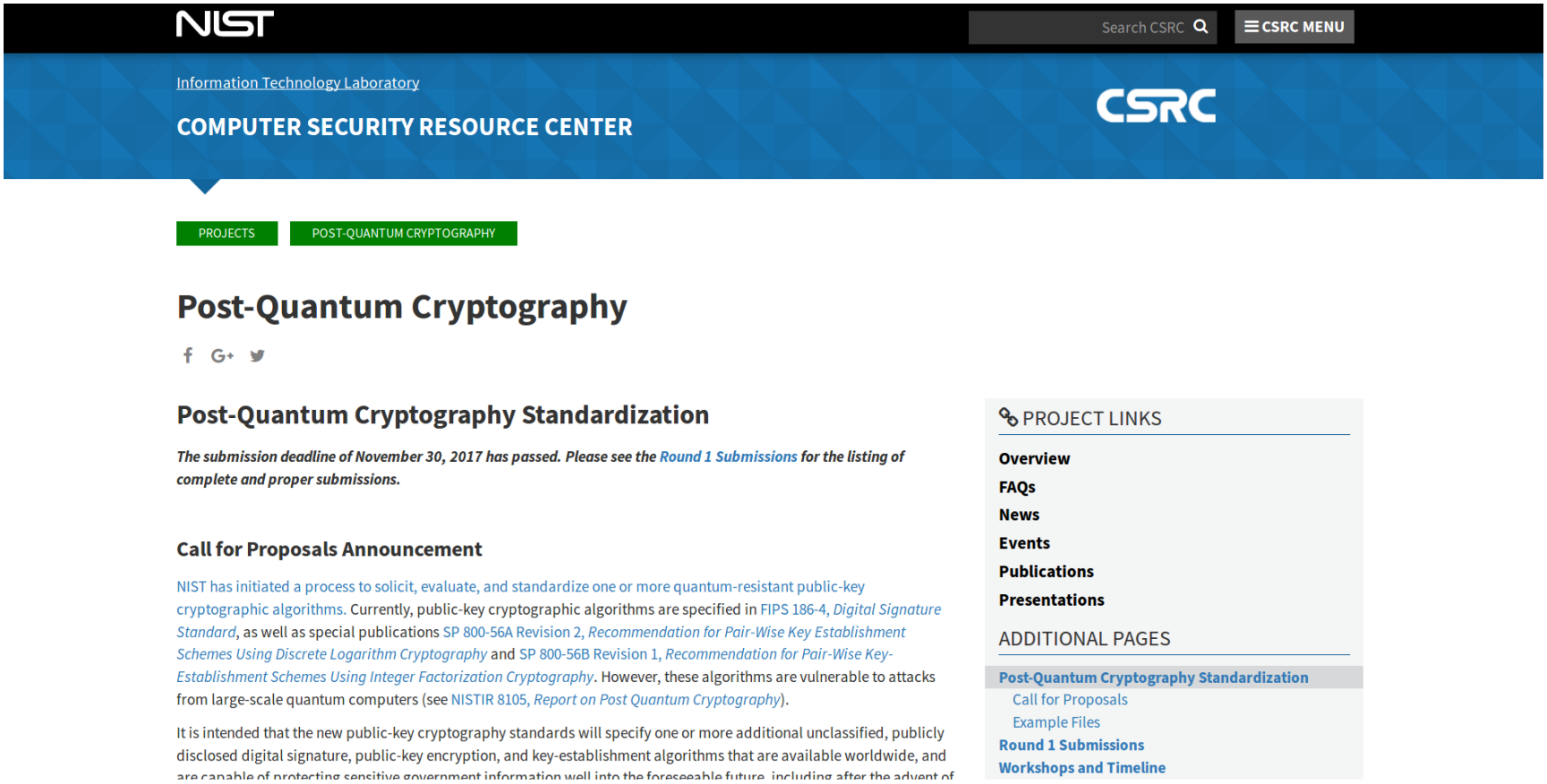
Classic McEliece: conservative code-based cryptography

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

29 June 2018
PQCRYPTO Mini-School and Workshop

NIST's Call



NIST

Search CSRC

CSRC MENU

Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER

CSRC

PROJECTS | POST-QUANTUM CRYPTOGRAPHY

Post-Quantum Cryptography

Post-Quantum Cryptography Standardization

The submission deadline of November 30, 2017 has passed. Please see the Round 1 Submissions for the listing of complete and proper submissions.

