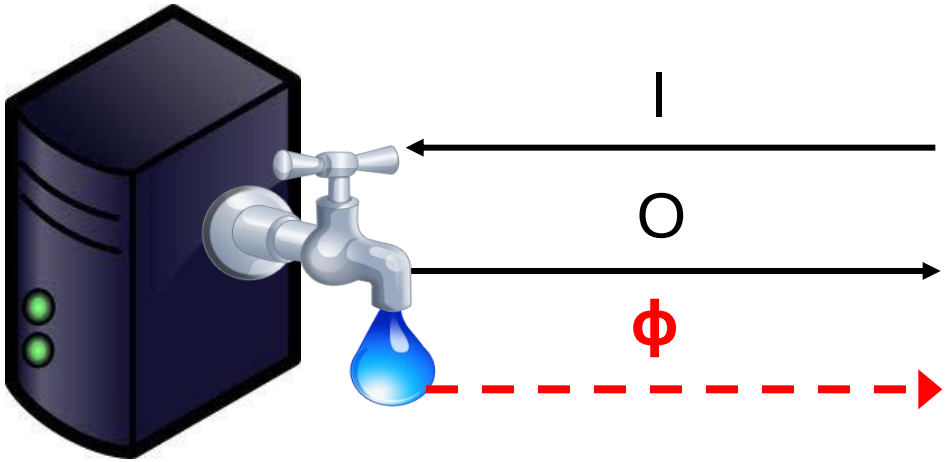
Leakage-Resilient Zero Knowledge

Sanjam Garg Abhishek Jain Amit Sahai

UCLA

Leakage-Resilient Cryptography

- Traditional Cryptography: adv has only **black-box** access to a cryptosystem



- LR-Cryptography: “open the black-box” more & more

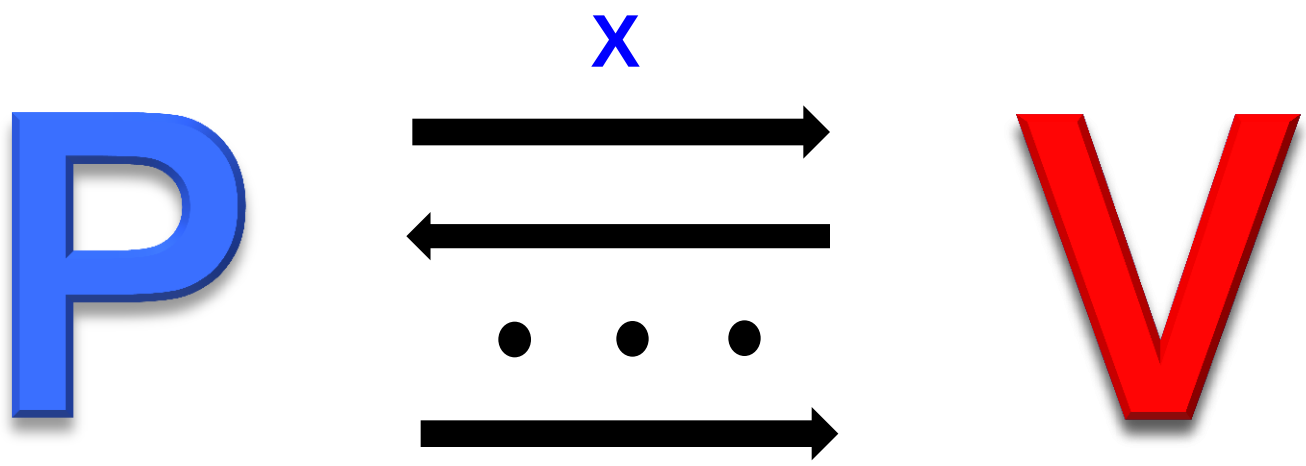
Prior Work

- Leakage-Resilient (Stateless) Primitives
  - [DP08, AGV09, Pie09, DKL09, NS09, ADW09, KV09, FKPR…KKV10, …]
- Le…
  - [ISW03, IPSW06, FRR…, …11]

