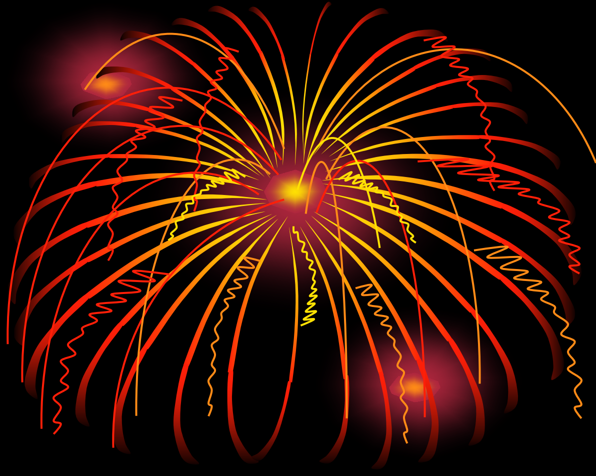
14th Week



Quantum Cryptographic Systems and Quantum Functions

Synopsis.

- Quantum Public-Key Cryptosystems
- Quantum Bit Commitment
- Quantum Hardcore
- Quantum List Decoding
- Quantum Functions

July 9, 2018. 23:59

Course Schedule: 16 Weeks

Subject to Change

- Week 1: Basic Computation Models
- Week 2: NP-Completeness, Probabilistic and Counting Complexity Classes
- Week 3: Space Complexity and the Linear Space Hypothesis
- Week 4: Relativizations and Hierarchies
- Week 5: Structural Properties by Finite Automata
- Week 6: Stype-2 Computability, Multi-Valued Functions, and State Complexity
- Week 7: Cryptographic Concepts for Finite Automata
- Week 8: Constraint Satisfaction Problems
- Week 9: Combinatorial Optimization Problems
- Week 10: Average-Case Complexity
- Week 11: Basics of Quantum Information
- Week 12: BQP, NQP, Quantum NP, and Quantum Finite Automata
- Week 13: Quantum State Complexity and Advice
- Week 14: Quantum Cryptographic Systems and Quantum Functions
- Week 15: Quantum Interactive Proofs and Quantum Optimization
- Week 16: Final Evaluation Day (no lecture)

YouTube Videos

- This lecture series is based on numerous papers of T. Yamakami. He gave conference talks (in English) and invited talks (in English), some of which were video-recorded and uploaded to YouTube.
- Use the following keywords to find a playlist of those videos.
- YouTube search keywords:

  Tomoyuki Yamakami conference invited talk playlist

Conference talk video

Main References by T. Yamakami I

- T. Yamakami. A foundation of programming a multi-tape quantum Turing machine. In Proc. of MFCS 1999, LNCS, Vol.1672, pp.430-441 (1999)
- A. Kawachi and T. Yamakami. Quantum hardcore functions by complexity-theoretical quantum list decoding. SIAM Journal on Computing 39, 2941-2969 (2010)
- A. Kawachi. T. Koshiba, H. Nishimura, and T. Yamakami. Computational indistinguishability between quantum states and its cryptographic application. Journal of Cryptology 25, 528-555 (2012)
- T. Yamakami. Straight construction of non-interactive quantum bit commitment schemes from indistinguishable quantum state ensembles. In the Proc. of TPNC 2015, LNCS, vol. 9477, p. 121-133 (2015)

(To be continued)

Main References by T. Yamakami II

✎ T. Yamakami. Quantum list decoding from quantumly corrupted codewords for classical block codes of polynomially small rate. Baltic Journal of Modern Computing 4(4), 753-788 (2016)

✎ T. Yamakami. A recursive definition of quantum polynomial time computability (extended abstract). In Proc. of NCMA 2017, Österreichische Computer Gesellschaft 2017, the Austrian Computer Society, pp. 243-258 (2017)

I. Schematic Definition of Polynomial-Time Quantum Computation

1. Quantum Strings or Qustrings
2. Quantum Functions from H_∞ to H_∞
3. Schematic Definitions of $\square_1^{QP}$
4. Examples of $\square_1^{QP}$-Functions
5. Characterization of $\square_1^{QP}$

Quantum Strings or Quastrings

$$H_{\infty} = \bigcup_{n \geq 1} H_{2^n}$$

- Let n be any number in N.
- H_n = Hilbert space of dimension n
- We define the size function $\ell : H_{\infty} \to N$.
  - $\ell(|\varphi\rangle) = 0 \Leftrightarrow |\varphi\rangle$ is the null vector, and
  - $\ell(|\varphi\rangle) = n \Leftrightarrow |\varphi\rangle$ is in H_k, where $k = 2^n$ and k>0.

- A quantum string of length (or size) n is a unit-norm vector in the Hilbert space of dimension 2^n.
- We simply call it a qustring of size n.
- When n=0, a qustring is the null vector.
- Φ_n = the set of all qustrings of size n

$$\Phi_{\infty} = \bigcup_{n \geq 0} \Phi_n$$

Quantum Functions from H_∞ to H_∞

- Yamakami (2003) earlier studied quantum functions that produce the acceptance probabilities of quantum computation.
- (*) The above notion was discussed in Week 12.
- Different from the above notion, Yamakami (2017) considered quantum functions that map H_∞ to H_∞.
- We say that such a quantum function is polynomial-time computable if there is a P-uniform family $\{C_n\}_{n\in N}$ of quantum circuits such that, on any input x, $C_{|x|}$ exactly produces the quantum state f(x).

Convention for the Bra- and Ket-Notations

- Here, we use the following conventional notation for bra- and ket-notations.
- Let |φ⟩ be a quantum state in H_m with $m=2^{n+1}$:

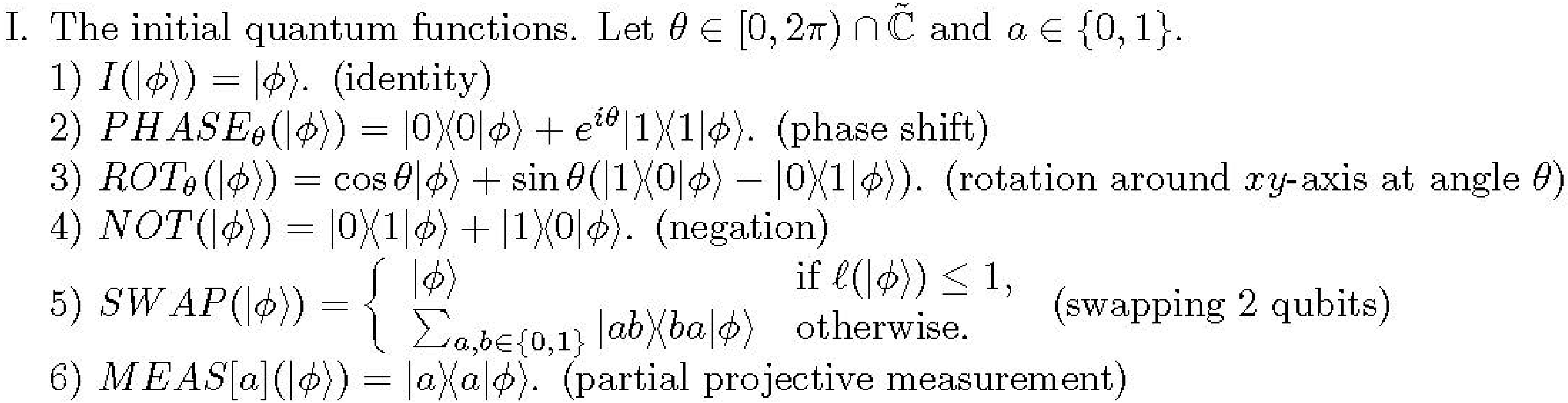
$$\left|\varphi\right\rangle = \sum_{s\in\{0,1\}^{n+1}} \alpha_s \left|s\right\rangle = \sum_{t\in\{0,1\}^n} \left(\alpha_{0t}\left|0t\right\rangle + \alpha_{1t}\left|1t\right\rangle\right)$$

- ⟨0|φ⟩ denotes $\left\langle 0\middle|\varphi\right\rangle = \sum_{t\in\{0,1\}^n} \alpha_{0t}\left|t\right\rangle$
- ⟨1|φ⟩ denotes $\left\langle 1\middle|\varphi\right\rangle = \sum_{t\in\{0,1\}^n} \alpha_{1t}\left|t\right\rangle$

- Hence, it follows that $\left|\varphi\right\rangle = \left\langle 0\middle|\varphi\right\rangle + \left\langle 1\middle|\varphi\right\rangle$

Schematic Definitions of □$_1^{QP}$ I

- □$_1^{QP}$ consists of quantum functions constructed recursively from Scheme I and by applying Schemata II-IV.



I. The initial quantum functions. Let $\theta \in [0, 2\pi) \cap \tilde{\mathbb{C}}$ and $a \in \{0, 1\}$.

1) $I(|\phi\rangle) = |\phi\rangle$. (identity)

2) $PHASE_\theta(|\phi\rangle) = |0\rangle\langle 0|\phi\rangle + e^{i\theta}|1\rangle\langle 1|\phi\rangle$. (phase shift)

3) $ROT_\theta(|\phi\rangle) = \cos\theta|\phi\rangle + \sin\theta(|1\rangle\langle 0|\phi\rangle - |0\rangle\langle 1|\phi\rangle)$. (rotation around xy-axis at angle θ)

4) $NOT(|\phi\rangle) = |0\rangle\langle 1|\phi\rangle + |1\rangle\langle 0|\phi\rangle$. (negation)

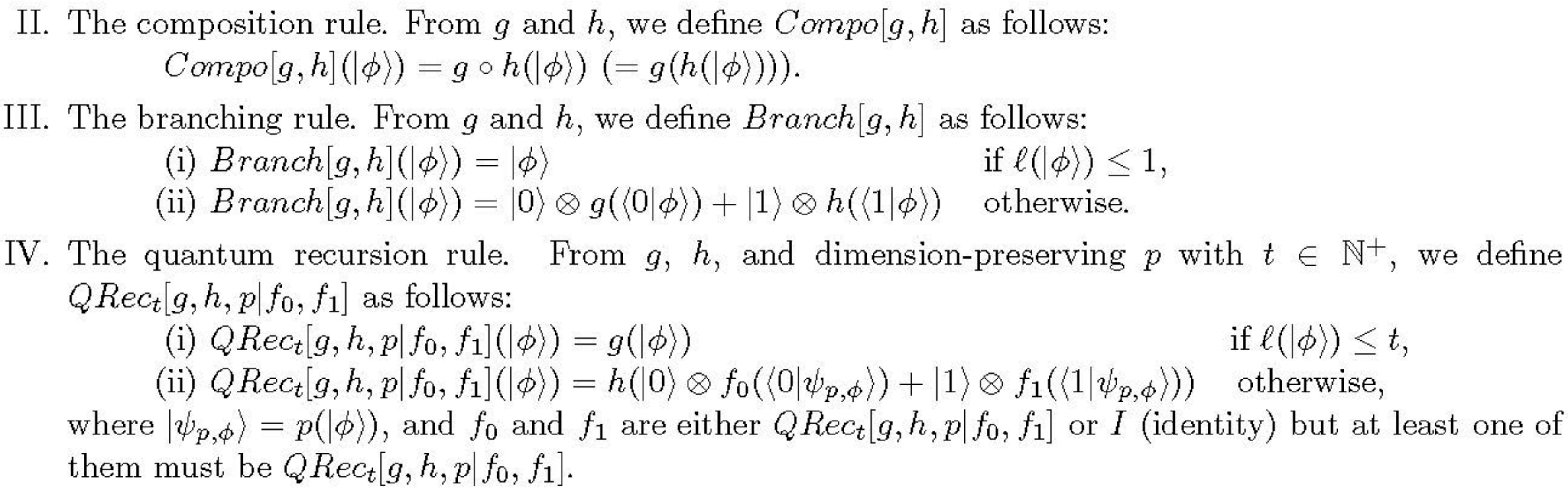
5) $SWAP(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) \leq 1, \\ \sum_{a,b\in\{0,1\}} |ab\rangle\langle ba|\phi\rangle & \text{otherwise.} \end{cases}$ (swapping 2 qubits)

6) $MEAS[a](|\phi\rangle) = |a\rangle\langle a|\phi\rangle$. (partial projective measurement)

- □$_1^{QP*}$ is a subclass of □$_1^{QP*}$ defined by all schemes except for MEAS[ ].

