Public Key Cryptography based on Coding Theory

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung der Grades
Doktor rerum naturalium (Dr. rer. nat.)

von

Dipl.-Math. Raphael Overbeck
geboren in Frankfurt am Main

Referenten: Prof. Dr. Johannes Buchmann
Dr. Nicolas Sendrier, Directeur de Recherche, I.N.R.I.A.

Tag der Einreichung:	16. Januar 2007
Tag der mündlichen Prüfung:	24. April 2007

Darmstadt, 2007
Hochschulkennziffer: D 17

Curriculum Vitae Scientiae

Oct. 1998 – Apr. 2004 Studies of mathematics with minor subjects economics/business economics and computer science at the TU-Darmstadt, Germany.

Oct. 2000 – Apr. 2001 Studies at the Universidad de Salamanca as part of the ERASMUS-program of the EU.

May 2004 – today Grant of the Ph.D.-Program 492 by DFG “Enabeling Technologies for Electronic Commerce” located at the TU-Darmstadt.

Mar. 2006 – May 2006 Hosted by ENSTA and ENS, Paris.

Aug. 2006 – Dec. 2006 Hosted by I.N.R.I.A. Rocquencourt, Le Chesnay Cedex.

Wissenschaftlicher Werdegang

Okt. 1998 – Apr. 2004 Studium der Mathematik mit Nebenfächern BWL/VWL und Informatik an der TU-Darmstadt.

Okt. 2000 – Apr. 2001 Auslandsjahr an der Universidad de Salamanca im Rahmen des ERASMUS Programms der EU.

Mai 2004 – heute Stipendiat des Graduiertenkollegs 492 der DFG “Infrastrukturen für den elektronischen Markt” and der TU-Darmstadt.

März 2006 – Mai 2006 Gastaufenthalte an der ENSTA und ENS, Paris.

Aug. 2006 – Dez. 2006 Gast bei I.N.R.I.A. Rocquencourt, Le Chesnay Cedex.

Publications / Publikationen

[10] D. Engelbert, R. Overbeck, and A. Schmidt. A summary of McEliece-type cryptosystems and their security. *Journal of Mathematical Cryptology*, 1(2):151–199, 2007.

[30] P. Loidreau and R. Overbeck. Decoding rank errors beyond the error-correction capability. In *Proc. of ACCT-10, Zvenigorod*, 2006.

[37] R. Overbeck. Structural attacks for public key cryptosystems based on Gabidulin codes. *Journal of Cryptology.* accepted for publication.

[38] R. Overbeck. A new structural attack for GPT and variants. In *Proc. of Mycrypt 2005*, volume 3715 of *LNCS*, pages 50–63. Springer Verlag, 2005.

[39] R. Overbeck. Extending Gibson's attacks on the GPT cryptosystem. In *Proc. of WCC 2005*, volume 3969 of *LNCS*, pages 178–188. Springer Verlag, 2006.

[40] R. Overbeck. Statistical decoding revisited. In *Proc. of ACISP 2006*, volume 4058 of *LNCS*, pages 283–294. Springer Verlag, 2006.

Für meine Eltern

Acknowledgments

First of all, I wish to thank Prof. Dr. J. Buchmann for promoting this thesis as my supervisor and his most helpful support during my time as a Ph.D. student. His confidence in my abilities and the freedom he gave me encouraged this work and helped a lot to make it a success. I also want to thank my second referee Dr. Nicolas Sendrier, which not only accepted reviewing this work, but as well was so kind to host me at INRIA.

Second, I thank Pierre Loidreau for the most helpful discussions, the excellent collaboration, his invitation to visit him at ENSTA and for establishing the link to the research group at INRIA. I would like to thank my whole research group at the TU-Darmstadt and in particular Prof. Dr. Alexander May, Dr. Ulrich Vollmer and my coauthors for their support and the exchange on many interesting subjects. A thanks to Maike Ritzenhofen for reading part of my thesis and her helpful comments. I acknowledge David Pointcheval from ENS for the possibility to visit his research group. I'd further like to thank the research groups at ENS, ENSTA and I.N.R.I.A. and in special Krzysztof Pietrzak, Andrea Röck, Frédéric Didier, Yann Laigle-Chapuy and Maria Naya Plasencia for their help and the social events in Paris.

Finally I would like to mention my parents, my family and my friends from Darmstadt and Paris for the encouragement, continuous support and distractions which turned my Ph.D. studies into a wonderful experience.

I also want to thank the German Research Foundation, which funded this thesis entirely by their Ph.D.-Program "Enabling Technologies for Electronic Commerce". Thanks as well to Prof. Dr. Alejandro Buchmann and Marion Braun for their support and advice in fundraising this Ph.D. thesis, my participation at most interesting conferences and my stays in Paris.

Preface

In the beginning of cryptographic research the main intention was to secure the communication between two parties against adversaries. Nowadays different needs for cryptographic concepts have moved into the focus as well. The Internet has not only become a resource of information and space for social interaction, but serves as well as a business platform. For example, anyone wishing to order some product paying the bill by bank transfer may choose to do so via Internet. Such kind of business, called e-Business, has to be secured. However, digital communication in general and the Internet in special lack of equivalents to former methods to secure and authenticate transactions like envelopes and handwriting. This causes several security problems, which are quite hard to resolve by early cryptographic means because of scaling problems.

Until the year 1977 two parties wishing to communicate secretly were forced to agree on a common secret in advance or to establish an exclusive communication channel. This concept called symmetric cryptography was mainly realized for military operations and did not find its way to everyday life. However, in 1978 new interesting cryptographic systems were presented, e.g., the RSA and the McEliece scheme. In both schemes it is not longer necessary to agree on a common secret, but only the recipient of the message has to keep a secret. Being assured by a so called public key that the receiver knows a certain secret (for example the factorization of a large number like in the case of RSA) anyone wishing to send her a message may do so without the need and the possibility of recovering the secret. Instead, he might look up the public key in a secured database. Because of the asymmetric aspect of the need to keep a secret, this kind of cryptography is called asymmetric cryptography or (as we will do in the following) public key cryptography. The concept of public key cryptography proved extremely useful to solve problems coming up with the possibilities of the Internet, realizing not only secure communication but for example digital signatures or digital authentication protocols as well. Nevertheless, public key cryptography allows a much wider range of applications, such as key exchange protocols, electronic cash, eVoting or electronic gambling.

The cryptographic context

All public key cryptosystems employed to secure digital communication today are based on the hardness of some problems in number theory. Roughly saying this means that anybody able to break RSA by solving the problem of factorizing large integers or by determining discrete logarithms could get around many security mechanisms of modern digital communication. This would cause a tremendous shock to the worlds economy and security. Even if no one has come up with any method of doing so with feasible resources today, P. Shor showed how to do so in a futuristic scenario. Employing not classical but quantum computers one could use his algorithm [42] to break number theoretic cryptosystems like RSA with feasible time resources. Fortunately, today no one is believed to be in the possession of quantum computers, but research is working under pressure and physicists claim that quantum computers of considerable size could be build within the next decades (see e.g. [9] and [25]).

While provably secure two-party communication can be realized in an idealistic scenario by securing a communication channel using quantum mechanics (compare e.g. [2]), it is still strongly discussed how to do so in multi-party scenarios like the Internet. The threat of quantum computers has reinitialized the research on alternatives for public key cryptography based on number theory. Fortunately, a long known but in comparison to RSA poorly studied alternative exists - the McEliece scheme. Unlike other candidates for public key cryptosystems its concept has resisted all attacks and is easy to realize even with limited computing power, which could turn it very useful for handheld devices with limited power supply. As the McEliece PKC is based on error correcting codes, its security is related to the hardness of the general decoding problem from coding theory, which presumably cannot be solved more efficiently with quantum computers than with normal ones, as it is $\mathcal{NP}$-hard [3]. Furthermore, the fact that the McEliece scheme has variants which meet the cryptographic notion of CCA2-security [26] and that it can even be used to build signatures [8] make it especially interesting. Another strong point of cryptography based on coding theory is that the concept is not limited to public key cryptography. For example one can build (fast) hash functions and random number generators using the principles of coding theory (see [1], [12]). Such concepts can not only be used for symmetric cryptography, but also for building CCA2-secure public key cryptosystems. However, one of the main drawbacks and the reason why the McEliece scheme was never considered for use in real life is the size of the public key.

State of the art in code based cryptography

The public key size of the McEliece cryptosystem is due to the state of the art algorithms for solving the general decoding problem for binary linear codes. While even an improvement of the initial algorithm by McEliece proposed by Lee/Brickell [27] was not able to attack the parameter sets originally proposed, the variant of Sterns algorithm by Canteaut and Chabaud [7] succeeded. As a consequence, parameter sets for the McEliece cryptosystem had to be modified, resulting in key sizes of 88KByte to 130 KByte. These parameter sets have not changed since 1995, and it seems not probable that one can achieve a considerable improvement in the running times of the mentioned algorithms. Nevertheless, there exist different approaches like iterative decoding [13] and statistical decoding [23] to solve the general decoding problem. Even if these concepts do not affect the McEliece scheme today, it is an important question whether these algorithms may be improved and thereby enforce a new change of parameter sets for the McEliece cryptosystem.

While the public key size of the original McEliece scheme will most probably get even larger, reducing the public key size by modifying the original system could lead way to a practicable and accepted cryptographic scheme. The error correcting codes to generate the public key for the McEliece cryptosystem are Goppa codes. Many attempts to replace Goppa codes by different codes as e.g. GRS codes [35] or Reed-Muller codes [43] were proven to be insecure ([44] and [34]), but the security status of other proposals is still unknown.

Abstract

In this thesis we view the statistical decoding algorithm, which tries to solve the general decoding problem as well as the variants of the McEliece cryptosystem based on Gabidulin codes.

The first part of the thesis is dedicated to the general concept of public key cryptography on the basis of coding theory and the security of the underlying problems. Thus after presenting the basic principles, we study the proposal of statistical decoding (which can be seen as a variant of iterative decoding). For a given code, the statistical decoding algorithm precomputes many low weight check vectors and is afterwards able to correct a certain fraction of erroneous codewords in constant time. This can be a great advantage if there are many erroneous messages to decode.

Unfortunately, in the original paper [23] an analysis of the precomputation phase is not included and in experiments given bounds for the space complexity of the statistical decoding algorithm turned out to be too optimistic. We give a robust space complexity analysis of the proposed algorithm and deduce new theoretical bounds. In experiments, these new bounds proof to be more accurate than the previous ones, corroborating some simplifying assumptions in our analysis. Further, we analyze the time complexity of the precomputation phase and draw the conclusion that it is much higher than estimated.

A main flaw of the initial algorithm is the fact that most of the information obtained during the precomputation phase is discarded. We improve the statistical decoding algorithm by taking more information out of the precomputation. This results in an algorithm with better success probability as the initial one. Nevertheless, even this improved algorithm turns out to be slower than a single run of the Canteaut and Chabaud algorithm. We thus conclude that for the McEliece PKC the parameter sets currently proposed remain secure. However, following our approach, further improvement of the statistical algorithm seems to be possible, especially if one could achieve a significant speed-up of the precomputation. Further, the presented methods could be combined with the iterative decoding approach. Therefore, the question if there are better attacks on the McEliece cryptosystem than the existing ones remains open.

The second part of the thesis is dedicated to Gabidulin codes and their application to cryptography like in the GPT proposal from EuroCrypt'91 [18] and its variants. Gabidulin codes use rank distance instead of hamming

distance and thus can be used to correct pattern errors in communication channels. We present a new error correction algorithm for Gabidulin codes, which can be extended to interleaved Gabidulin codes. We show that this extension allows to correct errors in rank metric up to the amount of redundancy in a large number of cases, which is far beyond the initial error correction bound. Consequently our result is analogous to the one of Bleichenbacher, Kiayias and Yung for GRS codes [6].

The question whether Gabidulin codes can be used for cryptographic applications was strongly discussed in the last years, but remained unsolved. The GTP proposal by Gabidulin, Paramonov and Tretjakov is promising, as the general decoding problem in rank metric is more difficult than in hamming metric [24]. Thus, this variant offers more resistance to general decoding attacks than the McEliece scheme while having a much smaller public key size. However, the GPT cryptosystem was attacked by Gibson in '95 and '96 ([20], [22]), who showed how to recover the secret key for initial parameter sets. We gather up the sequently proposed strategies to prevent an attacker from recovering the secret key, which are highly interesting as most of them are applicable to all code based cryptosystems and can (but do not necessarily) lead to secure public key cryptosystems. Further, we analyze the effectiveness of these strategies in the case of GPT under two different aspects: The security of the ciphertexts and the security of the secret keys. First, we show how to take profit of our new error correction algorithm for Gabidulin codes, to attack ciphertexts of cryptosystems using Gabidulin codes in polynomial time. In a second part, we show how to identify the structure of the underlying Gabidulin code in the public key and develop a polynomial time key recovery attack.

Structure of the thesis

The thesis is structured as follows: We give an introduction into basic principles of cryptography on the base of error correcting codes first. Then, we highlight the underlying problems in coding theory and discuss the statistical decoding algorithm. In the last major part, we analyze Gabidulin codes and show why they cannot be used for secret communication. At the end, we make a resume and point out open problems and future fields of research.

Überblick

In dieser Arbeit betrachten wir sowohl den statistischen Fehlerkorrekturalgorithmus zum Lösen des allgemeinen Problems der Fehlerkorrektur als auch die Varianten des Kryptosystems von J.R. McEliece, welche auf Gabidulincodes basieren.

Der erste Teil der Doktorarbeit behandelt die generellen Konzepte codierungstheoriebasierter Kryptographie und die Sicherheit der zugrundeliegenden Probleme. Nach einer Einleitung zu den grundlegenden Begriffen und Prinzipien betrachten wir zunächst den statistischen Fehlerkorrekturalgorithmus [23] (welcher als eine Variante des iterativen Decodierens gesehen werden kann [13]). Für einen gegebenen Code sucht der statistischen Fehlerkorrekturalgorithmus zunächst eine große Menge kleiner Codewörter im dualen Code und ist dann imstande, einen gewissen Anteil fehlerhafter Codewörter in konstanter Zeit zu korrigieren. Dieses kann ein großer Vorteil sein, wenn man viele fehlerhafte Nachrichten korrigieren muß.

Im ursprünglichen Artikel [23] findet sich leider keine Analyse der Phase der Vorberechnungen. Experimente belegen, daß die dort angegebenen Grenzen der Speicherkomplexität zu optimistisch sind. Wir analysieren detailliert die Speicherkomplexität des Algorithmus und bestimmen neue theoretisch fundierte Grenzen. In unseren Experimenten zeigt sich, daß diese neuen Grenzen präziser als die vorherigen sind, welches die vereinfachenden Annahmen bestätigt, welche wir für unsere Analyse benötigen. Weiterhin analysieren wir die Zeitkomplexität der Vorberechnungen und folgern, daß diese wesentlich höher ist, als vom Autor des ursprünglichen Artikels angegeben.

Ein Nachteil der ursprünglichen Version des Algorithmus ist das Vernachlässigen eines Großteils der mit der Vorberechnung gewonnenen Information. Wir zeigen, wie man den Algorithmus verbessern kann, indem man die Information aus den Vorberechnungen besser nutzt. Trotz der Verbesserung ist unsere Variante des statistischen Fehlerkorrekturalgorithmus langsamer als ein einzelner Aufruf des Algorithmus von Canteaut und Chabaud zum Lösen des allgemeinen Problems der Fehlerkorrektur [7]. Folglich schließen wir, daß die aktuell vorgeschlagenen Parameter für das McEliece Kryptosystem weiterhin als sicher anzunehmen sind. Nichtsdestotrotz scheint eine weitere Verbesserung des statistischen Fehlerkorrekturalgorithmus möglich, falls ein signifikantes Beschleunigen der Vorberechnungen erreicht werden kann. Ferner könnten die von uns dargestellten Methoden auf das iterative Decodieren übertragen werden. Deshalb bleibt es eine offene Frage, ob

Angriffe auf das McEliece Kryptosystem existieren, welche besser als die bislang bekannten sind.

Der zweite Teil der Arbeit ist Gabidulincodes und ihrer Anwendung in der Kryptographie wie im GPT Kryptosystem von der EuroCrypt'91 [18] und dessen Varianten gewidmet. Die in Gabidulincodes verwendete Norm ist die Rangnorm und nicht die Hamming Norm, weswegen sie die Korrektur von Fehlermustern ermöglichen. Wir präsentieren einen neuen Fehlerkorrekturalgorithmus, welcher auf "interleaved" Gabidulincodes übertragbar ist. Dort ermöglicht er in den meisten Fällen die Korrektur von Rangdistanzfehlern bis zum Anteil der im Codewort redundanten Information, welches weit über die normale Fehlerkorrekturkapazität hinaus geht. Unser Resultat ist damit analog zu dem von Bleichenbacher, Kiayias und Yung für GRS Codes [6].

