Funded by
the European Union

European Research Council
Established by the European Commission

CROSSING
Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

PIRANHAS: Privacy-Preserving Remote Attestation in Non-Hierarchical Asynchronous Swarms

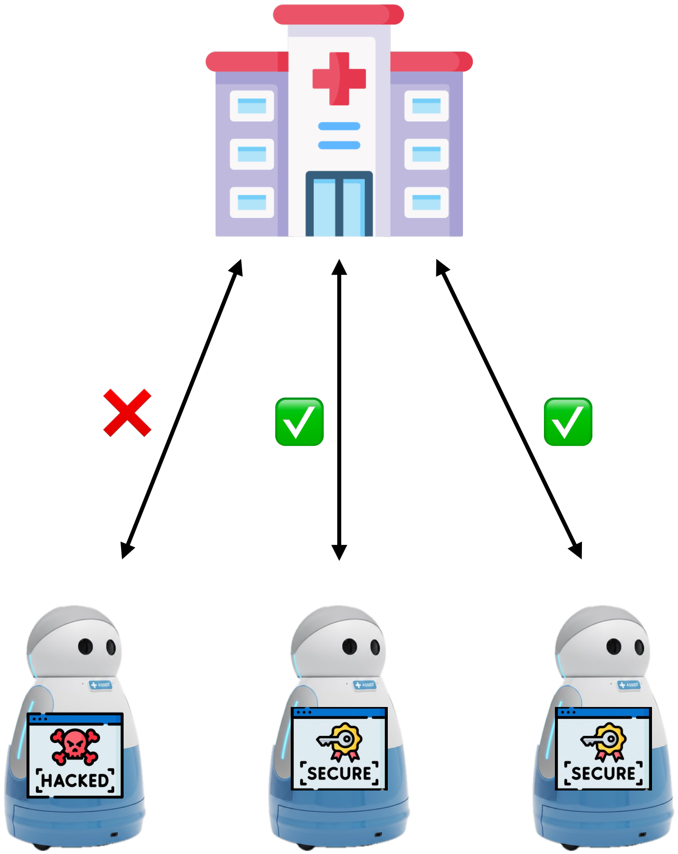
Artifact Evaluated
NDSS SYMPOSIUM
Available
Functional
Reproduced

Jonas Hofmann[1], ***Philipp-Florens Lehwalder***[1], *Shahriar Ebrahimi*[2], *Parisa Hassanizadeh*[3], *Sebastian Faust*[1]

