From: Tanja Lange <tanja@hyperelliptic.org>
Sent: Friday, September 30, 2022 4:55 PM
To: pqc-comments
Cc: pqc-forum; authorcontact-mceliece-merged@box.cr.yp.to
Subject: ROUND 4 OFFICIAL COMMENT: Classic McEliece
Attachments: mods3.pdf

The attached document "modifications for round 4" specifies the Classic McEliece tweaks for round 4. We will provide updated documentation and software matching this.

Tanja, on behalf of the Classic McEliece team

From: Michael Lyons <mlyons3@gmu.edu>
Sent: Friday, September 30, 2022 5:46 PM
To: pqc-forum
Cc: Tanja Lange; pqc-forum; authorcontact-...@box.cr.yp.to; pqc-comments
Subject: Re: ROUND 4 OFFICIAL COMMENT: Classic McEliece

On page 2 in the sentence [SECDED means “single error correction, double error correction”.]
I believe the last word should be "detection".

Regards,
Mike Lyons
Cryptographic Engineering Research Group
George Mason University

From: pqc-forum@list.nist.gov on behalf of Tung Chou <blueprint@crypto.tw>
Sent: Friday, September 30, 2022 9:07 PM
To: Michael Lyons
Cc: pqc-forum; pqc-comments
Subject: Re: [pqc-forum] Re: ROUND 4 OFFICIAL COMMENT: Classic McEliece

Hi Mike,

You are right. Thank you for pointing this out.

Tung Chou

On Sat, 1 Oct 2022 at 05:45, Michael Lyons <mlyons3@gmu.edu> wrote:

> On page 2 in the sentence [SECDED means “single error correction, double error correction”.]
> I believe the last word should be "detection".
>
> Regards,
> Mike Lyons
> Cryptographic Engineering Research Group
> George Mason University

From: 'Moody, Dustin (Fed)' via pqc-forum <pqc-forum@list.nist.gov>
Sent: Monday, October 3, 2022 9:29 AM
To: Tanja Lange
Cc: pqc-forum; authorcontact-mceliece-merged@box.cr.yp.to
Subject: [pqc-forum] Re: ROUND 4 OFFICIAL COMMENT: Classic McEliece

Thanks Tanja (and team),

When do you think you can send us the updated specs and software?

Dustin

From: Tanja Lange <tanja@hyperelliptic.org>
Sent: Friday, September 30, 2022 4:54 PM
To: pqc-comments <pqc-comments@nist.gov>
Cc: pqc-forum <pqc-forum@list.nist.gov>; authorcontact-mceliece-merged@box.cr.yp.to <authorcontact-mceliece-merged@box.cr.yp.to>
Subject: ROUND 4 OFFICIAL COMMENT: Classic McEliece

The attached document "modifications for round 4" specifies the
Classic McEliece tweaks for round 4. We will provide updated
documentation and software matching this.

Tanja, on behalf of the Classic McEliece team

--

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Tuesday, October 25, 2022 8:01 AM
To: pqc-comments
Cc: pqc-forum; authorcontact-mceliece-merged@box.cr.yp.to
Subject: ROUND 4 OFFICIAL COMMENT: Classic McEliece
Attachments: signature.asc

The round-4 Classic McEliece submission is available here:

https://classic.mceliece.org/nist/mceliece-20221023.tar.gz

As before, KATs have been split into a separate file:

https://classic.mceliece.org/nist/mceliece-kat-20221023.tar.gz

---D. J. Bernstein, on behalf of the Classic McEliece team

From: Wrenna Robson <wren.robson@gmail.com>
Sent: Tuesday, October 25, 2022 8:17 AM
To: pqc-forum; authorcontact-mceliece-merged@box.cr.yp.to; pqc-comments
Subject: Re: [pqc-forum] ROUND 4 OFFICIAL COMMENT: Classic McEliece

Thanks for this, Dan.

