Zero-Knowledge proof of knowledge transfer

Perm summer school on blockchain 2018

Teleport

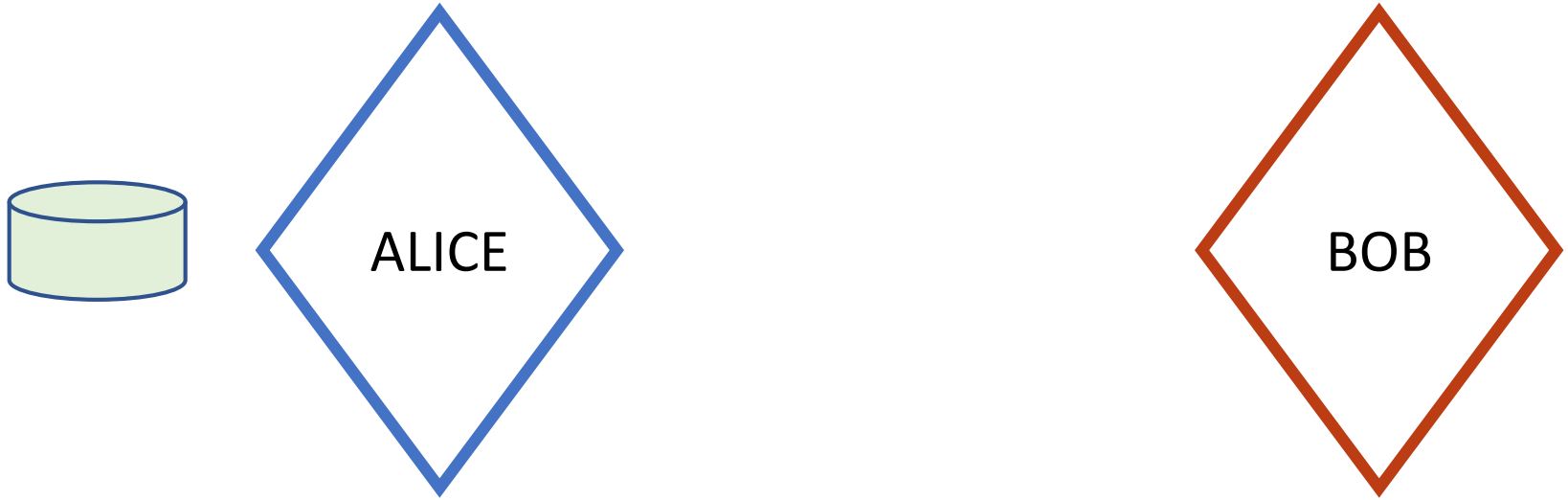
Teleport was born in 2016 from the idea to bring the power of peer-to-peer traffic distribution technology like BitTorrent to the solution of traffic bottleneck problem that each video streaming service and each of their customers familiar with.

Teleport in 2018 is the running business with the top-notch online video streaming platforms as the paying customers, and with the solution that is capable of building p2p network in real-time supporting "heavy" video traffic delivery to millions of simultaneous viewers with the unprecedented quality.

Teleport by 2023 is the “Filecoin of CDN”, the decentralized analogue of AWS or Akamai. Decentralized CDN software runs on top of millions of devices such as PC, smartphone, internet router, smart TV, and many more, letting their owners lease the internet bandwidth that is unused by the device owner and earn cryptocurrency for that.

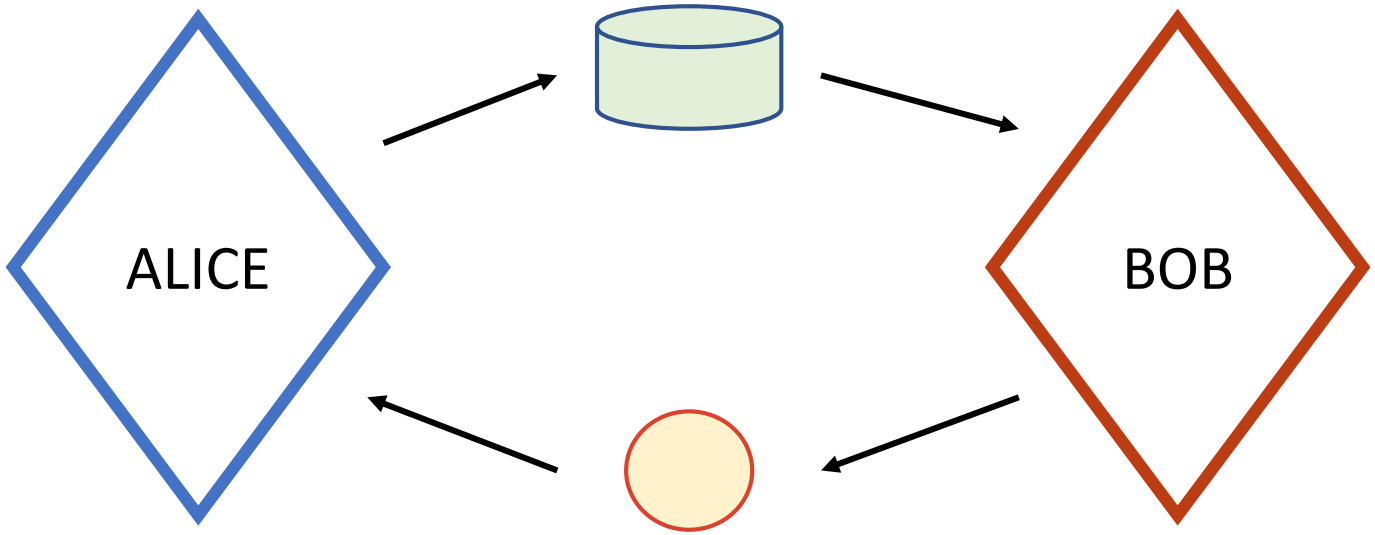
Traffic delivery set-up

Traffic delivery set-up



Traffic delivery set-up

Traffic delivery set-up



Traffic delivery set-up

Traffic delivery set-up

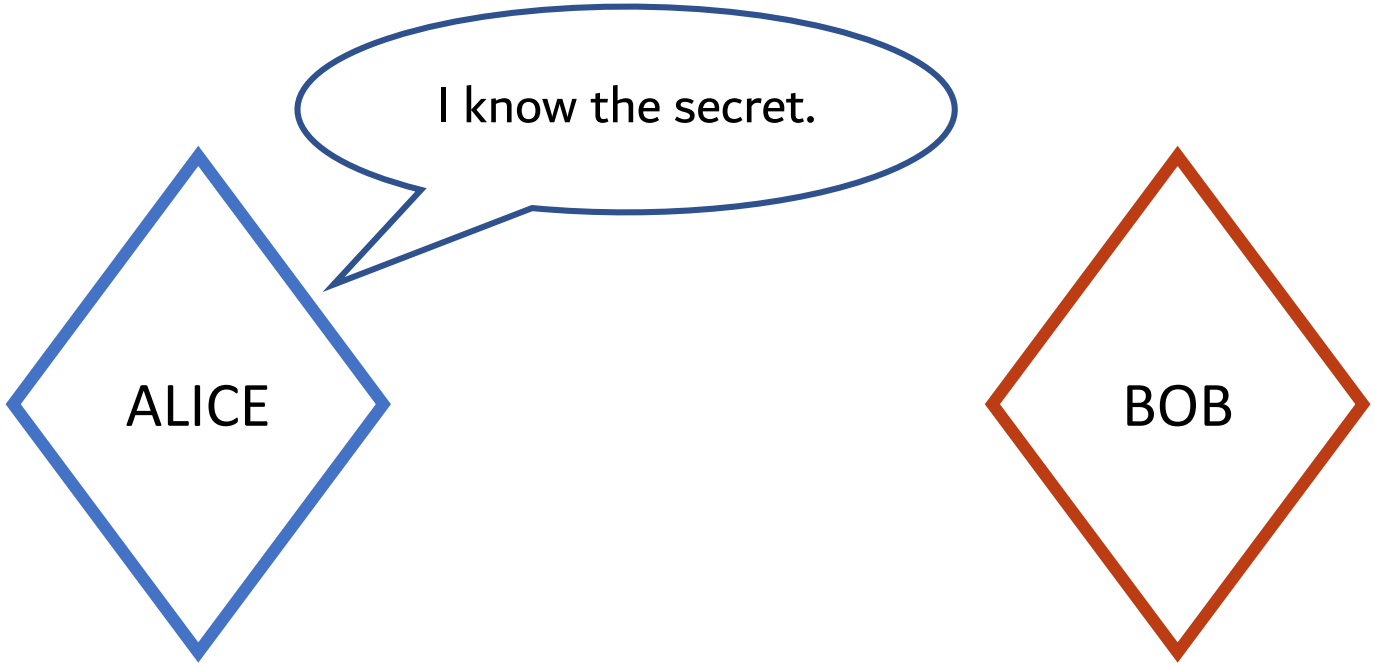
Double spending of coin?

“Double spending” of file?