[1] TECHNISCHE UNIVERSITÄT DARMSTADT [2] The Alan Turing Institute [3] IPPT PAN

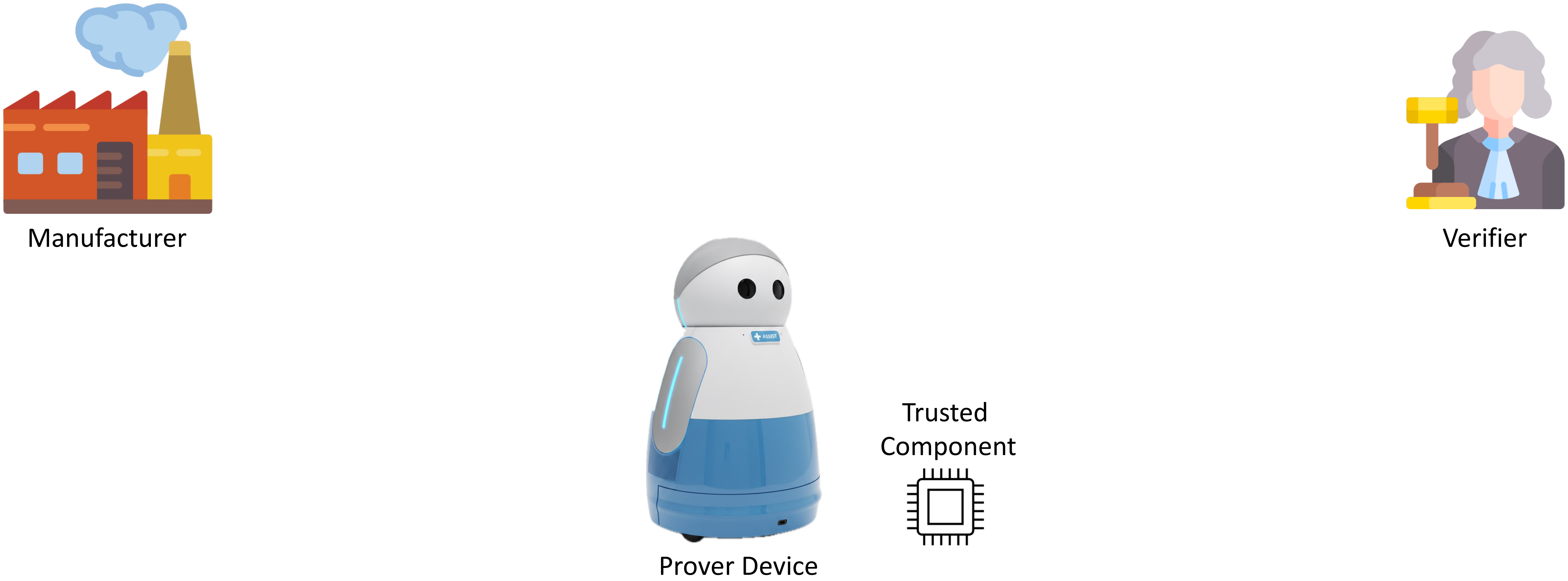
Remote Attestation

- Verify integrity of a device before trusting it with sensitive data



Traditional Remote Attestation

Setup



Manufacturer

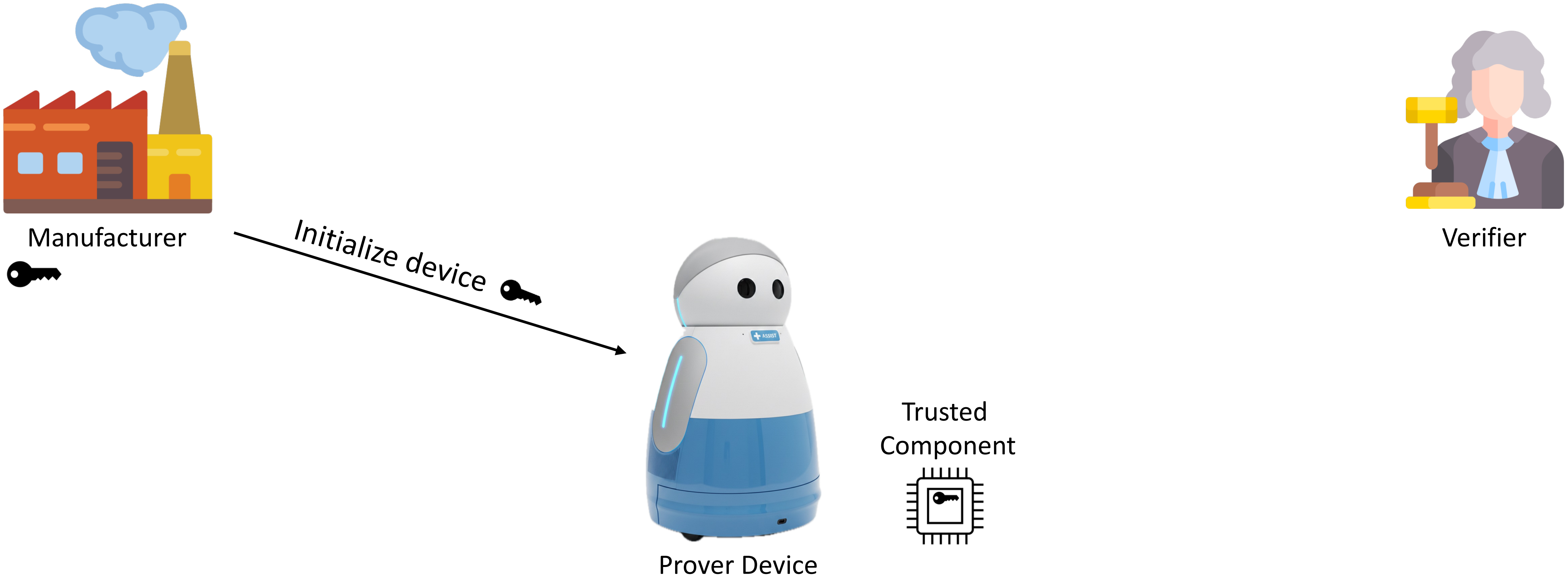
Verifier

Trusted Component

Prover Device

Traditional Remote Attestation

Setup



Manufacturer

Initialize device

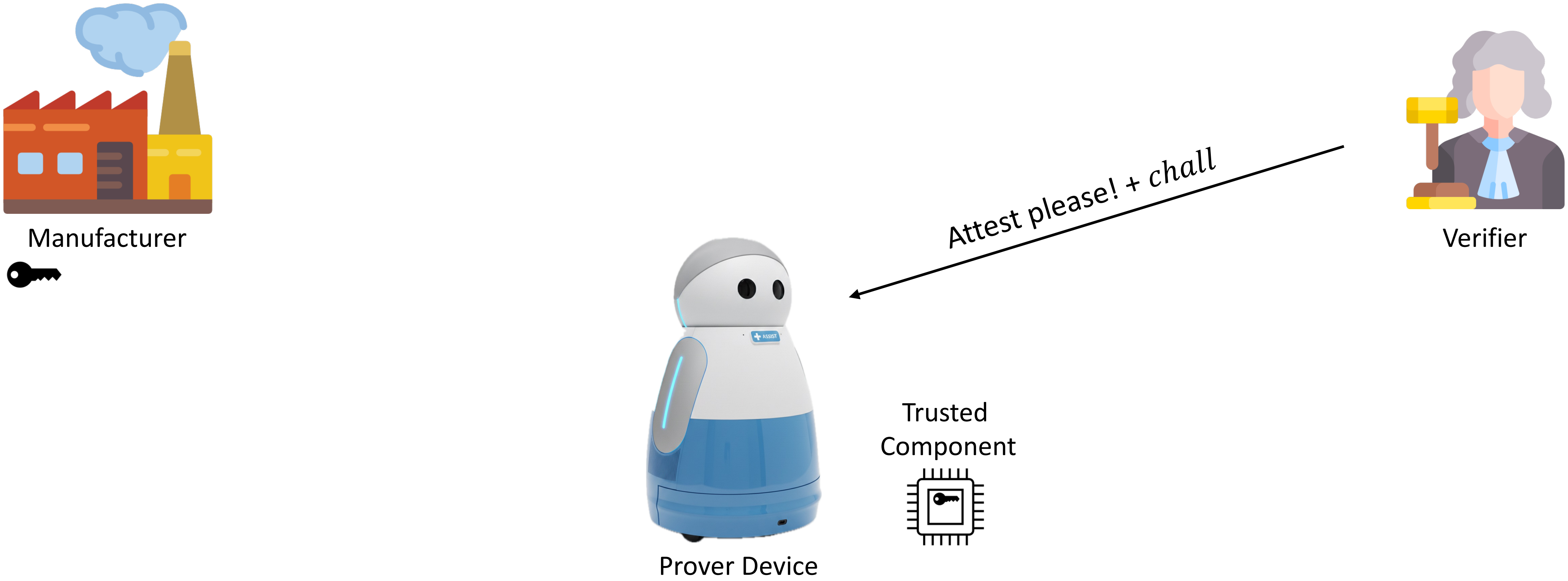
Verifier

Trusted Component

Prover Device

Traditional Remote Attestation

Attest



Manufacturer

Attest please! + $chall$

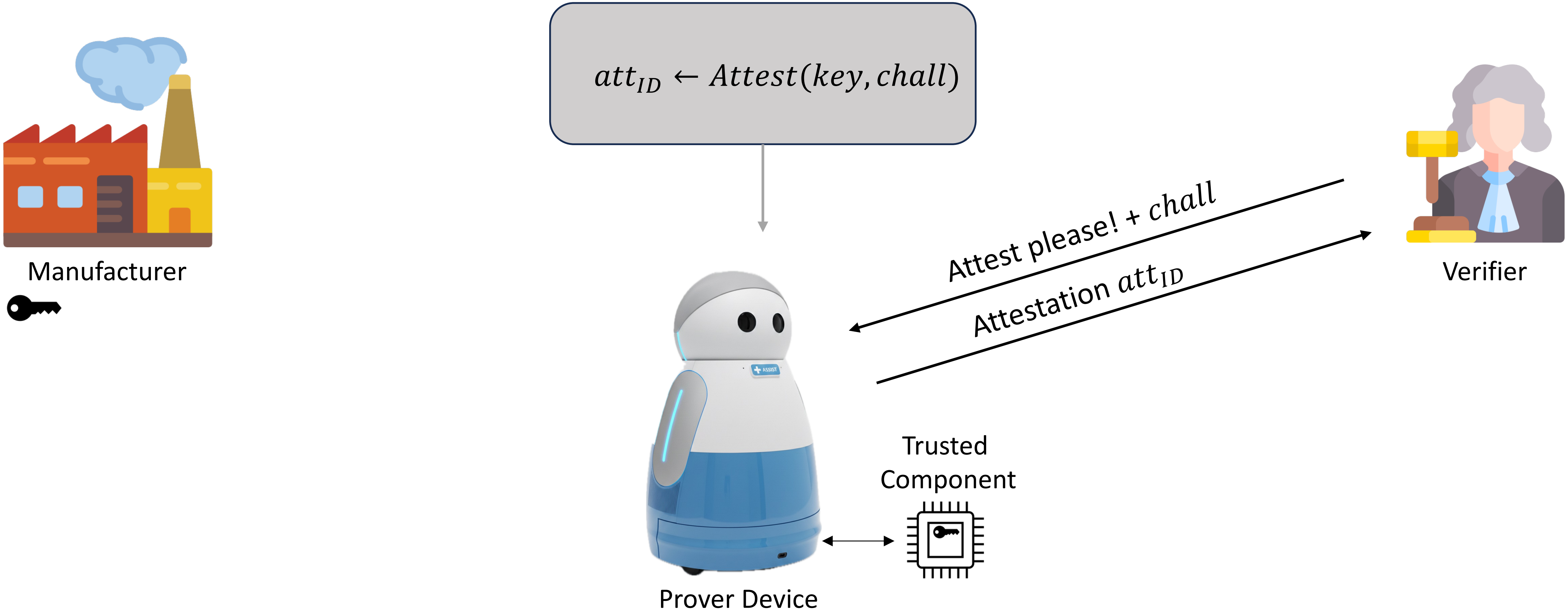
Verifier

Trusted Component

Prover Device

Traditional Remote Attestation

Attest



$att_{ID} \leftarrow Attest(key, chall)$

Manufacturer

Attest please! + $chall$

Attestation att_{ID}

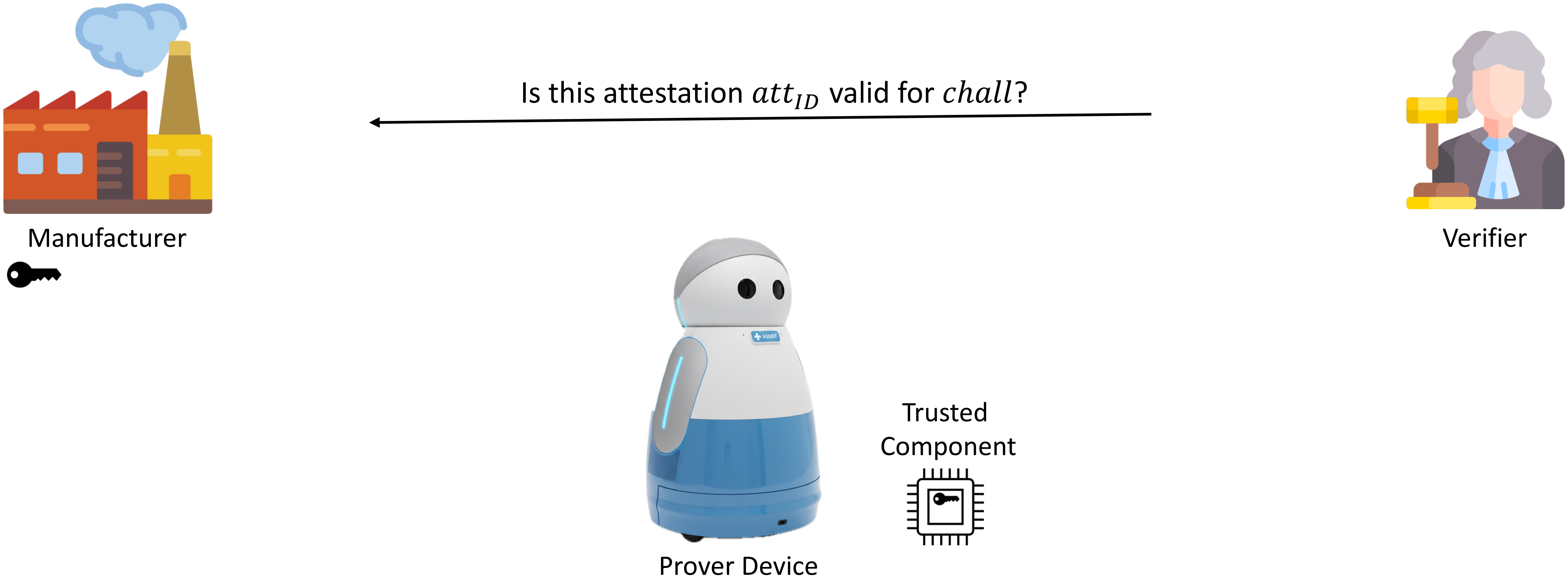
Verifier

Trusted Component

Prover Device

Traditional Remote Attestation

Verification



Is this attestation att_{ID} valid for $chall$?

