University of Stuttgart
Institute of
Information Security

Universal Composability: A Comparison of Different Models

Daniel Rausch

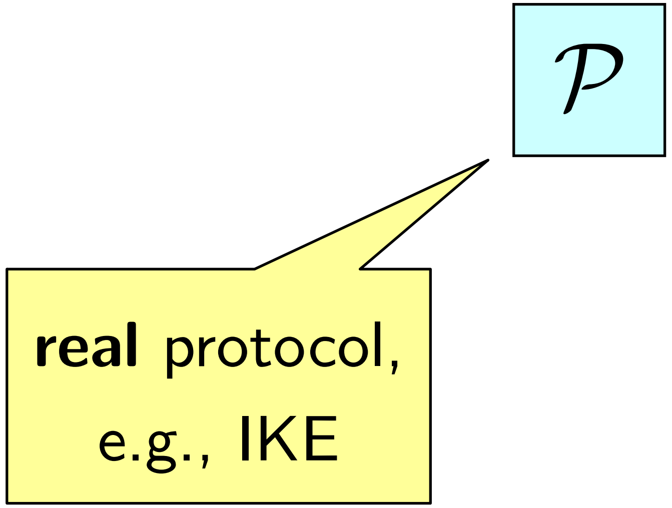
FOSAD 2018

Simulation-Based Security

Definition of simulatability (basic idea):

Simulation-Based Security

Definition of simulatability (basic idea):

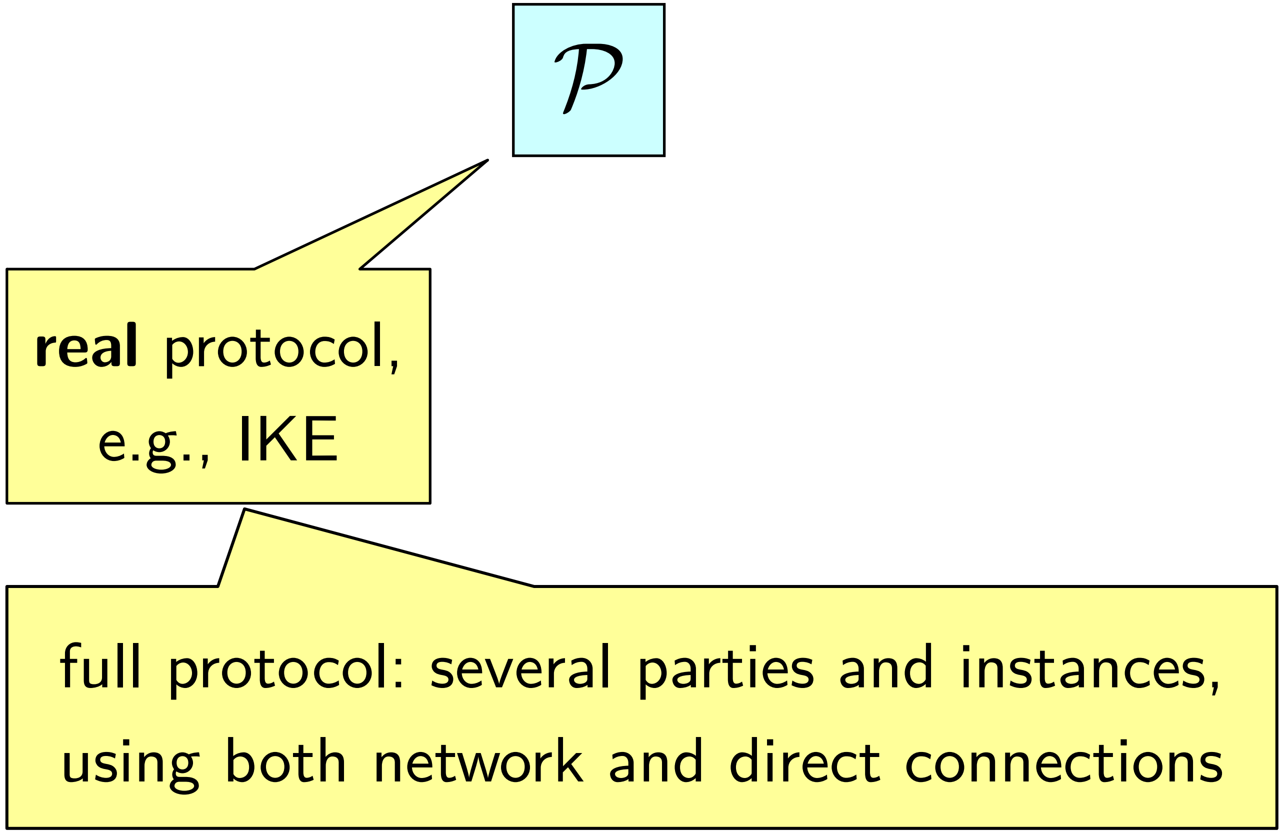


$\mathcal{P}$

real protocol, e.g., IKE

Simulation-Based Security

Definition of simulatability (basic idea):



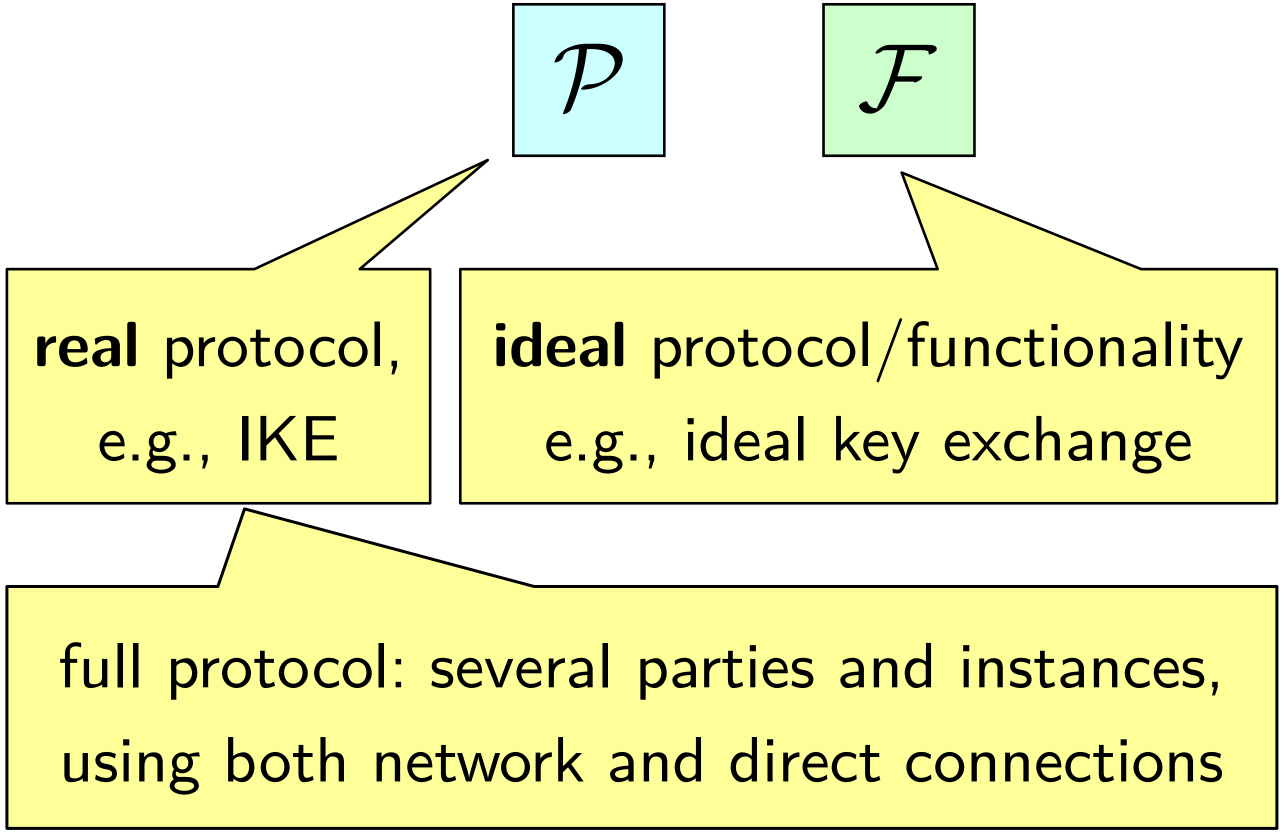
$\mathcal{P}$

real protocol, e.g., IKE

full protocol: several parties and instances, using both network and direct connections

Simulation-Based Security

Definition of simulatability (basic idea):



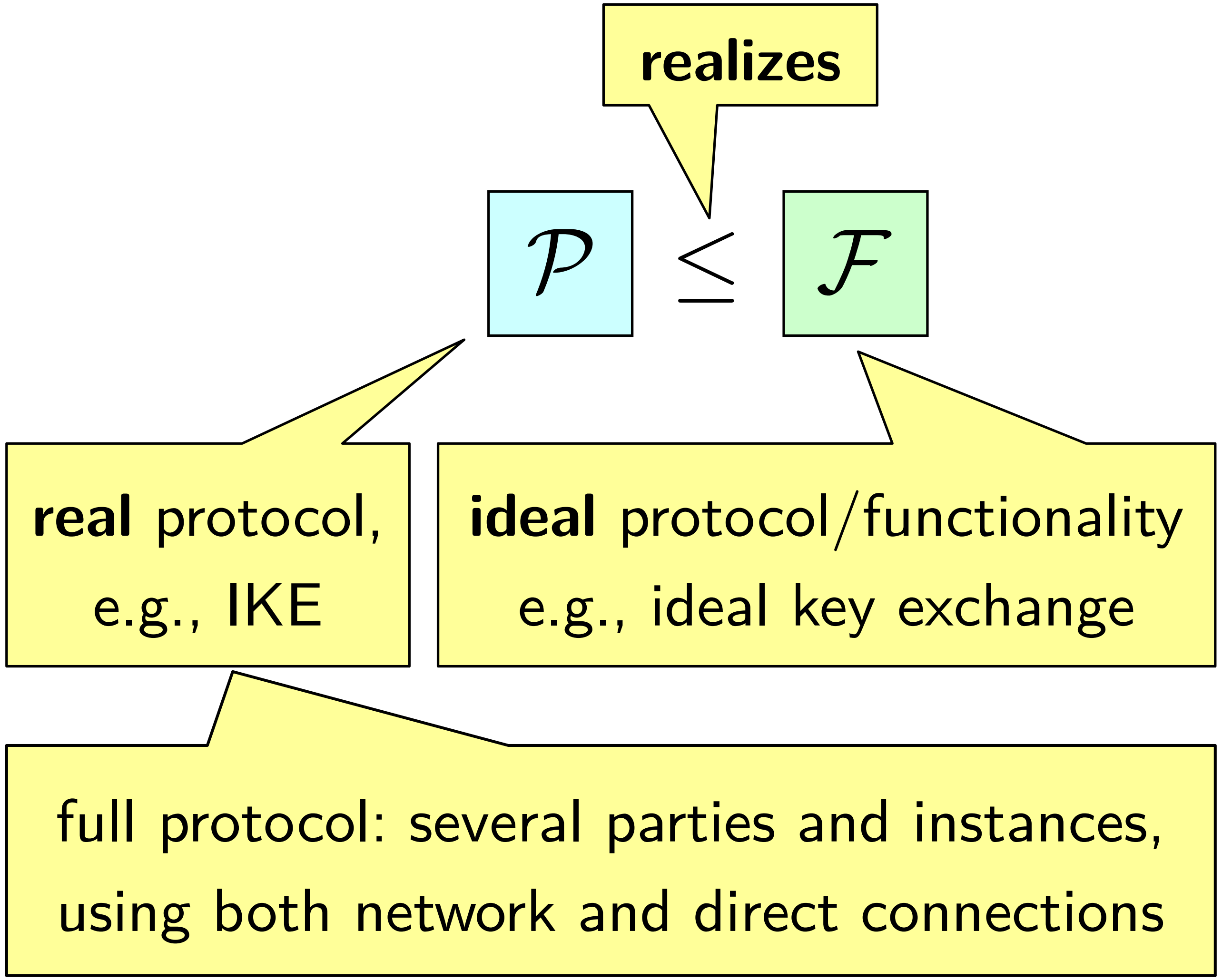
real protocol, e.g., IKE

ideal protocol/functionality e.g., ideal key exchange

full protocol: several parties and instances, using both network and direct connections

Simulation-Based Security

Definition of simulatability (basic idea):



realizes

$\mathcal{P} \leq \mathcal{F}$

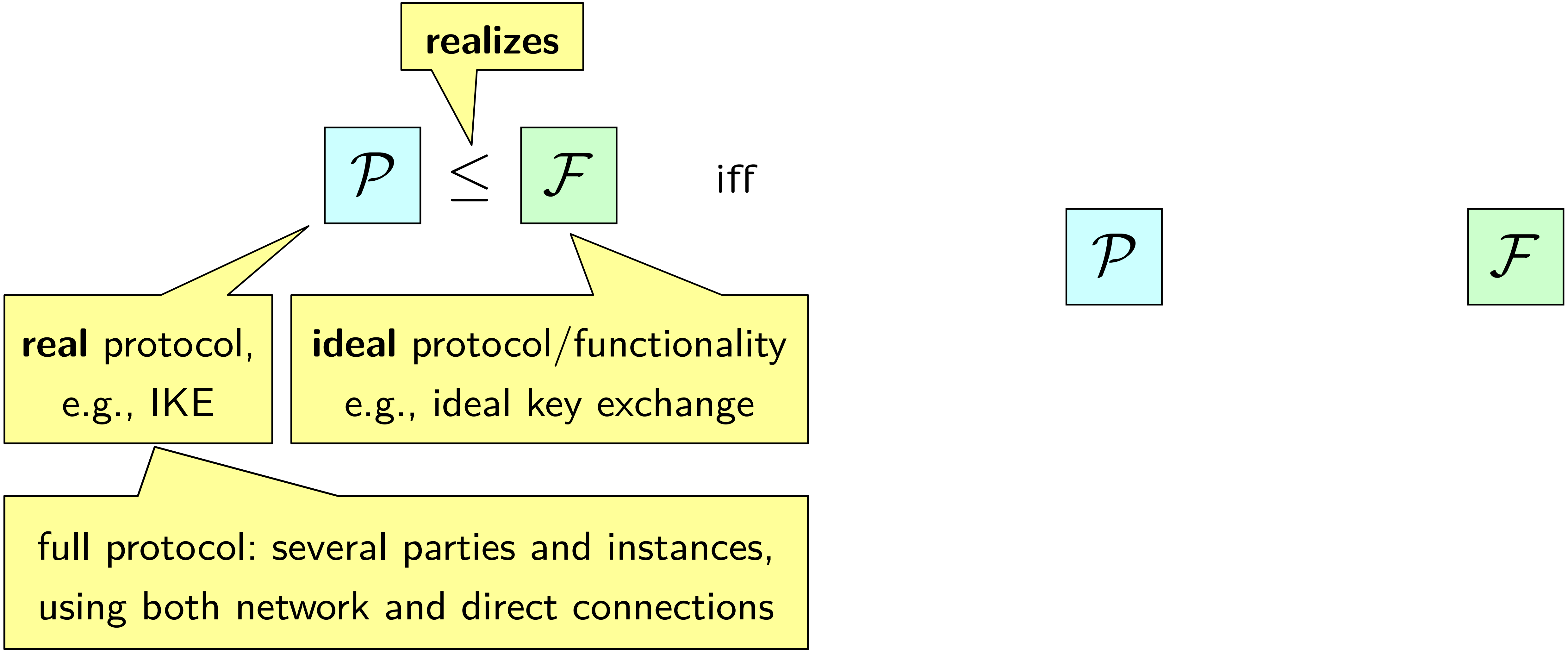
real protocol,
e.g., IKE

ideal protocol/functionality
e.g., ideal key exchange

full protocol: several parties and instances,
using both network and direct connections

Simulation-Based Security

Definition of simulatability (basic idea):



realizes

$\mathcal{P} \leq \mathcal{F}$ iff

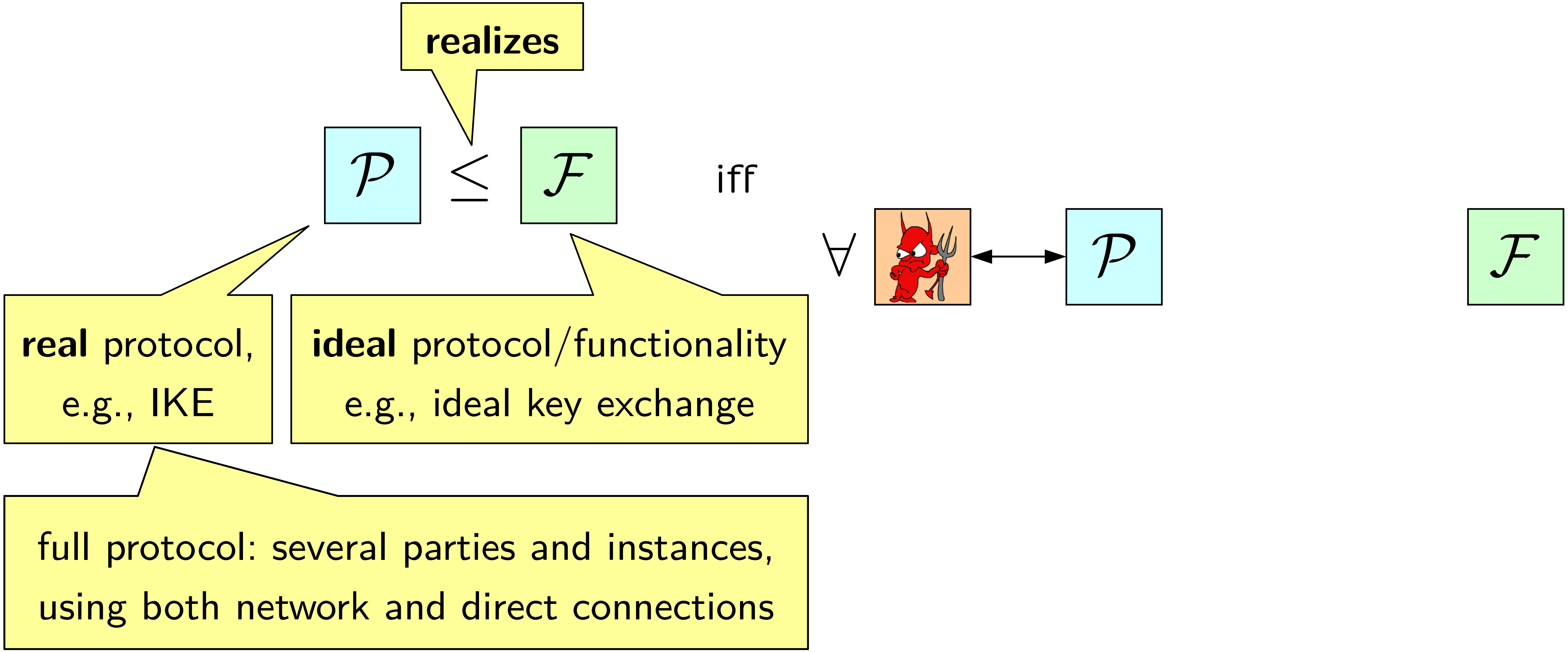
real protocol, e.g., IKE

ideal protocol/functionality e.g., ideal key exchange

full protocol: several parties and instances, using both network and direct connections

Simulation-Based Security

Definition of simulatability (basic idea):



realizes

$\mathcal{P} \leq \mathcal{F}$ iff

$\forall$ $\longleftrightarrow \mathcal{P}$ $\mathcal{F}$

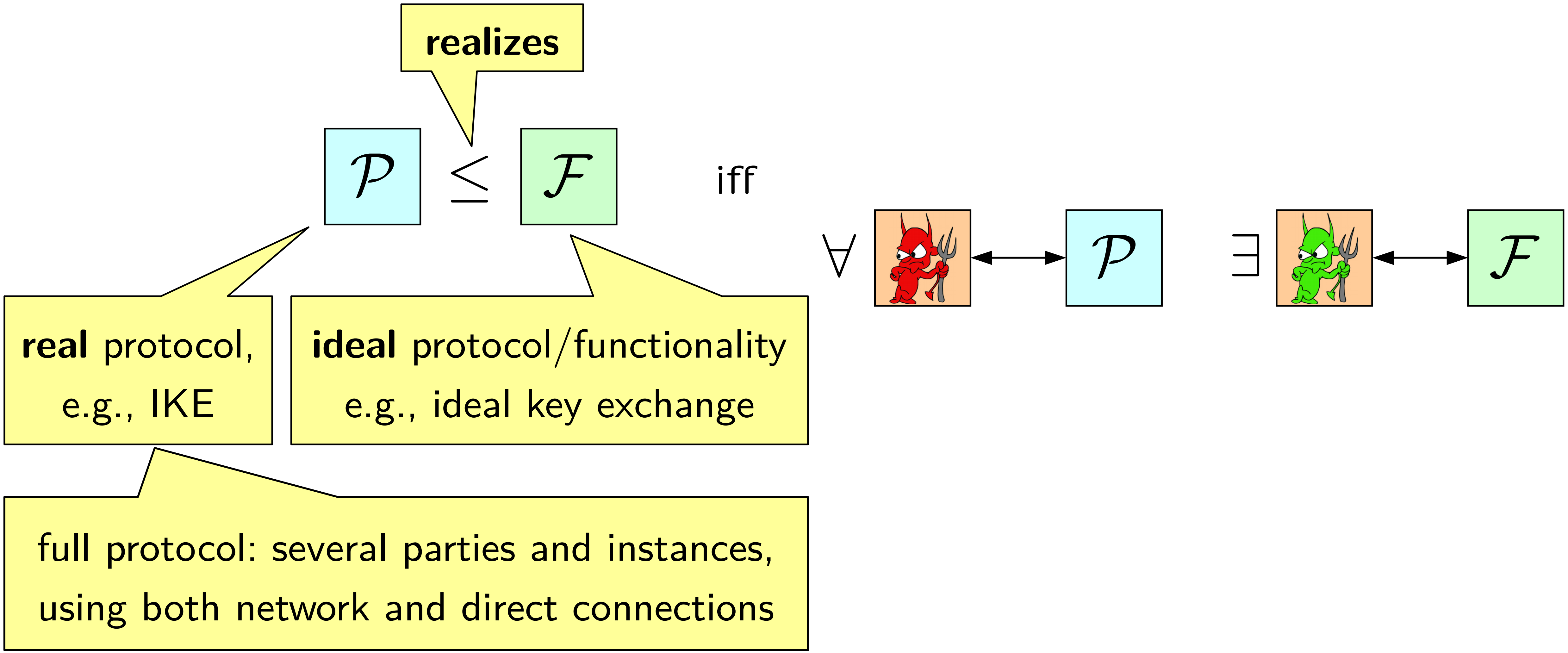
real protocol, e.g., IKE

ideal protocol/functionality e.g., ideal key exchange

full protocol: several parties and instances, using both network and direct connections

