Bitcoin in the Post-Quantum Era

A solution for a problem that doesn't exist yet… (as far as we know)
How often does that happen?

Dragos I. Ilie

Centre for Cryptocurrency Research and Engineering
Imperial College London

Imperial College
London

Plan for the next 45 mins

- ➔ Quantum World
  - ◆ Basics: Qubit, Superposition
  - ◆ Grover’s Algorithm – Unstructured Search
  - ◆ Shor’s Algorithm – Structured Search
- ➔ Problem in Bitcoin
  - ◆ Revealed Public Keys
  - ◆ (Live) Transaction Hijacking
- ➔ One Solution
  - ◆ Quantum Resistant Surrogate
  - ◆ Proof of Common Ownership
- ➔ Questions

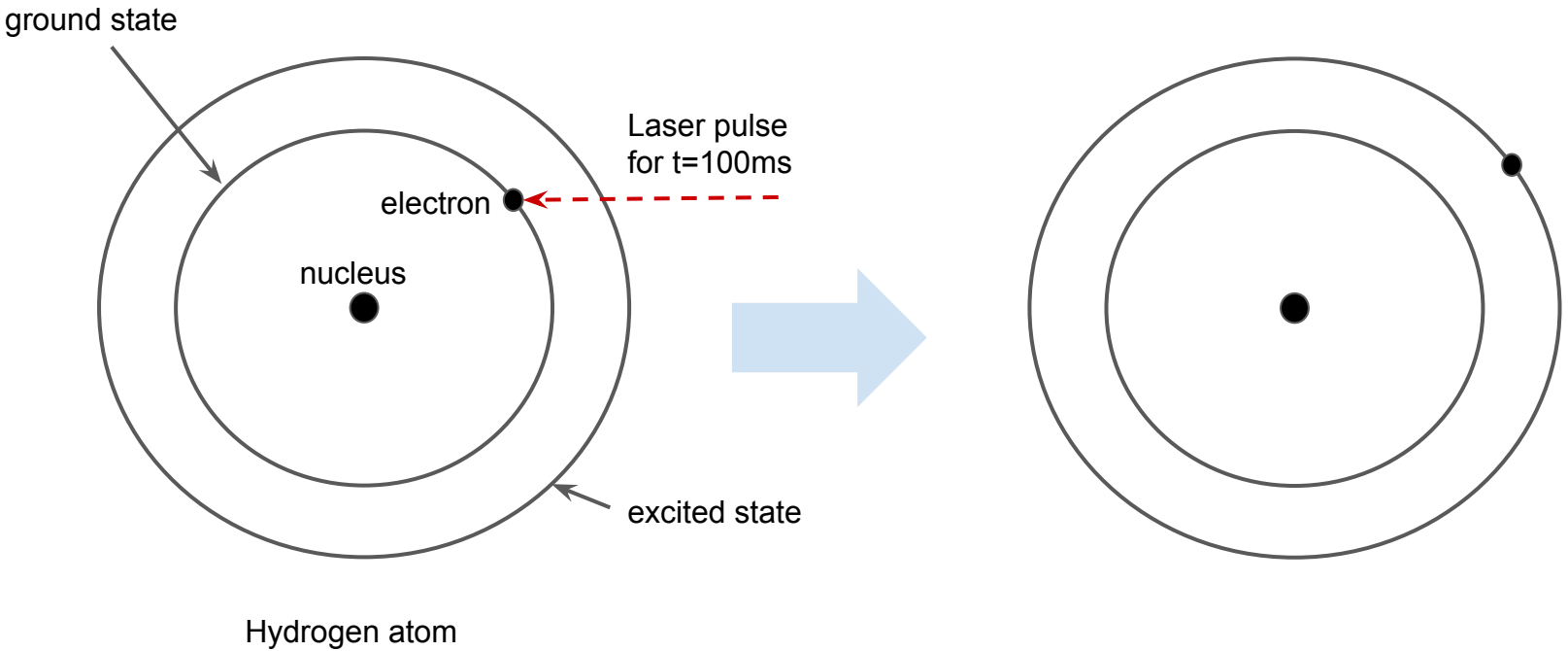
Quantum Mechanics – Qubit

Any quantum system with two states (also called basis states):

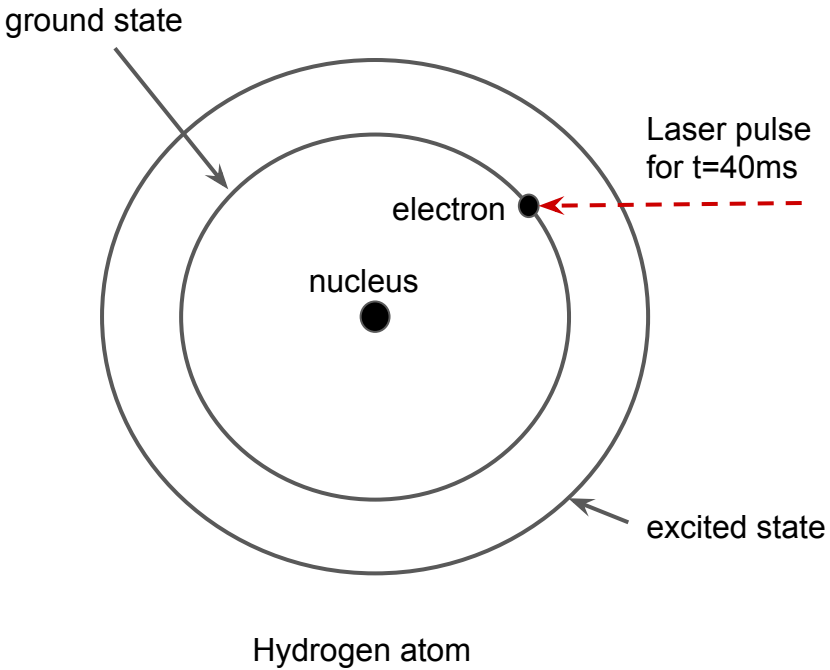
- Electron on ground energy level or excited energy level
- Photon polarized horizontally or vertically
- Electron spin up or down

When measuring the qubit, the result will be one of the basis states, buuuuuut… when we are not measuring, the qubit can exist in a superposition of basis states.

Quantum Mechanics – Superposition

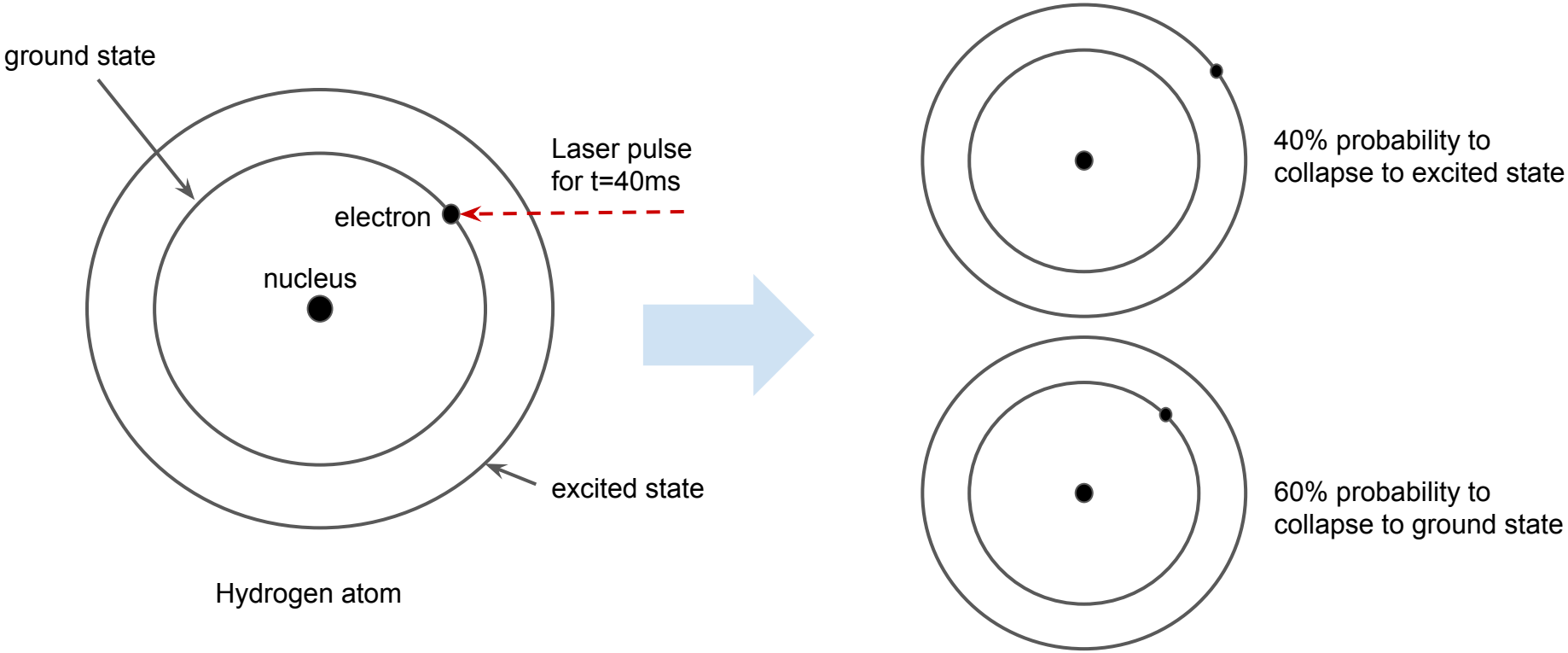


Quantum Mechanics – Superposition



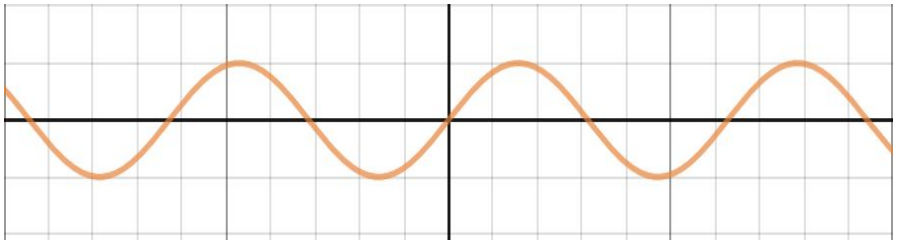
Hydrogen atom

Quantum Mechanics – Superposition

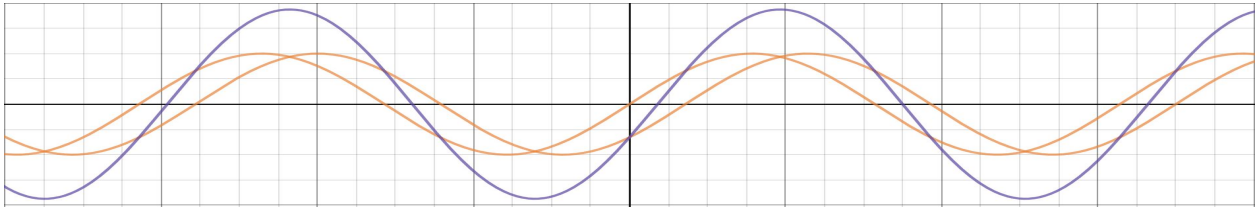


Quantum Mechanics – Everything is a wave?

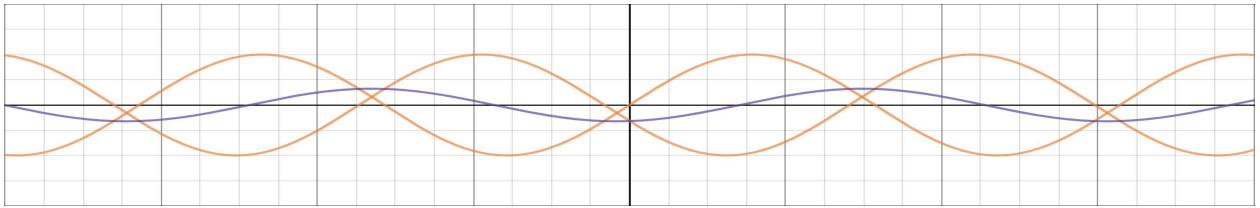
Wave:



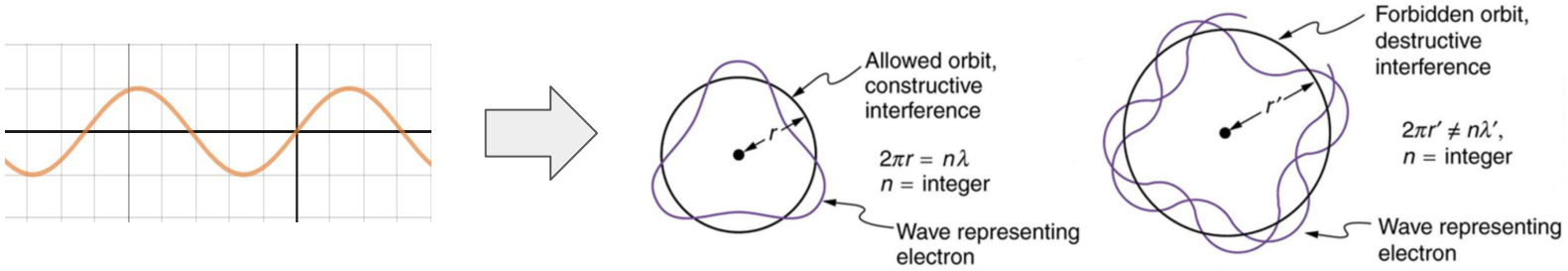
Constructive Interference:

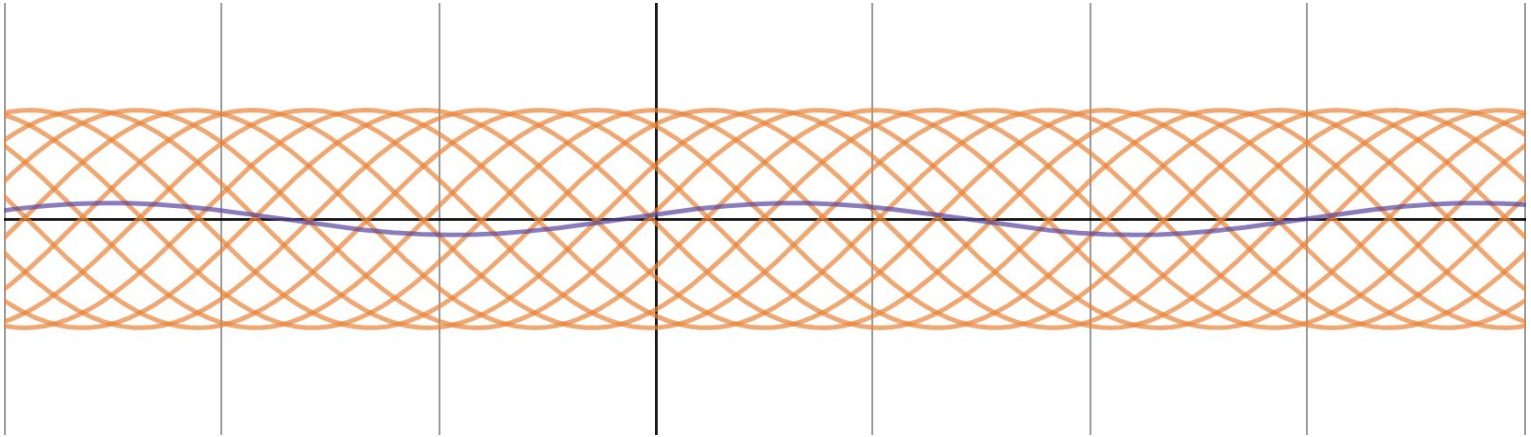


Destructive Interference:



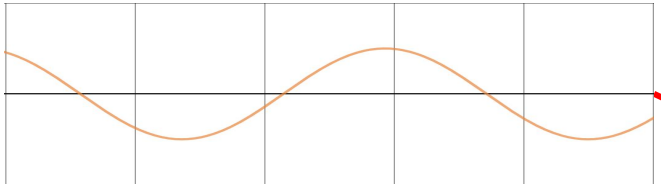
Quantum Mechanics – Why collapse?



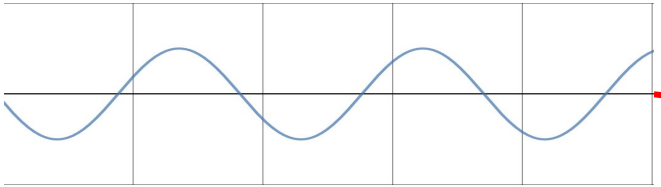


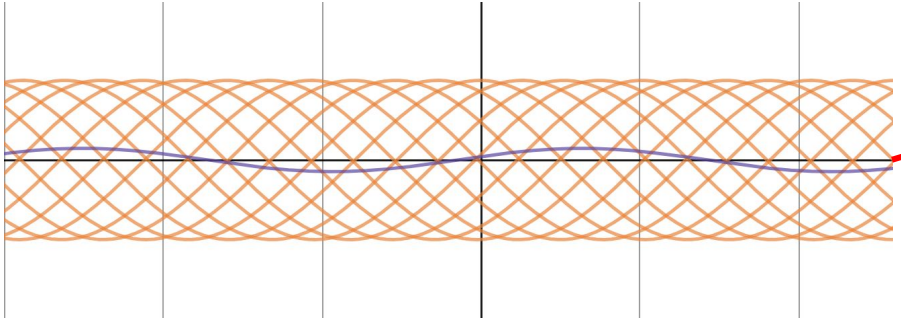
Quantum Mechanics – Why collapse?

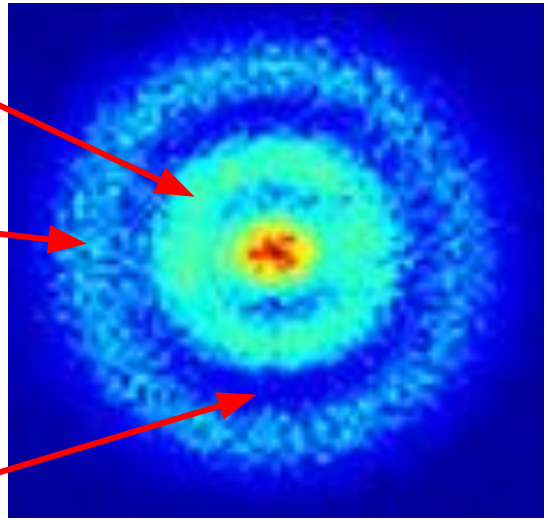
Ground Level:



Excited Level:



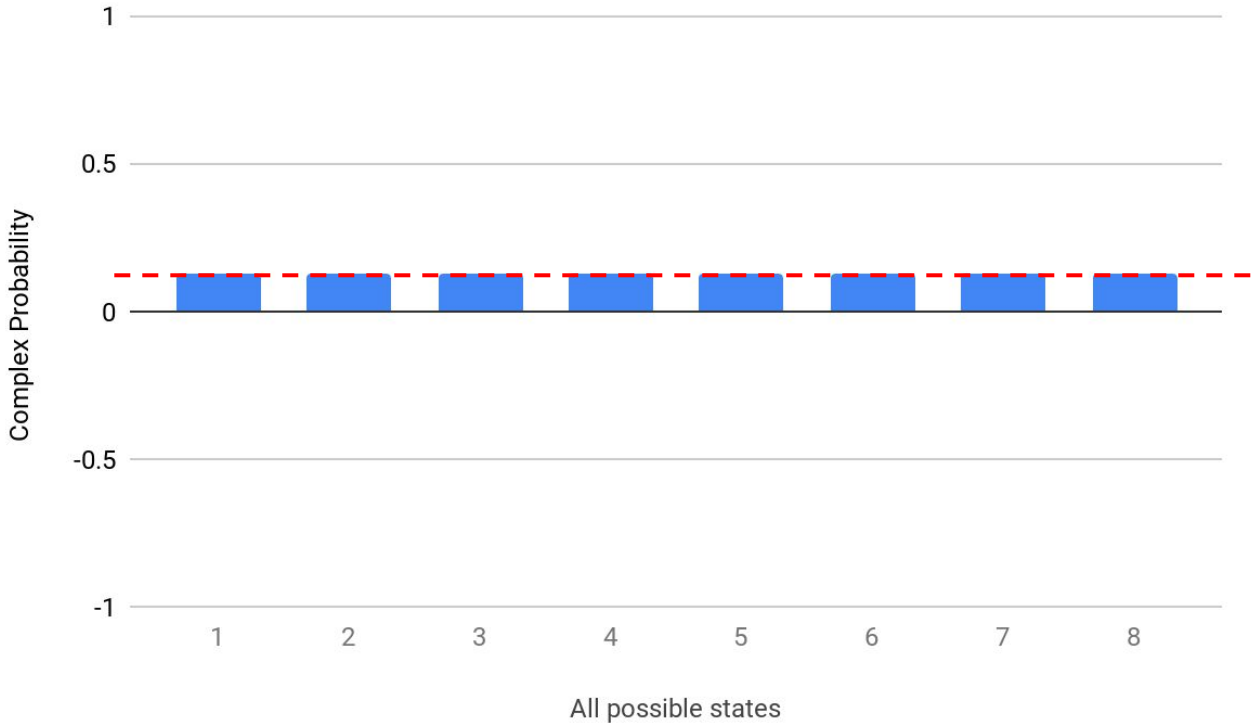




Quantum Computing – Grover's Algorithm

Searching unstructured data. Eg: Find x such that x+1 = 5.

Arrange state as a superposition of all possible inputs.

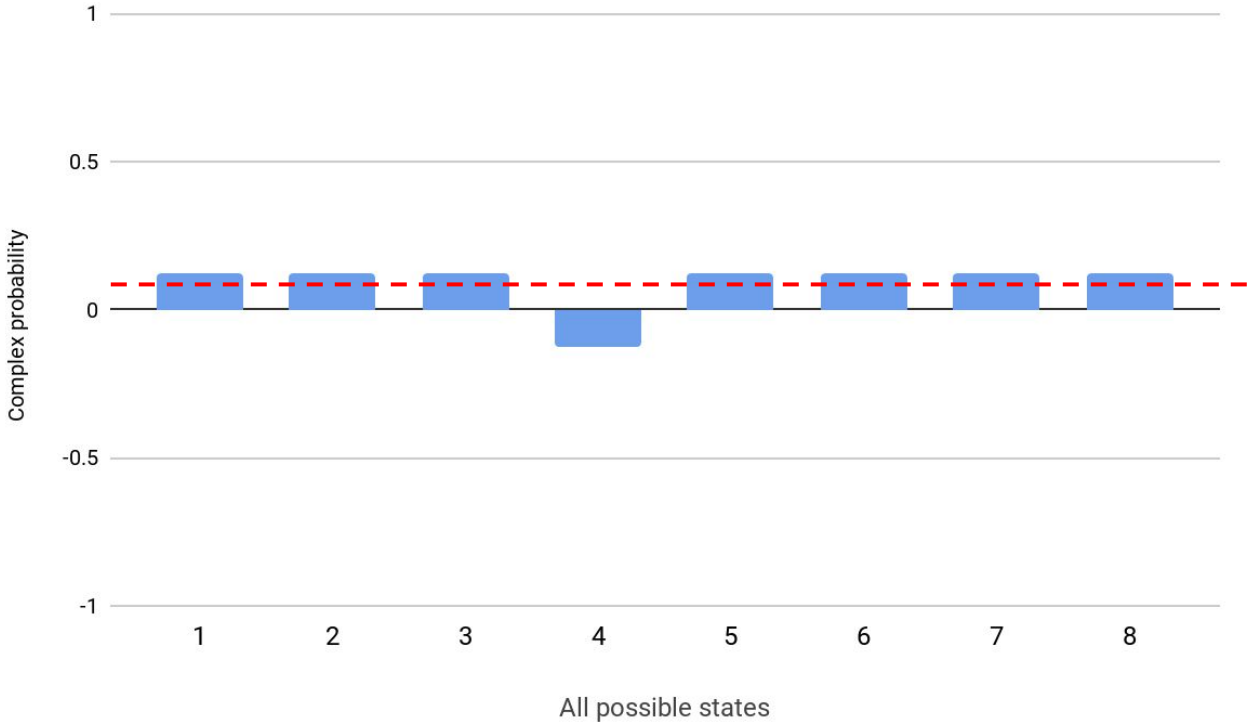


Quantum Computing – Grover's Algorithm

Searching unstructured data. Eg: Find x such that x+1 = 5.

Arrange state as a superposition of all possible inputs.