Manufacturer

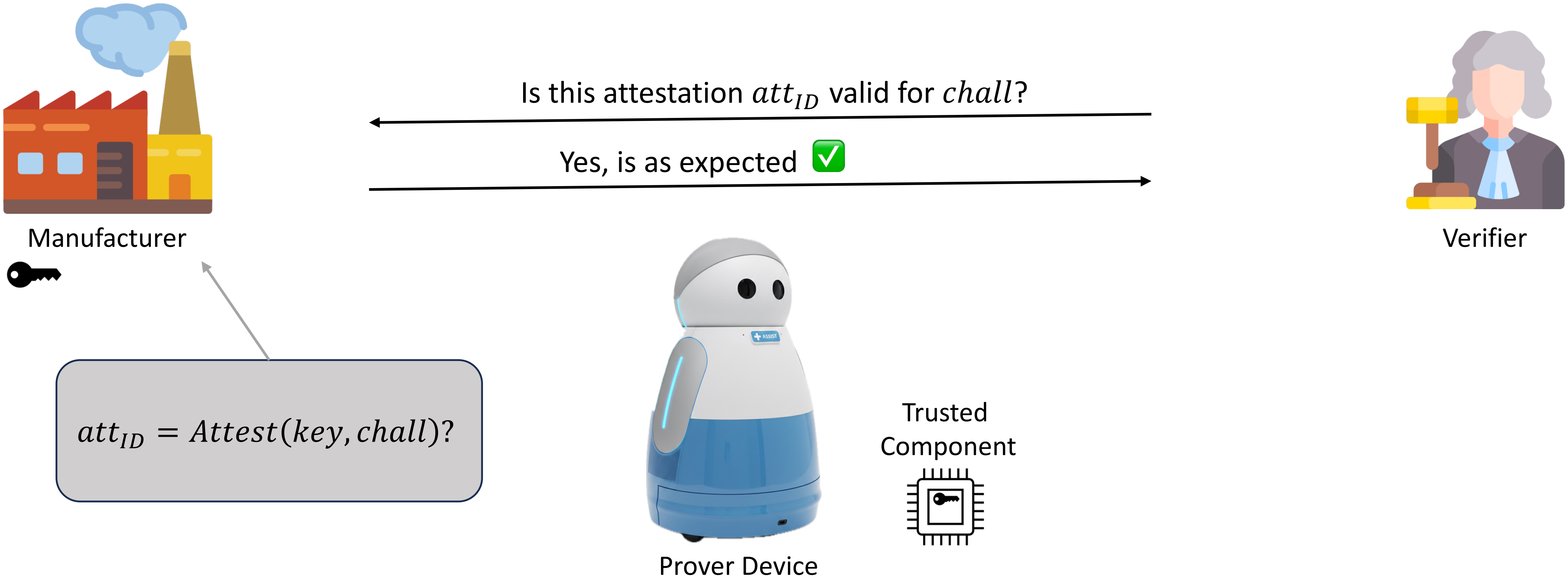
Verifier

Trusted Component

Prover Device

Traditional Remote Attestation

Verification



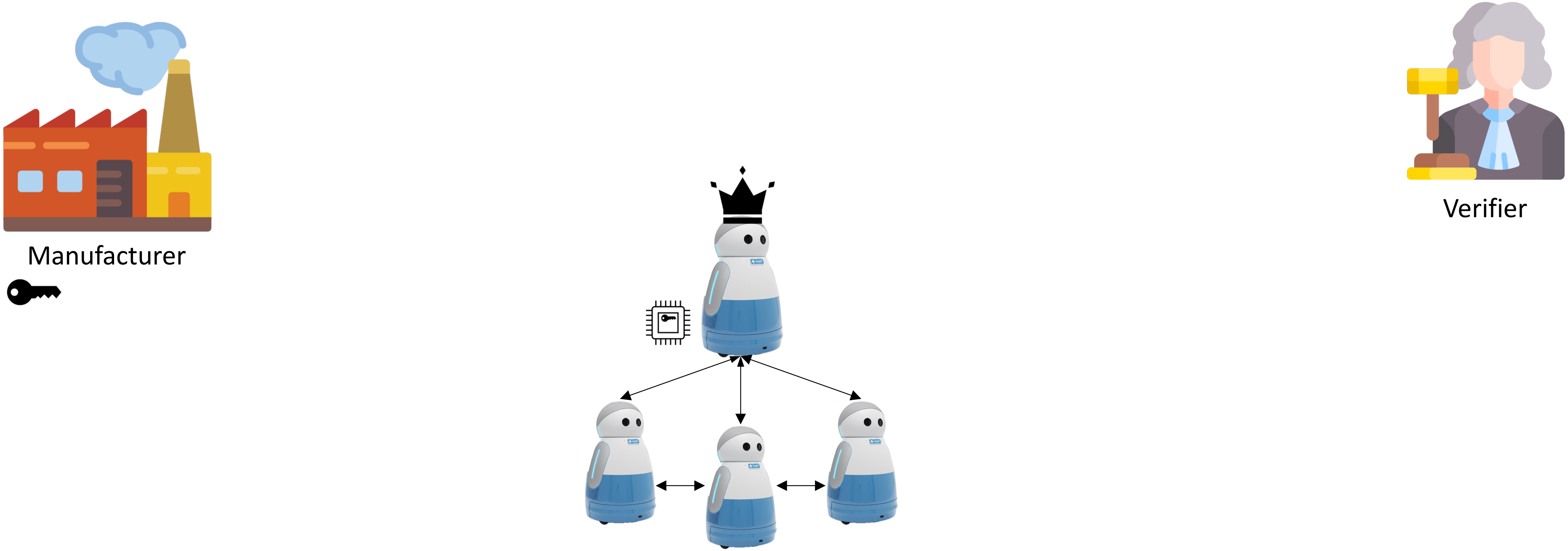
Is this attestation att_{ID} valid for $chall$?

Yes, is as expected ✅

$att_{ID} = Attest(key, chall)$?

Traditional Swarm Attestation

- Should be faster than individual attestations

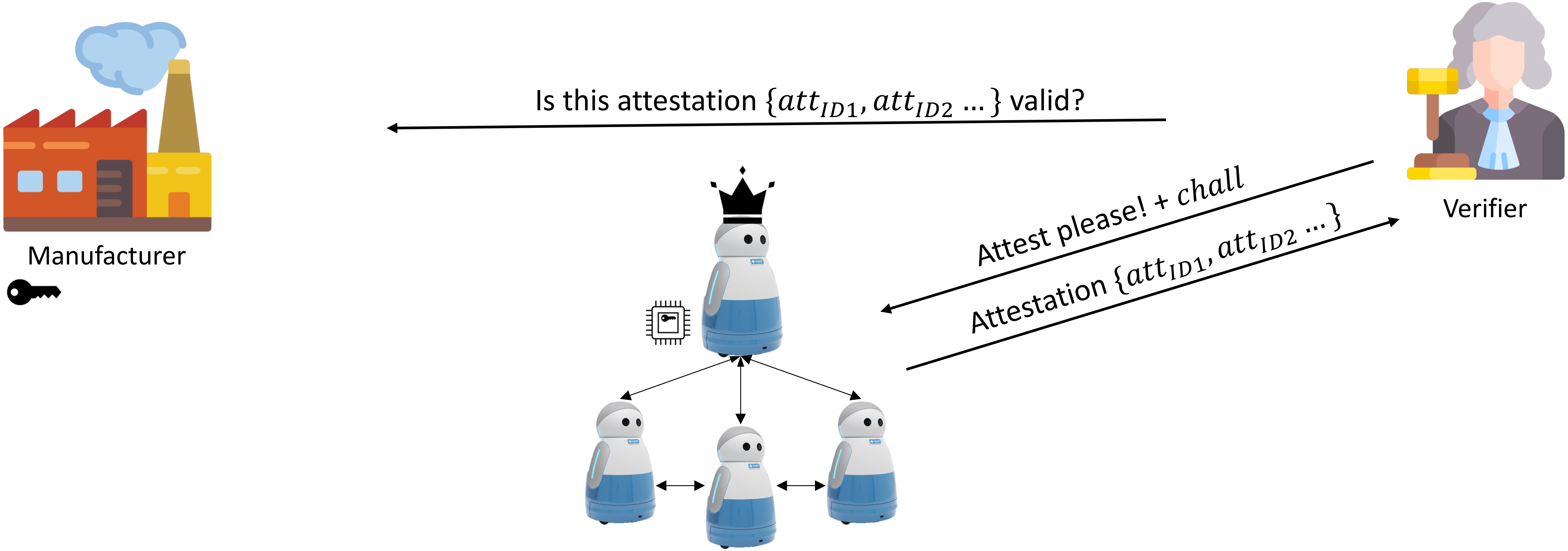


Manufacturer

Verifier

Traditional Swarm Attestation

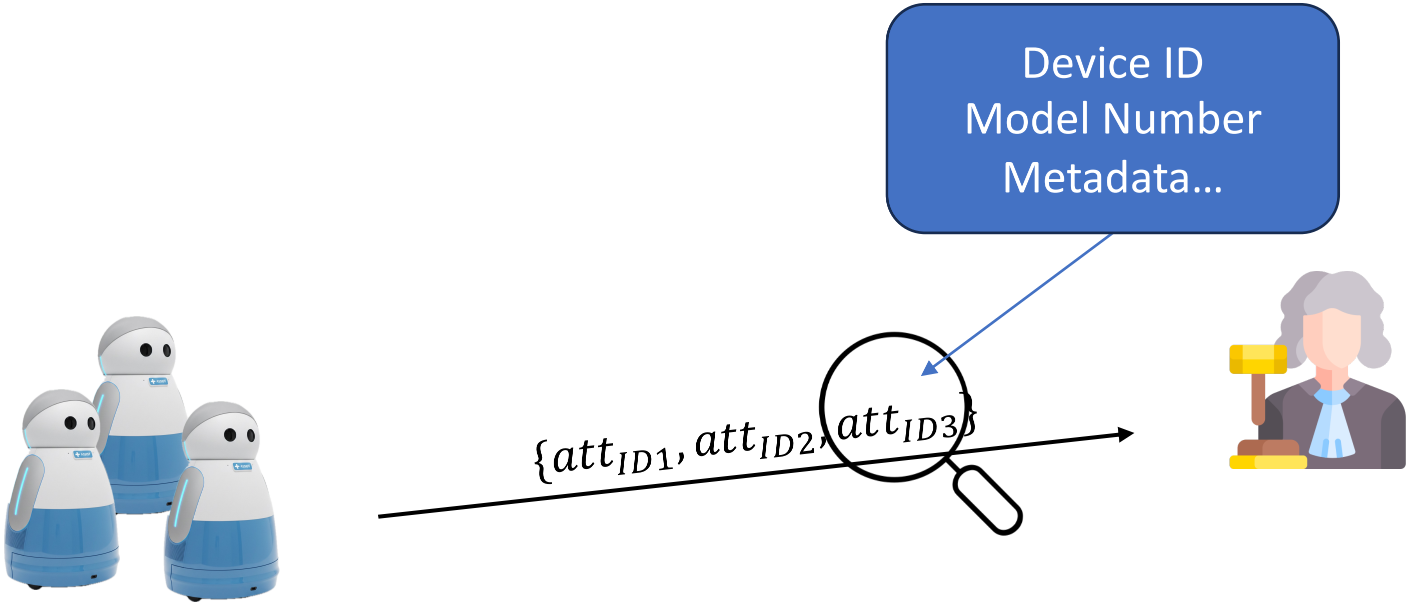
- ➢Should be faster than individual attestations


