Broadcast Attacks against Code-Based Schemes

Robert Niebuhr[1] and Pierre-Louis Cayrel[2]

[1] Technische Universität Darmstadt
Fachbereich Informatik
Kryptographie und Computeralgebra,
Hochschulstraße 10
64289 Darmstadt
Germany
rniebuhr@cdc.informatik.tu-darmstadt.de
[2] Laboratoire Hubert Curien, UMR CNRS 5516,
Bâtiment F 18 rue du Professeur Benoît Lauras
42000 Saint-Etienne
France
pierre.louis.cayrel@univ-st-etienne.fr

Abstract. Code-based cryptographic schemes are promising candidates for post-quantum cryptography since they are fast, require only basic arithmetic, and because their security is well understood. While there is strong evidence that cryptosystems like McEliece and Niederreiter are secure, they have certain weaknesses when used without semantic conversions. An example is a broadcast scenario where the same message is send to different users, encrypted with the respective keys.

In this paper, we show how an attacker can use these messages to mount a broadcast attack, which allows to break the Niederreiter and the HyMES cryptosystem using only a small number of messages. While many code-based cryptosystems use certain classes of codes, e.g. binary Goppa codes, our attack is completely independent from this choice and solves the underlying problem directly. Since the number of required messages is very small and since the attack is also possible if related, not identical messages are sent, this has many implications on practical cryptosystem implementations. We discuss possible countermeasures, and provide a CCA2-secure version of the Niederreiter cryptosystem using the Kobara-Imai conversion.

Keywords: Broadcast Attack, Niederreiter, McEliece, codes, post quantum, cryptography.

Introduction

In 1988, J. Håstad [9] presented an attack against public key cryptosystems. This attack was originally aimed at the RSA cryptosystem, when a single message is sent to different recipients using their respective public keys. Håstad showed how to recover the message in this broadcast scenario. While this result is known for

F. Armknecht and S. Lucks (Eds.): WEWoRC 2011, LNCS 7242, pp. 1–17, 2012.

a long time, this type of attack has not been considered for cryptosystems based on error-correcting codes.

Our Contributions

In the following, we show how and under what conditions an attacker can mount a broadcast attack against two code-based encryption schemes: the Niederreiter and the HyMES cryptosystem. Our attack allows to recover the secret message. We show that if the public keys corresponding to the intercepted messages are independent from each other, we expect to require no more than N_r recipients to run the attack:

$$N_r := \left\lceil \frac{n+2}{r} \right\rceil,$$

where n and $r = n-k$ denote the code parameters of the users' secret keys. That means that with near certainty, $N \geq N_r$ recipients suffice to run the attack. Table 1 shows the number of required recipients for selected parameter sets taken from Bernstein et al. [2] and Biswas [5]. In most cases, a very small number of identical (or similar) message is sufficient to completely break the schemes. We achieve this by combining the intercepted information into a large set of linear equations until the system has full rank and can be solved.

In addition to that, we treat the cases when the attacker receives less messages than required for this attack, and when the cleartexts are related, instead of identical. In the former case, the attack complexity is higher compared with the broadcast attack before, but lower than a generic attack: after setting up the linear equation system, the attacker runs an ISD attack, the complexity of which is smaller the more messages are intercepted. In the latter case, more messages are required to perform the broadcast attack: $N'_r := \lceil \frac{n+2}{r-(u+2)} \rceil$, where u denotes the number of bits where not all messages are identical to each other.

While many code-based cryptosystems use certain classes of codes, e.g. binary Goppa codes, our attack is completely independent of this choice and solves the underlying problem directly. That means, no matter what class of codes is used, the attack complexity cannot be greater than our results; it might be possible, though, to achieve even better results against certain classes by exploiting the structure of the code. To illustrate our results, we apply our broadcast attack implementation against the Niederreiter cryptosystem in the FlexiProvider package [6], recovering the message in only a few seconds to minutes (see Table 2 on page 15). We conclude this section with a discussion on possible countermeasures and provide a CCA2-secure version of the Niederreiter cryptosystem using the Kobara-Imai conversion.

Related Work

Our attack is related to the one by Plantard and Susilo [15] who studied broadcast attacks against lattice-based schemes. Our analysis, however, is more

thorough since we prove an explicit bound for the expected number of recipients that is required to run our attack. We also cover the case where too few messages are intercepted by an attacker and analyze the situation where the broadcasted messages are not identical but similar to each other. Finally, we discuss in detail the use of semantic conversions to protect against broadcast attacks. An implementation of our attack successfully demonstrated its efficiency, even though it is in Java and not at all optimized for speed.

A recent paper [18] by Pan and Deng presents a broadcast attack against the NTRU cryptosystem. The authors use a Learning with Errors (LWE) algorithm by Ding for their attack.

Organization of the Paper

In Section 1 we begin with a review on code-based encryption schemes. The next section covers our broadcast attack, including the two variants of insufficient number of messages and related-messages. We discuss possible countermeasures in Section 3. In the subsequent Section 4 we present our implementation and provide numerical results. We conclude in Section 5.

