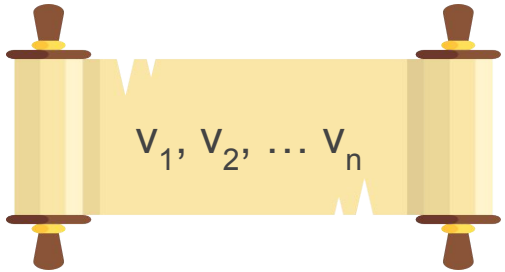
Vector Commitments

and Applications

Anca Nitulescu

Protocol Labs

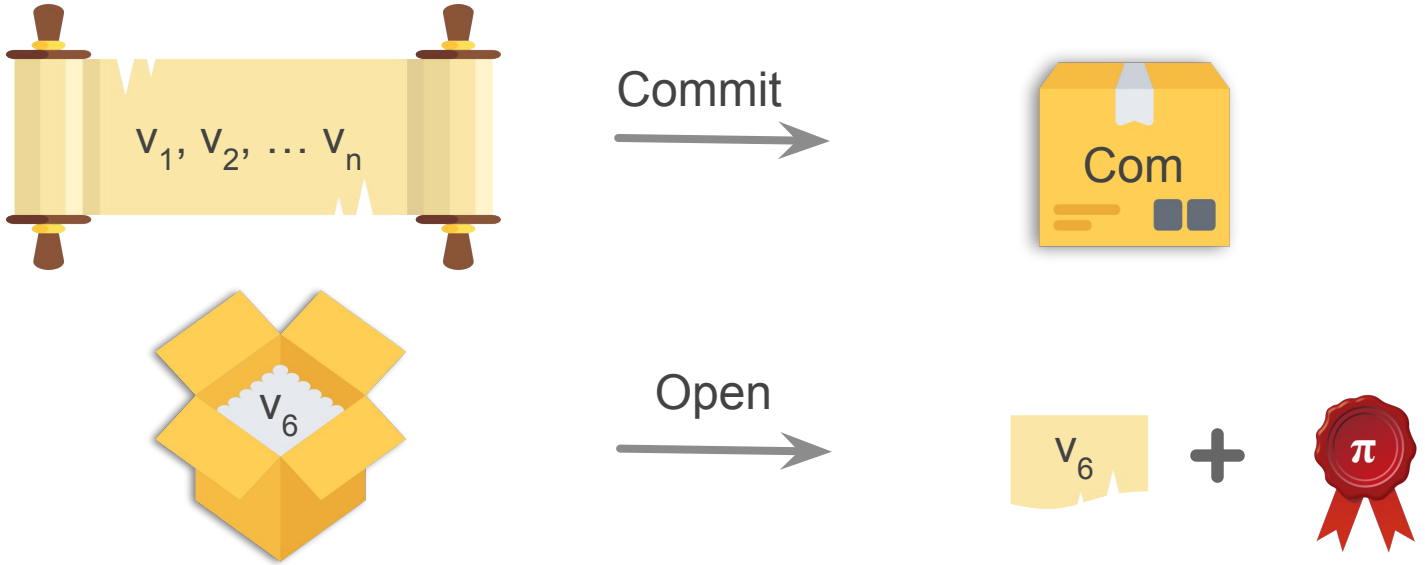
Vector Commitments



Commit

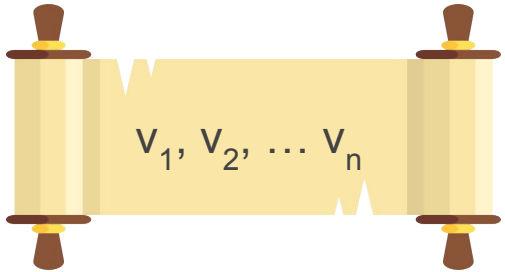
Vector Commitments and their Applications, by Dario Catalano and Dario Fiore