Die Frage, ob Gabidulincodes für die Anwendung in der Kryptographie geeignet sind, wurde zwar in den vergangenen Jahren verstärkt diskutiert, blieb aber ungelöst. Der Vorschlag von Gabidulin, Paramonov und Tretjakov ist vielversprechend, da das Problem der Fehlerkorrektur in der Rangnorm schwieriger ist als in der Hamming Norm [24]. Daher bieten solche Codes eine bessere Sicherheit gegenüber allgemeinen Algorithmen zur Fehlerkorrektur, während die Größe des öffentlichen Schlüssels kleiner ist als beim Kryptosystem von McEliece.

Trotz der höheren Sicherheit gegenüber den Angriffen auf die Schlüsseltexte gelang es Gibson in den Jahren '95 und '96 das GPT Kryptosystem mit den ursprünglich vorgeschlagenen Parametern zu brechen, indem er den privaten Schlüssel angriff ([20], [22]). Wir fassen die in der Folge vorgeschlagenen Strategien zusammen, die Angriffe auf den privaten Schlüssel verhindern sollten. Diese Strategien sind sehr interessant, da die meisten bei allen auf Codierungstheorie basierten Kryptosystemen eingesetzt werden können und zu einem sicheren asymmetrischen Kryptosystem führen können (aber dies nicht notwendigerweise tun). Weiterhin analysieren wir diese Strategien auf ihre Wirksamkeit bei Gabidulincodes, und betrachten sowohl die Sicherheit der Schlüsseltexte als auch die der privaten Schlüssel. Zunächst zeigen wir, wie man mit unserem neuen Fehlerkorrekturalgorithmus für Gabidulincodes Schlüsseltexte des GPT Kryptosystems in Polynomialzeit angreifen kann. Danach übertragen wir unsere Überlegungen auf Angriffe auf den privaten Schlüssel. Wir zeigen abschließend, daß die Struktur der Gabidulincodes erlaubt, den privaten Schlüssel in allen Parametersätzen und Varianten des GPT Kryptosystems anzugreifen.

Contents

1 Coding Theory and Cryptography

We give a short introduction into the basic concepts and definitions of coding theory and its application to cryptography. We will limit ourselves to linear codes over finite fields, thus we make the following definition:

Definition 1.1 An $[n,k]$-code $\mathcal{C}$ over a finite field $\mathbb{F}$ is a k-dimensional subvectorspace of the vector space $\mathbb{F}^n$. We call the code $\mathcal{C}$ an $[n,k,d]$ code if $d = \min_{x,y \in \mathcal{C}} \|x - y\|$ for some norm $\| \cdot \|$. The number of positions of an vector $\mathbf{x} \in \mathbb{F}^n$, which differ from zero is called *weight* of $\mathbf{x}$ and corresponds to the Hamming norm.

Any subvectorspace of $\mathcal{C}$ is said to be a subcode of $\mathcal{C}$. If $\mathcal{C}$ is a code over $\mathbb{F}$ and $\mathbb{F}_{\mathsf{SUB}}$ is a subfield of $\mathbb{F}$, then the $\mathbb{F}_{\mathsf{SUB}}$-(subfield) subcode of $\mathcal{C}$ is the code consisting of all words of $\mathcal{C}$, which have only entries in $\mathbb{F}_{\mathsf{SUB}}$. A $\mathbb{F}_{\mathsf{SUB}}$-subfield subcode is a $\mathbb{F}_{\mathsf{SUB}}$-linear code and may be represented as an $[n' \geq n, k' \leq k]$ code over $\mathbb{F}_{\mathsf{SUB}}$. As codes are treated as vector spaces, we will often define them by the matrices related to the code:

Definition 1.2 The matrix $\mathsf{C} \in \mathbb{F}^{k \times n}$ is a *generator matrix* for the $[n,k]$ code $\mathcal{C}$ over $\mathbb{F}$, if the rows of C span $\mathcal{C}$ over $\mathbb{F}$. We write $\mathcal{C} = \langle \mathsf{C} \rangle$. A generator matrix C is said to be in *systematic form*, if its first k columns form the identity matrix. The matrix $\mathsf{H} \in \mathbb{F}^{n \times (n-k)}$ is called *check matrix* for the code $\mathcal{C}$ if it is the right kernel of C. Thus, a word $\mathbf{c}$ is in $\mathcal{C}$ if its *syndrome* $\mathbf{c}\mathsf{H}$ is zero. The code generated by $\mathsf{H}^\top$ is the *dual code* of $\mathcal{C}$ and denoted by $\mathcal{C}^\perp$. If the rows of an $(n-k) \times n$ matrix M span $\mathcal{C}^\perp$ we write $\mathsf{C}^\perp = \mathsf{M}$. With this notation $\mathsf{M}^\top$ is a check matrix of $\mathcal{C}$.

For the ease of notation we will use the following notation throughout the paper: We will identify $\mathbf{x} \in \mathbb{F}^n$ with $(x_1, \cdots, x_n), x_i \in \mathbb{F}$ for $i = 1, \cdots, n$. For any (ordered) subset $\{j_1, \cdots, j_m\} = J \subseteq \{1, \cdots, n\}$ we denote the vector $(x_{j_1}, \cdots, x_{j_m}) \in \mathbb{F}^m$ with $\mathbf{x}_J$. Similarly, we denote by $\mathsf{M}_{\bullet J}$ the submatrix of a $k \times n$ matrix M consisting of the columns corresponding to the indices of J and $\mathsf{M}_{J' \bullet} = \left(\left(\mathsf{M}^\top \right)_{\bullet J'} \right)^\top$ for any (ordered) subset J' of $\{1, \cdots, k\}$. Block matrices will be given in brackets. A set J of columns is said to contain an *information set* of a code $\mathcal{G} = \langle \mathsf{G} \rangle$ if $\mathsf{G}_{\bullet J}$ has full rank.

1.1 GRS and Goppa Codes

An important class of codes are the GRS codes, which are strongly related to the class of Goppa codes used by McEliece to define his cryptosystem. Thus, we briefly introduce them:

Definition 1.3 A GRS code over $\mathbb{F}_{q^m}$ of length n with designed minimum distance $t+1$ is defined by two vectors $a, z \in \mathbb{F}_{q^m}^n$, where $a_i \neq a_j$ for $i \neq j$ and all $z_i \neq 0$. GRS codes are Hamming distance codes. The canonical check matrix of the GRS code is of the form

$$\mathsf{H} = \begin{pmatrix} z_1 a_1^0 & z_1 a_1^1 & \cdots & z_1 a_1^{t-1} \\ z_2 a_2^0 & z_2 a_2^1 & \cdots & z_2 a_2^{t-1} \\ \vdots & & \ddots & \vdots \\ z_n a_n^0 & z_n a_n^1 & \cdots & z_n a_n^{t-1} \end{pmatrix} \in \mathbb{F}_{q^m}^{n \times t}. \tag{1}$$

A $\mathbb{F}_q$-subfield subcode of a GRS code is called an *alternant code* and has dimension $k \geq n - mt$. If for a GRS code, there exists a polynomial $g \in \mathbb{F}_{q^m}[X]$ of degree t, for which $g(a_i) = 1/z_i$, the polynomial is called *Goppa polynomial* and the $\mathbb{F}_q$-subfield subcode is called *Goppa code* (see e.g. [32] or [10]). If there exists an irreducible Goppa polynomial, then the $\mathbb{F}_q$-subfield subcode of $\langle \mathsf{H}^\perp \rangle$ has minimum distance $2 \cdot t + 1$ and is called an *irreducible Goppa code.* For GRS codes, as well as for Goppa codes, there exist algorithms for correcting errors of hamming norm up to half of the minimum distance in $\mathcal{O}(n^2)$ respectively $\mathcal{O}(n \cdot t \cdot m^2)$ binary operations, see e.g. [5] and [10].

1.2 McEliece-like Cryptosystems

Even if R.J. McEliece used binary Goppa codes with irreducible generator polynomials in his original cryptosystem, he led way to a large class of cryptographic systems. Following his ideas, every class of error correcting codes can be used to construct a public key cryptosystem – even if the security status is not known a priori. A pseudo-description of such cryptosystems would be the following:

Definition 1.4 A McEliece-like code based public key cryptosystem consists of three algorithms:

(i) The **key generation** algorithm, which takes in a (set of) security parameter(s) and returns a secret key, which consists of

- a (set of) secret code(s) over a finite field $\mathbb{F}$, which allow(s) to efficiently correct up to t errors according to a certain norm and
- an efficiently invertible transformation, which maps (tuples of) codewords of the secret code(s) to codewords of a public code $\mathcal{G}^{\text{pub}}$.

The public key consists of the matrix G^{pub} generating $\mathcal{G}^{\text{pub}}$ and the number r of errors one can correct in $\mathcal{G}^{\text{pub}}$ knowing the secret key.

(ii) The **encryption** algorithm, which takes in a message $\mathbf{x}$, generates a random vector $\mathbf{e}$ of norm r and returns the ciphertext $\mathbf{c} = \mathbf{x}\mathsf{G}^{\text{pub}} + \mathbf{e}$.

(iii) The **decryption** algorithm takes in the ciphertext $\mathbf{c}$, uses secret transformation to recover the error $\mathbf{e}$ and returns the message $\mathbf{x}$.

In this original variant G^{pub} should not be systematic. Otherwise the first k positions of the ciphertext would have strong correlation with the message. However, one could as well encode the message in the error vector $\mathbf{e}$ and choose a random vector $\mathbf{x}$ to generate the vector $\mathbf{c}$. In this variant one can publish an systematic check matrix of G^{pub} instead of G^{pub} itself, which reduces the public key size. As a consequence, the syndrome $\mathbf{s}$ of $\mathbf{c}$ is sufficient to recover the message and can be treated as ciphertext. This variant introduced by Niederreiter [35] and the original McEliece PKC have equivalent security [29].

The security of code based cryptosystems depends on the difficulty of the following two attacks:

(i) **Structural Attack:** Recover the secret transformation and the description of the secret code(s) from $(\mathsf{G}^{\text{pub}}, r)$.

(ii) **Ciphertext-Only Attack:** Recover the original message from the ciphertext and the public key.

If a code based cryptosystem resists both types of attacks, one can use general or specific conversions to obtain a cryptosystem which meets the CCA2 security notions as studied for example in [26]. Note that for CCA2-secure variants of the McEliece PKC one can choose G^{pub} of systematic form, which reduces the public key size like in the case of the Niederreiter variant.

The difficulty of the ciphertext-only attack is related to the general decoding problem, which we will highlight in section 2. However, there may exist other ways to attack a code based cryptosystem by this kind of attack as we will see in section 3. In general, the difficulty of structural attacks is not related to any classic coding theoretic problem and mainly depends on the class of codes and the secret transformation used. In this section we present the known techniques of how to generate the secret transformation. We assume, that it is sufficient to know a certain matrix $\mathsf{G} \in \mathbb{F}^{k \times n}$ to correct errors of norm at most t in the secret code. This assumption is true for the

McEliece cryptosystem, but as well for most of the proposed variants, see e.g. [41] and [18]. To hide the structure of the secret code (i.e. G), one can apply one or several of the transformations from table 1.1.

(i) **Row Scrambler [33]:** Multiply G by a random invertible matrix $\mathsf{S} \in \mathbb{F}^{k \times k}$ from the left. As $\langle \mathsf{G} \rangle = \langle \mathsf{SG} \rangle$, one can use the known error correction algorithm. Publishing a systematic generator matrix provides the same security against structural attacks as a random S.

(ii) **Column Scrambler / Isometry [33]:** Multiply G by a random invertible matrix $\mathsf{T} \in \mathbb{F}^{n \times n}$ from the right, where T preserves the norm. Obviously one can correct errors of norm up to t in $\langle \mathsf{GT} \rangle$, if G and T are known.

(iii) **Subcode [35]:** Let $0 < l < k$. Multiply G by a random matrix $\mathsf{S} \in \mathbb{F}^{l \times k}$ of full rank from the left. As $\langle \mathsf{SG} \rangle \subseteq \langle \mathsf{G} \rangle$, the known error correction algorithm may be used.

(iv) **Subfield Subcode [33]:** Take the $\mathbb{F}_{\mathrm{SUB}}$-subfield subcode of the secret code for a subfield $\mathbb{F}_{\mathrm{SUB}}$ of $\mathbb{F}$. As before, one can correct errors by the error correcting algorithm for the secret code. However, sometimes one can correct errors of larger norm in the subfield subcode than in the original code, compare definition 1.3 and following.

(v) **Concatenation [43]:** Take the code $\left\langle \left[\begin{array}{c|c} \mathsf{G} & \mathsf{SG} \end{array}\right] \right\rangle$ for an invertible matrix $\mathsf{S} \in \mathbb{F}^{k \times k}$. In Hamming norm, the secret key holder can correct $2t + 1$ errors in this code, as he can correct the errors in the first or the second n columns.

(vi) **Random Redundancy [14]:** Add a number l of random columns at the left side of the matrix G. Errors can be corrected in the last n columns.

(vii) **Artificial Errors [18]:** One can choose to modify the matrix G at a small number of positions. However, the minimum distance of the code obtained might not be the same and if one uses the error correction algorithm of the secret code, one will not longer be able to correct t errors, but a smaller number.

(viii) **Reducible Codes [17]:** Choose some matrix $\mathsf{Y} \in \mathbb{F}^{k \times n}$ and take the code generated by

$$\left[\begin{array}{c|c} \mathsf{G} & \mathsf{0} \\ \hline \mathsf{Y} & \mathsf{G} \end{array}\right].$$

Error correction by the algorithm for the secret code is possible if one corrects errors in sections, beginning from the right. One might extend this strategy by replacing one of the matrices G by a second secret code, compare section 3.3.2.

Table 1.1: Strategies for hiding the structure of a code

Note that it is essential to use certain transformations in combination. We would like to remark two further facts: Using a concatenation may be seen as the combination of the reducible code and the subcode modification. One could as well treat the subfield subcode transformation as a subcode transformation for structural attacks, but we prefer treating them separately. Table 1.2 shows a classification of some code based public key cryptosystems and whether resistance against structural attacks may be achieved for appropriate parameter sets (compare [34], [41] and [47]).

PKC	McEliece	Niederreiter	Modified Niederreiter	Sidelnikov
Class of secret code	GRS [1]	GRS	GRS	Reed-Muller
Row Scrambler	●	●	●	●
Isometry	●	●	●	●
Subcode	-	-	●	-
Subfield Subcode	●	-	-	-
Random Redundancy	-	-	-	-
Artificial Errors	-	-	-	-
Concatenated Code	-	-	-	- / ●
Reducible Codes	-	-	-	-
Security against structural attacks	$\surd$	no	$\surd$	no/no

[1] Goppa codes are subfield subcodes of certain GRS codes.

Table 1.2: Classification of code based cryptosystems

Remark 1.5 (The McEliece PKC) According to our notation a McEliece PKC key pair is generated in the following way: On input of the security parameter $(n = 2^m, t)$, a binary irreducible Goppa polynomial $g \in \mathbb{F}_{q^m}[X]$ of degree t is created and a corresponding $[n, n-t, t+1]$ GRS code $\mathcal{G}_{\mathrm{GRS}}$ over $\mathbb{F}_{2^m}$ of length n with check matrix H according to equation (1) is computed. Afterwards the matrix G generating the $\mathbb{F}_2$-subfield subcode of $\mathcal{G}_{\mathrm{GRS}}$ is determined. Note, that this is an $[n, k \geq n - mt, 2t+1]$ Goppa code. Afterwards an invertible $S \in \mathbb{F}_2^{k \times k}$ and a permutation matrix $T \in \mathbb{F}_2^{n \times n}$ are generated at random. The public McEliece key is

$$(\mathsf{G}^{\mathrm{pub}}, r) = (SGT, t)$$

and the secret key consists for example of H and the secret transformation

$$\begin{array}{rll} \pi: & \mathcal{G}_{\mathrm{GRS}} & \rightarrow \left\langle \mathsf{G}^{\mathrm{pub}} \right\rangle \\ & y & \rightarrow \begin{cases} yP & y \in \mathbb{F}_2^n \\ 0 & \text{otherwise} \end{cases} . \end{array}$$

Note, that we omit storing S in the secret key, as it may be easily be recovered by the secret key holder and is uniquely determined if $\mathsf{G}^{\mathrm{pub}}$ is systematic. In the latter case, S is not needed for decryption. Example parameter sets will be given in table 1.3.

1.3 CFS-like Signature Schemes

A signature scheme can be built using a key pair of a McEliece-like PKC [8] if the ratio of decryptable syndromes to the total number of syndromes is not too small. If a McEliece PKC key pair is used, we call the resulting signature scheme *CFS scheme* [8].

Definition 1.6 A CFS-like code based public key signature scheme consists of three algorithms:

(i) The **key generation** algorithm, which takes in a (set of) security parameter(s) and returns a key pair of a McEliece-like code based public key cryptosystem with a not too small ratio of decryptable syndromes:

$$\frac{\left|\left\{ \mathbf{e} \in \mathbb{F}^n \;\middle|\; \|\mathbf{e}\| \leq r \right\}\right|}{|\{\mathbf{s} \in \mathbb{F}^{n-k}\}|} .$$

(In the case of the CFS scheme, the ratio is about $\frac{1}{r!}$ [8].)

