Postquantum cryptography: An overview on the McEliece cryptosystem

Jannis Priesnitz
University of Applied Sciences Darmstadt
Department of Computer Science
Schöfferstraße 3
64295 Darmstadt
Email: jannis.priesnitz@stud.h-da.de

***Abstract*—This article gives an elaborated overview of the McEliece cryptosystem from todays perspective. Starting with some basics on post quantum computing the reason why such an cryptoscheme is necessary is enlighted. The cryptosystem itself is described into detail supported by an example. The Niederreiter cryptoschme as an famous variant is described. Finally state of the art issues and attempts to solve them are presented.**

***Index Terms*—McEliece, Niederreiter, post quantum cryptography**

I. Introduction

State of the art asymmetric cryptosystems compared with usage of sufficient long Keys are considered as "safe" on current computer systems. This changes immediately when quantum computers enter the scene. But why is this the case? In this paper I briefly present the main differences between normal computer systems and quantum computers and why state of the art cryptosystems like RSA and ECC are not save in the quantum computation world. In addition to that I, present a extensive introduction to the McEliece Cryptosystem explain the strengths and downsides of the algorithm and give some details about the developments of codes which can be used for the system.

Since decades computer systems have been seen as digital circuits based on the rules of physics. A bit in this systems is seen as a value of voltage. If this value is above a certain, level the binary representation is one, otherwise it's zero. On quantum computers, information is handled in binary, too. The main difference between the bit representation and the quantum bit representation (so called qubit) is that in addition to the two states zero and one, there are several more states which one qubit can reach due to a superposition. This means that a qubit is not either in state one or in state zero but in theory can have an arbitrary number of different states. All these states can appear with an certain probability. Each of these states is able to compute one possibility of a NP hard problem.

Due to this fact, quantum computers are able to solve problems which are NP hard in sense of complexity theory much faster than traditional computers.

One of these computational problems, which are traditionally hard to solve, is the prime factorization problem which is in serveral variations the core of many public key algorithms such as the RSA cryptosystem. The second main category, the elliptic curves cryptography is even more affected by this problem as the logarithm of a finite field of an elliptic curve can be computed efficiently and can be broken in less time compared to the RSA algorithm because of lower keysizes. In conclusion, there is no established asymmetric cryptoscheme which is post quantum resistant. Therefore completely different approaches are needed.

Outline

In section II, some background information regarding post quantum cryptography is given. With help of Shors algorithm the principle of quantum cryptography is described more precisely. Section III takes a detailed look into the McEliece cryptosystem supported by an example in section VII. The Niederreiter cryptosystem follows in section IV with an additional view on signing with it in section V. The underlying Goppa codes are described in section VI. Finally in section VIII a slight overview on attacker models and current developments such as wild McElice are given.

II. Post quantum cryptography

In order to reach post quantum proof cryptographic algorithms, some fundamentally different approaches then the established ones are required. Shors Algorithm in II-A reveals that cryptography based on integer factorization or the discrete logarithm problem is no longer an issue for quantum computers. Therefore other algorithm types are needed which are shortly described in II-A0b.

A. Shors Algorithm

Peter Shor presented in 1994 an algorithm which is able to factorize a composite number n into its prime factors on a quantum computer. This algorithm only needs $logn$ qubits and has a runtime of $O((log\ n)^3)$ for finding a non trivial factor of n.

Procedure Shors algorithm divides into a classical part which can be executed on a conventional computer and a quantum part which has to be executed on a quantum computer in order tp perform the computation efficiently.

