Formal Verification of Secure Forwarding Protocols

Tobias Klenze, Christoph Sprenger, David Basin

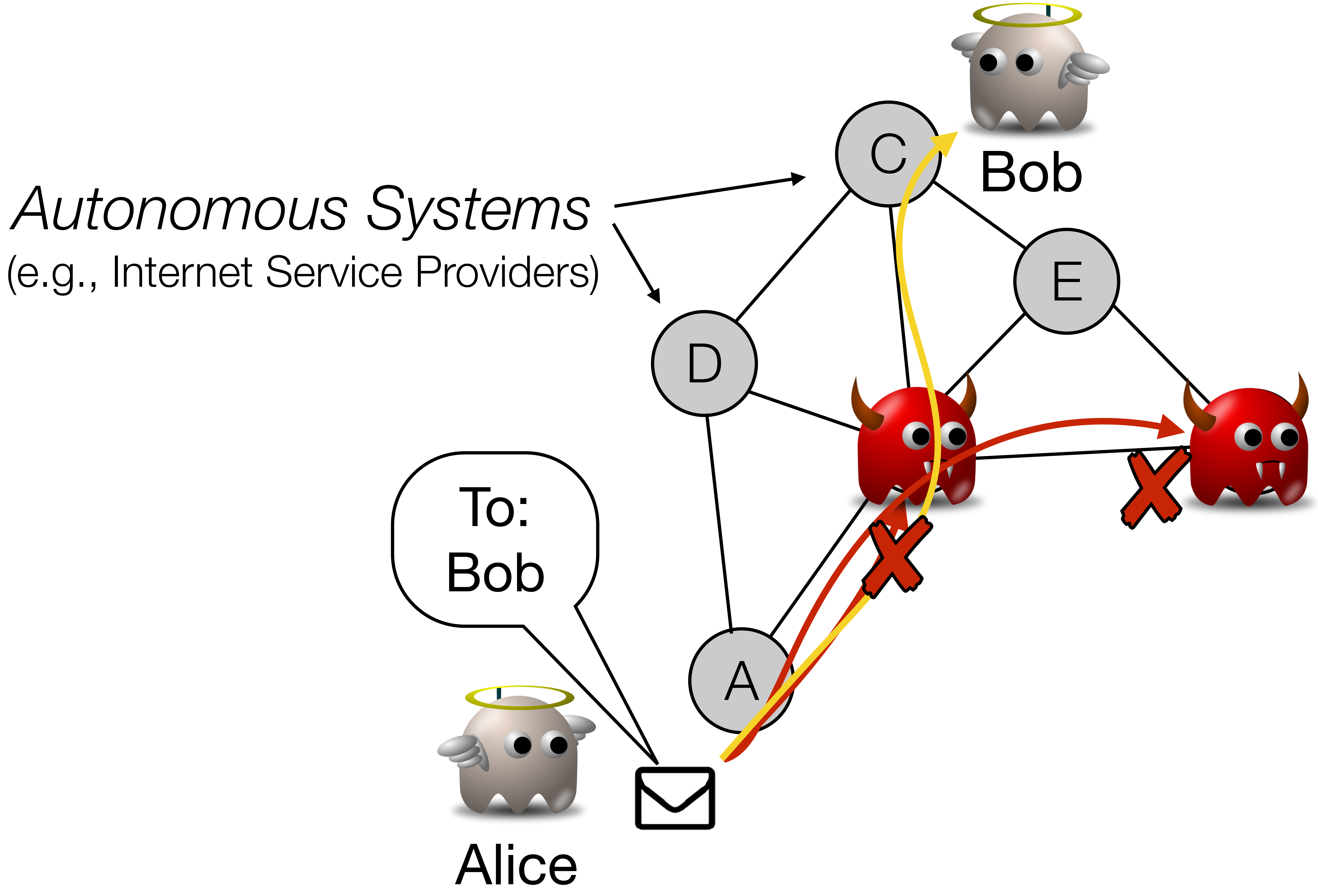
CSF'21
June 2021

ETH zürich

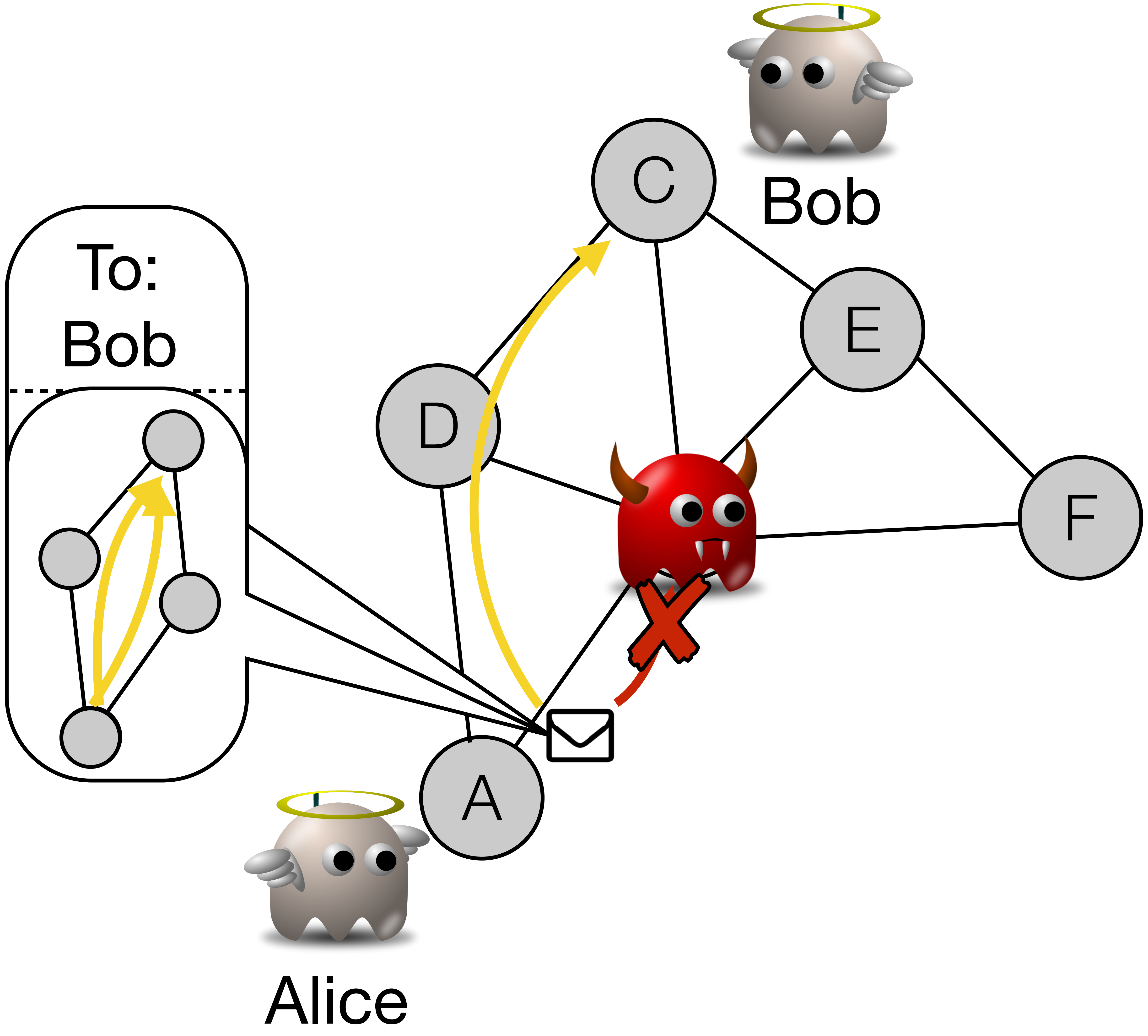
Proofs available!