This work: Leakage on entire state of honest party during protocol execution

- Leakage-Resilient (Interactive) Protocols
  - [IKOS09, ADW09, DHLW10]
  - Limited leakage during protocol execution

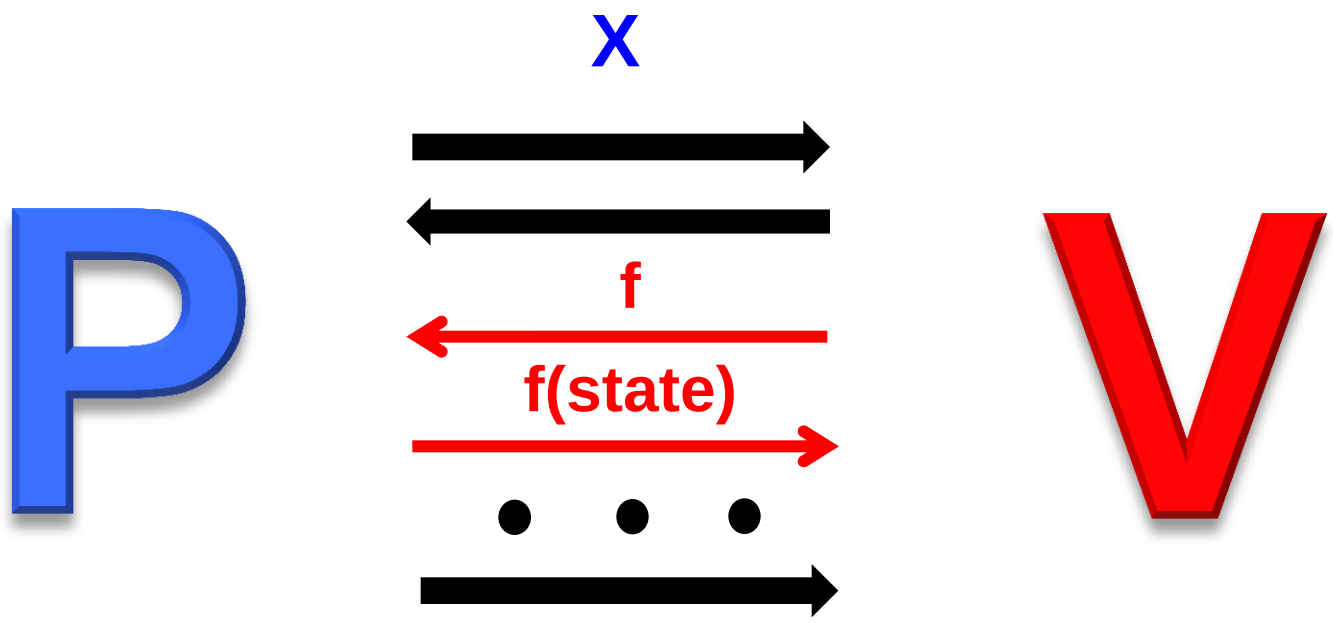
Zero Knowledge Proofs [GMR]



Verifier learns nothing beyond validity of **x**

(For every V, there exists S that “simulates” the view of V)

Zero Knowledge with Leakage?



Verifier learns something beyond validity of **X**

Can not be achieved.

Leakage-Resilient Zero Knowledge?