Schematic Definitions of □$_1^{\mathrm{QP}*}$ II

- □$_1^{\mathrm{QP}*}$ consists of quantum functions constructed recursively from Scheme I and by applying Schemata II-IV.



II. The composition rule. From g and h, we define $Compo[g,h]$ as follows:

$$Compo[g,h](|\phi\rangle) = g \circ h(|\phi\rangle) \ (= g(h(|\phi\rangle))).$$

III. The branching rule. From g and h, we define $Branch[g,h]$ as follows:

$$\begin{array}{lll} \text{(i)} & Branch[g,h](|\phi\rangle) = |\phi\rangle & \text{if } \ell(|\phi\rangle) \leq 1, \\ \text{(ii)} & Branch[g,h](|\phi\rangle) = |0\rangle \otimes g(\langle 0|\phi\rangle) + |1\rangle \otimes h(\langle 1|\phi\rangle) & \text{otherwise.} \end{array}$$

IV. The quantum recursion rule. From g, h, and dimension-preserving p with $t \in \mathbb{N}^+$, we define $QRec_t[g,h,p|f_0,f_1]$ as follows:

$$\begin{array}{lll} \text{(i)} & QRec_t[g,h,p|f_0,f_1](|\phi\rangle) = g(|\phi\rangle) & \text{if } \ell(|\phi\rangle) \leq t, \\ \text{(ii)} & QRec_t[g,h,p|f_0,f_1](|\phi\rangle) = h(|0\rangle \otimes f_0(\langle 0|\psi_{p,\phi}\rangle) + |1\rangle \otimes f_1(\langle 1|\psi_{p,\phi}\rangle)) & \text{otherwise,} \end{array}$$

where $|\psi_{p,\phi}\rangle = p(|\phi\rangle)$, and f_0 and f_1 are either $QRec_t[g,h,p|f_0,f_1]$ or I (identity) but at least one of them must be $QRec_t[g,h,p|f_0,f_1]$.

Examples of □$_1^{QP}$-Functions

- Controlled-NOT

$$CNOT\left(|\varphi\rangle\right) = \begin{cases} |\varphi\rangle & \text{if } \ell\left(|\varphi\rangle\right) \le 1, \\ |0\rangle\langle 0|\varphi\rangle + |1\rangle\langle 1|\varphi\rangle & \text{otherwise.} \end{cases}$$

CNOT = Branch[I,NOT]

- Walsh-Hadamard transform

$$WH\left(|\varphi\rangle\right) = \frac{1}{\sqrt{2}}|0\rangle \otimes \left(\langle 0|\varphi\rangle + \langle 1|\varphi\rangle\right) + \frac{1}{\sqrt{2}}|1\rangle \otimes \left(\langle 0|\varphi\rangle - \langle 1|\varphi\rangle\right)$$

WH = Comp[ROT$_{\pi/4}$,NOT]

- k-qubit QFT (quantum Fourier transform) k≥2

$$F_k\left(|\varphi\rangle\right) = \begin{cases} |\varphi\rangle & \text{if } \ell\left(|\varphi\rangle\right) < k, \\ \dfrac{1}{2^{k/2}} \sum_{t:|t|=k} \sum_{s:|s|=k} \omega_k^{num(s)num(t)} \, |s\rangle\langle t|\varphi\rangle & \text{otherwise.} \end{cases}$$

Characterization of FBQP by $\square_1^{QP}$ I

- We take the following encoding (with a blank symbol b).
  - Let $0^* = 00$, $1^* = 01$, $b^* = 10$, $2^* = 11$, and $3^* = 10$.
  - For a string $s = s_1s_2 \ldots s_n$ with $s_i \in \{0,1,b\}$, we set $s^* = s_1^*s_2^* \ldots s_n^*$.
- Note that $|s^*| = 2|s|$.

- Take a polynomial p. Define:
  - $|\phi^p(x)\rangle = |0^{p(|x|)}01^{9p(|x|)}1\rangle|x\rangle$.
  - $|\phi^{p,f}(x)\rangle = |0^{f(|x|)^*}\rangle|\phi^p(x)\rangle$.
  - $|\phi_g^p(x)\rangle = g(|\phi^p(x)\rangle)$ and $|\phi_g^{p,f}(x)\rangle = g(|\phi^{p,f}(x)\rangle)$.

Characterization of FBQP by $\square_1^{QP}$ II

- Yamakami (2017) proved the following characterization of FBQP in terms of $\square_1^{QP*}$.
- Theorem: [Yamakami (2017)]

  Let f be a function on {0,1}*. The following 3 statements are logically equivalent to each other.

  1. f is computable in polynomial time (i.e., $f \in$ FBQP).
  2. For any constant $\varepsilon \in [0,1/2)$, there exists a quantum function g in $\square_1^{QP*}$ and a polynomial p such that, for all $x \in \{0,1\}^*$, $|f(x)| \leq p(|x|)$ and $\| \langle f(x)^* | \phi_g^p(x) \rangle \|^2 \geq 1-\varepsilon$.
  3. For any constant $\varepsilon \in [0,1/2)$, there exists a quantum function g in $\square_1^{QP}$ and a polynomial p such that, for all $x \in \{0,1\}^*$, $|f(x)| \leq p(|x|)$ and $\| \langle \Psi_{f(x)} | \phi_g^{p,f}(x) \rangle \|^2 \geq 1-\varepsilon$, where $|\Psi_{f(x)}\rangle = |f(x)\rangle|\phi_g^p(x)\rangle$.

Open Problems

- Here is a nagging open problem associated with the schematic definitions.

- Find a simpler, more reasonable schematic definition for $\square_1^{QP}$-functions, which should be capable of precisely characterizing BQP and FBQP.

II. An Ensemble of Quantum States

1. Permutations and Symmetric Groups
2. Special Quantum States
3. Properties of Quantum States
4. Distinguishability
5. Relevant Background
6. k-QSCD
7. Advantage and Average Advantage

Density Operators or Matrices

- There is another way to express quantum states using matrices. Let I be any nonempty index set.

- A density operator ρ associated with an ensemble $\{ p_i, |\psi_i\rangle \mid i \in I \}$ has the form

$$\rho = \sum_{i \in I} p_i |\psi_i\rangle\langle\psi_i| \qquad \text{(provided that } \sum_{i \in I} p_i = 1\text{)}$$

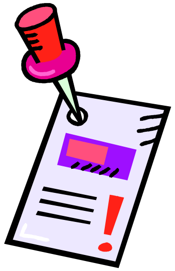
- Equivalently, ρ satisfies the following two conditions:
  1. ρ has trace equal to one, and
  2. ρ is a positive operator.

- A completely mixed state ι is of the form

$$\iota = \frac{1}{|I|} \sum_{z \in I} |z\rangle\langle z|$$

An Ensemble of Special Quantum States

- For a later use, we want to introduce an ensemble of special quantum states, which are obtained in a group-theoretical manner.
- We start with symmetric groups consisting of permutations.

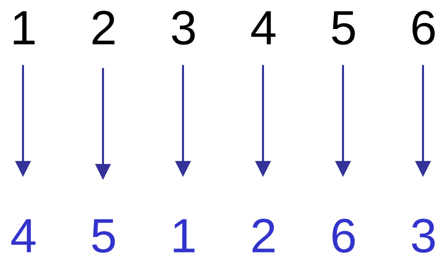


Permutations and Symmetric Groups

- n: security parameter (even and n/2 is odd)
- S_n : the symmetric group of degree n (i.e., the set of all permutations on {1,2,…,n})
- Each permutation π can be expressed in binary using O(nlog(n)) bits.
- Define $K_n = \{ \pi \in S_n \mid \pi^2 = \mathrm{id}, \forall i\ [\, \pi(i) \neq i \,] \} \subseteq S_n$
- In what follows, we take an arbitrary permutation π in K_n.

n = 6

permutation π



1 2 3 4 5 6

4 5 1 2 6 3

$\forall i\ [\, \pi(i) \neq i \,]$

Special Quantum States I

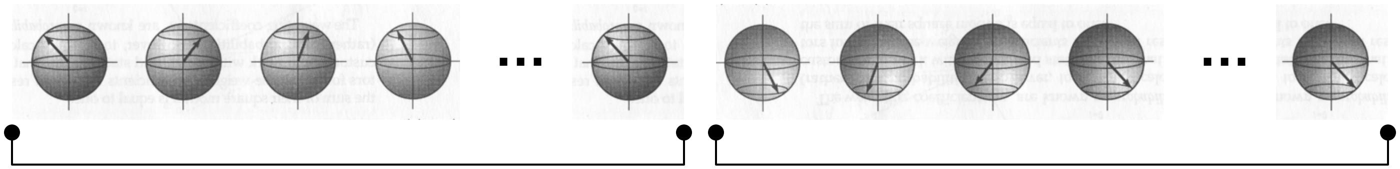
- n: security parameter (even and n/2 is odd)
- Choose a hidden permutation $\pi \in K_n$ and a bit $b \in \{0,1\}$.

- We define $|\phi^{(\pi)}{}_{\sigma b}\rangle$, $|\Phi^{(\pi)}{}_{b}\rangle$, and $\rho^{(\pi)}{}_{b}$ as follows.
  - $|\phi^{(\pi)}{}_{\sigma b}\rangle = (1/\sqrt{2})(\,|\sigma\rangle + (-1)^{b}|\sigma\pi\rangle\,)$
  - $|\Phi^{(\pi)}{}_{b}\rangle = (1/|S_n|)\Sigma_{\sigma\in Sn}|\sigma\rangle|\phi^{(\pi)}{}_{\sigma b}\rangle$

1st register

2nd register

$|\Phi^{(\pi)}{}_{b}\rangle =$



$(1/|S_n|)\ \Sigma_{\sigma\in Sn}$ $|\sigma\rangle$ $\otimes$ $|\phi^{(\pi)}{}_{\sigma b}\rangle$

Special Quantum States II

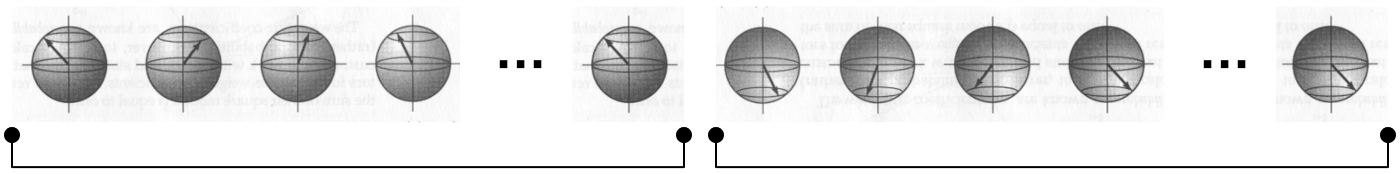
- Trace out the first register of $|\Phi^{(\pi)}_b\rangle$ to obtain:
  - $\rho^{(\pi)}_b = \mathrm{tr}_1(|\Phi^{(\pi)}_b\rangle\langle\Phi^{(\pi)}_b|)$.
- Note that $\rho^{(\pi)}_b = \mathrm{tr}_1(|\Phi^{(\pi)}_b\rangle\langle\Phi^{(\pi)}_b|) = \mathrm{tr}_2(|\Phi^{(\pi)}_b\rangle\langle\Phi^{(\pi)}_b|)$.

