QUANTUM BOOTCAMP

JANUARY 2020

YFKE DULEK

AUTHENTICATING QUANTUM STATES

QUANTUM AUTHENTICATION CODES (QAC)

- prevent quantum information from being altered (while sending or storing)

- One-time: secret key k usable only once
- If decoding rejects, information may be lost!

QAC AS A TOOL

- Authentication of a state: preventing that state from being altered **at all**
- Verification of a computation: preventing the input from being altered, except **in one specific way**
- QACs can be an ingredient of (cryptographic) verification protocols!

OVERVIEW

- PART I: quantum authentication codes (**the tool**)
  - Definition
  - Two different codes
  - Relation to encryption
- PART II: verifiable computation (**the applications**)
  - Scenario: client and server
  - Scenario: multi-party computation

PART I: QUANTUM AUTHENTICATION CODES

DEFINITION [BCG+02]

- **Correctness**: if no attack happens, decryption "accepts", and the original message is always recovered:

$$Dec_k \circ Enc_k = \mathrm{Id}$$

- **Security** (first attempt): if decryption accepts, the recovered message is close to the original message:

$\forall \Phi_{\text{attack}} \quad \exists a \in [0,1] \quad \forall \rho :$

$$\underbrace{\mathbb{E}_k (Dec_k \circ \Phi_{\text{attack}} \circ Enc_k)(\rho)}_{\text{"REAL"}} \;\approx\; \underbrace{a \cdot \rho + (1-a) \cdot |\mathrm{rej}\rangle\langle \mathrm{rej}|}_{\text{"IDEAL"}}$$

[BCG+02] Barnum, Crépeau, Gottesman, Smith, Tapp; FOCS 2002

DEFINITION

- **Security** (first attempt): if decryption accepts, the recovered message is close to the original message:
  $\forall \Phi_{\text{attack}} \quad \exists a \in [0,1] \quad \forall \rho :$

$$\mathbb{E}_k (Dec_k \circ \Phi_{\text{attack}} \circ Enc_k)(\rho) \;\approx\; a \cdot \rho + (1-a) \cdot |\text{rej}\rangle\langle\text{rej}|$$

- Security (second attempt): with **side information** [DNS12]:
  $\forall \Phi_{\text{attack}} \quad \exists \Phi_{\text{acc}}, \Phi_{\text{rej}} \quad \forall \rho :$

$$\mathbb{E}_k ((Dec_k \otimes \mathsf{Id}_S) \circ \Phi_{\text{attack}} \circ (Enc_k \otimes \mathsf{Id}_S))(\rho_{MS}) \approx (\mathsf{Id}_M \otimes \Phi_{\text{acc}})(\rho_{MS}) + |\text{rej}\rangle\langle\text{rej}| \circ \Phi_{\text{rej}}(\rho_S)$$

[DNS12] Dupuis, Nielsen, Salvail; CRYPTO 2012

EXAMPLE: CLIFFORD CODE [BCG+02]

- Key: $C \in_R \text{Clifford}_{n+1}$
- Encoding: $|\psi\rangle \mapsto C(|\psi\rangle \otimes \underbrace{|0^n\rangle}_{\text{“TRAPS”}})$
- Decoding: apply $C^\dagger$, measure traps

THEOREM: The Clifford code is a secure QAC.

(the probability to alter the state undetected is $\leq 2^{-n}$.)

[BCG+02] Barnum, Crépeau, Gottesman, Smith, Tapp; FOCS 2002

EXAMPLE: TRAP CODE [BGS13]