1 Code-Based Encryption Schemes

Code-based cryptographic schemes are based on the difficulty of two hard problems: Code distinguishing and syndrome decoding. In this paper, we will focus on the latter, which is defined as follows:

Problem 1 (Syndrome-decoding problem). Given a matrix H and a vector c, both over $\mathbb{F}_q$, and a non-negative integer t; find a vector $x \in \mathbb{F}_q^n$ of (Hamming) weight t such that $Hx^T = c^T$.

Among these are the McEliece and Niederreiter cryptosystems. They are both encryption schemes, and the latter is the basis for the CFS signature scheme. Another cryptosystem we analyze in this paper is HyMES (Hybrid McEliece Encryption Scheme)[1], which uses techniques from both schemes above: It encrypts similar to the McEliece scheme, but encodes part of the messages in the error vector, similar to the Niederreiter scheme.

In this section, we will briefly describe these three cryptosystems. In the following, let G be a $k \times n$ generator matrix for an $(n, k = n - r, t)$ Goppa code $\mathcal{C}$, H be a corresponding $r \times n$ parity check matrix, and c a vector of length r. Let the message m be a vector of length k, and φ a bijective function mapping an integer to a word of length n and weight t. All matrices and vectors are defined over a finite field $\mathbb{F}_q$, where q is a prime power.

[1] `http://www-rocq.inria.fr/secret/CBCrypto/index.php?pg=hymes`

Notation. In our algorithms, we use the following notation:

$\mathrm{wt}(v)$	: The (Hamming) weight of vector v
$\mathrm{PRG}(x)$	: Cryptographically secure pseudo random number generator
l	: $\lfloor \log_q \binom{n}{t} \rfloor$
$\varphi()$	: Bijective function mapping an integer in $\mathbb{Z}_{q^l} = \mathbb{Z}/q^l\mathbb{Z}$ to a word of length n and weight t. We apply φ to vectors in $\mathbb{F}_q^k$ by enumerating these vectors first, and then apply φ
$\mathrm{MSB}_x(v)$	: The left x bits of v
$\mathrm{LSB}_x(v)$	: The right x bits of v
$\mathrm{len}(v)$	: Length of vector v
$h()$	: Cryptographic secure hash function to a word of length l

Additionally, we write $A = (B|C)$ and $A' = \langle B, C \rangle$ to denote the horizontal and vertical concatenation of B and C, respectively, where these can be vectors or matrices. For a matrix A and $J \subseteq \{1, 2, \ldots n\}$, $A_{.J}$ denotes the submatrix of A consisting of those columns indexed by J.

1.1 McEliece

The McEliece public-key encryption scheme was presented by R. McEliece in 1978 [12]. The original scheme uses binary Goppa codes, for which it remains unbroken (with suitable parameters), but the scheme can be used with any class of codes for which an efficient decoding algorithm is known.

1.2 Niederreiter

In 1986, H. Niederreiter proposed a cryptosystem [14] which can be seen as dual to the McEliece scheme. It uses the parity check matrix of a (usually binary Goppa) code to compute the syndrome of the message, which serves as the ciphertext. Even though the Niederreiter cryptosystem has been proven equally secure as the McEliece system [11], it is threatened by broadcast attacks.
Since the underlying Goppa code can only correct a certain number $t < n$ of errors, the Niederreiter scheme uses a function φ which maps the message to a word of weight t, which is then encrypted.

1.3 HyMES

The HyMES hybrid McEliece cryptosystem developed by N. Sendrier and B. Biswas increases the efficiency of the McEliece scheme by encoding part of the message into the error vector. While in the usual scenario this scheme is as secure as the original McEliece scheme, we will show that it is vulnerable to a broadcast attack.

The HyMES scheme works as follows: The message m is split into two parts $m = (m_1|m_2)$. The first part m_1 corresponds to the message in the original

Algorithm 1. The McEliece cryptosystem

Notation for Algorithm 1:

G : A $k \times n$ generator matrix
P : An $n \times n$ random permutation matrix
S : A $k \times k$ invertible matrix
$\mathcal{D}_G$: A decoding algorithm for the underlying (n, k, t) code $\mathcal{C}$

Encryption $\mathbf{Enc}^{\mathbf{McEliece}}$
INPUT: Message $m \in \mathbb{F}_q^k$ and random seed $r \in \{0,1\}^*$
OUTPUT: Ciphertext $c \in \mathbb{F}_q^n$

$\widehat{G} \leftarrow SGP$
$e \leftarrow \mathrm{PRG}(r)$, such that $\mathrm{wt}(e) = t$
$c \leftarrow m\widehat{G} + e$

Return c

Decryption $\mathbf{Dec}^{\mathbf{McEliece}}$
INPUT: Ciphertext c
OUTPUT: Message m