The basic idea is that the classical part reduces the problem while the quantum part finds the order of the group in which n is. In this section Shors algorithm is shortly sketched to get an intend of the method of quantum computing and the principles an quantum proof algorithm has to follow.

a) Classical part The classical part of the algorithm mainly contains computation of the greatest common devisor of a randomly selected number lower than n and n itself. Then it is needed to compute the order r of x which is where the quantum part emerges. The classical part is executed in a loop while the order r is odd or x to the power of $r/2$ is equivalent to $-1\ mod\ n$ If this is not the case the gcd of $x^{(r/2)} - 1$ n is computed.

why?

b) Quantum part At first a q is determined which is a power of 2 and lies between n^2 and $2 * n^2$. To prepare the input quantum register the superposition of all states $a\ mod\ q$ is loaded. In this case a is smaller than n. The output quantum register is initialized with all states of $x^a(modn)$. A quantum Fourier transformation is computed on the input register which is treated as black box in this work. The result values are gathered from the output register.

To sum up Shors algorithm with support of a quantum computer is able to find the period of a prime in polynomial time.

With this algorithm all cryptography based on prime factorization can be broken by a quantum computer in polynomial time. Note that not the quantum fourier transformation, which takes only $O(1)$ operations, is the bottleneck but the fast exponentiation which takes $O(log\ n)$ with the currently best implementation[1]. Together with the pre- and postprocessing the time complexity is at $O(log(n)^3)$.

instead of?

Candidates for post quantum cryptography

In this section a short overview over promising state of the art post quantum cryptoschemes based on [2] is given.

Lattice-based cryptography One of the most studied types of algorithm is the lattice based cryptography which exists in several variants. Algorithm works on a lattice over a n-dimensional finite Euclidian field L with a strong periodicity property. A set of vectors provides the basis of L in the way that every element is uniquely represented. The cryptographic problem is to find the closest vector to a given lattice point e.g. by adding an error vector[2][3].

Multivariate cryptography Multivariate cryptography is based on a multivariate polynomials over a finite field F which are defined over both a ground and an extension field. In the case of solving these multivariate polynomial equation systems they are NP-complete and due to this fact a candidate for post quantum cryptography. They are topic of studies for a long time and are promising especially for signature schemes[2][4].

Hash-based cryptography Hash-based algorithms such as Lamport- [5] and the Merkle [6] signature scheme are based on strong hash functions but have the drawback that only a limited count signatures can be created per key. The algorithm reduces the one time signature to an hash value using a hash function[2].

Code-based cryptography The forth group, the code based algorithms, are based on error-correcting codes. First investigations have been developed by Robert McEliece using random Goppa codes[7]. This paper deals with the properies of McEliece- and the related Niederreiter cryptosystem[2][8].

This raw overview of some state of the art algorithms show different approaches but in general it could be said that a much higher effort has to be taken to achieve a strong system compared to the traditional ones.

III. The McEliece cryptosystem

Was macht die Matrix S und was macht die Matrix P. Warum invertierbare und warum Permutationsmatrix. Verstehen Sie den Sinn hinter diesen beiden Matrizen?

In 1978, Robert McEliece suggested an asymmetric quantum resistant cryptosystem based on the theory of algebraic codes. He selected binary Goppa codes with the property of irreducibility as base a for the cryptosystem [9]. The chosen code C has a length of $n = 2^m$ and a dimension of $k >= n - tm$ where t denotes the degree of the polynomial over $GF(2^m)$. In this case m indicates the size of the binary field. These codes are able to correct any pattern of t or fewer errors. For each of this codes, there exists an irreducible polynomial of degree t. The main reason for McEliece to select this setup is that there exists an fast algorithm to decode these codes [10]. A illustrating example can be found in VII.

Key generation

For the key-generation, a n and t with above mentioned properties are picked. Additionally, an irreducible polynomial of degree t over $GF(2^m)$ is selected randomly. The probability that this selection leads to an irreducible polynomial is $1/t$ and there is an efficient algorithm to prove the irreducibility[11]. In the next step a generator matrix G which is of size $n \times k$ is produced. This can be transformed into canonical form.

Now the information of G has to be obscured. Therefore a random dense $k \times k$ matrix S which is nonsingular and a random $n \times n$ permutation matrix P is selected. Both of them are multiplied to $G' = SGP$. Due to the matrix multiplication properties, the linear code generated by G' has the same rate and distance like G. G' is the public generator matrix and is sent to the encrypting entity.

