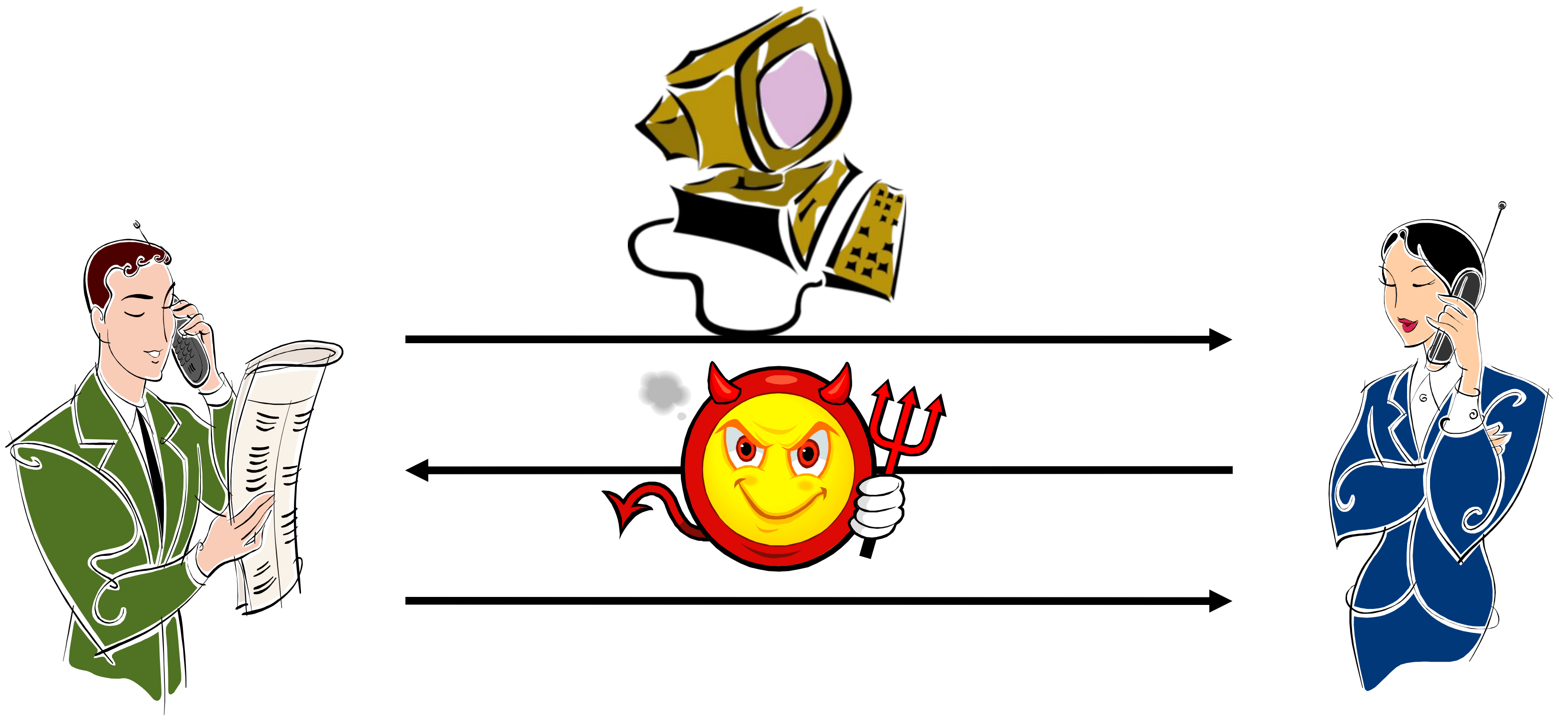
Towards a Quantum-Safe Central Bank Digital Currency