Contact: tobias.klenze@inf.ethz.ch

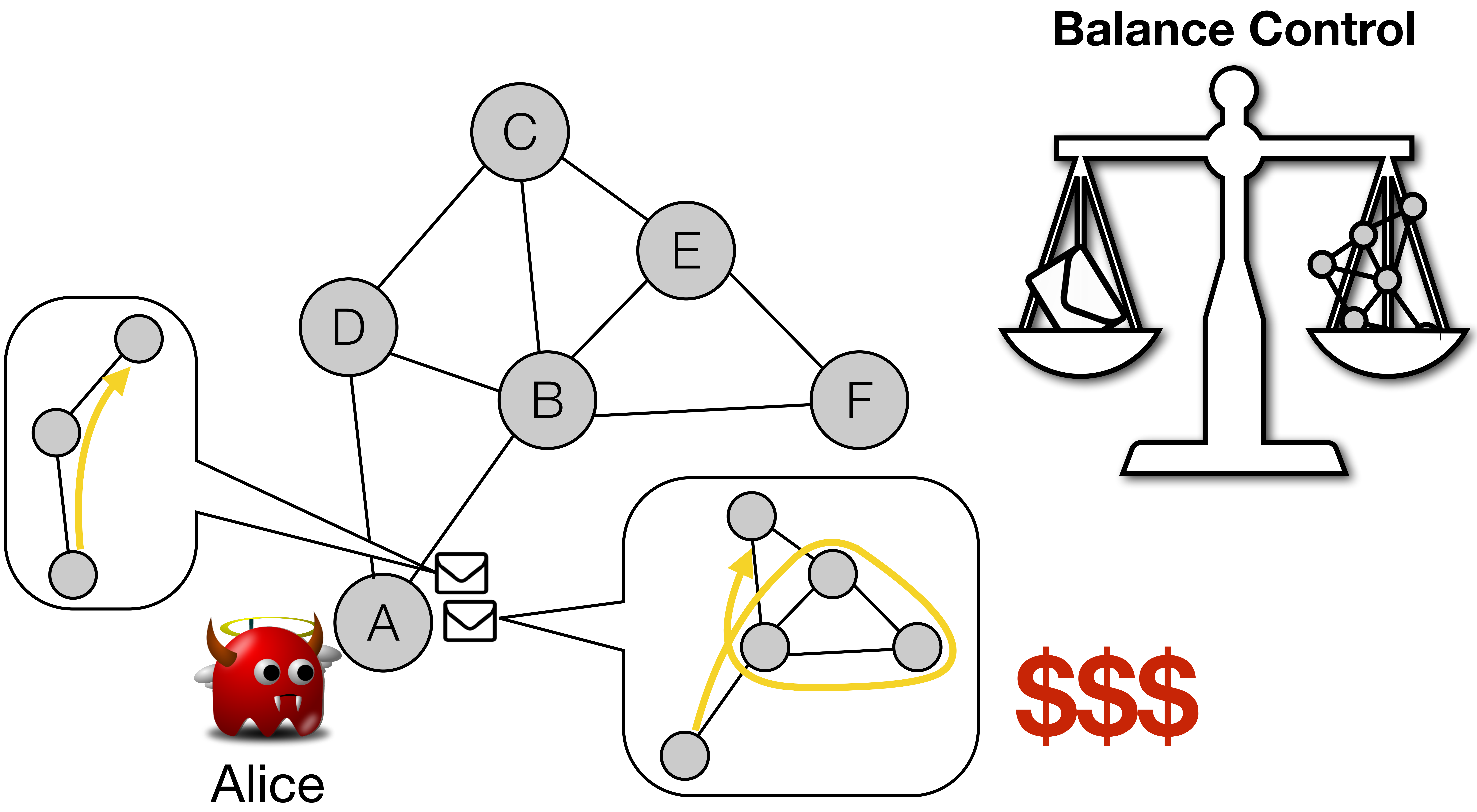
The Internet lacks network security



Path-aware Internet

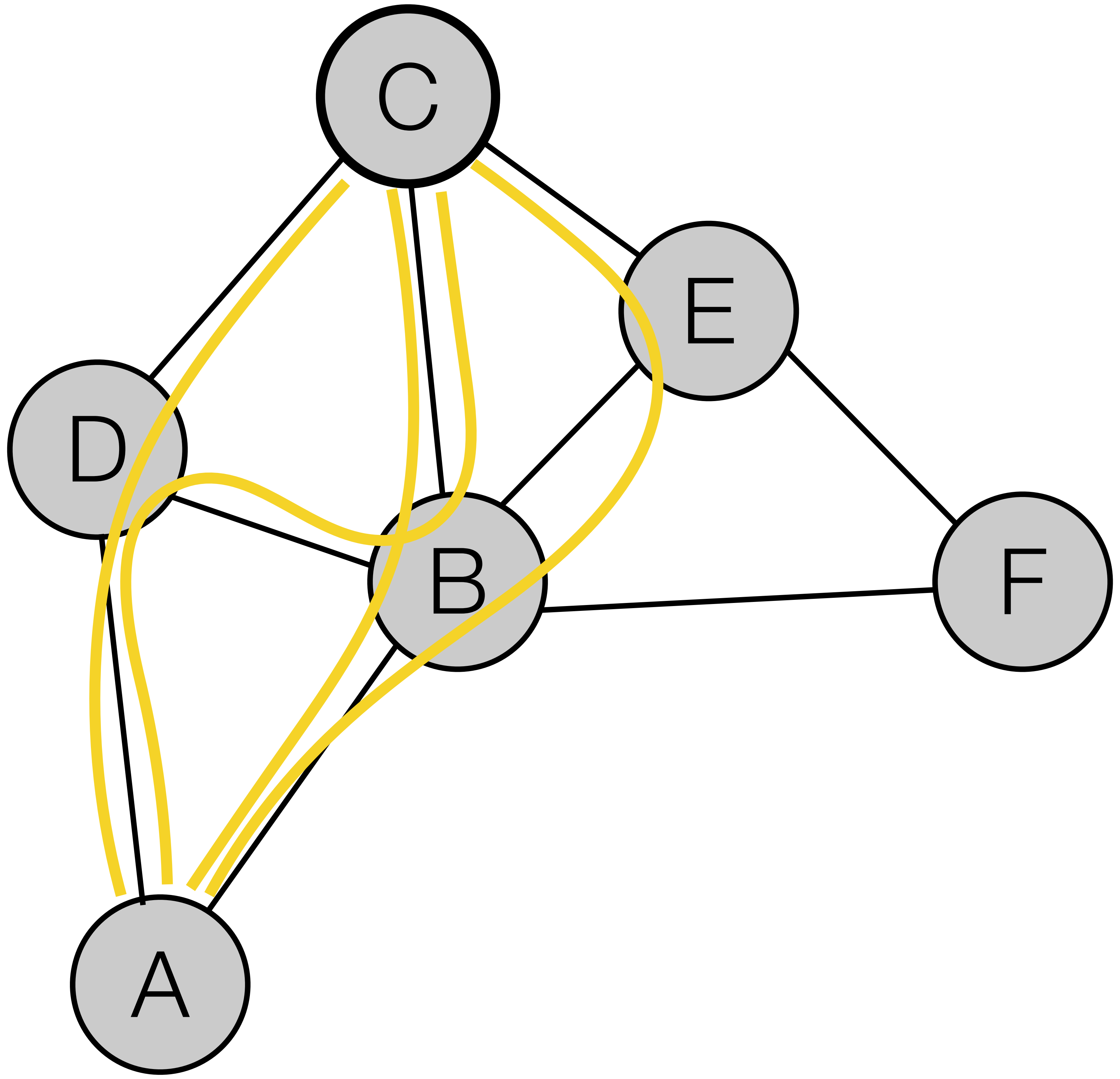


