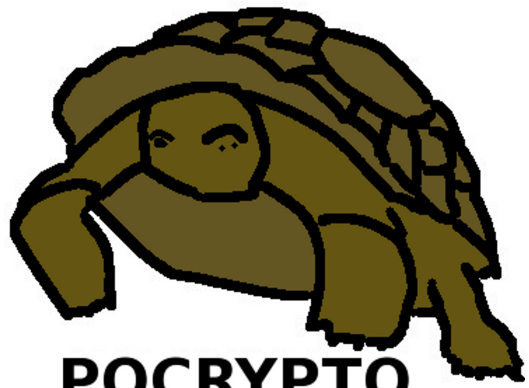
Post-quantum crypto and side-channel attacks – getting ready for the real world

Tanja Lange



21 Jul 2016

ArcticCrypt

History of post-quantum cryptography

- 2003 Daniel J. Bernstein introduces term Post-quantum cryptography.
- PQCrypto 2006: International Workshop on Post-Quantum Cryptography.

History of post-quantum cryptography

- 2003 Daniel J. Bernstein introduces term Post-quantum cryptography.
- PQCrypto 2006: International Workshop on Post-Quantum Cryptography.
- PQCrypto 2008, PQCrypto 2010, PQCrypto 2011, PQCrypto 2013.
- 2014 EU publishes H2020 call including post-quantum crypto as topic.
- PQCrypto 2014.
- April 2015 NIST hosts first workshop on post-quantum cryptography
- August 2015 NSA wakes up

NSA announcements

August 11, 2015

IAD recognizes that there will be a move, in the not distant future, to a quantum resistant algorithm suite.

NSA announcements

August 11, 2015

> *IAD recognizes that there will be a move, in the not distant future, to a quantum resistant algorithm suite.*

August 19, 2015

> *IAD will initiate a transition to quantum resistant algorithms in the not too distant future.*

NSA comes late to the party and botches its grand entrance.

NSA announcements

August 11, 2015

> *IAD recognizes that there will be a move, in the not distant future, to a quantum resistant algorithm suite.*

August 19, 2015

> *IAD will initiate a transition to quantum resistant algorithms in the not too distant future.*

NSA comes late to the party and botches its grand entrance.

Worse, now we get people saying "Don't use post-quantum crypto, the NSA wants you to use it!".

NSA announcements

August 11, 2015

> *IAD recognizes that there will be a move, in the not distant future, to a quantum resistant algorithm suite.*

August 19, 2015

> *IAD will initiate a transition to quantum resistant algorithms in the not too distant future.*

NSA comes late to the party and botches its grand entrance.

Worse, now we get people saying "Don't use post-quantum crypto, the NSA wants you to use it!". Or "NSA says NIST P-384 is post-quantum secure".

NSA announcements

August 11, 2015

> *IAD recognizes that there will be a move, in the not distant future, to a quantum resistant algorithm suite.*

August 19, 2015

> *IAD will initiate a transition to quantum resistant algorithms in the not too distant future.*

NSA comes late to the party and botches its grand entrance.

Worse, now we get people saying "Don't use post-quantum crypto, the NSA wants you to use it!". Or "NSA says NIST P-384 is post-quantum secure". Or "NSA has abandoned ECC."

Post-quantum becoming mainstream

- PQCrypto 2016: 22–26 Feb in Fukuoka, Japan, with more than 200 participants

- NIST is calling for post-quantum proposals; expect a small competition.
- PQCrypto 2017, Netherlands:
  - Jun 19 – 23 PQC school; Jun 22 & 23 Executive school
  - Jun 26 – 28 PQCrypto

In the long term, all encryption needs to be post-quantum

- Mark Ketchen, IBM Research, 2012, on quantum computing: "Were actually doing things that are making us think like, 'hey this isn't 50 years off, this is maybe just 10 years off, or 15 years off.' It's within reach."

In the long term, all encryption needs to be post-quantum

- Mark Ketchen, IBM Research, 2012, on quantum computing: "Were actually doing things that are making us think like, 'hey this isn't 50 years off, this is maybe just 10 years off, or 15 years off.' It's within reach."
- Fast-forward to 2022, or 2027. Quantum computers exist.
- Shor's algorithm solves in polynomial time:
  - Integer factorization.
  - The discrete-logarithm problem in finite fields.
  - The discrete-logarithm problem on elliptic curves.
- This breaks all current public-key encryption on the Internet!

