Critical attacks in code-based cryptography

Anonymized for submission

Abstract. In this paper we present a survey on critical attacks in code-based cryptography. In particular, we consider three cryptosystems – McEliece, Niederreiter, and HyMES – and analyze their vulnerability against a number of these attacks. All cryptosystems show a weakness against several attacks. We conclude with a discussion of techniques to protect against critical attacks.

Keywords: Code-based cryptography, critical attacks.

1 Introduction

In 1994, P. Shor [14] showed that quantum computers can break most "classical" cryptosystems, e.g. those based on the integer factorization problem or on the discrete logarithm problem. It is, therefore, crucial to develop cryptosystems that are resistant to quantum computer attacks. Cryptography based on error-correcting codes is a very promising candidate for post-quantum cryptography since code-based cryptographic schemes are usually fast and do not require special hardware, specifically no cryptographic co-processor.

Error-correcting codes have been applied in cryptography for at least three decades, ever since R. J. McEliece published his paper in 1978 [10].

The first public key encryption scheme based on coding theory was proposed by Robert J. McEliece in 1978 [10]. The idea behind the McElice cryptosystem exploits the hardness of decoding a linear code. For the encryption scheme a binary $(n \times k)$ linear code is selected that is able to detect t errors efficiently. This code is disguised as a general linear code through multiplication with a scrambler matrix. For encryption the plaintext is encoded and t errors are randomly introduced. Without knowledge of the linear code correcting these errors and the used scrambler matrix decryption of the message is NP hard. Robert J. McEliece proposed the usage of binary Goppa codes which can easily be decoded using Pattersons algorithm if the Goppa code is known. With sufficient parameters this variant has resisted cryptanalysis so far and is seen as secure against quantum attacks[3].

McEliece (and the Niederreiter cryptosystem [13] from 1986) has some advantages over other public key encryption schemes, for example the encryption and decryption are faster than the widely spread RSA algorithm. However

some drawbacks exist, a large key (>100kb for reasonable security parameters) and a lower information rate compared to RSA. Nicolas Sendrier and Bhaskar Biswas addressed these problems with HyMES (Hybrid McEliece Encryption Scheme). They increased the information rate by encoding information in the error and reduced the public key size by using a generator matrix in systematic form $(ID|R)$.

These cryptosystems are based on the syndrome decoding (SD) problem (or the general decoding problem, which can be reduced to it), which has been proved NP-complete [2]. There are generic attacks against these cryptosystems, e.g. based on information set decoding or the generalized birthday algorithm, but these can be rendered infeasible by choosing appropriate parameters.

In practical applications, however, an attacker might not have to break the SD problem in order to decrypt a message. These critical attacks are possible usually when the attacker has some additional capability (e.g. a decryption oracle) or additional information (e.g. partial information about the plaintext).

Our contribution

In this paper, we provide a survey of critical attacks against the three cryptosystems above and discuss techniques to protect against them.

Related work

In [8], Kobara and Imai discuss some critical attacks (a subset of our list) against the McEliece cryptosystem and propose a conversion that protects against these attacks.

Organization of the paper

Section 2 describes the three code-based encryption schemes considered. In Section 3 we present the different critical attacks and their application to the cryptosystems above. Section 4 concludes the paper.

2 Code-based encryption schemes

In this paper, a (n, k, t) code denotes a linear code of length n and dimension k capable of correcting t errors. If t is not relevant, we write (n, k) code for short. The (Hamming) weight of a vector v is denoted $\mathrm{wt}(v)$ and refers to the number of non-zero entries.

2.1 McEliece

The McEliece public-key encryption scheme was presented by R. McEliece in 1978 [10]. 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 effcient decoding algorithm is known.

Let G be a $k \times n$ generator matrix for a (n, k, t) Goppa code, P an $n \times n$ random permutation matrix, S a $k \times k$ invertible matrix and $\mathcal{D}_G$ a decoding algorithm for the code generated by G. All matrices and vectors are defined over a finite field $\mathbb{F}_q$.
The private key is $(S, G, P, \mathcal{D}_G)$, while $\widehat{G} = SGP$ and t are made public.

Encryption To encrypt a message $m \in \mathbb{F}_q^k$, the sender generates a random vector $e \in \mathbb{F}_q^n$ with $\mathrm{wt}(e) = t$ and computes the ciphertext $c = m\widehat{G} + e$.

