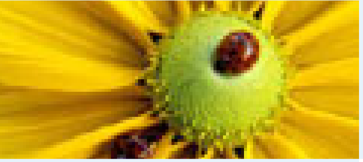
NIST's Post-Quantum Cryptography Project

René Peralta
Computer Security Division, NIST

NUTMIC, Warsaw
September 12, 2017

Quantum Computers

- It appears they are much more powerful than today's computers.
  but can we build them?.



Quantum Computers

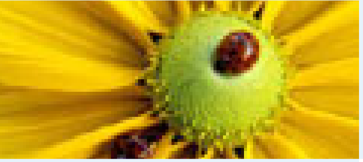
- It appears they are much more powerful than today's computers.
  but can we build them?.
- We think they will not be able to solve NP-hard problems efficiently.

Quantum Computers

- It appears they are much more powerful than today's computers.
  but can we build them?.

- We think they will not be able to solve NP-hard problems efficiently.

- They will be able to factor integers and solve discrete logarithms.

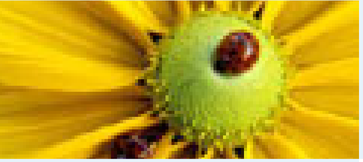


Quantum Computers

- It appears they are much more powerful than today's computers.
  but can we build them?.
- We think they will not be able to solve NP-hard problems efficiently.
- They will be able to factor integers and solve discrete logarithms.
- They will be able to invert functions asymptotically faster than classical computers.
  how much faster?

Implications for Crypto

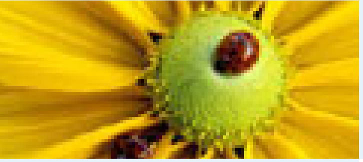
- RSA, Diffie-Hellman key exchange, elliptic curve crypto would be broken.



Implications for Crypto

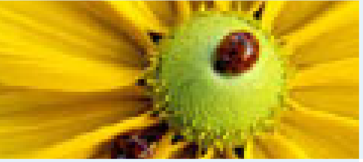
- RSA, Diffie-Hellman key exchange, elliptic curve crypto would be broken.

- Symmetric crypto will need longer keys.
  how much longer?



Is It Urgent?

- Full transition to alternatives takes a long time.
  maybe $>$ 10 years beyond the time of standardization



Is It Urgent?

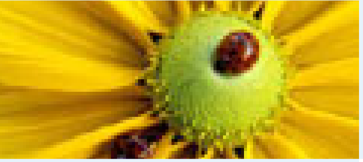
- Full transition to alternatives takes a long time.
  maybe > 10 years beyond the time of standardization

- Today's data needs to remain secure 5-10 years.
  longer in some cases, such as medical data

NIST's PQC Project

NIST's PQC Project

- To monitor progress in quantum computers and quantum algorithms.



NIST's PQC Project

- To monitor progress in quantum computers and quantum algorithms.
- To find and standardize quantum-resistant alternatives for PKE, key-agreement, and digital signatures.

NIST's PQC Project

- To monitor progress in quantum computers and quantum algorithms.
- To find and standardize quantum-resistant alternatives for PKE, key-agreement, and digital signatures.
- To ensure transparency of the process, community involvement, and legitimacy of the outcome.

This Process Is Not A Competition

This Process Is Not A Competition

- We hope at the end of the day there will be significant community consensus.

This Process Is Not A Competition

- We hope at the end of the day there will be significant community consensus.
- We may standardize several algorithms.

This Process Is Not A Competition

- We hope at the end of the day there will be significant community consensus.
- We may standardize several algorithms.
- The evaluation criteria is not set in stone, it may evolve during the next few years.

The Call For Proposals

- Candidate algorithms may now be submitted **here**.
- Deadline is November 30, 2017

The PQC Forum

- The wording of the CFP followed public discussion on the **pqc-forum** (email:pqc-forum@nist.gov).
- This is also where submissions and germane issues -such as evaluation criteria is discussed.
- To join send mail to pqc-forum-request@nist.gov with subject=subscribe.

