Efficient Quantum Pseudorandomness

from Hamiltonian Phase States

John Bostanci

with Jonas Haferkamp, Dominik Hangleiter, and Alexander Poremba

Quantum computation and cryptography

Quantum computers have lots of implications for cryptography.

Quantum computation and cryptography

Quantum computers have lots of implications for cryptography.

- On one hand, people are worried that they break cryptography.

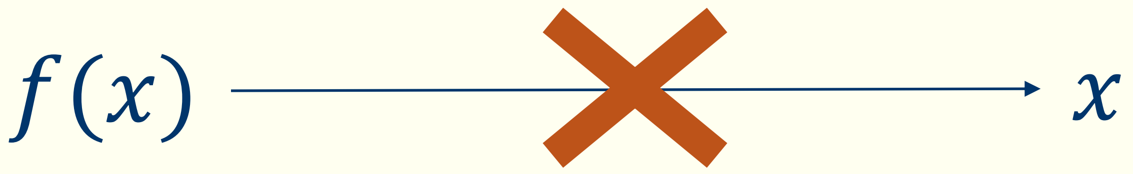
Quantum computation and cryptography

Quantum computers have lots of implications for cryptography.

- On one hand, people are worried that they break cryptography.
- Recently, there has been an explosion into research on using quantum computers to actually do cryptography!

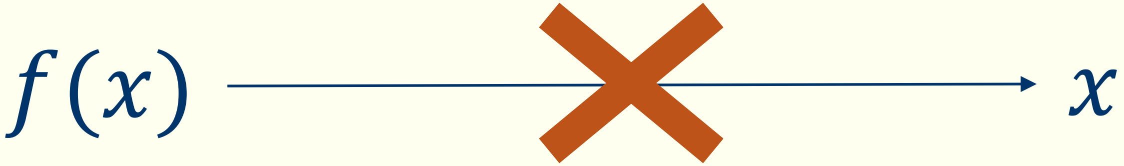
Cryptography without one-way functions

One-way functions are a family of functions that can be efficiently evaluated, but whose inverse is computationally hard to evaluate.



Cryptography without one-way functions

One-way functions are a family of functions that can be efficiently evaluated, but whose inverse is computationally hard to evaluate.



They are almost universally agreed upon as the minimal assumption in classical cryptography.

Cryptography without one-way functions

One exciting discovery is that one-way functions are not necessary for useful cryptography, without them we might still be able to do:

- Bit commitments
- Pseudorandom unitaries
- Public-key cryptography with quantum public keys

Cryptography without one-way functions

One exciting discovery is that one-way functions are not necessary for useful cryptography, without them we might still be able to do:

- Bit commitments
- Pseudorandom unitaries
- Public-key cryptography with quantum public keys

How would we actually implement these without OWFs?

Cryptography without one-way functions

Random quantum circuits?

- Probably hard to learn given samples or query access, but…

Cryptography without one-way functions

Random quantum circuits?

- Probably hard to learn given samples or query access, but…

- They probably aren't that efficient in practice.

Cryptography without one-way functions

Random quantum circuits?

- Probably hard to learn given samples or query access, but…

- They probably aren't that efficient in practice.
- They are very unstructured, and we usually want structure to exploit when building cryptography.