The following encryption algorithm is published so that the encrypting entity can use it.

Encryption

First of all, the message m which is to be encrypted has to be devided into k-bit blocks. The public key encryption is performed by $x = uG' + z$ with u being one of such k-bit blocks. z denotes a randomly generated vector with length n and weight t [1].

[1]The weight of an vector is defined as Hamming weight.

The vector x is the encrypted message which is transm why? to the private key owner who can decrypt the message block as follows.

Decryption

The decryption of one block x starts with computing $x' = xP^{-1}$ with P^{-1} as inverse of the permutation matrix P.

With an error correcting algorithm for the code C, the codeword u' "next" to x', is computed. As already mentioned with Pattersons algorithm there is an efficient method for computing the error correction, which is described in [13]. To get a plaintext message block the calculation $u = u'S^{-1}$ is performed[9].

Correctness

Assuming that P is a permutation matrix and random vector z with length n and weight t, it obvious and can easily be computed that zP^{-1} has a weight of t or less. As discussed, the computation is $x' = xP^{-1} = (mG' + z)P^{-1} = mSG + zP^{-1}$. The chosen Goppa code C is designed to correct up to t errors. On the other hand, mSG has a maximum distance from xP^{-1} of t. This leads to the fact that the correct code mS is determined by the algorithm. To obtain the message block m from mS, we can easily multiply the inverse $m = mSS^{-1}$[12].

Security properties

The security of the presented scheme refers on the one hand to the basics of learning with errors principle. More precise the hypothesis of *Learning Parity with Noise*[14] which is not discussed in this article. On the other hand it refers to the hypothesis that the generator matrix G has to be indistinguishable from any other $k \times n$-matrix. This leads to the property of a trapdoor function.

auswei ten –¿ pa- per

IV. The Niederreiter cryptosystem in comparison to McEliece

Braun: Bei Niederreiter: Was ist eine Kontrollmatrix.

The Niederreiter cryptosystem is highly comparable to the McEliece cryptosystem due to the fact that it is following the same basic idea.

Niederreiter designed his $(n, k, 2t+1)$ linear code C over a Galois field too. In contrast to McEliece the code size does not have to be a power of 2 instead of an arbitary integer $GF(q)$. Another difference to McEliece is the usage of a $(n-k) \times (n)$ parity check matrix H instead of the generator matrix G. The nonsigular $(n-k) \times (n-k)$ matrix M is defined sightly different from McEliece $(k \times k)$. The permutation matrix P, an arbitrary $n \times n$ matrix, is exactly defined in the same way compared to McEliece. The private key is then defined as M, H and P, the public key consists of $H' = MHP$ and the hamming weight t which is quite the same compared to McEliece.

why?

The messages y in the system of Niederreiter have to be n dimensional vectors over $GF(q)$ and they must have a hamming weight of t. This is an important fact issuing the signature creation in section V. Encryption with Niederreiter is performed with $z = yH'^T$ which again is comparable with the encryption operation $x = uG' + z$ in McEliece except that there is no error vector needed because it is already represented in H'. The ciphertext consist of the syndrome and has only $n-k$ bit dimension compared to n bit in McEliece.

The decryption is computed firstly with $(yP^T)H^T = z(M^T)^{-1}$ Then H is eliminated by an error correction algorithm which leads to $(yP)^T$ which can easily computed to the plaintext y. [15][16][17]

V. Signing with Niederreiter

Besides en- and decryption signature building and verification is an common requirement to an asymmetric cryptoscheme. In state of the art algorithms principal is quite simple: The message to be signed is *de*crypted with a given public key. The verifier *en*crypts the message with his private key an compares the result with the message.

In the case of McEliece this is not so easy because there is no possibility to decrypt (= sign) an message before encrypting (= verify) it. More precise in most cases the process of signing produces a syndrome whose error pattern is bigger than the dessen error correcting property t. In fact it is hard to create a ciphertext that fits to the error correcting properties of the encryption without using it.