Simulation-Based Security

Definition of simulatability (basic idea):



realizes

real protocol,
e.g., IKE

ideal protocol/functionality
e.g., ideal key exchange

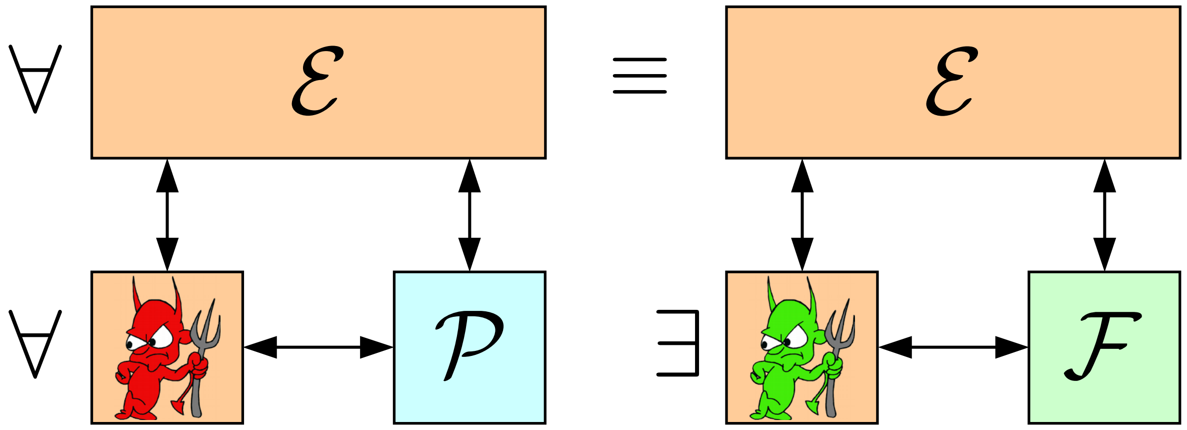
full protocol: several parties and instances,
using both network and direct connections

Simulation-Based Security

Definition of simulatability (basic idea):

realizes

$\mathcal{P} \leq \mathcal{F}$ iff



real protocol, e.g., IKE

ideal protocol/functionality e.g., ideal key exchange

full protocol: several parties and instances, using both network and direct connections

Compositional Protocol Analysis

Assume:

$$\boxed{\mathcal{P}} \leq \boxed{\mathcal{F}}$$

Compositional Protocol Analysis

Assume: $\boxed{\mathcal{P}} \leq \boxed{\mathcal{F}}$

e.g., ideal key exchange

Compositional Protocol Analysis

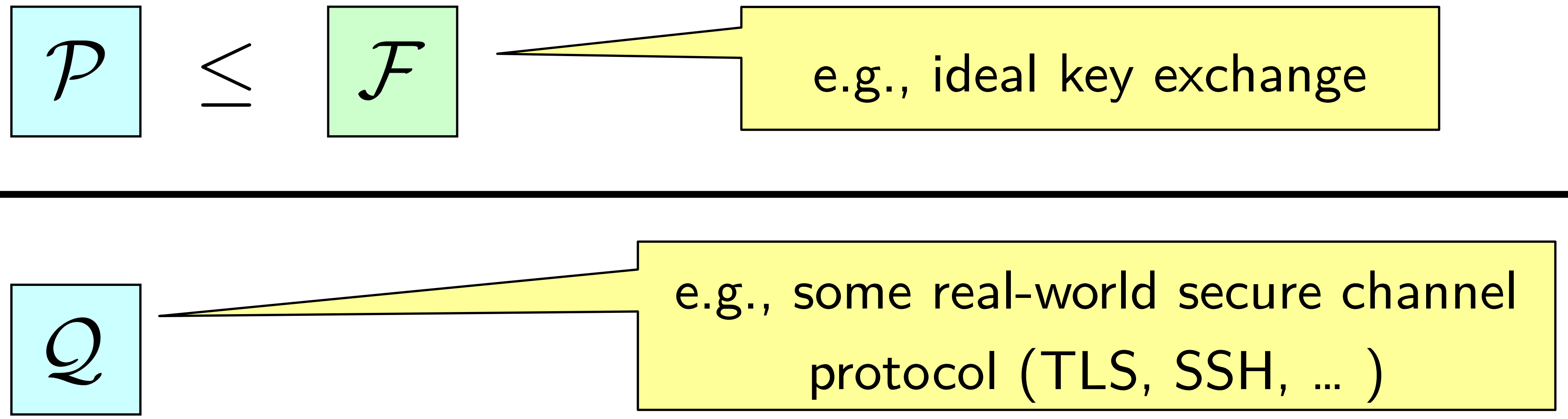
Assume: $\boxed{\mathcal{P}} \leq \boxed{\mathcal{F}}$

e.g., ideal key exchange

Prove: $\boxed{\mathcal{Q}}$

Compositional Protocol Analysis

Assume:



$\mathcal{P} \leq \mathcal{F}$

e.g., ideal key exchange

Prove:

$\mathcal{Q}$

e.g., some real-world secure channel protocol (TLS, SSH, …)

Compositional Protocol Analysis

Assume:

$$\boxed{\mathcal{P}} \leq \boxed{\mathcal{F}}$$

e.g., ideal key exchange

Prove:

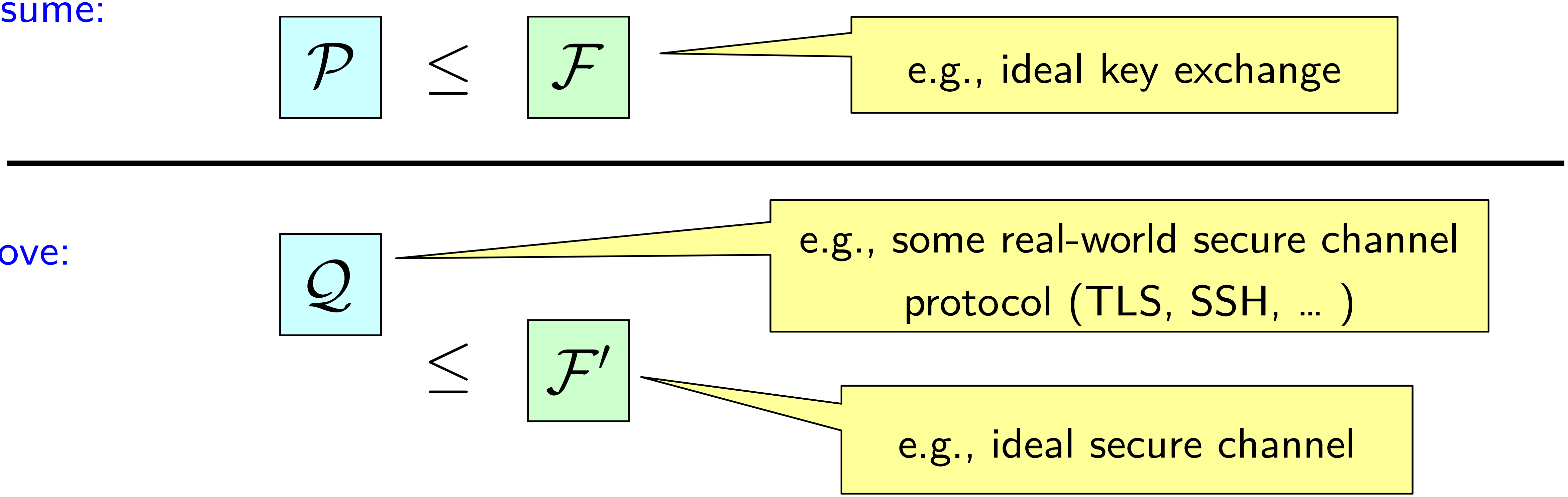
$$\boxed{\mathcal{Q}} \leq \boxed{\mathcal{F}'}$$

e.g., some real-world secure channel protocol (TLS, SSH, …)

Compositional Protocol Analysis

Assume:

Prove:



$\mathcal{P} \leq \mathcal{F}$

e.g., ideal key exchange

$\mathcal{Q} \leq \mathcal{F}'$

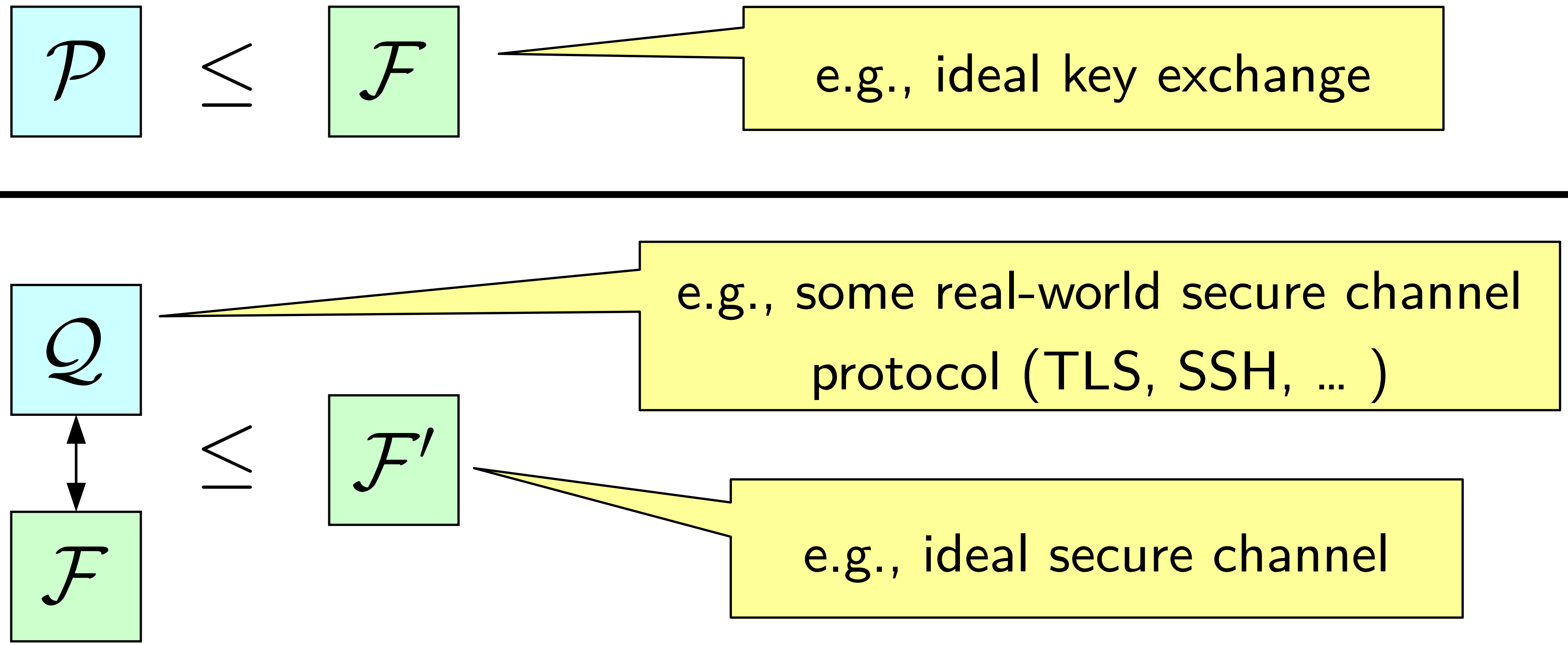
e.g., some real-world secure channel protocol (TLS, SSH, …)

e.g., ideal secure channel

Compositional Protocol Analysis

Assume:

Prove:

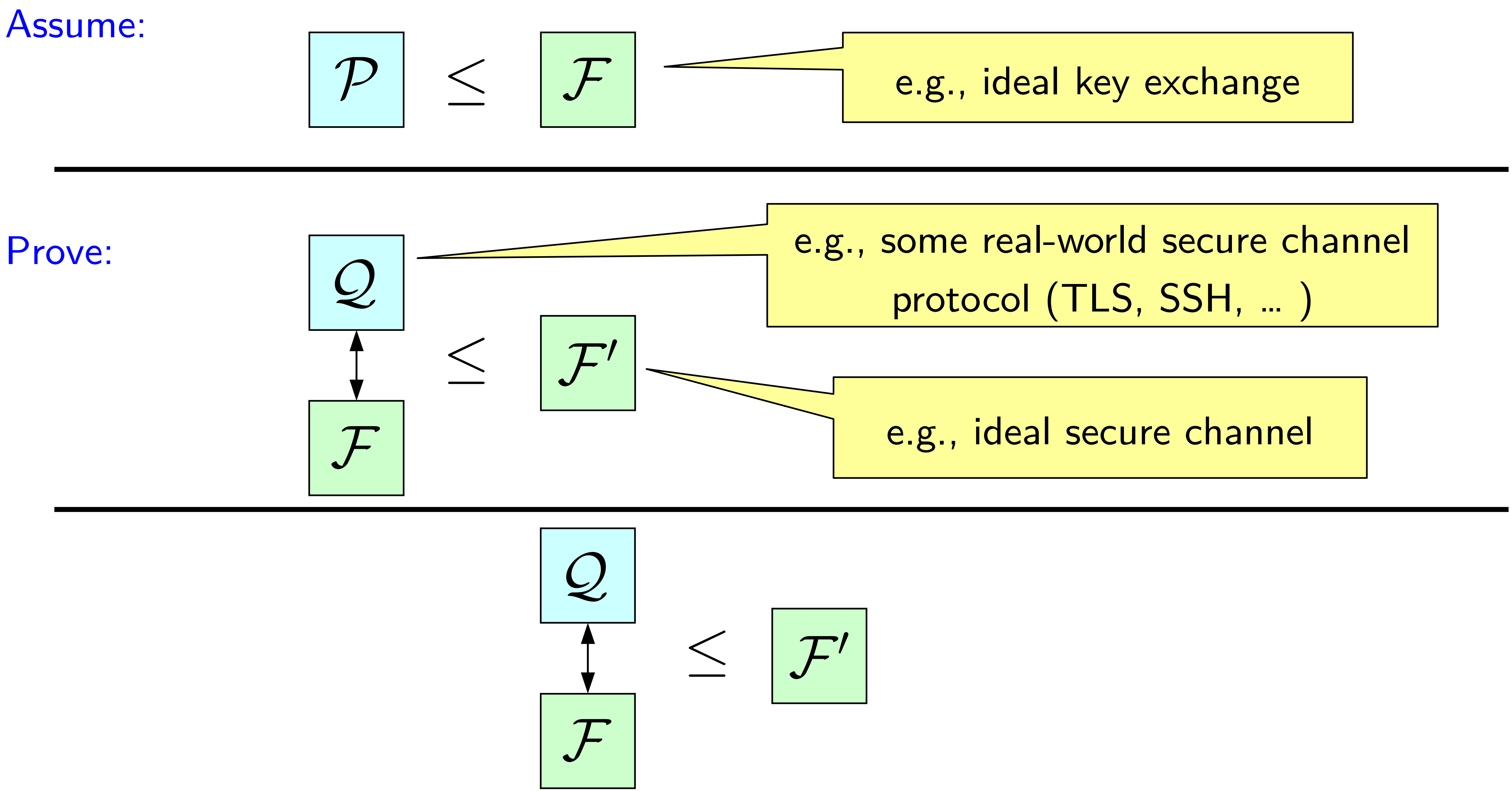


e.g., ideal key exchange

e.g., some real-world secure channel protocol (TLS, SSH, …)

e.g., ideal secure channel

Compositional Protocol Analysis



Assume:

$\mathcal{P} \leq \mathcal{F}$

e.g., ideal key exchange

Prove:

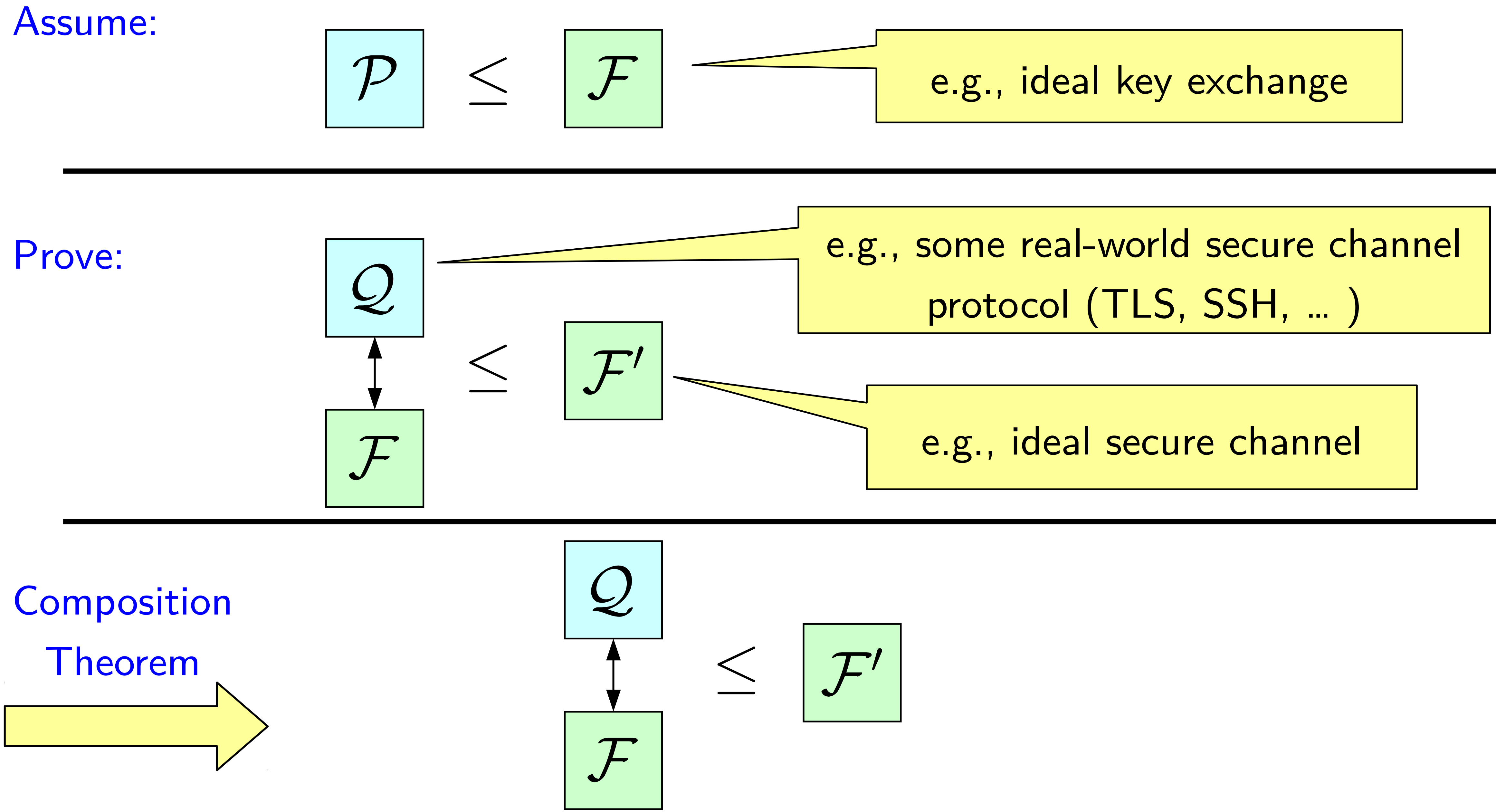
$\mathcal{Q} \leftrightarrow \mathcal{F} \leq \mathcal{F}'$

e.g., some real-world secure channel protocol (TLS, SSH, …)

e.g., ideal secure channel

$\mathcal{Q} \leftrightarrow \mathcal{F} \leq \mathcal{F}'$

Compositional Protocol Analysis



Assume:

$\mathcal{P} \leq \mathcal{F}$

e.g., ideal key exchange

Prove:

$\mathcal{Q} \leftrightarrow \mathcal{F} \leq \mathcal{F}'$

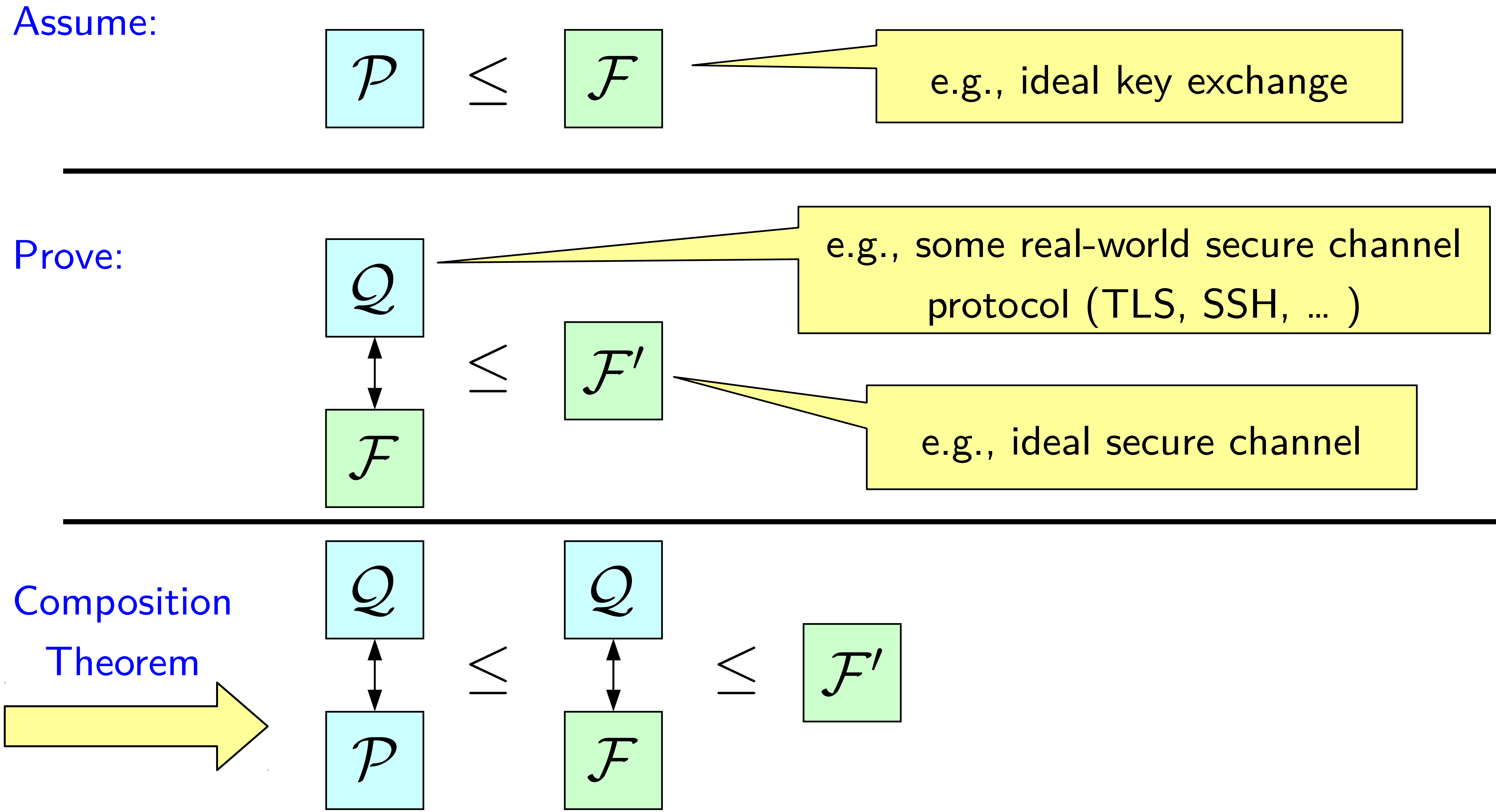
e.g., some real-world secure channel protocol (TLS, SSH, …)

e.g., ideal secure channel

Composition Theorem

$\mathcal{Q} \leftrightarrow \mathcal{F} \leq \mathcal{F}'$

Compositional Protocol Analysis



Assume:

$\boxed{\mathcal{P}} \le \boxed{\mathcal{F}}$

e.g., ideal key exchange

Prove:

$\begin{array}{c}\boxed{\mathcal{Q}}\\ \updownarrow \\ \boxed{\mathcal{F}}\end{array} \le \boxed{\mathcal{F}'}$

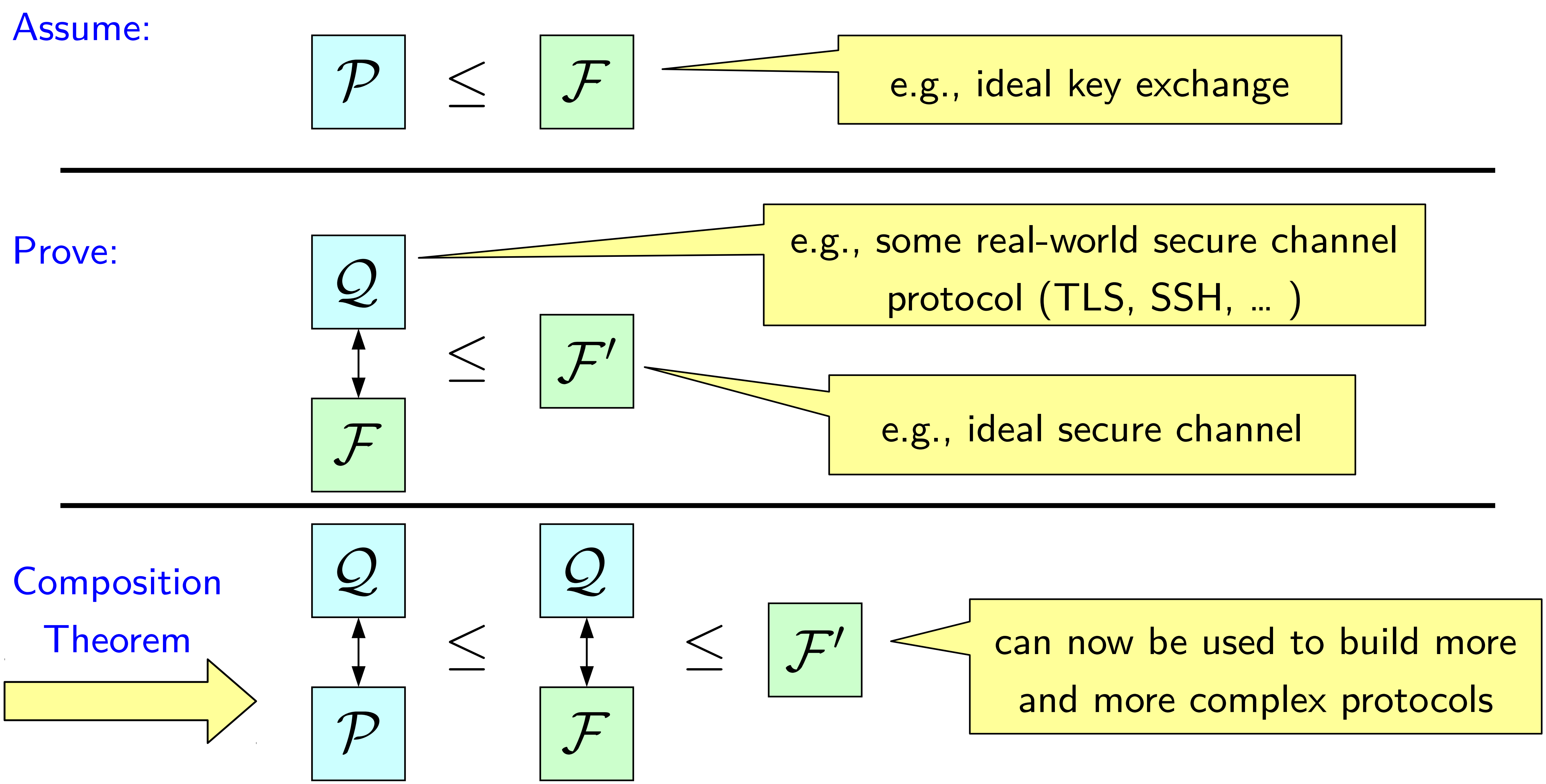
e.g., some real-world secure channel protocol (TLS, SSH, …)

e.g., ideal secure channel

Composition Theorem

$\begin{array}{c}\boxed{\mathcal{Q}}\\ \updownarrow \\ \boxed{\mathcal{P}}\end{array} \le \begin{array}{c}\boxed{\mathcal{Q}}\\ \updownarrow \\ \boxed{\mathcal{F}}\end{array} \le \boxed{\mathcal{F}'}$

Compositional Protocol Analysis



Assume:

$$\mathcal{P} \leq \mathcal{F}$$

e.g., ideal key exchange

Prove:

$$\begin{array}{c}\mathcal{Q}\\ \updownarrow\\ \mathcal{F}\end{array} \leq \mathcal{F}'$$

e.g., some real-world secure channel protocol (TLS, SSH, …)

e.g., ideal secure channel

Composition Theorem

$$\begin{array}{c}\mathcal{Q}\\ \updownarrow\\ \mathcal{P}\end{array} \leq \begin{array}{c}\mathcal{Q}\\ \updownarrow\\ \mathcal{F}\end{array} \leq \mathcal{F}'$$

can now be used to build more and more complex protocols

Universal Composability Models

- UC model [Canetti 2001]
- GNUC model [Hofheinz, Shoup 2011]
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- …

Universal Composability Models

- UC model [Canetti 2001]
- GNUC model [Hofheinz, Shoup 2011]
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- ...

Universal Composability Models

- UC model [Canetti 2001]
- GNUC model [Hofheinz, Shoup 2011]
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- ...

Ideally, a good model should be ...

→ formally sound

→ expressive

→ easy to use

Universal Composability Models

- UC model [Canetti 2001]
- GNUC model [Hofheinz, Shoup 2011]
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- ...

Ideally, a good model should be ...

- → formally sound
- → expressive
- → easy to use

UC Model

The UC model has several severe issues:

UC Model

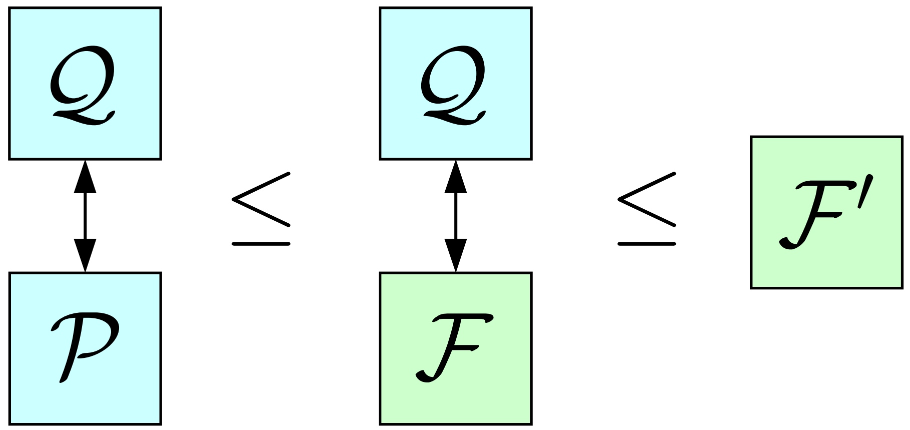
The UC model has several severe issues:

- composition theorem formally does not hold true

UC Model

The UC model has several severe issues:

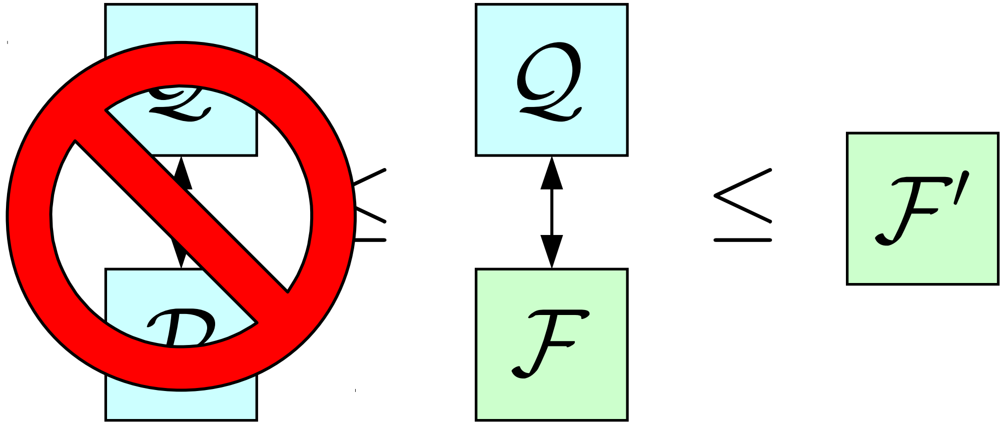
- composition theorem formally does not hold true



UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true



UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true

UC Model

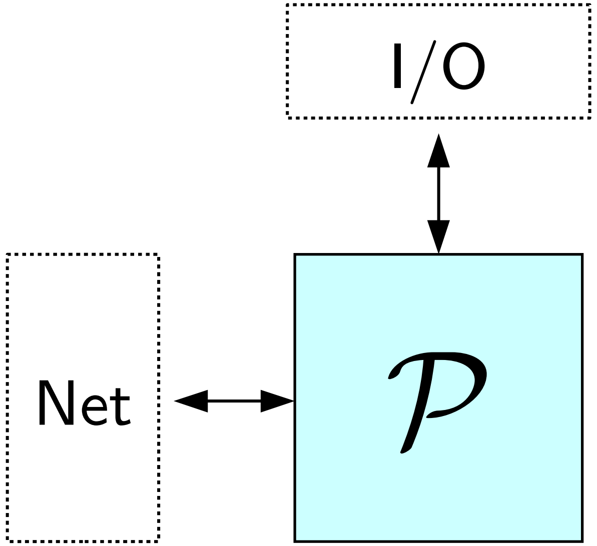
The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

UC Model

The UC model has several severe issues:

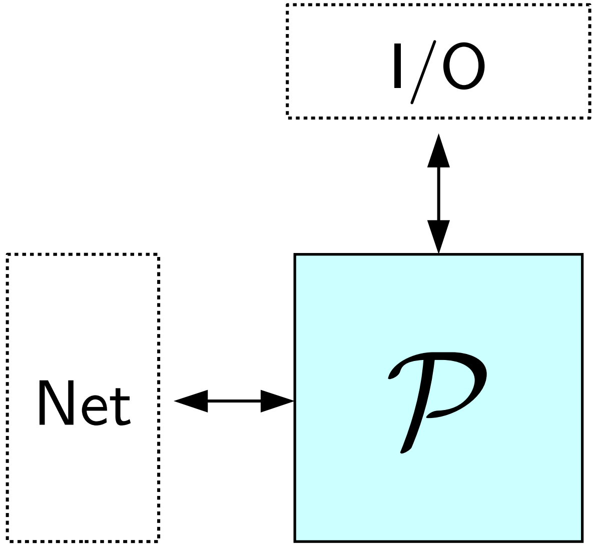
- composition theorem formally does not hold true
- unrealistic runtime definition



UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

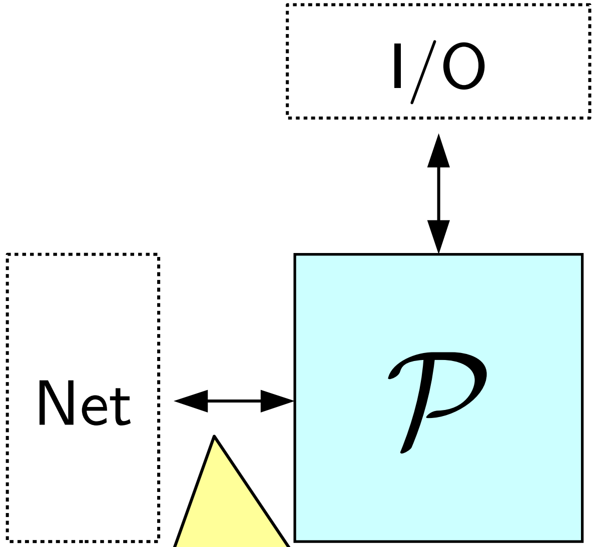


protocols must be ppt in the I/O input (simplified)

UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition



protocols must be ppt in the I/O input (simplified)

if too many messages from the network are received, the protocol exhausts its runtime and stops

UC Model

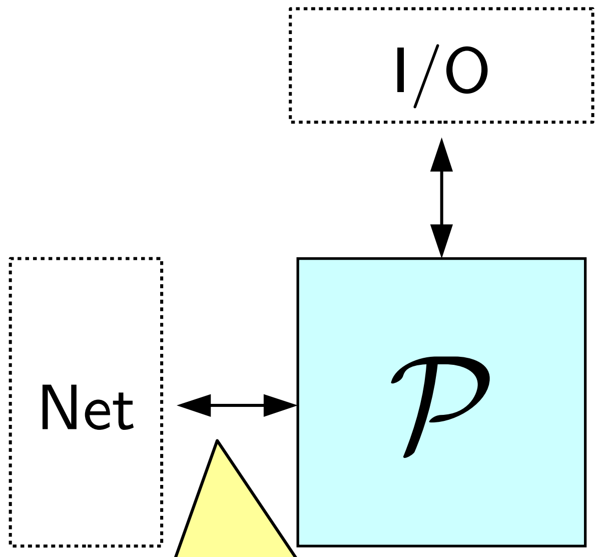
The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

→ limited expressiveness

→ annoying to deal with

protocols must be ppt in the I/O input (simplified)



if too many messages from the network are received, the protocol exhausts its runtime and stops

UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

  → limited expressiveness

  → annoying to deal with

UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

  → limited expressiveness

  → annoying to deal with
- modeling assumptions

UC Model

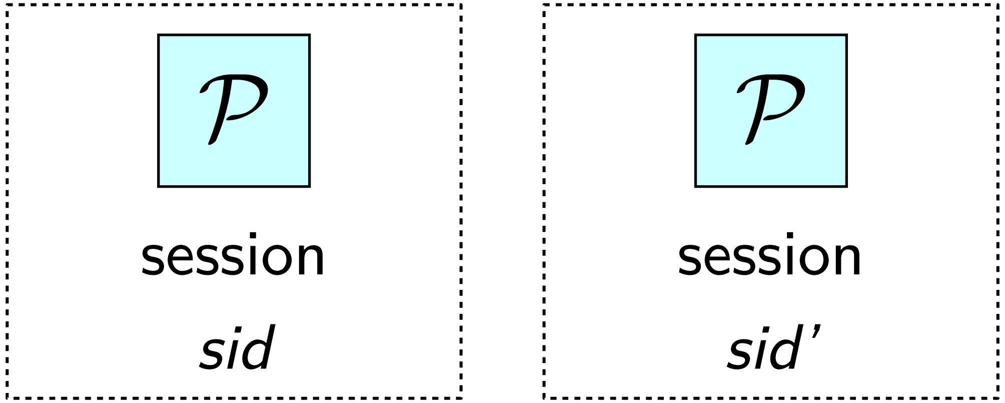
The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

  → limited expressiveness

  → annoying to deal with
- modeling assumptions

Example:



UC Model

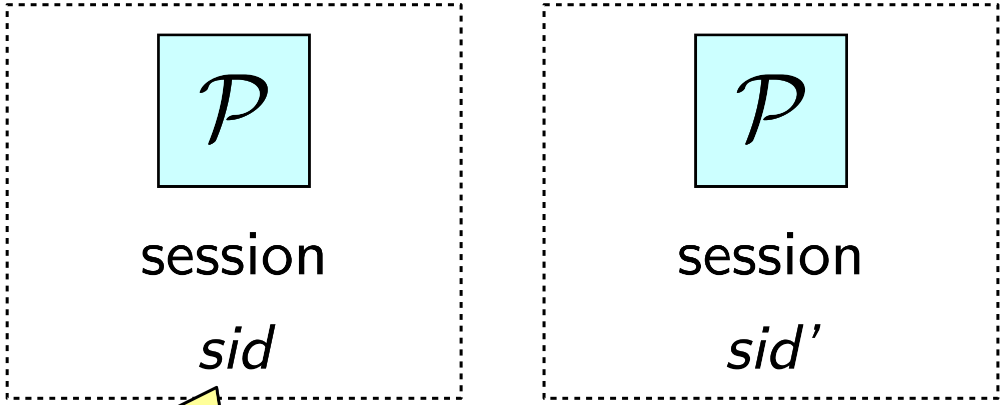
The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

  → limited expressiveness

  → annoying to deal with
- modeling assumptions

Example:



Protocol instances from the same session are assumed to have a shared global session ID

UC Model

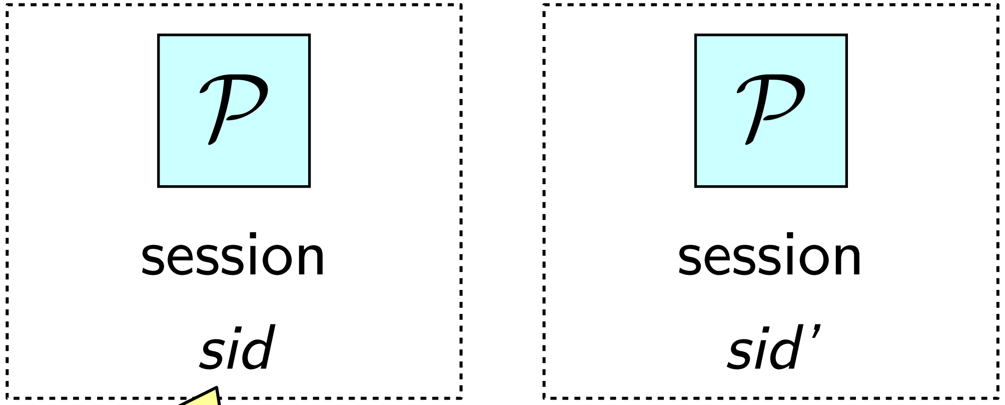
The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

  → limited expressiveness

  → annoying to deal with
- modeling assumptions

Example:



Protocol instances from the same session are assumed to have a shared global session ID

→ cannot faithfully analyze protocols without explicit SID establishment at the start

UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

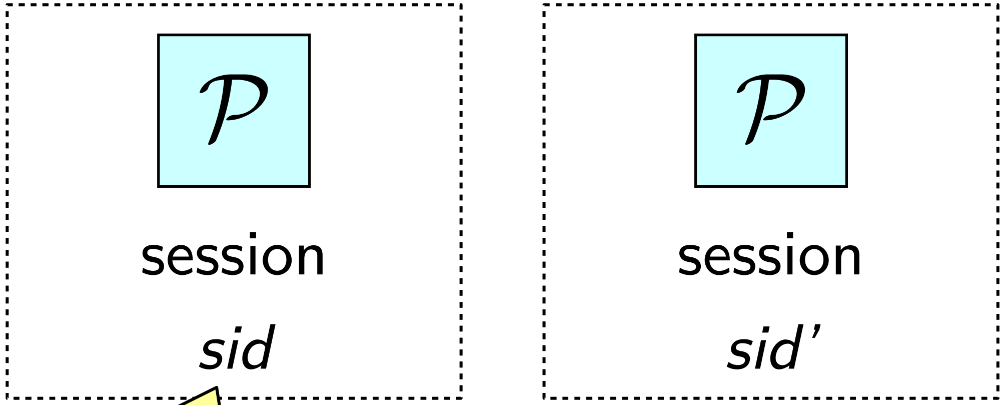
  → limited expressiveness

  → annoying to deal with
- modeling assumptions

  → limited expressiveness

  → extensions necessary shared state

Example:



Protocol instances from the same session are assumed to have a shared global session ID

→ cannot faithfully analyze protocols without explicit SID establishment at the start

UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

  → limited expressiveness

  → annoying to deal with
- modeling assumptions

  → limited expressiveness

  → extensions necessary shared state
- high complexity

  → causes artificial attacks

  → hard to deal with

UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

  → limited expressiveness

  → annoying to deal with
- modeling assumptions

  → limited expressiveness

  → extensions necessary shared state
- high complexity

  → causes artificial attacks

  → hard to deal with

Canetti et al.:
”We are not aware of any written proof in the UC framework that actually takes these details [of the model] into account.”

