Quantum OTPs

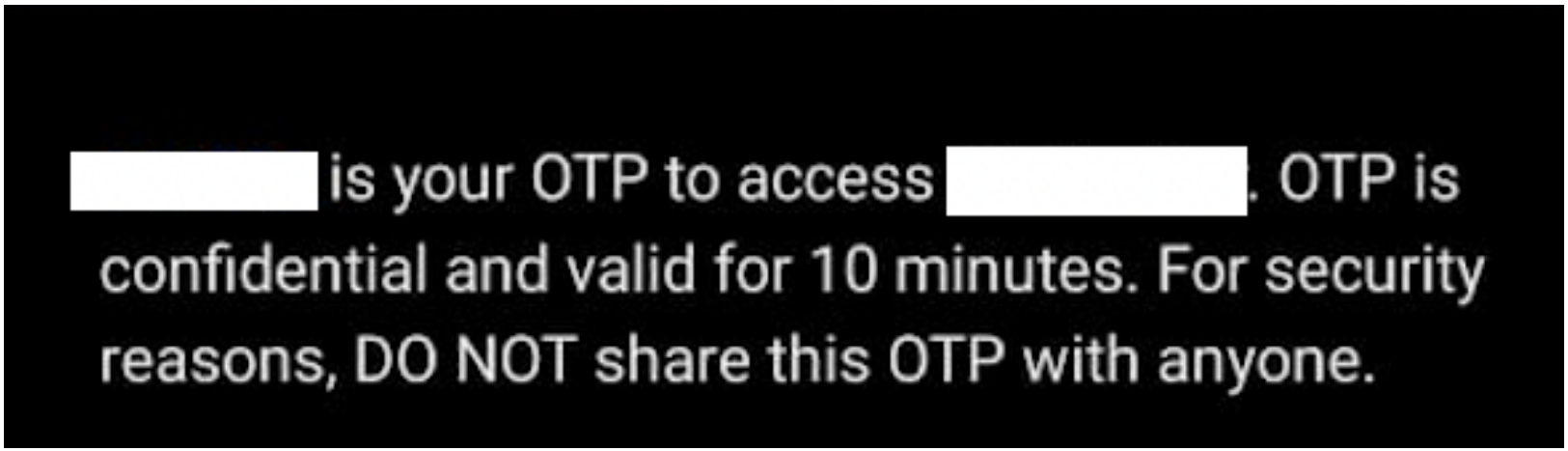
Ishan Mankodi, Siddhartha Chaganti

Indian Institute of Technology Madras

3rd November 2022

OTP

- ~~One Time Password~~



OTP

- ~~One Time Password~~
- ~~One Time Pad~~

OTP

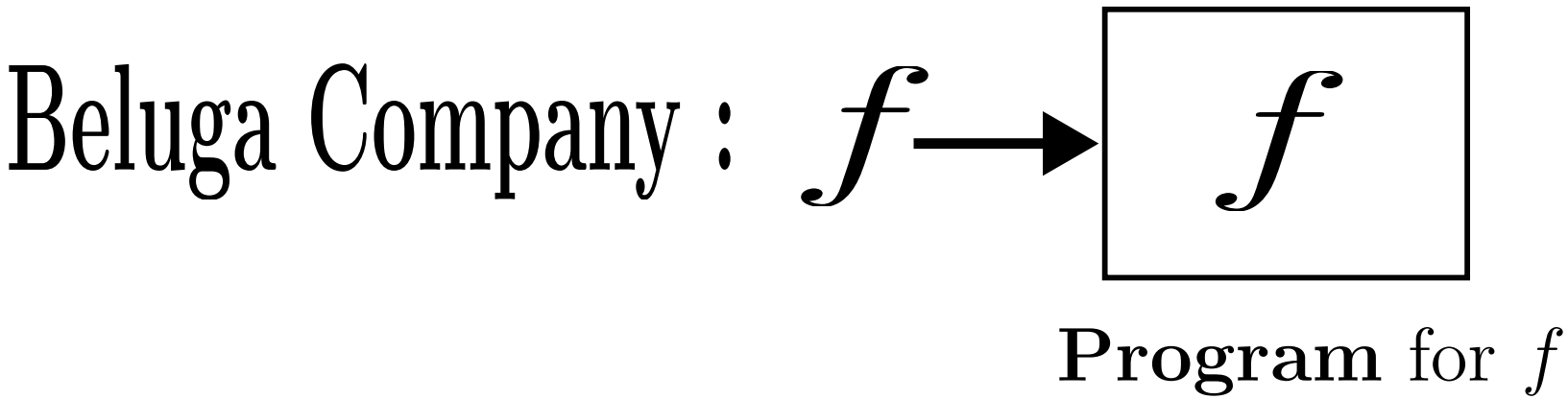
- ~~One Time Password~~
- ~~One Time Pad~~
- One Time Program

What is a one-time program?

An OTP for a function

$$f : \{\text{bit strings}\} \rightarrow \{\text{bit strings}\}$$

is a cryptographic primitive by which a user evaluates f on one input x chosen at run-time.

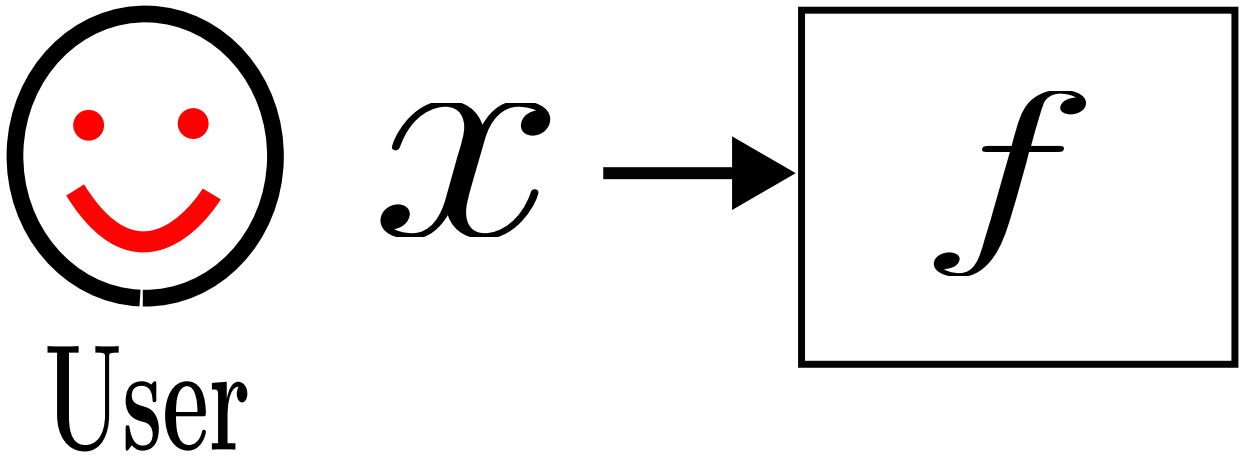


What is a one-time program?

An OTP for a function

$$f : \{\text{bit strings}\} \rightarrow \{\text{bit strings}\}$$

is a cryptographic primitive by which a user non-interactively evaluates f on one input x chosen at run-time.

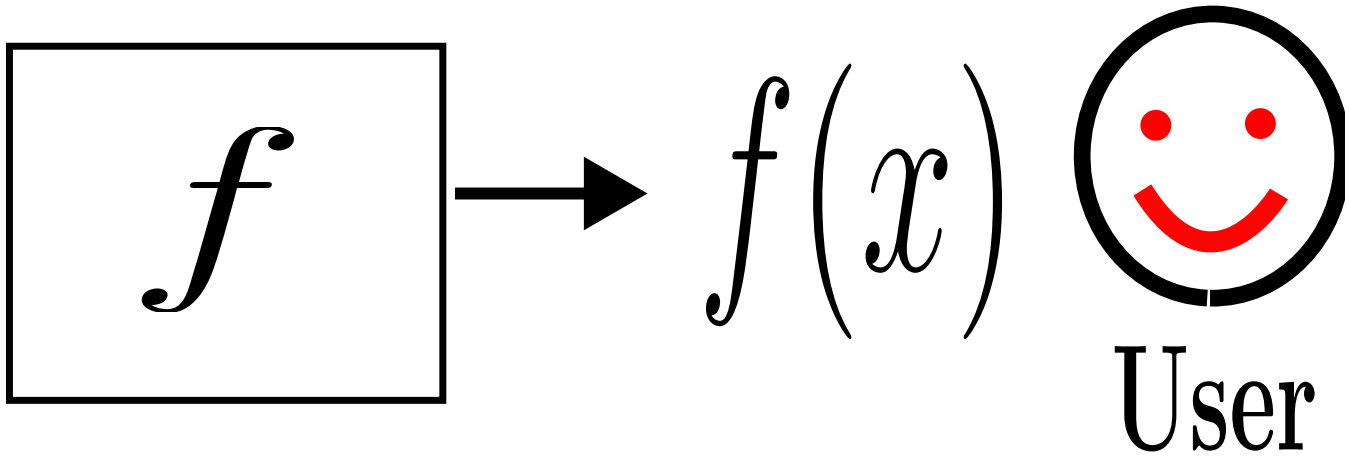


What is a one-time program?

An OTP for a function

$$f : \{\text{bit strings}\} \rightarrow \{\text{bit strings}\}$$

is a cryptographic primitive by which a user non-interactively evaluates f on one input x chosen at run-time.

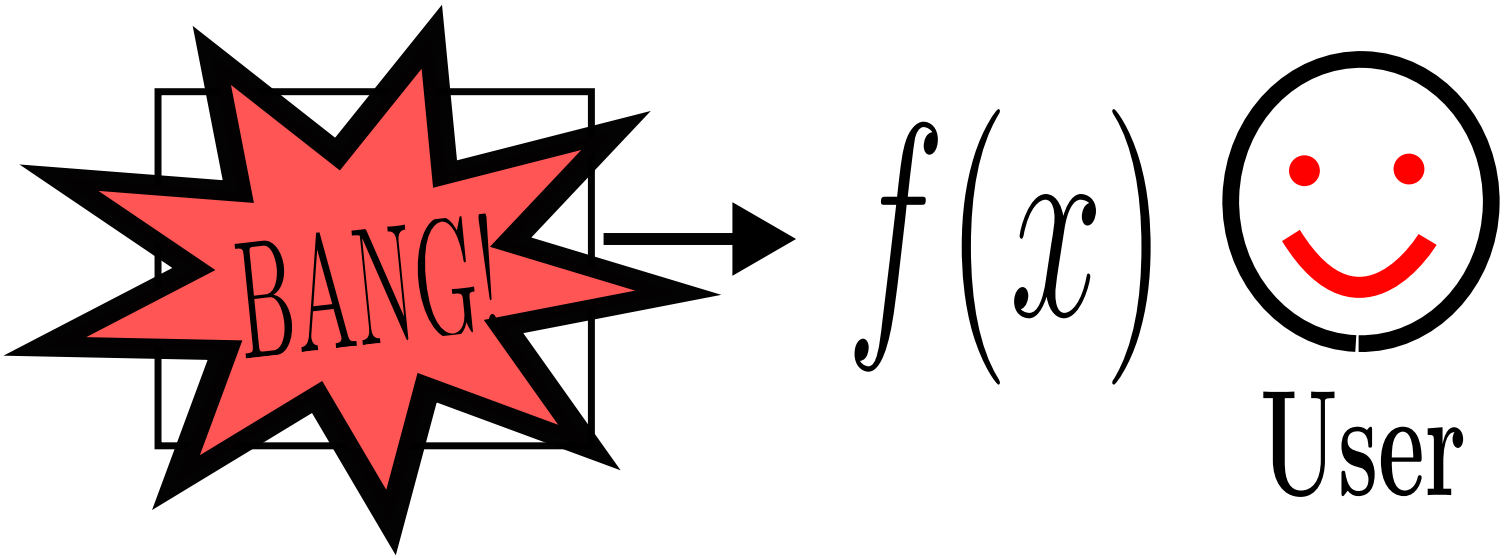


What is a one-time program?