Vadim Lyubashevsky

IBM Research Europe, Zurich

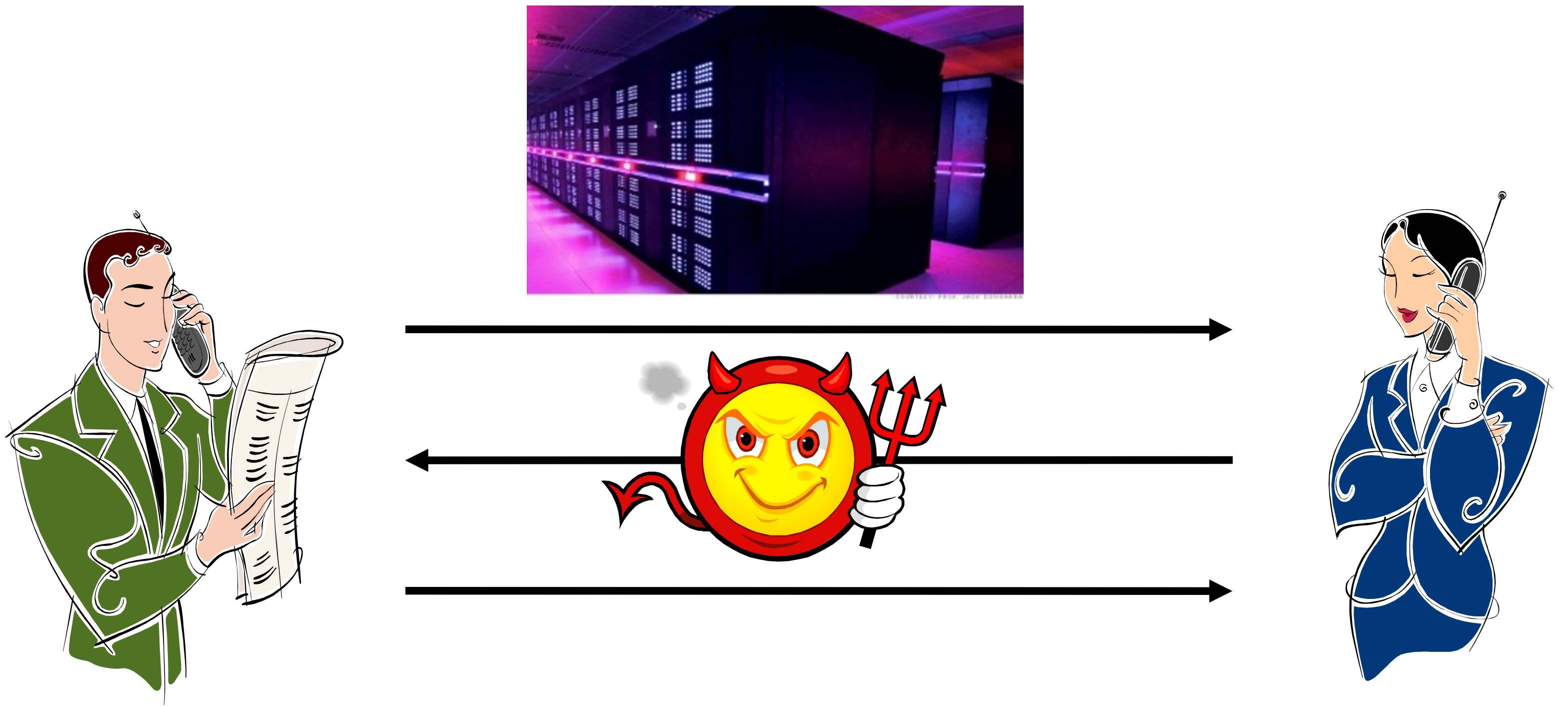
Cryptography

Allows for secure communication in the presence of malicious parties



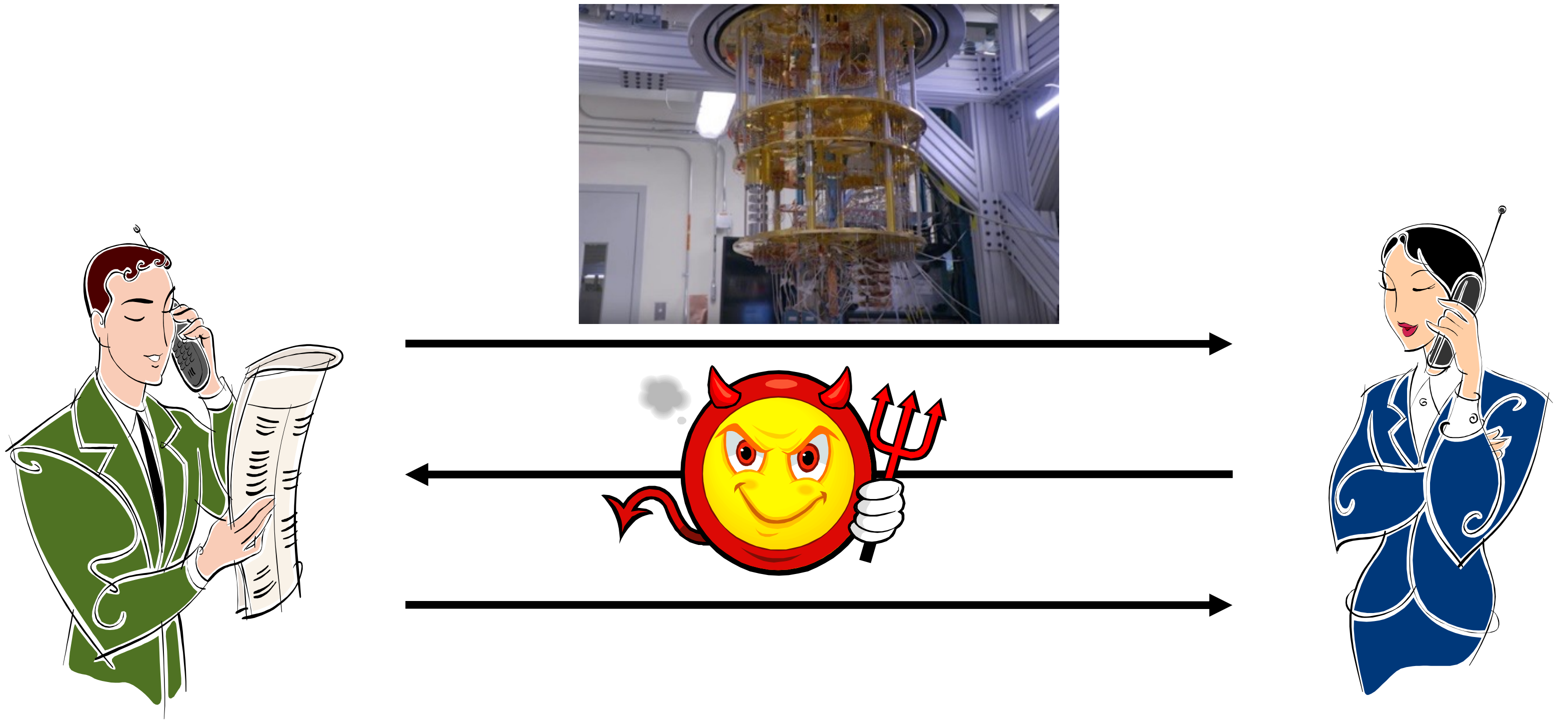
Cryptography

Large increase in the adversary's computing power requires only a small increase in the key size



Cryptography

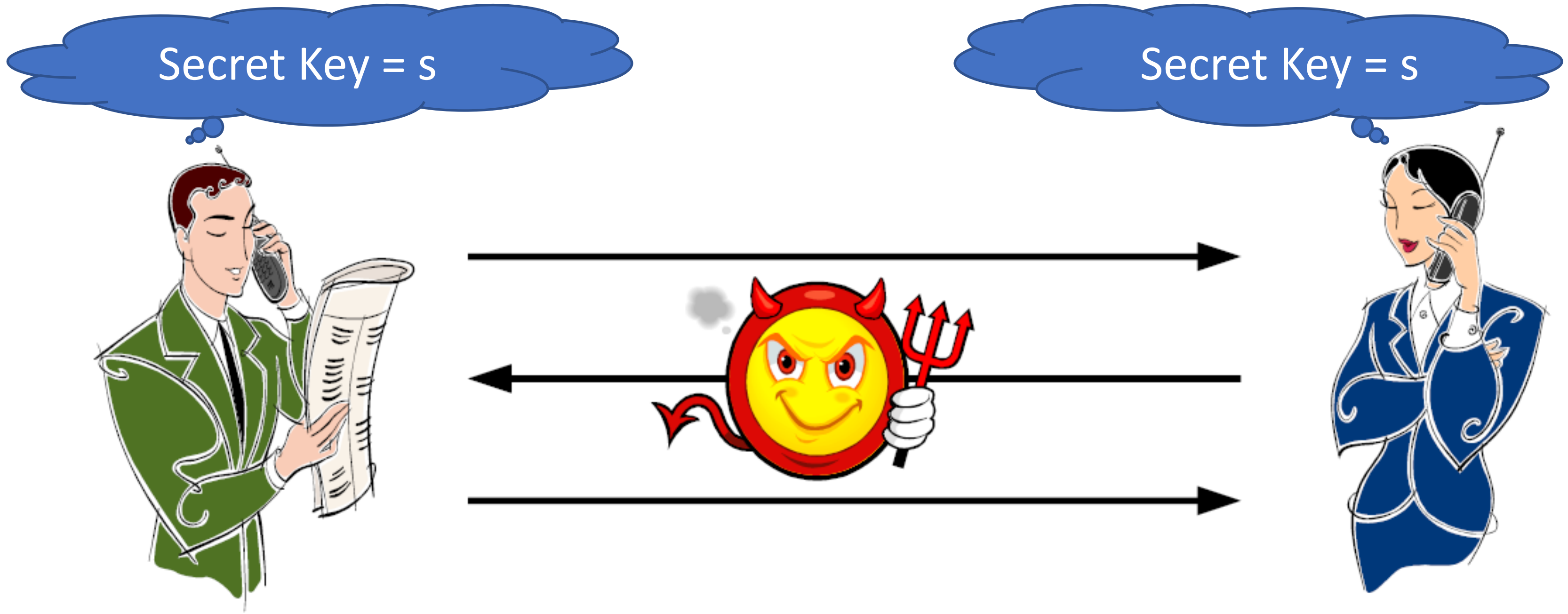
A quantum computer is outside the classical model of computation for efficiency purposes



