Ball-collision decoding

Daniel J. Bernstein[1], Tanja Lange[2], and Christiane Peters[2] ⋆

[1] Department of Computer Science
University of Illinois at Chicago, Chicago, IL 60607–7045, USA
djb@cr.yp.to
[2] Department of Mathematics and Computer Science
Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, Netherlands
tanja@hyperelliptic.org, c.p.peters@tue.nl

Abstract. This paper introduces a new generic decoding algorithm that is asymptotically faster than any previous attack against the McEliece cryptosystem. At a 256-bit security level, the attack costs 2.6 times fewer bit operations than the best previous attack; at a theoretical 1000-bit security level, the attack costs 15.5 times fewer bit operations than the best previous attack. The algorithm is asymptotically even faster than the Finiasz–Sendrier "lower bound" published at Asiacrypt 2009, demonstrating that the Finiasz–Sendrier parameter recommendations are not as secure as claimed. This paper proposes much safer, but still reasonably efficient, parameters based on an analysis of the fundamental bottleneck in all algorithms of this type.

Keywords: information-set decoding, collision decoding, ball-collision decoding, attacks, McEliece cryptosystem, Niederreiter cryptosystem

1 Introduction

In 1978 McEliece introduced a fast code-based public-key cryptosystem that has maintained remarkable strength against every proposed attack. Straightforward algorithms for encryption and decryption take time $b^{2+o(1)}$ when parameters are chosen to provide b-bit security against the best attack known today — or when parameters are chosen to provide b-bit security against the best *quantum* attack known today.

For comparison, low-exponent RSA encryption takes time $b^{3+o(1)}$ when RSA moduli are chosen to provide b-bit security against the best attack known today.[3] Even more time is required for RSA *decryption*, Diffie–Hellman key exchange (in its original form using multiplicative groups), etc. Furthermore, the introduction of Shor's algorithm in 1994 [65] showed that RSA would need encryption time at least $2^{(1/2+o(1))b}$ to provide b-bit security against quantum attacks. Elliptic-curve cryptography offers $b^{2+o(1)}$ encryption time and decryption time but, like RSA, will not survive quantum computers.

Attack optimization and parameter selection. Generic decoding algorithms such as "information-set decoding" have always been the top threat against the McEliece cryptosystem. There have been dozens of papers analyzing and improving these algorithms. The relevant ideas are explained and cited later in this paper.

⋆ This work was supported by the Cisco University Research Program and by the European Commission under Contract ICT-2007-216646 ECRYPT II. Permanent ID of this document: 0e8c929565e20cf63e6a19794e570bb1. Date: 2010.11.17.

[3] Modern algorithms to factor n-bit integers take time $2^{n^{1/3+o(1)}}$, so providing b-bit security requires key size $b^{3+o(1)}$. Low-exponent encryption using an asymptotically fast FFT-based multiplication algorithm takes time essentially linear in the key size, and therefore time $b^{3+o(1)}$. Key size $b^{2+o(1)}$ and time $b^{2+o(1)}$ would have sufficed against Schroeppel's linear sieve, the best factorization algorithm mentioned in the original 1978 RSA paper [63], but the introduction of the number-field sieve in the early 1990s [51] forced asymptotically much larger key sizes.

The following example illustrates the cumulative impact of these speedups. McEliece's original parameter suggestions ("$n = 1024, k = 524, t = 50$") take about $524^3\binom{1024}{50}/\binom{500}{50} \approx 2^{81}$ operations to break by the simple information-set-decoding attack explained in McEliece's original paper [54, Section 3]. (McEliece estimated the attack cost as $524^3(1-50/1024)^{-524} \approx 2^{65}$; this underestimate was corrected by Adams and Meijer in [2, Section 3].) The attack presented by Bernstein, Lange, and Peters in [9], thirty years after McEliece's paper, builds on several improvements and takes only about $2^{60.5}$ operations for the same parameters. That attack was carried out successfully, decrypting a challenge ciphertext.

The same algorithmic improvements have forced almost a $2\times$ increase in the McEliece key size, encryption time, and decryption time at high security levels. This is not a bad security record over three decades, and has no impact on the asymptotic cost $b^{2+o(1)}$ mentioned earlier; but implementors selecting parameters for the McEliece cryptosystem need to know the limits of these algorithms, the same way that implementors selecting RSA key sizes need to know the limits of factorization algorithms.

In an Asiacrypt 2009 paper "Security bounds for the design of code-based cryptosystems" [32], Finiasz and Sendrier presented "lower bounds on the effective work factor of existing real algorithms, but also on the future improvements that could be implemented." For example, they computed $2^{59.9}$ as a bound for McEliece's original parameters, and $2^{128.5}$ as a bound for the larger parameters $(4096, 3604, 41)$. They said that beating these bounds would require the introduction of "new techniques, never applied to code-based cryptosystems", and concluded by suggesting these bounds as a tool to select safe parameters.

Note that the bound $2^{59.9}$ is quite close to the $2^{60.5}$ mentioned above. It seemed in retrospect that thirty years of refinements had been gradually converging towards the lower bound identified in [32].

Contents of this paper. This paper introduces a new decoding algorithm that asymptotically beats the Asiacrypt 2009 lower bound. We call this algorithm "ball-collision decoding" because of a geometric interpretation explained in Section 4.

Previous algorithms can be seen as special cases of ball-collision decoding; we prove that those cases are never asymptotically optimal. The asymptotic speedup is easiest to state for constant code rate k/n and constant error fraction w/n as $n \to \infty$: if T is the time taken by the best previous algorithm then the new algorithm takes time only $T/T^{c+o(1)}$, where c is a positive constant. The asymptotic speedup factor in the McEliece setting, where w/n converges slowly to 0 as $n \to \infty$, is more complicated to state but still grows superpolynomially.

This paper carefully evaluates the exact cost of ball-collision decoding, using the same bit-operation-counting rules as in the previous literature. For the parameters $(6624, 5129, 117)$ proposed in [9, Section 7], the cost of ball-collision decoding is almost 3 times smaller than the cost of the best previous attack.

Of course, actually breaking those parameters remains very far out of reach, and our results should not be interpreted as damaging the viability of the McEliece cryptosystem. However, our results *do* raise new questions regarding the proper choice of parameters for the McEliece cryptosystem.

The Asiacrypt 2009 lower bound was implicitly based on a combination of several bottlenecks, some of which are avoided by ball-collision decoding. We propose much safer parameter choices, based on a single fundamental bottleneck. Selecting parameters according to this proposal loses a small percentage in efficiency compared to merely protecting against known

attacks, but it provides a much higher level of confidence that the parameters will remain secure against future attacks.

Caveat 1: key size. The McEliece key size $b^{2+o(1)}$ is *asymptotically* smaller than the RSA key size $b^{3+o(1)}$, but the RSA key size is smaller for all practical values of b. The ECC key size is even smaller, just $2b$ bits. For example, the smallest McEliece key size proposed in [9, Section 7] for $b = 128$ was 192192 *bytes*, while RSA keys at the same security level are just 3072 bits, and ECC keys are just 256 bits.

The relatively large key size has made the McEliece cryptosystem unsuitable for many applications over the past thirty years. On the other hand, increased network bandwidth and increased storage space are continuing to reduce the impact of large key sizes. A modern 1.5-terabyte hard drive costing $80 can store several million 192192-byte keys, and a server storing millions of keys can use those keys to protect much larger volumes of network traffic. At the low end, Eisenbarth, Güneysu, Heyse, and Paar at CHES 2009 [30] reported successfully implementing the McEliece cryptosystem (at a somewhat lower security level) on an AVR microcontroller and a Spartan FPGA.

There have been many proposals that reduce the McEliece key size by deviating in various ways from McEliece's original selection of random binary Goppa codes as error-correcting codes. Several of these proposals have been broken by "structural attacks" that exploit non-randomness in the public key. For example, [34] and [31] broke many cases of [55]. The attacks we consider in this paper are *generic* attacks that work against any code-based cryptosystem. The maximum security that any designer can hope to achieve is security against the new generic decoding attack explained in this paper.

Caveat 2: chosen-ciphertext attacks. "Attacks" above refer only to passive single-target inversion attacks. The original McEliece cryptosystem, like the original RSA cryptosystem, is really just a trapdoor one-way function; when used naively as a public-key cryptosystem it is trivially broken by chosen-ciphertext attacks such as Berson's attack [11] and the Verheul–Doumen–van Tilborg attack [71].

Protecting the McEliece system against these attacks, to meet the standard notion of IND-CCA2 security for a public-key cryptosystem, requires appropriate padding and randomization, similar to RSA-OAEP. As shown by Kobara and Imai in [48], adding this protection does not significantly increase the cost of the McEliece cryptosystem.

Caveat 3: quantum computers. The exponent of this paper's attack is not as low as the exponent for quantum information-set decoding reported by Bernstein in [7]. We do not claim that our parameter recommendations are suitable for post-quantum cryptography; we also do not claim that our attack optimizations will remain productive in a quantum context. This paper takes the same view as papers on factorization and discrete logarithms, such as [42] and [43]: it focuses on achieving security under the (currently reasonable) assumption that large quantum computers do not exist.

2 Review of the McEliece cryptosystem

The public key in the McEliece cryptosystem consists of a random-looking rank-k matrix $G \in \mathbf{F}_2^{k \times n}$. The sender encrypts a message m in $\mathbf{F}_2^k$ by first multiplying it with the matrix G, producing mG; choosing uniformly at random a word e in $\mathbf{F}_2^n$ of Hamming weight w; and adding e to mG, producing a ciphertext $mG + e$. The cryptosystem parameters are n, k, w.

The legitimate receiver decrypts $mG + e$ using a secret key which consists of a secret decoding algorithm producing the *error vector* e given $mG + e$. The details are not relevant to the attacks described in this paper and can be found in, e.g., [58].

An attacker is faced with the problem of determining e given G and $mG + e$. Note that finding e is equivalent to finding the message m: subtracting e from $mG + e$ produces mG, and then simple linear transformations produce m.

The set $\mathbf{F}_2^k G = \left\{mG : m \in \mathbf{F}_2^k\right\}$ is called a *linear code* of *length* n and *dimension* k, specifically the linear code *generated by* G. The matrix G is called a *generator matrix* for this code. The elements of $\mathbf{F}_2^k G$ are called *codewords*. If the linear code $\mathbf{F}_2^k G$ equals $\{c \in \mathbf{F}_2^n : Hc = 0\}$ then the matrix H is called a *parity-check matrix* for the code.

Without loss of generality one can assume that the matrix G in a CCA2-secure version of the McEliece cryptosystem is given in *systematic form* $G = (I_k|-A^T)$ where I_k is a $k \times k$ identity matrix and A an $(n-k) \times k$ matrix. Then the matrix $H = (A|I_{n-k})$ is a parity-check matrix for the code generated by G.

An *information set* Z for H is a set of k integers in $\{1, 2, \ldots, n\}$ for which the $n-k$ columns of H that are not indexed by Z are linearly independent. Applying Gaussian elimination to those $n - k$ columns shows that codewords are determined by their Z-indexed components. For example, $\{1, 2, \ldots, k\}$ is an information set for $H = (A|I_{n-k})$; codewords are determined by their first k components.

