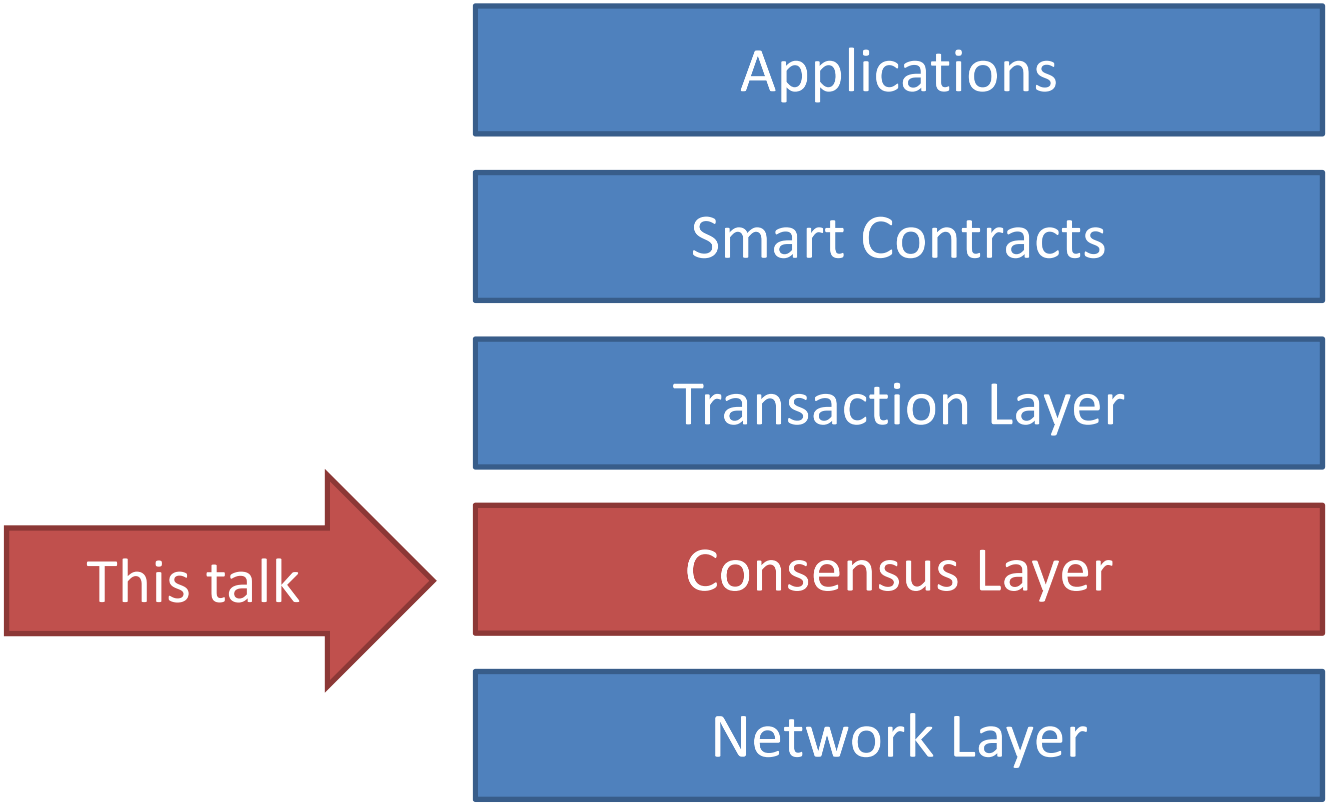
Proofs of Replicated Storage Without Timing Assumptions

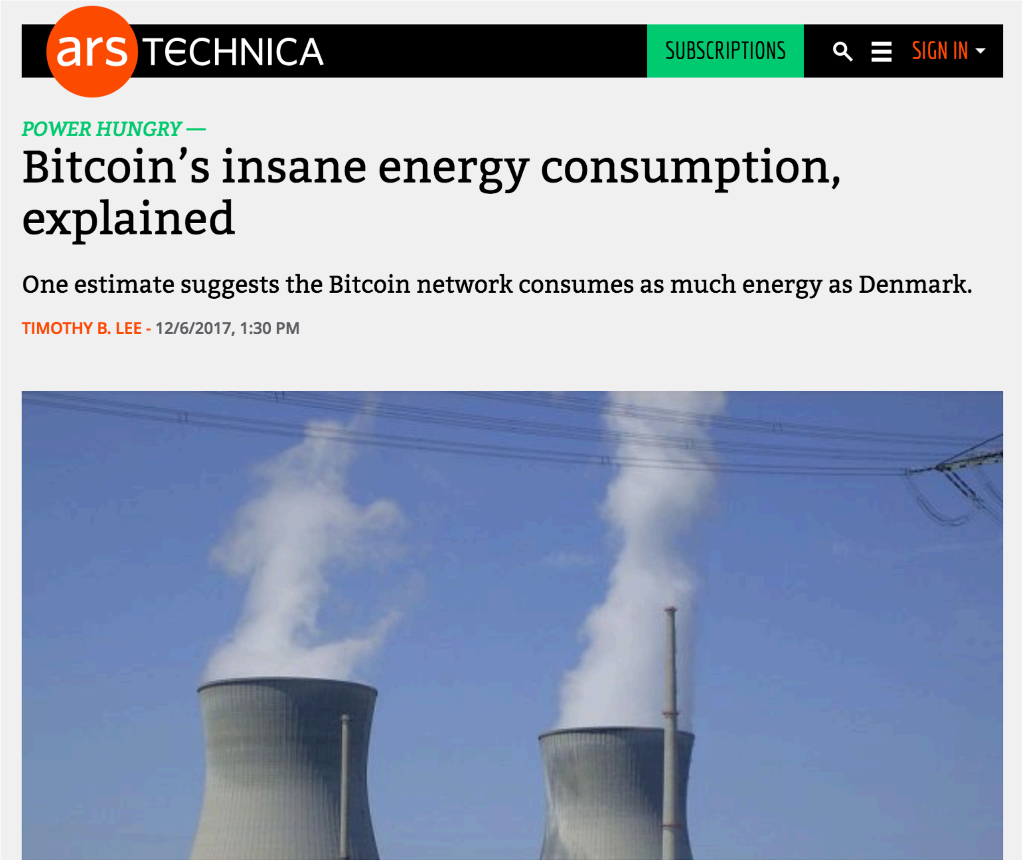
Ivan Damgård, Chaya Ganesh,
Claudio Orlandi

