Title: Quantum one-time programs

Date: Dec 06, 2012 02:30 PM

URL: http://pirsa.org/12120028

Abstract: A "one-time program" for a channel C is a hypothetical cryptographic primitive by which a user may evaluate C on only one input state of her choice. (Think Mission Impossible: "this tape will self-destruct in five seconds.") One-time programs cannot be achieved without extra assumptions such as secure hardware; it is known that one-time programs can be constructed for classical channels using a very basic hypothetical hardware device called a "one-time memory".

Our main result is the construction of a one-time program for any quantum channel specified by a circuit, assuming the same basic one-time memory devices used for classical channels. The construction achieves universal composability -- the strongest possible security -- against any quantum adversary. It employs a technique for computation on authenticated quantum data and we present a new authentication scheme called the "trap" scheme for this purpose.

Finally, we observe that there is a pathological class of channels that admit trivial one-time programs without any hardware assumptions whatsoever. We characterize these channels, assuming an interesting conjecture on the invertible (or decoherence-free) subspaces of an arbitrary channel.

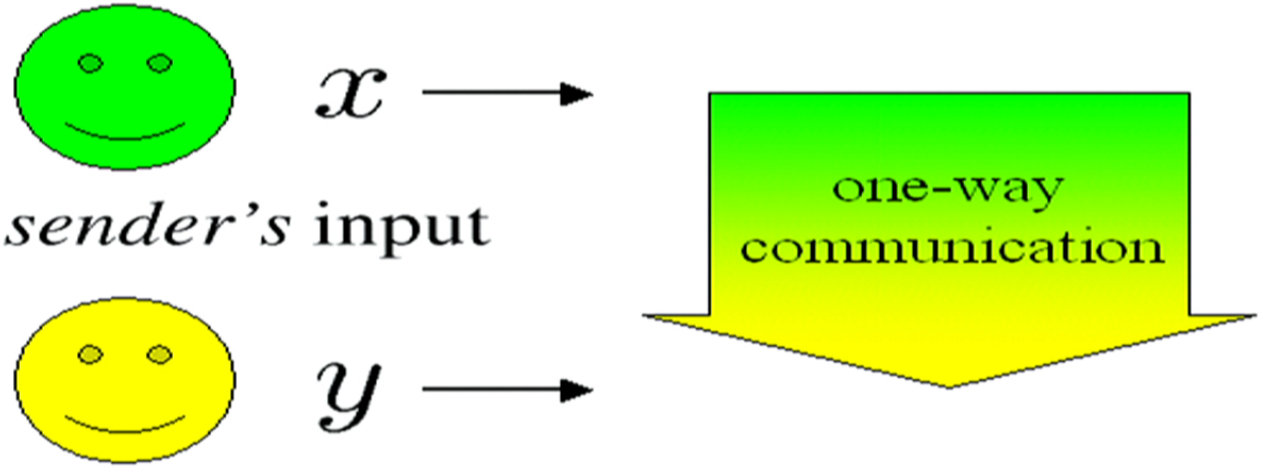
Joint work with Anne Broadbent and Douglas Stebila.

http://arxiv.org/abs/1211.1080

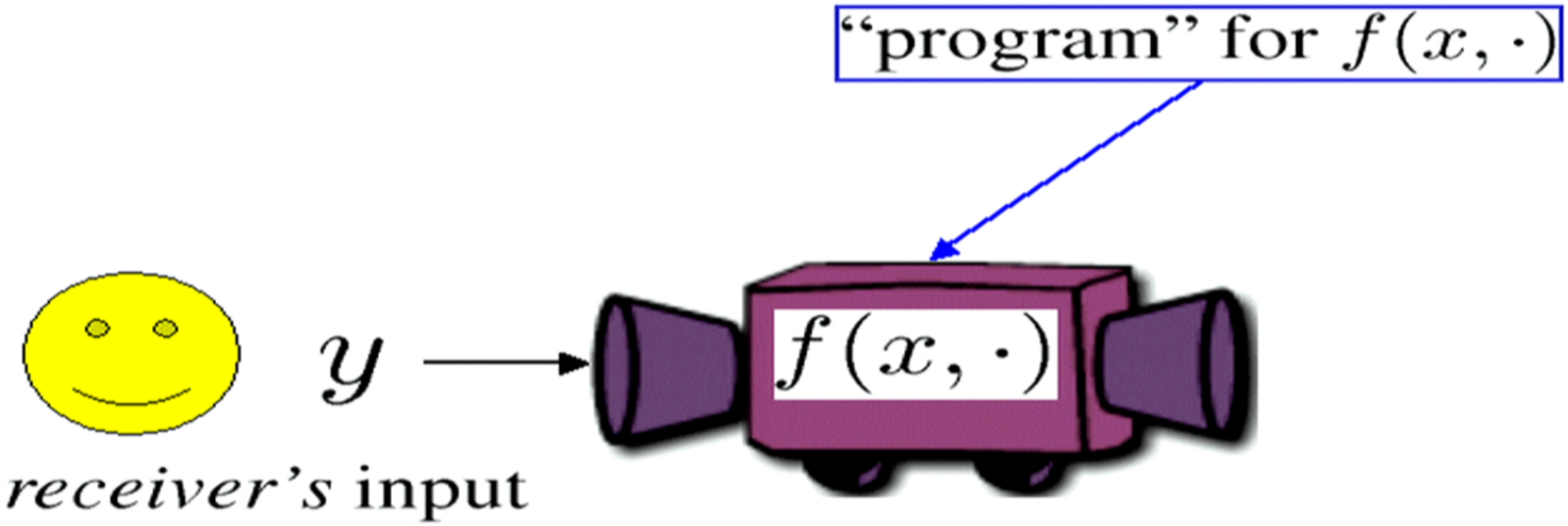
Q: What is a one-time program?

A: A cryptographic primitive allowing two parties to evaluate a publicly known function (specified by a TM or circuit)

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

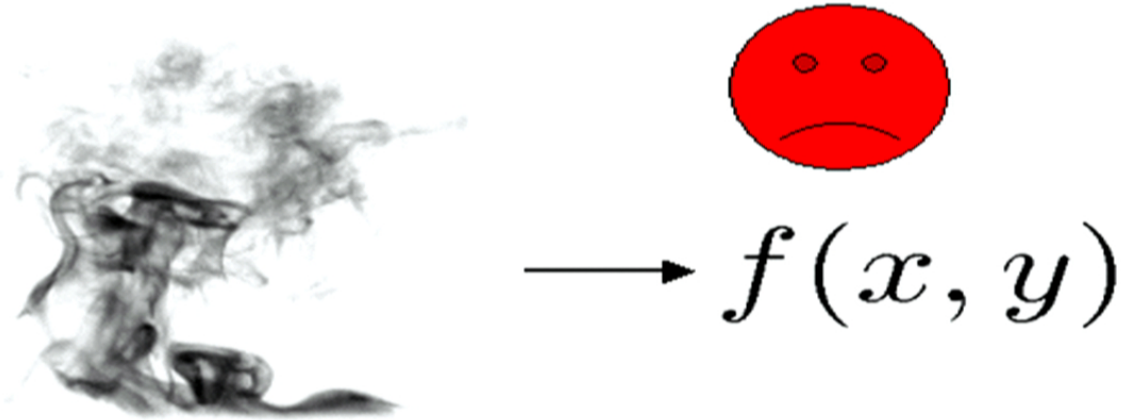


Q: What is a one-time program?



Q: What is a one-time program?

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



Potential applications include:

1. Software protection.



2. Digital rights management.

3. Electronic cash.





Potential applications include:

1. Software protection.



2. Digital rights management.

3. Electronic cash.





Potential applications include:

1. Software protection.



2. Digital rights management.

3. Electronic cash.





Relation to multi-party computation

One-time programs are a special form of *multi-party computation*, which is a familiar cryptographic primitive.

In this talk,

"one-time program"

= "non-interactive two-party computation"

---

multi-party quantum computation:



Relation to multi-party computation

One-time programs are a special form of *multi-party computation*, which is a familiar cryptographic primitive.

In this talk,

"one-time program"

= "non-interactive two-party computation"

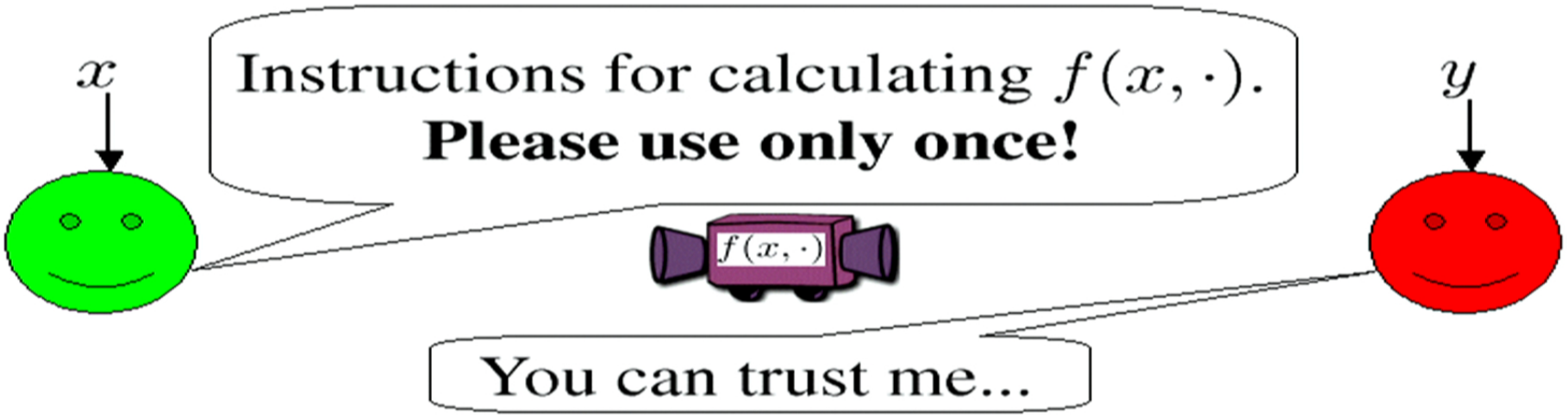
---

multi-party quantum computation:



Software can always be copied

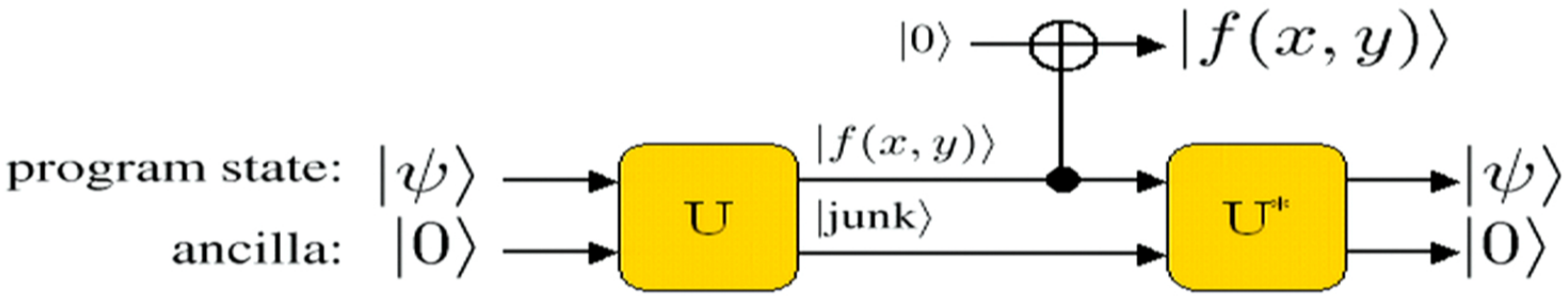
Without any assumptions, classical one-time programs are *trivially impossible*.



$\Longrightarrow$ need assumptions such as quantum mechanics or secure hardware.

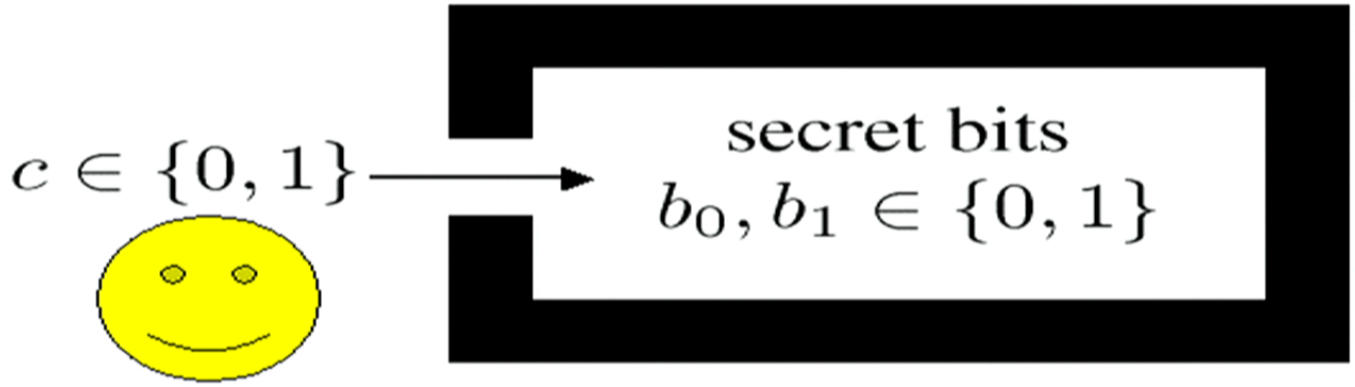
Q: Does quantum information suffice?

A: No. A reversible adversary can always recover a "program state" and evaluate $f(x, \cdot)$ multiple times.