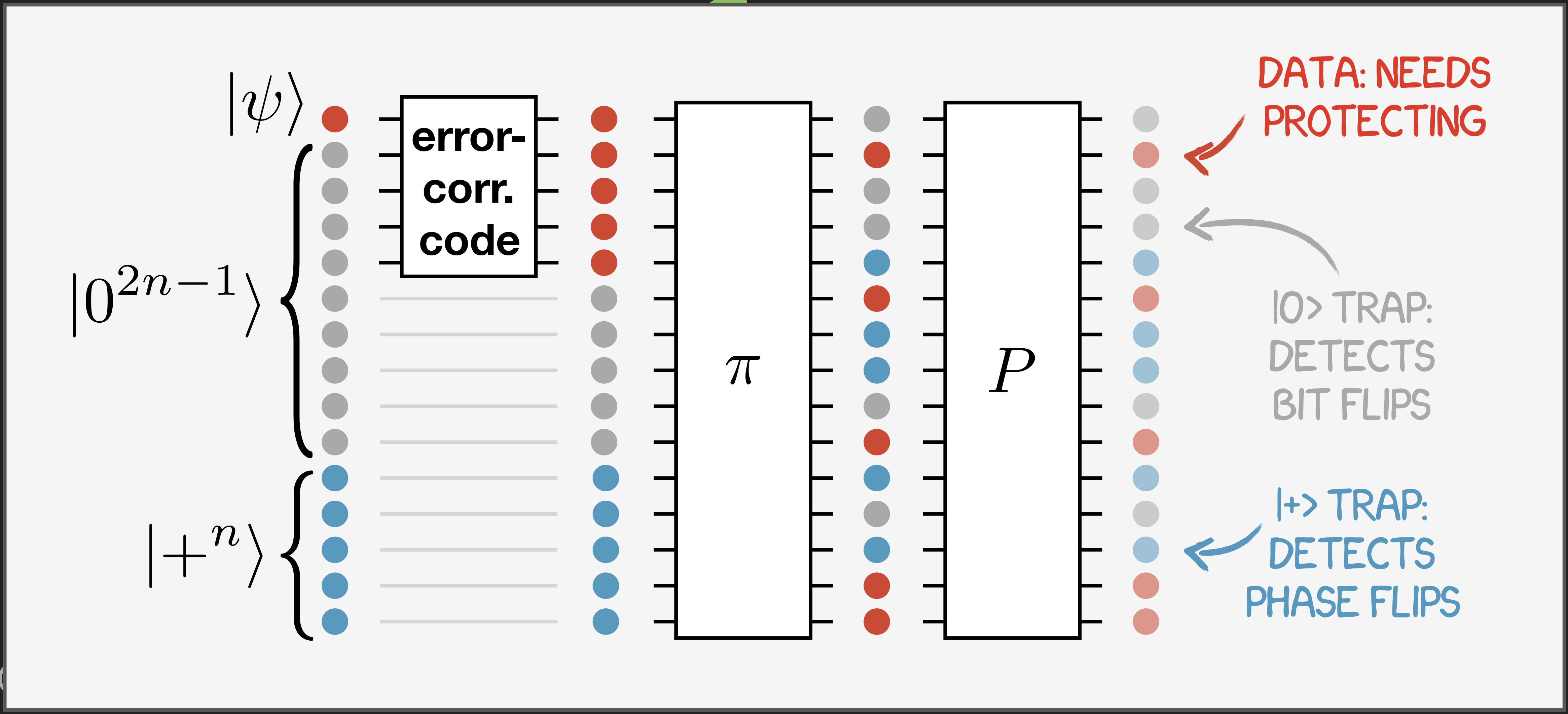
- Key: $P \in_R \mathrm{Pauli}_{3n}\,, \quad \pi \in_R S_{3n}$
- Encoding: $|\psi\rangle \mapsto P(\pi(\mathsf{ECC}|\psi\rangle \otimes |0^n\rangle \otimes |+^n\rangle))$

$[n,\, 1,\, d]$ ERROR-CORRECTING CODE

[BGS13] Broadbent, Gus, Stebila; CRYPTO 2013

EXAMPLE: TRAP CODE [BGS13]

- Key: $P \in_R \text{Pauli}_{3n}, \quad \pi \in_R S_{3n}$
- Encoding: $|\psi\rangle \mapsto P(\pi(\mathsf{ECC}|\psi\rangle \otimes |0^n\rangle \otimes |+^n\rangle))$



[B

EXAMPLE: TRAP CODE [BGS13]

- Key: $P \in_R \mathrm{Pauli}_{3n}, \quad \pi \in_R S_{3n}$
- Encoding: $|\psi\rangle \mapsto P(\pi(\mathsf{ECC}|\psi\rangle \otimes |0^n\rangle \otimes |+^n\rangle))$
- Decoding: apply $\pi^{-1}P^\dagger$, measure traps & ECC syndrome

$[n, 1, d]$ ERROR-CORRECTING CODE

THEOREM: The trap code is a secure QAC.

(the probability to alter the state undetected is $\leq (2/3)^{d/2}$.)

[BGS13] Broadbent, Gus, Stebila; CRYPTO 2013

WHICH CODE IS "BETTER"?