@claudiorlandi

Blockchain Research



Motivation…



POWER HUNGRY —

Bitcoin’s insane energy consumption, explained

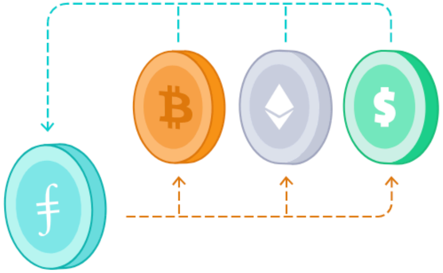
One estimate suggests the Bitcoin network consumes as much energy as Denmark.

TIMOTHY B. LEE - 12/6/2017, 1:30 PM

- Proof of Work is wasteful!
- Why not do “proofs of something useful?”

Filecoin

A MASSIVE AMOUNT OF STORAGE SITS UNUSED IN DATA CENTERS AND HARD DRIVES AROUND THE WORLD.



EARN FILECOIN FOR HOSTING FILES

Put your unused storage to work by becoming a Filecoin miner. Use the Filecoin mining software to get paid for fulfilling storage requests on the Filecoin market.

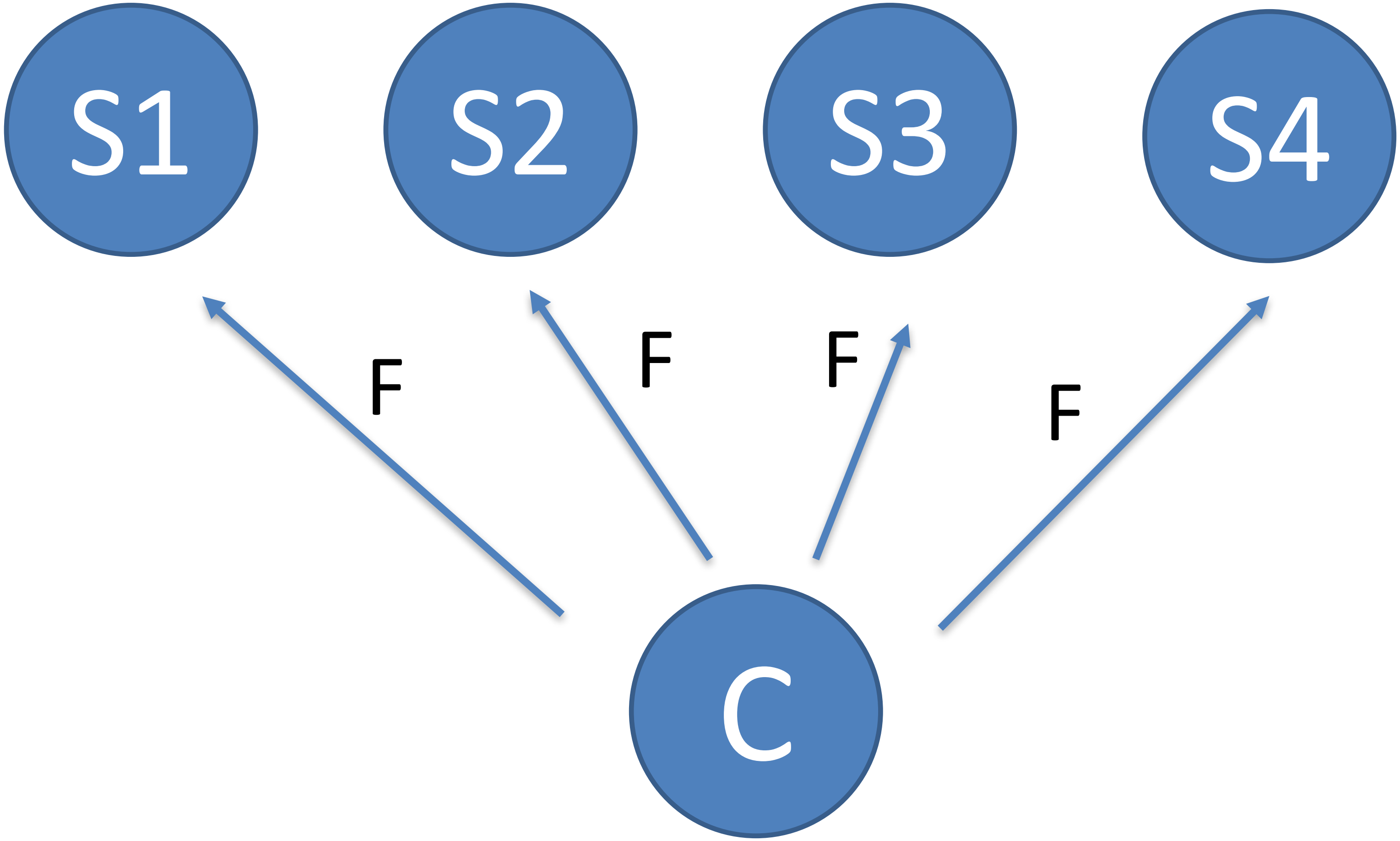
EXCHANGE FILECOIN FOR USD, BTC, ETH AND MORE

The Filecoin currency will be traded on a number of exchanges and supported by multiple cryptocurrency wallets, allowing you to easily exchange Filecoin for other currencies like US Dollars, Bitcoin, and Ether.

