Quantum one-time programs

CRYPTO 2013

Anne Broadbent

University of Waterloo

Gus Gutoski

Perimeter Institute

Douglas Stebila

Queensland University of Technology

One-time program for f

a.k.a. non-interactive secure 2-party computation

sender

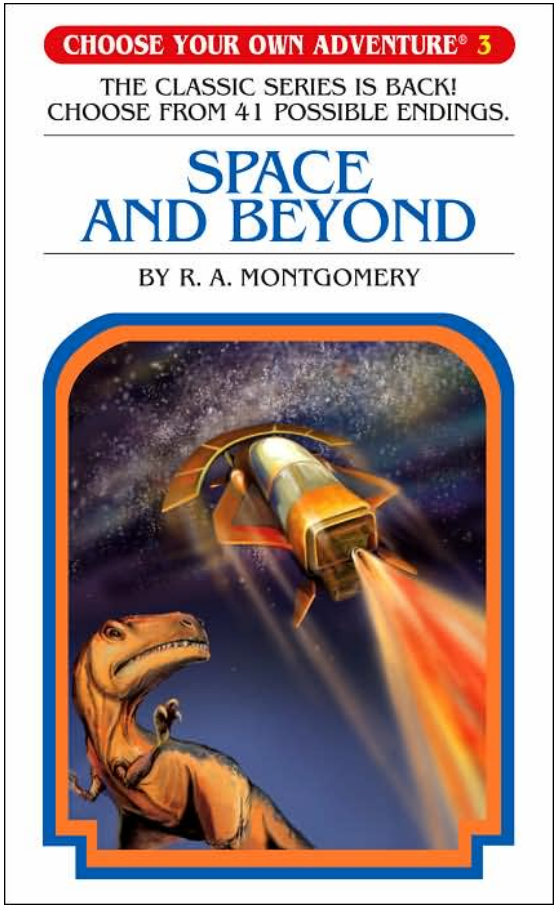
$x \longrightarrow$

receiver

$y \longrightarrow \qquad \longrightarrow f(x, y)$

Motivation

"This message will self-destruct in 5 seconds"



"If you break the lock and open the box, turn to page 23.

If you give up and go back to bed, turn to page 40."

Motivation

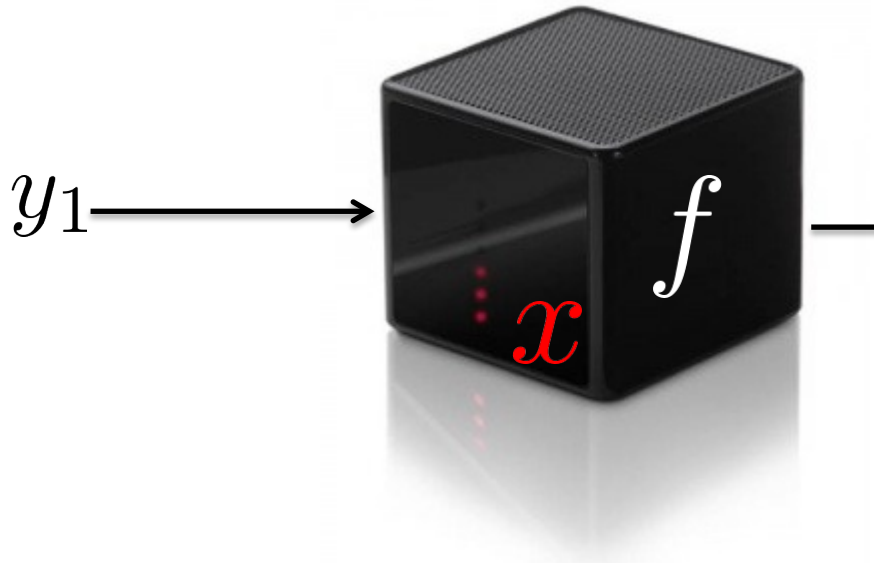
- Electronic cash
- Software copy protection
- Digital rights management

break the
d open the
rn to page

“This m
self-de
seconds”

ive up and
k to bed,
turn to page 40.”

Classical programs can be copied



Therefore, classical one-time programs are not possible in the plain model (even if we allow computational assumptions).

Hardware token model:

assume hardware tokens called one-time memories (OTMs)

sender

(s_0, s_1) ⟶

receiver

$i \in \{0,1\}$ ⟶ ⟶ s_i

Hardware token model:

assume hardware tokens called one-time memories (OTMs)

sender

Why use OTMs?

- Generic objects
- Independent of protocol
- Independent of input
- Could be mass-produced

receiver

$i \in \{0,1\}$ ⟶ ⟶ s_i

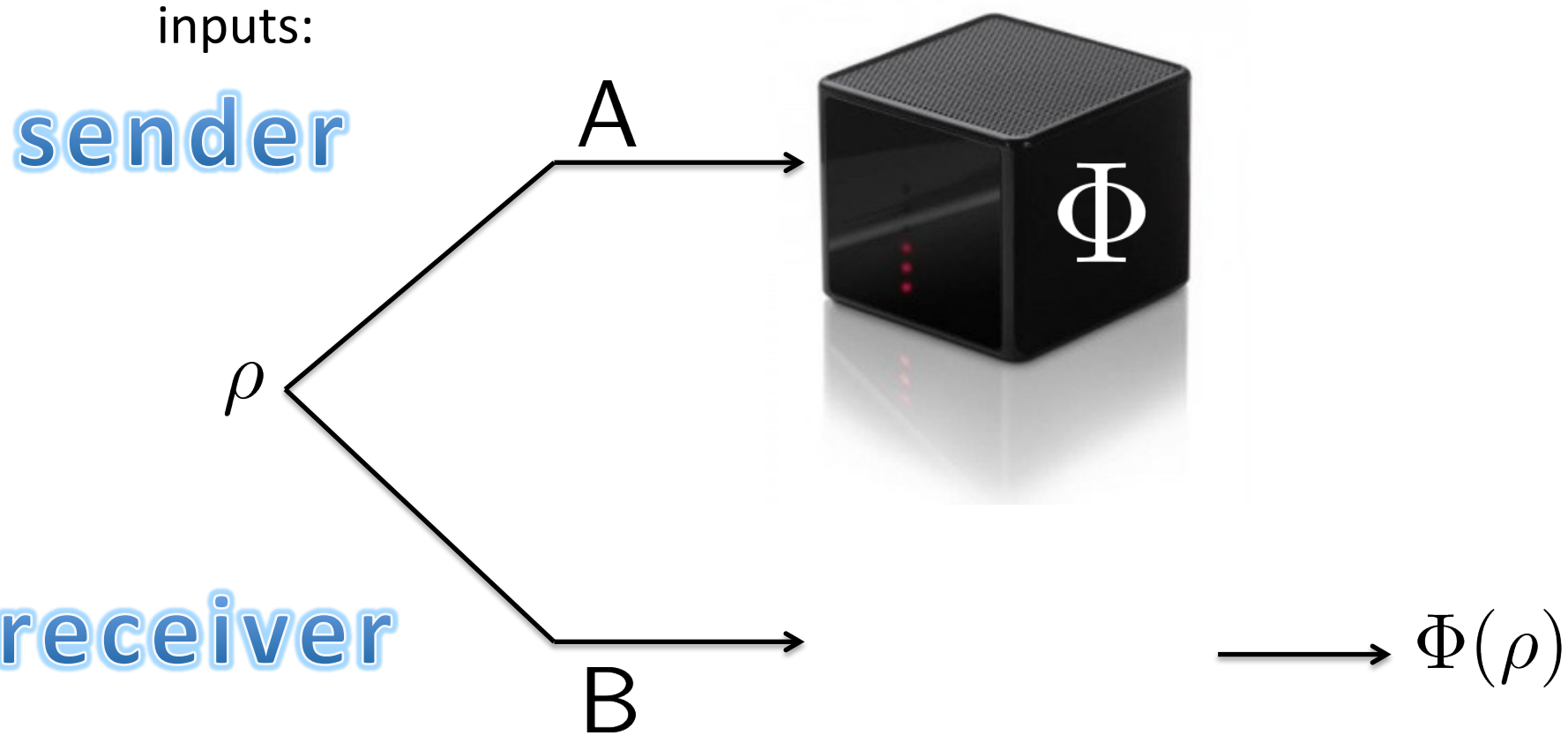
Classical one-time programs in the hardware token model

Goal: a compiler that transforms (f,x) into a one-time program.

1. Goldwasser, Kalai, G. Rothblum (CRYPTO '08): One-time programs in the string-OTM model,
   - computational security
   - standalone security definition
2. Goyal, Ishai, Sahai, Venkatesan, Wadia (TCC '10): One-time programs in the bit-OTM model,
   - statistical (information-theoretic) security
   - universal composability (UC) setting

Quantum twists

1. The no-cloning theorem prevents the basic copying attack. Could OTPs be possible in the plain quantum model?
2. OTPs for quantum channels need to handle entangled inputs:



Our questions / results

1. Does quantum information enable one-time programs for classical functions in the plain model?
   - NO! (for all but "trivial" functions)
2. Does quantum information enable one-time programs for quantum channels in the plain model?
   - NO! (for all but "trivial" channels)
3. Do quantum one-time programs exist for quantum channels in the bit-OTM model?
   - YES! (for all channels, with statistical UC security)
   - Main techniques:
     - new quantum authentication code
     - method to compute on authenticated data.

Related cryptographic tasks

1. Software copy-protection
   - Can be evaluated multiple times, but not "split" or "copied" into two parts that allow separate executions.
   - Clearly impossible with classical information alone
   - OTPs provide a solution
   - Aaronson (CCC '09): solution in the plain model using quantum information.
   - Open question: general quantum software copy-protection based on standard cryptographic assumptions.

2. Program obfuscation
   - Can be evaluated multiple times, but the "code" of the program does not leak any information beyond what can be learned by running the program.
   - Impossible with classical information alone (Barak, Goldreich, Impagliazzo, Rudich, Sahai, Vadhan, Yang, CRYPTO '01).
   - OTPs provide a solution
   - Open question: quantum program obfuscation (in the plain model).

1. IMPOSSIBILITY

Quantum one-time programs do not exist in the plain model

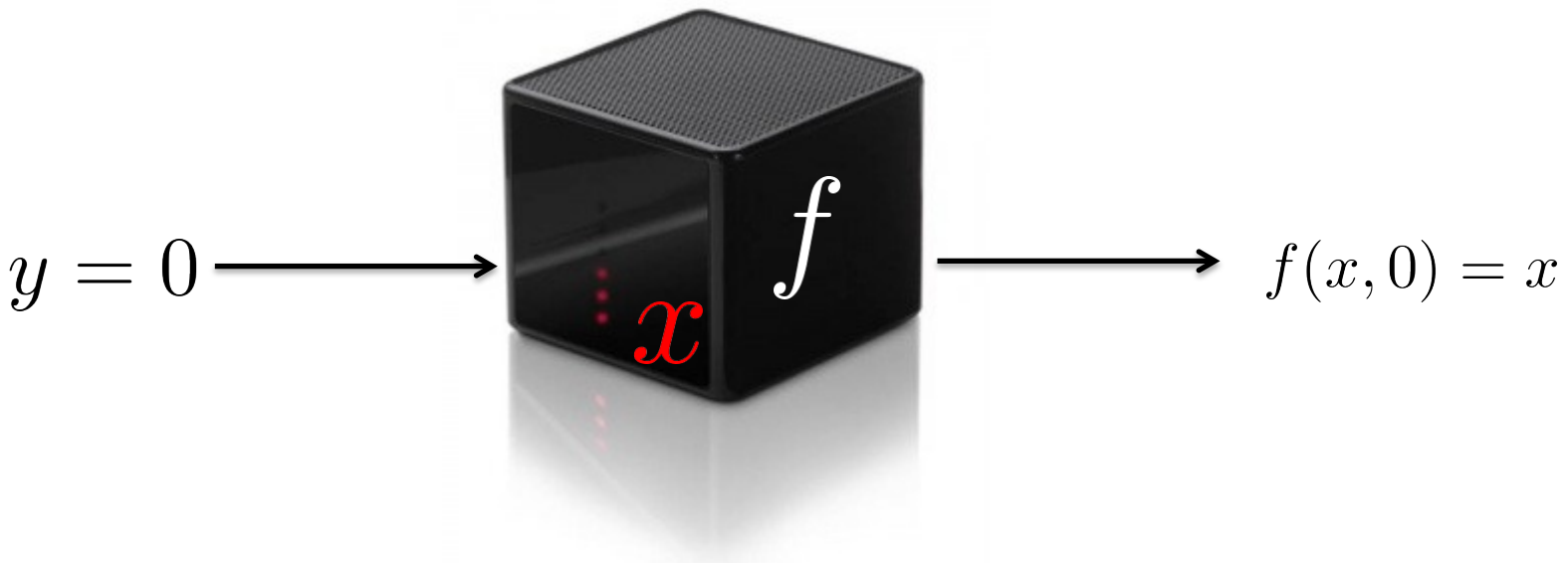
… except for some trivial cases

“Trivial” one-time programs

$$f(x, y) = x + y$$

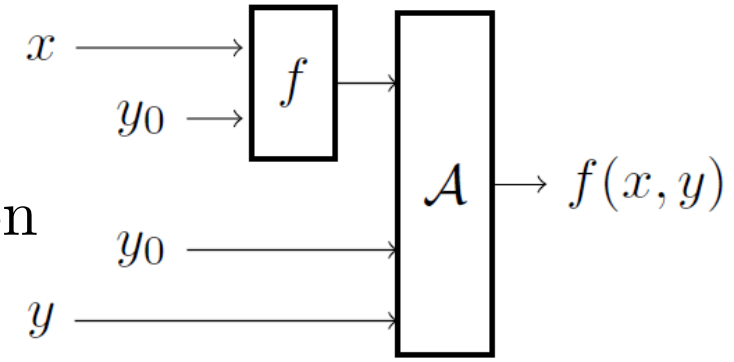
One-time program for f: sender reveals x.

This is “secure” because a single query to $f(x, \cdot)$ will also reveal x.



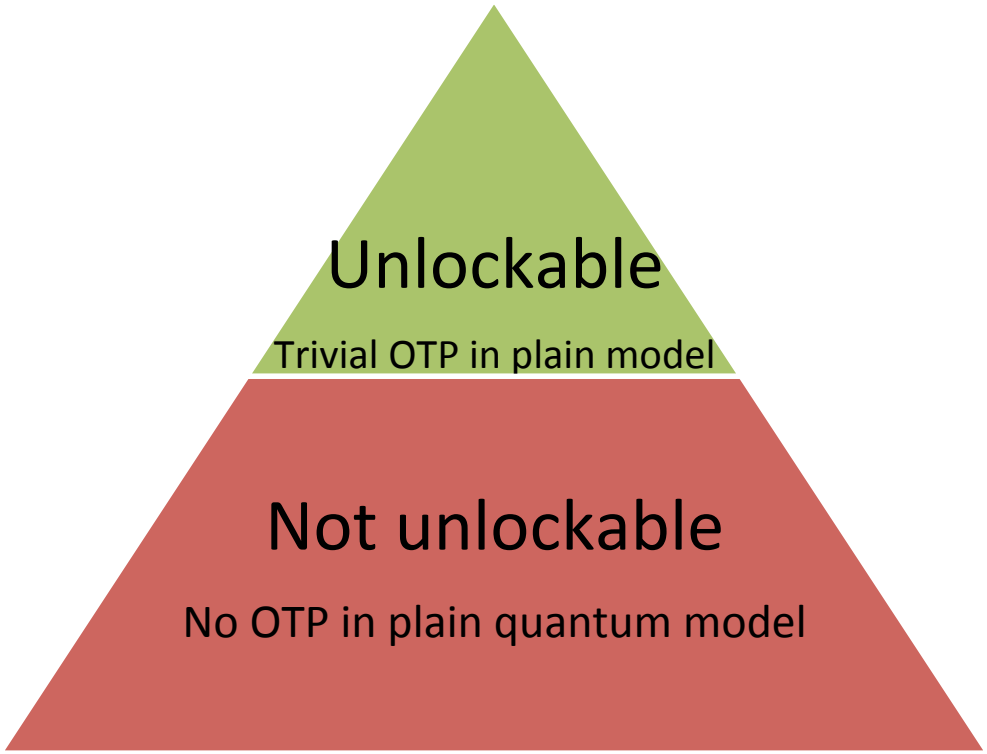
Unlockable functions

A function f is *unlockable* if there exists a key input y_0 and a recovery algorithm $\mathcal{A}$ that allows computation of $f(x, y)$ for any y.

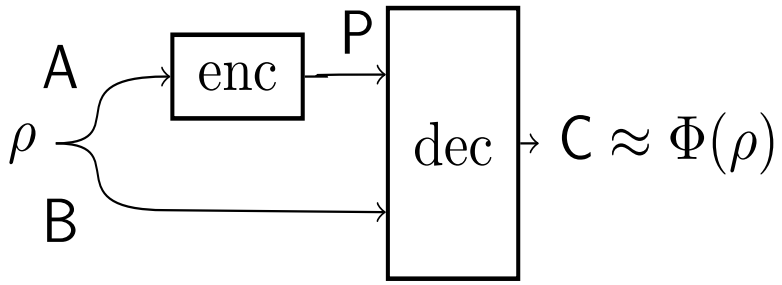


Theorem: If f is unlockable, then f has a secure OTP in the plain classical model.

Theorem: If f has a secure OTP in the plain quantum model, then f is unlockable.



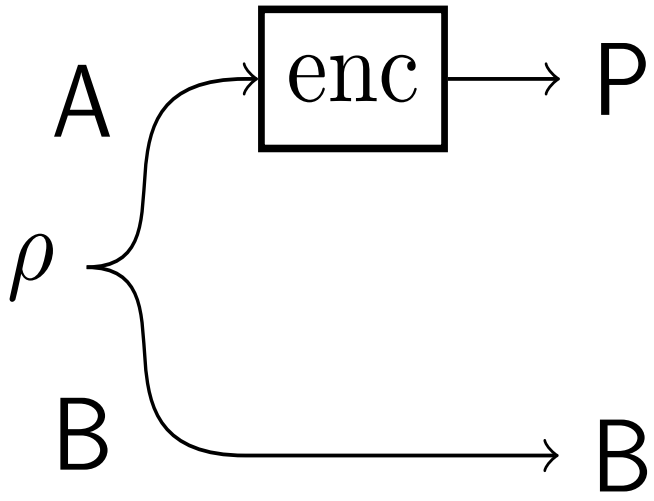
Definition of quantum OTP



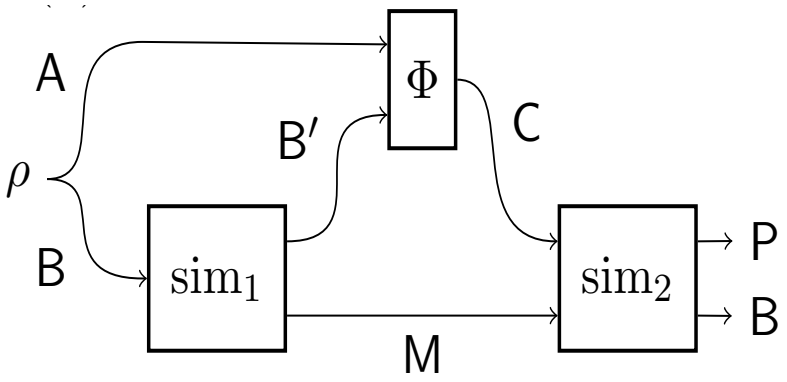
- Φ: public channel
- $\rho = (\mathsf{A}, \mathsf{B})$: sender & receiver input
- enc: creates program state for sender's input
- P: program state
- dec: run program state with receiver's input
- C: output

Security of OTP

Real world



Ideal world

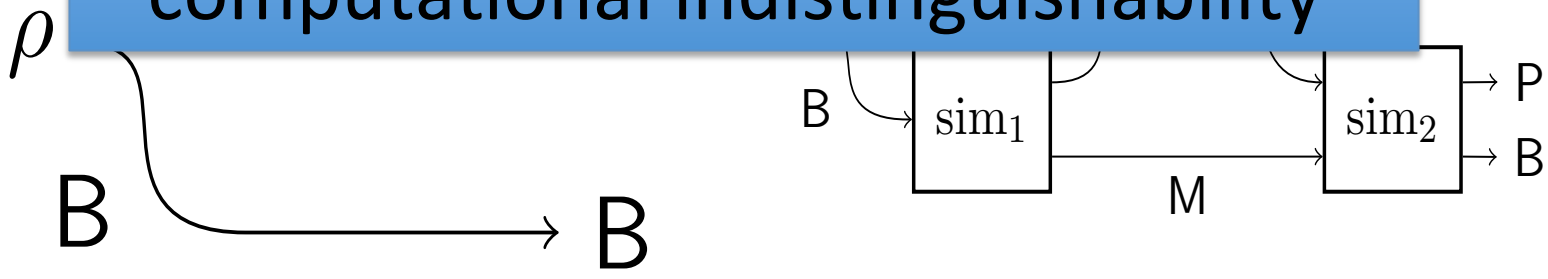


In this model, a protocol is secure if the joint state of registers (P,B) (before dec is applied) can be re-created by a simulator (sim_1, sim_2) that has one-shot access to channel Φ.

Security of OTP

Real world **Ideal world**

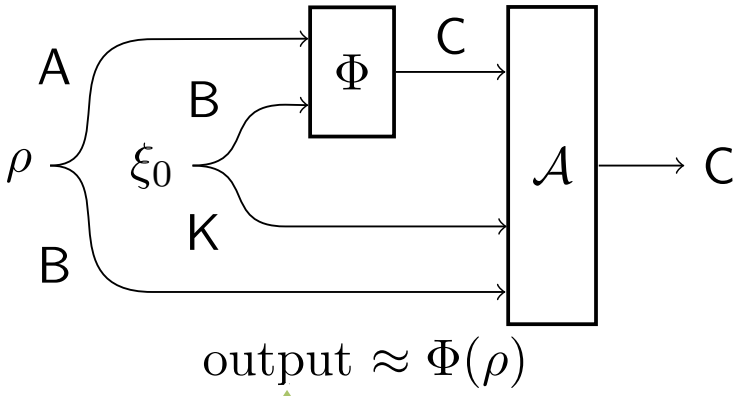
Definition in UC framework
- talk focuses on perfect case
- results hold for statistical and computational indistinguishability



In this model, a protocol is secure if the joint state of registers (P,B) (before dec is applied) can be re-created by a simulator (sim_1, sim_2) that has one-shot access to channel Φ.

Unlockable channels

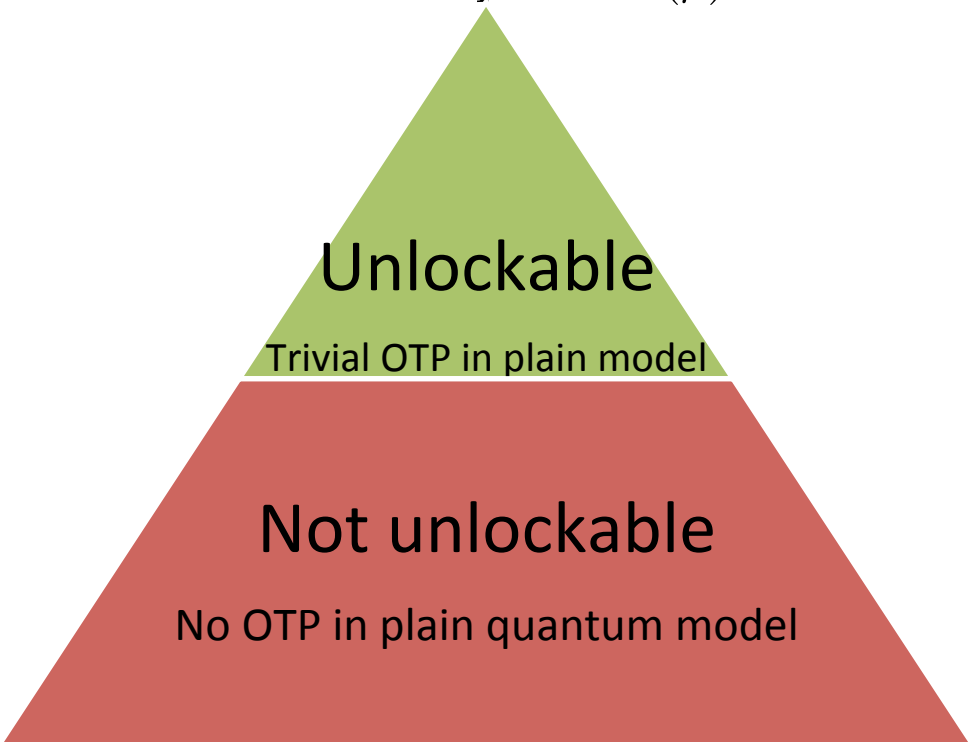
A channel Φ is *unlockable* if there exists a key state ξ_0 and a recovery algorithm $\mathcal{A}$ that allows computation of $\Phi(\rho)$ for any ρ.



Theorem: If Φ is unlockable, then it has a secure QOTP in the plain quantum model.

Theorem: If Φ has a secure QOTP in the plain quantum model, then it is unlockable.

(tighter result than in proceedings version)



2. POSSIBILITY

All quantum channels admit a UC-secure quantum one-time program in the classical one-time memory model.

Overview: OTPs for quantum channels in the OTM model

Main idea: "tamper-proof" computation

The QOTP includes

- the sender's input
- some auxiliary qubits

encoded in a "tamper-proof" but malleable way:

- the receiver is allowed to perform gates on the encoded data.

At the end, the receiver gets the output as long as he performed the sequence of gates as instructed.

Main tools

quantum authentication codes

= "tamper-proof encoding"

Uses a classical key; detects tampering with high probability.

Quantumly, authentication implies encryption.

- Barnum, Crépeau, Gottesman, Smith, Tapp (FOCS 2002)

quantum computing on authenticated data (QCAD)

= performing gates on "tamper-proof" encodings.

QCAD normally requires classical interaction with the sender; we substitute this with a classical, UC-secure OTP as given by prior work.

Quantum authentication codes

We use an encode + Pauli encrypt scheme.

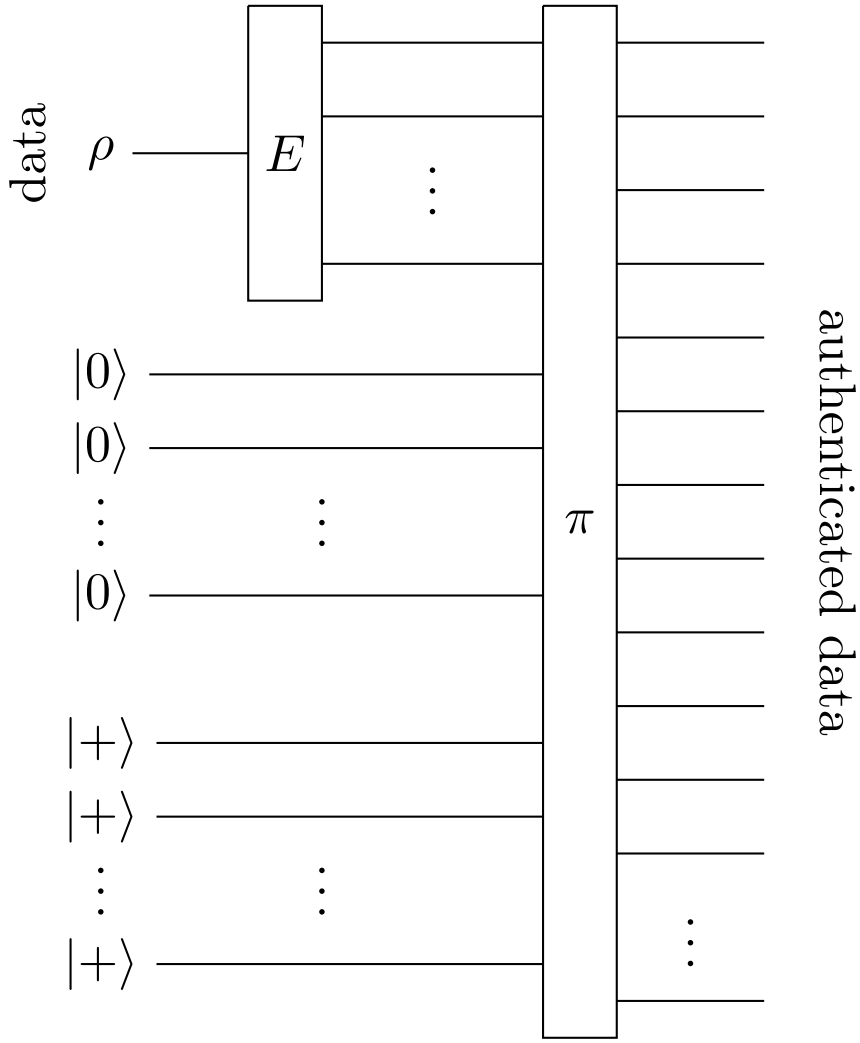
- Pauli encryption maps an arbitrary attack into a mixture of Pauli attacks (Pauli twirl)
- So all we need is a family of codes that is secure against Pauli attacks.

Trap authentication code

Let *E* be self-dual CSS code of distance *d*, encoding 1 logical qubit into *n* physical qubits.

Theorem: The family of trap codes is $(2/3)^{d/2}$–secure against Pauli attacks.

(Trap codes first used implicitly by Shor and Preskill (PRL '00).)

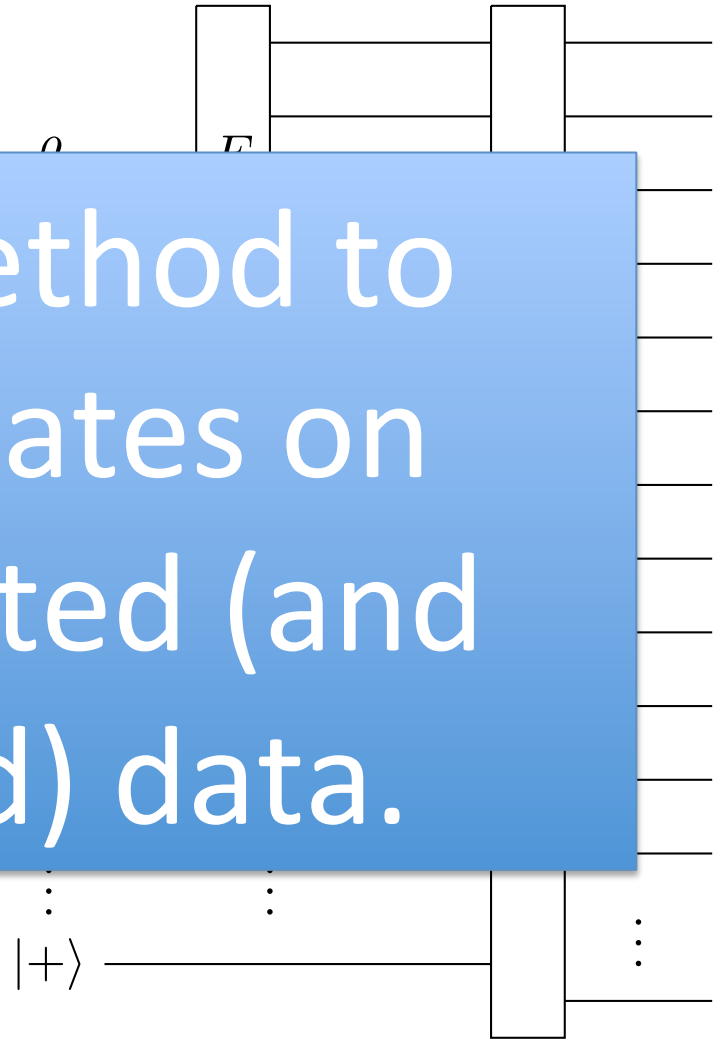


Trap authentication code

Let *E* be self-dual CSS code
of di
logic
qubi

Thec
code
agai

(Tra
implicitly by Shor and
Preskill (PRL '00).)



Need a method to perform gates on authenticated (and encrypted) data.

Gadgets for trap code universal QCAD

Techniques inspired by fault-tolerant quantum computation.

QCAD originally established for the signed polynomial authentication code

- (Ben-Or, Crépeau, Gottesman, Hassidim, Smith, FOCS 2006)

Also known for the Clifford authentication code

- (Dupuis, Nielsen, Salvail CRYPTO '12)

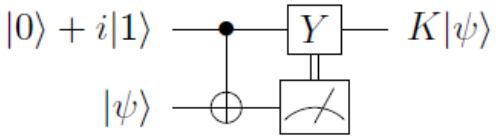
1. Measurement:
   - computational basis measurement of logical data
   - = qubit-wise measurements of physical data + classical decoding
2. Pauli gates:
   - receiver does nothing
   - sender updates the Pauli encryption key

Gadgets for trap code universal QCAD

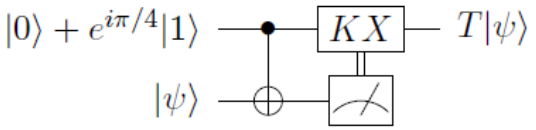
3. CNOT:
   - Bitwise CNOT
   - Simple Pauli key updates

$$\text{CNOT} : |0\rangle|0\rangle \mapsto |0\rangle|0\rangle$$
$$: |+\rangle|+\rangle \mapsto |+\rangle|+\rangle$$

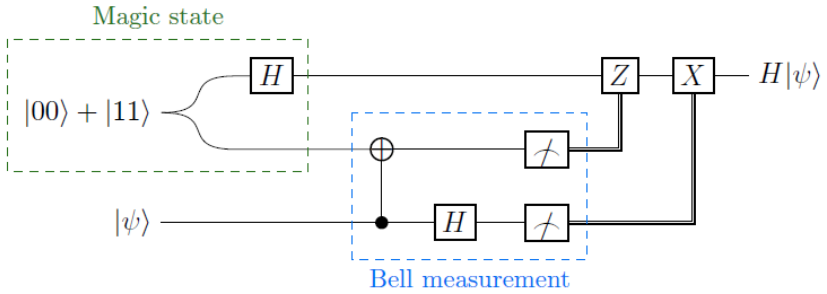
4. *i* gate
   - Auxiliary (magic state) prepared by the sender
   - one-way communication to the sender required



5. π/8 gate
   - Like *i*-gate, but sender decodes result and returns it to the receiver.



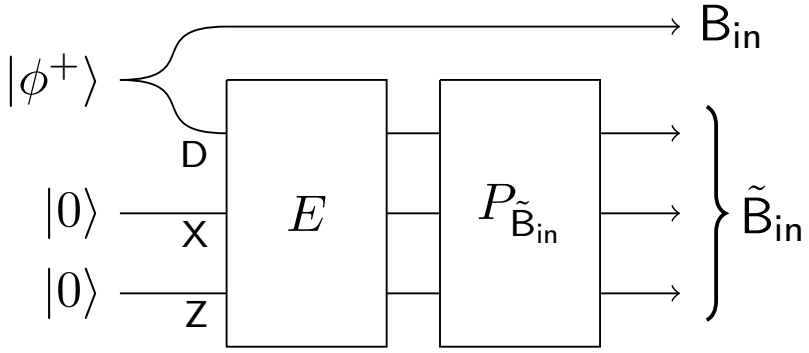
6. Hadamard
   - Use gate teleportation (Gottesman and Chuang)



Encoding and decoding gadgets

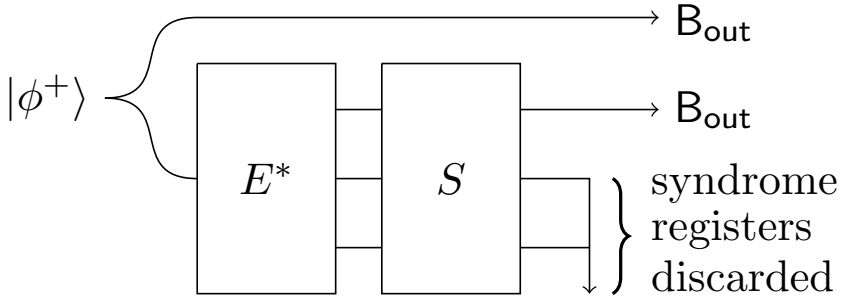
How does the receiver get an authenticated version of his input?

- Use gate teleportation!



How does the received get an unauthenticated version of the output?

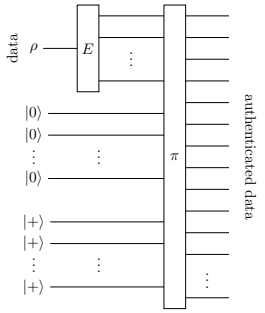
- Use gate teleportation!



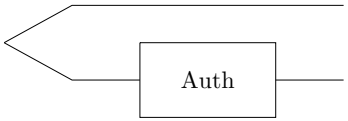
Protocol for QOTP for Φ

To prepare a QOTP:

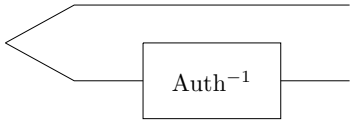
- sender's input encoded



- encoding gadget



- decoding gadget



- classical OTP implementing interaction for QCAD

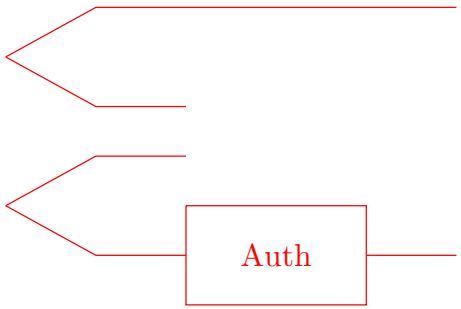
To use a QOTP:

1. Teleport receiver's input through encoding gadget.
2. Perform target circuit using QCAD.
3. Teleport receiver's output through decoding gadget.
4. All classical interaction is done via the classical OTP.

Simulator for security proof

Simulator prepares fake QOTP

- sender's input encoded
- encoding gadget
- decoding gadget
- classical OTP implementing interaction for QCAD



Simulator runs the protocol

1. Extract receiver's input using the first half of "encoding gadget".
2. Use this input as input into the ideal functionality.
3. Teleport the output of the ideal functionality through second half of "encoding gadget".
4. Continue protocol as in the real world, ensuring same output occurs in real and ideal setting.

Simulator for security proof

Proof:
The final states held by the environment in the real and ideal world are close in trace distance.

Proof applies to any encode-encrypt authentication scheme that admits QCAD.

Simulator runs the protocol

1. Extract receiver's input using the first half of "encoding gadget".
2. Use this input as input into the ideal functionality.
3. Teleport the output of the ideal functionality through second half of "encoding gadget".
4. Continue protocol as in the real world, ensuring same output occurs in real and ideal setting.

Summary

1. Quantum information does not allow for QOTPs of classical functions or quantum channels in the plain model.
   - except for trivial "unlockable" functions

2. UC-secure protocol for QOTPs for quantum channels in the classical bit one-time memory (OTM) model.
   - new quantum authentication code: "trap scheme"
   - method to compute on authenticated data

Open question: possibility/impossibility of quantum program obfuscation.