Improving post-quantum cryptography through cryptanalysis

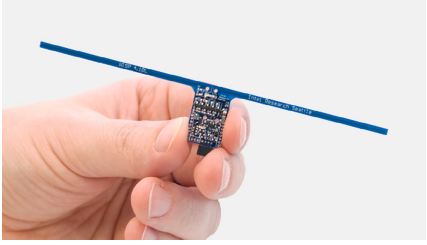
John M. Schanck

Institute for Quantum Computing
Depatment of Combinatorics and Optimization
University of Waterloo

September 4, 2019

1. A machine model for computation.
2. Cryptanalytic applications of that model.
3. The security (and efficiency) of post-quantum cryptosystems.

We want crypto that runs on...



We want crypto that runs on...

We want crypto that runs on...

but is secure against...

but is secure against...

or even…

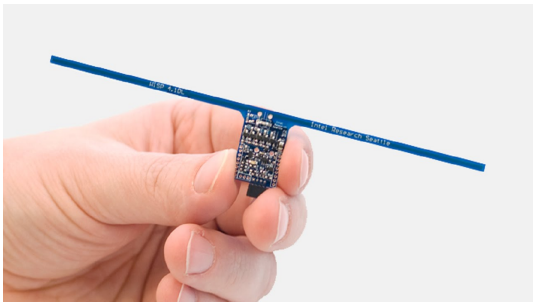

or even...

How to quantify security?

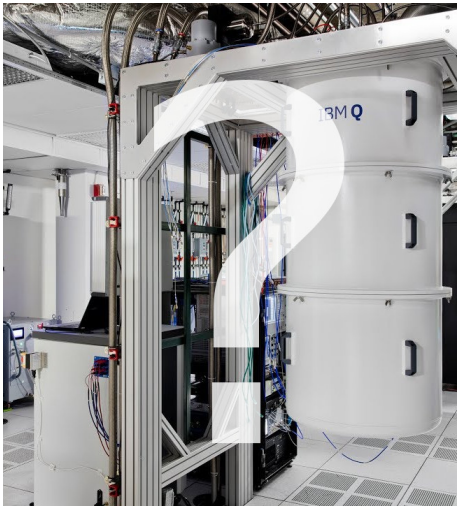
NIST (2017): A system meets the requirements of "security category 1" if...

> *Any attack that breaks the relevant security definition must require* **computational resources** *comparable to or greater than those required for key search on a block cipher with a 128-bit key (e.g. AES128).*

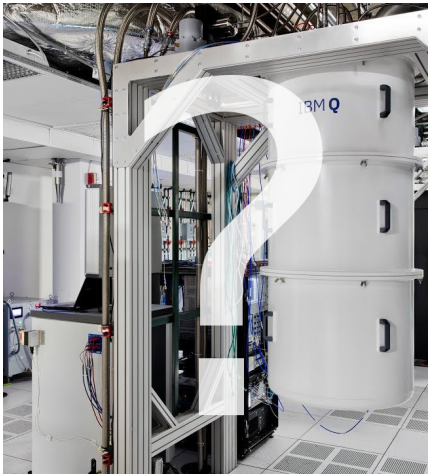
A difference of scale



Complication:



A difference of kind



Primary motivation for a new machine model:
We need to account for diverse resources. Especially when we cost quantum algorithms that use significant classical co-processing.

Secondary motivation:
Current proposals for post-quantum cryptography are *big* and *slow*.

	pre-quantum	post-quantum	
	`curve25519`	`ntruhrss701`	`sikep503`
public key bytes	32	1138	378
key gen cycles	144**k**	251**k**	15 018**k**

More fine-grained security analysis may let us *improve* performance.

Primary motivation for a new machine model:
We need to account for diverse resources. Especially when we cost quantum algorithms that use significant classical co-processing.

Secondary motivation:
Current proposals for post-quantum cryptography are *big* and *slow*.

	pre-quantum	post-quantum	
	`curve25519`	`ntruhrss701`	`sikep503`
public key bytes	32	1138	378
key gen cycles	144**k**	251**k**	15 018**k**

More fine-grained security analysis may let us *improve* performance.

Primary motivation for a new machine model:
We need to account for diverse resources. Especially when we cost quantum algorithms that use significant classical co-processing.