Cryptography without one-way functions

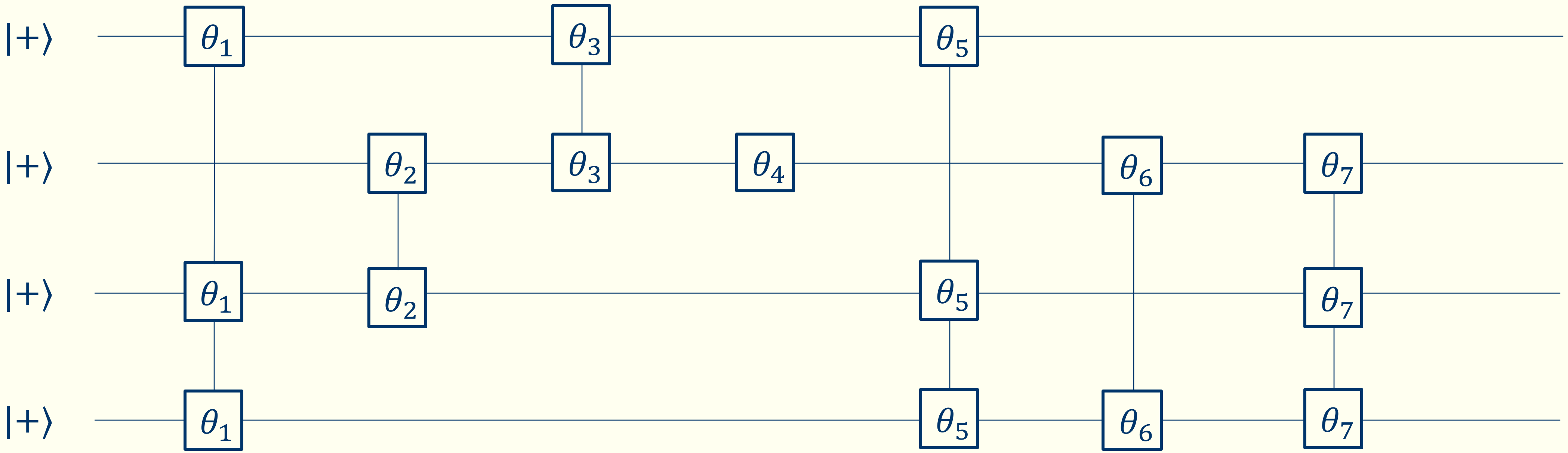
Our motivation: Can we find a plausible assumption, that is probably independent of one-way functions, and implies useful cryptography?

Cryptography without one-way functions

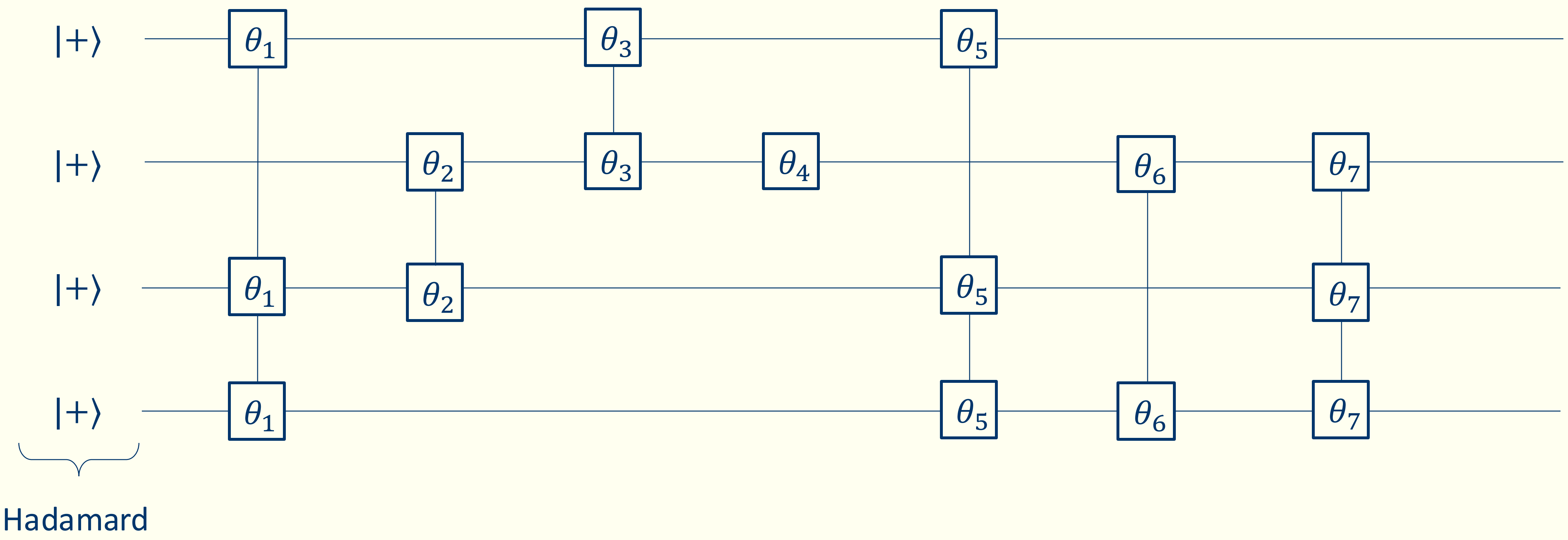
Our motivation: Can we find a plausible assumption, that is probably independent of one-way functions, and implies useful cryptography?