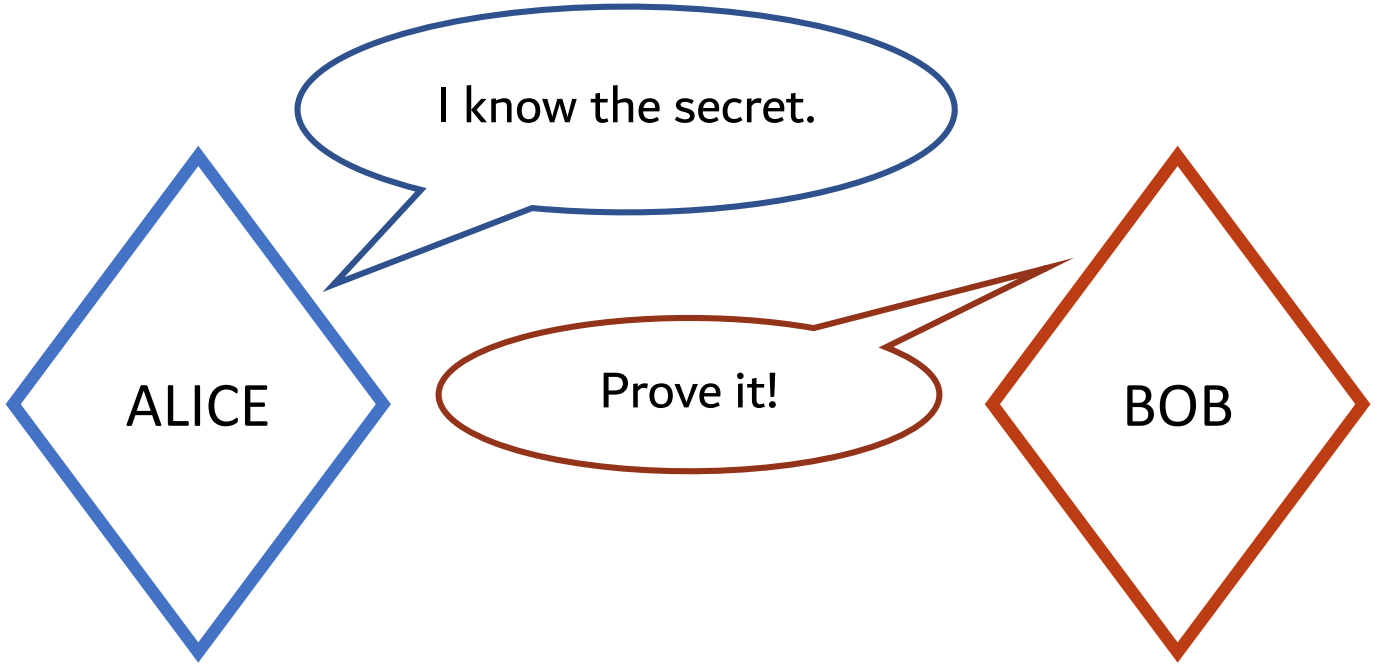
~~Double spending of coin~~

“Double spending” of file!

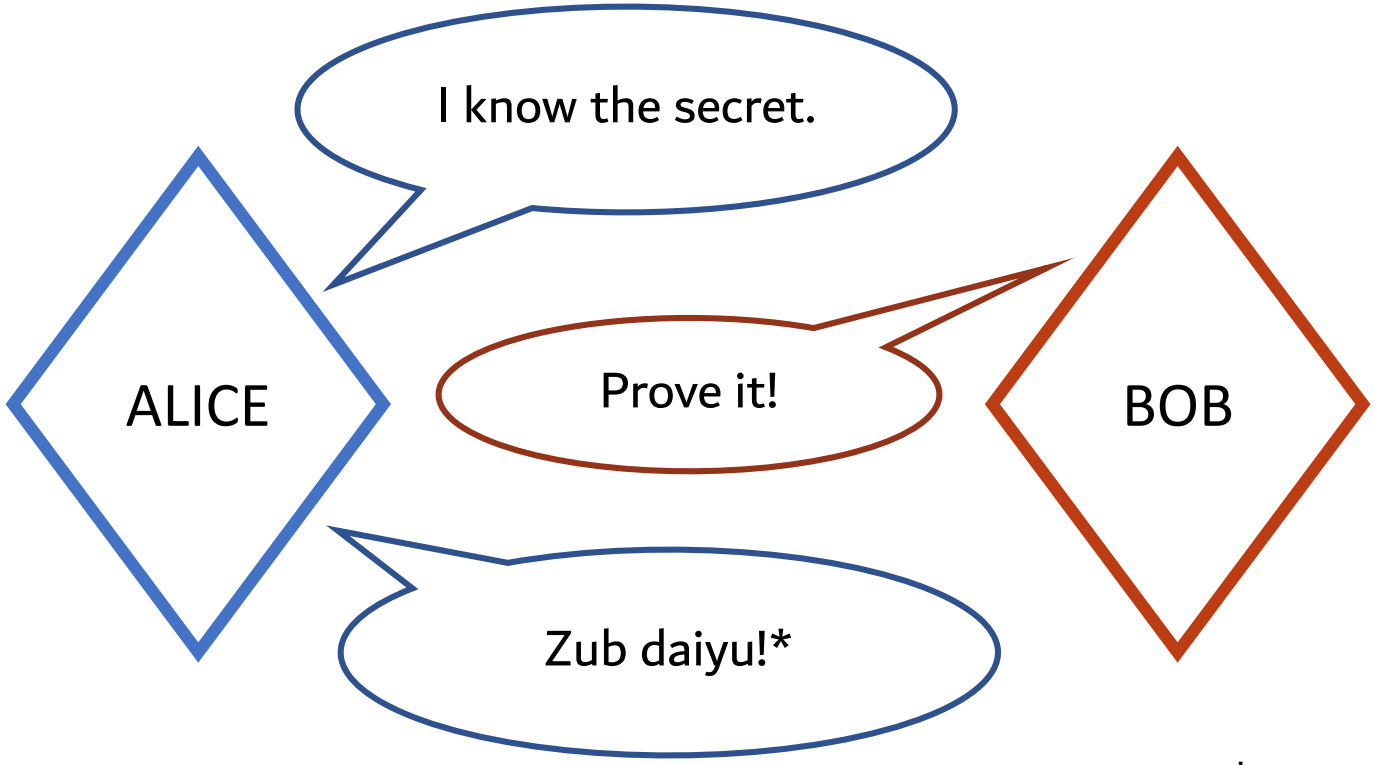
Zero-Knowledge



Zero-Knowledge



Zero-Knowledge



* the promise of terrifying body self-injury, eternal physical and moral suffering in case of malicious behaviour

Zero-Knowledge

Zero-knowledge proof – is the statement that shall prove that Prover knows something without disclosure anything about the knowledge, except the fact that Prover actually know it.

Zero-Knowledge

Zero-knowledge proof – is the statement that shall prove that Prover knows something without disclosure anything about the knowledge, except the fact that Prover actually know it.

Key requirements:

Completeness – is the assurance that if the Prover truly know something, then the Verifier must become sure of that fact from the proof.

Zero-Knowledge

Zero-knowledge proof – is the statement that shall prove that Prover knows something without disclosure anything about the knowledge, except the fact that Prover actually know it.

Key requirements:

Completeness – is the assurance that if the Prover truly know something, then the Verifier must become sure of that fact from the proof.

Soundness – is the assurance that the knowledge corresponds to certain proof is unable to forge

Zero-Knowledge

Zero-knowledge proof – is the statement that shall prove that Prover knows something without disclosure anything about the knowledge, except the fact that Prover actually know it.

Key requirements:

Completeness – is the assurance that if the Prover truly know something, then the Verifier must become sure of that fact from the proof.

Soundness – is the assurance that the knowledge corresponds to certain proof is unable to forge ... or at least it is deadly hard to forge

Zero-Knowledge

Cryptographic signature verification

Alice possesses private key PrK
Alice wants the whole world to know that she's agree with the certain message M
Alice process her message M through function S (M, H(M), PrK)

Cryptographic signature verification

Alice possesses private key PrK
Alice wants the whole world to know that she's agree with the certain message M
Alice process her message M through function S (M, H(M), PrK)

Anyone can learn Alice's public key PbK
Anyone can run verification function V (S , H(M), PbK)

Zero-Knowledge

Cryptographic signature verification

Alice possesses private key PrK
Alice wants the whole world to know that she agrees with the certain message M
Alice process her message M through function S (M, H(M), PrK)

Anyone can learn Alice's public key PbK
Anyone can run verification function V (S , H(M), PbK)

