Notes on recent speculation that a mceliece348864 key-recovery attack uses only 2^{610} operations

Classic McEliece Team

23 June 2026

Contact address: authorcontact-mceliece-merged@box.cr.yp.to

Our April 2025 report noted a long history of enthusiastic papers on "algebraic" attacks. For example, a 2002 paper [17] says that its attacks are "an important threat for ciphers" such as AES and may have cost "subexponential when the number of rounds grows", which "would be already an important breakthrough". Papers investigating algebraic attacks against the McEliece cryptosystem include [23] from 2010 (describing itself as a "breakthrough in code-based cryptography"); [19]; [22]; [35]; [33]; [21]; [18]; [4]; [26]; [32]; [34]; [29]; [36]; [3]; and [13] (describing [34], by an author of [13], as "breakthrough work").

The most recent paper [13] has title "A heuristic subexponential attack on the McEliece cryptosystem". The first version of the paper included the following claims in the abstract: "We provide a new way of performing an algebraic attack on the McEliece cryptosystem based on binary Goppa codes. ... Such matrices are then used to recover the secret algebraic structure of the code. This breaks the scheme." The wording was revised a few days later: "We provide a new way of performing an algebraic attack on the McEliece cryptosystem based on binary Goppa codes. ... Such matrices are then used to recover the secret algebraic structure of the code, from which an equivalent secret key can be efficiently derived, leading to a full key-recovery attack." The abstract claims to "demonstrate the effectiveness of our attack" for specific sizes. The abstract ends by speculating "that this attack has a complexity which is of the same nature as the distinguisher, namely subexponential in the security parameter". Page 2 of the paper says that "strong resistance to attacks was a major argument during the NIST post-quantum cryptography competition for the IND-CCA secure variant Classic McEliece".

This report points out several severe flaws in [13]. The rest of this report is organized into the following sections:

1. Contrary to what readers would expect from the words "efficiently derived, leading to a full key-recovery attack": The attacks in [13] are, according to the cost estimates in [13], much slower than a ludicrously expensive key-recovery attack already explained in the Classic McEliece documentation.

2. Contrary to what readers would expect from the words "demonstrate the effectiveness of our attack": The largest demo reported in [13] attacks a toy size $(n, t) = (482, 7)$. This demo reportedly used 64000 seconds (times an unspecified number of cores), whereas ISD software from 2008 breaks the same size $(482, 7)$ in a fraction of a second on one core.

3. The phrase "subexponential in the security parameter" in [13], summarized in the title

of [13] as a "subexponential attack on the McEliece cryptosystem", is actually referring to a strawman parameterization. The original McEliece paper and Classic McEliece use size parameters, not a security parameter.

4. [13] makes various conjectures regarding the performance of its distinguishers. However, the distinguishing property targeted in [13] is not in the original McEliece paper. It is a property that Classic McEliece explicitly avoids relying upon; not the McEliece one-wayness property that Classic McEliece explicitly relies upon; and not Classic McEliece's explicit security goal, namely IND-CCA2.

5. [13] contains speculations regarding the asymptotic performance of its attacks. However, this is attacking asymptotic claims that appear neither in McEliece's paper nor in Classic McEliece. Classic McEliece explicitly overrides asymptotics with concrete analysis.

Perhaps [13] is correct in conjecturing that its distinguisher works in only 2^{610} operations against `mceliece348864`, the smallest Classic McEliece parameter set; perhaps [13] is also correct in speculating that its key-recovery attack works in only 2^{610} operations against `mceliece348864`. This report *assumes* that the conjectures and speculations in [13] will turn out to be correct. This report instead emphasizes that [13] is missing comparisons to much faster previous algorithms, is already in its title attacking a strawman, has no effect on McEliece's original paper, has no effect on the Classic McEliece security analysis [16], and has no effect on Classic McEliece users.

The last part of this is already admitted in [13, bottom of page 7], which says that it "does not, for the time being, constitute a threat to Classic McEliece". This falls short of making the actual situation clear: [13] is simply the latest in a long line of failed attempts to break the McEliece cryptosystem.

1 The attacks have cost far above 2^{256}

Contrary to what readers would expect from the words "efficiently derived, leading to a full key-recovery attack": The attacks in [13] are, according to the cost estimates in [13], much slower than a ludicrously expensive key-recovery attack already explained in the Classic McEliece documentation.

Specifically, an attacker can recover private keys in at most 2^{256} key-generation operations by searching through all 256-bit seeds. This attack is noted in, e.g., the Classic McEliece security guide [16, page 24].

The reason this attack is not of concern is that searching through guesses for 256-bit secrets is infeasible, even with a future quantum computer. As [16, page 3] puts it, "guessing 256-bit secrets" is "not a real-world threat".

For comparison, [13, Table 6.1] estimates various numbers far above 2^{256} for the "Complexity of the distinguisher proposed here ... on the Classic McEliece parameters".

For example, the smallest Classic McEliece parameter set, `mceliece348864`, uses $(n, t) = (3488, 64)$. For this parameter set, [13, Table 6.1] estimates complexity

- 2^{515} operations using "consistency bi-degree" and a "questionable" (i.e., miscalculated; see [2, Appendix A]) assumption of exponent 2.373 for linear algebra, or
- 2^{610} operations using "consistency bi-degree" and assuming exponent 2.81, or
- 2^{454} operations using "Wiedemann's algorithm for the distinguisher using the consistency bi-degree", or
- 2^{513} operations using "critical bi-degree" and the "questionable" assumption, or
- 2^{607} operations using "critical bi-degree" and assuming exponent 2.81.

The numbers are even larger for the other Classic McEliece parameter sets.

