Quantum-Safe Crypto Why & How?

JP Aumasson, Kudelski Security

Sorry, this product is no longer available!

See larger image

100 pcs lot Lava pendants Energy quantum scalar pendant energy card and Resist radiation JHE0151

Price: **US $263.72 - 341.67** / Lot (US $2.64 - 3.42 / Piece) Reference Currency

100 Pieces / Lot

Wholesale Price (Lot):	1 +	3 +	15 +	16 +	32 +
	US $341.67	US $334.70	US $317.96	US $314.75	US $313.17

Quantity: 1 Lot

Shipping Cost: **US $0.95** to Switzerland Via China Post Air Mail

Estimated delivery time: Dec 26 and Jan 5, ships out within 7 business days ?

Total Cost: US $342.62

Item sold out | Add to Favorite Items (0)

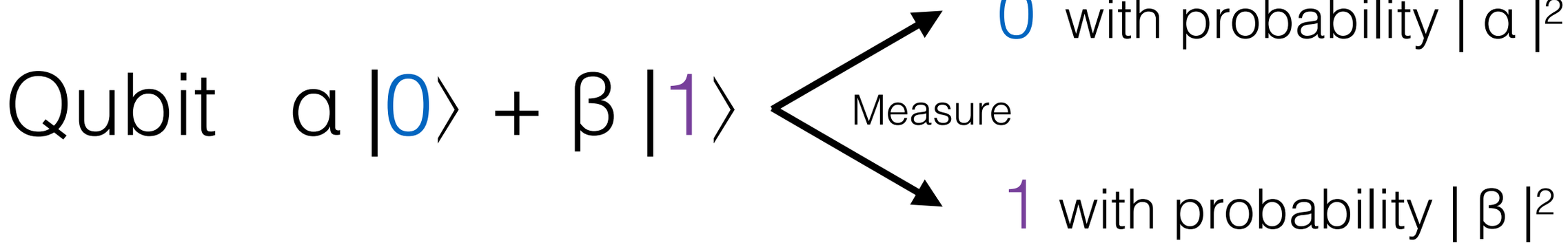
Health Benefits:

1. Reduces inflammation.
2. Promotes unclamping of cells.
3. Enhances immune and endocrine systems.
4. Helps to protect DNA from damage.
5. Improves stamina, endurance and strength.
6. Alleviates soreness, aches and pains, and improves flexibility.
7. Helps to retard the ageing process.
8. Helps to fight cancer cells.
9. Has the ability to destroy viruses and bacteria.
10. Enhances cellular nutrition and detoxification.
11. Enhances cellular permeability.
12. Increases energy.
13. Strengthens the body's biofield preventing electro-magnetic waves from affecting one's health.
14. Increases focus and concentration.
15. Improves blood Circulation.
16. Energizes block cells and reduces "stickiness".

Flight plan

- What's a quantum computer?
- How broken are your public keys?
- AES vs. quantum search
- Hidden quantum powers
- Defeating quantum computing
- Hash functions to the rescue

Qubits instead of bits



Stay 0 or 1 forever

Generalizes to more than 2 states: qutrits, qubytes, etc.

Complex, negative probabilities (amplitudes), **real randomness**

Quantum computer

Just high-school linear algebra

Quantum registers, a bunch of quantum states

~ N qubits encode a list of 2^N amplitudes

Quantum assembly instructions

~ Matrix multiplications preserving amplitudes' normalization

Quantum circuits usually end with a **measurement**

Can’t be simulated classically! (needs 2^N storage/compute)

Quantum speedup

When quantum computers can solve a problem faster than classical computers

Most interesting: **Superpolynomial** quantum speedup

List on the Quantum Zoo: http://math.nist.gov/quantum/zoo/

Quantum parallelism

Quantum computers sort of encode all values simultaneously

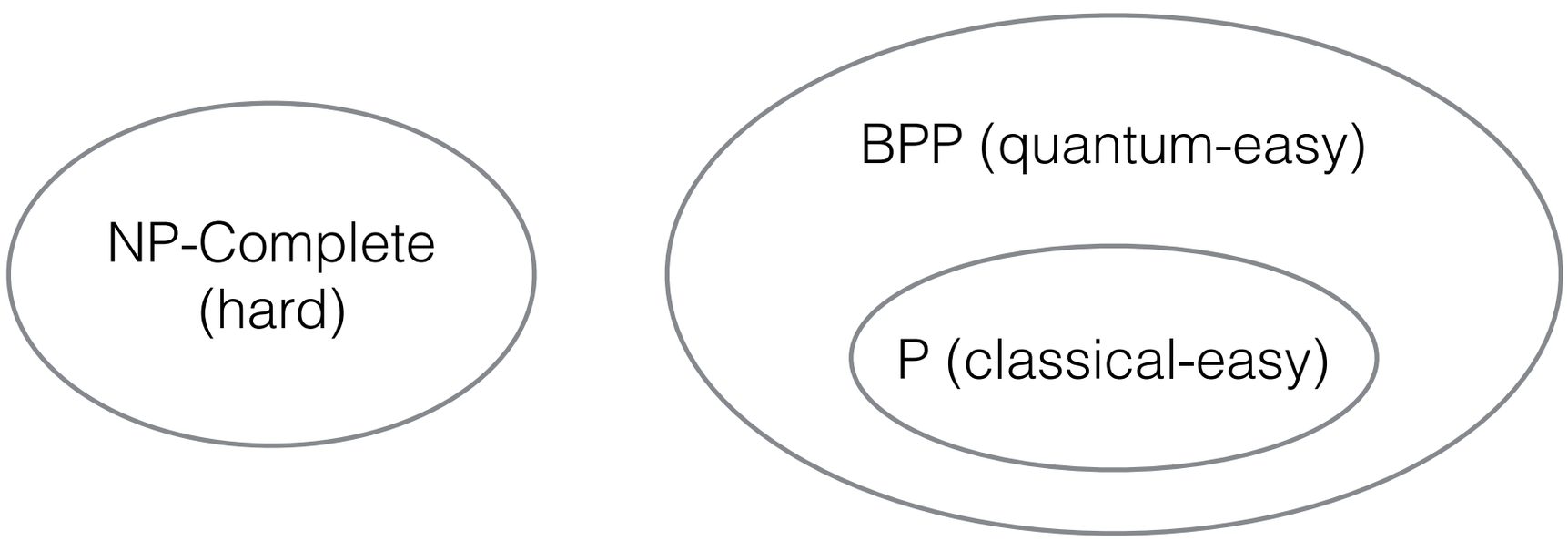
But they do not try every answer in parallel and pick the best one