Proposals Sought For

- Public-key encryption
- Key-encapsulation
- Digital signatures

Security

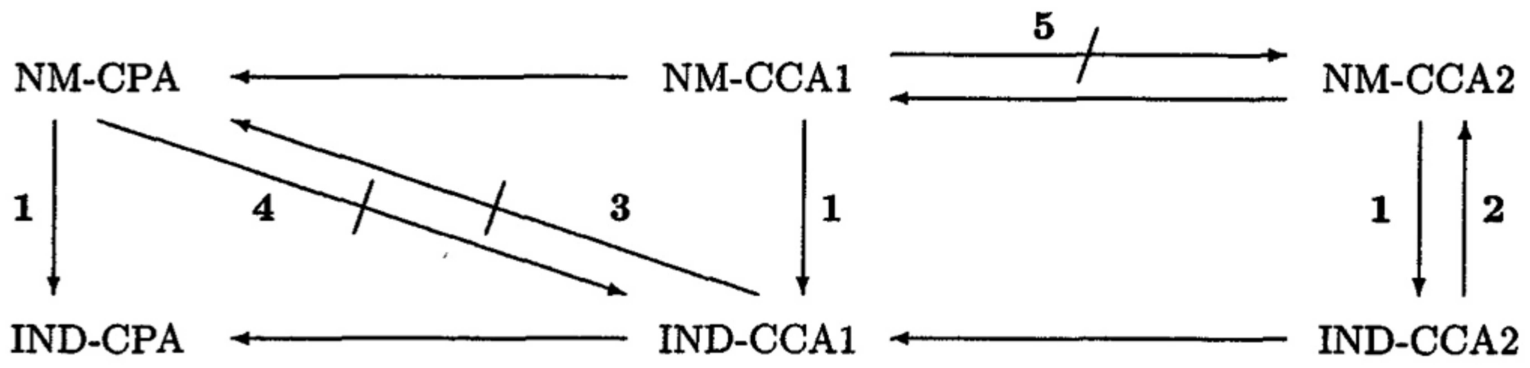


Figure 1: Current security definitions are precise and not easy to meet (drawing by Bellare).

Security

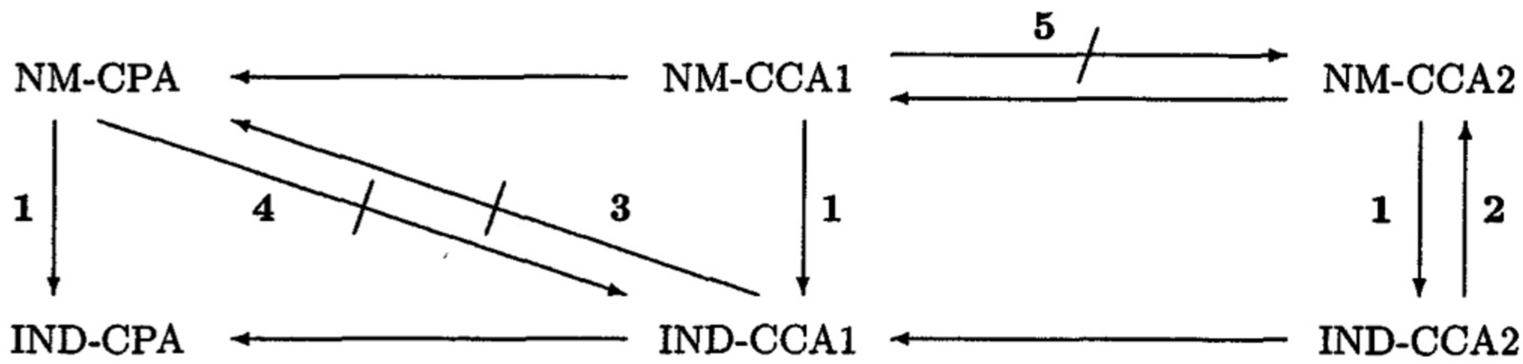


Figure 1: Current security definitions are precise and not easy to meet (drawing by Bellare).

These notions roughly say that encrypted messages can not be produced or even distinguished from random without the keys.

Security

Security

- Cryptanalysis: what are the best known attacks?

Security

- Cryptanalysis: what are the best known attacks?
- Foundations: do we believe an underlying primitive is hard for quantum computers?

Security

- Cryptanalysis: what are the best known attacks?
- Foundations: do we believe an underlying primitive is hard for quantum computers?

  In practice we are likely to see two assertions:

  - An underlying problem is hard for classical computers;
  - No clear quantum speedup beyond Grover's.

Security

- Cryptanalysis: what are the best known attacks?
- Foundations: do we believe an underlying primitive is hard for quantum computers?

  In practice we are likely to see two assertions:

  - An underlying problem is hard for classical computers;
  - No clear quantum speedup beyond Grover's.

- Do you actually have a reduction to a hard primitive?