$\widehat{c} \leftarrow cP^{-1} = mSG + eP^{-1}$
$mSG \leftarrow \mathcal{D}_G(\widehat{c})$
▷ Let $J \subseteq \{1, \ldots, n\}$ be a set such that $G_{\cdot J}$ is invertible
$m \leftarrow mSG \cdot G_{\cdot J}^{-1} \cdot S^{-1}$

Return m

Algorithm 2. The Niederreiter cryptosystem

Notation for Algorithm 2:

H : A $r \times n$ parity check matrix
$\mathcal{D}_H$: A decoding algorithm for the underlying $(n, k = n - r, t)$ code $\mathcal{C}$

Encryption $\mathbf{Enc}^{\mathbf{N}}$
INPUT: Message $m \in \mathbb{F}_q^l$
OUTPUT: Ciphertext $c \in \mathbb{F}_q^r$

$c \leftarrow H \cdot \varphi(m)^T$

Return c

Decryption $\mathbf{Dec}^{\mathbf{N}}$
INPUT: Ciphertext c
OUTPUT: Message m

$m \leftarrow \varphi^{-1}(\mathcal{D}_H(c))$

Return m

McEliece scheme, while the second part is encoded into a word of weight t and serves as the error vector $e = \varphi(m_2)$. There are many possible encoding functions φ, e.g. enumerative encoding or encoding into regular words, but the choice of φ is not relevant in our context.

Algorithm 3. The HyMES cryptosystem

Notation for Algorithm 3:
- G : A $k \times n$ generator matrix
- P : An $n \times n$ random permutation matrix
- S : A $k \times k$ invertible matrix
- $\mathcal{D}_G$: A decoding algorithm for the underlying (n, k, t) code $\mathcal{C}$

Encryption $\mathbf{Enc^{HyMES}}$
INPUT: Message $m \in \mathbb{F}_q^{k+l}$
OUTPUT: Ciphertext $c \in \mathbb{F}_q^n$

$\widehat{G} \leftarrow SGP$
$m_1 \leftarrow \mathrm{MSB}_k(m)$
$m_2 \leftarrow \mathrm{LSB}_l(m)$
$c \leftarrow m_1\widehat{G} + \varphi(m_2)$

Return c

Decryption $\mathbf{Dec^{HyMES}}$
INPUT: Ciphertext c
OUTPUT: Message m

$\widehat{c} \leftarrow cP^{-1} = m_1SG + \varphi(m_2)P^{-1}$
$mSG \leftarrow \mathcal{D}_G(\widehat{c})$
▷ Let $J \subseteq \{1, \ldots, n\}$ be a set such that $G_{.J}$ is invertible
$m_1 \leftarrow mSG \cdot G_{.J}^{-1} \cdot S^{-1}$
$m_2 \leftarrow \varphi^{-1}(c - m_1\widehat{G})$

Return $(m_1|m_2)$

2 Broadcast Attacks against Niederreiter/HyMES

In this section, we will show how to mount a broadcast attack against the Niederreiter and HyMES schemes. The problem solved by a broadcast attack is the following:

Problem 2 (Broadcast attack). Given N ciphertexts c_i of the same message m, encrypted using N corresponding public keys G_i (HyMES) or H_i (Niederreiter), find m.

Both the McEliece and the Niederreiter cryptosystem rely on the hardness of the decoding prob- lem, i.e. finding a codeword in a certain distance to a given

word. The main difference is that in McEliece the cleartext determines the codeword, and the error vector is random, while Niederreiter works essentially vice versa. This difference results in a weakness of the McEliece cryptosystem, the malleability of its ciphertexts: Adding rows of the public key to a ciphertext results in a new valid ciphertext.

Another implication is McEliece's weakness to message-resend and related-message attacks. Two ciphertexts of messages m_1 and m_2, where the messages have a known relation (or are identical), allow an attacker to easily find at least k error-free positions, which allows efficient guessing of error bits. See [2] for more details. The Niederreiter cryptosystem, however, is not vulnerable to these attacks. It is interesting to note that, facing a broadcast attack, the situation is reversed.

2.1 Attacking Niederreiter

Niederreiter ciphertexts are generated by computing $c_i^T = H_i\varphi(m)^T$. Attempting to solve this equation for m can be done by solving the corresponding linear equation system with n variables and r equations. Since $\varphi(m)$ is identical in all computations, we can (vertically) append the matrices $H = \langle H_1, \ldots, H_N\rangle$ and the syndromes $c = (c_1|\ldots|c_N)$. The number of variables stays constant, whereas the number of equations increases. Some of the new equations might be combinations of old ones, so the new number of equations can be smaller than Nr. In Section 2.3, we will show that the number of redundant equations is very small. Since usually $n \approx 2r$, we need very few c_i to increase the rank of H to n, at which point we can solve the system. We will compute the expected number of messages required to break the system in Section 2.3.

