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
Anti-Counterfieting Strategies

Isaac Newton

Holograms, embedded strips, “microprinting,” special inks
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

\[ x \rightarrow (x, x) \text{ is an easy computation} \]
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
The No-Cloning Theorem

Unknown quantum state
Private Quantum Money
Wiesner ~1969

- Each Bill has $n$ qubits
- Each qubit is secretly prepared in one of four BB84 states
  $|0\rangle,|1\rangle,|+\rangle,|-\rangle$
Quantum Money

Private Quantum Money

- In a giant database, the bank stores a description of the quantum state corresponding to serial number $s$.
- 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?
Public Quantum Money

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

Levels of Security of Hash functions

Classical Collision Resistance

$$\Pr[H(x_1) = H(x_2)] < \text{negl}$$
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:

- \( \text{MeasureOutput} \left( \sum_x a_x |x\rangle \right) \): 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 \( |\psi_0\rangle \).

- \( \text{MeasureInput} \left( \sum_x a_x |x\rangle \right) \): Given the quantum state \( \sum_x a_x |x\rangle \), measure it to get a random \( x \) and return \( |\psi_1\rangle = |x\rangle |H(x)\rangle \).
How to build it?

Requires a hash function that is Collision-Resistant but Equivocal
How to build it?

$H$ is a one-shot chameleon hash function if:
- $\text{Gen}(H) \rightarrow (sk, y)$
- $\text{Inv}(sk, x) \rightarrow r$ such that:
  - $H(x, r) = y$

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

From one-shot chameleon to one-shot signature:

$H(x, r) = y$

$x: m$, $r: \sigma$ and $y: pk$
Applications - Signature Delegation

Verification \((vk, m, r, \sigma)\):
1. \(y = H(m, r)\)
2. \(Ver(vk, y, \sigma)\)
Applications - Signature Delegation

\[ \sigma = \text{Sign}(sk, y) \]

\[ r \leftarrow \text{Inv}(sk, y') \]

Verification \((vk, m, r, r', \sigma)\):
1. \( y' = H(m, r') \)
2. \( y = H(y', r) \)
3. \( \text{Ver}(vk, y, \sigma) \)
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

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