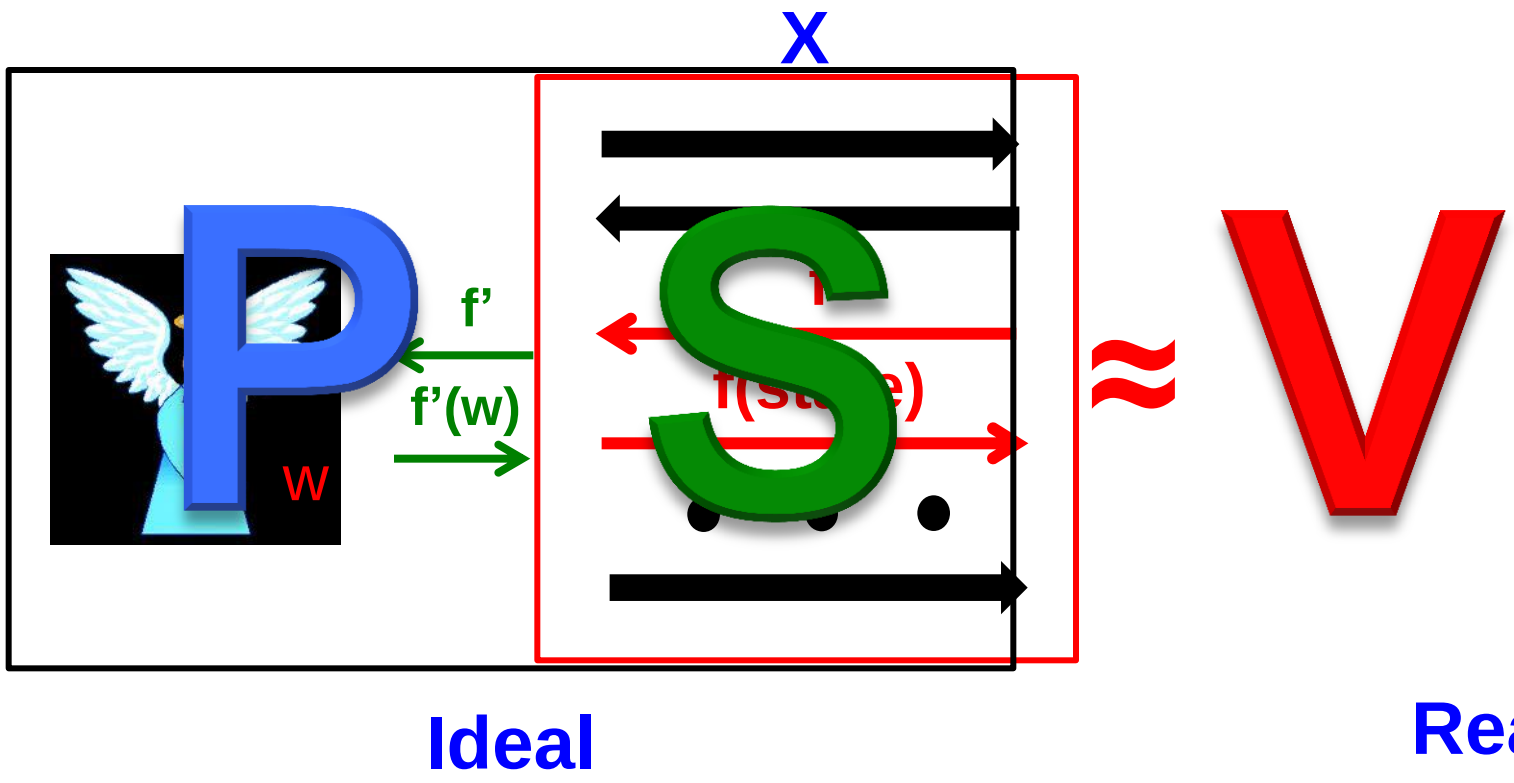
- Only computation leaks information [MR'04]
  - Often problematic (e.g. cold-boot attacks [HSH+08])
  - Standard ZK impossible
- "Leakage-free" pre-processing
  - Limits applicability; impossible to yield standard ZK

Leakage-Resilient Zero Knowledge?