Decryption Recieving a ciphertext c, the recipient computes $\widehat{c} = cP^{-1} = mSG + eP^{-1}$. Since P is a permutation, $\mathrm{wt}(eP^{-1}) = \mathrm{wt}(e)$, so $\mathcal{D}_G$ can be used to decode it: $mSG = \mathcal{D}_G(\widehat{c})$. The recipient then chooses a set $J \subseteq \{1, \ldots, n\}$ such that $G_{.J}$ (the matrix formed by the columns of G indexed by J) is invertible, and computes $m = mSG \cdot G_{.J}^{-1} \cdot S^{-1}$.

2.2 Niederreiter

In 1986, H. Niederreiter proposed a cryptosystem [13] which can be seen as dual to the McEliece scheme. It uses the parity check matrix of a (usually 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 [9], it is threatened by different critical attacks.

Let H be a $r \times n$ parity check matrix for a (n, k, t) Goppa code, where $r = n - k$, and $\mathcal{D}_H$ a decoding algorithm for the code defined by H. Since the underlying Goppa code can only correct a certain number $t < n$ of errors, the

Niederreiter scheme uses a function φ to map the message to a word of length n and weight t: $\varphi : \mathbb{F}_q^l \mapsto \mathcal{W}_{n,t}$, where $l = \lceil \log_q \binom{n}{t}(q-1)^t \rceil$ and $\mathcal{W}_{n,t}$ is the set of vectors of length n and weight t.
The private key is $\mathcal{D}_H$, and H, t, and φ are made public.

Encryption Let $m \in \mathbb{F}_q^l$ be the message, then the ciphertext c is computed as $c = H \cdot \varphi(m)^T$.

Decryption Recieving a ciphertext c, the recipient decrypts it as $m = \varphi^{-1}(\mathcal{D}_H(c))$.

2.3 HyMES

The HyMES Hybrid McEliece cryptosystem developed by N. Sendrier and B. Biswas [5] increases the effciency 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, it behaves differently facing critical attacks.

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 McEliece scheme, while the second part is encoded into a word of weight t and serves as the error vector $e = \varphi(m_2)$.

Let G, P, S, and $\mathcal{D}_G$ be defined as for McEliece. Let φ be a function like in the Niederreiter scheme.
The private key is $(S, G, P, \mathcal{D}_G)$, while $\widehat{G} = SGP$, t, and φ are made public.

Encryption Let $m \in \mathbb{F}_q^{k+l}$ be the message, with l as above. Let m_1 be the first k bits of m and m_2 the remaining l bits. The ciphertext c is computed as $c = m_1\widehat{G} + \varphi(m_2)$.

Decryption The recipient first recovers m_1 as in the McEliece scheme above by computing $\widehat{c} = cP^{-1} = m_1SG + \varphi(m_2)P^{-1}$, applying the decoding algorithm $mSG = \mathcal{D}_G(\widehat{c})$, and finding $m_1 = mSG \cdot G_{.J}^{-1} \cdot S^{-1}$ with J as above. The second part of m is found by computing $m_2 = \varphi^{-1}(c - m_1\widehat{G})$.

3 Critical attacks

3.1 Description

Attack models specify how much information a cryptanalyst has access to when attacking a ciphertext. Some common attack models are:

1. Broadcast
2. Known partial plaintext
3. Message-resend
4. Related-message
5. (Adaptively) chosen ciphertext and Lunchtime
6. Reaction attack
7. Malleability

The ciphertext-only attack model is the weakest because it implies that the cryptanalyst has just the encoded message.

Different attack models are used for other cryptographic primitives, or more generally for all kind of security systems.

In this section, we give a brief description about the different critical attack we include in our analysis. The results are summarized in Figure 1.

3.2 Broadcast

The general idea behind a broadcast scenario is that a sender send an identical message (or very similar messages) to a number of recipients. The message is encrypted with each recipient's own public key. A broadcast attack attempts to exploit the knowledge that the ciphertexts correspond to the same or similar messages in order to reveal the cleartext.
In 1988, J. Håstad [7] 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 a long time, this type of attack has been considered only recently for cryptosystems based on error-correcting codes.
In [12], Niebuhr et al. presented a broadcast attack on the Niederreiter and HyMES cryptosystems that allowed to recover the cleartext in negligible time (10-20 seconds on a desktop PC) using only a small number of recipients – 3 for the Niederreiter parameters $(n, k) = (1024, 644)$.