- In other words,
  - $\rho^{(\pi)}_0 = (1/2n!)\Sigma_{\sigma\in Sn}(|\sigma\rangle+|\sigma\pi\rangle)(\langle\sigma|+\langle\sigma\pi|)$
  - $\rho^{(\pi)}_1 = (1/2n!)\Sigma_{\sigma\in Sn}(|\sigma\rangle-|\sigma\pi\rangle)(\langle\sigma|-\langle\sigma\pi|)$

1st register

2nd register

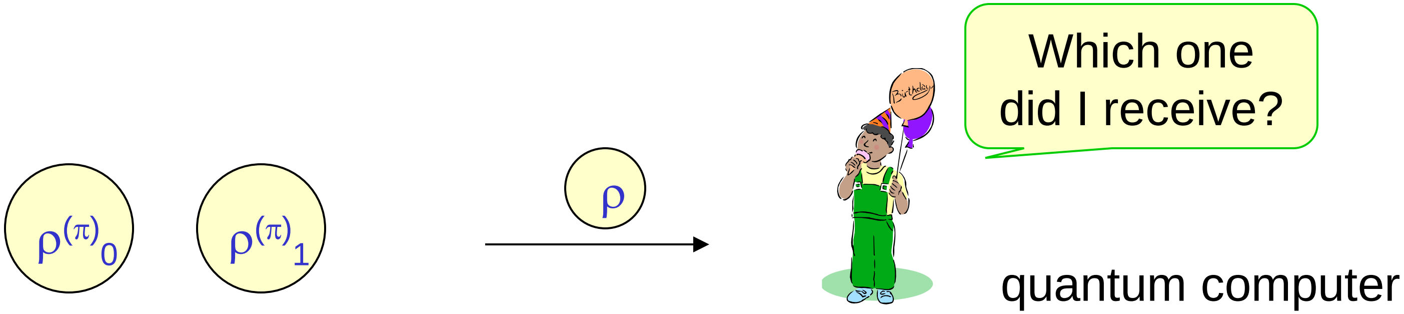
$|\Phi^{(\pi)}_b\rangle =$



$(1/|S_n|)\ \Sigma_{\sigma\in Sn}$ $|\sigma\rangle$ $\otimes$ $|\phi^{(\pi)}_{\sigma b}\rangle$

Properties of the Quantum States

- We have defined quantum states $\rho^{(\pi)}_0$ and $\rho^{(\pi)}_1$. Distinguishing these two quantum states is in general difficult for a quantum computer.
- More precisely, it is hard to distinguish between $\rho^{(\pi)}_0$ and $\rho^{(\pi)}_1$ with high probability using polynomial-time quantum computation.



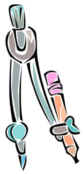
$\rho^{(\pi)}_0$ $\rho^{(\pi)}_1$

ρ

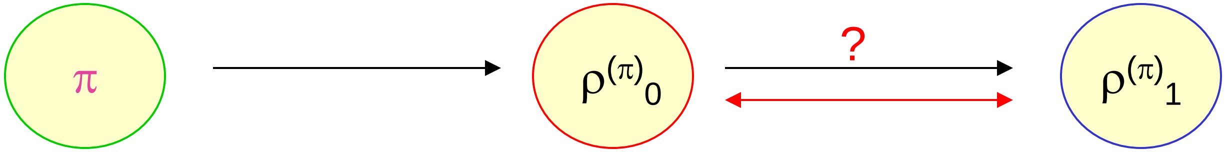
Which one did I receive?

quantum computer

What is Easy to Do?



- It is easy to generate $\rho^{(\pi)}_0$ from $\pi \in K_n$ (by Hadamard and Controlled-π).
- It is easy to convert $\rho^{(\pi)}_0$ to $\rho^{(\pi)}_1$ and keep ι as it is with certainty (by phase encoding).
- It is easy to distinguish between $\rho^{(\pi)}_0$ and $\rho^{(\pi)}_1$ with certainty <u>if π is known</u> (by Hadamard, Controlled-π, and the property π^2=id). (trapdoor property)

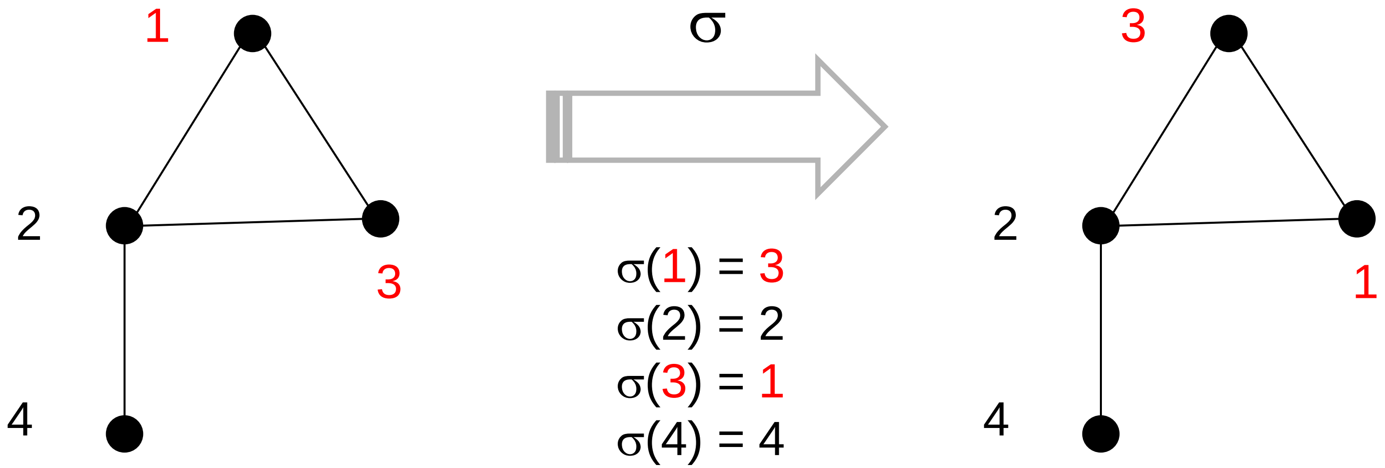


- However, it seems difficult to distinguish them if we do not know π

What is a Graph Automorphism? I

- The distinction problem between $\rho^{(\pi)}_0$ and $\rho^{(\pi)}_1$ is related to the graph isomorphism problem.

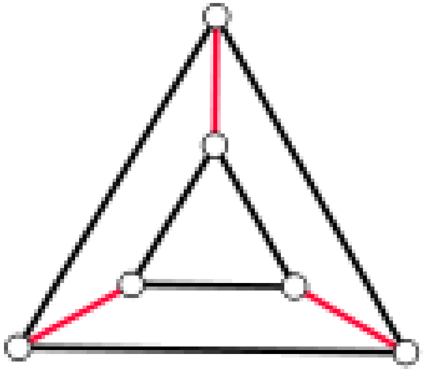
- An automorphism of a graph $G = (V,E)$ is a permutation σ of the vertex set V, such that the pair of vertices (u,v) form an edge iff the pair $(\sigma(u),\sigma(v))$ also form an edge.

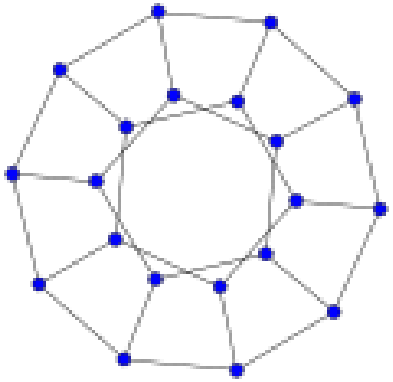


$\sigma(1) = 3$
$\sigma(2) = 2$
$\sigma(3) = 1$
$\sigma(4) = 4$

What is a Graph Automorphism? II

- There are practical applications of graph automorphism.
- For example,
  1. graph drawing and other visualization tasks,
  2. solving structured instances of Boolean satisfiability arising in the context of formal verification

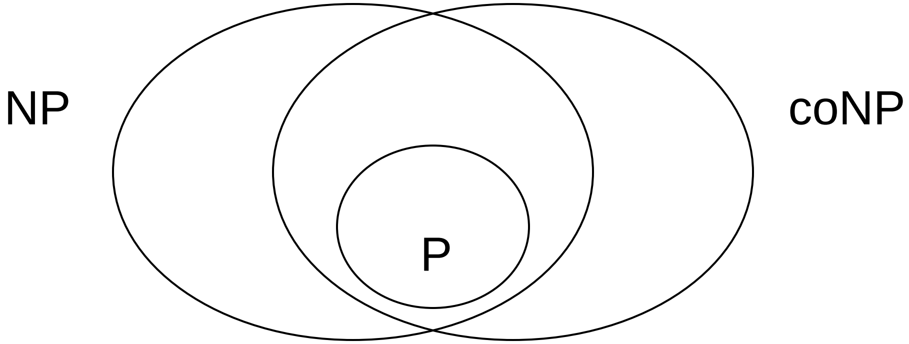




Graph Automorphism Problem (GA)

- Graph Automorphism Problem (GA)
  - Input: an undirected graph G=(V,E);
  - Output: YES if G has a non-trivial automorphism, and NO otherwise.

GA is not known to be in P or NP∩coNP.



How Difficult is it to Distinguish Quantum States?

- Theorem: [Kawachi- Koshiba- Nishimura-Yamakami (2012)]

  If we can efficiently distinguish between those two quantum states on the average (for a uniformly random π), then we can distinguish them even in the worst case.

- Theorem: [Kawachi- Koshiba- Nishimura-Yamakami (2012)]

  If we can efficiently distinguish those two quantum states, then we can efficiently solve the graph automorphism problem.

Relevant Background

- Our distinction problem (between $\rho^{(\pi)}_0$ and $\rho^{(\pi)}_1$) is closely related to the so-called hidden subgroup problem (HSP) on the symmetric groups.
  - This HSP seems very hard to solve even on a quantum computer.
- It is shown that a “natural” extension of Shor’s algorithm cannot solve the distinction problem between $\rho^{(\pi)}_0$ and ι (completely mixed state) [Hallgren-Moore-Rötteler-Russell-Sen (2006)].
  - We can show that our distinguishability problem can be reduced from the distinguishability between $\rho^{(\pi)}_0$ and ι.

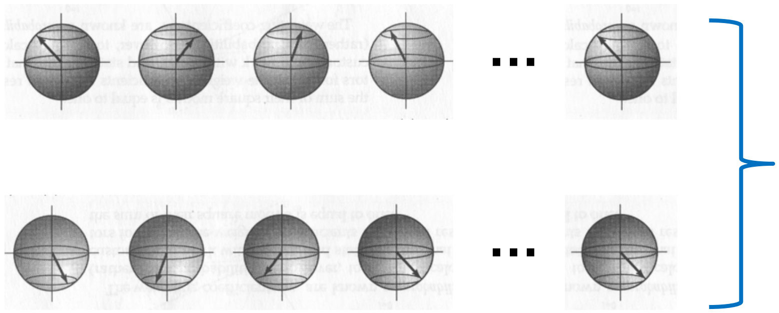
k-Quantum State Computational Distinction Problem (k-QSCD)

- We introduce our distinction problem on k quantum states.

- k-Quantum State Computational Distinction Problem
  - Instance: 1^n, $\rho^{\otimes k}$ with $\rho \in \{\ \rho^{(\pi)}_0(n),\ \rho^{(\pi)}_1(n)\ \}$ for a fixed but hidden permutation $\pi \in K_n$.
  - Output: YES, if $\rho = \rho^{(\pi)}_0(n)$; NO, otherwise.

$\rho^{(\pi)}_0(n)$

$\rho^{(\pi)}_1(n)$



Can we distinguish these two reduced states?

Advantage and Average Advantage

- M: quantum algorithm, π: permutation in K_n
- M solves k-QSCD with advantage p(n) w.r.t. π $\Leftrightarrow$ M distinguishes between $\{\rho^{(\pi)}_0(n)^{\otimes k}\}_n$ and $\{\rho^{(\pi)}_1(n)^{\otimes k}\}_n$ with advantage p(n); that is, for every n,

$$p(n) = \left| \mathrm{Prob}[A(1^n, \rho_0^{(\pi)}(n)^{\otimes k}) = 1] - \mathrm{Prob}[A(1^n, \rho_1^{(\pi)}(n)^{\otimes k}) = 1] \right|$$

- M solves k-QSCD with average advantage γ for length n $\Leftrightarrow$ γ = the expectation, over all $\pi \in K_n$ chosen uniformly at random, of the advantage with which A distinguishes between $\{\rho^{(\pi)}_0(n)^{\otimes k}\}_n$ and $\{\rho^{(\pi)}_1(n)^{\otimes k}\}_n$.

III. Quantum Public-Key Cryptosystems

1. A Scheme of PKC
2. Can We Distinguish Two Quantum States?
3. How Difficult to Distinguish Quantum States?
4. How to Build a Secure Cryptosystem?

A Scheme of PKC

- We want to construct a presumably secure quantum public-key cryptosystem (QPKC).
- We quickly mention a scheme of PKC.
- Assume that Alice wants to send a bit b to Bob securely.

