On Actively-Secure Elementary MPC Reductions

Benny Applebaum

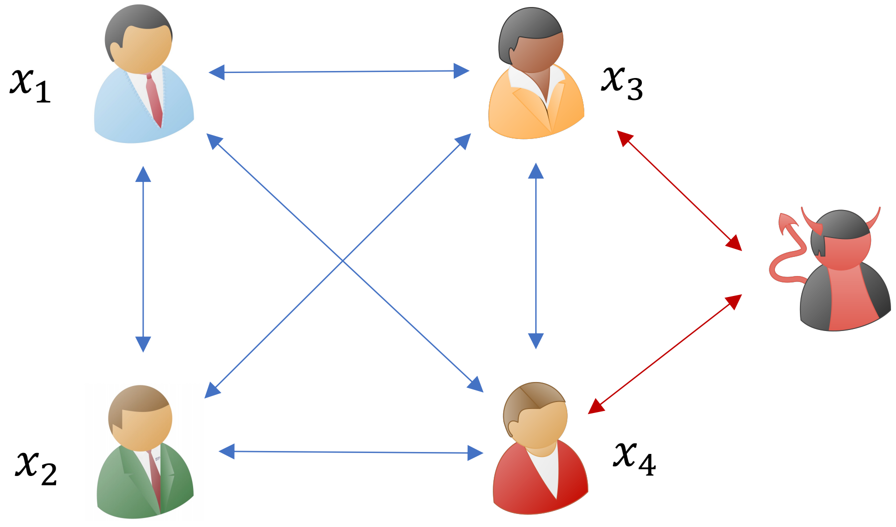
TEL AVIV UNIVERSITY

Aarushi Goel

JOHNS HOPKINS UNIVERSITY

TCC 2021

Secure Multiparty Computation (MPC)



Adversary learns nothing beyond the output y

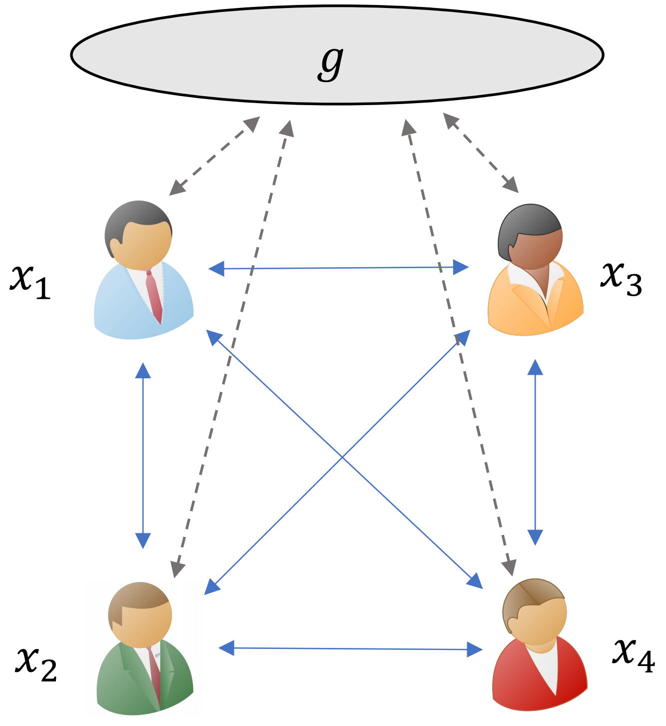
MPC protocol for computing $y = f(x_1, x_2, x_3, x_4)$

Secure MPC Reduction

Securely computing a simpler functionality g

Reduction

Securely computing a function f



A secure protocol for computing f assuming parties have oracle access to g.

Given such a reduction, we only need to design a secure protocol for the simpler functionality g.

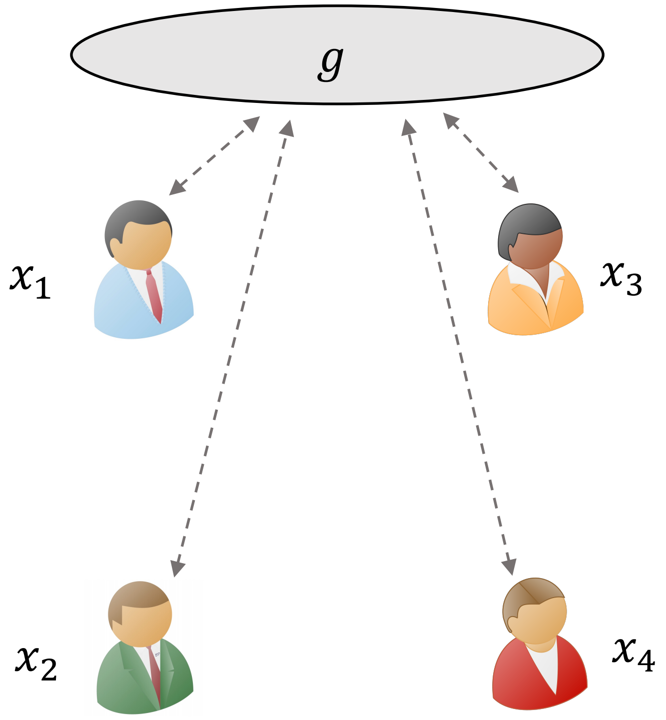
Classical Examples: [Yao'86, GMW'90] show such a secure reduction from any polynomial function to a two-party OT functionality

Non-interactive MPC Reduction

Securely computing a simpler functionality g

Non-interactive Reduction

Securely computing a function f



A non-interactive secure protocol for computing f assuming parties have oracle access to g.

Parties only make a single call to oracle g, but do not talk to each other.

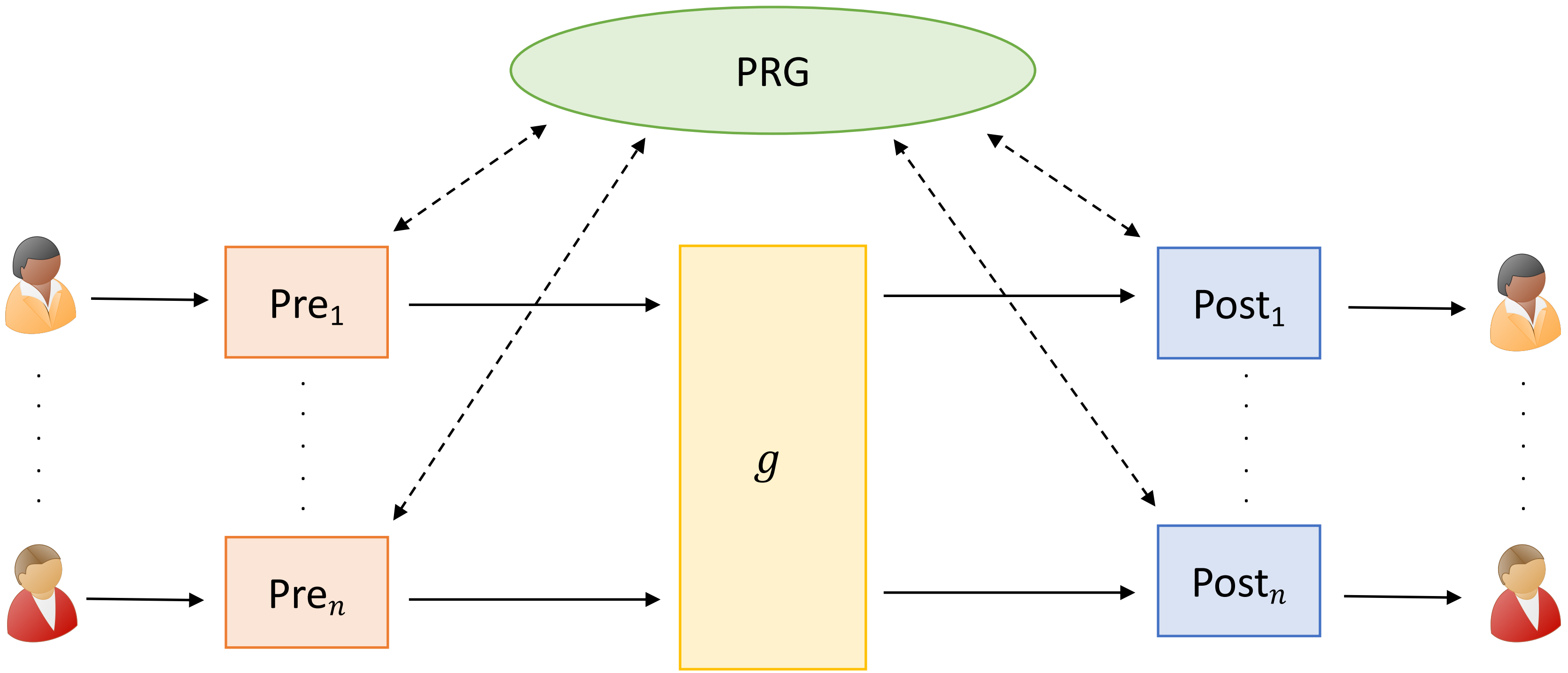
- Functionality g is allowed to have internal randomness
- There exists a general non-interactive reduction from such functionalities to deterministic ones [IK'02, AIK'04].

Elementary MPC Reduction

A non-interactive reduction is elementary if

- g is a constant degree functionality
- A PRG can only be used by the reduction in a black–box way
- g is independent of PRG

Elementary Reduction



g is a constant degree functionality and is independent of PRG

Elementary Reduction

Result	Corruption	Functions	Security	Security
[Yao'86, BMR'90]	$t < n$	P/Poly	Passive	Full Security
[DI'05]	$t < n/2$	P/Poly	Active	Full Security
[IK'00]	$t < n$	NC^1	Active (IT)	Full Security
[IPS'08, LPSY'15]	$t < n$	P/Poly	Active	Security with Abort

Elementary Reduction

Result	Corruption	Functions	Security	Security
[Yao'86, BMR'90]	$t < n$	P/Poly	Passive	Full Security
[DI'05]	$t < n/2$	P/Poly	Active	Full Security
[IK'00]	$t < n$	NC^1	Active (IT)	Full Security
[IPS'08, LPSY'15]	$t < n$	P/Poly	Active	Security with Abort
??	$t < n$	P/Poly	Active	Full Security

Elementary Reduction

Result	Corruption	Functions	Security	Security
[Yao'86, BMR'90]	$t < n$	P/Poly	Passive	Full Security
[DI'05]	$t < n/2$	P/Poly	Active	Full Security
[IK'00]	$t < n$	NC^1	Active (IT)	Full Security
[IPS'08, LPSY'15]	$t < n$	P/Poly	Active	Security with Abort
??	$t < n$	P/Poly	Active	Full Security

[AIK05] shows (non-elementary) non-interactive reduction in this setting to a constant-degree function g, but g depends on PRG (in NC^1)

Main Question: Does an elementary reduction exist in this setting?

Our Contributions

Result	Corruption	Functions	Security	Security
[Yao'86, BMR'90]	$t < n$	P/Poly	Passive	Full Security
[DI'05]	$t < n/2$	P/Poly	Active	Full Security
[IK'00]	$t < n$	NC^1	Active (IT)	Full Security
[IPS'08, LPSY'15]	$t < n$	P/Poly	Active	Security with Abort
Unlikely	$t < n$	P/Poly	Active	Full Security
Exists	$t < n$	P/Poly	Active	Identifiable Abort

Our Contributions (Lower Bound)

No black-box calls to PRG

Result	Corruption	Functions	Security	Security
Unlikely	$t < n$	P/Poly	Active	Full Security

For $n = 2$, existence of such an elementary reduction with partial fairness

Fairness when only one party is corrupt

➡

Existence of an information theoretic elementary reduction from any function in P/Poly to a constant degree function in the CRS model with inverse-polynomial average-case privacy against passive adversaries.

Our Contributions (Lower Bound)

Result	Corruption	Functions	Security	Security
Unlikely	$t < n$	P/Poly	Active	Full Security

For $n = 2$, existence of such an elementary reduction with partial fairness

➡ Existence of an information theoretic elementary reduction from any function in P/Poly to a constant degree function in the CRS model with inverse-polynomial average-case privacy against passive adversaries.

➡ A constant-round protocol ∀ 2-party function in P/Poly with inverse-polynomial average-case information-theoretic security in OT-hybrid model.

3 Decade old open problem!!

➡ A constant-round protocol ∀ 3-party function in P/Poly with inverse-polynomial average-case information-theoretic security.

