Grover vs. McEliece

Daniel J. Bernstein [⋆]

Department of Computer Science (MC 152)
The University of Illinois at Chicago
Chicago, IL 60607–7053
djb@cr.yp.to

Abstract. This paper shows that quantum information-set-decoding attacks are much faster than non-quantum information-set-decoding attacks.

1 Introduction

Assume that large quantum computers are built, and that they scale as smoothly as one could possibly hope. Shor's algorithm and its generalizations will then completely break RSA, ECDSA, and many other popular cryptographic systems: for example, a quantum computer will find an RSA secret key at essentially the same speed that an RSA user can apply the key. See [21] for Shor's original algorithm, [24] for a detailed complexity analysis, and [16] for a survey of generalizations.

It seems inconceivable, however, that Shor's idea will ever have any relevance to the vast majority of "one-way" cryptographic systems: secret-key stream ciphers, hash-based public-key signature systems, etc. There are, furthermore, several "trapdoor one-way" cryptographic systems—most importantly, public-key encryption systems—that seem immune to Shor's algorithm. The oldest example is the code-based public-key system introduced by McEliece in [19] thirty years ago.

The conventional wisdom is that all of these systems will nevertheless need to double their key sizes in order to survive quantum computers. Shor's algorithm is not the only application of quantum computers! A quantum algorithm introduced by Grover in [13] and [14] finds (e.g.) a 256-bit AES key in only about 2^{128} quantum operations, given a few known plaintexts encrypted under that key. Users who want to push the attacker's cost significantly higher than 2^{128}—the original motivation for

[⋆] Permanent ID of this document: `e2bbcdd82c3e967c7e3487dc945f3e87`. Date of this document: 2009.11.23. This work was supported in part by the National Science Foundation under grant ITR–0716498, and in part by Cisco's University Research Program.

256-bit AES—will need a cipher with significantly more than a 256-bit key.

There is, however, no reason to think that the doubled key size for a secret-key cipher will be matched by a doubled hash-function output length, a doubled key size for McEliece, etc. Consider the following examples:

- An often-quoted paper [8] by Brassard, Høyer, and Tapp argues that quantum computers force a $1.5\times$ expansion in the output length of a collision-resistant hash function. My new paper [3] argues that quantum computers actually have no impact on the difficulty of finding hash collisions.
- Quantum computers obviously have no impact on the key length required for the Gilbert–MacWilliams–Sloane authenticator [11] and other information-theoretically secure cryptographic systems.
- Overbeck and Sendrier argue in [20, Section 3.5] that quantum computers have only a small impact on the McEliece public-key system, reducing the attacker's decoding cost from (e.g.) 2^{140} to 2^{133} for a code of length 4096 and dimension $4096 - 45 \cdot 12 = 3556$.

Grover's algorithm takes only square-root time compared to a brute-force key search, but this does not mean that it takes less time than the more sophisticated algorithms used to find hash collisions, McEliece error vectors, etc. Sometimes Grover's idea can be used to speed up the more sophisticated algorithms, but understanding the extent of the speedup requires careful analysis.

Contents of this paper. This paper points out a way to attack the McEliece system with Grover's algorithm. This attack is asymptotically much more effective than the approach analyzed in [20, Section 3.5].

State-of-the-art information-set-decoding algorithms take time just $c^{(1+o(1))n/\lg n}$ to break a length-n rate-R code, where $c = 1/(1-R)^{1-R}$; see [6] for a much more detailed analysis. What this paper shows is that quantum versions of the same information-set-decoding algorithms take time only $c^{(1/2+o(1))n/\lg n}$. Protecting against this attack requires replacing n by $(2+o(1))n$, essentially *quadrupling* the McEliece key size.

Users who were already making the worst-case assumption regarding the impact of Grover's algorithm, namely a square-root speedup in all attacks, will not be affected by this paper. However, users who were making more optimistic assumptions to reduce key size, decryption time, etc. will need to change their parameters in the McEliece system and in other code-based systems to survive quantum computers.

2 Review of attacks against the McEliece system

This section reviews the McEliece cryptosystem; information-set-decoding attacks against the system; and some claims in the literature regarding the applicability of quantum computers to information-set decoding.