Quantum parallelism is a more complicated notion…

NP-complete problems

- Solution hard to find, but easy to verify
- SAT, scheduling, Candy Crush, etc.
- Sometimes used in crypto

Can’t be solved faster with quantum computers



How broken are your public keys?

Shor them all

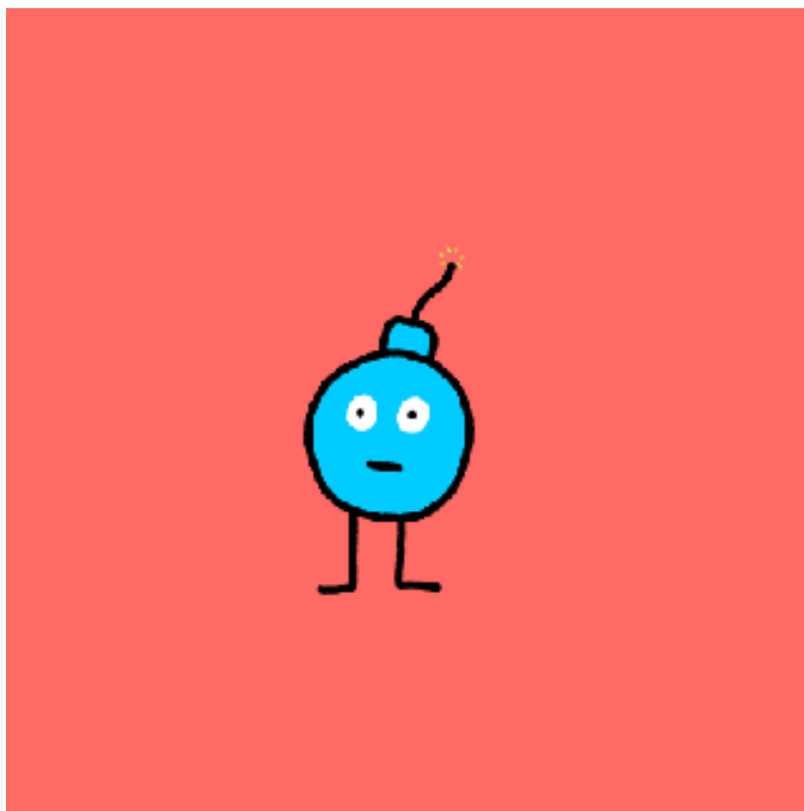
Shor's algorithm finds a structure in abelian subgroups:

- Finds **p** given **n** = **pq**
- Finds **d** given **y** = $\mathbf{x^d}$ mod **p**

Fast on a quantum computer

Practically impossible classically

#ExponentialSpeedup



How bad is it?