Compared to en- and decryption signing and verifying is much harder to realise with McEliece. Just in 2001 a digital signature scheme were presented by Courtois et. al. The problem with signing a given hash value n is that generally it has a higher hamming weigth than the decoding capacity t of the used code is. More general one can say that it is difficult to generate a random ciphertext without using the encyption algorithm. [18]

Complete decoding One possible solution would be to use complete decoding. Therefore not only words within the radius of t can be decoded but all words laying in the code space. In other words with complete decoding we can find an error pattern to any given syndrome as long as it is in the code space. This means that we have to add a δ with random collumns from the parity check matrixto t. The decoding works exactly when all of the δ-columns fit to an error position because then the syndrome will fit to an word of weight t. Else we have to iteratively add another δ to t and try again.

why pc- matrix

From this properties we now can construct a digital signature scheme: We have to select a δ which is small enough to get an usable key size but on the other hand has a good security.

For achieving a small δ the code has to be selected carefully in the way that it has to have a high density of decodable syndromes. This makes sure that the δ is kept small because the probability of finding a fitting one is high. For building up a signature the signer now takes a syndrome and hashes it together with the document. This is tried as long as he gets a decodable syndrome with modifying the document with every try (possibly with some kind of padding) . [18]

VI. Codes can be used for McEliece and Niederreiter

both the same?

Was macht prinzipiell ein Decodierungsalgorithmus? Wie wird aus einem empfangenen Vektor ein Codewort? Was ist eine Generatormatrix?

As described in the last sections the McEliece cryptosystem is based on the properties error correcting codes. In principle many codes with a good decoding algorithm could be used for the cryptosystem. However there are some properties which leads to higher security than others. In this section a short introduction to the basics of error correcting codes is presented followed by a deeper view on Goppa codes which are suggested by McEliece and are still the best choice.

what makes a code usable

Principles of error correcting codes

As the name states the basic purpose of error correcting code is to detect and correct errors which occur during the transition of messages. Therefore additional information is added to the message by the sender. Knowing the encoding algorithm receiver is able to decode the original word, detect errors and correct them with help of the additional information and the corresponding decode algorithm.

start earlier: principles of error correction

Binary Due to the fact that almost all transition is performed by computers and is in binary the explanations in following are focussed on the binary case too. Never the less all of them hold in the general case.

For generating these additional information a vector with binary indexes $V = [c1c2...cn]|ci \in 0, 1$ is created as set of $\mathbb{F}_2$. Further the addition is defined componend-wise modulo 2, which also can be seen as XOR operation. With this definition V covers all n-tuples in $\mathbb{F}_2$. Having this, a binary, linear $[n, k]$ code can be defined as subspace of $\mathbb{F}_2$ in which the n is the length and the k is the dimension. A codeword is represented as vector in the code.

Hamming distance Another important variable for error correcting codes is the hamming weight or more precise the minimum weight d of a vector. The weight is simply defined as count of non-zero positions in the vector. Furthermore the minimum weight is defined as minimum is the smallest weight of a codeword under the condition that it is not zero. A $[n, k, d]$ code can correct $t = \lfloor \frac{d-1}{2} \rfloor$ errors.

Goppa Codes

Goppa codes follow the principals described previously. In this section highlights the special basics (of Goppa codes) regarding the McEliece cryptosystem. Firstly some important basic properties and terminologies are presented. In the second part an example is discussed.

DO SO

Linearity Goppa Codes are linear codes. This is not a conincidence because non-linear codes do not provide an efficient decoding algorithm like Pattersons algorithm for linear codes does[13]. In [19] some remarkable results to achieve more efficiency are stated but also it is mentioned, that nonlinear codes are not able to achieve more comparable efficiency. However most of the popular codes are linear ones this condition does not rule out so many codes.

Irreducibility y

das könnte auch in ECC, da alle eccs durch polynomer erzeugt werden können und die irreduzibel sein müssen oder?

