Quantum-enhanced Financial Technologies

An overview of QSig Workshop
Edinburgh, UK
26 January 2024

Presented by: Mahshid Delavar, PhD
23 Feb 2024

Agenda

- Money and anti-counterfeiting strategies
- Quantum Money (Private and Public)
- Extensions of Quantum Money: Quantum Lightning and One-shot Signature
- One-shot Signature: How to build it and its applications

History

Anti-Counterfieting Strategies

Isaac Newton

Holograms, embedded strips, “microprinting,” special inks

History

Anti-Counterfieting Strategies

Problem: From a CS perspective, uncopyable cash seems impossible for trivial reasons

Any printing device a good guy can build, a determined bad guy can also build

x → (x,x) is an easy computation

History

Anti-Counterfieting Strategies in Digital World

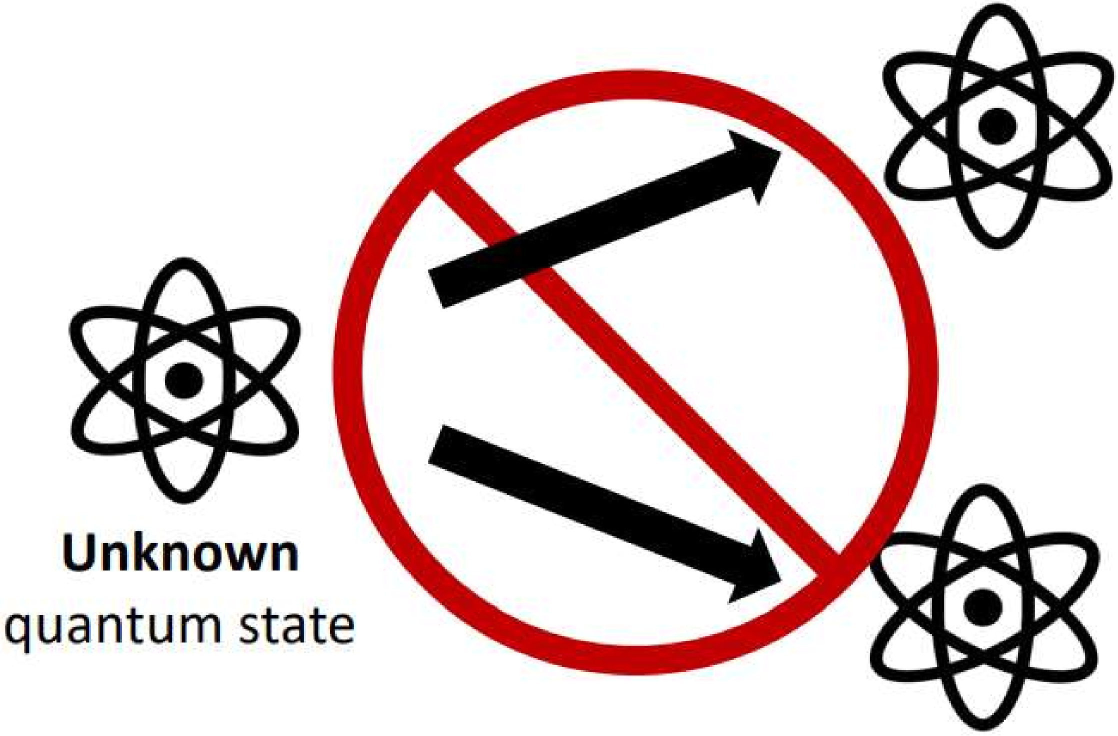
A trusted third party authorizes every transaction

Trusted third party is distributed over the Internet

OK, but sometimes we need cash, especially for privacy reasons, and that seems impossible to secure, at least in classical physics