Is this attestation $\{att_{ID1}, att_{ID2} \ldots\}$ valid?
Manufacturer
Verifier
Attest please! + $chall$
Attestation $\{att_{ID1}, att_{ID2} \ldots\}$

Traditional Remote/Swarm Attestation

Lack of Anonymity

- Attestation reveals (at least) identifiers
- Enables tracking of devices



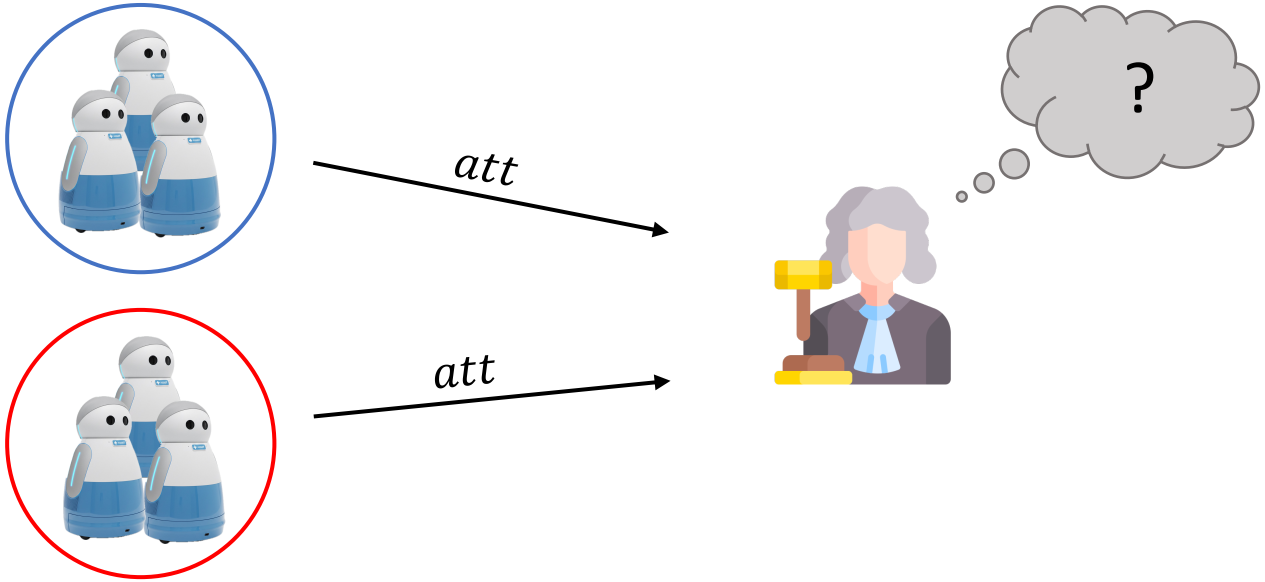
Device ID
Model Number
Metadata…

$\{att_{ID1}, att_{ID2}, att_{ID3}\}$

Anonymous Remote/Swarm Attestation

Anonymity

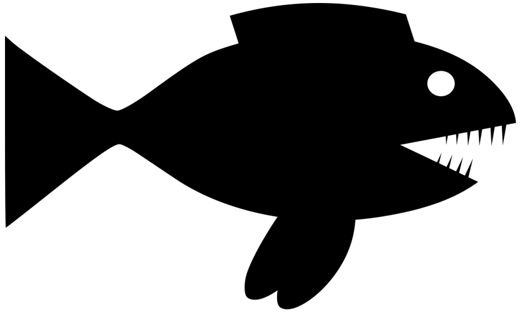
- ➢Attestation should reveal nothing beyond validity
- ➢Achieve unlinkability of attestations



Swarm Attestation Schemes

Scheme	Anonymous	Pub. Verifiable	Non-Interactive	Topology
SEDA [ABI+15]	❌	❌	❌	Spanning Tree
Leg-IoT [NDK+20]	❌	✅	✅	any
SCRAPS [PYD+22]	❌	✅	❌	Pub-Sub
Privé [EHS+25]	(✅) not within swarm	❌	❌	Hierarchical
SPARK [HKR+25]	(✅) not within swarm	✅	✅	Hierarchical
PIRANHAS	✅	✅	✅	any

PIRANHA & PIRANHAS



PIRANHA

Transform any traditional RA scheme to be:

➢Non-interactive

➢Publicly-verifiable

➢Anonymous

Using zk-SNARKs

Succinct non-interactive zero-knowledge proofs

Based on zRA (Ebrahimi et al. NDSS'24)

PIRANHAS

Transform any traditional RA scheme to a **swarm attestation** scheme:

- Non-interactive
- Publicly-verifiable
- Anonymous → Verifier only learns the size of the swarm

Using **recursive** zk-SNARKs

zk-SNARKs that can verify another SNARK proof

PIRANHA(S)

Setup

1. Manufacturer samples $chall_1, \ldots, chall_n$

Same for all devices

2. Precompute $att_i \leftarrow Attest(key, chall_i)$ for $i \in 1, \ldots, n$
3. Accumulate all att_i in Merkle tree $\mathcal{T}$
4. Sign $\sigma \leftarrow Sign(sk, \mathcal{T})$

For each device

➢Provide $(\sigma, \mathcal{T})$ to device

PIRANHA(S)

Challenge Publication

Periodically publish new $chall$

Attestation

1. Retrieve current $chall$
2. Trusted component computes $att \leftarrow Attest(key\ , chall)$
3. Create ZKP $\pi \leftarrow Prove(att, chall, \mathcal{T}, \sigma)$:
   - *“Attestation att is contained in Merkle tree $\mathcal{T}$”*
   - *“I know a signature σ on $\mathcal{T}$ valid under manufacturer pk”*

PIRANHA(S)

Challenge Publication

Periodically publish new $chall$

Verification

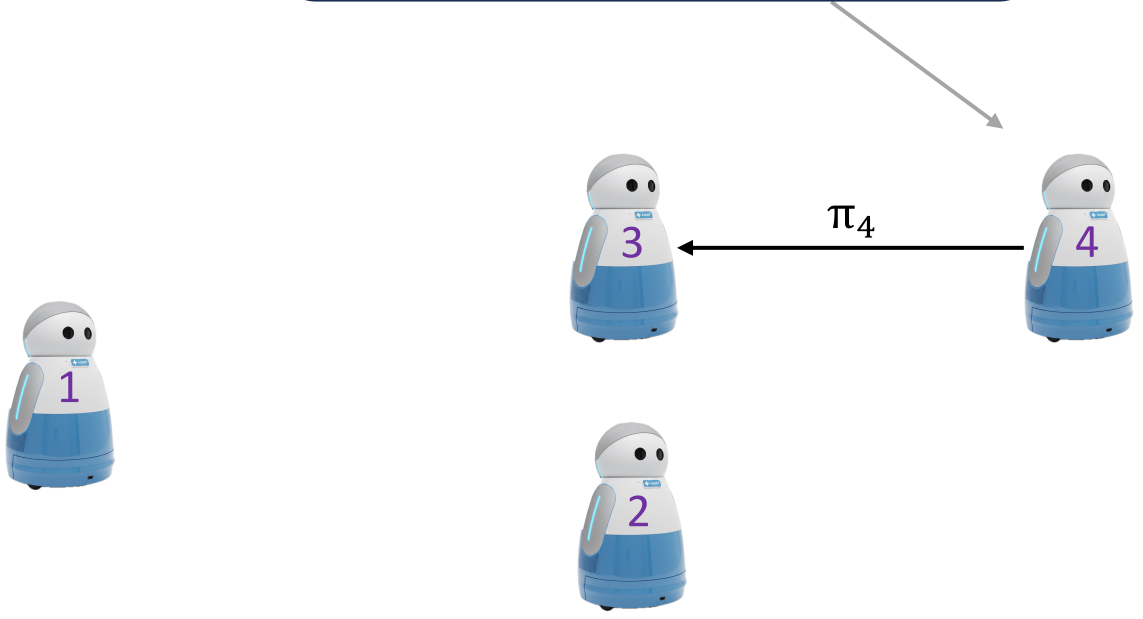
1. Retrieve current $chall$
2. Check $1 = Verify(\pi, chall, pk)$

➢Verification only using $chall$ and manufacturer pk!

