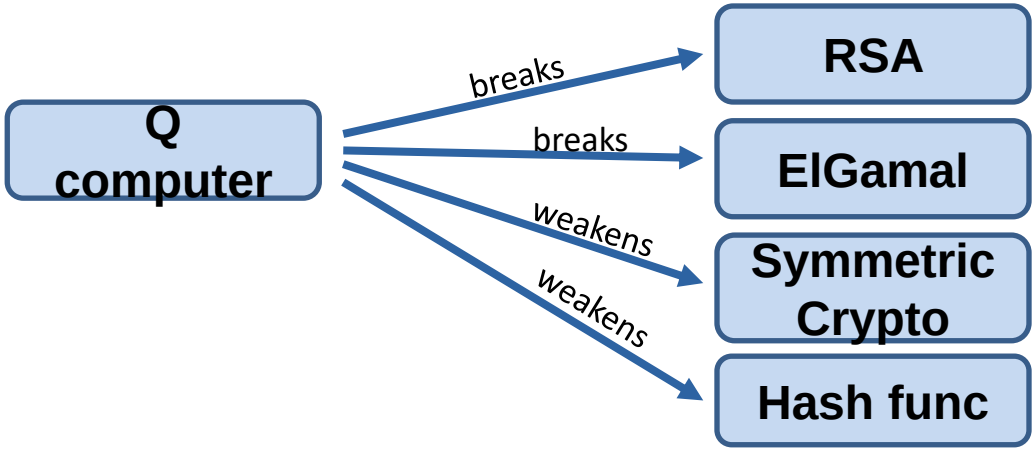
UNIVERSITY OF TARTU
1632

Quantum crypto and the difficulties of proving it

Dominique Unruh

University of Tartu
RWTH Aachen

Postquantum crypto



Solutions:

- Alternatives to RSA, ElGamal ("Post-quantum crypto", NIST competition)
- Use Q mechanics for building crypto!

How can post-quantum crypto fail?

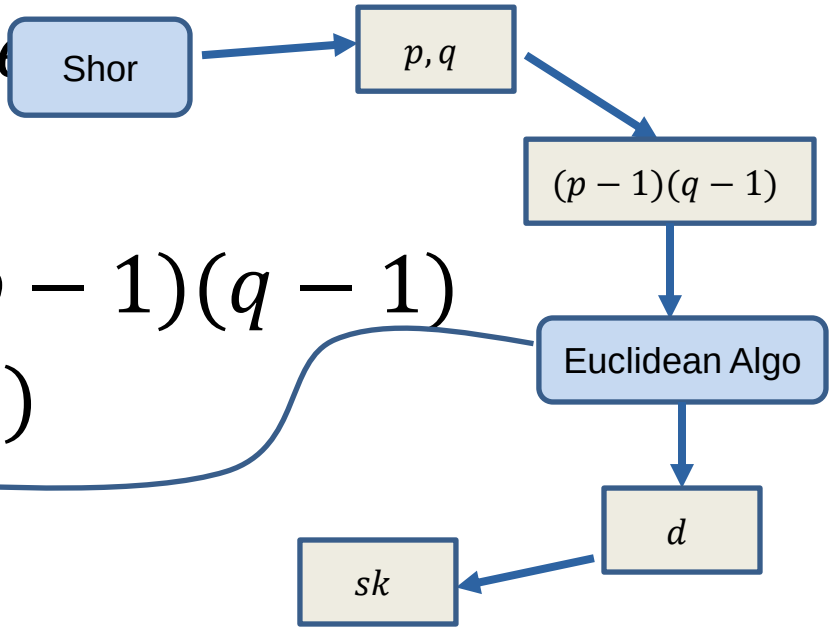
- Underlying assumption wrong.
- Scheme based on assumption broken / not provable
- Scheme secure but weakened

Underlying assumption wrong

RSA (simplified):

Key generation:

- Pick primes p, q. Integer e
- $N := pq$
- $d :=$ inverse of e mod $(p-1)(q-1)$
- $pk = (N, e) \quad sk = (N, d)$