❖ Alice

1. She encodes b to an encoded string χ_b.
2. She sends χ_b to Bob through an unsecure channel.

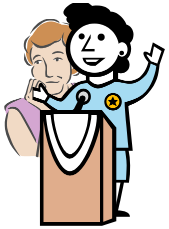
❖ Bob

1. He receives χ_b.
2. He decodes χ_b back to b.

- Requirement: Eavesdropper Eve cannot know what b is.

Why Public-Key Cryptosystems?

SKCs vs. PKCs

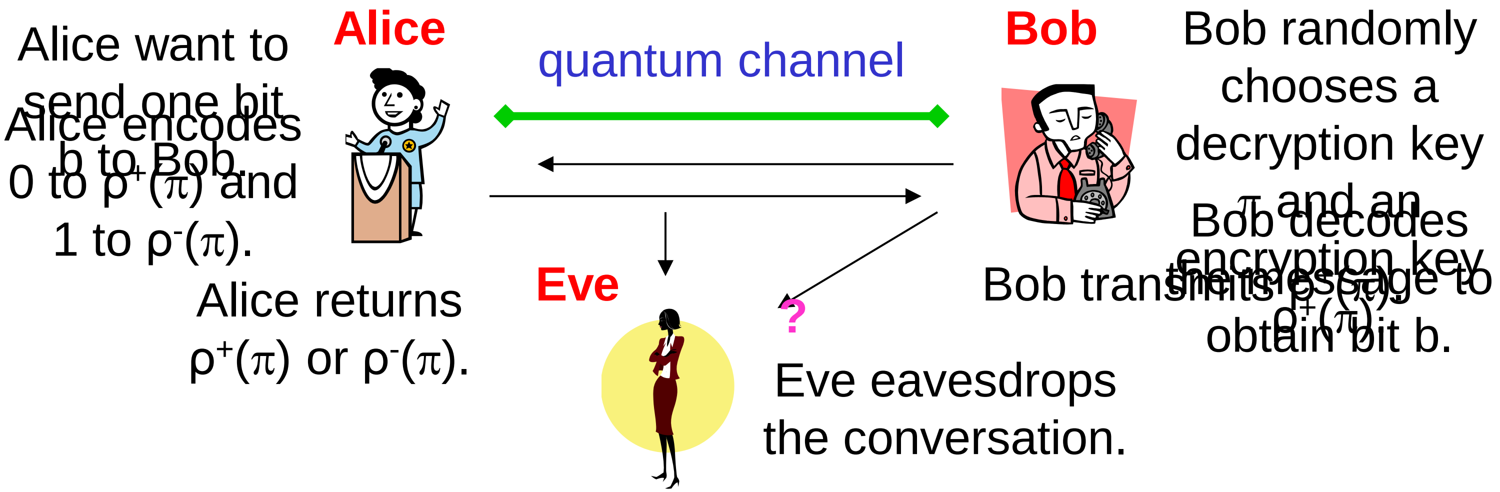


- Advantages and disadvantages of symmetric-key cryptosystems (SKCs) and public-key cryptosystems (PKCs)

1. Quantum key distribution protocol BB84 achieves unconditionally secure sharing of secret keys for SKCs using an authenticated communication channel.
2. However, SKCs require a number of secret keys in a large scale network.
3. By contrast, PKCs can save a number of secret keys in such a large network.
4. It is known that PKCs are vulnerable to the man-in-the-middle attack.

How to Build a Secure quantum PKC

- We can build a “secure” quantum public-key cryptosystem (quantum PKC) against the chosen plaintext quantum attack (during message transmission) using the quantum state indistinguishability.
- Our cryptosystem works as follows.



Alice want to send one bit b to Bob.

Bob randomly chooses a decryption key π and an encryption key $\rho^{+}(\pi)$.

Bob transmits $\rho^{+}(\pi)$.

Alice encodes 0 to $\rho^{+}(\pi)$ and 1 to $\rho^{-}(\pi)$.

Alice returns $\rho^{+}(\pi)$ or $\rho^{-}(\pi)$.

Bob decodes the message to obtain bit b.

Eve eavesdrops the conversation.

Open Problems

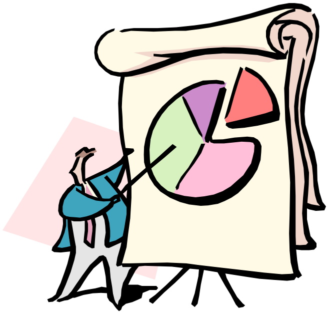
- Studying quantum cryptographic primitives
  - Quantum one-way functions and quantum hardcores
  - Quantum commitment
  - Quantum oblivious transfer
  - Quantum zero-knowledge proof systems
- Finding relationships to other complexity issues
  - Black-box oracle computation
  - Quantum state distinguishability
- Building secure quantum cryptosystems
  - Public-key encryption schemes

IV. Quantum Bit Commitment

1. Bit Commitment
2. Quantum Bit Commitment
3. Hiding Conditions
4. Binding Conditions
5. Limitation of QBC Schemes
6. First Theorem
7. Second Theorem

Bit Commitment

- Bit commitment (BC) is a fundamental cryptographic primitive.
- BC consists of two phases.
  - Committing phase
  - Opening (or Revealing) phase
- BC has various applications to:
  - Secure coin flipping
  - Zero-knowledge proofs
  - Secure multiparty protocols
  - Signature schemes
  - Secret sharing



Quantum Bit Commitment

- A quantum bit commitment (QBC) scheme consists of the following two phases.
- Committing Phase
  - Alice commits to a bit $a \in \{0,1\}$.
  - She encrypts a to a quantum state.
  - She sends a reduced quantum state χ to Bob.
- Opening Phase
  - Alice reveals a to Bob.
  - Additionally, she sends extra information on a and χ.
  - Bob verifies that a is correct, using χ.

Hiding Conditions for QBC Schemes I

- QBC schemes must satisfy the hiding and binding conditions.
- Here, we use the formalism of Dumais, Mayers, and Salvail (2000).
- Let A be a QBC scheme and n be a security parameter
- In Committing Phase, Alice starts with $|0\rangle$.
- She commits to a bit $a \in \{0,1\}$.
- She applies a quantum operator U_1 to encrypt a.
- She sends a reduced quantum state χ_a to Bob.
- The hiding condition ensures that Bob cannot know Alice’s committed bit before the opening phase.

Hiding Conditions for QBC Schemes II

- Recall that A is a QBC scheme and n is a security parameter
- In the committing phase, Bob receives a reduced quantum state, either χ_0 or χ_1.

- Computationally hiding
- For any positive polynomial p, no polynomial-time quantum algorithm outputs a from instance χ_a with success probability at least 1/2+1/p(n) for all n in N.

- Perfectly hiding
- $\chi_0 = \chi_1$

Binding Conditions for QBC Schemes I

- Let A be a QBC scheme, and n be a security parameter
- Let $U = (U_1, U_2^{(0)}, U_2^{(1)})$ be Alice’s cheating strategy
- From the beginning, Alice plans to deceive Bob by revealing a willfully chosen bit $b \in \{0,1\}$.
- Let $U_2^{(b)}$ be the operator Alice secretly applies, according to b.
- Let $T_b^{(U)}(n)$ be the probability that Bob convinces himself that b is her committed bit after she applies $U_2^{(b)}$, provided that Bob faithfully follows the scheme
- Note that $0 \leq \frac{1}{2}\left(T_0^{(U)}(n) + T_1^{(U)}(n)\right) \leq 1$ (average value)
- The binding condition says that Alice cannot change her mind to cheat Bob during the whole scheme.

Binding Conditions for QBC Schemes II

- Recall that A is a QBC scheme and n is a security parameter.
- $U = (U_1, U_2^{(0)}, U_2^{(1)})$: Alice's cheating strategy

- Computationally binding
- There exists a negligible function $\varepsilon(n)$ s.t., for any poly-time computable cheating strategy $U = (U_1, U_2^{(0)}, U_2^{(1)})$, the average success probability $(1/2)(T_0^{(U)}(n) + T_1^{(U)}(n))$ is at most $1/2 + \varepsilon(n)$ for every n in N.

- Statistically binding
- In the above definition, Alice can use time-unbounded cheating strategy.

Limitation of QBC Schemes

- We say that a QBC scheme is unconditionally secure if it is both statistically hiding and statistically binding.
- Unfortunately, it is known that we cannot achieve the unconditional security.
- Theorem: [Lo-Chau (1997), Mayers (2001)]

  No QBC scheme is unconditionally secure (that is).

First Theorem

- We obtain the following results.

- Theorem: [Yamakami (2015)]
  1) There exists a scheme for non-interactive QBC for which the scheme is polynomial-time executable and has an explicit, direct construction from the ensembles $\{\rho^{(\pi)}_0(n),\rho^{(\pi)}_1(n)\}_{n,\pi}$.
  2) Moreover, if no quantum algorithm solves k-QSCD in polynomial time with non-negligible average advantage for a certain $k\geq 2$, then the scheme achieves perfect hiding and computational binding.

Second Theorem

- Similarly to the first main theorem, we obtain the following.
- Theorem: [Yamakami (2015)]
  1) There exists a scheme for non-interactive QBC for which the scheme is polynomial-time executable and has an explicit, direct construction from the ensembles $\{\rho^{(\pi)}_0(n),\rho^{(\pi)}_1(n)\}_{n,\pi}$.
  2) Moreover, if no quantum algorithm solves k-QSCD in polynomial time with non-negligible average advantage for a certain $k\geq 2$, then the scheme achieves computational hiding and statistical binding.

Open Problems

1. Construct much more efficient QBC schemes.
   - The proposed schemes use O(nlog(n)) qubits.
2. Find other applications of the quantum state ensemble.
   - Currently known applications are quantum public-key cryptosystems and quantum bit commitment.
3. Explore more interesting features of the quantum state ensemble.
   - We used only a few features of the ensemble. There might be more features to explore.

V. Quantum List Decoding

1. A New Encoding and List-Decoding Scheme
2. Quantumly Corrupted Words
3. Presence of Codewords
4. Quantum List-Decoding Problems
5. Phase Orthogonality

Classical Block Codes and Codewords

- We follow the general framework of Akavia, Goldwasser, and Safra (2003).
- A code (family) C consists of codewords of different lengths.
- An $(M(n),n)_{q(n)}$-code C is viewed as a function:

  $$C: \{0,\ldots,q-1\}^n x\{0,1\}^{\log(M(n))} \rightarrow \{0,\ldots,q-1\}.$$
- A codeword C_x of message x is a function defined

  $$C_x(\bullet) = C(x,\bullet): \{0,1\}^{\log(M(n))} \rightarrow \{0,\ldots,q-1\}.$$
- If the minimal (Hamming) distance d(n) of C is given, we call C an $(M(n),n,d(n))_{q(n)}$-code.

This is also known as the GL predicate.

- Example: the q-ary Hadamard Code $HAD^{(q)} = \{HAD^{(q)}_x\}_{x\in\{0,1\}^*}$.
  - ➢ $HAD^{(q)}_x(r) = x \bullet r \mod q$,

    where x and r are expressed in q-ary

    and ● denotes the (standard) inner product.

Classical Codes and Codewords

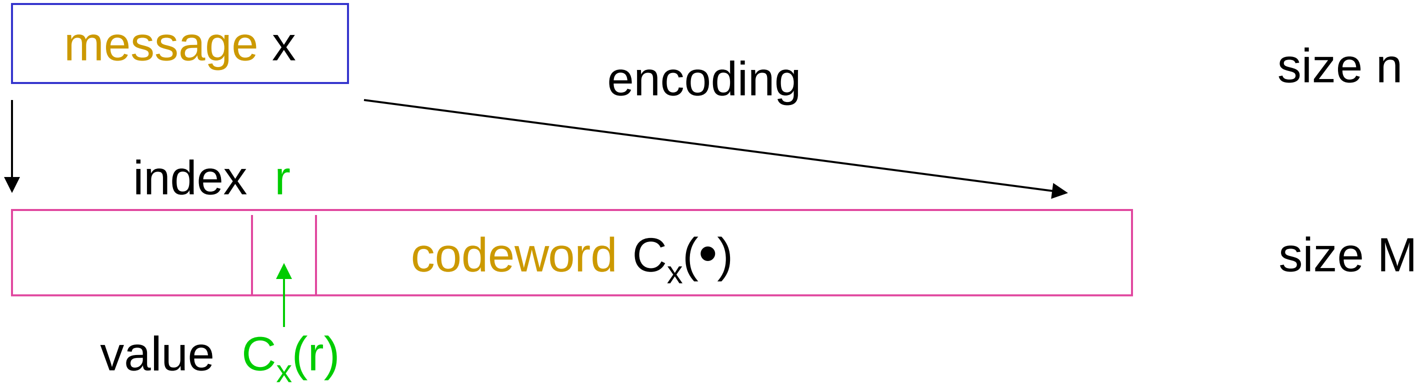
(*) Slightly different from a standard coding-theoretical formulation, here uses a complexity-theoretical formulation of codes and codewords.