“A Simpler Variant of Universally Composable Security for Standard Multiparty Computation”, CRYPTO 2015

UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

  → limited expressiveness

  → annoying to deal with
- modeling assumptions

  → limited expressiveness

  → extensions necessary shared state
- high complexity

  → causes artificial attacks

  → hard to deal with
- ...

UC Model

The UC model has several severe issues:

- composition theorem formally does not hold true
- unrealistic runtime definition

  → limited expressiveness

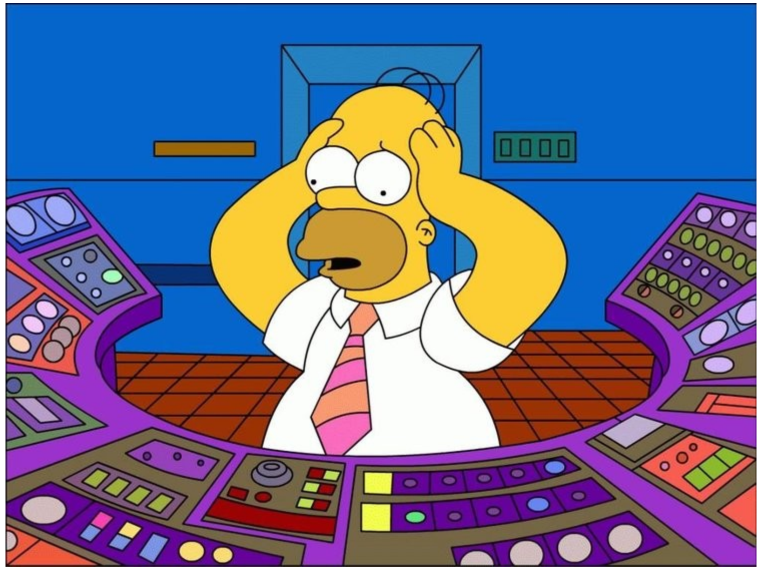
  → annoying to deal with
- modeling assumptions

  → limited expressiveness

  → extensions necessary shared state
- high complexity

  → causes artificial attacks

  → hard to deal with
- ...



No one wants to care about these issues when analyzing a protocol!

Universal Composability Models

- UC model [Canetti 2001]
- GNUC model [Hofheinz, Shoup 2011]
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- ...

Ideally, a good model should be ...

- ➔ formally sound
- ➔ expressive
- ➔ easy to use

Universal Composability Models

- UC model [Canetti 2001] 🚫
- GNUC model [Hofheinz, Shoup 2011]
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- ...

Ideally, a good model should be ...

→ formally sound

→ expressive

→ easy to use

Universal Composability Models

- UC model [Canetti 2001] 🚫
- GNUC model [Hofheinz, Shoup 2011]
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- ...

Ideally, a good model should be ...

- ➔ formally sound
- ➔ expressive
- ➔ easy to use

GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

composition works ✓

GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

composition works ✓

However, it is still not a perfect model:

GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

composition works ✓

However, it is still not a perfect model:

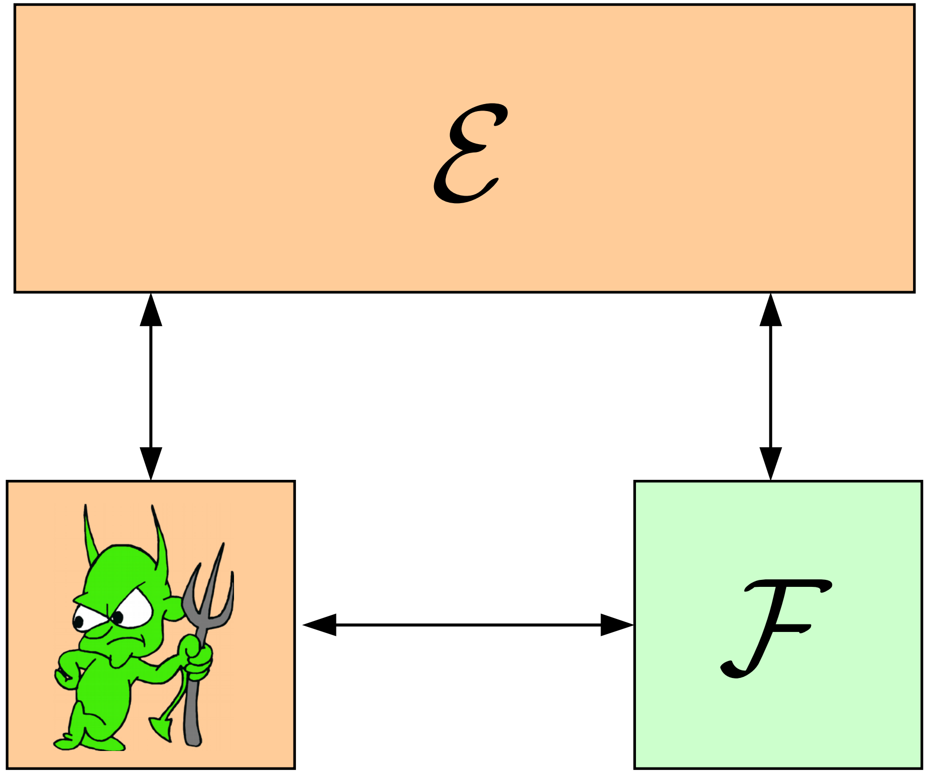
- cumbersome flow bounds

GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

- cumbersome flow bounds

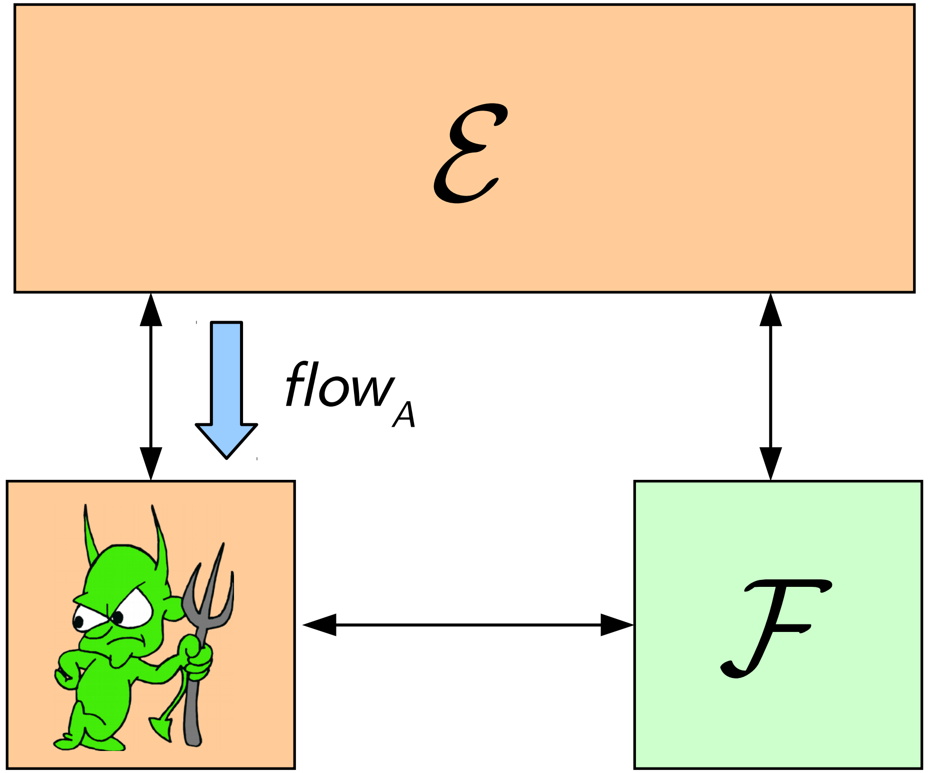


GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

- cumbersome flow bounds

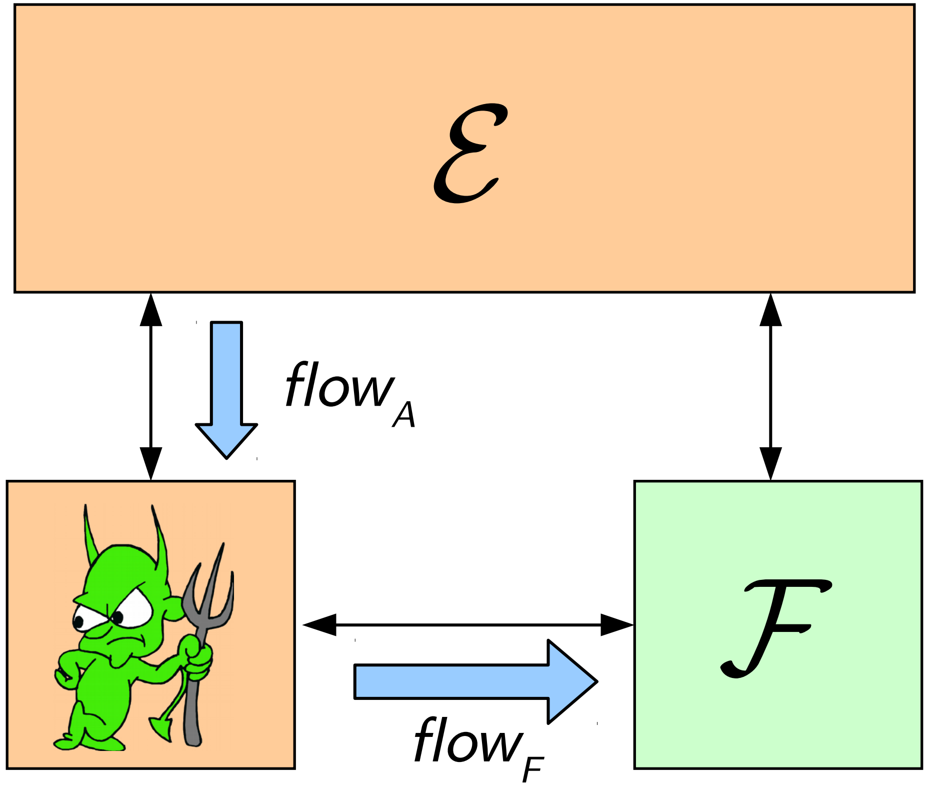


GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

- cumbersome flow bounds

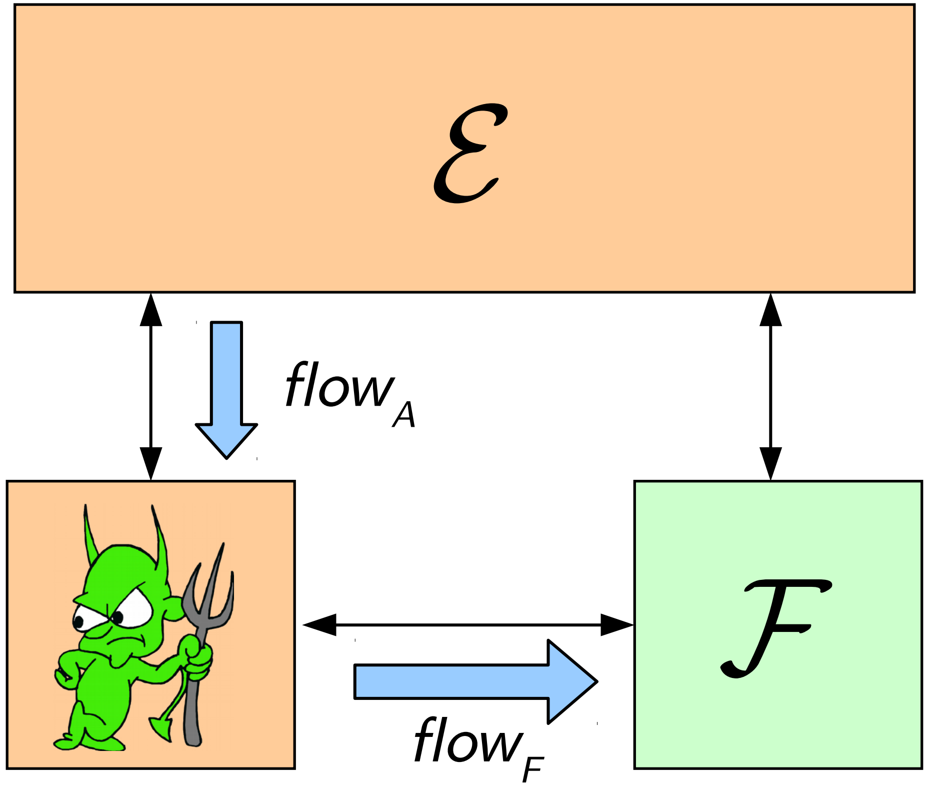


GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

- cumbersome flow bounds



It must hold true that (simplified):

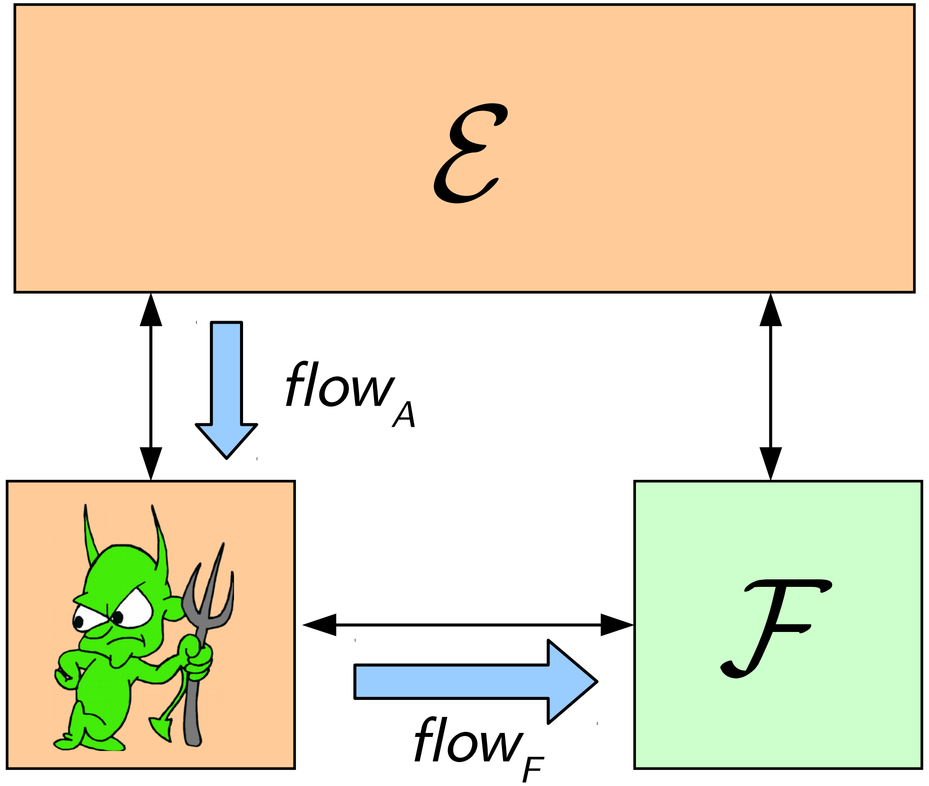
$flow_F < p(flow_A)$ (for some polynomial p)

GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

- cumbersome flow bounds



It must hold true that (simplified):

$$flow_F < p(flow_A) \qquad \text{(for some polynomial } p\text{)}$$

There are natural ideal protocols with corresponding real protocols where no simulator meets this requirement.

GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

- cumbersome flow bounds

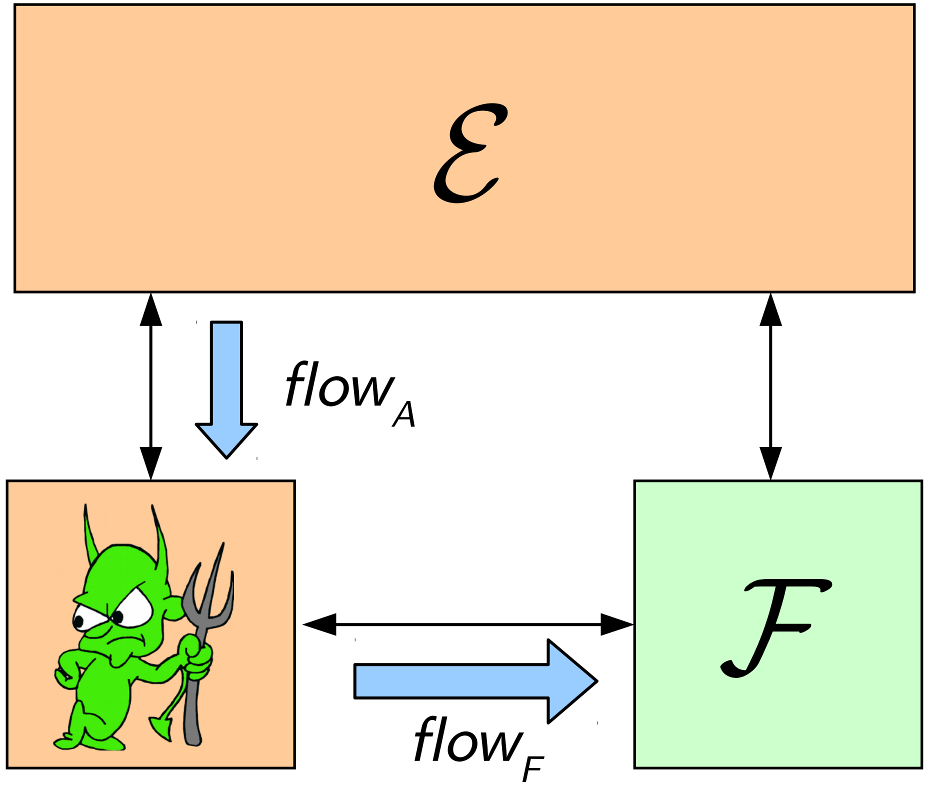
  → limited expressiveness

  → annoying to deal with

It must hold true that (simplified):

$flow_F < p(flow_A)$ (for some polynomial p)

There are natural ideal protocols with corresponding real protocols where no simulator meets this requirement.



GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

- cumbersome flow bounds

  → limited expressiveness

  → annoying to deal with

GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

- cumbersome flow bounds

  → limited expressiveness

  → annoying to deal with

- modeling assumptions

GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

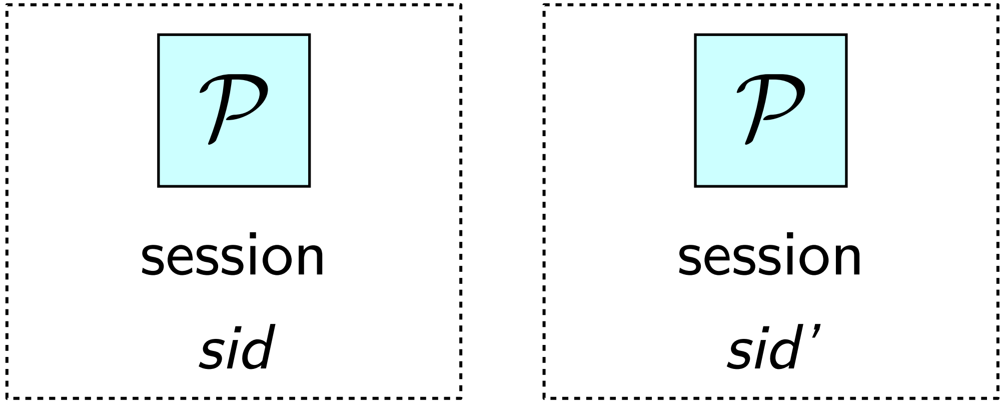
- cumbersome flow bounds

  → limited expressiveness

  → annoying to deal with

- modeling assumptions

Example 1: global SIDs (as in UC)



GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

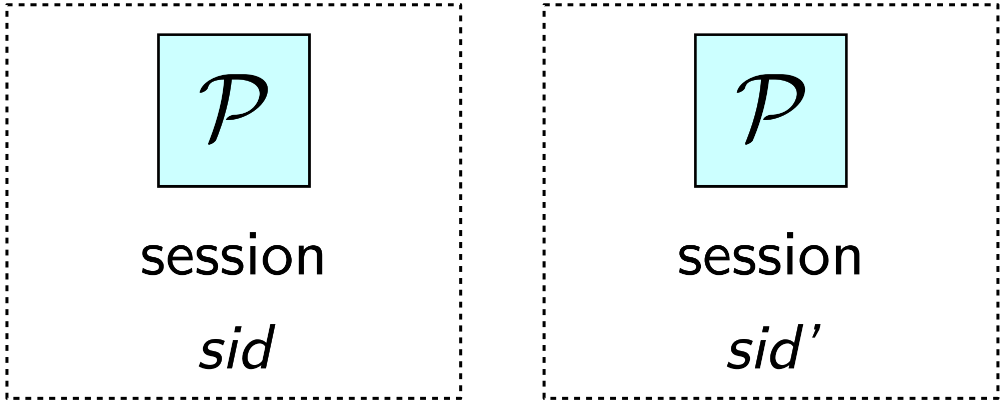
- cumbersome flow bounds

  → limited expressiveness

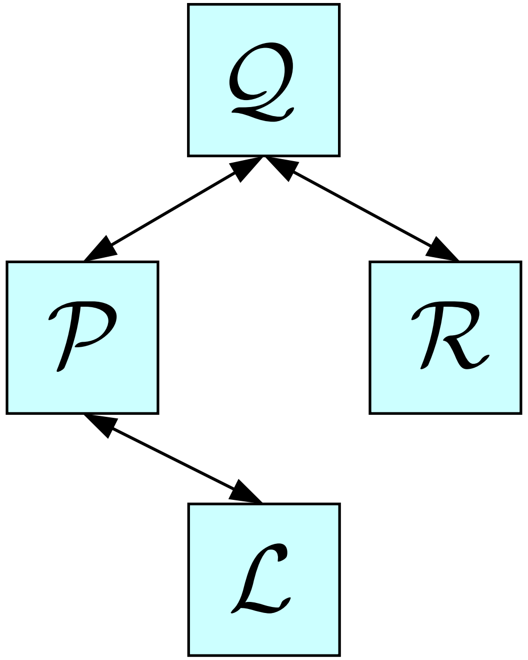
  → annoying to deal with

- modeling assumptions

Example 1: global SIDs (as in UC)



Example 2: unique callers for real subroutines



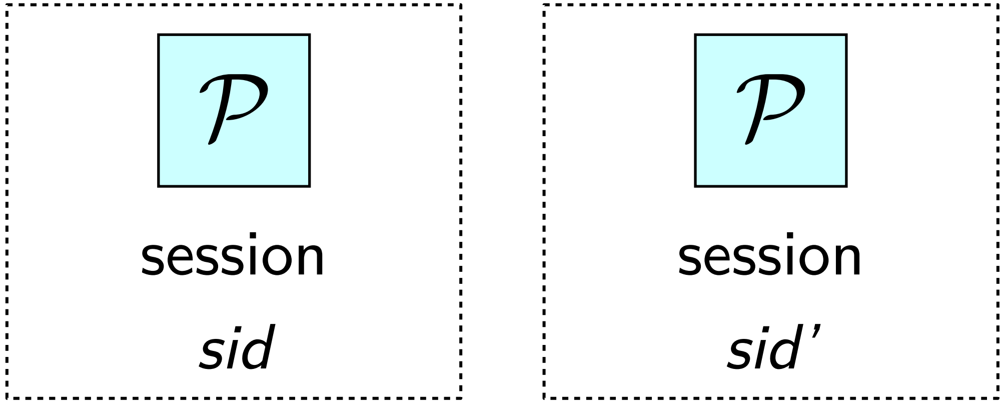
GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

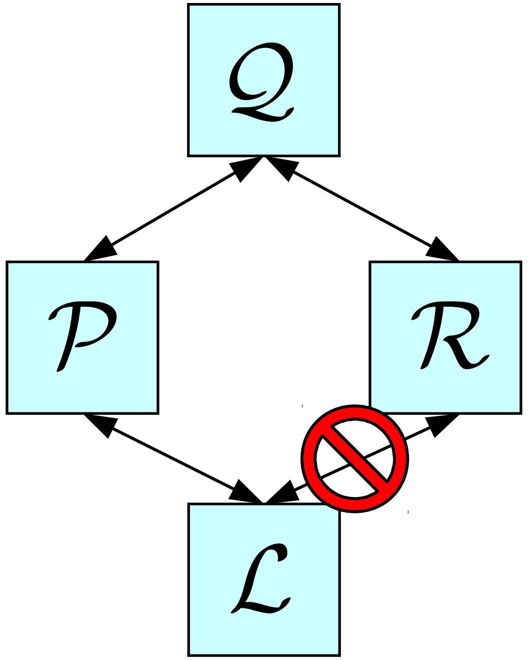
However, it is still not a perfect model:

- cumbersome flow bounds
  - → limited expressiveness
  - → annoying to deal with
- modeling assumptions

Example 1: global SIDs (as in UC)



Example 2: unique callers for real subroutines



GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

- cumbersome flow bounds

  → limited expressiveness

  → annoying to deal with

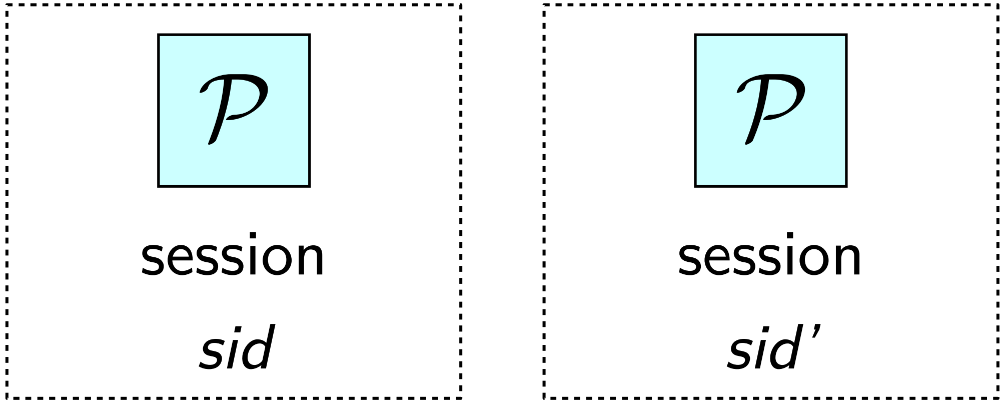
- modeling assumptions

  → partially more restrictive than UC model

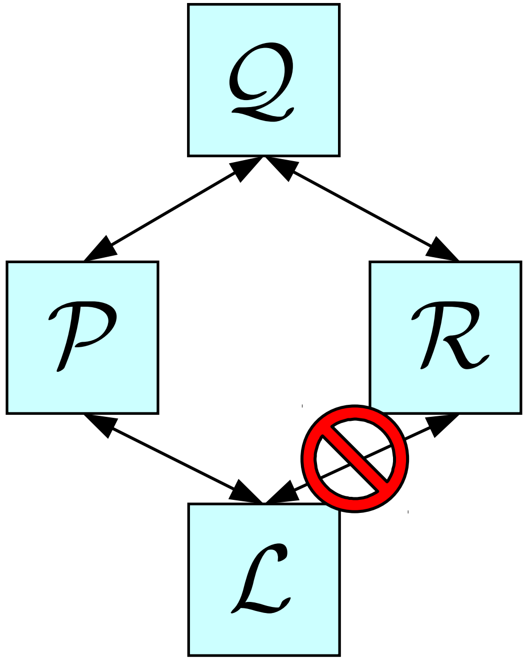
  → limited expressiveness

  → also needs extensions for shared state

Example 1: global SIDs (as in UC)



Example 2: unique callers for real subroutines



GNUC Model

The GNUC model was developed as a formally sound alternative to the UC model.

However, it is still not a perfect model:

- cumbersome flow bounds

  → limited expressiveness

  → annoying to deal with

- modeling assumptions

  → partially more restrictive than UC model

  → limited expressiveness

  → also needs extensions for shared state

sound model, but rather rigid structure

Universal Composability Models

- UC model [Canetti 2001] 🚫
- GNUC model [Hofheinz, Shoup 2011]
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- …

Ideally, a good model should be …

- → formally sound
- → expressive
- → easy to use

Universal Composability Models

- UC model [Canetti 2001] 🚫
- GNUC model [Hofheinz, Shoup 2011] ~
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- …

Ideally, a good model should be …

- → formally sound
- → expressive
- → easy to use

Universal Composability Models

- UC model [Canetti 2001] 🚫
- GNUC model [Hofheinz, Shoup 2011] ~
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- …

Ideally, a good model should be …

➔ formally sound

➔ expressive

➔ easy to use

IITM Model

The IITM model is a very general model that makes as few modeling assumptions as are necessary to show the composition theorem.

IITM Model

The IITM model is a very general model that makes as few modeling assumptions as are necessary to show the composition theorem.

Main features are:

- formally sound

IITM Model

The IITM model is a very general model that makes as few modeling assumptions as are necessary to show the composition theorem.

Main features are:

- formally sound
- simple

  e.g., natural runtime notion, no flow bounds

IITM Model

The IITM model is a very general model that makes as few modeling assumptions as are necessary to show the composition theorem.

Main features are:

- formally sound
- simple

  e.g., natural runtime notion, no flow bounds
- highly flexible

  e.g., session structure not fixed by the model
  No extensions necessary for shared state

IITM Model

The IITM model is a very general model that makes as few modeling assumptions as are necessary to show the composition theorem.

Main features are:

- formally sound
- simple

  e.g., natural runtime notion, no flow bounds
- highly flexible

  e.g., session structure not fixed by the model
  No extensions necessary for shared state
- enhanced usability due to responsive environments

IITM Model

The IITM model is a very general model that makes as few modeling assumptions as are necessary to show the composition theorem.

Main features are:

- formally sound
- simple

  e.g., natural runtime notion, no flow bounds
- highly flexible

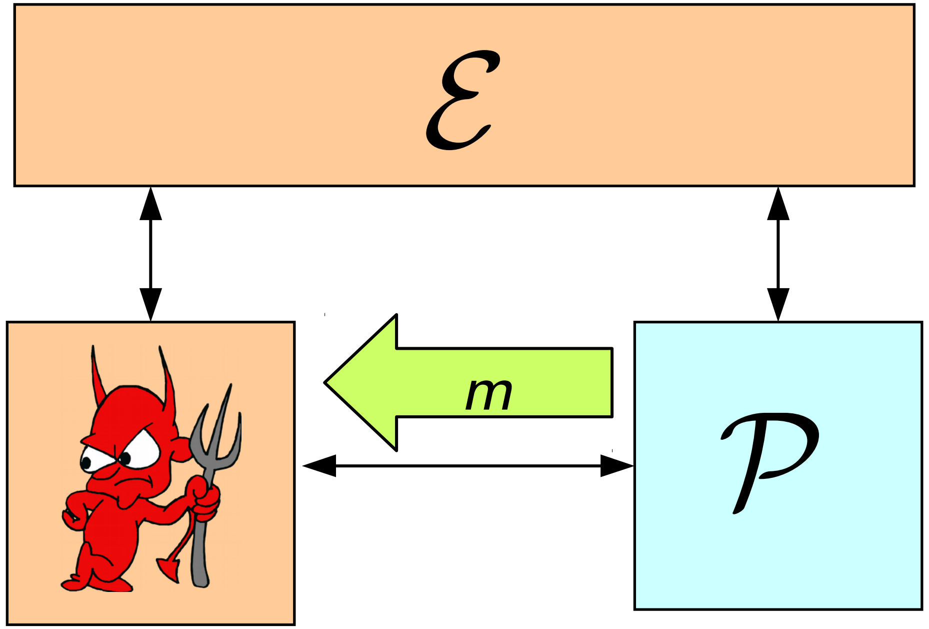
  e.g., session structure not fixed by the model
  No extensions necessary for shared state
- enhanced usability due to responsive environments

IITM Model With Responsive Environments

Responsive environments ease handling of so-called urgent requests:

IITM Model With Responsive Environments

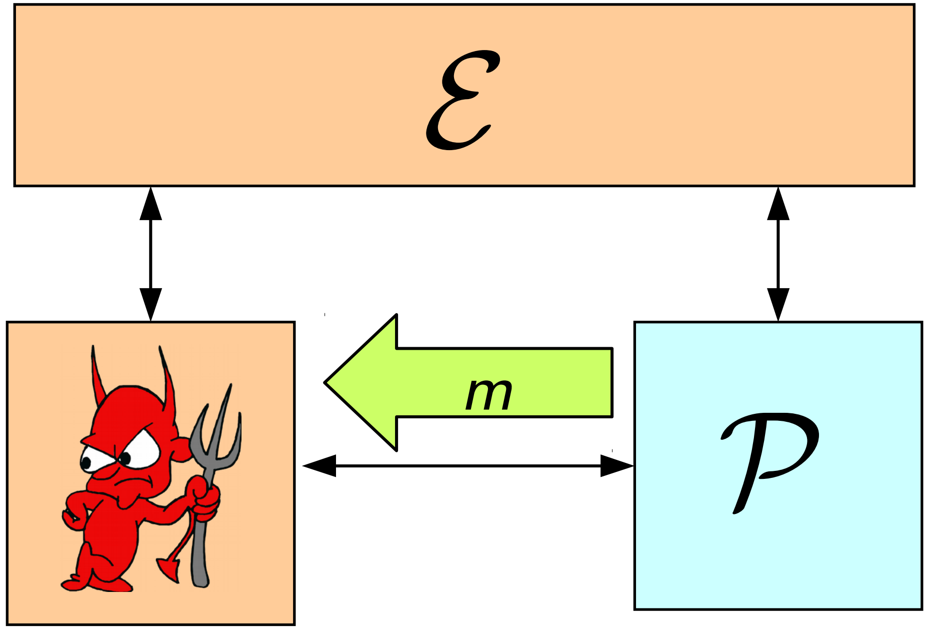
Responsive environments ease handling of so-called urgent requests:



IITM Model With Responsive Environments

Responsive environments ease handling of so-called urgent requests:

- about meta-information, e.g.,
  - request corruption status
  - request cryptographic material (keys, …)
  - information leakage
  - signaling information

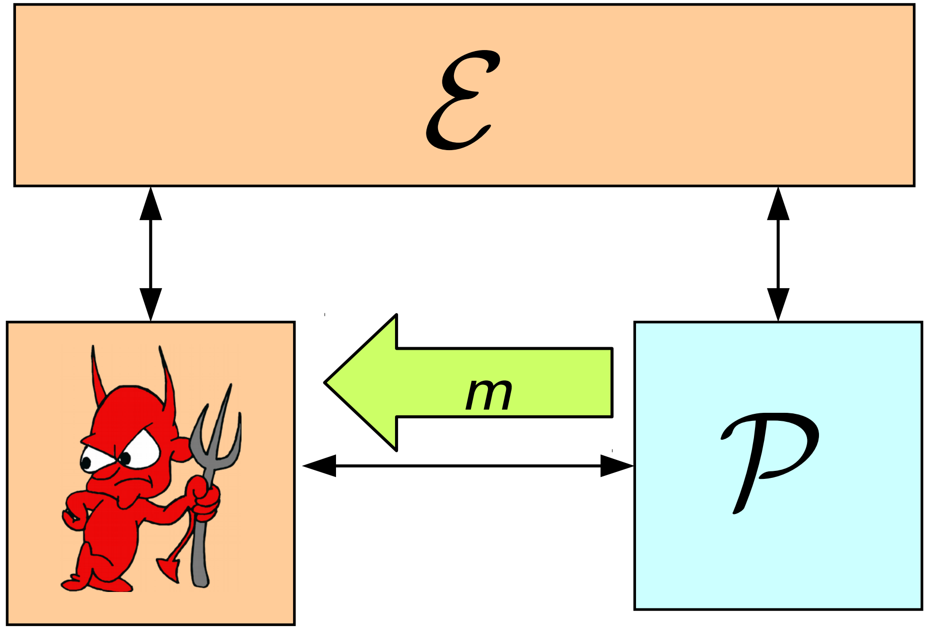


IITM Model With Responsive Environments

Responsive environments ease handling of so-called urgent requests:

- about meta-information, e.g.,
  - request corruption status
  - request cryptographic material (keys, …)
  - information leakage
  - signaling information
- only exist for modeling purposes

  → in reality, protocol execution continues without interference from adversary



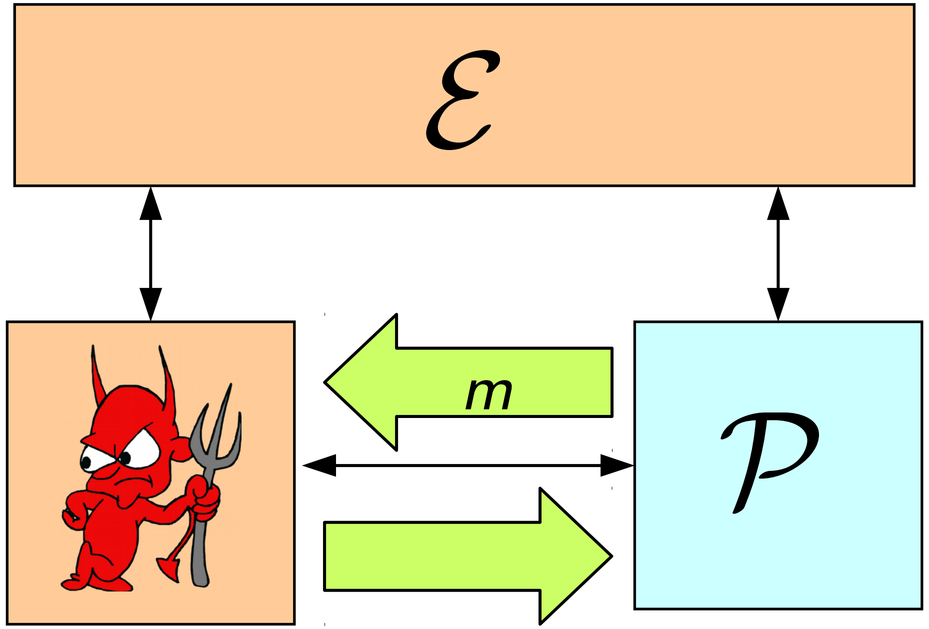
IITM Model With Responsive Environments

Responsive environments ease handling of so-called urgent requests:

- about meta-information, e.g.,
  - request corruption status
  - request cryptographic material (keys, …)
  - information leakage
  - signaling information
- only exist for modeling purposes

  → in reality, protocol execution continues without interference from adversary

  → natural to expect immediate response in the protocol model



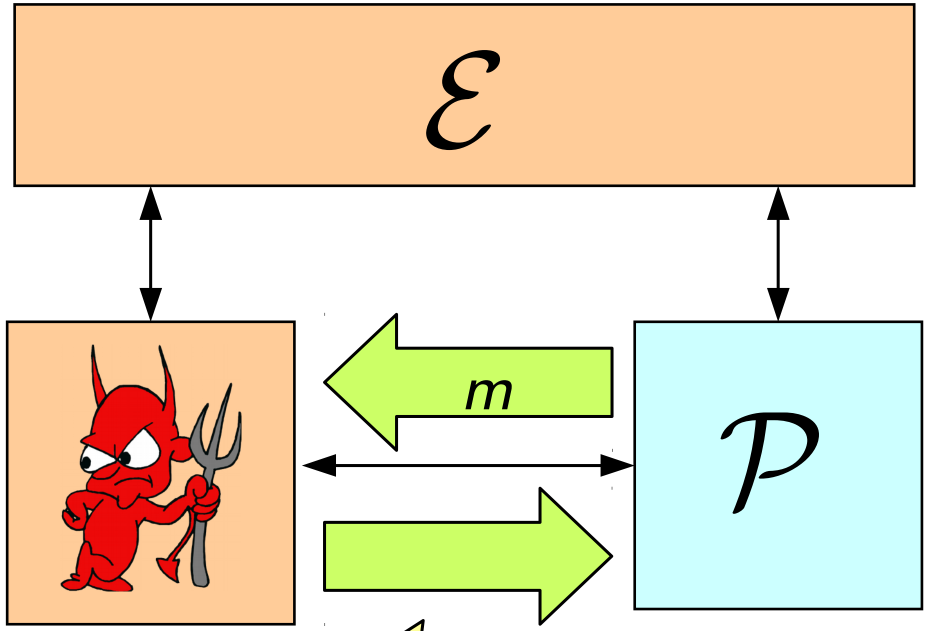
IITM Model With Responsive Environments

Responsive environments ease handling of so-called urgent requests:

- about meta-information, e.g.,
  - request corruption status
  - request cryptographic material (keys, …)
  - information leakage
  - signaling information
- only exist for modeling purposes

  → in reality, protocol execution continues without interference from adversary

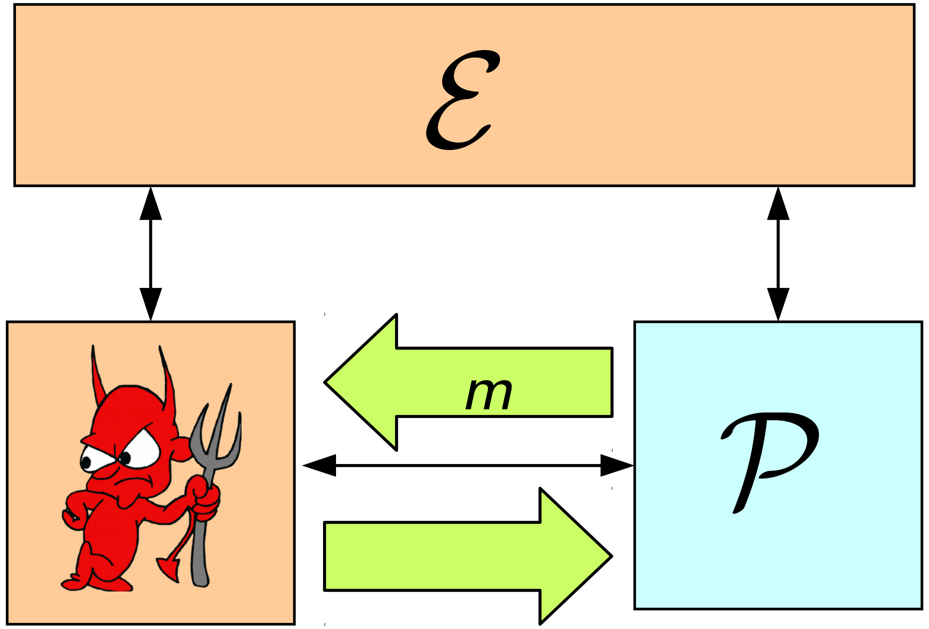
  → natural to expect immediate response in the protocol model



not the case in any of the models without responsive environments!