The paper provides an argument that, with this complexity, the distinguisher has high success probability, assuming the paper's new Conjecture 1 and Conjecture 2. The paper also speculates, by extrapolation from a few tiny examples, that its key-recovery attack is not much more expensive than the distinguishers. (It is structurally clear that the key-recovery attack cannot be faster than the distinguishers; also, [13] does not claim that the distinguisher using Wiedemann's algorithm is applicable to key recovery.) Whether or not these conjectures and speculations are correct, all of these algorithms are far slower than the 2^{256}-operation key-recovery attack explained in the Classic McEliece documentation.

Unfortunately, the extremely high cost of the paper's algorithms is not mentioned in [13, pages 1–7]. There is also no comparison anywhere in [13] to the 2^{256}-operation key-recovery attack explained in the Classic McEliece documentation. What makes this missing comparison particularly surprising is that we had already pointed out that an earlier paper by one of the authors of [13] was also missing this comparison; see our April 2025 report.

There are many previous papers on infeasible methods to detect the structure of McEliece keys. The Classic McEliece security guide [16, Section 3.1] says the following:

> For all parameters of interest, the fastest attacks known against the OW-CPA problem treat G as a general matrix with no evident structure. ... There are also much slower "structural attacks" known that try to exploit the secret structure of G, for example by recovering Goppa-code parameters for G.

"Distinguishers" costing far above 2^{256}, whether or not they recover keys, are consistent with this statement and with the rest of the security analysis. [16, Section 3.6] already mentions various examples of very slow key-recovery attacks.

The "fastest attacks" comment quoted above is referring to ISD, which takes about 2^{150} bit operations for $(3488, 64)$. See [9] for precise quantification and high assurance regarding the number of bit operations used by many ISD variants.

2 The demos are for toy parameters; ISD is much faster than these demos

Contrary to what readers would expect from the words "demonstrate the effectiveness of our attack": The largest demo reported in [13] attacks a toy size $(n, t) = (482, 7)$. This demo reportedly used 64000 seconds (times an unspecified number of cores), whereas ISD software from 2008 breaks the same size $(482, 7)$ in a fraction of a second on one core.

Section 2.1 explains how to verify the performance of existing ISD software for these parameters. Section 2.2 shows that the choice $(n, t) = (482, 7)$ targeted in the demo from [13] is not a scaled-down McEliece parameter set; it instead tweaks such a parameter set to force t to be artificially small. This was already pointed out years ago: Section 2.3 reviews the introduction of these artificial sizes in [6] in 2023 and the response in [7, 8] in 2023—including a statement that algebraic attacks against these sizes would be unsurprising.

2.1 Performance of ISD for these toy parameters

Here is how to check that standard ISD attacks break this toy size $(482, 7)$ in well under one second on one core. "Break" means recovering a secret randomly generated plaintext given a public key and the corresponding ciphertext; i.e., breaking the OW-CPA property, exactly the property that Classic McEliece relies on for security (see Section 4).

The following commands download ISD software from [11] from 2008 (as part of a 2022 package that also includes an easy-to-use Python script to run attacks), download a public key and ciphertext at the toy size $(n, t) = (482, 7)$, and run the ISD software on one core to extract the corresponding plaintext:

```
wget https://cr.yp.to/software/lowweight-20220616.tar.gz
tar -xzf lowweight-20220616.tar.gz
cd lowweight-20220616
wget https://isd.mceliece.org/toy-482-7.txt
time env THREADS=1 ./attack.py < toy-482-7.txt
```

This ISD attack takes 0.17 seconds on one core of a 2.245GHz AMD EPYC 7742 with overclocking disabled. There is random variation from one run to another, as in searches for cipher keys; the 0.17 is an observed average over 100 tries. The reader can also try replacing the last line in `toy-482-7.txt` with the XOR of 7 randomly chosen previous lines (from among the lines shown and an implicit starting 63×63 identity matrix) to check that the software is able to find the secret selection of lines.

For comparison, [13] does not specify the machine used for its demo, but if it is the same as the computer "equipped with an Intel Xeon Gold 6240L (Cascade Lake-SP), x86_64, 2.60GHz, 4 CPUs/node, 18 cores/CPU" in [3] then the demo in [13] consumed 64000 seconds on 72

cores, i.e., $2^{53.4}$ core-cycles, compared to $2^{28.5}$ core-cycles for ISD; i.e., [13]'s demo was 30 million times slower than ISD at breaking OW-CPA for these toy parameters.

One can object to this focus on single-target OW-CPA as follows: recovering a private key allows each ciphertext to be broken more quickly, so spending 30 million times as much effort can be worthwhile if there are more than 30 million ciphertexts. But this objection is an artifact of taking toy parameters for which OW-CPA is easy to break in the first place. Moving up to non-toy parameters rapidly increases the cost ratio, and moving to cryptographic parameters increases the cost ratio even more. Recall, for example, that [13] reports more than 2^{400} operations for $(3488, 64)$, where ISD takes only about 2^{150} operations. A key will not be used for 2^{250} ciphertexts.

Formally, defending against key recovery per se is not a meaningful concept of security. Cryptosystem designers can trivially prevent key recovery by simply expanding the private key to include another randomly generated secret. (None of the attacks in [13] recover the "s" component of the Classic McEliece private key, for example.) One can fix this concept by considering a broader notion of "equivalent" key recovery, meaning recovery in time R of an object that allows decryption in time D per ciphertext, but notice that this broader notion also includes plaintext-recovery attacks, the case of an empty object with $R = 0$ and suitable D. Ultimately the attacker sees a pool of ciphertexts and wants to know the quantitative tradeoff between the attack resources and the number of broken ciphertexts; key recovery is just one of many attack strategies.

Unfortunately, [13] does not compare its costs to the costs of ISD.

2.2 These toy parameters started with scaled-down McEliece parameters but then artificially reduced t

The binary Goppa codes used inside McEliece's cryptosystem have code rate R when $t = (1 - R)n/\lceil \log_2 n \rceil$. The example $(n, t) = (1024, 50)$ from McEliece's original paper has code rate (approximately) 51%, with $t \approx 0.49n/\lceil \log_2 n \rceil$.