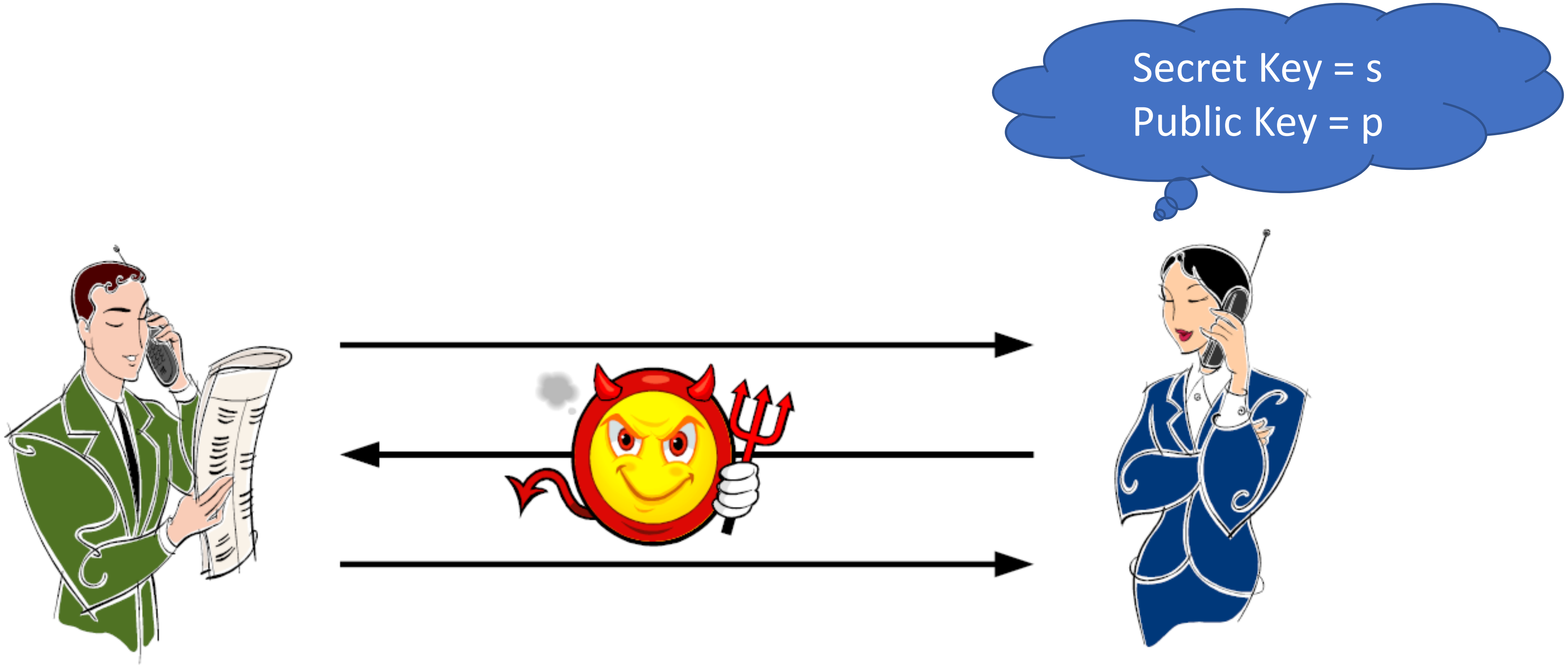
Symmetric-Key Cryptography

Symmetric-Key Cryptography

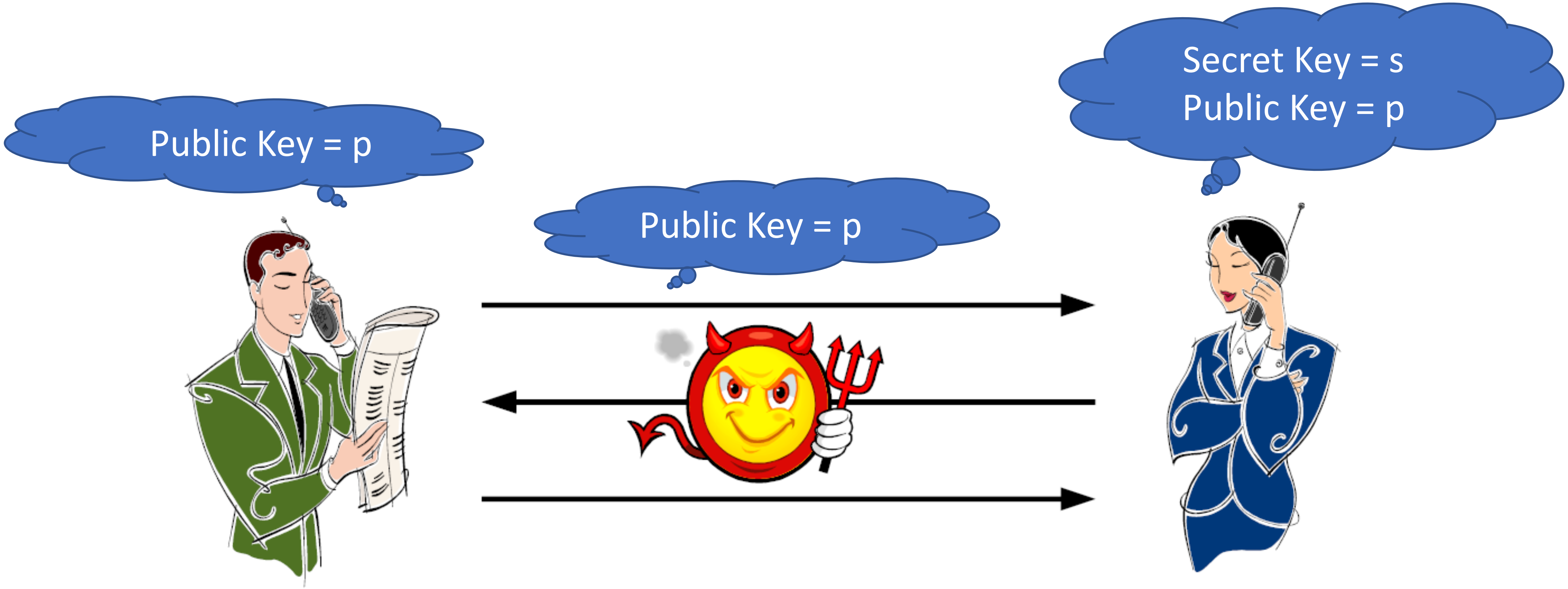
Will still exist if quantum computers are built



Public-Key Cryptography



Public-Key Cryptography



Mathematical Assumptions for Public-Key Cryptography

Factoring is hard

$N=pq$

Computing discrete logs is hard

$g^x=y \bmod p$

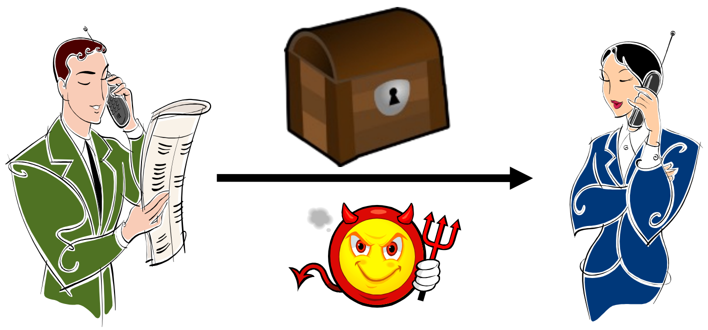
Mostly problems from number theory

All broken once a quantum computer is built

Consequence of quantum computing

Current public key schemes will be broken

Quantum computers will recover all of **today's** secrets



Do not need quantum to defend against quantum

Quantum computers are not all-powerful.

They simply solve some problems faster.

Base cryptography on problems they don’t solve.

How do we know that (quantum) computers don’t solve a problem?

We don’t … all we can say is that researchers tried to solve the problem for X decades and failed.

Categories of Quantum-Safe Crypto

No Changes Necessary

Symmetric Cryptography:

- AES
- SHA-256 / SHA-3
- HMAC
- etc.

Done.

Almost Drop-in Replacements

NIST standardizations:

- Public Key Encryption
- Key Exchange
- Digital Signatures

A few other things:

- Identity-Based Encryption

Almost standards. Ready for deployment.

Serious Alterations of Protocols Required

Advanced Primitives:

- Zero-Knowledge Proofs
- Distributed Privacy
- Many blockchain privacy applications

Lots of recent progress on design. Near-optimality has just been achieved for certain primitives. Implementation starting at ZRL.

Can Only Be Done with Lattice Cryptography

- Fully-Homomorphic Encryption (FHE) - computation over encrypted data
- Some Obfuscation (still unclear if it can be efficient or have any useful applications)

Implementation / deployment of FHE at Haifa.

Categories of Quantum-Safe Crypto

No Changes Necessary

Symmetric Cryptography:
- AES
- SHA-256 / SHA-3
- HMAC
- etc.

Done.

Almost Drop-in Replacements

NIST standardizations:
- Public Key Encryption
- Key Exchange
- Digital Signatures

A few other things:
- Identity-Based Encryption

Almost standards. Ready for deployment.