PIRANHAS

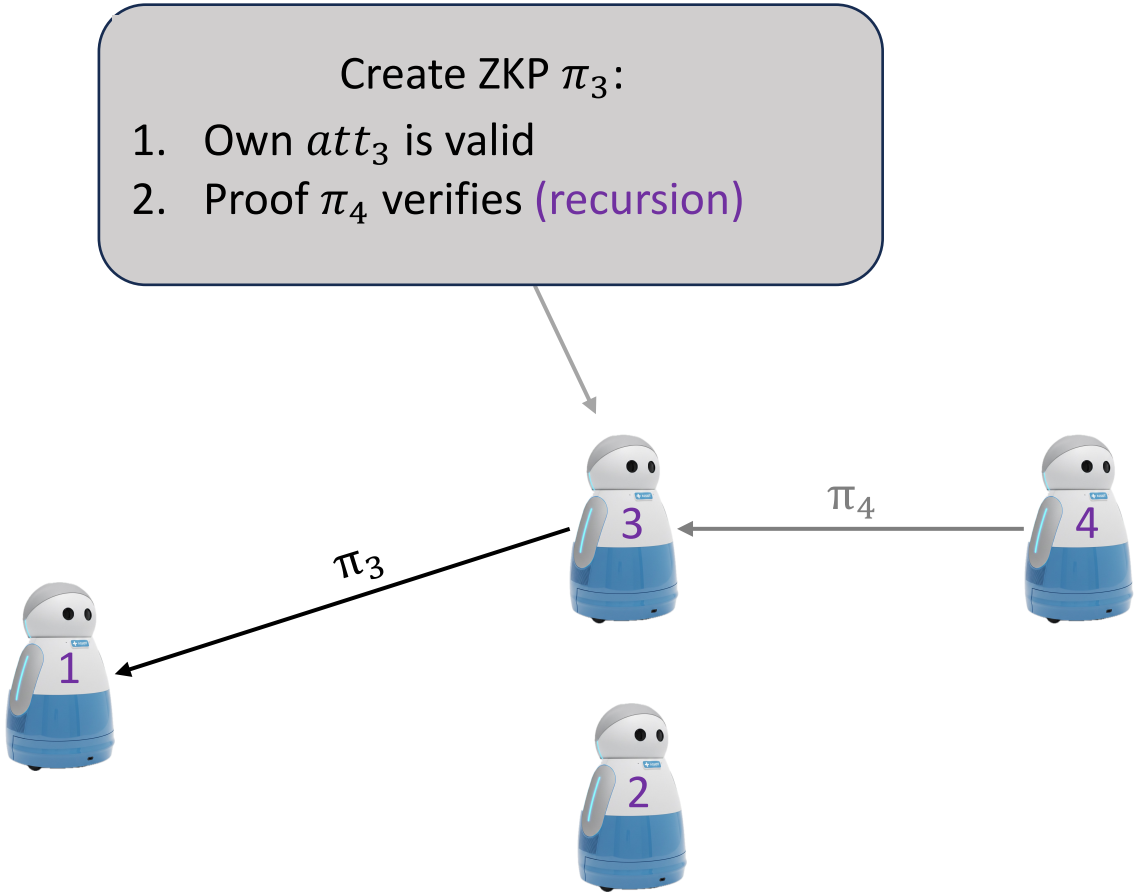
Swarm Attestation

1. Compute attestation att_4
2. Create ZKP π_4 that att_4 is valid



PIRANHAS

Swarm Attestation



Create ZKP π_3:

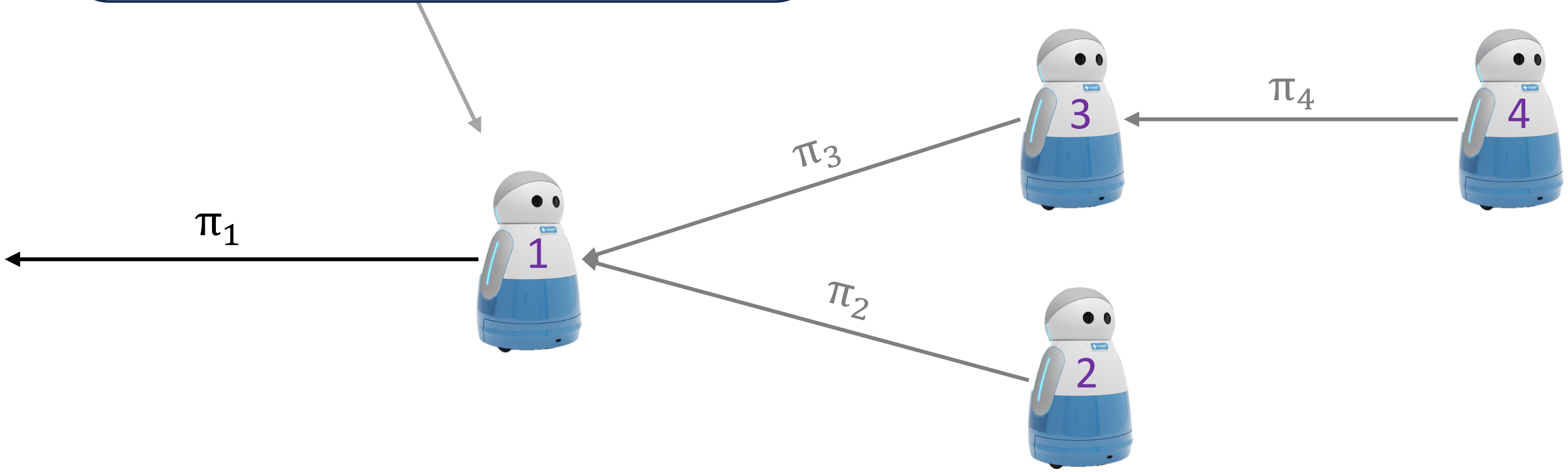
1. Own att_3 is valid
2. Proof π_4 verifies (recursion)

PIRANHAS

Swarm Attestation

Create ZKP π_1:

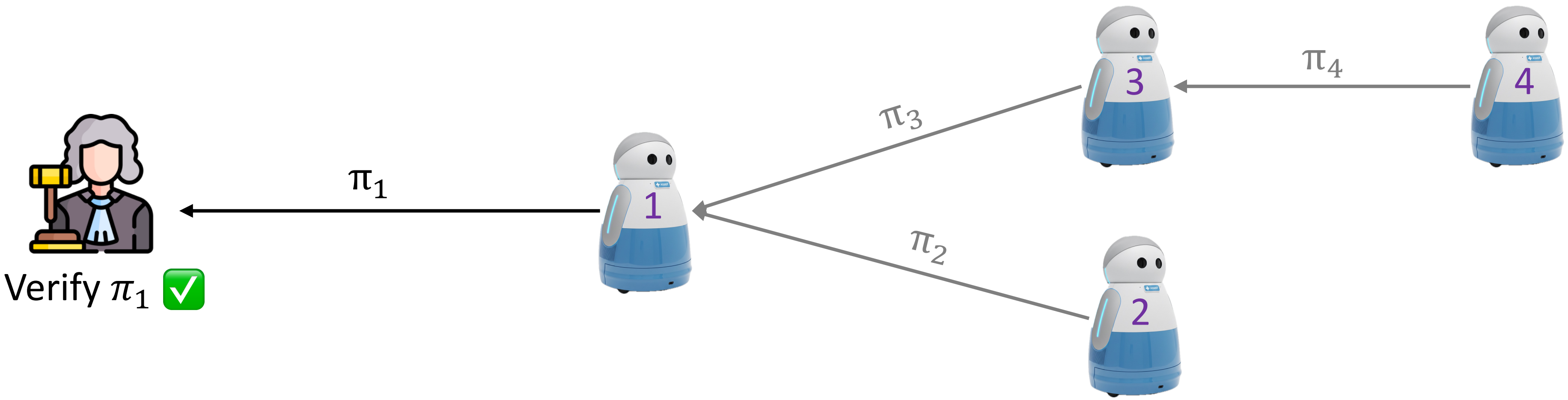
1. Own att_1 is valid
2. Proof π_2 & π_3 verifies (recursion)



PIRANHAS

Swarm Attestation

➢Single proof convinces of a correctly attested swarm



Verify π_1 ✅

π_1

π_3

π_2

π_4

1

2

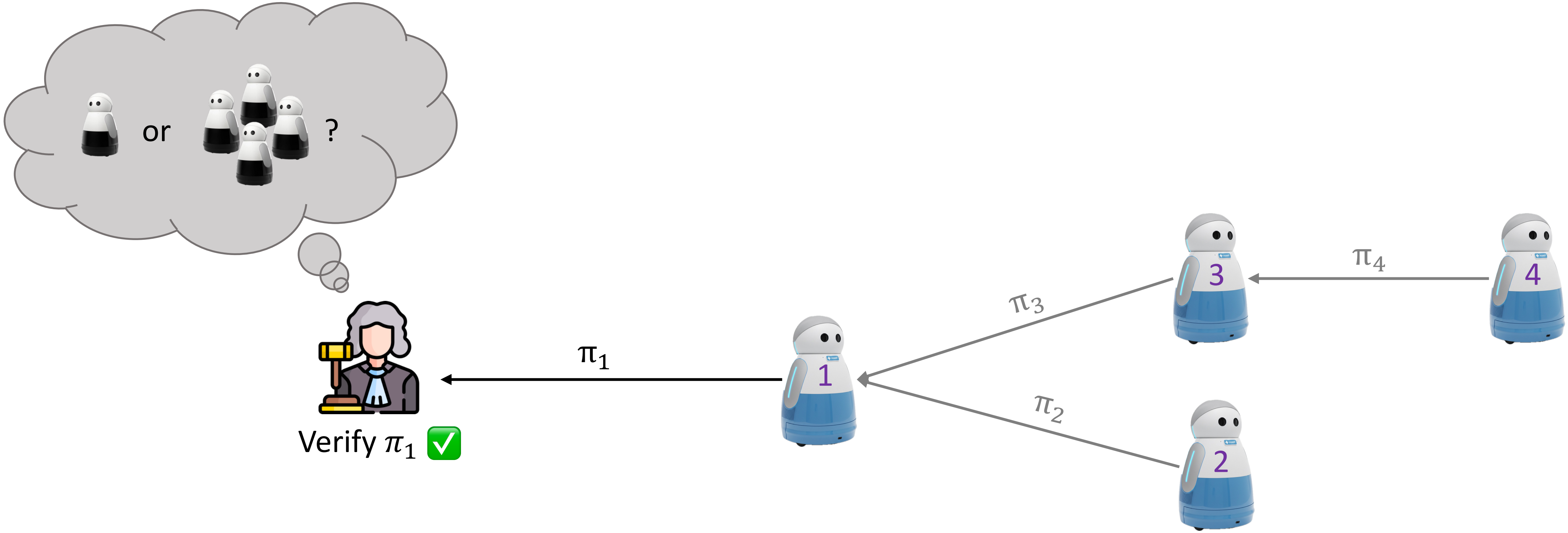
3

4

PIRANHAS

Swarm Attestation

- How does the verifier know the swarm size?



or ?