Call for Proposals Announcement

NIST has initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms. Currently, public-key cryptographic algorithms are specified in FIPS 186-4, *Digital Signature Standard*, as well as special publications SP 800-56A Revision 2, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography* and SP 800-56B Revision 1, *Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography*. However, these algorithms are vulnerable to attacks from large-scale quantum computers (see NISTIR 8105, *Report on Post Quantum Cryptography*).

It is intended that the new public-key cryptography standards will specify one or more additional unclassified, publicly disclosed digital signature, public-key encryption, and key-establishment algorithms that are available worldwide, and are capable of protecting sensitive government information well into the foreseeable future, including after the advent of

PROJECT LINKS

Overview
FAQs
News
Events
Publications
Presentations

ADDITIONAL PAGES

Post-Quantum Cryptography Standardization
Call for Proposals
Example Files
Round 1 Submissions
Workshops and Timeline

Classic McEliece

Classic McEliece

The first code-based public-key cryptosystem was introduced in 1978 by McEliece. The public key specifies a random binary Goppa code. A ciphertext is a codeword plus random errors. The private key allows efficient decoding: extracting the codeword from the ciphertext, identifying and removing the errors.

The McEliece system was designed to be one-way (OW-CPA), meaning that an attacker cannot efficiently find the codeword from a ciphertext and public key, when the codeword is chosen randomly. The security level of the McEliece system has remained remarkably stable, despite dozens of attack papers over 40 years. The original McEliece parameters were designed for only 2^{64} security, but the system easily scales up to "overkill" parameters that provide ample security margin against advances in computer technology, including quantum computers.

The McEliece system has prompted a tremendous amount of followup work. Some of this work improves efficiency while clearly preserving security: this includes a "dual" PKE proposed by Niederreiter, software speedups, and hardware speedups.

Furthermore, it is now well known how to efficiently convert an OW-CPA PKE into a KEM that is IND-CCA2 secure against all ROM attacks. This conversion is tight, preserving the security level, under two assumptions that are satisfied by the McEliece PKE: first, the PKE is deterministic (i.e., decryption recovers all randomness that was used); second, the PKE has no decryption failures for valid ciphertexts. Even better, very recent work suggests the possibility of achieving similar tightness for the broader class of QROM attacks. The risk that a hash-function-specific attack could be faster than a ROM or QROM attack is addressed by the standard practice of selecting a well-studied, high-security, "unstructured" hash function.

Classic McEliece brings all of this together. It is a KEM designed for IND-CCA2 security at a very high security level, even against quantum computers. The KEM is built conservatively from a PKE designed for OW-CPA security, namely Niederreiter's dual version of McEliece's PKE using binary Goppa codes. Every level of the construction is designed so that future cryptographic auditors can be confident in the long-term security of

Classic McEliece: a quick look

Cons

- Large public key size ($1 \sim 1.3$ MB)

Pros

- Based on a **40-year-old** code-based cryptosystem
- Small ciphertext size ($226 \sim 240$ bytes)
- Fast, constant-time en/decapsulation ($\leq 500\,000$ cycles)

40 years and more than 30 analysis papers later

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**).

40 years and more than 30 analysis papers later

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**).

The McEliece system uses $(c_0 + o(1))\lambda^2(\lg \lambda)^2$-bit keys as $\lambda \to \infty$ to achieve 2^λ security against all attacks known today.
Same $c_0 \approx 0.7418860694$.

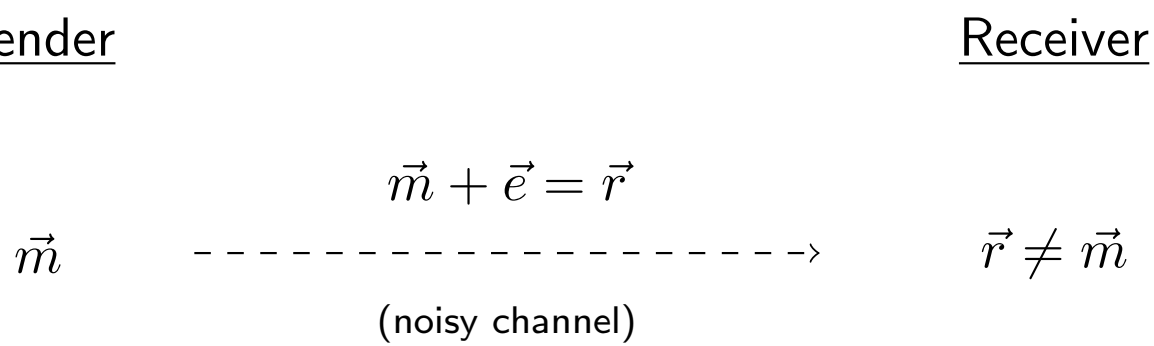
40 years and more than 30 analysis papers later