3.3 Known partial plaintext

A chosen-plaintext attack (CPA) is an attack model for cryptanalysis which presumes that the attacker has the capability to choose arbitrary plaintexts to be encrypted and obtain the corresponding ciphertexts. The goal of the attack is to gain some further information which reduces the security of the encryption scheme. In the worst case, a chosen-plaintext attack could reveal the scheme's secret key.

This appears, at first glance, to be an unrealistic model; it would certainly be unlikely that an attacker could persuade a human cryptographer to encrypt large amounts of plaintexts of the attacker's choosing. Modern cryptography, on the other hand, is implemented in software or hardware and is used for a diverse range of applications; for many cases, a chosen-plaintext attack is often very feasible. Chosen-plaintext attacks become extremely important in the context of public key cryptography, where the encryption key is public and attackers can encrypt any plaintext they choose.

Any encryption scheme that can prevent chosen-plaintext attacks is then also guaranteed to be secure against known-plaintext and ciphertext-only attacks; this is a conservative approach to security.

Two forms of chosen-plaintext attack can be distinguished:

- Batch chosen-plaintext attack, where the cryptanalyst chooses all plaintexts before any of them are encrypted. This is often the meaning of an unqualified use of "chosen-plaintext attack".
- Adaptive chosen-plaintext attack, where the cryptanalyst makes a series of interactive queries, choosing subsequent plaintexts based on the information from the previous encryptions.

Non-randomized (deterministic) public key encryption algorithms are vulnerable to simple "dictionary"-type attacks, where the attacker builds a table of likely messages and their corresponding ciphertexts. To find the decryption of some observed ciphertext, the attacker simply looks the ciphertext up in the table. As a result, public-key definitions of security under chosen-plaintext attack require probabilistic encryption (i.e., randomized encryption). Conventional symmetric encryption schemes, in which the same key is used to encrypt and decrypt a text, may also be vulnerable to other forms of chosen-plaintext attack, for example, differential cryptanalysis of block ciphers.

The complexity of decoding the ciphertext decreases exponentially with every known bit. For example, attacking a ciphertext encrypted with McEliece using parameters (n, k) and knowing k_l bits is equivalent to attacking a McEliece ciphertext encrypted using parameters $(n, k - k_l)$. See [11,6,8] for more details.

3.4 Message-resend

A message-resend condition is given if the same message is encrypted and sent twice (or several times) to the same recipient. If the subsequent messages are related to the first by a known relation, it is called a related-message condition.

Suppose now that, through some accident, or as a result of action in the part of the cryptanalyst, both

$$c_1 = mSGP + e_1$$

and

$$c_2 = mSGP + e_2$$

are sent. We call this a message-resend condition. In this case it is easy for the cryptanalyst to recover m from the system of c_i.

Notice that $c_1 + c_2 = e_1 + e_2 (mod\ 2)$.

A message-resend condition can easily be detected by observing the Hamming weight of the sum of any two cryptograms. When the underlying messages are different, the expected weight of the sum is about 512. When the underlying messages are identical, the weight of the sum cannot exceed 100.

Berson [4] showed that a message-resend condition can be detected and how to exploit it.

In the case of McEliece and HyMES, these conditions can be detected by observing the Hamming weight of the sum of two ciphertexts. By comparing the two ciphertexts, an attacker can identify those bits where

- with high probability, neither ciphertexts contains an error, or
- with certainty, exactly one contains an error.

This allows to recover the ciphertexts in negligible time [11].

3.5 Related-message

In a related-message attack against a cryptosystem, the attacker obtains several ciphertexts such that there exists a known relation between the corresponding plaintexts. These ciphertexts may thus help the attacker to achieve a certain goal, for instance to gain useful information about the hidden plaintexts.

Such scenario is actually highly relevant in practise, as it is common that the apriori knowledge of the adversary on a message ïňĆow translates into a known relationship between the incoming messages. For example, this could

happen if the content of an encrypted message is followed by a serial number. Then an attacker who pretends to be the legitimate recipient could ask for a message-resend and, if he is able to determine the increment in the serial number, he will obtain encryptions of two messages with a known relation between them.