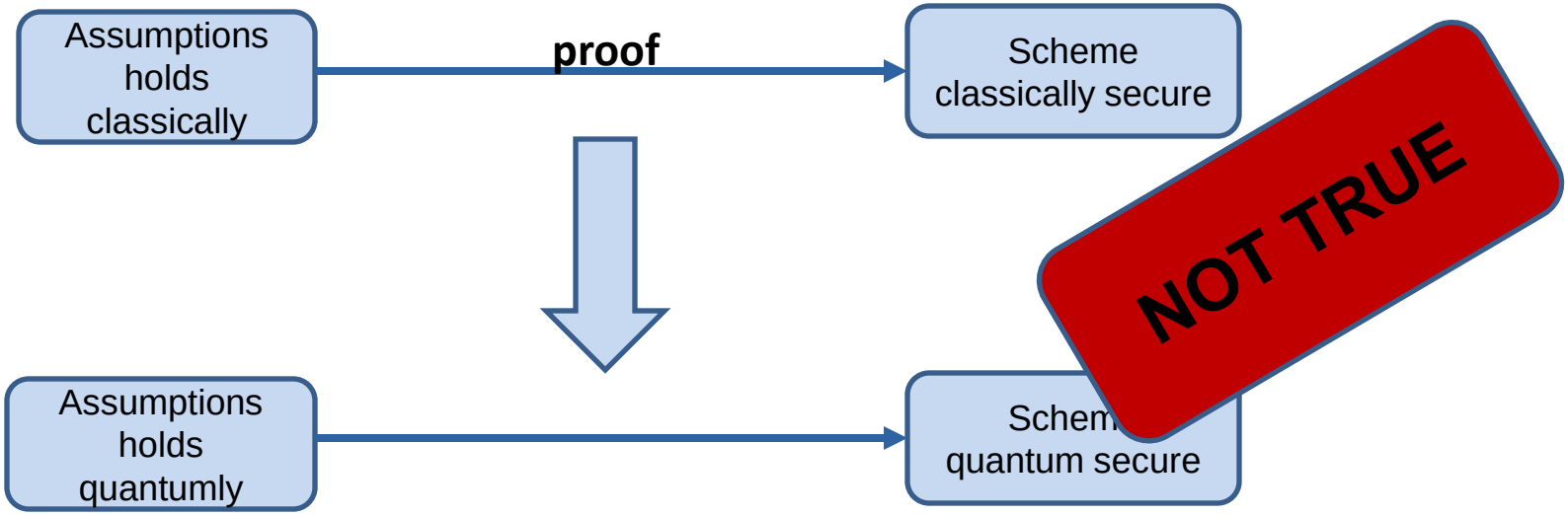


Encryption/Decryption:

* $c = m^e; \quad m = c^d$

Scheme unprovable

The post-quantum fallacy:



Scheme unprovable

- A security proof works by “reductions”
- Transformations done on a quantum/classical adversary
  - From scheme breaker to assumption breaker
- Can’t do the same things to quantum adversaries as to classical ones

Graph Isomorphism

Prover

Graphs G and H are isomorphic

Verifier

Permute G

Permuted graph J

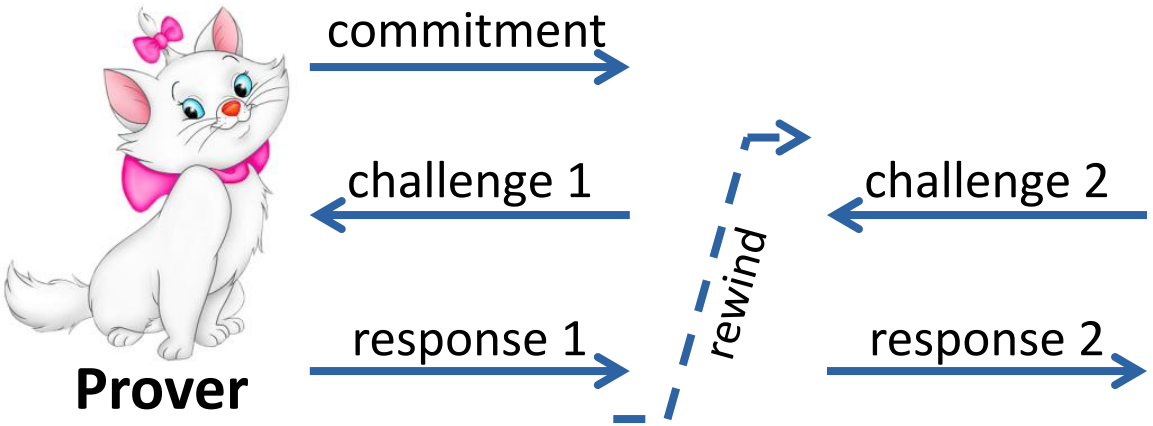
G or H

Pick G or H

Iso between J and G or J and H

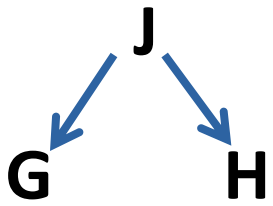
Proof of knowledge: how to show?