To achieve a finite field with usable properties for a Goppa code the generator polynomial has to bew irreducible. This means that for a polynomial p over a finite field $GF(p^m)$ there exists no polynomial $p' \in GF(p^m)$ of lower degree which divides p.

Efficency(?) To keep the keyspace as efficient as possible the factor between the matrix dimensions n, k and the error correction range t (respectively the minimum hamming distance d) should be as high as possible. The most important point is that Goppa Codes are a suspect of research for years so the probability of is really low.

VII. Example of the McEliece cryptosystem

In section III the threoretical the aspects of the McEliece cryptsystem are described. To illustrate the algorithm more into detail an example is provided in this section. Note that the parameter are far to small an the code is based on much simpler hamming codes. ?

Consider Alice and Bob as two entities who want to communicate encrypted. Therefore Bob chooses the following matrix G as generator:

$$G = \begin{Bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{Bmatrix}$$

Additionally a matrix S to obscure the generator is chosen by Bob:

$$S = \begin{Bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{Bmatrix}$$

A permutation matrix P is also needed:

$$P = \begin{Bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{Bmatrix}$$

Now Bob calculates the public key G':

$$G' = SGP = \begin{Bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{Bmatrix}$$

G' is sent to Alice who wants to encrypt the message

$$x = \{1 \quad 1 \quad 0 \quad 1\}$$

Therefore she chooses a random error vector e:

$$e = \{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0\}$$

This matrix is sent to Alice who calculates $y = x * G' + e$:

$$y = x * G' + e = \{0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0\}$$

This codeword is sent to Bob who decodes the word with P^{-1} which leads to:

$$y' = y * G^{-1} = \{1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0\}$$

Now the fast decoding algorithm enters the scene[2]. In this case Bob compares the result to the rows of G. The row with the lowest Hamming distance is the corresponding codeword which leads to:

$$y = x * G' + e = \{\mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad 1 \quad 1 \quad 0\}$$

Because of the structure in G is the identity matrix the first four digits are taken as error vector . The last step to decrypt the message is to multiply the error vector xS with S^{-1}:

$$x = y * S^{-1} = \{1 \quad 1 \quad 0 \quad 1\}$$

So Bob is able to decrypt the message from Alice properly.

A. Different message

If Alice wants to encrypt the message $x = \{1 \quad \mathbf{0} \quad 0 \quad 1\}$ instead of $\{1 \quad 1 \quad 0 \quad 1\}$ the decrypted word which is sent to Bob is $y = \{0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0\}$. He performs $y' = y * G^{-1} = \{0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0\}$ which decodes as described to $\{0 \quad 1 \quad 0 \quad 0\}$ and finally with $x = y * S^{-1}$ to $\{1 \quad 0 \quad 0 \quad 1\}$. From these two examples we can see that flipping a single bit can lead to a code of an other "domain" ($\{0 \quad 1 \quad 0 \quad 0\}$ in this case) during encryption which is correctly decrypted because of the usage of similar codes. The behavior can be seen when the 1-digit of the error vector is changed. Note that an error vector of Hammingweight > 1 is not working properly because of the error correcing capabilities of the code.

ab hier eher ausblick –¿ further topics

VIII. Further topics

To get an elaborated overview over all facets of the McEliece cryptosystem many more topics have to be issued which are not covered in detail in the article. In this sections some of the most important topics are described slightly.

ausweiten aber niedrige prio

A. Resistency against various attacker models

For a cryptoanalyst attacker models are the first choice to design attacks an given scheme. Two of the most common attacker model and resistance approaches are slightly presented in this section.

[2]Note that in this example the Hamming deconding is executed to demonstrate the wordwise.

Chosen plain text resistance Chosen plaintext restance means that an attacker own two different plaintexts p_1 and p_2 and gets two belonging ciphertexts c_1 and c_2 he has no chance to gather information from the ciphertext which tells him that it belongs to plaintext p_1 or p_2. The attaacke is able to use the two known plantexts and a randomly chosen message to compute a sample c. This is now subtracted during the decryption because Now the attacker can easily compute the message because one of e_1 or e_2 is the right error vector. He can determine the right message by seeing that it has less than 2t-1 non zero terms in it.