Cool: signatures

- Reissue signatures with a post-quantum algorithm

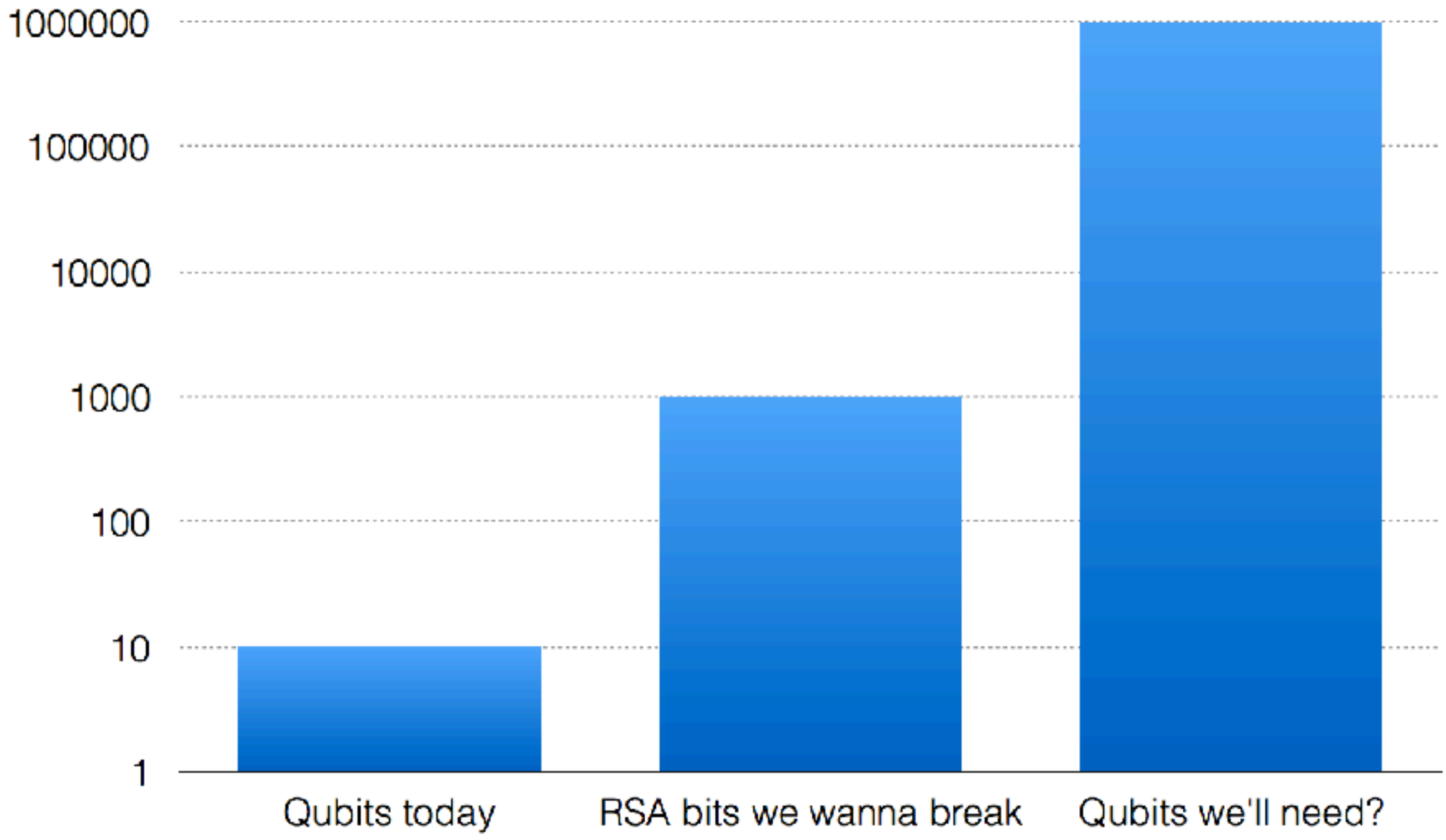
Bad: key agreement

- Can be mitigated with secret states (ratcheting)

Ugly: encryption

- Encrypted messages compromised forever

We’re not there yet



(log scale)

Designing a Million-Qubit Quantum Computer Using Resource Performance Simulator

Muhammad Ahsan, Rodney Van Meter, Jungsang Kim

(Submitted on 2 Dec 2015)

The optimal design of a fault-tolerant quantum computer involves finding an appropriate balance between the burden of large-scale integration of noisy components and the load of improving the reliability of hardware technology. This balance can be evaluated by quantitatively modeling the execution of quantum logic operations on a realistic quantum hardware containing limited computational resources. In this work, we report a complete performance simulation software tool capable of (1) searching the hardware design space by varying resource architecture and technology parameters, (2) synthesizing and scheduling fault-tolerant quantum algorithm within the hardware constraints, (3) quantifying the performance metrics such as the execution time and the failure probability of the algorithm, and (4) analyzing the breakdown of these metrics to highlight the performance bottlenecks and visualizing resource utilization to evaluate the adequacy of the chosen design. Using this tool we investigate a vast design space for implementing key building blocks of Shor's algorithm to factor a 1,024-bit number with a baseline budget of 1.5 million qubits. We show that a trapped-ion quantum computer designed with twice as many qubits and one-tenth of the baseline infidelity of the communication channel can factor a 2,048-bit integer in less than five months.

AES vs. quantum search

AES

NIST's "**Advanced Encryption Standard**"

- THE symmetric encryption standard
- Supports keys of 128, 192, or 256 bits
- **Everywhere**: TLS, SSH, IPsec, quantum links, etc.

Quantum search

Grover's algorithm: searches in N items in $\sqrt{N}$ queries!

=> AES broken in $\sqrt{(2^{128})} = 2^{64}$ operations

Caveats behind this simplistic view:

- It's actually **O(√N)**, constant factor in O()'s may be huge
- Doesn't easily parallelize as classical search does

Quantum-searching AES keys

	#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

Table 5. Quantum resource estimates for Grover's algorithm to attack AES-k, where $k \in \{128, 192, 256\}$.

https://arxiv.org/pdf/1512.04965v1.pdf

If gates are the size of a hydrogen atom (12pm) this depth is the **diameter of the solar system** (~10^{13}m)
(Yet worth less than 5 grams of hydrogen)