Our Contributions (Positive Result)

Result	Corruption	Functions	Security	Security
Exists	$t < n$	P/Poly	Active	Identifiable Abort

Similar reduction is implicit in [BOSSV20].

If parties are allowed to interact twice with g, then we can achieve fairness.

Can get full-security if g is allowed to depend on the PRG.

Our Main Ideas
(Lower Bound)

Lower Bound (Talk Outline)

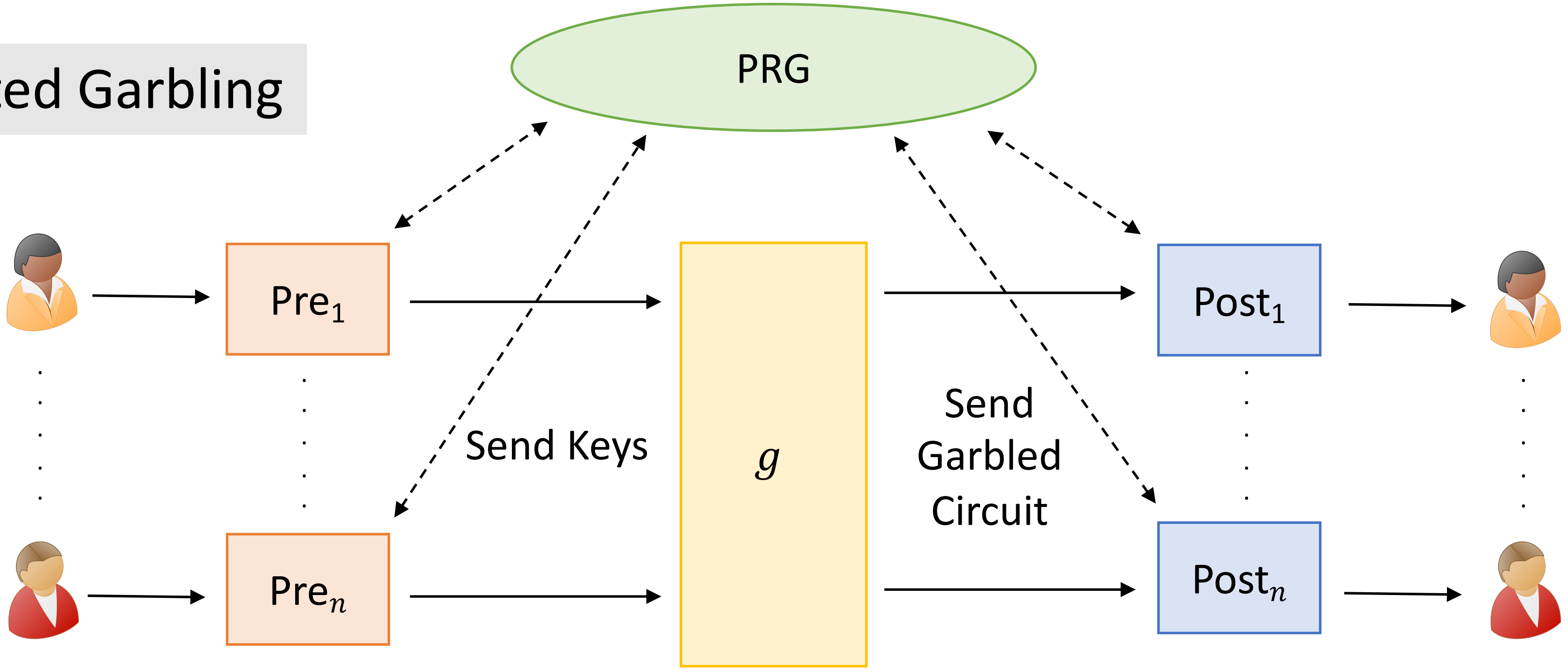
Warm-up	Why existing passively secure elementary reductions fail to achieve full-security against active adversaries
Main Theorem	Why actively secure elementary reductions with full security are unlikely to exist for general efficiently computable functions

Lower Bound (Talk Outline)

Warm-up	Why existing passively secure elementary reductions fail to achieve full-security against active adversaries
Main Theorem	Why actively secure elementary reductions with full security are unlikely to exist for general efficiently computable functions

Existing Passively Secure Elementary Reductions

Distributed Garbling



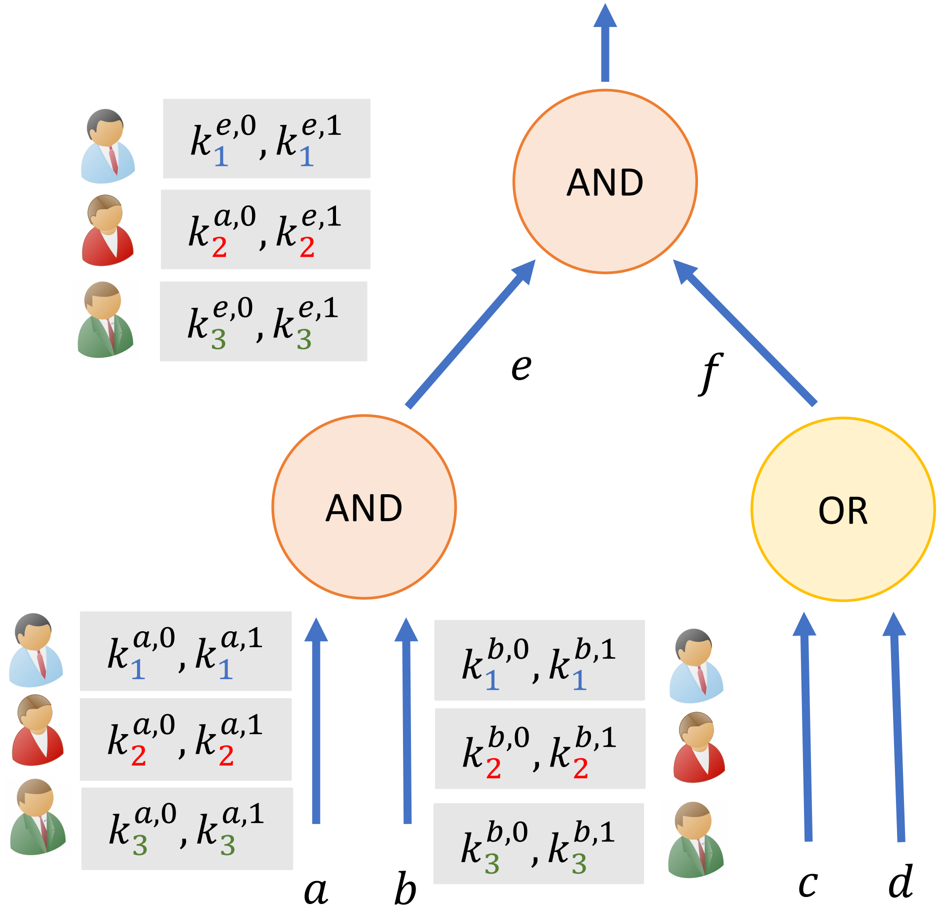
Sample random keys for each wire in the circuit representation of f

Garbling function (Constant-round)

Evaluate the garbled circuit

What are the PRG calls used for?

Distributed Garbling



Each gate in the circuit is individually garbled

Each garbled gate: Set of 4 randomly permuted ciphertexts

Each ciphertext is a distributed encryption, where:

$$keys = \left((k_1^{a,\alpha}, k_2^{a,\alpha}, k_3^{a,\alpha}), \left(k_1^{b,\beta}, k_2^{b,\beta}, k_3^{b,\beta}\right)\right)$$

$$msg = \left(k_1^{e,\gamma}, k_2^{e,\gamma}, k_3^{e,\gamma}\right)$$

for some $\alpha, \beta \in \{0,1\}$ and $\gamma = \text{AND}(\alpha, \beta)$

For circuits with more than polylog depth, $keys$ must be shorter than msg

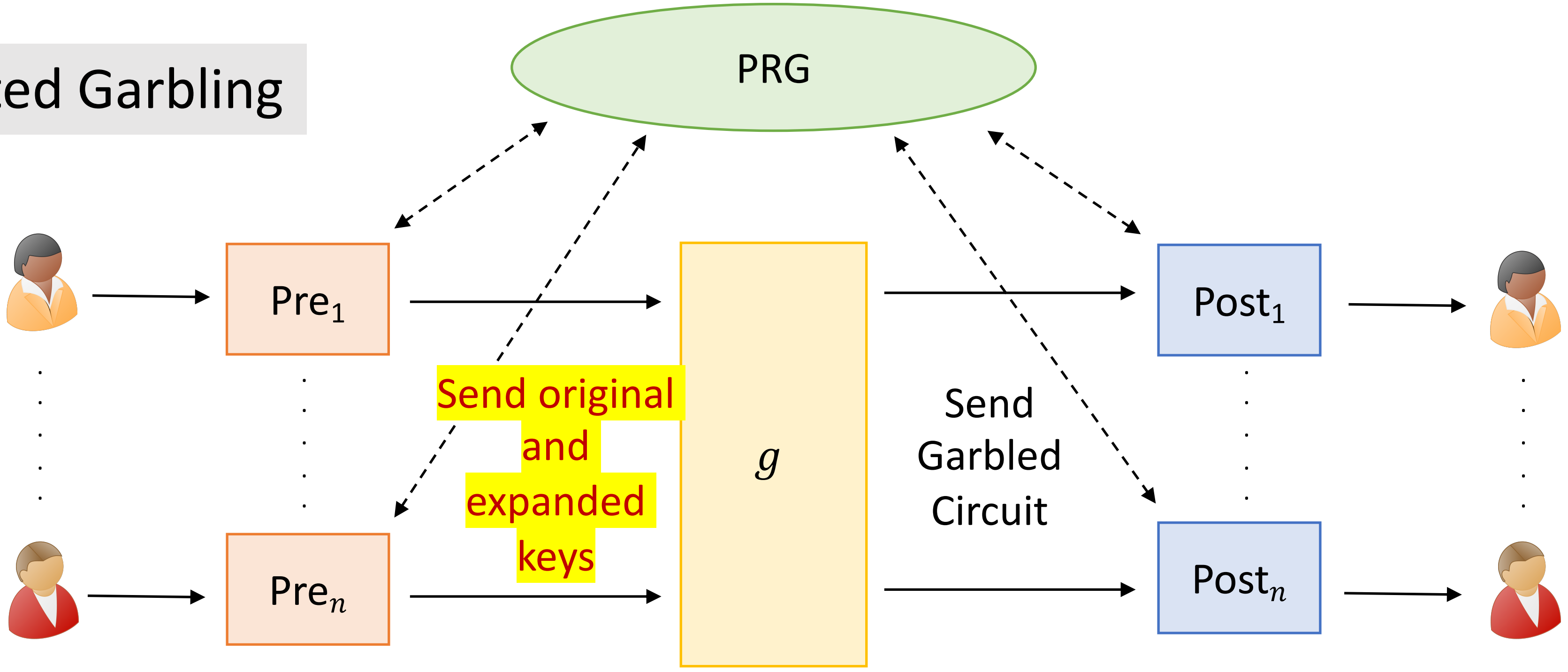
Distributed encryption/decryption uses PRGs to expand $keys$