Enter Quantum

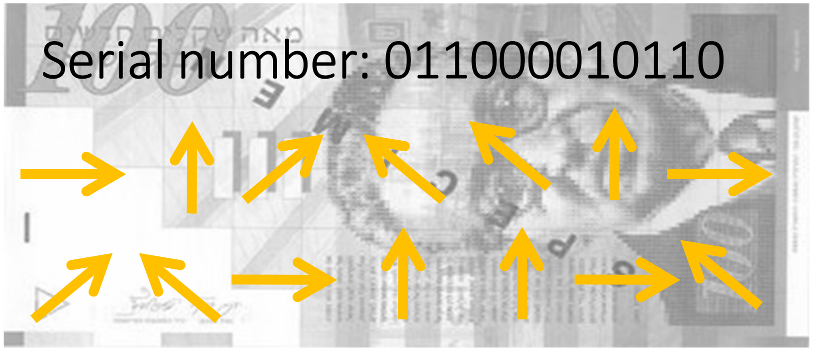
The No-Cloning Theorem



Quantum Money

Private Quantum Money

Wiesner ~1969



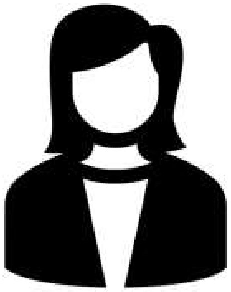
Serial number: 011000010110

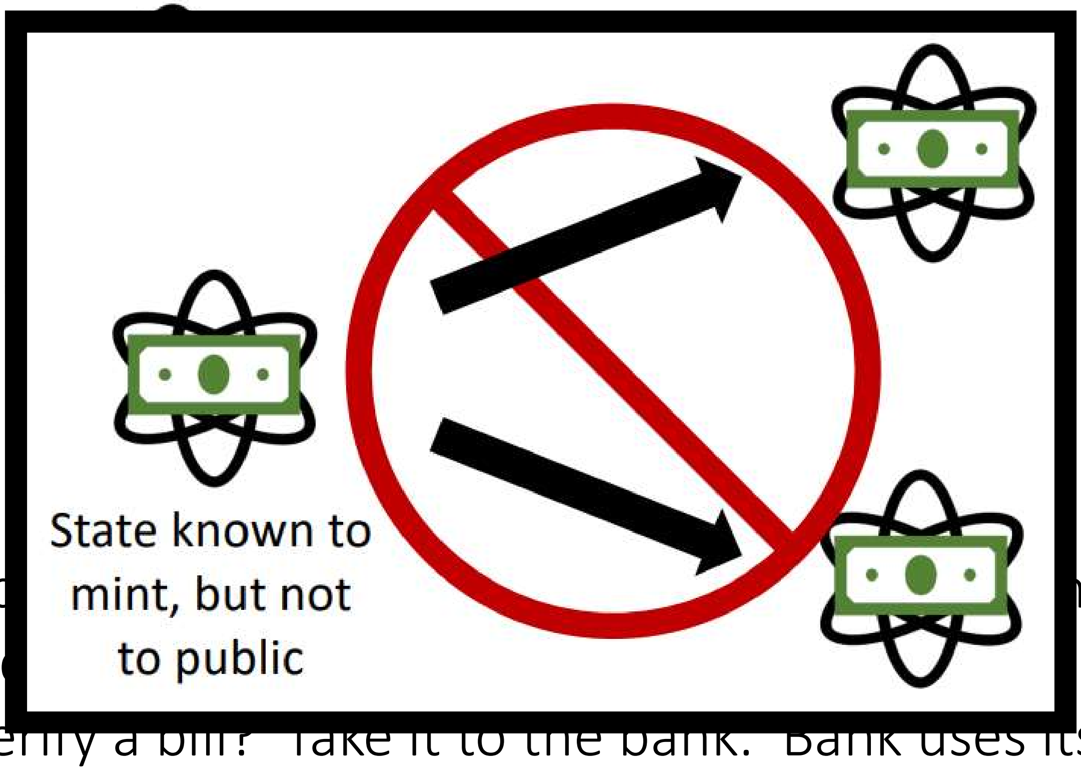
|||

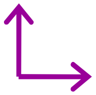
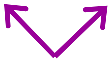
- Each Bill has n qubits
- Each qubit is secretly prepared in one of four BB84 states |0〉,|1〉,|+〉,|-〉

Quantum Money

Private Quantum Money





- In a giant d he quantum state correspond
- Want to verify a bill? Take it to the bank. Bank uses its knowledge to measure each qubit in the correct basis:  OR 

