Quantum Homomorphic Encryption for Polynomial-Size Circuits

Christian Schaffner

Joint work with
Yfke Dulek and Florian Speelman

http://arxiv.org/abs/1603.09717

Institute for Logic, Language and Computation (ILLC)
University of Amsterdam

QuSoft
Research Center for Quantum Software

CWI
Centrum Wiskunde & Informatica

Symposium on the Work of Ivan Damgård
Friday, 1 April 2016

NWO
Nederlandse Organisatie voor Wetenschappelijk Onderzoek

Roadmap

- (Classical) Homomorphic Encryption
- Quantum Homomorphic Encryption
- Computation by teleportation
- Our scheme

Homomorphic encryption

Classical case

- Encrypt data so that another party can perform calculations on the encrypted data

- Many applications

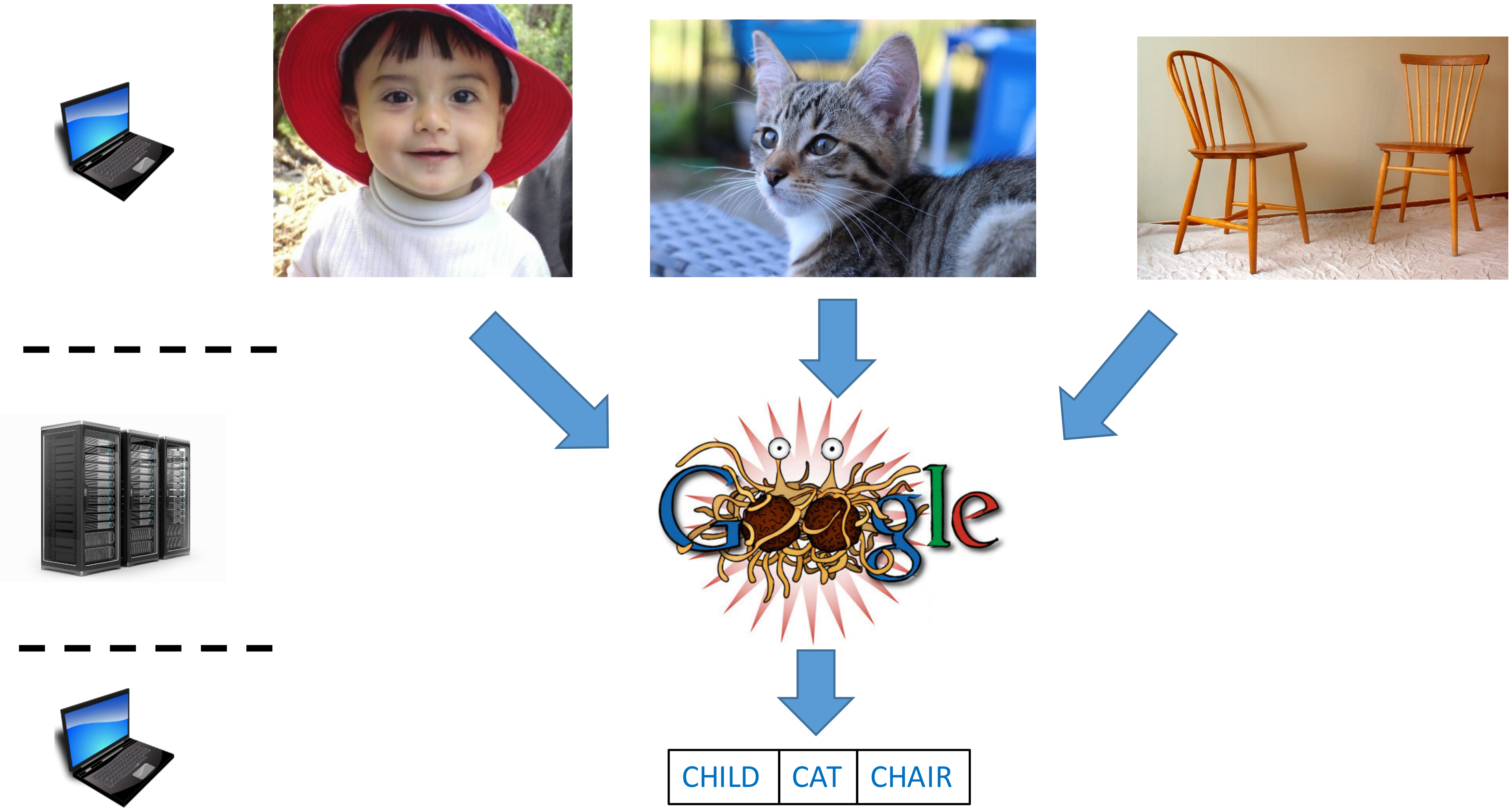
CHILD

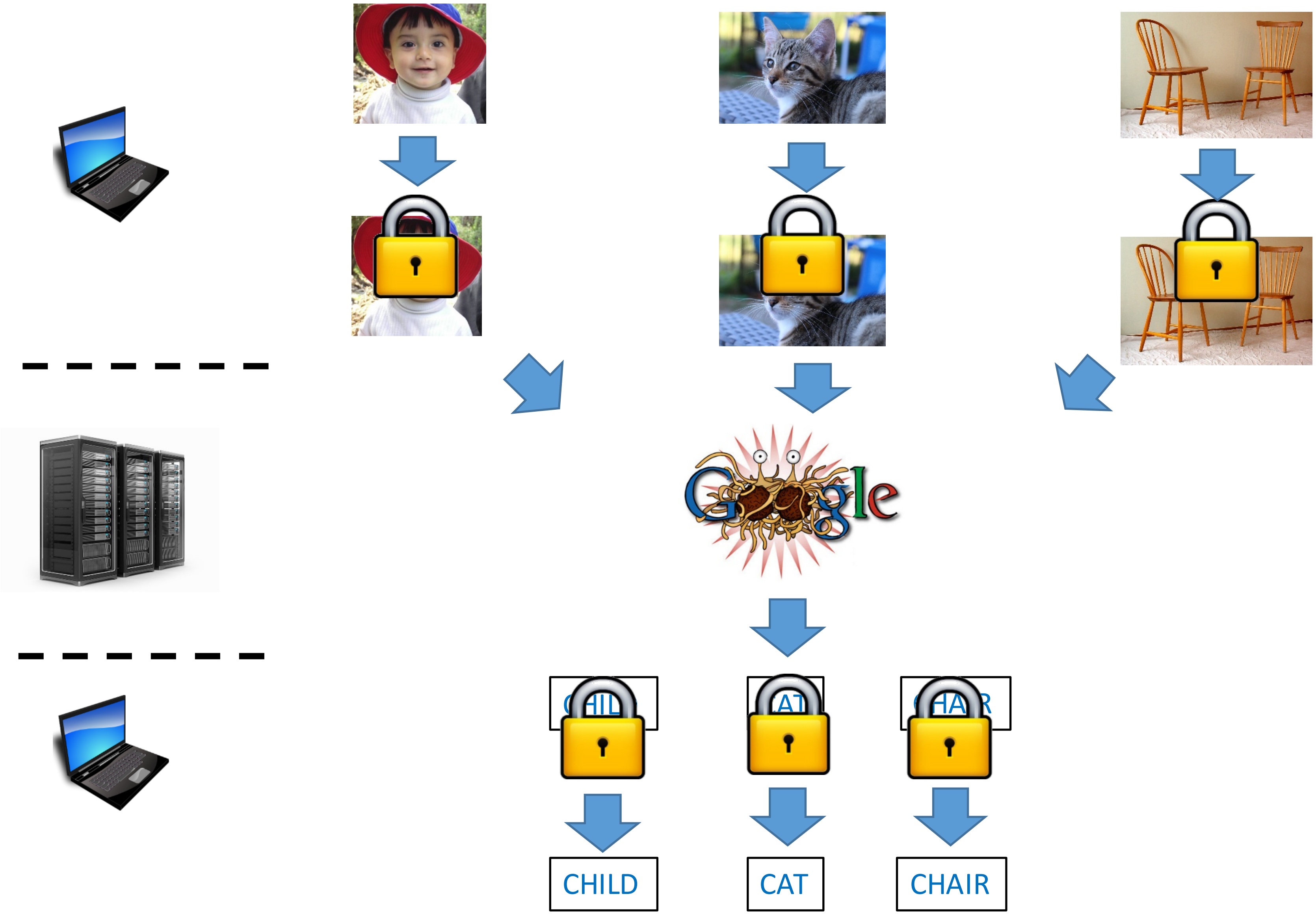
CAT

CHAIR

Tagging

Google DeepMind


CHILD
CAT
CHAIR


CHILD
CAT
CHAIR

RSA

Multiplicative homomorphic

- Public key: exponent e and modulus N
- Encryption of a message : $\mathrm{Enc}(x) = x^e \bmod N$

Given encryptions of messages x and y
possible to compute the encryption of the product:

$$(x^e \bmod N)(y^e \bmod N) = (xy)^e \bmod N$$

$$\mathrm{Enc}(x)\mathrm{Enc}(y) = \mathrm{Enc}(xy)$$

Fully Homomorphic Encryption