Serious Alterations of Protocols Required

Advanced Primitives:
- Zero-Knowledge Proofs
- Distributed Privacy
- Many blockchain privacy applications

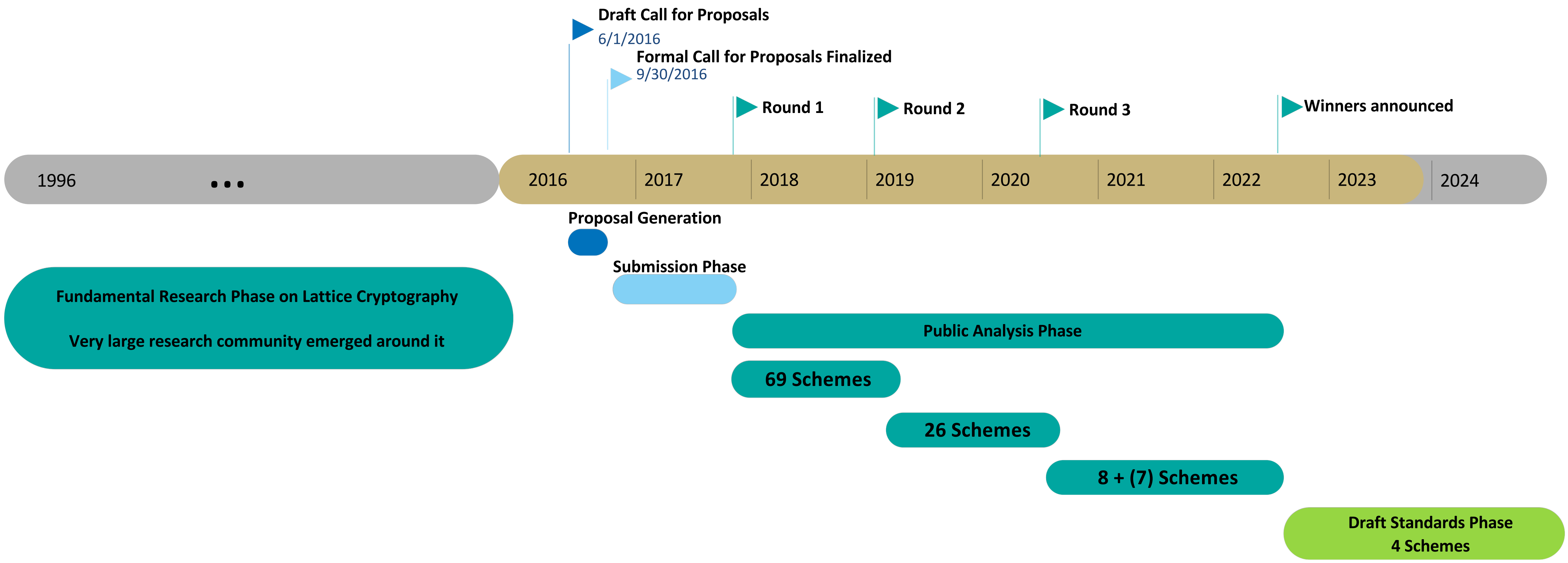
Lots of recent progress on design. Near-optimality has just been achieved for certain primitives. Implementation starting at ZRL.

Can Only Be Done with Lattice Cryptography

- Fully-Homomorphic Encryption (FHE) - computation over encrypted data
- Some Obfuscation (still unclear if it can be efficient or have any useful applications)

Implementation / deployment of FHE at Haifa.

NIST Quantum Safe Standardization



NIST Selection (July 2022)

KEM (Encryption Scheme)

- CRYSTALS-Kyber — Primary

Digital Signature

- CRYSTALS-Dilithium — Primary
- FALCON — Specialized
- SPHINCS+ — Specialized

NSA Selection for CNSA 2.0 (September 2022)

KEM (Encryption Scheme)

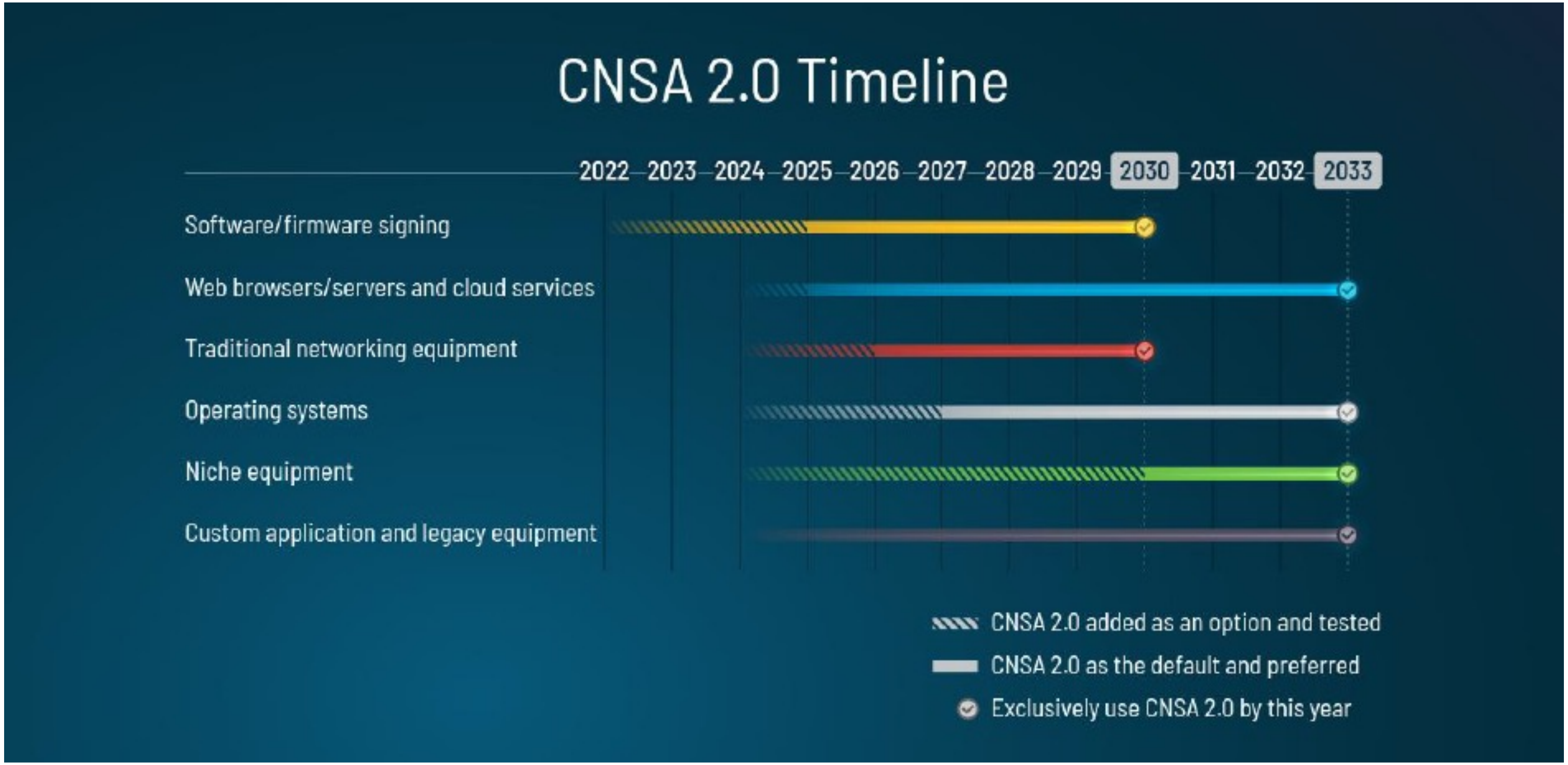
- CRYSTALS-Kyber (Security Level 5: 256-bit security target)

Digital Signature

- CRYSTALS-Dilithium (Security Level 5: 256-bit security target)
- LMS For firmware and software signing only
- XMSS For firmware and software signing only

LMS and XMSS are the **stateful** versions of SPHINCS+

Time for Transition



Categories of Quantum-Safe Crypto

No Changes Necessary

Symmetric Cryptography:
- AES
- SHA-256 / SHA-3
- HMAC
- etc.

Done.

Almost Drop-in Replacements

NIST standardizations:
- Public Key Encryption
- Key Exchange
- Digital Signatures

A few other things:
- Identity-Based Encrypt

Almost standards. Ready for deployment.

