Efficient and Timely Revocation of V2X Credentials

Artifact Evaluated
NDSS SYMPOSIUM
Available
Functional
Reproduced

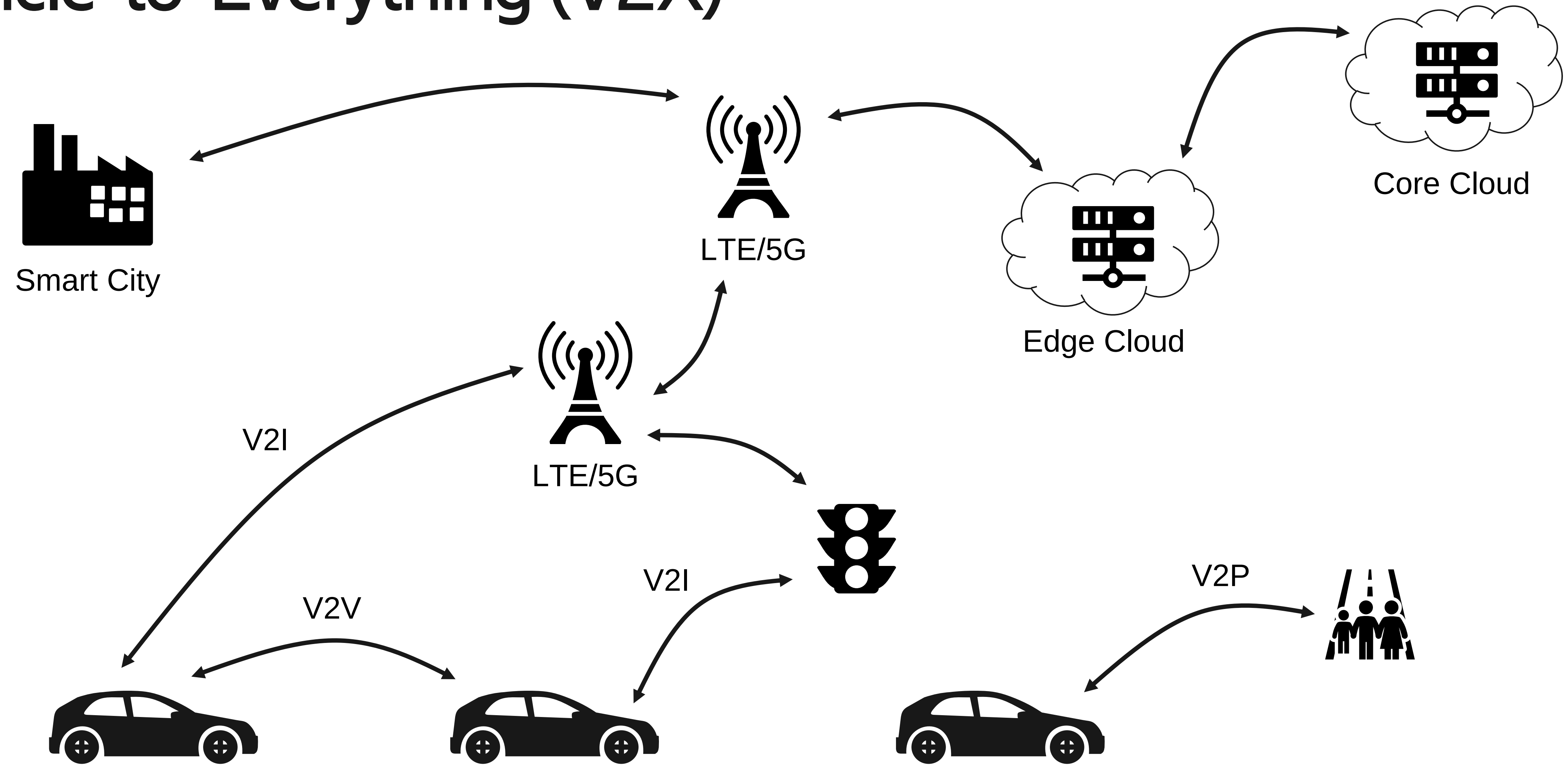
Gianluca Scopelliti, Christoph Baumann, Fritz Alder, Eddy Truyen, Jan Tobias Mühlberg.

Network and Distributed System Security (NDSS) Symposium 2024. San Diego, CA.

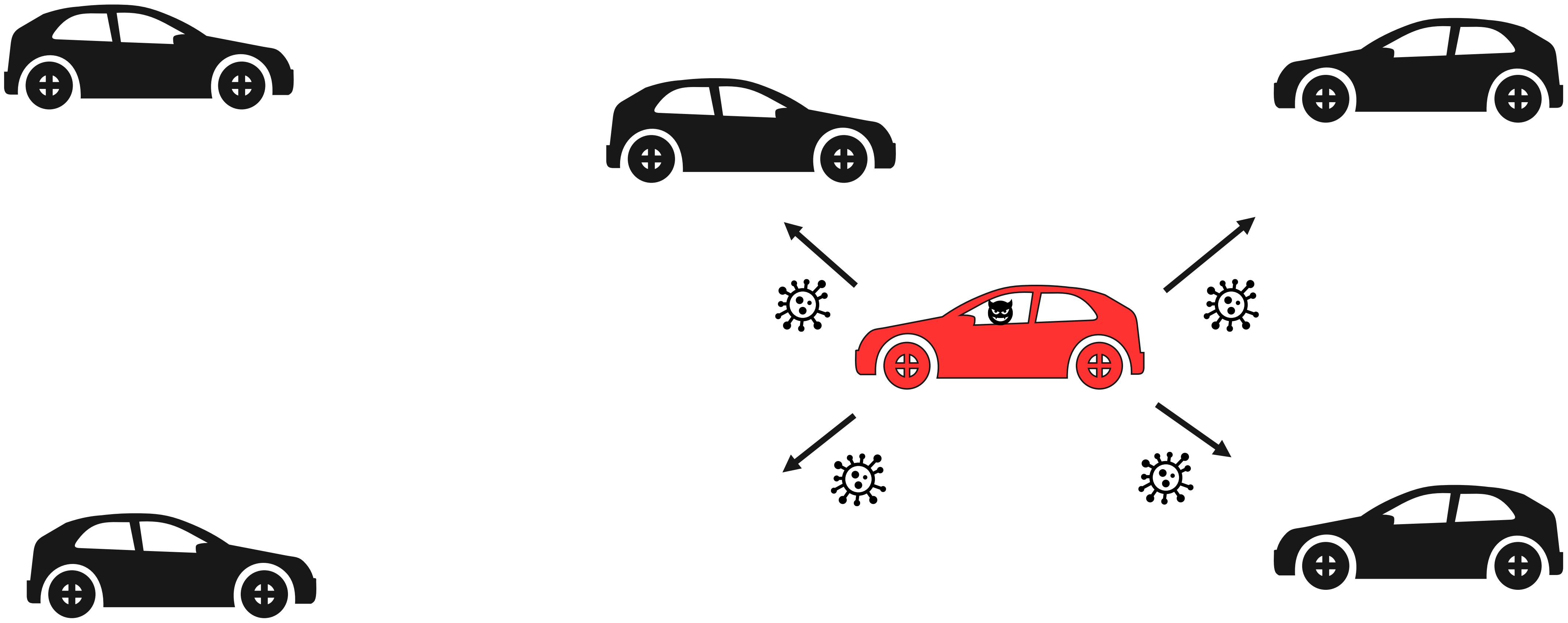
gianluca.scopelliti@ericsson.com

ERICSSON KU LEUVEN DistriNet ULB CYBERSECURITY RESEARCH CENTER

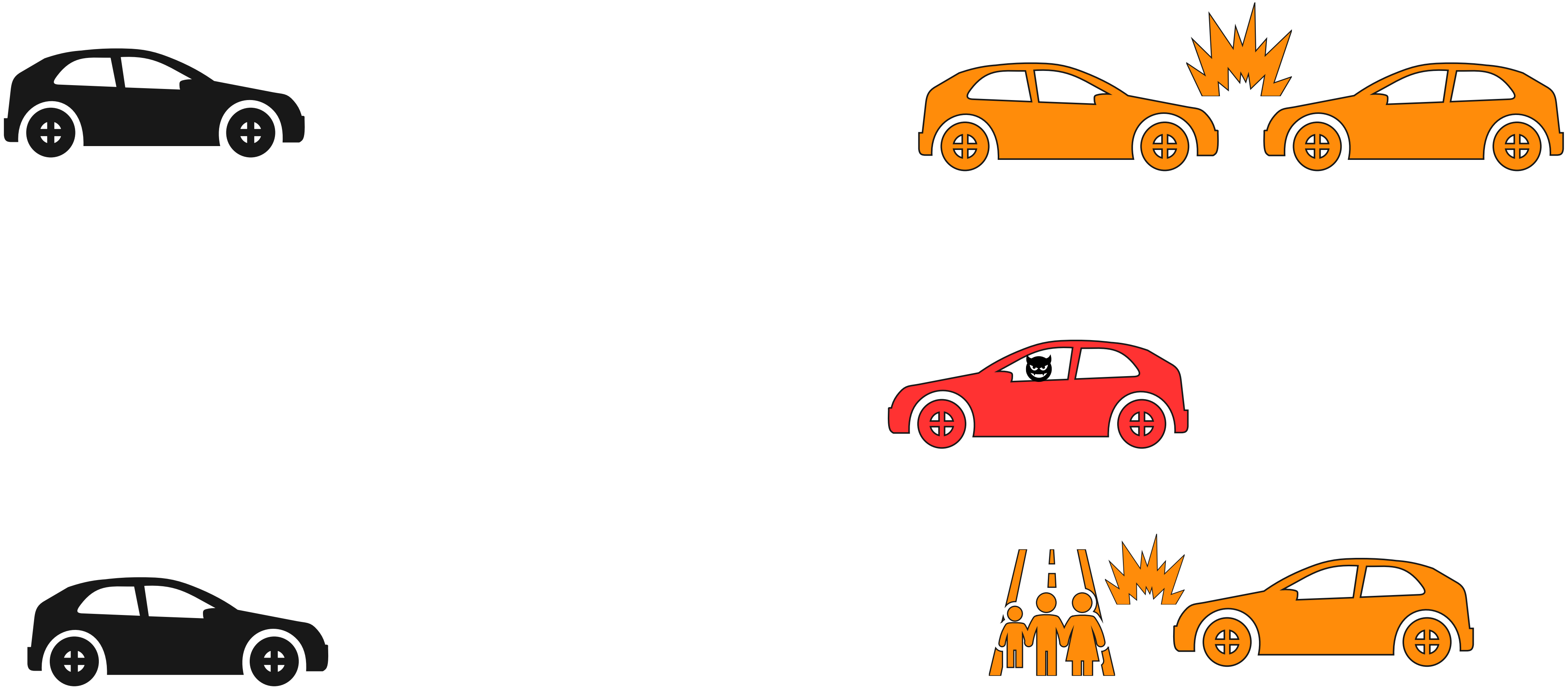
Vehicle-to-Everything (V2X)



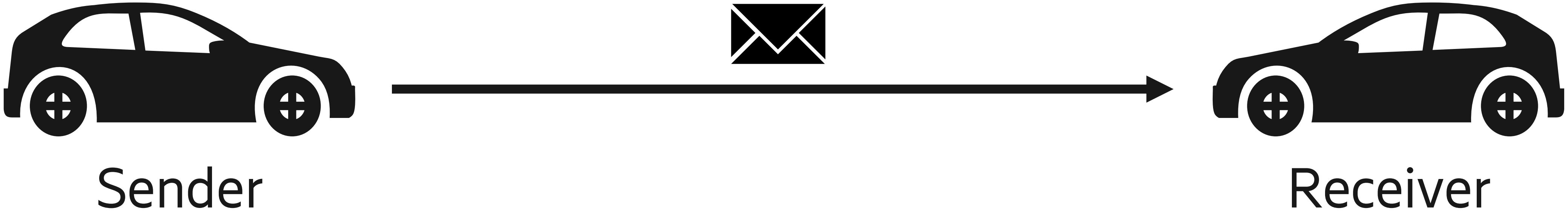
What if a vehicle is malicious?



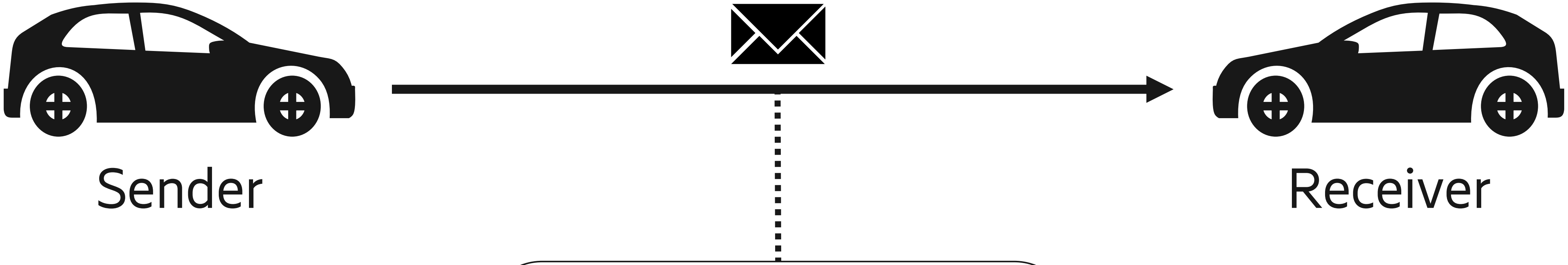
What if a vehicle is malicious?



Vehicle communication in V2X needs to be properly protected

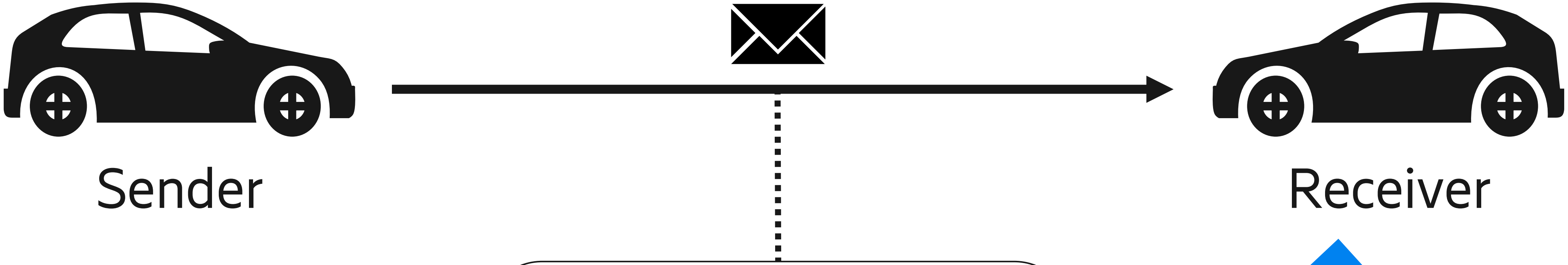


Vehicle communication in V2X needs to be properly protected



```
speed: 90
location: 23.90437, -140.51174
direction: NE
...

timestamp: 1709025600
pseudonym: deadbeef
expiration: 1711531200
signature: <some bytes>
```

Vehicle communication in V2X needs to be properly protected

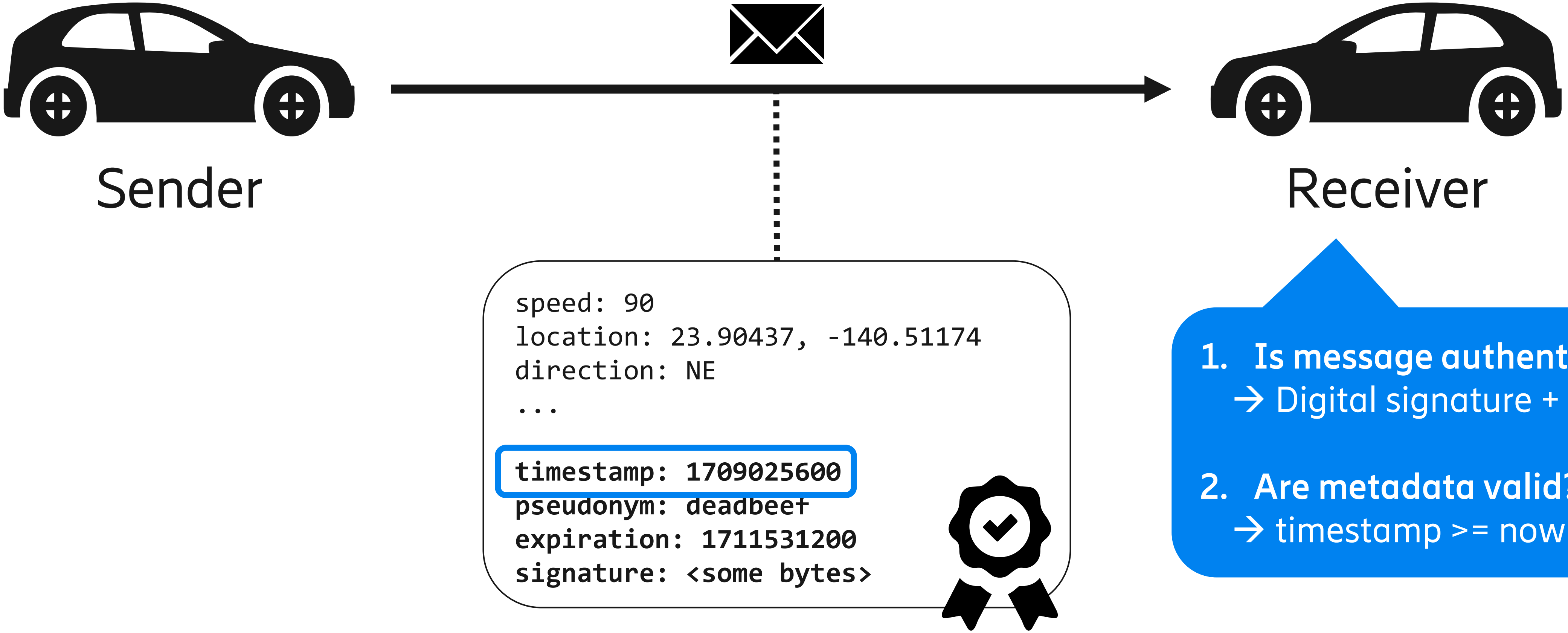


```
speed: 90
location: 23.90437, -140.51174
direction: NE
...

timestamp: 1709025600
pseudonym: deadbeef
expiration: 1711531200
signature: <some bytes>
```

1. **Is message authentic?**
   → Digital signature + identity

Vehicle communication in V2X needs to be properly protected

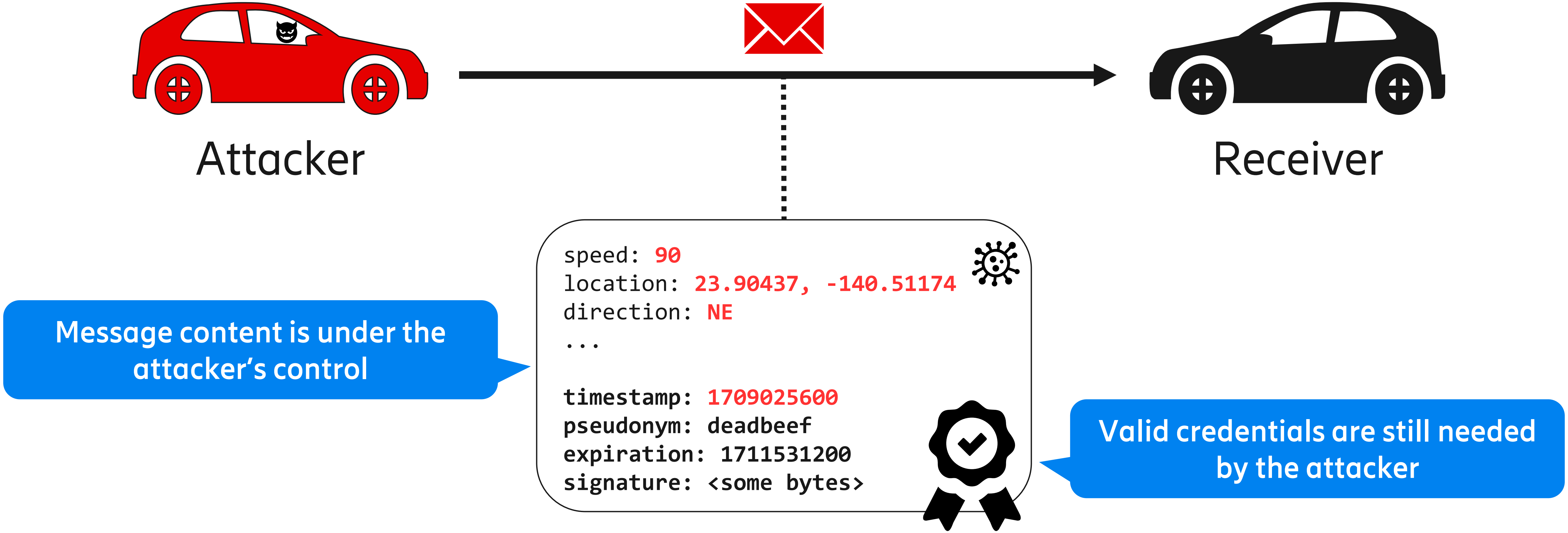
T_V : **tolerance** for network messages



```
speed: 90
location: 23.90437, -140.51174
direction: NE
...

timestamp: 1709025600
pseudonym: deadbeef
expiration: 1711531200
signature: <some bytes>
```

1. **Is message authentic?**
   → Digital signature + identity
2. **Are metadata valid?**
   → timestamp >= now() - T_V

Malicious participants may spread false information and cause accidents



```
speed: 90
location: 23.90437, -140.51174
direction: NE
...

timestamp: 1709025600
pseudonym: deadbeef
expiration: 1711531200
signature: <some bytes>
```

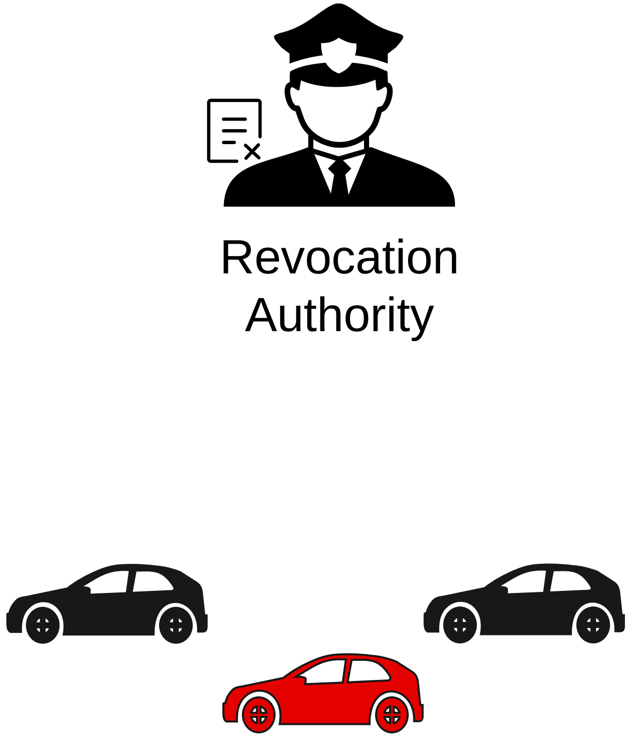
Message content is under the attacker's control

Valid credentials are still needed by the attacker

State of the art in revocation schemes for V2X

State of the art in revocation schemes for V2X

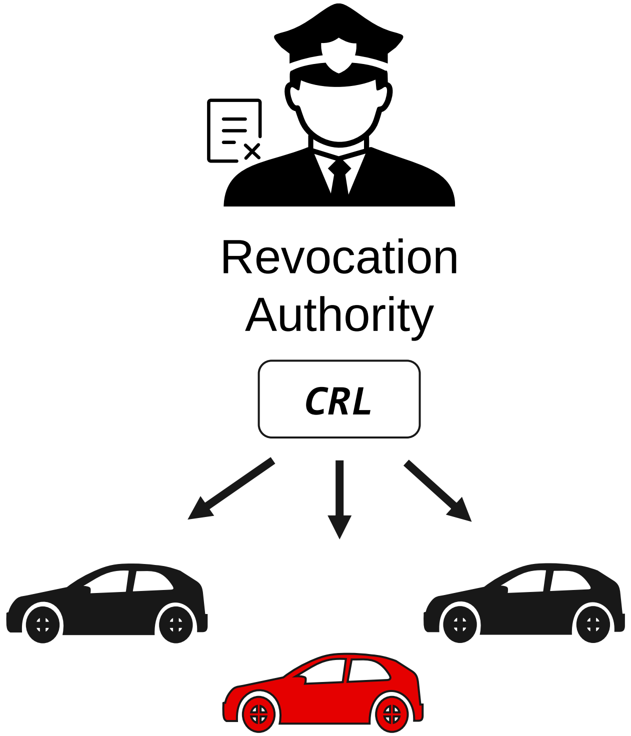
Active revocation (IEEE 1609.2.1 – SCMS [1])



[1] IEEE Std 1609.2.1-2022 "IEEE WAVE - Certificate Management Interfaces for End Entities”

State of the art in revocation schemes for V2X

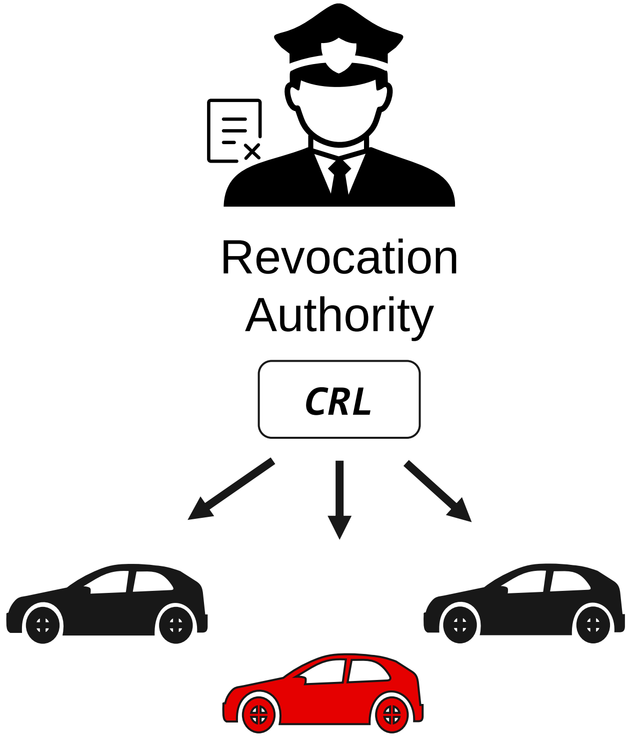
Active revocation (IEEE 1609.2.1 – SCMS [1])



[1] IEEE Std 1609.2.1-2022 "IEEE WAVE - Certificate Management Interfaces for End Entities"