- Encrypt data so that another party can perform calculations on the encrypted data
- RSA (and ElGamal) are homomorphic with respect to multiplication
- Other schemes (e.g. Goldwasser-Micali) are additively homomorphic

$$\mathrm{Enc}(x) \cdot \mathrm{Enc}(y) = \mathrm{Enc}(x \oplus y)$$

- Universal computation needs **both**

  ADD: $\mathrm{ADD}\big(\mathrm{Enc}(x), \mathrm{Enc}(y)\big) = \mathrm{Enc}(x + y)$

  MULT: $\mathrm{MULT}\big(\mathrm{Enc}(x), \mathrm{Enc}(y)\big) = \mathrm{Enc}(xy)$

  while staying compact (complexity of Dec does not depend on evaluation circuit)
- First breakthrough proposal by Gentry 2009, currently multiple candidates

still slow: seconds per bit operation, but some of you know better than I do…

Roadmap

✓ (Classical) Homomorphic Encryption

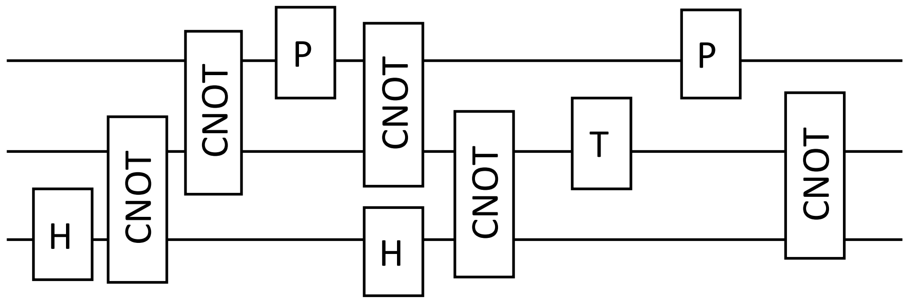
- $\boxed{\text{Quantum Homomorphic Encryption}}$
- New ingredients – computation by teleportation
- Our scheme

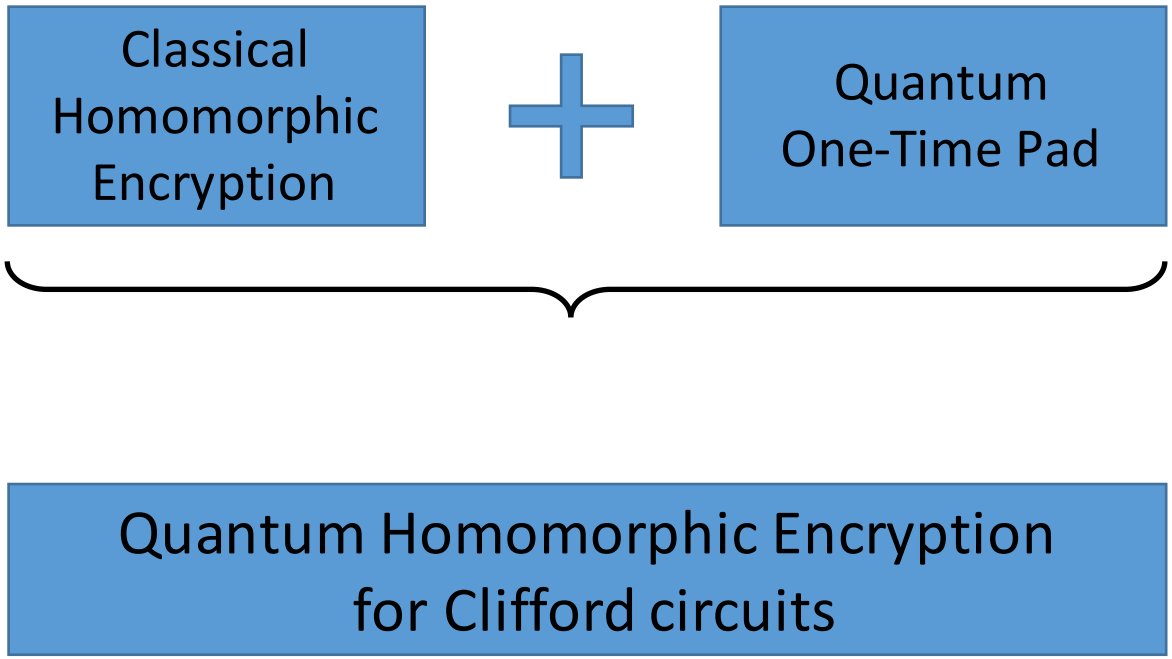
Quantum Homomorphic Encryption

- Encrypt *quantum state* instead of classical data

$$\rho \to \mathrm{QEnc}(\rho)$$

- Execute *quantum circuit* on encrypted data




Classical Homomorphic Encryption
Quantum One-Time Pad
Quantum Homomorphic Encryption for Clifford circuits