(ii) The **signature** algorithm, which takes in a message $\mathbf{x} \in \mathbb{F}^k$, chooses a random vector $\mathbf{i}$ and tries to decrypt a ciphertext corresponding to the syndrome (deduced from) $\mathbf{s} = h(\left[\ \mathbf{x} \;\middle|\; \mathbf{i}\ \right])$, where h is a hash function. The procedure is repeated until a decryptable syndrome is found. The signature of $\mathbf{x}$ is $(\mathbf{e}, \mathbf{i})$, where $\mathbf{e}$ is the error vector corresponding to $\mathbf{s}$.

(iii) The **verification** algorithm takes in a signature $(\mathbf{e}, \mathbf{i})$ and a message x. The verification algorithm accepts a signature if the syndrome corresponding to $\mathbf{e}$ is (deduced from) $\mathbf{s} = h(\left[\ \mathbf{x} \;\middle|\; \mathbf{i}\ \right])$.

1.4 Performance of Code Based PKCs

Parameter sets, performance and security against state of the art attacks for the McEliece PKC and the CFS signature scheme may be found in table 1.3. Please remember, that the code used will be a binary irreducible Goppa code. As structural attacks for the McEliece PKC are either very slow (see e.g. [21]) or applicable only to a negligible fraction of keys (compare [31]), we mention only the ciphertext-only attack. One can see that the only inconvenience is the public key size, which is much larger than for RSA with the same security level, compare table 1.4.

McEliece system parameters $[n, k, d = 2t + 1]$	Size public key in bytes		Workfactor (binary operations)		
	plain	CCA2-secure	encryption	decryption	best attack [2]
$[1024, 524, 101]$	67,072	32,750	2^{18}	2^{22}	2^{64}
$[2048, 1608, 81]$	411,648	88,440	$2^{20.5}$	2^{23}	2^{98}
$[2048, 1278, 141]$	327,168	123,008	2^{20}	2^{24}	2^{110}
$[2048, 1025, 187]$	262,400	131,072	2^{20}	$2^{24.5}$	2^{106}
$[4096, 2056, 341]$	1,052,672	524,280	2^{22}	$2^{26.5}$	2^{184}
$[2^{16}, 65392, 19]$ [3]	$\approx 535 \cdot 10^6$	1,177,056	2^{31}	2^{27}	$2^{83.7}$

[2] Approximation of general decoding attack from [7], compare section 2.2

[3] This parameter set is used for the CFS signature scheme, with an average signature cost of 2^{37} and a verification cost of 2^{19} binary operations.

Table 1.3: Performance of the McEliece PKC

System	Size public key in bytes	Workfactor (binary operations)		
		encryption	decryption	best attack [4]
RSA 1024-bit Modulus	256	2^{30}	2^{30}	2^{79}
RSA 2048-bit Modulus	512	2^{33}	2^{33}	2^{95}
RSA 4096-bit Modulus	1024	2^{36}	2^{36}	2^{115}

[4] this is the NFS attack for factoring the RSA modulus, see [28].

Table 1.4: Performance of the RSA PKC

2 On the General Decoding Problem

The most promising part about the McEliece cryptosystem and its variants is the fact, that a ciphertext-only attack is related to two well known problems in coding theory, which both are $\mathcal{NP}$-hard in Hamming norm. If a structural attack on a McEliece-like cryptosystem is impossible an attacker is apparently faced with one of the following problems:

Definition 2.1 The *general decoding problem* for linear codes in a norm $\|\cdot\|$ over $\mathbb{F}^n$ is defined as follows:

- Let $\mathsf{C} \in \mathbb{F}^{k\times n}$ define an $[n,k]$ linear code $\mathcal{C}$ over $\mathbb{F}$ and let $\mathbf{y}$ be in $\mathbb{F}^n$.
- Find $\mathbf{x} \in \mathcal{C}$ where $\|\mathbf{y}-\mathbf{x}\|$ is minimal.

Let d be the minimum distance of $\mathcal{C}$ in respect to the given norm and $\mathbf{e}$ be a vector of norm $\leq t := \left\lfloor \frac{d-1}{2} \right\rfloor$ and $\mathbf{x} \in \mathcal{C}$. Then there is a unique solution to the general decoding problem for $\mathbf{y} = \mathbf{x} + \mathbf{e}$. If it is assured that the vector $\mathbf{y}$ (in the general decoding problem) is of the form above, we call the corresponding problem the *bounded distance decoding problem.* The latter problem can be solved as well by solving the problem of finding a vector of norm t in the code generated by

$$\left[\begin{array}{c} \mathsf{C} \\ \hline \mathbf{y} \end{array}\right].$$

The latter problem has a more general version:

Definition 2.2 The *problem of finding weights* (SUBSPACE WEIGHTS) of a linear code is defined as follows:

- Given an $[n,k]$ linear code $\mathcal{C}$ over $\mathbb{F}$ and $w \in \mathbb{N} = \{1,2,3,\cdots\}$.
- Find a $\mathbf{x} \in \mathcal{C}$ satisfying $\|\mathbf{x}\| = w$.

Our hope that we might be able to construct secure cryptosystems based on the problems above is based on a result from [4]:

Theorem 2.3 *The general decoding problem and the problem of finding weights are $\mathcal{NP}$-hard if the norm $\|\cdot\|$ is the Hamming norm.*

However, for the McEliece PKC the problem of recovering the plaintext from a ciphertext has always a unique solution. Consequently, attacking a ciphertext is to solve the bounded distance decoding problem in a permuted Goppa code. This problem is not proven to be $\mathcal{NP}$-hard.

In this section we will present, analyze and improve the *statistical decoding algorithm*, which tries to solve the general decoding problem by solving weak instances of the problem of finding weights. Further, we will analyze the resulting attack on the McEliece cryptosystem.

2.1 Statistical Decoding

This general decoding algorithm was presented by A Kh. Al Jabri in [23]. The idea of statistical decoding is quite similar to the one of iterative decoding (see e.g. [13]) and may be described as follows:

Given an $[n, k, d]$ code $\mathcal{G}$, we first compute a sufficiently large set $\mathcal{H}_w$ of dual vectors of weight w (i.e. an alternative description of $\mathcal{G} = \mathcal{H}_w^\perp$). In the following we assume that $w < n/2$. All observations are analogous for $w > n/2$. Given a word $\mathbf{y} = \mathbf{x} + \mathbf{e}$, where $\mathbf{x} \in \mathcal{G}$ and $\mathrm{wt}(\mathbf{e})$ is small, we take a vector $\mathbf{h} \in \mathcal{H}_w$, where $\mathbf{y}\mathbf{h}^\top \neq 0$. As $\mathbf{x}\mathbf{h}^\top = 0$, the non-zero positions of $\mathbf{h}$ reveals some information about $\mathbf{e}$. (Let e.g. $\mathrm{wt}(\mathbf{e}) = 4$, then either one or three non-zero entries of $\mathbf{e}$ correspond to non-zero entries of $\mathbf{h}$). Collecting the information each of the different vectors $\mathbf{h} \in \mathcal{H}_w$ reveals, we are able to find $\mathbf{e}$ in some cases. In contrary to iterative decoding, the statistical decoding algorithm tries to find a set of error-free positions and not to identify error positions.

There are three major questions regarding this technique, which we will address in the following sections: “How to compute the set $\mathcal{H}_w$?” (section 2.2), “How to combine the information the vectors of $\mathcal{H}_w$ reveal about $\mathbf{e}$?” (the following section) and “What is the probability of identifying $\mathbf{e}$?” (section 2.1.2). In section 2.1.3 we show how to improve the success probability of correct decoding. For now, we present the initial verison:

2.1.1 The Initial Algorithm

Let $\mathcal{H}_w$ be a set of vectors of weight w of the dual space of the $[n, k, 2t + 1]$ linear binary code $\mathcal{G}$ with generator Matrix G. Let $\mathbf{y}$ be the sum of a codeword $\mathbf{u}\mathsf{G} \in \mathcal{G}$ and a error vector $\mathbf{e}$ with weight at most t. A Kh. Al Jabri points out, that for randomly generated codes the probability that a value of 1 appears in the i-th position of $\mathbf{h} \in \mathcal{H}_w$ with $\mathbf{y}\mathbf{h}^T = 1$ depends on i being a erroneous position in the vector $\mathbf{y}$. We say that we have an *odd error detection* in i if $\mathbf{y}\mathbf{h}^T = 1$ and $\mathbf{h}_i = 1$. Under that condition, let p_w^+ be the probability that i is a erroneous position and q_w^+ be the probability that

i is a non-erroneous position. We can compute these probabilities as

$$p_w^+ = \frac{\sum_{m \text{ odd}}^{m \le t} \binom{n-t}{w-m}\binom{t-1}{m-1}}{\sum_{m \text{ odd}}^{m \le t} \binom{t}{m}\binom{n-t}{w-m}}, \; q_w^+ = \frac{\sum_{m \text{ odd}}^{m \le t} \binom{n-t-1}{w-m-1}\binom{t}{m}}{\sum_{m \text{ odd}}^{m \le t} \binom{t}{m}\binom{n-t}{w-m}}.$$

Since $w < n/2$ the inequation $p_w^+ > q_w^+$ holds, although for large w the difference is small. We define $v_{\mathbf{y},w}^+ := \sum_{\mathbf{h} \in \mathcal{H}_w} \left(\mathbf{y}\mathbf{h}^T \mod 2\right)$. Then, for $i \in \{1, \cdots, n\}$ an (non-)error position the random variable

$$\frac{1}{v_{\mathbf{y},w}^+} \sum_{\mathbf{h} \in \mathcal{H}_w} \left(\mathbf{y}\mathbf{h}^T \mod 2\right) \mathbf{h}_i$$

is the relative frequency estimate for p_w^+ (q_w^+ respectively). Its variance is $\left(\sigma_w^+\right)^2 = p_w^+(p_w^+ - 1)/v_{\mathbf{y},w}^+$. Thus, we can recover $\mathbf{u}$ using algorithm 2.1.1 if $\mathcal{H}_w$ is chosen in a way so that we can distinguish between p_w^+ and q_w^+.

Algorithm 2.1.1 StatDec

```
Input: $\mathcal{H}_w$, $\mathbf{y}$.
Output: $\mathbf{u}$, the information vector.

$\mathbf{v} = \sum_{\mathbf{h} \in \mathcal{H}_w} \left(\mathbf{y}\mathbf{h}^\top \mod 2\right) \mathbf{h} \in \mathbb{Z}^n$.

choose $I = \{$positions of the $k$ smallest entries of $\mathbf{v}\}$ s.t. $\mathsf{G}_{\cdot I}$ is invertible.

$\mathbf{u} = \mathbf{y}_I \mathsf{G}_{\cdot I}^{-1}$
```

Al Jabri claims, that precomputing a set $\mathcal{H}_w$ with

$$|\mathcal{H}_w| \approx 625 \cdot 10^{-6} \cdot p_w^+ \left(1 - p_w^+\right) \epsilon^{-2} \tag{2}$$

vectors is sufficient for correct decoding [23]. However, Al Jabri's initial analysis of the size of $\mathcal{H}_w$ needed for error correction seems to be too optimistic (compare as well [13]).

The work factor for algorithm 2.1.1 is

$$\mathcal{O}\left(n \cdot |\mathcal{H}_w| + 2k^3 + kn\right)$$

binary operations having computed the set $\mathcal{H}_w$ in advance. The author of [23] claims that the latter can be done e.g. by the methods of [7], which is to be doubted (compare section 2.2,[13] and [40]). Computing the set $\mathcal{H}_w$ is solving problem 2.2, which is a $\mathcal{NP}$-hard problem in general. In addition, a set $\mathcal{H}_w$ of the desired size will not even exist if w is chosen too small. Goppa codes, as BCH codes and GRS codes have a weight distribution "close" to the expected weight distribution of random code, which is the binomial distribution [23]. Consequently, we have the bound

$$|\mathcal{H}_w| \leq \binom{n}{w} 2^{-k} \tag{3}$$

if we want to decode e.g. a random code or a Goppa code. We will come back to this problem in section 2.2, but first we want to analyze the success probability of StatDec.

2.1.2 The Success Probability of Statistical Decoding

The first point of critique on StatDec is its success probability. In our experiments for small parameter sets we had difficulties to correct errors with a set $\mathcal{H}_w$ of size given in equation (2). It seems, that the set has to be about 2^{13} times larger than claimed by Al Jabri to allow correct decoding in most cases. We give a brief example: For a $[2^6, 40, 9]$ Goppa code (or a $[2^6, 40, 9]$ random code), Al Jabri's estimation for $\mathcal{H}_{17}$ is $|\mathcal{H}_{17}| = 1 \leq \binom{64}{17} 2^{-40} \approx 2^{10}$. However, one vector of the dual code can not be sufficient for correct decoding in most cases. Therefore we want to take a closer look at the success probability of statistical decoding. Later we show how to improve StatDec and give examples.

In the following, we assume, that every set $\mathcal{H}_w$ consists of random vectors of weight w. If the vectors in $\mathcal{H}_w$ are somehow related, the probability for finding the correct error vector changes.

We return to the notations previously used. On input $\mathcal{H}_w$ and $\mathbf{y}$, StatDec returns the correct error vector iff for some δ with $-p_w^+ < \delta < 1 - p_w^+$ the following two conditions hold:

(i) For every error position i:

$$\mathbf{v}_i > (p_w^+ + \delta) v_{\mathbf{y},w}^+.$$

(ii) There are at least k non-error positions j, such that

$$\mathbf{v}_j < (p_w^+ + \delta) v_{\mathbf{y},w}^+.$$

We may assume, that $v_{\mathbf{y},w}^+ \approx \frac{1}{2} |\mathcal{H}_w|$, and thus the probability, that a certain δ fulfills the first condition is smaller than

$$\mathcal{P} := \Phi\left(-\delta / \sigma_w^+\right)^t = \Phi\left(-\delta \sqrt{\frac{\frac{1}{2}|\mathcal{H}_w|}{p_w^+(p_w^+ - 1)}}\right)^t, \tag{4}$$

where Φ refers to the distribution function of the standardized normal distribution. Thus, we have to choose

$$2\left(\Phi^{-1}\left(\mathcal{P}^{1/t}\right)\right)^2 p_w^+(1-p_w^+)\delta^{-2} \le |\mathcal{H}_w| \le \binom{n}{w}2^{-k}. \tag{5}$$

Assume $k \approx (n-t)/2$, then it is very probable, that k values $\mathbf{v}_j$ for non error positions j will be below their mean value $p_w^+ v_{\mathbf{y},w}^+$. Thus, if there exists an δ for a given ciphertext y, such that the two conditions above are fulfilled, then it will probably be at least $q_w^+ - p_w^+$. Since $\Phi^{-1}(0.95) = 1.65$ we conclude, that with a set of size

$$|\mathcal{H}_w| \approx 5.4 p_w^+(1-p_w^+)\frac{1}{(p_w^+ - q_w^+)^2}. \tag{6}$$

we can correct errors with a probability about 0.95^t. Note, that this number is a factor 2^{13} larger than the one given by Al Jabri (compare as well [13]). We expect that with a set of size given in equation (2) we could correct errors with a probability about $1/2^t$, only.

2.1.3 An Improved Version of Statistical Decoding

To improve the probability of correct error correction, we want to include *even error detection.* With the notation of this section we have an even error detection if $\mathbf{yh}^T = 0$ and $\mathbf{h}_i = 1$. Let p_w^- be the probability that i is a erroneous position and q_w^- be the probability that i is a non-erroneous position in the case of an even error detection. These probabilities can be computed as follows:

$$p_w^- = \frac{\sum_{2\le m \text{ even}}^{m\le t}\binom{n-t}{w-m}\binom{t-1}{m-1}}{\sum_{m \text{ even}}^{m\le t}\binom{t}{m}\binom{n-t}{w-m}}, \quad q_w^- = \frac{\sum_{m \text{ even}}^{m\le t}\binom{n-t-1}{w-m-1}\binom{t}{m}}{\sum_{m \text{ even}}^{m\le t}\binom{t}{m}\binom{n-t}{w-m}}.$$

We define $v_{\mathbf{y},w}^- := \sum_{\mathbf{h}\in\mathcal{H}_w}\left(1 - \mathbf{yh}^T \mod 2\right)$. Then, for an (non-)error position i the value

$$\frac{1}{v_{\mathbf{y},w}^-}\sum_{\mathbf{h}\in\mathcal{H}_w}\left(1 - \mathbf{yh}^T \mod 2\right)\mathbf{h}_i$$

is the relative frequency estimate for p_w^- (q_w^- respectively). We observe, that if $p_w^+ > q_w^+$, then $p_w^- < q_w^-$.

For all possible weights, the relative frequency estimates of p_w^+ and p_w^- are approximately normal distributed if $|\mathcal{H}_w|$ is large enough. Therefore we can use the standard transformation, s.t. all the relative frequency estimates are $\mathcal{N}(0,1)$ distributed. It follows, that one can sum the scaled relative frequency estimates obtained by several sets containing dual vectors of different weights. As a consequence, we consider $\mathcal{H}$ as the set of all dual vectors of weight w satisfying $b \leq w \leq B < n/2$, i.e. $\mathcal{H} = \bigcup_{w=b}^{B} \mathcal{H}_w$. All in all, we get the modified algorithm 2.1.2. With the notation of StatDec+: If i is an error position, then for all $\mathbf{v}$, $(\mathbf{v})_i$ has mean value 0. For an implementation one should omit the previous computation of σ_w^+ and σ_w^-. and compute these values while computing $\mathbf{v}_w$.