Key-expansion using PRGs can be done by the parties locally

Parties sample random keys for each wire in the circuit

Existing Passively Secure Elementary Reductions

Distributed Garbling



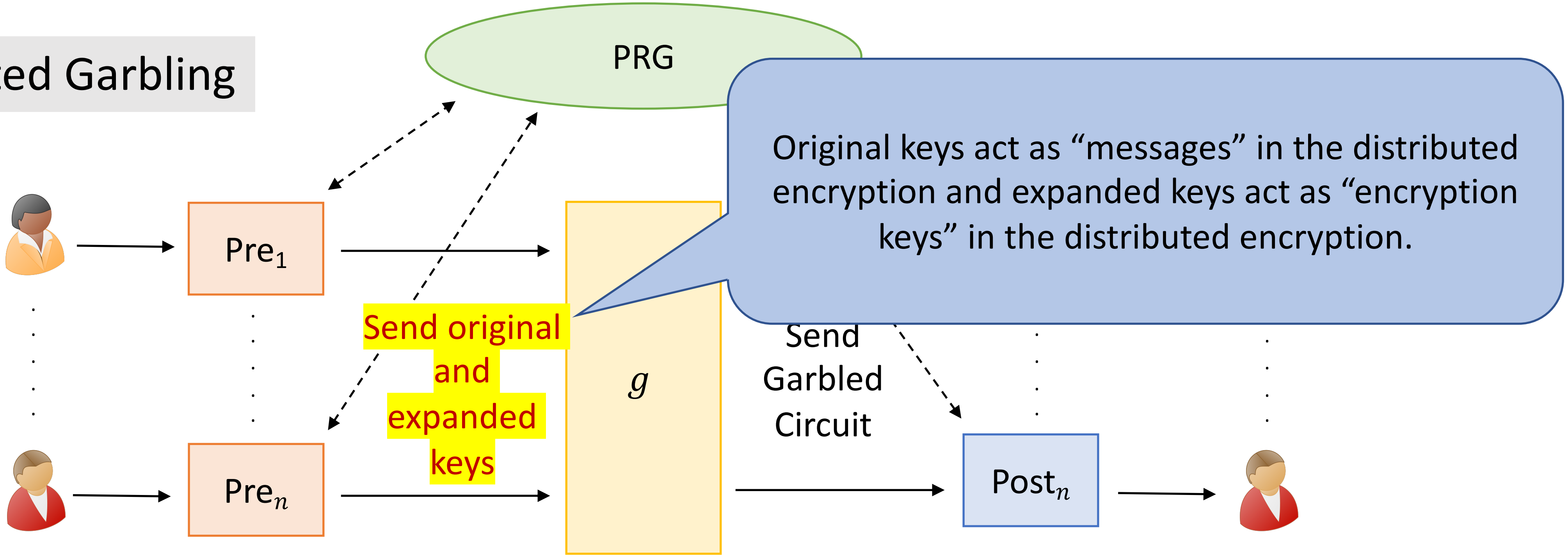
Sample random keys for each wire in the circuit representation of f and expands them using PRG

Garbling function implements distributed encryptions using expanded keys

Evaluate the garbled circuit

Existing Passively Secure Elementary Reductions

Distributed Garbling



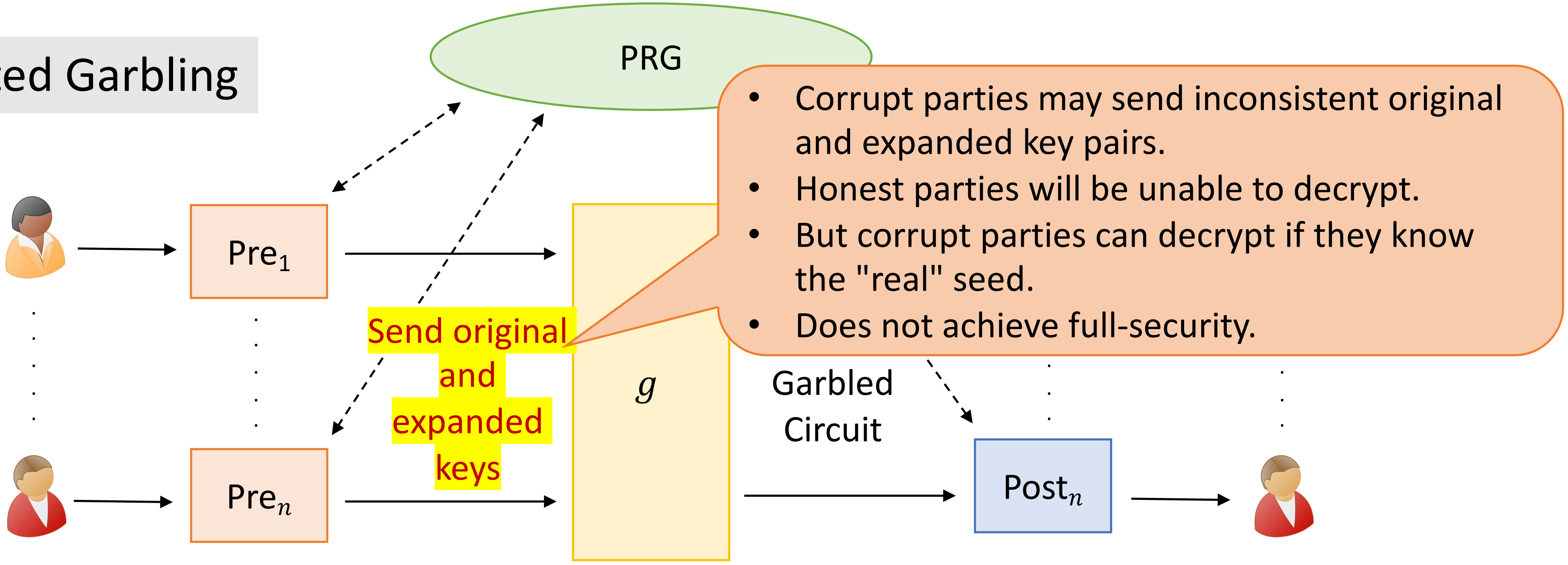
Sample random keys for each wire in the circuit representation of f and expands them using PRG

Garbling function implements distributed encryptions using expanded keys

Evaluate the garbled circuit

Problem with Active Adversaries

Distributed Garbling



Sample random keys for each wire in the circuit representation of f and expands them using PRG

Garbling function implements distributed encryptions using expanded keys

Evaluate the garbled circuit

Lower Bound (Talk Outline)

Warm-up

Why existing passively secure elementary reductions fail to achieve full-security against active adversaries

Main Theorem

Why actively secure elementary reductions with full security are unlikely to exist for general efficiently computable functions

Lower Bound (Talk Outline)

Warm-up

Why existing passively secure elementary reductions fail to achieve full-security against active adversaries

Main Theorem

Why actively secure elementary reductions with full security are unlikely to exist for general efficiently computable functions

For $n = 2$, existence of such an elementary reduction with partial fairness

$\Rightarrow$

Existence of an information theoretic elementary reduction from any function in P/Poly to a constant degree function in the CRS model with inverse-polynomial average-case privacy against passive adversaries.

Lower Bound (Talk Outline)

Warm-up

Why existing passively secure elementary reductions fail to achieve full-security against active adversaries

Main Theorem

Why actively secure elementary reductions with full security are unlikely to exist for general efficiently computable functions

Holds even if the parties have access to a Random Oracle (RO) !!

For $n = 2$, existence of such an elementary reduction with partial fairness

Existence of an information theoretic elementary reduction from any function in P/Poly to a constant degree function in the CRS model with inverse-polynomial average-case privacy against passive adversaries.

Lower Bound (Talk Outline)

Warm-up

Why existing passively secure elementary reductions fail to achieve full-security against active adversaries

Main Theorem

Why actively secure elementary reductions with full security are unlikely to exist for general efficiently computable functions

Holds even if the parties have access to a Random Oracle (RO) !!

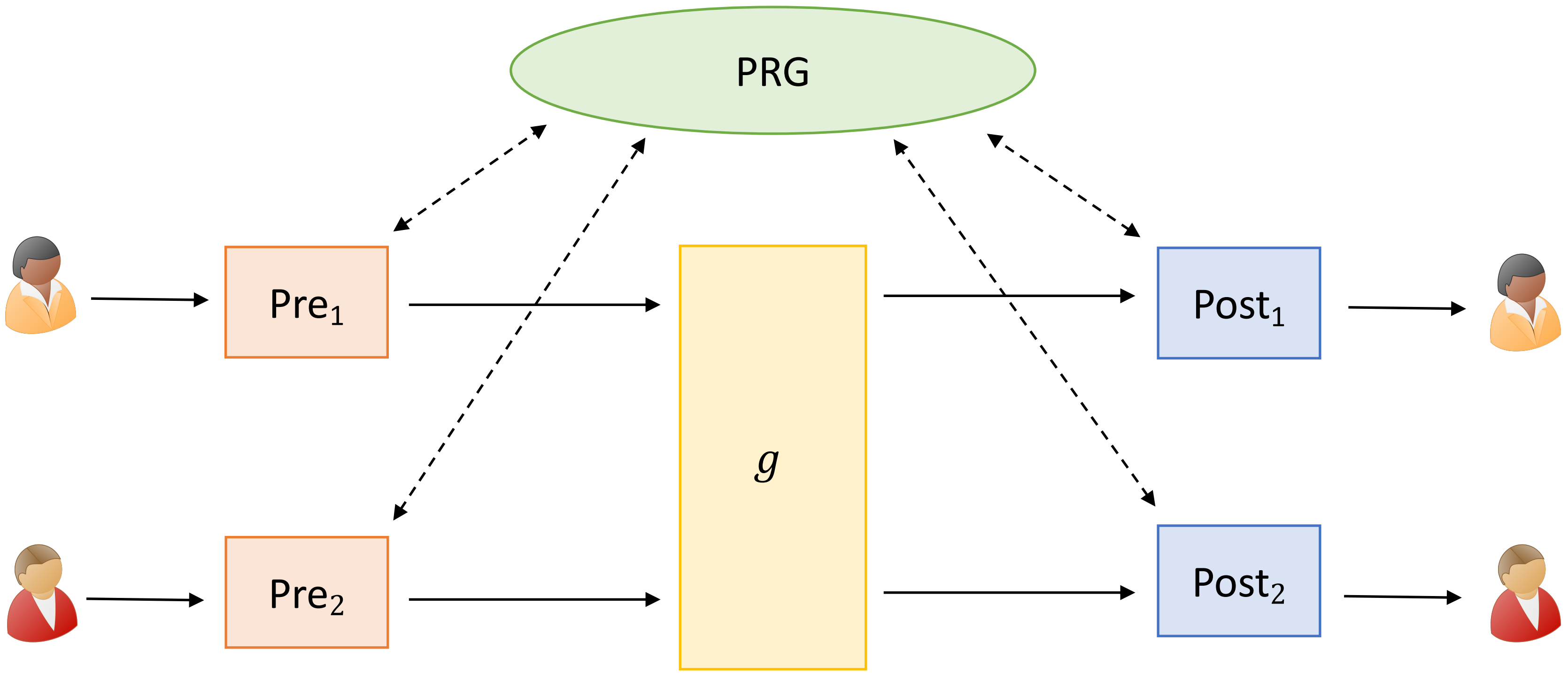
For $n = 2$, existence of such an elementary reduction with partial fairness

Existence of an information theoretic elementary reduction from any function in P/Poly to a constant degree function in the CRS model with inverse-polynomial average-case privacy against passive adversaries.

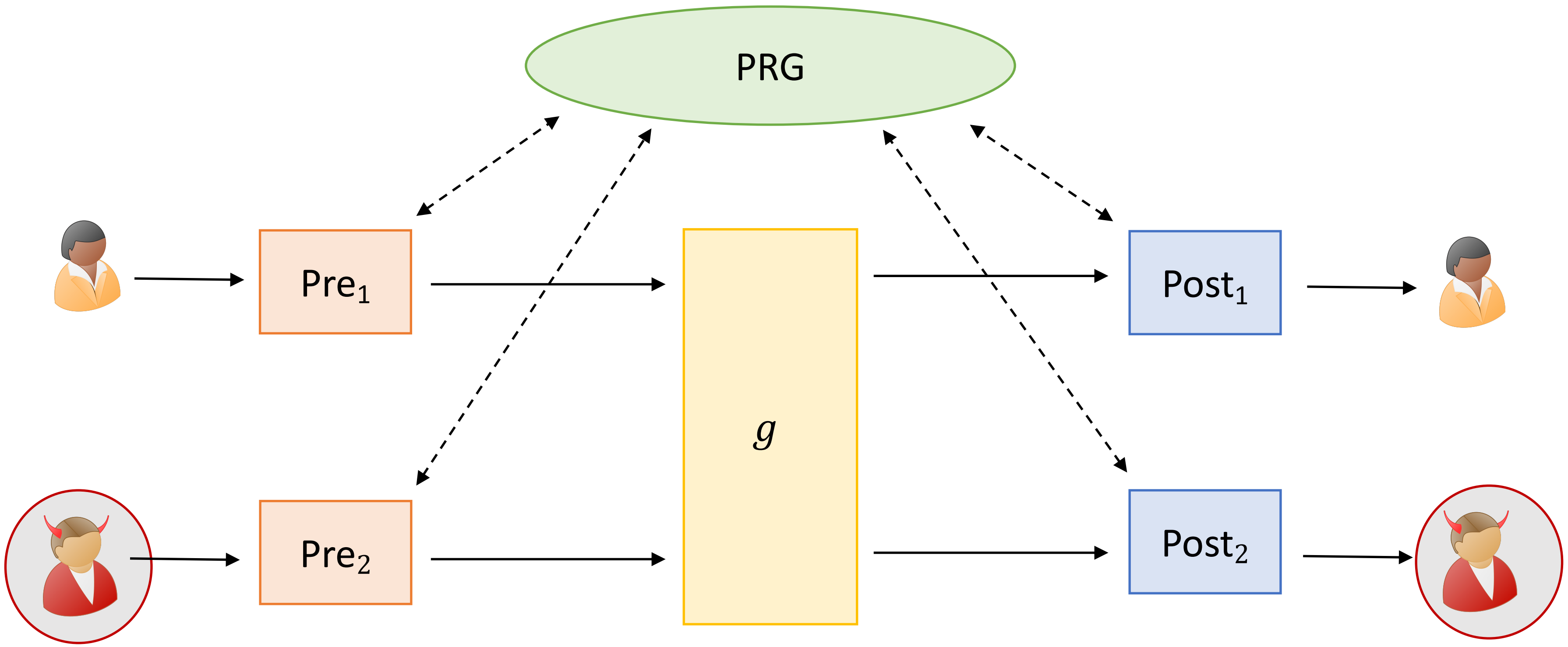
This restriction makes the theorem stronger