Obviously I've just had a quick glance, and will read in detail in the fullness of time, but I just want to say that I love the restructuring of the supporting documentation and the separation of content into the different documents for different purposes, and the rewriting and clarification of the content that I've seen already. It looks really great.

Best,

Wrenna

On Tue, 25 Oct 2022 at 13:01, D. J. Bernstein <djb@cr.yp.to> wrote:
>

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Wednesday, July 24, 2024 2:24 PM
To: pqc-comments
Cc: pqc-forum
Subject: ROUND 4 OFFICIAL COMMENT: Classic McEliece
Attachments: signature.asc

Summary: This comment points out a mathematical explanation for _why_ Classic McEliece has much smaller ciphertexts than, e.g., Kyber.

Context: The primary motivation for this KEM is security, but a frequent observation is that this KEM has much smaller ciphertexts than any of the alternatives. An interesting consequence is that the minimum network traffic for frequently used long-term post-quantum public keys (e.g., server identity keys) is achieved by this KEM. Various post-quantum deployments, such as Rosenpass, would lose efficiency if they switched the long-term keys from this KEM to alternatives.

The underlying cryptanalytic facts are that, for any particular ciphertext size, the ranking of costs of known attacks is as follows:

* Fastest: attacking ciphertexts or keys for Kyber, NTRU, etc.
* Much slower: attacking McEliece ciphertexts.
* Slowest: attacking McEliece keys.

People often wonder whether this means that the McEliece system hasn't been studied: i.e., whether further study would eliminate the McEliece ciphertext-size advantage. Normally this question is answered with a pointer to the long history of McEliece attack papers. I'm filing this comment to highlight a different answer.

I've recently given a talk with a unified description of code/lattice cryptosystems, pinpointing what's different about the McEliece system:

https://cr.yp.to/talks/2024.07.17/slides-djb-20240717-mceliece-4x3.pdf

Attacks against all of these cryptosystems can be viewed as finding a lattice vector close to a target point. Barak similarly commented in

https://eprint.iacr.org/2017/365

on the "family" of "coding/lattice" systems, and claimed that "all the known lattice-based public-key encryption schemes can be broken using oracle access to an O(√n) approximation algorithm for the lattice closest vector problem". A closer look shows, however, that it's easy to write down systems where that level of approximation doesn't break the system. For McEliece, the gap is only polylogarithmic: the distance d from the target to the lattice is so high that exponentially many lattice points are within distance d*polylog of the target.

The fundamental reason for this difference is that, for any particular ciphertext size, keygen and dec use a quantitatively much more powerful decoder for McEliece than for the alternatives. Simple attacks that exploit the low distances allowed by the alternative decoders are faster than state-of-the-art attacks against the much larger distances allowed by the Goppa decoder used by McEliece.

It's also worth noting that the structure of the alternative decoders is the basic reason that the weakness of the alternatives is shared between ciphertexts and keys. In principle the alternatives can strengthen ther keys (see, e.g., https://eprint.iacr.org/2013/004), but this costs size (https://link.springer.com/chapter/10.1007/978-3-319-11659-4_2), making ciphertexts even easier to attack for any particular ciphertext size.

---D. J. Bernstein

From: pqc-forum@list.nist.gov on behalf of Bobby McGee <janewaykilledtuvix@gmail.com>
Sent: Thursday, July 25, 2024 3:07 PM
To: pqc-forum
Cc: D. J. Bernstein; pqc-forum; pqc-comments; pqc-comments
Subject: [pqc-forum] Re: ROUND 4 OFFICIAL COMMENT: Classic McEliece

So

- McEliece: "maximal" error with fancy algebraic decoder,
- Kyber: simple modulation and rounding ("compress/decompress").

Maybe this question is naive or ignorant, but isn't there some way to improve LWE schemes by, e.g., first encoding a message for error-correction, then using larger noise, or maybe replacing some part of the "noisy Elgamal"-style scheme with something inspired by coding theory? I think I asked this somewhere else and was told that this had been considered and there wasn't any good trade-off, but I don't remember.