Algorithm 2.1.2 StatDec+

```
Input: $\mathcal{H} = \bigcup_{w=b}^{B} \mathcal{H}_w$, $\mathbf{y}$.
Output: $\mathbf{u}$, the information vector.

for $w = b$ to $B$ do
  $(\sigma_w^+)^2 = p_w^+ \cdot (1 - p_w^+) \cdot v_{\mathbf{y},w}^+$.
  $(\sigma_w^-)^2 = p_w^- \cdot (1 - p_w^-) \cdot v_{\mathbf{y},w}^-$.

$\mathbf{1} = (1, 1, \cdots, 1) \in \{0,1\}^n$.
for $w = b$ to $B$ do
  $\mathbf{v}_w = \sum_{\mathbf{h} \in \mathcal{H}_w} \big(\mathbf{y}\mathbf{h}^\top \mod 2\big) (\mathbf{h} - p_w^+ \mathbf{1})/\sigma_w^+ \in \mathbb{R}^n$.
  $\mathbf{v}_{w+B} = -\sum_{\mathbf{h} \in \mathcal{H}_w} \big(1 - \mathbf{y}\mathbf{h}^\top \mod 2\big) (\mathbf{h} - p_w^- \mathbf{1})/\sigma_w^- \in \mathbb{R}^n$.

for all binary combinations $\mathbf{v}$ of the different $\mathbf{v}_l$ do
  choose $I$ = {positions of the $k$ smalles entries of $\mathbf{v}$} s.t. $\mathsf{G}_{\cdot I}$ is invertible.
  $\mathbf{u} = \mathbf{y}_I \mathsf{G}_{\cdot I}^{-1}$
  if weight($\mathbf{u}\mathsf{G} \oplus \mathbf{y}$) $\leq t$ then
    return $\mathbf{u} = \mathbf{u}$
```

Let us assume, that the different relative frequency estimates are independent. We define $\mathbf{v} = \sum_{w=b}^{B} e_w \mathbf{v}_w + \sum_{w=b}^{B} e_{w+B} \mathbf{v}_{w+B}$, where each $e_i \in \{0,1\}$. Then for an error position j, $(\mathbf{v})_j$ is normal distributed with mean value 0 and variance σ^2 equal to the number of $e_w \neq 0$. If j is a non-error position, then $(\mathbf{v})_j$ is normal distributed with mean value

$$E := \sum_{w=b}^{B} e_w \left(\frac{q_w^+ - p_w^+}{\sigma_w^+} v_{\mathbf{y},w}^+ \right) + \sum_{w=b}^{B} e_{w+B} \left(\frac{p_w^- - q_w^-}{\sigma_w^-} v_{\mathbf{y},w}^- \right) < 0$$

and variance

$$S^2 = \sum_{w=b}^{B} w_w \left(\frac{q_w^+(1-q_w^+)}{\left(\sigma_w^+\right)^2} v_{\mathbf{y},w}^+ \right) + \sum_{w=b}^{B} w_{w+B} \left(\frac{q_w^-(1-q_w^-)}{\left(\sigma_w^-\right)^2} v_{\mathbf{y},w}^- \right)$$

In most cases we will have $2v_{\mathbf{y},w}^+ \approx 2v_{\mathbf{y},w}^- \approx |\mathcal{H}_w|$. To distinguish between error and non-error positions by $\mathbf{v}$, we get the following conditions: There exists an $\delta \in \mathbb{R}$ such, that for every error position i the inequation $\mathbf{v}_i > \delta$ holds and there are at least k non-error positions j, such that $\mathbf{v}_j < \delta$. The probability, that a certain δ fulfills this conditions is smaller than $\Phi\left(-\delta/\sigma\right)^t$. Again, we expect, that the condition $\delta \geq E$ has to be true in most cases, and thus we get

$$\mathcal{P} \approx \Phi\left(\frac{1}{\sigma}\left(\sum_{w=b}^{B} e_w \sqrt{\frac{\left(p_w^+ - q_w^+\right)^2 |\mathcal{H}_w|}{2p_w^+(1-p_w^+)}} + \sum_{w=b}^{B} e_{w+B} \sqrt{\frac{\left(q_w^- - p_w^-\right)^2 |\mathcal{H}_w|}{2p_w^-(1-p_w^-)}}\right)\right)^t$$

as a suitable estimate for the probability of correct decoding with StatDec+. However we are not able to prove, that the different relative frequency estimates for p_w^+ and q_w^+ are independent. Nevertheless, for an implementation it seems recommendable, to start with the vectors $\mathbf{v}$ where $|\{e_i \neq 0\}|$ is large.

2.1.4 Experimental Results

We made several experiments for codes of small length. As expected, the proposed variant StatDec+ of the initial algorithm allows error correction in a significant larger number of cases than StatDec, especially when the size of the sets $\mathcal{H}_w$ is small. Further, it seems recommendable to include sets $\mathcal{H}_w$ with small w, even if their size is smaller than desired (e.g. up to a factor 4).

In the following we present three examples of our experiments. Note that for all our examples the bound for $|\mathcal{H}_w|$ given by equation (2) is useless, as it is smaller than 0. Further, the precomputation to find the sets $\mathcal{H}_w$ was quite time-consuming and an exhaustive search in some cases. The time needed to perform the precomputation for StatDec+ is the same as for StatDec.

In our first example we considered a $[2^6, 40, 9]$ Goppa code. For this code the relative frequency estimates and the desired sizes of each $\mathcal{H}_w$ resulting from equation (6) are given in table 2.5. We computed a set

w	p_w^+	q_w^+	p_w^-	q_w^-	$\lvert\mathcal{H}_w\rvert$
16	0.295	0.248	0.210	0.263	1433
17	0.302	0.263	0.232	0.268	2160
18	0.311	0.280	0.254	0.284	3393

Table 2.5: Correcting errors of weight 4 in a $[64, 40]$ code.

$\mathcal{H} = \{\mathcal{H}_{16}, \mathcal{H}_{17}, \mathcal{H}_{18}\}$, where each of the sets $\mathcal{H}_w$ consisted of 100 random vectors. With StatDec+ we were able to correct errors of weight 4 in 93.2% of the cases. With the original algorithm, called with each set $\mathcal{H}_w$, correct error correction was possible in 17.5% of the cases, only.

In the second example, we looked at the same code as in the first example, but chose each $\mathcal{H}_w$ to be the set of all vectors of weight w. For our particular Goppa code, we got: $|\mathcal{H}_{16}| = 345$, $|\mathcal{H}_{17}| = 1234$ and $|\mathcal{H}_{18}| = 3149$. In this case, error correction was possible with StatDec and StatDec+ in all cases. An correct error correction with StatDec would not have been possible in all cases, if only one of the sets $\mathcal{H}_w$ would have been used.

w	p_w^+	q_w^+	p_w^-	q_w^-	$\lvert\mathcal{H}_w\rvert$	StatDec success rate
8	0.183	0.119	0.082	0.129	562	95.0%
9	0.189	0.136	0.102	0.145	835	79.4%
10	0.196	0.152	0.122	0.160	1283	73.8%

Table 2.6: Correcting errors of weight 6 in a $[64, 22]$ code.

In our last example, we looked at a $[2^6, 22, 13]$ random code. The values for the relative frequency estimates and the sizes of $\mathcal{H}_w$ resulting from equation (6) are given by table 2.6. The expected success probability of StatDec is $\approx 0.95^6 = 73.5\%$ for each set $\mathcal{H}_w$. In this case we were able to compute the desired sets in reasonable time. Again, we made 1000 attempts to correct errors of weight 6. The experimented success probability for StatDec with such sets is larger than expected, compare table 2.6. With StatDec+ we were able to correct all errors, whereas with StatDec we would have been able to correct them in 99.2% of the cases.

2.2 On the Problem of Finding Weights

Al Jabri proposes to use a variant of Sterns algorithm to solve the problem of finding weights, i.e. to compute $\mathcal{H}_w$. J. Stern designed his algorithm to find a (unique) shortest codeword of a binary linear code.

We recall the original algorithm of Stern [45], which tries to find a vector of low weight w. Let H be the check matrix of the code G. Given the parameters p and l, successively choose two disjoint sets of $p < k/2$ code positions $\mathcal{I}_1$ and $\mathcal{I}_2$ at random. If $\mathsf{H}_{\mathcal{I}_0 \bullet}$ with $\mathcal{I}_0 := \{1, \cdots, n\} \setminus (\mathcal{I}_1 \cup \mathcal{I}_2)$ is singular the algorithm fails at this point and is started anew. Else, a set $\mathcal{J}$ of l columns of H is chosen. We may assume without loss of generality, that $\mathcal{I}_1 = \{n-k+1, \cdots, n-k/2\}$, $\mathcal{I}_2 = \{n-k/2+1, \cdots, n\}$ and $\mathcal{J} = \{1, \cdots, l\}$. By Gaussian elimination we can assume that the check matrix is of the form

$$\mathsf{H}^\top = \left(\begin{array}{c|c} \mathsf{Id}_{n-k} & \begin{array}{c|c} \mathsf{Z}_1 & \mathsf{Z}_2 \\ \hline \multicolumn{2}{c}{\mathsf{B}} \end{array} \end{array} \right),$$

where Z_1 and Z_2 are $l \times k/2$ matrices, and B is a $(n-k-l) \times k$ matrix. For all pairs of vectors $(\mathbf{e}_1, \mathbf{e}_2) \in (\ \{0,1\}^{k/2}\ )^2$ where $\mathrm{wt}(\mathbf{e}_1) = \mathrm{wt}(\mathbf{e}_2) = p$ we check whether $\mathbf{e}_1 \mathsf{Z}_1^\top = \mathbf{e}_2 \mathsf{Z}_2^\top$. If the condition is fulfilled, we compute the unique vector $\mathbf{e}_0 \in \{0,1\}^{n-k}$, such that $\left[\ \mathbf{e}_0 \mid \mathbf{e}_1 \mid \mathbf{e}_2\ \right] \mathsf{H} = \mathbf{0}$. Each vector $\mathbf{e} = \left[\ \mathbf{e}_0 \mid \mathbf{e}_1 \mid \mathbf{e}_2\ \right]$ is a candidate for a short codeword. One can observe, that the fist l entries of $\mathbf{e}$ are zeros and thus the weight of $\mathbf{e}$ is smaller than $n-k-l+2p$. If none of the constructed vectors $\mathbf{e}$ is of the desired weight, then the algorithm fails. The success probability of one iteration of the algorithm is

$$\mathcal{P}_{p,l,w} = \frac{\binom{n-w}{k/2-p}\binom{w}{p}\binom{n-w-k/2-p}{k/2-p}\binom{w-p}{p}\binom{n-k-(w-2p)}{l}}{\binom{n}{k/2}\binom{n-k/2}{k/2}\binom{n-k}{l}}$$

in the case of a unique code word $\mathbf{e}'$ of weight w.

To improve the performance of Sterns algorithm, one can view its dual variant – depending on the ratio of k/n – and try to avoid the costly Gaussian elimination by choosing $\mathcal{I}_1$ and $\mathcal{I}_2$ iteratively and not at random. This method was introduced and analyzed by Canteaut and Chabaud, compare [7]. The success probability of the algorithm for finding the shortest codeword is to be modeled by a Markov chain in that case. We omit details and just take the result, that the work factor for one iteration becomes

$$\Omega_{p,l} = \left(\frac{1}{2} n(n-k) + 2l \binom{k/2}{p} (p-1) + (n-k-l)(2p-1) \binom{k/2}{p}^2 \frac{1}{2^l} \right).$$

The work factor of the resulting algorithm is lower bounded by $\mathcal{P}_{p,l,w}^{-1}\Omega_{p,l}$ and can be approximated by

$$\mathcal{O}(n^3)2^{-t\log_2(1-k/n)},$$

if t is small and k/n is not too close to one [41]. Since for the McEliece cryptosystem $n = 2^m$ and $k = n - tm$, N. Sendrier concludes, that the maximum degree of security for the McEliece cryptosystem against the general decoding attack from [7] is obtained for an information rate $k/n \approx 1 - 1/\exp(1)$. This would lead e.g. to the choice of $m = 11$ and $t = 70$ for the McEliece cryptosystem, compare table 1.3.

In the case of statistical decoding we use the mentioned algorithm from [7] not to find a single lowest weight code word, but several code words of a certain weight w. If there are several code words of weight w, the work factor decreases by a factor equal to the number of such code words. As the expected number of vectors of weight w is given by the binomial distribution, we get the expected workfactor to compute a set $\mathcal{H}_w$ of vectors of weight w as

$$\mathcal{W}_{p,l,w} = \frac{2^k}{\binom{n}{w}} \frac{\Omega_{p,l}}{\mathcal{P}_{p,l,w}} \cdot \sum_{i=0}^{|\mathcal{H}_w|-1} \left(1 - \frac{i \cdot 2^k}{\binom{n}{w}}\right)^{-1}. \tag{7}$$

If one wants to compute a set $\mathcal{H}$, which serves as an input for the StatDec+, we expect, that every execution of a single round of the algorithm returns

$$\sum_{w=b}^{B} \frac{2^k}{\binom{n}{w}} \mathcal{P}_{p,l,w}^{-1}$$

vectors of weight w satisfying $b \leq w \leq B$. However, using the algorithm from [7] might not always be the best choice when trying to find multiple words of any given weight, even if we did not find a better way to do so.

Unfortunately, we were not able to find an example parameter set, where the precomputation required for StatDec could be performed in less time than the one needs for a single call of Canteaut's and Chabaud's general decoding algorithm for the same code.

2.3 Attacking the McEliece PKC by statistical decoding

To attack the McEliece PKC with parameters $m = 10$ and $t = 50$ with statistical decoding, Al Jabri claims that computing a set $\mathcal{H}_w$ consisting of 2^{38} vectors is sufficient. Unfortunately Al Jabri does not name w, but we are quite sure, that he referred to the set $\mathcal{H}_{133}$. However, equation (4)

McEliece parameters $[2^m, k, d = 2t + 1]$	w	$\lvert p_w^+ - q_w^+ \rvert$	$\lvert \mathcal{H}_w \rvert$	Workfactor	
				StatDec	finding $\mathcal{H}_w$
$[1024, 524, 101]$	137	$0.2 \cdot 10^{-7}$	2^{51}	2^{61}	2^{152}
$[1024, 524, 101]$	153	$0.21 \cdot 10^{-8}$	2^{58}	2^{68}	2^{138}
$[2048, 1278, 141]$	363	$0.41 \cdot 10^{-14}$	2^{96}	2^{107}	2^{609}
$[65536, 65392, 9]$	32000	$0.17 \cdot 10^{-13}$	2^{93}	2^{109}	$\gg 2^{131}$

Table 2.7: StatDec for example parameter sets

implies, that the probability of correct decoding is about 2^{-50} in that case. A decoding attempt with StatDec takes 2^{48} binary operations for this input. Consequently, one would expect, that it would take approximately 2^{98} binary operations, before an attack on one of 2^{50} given ciphertexts is successful.

We have shown that an attacker would need a set $\mathcal{H}_{137}$ consisting of approximately 2^{51} vectors to attack ciphertext of the McEliece PKC with parameters $m = 10$ and $t = 50$. Even storing a set of this size seems impossible nowadays and the work factor for a single decoding attempt would be larger than 2^{61}, which is not much faster than the general decoding algorithm of Canteaut and Chabaud [7]. However, it takes at least 2^{152} binary operations to compute the set $\mathcal{H}_{137}$ with the algorithm proposed by Canteaut and Chabaud. For this parameter set, one iteration for $l = 19$ and $p = 2$ of the algorithm requires about 2^{24} binary operations. Most of the vectors returned by the algorithm will be of weight 241. For each one of 2^{-17} iterations, we will get only one of those vectors. Thus, after performing 2^{80} Operations, one will still have computed less than 2^{39} vectors of weight 241. With a range of $114 \leq w \leq 241$, we will not have enough vectors of the dual space to attack the McEliece cryptosystem. Thus, it is not possible to attack the McEliece cryptosystem with StatDec or StatDec+.

The situation for the signature scheme CFS is the same: Any set, that would allow correct decoding in a non-negligible fraction of the cases is to big to be stored efficiently and it is infeasible to perform the precomputation (compare Table 2.7). Further, even after the precomputation, StatDec has no or no significant advantage over the algorithm by Canteaut and Chabaud, compare table 1.3.

Thus, we obtain the same result as the authors of [13] which conclude, that like in the case of StatDec+, for iterative decoding a smaller set $\mathcal{H}_w$ as for the initial StatDec is sufficient. However, like for statistical decoding, the size of $\mathcal{H}_w$ needed for iterative decoding is far too large to be computed in feasible time.

3 Rank distance codes and Cryptography

In 1985, E.M. Gabidulin proposed a new class of codes, called Gabidulin codes [15]. These codes can correct *rank distance* errors (also called pattern errors) efficiently, which in general is harder than correcting Hamming distance errors [24]. At Eurocrypt'91, Gabidulin, Paramonov and Tretjakov proposed a cryptosystem based on rank distance codes (GPT, [18]). Because of its better resistance against general decoding attacks, smaller key sizes were proposed for GPT than for the McEliece PKC.