No doubts more efficient circuits will be designed…

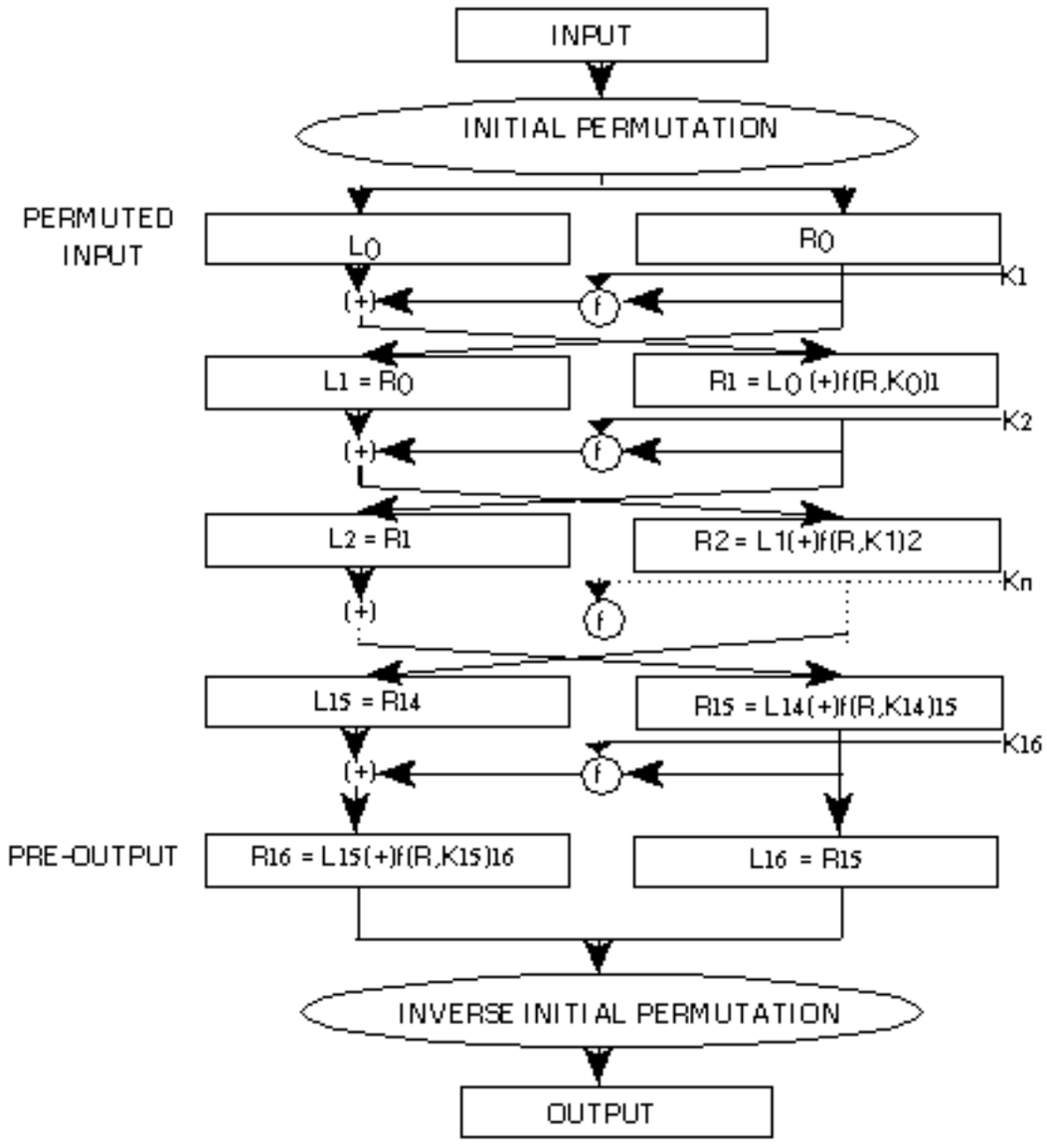
Grover is not a problem…

… just double key length

And that’s it, problem solved!

Hidden quantum powers

Crypto algorithms aren't black boxes



Can sometimes be viewed as quantum-friendly algorithms

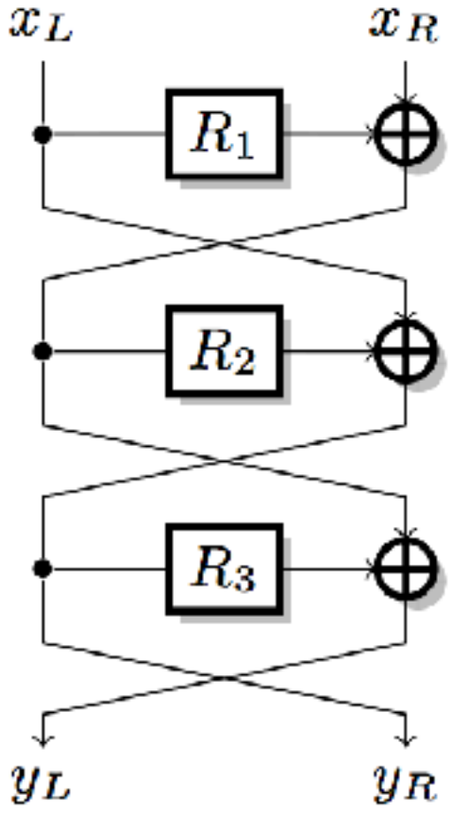


Fig. 3. Three-round Feistel scheme.