We will now generalize the message-resend attack. Suppose that there are two cryptograms

$$c_1 = m_1 SGP + e_1$$

and

$$c_2 = m_2 SGP + e_2$$

with $m_1 \neq m_2$ and $e_1 \neq e_2$. and that the cryptanalyst knows a linear relation, for example $m_1 + m_2$, between the messages. We call this a related-message condition. In this case it the cryptanalyst may recover the m_i from the set of c_i by doing one encoding and by then following the attack method described in [4].

Combining the two cryptograms we get

$$c_1 + c_2 = m_1 SGP + m_2 SGP + e_1 + e_2.$$

with $m_1 \neq m_2$ and $e_1 \neq e_2$.

related-message condition from the known relationship and the public key.

The cryptanalyst solves

$$c_1 + c_2 + (m_1 + m_2)SGP = e_1 + e_2$$

and proceeds with the attack, using $(c_1 + c_2 + (m_1 + m_2)SGP)$ in place of $(c_1 + c_2)$.

The message-resend attack is that special case of the related-message attack where $m_1 + m_2 = 0$.

3.6 (Adaptively) chosen ciphertext and lunchtime

In a chosen ciphertext attack, an attacker has access to a decryption oracle that allows to decrypt any chosen ciphertext (except the one the attacker attempts to reveal). In the general setting, the attacker has to choose all ciphertexts in advance before querying the oracle. In the adaptive chosen ciphertext attack, he is able to adapt this selection depending on the interaction with the oracle.

A specially noted variant of the chosen-ciphertext attack is the "lunchtime", "midnight", or "indifferent" attack, in which an attacker may make adaptive

chosen-ciphertext queries but only up until a certain point, after which the attacker must demonstrate some improved ability to attack the system.

The term "lunchtime attack" refers to the idea that a user's computer, with the ability to decrypt, is available to an attacker while the user is out to lunch. This form of the attack was the first one commonly discussed: obviously, if the attacker has the ability to make adaptive chosen ciphertext queries, no encrypted message would be safe, at least until that ability is taken away. This attack is sometimes called the "non-adaptive chosen ciphertext attack"; here, "non-adaptive" refers to the fact that the attacker cannot adapt their queries in response to the challenge, which is given after the ability to make chosen ciphertext queries has expired.

3.7 Reaction attack

This attack can be considered a weaker version of a chosen ciphertext attack. Instead of recieving the decrypted ciphertexts from the oracle, the attacker only observes the reaction of the oracle. Usually, this means whether the oracle was able to decrypt the ciphertext. In the context of side-channel-attacks, this can also mean observing decryption time, power consumption etc.
In one of the easiest variants, the attacker flips individual bits of the ciphertext he attempts to decode, and observes whether the oracle is able to decrypt it. If that is the case, the bit corresponds to an error bit (McEliece / HyMES). In the Niederreiter case, the same can be achieved by adding columns of the parity check matrix to the syndrome.

3.8 Malleability

Malleability is a property of some cryptographic algorithms. An encryption algorithm is malleable if it is possible for an adversary to transform a ciphertext into another ciphertext which decrypts to a related plaintext. That is, given an encryption of a plaintext m, it is possible to generate another ciphertext which decrypts to $f(m)$, for a known function f, without necessarily knowing or learning m.

Malleability is often an undesirable property in a general-purpose cryptosystem, since it allows an attacker to modify the contents of a message. For example, suppose that a bank uses a stream cipher to hide its financial information, and a user sends an encrypted message containing, say,

```
"TRANSFER 100 USDOLLARS TO ACCOUNT 199."
```

If an attacker can modify the message on the wire, and can guess the format of the unencrypted message, the attacker could be able to change the amount of the transaction, or the recipient of the funds, e.g.

"TRANSFER 100000 USDOLLARS TO ACCOUNT 227."

On the other hand, some cryptosystems are malleable by design. In other words, in some circumstances it may be viewed as a feature that anyone can transform an encryption of m into a valid encryption of $f(m)$ (for some restricted class of functions f) without necessarily learning m. Such schemes are known as homomorphic encryption schemes.

A cryptosystem may be semantically secure against chosen plaintext attacks or even non-adaptive chosen ciphertext attacks ($CCA1$) while still being malleable. However, security against adaptive chosen ciphertext attacks ($CCA2$) is equivalent to non-malleability.