However, the GPT cryptosystem was subject to structural attacks in '95 and '96 ([20] and [22]), which work only for small parameter sets since they have exponential time complexity. To better hide the structure of Gabidulin codes, in 2003 ([17], [14]) several variants of GPT were proposed. The modifications proposed for GPT are highly interesting, as most of them are applicable to all code based cryptosystems and intuitively the analogous to the modifications proposed for the basic multivariate schemes. As noted at PKC'06, the subcode modification for example leads to secure instances when using GRS codes, but does not offer advantage over the McEliece PKC [47]. For Gabidulin codes however, as proven by the author, all variants of GPT are strongly connected to each other [39].

In this section, we focus on Gabidulin codes first. After presenting the basic principles of Gabidulin codes, we develop a new algorithm for correcting errors beyond half of the minimum distance in interleaved Gabidulin codes. We show, that our new algorithm leads way to new attacks on cryptosystems build from Gabidulin codes, allowing to attack ciphertexts as well as the secret keys. As all our attacks run in cubic time, they are not even much slower than the original decryption procedure, which takes quadratic time. Further, we are able to show that our attack can easily be extended to all parameter sets of all variants of GPT. Especially the resulting attack on ciphertexts is interesting, as our attack evites solving any of the problems on which the security of the GPT-like cryptosystems was meant to rely.

3.1 Rank Distance Codes

Rank distance codes were presented by Gabidulin in 1985. They are linear codes over the finite field $\mathbb{F}_{q^m}$ for q (power of a) prime and $m \in \mathbb{N}$. As their name suggests they use a special concept of distance. In this section we recall the basic facts and give the notation used in the following sections.

Definition 3.1 Let $\mathbf{x} = (x_1, \cdots, x_n) \in \mathbb{F}_{q^m}^n$ and $b_1, \cdots, b_m$ a basis of $\mathbb{F}_{q^m}$

over $\mathbb{F}_q$. We can write $x_i = \sum_{j=1}^{m} x_{ij} b_j$ for each $i = 1, \cdots, n$ with $x_{ij} \in \mathbb{F}_q$. The *rank norm* $\| \cdot \|_q$ is defined as follows:

$$\|\mathbf{x}\|_q := \operatorname{rank}\left((x_{ij})_{1 \leq i \leq n,\ 1 \leq j \leq m} \right) .$$

The rank norm of a vector $\mathbf{x} \in \mathbb{F}_{q^m}^n$ is uniquely determined (independent of the choice of basis) and induces a metric, called *rank distance.* Note, that the Hamming distance of two vectors is never smaller than their rank distance. Further, if $\mathsf{T} \in \mathbb{F}_q^{n \times n}$ is an invertible matrix, then $\|\mathbf{x} \cdot \mathsf{T}\|_q = \|\mathbf{x}\|_q$. Thus, every invertible matrix over $\mathbb{F}_q$ is an isometry for the rank norm. In the following we will consider each basis of a field $\mathbb{F}_{q^m}$ over some subfield $\mathbb{F}$ will be a normal basis, i.e. $b_i = b^{q^i}, i = 1, \cdots, m$ for some element $b \in \mathbb{F}$.

In [24] Ourivski and Johansson presented two algorithms which solve the general decoding problem in $\mathcal{O}\left((k + \frac{d-1}{2})^3 (\frac{d-1}{2})^3 q^{(d-3)(m-(d-1)/2)/2}\right)$, respectively $\mathcal{O}\left((m\frac{d-1}{2})^3 q^{(d-3)(k+1)/2}\right)$ operations over $\mathbb{F}_q$ for $[n, k, d]$ rank distance codes over $\mathbb{F}_{q^m}$. However, there exists a class of rank distance codes, named *Gabidulin codes*, for which an efficient decoding algorithm exists [18]. We will define these codes by their generator matrix. For ease of notation we introduce the operator λ_f, which maps a matrix $\mathsf{M} = (m_{ij})$ to a blockmatrix:

$$\begin{array}{rccc} \lambda_f : & \mathbb{F}_{q^m}^{m \times n} & \longrightarrow & \mathbb{F}_{q^m}^{m(f+1) \times n} \\ & \mathsf{M} & \longmapsto & \begin{bmatrix} \mathsf{M} \\ \mathsf{M}^{[q]} \\ \vdots \\ \mathsf{M}^{[q^f]} \end{bmatrix} , \end{array} \tag{8}$$

where $\mathsf{M}^{[x]} := (m_{ij}^x)$.

Definition 3.2 Let $\mathbf{g} \in \mathbb{F}_{q^m}^n$ be a vector s.t. the components $g_i, i = 1, \cdots, n$ are linearly independent over $\mathbb{F}_q$. This implies that $n \leq m$. The $[n, k]$ Gabidulin code $\mathcal{G}$ is the rank distance code with generator matrix

$$\mathsf{G} = \lambda_{k-1}(\mathbf{g}) . \tag{9}$$

An $[n, k]$ Gabidulin code $\mathcal{G}$ has minimum distance $d = n - k + 1$ and corrects errors of rank $\lfloor \frac{n-k}{2} \rfloor$. The vector $\mathbf{g}$ is said to be a *generator vector* of the Gabidulin code $\mathcal{G}$ (It is not unique, as all vectors $a\mathbf{g}$ with $0 \neq a \in \mathbb{F}_{q^m}$ are generator vectors of $\mathcal{G}$). Further, if $\mathsf{T} \in \mathbb{F}_q^{n \times n}$ is an invertible matrix, then $\mathsf{G} \cdot \mathsf{T}$ is the generator matrix of the Gabidulin code with generator vector $\mathbf{g}\mathsf{T}$. A error correction algorithm based on the “right Euclidian division algorithm" runs in $\mathcal{O}\left(d^3 + dn\right)$ operations over $\mathbb{F}_{q^m}$ for $[n, k, d]$ Gabidulin codes [18]. The property, that a matrix G generates a Gabidulin code is invariant under the operator $\Lambda_f(\mathsf{M})$:

Lemma 3.3 *If* G *is a generator matrix of an* $[n, k]$ *Gabidulin code* $\mathcal{G}$ *with* $k < n$*, then* $\Lambda_f(\mathsf{G}^{\text{pub}})$ *is a generator matrix of the Gabidulin code with the same generator vector as* $\mathcal{G}$ *and dimension* $\min\{n, k + f\}$*.*

Another nice property of Gabidulin codes is, that the dual code of an $[n, k]$ Gabidulin code is an $[n, n - k]$ Gabidulin code (see [18]):

Lemma 3.4 *Let* $\mathcal{G}$ *be an* $[n, k]$ *Gabidulin code over* $\mathbb{F}_{q^m}$ *with generator vector* g*. Then* $\mathcal{G}$ *has a check matrix of the form*

$$\mathsf{H}^\top = \lambda_{n-k-1}\left(\mathbf{h}^{[1/q^{n-k-1}]}\right)^\top \in \mathbb{F}_{q^m}^{n-k\times n} .$$

Further, the vector $\mathbf{h}$ *is uniquely determined by* $\mathbf{g}$ *(independent from* k*) up to a scalar factor* $\gamma \in \mathbb{F}_{q^m} \setminus \{0\}$*. We will call* $\mathbf{h}$ *a check vector of* $\mathcal{G}$*.*

Proof. It is sufficient to prove, that if some $\mathbf{h}$ is in the dual space of the $[n, k]$ Gabidulin code $\mathcal{G}_k$ with generator vector $\mathbf{g}$, then $\mathbf{h}^{[1/q]}$ is in the dual space of the $[n, k - 1]$ Gabidulin code $\mathcal{G}_{k-1}$ with generator vector $\mathbf{g}$:

$$\mathbf{h} \in \mathcal{G}_k^\perp \Leftrightarrow \forall_{i\in\{0,\cdots,k-1\}} \sum_{j=1}^{n} \mathbf{h}_j \mathbf{g}_j^{q^i} = 0 \Rightarrow \forall_{i\in\{1,\cdots,k-1\}} \sum_{j=1}^{n} \mathbf{h}_j^{1/q} \mathbf{g}_j^{q^{i-1}} = 0.$$

■

If $\mathbb{F}$ is a subfield of $\mathbb{F}_{q^m}$, the $\mathbb{F}$-(subfield) subcode of $\mathcal{G}$ has check matrix $\lambda_{n-k-1}(\ \mathbf{h}_\mathbb{F}\ )$, where the matrix $\mathbf{h}_\mathbb{F}$ represents the check vector $\mathbf{h}$ of $\mathcal{G}$ by a normal basis over $\mathbb{F}$ [16].

For any selection J of $\widetilde{n} \geq k$ columns of the generator matrix G, the matrix $G_{\bullet J}$ defines an $[\widetilde{n}, k]$ Gabidulin code. For arbitrary vectors the selection of certain columns allows to prove the following fact:

Lemma 3.5 *If* $\mathbf{e} \in \mathbb{F}_{q^m}^n$ *is of rank norm* t*, then there exists an invertible matrix* $\mathsf{T} \in \mathbb{F}_q^{n\times n}$*, such that* $\mathbf{e}\mathsf{T}^{-1}$ *is zero at the positions* $t + 1, \cdots, n$*. It follows that* $\lambda_{k-1}(\mathbf{e})$ *has rank* $\min\{k, t\}$*.*

3.2 Interleaved Gabidulin Codes

In this section we introduce the general concept of interleaved codes and the application to Gabidulin codes. To do so, we define the mapping ϕ, where $b_1, \cdots, b_s$ is a basis of $\mathbb{F}_{q^{ms}}$ over $\mathbb{F}_{q^m}$:

$$\begin{array}{lll} \phi: & \mathbb{F}_{q^{sm}} & \to \mathbb{F}_{q^m}^s \ , \\ & x = \sum_{i=1}^s x_i b_i, \ \text{where } x_i \in \mathbb{F}_{q^m} & \mapsto (x_1, \cdots, x_s)^\top \ . \end{array}$$

Starting from an $[n, k, d]$ code $\mathcal{G}$ over $\mathbb{F}_{q^m}$ with generator matrix G, we build a $\mathbb{F}_{q^m}$-linear code over $\mathbb{F}_{q^{sm}}$ in the following way:

Definition 3.6 Let G be the generator matrix of an $[n, k, d]$ code $\mathcal{G}$ over $\mathbb{F}_{q^m}$, then the interleaved code $\mathcal{G}_\mathrm{I}$ consists of all vectors $\mathbf{y} \in \mathbb{F}_{q^{ms}}^n$, such that

$$\left[\ \phi(\mathbf{y}_1) \ \middle|\ \cdots \ \middle|\ \phi(\mathbf{y}_n)\ \right] = \left[\ \phi(\mathbf{x}_1) \ \middle|\ \cdots \ \middle|\ \phi(\mathbf{x}_k)\ \right] \mathsf{G} \tag{10}$$

for some vector $\mathbf{x} \in \mathbb{F}_{q^{ms}}^k$. The parameter s is called the amount of interleaving. It is easy to see, that the minimum distance between two vectors of $\mathcal{G}_\mathrm{I}$ is at least d.

Let $\mathbf{z} = \mathbf{y} + \mathbf{e}$ with $\mathbf{y} \in \mathcal{G}_\mathrm{I}$ and an error $\mathbf{e} \in \mathbb{F}_{q^{ms}}^n$ of norm $\|\mathbf{e}\| \leq (d-1)/2$. To correct the error $\mathbf{e}$ in $\mathbf{z}$ one can apply the error correction algorithm for $\mathcal{G}$ to each

$$(\phi(\mathbf{z}_1)_i, \cdots, \phi(\mathbf{z}_n)_i) = (\phi(\mathbf{y}_1)_i, \cdots, \phi(\mathbf{y}_n)_i) + (\phi(\mathbf{e}_1)_i, \cdots, \phi(\mathbf{e}_n)_i) \ ,$$

$i = 1, \cdots, s$ and recover $\mathbf{y}$ afterwards.

3.2.1 Correcting Rank Errors Beyond Minimum Distance

In [30] the authors present two algorithms for correcting rank errors beyond minimum distance in interleaved Gabidulin codes. Here, we present the probabilistic algorithm proposed by the author, the success probability of which depends on the input, only.

Let $\mathcal{G}_\mathrm{I}$ the interleaved code build from an $[n, k]$ Gabidulin code $\mathcal{G}$ over $\mathbb{F}_{q^m}$ with amount of interleaving s. Further, let $\mathbf{z} = \mathbf{y} + \mathbf{e}$, where $\mathbf{y} \in \mathcal{G}_\mathrm{I}$ and $\mathbf{e} \in \mathbb{F}_{q^{ms}}^n$ is of rank norm $t < n - k$. For error correction we compute the vector space

$$\mathcal{H}_\mathbf{e} := \left[\frac{\lambda_{n-t-2}(\mathbf{g})}{\lambda_{n-k-t-1}(\phi(\mathbf{z}))}\right]^\perp = \left[\frac{\lambda_{n-t-2}(\mathbf{g})}{\lambda_{n-k-t-1}(\phi(\mathbf{e}))}\right]^\perp . \tag{11}$$

This vector space has a very useful property:

Lemma 3.7 *If $\lambda_{n-k-t-1}(\phi(\mathbf{e}))$ has rank t, then every vector $\mathbf{h_e} \in \mathcal{H}_e \setminus \{\mathbf{0}\}$ has rank norm $n-t$. Further, if for an invertible matrix T the first t positions of $\mathbf{h_e}\mathsf{T}^\top$ are zero, then the last $n-t$ positions of $\mathbf{e}\mathsf{T}^{-1}$ are zero.*

Proof. Fist note, that $\lambda_{n-t-2}(\mathbf{g})$ has rank $n-t-1$. Thus, if $\lambda_{n-k-t-1}(\phi(\mathbf{e}))$ has rank t, then $\mathcal{H}_\mathbf{e}$ has dimension one. Let $\mathsf{T}_\mathbf{e}$ be a isometry such that the last $n-t$ columns of $\mathbf{e}\mathsf{T}_\mathbf{e}^{-1}$ are zero and $\mathbf{h_e}$ be the vector spanning $\mathcal{H}_\mathbf{e}$. Then the first t entries of $\mathbf{h_e}\mathsf{T}_\mathbf{e}^\top$ are zero and $(\mathbf{h_e}\mathsf{T}_\mathbf{e}^\top)_{\{t+1,\cdots,n\}}$ is the check vector of an $[n-t, n-t-1]$ Gabidulin code. Thus, $\mathbf{h_e}$ has rank norm $n-t$. Second, if T is of the above form, then $\lambda_{n-k-t-1}(\phi(\mathbf{e}))\mathsf{T}^{-1} \cdot \mathsf{T}\mathbf{h}_\mathbf{e}^\top = 0$. Let $\widehat{t}$ be the rank of $\lambda_{n-k-t-1}(\mathbf{e}\mathsf{T}^{-1})_{\bullet\{t+1,\cdots,n\}}$ over $\mathbb{F}_q$ and $\mathbb{F}_{q^m}$ (compare lemma 3.5). Then, there exists an invertible matrix $\widehat{\mathsf{T}} \in \mathbb{F}_q^{(n-t)\times(n-t)}$, such that the matrix $\lambda_{n-k-t-1}(\mathbf{e}\mathsf{T}^{-1})_{\bullet\{t+1,\cdots,n\}}\widehat{\mathsf{T}}^{-1}$ is zero at the $n-t-\widehat{t}$ rightmost positions. Consequently, the vector $(\mathbf{h_e}\mathsf{T}^\top)_{\bullet\{n-t,\cdots,n\}}\widehat{\mathsf{T}}^\top$ of rank norm $n-t$ is in its dual, which can only be if $\widehat{t} = 0$. ■

Thus, from each $\mathbf{h_e} \in \mathcal{H}_\mathbf{e}$ we can derive an invertible matrix $\mathsf{T} \in \mathbb{F}_q^{n\times n}$ such that the $n-t$ leftmost columns of $\mathbf{e}\mathsf{T}^{-1}$ are zero. Computing T can be done in $\mathcal{O}(n^3)$ operations as it requires only solving some linear equations, compare [39]. This is sufficient for error correction:

Lemma 3.8 *Let $\mathsf{T} \in \mathbb{F}_q^{n\times n}$ be such that the last $n-t$ positions of $\mathbf{e}\mathsf{T}$ are zero. Then, for the vector $\mathbf{x} \in \mathbb{F}_{q^{ms}}$ defining $\mathbf{y}$ by equation (10) the following equation holds:*

$$\left[\ \phi(\mathbf{x}_1) \mid \cdots \mid \phi(\mathbf{x}_k)\ \right](\mathsf{GT})_{\bullet\{t+1,\cdots,n\}} = \left[\ \phi((\mathbf{z}\mathsf{T})_{t+1}) \mid \cdots \mid \phi((\mathbf{z}\mathsf{T})_n)\ \right].$$

Proof. The lemma follows from the fact that $(\mathbf{e}\mathsf{T})_{\{t+1,\cdots,n\}} = 0$ yields that $(\mathbf{z}\mathsf{T})_{\{t+1,\cdots,n\}} = (\mathbf{y}\mathsf{T})_{\{t+1,\cdots,n\}}$. ■