This restriction can be removed if parties only make random queries to the RO

Proving the Main Theorem



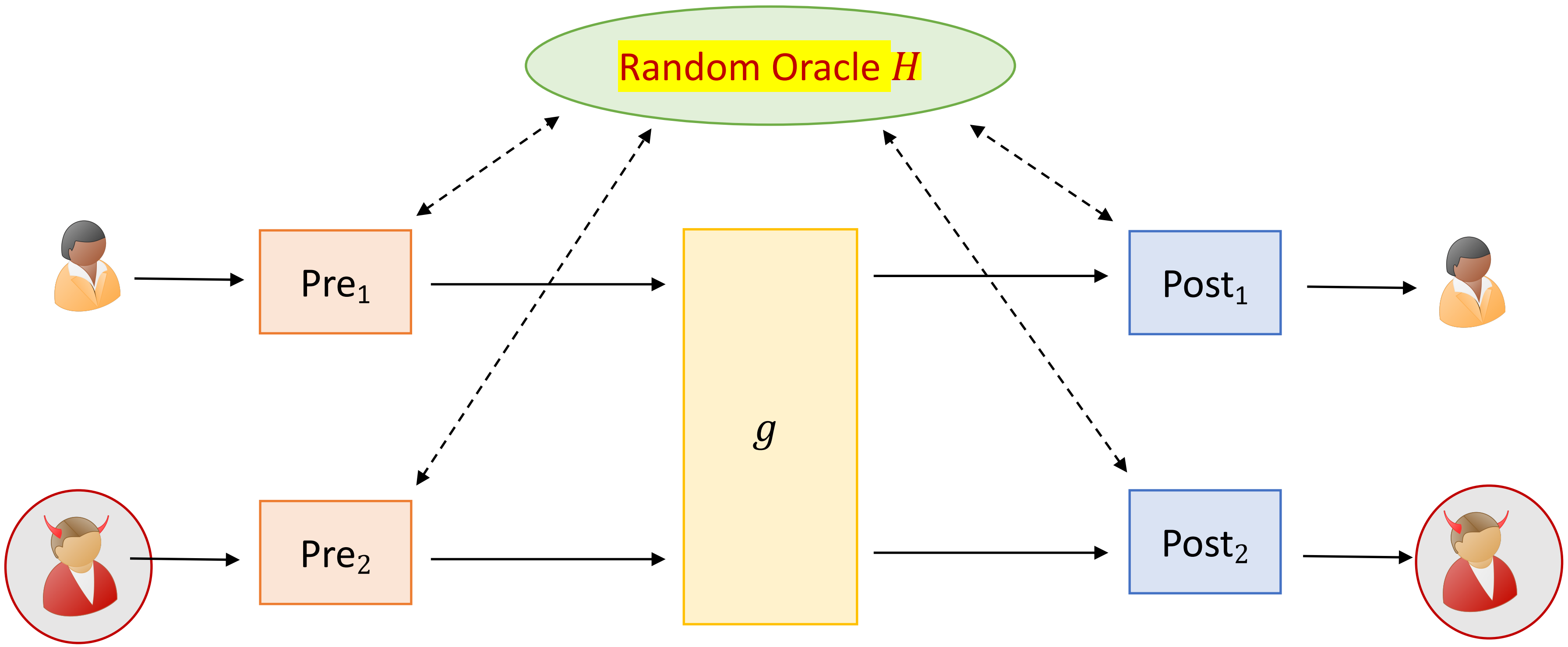
Proving the Main Theorem



Fairness only when Bob is corrupt

Assume FSOC, ∃ elementary reduction from every poly-sized 2-party function with partial fairness against active adversaries.

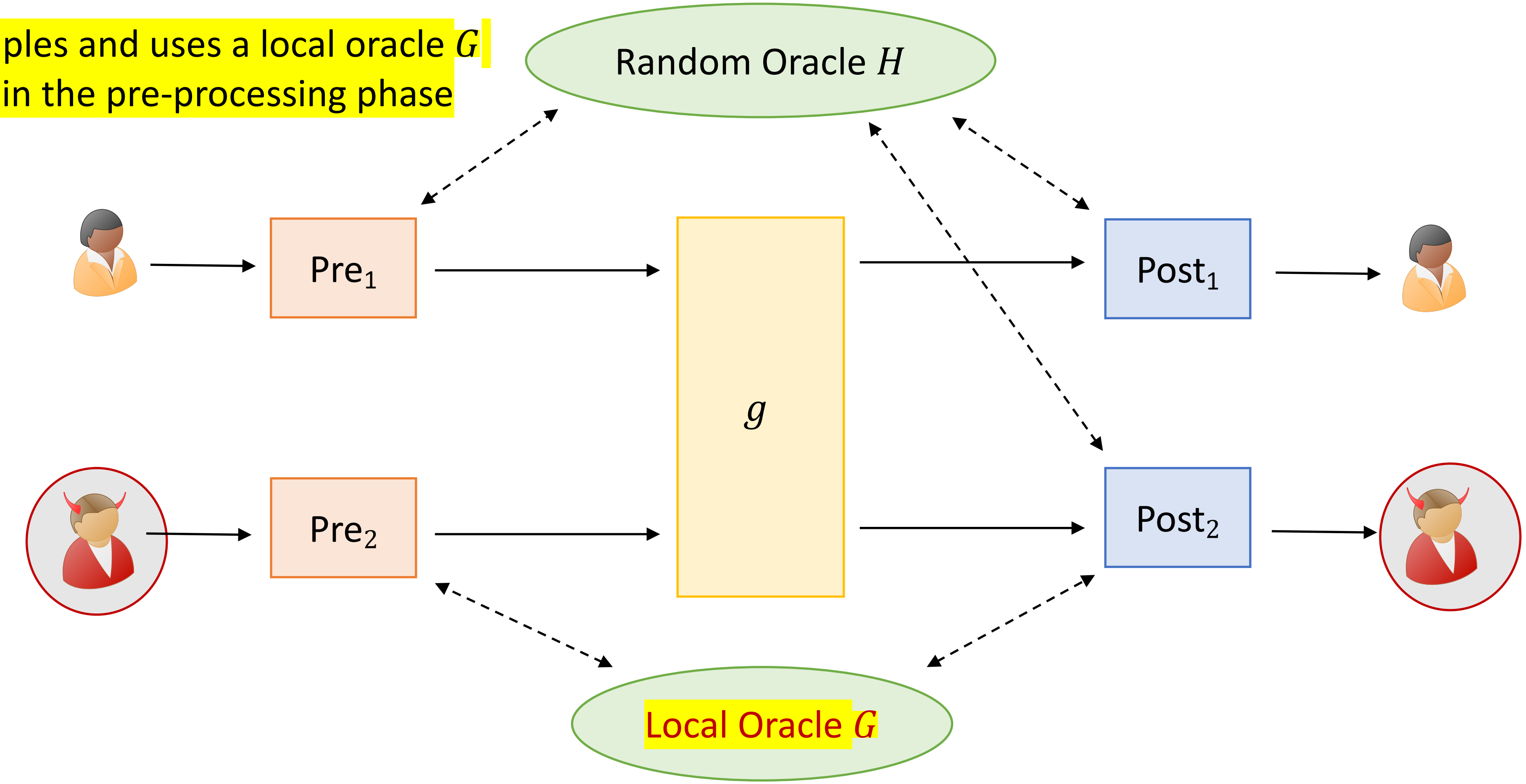
Proving the Main Theorem



Assume FSOC, ∃ elementary reduction from every poly-sized 2-party function with partial fairness against active adversaries.

Proving the Main Theorem

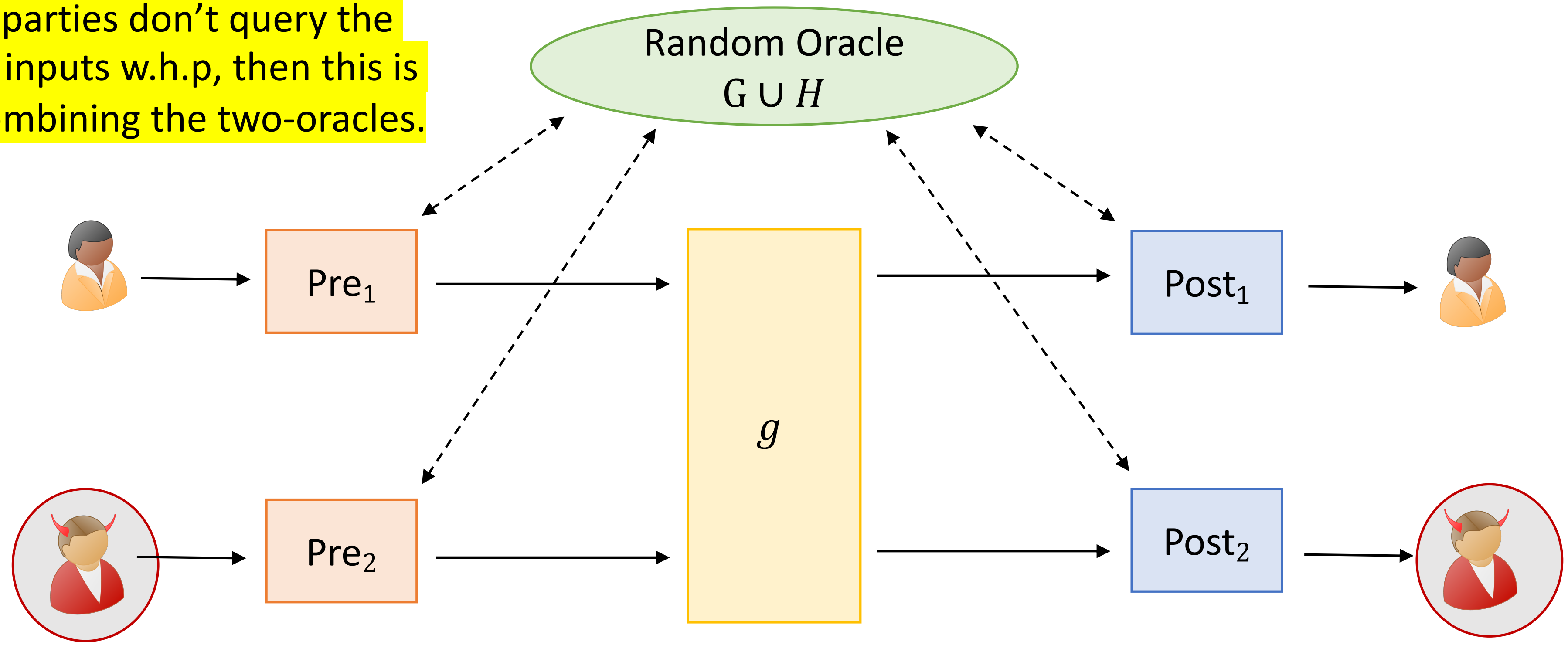
Adversary samples and uses a local oracle G instead of H in the pre-processing phase



Assume FSOC, ∃ elementary reduction from every poly-sized 2-party function with partial fairness against active adversaries.