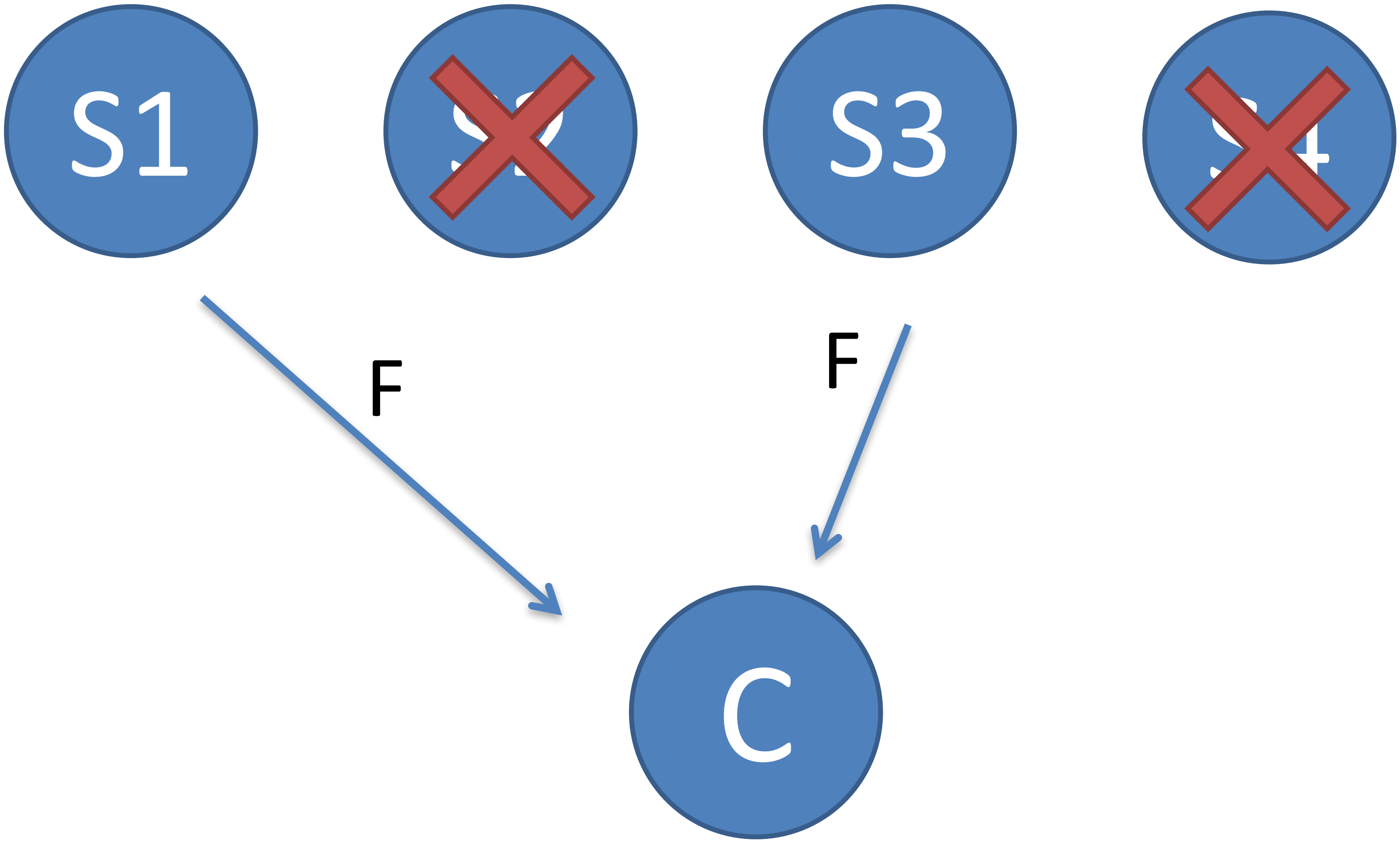
RELIABLY STORE FILES AT HYPERCOMPETITIVE PRICES

Clients can tune their storage strategy to suit their needs, creating a custom balance between redundancy, speed of retrieval, and cost. The worldwide Filecoin storage and retrieval markets make vendors compete to give you flexible options at the best prices.

