The Wonderful World of Global Random Oracles

Jan Camenisch[1], Manu Drijvers[1,2], Tommaso Gagliardoni[1], Anja Lehmann[1], Gregory Neven[1]

[1] IBM Research – Zurich
[2] ETH Zurich

Random Oracles are still practical!

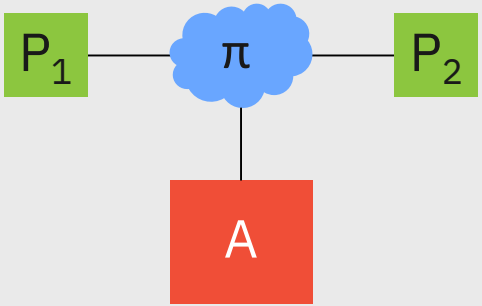
- Bellare & Rogaway ’93
- Model hash function as “random oracle”
  - On new input *m*, sample *h* uniformly from domain and output *h*

- Security proofs for many practical schemes
  - Signatures: Schnorr, RSA-FDH, RSA-PSS, …
  - Encryption: Fujisaki-Okamoto, …
  - Commitments: H(r, m)

- Canetti, Goldreich, Halevi: replacing RO with hash function does not guarantee security

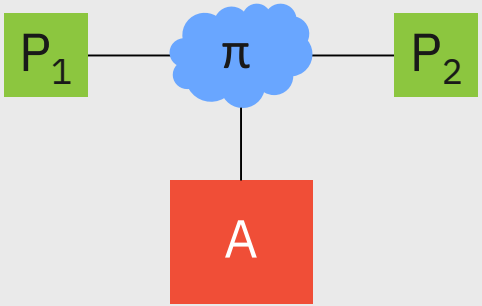
UC Framework (Canetti ‘00)

Real world

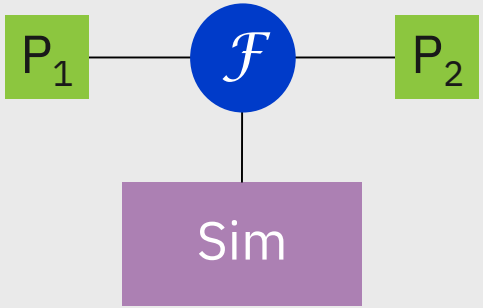


UC Framework (Canetti ‘00)

Real world

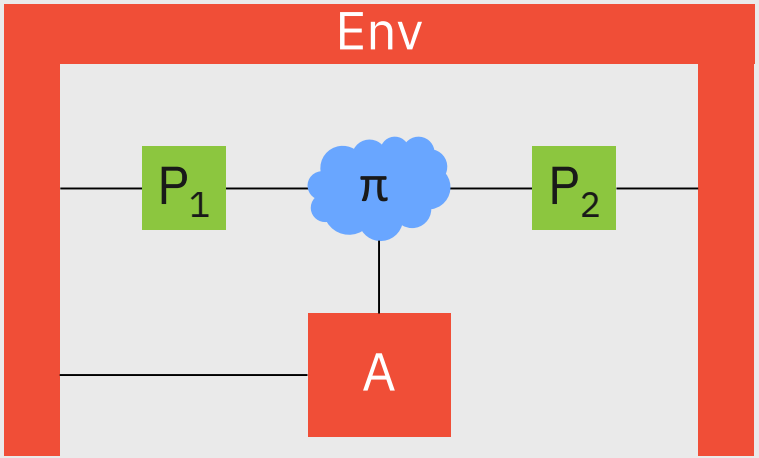


Ideal world



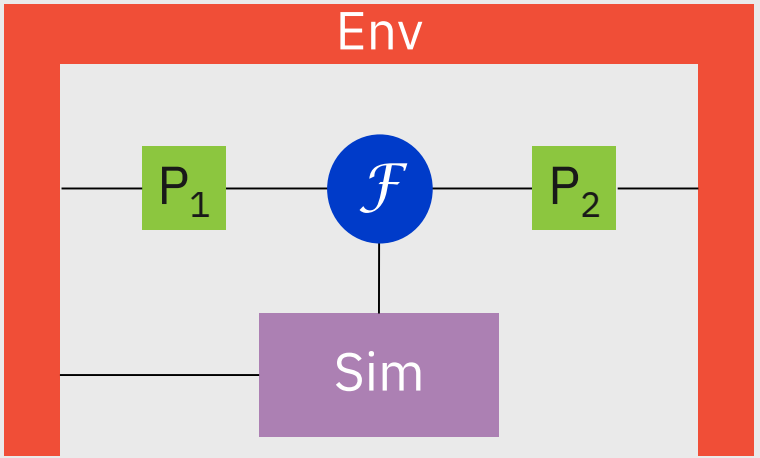
UC Framework (Canetti ‘00)

Real world

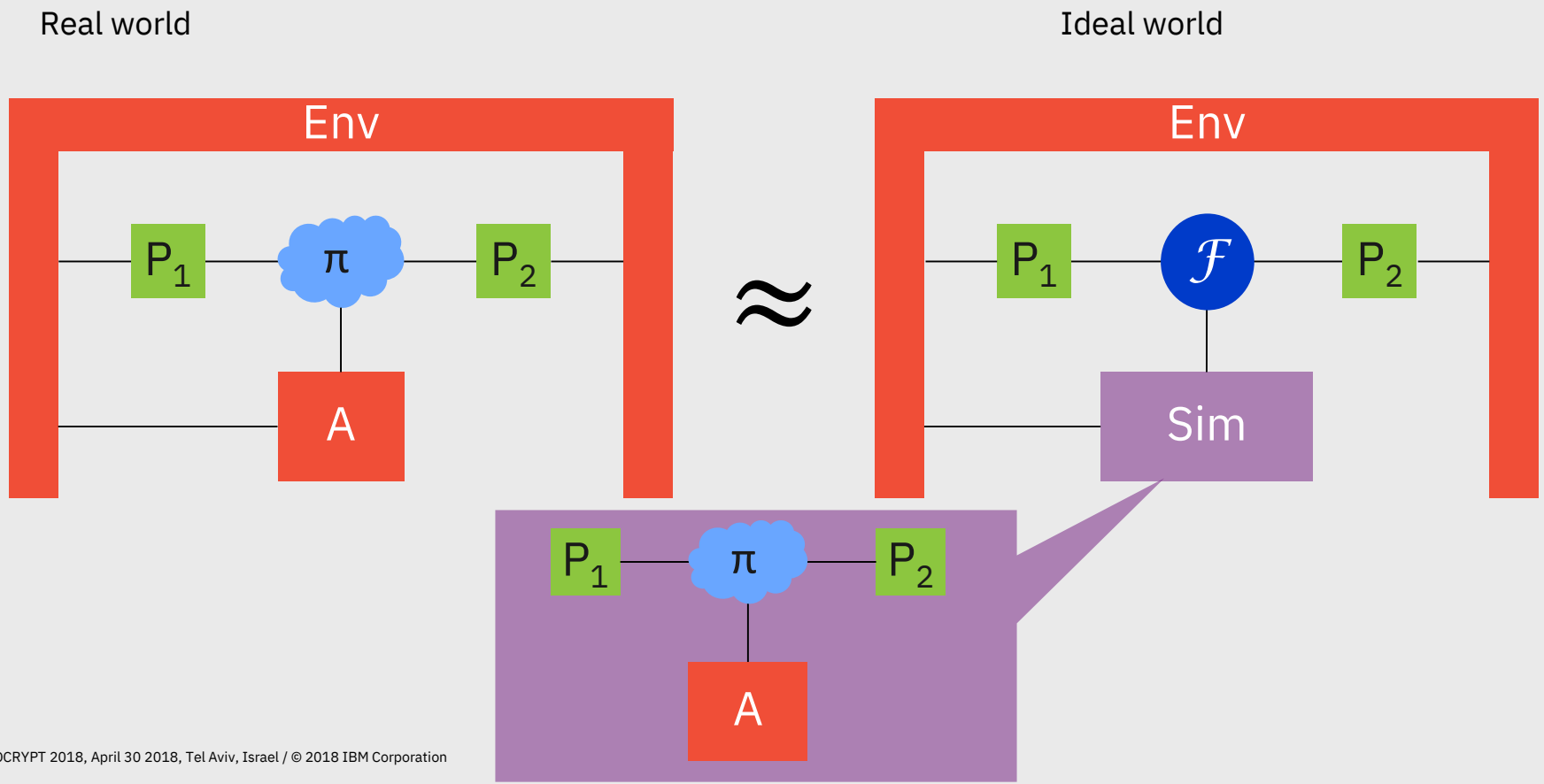


≈

Ideal world

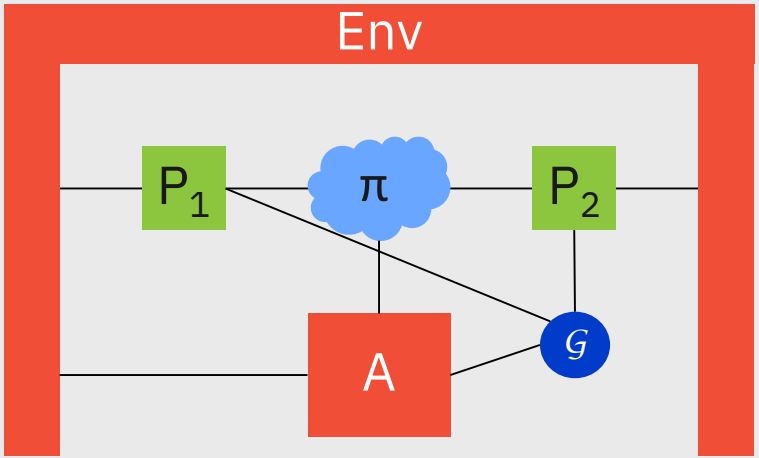


UC Framework (Canetti ‘00)