IITM Model With Responsive Environments

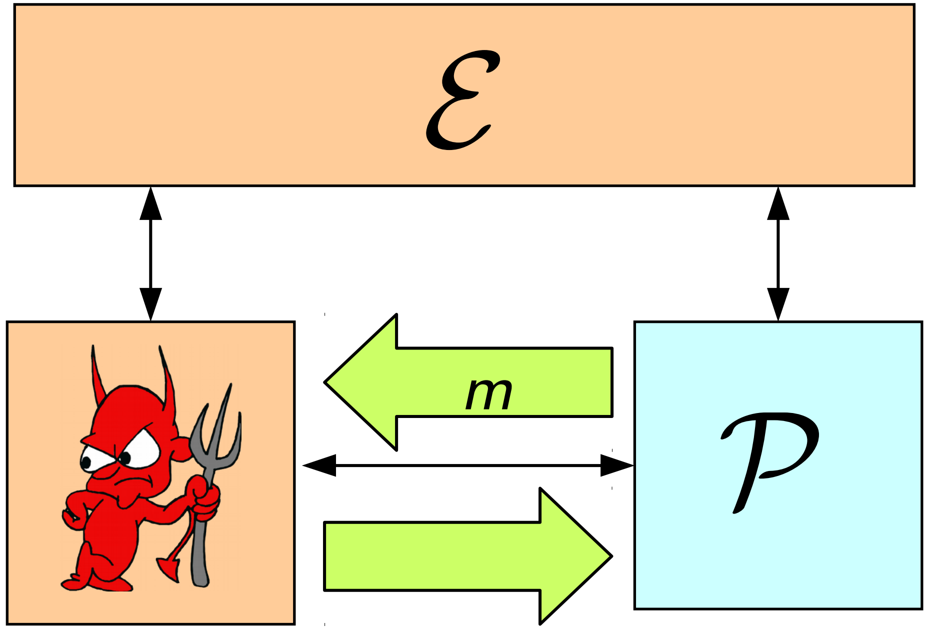
Instead of responding immediately to an urgent request, the adversary might:



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

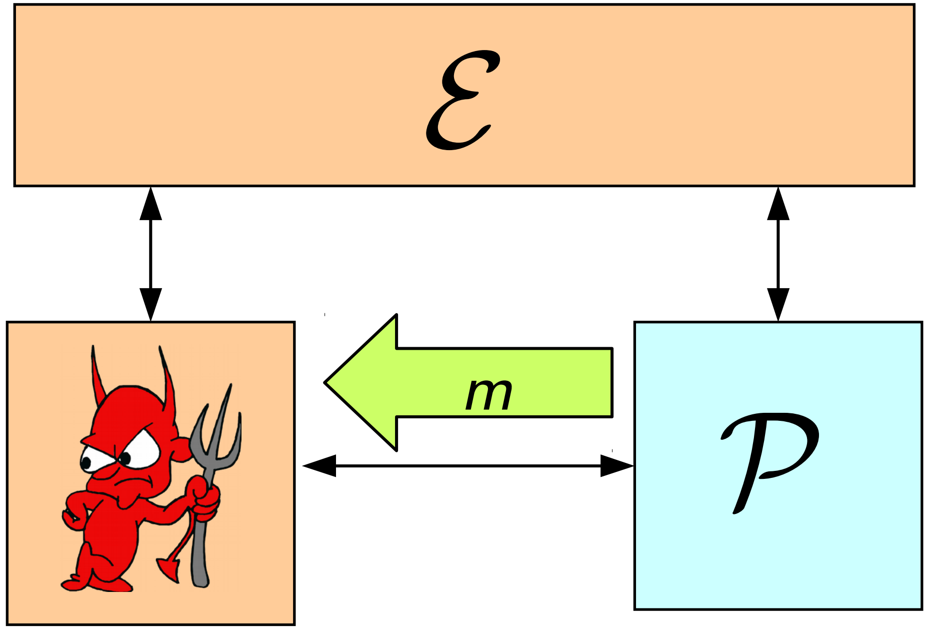
- activate the protocol in unexpected ways



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

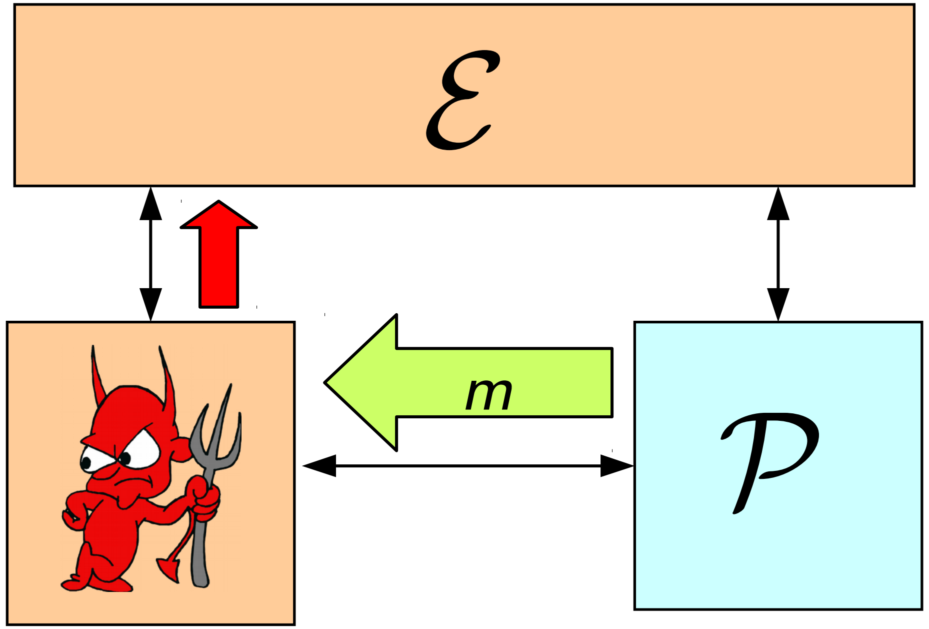
- activate the protocol in unexpected ways



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

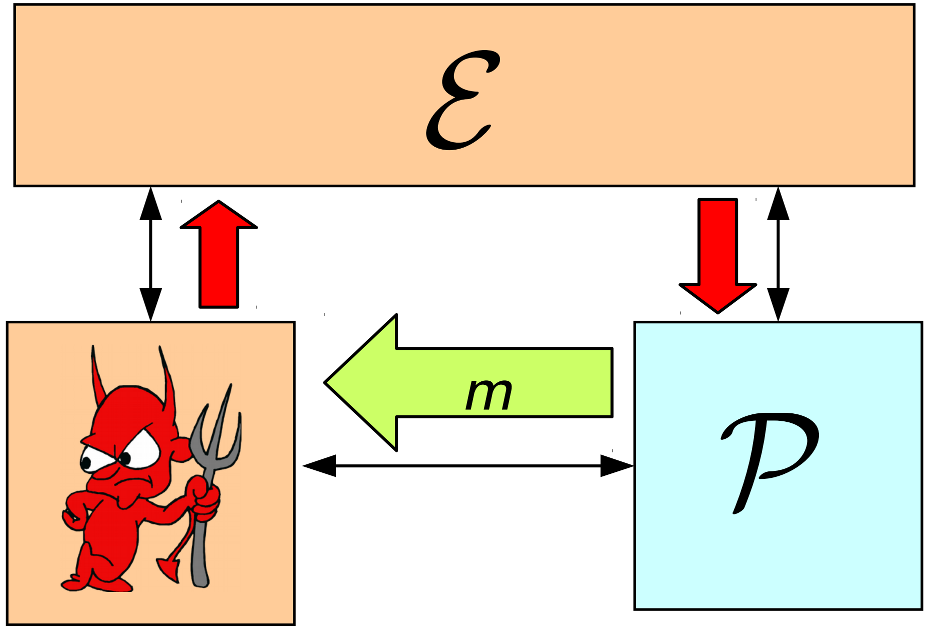
- activate the protocol in unexpected ways



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

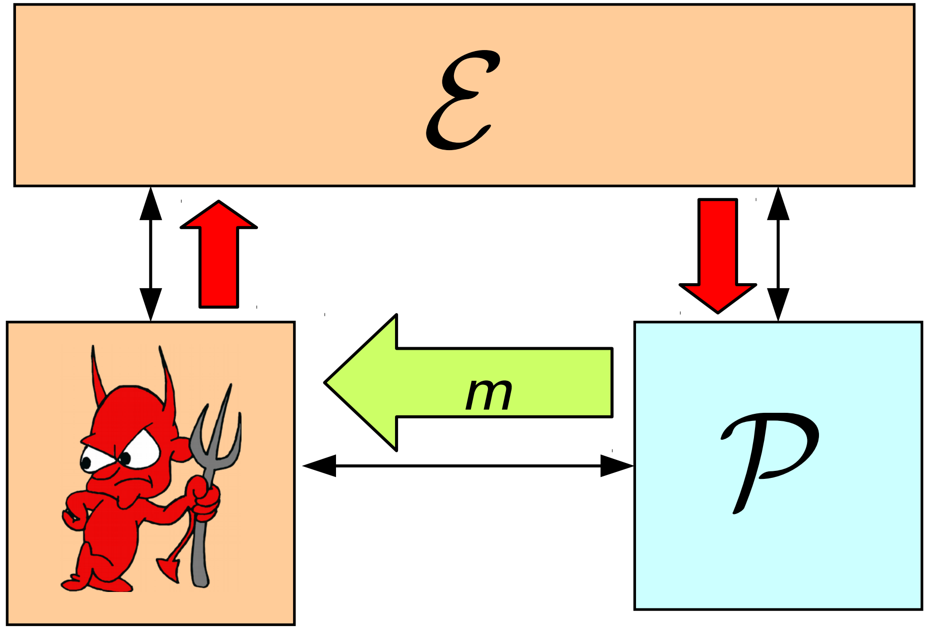
- activate the protocol in unexpected ways



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

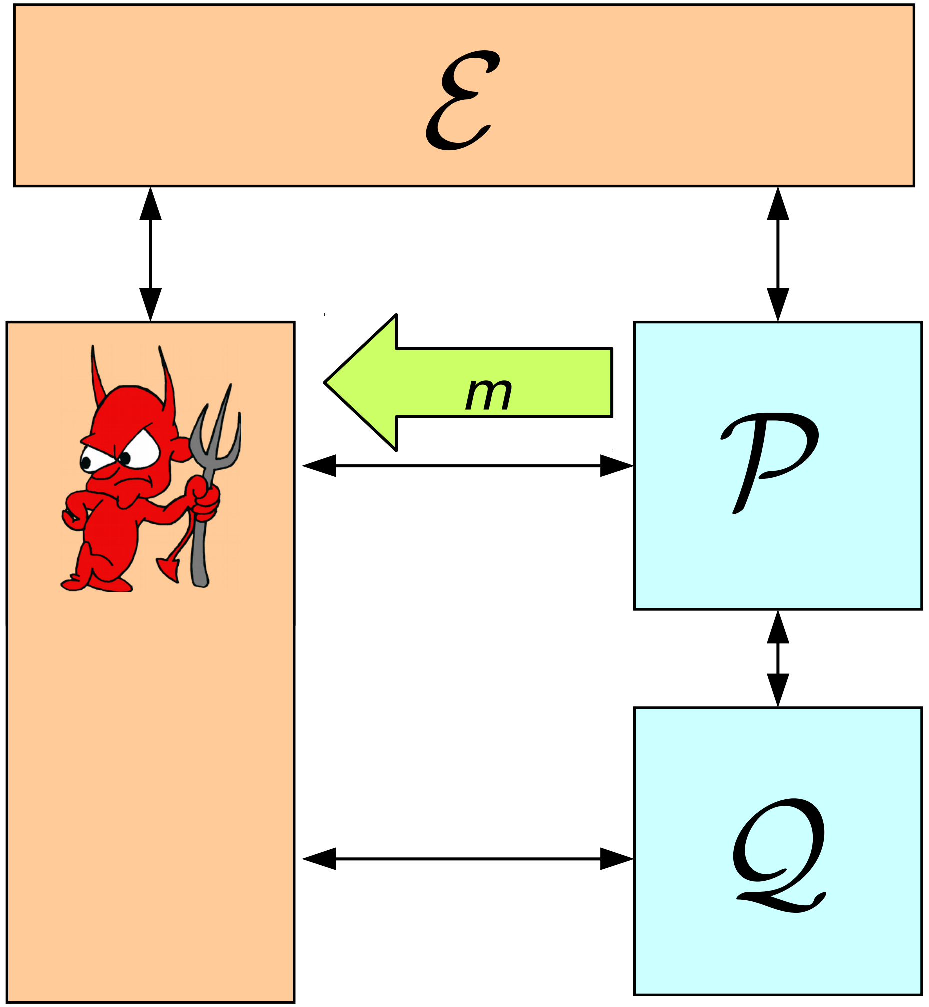
- activate the protocol in unexpected ways
- activate and change state of other parts of the protocol



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

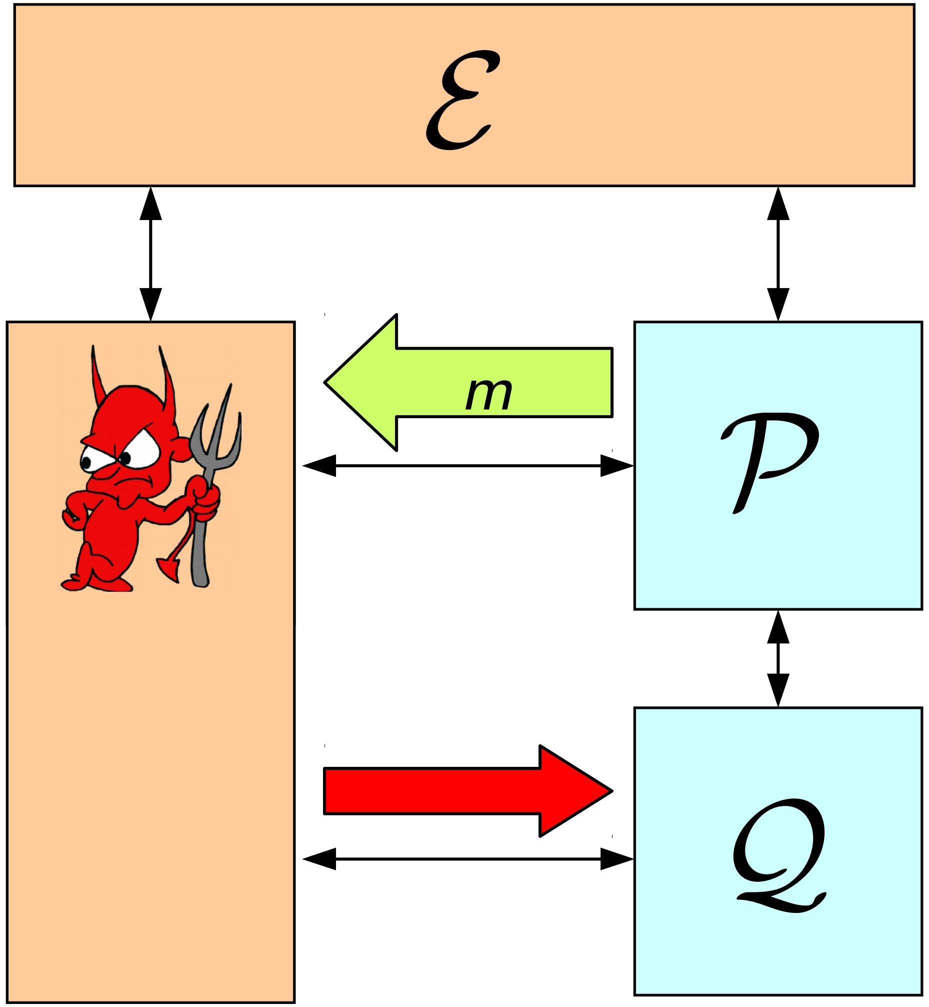
- activate the protocol in unexpected ways
- activate and change state of other parts of the protocol



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

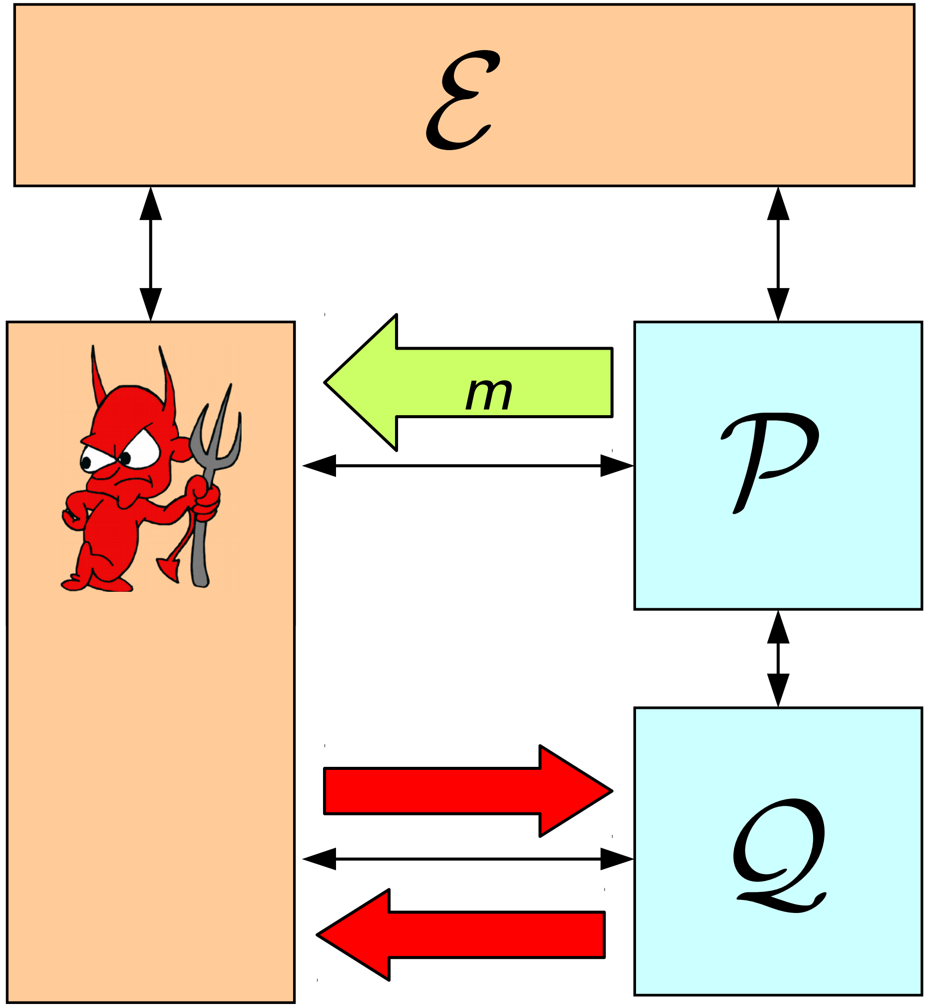
- activate the protocol in unexpected ways
- activate and change state of other parts of the protocol