Given successful prover, extract witness → violate assm

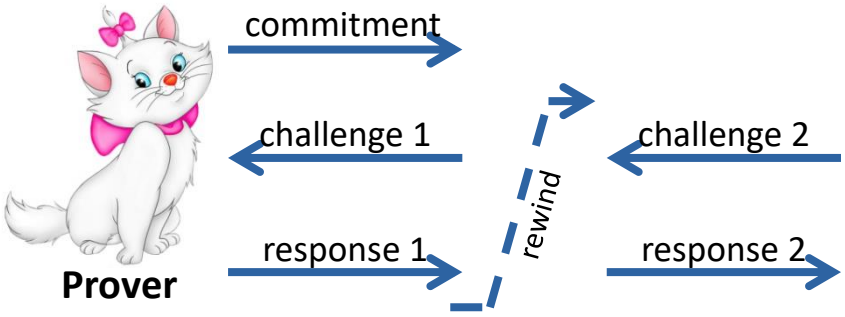


"Special soundness": Two different responses allow to compute witness

- E.g., isomorphisms from J to G and H give isomorphism between G and H



State copying



- Retry/rewind means:
  - Make a copy of the state before execution
  - Restore that state when rewinding
- Quantum setting: Cannot copy the state

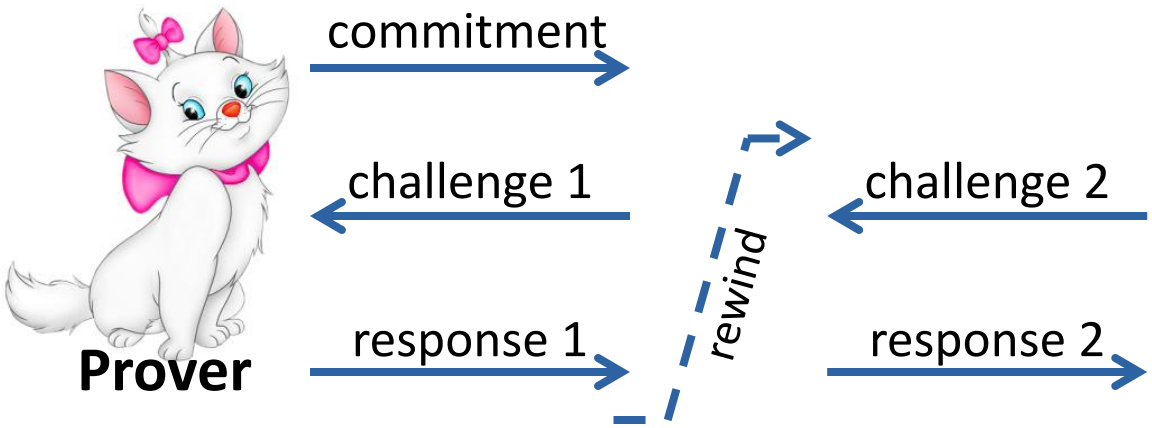
Counterexample

Relative to some oracles (or strong nonstandard assumptions):

- There is a sigma-protocol that
  - Has special soundness
  - is not a proof of knowledge

[Ambainis, Unruh, Rosmanis, Quantum Attacks on Classical Proof Systems]
[Zhandry, Quantum Lightning Never Strikes the Same State Twice]

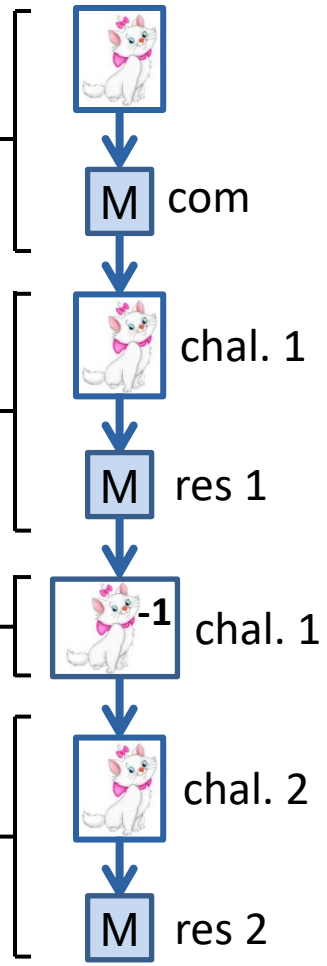
Quantum extractors?