In the long term, all encryption needs to be post-quantum

- Mark Ketchen, IBM Research, 2012, on quantum computing: "Were actually doing things that are making us think like, 'hey this isn't 50 years off, this is maybe just 10 years off, or 15 years off.' It's within reach."
- Fast-forward to 2022, or 2027. Quantum computers exist.
- Shor's algorithm solves in polynomial time:
  - Integer factorization.
  - The discrete-logarithm problem in finite fields.
  - The discrete-logarithm problem on elliptic curves.
- This breaks all current public-key encryption on the Internet!
- Also, Grover's algorithm speeds up brute-force searches.
- Example: Only 2^{64} quantum operations to break AES-128.

In the long term, all encryption needs to be post-quantum

- Mark Ketchen, IBM Research, 2012, on quantum computing: "Were actually doing things that are making us think like, 'hey this isn't 50 years off, this is maybe just 10 years off, or 15 years off.' It's within reach."
- Fast-forward to 2022, or 2027. Quantum computers exist.
- Shor's algorithm solves in polynomial time:
  - Integer factorization.
  - The discrete-logarithm problem in finite fields.
  - The discrete-logarithm problem on elliptic curves.
- This breaks all current public-key encryption on the Internet!
- Also, Grover's algorithm speeds up brute-force searches.
- Example: Only 2^{64} quantum operations to break AES-128.
- Need to switch the Internet to post-quantum encryption.

Confidence-inspiring crypto takes time to build

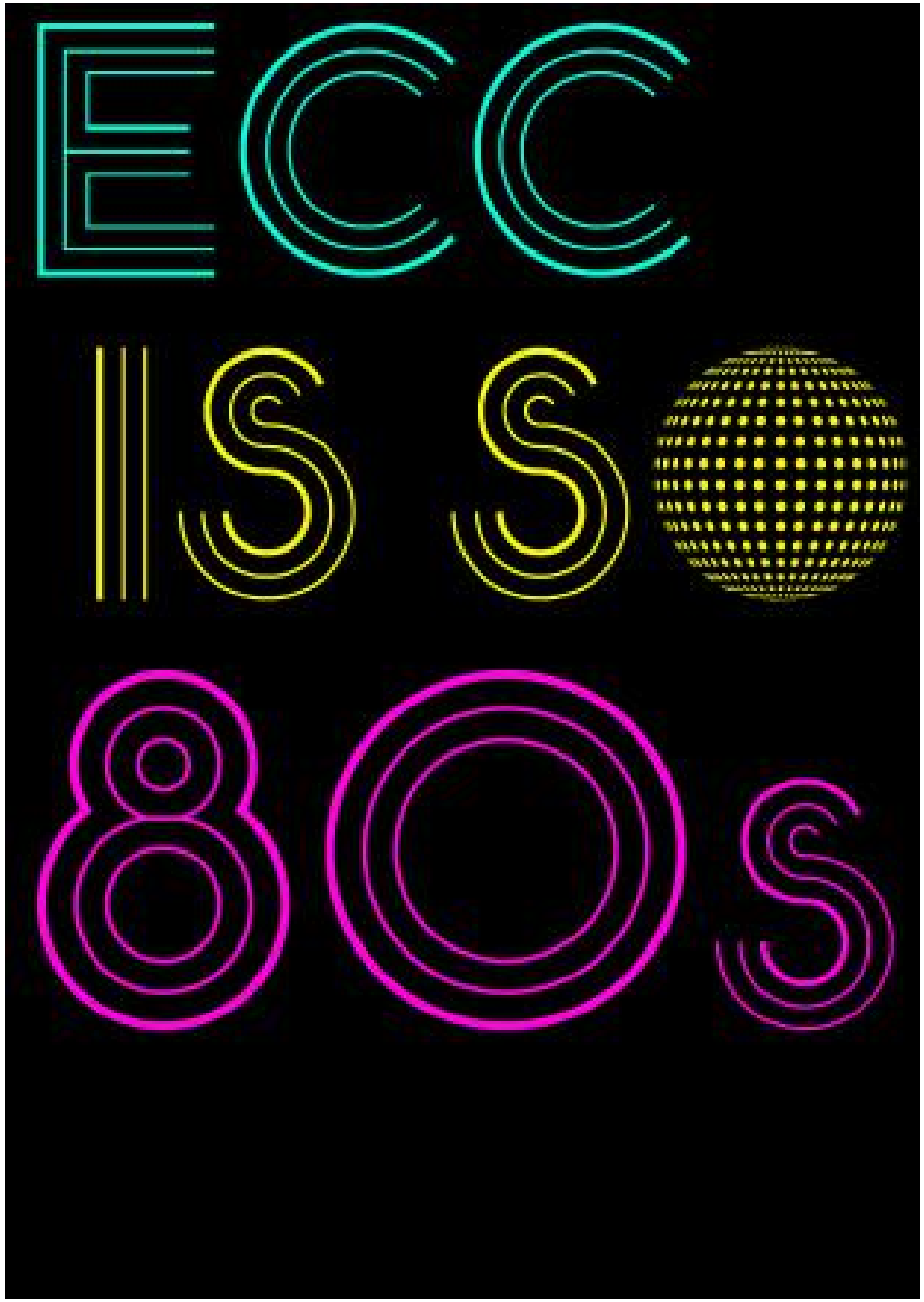
- Many stages of research from cryptographic design to deployment:
  - Explore space of cryptosystems.
  - Study algorithms for the attackers.
  - Focus on secure cryptosystems.