Cost metrics

Cost metrics

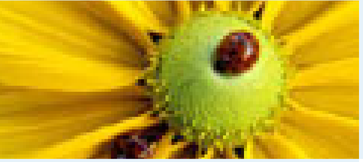
- Size of keys, time complexity

Cost metrics

- Size of keys, time complexity
- Memory

Cost metrics

- Size of keys, time complexity
- Memory
- Size of messages, size of signatures



Cost metrics

- Size of keys, time complexity
- Memory
- Size of messages, size of signatures
- Other resources (e.g. randomness, communication, interaction)

Cost metrics

- Size of keys, time complexity
- Memory
- Size of messages, size of signatures
- Other resources (e.g. randomness, communication, interaction)
- Set-up costs?

Cost metrics

- Size of keys, time complexity
- Memory
- Size of messages, size of signatures
- Other resources (e.g. randomness, communication, interaction)
- Set-up costs?
- How hard to protect from side-channel attacks?

How Things Look Like Now

How Things Look Like Now

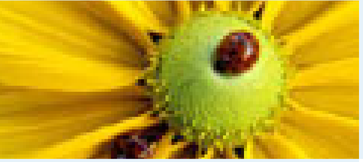
- Signatures: hash-based , code-based, lattice-based, multivariate ...

How Things Look Like Now

- Signatures: hash-based , code-based, lattice-based, multivariate ...
- PKE : lattice-based, code-based, multivariate, ...

How Things Look Like Now

- Signatures: hash-based , code-based, lattice-based, multivariate ...
- PKE : lattice-based, code-based, multivariate, ...
- Key agreement: PKE, lattice-based, isogeny-based, ...



How Things Look Like Now

- Signatures: hash-based , code-based, lattice-based, multivariate ...
- PKE : lattice-based, code-based, multivariate, ...
- Key agreement: PKE, lattice-based, isogeny-based, ...

This is not exhaustive.

Public Discussion

Public Discussion

- Much discussion regarding "security-levels" and derived parametrization, as well as which security notions are appropriate.

Public Discussion

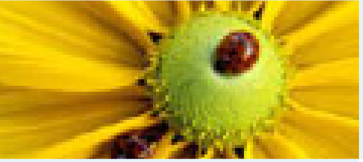
- Much discussion regarding “security-levels” and derived parametrization, as well as which security notions are appropriate.
- Suspicion that NIST is just doing NSA’s bidding.

Public Discussion

- Much discussion regarding "security-levels" and derived parametrization, as well as which security notions are appropriate.

- Suspicion that NIST is just doing NSA's bidding.

  This is a very public process. This is the current PQC team: Jacob Alperin-Sherif, Larry Basham, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone.



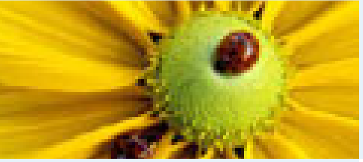
Public Discussion

- Much discussion regarding “security-levels” and derived parametrization, as well as which security notions are appropriate.

- Suspicion that NIST is just doing NSA’s bidding.

  This is a very public process. This is the current PQC team: Jacob Alperin-Sherif, Larry Basham, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone.

- Demands that future standards make bad implementations harder.



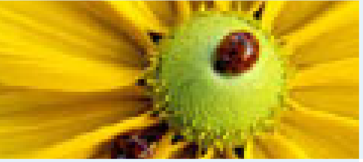
Timeline

- Dec 20, 2016 : Formal Call for Proposals.
- Nov 30 : Deadline for submissions.
- April 12-13 : Submitter's present their work at workshop in Ft. Lauderdale.
- next 3-5 years : Analysis phase - NIST will report findings in 1-2 workshops.
- 2 years later : Draft standards ready.

On A Bad Day

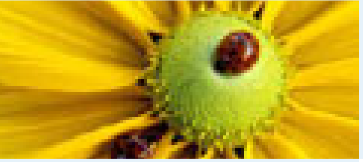
On A Bad Day

- NIST is needlessly rushing.



On A Bad Day

- NIST is needlessly rushing.
- Quantum-resistant cryptography is
  - New stuff.
  - Poorly understood.

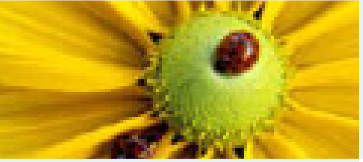


On A Bad Day

- NIST is needlessly rushing.
- Quantum-resistant cryptography is
  - New stuff.
  - Poorly understood.
- This is just too risky.