Serious Alterations of Protocols Required

Advanced Primitives:
- Zero-Knowledge Proofs
- Distributed Privacy
- Many blockchain privacy applications

Lots of recent progress. Near-optimality has just been achieved for certain primitives. Implementation starting at ZRL.

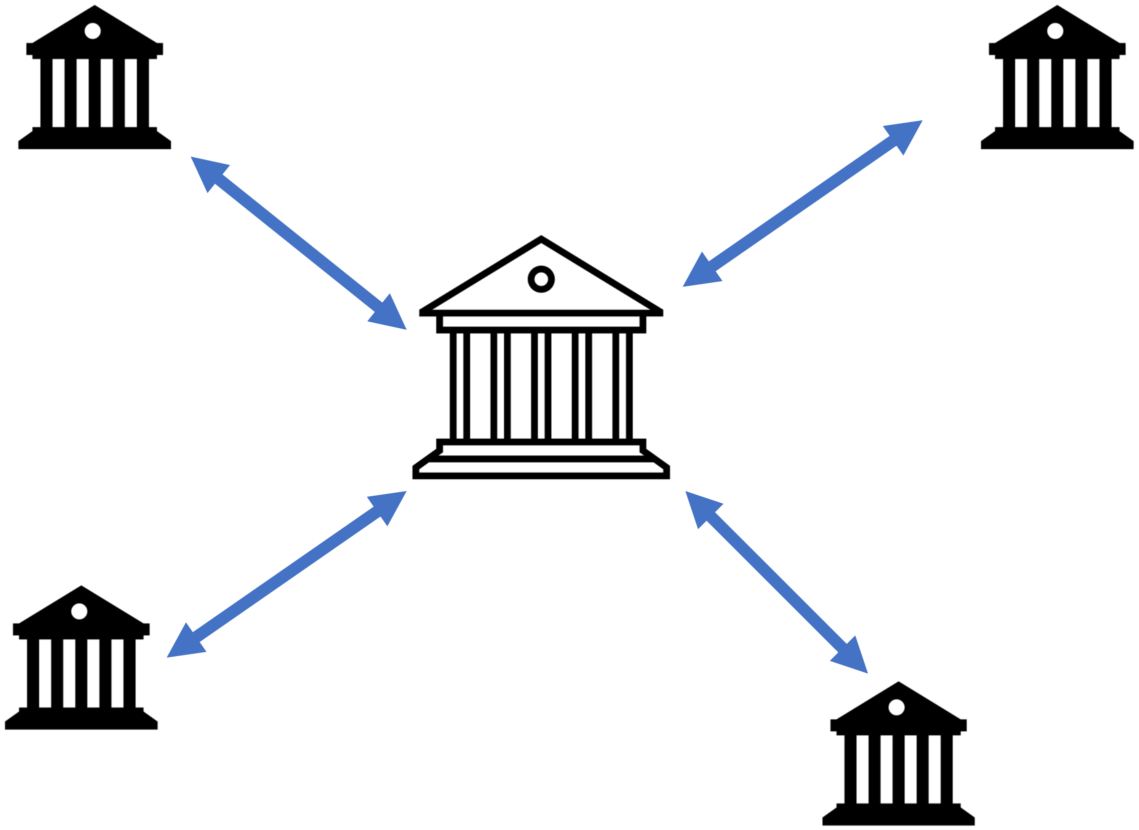
Can Only Be Done with Lattice Cryptography

- Fully-Homomorphic Encryption (FHE) - computation over encrypted data
- Some Obfuscation (still unclear if it can be efficient or have any useful applications)

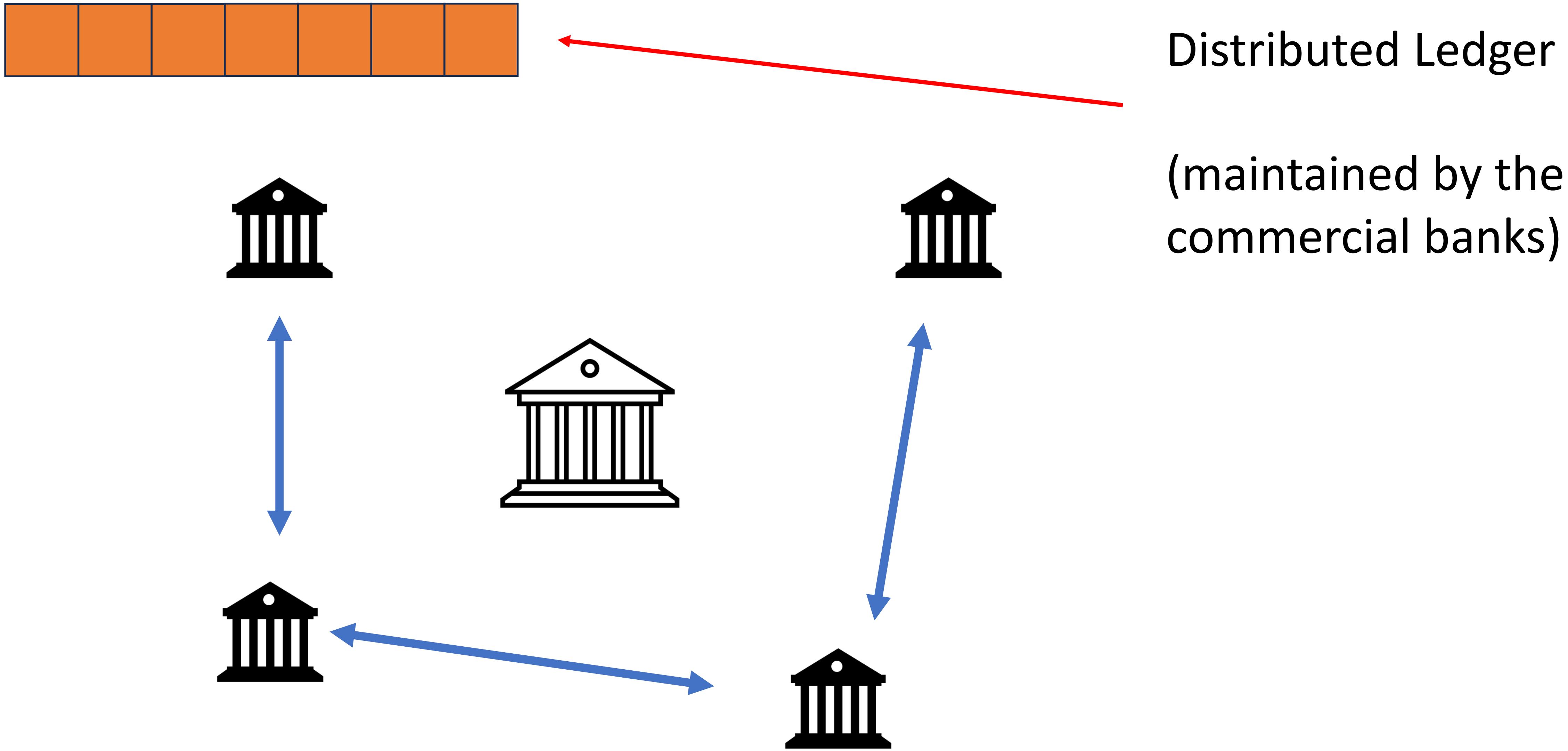
Implementation / deployment of FHE at Haifa.

