Post-Quantum Cryptography & Privacy

Andreas Hülsing
Privacy?

... the Panopticon must not be understood as a dream building: it is the diagram of a mechanism of power reduced to its ideal form.

Michel Foucault, *Discipline and Punish*, 1977
How to achieve privacy?
Under the hood...

Asymmetric Crypto
• ECC
• RSA
• DSA

Symmetric Crypto
• AES
• SHA2
• SHA1
• ...

Combination of both needed!
We need symmetric and asymmetric crypto to achieve privacy!
Quantum Computing
Quantum Computing

“Quantum computing studies theoretical computation systems (quantum computers) that make direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data.”

-- Wikipedia
Qubits

- Qubit state: $\alpha_0 |0\rangle + \alpha_1 |1\rangle$ with $\alpha_i \in \mathbb{C}$ such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$
- Ket: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- Qubit can be in state $\frac{|0\rangle+|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
- Computing with 0 and 1 at the same time!
Quantum computers are not almighty

• To learn outcome one has to measure.
  • Collapses state
  • 1 qubit leads 1 classical bit of information
  • Randomized process

• Only invertible computation.

• Impossible to clone (copy) quantum state.
The Quantum Threat
Shor’s algorithm (1994)

• Quantum computers can do FFT very efficiently
• Can be used to find period of a function
• This can be exploited to factor efficiently (RSA)
• Shor also shows how to solve discrete log efficiently (DSA, DH, ECDSA, ECDH)
Grover’s algorithm (1996)

- Quantum computers can search $N$ entry DB in $\Theta(\sqrt{N})$
- Application to symmetric crypto
- Nice: Grover is provably optimal (For random function)
- Double security parameter.
To sum up

• All asymmetric crypto is broken by QC
  • No more digital signatures
  • No more public key encryption
  • No more key exchange

• Symmetric crypto survives
  (with doubled key size / output length)
  • NOT ENOUGH!
Why care today?
Quantum Computing

“Quantum computing studies theoretical computation systems (quantum computers) that make direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data.”

-- Wikipedia
Bad news

I will not tell you when a quantum computer will be built!
Europe plans giant billion-euro quantum technologies project

Third European Union flagship will be similar in size and ambition to graphene and human brain initiatives.

Elizabeth Gibney
It’s a question of risk assessment
How soon do we need to worry?

Depends on:
- How long do you need your keys to be secure? \((x \text{ years})\)
- How much time will it take to re-tool the existing infrastructure with large-scale quantum-safe solution? \((y \text{ years})\)
- How long will it take for a large-scale quantum computer to be built (or for any other relevant advance)? \((z \text{ years})\)

Theorem 1: If \(x + y > z\), then worry.

What do we do here??

\[
\begin{array}{c}
\text{Secret keys revealed} \\
\text{time}
\end{array}
\]
Who would store all encrypted data traffic? That must be expensive!
Quantum Cryptography
Why not beat ‘em with their own weapons?

• QKD: Quantum Key distribution.
  • Based on some nice quantum properties: entanglement & collapsing measurements
  • Information theoretic security (at least in theory)
    -> Great!
  • For sale today!

• So why don’t we use this?

• Only short distance, point-to-point connections!
  • Internet? No way!

• Longer distances require „trusted-repeaters“ 😊
  • We all know where this leads...
PQCRYPTO to the rescue
Quantum-secure problems

No provably quantum resistant problems

We must look here

Bounded-Error Quantum Polynomial-Time

NP-complete

NP

Factoring

BQP

P

Credits: Buchmann, Bindel 2015
Conjectured quantum-secure problems

• Solving multivariate quadratic equations (MQ-problem)
  -> Multivariate Crypto

• Bounded-distance decoding (BDD)
  -> Code-based crypto

• Short(est) and close(st) vector problem (SVP, CVP)
  -> Lattice-based crypto

• Breaking security of symmetric primitives (SHAx-, AES-, Keccak-,... problem)
  -> Hash-based signatures / symmetric crypto
Multivariate Crypto

\[
4x + x^2 + y^2z \equiv 1 \mod 13
\]
\[
7y^2 + 2xz^2 \equiv 12 \mod 13
\]
\[
x + y^2 + 12xz^2 \equiv 4 \mod 13
\]