State of the art in revocation schemes for V2X

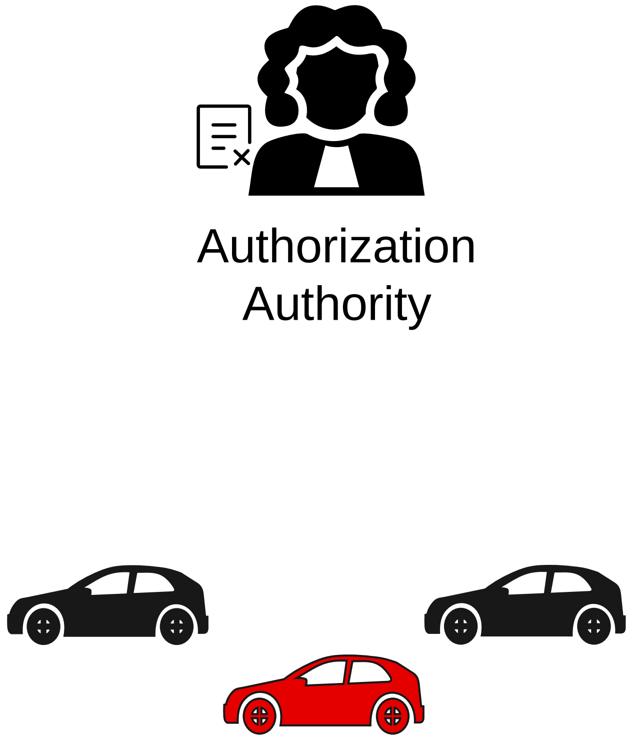
Active revocation (IEEE 1609.2.1 – SCMS [1])



Revocation Authority

CRL

Passive revocation (ETSI TS 102 941 [2])



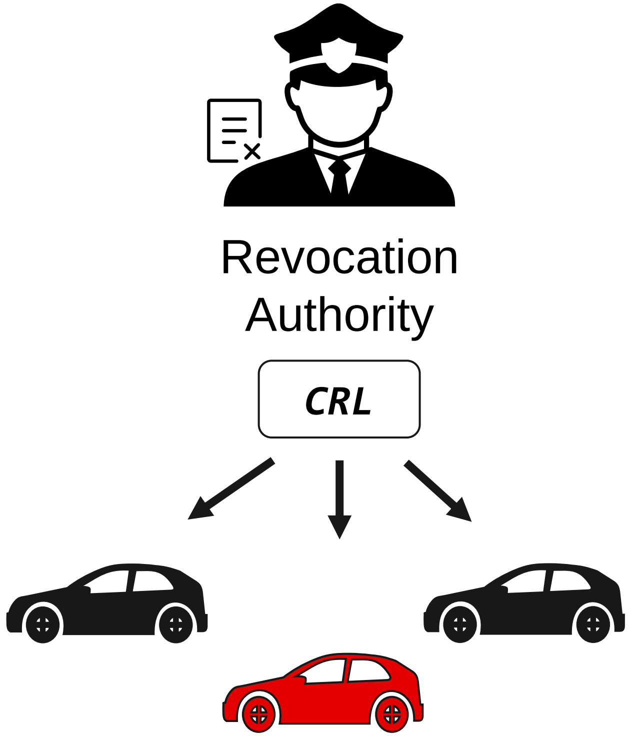
Authorization Authority

[1] IEEE Std 1609.2.1-2022 "IEEE WAVE - Certificate Management Interfaces for End Entities”

[2] ETSI TS 102 940 version 2.1.1, “Intelligent Transport Systems (ITS); Security, ITS communications security architecture and security management”

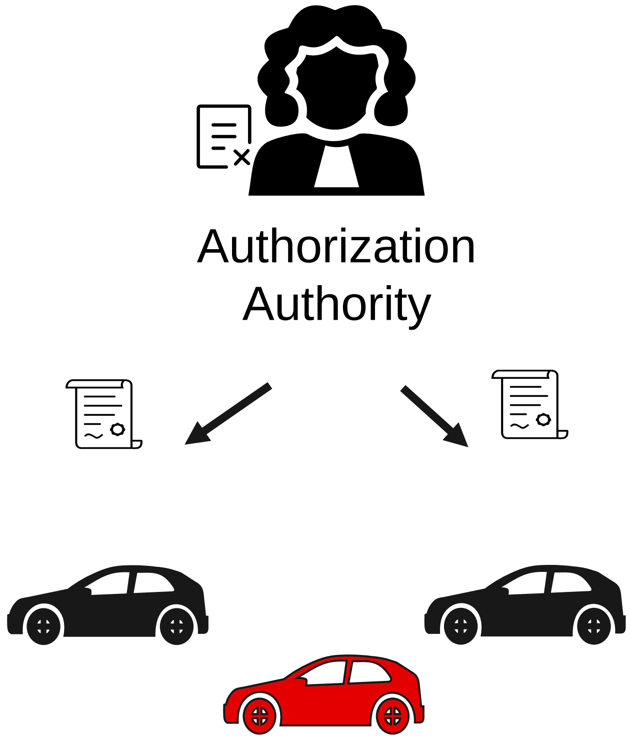
State of the art in revocation schemes for V2X

Active revocation (IEEE 1609.2.1 – SCMS [1])


Revocation Authority

CRL

Passive revocation (ETSI TS 102 941 [2])

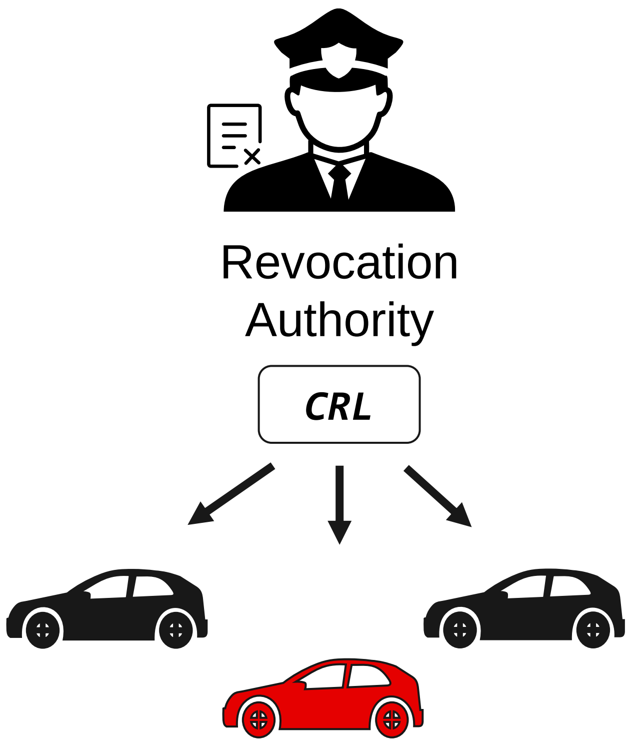

Authorization Authority

[1] IEEE Std 1609.2.1-2022 "IEEE WAVE - Certificate Management Interfaces for End Entities”

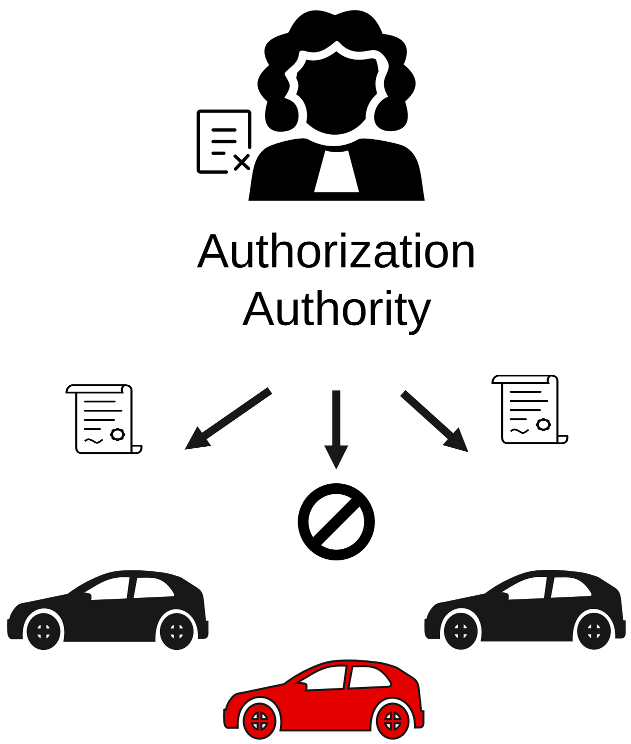
[2] ETSI TS 102 940 version 2.1.1, “Intelligent Transport Systems (ITS); Security, ITS communications security architecture and security management”

State of the art in revocation schemes for V2X

Active revocation (IEEE 1609.2.1 – SCMS [1])



Passive revocation (ETSI TS 102 941 [2])

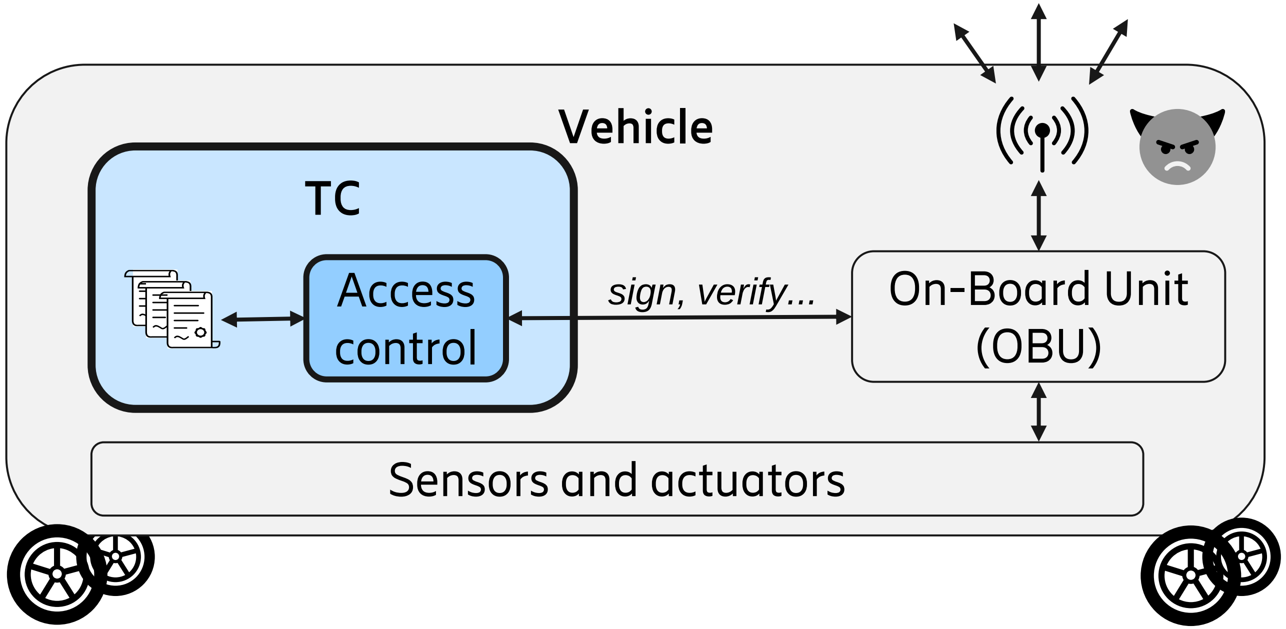


[1] IEEE Std 1609.2.1-2022 "IEEE WAVE - Certificate Management Interfaces for End Entities"

[2] ETSI TS 102 940 version 2.1.1, "Intelligent Transport Systems (ITS); Security, ITS communications security architecture and security management"

Putting trust in vehicles: Trusted Computing and Self-Revocation

- Vehicles equipped with a **Trusted Component (TC)**
- Credentials + message metadata are managed by the TC
- Academic proposals leverage TPMs and Direct Anonymous Attestation (DAA) [3]

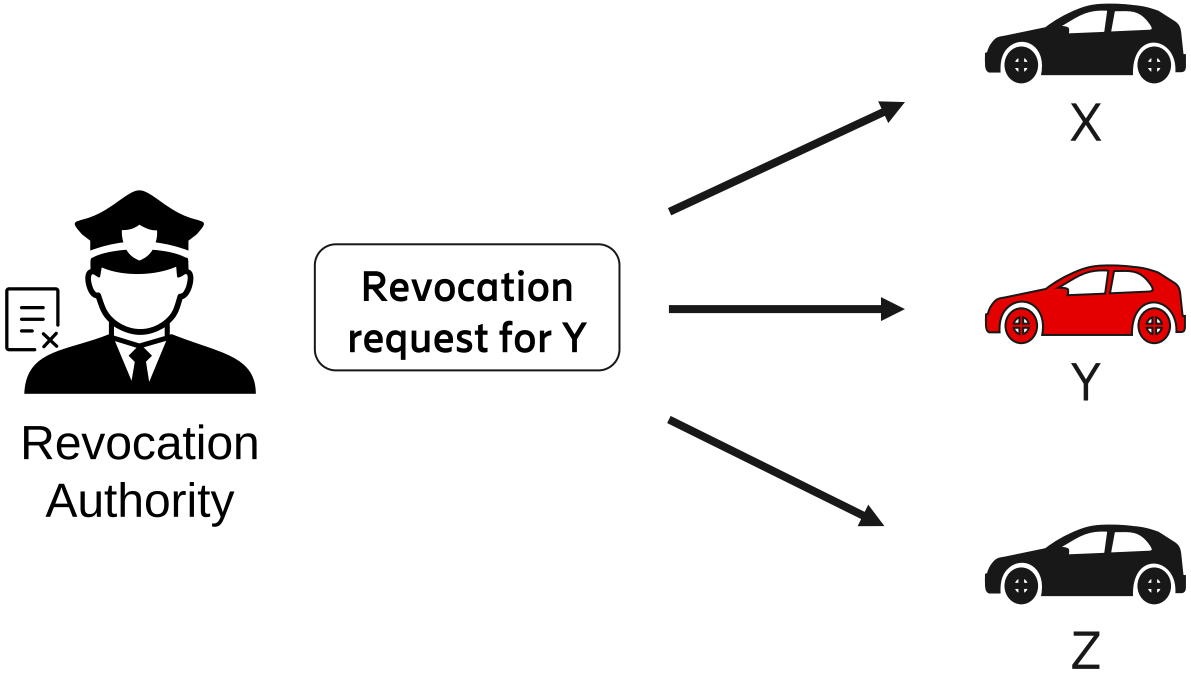


[3] Larsen et al., "Direct Anonymous Attestation on the Road: Efficient and Privacy-Preserving Revocation in C-ITS", WiSec '21.

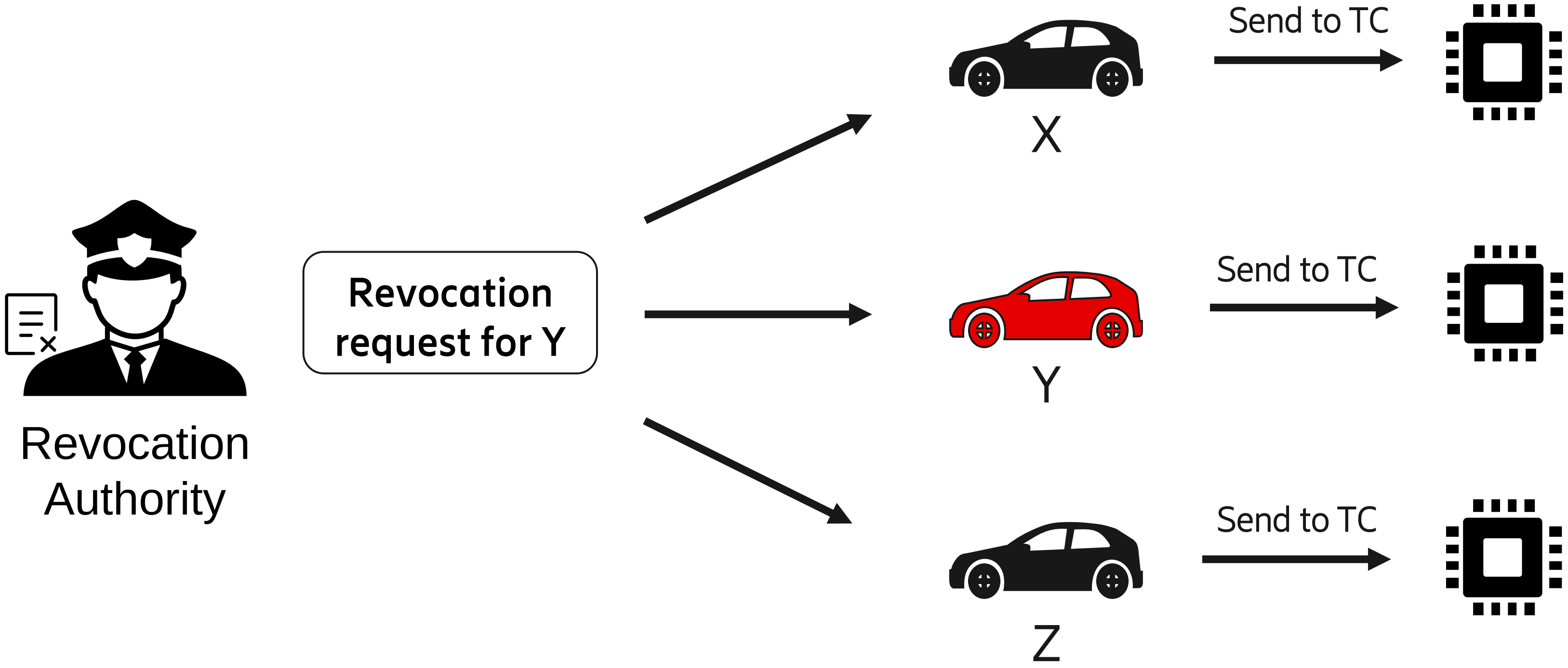
Self-revocation in practice

X
Y
Z
Revocation Authority

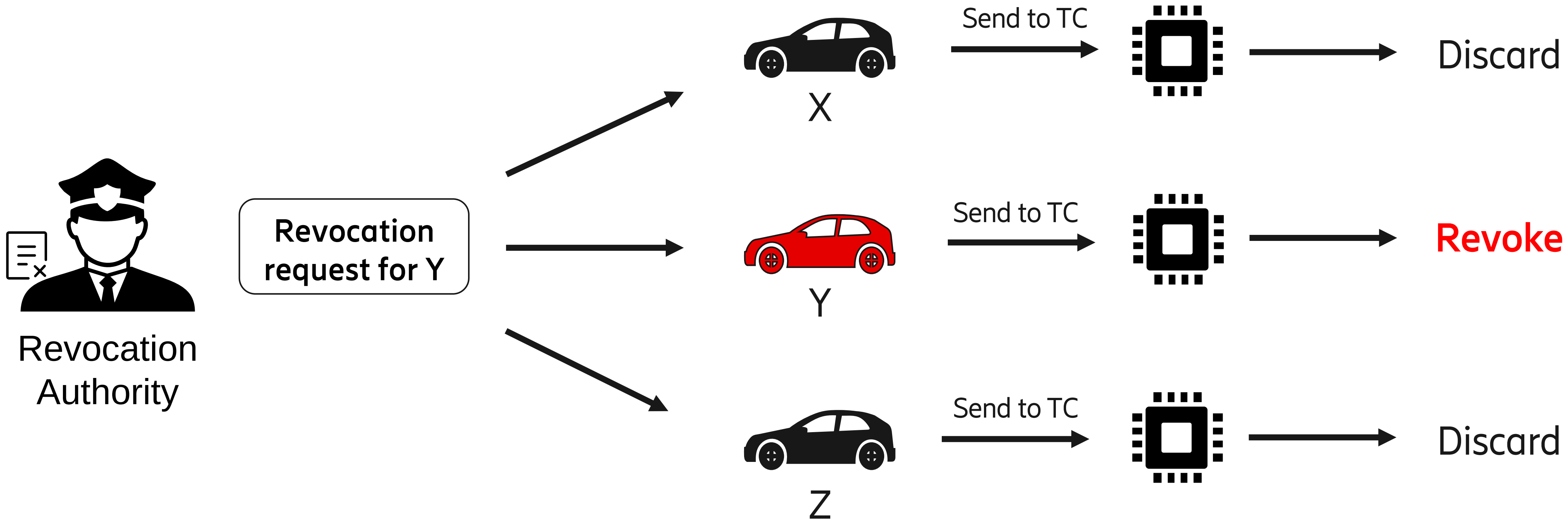
Self-revocation in practice



Self-revocation in practice

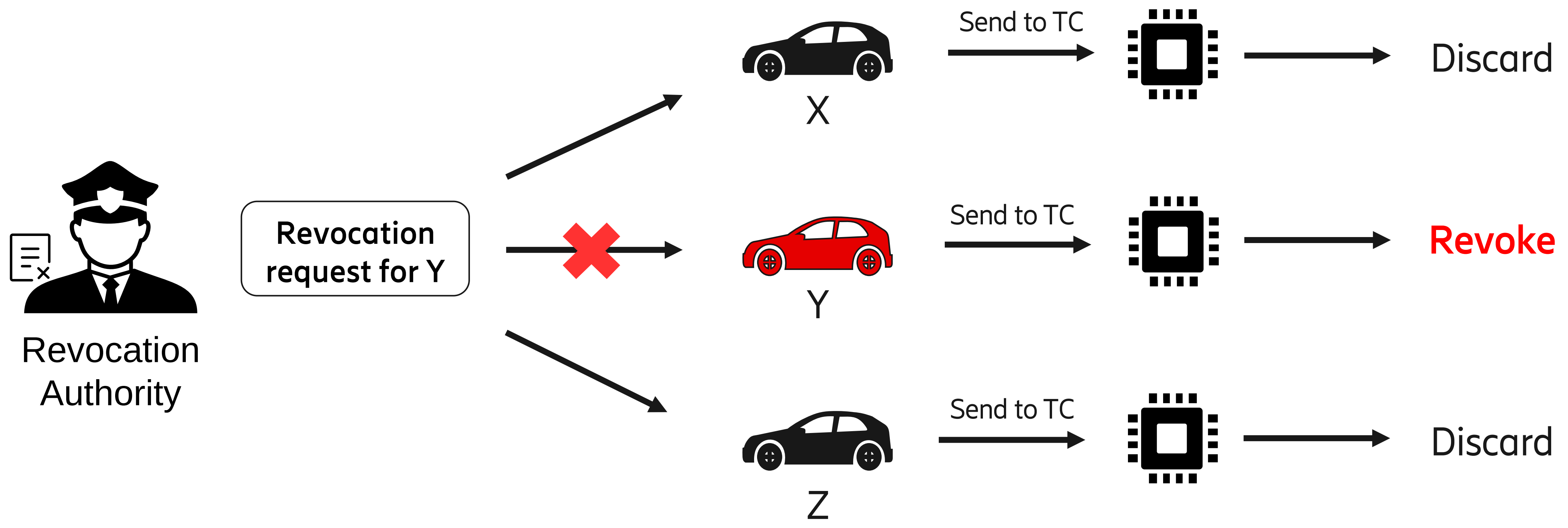


Self-revocation in practice



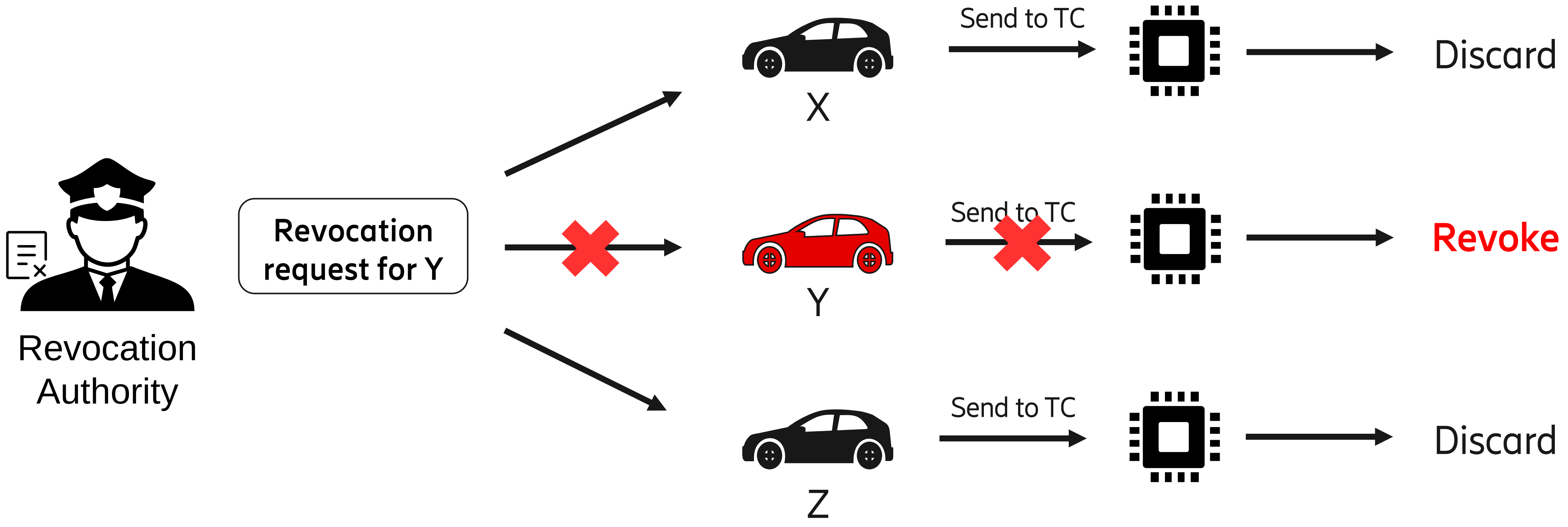
Self-revocation in practice

Dolev-Yao attacker



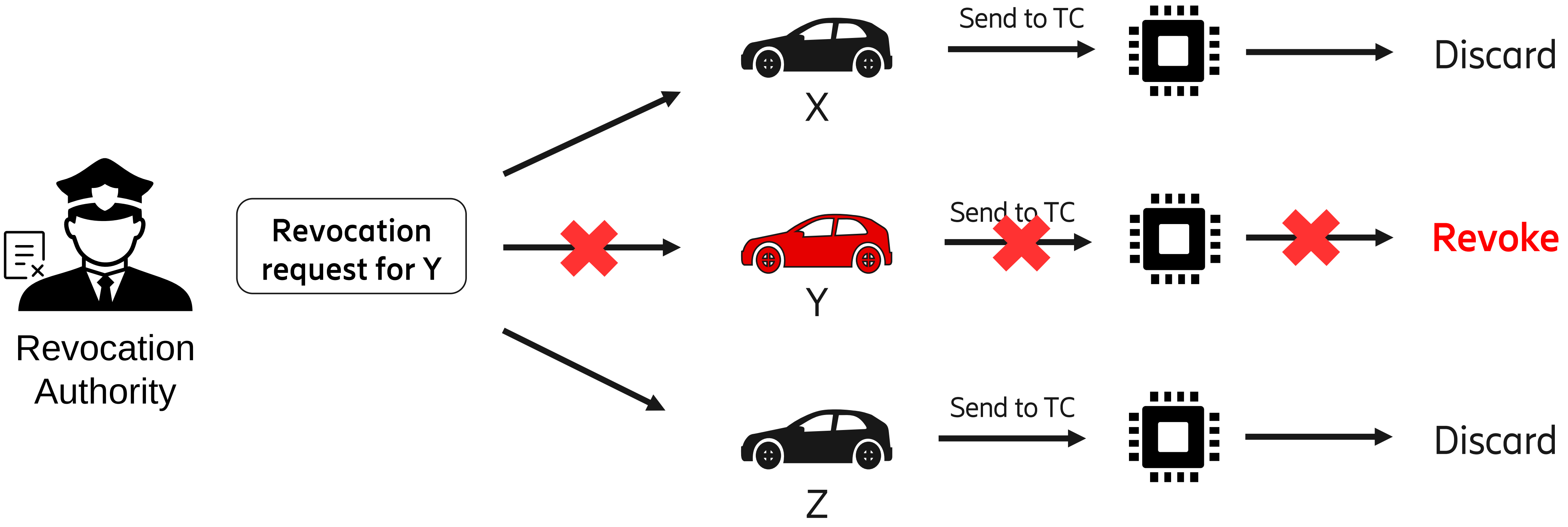
Self-revocation in practice

Dolev-Yao attacker



Self-revocation in practice

Dolev-Yao attacker



Revocation Authority

Revocation request for Y

Goals

Security

Guaranteed revocation with fixed upper bound

⬇

Formal verification

Goals

Security

Guaranteed revocation with fixed upper bound

↓

Formal verification

Usability

Resistant against network delays and interruptions

↓

Simulation

Goals

Security

Guaranteed revocation with fixed upper bound

↓

Formal verification

Usability

Resistant against network delays and interruptions

↓

Simulation

Efficiency

Small overhead, scalable with large number of revocations

↓

Probabilistic model

Constraints

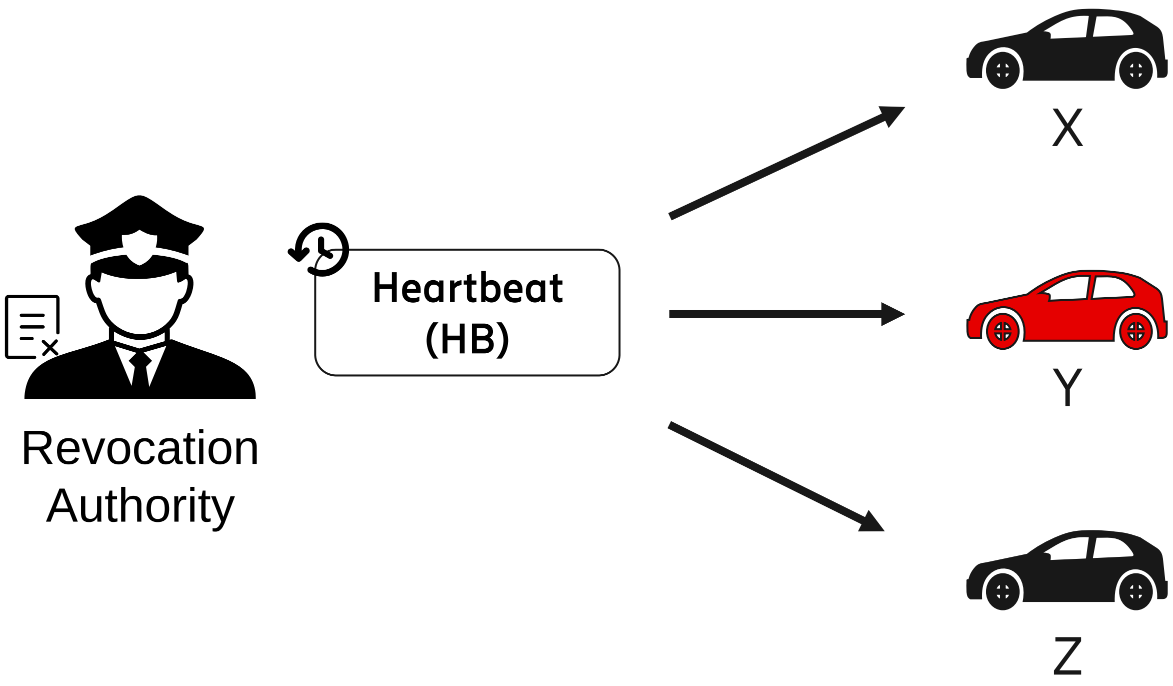
- TC does **not** have access to a trusted time source
  - Common issue with most TEEs