#QuantumSpeedup

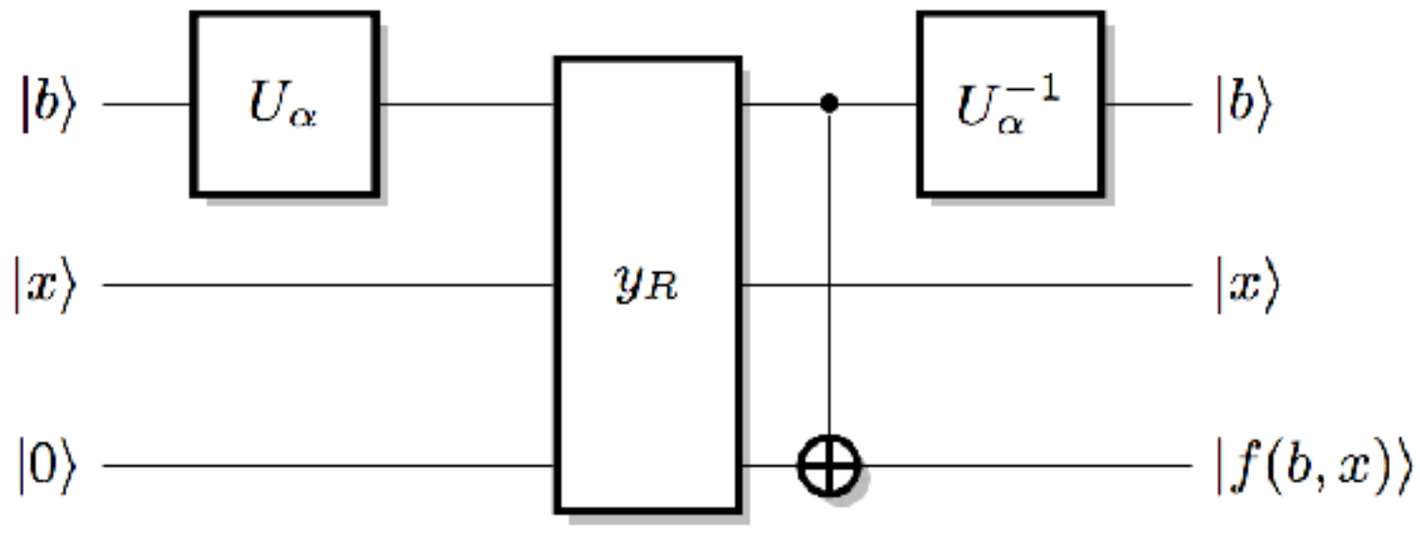


Fig. 4. Simon's function for Feistel.

https://arxiv.org/pdf/1602.05973v3.pdf

Simon's problem

The poster child of quantum exponential speedups:

- Black box function **f** from *n*-bit strings to *n*-bit strings
- Find the value *M* such that for any *X*: $\mathbf{f}(X) = \mathbf{f}(X \oplus M)$
  (Such *M* is guaranteed to exist)

$O(2^{n/2})$ classically, $O(n)$ quantumly!

Breaks authentication in AES modes GCM and OCB!

Caveat: needs *superposition queries* (unlikely)

Defeating quantum computing

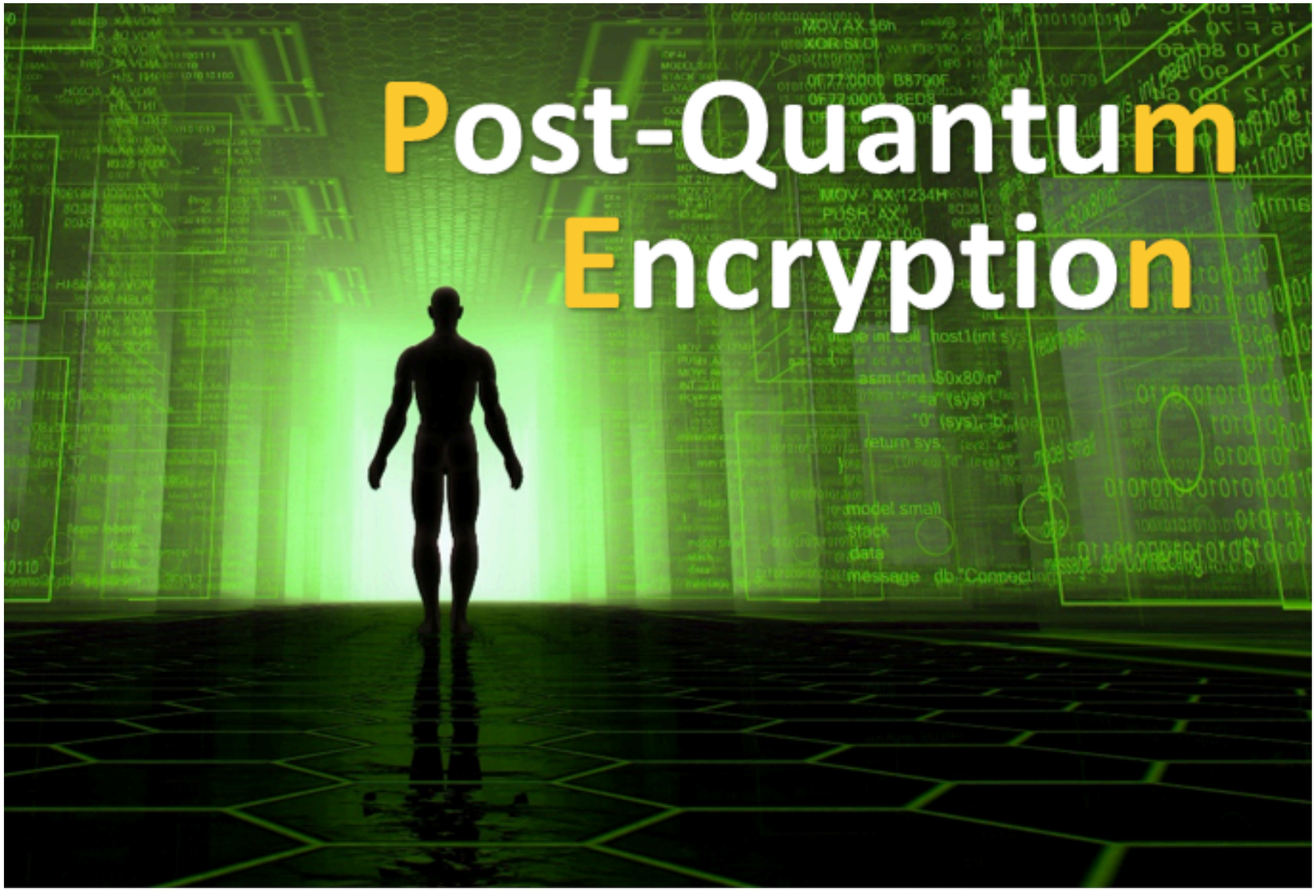


Image credit: crypto company Dyadic Security

Post-quantum crypto

A.k.a. “quantum-safe”, “quantum-resilient”