Invert wanted input.
$x \rightarrow -1^{f(x)} x$, where
$f(x) = \begin{cases} 1, \text{ if } x+1=5 \\ 0, \text{ otherwise} \end{cases}$



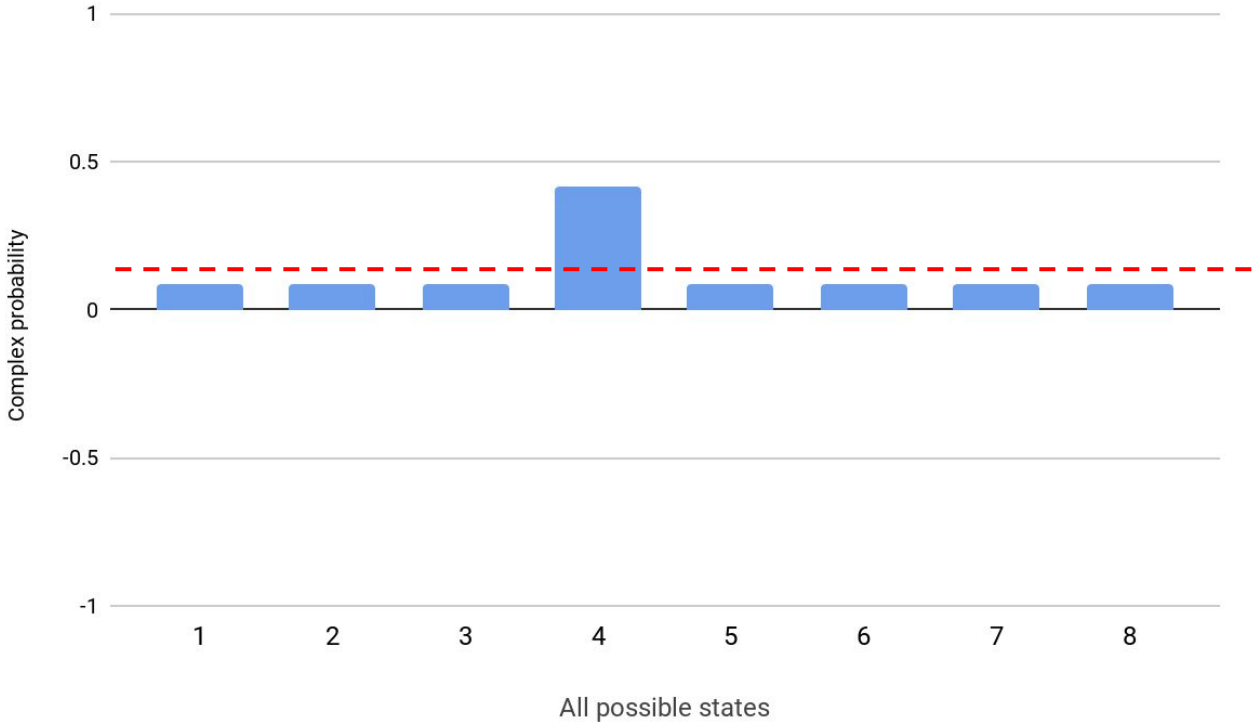
Quantum Computing – Grover's Algorithm

Searching unstructured data. Eg: Find x such that x+1 = 5.

Arrange state as a superposition of all possible inputs.

Invert wanted input.
$x \rightarrow -1^{f(x)} x$, where
$f(x) = \begin{cases} 1, \text{ if } x+1=5 \\ 0, \text{ otherwise} \end{cases}$

Reflect around average.
$x \rightarrow 2A - x$



Quantum Computing – Grover's Algorithm

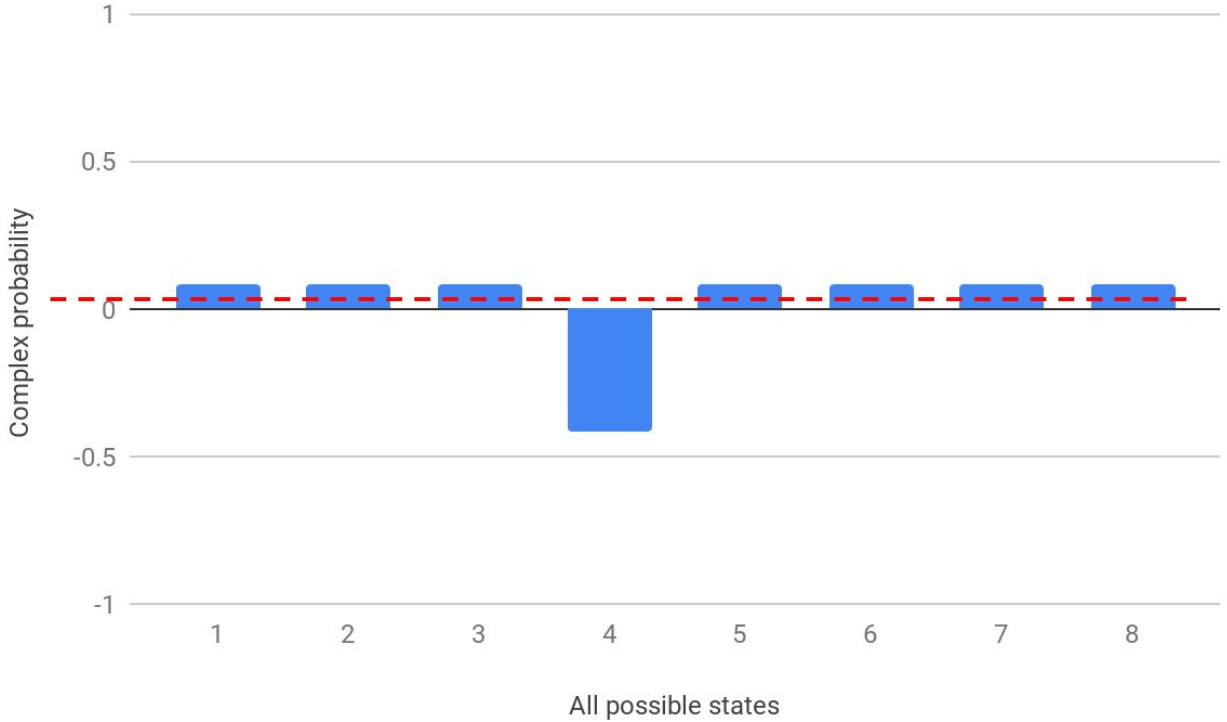
Searching unstructured data. Eg: Find x such that x+1 = 5.

Arrange state as a superposition of all possible inputs.

Invert wanted input.
$x \rightarrow -1^{f(x)} x$, where
$f(x) = \begin{cases} 1, \text{ if } x+1=5 \\ 0, \text{ otherwise} \end{cases}$

Reflect around average.
$x \rightarrow 2A - x$

And repeat previous two steps.



Quantum Computing – Grover's Algorithm

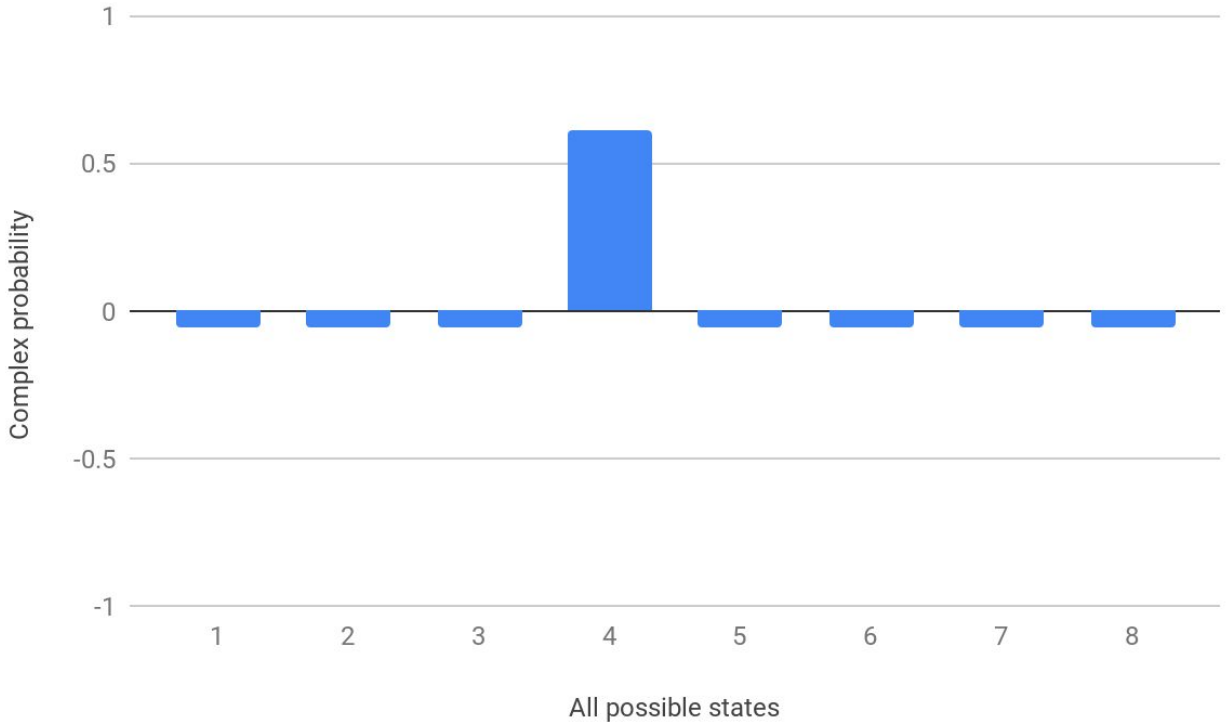
Searching unstructured data. Eg: Find x such that x+1 = 5.

Arrange state as a superposition of all possible inputs.

Invert wanted input.
$x \rightarrow -1^{f(x)} x$, where
$f(x) = \begin{cases} 1, \text{ if } x+1=5 \\ 0, \text{ otherwise} \end{cases}$

Reflect around average.
$x \rightarrow 2A - x$

And repeat previous two steps. For maximum probability you need to repeat exactly $\sqrt{N}$ times.



Quantum Computing – Grover's Algorithm

superposition

invert around the mean: x → 2A - x

$$|00...00\rangle \Rightarrow \alpha \sum_{x} |x\rangle \Rightarrow -\alpha|m\rangle + \alpha \sum_{x \neq m} |x\rangle \Rightarrow (2A+\alpha)|m\rangle + (2A-\alpha) \sum_{x \neq m} |x\rangle$$

invert amplitude of wanted value

- Searching unstructured data for some relatively rare value; e.g. finding a nonce s.t. H(M || nonce) < t
- Time: O(√N) queries vs O(N)
- Space: O(log(N)) qubits
- Relevant Uses:
  - Breaking hashes (not quite because $N = 2^{256}$ so $\sqrt{N} = 2^{128}$)
  - Mining (debatable because we actually need only √(N/t) steps or less if we run the computation in parallel on multiple quantum computers)

Quantum Computing – Shor's Algorithm

Searching structured data.

- Solves the Hidden subgroup problem (period finding)
  - Factoring (RSA)
  - Discrete Logarithm (ECDSA)
- Time: $O(n^3)$ vs $O(2^n)$
- Space: O(n) qubits (approx. 6n)
- With about 1500 qubits you can break an ECDSA private key of 256 bits

Find period r of function f:

Superposition: $|00...00\rangle|00...00\rangle \Rightarrow \sum_x |x\rangle|00..00\rangle$

Compute f: $\sum_x |x\rangle|00..00\rangle \Rightarrow \sum_x |x\rangle|f(x)\rangle$

Measure f(x): $\Rightarrow \sum_x |x\rangle|f(x)\rangle \Rightarrow \sum_j \alpha|jr+l\rangle|m\rangle$

Measure f(x): $\Rightarrow \sum_x |x\rangle|f(x)\rangle \Rightarrow \sum_j \alpha|jr+l'\rangle|m'\rangle$

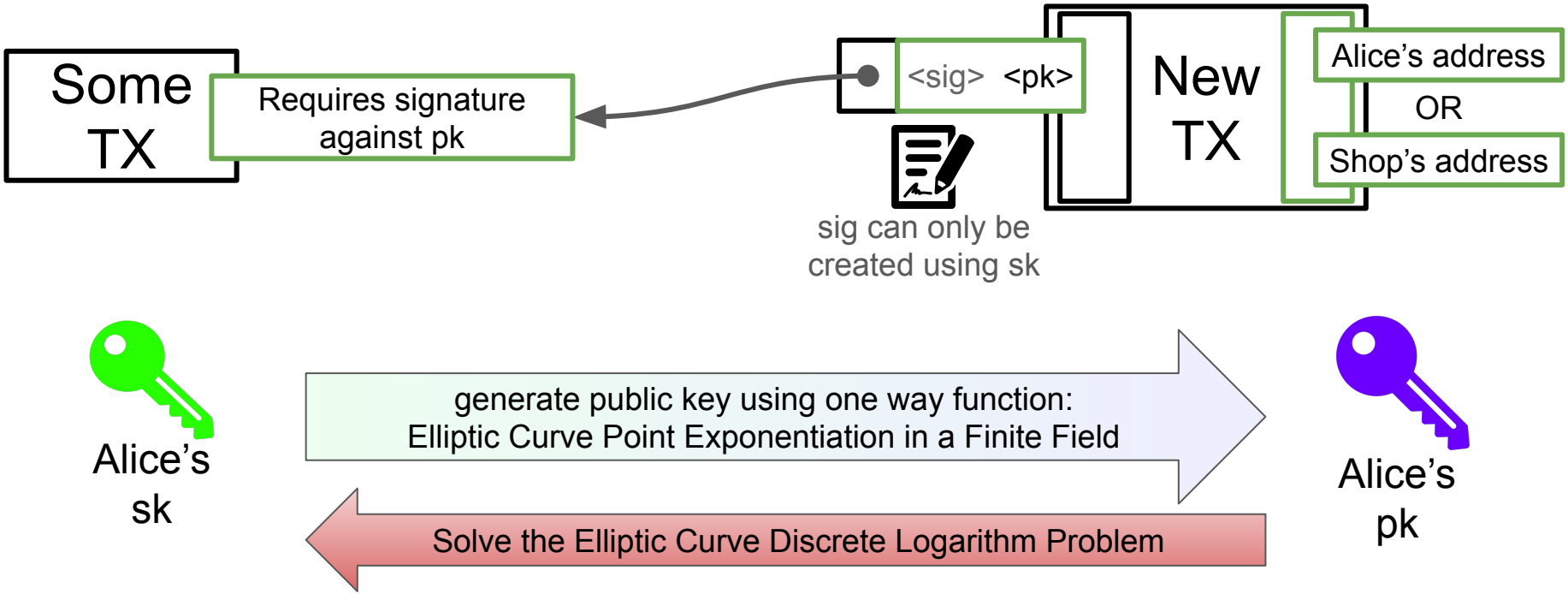
Apply Quantum Fourrier Transform: $\Rightarrow (\sum_j \alpha|jr\rangle)$

Measure register: $|j_1 r\rangle$ OR $|j_2 r\rangle$

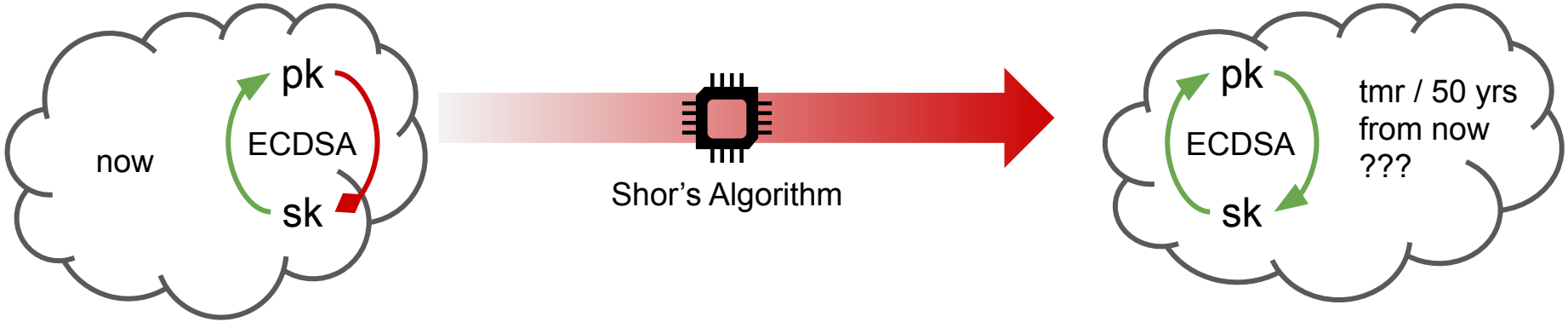
Enough Quantum… Let’s talk about Bitcoin

Digital Signatures in Bitcoin

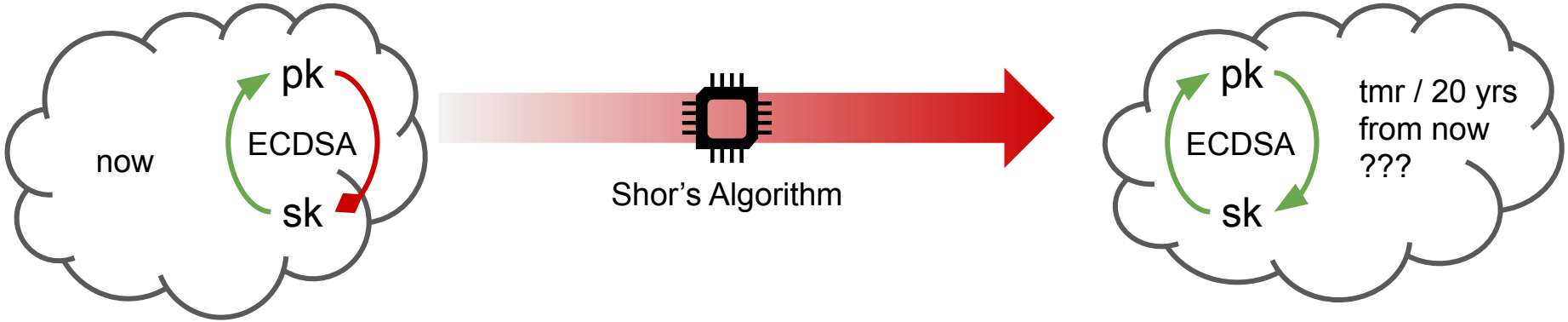
Elliptic Curve Digital Signature Algorithm (ECDSA)



Problem – ECDSA will be broken… at some point



Problem – ECDSA will be broken… at some point



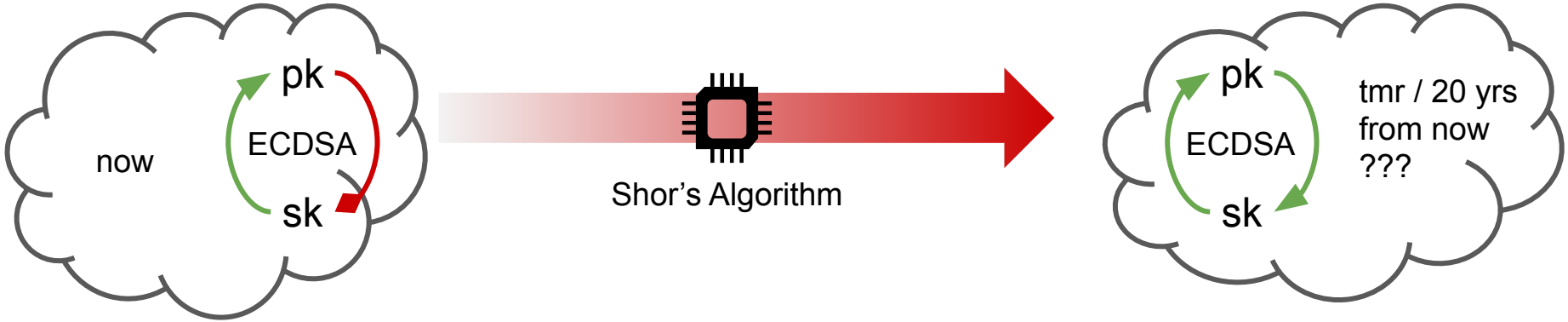
All revealed public keys are under attack, even for slow Quantum Computers!

Bitcoins aggregated by public key visibility

10.2%

pk in output

Problem – ECDSA will be broken… at some point

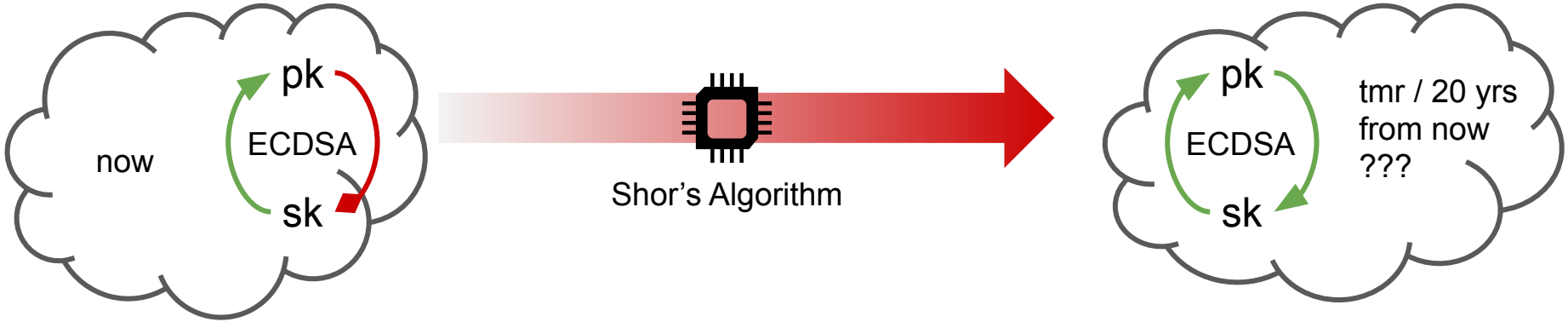


All revealed public keys are under attack, even for slow Quantum Computers!

Bitcoins aggregated by public key visibility

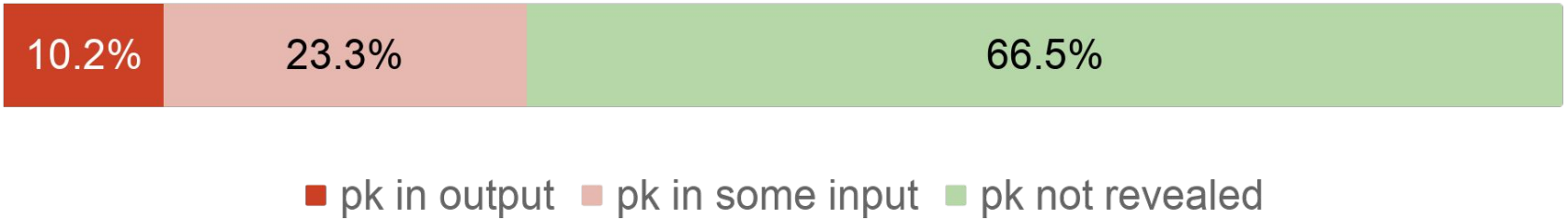
pk in output pk in some input

Problem – ECDSA will be broken… at some point

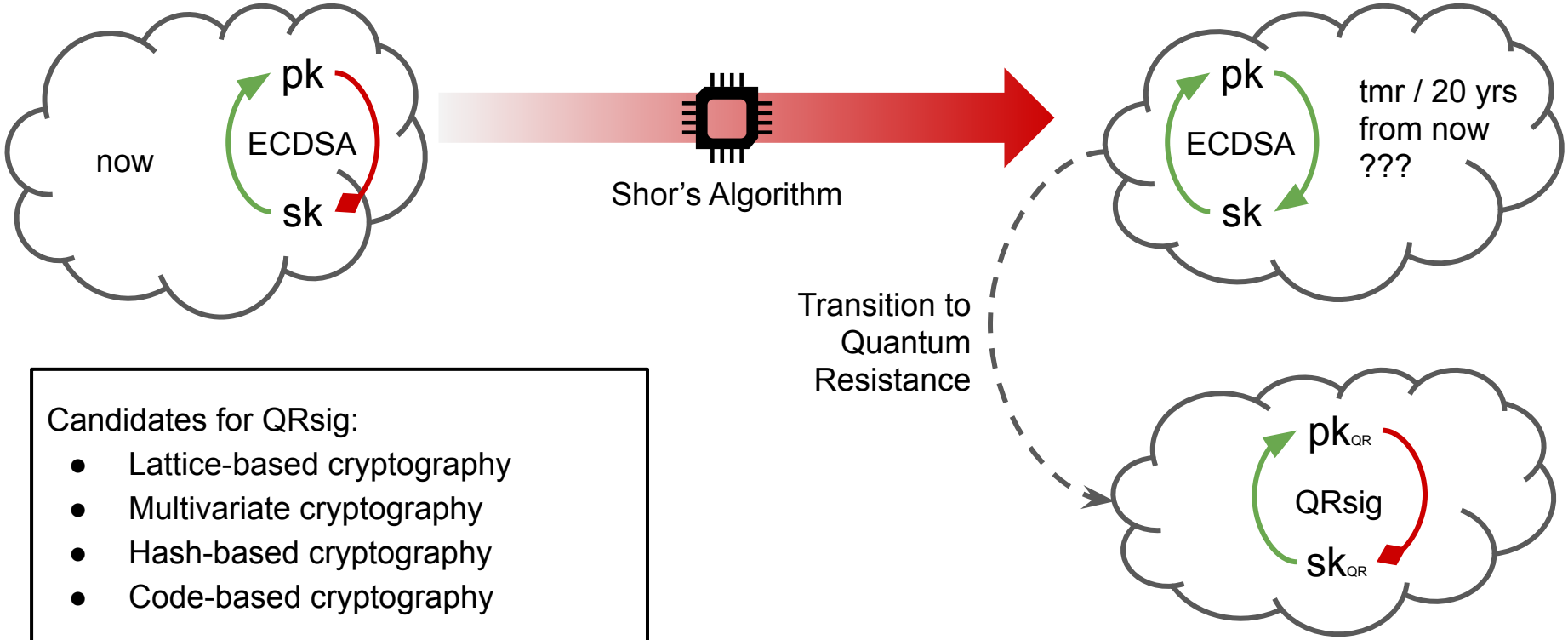


All revealed public keys are under attack, even for slow Quantum Computers!