On Wednesday, July 24, 2024 at 12:24:37 PM UTC-6 D. J. Bernstein wrote:

> Summary: This comment points out a mathematical explanation for _why_
> Classic McEliece has much smaller ciphertexts than, e.g., Kyber.
>
> Context: The primary motivation for this KEM is security, but a frequent
> observation is that this KEM has much smaller ciphertexts than any of
> the alternatives. An interesting consequence is that the minimum network
> traffic for frequently used long-term post-quantum public keys (e.g.,
> server identity keys) is achieved by this KEM. Various post-quantum
> deployments, such as Rosenpass, would lose efficiency if they switched
> the long-term keys from this KEM to alternatives.
>
> The underlying cryptanalytic facts are that, for any particular
> ciphertext size, the ranking of costs of known attacks is as follows:
>
> * Fastest: attacking ciphertexts or keys for Kyber, NTRU, etc.
> * Much slower: attacking McEliece ciphertexts.
> * Slowest: attacking McEliece keys.
>
> People often wonder whether this means that the McEliece system hasn't
> been studied: i.e., whether further study would eliminate the McEliece
> ciphertext-size advantage. Normally this question is answered with a
> pointer to the long history of McEliece attack papers. I'm filing this
> comment to highlight a different answer.
>
> I've recently given a talk with a unified description of code/lattice
> cryptosystems, pinpointing what's different about the McEliece system:
>
> https://cr.yp.to/talks/2024.07.17/slides-djb-20240717-mceliece-4x3.pdf
>
> Attacks against all of these cryptosystems can be viewed as finding a
> lattice vector close to a target point. Barak similarly commented in
>
> https://eprint.iacr.org/2017/365
>
> on the "family" of "coding/lattice" systems, and claimed that "all the
> known lattice-based public-key encryption schemes can be broken using
> oracle access to an O(√n) approximation algorithm for the lattice
> closest vector problem". A closer look shows, however, that it's easy
> to write down systems where that level of approximation doesn't break
> the system. For McEliece, the gap is only polylogarithmic: the distance
> d from the target to the lattice is so high that exponentially many
> lattice points are within distance d*polylog of the target.
>
> The fundamental reason for this difference is that, for any particular
> ciphertext size, keygen and dec use a quantitatively much more powerful
> decoder for McEliece than for the alternatives. Simple attacks that
> exploit the low distances allowed by the alternative decoders are faster

From: Mike Hamburg <mike@shiftleft.org>
Sent: Thursday, July 25, 2024 5:27 PM
To: Bobby McGee
Cc: pqc-forum; D. J. Bernstein; pqc-comments
Subject: Re: [pqc-forum] ROUND 4 OFFICIAL COMMENT: Classic McEliece

Hi Bobby,

It's been tried, but as far as I know, the decoders are not nearly as powerful as McEliece's Goppa decoder.

In the KEM standardization process, at least LAC, Hila5 (later was merged into Round5) and ThreeBears were using some form of error correction, and at least early versions of NewHope did as well. This gave them slightly smaller parameters at a cost in complexity.

You could of course push things farther. As an extreme example, I've tried various combinations of RLWR and somewhat more powerful error correction (E8+large-field LDPC) under the "Glowstick" family of toy schemes. These are complete toys, and do not attempt to achieve negligible decryption failure, CCA security, or side-channel resistance: the idea is just to explore how small you can make the public key and ciphertext by using a stronger code (still not as strong a code as Goppa, but also it's a soft-decision code). In this case you can achieve 32 + 243 + 323 bytes for the nonce + public key + ciphertext for "doubtfully 128-bit" parameters, or 32 + 307 + 403 bytes for "more probably 128+-bit" parameters, etc.

A more skilled coding practitioner could probably go even farther with this.

These schemes do not approach McEliece's ciphertext size, but of course the public key is much smaller.