It was pointed out in [1] that a larger code rate would improve tradeoffs between size and security against known attacks. This does not mean that code rates should be close to 100%. The calculations in [11] show that, for any given limit on public-key size, security increases as code rates increase towards 80%, but then decreases as code rates increase beyond 80%.

Furthermore, one of the main defenses identified in McEliece's paper, the number of choices of Goppa polynomials, relies on t not being very small. The Classic McEliece design rationale [15, Section 1] also noted that the algebraic distinguisher from [23] relied on t being very small, i.e., on very high code rates, so staying away from small t would leave a large security margin even if the distinguisher could be upgraded to an attack. The code rates for the selected Classic McEliece sizes are 78%, 73%, 75%, 78%, 80%.

The challenges posted in [28] in 2019 include a spectrum of sizes ranging from toy sizes to serious challenges, in each case with code rate close to 80%. The smallest 28 challenges

have been broken so far, the largest of those having $(n, t) = (1409, 26)$. Any new attack that is even slightly competitive should be able to demonstrate its performance on these scaled-down cryptosystem challenges; requiring such demos helps weed out erroneous claims of improved attacks.

But the choice $(482, 7)$ targeted in [13] is not one of the challenges from [28]. This choice has code rate 87%; i.e., t is only $0.13n/\lceil \log_2 n \rceil$. These are not scaled-down McEliece parameters, such as the toy size $(n, t) = (482, 11)$ from [28]. Instead the size $(n, t) = (482, 7)$ artificially reduces t, making attacks—especially algebraic attacks—much easier.

It is interesting to compare this to an incident from the 1990s. A Eurocrypt 1991 paper "Cryptanalysis of McEliece's public-key cryptosystem" [27] claimed an "efficient (i.e., polynomial time) algorithm for decoding an arbitrary linear code up to its guaranteed error-correcting limit"; claimed to break size $(n, t) = (1024, 37)$ of the McEliece system in "about 60 hours" on an IBM PC (i.e., about 2^{40} CPU cycles); and claimed that "During Eurocrypt '91, the above described cryptanalytic attack on the McEliece public-key cryptosystem was demonstrated". In 1997, Berson [12] wrote "There was some excitement and confusion about the cryptanalysis of the McEliece public-key cryptosystem a few years ago" and wrote that "the demonstration was only a toy", namely attacking $(n, t) = (63, 5)$. Notice that this toy has code rate 52%, so at least it was a reasonably scaled-down version of McEliece's original parameters.

2.3 Mislabeling of the artificial parameters

The new paper [13] does not describe its demos as attacking toy parameters rapidly broken by ISD (see Section 2.1), nor does it mention the artificially small choices of t (see Section 2.2).

Instead it describes the demos as follows: "We demonstrate the effectiveness of our attack on McEliece TII challenges, some of which having been studied in [BLT26], and aimed at having 83,89,119,166, 210 and even 248 bit security respectively ... *A heuristic subexponential attack that has been verified on TII challenges.* ... We have verified this attack both on TII challenges that were treated in [BLT26], namely instances claiming a security of 83, 89, 119, 166 and 210 bits against key attacks. In addition, our attack did break Instance 248 in less than a day, which clearly outperforms [BLT26] and sets a new record. ... Our modeling successfully broke this instance at degree 5 in less than a day (64000 seconds), with a total memory cost of more than 700GB."

The actual sizes $(n, t) = (482, 7)$ for "Instance 248" finally appear in [13, page 27, Table A.3], at which point a reader skeptical of this "record" verifies that ISD has much better performance; see Section 2.1. But the reader still wonders why artificial parameters—not scaled-down cryptosystem parameters, but instead adjustments to those to artificially reduce t—were presented as challenges in the first place.

TII announced its "TII McEliece challenges" on 28 May 2023 [6] as supposedly helping "enhance our understanding of the hardness of the problems that underpin the security of modern versions of the McEliece Cryptosystem, such as Classic McEliece". This claim of

relevance was disputed by Bernstein [7] on 29 May 2023:

> The key-recovery parameter sets in
>
> https://github.com/ElenaKirshanova/tii_decoding_challenge/blob/main/key_rec_instances
>
> generally have surprisingly small choices of t, and thus surprisingly few choices of the Goppa polynomial g, so they're omitting one of the central defenses in McEliece's original cryptosystem.
>
> I'm also skeptical of the security estimates reported in the same file. Reverse-engineering the numbers detects an underlying assumption that support splitting for multiple support sets requires each possibility for the support set to be handled separately---but the Classic McEliece submission already questions exactly this assumption, saying "there could be ways to merge work across possibilities". Concretely, it would not be surprising for the algebraic techniques of
>
> https://theses.hal.science/tel-01678829
> https://eprint.iacr.org/2022/1715
>
> to end up recovering keys more quickly than estimated in
>
> https://github.com/ElenaKirshanova/tii_decoding_challenge/blob/main/key_rec_instances
>
> for the parameter sets in that file.
>
> It is, in any case, definitely not true that Classic McEliece relies on this questionable assumption. On the contrary, Classic McEliece is designed so that its QROM IND-CCA2 security relies purely on the OW-CPA security of the original McEliece system; and the original McEliece system has n=q, so there's only one possibility for the support set to begin with. Section 4.1 of
>
> https://classic.mceliece.org/mceliece-security-20221023.pdf
>
> shows that the security of n<=q in Classic McEliece follows from the security of n=q as in McEliece's original system, and gives examples of calculating the (very small) tightness loss in the reduction.
>
> To avoid confusion, I recommend changing the "McEliece key recovery" and "Security Foundation" labels for this challenge, changing the description of these parameter sets as merely being "reduced-size instances", correcting the claim that this is a problem "on which the security of McEliece is based", naming the source of the new security

> estimates, acknowledging that the assumption underlying these estimates
> was already questioned in the Classic McEliece submission, and citing at
> least some representative examples of the extensive previous literature
> on structural attacks.