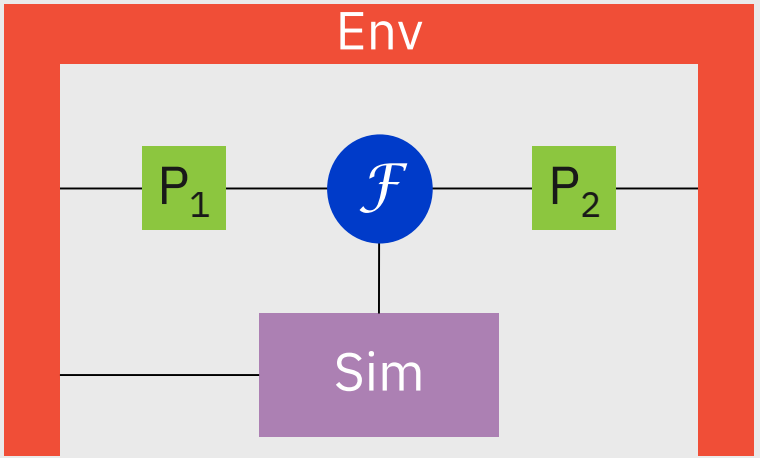
UC Framework (Canetti ‘00)

Real world with Hybrid $\mathcal{G}$



$\approx$

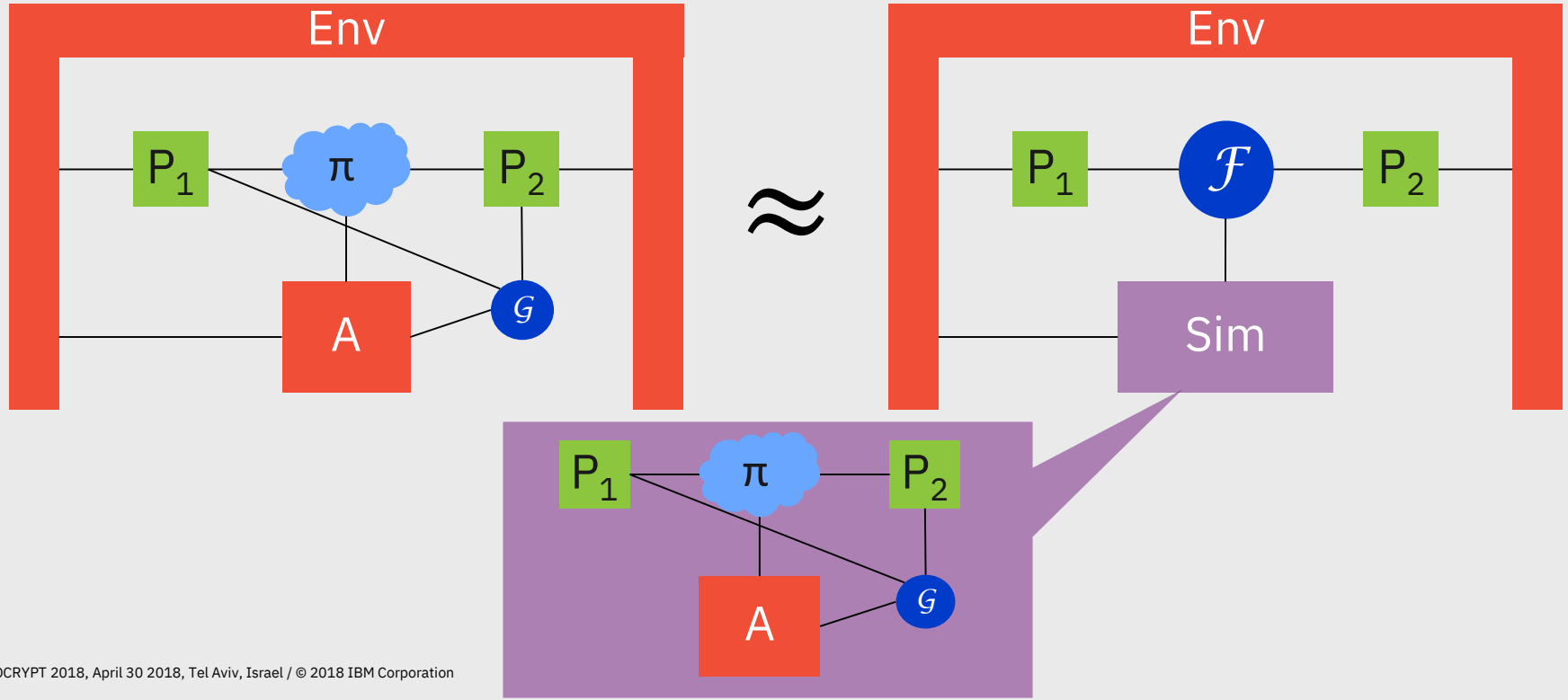
Ideal world



UC Framework (Canetti ‘00)

Real world with Hybrid $\mathcal{G}$

Ideal world

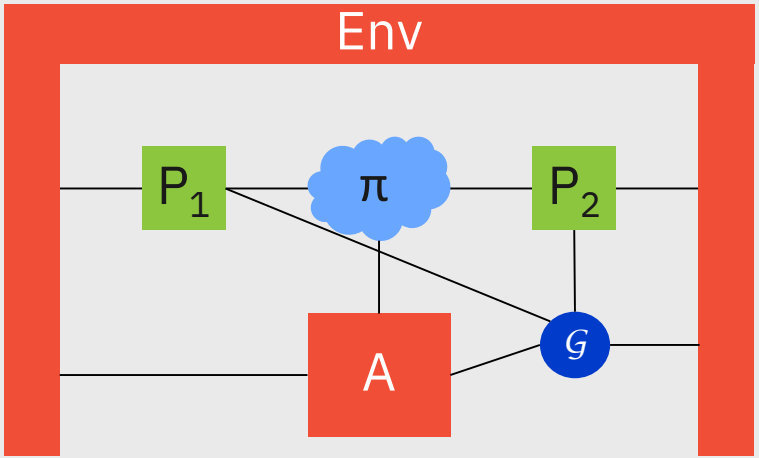


GUC Framework (Canetti, Dodis, Pass, Walfish ‘07)

Real world with **global** resource 

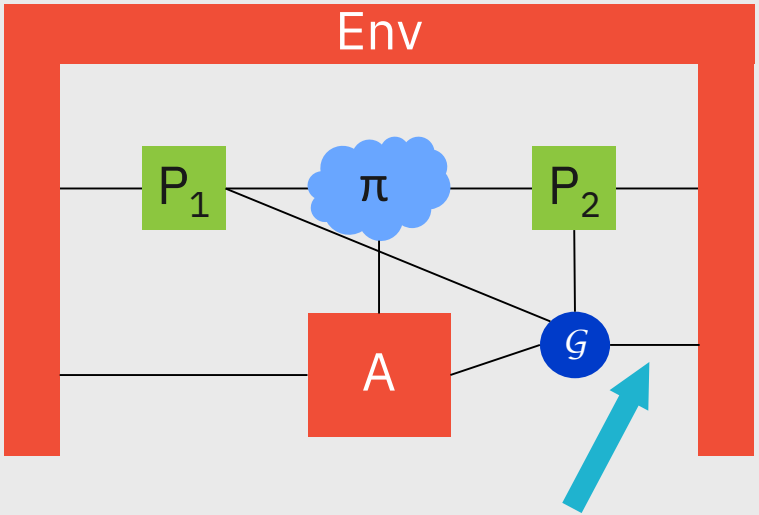
GUC Framework (Canetti, Dodis, Pass, Walfish ‘07)

Real world with **global** resource $\mathcal{G}$



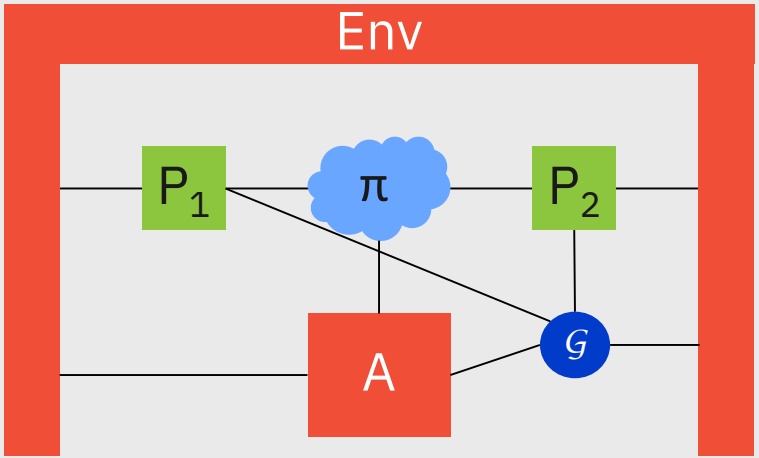
GUC Framework (Canetti, Dodis, Pass, Walfish ‘07)