Quantum Money

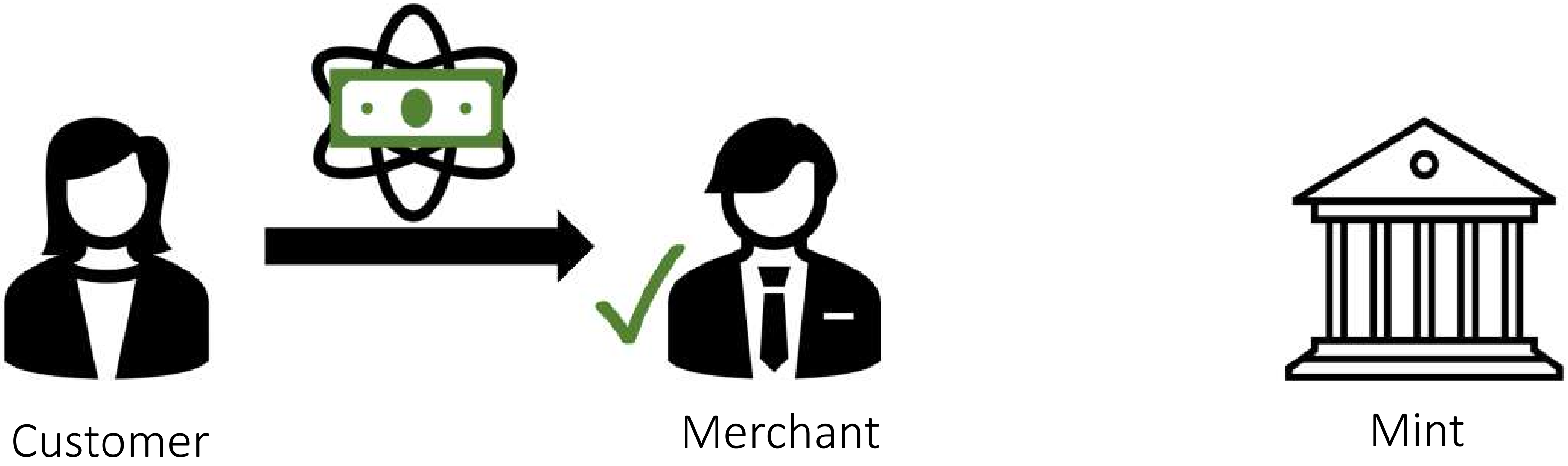
Private Quantum Money

Solves the copyable problem of the cash

But Still, if only the bank can verify the bills, doesn’t that sort of defeat the purpose of cash?

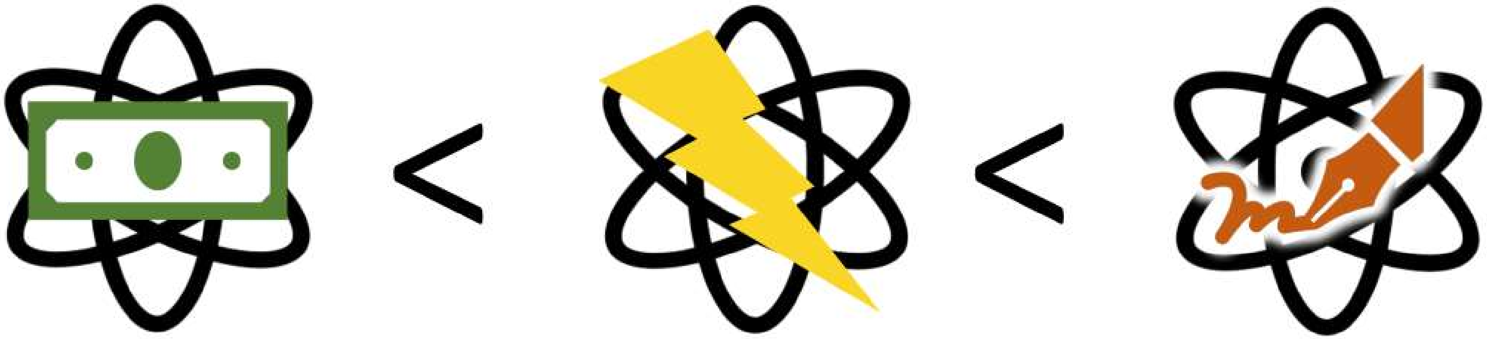
Quantum Money

Public Quantum Money



- Mint only involved in making new notes, not verification
- The procedure to generate new banknotes is kept secret.