Path-aware Internet



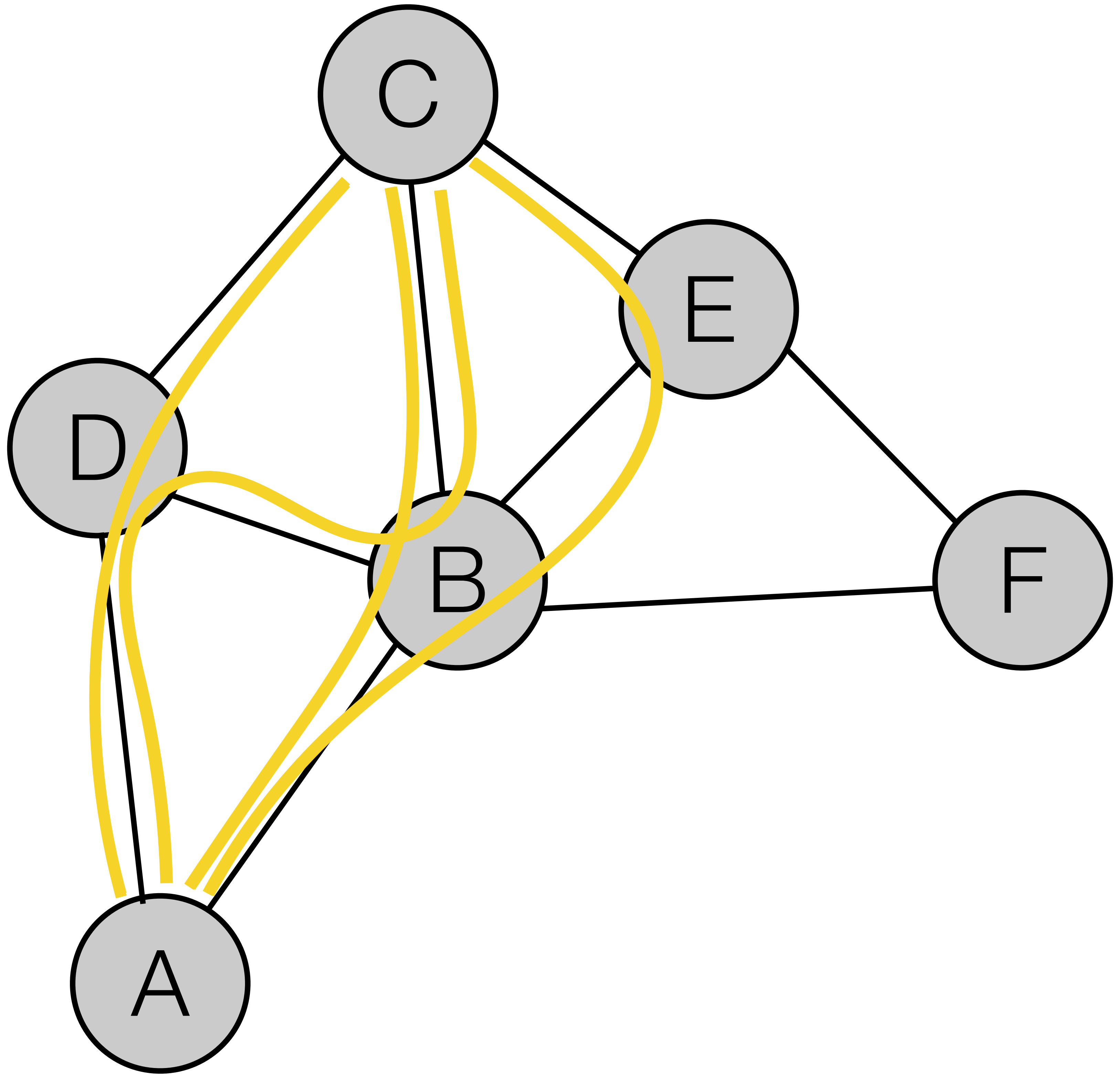
Path-aware Internet

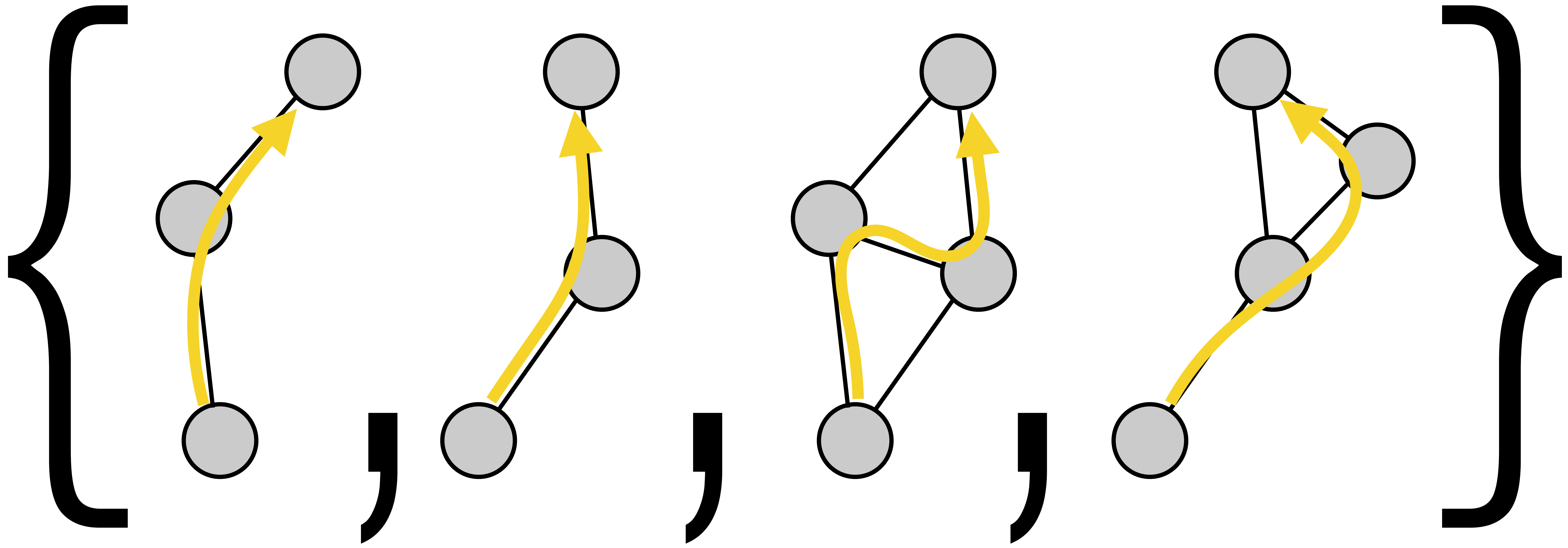
Two parts: 1. **Routing** *(creating & authorising paths), …*