- A code (family) C consists of codewords of different lengths.
- An $(M,n)_q$-code C is a function with two arguments:

$$C: \{0,\ldots,q-1\}^n x\{0,1\}^{\log(M)} \rightarrow \{0,\ldots,q-1\}.$$

- A codeword C_x of message x is a function defined from C by fixing x:

$$C_x(\bullet) = C(x,\bullet): \{0,1\}^{\log(M)} \rightarrow \{0,\ldots,q-1\}.$$



Example: Hadamard Codes

This is also known as the GL predicate.

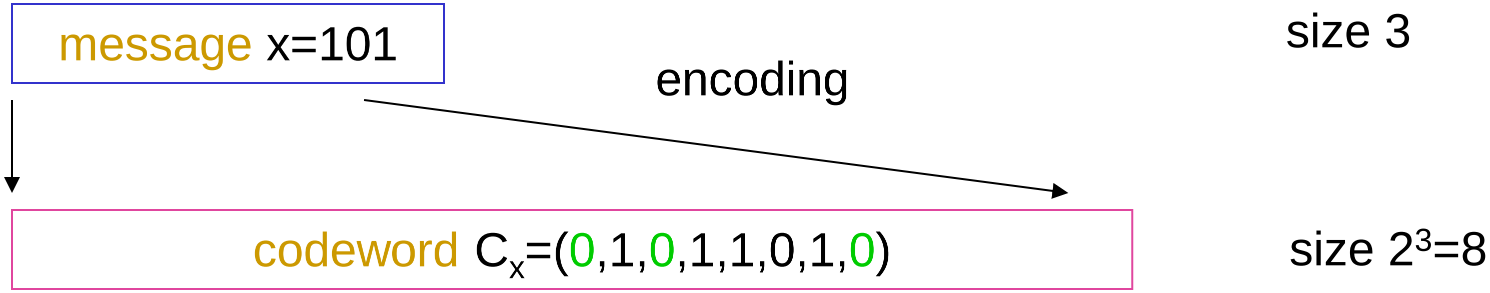
q = a prime number (for simplicity)

q-ary Hadamard code $HAD^{(q)} = \{HAD^{(q)}_x\}_{x \in \{0,1\}^*}$.

$HAD^{(q)}_x(r) = x \bullet r \pmod{q}$,

where x and r are expressed in q-ary and ● denotes the (standard) inner product.

Example: q=2 (binary)



message x=101

size 3

encoding

codeword $C_x=(0,1,0,1,1,0,1,0)$

size $2^3=8$

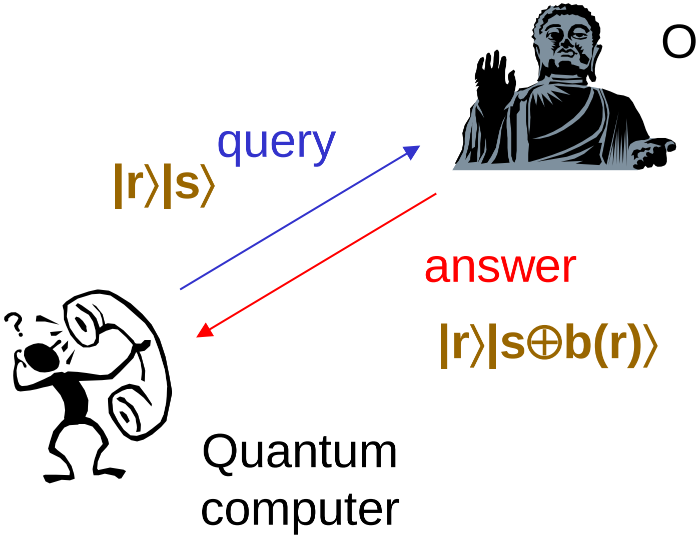
In other words, $C_x(000)=0$, $C_x(001)=1$, $C_x(010)=0$, …., $C_x(110)=1$, $C_x(111)=0$

How to Access Input Information (revisited)

Implicit Input is Given as an Oracle

Let b be any function from $\{0,1\}^n$ to $\{0,1\}^l$.

Oracle O_b is used to represent this function b.



Oracle: O_b

query

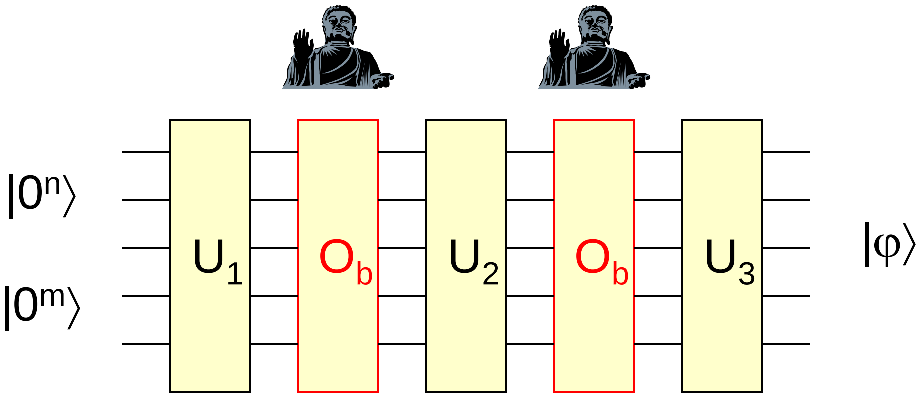
$|r\rangle|s\rangle$

answer

$|r\rangle|s\oplus b(r)\rangle$

Quantum computer

A computation proceeds as a chain of unitary operations and oracles.



$|0^n\rangle$

$|0^m\rangle$

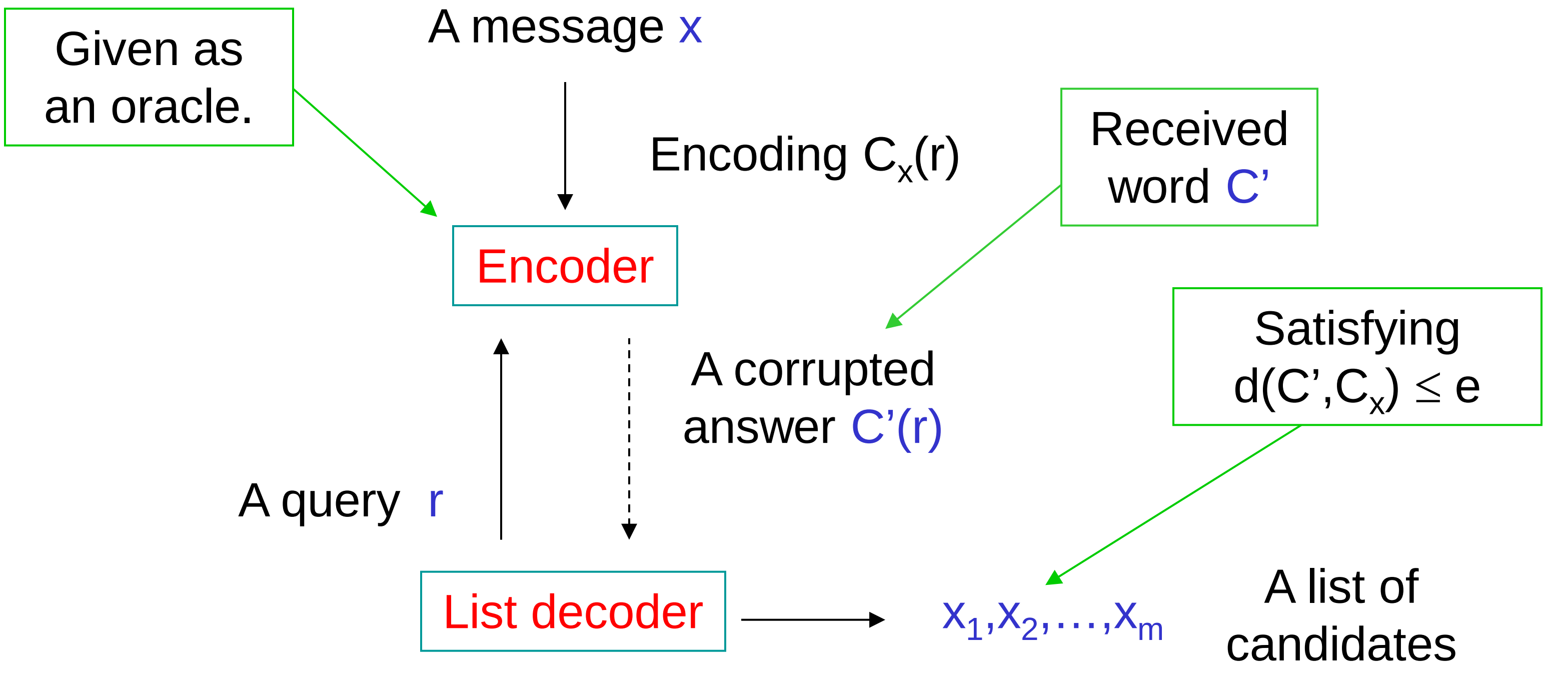
U_1 O_b U_2 O_b U_3

$|\varphi\rangle$

Instead of starting standard input x, the input information is given through oracle queries.

Classical Encoding and List-Decoding

- Here is a schematics of the standard (complexity-theoretical) setting of encoding and list-decoding of a code. Let e be an error bound.



Given as an oracle.

A message x

Encoding $C_x(r)$

Received word C'

Encoder

A corrupted answer $C'(r)$

Satisfying $d(C',C_x) \leq e$

A query r

List decoder

$x_1,x_2,\ldots,x_m$

A list of candidates

Various Decoding Problems

- Here are 5 methods of algorithmically decoding of classical codes. **e** denotes error bound and **r** is a received word.

1. Maximum Likelyhood Decoding (MLD)

   Given a distribution *D* on the error patterns, output a single codeword **c** that gives the maximal probability of obtaining **r**.

2. Nearest Codeword Problem (NCP)

   Output a single codeword **c** that is closest to **r** in distance.

3. List Decoding Problem (LDP)

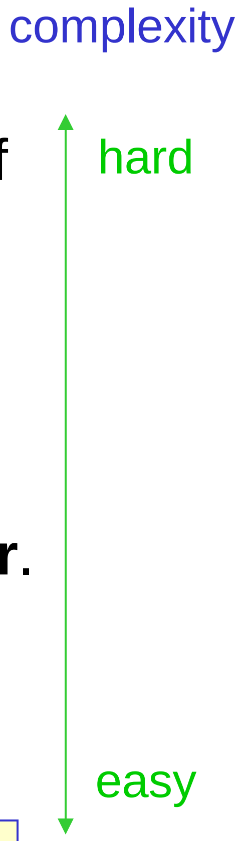
   Output the set of all codewords within distance **e** from **r**.

4. Bounded Distance Decoding (BDD)

   Output a single vector **c** within distance e from **r** if one exists or an empty set otherwise.

5. Unambiguous Decoding Problem (UDP)

   BDD with distance **e** set to (d(C)-1)/2.



Here, we focus on this problem.

What if an Encoder Produces Errors?

Introduction of Quantumly Corrupted Codewords

An imperfect encoder O produces a quantum state including erroneous terms.