Confidence-inspiring crypto takes time to build

- Many stages of research from cryptographic design to deployment:
  - Explore space of cryptosystems.
  - Study algorithms for the attackers.
  - Focus on secure cryptosystems.
  - Study algorithms for the users.
  - Study implementations on real hardware.
  - Study side-channel attacks, fault attacks, etc.
  - Focus on secure, reliable implementations.
  - Focus on implementations meeting performance requirements.
  - Integrate securely into real-world applications.

Confidence-inspiring crypto takes time to build

- Many stages of research from cryptographic design to deployment:
  - Explore space of cryptosystems.
  - Study algorithms for the attackers.
  - Focus on secure cryptosystems.
  - Study algorithms for the users.
  - Study implementations on real hardware.
  - Study side-channel attacks, fault attacks, etc.
  - Focus on secure, reliable implementations.
  - Focus on implementations meeting performance requirements.
  - Integrate securely into real-world applications.
- Example: ECC introduced **1985**; big advantages over RSA. Robust ECC is starting to take over the Internet in **2015**.
- Post-quantum research can't wait for quantum computers!


ECC
IS
80s

Even higher urgency for long-term confidentiality

- Today's encrypted communication can be (and is being!) stored by attackers and can be decrypted later with quantum computer – think of medical records, legal proceedings, and state secrets.

- Post-quantum secure cryptosystems exist (to the best of our knowledge) but are under-researched – we can recommend secure systems now, but they are big and slow

Even higher urgency for long-term confidentiality

- Today's encrypted communication can be (and is being!) stored by attackers and can be decrypted later with quantum computer – think of medical records, legal proceedings, and state secrets.

- Post-quantum secure cryptosystems exist (to the best of our knowledge) but are under-researched – we can recommend secure systems now, but they are big and slow hence the logo of the PQCRYPTO project.
- PQCRYPTO is an EU project in H2020, running 2015 – 2018.

Standardize now? Standardize later?

- Standardize now!
  - Rolling out crypto takes long time.
  - Standards are important for adoption (?)
  - Need to be up & running when quantum computers come.

Standardize now? Standardize later?

- Standardize now!
  - Rolling out crypto takes long time.
  - Standards are important for adoption (?)
  - Need to be up & running when quantum computers come.
- Standardize later!
  - Current options are not satisfactory.
  - Once rolled out, it's hard to change systems.
  - Please wait for the research results, will be much better!

Standardize now? Standardize later?

- Standardize now!
  - Rolling out crypto takes long time.
  - Standards are important for adoption (?)
  - Need to be up & running when quantum computers come.
- Standardize later!
  - Current options are not satisfactory.
  - Once rolled out, it's hard to change systems.
  - Please wait for the research results, will be much better!
- But what about users who rely on long-term secrecy of today's communication?
- Recommend now, standardize later.
- Recommend very conservative systems now; users who care will accept performance issues and gladly update to faster/smaller options later.
- But: standardization takes lots of time to even start. Important to raise awareness.