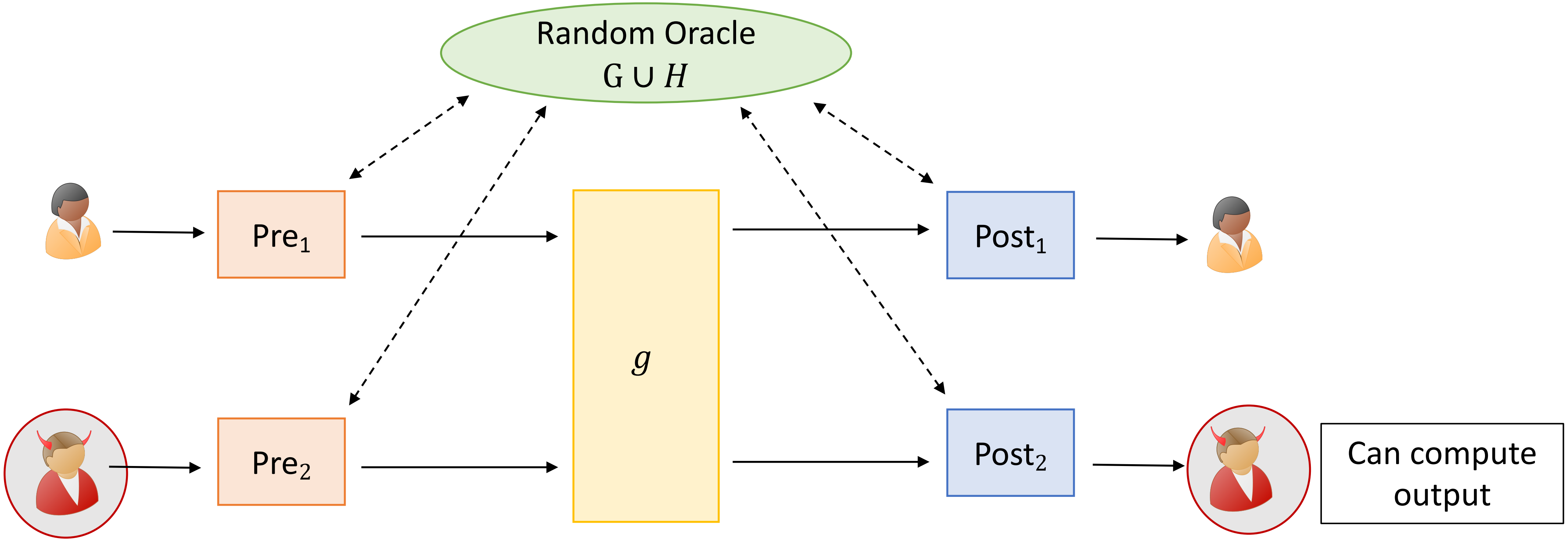
Proving the Main Theorem

Assuming the parties don't query the oracle on same inputs w.h.p, then this is equivalent to combining the two-oracles.



Assume FSOC, ∃ elementary reduction from every poly-sized 2-party function with partial fairness against active adversaries.

Proving the Main Theorem



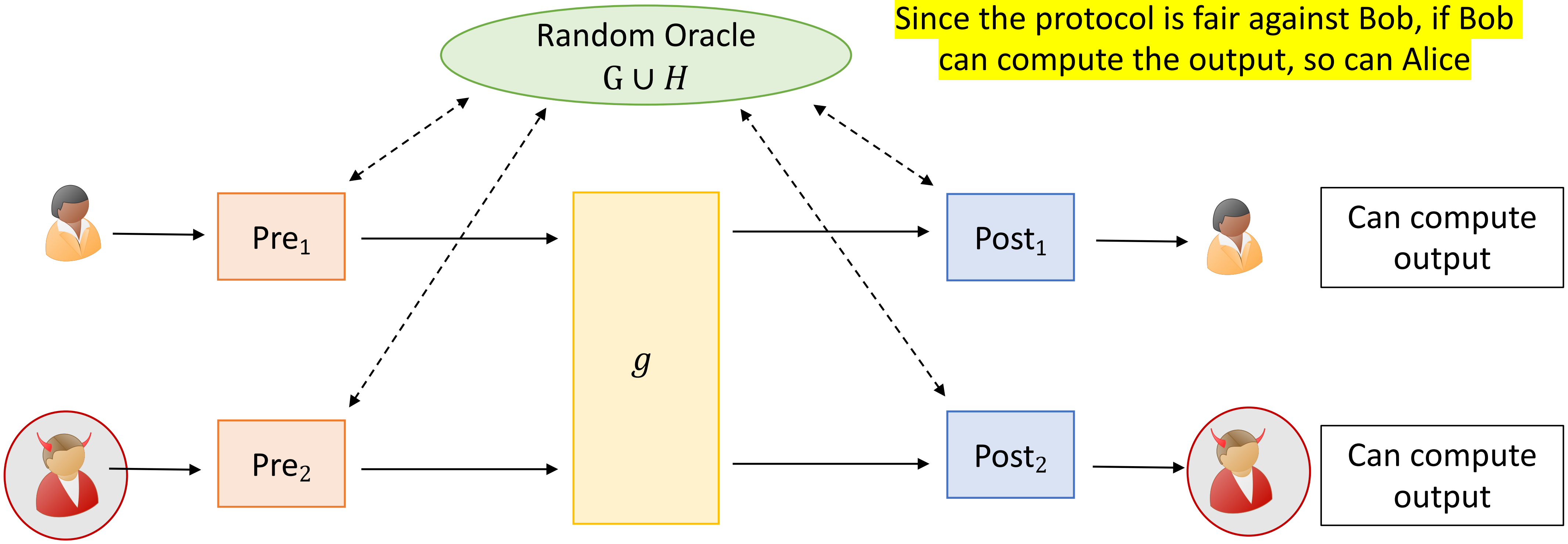
Since the Bob knows both G and H, it can compute the correct output.

Assume FSOC, ∃ elementary reduction from every poly-sized 2-party function with partial fairness against active adversaries.

Proving the Main Theorem

This modified protocol has correctness.

Since the protocol is fair against Bob, if Bob can compute the output, so can Alice

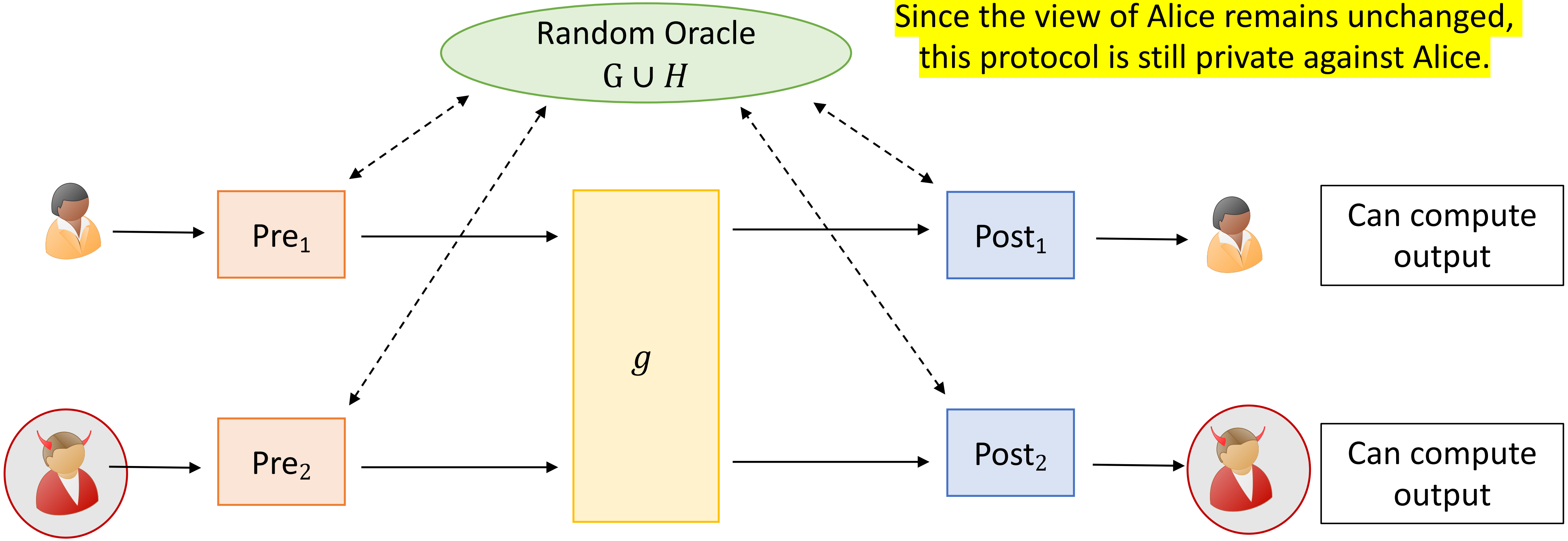


Assume FSOC, ∃ elementary reduction from every poly-sized 2-party function with partial fairness against active adversaries.

Proving the Main Theorem

This modified protocol has privacy.

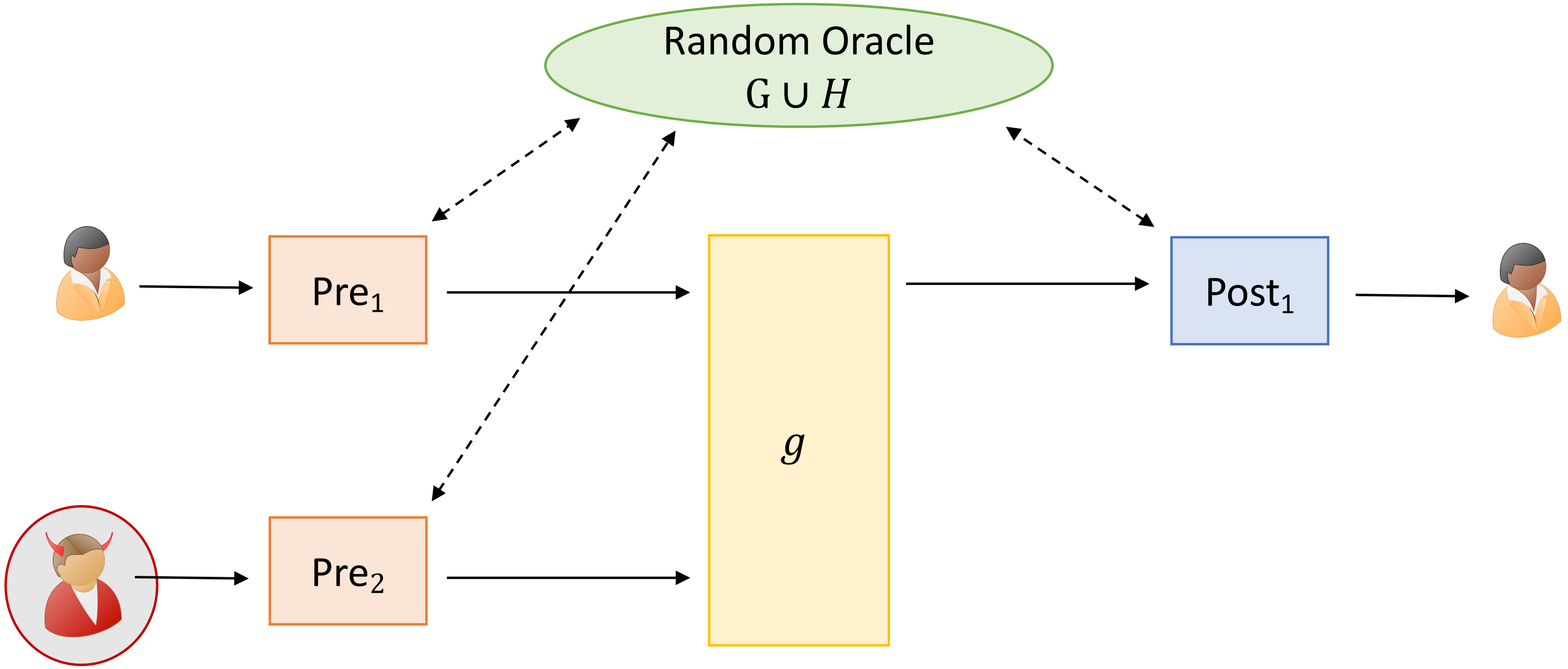
Since the view of Alice remains unchanged, this protocol is still private against Alice.



Assume FSOC, ∃ elementary reduction from every poly-sized 2-party function with partial fairness against active adversaries.