Path-aware Internet

Two parts: (1.) **Routing** *(creating & authorising paths), …*

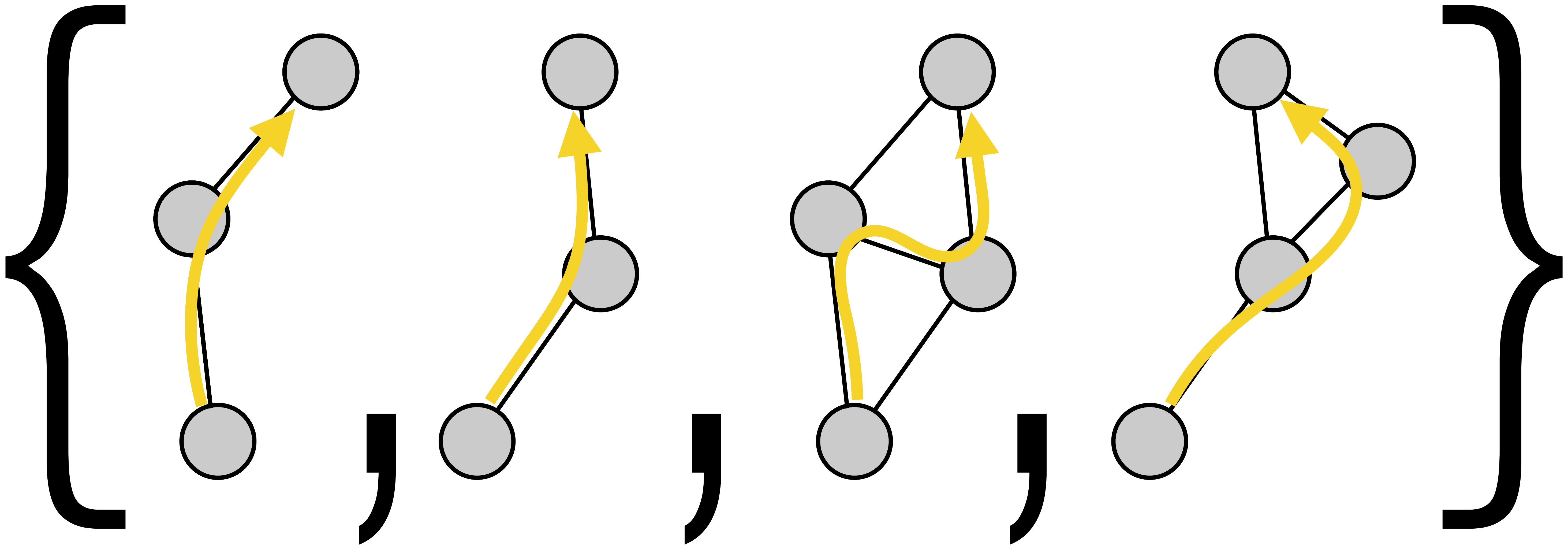


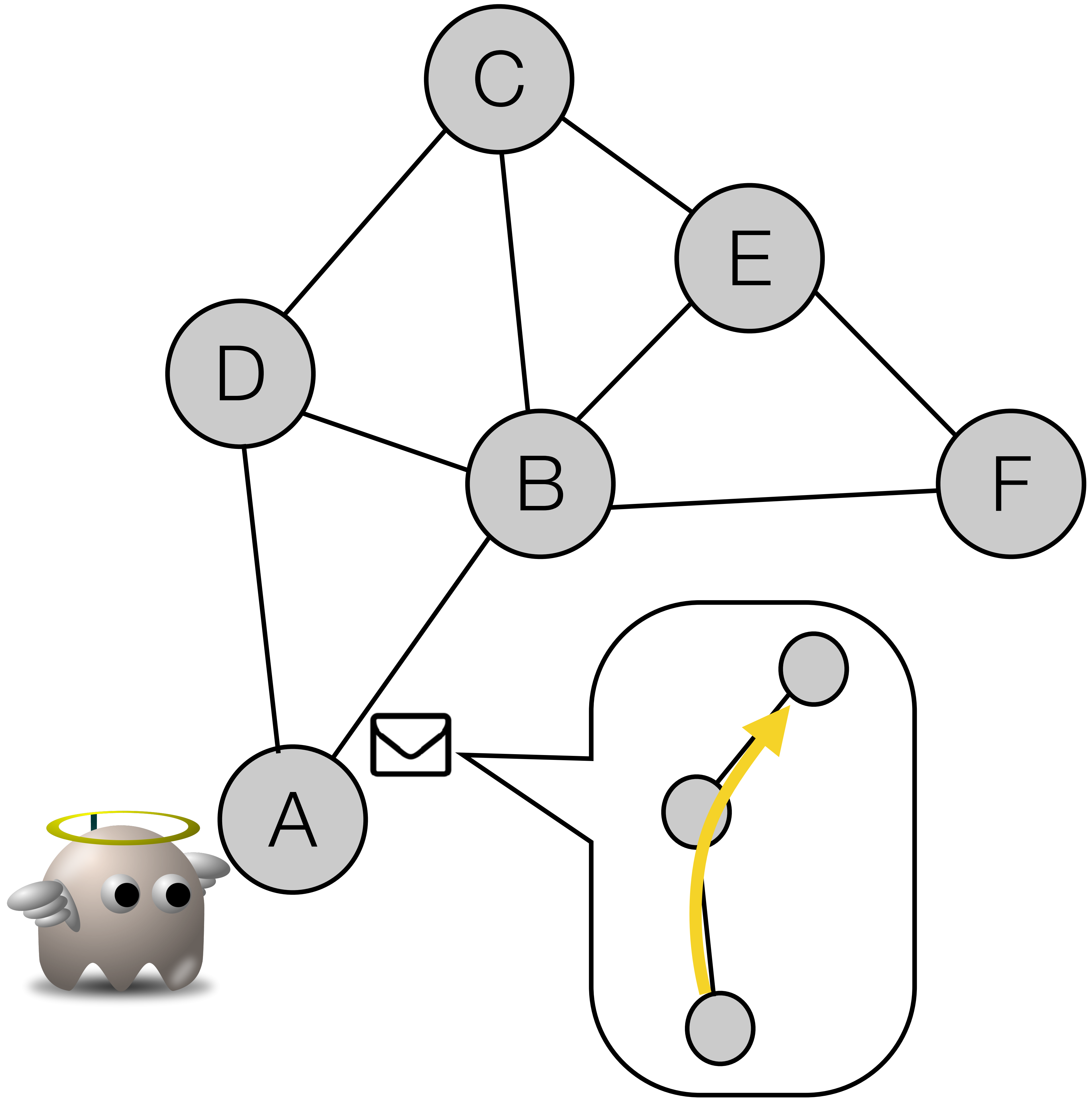


Set of authorized paths

Path-aware Internet

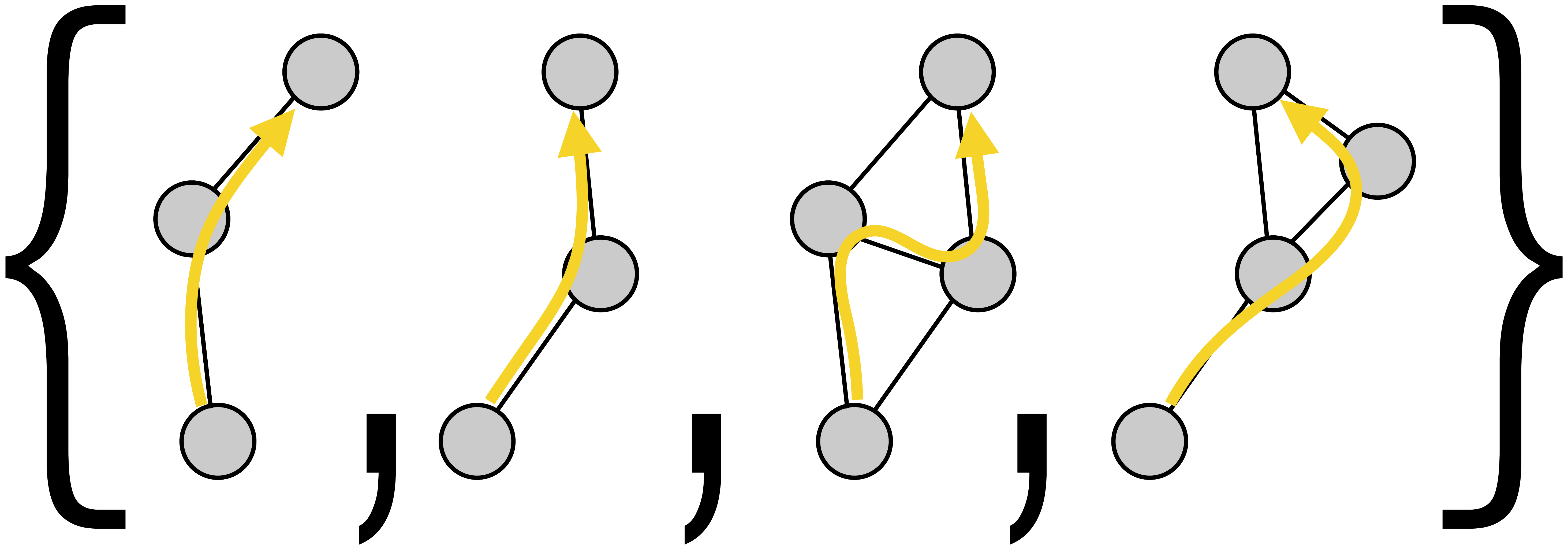
Two parts: **1.** **Routing** *(creating & authorising paths),* **2.** **Forwarding** *(using paths)*