On 2 June 2023, replying to TII's followup comment "We agree that our bit security estimates might not precisely capture the difficulty of these challenges", Bernstein [8] wrote the following:

> My concern here is not with the level of precision. My concern is with
> new strawman security estimates being incorrectly labeled as being part
> of the security foundation of the McEliece system.
>
> If these new estimates are (unsurprisingly) broken by the techniques
> that I cited, then the "security foundation"/"McEliece"/... labeling
> will make people believe, incorrectly, that the break is a problem for
> the McEliece system. The labeling should be fixed.
>
> The new FAQ entry exacerbates this problem by indicating, incorrectly,
> that these challenges are simply the result of "down-scaling" normal
> parameter selection to reach "approachable" sizes. This was already
> suggested by the "reduced-size" terminology but is now more explicit.
>
> As a concrete example, describing the challenge
>
>   {"n": 1008, "k": 898, "t": 11, "m": 10, "bitcomplexity": 252.052342314491}
>
> as the result of "down-scaling" makes people think, incorrectly, that
> this 2^252 is the result of scaling down security estimates that the
> literature gives for cryptographic sizes.
>
> Even if the "bitcomplexity" number is ignored, the "t" is surprisingly
> small. One consequence is that the rate k/n = 898/1008 is very close to
> the 92% cutoff for the https://eprint.iacr.org/2010/331 distinguisher.
> It's well known that a feasible search will change the numbers a bit so
> such cutoffs shouldn't be treated as sharp edges.
>
> I'm not saying distinguishers necessarily turn into attacks. The point
> is that this parameter set is not following normal parameter-selection
> procedures. Normal parameter selection uses rates between 70% and 80%,
> with much larger values of t than this challenge, many more choices for
> the secret Goppa polynomial, and a much larger security margin against
> key recovery.
>
> For example, the smallest selected Classic McEliece parameter set has

```
(n,k,t) = (3488,2720,64), with rate 78%. The recommended 6688 and 6960
parameter sets have rates 75% and 78% respectively. The reasonably
scaled parameter set in https://decodingchallenge.org/goppa/record/26
has (n,k,t) = (1347,1072,25), with rate 80%.
```

TII never answered any of this. TII continued labeling, e.g., its artificial $(482, 7)$ size with rate 87% as having "security" 2^{248}, and continued prominently describing the goal of its challenges as understanding "the hardness of the problems that underlie the security of the modern versions of the McEliece Cryptosystems like Classic McEliece". [13] then copied this mislabeling, and—as [8] had already predicted—incorrectly portrayed a faster algebraic attack against this size as demonstrating a problem for the McEliece system.

3 The security-parameter claims are attacking a strawman

The phrase "subexponential in the security parameter" in [13], summarized in the title of [13] as a "subexponential attack on the McEliece cryptosystem", is actually referring to a strawman parameterization. The original McEliece paper and Classic McEliece use size parameters, not a security parameter.

Section 3.1 reviews the standard Goldwasser–Micali definition of a security parameter, gives a simple example of a cryptosystem that can be fit into the Goldwasser–Micali model, and uses this example to illustrate that asking whether an attack is "subexponential in the security parameter" is a meaningless question about cryptosystems that fit this model.

Section 3.2 reviews common reasons to avoid the Goldwasser–Micali model, and in particular reviews the usage of size parameters—not a security parameter—in the original McEliece cryptosystem and in Classic McEliece.

3.1 The standard definition of a security parameter

Goldwasser and Micali [24] defined an asymptotic form of cryptosystem security in terms of (randomized) algorithms taking (average) time polynomial in a "security parameter" λ. Each cryptosystem operation is required to run in time $\leq P(\lambda)$ for some polynomial P. Security is defined to mean that there cannot be any (A, Q, R) such that A is an attack taking time $\leq Q(\lambda)$ and having reciprocal success probability $\leq R(\lambda)$ for polynomials Q, R. The definitions in [24] were for public-key encryption, but the same concept was then extended to a much wider range of cryptographic operations. See, e.g., the definition of signature systems in [25] by Goldwasser, Micali, and Rivest.

Fitting a cryptosystem into the Goldwasser–Micali model requires choosing a mapping from a security parameter λ to the cryptosystem size, with the objective of legitimate computations taking time polynomial in λ while attack costs are superpolynomial in λ. The following

simple example illustrates how two different choices of mapping for the same cryptosystem (1) achieve this objective either way but (2) produce different conclusions as to whether the usual attacks take subexponential time in the security parameter.

The cryptosystem in this example is textbook RSA signing with exponent 3. The private key has two different prime numbers p, q between 2^{B-1} and 2^B. The public key is pq. A signature on a message $m \in \{0, 1, \ldots, 2^{2B-2} - 1\}$ is a cube root of m modulo pq.

This signature system is broken rapidly by adaptive attacks (e.g., multiplying a message and signature by 8 and 2 respectively modulo pq), existential-forgery attacks (e.g., message 8 has signature 2), and quantum attacks. But assume that the security goal is merely to protect against non-quantum non-adaptive universal forgery, and assume that the best attack is then to use NFS to factor pq into (p, q).

Fitting this signature system into the Goldwasser–Micali model (more precisely, Goldwasser–Micali–Rivest for signatures) requires mapping a security parameter, an integer $\lambda \in \{1, 2, 3, \ldots\}$, to the number of bits B in the primes p, q. The first choice of mapping considered here is $B = (\lambda+1)^4$. The second choice of mapping considered here is $B = (\lfloor \lambda^{1/4} \rfloor + 1)^4$. These two choices cover exactly the same set of B. There are many other possible choices of mapping, but comparing these two choices will be sufficient for demonstrating that "subexponential in the security parameter" is ill-defined.