Let $c = mG$ for $m \in \mathbf{F}_2^k$ and $e \in \mathbf{F}_2^n$ with $\mathrm{wt}(e) = w$. Then by linearity one has $H(c+e) = Hc + He = He$ since $Hc = 0$. The result $s = He$ is called the *syndrome*. It is the sum of the w columns of H that are indexed by the positions of 1's in e. The attacker's task is equivalent to finding e given H and $s = He$.

3 The ball-collision-decoding algorithm

This section introduces ball-collision decoding. It first presents a simplified statement of the algorithm and then discusses various optimizations. Section 4 explains how this algorithm relates to previous algorithms.

The algorithm is given a parity-check matrix $H \in \mathbf{F}_2^{(n-k)\times n}$, a syndrome $s \in \mathbf{F}_2^{n-k}$, and a weight $w \in \{0, 1, 2, \ldots\}$. The goal of the algorithm is to find a corresponding error vector e: i.e., a vector $e \in \mathbf{F}_2^n$ of weight w such that $s = He$.

Ball-collision decoding has its roots in information-set decoding, which was used against the McEliece system in, e.g., [67], [16], [17], and [9]. The previous algorithms select a random information set in the parity-check matrix and then search for vectors having a particular pattern of non-zero entries. Ball-collision decoding is similar but searches for a more complicated, and more likely, pattern. See Section 4 for further discussion of the previous work.

The reader is encouraged to consider, while reading the algorithm, the case that the algorithm is given a matrix H already in systematic form and that it chooses $Z = \{1, 2, \ldots, k\}$ as information set. The matrix U in Step 4 is then the identity matrix I_{n-k}. The algorithm divides H into blocks, and divides the syndrome s into corresponding blocks, as specified by algorithm parameters ℓ_1, ℓ_2:

$$H = \begin{pmatrix} A_1 & I_1 & 0 \\ A_2 & 0 & I_2 \end{pmatrix}, \qquad s = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix},$$

where $s_1 \in \mathbf{F}_2^{\ell_1+\ell_2}$, $s_2 \in \mathbf{F}_2^{n-k-\ell_1-\ell_2}$, $A_1 \in \mathbf{F}_2^{(\ell_1+\ell_2)\times k}$, $A_2 \in \mathbf{F}_2^{(n-k-\ell_1-\ell_2)\times k}$, and each I_i is an identity matrix.

One iteration of ball-collision decoding:
CONSTANTS: $n, k, w \in \mathbf{Z}$ with $0 \le w \le n$ and $0 \le k \le n$.
PARAMETERS: $p_1, p_2, q_1, q_2, k_1, k_2, \ell_1, \ell_2 \in \mathbf{Z}$ with $0 \le k_1$, $0 \le k_2$, $k = k_1 + k_2$, $0 \le p_1 \le k_1$, $0 \le p_2 \le k_2$, $0 \le q_1 \le \ell_1$, $0 \le q_2 \le \ell_2$, and $0 \le w - p_1 - p_2 - q_1 - q_2 \le n - k - \ell_1 - \ell_2$.
INPUT: $H \in \mathbf{F}_2^{(n-k)\times n}$ and $s \in \mathbf{F}_2^{n-k}$.
OUTPUT: Zero or more vectors $e \in \mathbf{F}_2^n$ with $He = s$ and $\mathrm{wt}(e) = w$.

1. Choose a uniform random information set Z.
2. Choose a uniform random partition of Z into parts of sizes k_1 and k_2. Subsequent steps of the algorithm write "$\mathbf{F}_2^{k_1}$" and "$\mathbf{F}_2^{k_2}$" to refer to the corresponding subspaces of $\mathbf{F}_2^Z$.
3. Choose a uniform random partition of $\{1, 2, \ldots, n\} \setminus Z$ into parts of sizes ℓ_1, ℓ_2, and $n-k-\ell_1-\ell_2$. Subsequent steps of the algorithm write "$\mathbf{F}_2^{\ell_1}$" and "$\mathbf{F}_2^{\ell_2}$" and "$\mathbf{F}_2^{n-k-\ell_1-\ell_2}$" to refer to the corresponding subspaces of $\mathbf{F}_2^{\{1,2,\ldots,n\}\setminus Z}$.
4. Find an invertible $U \in \mathbf{F}_2^{(n-k)\times(n-k)}$ such that the columns of UH indexed by $\{1, 2, \ldots, n\} \setminus Z$ are an $(n-k) \times (n-k)$ identity matrix. Write the columns of UH indexed by Z as $\begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$ with $A_1 \in \mathbf{F}_2^{(\ell_1+\ell_2)\times k}$, $A_2 \in \mathbf{F}_2^{(n-k-\ell_1-\ell_2)\times k}$.
5. Write Us as $\begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$ with $s_1 \in \mathbf{F}_2^{\ell_1+\ell_2}$, $s_2 \in \mathbf{F}_2^{n-k-\ell_1-\ell_2}$.
6. Compute the set S consisting of all triples $(A_1x_0 + x_1, x_0, x_1)$ where $x_0 \in \mathbf{F}_2^{k_1} \times \{0\}^{k_2}$, $\mathrm{wt}(x_0) = p_1$, $x_1 \in \mathbf{F}_2^{\ell_1} \times \{0\}^{\ell_2}$, $\mathrm{wt}(x_1) = q_1$.
7. Compute the set T consisting of all triples $(A_1y_0 + y_1 + s_1, y_0, y_1)$ where $y_0 \in \{0\}^{k_1} \times \mathbf{F}_2^{k_2}$, $\mathrm{wt}(y_0) = p_2$, $y_1 \in \{0\}^{\ell_1} \times \mathbf{F}_2^{\ell_2}$, $\mathrm{wt}(y_1) = q_2$.
8. For each $(v, x_0, x_1) \in S$:
   For each y_0, y_1 such that $(v, y_0, y_1) \in T$:
   If $\mathrm{wt}(A_2(x_0 + y_0) + s_2) = w - p_1 - p_2 - q_1 - q_2$:
   Output the vector $e \in \mathbf{F}_2^n$ whose Z-indexed components are $x_0 + y_0$ and whose remaining components are $(x_1 + y_1 || A_2(x_0 + y_0) + s_2)$.

Note that Step 8 is a standard "join" operation between S and T; it can be implemented efficiently by sorting or by hashing. Bernstein, Lange, and Peters in [9, Section 6] describe an efficient implementation of essentially the same operation using only about $2^{\ell_1+\ell_2+1}$ bits of memory. We do not discuss memory issues further in this paper.

Theorem 3.1 (Correctness of ball-collision decoding) *The set of output vectors e of the ball-collision decoding algorithm is the set of vectors e that satisfy $He = s$, have weights p_1, p_2 in blocks of length k_1, k_2 in the Z-indexed components, and have weights $q_1, q_2, w - p_1 - p_2 - q_1 - q_2$ in blocks of length $\ell_1, \ell_2, n - k - \ell_1 - \ell_2$ in the remaining components.*

Proof. Each element $(v, x_0, x_1) \in S$ satisfies $x_0 \in \mathbf{F}_2^{k_1} \times \{0\}^{k_2}$ with $\mathrm{wt}(x_0) = p_1$; $v = A_1x_0 + x_1$ and $x_1 \in \mathbf{F}_2^{\ell_1} \times \{0\}^{\ell_2}$ with $\mathrm{wt}(x_1) = q_1$. Similarly $y_0 \in \{0\}^{k_1} \times \mathbf{F}_2^{k_2}$ with $\mathrm{wt}(y_0) = p_2$; $v = A_1y_0 + y_1 + s_1$; $y_1 \in \{0\}^{\ell_1} \times \mathbf{F}_2^{\ell_2}$ with $\mathrm{wt}(y_1) = q_2$. Now, with Z-indexed columns visualized as coming before the remaining columns, we have

$$UHe = UH \begin{pmatrix} x_0 + y_0 \\ x_1 + y_1 \\ A_2(x_0 + y_0) + s_2 \end{pmatrix} = \begin{pmatrix} A_1(x_0 + y_0) + x_1 + y_1 \\ A_2(x_0 + y_0) + A_2(x_0 + y_0) + s_2 \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = Us$$

so $He = s$. Furthermore, $x_0 + y_0 \in \mathbf{F}_2^{k_1+k_2}$ has weights p_1, p_2 in blocks of lengths k_1, k_2; $x_1 + y_1 \in \mathbf{F}_2^{\ell_1+\ell_2}$ has weights q_1, q_2 in blocks of lengths ℓ_1, ℓ_2; and $\mathrm{wt}(A_2(x_0 + y_0) + s_2) = w - p_1 - p_2 - q_1 - q_2$.

Conversely, the iteration finds every vector e having this weight distribution and satisfying $He = s$. Indeed, write the Z-indexed columns of e as $x_0 + y_0$ and the remaining columns of e as $(x_1 + y_1||e_2)$ with $x_0 \in \mathbf{F}_2^{k_1} \times \{0\}^{k_2}$, $y_0 \in \{0\}^{k_1} \times \mathbf{F}_2^{k_2}$, $x_1 \in \mathbf{F}_2^{\ell_1} \times \{0\}^{\ell_2}$, $y_1 \in \{0\}^{\ell_1} \times \mathbf{F}_2^{\ell_2}$, and $e_2 \in \mathbf{F}_2^{n-k-\ell_1-\ell_2}$. By hypothesis the weights of x_0, y_0, x_1, y_1, e_2 are $p_1, p_2, q_1, q_2, w - p_1 - p_2 - q_1 - q_2$, respectively. Now define $v = A_1x_0 + x_1$. The equation $UHe = Us$ implies $v = A_1y_0 + y_1 + s_1$; and $e_2 = A_2(x_0 + y_0) + s_2$. Hence $(v, x_0, x_1) \in S$ and $(v, y_0, y_1) \in T$. Finally $\mathrm{wt}(A_2(x_0 + y_0) + s_2) = \mathrm{wt}(e_2) = w - p_1 - p_2 - q_1 - q_2$ so the algorithm prints e as claimed. □

Finding an information set. The simplest way to choose a uniform random information set is to repeatedly choose a uniform random size-k subset $Z \subseteq \{1, 2, \ldots, n\}$ until the $n - k$ columns of H indexed by $\{1, 2, \ldots, n\} \setminus Z$ are linearly independent. Standard practice (see, e.g., Stern [67]) is to eliminate the fruitless Gaussian-elimination steps here, at the expense of negligible bias, by assembling the information set one column at a time, ensuring that each newly added column is linearly independent of the previously selected columns. After this optimization there is only one Gaussian-elimination step per iteration.

Reusing intermediate sums. Computing the vector A_1x_0 for a weight-p_1 word x_0 in $\mathbf{F}_2^{k_1} \times \{0\}^{k_2}$ can be done by adding the specified p_1 columns of A_1 in $p_1 - 1$ additions in $\mathbf{F}_2^{\ell_1+\ell_2}$.

Computing A_1x_0 for *all* the $\binom{k_1}{p_1}$ vectors x_0 can be done more efficiently than repeating this process for each of them. Start by computing all $\binom{k_1}{2}$ sums of 2 columns of A_1; each sum costs one addition in $\mathbf{F}_2^{\ell_1+\ell_2}$. Then compute all $\binom{k_1}{3}$ sums of 3 columns of A_1 by adding one extra column to the previous results. Proceed in the same way until all $\binom{k_1}{p_1}$ sums of p_1 columns of A_1 are computed. This produces all required sums in only marginally more than one $\mathbf{F}_2^{\ell_1+\ell_2}$ addition per sum; see Section 5 for a precise operation count.