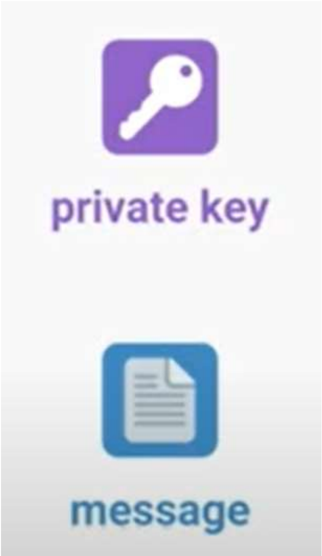
Beyond Quantum Money



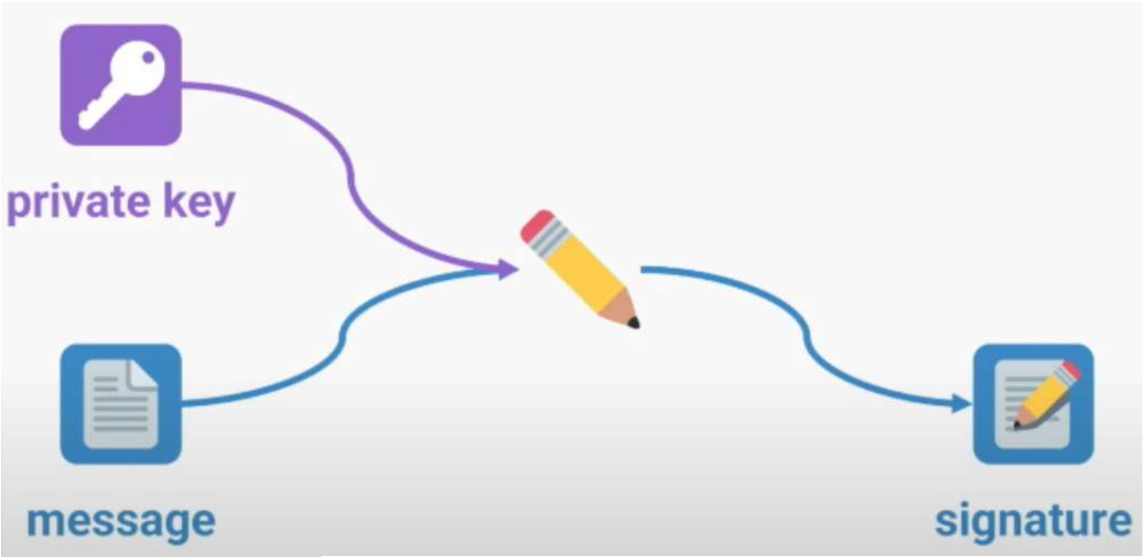
Public Quantum Money Quantum Lightning One-Shot Signature

- Quantum Lightning is a primitive to build a Public Quantum Money where the procedure of creating banknotes is **publicly known**.
- One-shot Signature is a primitive to build a Public Quantum Money with **Classical Communication**.
- One of their applications is creating *Decentralized Blockchain-Less Cryptocurrency*