Our proposal: the Hamiltonian Phase States assumption (HPS).

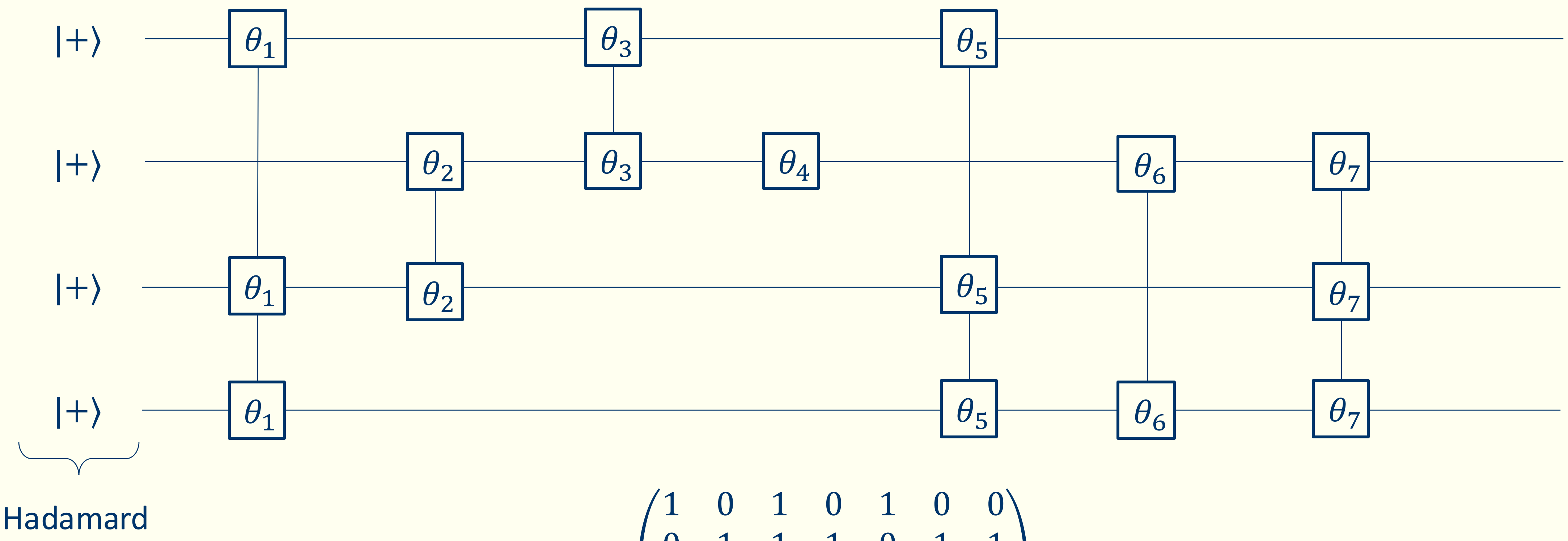
Hamiltonian Phase States, informally



Hamiltonian Phase States, informally

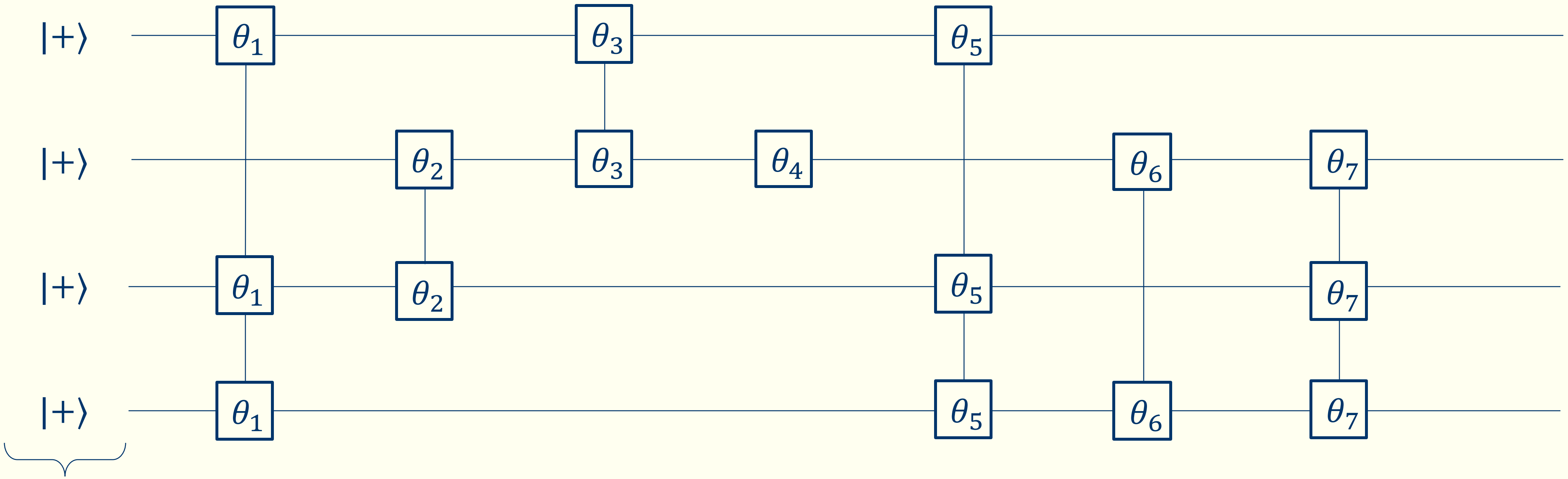


Hamiltonian Phase States, informally



Architecture matrix, $A = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$

Hamiltonian Phase States, informally



Hadamard

Architecture matrix, $A = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$

Phases, $\theta = (\theta_1, \theta_2, \dots, \theta_7)$

Hamiltonian Phase States

- A random binary matrix $A \in \mathbb{Z}_2^{m \times n}$ (the architecture).
- A random vector $\theta = (\theta_1, \dots, \theta_m) \in (0, 2\pi]^m$ (the phases).

Hamiltonian Phase States

- A random binary matrix $A \in \mathbb{Z}_2^{m\times n}$ (the architecture).
- A random vector $\theta = (\theta_1, \ldots, \theta_m) \in (0, 2\pi]^m$ (the phases).

The Hamiltonian Phase State with architecture A and phases θ is given by:

$$\left|\Phi_\theta^A\right\rangle = \exp\left(i \sum_{j=1}^{m} \theta_j \bigotimes_{k=1}^{n} Z^{A_{jk}}\right) H^{\otimes n}|0^n\rangle$$

