Communications protocol implications of using code-based cryptography

Tanja Lange

Eindhoven University of Technology

NCCoE workshop on migrating to post-quantum cryptography

History of code-based cryptography

1978 McEliece: Public-key encryption using error-correcting codes.

- Original parameters designed for 2^{64} security.
  Using 1962 Prange information-set decoding for parameter choice.
- 2008 Bernstein–Lange–Peters: broken original parameters in $\approx 2^{60}$ cycles.
- Easily scale up for higher security.
- The McEliece system (with later key-size optimizations)
  achieves 2^λ security against Prange's attack
  using $(0.741186\ldots + o(1))\lambda^2(\log_2 \lambda)^2$-bit keys as $\lambda \to \infty$.

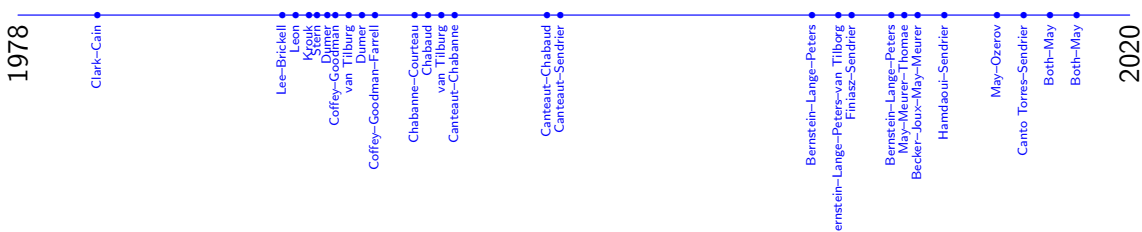
Security analysis of McEliece encryption

Some papers studying algorithms for attackers:
1962 Prange; 1981 Clark–Cain, crediting Omura; 1988 Lee–Brickell; 1988 Leon;
1989 Krouk; 1989 Stern; 1989 Dumer; 1990 Coffey–Goodman; 1990 van Tilburg;
1991 Dumer; 1991 Coffey–Goodman–Farrell; 1993 Chabanne–Courteau; 1993 Chabaud;
1994 van Tilburg; 1994 Canteaut–Chabanne; 1998 Canteaut–Chabaud;
1998 Canteaut–Sendrier; 2008 Bernstein–Lange–Peters; 2009 Bernstein–Lange–Peters–van Tilborg; 2009 Bernstein (**post-quantum**); 2009 Finiasz–Sendrier;
2010 Bernstein–Lange–Peters; 2011 May–Meurer–Thomae; 2012 Becker–Joux–May–Meurer;
2013 Hamdaoui–Sendrier; 2015 May–Ozerov; 2016 Canto Torres–Sendrier;
2017 Kachigar–Tillich (**post-quantum**); 2017 Both–May; 2018 Both–May; 2018 Kirshanova (**post-quantum**).

All of these attacks involve huge searches, like attacking AES.
The quantum attacks (Grover etc.) leave at least half of the bits of security.

Attack progress over time

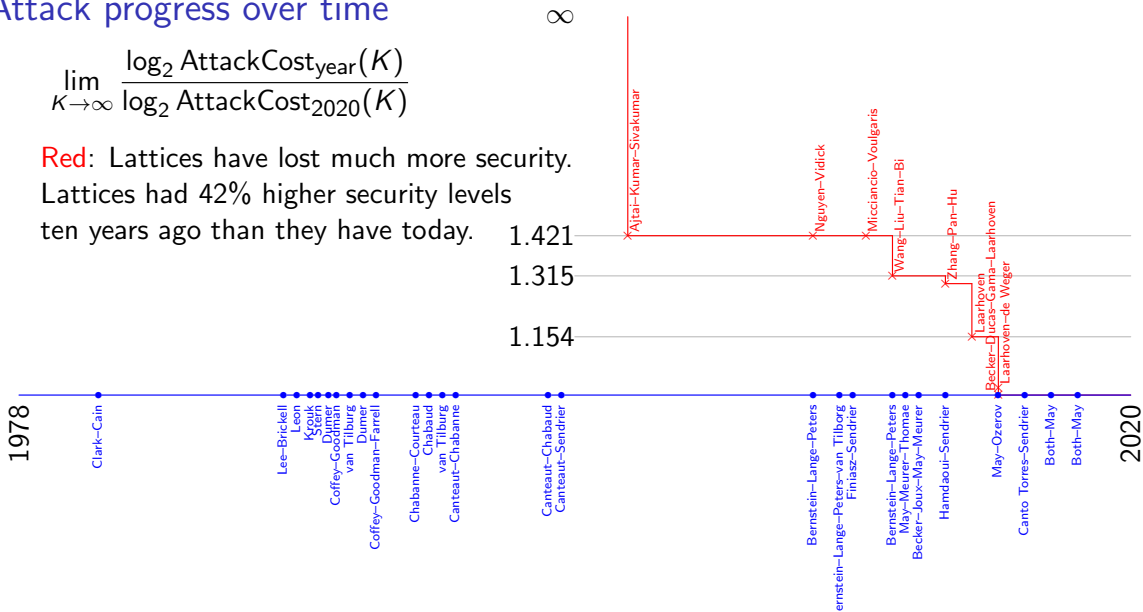
$$\lim_{K\to\infty} \frac{\log_2 \mathrm{AttackCost}_{\mathrm{year}}(K)}{\log_2 \mathrm{AttackCost}_{2020}(K)}$$



