Classic McEliece: conservative code-based cryptography Round 2

https://classic.mceliece.org/

Daniel J. Bernstein[1], Tung Chou[2], Tanja Lange[3],
Ingo von Maurich, Rafael Misoczki[4], Ruben Niederhagen[5],
Edoardo Persichetti[6], Christiane Peters, Peter Schwabe[7],
Nicolas Sendrier[8], Jakub Szefer[9], Wen Wang[9]

[1]University of Illinois at Chicago, [2]Osaka University,
[3]Technische Universiteit Eindhoven, [4]Intel Corporation, [5]Fraunhofer SIT,
[6]Florida Atlantic University, [7]Radboud University, [8]Inria, [9]Yale University

24 August 2019
Second NIST PQC workshop

Conservative code-based encryption

“This is going to be the most boring submission of them all”.
(T. Lange, April 2018)

Conservative code-based encryption

“This is going to be the most boring submission of them all”.
(T. Lange, April 2018)

This is still the case.

Conservative code-based encryption

“This is going to be the most boring submission of them all”.
(T. Lange, April 2018)

This is still the case.

Nothing has changed in more than 40 years in the asymptotics of OW-Passive security for McEliece.

Conservative code-based encryption

"This is going to be the most boring submission of them all".
(T. Lange, April 2018)

This is still the case.

Nothing has changed in more than 40 years in the asymptotics of OW-Passive security for McEliece.

We follow best practices to obtain an IND-CCA KEM.

Conservative code-based encryption

“This is going to be the most boring submission of them all”.
(T. Lange, April 2018)

This is still the case.

Nothing has changed in more than 40 years in the asymptotics of OW-Passive security for McEliece.

We follow best practices to obtain an IND-CCA KEM.

For Round 2, we added more parameter sets, as requested.

One-wayness (OW-Passive)

Fundamental security question (SDP):
Given random parity-check matrix H and syndrome s,
can attacker efficiently find e with $s = He$?

One-wayness (OW-Passive)

Fundamental security question (SDP):
Given random parity-check matrix H and syndrome s,
can attacker efficiently find e with $s = He$?

- Write $H = (I_{n-k}|T)$, public key is $(n-k) \times k$ matrix T, $n - k = t \log_2 q$. H constructed from binary Goppa code.
- Encapsulate using e of weight t.
- Decapsulate using Goppa decoding algorithm.

One-wayness (OW-Passive)

Fundamental security question (SDP):
Given random parity-check matrix H and syndrome s,
can attacker efficiently find e with $s = He$?

- Write $H = (I_{n-k}|T)$, public key is $(n-k) \times k$ matrix T, $n - k = t \log_2 q$. H constructed from binary Goppa code.
- Encapsulate using e of weight t.
- Decapsulate using Goppa decoding algorithm.

Classic McEliece only uses Niederreiter's "dual" framework, and some decoding speedups. This improves efficiency while clearly preserving security.

Parameter sets

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.				

Parameter sets

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.				

Ciphertext size

Classic McEliece has very short ciphertexts.

Ciphertext size

Classic McEliece has very short ciphertexts.

We could save another 32 bytes of ciphertext by removing plaintext confirmation in the IND-CCA transform.
However, plaintext confirmation has security advantages.

Ciphertext size

Classic McEliece has very short ciphertexts.

We could save another 32 bytes of ciphertext by removing
plaintext confirmation in the IND-CCA transform.
However, plaintext confirmation has security advantages.

Even including these 32 bytes,
Classic McEliece has the smallest ciphertexts in the competition.

Ciphertext size

Classic McEliece has very short ciphertexts.

We could save another 32 bytes of ciphertext by removing
plaintext confirmation in the IND-CCA transform.
However, plaintext confirmation has security advantages.

Even including these 32 bytes,
Classic McEliece has the smallest ciphertexts in the competition.

High degree of flexibility in choice of parameters.
Could increase key size to obtain even smaller ciphertexts.

Optimized implementations

We provided four implementations for each parameter set, all constant-time: `ref`, `vec`, `sse`, `avx`.

Optimized implementations

We provided four implementations for each parameter set, all constant-time: `ref`, `vec`, `sse`, `avx`.

Times improved: e.g. for `mceliece8192128` (Haswell cycles)

- 4,000,000,000 → 811,681,256 for keygen
- 300,000 → 194,500 for encaps
- 450,000 → 322,236 for decaps

Optimized implementations

We provided four implementations for each parameter set, all constant-time: `ref`, `vec`, `sse`, `avx`.

Times improved: e.g. for `mceliece8192128` (Haswell cycles)

- 4,000,000,000 → 811,681,256 for keygen
- 300,000 → 194,500 for encaps
- 450,000 → 322,236 for decaps

Very fast in hardware (Artix-7/Virtex-7).

Optimized implementations

We provided four implementations for each parameter set, all constant-time: `ref`, `vec`, `sse`, `avx`.

Times improved: e.g. for `mceliece8192128` (Haswell cycles)

- 4,000,000,000 → 811,681,256 for keygen
- 300,000 → 194,500 for encaps
- 450,000 → 322,236 for decaps

Very fast in hardware (Artix-7/Virtex-7).

For `mceliece8192128` (time-optimized)

- 1,286,179 for keygen
- 6,528 for encaps
- 26,237 for decaps

(cycles at 28.4MHz on Virtex-7 XC7V2000T FPGA).

Key-generation speed

Classic McEliece uses keys in systematic form.
We choose to abort if left $r \times r$ submatrix has not full rank.
This works about 29% of the time.

Key-generation speed

Classic McEliece uses keys in systematic form.
We choose to abort if left $r \times r$ submatrix has not full rank.
This works about 29% of the time.

NTS-KEM uses permuted systematic form.
This works about 100% of the time, but pivoting
makes constant-time Gaussian elimination much slower.

Key-generation speed

Classic McEliece uses keys in systematic form.
We choose to abort if left $r \times r$ submatrix has not full rank.
This works about 29% of the time.

NTS-KEM uses permuted systematic form.
This works about 100% of the time, but pivoting makes constant-time Gaussian elimination much slower.

We introduced and analyzed (μ, ν)-semi-systematic form to

- achieve KeyGen success probability about $1 - 2^{\mu-\nu}$,
- obtain a fast constant-time implementation of Gaussian elimination with pivoting limited by (μ, ν).

We have implemented 5 additional parameter sets with $(\mu, \nu) = (32, 64)$ as possible future proposals.

Large keys in practice

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

Large keys in practice

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

Key generation is well under a second even with largest parameters.

Large keys in practice

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

Key generation is well under a second even with largest parameters.

Even more efficient in hardware.

Large keys in practice

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

Key generation is well under a second even with largest parameters.

Even more efficient in hardware.

Public keys can use efficient broadcast networks
and do not add much to modern Internet traffic.

Large keys in practice

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

Key generation is well under a second even with largest parameters.

Even more efficient in hardware.

Public keys can use efficient broadcast networks
and do not add much to modern Internet traffic.

Bernstein-Lange “McTiny” fits McEliece into tiny network servers,
even with forward secrecy.

NIST submission Classic McEliece

- Security asymptotics unchanged by 40 years of cryptanalysis.
- Short ciphertexts.
- Efficient and straightforward conversion
  OW-CPA PKE → IND-CCA KEM.
- Open-source (public domain) implementations.
  - Constant-time software implementations.
  - FPGA implementation of full cryptosystem.
- No patents.

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