Secondary motivation:
Current proposals for post-quantum cryptography are *big* and *slow*.

	pre-quantum	post-quantum	
	`curve25519`	`ntruhrss701`	`sikep503`
public key bytes	32	1138	378
key gen cycles	144**k**	251**k**	15 018**k**

More fine-grained security analysis may let us *improve* performance.

How *should* we choose a machine model?

Fredkin–Toffoli (1982)

Computation — whether by man or by machine — is a physical activity, and is ultimately governed by physical principles. An important role for mathematical theories of computation is to condense in their axioms, in a stylized way, certain facts about the ultimate physical realizability of computing processes. [...] one of the first things to do is find out what aspects of physics are reflected in the axioms; perhaps one can represent in the axioms more realistic physics and reveal hitherto

Fredkin–Toffoli (1982)

Computation — whether by man or by machine — is a physical activity, and is ultimately governed by physical principles. An important role for mathematical theories of computation is to condense in their axioms, in a stylized way, certain facts about the ultimate physical realizability of computing processes. [...] one of the first things to do is find out what aspects of physics are reflected in the axioms: perhaps one can represent in the axioms more realistic physics and reveal hitherto

Fredkin–Toffoli (1982)

Computation — whether by man or by machine — is a physical activity, and is ultimately governed by physical principles. *An important role for mathematical theories of computation is to condense in their axioms, in a stylized way, certain facts about the ultimate physical realizability of computing processes.* *[...] one of the first things to do is find out what aspects of physics are reflected in the axioms: perhaps one can represent in the axioms more realistic physics and reveal hitherto*

Fredkin–Toffoli (1982)

Computation — whether by man or by machine — is a physical activity, and is ultimately governed by physical principles. An important role for mathematical theories of computation is to condense in their axioms, in a stylized way, certain facts about the ultimate physical realizability of computing processes. [...] one of the first things to do is find out what aspects of physics are reflected in the axioms: perhaps one can represent in the axioms more realistic physics and reveal hitherto unsuspected possibilities.

Fredkin–Toffoli (1982)

Computation — whether by man or by machine — is a physical activity, and is ultimately governed by physical principles. An important role for mathematical theories of computation is to condense in their axioms, in a stylized way, certain facts about the ultimate physical realizability of computing processes. [...] one of the first things to do is find out what aspects of physics are reflected in the axioms: perhaps one can represent in the axioms more realistic physics and reveal hitherto unsuspected possibilities.

I would add: and limitations.

Fredkin–Toffoli (1982)

From Turing's original discussion (Turing, 1936) it is clear that he intended to capture certain general physical constraints to which all concrete computing processes are subjected [...]

- *P1. The speed of propagation of information is bounded. (No "action at a distance": causal effects propagate through local interactions.)*
- *P2. The amount of information which can be encoded in the state of a finite system is bounded [...].*
- *P3. It is possible to construct macroscopic, dissipative physical devices which perform in a recognizable and reliable way the logical functions AND, NOT, and FAN-OUT. (This is a statement of technological fact.)*

Fredkin–Toffoli (1982)

From Turing's original discussion (Turing, 1936) it is clear that he intended to capture certain general physical constraints to which all concrete computing processes are subjected [...]

- *P1. The speed of propagation of information is bounded. (No "action at a distance": causal effects propagate through local interactions.)*
- *P2. The amount of information which can be encoded in the state of a finite system is bounded [...].*
- *P3. It is possible to construct macroscopic, dissipative physical devices which perform in a recognizable and reliable way the logical functions AND, NOT, and FAN-OUT. (This is a statement of technological fact.)*

Fredkin–Toffoli (1982)

From Turing's original discussion (Turing, 1936) it is clear that he intended to capture certain general physical constraints to which all concrete computing processes are subjected [...]

- *P1. The speed of propagation of information is bounded. (No "action at a distance": causal effects propagate through local interactions.)*
- *P2. The amount of information which can be encoded in the state of a finite system is bounded [...].*
- *P3. It is possible to construct macroscopic, dissipative physical devices which perform in a recognizable and reliable way the logical functions AND, NOT, and FAN-OUT. (This is a statement of technological fact.)*

Fredkin–Toffoli (1982)

From Turing's original discussion (Turing, 1936) it is clear that he intended to capture certain general physical constraints to which all concrete computing processes are subjected [...]