- What we want :
  - Leakage on entire state of prover, anytime during the protocol
  - No “leakage-free” phase
  - Meaningful notion; useful in application scenarios

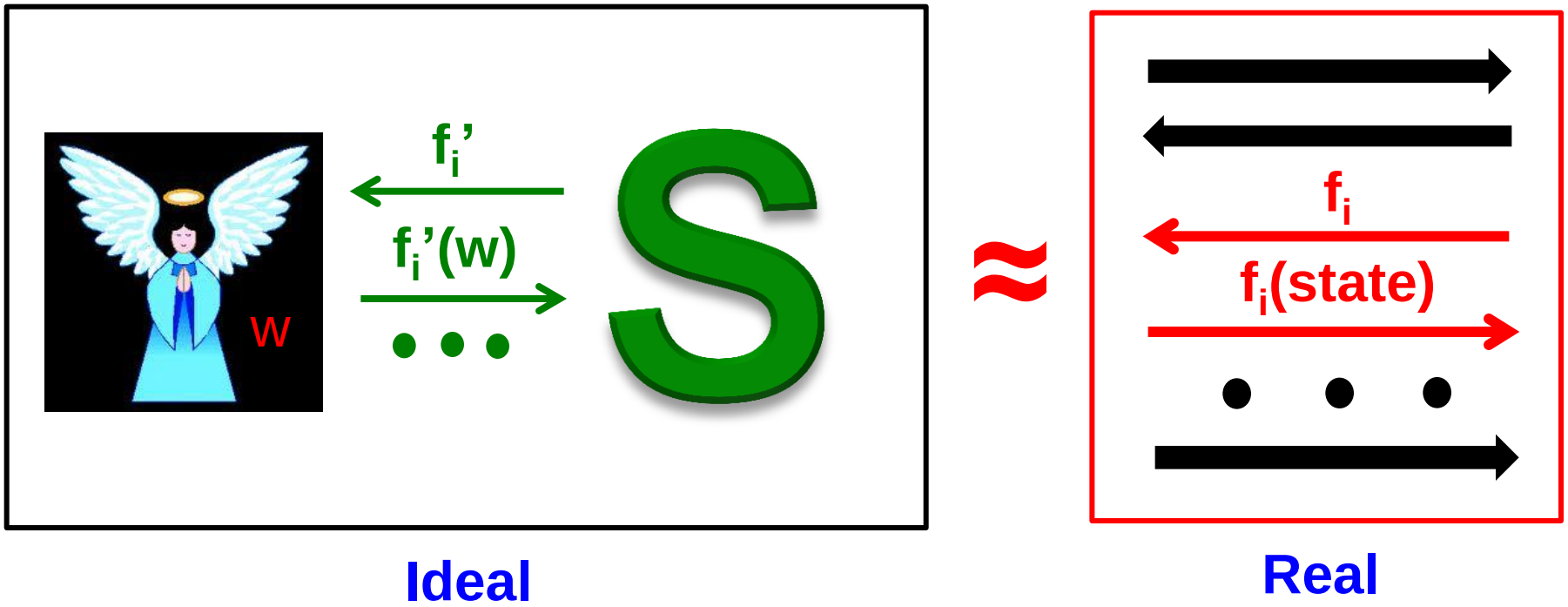
Cannot achieve standard ZK guarantee since simulator cannot simulate leakage queries on the witness

Our Definition



- Real/Ideal paradigm, where Ideal is also leaky

Our Definition …



- Total Ideal Leakage ≤ **λ** × (Total Real Leakage)
- When **λ≈1**: *Verifier learns nothing beyond validity of* **X** *and leakage information*

How much leakage in the ideal world?