Algorithm 4. Broadcast attack against the Niederreiter cryptosystem

INPUT: Parity check matrices $H_i \in \mathbb{F}_q^{r\times n}$ and corresponding ciphertexts $c_i \in \mathbb{F}_q^r$ for $i \in \mathbb{Z}_N$, and finite field $\mathbb{F}_q$
OUTPUT: Message $m \in \mathbb{F}_q^n$

$H \leftarrow \langle H_1, \ldots, H_N\rangle$
$c \leftarrow (c_1|\ldots|c_N)$

Solve the linear equation system $Hm^T = c^T$ over $\mathbb{F}_q$

Return m

Remark 1. There is a different way to describe our attack, seen from another perspective: By adding c_i as the $(n+1)$-th column of H_i (for all i), we add $(\varphi(m)|(q-1))$ as a codeword to every code, since

$$(H_i|c_i)\cdot(\varphi(m)|(q-1))^T = H_i\cdot\varphi(m)^T + (q-1)c_i = qc_i = 0.$$

The message can be found by intersecting these new codes. There are several implementations to compute the intersection of codes, e.g. in Magma. However, we have chosen the approach above since it allows easier understanding of the required number of recipients.

2.2 Attacking HyMES

While the HyMES implementation does provide methods for padding, a technique used to prevent attacks that exploit relations between cleartexts and/or ciphertexts, this method seems to be a placeholder, since it does not add any security.

Similar to our description of the scheme above, in the broadcast scenario we have

$$c_i = m_1 G_i + \varphi(m_2)$$

as the ciphertexts, where $m = (m_1|m_2)$ and $i = 1 \dots N$. Attacking this scheme can be done by finding $\varphi(m_2)$, since this allows to find m_1 by simple linear algebra. We can reduce this problem to the Niederreiter case:

First, the attacker uses the matrices G_i to compute the respective parity check matrices H_i. One way to do this is to compute the *standard form* of G_i, $G_i' = UG_iQ = (I_k|R)$, where I_k is the identity matrix of size k. Then $H_i' = (-R^T|I_{n-k})$ and $H_i = H_i'Q^{-1}$.

Then the attacker computes the syndromes $s_i = H_i \cdot c_i^T$. Since

$$s_i = H_i(m_1 G_i + \varphi(m_2))^T = H_i \cdot \varphi(m_2)^T,$$

we have reduced the problem to the Niederreiter case above.

Algorithm 5. Broadcast attack against the HyMES cryptosystem

INPUT: Generator matrices G_i and corresponding ciphertexts c_i, for $i \in \mathbb{Z}_N$, and finite field $\mathbb{F}_q$
OUTPUT: Message $m \in \mathbb{F}_q^n$

$\forall i \in \mathbb{Z}_n$ perform the following computations
    Find U_i and Q_i such that $G_i' = U_i G_i Q_i = (I_k|R_i)$
    $H_i' \leftarrow (-R^T|I_{n-k})$
    $H_i \leftarrow H_i' Q_i^{-1}$
    $s_i \leftarrow H_i \cdot c_i^T$

$H \leftarrow \langle H_1, \dots, H_N \rangle$
$c \leftarrow (c_1| \dots |c_N)$

Solve the linear equation system $Hm^T = c^T$ over $\mathbb{F}_q$

Return m

Remark 2. As we noted above, the McEliece cryptosystem does not show a vulnerability to broadcast attacks as Niederreiter/HyMES do. We can sum the information contained in the ciphertexts and public keys into a single equation by (horizontal) concatenation: $G = (G_1|\dots|G_N)$, $c = (c_1|\dots|c_N)$, $e = (e_1|\dots|e_N)$, and writing $c = mG + e$.

This code has length nN, dimension k and minimum distance dN, the weight of e is tN. While this concatenated code is somewhat weaker than the original codes, this can be compensated by larger parameters, and it shows no structural weakness like in the Niederreiter case. Figure 1 shows the work factor to break the concatenated code using an Information Set Decoding (ISD) based attack.