Real world with **global** resource $\mathcal{G}$



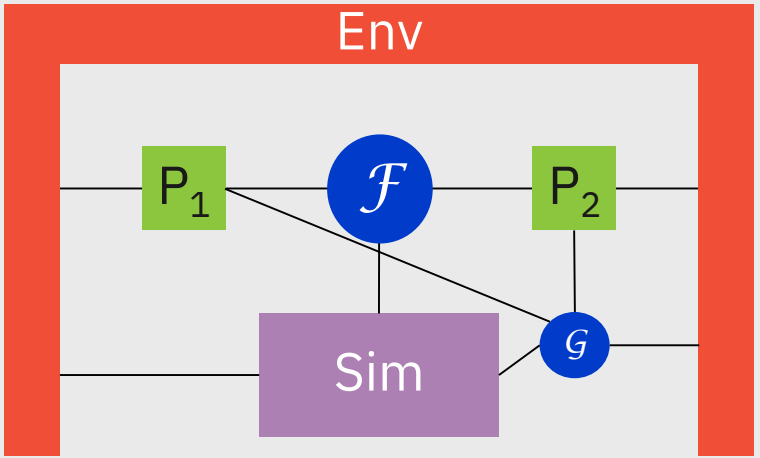
GUC Framework (Canetti, Dodis, Pass, Walfish ‘07)

Real world with **global** resource $\mathcal{G}$



≈

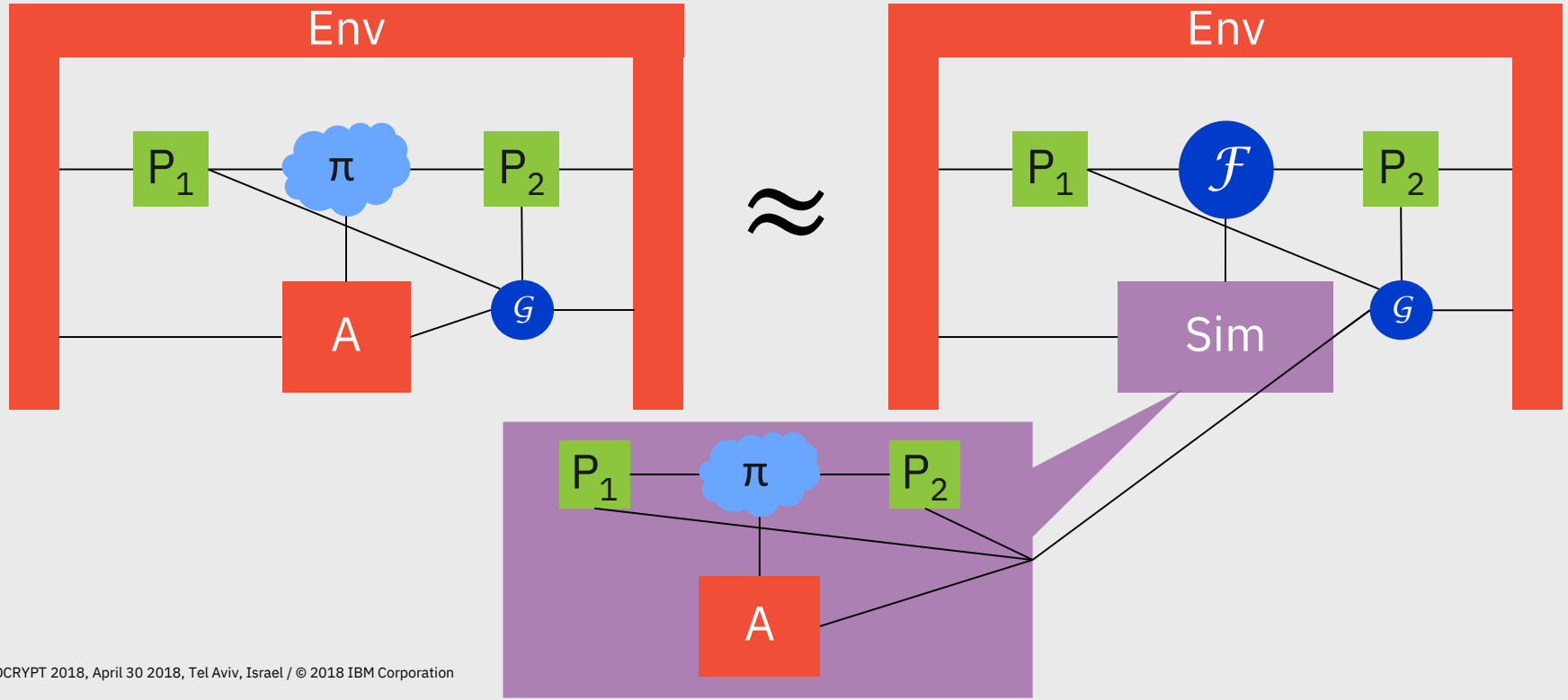
Ideal world with **global** resource $\mathcal{G}$



GUC Framework (Canetti, Dodis, Pass, Walfish '07)

Real world with **global** resource $\mathcal{G}$

Ideal world with **global** resource $\mathcal{G}$



Global Random Oracles in UC

- Random oracle should model single hash function
- Canetti, Jain, Scafuro ‘14: Global RO in GUC
  - UC commitment from observable global RO
  - Definition of RO is quite involved
  - Most efficient schemes cannot be proven secure in their model

- What can we do with a natural global RO?
- Can we prove efficient ROM schemes secure with global RO?

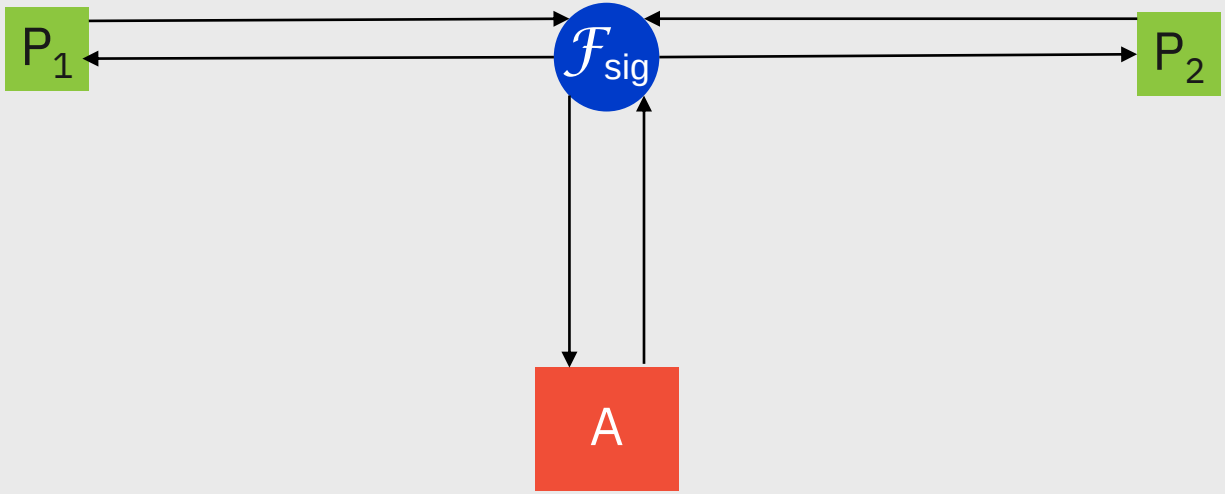
Global Random Oracles in UC

- Random oracle should model single hash function
- Canetti, Jain, Scafuro ‘14: Global RO in GUC
  - UC commitment from observable global RO
  - Definition of RO is quite involved
  - Most efficient schemes cannot be proven secure in their model

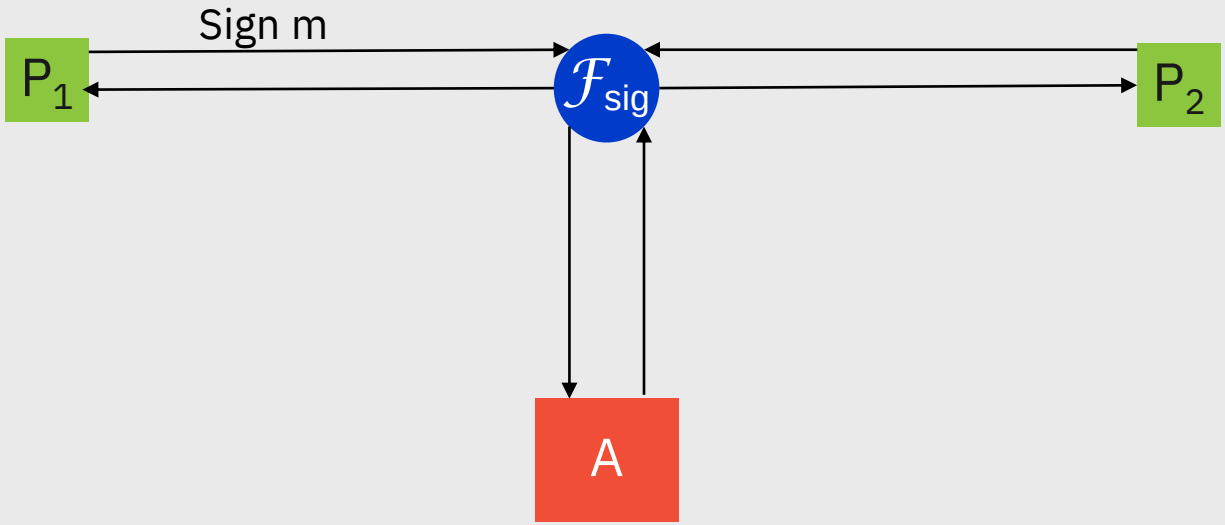
- What can we do with a natural global RO?
- Can we prove efficient ROM schemes secure with global RO?

