# Optimizing Information Set Decoding Algorithms to Attack Cyclosymmetric MDPC Codes

Ray Perlner

National Institute of Standards and Technology,
Gaithersburg, Maryland, USA

ray.perlner@nist.gov

**Abstract.** The most important drawback to code-based cryptography has historically been its large key sizes. Recently, several promising approaches have been proposed to reduce keysizes. In particular, significant keysize reduction has been achieved by using structured, but non-algebraic codes, such as quasi-cyclic (QC) Moderate Density Parity Check (MDPC) codes. Biasi et al. propose further reducing the keysizes of code-based schemes using cyclosymmetric (CS) codes. Biasi et al. analyze the complexity of attacking their scheme using standard information-set-decoding attacks. However, the research presented here shows that information set decoding algorithms can be modified, by choosing the columns of the information set in a way that takes advantage of the added symmetry. The result is an attack that significantly reduces the security of the proposed CS-MDPC schemes to the point that they no longer offer an advantage in keysize over QC-MDPC schemes of the same security level.

**Key words:** information set decoding, code-based cryptography, moderate density parity check (MDPC) codes, cyclosymmetric

## 1 Introduction

The McEliece cryptosystem [1] is one of the oldest and most studied candidates for a postquantum cryptosystem. However, its keysizes, on the order of a million bits, are a major drawback. The most aggressive approaches to keysize reduction have focused on imposing structure on the public generator and parity check matrices such that they consist of cyclic [2] or dyadic [3] blocks, each of which can be represented using only the top row of the block.

However, these matrices have significant algebraic structure, and when the private code is itself an algebraic code, like the Goppa codes used in the original McEliece cryptosystem, such schemes tend to be open to algebraic attack[4]. A promising solution to this problem is to use nonalgebraic codes. In particular Misoczki et al. proposed [5] using moderate density parity check (MDPC) codes with quasicyclic structure (QC-MDPC).

A typical approach to attacking a scheme based on MDPC codes is to use information set decoding techniques to find low weight codewords in the dual code space (i.e. the row space of the public parity check matrix.) The concept of information set decoding originates with Prange [6]. Further optimizations were subsequently proposed by Lee and Brickell [7], Leon [8] and Stern [9].

Biasi et al. [10] attempt further keysize reduction by replacing blockwise cyclic structure with blockwise cyclosymmetric (CS) structure. The advantage of such matrices is that they can be represented by only half of the elements of their top rows. Indeed, a cyclosymmetric matrix consisting of smaller cyclosymmetric blocks can be represented using only a quarter of the elements in its top row, which would seem to provide significant opportunities for keysize reduction above and beyond what can be achieved using cyclic matrices. This further optimization was suggested by Biasi et al. in earlier versions of their paper[11],[12], but not in the published version, for reasons discussed in Section 4.

This paper demonstrates that information set decoding techniques can be improved by restricting the selection of information set columns to take advantage of CS symmetry. The complexity of the resulting attacks on a blockwise cyclosymmetric code is almost identical to the complexity of attacking a similar blockwise cyclic code with half the dimension, and half the row weight.

## 2 Cyclosymmetric Matrices

Ordinary cyclic matrices are those of the form:

$$A = \begin{bmatrix} a_0 & a_1 & \dots & a_{r-1} \\ a_{r-1} & a_0 & \dots & a_{r-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_0 \end{bmatrix}. \tag{1}$$

Each row is the right-cyclic rotation of the row above it. When their entries are elements of a field $\mathbb{F}$, cyclic matrices form a commutative ring under matrix multiplication and addition, isomorphic to the polynomial ring $\mathbb{F}[x]/(x^r - 1)$. (In most code-based-cryptography applications, including the scheme attacked in this paper, $\mathbb{F}$ is $\mathbb{F}_2$.)