- Quantum case:
  Rewinding = copying. Not possible

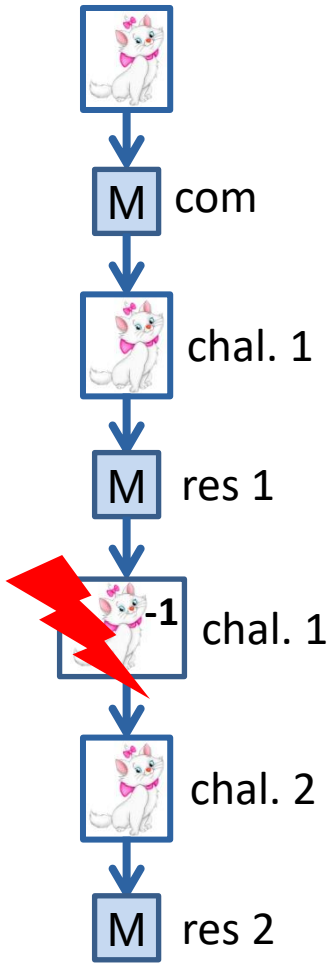
“Canonical extractor”

1. Run prover, measure commitment
2. Run prover on “challenge 1”, measure response 1
3. Run inverse prover
4. Run prover on “challenge 2”, measure response 2



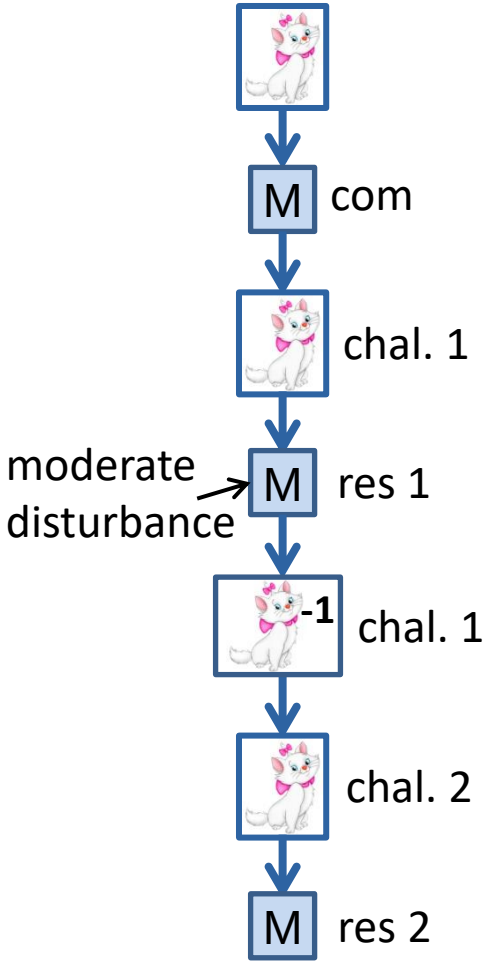
Canonical extractor (ctd.)

- Does it work?
- Measuring “response 1” disturbs state
- Rewinding fails…



Making extraction work

- Thought experiment: "response" was only 1 bit
- Then: measuring "res 1" disturbs only moderately
- Extraction would work

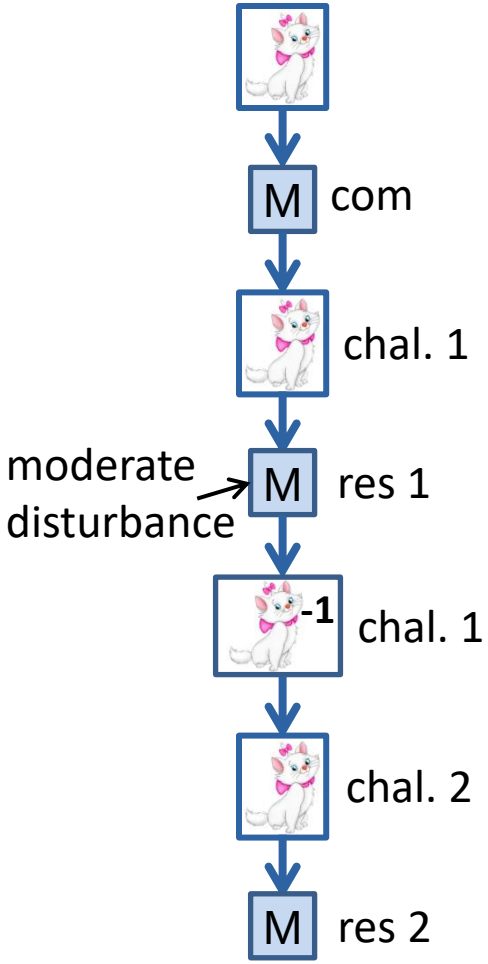


Making extraction work (ctd.)

- Idea: Make "response" effectively be 1 bit
- **"Strict soundness":** For any challenge, exists at most 1 valid response

Needs **rigid** graphs!

- Given strict soundness, canonical extractor works!