Bitcoins aggregated by public key visibility



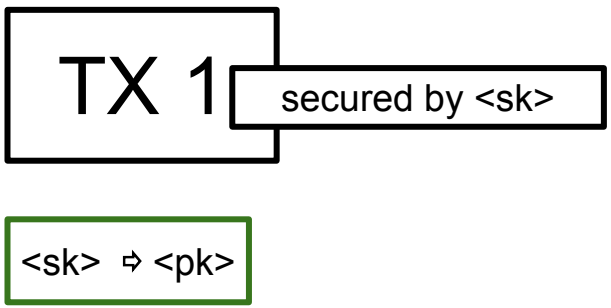
Solution – Easy… replace ECDSA asap

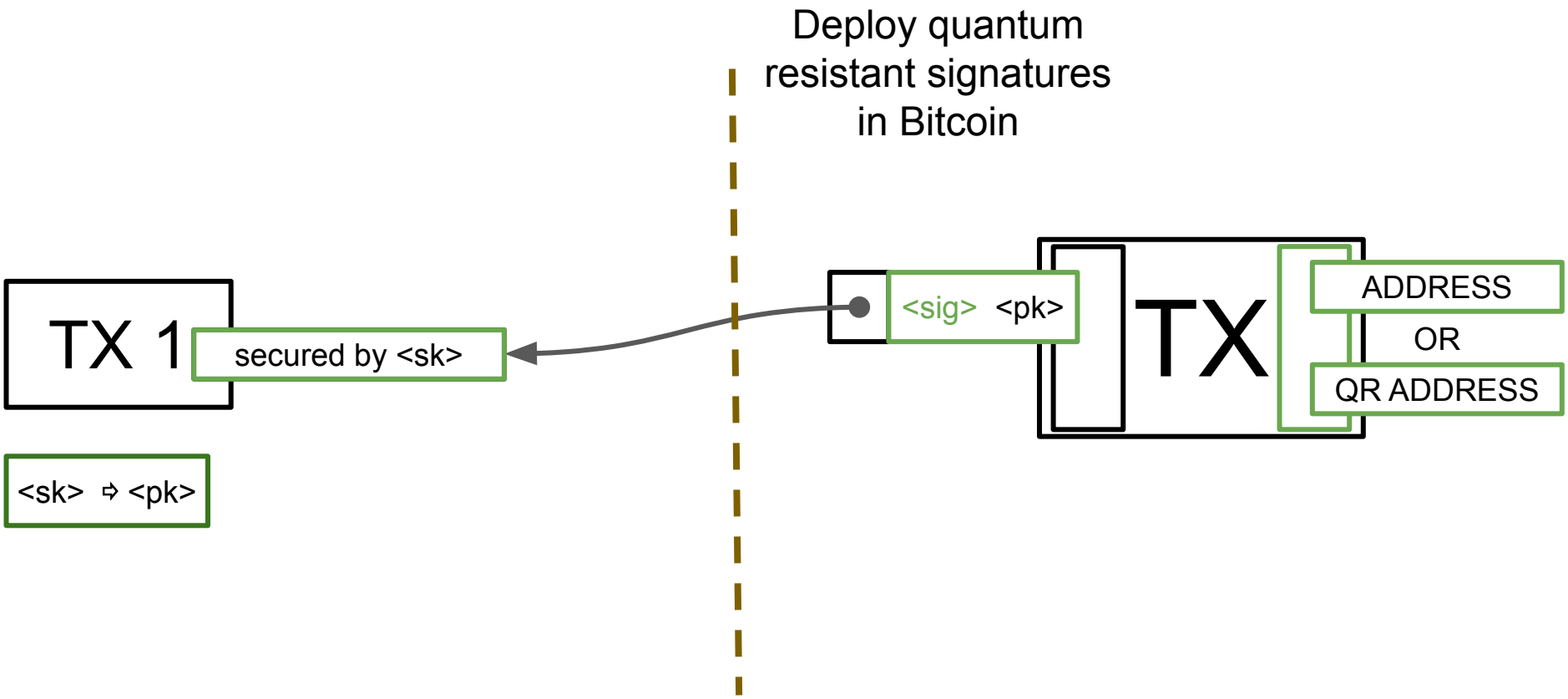


Candidates for QRsig:

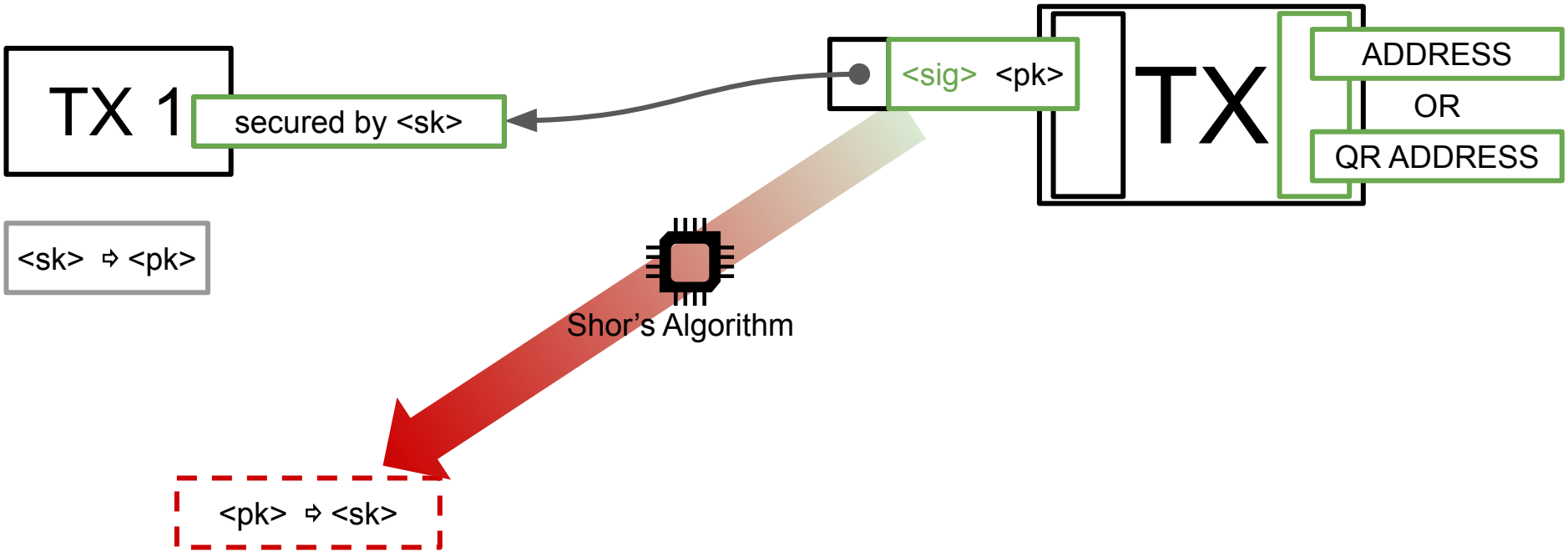
- Lattice-based cryptography
- Multivariate cryptography
- Hash-based cryptography
- Code-based cryptography

Deploy quantum resistant signatures in Bitcoin

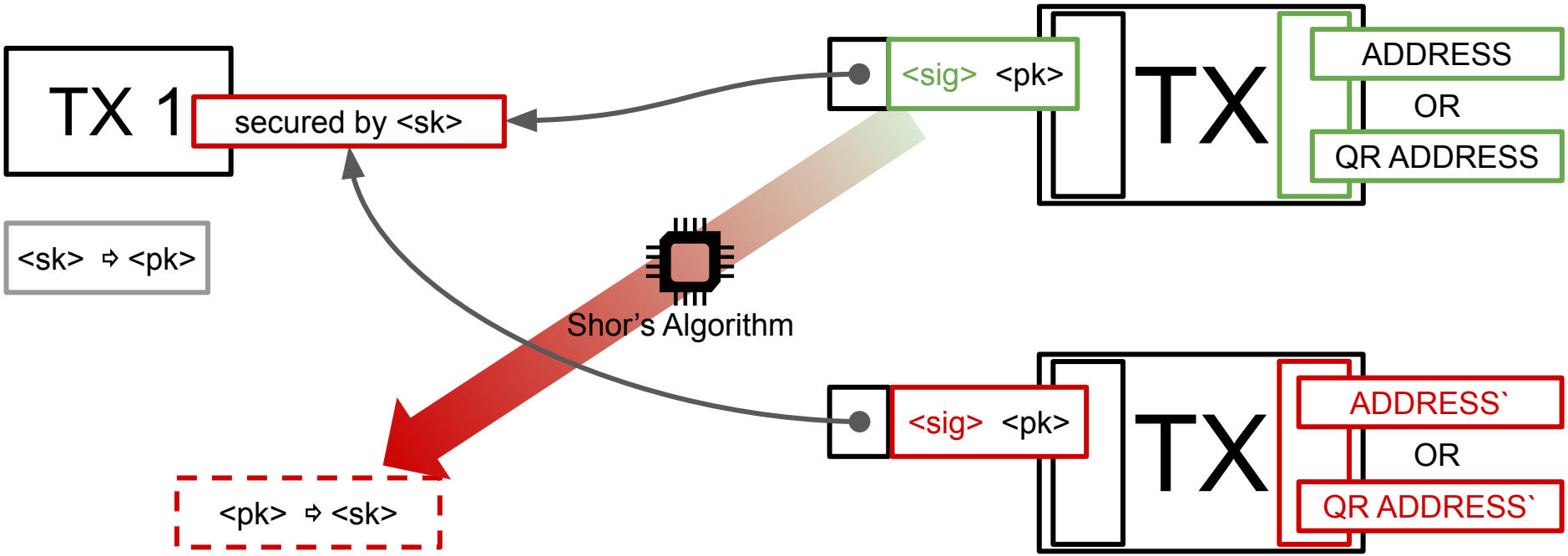



Deploy quantum
resistant signatures
in Bitcoin
TX 1
secured by <sk>
<sig> <pk>
TX
ADDRESS
OR
QR ADDRESS
<sk> ⇨ <pk>