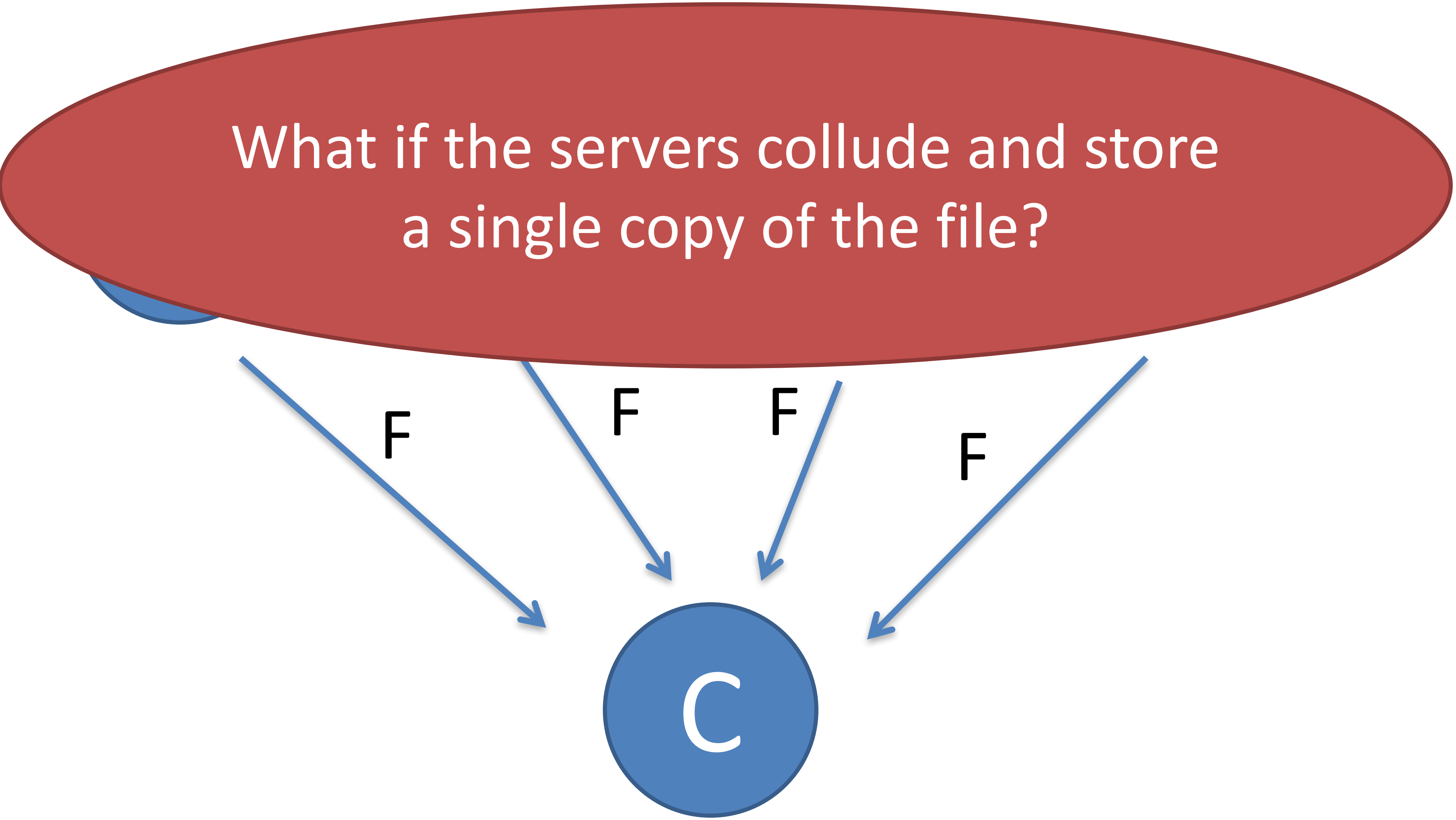
Replicated Storage



Replicated Storage



Replicated Storage



What if the servers collude and store a single copy of the file?

Related Concepts

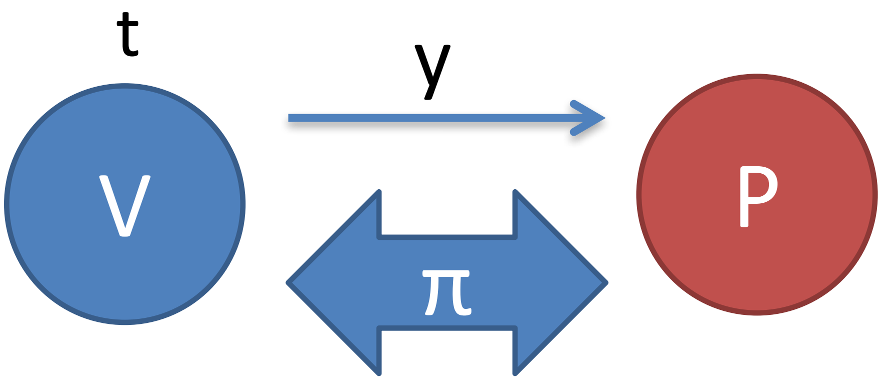
- **Proof of Space** [DFKP15], [ABFG14]
  - Proves that some space has been wasted
- **Proof of Catalytic Space** [Pie18]
  - Proves that some space has been used - without wasting it
- **Proof of Retrievability** [JK07], [SW08], [DVW09]…
  - Proves that a specific file is being stored!