- Clifford code: simple, clean analysis. Strong security.
- Trap code: more structure, encoding/decoding requires "less quantum"
- There are more: polynomial code [BCG+06], Auth-QFT-Auth [GYZ17], strong trap code [**D**S18], …

[BCG+06] Ben-Or, Crépeau, Gottesman, Hassidim, Smith; FOCS 2006.
[GYZ17] Garg, Yuen, Zhandry; CRYPTO 2017.
[DS18] Dulek, Speelman; TQC 2018.

THE "LANDSCAPE" OF DEFINITIONS

- Stronger: key recycling [HLM16, GYZ17] (if decoding accepts, the key can be reused)
  - ✓ Clifford code ✗ Trap code
- Stronger: ciphertext authentication [AGM16]
  - ✓ Clifford code ✓ Trap code
- **THEOREM.** Q authentication implies encryption. [BCG+02]

[HLM16] Hayden, Leung, Meyers; arXiv:1610.09434.
[GYZ17] Garg, Yuen, Zhandry; CRYPTO 2017.
[AGM16] Alagic, Gagliardoni, Majenz; Eurocrypt 2016.
[BCG+02] Barnum, Crépeau, Gottesman, Smith, Tapp; FOCS 2002.

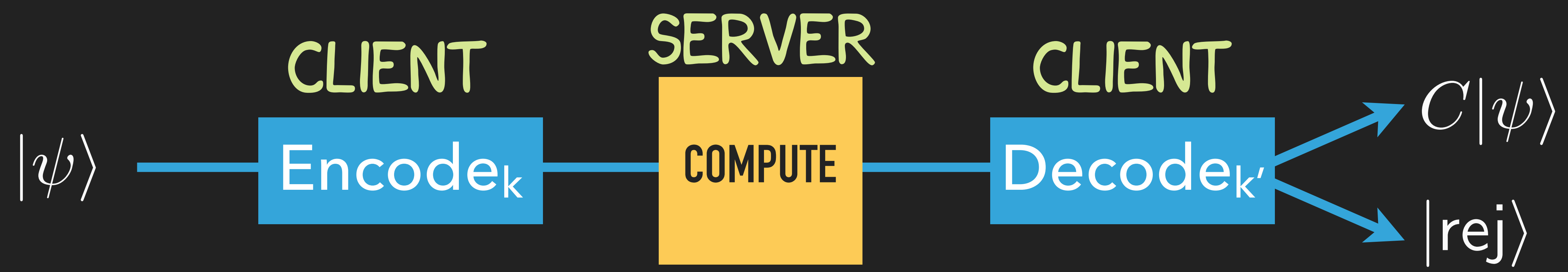
OPEN PROBLEMS

- For known codes: if decoding rejects, how much of the key is compromised? [GYZ17]
- More generally: design of “many-time” codes? [AGM16]
- How to deal with noise?
- Minimal quantum capabilities of the encoder/decoder? Can a classical client “outsource” encoding? [GV19]

[GYZ17] Garg, Yuen, Zhandry; CRYPTO 2017.
[AGM16] Alagic, Gagliardoni, Majenz; Eurocrypt 2016.
[GV19] Gheorghiu, Vidick; FOCS 2019.

PART II: VERIFIABLE COMPUTATION

SCENARIO: CLIENT AND SERVER

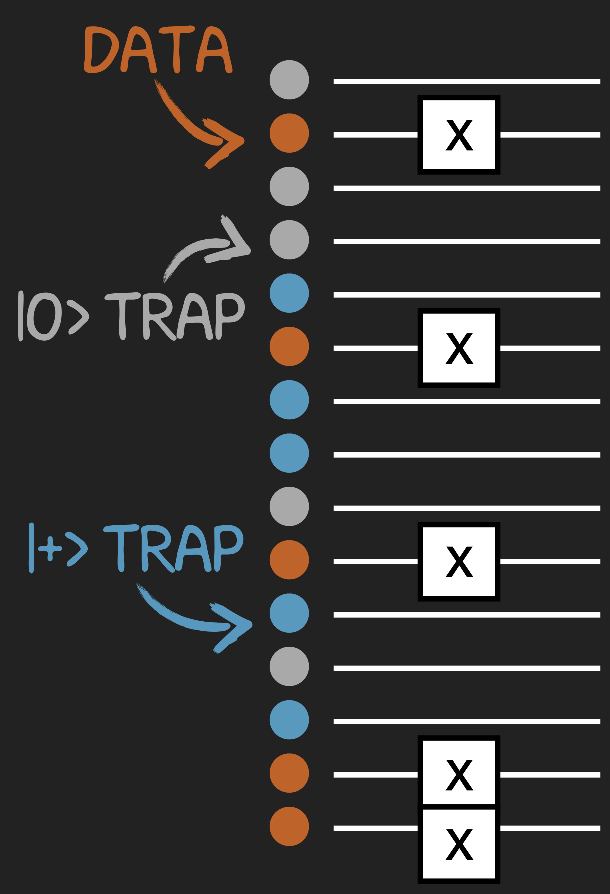


- Server does not know the key
- Client "updates" the key during/after computation
- Protocol using trap code [BGS13]: some gates are simple, some require "magic states"