π_1

π_2

π_3

π_4

1

2

3

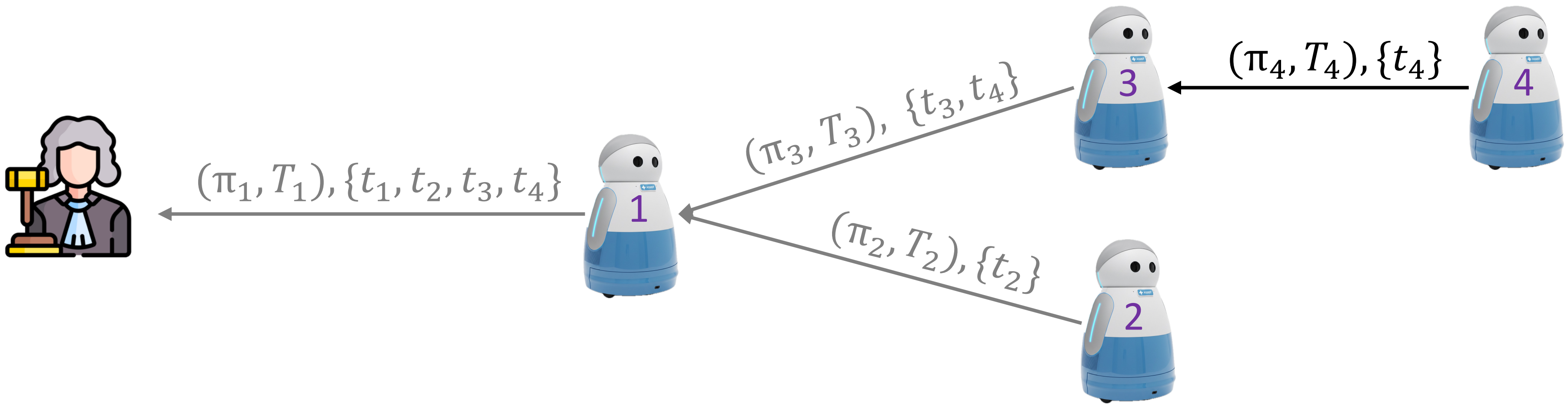
4

Verify π_1 ✅

PIRANHAS

Swarm Attestation

- ➢Introduce “linkage tags” t_i
- ➢Unique for each device per $chall$
- ➢Tags are aggregated as hash product in T

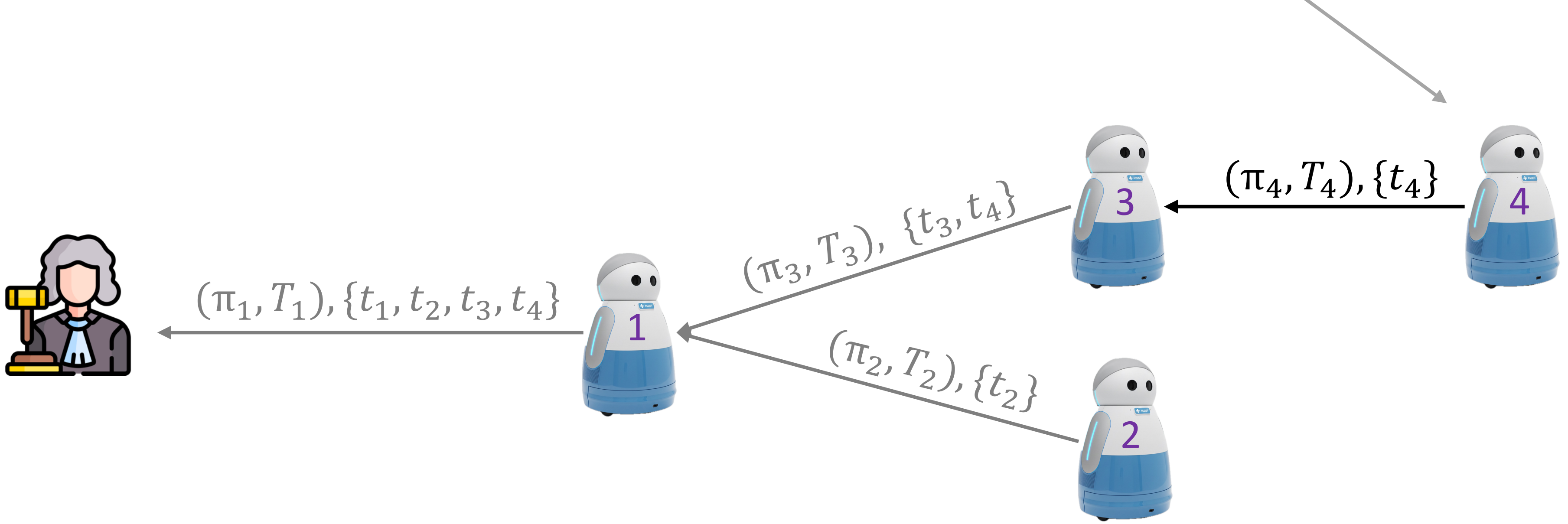

$(\pi_1, T_1), \{t_1, t_2, t_3, t_4\}$
1
$(\pi_3, T_3), \{t_3, t_4\}$
$(\pi_2, T_2), \{t_2\}$
3
$(\pi_4, T_4), \{t_4\}$
4
2

PIRANHAS

Swarm Attestation

- ➢Introduce “linkage tags” t_i
- ➢Unique for each device per $chall$
- ➢Tags are aggregated as hash product in T

1. Compute attestation att_4
2. **Compute $\boldsymbol{t_4} \leftarrow \boldsymbol{PRF}(\dots, \boldsymbol{chall})$**
3. Create ZKP π_4: att$_4$ is valid,
   $\boldsymbol{t_4}$ is correctly computed,
   initial **agg. tag is $\boldsymbol{T_4} = \boldsymbol{H}(\boldsymbol{t_4})$**

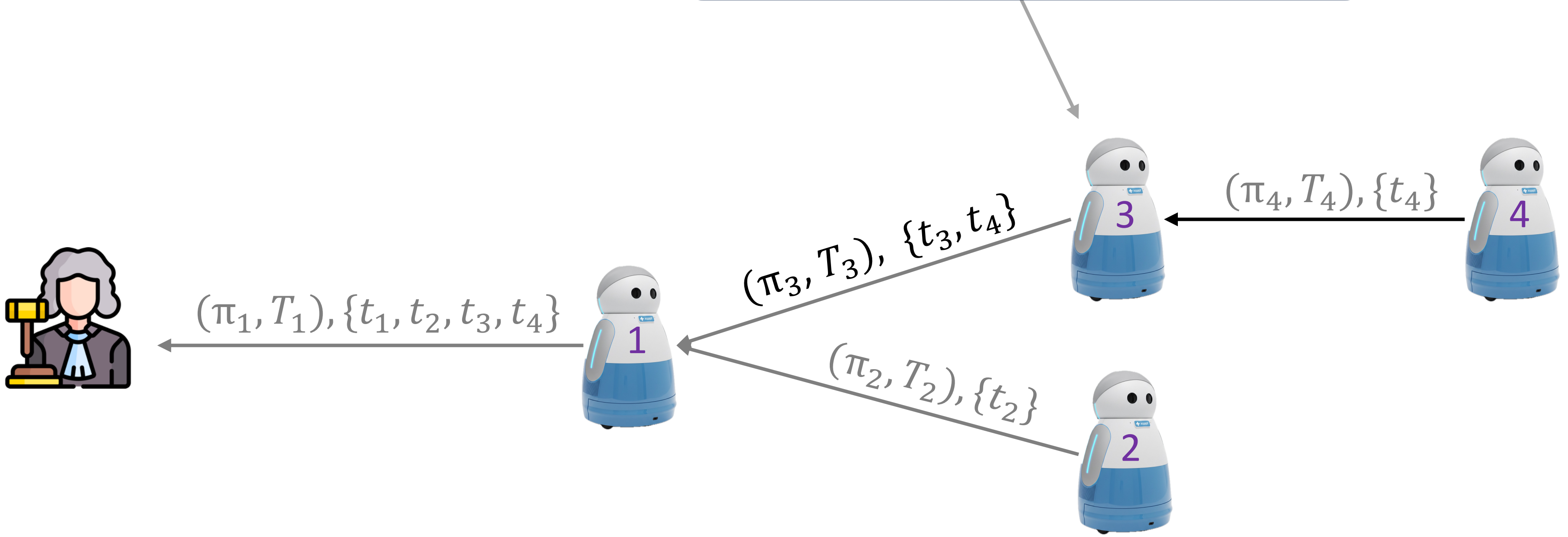

$(\pi_4, T_4), \{t_4\}$
4
3
$(\pi_3, T_3),\ \{t_3, t_4\}$
$(\pi_2, T_2), \{t_2\}$
2
1
$(\pi_1, T_1), \{t_1, t_2, t_3, t_4\}$

