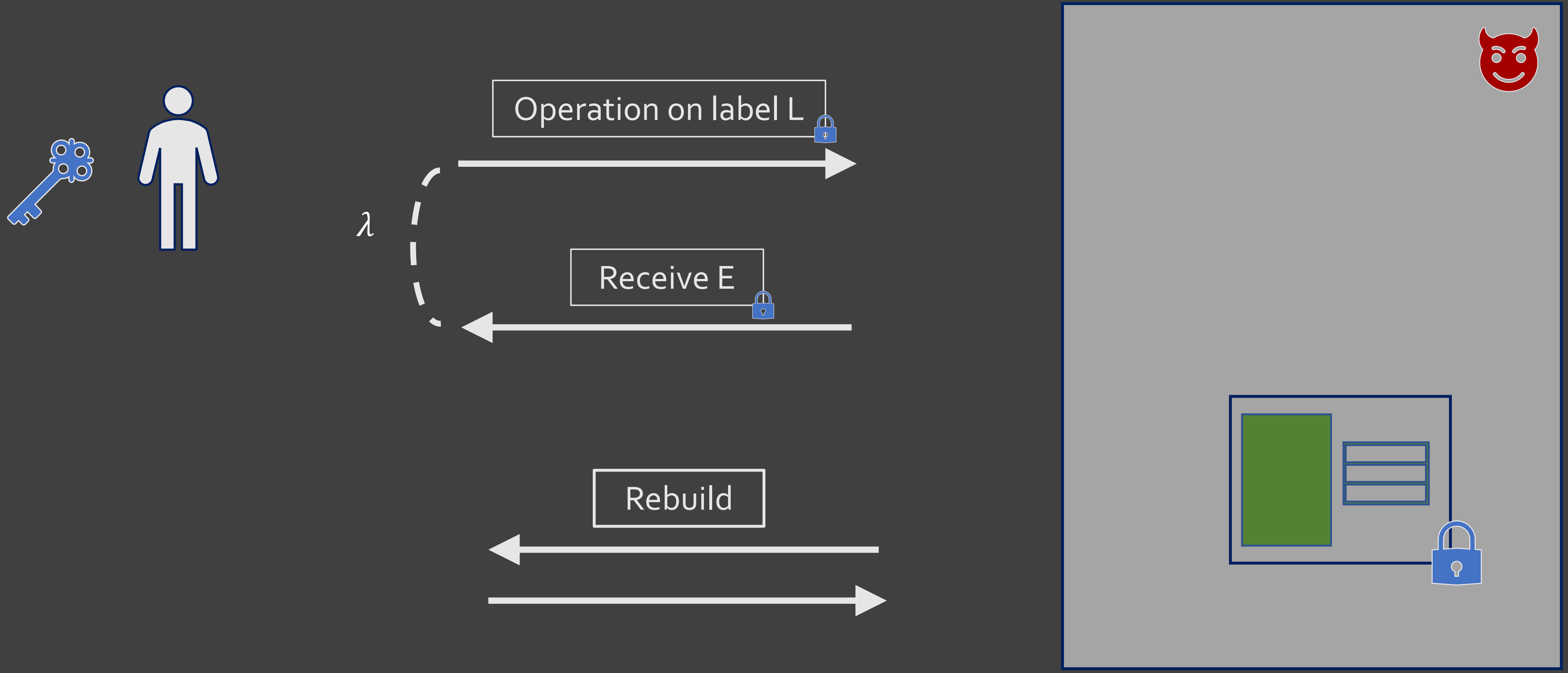
# Dynamic Framework in Action



# Dynamic Framework in Action



# Dynamic Framework in Action



# Efficiency Comparison

Over a sequence of λ=64 Add operations to a multi-map containing $2^{16}$ values

| Efficiency Measure | Our framework applied to AVLH [KM19] | Black-box simulation with Path ORAM [SPS14] | Standard Dynamic EMM [$\pi_{bas}^{dyn}$, CJJ+14] |
|---|---|---|---|
| Client State (Mbits) | | | |
| Server Storage (Mbits) | | | |
| Communication (Mbits) | | | |
| Leakage | | | |

# Efficiency Comparison

Over a sequence of λ=64 Add operations to a multi-map containing $2^{16}$ values

| Efficiency Measure | Our framework applied to AVLH [KM19] | Black-box simulation with Path ORAM [SPS14] | Standard Dynamic EMM [$\pi_{bas}^{dyn}$, CJJ+14] |
|---|---|---|---|
| Client State (Mbits) | 0.486 | 4.78 | 0.066 |
| Server Storage (Mbits) | 44.062 | 52424.7 | 20.992 |
| Communication (Mbits) | 1827.1 | 1995.534 | 10.485 |
| Leakage | Total number of labels, values, max. tuple length<br>Updated number of labels, values, max. tuple length (after λ operations) | (Upper-bound) number of labels, max. tuple length | Volume, Query equality |

# Efficiency Comparison

Over a sequence of λ=64 Add operations to a multi-map containing $2^{16}$ values

| Efficiency Measure | Our framework applied to AVLH [KM19] | Black-box simulation with Path ORAM [SPS14] | Standard Dynamic EMM [$\pi_{bas}^{dyn}$, CJJ+14] |
|---|---|---|---|
| Client State (Mbits) | 0.486 | 4.78 | 0.066 |
| Server Storage (Mbits) | 44.062 | 52424.7 | 20.992 |
| Communication (Mbits) | 1827.1 | 1995.534 | 10.485 |
| Leakage | Total number of labels, values, max. tuple length<br>Updated number of labels, values, max. tuple length (after λ operations) | (Upper-bound) number of labels, max. tuple length | Volume, Query equality |

# Efficiency Comparison

Over a sequence of λ=64 Add operations to a multi-map containing $2^{16}$ values

| Efficiency Measure | Our framework applied to AVLH [KM19] | Black-box simulation with Path ORAM [SPS14] | Standard Dynamic EMM [$\pi_{bas}^{dyn}$, CJJ+14] |
|---|---|---|---|
| Client State (Mbits) | 0.486 | 4.78 | 0.066 |
| Server Storage (Mbits) | 44.062 | 52424.7 | 20.992 |
| Communication (Mbits) | 1827.1 | 1995.534 | 10.485 |
| Leakage | Total number of labels, values, max. tuple length<br>Updated number of labels, values, max. tuple length (after λ operations) | (Upper-bound) number of labels, max. tuple length | Volume, Query equality |

# Dynamic Framework: In Summary

- We construct a **dynamic operation equality suppressing** framework, answering an open question [KMO18]

- We apply our framework to AVLH [KM19] and PBS [KMO18] to produce **three new fully-dynamic almost-zero leakage** STE schemes

- We prove that for certain natural assumptions, our schemes are **asymptotically more efficient** than black-box ORAM simulation

- Please see our paper for more details!