Initial recommendations of long-term secure post-quantum systems

Daniel Augot, Lejla Batina, Daniel J. Bernstein, Joppe Bos, Johannes Buchmann, Wouter Castryck, Orr Dunkelman, Tim Güneysu, Shay Gueron, Andreas Hülsing, Tanja Lange, Mohamed Saied Emam Mohamed, Christian Rechberger, Peter Schwabe, Nicolas Sendrier, Frederik Vercauteren, Bo-Yin Yang

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** McEliece with binary Goppa codes:
  - length $n = 6960$, dimension $k = 5413$, $t = 119$ errors

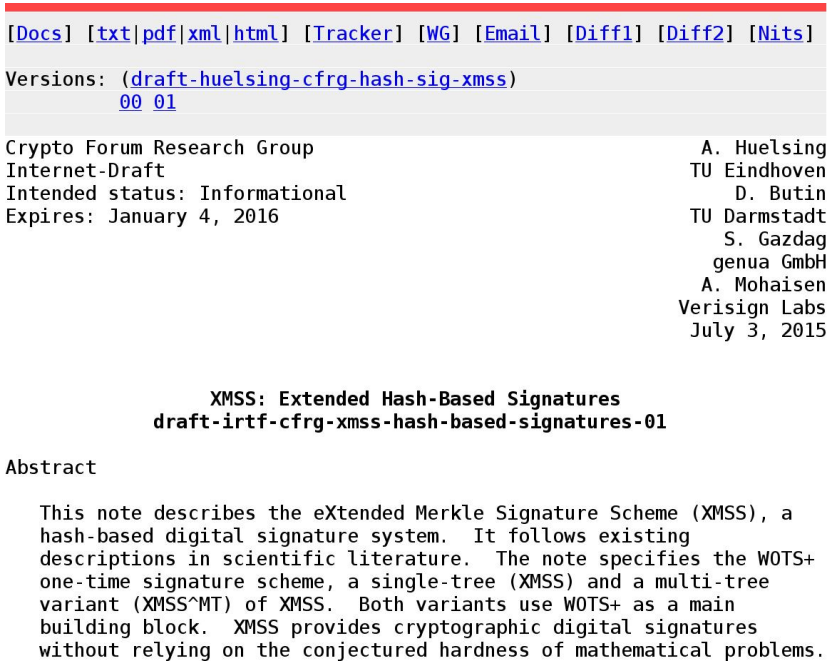
  Evaluating: QC-MDPC, Stehlé-Steinfeld NTRU, ...
- **Public-key signatures** Hash-based (minimal assumptions):
  - XMSS with any of the parameters specified in CFRG draft
  - SPHINCS-256

  Evaluating: HFEv-, ...

Hash-based signatures

Pros:

- Post quantum
- Only need secure hash function, e.g. SHA3-512, ...
- Need signatures anyways.
- Small public key
- Security well understood
- Fast
- Proposed for standards: https://tools.ietf.org/html/draft-irtf-cfrg-xmss-hash-based-signatures-01



[Docs] [txt|pdf|xml|html] [Tracker] [WG] [Email] [Diff1] [Diff2] [Nits]

Versions: (draft-huelsing-cfrg-hash-sig-xmss)
00 01

Crypto Forum Research Group
Internet-Draft
Intended status: Informational
Expires: January 4, 2016

A. Huelsing
TU Eindhoven
D. Butin
TU Darmstadt
S. Gazdag
genua GmbH
A. Mohaisen
Verisign Labs
July 3, 2015

XMSS: Extended Hash-Based Signatures
draft-irtf-cfrg-xmss-hash-based-signatures-01