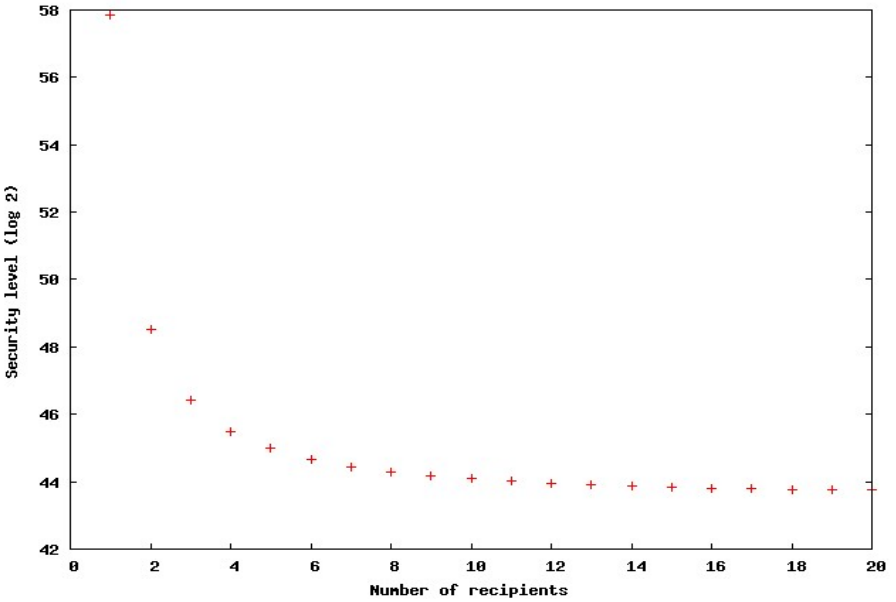

Fig. 1. Work factor to perform an ISD attack against the McEliece cryptosystem with parameters $(n, k, t) = (1024, 524, 50)$ using an increasing number of broadcasted messages

2.3 Expected Number of Recipients Required to Break the Niederreiter/HyMES Scheme

In order to estimate the number of recipients N_r and thus encrypted messages we need to recover the message encrypted by the above Niederreiter/HyMES

schemes, we first assume that the parity check matrices H_i are actually random matrices of full rank (even though it has been shown recently in [7] that we can distinguish random codes from binary Goppa codes for certain parameters, e.g. CFS parameters). We will do the analysis using the Niederreiter scheme, since we reduced the HyMES problem to this one.

We start by estimating the probability that a random vector x is linearly independent from all vectors in A, where A is a given set of r linearly independent vectors over $\mathbb{F}_q$.

A vector is linearly independent from a set of vectors if it cannot be expressed by a linear combination of these vectors. There are q^r (different) linear combinations of vectors in A, and the whole space has dimension q^n. Thus, the probability that x is linearly independent from A is

$$\mathfrak{P} = 1 - q^{r-n} = \frac{q^n - q^r}{q^n}. \tag{1}$$

Therefore, when we start from the system of linear equations

$$H_1 \cdot \varphi(m)^T = c_1^T$$

and add the first row of H_2 to H_1, with probability $\mathfrak{P} = 1 - q^{r-n}$ we add a new equation to the system. Hence, if we assume the vectors in H_2 to be independent from each other, we expect to add $\mathfrak{P}^{-1}$ rows to get one new equation. The fact that the vectors in H_2 are not independent from each other does in fact slightly increases the chance to find a new equation: subsequent vectors in H_2 are guaranteed to be linearly independent to the previous ones already considered, so a small subset of undesired vectors is excluded.

Thus, in order to increase the number of linearly independent equations to n, which allows us to solve the system, we need to add

$$T = \sum_{i=r}^{n-1} \frac{q^n}{q^n - q^i}$$

rows on average.

The codes used in the two cryptosystems are not random, but Goppa codes. Since every non-zero vector in $\mathbb{F}_q^n$ has the same probability to be contained in a Goppa code chosen uniformly at random, the probability $\mathfrak{P}$ and thus the subsequent arguments above remain valid. For example, for the Niederreiter parameters $[n, k] = [1024, 644]$, we need to add 646 rows, which corresponds to 3 recipients.

The expected number D of linearly dependent rows encountered when setting up the system is nearly constant: We add T rows in order to be able to solve the system, out of which $(n - r)$ are not redundant (they complement the initial r rows to a linearly independent set of n rows), and hence

$$\begin{aligned} D &= \sum_{i=r}^{n-1} \frac{q^n}{q^n - q^i} - (n-r) \\ &= \sum_{i=r}^{n-1} \left(\frac{q^n}{q^n - q^i} - 1 \right) \\ &= \sum_{i=r}^{n-1} \frac{q^i}{q^n - q^i} \\ &= \sum_{i=1}^{n-r} \frac{1}{q^i - 1} < 1.7. \end{aligned}$$

The last sum converges quickly, and it decreases with increasing q:

$$\sum_{i=1}^{n-r} \frac{1}{q^i - 1} \leq \sum_{i=1}^{\infty} \frac{1}{2^i - 1} = \sum_{i=1}^{4} \frac{1}{2^i - 1} + \sum_{i=5}^{\infty} \frac{1}{2^i - 1} \leq \sum_{i=1}^{4} \frac{1}{2^i - 1} + \sum_{i=4}^{\infty} \frac{1}{2^i} < 1.7$$

Therefore, we have

$$N_r = \left\lceil \frac{n+2}{r} \right\rceil.$$

Table 1 shows the number of required recipients for selected parameter sets taken from Bernstein et al. [2,3,4] and Biswas [5]. The first column shows the cryptosystem for which the parameters were developed. The number of required recipients, however, does not depend on whether the parameters are used with the Niederreiter or the HyMES cryptosystem.