Related Notion: Knowledge Complexity [GP'91]

- Witness oracle (or leakage on witness in ideal world) is not a new concept

- Main difference: In their case protocol inherently leaked information

- Our Setting: Leakage is because of side channel attacks

Leakage-Oblivious Simulation

- Leakage oracle should only help S to answer leakage queries of V
- Leakage oblivious simulation: S does not see answers to leakage queries
- Necessary for some scenarios

Our Results

- Main result: $(1+\varepsilon)$-LR-ZK interactive proof system (based on general assumptions)
  - almost optimal leakage parameter (λ-LR-ZK for $\lambda<1$ impossible)
  - first positive result on handling arbitrary leakage during protocol exec
- LR-NIZK proofs (under standard assumptions)
- Exciting concurrent work [BCH'11]

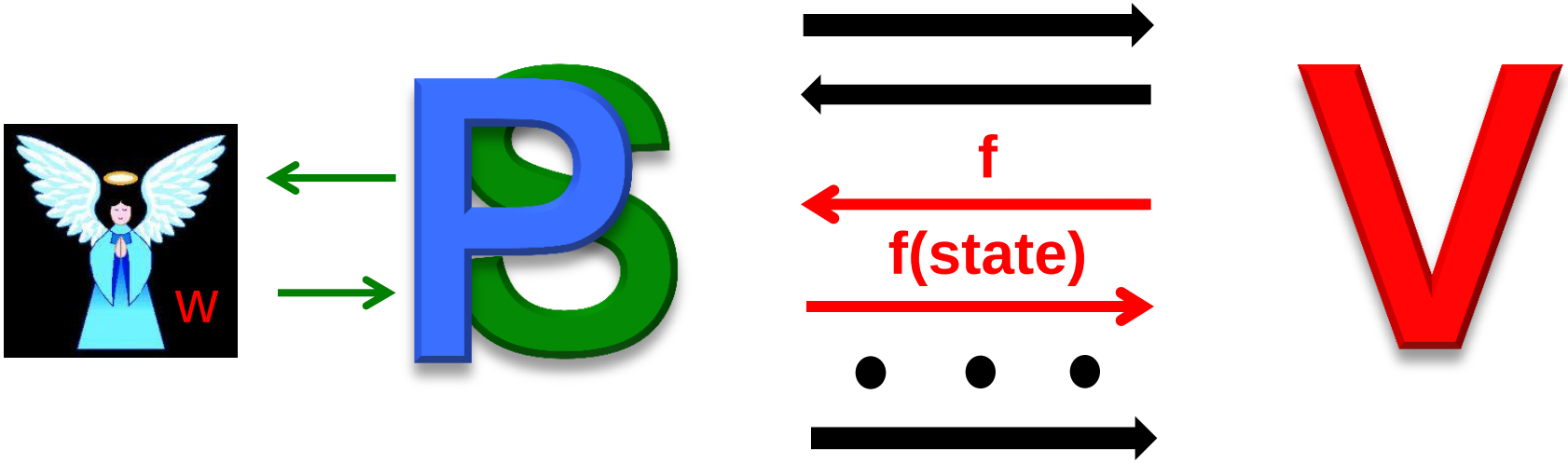
Our Results …

- Applications of LR-ZK
  - Universally Composable Secure Multi-party Computation in the "leaky token model"
    - All prior works require completely leakage-resilient tokens
  - Fully LR-Signatures in bounded leakage (and continual leakage) model
    - Recently constructed by [MTVY11, BSW11, LLW11]
    - Our scheme also secure in "noisy leakage" model