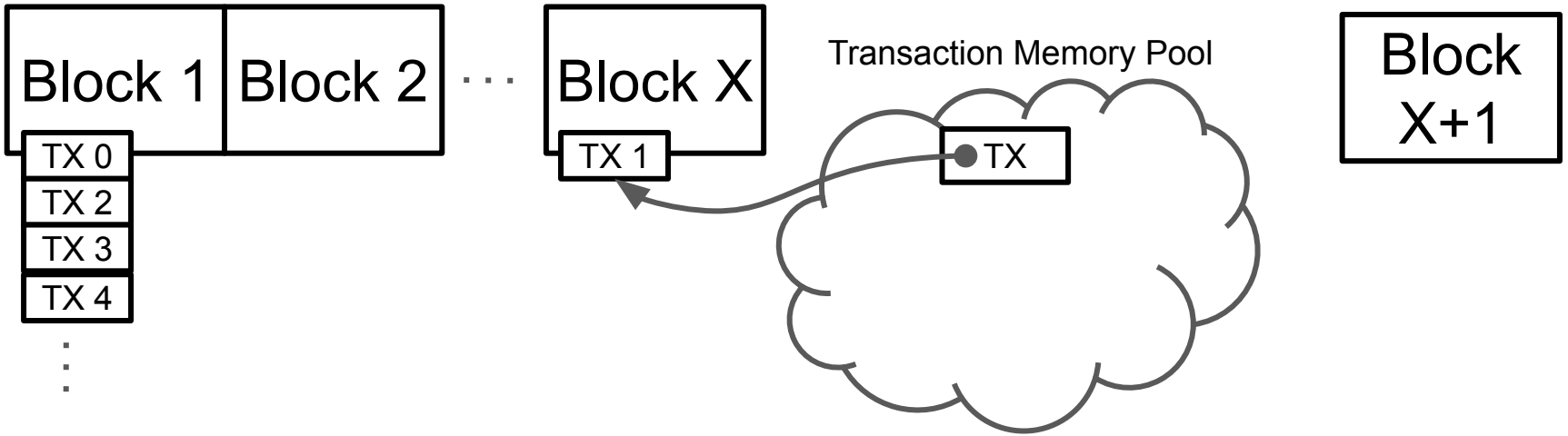
Problem – Transaction Hijacking



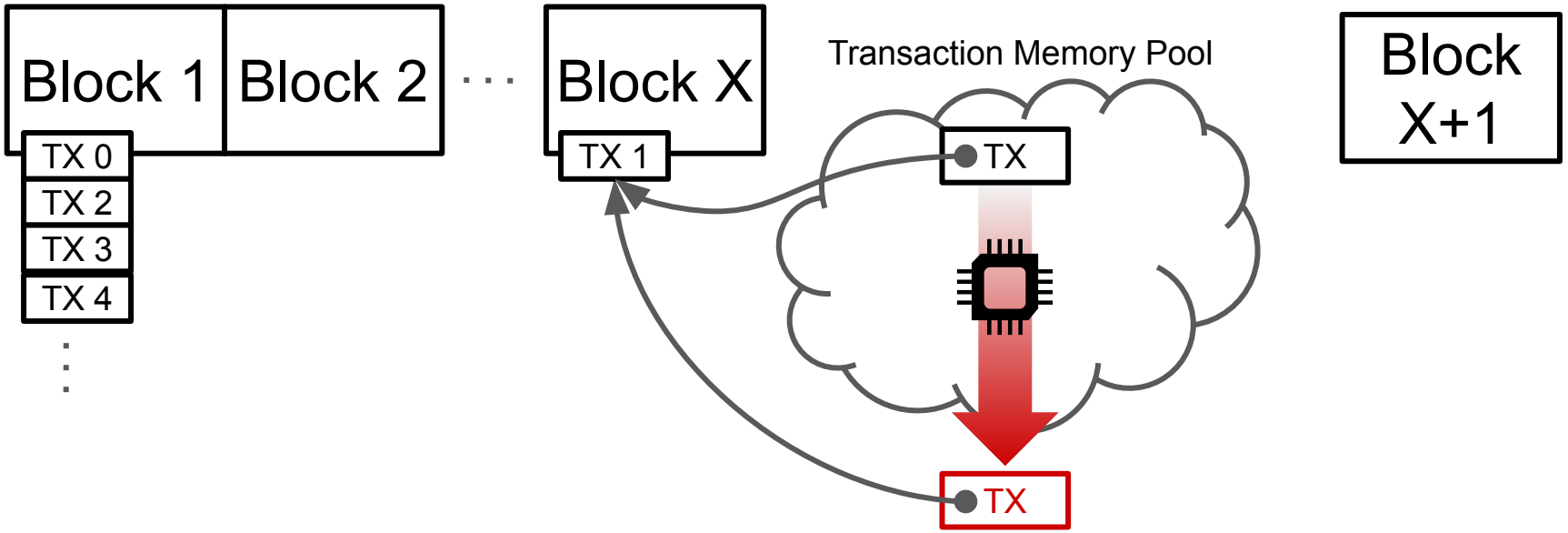
Problem – Transaction Hijacking



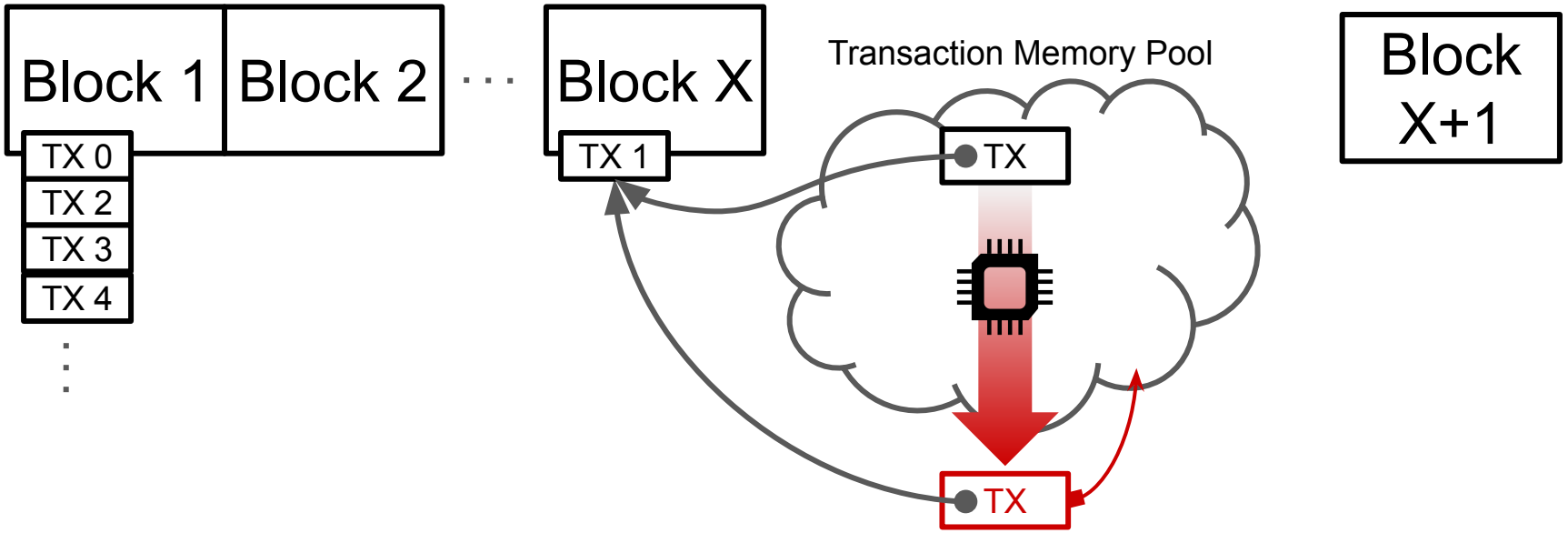
Problem – Live Transaction Hijacking



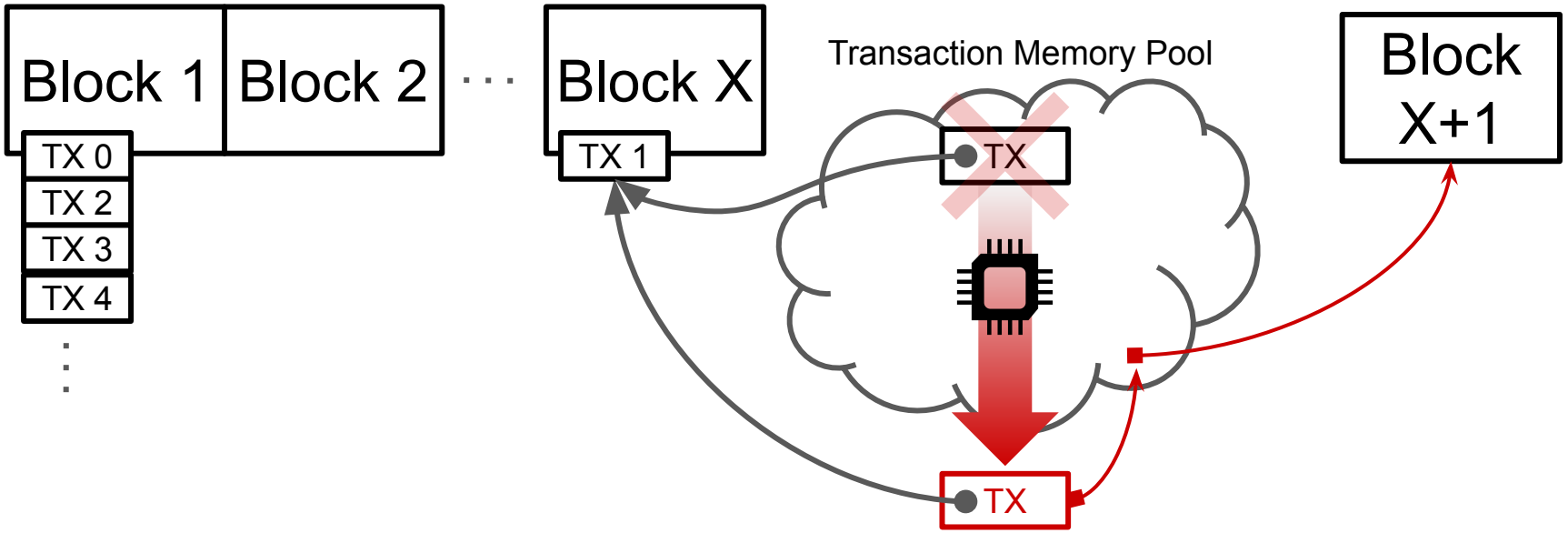
Problem – Live Transaction Hijacking



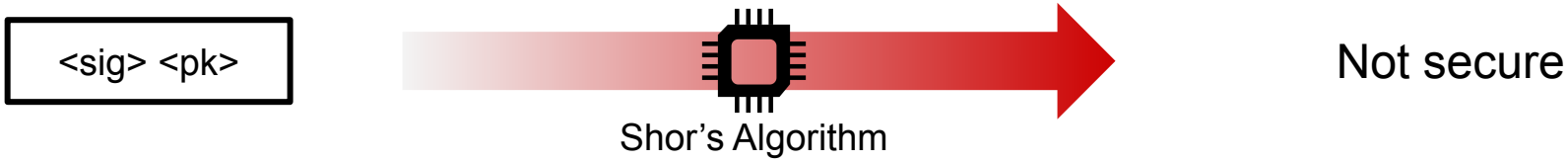
Problem – Live Transaction Hijacking



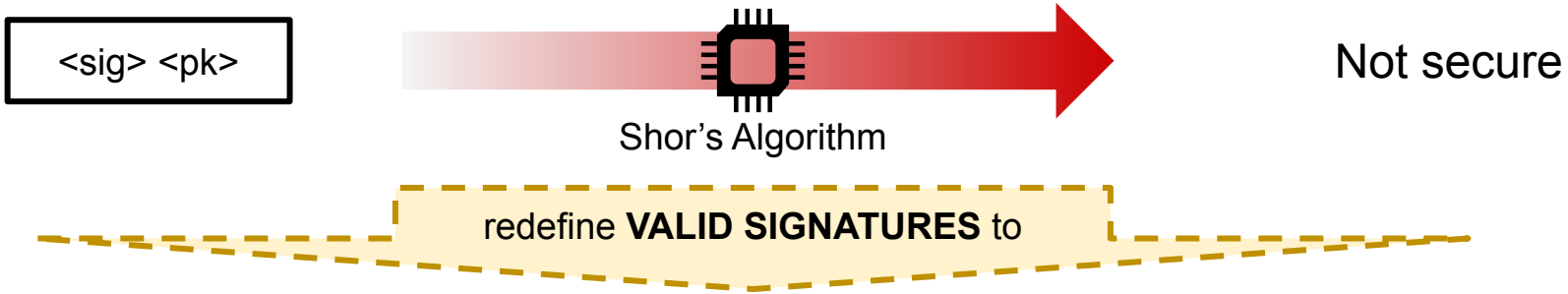
Problem – Live Transaction Hijacking



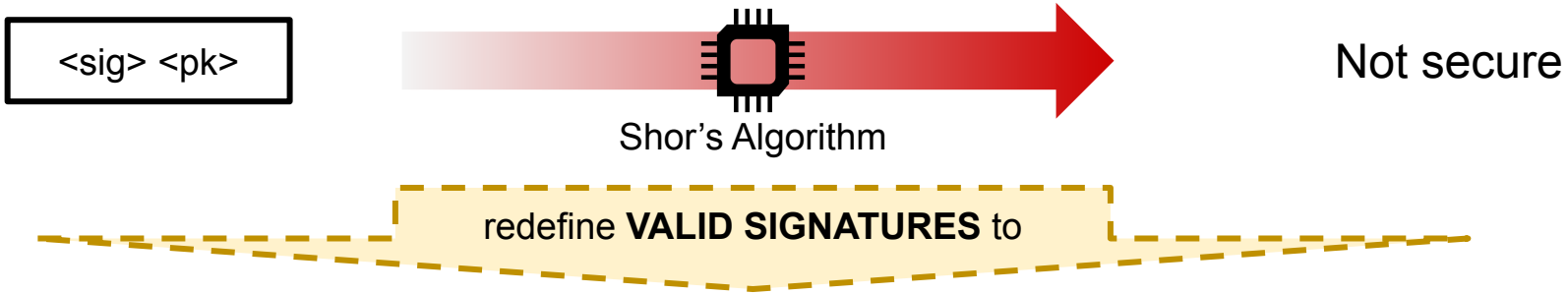
Solution – Quantum Surrogate



Solution – Quantum Surrogate



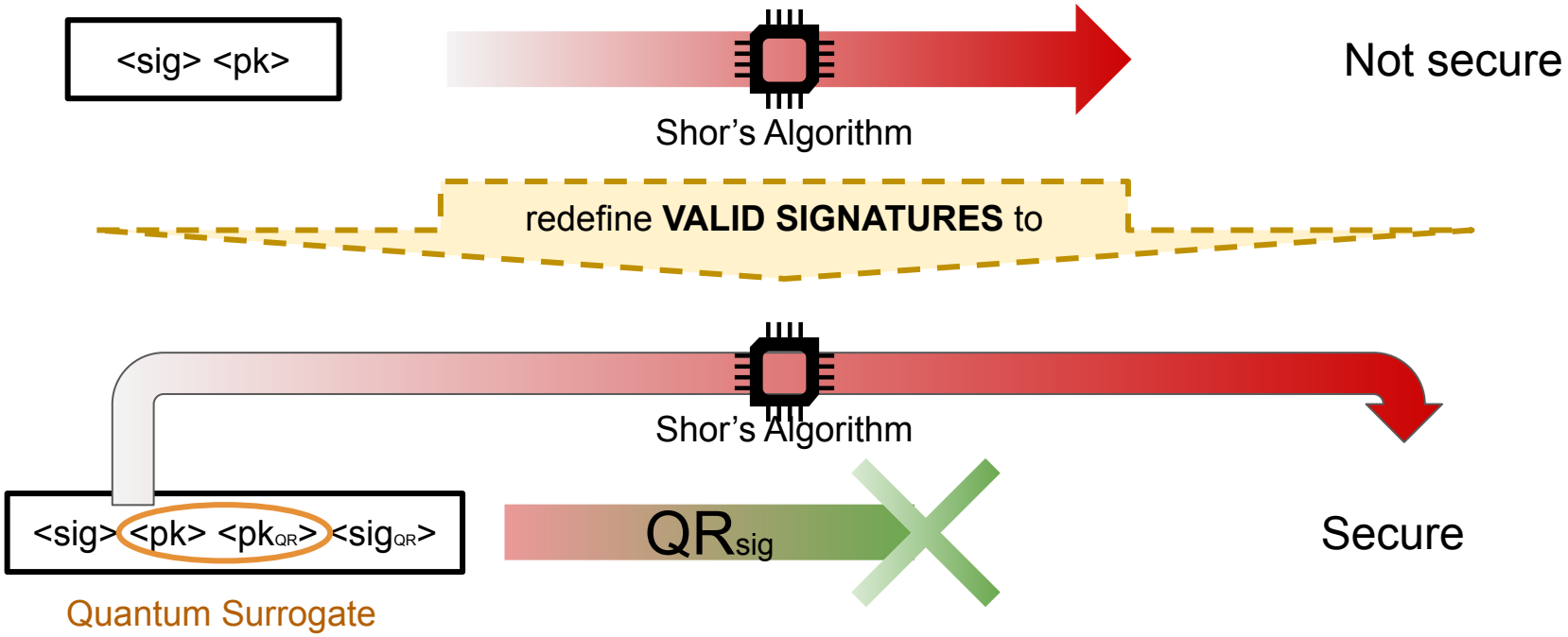
Solution – Quantum Surrogate



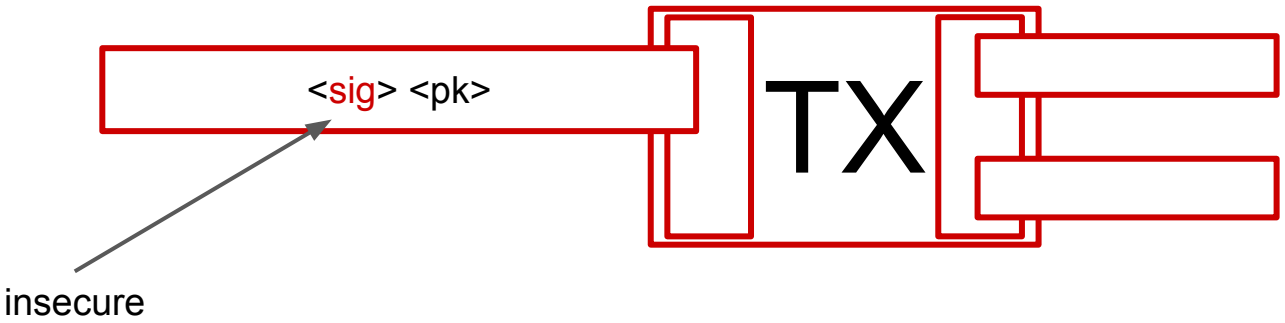
<sig> <pk> <pk$_{QR}$> <sig$_{QR}$>

Quantum Surrogate

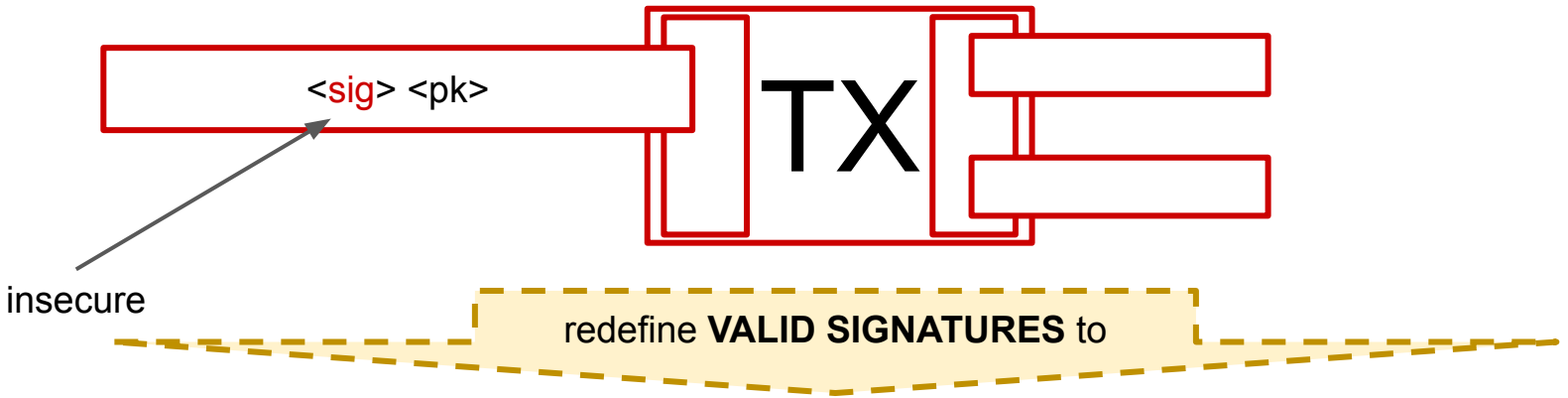
Solution – Quantum Surrogate



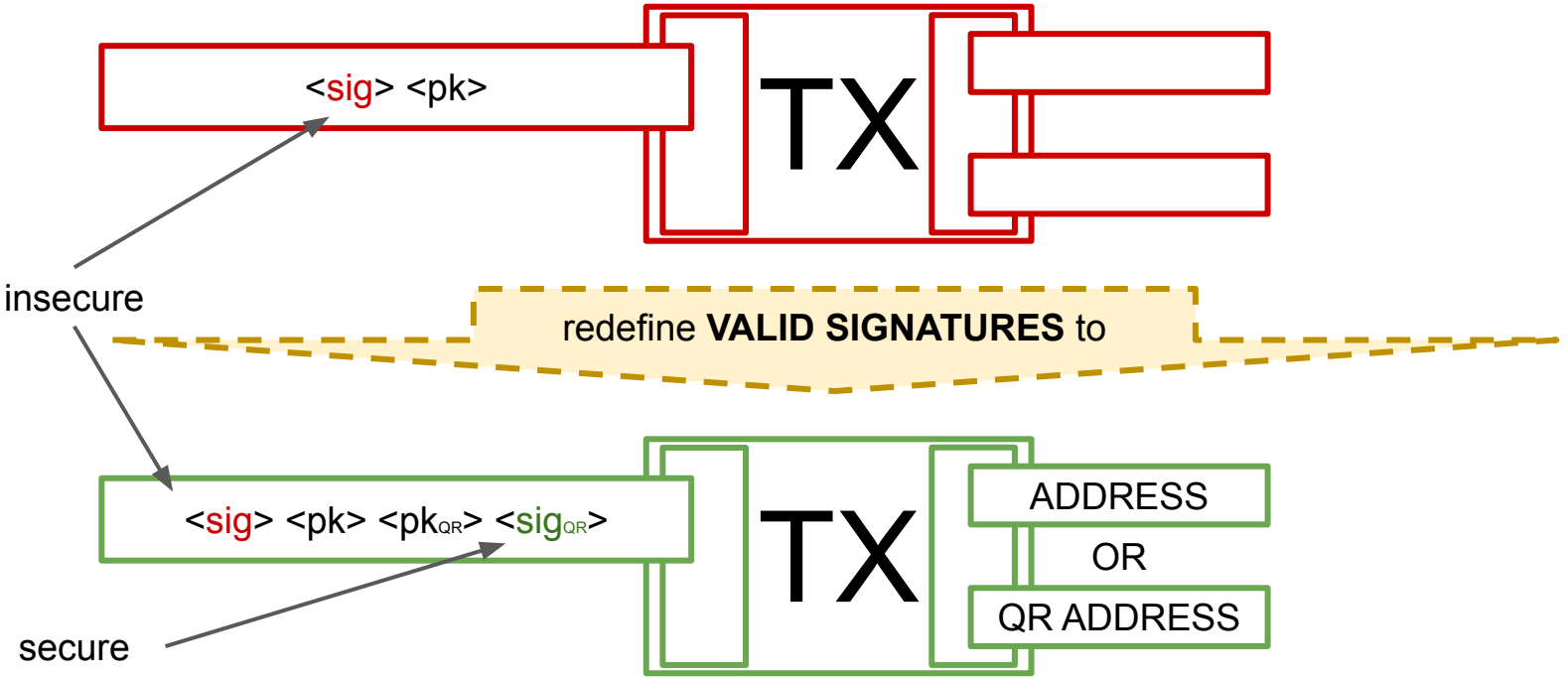
Solution – Quantum Surrogate



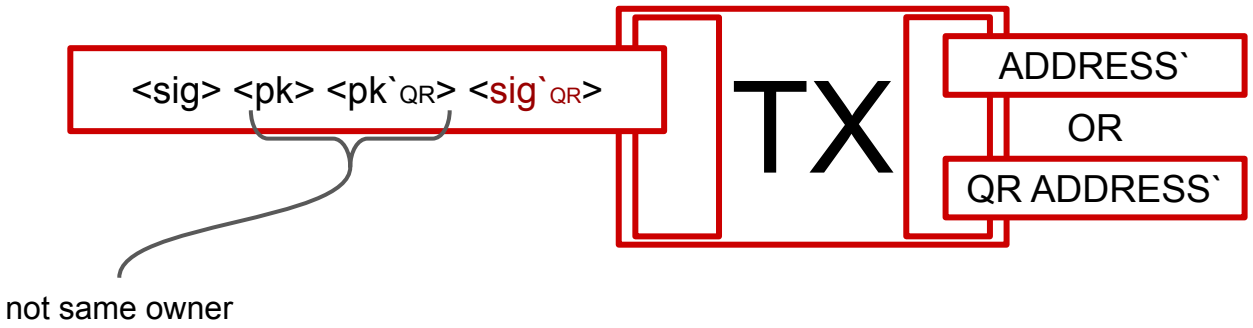
Solution – Quantum Surrogate



Solution – Quantum Surrogate

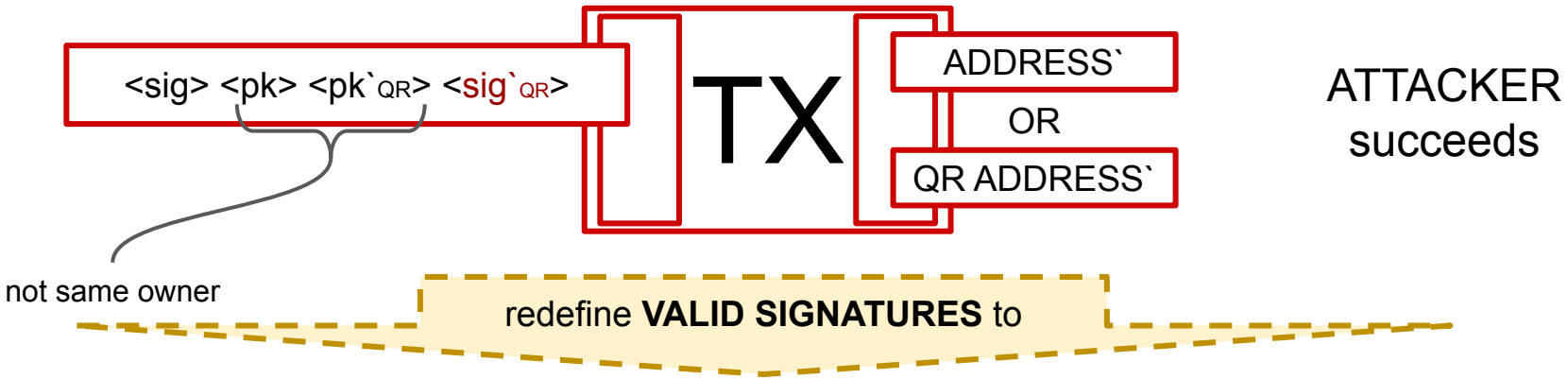


Solution – Quantum Surrogate

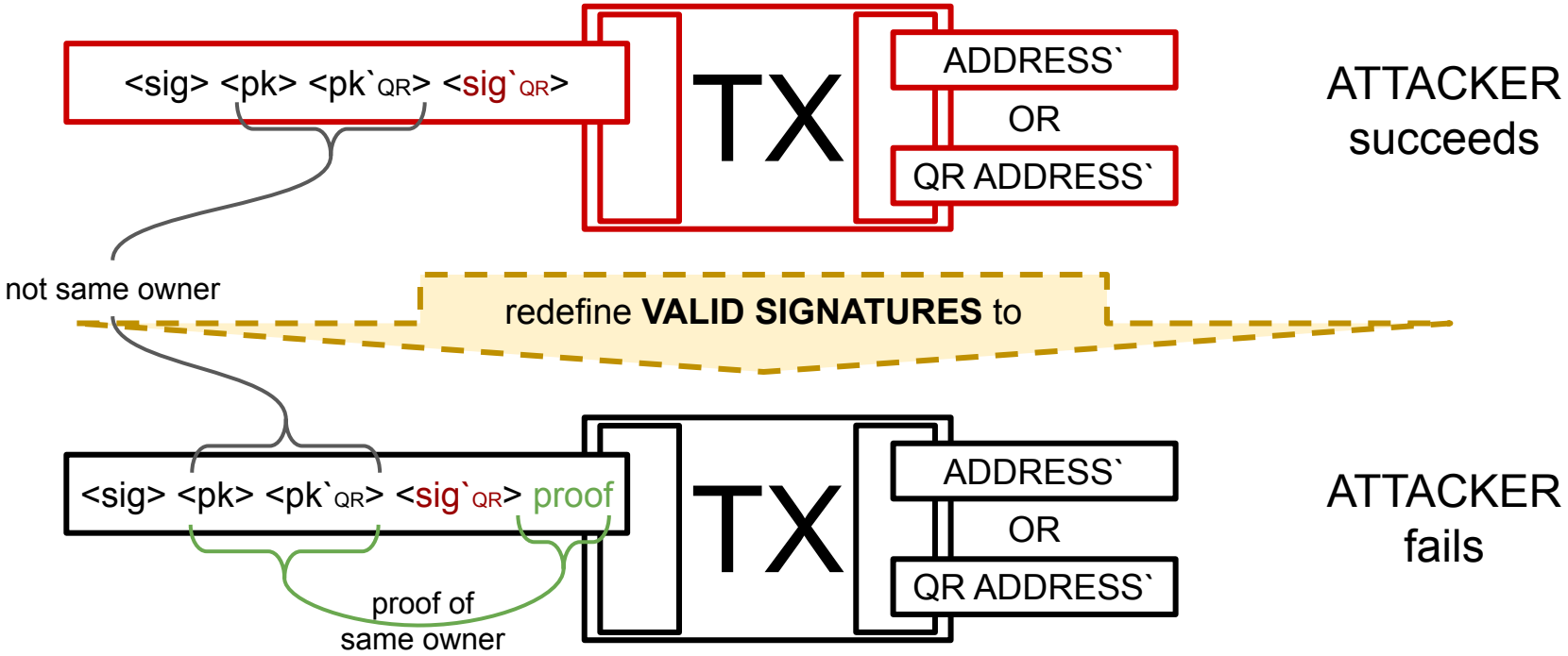


ATTACKER
succeeds

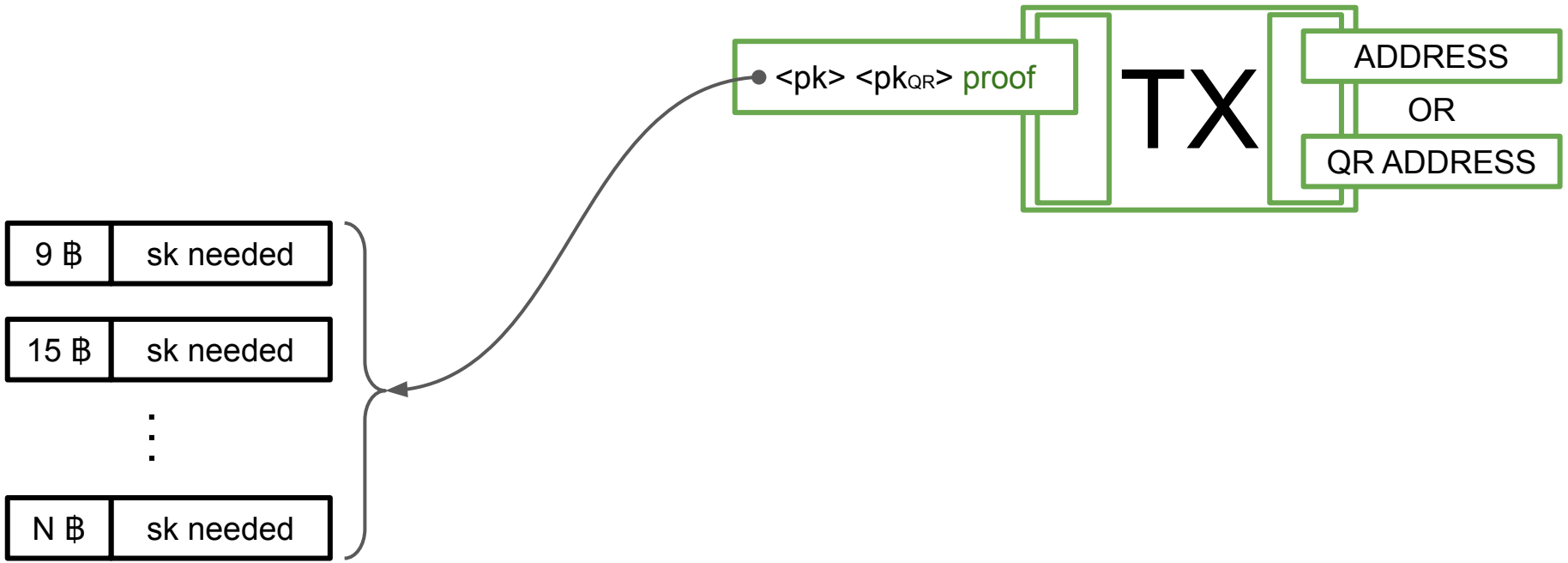
Solution – Quantum Surrogate



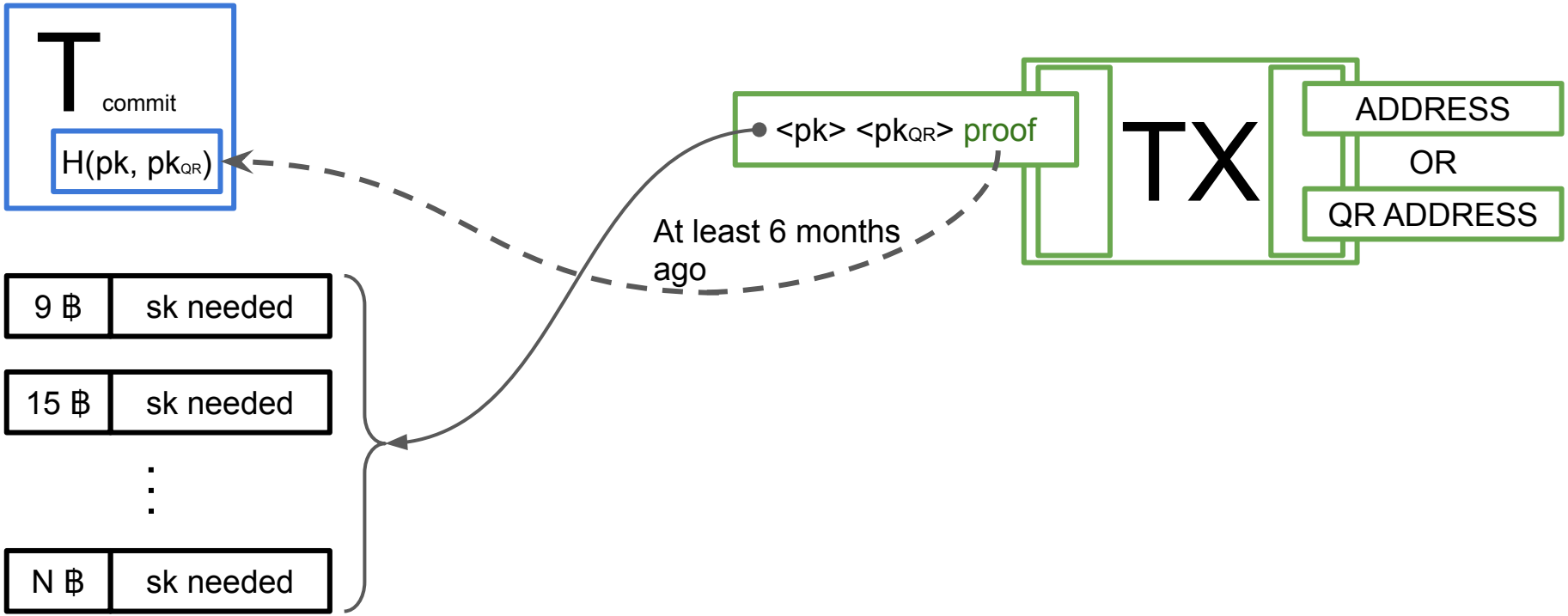