[Unruh, Quantum proofs of knowledge]

Summary

- Quantum ZK “proof of knowledge”:
  - Classical security proofs fails,
    even if no assumption quantum-broken
  - In graph-isomorphism case:
    Works, but with harder proof
  - Only for rigid graphs
    (Different protocol can fix that)
- Similar problems in other areas
  - E.g., Fiat-Shamir

Quantum verification / logics

How do we know things are correct?

- Verification of (post-)quantum crypto
- Development of logics for reasoning about quantum programs

Formal verification

- Computer checks correctness/security
- For high assurance systems
- Program verification
- Crypto verification
- Q program verification
- Q crypto verification

} Established research fields

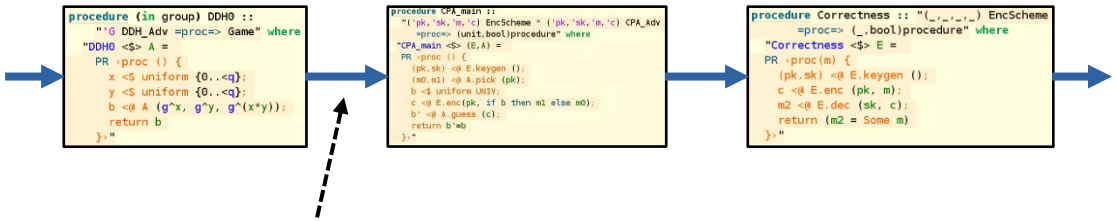
} More novel

} Quite novel

Crypto proofs – bird's eye view

"Sequences of Games"

(established approach for crypto proofs)



How to prove relationship?

Verifying game-based proofs: 2 approaches

- CryptoVerif approach:
  - Have a set of pre-proven rewrite rules
  - Automatically apply them to simplify game

- EasyCrypt approach:
  - User **manually** writes the games
  - User **manually** proves equivalences in "relational Hoare logic"

Hard work, more powerful (this talk)

Important insight

Crypto verification boils down to reasoning about programs

(E.g., Hoare logics and similar)

Relational Hoare Logic (RHL)

- Describes relation of two programs
- How do the variables of the two programs relate?

$$\{x = y\} \quad x \coloneqq x + 1 \sim z \coloneqq y \quad \{x = z + 1\}$$

- Used, e.g., in EasyCrypt for classical verification (using a probabilistic variant)

How about quantum?

- "Games" may contain quantum vars / quantum operations

- Need a quantum version of RHL

Quantum Relational Hoare Logic (qRHL)

- Quantum variables X, Y

Elementary while-language

Subspaces

$$\{X \equiv Y\} \quad apply\ U\ to\ X$$
$$\sim \quad apply\ U\ to\ Y \quad \{X \equiv Y\}$$

- What does $X \equiv Y$ mean?
- How to formalize semantics of qRHL?

Main source of trouble:

Entanglement

[U, Quantum Relational Hoare Logic, POPL 2019]

Specific Challenges

- **EQUAL rule:**

$$\frac{\mathrm{fv}(\boldsymbol{c}) = X}{\{\mathrm{X}_1 \equiv \mathrm{X}_2\}\ \boldsymbol{c} \sim \boldsymbol{c}\ \{X_1 \equiv X_2\}}$$

For reasoning about unknown code (adversaries)

- **FRAME rule:**

$$\frac{R \text{ independent of } \boldsymbol{c}, \boldsymbol{d} \qquad \{A\}\ \boldsymbol{c} \sim \boldsymbol{d}\ \{B\}}{\{A \cap R\}\ \boldsymbol{c} \sim \boldsymbol{d}\ \{B \cap R\}}$$

For modular reasoning

Definition of qRHL

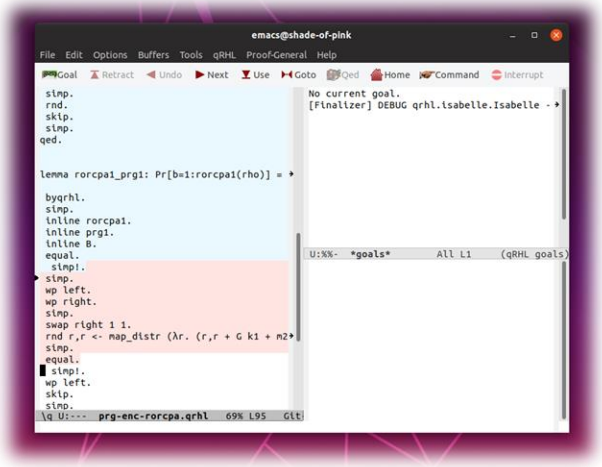
$$\{X = Y\} \quad apply\ U\ to\ X \sim apply\ U\ to\ Y \quad \{X = Y\}$$