As shown in [8], McEliece is malleable when Niederreiter is not. HyMES is also malleable.

3.9 Results

The results of our analysis are summarized in Fig. 1.

		McEliece	Niederreiter	HyMES
Plaintext	Broadcast [12]	no	*	*
	Known partial [11]	*	*	*
	Message-resend [4,11]	*	no	*
	Related-message [4,11]	*	no	*
Ciphertext	Chosen	*	*	*
	Adapt. chosen	*	*	*
	Batch chosen	*	*	*
	Lunchtime	*	*	*
	Adapt. chosen [15]	*	*	*
	Reaction [8]	*	*	*
	Malleability [8]	*	no	*

Fig. 1. Critical attacks

From this table, we remark that HyMES inherits of the bad properties of McEliece faces message-resend, related-message attacks and malleability, and also the bad property of HyMES faces broadcast attack. Its special construction is not adapted to critical attacks but using Imai-Kobara's like construction it can be turned into a CCA2 secure encryption scheme as detailed in [12].

The contribution of this paper is essentially the last column about the resistance of HyMES faces critical attacks.

4 Conclusion

In this paper we have analyzed the vulnerability of the McEliece, Niederreiter, and HyMES cryptosystems against several critical attacks. All schemes show a weakness against several of these attacks, HyMES against all of them. This result emphasizes the importance of conversions for these cryptosystems that protect against critical attacks. When choosing appropriate conversions, it is important to consider all critical attack above, since some conversions protect against only some of them; e.g. the well-known Optimal Asymmetric Encryption Padding (OAEP) by Bellare and Rogaway [1] is unsuitable for the McEliece/Niederreiter cryptosystems since it does not prevent reaction attacks. A secure conversion for the McEliece cryptosystem has been proposed in [8], and for the Niederreiter cryptosystem in [8]. These are not applicable to the HyMES cryptosystem; however, the McEliece conversion contains a similar technique as is used in HyMES, so the resulting scheme contains the efficiency improvements introduced in HyMES.

References

1. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *EUROCRYPT*, pages 92–111, 1994.
2. E. Berlekamp, R. McEliece, and H. van Tilborg. On the Inherent Intractability of Certain Coding Problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.
3. D. J. Bernstein, J. Buchmann, and E. Dahmen. *Post-Quantum Cryptography*. Springer, 2008.
4. T. A. Berson. Failure of the McEliece Public-Key Cryptosystem Under Message-Resend and Related-Message Attack. *Crypto 97*, 1997.
5. B. Biswas and N. Sendrier. Mceliece Cryptosystem Implementation: Theory and Practice. In *PQCrypto*, pages 47–62, 2008.
6. A. Canteaut and N. Sendrier. Cryptoanalysis of the original McEliece cryptosystem. In *ASIACRYPT*, pages 187–199, 1998.
7. J. Håstad. Solving simultaneous modular equations of low degree. *SIAM J. Comput.*, 17(2):336–341, 1988.
8. H. Imai and K. Kobara. Semantically Secure McEliece Public-Key Cryptosystems - Conversions for McEliece PKC. *Proc. of 4th International Workshop on Practice and Theory in Public Key Cryptosystems (PKC 01)*, pages 19–35, 2001.
9. Y.X. Li, R.H. Deng, and X.M. Wang. The equivalence of McEliece's and niederreiter's public-key cryptosystems. *IEEE Trans. Inform. Theory*, 40:271–273, 1994.
10. R. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. The Deep Space Network Progress Report, DSN PR 42–44, 1978. `http://ipnpr.jpl.nasa.gov/progressreport2/42-44/44N.PDF`.
11. R. Niebuhr. *Application of Algebraic-Geometric Codes in Cryptography*. Verlag Dr. Müller, 2008.
12. R. Niebuhr and P.-L. Cayrel. Broadcast attacks against code-based encryption schemes. In *WEWORC*, 2011.
13. H. Niederreiter. Knapsack-type Cryptosystems and Algebraic Coding Theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.
14. P. W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*.
15. H.-M. Sun. Further cryptanalysis of the mceliece public-key cryptosystem. *IEEE Communications Letters*, 4(1):18–19, Jan 2000.