Algorithms not broken by a quantum computer…

- Must not rely on factoring or discrete log problems
- Must be well-understood with respect to quantum

Have sometimes been broken.. classically ¯_(ツ)_/¯

Why care?

Insurance against QC threat:

- “QC has a probability *p* work in year 2YYY”

- “I’d like to eliminate this risk"

Why care?

NSA recommendations for National Security Systems

"we anticipate a need to shift to quantum-resistant cryptography in the near future."

(In CNSS advisory 02-15)

Why care?

CSRC HOME > GROUPS > CT > POST-QUANTUM CRYPTOGRAPHY PROJECT

POST-QUANTUM CRYPTO PROJECT

NEWS -- August 2, 2016: The National Institute of Standards and Technology (NIST) is requesting comments on a new process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms. Please see the Post-Quantum Cryptography Standardization menu at left.

Fall 2016	Formal Call for Proposals
Nov 2017	Deadline for submissions
Early 2018	Workshop - Submitter's Presentations
3-5 years	Analysis Phase - NIST will report findings *1-2 workshops during this phase*
2 years later	Draft Standards ready

Lattice-based crypto

Based on problems such as **learning with errors** (LWE):

- **S** a secret vector of numbers modulo q
- Receive pairs for (**A**, **B** = <**S**, **A**> + **E**)
  - **A** known, uniformly random
  - **E** unknown Gaussian distribution

Goal: find **S**, or distinguish (**A**, **B**) from uniform random

Lattice-based crypto

TECHNOLOGY

Google Experimenting With 'New Hope' Post-Quantum Encryption To Safeguard Chrome

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Experimenting with Post-Quantum Cryptography

July 7, 2016

Lattice-based crypto

Google's Post-Quantum Cryptography

News has been bubbling about an announcement by Google that it's starting to experiment with public-key cryptography that's resistant to cryptanalysis by a quantum computer. Specifically, it's experimenting with the New Hope algorithm.

It's certainly interesting that Google is thinking about this, and probably okay that it's available in the Canary version of Chrome, but this algorithm is by no means ready for operational use. Secure public-key algorithms are *very* hard to create, and this one has not had nearly enough analysis to be trusted. Lattice-based public-key cryptosystems such as New Hope are particularly subtle -- and we cryptographers are still learning a lot about how they can be broken.

Targets are important in cryptography, and Google has turned New Hope into a good one. Consider this an opportunity to advance our cryptographic knowledge, not an offer of a more-secure encryption option. And this is the right time for this area of research, before quantum computers make discrete-logarithm and factoring algorithms obsolete.

Challenges with lattices

- Estimate security level for given parameters

- Make sure that it's secure against **all** computers

More post-quantumness

- Based on **coding theory** (McEliece, Niederreiter):
  - Solid foundations (late 1970s)
  - Large keys (dozen kBs)
  - *Encryption only*

- Based on **multivariate polynomials** evaluation
  - Secure in theory, not always in practice
  - *Mostly for signatures*

Hash functions to the rescue

Hash functions

- Input of any size, output of 256 or 512 bits
- Can't invert, can't find collisions
- BLAKE2, SHA-3, SHA-256, ~~SHA-1~~, ~~MD5~~…

Hash-based signatures

Unique compared to other post-quantum schemes:

- No mathematical/structured hard problem
- As secure as underlying hash functions
- Good news: we have secure hash functions!

Hash-based signatures

But there’s a catch…

Hash-based signatures

- Not fast (but not always a problem)
- Large signatures (dozen of kBs)
- Statefulness problem…

One-time signatures

Lamport, **1979**:

1. Generate a key pair
   - Pick random strings $\mathbf{K}_0$ and $\mathbf{K}_1$ (your **private key**)
   - The public key is the two values $\mathbf{H}(\mathbf{K}_0)$, $\mathbf{H}(\mathbf{K}_1)$

2. To sign the bit 0, show $\mathbf{K}_0$, to sign 1 show $\mathbf{K}_1$

One-time signatures

- Need as many keys as there are bits
- A key can only be used once

Sign more than 0 and 1

Winternitz, **1979**:

1. Public key is **H**(**H**(**H**(**H**(…. (**K**)…)) = **H**$^{\mathbf{w}}$(**K**). (**w** times)

2. To sign a number **x** in [0; **w** – 1], compute **S**=**H**$^{\mathbf{x}}$(**K**)

Verification: check that **H**$^{\mathbf{w-x}}$(**S**) = public key

A key must still be used only once

Few-time signatures

HORS (Hash to Obtain Random Subset)
Reyzin, Reyzin, **2002**

To sign **M**, use a *selection function S*: **M** → indexes

	1	2	3	4	5	…	*n*
Private keys	**K**$_1$	**K**$_2$	**K**$_3$	**K**$_4$	**K**$_5$	…	**K**$_n$
	↓	↓	↓	↓	↓		↓
Public keys	**H**(**K**$_1$)	**H**(**K**$_2$)	**H**(**K**$_3$)	**H**(**K**$_4$)	**H**(**K**$_5$)	…	**H**(**K**$_n$)