Cyclosymmetric matrices are further restricted to be symmetric matrices, i.e. equal to their transpose. Using the commutativity of the ring of cyclic matrices we can show that the cyclosymmetric matrices are closed under multiplication and therefore form a subring of the cyclic matrices:

$$(AB)^T = B^T A^T = BA = AB. \tag{2}$$

A relevant fact about cyclosymmetric matrices is that $\lfloor \frac{r-1}{2} \rfloor$ pairs of entries in the top row of a cyclosymmetric matrix are constrained by symmetry to be equal:

$$\forall x | 1 \leq x < \frac{r-1}{2} : a_x = a_{r-x}. \tag{3}$$

## 3 MDPC cryptosystems

The scheme of Biasi et al. [10] modifies an earlier proposal by Misoczki et al. [5]. Both schemes are variants of the Niederreiter[13] cryptosystem : The public key, $H_{pub}$ is a $(n-k) \times n$ parity check matrix for a binary linear code, in systematic form —$[M|I]$. The plaintext, $m$, is encoded as an $n$-bit vector of Hamming weight at most $t$. The ciphertext is $H_{pub}m^T$. In the language of coding theory, the plaintext is the error vector, while the ciphertext is the syndrome. As in all variants of the Neiderreiter cryptosystem, the private key consists of trapdoor information that allows the owner to efficiently reconstruct the error vector $m$ from the syndrome $H_{pub}m^T$

In the case of MDPC cryptosystems, the private key is a low density parity check matrix $H$ sharing the same codespace as $H_{pub}$. The cryptographic scheme is described as using a moderate density parity check (MDPC) code, in contrast to the related low density parity check (LDPC) codes used for error correction in telecommunications applications. LDPC codes employ a significantly less dense parity check matrix and they correct more errors than the codes used in the proposed cryptographic scheme. The quasicyclic and cyclosymmetric variants of the MDPC encryption scheme construct the matrix $H$ from $n_0$ cyclic or cyclosymmetric blocks each with row weight $d_v$, but otherwise randomly chosen:

$$H = \left[H_0\ H_1\ \ldots\ H_{n_0-1}\right]. \tag{4}$$

Once a private parity check matrix is chosen as above, the public key is constructed from it as follows:

$$H_{pub} = H_{n_0-1}^{-1}H = \left[H_{n_0-1}^{-1}H_0 \mid H_{n_0-1}^{-1}H_1 \mid \ldots \mid H_{n_0-1}^{-1}H_{n_0-2} \mid I\right]. \tag{5}$$

## 4 Previous Attack on Cyclosymmetric Matrices

In their paper, Biasi et al. note that there is a more compact representation of the ring of cyclosymmetric matrices than that given in equation 1. For example, matrices of the form:

$$M(a,b,c,d) = \begin{bmatrix} a & b & c & d & c & b \\ b & a & b & c & d & c \\ c & b & a & b & c & d \\ d & c & b & a & b & c \\ c & d & c & b & a & b \\ b & c & d & c & b & a \end{bmatrix} \tag{6}$$

obey exactly the same multiplication rules as matrices of the form

$$M'(a,b,c,d) = \begin{bmatrix} a & 2b & 2c & d \\ b & a+c & b+d & c \\ c & b+d & a+c & b \\ d & 2c & 2b & a \end{bmatrix}. \tag{7}$$

This however does not completely break the scheme. While, this observation allows the attacker to reduce the dimension of the scheme being attacked by a factor of 2 for large matrices, it does so at the cost of reducing the sparsity (increasing the row weight) of the target private matrix by a factor of 2. This observation forced Biasi et al. to make their parameter choices less agressive, but it did not force them to abandon the possibility of keysize reduction through cyclosymmetric matrices altogether.

## 5 Modifying Information Set Decoding Techniques

The goal of the attack presented in this paper is to extract the private key $H$, from the public key $H_{pub}$. As is clear from equations 4 and 5, the rows of $H$ are linear combinations of the rows of $H_{pub}$. In particular, as will become relevant later in this section, $h = h_{n_0-1}H_{pub}$, where $h$ and $h_{n_0-1}$ represent the top rows of the matrices $H$ and $H_{n_0-1}$ respectively. The rows of $H$ are distinguished from other linear combinations of the rows of $H_{pub}$ in that they are sparse. As it happens, finding sparse linear combinations of the rows of a binary matrix is precisely the application for which classical information set decoding algorithms were invented.

All information set decoding algorithms follow the same basic script[1]:

1. Permute the columns of $H_{pub}$ :
$$H'_{pub} = H_{pub}P. \tag{8}$$
2. Check that the first $r$ columns of the new matrix, $H'_{pub}$, form an invertible matrix $A$. These columns are referred to as the "information set." If $A$ is not invertible go back to step 1.
3. Left-multiply by $A^{-1}$, resulting in a matrix of the form:
$$M = A^{-1}H'_{pub} = \left[I_r \mid Q\right]. \tag{9}$$
4. Search for low weight row-vectors among linear combinations involving small subsets of the rows of $M$. If none are found, go back to step 1. If a low weight vector $x' = vM$ is found, return $x = vMP^{-1}$.