Regards,
-- Mike

> On Jul 25, 2024, at 9:06 PM, Bobby McGee <janewaykilledtuvix@gmail.com> wrote:
>
> So
>
> - McEliece: "maximal" error with fancy algebraic decoder,
> - Kyber: simple modulation and rounding ("compress/decompress").
>
> Maybe this question is naive or ignorant, but isn't there some way to improve LWE schemes by, e.g., first encoding a message for error-correction, then using larger noise, or maybe replacing some part of the "noisy Elgamal"-style scheme with something inspired by coding theory? I think I asked this somewhere else and was told that this had been considered and there wasn't any good trade-off, but I don't remember.
>
> On Wednesday, July 24, 2024 at 12:24:37 PM UTC-6 D. J. Bernstein wrote:
>
> Summary: This comment points out a mathematical explanation for _why_ Classic McEliece has much smaller ciphertexts than, e.g., Kyber.
>
> Context: The primary motivation for this KEM is security, but a frequent observation is that this KEM has much smaller ciphertexts than any of the alternatives. An interesting consequence is that the minimum network traffic for frequently used long-term post-quantum public keys (e.g., server identity keys) is achieved by this KEM. Various post-quantum deployments, such as Rosenpass, would lose efficiency if they switched the long-term keys from this KEM to alternatives.
>
> The underlying cryptanalytic facts are that, for any particular ciphertext size, the ranking of costs of known attacks is as follows:
>
> * Fastest: attacking ciphertexts or keys for Kyber, NTRU, etc.
> * Much slower: attacking McEliece ciphertexts.
> * Slowest: attacking McEliece keys.
>
> People often wonder whether this means that the McEliece system hasn't been studied: i.e., whether further study would eliminate the McEliece ciphertext-size advantage. Normally this question is answered with a

From: 'John Mattsson' via pqc-forum <pqc-forum@list.nist.gov>
Sent: Saturday, October 26, 2024 3:11 PM
To: pqc-forum
Subject: [pqc-forum] Round 4 (Code-based KEMs) OFFICIAL COMMENT

We strongly think NIST should standardize Classic McEliece, which has properties that makes it the best choice in many different applications. We are planning to use Classic McEliece.

- Classic McEliece is the most conservative KEM and Classic McEliece category 5 is the best option for protecting various other keys (ML-KEM, ML-DSA, SLH-DSA, FN-DSA, LMS, XMSS, etc.) in transit and storage. Classic McEliece occupies a role similar to SLH-DSA, providing a very conservative security assurance.

- The small ciphertexts and good performance makes Classic McEliece the best choice for many applications of static encapsulation keys of which there are many (WireGuard, S/MIME, IMSI encryption, File encryption, Noise, EDHOC, etc.). For many such applications, key generation time is not important, and the public key can be provisioned out-of-band. When the public key is provisioned in-band, Classic McEliece has the best performance after a few hundred encapsulations. For static encapsulation use cases where ML-KEM provides the best performance, Classic McEliece is the best backup algorithm. The memory requirement can be kept low by streaming the key.

We think NIST should standardize mceliece348864 (category 1), mceliece460896 (category 3), and one of mceliece6688128, mceliece6960119, and mceliece8192128 (category 5). 261 kB and 524 kB encapsulation keys can be used where 1 MB public keys cannot.

In addition, we think NIST should standardize one of BIKE and HQC. BIKE and HQC are the best backup algorithms to ML-KEM for ephemeral encapsulation keys. Additionally, ML-KEM+BIKE and ML-KEM+HQC hybrids seems like more conservative choices than FrodoKEM while also providing better performance. We are currently not planning to use BIKE or HQC, but we would like to see a standardized backup algorithm for ML-KEM in case attacks are found. Such a backup algorithms should have a different construction than ML-KEM. This practice of implementing independent cryptographic backup algorithms has long been a guiding principle in the telecom industry.

Cheers,
John Preuß Mattsson
Expert Cryptographic Algorithms and Security Protocols, Ericsson
--