First result: Any EUF-CMA signature scheme in the ROM is a UC secure signature scheme with a strict global RO

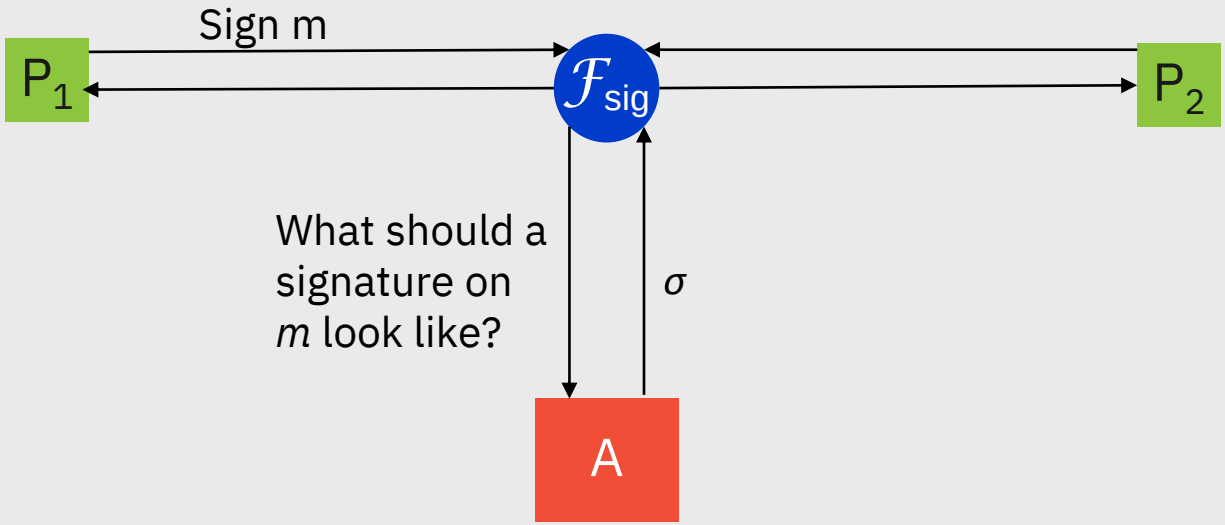
Signatures in UC



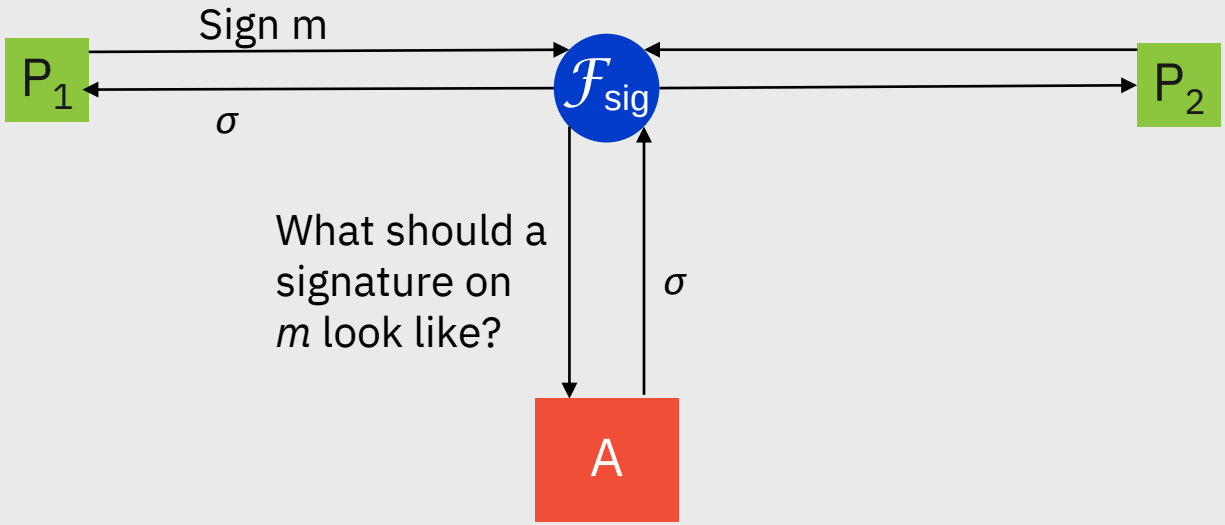
Signatures in UC



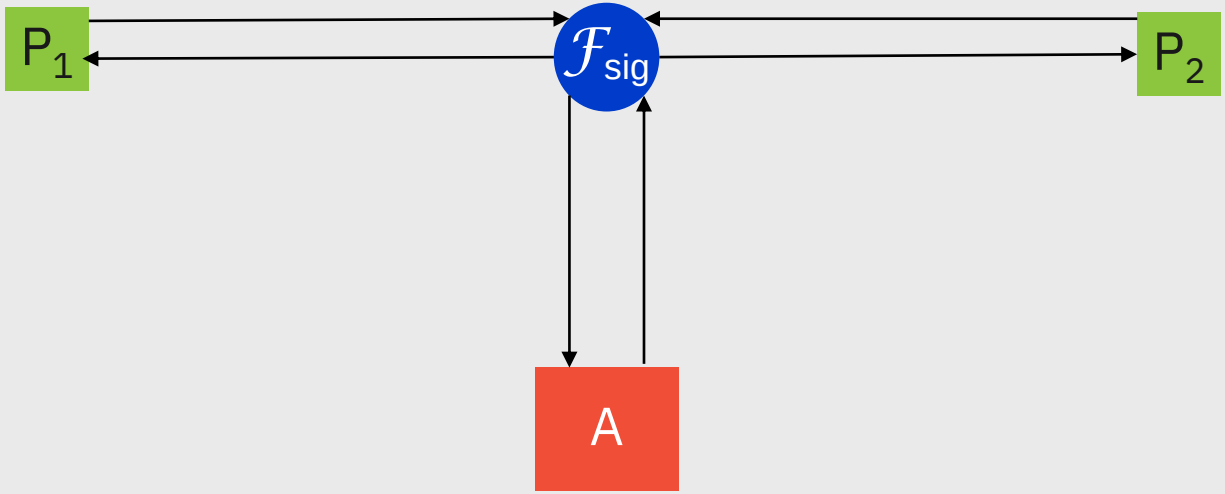
Signatures in UC



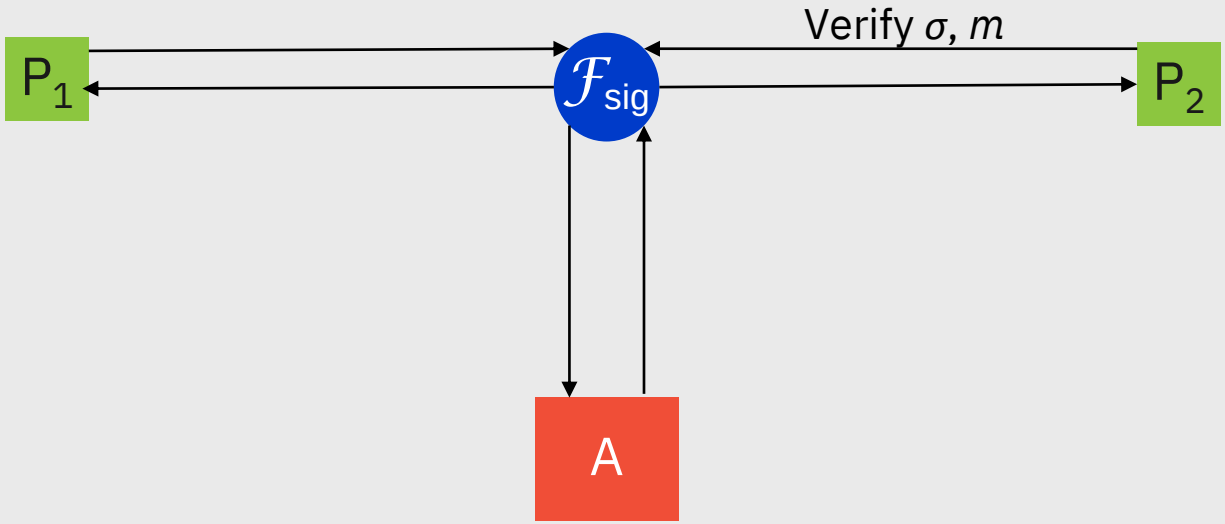
Signatures in UC