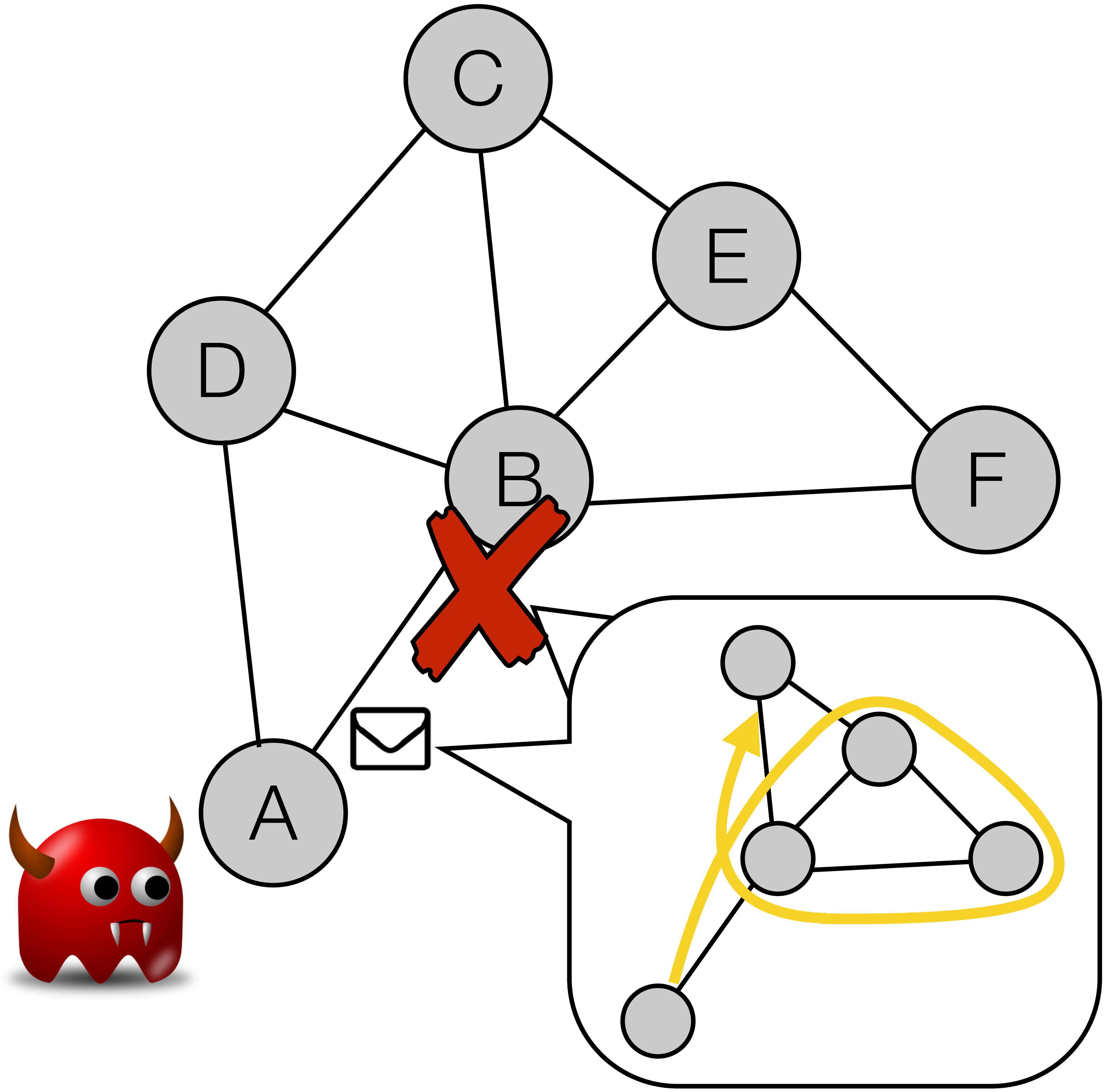




Path-aware Internet

Two parts: 1. **Routing** *(creating & authorising paths)*, 2. **Forwarding** *(using paths)*



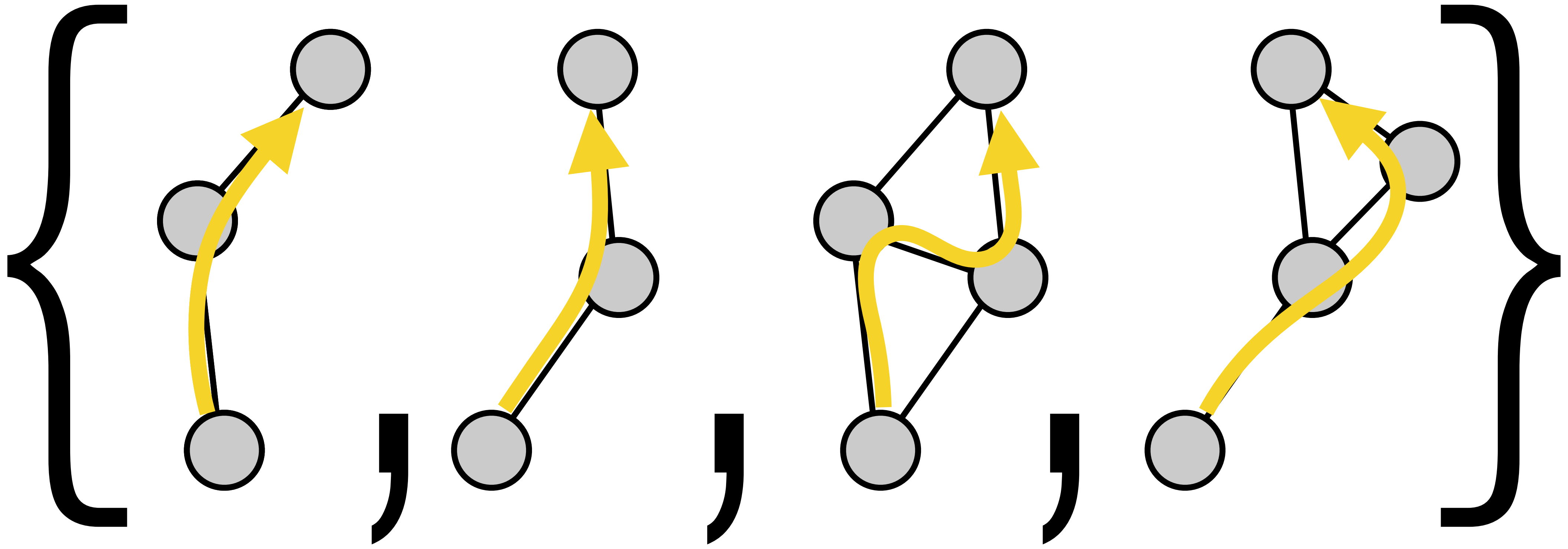


Path Authorization:

Packets traverse the network only along authorized paths.

Challenges for the Verification of Path Authorization

Challenge #1

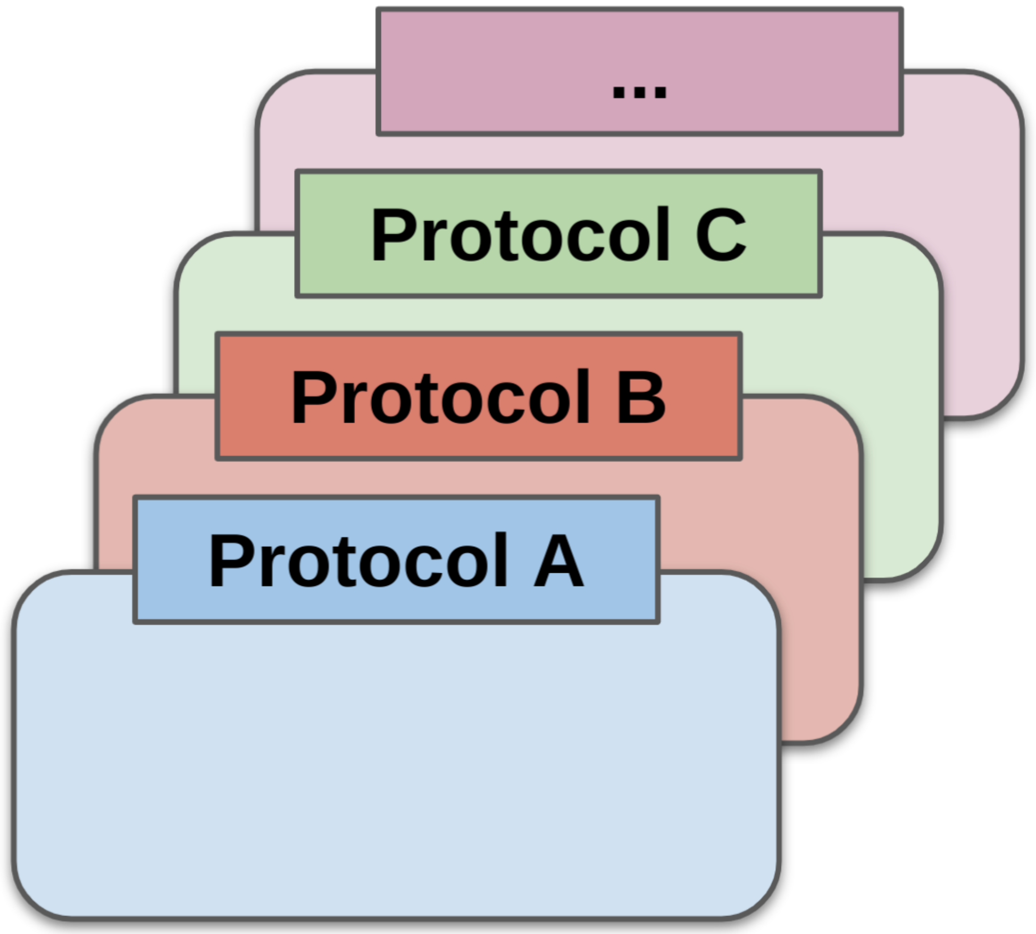


Arbitrary, **unbounded** set of authorized paths, and unbounded path length.