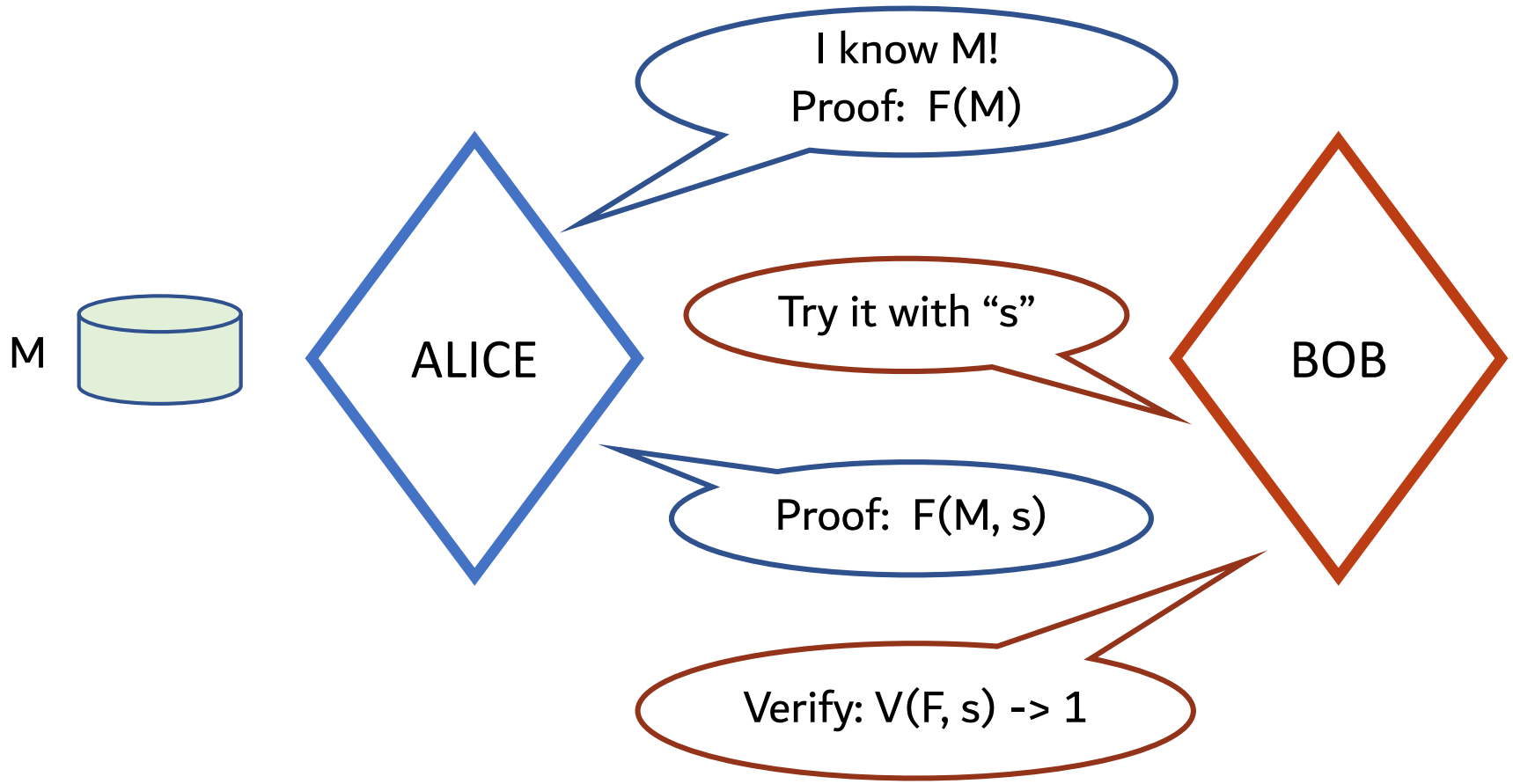
Function releases two statements:
1 – the message with the H(M) has been signed by private key which is the pair to PbK
0 – the message with the H(M) has not been signed with the private key which is the pair to PbK

“ZKP in my heart”

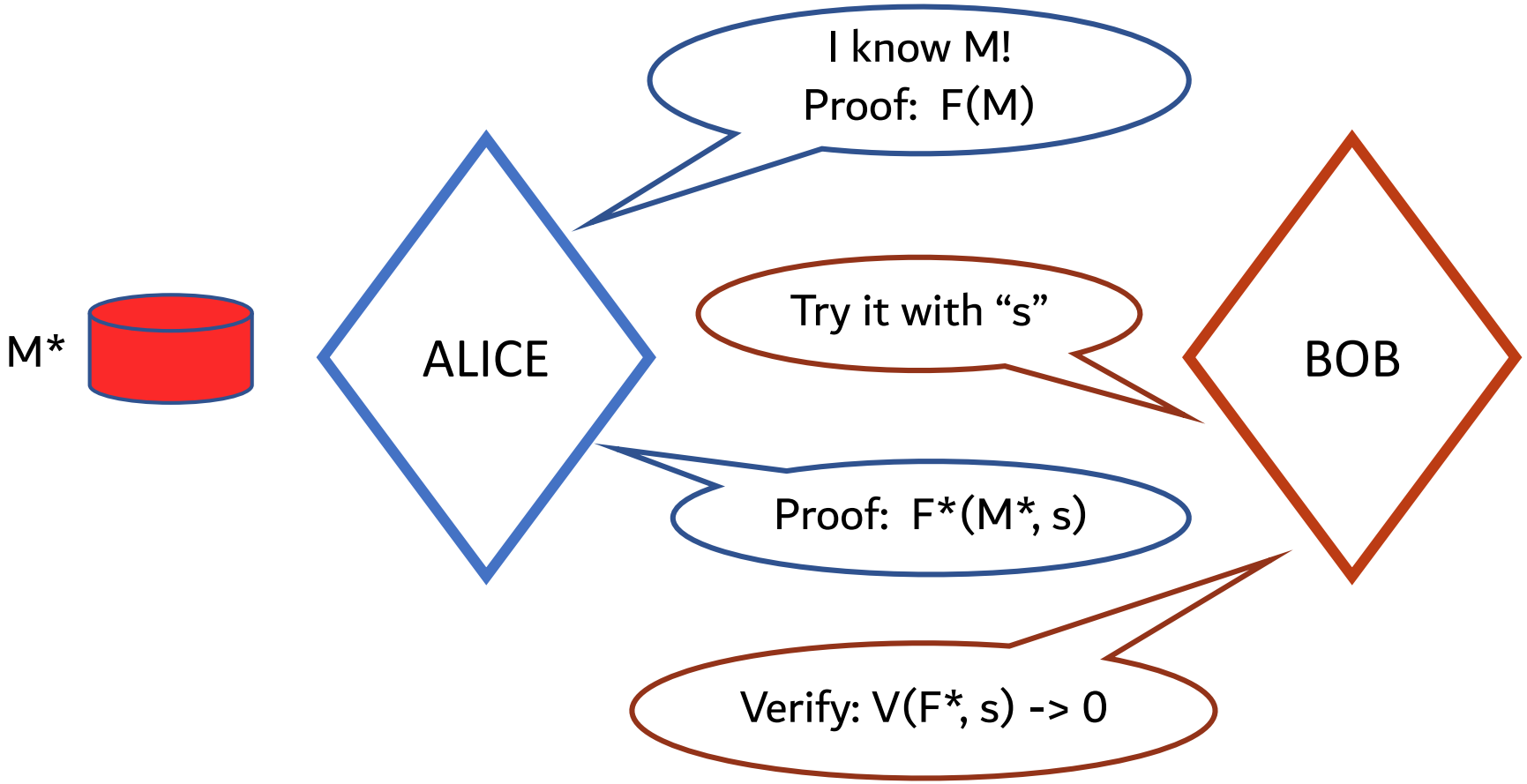
@blockchain

Perm summer school on blockchain 2018

Proof of knowledge



Proof of knowledge



Threats of Decentralized CDN

Sybil attack

Assume the networks stores multiple files. How organize files search in such network?
IPFS –hash of file is the search index

Each Alice shall produce H(M)
Each Bob will search files with H(M)

Malicious Alice may sniff all H(M) and pretend she's distributing everything. Many such Alices paralyze the network