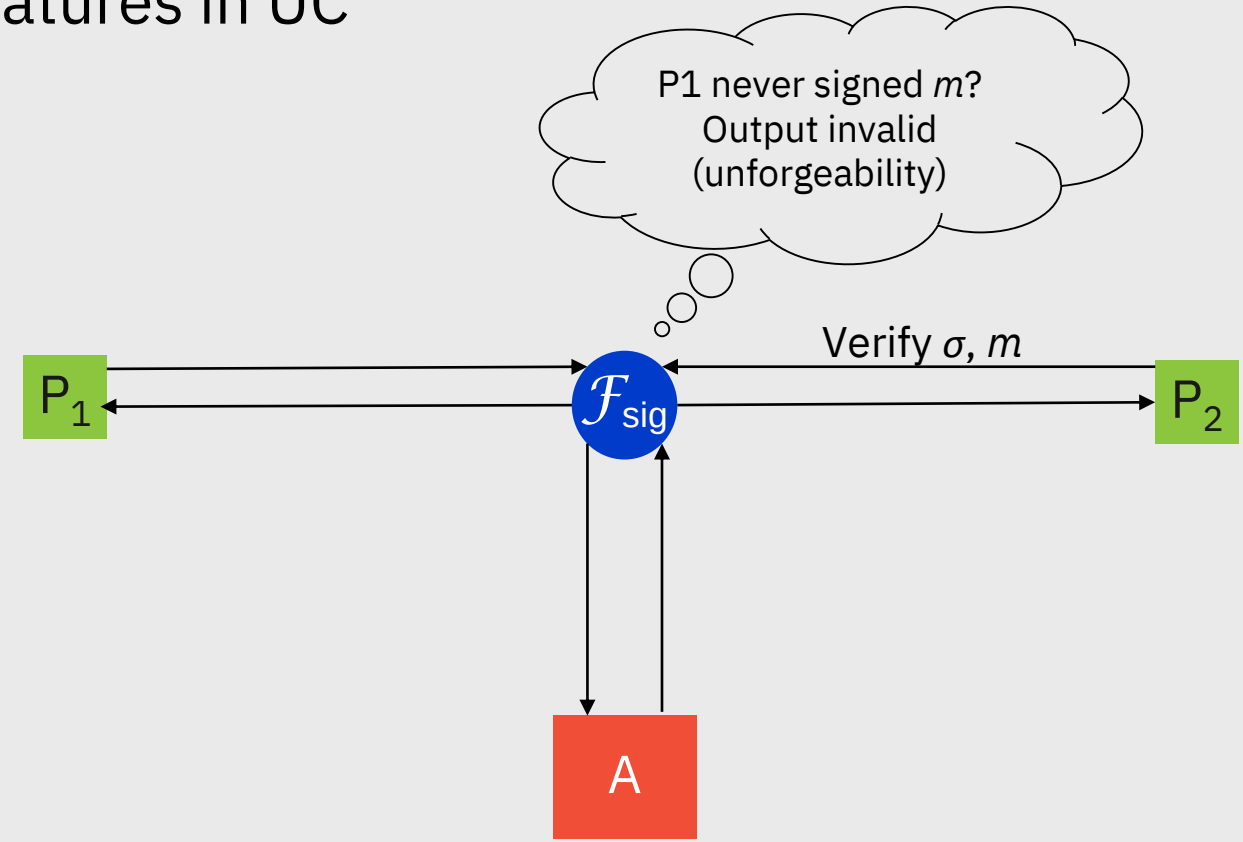
Signatures in UC



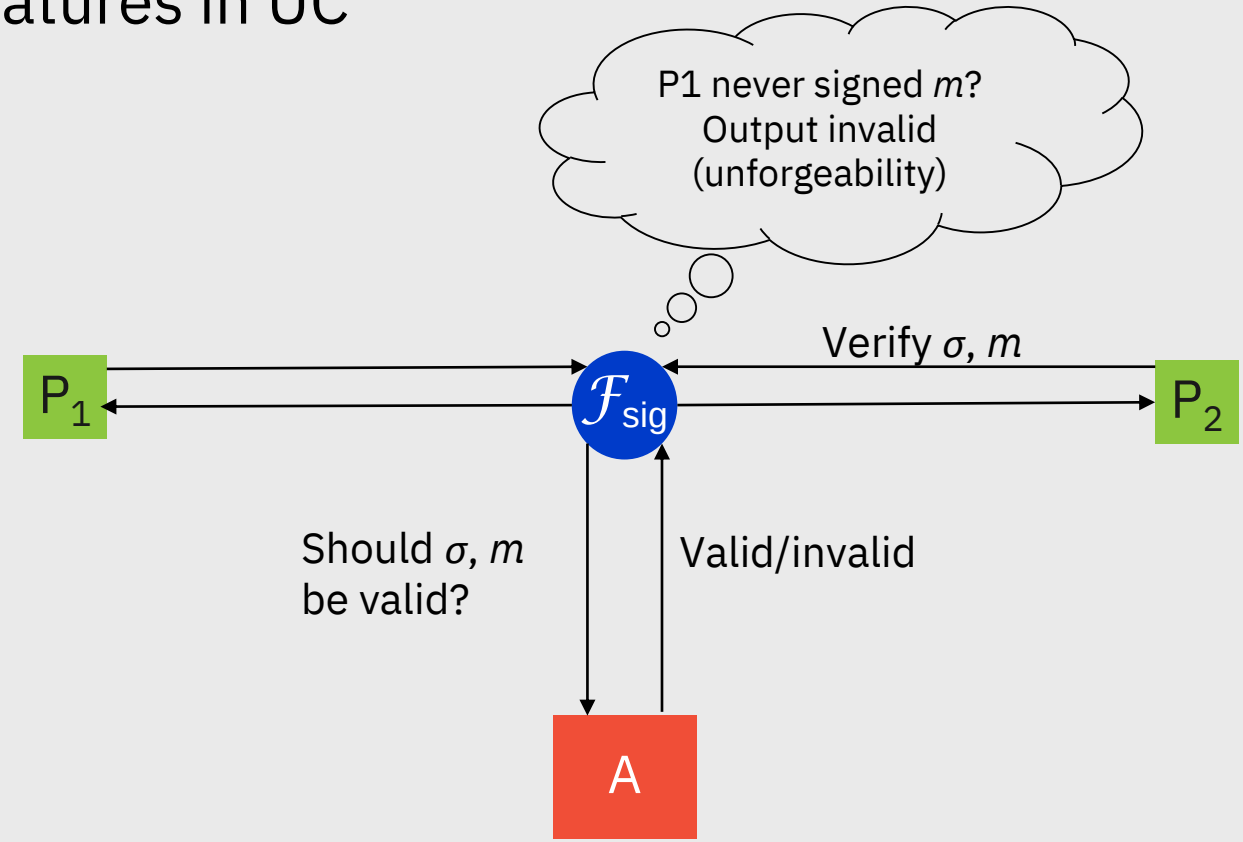
Signatures in UC



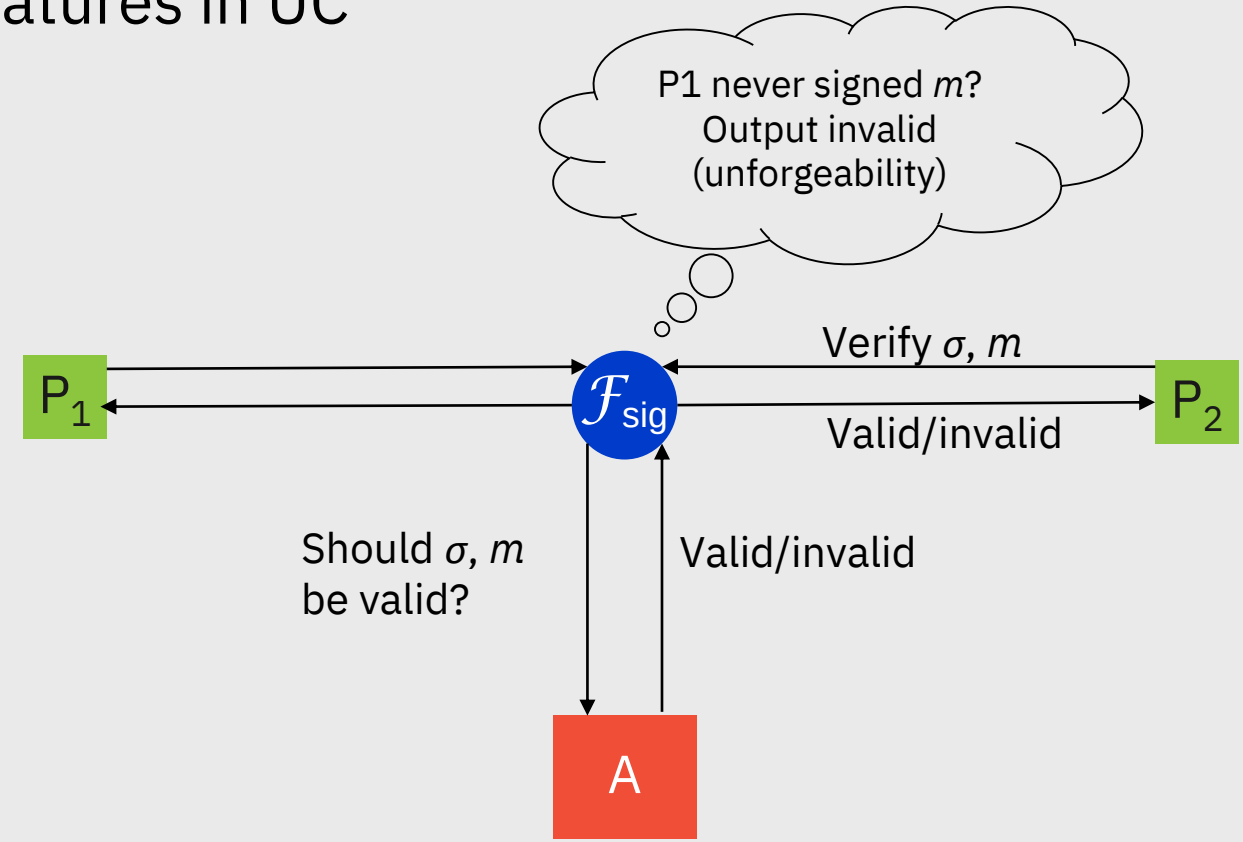
Signatures in UC



Signatures in UC

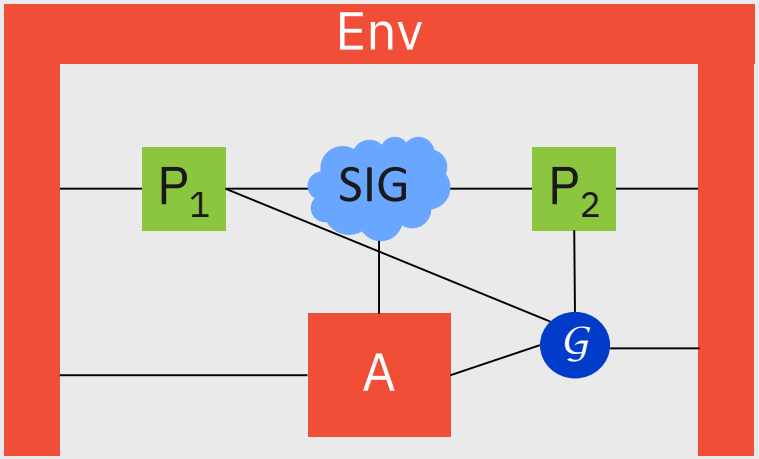


Signatures in UC



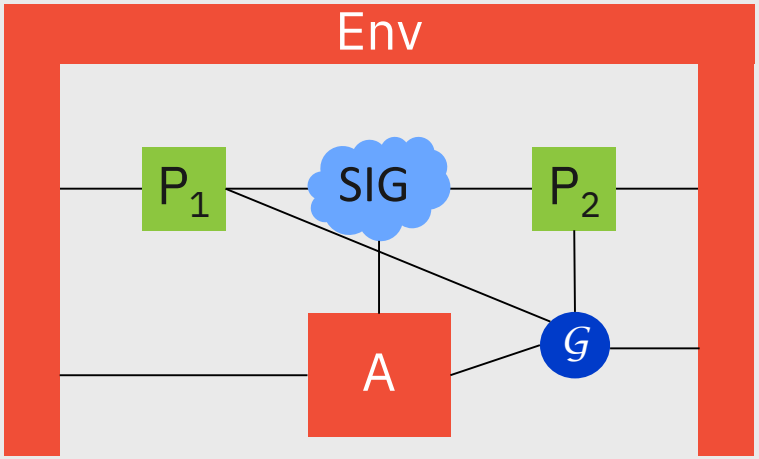
Signatures from strict global RO: simulator