Hamiltonian Phase States

- A random binary matrix $A \in \mathbb{Z}_2^{m \times n}$ (the architecture).
- A random vector $\theta = (\theta_1, \dots, \theta_m) \in (0, 2\pi]^m$ (the phases).

The Hamiltonian Phase State with architecture A and phases θ is given by:

$$\left|\Phi_\theta^A\right\rangle = \exp\left(i \sum_{j=1}^{m} \theta_j \bigotimes_{k=1}^{n} Z^{A_{jk}} \right) H^{\otimes n} |0^n\rangle$$

Hamiltonian Phase States

- A random binary matrix $A \in \mathbb{Z}_2^{m \times n}$ (the architecture).
- A random vector $\theta = (\theta_1, \dots, \theta_m) \in (0, 2\pi]^m$ (the phases).

The Hamiltonian Phase State with architecture A and phases θ is given by:

$$\left|\Phi_\theta^A\right\rangle = \underbrace{\exp\left(i \sum_{j=1}^{m} \theta_j \bigotimes_{k=1}^{n} Z^{A_{jk}}\right)}_{U_\theta^A} H^{\otimes n} |0^n\rangle$$

Hamiltonian Phase States

- A random binary matrix $A \in \mathbb{Z}_2^{m \times n}$ (the architecture).
- A random vector $\theta = (\theta_1, \dots, \theta_m) \in (0, 2\pi]^m$ (the phases).