Vector Commitments

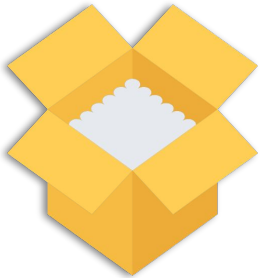


Vector Commitments and their Applications, by Dario Catalano and Dario Fiore

SubVector Commitment

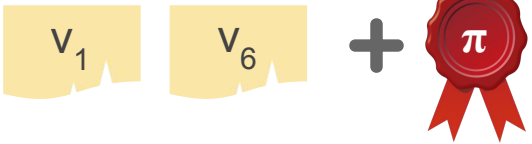


Commit

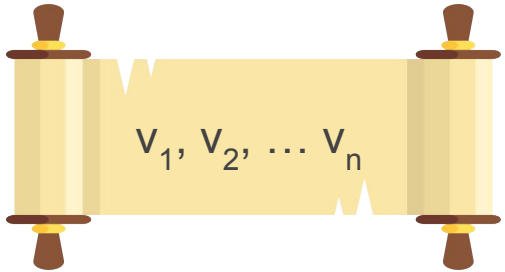


Open

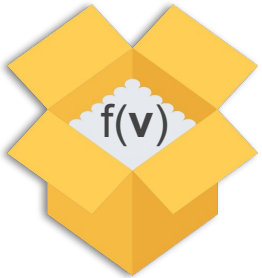
subvector



Functional VC

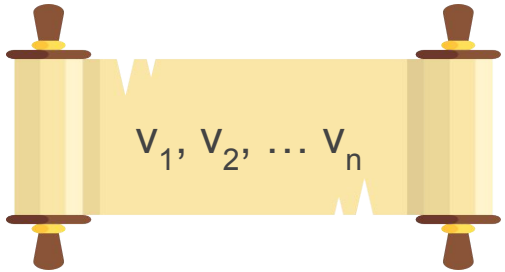


Commit

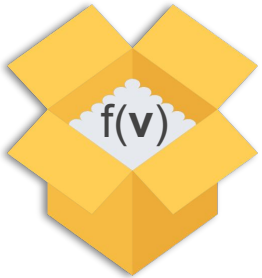


Open

Linear-Map VC



Commit



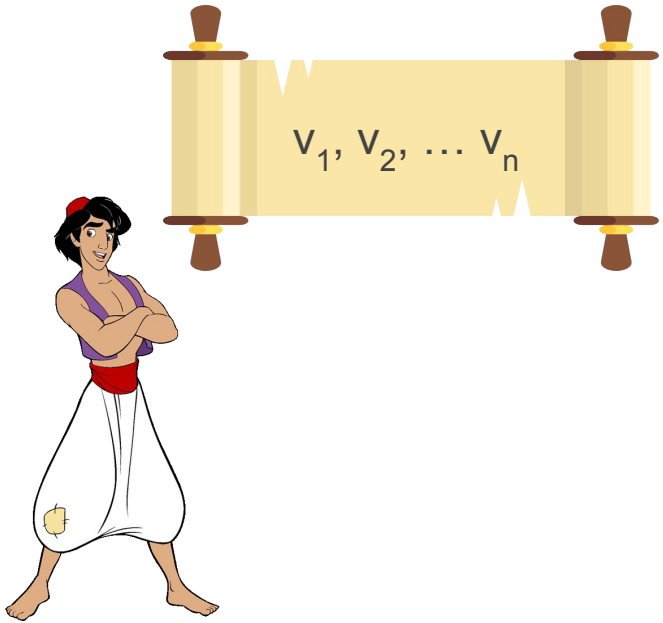
$\mathbf{v} \in \mathbb{F}^n$

Open

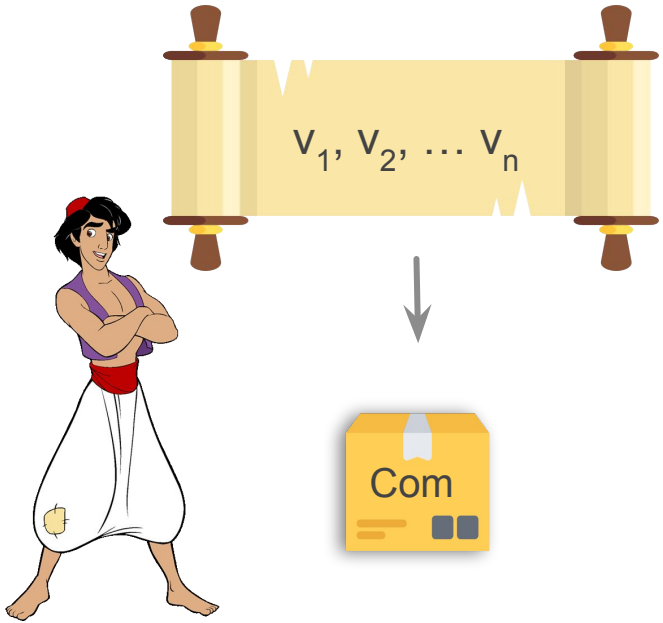
$f : \mathbb{F}^n \to \mathbb{F}^\ell$

Motivation

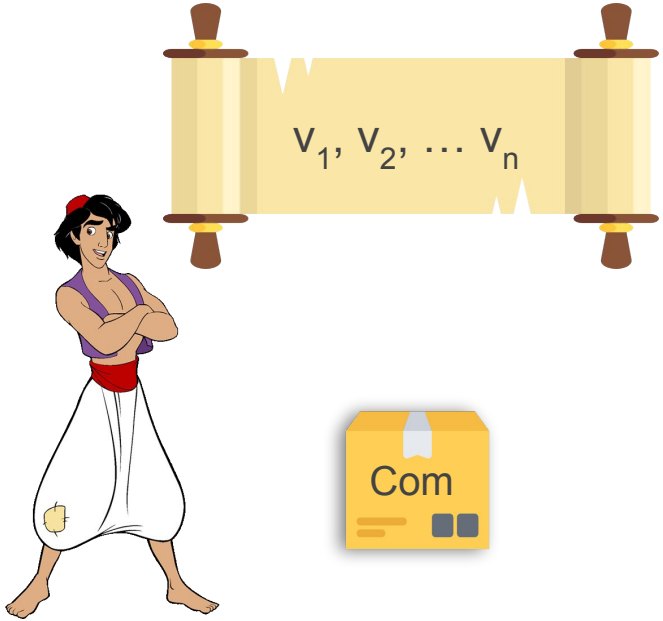
Storage Delegation



Storage Delegation

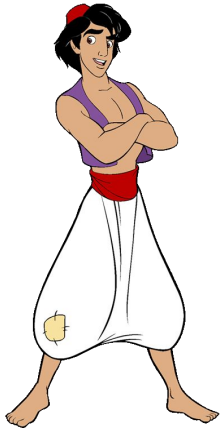


Storage Delegation

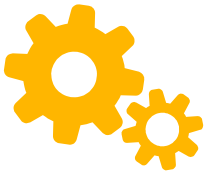


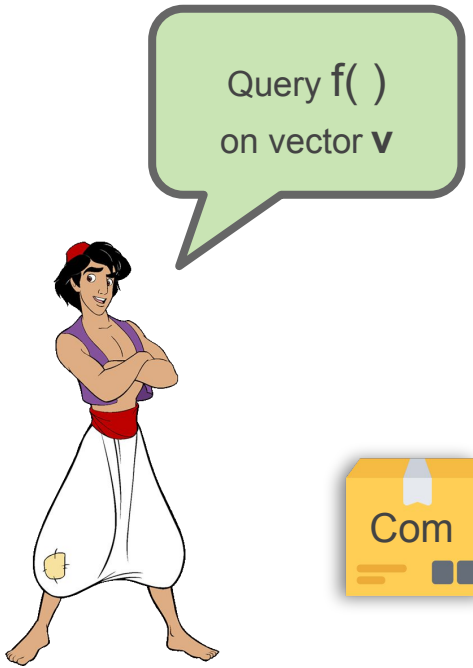
v

Storage Delegation

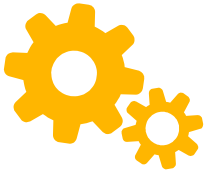


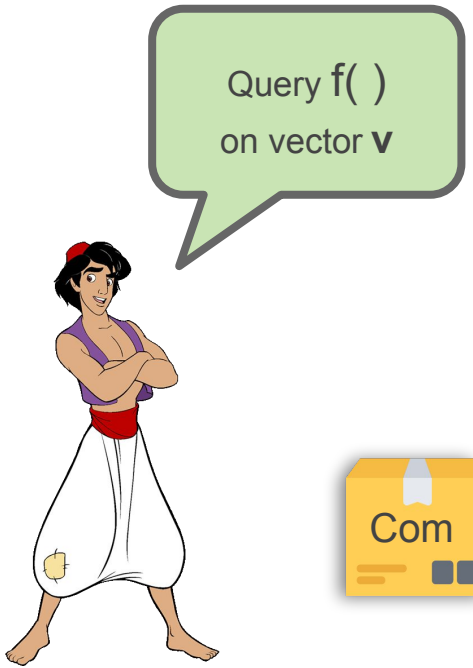
Storage Delegation

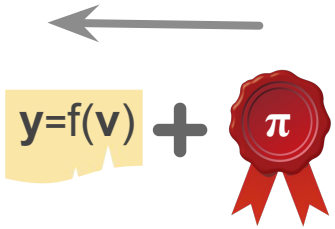




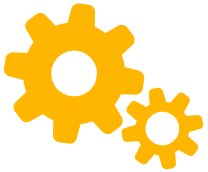
Storage Delegation

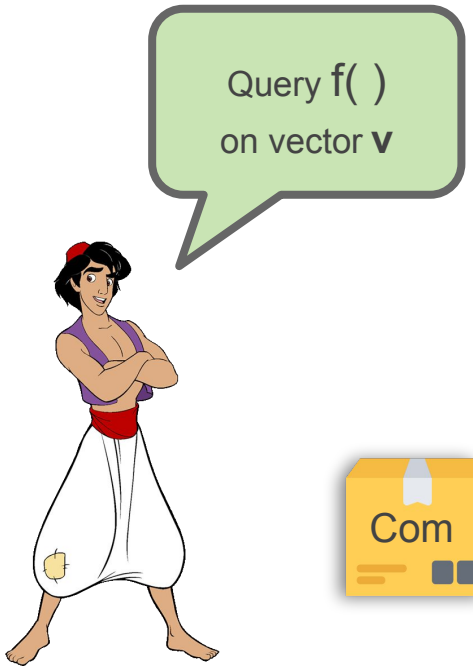


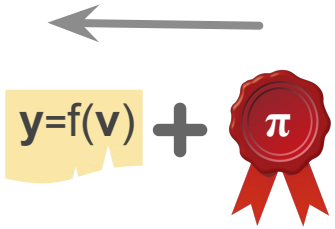




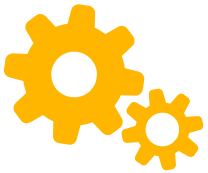
Storage Delegation

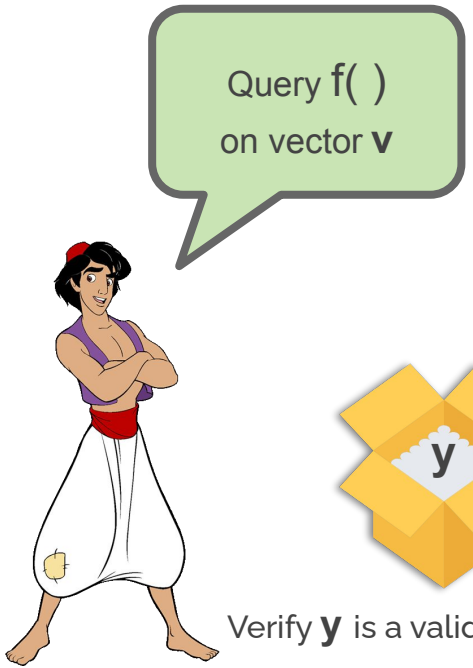






Storage Delegation





Verify **y** is a valid opening wrt Com

Proof of Space

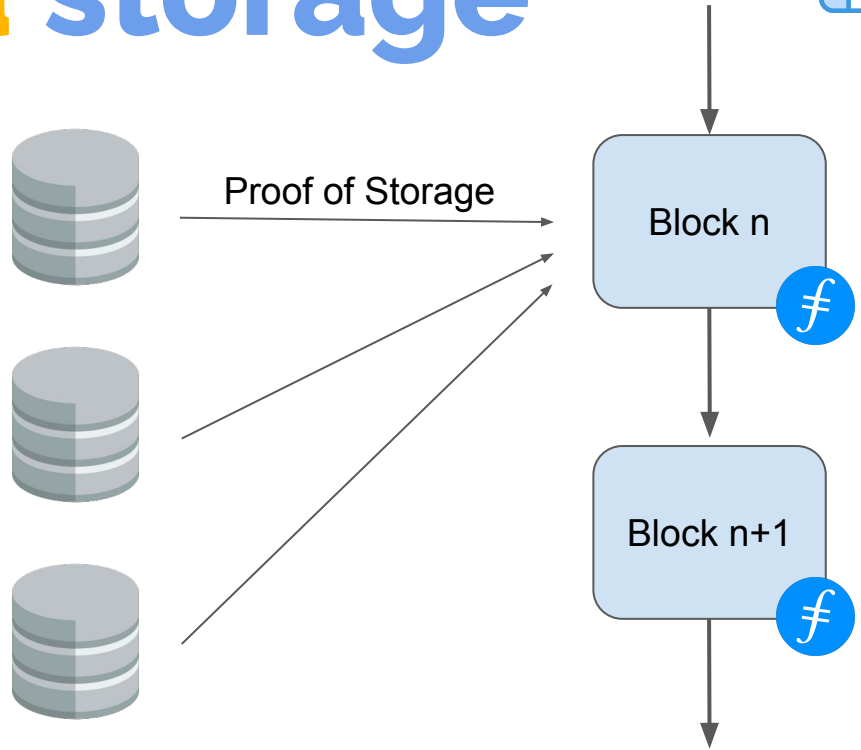
Distributed storage

Storage providers

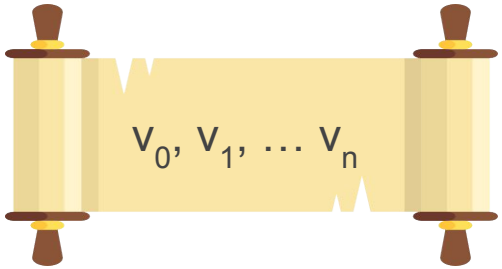
- onboard storage capacity
- earn block rewards
- encode the space on their disk
- regularly prove the storage

Nodes in network

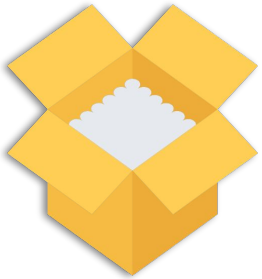
- ensure data is being stored, maintained, and secured
- regularly check proofs of storage