Constraints

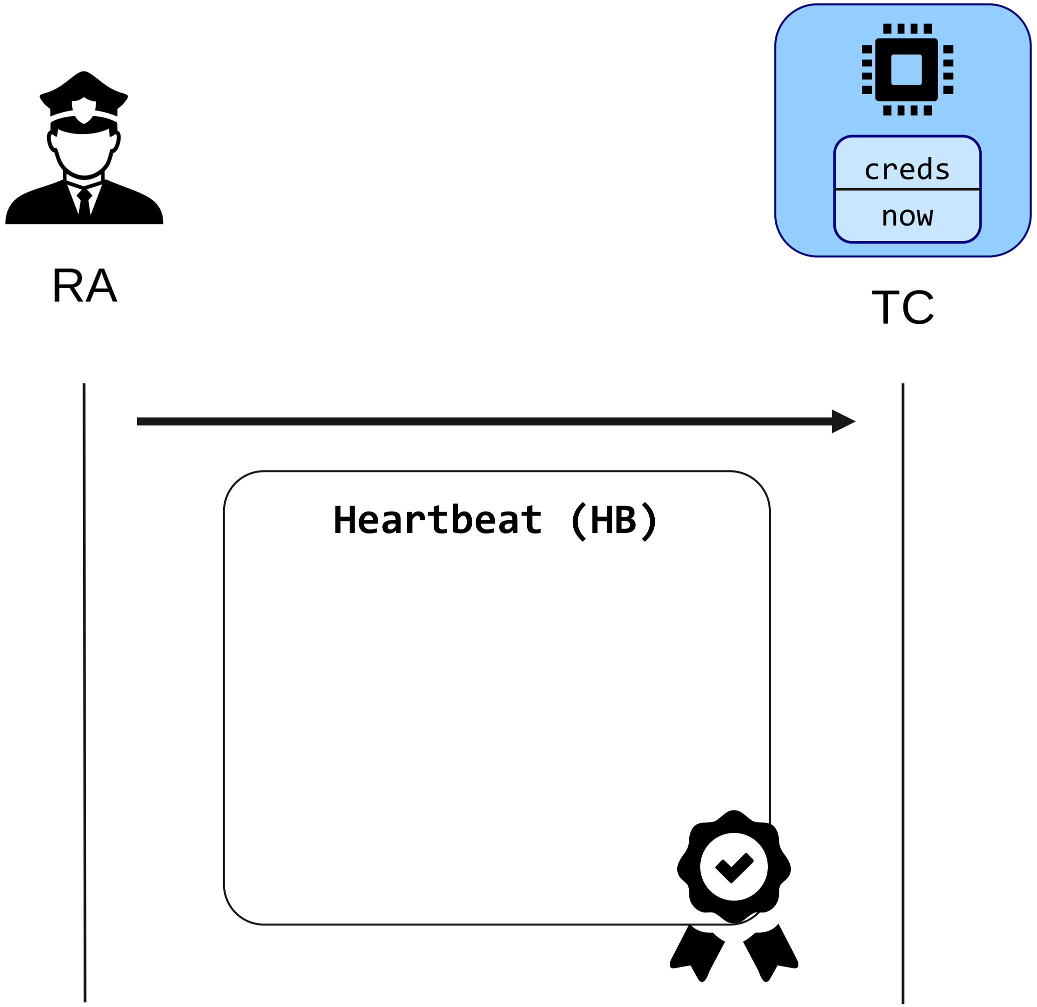
- TC does **not** have access to a trusted time source
  - Common issue with most TEEs

- TC is a **passive device**
  - Process request from untrusted host (e.g., *sign*), return response

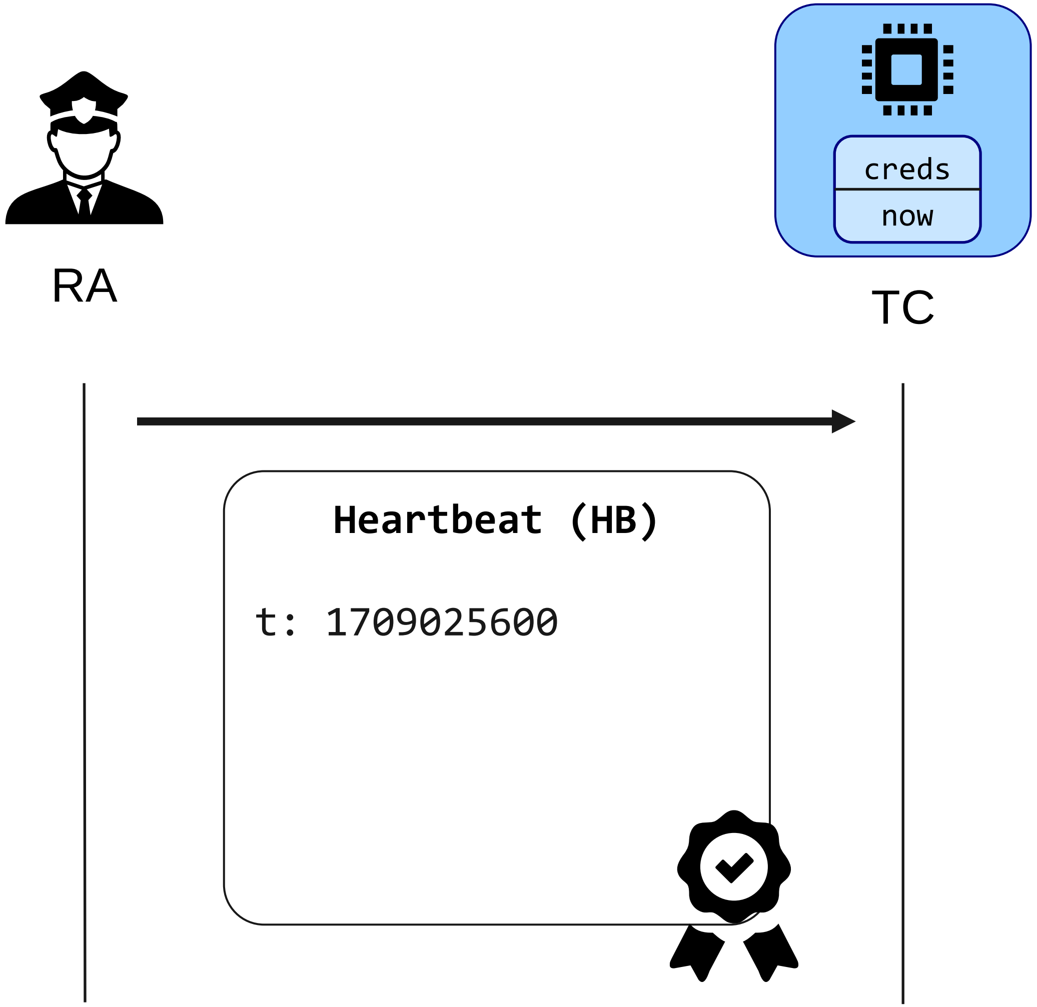
Our approach: periodic heartbeats (HBs)



Processing a HB



Processing a HB



Heartbeat (HB)

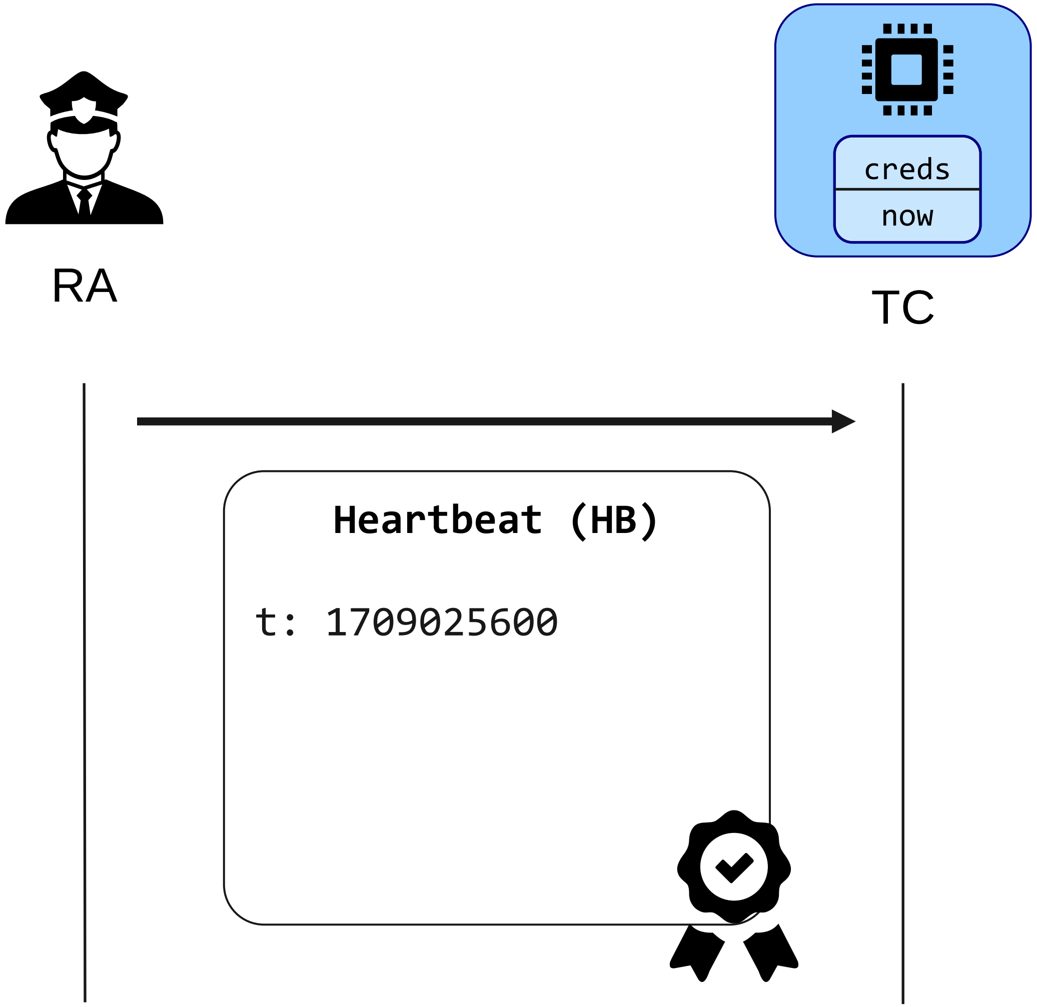
t: 1709025600

Signature check

Freshness check

$t >= \mathrm{now} - T_v$

Processing a HB



Signature check

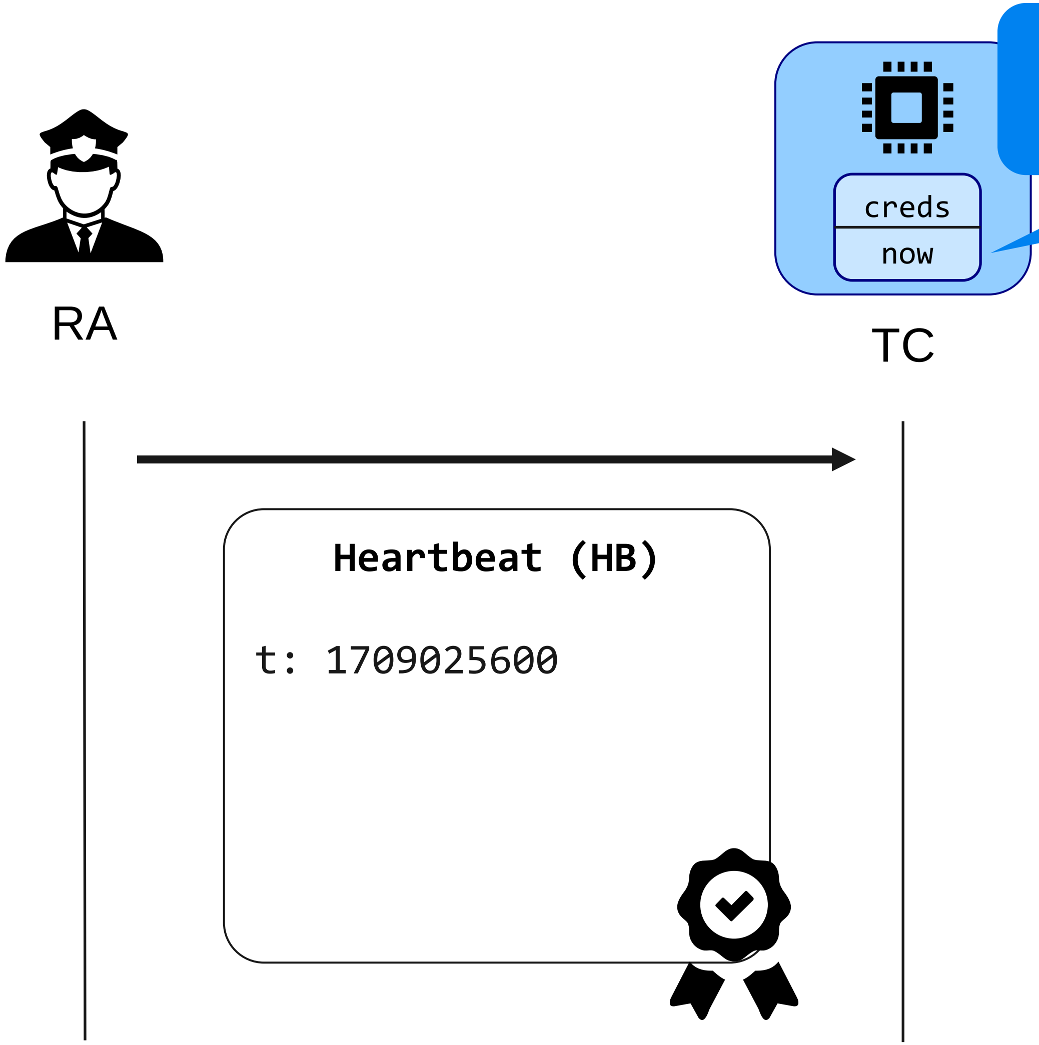
Freshness check

$t >= \text{now} - T_v$

Time synchronization

$\text{now} = t$

Processing a HB



RA

TC

creds

now

Heartbeat (HB)

t: 1709025600

Initial timestamp (epoch) given during initialization (e.g., enrolment)

check

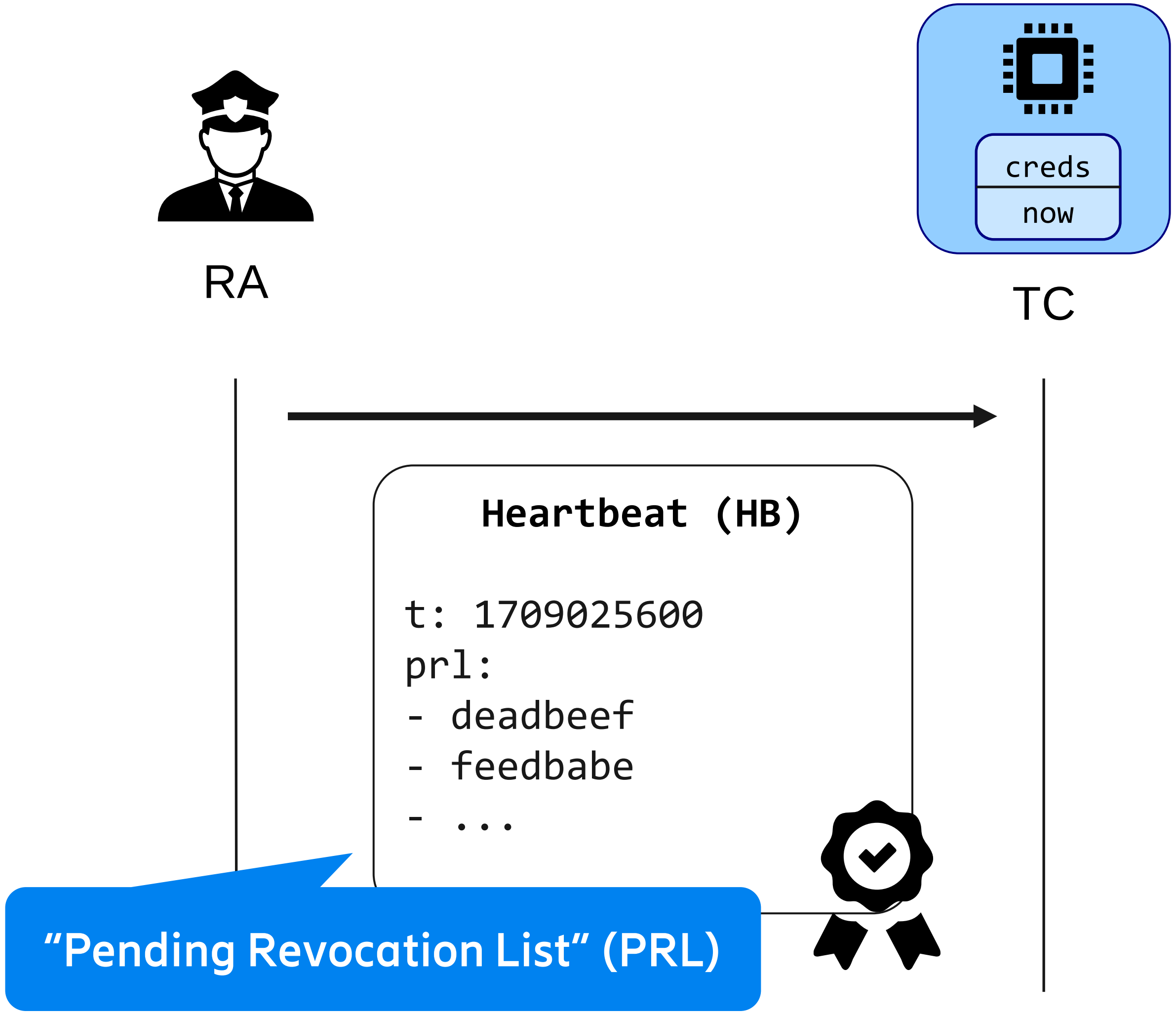
Freshness check

`t >= now - `T_v

Time synchronization

`now = t`

Processing a HB



Heartbeat (HB)

```
t: 1709025600
prl:
- deadbeef
- feedbabe
- ...
```

"Pending Revocation List" (PRL)

Signature check

Freshness check

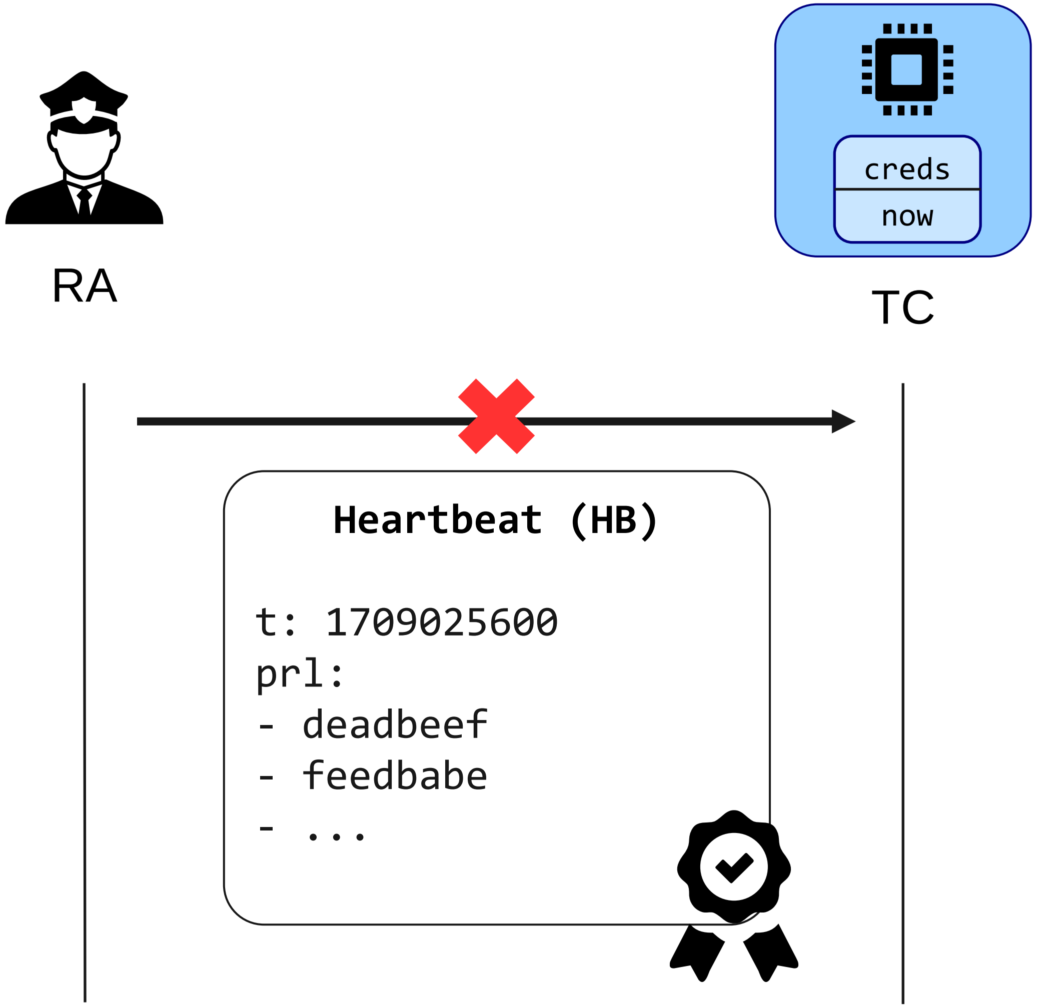
$t >= now - T_v$

Time synchronization

$now = t$

Revocation check

(Not) Processing a HB



Heartbeat (HB)

```
t: 1709025600
prl:
- deadbeef
- feedbabe
- ...
```

Signature check

Freshness check

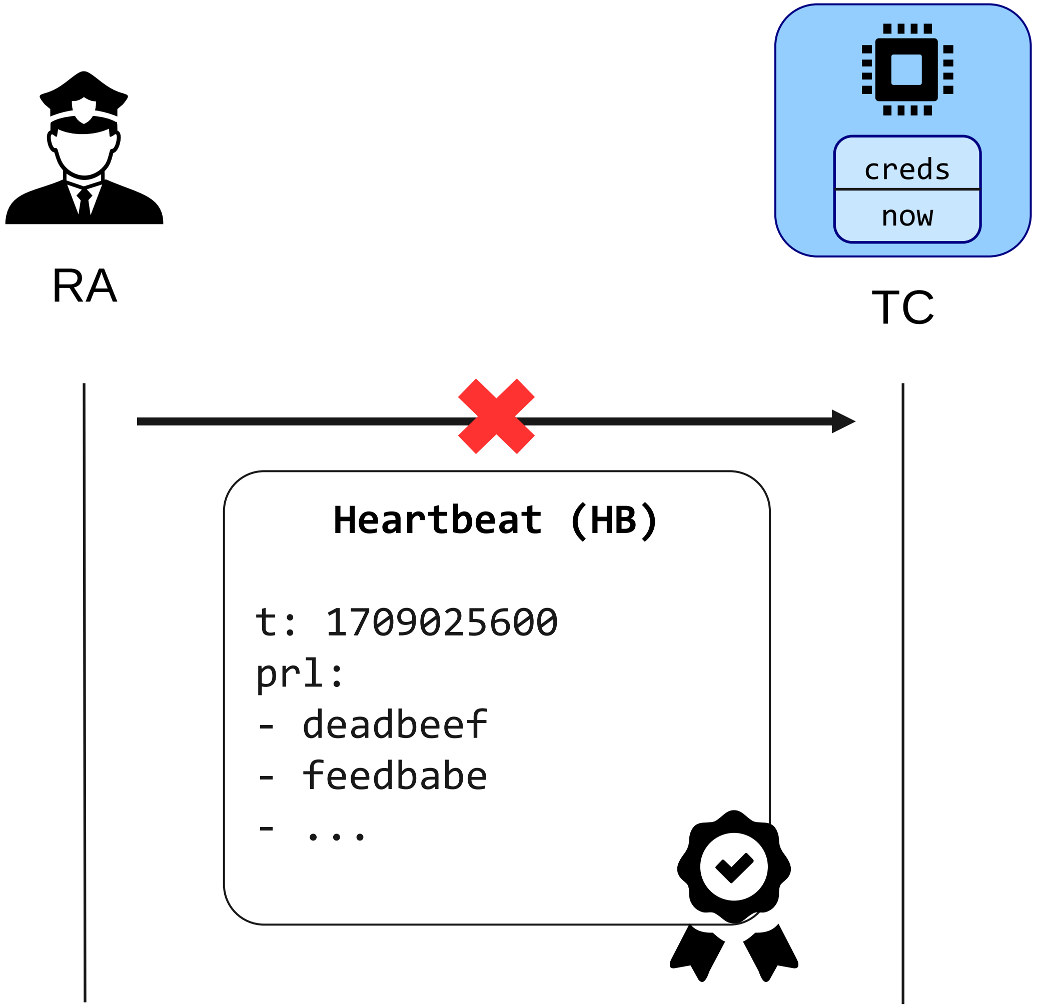
$t >= now - T_v$

Time synchronization

$now = t$

Revocation check

(Not) Processing a HB



```
Heartbeat (HB)

t: 1709025600
prl:
- deadbeef
- feedbabe
- ...
```