Table 1. Number of required recipients for Niederreiter and HyMES parameter sets

Cryptosystem	n	k	q	Number of recipients
Niederreiter	2048	1696	2	6
	2048	1608	2	5
	4096	3832	2	16
	4096	3556	2	8
HyMES	1024	524	2	3
	2048	1608	2	5
	2048	1696	2	6
	4096	3604	2	9
	8192	7815	2	22
Niederreiter	3009	2325	2	5
	1931	1491	5	5
	1608	1224	7	5
	1696	1312	9	5
McEliece	1616	1253	2	5
	2928	2244	2	5
	6544	5010	2	5

2.4 Performing a Broadcast Attack When $N < N_r$

When an attacker receives less than N_r broadcast messages, the resulting system $H \cdot x^T = c^T$ is under-determined. It can be used nonetheless to mount an ISD attack. There are different ISD-like attacks, but the basic steps are as follows:

1. Choose a random $n \times n$ permutation matrix Q and compute $H' = QH$.
2. Perform Gaussian elimination on H' and s to get $(I_K|R) = s'$, where I_K is the identity matrix of size K, and $n - K$ is the number of rows of H.
3. Search for $p \leq t$ columns of R such that their sum S has Hamming distance $t - p$ to the syndrome s.
4. The non-zero entries of $S - s$ locate the remaining $t - p$ entries of $\varphi(m)$.

The work factor of this attack can be computed using the formulae in [13]. Figure 2 shows the corresponding attack complexity.

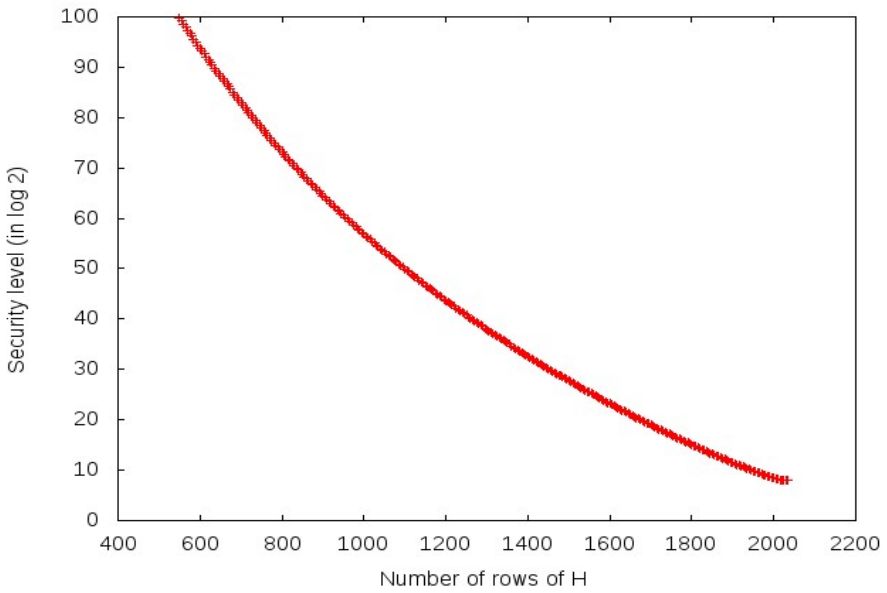

Fig. 2. Work factor for a broadcast attack against McEliece with parameters $(n, k, t) = (1024, 524, 50)$ using ISD when $N < N_r$

This result is supported by [17], where N. Sendrier points out that ISD-based algorithms have an approximate complexity of

$$C(n, R) = a(n) \cdot 2^{-t \cdot \log_2(1-R)},$$

where $R = k/n$ is the information rate and $a(n)$ a polynomial in n. Increasing the number of rows in the parity check matrix decreases R, so $C(n, R)$ decreases exponentially.

2.5 Related-Message Broadcast Attack

In the previous sections, the message m sent to all recipients has been identical. In the following, we show how a broadcast attack can be performed if the messages are *not* identical, but *related*. More concretely, let $M = \{m_i : i = 1..N\}$ and we define the following property:

Definition 1. *A set M of messages $m_i \in \mathbb{F}_q^n$ is called b-related if there are exactly $b \leq n$ coordinates such that all messages are identical on these coordinates. We denote this property by*

$$\rho(M) = b.$$

Since the Niederreiter and HyMES cryptosystems do not encrypt $m_i \in M$ directly, but $\varphi(m_i)$ instead, the choice of the encoding function φ will influence $\rho(\varphi(M))$. To keep our analysis independent of the choice of φ, we will assume that

$$\rho(\varphi(M)) = b,$$

where $n-b$ is small (this is the case, for instance, if φ is a regular words encoding).

Solving the Linear Equation System. For simpler notation, let the messages in M be identical on the left b bits, and (potentially) different on the rightmost $u := n - b$ bits. Since the messages are not identical, the parity check matrices cannot be used directly to form the final system of equations.

Let

$$H_i = (H_i^1 | H_i^2), \quad i = 1..N,$$

where H_i^1 contains the leftmost b columns of H_i.

For 2 recipients, we have the following system of equations

$$\begin{pmatrix} H_1^1 & H_1^2 & 0 \\ H_2^1 & 0 & H_2^2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix},$$

and similarly for N recipients.