Solution – Quantum Surrogate



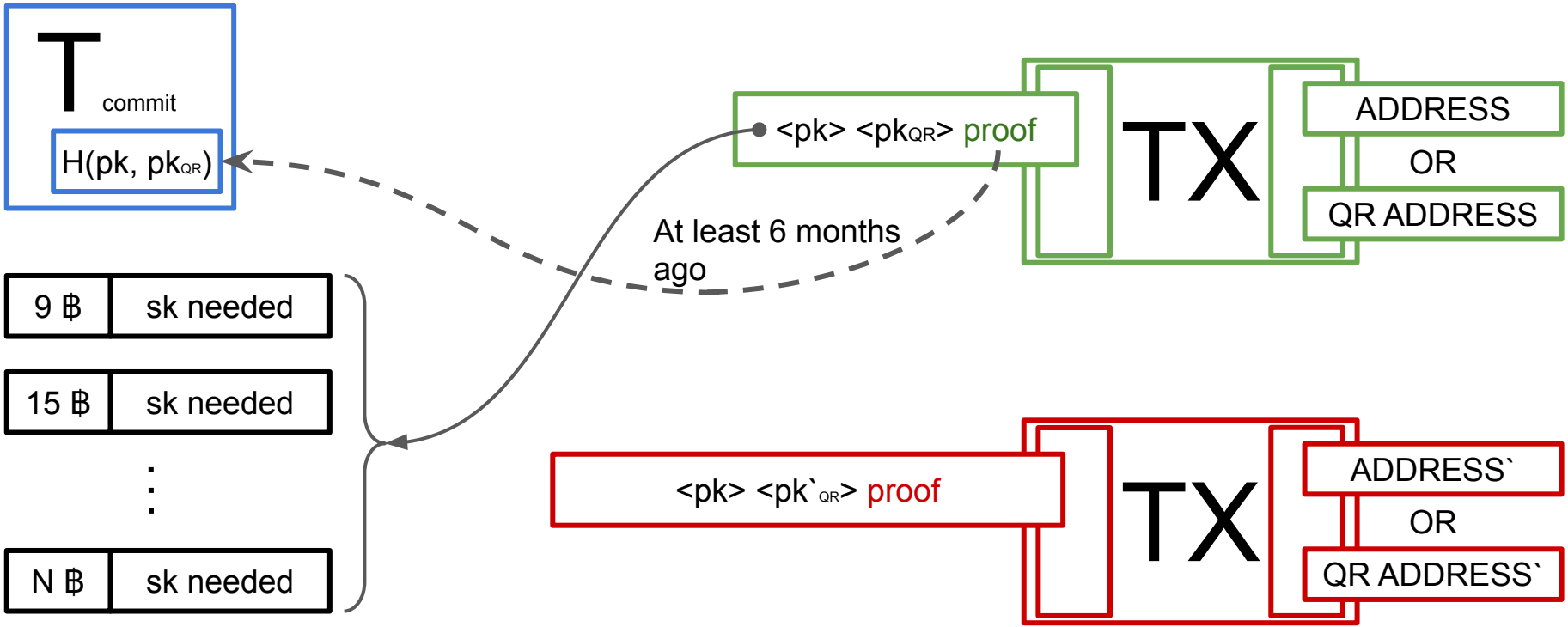
Solution – Proof of Common Ownership



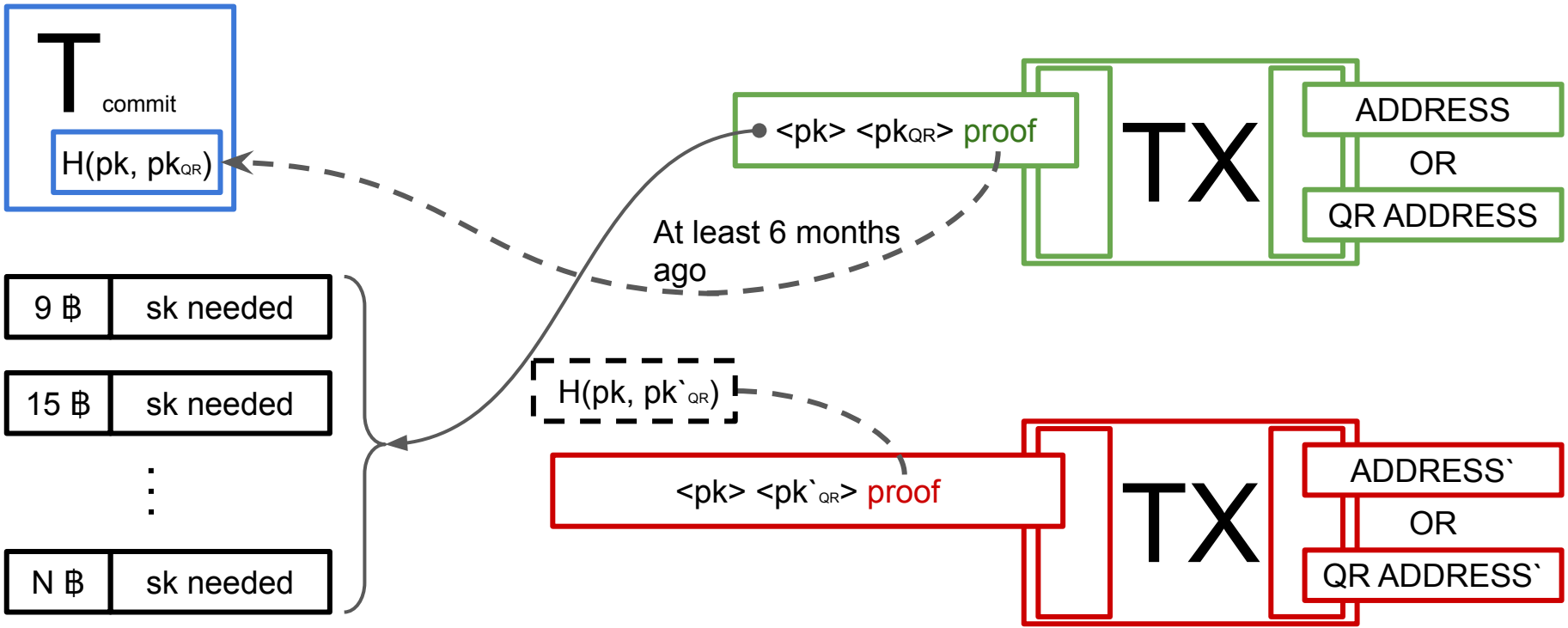
Solution – Proof of Common Ownership with Fixed Delay



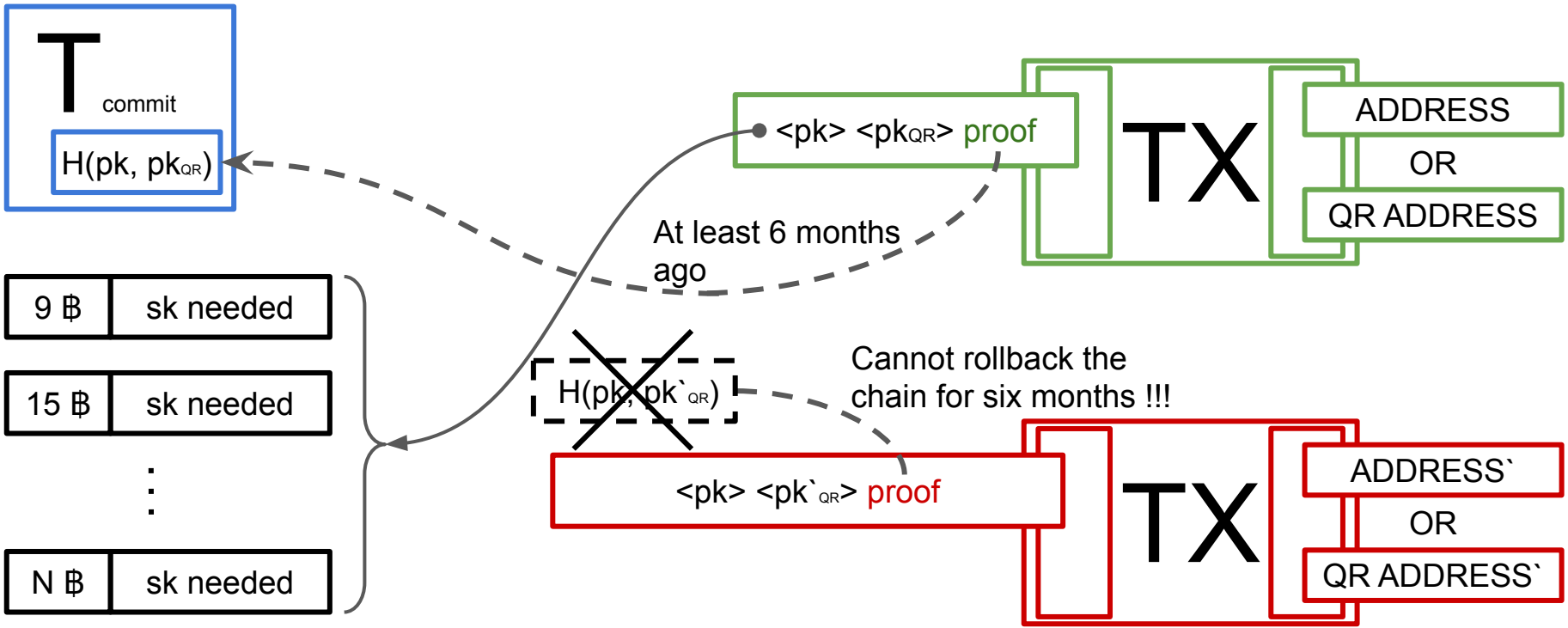
Solution – Proof of Common Ownership with Fixed Delay



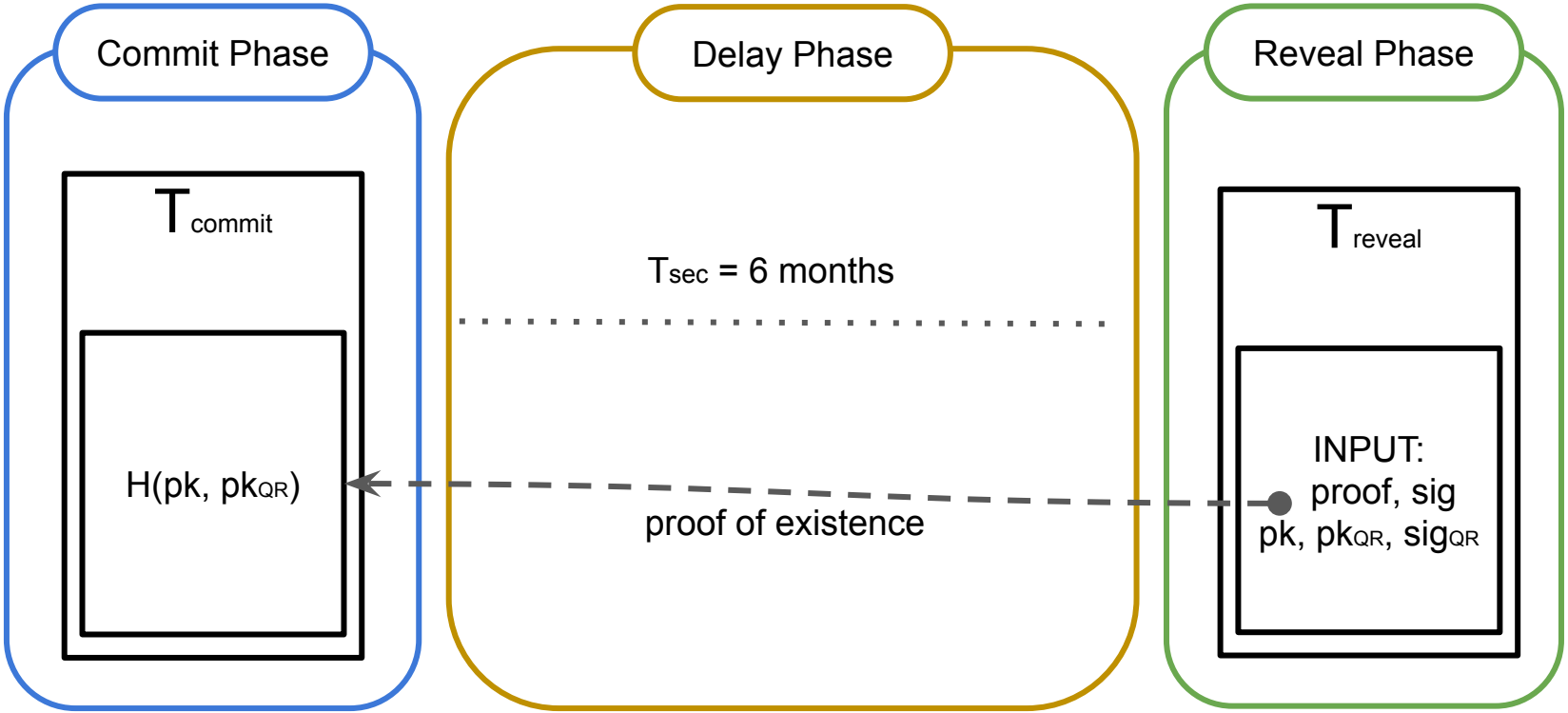
Solution – Proof of Common Ownership with Fixed Delay



Solution – Proof of Common Ownership with Fixed Delay



Solution – Proof of Common Ownership with Fixed Delay



References

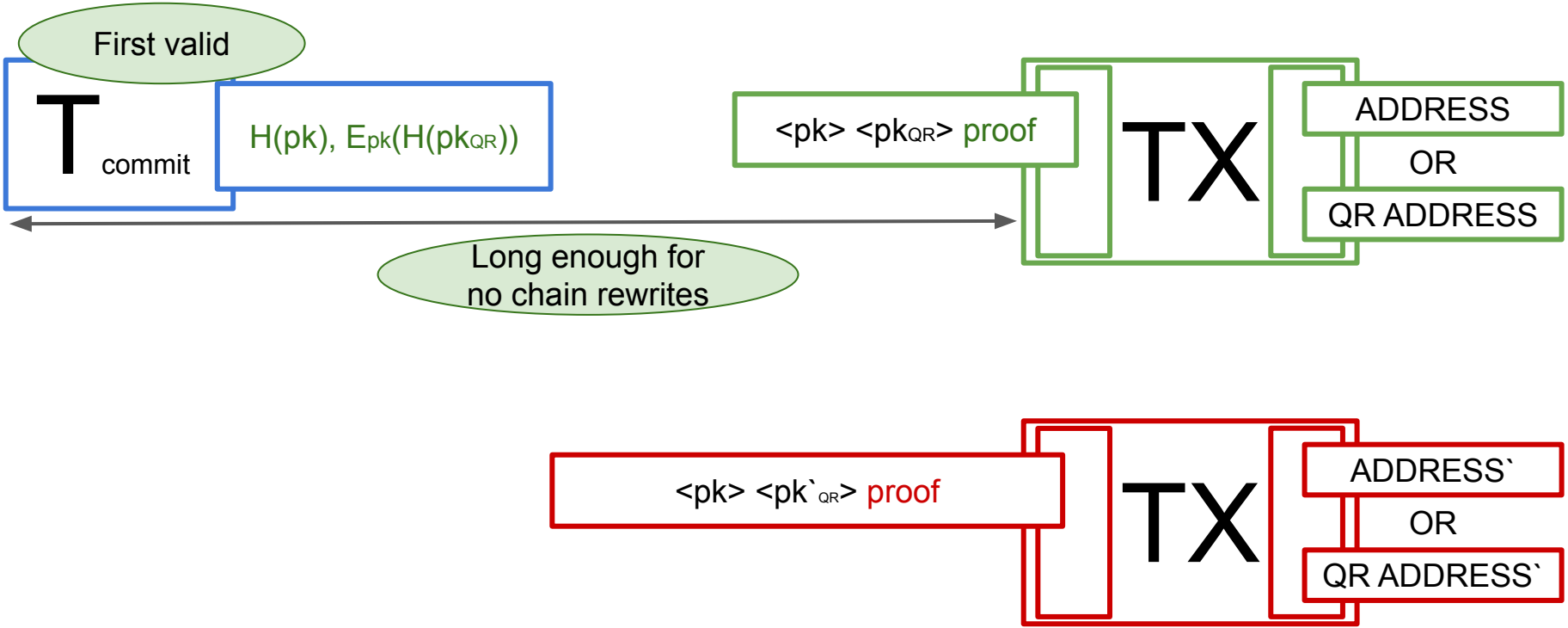
1. I. Stewart, D. Ilie, A. Zamyatin, S. Werner, M. F. Torshizi and W. J. Knottenbelt:
   Committing to quantum resistance: A slow defence for Bitcoin against a fast quantum computing attack
   tiny.cc/qrbtc

2. D. Ilie, W. J. Knottenbelt, and I. Stewart:
   Committing to Quantum Resistance, Better: A Speed-and-Risk-Configurable Defence for Bitcoin against a Fast Quantum Computer Attack
   tiny.cc/betterqrbtc

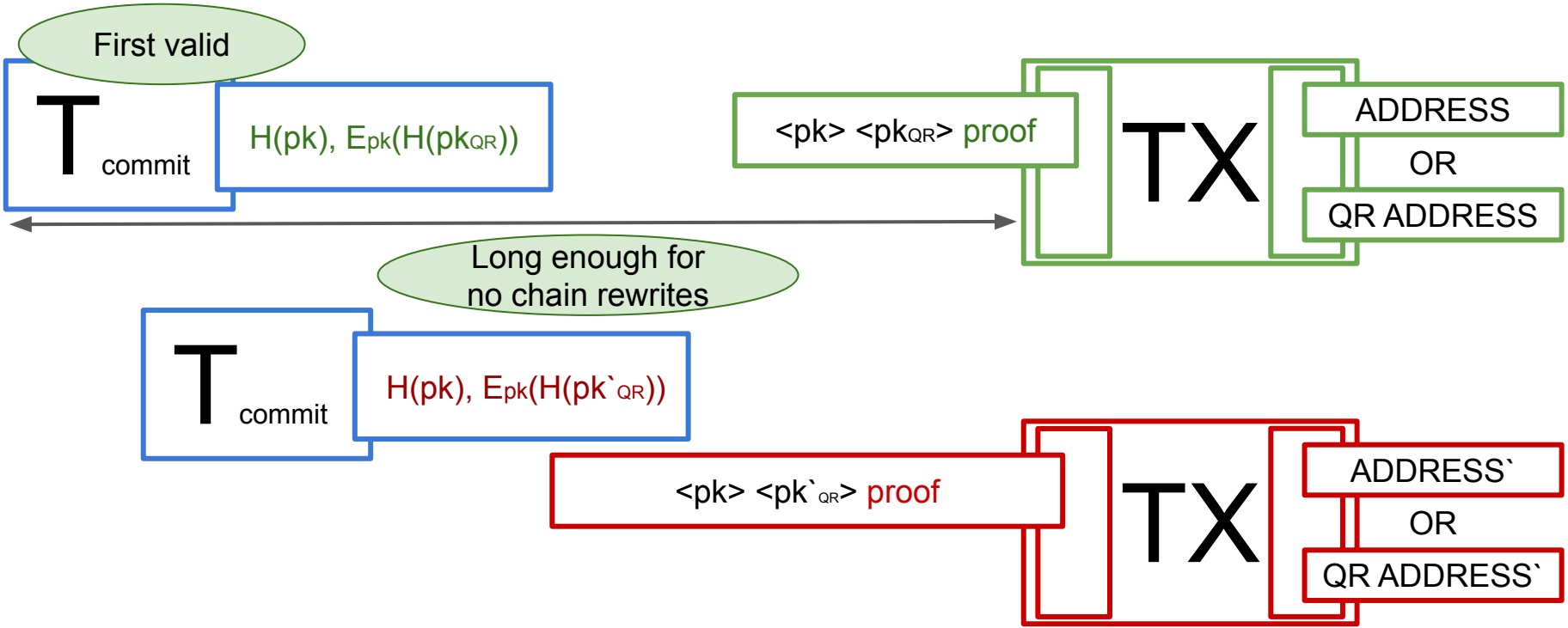
3. Post-Quantum Cryptography: pqcrypto.org or pqcrypto.eu.org

Solution – Proof of Common Ownership with First Valid Commitment

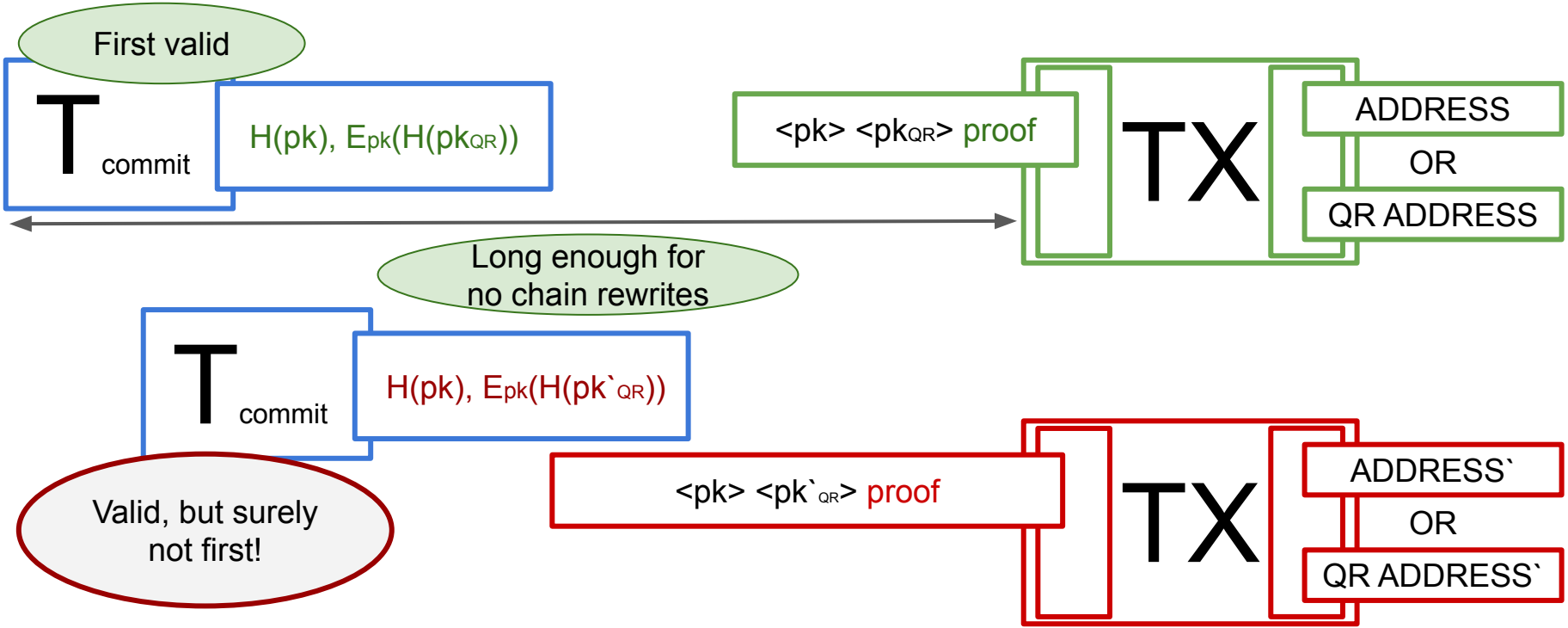
Solution – Proof of Common Ownership with First Valid Commitment