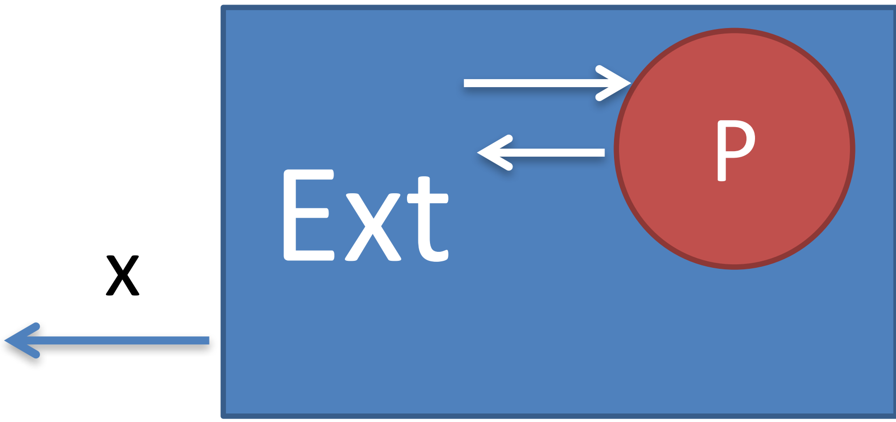
Proof of Retrievability

- Store(x) → (t,y)
- P(y) ⇄ V(t) → 0/1

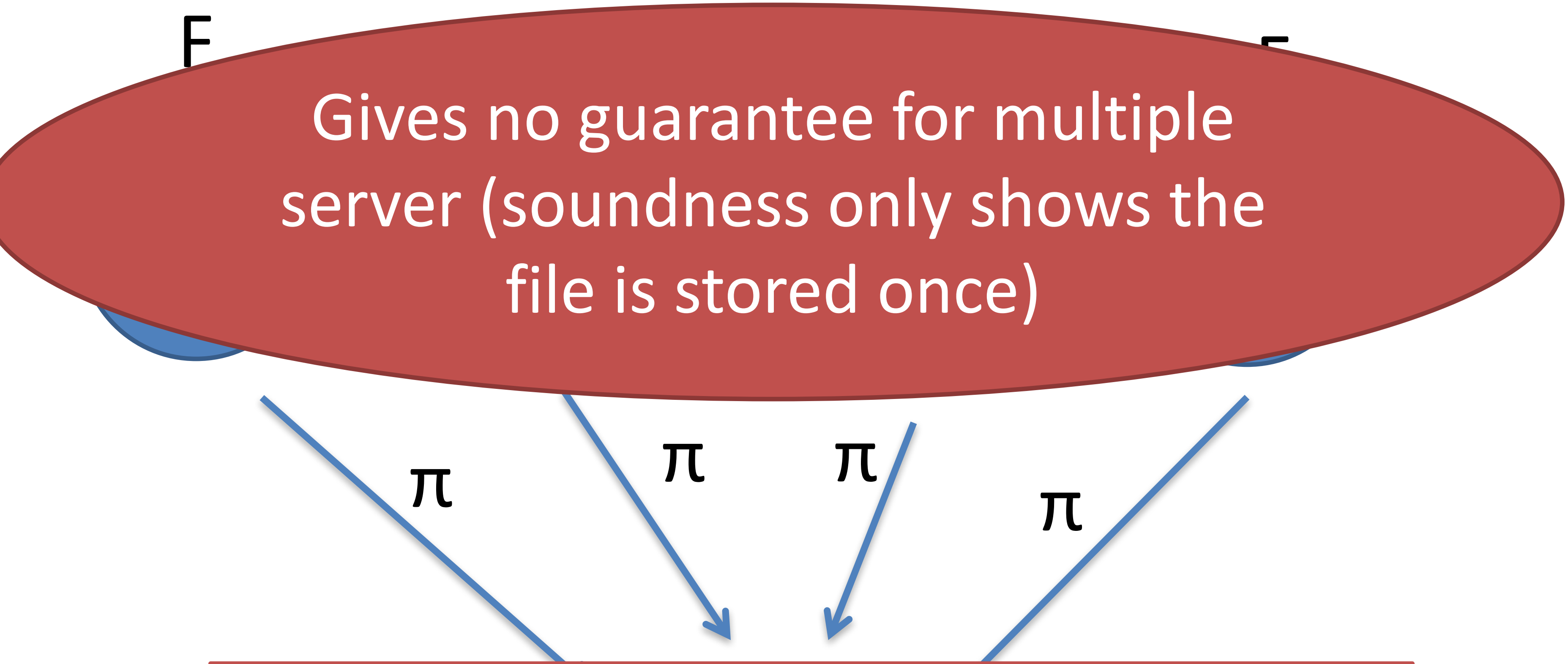
- |proof|< |x|

- **Soundness**:
  if verifier accepts, the file can be extracted





Proof of Retrievability



Gives no guarantee for multiple server (soundness only shows the file is stored once)

For the sake of this presentation, we ignore PoR from now on
(just assume retrieve = download)

Proof of Replication Requires Different Encodings

- **Encrypt everything?**
- Secure encryption looks random. Cannot be de-duplicated. ☺
- Requires client to store secret state. ☹
- Cannot be publicly verified ☹

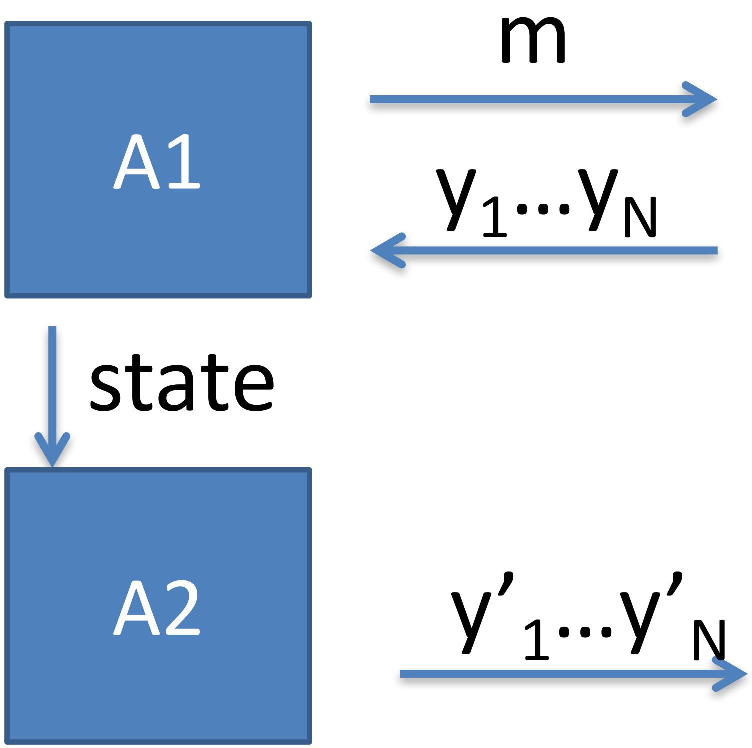
- **Slow Encodings?**
- Enc is “slow” to compute
  - [FileCoin], [Pie18], [BF?].
- Accept proof only if prover is “fast” → if prover is not storing file, proof will fail ☺
- Requires timing assumption ☹

Our results: Replica Encoding and Proofs of Replicated Storage without Timing Assumptions

Replica Encoding

- rEnc(m,r) → y
- rDec(y) → m

- **Soundness**:



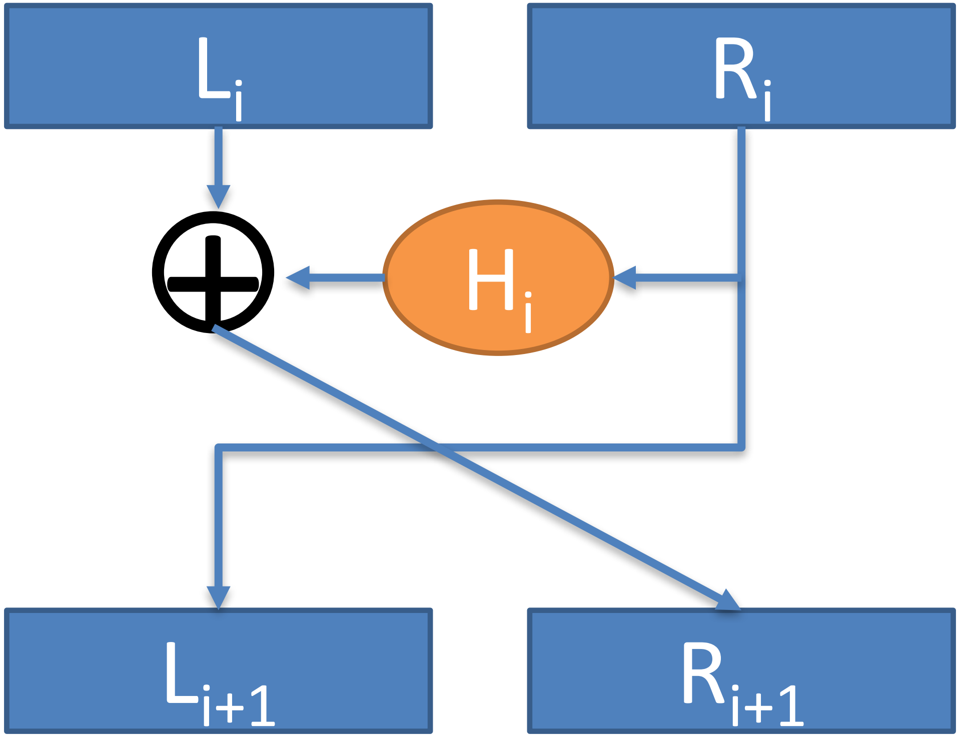
(A1,A2) wins if $|state| < c\ |y|\ N'$

Arbitrary constant < 1

i: $y'_i = y_i$

Building Replica Encoding: Tools

- **T is an invertible Random Oracle**
- *(T for "All-or-Nothing Transform")*
  - E.g., many rounds Feistel Cipher using RO H



Building Replica Encoding: Tools

- **(E,D) is a trapdoor permutation**
  - E.g, RSA
  - The function E is public

    $$E(x) = x^e \bmod N = y$$

  - The function D is secret

    $$D(y) = y^d \bmod N = x$$

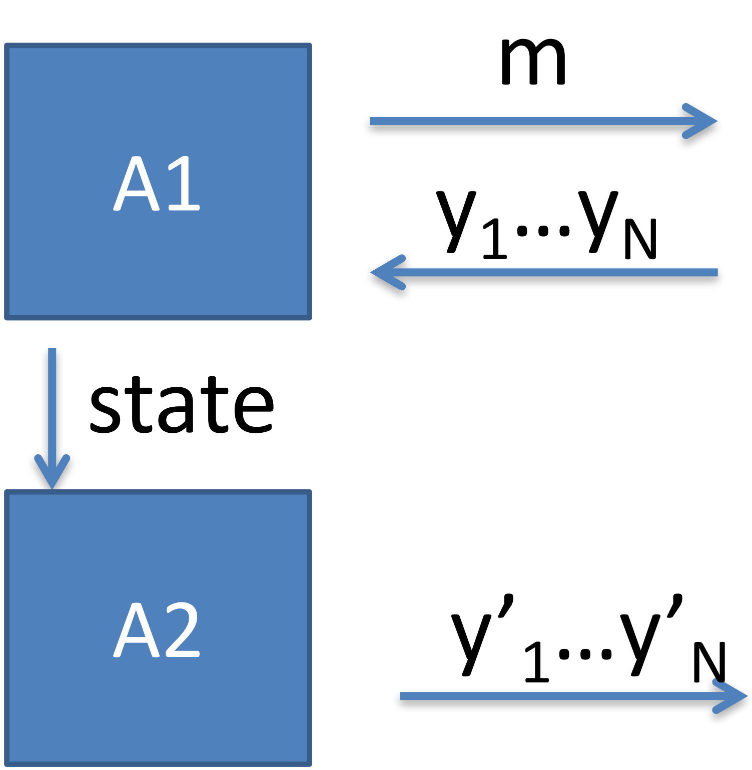
Replica Encoding: first attempt

- rEnc(m,r) :
  - (E,D)← Gen()
  - x = (m,r)
  - t = T(x)
  - z = D(t)
  - Output y=(z,E)

- rDec(y)
  - Parse y=(z,E)
  - t = E(z)
  - x = T^{-1}(t)
  - Parse x=(m,r)
  - Output m

Soundness?

- rEnc(m,r) → y
- rDec(y) → m
- **Soundness**:

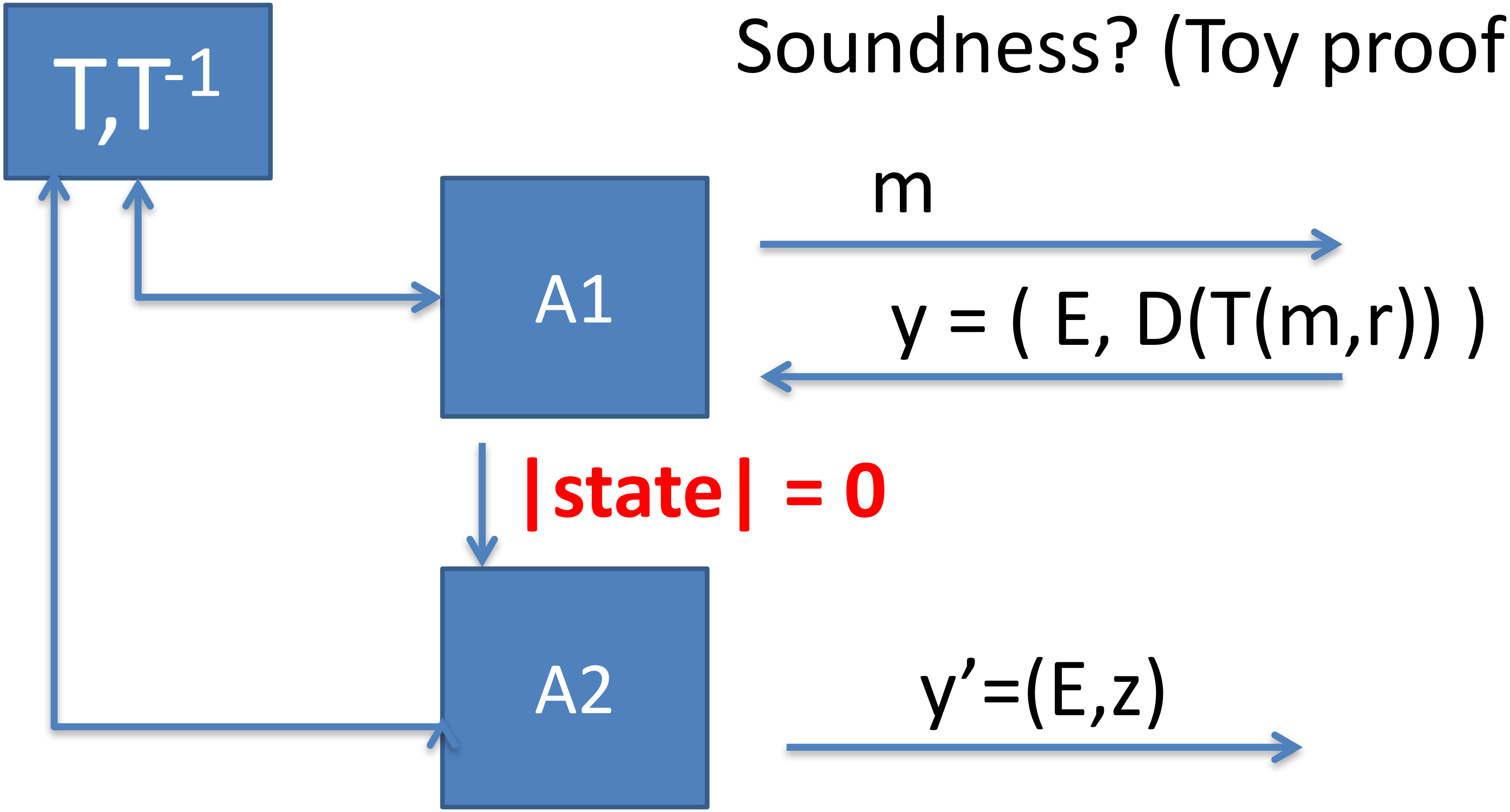


Arbitrary constant < 1

(A1,A2) wins if $|\text{state}| < c\ |y|\ N'$

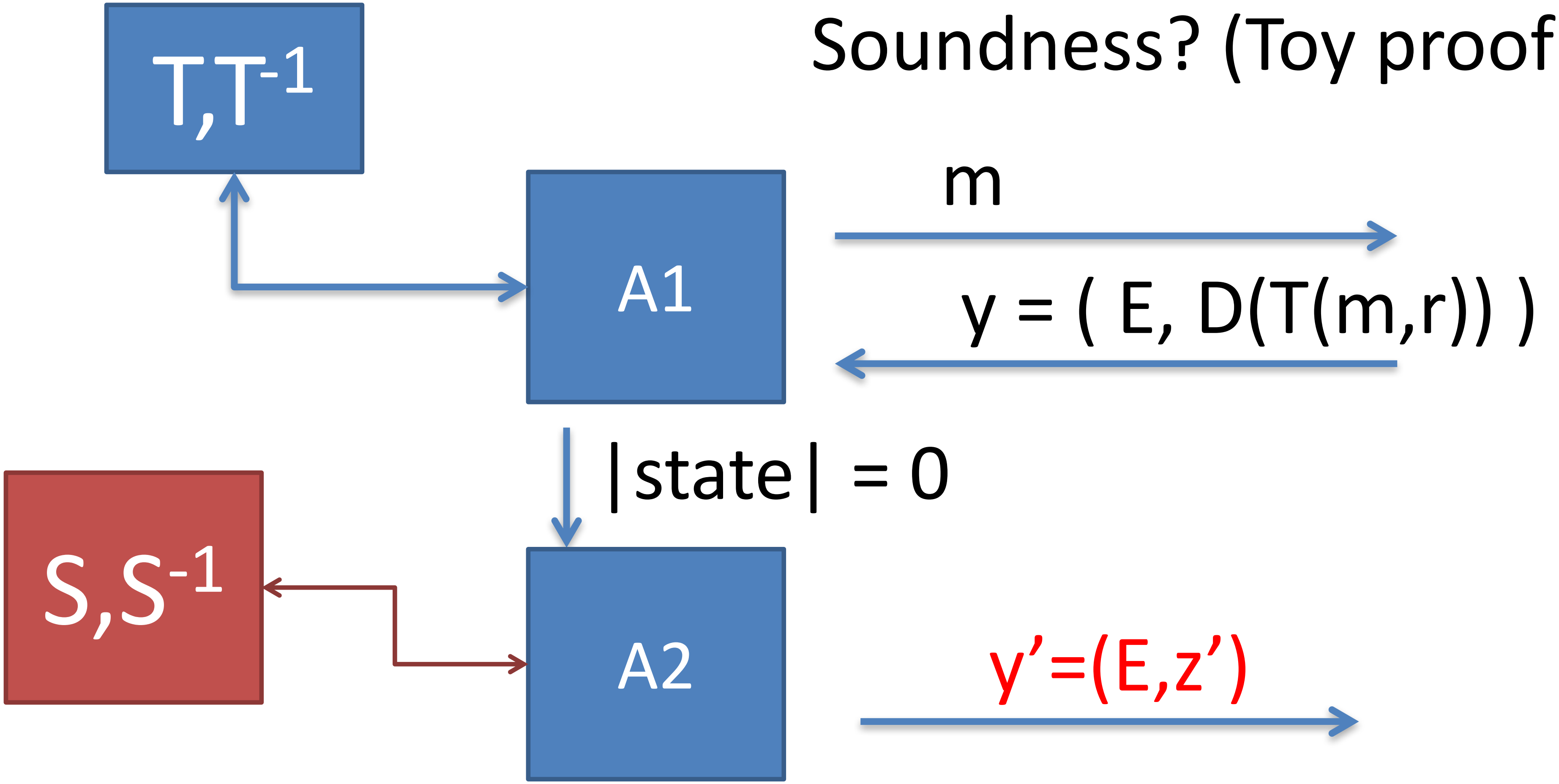
i: $y'_i = y_i$

Soundness? (Toy proof)