Our Results

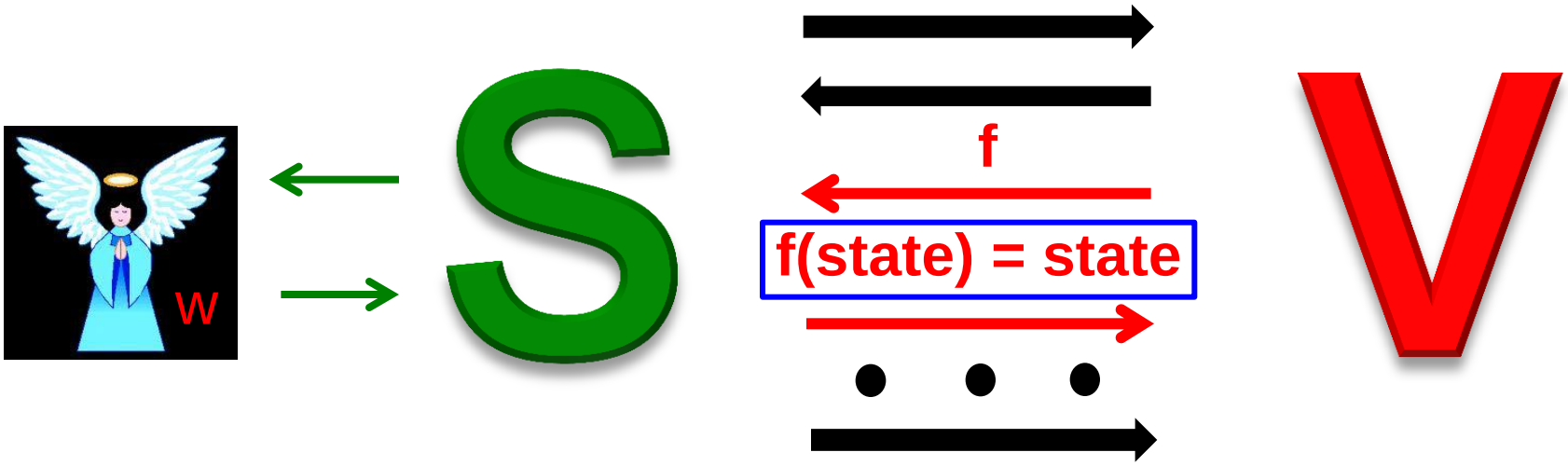
I. (1+ε)-Leakage-Resilient Zero Knowledge Proof System

Main Ideas



- f(state) must be “consistent” with past actions of **S**
- f(state) should not reveal **S** is cheating