Review of McEliece encryption. Recall that the McEliece public key is a "random" full-rank $k \times n$ matrix G with entries in $\mathbf{F}_2$. Here k and n are system parameters. McEliece originally suggested $k = 524$ and $n = 1024$, aiming for 64-bit security, but these parameters were broken in [5]. Users today take much larger parameters, such as $k = 3556$ and $n = 4096$.

The matrix G specifies a linear map from $\mathbf{F}_2^k$ to $\mathbf{F}_2^n$. The sender encrypts a suitably randomized message $m \in \mathbf{F}_2^k$, together with a uniform random vector e of Hamming weight t, as $Gm + e \in \mathbf{F}_2^n$. Here t is another system parameter. Typically $t = (n-k)/\lceil \lg n \rceil$.

The receiver secretly generates G as a scrambled Goppa matrix, and uses this structure to quickly decode $Gm + e$, obtaining m and e. This structure means that G is not actually a uniform random full-rank matrix. However, all known "structural attacks" that discover the secret, or that merely distinguish G from uniform random, are much slower than the information-set-decoding attacks discussed below.

Review of basic information-set decoding. Basic information-set decoding works as follows. Choose a uniform random size-k subset $S \subseteq \{1, 2, \ldots, n\}$, and consider the natural projection $\mathbf{F}_2^n \to \mathbf{F}_2^S$ that extracts the coordinates indexed by S, discarding all other coordinates. Assume that the following two events occur simultaneously:

- The error vector e projects to $0 \in \mathbf{F}_2^S$; i.e., the entries of e indexed by S are all 0.
- The composition $\mathbf{F}_2^k \xrightarrow{G} \mathbf{F}_2^n \to \mathbf{F}_2^S$ is invertible; i.e., the columns of G indexed by S form an invertible $k \times k$ matrix.

Obtain m by applying the inverse to the projection of $Gm + e$, and obtain e by subtracting Gm from $Gm + e$. If this fails—i.e., if the composition is not invertible, or if the resulting e does not have Hamming weight t—then go back to the beginning and try another set S.

The first event occurs for exactly $\binom{n-t}{k}$ out of the $\binom{n}{k}$ choices of S, so it occurs on average after $\binom{n}{k}/\binom{n-t}{k}$ iterations. If $k = Rn$ and $t \approx (1-R)n/\lg n$ then

$$\frac{\binom{n}{k}}{\binom{n-t}{k}} = \frac{n \cdots (n-t+1)}{(n-k) \cdots (n-k-t+1)} \approx \left(\frac{n}{n-k}\right)^t = \left(\frac{1}{1-R}\right)^t \approx c^{n/\lg n}$$

where $c = 1/(1-R)^{1-R}$. These approximations are quite crude, but a more careful analysis shows that $\binom{n}{k}/\binom{n-t}{k} \in c^{(1+o(1))n/\lg n}$ when t is sufficiently close to $(1-R)n/\lg n$.

For any particular S, the second event occurs for approximately 29% of all matrices G, since approximately 29% of all $k \times k$ matrices over $\mathbf{F}_2$ are invertible. It is tempting to leap to the conclusion that, for any particular G, the second event occurs for approximately 29% of all choices of S, and that the combination of events occurs for approximately $0.29\binom{n-t}{k}$ out of the $\binom{n}{k}$ choices of S. This conclusion is wrong for some choices of G but does appear to be correct for McEliece public keys. For further discussion of this point see [6, Section 2, under "Model of the number of iterations"].

Review of advanced information-set decoding. Advanced forms of information-set decoding complicate each iteration but decrease the number of iterations required, for example by allowing e to have a few bits set within S and combinatorially searching for those bits. There are also many techniques to reduce the cost of finding appropriate sets S, computing inverses, checking whether m is correct, etc. See generally [5].

The analysis in [6] shows that these improvements reduce the cost of information-set decoding by more than a constant power of n but that the final cost is still $c^{(1+o(1))n/\lg n}$.

Review of previous analyses of quantum attacks. I am aware of two previous attempts to quantify the impact of quantum computers upon information-set decoding. The first is by Barg and Zhou in [2, Section 1]. The second is by Overbeck and Sendrier in [20, Section 3.5], as mentioned above.