Tool: Vector Commitments



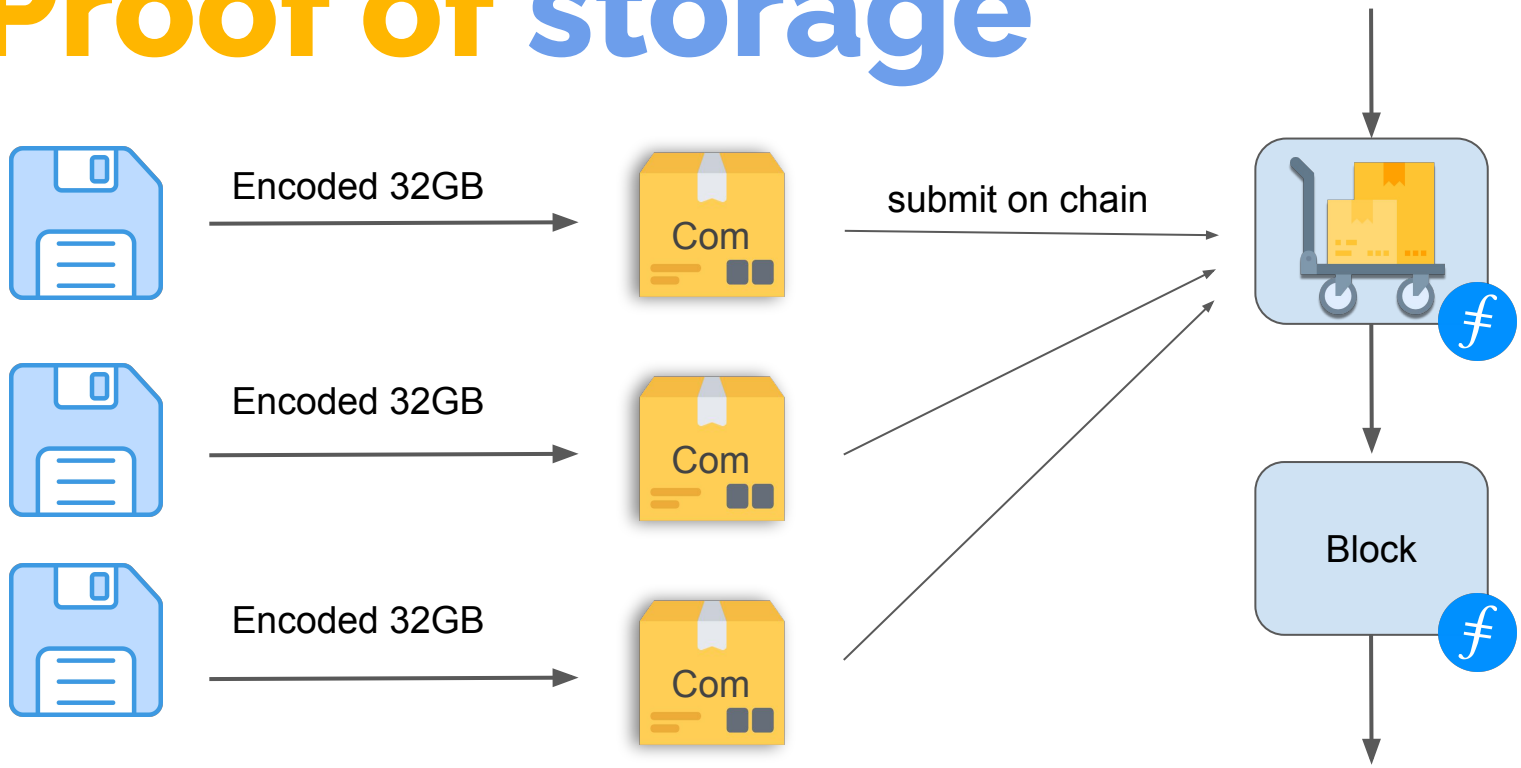
Commit



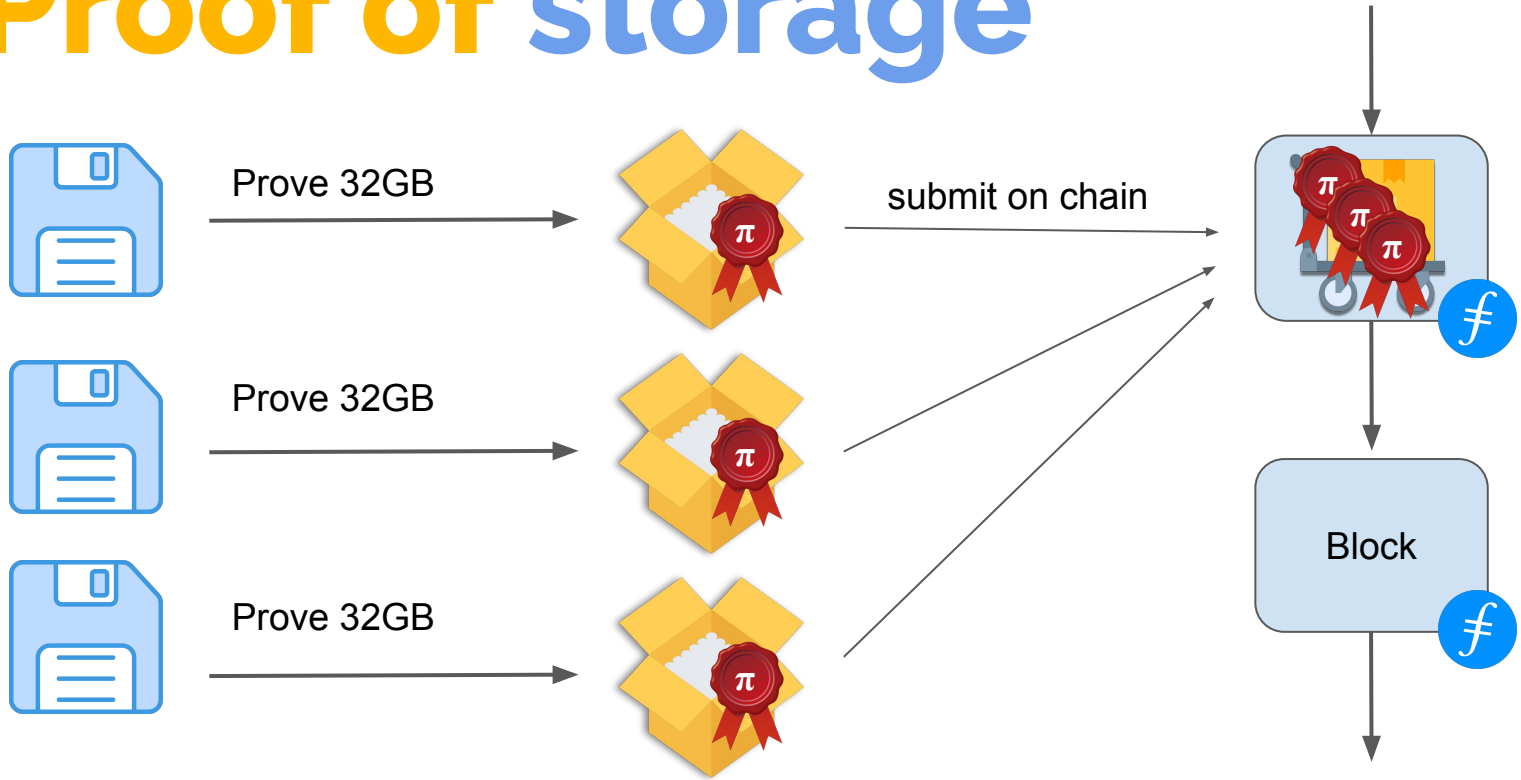
Open

Vector Commitments and their Applications, by Dario Catalano and Dario Fiore

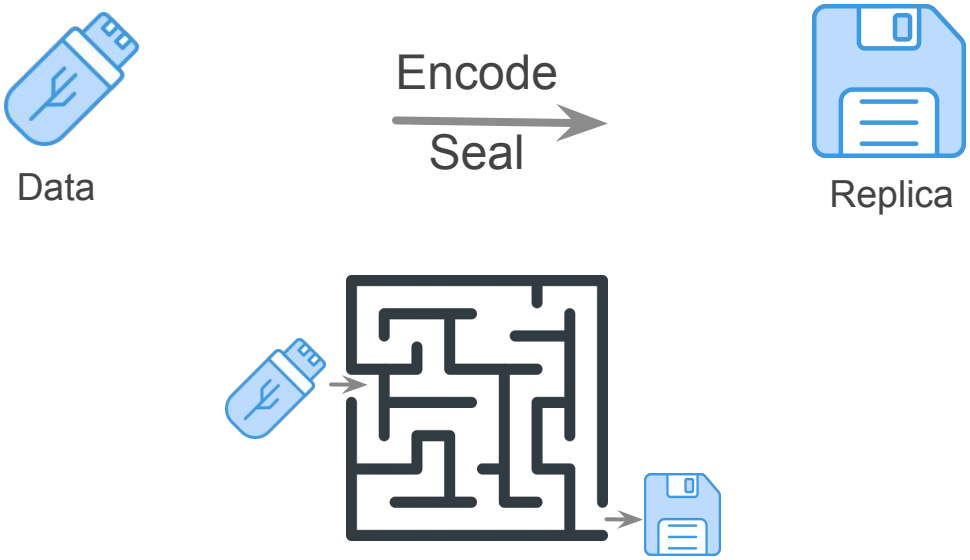
Proof of storage



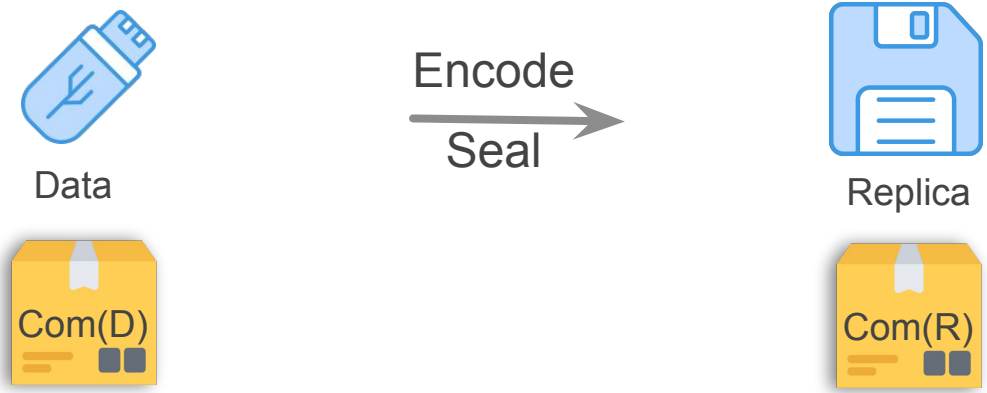
Proof of storage



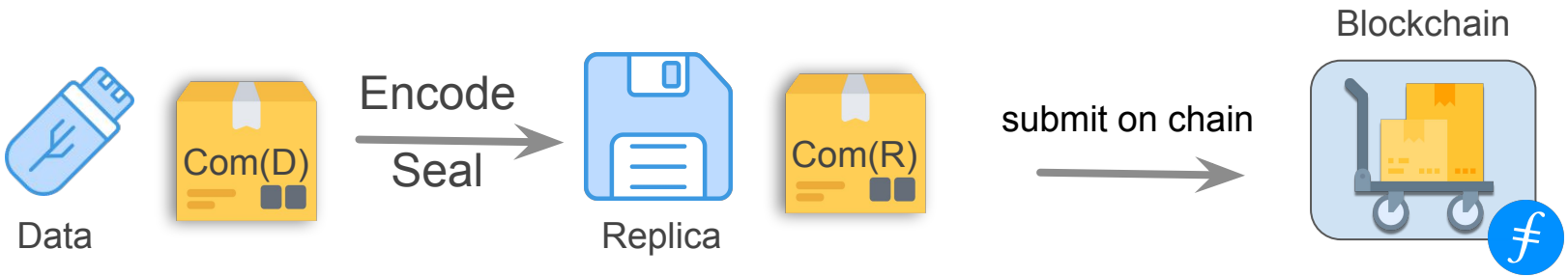
Proof of Replication



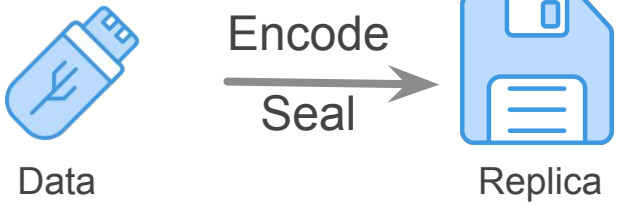
Proof of Replication

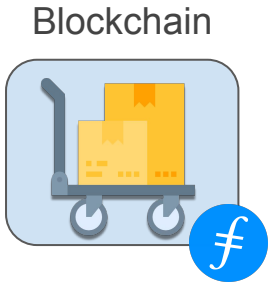


Proof of Replication

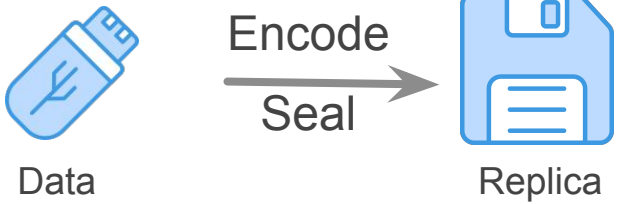


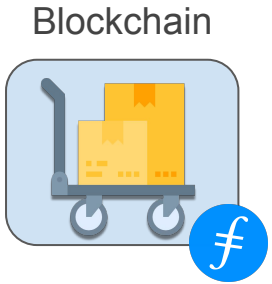
Proof of Replication

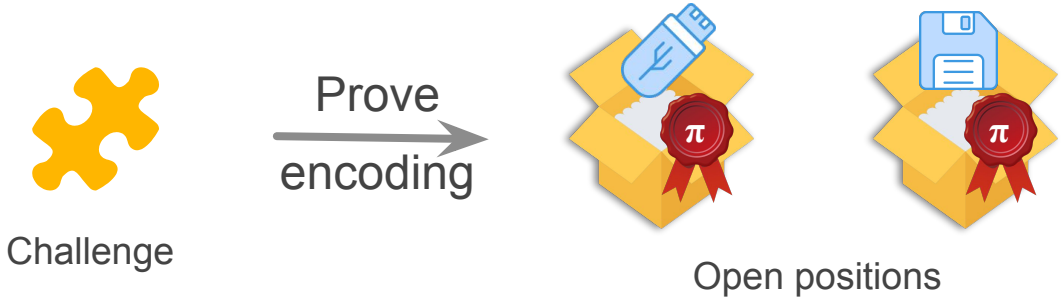




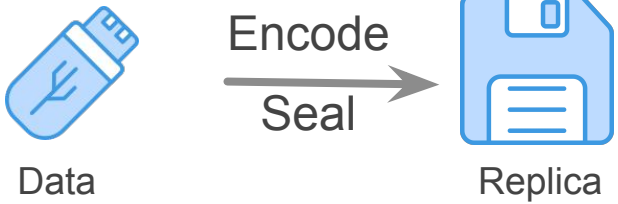
Proof of Replication

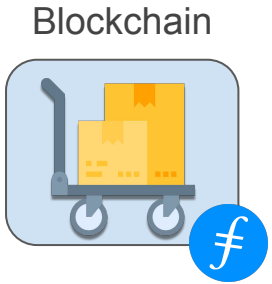


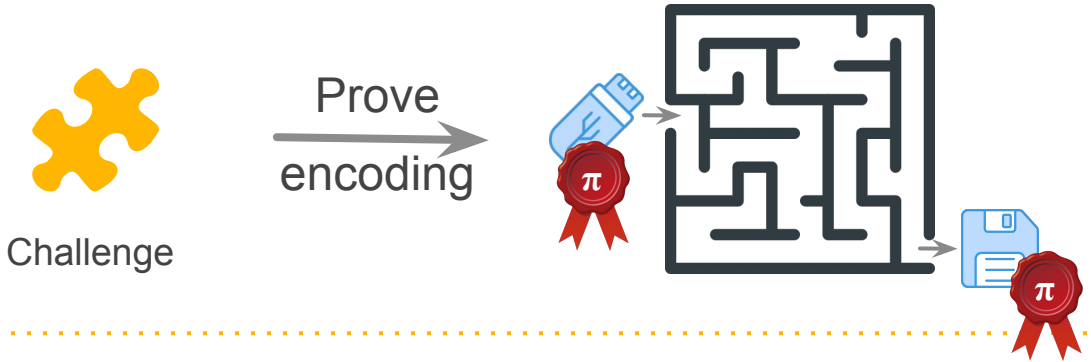




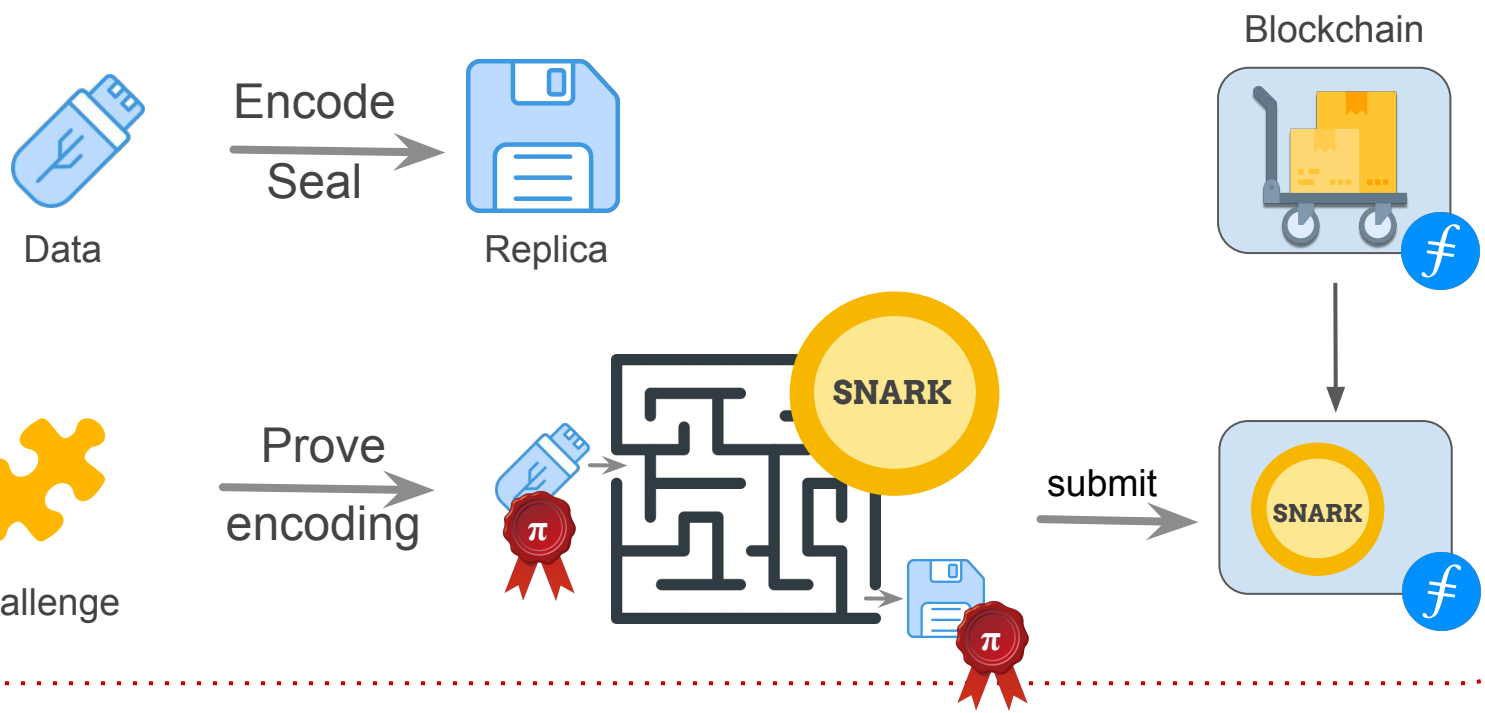
Proof of Replication



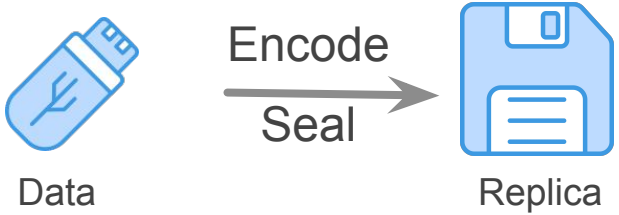


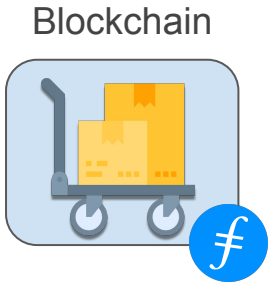


Proof of Replication



Proof of SpaceTime

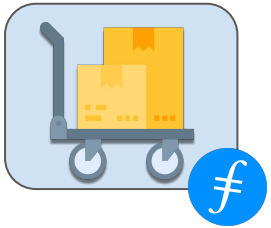