The error correction procedure is summarized in algorithm 3.2.1. As algorithm 3.2.1 only requires solving some linear equations it has runtime $\mathcal{O}(n^3)$ operations. The correctness follows from lemmas 3.7 and 3.8 if the rank of $\lambda_{n-k-t-1}(\phi(\mathbf{e}))$ is t. It remains to determine the rank of $\lambda_{n-k-t-1}(\phi(\mathbf{e}))$. After [46] (compare lemma 3.13) the rank of $\phi(\mathbf{e})$ is s with probability

$$\prod_{i=0}^{s-1} \frac{\left(q^{mt} - q^{mi}\right)}{q^{mt}} \geq \left(\frac{q^{mt} - q^{ms}}{q^{mt}}\right)^s.$$

Algorithm 3.2.1 Decoding Interleaved Gabidulin codes

```
Input: $\mathbf{z} = \mathbf{y} + \mathbf{e}$ with $\mathbf{y} \in \mathcal{G}_{\mathrm{I}}$ and $\mathbf{e}$ of norm $t < n - k$.
Output: $\mathbf{y} \in \mathcal{G}_{\mathrm{I}}$ or failure.

Compute $\mathcal{H}_{\mathbf{e}}$ as in equation (11).
if $\dim \mathcal{H}_{\mathbf{e}} > 1$ then
  return failure
else
  Compute a vector $\mathbf{h}$ of rank norm $\geq k$ in $\mathcal{H}_{\mathbf{e}}$.
Compute an invertible matrix $\mathsf{T} \in \mathbb{F}_q^{n \times n}$, such that $(\mathbf{h}\mathsf{T}^\top)_{\{1,\cdots,t\}} = 0$.
Set $\mathsf{T} = \mathsf{T}^{-1}$.
Solve the equation from lemma 3.8 to compute the vector $\mathbf{y}$.
return $\mathbf{y}$
```

As we will see later (theorem 3.11), it follows that with probability

$$\geq \left(1 - \frac{4}{q^m}\right)\left(\frac{q^{mt} - q^{ms}}{q^{mt}}\right)^s \tag{12}$$

the matrix $\lambda_{n-k-t-1}(\phi(\mathbf{e}))$ has rank $\min\{(n-t-k)s, t\}$. We conclude:

Theorem 3.9 *Let $\mathcal{G}_{\mathrm{I}}$ be the interleaved code build from the $[n, k]$ Gabidulin code $\mathcal{G}$ over $\mathbb{F}_{q^m}$, where s is the amount of interleaving. If $s \ll (n - k)$, correction errors in $\mathcal{G}_{\mathrm{I}}$ of rank up to*

$$t = \frac{s}{s+1}(n-k)$$

with algorithm 3.2.1 succeeds with probability given in equation (12).

A possible parameter set would be $q = 2$, $m = n = 24$, $k = 10$ and $s = 6$. In this setting, the correction of errors of rank 12 with algorithm 3.2.1 fails in less than one of 2^{22} cases.

Remark 3.10 If $s = 1$, then $\mathcal{G}_{\mathrm{I}} = \mathcal{G}$ is an $[n, k, d]$ Gabidulin code. If further the conditions of the above theorem are true, algorithm 3.2.1 never fails as the rank of $\lambda_{n-k-t-1}(\phi(\mathbf{e}))$ is $t \leq (n-k)/2$, see lemma 3.5. Thus, algorithm 3.2.1 can be used to correct errors of rank up to the standard bound $(d-1)/2$ in cubic time.

3.2.2 The Probability of Correct Decoding

To determine the probability of correct decoding we need to determine a upper bound for the probability, that the rank of $\lambda_{n-k-t-1}(\phi(\mathbf{e}))$ has rank smaller than t. This probability is lower than the one that the rank of $\lambda_{\lfloor \frac{t-1}{s} \rfloor}(\mathsf{M})$ is smaller than t if M is a random $s \times t$ matrix over $\mathbb{F}_{q^m}$ with full rank over $\mathbb{F}_q$. For easier notation we write $\|\mathsf{M}\|_q$ if we refer to the rank of M over $\mathbb{F}_q$, and analogous $\|\mathsf{M}\|_{q^m}$ for the rank of M over $\mathbb{F}_{q^m}$. Our goal is to prove the following theorem:

Theorem 3.11 *Let* M *be a random* $s \times t$ *matrix over* $\mathbb{F}_{q^m}$ *with* $s \leq t \leq m$. *Then*

$$\mathcal{P}rob\Big(\ \|\lambda_f(\mathsf{M})\|_{q^m} < t \quad \Big| \quad \|\mathsf{M}\|_q = t \ \Big) \leq \frac{4}{q^m},$$

where $f = \lfloor \frac{t-1}{s} \rfloor$.

As a direct consequence, we can bound the following probability, too:

Lemma 3.12 *Let* M *be a random* $s \times t$ *matrix over* $\mathbb{F}_{q^m}$ *with* $s \leq t \leq m$. *Then for all* k

$$\mathcal{P}rob\big(\ \|\lambda_k(\mathsf{M})\|_{q^m} < \min\{s(k+1), t\} \quad \big| \quad \|\mathsf{M}\|_q = t \ \big) \leq \frac{4}{q^m}.$$

Before we are going to prove the theorem, we would like to recall the following fact:

Lemma 3.13 *The fraction of all* $m \times n$ *matrices over* $\mathbb{F}_q$ *which have full rank is larger than* 0.288.

Proof. Considering all $m \times n$ matrices over $\mathbb{F}_q$, the fraction of the matrices of rank k is

$$\frac{1}{q^{mn}} \prod_{j=0}^{k-1} \frac{\left(q^m - q^j\right)\left(q^n - q^j\right)}{\left(q^k - q^j\right)},$$

see [46]. With the results from [11] we get the following bound for the fraction of $i \times i$ matrices of full rank:

$$\frac{1}{q^{i^2}} \cdot \prod_{j=0}^{i-1} \left(q^i - q^j\right) = \prod_{j=1}^{i} \left(1 - q^{-j}\right) \geq 0.288788,$$

which we will approximate by 1/4 in the following. ■

Unfortunately, we are not able to count the number of matrices M with $\left\|\lambda_f\left(\mathsf{M}\right)\right\|_{q^m} < t$ directly. Thus, we have to rewrite the condition:

Lemma 3.14 *For any $s \times t$ matrix M over $\mathbb{F}_{q^m}$ with $s \leq t \leq m$ and $\left\|\mathsf{M}\right\|_q = t$, the following two statements are equivalent:*

$$\left\|\lambda_f\left(\mathsf{M}\right)\right\|_{q^m} < t \tag{13}$$

$$\Longleftrightarrow$$

$$\exists_{\mathbf{h}\in\mathbb{F}_{q^m}^n, \left\|\mathbf{h}\right\|_q > f+1} \forall_{\alpha\in\mathbb{F}_{q^m}^\times} \left(\lambda_f\left(\alpha\mathbf{h}\right)\cdot\mathsf{M}^\top = 0\right). \tag{14}$$

Proof. The proof for (14) ⇒ (13) is quite simple and based on the following observation for two vectors $\mathbf{h}, \mathbf{m} \in \mathbb{F}_{q^m}^n$:

$$\left(\mathbf{h}\mathbf{m}^\top = 0 \wedge \mathbf{h}^{[q]}\mathbf{m}^\top = 0\right) \Rightarrow \left(\mathbf{h}^{[q]}(\mathbf{m}^{[q]})^\top = 0 \wedge \mathbf{h}^{[q]}\mathbf{m}^\top = 0\right).$$

From that, it follows immediately, that if a $\mathbf{h}$ exists, such that (14) is fulfilled, then $\mathbf{h}^{[q^k]}$ is in the dual space of $\lambda_k\left(\mathsf{M}\right)$ for all $0 \leq k \leq f$.
To proof (14) ⇐ (13), we observe first, that it follows from (13), that there exists an $\mathbf{h} \in \mathbb{F}_{q^m}^n$ in the dual space of $\lambda_f\left(\mathsf{M}\right)$. Consequently all $\alpha\mathbf{h}$ with $\alpha \in \mathbb{F}_{q^m}^\times$ are in that space, too. Using the fact, that

$$\left(\mathbf{m}\mathbf{h}^\top = 0 \wedge \mathbf{m}^{[q]}\mathbf{h}^\top = 0\right) \Rightarrow \left(\mathbf{m}\mathbf{h}^\top = 0 \wedge \mathbf{h}^{[1/q]}\mathbf{m}^\top = 0\right),$$

we conclude, that

$$(13) \;\Rightarrow \exists_{\mathbf{h}\in\mathbb{F}_{q^m}} \forall_{\alpha\in\mathbb{F}_{q^m}^\times} \left(\lambda_f\left(\alpha\mathbf{h}\right)\cdot\mathsf{M}^\top = 0\right).$$

It remains to show, that such an $\mathbf{h}$ of norm $\left\|\mathbf{h}\right\|_q > f+1$ exists. If $\left\|\mathbf{h}\right\|_q = r \leq f+1$, then there exists an invertible matrix $\mathsf{T} \in \mathbb{F}_q^{t\times t}$, such that the matrix $\lambda_f\left(\mathbf{h}\right)\mathsf{T}$ has non-zero entries in the r rightmost columns, only. Since the submatrix of $\lambda_f\left(\mathbf{h}\right)\mathsf{T}$ consisting of the r rightmost columns has full rank, the r rightmost columns of $\mathsf{T}^{-1}\mathsf{M}^\top$ have only zero entries, which is a contradiction to the premise that $\left\|\mathsf{M}\right\|_q = t$. We conclude, that $\mathbf{h}$ has rank norm $> f+1$, which proves the lemma. ■

With this modified statement, we are able to give an upper bound of the number of matrices M, where $\left\|\lambda_f\left(\mathsf{M}\right)\right\|_{q^m} < t$. By this, we can finally prove the theorem:

Proof. (of Theorem 3.11) To compute the probability of (14) we first determine the probability, that for a fixed $\mathbf{h} \in \mathbb{F}_{q^m}^n$ with $\|\mathbf{h}\|_q > f+1$ we have

$$\Big(\lambda_f\left(\alpha\mathbf{h}\right) \cdot \mathsf{M}^\top = 0\Big).$$

if M is a random $s \times t$ matrix with $\|\mathsf{M}\|_q = t$. As the rank of $\lambda_f\left(\mathbf{h}\right)$ over $\mathbb{F}_{q^m}$ is exact $f+1$, there exist at most $(q^m)^{s(t-f-1)}$ possibilities to choose M, such that the condition above is fulfilled. After lemma 3.13, there are more than $\frac{1}{4} \cdot (q^m)^{st}$ possibilities to choose a random $s \times t$ matrix M with $\|\mathsf{M}\|_q = t$. Thus, for a fixed $\mathbf{h}$, the probability, that the condition above is fulfilled for a random $s \times t$ matrix M of full rank over $\mathbb{F}_q$ is smaller than

$$4 \cdot (q^m)^{-s(f+1)}.$$

Now we determine the number of different vector spaces $\langle\, \lambda_f\left(\mathbf{h}\right) \rangle$ defined by some $\mathbf{h} \in \mathbb{F}_{q^m}^n$, where the norm of $\mathbf{h}$ is not to small. This number is smaller than

$$(q^{mt}-1)/(q^m-1) \approx q^{m(t-1)},$$

as $\mathbf{h} \neq 0$ and all $\alpha\mathbf{h}$ with $\alpha \in \mathbb{F}_{q^m}^\times$ define the same vector space. Thus, the probability, that the condition (14) is fulfilled for a random matrix M is smaller than the sum of the probabilities for the fixed $\mathbf{h}$ over the possible different vector spaces they define. As by lemma 3.14 we have (14) $\Leftrightarrow$ (13), we get the following bound:

$$\begin{aligned} \mathcal{P}rob\Big(\ \|\lambda_f\left(\mathsf{M}\right)\|_{q^m} < t \quad \Big| \quad \|\mathsf{M}\|_q = t\ \Big) \ &\leq\ q^{m(t-1)} \cdot 4 \cdot (q^m)^{-s(f+1)}. \\ &\leq\ 4 \cdot q^{-m}, \end{aligned}$$

which proves the theorem. ■

Note, that theorem 3.11 gives an estimation of the number of subspace subcodes of $[n,k]$ Gabidulin codes over $\mathbb{F}_{q^m}$, which do not have minimal dimension. For $n = m$ it was already proven in [16], that this number is 0.

Lemma 3.15 *Let $\mathcal{G}$ be an $[n,k]$ Gabidulin code over $\mathbb{F}_{q^N}$, where $N = ms > n$. Then, the probability that the $\mathbb{F}_{q^m}$-subcode of $\mathcal{G}$ has dimension larger than* $\min\{0, n-s(n-k)\}$ *is smaller than $4/q^m$.*

Proof. The $\mathbb{F}_{q^m}$-subcode of $\mathcal{G}$ has a check matrix of the form $\lambda_{n-k-1}\left(\mathsf{M}\right)$, where the i-th column of $\mathsf{M} \in \mathbb{F}_{q^m}^{s \times n}$ represents the i-th entry of the generator vector of $\mathcal{G}$ over $\mathbb{F}_{q^m}$, e.g. by employing ϕ. Thus, the lemma follows directly from theorem 3.11. ■

3.3 The GPT Cryptosystem

The GPT cryptosystem was first presented at Eurocrypt'91 by Gabidulin, Paramonov and Tretjakov [18]. We present the more general version developed by the author in 2005 (GGPT, see [39]) first, which may be used to describe the original GPT cryptosystem as well as the variant with column scrambler (CS-GPT, [14]) from 2003. Afterwards we give descriptions of the latter ones according to the results of [39].

- **System Parameters:** q, m, n, k, t and $s \in \mathbb{N}$, where $k < n \leq m$, $t < n - k - 1$ and $s \leq \min\{t, k\}$
- **Key Generation:** First generate the following matrices:

  $\mathsf{G} \in \mathbb{F}_{q^m}^{k\times n}$ generator matrix of an $[n, k, d]$ Gabidulin code,
  $\mathsf{X} \in \mathbb{F}_{q^m}^{k\times t}$ random matrix of rank s over $\mathbb{F}_{q^m}$ and rank t over $\mathbb{F}_q$,
  $\mathsf{S} \in \mathbb{F}_{q^m}^{k\times k}$ random, non-singular matrix (the row scrambler) and
  $\mathsf{T} \in \mathbb{F}_{q}^{n\times n}$ random, non-singular matrix (the column scrambler).

  Then compute the $k \times n$ matrix

  $$\begin{aligned} \mathsf{G}^{\text{pub}} &= \mathsf{S}\left(\left[\ \mathsf{X} \mid \mathsf{0}\ \right] + \mathsf{G}\right) T \\ &= \mathsf{S}\left[\ \mathsf{G}_{\bullet\{1,\cdots,t\}} + \mathsf{X} \mid \mathsf{G}_{\bullet\{t+1,\cdots,n\}}\ \right] \mathsf{T} \in \mathbb{F}_{q^m}^{k\times n}\ , \end{aligned} \tag{15}$$

  where $\mathsf{0}$ denotes the $k \times (n - t)$ zero matrix. Choose $r = \frac{n-k-t}{2}$. Further let $\mathcal{D}_\mathcal{G}$ be an efficient decoding algorithm for the Gabidulin code $\mathcal{G}$ generated by the matrix $\mathsf{G}_{\bullet\{t+1,\cdots,n\}}$.
- **Public Key:** $\left(\mathsf{G}^{\text{pub}}, r\right)$
- **Private Key:** $(\mathcal{D}_\mathcal{G}, \mathsf{S}, \mathsf{T})$ or $(\mathsf{G}, \mathsf{S}, \mathsf{T})$ where G is of the form in (9).
- **Encryption:** To encode a plaintext $\mathbf{x} \in \mathbb{F}_{q^m}^k$ choose a vector $\mathbf{z} \in \mathbb{F}_{q^m}^n$ of rank norm r at random and compute the ciphertext

  $$\mathbf{y} = \mathbf{x}\mathsf{G}^{\text{pub}} + \mathbf{z}\ .$$
- **Decryption:** To decode a ciphertext $\mathbf{y}$ apply the decoding algorithm $\mathcal{D}_\mathcal{G}$ for $\mathcal{G}$ to $\mathbf{y}' = \left(\mathbf{c}\mathsf{T}^{-1}\right)_{\{t+1,\cdots,n\}}$. As T is an invertible matrix over $\mathbb{F}_q$, the rank norm of a vector does not change if it is multiplied with T^{-1}. Thus $\mathbf{y}'$ has at most rank distance $\frac{n-k-t}{2}$ to $\mathcal{G}$ and we obtain the codeword

  $$\mathbf{x}\mathsf{S}\mathsf{G}_{\bullet\{t+1,\cdots,n\}} = \mathcal{D}_\mathcal{G}\left(\mathbf{y}'\right)\ .$$

  Now, we can compute the plaintext $\mathbf{x}$.