Secure hardware: one-time memory

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



user selects a bit

[Goldwasser, Rothblum, Kalai 2008]
[Goyal, Ishai, Sahai, Venkatesan, Wadia 2010]

What's already known

$\exists$ 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]

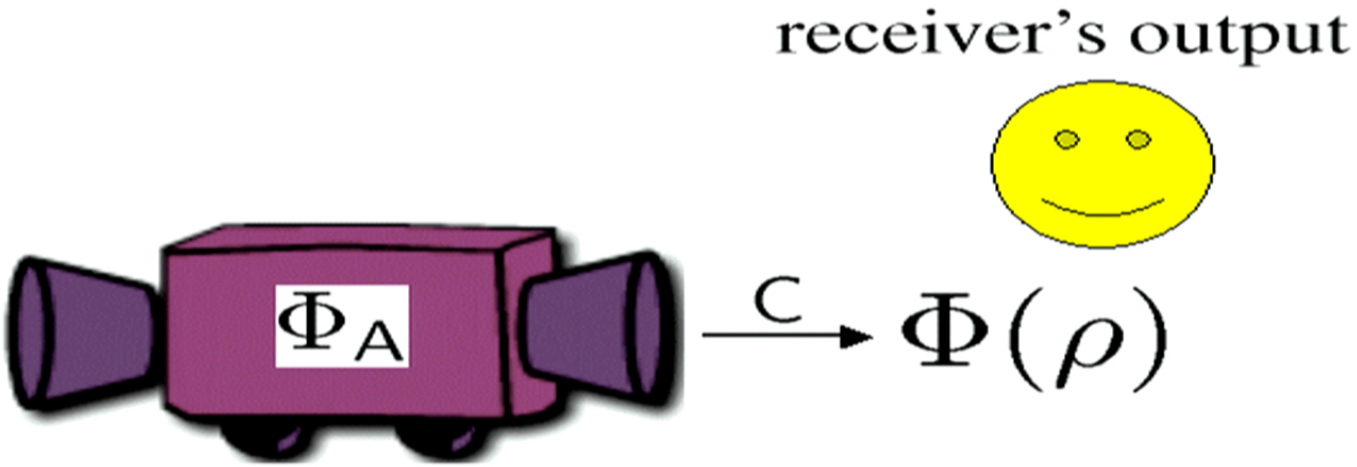
What's already known

$\exists$ 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]

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.



receiver's output



$\xrightarrow{C} \Phi(\rho)$

Recall existing results for COTPs:

$\exists$ 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]

Our primary result:

channel Φ

∃ quantum 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.)

Future project

The very same!
(No need for "quantum" OTMs.)

Our primary result:

channel Φ

$\exists$ quantum 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.~~



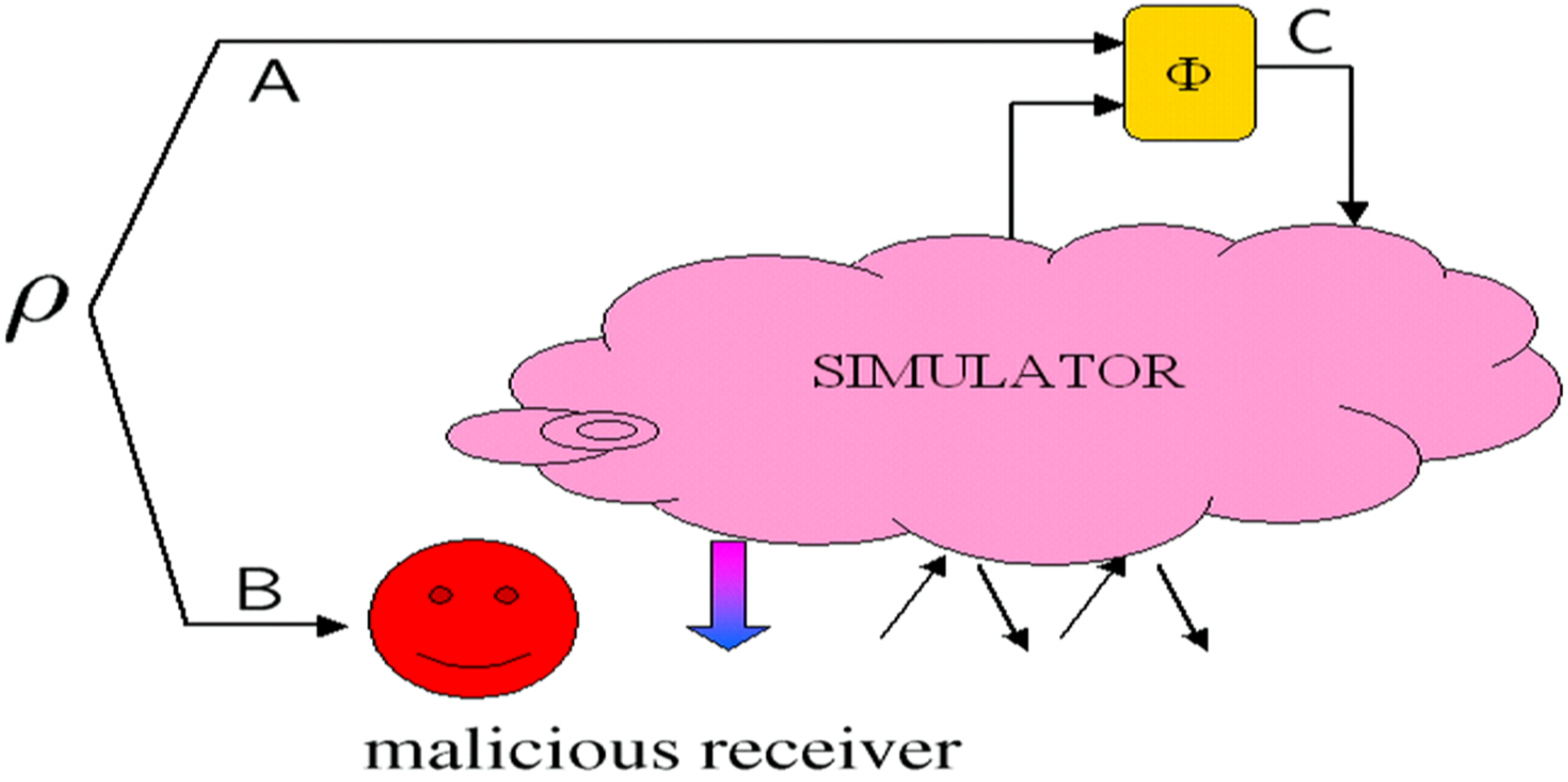
3. Universally composable.
   (Secure even against parallel attacks.)

Future project

The very same!
(No need for "quantum" OTMs.)

Security requirement

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



Encode-encrypt makes life easy

Any attack on an encode-encrypt scheme is equivalent to a mixture of Pauli attacks on $\mathscr{E}$.



Encode-encrypt makes life easy

Any attack on an encode-encrypt scheme is equivalent to a mixture of Pauli attacks on $\mathscr{E}$.



Thus, any family $\mathscr{E}$ secure against Pauli attacks induces a secure encode-encrypt authentication scheme.

(Defining security for authentication schemes is tricky; encode-encrypt schemes make it easy.)

The trap scheme

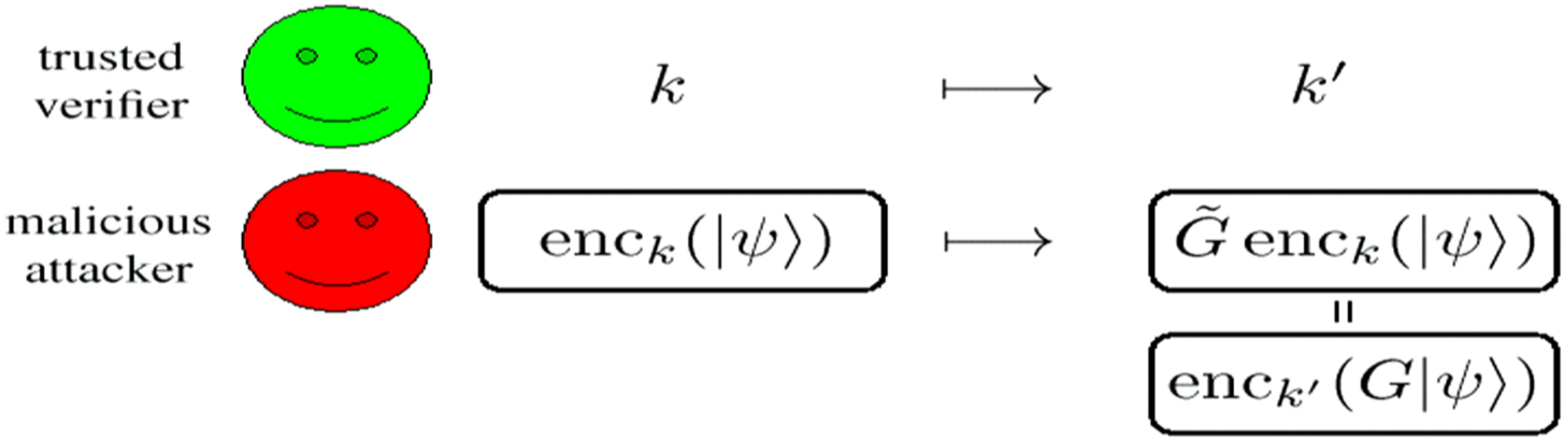
Proposition. *Trap codes are* $(2/3)^{d/2}$*-secure against Pauli attacks.*

Proof sketch. In order to act non-trivially, any Pauli Q must have weight $\geq d$.
To avoid detection, Q must avoid all the traps.
Simple analysis shows $\Pr_\pi \leq (2/3)^{d/2}$. □

Corollary. *Trap codes induce a secure encode-encrypt authentication scheme with key* (π, P).

3. Quantum computing on authenticated data (QCAD)

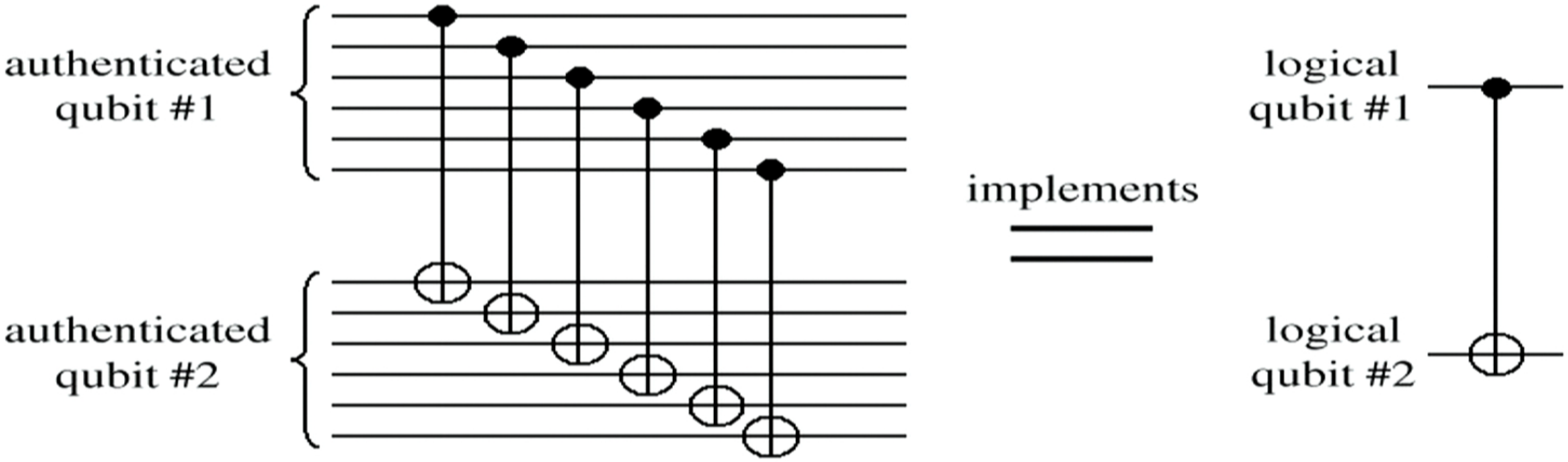
Goal: Apply a gate G to authenticated data without knowing the key k.



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

Example: CNOT in the trap scheme

For a trap scheme based on a CSS code E, the CNOT admits a bitwise implementation.



Key update:

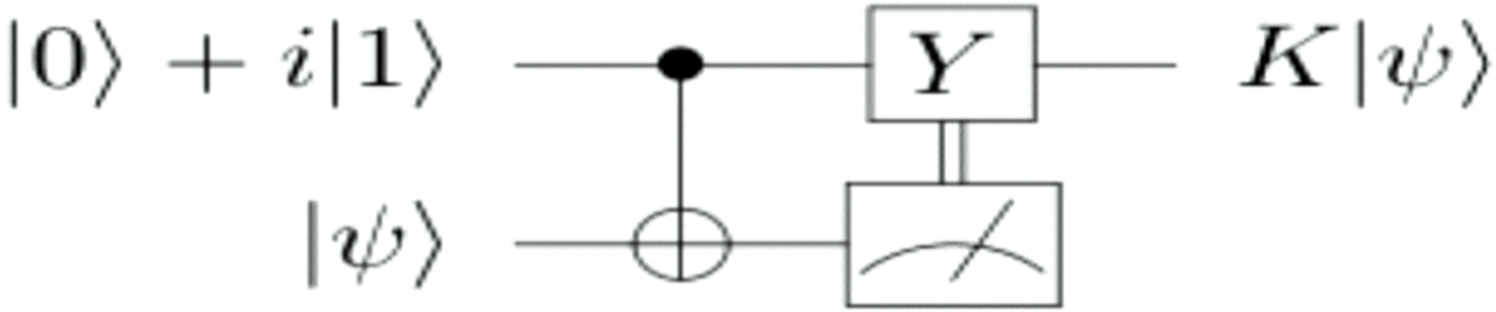
$$\text{qubit \#1:} \quad (\pi, X^p Z^q) \mapsto (\pi, X^p Z^{q+s})$$

$$\text{qubit \#2:} \quad (\pi, X^r Z^s) \mapsto (\pi, X^{p+r} Z^s)$$

Some gates require "magic states"

Universality cannot be achieved by bitwise gates alone.
Example: $K = \left(\begin{smallmatrix} 1 & 0 \\ 0 & i \end{smallmatrix}\right)$ in the trap scheme.

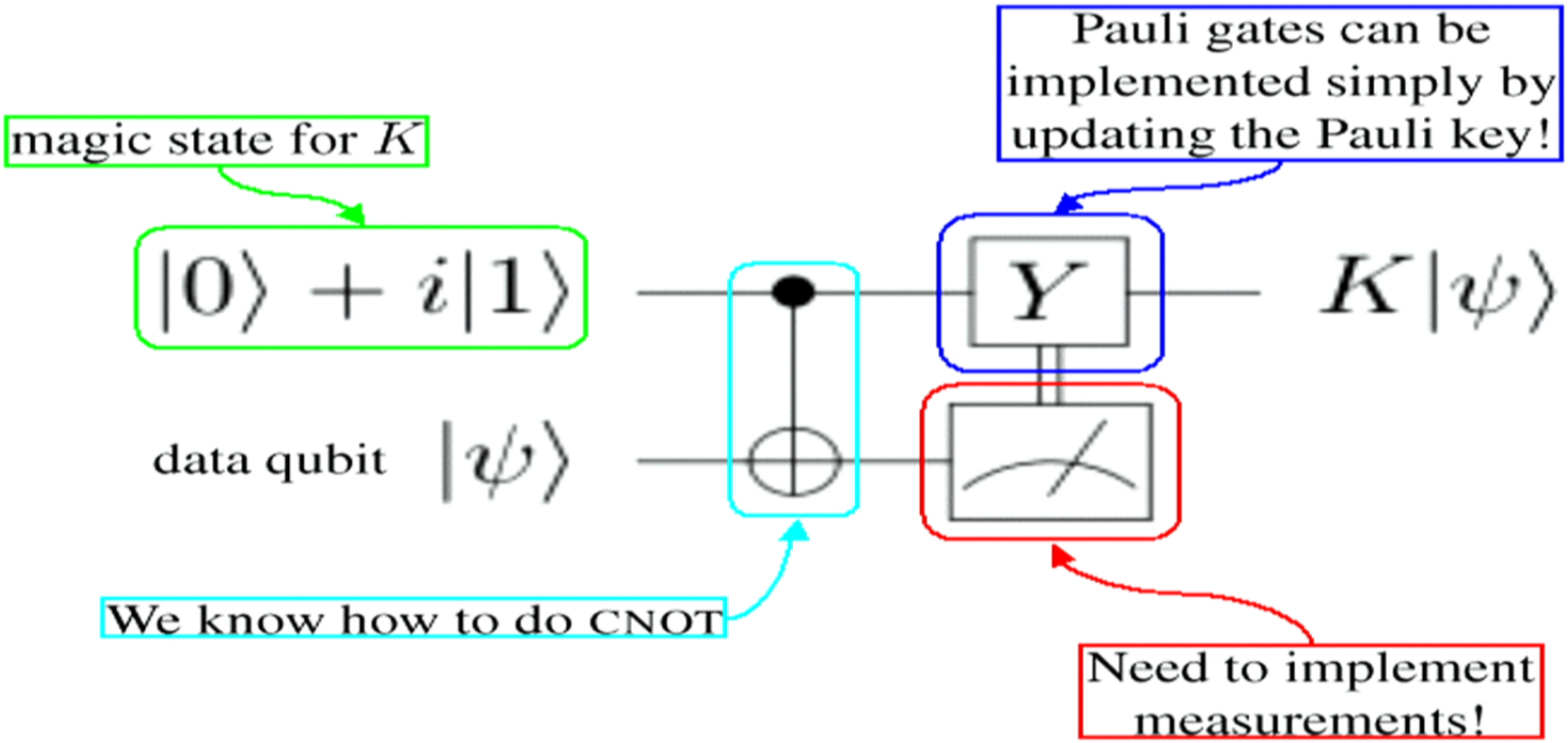
Logical circuit for K. (Look familiar?)



Some gates require "magic states"

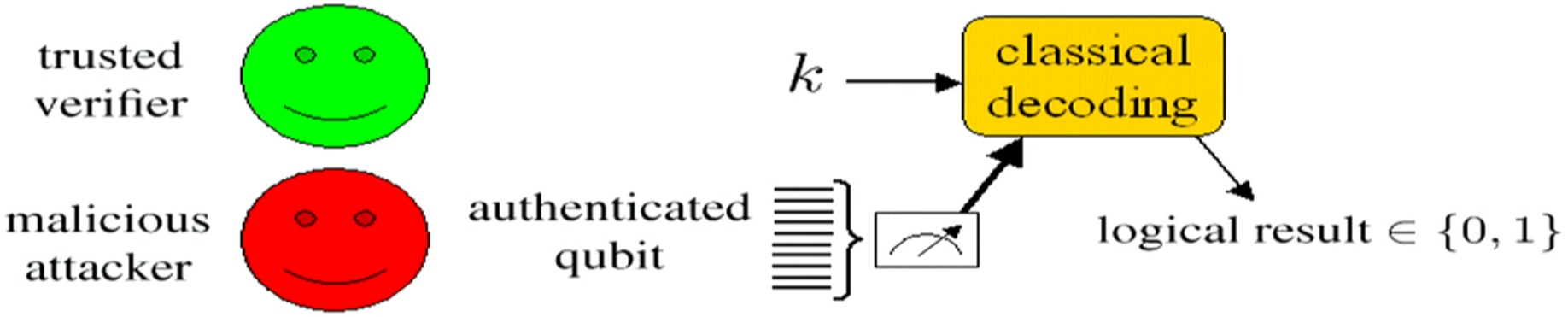
Universality cannot be achieved by bitwise gates alone.

Example: $K = \left(\begin{smallmatrix} 1 & 0 \\ 0 & i \end{smallmatrix}\right)$ in the trap scheme.



Measurement in encode-encrypt schemes

Any encode-encrypt scheme based on a family $\mathscr{E}$ of CSS codes admits measurement of authenticated data. (*e.g.* Trap scheme based on a CSS code E.)

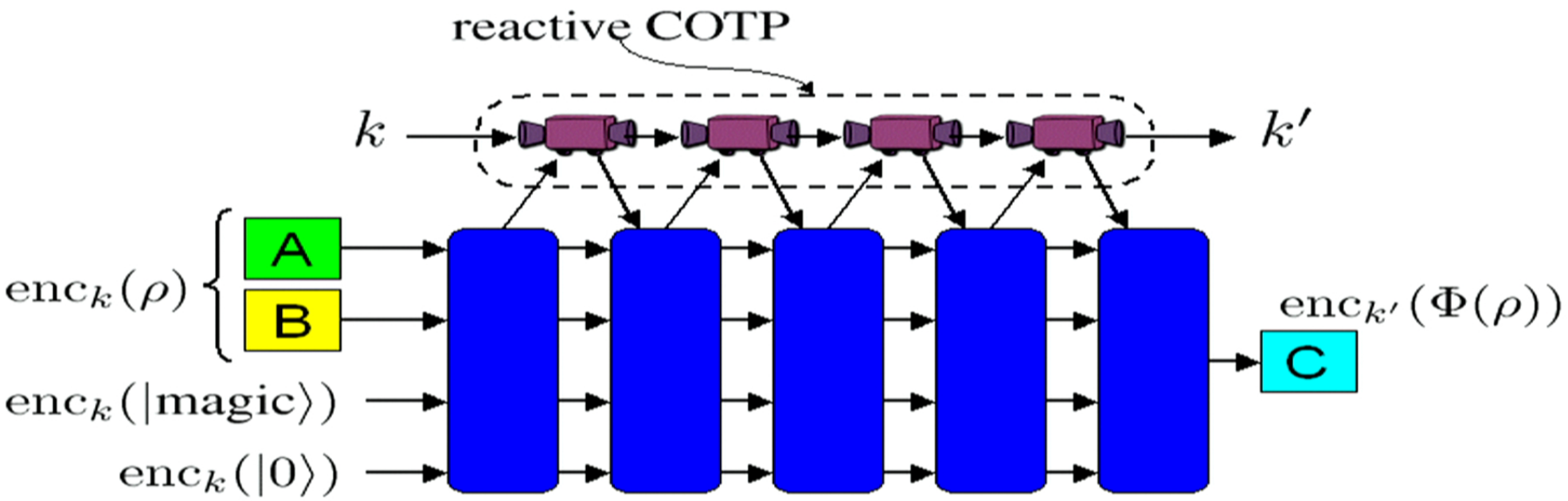


Long story short: Any circuit can be implemented on authenticated data, given classical interaction with the verifier.

4. COTP for classical interaction

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

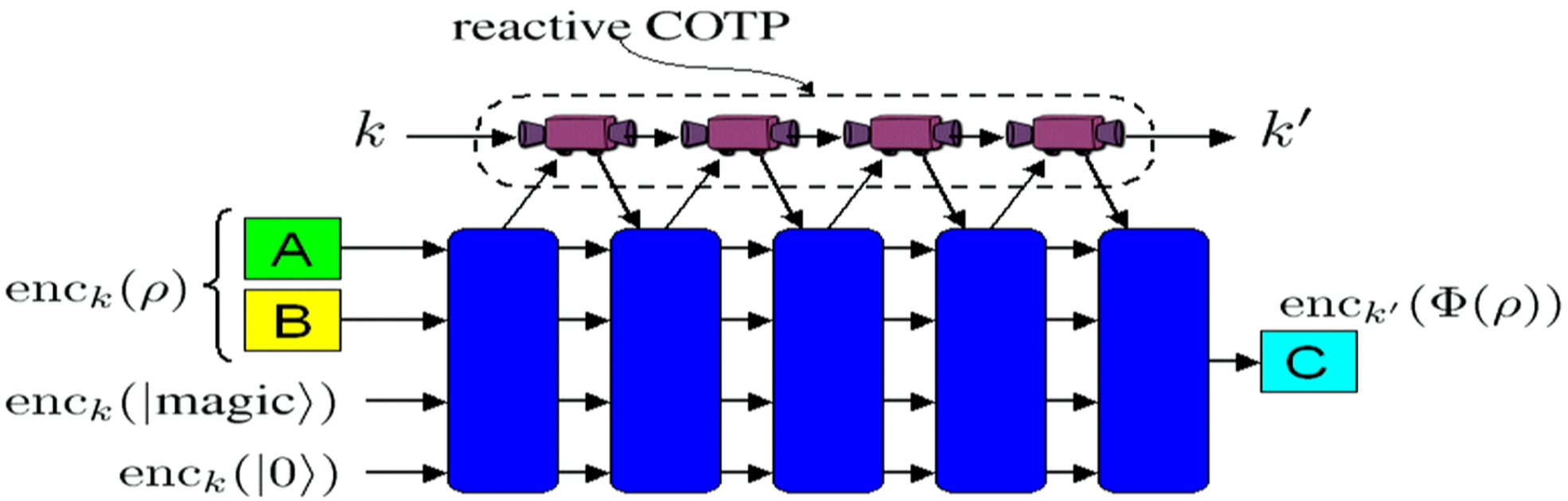
$\Longrightarrow$ Receiver can compute Φ!



4. COTP for classical interaction

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

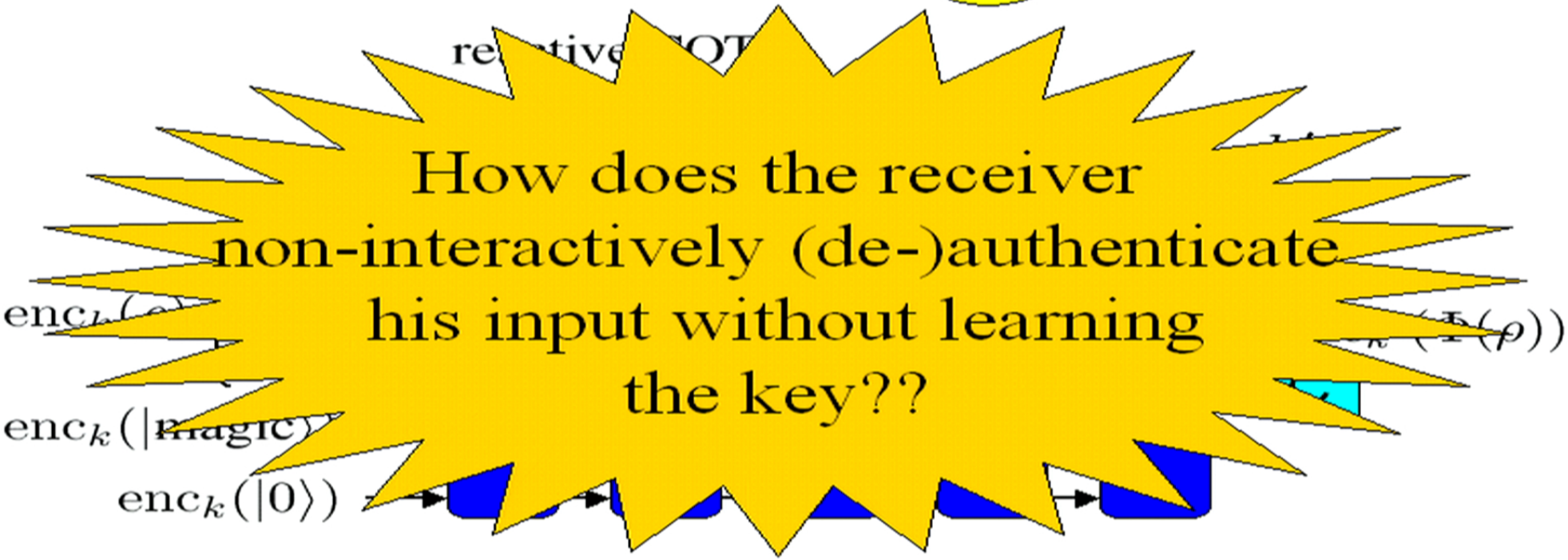
$\Longrightarrow$ Receiver can compute Φ!



4. COTP for classical interaction

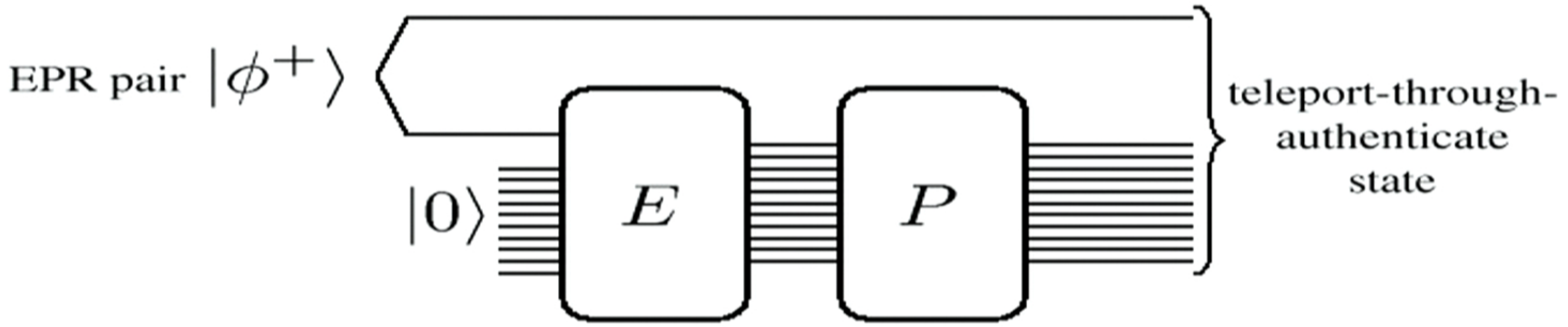
Suppose the receiver holds authenticated inputs (A, B) plus magic states.

$\Longrightarrow$ Receiver can compute Φ!

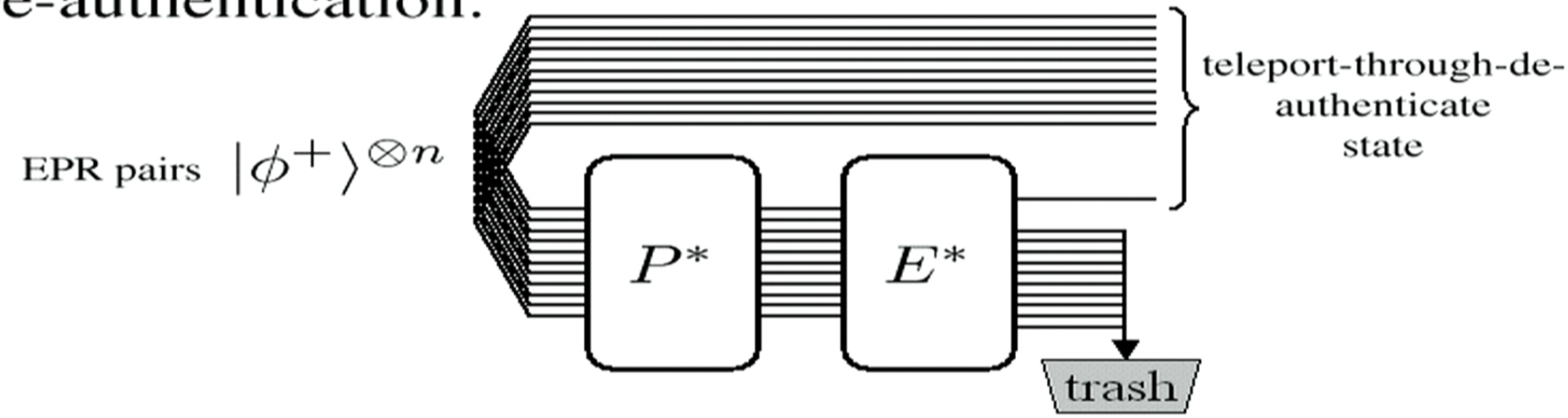


5. Teleport-through-(de)-authentication

Authentication:

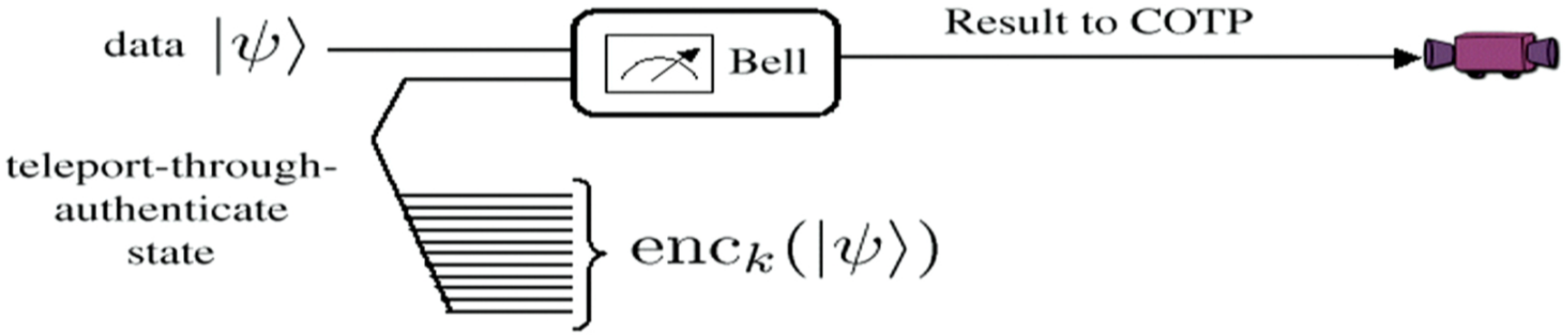


De-authentication:



Teleportation handled by the COTP

Teleportation (Bell) measurement re-encrypts with another Pauli T.
The COTP needs to know T so it can update its secret key.



(Similar for de-authentication.)

A QOTP consists of:

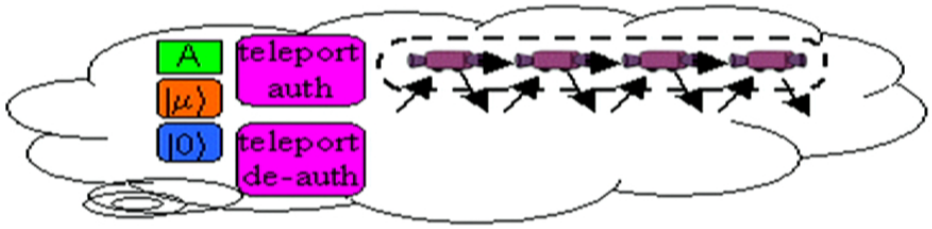
1. Authenticated registers:

   (a) Sender's input register A.

   (b) Magic states.

   (c) $|0\rangle$ states.

   (d) Teleport-through-(de)-authentication states.



2. A reactive COTP.

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

A QOTP consists of:

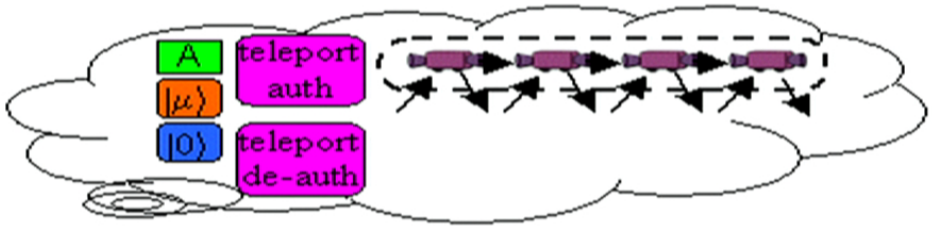
1. Authenticated registers:

   (a) Sender's input register A.

   (b) Magic states.

   (c) $|0\rangle$ states.

   (d) Teleport-through-(de)-authentication states.



2. A reactive COTP.

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

A QOTP consists of:

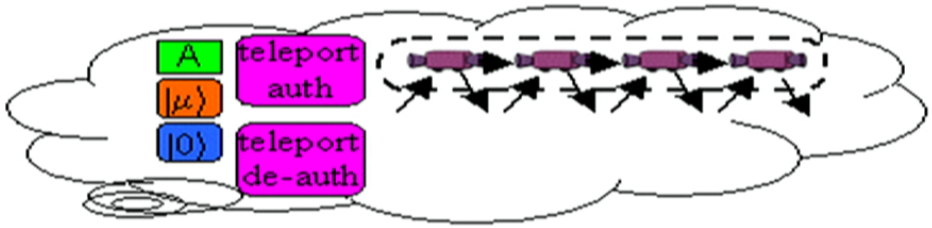
1. Authenticated registers:

   (a) Sender's input register A.

   (b) Magic states.

   (c) $|0\rangle$ states.

   (d) Teleport-through-(de)-authentication states.

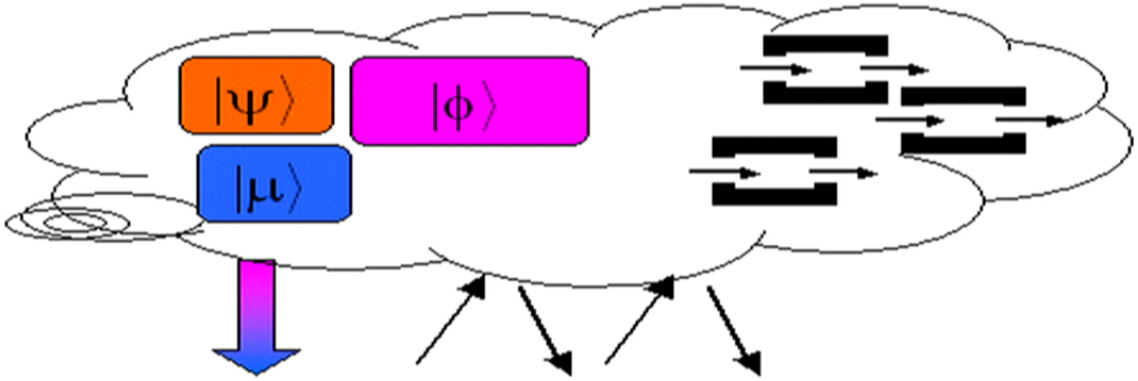


2. A reactive COTP.

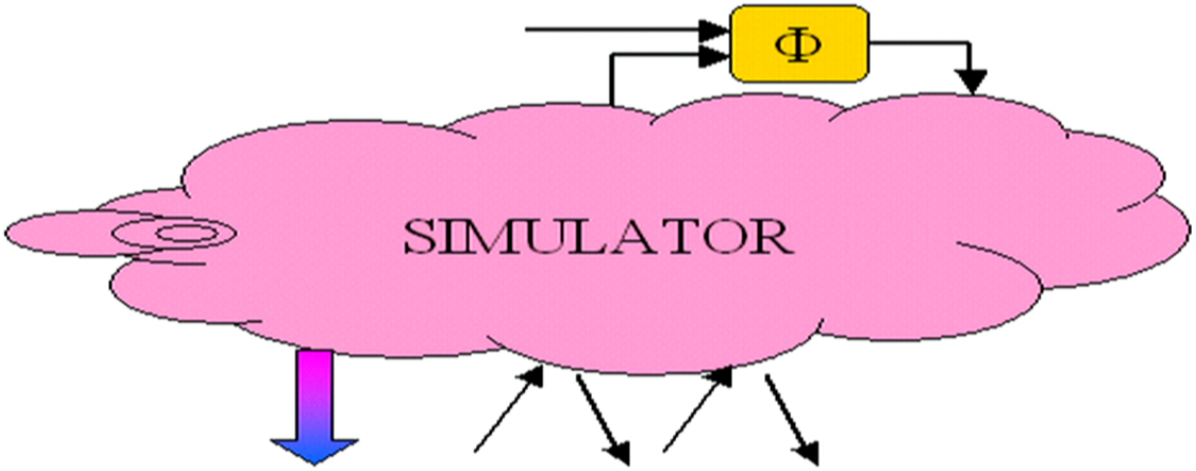
After de-authentication, the final state $\Phi(\rho)$ is encrypted (but not encoded).
The COTP reveals the decryption key only if all measurement results were consistent with secret key.

Security: a simulator for our QOTP

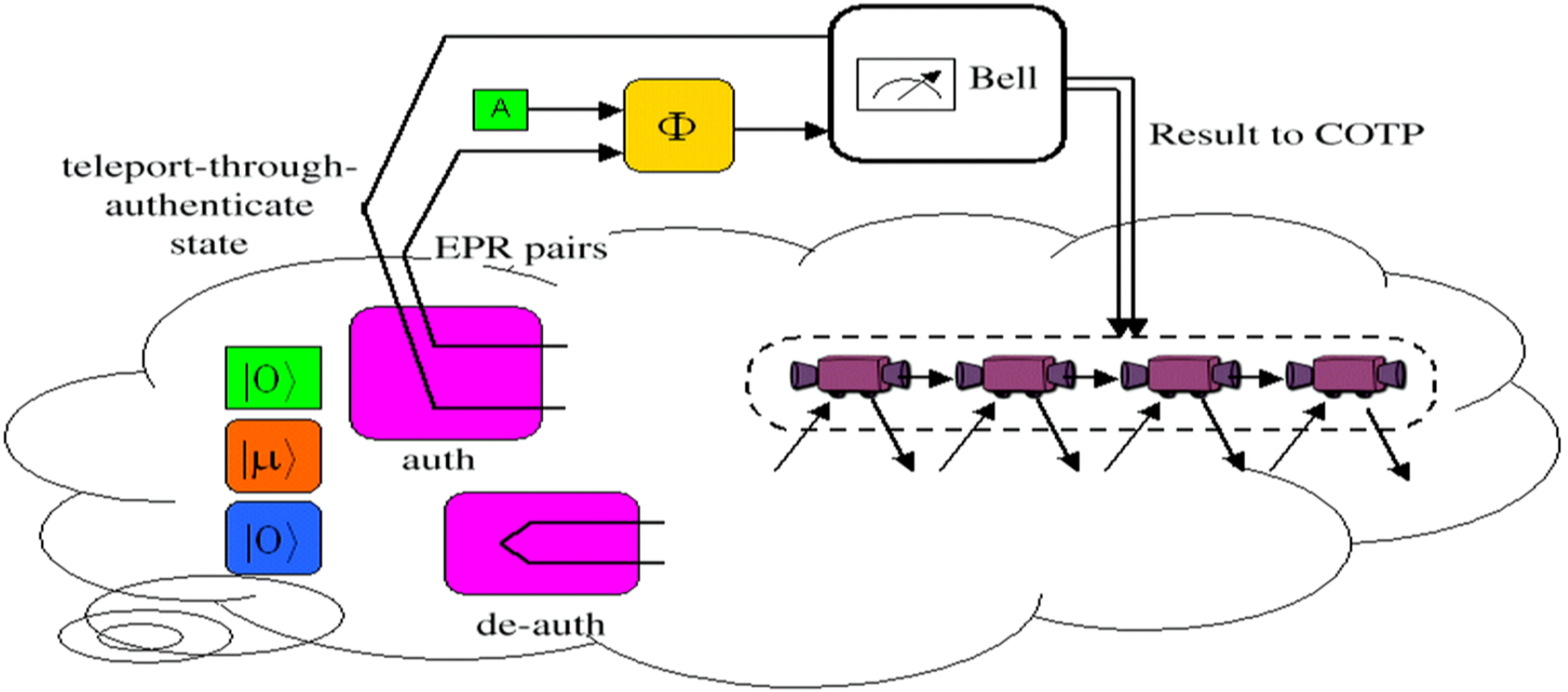
Recall: We need to simulate the sender's QOTP...



...with one-shot access to Φ.

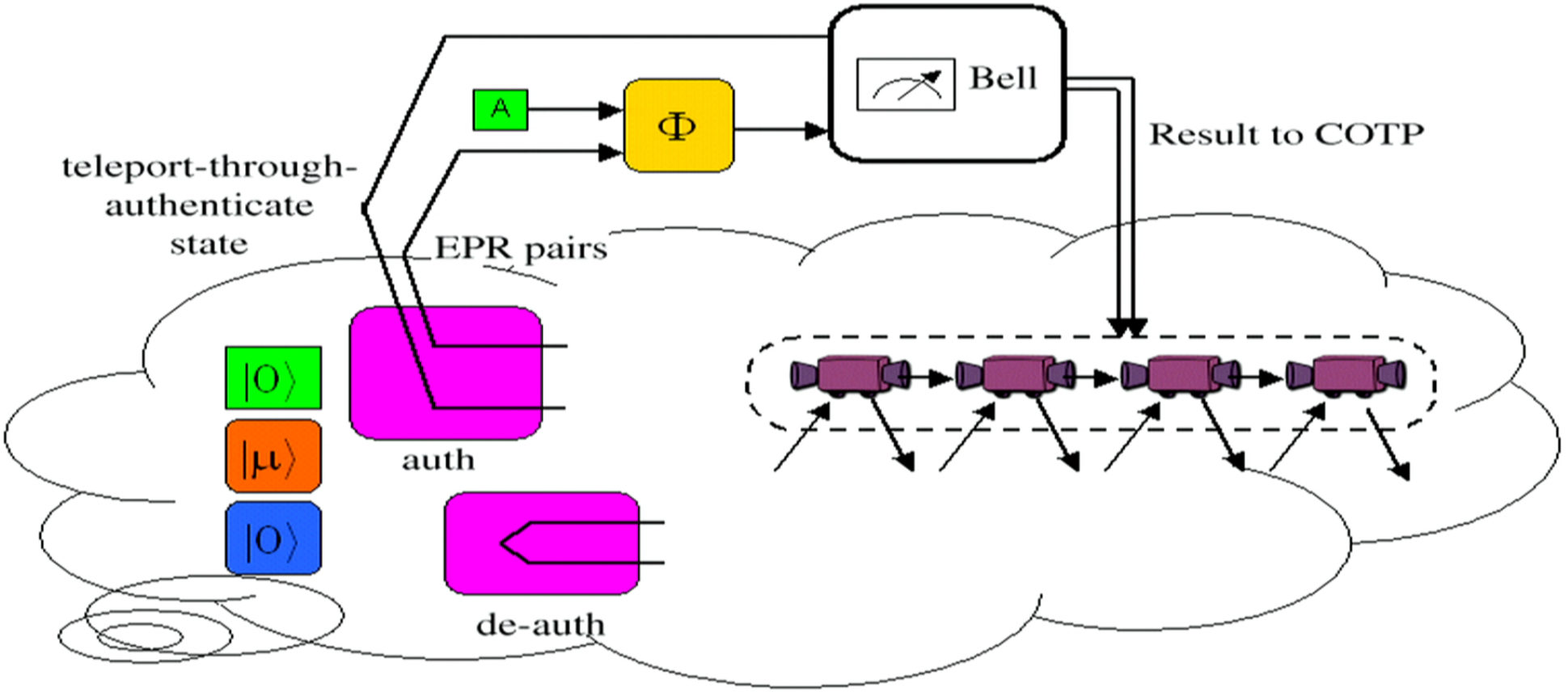


Simulator in pictures



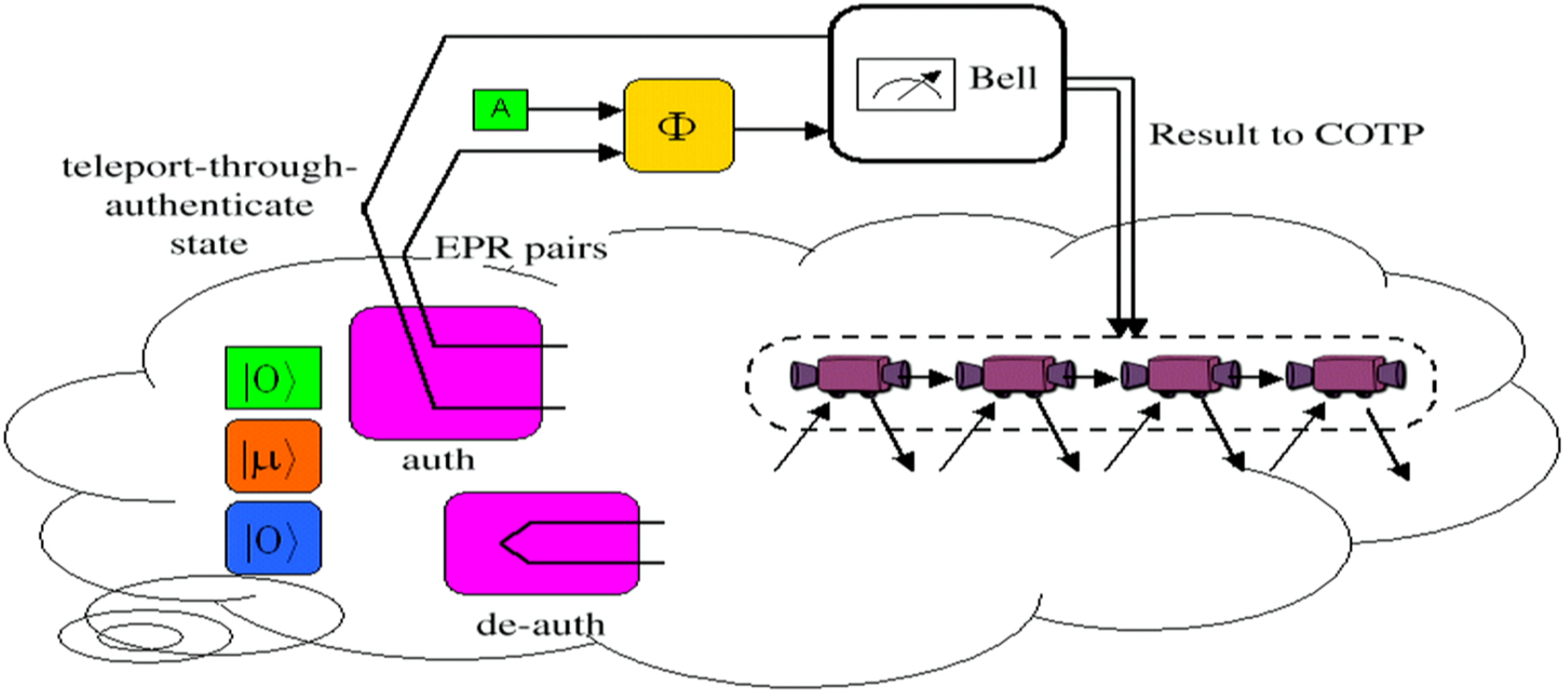
Insert Φ at the beginning, run a dummy computation.

Simulator in pictures



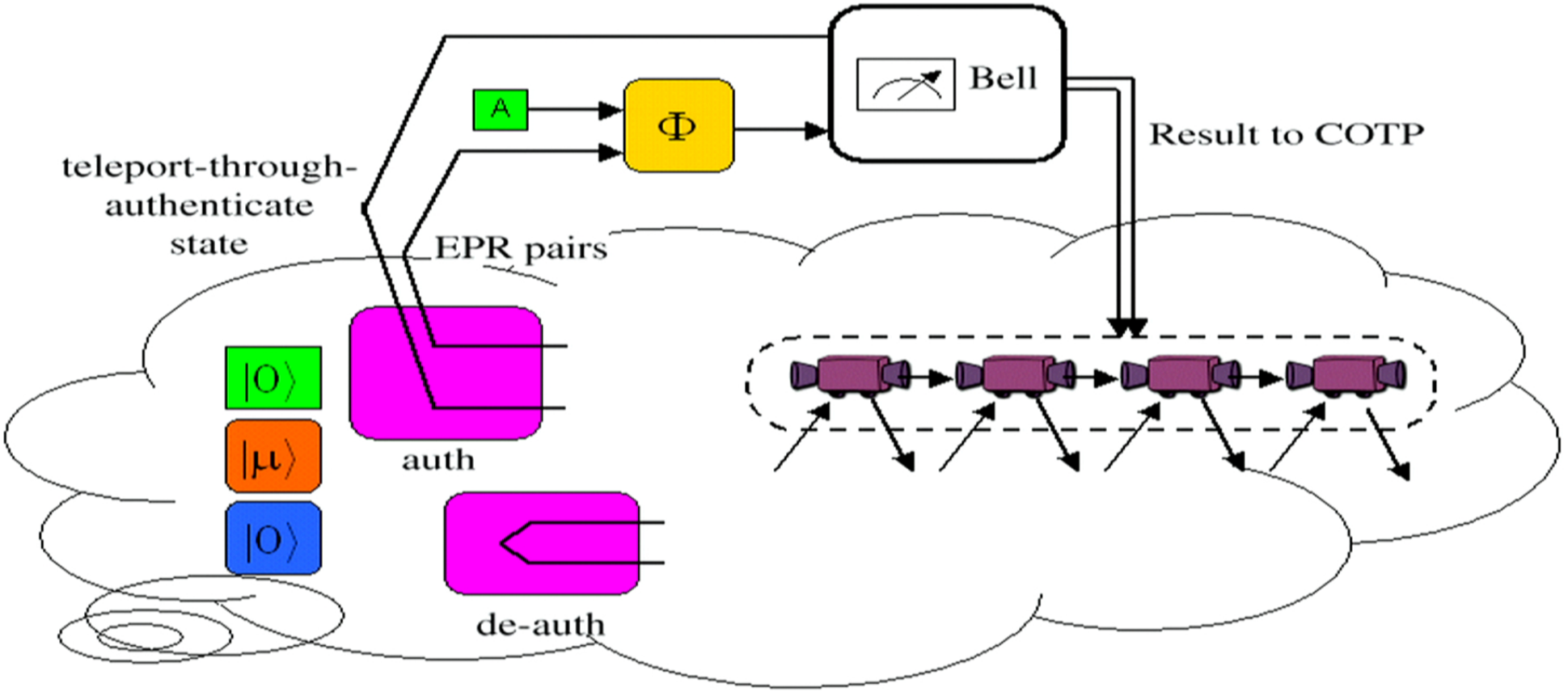
Insert Φ at the beginning, run a dummy computation.

Simulator in pictures



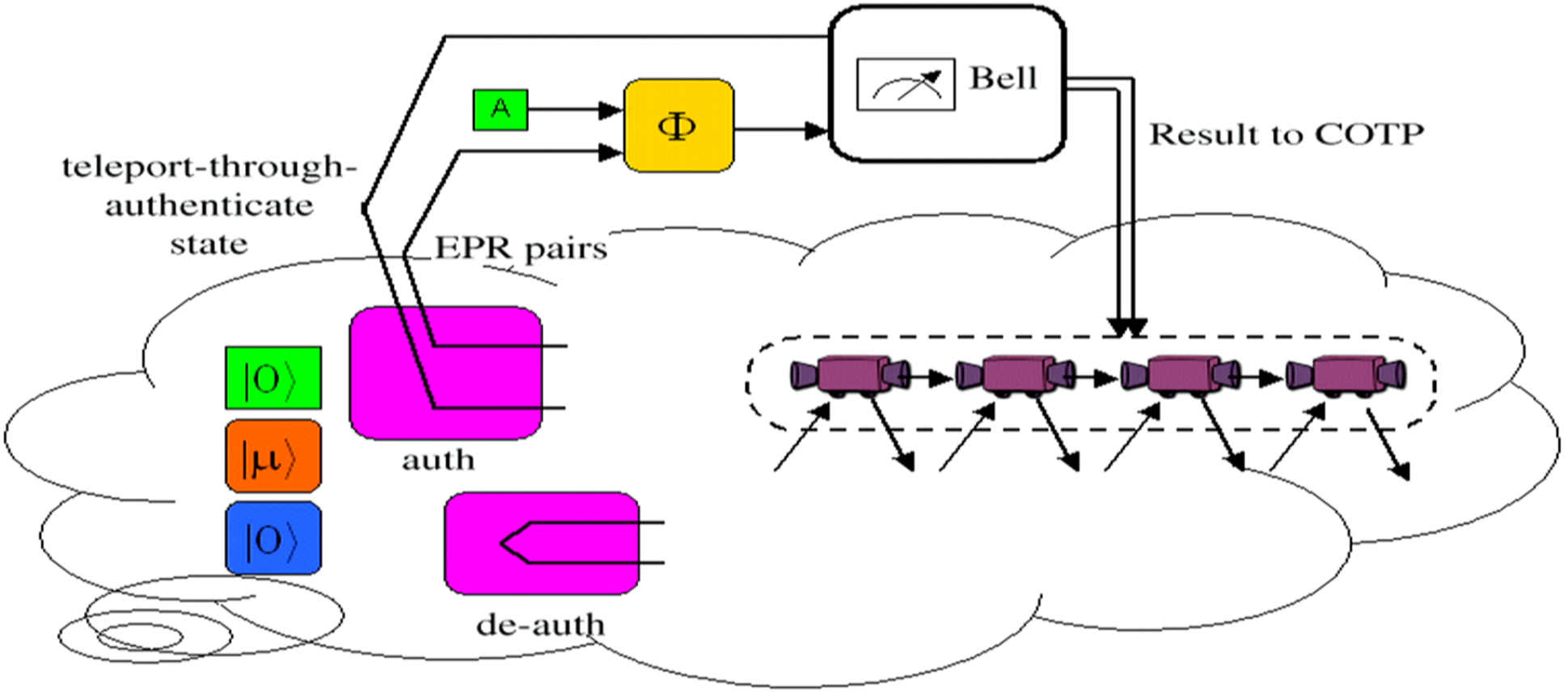
Insert Φ at the beginning, run a dummy computation.

Simulator in pictures



Insert Φ at the beginning, run a dummy computation.

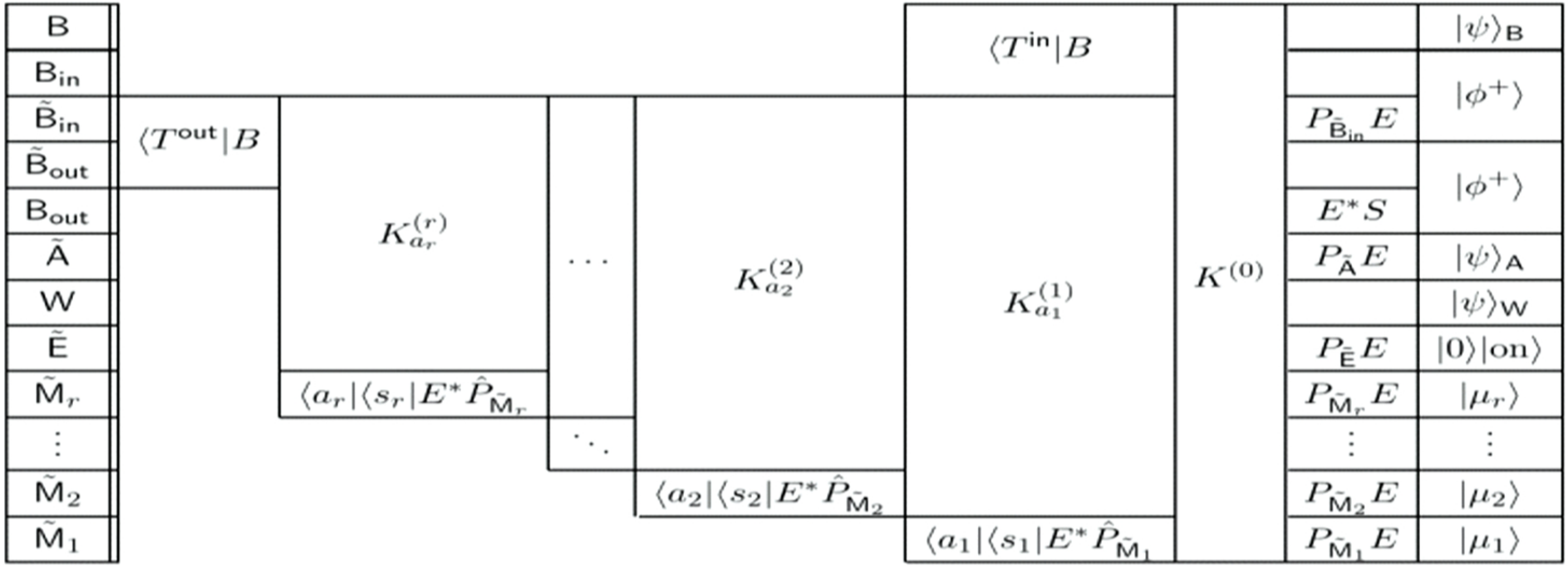
Simulator in pictures



Insert Φ at the beginning, run a dummy computation.

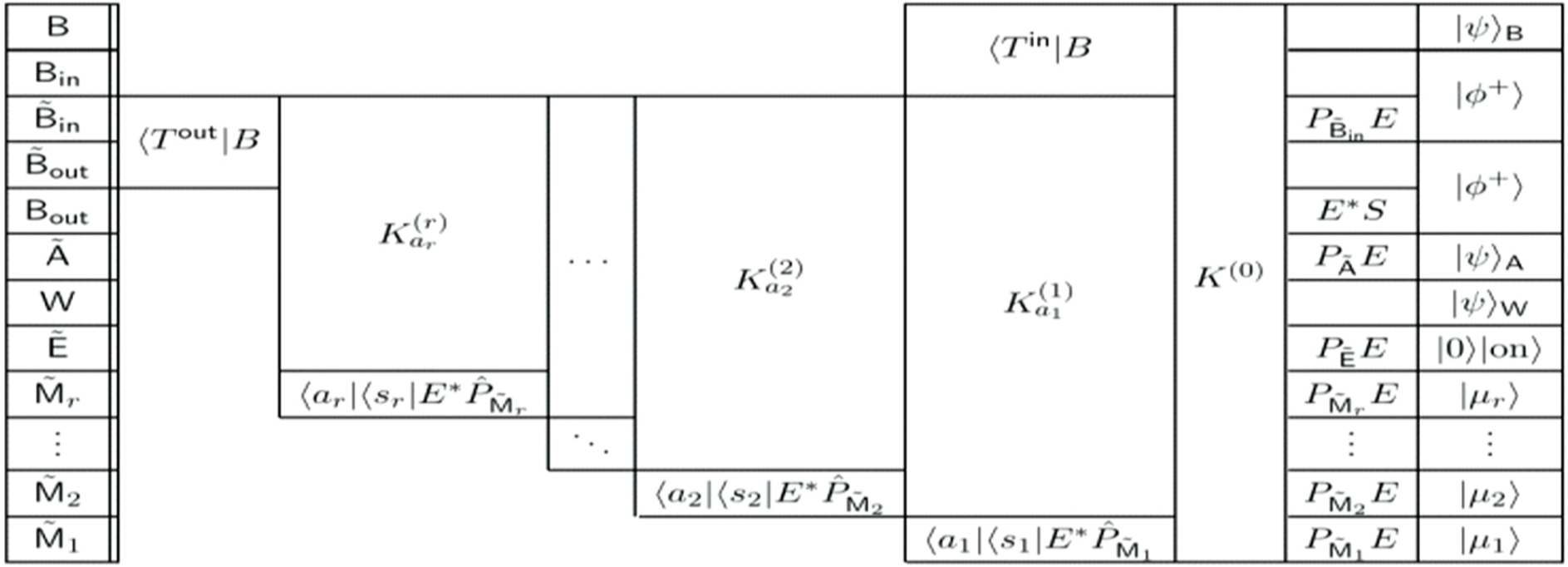
Tabular representation for large states

A composition of product operators acting on a multi-register state:



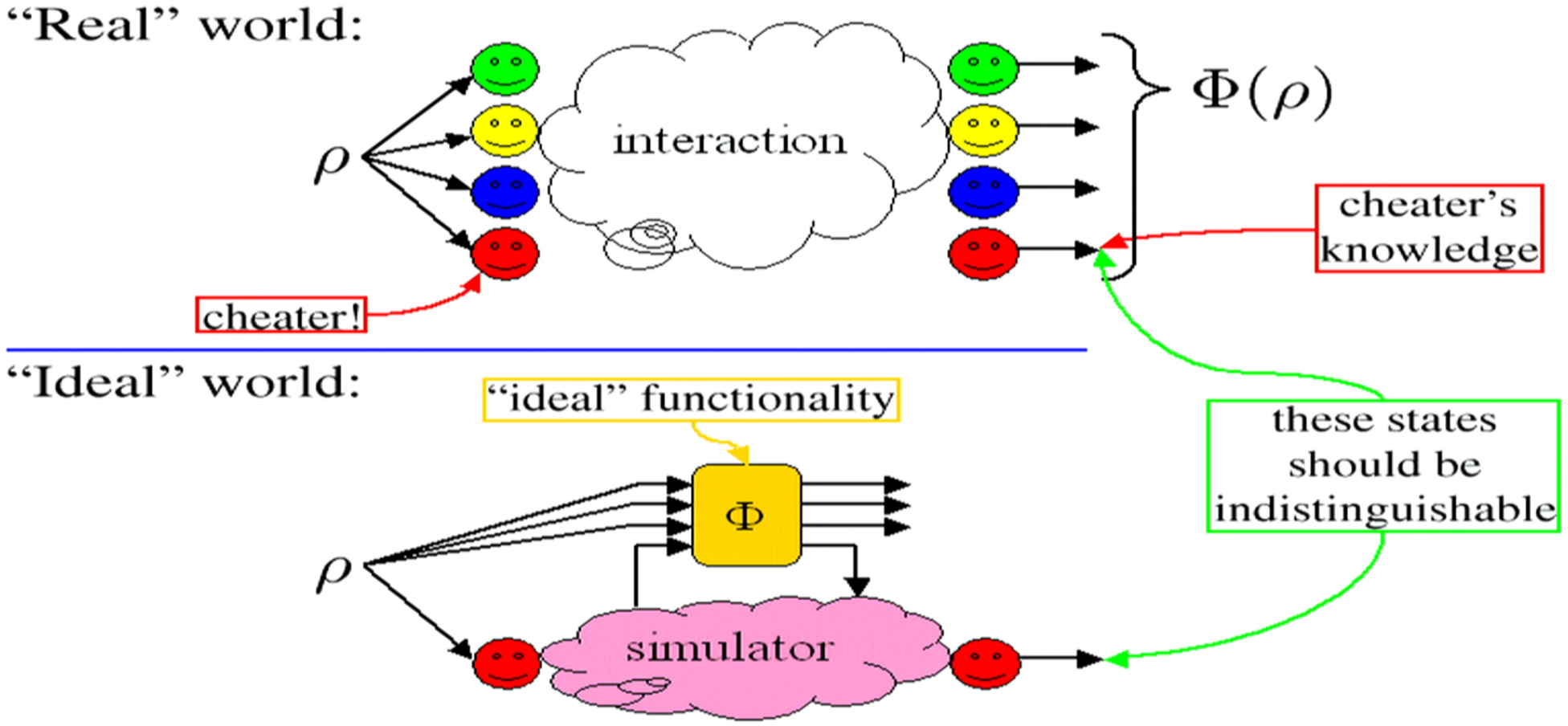
Tabular representation for large states

A composition of product operators acting on a multi-register state:



The origins of simulation-based security

Example: multi-party computation



History of QCAD

Few authentication schemes are known to admit QCAD:

1. Polynomial scheme (encode-encrypt, qudits).
   [Ben-Or, Crépeau, Gottesman, Hassidim, Smith 2006]

2. Clifford scheme.
   [Dupuis, Nielsen, Salvail 2012]

3. Trap scheme (encode-encrypt).
   [this talk]

The trap scheme is a relatively simple qubit scheme.

History of QCAD

Few authentication schemes are known to admit QCAD:

1. Polynomial scheme (encode-encrypt, qudits).
   [Ben-Or, Crépeau, Gottesman, Hassidim, Smith 2006]

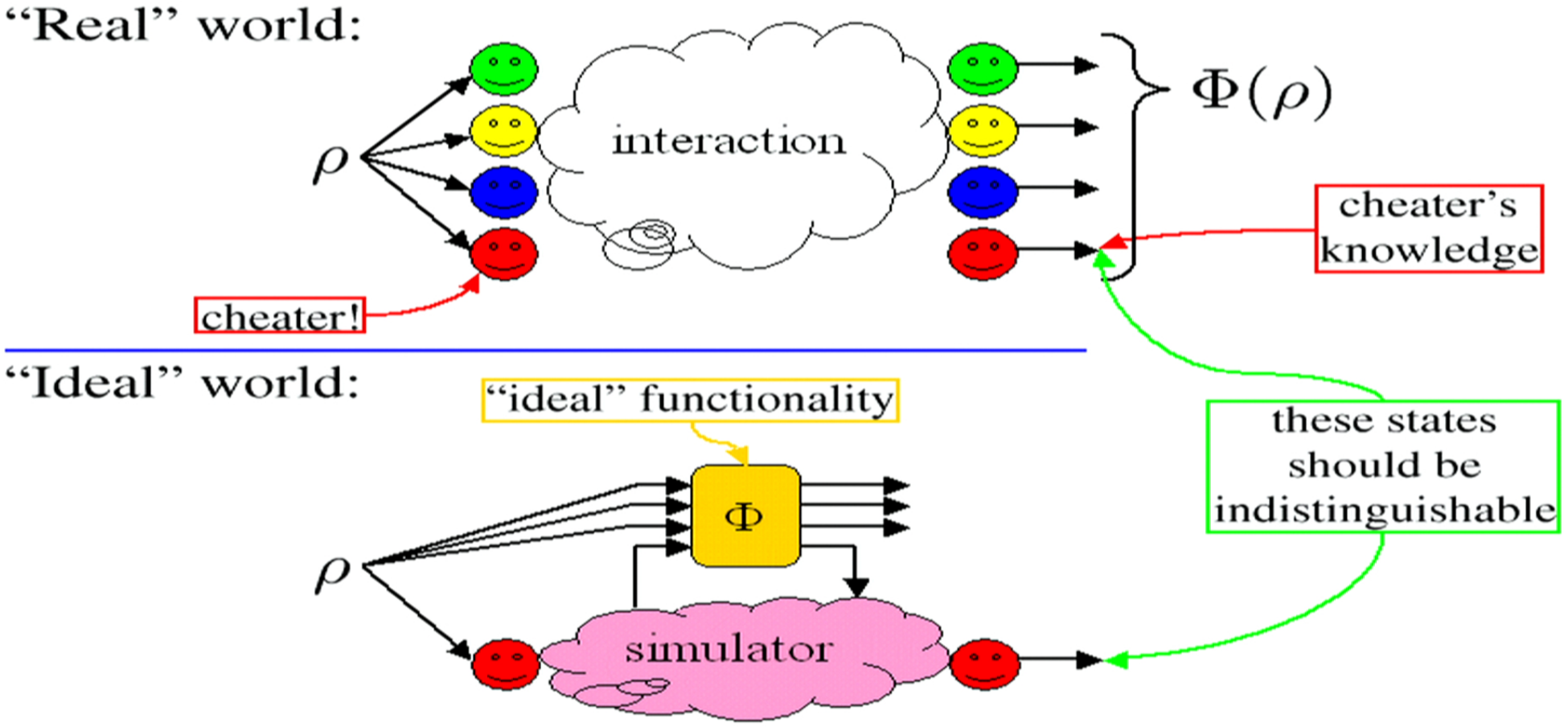
2. Clifford scheme.
   [Dupuis, Nielsen, Salvail 2012]

3. Trap scheme (encode-encrypt).
   [this talk]

The trap scheme is a relatively simple qubit scheme.

The origins of simulation-based security

Example: multi-party computation



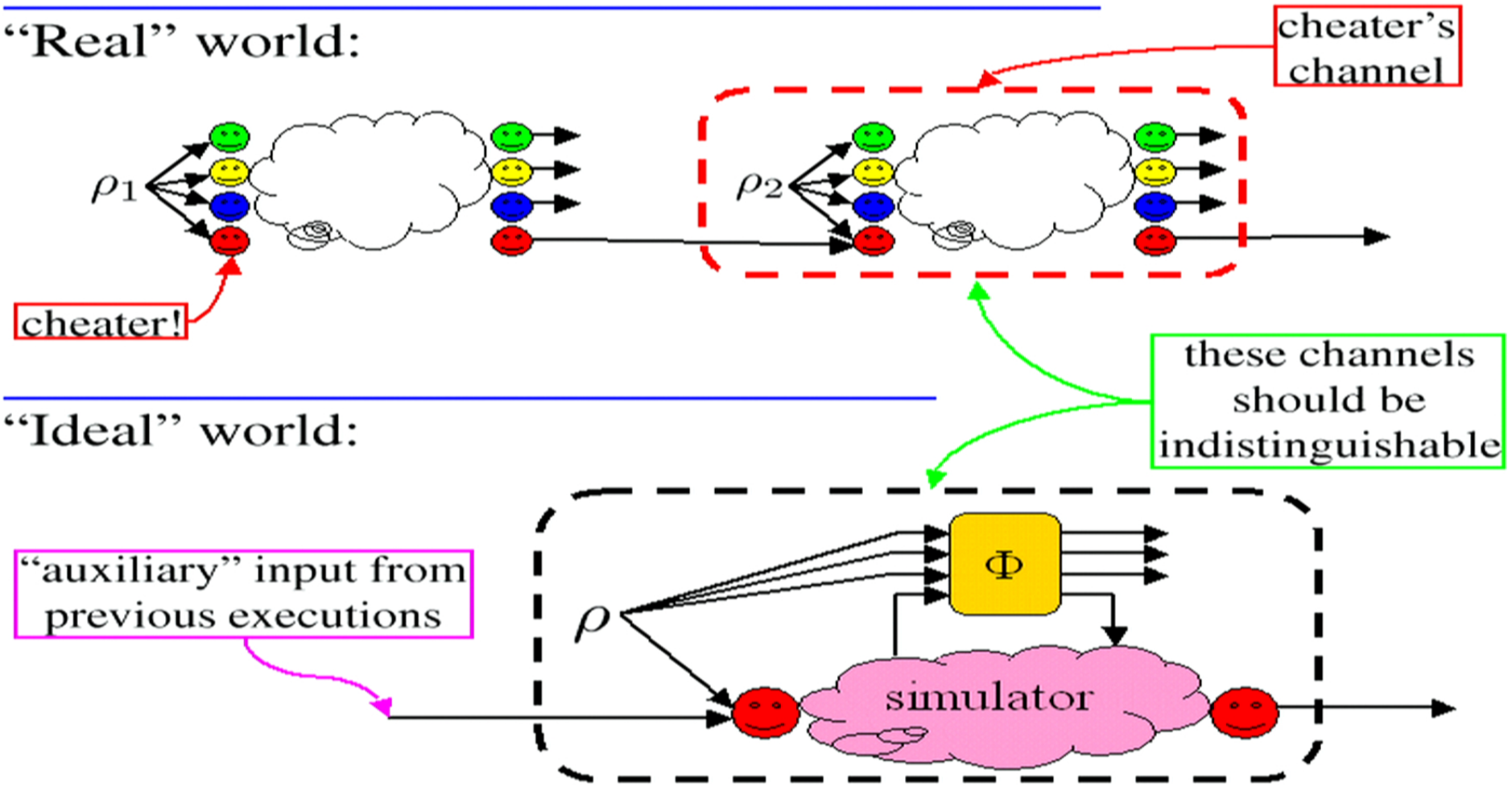
Stand-alone (weak) security

ρ_{real}: cheater's state in the "real" world.
ρ_{sim}: cheater's simulated state.

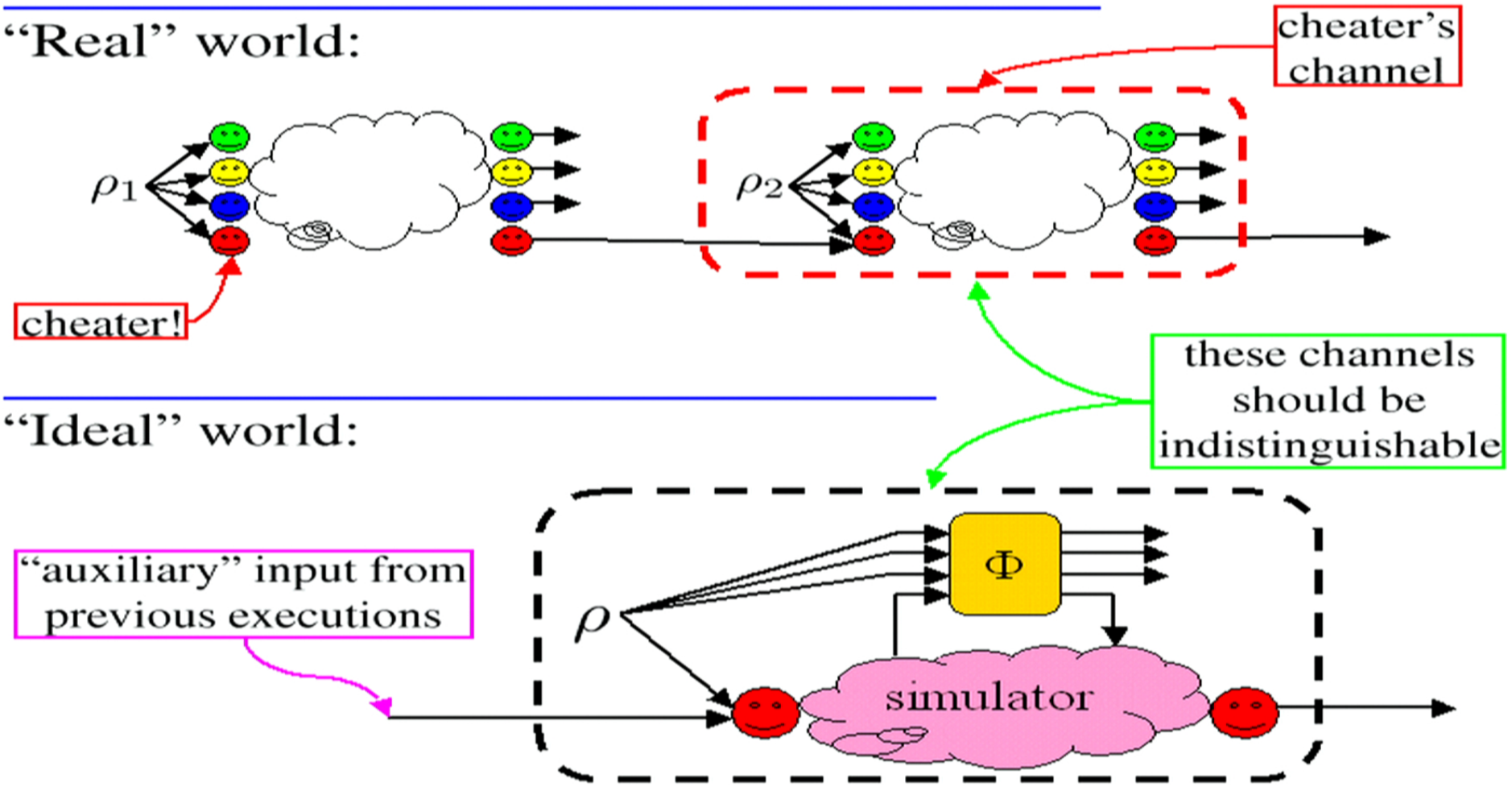
Definition. A protocol is *stand-alone secure* if $\forall$ cheaters $\exists$ simulator such that $\| \rho_{\text{real}} - \rho_{\text{sim}} \|_{\text{Tr}} <$ small.

Q: What if the protocol is executed twice?

Repeated protocols induce channels



Repeated protocols induce channels



Sequential (stronger) security

Φ_{real}: cheater's channel in the "real" world.
Φ_{sim}: cheater's simulated channel.

Definition. A protocol is *sequentially secure* if $\forall$ cheaters $\exists$ simulator such that $\| \Phi_{\text{real}} - \Phi_{\text{sim}} \|_{\diamond} <$ small.

stand-alone security $\neq$ sequential security

Q: What if two protocols are executed in parallel?

Three flavours of security

Stand-alone. Trace norm $\| \rho_{\text{real}} - \rho_{\text{sim}} \|_{\text{Tr}}$.



Sequential. Diamond norm $\| \Phi_{\text{real}} - \Phi_{\text{sim}} \|_{\diamond}$.



Universally composable. Strategy r-norm $\| S_{\text{real}} - S_{\text{sim}} \|_{\diamond r}$.



Notes on the definition of UC

1. The original definition of UC is very complicated.

2. This new definition invokes the *strategy* formalism of [G. Watrous 2007]. (See also [Chiribella, D'Ariano, Perinotti 2008].)

3. Our security proof for QOTPs does not invoke the strategy formalism. :(

4. There is potential for dramatic simplification of UC via strategies. ("Drive-by shooting.")

Notes on the definition of UC

1. The original definition of UC is very complicated.

2. This new definition invokes the *strategy* formalism of [G. Watrous 2007]. (See also [Chiribella, D'Ariano, Perinotti 2008].)

3. Our security proof for QOTPs does not invoke the strategy formalism. :(

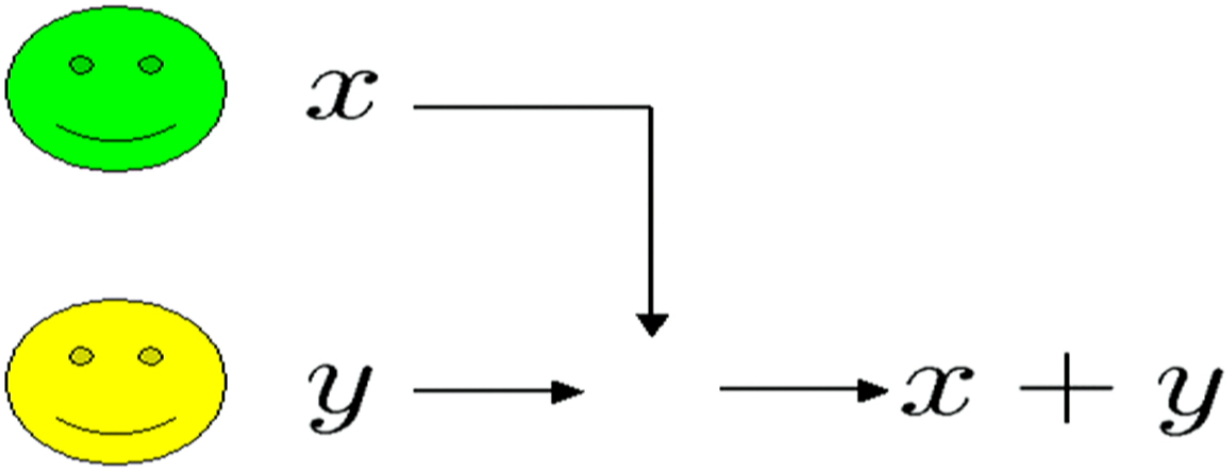
4. There is potential for dramatic simplification of UC via strategies. ("Drive-by shooting.")

Trivial one-time programs

Consider the function

$$f : (x, y) \mapsto x + y.$$

Claim. f admits a tivial one-time program with no hardware assumptions!



Unlockable classical functions



Definition. A function $f(x, y)$ is *unlockable* if $\exists$ a "key" input y_0 and a "recovery algorithm" $\mathcal{A}$ such that $\forall\ x, y$ it holds that $\mathcal{A}(f(x, y_0), y) = f(x, y)$.

Theorem. f admits a one-time program with no hardware assumptions if and only if f is unlockable.

Unlockable functions are "pathological."

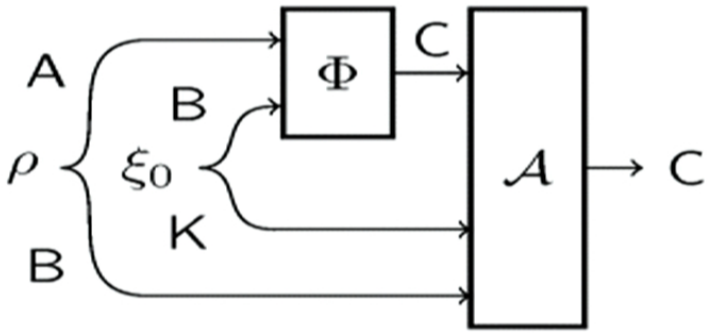


Unlockable quantum channels

Definition. A channel Φ is *strongly unlockable* if $\exists$ "key" state ξ_0 and a channel $\mathcal{A}$ as in (a).

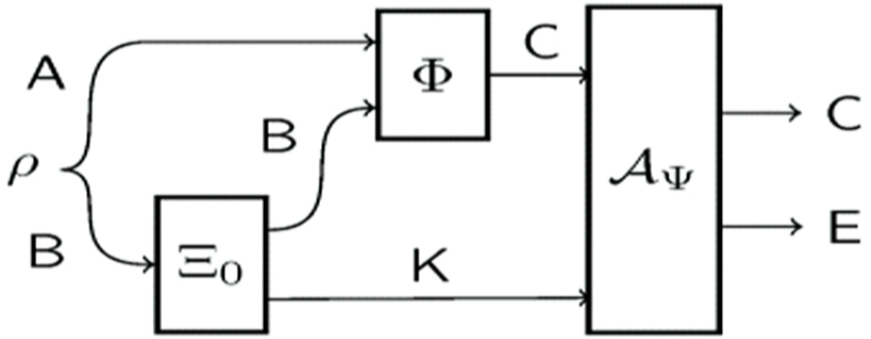
Definition. A channel Φ is *weakly unlockable* if
$\exists$ "key" channel Ξ_0 s.t.
$\forall$ receiver channels Ψ on B
$\exists$ channel $\mathcal{A}_\Psi$ as in (b).

(a) Strongly unlockable



output $= \Phi(\rho)$

(b) Weakly unlockable



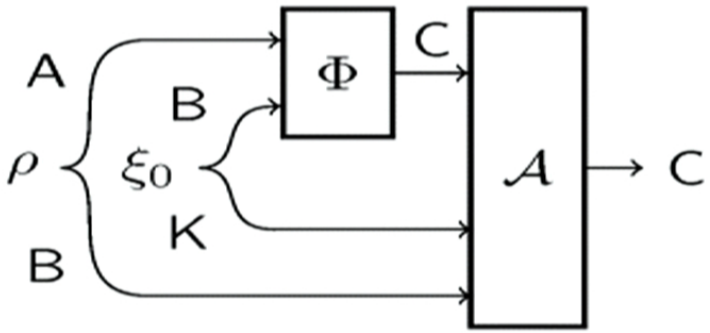
output $= \Phi(\Psi(\rho))$

Unlockable quantum channels

Definition. A channel Φ is *strongly unlockable* if $\exists$ "key" state ξ_0 and a channel $\mathcal{A}$ as in (a).

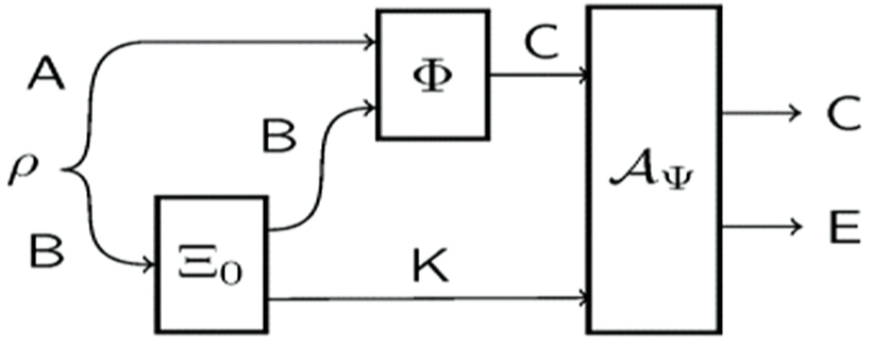
Definition. A channel Φ is *weakly unlockable* if
$\exists$ "key" channel Ξ_0 s.t.
$\forall$ receiver channels Ψ on B
$\exists$ channel $\mathcal{A}_\Psi$ as in (b).