Signature check

Freshness check

t >= now – T_v

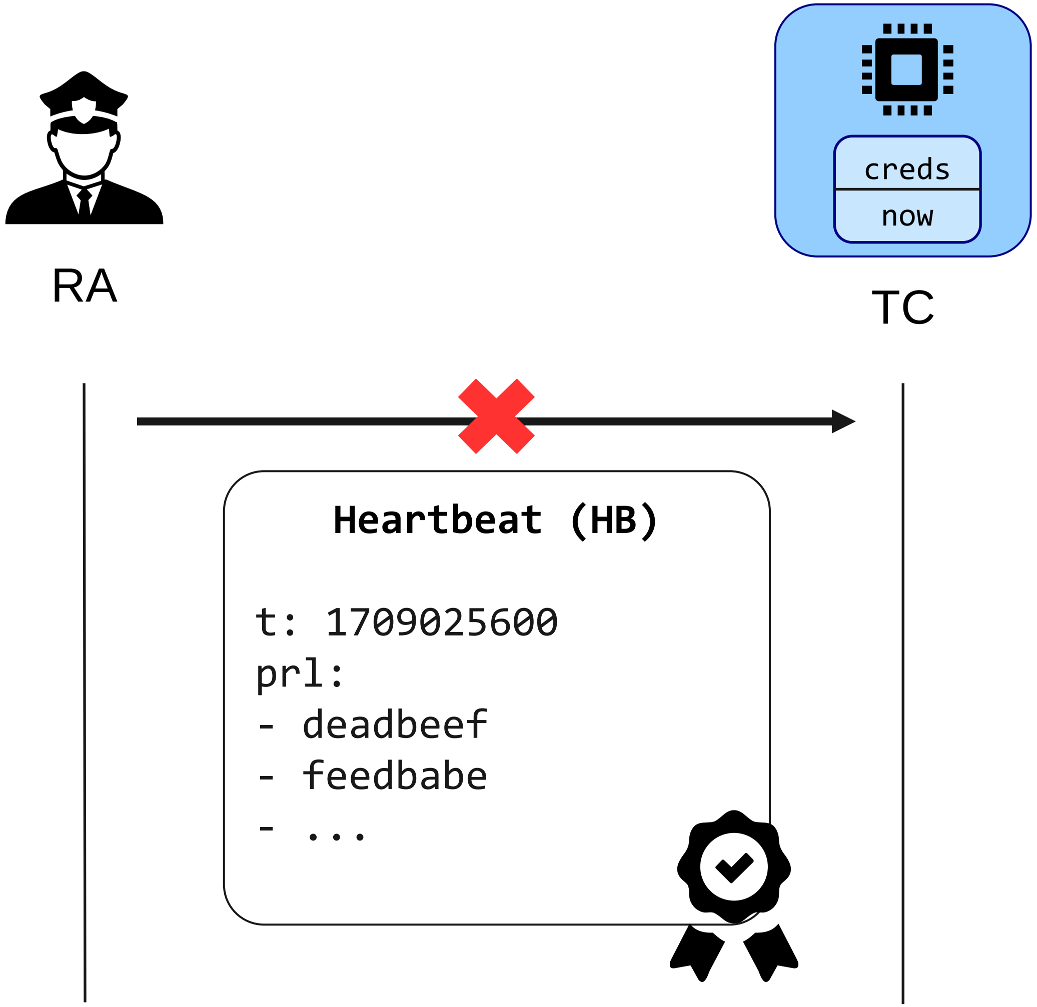
Time synchronization

now = t

✗ Revocation check

Good for attacker: credentials remain valid

(Not) Processing a HB



Heartbeat (HB)

```
t: 1709025600
prl:
- deadbeef
- feedbabe
- ...
```

Signature check

Freshness check

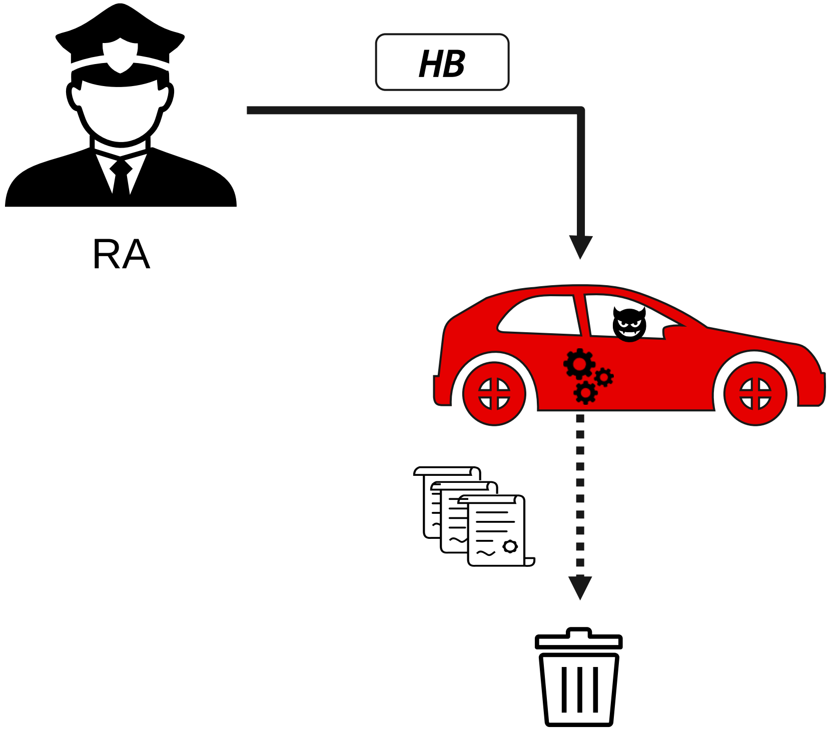
Bad for attacker: TC not in sync with other vehicles

Time synchronization

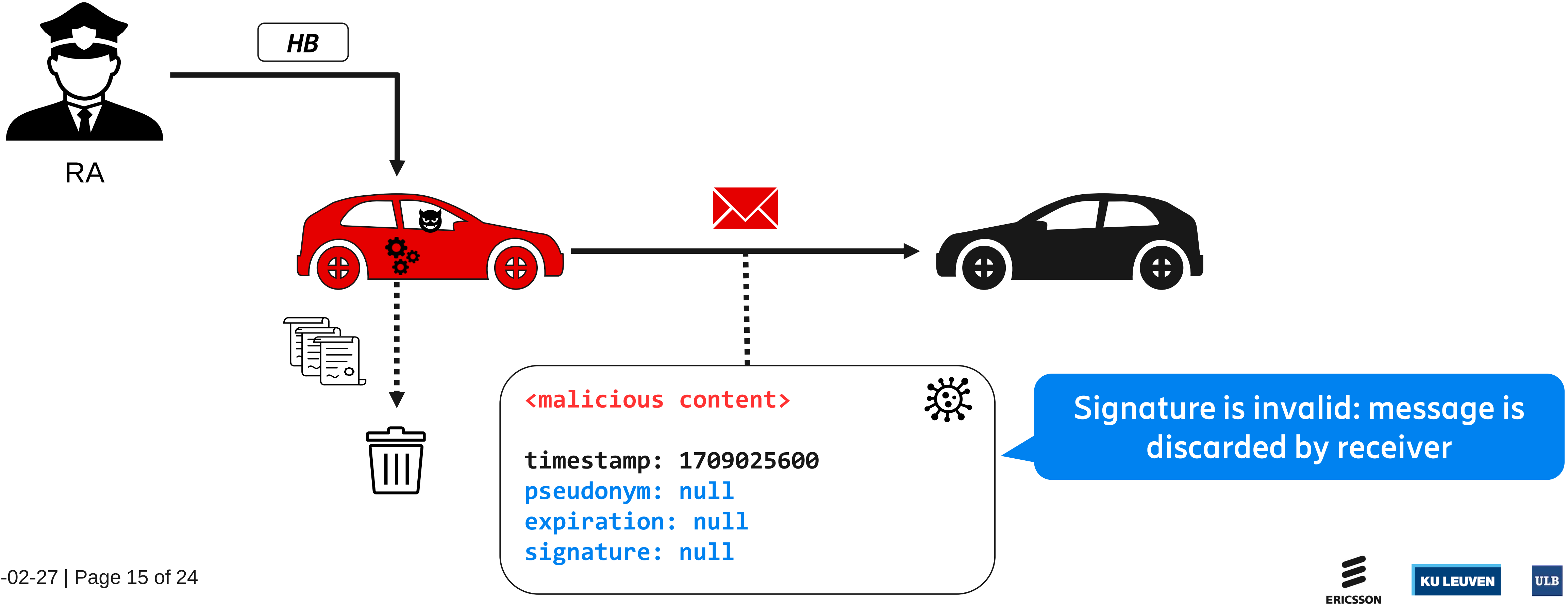
`now = t`

Revocation check

“Cooperative” attacker: HBs are forwarded to the TC and credentials self-revoked



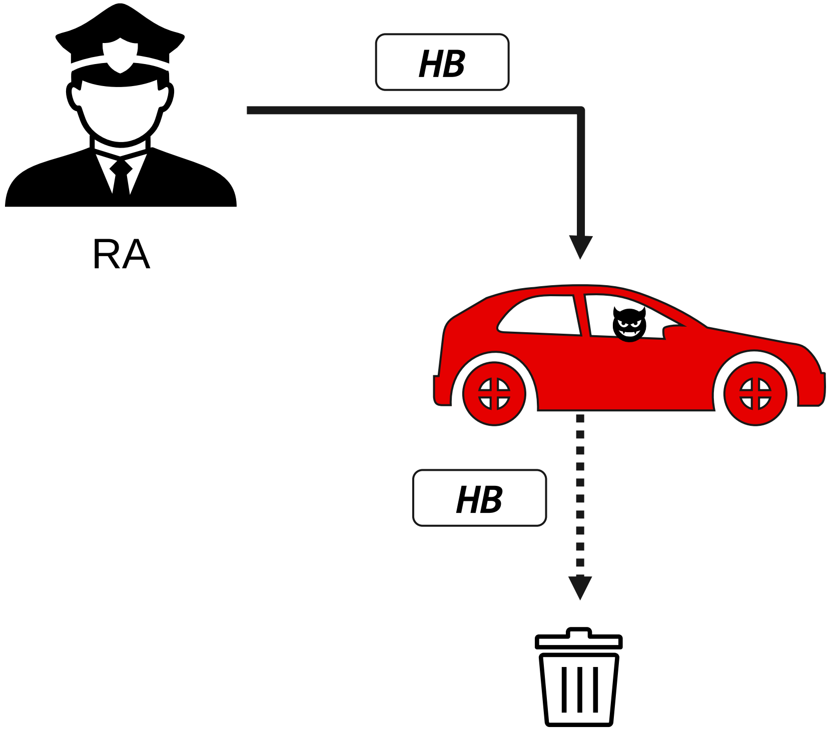
“Cooperative” attacker: HBs are forwarded to the TC and credentials self-revoked



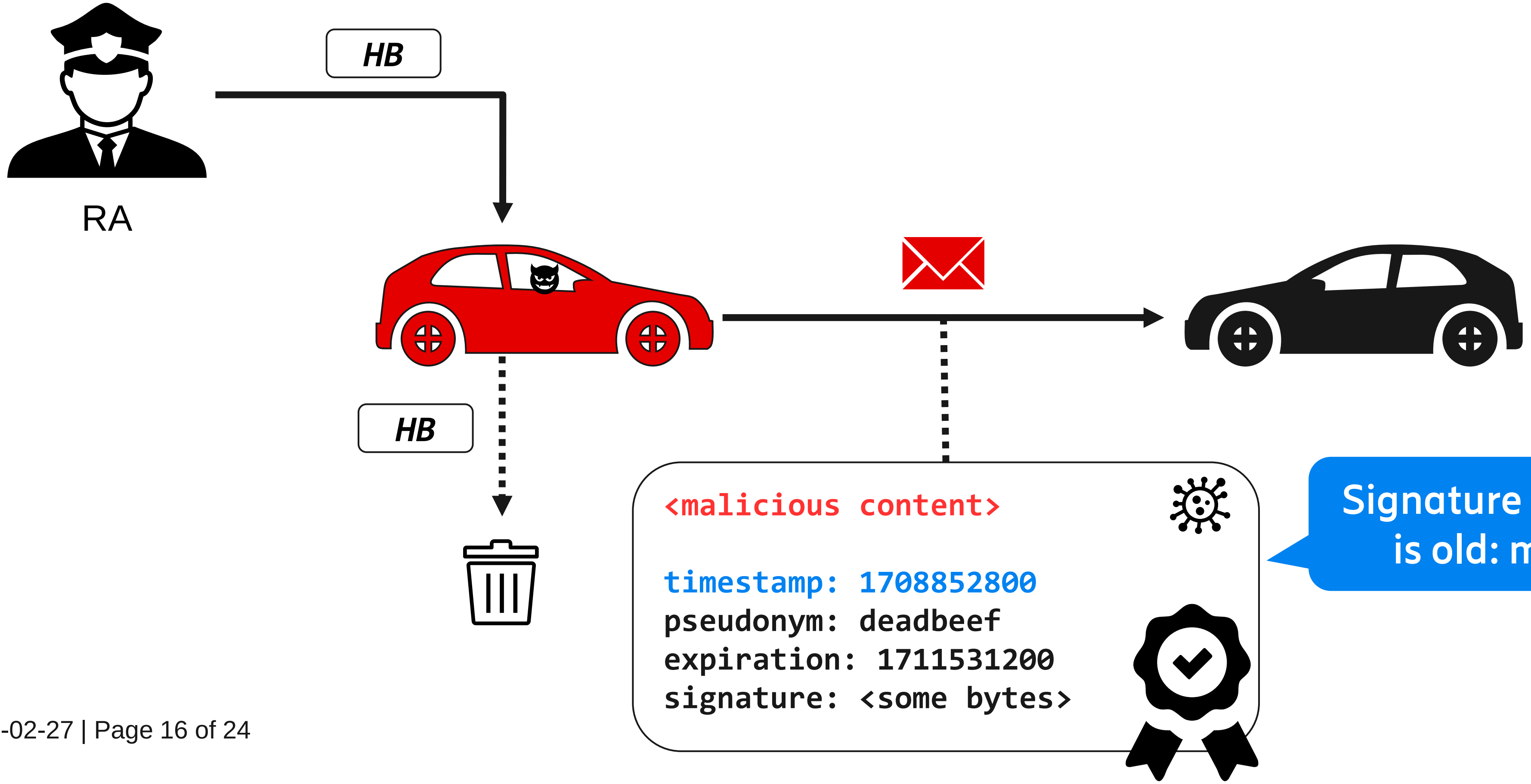
```
<malicious content>

timestamp: 1709025600
pseudonym: null
expiration: null
signature: null
```

Signature is invalid: message is discarded by receiver

“Non-cooperative” attacker: HBs are dropped to elude revocation



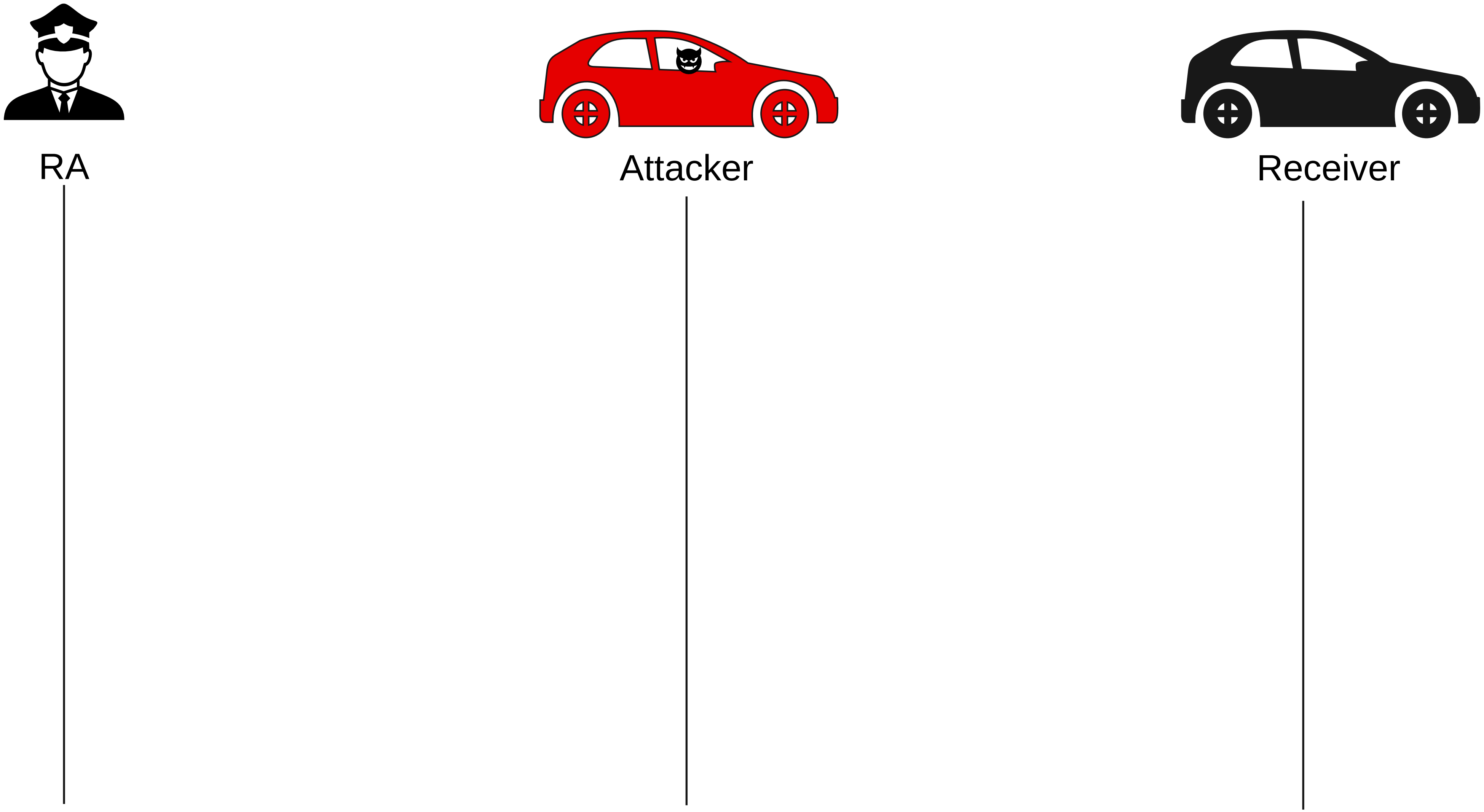
“Non-cooperative” attacker: HBs are dropped to elude revocation



```
<malicious content>

timestamp: 1708852800
pseudonym: deadbeef
expiration: 1711531200
signature: <some bytes>
```