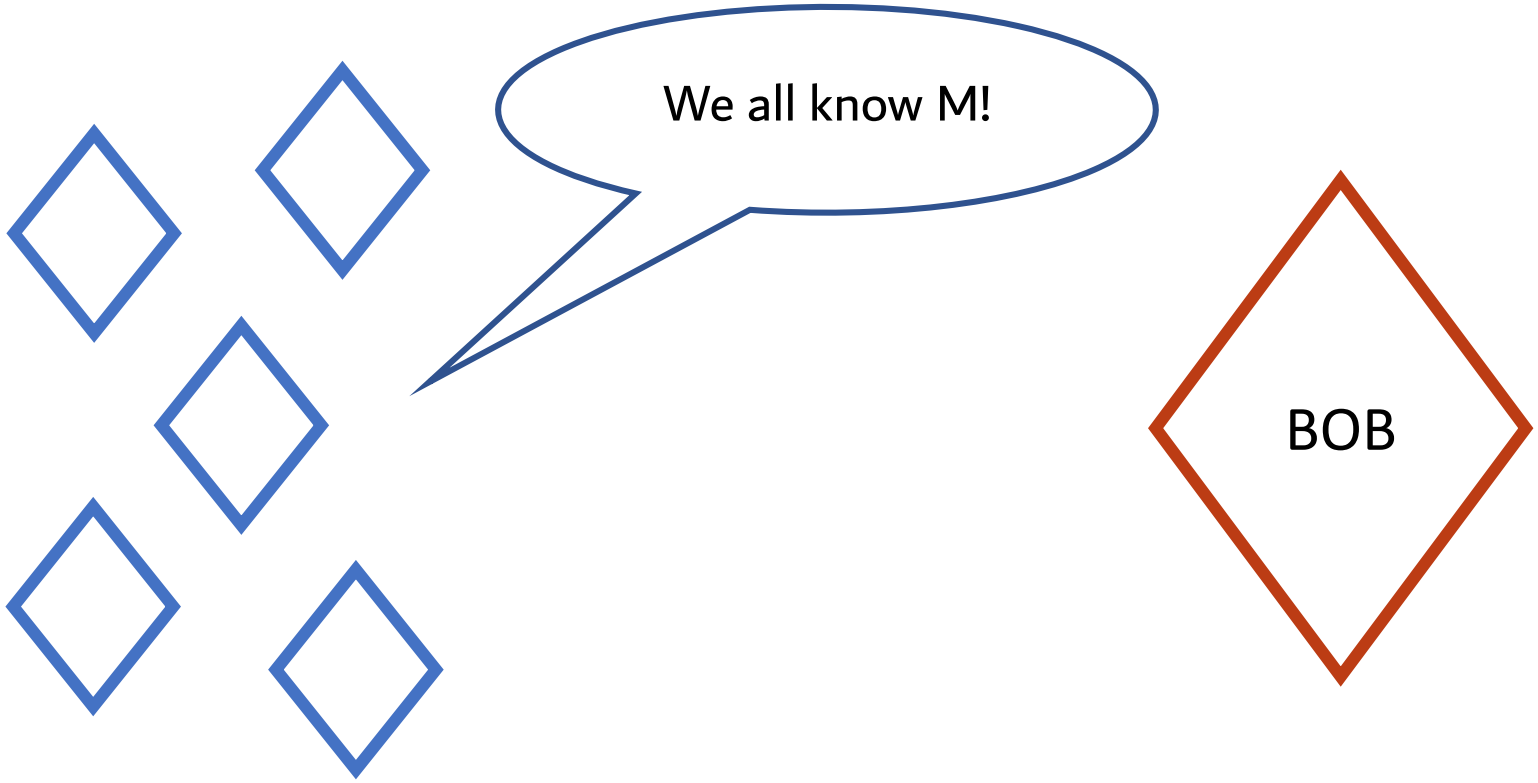
Threats of Decentralized CDN

Mitigating Sybil attack

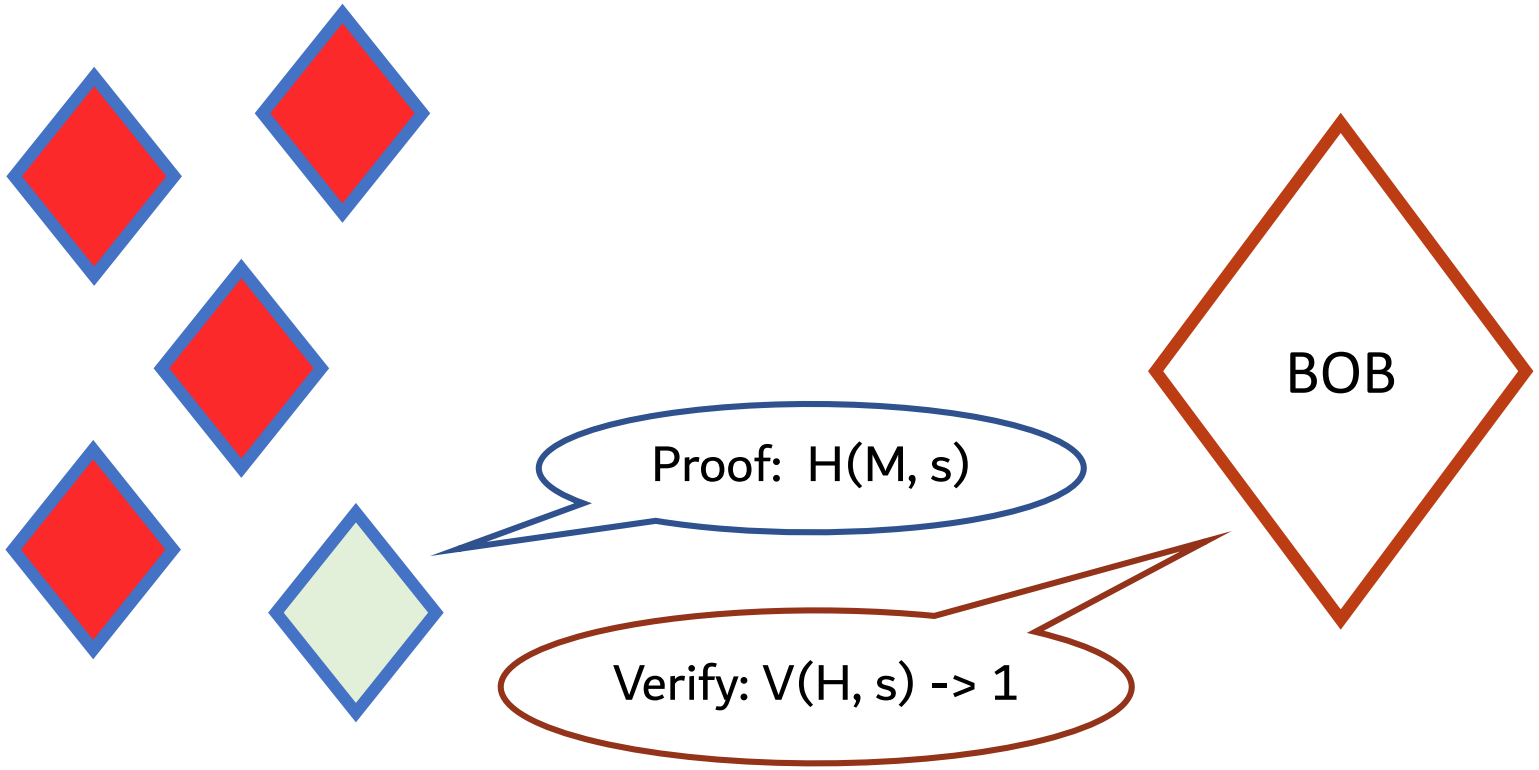
Bob shall not search by “true” hash of file, using “keys” instead.

Alice shall prove that she knows the “true” file using ZKP

Threats of Decentralized CDN



Threats of Decentralized CDN



Proof of knowledge in Teleport

Serverless conditional access system

K(M,B) = F (H(M), PbB)

1. Alice produce CryptoHave = F (K(M,B), PbB) = F (H(M), PbB, PbB) = CH

2. Bob may use all his K(M,B) to produce all possible CryptoHaves and compare it to Alice's. Thus Bob may be sure Alice may produce K of M (or at least knows K)

3. Bob produce CryptoMatch = F (CH, K, PbA)

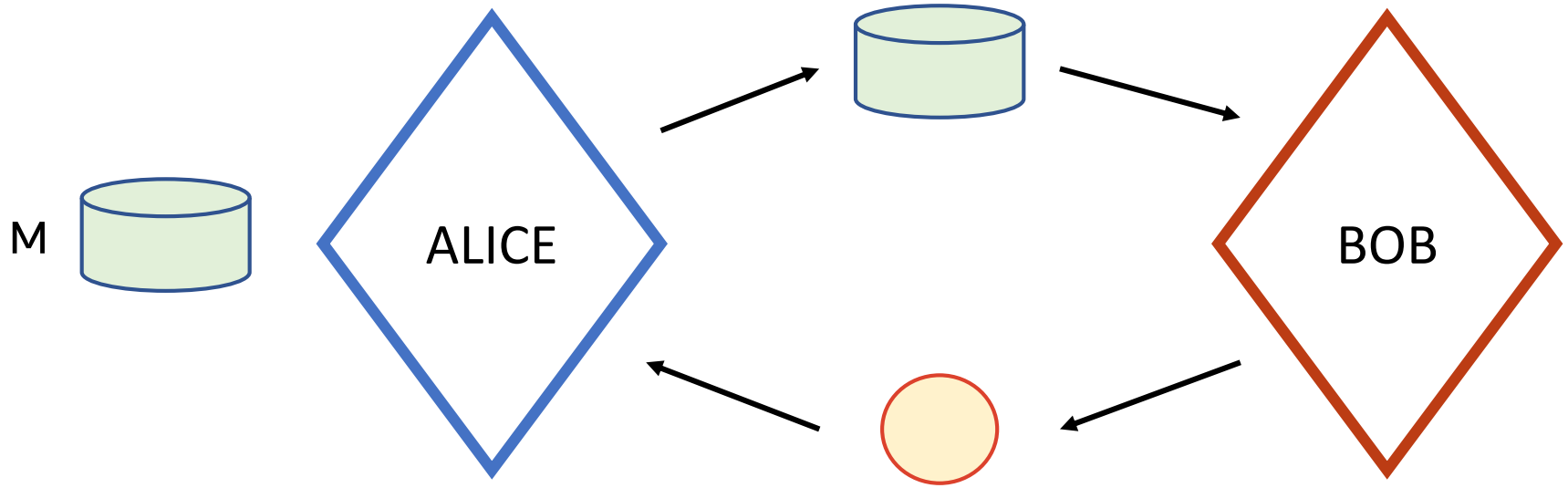
4. Alice may use all her CH to produce all possible CM and compare it to Bob's. Thus Alice may be sure Bob knows K