Proving the Main Theorem

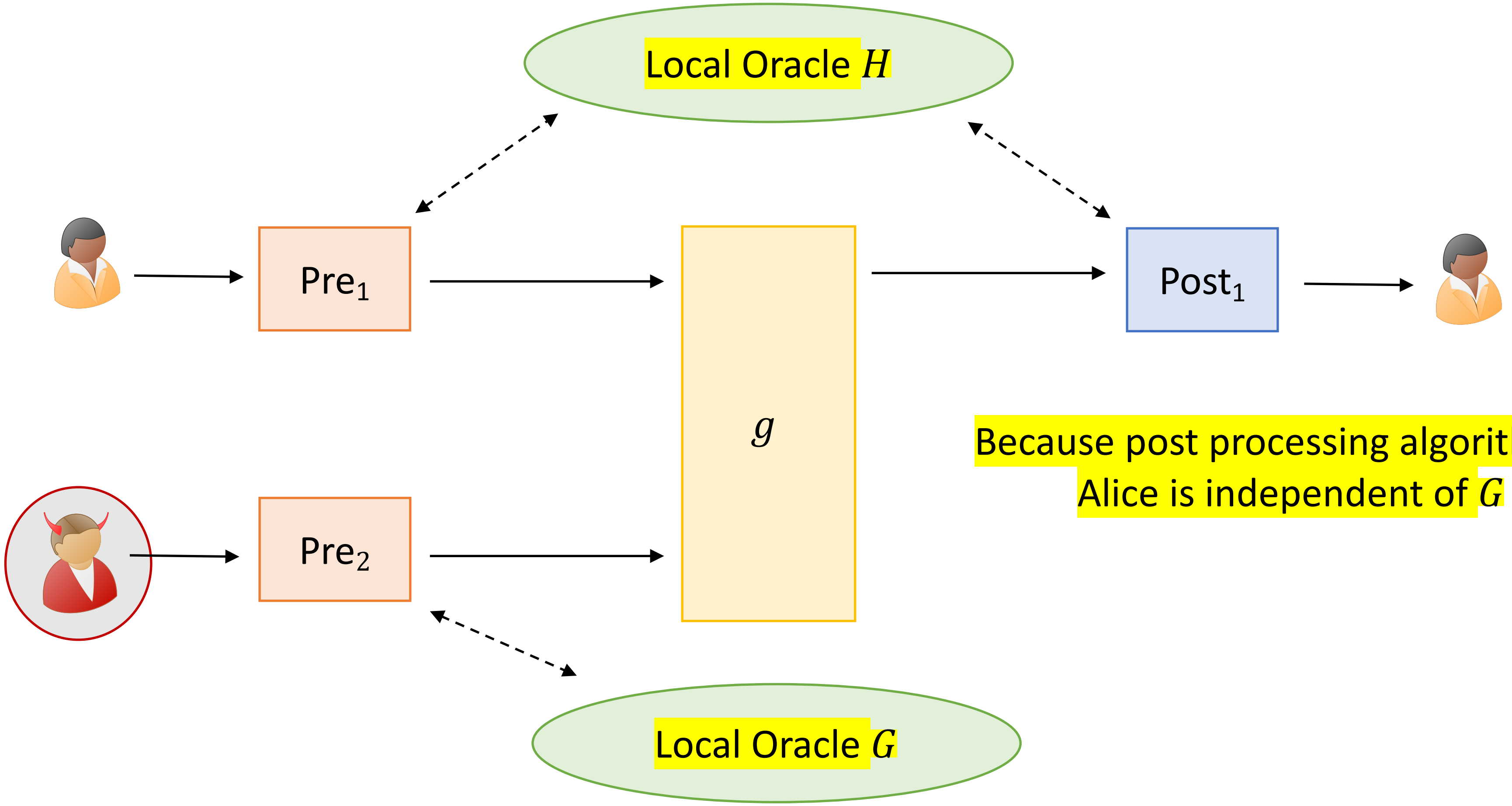
A modified protocol for single output functionality



Assume FSOC, ∃ elementary reduction from every poly-sized 2-party function with partial fairness against active adversaries.

Proving the Main Theorem

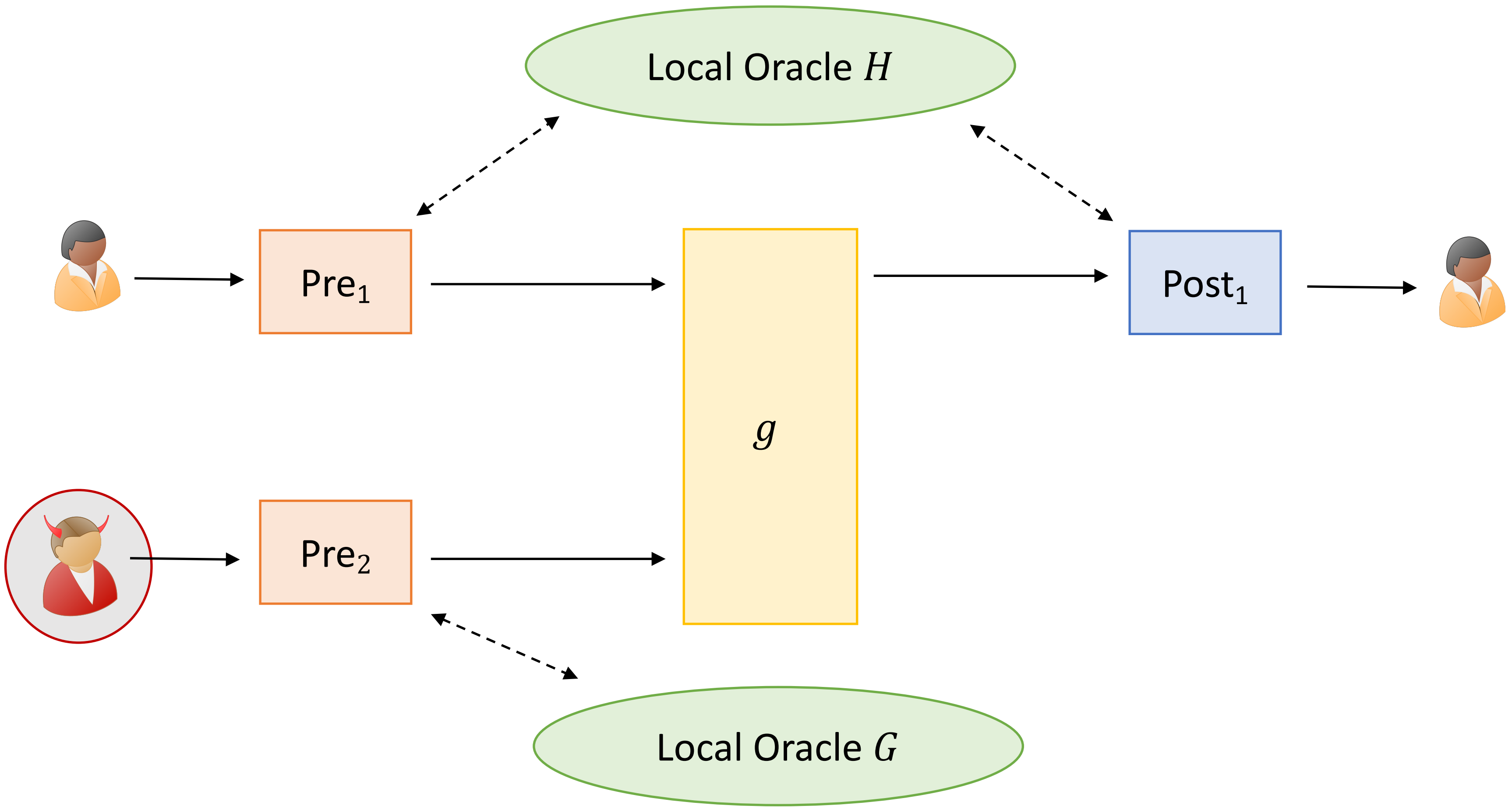
A modified protocol for single output functionality



Because post processing algorithm of Alice is independent of G

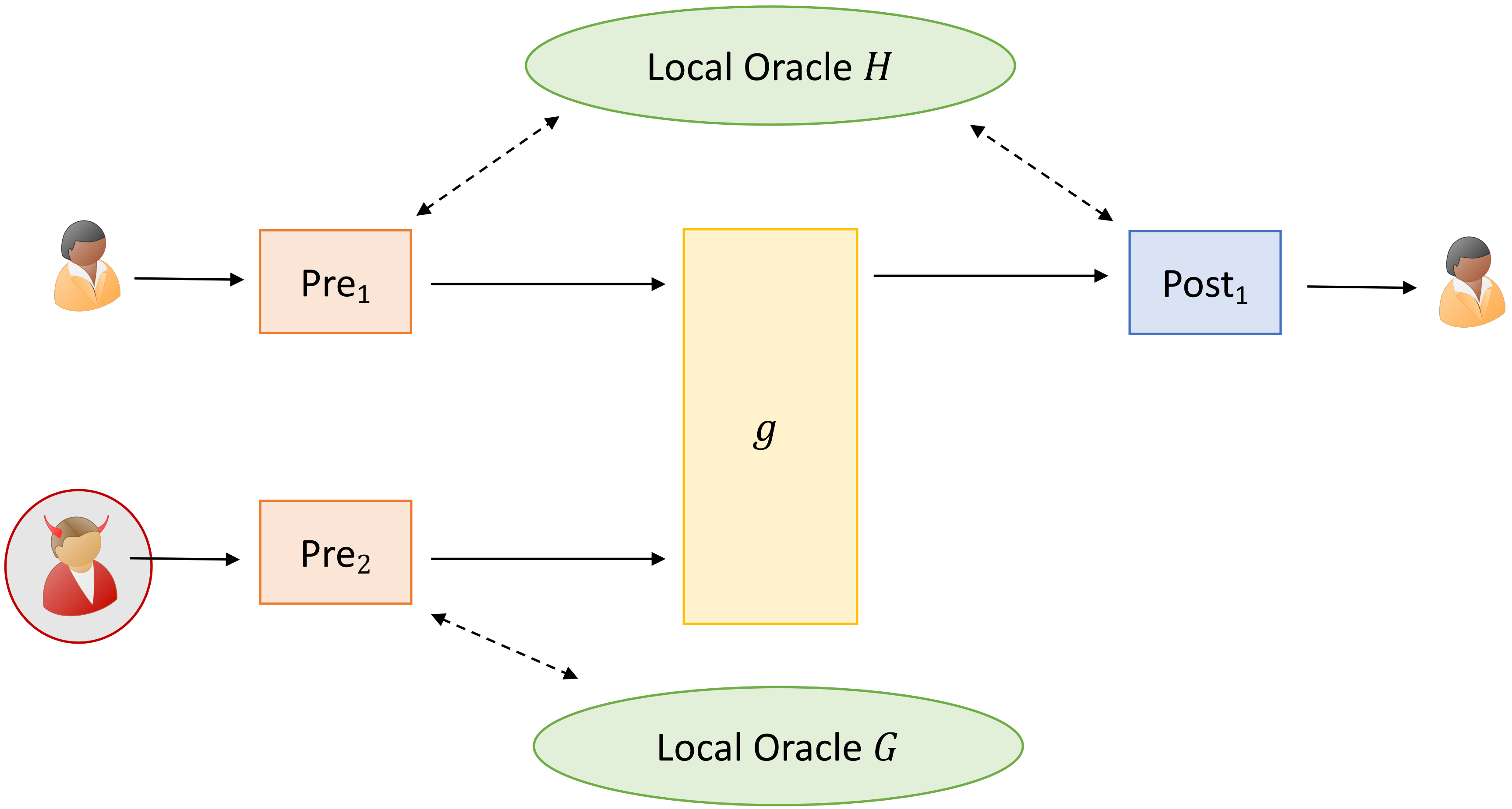
Assume FSOC, ∃ elementary reduction from every poly-sized 2-party function with partial fairness against active adversaries.

Proving the Main Theorem



This is an information-theoretic passively-secure elementary reduction for single output functionalities.

Proving the Main Theorem



Two-copies of the above reduction gives an information-theoretic elementary reduction for all two-input functionalities.

Our Lower Bound: Removing Simplifying Assumptions

Simplifying Assumption I

Simulation-based definition of fairness $\Longrightarrow$ If corrupt Bob gets the output so does Alice

How to Remove it

- Use authenticated functionalities that give Bob a MAC computed on his input, under a key chosen by Alice.
- Fairness w.r.t. such functionalities implies the above simplified notion.

Simplifying Assumption II

Alice and Bob's queries to the PRG do not intersect

How to Remove it

- Identify ''heavy queries'' [BM09].
- Corrupt Bob only queries its local oracle on ``non-heavy'' queries.
- Only allows us to get inverse-polynomial average-case security.
- Ensure correctness by adding a "detect-and-reveal" mechanism to functionality g.