A solution $(e_0|e_1|\ldots|e_N)$ (where e_0 has length b, and the other blocks have size $n - b$) yields solutions to the original problem with $m_i = (e_0|e_i)$.

In contrast to the identical-message broadcast attack above, every additional recipient adds equations as well as unknowns to the system. The system will eventually be solvable if (and only if) the number of new equations is greater than the number of new variables. Since we expect a total of at most 2 linearly dependent rows for the system, it is sufficient if $r > n - b + 2 = u + 2$.

The number of recipients N_r' required to solve the system of a related-message broadcast attack is

$$N_r' = \left\lceil \frac{n+2}{r-(u+2)} \right\rceil.$$

3 Countermeasures

Our broadcast attack exploits the fact that the received ciphertexts are related since they correspond to the same message m. A similar fact is used in other types of attack like message-resend, related message, chosen ciphertext etc. Therefore, broadcast attacks can be prevented by using one of the CCA2 conversions that have been proposed for these other types of attacks.

3.1 Unsuitable Conversions

Padding schemes like the well-known Optimal Asymmetric Encryption Padding (OAEP) by Bellare and Rogaway [1] are unsuitable for the McEliece/Niederreiter cryptosystems since they do not prevent reaction attacks: By randomly flipping bits and observing the reaction of the receiver, an attacker can recover the cleartext, and apply the conversion to reveal the message.

There are (at least) two generic conversion, proposed by Pointcheval [16] and by Fujisaki and Okamoto [8], that work with the McEliece and Niederreiter cryptosystems. However, they have the disadvantage of adding a large amount of redundancy to the ciphertexts. In both cases, the general idea is the following: Instead of encrypting the message with the (asymmetric) cryptosystem, say McEliece, it is encrypted with a symmetric system, and the corresponding key is encrypted with McEliece. Both outputs are appended to form the ciphertext. Since the output block size of McEliece and Niederreiter is large, a lot of redundancy is thereby added, which decreases the efficiency.

3.2 Kobara-Imai Conversion

More suitable is the conversion by Kobara and Imai [10]. This conversion was proposed for the McEliece cryptosystem, and for large messages it manages to reduce the redundancy of the ciphertext even below that of the unconverted cryptosystem. This conversion can also be applied to the Niederreiter cryptosystem. It can not be applied to the HyMES cryptosystem, since it uses a similar technique to encode part of the message into the error vector, hence applying the Kobara-Imai-conversion to the McEliece cryptosystem will achieve a similar efficiency improvement as the HyMES scheme. For the sake of completeness, we will briefly describe this conversion here. A more detailed description can be found in [2]. The resulting cryptosystem is a CCA2-secure variant of Niederreiter, which allows to implement secure and efficient cryptographic applications.

This conversion consists of two modifications. Firstly, it introduces randomness into the message, thereby rendering the output indistinguishable from a random ciphertext. This prevents attacks that rely on the relation of ciphertexts and/or cleartexts, e.g. message-resend, related-message, or broadcast attacks. Secondly, both the message vector m as well as the error vector e are computed from the message. This prevents reaction attacks, since a modified error vector results in a different cleartext, which can be detected by the honest user.

The pseudo code of the conversion can be found in the appendix.

4 Implementation

We have implemented the broadcast attack described above in Java. Target was the Niederreiter cryptosystem in the FlexiProvider package [6]. Table 2 shows the runtime for different parameters on an Intel i5-2500 CPU using one core. Note that our attack is not tweaked for performance, so we expect that this time can be improved further.

Table 2. Runtime of our algorithm for different parameters sets for the Niederreiter encryption scheme

Parameters (n, k, t)	Security level against ISD	Number of recipients	Runtime in sec
$(1024, 764, 26)$	57	4	6
$(1024, 524, 50)$	58	3	6
$(2048, 948, 100)$	97	2	35
$(2048, 1498, 50)$	101	4	50
$(4096, 3136, 80)$	174	5	460
$(4096, 2896, 100)$	184	4	430
$(4096, 2716, 115)$	188	3	352

Note that the runtimes increase cubic in n. This is expected, since the main work of the attack is to solve a system of linear equations, the complexity of which is in $\mathcal{O}(n^3)$.

5 Conclusion and Outlook

In this paper we have shown how a broadcast attack can be mounted against the Niederreiter and HyMES cryptosystem. We have calculated the number of recipients that are required in order to run our attack. Even though this number is usually very small, it is possible that an attacker intercepts only a smaller number of messages. We have shown that is it still possible to use this information to run a broadcast attack using Information Set Decoding, and the complexity of this attack decreases exponentially with the number of messages intercepted.

Our results have been tested experimentally, and our Java implementation was able to recover the secret message in < 20 seconds. The tests were run on an Intel Core i5 3.3 GHz CPU (one core), using the Niederreiter parameters $(n, t) = (2048, 50)$.

Finally, we showed that this type of attack can be prevented by applying a conversion on the message, e.g. Kobara-Imai's γ conversion.