[BGS13] Broadbent, Gus, Stebila; CRYPTO 2013

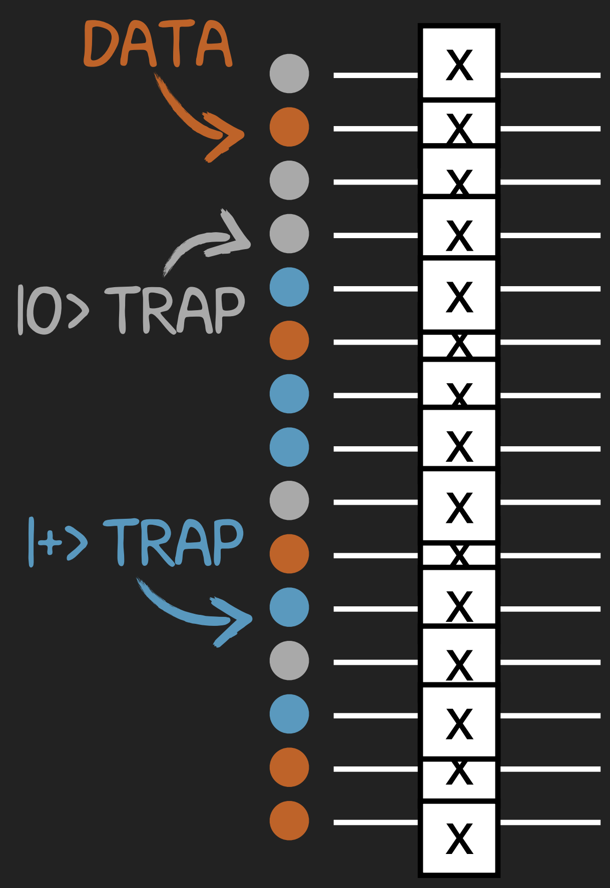
PROTOCOL: CLIENT AND SERVER

- Idea: transversal gate application



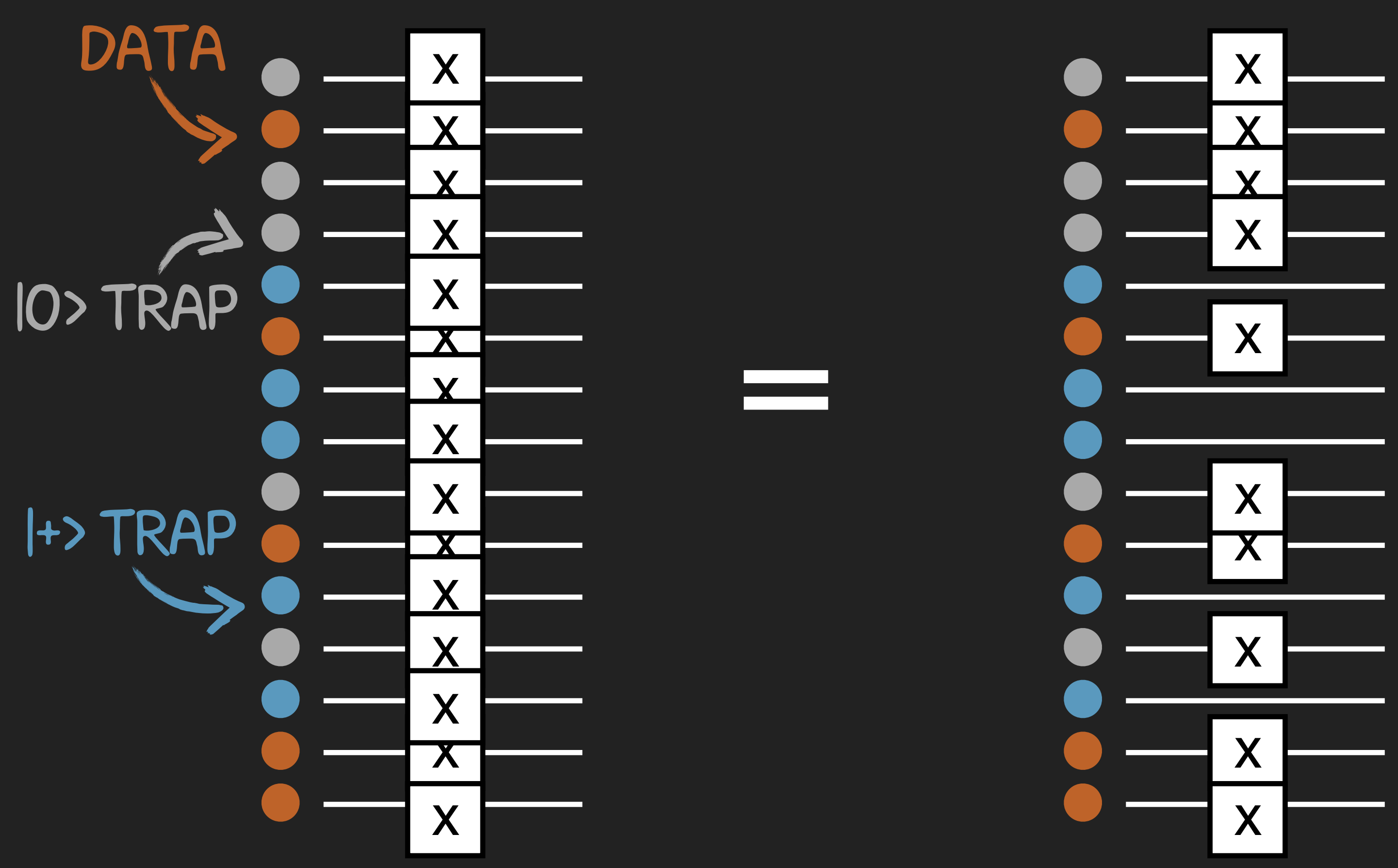
PROTOCOL: CLIENT AND SERVER

- Idea: transversal gate application



PROTOCOL: CLIENT AND SERVER

- Idea: transversal gate application

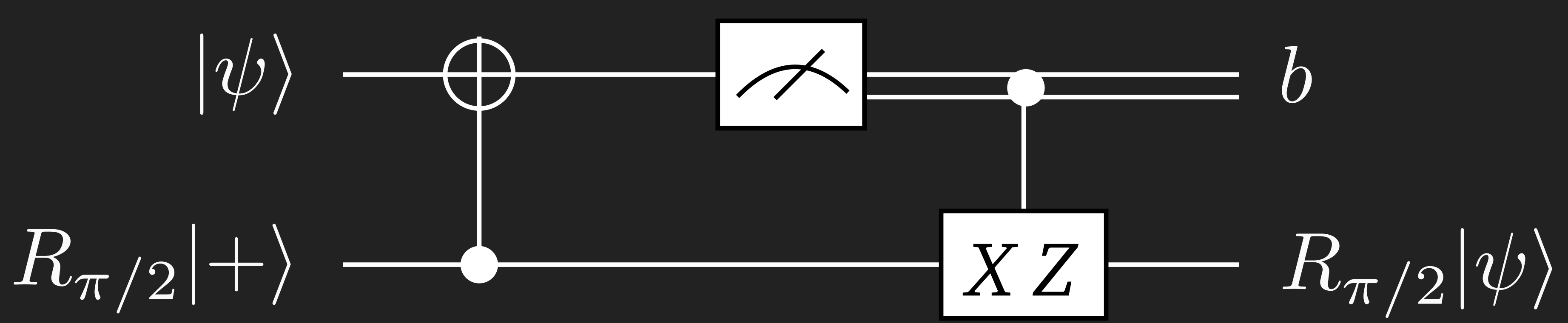


PROTOCOL: CLIENT AND SERVER

- Idea: transversal gate application

MAGIC STATES

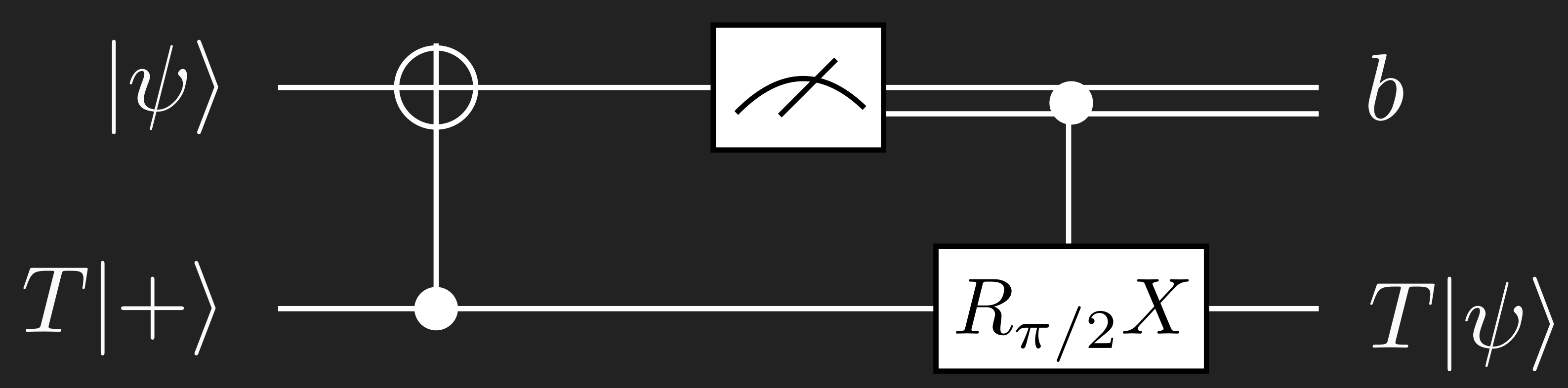
- Transversal computation works for X, Z, CNOT, and even computational-basis measurement!
- Other gates need a different trick: magic states



$$R_{\pi/2} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

PROTOCOL: CLIENT AND SERVER

- Compute on authenticated data gate-by-gate
- Some gates ($R_{\pi/2}, H, T = R_{\pi/4}$) require **authenticated magic states**, created by the client.
- Some gates (T) even require **interaction** between client and server.



SCENARIO: MULTIPARTY QUANTUM COMPUTATION

$$Enc_{k_1}(|\psi_1\rangle) \longrightarrow ? \longleftarrow Enc_{k_2}(|\psi_2\rangle)$$