Recall that NFS takes time exponential in $B^{1/3+o(1)}$ as $B \to \infty$. With the first mapping, this time is exponential in $\lambda^{4/3+o(1)}$, certainly superpolynomial in λ, so this signature system reaches its security goal in this asymptotic model. With the second mapping, this time is exponential in $\lambda^{1/3+o(1)}$, again superpolynomial in λ, so this signature system again reaches its security goal in this asymptotic model.

The different choice of mapping has no effect on the Goldwasser–Micali model. Either way, the cryptosystem operations take polynomial time while the best attack, NFS, does not. However, NFS has cost *superexponential* in the security parameter with the first choice, and cost *subexponential* in the security parameter with the second choice.

Consequently, asking whether the cost is subexponential in the security parameter is asking about a superficial labeling choice that is irrelevant to the model and that has nothing to do with the security of this cryptosystem.

As another example of subexponentiality being irrelevant to the standard security-parameter concept, consider Goldwasser–Micali–Rivest [25, Section 6.3] hypothesizing that every *polynomial-time* algorithm for factoring pq into p, q, when p, q are λ-bit primes congruent to 3 and 7 respectively modulo 8, has reciprocal success probability larger than any *polynomial* in the security parameter λ; [25] then builds a signature system on top of this hypothesis. It was already known (and even proven without heuristics—see [20]) that factoring integers below $2^{2\lambda}$ takes time subexponential in λ. This does not contradict the hypothesis from [25]: those factoring algorithms have superpolynomial subexponential run time.

3.2 Cryptosystems beyond the security-parameter model

The purpose of the standard Goldwasser–Micali security-parameter concept reviewed above is to fit cryptography into the theory of polynomial-time algorithms. The set of polynomial-time algorithms has features that simplify proofs, such as being closed under composition. However, this approach also has drawbacks:

- Security in the Goldwasser–Micali model inherently requires unbounded cryptosystem sizes. Most symmetric cryptosystems and many public-key cryptosystems provide only a finite list of sizes: for example, AES has only 128-bit, 192-bit, and 256-bit keys, and X25519 has just one size. As Bellare, Canetti, and Krawczyk wrote in [5] in 1996: "Here, however, we deal with primitives that consist of a *single instance*: there is a single MD5, a single SHA and a single DES. Asymptotic behavior in the security parameter has no meaning here."
- Even for a cryptosystem that provides unbounded sizes, such as RSA, any particular ciphertext or signature uses a particular cryptosystem size, such as RSA-2048. The Goldwasser–Micali model cannot express the security of any particular size. Cryptanalysis papers typically look at concrete attack costs, not just at whether attacks take polynomial time.

Many cryptosystem designs do not have a security parameter and do not attempt to fit the Goldwasser–Micali model. Cryptosystems often instead have one or more *size* parameters—perhaps bounded, perhaps not. The literature analyzes the concrete costs of attacks against proposed cryptosystem sizes, considers risks of improved attacks against those sizes, and, in some cases, makes recommendations for acceptable cryptosystems, including recommendations of the sizes to use.

McEliece's original paper [30] predates the Goldwasser–Micali paper and in any case does not have a security parameter. It has two parameters "n and t" specifying the size of an underlying binary Goppa code. McEliece writes that recovering the private key "seems hopeless if n and t are large enough because there are so many possibilities for G [the Goppa polynomial], not to mention the possibilities for S and P". McEliece also states an expectation that attacking a ciphertext will also be infeasible "if the code parameters are large enough". McEliece writes "In particular, suppose we chose $n = 1024 = 2^{10}$, $t = 50$", considers some specific attacks for that size, and estimates 2^{65} bit operations for the fastest of those attacks, a straightforward form of ISD.

Retroactively declaring n and t in [30] to be "security parameters" would not fit the Goldwasser–Micali model (and cannot support the claims in [13] about "the security parameter" of the McEliece cryptosystem). The model requires a single security parameter, an unbounded positive integer; it defines security as a superpolynomial decrease of attack effectiveness as this integer grows. For McEliece, t and n are each unbounded but t has a limited range depending on n, and McEliece's security analysis already shows that security disappears when t is at either end of the range.

There are many ways that one *could* add a security parameter to McEliece's cryptosystem,

and that one could articulate a goal of security exponential in the security parameter. But this is a strawman, not what McEliece did.

Classic McEliece also does not have a security parameter. Classic McEliece chooses specific McEliece parameter sets (n, t) to meet various size constraints, such as public keys fitting into 1MB. See, e.g., the statement in [14] that the $(6688, 128)$ parameter set is designed for "optimal security within 2^{20} bytes if n and t are required to be multiples of 32"; here "2^{20} bytes" refers to the public-key size, and "optimal security" refers to the balance between n and t (see [11] and Section 2.2). The Classic McEliece rationale [15, Section 5.2] explains that "choosing parameter sets to maximize security subject to a specified size constraint is much more robust than choosing parameter sets to minimize size for a specific target security level".

Retroactively declaring the size limit for the Classic McEliece public key to be "the security parameter" would run into yet another definitional problem: Classic McEliece has further parameters for which it specifies fixed choices (256-bit secret keys, SHAKE256 as the hash function, etc.). Classic McEliece explicitly disclaims security beyond those sizes ("all of the selected KEM parameter sets use 256-bit secrets, even when the parameters could otherwise reach higher security levels").

With enough effort, one could extend Classic McEliece to specify unbounded sizes, defining variable-size hash functions etc. in terms of the size limit on the public key, and then articulate a goal of security exponential in the size limit. But this is a strawman, not what Classic McEliece does. Furthermore, many standard McEliece attacks have cost subexponential in the McEliece public-key size; i.e., this strawman was already dead.

An earlier paper by one of the authors of [13] made asymptotic claims about "the security parameter in the Classic McEliece system". We noted in reply that Classic McEliece does not use a "security parameter". See our April 2025 report.

