Classic McEliece: conservative code-based cryptography

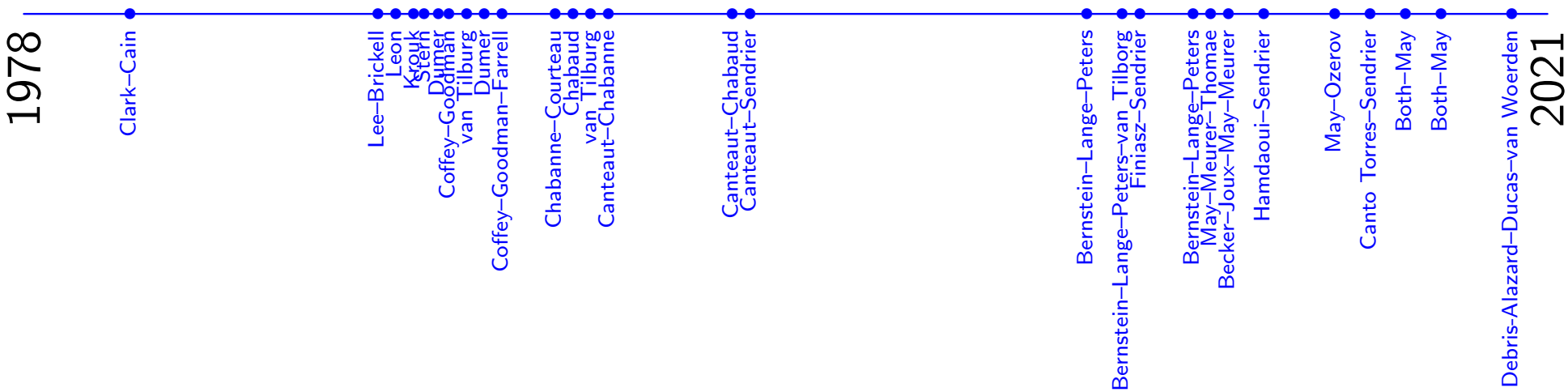
Martin R. Albrecht, Daniel J. Bernstein, Tung Chou,
Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram,
Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen,
Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters,
Peter Schwabe, Nicolas Sendrier, Jakub Szefer,
Cen Jung Tjhai, Martin Tomlinson, Wen Wang

https://classic.mceliece.org/

7 – 9 June 2021
Third NIST PQC workshop

McEliece security stability

$$\lim_{K \to \infty} \frac{\log_2 \mathrm{AttackCost}_{\mathrm{year}}(K)}{\log_2 \mathrm{AttackCost}_{2021}(K)}$$



McEliece security stability

Blue: McEliece.

Red: Lattices have lost much more security. Lattices had 42% higher security levels a decade ago than they have today.

Concrete analysis

Attacking and defending the McEliece cryptosystem (Bernstein, Lange, Peters; 2008)

- McEliece original parameters (from 1978) $n = 1024, t = 50$, designed for 2^{64} security.
- Security estimate $2^{60.4}$ using 2008 Markov-chain analysis.
- Broken in 8000 core-days on CPU mix $\approx 2^{60.0}$ cycles on Core 2.
- Using improved version of Stern's algorithm.

A Finite Regime Analysis of Information Set Decoding Algorithms (Baldi, Barenghi, Chiaraluce, Pelosi, Santini; 2019)

- Covers Prange to BJMM and quantum versions.
- Bit operations & memory for all Classic McEliece parameters.

Concrete analysis

Attacking and defending the McEliece cryptosystem (Bernstein, Lange, Peters; 2008)

- McEliece original parameters (from 1978) $n = 1024, t = 50$, designed for 2^{64} security.
- Security estimate $2^{60.4}$ using 2008 Markov-chain analysis.
- Broken in 8000 core-days on CPU mix $\approx 2^{60.0}$ cycles on Core 2.
- Using improved version of Stern's algorithm.

A Finite Regime Analysis of Information Set Decoding Algorithms (Baldi, Barenghi, Chiaraluce, Pelosi, Santini; 2019)

- Covers Prange to BJMM and quantum versions.
- Bit operations & memory for all Classic McEliece parameters.

Syndrome Decoding in the Goppa-McEliece Setting. Details on record 24 (Esser, May, Zweydinger; 2021)

- Challenge parameters $n = 1223, t = 23$.
- Security estimate $2^{58.9}$ using 2008 Markov-chain analysis.
- Broken in 3 days on 4 EPYC 7742 $\approx 2^{57.5}$ cycles on EPYC.
- Using BJMM/MMT variant.

Parameter sets – round 1

n	t	public key	secret key	ciphertext
8 192	128	1 357 824 bytes	14 080 bytes	240 bytes
Both n and t powers of 2.				
6 960	119	1 047 319 bytes	13 908 bytes	226 bytes
Max security with pkbytes $\leq 2^{20}$.				