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**).

The McEliece system uses $(c_0 + o(1))\lambda^2(\lg \lambda)^2$-bit keys as $\lambda \to \infty$ to achieve 2^λ security against all attacks known today.
Same $c_0 \approx 0.7418860694$.

Replacing λ with 2λ stops all known *quantum* attacks.

McEliece/Niederreiter cryptosystem



McEliece/Niederreiter cryptosystem

Sender		Receiver
	$\vec{c} + \vec{e} = \vec{r}$	
$\vec{c} = \vec{m}G$	- - - - - - - - - - - - - - - - - - - -›	$\vec{c},\ \vec{e} = \text{Decode}(\vec{r})$
	(noisy channel)	

McEliece/Niederreiter cryptosystem

Sender		Receiver
$\vec{r} = \vec{m}G + \vec{e}$	$\xrightarrow{\vec{r}}$	$\vec{c},\ \vec{e} = \mathsf{Decode}(\vec{r})$

(McEliece, 1978)

McEliece/Niederreiter cryptosystem

Sender		Receiver
$\vec{r} = \vec{m}G + \vec{e}$	$\xrightarrow{\vec{r}}$	$\vec{c},\ \vec{e} = \mathsf{Decode}(\vec{r})$

(McEliece, 1978)

$\vec{r} = H\vec{e}$	$\xrightarrow{\vec{r}}$	$\vec{e} = \mathsf{Decode}(\vec{r})$

(Niederreiter, 1986)

McEliece/Niederreiter using binary Goppa code

- Definition of the code $C \subset \mathbb{F}_2^n$:

$$\frac{c_1}{x-\alpha_1} + \frac{c_2}{x-\alpha_2} + \cdots \frac{c_n}{x-\alpha_n} \equiv 0 \mod g(x)$$

- **Support** $(\alpha_1, \ldots, \alpha_n)$: n distinct elements in $\mathbb{F}_{2^m}$
- **Goppa polynomial**: random irreducible degree-t $g(x)$

McEliece/Niederreiter using binary Goppa code

- Definition of the code $C \subset \mathbb{F}_2^n$:

$$\frac{c_1}{x-\alpha_1} + \frac{c_2}{x-\alpha_2} + \cdots \frac{c_n}{x-\alpha_n} \equiv 0 \mod g(x)$$

- **Support** $(\alpha_1, \ldots, \alpha_n)$: n distinct elements in $\mathbb{F}_{2^m}$
- **Goppa polynomial**: random irreducible degree-t $g(x)$

- Secret key: $(\alpha_1, \ldots, \alpha_n)$, $g(x)$
- Public key: generating/parity-check matrix of C

- Classic McEliece: Niederreiter + binary Goppa code

Classic McEliece: parameter sets

`mceliece8192128`

- (m, n, t) = (13, 8192, 128)
- 1357824 bytes for public key.
- 14080 bytes for secret key.
- 240 bytes for ciphertext.
- More natural for software implementation

`mceliece6960119`

- (m, n, t) = (13, 6960, 119)
- 1047319 bytes for public key.
- 13908 bytes for secret key.
- 226 bytes for ciphertext.
- Fits into 1 megabyte

Classic McEliece: OW-CPA to ROM IND-CCA2

Secret key:

- $g, (\alpha_1, \ldots, \alpha_n)$, and an n-bit string s

Encapsulation:

- ciphertext $C = (C_0, C_1) = (He, \mathcal{H}_2(e))$
- session key $K = \mathcal{H}_1(e, C)$

Decapsulation:

- **decode** C_0 to get e^*
- compare C_1 with $\mathcal{H}_2(e^*)$
- $K^* = \mathcal{H}_0(s, C)$, if decoding or comparison failed
- $K^* = \mathcal{H}_1(e^*, C)$, if decoding and comparison both succeeded

Comparison with NTS-KEM

https://classic.mceliece.org/nist/vsntskem-20180629.pdf

Comparison with NTS-KEM: advertisement

"The NTS-KEM submission delares a US patent application and a granted UK patent describing a method by which a McEliece ciphertext may be shortened and have the same security as the full length McEliece ciphertext. The same method is used in NTS-KEM but in no other PQC submission as far as we can tell."