- *P1. The speed of propagation of information is bounded. (No "action at a distance": causal effects propagate through local interactions.)*
- *P2. The amount of information which can be encoded in the state of a finite system is bounded [...].*
- *P3. It is possible to construct macroscopic, dissipative physical devices which perform in a recognizable and reliable way the logical functions AND, NOT, and FAN-OUT. (This is a statement of technological fact.)*

Fredkin–Toffoli (1982)

From Turing's original discussion (Turing, 1936) it is clear that he intended to capture certain general physical constraints to which all concrete computing processes are subjected [...]

- *P1. The speed of propagation of information is bounded. (No "action at a distance": causal effects propagate through local interactions.)*
- *P2. The amount of information which can be encoded in the state of a finite system is bounded [...].*
- *P3. It is possible to construct macroscopic, dissipative physical devices which perform in a recognizable and reliable way the logical functions AND, NOT, and FAN-OUT. (This is a statement of technological fact.)*

Physical principles

Realistic machine models are

- Local,
- Finite, and
- Reliable.

Consider a **single tape Turing machine**:

- The head interacts **locally** with its tape.
- The tape alphabet is **finite**.
- We can prove, from physical assumptions, that **reliable** components can be built.

Physical principles

Realistic machine models are

- Local,
- Finite, and
- Reliable.

Consider a **single tape Turing machine**:

- The head interacts **locally** with its tape.
- The tape alphabet is **finite**.
- We can prove, from physical assumptions, that **reliable** components can be built.

Memory peripheral models

Memory peripheral models

- A **memory** is a physical system that changes over time.
- A **memory controller** is a computer that intracts with a memory.
- The **cost** of a computation is the number of interactions.

Memory peripheral models

Definition: Memory peripheral

A memory peripheral is a tuple $(\mathcal{H}, H)$ where $\mathcal{H}$ is a Hilbert space and H is a Hermitian operator on $\mathcal{H}$.

Definition: Memory peripheral model (informal)

A symmetric monoidal category whose objects are memory peripherals.

Memory peripheral models

Definition: Memory peripheral

A memory peripheral is a tuple $(\mathcal{H}, H)$ where $\mathcal{H}$ is a Hilbert space and H is a Hermitian operator on $\mathcal{H}$.

Definition: Memory peripheral model (informal)

A symmetric monoidal category whose objects are memory peripherals.

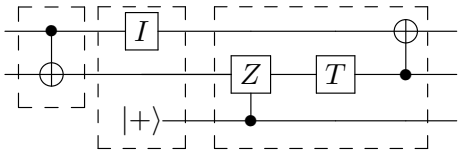
Memory peripheral models

- Morphisms of memory peripherals are quantum channels between the associated Hilbert spaces.
- Partition into "free" and "costly" morphisms.
- Interesting case: time-evolution is the distinguished free morphism.

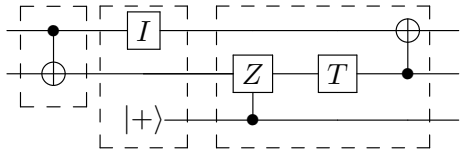
$$|\psi\rangle \mapsto e^{-iHt} |\psi\rangle$$

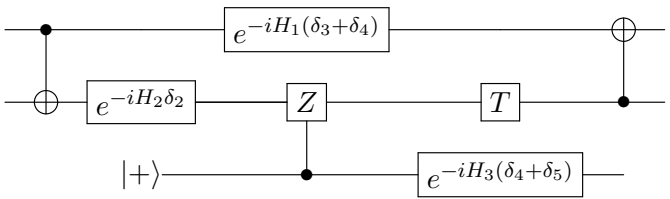
- Cost model: count the costly morphisms.

Memory peripheral models



Memory peripheral models





Examples of memory peripherals

Ideal qubit memories:

- $\mathcal{H}$ is $\mathbb{C}^2$ and H is 0.

Memories capable of **ballistic** computation:

- $\mathcal{H}$ is an array of bits and H is Benioff's Hamiltonian Turing machine.
- $\mathcal{H}$ is an array of (qu)bits and H is a Feynmann–Kitaev circuit Hamiltonian.

Memories that are **self-correcting**:

- $\mathcal{H}$ is a $2D$ lattice of bits and H is an Ising ferromagnet memory.
- $\mathcal{H}$ is a $4D$ lattice of qubits and H is Kitaev's toric code.

Examples of memory peripherals

Ideal qubit memories:

- $\mathcal{H}$ is $\mathbb{C}^2$ and H is 0.