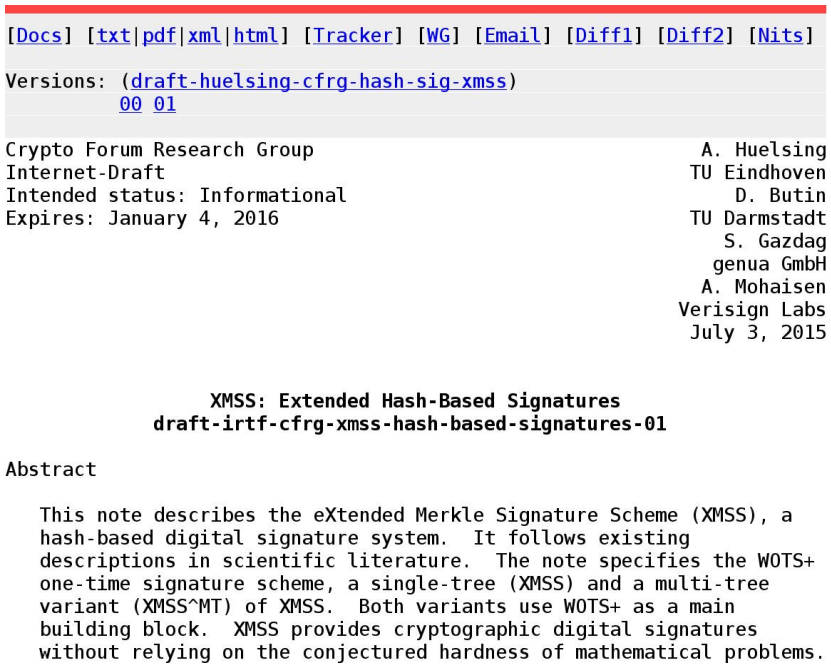
Abstract

This note describes the eXtended Merkle Signature Scheme (XMSS), a hash-based digital signature system. It follows existing descriptions in scientific literature. The note specifies the WOTS+ one-time signature scheme, a single-tree (XMSS) and a multi-tree variant (XMSS^MT) of XMSS. Both variants use WOTS+ as a main building block. XMSS provides cryptographic digital signatures without relying on the conjectured hardness of mathematical problems.

Hash-based signatures

Pros:

- Post quantum
- Only need secure hash function, e.g. SHA3-512, ...
- Need signatures anyways.
- Small public key
- Security well understood
- Fast
- Proposed for standards: https://tools.ietf.org/html/draft-irtf-cfrg-xmss-hash-based-signatures-01



[Docs] [txt|pdf|xml|html] [Tracker] [WG] [Email] [Diff1] [Diff2] [Nits]

Versions: (draft-huelsing-cfrg-hash-sig-xmss)
00 01

Crypto Forum Research Group
Internet-Draft
Intended status: Informational
Expires: January 4, 2016

A. Huelsing
TU Eindhoven
D. Butin
TU Darmstadt
S. Gazdag
genua GmbH
A. Mohaisen
Verisign Labs
July 3, 2015

XMSS: Extended Hash-Based Signatures
draft-irtf-cfrg-xmss-hash-based-signatures-01

Abstract

This note describes the eXtended Merkle Signature Scheme (XMSS), a hash-based digital signature system. It follows existing descriptions in scientific literature. The note specifies the WOTS+ one-time signature scheme, a single-tree (XMSS) and a multi-tree variant (XMSS^MT) of XMSS. Both variants use WOTS+ as a main building block. XMSS provides cryptographic digital signatures without relying on the conjectured hardness of mathematical problems.

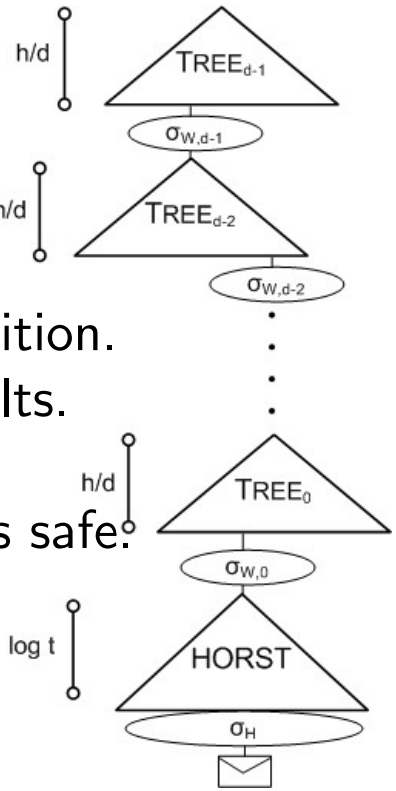
Cons:

- Biggish signature
- Stateful
  Adam Langley "for most environments it's a huge foot-cannon."