4 The distinguisher claims are attacking a strawman

[13] makes various conjectures regarding the performance of its distinguishers. However, the distinguishing property targeted in [13] is not in the original McEliece paper. It is a property that Classic McEliece explicitly avoids relying upon; not the McEliece one-wayness property that Classic McEliece explicitly relies upon; and not Classic McEliece's explicit security goal, namely IND-CCA2.

A top-down view of the security analysis in the Classic McEliece security guide is as follows:

- The Classic McEliece security goal is IND-CCA2. See [16, Section 1]. (We recommend using separate modules for generic transformations that aim at anything beyond an IND-CCA2 KEM; see [16, Section 6].)
- IND-CCA2 attacks much faster than QROM IND-CCA2 attacks would be surprising for the selected hash function (SHAKE256). See [16, Section 5.3.3]. This lets reviewers

focus on QROM IND-CCA2 attacks.

- QROM IND-CCA2 security follows tightly from the hypothesis of OW-CPA security (i.e., one-wayness) of the underlying PKE. See [16, Section 5]. After checking this, reviewers can focus on OW-CPA security.
- OW-CPA security of the underlying PKE follows tightly from the hypothesis of OW-CPA security of the original McEliece PKE. See [16, Section 4]. This lets reviewers focus on McEliece OW-CPA security.
- [16, Section 3] reviews the state of the art in McEliece OW-CPA attacks.

This provides a clear structure for Classic McEliece security reviews. The only ways that something can possibly go wrong are some sort of disaster involving SHAKE256, a mistake in the tight reductions, or a better OW-CPA attack against the original McEliece system.

The literature sometimes mentions the following trivial implication: McEliece OW-CPA security follows from OW-CPA security of a uniform random matrix if the McEliece public key is indistinguishable from a uniform random matrix. Of course, finding a distinguishing attack faster than the target OW-CPA security level would render this implication vacuous.

The Classic McEliece security guide [16, Section 3.1] notes this motivation for studying distinguishing attacks, but it also notes that this is "not a two-way implication—perhaps there are distinguishers even if OW-CPA is secure". A fast high-probability distinguisher would not change any of the Classic McEliece security claims.

As a (real) pre-quantum analogy for such (hypothetical) distinguishers, consider the following trivial implication: if RSA public keys are indistinguishable from uniform random integers of the same size then OW-CPA for an RSA public key follows from OW-CPA for the same system with a uniform random integer. This is vacuous, since RSA keys are efficiently distinguishable from uniform. OW-CPA is nevertheless a reasonable pre-quantum hypothesis for RSA. The distinguisher does not threaten any of the security properties that are normally claimed for RSA.

In other words, for an algorithm to have an effect on the Classic McEliece security analysis, it would have to be much faster than the distinguishers in [13] *and* would have to be a key-recovery attack or something else that breaks OW-CPA.

We had already pointed this out in response to an earlier paper by one of the authors of [13]. See our April 2025 report. But most of [13] continues to focus on distinguishers, except for a short section at the end speculating about the performance of key recovery; see Section 5 below.

5 The asymptotic claims are attacking a strawman

[13] contains speculations regarding the asymptotic performance of its attacks. However, this is attacking asymptotic claims that appear neither in McEliece's paper nor in Classic

McEliece. Classic McEliece explicitly overrides asymptotics with concrete analysis.

Specifically, when [13] says that it has a "Heuristic Subexponential Attack on the McEliece Cryptosystem ... subexponential in the security parameter ... for the *first time* a *subexponential attack* on the McEliece cryptosystem", it is highlighting the portions of [13] that go beyond distinguishers, and it is telling the reader that the security level was previously believed to be exponential in the security parameter. As noted above,

- "the security parameter" is a strawman (see Section 3),
- the attack is much slower than previous attacks at toy sizes (see Section 2), and
- the attack is even less competitive at cryptographic sizes (see Section 1),

but imagine the claim being rephrased as merely claiming an *asymptotically faster* attack than what was previously believed possible. Asymptotically faster means faster for all sufficiently large sizes; this is not contradicted by being slower for all sizes of interest.

This is still attacking a strawman. The claim does not contradict anything in McEliece's original paper [30] or the Classic McEliece security guide [16]; see Section 5.1. Section 5.2 provides another way to see that the claims of [13] are irrelevant to Classic McEliece.

5.1 Concrete security goals

The review of McEliece OW-CPA attacks in the Classic McEliece security guide [16, Section 3] begins by reviewing the McEliece OW-CPA problem and continues in [16, Section 3.1] with a statement also quoted above: "For all parameters of interest, the fastest attacks known against the OW-CPA problem treat G as a general matrix with no evident structure." Structural attacks that are slower than ISD for all parameters of interest do not require any updates in this statement.

The next sentence is as follows: "Sections 3.2, 3.3, 3.4, and 3.5 review the performance of algorithms to recover a from $(G, Ga + e)$ for general matrices G." Specifically, those subsections are "3.2 Prange's information-set-decoding algorithm"; "3.3 Improvements to information-set decoding"; "3.4 Asymptotic costs of information-set decoding"; and "3.5 Concrete costs of information-set decoding".

In particular, [16, Section 3.5] begins by emphasizing that asymptotics do not state attack costs for any specific cryptosystem sizes: "It is important to realize that $o(1)$ does not mean 0: it means something that converges to 0 as $n \to \infty$. More detailed attack-cost evaluation is therefore required for any particular parameters."

The same point appears in many papers on concrete security. For example, Micali and Reyzin wrote the following in [31, author version] in 2000: "Goldwasser, Micali and Rivest's ([GMR88]) classical notion of security for a digital signature scheme is asymptotic in nature. ... It has been often pointed out that this asymptotic approach, which uses notions such as 'polynomial time' and 'sufficiently large,' is too coarse for practical security recommendations. Knowing that no polynomial-time adversary has a better than exponentially

small chance of forgery for a sufficiently large security parameter does not provide one with an answer to the practical problem of finding the appropriate security parameters to ensure security against adversaries with certain concrete capabilities."