An OTP for a function

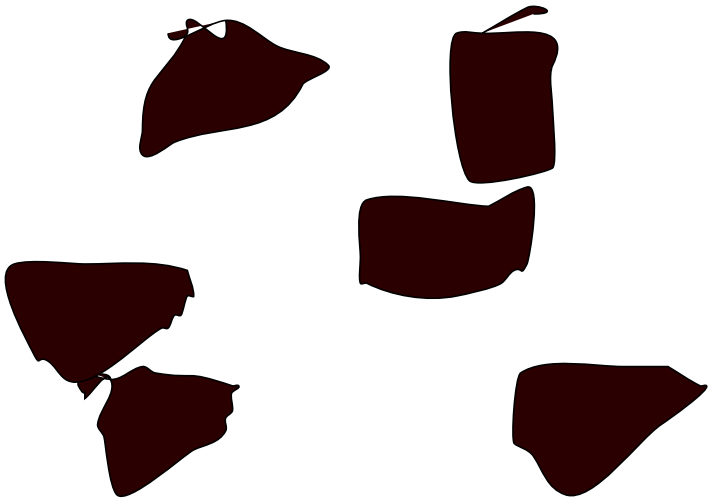
$$f : \{\text{bit strings}\} \rightarrow \{\text{bit strings}\}$$

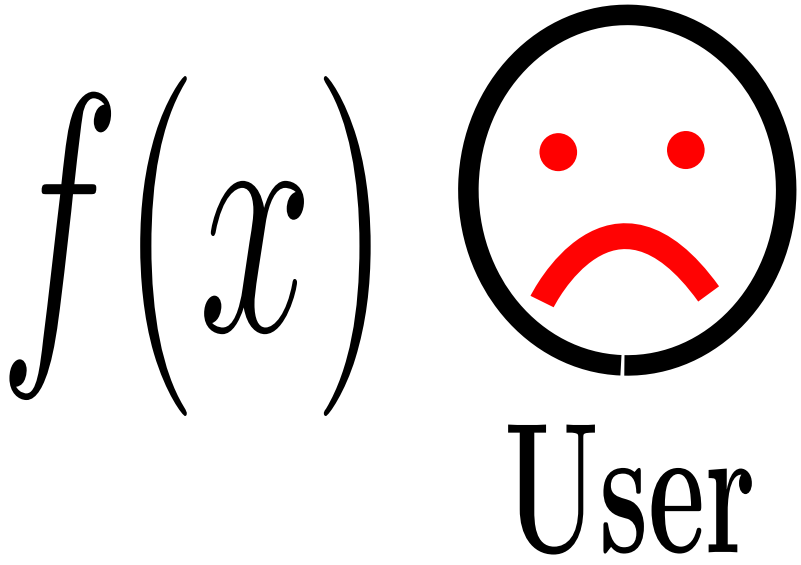
is a cryptographic primitive by which a user non-interactively evaluates f on one input x chosen at run-time.



What is a one-time program?

No user, after evaluating $f(x)$, should be able to learn anything about $f(x')$ for any $x' \neq x$ beyond that which can be learned from $f(x)$.





OTPs cannot be achieved by software

Software can be copied and re-run:

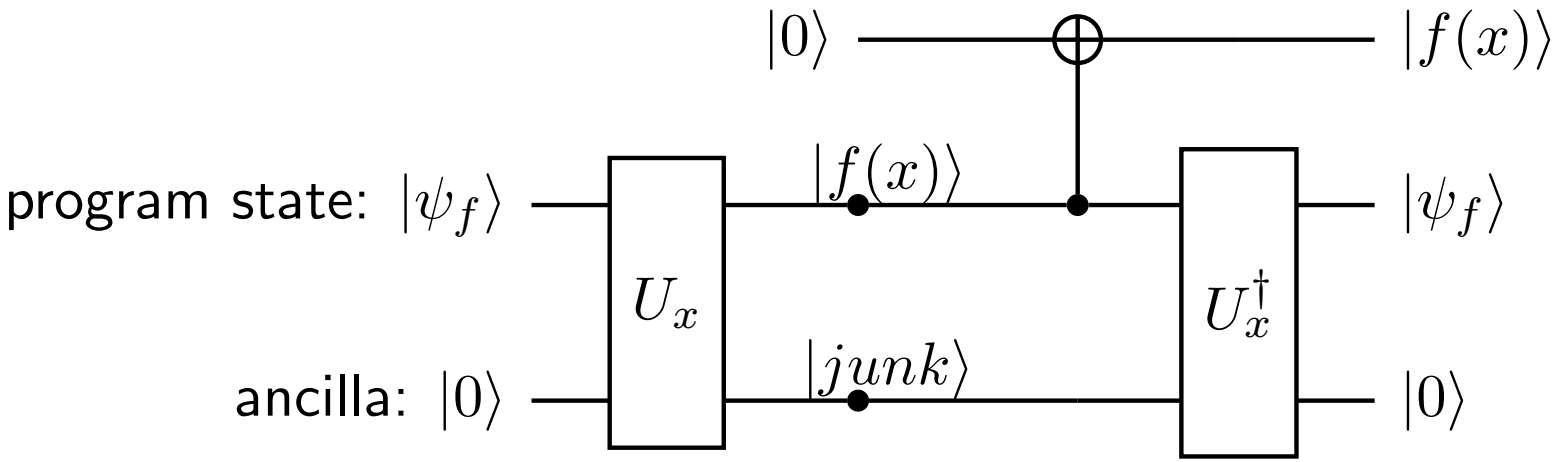
If $\boxed{f}$ is simply string $enc(f)$ then there is no way to ensure that $enc(f)$ self-destructs after one use.

$\Longrightarrow$ need additional assumptions such as:

- Quantum information
- Secure Hardware

Does quantum information suffice?

No. A reversible adversary can always recover a "program state" $|\psi_f\rangle$and evaluate f multiple times.

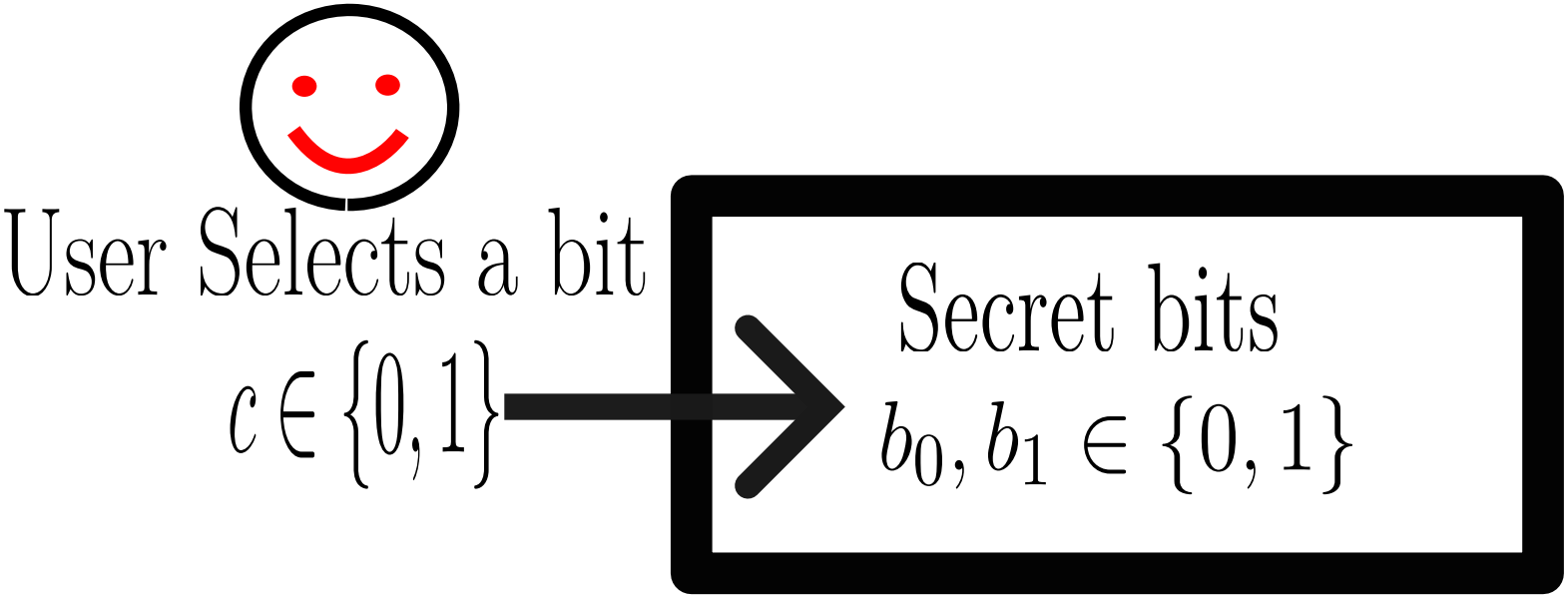


What about secure hardware?

Yes, but we must be careful not to assume the whole problem away!

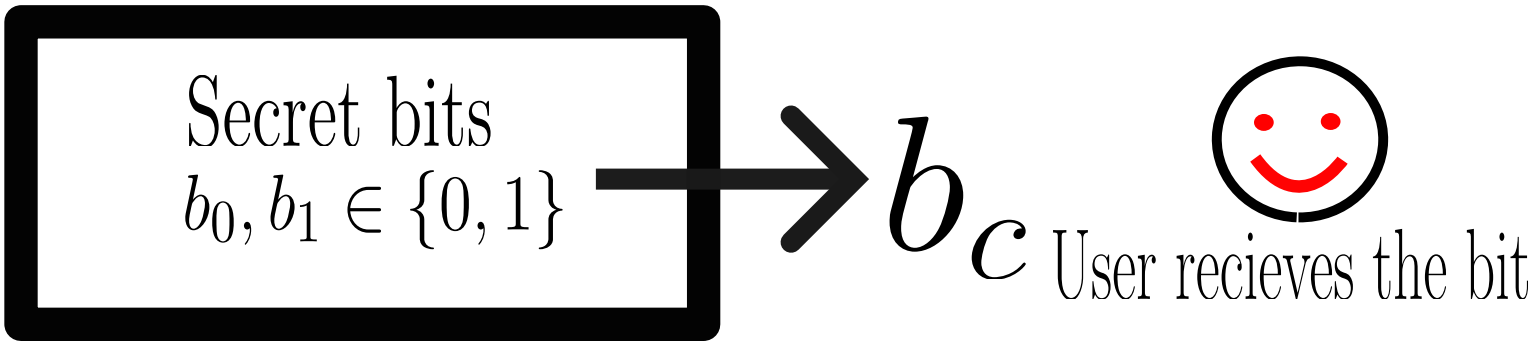
Secure hardware: one-time memory

Classical one-time programs can be constructed from very basic hypothetical hardware devices.

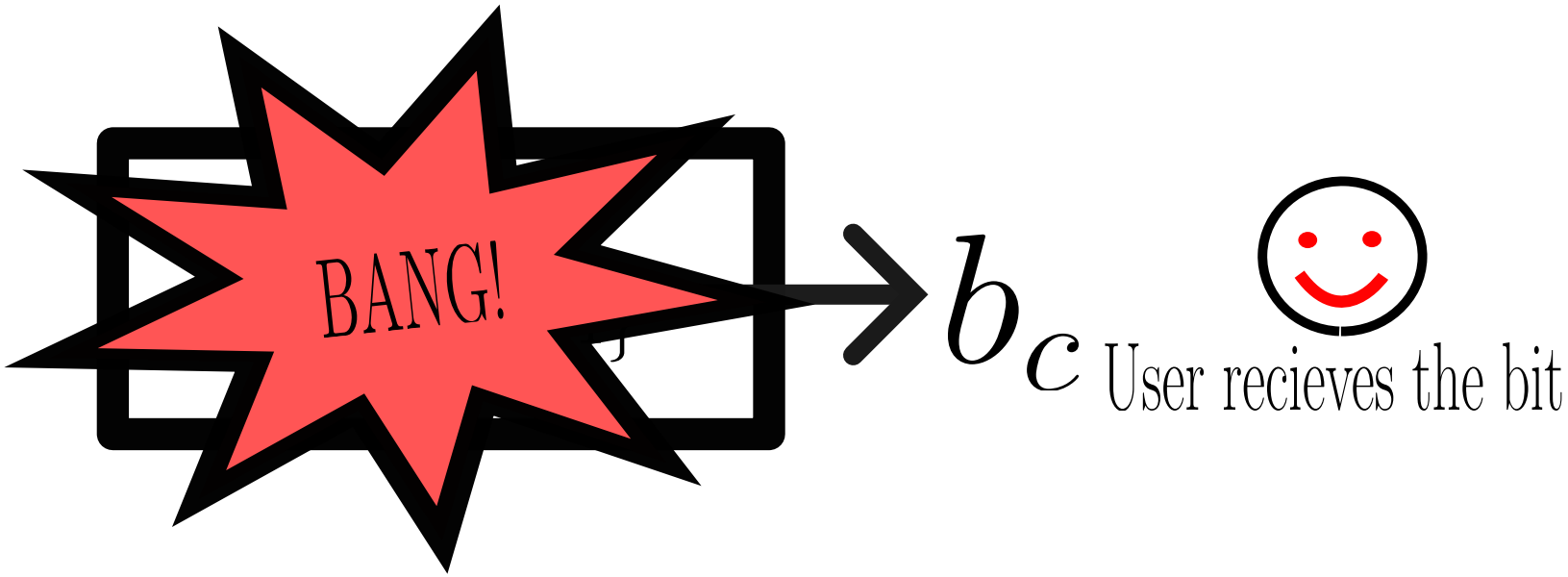


[Goldwasser, Rothblum, Kalai 2008] [Goyal, Ishai, Sahai, Venkatesan, Wadia, 2010] [Bellare, Hoang, Rogaway 2012]

Secure hardware: one-time memory

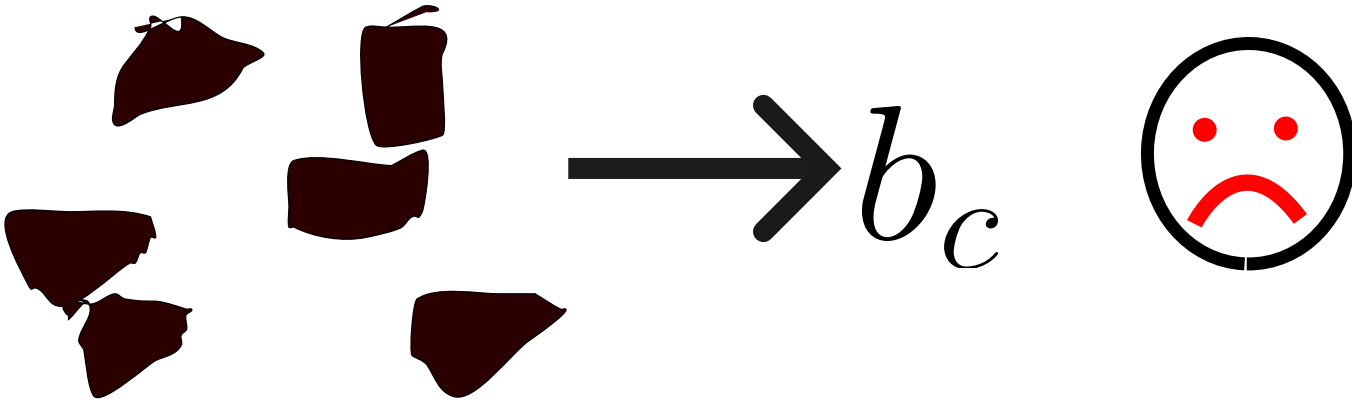


Secure hardware: one-time memory



Secure hardware: one-time memory

The value of the other bit $b_{\bar{c}}$ is lost forever.



OTM = secure non-interactive oblivious transfer.

Advantages of one-time memories

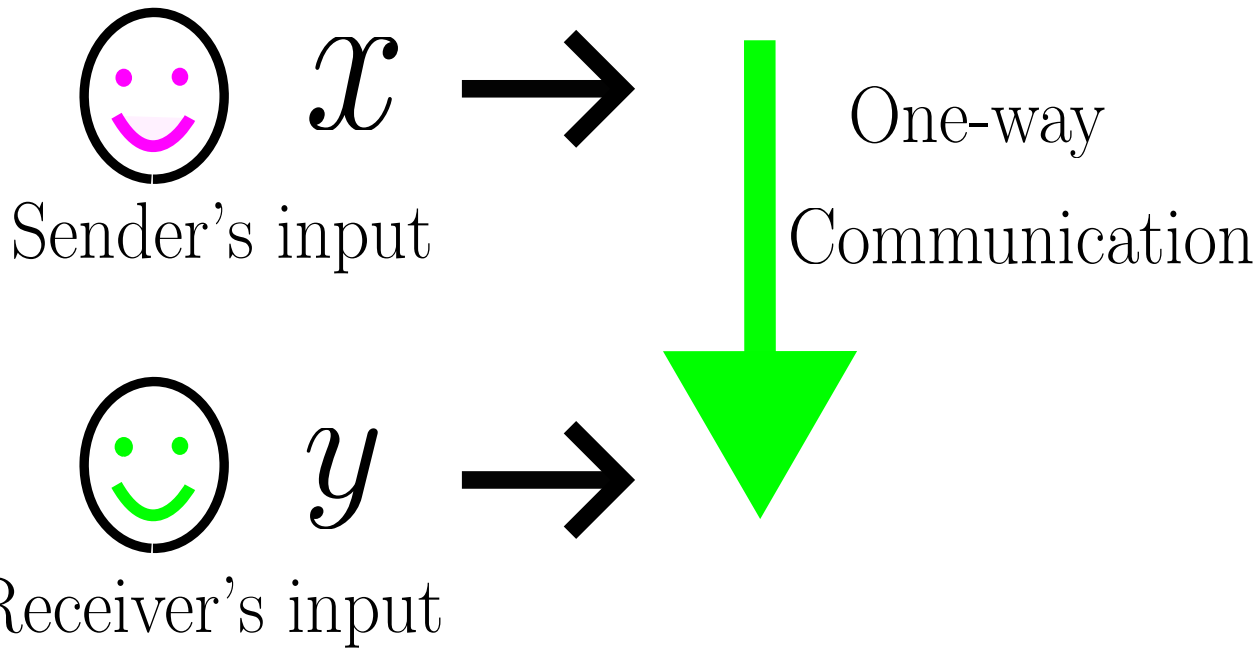
Extremely simple. Easier to avoid hardware flaws.

Can be mass produced. OTMs are independent of any specific **program** f .

Non-interactive two party computation

A slight generalization of OTPs: two parties (sender, receiver) wish to non-interactively evaluate a public known function

$$g : \{\text{bit strings}\} \times \{\text{bit strings}\} \to \{\text{bit strings}\}$$



Non-interactive two party computation

y

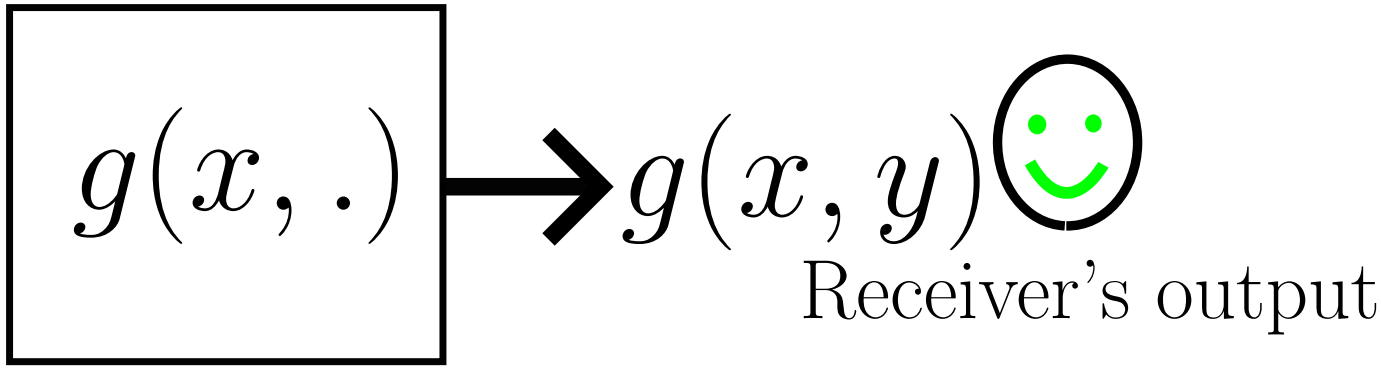
Receiver's input

$\rightarrow$

$g(x, .)$

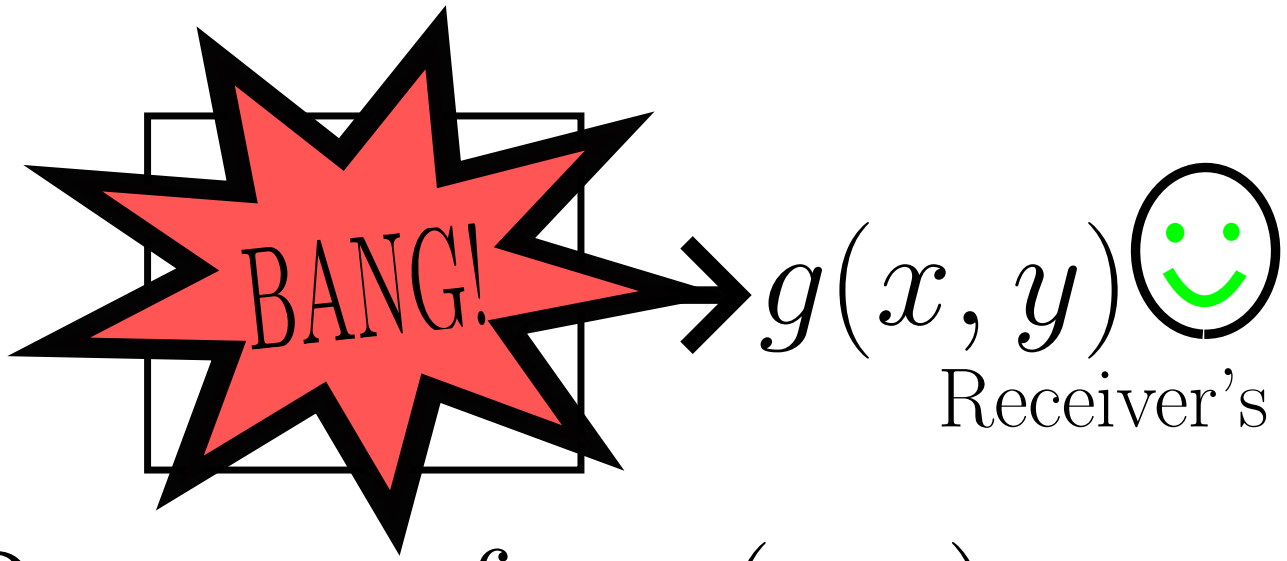
Program for $g(x, .)$

Non-interactive two party computation



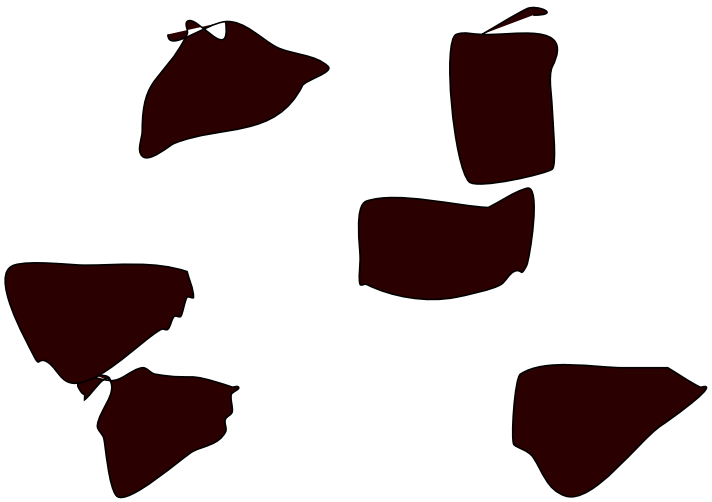
Program for $g(x, .)$

Non-interactive two party computation



Non-interactive two party computation

Receiver learns nothing beyond that which can be inferred from one-shot access to an oracle for $g(x, .)$.



$g(x, y)$

Why NI2PC?

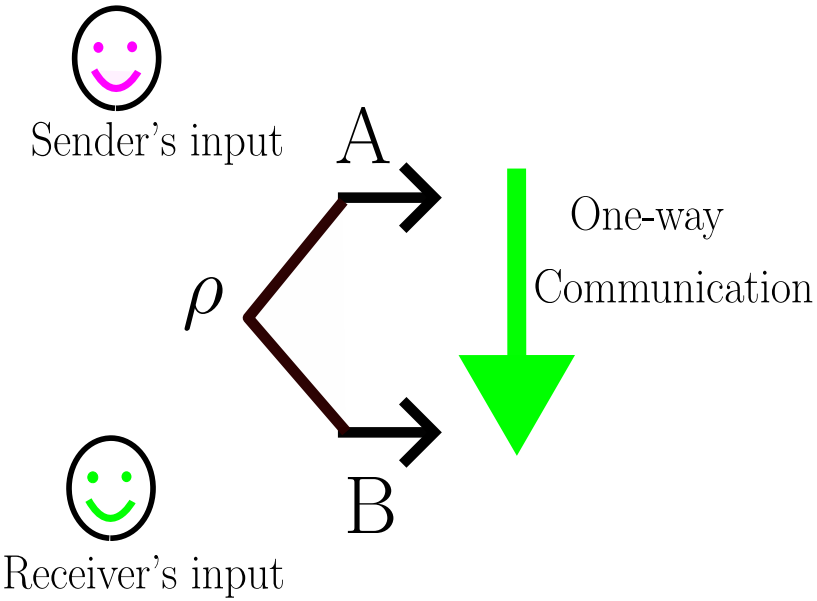
- Two-party computation is a familiar primitive.
- OTPs arise as a special case when g is **Universal computer**:

$$g : (enc(f), x) \rightarrow f(x)$$

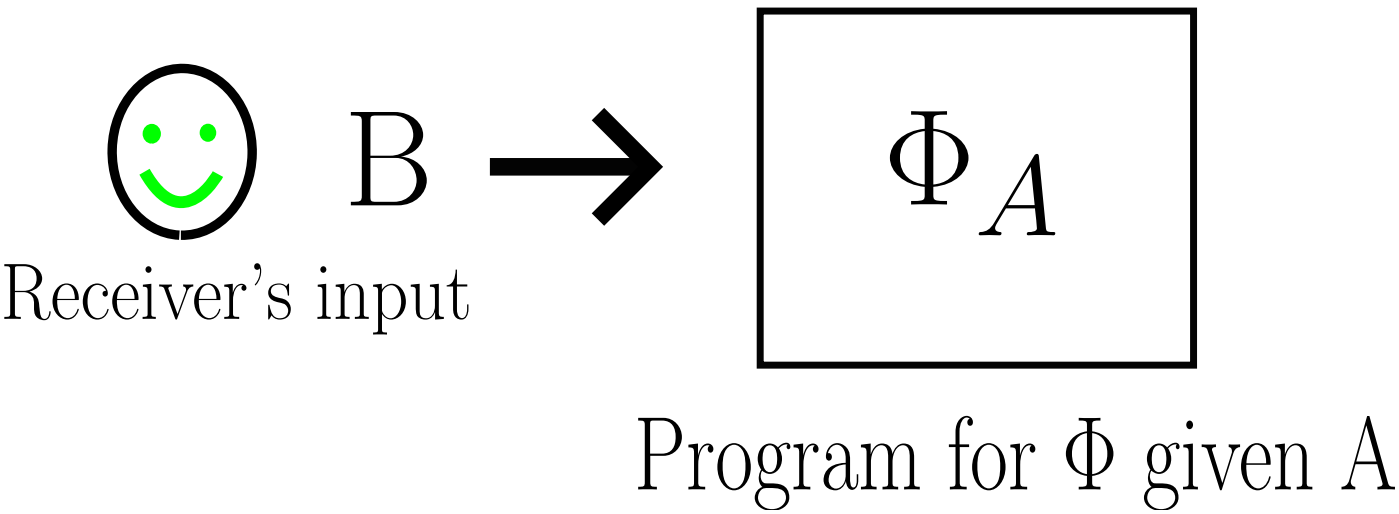
- All results apply to NI2PC anyway.
- For the rest of this talk,
  One-time program $=$ non-interactive two-party computation.

Quantum One-time programs

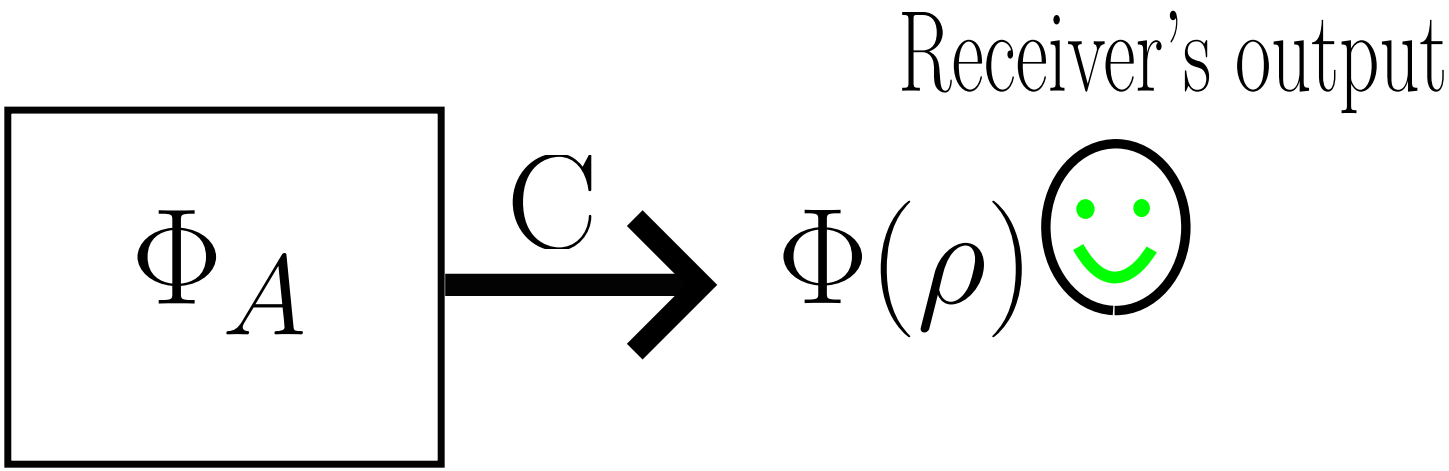
Evaluate a publicly known channel $\Phi : (A, B) \to C$ (specified by a quantum circuit).



Quantum One-time programs

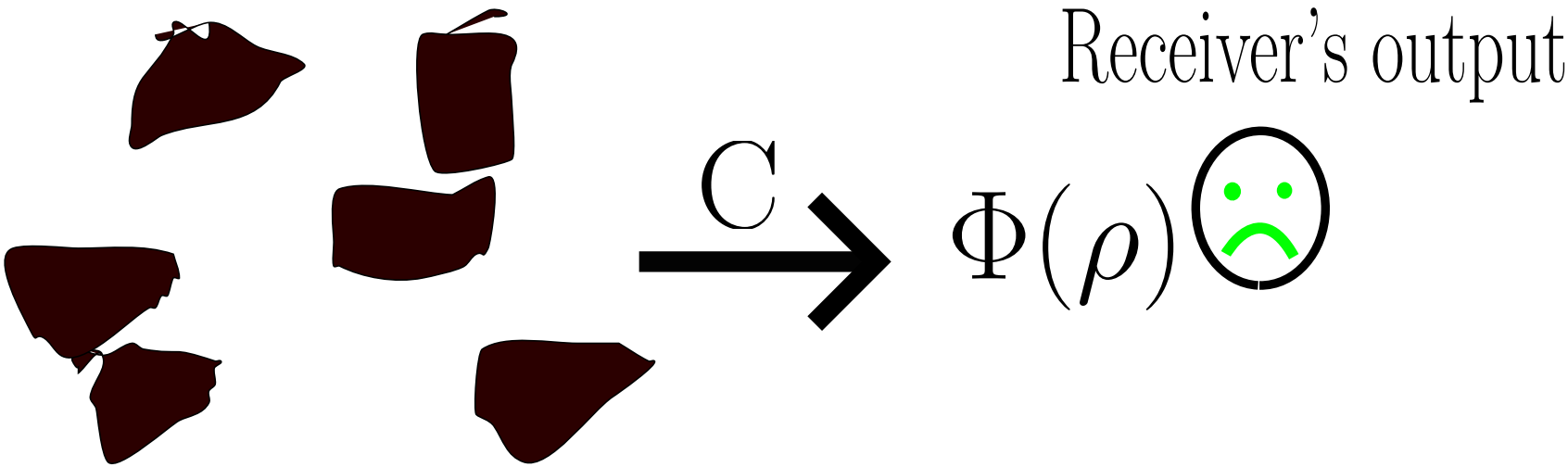


Quantum One-time programs



Quantum One-time programs

Receiver learns nothing beyond that which can be inferred from one-shot access to an oracle for Φ given A.



What's known about classical OTPs

$\exists$ classical one-time programs for any function f using one-time memories that are:

1. Secure against any malicious receiver. (No restrictions required).
2. Secure against any malicious sender.
3. Universally composable. (Secure even against parallel attacks.)

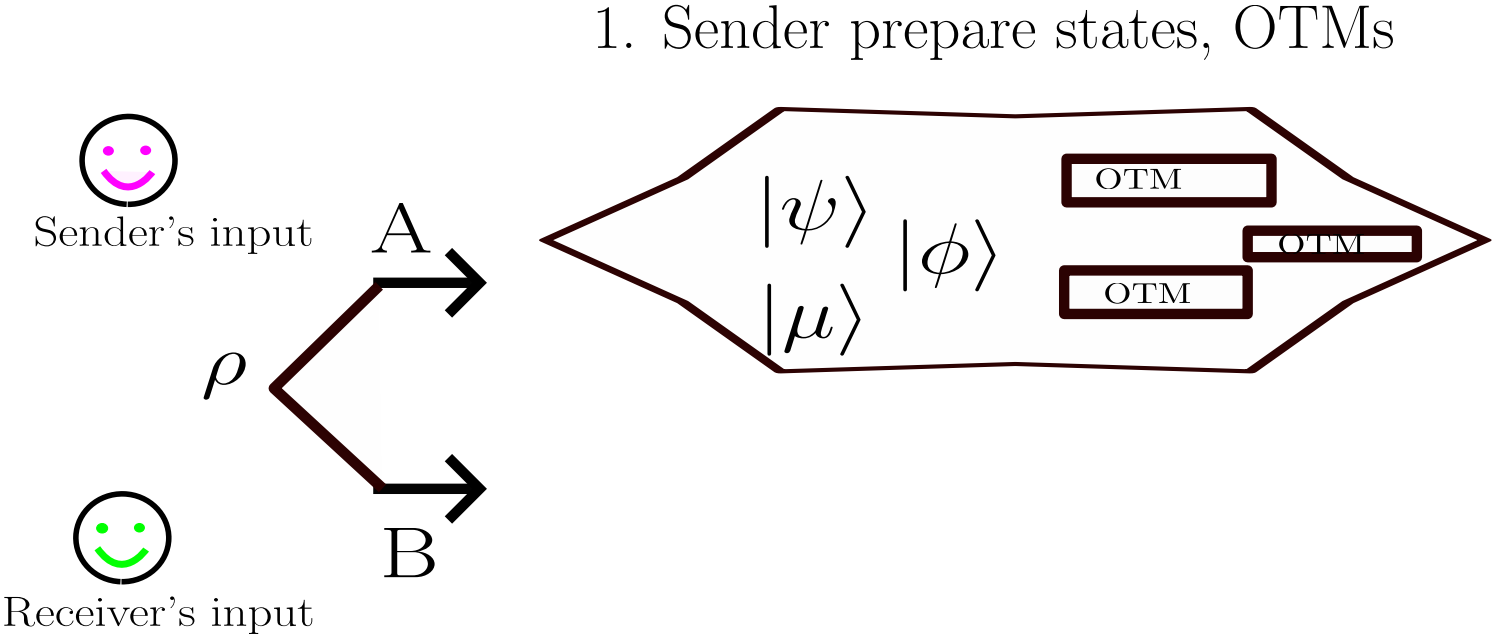
[Goldwasser, Rothblum, Kalai 2008] [Goyal, Ishai, Sahai, Venkatesan, Wadia, 2010] [Bellare, Hoang, Rogaway 2012]

Primary result

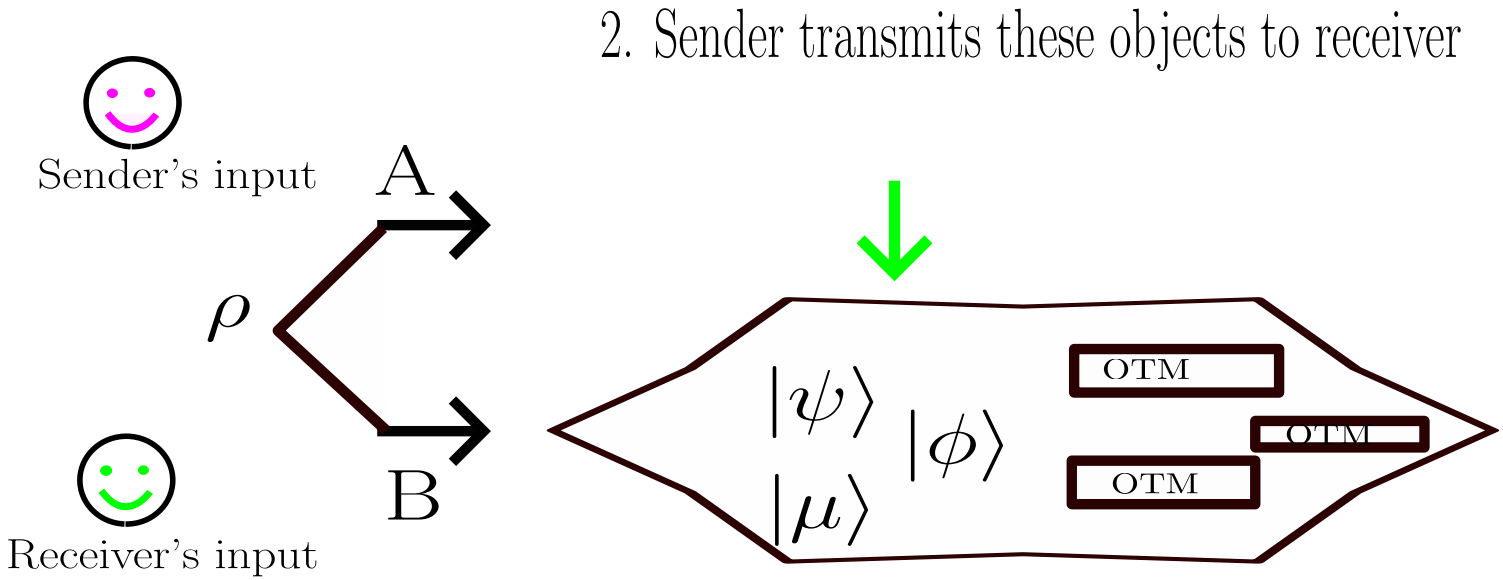
$\exists$ quantum one-time programs for any channel Φ using one-time memories(The very same! No need for **quantum** OTMs) that are:

1. Secure against any malicious receiver. (No restrictions required).
2. ~~Secure against any malicious sender.~~ (Future Project)
3. Universally composable. (Secure even against parallel attacks.)

Primary result... in more detail

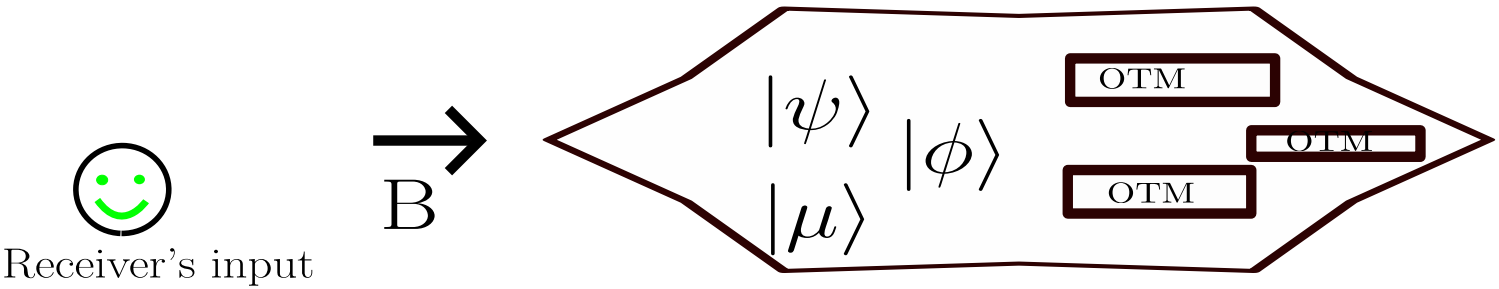


Primary result... in more detail



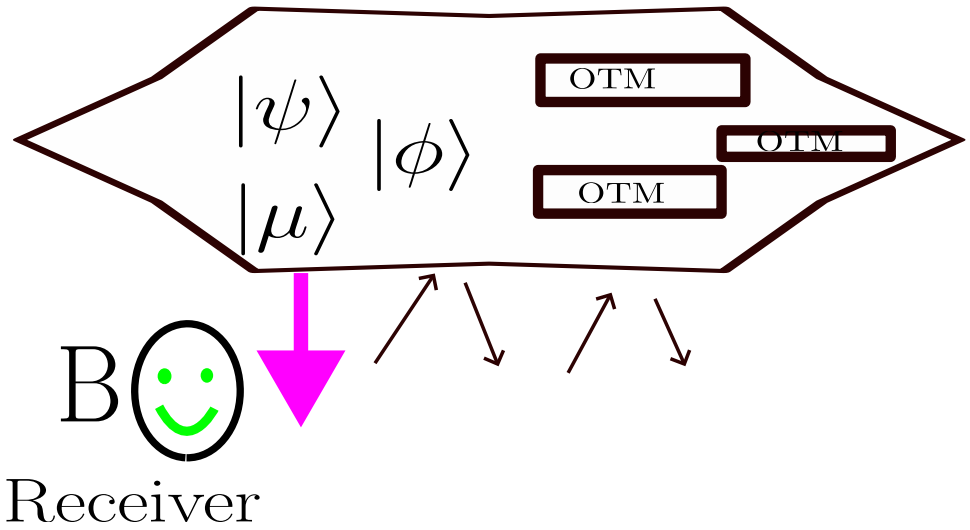
Primary result... in more detail

(Sender is no longer involved.)



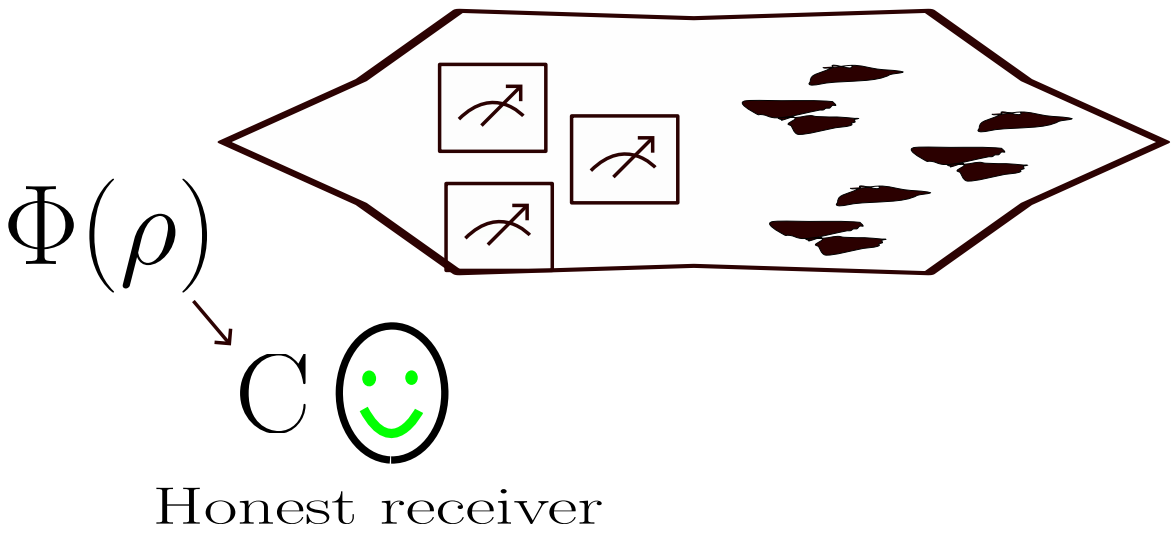
Primary result... in more detail

3. Receiver looks at the states, queries the OTMs.



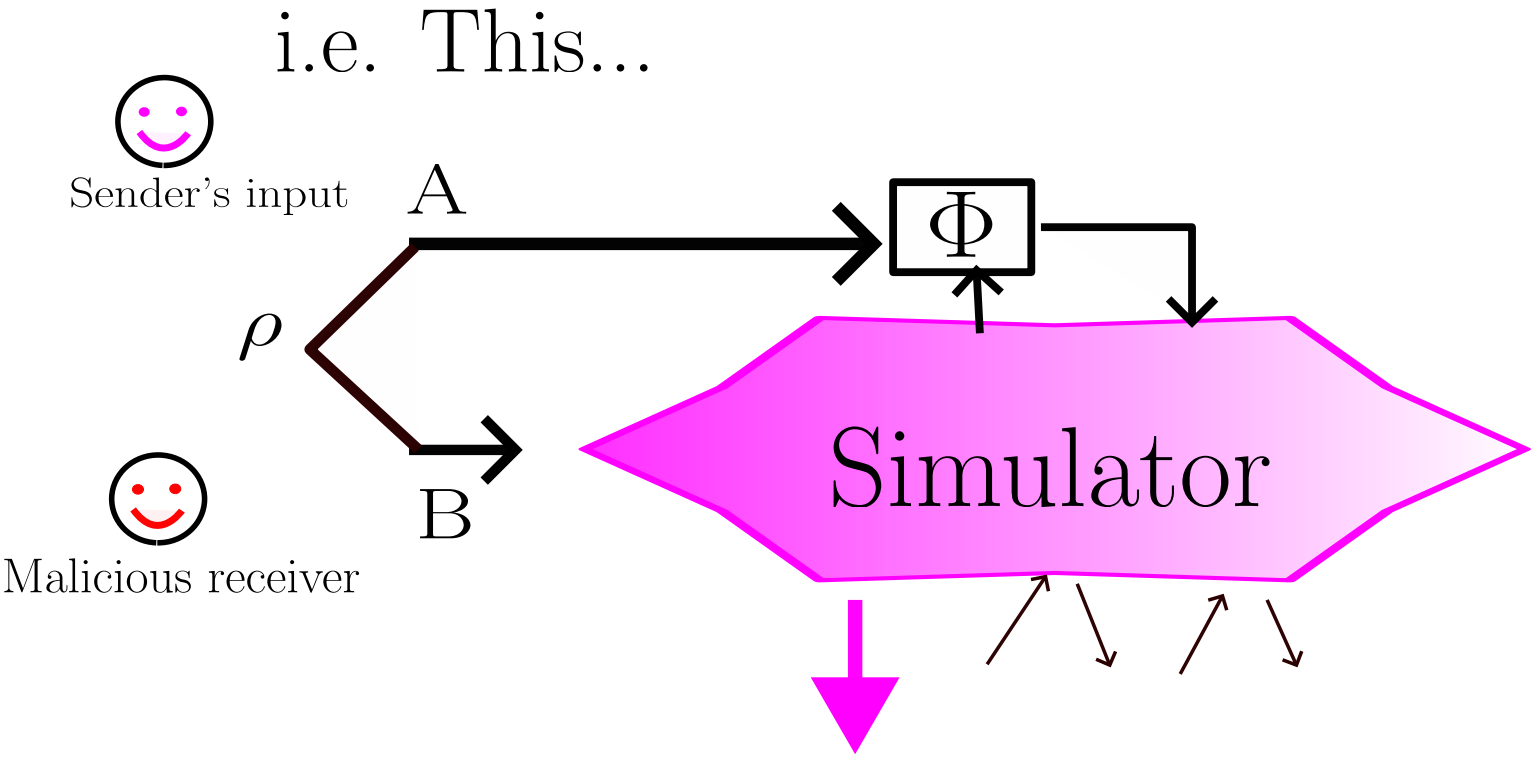
Primary result... in more detail

An honest receiver should be able to recover $\Phi(\rho)$



Security requirement

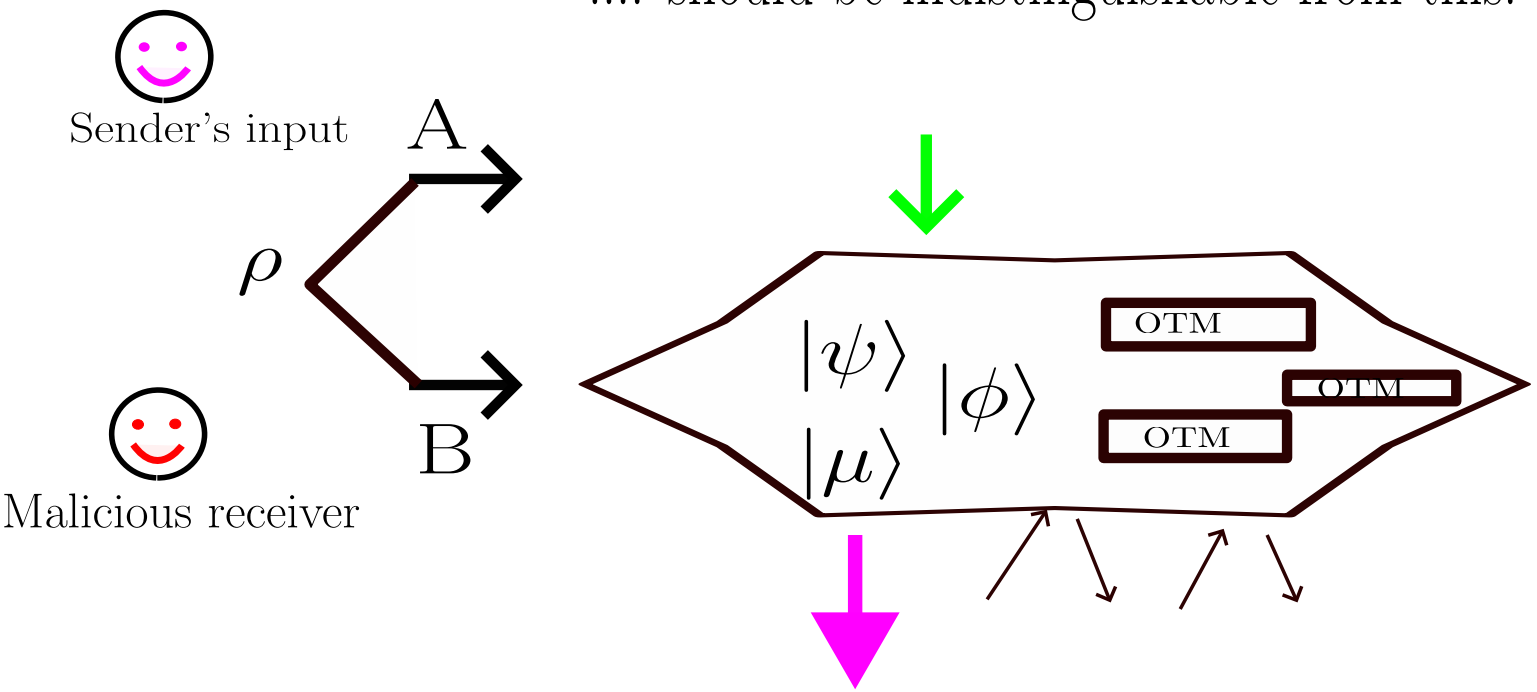
$\exists$ simulator using one-shot access to Φ that mimics the behavior of the sender's QOTP.



Security requirement

$\exists$ simulator using one-shot access to Φ that mimics the behavior of the sender's QOTP.

.... should be indistinguishable from this.

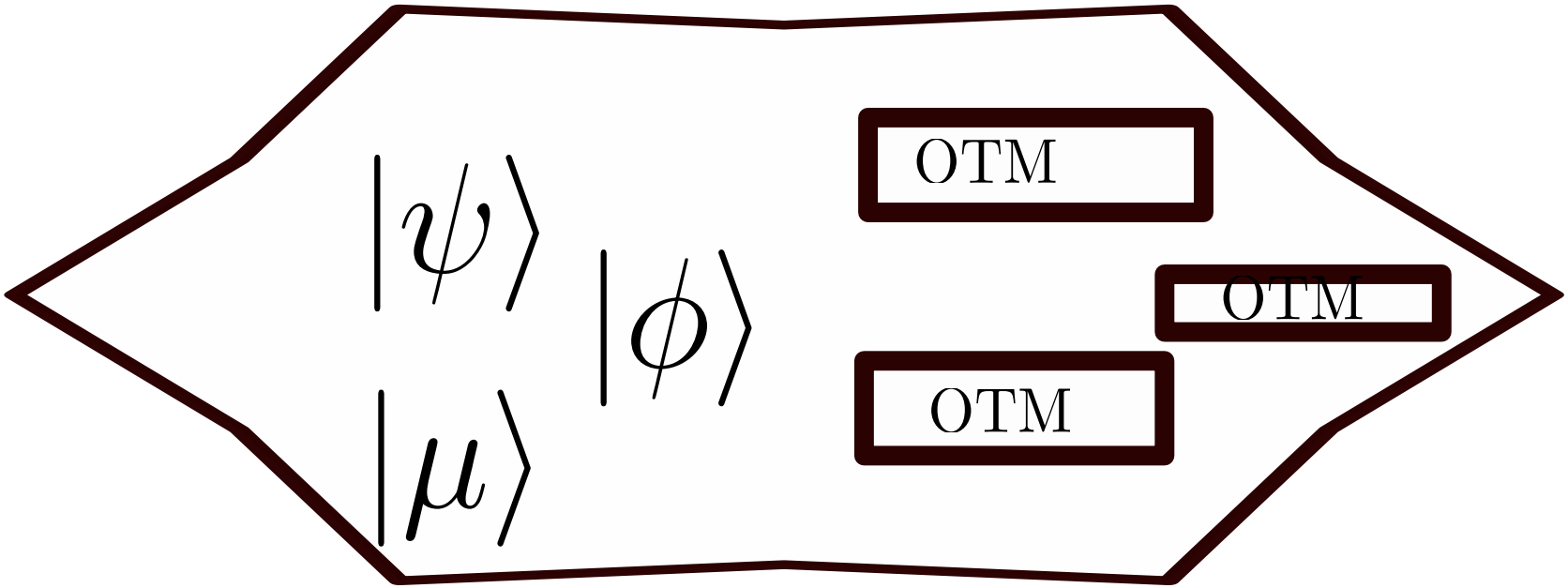


Overview: Honest Receiver Case

1. Use OTMs to build a reactive COTP
2. Quantum Authentication schemes
3. Quantum Computing on Authenticated Data
4. Classical interaction handled by reactive COTP
5. Teleportation-through-(de)authentication

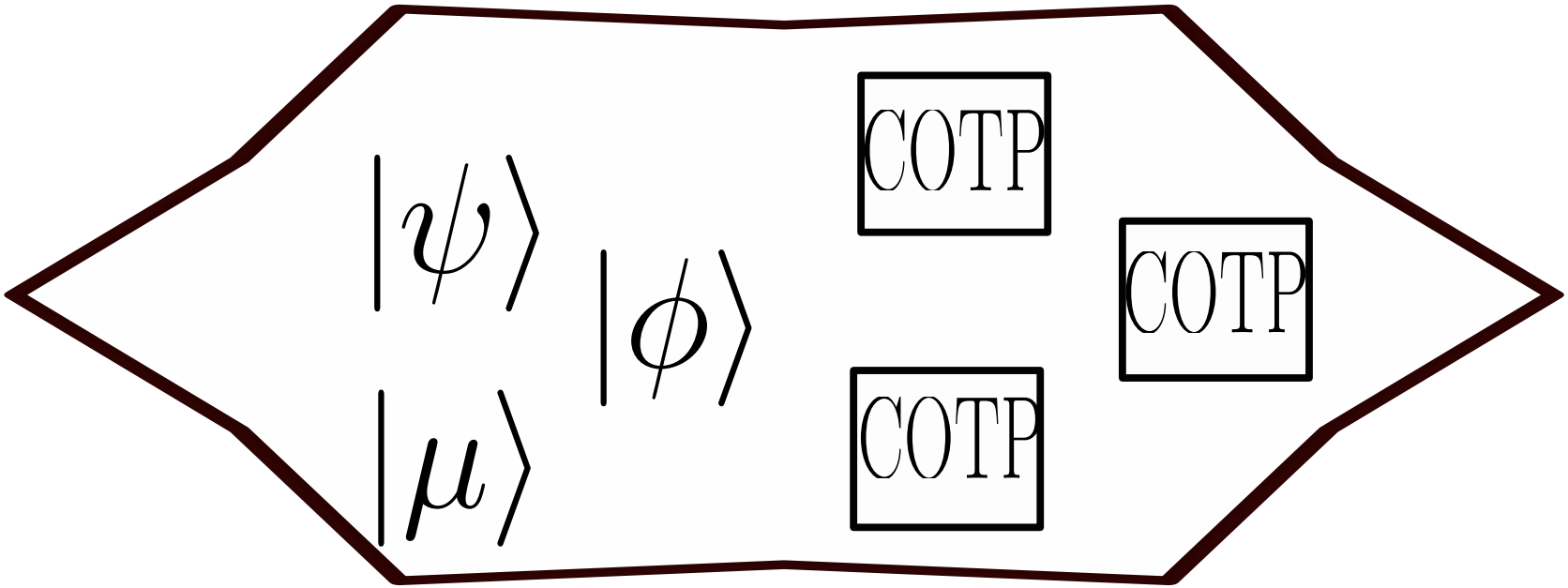
1. Use OTMs to build reactive COTP

Recall: Sender's message consists of qubits and OTMs



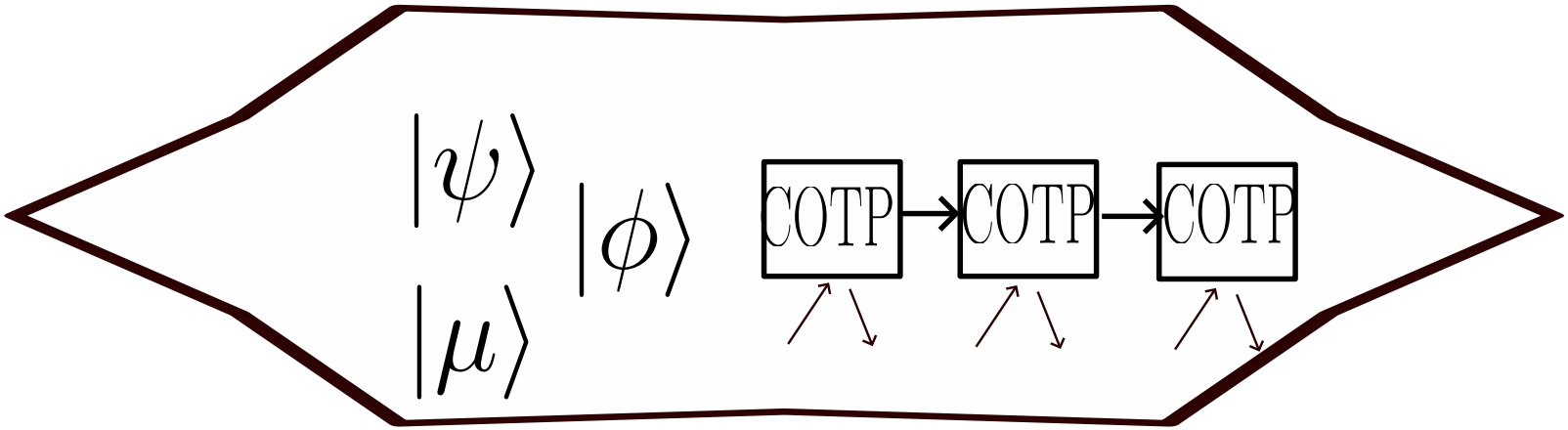
From OTMs to COTPs

Already known: We can get COTPs from OTMs



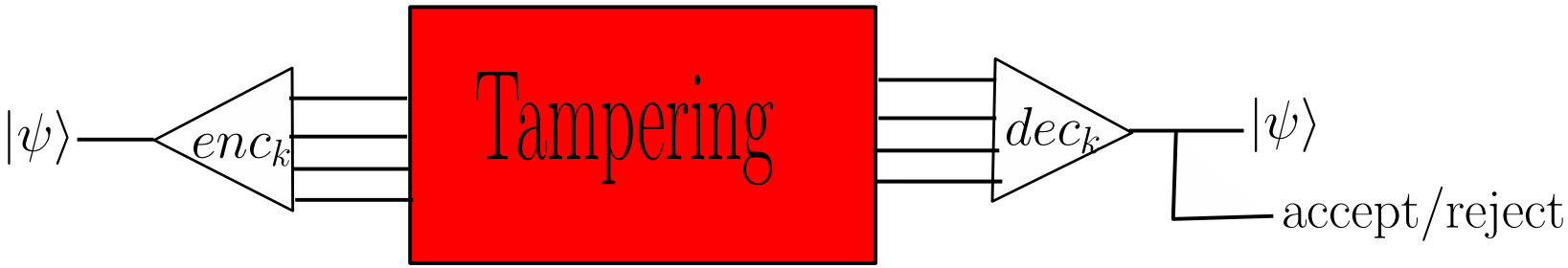
... and from COTPs to reactive COTPs

Easy to show: Can get reactive COTPs from COTPs



2. Quantum authentication schemes

Encode/decode qubits with classical key k such that any tampering is detected w.h.p. over k.



[Barnum, Crepeau, Gottesman, Smith, Tapp 2002]

Quantum Computing on Authenticated Data (QCAD)

Goal: Apply a logical gate G without knowing the key k by means of a gadget $\tilde{G}$.

$$\text{trusted verifier } k \to k'$$

$$\text{Malicious attacker } \boxed{enc_k(|\psi\rangle)} \to \boxed{\tilde{G}enc_k(|\psi\rangle)} = \boxed{enc'_k(G(|\psi\rangle))}$$

Updating the key $k \to k'$ forces the attacker to apply $\tilde{G}$, as otherwise the state would fail verification.

However, there are problems

- Some schemes admit gadgets for certain gates
- No scheme admits gadgets for a universal set of gates
- Universality can be obtained using some tricks: ‘
  - logical measurement
  - ”magic” states
  - interaction between verifier and attacker

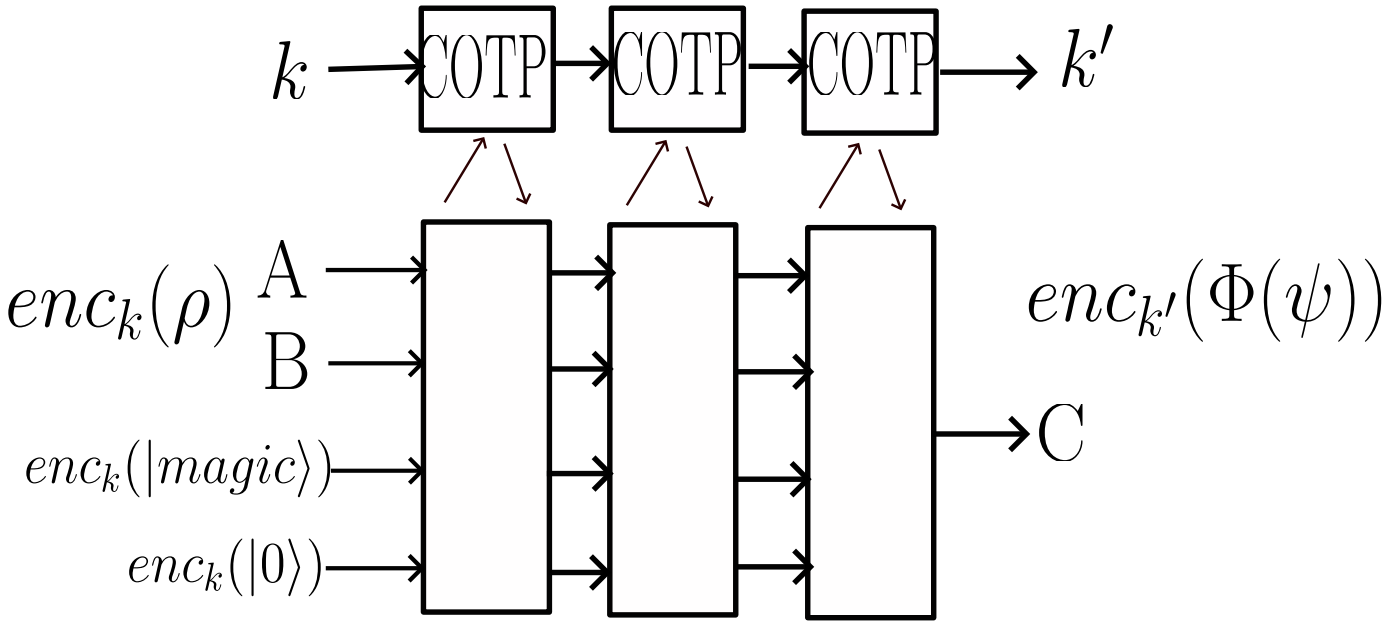
Long Story Short

Any circuit can be implemented on authenticated data, given classical interaction with the verifier.
For example, Trap scheme is a simple qubit scheme that the authors have used.
There are other schemes like the poly scheme, Clifford scheme etc.

COTP for classical interaction

Suppose the receiver somehow holds authenticated inputs (A,B) plus magic states.