Central Bank Digital Currency

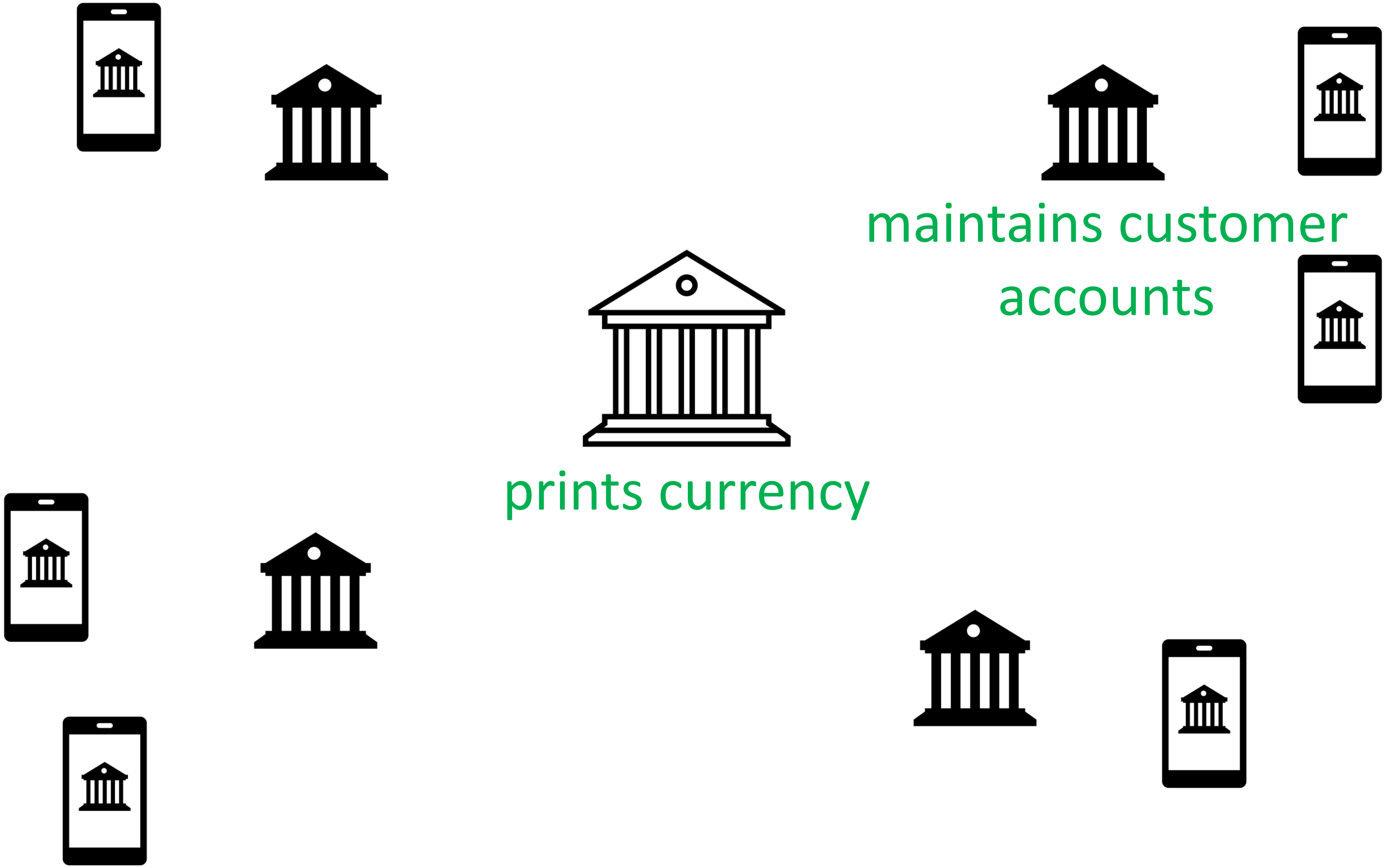
Wholesale CBDC



Wholesale CBDC



Retail CBDC – should have the privacy of cash



(Naïve) Digital Cash

My account is 12345.
Give me $1.

Signature(x)

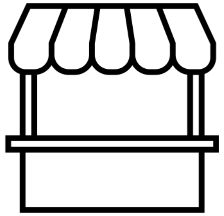
pick random x

Spent list

x

No privacy!

x, Signature(x)



check that x is not on the “spent list”

By seeing the x, the bank traces the purchase to the customer

The Blind Signature Approach

Blind Signature [Cha '82]

pick random x

My account is 12345.
Give me $1.
f(x)

BlindSignature(f(x))

Use a "blind signature" to sign f(x)

derive signature of x

Spent list

x

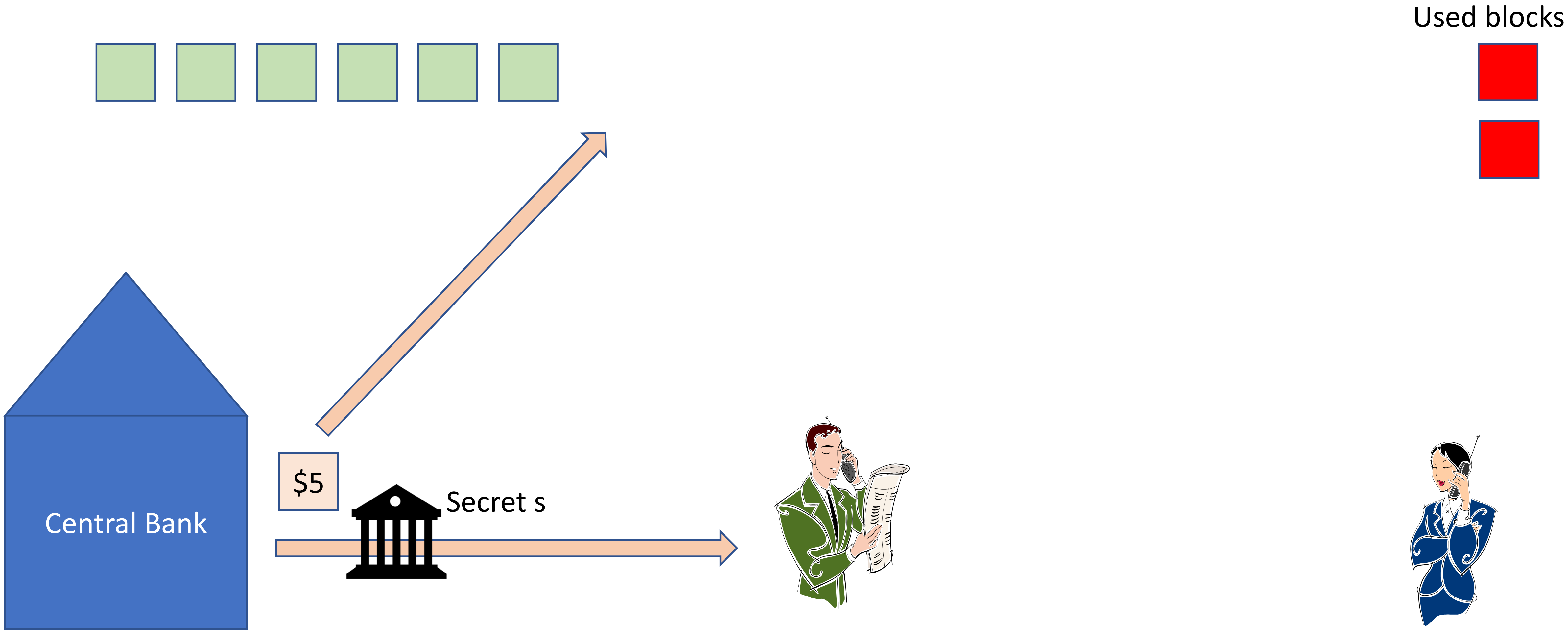
Perfect privacy! Bank cannot trace f(x) to x

x, Signature(x)