Chosen cipher text resistance Like the RSA cryptosystem in the original idea is not resistant against chosen cipher attacks.

This means if an attacker has access to an oracle which is able to decrypt an given ciphertext without knowing the key and the attacker is permitted to give all ciphertexts except the one he is asked to crack to the oracle he is not able to gather any reasonable information from the oracle. This goes over several iterations and is called CCA2-attack in the literature. Resistance against this attack is described as IND-CCA2.

Like the RSA cryptosystem uses padding standards such as PSS for signing and OAEP for encrypting to achieve IND-CCA2 the MECS has some advanced padding and randomization schemes as well. The common approach on this is to add random vectors and perform shift operations on the ciphertext during the encryption[20].

Practical security attacks

Every cryptosystem with certain parameters has to hold against attacks to be considered as safe. These attacks can be very different for every cipher suite. For the McEliece cryptosystem there are two types of promising attacks named structural and information set decoding attacks.

Structural attacks describes the approach of constructing a code for the information of code generated by the public key G'. If this succeeds the private key G is revealed. To fullfill such an attack an equivalent code E to G' has to be found, which means that a permutation m exists which permutes G' to E'. Then E' generated by G' and the E generated by G are from the same class. Because of knowing this dependence an attacker can compare a representant from each of such classes to E' to get an equivalent code. This still is a huge computational problem which is not solvable is sufficient time because the cardinality of such an equivalent matrix is small. [21]

Information set decoding is the most promising method of breaking McEliece so far. The idea behind this is to find a set of error free coordinates in a codeword. If the corresponding columns in the generator matrix form a invertible submatrix the information from original message word can be obtained easily.

If there is a (n,k,t) code of a generator G and a ciphertext $c = uG + e \in \mathbb{F}_2^n$. An attacker now randomly selects a subset (the information set) $I \subset 1, ..., n$ of size k beeing linearly

independent. Now an "masking" function δ is needed which projects c on I such as $\delta(c)_{i\in I} = (uG)_{i\in I} + e_{i\in I}$

If this projection leads to a error free result (also $weight(e) = 0$) we can obtain the message word from the code by $\delta(r) * \delta(G)^{-1} = \delta(uG)^{-1} * \delta(G)^{-1} = u$. But this is not the common case an typically needs many iterations.

Note that information set decoding is not an total break of the system because it does not reveal the private key but only the message word corresponding to th codeword.

Recommended parameters

In [22] Niebuhr et al computed some key length based on the Model of Lenstra and Verheul. They motivated that keylength have to grow with speedup of computer hardware and concluded state of the art keylenths of traditional algorithms which is ported to the McEliece Cryptosystem by Niebuhr et al. Based on the best known attack of Sendrier at al, which is able to break the original parameters ($n = 1024$, $k = 524$, $t = 50$) with a minimal binary work factor of $2^59,9$ operations [23], they presented the following values as secure parameters until the year 2050: $n = 2804$, $k = 2048$ and $t = 66$. A public key generated has a size of about 189 kilobyte. This is almost 400 times more keyspace than a RSA key and almost 3000 times more than an eliptic curves key which promises safety until 2050 too, but only on conventional computers. The key size issue is decribed a liitle more into detail in VIII.

Challenges and optimizations

From the practical point of view the presented schemes more sketches of one way trapdoor functions than fully filled crypto schemes. These trapdoors can be used in many different ways to generate a "standard" crypto scheme maybe as replacement for RSA- and ECC-cryptography, but this has to be done carefully. In this section a short summary of the challenges are such as usability properties, computation time and key sizes is presented These three topics are highly connected to each other.

followed by some variants of the scheme ??? from??