Parameters				size public key	WF general
m	k	t	s	in bytes	decoding
48	10	16	3	$2,880$	2^{134}
48	16	18	4	$1,608$	2^{124}
64	8	40	1	$3,584$	2^{87}

Table 3.8: Previously proposed parameters for GPT / GGPT

The *distortion matrix* X may be seen as a matrix of artificial errors (compare table 1.1) and is essential to mask the structure of G. Otherwise, a check vector $\mathbf{h}$ may be revealed from SGT by computing $\mathbf{h} = \lambda_{n-k-1}(\mathsf{SGT})^{\perp}$. Example parameter sets may be found in table 3.8, where $n = m$ and $q = 2$ (WF = operations over $\mathbb{F}_q$) .

We would like to draw the readers attention to the fact, that G^{pub} may be viewed as an erroneous codeword of an interleaved code with interleaving degree k. Here, however, the error $[\mathsf{X}|0]\,\mathsf{T}$ does not have full rank (i.e. $\min\{t, k\}$) over $\mathbb{F}_{q^m}$, but s. An attacker trying to recover the original secret key is thus faced to correct an considerable error in a code he does not know. Nevertheless, it is not necessary to know the secret key to recover the plaintext from a ciphertext, as we will show in the following.

3.3.1 Simple Variants of GPT

The original approach of the GPT cryptosystem was to choose the parameters r and t such that $r = \frac{n-k}{2} - t$. If one does so, the legitimate user may recover $\mathbf{x}\mathsf{SGT}$ by applying the error correction algorithm for $\langle \mathsf{GT} \rangle$ (which is a Gabidulin code, too) to the ciphertext $\mathbf{y}$. An alternative description of the public generator matrix would be $\mathsf{G}^{\text{pub}} = \mathsf{S}\,(\,\widetilde{\mathsf{G}} + \widetilde{\mathsf{X}}\,)$, where $\widetilde{\mathsf{G}} = \mathsf{GT}$ and $\widetilde{\mathsf{X}} = \left[\ \mathsf{X} \ \middle|\ 0\ \right] \mathsf{T}$.

Another variantis the CS-GPT: G, X and S are chosen s.t. all entries of G^{pub} are in a subfield $\mathbb{F}_{\text{SUB}}$ of $\mathbb{F}_{q^m}$ (this is not a subfield subcode version). In this case, the plaintext and random errors $\mathbf{z}$ are chosen from $\mathbb{F}_{\text{SUB}}$ as well. This saves space when storing the public key. The most common instances of CS-GPT are the ones, where the public generator matrix may be written as

$$\mathsf{G}^{\text{pub}} = \mathsf{S} \left[\ \mathsf{Y} \ \middle|\ \widetilde{\mathsf{G}}\ \right] \cdot \mathsf{T} \in \mathbb{F}_{\text{SUB}}^{k \times n},$$

where $\mathsf{Y} \in \mathbb{F}_{\text{SUB}}^{k \times t}$ is arbitrary, $\widetilde{\mathsf{G}} \in \mathbb{F}_{\text{SUB}}^{k \times (n-t)}$ defines an $[n-t, k]$ Gabidulin code and S is in $\mathbb{F}_{\text{SUB}}^{k \times k}$. The latter can be interpreted as adding random redundancy to the secret code.

Following the guidelines from table 1.1 one could try to publish only a subcode of the GGPT public key [36]. In this case, one might even try to omit the column scrambler and the distortion matrix, which leads to the Niederreiter GTP.

A further intuitive variant would be to use subfield subcodes for GPT. However, as already mentioned in section 3.1, the subfield subcode of a Gabidulin code can be defined by the operator λ_f, too. It follows, that any subfield subcode version of GPT would be very similar to the other GPT variants, compare [37]. However, for them, a different error correction procedure is possible. This approach is generalized in the concept of reducible rank codes.

3.3.2 The RRC-GPT Variant

In [17], the authors proposed to substitute the underlying code by a reducible code (RRC-GPT). Unlike all other variants, the RRC-GPT is an extension of the concept of GPT, whose instances may not be expressed by the means of GGPT.

Definition 3.16 Let $\mathcal{G}_i = \langle G_i \rangle$, $i = 1, \cdots, w$ be a family of $[n_i, k_i, d_i]$ codes over $\mathbb{F}_{q^m}$. Then the (linear) code $\mathcal{G}$ given by the generator matrix of the form

$$\mathsf{G} = \begin{bmatrix} \mathsf{G}_1 & \mathsf{0} & \cdots & \mathsf{0} \\ \mathsf{Y}_{21} & \mathsf{G}_2 & \cdots & \mathsf{0} \\ \vdots & & \ddots & \vdots \\ \mathsf{Y}_{w1} & \mathsf{Y}_{w2} & \cdots & \mathsf{G}_w \end{bmatrix} \in \mathbb{F}_{q^m}^{\sum k_i \times \sum n_i}$$

for some matrices $\mathsf{Y}_{ij} \in \mathbb{F}_{q^m}^{k_i \times n_j}$ is called a reducible code. This code is an $[n = \sum_{i=1}^{w} n_i,\ k = \sum_{i=1}^{w} k_i,\ d = \min_{1 \le i \le w} \{d_i\}]$ code. Error correction may be done in sections, starting from the right. If all codes $\mathcal{G}_i$ are rank distance codes, we call $\mathcal{G}$ a reducible rank code.

Reducible rank codes build from Gabidulin codes are strongly connected to subfield subcodes of Gabidulin codes:

Remark 3.17 A $\mathbb{F}_{q^m}$-subfield subcode of an $[n, k]$ Gabidulin code over $\mathbb{F}_{q^{2m}}$, where $k > m$ is a reducible rank code up to isometry. Further, every reducible rank code build from $w = 2$ Gabidulin codes, where $\mathsf{Y}_{21} = \mathsf{0}$ and $d_2 \le d_1$ is a subcode of (a $\mathbb{F}_{q^m}$-subcode of) an $[n, n - (n_2 - k_2)]$ Gabidulin code over $\mathbb{F}_{q^{2m}}$.

This observation holds analogous for reducible rank codes build from more than two Gabidulin codes. Considering the structure of the mentioned subfield subcodes and the possibility to correct the errors by section, a special distortion matrix may be applied to hide the structure of the code:

In the examples from [17] the authors propose to take two Gabidulin codes G_1 and G_2 over $\mathbb{F}_{q^m}$ (with length n_i and dimension k_i, $i = 1, 2$) and $\mathsf{Y}_{21} = \mathsf{0}$ to build a reducible rank code $\mathcal{G}$. As public generator matrix they choose

$$\mathsf{G}^{\text{pub}} = \mathsf{S}\left(\mathsf{G} + \begin{bmatrix} \mathsf{X}_1 & \mathsf{0} \\ \mathsf{Y}_1 & \mathsf{X}_2 \end{bmatrix}\right) T \ , \tag{16}$$

where $\mathsf{S} \in \mathbb{F}_{q^m}^{k \times k}$ and $\mathsf{T} \in \mathbb{F}_q^{n \times n}$ are non-singular, $\mathsf{Y}_1 \in \mathbb{F}_{q^m}^{k_2 \times n_1}$ is arbitrary and the rank of $\mathsf{X}_i \in \mathbb{F}_{q^m}^{k_i \times n_i}$ over $\mathbb{F}_q$ is less than t_i for $i = 1, 2$. Using this construction, the authors of [17] propose that the random errors added at encryption should have a rank less than $r = \min_{i=1,2}\left(\frac{n_i - k_i}{2} - t_i\right)$, where en- and decryption work as with GGPT. The authors of [17] claimed every parameter set with $m_i \geq 24$ and $r \geq 4$ to provide sufficient security, even if X_1 and X_2 are zero matrices. They propose to choose $m = n_1 = n_2 = 24$, $k_1 = k_2 = 14$, $t_1 = t_2 = 1$. Note, that because of the use of the column scrambler, we may choose X_i s.t. only the first t_i columns contain non-zero entries. All other choices correspond to an equivalent private key with X_i of the desired form and different T and G. This allows to choose $r = \min_{i=1,2}\{(n_i - k_i - t_i)/2\}$ like for GGPT. Several other modifications like e.g. a subcode variant are possible as well. Further, analogous to the construction above, one might choose to build the reducible rank code from $w > 0$ Gabidulin codes and an adapted distortion matrix as already mentioned in [17]. In the case of $w = 1$, this leads to GGPT. However, we will distinguish between GGPT and RRC-GPT for the ease of comprehensibility.

3.4 Ciphertext Attacks for GPT

Even if the previously known attacks work well for some parameter sets, they still fail for others. In this section we develop an new kind of attack on GPT-like PKCs. Previously, only general decoding attacks or structural attacks (i.e. the ones aiming to recover the secret key from the public key) have been considered. We, however, attack ciphertexts by taking advantage of the recently presented method for decoding interleaved Gabidulin codes beyond minimum distance [30]. Our new attack is superior to all previous attacks: It runs in cubic time and works for all parameter sets of all variants of GPT.

3.4.1 Attacking Ciphertexts of GGPT

Let $(\mathsf{G}^{\text{pub}}, r)$ be the public key of an instance of GGPT. Further, let the ciphertext $\mathbf{y}$ be of the form $\mathbf{y} = \mathbf{x}\mathsf{G}^{\text{pub}} + \mathbf{z}$, where $\mathbf{z} = \left[\begin{array}{c|c} \mathbf{Z} & \mathbf{0} \end{array}\right] \mathsf{T}_{\mathbf{Z}}^{-1}$ is of rank norm r with $\mathbf{Z} \in \mathbb{F}_{q^m}^{r}$ and $\mathsf{T}_{\mathbf{Z}} \in \mathbb{F}_q^{n\times n}$ invertible. To recover the plaintext, an attacker may use a modified version of the error correction procedure for interleaved codes [30]. The major difference is that we use a matrix and not a vector to identify the error vector: We define the space

$$\mathcal{H}_{\mathbf{z}} = \begin{bmatrix} \lambda_{r-1}\left(\mathsf{G}^{\text{pub}}\right) \\ \lambda_{r-1}\left(\mathbf{y}\right) \end{bmatrix}^{\perp} = \begin{bmatrix} \lambda_{r-1}\left(\mathsf{G}^{\text{pub}}\right) \\ \lambda_{r-1}\left(\mathbf{z}\right) \end{bmatrix}^{\perp}. \tag{17}$$

The attack on GGPT is given in algorithm 3.4.1 and succeeds for all parameter sets in polynomial time.

Algorithm 3.4.1 Attacking Ciphertexts of GPT-like Cryptosystems

```
**Input:** A ciphertext $\mathbf{y}$ and the corresponding GGPT public key $(\mathsf{G}^{\text{pub}}, r)$.
**Output:** The plaintext $\mathbf{x}$.

Compute the matrix $\mathsf{H}_{\mathbf{z}}$ generating $\mathcal{H}_{\mathbf{z}}$ of rank $p > k$ over $\mathbb{F}_q$.
Compute an invertible matrix $\widehat{\mathsf{T}} \in \mathbb{F}_q^{n\times n}$, such that $(\mathsf{H}_{\mathbf{z}}\widehat{\mathsf{T}}^{\top})_{\{1,\cdots,n-p\}} = 0$.
Set $\widehat{\mathsf{T}} = \widehat{\mathsf{T}}^{-1}$.
Solve the equation $\mathbf{x}\mathsf{G}^{\text{pub}}\widehat{\mathsf{T}}_{\bullet\{n-p+1,\cdots,n\}} = \mathbf{y}\widehat{\mathsf{T}}_{\bullet\{n-p+1,\cdots,n\}}$.
return $\mathbf{x}$
```

Theorem 3.18 *Algorithm 3.4.1 works correct and has a runtime complexity of $\mathcal{O}(n^3)$ operations over $\mathbb{F}_{q^m}$.*

Proof. Obviously, $\lambda_{r-1}(\mathbf{z})$ has rank r (lemma 3.5), and thus for all vectors $\mathbf{h}_{\mathbf{z}} \in \mathcal{H}_{\mathbf{z}}$:

$$\left(\mathbf{h}_{\mathbf{z}}\mathsf{T}_{\mathbf{Z}}^{\top}\right)_{\{1,\cdots,r\}} = \mathbf{0}.$$

Therefore, the matrix $\mathsf{H}_{\mathbf{z}}$ generating $\mathcal{H}_{\mathbf{z}}$ has rank $p < n - r$ over $\mathbb{F}_q$. Now we assume, that $p \geq k$. We will prove this assumption in lemma 3.19. Let $\widehat{\mathsf{T}} \in \mathbb{F}_q^{n\times n}$ be an invertible matrix satisfying that only the p rightmost columns of $\mathsf{H}_{\mathbf{z}}\widehat{\mathsf{T}}^{\top}$ contain non-zero entries. Such a $\widehat{\mathsf{T}}$ is easy to recover from $\mathsf{H}_{\mathbf{z}}$ by solving linear equations (compare [38]). It follows analogous to lemma 3.7, that the p rightmost positions of $\mathbf{y}\widehat{\mathsf{T}}^{-1}$ have no influence from the error $\mathbf{z}$. This is sufficient for identifying $\mathbf{x}$ since the p rightmost positions of

$\mathsf{G}^{\mathrm{pub}}\widehat{\mathsf{T}}^{-1}$ contain an information set (i.e. the rank of $(\mathsf{G}^{\mathrm{pub}}\widehat{\mathsf{T}}^{-1})_{\bullet\{n-p+1,\cdots,n\}}$ is k). ■

We prove our estimation of the rank of $\mathcal{H}_{\mathbf{z}}$:

Lemma 3.19 *With the notations above: There exists at least one vector of rank norm $\geq k$ in $\mathcal{H}_{\mathbf{z}}$.*

Proof. The secret key holder has to correct the error vector

$$\left(\mathbf{z}\mathsf{T}^{-1}\right)_{\{t+1,\cdots,n\}}$$

of rank norm $\leq r$ in the secret code. Thus, there exists an invertible matrix $\widetilde{\mathsf{T}} \in \mathbb{F}_q^{n\times n}$, such that

$$\left[\ \mathsf{X} \mid \mathsf{0}\ \right] \mathsf{T}\widetilde{\mathsf{T}}^{-1} = \left[\ \mathsf{X} \mid \mathsf{0}\ \right] \text{ and } \left(\mathbf{z}\widetilde{\mathsf{T}}^{-1}\right)_J = \mathbf{0},$$

where $J = \{t+r+1,\cdots,n\}$. Now, let $\mathbf{h}_J$ be some check-vector of the $[n-t-r, k+r-1]$ Gabidulin code $\lambda_{r-1}\left(\mathsf{G}\mathsf{T}\widetilde{\mathsf{T}}^{-1}\right)_{\bullet J}$, then $\left(\ \mathbf{0} \mid \mathbf{h}_J\ \right)(\widetilde{\mathsf{T}}^{-1})^\top$ is in $\mathcal{H}_{\mathbf{z}}$ and has rank norm $n-t-r = k+r \geq k$. ■

We made a large number of experiments with a proof of concept Java implementation of our attack. For parameters from table 3.8 we considered random instances, that is $\mathsf{G}, \mathsf{S}, \mathsf{T}$ and X were randomly chosen from the uniform distibution over the possible matrices. For this and all following experiments we used a standard laptop at 1500 MHz. Timings are given in table 3.9.