Few-time signatures

HORS (Hash to Obtain Random Subset)
Reyzin, Reyzin, **2002**

To sign **M**, use a *selection function S*: **M** → indexes

For example, if *S*(**M**) = **{1, 5}** publish $\mathbf{K}_1$ and $\mathbf{K}_5$

	1	2	3	4	5	…	*n*
Private keys	$\mathbf{K}_1$	$\mathbf{K}_2$	$\mathbf{K}_3$	$\mathbf{K}_4$	$\mathbf{K}_5$	…	$\mathbf{K}_n$
	↓	↓	↓	↓	↓		↓
Public keys	$\mathbf{H}(\mathbf{K}_1)$	$\mathbf{H}(\mathbf{K}_2)$	$\mathbf{H}(\mathbf{K}_3)$	$\mathbf{H}(\mathbf{K}_4)$	$\mathbf{H}(\mathbf{K}_5)$	…	$\mathbf{H}(\mathbf{K}_n)$

Few-time signatures

HORS (Hash to Obtain Random Subset)
Reyzin, Reyzin, **2002**

To sign **M**, use a *selection function S*: **M** → indexes

Then if *S*(**M'**) = **{2, 3}** publish $\mathbf{K}_2$ and $\mathbf{K}_3$

	1	2	3	4	5	…	*n*
Private keys	$\mathbf{K}_1$	$\mathbf{K}_2$	$\mathbf{K}_3$	$\mathbf{K}_4$	$\mathbf{K}_5$	…	$\mathbf{K}_n$
	↓	↓	↓	↓	↓		↓
Public keys	$\mathbf{H}(\mathbf{K}_1)$	$\mathbf{H}(\mathbf{K}_2)$	$\mathbf{H}(\mathbf{K}_3)$	$\mathbf{H}(\mathbf{K}_4)$	$\mathbf{H}(\mathbf{K}_5)$	…	$\mathbf{H}(\mathbf{K}_n)$

Few-time signatures

HORS (Hash to Obtain Random Subset)
Reyzin, Reyzin, **2002**

To sign **M**, use a *selection function S*: **M** → indexes

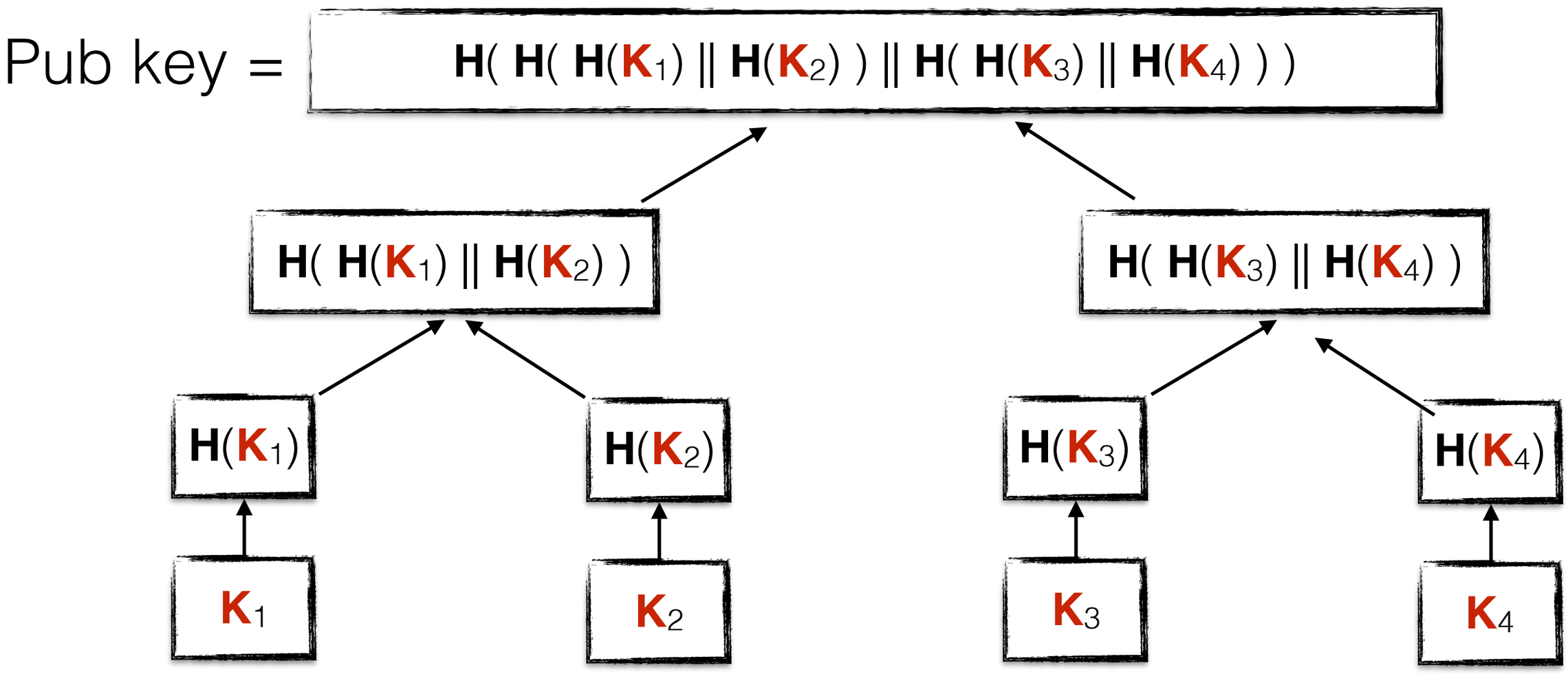
If too many messages signed, signatures guessable