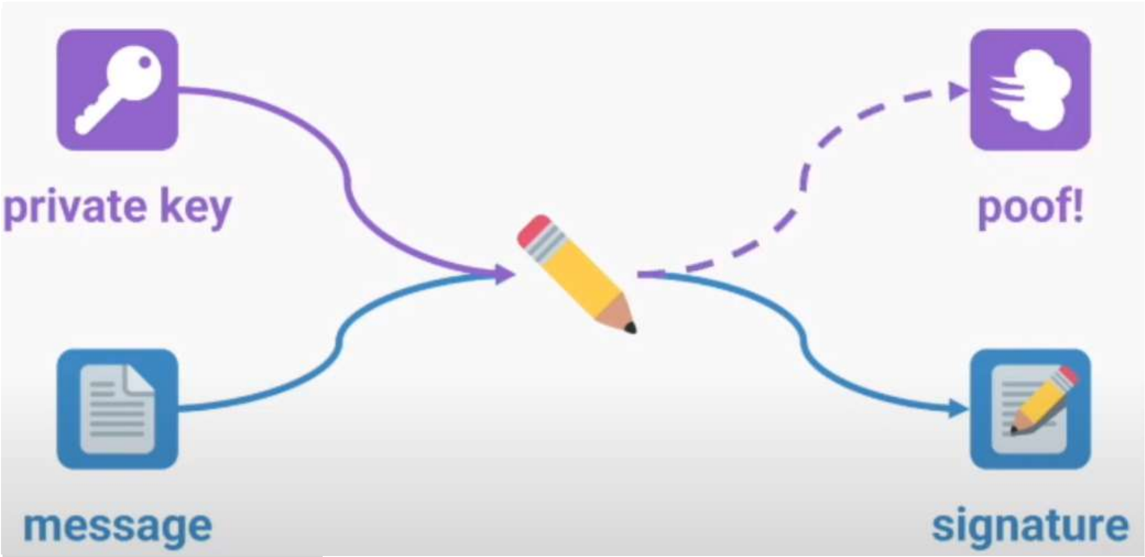
One-Shot Signature



One-Shot Signature



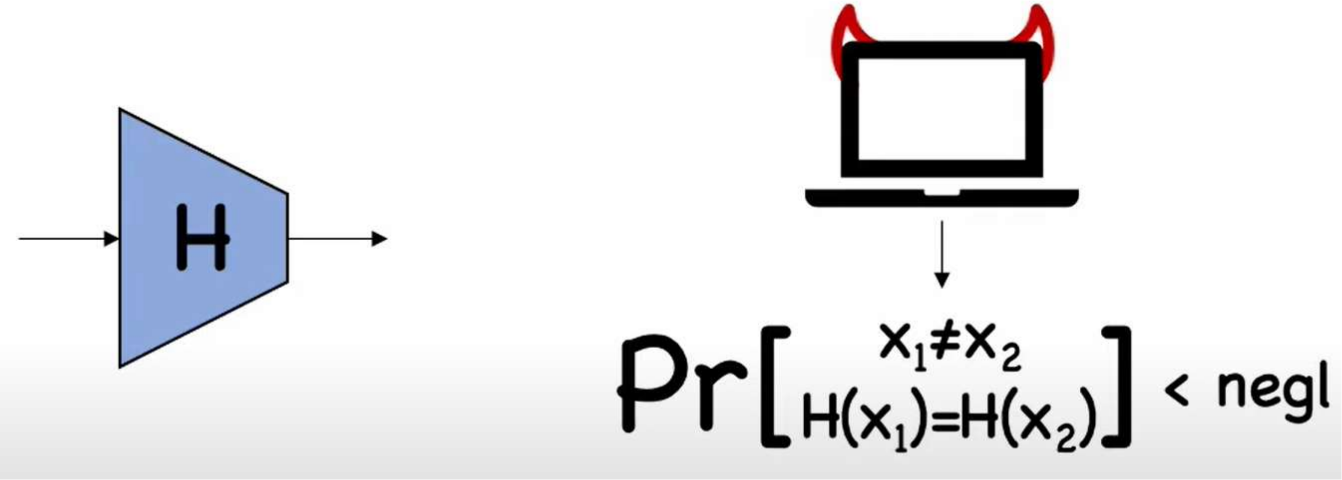
One-Shot Signature



One-Shot Signature

How to build it?

Levels of Security of Hash functions



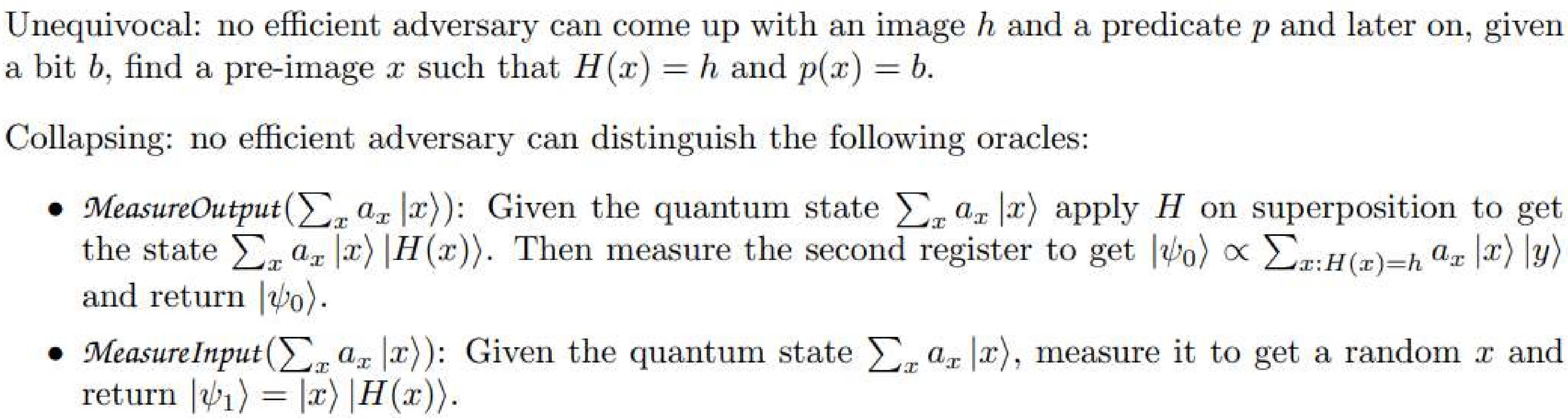
$$\Pr\left[\begin{matrix}x_1 \neq x_2 \\ H(x_1)=H(x_2)\end{matrix}\right] < \text{negl}$$

Classical Collision Resistance

One-Shot Signature

How to build it?