Confidence One of the most delicate tasks when putting an abstract algorithm into a real crypto application is to achieve confidence intoit. Many pitfalls like wrong codes or parameters and weak padding algorithms have to discovered by cryptoanalysts to increase the confidence level. However both, the RSA and the McElice where suggested over 35 years ago, the development on RSA was pushed forward much faster because of the adavantage of a lower keyspace. This development has to be done in the next years for McEliece to get over issues discoverd in the past and achieve an optimized reliable system.

Efficiency In most cases the abstract term "efficiency" is divided in time- and space efficiency. Daniel J. Bernstein made [2] a short comparison with the RSA which should be summed up here shortly.

Time effciency is seen as time needed to perform the key generation, encryption and decryption. More precisely the different "operations" executed and their complexity are reviewed.

Compared to RSA the McEliece is quite time efficient. As described in section III the main computations are matrix operations in the binary field, which can be broken down to highly efficient additions, multiplications, shifts etc. for which no special hardware requirements are needed[2].

ggf gegensatz zu rsa beschreiben

Space efficiency refers to the storage needed to be provided in order to execute the algorithm. In our case the bytes needed to transmit the public key to the encryption entity or signer additionally is an important topic.

Is most important drawback is the key sizes to be transferred. With n denoted as code length respectively key length McEliece needs about $b^2 * (lgb)^2$ bits whereas the RSA only needs $(\approx 0,16) * b^3/(lgb)^2$ bits. One might notice the higher exponent in RSA formular which leads to an faster growing key space if the security level goes to ∞ but the break even point is fare beyond practical relevant key sizes. For keys assumed as secure in these days the McEliece key is size is about 10 times higher than RSA[3]. Refer also to section VIII for a deeper view on keylenght with different parameters.

heißt das so??

Usability The last important point is to provide an usable interface for the cryptosystem. New crypto protocols can only be successful if many applications in different domains are supporting them. Therefore padding scheme like addressed in section VIII has to be provided and standardised to make sure that different software systems on different platforms can speak to each other.

Wild McEliece

Due to the fact that the McEliece cryptosystem was published over 30 years ago and still is one of the most promising post quantum security algorithms many variants came up. Many of the previous described issues could be solved by reducing the key space. In conclusion this means optimising on the used codes. Many codes like Reed-Solomon codes, quasi dyadic codes, and variants of Goppa codes have been tried, seen as secure and been broken a little later.

The most promising optimization is idea of using "wild" Goppa codes[24]. Wild Goppa codes are codes which are no longer over a field F_2 but on F_q with q as small prime number. These approach was broken by generating the squeare code of the public key and revealing information of the original private key, because the square product lines are not equally distributed compared to the binary code. In [25] the approach wild McEliece idea is improved with an "incognito" variant. As the name says the wild Goppa codes are hidden by multiplying a extra factor f to the code and by using only special codes described in [26]. This topic and the question if squaring technique is still considerable for the incognito version, is still a topic of research.

[3]Assuming unoptimized but optimal algorithms without attention of quantum computers.

IX. Conclusion

The topic of this work was to present an basic introduction on postquantum cryptography with focus on the McElice cryptosystem. With basic information of the underlying Goppa codes a deeper view on the algorithm was given supported by an example computation. Moreover special properties and downside were lighted.

In conclusion the McEliece cryptosystem stays one of the most promissing canditates for a post quantum cryptography and is still an huge topic of research as the literature references show. Never the less there are huge downsides, especially the large keysize and with a view on the practical usability.

References

[1] I. L. Markov and M. Saeedi, "Constant-optimized quantum circuits for modular multiplication and exponentiation," *arXiv preprint arXiv:1202.6614*, 2012.

[2] D. J. Bernstein, "Introduction to post-quantum cryptography," in *Post-quantum cryptography*, pp. 1–14, Springer, 2009.

[3] Wikipedia, "Lattice-based cryptography — wikipedia, the free encyclopedia," 2017. [Online; accessed 10-March-2017].