Perfect Encoder

$$O|r\rangle|s\rangle = |r\rangle|s\oplus \underline{C_x(r)}\rangle$$

Correct term

Error term

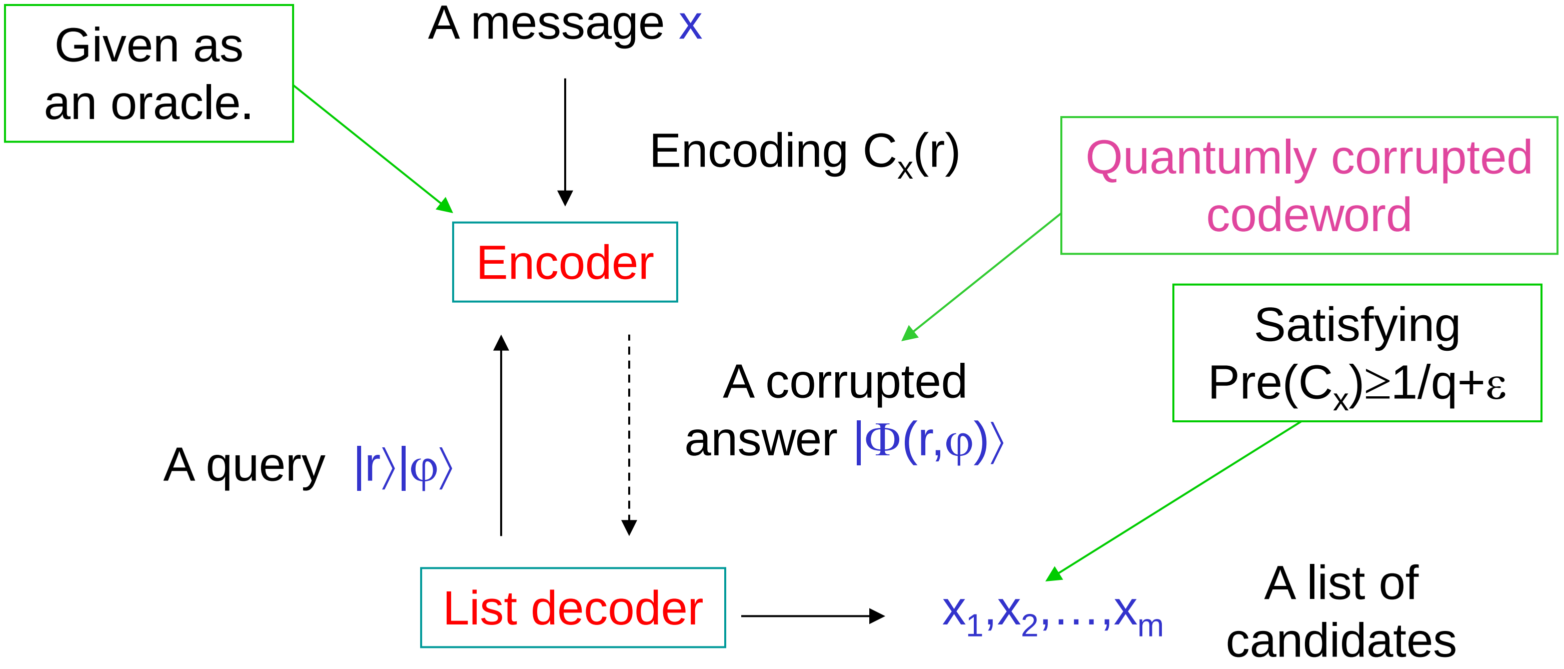
Imperfect Encoder

$$O|r\rangle|s\rangle|t\rangle = \alpha_{r,C(r)}|r\rangle|s\oplus \underline{C_x(r)}\rangle|t\oplus v_{r,C(r)}\rangle + \Sigma_{z\neq C(r)}\alpha_{r,z}|r\rangle|s\oplus \underline{z}\rangle|t\oplus v_{r,z}\rangle$$

For convenience, we call this O a quantumly corrupted codeword.

Message-Encoding and Quantum List-Decoding

- In our quantum setting, we consider the following scenario of encoding and list-decoding of a classical code.



Given as an oracle.

A message x

Encoding $C_x(r)$

Quantumly corrupted codeword

Encoder

Satisfying $Pre(C_x) \geq 1/q+\varepsilon$

A corrupted answer $|\Phi(r,\varphi)\rangle$

A query $|r\rangle|\varphi\rangle$

List decoder

$x_1,x_2,\ldots,x_m$

A list of candidates

Presence of Codewords

- We introduce the notion of presence of a codeword.
- First, recall a quantumly corrupted codeword O:

Correct term

$$O|r\rangle|s\rangle|t\rangle = \alpha_{r,C(r)}|r\rangle|s\oplus\underline{C_x(r)}\rangle|t\oplus v_{r,C(r)}\ \rangle$$

Error term

$$+ \Sigma_{z\neq C(r)}\alpha_{r,z}|r\rangle|s\oplus\underline{z}\rangle|t\oplus v_{r,z}\ \rangle$$

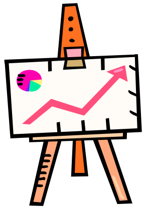
- The average success probability of receiving C_x is $(1/M)\Sigma_{r=1}^{M}|\alpha_{r,C(r)}|^2$.
- We call this value the presence of C_x in O and denote it by

$$Pre_O(C_x) = (1/M)\Sigma_{r=1}^{M}|\alpha_{r,C(r)}|^2.$$

- In classical decoding, the error rate e is expressed by our presence notion as follows:

$$Pre_O(C_x) = (1/M)(M - d(C_x,O)) = (1/M)(M - eM) = 1 – e.$$

Quantum Johnson Bounds



- How many message candidates are there?
- In classical list-decoding, Johnson bound gives an upper bound of the number of message candidates within distance e.
- Here, we give a quantum version of Johnson bound.
- Let $l(n) = (1-1/q(n))[1-d(n)/M(n)(1+1/(q(n)-1))]^{1/2}$.
- Theorem: [Kawachi-Yamakami (2010)]

  For any $(M(n),nd(n))q(n)$-code C and quantumly corrupted codeword O, it holds the following.

  1. If $\varepsilon(n) > l(n)$, then there are at most J(n) messages $x \in \Gamma_n$ such that $Pre_O(C_x) \geq 1/q(n) + \varepsilon(n)$, where $Q(n) = 1-1/q(n)$ and

     $J(n) = \min\{M(n)(q(n)-1), [d(n)Q(n)]/[d(n)Q(n)+M(n)\varepsilon(n)^2-M(n)Q(n)^2]\}$.

  2. If $\varepsilon(n) = l(n)$, then there are at most J(n) messages $x \in \Gamma_n$ such that $Pre_O(C_x) \geq 1/q(n) + \varepsilon(n)$, where

     $J(n) = 2M(n)(q(n)-1) -1$

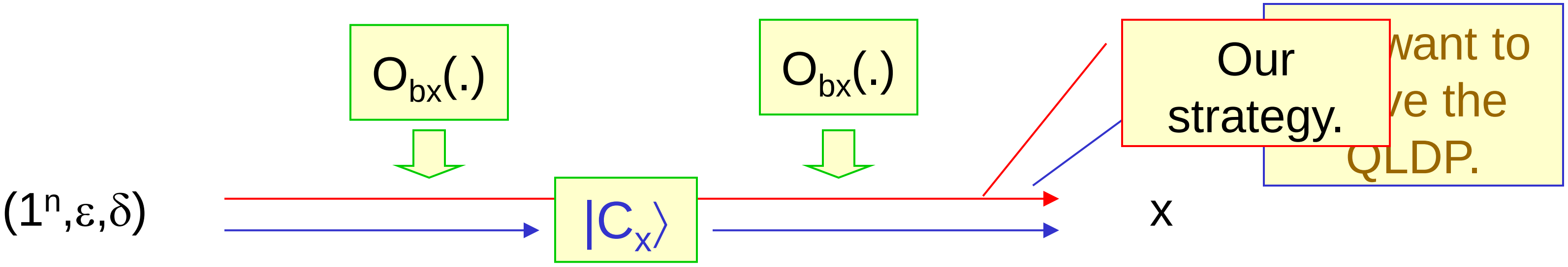
Quantum List-Decoding Problems (QLDPs)

- We formally define a quantum list-decoding problem for code C.
- Let C be any $(M,n,d)_q$-code consisting of codewords C_x with hidden messages x.
- ε-Quantum List Decoding Problem (QLDP) for C
  - Input: two parameters, n and 1/ε
  - Implicit Input: a quantumly corrupted codeword O
  - Output: a list of messages including all x's s.t. $\mathrm{Pre}_O(C_x) \geq 1/q + \varepsilon$.
- Now, our task is to solve this QLDP for code C with high probability with access to a quantumly corrupted codeword O.

How to Solve the QLDP

Introduction of Quantum Codeword States

- To solve the QLDP, we introduce a new notion of quantum codeword states, which are useful to deal with erroneous computation.
- For simplicity, we consider only the following types of quantum codes. Let $\omega_L = e^{2\pi i/L}$.
  - For each message $x \in \{0,1\}^n$, a quantum codeword state of x is a quantum state $|C_x\rangle = (1/\sqrt{M})\sum_r \omega_L^{C(x,r)}|r\rangle$, where $r \in \{0,1\}^{m(n)}$, $M=2^{m(n)}$, and $L=2^{l(n)}$. (We can further generalize this notion!)



Generating a quantum codeword state.

Robust Quantum Computation

- We can prove the following useful theorem.
- Theorem: [Kawachi-Yamakami (2010)]

  If we can decode quantum codeword state $|C_x\rangle$ to x with high success probability, then we can solve the QLDP for C_x with noticeable probability.
- This theorem follows from the next lemma on a robust generation of a quantum
- Key Lemma: [Kawachi-Y

  There is an efficient quantum algorithm that can generate the quantum codeword state $|C_x\rangle$ with access to a quantumly corrupted codeword O_{Cx} for C_x.

A real function $\varepsilon(n)$ is noticeable if $\varepsilon(n) \geq 1/p(n)$ for a certain polynomial p and for almost all positive integers n.

Three Quantum List-Decodable Codes

- Using our theorem, we can prove that the following three codes are quantum list decodable.

1. q-ary Hadamard Code (for fixed prime q)
   - ➢ $HAD_x^{(q)}(r) = \sum_{i=1}^{|r|-1} x_i r_i$
2. Shifted Legendre Symbol Code (for fixed prime p)
   - ➢ $SLS_x^{(p)}(r) = 1$ if x+r mod p is not a quadratic residue for p.
   - ➢ $SLS_x^{(p)}(r) = 0$ otherwise.

This q is a quadratic residue (mod p) iff $\exists x$ s.t. $x^2 \equiv q \pmod{p}$.)

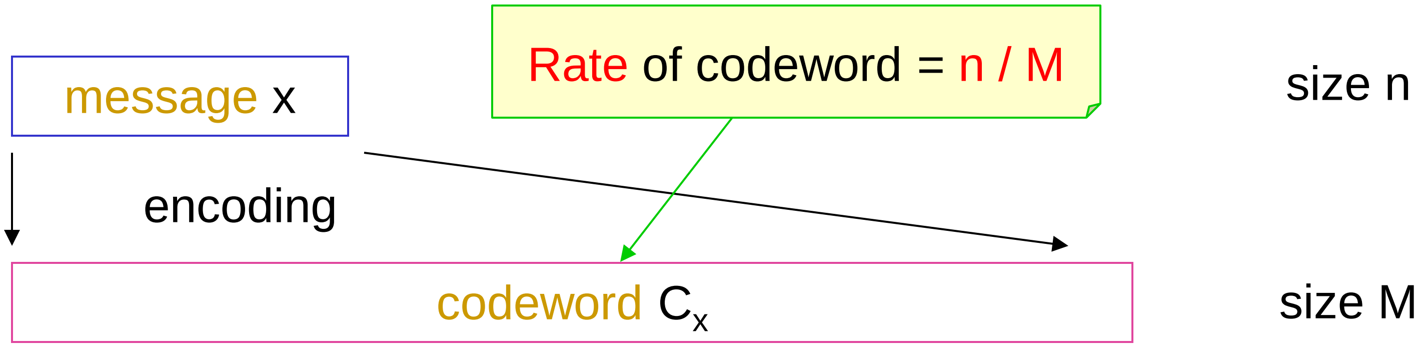
3. Pairwise Equality Code
   - ➢ $PEQ_x(r) = \oplus_{i=0}^{n/2} EQ(x_{2i}x_{2i+1}, r_{2i}r_{2i+1})$, where EQ is the equality predicate.

VI. Complexity of Codes

1. Polynomially Small Rate
2. Guruswami-Sudan Polynomial Interpolation
3. Concatenated Codes
4. Direct Consequences
5. Application to Quantum Search Problems
6. How to Use Quantum List-Decoders

Codes with Polynomially Small Rate

The rate of a codeword is a ratio between message length and codeword length.