ELIMINATE
THE STATE

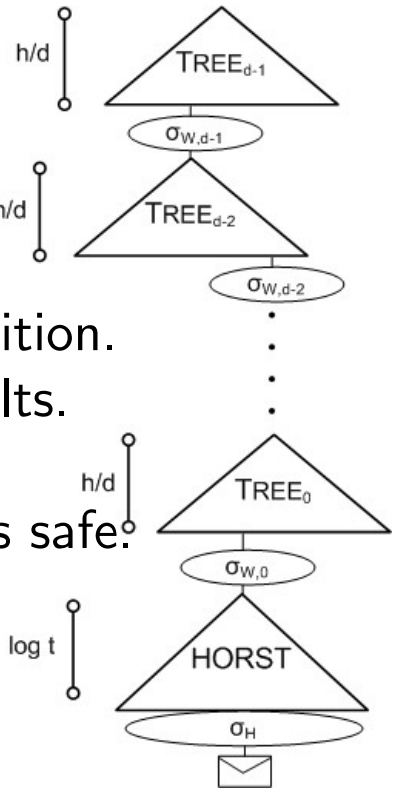
Stateless hash-based signatures

- Idea from 1987 Goldreich:
  - Signer builds huge tree of certificate authorities.
  - Signature includes certificate chain.
  - Each CA is a hash of master secret and tree position.
    This is deterministic, so don't need to store results.
  - **Random** bottom-level CA signs message.
    Many bottom-level CAs, so one-time signature is safe.

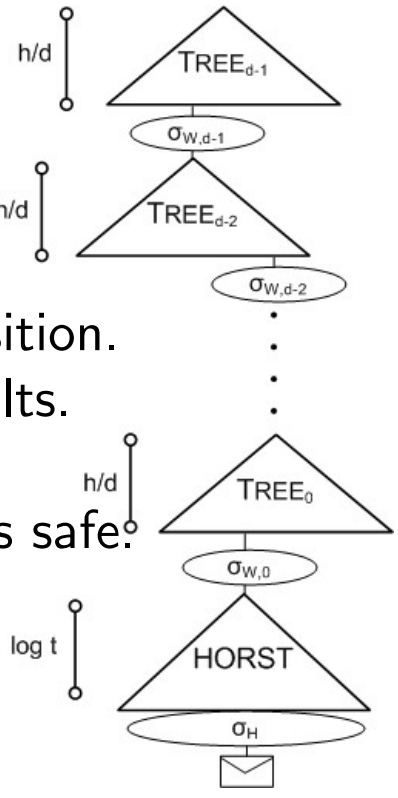


Stateless hash-based signatures

- Idea from 1987 Goldreich:
  - Signer builds huge tree of certificate authorities.
  - Signature includes certificate chain.
  - Each CA is a hash of master secret and tree position.
    This is deterministic, so don't need to store results.
  - **Random** bottom-level CA signs message.
    Many bottom-level CAs, so one-time signature is safe.
- 0.6 MB: Goldreich's signature with
  good 1-time signature scheme.
- 1.2 MB: average Debian package size.
- 1.8 MB: average web page in Alexa Top 1000000.



Stateless hash-based signatures



- Idea from 1987 Goldreich:
  - Signer builds huge tree of certificate authorities.
  - Signature includes certificate chain.
  - Each CA is a hash of master secret and tree position.
    This is deterministic, so don't need to store results.
  - **Random** bottom-level CA signs message.
    Many bottom-level CAs, so one-time signature is safe.
- 0.6 MB: Goldreich's signature with
  good 1-time signature scheme.
- 1.2 MB: average Debian package size.
- 1.8 MB: average web page in Alexa Top 1000000.

Our (lots of people) proposal: SPHINCS `sphincs.cr.yp.to`
0.041 MB signature; new optimization of Goldreich.
Modular, guaranteed as strong as its components (hash, PRNG).
Well-known components chosen for 2^{128} post-quantum security.

McBits (Bernstein, Chou, Schwabe, CHES 2013)

- Encryption is super fast anyways (just a vector-matrix multiplication).
- Main step in decryption is decoding of Goppa code. The McBits software achieves this in constant time.
- Decoding speed at 2^{128} pre-quantum security: $(n; t) = (4096; 41)$ uses 60493 Ivy Bridge cycles.
- Decoding speed at 2^{263} pre-quantum security: $(n; t) = (6960; 119)$ uses 306102 Ivy Bridge cycles.
- Grover speedup is less than halving the security level, so the latter parameters offer at least 2^{128} post-quantum security.
- More at https://binary.cr.yp.to/mcbits.html.