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

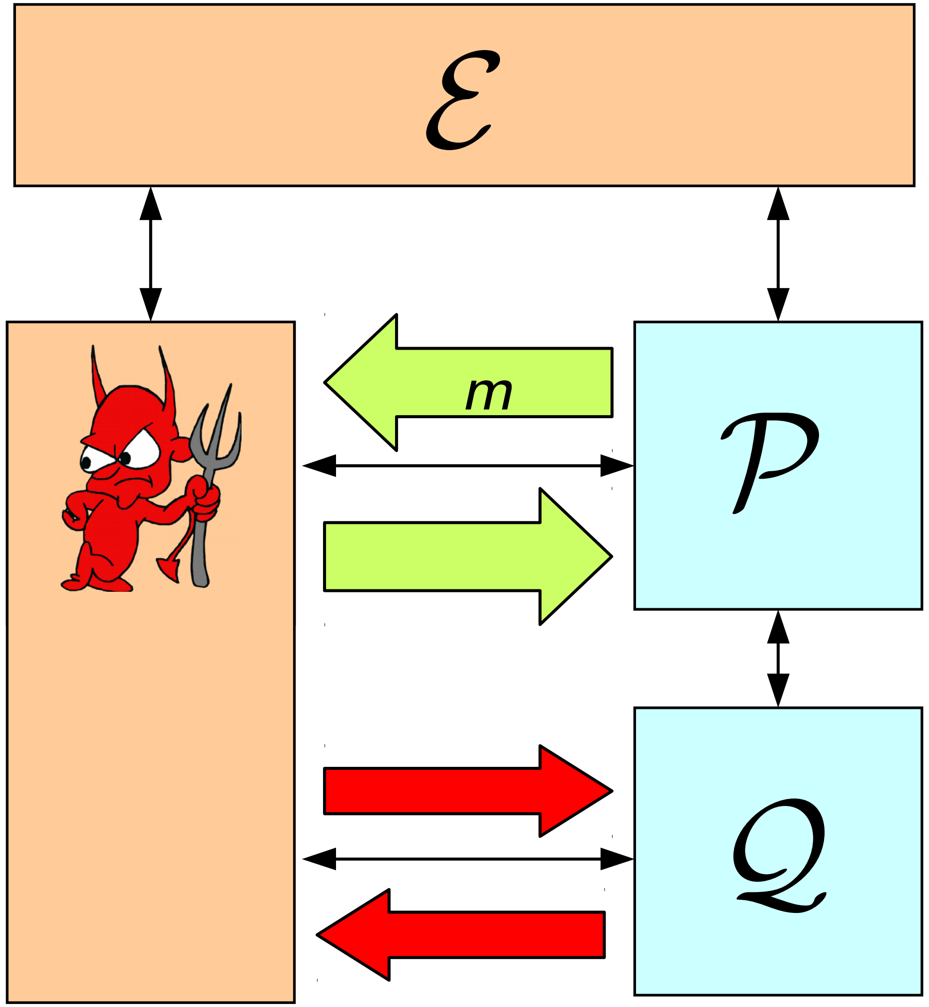
- activate the protocol in unexpected ways
- activate and change state of other parts of the protocol



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

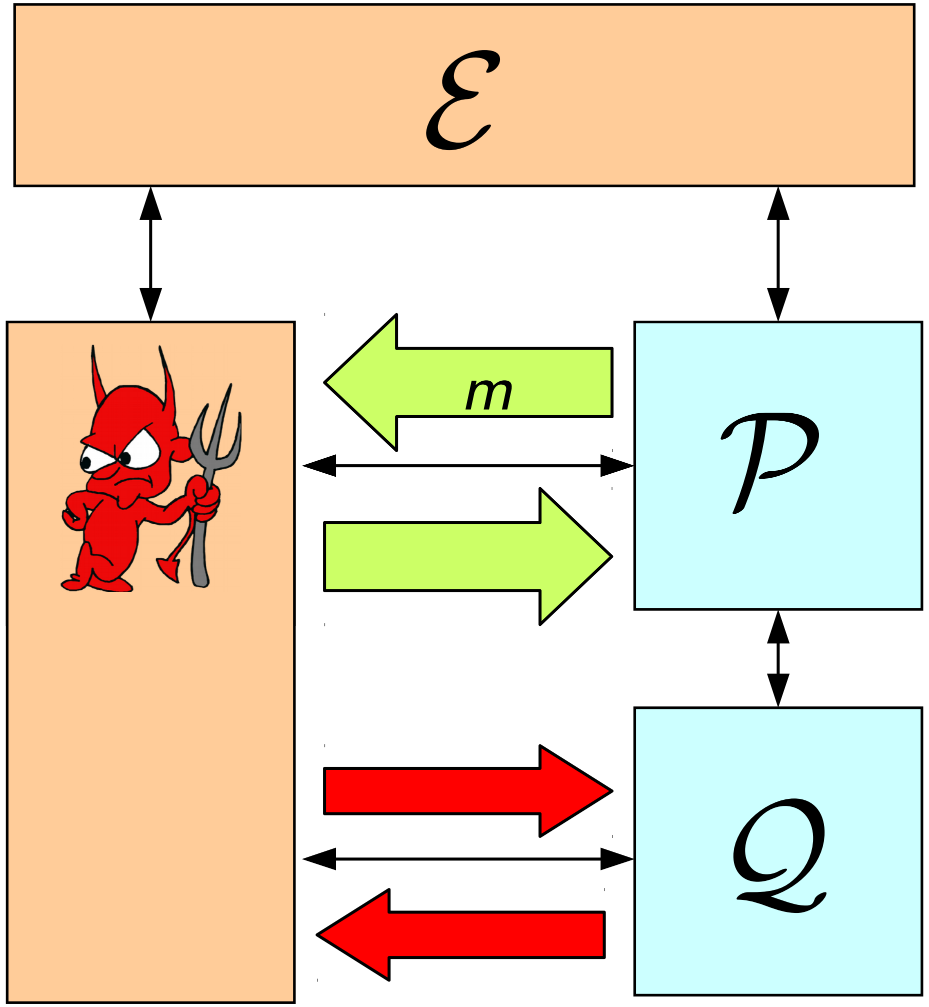
- activate the protocol in unexpected ways
- activate and change state of other parts of the protocol



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

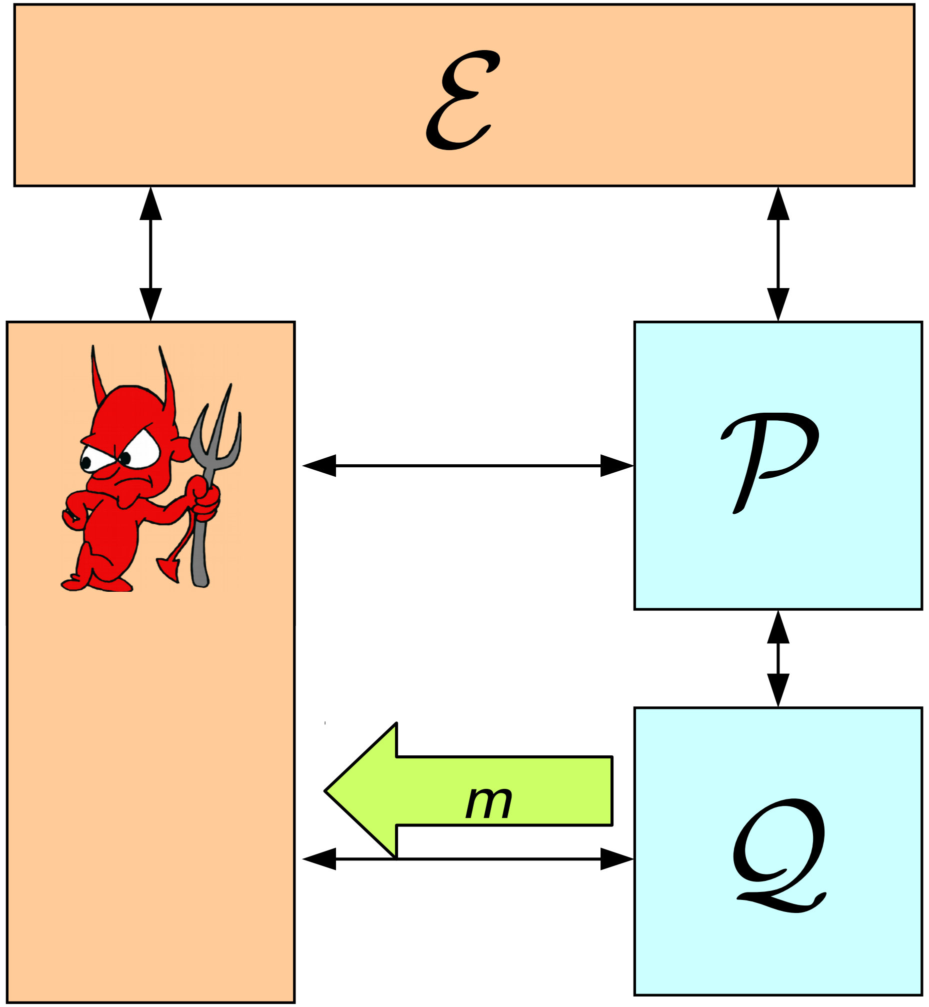
- activate the protocol in unexpected ways
- activate and change state of other parts of the protocol
- block parts of the protocol by never responding



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

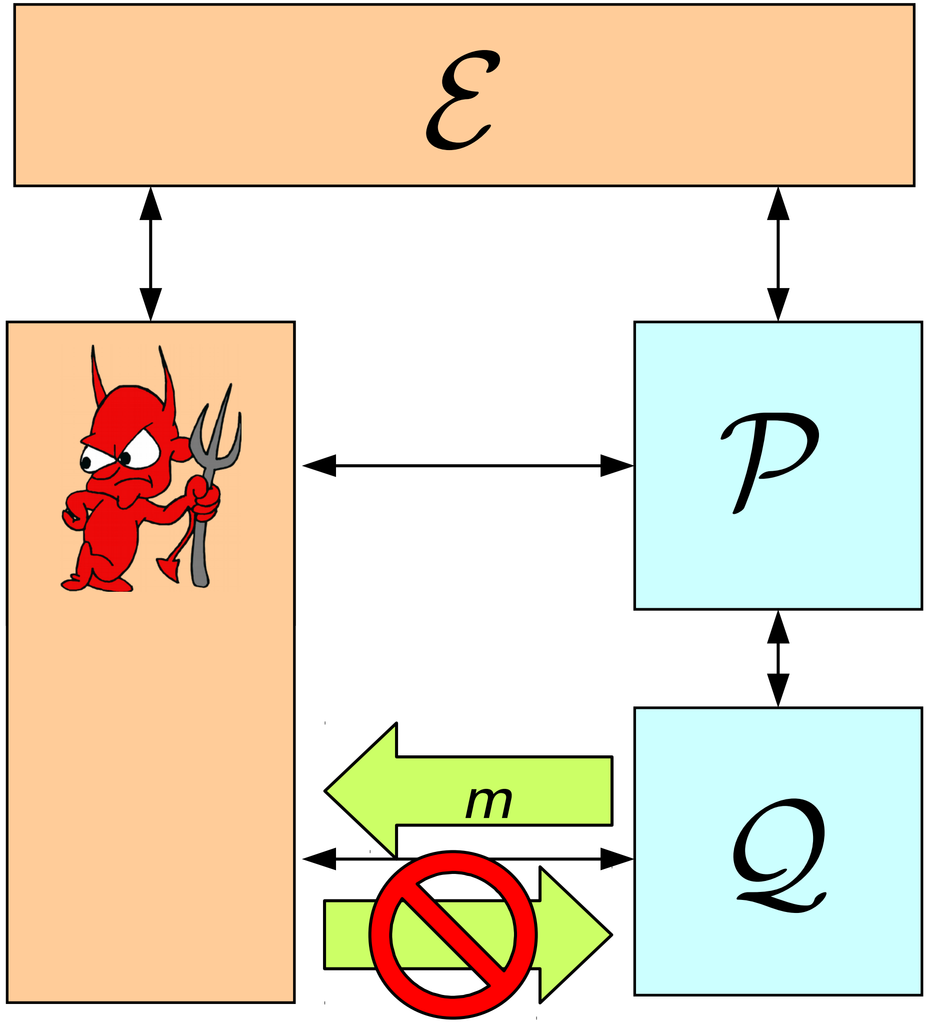
- activate the protocol in unexpected ways
- activate and change state of other parts of the protocol
- block parts of the protocol by never responding



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

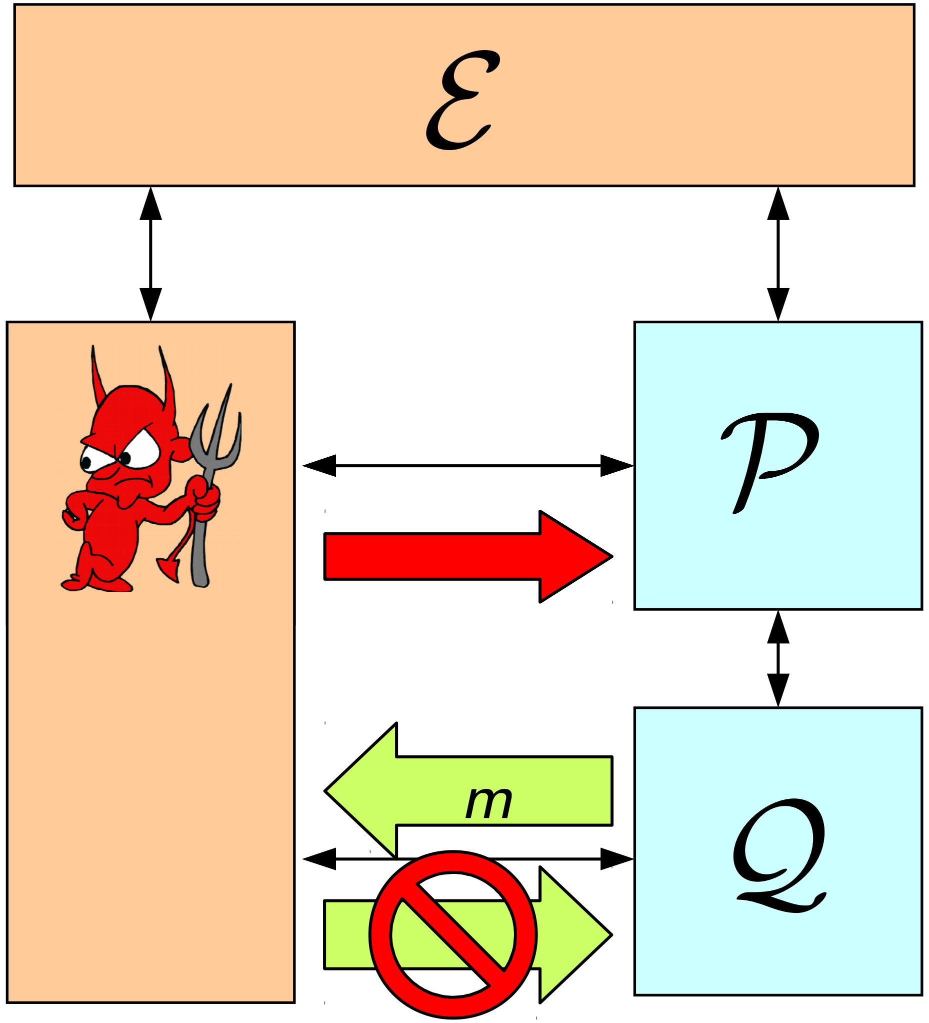
- activate the protocol in unexpected ways
- activate and change state of other parts of the protocol
- block parts of the protocol by never responding



IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

- activate the protocol in unexpected ways
- activate and change state of other parts of the protocol
- block parts of the protocol by never responding

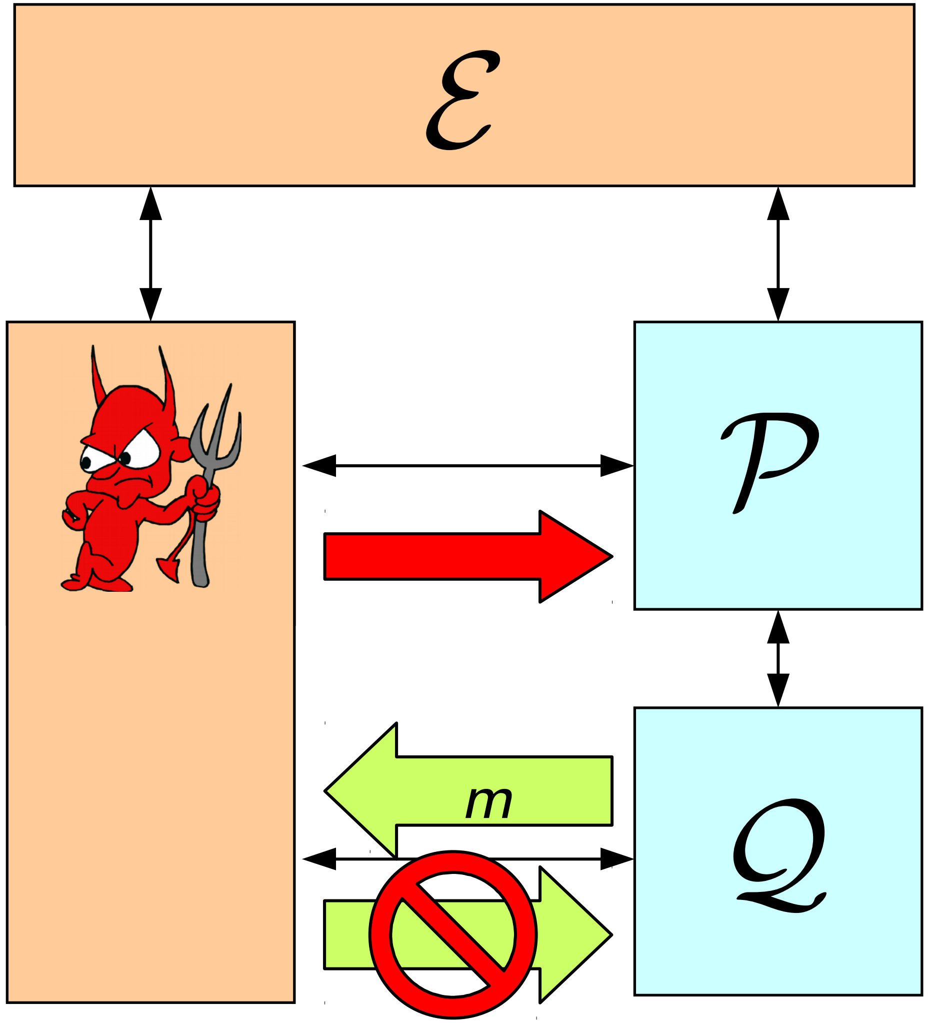


IITM Model With Responsive Environments

Instead of responding immediately to an urgent request, the adversary might:

- activate the protocol in unexpected ways
- activate and change state of other parts of the protocol
- block parts of the protocol by never responding

→ protocol designer has to deal with unintended behavior and artificial attacks!