Proof of knowledge transfer in Teleport

Serverless conditional access system

5. Once Bob have downloaded the file he produces CryptoConfirm
CryptoConfirm = F (CM, H(M))

Threats of Decentralized CDN

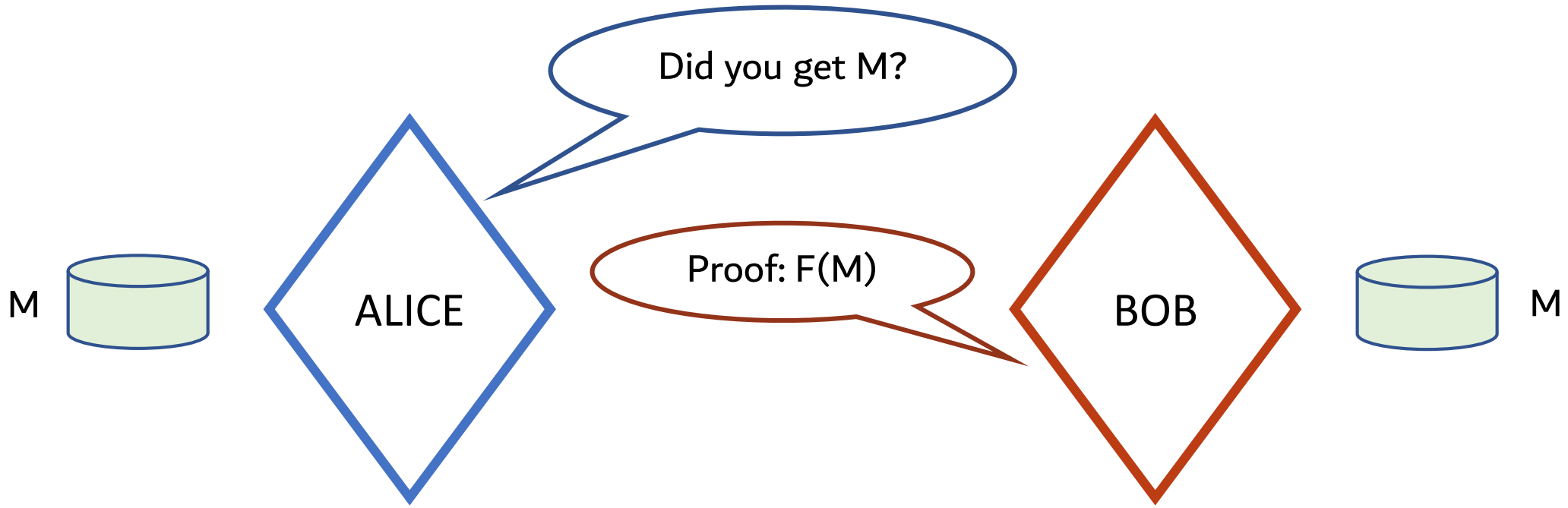


Honest Alice and Bob

Threats of Decentralized CDN

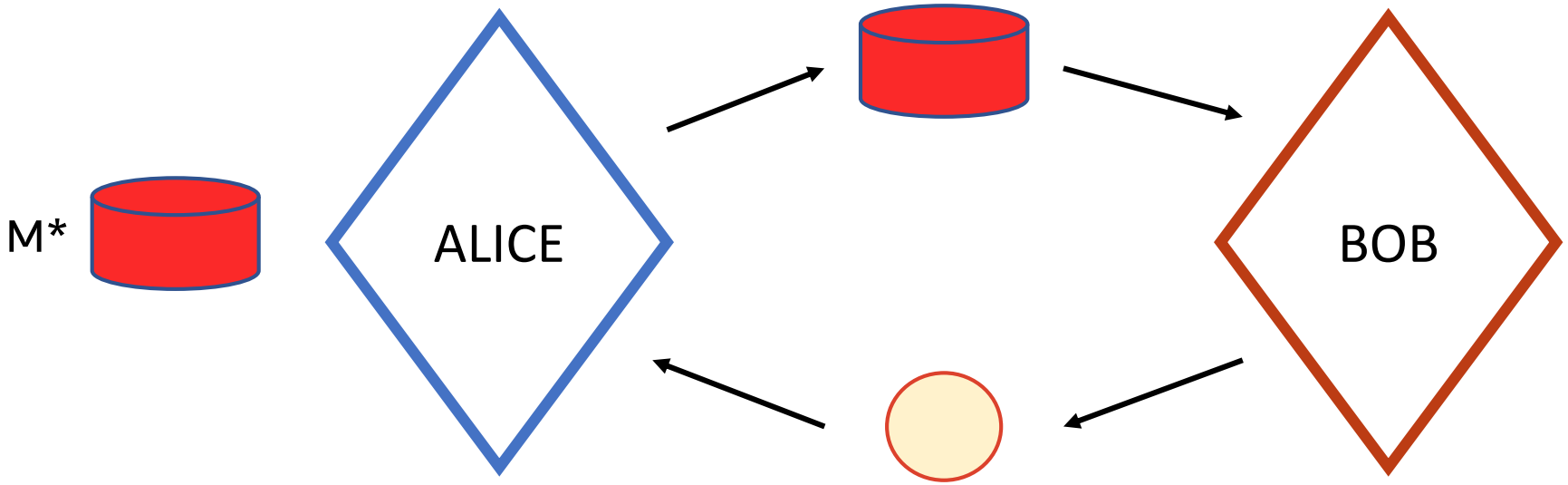
Honest Alice and Bob

Threats of Decentralized CDN



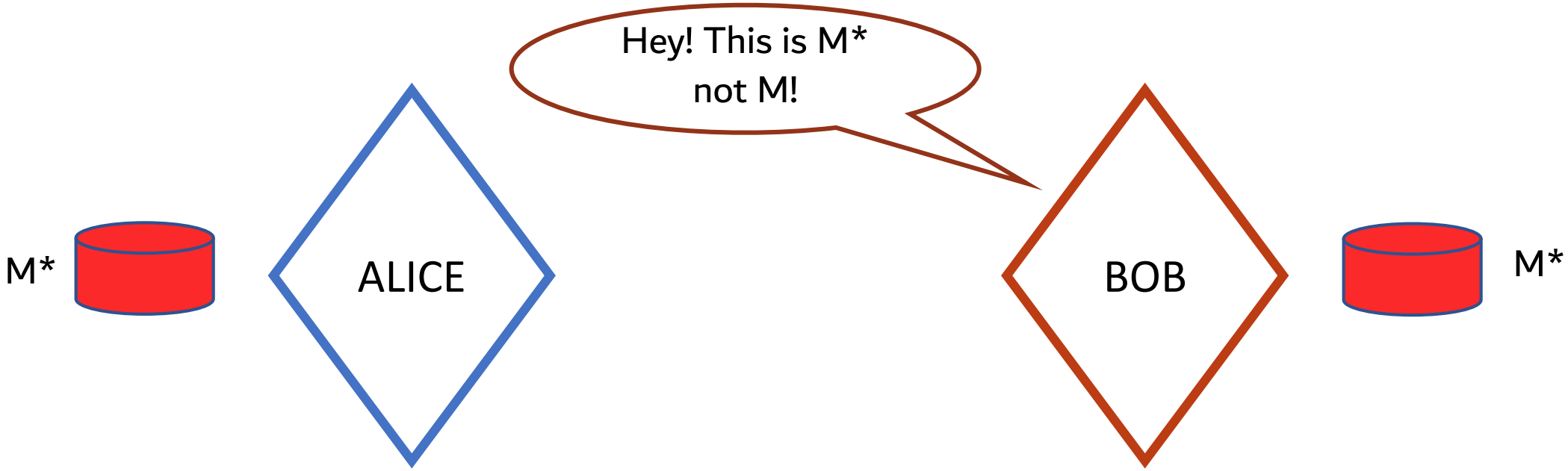
Honest Alice and Bob

Threats of Decentralized CDN



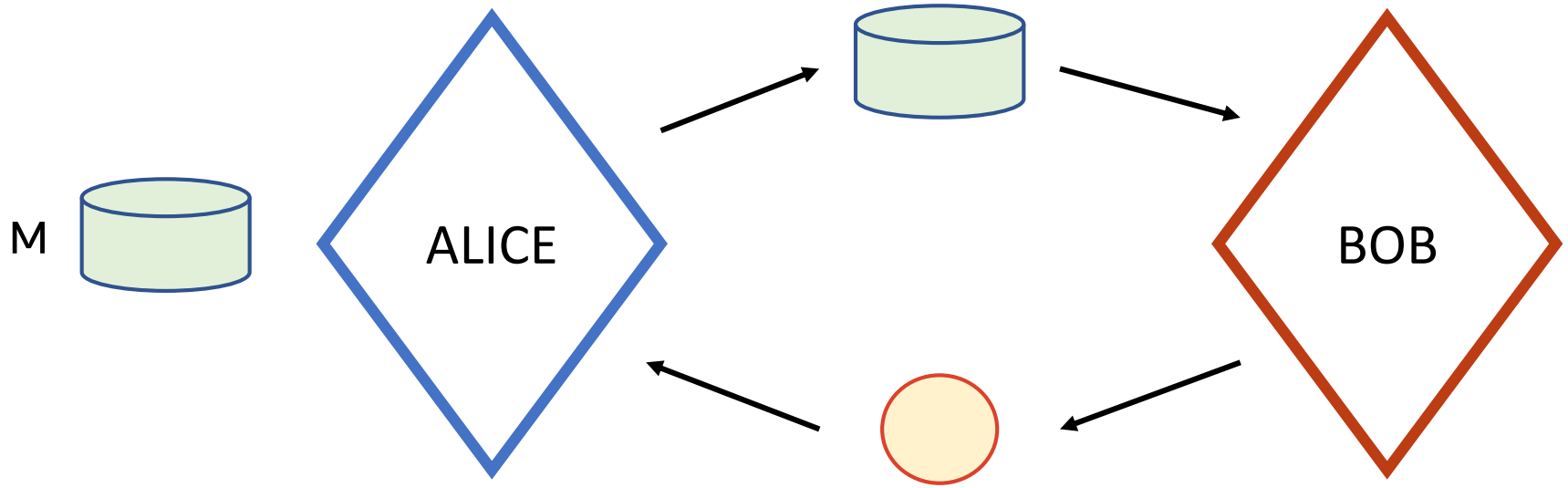
Malicious Alice / Honest Bob

Threats of Decentralized CDN



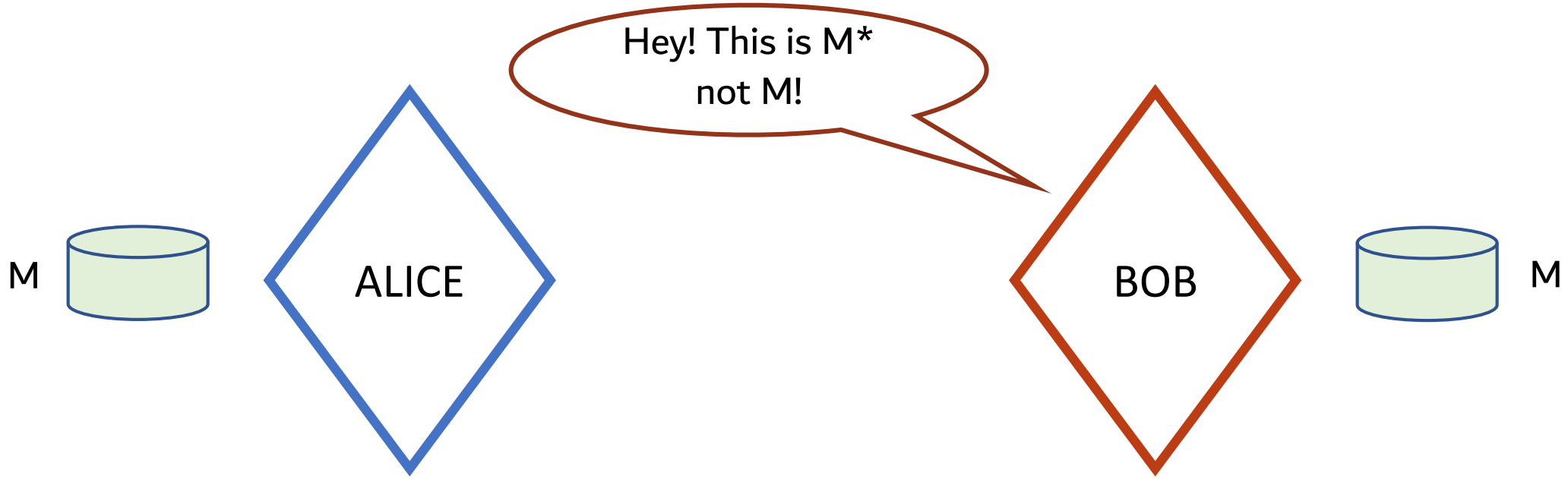
Malicious Alice / Honest Bob

Threats of Decentralized CDN



Honest Alice / Malicious Bob

Threats of Decentralized CDN



Honest Alice / Malicious Bob

Proof of knowledge transfer in Teleport

Financial solution

The price of delivery 1 Gb varies from $0.1 to $0.005.

Let’s just break file into chunks of 10Mb.

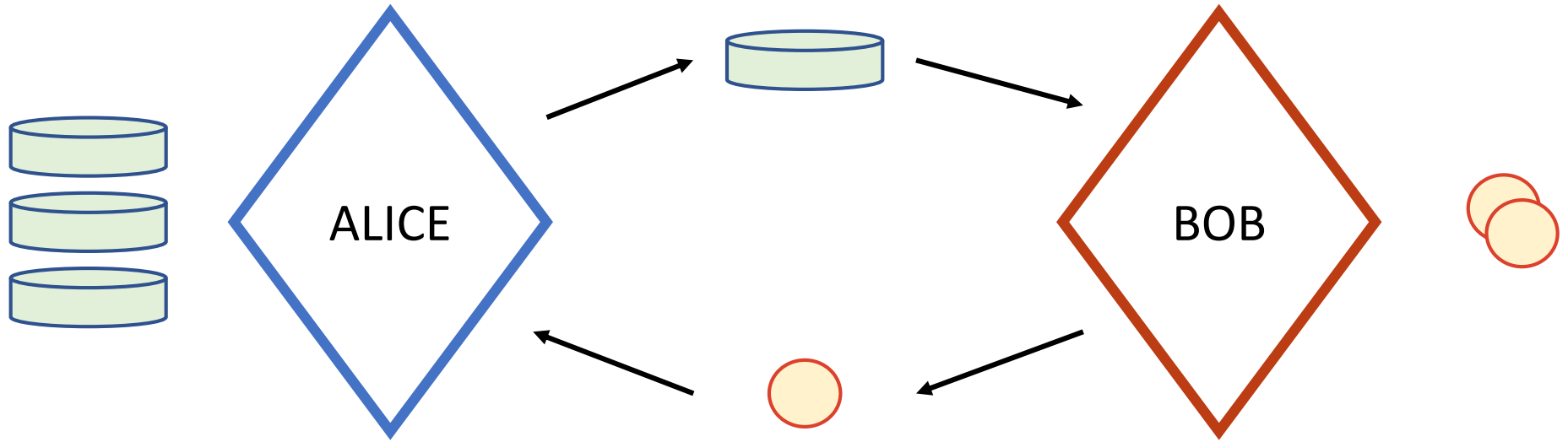