QcBits (Chou, CHES 2016)

- Focus on key size, use QC-MDPC (quasi-cyclic medium-density parity-check) codes; size 1KB.
- Main step in decryption is decoding of QC-MDPC code.
- Thomas talked about decoding algorithms on Monday.
- Tricky part for deployment: implement bit-flipping algorithm (statistical decoding) in constant time.
- Chou's QcBits software is first to achieve constant time for 32/64-bit platforms.
- Nice idea for efficiency and math tricks: work more with $\mathbf{F}_2[x]/(x^n+1)$. Study of dense and sparse multiplications.
- Must fix decoding rounds. Strategy for 2^{80}-security parameters gives no failures in 10^8 runs.

Platform	Key-pair	Encrypt	Decrypt	reference	scheme
Haswell	890 820	197 148	1 550 056	**Tung Chou**	KEM/DEM
	14 234 347	34 123	3 104 624	ACMTECS	McEliece
Cortex-M4	140 358 227	2 227 151	14 378 769	**Tung Chou**	KEM/DEM
	63 185 108	2 623 432	18 416 012	PQCrypto'16	KEM/DEM
	148 576 008	7 018 493	42 129 589	PQCrypto'14	McEliece

- More at https://www.win.tue.nl/~tchou/qcbits/.

Flush, Gauss, and Reload A Cache-Attack on the BLISS Lattice-Based Signature Scheme

Leon Groot Bruinderink, Andreas Hülsing,
Tanja Lange, and Yuval Yarom

How about other interesting candidates?

- Bimodal Lattice Signature Scheme (BLISS) (CRYPTO '13 by Léo Ducas and Alain Durmus and Tancrède Lepoint and Vadim Lyubashevsky)
- Pretty short and efficient; already included in strongSwan (library for IPsec-based VPN).
- Needs noise from discrete Gaussian distribution.
- Obvious issues about constant time – but hard to exploit cumulative measurement. What can we actually attack?
- Our paper (CHES 2016) is first side-channel attack on a lattice-based signature scheme.

Background

- Work in $R = \mathbf{Z}[x]/(x^n + 1)$, $n = 2^r$, and $R_q = (\mathbf{Z}/q)[x]/(x^n + 1)$ for q prime.
- Switch representation between polynomial and vector notation.

$$f(x) = \sum_{i=0}^{n-1} f_i x^i \Leftrightarrow f = (f_{n-1}, f_{n-2}, \ldots, f_1, f_0).$$

- Polynomial multiplication then corresponds to vector-matrix multiplication. Let $f, g, \in R_q$, then

$$f \cdot g = fG = gF,$$

where $F, G \in (\mathbf{Z}/q)^{n \times n}$ match vectors of $x^i f$ and $x^j g$.

$$\begin{pmatrix} f_0 & -f_{n-1} & -f_{n-2} & \cdots & -f_1 \\ f_1 & f_0 & -f_{n-1} & \cdots & -f_2 \\ \vdots & \vdots & \ddots & \vdots & \\ f_{n-1} & f_{n-2} & f_{n-3} & \cdots & f_0 \end{pmatrix}$$

Simplified BLISS

- Secret key $S = (s_1, s_2) = (f, 2g + 1) \in R_q^2$, f, g sparse in $\{0, \pm 1\}^n$.
- Public key $A = (a_1, a_2) \in R_{2q}^2$, with key equation $a_1 s_1 + a_2 s_2 \equiv q \bmod 2q$.
- Computed as $a_q = (2g + 1)/f \bmod q$ (restart if f is not invertible); then $A = (2a_q, q - 2) \bmod 2q$.

Simplified BLISS

- Secret key $S = (s_1, s_2) = (f, 2g + 1) \in R_q^2$, f, g sparse in $\{0, \pm 1\}^n$.
- Public key $A = (a_1, a_2) \in R_{2q}^2$, with key equation $a_1 s_1 + a_2 s_2 \equiv q \bmod 2q$.
- Computed as $a_q = (2g + 1)/f \bmod q$ (restart if f is not invertible); then $A = (2a_q, q - 2) \bmod 2q$.
- Can verify key guess for f with key equation; g computable.
- Focus on s_1; $-S$ just as good as S.