Memories capable of **ballistic** computation:

- $\mathcal{H}$ is an array of bits and H is Benioff's Hamiltonian Turing machine.
- $\mathcal{H}$ is an array of (qu)bits and H is a Feynmann–Kitaev circuit Hamiltonian.

Memories that are **self-correcting**:

- $\mathcal{H}$ is a $2D$ lattice of bits and H is an Ising ferromagnet memory.
- $\mathcal{H}$ is a $4D$ lattice of qubits and H is Kitaev's toric code.

Examples of memory peripherals

Ideal qubit memories:

- $\mathcal{H}$ is $\mathbb{C}^2$ and H is 0.

Memories capable of **ballistic** computation:

- $\mathcal{H}$ is an array of bits and H is Benioff's Hamiltonian Turing machine.
- $\mathcal{H}$ is an array of (qu)bits and H is a Feynmann–Kitaev circuit Hamiltonian.

Memories that are **self-correcting**:

- $\mathcal{H}$ is a $2D$ lattice of bits and H is an Ising ferromagnet memory.
- $\mathcal{H}$ is a $4D$ lattice of qubits and H is Kitaev's toric code.

Physical principles

- Locality – Not inherent. Can impose additional geometry on "sub-peripherals."
- Finiteness – Not inherent. Can impose constraint on the number of (costly) morphisms available.
- Reliability – Can augment H with a thermal bath and interaction terms.

Difficult to satisfy all three!

Major open problem in physics: Is it possible to construct a self-correcting quantum memory (some H) from local interactions in $3D$ eucliean space?

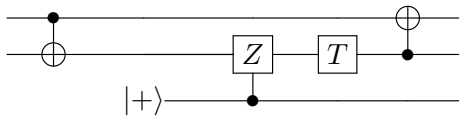
Physical principles

- Locality – Not inherent. Can impose additional geometry on "sub-peripherals."
- Finiteness – Not inherent. Can impose constraint on the number of (costly) morphisms available.
- Reliability – Can augment H with a thermal bath and interaction terms.

Difficult to satisfy all three!

Major open problem in physics: Is it possible to construct a self-correcting quantum memory (some H) from local interactions in $3D$ eucliean space?

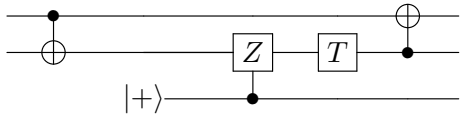
Gate cost (G-cost)



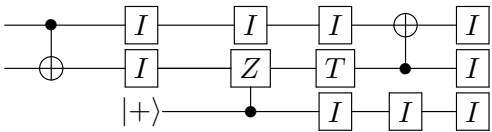
Depth-Width cost (DW-cost)

Can assume units of "RAM operations" — a classical computer triggers each gate.

Gate cost (G-cost)

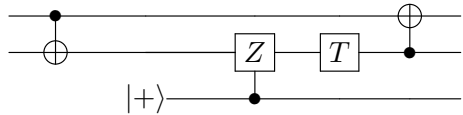


Depth-Width cost (DW-cost)

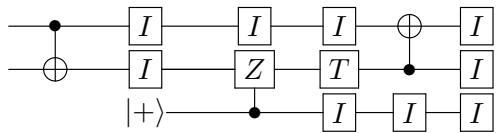


Can assume units of "RAM operations" — a classical computer triggers each gate.

Gate cost (G-cost)



Depth-Width cost (DW-cost)



Can assume units of "RAM operations" — a classical computer triggers each gate.

Grassl–Langenberg–Roetteler–Steinwandt AES attack circuit (2016)

	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	2,953
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	4,449
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

Any attack that breaks the relevant security definition must require **computational resources** *comparable to or greater than those required for key search on a block cipher with a 128-bit key (e.g. AES128).*

Grassl–Langenberg–Roetteler–Steinwandt AES attack circuit (2016)

	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	2,953
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	4,449
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

Any attack that breaks the relevant security definition must require **computational resources** *comparable to or greater than those required for key search on a block cipher with a 128-bit key (e.g. AES128).*

Grassl–Langenberg–Roetteler–Steinwandt AES attack circuit (2016)

	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	2,953
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	4,449
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

(Unoptimized) DW-cost is

$$1.16 \cdot 2^{81} \cdot 2953 = 2^{92.74\ldots} \text{ RAM ops}$$

T-gates are more expensive than Clifford gates. At 1 ns / gate, depth 2^{81} is 2^{26} years, and parallelization increases costs. Respecting locality increases costs. Realistic error correction increases costs.