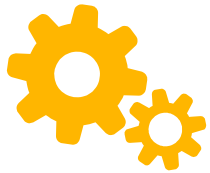
Proof of SpaceTime

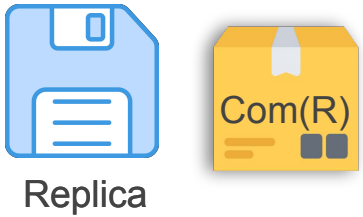
Replica

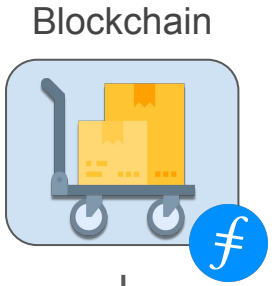
Blockchain



Proof of SpaceTime







Challenge

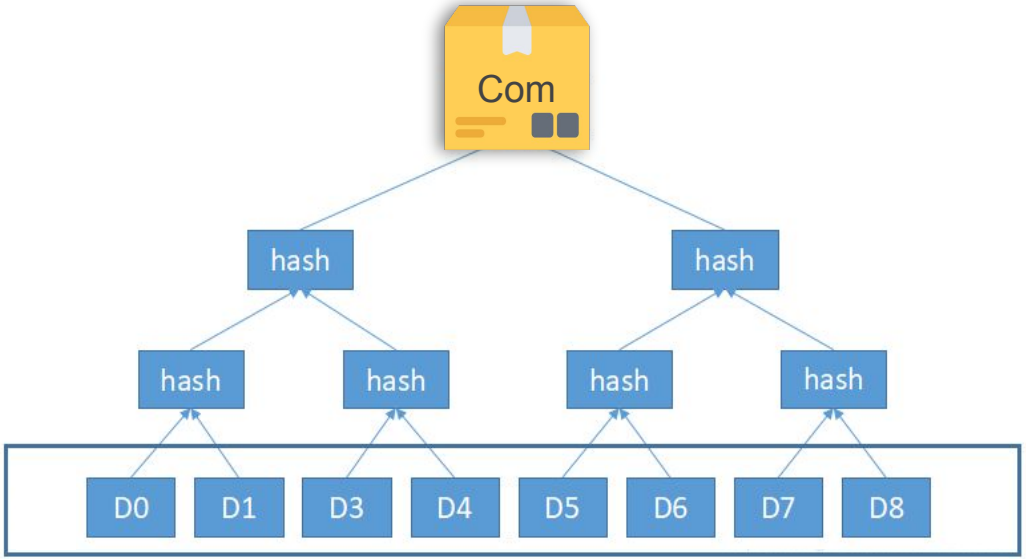
Prove

storage

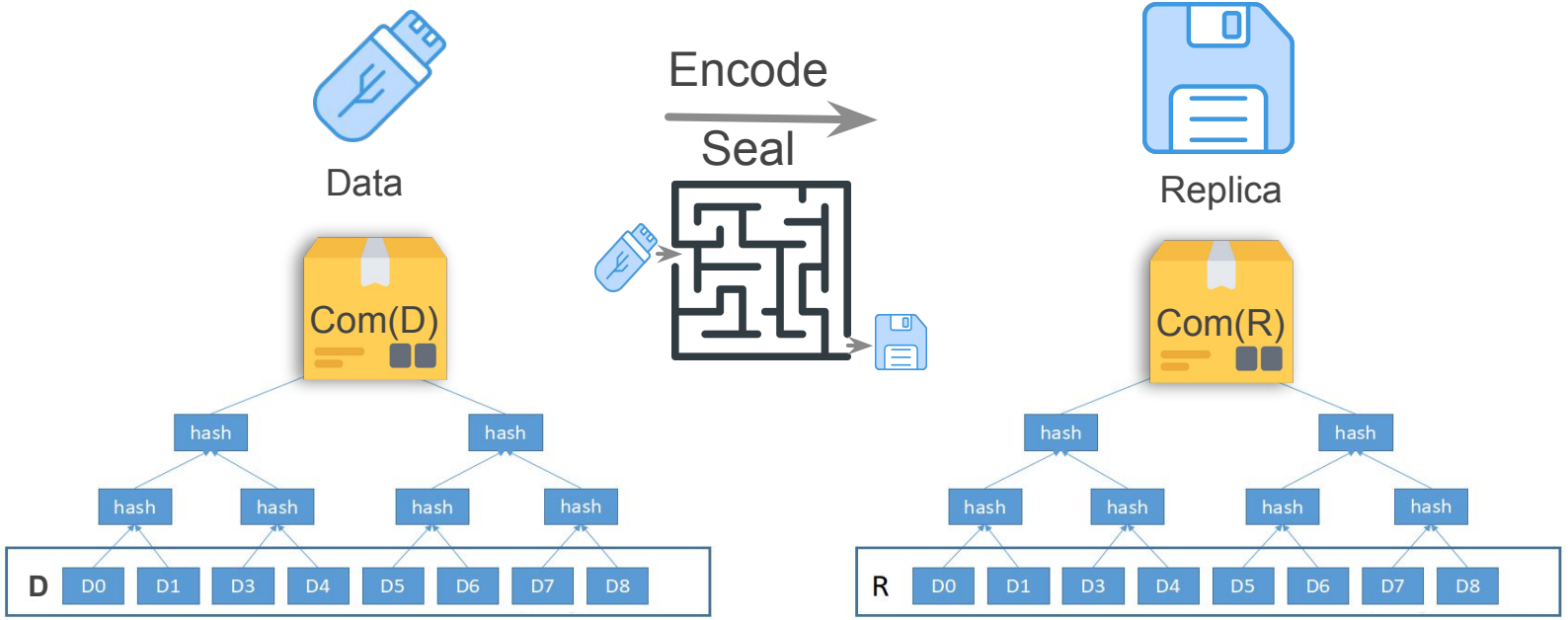
Open positions

submit

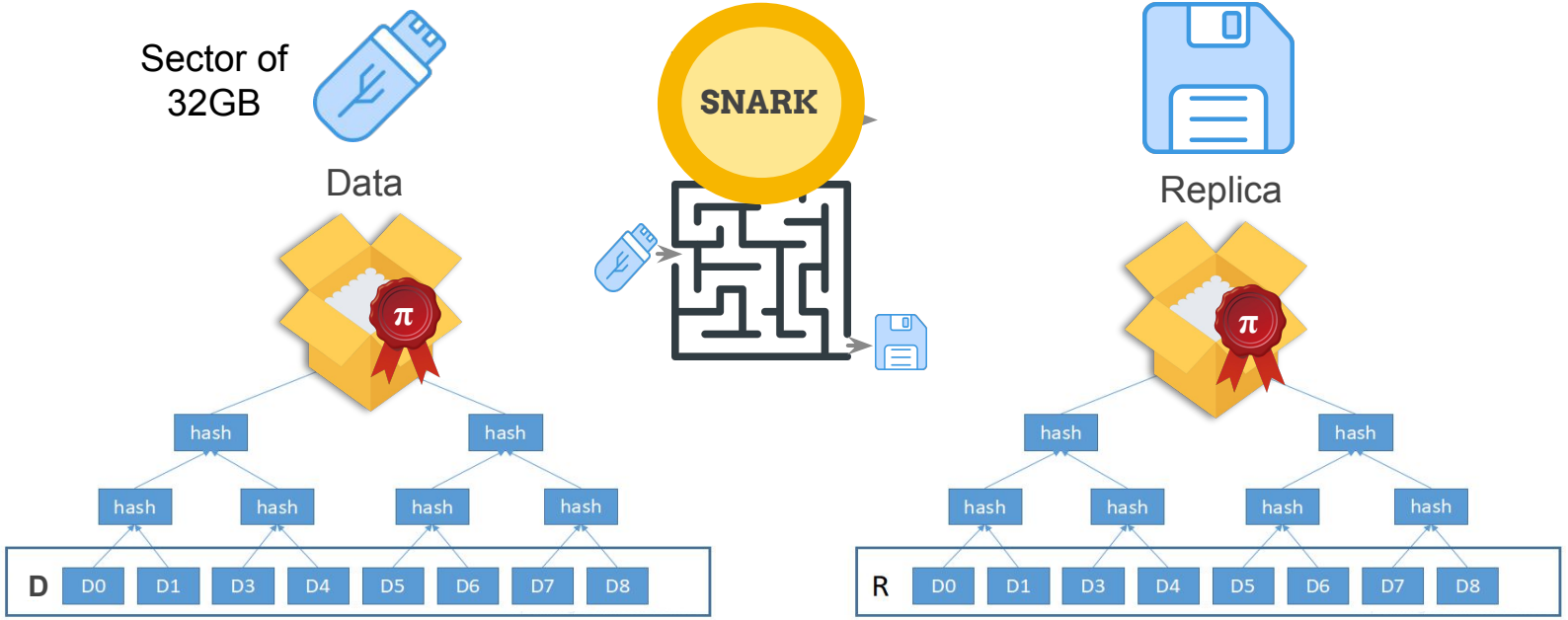
Tool: Merkle Tree



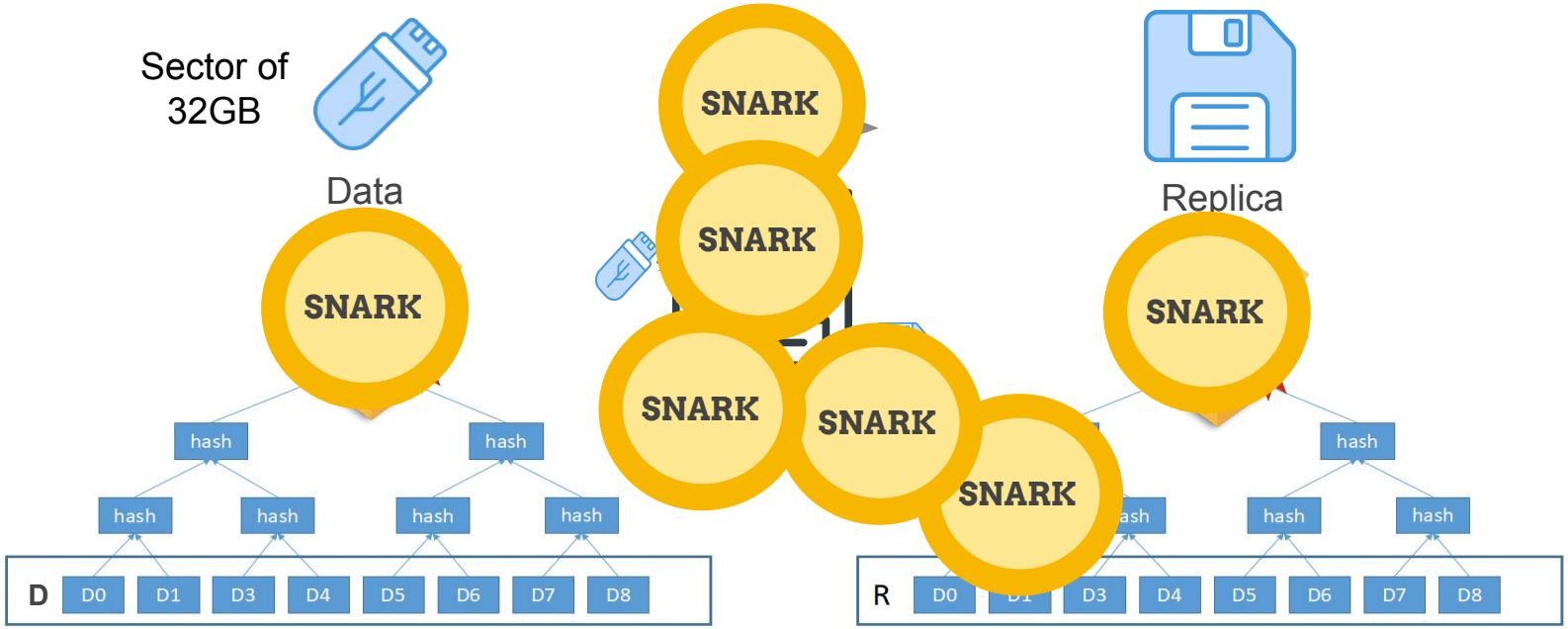
Proof of Replication



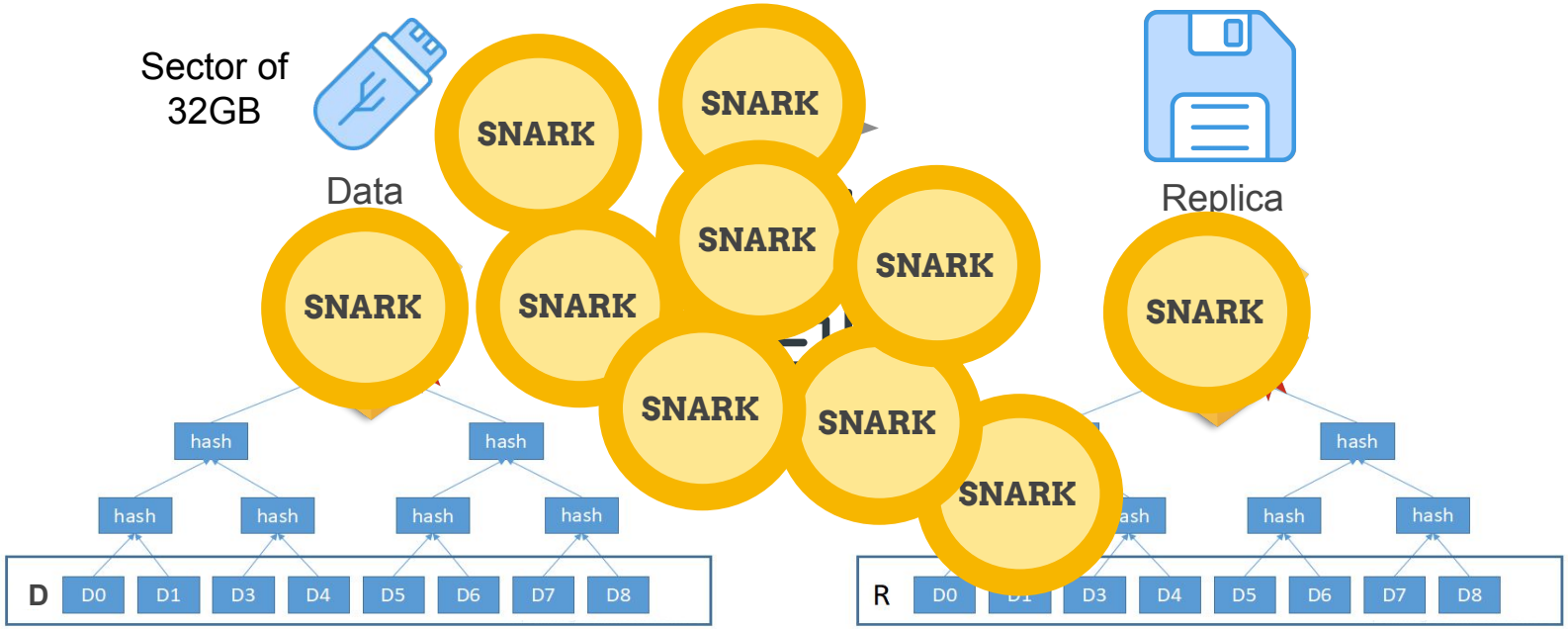
Proof of Replication



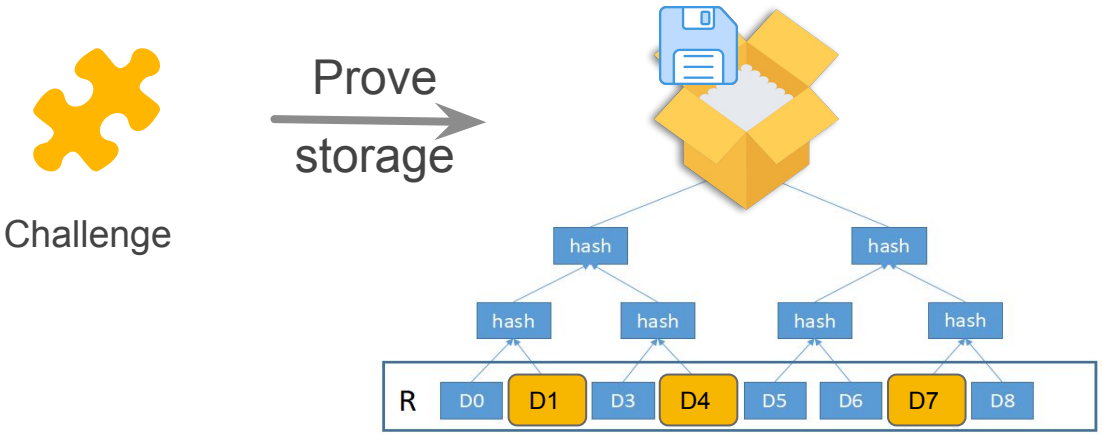
Proof of Replication



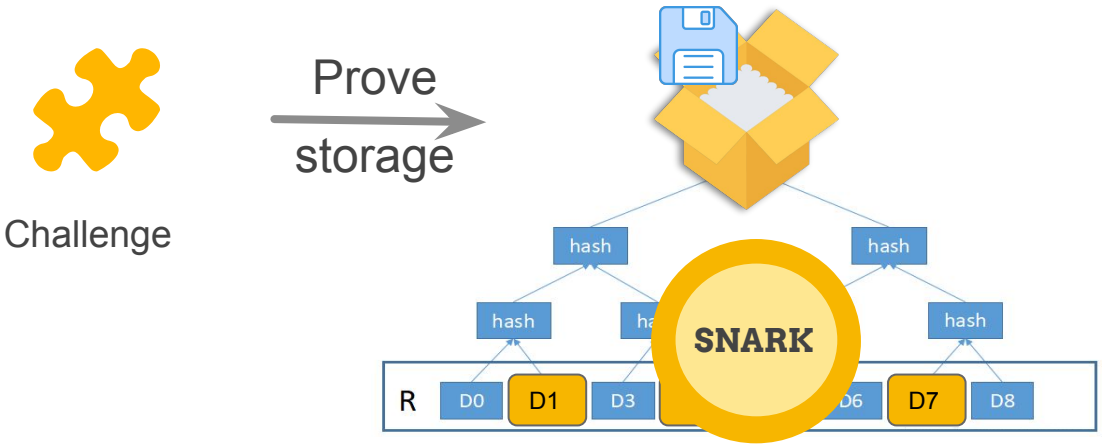
Proof of Replication



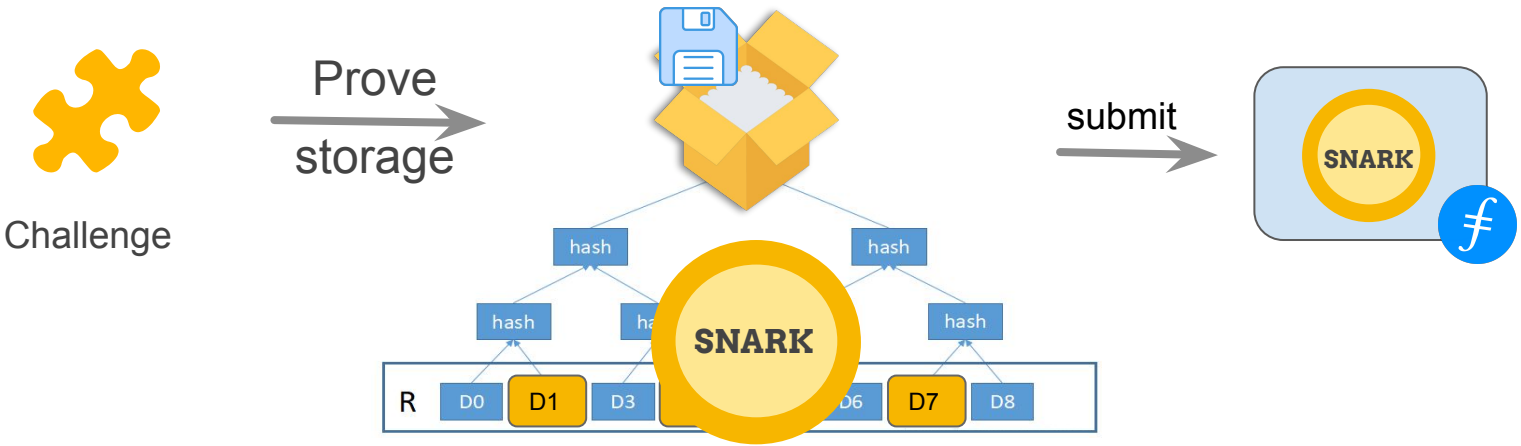
Proof of SpaceTime



Proof of SpaceTime



Proof of SpaceTime

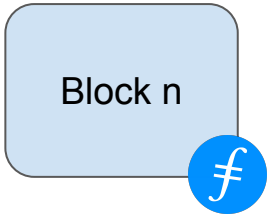


Future Directions

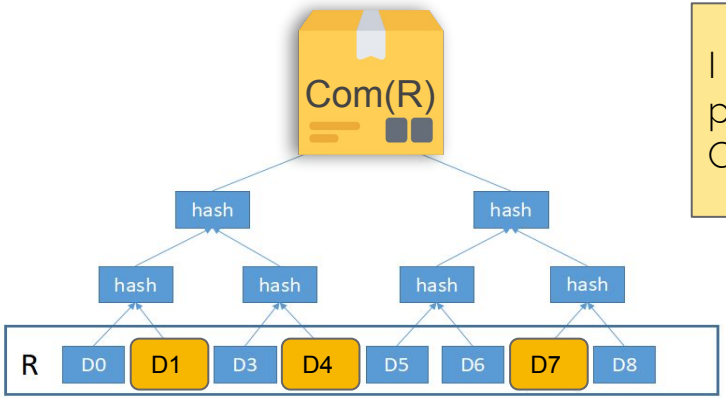
Better VC: Proof of storage

Commit 32GB

submit Proof

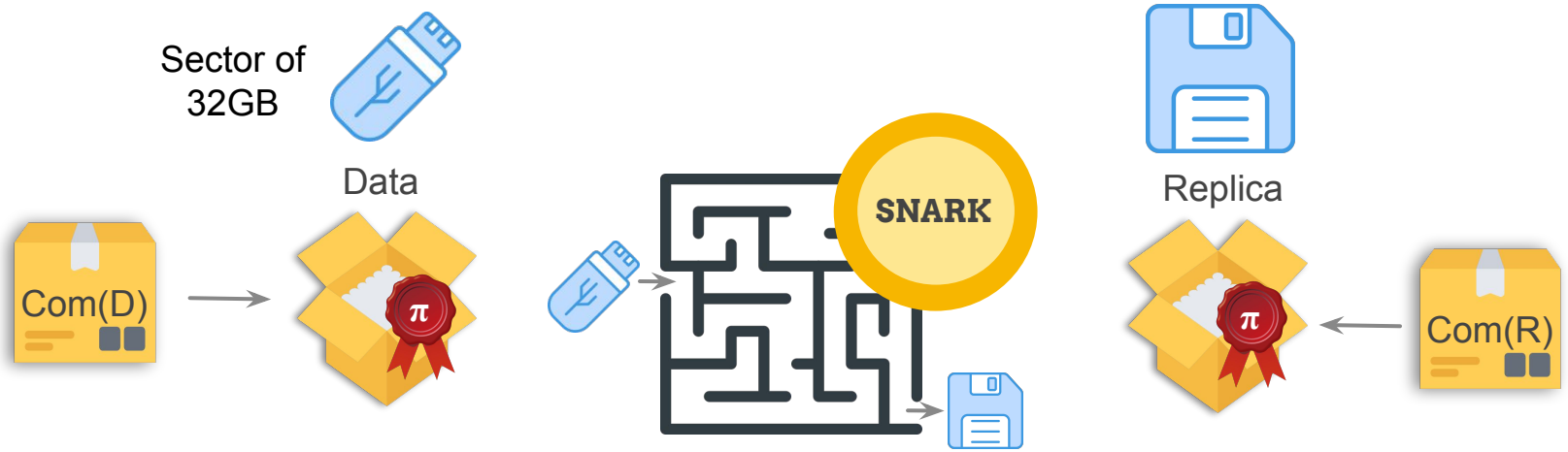


I know "enough" positions in Com(R)



Can we find better vector commitments that allow to prove knowledge of a subvector more efficiently (less prover overhead -- smaller circuit)