The honest Bob may just disconnect when discovers malicious behaviour. The profit of Alice from one such Bob would be just a fraction of US cent.

The honest Alice may not continue file transfer if Bob hasn’t signed the previous transaction. Then Bob would not download the complete file.

Looks more or less like “Lightning channels” in Bitcoin

Traffic-to-coin channel

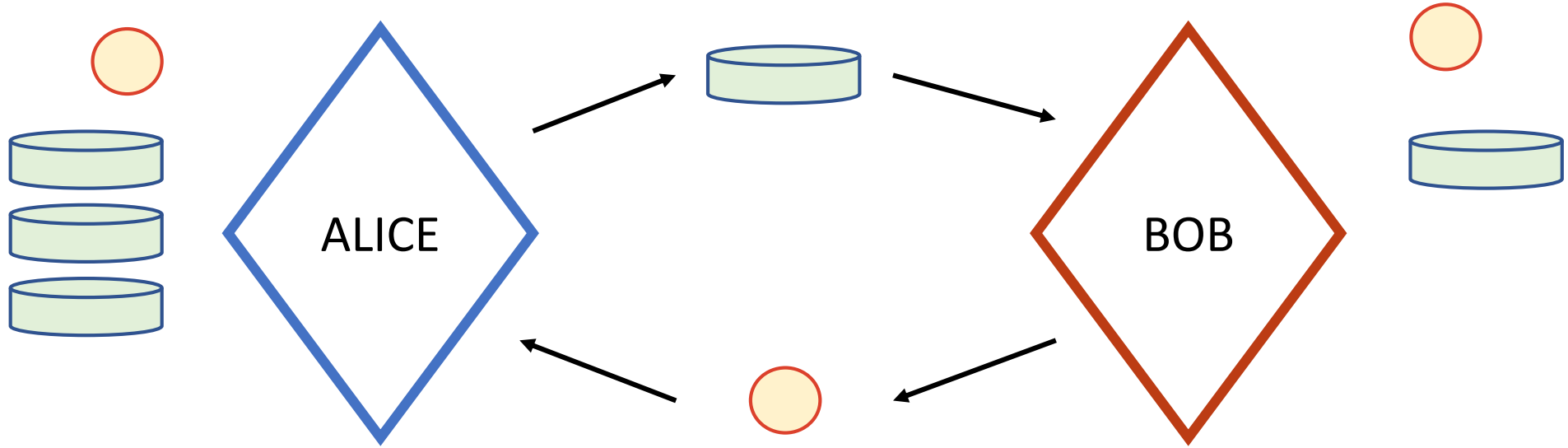
Traffic-to-coin channel



Traffic-to-coin channel

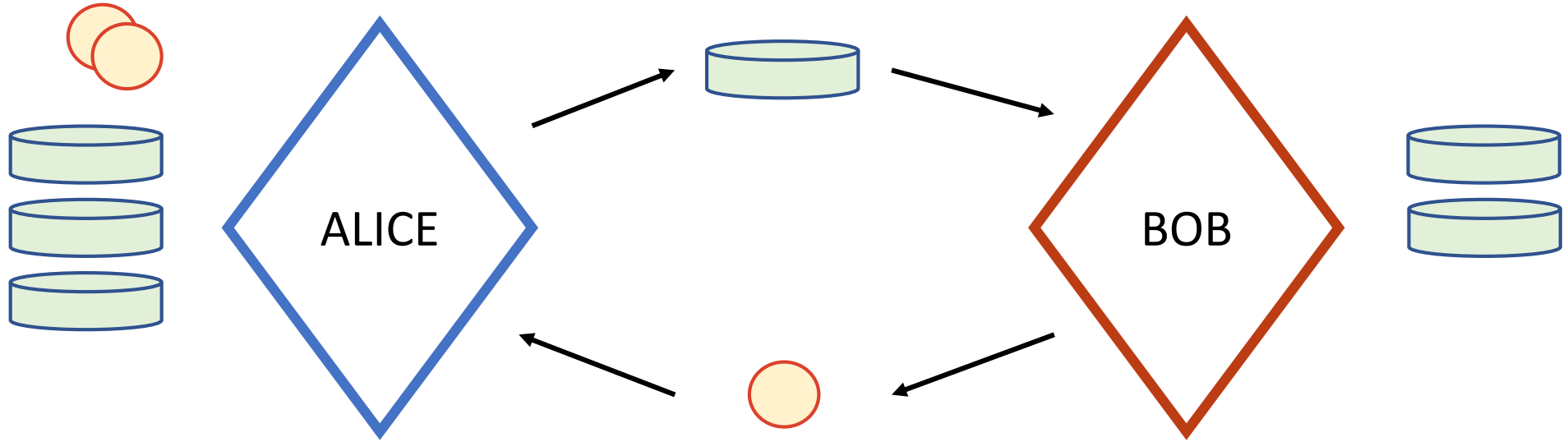
Traffic-to-coin channel

Traffic-to-coin channel



Traffic-to-coin channel

Traffic-to-coin channel



Traffic-to-coin channel

Proof of knowledge transfer in Teleport

Financial solution

When Alice is dishonest, she will not get coins since Bob will not confirm this payment. In the worst case scenario, Alice will send all correct chunks except the last one and will have signed receipts to publish on blockchain to get coins. Bob will not have the last chunk that he can download elsewhere