For two quantum states that are marginal of a state satisfying this separable

After applying these programs

The final states are marginal of some state satisfying this separable

Our Tool



- Tactics for qRHL
- Verification conditions (VCs): Inequalities of subspaces

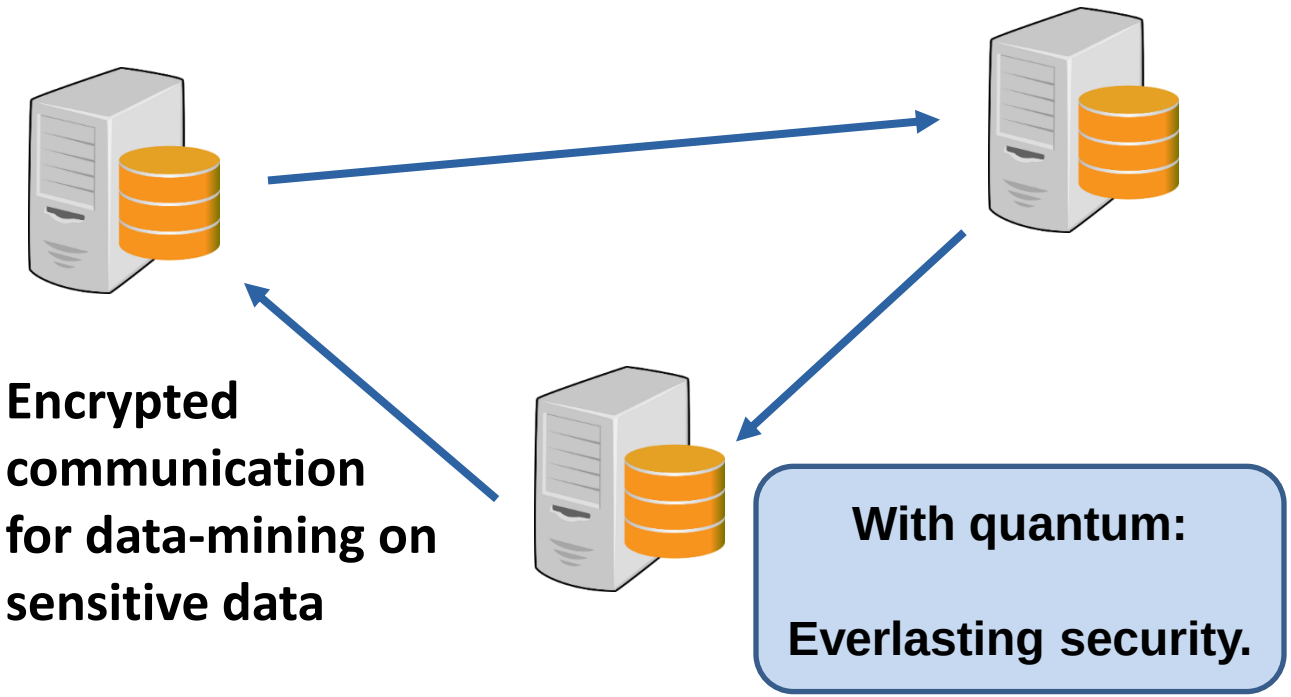
[https://tinyurl.com/qrhl-tool]

- Backend for representing pre-/postconditions
- Reasoning about VCs
- Smooth path towards full verification