Remarks about our Main Theorem

Our main theorem shows an example of a cryptographic problem for which

An information-theoretic solution cannot be ruled out.

Black-box use of a given primitive is useless for solving the problem

A non-black-box use of the primitive allows us to solve the problem

Remarks about our Main Theorem

Our main theorem shows an example of a cryptographic problem for which

An information-theoretic solution cannot be ruled out.

Black-box use of a given primitive is useless for solving the problem

A non-black-box use of the primitive allows us to solve the problem

Existing examples only satisfy at most 2 of these

[HOZ'13,MMP'14]: Random oracles are "useless" for secure 2-party computation of various functionalities.

[ABGIS'20]: Impossibility of elementary reductions to oblivious transfer

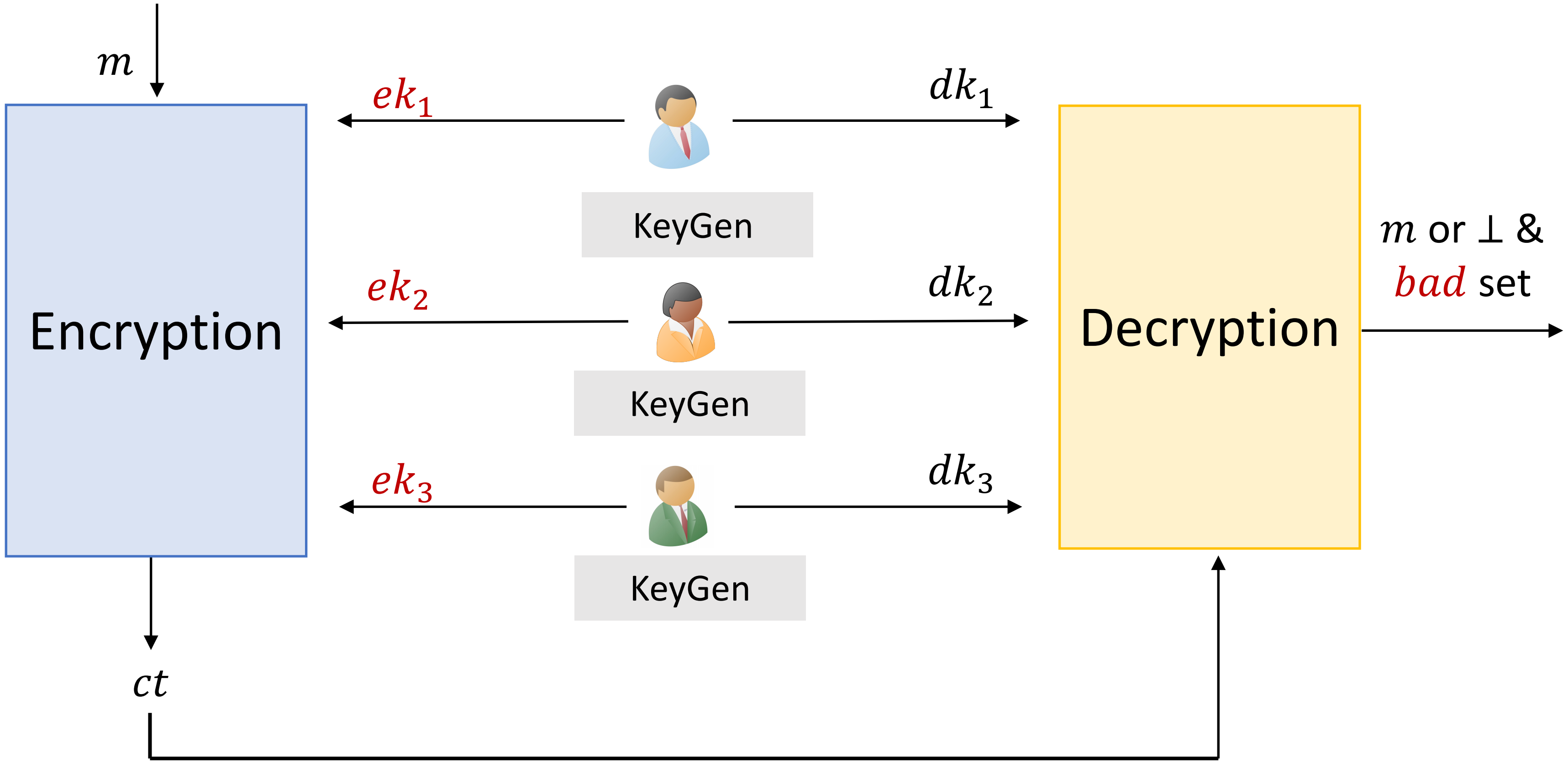
Our Main Ideas
(Positive Result)

Positive Results

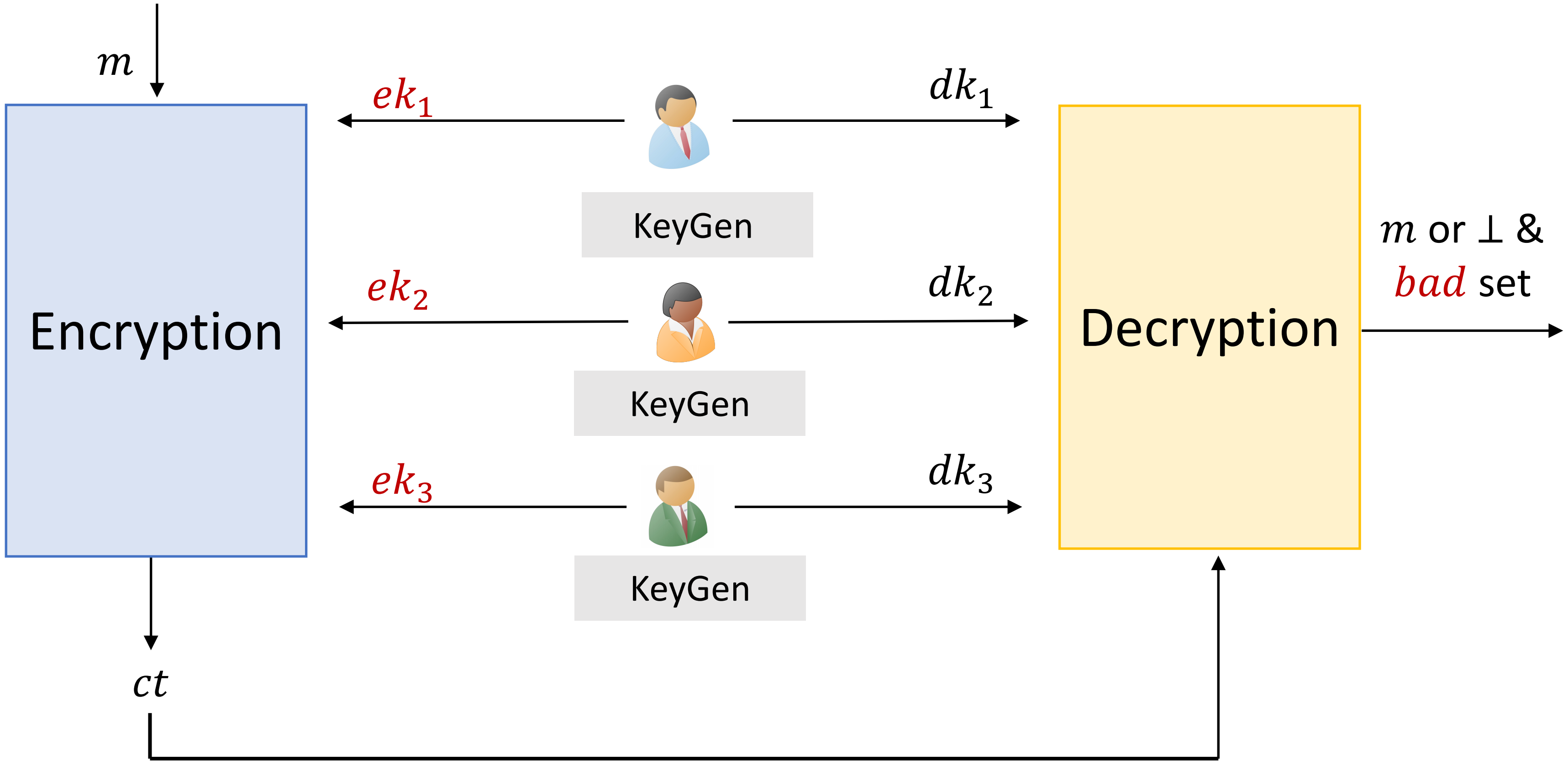
Elementary reduction from every poly-sized n-input functionality, that achieves security with identifiable abort against any $t < n$ active corruptions.

- Define a notion of distributed encryption with identifiable abort and give a construction
- This distributed encryption when used with the standard garbling protocol achieves security with identifiable abort

Distributed Encryption



Distributed Encryption

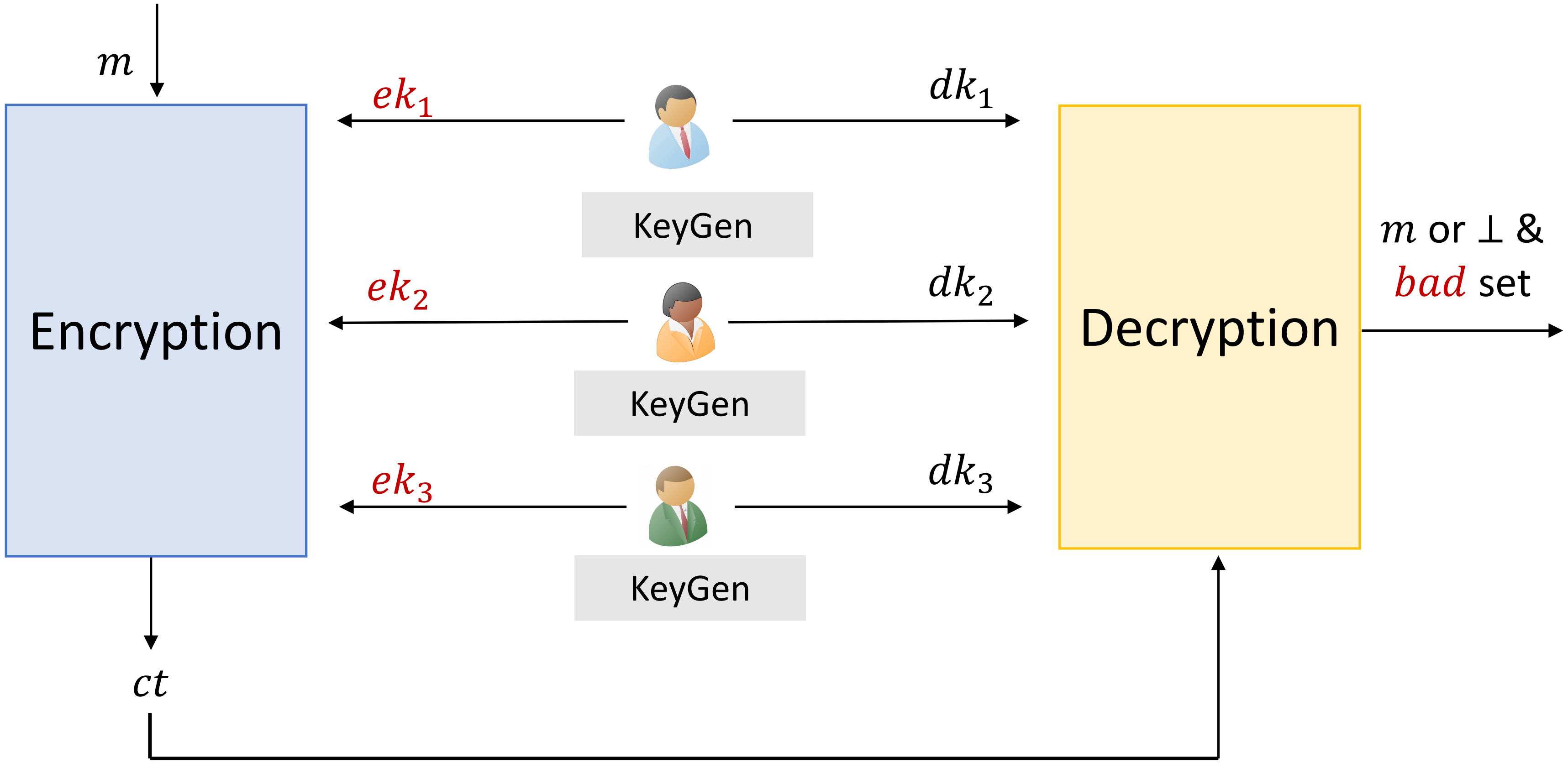


Symmetric-key Encryption

Only KeyGen and Decryption are allowed to depend on a PRG.