Assume that [13] is correct in conjecturing that its distinguisher has exponent $\Theta(n(\log\log n)^3/(\log n)^2)$ for asymptotically constant-rate Goppa codes (see [13, page 32, formula (7.1)]), and in speculating that its key-recovery attack also has exponent $\Theta(n(\log\log n)^3/(\log n)^2)$. This exponent is asymptotically smaller than the exponent $\Theta(n/\log n)$ for information-set decoding: no matter what the asymptotic Θ constants are, those constants will for sufficiently large n be overwhelmed by the ratio $(\log\log n)^3/\log n$ converging to 0. For example, $(\log\log n)^3/\log n$ is below 1 for all $n > 2^{134.69}$, is below 1/2 for all $n > 2^{666.39}$, and is below 1/4 for all $n > 2^{2323.71}$.

These statements about uninteresting parameters have no effect on the Classic McEliece security analysis. They also have no effect on McEliece's original paper, which aimed for security "if the code parameters are large enough"—i.e., being able to reach cryptographically useful sizes. This is a concrete goal, not an asymptotic goal.

5.2 Asymptotic McEliece

Finally, imagine a cryptosystem Asymptotic McEliece that adds support for arbitrarily large sizes to Classic McEliece as follows.

It was shown in [10] that generalizing McEliece to allow higher genus, and allowing the genus to gradually increase with n while t is chosen appropriately, gradually increases security against ISD. For $n > 10000$, use higher-genus McEliece for Asymptotic McEliece, and adjust other parameters appropriately to support arbitrarily large values of n—e.g., replace SHAKE256 with something that scales to arbitrary sizes. For $n \leq 10000$, simply use genus 0, as in Classic McEliece, copying the existing Classic McEliece parameter sets for Asymptotic McEliece. Optimizing the cutoff 10000 is unnecessary for purposes of this section.

Asymptotically, higher-genus McEliece has $t \in \Theta(n)$, and has ISD exponent $\Theta(n)$. There appears to be a quadratic effect of t on the exponent of [13] (also explaining why the demo from [13] used an artificially reduced t; see Section 2.2), so presumably higher-genus McEliece increases the cost of [13] beyond $\Theta(n)$, eliminating the claimed asymptotic advantage of [13] over ISD.

Suppose the claimed asymptotic advantage is viewed as a problem for Classic McEliece. This problem disappears for Asymptotic McEliece. In particular, this is not a problem for the `asymptoticmceliece6960119` parameter set. But this is the same as `mceliece6960119`, so there is no problem for `mceliece6960119`. Same for the other Classic McEliece parameter sets, all of which have $n < 10000$. Consequently, there was no problem for Classic McEliece in the first place.

References

[1] Carlisle M. Adams and Henk Meijer. Security-related comments regarding McEliece's public-key cryptosystem. *IEEE Transactions on Information Theory*, 35(2):454–455, 1989. doi:10.1109/18.32140.

[2] Josh Alman and Hantao Yu. Improving the leading constant of matrix multiplication. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 1933–1971. SIAM, 2025. URL: https://arxiv.org/abs/2410.20538v1, doi:10.1137/1.9781611978322.61.

[3] Magali Bardet, Axel Lemoine, and Jean-Pierre Tillich. An attack on the CFS scheme and on TII McEliece challenges. 2026. URL: https://eprint.iacr.org/2026/430.

[4] Magali Bardet, Rocco Mora, and Jean-Pierre Tillich. Polynomial time key-recovery attack on high rate random alternant codes. *IEEE Transactions on Information Theory*, 70(6):4492–4511, 2024. URL: https://arxiv.org/abs/2304.14757.

[5] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th Annual Symposium on Foundations of Computer Science, FOCS 1996, Burlington, Vermont, USA, 14–16 October, 1996*, pages 514–523. IEEE Computer Society, 1996. doi:10.1109/SFCS.1996.548510.

[6] Emanuele Bellini, Andre Esser, and Elena Kirshanova. TII McEliece Challenges, May 2023. URL: https://web.archive.org/web/20260616122319/https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/KzmgtZ3qBME/m/bGh8ed_-AQAJ.

[7] Daniel J. Bernstein. Re: TII McEliece Challenges, May 2023. URL: https://web.archive.org/web/20260616121832/https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/KzmgtZ3qBME/m/fBZ6p7MSAgAJ.

[8] Daniel J. Bernstein. Re: TII McEliece Challenges, June 2023. URL: https://web.archive.org/web/20260616122115/https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/KzmgtZ3qBME/m/LTiMRSnKAQAJ.

[9] Daniel J. Bernstein and Tung Chou. CryptAttackTester: high-assurance attack analysis. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology—CRYPTO 2024—44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part VI*, volume 14925 of *Lecture Notes in Computer Science*, pages 141–182. Springer, 2024. URL: https://cat.cr.yp.to.

[10] Daniel J. Bernstein, Tanja Lange, and Alex Pellegrini. Higher-genus McEliece. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology—ASIACRYPT 2025—31st International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, VIC, Australia, December 8–12, 2025, Proceedings, Part IV*, volume 16248 of *Lecture Notes in Computer Science*, pages 498–531. Springer, 2025. doi:10.1007/978-981-95-5113-2_16.

[11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the McEliece cryptosystem. In Johannes A. Buchmann and Jintai Ding, editors, *Post-Quantum Cryptography, Second International Workshop, PQCrypto 2008, Cincinnati, OH, USA, October 17–19, 2008, Proceedings*, volume 5299 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2008. https://eprint.iacr.org/2008/318. doi:10.1007/978-3-540-88403-3_3.

[12] Thomas A. Berson. Failure of the McEliece public-key cryptosystem under message-resend and related-message attack. In Burton S. Kaliski Jr., editor, *Advances in Cryptology—CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17–21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 213–220. Springer, 1997. https://doi.org/10.1007/BFb0052237. URL: https://doi.org/10.1007/BFb0052223, doi:10.1007/BFb0052237.

[13] Pierre Briaud, Axel Lemoine, Hugues Randriambololona, and Jean-Pierre Tillich. A heuristic subexponential attack on the McEliece cryptosystem, 2026. URL: https://eprint.iacr.org/2026/1232.

[14] Classic McEliece Team. Classic McEliece: conservative code-based cryptography: modifications for round 2, 2019. URL: https://classic.mceliece.org/nist/mceliece-20190331-mods.pdf.

[15] Classic McEliece Team. Classic McEliece: conservative code-based cryptography: design rationale, 2022. URL: https://classic.mceliece.org/mceliece-rationale-20221023.pdf.

[16] Classic McEliece Team. Classic McEliece: conservative code-based cryptography: guide for security reviewers, 2022. URL: https://classic.mceliece.org/mceliece-security-20221023.pdf.

[17] Nicolas T. Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In Yuliang Zheng, editor, *Advances in Cryptology—ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1–5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287. Springer, 2002. URL: https://iacr.org/archive/asiacrypt2002/25010267/25010267.pdf.

[18] Alain Couvreur, Rocco Mora, and Jean-Pierre Tillich. A new approach based on quadratic forms to attack the McEliece cryptosystem. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology—ASIACRYPT 2023—29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4—8, 2023, Proceedings, Part IV*, volume 14441 of *Lecture Notes in Computer Science*, pages 3–38. Springer, 2023. URL: https://eprint.iacr.org/2023/950.

[19] Alain Couvreur, Ayoub Otmani, and Jean-Pierre Tillich. Polynomial time attack on wild McEliece over quadratic extensions. In Phong Q. Nguyen and Elisabeth Oswald, editors,

Advances in Cryptology—EUROCRYPT 2014—33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11–15, 2014. Proceedings, volume 8441 of *Lecture Notes in Computer Science*, pages 17–39. Springer, 2014. URL: https://eprint.iacr.org/2014/112.

[20] John D. Dixon. Asymptotically fast factorization of integers. *Mathematics of Computation*, 36:255–260, 1981. URL: https://pubs.ams.org/mcom/1981-36-153/S0025-5718-1981-0595059-1/.

[21] Freja Elbro and Christian Majenz. An algebraic attack against McEliece-like cryptosystems based on BCH codes. In *IEEE Information Theory Workshop, ITW 2023, Saint-Malo, France, April 23–28, 2023*, pages 70–75. IEEE, 2023. URL: https://eprint.iacr.org/2022/1715.

[22] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, Frédéric de Portzamparc, and Jean-Pierre Tillich. Structural cryptanalysis of McEliece schemes with compact keys. *Designs, Codes and Cryptography*, 79(1):87–112, 2016. URL: https://eprint.iacr.org/2014/210.

[23] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high-rate McEliece cryptosystems, 2010. URL: https://eprint.iacr.org/archive/2010/331/1275979644.pdf.

[24] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. doi:10.1016/0022-0000(84)90070-9.

[25] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988. doi:10.1137/0217017.

[26] Sabira El Khalfaoui, Mathieu Lhotel, and Jade Nardi. Goppa-like AG codes from $C_{a,b}$ curves and their behavior under squaring their dual. *IEEE Transactions on Information Theory*, 70(5):3330–3344, 2024. doi:10.1109/TIT.2023.3334096.

[27] Valery I. Korzhik and Andrew I. Turkin. Cryptanalysis of McEliece's public-key cryptosystem. In Donald W. Davies, editor, *Advances in Cryptology—EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8–11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 68–70. Springer, 1991. doi:10.1007/3-540-46416-6_5.

[28] Julien Lavauzelle, Matthieu Lequesne, and Nicolas Aragon. Syndrome decoding in the Goppa-McEliece setting, 2019. URL: https://decodingchallenge.org/goppa/.

[29] Axel Lemoine, Rocco Mora, and Jean-Pierre Tillich. Understanding the new distinguisher of alternant codes at degree 2. *Designs, Codes and Cryptography*, 93(8):3083–3105, 2025. URL: https://doi.org/10.1007/s10623-025-01626-8, doi:10.1007/S10623-025-01626-8.

[30] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. Technical report, NASA, 1978. https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF.

[31] Silvio Micali and Leonid Reyzin. Improving the exact security of digital signature schemes. *Journal of Cryptology*, 15(1):1–18, 2002. URL: https://www.cs.bu.edu/~reyzin/papers/exact-security.pdf, doi:10.1007/S00145-001-0005-8.

[32] Rocco Mora. On the matrix code of quadratic relationships for a Goppa code. *Advances in Mathematics of Communications*, 19(3):829–852, 2025. URL: https://doi.org/10.3934/amc.2024026, doi:10.3934/AMC.2024026.

[33] Rocco Mora and Jean-Pierre Tillich. On the dimension and structure of the square of the dual of a Goppa code. *Designs, Codes and Cryptography*, 91(4):1351–1372, 2023. URL: https://doi.org/10.1007/s10623-022-01153-w, doi:10.1007/S10623-022-01153-W.

[34] Hugues Randriambololona. The syzygy distinguisher. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology—EUROCRYPT 2025—44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4–8, 2025, Proceedings, Part VI*, volume 15606 of *Lecture Notes in Computer Science*, pages 324–354. Springer, 2025. doi:10.1007/978-3-031-91095-1_12.

[35] Mohamed Ahmed Saeed. *Algebraic approach for code equivalence. (Approche algébrique sur l'équivalence de codes)*. PhD thesis, Normandy University, France, 2017. URL: https://tel.archives-ouvertes.fr/tel-01678829.

[36] Andreas Wiemers and Tobias Hemmert. Distinguishing Goppa codes using higher-order vanishing, 2025. URL: https://eprint.iacr.org/2025/1661.