[4] Wikipedia, "Multivariate cryptography — wikipedia, the free encyclopedia," 2017. [Online; accessed 10-March-2017].

[5] Wikipedia, "Lamport signature — wikipedia, the free encyclopedia," 2017. [Online; accessed 10-March-2017].

[6] Wikipedia, "Merkle signature scheme — wikipedia, the free encyclopedia," 2017. [Online; accessed 10-March-2017].

[7] Wikipedia, "Binary goppa code — wikipedia, the free encyclopedia," 2015. [Online; accessed 10-March-2017].

[8] Wikipedia, "Niederreiter cryptosystem — wikipedia, the free encyclopedia," 2017. [Online; accessed 10-March-2017].

[9] R. J. McEliece, "A public-key cryptosystem based on algebraic," *Coding Thv*, vol. 4244, pp. 114–116, 1978.

[10] R. McEliece, *The theory of information and coding*. Cambridge University Press, 2002.

[11] E. R. Berlekamp, "Algebraic coding theory," 1968.

[12] Wikipedia, "Mceliece cryptosystem — wikipedia, the free encyclopedia," 2017. [Online; accessed 12-March-2017].

[13] N. Patterson, "The algebraic decoding of goppa codes," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 203–207, 1975.

[14] K. Pietrzak, "Cryptography from learning parity with noise," in *International Conference on Current Trends in Theory and Practice of Computer Science*, pp. 99–114, Springer, 2012.

[15] N. Sendrier, "Niederreiter encryption scheme," in *Encyclopedia of cryptography and security*, pp. 842–843, Springer, 2011.

[16] Y. X. Li, R. H. Deng, and X. M. Wang, "On the equivalence of mceliece's and niederreiter's public-key cryptosystems," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 271–273, 1994.

[17] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *PROBLEMS OF CONTROL AND INFORMATION THEORY-PROBLEMY UPRAVLENIYA I TEORII INFORMATSII*, vol. 15, no. 2, pp. 159–166, 1986.

[18] N. T. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a mceliece-based digital signature scheme," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 157–174, Springer, 2001.

[19] F. Zeng, *Nonlinear codes: representation, constructions, minimum distance computation and decoding*. PhD thesis, Universitat Autònoma de Barcelona, 2014.

[20] N. Dottling, R. Dowsley, J. Muller-Quade, and A. C. Nascimento, "A cca2 secure variant of the mceliece cryptosystem," *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6672–6680, 2012.

[21] S. Au, C. Eubanks-Turner, and J. Everson, "The mceliece cryptosystem," *Unpublished manuscript*, vol. 5, 2003.

[22] R. Niebuhr, M. Meziani, S. Bulygin, and J. Buchmann, "Selecting parameters for secure mceliece-based cryptosystems," *International Journal of Information Security*, vol. 11, no. 3, pp. 137–147, 2012.

[23] M. Finiasz and N. Sendrier, "Security bounds for the design of code-based cryptosystems," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 88–105, Springer, 2009.

[24] D. J. Bernstein, T. Lange, and C. Peters, "Wild mceliece," in *International Workshop on Selected Areas in Cryptography*, pp. 143–158, Springer, 2010.

[25] B.-Y. Yang, *Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29-December 2, 2011, Proceedings*, vol. 7071. Springer, 2011.

[26] T. P. Berger and P. Loidreau, "How to mask the structure of codes for a cryptographic use," *Designs, Codes and Cryptography*, vol. 35, no. 1, pp. 63–79, 2005.

[27] P. Jiménez, F. Thomas, and C. Torras, "3d collision detection: a survey," *Computers & Graphics*, vol. 25, no. 2, pp. 269–285, 2001.

[28] Wikipedia, "Post-quantum cryptography — wikipedia, the free encyclopedia," 2017. [Online; accessed 10-March-2017].

[29] M. Baldi, "The mceliece and niederreiter cryptosystems," in *QC-LDPC Code-Based Cryptography*, pp. 65–89, Springer, 2014.