- A1,A2 win → y=y'
  → E(z)=T(m,r) is a random number
  → Since |state|=0 and incompressibility
  → A2 **must** query T on (m,r) to produce z

Soundness? (Toy proof)



- We can now use A2 to invert a TDP challenge c
  → |state|=0 → A2 can't remember T(m,r)
  → Program the 2nd RO S(m,r)=c *!=T(m,r)*
  → If (A1,A2) wins soundness → z' : E(z')=c

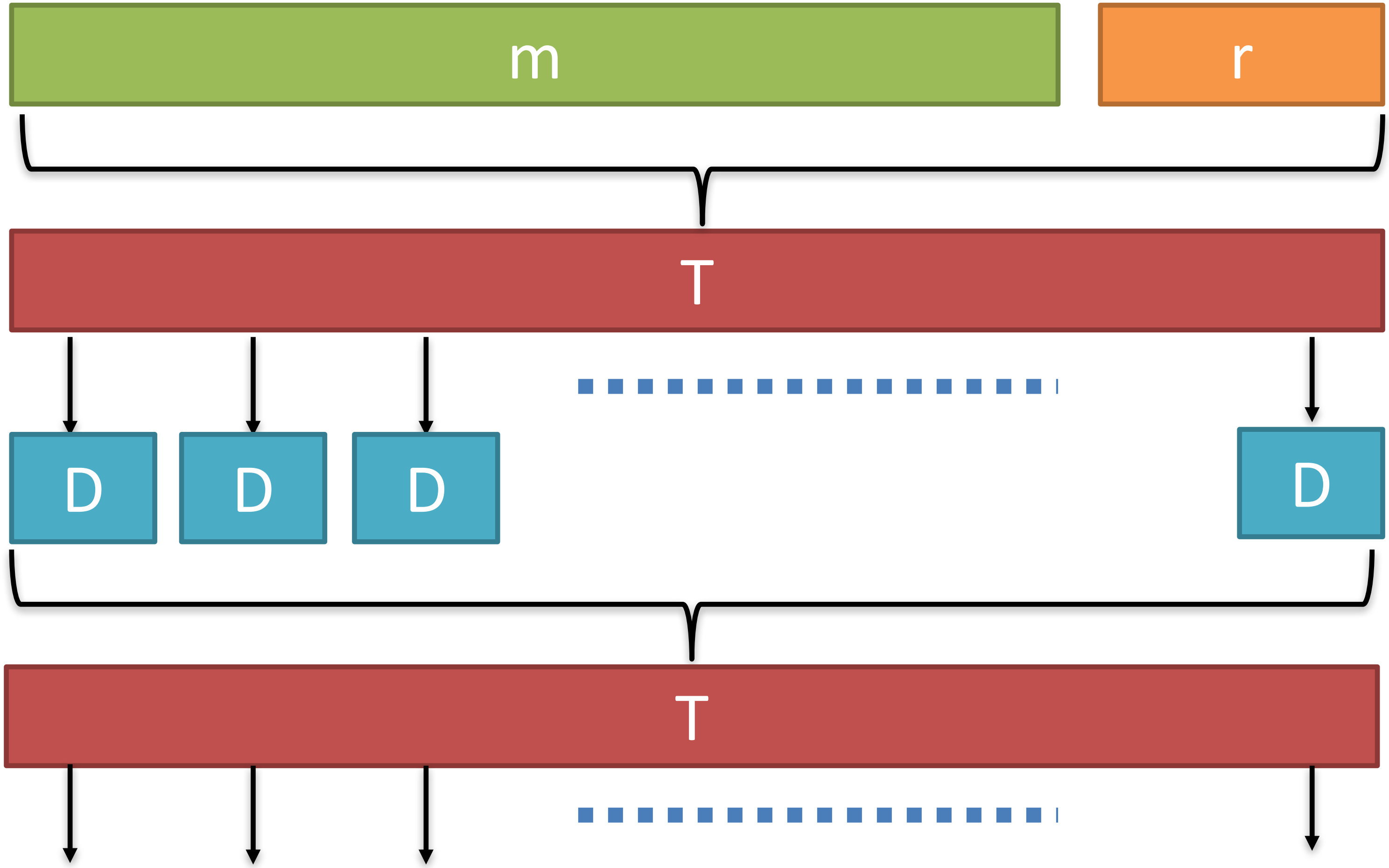
What if |state|> 0 ?

- If |state|> 0 the adversary may store arbitrary information about the preimage of D(c)
  → we cannot embed an RSA challenge in the RO queries!

- Idea: repeat encoding for many rounds
  - y' = (E, D(T(…(D(T(m,r))…)))

- If *#rounds > c #replicas*, there must be at least one query from the RO that the adversary "forgot"
  → use that to embed the RSA challenge.

- How to deal with large files
  - If |m| > RSA modulo
  - Split in block, and use “all or nothing transform”



Conclusion

- We provide the first **Replica Encoding** which does not require timing assumptions, and that can be publicly decoded.
  - Based on simple tools: RSA and RO
- **Replica Encoding** + **Proof of Retrievability** = **Proof of Replicated Storage**
- Our encoding requires many rounds: can you come up with a more efficient version?

Thanks!