PIRANHAS

Swarm Attestation

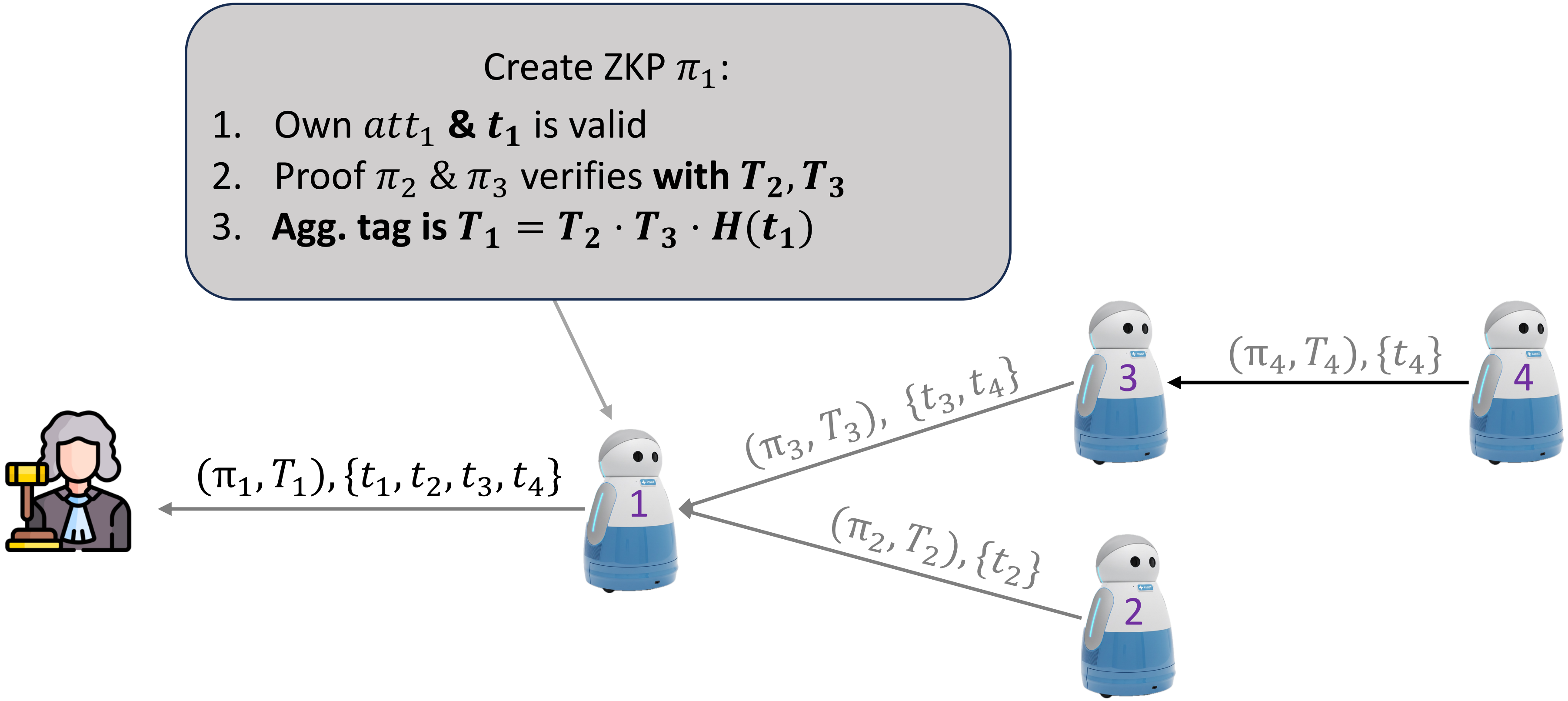
Create ZKP π_3:

1. Own att_3 **&** $\boldsymbol{t_3}$ is valid
2. Proof π_4 verifies **with** $\boldsymbol{T_4}$
3. **Agg. tag is** $\boldsymbol{T_3 = T_4 \cdot H(t_3)}$


$(\pi_4, T_4), \{t_4\}$
4
3
$(\pi_3, T_3), \{t_3, t_4\}$
$(\pi_2, T_2), \{t_2\}$
2
1
$(\pi_1, T_1), \{t_1, t_2, t_3, t_4\}$

PIRANHAS

Swarm Attestation



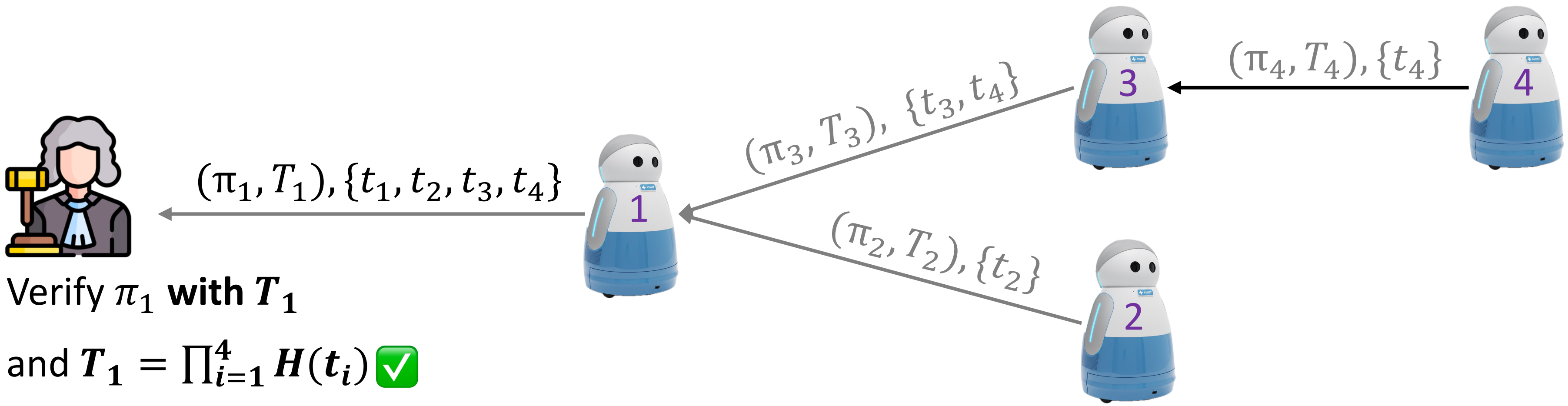
Create ZKP π_1:

1. Own att_1 **&** $\boldsymbol{t_1}$ is valid
2. Proof π_2 & π_3 verifies **with** $\boldsymbol{T_2, T_3}$
3. **Agg. tag is** $\boldsymbol{T_1 = T_2 \cdot T_3 \cdot H(t_1)}$

$(\pi_1, T_1), \{t_1, t_2, t_3, t_4\}$
1
$(\pi_3, T_3), \{t_3, t_4\}$
3
$(\pi_4, T_4), \{t_4\}$
4
$(\pi_2, T_2), \{t_2\}$
2

PIRANHAS

Swarm Attestation



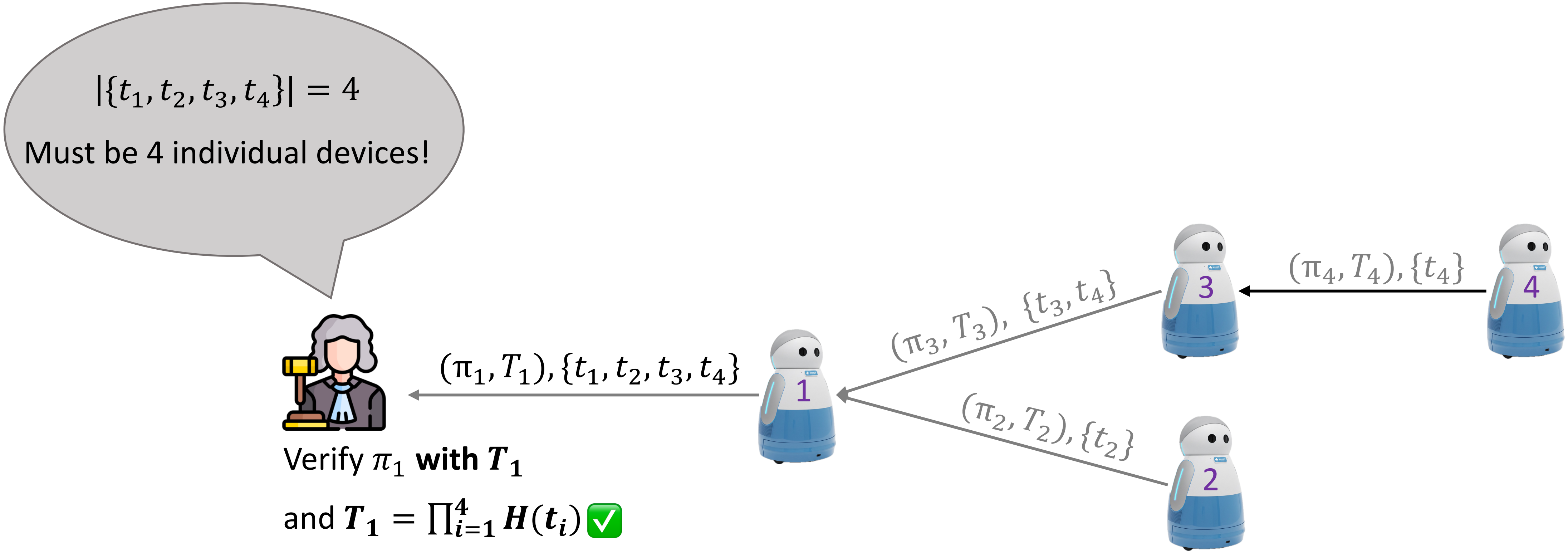
$(\pi_4, T_4), \{t_4\}$
$(\pi_3, T_3), \{t_3, t_4\}$
$(\pi_2, T_2), \{t_2\}$
$(\pi_1, T_1), \{t_1, t_2, t_3, t_4\}$
1
2
3
4