– Martin Tomlinson, 3 Jan., 2018

"We have decided to eliminate any uncertainty by abandoning the patent with immediate effect. Our submission will no longer be subject to any patents and is free for anyone to experiment with."

– Martin Tomlinson, 27 Apr., 2018

Comparison with NTS-KEM: implementations

	sec	key-gen	encapsulation	decapsulation	platform
CM-13-128	5	~~4010278828~~	295932	458476	Haswell
NTSKEM-13-80	3	123761512	368946	604459	Broadwell
NTSKEM-13-136	5	221106162	478323	1123879	Broadwell
NTSKEM-13-80	3	51275xxx	178xxx	332xxx	Skylake
NTSKEM-13-136	5	108501xxx	265xxx	644xxx	Skylake

Comparison with NTS-KEM: implementations

	sec	key-gen	encapsulation	decapsulation	platform
CM-13-128	5	~~4010278828~~	295932	458476	Haswell
NTSKEM-13-80	3	123761512	368946	604459	Broadwell
NTSKEM-13-136	5	221106162	478323	1123879	Broadwell
NTSKEM-13-80	3	51275xxx	178xxx	332xxx	Skylake
NTSKEM-13-136	5	108501xxx	265xxx	644xxx	Skylake

Some issues:

- problem in NTS-KEM's Skylake cycles: Turbo-boosted?
- constant-time vs non-constant-time key generation
- distributions of keys are different: are the support and $g(x)$ independent

Comparison with NTS-KEM: security

Decapsulation:

- decode C_0 to get e^*
- compare C_1 with $\mathcal{H}_2(e^*)$ — (1) **plaintext confirmation**
- $K^* = \mathcal{H}_0(s, C)$, if decoding or comparison failed — (2) **implicit rejection**
- $K^* = \mathcal{H}_1(e^*, C)$, if decoding and comparison both succeeded

Security

- Both schemes achieves ROM IND-CCA2
- Classic McEliece is more conservative: NTS-KEM only has (1)
- Simpler proof for Classic McEliece
- Classic McEliece has more chance of proving QROM IND-CCA2

Comparison with NTS-KEM: Goppa polynomial

Classic McEliece

- Irreducible g

NTS-KEM

- Valid square-free g (without linear factors)

Comparison with NTS-KEM: Goppa polynomial

Classic McEliece

- Irreducible g

NTS-KEM

- Valid square-free g (without linear factors)

Roughly $\delta = \exp(1)/t$ of valid square-free are irreducible

- which means that the potential gain in security level is bounded by $\log_2(1/\delta) = \log_2(t) - 1.44$.

Comparison with other code-based schemes

scheme	code
BigQuake	QC-Goppa
BIKE	QC-MDPC
Classic McEliece	Goppa
DAGS	dyadic GS
HQC	QC-MDPC
LAKE	rank
LEDAkem	QC-MDPC
LEDApkc	QC-MDPC
Lepton	(LPN)
LOCKER	rank
McNie	rank
NTS-KEM	Goppa
Ouroboros-R	rank
QC-MDPC KEM	QC-MDPC
RLCE-KEM	Goppa
RQC	rank

(schemes collected by Ryo Fujita)

Comparison with other code-based schemes

code	Goppa	QC-MDPC	rank-metric
submissions	Classic McEliece	BIKE	McNie
since	1978	2013	?
key size	1 MB	8 KB	630 B
ciphertext size	240 B	8 KB	761 B
decoding failure	no	yes	yes

Comparison with FrodoKEM

submissions	Classic McEliece	FrodoKEM
key size	1 MB	16 KB
ciphertext size	240 B	16 KB
enc./dec. cycles	$< 5 \cdot 10^5$	$> 10^7$
hard problem	well-studied	?