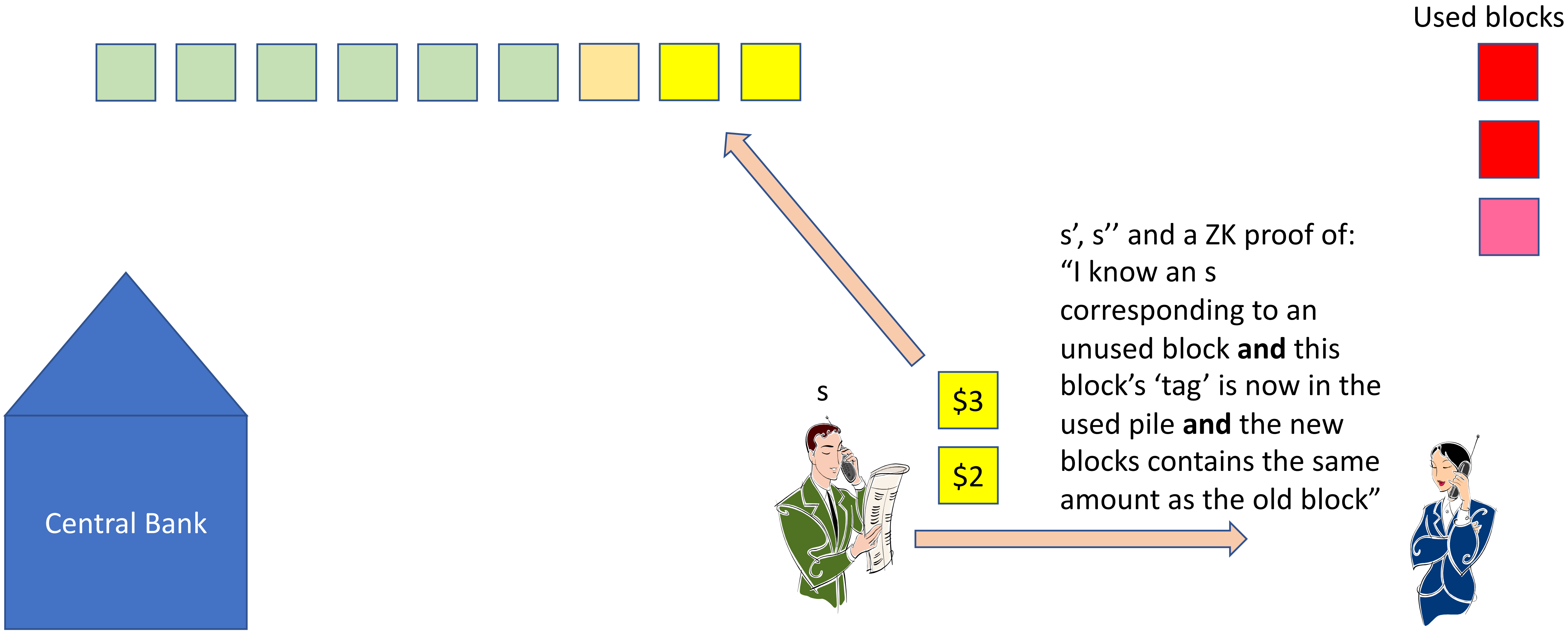
check that x is not on the "spent list"

The Zero-Knowledge Approach

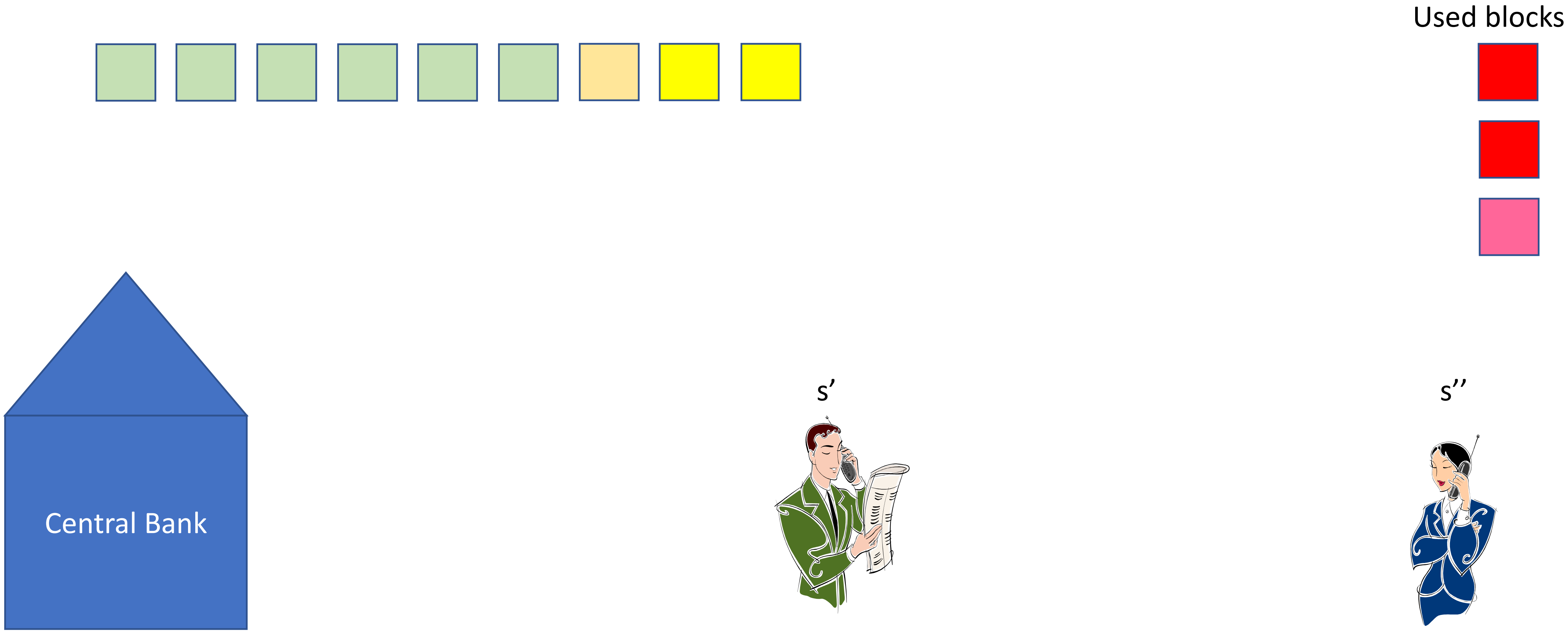
Zero-Knowledge (ZK) and Central Bank Digital Currency (CBDC)



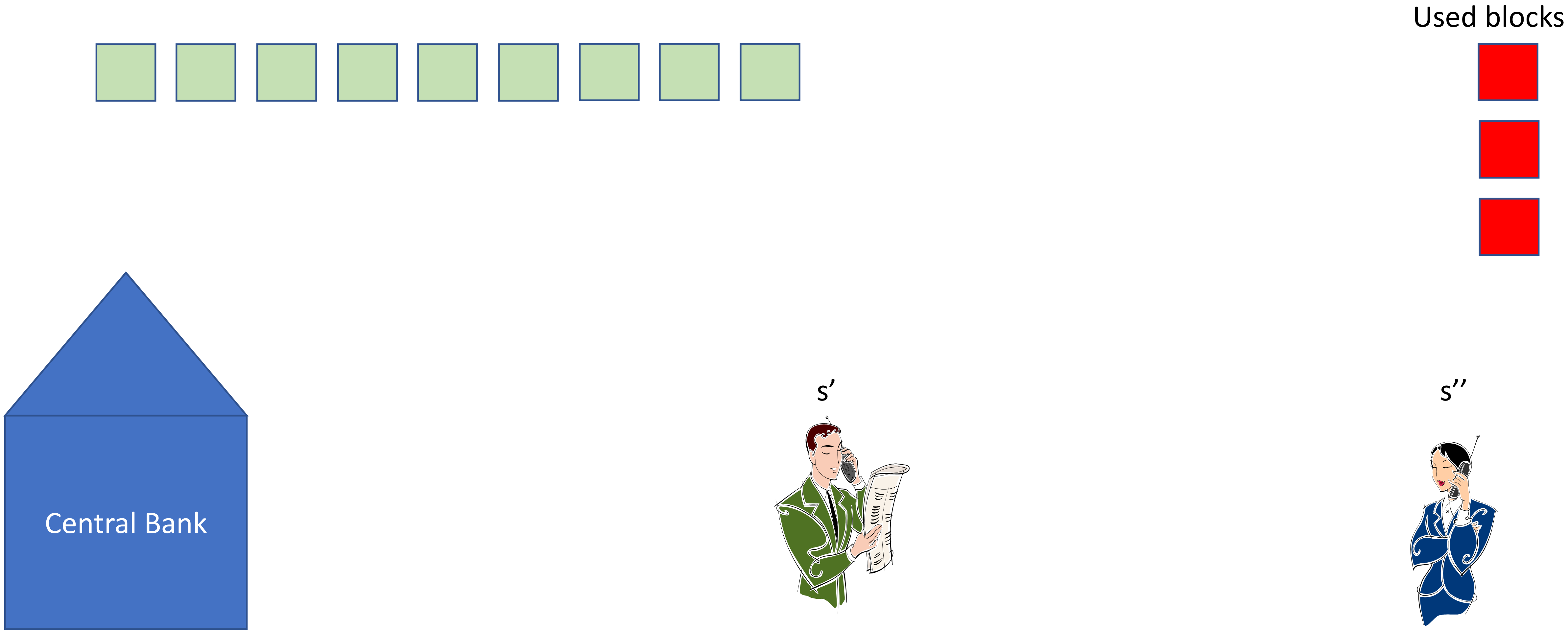
Zero-Knowledge (ZK) and Central Bank Digital Currency (CBDC)