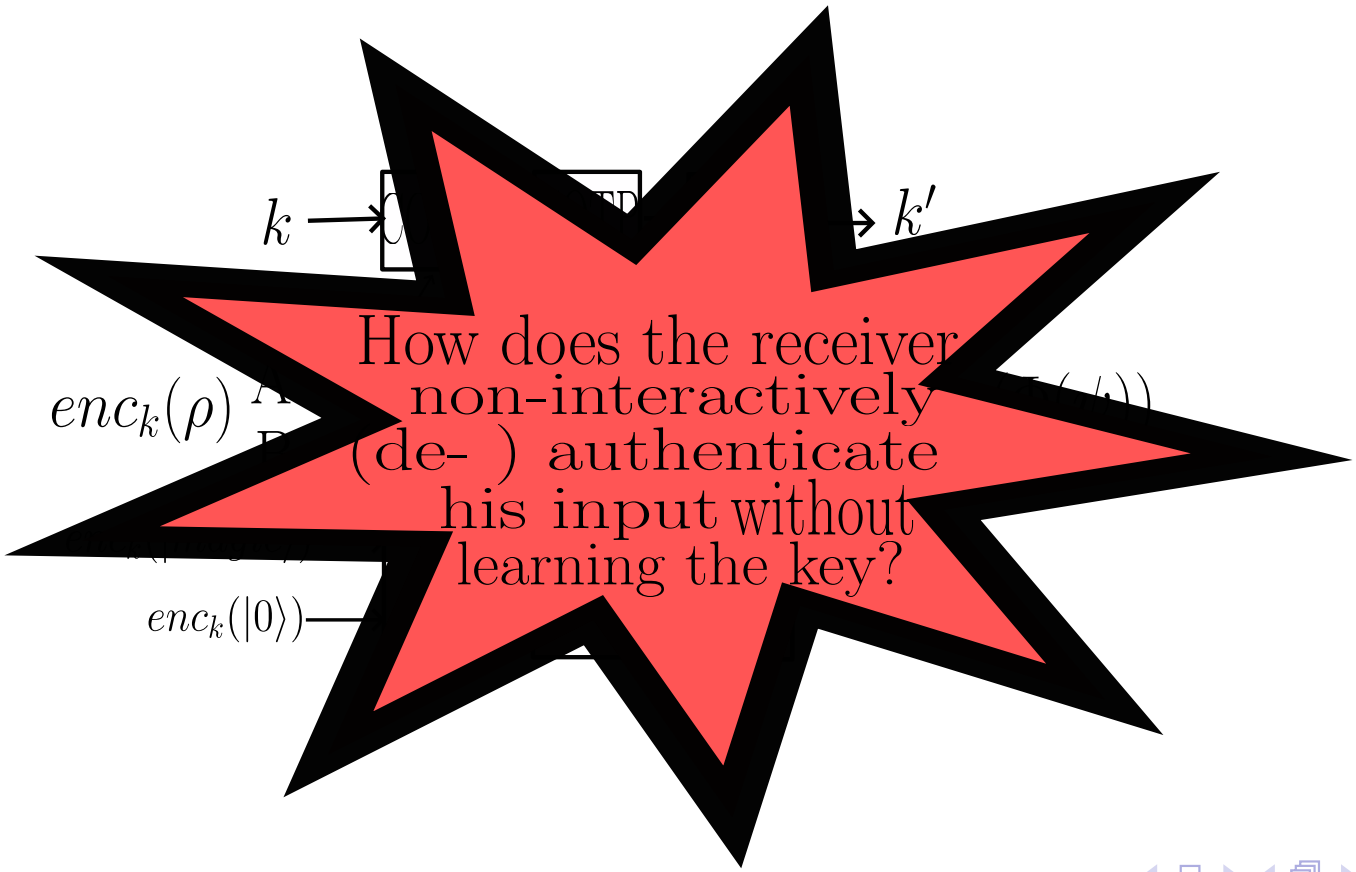
$\implies$ Receiver can compute Φ



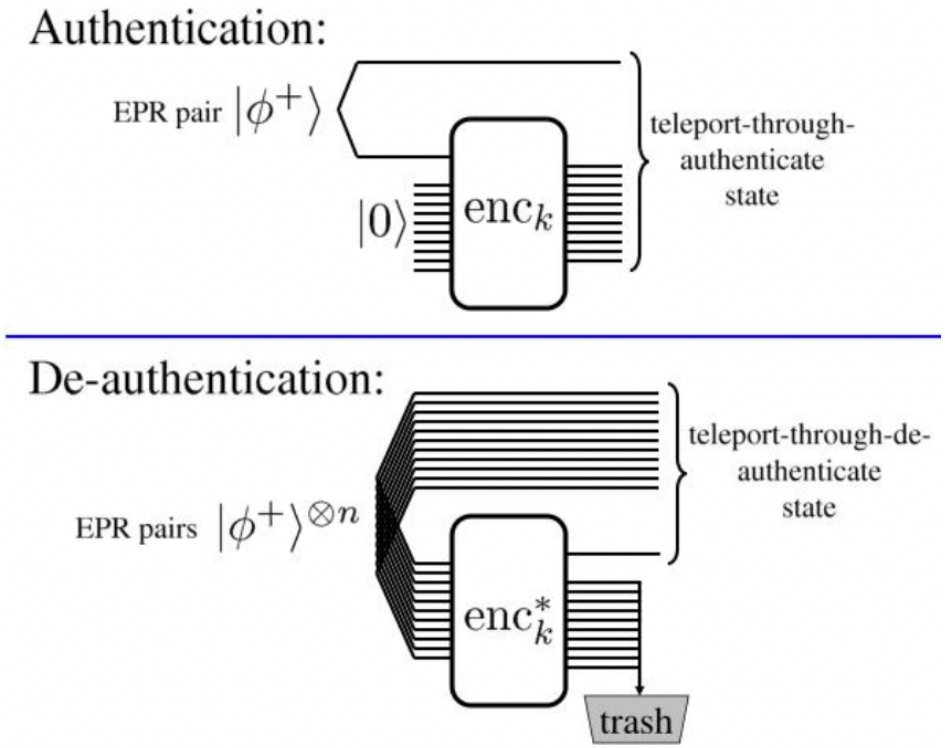
COTP for classical interaction

Suppose the receiver somehow holds authenticated inputs (A,B) plus magic states.

$\implies$ Receiver can compute Φ



Teleport-through-(de)-authentication



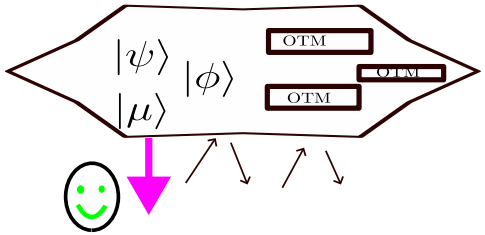
A QOTP consists of

1. Authenticated Registers
   - Sender's input register A
   - Magic states
   - $|0\rangle$ states
   - Teleport-through-(de)-authentication
2. A reactive COTP

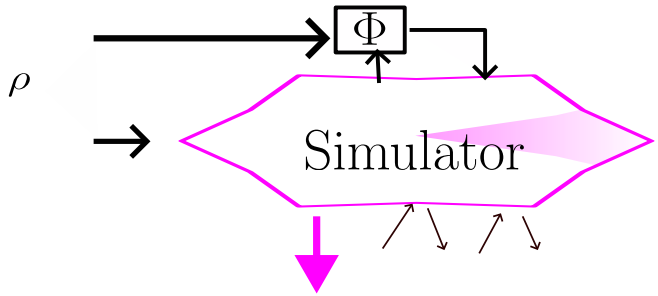
After de-authentication, the final state $\Psi(\rho)$ is encrypted (but not encoded).
The COTP reveals the decryption key only if all measurement results were consistent with the secret key.

Security: a simulator for our QOTP

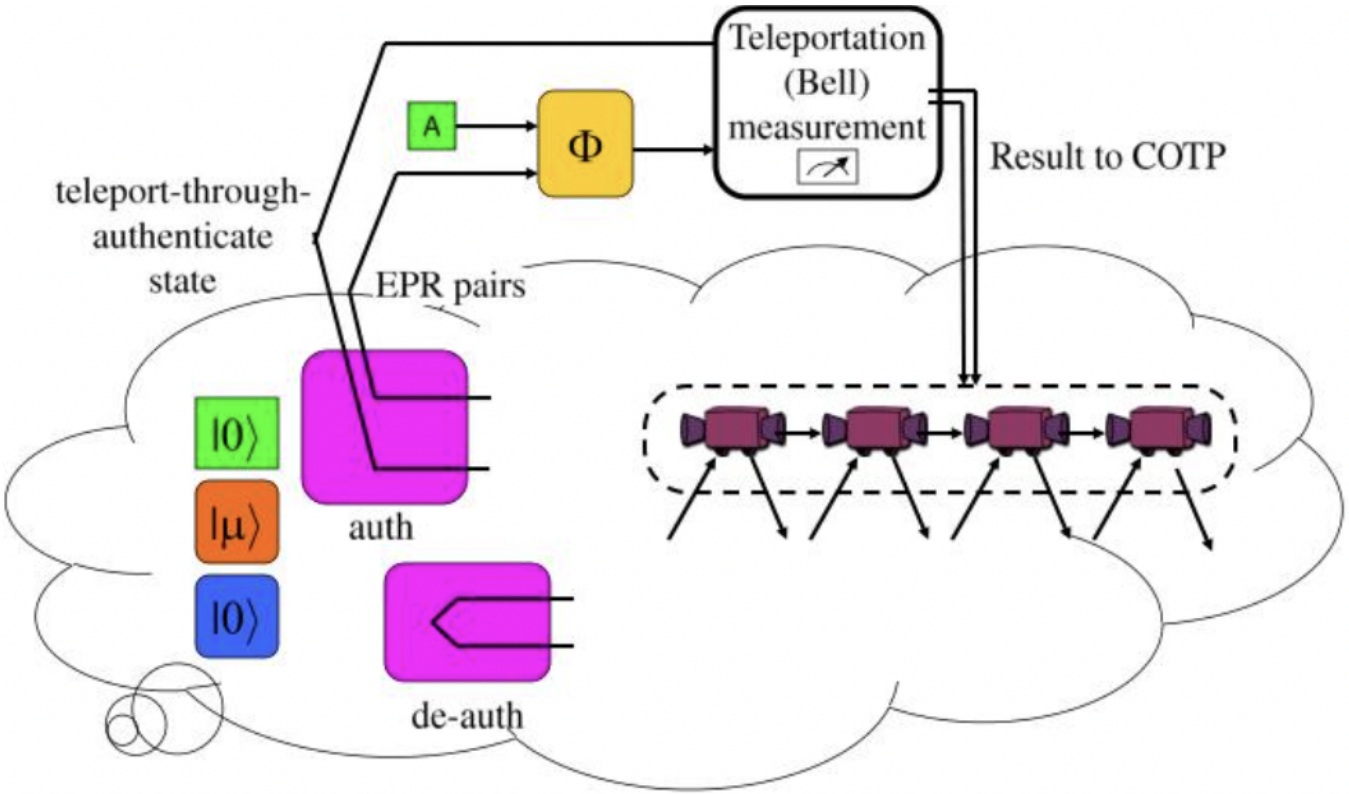
Recall: We need to simulate the sender's QOTP...



... with one-shot access to Φ.



Simulator in pictures



Insert Φ at the beginning, run a dummy computation.

References

- extended abstract: Cryptology ePrint Archive, Report 2013/343
- full version (old): arXiv:1211.1080[quant-ph]
- Gus Gutoski Presentation