Attack progress over time

$$\lim_{K\to\infty} \frac{\log_2 \mathrm{AttackCost}_{\mathrm{year}}(K)}{\log_2 \mathrm{AttackCost}_{2020}(K)}$$

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



NIST PQC submission Classic McEliece

No patents. ✓
Shortest ciphertexts of all Round-2 candidates. ✓
Fast open-source constant-time software implementations. ✓
Very conservative system, expected to last; has strongest security track record. ✓

Sizes with similar post-quantum security to AES-128, AES-192, AES-256:

Metric	**mceliece348864**	**mceliece460896**	**mceliece6960119**
Public-key size	261120 bytes	524160 bytes	1047319 bytes
Secret-key size	6452 bytes	13568 bytes	13908 bytes
Ciphertext size	128 bytes	188 bytes	226 bytes
Key-generation time	52415436 cycles	181063400 cycles	417271280 cycles
Encapsulation time	43648 cycles	77380 cycles	143908 cycles
Decapsulation time	130944 cycles	267828 cycles	295628 cycles

See https://classic.mceliece.org for authors, details & parameters.

Key issues for McEliece

Key issues for McEliece

BIG PUBLIC KEYS.

Key issues for McEliece

- Sending 1MB takes time and bandwidth.
- Google–Cloudflare experiment:

  > *in some cases the public-key + ciphertext size was too large to be viable in the context of TLS*

  and even 10KB messages dropped.

But users send big data anyway. We have lots of bandwidth.
A key takes less space than a kitten picture.

Each client spends a small fraction of a second generating new ephemeral 1MB key.

Key issues for McEliece

- Sending 1MB takes time and bandwidth.
- Google–Cloudflare experiment:

  > *in some cases the public-key + ciphertext size was too large to be viable in the context of TLS*

  and even 10KB messages dropped.

But users send big data anyway. We have lots of bandwidth.
A key takes less space than a kitten picture.

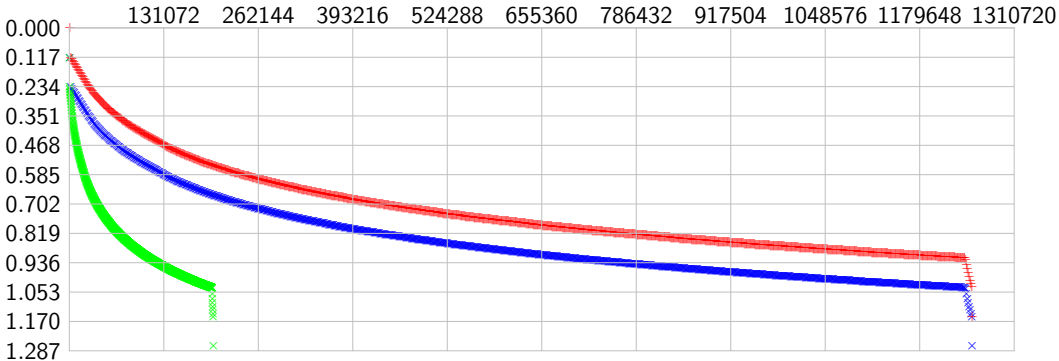
Each client spends a small fraction of a second generating new ephemeral 1MB key.

- But: If any client is allowed to send a new ephemeral 1MB McEliece key to server, an attacker can easily flood server's memory. **This invites DoS attacks.**

Use cases for Classic McEliece

- Standard public-key encryption, e.g. GnuPG/PGP with long-term keys.
- 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 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.

Measurements of our software (https://mctiny.org)



Client time vs. bytes sent, bytes acknowledged, bytes in acknowledgments.
Curve shows packet pacing from our new user-level congestion-control library.