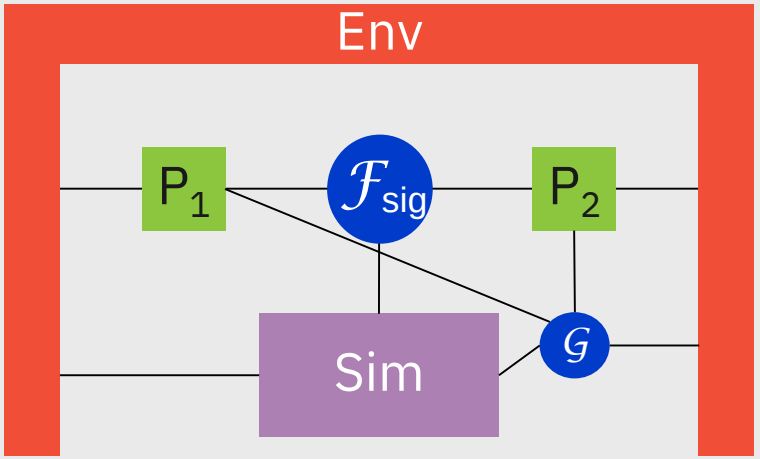
- Let SIG be EUF-CMA in the ROM



Signatures from strict global RO: simulator

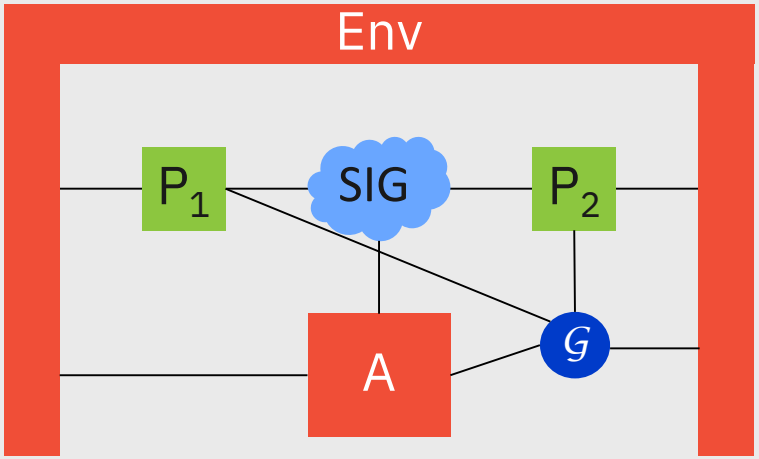
- Let SIG be EUF-CMA in the ROM

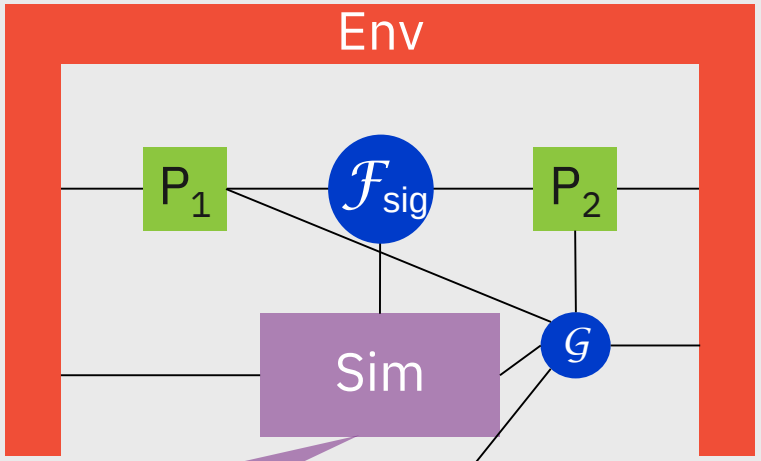


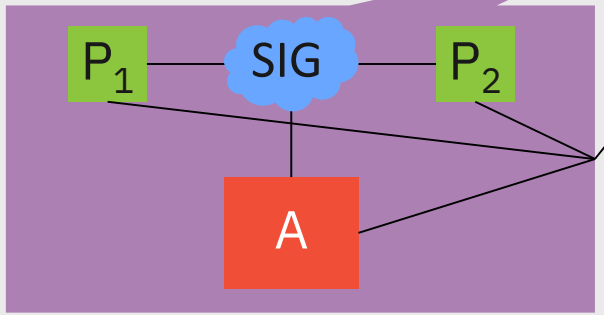


Signatures from strict global RO: simulator

- Let SIG be EUF-CMA in the ROM



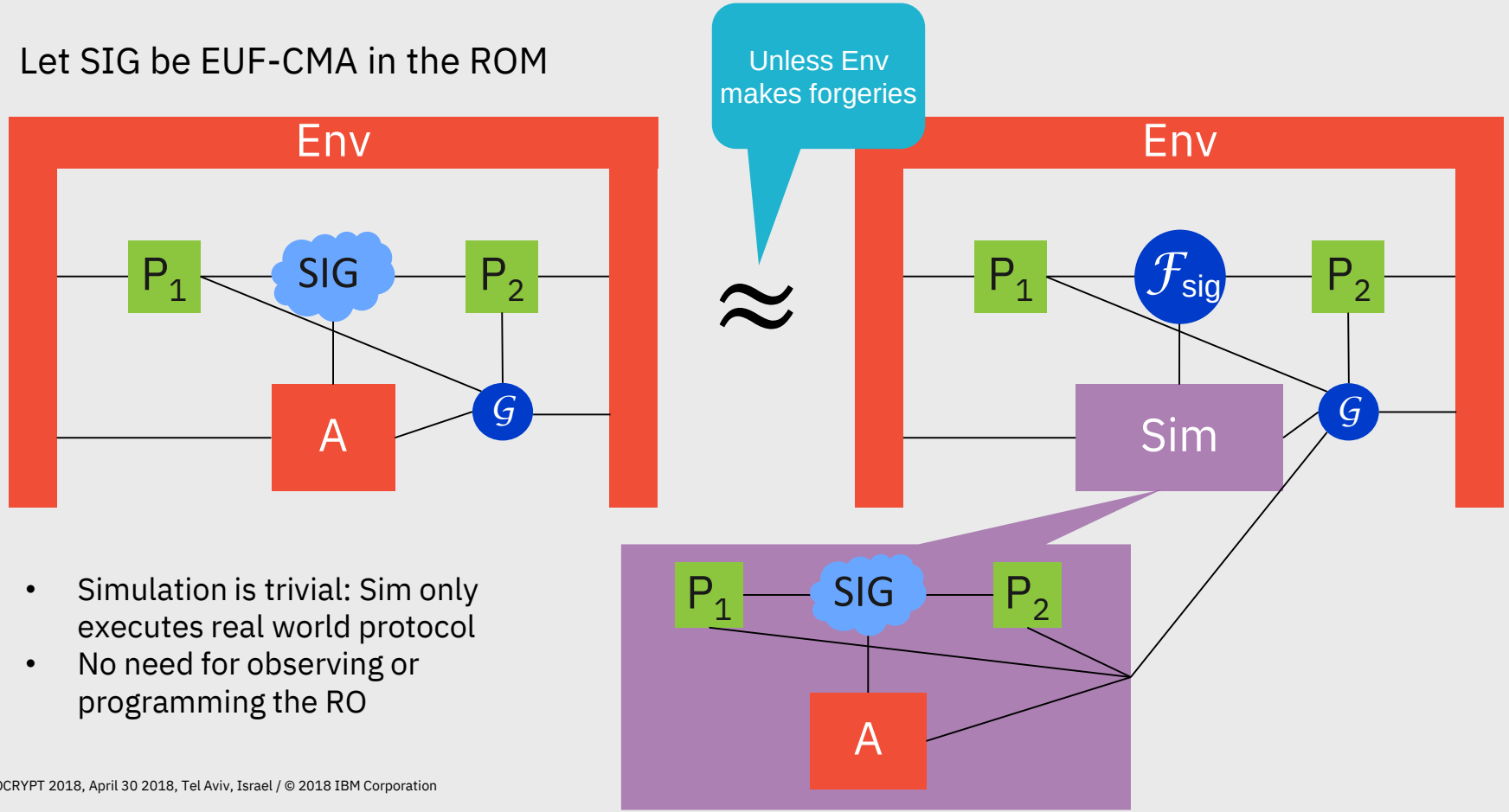




- Simulation is trivial: Sim only executes real world protocol
- No need for observing or programming the RO

Signatures from strict global RO: simulator

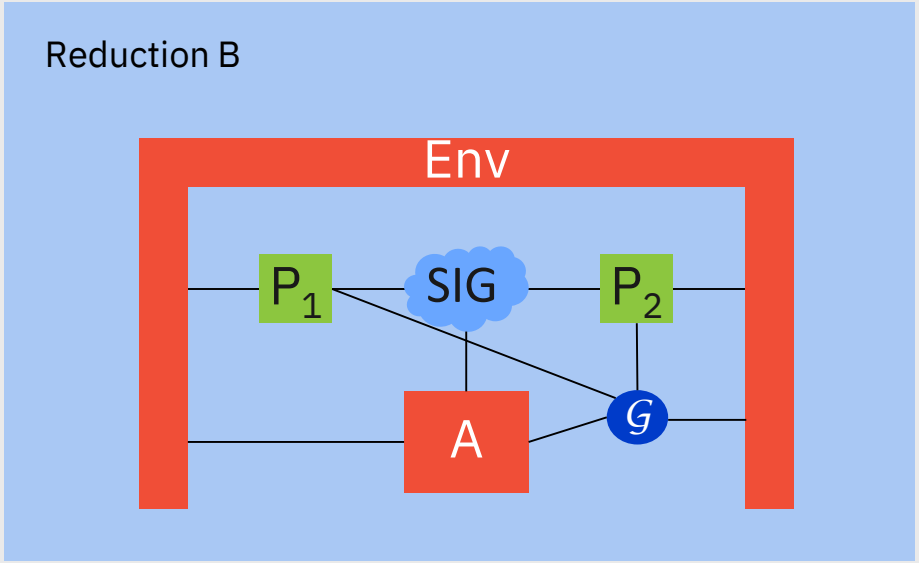
- Let SIG be EUF-CMA in the ROM



- Simulation is trivial: Sim only executes real world protocol
- No need for observing or programming the RO

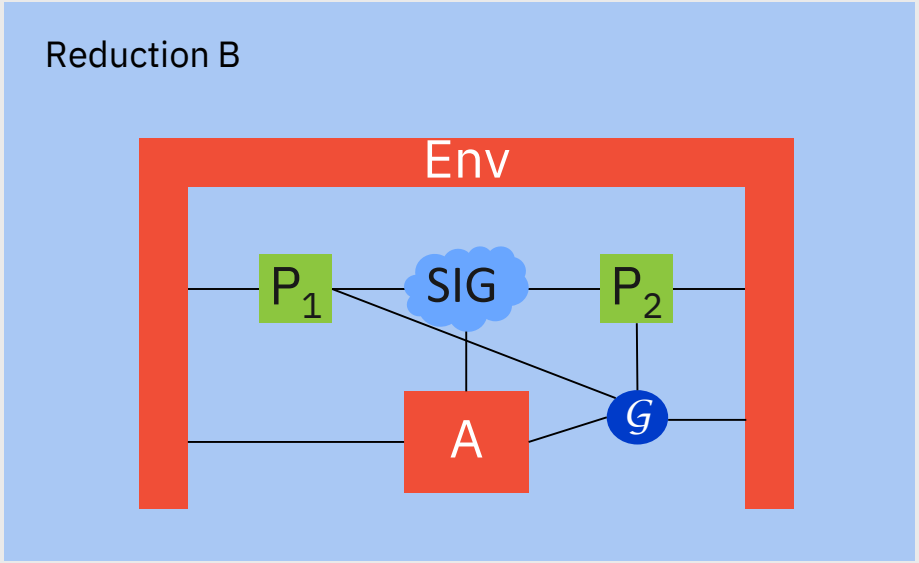
Proving indistinguishability