Simplified BLISS

- Secret key $S = (s_1, s_2) = (f, 2g+1) \in R_q^2$, f, g sparse in $\{0, \pm 1\}^n$.
- Public key $A = (a_1, a_2) \in R_{2q}^2$, with key equation $a_1 s_1 + a_2 s_2 \equiv q \bmod 2q$.
- Computed as $a_q = (2g+1)/f \bmod q$ (restart if f is not invertible); then $A = (2a_q, q-2) \bmod 2q$.
- Can verify key guess for f with key equation; g computable.
- Focus on s_1; $-S$ just as good as S.
- To sign, sample y from discrete n-dim Gaussian $D_{\mathbf{Z}^n,\sigma}$.
- $c = H(a_1, y, \text{public stuff})$ // H special hash function.
- choose a random bit b.
- Signature: (z, c) with $z = y + (-1)^b s_1 \cdot c \bmod 2q$.
- Can get $\pm s_1 = (z-y)/c \in R_q$ if we know y, the error vector/polynomial; (c needs to be invertible).

Use partial information on y to attack

- Rename s_1 to s. $z = y + (-1)^b s \cdot c \bmod 2q$.
- SCA might give us only one coefficient of y per signature. How to combine?

Use partial information on y to attack

- Rename s_1 to s. $z = y + (-1)^b s \cdot c \bmod 2q$.
- SCA might give us only one coefficient of y per signature. How to combine?
- Note that z_i corresponds to coefficient of x^i.
- Then $z_i = y_i + (-1)^b \left(\sum_{j=0}^{i} s_j c_{i-j} - \sum_{j=i+1}^{n-1} s_j c_{n+i-j} \right)$.
- Have z from (z, c). Write $sc = sC, C \in \{0, \pm 1\}^{n \times n}$. Then C has known columns $c_0 = c, c_1 = xc, \ldots, c_{n-1} = x^{n-1}c$ and $z_i - y_i = (-1)^b \langle s, c_i \rangle$.
- Collect many such relations. No need to go for unique i; because of rotation. Can get first entry 3 times, and second one never.
- Build system of equations.

System of equations

- Build system of equations (i is now just the i-th entry).
- c, y, and z vary; c_i are vectors/polynomials. s is fixed.
- $$\begin{pmatrix} (-1)^{b_0}(z_0 - y_0) \\ (-1)^{b_1}(z_1 - y_1) \\ \dots \\ (-1)^{b_{n-1}}(z_{n-1} - y_{n-1}) \end{pmatrix} = (s_0 s_1 \dots s_{n-1}) \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix}$$
- Looks powerful, but exhaustive search through b_i options kills this.

System of equations

- Build system of equations (i is now just the i-th entry).
- c, y, and z vary; c_i are vectors/polynomials. s is fixed.
- $$\begin{pmatrix} (-1)^{b_0}(z_0 - y_0) \\ (-1)^{b_1}(z_1 - y_1) \\ \cdots \\ (-1)^{b_{n-1}}(z_{n-1} - y_{n-1}) \end{pmatrix} = (s_0 s_1 \ldots s_{n-1}) \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix}$$
- Looks powerful, but exhaustive search through b_i options kills this.
- Splurge: use only entries where known z_i equals side channel info $z_i = y_i$ removes b's.
- Then solve for s.

Results

- Analysis of CDT sampling with guide table and of rejection (Bernoulli) sampling. We get useful side-channel information.
- Interesting and hard-to-avoid leakage.
- Different precision of information, but enough from cache-timing attacks.
- Above strategy with $z_i = y_i$ in a very small set (strong side channel, bur somewhat rare) takes on average 1671 signatures (for BLISS-I with $n = 512$)
- For CDT sampling always have some with uncertainty in y_i; but use much more common events.
- Use LLL to deal with uncertainty and just 441 signatures to attack CDT version,
- Run (well synchronized) spy process on same device; sample and break; actually get expected data.
- More uncertainty, more LLL (for both, CDT and Bernoulli) but works! Need about 100 extra equations for BLISS-I.

Further resources on post-quantum

- https://pqcrypto.org: Our (Dan and Tanja) survey site.
  - Many pointers: e.g., PQCrypto 2016.
  - Bibliography for 4 major PQC systems.
- https://pqcrypto.eu.org: PQCRYPTO EU project.
  Coming soon:
  - Expert recommendations.
  - Free software libraries.
  - More benchmarking to compare cryptosystems.
  - 2017: workshop and spring/summer school.
- https://twitter.com/pqc_eu: PQCRYPTO Twitter feed.
- https://sphincs.cr.yp.to
- https://binary.cr.yp.to/mcbits.html
- https://www.win.tue.nl/~tchou/qcbits/
- https://eprint.iacr.org/2016/300 (SCA on BLISS)