Grassl–Langenberg–Roetteler–Steinwandt AES attack circuit (2016)

	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	2,953
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	4,449
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

(Unoptimized) DW-cost is

$$1.16 \cdot 2^{81} \cdot 2953 = 2^{92.74\ldots} \text{ RAM ops}$$

T-gates are more expensive than Clifford gates. At 1 ns / gate, depth 2^{81} is 2^{26} years, and parallelization increases costs. Respecting locality increases costs. Realistic error correction increases costs.

Grassl–Langenberg–Roetteler–Steinwandt AES attack circuit (2016)

	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	2,953
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	4,449
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

(Unoptimized) DW-cost is

$$1.16 \cdot 2^{81} \cdot 2953 = 2^{92.74\ldots} \text{ RAM ops}$$

T-gates are more expensive than Clifford gates. At 1 ns / gate, depth 2^{81} is 2^{26} years, and parallelization increases costs. Respecting locality increases costs. Realistic error correction increases costs.

Grassl–Langenberg–Roetteler–Steinwandt AES attack circuit (2016)

	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	2,953
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	4,449
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

(Unoptimized) DW-cost is

$$1.16 \cdot 2^{81} \cdot 2953 = 2^{92.74\ldots} \text{ RAM ops}$$

T-gates are more expensive than Clifford gates. At 1 ns / gate, depth 2^{81} is 2^{26} years, and parallelization increases costs. Respecting locality increases costs. Realistic error correction increases costs.

Grassl–Langenberg–Roetteler–Steinwandt AES attack circuit (2016)

	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	2,953
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	4,449
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

(Unoptimized) DW-cost is

$$1.16 \cdot 2^{81} \cdot 2953 = 2^{92.74\ldots} \text{ RAM ops}$$

T-gates are more expensive than Clifford gates. At 1 ns / gate, depth 2^{81} is 2^{26} years, and parallelization increases costs. Respecting locality increases costs. Realistic error correction increases costs.

Grassl–Langenberg–Roetteler–Steinwandt AES attack circuit (2016)

	#gates		depth		#qubits
k	T	Clifford	T	overall	
128	$1.19 \cdot 2^{86}$	$1.55 \cdot 2^{86}$	$1.06 \cdot 2^{80}$	$1.16 \cdot 2^{81}$	2,953
192	$1.81 \cdot 2^{118}$	$1.17 \cdot 2^{119}$	$1.21 \cdot 2^{112}$	$1.33 \cdot 2^{113}$	4,449
256	$1.41 \cdot 2^{151}$	$1.83 \cdot 2^{151}$	$1.44 \cdot 2^{144}$	$1.57 \cdot 2^{145}$	6,681

Likely cheaper than classical search (2^{143} RAM operations).
But it's not entirely clear what "computational resources" are required.

This thesis

- More clearly state what "computational resources" are believed to be required for various cryptanalytic tasks.
- Be realistic without being too dependent on any particular technology.
- Provide **comparable** security estimates for various post-quantum cryptosystems.

Examples

Matthew Amy, Olivia Di Matteo, Vlad Gheorghiu, Michele Mosca, Alex Parent, and John M. Schanck. Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3.
In *International Conference on Selected Areas in Cryptography*, pages 317–337. Springer, 2016.
https://eprint.iacr.org/2016/992

Gives quantum circuits for generic pre-image search on SHA-256/SHA3-256.

$\Rightarrow$ The 2^{128} query attacks could have DW-cost as high as 2^{166}.

Examples

John M. Schanck. Multi-power post-quantum RSA.
Cryptology ePrint Archive, Report 2018/325, 2018.
https://eprint.iacr.org/2018/325

Gives an analysis of using Shor's algorithm to attack "multi-power RSA" moduli.

$\Rightarrow$ 100 000$\times$ speedup for pqRSA key generation with only a small loss of security.

Examples

Samuel Jaques and John M. Schanck. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE.
In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 32–61, Cham, 2018. Springer International Publishing.
https://eprint.iacr.org/2019/103

Introduces memory peripheral models.
Improves on the best known DW-cost of basic data structures like sorted lists.
Gives an analysis of quantum walks on Johnson graphs incl. Tani's claw-finding algorithm.