	1	2	3	4	5	…	*n*
Private keys	$\mathbf{K}_1$	$\mathbf{K}_2$	$\mathbf{K}_3$	$\mathbf{K}_4$	$\mathbf{K}_5$	…	$\mathbf{K}_n$
	↓	↓	↓	↓	↓		↓
Public keys	$\mathbf{H}(\mathbf{K}_1)$	$\mathbf{H}(\mathbf{K}_2)$	$\mathbf{H}(\mathbf{K}_3)$	$\mathbf{H}(\mathbf{K}_4)$	$\mathbf{H}(\mathbf{K}_5)$	…	$\mathbf{H}(\mathbf{K}_n)$

Many-time signatures

Actually, just one-time signatures, but many keys…

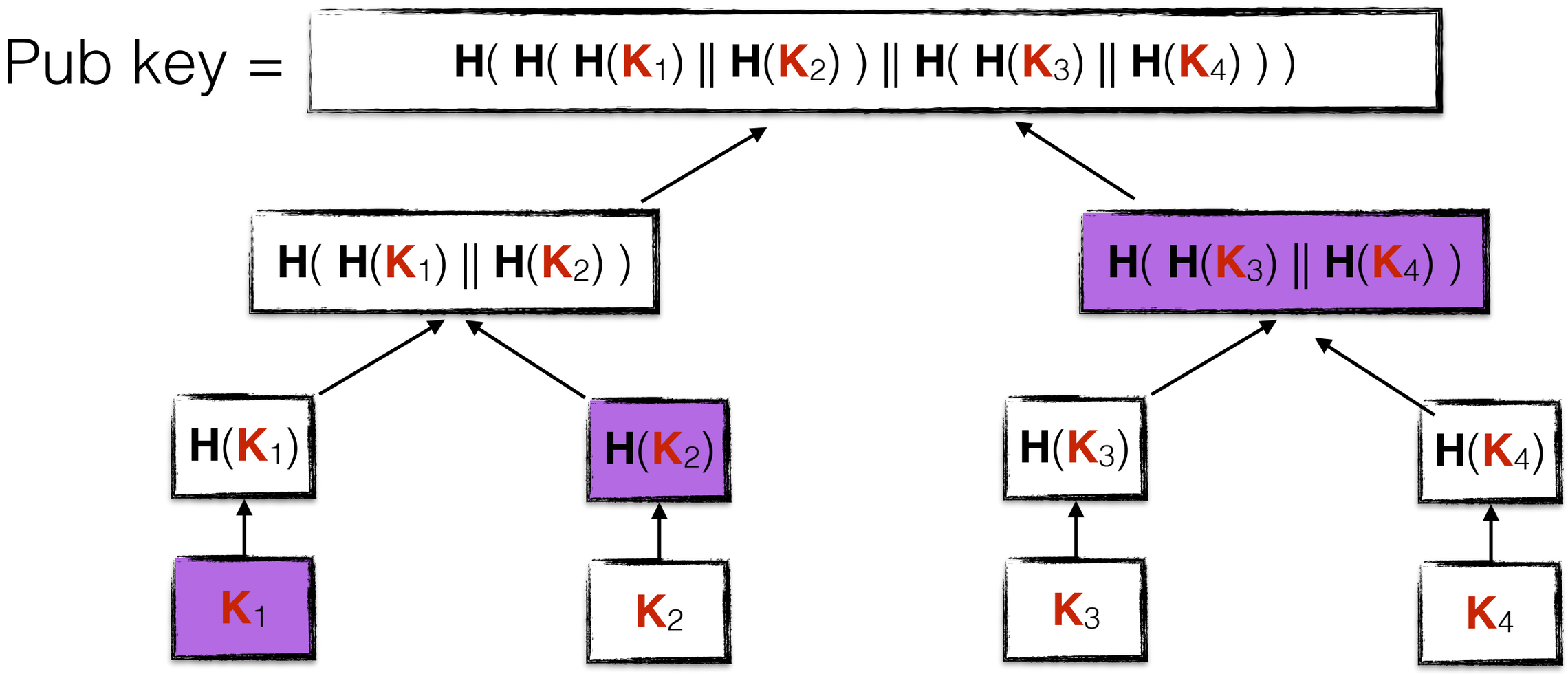
… represented in a compact way, using a **hash tree**



Many-time signatures

When a new **one-time public key K_i**, is used…

… give its **authentication path** to the root pub key



State-of-the-art schemes

- **XMSS**: custom hash trees and Winternitz signatures
- **SPHINCS**: similar plus HORS few-time signatures
  - Uses trees of trees (“hypertrees")
  - Avoids the need for a persistent state (**stateless**)

Conclusion

When/if a scalable and quantum computer is built…

- Public keys could be broken after some effort…
- Symmetric-key security will be at most halved
- We'll have NIST-standardized algorithms
  - Likely a lattice-based key agreement
  - Likely a hash-based signature scheme
  - ?

Thank you!