- All players have inputs
- What happens if the inputs (and keys) get combined? Who can check/decode the result?

SOLUTION 1: "STACK" AUTHENTICATIONS [DNS12]

$Enc_{k_2}(Enc_{k_1}(|\psi_1\rangle))$

$Enc_{k_1}(Enc_{k_2}(|\psi_2\rangle))$

All players encode all states, so they can check all states

- ＋ plug and play security
- — extra work if player wants to check "inner" encoding
- — more players = longer ciphertexts

[DNS12] Dupuis, Nielsen, Salvail; CRYPTO 2012

SOLUTION 2: PUBLIC AUTHENTICATION TEST [DGJ+20]

$$C_1 C_2 = C$$

$$Enc_k(|\psi\rangle) = C(|\psi\rangle \otimes |0^n\rangle)$$

- Players share the key to a single encoding
- If **one** player decodes, **all** players can verify the check:
  - Decoding player appends another $|0^n\rangle$ $GL(2n, \mathbb{F}_2)$
  - "Spread out" any errors: $(C' \otimes \mathsf{X}^r)(\mathbb{I} \otimes g)(C^\dagger \otimes \mathbb{I}_n)$
  - Decoding player measures n last qubits, reports r

[DGJ+20] Dulek, Grilo, Jeffery, Majenz, Schaffner; Eurocrypt 2020.

PROTOCOL: MULTIPARTY COMPUTATION

- All players' inputs are encoded. Two options:
  - "stacked" encoding
  - shared encoding (with public authentication test)
- Gate-by-gate computation using similar ideas as in client/server setting
- Classical control using classical multiparty computation

OPEN PROBLEMS

- Other applications of "public authentication test"? [**D**GJ+20]
- Applications of multiparty computation: zero-knowledge, digital signatures, …?
- Obfuscation of quantum circuits [AF16]
- Post-quantum secure classical multi-party computation [**D**GJ+20, Section 2.2]

[DGJ+20] Dulek, Grilo, Jeffery, Majenz, Schaffner; Eurocrypt 2020.
[AF16] Alagic, Fefferman; arXiv:1602.01771

SUMMARY

- Quantum authentication codes (Enc_k / Dec_k) guarantee that a quantum state is unaltered (unless decoding rejects)
- Clifford code / trap code are examples of such codes
- Tool for:
  - Client-server setting: verifying that the server did the right computation
  - Multiparty computation: verifying that the other players did the right computation (but we need a way to combine encodings from different players)
  - …

THANK YOU!