The Hamiltonian Phase State with architecture A and phases θ is given by:

$$\left|\Phi_\theta^A\right\rangle = \underbrace{\exp\left(i \sum_{j=1}^{m} \theta_j \bigotimes_{k=1}^{n} Z^{A_{jk}}\right)}_{U_\theta^A} H^{\otimes n} |0^n\rangle$$

Think of $m \gg \log n$.

The HPS assumption

For all polynomial-time adversaries who are given copies of $|\Phi_\theta^A\rangle$, it is hard to…

The HPS assumption

For all polynomial-time adversaries who are given copies of $|\Phi_\theta^A\rangle$, it is hard to…

- (Search) Output (A, θ) (or something close to it).

The HPS assumption

For all polynomial-time adversaries who are given copies of $|\Phi_\theta^A\rangle$, it is hard to…

- (Search) Output (A, θ) (or something close to it).
- (Decision) Distinguish them from copies of a Haar random state.

The HPS assumption

For all polynomial-time adversaries who are given copies of $|\Phi_\theta^A\rangle$, it is hard to…

- (Search) Output (A, θ) (or something close to it).
- (Decision) Distinguish them from copies of a Haar random state.

Even if the adversary is given access to the architecture A.

Reasons to believe the HPS assumption

We provide three reasons to believe in the HPS assumption:

- Worst-to-average case reduction.
- HPS states satisfy a t-design properties.
- Best known algorithms seem to take exponential time.

Worst-to-average case reduction

Just because a problem has hard instances, it doesn’t mean it is useful for cryptography unless you can easily sample them.

Worst-to-average case reduction

Just because a problem has hard instances, it doesn't mean it is useful for cryptography unless you can easily sample them.

For HPS, we show how to do the following:

- Re-randomizing the angles, given the architecture.
- Re-randomizing the architecture.

Worst-to-average case reduction

Two observations:

- Permuting the wires of the state permutes the rows of A.
- For any n by n matrix R, applying $U_R|x\rangle \mapsto |R^{-1}x\rangle$ maps A to AR.

Worst-to-average case reduction

Two observations:

- Permuting the wires of the state permutes the rows of A.
- For any n by n matrix R, applying $U_R|x\rangle \mapsto |R^{-1}x\rangle$ maps A to AR.

This means we can do (including adding ancilla):

$$A \mapsto \Pi \begin{pmatrix} A & 0 \\ B & C \end{pmatrix} R$$

Worst-to-average case reduction

- When $m \leq n$, this is statistically indistinguishable from an independently sampled matrix.

Worst-to-average case reduction

- When $m \leq n$, this is statistically indistinguishable from an independently sampled matrix.
- When $m > n$, can be shown to be close to uniformly random in some cases (similar to re-randomizing LPN instances).

HPS form approximate t-designs

t-designs are families of states that are statistically indistinguishable from random states, up to a few copies.

$$\left|\Phi_\theta^A\right\rangle^{\otimes t} \sim \text{Haar random } |\psi\rangle^{\otimes t}$$

HPS form approximate t-designs

t-designs are families of states that are statistically indistinguishable from random states, up to a few copies.

$$\left|\Phi_\theta^A\right\rangle^{\otimes t} \sim \text{Haar random } |\psi\rangle^{\otimes t}$$

We show for $m = 2t(2nt + \log(1/\epsilon))$, Hamiltonian phase states are ϵ-approximate t-designs.

Applications of HPS

We show that HPS implies a number of interesting primitives that you couldn’t get just by assuming pseudo-random states exist:

- Public-key cryptography with quantum public keys
- Pseudo-entangled states
- Pseudorandom unitaries

Public key cryptography from HPS

Coladangelo'23 showed that quantum trapdoor functions imply public key cryptography, so we just need to build those.