Signature is valid, but timestamp is old: message is discarded

Effective revocation time

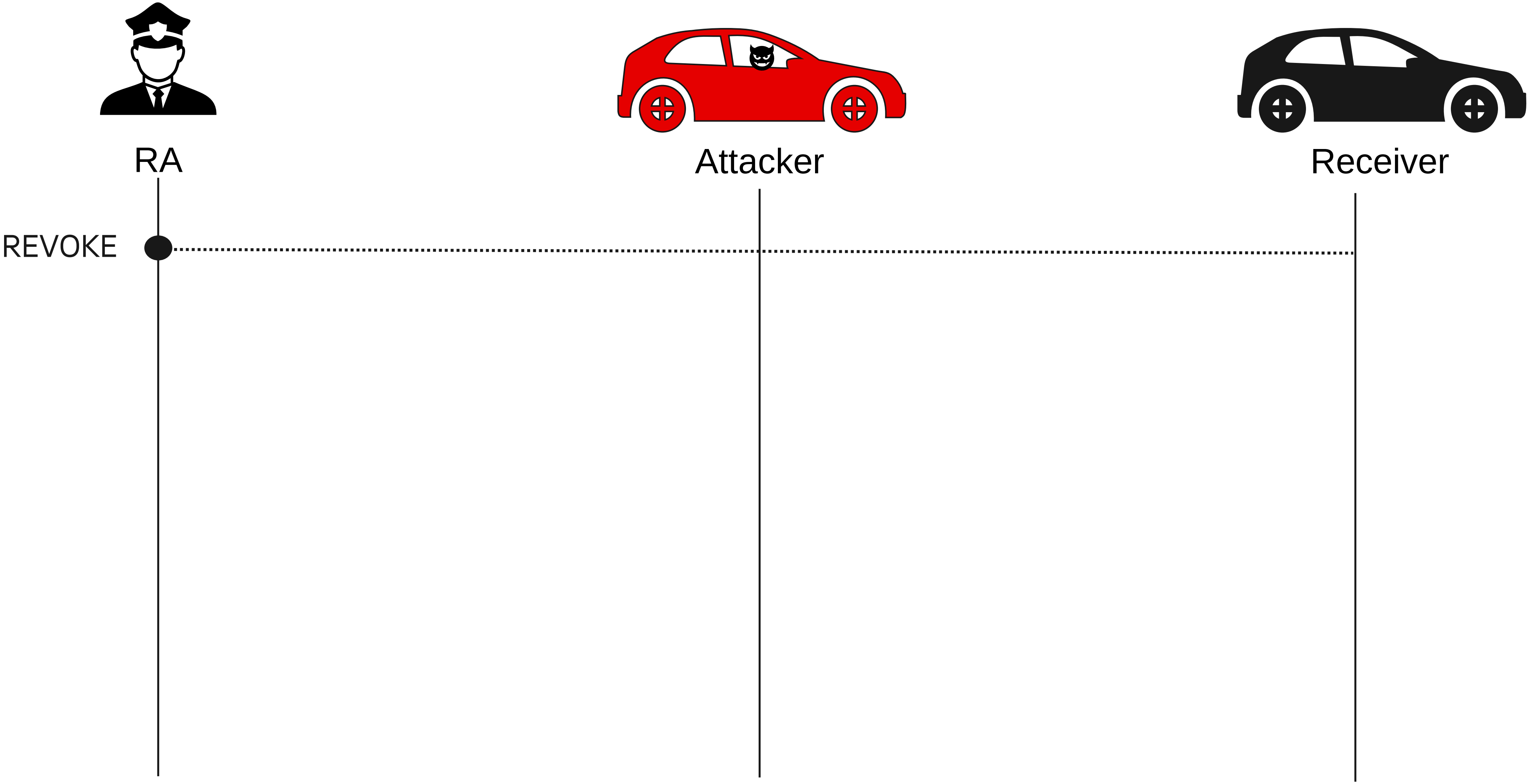


RA

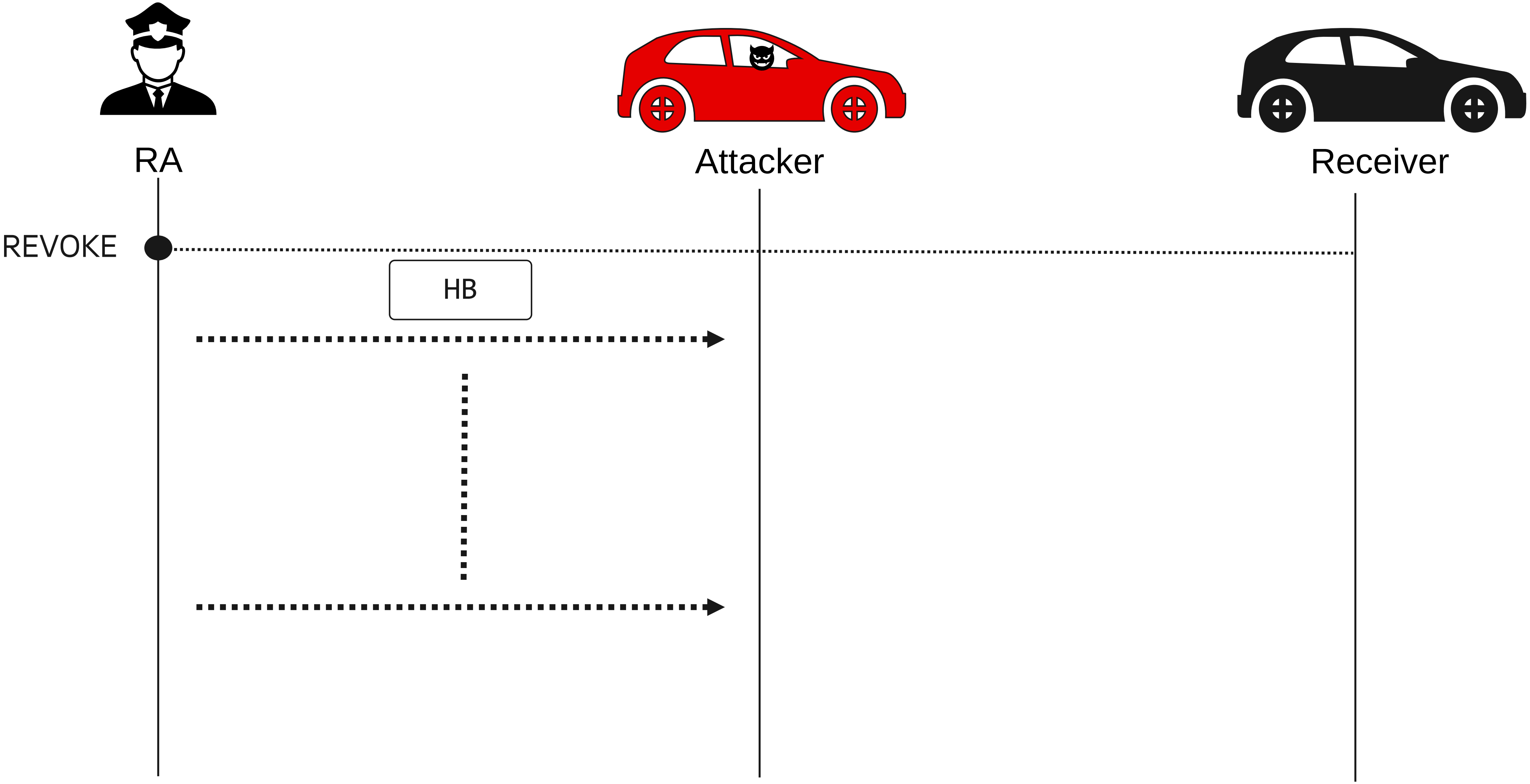
Attacker

Receiver

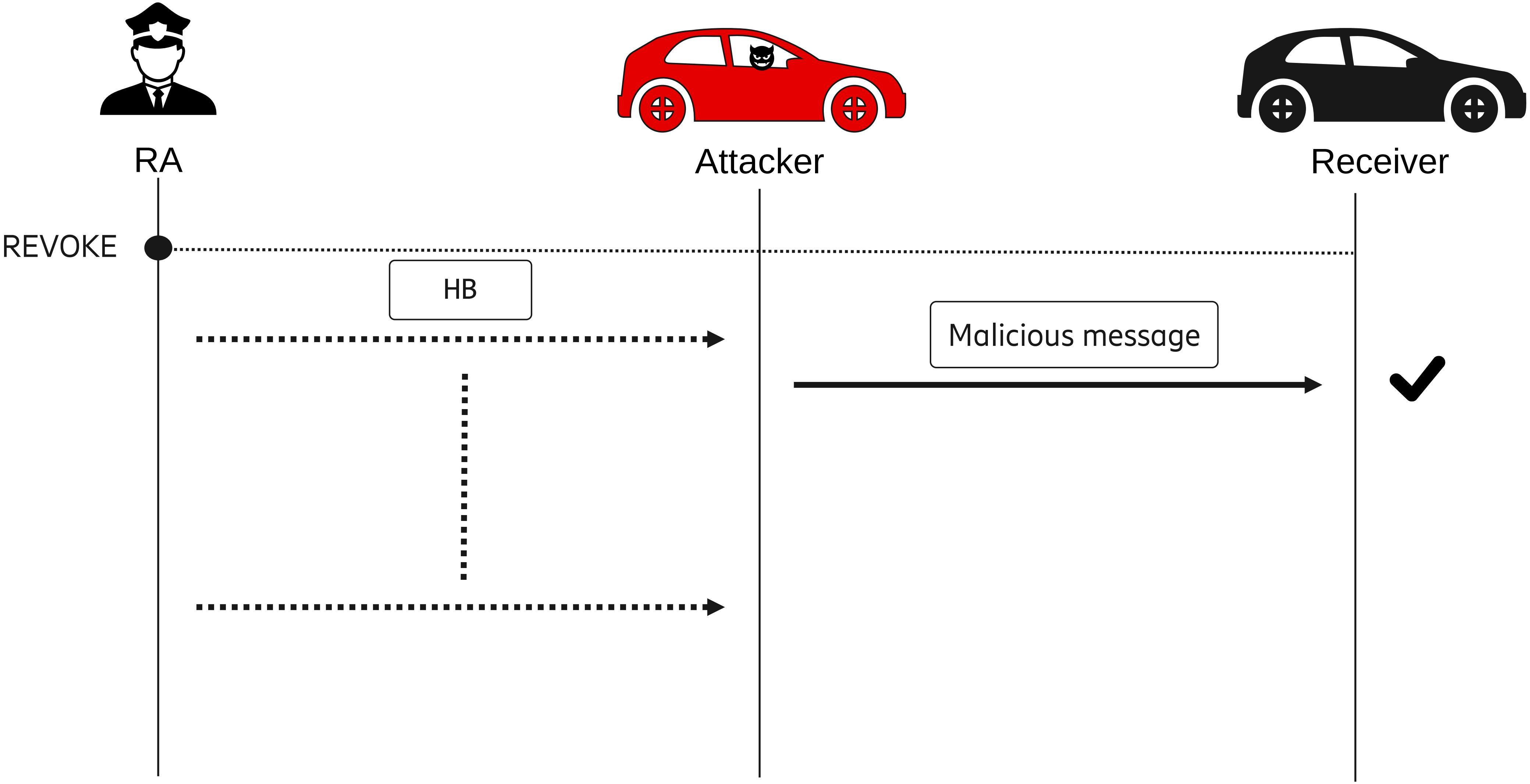
Effective revocation time



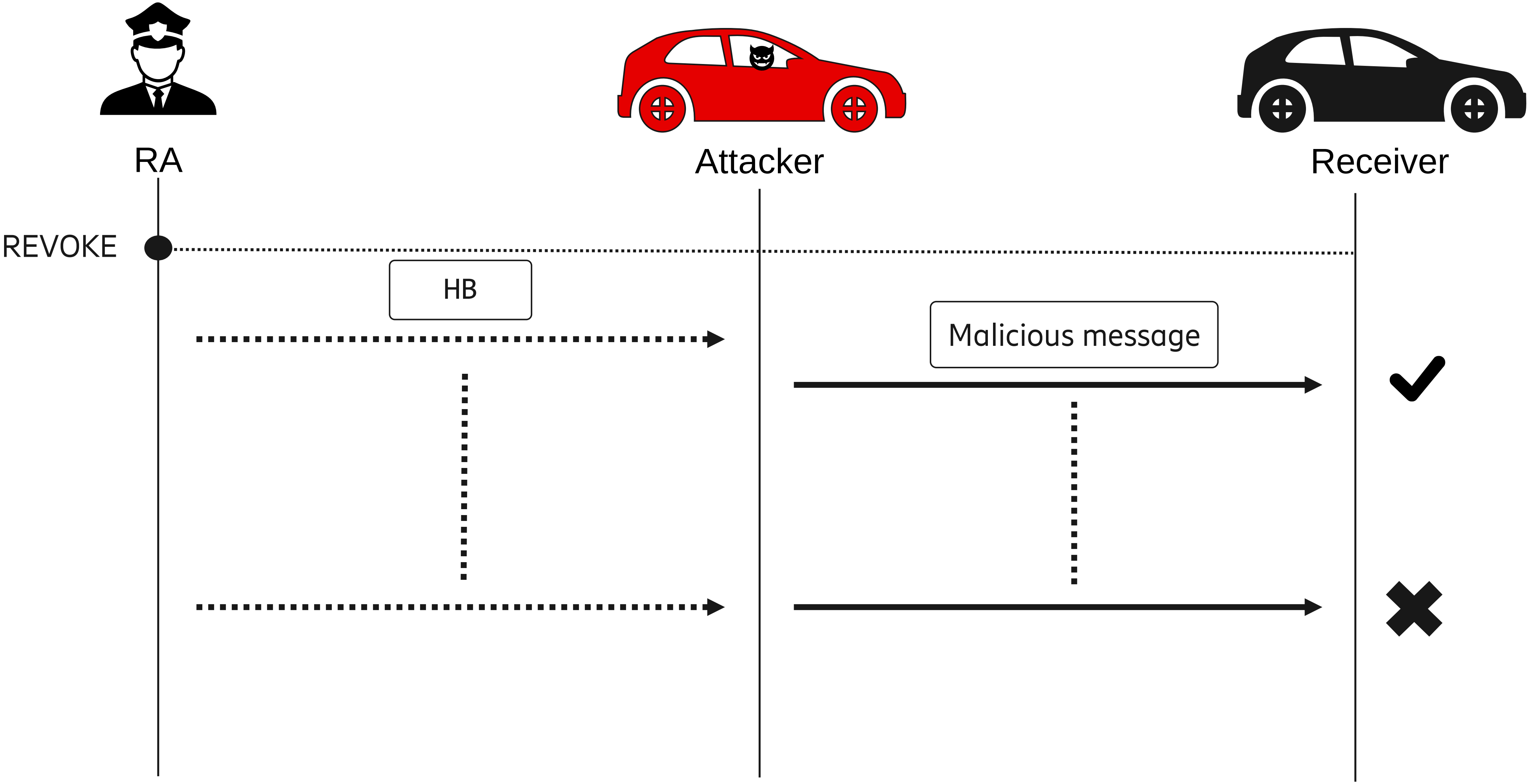
Effective revocation time



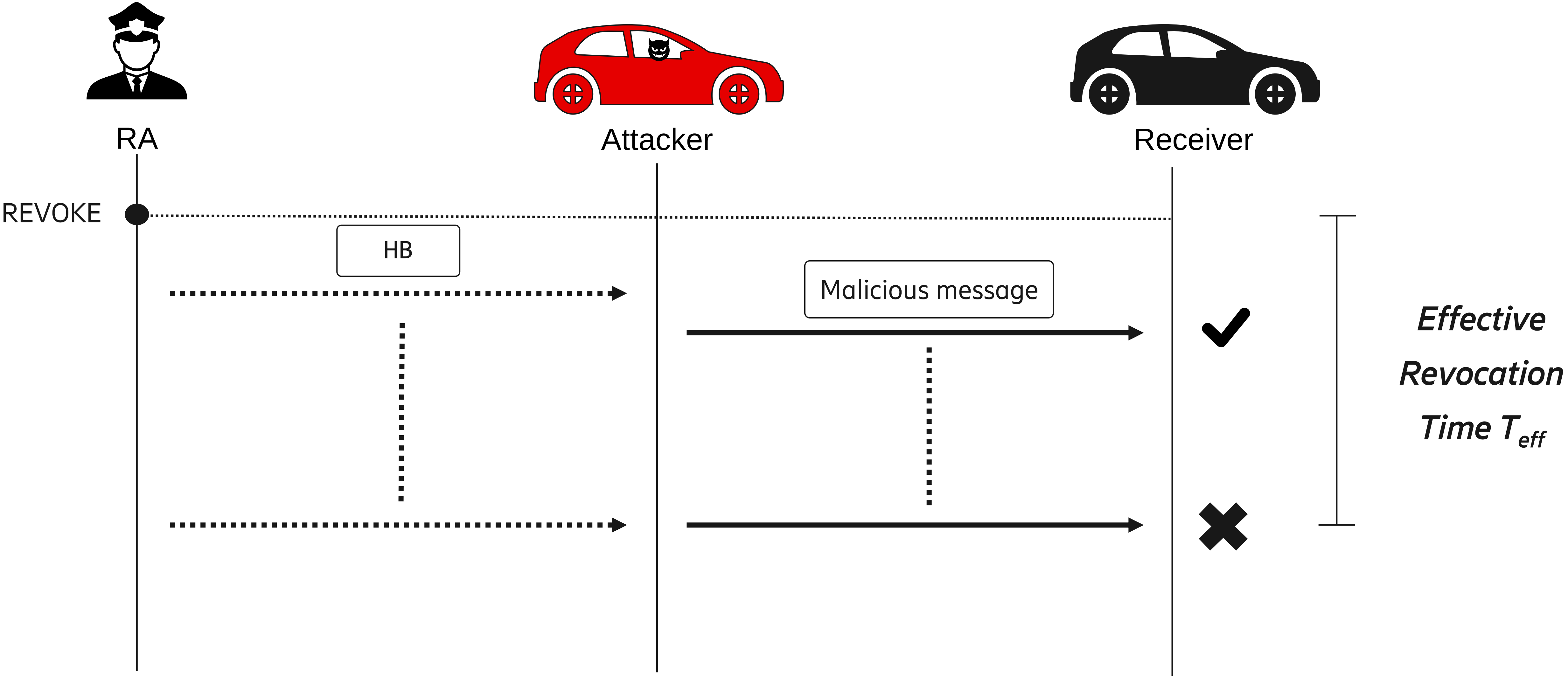
Effective revocation time



Effective revocation time



Effective revocation time



Goal #1: Security

Tamarin Prover. https://tamarin-prover.com

Goal #1: Security

```
lemma effective_revocation [heuristic=o "oracle.py"]:
"
All msg ps t #i . MessageAccepted(msg, ps, t)@i ==>
    Ex tv #j . SystemInitialized(tv)@j & j<i
    & not (
      Ex ps2 t_rev #k . RevocationIssued(ps2, t_rev)@k
        & GreaterThan(t, t_rev + tv)
    )
"
```

*If revocation occurs at time **t**, a receiver will discard all messages from the attacker when its internal time reaches* $\boldsymbol{t + T_v}$

Tamarin Prover. https://tamarin-prover.com

Goal #1: Security

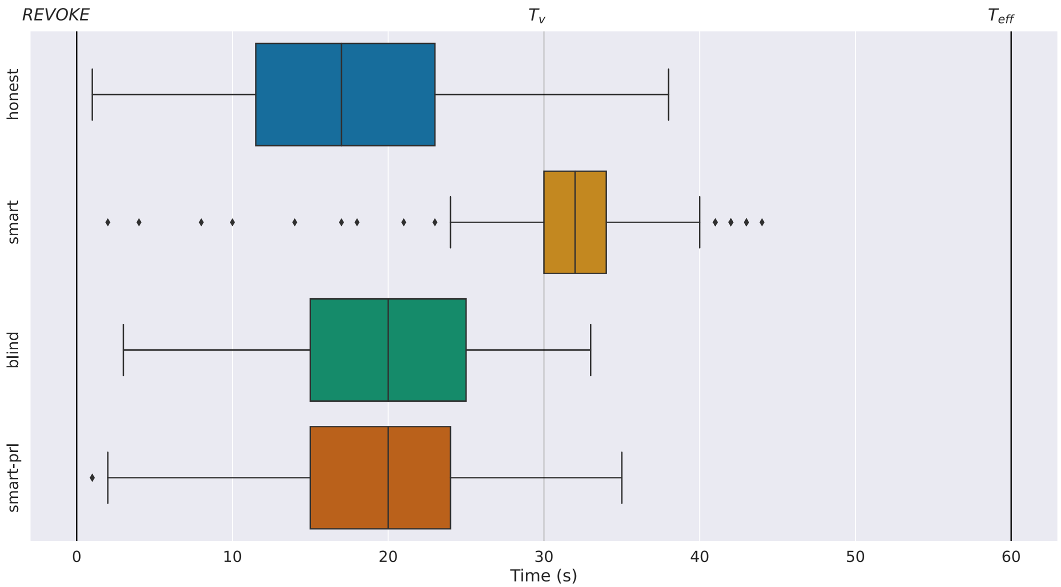
```
lemma effective_revocation [heuristic=o "oracle.py"]:
"
All msg ps t #i . MessageAccepted(msg, ps, t)@i ==>
    Ex tv #j . SystemInitialized(tv)@j & j<i
    & not (
      Ex ps2 t_rev #k . RevocationIssued(ps2, t_rev)@k
        & GreaterThan(t, t_rev + tv)
    )
"
```

*If revocation occurs at time **t**, a receiver will discard all messages from the attacker when its internal time reaches* $\boldsymbol{t + T_v}$

→ Assuming that honest receivers are at most T_V behind the RA time: $T_{eff} = 2T_v$

Tamarin Prover. https://tamarin-prover.com

Goal #2: Usability

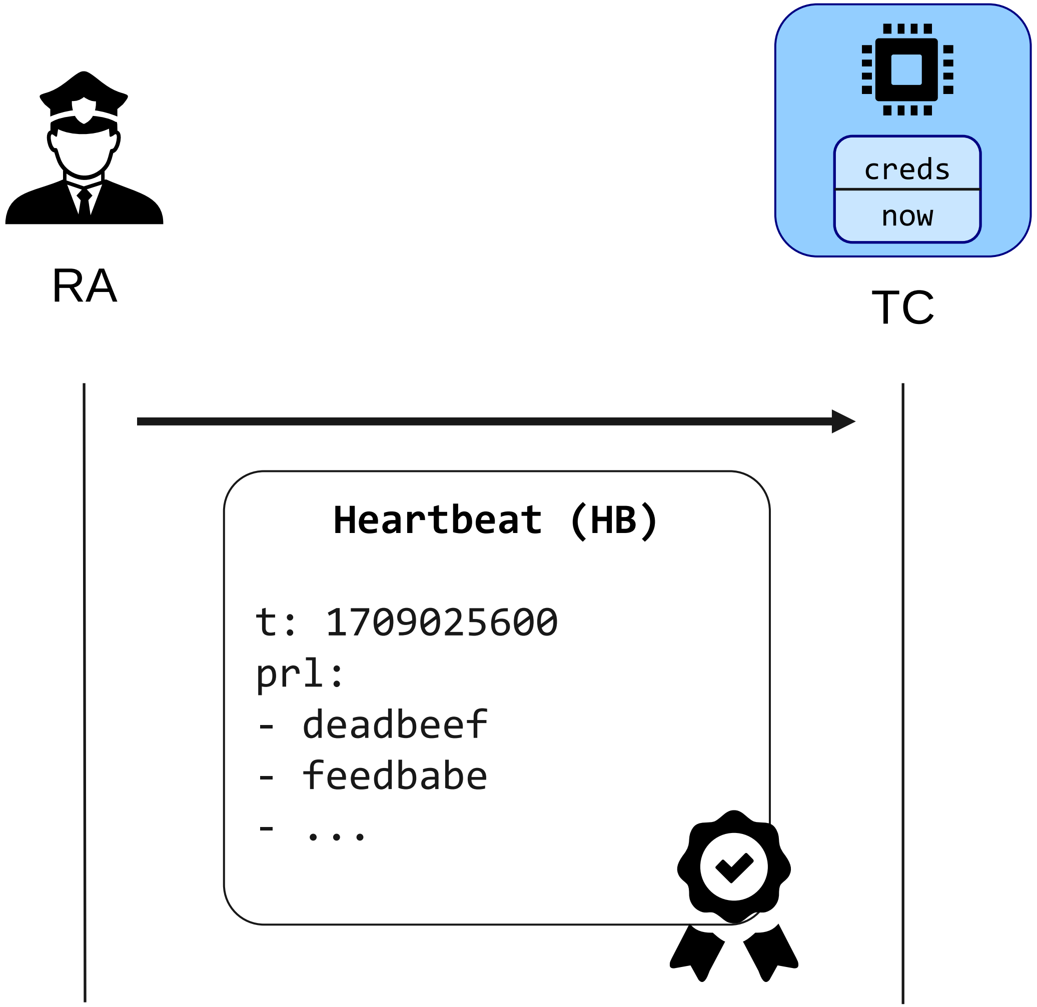


Distribution of revocation times for each class of attacker (lower is better)

- Simulation of a small V2X network in Kubernetes
  - Severe network malfunctions (delays, interruptions)
  - Attackers trying to evade revocation
- Evaluated different scenarios with different parameters → **more info on the paper!**

Kubernetes. https://www.kubernetes.io

What about efficiency?



Heartbeat (HB)

```
t: 1709025600
prl:
- deadbeef
- feedbabe
- ...
```

Signature check

Freshness check

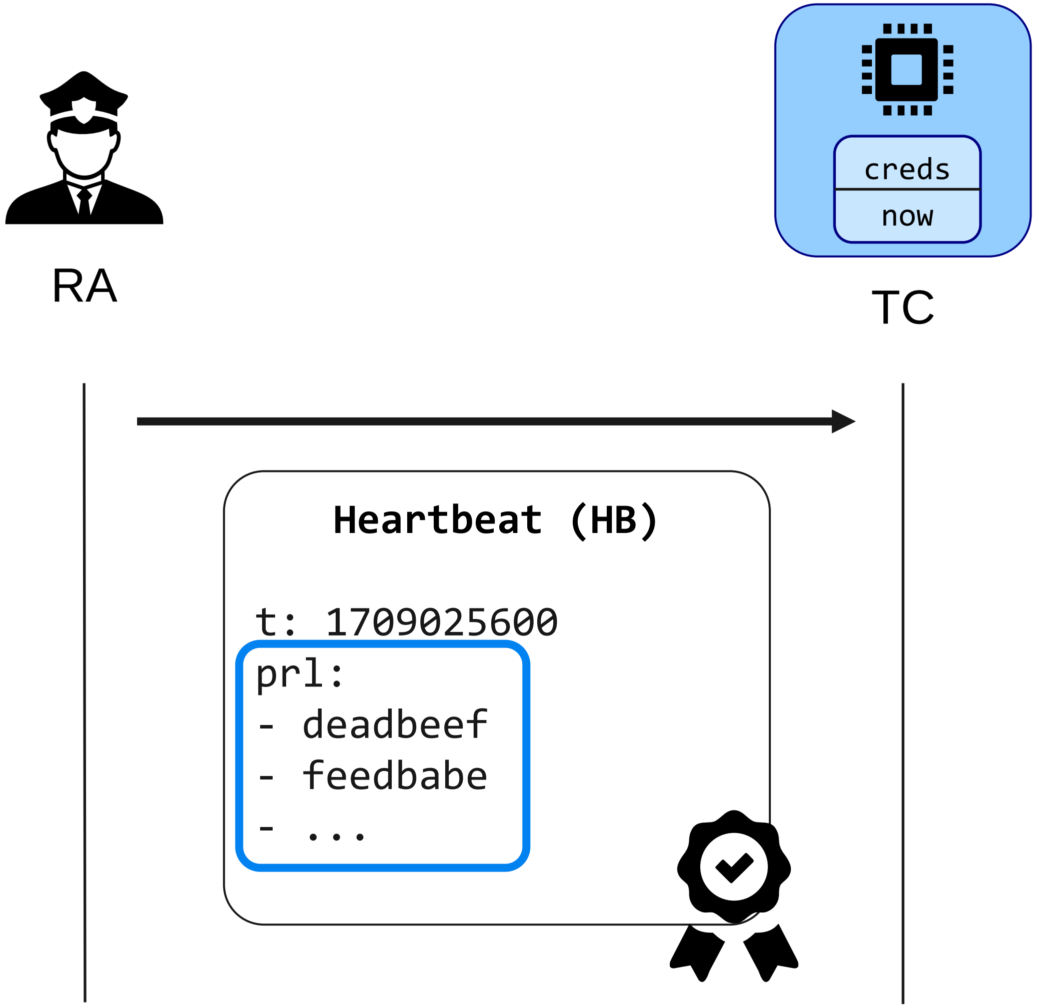
$t >= now - T_v$

Time synchronization

$now = t$

Revocation check

What about efficiency?



Heartbeat (HB)

```
t: 1709025600
prl:
- deadbeef
- feedbabe
- ...
```

Signature check

Freshness check

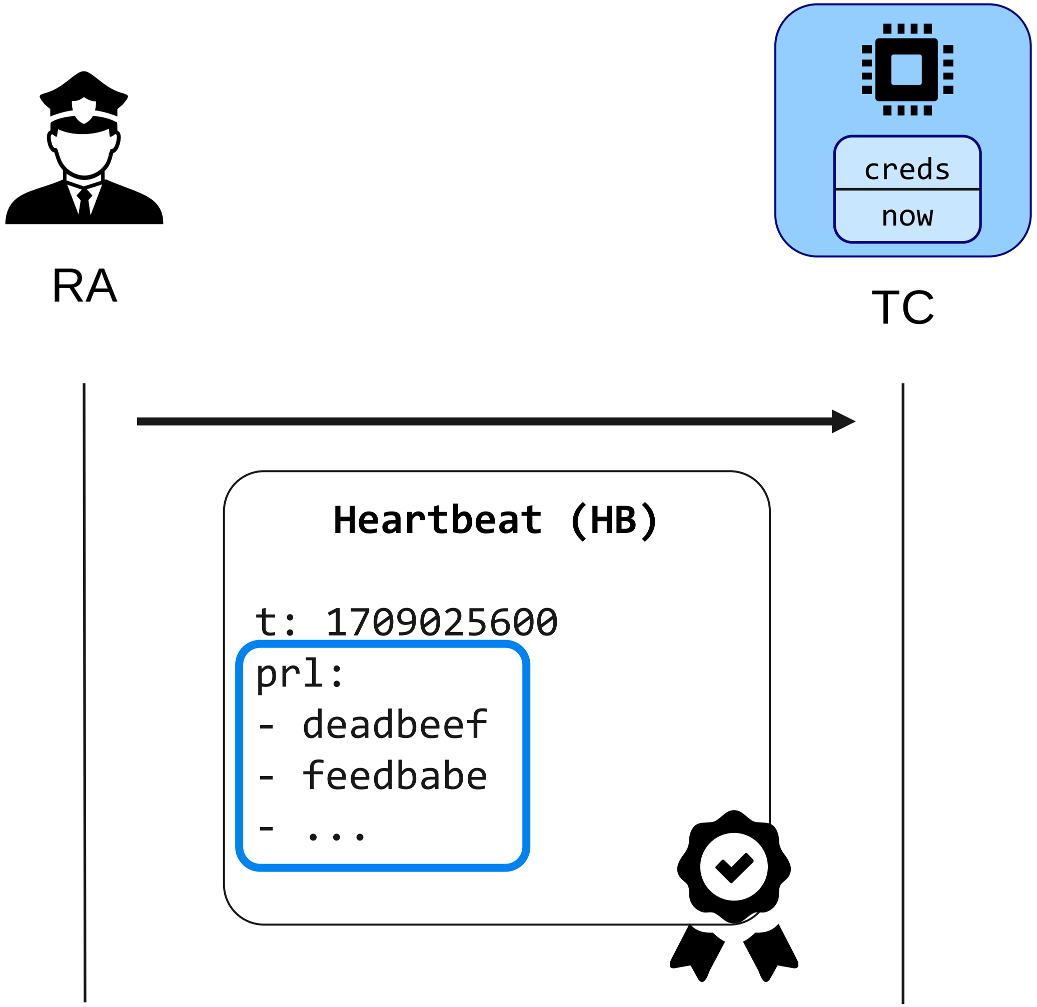
$t >= now - T_v$

Time synchronization

$now = t$

Revocation check

What about efficiency?