Early abort. The vector $A_2(x_0 + y_0) + s_2$ is computed as a sum of $p_1 + p_2 + 1$ vectors of length $n - k - \ell_1 - \ell_2$. Instead of computing the sum on all $n - k - \ell_1 - \ell_2$ positions one computes the sum row by row and simultaneously checks the weight. If the weight exceeds $w - p_1 - p_2 - q_1 - q_2$ one can discard this particular pair (x_0, y_0).

We comment that one can further reduce the cost of this step by precomputing sums of smaller sets of columns, but we do not use this idea in our analysis, because it is not critical for the algorithm's performance.

4 Relationship to previous algorithms

This section discusses the relationship of ball-collision decoding to previous information-set-decoding algorithms.

Collision decoding vs. ball-collision decoding. We use the name "collision decoding" for the special case $q_1 = q_2 = 0$ of ball-collision decoding. The idea of collision decoding is more than twenty years old: Stern's algorithm in [67] is, aside from trivial details, exactly the special case $q_1 = q_2 = 0$, $p_1 = p_2$, $k_1 \approx k_2$. Dumer in [27] independently introduced the core idea, although in a more limited form, and in [28] achieved an algorithm similar to Stern's.

All state-of-the-art decoding attacks since [67] have been increasingly optimized forms of collision decoding. Other approaches to decoding, such as "gradient decoding" ([4]), "supercode decoding" ([5]), and "statistical decoding" (see [3] and [57]), have never been competitive with Stern's algorithm. This does not mean that those approaches should be ignored; our generalization from collision decoding to ball-collision decoding is inspired by one of the steps in supercode decoding.

Collision decoding searches for collisions in $\mathbf{F}_2^{\ell_1+\ell_2}$ between points A_1x_0 and points $A_1y_0+s_1$. Ball-collision decoding expands each point A_1x_0 into a small ball (in the Hamming metric), namely $\{A_1x_0+x_1 : x_1 \in \mathbf{F}_2^{\ell_1} \times \{0\}^{\ell_2}, \mathrm{wt}(x_1) = q_1\}$; similarly expands each point A_1y_0 into a small ball; and searches for collisions between these balls.

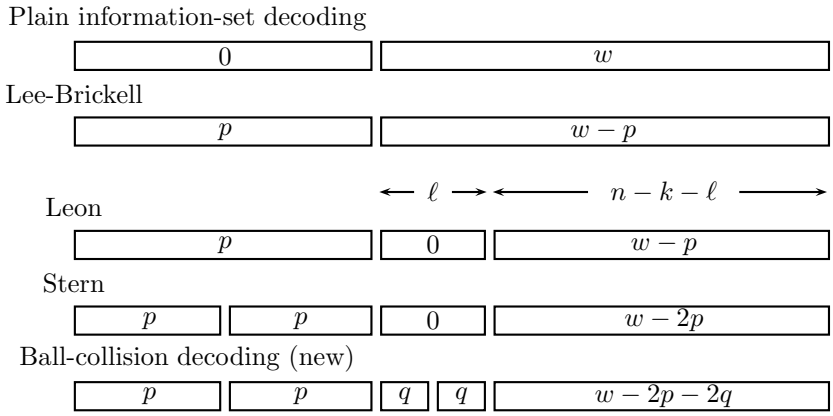
From the perspective of ball-collision decoding, the fundamental disadvantage of collision decoding is that errors are required to avoid an asymptotically quite large stretch of $\ell_1+\ell_2$ positions. Ball-collision decoding makes a much more reasonable hypothesis, namely that there are asymptotically increasingly many errors in those positions. It requires extra work to enumerate the points in each ball, but the extra work is only about the square root of the improvement in success probability. The cost ratio is asymptotically superpolynomial; see Section 7.

Collision decoding also has a more superficial disadvantage compared to ball-collision decoding: its inner loop is slower, since computing A_1x_0 for a new x_0 is considerably more expensive than adding x_1 for a new x_1. The cost ratio here is only polynomial, and is not relevant to the asymptotic analysis (see Section 7), but is accounted for in the bit-operation count (see Section 5).

Additional credits. The simplest form of information-set decoding, introduced by Prange in [61], did not allow errors in the information set. For asymptotic analyses see [54], [1], and [2].

The idea of allowing errors was published by Lee and Brickell in [50], by Leon in [52], and by Krouk in [49], but without Stern's collision idea; in the terminology of ball-collision decoding, with $p_2 = 0$, $q_1 = q_2 = 0$, and $\ell_2 = 0$. For each pattern of p_1 errors in k columns, Lee and Brickell checked the weight of the remaining $n-k$ columns; Leon and Krouk required ℓ_1 columns to have weight 0, and usually checked only those columns. For asymptotic analyses see [49], [24], and [25].

Overbeck and Sendrier [58] give a visual comparison of the algorithms by comparing to which interval they restrict how many errors. The following picture extends their picture to include ball-collision decoding. It shows that the new algorithm allows errors in an interval that had to be error-free in Leon's and Stern's algorithms.



One way to speed up Gaussian elimination is to change only one information-set element in each iteration. This idea was introduced by Omura, according to [22, Section 3.2.4]. It was applied to increasingly optimized forms of information-set decoding by van Tilburg in [68] and [69], by Chabanne and Courteau in [19], by Chabaud in [20], by Canteaut and Chabanne in [15], by Canteaut and Chabaud in [16], and by Canteaut and Sendrier in [17]. Bernstein, Lange, and Peters in [9] improved the balance between Gaussian-elimination cost and error-searching cost by changing c information-set elements in each iteration for an optimized value of c.

The ideas of reusing sums and aborting weight calculations also appeared in [9], in the context of an improved collision-decoding algorithm; we generalize to ball-collision decoding. The most recent improvements are an asymptotic $\Theta(p^{1/4})$ "birthday" speedup achieved by Finiasz and Sendrier in [32] by dropping Stern's left-right separation, and an optimized generalization to $\mathbf{F}_q$ by Peters in [59].

5 Complexity analysis

This section analyzes the complexity of ball-collision decoding. In particular, this section analyzes the success probability of each iteration and the number of bit operations needed for each iteration.

Success probability. Assume that e is a uniform random vector of weight w. One iteration of ball-collision decoding finds e exactly if it has the right weight distribution, namely weight p_1 in the first k_1 positions specified by the information set, weight p_2 in the remaining k_2 positions specified by the information set, weight q_1 on the first ℓ_1 positions outside the information set, and weight q_2 on the next ℓ_2 positions outside the information set.

The probability that e has this weight distribution is, by a simple counting argument, exactly

$$b(p_1,p_2,q_1,q_2,\ell_1,\ell_2) = \binom{n}{w}^{-1}\binom{n-k-\ell_1-\ell_2}{w-p_1-p_2-q_1-q_2}\binom{k_1}{p_1}\binom{k_2}{p_2}\binom{\ell_1}{q_1}\binom{\ell_2}{q_2}.$$

The expected number of iterations of the outer loop is, for almost all H, very close to the reciprocal of the success probability of a single iteration. We explicitly disregard, without further comment, the extremely unusual codes for which the average number of iterations is significantly different from the reciprocal of the success probability of a single iteration. For further discussion of this issue and how unusual it is see, e.g., [25] and [10].

Gaussian elimination. There are several ways to speed up Gaussian elimination, as discussed in Section 4, and implementors are encouraged to use those optimizations. However, in this paper we will be satisfied with a quite naive form of Gaussian elimination, taking $(1/2)(n-k)^2(n+k)$ bit operations; our interest is in large input sizes, and Gaussian elimination takes negligible time for those sizes.

Building the set S. Using intermediate sums, the total cost amounts to

$$(\ell_1+\ell_2)\left(\binom{k_1}{2}+\binom{k_1}{3}+\cdots\binom{k_1}{p_1}\right).$$

Using $L(k,p)=\sum_{i=1}^{p}\binom{k}{i}$ as a shorthand, the costs can be written as $(\ell_1+\ell_2)\,(L(k_1,p_1)-k_1)$.

Then, for each A_1x_0 all $\binom{\ell_1}{q_1}$ possible words x_1 in $\mathbf{F}_2^{\ell_1}\times 0^{\ell_2}$ of weight at most q_1 are added to compute vectors $A_1x_0+x_1$. Again intermediate sums can be used, so this step takes $\min\{1,q_1\}\binom{k_1}{p_1}L(\ell_1,q_1)$ bit operations; note that for $q_1=0$ the cost of this step is indeed 0.

Each choice of (x_0,x_1) adds one element to S. Hence, the number of elements in S equals exactly the number of choices for x_0 and x_1, i.e. $\#S=\binom{k_1}{p_1}\binom{\ell_1}{q_1}$.

Building the set T. The set T is built similarly to the set S. The only difference is that the expression $A_1y_0+y_1+s_1$ involves adding s_1 and thus the single columns (corresponding to weight-1 words y_0) already cost $(\ell_1+\ell_2)\binom{k_2}{1}$ bit operations. In total this step takes $(\ell_1+\ell_2)L(k_2,p_2)+\min\{1,q_2\}\binom{k_2}{p_2}L(\ell_2,q_2)$.

The set T contains exactly $\#T=\binom{k_2}{p_2}\binom{\ell_2}{q_2}$ elements.

Checking collisions. The last step does one check for every (x_0,x_1,y_0,y_1) satisfying the equation $A_1x_0+x_1=A_1y_0+y_1+s_1$. There are $\binom{k_1}{p_1}\binom{k_2}{p_2}\binom{\ell_1}{q_1}\binom{\ell_2}{q_2}$ choices of (x_0,x_1,y_0,y_1).

If the vectors v appearing in S and T were uniformly distributed among the $2^{\ell_1+\ell_2}$ possible values then on average $\#S\cdot\#T\cdot 2^{-\ell_1-\ell_2}$ checks would be done. The expected number of checks is extremely close to this for almost all H; as above we disregard the extremely unusual codes with different behavior.

Each check consists of computing $\mathrm{wt}(A_2(x_0+y_0)+s_2)$ and testing whether it equals $w-p_1-p_2-q_1-q_2$. When using the early-abort weight calculation, on average only $2(w-p_1-p_2-q_1-q_2+1)$ bits of the result are computed before the weight is found too high. Each bit of the result costs p_1+p_2 bit operations because x_0+y_0 has weight p_1+p_2.

Cost of one iteration. To summarize, the total cost per iteration of the inner loop with parameters $p_1,p_2,q_1,q_2,\ell_1,\ell_2$ amounts to

$$\begin{aligned}c(p_1,p_2,q_1,q_2,\ell_1,\ell_2)=&\frac{1}{2}(n-k)^2(n+k)+(\ell_1+\ell_2)\big(L(k_1,p_1)+L(k_2,p_2)-k_1\big)\\&+\min\{1,q_1\}\binom{k_1}{p_1}L(\ell_1,q_1)+\min\{1,q_2\}\binom{k_2}{p_2}L(\ell_2,q_2)\\&+2(w-p_1-p_2-q_1-q_2+1)(p_1+p_2)\binom{k_1}{p_1}\binom{k_2}{p_2}\binom{\ell_1}{q_1}\binom{\ell_2}{q_2}2^{-\ell_1-\ell_2}.\end{aligned}$$

6 Concrete parameter examples

This section considers concrete examples in order to show the speedup gained by ball-collision decoding in comparison to collision decoding. The first parameters were previously proposed to

achieve 256-bit security against current attacks. We designed the second parameters according to similar rules to achieve a 1000-bit security level against current attacks. We do not mean to suggest that 1000-bit security is of any real-world relevance; we consider it to demonstrate the asymptotic superiority of ball-collision decoding.