$\Rightarrow$ A $n^{1/3+o(1)}$ query algorithm has G and DW-cost $n^{1/2+o(1)}$ (different $o(1)$).
$\Rightarrow$ SIKEp434 offers NIST level 1 security (original claim SIKEp751)

Examples

Martin Albrecht, Vlad Gheorghiu, Eammon Postlethwaite, and John M. Schanck.

Quantum near neighbor search and lattice sieves.
In preparation, 2019

Analysis of quantum variants of near neighbor search algorithms with application to solving the shortest vector problem in a lattice.

$\Rightarrow$ A classical attack with cost exponent $(0.292\ldots + o(1))d$ is more relevant than a quantum attack with cost exponent $(0.265\ldots + o(1))d$ for lattice dimensions d of cryptographic interest (even if access to a $2^{O(d)}$-bit qRAM is unit cost).

Post-quantum RSA

- Take $n = p_1 p_2 \cdots p_\ell$.
- Each prime $(\lg \lg n)^{2+o(1)}$ bits.
- Key generation, encryption, and decryption can all be computed at a cost of $(\lg n)(\lg \lg n)^{O(1)}$ RAM operations using fast multiplication techniques.
- Shor's algorithm costs $(\lg n)^{2+o(1)}$ qubit operations, assuming fast multiplication.

Cost/performance ratio: $(\lg n)^{2+o(1)}/(\lg n)(\lg \lg n)^{O(1)} = (\lg n)^{1+o(1)}$.

Two ways to improve this:

- Improve performance without changing cost of Shor.
- Show Shor is more expensive.

Post-quantum RSA

- Take $n = p_1 p_2 \cdots p_\ell$.
- Each prime $(\lg \lg n)^{2+o(1)}$ bits.
- Key generation, encryption, and decryption can all be computed at a cost of $(\lg n)(\lg \lg n)^{O(1)}$ RAM operations using fast multiplication techniques.
- Shor's algorithm costs $(\lg n)^{2+o(1)}$ qubit operations, assuming fast multiplication.

Cost/performance ratio: $(\lg n)^{2+o(1)}/(\lg n)(\lg \lg n)^{O(1)} = (\lg n)^{1+o(1)}$.

Two ways to improve this:

- Improve performance without changing cost of Shor.
- Show Shor is more expensive.

Post-quantum RSA

- Take $n = p_1 p_2 \cdots p_\ell$.
- Each prime $(\lg \lg n)^{2+o(1)}$ bits.
- Key generation, encryption, and decryption can all be computed at a cost of $(\lg n)(\lg \lg n)^{O(1)}$ RAM operations using fast multiplication techniques.
- Shor's algorithm costs $(\lg n)^{2+o(1)}$ qubit operations, assuming fast multiplication.

Cost/performance ratio: $(\lg n)^{2+o(1)}/(\lg n)(\lg \lg n)^{O(1)} = (\lg n)^{1+o(1)}$.

Two ways to improve this:

- Improve performance without changing cost of Shor.
- Show Shor is more expensive.

Post-quantum RSA

Prime generation dominates user's costs:

- In one test, prime generation took 1 975 000 core-hours or "four months running on spare compute capacity of a 1,400-core cluster" and evaluating the product tree took "about four days".

Bernstein–Heninger–Lou–Valenta (2017):

> *One can try to further accelerate key generation using Takagi's idea of choosing n as $p^{k-1}q$. We point out two reasons that this is worrisome. The first reason is lattice attacks. The second reason is that any nth power modulo n has small order, namely some divisor of $(p-1)(q-1)$; Shor's algorithm finds the order at relatively high speed once the nth power is computed.*

Post-quantum RSA

Prime generation dominates user's costs:

- In one test, prime generation took 1 975 000 core-hours or "four months running on spare compute capacity of a 1,400-core cluster" and evaluating the product tree took "about four days".

Bernstein–Heninger–Lou–Valenta (2017):

> *One can try to further accelerate key generation using Takagi's idea of choosing n as $p^{k-1}q$. We point out two reasons that this is worrisome. The first reason is lattice attacks. The second reason is that any nth power modulo n has small order, namely some divisor of $(p-1)(q-1)$; Shor's algorithm finds the order at relatively high speed once the nth power is computed.*

What's "worrisome"

- Suppose $n = p^{k-1}q$.
- The (multiplicative) order of 3 divides $\varphi(n) = p^{k-2}(p-1)(q-1)$.
- The order of 3^n divides $\varphi(n)/\gcd(n, \varphi(n)) = (p-1)(q-1)$
- If order of a is less than S, then the cost of Shor's algorithm is dominated by $O(\lg S)$ modular multiplications.
- Take $a = 3^n \bmod n$.