Heartbeat (HB)

```
t: 1709025600
prl:
- deadbeef
- feedbabe
- ...
```

Fail? Discard, TC is out of sync
(re-synchronization needed, e.g., via re-enrollment)

Freshness check

$t >= now - T_v$ && $t <= now + T_v$

Time synchronization

now = t

Revocation check

Goal #3: Efficiency

```
lemma no_heartbeats_processed_after_tolerance [heuristic=o
"oracle.py"]:
"
All prl t_hb t #i . HeartbeatProcessed(<prl, t_hb>, t)@i ==>
    Ex tv #j . SystemInitialized(tv)@j & j<i
    & not (
      Ex ps t_rev #k . RevocationIssued(ps, t_rev)@k
        & k<i
        & GreaterThan(t_hb, t_rev + tv)
    )
"
```

If revocation occurs at time ***t****, the attacker will not be able to process any HBs containing timestamp >=* ***t*** **+** T_v

Goal #3: Efficiency

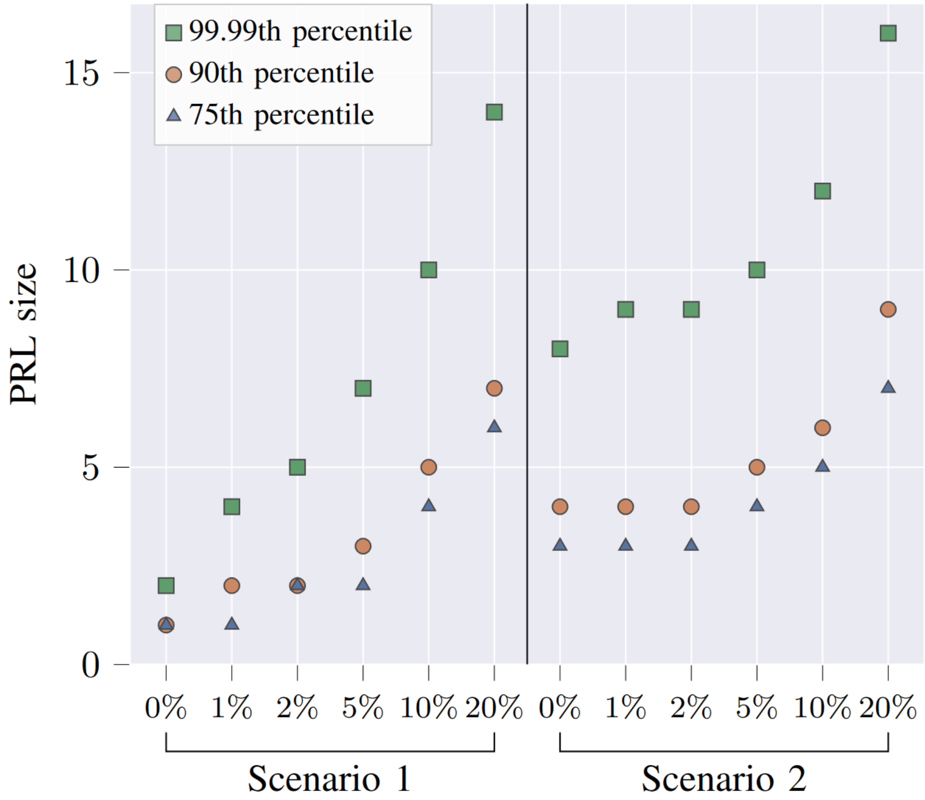
```
lemma no_heartbeats_processed_after_tolerance [heuristic=o
"oracle.py"]:
"
All prl t_hb t #i . HeartbeatProcessed(<prl, t_hb>, t)@i ==>
    Ex tv #j . SystemInitialized(tv)@j & j<i
    & not (
      Ex ps t_rev #k . RevocationIssued(ps, t_rev)@k
        & k<i
        & GreaterThan(t_hb, t_rev + tv)
    )
"
```

*If revocation occurs at time **t**, the attacker will not be able to process any HBs containing timestamp >=* $\boldsymbol{t} + T_v$

→ Each revoked credential can be safely removed from the HB after $T_{prl} = T_v$ since insertion

Goal #3: Efficiency

Lower is better



% of attackers in the network

(n = 800, T_{prl} = 30)

- PRL as a Markov Model
  - Adding elements with probability *p*
  - Removing elements with probability *$1/T_{prl}$*

- Evaluated different scenarios with different parameters → **more info on the paper!**

Limitations

Limitations

- TC is needed in vehicles
  - Requires changes in V2X standards

Limitations

- TC is needed in vehicles
  - Requires changes in V2X standards

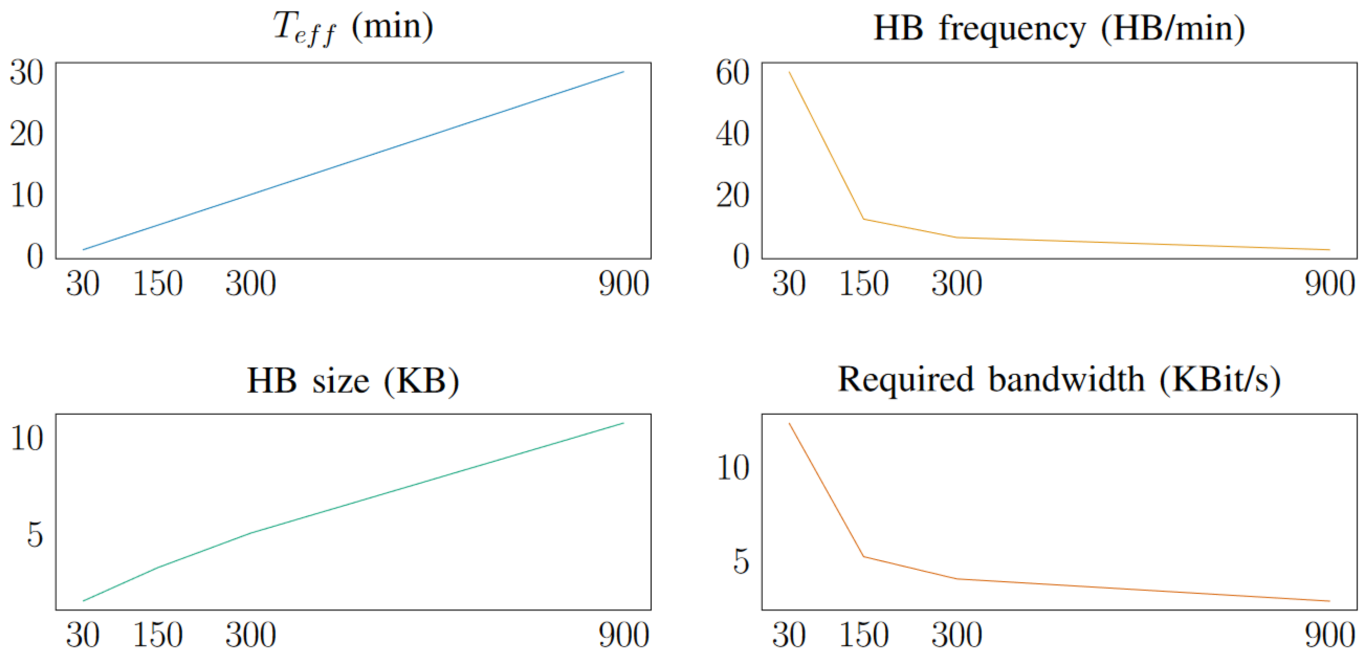
- Vehicles need continuous connectivity to the infrastructure
  - Offline periods up to T_V are tolerated
  - For longer periods, the TC needs to re-authenticate to the infrastructure

Limitations

- TC is needed in vehicles
  - Requires changes in V2X standards

- Vehicles need continuous connectivity to the infrastructure
  - Offline periods up to T_V are tolerated
  - For longer periods, the TC needs to re-authenticate to the infrastructure

Lower is better



X axis: values of T_V in seconds

Efficient and Timely Revocation of V2X Credentials

Artifact Evaluated
NDSS
Available
Functional
Reproduced

- A formally verified revocation scheme based on trusted computing and self-revocation
- Guaranteed upper bound on revocation time (*“effective revocation”*)
- Tolerance parameter T_V gives a trade-off between security, usability and efficiency
- Open-source!*

Paper & Artifacts

Gianluca Scopelliti, Christoph Baumann, Fritz Alder, Eddy Truyen, Jan Tobias Mühlberg.

Network and Distributed System Security (NDSS) Symposium 2024. San Diego, CA.

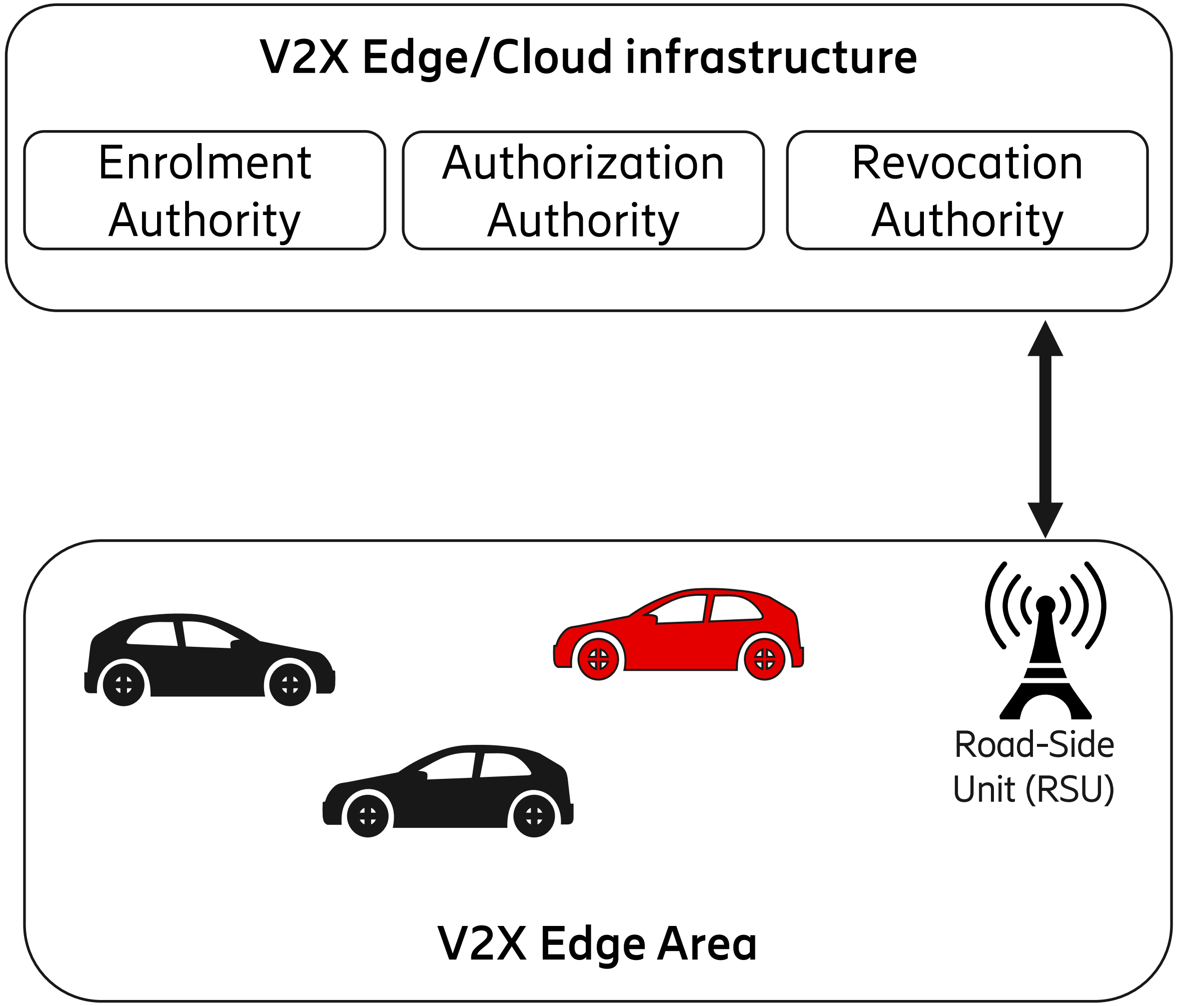
gianluca.scopelliti@ericsson.com

*github.com/EricssonResearch/v2x-self-revocation

ERICSSON   KU LEUVEN   DistriNet   ULB   CYBERSECURITY RESEARCH CENTER

Backup

System and attacker model

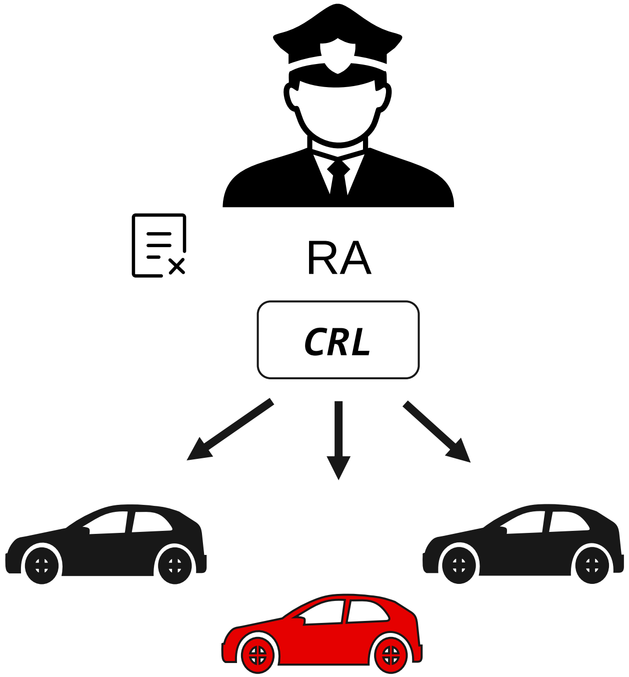


- **Attacker model:**
  - V2X Edge/Cloud infrastructure: **trusted**
  - Vehicles: **potentially malicious**

- **Attacker's goal:**
  - Obtain V2X credentials / compromise vehicle
  - Spread malicious information

State of the art in revocation schemes

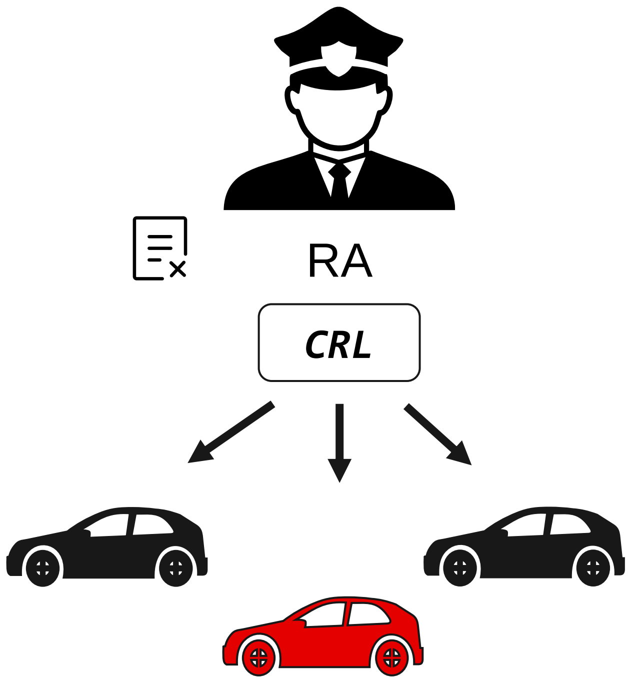
Active revocation (IEEE 1609.2.1 – SCMS [1])



[1] IEEE Std 1609.2.1-2022 "IEEE WAVE - Certificate Management Interfaces for End Entities“

State of the art in revocation schemes

Active revocation (IEEE 1609.2.1 – SCMS [1])

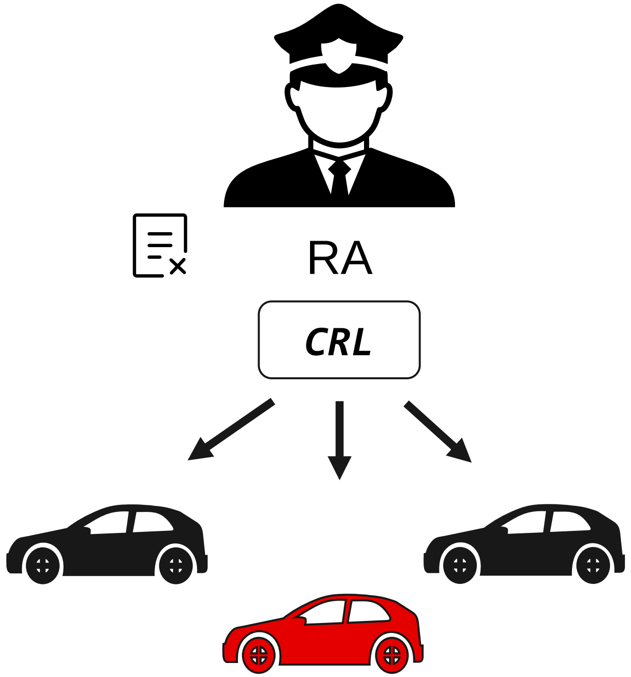


- **Relatively fast response**: revocation is achieved as soon as CRL update is received
  - Delays? Network interruptions?

[1] IEEE Std 1609.2.1-2022 "IEEE WAVE - Certificate Management Interfaces for End Entities"

State of the art in revocation schemes

Active revocation (IEEE 1609.2.1 – SCMS [1])

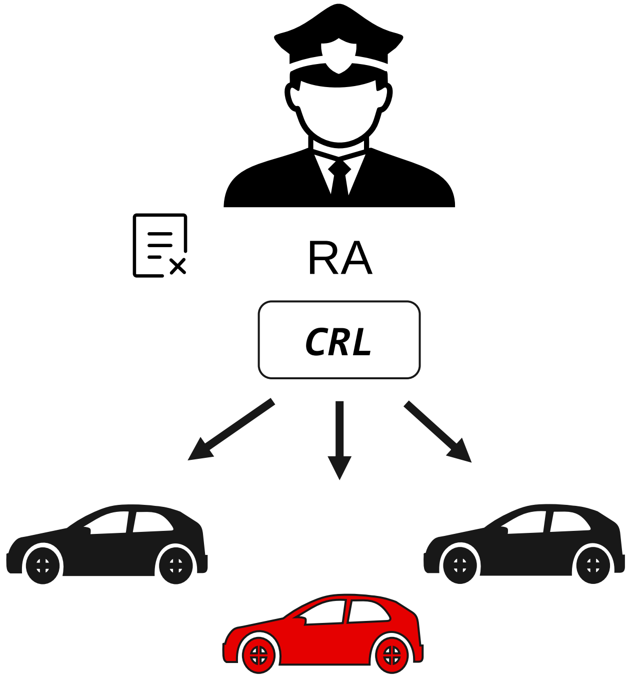


- **Relatively fast response**: revocation is achieved as soon as CRL update is received
  - Delays? Network interruptions?
- **High latency:** each received message requires checking the pseudonym against the CRL

[1] IEEE Std 1609.2.1-2022 "IEEE WAVE - Certificate Management Interfaces for End Entities"

State of the art in revocation schemes

Active revocation (IEEE 1609.2.1 – SCMS [1])

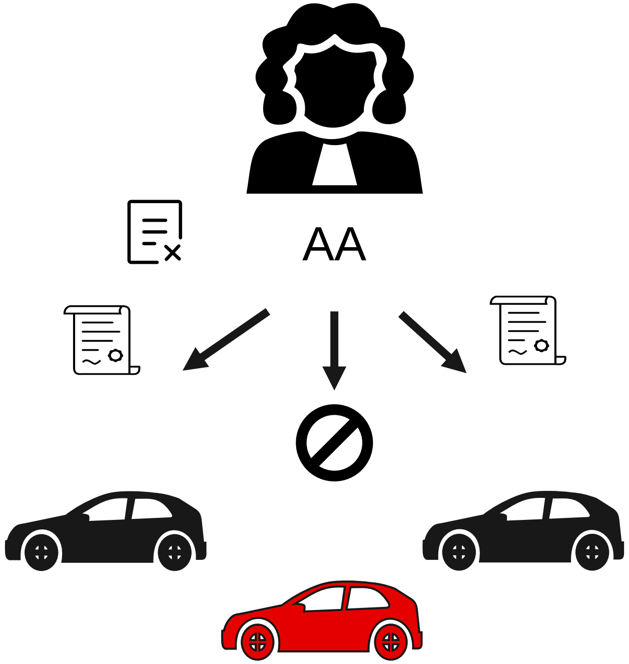


- **Relatively fast response**: revocation is achieved as soon as CRL update is received
  - Delays? Network interruptions?
- **High latency:** each received message requires checking the pseudonym against the CRL
- **Not scalable**: CRLs grow bigger and bigger over time

[1] IEEE Std 1609.2.1-2022 "IEEE WAVE - Certificate Management Interfaces for End Entities"

State of the art in revocation schemes

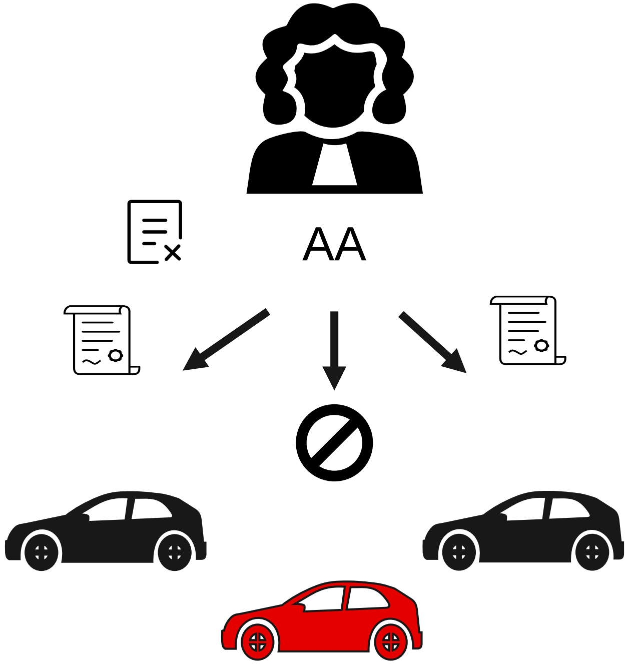
Passive revocation (ETSI TS 102 941 [2])



[2] ETSI TS 102 940 version 2.1.1, "Intelligent Transport Systems (ITS); Security, ITS communications security architecture and security management"

State of the art in revocation schemes

Passive revocation (ETSI TS 102 941 [2])

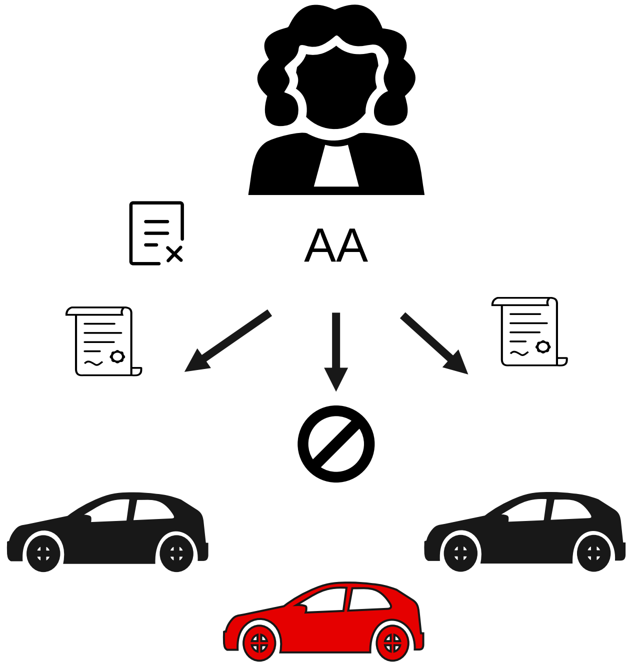


- **Slow response**: revocation is achieved when all the attacker’s pseudonyms have expired

[2] ETSI TS 102 940 version 2.1.1, “Intelligent Transport Systems (ITS); Security, ITS communications security architecture and security management”

State of the art in revocation schemes

Passive revocation (ETSI TS 102 941 [2])

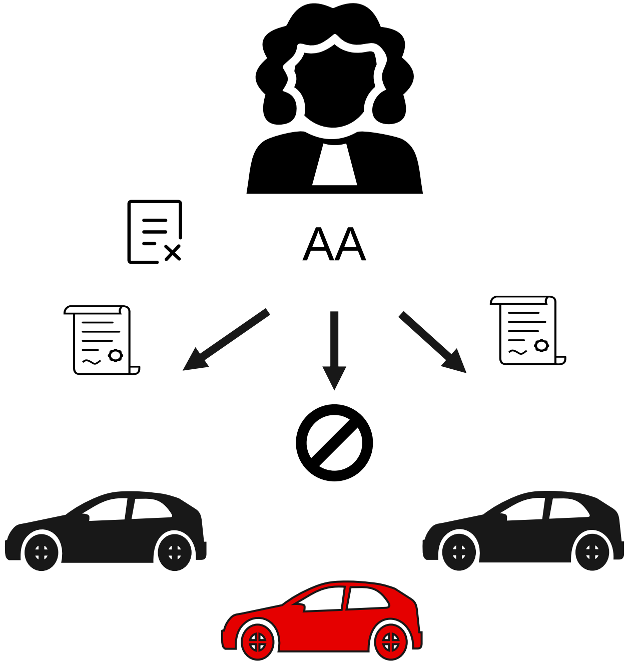


- **Slow response**: revocation is achieved when all the attacker’s pseudonyms have expired
- **Low latency:** no additional verification checks are required

[2] ETSI TS 102 940 version 2.1.1, “Intelligent Transport Systems (ITS); Security, ITS communications security architecture and security management”

State of the art in revocation schemes

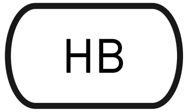
Passive revocation (ETSI TS 102 941 [2])



- **Slow response**: revocation is achieved when all the attacker’s pseudonyms have expired
- **Low latency:** no additional verification checks are required
- **Not scalable**: Increased traffic and computational resources due to frequent pseudonym change

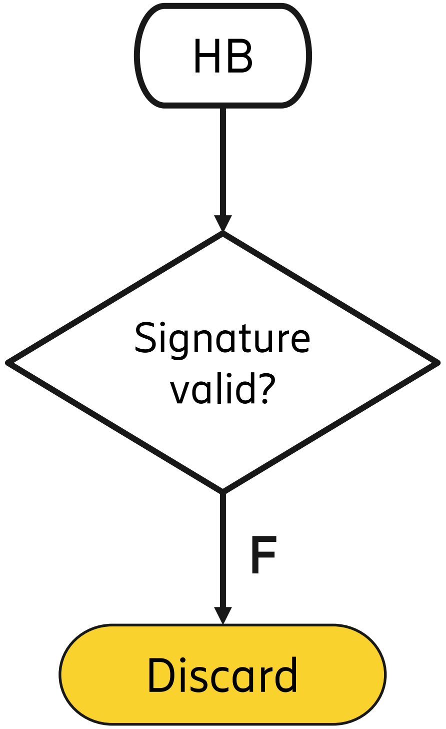
[2] ETSI TS 102 940 version 2.1.1, “Intelligent Transport Systems (ITS); Security, ITS communications security architecture and security management”