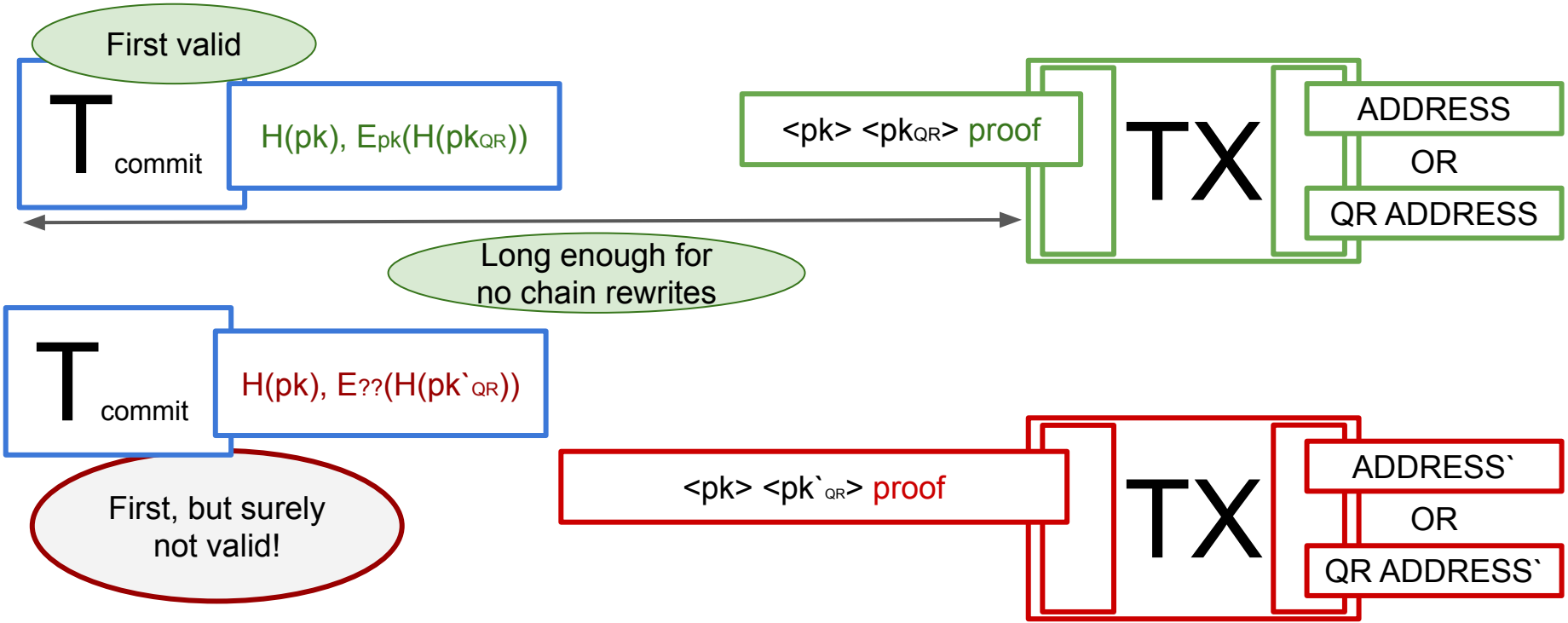
Solution – Proof of Common Ownership with First Valid Commitment