For each set of parameters we consider the following costs:

(1) the cost of collision decoding ($q_1 = q_2 = 0$),
(2) the cost of collision decoding using the birthday trick from [32] as analyzed in [59],
(3) the lower bound given by Finiasz and Sendrier in [32], and
(4) the cost of ball-collision decoding.

Note that (1), (2), and (4) are actual algorithm costs whereas (3) is merely a lower bound.

256-security revisited. According to [9, Section 7] a binary code with length $n = 6624$, $k = 5129$, $w = 117$ achieves 256-bit security. The best collision-decoding parameters are actually slightly below 2^{256} bit operations: they use $2^{181.4928}$ iterations (on average), each taking $2^{74.3741}$ bit operations, for a total of $2^{255.8669}$ bit operations.

Collision decoding with the birthday trick takes, with optimal parameters, $2^{255.54880}$ bit operations. The birthday trick increases the cost per iteration by a factor of 2.2420 compared to the classical collision-decoding algorithm, to $2^{75.5390}$ bit operations. However, the trick increases the chances of finding the desired error vector noticeably, reducing the number of iterations by a factor of 2.7951, to $2^{180.0099}$. Thus the birthday trick yields an overall 1.2467039× speedup.

The Finiasz–Sendrier lower bound is $2^{255.1787}$ bit operations, 1.6112985× smaller than the cost of collision decoding.

Ball-collision decoding with parameters $k_1 = 2565$, $k_2 = 2564$, $\ell_1 = \ell_2 = 47$, $p_1 = p_2 = 8$, and $q_1 = q_2 = 1$ needs only $2^{254.1519}$ bit operations to attack the same system. On average the algorithm needs $2^{170.6473}$ iterations each taking $2^{83.504570}$ bit operations.

Ball-collision decoding thus costs 3.2830× less than collision decoding, 2.6334× less than collision decoding with the birthday trick, and 2.0375× less than the Finiasz–Sendrier lower bound.

1000-bit security. Attacking a system based on a code of length $n = 30332$, $k = 22968$, $w = 494$ requires $2^{1000.9577}$ bit operations using collision decoding with the optimal parameters $k_1 = k_2 = 11484$, $\ell_1 = \ell_2 = 140$, $p_1 = p_2 = 27$ and $q_1 = q_2 = 0$.

The birthday trick reduces the cost by a factor of 1.7242831, to $2^{1000.1717}$ bit operations. This means that this system offers 1000-bit security against all previously known attacks.

The Finiasz–Sendrier lower bound is $2^{999.45027}$ bit operations, 2.8430× smaller than the cost of collision decoding and 1.6488× smaller than the cost of collision decoding with the birthday trick.

Ball-collision decoding with parameters $k_1 = k_2 = 11484$, $\ell_1 = \ell_2 = 156$, $p_1 = p_2 = 29$, and $q_1 = q_2 = 1$ needs only $2^{996.21534}$ bit operations. This is 26.767× smaller than the cost of collision decoding, 15.523× smaller than the cost of collision decoding with the birthday trick, and 9.415× smaller than the Finiasz–Sendrier lower bound.

7 Asymptotic complexity of ball-collision decoding

This section analyzes the asymptotic behavior of the cost of ball-collision decoding, and shows that it always has a smaller asymptotic exponent than the cost of collision decoding.

For comparison, Finiasz and Sendrier say in [32, Section 3.3] that their birthday trick and their lower bound gain only $\Theta(p^{1/4}) \leq \Theta(n^{1/4})$ compared to collision decoding. Any polynomial factor in n makes *no* change in the asymptotic cost exponent, so the speedup from ball-collision decoding is asymptotically much larger than the speedup from the birthday trick.

Input sizes. Fix a real number W with $0 < W < 1/2$, and fix a real number R with $-W \log_2 W - (1-W) \log_2 (1-W) \leq 1 - R < 1$.

Consider codes and error vectors of very large length n, where the codes have dimension $k \approx Rn$, and the error vectors have weight $w \approx Wn$. More precisely, fix functions $k, w : \{1, 2, \ldots\} \to \{1, 2, \ldots\}$ that satisfy $\lim_{n\to\infty} k(n)/n = R$ and $\lim_{n\to\infty} w(n)/n = W$; more concisely, $k/n \to R$ and $w/n \to W$.

Attack parameters. Fix real numbers P, Q, L with $0 \leq P \leq R/2$, $0 \leq Q \leq L$, and $0 \leq W - 2P - 2Q \leq 1 - R - 2L$. Fix ball-collision parameters $p_1, p_2, q_1, q_2, k_1, k_2, \ell_1, \ell_2$ with $p_i/n \to P$, $q_i/n \to Q$, $k_i/n \to R/2$, and $\ell_i/n \to L$.

We have also analyzed more general asymptotic parameter spaces, for example splitting P into P_1, P_2 where $p_i/n \to P_i$. Balanced parameters always turned out to be asymptotically optimal (as one would expect), so this section focuses on the parameter space (P, Q, L) stated above. Note that the asymptotic optimality of $P_1 = P_2$ does *not* imply the concrete optimality of $p_1 = p_2$; for example, $(p_1, p_2) = (2, 1)$ appears to be optimal for some small input sizes.

In the formulas below, expressions of the form $x \log_2 x$ are extended (continuously but not differentiably) to 0 at $x = 0$. For example, the expression $P \log_2 P$ means 0 if $P = 0$.

Success probability. We repeatedly invoke the standard asymptotic formula for binomial coefficients, namely

$$\frac{1}{n} \log_2 \binom{(\alpha + o(1))n}{(\beta + o(1))n} \to \alpha \log_2 \alpha - \beta \log_2 \beta - (\alpha - \beta) \log_2 (\alpha - \beta),$$

to compute the asymptotic exponent of the success probability of a single iteration of ball-collision decoding:

$$\begin{aligned} B(P,Q,L) &= \lim_{n\to\infty} \frac{1}{n} \log_2 \left(\binom{n}{w}^{-1} \binom{n-k-\ell_1-\ell_2}{w-p_1-p_2-q_1-q_2} \binom{k_1}{p_1} \binom{k_2}{p_2} \binom{\ell_1}{q_1} \binom{\ell_2}{q_2} \right) \\ &= W \log_2 W + (1-W) \log_2 (1-W) \\ &\quad + (1-R-2L) \log_2 (1-R-2L) - (W-2P-2Q) \log_2 (W-2P-2Q) \\ &\quad - (1-R-2L-(W-2P-2Q)) \log_2 (1-R-2L-(W-2P-2Q)) \\ &\quad + R \log_2 (R/2) - 2P \log_2 P - (R-2P) \log_2 (R/2-P) \\ &\quad + 2L \log_2 L - 2Q \log_2 Q - 2(L-Q) \log_2 (L-Q). \end{aligned}$$

The success probability of a single iteration is asymptotically $2^{n(B(P,Q,L)+o(1))}$.

Iteration cost. We similarly compute the asymptotic exponent of the cost of an iteration:

$$\begin{aligned}C(P,Q,L) &= \lim_{n\to\infty}\frac{1}{n}\log_2\left(\binom{k_1}{p_1}\binom{\ell_1}{q_1}+\binom{k_2}{p_2}\binom{\ell_2}{q_2}+\binom{k_1}{p_1}\binom{\ell_1}{q_1}\binom{k_2}{p_2}\binom{\ell_2}{q_2}2^{-\ell_1-\ell_2}\right)\\ &= \max\{(R/2)\log_2(R/2)-P\log_2 P-(R/2-P)\log_2(R/2-P)\\ &\qquad +L\log_2 L-Q\log_2 Q-(L-Q)\log_2(L-Q),\\ &\qquad R\log_2(R/2)-2P\log_2 P-(R-2P)\log_2(R/2-P)\\ &\qquad +2L\log_2 L-2Q\log_2 Q-2(L-Q)\log_2(L-Q)-2L\}.\end{aligned}$$

The cost of a single iteration is asymptotically $2^{n(C(P,Q,L)+o(1))}$. Note that we have simplified the iteration cost to $\binom{k_1}{p_1}\binom{\ell_1}{q_1}+\binom{k_2}{p_2}\binom{\ell_2}{q_2}+\binom{k_1}{p_1}\binom{\ell_1}{q_1}\binom{k_2}{p_2}\binom{\ell_2}{q_2}2^{-\ell_1-\ell_2}$. The cost is actually larger than this, but only by a factor $\le \mathrm{poly}(n)$, which we are free to disregard since $\frac{1}{n}\log_2\mathrm{poly}(n)\to 0$. We also comment that the bounds are valid whether or not $q_i = 0$.

Overall attack cost. The overall asymptotic ball-collision-decoding-cost exponent is the difference $D(P,Q,L)$ of the iteration-cost exponent $C(P,Q,L)$ and the success-probability exponent $B(P,Q,L)$.

For example, take $W = 0.04$ and $R = 1+W\log_2 W+(1-W)\log_2(1-W) = 0.7577078109\ldots$. Choose $P = 0.004203556640625$, $Q = 0.000192998046875$, and $L = 0.017429431640625$. The success-probability exponent is $-0.0458435310\ldots$, and the iteration-cost exponent is $0.0348588632\ldots$, so the ball-collision decoding exponent is $0.0807023942\ldots$. Ball-collision decoding with these parameters therefore costs $2^{(0.0807023942\ldots+o(1))n}$ to correct $(0.04+o(1))n$ errors in a code of rate $0.7577078109\ldots+o(1)$.

Collision-decoding cost and the lower bound. Traditional collision decoding is the special case $p_1 = p_2$, $k_1 = k_2$, $\ell_1 = \ell_2$, $q_1 = q_2 = 0$ of ball-collision decoding. Its asymptotic cost exponent is the case $Q = 0$ of the ball-collision decoding exponent stated above.

Consider again $W = 0.04$ and $R = 1 + W\log_2 W + (1-W)\log_2(1-W)$. Choosing $P = 0.00415087890625$, $Q = 0$, and $L = 0.0164931640625$ achieves decoding exponent $0.0809085120\ldots$. We partitioned the (P,L) space into small intervals and performed interval-arithmetic calculations to show that $Q = 0$ cannot do better than 0.0809; ball-collision decoding therefore has a slightly smaller exponent than collision decoding in this case.

We performed similar calculations for other pairs (W,R) and in each case found that the infimum of all collision-decoding-cost exponents was beaten by a ball-collision-decoding-cost exponent. Ball-collision decoding therefore has a smaller exponent than collision decoding, as stated in the introduction of this paper.

The case $Q = 0$ is always suboptimal. The interval-arithmetic calculations described above are proofs of the suboptimality of $Q = 0$ for some specific pairs (W,R). These proofs have the advantage of computing explicit bounds on the collision-decoding-cost exponents for those pairs (W,R), but the proofs have two obvious disadvantages.