Security if at least one key-pair is honestly generated

Distributed Encryption

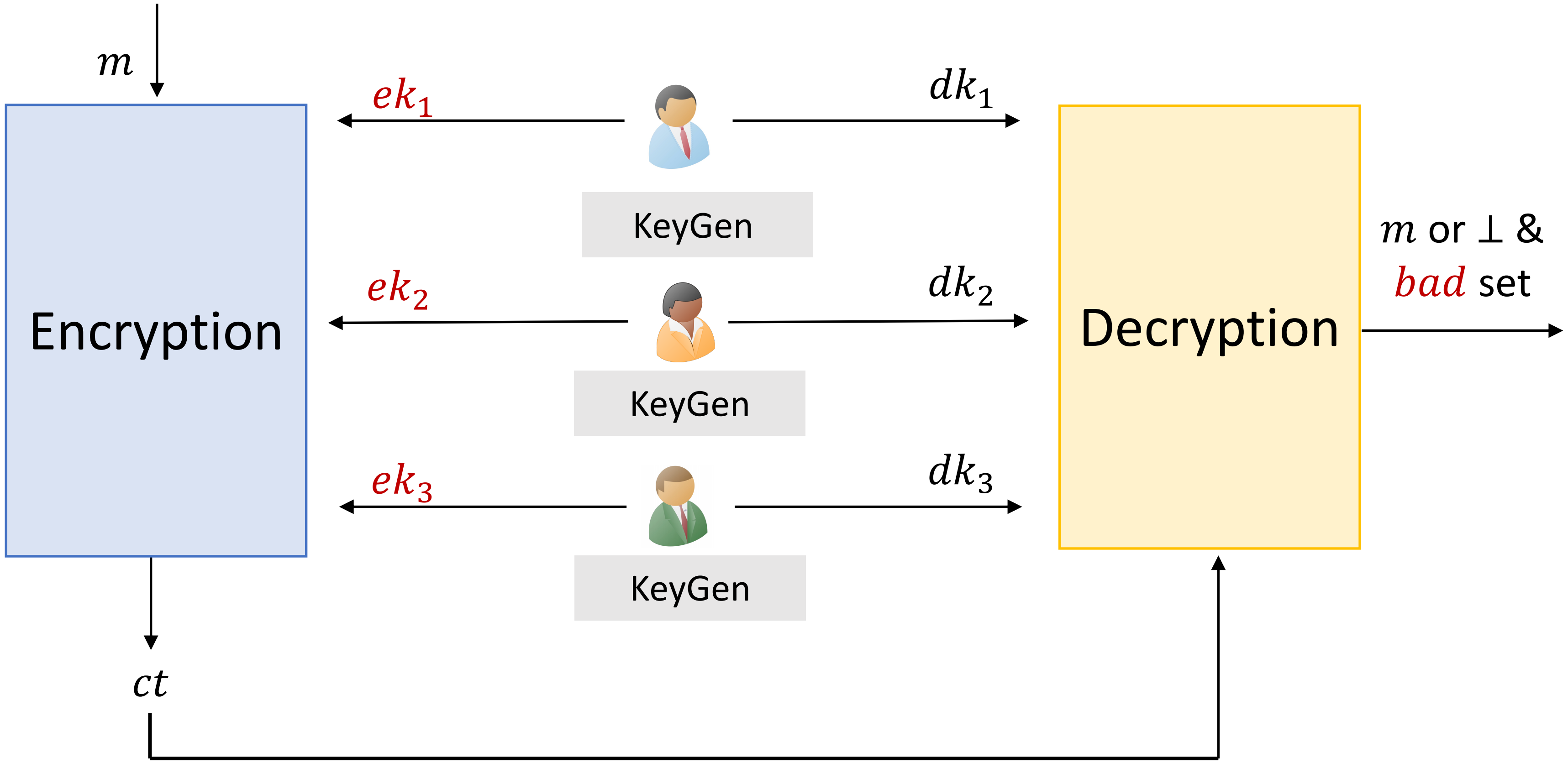


Security: Key-pairs are sufficient to simulate the outcome of Decryption

With abort: outcome is a valid message or $\perp$.

With identifiable abort: outcome is a valid message or $\perp$ and bad set.

Distributed Encryption



Security: Key-pairs are sufficient to simulate the outcome of Decryption

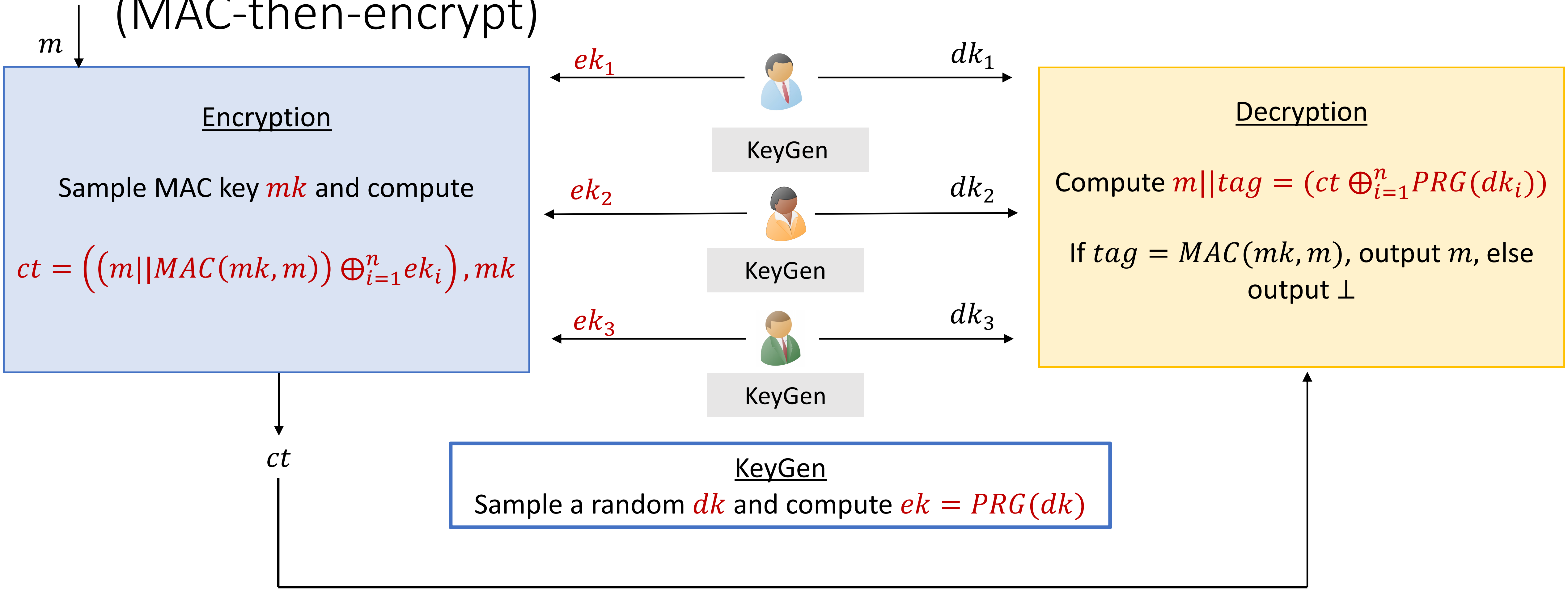
With abort: outcome is a valid message or $\perp$.

Cut-and-choose

With identifiable abort: outcome is a valid message or $\perp$ and bad set.

Distributed Encryption with Abort

(MAC-then-encrypt)



Simulating the Outcome of Decryption: Output $\perp$, if $\oplus_{i=1}^{n}(ek_i \oplus PRG(dk_i))$ is a non-zero-string

Distributed Encryption with Identifiable Abort

m

Encryption

- Sample a random $\frac{k}{2}$ subset of the $ek's$ and output them.
- Use the remaining sets of $ek's$ for encrypting m.

ek_1 KeyGen dk_1

ek_2 KeyGen dk_2

ek_3 KeyGen dk_3

Decryption

- Check if the revealed $ek's$ are consistent with the corresponding $dk's$ and identify any bad key-pairs.
- Decrypt the ciphertexts computed using remaining keys and take a majority of decrypted values.

ct

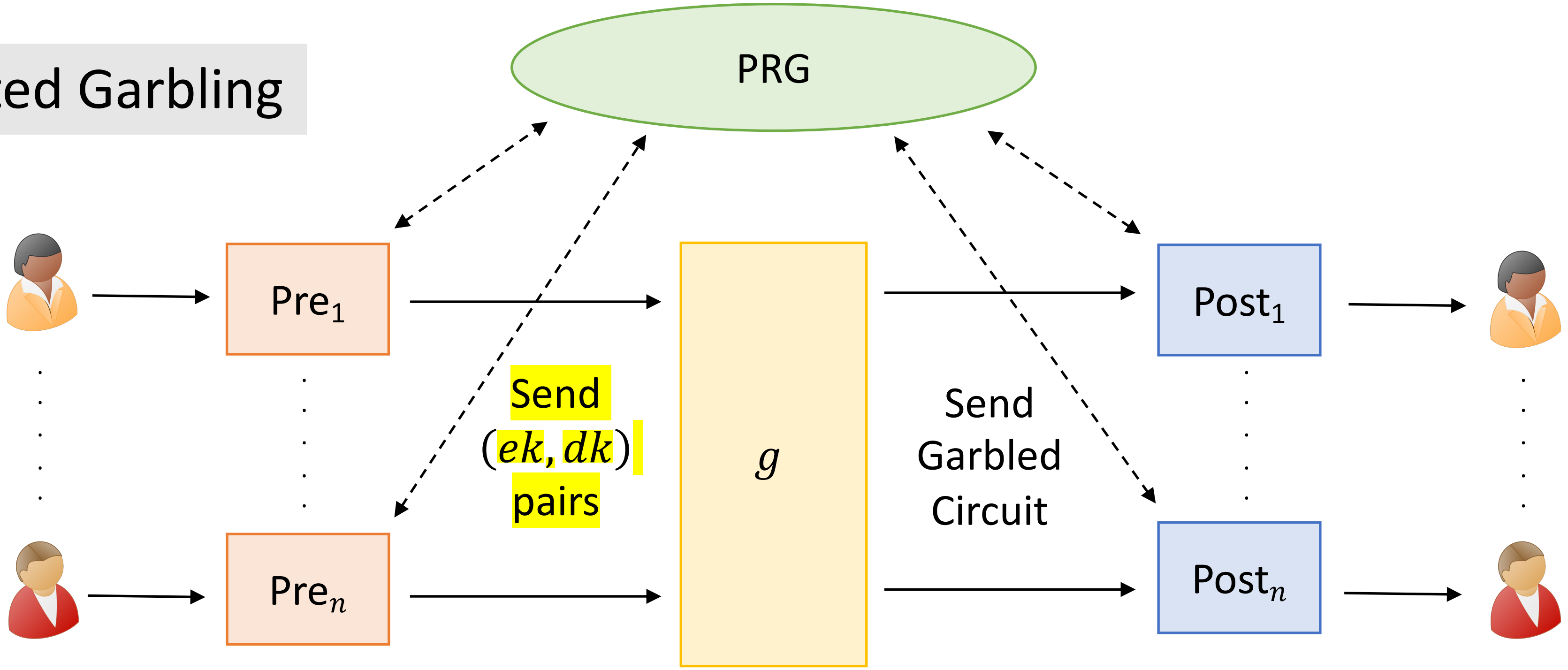
<u>KeyGen</u>

Sample k key pairs of DE with abort.

<u>Simulating the Outcome of Decryption</u>: Sample a random $\frac{k}{2}$ subset of $ek's$ and check if they are consistent with the corresponding $dk's$ and identify any bad key-pairs.

Elementary Reduction with Identifiable Abort

Distributed Garbling



Run key generation to sample (ek, dk) pairs for each wire in the circuit representation of f and expands them using PRG

Garbling function implements encryption algorithm of distributed encryption scheme with identifiable abort

Evaluate the garbled circuit by running decryption algorithms

Conclusion

Elementary reduction for all efficiently computable functions that achieve full-security against any $t < n$ active corruptions is unlikely

Existence of elementary reduction for all efficiently computable functions that achieve identifiable abort against any $t < n$ active corruptions.

https://eprint.iacr.org/2021/1208

Thank You