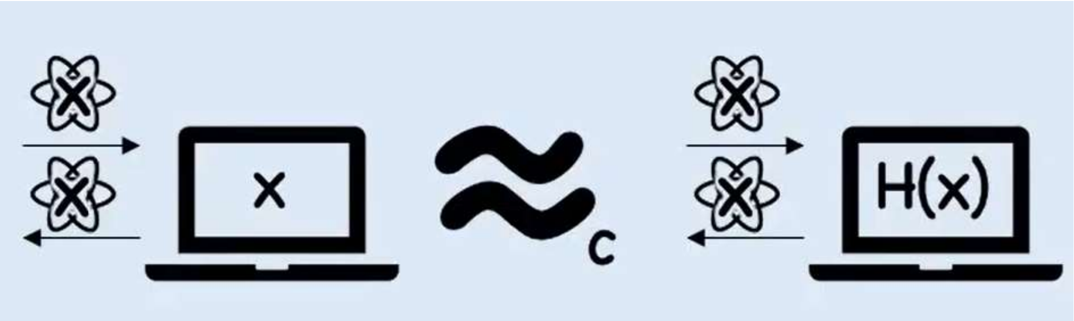
Levels of Security of Hash functions



Unequivocal: no efficient adversary can come up with an image h and a predicate p and later on, given a bit b, find a pre-image x such that $H(x) = h$ and $p(x) = b$.

Collapsing: no efficient adversary can distinguish the following oracles:

- $\mathcal{MeasureOutput}(\sum_x a_x |x\rangle)$: Given the quantum state $\sum_x a_x |x\rangle$ apply H on superposition to get the state $\sum_x a_x |x\rangle |H(x)\rangle$. Then measure the second register to get $|\psi_0\rangle \propto \sum_{x:H(x)=h} a_x |x\rangle |y\rangle$ and return $|\psi_0\rangle$.
- $\mathcal{MeasureInput}(\sum_x a_x |x\rangle)$: Given the quantum state $\sum_x a_x |x\rangle$, measure it to get a random x and return $|\psi_1\rangle = |x\rangle |H(x)\rangle$.



One-Shot Signature

How to build it?

Requires a hash function that is
Collision-Resistant but Equivocal

One-Shot Signature

How to build it?

H is a one-shot chameleon hash function if:

- $Gen(H) \rightarrow (sk, y)$
- $Inv(sk, x) \rightarrow r$ such that:
    - $H(x, r) = y$

One-shot Signature

- $Gen(crs) \rightarrow (sk, pk)$
- $Sign(sk, m) \rightarrow \sigma$
- $Vrfy(crs, pk, m, \sigma) = \{0,1\}$

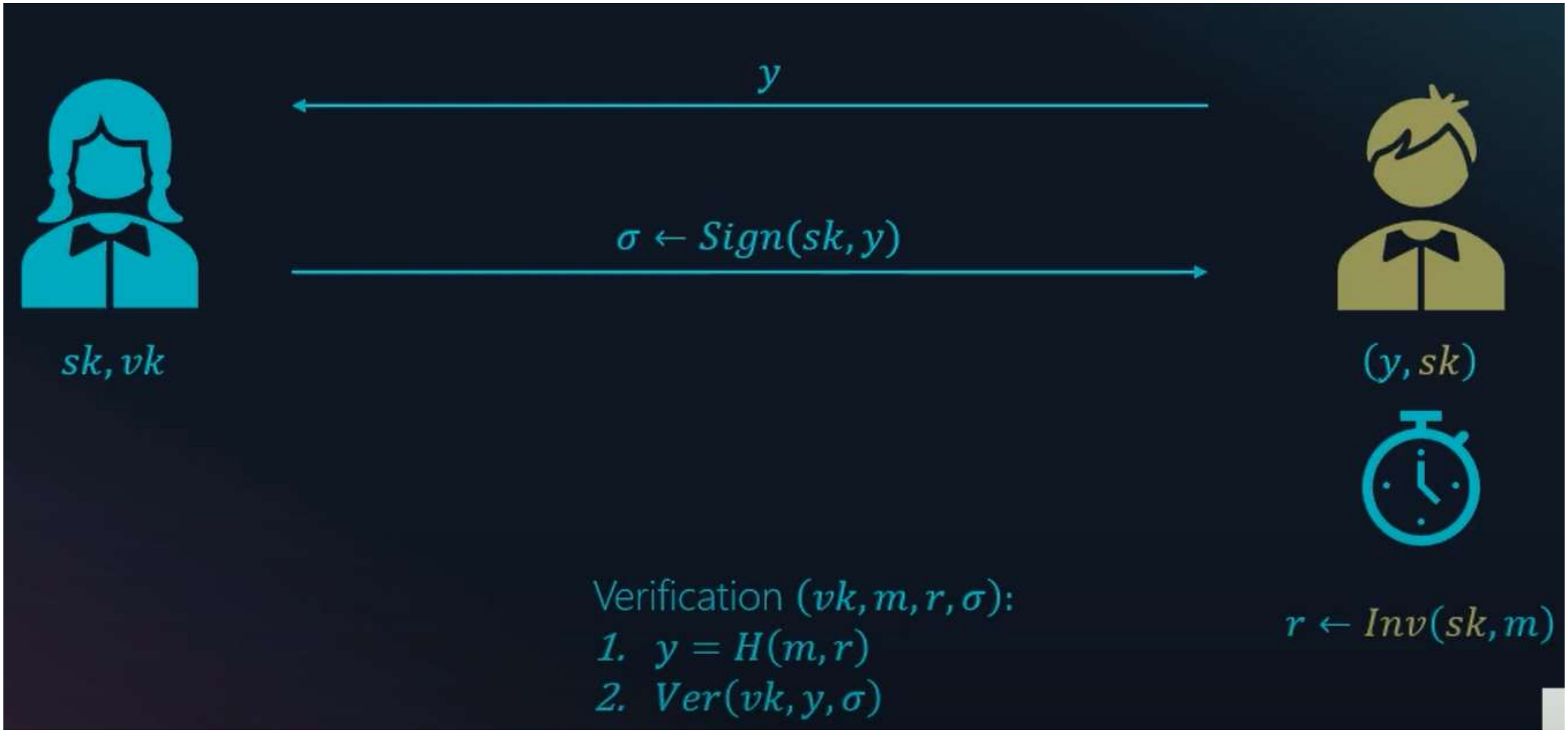
From one-shot chameleon to one-shot signature:

$$H(x, r) = y$$

$x{:}\, m$, $r{:}\, \sigma$ and $y{:}\, pk$

One-Shot Signature

Applications - Signature Delegation



y

$\sigma \leftarrow Sign(sk, y)$

sk, vk

(y, sk)

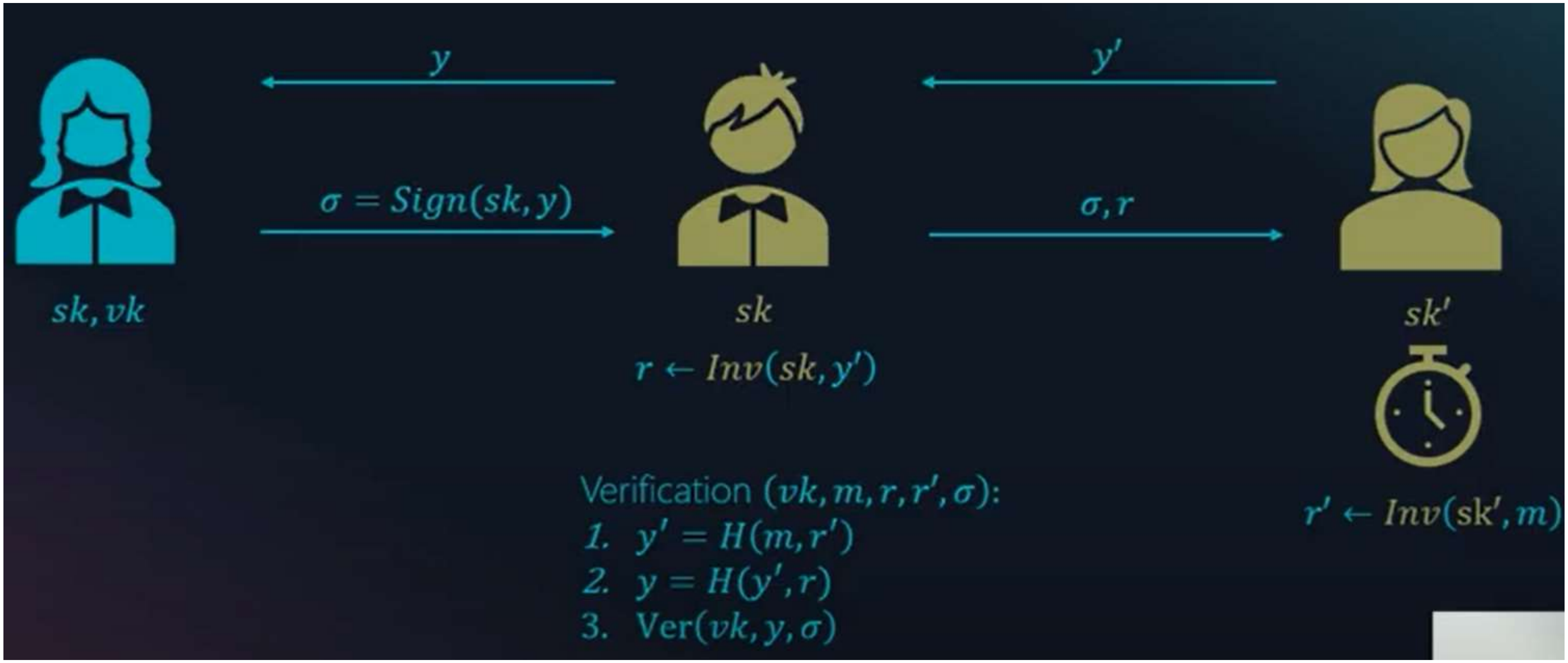
$r \leftarrow Inv(sk, m)$

Verification (vk, m, r, σ):

1. $y = H(m, r)$
2. $Ver(vk, y, \sigma)$

One-Shot Signature

Applications - Signature Delegation

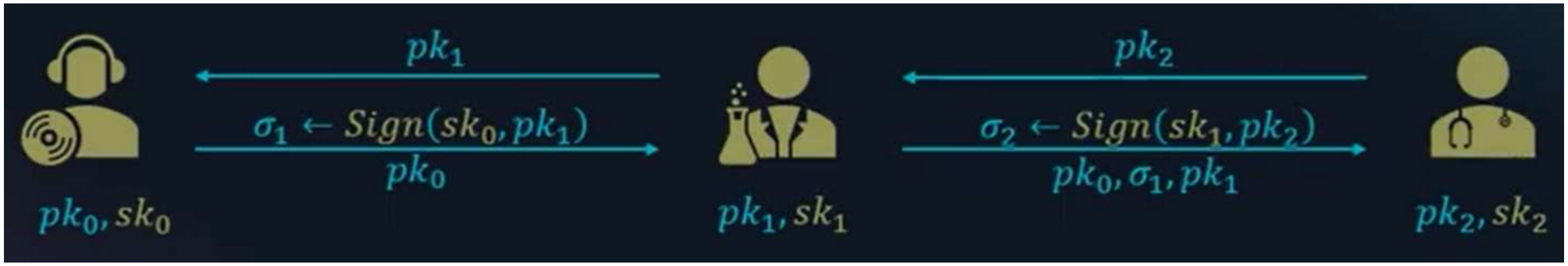

y
y'
$\sigma = Sign(sk, y)$
σ, r
sk, vk
sk
$r \leftarrow Inv(sk, y')$
sk'
$r' \leftarrow Inv(\mathrm{sk}', m)$

Verification (vk, m, r, r', σ):

1. $y' = H(m, r')$
2. $y = H(y', r)$
3. $\mathrm{Ver}(vk, y, \sigma)$

Applications – Blockchain-less Cryptocurrency with Classical Communication

- Mining using Proof of Work: Run $Gen(crs) \rightarrow (sk, pk)$ until the public key starts with 80 zeros


pk_1
$\sigma_1 \leftarrow Sign(sk_0, pk_1)$
pk_0
pk_2
$\sigma_2 \leftarrow Sign(sk_1, pk_2)$
pk_0, σ_1, pk_1
pk_0, sk_0
pk_1, sk_1
pk_2, sk_2

- No need to maintain a public ledger
- Consensus is required only on the crs
- Sending money requires classical communication

Any Questions?