Processing a HB: flowchart

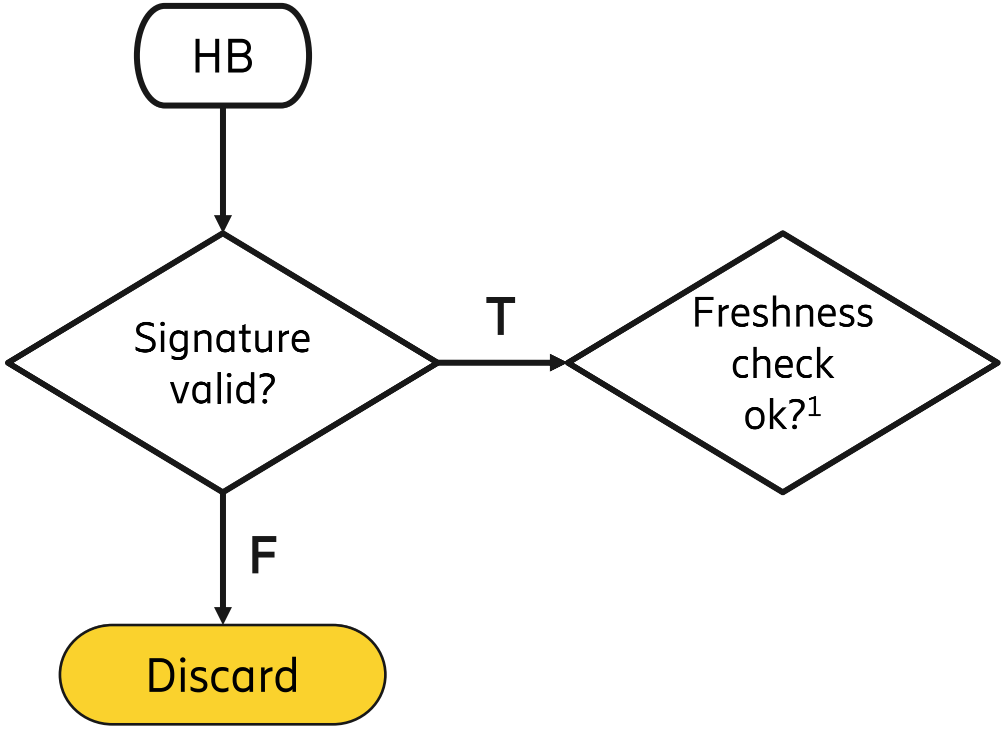


HB

Processing a HB: flowchart

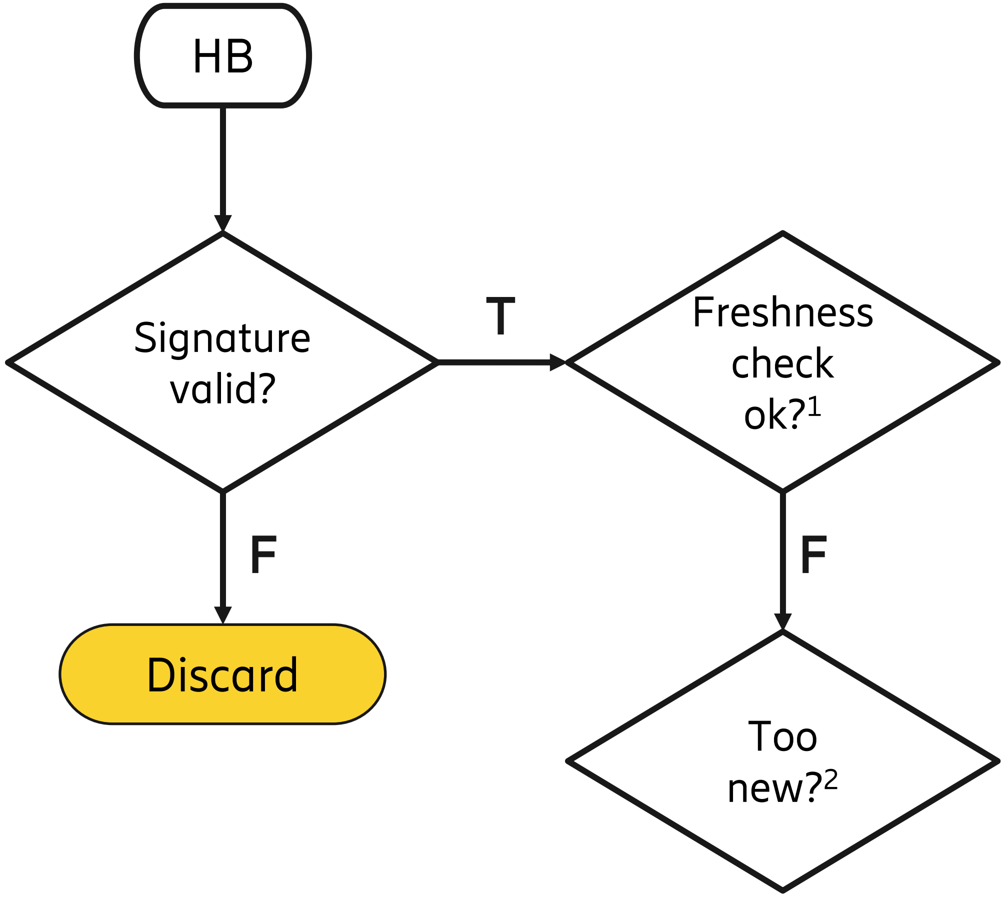


Processing a HB: flowchart



[1] $t - T_V <= t_{HB} <= t + T_V$

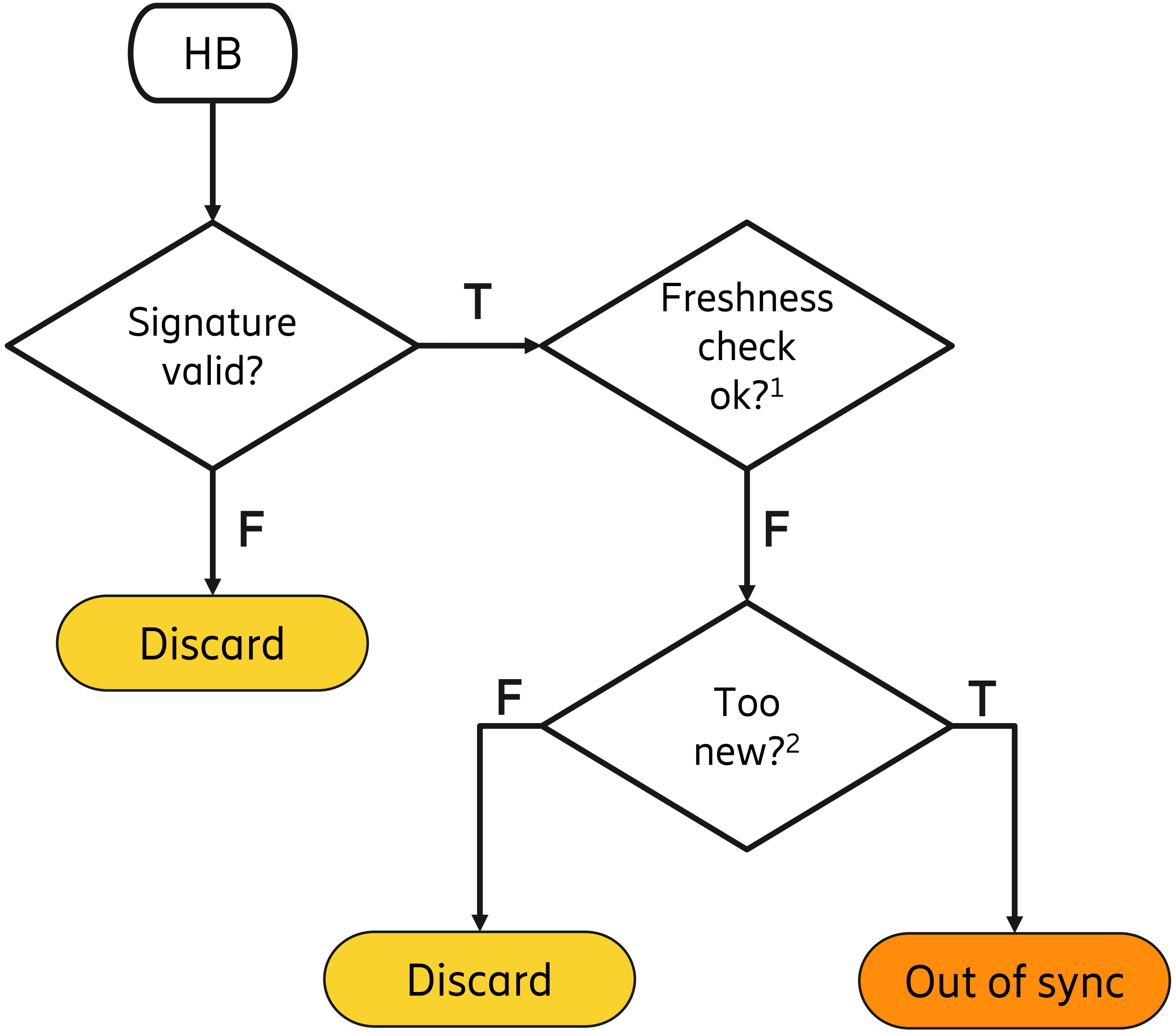
Processing a HB: flowchart



1) $t - T_V <= t_{HB} <= t + T_V$

2) $t_{HB} > t + T_V$

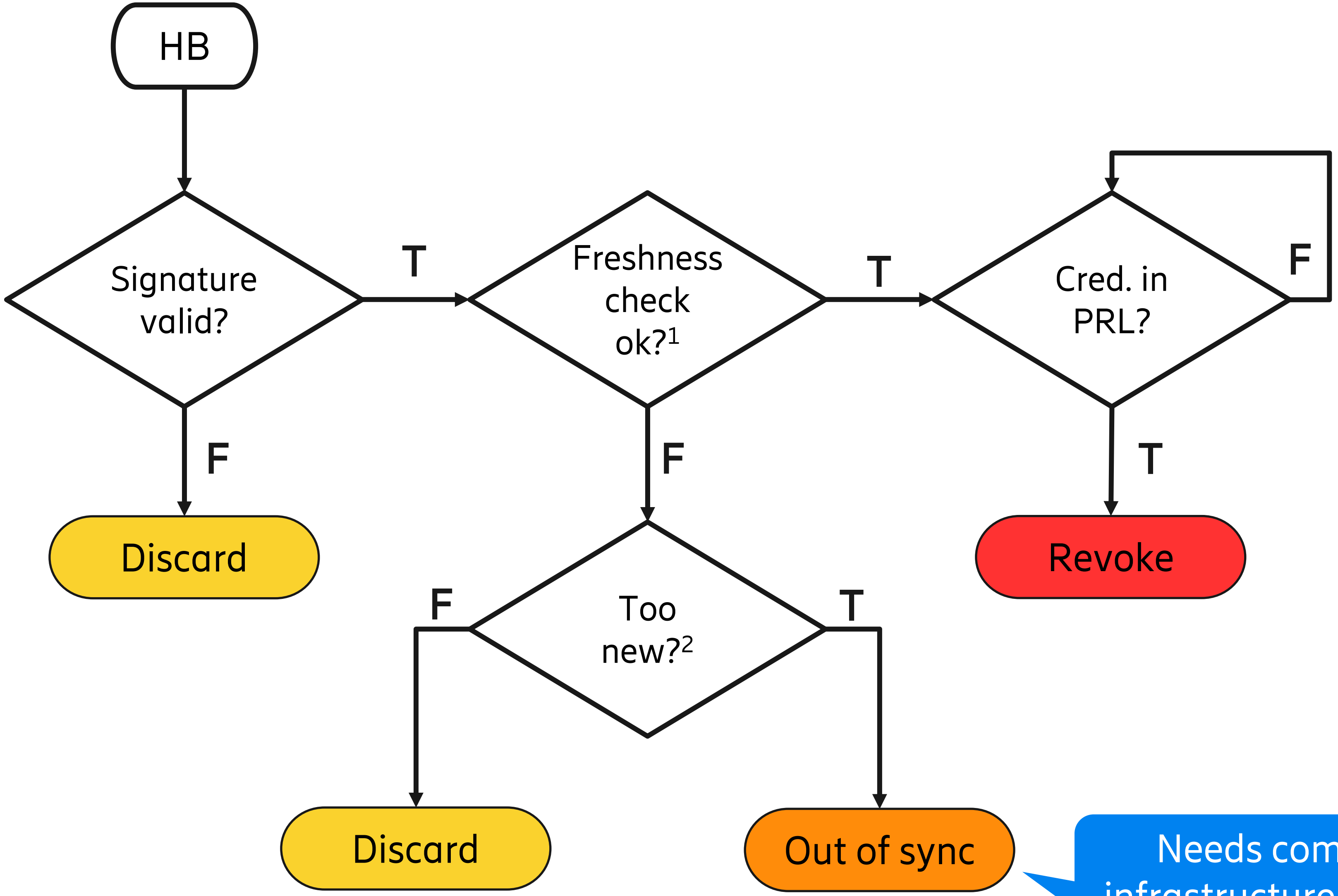
Processing a HB: flowchart



Needs communication with infrastructure (e.g., re-enrolment)

1) $t - T_V <= t_{HB} <= t + T_V$
2) $t_{HB} > t + T_V$

Processing a HB: flowchart

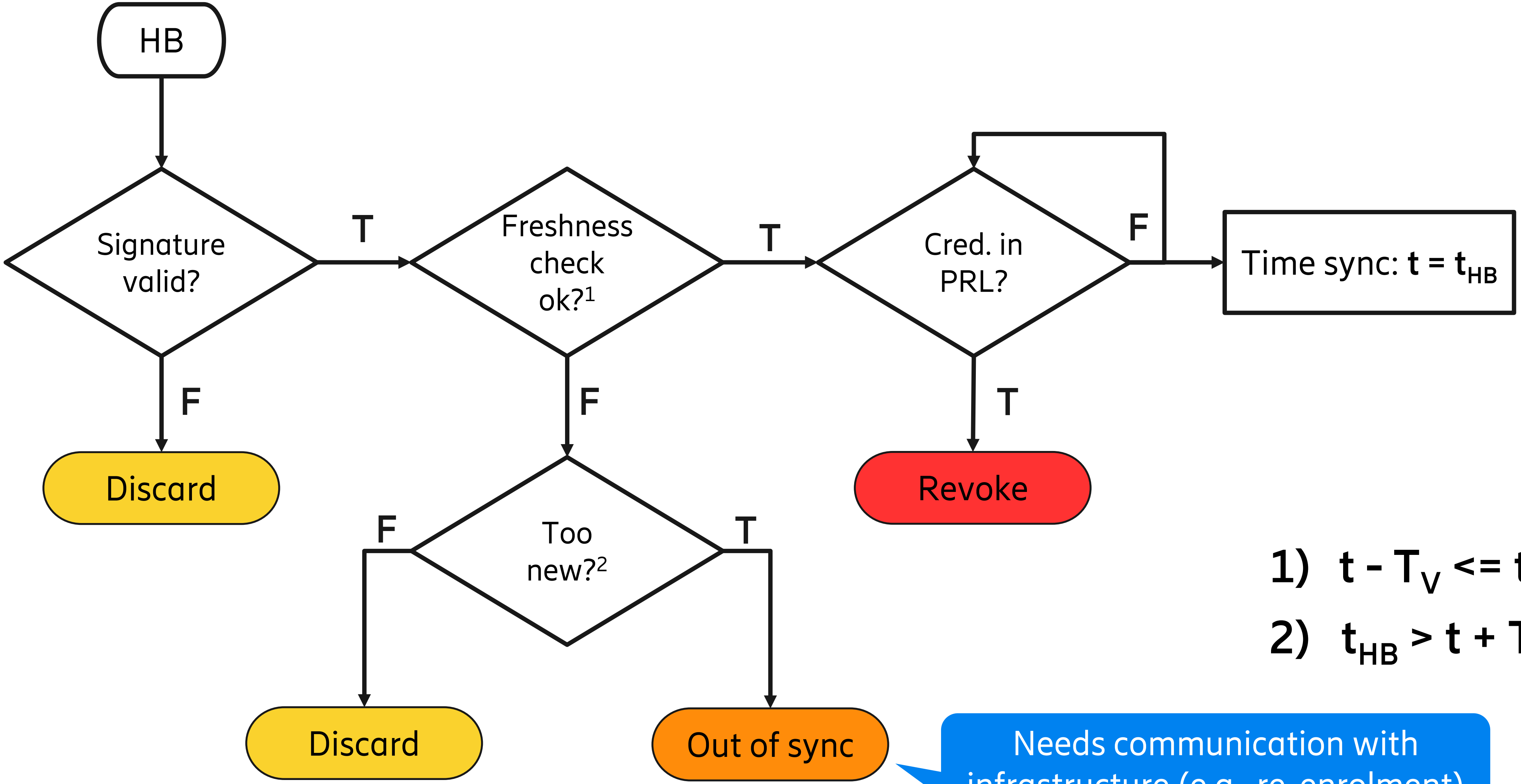


Needs communication with infrastructure (e.g., re-enrolment)

1) $t - T_V <= t_{HB} <= t + T_V$

2) $t_{HB} > t + T_V$

Processing a HB: flowchart

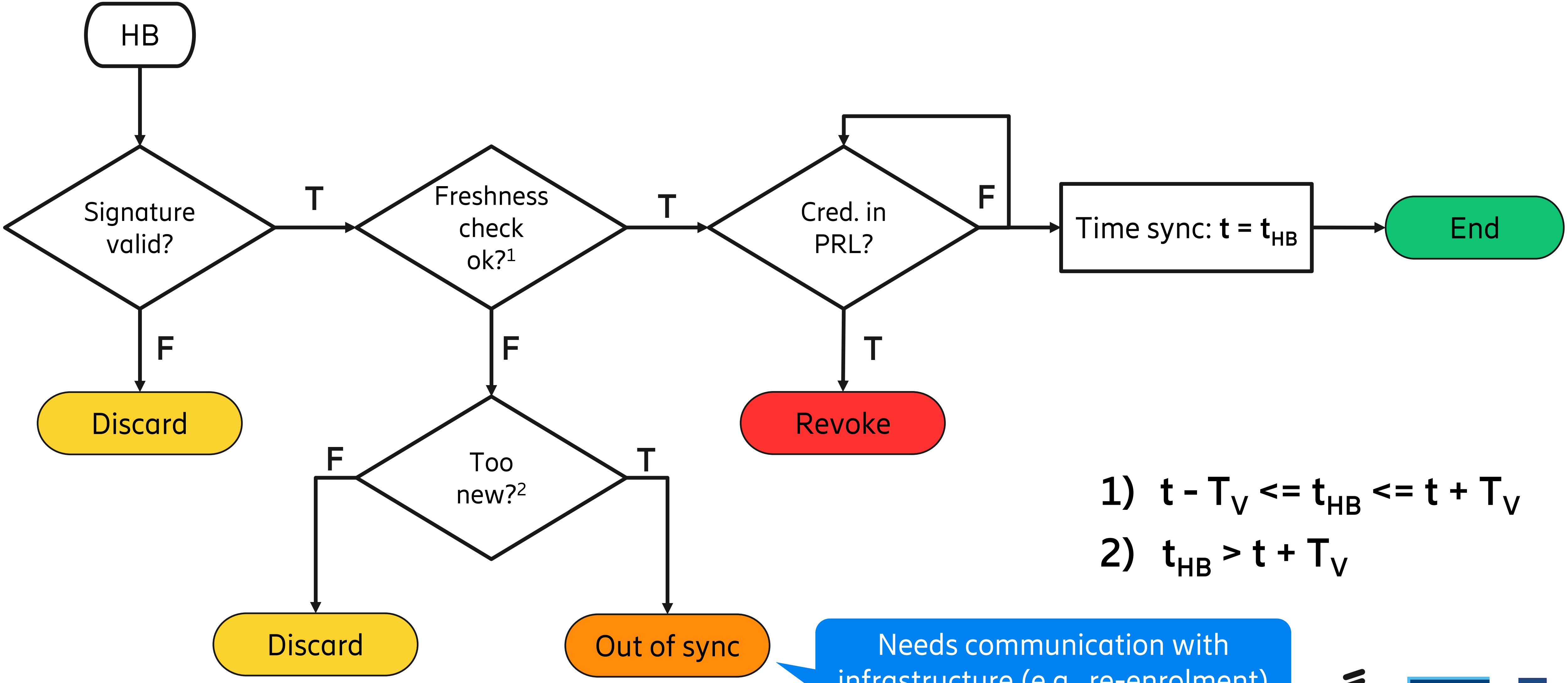


1) $t - T_V <= t_{HB} <= t + T_V$

2) $t_{HB} > t + T_V$

Needs communication with infrastructure (e.g., re-enrolment)

Processing a HB: flowchart

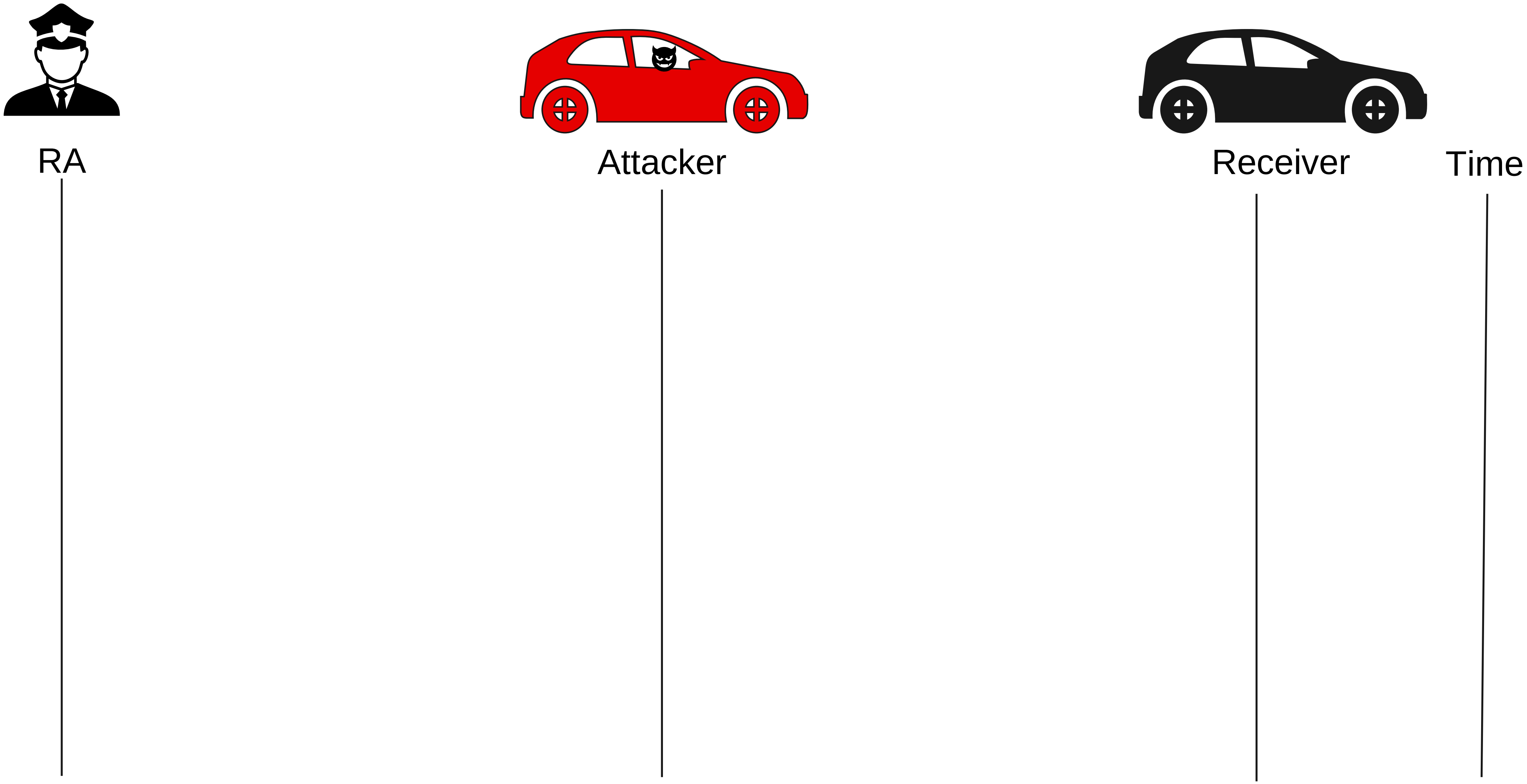


1) $t - T_V <= t_{HB} <= t + T_V$

2) $t_{HB} > t + T_V$

Needs communication with infrastructure (e.g., re-enrolment)