Better VC: Replication



What's next?

- Reusable/Transparent Setup Vector Commitments
- Functional Vector Commitments
- Post-Quantum Vector Commitments
- Aggregation for Subvector Commitments
- Argument of Knowledge of Openings
- Structure-Preserving Vector Commitments
- Trade-offs for storage/computation for Provers

Stateless Cryptocurrency

Distributed Ledger

New Block

A → B, **5 Coins**
C → D, **10 Coins**
A → C, **5 Coins**

Distributed Ledger

New Block

A → B, **5 Coins**
C → D, **10 Coins**
A → C, **5 Coins**

previous state

A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

Distributed Ledger

New Block

A → B, **5 Coins**
C → D, **10 Coins**
A → C, **5 Coins**

previous state

Validate Block

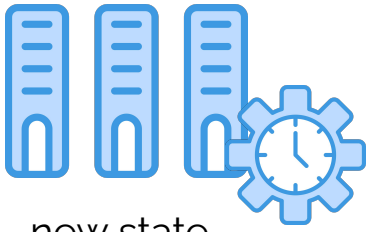
A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

Distributed Ledger

New Block

A → B, **5 Coins**
C → D, **10 Coins**
A → C, **5 Coins**

previous state



new state

A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

A: 10 Coins
B: 60 Coins