Solution: \( x = 15, \ y = 29, \ z = 45 \)

Credits: Buchmann, Bindel 2015
MQ-Problem

Let \( x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n \) and \( \text{MQ}(n, m, \mathbb{F}_q) \) denote the family of vectorial functions \( F: \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m \) of degree 2 over \( \mathbb{F}_q \):

\[
\text{MQ}(n, m, \mathbb{F}_q) = \left\{ F(x) = (f_1(x), \ldots, f_m(x)) \mid f_s(x) = \sum_{i,j} a_{i,j} x_i x_j + \sum_i b_i x_i, \quad s \in [1, m] \right\}
\]

The MQ Problem \( \text{MQ}(F, \nu) \) is defined as given \( \nu \in \mathbb{F}_q^m \) find, if any, \( s \in \mathbb{F}_q^n \) such that \( F(s) = \nu \).

Decisional version is NP-complete [Garey, Johnson´79]
Multivariate Signatures

\[ P : F^n \rightarrow F^m, \text{ easily invertible non-linear} \]
\[ S : F^n \rightarrow F^n, \ T : F^m \rightarrow F^m, \text{ affine linear} \]

Public key: \[ G = S \circ P \circ T, \text{ hard to invert} \]

Secret Key: \[ S, P, T \text{ allows to find } G^{-1} \]
\[ G^{-1} = T^{-1} \circ P^{-1} \circ S^{-1} \]

Signing: \[ s = T^{-1} \circ P^{-1} \circ S^{-1}(m) \]

Verifying: \[ G(s) = ? m \]

Forging signature: Solve \[ G(s) - m = 0 \]

Credits: Buchmann, Bindel 2015
Multivariate Cryptography

• Breaking scheme ⇔ Solving MQ-Problem
  -> Not a random instance
  -> Not NP-hard (there might be easy instances)
  -> New proposal with security reduction, small keys, but large signatures.

• Many broken proposals
  -> Oil-and-Vinegar, SFLASH, MQQ-Sig, (Enhanced) TTS, Enhanced STS.
  -> Security somewhat unclear

• Only signatures
  -> (new proposal for encryption exists but too recent)

• Really large keys
Coding-based cryptography - BDD

Given:
• Linear code \( C \subseteq F_2^n \)
• \( y \in F_2^n \)
• \( t \in \mathbb{N} \)

Find:
• \( x \in C: \text{dist}(x, y) \leq t \)

BDD is NP-complete (Berlekamp et al. 1978) (Decisional version)

Credits: Buchmann, Bindel 2015
McEliece PKE (1978)

S, G, P matrices over F

G generator matrix for Goppa code

Public key: \( G' = S \circ G \circ P, t \)

Secret Key: P, S, G

Encryption: \( c = mG' + z \in F^n \)

Decryption: \( x = cP^{-1} = mSG + zP^{-1} \)

solve BDD to get \( y = mSG \)

decode to obtain m

Fast

Large public keys!
500 kBits for 100 bit security
Compared to 1776 bit RSA modulus

IND-CPA secure version

Credits: Buchmann, Bindel 2015
Code-based cryptography

• Breaking scheme ↔ Solving BDD
  -> Not a random instance
  -> Not NP-hard (there might be easy instances)

• However, McEliece with binary Goppa codes survived for almost 40 years (similar situation as for e.g. AES)

• Using more compact codes often leads to break
• So far, no practical signature scheme

• Really large public keys
Lattice-based cryptography

Basis: \( B = (b_1, b_2) \in \mathbb{Z}^{2 \times 2}; b_1, b_2 \in \mathbb{Z}^2 \)

Lattice: \( \Lambda(B) = \{x = By \mid y \in \mathbb{Z}^2\} \)
Shortest vector problem (SVP)
(Worst-case) Lattice Problems

- **SVP**: Find shortest vector in lattice, given random basis. NP-hard (Ajtai’96)

- **Approximate SVP (αSVP)**: Find short vector (norm < α times norm of shortest vector). Hardness depends on α (for α used in crypto not NP-hard).