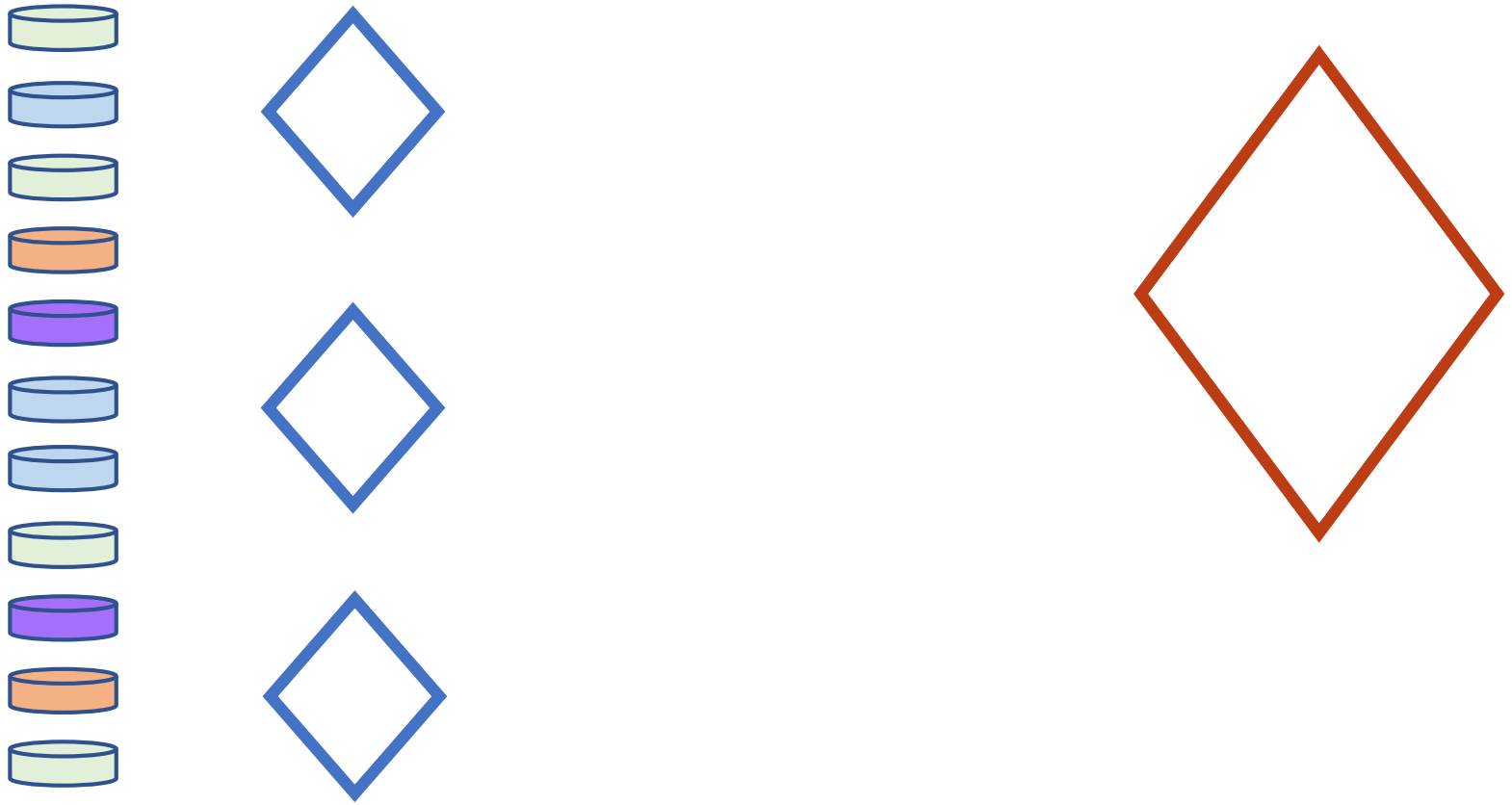
Bob's money is secure.

When Bob is dishonest, Alice will not send the following chunks. In the worst case scenario, Alice will transfer one chunk for free.

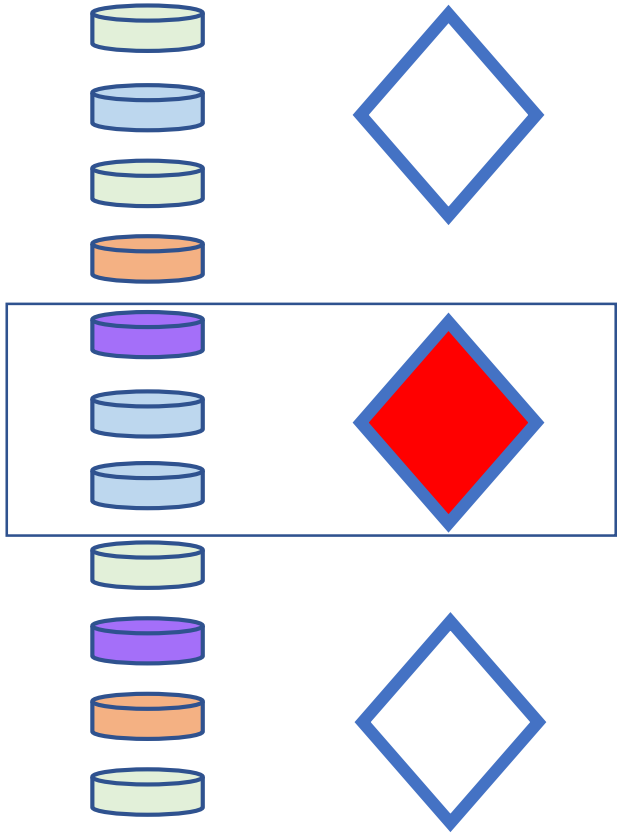
Alice may "lose" one revenue opportunity for one chunk

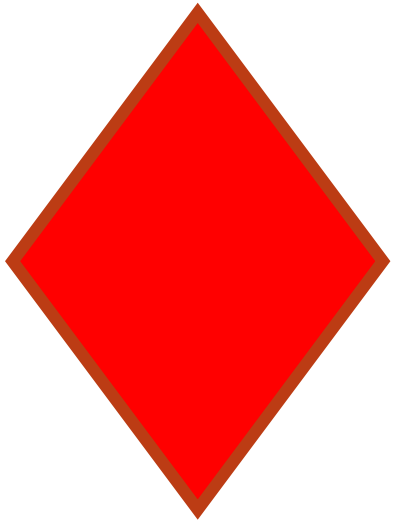
Psst... Bob!
Let’s form a group and sign
whatever we need to sign

Traffic delivery witness

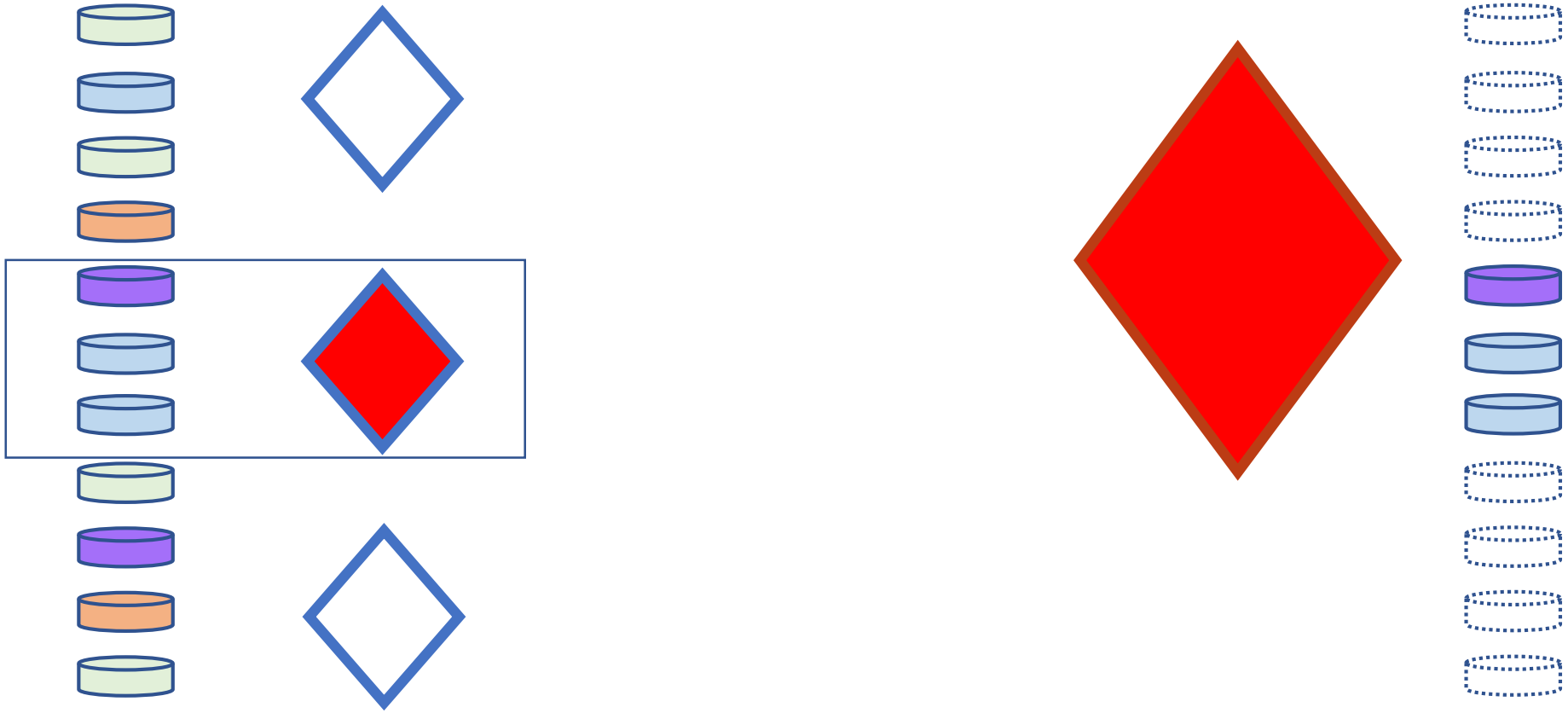


Traffic delivery witness





Traffic delivery witness



I have a single file that is so unique...