Predicting The Future

We always overestimate the change that will occur in the next two years and underestimate the change that will occur in the next ten. Don't let yourself be lulled into inaction.
(Bill Gates)



On A Good Day

On A Good Day

- Signatures: hash-based , **code-based, lattice-based, multivariate** ...
- PKE : **lattice-based, code-based, multivariate** , ...
- Key agreement: PKE, **lattice-based**, isogeny-based, ...

These have been around for a long time.

Multivariate

Do we know how to efficiently sample from hard distributions of multivariate equations?

Multivariate

Do we know how to efficiently sample from hard distributions of multivariate equations?

How about AES, or any of our hashing functions?

Multivariate

Do we know how to efficiently sample from hard distributions of multivariate equations?

How about AES, or any of our hashing functions?

These can be written as systems of quadratic and linear equations over GF2.

Multivariate

Do we know how to efficiently sample from hard distributions of multivariate equations?

How about AES, or any of our hashing functions?

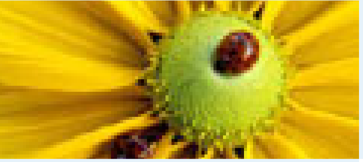
These can be written as systems of quadratic and linear equations over GF2.

multiplicative-complexity problem: minimize the number of quadratic equations. Boyar,Courtois and others have done work on this. There was a meeting on multiplicative complexity co-located with Eurocrypt 2017.

Lattice-based And Code-based

Lattice-based And Code-based

- 1978 Merkle-Hellman knapsack cryptosystem.
  Failed attempt to create a trapdoor cryptosystem based on an NP-Hard problem.



Lattice-based And Code-based

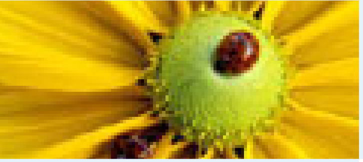
- 1978 Merkle-Hellman knapsack cryptosystem.
  Failed attempt to create a trapdoor cryptosystem based on an NP-Hard problem.

- 1978 MacEliece.
  System $Ax \approx y$ where A is a matrix and x, y are vectors, all over GF2.

Lattice-based And Code-based

- 1978 Merkle-Hellman knapsack cryptosystem.
  Failed attempt to create a trapdoor cryptosystem based on an NP-Hard problem.

- 1978 MacEliece.
  System $Ax \approx y$ where A is a matrix and x, y are vectors, all over GF2.
  $u \approx v$ means $u + v$ has low hamming weight.



Lattice-based And Code-based

- 1978 Merkle-Hellman knapsack cryptosystem.
  Failed attempt to create a trapdoor cryptosystem based on an NP-Hard problem.

- 1978 MacEliece.
  System $Ax \approx y$ where A is a matrix and x, y are vectors, all over GF2.
  $u \approx v$ means $u + v$ has low hamming weight.

  More precisely: given matrix A and vector y find vector x such that $Ax + y$ has low hamming weight.

Lattice-based And Code-based

- 1978 Merkle-Hellman knapsack cryptosystem.
  Failed attempt to create a trapdoor cryptosystem based on an NP-Hard problem.

- 1978 MacEliece.
  System $Ax \approx y$ where A is a matrix and x, y are vectors, all over GF2.
  $u \approx v$ means $u + v$ has low hamming weight.

  More precisely: given matrix A and vector y find vector x such that $Ax + y$ has low hamming weight.

  THIS QUANTUM-RESISTANT ENCRYPTION SYSTEM REMAINS UNBROKEN

Lattice-based And Code-based

- 1982 LLL.

Lattice-based And Code-based

- 1982 LLL.

  Even exponentially bad approximation to the shortest vector problem on lattices is a powerful tool.

  With LLL we can factor polynomials over Q.

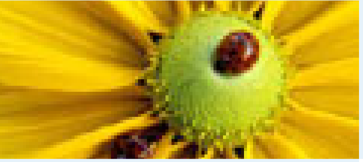
Lattice-based And Code-based

- 1982 LLL.

  Even exponentially bad approximation to the shortest vector problem on lattices is a powerful tool.

  With LLL we can factor polynomials over Q.

  Flip side: short vectors on a lattice are likely to be hard.



Lattice-based And Code-based

- 1996-2016 NTRU system
  20 years tinkering, latest is NTRU Prime (Bernstein et al).