C: 25 Coins
D: 10 Coins
E: 90 Coins
F: 10 Coins
...

Grows over time

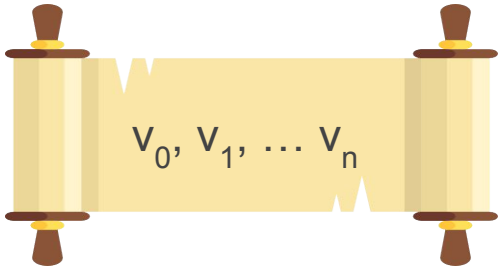
High storage & download cost

Distributed Ledger

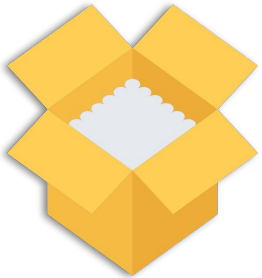
Grows over time

High storage & download cost

Vector Commitments



Commit

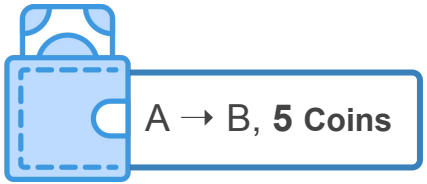


Open

Vector Commitments and their Applications, by Dario Catalano and Dario Fiore

Transaction validation

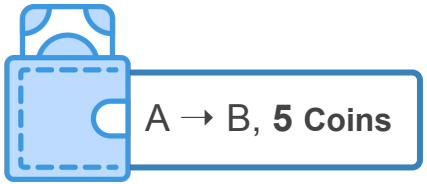
current state



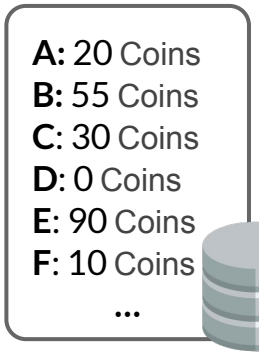
A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

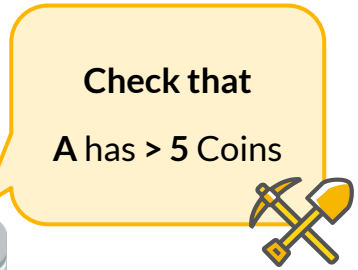
Transaction validation

current state

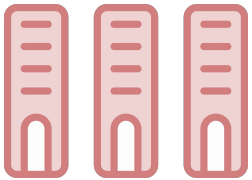


Validate Transaction

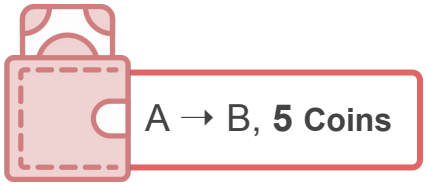




Stateful validation



current state



Validate Transaction

A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

Stateless validation

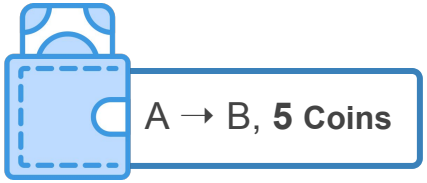
State Commitment
(vector of balances)

A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

Stateless validation