The Barg–Zhou analysis is a brief sentence claiming that Grover's algorithm can decode any length-n code C, linear or not, "on a quantum computer of circuit size $O(n|C|^{1/2})$ in time $O(n|C|^{1/2})$, which is essentially optimal following a result in [1997 Bennett et al.]." Note that if C has rate R then $|C| = 2^{Rn}$ and $|C|^{1/2} = 2^{Rn/2}$.

There are many reasons to question the Barg–Zhou claim. Specifying an arbitrary non-linear code of length n and rate R requires almost $2^{Rn}n$ bits of information; there is no theoretical obstacle to this information being packed into only $O(2^{Rn/2}n)$ qubits, but Barg and Zhou give no explanation of how to extract the information again from those qubits, never mind the question of what this has to do with Grover's algorithm.

There is no difficulty in building a small computer to enumerate a large *linear* code. In this case a naive application of Grover's algorithm would take essentially $2^{Rn/2}$ iterations but would require only a polynomial-size quantum computer, far below the "optimal" circuit size claimed by Barg and Zhou. Furthermore, information-set decoding takes time only about $c^{n/\lg n}$ on a small non-quantum computer, asymptotically far below the "optimal" time $2^{Rn/2}$ claimed by Barg and Zhou.

The Overbeck–Sendrier analysis is more detailed and is aimed at giving an "intuition why Grover's algorithm is not able [to] give a significant speed-up for the existing attacks." Overbeck and Sendrier begin with "the simplifying assumption that by Grover's algorithm we are able to search a set of size N in $O(\sqrt{N})$ operations on a quantum computer with at least $\log_2(N)$ QuBits." They say that the combinatorial search in advanced forms of information-set decoding (e.g., the collision search introduced by Stern in [23]) "achieves the same speed-up as Grover's algorithm would achieve."

Overbeck and Sendrier also briefly consider, but dismiss, the idea of using Grover's algorithm for "the guessing phase," i.e., to search for sets S having both of the desired properties. They say that "this would either require an iterative application of Grover's algorithm (which is not possible) or a memory of size of the whole search space, as the search function in the second step depends on the first step. This would clearly ruin the 'divide-and-conquer' strategy and is thus not possible either."

This paper shows the opposite: Grover's algorithm can in fact be used to drastically speed up the search for sets S. One might speculate that previous authors were misled by Grover's often-repeated description of his algorithm as searching a "database." See the next section for further discussion of what Grover's algorithm actually does.

3 Quantum information-set decoding

Grover's algorithm is properly understood not as searching through a "database" but as searching for roots of a function. It is easy, from this perspective, to see how to apply Grover's algorithm to information-set decoding. This section spells out the details.

Grover's algorithm. Grover's algorithm is actually a generic constructive transformation from conventional circuits into quantum root-finding circuits. The input to the transformation is a circuit that computes a function $f : \mathbf{F}_2^b \to \mathbf{F}_2$. The output is a quantum circuit that computes a root of f (if one exists): a b-bit string x such that $f(x) = 0$.

In this paper I will be satisfied with a limited class of b-bit-to-1-bit circuits, namely "combinatorial" circuits: i.e., directed acyclic graphs where each node has two incoming edges and computes the NAND of its predecessor nodes. There is a loss of space efficiency from unrolling a long computation into a combinatorial circuit, but the specific functions used in this paper do not take very long to compute, at least compared to the speedups discussed in this paper.

To build a quantum circuit for f one must first build a "reversible" circuit for f: a non-erasing circuit built from Toffoli gates $(x, y, z) \mapsto (x, y, x + yz)$ rather than NANDs. This costs small constant factors in the number of input bits and in the size of the circuit. Replacing the bits by qubits, and replacing the Toffoli gates by quantum Toffoli gates, then produces a circuit of essentially the same size, and essentially the same speed, that computes f on a quantum *superposition* of inputs.

Grover assumes for simplicity that f has a unique root; combines the quantum circuit for f with a quantum rotation and a Hadamard transformation; and iterates the resulting quantum circuit approximately $\sqrt{2^b}$ times, obtaining the root of f with high probability. The rotation and the Hadamard transformation take negligible time and space compared to typical functions f.