Challenge #2

Expressiveness to formulate path authorization.

Challenge #3



Large class of protocols.

Verification of Path Authorization

Our approach: Refinement in Isabelle/HOL.

Environment parameter

Arbitrary, **unbou**
of authorized pa
unbounded pat

Abstract

prove path authorization

refine

Concrete

prove refinement

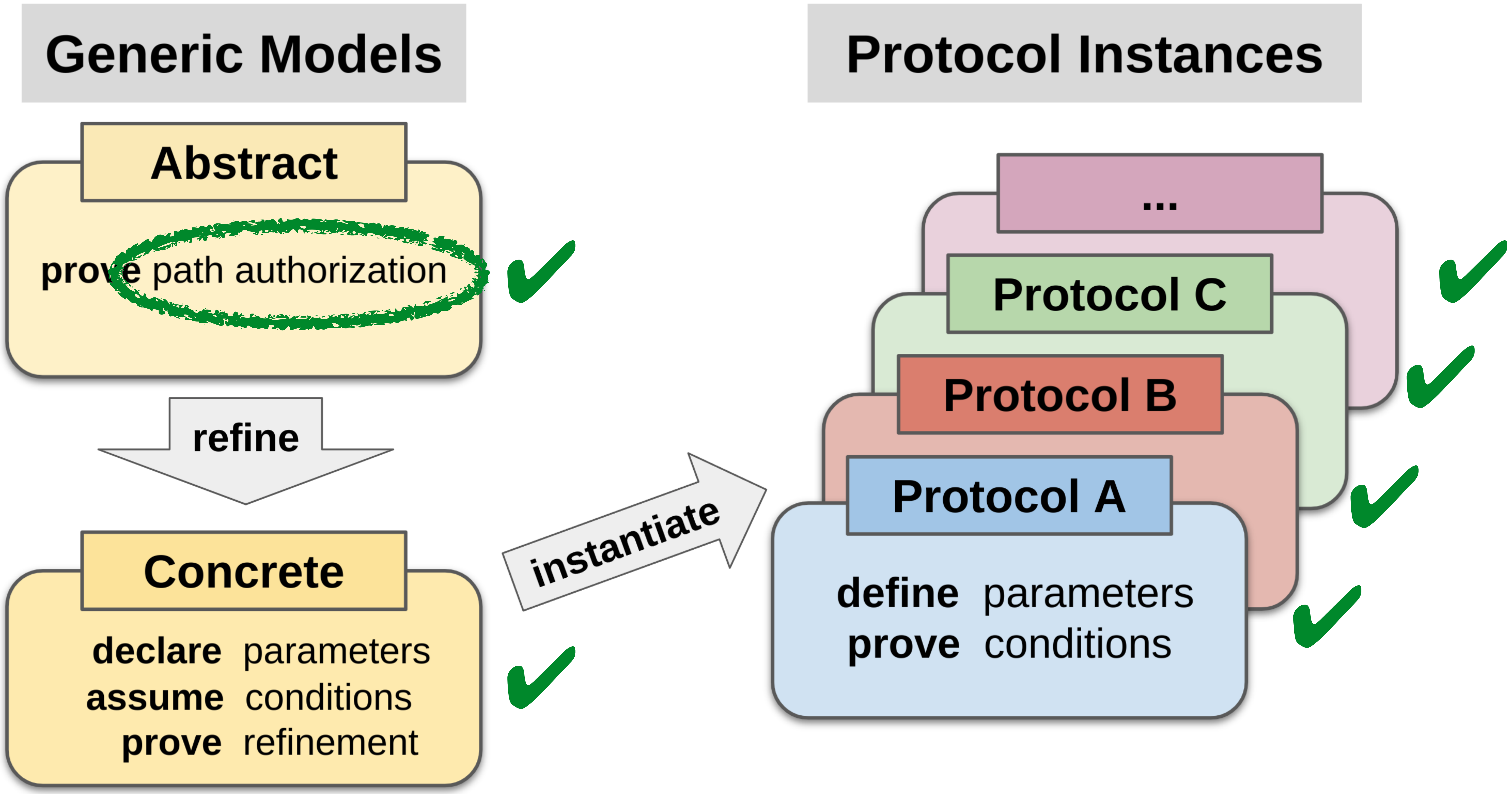
No attacker

No authenticators

Distributed, colluding Dolev-Yao attacker

Cryptographic authenticators

Parametrized Verification Framework



Property preservation

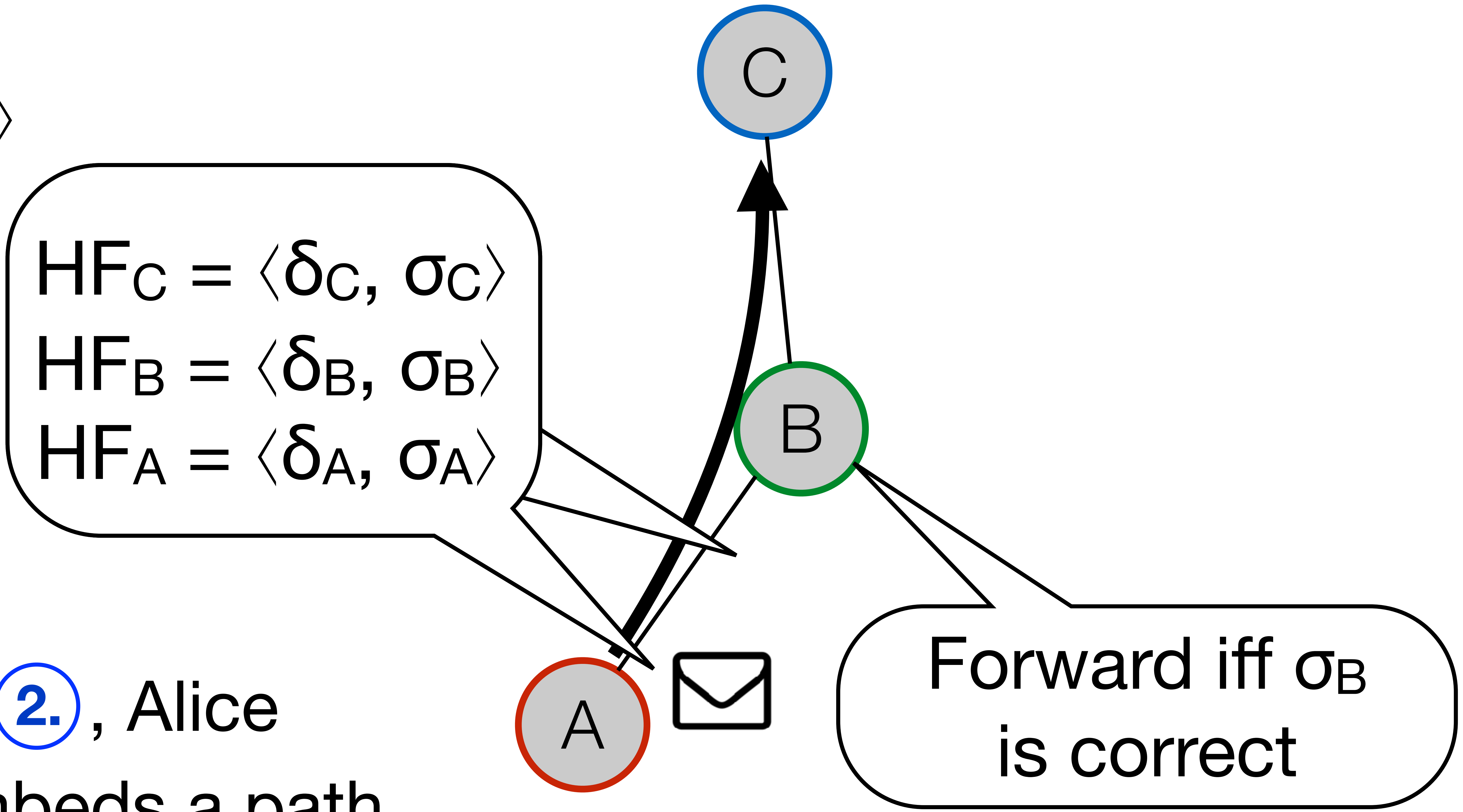
Contributions:

- Proving security of a class of forwarding protocols
- Insights into protocol class
- Low-effort proofs: Eight instances, only static reasoning, not about transitions

Modelling Forwarding

In (1.), paths are created:
one Hop Field $HF_i = \langle \delta_i, \sigma_i \rangle$
per node i.

- δ_i: local forwarding information
- σ_i: authenticator (e.g., MAC)

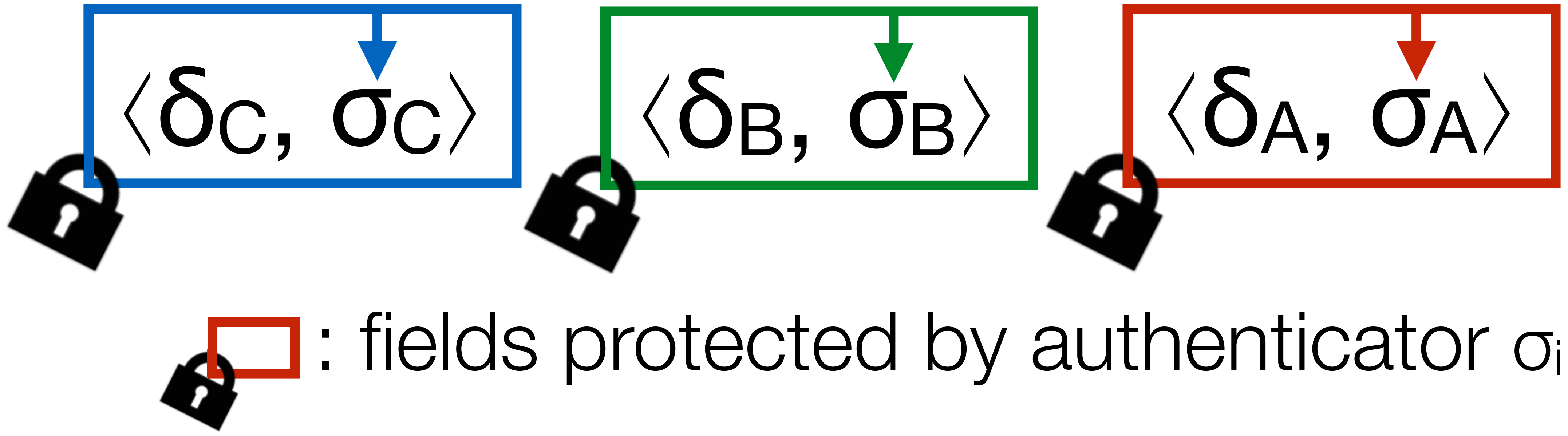


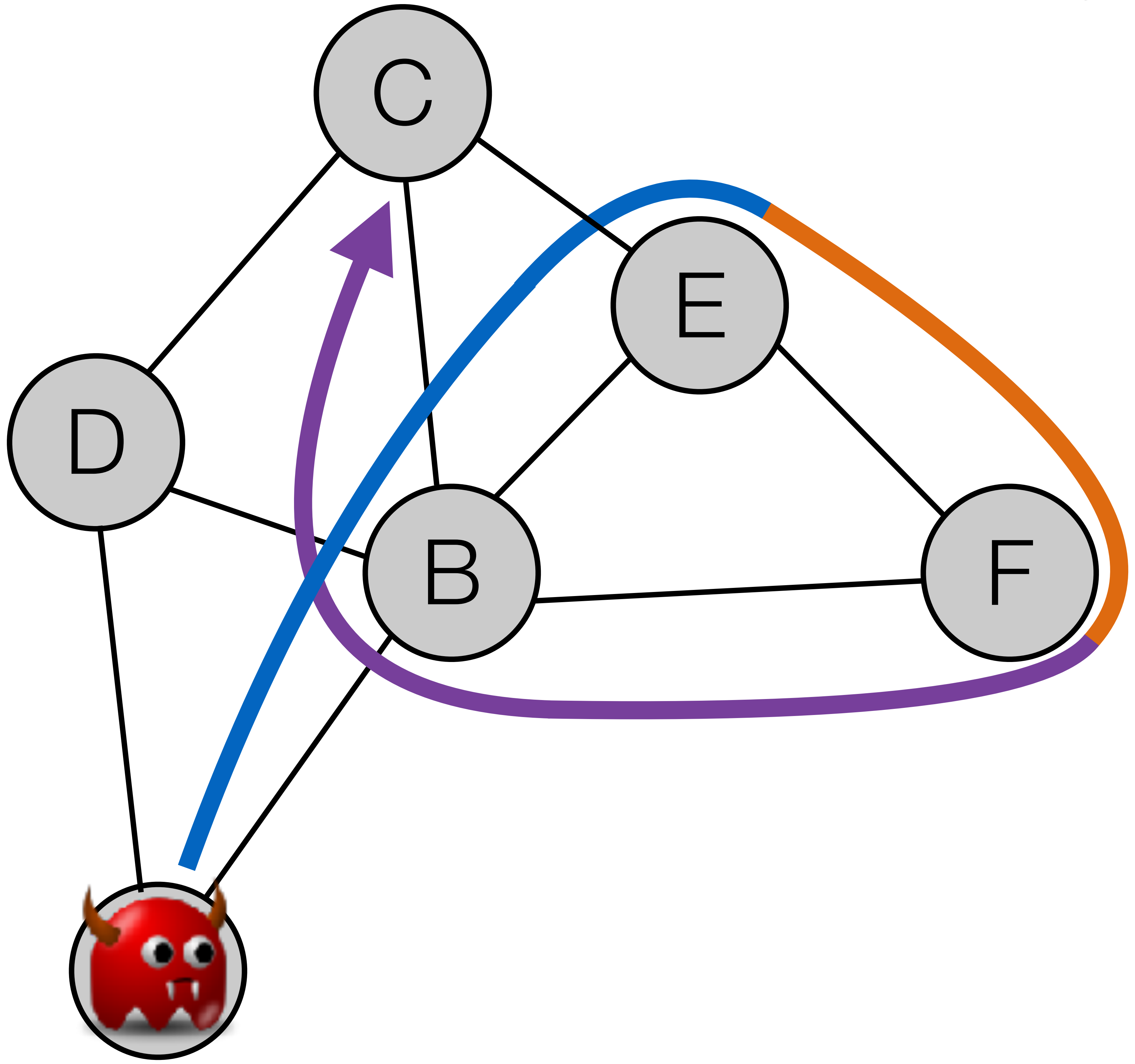
In (2.), Alice embeds a path.

In (2.), routers check validity of authenticator.