As further work we propose to analyze if a structured code, e.g. a quasi-cyclic or quasi-dyadic code, is more vulnerable to this attack, resulting in a smaller number of required messages.

References

1. Bellare, M., Rogaway, P.: Optimal Asymmetric Encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
2. Bernstein, D.J., Buchmann, J., Dahmen, E.: Post-Quantum Cryptography. Springer (2008)
3. Bernstein, D.J., Lange, T., Peters, C.: Attacking and Defending the McEliece Cryptosystem. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 31–46. Springer, Heidelberg (2008)
4. Bernstein, D.J., Lange, T., Peters, C.: Wild McEliece. Cryptology ePrint Archive, Report 2010/410 (2010), `http://eprint.iacr.org/`
5. Biswas, B.: Implementational aspects of code-based cryptography. PhD thesis, École Polytechnique, Paris, France (2010)
6. Buchmann, J.: FlexiProvider. Developed by the Theoretical Computer Science Research Group of Prof. Dr. Johannes Buchmann at the Departement of Computer Science at Technische Universität Darmstadt, Germany, `http://www.flexiprovider.de/`
7. Faugère, J.-C., Otmani, A., Perret, L., Tillich, J.-P.: A Distinguisher for High Rate McEliece Cryptosystems. eprint Report 2010/331 (2010)
8. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)
9. Håstad, J.: Solving simultaneous modular equations of low degree. SIAM J. Comput. 17(2), 336–341 (1988)
10. Kobara, K., Imai, H.: Semantically Secure McEliece Public-Key Cryptosystems - Conversions for McEliece PKC. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 19–35. Springer, Heidelberg (2001)
11. Li, Y.X., Deng, R.H., Wang, X.M.: The equivalence of McEliece's and niederreiter's public-key cryptosystems. IEEE Trans. Inform. Theory 40, 271–273 (1994)
12. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. DNS Progress Report, 114–116 (1978)
13. Niebuhr, R., Cayrel, P.-L., Bulygin, S., Buchmann, J.: On lower bounds for Information Set Decoding over $\mathbb{F}_q$. In: SCC 2010, RHUL, London, UK (2010)
14. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory 15(2), 159–166 (1986)
15. Plantard, T., Susilo, W.: Broadcast Attacks against Lattice-Based Cryptosystems. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 456–472. Springer, Heidelberg (2009)
16. Pointcheval, D.: Chosen-Ciphertext Security for Any One-Way Cryptosystem. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 129–146. Springer, Heidelberg (2000)
17. Sendrier, N.: On the security of the McEliece public-key cryptosystem. In: Blaum, M., Farrell, P.G., van Tilborg, H. (eds.) Information, Coding and Mathematics, pp. 141–163. Kluwer (2002); Proceedings of Workshop honoring Prof. Bob McEliece on his 60th birthday
18. Deng, Y., Pan, Y.: A broadcast attack against ntru using ding's algorithm. Cryptology ePrint Archive, Report 2010/598 (2010), `http://eprint.iacr.org/`

Appendix: Pseudo Code for the Kobara-Imai Conversion

Algorithm 6. Kobara-Imai's γ conversion, modified for the Niederreiter cryptosystem.

Notation for Algorithm 6:
$\mathbf{Enc^N}$: The Niederreiter encryption
$\mathbf{Dec^N}$: The Niederreiter decryption

Additional system parameters: The length of the random seed len_s, and a constant *const*.

Encryption Enc$^\gamma$
INPUT: Message $m \in \mathbb{F}_q^*$
OUTPUT: Ciphertext $c \in \mathbb{F}_q^t$, where $t = r + \text{len}(m) + \text{len}(c) + \text{len}(s) - l$

Generate a random seed s of length len_s
$y_1 \leftarrow \text{PRG}(s) \oplus (m|const)$
$y_2 \leftarrow s \oplus h(y_1)$

$(y_4|y_3) \leftarrow (y_2|y_1)$, such that
$\text{len}(y_3) = l$
$\text{len}(y_4) = \text{len}(m) + \text{len}(c) + \text{len}(s) - l$

$z \leftarrow \varphi(y_3)$
$c \leftarrow y_4|\mathbf{Enc^N}(z)$

Return c

Decryption Dec$^\gamma$
INPUT: Ciphertext $c \in \mathbb{F}_q^t$, where $t = r + \text{len}(m) + \text{len}(c) + \text{len}(s) - l$
OUTPUT: Message $m \in \mathbb{F}_q^*$

$y_4 \leftarrow \text{MSB}_{\text{len}(c)-n}(c)$
$z \leftarrow \mathbf{Dec^N}(\text{LSB}_n(c))$
$y_3 \leftarrow \varphi^{-1}(z)$
$(y_2|y_1) \leftarrow (y_4|y_3)$
$r \leftarrow y_2 \oplus h(y_1)$
$(m|const') \leftarrow y_1 \oplus \text{PRG}(s)$

if const'=const **then**
  Return m
else
  Reject c
end if