Multi-power pqRSA

- Take $n = p_1^{\pi_1} p_2^{\pi_2} \cdots p_\ell^{\pi_\ell}$ with π_i the i-th prime.
- Order of 3^n is roughly $\sqrt{n}$.
- Shor costs $(\lg n)^{1.5+o(1)}$ (rather than $(\lg n)^{2+o(1)}$).
- **But!** the classical cost of computing $3^n \bmod n$ is still $(\lg n)^{2+o(1)}$
- This key form reduces key generation time from 4 months to 5 days.
  With 1 terabyte n, attack costs $\approx 2^{83}$ quantum gates and $\approx 2^{100}$ classical gates.
  Attack on ordinary pqRSA costs 2^{100} quantum gates and a negligible amount of classical co-processing.

Multi-power pqRSA

- Take $n = p_1^{\pi_1} p_2^{\pi_2} \cdots p_\ell^{\pi_\ell}$ with π_i the i-th prime.
- Order of 3^n is roughly $\sqrt{n}$.
- Shor costs $(\lg n)^{1.5+o(1)}$ (rather than $(\lg n)^{2+o(1)}$).
- **But!** the classical cost of computing $3^n \bmod n$ is still $(\lg n)^{2+o(1)}$
- This key form reduces key generation time from 4 months to 5 days.
  With 1 terabyte n, attack costs $\approx 2^{83}$ quantum gates and $\approx 2^{100}$ classical gates.
  Attack on ordinary pqRSA costs 2^{100} quantum gates and a negligible amount of classical co-processing.

Multi-power pqRSA

- Take $n = p_1^{\pi_1} p_2^{\pi_2} \cdots p_\ell^{\pi_\ell}$ with π_i the i-th prime.
- Order of 3^n is roughly $\sqrt{n}$.
- Shor costs $(\lg n)^{1.5+o(1)}$ (rather than $(\lg n)^{2+o(1)}$).
- **But!** the classical cost of computing $3^n \bmod n$ is still $(\lg n)^{2+o(1)}$
- This key form reduces key generation time from 4 months to 5 days.
  With 1 terabyte n, attack costs $\approx 2^{83}$ quantum gates and $\approx 2^{100}$ classical gates.
  Attack on ordinary pqRSA costs 2^{100} quantum gates and a negligible amount of classical co-processing.

Might Shor cost more than $(\lg n)^{2+o(1)}$?

Some interesting lower bounds in Area-Time or "VLSI" models:

- Thompson (1979) for DFT.
- Brent–Kung (1980) for binary tree layouts.
- Brent–Kung (1981) for binary multiplication.

Can we translate these to a quantum setting?
Maybe in an anyon model? Or surface codes with lattice surgery?
Direct translation would imply a $(\lg n)^{2.5+o(1)}$ **lower bound** on cost of Shor.
(And make multi-power pqRSA less attractive.)

Other future work

- Secure CSIDH parameters (Kuperberg sieve).
- Analysis of quantum lattice point enumeration algorithms (backtracking).
- Analysis of quantum information set decoding algorithms (another Johnson graph algorithm).
- Decryption failure attacks on lattice schemes.
  (Quantum algorithm for sampling from hard-core distribution of given fugacity?)


Martin Albrecht, Vlad Gheorghiu, Eammon Postlethwaite, and John M. Schanck.
Quantum near neighbor search and lattice sieves.
In preparation, 2019.


Matthew Amy, Olivia Di Matteo, Vlad Gheorghiu, Michele Mosca, Alex Parent, and John M. Schanck.
Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3.
In *International Conference on Selected Areas in Cryptography*, pages 317–337. Springer, 2016.
https://eprint.iacr.org/2016/992.


Samuel Jaques and John M. Schanck.
Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE.
In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 32–61, Cham, 2018. Springer International Publishing.
https://eprint.iacr.org/2019/103.


John M. Schanck.
Multi-power post-quantum RSA.
Cryptology ePrint Archive, Report 2018/325, 2018.