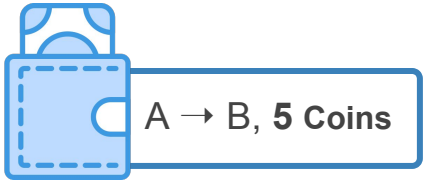
State Commitment
(vector of balances)

?



A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

Stateless validation



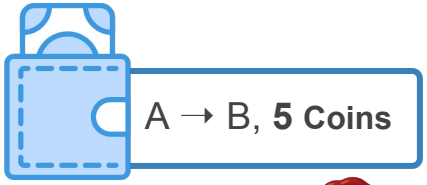
State Commitment
(vector of balances)

A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

π_A
π_B
π_C
...

Stateless validation

State Commitment
(vector of balances)



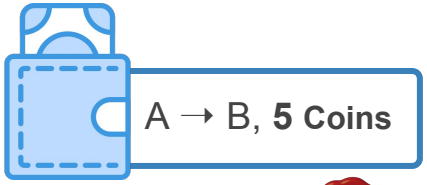
A: 20

A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

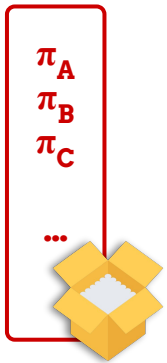
π_A
π_B
π_C
...

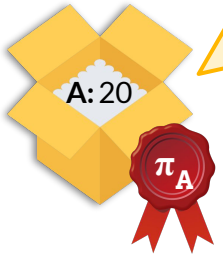
Stateless validation

State Commitment
(vector of balances)



A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...



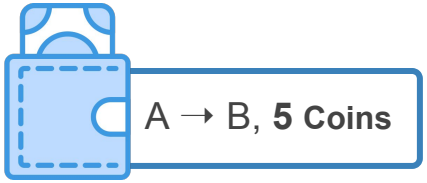


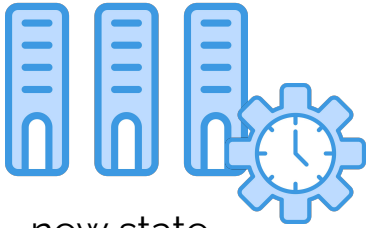
Check that

π_A is a valid opening for **Com**

More Requirements

State Commitment
(vector of balances)

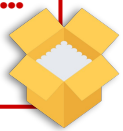




new state

A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

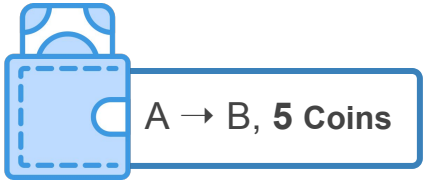
π_A
π_B
π_C
...