Solution – Proof of Common Ownership with First Valid Commitment

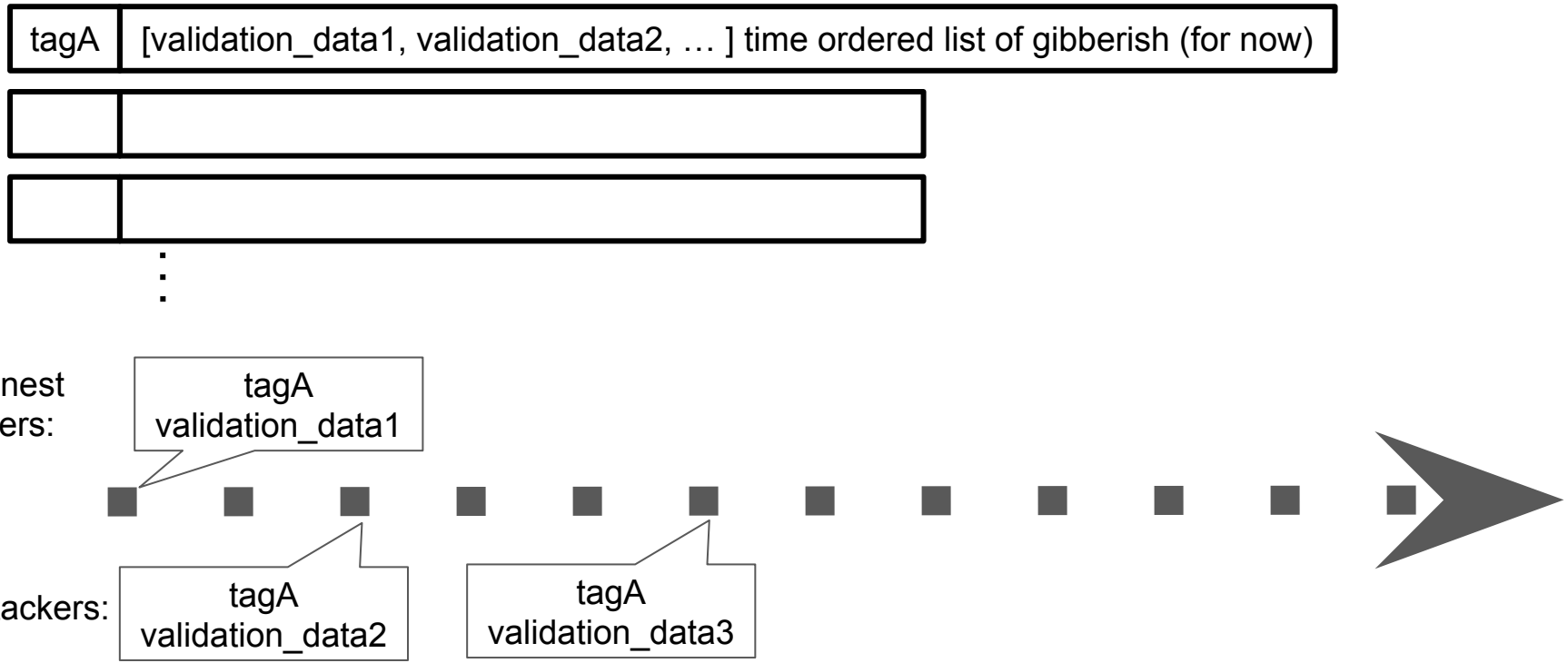


Solution – Proof of Common Ownership with First Valid Commitment



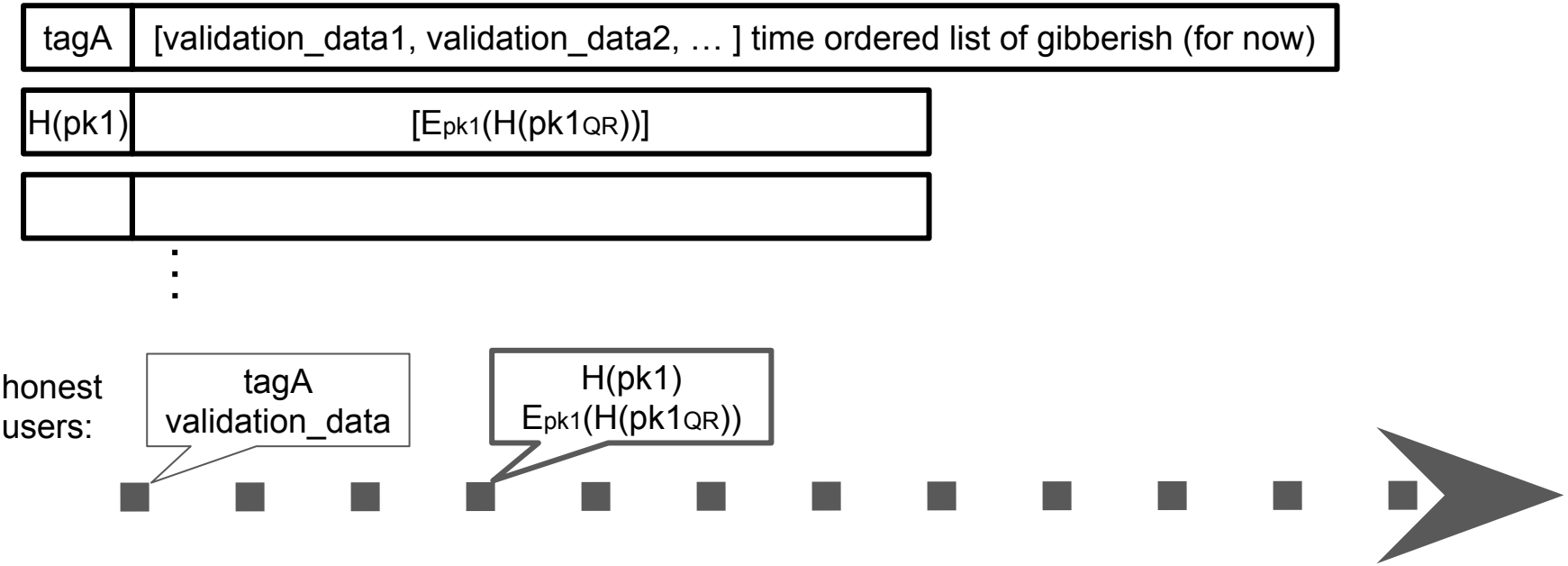
Solution – First Valid Commitment Index

Miner's point of view



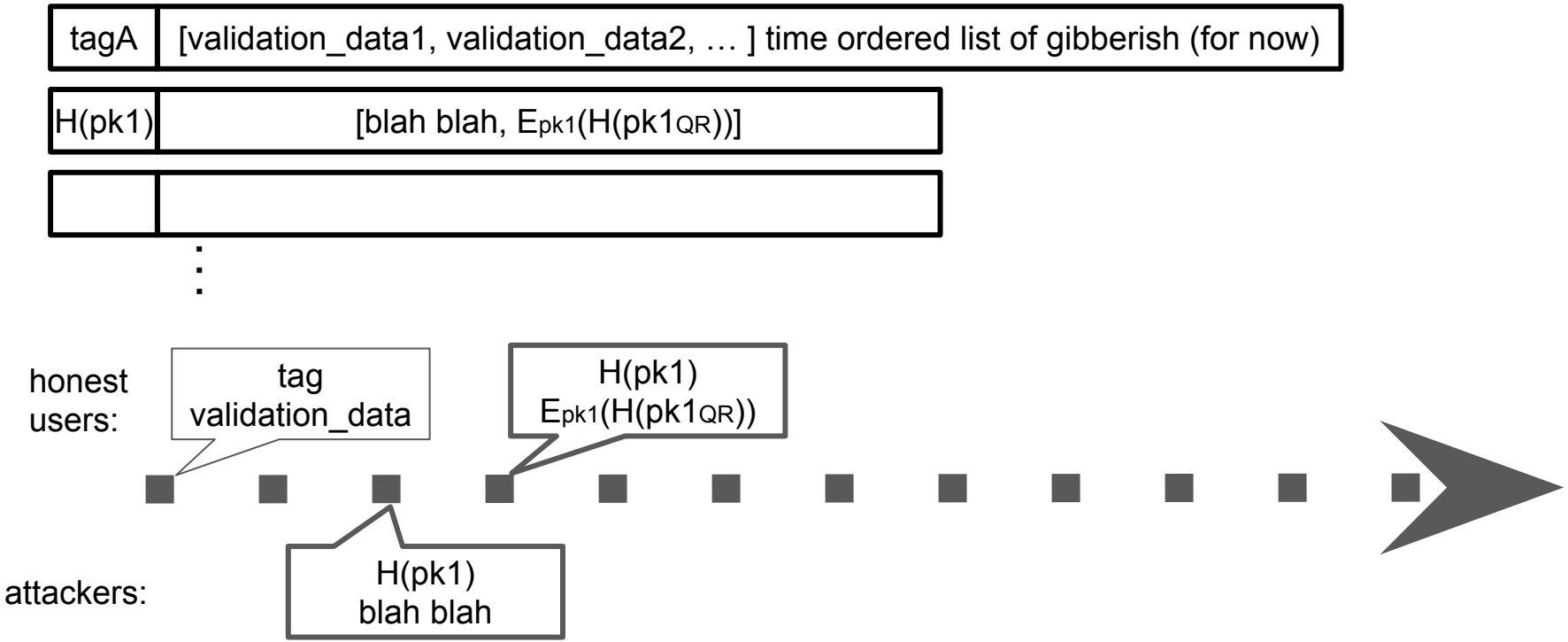
Solution – First Valid Commitment Index

Miner's point of view



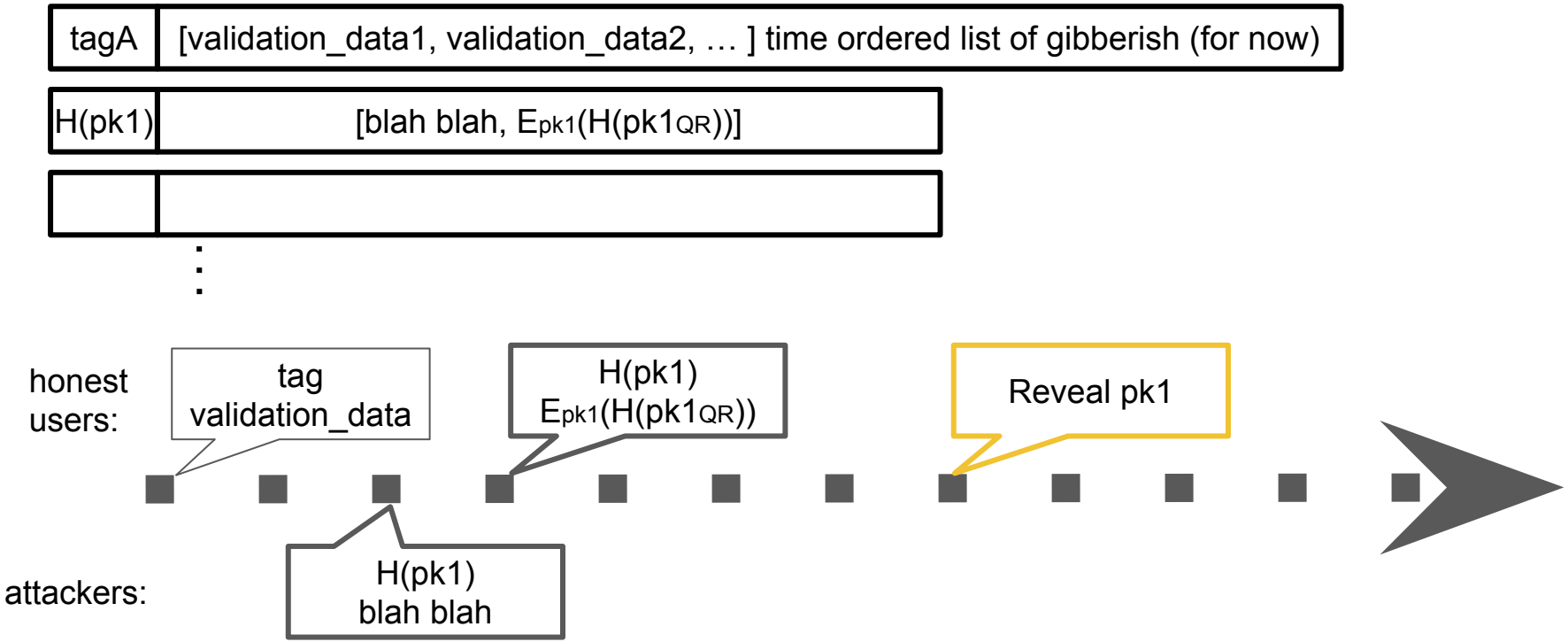
Solution – First Valid Commitment Index

Miner's point of view



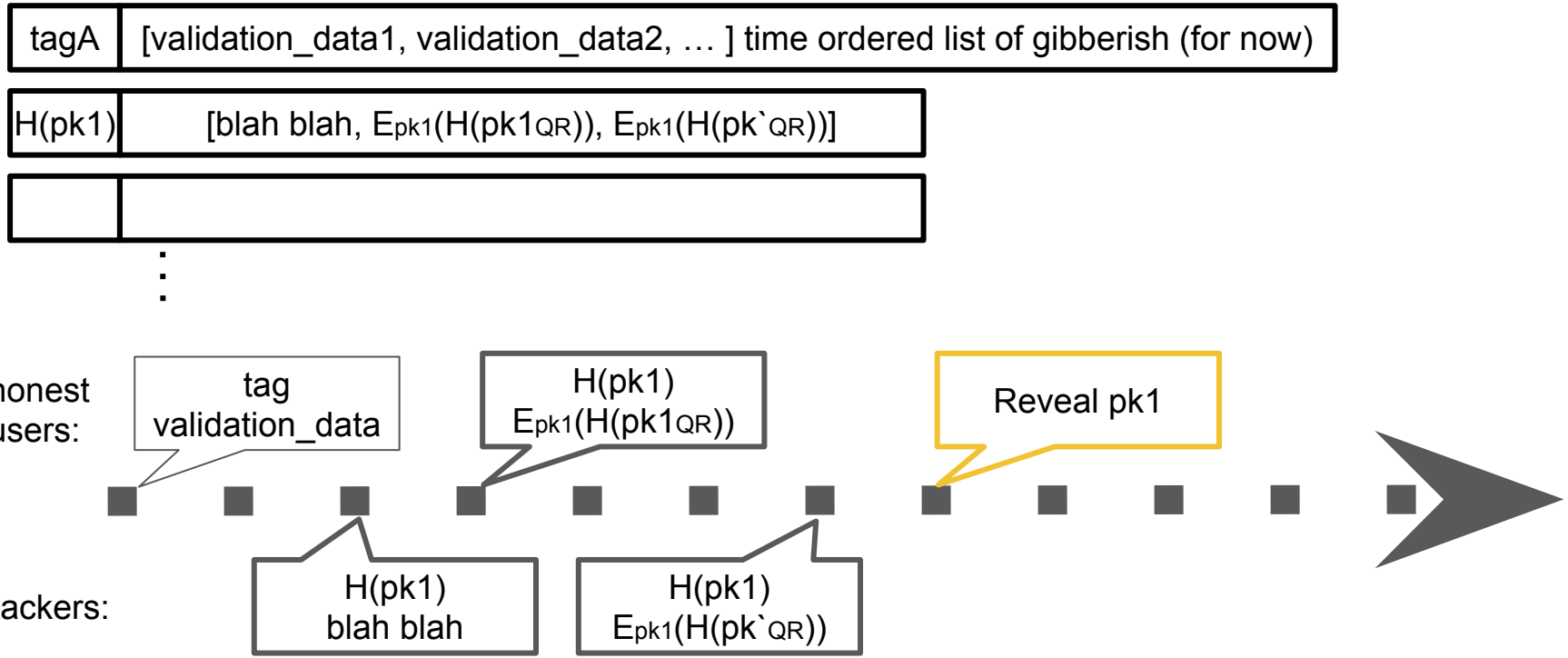
Solution – First Valid Commitment Index

Miner's point of view



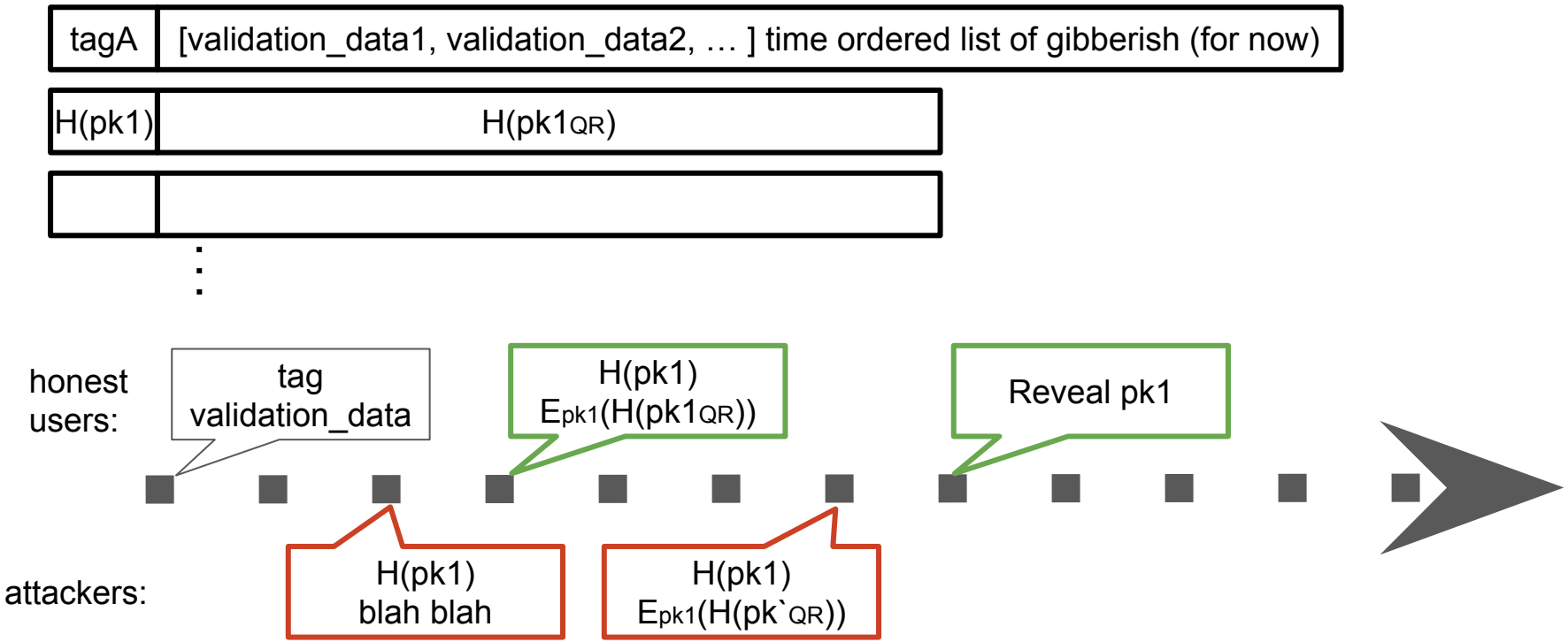
Solution – First Valid Commitment Index

Miner's point of view



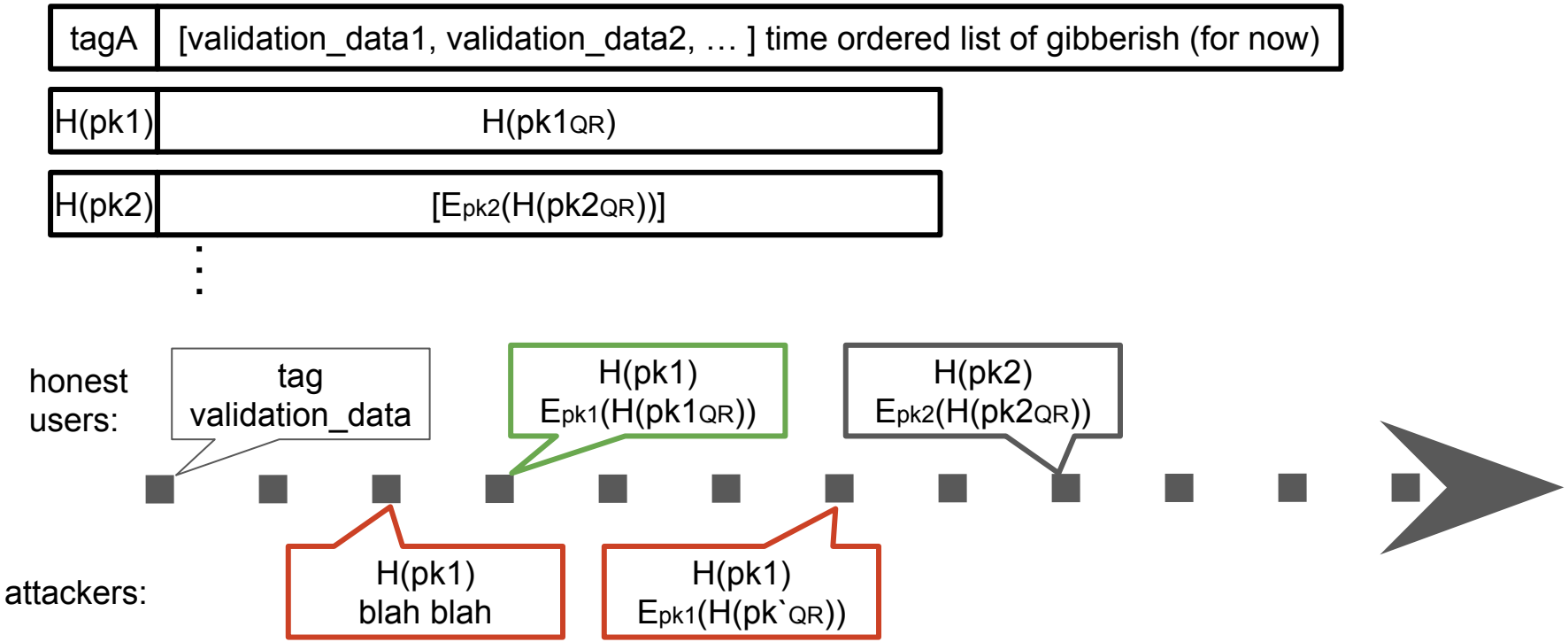
Solution – First Valid Commitment Index

Miner's point of view



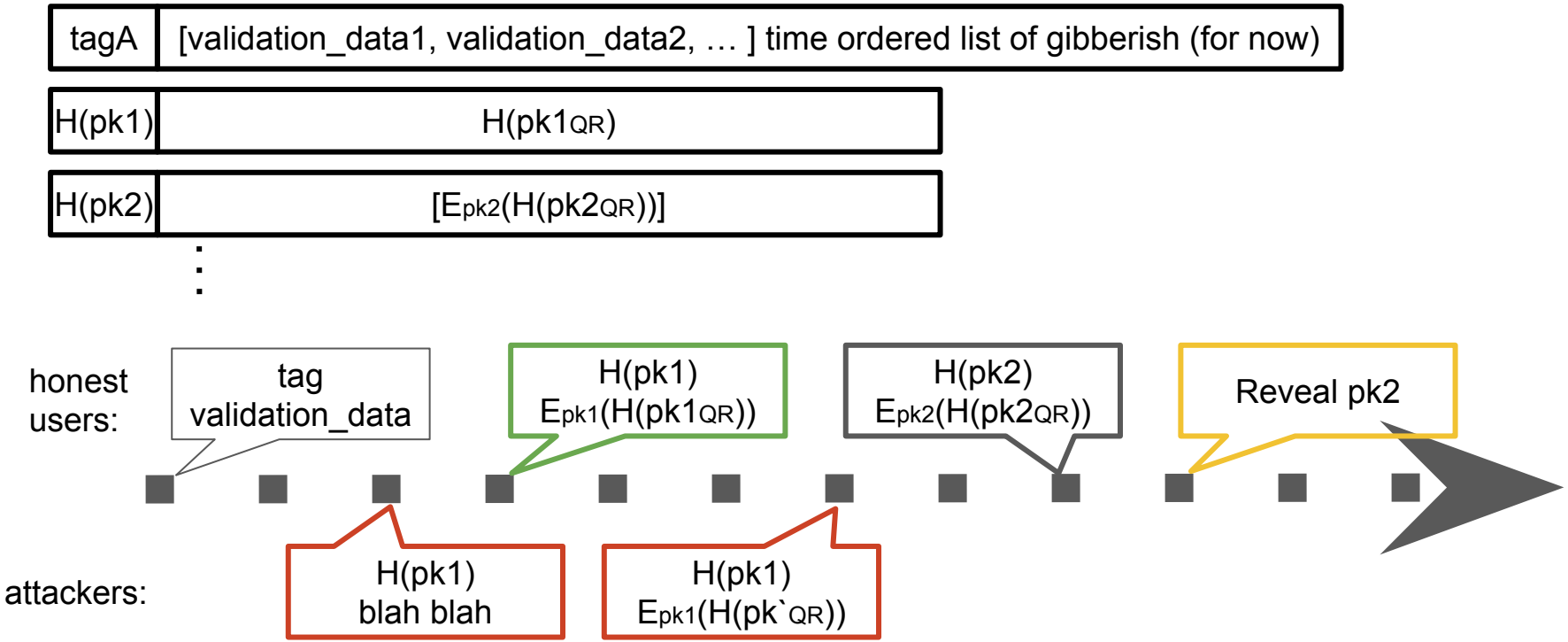
Solution – First Valid Commitment Index

Miner's point of view



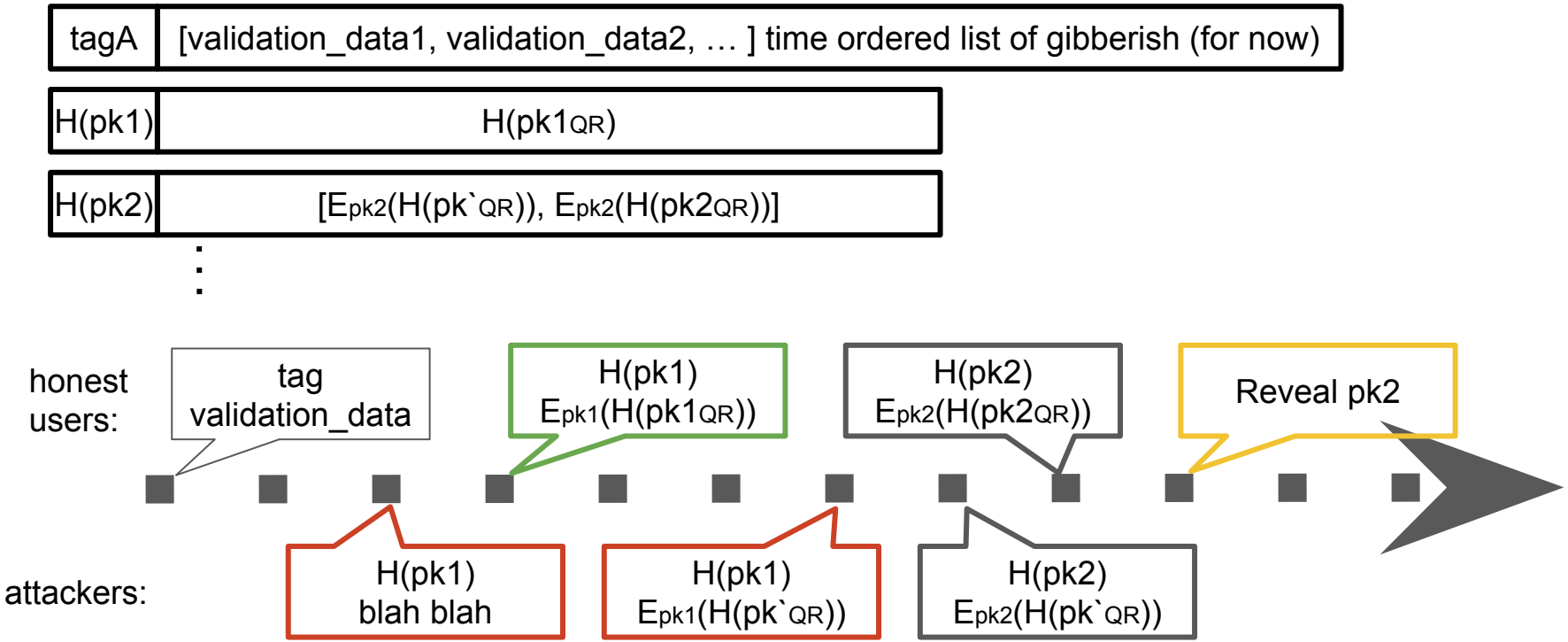
Solution – First Valid Commitment Index

Miner's point of view



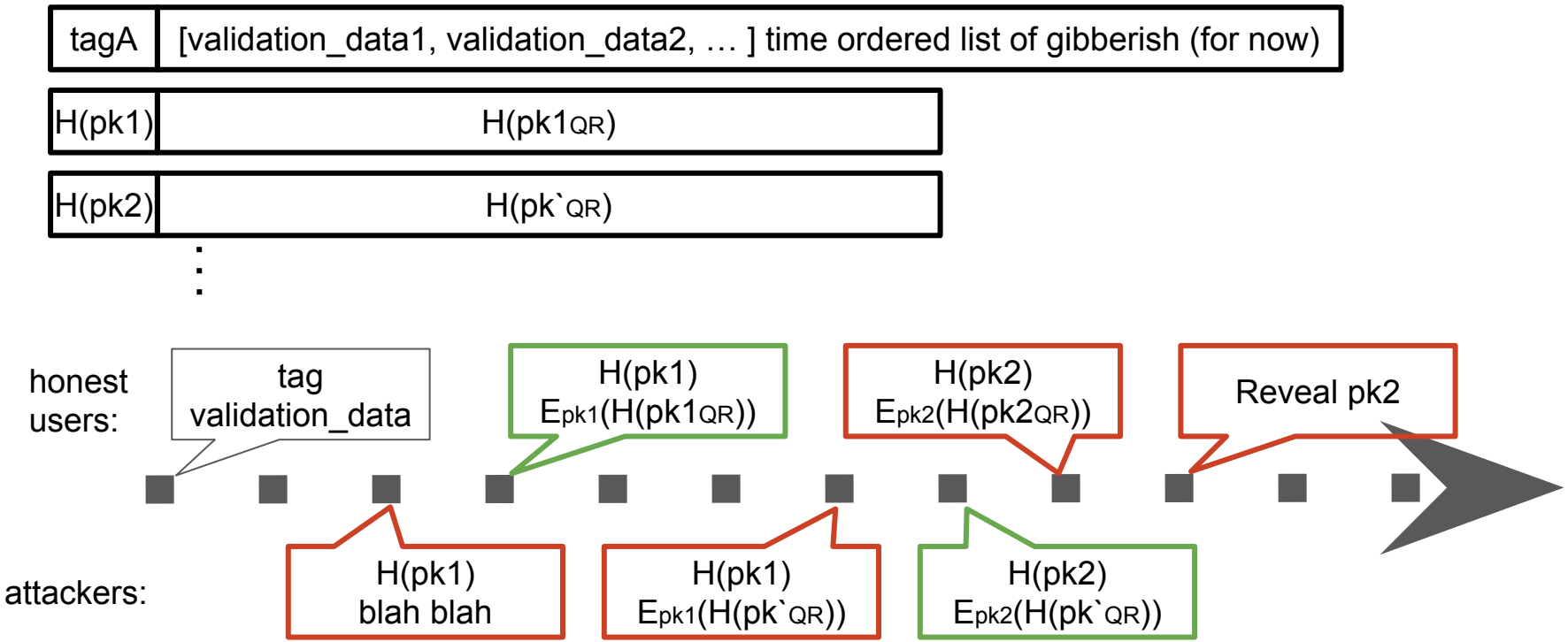
Solution – First Valid Commitment Index

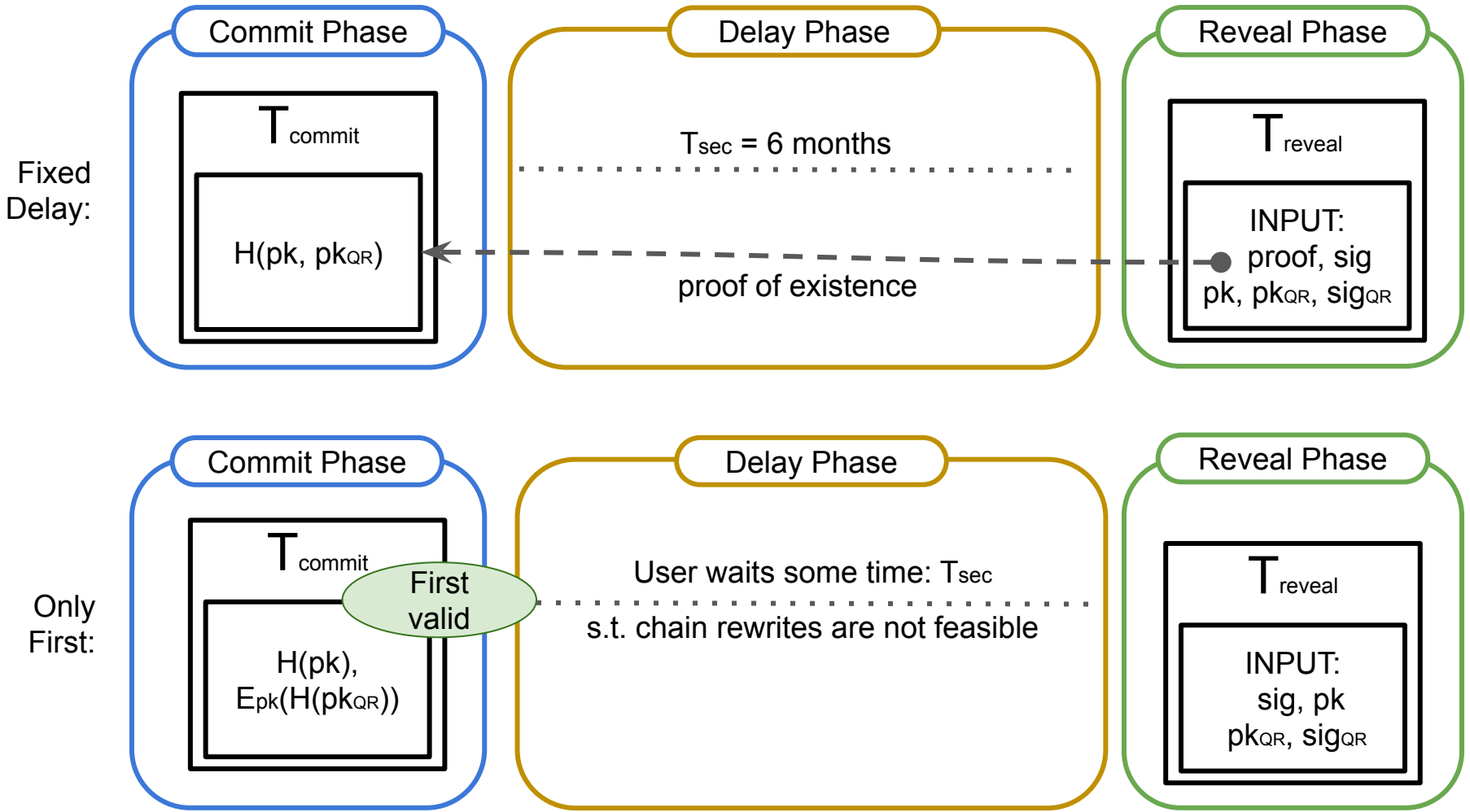
Miner's point of view



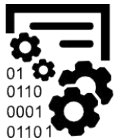
Solution – First Valid Commitment Index

Miner's point of view




Fixed Delay:
Commit Phase
T commit
H(pk, pkQR)
Delay Phase
Tsec = 6 months
proof of existence
Reveal Phase
T reveal
INPUT:
proof, sig
pk, pkQR, sigQR
Only First:
Commit Phase
T commit
First valid
H(pk),
Epk(H(pkQR))
Delay Phase
User waits some time: Tsec
s.t. chain rewrites are not feasible
Reveal Phase
T reveal
INPUT:
sig, pk
pkQR, sigQR

Hashes

Arbitrarily long data 

Hash Function →  Fixed size small “ID” (fingerprint)

SHA256, RIPEMD160, etc.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequa.

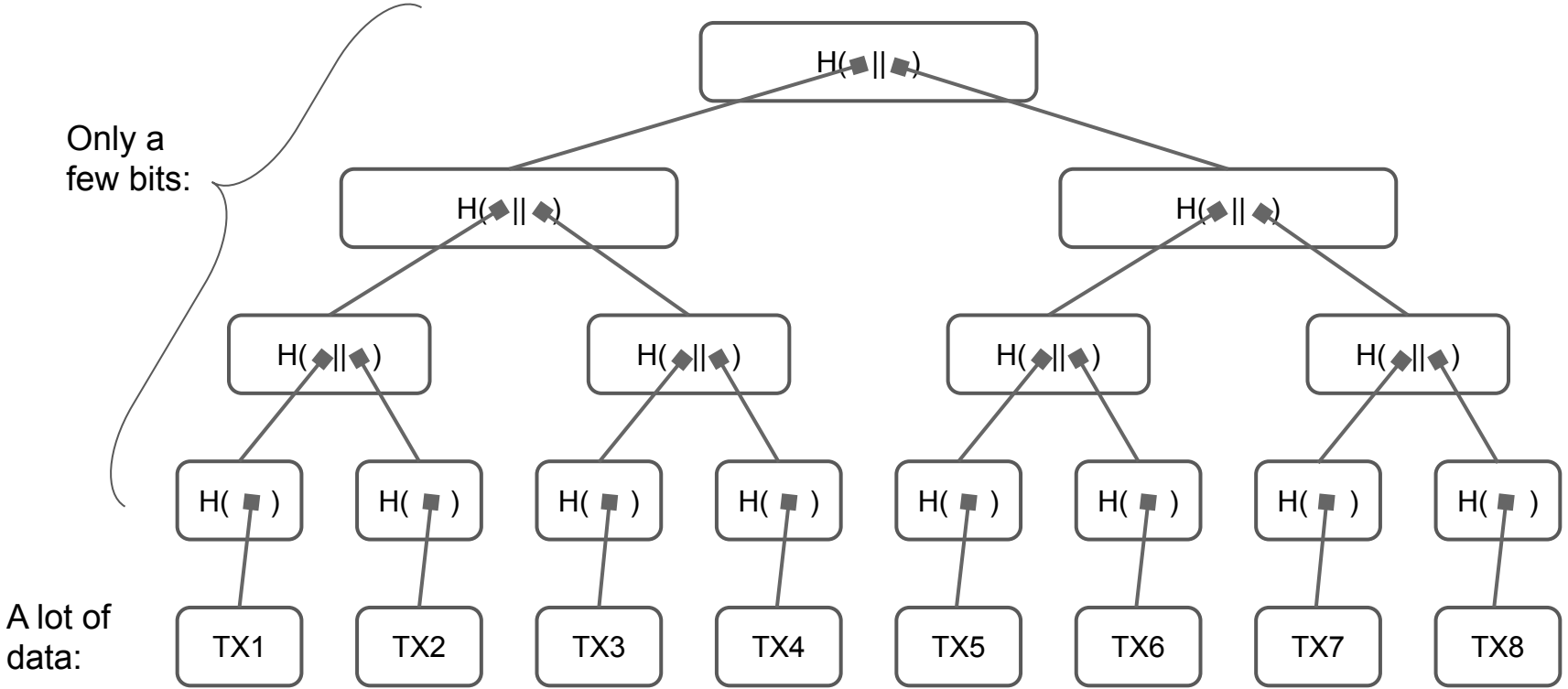
Hash Function → ag3k4mrk

- Deterministic
- Pre-image resistance: Given **y**, cannot find **x**, s.t. **H(x) = y**
- 2nd pre-image resistance: Given **x**, cannot find **z != x**, s.t. **H(z) = H(x)**
- Collision resistance: Cannot find **x,z** with **z != x**, s.t. **H(z) = H(x)**

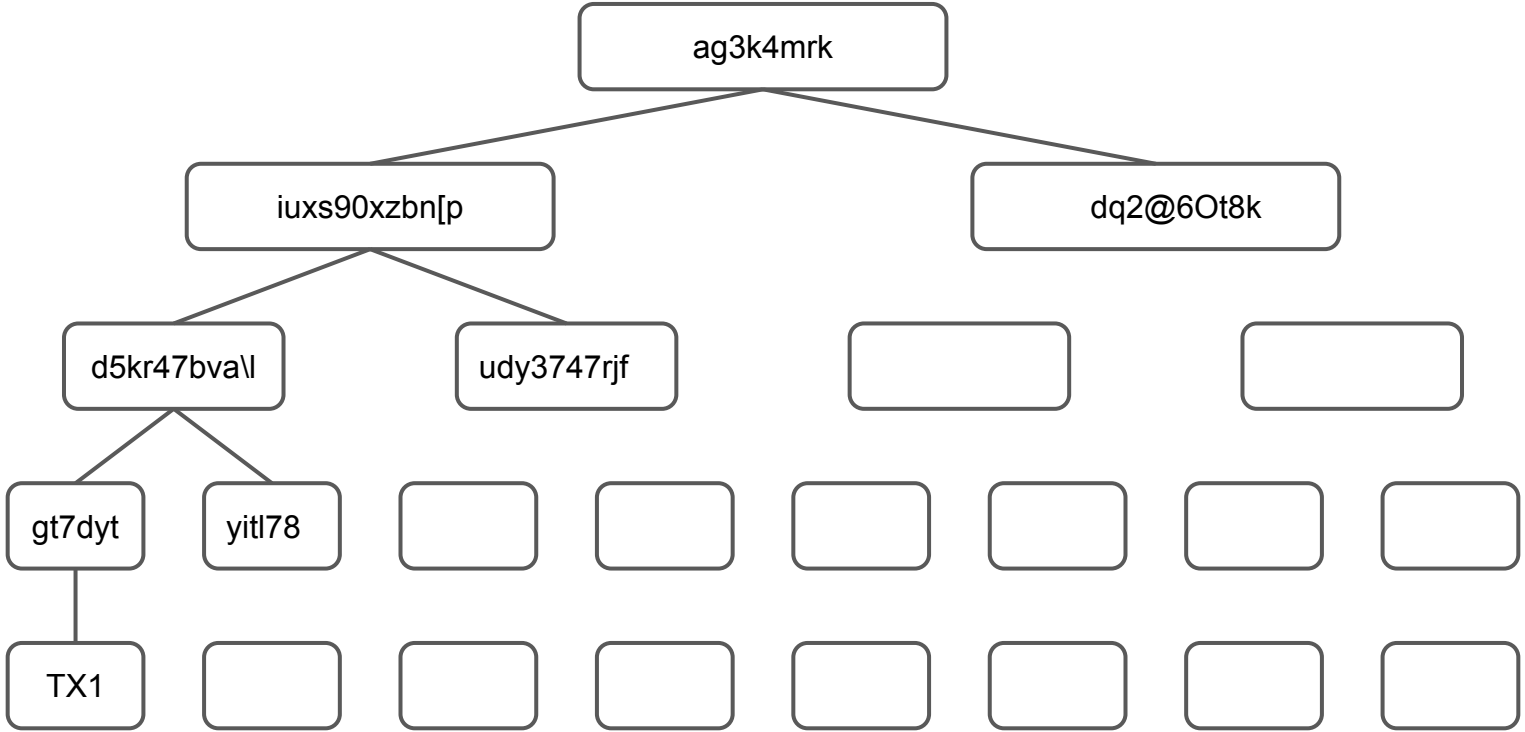
Hashes – Proof of Work

- Find a nonce such that H(block_header || nonce) < threshold
- 2^{256} possible H outputs – that's a 78 digit number
- threshold = 6,379,265,451,411 – only 13 digit number
- The best chance is to just randomly try loads of values for nonce