Most optimizations to information set decoding algorithms, for example that of Stern [9], involve step 4. However, the special blockwise cyclosymmetric form of $H_{pub}$ allows us to make a much larger optimization, based on the choice of the permutation $P$ in step 1. To see how this works, we need to understand the significance of the row vector $v$ in step 4: In particular, since the first $r$ columns of $M$ form an identity matrix, the first $r$ bits of the candidate low weight row vector $x'$ are equal to $v$. Moreover:

[1] The variable names are chosen to reflect the scheme being attacked. For example the matrix being attacked is represented as a parity check matrix $H_{pub}$ rather than a generator matrix $G$, and its dimensions are given as $r \times n_0 r$ rather than $k \times n$

**Theorem 1** *When computed by an information set decoding algorithm as outlined by steps 1-4 above, $x'$ is the unique element of the rowspace of $H'_{pub}$ whose first $r$ bits equal $v$.*

*Proof.* Suppose there were another element of the rowspace of $H'_{pub}$, $yH'_{pub}$ whose first $r$ bits equalled $v$. Then, since $yH'_{pub}$ expands as:

$$yH'_{pub} = yAM = yA|yAQ. \tag{10}$$

We may rewrite our requirement as

$$yA = v. \tag{11}$$

Since A is invertible, this implies $y = vA^{-1}$ and therefore, $yH'_{pub} = vA^{-1}H'_{pub} = vM = x'$.

Thus, given the existence of a low weight vector $x$ in the rowspace of $H_{pub}$, $v$ represents a guess of all the bits of $x$ within the information set. Since the most probable value of a bit contained within a sparse vector is zero, the choice of $v$ with the highest probability of success is the guess which contains as many zeroes as possible. (Note that $v$ must contain at least one nonzero bit, since we're looking for a nontrivial solution.) As it happens, the best strategy involves checking multiple guesses of $v$ for each choice of $P$, since checking a guess is computationally cheaper than inverting a matrix, but the point remains that our probability of success relies on the probability that we will choose an information set, such that the restriction $v$ of $x$ to the information set is significantly sparser than $x$ itself.

This is where the choice of permutation helps us. We are much more likely to get $x$ to be oversparse on the information set, if the bits we are guessing are not independent. As it happens, the top row, $h$ of the private parity check matrix is a sparse vector, consisting of subvectors, $h_0 \dots h_{n_0-1}$, whose bits come in pairs obeying the relation given in equation 3. $x = h$ will then be the target of our attack. If we restrict the permutation $P$ to either leave both elements of such linked pairs outside of the information set, or to bring both elements in, then the probability of $h$ matching one of our oversparse guesses $v$ on the information set is significantly higher than it would be if $P$ were chosen randomly.

To give an example (based on the parameters given by Biasi at all for 128-bit security) if $n_0 = 3$, $r = 7232$, and the row/column weight, $d_v$, of the submatrices $H_0$, $H_1$, and $H_2$, is equal to 98, then for a random choice of $P$ the probability that $Truncate(r, hP)$ has weight 2 is $\frac{\binom{7232}{2}\binom{2\cdot 7232}{292}}{\binom{3\cdot 7232}{294}} = 2^{-160}$. However, for a choice of P restricted to bring mirrored pairs of bits into the information set together, the probability is $\frac{\binom{3616}{1}\binom{2\cdot 3616}{146}}{\binom{3\cdot 3616}{147}} = 2^{-80}$. Thus, a (rather poorly optimized) information set decoding algorithm, which tried all the values of $v$ with weight 2, would require $2^{160}$ matrix inversions on average to succeed if $P$ were chosen randomly. Our optimization brings the complexity down to $2^{80}$ matrix inversions, which, even accounting for the nontrivial complexity of the matrix inversion step, is already well below the claimed security level of the scheme.

## 6 Modified Stern Algorithm

In this section we present a variant of Stern's algorithm modified to find the top row, $h$ of the private parity check matrix of the CS-MDPC scheme of Biasi et al. The other rows of $H$ may then be trivially computed as rotations of $h$. The attacker is given $H_{pub}$ generated from $H$ as in equation 5. Both $H$ and $H_{pub}$ have dimensions $r \times n_0 r$, and consist of $r \times r$ cyclosymmetric blocks. $H$ has column weight $d_v$ and row weight $n_0 d_v$. The algorithm is parametrized by integers $p$ and $l$.