Rate of codeword = n / M

message x

size n

encoding

codeword C_x

size M

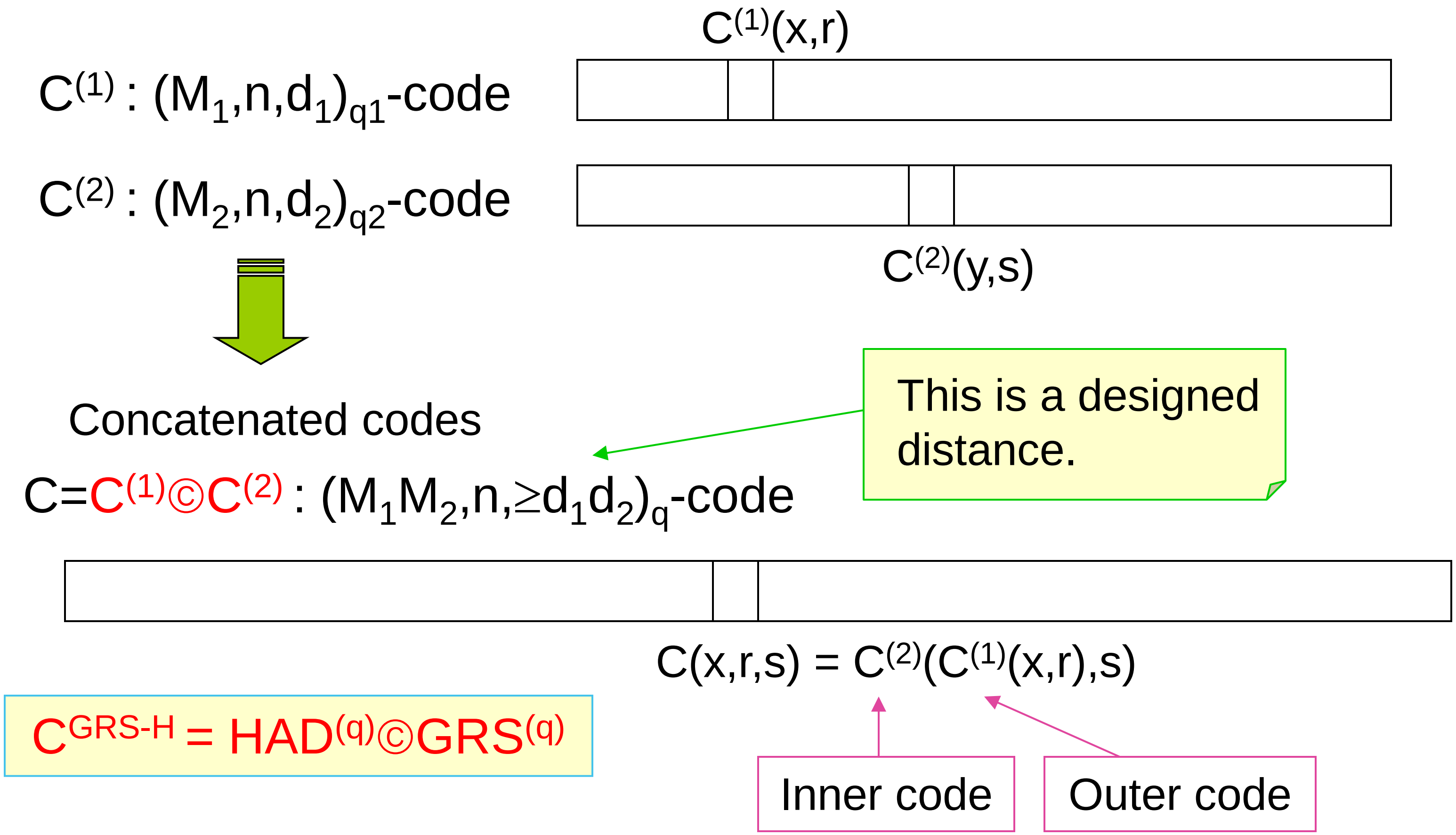
If M=poly(n), then the rate is 1/poly(n), polynomially small in n.

Remark: All known quantum list-decodable codes have exponentially small rates.

Question: Is there any quantum list-decodable code with polynomially small rate?

Concatenated Codes $C^{GRS\text{-}H}$

- We introduce $C^{GRS\text{-}H}$ by concatenating Hadamard Codes and Generalized Reed-Solomon Codes



$C^{(1)}(x,r)$

$C^{(1)}$: $(M_1,n,d_1)_{q1}$-code

$C^{(2)}$: $(M_2,n,d_2)_{q2}$-code

$C^{(2)}(y,s)$

Concatenated codes

$C=C^{(1)} © C^{(2)}$: $(M_1M_2,n,\geq d_1d_2)_q$-code

This is a designed distance.

$C(x,r,s) = C^{(2)}(C^{(1)}(x,r),s)$

$C^{GRS\text{-}H} = HAD^{(q)} © GRS^{(q)}$

Inner code

Outer code

A Key Lemma

quantum reduction between quantumly corrupted codewords

- Lemma: [Yamakami (2016)]

  Let D = HAD©C and let O_D be any quantumly corrupted codeword for D. There exists a polynomial-time quantum algorithm B and a quantumly corrupted codeword O_C for C such that

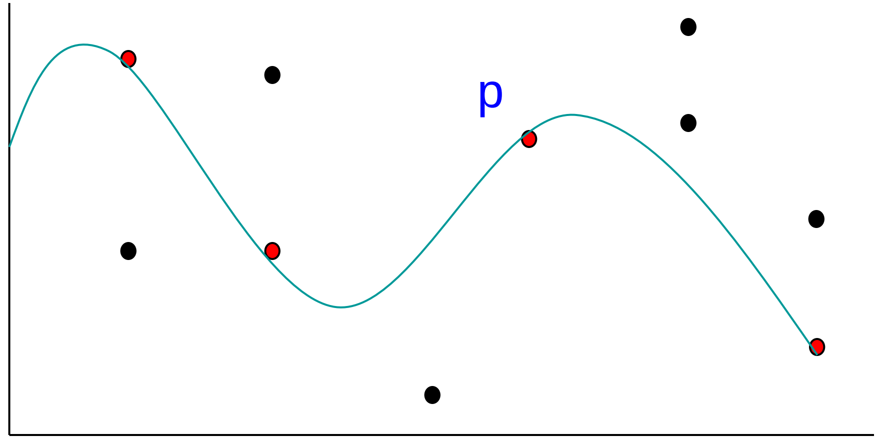
  1) If $Pre_{OD}(D_x) \geq 1/q+\varepsilon$, then $Pre_{OC}(C_x) \geq 1/q^m+\varepsilon^3q^2/(q-1)^3-1/q^{2m}$.
  2) B realizes O_C with access to O_D as an oracle.

- Corollary: [Yamakami (2016)]

  If GRS is quantumly list decodable, then $C^{GRS\text{-}H}$ is also quantumly list decodable.

Polynomial Reconstruction Problem

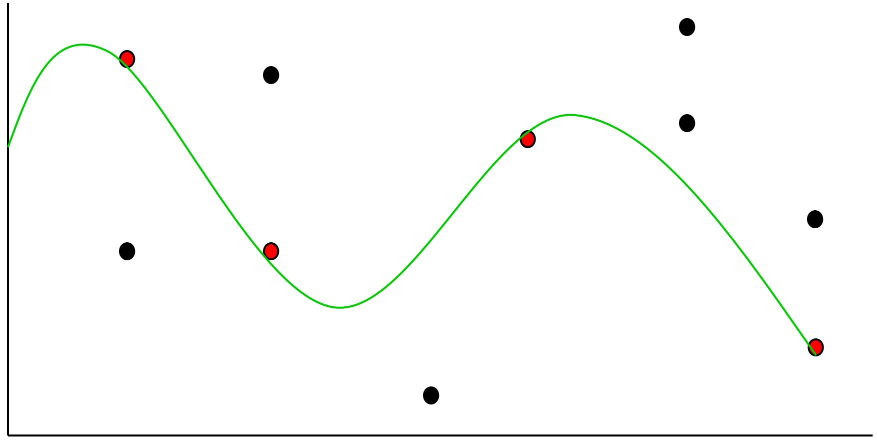
- Polynomial Reconstruction Problem
  - instance: 3 integers m’,n’,t>0, m’ points $\{(x_i,y_i)\}_{i\in[m']}$ $\subseteq[q]\times[q]$
  - output: all univariate polynomials p of degree ≤n’ that lie on at most t points, provided that t $\geq\sqrt{m'n'}$



Guruswami-Sudan Polynomial Interpolation

- Theorem: [Guruswami-Sudan (1999)]

  There exists a classical algorithm that solves the polynomial reconstruction problem in time polynomial in (m,log(q)).



A quantum algorithm for GRS:

1. Query all points;
2. Observe their oracle answers;
3. Apply the GS algorithm.

Direct Consequences

- Relatively large bias case
- Theorem: [Yamakami (2016)]

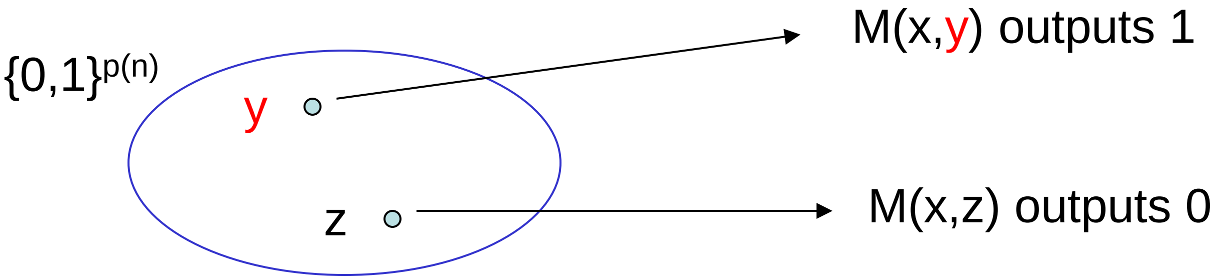
  There exists a polynomial-time quantum algorithm that solves the QLDP for C^{GRS-H} when its bias is only polynomially small.

- Arbitrary small bias case
- Theorem: [Yamakami (2016)]

  If there is a polynomial-time quantum algorithm for the QLDP for C^{GRS-H} for arbitrary bias, then NP can be solved on quantum computers in polynomial time.

Application to Quantum Search Problems

- We apply our quantum list-decoding to complexity theory.
- L is in QCMA ⇔ for any x,
  - If $x \in L$, then $\exists y \in \{0,1\}^{p(n)}$ s.t. $M(x,y)$ outputs 1 with prob. $\geq 2/3$, and
  - If $x \notin L$, then $\forall y \in \{0,1\}^{p(n)}$, $M(x,y)$ outputs 1 with prob. $\leq 1/3$.



- A solution function f for (L,M) ⇔
  - $f(x) \in \{0,1\}^{p(n)} \cup \{\perp\}$,
  - If $x \in L$, then $M(x,f(x))$ outputs 1 with prob $\geq 2/3$, and
  - If $x \notin L$, then $f(x)=\perp$.

How to Use Quantum List-Decoders

- Theorem: [Yamakami (2016)]

  Assume that QCMA $\neq$ BQP. Let p,p' be any polynomials with p'(n)>p(n) for all n. There exists a QCMA search problem such that, for any solution function f, no polynomial-time quantum algorithm finds y, on each input x of length n, the relative distance Δ(y,f(x)) is at most 1/2-1/p(n) with probability at least 1-2p(n)/(p'(n)(p(n)+2)).

❑ Proof Strategy:

1. Encode a solution into $C^{GRS\text{-}H}$.
2. Quantum list decode a quantumly corrupted codeword for $C^{GRS\text{-}H}$.
3. Check if candidates are truly solutions.

Open Problems

- Challenging Reed-Solomon Codes
  1. Find a truly “quantum” list-decoding algorithm for GRS codes.
  2. Find its non-trivial relationships to other known problems.
- Developing a Theory of Quantum List-Decoding
  1. Find quantum algorithms for popular codes, such as algebraic-geometric codes.
  2. Cultivate the foundations of this theory.
  3. Show tight bounds of presence.
  4. Find useful applications to quantum complexity theory.