- Construct reduction B which simulates the protocol and RO towards Env
  - RO now simulated in reduction, meaning we have full control to observe and program
- B breaks EUF-CMA of SIG



Proving indistinguishability

- Construct reduction B which simulates the protocol and RO towards Env
  - RO now simulated in reduction, meaning we have full control to observe and program
- B breaks EUF-CMA of SIG



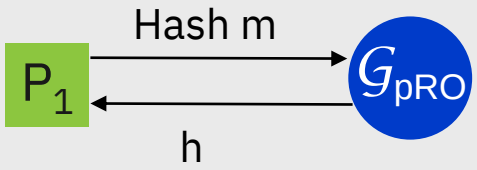
- This works, because Sim does not need observability/programmability
- Similar results for CCA2-PKE in ROM

• Adaptively secure PKE (non-committing encryption)

- Simulator needs programmability (Nielsen, Crypto ‘02)
- Define programmable global RO with explicit “program” interface
- Camenisch et al. (CSF ‘17) secure with programmable global RO

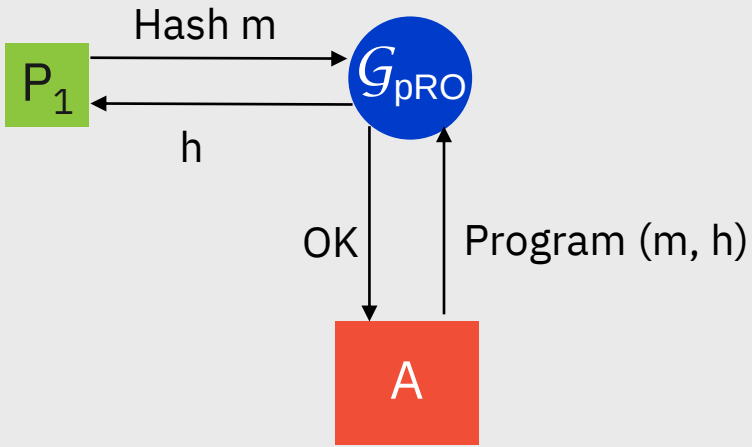
• Adaptively secure PKE (non-committing encryption)