The first disadvantage is that these proofs do not cover *all* pairs (W,R); they leave open the possibility that ball-collision decoding has the same exponent as collision decoding for other pairs (W,R). The second disadvantage is that the proofs are much too long to verify by hand. The first disadvantage could perhaps be addressed by much more extensive interval-arithmetic calculations, partitioning the space of pairs (W,R) into boxes so small that, within each box, the ball-collision-decoding exponent is uniformly better than the minimum collision-decoding exponent; but this would exacerbate the second disadvantage.

To address both of these disadvantages we give, in Appendix A, a proof that $Q = 0$ is always suboptimal: for *every* (W, R), ball-collision decoding has a smaller asymptotic cost exponent than collision decoding. Specifically, we prove the following theorem about the overall asymptotic cost exponent:

Theorem 7.1 *For each R, W it holds that*

$$\begin{aligned}&\min\{D(P,0,L) : 0 \le P \le R/2,\ 0 \le W - 2P \le 1 - R - 2L\}\\ >\ &\min\{D(P,Q,L) : 0 \le P \le R/2,\ 0 \le Q \le L,\ 0 \le W - 2P - 2Q \le 1 - R - 2L\}.\end{aligned}$$

Note that $\{(P,0,L)\}$ and $\{(P,Q,L)\}$ are compact sets, and D is continuous, so we are justified in writing "min" rather than "inf". The proof relies on one small computer calculation (proving that $(-W \log_2 W)X > 2W$ for $0 < W \le 0.1$, where X is a function of W defined in the appendix) but aside from this is completely hand-verifiable.

Asymptotics for non-constant error fractions. Constant rates and constant error fractions are traditional in the study of coding-theory asymptotics, but they are not exactly right in the study of code-based cryptography. McEliece uses error fraction approximately $(1 - R)/\log_2 n$, and $1/\log_2 n$ slowly decreases to 0 as $n \to \infty$. Asymptotics for collision-decoding cost in this context appeared recently in [10], and in general appear to have the form

$$(1 - R)^{-(1-R)n/\log_2 n + (\text{constant} + o(1))n/(\log_2 n)^2}.$$

With some effort one can use the same techniques to check that the ball-collision-decoding speedup factor is asymptotically $2^{(c+o(1))n/(\log_2 n)^2}$ with $c > 0$. This factor is asymptotically much larger than any of the recent speedups discussed in [9] and [32].

8 Choosing McEliece parameters

The traditional approach to selecting cryptosystem parameters is as follows:

- Consider the fastest known attacks against the system. For example, in the case of RSA, consider the latest refinements [47] of the number-field sieve.
- Restrict attention to parameters for which these attacks take time at least $2^{b+\delta}$. Here b is the desired security level, and δ is a "security margin" meant to protect against the possibility of further improvements in the attacks.
- Within the remaining parameter space, choose the most efficient parameters. The definition of efficiency depends on the target application: it could mean minimal key size, for example, or minimum decryption time.

This approach does not make clear how to choose the security margin δ. Some applications have ample time and space for cryptography, and can simply increase δ to the maximum value for which the costs of cryptography are still insignificant; but in some applications cryptography is an important bottleneck, and users insist on minimizing δ for the sake of performance.

Finiasz and Sendrier in [32] identified a bound on "future improvements" in attacks against the McEliece cryptosystem, and suggested that designers use this bound to "choose durable parameters". The general idea of identifying bottlenecks in any possible attack, and of using those bottlenecks to systematically choose δ, is quite natural and attractive, and has been used

successfully in many contexts. However, ball-collision decoding disproves the specific bound in [32], leaving open the question of how the general idea can be applied to the McEliece cryptosystem.

We propose replacing the bound in [32] with the simpler bound

$$\min\left\{\frac{1}{2}\binom{n}{w}\binom{n-k}{w-p}^{-1}\binom{k}{p}^{-1/2} : p \geq 0\right\};$$

i.e., choosing the code length n, code rate k/n, and error fraction w/n so that this bound is at least 2^b. As usual, implementors can exploit the remaining flexibility in parameters to optimize decryption time, compressed key size $k(n-k)$, or efficiency in any other metric of interest.

This bound has several attractive features. It is easy to estimate via standard binomial-coefficient approximations. It is easy to compute exactly. It covers a very wide class of attacks, as we explain in a moment. It is nevertheless in the same ballpark as the cost of known attacks: for example, it is $2^{49.69}$ for the original parameters $(n,k,w) = (1024,524,50)$, and $2^{236.49}$ for $(n,k,w) = (6624,5129,117)$. Note that these numbers give lower bounds on the cost of the attack. Parameters protecting against this bound pay only about a 20% performance penalty at high security levels, compared to parameters that merely protect against known attacks.

The reader can easily verify that parameters $(n,k,w) = (3178,2384,68)$ achieve 128-bit security against this bound. For 256-bit security $(n,k,w) = (6944,5208,136)$ are recommended.

Here is the class of attacks mentioned above. Assume that each iteration of the attack chooses an information set, hoping for exactly p errors in the set; that the choices of information sets are independent of the target syndrome; that each iteration considers at least $\binom{k}{p}^{1/2}$ error patterns within the information set; and that testing each pattern costs at least 1. The $\binom{k}{p}^{1/2}$ iterations model the cost of a birthday-type attack on all vectors of length k with Hamming weight p.

For each $\epsilon \geq 0$, a cost bound of $\epsilon\binom{n}{w}\binom{n-k}{w-p}^{-1}\binom{k}{p}^{-1/2}$ allows at most $\epsilon\binom{n}{w}\binom{n-k}{w-p}^{-1}\binom{k}{p}^{-1}$ iterations, and each iteration covers at most $\binom{n-k}{w-p}\binom{k}{p}$ patterns of w errors, so overall the iterations cover at most $\epsilon\binom{n}{w}$ possible patterns; i.e., the attack succeeds with probability at most ϵ. The average attack time is therefore at least $\frac{1}{2}\binom{n}{w}\binom{n-k}{w-p}^{-1}\binom{k}{p}^{-1/2}$. Note that batching attacks, i.e., attacking multiple targets at once, does not provide any benefits in this approach. Thus the Johansson–Jonsson speedups for attacking batches of McEliece ciphertexts [41] are subject to the same bound, as are the Fossorier–Kobara–Imai speedups [33].

One can object that this class does not include, e.g., attacks that hope for *at most* p errors in the information set, or attacks that consider fewer error patterns per iteration at the expense of success probability. One can object, in the opposite direction, that the conditional success probability per error pattern inspected is actually a constant factor smaller than the $\binom{k}{p}^{-1/2}$ hypothesized above; see generally [32, Appendix A]. A more complicated bound that accounts for these variations and limitations would be slightly larger than the bound stated above but would also be more difficult to compute; our view is that a simpler, slightly smaller bound is more useful. In any event, it is clear that beating this bound would be an astonishing breakthrough.

References

[1] Carlisle M. Adams, Henk Meijer, *Security-related comments regarding McEliece's public-key cryptosystem*, in Crypto '87 [60] (1987), 224–228; see also newer version [2]. MR 0956653. Citations in this document: §4.

[2] Carlisle M. Adams, Henk Meijer, *Security-related comments regarding McEliece's public-key cryptosystem*, IEEE Transactions on Information Theory **35** (1988), 454–455; see also older version [1]. MR 0999658. Citations in this document: §1, §4.

[3] Abdulrahman Al Jabri, *A statistical decoding algorithm for general linear block codes*, in IMA 2001 [39] (2001), 1–8. MR 2074098. Citations in this document: §4.

[4] Alexei E. Ashikhmin, Alexander Barg, *Minimal vectors in linear codes*, IEEE Transactions on Information Theory **44** (1998), 2010–2017. Citations in this document: §4.

[5] Alexander Barg, Evgueni A. Krouk, Henk C. A. van Tilborg, *On the complexity of minimum distance decoding of long linear codes*, IEEE Transactions on Information Theory **45** (1999), 1392–1405. Citations in this document: §4.

[6] Lynn Batten, Reihaneh Safavi-Naini (editors), *Information security and privacy: 11th Australasian conference, ACISP 2006, Melbourne, Australia, July 35, 2006, proceedings*, Lecture Notes in Computer Science, 4058, Springer, 2006. See [57].

[7] Daniel J. Bernstein, *Grover vs. McEliece*, in Post-Quantum Cryptography [64] (2010), 72–80. Citations in this document: §1.

[8] Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen (editors), *Post-quantum cryptography*, Springer, 2009. ISBN 978-3-540-88701-0. See [58].

[9] Daniel J. Bernstein, Tanja Lange, Christiane Peters, *Attacking and defending the McEliece cryptosystem*, in PQCrypto 2008 [13] (2008), 31–46. URL: http://eprint.iacr.org/2008/318. Citations in this document: §1, §1, §1, §3, §3, §4, §4, §6, §7.

[10] 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: §5, §7.

[11] Thomas A. Berson, *Failure of the McEliece public-key cryptosystem under message-resend and related-message attack*, in Crypto '97 [45] (1997), 213–220. Citations in this document: §1.

[12] Mario Blaum, Patrick G. Farrell, Henk C. A. van Tilborg (editors), *Information, coding and mathematics*, Kluwer International Series in Engineering and Computer Science, 687, Kluwer, 2002. MR 2005a:94003. See [71].

[13] Johannes Buchmann, Jintai Ding (editors), *Post-quantum cryptography, second international workshop, PQCrypto 2008, Cincinnati, OH, USA, October 17-19, 2008, proceedings*, Lecture Notes in Computer Science, 5299, Springer, 2008. See [9].

[14] Paul Camion, Pascale Charpin, Sami Harari (editors), *Eurocode '92: proceedings of the international symposium on coding theory and applications held in Udine, October 23–30, 1992*, Springer, 1993. ISBN 3-211-82519-3. MR 94k:94001. See [20].

[15] Anne Canteaut, Herve Chabanne, *A further improvement of the work factor in an attempt at breaking McEliece's cryptosystem*, in EUROCODE 94 [21] (1994). URL: http://www.inria.fr/rrrt/rr-2227.html. Citations in this document: §4.

[16] Anne Canteaut, Florent Chabaud, *A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511*, IEEE Transactions on Information Theory **44** (1998), 367–378. MR 98m:94043. URL: ftp://ftp.inria.fr/INRIA/tech-reports/RR/RR-2685.ps.gz. Citations in this document: §3, §4.

[17] Anne Canteaut, Nicolas Sendrier, *Cryptanalysis of the original McEliece cryptosystem*, in Asiacrypt '98 [56] (1998), 187–199. MR 2000i:94042. Citations in this document: §3, §4.

[18] Aydano B. Carleial, Martin E. Hellman, *A note on Wyner's wiretap channel*, in IEEE Transactions on Information Theory **23** (1977), 387–390. ISSN 0018-9448.

[19] Herve Chabanne, B. Courteau, *Application de la méthode de décodage itérative d'Omura à la cryptanalyse du système de McEliece*, Université de Sherbrooke, Rapport de Recherche, number 122 (1993). Citations in this document: §4.

[20] Florent Chabaud, *Asymptotic analysis of probabilistic algorithms for finding short codewords*, in [14] (1993), 175–183. MR 95e:94026. Citations in this document: §4.

[21] Pascale Charpin (editor), *EUROCODE 94*, 1994. See [15].

[22] George C. Clark, Jr., J. Bibb Cain, *Error-correcting coding for digital communication*, Plenum, 1981. ISBN 0-306-40615-2. Citations in this document: §4.