Hashes – Merkle Trees



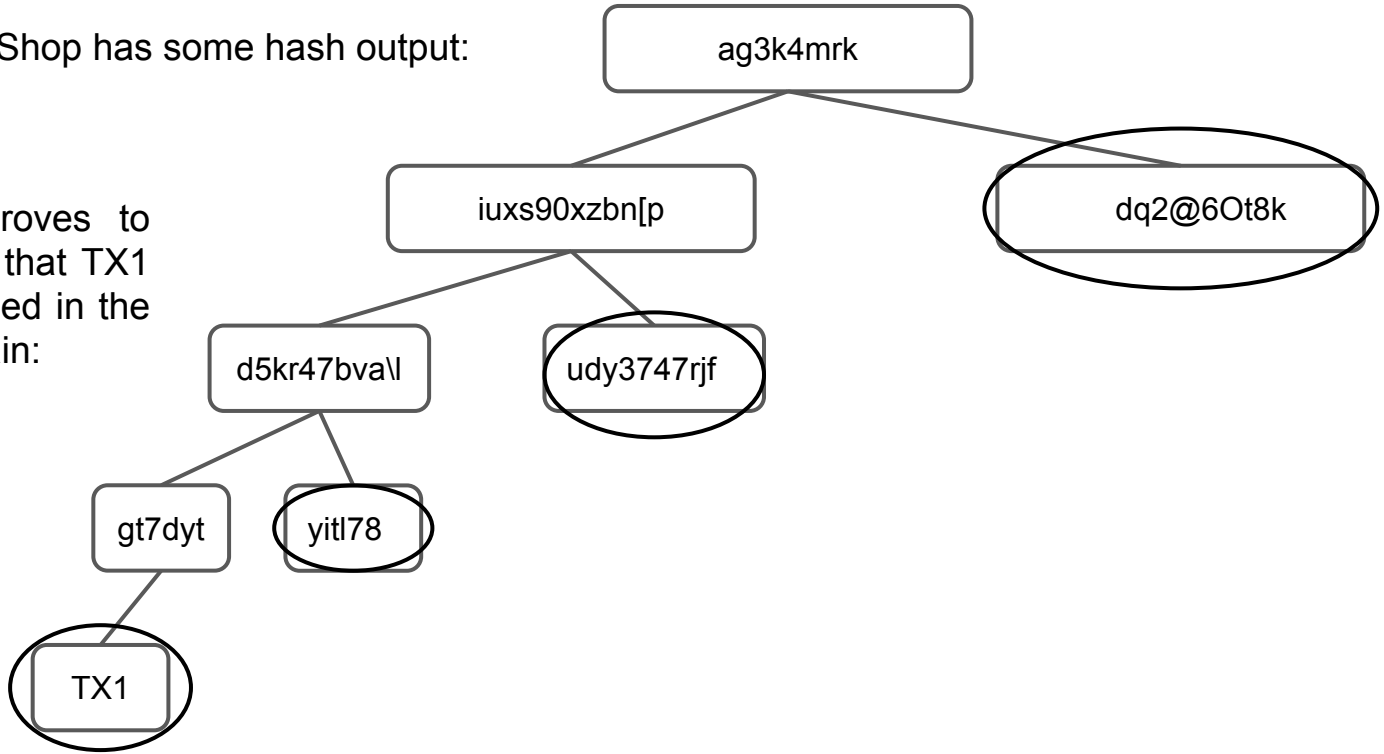
Hashes – Proof of existence



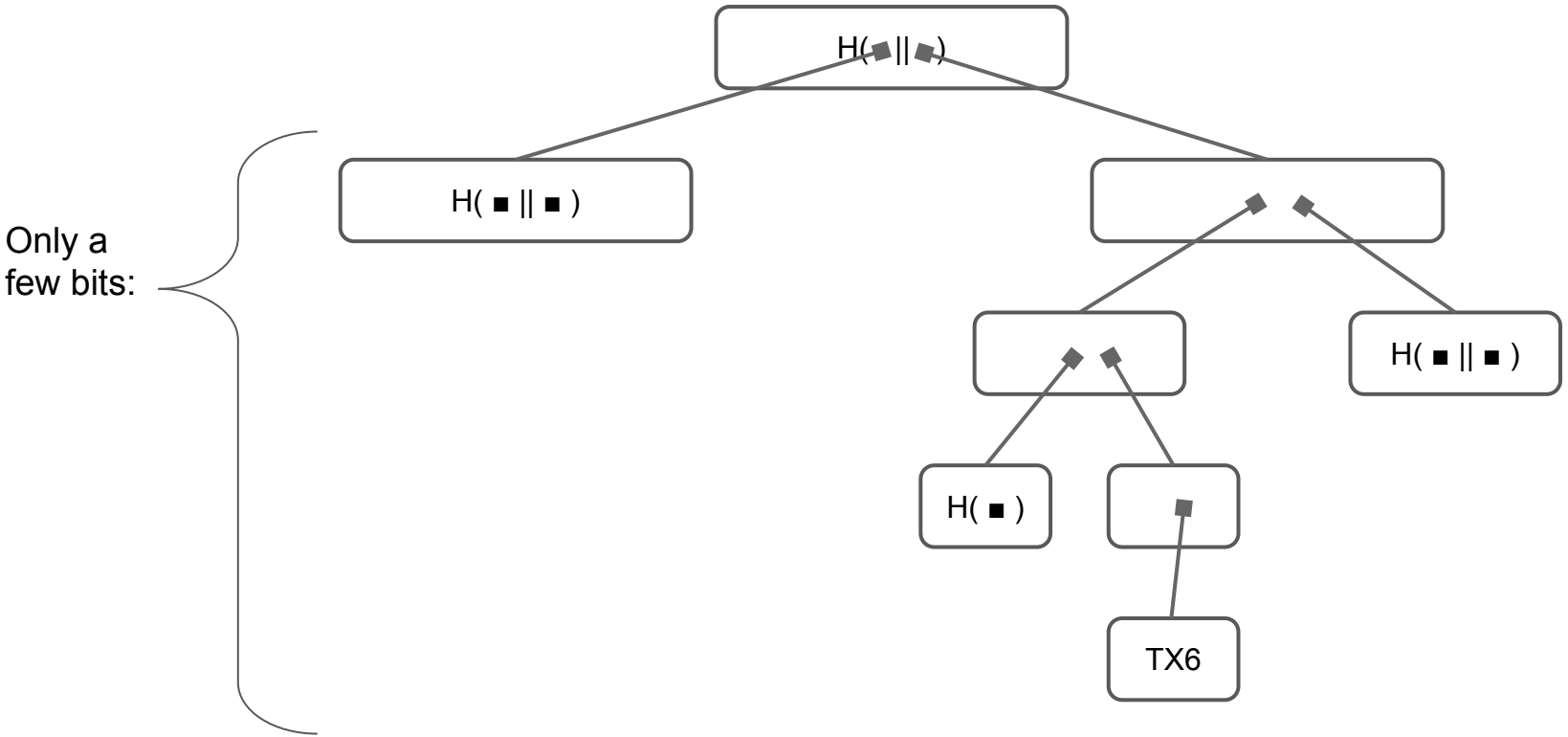
Hashes – Proof of existence

Shop has some hash output:

Buyer proves to shopper that TX1 is recorded in the blockchain:

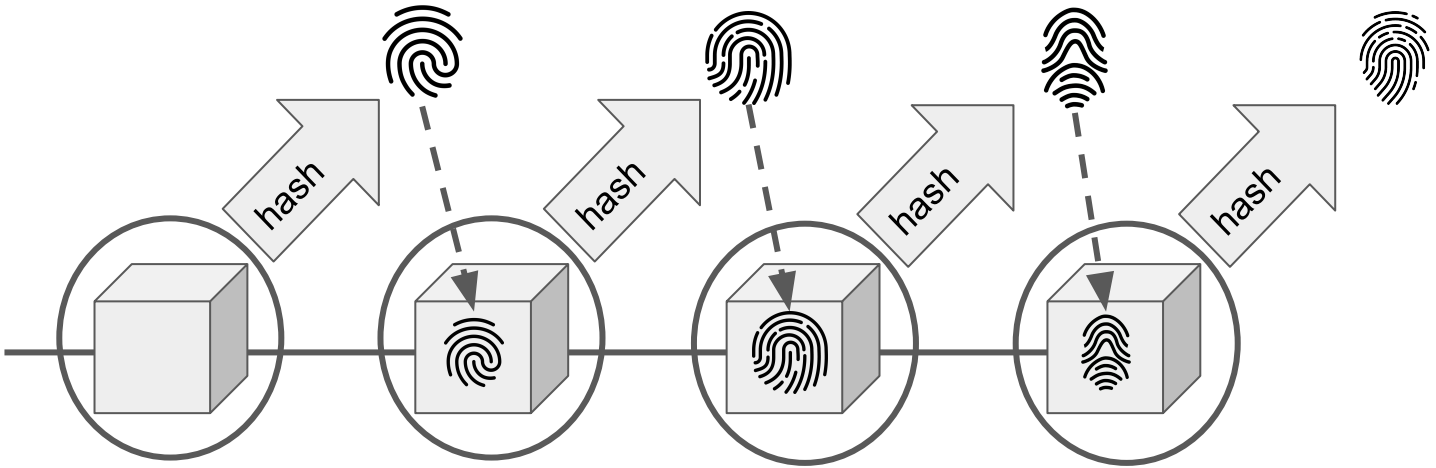


Hashes – Proof of existence



Hashes – Immutability

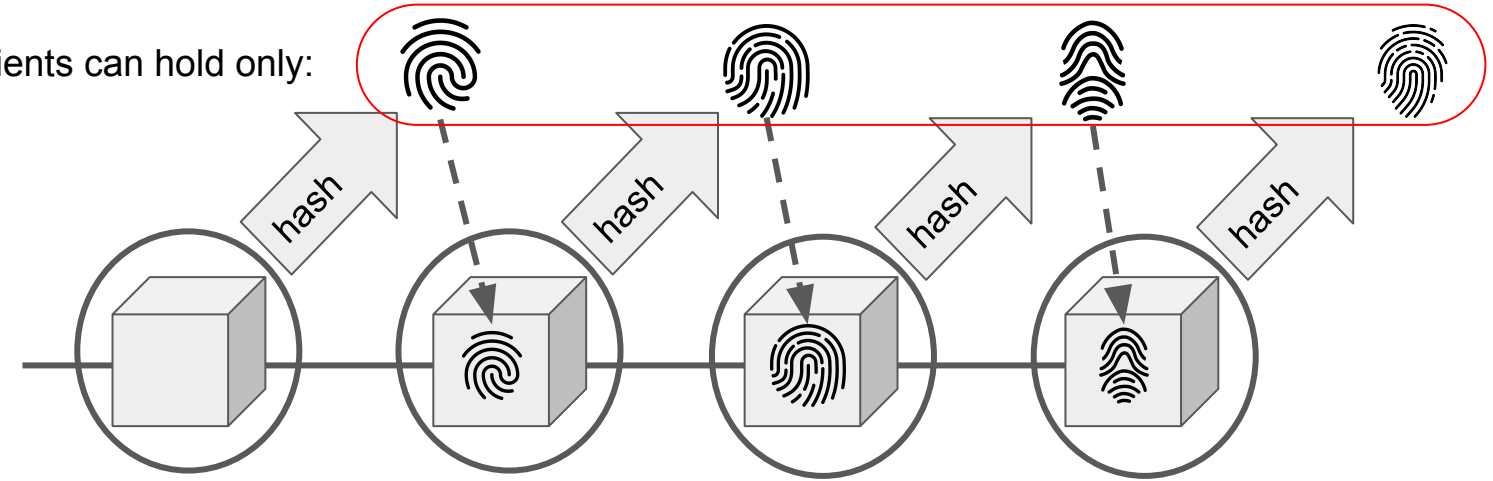
Each block contains a merkle tree of transactions and the hash of the previous block:



Hashes – Immutability

Each block contains a merkle tree of transactions and the hash of the previous block:

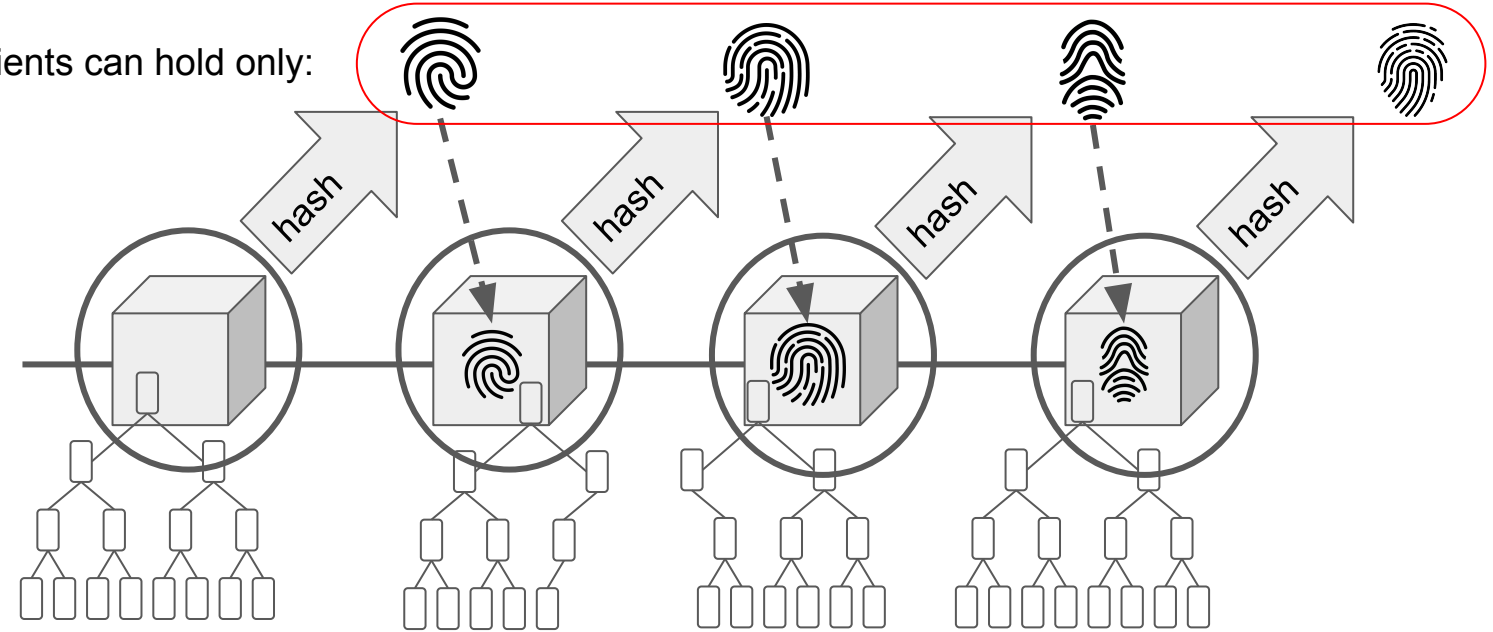
Light-clients can hold only:



Hashes – Immutability

Each block contains a merkle tree of transactions and the hash of the previous block:

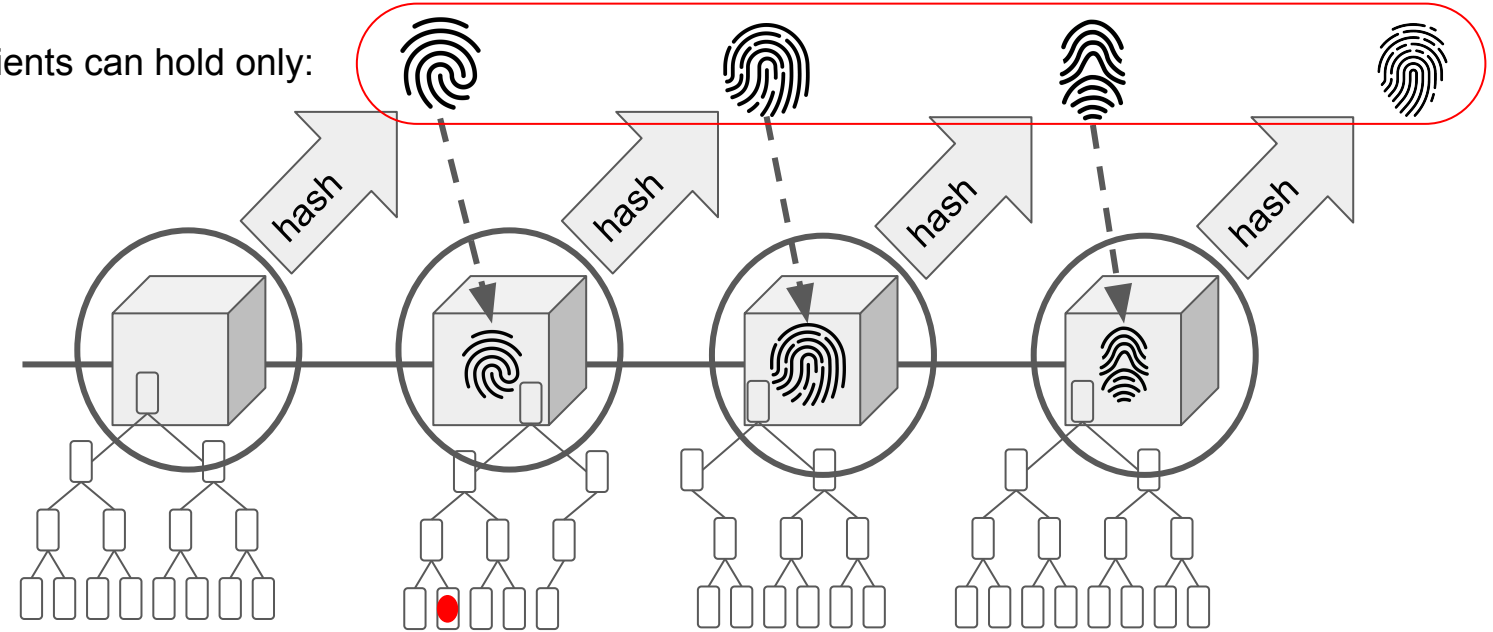
Light-clients can hold only:



Hashes – Immutability

Each block contains a merkle tree of transactions and the hash of the previous block:

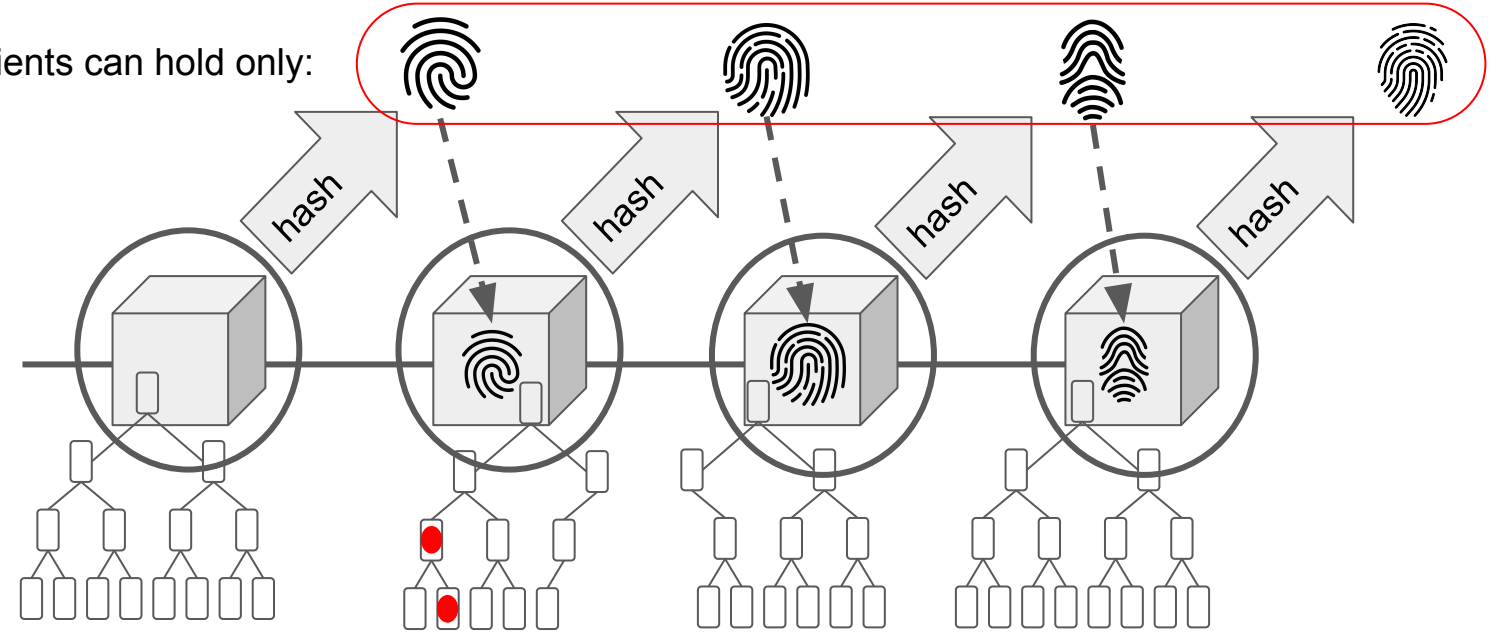
Light-clients can hold only:



Hashes – Immutability

Each block contains a merkle tree of transactions and the hash of the previous block:

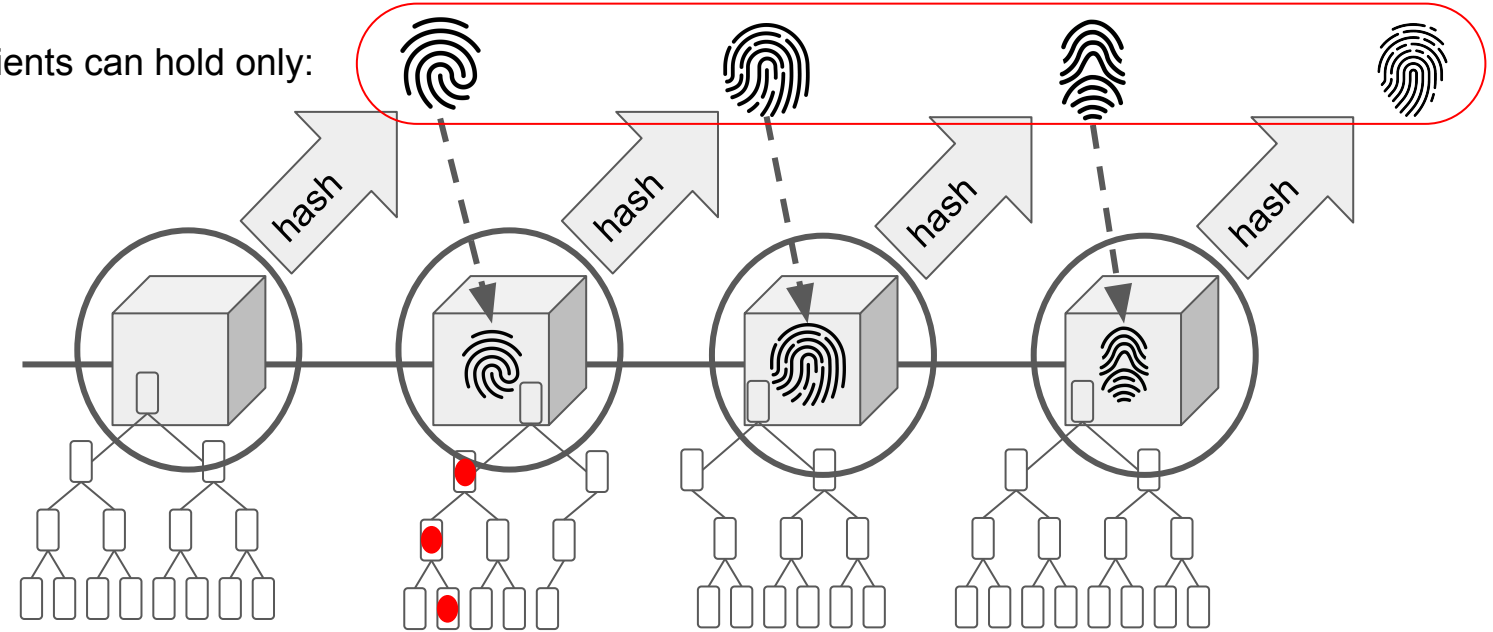
Light-clients can hold only:



Hashes – Immutability

Each block contains a merkle tree of transactions and the hash of the previous block:

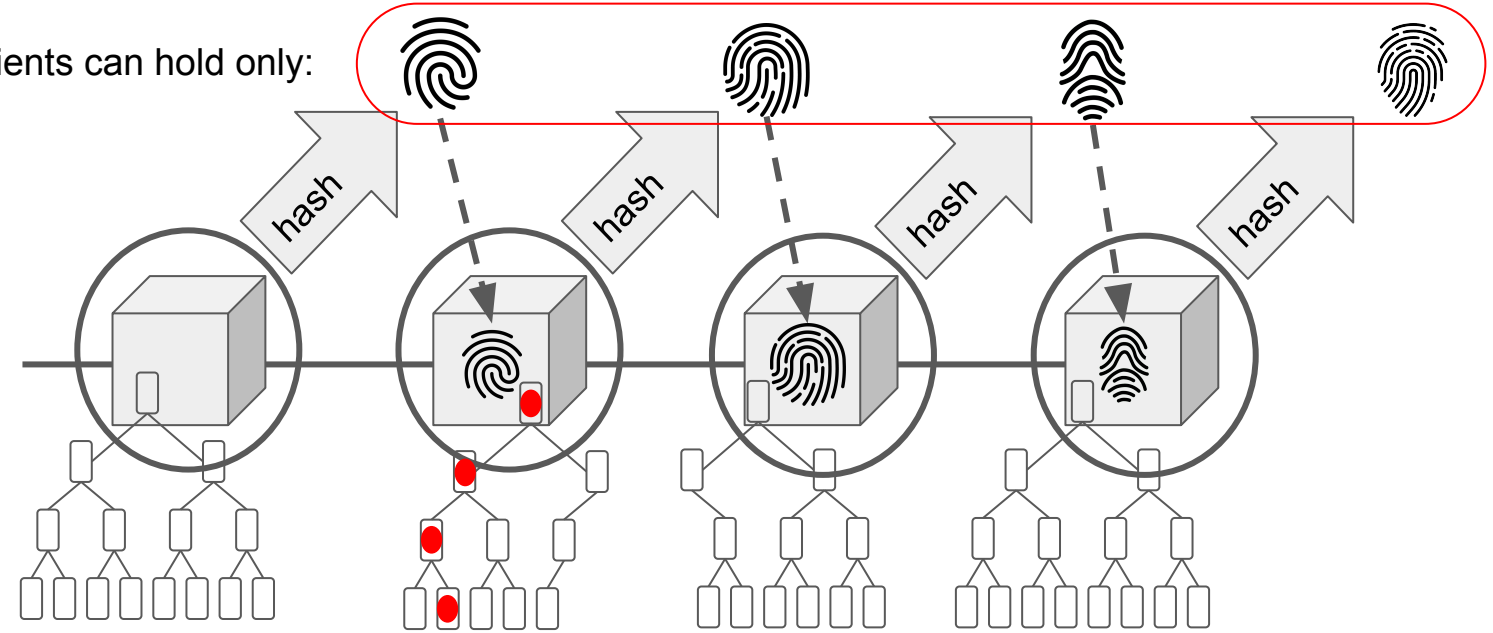
Light-clients can hold only:



Hashes – Immutability

Each block contains a merkle tree of transactions and the hash of the previous block:

Light-clients can hold only:



Hashes – Immutability

Each block contains a merkle tree of transactions and the hash of the previous block:

Light-clients can hold only: