On the power of classical control

Elham Kashefi

THE UNIVERSITY *of* EDINBURGH
informatics

The question

Can program be "quantised" same as data ?

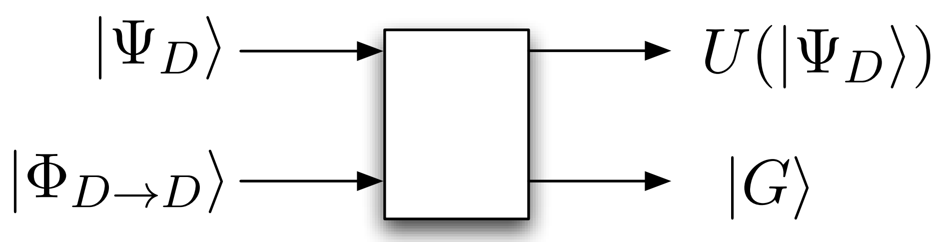
The question

Can program be "quantised" same as data ?

$$D = D \to D$$

Programmable Quantum Circuit

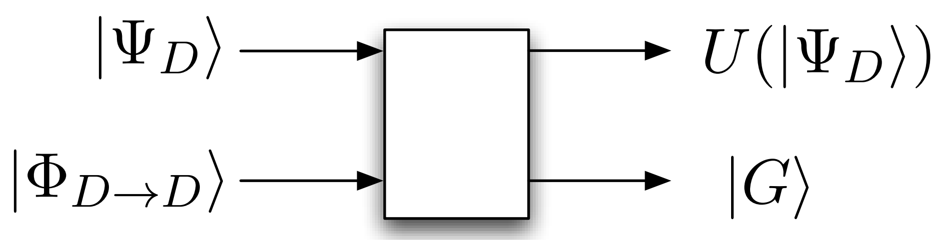
There exist no *universal* quantum processor



Nielsen and Chuang Phys. Rev. Lett. 97

Programmable Quantum Circuit

There exist no *universal* quantum processor

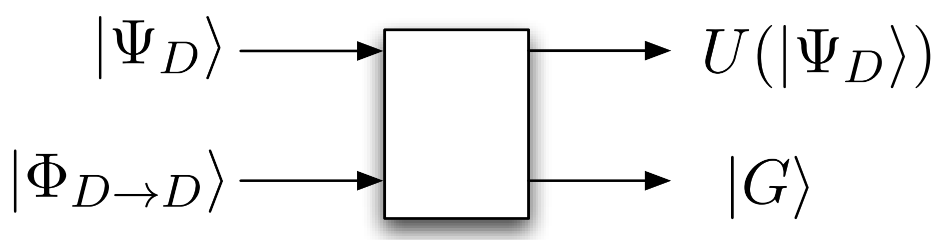


Orthogonal program states hence classical states

Nielsen and Chuang Phys. Rev. Lett. 97

Programmable Quantum Circuit

There exist no *universal* quantum processor



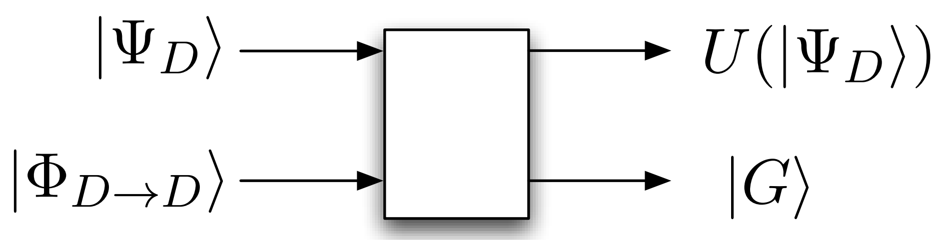
Orthogonal program states hence classical states

A new dimension is needed for each unitary operators

Nielsen and Chuang Phys. Rev. Lett. 97

Programmable Quantum Circuit

There exist no *universal* quantum processor



Orthogonal program states hence classical states

A new dimension is needed for each unitary operators

No higher order function

Nielsen and Chuang Phys. Rev. Lett. 97

Probabilistic Programmable QC

Storing quantum dynamics in quantum states

Vidal, Masanes, Cirac, Phys. Rev. Lett. 2002
Hillery, Ziman, Buzek, Phys. Rev. A. 2002

Probabilistic Programmable QC

Storing quantum dynamics in quantum states

Single-qubit program state

$$|\alpha\rangle \equiv \frac{1}{\sqrt{2}}(e^{i\alpha}|0\rangle + e^{-i\alpha}|1\rangle),$$

Vidal, Masanes, Cirac, Phys. Rev. Lett. 2002
Hillery, Ziman, Buzek, Phys. Rev. A. 2002

Probabilistic Programmable QC

Storing quantum dynamics in quantum states

Single-qubit program state

$$|\alpha\rangle \equiv \frac{1}{\sqrt{2}}(e^{i\alpha}|0\rangle + e^{-i\alpha}|1\rangle),$$

↓

$$U_\alpha \equiv \exp(i\alpha\sigma_z)$$

Vidal, Masanes, Cirac, Phys. Rev. Lett. 2002
Hillery, Ziman, Buzek, Phys. Rev. A. 2002

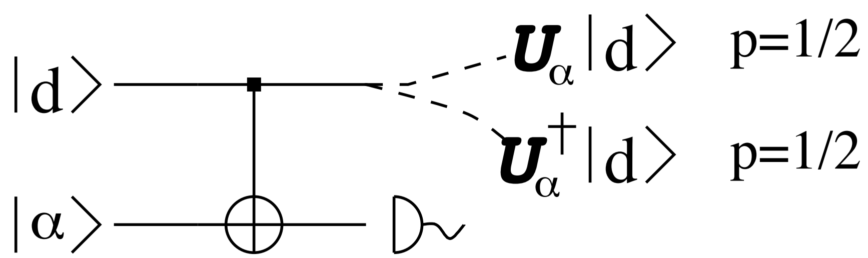
Probabilistic Programmable QC

Storing quantum dynamics in quantum states

Single-qubit program state

$$|\alpha\rangle \equiv \frac{1}{\sqrt{2}}(e^{i\alpha}|0\rangle + e^{-i\alpha}|1\rangle),$$

$$U_\alpha \equiv \exp(i\alpha\sigma_z)$$



Vidal, Masanes, Cirac, Phys. Rev. Lett. 2002
Hillery, Ziman, Buzek, Phys. Rev. A. 2002

Probabilistic Programmable QC

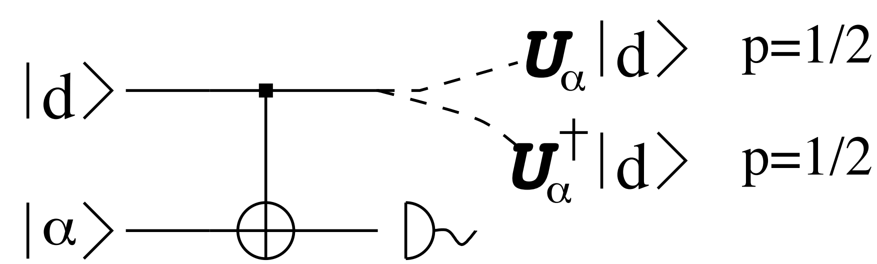
Storing quantum dynamics in quantum states

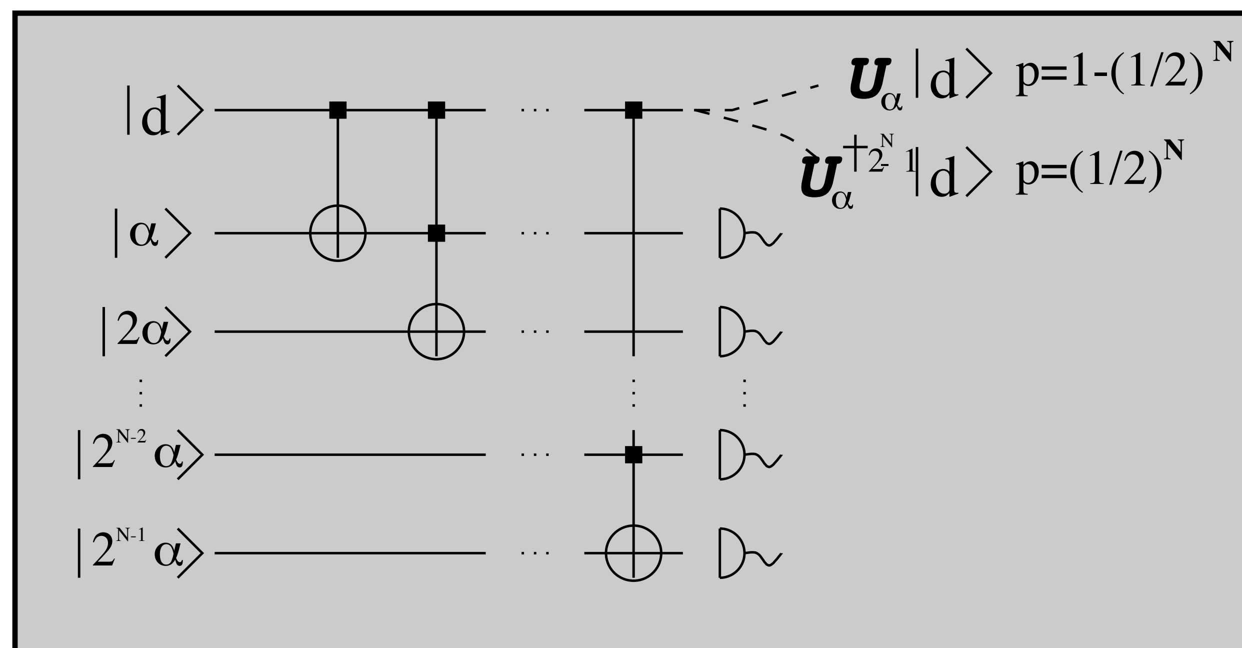
Single-qubit program state

$$|\alpha\rangle \equiv \frac{1}{\sqrt{2}}(e^{i\alpha}|0\rangle + e^{-i\alpha}|1\rangle),$$

↓

$$U_\alpha \equiv \exp(i\alpha\sigma_z)$$





Vidal, Masanes, Cirac, Phys. Rev. Lett. 2002
Hillery, Ziman, Buzek, Phys. Rev. A. 2002

Probabilistic Programmable QC

By providing more information about the program

↓

Several methods to improve performances

Vidal, Masanes, Cirac, Phys. Rev. Lett. 2002
Hillery, Ziman, Buzek, Phys. Rev. A. 2002

Probabilistic Programmable QC

By providing more information about the program

↓

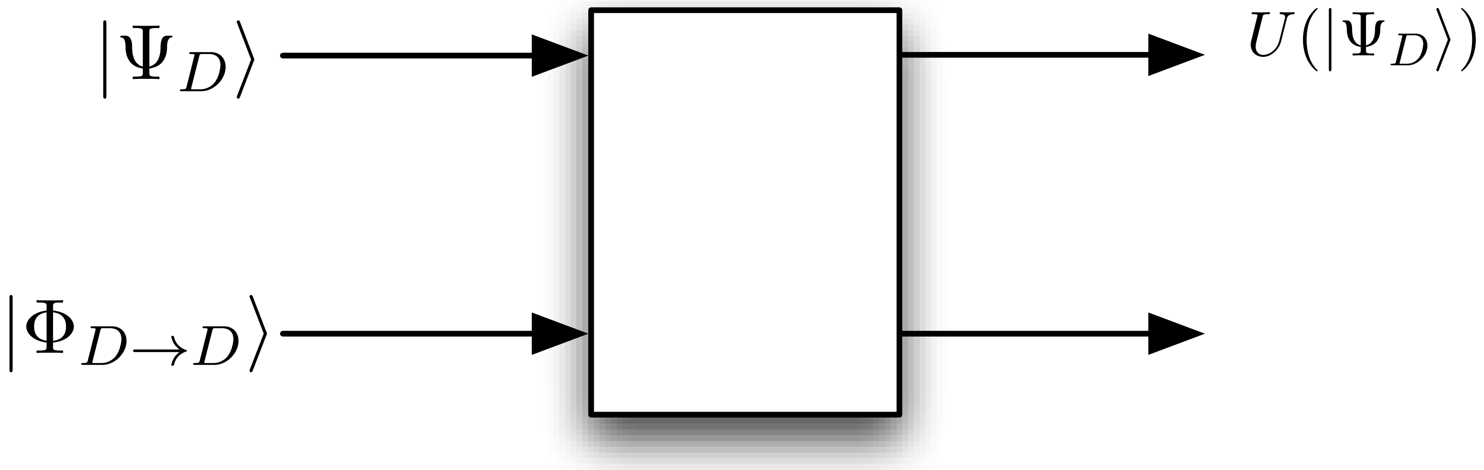
Several methods to improve performances

Trade off

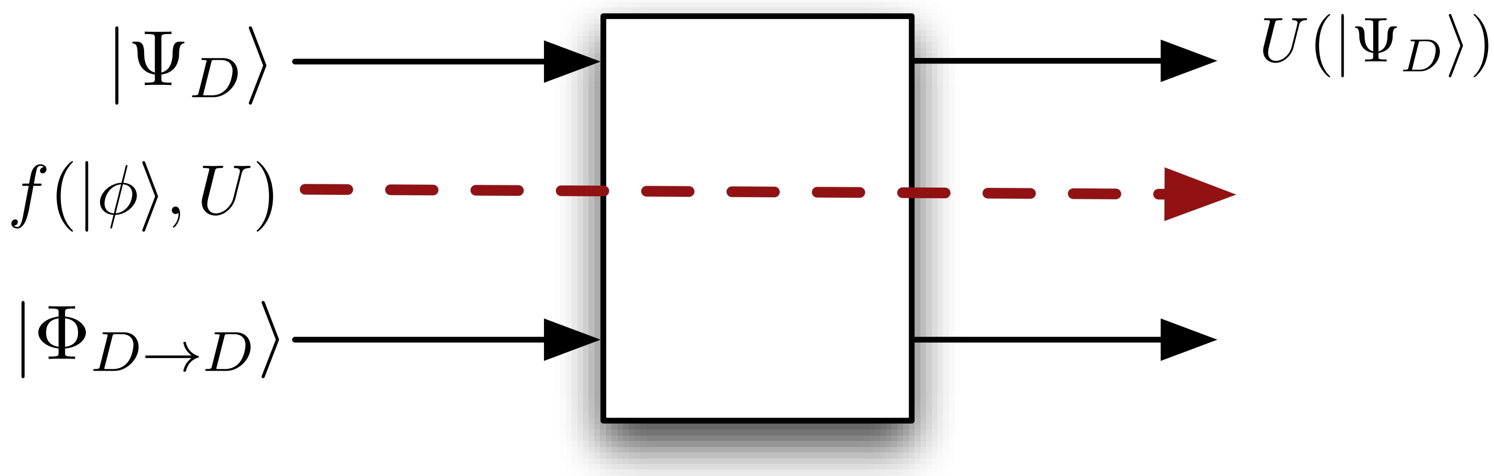
hiding program vs performing program

Vidal, Masanes, Cirac, Phys. Rev. Lett. 2002
Hillery, Ziman, Buzek, Phys. Rev. A. 2002

Adding Classical Control



Adding Classical Control



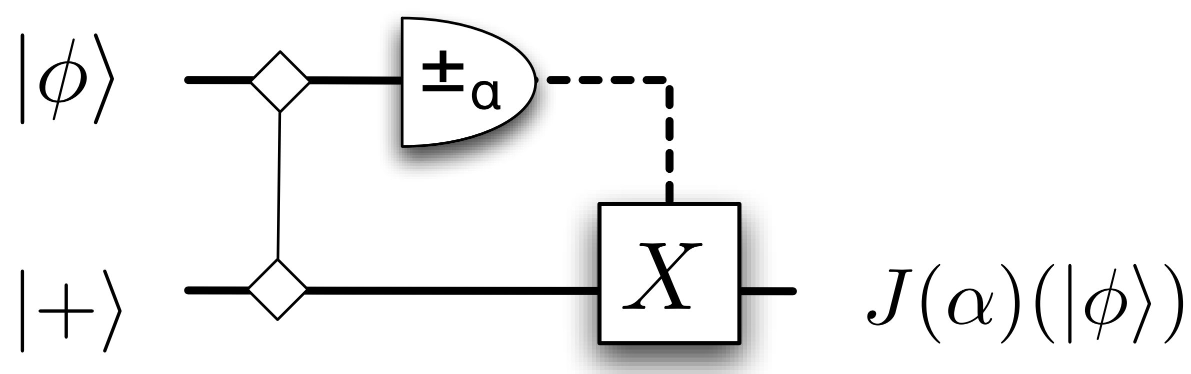
Thinking inside the box

$$J(\alpha) \;:=\; \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix}$$

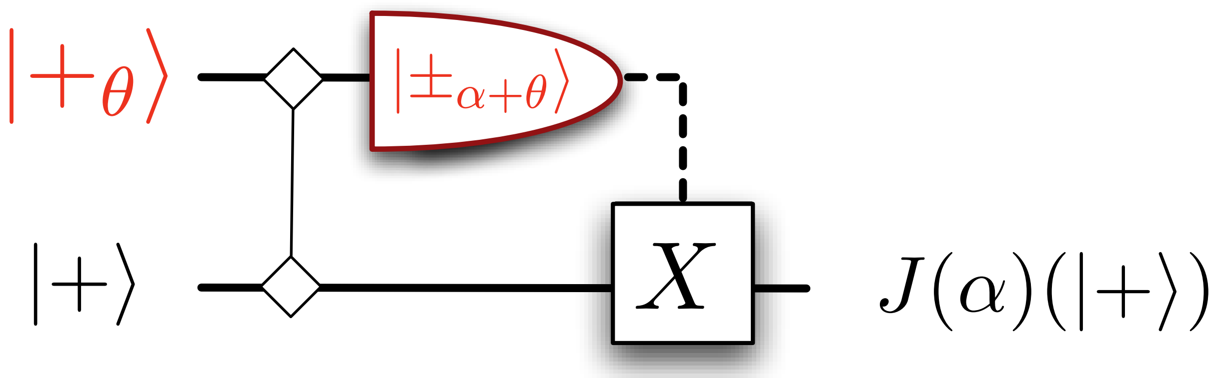
Thinking inside the box

$$J(\alpha) \;:=\; \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha} \end{pmatrix}$$

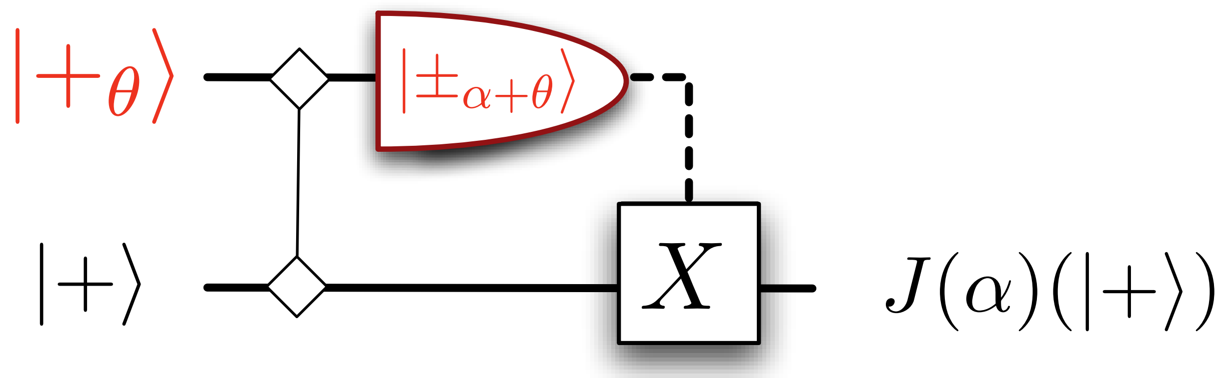
gate teleportation

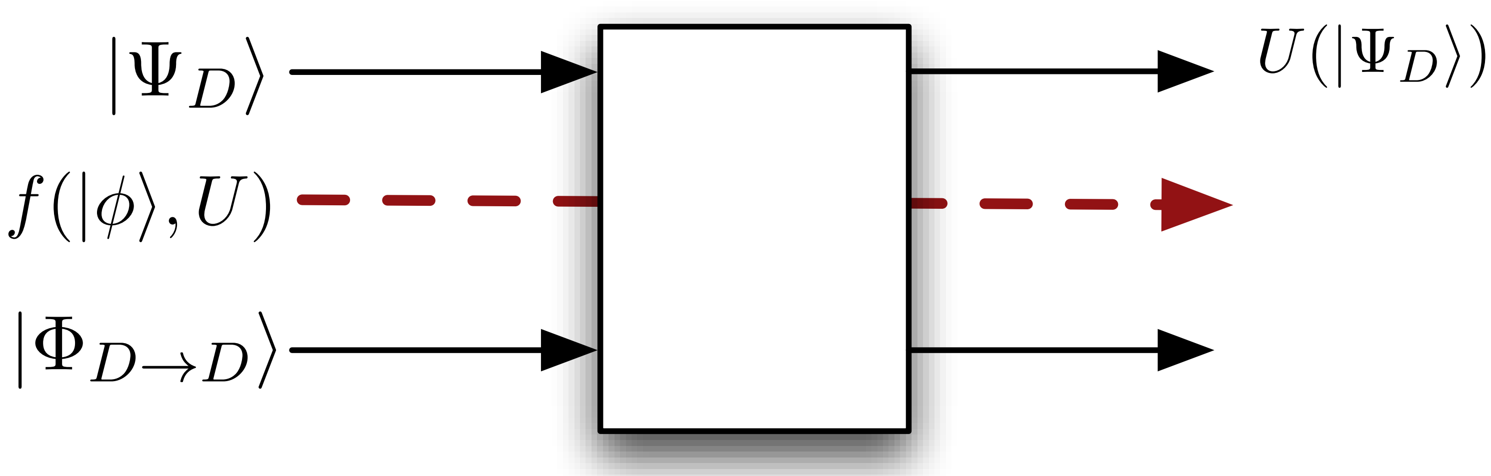


Thinking Inside the box

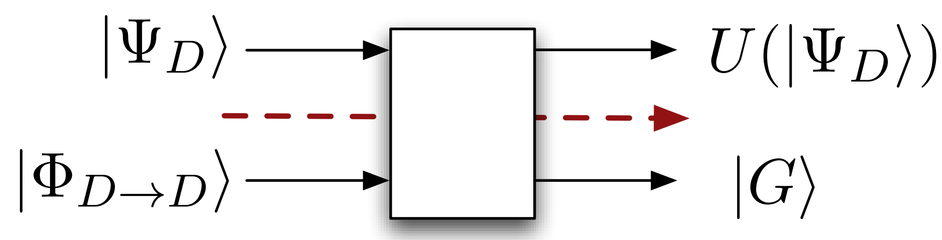


Thinking Inside the box

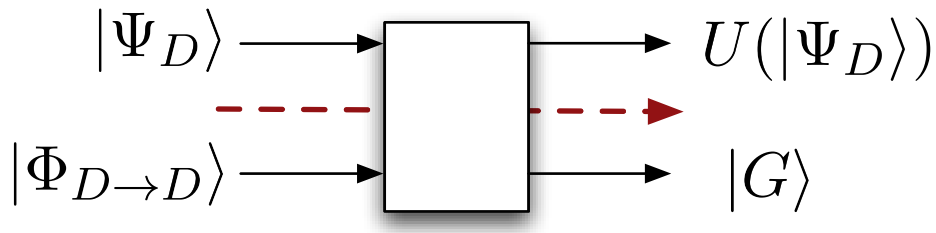




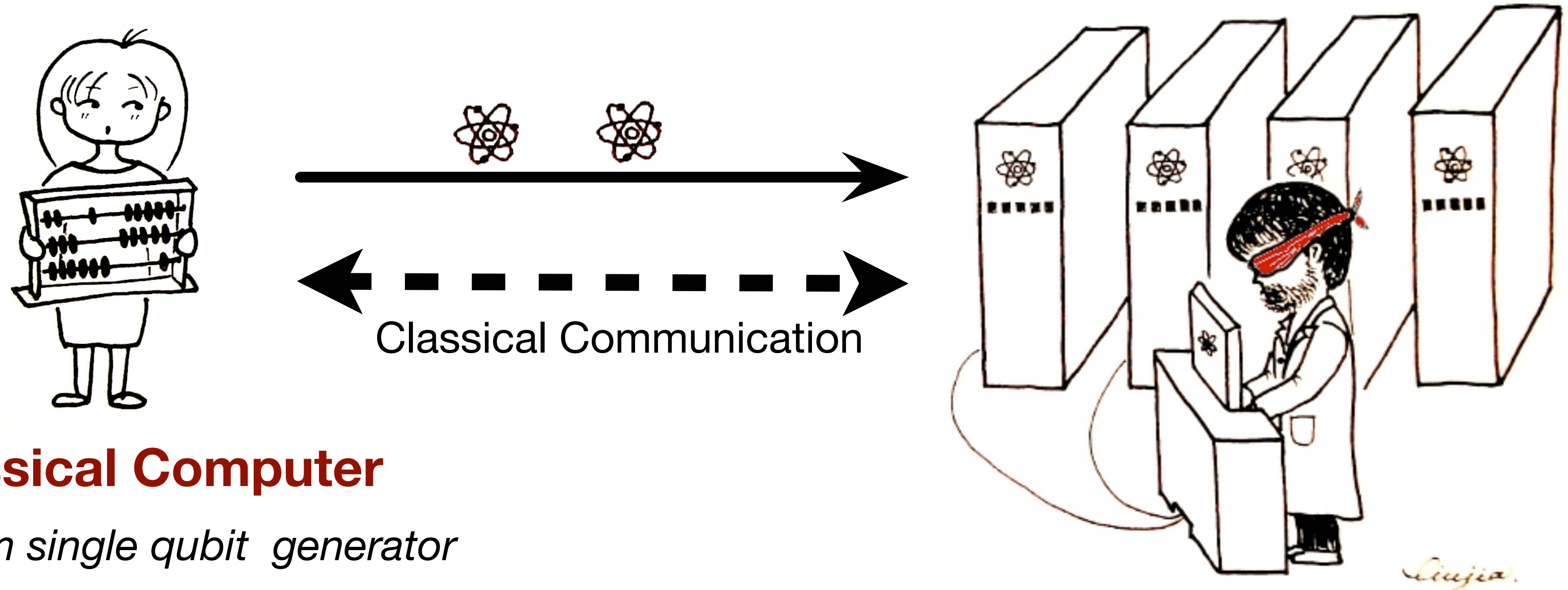
Deterministic Perfectly Hiding Programmable QC



Deterministic Perfectly Hiding Programmable QC



Universal Blind QC



Classical Computer

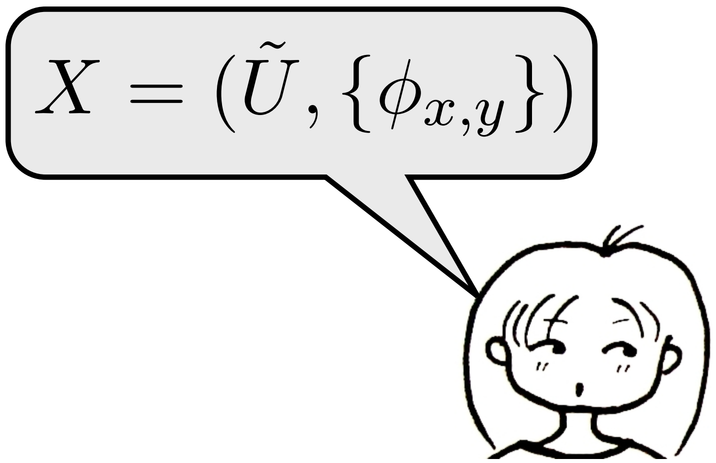
random single qubit generator

Unconditional Perfect Privacy

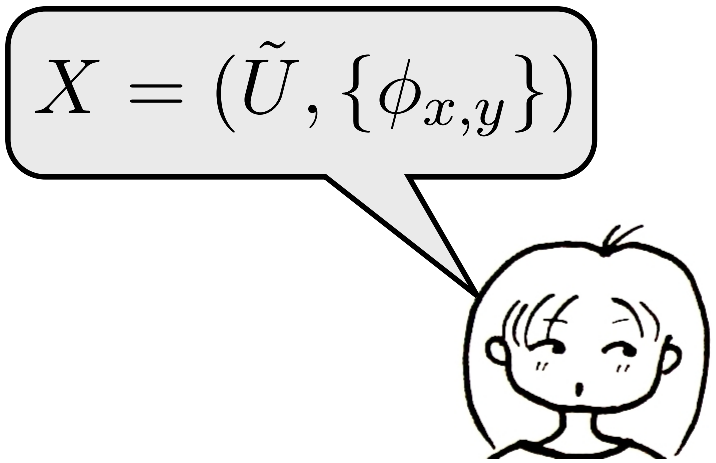
Server learns nothing about client's input/output/computation

Broadbent,Fitzsimons and Kashefi, FOCS. 2010

Universal Blind Quantum Computings



Universal Blind Quantum Computings

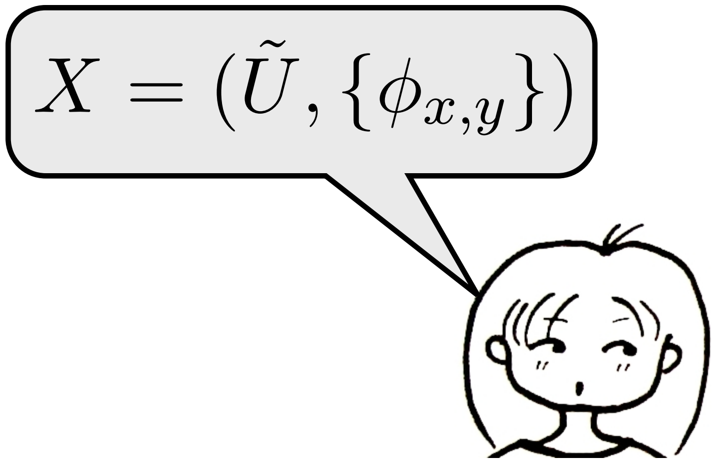


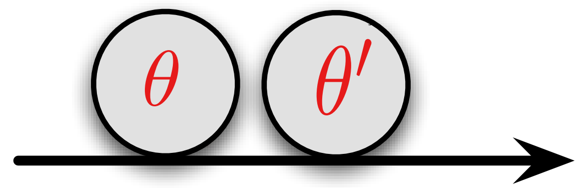
random single qubit generator

$$1/\sqrt{2}\left(|0\rangle + e^{i\theta}|1\rangle\right)$$

$$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$$

Universal Blind Quantum Computings



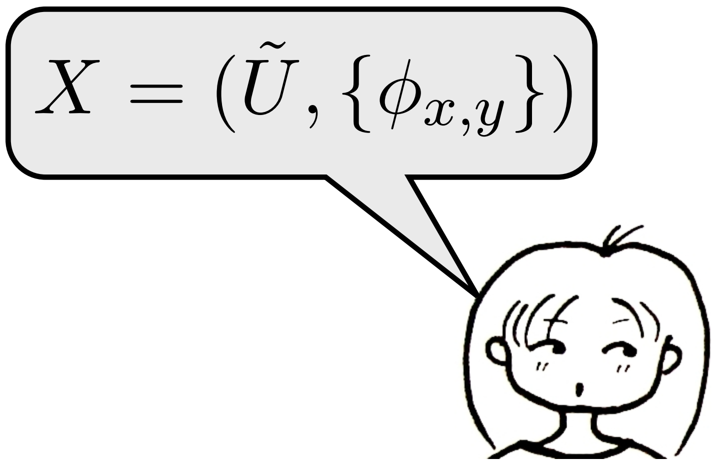


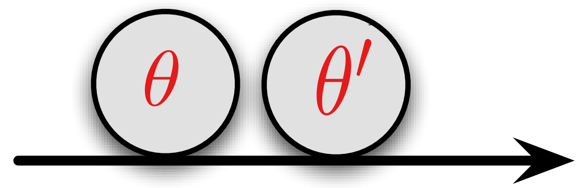
random single qubit generator

$$1/\sqrt{2}\left(|0\rangle + e^{i\theta}|1\rangle\right)$$

$$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$$

Universal Blind Quantum Computings

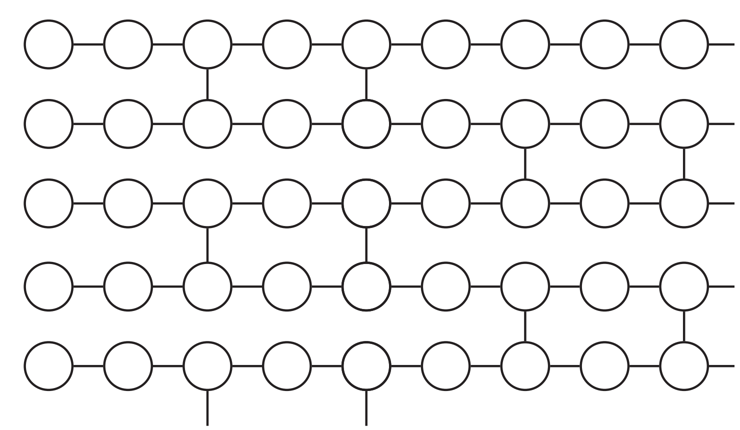




random single qubit generator

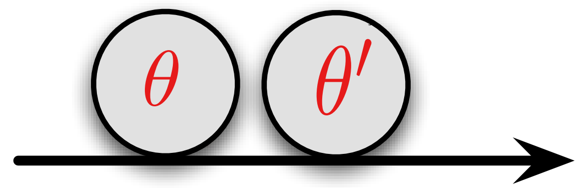
$$1/\sqrt{2}\left(|0\rangle + e^{i\theta}|1\rangle\right)$$

$$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$$



Universal Blind Quantum Computings

$$X = (\tilde{U}, \{\phi_{x,y}\})$$



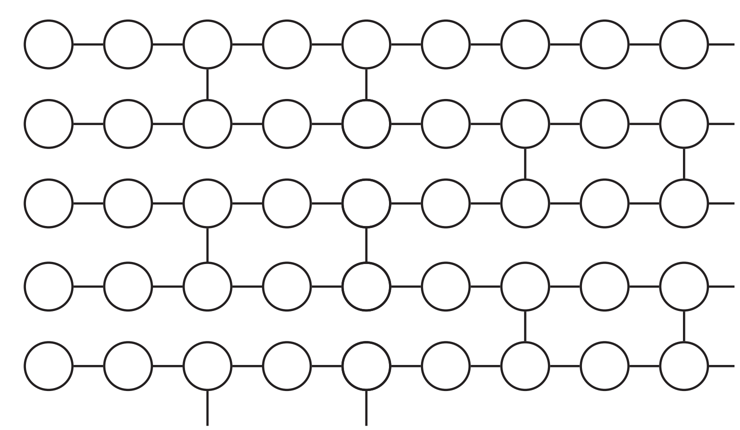
random single qubit generator

$$1/\sqrt{2}\left(|0\rangle + e^{i\theta}|1\rangle\right)$$

$$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$$

$$r_{x,y} \in_R \{0,1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$



Universal Blind Quantum Computings

$X = (\tilde{U}, \{\phi_{x,y}\})$

θ θ'

random single qubit generator

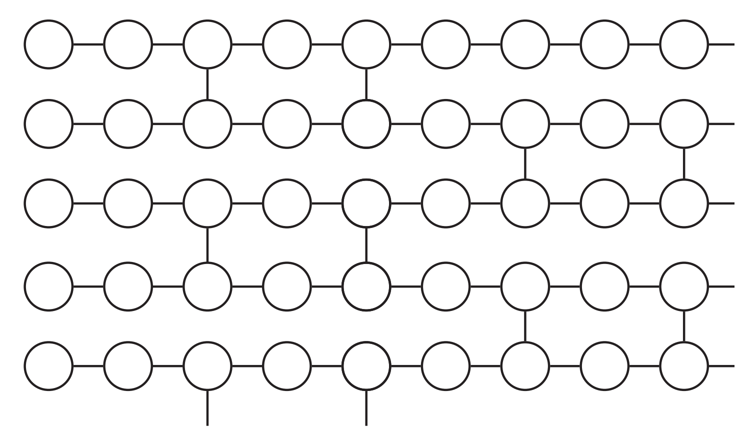
$$1/\sqrt{2}\left(|0\rangle + e^{i\theta}|1\rangle\right)$$

$$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$$

$\delta_{x,y}$

$$r_{x,y} \in_R \{0,1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$



Universal Blind Quantum Computings

$$X = (\tilde{U}, \{\phi_{x,y}\})$$

θ θ'

random single qubit generator

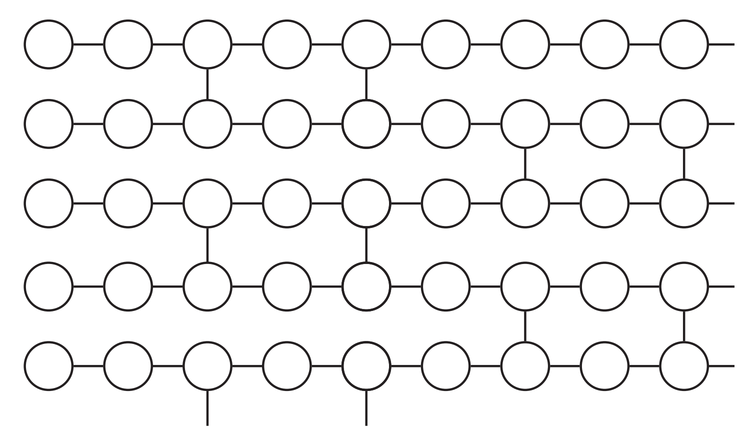
$$1/\sqrt{2}\left(|0\rangle + e^{i\theta}|1\rangle\right)$$

$$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$$

$\delta_{x,y}$

$$r_{x,y} \in_R \{0,1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$



$$\left\{\left|+_{\delta_{x,y}}\right\rangle, \left|-_{\delta_{x,y}}\right\rangle\right\}$$

Universal Blind Quantum Computings

$$X = (\tilde{U}, \{\phi_{x,y}\})$$

random single qubit generator

$$1/\sqrt{2}\left(|0\rangle + e^{i\theta}|1\rangle\right)$$

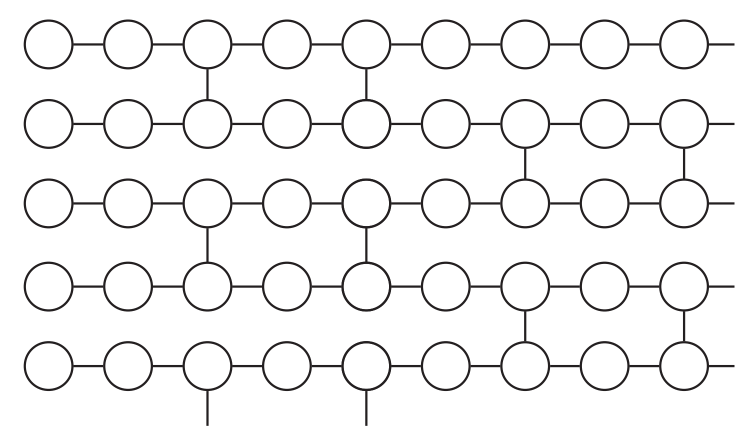
$$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$$

θ θ' →

$\delta_{x,y}$

$$r_{x,y} \in_R \{0,1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$



$$s_{x,y} \in \{0,1\}$$ ←

$$\{|+_{\delta_{x,y}}\rangle, |-_{\delta_{x,y}}\rangle\}$$

Universal Blind Quantum Computings

$X = (\tilde{U}, \{\phi_{x,y}\})$

random single qubit generator

$$1/\sqrt{2}\left(|0\rangle + e^{i\theta}|1\rangle\right)$$

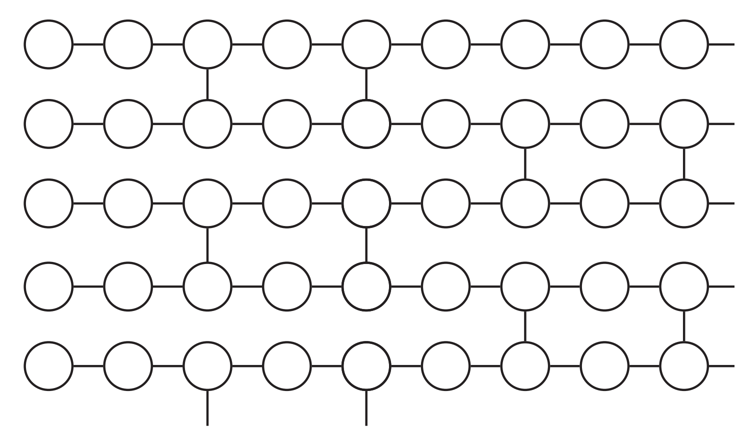
$$\theta = 0, \pi/4, 2\pi/4, \ldots, 7\pi/4$$

θ θ'

$\delta_{x,y}$

$r_{x,y} \in_R \{0,1\}$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$



$\{|+_{\delta_{x,y}}\rangle, |-_{\delta_{x,y}}\rangle\}$

$s_{x,y} \in \{0,1\}$

$$s_{x,y} := s_{x,y} + r_{x,y}$$

A lifting theorem

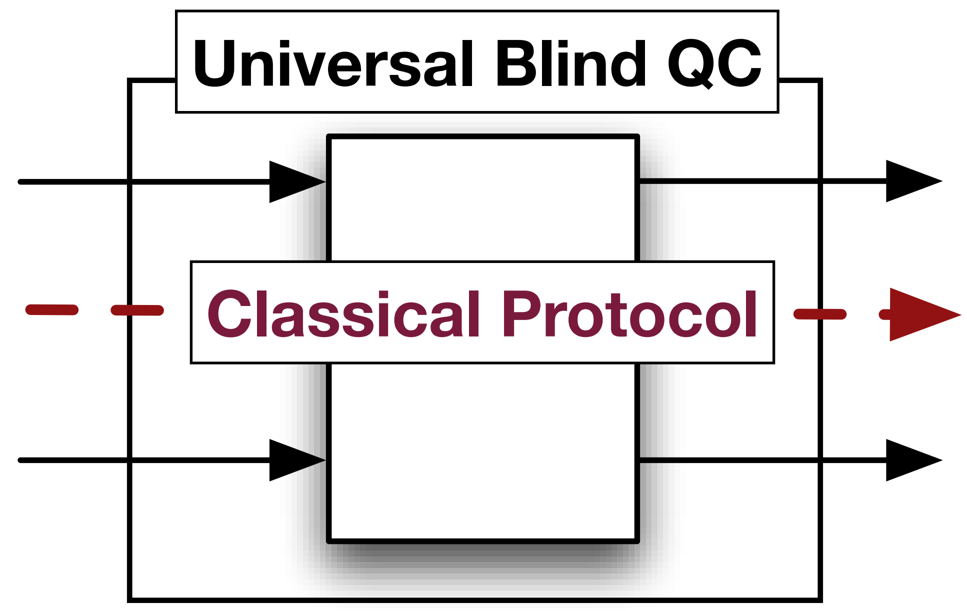
Any classical cryptographic protocol could be lifted to a corresponding quantum protocols via UBQC

A lifting theorem

Any classical cryptographic protocol could be lifted to a corresponding quantum protocols via UBQC

A lifting theorem

Any classical cryptographic protocol could be lifted to a corresponding quantum protocols via UBQC



Classical Crypto

Yao Garbled Circuit

Fully Homomorphic Encryption

One-time program

Secure Multi Party Computation

Secure Cloud Computing

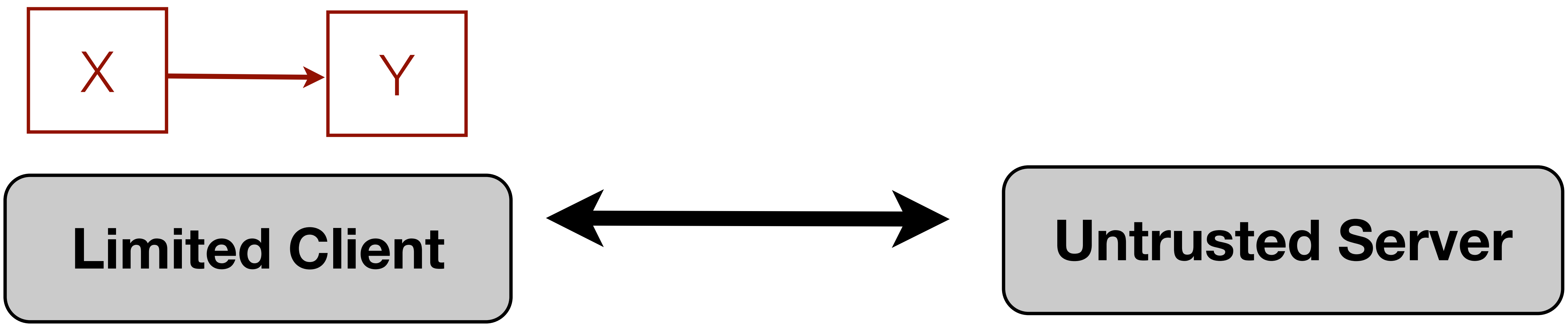
Rivest 78: Processing encrypted data without decrypting it first

Secure Cloud Computing

Rivest 78: Processing encrypted data without decrypting it first

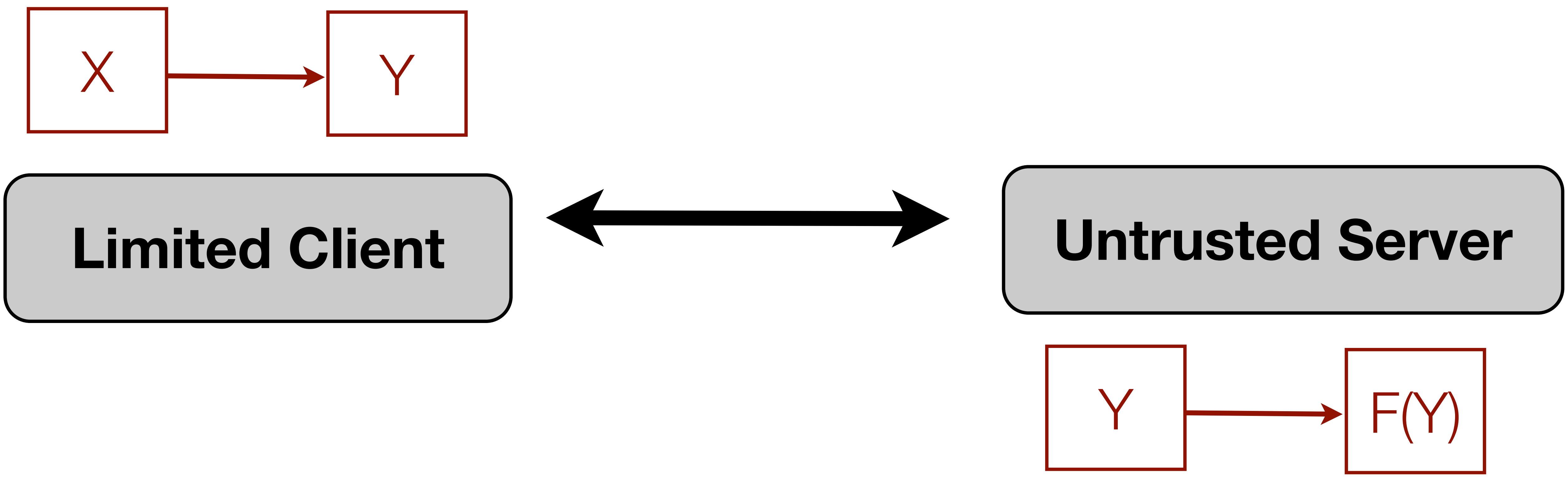
Secure Cloud Computing

Rivest 78: Processing encrypted data without decrypting it first



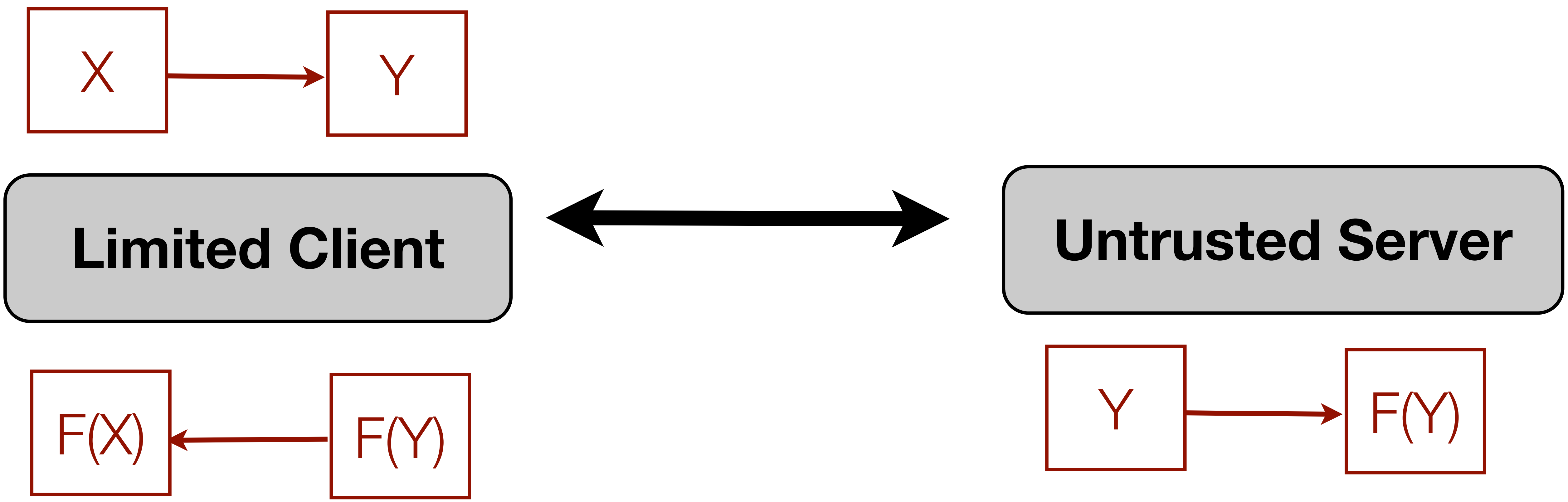
Secure Cloud Computing

Rivest 78: Processing encrypted data without decrypting it first



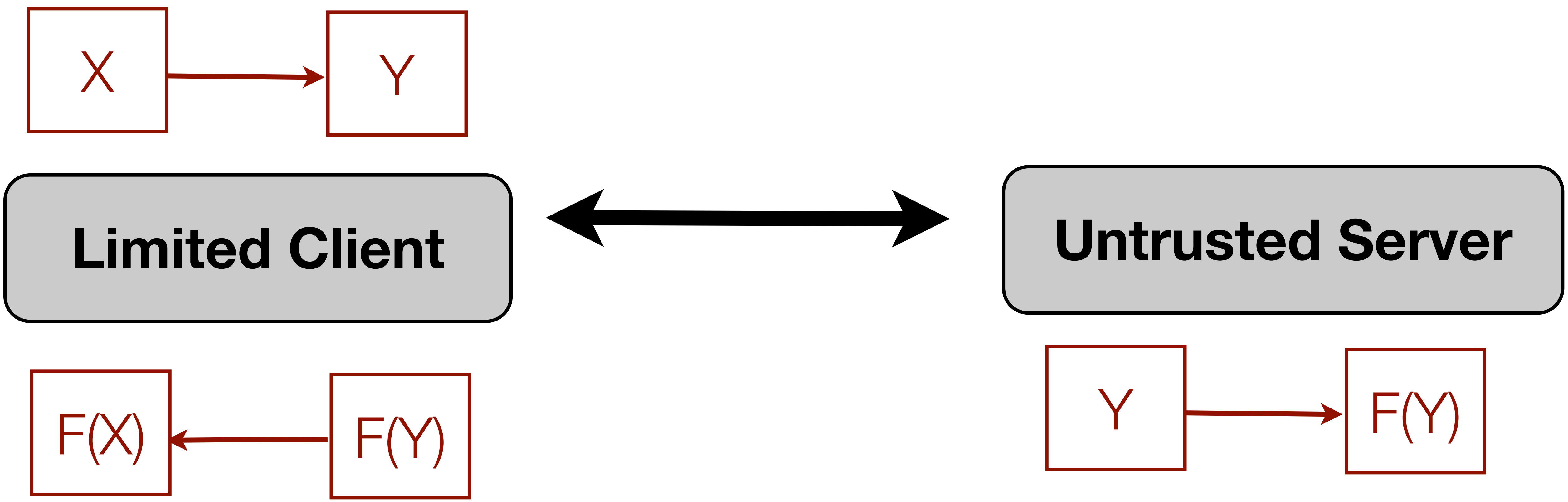
Secure Cloud Computing

Rivest 78: Processing encrypted data without decrypting it first



Secure Cloud Computing

Rivest 78: Processing encrypted data without decrypting it first



Gentry 09: A Lattice-based cryptosystem that is fully homomorphic but **inefficient** and only **computationally secure**

UBQC as FHE

$$X = (\tilde{U}, \{\phi_{x,y}\})$$

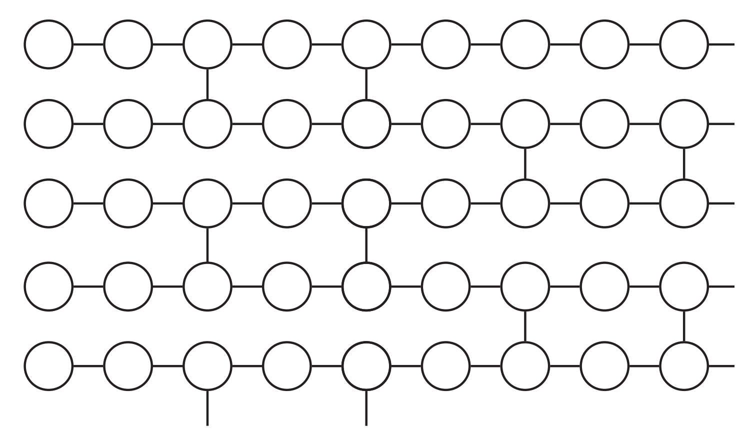
$$|\psi_{x,y}\rangle \in_R \{|+_\theta\rangle\}$$

$$\phi'_{x,y} = (-1)^{s^X_{x,y}} \phi_{x,y} + s^Z_{x,y}\pi$$

$$\delta_{x,y}$$

$$r_{x,y} \in_R \{0,1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$



$$\{|+_{\delta_{x,y}}\rangle, |-_{\delta_{x,y}}\rangle\}$$

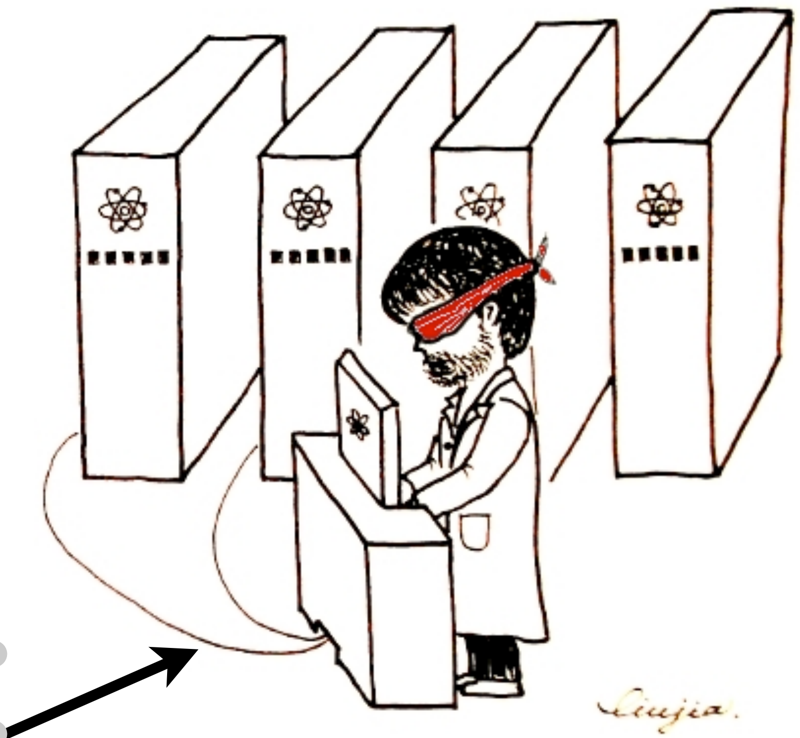
$$s_{x,y} \in \{0,1\}$$

$$s_{x,y} := s_{x,y} + r_{x,y}$$

UBQC as FHE

$X = (\tilde{U}, \{\phi_{x,y}\})$

$|\psi_{x,y}\rangle \in_R \{|+_\theta\rangle\}$

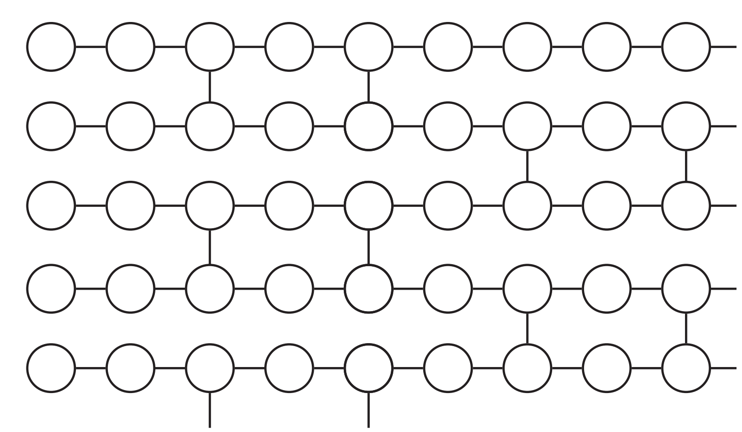


$$\phi'_{x,y} = (-1)^{s^X_{x,y}} \phi_{x,y} + s^Z_{x,y}\pi$$

$$r_{x,y} \in_R \{0, 1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$

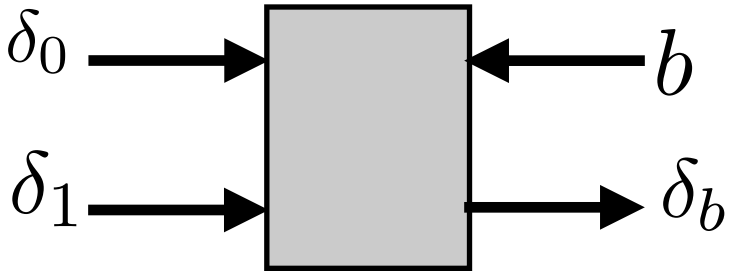
Interaction



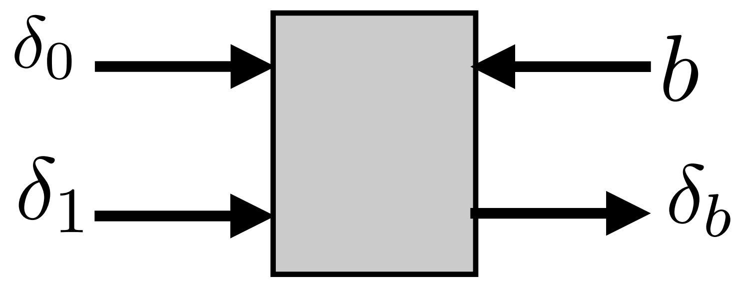
$$\{|+_{\delta_{x,y}}\rangle, |-_{\delta_{x,y}}\rangle\}$$

$$s_{x,y} := s_{x,y} + r_{x,y}$$

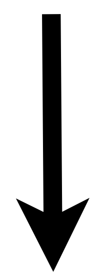
One-time Memory



One-time Memory



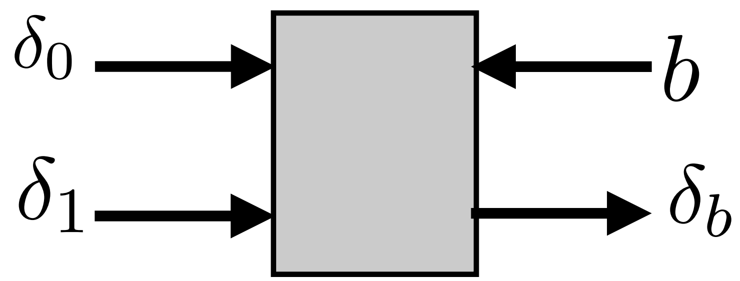
Founding Cryptography on Tamper-Proof Hardware Tokens



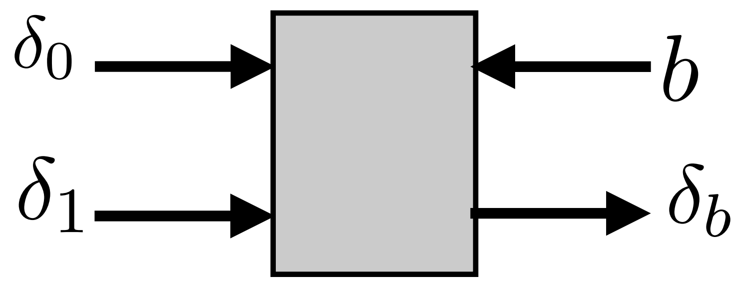
Unconditional non-interactive secure computation

Goldwasser, Kalai and Rothblum, Crypto, 2008
Goyal, Ishai, Sahai, Venkatesan, Wadai, TCC, 2010

Non-interactive UBQC using OTM



Non-interactive UBQC using OTM



UBQC on a constant degree graph

↓

Linear many OTM (in the size of input circuit)
is required to make UBQC non-interactive

Kashefi, In preparation 2014

Somewhat QFHE

$$X = (\tilde{U}, \{\phi_{x,y}\})$$

Secret input

$$|\psi_{x,y}\rangle \in_R \{|+_\theta\rangle\}$$

$$r_{x,y} \in_R \{0,1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$

Encryption

Somewhat QFHE

$$X = (\tilde{U}, \{\phi_{x,y}\})$$

Secret input

$$|\psi_{x,y}\rangle \in_R \{|+_\theta\rangle\}$$

$$r_{x,y} \in_R \{0,1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$

Encryption

Qubits and OTM

Somewhat QFHE

$X = (\tilde{U}, \{\phi_{x,y}\})$

Secret input

$$|\psi_{x,y}\rangle \in_R \{|+_\theta\rangle\}$$

$$r_{x,y} \in_R \{0,1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$

Encryption

Qubits and OTM

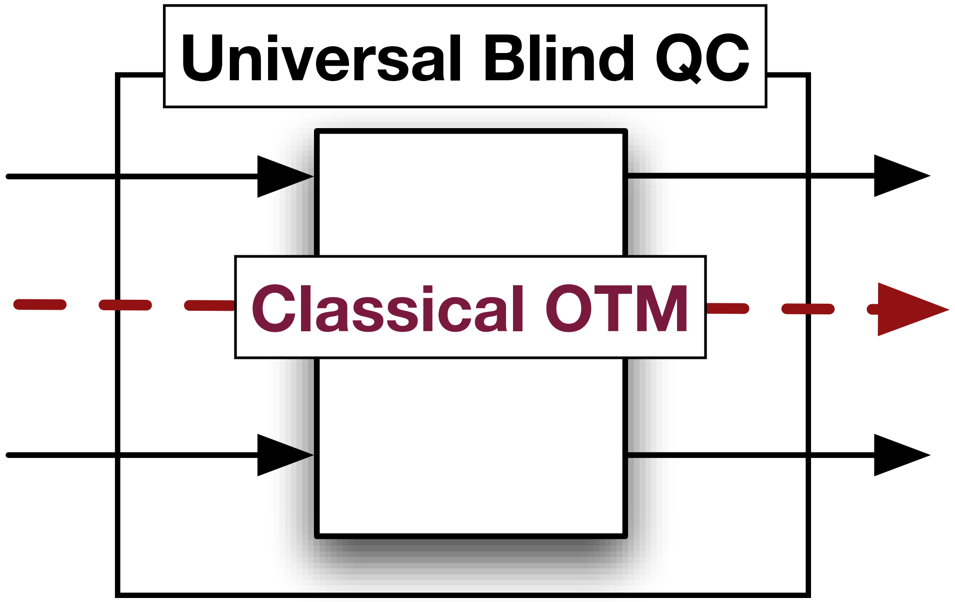
Decryption

$$s_{x,y} := s_{x,y} + r_{x,y}$$

Quantum FHE

Quantum FHE

Quantum FHE



Secure Multi-Party Computing

Yao 86. A set of participants with private inputs x_i want to compute a function f(x_1, ..., x_i, ... x_n) while keeping their local data secret

Secure Multi-Party Computing

Yao 86. A set of participants with private inputs x_i want to compute a function $f(x_1, ..., x_i, ... x_n)$ while keeping their local data secret

Security

Active internal or external adversaries cannot find more than output of function

Goldreich, Micali, and Wigderson, STOC, 87

UBQC as Quantum SMPC

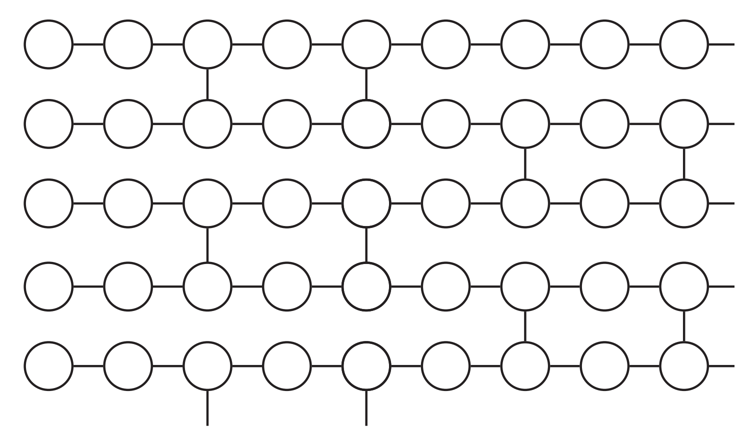
$X = (\tilde{U}, \{\phi_{x,y}\})$

$|\psi_{x,y}\rangle \in_R \{|+_\theta\rangle\}$

Trusted Alice

$\phi'_{x,y}$ ... $_{,y}\pi$

$\delta_{x,y}$



$r_{x,y} \in_R \{0,1\}$

$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$

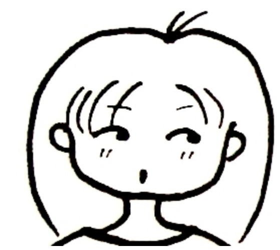
$\{|+_{\delta_{x,y}}\rangle, |-_{\delta_{x,y}}\rangle\}$

$s_{x,y} \in \{0,1\}$

$s_{x,y} := s_{x,y} + r_{x,y}$

UBQC as Quantum SMPC

UBQC as Quantum SMPC

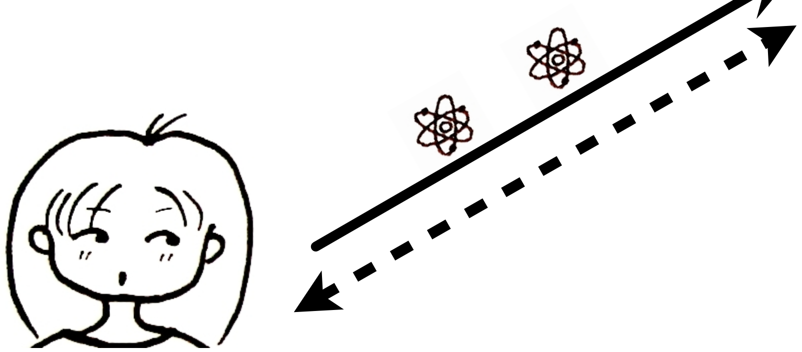


Fulop, Kapourniotis, Kashefi, In preparation 2014

UBQC as Quantum SMPC

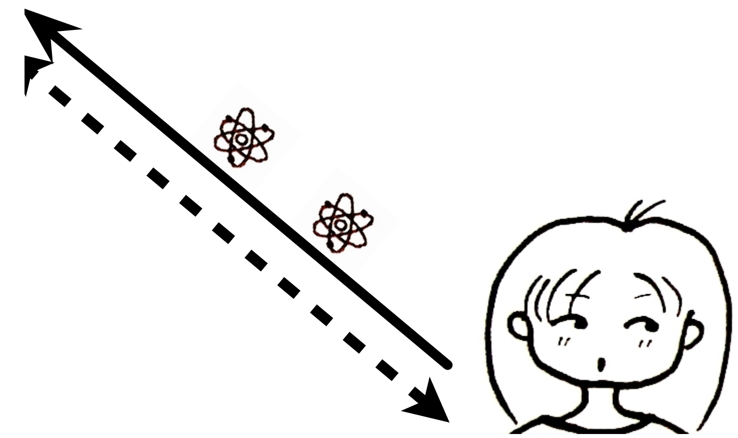
$|\underset{1}{\psi}_{x,y}\rangle \in_R \{|+_\theta\rangle\}$

Private Secret of Client



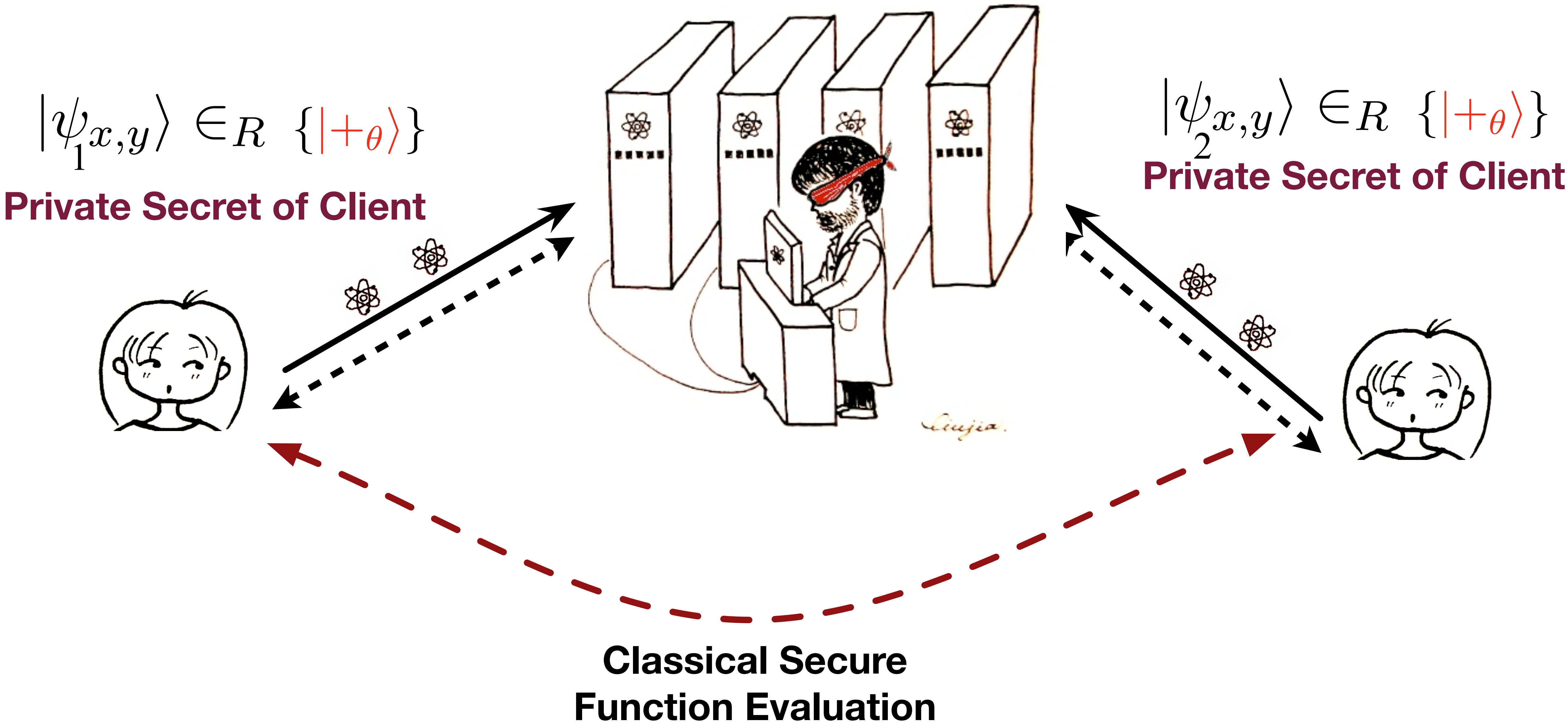
$|\underset{2}{\psi}_{x,y}\rangle \in_R \{|+_\theta\rangle\}$

Private Secret of Client



Fulop, Kapourniotis, Kashefi, In preparation 2014

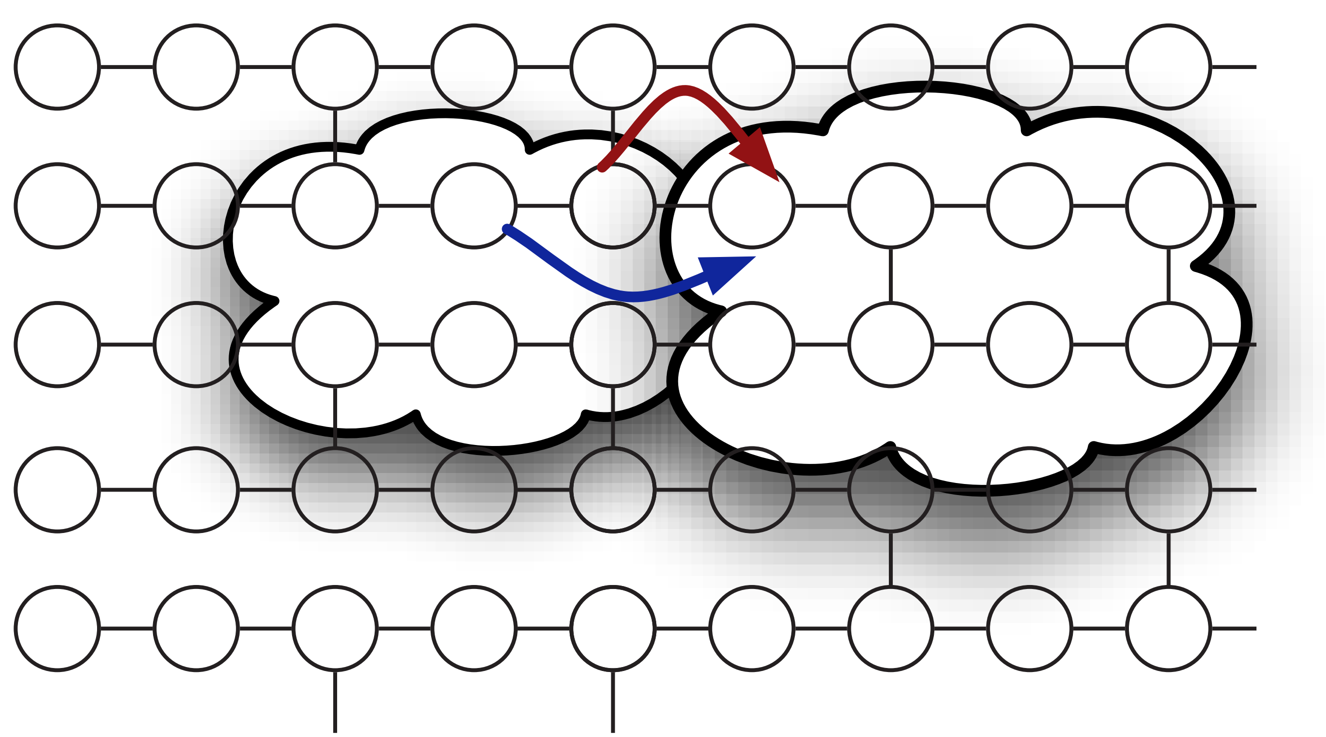
UBQC as Quantum SMPC



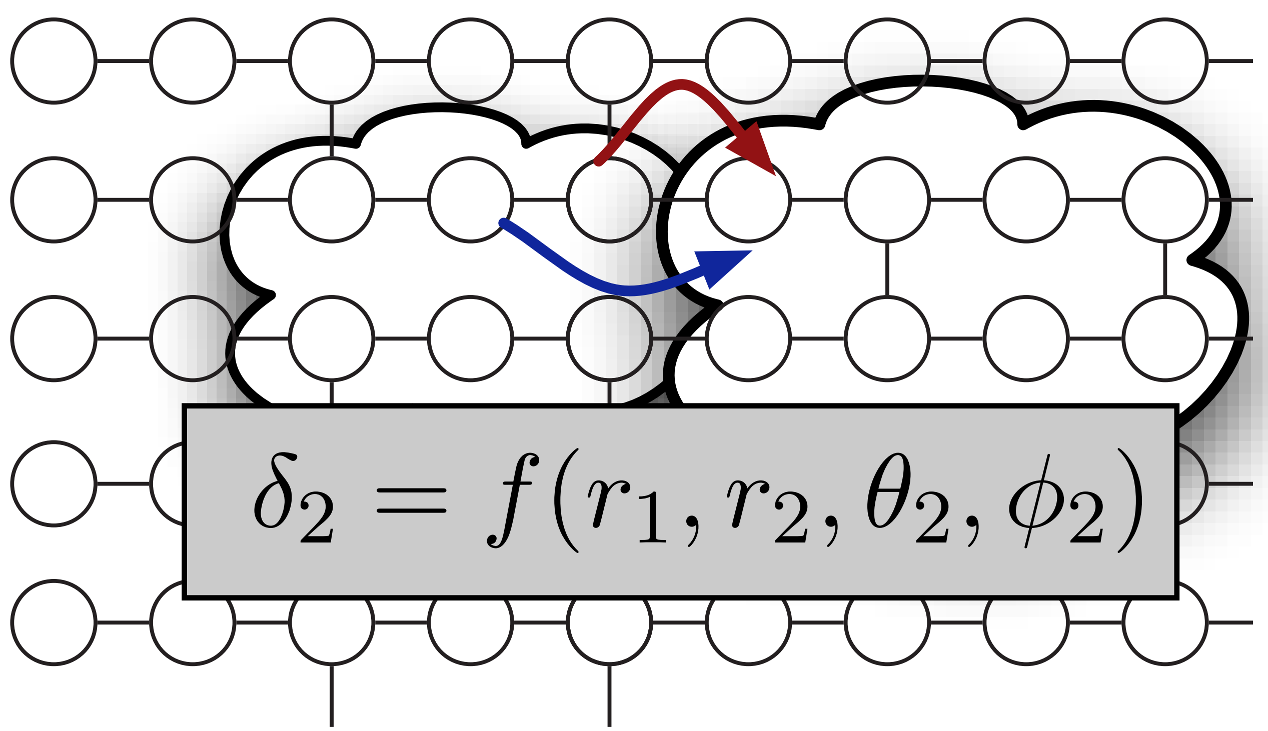
Fulop, Kapourniotis, Kashefi, In preparation 2014