Main Ideas …



- Same as corrupting the prover during the protocol
- **S** must "explain" its actions as an honest prover

Adaptive Security!

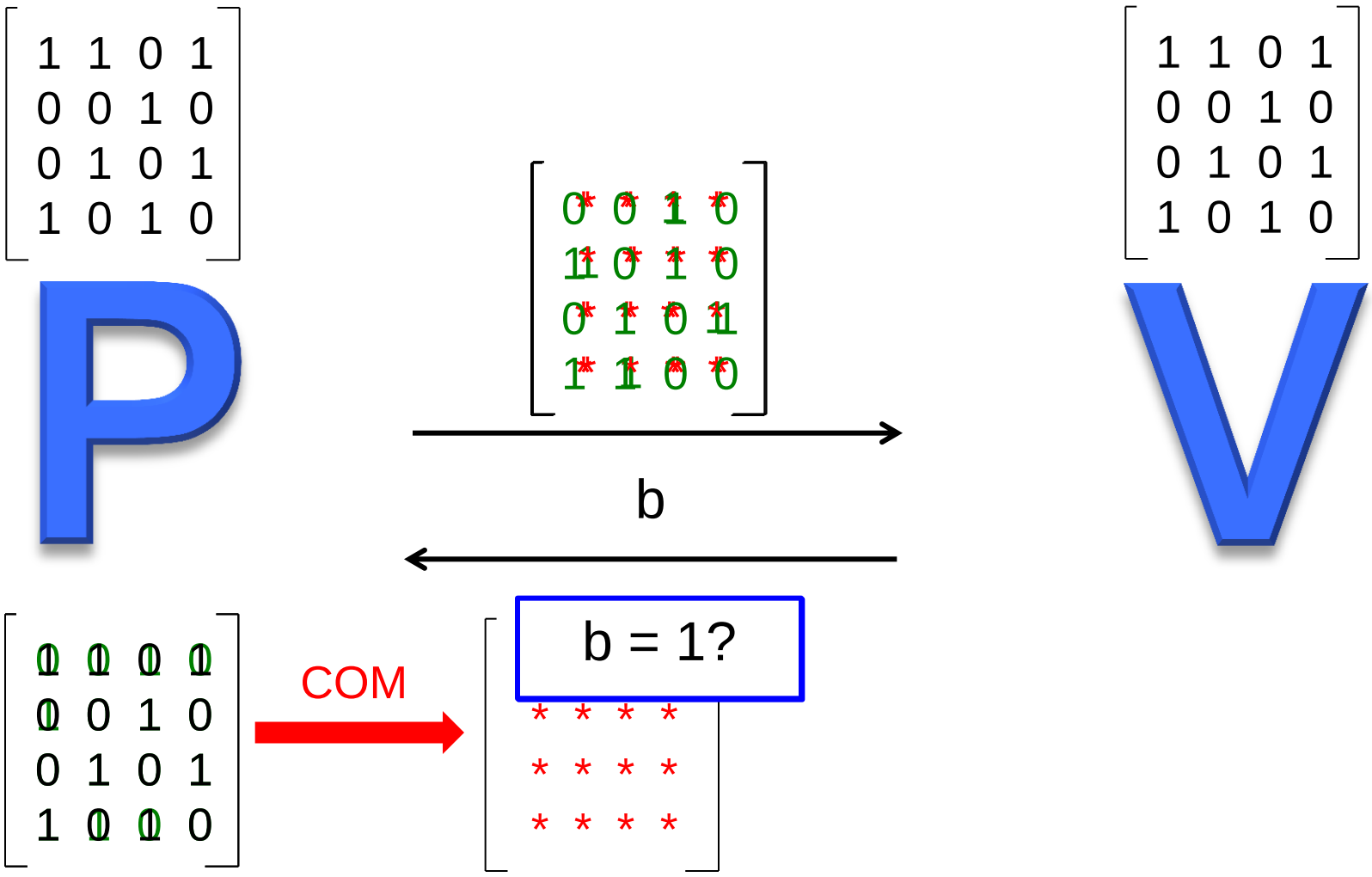
Adaptive Security [CFGN96, B96]