[23] Christophe Clavier, Kris Gaj (editors), *Cryptographic hardware and embedded systems — CHES 2009, 11th international workshop, Lausanne, Switzerland, September 6–9, 2009, proceedings*, Lecture Notes in Computer Science, 5747, Springer, 2009. ISBN 978-3-642-04137-2. See [30].

[24] John T. Coffey, Rodney M. Goodman, *The complexity of information set decoding*, IEEE Transactions on Information Theory **35** (1990), 1031–1037. Citations in this document: §4.

[25] John T. Coffey, Rodney M. Goodman, P. Farrell, *New approaches to reduced complexity decoding*, Discrete and Applied Mathematics **33** (1991), 43–60. Citations in this document: §4, §5.

[26] Gérard D. Cohen, Jacques Wolfmann (editors), *Coding theory and applications*, Lecture Notes in Computer Science, 388, Springer, 1989. See [67].

[27] Ilya I. Dumer, *Two decoding algorithms for linear codes*, Problemy Peredachi Informatsii **25** (1989), 24–32. Citations in this document: §4.

[28] Ilya I. Dumer, *On minimum distance decoding of linear codes*, in [44] (1991), 50–52. Citations in this document: §4.

[29] Cynthia Dwork (editor), *Advances in cryptology — CRYPTO 2006, 26th annual international cryptology conference, Santa Barbara, California, USA, August 20–24, 2006, proceedings*, Lecture Notes in Computer Science, 4117, Springer, 2006. ISBN 3-540-37432-9. See [43].

[30] Thomas Eisenbarth, Tim Güneysu, Stefan Heyse, Christof Paar, *MicroEliece: McEliece for embedded devices*, in CHES 2009 [23] (2009), 49–64. Citations in this document: §1.

[31] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, Jean-Pierre Tillich, *Algebraic cryptanalysis of McEliece variants with compact keys*, in Advances in Cryptology – EUROCRYPT 2010 [35] (2010), 279–298. Citations in this document: §1.

[32] Matthieu Finiasz, Nicolas Sendrier, *Security bounds for the design of code-based cryptosystems*, in Asiacrypt 2009 [53] (2009). URL: http://eprint.iacr.org/2009/414. Citations in this document: §1, §1, §4, §2, §3, §7, §7, §8, §8, §8, §8.

[33] Marc P. C. Fossorier, Kazukuni Kobara, Hideki Imai, *Modeling bit flipping decoding based on nonorthogonal check sums with application to iterative decoding attack of McEliece cryptosystem*, IEEE Transactions on Information Theory **53** (2007), 402–411. MR 2007m:94158. Citations in this document: §8.

[34] Valerie Gauthier Umana, Gregor Leander, *Practical key recovery attacks on two McEliece variants* (2009). URL: http://eprint.iacr.org/2009/509.pdf. Citations in this document: §1.

[35] Henri Gilbert (editor), *Advances in Cryptology — EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010, proceedings*, Lecture Notes in Computer Science, 6110, Springer. ISBN 978-3-642-13189-9. See [31].

[36] 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 [65].

[37] Shafi Goldwasser (editor), *Advances in cryptology — CRYPTO '88, proceedings of the conference on the theory and application of cryptography held at the University of California, Santa Barbara, California, August 21–25, 1988*, Lecture Notes in Computer Science, 403, Springer, 1990. ISBN 3-540-97196-3. MR 90j:94003. See [68].

[38] Christoph G. Günther, *Advances in cryptology — EUROCRYPT '88, proceedings of the workshop on the theory and application of cryptographic techniques held in Davos, May 25–27, 1988*, Lecture Notes in Computer Science, 330, Springer-Verlag, Berlin, 1988. ISBN 3-540-50251-3. MR 90a:94002. See [50].

[39] Bahram Honary (editor), *Cryptography and coding: proceedings of the 8th IMA international conference held in Cirencester, December 17–19, 2001*, Lecture Notes in Computer Science, 2260, Springer, 2001. See [3].

[40] Michael J. Jacobson Jr., Vincent Rijmen, Reihaneh Safavi-Naini (editors), *Selected Areas in Cryptography*, Lecture Notes in Computer Science, 5867, Springer, 2009. See [55].

[41] Thomas Johansson, Fredrik Jonsson, *On the complexity of some cryptographic problems based on the general decoding problem*, IEEE Transactions on Information Theory **48** (2002), 2669–2678. URL: http://www.it.lth.se/cryptology/e-papers/paper054.pdf. Citations in this document: §8.

[42] Antoine Joux, Reynald Lercier, *The function field sieve in the medium prime case*, in Eurocrypt 2006 [70] (2006), 254–270. Citations in this document: §1.

[43] Antoine Joux, Reynald Lercier, Nigel P. Smart, Frederik Vercauteren, *The number field sieve in the medium prime case*, in Crypto 2006 [29] (2006), 326–344. Citations in this document: §1.

[44] Grigori A. Kabatianskii (editor), *Fifth joint Soviet-Swedish international workshop on information theory, Moscow, 1991*, 1991. See [28].

[45] Burton S. Kaliski Jr. (editor), *Advances in cryptology — CRYPTO '97: 17th annual international cryptology conference, Santa Barbara, California, USA, August 17–21, 1997, proceedings*, Lecture Notes in Computer Science, 1294, Springer, 1997. ISBN 3-540-63384-7. MR 99a:94041. See [11].

[46] Kwangjo Kim (editor), *Public key cryptography: proceedings of the 4th international workshop on practice and theory in public key cryptosystems (PKC 2001) held on Cheju Island, February 13–15, 2001*, Lecture Notes in Computer Science, 1992, Springer, 2001. See [48].

[47] Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen K. Lenstra, Emmanuel Thomé, Joppe W. Bos, Pierrick Gaudry, Alexander Kruppa, Peter L. Montgomery, Dag Arne Osvik, Herman te Riele, Andrey Timofeev, Paul Zimmermann, *Factorization of a 768-bit RSA modulus*, in CRYPTO 2010 [62] (2010), 333–350. URL: http://eprint.iacr.org/2010/006. Citations in this document: §8.

[48] Kazukuni Kobara, Hideki Imai, *Semantically secure McEliece public-key cryptosystems — conversions for McEliece PKC*, in PKC 2001 [46] (2001), 19–35. MR 2003c:94027. Citations in this document: §1.

[49] Evgueni A. Krouk, *Decoding complexity bound for linear block codes*, Problemy Peredachi Informatsii **25** (1989), 103–107. Citations in this document: §4, §4.

[50] Pil Joong Lee, Ernest F. Brickell, *An observation on the security of McEliece's public-key cryptosystem*, in Eurocrypt '88 [38] (1988), 275–280. MR 0994669. URL: http://dsns.csie.nctu.edu.tw/research/crypto/HTML/PDF/E88/275.PDF. Citations in this document: §4.

[51] Arjen K. Lenstra, Hendrik W. Lenstra, Jr. (editors), *The development of the number field sieve*, Lecture Notes in Mathematics, 1554, Springer-Verlag, Berlin, 1993. ISBN 3-540-57013-6. MR 96m:11116. Citations in this document: §1.

[52] Jeffrey S. Leon, *A probabilistic algorithm for computing minimum weights of large error-correcting codes*, IEEE Transactions on Information Theory **34** (1988), 1354–1359. MR 89k:94072. Citations in this document: §4.

[53] Mitsuru Matsui (editor), *Advances in cryptology — ASIACRYPT 2009, 15th international conference on the theory and application of cryptology and information security, Tokyo, Japan, December 6–10, 2009, proceedings*, Lecture Notes in Computer Science, 5912, Springer, 2009. ISBN 978-3-642-10365-0. See [32].

[54] 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, §4.

[55] Rafael Misoczki, Paulo S. L. M. Barreto, *Compact McEliece keys from Goppa codes*, in SAC 2009 [40] (2009), 376–392. Citations in this document: §1.

[56] Kazuo Ohta, Dingyi Pei (editors), *Advances in cryptology — ASIACRYPT'98: proceedings of the international conference on the theory and application of cryptology and information security held in Beijing*, Lecture Notes in Computer Science, 1514, Springer, 1998. ISBN 3-540-65109-8. MR 2000h:94002. See [17].

[57] Raphael Overbeck, *Statistical decoding revisited*, in ACISP 2006 [6] (2006), 283–294. Citations in this document: §4.

[58] Raphael Overbeck, Nicolas Sendrier, *Code-based cryptography*, in [8] (2009), 95–145. Citations in this document: §2, §4.

[59] Christiane Peters, *Information-set decoding for linear codes over* $\mathbf{F}_q$, in Post-Quantum Cryptography [64] (2010), 81–94. Citations in this document: §4, §2.

[60] Carl Pomerance (editor), *Advances in cryptology — CRYPTO '87, proceedings of the conference on the theory and applications of cryptographic techniques held at the University of California, Santa Barbara, California, August 16–20, 1987*, Lecture Notes in Computer Science, 293, Springer, 1987. ISBN 3-540-18796-0. MR 89b:68005. URL: http://dsns.csie.nctu.edu.tw/research/crypto/HTML/PDF/C87/224.PDF. See [1].

[61] Eugene Prange, *The use of information sets in decoding cyclic codes*, IRE Transactions on Information Theory **IT-8** (1962), S5–S9. Citations in this document: §4.

[62] Tal Rabin (editor), *Advances in Cryptology — CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010, proceedings*, Lecture Notes in Computer Science, 6223, Springer, 2010. See [47].

[63] Ronald L. Rivest, Adi Shamir, Leonard M. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM **21** (1978), 120–126. ISSN 0001–0782. Citations in this document: §1.

[64] Nicolas Sendrier (editor), *Post-quantum cryptography, third international workshop, PQCrypto, Darmstadt, Germany, May 25-28, 2010, proceedings*, Lecture Notes in Computer Science, 6061, Springer, 2010. See [7], [59].

[65] Peter W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring.*, in FOCS 1994 [36] (1994), 124–134; see also newer version [66]. MR 1489242. Citations in this document: §1.

[66] 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 [65]. MR MR 98i:11108.

[67] Jacques Stern, *A method for finding codewords of small weight*, in [26] (1989), 106–113. Citations in this document: §3, §3, §4, §4.

[68] Johan van Tilburg, *On the McEliece public-key cryptosystem*, in Crypto '88 [37] (1990), 119–131. MR 1046386. Citations in this document: §4.

[69] Johan van Tilburg, *Security-analysis of a class of cryptosystems based on linear error-correcting codes*, Ph.D. thesis, Technische Universiteit Eindhoven, 1994. ISBN 90-72125-45-2. MR 95k:94025. Citations in this document: §4.

[70] Serge Vaudenay (editor), *Advances in cryptology — EUROCRYPT 2006, 25th annual international conference on the theory and applications of cryptographic techniques, St. Petersburg, Russia, May 28–June 1, 2006, proceedings*, Lecture Notes in Computer Science, 4004, Springer, 2006. ISBN 3-540-34546-9. See [42].

[71] Eric R. Verheul, Jeroen M. Doumen, Henk C. A. van Tilborg, *Sloppy Alice attacks! Adaptive chosen ciphertext attacks on the McEliece public-key cryptosystem*, in [12] (2002), 99–119. MR 2005b:94041. Citations in this document: §1.