One-time pad

Plaintext n-bit string $x \in \{0,1\}^n$

Key n-bit string $k \leftarrow \{0,1\}^n$ ← randomly chosen

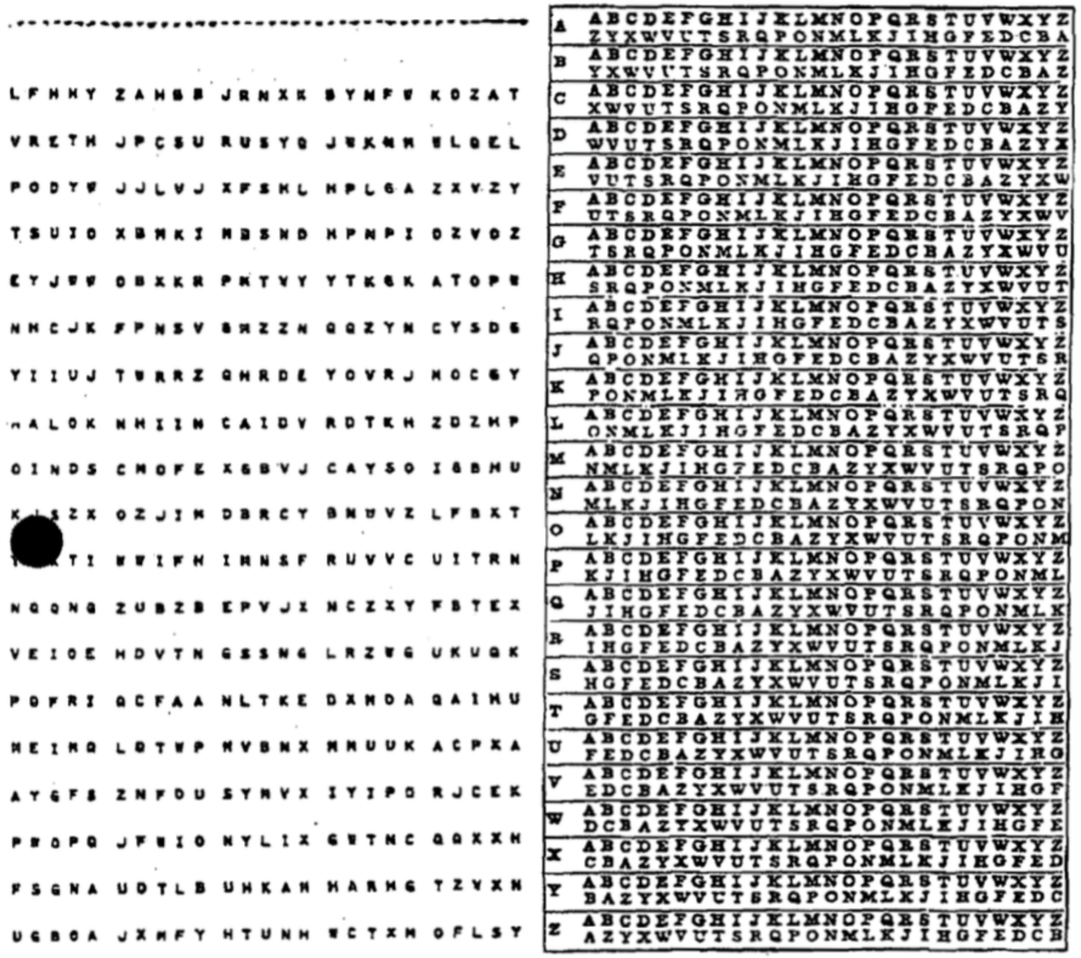
Ciphertext $c = x \oplus k$

$\boldsymbol{x}$	0	1	1	0	1
$\mathbf{k}$	1	1	0	0	1
$\boldsymbol{c}$	**1**	**0**	**1**	**0**	**0**

Properties:

Perfectly secure

Key same size as message – each key can be used only once



One-time pad table (U.S. National Security Agency)

Quantum One-time Pad

- Pauli operators $\mathrm{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \mathrm{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \mathrm{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$
- Self-inverse: $X^2 = \mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \;\; Y^2 = \mathbb{I}, \;\; Z^2 = \mathbb{I}$
- Anti-commute: $XZ = -ZX, \; XY = -YX, \; YZ = -ZY$

- Flip two random bits $a, b \leftarrow \{0,1\}$,
  encryption of a qubit ρ: $X^a Z^b \; \rho \; X^a Z^b$
- Perfect security: not knowing a, b, density matrix becomes *fully mixed*:
  $\frac{1}{4} \sum_{a,b} X^a Z^b \rho Z^a X^b = \mathbb{I}/2$

[AMTW00] A. Ambainis, M. Mosca, A. Tapp, and R. De Wolf. Private quantum channels. FOCS'00

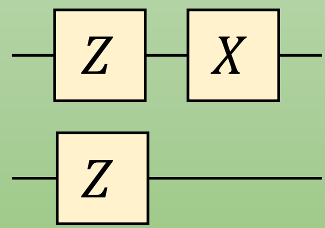
Pauli Group on n Qubits

Pauli operators
$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

- Pauli group $P_n := \{\phi X^{\vec{a}} Z^{\vec{b}} : \vec{a}, \vec{b} \in \{0,1\}^n, \phi \in \{\pm i, \pm 1\}\}$

2-qubit example:
$\vec{a} = 10, \vec{b} = 11$

$X^{\vec{a}} Z^{\vec{b}} =$



- Encryption of n qubits ρ: $X^{\vec{a}} Z^{\vec{b}} \rho X^{\vec{a}} Z^{\vec{b}}$ for random $\vec{a}, \vec{b} \in \{0,1\}^n$
- Perfect security: not knowing $\vec{a}, \vec{b}$, density matrix becomes *fully mixed*:

$$\frac{1}{4^n} \sum_{\vec{a}, \vec{b}} X^{\vec{a}} Z^{\vec{b}} \rho X^{\vec{a}} Z^{\vec{b}} = \mathbb{I}/2^n$$

[AMTW00] A. Ambainis, M. Mosca, A. Tapp, and R. De Wolf. Private quantum channels. FOCS'00
(based on Stacey Jeffery's slides)

The Clifford group

Pauli operators

$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

- Clifford group is the *normalizer of the Pauli group*:
  For all Cliffords C, for all Paulis $X^{\vec{a}}Z^{\vec{b}}$,
  there exist $\vec{c}, \vec{d} \in \{0,1\}^n$ such that $CX^{\vec{a}}Z^{\vec{b}} = X^{\vec{c}}Z^{\vec{d}}C$
- Generated by $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \; CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
- Examples: $HX = ZH, \quad PZ = ZP, \quad PX = XZP$
- Not a universal gate set
  - Classical simulation possible