Worst-case T_{eff}



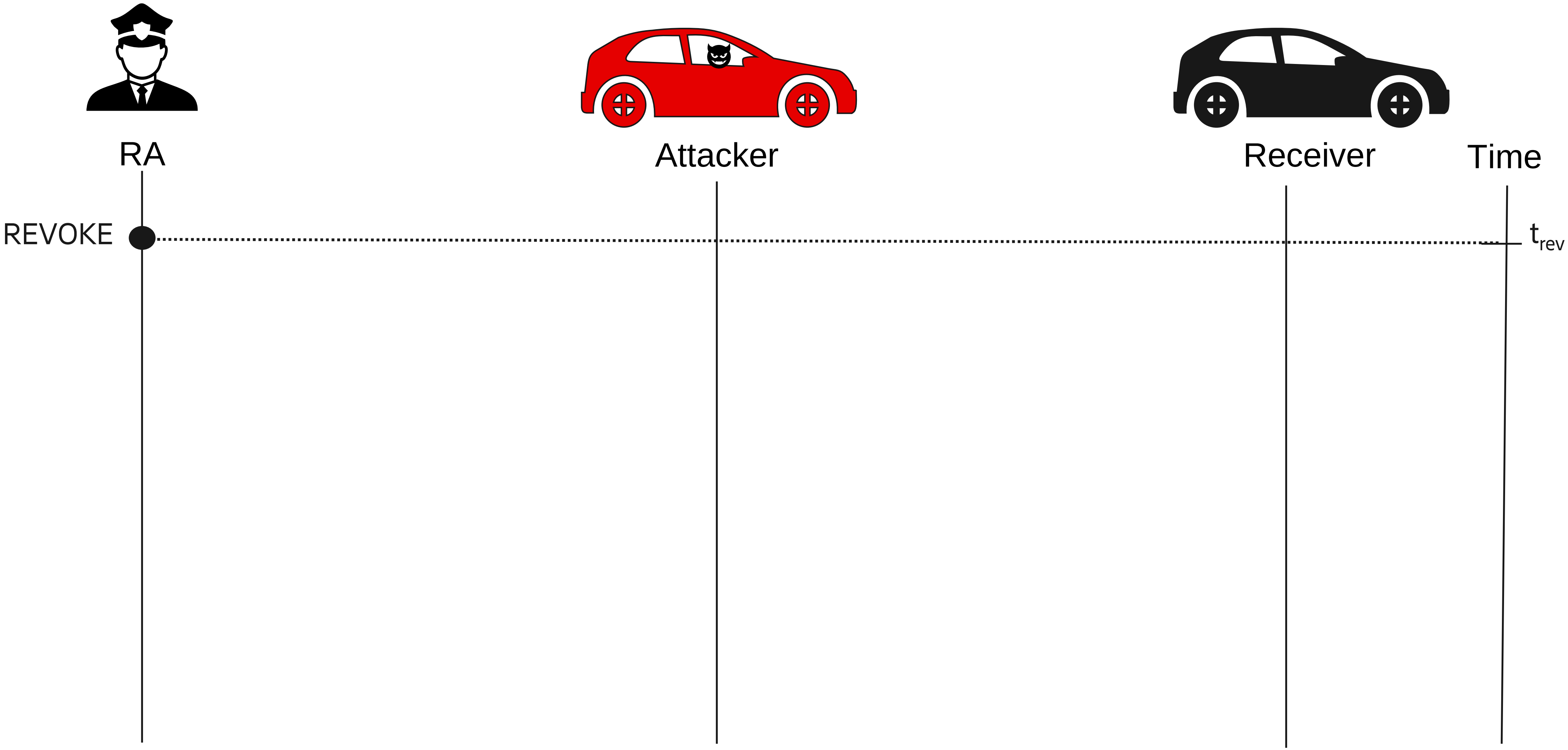
RA

Attacker

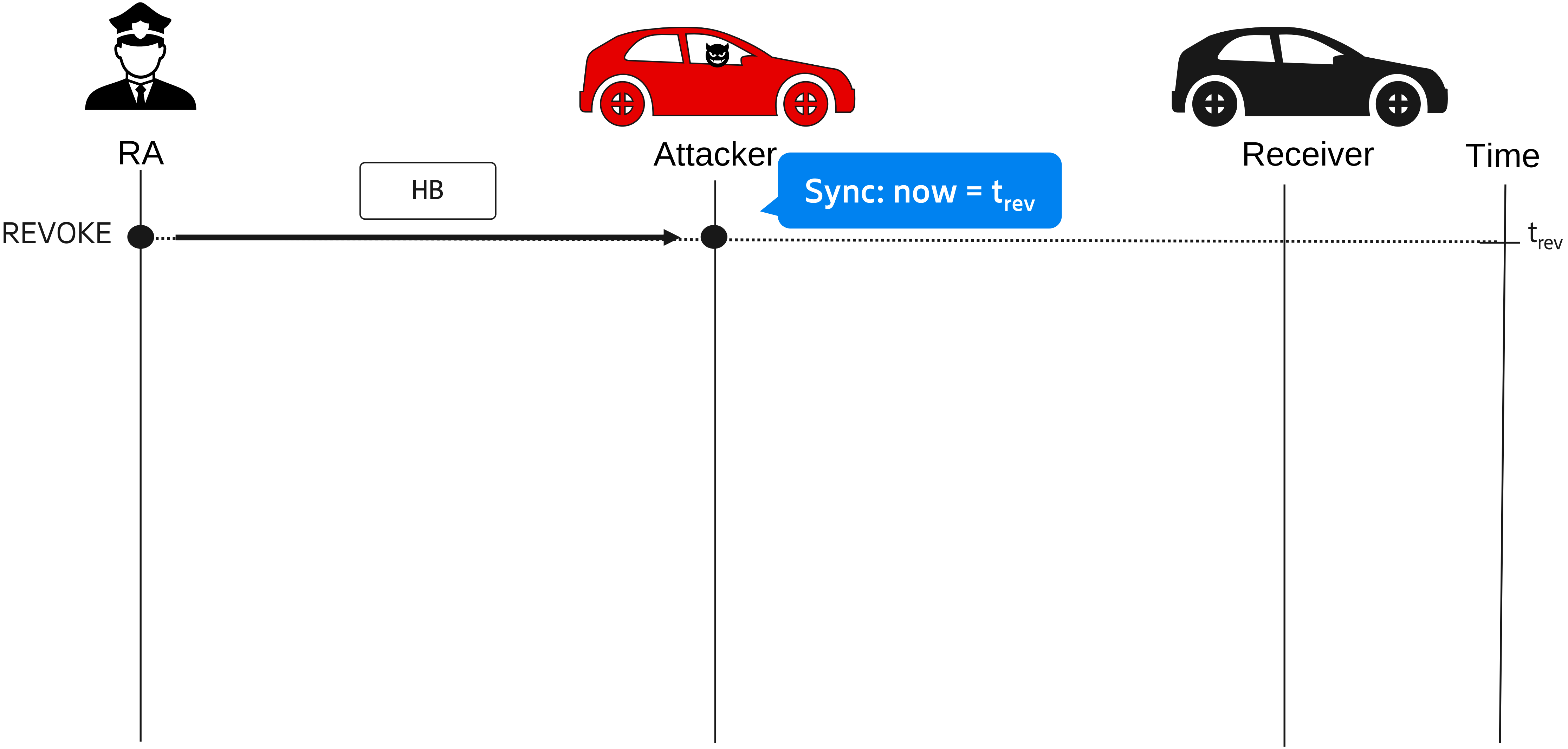
Receiver

Time

Worst-case T_{eff}



Worst-case T_{eff}



RA

Attacker

Receiver

Time

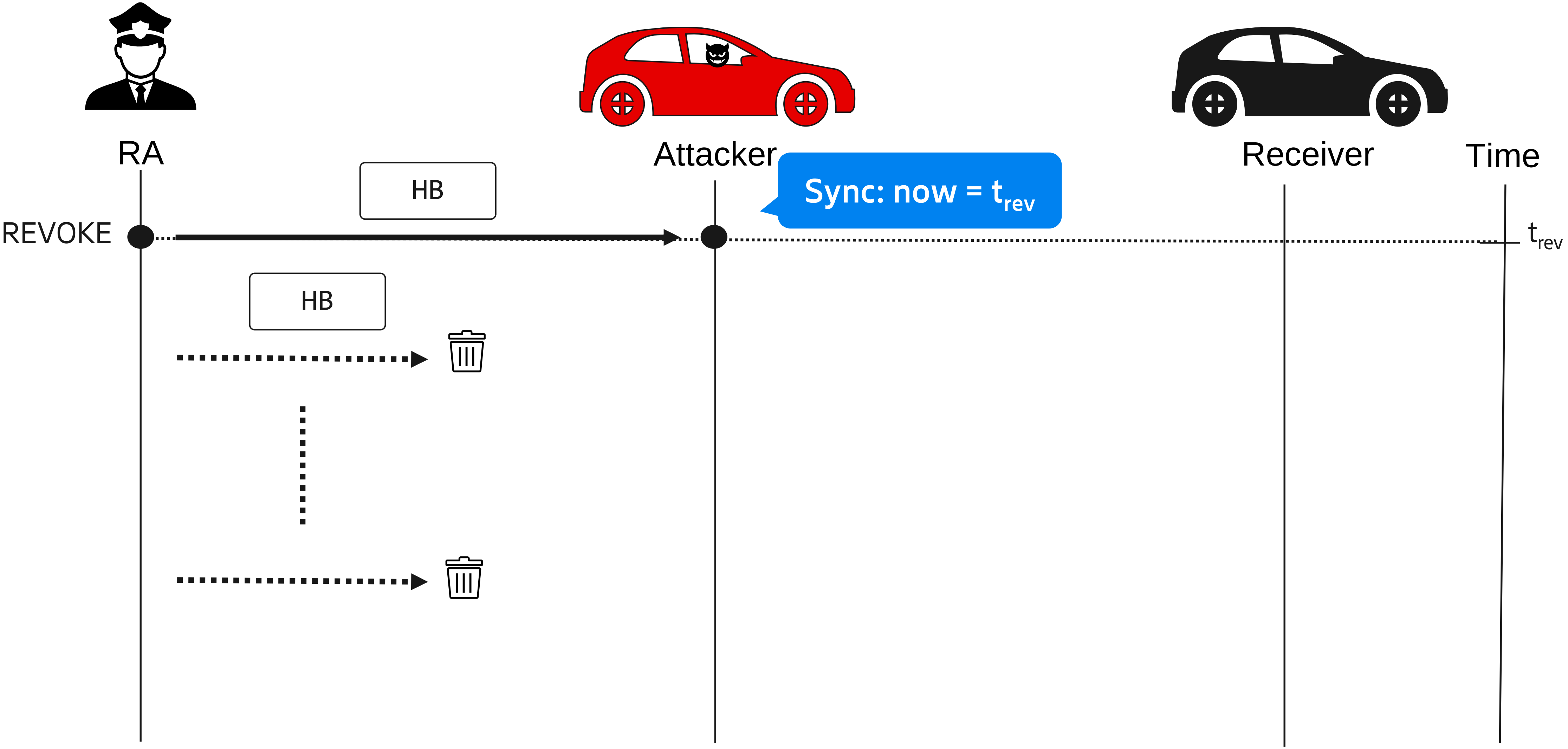
HB

REVOKE

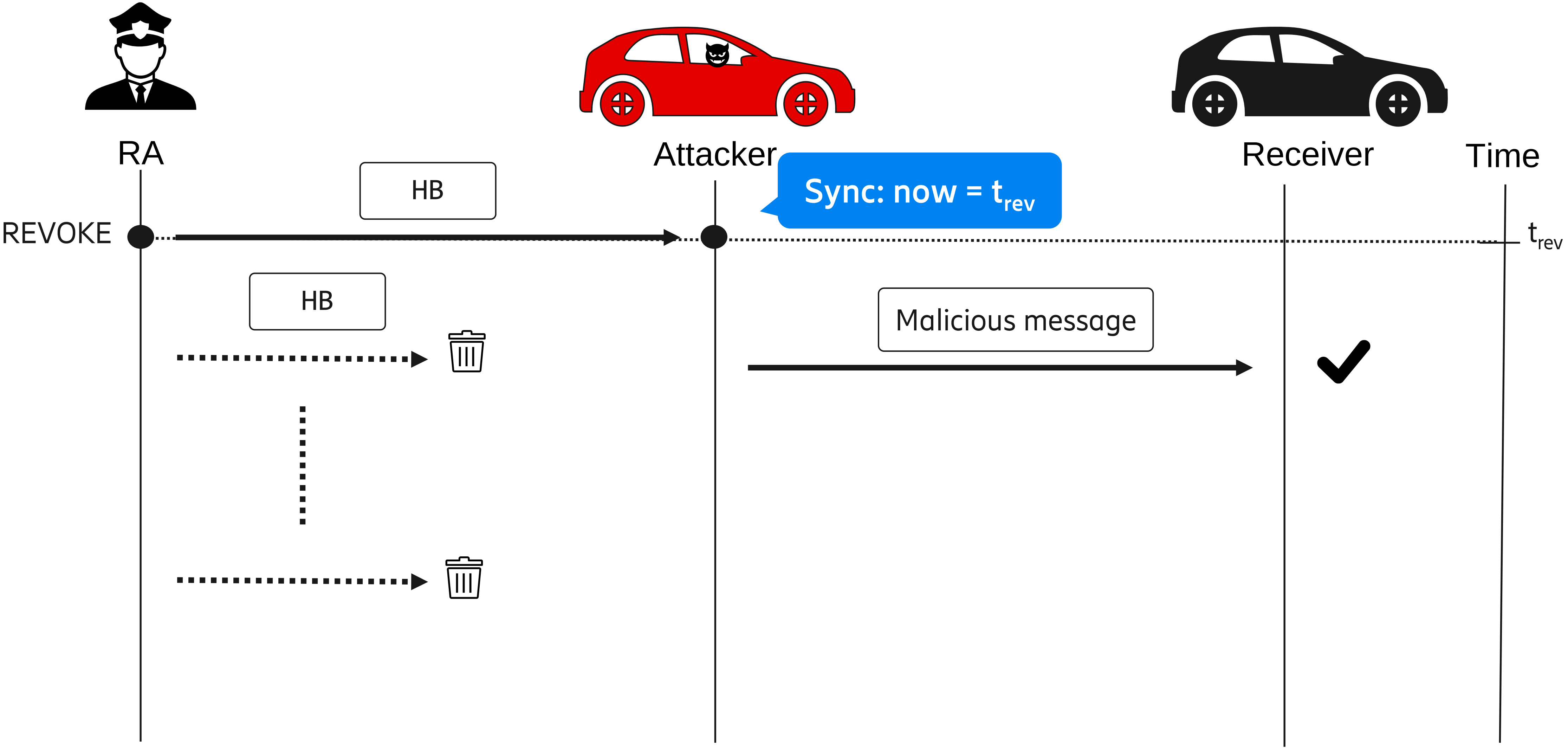
Sync: now = t_{rev}

t_{rev}

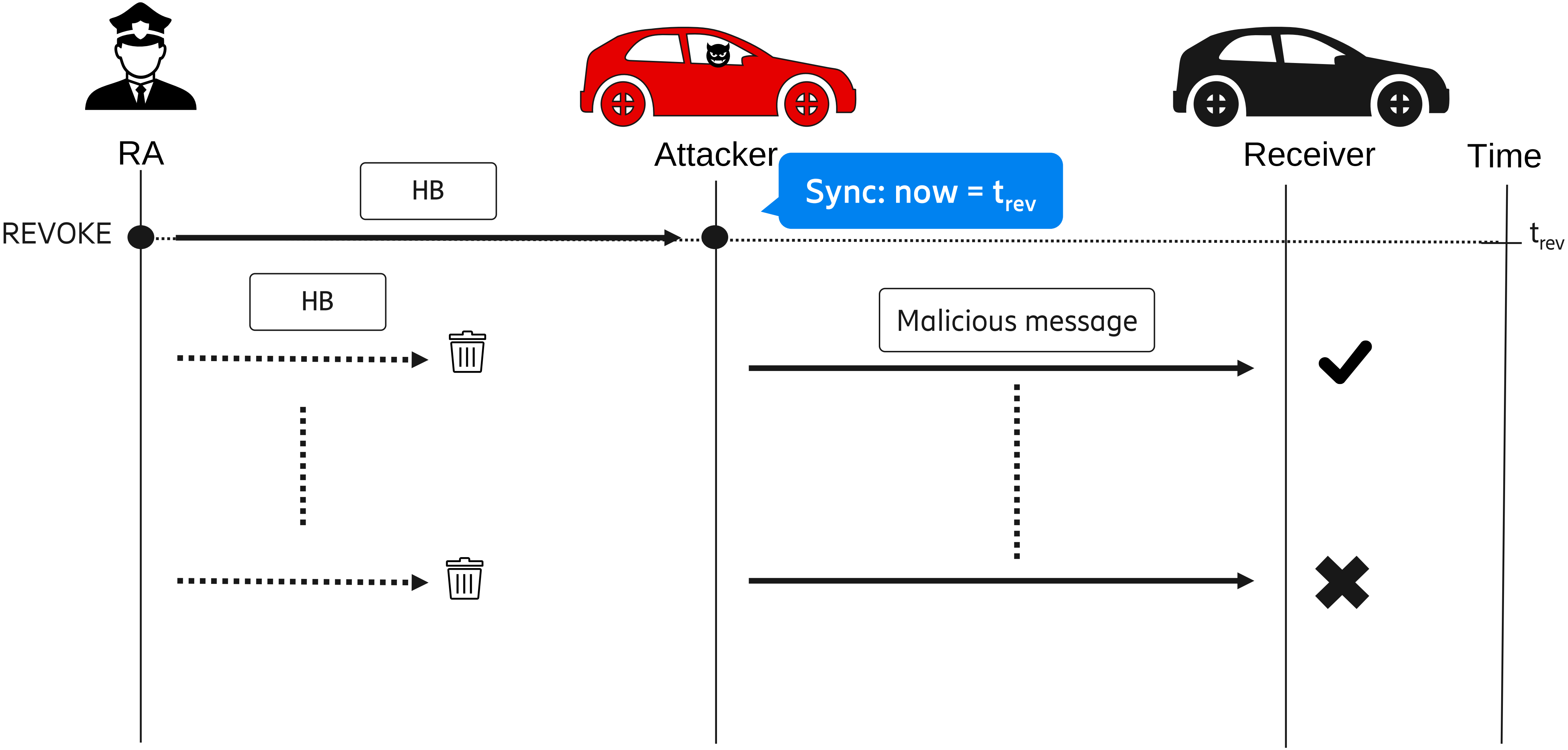
Worst-case T_{eff}



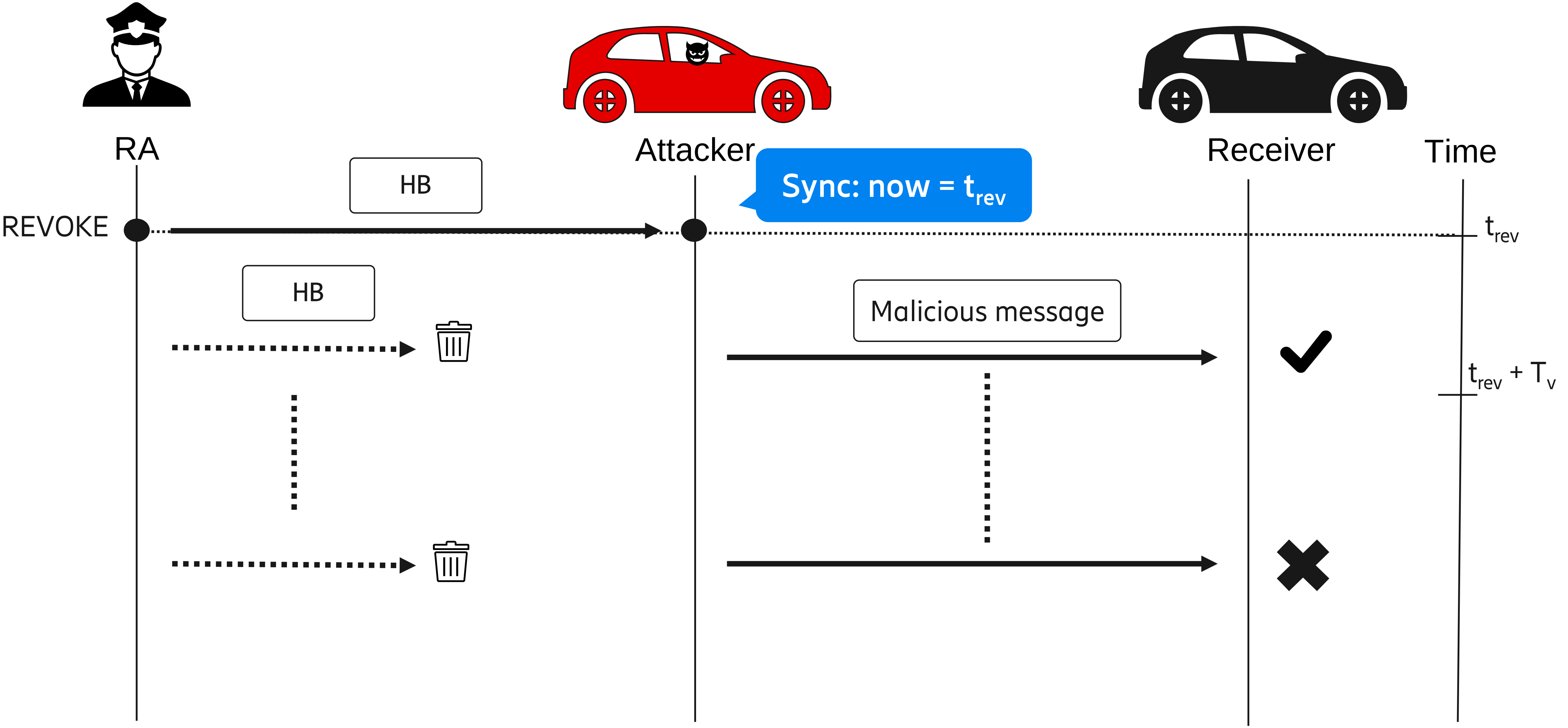
Worst-case T_{eff}



Worst-case T_{eff}

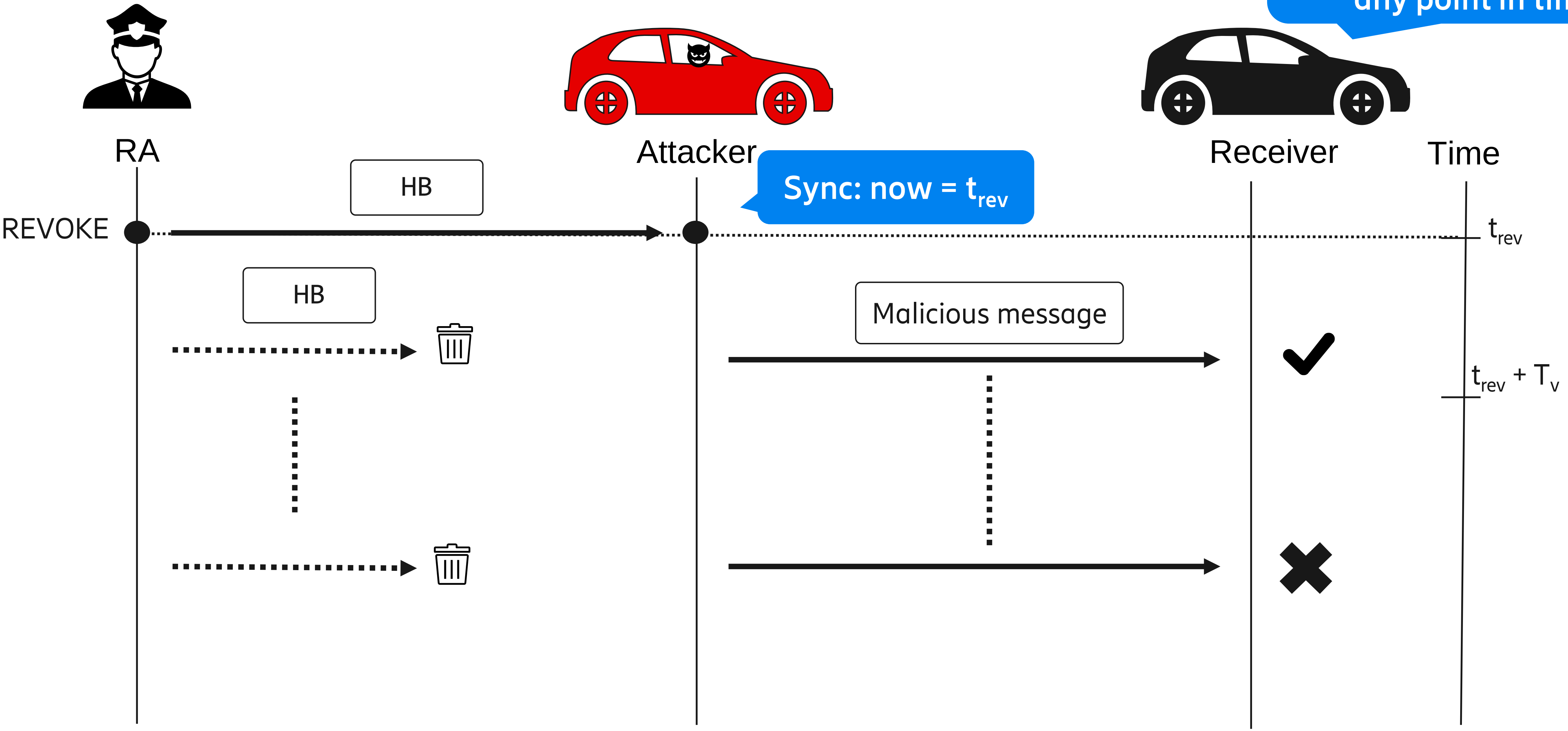


Worst-case T_{eff}



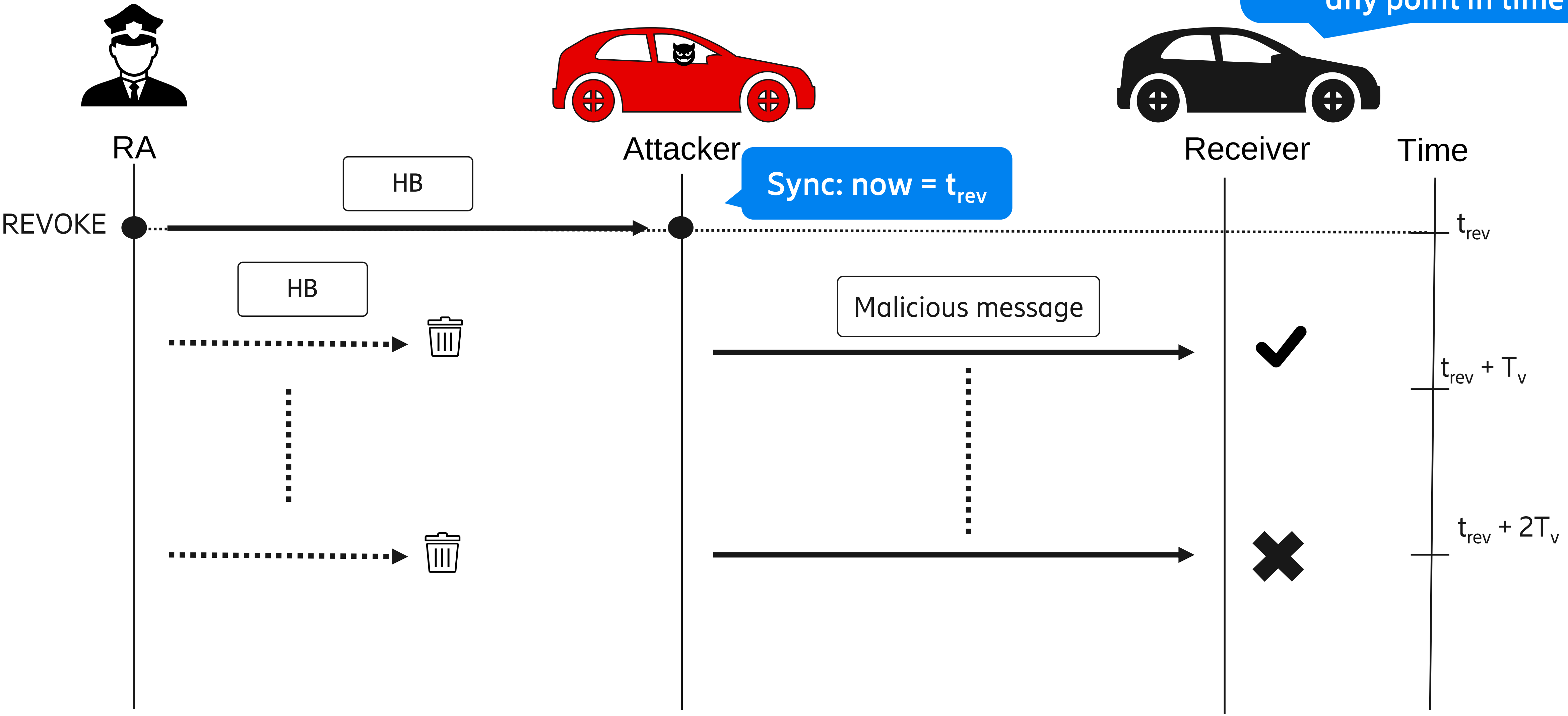
Worst-case T_{eff}

Assumption: receiver is at most T_v behind the RA at any point in time



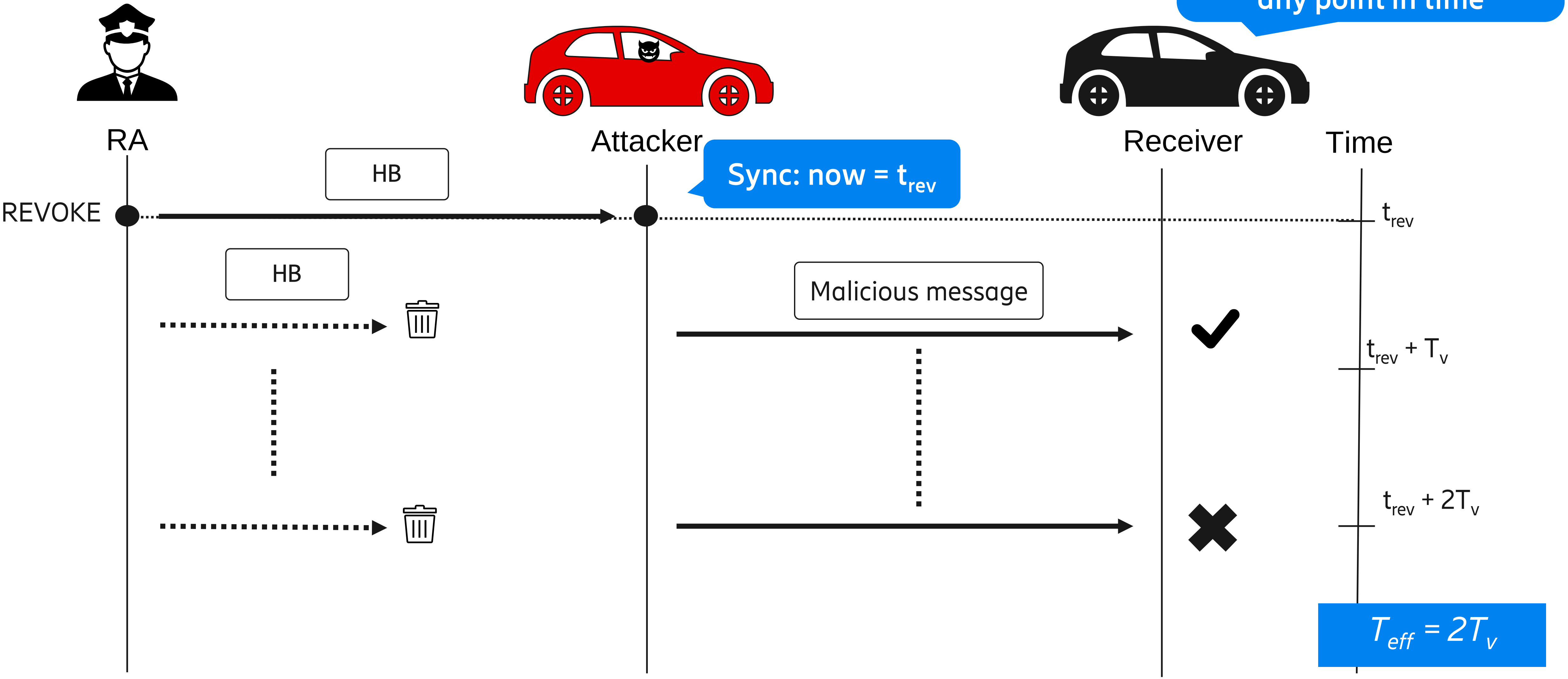
Worst-case T_{eff}

Assumption: receiver is at most T_v behind the RA at any point in time



Worst-case T_{eff}

Assumption: receiver is at most T_v behind the RA at any point in time



Sync: now = t_{rev}

$T_{eff} = 2T_v$