- **CVP**: Given random point in underlying vectorspace (e.g. \(\mathbb{Z}^n\)), find the closest lattice point. (Generalization of SVP, reduction from SVP)

- **Approximate CVP (αCVP)**: Find a „close“ lattice point. (Generalization of αSVP)
(Average-case) Lattice Problems
Short Integer Solution (SIS)

\[ \mathbb{Z}_p^n = \text{n-dim. vectors with entries mod } p \ (\approx n^3) \]

Goal:
Given \( A = (a_1, a_2, \ldots, a_m) \in \mathbb{Z}_p^{n \times m} \)
Find "small" \( s = (s_1, \ldots, s_m) \in \mathbb{Z}^m \) such that

\[ As = 0 \mod p \]

Reduction from worst-case \( \alpha \text{SVP} \).
Hash function

Set $m > n \log p$ and define $f_A: \{0,1\}^m \rightarrow \mathbb{Z}_p^n$ as

$$f_A(x) = Ax \mod p$$

**Collision-resistance:** Given short $x_1, x_2$ with $Ax_1 = Ax_2$ we can find a short solution as

$$Ax_1 = Ax_2 \Rightarrow Ax_1 - Ax_2 = 0$$

$$A(x_1 - x_2) = 0$$

So, $z = x_1 - x_2$ is a solution and it is short as $x_1, x_2$ are short.
Lattice-based crypto

- SIS: Allows to construct signature schemes, hash functions, ... , basically minicrypt.
- For more advanced applications: Learning with errors (LWE)
  - Allows to build PKE, IBE, FHE,...
- Performance: Sizes can almost reach those of RSA (just small const. factor), really fast (for lattices defined using polynomials).
- BUT: Exact security not well accessed, yet. Especially, no good estimate for quantum computer aided attacks.
Hash-based Signature Schemes

[Mer89]

- Post quantum
- Only secure hash function
- Security well understood
- Fast
RSA – DSA – EC-DSA...

Intractability Assumption

Cryptographic hash function

Digital signature scheme

RSA, DH, SVP, MQ, …
Merkle’s Hash-based Signatures

SIG = (i=2, , , , )
Hash-based signatures

- Only signatures
- Minimal security assumptions
- Well understood
- Fast & compact (2kB, few ms), but stateful, or
- Stateless, bigger and slower (41kB, several ms).
Initial recommendations

- **Symmetric encryption** Thoroughly analyzed, 256-bit keys:
  - AES-256
  - Salsa20 with a 256-bit key

Evaluating: Serpent-256, ...

- **Symmetric authentication** Information-theoretic MACs:
  - GCM using a 96-bit nonce and a 128-bit authenticator
  - Poly1305

- **Public-key encryption** Efficiency with binary Goppa codes:
  - length $n = 66$, dimension $k = 5413$, $t = 119$ errors

Evaluating: MDPC, Stehlé-Steinfeld NTRU, ...

- **Public-key signatures** Hash-based (minimal assumptions):
  - XMSS with any of the parameters specified in CFRG draft
  - SPHINCS-256

Evaluating: HFEv-, ...

Confidence inspiring solutions are slow, too big, ...
TODOs

• Increase confidence for other schemes: (Quantum) cryptanalysis

• Improve existing schemes

• Create code-base

Basis for standards, certification, ... , deployment
Thank you!
Questions?