Parameter sets – round 1, 2, and 3

n	t	public key	secret key	ciphertext
8 192	128	1 357 824 bytes	14 080 bytes	240 bytes
Both n and t powers of 2. Same as Round 1.				
6 960	119	1 047 319 bytes	13 908 bytes	226 bytes
Max security with pkbytes $\leq 2^{20}$. Same as Round 1.				
6 688	128	1 044 992 bytes	13 892 bytes	240 bytes
Max security with pkbytes $\leq 2^{20}$ if n and t are multiples of 32.				
4 608	96	524 160 bytes	13 568 bytes	188 bytes
Max security with pkbytes $\leq 2^{19}$ if n and t are multiples of 32.				
3 488	64	261 120 bytes	6 452 bytes	128 bytes
Max security with pkbytes $\leq 2^{18}$ if n and t are multiples of 32.				

Small ciphertext makes a large difference

PQ-WireGuard (Hülsing, Ning, Schwabe, Weber, Zimmermann; IEEE S&P 2021).

- Uses McEliece for long-term identity key in KEM-KEM construction.
- McEliece key exchanged out of band at registration.
- Strong benefit from short ciphertexts.
- Combined with lattice-based scheme for ephemeral keys.

McTiny (Bernstein, Lange; USENIX Security 2020)

- McEliece also used for ephemeral keys.
- Avoids DoS memory flooding attacks by using structure of code-based encryption. Server returns partial encryption and state in cookie encrypted to itself; cookie is smaller than network packet sent to server.
- Good speed and security with congestion control and surrounding protocol.

Timings on Haswell, in cycles

IND-CCA means we can generate key once and use it many times.

Timings for encapsulation and decapsulation matter.

system	keygen	keygen `f`	encap	decap
`mceliece348864`	46 526 112	36 627 388	43 832	134 184
`mceliece460896`	158 155 696	116 914 656	115 540	270 856
`mceliece6688128`	458 561 448	284 468 140	149 080	322 988
`mceliece6960119`	330 214 944	246 291 008	159 116	300 688
`mceliece8192128`	409 854 088	316 166 640	177 480	325 744

`f` version:

- Permit limited swaps of columns, $(\mu, \nu) = (32, 64)$.
- Increase chance of full-rank submatrix.
- Limit makes constant-time algorithm efficient.
- Introduced in round 2. Added to proposal in round 3.

Optimized implementations for Cortex-M4

pqm4, 2019: Classic McEliece public keys are "too large to fit into the memory of our platform"

Optimized implementations for Cortex-M4

pqm4, 2019: Classic McEliece public keys are "too large to fit into the memory of our platform"

Classic McEliece implementation with low memory footprint (Roth, Karatsiolis, Krämer; CARDIS 2020). "an implementation of Classic McEliece on an ARM Cortex-M4 processor, optimized to overcome memory constraints"; stream public key off device

Classic McEliece on the ARM Cortex-M4 (Chen, Chou; CHES 2021). `mceliece348864` fits on Cortex-M4, including public key!

- 2 146 932 033 keygen (only 1 430 811 294 for `f` version).
- 582 199 encap
- 2 706 681 decap

`mceliece8192128`: 7 481 747 for decap (private keys are tiny).

See earlier talk Classic McEliece on the ARM Cortex-M4.

High-assurance

- New control-bits algorithm fully verified.
  See separate talk Fast verified post-quantum software, part 1: RAM subroutines.
- Ongoing: Formal verification of Classic McEliece software (using Cryptol/SAW).

High-assurance

- New control-bits algorithm fully verified.
  See separate talk Fast verified post-quantum software, part 1: RAM subroutines.
- Ongoing: Formal verification of Classic McEliece software (using Cryptol/SAW).
- Classic McEliece PKE features
  - PKE is deterministic.
  - PKE has no decryption failures.

High-assurance

- New control-bits algorithm fully verified.
  See separate talk Fast verified post-quantum software, part 1: RAM subroutines.
- Ongoing: Formal verification of Classic McEliece software (using Cryptol/SAW).
- Classic McEliece PKE features
  - PKE is deterministic.
  - PKE has no decryption failures.

  This means the simplest and tightest proofs for FO apply.
  See Tighter proofs of CCA security in the quantum random oracle model (Bindel, Hamburg, Hövelmanns, Hülsing, Persichetti; TCC 2019).

Classic McEliece – ready for standardization

- In anticipation of standard, Round-3 expanded documentation:
  - Details secret key format, incl. options for compression.
  - Specifies full dataflow from RNG outputs.
  - Internal re-implementation from spec matches test vectors of optimized implementation.
- Open-source (public domain) implementations.
  - Constant-time software implementations.
    `ref`, `vec`, `sse`, `avx`, `m4`.
  - FPGA implementation.
- No patents.

See https://classic.mceliece.org for more details.