A Proof of suboptimality of $Q = 0$ (Theorem 7.1)

This appendix shows that, for each pair (W, R) within the range considered in Section 7, there are asymptotic parameters (P, Q, L) for ball-collision decoding whose cost exponents are smaller than the minimum collision-decoding-cost exponent, i.e., smaller than the minimum cost exponent for parameters $(P, 0, L)$.

Input space and parameter space. Throughout this appendix W and R are real numbers with $0 < W < 1/2$ and $-W \log_2 W - (1-W)\log_2(1-W) \le 1 - R < 1$.

The *parameter space* is the set of vectors (P, Q, L) of real numbers satisfying $0 \le P \le R/2$, $0 \le Q \le L$, and $0 \le W - 2P - 2Q \le 1 - R - 2L$. This parameter space depends implicitly on W and R.

Section 7 considers codes of length n and dimension k, and errors of weight w, where $n \to \infty$, $k/n \to R$, and $w/n \to W$. The ball-collision parameters $p_1, p_2, q_1, q_2, k_1, k_2, \ell_1, \ell_2$ satisfy $p_i/n \to P$, $q_i/n \to Q$, $k_i/n \to R/2$, and $\ell_i/n \to L$. The proof does not rely on this coding-theoretic interpretation of W, R, P, Q, L, but readers already familiar with collision decoding may find the interpretation helpful in understanding Lemma A.1 below.

Cost exponent for collision decoding. Most of the proof consists of analyzing the asymptotic cost exponent $D(P, 0, L)$ for collision decoding, namely

$$\begin{aligned}&\max\{-(R/2)\log_2(R/2) + P\log_2 P + (R/2-P)\log_2(R/2-P), -2L\}\\&- W\log_2 W - (1-W)\log_2(1-W) - (1-R-2L)\log_2(1-R-2L)\\&+ (W-2P)\log_2(W-2P) + (1-R-2L-(W-2P))\log_2(1-R-2L-(W-2P)).\end{aligned}$$

As mentioned earlier, $D(P, 0, L)$ is a continuous function of the parameters $(P, 0, L)$, and the parameter space is compact, so there exist optimal collision-decoding parameters $(P, 0, L)$, i.e., parameters that achieve the infimum of collision-decoding costs. This does not imply, and the proof of Theorem 7.1 does not use, *uniqueness* of the optimal parameters.

Optimal collision-decoding parameters. The proof that ball-collision decoding beats collision decoding relies on the following three facts about optimal collision-decoding parameters $(P, 0, L)$:

Lemma A.1 *If (P, L) are optimal collision-decoding parameters than*

$$0 < L; \qquad 0 < W - 2P; \qquad \text{and} \qquad W - 2P < (1 - R - 2L)/2.$$

In other words, the collision space $\mathbf{F}_2^{\ell_1+\ell_2}$ is asymptotically quite large, and the uncontrolled $n-k_1-k_2-\ell_1-\ell_2$ positions include asymptotically many error positions, although asymptotically more non-error positions than error positions.

We do not claim that the three facts in Lemma A.1 are news to the many authors who have written previous papers on collision decoding. However, we have not found *proofs* of these facts in the literature, so for completeness we include proofs here.

The proofs do not require any background in coding theory. The main tool is nothing more than basic calculus. In order to study the growth of the collision-cost exponent induced by an increase or decrease in the values of P and L, we use the Taylor-series expansion of the logarithm function: for example, a term such as $(L+\epsilon)\log_2(L+\epsilon)$ has series expansion $L\log_2 L+\epsilon\log_2(eL)+O(\epsilon^2)$ around $\epsilon=0$. Here $e=\exp(1)$ is Euler's constant. Beware that extra work is required in moving away from corners of the parameter space: for example, $L\log_2 L$ is not differentiable at $L=0$.

How to improve upon optimal collision-decoding parameters. Before proving the three parts of Lemma A.1 we show how to deduce Theorem 7.1 from Lemma A.1. The proof is constructive, showing how to slightly adjust optimal parameters for collision decoding to obtain better parameters for ball-collision decoding.

Proof (of Theorem 7.1). Start with optimal collision-decoding parameters $(P,0,L)$. Now consider the impact of increasing Q from 0 to δ and increasing L by $-(1/2)\delta\log_2\delta$, for very small δ. Of course, the increase in Q requires generalizing from collision decoding to ball-collision decoding. Lemma A.1 says that optimal collision-decoding parameters $(P,0,L)$ must have $0<L$ and $0<W-2P<(1-R-2L)/2$; consequently the parameter space has room for Q and L to increase.

The quantity $L\log_2 L-Q\log_2 Q-(L-Q)\log_2(L-Q)$ increases by $-\delta\log_2\delta+O(\delta)$, and $2L\log_2 L-2Q\log_2 Q-2(L-Q)\log_2(L-Q)-2L$ also increases by $-\delta\log_2\delta+O(\delta)$. The iteration-cost exponent therefore increases by $-\delta\log_2\delta+O(\delta)$. The success-probability exponent increases by $\delta\log_2\delta\log_2(e(1-R-2L))-\delta\log_2\delta\log_2(e(1-R-2L-(W-2P)))-2\delta\log_2\delta+O(\delta)$. The total cost exponent therefore increases by $(\delta\log_2\delta)(1+\log_2(1-R-2L-(W-2P))-\log_2(1-R-2L))+O(\delta)$.

Rewrite $W-2P<(1-R-2L)/2$ as $1+\log_2(1-R-2L-(W-2P))-\log_2(1-R-2L)>0$, and deduce that the increase in the cost exponent is negative for all sufficiently small $\delta>0$; note here that $\log_2\delta$ is negative, and that $O(\delta)/(\delta\log_2\delta)\to 0$ as $\delta\to 0$. Consequently the optimal collision-decoding parameters $(P,0,L)$ are beaten by $(P,\delta,L-(1/2)\delta\log_2\delta)$ for all sufficiently small $\delta>0$. □

How to optimize collision decoding. We build up to Lemma A.1 using several lemmas. The first lemma requires the most difficult calculation, establishing a useful inequality. The next three lemmas show that optimal collision-decoding parameters (P,L) can never have $L=0$: Lemma A.3 covers the case $P=0$; Lemma A.4 covers the case $P=R/2$; Lemma A.5 covers the intermediate cases $0<P<R/2$. Each of the proofs is constructive, showing how to move from $(P,0)$ to better collision-decoding parameters.

The next two lemmas show similarly that optimal collision-decoding parameters (P,L) cannot have $0=W-2P=1-R-2L$, and cannot have $0=W-2P<1-R-2L$, so they must have $0<W-2P$. Proving Lemma A.1 then boils down to proving $W-2P<(1-R-2L)/2$; that proof concludes the appendix.

If $(P',0,L')$ and $(P,0,L)$ are in the parameter space and $D(P',0,L') < D(P,0,L)$ then we say that (P',L') *improves upon* (P,L). We also say that (P',L') improves upon (P,L) in the vacuous case that $(P,0,L)$ is not in the parameter space.

Lemma A.2 *Each* $(P,0,L)$ *in the parameter space satisfies*

$$1-R-((R/2)\log_2(R/2)-P\log_2 P-(R/2-P)\log_2(R/2-P))>2W.$$

Proof. The proof proceeds in two steps. The first step handles $0.1 < W < 0.5$. The second step handles $0 < W \le 0.1$.

Recall that $1-R \ge -W\log_2 W-(1-W)\log_2(1-W)$. Hence $1-(3/2)R-2W \ge -(1/2)-2W-(3/2)W\log_2 W-(3/2)(1-W)\log_2(1-W)$. The values of this lower bound at $0.1, 0.3, 0.5$ are $\approx 0.0034, \approx 0.2218, 0$ respectively; the derivative of the lower bound is $-2-(3/2)\log_2(eW)+(3/2)\log_2(e(1-W))$, which has a unique zero at $W = 1/(1+2^{4/3}) \approx 0.2841$; so the lower bound is positive for all W with $0.1 < W < 0.5$. In particular $2W < 1-(3/2)R$ if $0.1 < W < 0.5$.

The maximum possible value of $(R/2)\log_2(R/2)-P\log_2 P-(R/2-P)\log_2(R/2-P)$ is $(R/2)\log_2(R/2)-2(R/4)\log_2(R/4) = R/2$, so $1-R-((R/2)\log_2(R/2)-P\log_2 P-(R/2-P)\log_2(R/2-P)) \ge 1-(3/2)R > 2W$ if $0.1 < W < 0.5$. This concludes the case $0.1 < W < 0.5$.

From now on assume $0 < W \le 0.1$. Abbreviate $(1+W\log_2 W+(1-W)\log_2(1-W))/2$ as G. Then $G-W$ has derivative $(1/2)\log_2(eW)-(1/2)\log_2(e(1-W))-1 < 0$ for $0 < W \le 0.1$, and value $0.1655\ldots > 0$ at $W = 0.1$, so $G > W$ for $0 \le W \le 0.1$.

Furthermore $R/2 \le G$ by definition of G and the parameter space. So $(R/2)\log_2(R/2)-P\log_2 P-(R/2-P)\log_2(R/2-P) \le G\log_2 G-P\log_2 P-(G-P)\log_2(G-P)$; note that for any fixed $c > 0$ and $x > c$ the function $x\log_2 x-(x-c)\log_2(x-c)$ is increasing (check its derivative).

The parameter space forces $P \le W/2 < G/2$ so $G\log_2 G-P\log_2 P-(G-P)\log_2(G-P) \le G\log_2 G-(W/2)\log_2(W/2)-(G-(W/2))\log_2(G-(W/2))$.

Consequently using these inequalities and $R \le 2G$ one yields $1-R-((R/2)\log_2(R/2)-P\log_2 P-(R/2-P)\log_2(R/2-P)) \ge 1-R-(G\log_2 G-(W/2)\log_2(W/2)-(G-(W/2))\log_2(G-(W/2))) \ge (-W\log_2 W)X$ where

$$\begin{aligned}X &= \frac{1-2G-G\log_2 G+(W/2)\log_2(W/2)+(G-(W/2))\log_2(G-(W/2))}{-W\log_2 W}\\ &= \frac{1}{2}+\frac{-(1-W)\log_2(1-W)-G\log_2 G-(W/2)+(G-(W/2))\log_2(G-(W/2))}{-W\log_2 W}\\ &= \frac{1}{2}+\frac{1-W}{\log_2 W}\frac{\log_2(1-W)}{W}+\frac{G}{\log_2 W}\frac{\log_2 G-\log_2(G-(W/2))}{W}+\frac{1+\log_2(G-(W/2))}{2\log_2 W}\end{aligned}$$

Each of the ratios here is continuous for $0 \le W \le 0.1$. A straightforward interval-arithmetic calculation shows that $X \ge 0.5$ for $0 \le W \le 0.05$, implying $(-W\log_2 W)X > 2W$ for $0 < W \le 0.05$, and that $X \ge 0.7$ for $0.05 \le W \le 0.1$, implying $(-W\log_2 W)X > 2W$ for $0.05 < W \le 0.1$. Therefore $1-R-((R/2)\log_2(R/2)-P\log_2 P-(R/2-P)\log_2(R/2-P)) > 2W$ in all cases. □

Lemma A.3 *There is a real number* $\delta > 0$ *such that* $(\delta, -(1/2)\delta\log_2\delta)$ *improves upon* $(0,0)$.