Verify π_1 **with** $\boldsymbol{T_1}$

and $\boldsymbol{T_1} = \prod_{\boldsymbol{i=1}}^{\boldsymbol{4}} \boldsymbol{H}(\boldsymbol{t_i})$ ✅

PIRANHAS

Swarm Attestation



$|\{t_1, t_2, t_3, t_4\}| = 4$

Must be 4 individual devices!

$(\pi_1, T_1), \{t_1, t_2, t_3, t_4\}$

$(\pi_3, T_3), \{t_3, t_4\}$

$(\pi_4, T_4), \{t_4\}$

$(\pi_2, T_2), \{t_2\}$

1

2

3

4

Verify π_1 **with** $\boldsymbol{T_1}$

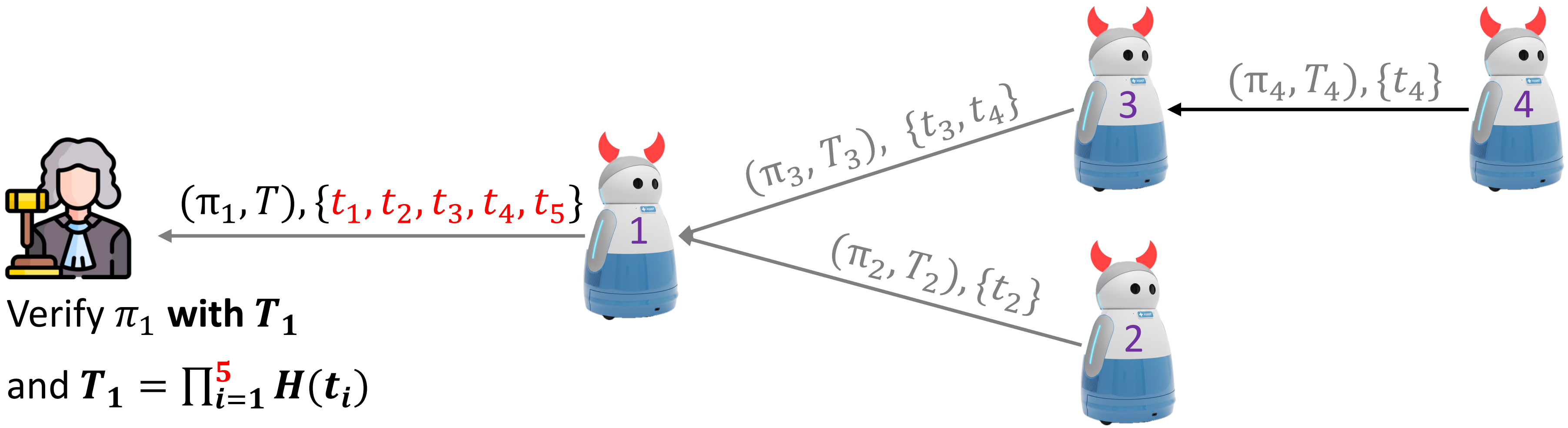
and $\boldsymbol{T_1} = \prod_{i=1}^{4} \boldsymbol{H(t_i)}$ ✅

PIRANHAS

Swarm Attestation

Must not be possible to output attestation verifying for more devices

➢Otherwise, must have broken DL assumption in ROM


$(\pi_4, T_4), \{t_4\}$
4
3
$(\pi_3, T_3),\ \{t_3, t_4\}$
$(\pi_2, T_2), \{t_2\}$
2
1
$(\pi_1, T), \{t_1, t_2, t_3, t_4, t_5\}$

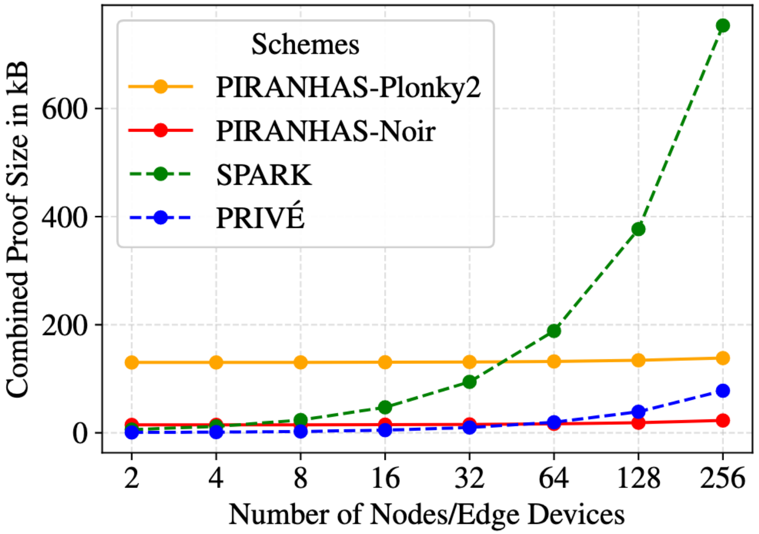
Verify π_1 **with** $\boldsymbol{T_1}$

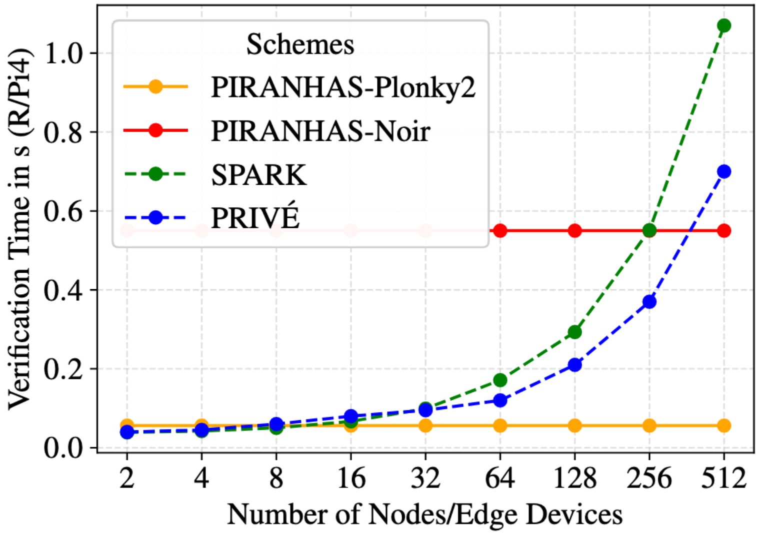
and $\boldsymbol{T_1} = \prod_{\boldsymbol{i=1}}^{\boldsymbol{5}} \boldsymbol{H(t_i)}$

Benchmarks

- Implemented using Noir and Plonky2
- Aggregation runtime of 356ms (Plonky2, laptop)
- Runtime of 8s for a swarm of 128 devices (Plonky2, laptop)

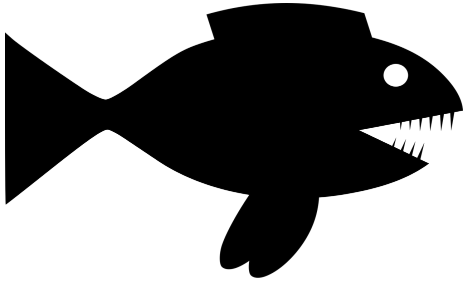
Proof size & verification time outperforms existing anonymous schemes:





Conclusion

- Transformation of any traditional RA scheme
    - ➢ Non-interactive, pub. verifiable, anonymous
- First anonymous swarm attestation scheme
    - ➢ No fixed hierarchy/topology, non-interactive, pub. verifiable
- Proofs of security (unforgeability & anonymity)
- Implementation & benchmarks



Thank you! Questions?

Paper:

Code:

Artifact Evaluated

NDSS SYMPOSIUM

Available

Functional

Reproduced

References

[ABI+15]: Asokan, Nadarajah, et al. "Seda: Scalable embedded device attestation." 2015.

[NDK+20]: Neureither, Jens, et al. "LegIoT: Ledgered trust management platform for IoT." 2020.

[PYD+22] Dushku, Edlira, et al. "PROVE: Provable remote attestation for public verifiability." 2023.

[EHS+25]: El Kassem, Nada, et al. "PRIVE: Towards Privacy-Preserving Swarm Attestation.” 2025.

[HKR+25]: Hellemans, Wouter, et al. "SPARK: Secure Privacy-Preserving Anonymous Swarm Attestation for In-Vehicle Networks." 2025