UBQC as Quantum SMPC

UBQC as Quantum SMPC



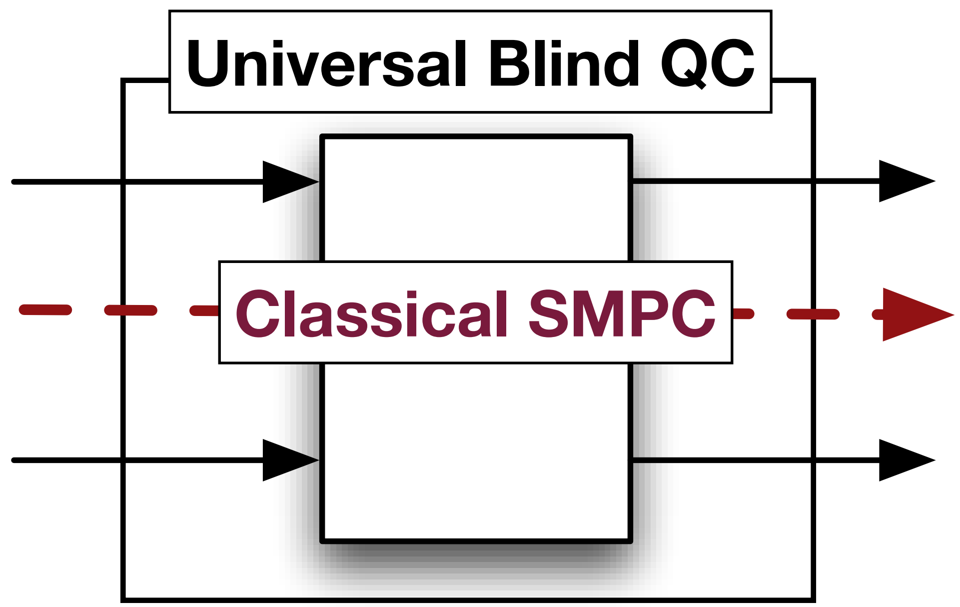
UBQC as Quantum SMPC



Quantum SMPC

Quantum SMPC

Quantum SMPC



Classical lifting

A **hybrid** network of classical protocols with quantum gadgets

boosting efficiency and security

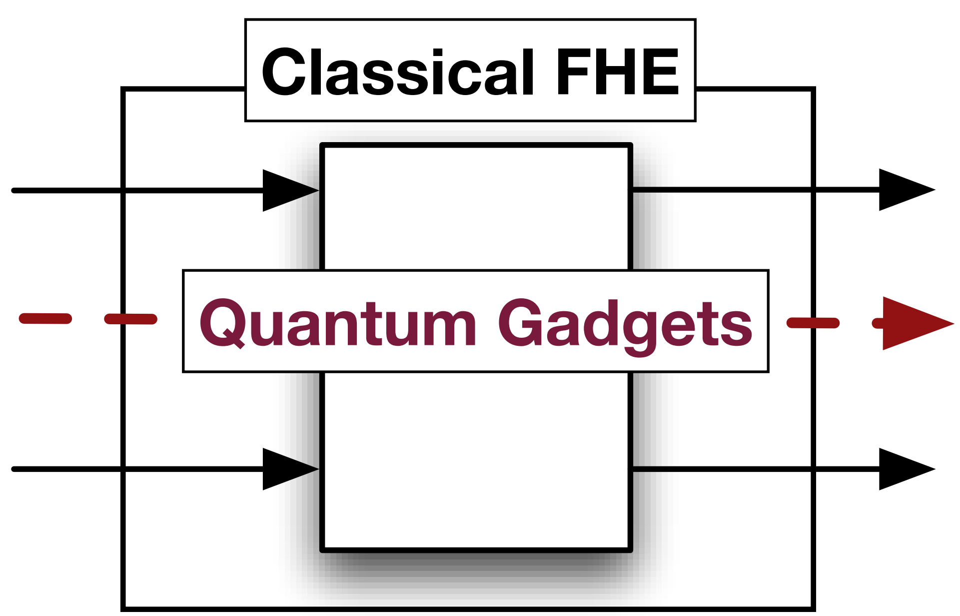
of every task achievable against classical attackers against quantum attackers

Classical lifting

A **hybrid** network of classical protocols with quantum gadgets

boosting efficiency and security

of every task achievable against classical attackers against quantum attackers



UBQC as a gadget

$X = (\tilde{U}, \{\phi_{x,y}\})$

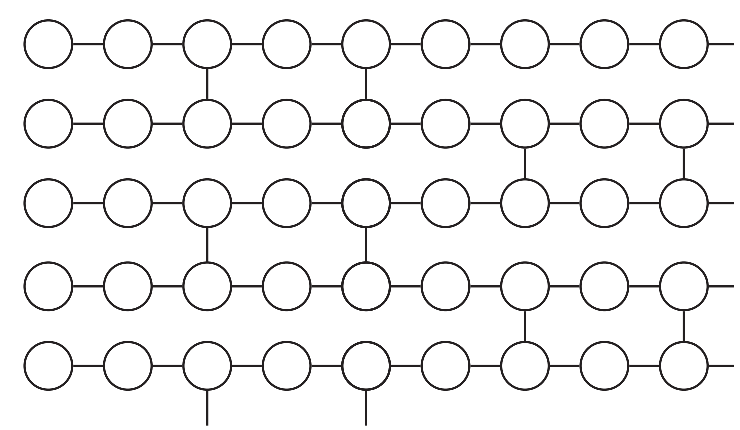
$|\psi_{x,y}\rangle \in_R \{|+_\theta\rangle\}$

$$\phi'_{x,y} = (-1)^{s^X_{x,y}} \phi_{x,y} + s^Z_{x,y}\pi$$

$$r_{x,y} \in_R \{0,1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$

$\delta_{x,y}$



$\{|+_{\delta_{x,y}}\rangle, |-_{\delta_{x,y}}\rangle\}$

$s_{x,y} \in \{0,1\}$

$s_{x,y} := s_{x,y} + r_{x,y}$

UBQC as a gadget

$$X = (\tilde{U}, \{\phi_{x,y}\})$$

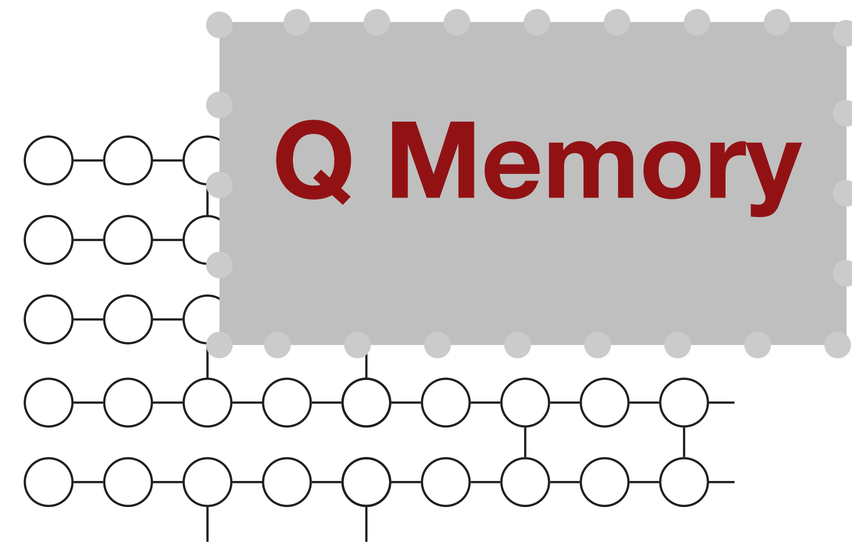
$$|\psi_{x,y}\rangle \in_R \{|+_\theta\rangle\}$$

$$\phi'_{x,y} = (-1)^{s^X_{x,y}}\phi_{x,y} + s^Z_{x,y}\pi$$

$$r_{x,y} \in_R \{0,1\}$$

$$\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$$

$$\delta_{x,y}$$



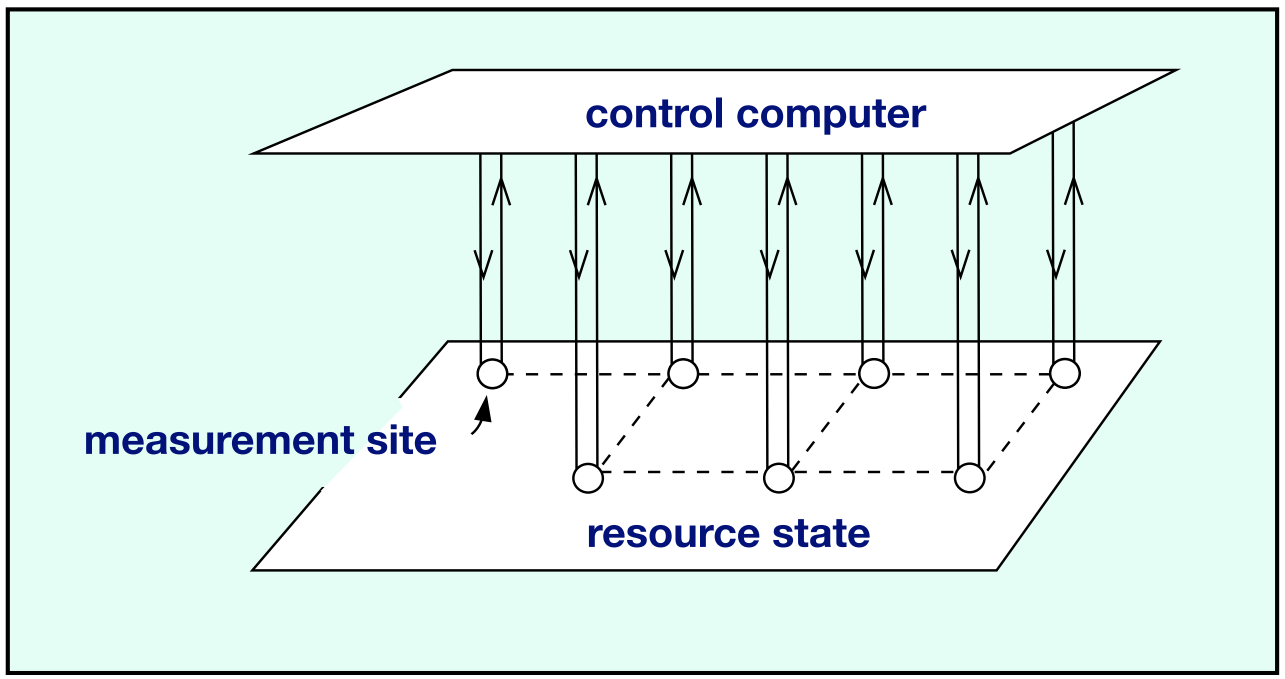
$$\{|+_{\delta_{x,y}}\rangle, |-_{\delta_{x,y}}\rangle\}$$

$$s_{x,y} \in \{0,1\}$$

$$s_{x,y} := s_{x,y} + r_{x,y}$$

Q Memory

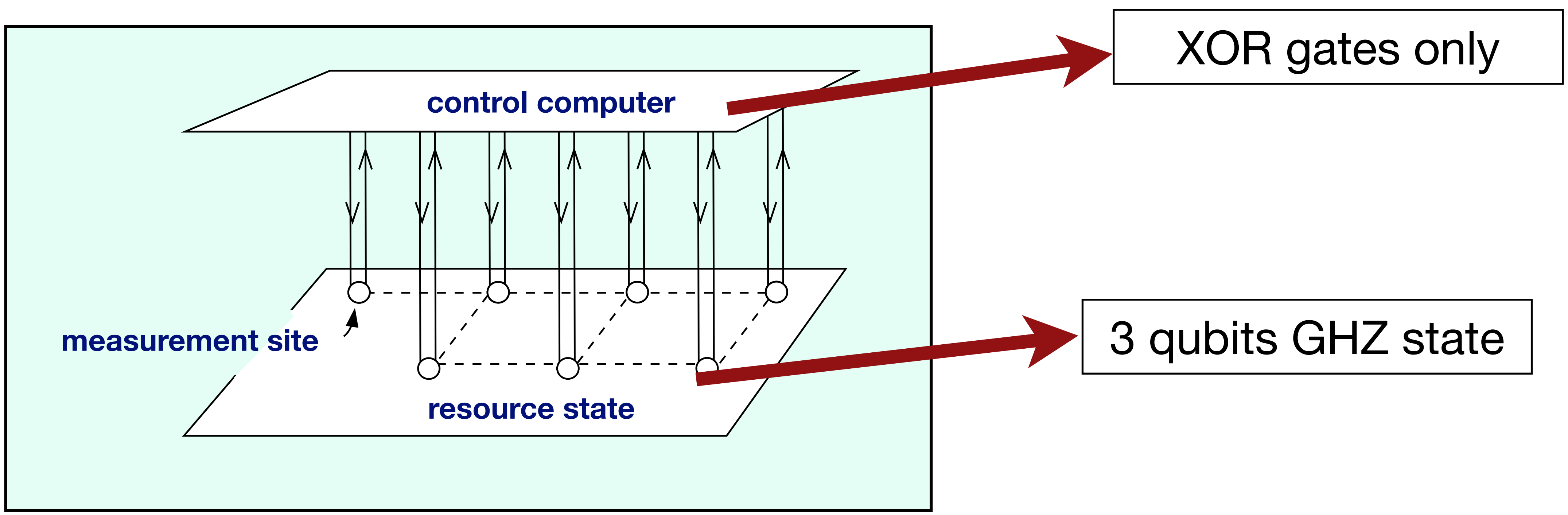
UBQC for secure evaluation of classical function