1. Permute the columns of $H_{pub}$ :

$$H'_{pub} = H_{pub} P \tag{12}$$

choosing $P$ with the restriction that cyclosymmetry forces:

$$(hP)_{2i} = (hP)_{2i+1} \ \ for\ i = 0 \ldots \lfloor r/2 \rfloor + l. \tag{13}$$

2. Check that the first $r$ columns of the new matrix, $H'_{pub}$, form an invertible matrix $A$. If $A$ is not invertible go back to step 1.

3. Left-Multiply by $A^{-1}$, resulting in a matrix of the form:

$$M = A^{-1} H'_{pub} = \left[ I_r \mid Q \right]. \tag{14}$$

4. Search for low-weight row-vectors among linear combinations involving small subsets of the rows of $M$. In particular these will involve $2p$ of the first $\frac{r}{2}$ rows and $2p$ of the remaining rows. The search will succeed if $hP$ has weight 2p on its first $\frac{r}{2}$ bits, weight 2p on the next $r/2$ bits, and weight 0 on the next $l$ bits.

   (a) Sum paired rows and compile in two equal length lists, i.e.:
   for $0 \leq i < \frac{r}{4}$

$$x_i = row_{2i}(M) + row_{2i+1}(M) \tag{15}$$

   and for $\frac{r}{4} \leq j < \frac{r}{2}$

$$y_i = row_{2j}(M) + row_{2j+1}(M) \tag{16}$$

   (b) compute all the sums of $p$ $x_i$s and all the sums of $p$ $y_i$s and check for collisions on bits $r \ldots r + 2l - 1$

$$bits_{r \ldots r+2l+1}(x_{i_1} + \ldots + x_{i_p}) = bits(r \ldots r + 2l + 1, y_{j_1} + \ldots + y_{j_p}) \tag{17}$$

   (c) When such a collision is found, check the total weight of the sum $w$ of the $2p$ colliding row vectors.

$$w = x_{i_1} + \ldots + x_{i_p} + y_{j_1} + \ldots + y_{j_p}. \tag{18}$$

   If the weight of any such $w$ is less than or equal to $n_0 d_v$ retrurn $wP$. Otherwise, go back to step 1.

## 7 Attack Complexity for Suggested Parameters

The major contributions to the overall complexity of each iteration of the modified Stern's algorithm above may be approximated as: $n_0 r^3$ for the matrix inversion (step 3), $2(p-1)n_0 r\binom{\frac{r}{4}}{p}$ for the construction of hash tables for collision search (step 4b), and $\frac{n_0 r\binom{\frac{r}{4}}{p}^2}{2^l}$ for testing candidate low-weight vectors, $w$ (step 4c). However, the units for these complexity figures are single-bit addition operations. Since legitimate parties do computations on the order of $n_0 r^2$ during both public and private-key operations, it is reasonable to divide this factor out leaving a per iteration complexity estimate of:

$$r + \frac{2(p-1)}{r}\binom{\frac{r}{4}}{p} + \frac{1}{2^l r}\binom{\frac{r}{4}}{p}^2. \tag{19}$$

The expected number of iterations is the inverse probability of success per iteration, which is:

$$\binom{\frac{n_0 r}{2}}{\frac{n_0 d_v}{2}}\binom{\frac{r}{4}}{p}^{-2}\binom{\frac{(n_0-1)r}{2}-l}{\frac{n_0 d_v}{2}-2p}^{-1}. \tag{20}$$

Note that the iteration count (equation 20) is identical to the iteration count of an unmodified Stern's algorithm applied to a code with $r' = \frac{r}{2}$ and $d'_v = \frac{d_v}{2}$, and the per iteration cost (equation 19) is identical up to polynomial factors in $r/r' = 2$ (The discrepancy is due to the fact that linear algebra operations are being performed on a larger matrix.) Thus, our attack may be thought of as reducing the security of a cyclosymmetric MDPC scheme with block dimension $r$ and private row density $\frac{d_v}{r}$ to that of a corresponding cyclic scheme which with dimension $\frac{r}{2}$ and the same private row density.