- Adv can corrupt parties during protocol exec
- When a party P is corrupted:
  - Adv learns entire state (input and random coins) of P
  - Given input of P, Sim must produce random coins consistent with transcript and honest P strategy
- Standard technique: equivocal commitments
  - Possible to decommit in any manner given trapdoor (otherwise binding)

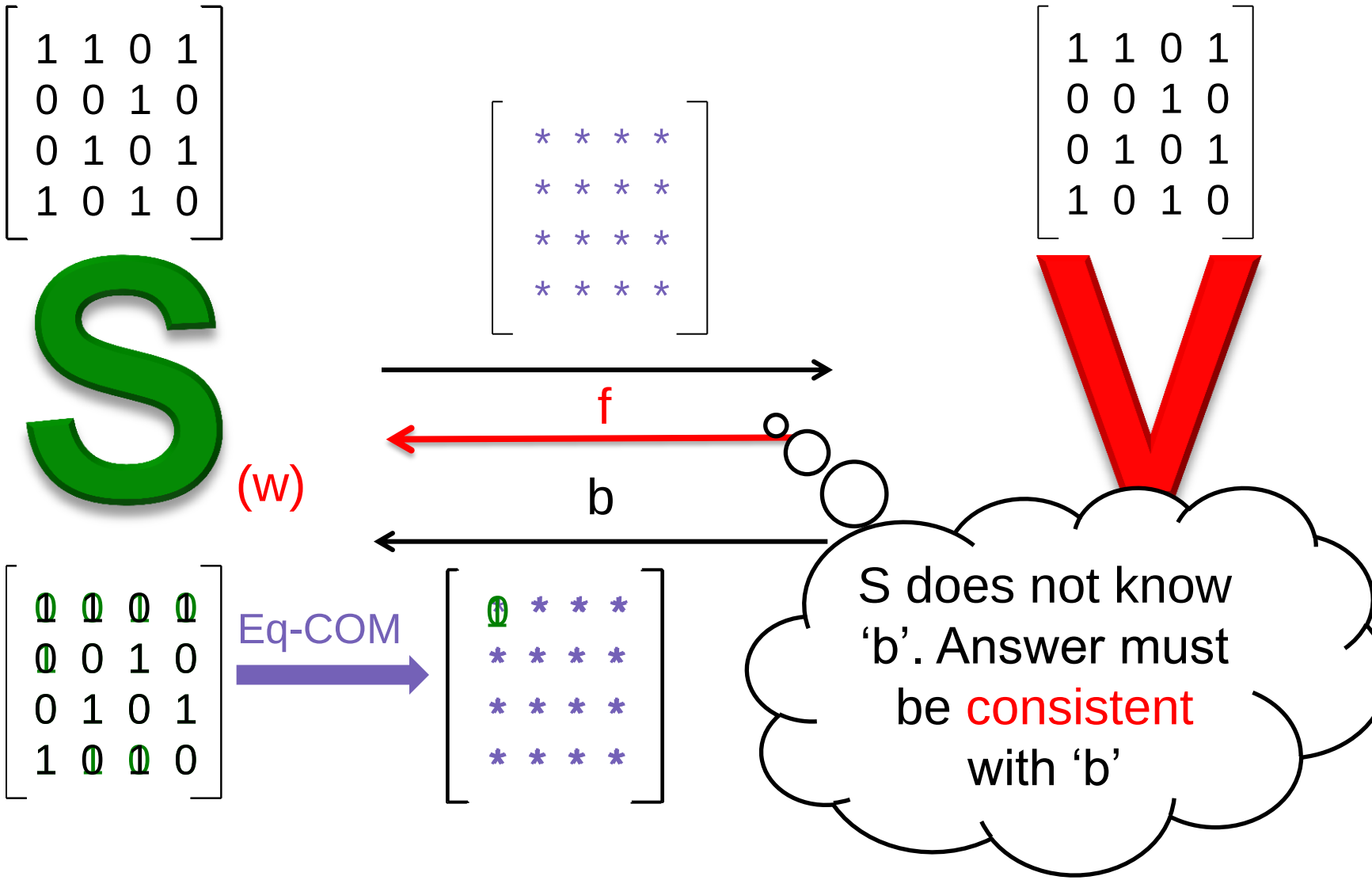
Question

Adaptive Security → LR-ZK ?

Graph Hamiltonicity



LR-ZK?

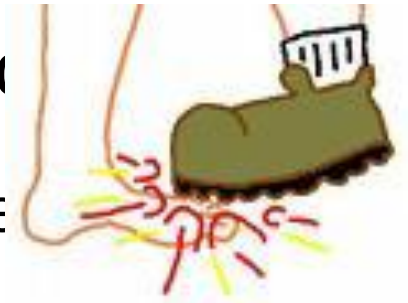

1 1 0 1
0 0 1 0
0 1 0 1
1 0 1 0
S
(w)
* * * *
* * * *
* * * *
* * * *
f
b
1 1 0 1
0 0 1 0
0 1 0 1
1 0 1 0
V
Eq-COM
S does not know 'b'. Answer must be consistent with 'b'

Adaptive security does not imply LR-ZK

- Adaptive ZK: No need to simulate P after corruption
- LR-ZK: *Must continue to simulate even after a leakage query*
  - Without knowledge of what was leaked!
  - "Future" messages must be "consistent" with leakage

Main Ideas

- Two ways for simulator to cheat (instead
  - One cheating mode to simulate protocol me
  - Another cheating mode to answer leakage queries
- Extract V's challenge for simulation of messages
- Precise Simulation [MP06]
  - In order to bound the amount of leakage



Our Results

II. (1)-Leakage-Resilient NIZK proofs

LR-NIZK

- Adaptive NIZK implies LR-NIZK
  - ➢ no “future” messages to simulate after leakage

A NIZK proof with “adaptive security” [GOS06] is also a LR-NIZK proof system

(GOS NIZK proof system is leakage-resilient)

Thank You!