Boyer, Brassard, Høyer, and Tapp in [7] presented a generalization of Grover's algorithm using $\sqrt{2^b/r}$ iterations to handle a function f having r roots. The number r need not be known in advance. The generalization in [7] is not actually necessary: one can simply apply Grover's algorithm to random restrictions of f having 1 input, 2 inputs, 4 inputs, etc. With either approach, the number of iterations used by quantum search is only about the square root of the number of iterations used by a traditional brute-force search.

Basic quantum information-set decoding. Fix $y \in \mathbf{F}_2^n$, and fix a $k \times n$ matrix G. Consider the function that, given a size-k subset $S \subseteq \{1, 2, \ldots, n\}$,

- inverts the composition $\mathbf{F}_2^k \xrightarrow{G} \mathbf{F}_2^n \to \mathbf{F}_2^S$, giving up if the composition is not invertible;
- applies the inverse to y, obtaining a vector $m \in \mathbf{F}_2^k$;
- computes $Gm \in \mathbf{F}_2^n$;
- gives up if $Gm - y$ does not have Hamming weight t; and, finally,
- returns 0.

This function can easily be computed by a combinatorial circuit consisting of $O(n^3)$ bit operations.

Recall that basic information-set decoding searches randomly for a root of this function. The search uses approximately $\binom{n}{k}/0.29\binom{n-t}{k} \approx c^{n/\lg n}$ function evaluations on average.

This paper's basic quantum information-set-decoding algorithm finds a root of the same function by Grover's algorithm. Grover's algorithm uses only about $\sqrt{\binom{n}{k}/0.29\binom{n-t}{k}} \approx c^{(1/2)n/\lg n}$ iterations. Each iteration is a quantum function evaluation performing $O(n^3)$ qubit operations; each iteration thus takes time $n^{O(1)}$ on a quantum computer of size $n^{O(1)}$. The total time to find S is $c^{(1/2+o(1))n/\lg n}$ on a quantum computer of size $n^{O(1)}$. Having found S one can compute m and e with negligible extra effort.

Consider again the example $(n, k, t) = (4096, 3556, 45)$ from [20]. Basic quantum information-set decoding performs only about 2^{68} evaluations of this function. Each function evaluation takes a few billion bit operations, so the total cost is approximately 2^{100} qubit operations. Evidently these parameters are far below a safe 128-bit security level, contrary to the analysis in [20, Section 3.5].

Advanced quantum information-set decoding. Recall that more advanced forms of information-set decoding evaluate more complicated functions that have more roots S. One can—and, to properly optimize parameters, should—consider analogous forms of quantum information-set decoding.

Beware that optimization of quantum information-set decoding is not the same as optimization of non-quantum information-set decoding. A $100\times$ increase in the cost of function evaluation has a $100\times$ impact in both settings, but a $100\times$ increase in the number of roots has only a $10\times$ impact in the quantum setting.

For example, the improvement introduced by Lee and Brickell in [17] increases the number of roots by a factor $n^{2+o(1)}$, while increasing the cost of each iteration by only a constant factor. See [6, Section 3] for a detailed analysis of these factors. This improvement therefore saves a factor $n^{1+o(1)}$ in quantum information-set decoding.

References

[1] — (no editor), *Proceedings of the twenty-eighth annual ACM symposium on the theory of computing, held in Philadelphia, PA, May 22–24, 1996*, Association for Computing Machinery, 1996. ISBN 0-89791-785-5. MR 97g:68005. See [13].

[2] Alexander Barg, Shiyu Zhou, *A quantum decoding algorithm of the simplex code*, in Proceedings of the 36th Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, 23–25 September 1998. URL: http://www.enee.umd.edu/~abarg/reprints/rm1dq.pdf. Citations in this document: §2.

[3] Daniel J. Bernstein, *Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete?*, in Workshop Record of SHARCS'09: Special-purpose

Hardware for Attacking Cryptographic Systems (2009). URL: http://cr.yp.to/papers.html#collisioncost. Citations in this document: §1.

[4] Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen (editors), *Post-quantum cryptography*, Springer, 2009. ISBN 978–3–540–88701–0. See [16], [20].