Proof. The parameters $(P, L) = (\delta, -(1/2)\delta \log_2 \delta)$ satisfy the constraints $0 \leq P \leq R/2$, $0 \leq L$, and $0 \leq W - 2P \leq 1 - R - 2L$ for all sufficiently small real numbers $\delta \geq 0$, since $0 < R$ and $0 < W$. The collision-cost exponent is $\max\{\delta \log_2 \delta + O(\delta), \delta \log_2 \delta\} - (1-W)\log_2(1-W) - (1-R)\log_2(1-R) + (1-R-W)\log_2(1-R-W) - (\delta \log_2 \delta)\log_2(e(1-R)) + (\delta \log_2 \delta)\log_2(e(1-R-W)) + O(\delta) = -(1-W)\log_2(1-W) - (1-R)\log_2(1-R) + (1-R-W)\log_2(1-R-W) + (\delta \log_2 \delta)(1 + \log_2(1-R-W) - \log_2(1-R)) + O(\delta)$. The inequality $2W < 1 - R$ implies $1 + \log_2(1-R-W) - \log_2(1-R) > 0$, so $(\delta \log_2 \delta)(1 + \log_2(1-R-W) - \log_2(1-R)) + O(\delta)$ is negative for all sufficiently small $\delta > 0$, improving upon $(0, 0)$. □

Lemma A.4 *There is a real number $\delta > 0$ such that $(R/2 - \delta, -(1/2)\delta \log_2 \delta)$ improves upon $(R/2, 0)$.*

Proof. If $W < R$ then $(R/2, 0, 0)$ is outside the parameter space so the conclusion is vacuously satisfied. Assume from now on that $W \geq R$.

The parameters $(P, L) = (R/2 - \delta, -(1/2)\delta \log_2 \delta)$. satisfy the constraints $0 \leq P \leq R/2$, $0 \leq L$, and $0 \leq W - 2P \leq 1 - R - 2L$ for all sufficiently small real numbers $\delta \geq 0$. The iteration-cost exponent is $\max\{\delta \log_2 \delta + O(\delta), \delta \log_2 \delta\} - W \log_2 W - (1-R)\log_2(1-R) + (W - R + 2\delta)\log_2(W - R + 2\delta) - (\delta \log_2 \delta)\log_2(e(1-R)) + (\delta \log_2 \delta)\log_2(e(1-W)) + O(\delta) = -W \log_2 W - (1-R)\log_2(1-R) + (W - R + 2\delta)\log_2(W - R + 2\delta) + (\delta \log_2 \delta)(1 + \log_2(1-W) - \log_2(1-R)) + O(\delta)$.

If $W = R$ then $(W - R + 2\delta)\log_2(W - R + 2\delta) + (\delta \log_2 \delta)(1 + \log_2(1-W) - \log_2(1-R)) + O(\delta) = 3\delta \log_2 \delta + O(\delta)$. This is negative for all sufficiently small $\delta > 0$.

Otherwise $W > R$ so $(W - R + 2\delta)\log_2(W - R + 2\delta) + (\delta \log_2 \delta)(1 + \log_2(1-W) - \log_2(1-R)) + O(\delta) = (\delta \log_2 \delta)(1 + \log_2(1-W) - \log_2(1-R)) + O(\delta)$. This is also negative for all sufficiently small $\delta > 0$: recall that $2(1-W) > 1 > 1 - R$, so the coefficient $1 + \log_2(1-W) - \log_2(1-R)$ is positive.

Either way $(P, L) = (R/2 - \delta, -(1/2)\delta \log_2 \delta)$ improves upon $(R/2, 0)$. □

Lemma A.5 *If $0 < P < R/2$ then there is a real number $\delta > 0$ such that (P, δ) improves upon $(P, 0)$.*

Proof. Consider the impact of changing L from 0 to δ. The quantity $-(R/2)\log_2(R/2) + P \log_2 P + (R/2 - P)\log_2(R/2 - P)$ is negative and unchanged, and $-2L$ changes from 0 to -2δ, so $\max\{-(R/2)\log_2(R/2) + P \log_2 P + (R/2 - P)\log_2(R/2 - P), -2L\}$ increases by -2δ if δ is sufficiently small. The quantity $-(1 - R - 2L)\log_2(1 - R - 2L)$ increases by $2\delta \log_2(e(1-R)) + O(\delta^2)$. The quantity $(1 - R - 2L - (W - 2P))\log_2(1 - R - 2L - (W - 2P))$ increases by $-2\delta \log_2(e(1 - R - (W - 2P))) + O(\delta^2)$.

The total collision-cost exponent increases by $2\delta(-1 + \log_2(1-R) - \log_2(1 - R - (W - 2P))) + O(\delta^2)$. The coefficient $-1 + \log_2(1-R) - \log_2(1 - R - (W - 2P))$ is negative since $W - 2P < (1-R)/2$. Hence (P, δ) improves upon $(P, 0)$ for all sufficiently small $\delta > 0$. □

Lemma A.6 *There is a real number $c \geq 2$ satisfying the following condition: if $W < R$ then $c \log_2 c - (c-1)\log_2(c-1) > (1/2)(\log_2(R - W) - \log_2 W)$. For any such c there is a real number $\delta > 0$ such that $((W - \delta)/2, (1 - R - c\delta)/2)$ improves upon $(W/2, (1-R)/2)$.*

Proof. If $W > R$ then $(W/2, (1-R)/2)$ is outside the parameter space and the conclusions are vacuously satisfied for, e.g., $c = 2$ and $\delta = 1$. Assume from now on that $W \leq R$.

Choose a real number c large enough to meet both of the following constraints: first, $c \geq 2$; second, if $W < R$ then $c \log_2 c - (c-1)\log_2(c-1) > (1/2)(\log_2(R-W) - \log_2 W)$. This can always be done: $c \log_2 c - (c-1)\log_2(c-1) \to \infty$ as $c \to \infty$.

Consider the impact of changing L from $(1-R)/2$ to $(1-R-c\delta)/2$, and at the same time changing P from $W/2$ to $(W-\delta)/2$. This change fits the parameter constraints for sufficiently small $\delta > 0$.

The quantity $-(1-R-2L)\log_2(1-R-2L)$ changes from 0 to $-c\delta\log_2(c\delta)$. The quantity $(W-2P)\log_2(W-2P)$ changes from 0 to $\delta\log_2\delta$. The quantity $(1-R-2L-(W-2P))\log_2(1-R-2L-(W-2P))$ changes from 0 to $(c-1)\delta\log_2((c-1)\delta)$. The quantity $\max\{-(R/2)\log_2(R/2) + P\log_2 P + (R/2-P)\log_2(R/2-P), -2L\}$ is dominated by its first term since $2L = 1-R > 2W + ((R/2)\log_2(R/2) - P\log_2 P - (R/2-P)\log_2(R/2-P))$ by Lemma A.2.

It thus increases by $((W-\delta)/2)\log_2((W-\delta)/2) - (W/2)\log_2(W/2) + ((R-W+\delta)/2)\log_2((R-W+\delta)/2) - ((R-W)/2)\log_2((R-W)/2)$.

The total cost exponent increases by $\delta((c-1)\log_2(c-1) - c\log_2 c) + ((W-\delta)/2)\log_2((W-\delta)/2) - (W/2)\log_2(W/2) + ((R-W+\delta)/2)\log_2((R-W+\delta)/2) - ((R-W)/2)\log_2((R-W)/2)$.

If $W = R$ then this increase is $(\delta/2)\log_2(\delta/2) + O(\delta)$ and is therefore negative for all sufficiently small $\delta > 0$.

If $W < R$ then this increase is $((c-1)\log_2(c-1) - c\log_2 c + (1/2)(\log_2(e(R-W)/2) - \log_2(eW/2)))\delta + O(\delta^2)$, The coefficient of δ is negative by choice of c, so the increase is negative for all sufficiently small $\delta > 0$.

In all cases $((W-\delta)/2, (1-R-c\delta)/2)$ improves upon $(W/2, (1-R)/2)$. □

Lemma A.7 *Assume that $0 < 1-R-2L$. Then there is a real number $\delta > 0$ such that $((W-\delta)/2, L)$ improves upon $(W/2, L)$.*

Proof. Consider collision-decoding parameters (P, L) with $0 = W - 2P < 1 - R - 2L$. As before $P \leq R/2$ forces $W \leq R$.

Consider the impact of changing P from $W/2$ to $(W-\delta)/2$. This change fits the parameter constraints for sufficiently small $\delta > 0$.

The quantity $(W-2P)\log_2(W-2P)$ increases by $\delta\log_2\delta$. The quantity $(1-R-2L-(W-2P))\log_2(1-R-2L-(W-2P))$ increases by $O(\delta)$. The quantity

$$\max\{-(R/2)\log_2(R/2) + P\log_2 P + (R/2-P)\log_2(R/2-P), -2L\}$$

increases by something between 0 and $((W-\delta)/2)\log_2((W-\delta)/2) - (W/2)\log_2(W/2) + ((R-W+\delta)/2)\log_2((R-W+\delta)/2 - ((R-W)/2)\log_2((R-W)/2)$, which is $(\delta/2)\log_2(\delta/2) + O(\delta)$ if $W = R$ and $O(\delta)$ if $W < R$. The total increase in the cost is between $\delta\log_2\delta + O(\delta)$ and $(3/2)\delta\log_2\delta + O(\delta)$, and is therefore negative for all sufficiently small $\delta > 0$. □

Proof (of Lemma A.1). The hypothesis is that (P, L) minimizes $D(P, 0, L)$, i.e., that nothing improves upon (P, L).

The definition of the parameter space implies $L \geq 0$. Suppose that $L = 0$. Then $P < 0$ would contradict the definition of the parameter space; $P = 0$ would contradict Lemma A.3; $0 < P < R/2$ would contradict Lemma A.5; $P = R/2$ would contradict Lemma A.4; and $P > R/2$ would contradict the definition of the parameter space. Hence $L > 0$.

The definition of the parameter space also implies $0 \leq W - 2P$. Suppose that $0 = W - 2P$. Then $0 = 1 - R - 2L$ would force $(P, L) = (W/2, (1-R)/2)$, contradicting Lemma A.6;

$0 < 1 - R - 2L$ would contradict Lemma A.7; and $0 > 1 - R - 2L$ would contradict the definition of the parameter space. Hence $0 < W - 2P$.

Suppose that $2L > (R/2)\log_2(R/2) - P\log_2 P - (R/2 - P)\log_2(R/2 - P)$. Consider the impact of decreasing L by δ. The quantity

$$\max\{-(R/2)\log_2(R/2) + P\log_2 P + (R/2 - P)\log_2(R/2 - P), -2L\}$$

is dominated by the first term, so it is unchanged for sufficiently small δ. The total cost decreases by $(2\log_2(1-R-2L) - 2\log_2(1-R-2L-(W-2P)))\delta + O(\delta^2)$, contradicting the optimality of (P, L); note that the coefficient $2\log_2(1-R-2L) - 2\log_2(1-R-2L-(W-2P))$ is positive since $W - 2P > 0$.

Therefore $2L \leq (R/2)\log_2(R/2) - P\log_2 P - (R/2 - P)\log_2(R/2 - P)$, and $1 - R - 2L > 2W \geq 2(W - 2P)$ as claimed. □