- Simulator needs programmability (Nielsen, Crypto ‘02)
- Define programmable global RO with explicit “program” interface
- Camenisch et al. (CSF ‘17) secure with programmable global RO



• Adaptively secure PKE (non-committing encryption)

- Simulator needs programmability (Nielsen, Crypto ‘02)
- Define programmable global RO with explicit “program” interface
- Camenisch et al. (CSF ‘17) secure with programmable global RO



UC Commitments

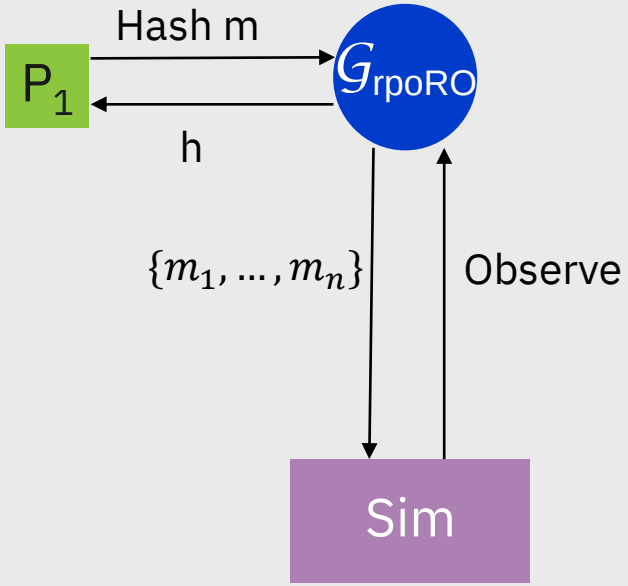
- Simulator needs extra power (Canetti, Fischlin 01)
- CJS14
  - “Restricted” Observable RO
  - 2 round commit and 2 round opening
- What about $c = H(r, m)$

- Extend Restricted observable RO with restricted programmability
  - Folklore commitment scheme secure in this model

UC Commitments

- Simulator needs extra power (Canetti, Fischlin 01)
- CJS14
  - "Restricted" Observable RO
  - 2 round commit and 2 round opening
- What about $c = H(r, m)$

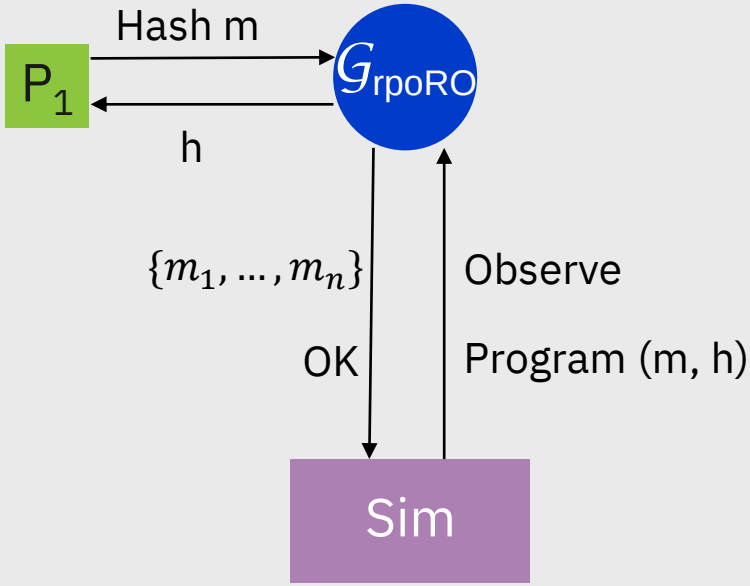
- Extend Restricted observable RO with restricted programmability
  - Folklore commitment scheme secure in this model



UC Commitments

- Simulator needs extra power (Canetti, Fischlin 01)
- CJS14
  - "Restricted" Observable RO
  - 2 round commit and 2 round opening
- What about $c = H(r, m)$

- Extend Restricted observable RO with restricted programmability
  - Folklore commitment scheme secure in this model



Unifying different global Random Oracles

Unifying different global Random Oracles

Strict GRO
Signatures,
PKE

Unifying different global Random Oracles

Programmable GRO
Adaptively secure
PKE (NCE)

Strict GRO
Signatures,
PKE

Unifying different global Random Oracles

Restricted Observable GRO (CJS14)
Interactive commitments

Programmable GRO
Adaptively secure PKE (NCE)

Strict GRO
Signatures, PKE

Unifying different global Random Oracles

Restricted programmable & observable GRO
Non-interactive commitments
$c = H(r, m)$

Restricted Observable GRO (CJS14)
Interactive commitments

Programmable GRO
Adaptively secure PKE (NCE)

Strict GRO
Signatures, PKE

Unifying different global Random Oracles

Restricted programmable & observable GRO
Non-interactive commitments
$c = H(r, m)$

Restricted programmable GRO
Interactive commitments

Restricted Observable GRO (CJS14)
Interactive commitments

Programmable GRO
Adaptively secure PKE (NCE)

Strict GRO
Signatures, PKE

Unifying different global Random Oracles

Stronger assumption,
more efficient schemes

Restricted programmable & observable GRO
Non-interactive commitments
$c = H(r, m)$

Restricted programmable GRO
Interactive commitments

Restricted Observable GRO (CJS14)
Interactive commitments

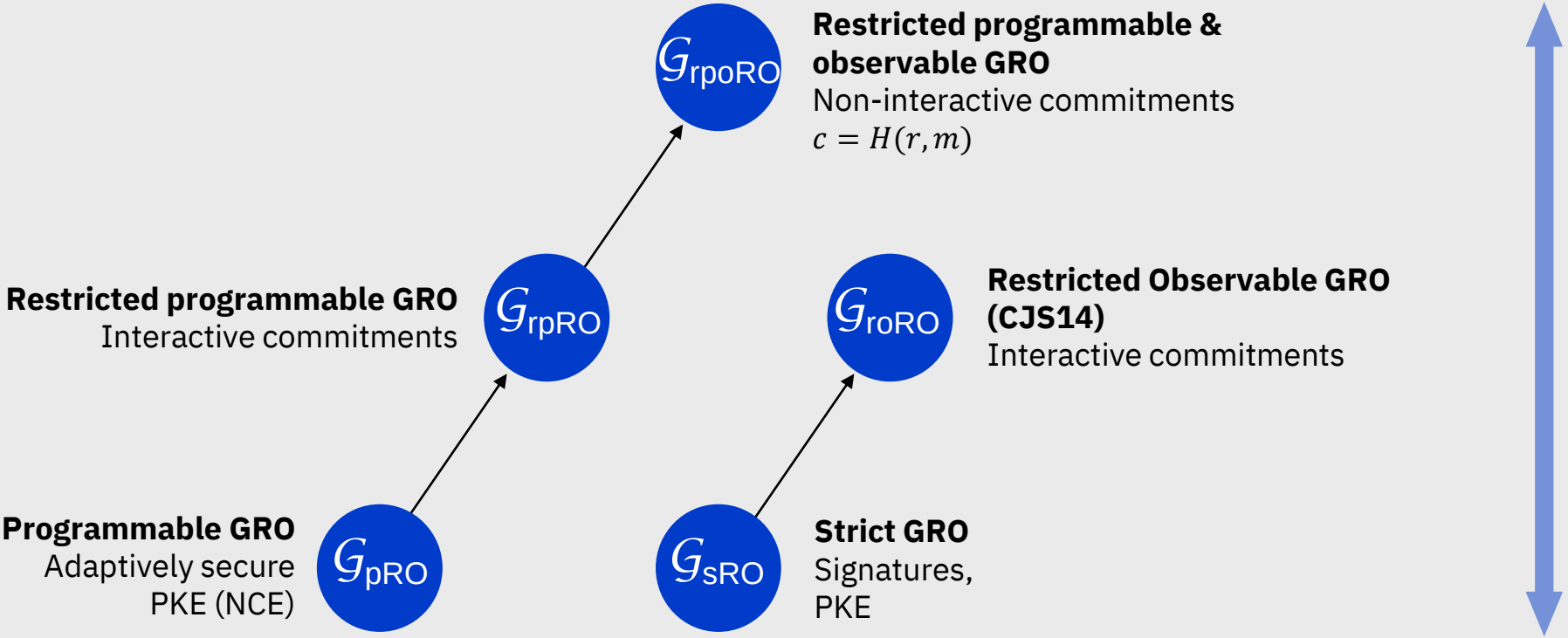
Programmable GRO
Adaptively secure
PKE (NCE)

Strict GRO
Signatures,
PKE

More natural modeling
of random oracle

Unifying different global Random Oracles

Stronger assumption,
more efficient schemes



More natural modeling
of random oracle

Conclusion

- Random oracles should be modeled as **global** resources

- Many of the most efficient primitives can be proven secure with global random oracles

Conclusion

- Random oracles should be modeled as **global** resources

- Many of the most efficient primitives can be proven secure with global random oracles

Thanks!

ia.cr/**2018/165**