[5] Daniel J. Bernstein, Tanja Lange, Christiane Peters, *Attacking and defending the McEliece cryptosystem*, in [9] (2008), 31–46. URL: http://eprint.iacr.org/2008/318. Citations in this document: §2, §2.

[6] Daniel J. Bernstein, Tanja Lange, Christiane Peters, Henk van Tilborg, *Explicit bounds for generic decoding algorithms for code-based cryptography*, in WCC 2009 (2009). Citations in this document: §1, §2, §2, §3.

[7] Michel Boyer, Gilles Brassard, Peter Høyer, Alain Tapp, *Tight bounds on quantum searching* (1996). URL: http://arxiv.org/abs/quant-ph/9605034v1. Citations in this document: §3, §3.

[8] Gilles Brassard, Peter Høyer, Alain Tapp, *Quantum cryptanalysis of hash and claw-free functions*, in [18] (1998), 163–169. MR 99g:94013. Citations in this document: §1.

[9] Johannes Buchmann, Jintai Ding (editors), *Post-quantum cryptography, second international workshop, PQCrypto 2008, Cincinnati, OH, USA, October 1719, 2008, proceedings*, Lecture Notes in Computer Science, 5299, Springer, 2008. See [5].

[10] Gérard D. Cohen, Jacques Wolfmann (editors), *Coding theory and applications*, Lecture Notes in Computer Science, 388, Springer, 1989. See [23].

[11] Edgar N. Gilbert, F. Jessie MacWilliams, Neil J. A. Sloane, *Codes which detect deception*, Bell System Technical Journal **53** (1974), 405–424. ISSN 0005–8580. MR 55:5306. URL: http://cr.yp.to/bib/entries.html#1974/gilbert. Citations in this document: §1.

[12] Shafi Goldwasser (editor), *35th annual IEEE symposium on the foundations of computer science. Proceedings of the IEEE symposium held in Santa Fe, NM, November 20–22, 1994*, IEEE, 1994. ISBN 0-8186-6580-7. MR 98h:68008. See [21].

[13] Lov K. Grover, *A fast quantum mechanical algorithm for database search*, in [1] (1996), 212–219. MR 1427516. Citations in this document: §1.

[14] Lov K. Grover, *Quantum mechanics helps in searching for a needle in a haystack*, Physical Review Letters **79** (1997), 325–328. Citations in this document: §1.

[15] Christoph G. Günther, *Advances in cryptology: EUROCRYPT '88*, Lecture Notes in Computer Science, 330, Springer-Verlag, Berlin, 1988. ISBN 3–540–50251–3. MR 90a:94002. See [17].

[16] Sean Hallgren, Ulrich Vollmer, *Quantum computing*, in [4] (2009), 15–34. Citations in this document: §1.

[17] Pil Joong Lee, Ernest F. Brickell, *An observation on the security of McEliece's public-key cryptosystem*, in [15] (1988), 275–280. Citations in this document: §3.

[18] Claudio L. Lucchesi, Arnaldo V. Moura (editors), *LATIN'98: theoretical informatics. Proceedings of the 3rd Latin American symposium held in Campinas, April 20–24, 1998*, Lecture Notes in Computer Science, 1380, Springer, 1998. ISBN ISBN 3-540-64275-7. MR 99d:68007. See [8].

[19] Robert J. McEliece, *A public-key cryptosystem based on algebraic coding theory*, JPL DSN Progress Report (1978), 114–116. URL: http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF. Citations in this document: §1.

[20] Raphael Overbeck, Nicolas Sendrier, *Code-based cryptography*, in [4] (2009), 95–145. Citations in this document: §1, §1, §2, §3, §3.

[21] Peter W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring.*, in [12] (1994), 124–134; see also newer version [22]. MR 1489242. Citations in this document: §1.

[22] Peter W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing **26** (1997), 1484–1509; see also older version [21]. MR MR 98i:11108.

[23] Jacques Stern, *A method for finding codewords of small weight*, in [10] (1989), 106–113. Citations in this document: §2.

[24] Christof Zalka, *Fast versions of Shor's quantum factoring algorithm* (1998). URL: `http://arxiv.org/abs/quant-ph/9806084`. Citations in this document: §1.