Example interaction with quantum one-time pad:
$PX^aZ^b|\psi\rangle = X^aZ^{a\oplus b}P|\psi\rangle$

Quantum Homomorphic Encryption

For Clifford circuits

Encryption (of single qubit):

Input state: $|\psi\rangle$

Flip random classical bits a, b

Output: $X^a Z^b |\psi\rangle$, $\mathrm{Enc}(a)$, $\mathrm{Enc}(b)$

Circuit Evaluation:

Apply Clifford gate to quantum part

Homomorphically update classical keys according to commutation relations

Classical homomorphic scheme:
Encryption: $c = \mathrm{Enc}(x)$
Decryption: $x = \mathrm{Dec}(x)$

Example: evaluation of P gate:

- $\mathrm{P}X^a Z^b |\psi\rangle = X^a Z^{a \oplus b} P |\psi\rangle$
- homomorphic update
  $\mathrm{Enc}(b') \leftarrow$ ADD$(\mathrm{Enc}(a), \mathrm{Enc}(b))$

State maintains form:
$X^{a'} Z^{b'} |\psi'\rangle$, $\mathrm{Enc}(\mathrm{a}')$, $\mathrm{Enc}(\mathrm{b}')$

Folklore, last formalized by [BJ15] A. Broadbent, S. Jeffery. Quantum Homomorphic Encryption for Circuits of Low T-gate Complexity. CRYPTO 2015

Extending the gate set: T gate

T gate (also known as $\frac{\pi}{8}$ or R gate) is given by $\mathrm{T} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$

Clifford+T can approximate all quantum operations (universal set)

Trouble: Applying T gate on a one-time-pad encrypted state results in ciphertext: $TX^aZ^b|\psi\rangle = P^aX^aZ^bT|\psi\rangle,\ Enc(a), Enc(b)$

because $TZ = ZT,\ \boldsymbol{TX = PXT}$ *(not Clifford!)*

Who can remove this extra P-gate?

Evaluator only has **encrypted version of $\boldsymbol{a}$ and $\boldsymbol{b}$,** while the encrypting party knows the key

Previous Work: Overview

	homomorphic for	compactness	security
Not encrypting	Quantum circuits	yes	no
append evaluation description	Quantum circuits	Complexity of Dec prop to (# gates)	yes
Quantum OTP	no	yes	inf theoretic
Clifford Scheme	Clifford circuits	yes	computational
[BJ15]: AUX	Q circuits with constant T-depth	yes	computational
[BJ15]: EPR	Quantum circuits	Complexity of Dec prop to (#T-gates)^2	computational
Our result	Quantum circuits of polynomial size (levelled fully homorphic)	yes	computational

[BJ15] A. Broadbent, S. Jeffery. Quantum Homomorphic Encryption for Circuits of Low T-gate Complexity. CRYPTO 2015
(comparison based on Stacey Jeffery's slides)

Related Work

- Secure delegated quantum computing
  - Childs 2005; Broadbent, Fitzsimons, Kashefi 2009; Aharonov, Ben-Or, Eban 2010; Broadbent 2015
- Secure 2-party quantum computation
  - Dupuis, Nielsen, Salvail 2010; Dupuis, Nielsen, Salvail 2010
- Perfectly secure quantum FHE not possible with information-theoretic security
  - Yu, Perez-Delgado, Fitzsimons 2014
- Quantum homomorphic encryption with information leakage (not IND secure)
  - Tan, Kettlewell, Ouyang, Chen, Fitzsimons 2014

require interaction between encryptor and evaluator

(based on Stacey Jeffery's slides)

Roadmap

- ✓ (Classical) Homomorphic Encryption
- ✓ Quantum Homomorphic Encryption
- Computation by Teleportation
- Our scheme

Classical Homomorphic Encryption + Quantum One-Time Pad + Computation by teleportation

Quantum homomorphic encryption for polynomial-sized circuits

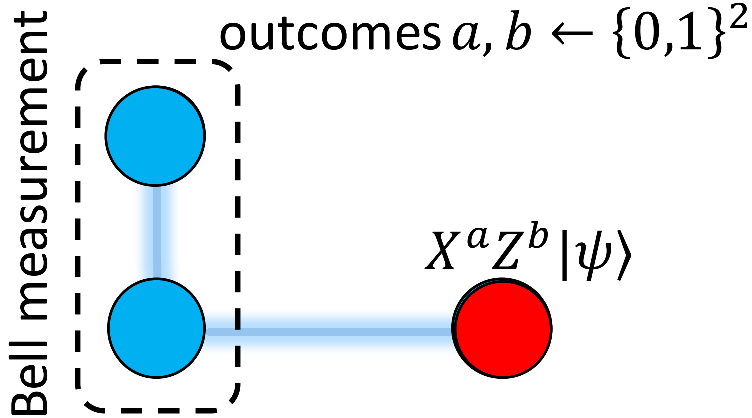
[GC99] Daniel Gottesman and Isaac L. Chuang. Quantum Teleportation is a Universal Computational Primitive. *Nature* '99

Entanglement and Quantum Teleportation

- Entanglement

  EPR pair: $\frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{00}\rangle + \frac{\mathbf{1}}{\sqrt{\mathbf{2}}}|\mathbf{11}\rangle$, state can not be written as two separate qubits

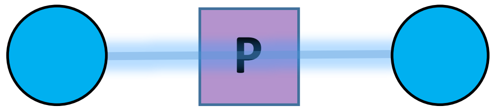
- Teleportation transfers a quantum bit using an EPR pair and two classical bits



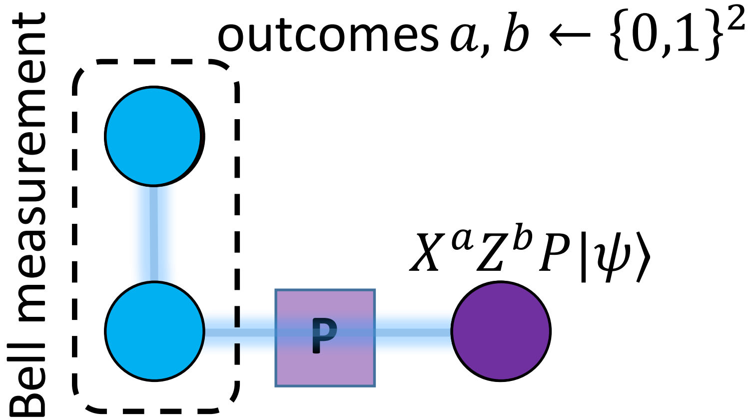
Teleportation of Clifford gates

$$P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

- Start with modified EPR pair: $\frac{1}{\sqrt{2}}|00\rangle + i\,\frac{1}{\sqrt{2}}|11\rangle$



- Teleportation:



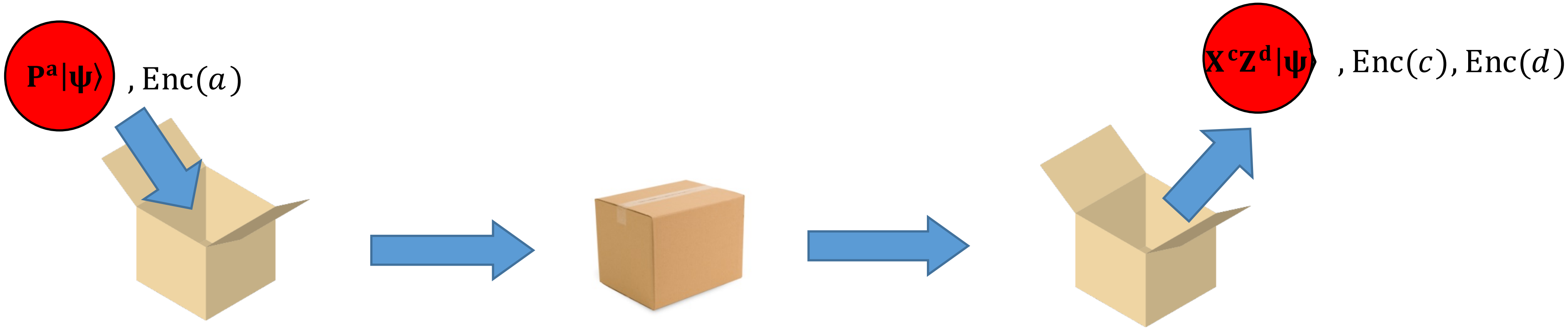
[GC99] Daniel Gottesman and Isaac L. Chuang. Quantum Teleportation is a Universal Computational Primitive. *Nature* '99

Creating a T-gate gadget

$$TX^aZ^b|\psi\rangle$$
$$= P^aX^aZ^bT|\psi\rangle, Enc(a), Enc(b)$$

Who can remove this extra P-gate?

Evaluator only has encrypted version of a, b, while encrypting party knows the key



Depending on a,
a phase gate is applied

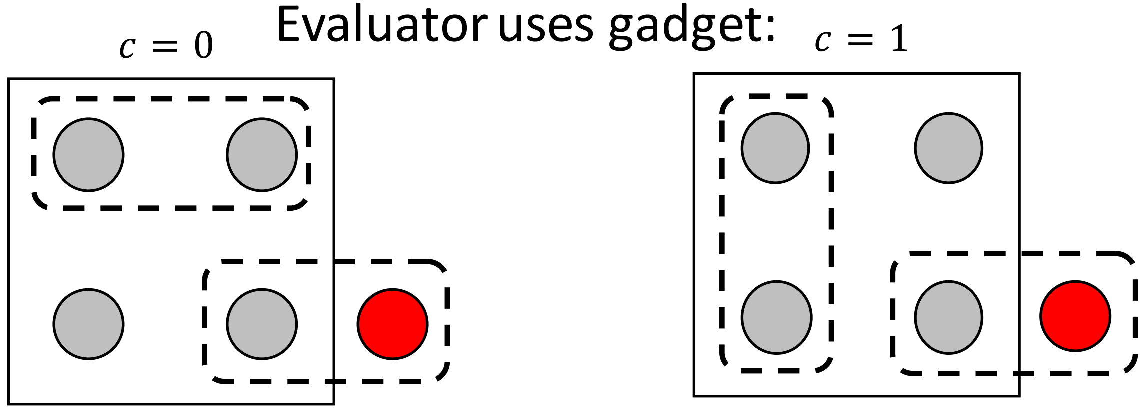
Toy example of gadget

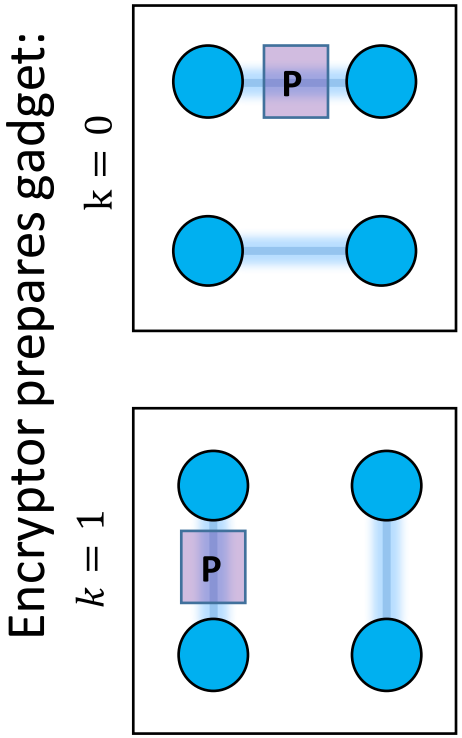
Encrypting party has: $\mathrm{k} \in \{0,1\}$

Evaluator has: $\mathrm{c} = \mathrm{Enc(a)} = a \oplus k \in \{0,1\}$

Want to apply a phase gate if $c \oplus k = a = 1$

Evaluator uses gadget:





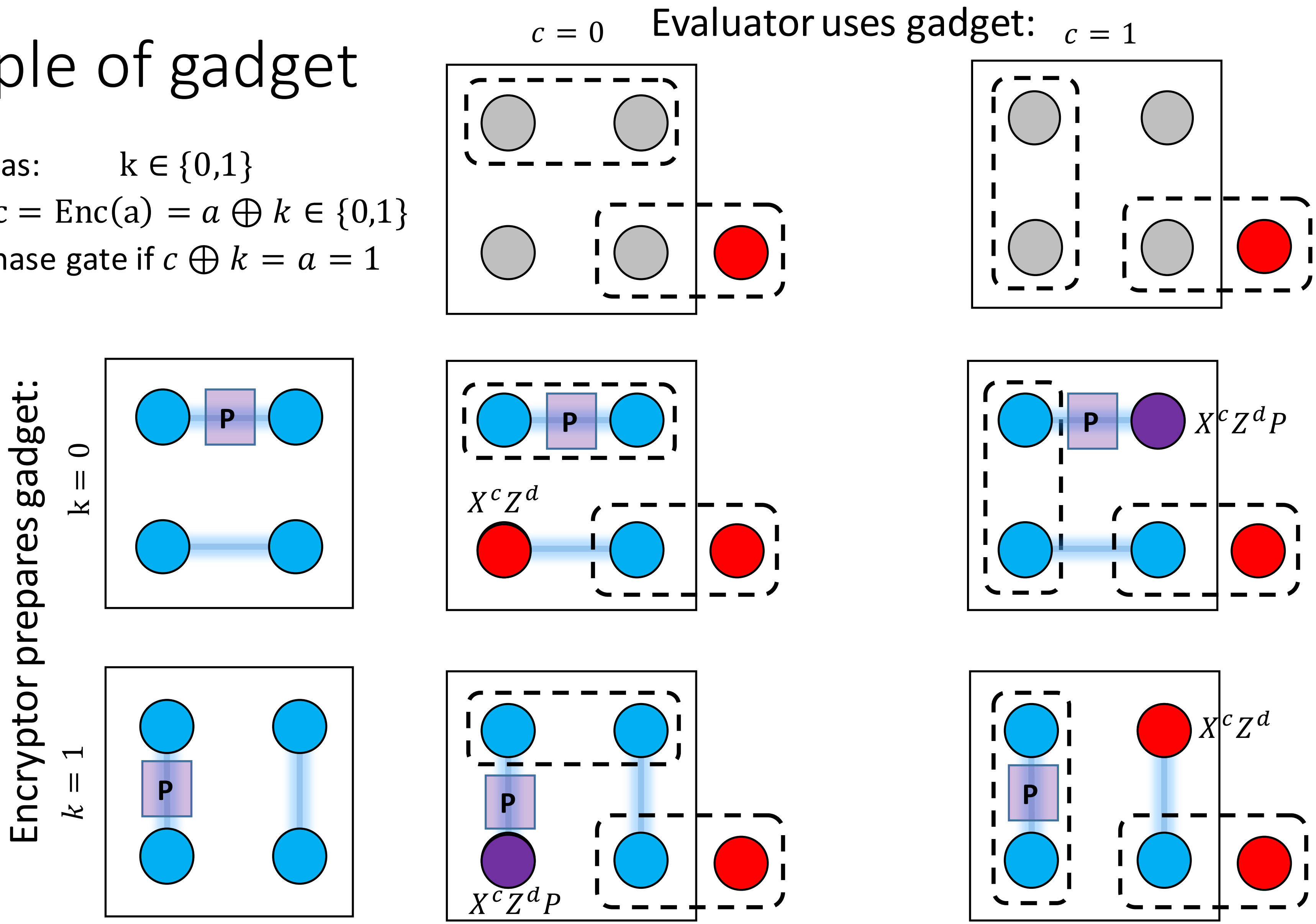
Toy example of gadget

Encrypting party has: $k \in \{0,1\}$

Evaluator has: $c = \mathrm{Enc}(a) = a \oplus k \in \{0,1\}$

Want to apply a phase gate if $c \oplus k = a = 1$

Using a fixed Bell state is insecure, but choice of Bell state can be randomized



Construction of T-gate gadget

- Using **Barrington's theorem**, we can construct gadgets for decryption functions computable in poly-sized log-depth circuits.
- Using techniques from the **garden-hose model**, we can create gadgets for any decryption function computable in log-space.
- Fortunately for us: all known classical homomorphic encryption schemes have a decryption function computable in log-space

[BFSS13] Harry Buhrman, Serge Fehr, Christian Schaffner, and Florian Speelman. The garden-hose model. *ITCS '13*

Homomorphic decryption

Most current schemes are based on *Learning With Errors (LWE)*

Brakerski-Vaikuntanathan (2011):

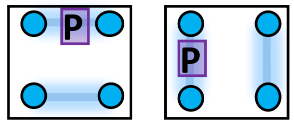
Key: $\boldsymbol{s} \in \mathbb{Z}_p^k$ (vector of length k over $\mathbb{Z}_p$)

Ciphertext: $(\boldsymbol{v}, w) \in \mathbb{Z}_p^k \times \mathbb{Z}_p$

Decryption: $m = w - \sum_{i=1}^{k} \boldsymbol{s}_i \boldsymbol{v}_{\boldsymbol{i}} \pmod{p} \pmod{2}$

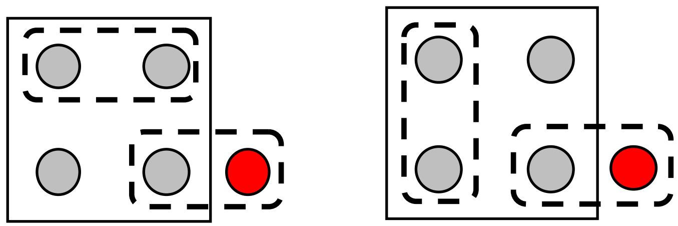
Putting the scheme together

- **Encryption:**
  - Encrypt qubits using Quantum One-Time Pad
  - Use classical HE to encrypt the key to the one-time pad
  - *Create extra helper-gadgets from private key:*



- **Evaluation:**
  - Clifford gates: execute and update keys
  - *T gates: execute and use gadget to correct the state*

measurement choices are given by classical encrypted information



Summary

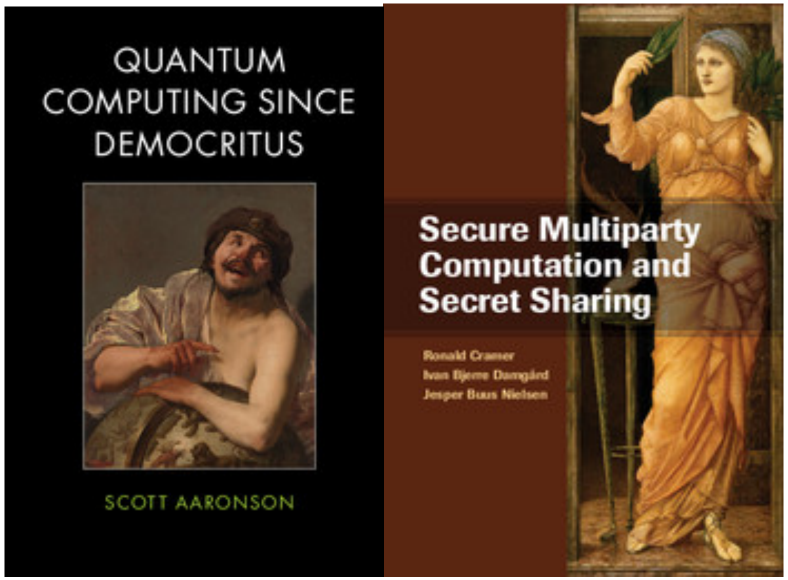
- Scheme for quantum homomorphic encryption
  - Single quantum gadget for every T gate
  - Polynomial-size for all current classical homomorphic schemes
  - We require the computational assumptions of classical scheme

- Main ingredients:
  - Classical homomorphic encryption
  - Quantum one-time pad
  - EPR gadgets (depending on secret key) to *conditionally* remove errors

Open questions / Future work

- Quantum Fully Homomorphic Encryption
  - Currently: helper gadgets required for evaluation of each T gate

- Other cryptographic primitives
  - (round-efficient) delegated quantum computation
  - Quantum Multi-Party Computation
  - Quantum circuit obfuscation
  - …?