Quantum trapdoor functions from HPS

Quantum trapdoor functions are described by the following:

Quantum trapdoor functions from HPS

Quantum trapdoor functions are described by the following:

- GenTrap: $1^n \rightarrow \mathrm{td}$

Quantum trapdoor functions from HPS

Quantum trapdoor functions are described by the following:

- GenTrap: $1^n \to \mathrm{td}$
- GenEval: $\mathrm{td} \to |\mathrm{eval}\rangle$

Quantum trapdoor functions from HPS

Quantum trapdoor functions are described by the following:

- GenTrap: $1^n \rightarrow \text{td}$
- GenEval: $\text{td} \rightarrow |\text{eval}\rangle$
- Eval: $(|\text{eval}\rangle, x) \rightarrow |\psi_x\rangle$

Quantum trapdoor functions from HPS

Quantum trapdoor functions are described by the following:

- GenTrap: $1^n \rightarrow \mathrm{td}$
- GenEval: $\mathrm{td} \rightarrow |\mathrm{eval}\rangle$
- Eval: $(|\mathrm{eval}\rangle, x) \rightarrow |\psi_x\rangle$
- Invert: $(\mathrm{td}, |\psi_x\rangle) \rightarrow x'$

Quantum trapdoor functions from HPS

Quantum trapdoor functions should satisfy:

- (Hard to invert) Without the trapdoor, it’s hard to find x from ψ_x, given arbitrary copies of the quantum evaluation state, for a random x.

Quantum trapdoor functions from HPS

Quantum trapdoor functions should satisfy:

- (Hard to invert) Without the trapdoor, it's hard to find x from ψ_x, given arbitrary copies of the quantum evaluation state, for a random x.
- (Correctness) With the trapdoor, Invert should always output the original x.

Quantum trapdoor functions from HPS

GenTrap(1^n): Sample a HPS instance $\rightarrow (\theta, A)$.

Quantum trapdoor functions from HPS

$\text{GenEval}(\theta, A) \rightarrow |\Phi_\theta^A\rangle$.

Quantum trapdoor functions from HPS

Eval$(|\Phi_\theta^A\rangle, x) \to |\psi_x\rangle = Z^{x_1} \otimes \cdots \otimes Z^{x_n}|\Phi_\theta^A\rangle$.

Quantum trapdoor functions from HPS

Invert$((\theta, A), |\psi_x\rangle)$: Apply $H^{\otimes n}\left(U_\theta^A\right)^{-1}$ and measure in the computational basis.

Quantum trapdoor functions from HPS

Hard to invert: Without the phases, the state looks like

$$\left(Z^x|\Phi_\theta^A\rangle, \left|\Phi_\theta^A\right\rangle^{\otimes t}\right) \sim (Z^x|\phi\rangle, |\phi\rangle^{\otimes t})$$ from the HPS assumption,

Quantum trapdoor functions from HPS

Hard to invert: Without the phases, the state looks like

$$\left(Z^x|\Phi_\theta^A\rangle, \left|\Phi_\theta^A\right\rangle^{\otimes t}\right) \sim (Z^x|\phi\rangle, |\phi\rangle^{\otimes t})$$ from the HPS assumption,

$$\sim(|\psi\rangle, |\phi\rangle^{\otimes t})$$ for two Haar random states.

Quantum trapdoor functions from HPS

Correctness:

- U_θ^A commutes with Z^x
- Applying their inverse removes the HPS diagonal matrix, and leaves the message written in the Hadamard basis.

Open questions

Can we build more interesting cryptography from HPS?

- In particular, can we build something that isn’t in “microcrypt”?

Open questions

Can we build more interesting cryptography from HPS?

- In particular, can we build something that isn’t in “microcrypt”?

Is the HPS assumption warranted?

- People should try cryptanalysis of HPS, and see if the assumption survives scrutiny!

Open questions

Can we build more interesting cryptography from HPS?

- In particular, can we build something that isn't in "microcrypt"?

Is the HPS assumption warranted?

- People should try cryptanalysis of HPS, and see if the assumption survives scrutiny!

Can we implement HPS in the real world?

- We believe our constructions should be much more efficient than constructions of quantum crypto from classical assumptions, can quantum computers today implement them?

Thanks for listening!