(a) Strongly unlockable



output $= \Phi(\rho)$

(b) Weakly unlockable

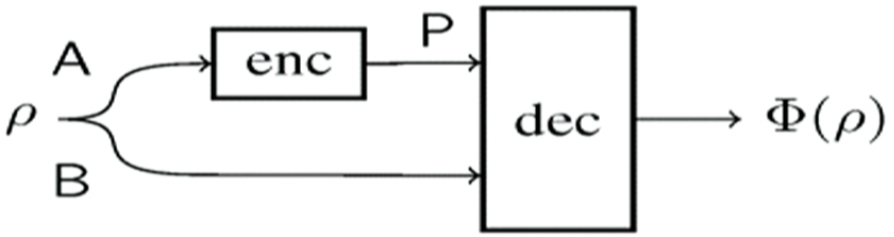


output $= \Phi(\Psi(\rho))$

Trivial QOTP => weakly unlockable

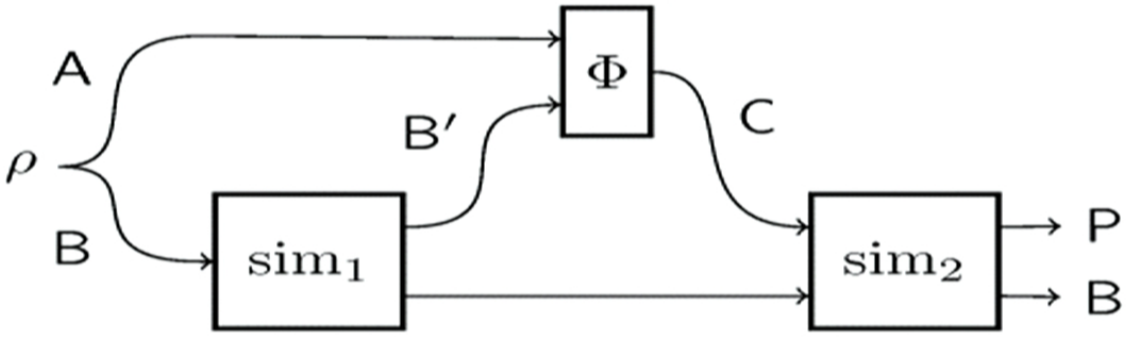
Theorem. If Φ admits a QOTP with no hardware assumptions then Φ is weakly unlockable.

QOTP for Φ:



enc: sender's channel
dec: receiver's channel
P: program register

Simulator recreates the program channel $\mathrm{enc} \otimes \mathbb{1}$:



Trivial QOTP => weakly unlockable

Claim. Φ is weakly unlockable with $\Xi_0 = \mathrm{sim}_1$ and $\mathcal{A}_\Psi = \mathrm{dec} \circ \Psi \circ \mathrm{sim}_2$.

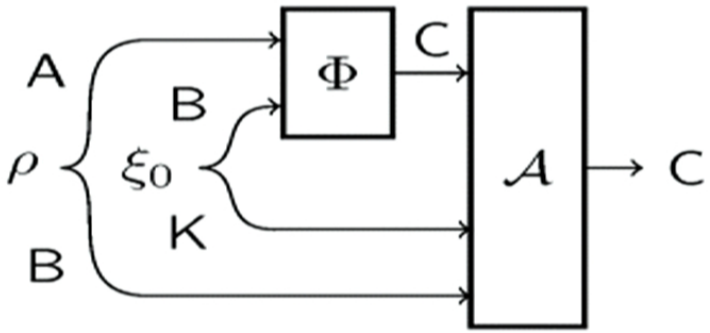
Proof. $A_\Psi \circ \Phi \circ \Xi_0 = \mathrm{dec} \circ \Psi \circ \mathrm{sim}_2 \circ \Phi \circ \mathrm{sim}_1 = \mathrm{dec} \circ \Psi \circ \mathrm{enc} = \mathrm{dec} \circ \mathrm{enc} \circ \Psi = \Phi \circ \Psi$ □

Unlockable quantum channels

Definition. A channel Φ is *strongly unlockable* if $\exists$ "key" state ξ_0 and a channel $\mathcal{A}$ as in (a).

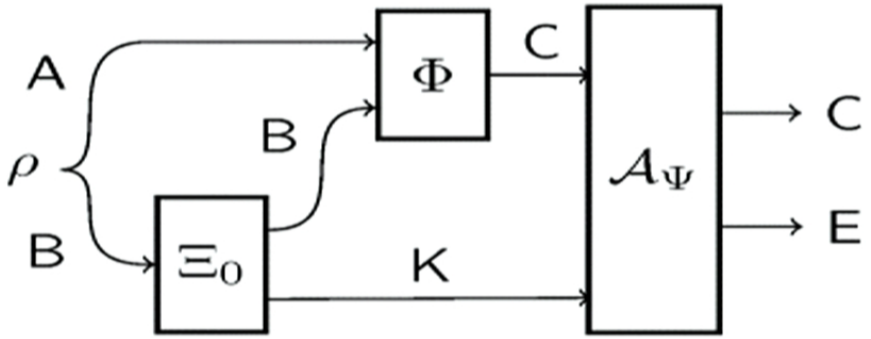
Definition. A channel Φ is *weakly unlockable* if
$\exists$ "key" channel Ξ_0 s.t.
$\forall$ receiver channels Ψ on B
$\exists$ channel $\mathcal{A}_\Psi$ as in (b).

(a) Strongly unlockable



output $= \Phi(\rho)$

(b) Weakly unlockable



output $= \Phi(\Psi(\rho))$

Trivial QOTP => weakly unlockable

Claim. Φ is weakly unlockable with $\Xi_0 = \mathrm{sim}_1$ and $\mathcal{A}_\Psi = \mathrm{dec} \circ \Psi \circ \mathrm{sim}_2$.

Proof. $A_\Psi \circ \Phi \circ \Xi_0 = \mathrm{dec} \circ \Psi \circ \mathrm{sim}_2 \circ \Phi \circ \mathrm{sim}_1 = \mathrm{dec} \circ \Psi \circ \mathrm{enc} = \mathrm{dec} \circ \mathrm{enc} \circ \Psi = \Phi \circ \Psi$ □

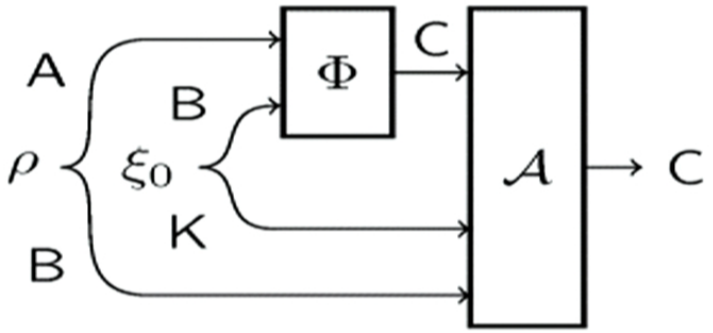
Trivial QOTP => weakly unlockable

Claim. Φ is weakly unlockable with $\Xi_0 = \mathrm{sim}_1$ and $\mathcal{A}_\Psi = \mathrm{dec} \circ \Psi \circ \mathrm{sim}_2$.

Proof. $A_\Psi \circ \Phi \circ \Xi_0 = \mathrm{dec} \circ \Psi \circ \mathrm{sim}_2 \circ \Phi \circ \mathrm{sim}_1 = \mathrm{dec} \circ \Psi \circ \mathrm{enc} = \mathrm{dec} \circ \mathrm{enc} \circ \Psi = \Phi \circ \Psi$ □

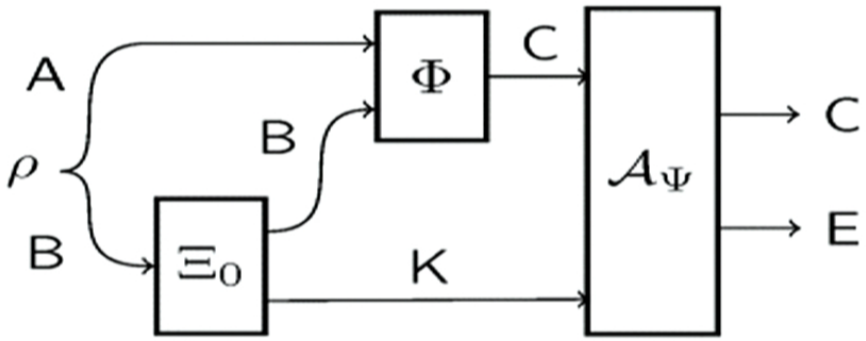
Conjecture: weak = strong unlockability

(a) Strongly unlockable



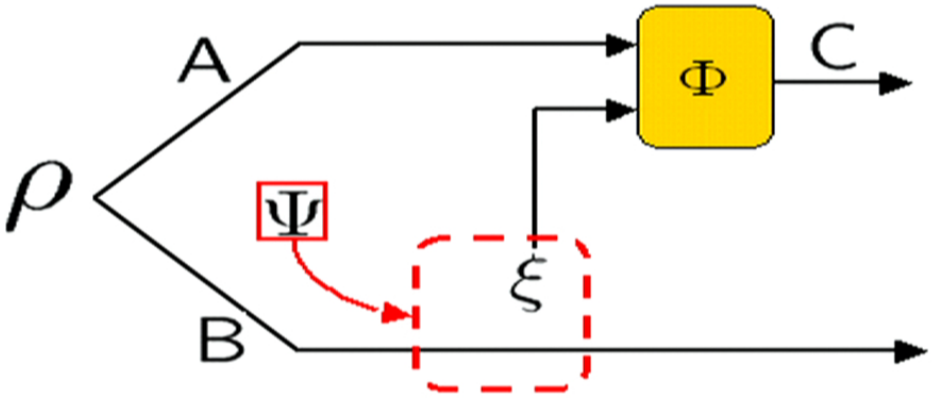
output $= \Phi(\rho)$

(b) Weakly unlockable



output $= \Phi(\Psi(\rho))$

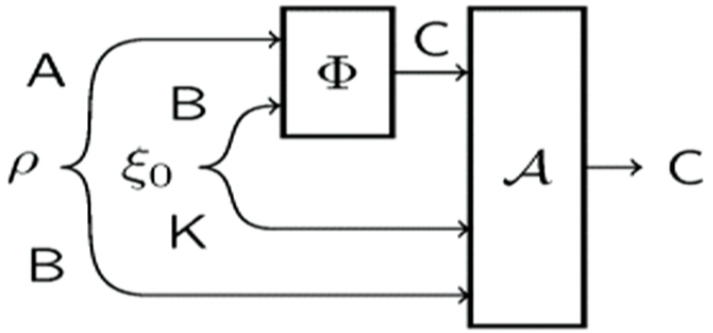
Let Φ be weakly unlockable. Choose Ψ as follows.



Then $\mathcal{A}_\Psi \circ \Phi \circ \Xi_0$ is the identity on B, so $\Phi \circ \Xi_0$ is invertible on B.

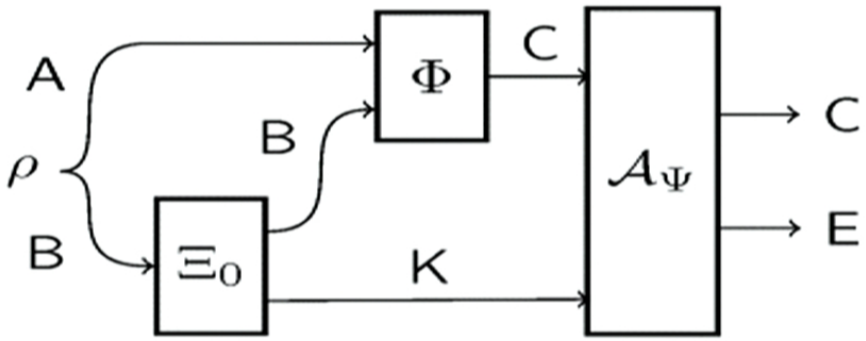
Conjecture: weak = strong unlockability

(a) Strongly unlockable



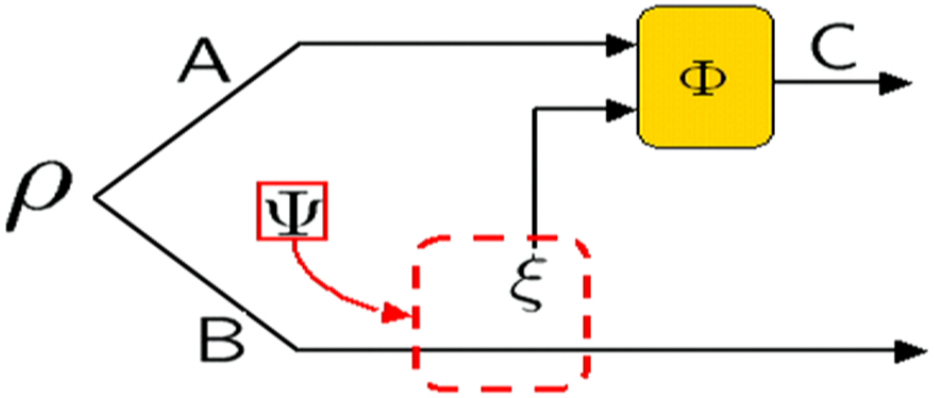
output $= \Phi(\rho)$

(b) Weakly unlockable



output $= \Phi(\Psi(\rho))$

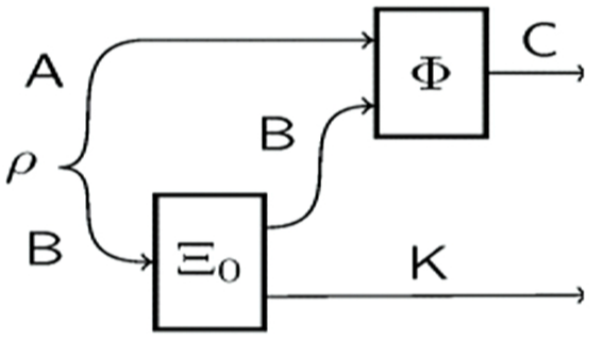
Let Φ be weakly unlockable. Choose Ψ as follows.



Then $\mathcal{A}_\Psi \circ \Phi \circ \Xi_0$ is the identity on B, so $\Phi \circ \Xi_0$ is invertible on B.

Conjecture: weak = strong unlockability

Invertible on B:



Assume WOLOG that
$\Xi_0 : \rho \mapsto S\rho S^*$
for some isometry S.

Φ must be invertible on the subspace $\mathcal{V}$ of B containing $\mathrm{image}(\mathrm{Tr}_{\mathsf{K}}(S|\psi\rangle\langle\psi|S^*)$ for all $|\psi\rangle$.

Swap $\mathcal{V}$ to K and replace it with a dummy state ξ_0. Φ is strongly unlockable with key state ξ_0.

I dare you to prove me wrong!

The end

Thank you for your attention!