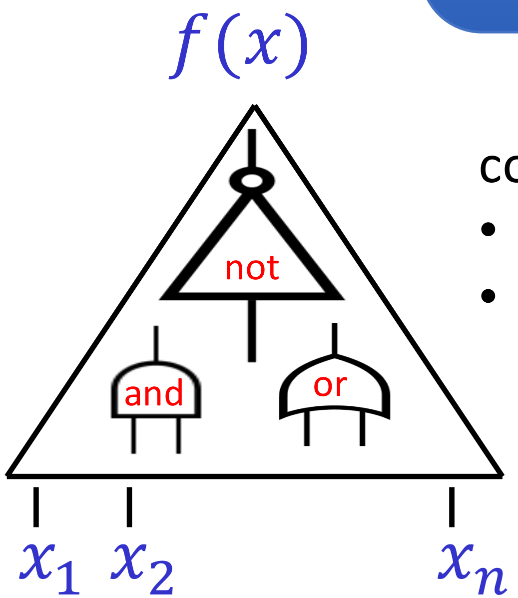


Thank you for your attention!

Questions

Barrington's Theorem [1989]

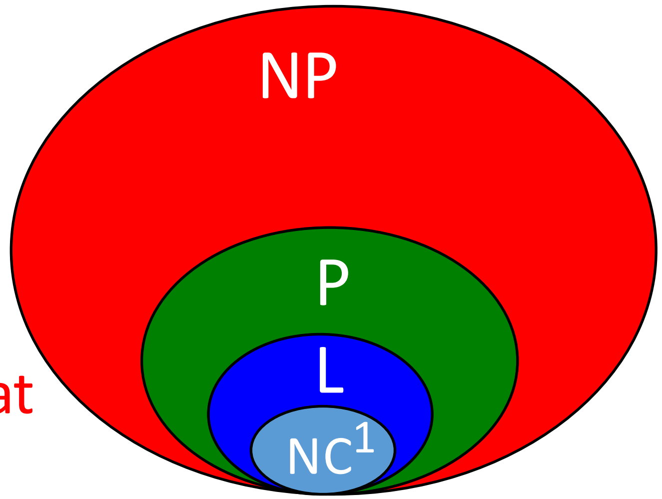
NC^1		Width-5 PBP
Boolean circuits with ■ Fan-in 2 gates ■ Polynomial size ■ Depth is log(n)	=	Branching Programs ■ Polynomial size ■ Permutations from S_5



contains many non-trivial functions:

- Majority, Parity, Equality
- Decryption of common FHE schemes

No proof that
$NP \neq NC^1$



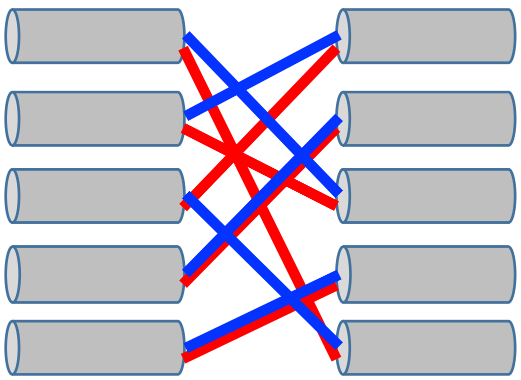
Width-5 Permutation Branching Programs

function: $f:\{0,1\}^3 \rightarrow \{0,1\}$ input: $x_1 x_2 x_3$ (this example: $n = 3$)

instructions: $(\boxed{\sigma_1^0}, \boxed{\sigma_1^1}), (\sigma_2^0, \sigma_2^1), (\sigma_3^0, \sigma_3^1), (\sigma_4^0, \sigma_4^1), \ldots, (\sigma_k^0, \sigma_k^1)$ $\sigma_j^i \in S_5$

$x_1 = 0$

$x_1 = 1$



Width-5 Permutation Branching Programs

function: $f: \{0,1\}^3 \rightarrow \{0,1\}$ input: $x_1 x_2 x_3$ (this example: $n = 3$)

instructions: $(\sigma_1^0, \sigma_1^1), (\sigma_2^0, \sigma_2^1), (\sigma_3^0, \sigma_3^1), (\sigma_4^0, \sigma_4^1), \dots, (\sigma_k^0, \sigma_k^1)$

$x_1 = 0$ $x_2 = 0$ $x_3 = 0$ $x_1 = 0$

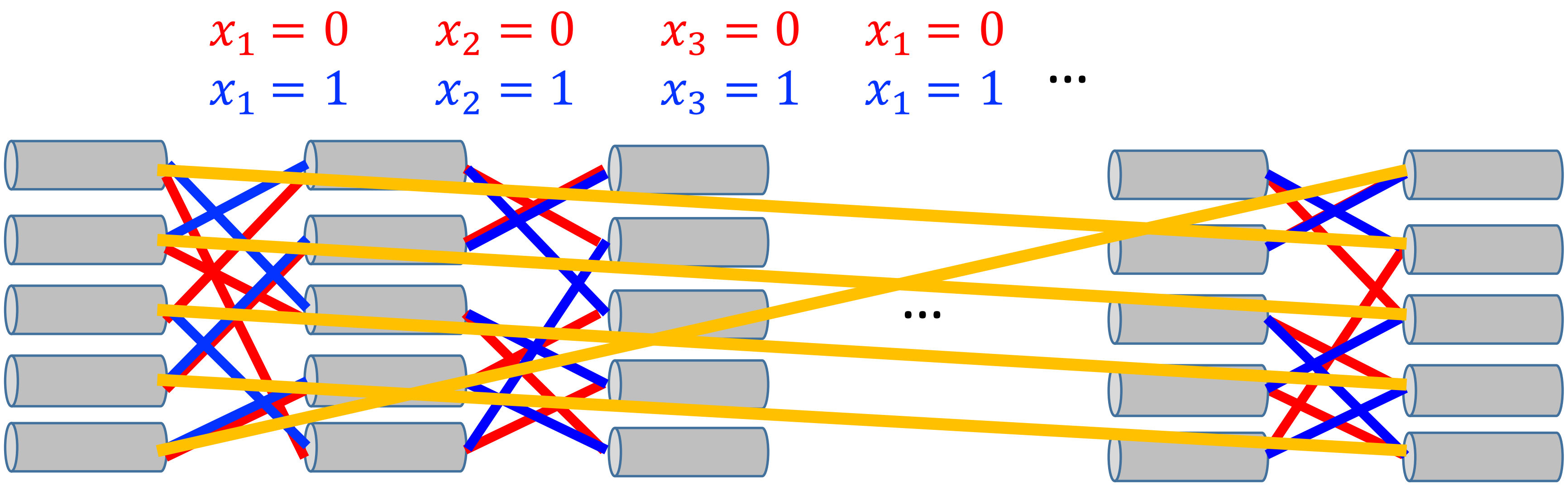
$x_1 = 1$ $x_2 = 1$ $x_3 = 1$ $x_1 = 1$ …

$$\sigma_1^{x_1} \cdot \sigma_2^{x_2} \cdot \sigma_3^{x_3} \cdot \sigma_4^{x_1} \cdots \sigma_k^{x_{k \bmod 3}} = \begin{cases} id & \text{if } f(x_1 x_2 x_3) = 1 \\ \pi & \text{if } f(x_1 x_2 x_3) = 0 \end{cases}$$