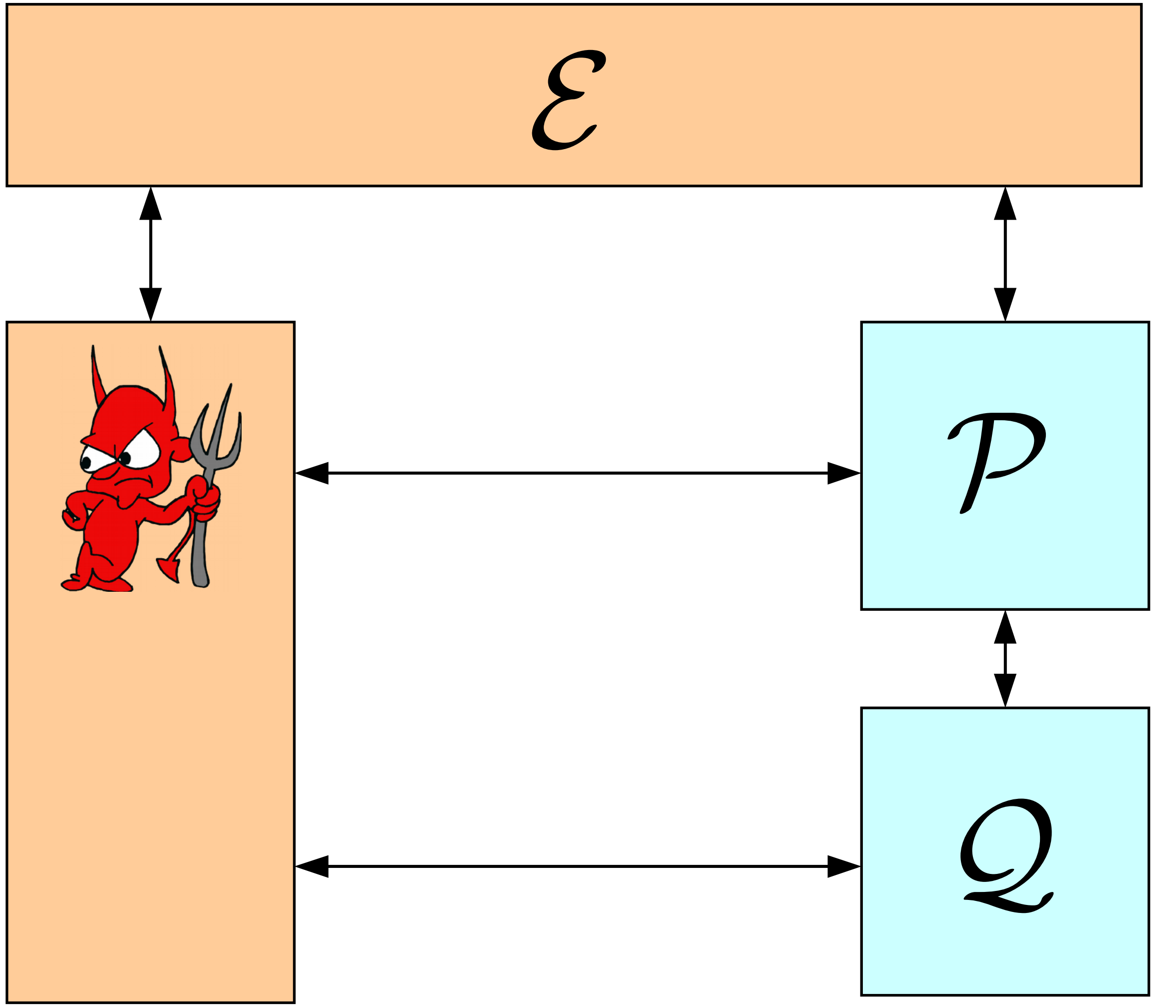


IITM Model With Responsive Environments

Responsive environments solve this situation as follows:

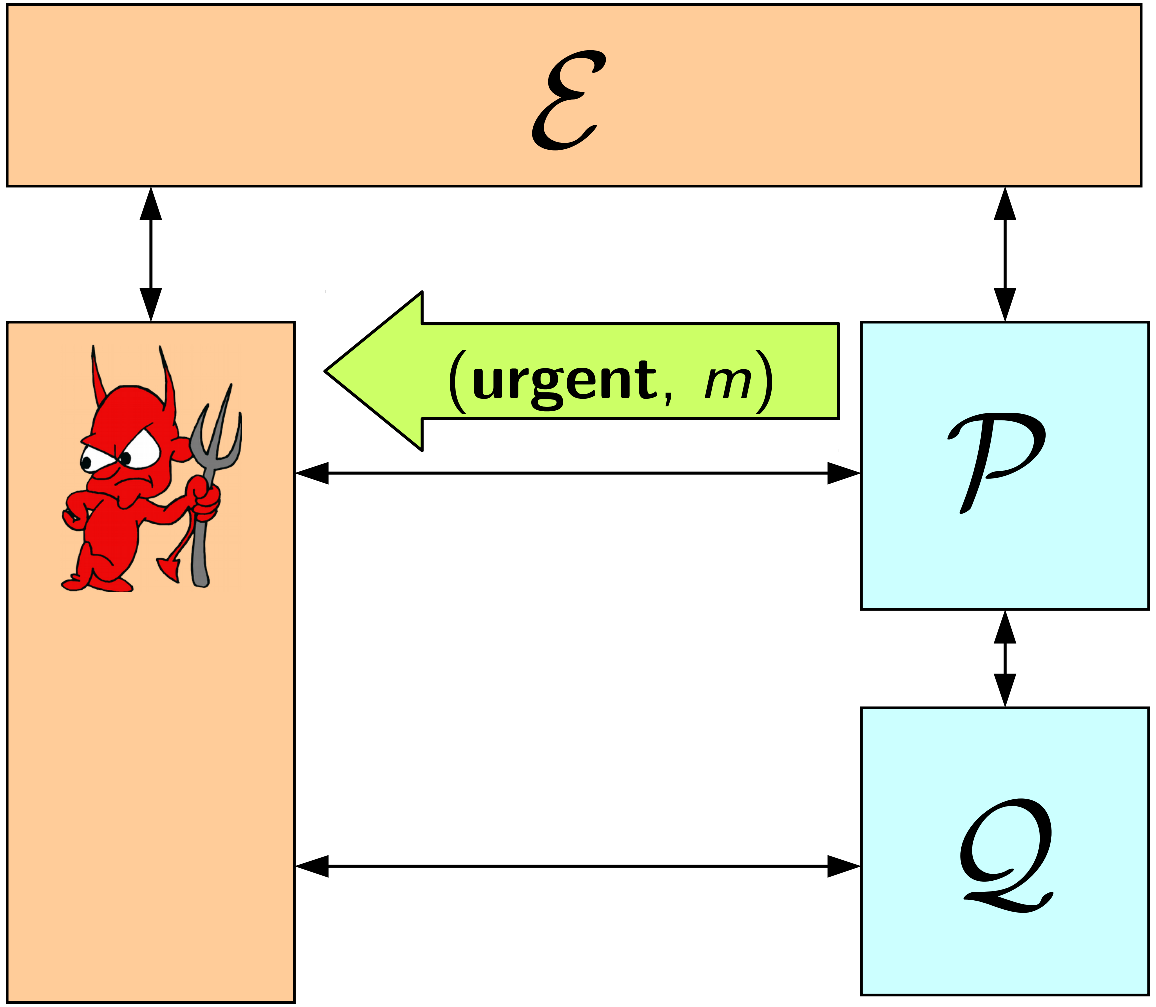
IITM Model With Responsive Environments

Responsive environments solve this situation as follows:



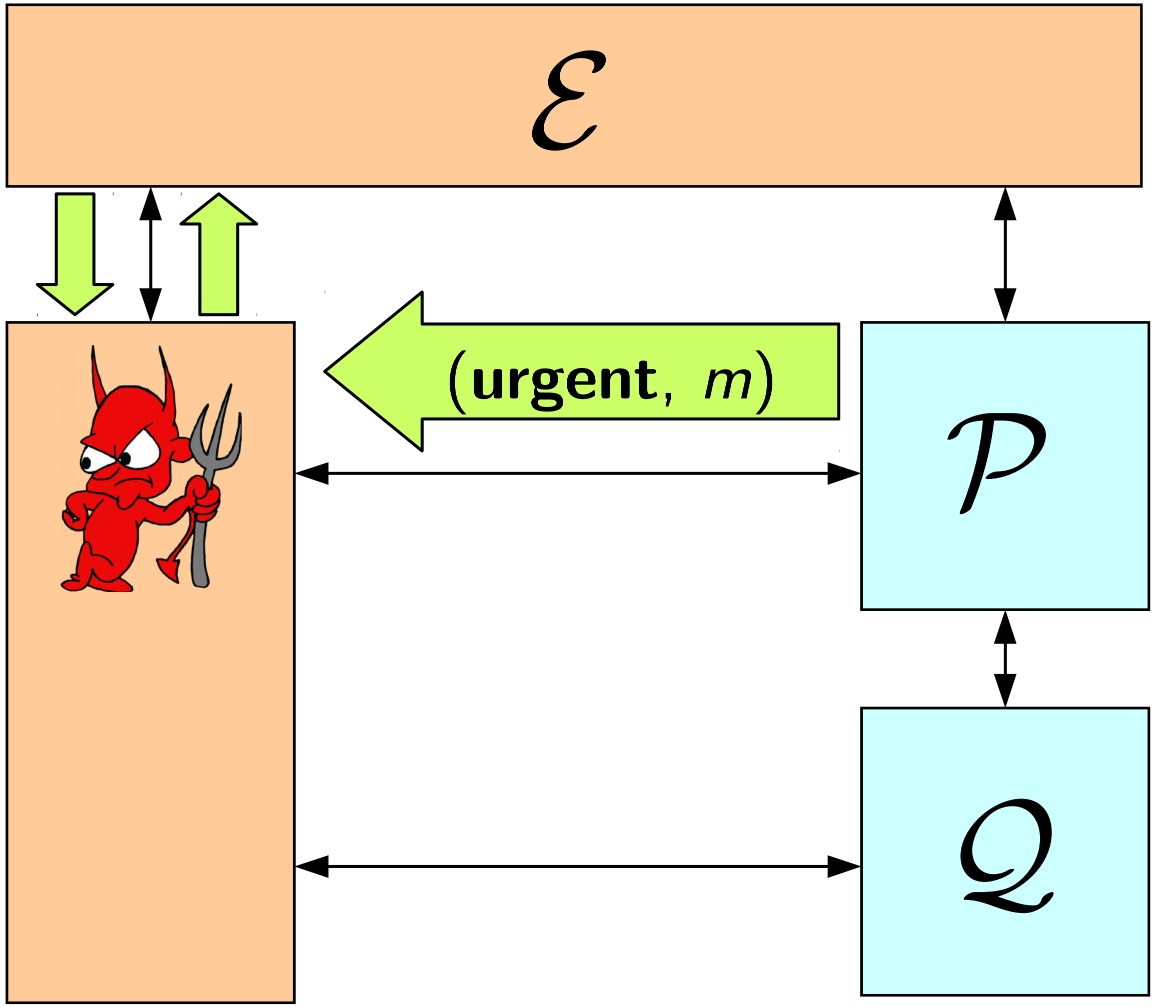
IITM Model With Responsive Environments

Responsive environments solve this situation as follows:



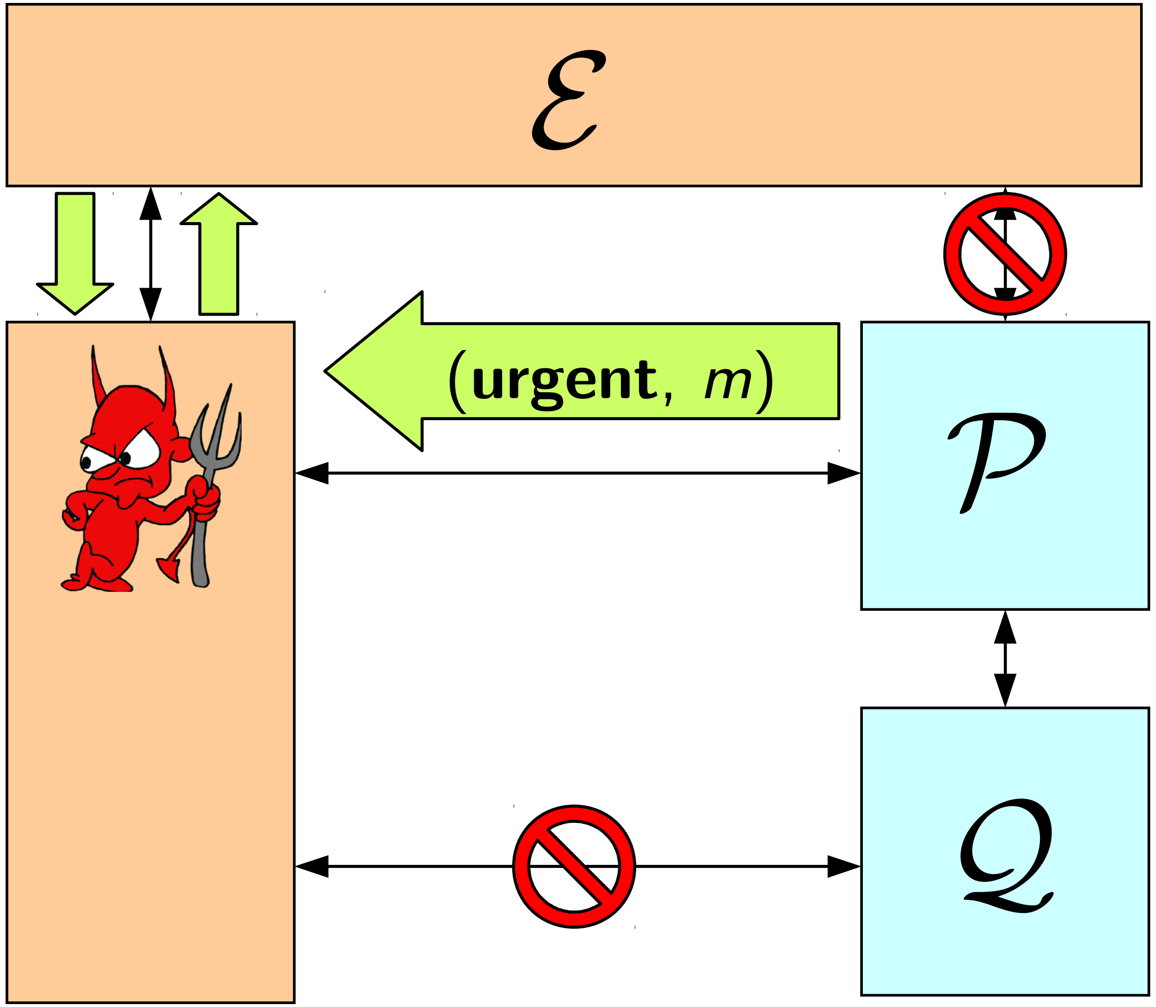
IITM Model With Responsive Environments

Responsive environments solve this situation as follows:



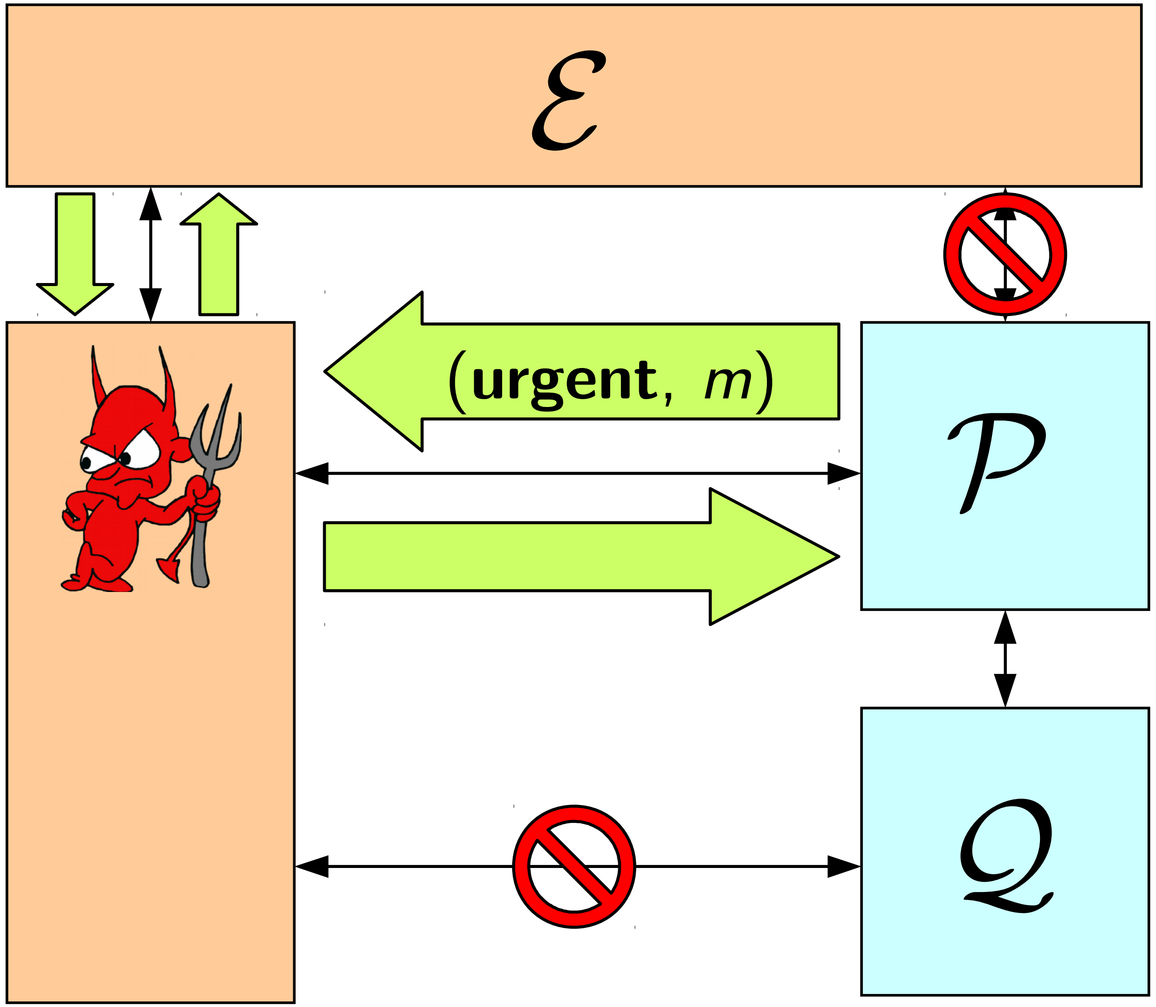
IITM Model With Responsive Environments

Responsive environments solve this situation as follows:



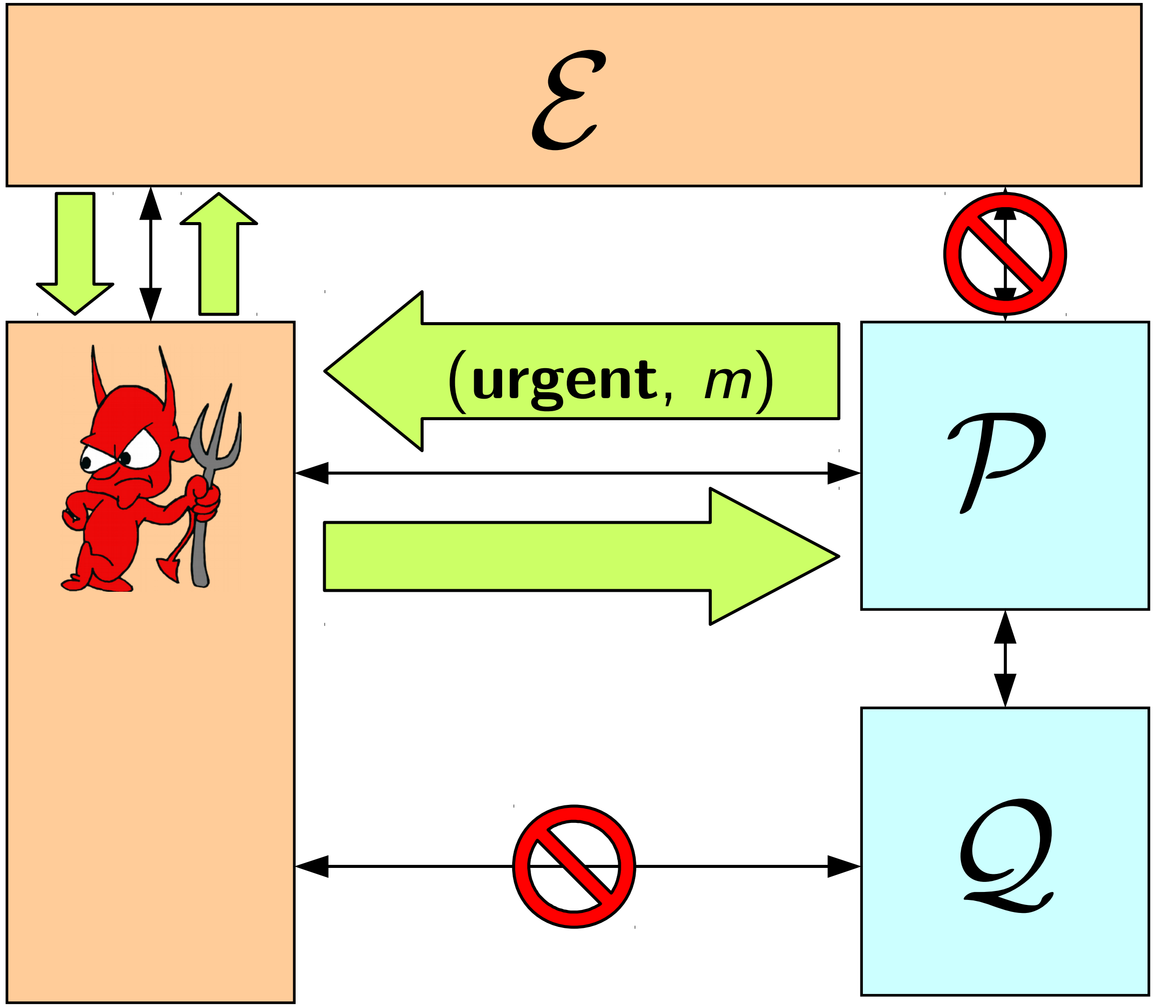
IITM Model With Responsive Environments

Responsive environments solve this situation as follows:



IITM Model With Responsive Environments

Responsive environments solve this situation as follows:



→ easy to use

→ simplifies both protocol modeling and analysis

Universal Composability Models

- UC model [Canetti 2001] 🚫
- GNUC model [Hofheinz, Shoup 2011] ∼
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016]
- ...

Ideally, a good model should be ...

- ➔ formally sound
- ➔ expressive
- ➔ easy to use

Universal Composability Models

- UC model [Canetti 2001] 🚫
- GNUC model [Hofheinz, Shoup 2011] ~
- IITM model [Küsters 2006]
  - IITM model with responsive environments [Küsters, Rausch 2016] ✓
- …

Ideally, a good model should be …

➔ formally sound

➔ expressive

➔ easy to use