Open Problems

- We formulate the notion of quantum codes and quantum codewords for erroneous communication.
- For simplicity, we consider only the following types of quantum codes. Let $\omega_L = e^{2\pi i/L}$.

1. For each word $x \in \{0,1\}^n$, a quantum codeword (qucodeword) of x is a pure quantum state $|C_x\rangle = (1/\sqrt{M}) \times \sum_r \omega_L^{C(x,r)}|r\rangle$, where $r \in \{0,1\}^{m(n)}$, $M=2^{m(n)}$, and $L=2^{l(n)}$.
2. A quantum code (qucode) C^Q is a series $\{|C_x\rangle\}_{x\in\{0,1\}^*}$ of qucodewords.

Challenges:

- Show robustness of code generation through noisy channels.
- Cultivate a general framework for decoding quantum codes.
- Find useful applications in error correction and cryptography.

VI. Quantum Hardcore Functions

1. Constructing Quantum Hardcore Functions
2. How to Obtain Quantum Hardcores
3. New Quantum Hardcore Functions

Constructing Quantum Hardcore Functions for any Quantum One-Way Function

- Consider a quantum hardcore function $P(x,r)$ for any quantum one-way function (of the form $f'(x,r)=(f(x),r)$).
- Such a quantum hardcore function actually exists!
- Adcock and Cleve (2002) showed that the inner-product-mod-2 function $GL(x,r) = x \bullet r \bmod 2$ is a quantum hardcore predicate for any quantum one-way function.

Are there any other quantum hardcore functions?

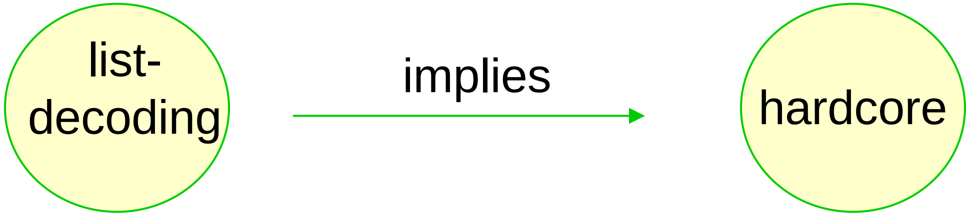
YES

First, we need to explore a close relationship between quantum hardcores and quantum list-decoding of classical block codes.

How to Obtain Quantum Hardcores

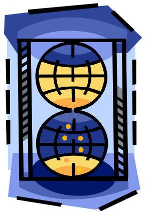
Quantum list-decoding implies quantum hardcores

- Let C(x,r) be any function.
- Assumption: Assume that there is an efficient quantum algorithm that quantumly list-decodes code C with noticeable probability for any x and r.
- Consequence: This function C is indeed a quantum hardcore function for the function f' induced by f'(x,r)=(f(x),r) for any quantum one-way function f.



Showing quantum list-decodability of code C.

Proving C to be a quantum hardcore for any QOWF.



New Quantum Hardcore Functions

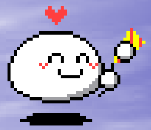
- Using our theorem, since the following three codes are quantum list decodable, they are also quantum hardcore predicates for any quantum one-way function.

1. q-ary Hadamard Code (for fixed prime q)
   - ➢ $HAD_x^{(q)}(r) = \sum_{i=1}^{|r|-1} x_i r_i$
2. Shifted Legendre Symbol Code (for fixed prime p)
   - ➢ $SLS_x^{(p)}(r) = 1$ if x+r mod p is not a quadratic residue for p.
   - ➢ $SLS_x^{(p)}(r) = 0$ otherwise.
3. Pairwise Equality Code
   - ➢ $PEQ_x(r) = \oplus_{i=0}^{n/2} EQ(x_{2i}x_{2i+1}, r_{2i}r_{2i+1})$, where EQ is the equality predicate.

- The last two predicates have not been known as classical hardcores.

Open Problems

- Find more natural quantum hardcore functions.
- Find useful applications of quantum hardcore functions.



Thank you for listening

Q & A

I’m happy to take your question!

END

Key Operations

- We define three important quantum operations.

- $P_1{}^*$ transforms $|0\rangle|\pi\rangle|id\rangle|id\rangle$ to $|0\rangle|\pi\rangle|\Phi^{(\pi)}{}_0\rangle$.
- P_2 transforms $|\sigma\rangle|\phi^{(\pi)}{}_{\sigma b}\rangle$ to $|\sigma\rangle|\phi^{(\pi)}{}_{\sigma b-1}\rangle$

  without knowing (b,π).
- P_{SPA} partitions χ to $\chi_0 \oplus \chi_1$ s.t. $\chi = \chi_0 \oplus \chi_1$,

  where $\chi_b = \Sigma_{\sigma\in Sn}\, p_{\sigma b}\, |\phi^{(\pi)}{}_{\sigma b}\rangle\langle\phi^{(\pi)}{}_{\sigma b}|$ $(b\in\{0,1\})$.

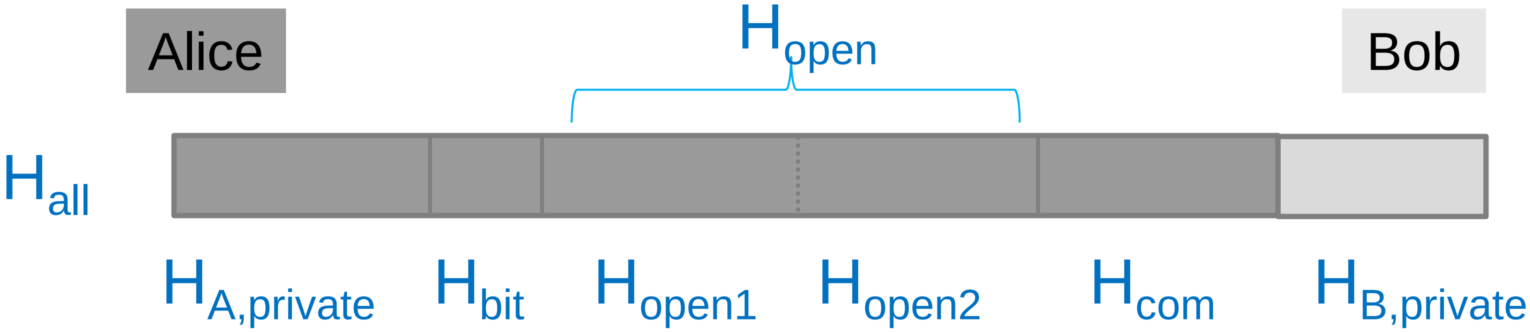
Committing Phase Protocol A_{com}

(C1) Alice starts with $|0\rangle$ in H_{all}. Choose a secret key $\pi \in K_n$ uniformly at random from H_{open2}.

(C2) She starts with $|id\rangle|id\rangle$ in $H_{open1} \otimes H_{com}$. Generate $(1/|S_n|)\Sigma_{\sigma \in Sn}|\sigma\rangle$ from $|id\rangle$. Create $|\Phi^{(\pi)}{}_0\rangle$ in H_{open}.

(C3) She chooses a committed bit $a \in \{0,1\}$ in H_{bit}. Transform $|a\rangle|\Phi^{(\pi)}{}_0\rangle$ in $H_{open1} \otimes H_{com}$ to $|a\rangle|\Phi^{(\pi)}{}_a\rangle$.

(C4) She sends a subsystem H_{com} to Bob. He receives a reduced quantum state $\chi = \rho^{(\pi)}{}_a$.



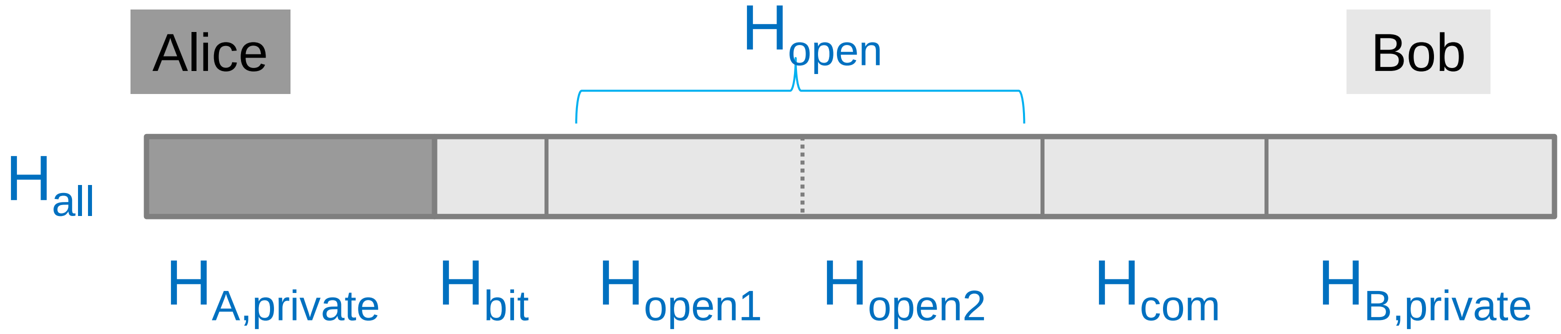
Opening Phase Protocol A_{open} I

Assume that Bob received χ in H_{com} in the previous phase.

(R1) Alice sends $H_{bit} \otimes H_{open}$ to Bob.

(R2) $H_{bit} \otimes H_{open2}$ contains (a,π) in superposition. If $\pi \notin K_n$, then Bob knows Alice has cheated.

(R3) Bob applies P_{SPA} to $|0\rangle\langle 0| \otimes \chi$ in $H_{B,private} \otimes H_{com}$.

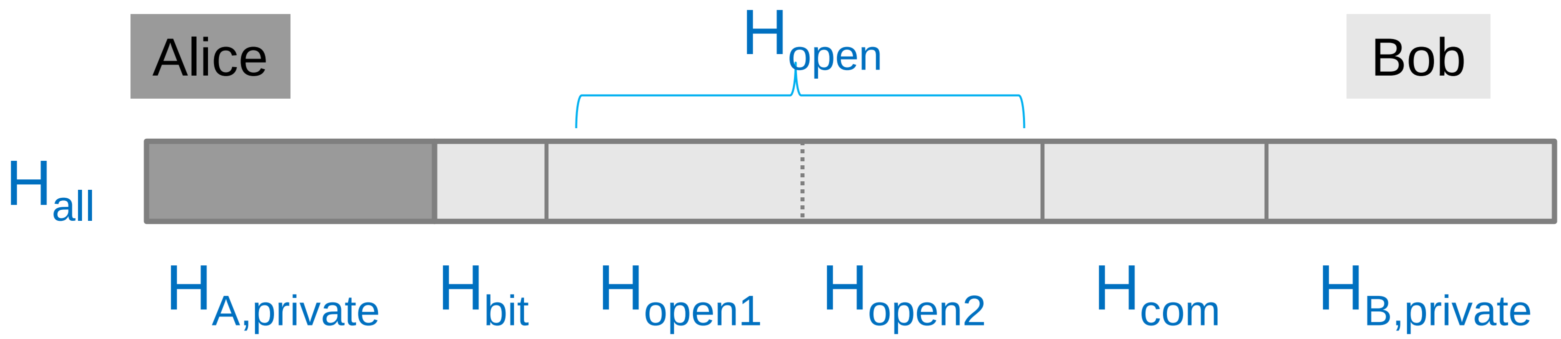


Opening Phase Protocol A_{open} II

(R3) Bob applies P_{SPA} to $|0\rangle\langle 0|\otimes\chi$ in $H_{B,private}\otimes H_{com}$.

(R4) Bob measures $H_{B,private}$. If the obtained bit does not match a in H_{bit}, Alice has cheated. Assume otherwise.

(R5) If a=1, Bob changes $|\Phi^{(\pi)}{}_1\rangle$ to $|\Phi^{(\pi)}{}_0\rangle$. Bob applies $P_1{}^{*-1}$ and observes H_{bit} to obtain a. Bob measures $H_{open1}\otimes H_{com}$ in state $|0\rangle|id\rangle$. If (0,id) is observed, Bob accepts a as Alice's committed bit. Otherwise, Bob knows Alice has cheated.



Alice
H_{open}
Bob
H_{all}
$H_{A,private}$
H_{bit}
H_{open1}
H_{open2}
H_{com}
$H_{B,private}$