How to define the authenticator?

$$\sigma_i = MAC_{Key(i)} \langle \delta_i \rangle$$

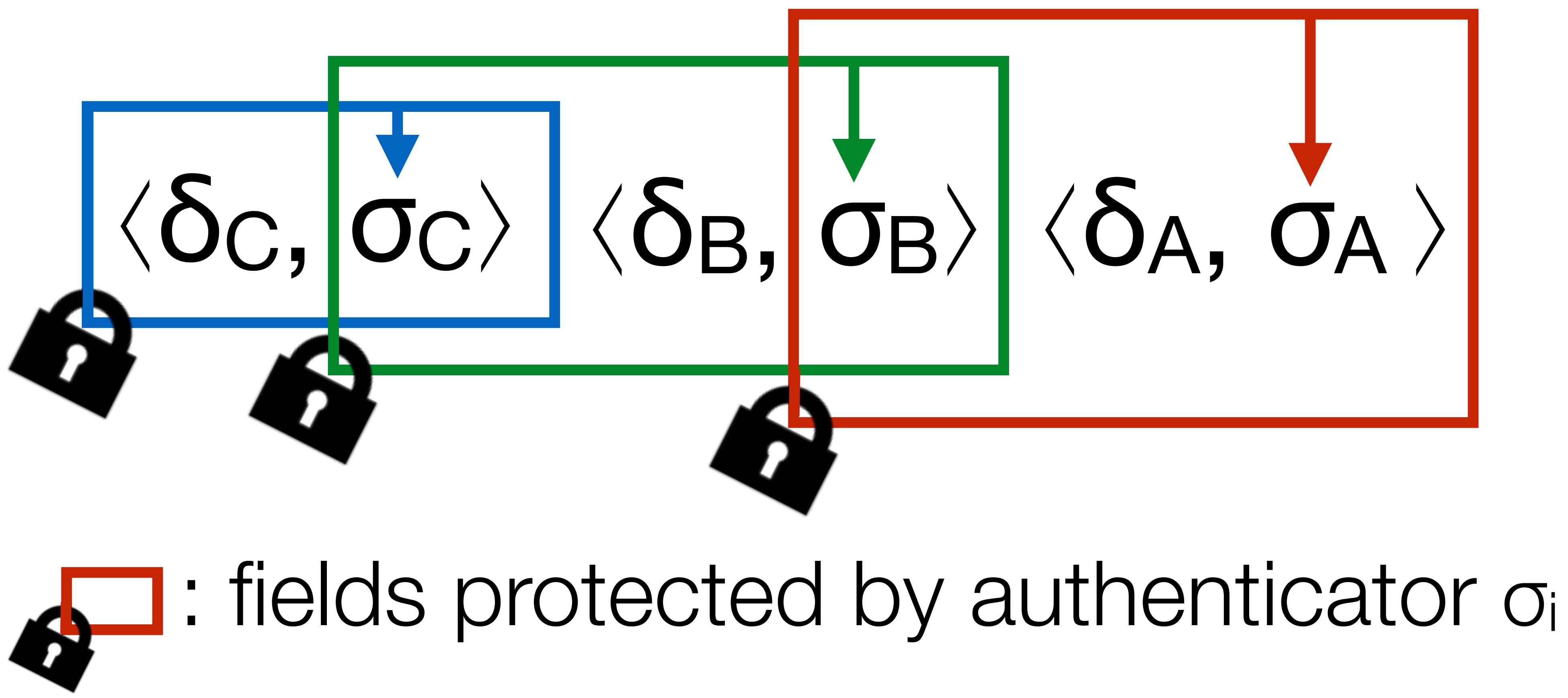




Authenticating local δ is not enough!

Authenticators must protect subsequent path

$$\sigma_i = \mathrm{MAC}_{Key(i)} \langle \delta_i, \underbrace{\sigma_{i+1}}_{\perp \text{ for last hop field}} \rangle$$



$$\sigma_A = \mathrm{MAC}_{Key(A)} \langle \delta_A, \sigma_B \rangle$$
$$\sigma_A = \mathrm{MAC}_{Key(A)} \langle \delta_A, \mathrm{MAC}_{Key(B)} \langle \delta_B, \sigma_C \rangle\rangle$$
$$\sigma_A = \mathrm{MAC}_{Key(A)} \langle \delta_A, \mathrm{MAC}_{Key(B)} \langle \delta_B, \mathrm{MAC}_{Key(C)} \langle \delta_C, \perp \rangle\rangle\rangle$$

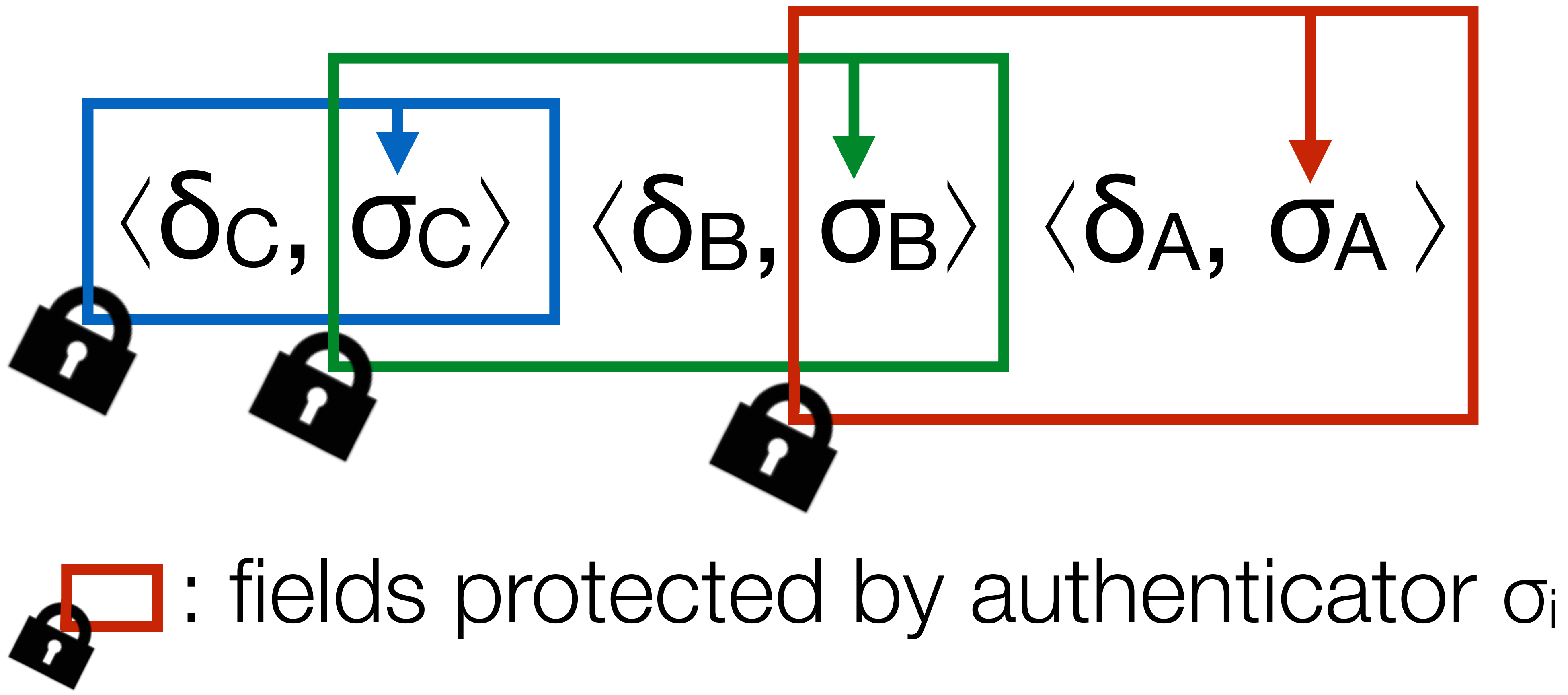
$$\mathrm{extract}(\sigma_A) = [\delta_A, \delta_B, \delta_C]$$

Authenticators must protect subsequent path

σ_i = **Cryptographic check**

⊥ for last hop field

Parameter



$$\sigma_A = MAC_{Key(A)} \langle \delta_A, MAC_{Key(B)} \langle \delta_B, MAC_{Key(C)} \langle \delta_C, \perp \rangle\rangle\rangle$$

extract(σ_A) = $[\delta_A, \delta_B, \delta_C]$

Parameter

Parametrized Concrete Model

- Three **protocol parameters**
- Five **static conditions**

Conclusion

Three verification challenges:

Arbitrary, **unbounded** sets of authorized paths	**Expressiveness** for path authorization	**Low effort proofs** for new protocol variants

We solved these challenges via **refinement** and **parametrization** in **Isabelle/HOL**

Future work: Whole Internet architectures to verify!

Contact: tobias.klenze@inf.ethz.ch