Proof of knowledge transfer in Teleport

Cryptographical solution

When network gets controversial information for Alice and Bob, they have to enter the special protocol of traffic delivery that must show who is lying.

If Alice or Bob refuses to do so the network punishes them.

Proof of knowledge transfer in Teleport

Cryptographical solution

When the network gets controversial information from Alice and Bob, they have to engage in the special protocol of traffic delivery that must show who is lying.

If Alice or Bob refuses to do so the network punishes them.

1. Start the protocol
2. Transfer the same file
3. ...
4. Profit!

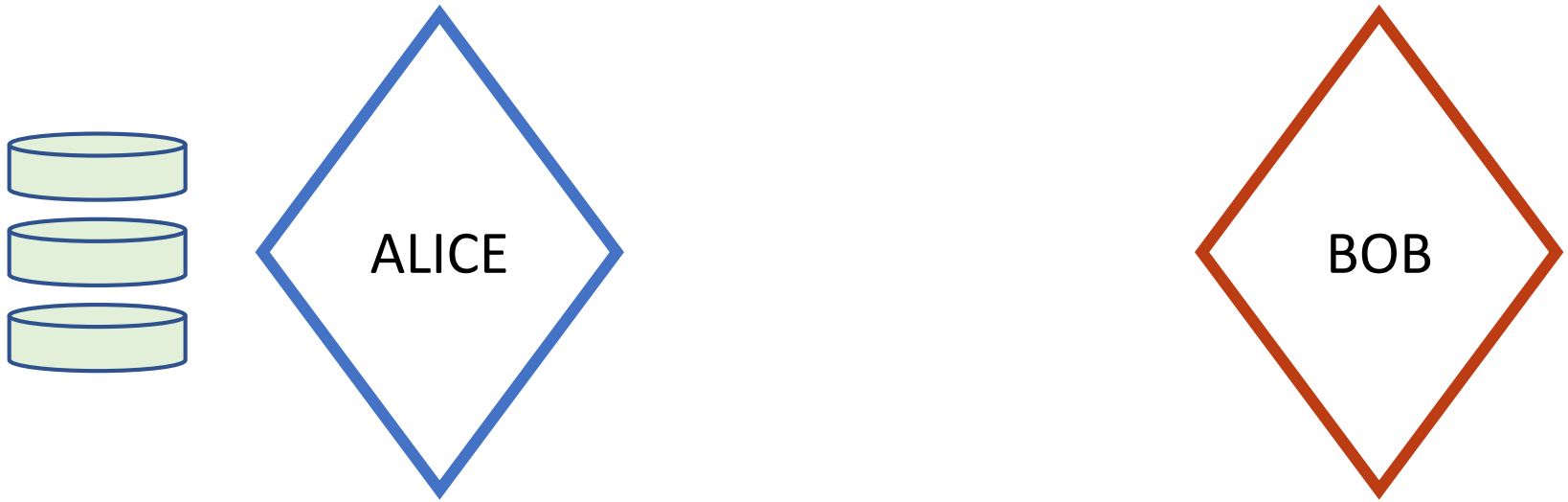
Proof of knowledge transfer in Teleport

Cryptographical solution

Oblivious transfer

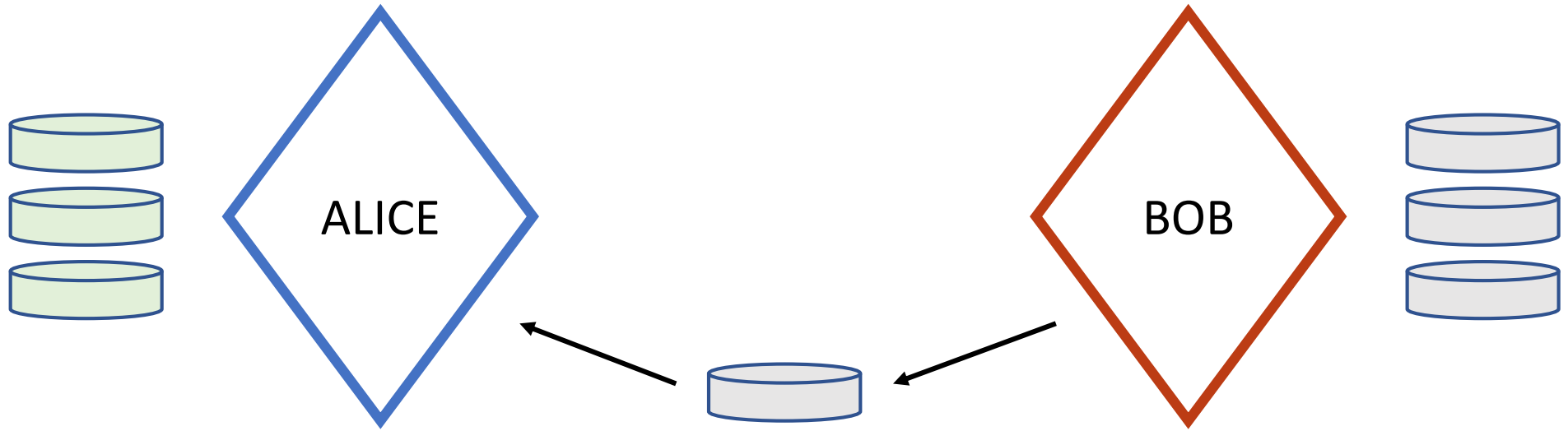
Secure multi-party computation

Proof of knowledge transfer

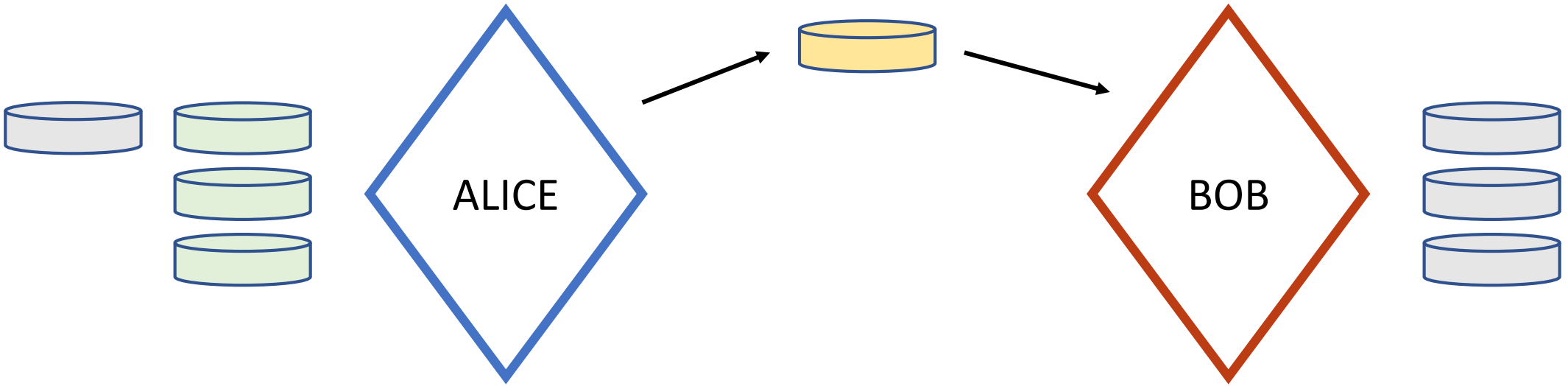


Proof of knowledge transfer

Proof of knowledge transfer



Proof of knowledge transfer

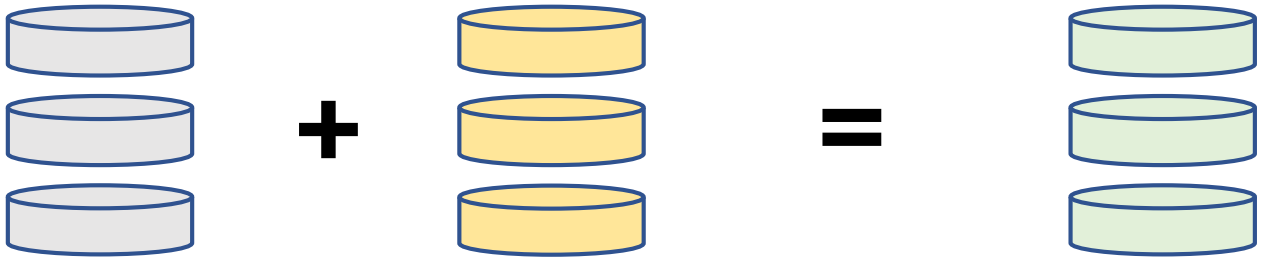


Proof of knowledge transfer

Proof of knowledge transfer

Proof of knowledge transfer

Proof of knowledge transfer



Zero-Knowledge proof of knowledge transfer

Perm summer school on blockchain 2018