Table 1 gives the results of our attack when applied to the parameters suggested by Biasi et al. For all parameter choices, the security level allowed by this attack is significantly lower than the claimed security level.

| Claimed Security (bits) | $n_0$ | $r$ | $d_v$ | Attack Complexity (bits) | $p$ | $l$ |
|---|---|---|---|---|---|---|
| 80 | 3 | 3072 | 53 | 46 | 2 | 20 |
| 112 | 3 | 5376 | 75 | 63 | 2 | 20 |
| 128 | 3 | 7232 | 97 | 81 | 2 | 22 |
| 160 | 3 | 19200 | 109 | 93 | 2 | 25 |

**Table 1.** Claimed security levels and the results of the modified Stern's algorithm attack for parameters given in [10]

As our attack brings the security of Biasi et al.'s proposed 128-bit parameters down to nearly exactly 80 bits of security, it is informative to compare these parameters to the 80-bit security parameters of Misoczki et al.'s QC-MDPC

scheme. Here we find that there is no longer any advantage to the cyclosymmetric scheme, either in public key size or cryptogram size:

| | CS-MDPC [10] | QC-MDPC [5] |
|---|---|---|
| Public Key Length | 7232 | 4801 |
| Cryptogram Size | 21696 | 9602 |

**Table 2.** Comparison of proposed CS-MDPC and QC-MDPC parameters at 80 bits of security given this paper's attack.

## 8 Conclusion

While the idea of using cyclosymmetric codes to reduce keysize beyond what is possible with blockwise cyclic codes seemed promising, the added structure appears to be as useful to the attacker as to the legitimate parties. In particular, information set decoding algorithms can be modified to take full advantage of the knowledge that the rows of the private parity check matrix of such a scheme are structured. It may be the case that cyclic MDPC codes are as far as we can go in keysize reduction for code-based cryptography.

## References

1. McEliece, R.J.: A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report **44** (1978) 114–116
2. Berger, T., Cayrel, P.L., Gaborit, P., Otmani, A.: Reducing key length of the mceliece cryptosystem. In Preneel, B., ed.: Progress in Cryptology AFRICACRYPT 2009. Volume 5580 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 77–97
3. Misoczki, R., Barreto, P.: Compact mceliece keys from goppa codes. In Jacobson, MichaelJ., J., Rijmen, V., Safavi-Naini, R., eds.: Selected Areas in Cryptography. Volume 5867 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 376–392
4. Faugre, J.C., Otmani, A., Perret, L., Tillich, J.P.: Algebraic cryptanalysis of mceliece variants with compact keys. In Gilbert, H., ed.: Advances in Cryptology EUROCRYPT 2010. Volume 6110 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 279–298
5. Misoczki, R., Tillich, J.P., Sendrier, N., Barreto, P.S.L.M.: Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. Cryptology ePrint Archive, Report 2012/409 (2012) `http://eprint.iacr.org/`.
6. Prange, E.: The use of information sets in decoding cyclic codes. Information Theory, IRE Transactions on **8** (1962) 5–9
7. Lee, P., Brickell, E.: An observation on the security of mcelieces public-key cryptosystem. In Barstow, D., Brauer, W., Brinch Hansen, P., Gries, D., Luckham, D., Moler, C., Pnueli, A., Seegmller, G., Stoer, J., Wirth, N., Gnther, C., eds.: Advances in Cryptology EUROCRYPT 88. Volume 330 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1988) 275–280

8. Leon, J.: A probabilistic algorithm for computing minimum weights of large error-correcting codes. Information Theory, IEEE Transactions on **34** (1988) 1354–1359
9. Stern, J.: A method for finding codewords of small weight. In Cohen, G., Wolfmann, J., eds.: Coding Theory and Applications. Volume 388 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1989) 106–113
10. Biasi, F., Barreto, P., Misoczki, R., Ruggiero, W.: Scaling efficient code-based cryptosystems for embedded platforms. Journal of Cryptographic Engineering (2014) 1–12
11. Barreto, P.: Can code-based keys and cryptograms get smaller than their rsa counterparts? (2012)
12. Biasi, F.P., Barreto, P.S., Misoczki, R., Ruggiero, W.V.: Scaling efficient code-based cryptosystems for embedded platforms. arXiv preprint arXiv:1212.4317 (2012)
13. Neiderreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory. Problemy Upravlenija i Teorii Informacii (**15**) 159–166