A: 15 Coins
B: 60 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

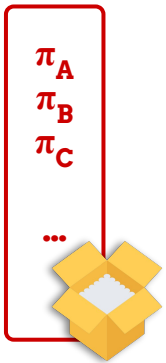
Updatability for Com

State Commitment
(vector of balances)



Update

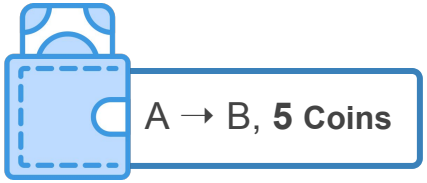
A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...



A: 15 Coins
B: 60 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

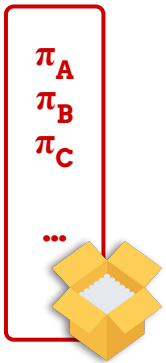
Updatability for Proofs

State Commitment
(vector of balances)

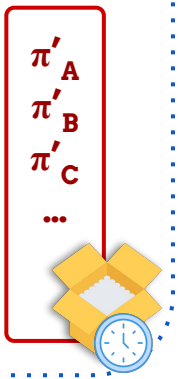


Update

A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

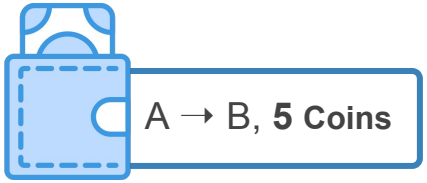


A: 15 Coins
B: 60 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...



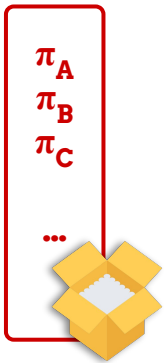
Updatability for Proofs

State Commitment
(vector of balances)



Update

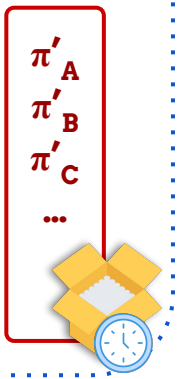
A: 20 Coins
B: 55 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...



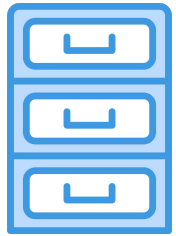
Updates types

- **hint**: needs opening
- **key**: needs static key
- **key-less**: no key

A: 15 Coins
B: 60 Coins
C: 30 Coins
D: 0 Coins
E: 90 Coins
F: 10 Coins
...

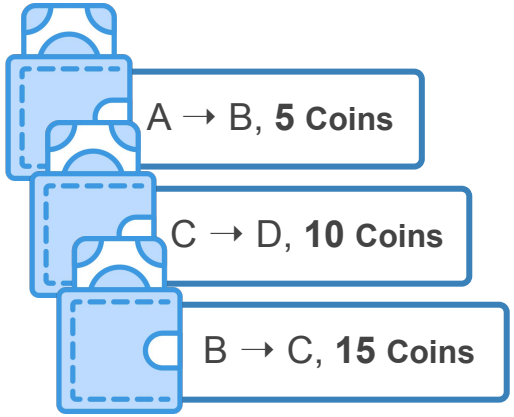


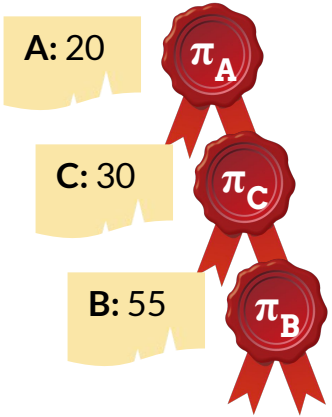
Aggregation



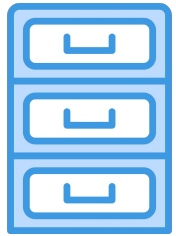
New Block

A → B, **5 Coins**
C → D, **10 Coins**
B → C, **15 Coins**



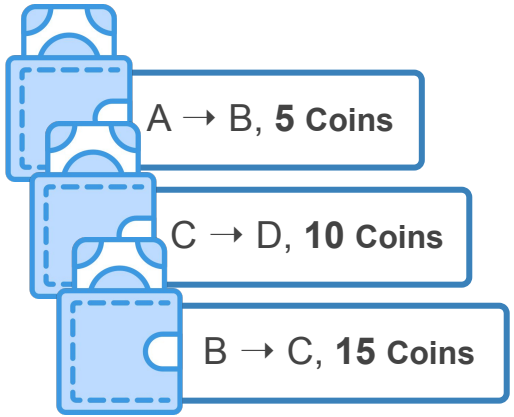


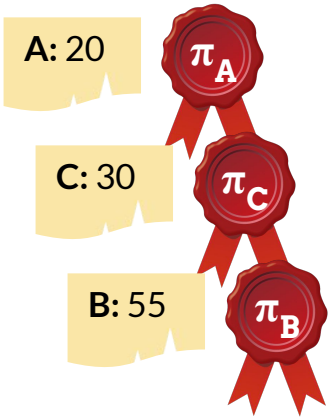
Aggregation



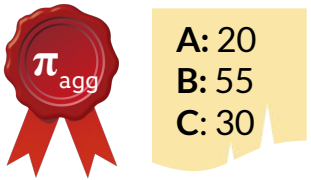
New Block

A → B, **5 Coins**
C → D, **10 Coins**
B → C, **15 Coins**

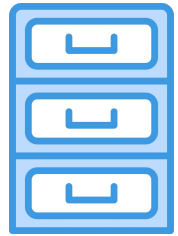




Aggregate

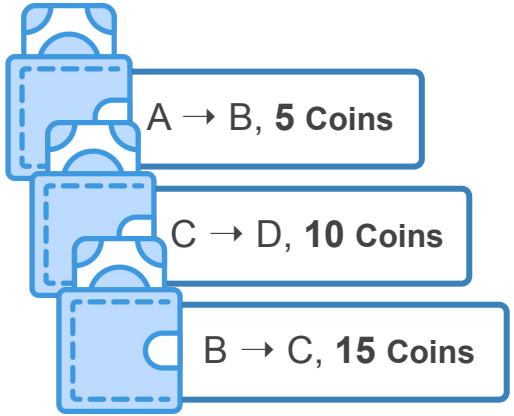


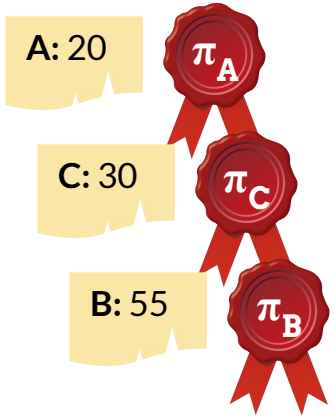
Aggregation



New Block

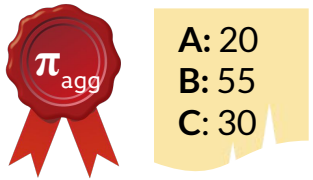
A → B, **5 Coins**
C → D, **10 Coins**
B → C, **15 Coins**





- same-commitment/**cross-commitment**
- one-hop/**unbounded**
- **incremental**: can disaggregate

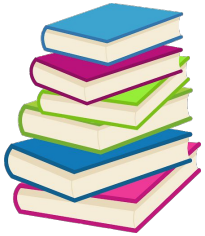
Aggregate



VC comparison

Schemes		|ck|	|vk|	|upk|	Updates	Aggregation	SVC	LVC $f : \mathbb{F}^n \to \mathbb{F}^\ell$
[TAB+20]		n	ℓ	n	key	same & one-hop	✓	✗
[LM19]	SVC	n^2	n	n	key	✗	✓	✗
	LVC	$n\ell$	n	n	key	✗	✓	✓
Pointproofs		n	n	n	key	cross & one-hop	✗	✗
[CFG+20]		1	1	1	hint	same & incremental	✓	✗
[CNRZ22]	Monomial	n	n	—	key-less	cross & unbounded	✓	✓
	Lagrange	n	ℓ	n	key	cross & unbounded	✓	✓

References



[BBF19] **Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains**, in CRYPTO'19, D. Boneh, B. Bünz, B. Fisch, https://eprint.iacr.org/2018/1188

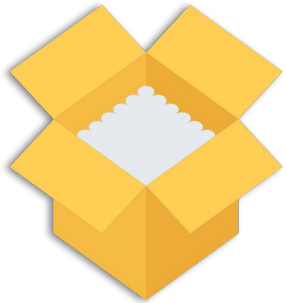
[CF13] **Vector Commitments and their Applications**, by D. Catalano, D. Fiore, in PKC'13, https://eprint.iacr.org/2011/495

[CFG+20] **Vector Commitment Techniques and Applications to Verifiable Decentralized Storage**, by M. Campanelli, D. Fiore, N. Greco, D. Kolonelos, L. Nizzardo, in ASIACRYPT'20, https://eprint.iacr.org/2020/149

[GRWZ20] **Pointproofs: Aggregating Proofs for Multiple Vector Commitments**, by S. Gorbunov, L. Reyzin, H. Wee, Z. Zhang, in CCS'20, https://eprint.iacr.org/2020/419

[LM19] **Subvector Commitments with Application to Succinct Arguments**, by R. W. F. Lai, G. Malavolta, in CRYPTO'19, https://eprint.iacr.org/2018/705

[TAB+20] **Aggregatable Subvector Commitments for Stateless Cryptocurrencies**, by Tomescu, A.; Abraham, I. Buterin, V. Drake, J. Feist, D. Khovratovich, in SCN'20



Thank You

Credits

Special thanks to all those who made and released these resources for free:

- Presentation template by SlidesCarnival
- Illustrations by Iconfinder