Lattice-based And Code-based

- 1996-2016 NTRU system
  20 years tinkering, latest is NTRU Prime (Bernstein et al).
- 1996 Ajtai.
  How to generate hard instances of lattice problems.

Lattice-based And Code-based

- 1996-2016 NTRU system
  20 years tinkering, latest is NTRU Prime (Bernstein et al).

- 1996 Ajtai.
  How to generate hard instances of lattice problems.

- 1997 Ajtai, Dwork.
  A public-key cryptosystem in which average case is as bad as the worst case.

Lattice-based And Code-based

- 2005 Regev
  Learning with errors (LWE).

Lattice-based And Code-based

- 2005 Regev
  Learning with errors (LWE).
  System $As \approx y$ where A is a matrix and s, y are vectors, all modulo q.

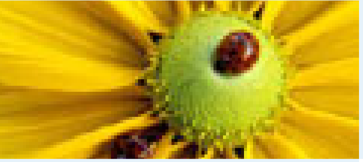
Lattice-based And Code-based

- 2005 Regev
  Learning with errors (LWE).
  System $As \approx y$ where A is a matrix and s, y are vectors, all modulo q.

  $u \approx v$ means all elements of $u - v$ are close to 0 modulo q.

Lattice-based And Code-based

- 2002-2006 Lyubashevsky, Micciancio, Peikert, Rosen , and others ...
  Ring-SIS is hard (SIS = Small Integer Solution)



Lattice-based And Code-based

- 2002-2006 Lyubashevsky, Micciancio, Peikert, Rosen , and others ...
  Ring-SIS is hard (SIS = Small Integer Solution)
- 2009 Gentry
  Fully homomorphic encryption.

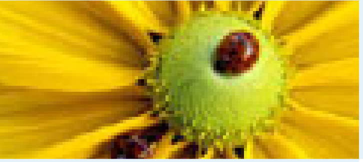
Lattice-based And Code-based

- 2002-2006 Lyubashevsky, Micciancio, Peikert, Rosen , and others ...
  Ring-SIS is hard (SIS = Small Integer Solution)
- 2009 Gentry
  Fully homomorphic encryption.
- TODAY
  - Ring-LWE on your Chrome browser.
  - Frodo , efficient key-exchange using LWE.

Lattice-based And Code-based

- 2002-2006 Lyubashevsky, Micciancio, Peikert, Rosen , and others ...
  Ring-SIS is hard (SIS = Small Integer Solution)
- 2009 Gentry
  Fully homomorphic encryption.
- TODAY
  - Ring-LWE on your Chrome browser.
  - Frodo , efficient key-exchange using LWE.

This just skims the surface.

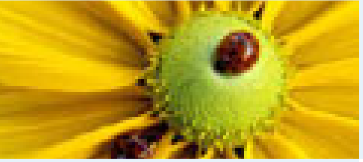


Lattice-based And Code-based

- 2002-2006 Lyubashevsky, Micciancio, Peikert, Rosen , and others ...
  Ring-SIS is hard (SIS = Small Integer Solution)
- 2009 Gentry
  Fully homomorphic encryption.
- TODAY
  - Ring-LWE on your Chrome browser.
  - Frodo , efficient key-exchange using LWE.

This just skims the surface.

It looks like quite a bit of history to me.



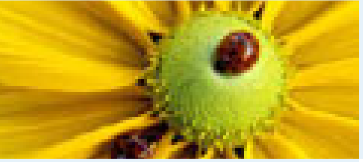
Lattice-based And Code-based

- 2002-2006 Lyubashevsky, Micciancio, Peikert, Rosen , and others ...
  Ring-SIS is hard (SIS = Small Integer Solution)
- 2009 Gentry
  Fully homomorphic encryption.
- TODAY
  - Ring-LWE on your Chrome browser.
  - Frodo , efficient key-exchange using LWE.

This just skims the surface.

It looks like quite a bit of history to me.

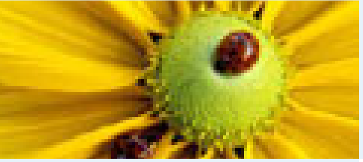
(sources are Wikipedia, memory, my NIST colleagues, and a 2010 survey by Regev.)



Estimate

In 15 years, we may be able to factor a 2Kb RSA number in about a day, using a dedicated nuclear power plant.

(Mariantoni, PQCrypto 2014)



Estimate

In 15 years, we may be able to factor a 2Kb RSA number in about a day, using a dedicated nuclear power plant.

(Mariantoni, PQCrypto 2014)

THANK YOU