Zero-Knowledge (ZK) and Central Bank Digital Currency (CBDC)



Zero-Knowledge (ZK) and Central Bank Digital Currency (CBDC)



ZK Proofs Past and Present

- The most efficient ZK proofs now are **not** quantum-safe
- CBDC will need to have a clear road map to quantum-safe
- The most efficient quantum-safe proofs seem to be based on lattices

Work of the Quantum-Safe group at ZRL

2018 → 2022

Linear-Size ZK Proofs

3MB → 12KB

Sub-Linear ZK Proofs

?? → 60KB

Lattices and Some Building Blocks

Hard Problem Intuition

$$\begin{pmatrix} & & \\ & \mathbf{A} & \\ & & \end{pmatrix} \begin{bmatrix} \\ \mathbf{y} \\ \\ \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ \\ \\ \end{bmatrix} \mod p$$

Given (**A**,**z**), find **y**

Easy! Use Gaussian elimination.

Hard Problem Intuition

$$\mathbf{A}\,\mathbf{y} + \mathbf{e} = \mathbf{z} \mod p$$

$\underbrace{\mathbf{y}, \mathbf{e}}$ Small coefficients

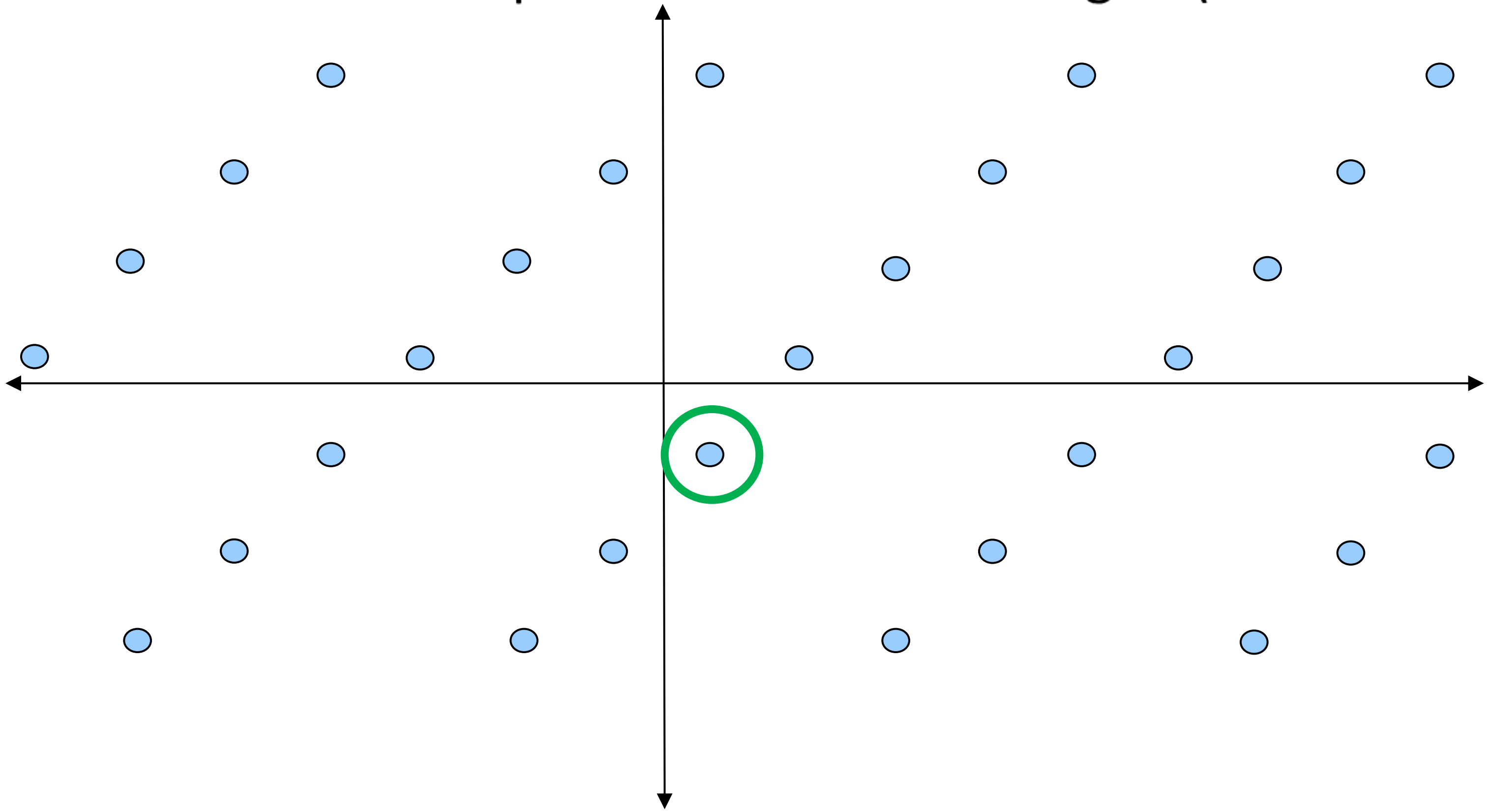
Given (**A**,**z**), find (**y**,**e**)

Seems hard.

Why is this "Lattice" Crypto?

All solutions $\binom{y}{e}$ to **Ay**+**e**=**z** mod p form a "shifted" lattice.

We want to find the point closest to the origin (BDD Problem).



Lattice (Assumption) Basics

Discrete log

- Public element g
- Secret integer s
- One-way function f: Z → Z_q

f(s) = g^s mod q

(g,g^s mod q) is random

Lattices

- Public random matrix A in $Z_q^{n \times m}$
- Secret integer vector s with ||s|| << q
- One-way function f : Z^m → Z_q^n

f(s) = As mod q

(A,As mod q) is pseudorandom

Can create A with a trapdoor that allows inversion of f

Lattice Blind Signatures from ZK Proofs

On the security of giving out pre-images

Random matrix A

An oracle that:

1. Generates a random **y**
2. Generates a small **s** from distribution D such that **As** = **y** mod p

is useless because the same distribution (**s**,**y**) can be generated by

1. Generate a small **s** from distribution D
2. Compute **As** = **y** mod p

The GPV signature scheme

Random matrix A

An oracle that:

1. When given any x
2. Generates a small **s** from distribution D such that **As** = **H(x)** mod p

is useless because the same distribution (**s**,**H(x)**) can be generated by

1. Generate a small **s** from distribution D
2. Compute **As** = **y** mod p
3. Program H(x)=y

Lattice-Based Blind Signature

Public key: A
Secret Key: Trapdoor for A
Public Randomness: B

Message m
Choose vector r with small norm

$t=Br + H(m,H(r))$
ZKPoK π_1 of r,m satisfying above

→

- Check π_1
- Use the trapdoor to compute s with small norm such that $As = t$

s

←

Signature is:

- m
- H(r)
- ZKPoK π_2 of r,s satisfying $As=Br+H(m,H(r))$