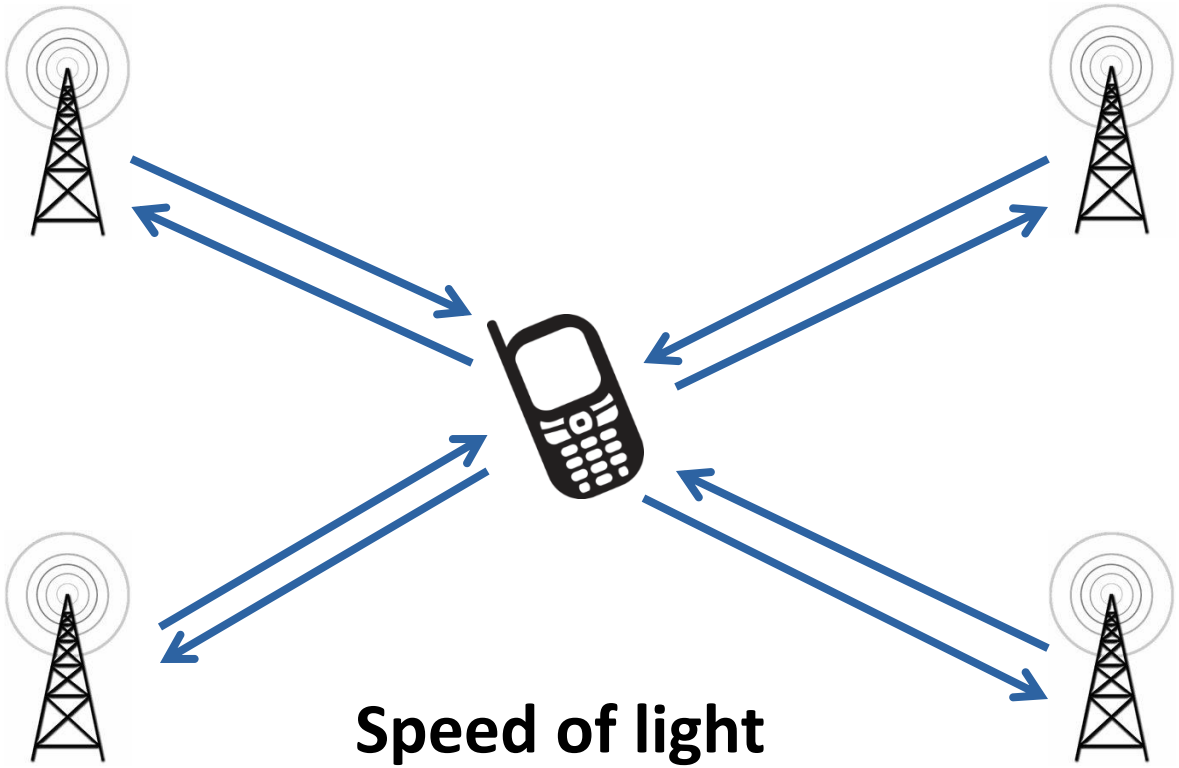
Quantum protocols (beyond classical)

- Can we do new things using quantum?
  - What if honest parties have Q comm/comp?
- Can circumvent classical impossibility
- Extra challenges for hardware, practicality hard

Privacy-preserving data-mining



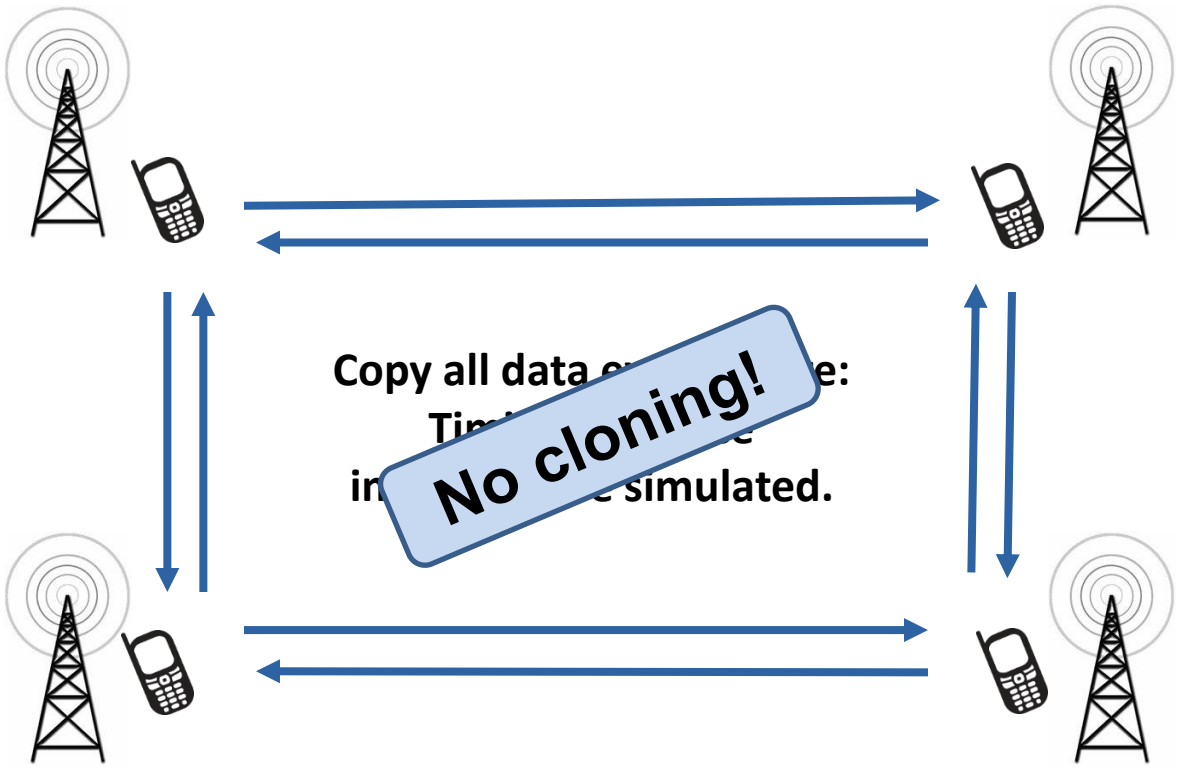
Position Verification



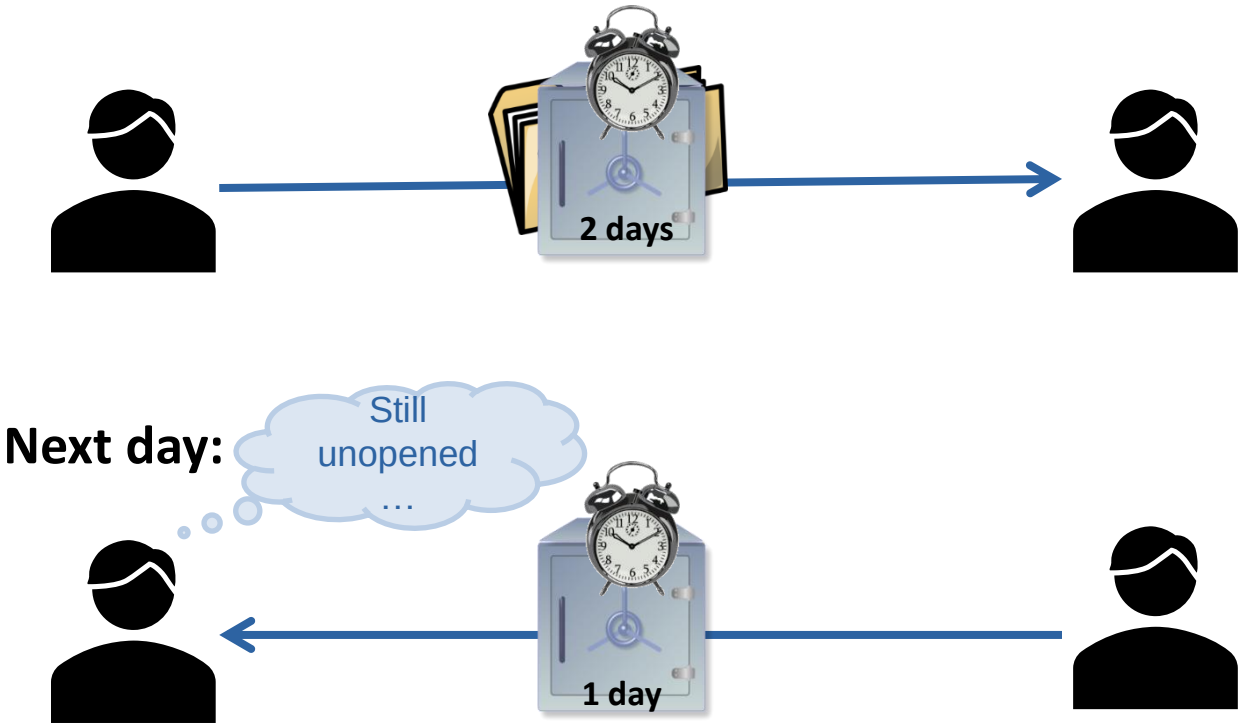
Speed of light

→ Position verified

Attack



Certified deletion



Tools

Easycrpt:

Established for crypto verification,
quantum mode broken.

Qrhl-tool:

Our development.
Still a long way to do.

Example derivation

$$\{x_1 = x_2\}$$
$$x \mathrel{+}= 1 \quad \sim \quad skip$$
$$\{x_1 - 1 = x_2\}$$
$$x \mathrel{+}= 1 \quad \sim \quad skip$$
$$\{x_1 - 1 - 1 = x_2\}$$
$$skip \quad \sim \quad x \mathrel{+}= 2$$
$$\{x_1 - 1 - 1 = x_2 - 2\}$$
$$= \{x_1 = x_2\}$$

$$\{x_1 = x_2\}$$
$$x \mathrel{+}= 1; x \mathrel{+}= 1$$
$$\sim x \mathrel{+}= 2$$
$$\{x_1 = x_2\}$$

Definition of qRHL (formal)

$$\{A\}\ \boldsymbol{c} \sim \boldsymbol{d}\ \{B\}$$

subspace of $\mathcal{H}_{mem} \otimes \mathcal{H}_{mem}$