Anders and Browne, PRL, 2009

Q Memory

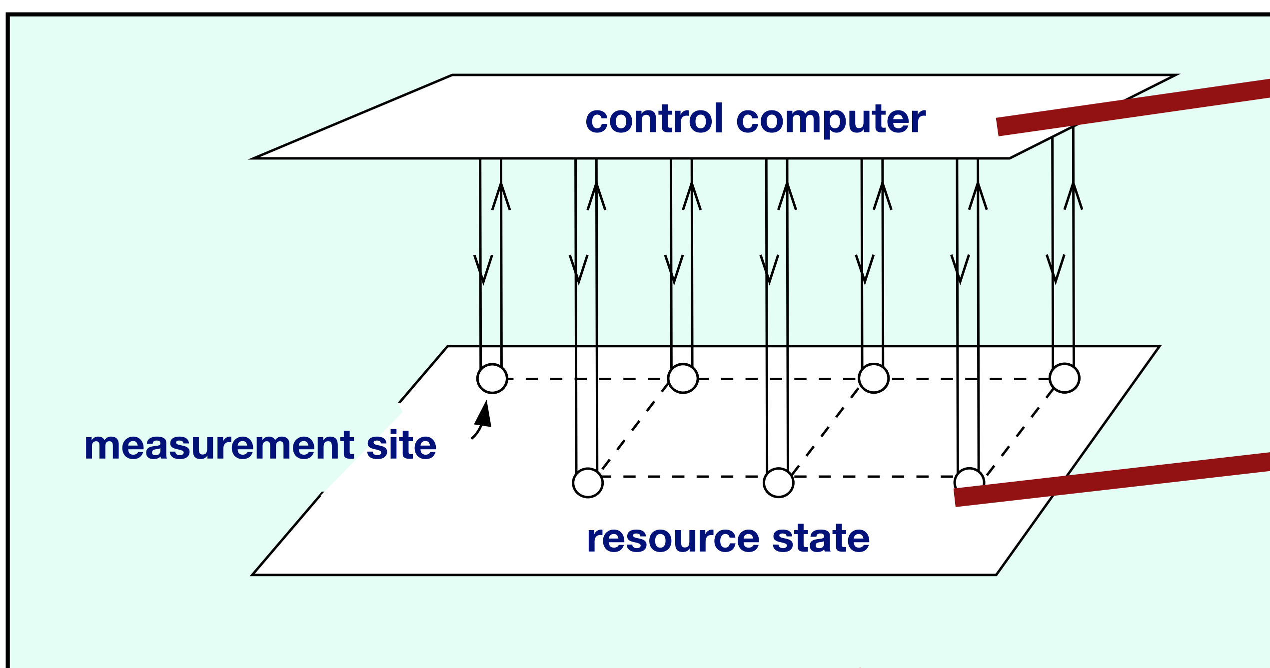
UBQC for secure evaluation of classical function



Anders and Browne, PRL, 2009

Q Memory

UBQC for secure evaluation of classical function



XOR gates only

3 qubits GHZ state

Universal Classical Computing

Anders and Browne, PRL, 2009

Restricted XOR Client

No classical protocol, with XOR client can securely delegate deterministic computation of NAND to a server.

Dunjko, Kapourniotis, Kashefi, arXiv:1405.4558, 2014

Restricted XOR Client

No classical protocol, with XOR client can securely delegate deterministic computation of NAND to a server.

Client's encoding: $C_1(a, b, \overrightarrow{x})$

Dunjko, Kapourniotis, Kashefi, arXiv:1405.4558, 2014

Restricted XOR Client

No classical protocol, with XOR client can securely delegate deterministic computation of NAND to a server.

Client's encoding: $C_1(a, b, \overrightarrow{x})$ (input: a, b; random: $\overrightarrow{x}$) **XOR computable function independent of the input**

Dunjko, Kapourniotis, Kashefi, arXiv:1405.4558, 2014

Restricted XOR Client

No classical protocol, with XOR client can securely delegate deterministic computation of NAND to a server.

Client's encoding: $C_1(a, b, \overrightarrow{x})$ — **input**: a, b; **random**: $\overrightarrow{x}$ — **XOR computable function independent of the input**

Server's computation: $S(C_1(a, b, \overrightarrow{x}))$

Dunjko, Kapourniotis, Kashefi, arXiv:1405.4558, 2014

Restricted XOR Client

No classical protocol, with XOR client can securely delegate deterministic computation of NAND to a server.

Client's encoding: $C_1(a, b, \overrightarrow{x})$ **XOR computable function independent of the input**

(**input**: a, b; **random**: $\overrightarrow{x}$)

Server's computation: $S(C_1(a, b, \overrightarrow{x}))$

Client's decoding: $C_2(a, b, \overrightarrow{x}, S(C_1(a, b, \overrightarrow{x}))) = NAND(a, b)$ **XOR computable function**

Dunjko, Kapourniotis, Kashefi, arXiv:1405.4558, 2014

Restricted XOR Client

No classical protocol, with XOR client can securely delegate deterministic computation of NAND to a server.

Client's encoding: $C_1(a, b, \overrightarrow{x})$ **XOR computable function independent of the input**

(input: a, b; random: $\overrightarrow{x}$)

Server's computation: $S(C_1(a, b, \overrightarrow{x}))$

Client's decoding: $C_2(a, b, \overrightarrow{x}, S(C_1(a, b, \overrightarrow{x}))) = NAND(a, b)$ **XOR computable function**

Constant

Dunjko, Kapourniotis, Kashefi, arXiv:1405.4558, 2014

Quantum Communication

$$Z^r S^a S^b \left(S^\dagger\right)^{a \oplus b} |+\rangle \quad = \quad Z^r Z^{a \wedge b} |+\rangle$$

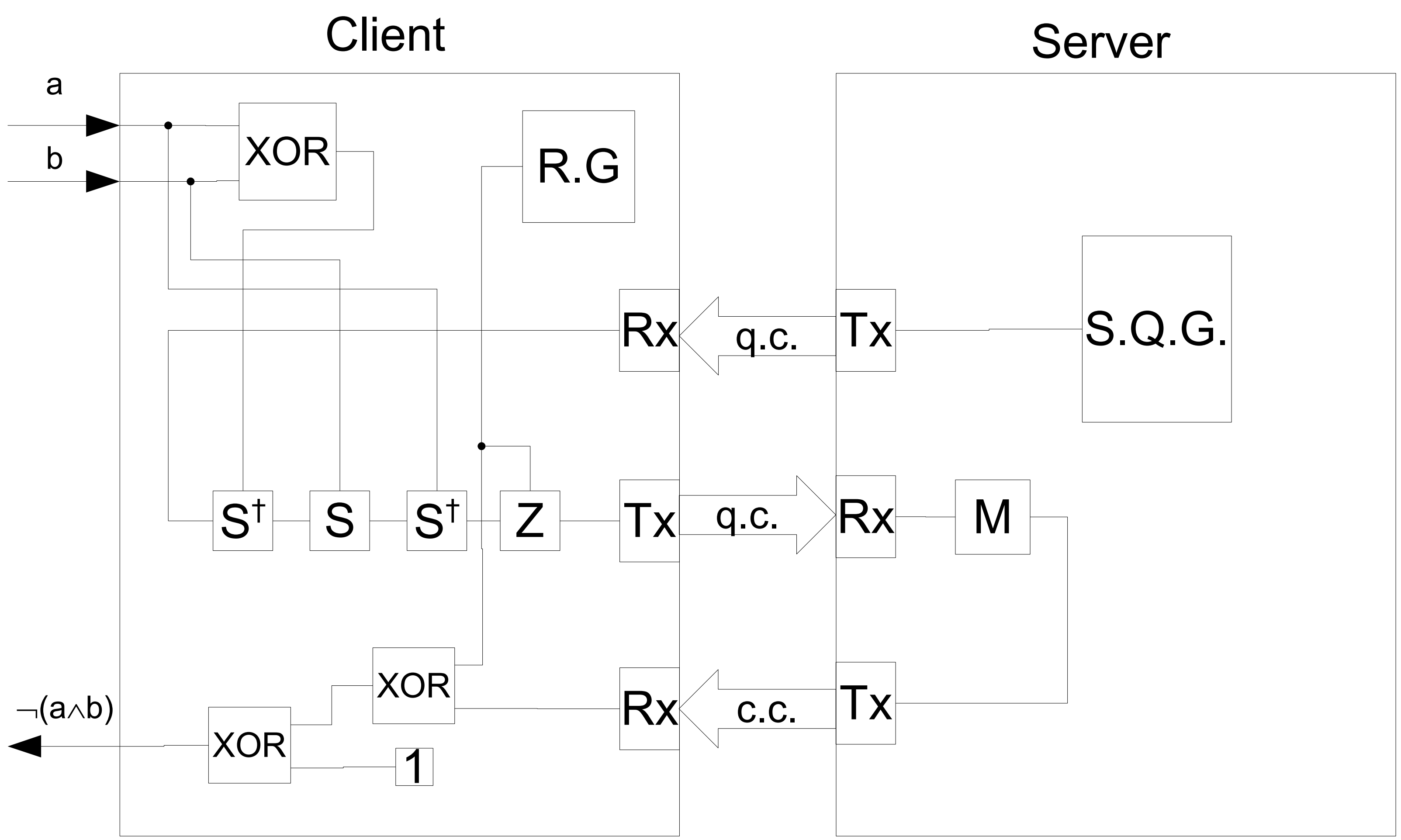
Quantum Communication

$$Z^r S^a S^b \left(S^\dagger\right)^{a \oplus b} |+\rangle \quad = \quad Z^r Z^{a \wedge b} |+\rangle$$

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{ib\pi/2} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{ia\pi/2} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{-i(a \oplus b)\pi/2} \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i(a \wedge b)\pi} \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Quantum Communication

Quantum Communication



How to become millionaire

How to become millionaire

A hybrid network of **LWE-based FHE** with **UBQC gadgets**

boosting efficiency and security

of classical delegated computing against quantum attackers

LWE problem

LWE problem

$$\left\{\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i\right\}_{i=1}^{\text{poly}(n)} \quad \overset{c}{\approx} \quad \left\{\mathbf{a}_i, u_i\right\}_{i=1}^{\text{poly}(n)}$$

LWE problem

$$\{\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i\}_{i=1}^{\text{poly}(n)} \stackrel{c}{\approx} \{\mathbf{a}_i, u_i\}_{i=1}^{\text{poly}(n)}$$

secret $\in \mathbb{Z}_q^n$

noise $\in \mathbb{Z}_q$

$\in \mathbb{Z}_q$ uniformly random

LWE problem

$$\{\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i\}_{i=1}^{\mathrm{poly}(n)} \stackrel{c}{\approx} \{\mathbf{a}_i, u_i\}_{i=1}^{\mathrm{poly}(n)}$$

secret $\in \mathbb{Z}_q^n$

noise $\in \mathbb{Z}_q$

$\in \mathbb{Z}_q$ uniformly random

Encryption Scheme based on (LWE)

$$c = (\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + 2e + m) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

Rewinding and Higher Order Function

We say: first suppose you have a cheating verifier V. When V talks to an honest prover, it outputs (a distribution of) some transcript t. We have to show how t
sample the same (or very close) distribution of t, without talking to any honest prover. It's not likely that we can analyze the code of V to "figure out what it's
doing." Instead, we have to treat V as a kind of black-box. Recall that V is designed to operate in an interactive fashion, so we have to feed protocol messag
into V, pretending to be the honest prover. We might feed into V a simulated "message 1" from the prover, and then later a simulated "message 2". Then, afte
seeing how V responded, we might go back to a previous internal state of V and feed in a different simulated "message 2" -- that's rewinding. We can rewinc
invoke V many different times, as long as we are careful to spend only polynomial time overall (assuming V itself is polynomial-time).

ME SYAING: Rewinding is some kind of if then else procedure to be used for creating ultimately the desired simulated transcript. Isn't the same problem in tl
quantum programming issue of defining if than else compactly leads to the same issue regarding rewinding ?