Width-5 Permutation Branching Programs

function: $f : \{0,1\}^3 \to \{0,1\}$ input: $x_1 x_2 x_3$ (this example: $n = 3$)

instructions: $(\sigma_1^0, \sigma_1^1), (\sigma_2^0, \sigma_2^1), (\sigma_3^0, \sigma_3^1), (\sigma_4^0, \sigma_4^1), \ldots, (\sigma_k^0, \sigma_k^1)$



$$\sigma_1^{x_1} \cdot \sigma_2^{x_2} \cdot \sigma_3^{x_3} \cdot \sigma_4^{x_1} \cdots \sigma_k^{x_{k \bmod 3}} = \begin{cases} id & \text{if } f(x_1 x_2 x_3) = 1 \\ \pi & \text{if } f(x_1 x_2 x_3) = 0 \end{cases}$$

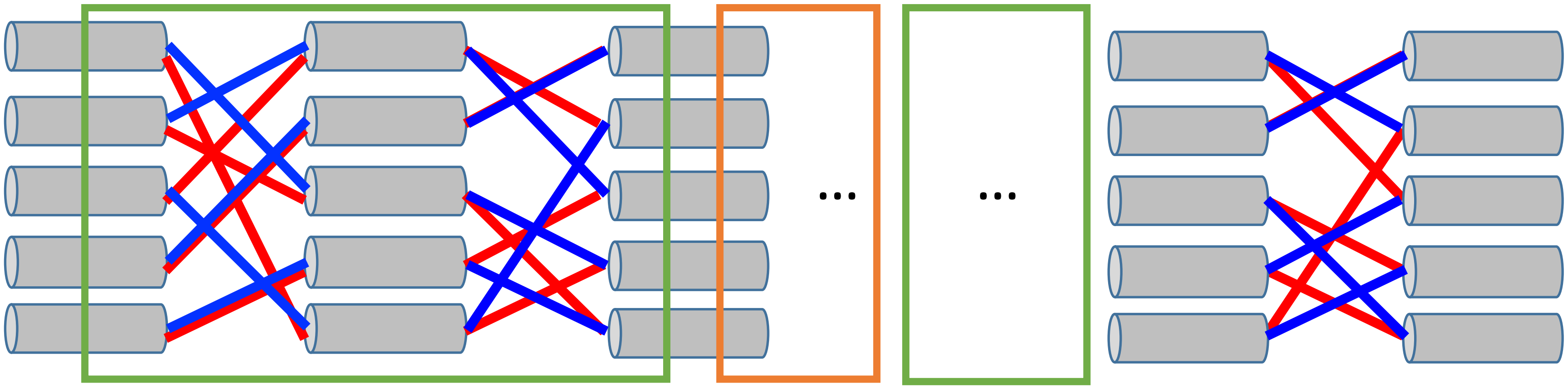
$\pi \in S_5$ a fixed 5-cycle

From Perm Branching Programs to Quantum Gadgets

function: $Dec(\ )$ input: $\boldsymbol{sk}, \boldsymbol{Enc(a)}$

instructions: $\boxed{(\sigma_1^0, \sigma_1^1), (\sigma_2^0, \sigma_2^1),} \boxed{(\sigma_3^0, \sigma_3^1), (\sigma_4^0, \sigma_4^1), \ldots} \boxed{\ldots\ \ldots} \ldots\ \ldots, (\sigma_k^0, \sigma_k^1)$

encryptor evaluator



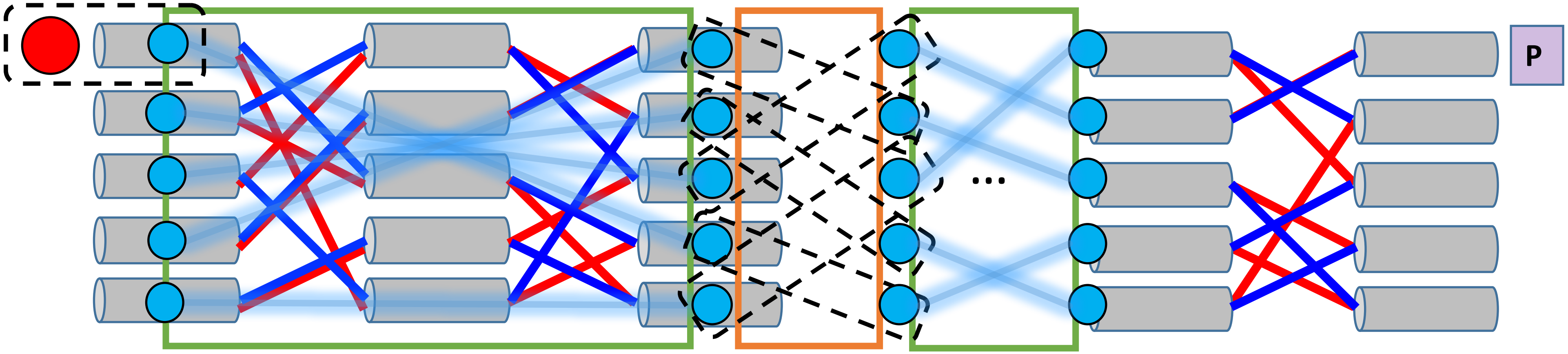
$$\boxed{\sigma_1^{x_1} \cdot \sigma_2^{x_2}} \cdot \boxed{\sigma_3^{x_3} \cdot \sigma_4^{x_1}} \cdot \boxed{\cdots\cdots} \cdot \sigma_k^{x_\cdot} = \begin{cases} id & \text{if } Dec(sk, Enc(a)) = 1 \\ \pi & \text{if } Dec(sk, Enc(a)) = 0 \end{cases}$$

From Perm Branching Programs to Quantum Gadgets

function: $Dec(\ )$ input: $\boldsymbol{sk}, \boldsymbol{Enc(a)}$

instructions: $\boxed{(\sigma_1^0,\sigma_1^1),(\sigma_2^0,\sigma_2^1),}\ \boxed{(\sigma_3^0,\sigma_3^1),(\sigma_4^0,\sigma_4^1),\ldots}\ \boxed{\ldots\ldots}\ \ldots\ldots,(\sigma_k^0,\sigma_k^1)$

encryptor evaluator



$$\boxed{\sigma_1^{x_1}\cdot\sigma_2^{x_2}}\cdot\boxed{\sigma_3^{x_3}\cdot\sigma_4^{x_1}}\cdot\boxed{\cdots\cdots}\cdot\sigma_k^{x_\cdot} = \begin{cases} id & \text{if } Dec(sk,Enc(a)) = 1 \\ \pi & \text{if } Dec(sk,Enc(a)) = 0 \end{cases}$$

- Finally, run all instructions in reverse to get the qubit to a known location