Parameters				average runtime	average runtime
m	k	t	s	decryption	algorithm 3.4.1
48	10	16	3	3 seconds	420 seconds
48	16	18	4	3 seconds	450 seconds
64	8	40	1	8 seconds	260 seconds

Table 3.9: Attacking the GGPT cryptosystem

Note, that the attack is applicable even if the column scrambler S is not of quadratic form (like in the case of the Niederreiter GPT) or if the matrix G is replaced by a generator matrix of a subfield subcode of a Gabidulin code. However, in the case of RRC-GPT the situation changes:

3.4.2 Attacking Ciphertexts of RRC-GPT

Let $(\mathsf{G}^{\text{pub}}, r)$ now be a public key of an instance of RRC-GPT as given in equation (16). Analogous to GGPT, a ciphertext has the form $\mathbf{y} = \mathbf{x}\mathsf{G}^{\text{pub}} + \mathbf{z}$, where $\mathbf{z} = \left[\begin{array}{c|c} \mathbf{Z} & \mathbf{0} \end{array}\right] \mathsf{T}_{\mathbf{Z}}^{-1}$ is of rank norm r with $\mathbf{Z} \in \mathbb{F}_{q^m}^{r}$ and $\mathsf{T}_{\mathbf{Z}} \in \mathbb{F}_q^{n \times n}$ invertible. To recover the message, an attacker can view the space $\mathcal{H}_{\mathbf{z}}$ as in equation (17). Again, for all vectors $\mathbf{h}_{\mathbf{z}} \in \mathcal{H}_{\mathbf{z}}$:

$$\left(\mathbf{h}_z \mathsf{T}_{\mathbf{Z}}^{\top}\right)_{\{1,\cdots,r\}} = \mathbf{0}.$$

In the case where no distortion matrix is used, we are able to show that the message $\mathbf{x}$ can always be recovered from $\mathbf{y}$ in polynomial time. On input of $\mathbf{z}$ and $(\mathsf{G}^{\text{pub}}, r)$ to algorithm 3.4.1 two cases may appear. Either, the matrix $\mathsf{H}_{\mathbf{z}}$ is of sufficiently large rank over $\mathbb{F}_q$ or the algorithm fails in the first step. However, if algorithm 3.4.1 fails, the secret key is revealed:

Theorem 3.20 *With the notations above: Let $t_1 = t_2 = 0$, then $\mathbf{x}$ may be revealed in $\mathcal{O}(n^3)$ operations over $\mathbb{F}_{q^m}$ since one of the following statements holds:*

(i) $\forall_{\mathbf{h}_{\mathbf{z}} \in \mathcal{H}_{\mathbf{z}}} \left(\mathbf{h}_{\mathbf{z}}(\mathsf{T}^{-1})^{\top}\right)_{\{1,\cdots,n_1\}} = \mathbf{0}$ *(algorithm 3.4.1 fails) or*

(ii) $\exists_{\mathbf{h}_{\mathbf{z}} \in \mathcal{H}_{\mathbf{z}}} \left(\mathbf{h}_{\mathbf{z}}(\mathsf{T}^{-1})^{\top}\right)_{\{1,\cdots,n_1\}} \neq \mathbf{0}$ *(algorithm 3.4.1 succeeds).*

Proof. Analogous to lemma 3.19, one can show, that there always exists a $\widehat{\mathbf{h}}_{\mathbf{z}} \in \mathcal{H}_{\mathbf{z}}$ of rank norm $k_2 + r$, such that $\left(\widehat{\mathbf{h}}_{\mathbf{z}}(\mathsf{T}^{-1})^{\top}\right)_{\{1,\cdots,n_1+t_2\}} = 0$ (even for arbitrary t_1 and t_2). Thus, in the first case one can recover a matrix $\widetilde{T} \in \mathbb{F}_q^{n \times n}$, such that the last $k_2 + r$ columns from $\mathsf{G}^{\text{pub}}\widetilde{\mathsf{T}}^{-1}$ have no influence from the columns corresponding to G_1 and thus allow to recover S. In consequence, T may be revealed, which is sufficient to recover $\mathbf{x}$.
In the second case, $\left(\mathbf{h}_{\mathbf{z}}(\mathsf{T}^{-1})^{\top}\right)_{\{1,\cdots,n_1\}}$ is in the dual of $\lambda_{k_1+r-2}(\mathbf{g}_1)$, where $\mathbf{g}_1$ is the generator vector of G_1. Thus, $\mathbf{h}_{\mathbf{z}}$ has rank norm $\geq k_1 + r$. Combining this with the observations for the first case, we conclude that a matrix generating $\mathcal{H}_{\mathbf{z}}$ has to have rank $\geq k_1 + k_2 + 2r$ over $\mathbb{F}_q$ and thus algorithm 3.4.1 returns the correct message $\mathbf{x}$. ■

If $t \neq 0$, the security analysis is more complicated. However, even if we are not able to show that $\mathbf{x}$ be recovered in every case, we want to point out why we conclude that no secure instances of RRC-GPT exist:

Remark 3.21 With the notations above, let t_1 and t_2 be arbitrary, then one of the following conditions is true:

(i) The rank of $\mathcal{H}_{\mathbf{z}}$ over $\mathbb{F}_q$ is $R \geq k_1 + k_2$ or

(ii) $\mathcal{H}_{\mathbf{z}}$ reveals S with high probability.

It follows, that we can recover $\mathbf{x}$ from $\mathcal{H}_{\mathbf{z}}$ with high probability.

Proof. In the first case, one can derive a non-singular matrix $\widetilde{\mathsf{T}} \in \mathbb{F}_q^{n\times n}$ from $\mathcal{H}_{\mathbf{z}}$ in cubic time, such that the last R positions of $\mathbf{z}\widetilde{\mathsf{T}}$ are zero. If the last R columns of $\mathsf{G}^{\mathrm{pub}}\widetilde{\mathsf{T}}$ contain an information set (which is the case with high probability), this reveals $\mathbf{x}$.

In the second case, the rank R of $\mathcal{H}_{\mathbf{z}}$ is $< k_1 + k_2$. Thus, an attacker may compute an invertible matrix $\widetilde{\mathsf{T}} \in \mathbb{F}_q^{n\times n}$ from $\mathcal{H}_{\mathbf{z}}$, such that the last R positions of $\mathbf{z}\widetilde{\mathsf{T}}$ are zero. However, the observations from theorem 3.20 show, that the last R columns of $\mathsf{G}^{\mathrm{pub}}\widetilde{\mathsf{T}}$ are of the form

$$\mathsf{S}\left[\begin{array}{c|c} \mathsf{A} & 0 \\ \hline \mathsf{B} & \mathsf{G}_R \end{array}\right]\mathsf{T}_R \in \mathbb{F}_{q^m}^{(k_1+k_2)\times R}$$

for some generator matrix G_R of an Gabidulin code of dimension k_2 and length $n_R \geq n_2 - t_2 - r > k_2$, some arbitrary matrices A and B over $\mathbb{F}_{q^m}$ and $\mathsf{T}_R \in \mathbb{F}_q^{R\times R}$ invertible. It follows, that A is in $\mathbb{F}_{q^m}^{k_1\times(R-n_R)}$ with $R - n_R < k_1$. Now, A will be of full rank with high probability, which reveals T_R and thus S. (However, even if the rank of A is not $R - n_R$, then we can use the methods described in [38] to reveal S if $\lambda_{r-1}(\mathsf{A})$ is of full rank.) If we know S, then it is easy to recover a possible secret key and by this the plaintext $\mathbf{x}$. ■

The remark above shows, that we can either recover a considerable fraction of plaintexts from the ciphertexts, or the secret key is revealed at some point. Further, we would like to point out, that in the case where the reducible rank code is build from more than two Gabidulin codes, analogous considerations hold. We omit giving timings for this attack since there were no serious parameter proposals after the attack from [37]. For random instances of the initial example from [17] algorithm 3.4.1 fails after about 380 seconds, revealing the private key.

3.5 Structural Attacks for GPT and variants

The fact that we can use nearly the same algorithm for correcting errors in Gabidulin codes and for attacking ciphertexts of GPT-like cryptosystems indicates the existence of structural attacks. We do not want to omit the latter, as these might lead to interesting results in coding theory.

Structural attacks take advantage of the main weakness of GPT in comparison with the McEliece PKC: Unlike Goppa codes, Gabidulin codes are highly structured. This property can be used, to distinguish a Gabidulin code from a random one (compare lemma 3.3 and 3.12).

Previous structural attacks fail short to recover a valid secret key from the public key in a feasible number of operations for all parameter sets. We first present the results of the former attacks and present our new attacks afterwards, which work for all possible variants of cryptosystems build from Gabidulin codes. The new attacks we are going to present vary slightly from the ones published by the author in [38] and [37] and are more powerful.

3.5.1 Gibson's Attacks

Gibson presented two structural attacks on the GPT cryptosystem. They recover an alternative private-key from the GGPT public-key G^{pub}. On input of $\mathsf{G}^{\text{pub}} = \mathsf{S}\left(\left[\ \mathsf{X} \mid \mathsf{0}\ \right] + \mathsf{G}\right)\mathsf{T}$, Gibson's attacks return $\widehat{\mathsf{G}}$, $\widehat{\mathsf{X}} \in \mathbb{F}_{q^m}^{k\times n}$ and $\widehat{\mathsf{S}} \in \mathbb{F}_{q^m}^{k\times k}$, satisfying that

- (i) $\widehat{\mathsf{G}}$ is a generator matrix of an $[n, k]$ Gabidulin code over $\mathbb{F}_{q^m}$,
- (ii) $\mathsf{G}^{\text{pub}} = \widehat{\mathsf{S}}\left(\widehat{\mathsf{G}} + \widehat{\mathsf{X}}\right)$ and
- (iii) the rank of $\widehat{\mathsf{X}}$ over $\mathbb{F}_q$ is not bigger than t.

Thus Gibson's attacks serve well for an attack on the GGPT cryptosystem, as an alternative column scrambler may be recovered from $\widehat{\mathsf{X}}$. Gibson's first attack was developed for the case that the GGPT parameter s is very small. It is a variation of the approach for GPT without distortion matrix ($s = 0$), which recovers a generator vector of a Gabidulin code from its systematic generator matrix by solving some linear equations. If the parameter s is small enough, the attacker can guess some unknown values to eliminate the effect of the distortion matrix. This first attack takes

$$\mathcal{O}\left(m^3\,(n-k)^3\,q^{ms}\right) \tag{18}$$

operations over $\mathbb{F}_{q^m}$. In [22] Gibson presented a different attack, which analyzes matrices of the form $\mathsf{G} + \mathsf{G}^{[q]}$. This attack is more efficient for larger values of s. It runs in

$$\mathcal{O}\left(k^3 + (k+t)\,f\cdot q^{f(k+2)} + (m-k)\,t\cdot q^f\right) \tag{19}$$

operations over $\mathbb{F}_{q^m}$, where $f \approx \max(0, t-2s, t+1-k)$. Note, that this attack runs in polynomial time iff $f = 0$. The success of both attacks is based on some assumptions, which are claimed to be fulfilled with high probability for random instances of the GGPT cryptosystem. Nevertheless Gibson's attacks are not fast enough to attack the GGPT cryptosystem for all parameter sets of practical interest.

3.5.2 Ourivski's Attack on the Niederreiter Variant

In 2003 A. Ourivski chose an approach similar to the one of the first attack from Gibson. He analyzed the public key and was able to recover the secret key by guessing some values and solving some linear equations afterwards. The number of elements an attacker has to guess using Ourivski's attack is expressed by the parameter f below.

Without loss of generality we may assume, that the public check matrix of an instance of the Niederreiter GPT is of the following form:

$$\mathsf{H}^{\text{pub}} = \left[\begin{array}{c|c} \mathsf{H} & \begin{array}{c} 0 \\ \mathsf{Id}_l \\ \widetilde{\mathsf{A}} \end{array} \end{array}\right] \in \mathbb{F}_{q^m}^{n \times l} \ ,$$

where $\widetilde{\mathsf{A}} \in \mathbb{F}_{q^m}^{(k-l)\times l}$, Id_k denotes the k-dimensional identity matrix and H is a check matrix of the secret Gabidulin code. Let $v \leq l$ be the column-rank of $\widetilde{\mathsf{A}}$ over $\mathbb{F}_q$ and $a \leq \min\{v, k-l\}$ be the rank of $\widetilde{\mathsf{A}}$ over $\mathbb{F}_{q^m}$. Ourivski's attack takes

$$\mathcal{O}\left(3m^3 + nf \cdot q^{m(f-1)}\right)$$

operations over $\mathbb{F}_q$, where $f \approx v + 1 - \min\{v, a(n-k)\}$ for most instances. Even if no proof is given, experiments corroborate Ourivski's estimation of f. Because $0 \leq v \leq l$, this attack runs in polynomial time, iff $l \leq a(n-k)$. Ourivski states, that the parameter a should not be to small (≥ 3), as otherwise a different attack approaches can be used to recover a private key. Thus, for the worst case with fixed a, the work factor for Ourivski's attack is

$$\mathcal{O}\left(3m^3 + nl \cdot q^{m(l-\mathcal{O}(1)(n-k))}\right) \ .$$

3.5.3 Attacking the Niederreiter Variant in Polynomial Time

Even if Ourivski's attack on the Niederreiter GPT works well, it still has exponential work factor for special parameter sets. Nevertheless, it is not the only way to recover the secret key for the Niederreiter GPT. We present a variation of an earlier attack by the author [38], which recovers an alternative secret key in polynomial time.

Theorem 3.22 *Let $\mathsf{G}_{\mathrm{SUB}}^{\mathrm{pub}}$ be the $k-\ell$ dimensional subcode of an $[n,k]$ Gabidulin code $\mathcal{G}$ over $\mathbb{F}_{q^m}$ defined by an instance of the Niederreiter GPT. Then we may recover a Gabidulin code $\widehat{\mathcal{G}}$ which contains $\mathcal{G}_{\mathrm{SUB}}$ from $\mathcal{G}_{\mathrm{SUB}}$ in $\mathcal{O}\left(n^3\right)$ operations over $\mathbb{F}_{q^m}$.*

Proof. Let $\mathsf{G}_{\mathrm{SUB}}$ be the generator matrix of $\mathcal{G}_{\mathrm{SUB}}$, then $\lambda_{n-k-1}(\mathsf{G}_{\mathrm{SUB}})$ is a subcode of an $[n,n-1]$ Gabidulin code. It follows, that $(\lambda_{n-k-1}(\mathsf{G}_{\mathrm{SUB}}))^{\perp} = \widehat{\mathcal{H}}$ contains a vector $\widehat{\mathbf{h}}$ of rank norm n. It might not be easy to determine such a vector over $\mathbb{F}_{q^m}$. Yet it is easy to find such a vector over $\mathbb{F}_{q^{am}}$, where $a \leq \dim \widehat{\mathcal{H}}$ as any matrix generating $\widehat{\mathcal{H}}$ represents such a $\widehat{\mathbf{h}}$ over $\mathbb{F}_{q^m}$. From the simple observation that for all $\mathbf{a},\mathbf{b} \in \mathbb{F}_{q^m}^n : (\mathbf{a}\mathbf{b}^\top = 0 \wedge \mathbf{a}^{[q]}\mathbf{b}^\top = 0) \Leftrightarrow (\mathbf{a}^{[q]}(\mathbf{b}^{[q]})^\top = 0 \wedge \mathbf{a}^{[q]}\mathbf{b}^\top = 0)$ it follows that $\mathcal{G}_{\mathrm{SUB}}$ is in the vector space spanned by the rows of $(\lambda_{n-k-1}(\widehat{\mathbf{h}}^{[1/q^{n-k-1}]}))^{\perp}$, which is a $[n,k]$ Gabidulin code. Thus, we have found a valid secret key in $\mathcal{O}\left(n^3\right)$ operations over $\mathbb{F}_{q^m}$. ■

Since there were no proposals for parameter sets for the Niederreiter GPT after Ourivski's attack, we did not carry out any experiments for this variant of GPT.

3.5.4 Recovering GGPT Private Keys in Polynomial Time

As we have seen in the previous section, the structure of Gabidulin codes allows to recover the original code from a subcode. The same holds for distorted Gabidulin codes like the public key of most GPT variants. In the following let $\left(\mathsf{G}^{\mathrm{pub}}, r\right)$ be the public key of an instance of the GGPT cryptosystem with parameters q, m, n, k, t and s and $(\mathsf{G}, \mathsf{S}, \mathsf{T})$ be a corresponding secret key as in section 3.3. The attack strategy is always the same and can be summarized in algorithm 3.5.1.

Note that this is a minor variation of the attack published by the author in [38]. In the following we will show that this strategy indeed allows an attacker to build a valid secret key.

A crucial point for this type of attacks on the private key of GGPT is the analysis of the structure of the dual of $\Lambda_f(\mathsf{G}^{\mathrm{pub}})$. It will show, that the second step of algorithm 3.5.1 does not fail. The structure of $\Lambda_f(\mathsf{G}^{\mathrm{pub}})$

Algorithm 3.5.1 Structural attack GGPT

```
Input: A GGPT public key $(\mathsf{G}^{\mathrm{pub}}, r)$.
Output: A secret key for $(\mathsf{G}^{\mathrm{pub}}, r)$.

Compute $\Lambda_f(\mathsf{G}^{\mathrm{pub}})^{\perp}$ for $f = 2r - 1$.

Choose a vector $\mathbf{h} \in \mathbb{F}_{q^{am}}^{n}$ dual to $\Lambda_f(\mathsf{G}^{\mathrm{pub}})$ of rank norm $N \geq n - t$.

Compute an invertible $\widehat{\mathsf{T}} \in \mathbb{F}_q^{n\times n}$ s.t. $\widehat{\mathbf{h}}_{\{1,\cdots,n-N\}} = 0$, where $\widehat{\mathbf{h}} = \mathbf{h}\widehat{\mathsf{T}}^{\top}$.

Build the $[N, N-(n-k)]$ Gabidulin code $\widehat{\mathcal{G}} = \langle\, \widehat{\mathsf{G}} \,\rangle$ with check vector $\widehat{\mathbf{h}}$.

Compute $\widehat{\mathsf{S}} \in \mathbb{F}_{q^m}^{k\times N-(n-k)}$ satisfying $\widehat{\mathsf{S}}\widehat{\mathsf{G}} = (\mathsf{G}^{\mathrm{pub}}\widehat{\mathsf{T}}^{-1})_{\bullet\{n-N+1,\cdots,n\}}$.

Return the secret key $\left(\mathcal{D}_{\widehat{\mathcal{G}}}, \widehat{\mathsf{S}}, \widehat{\mathsf{T}}\right)$ for $(\mathsf{G}^{\mathrm{pub}}, r)$.
```

depends mainly on f and the $k \times t$ distortion matrix X of rank s, which is used during the key generation phase of GGPT. We want to remind the reader that $n - t - k = 2r$.

Lemma 3.23 *For $0 \leq f \leq 2r - 1$ there exists a dual matrix of $\Lambda_f(\mathsf{G}^{\mathrm{pub}})$ of the form*

$$\Lambda_f(\mathsf{G}^{\mathrm{pub}})^{\perp} = \begin{bmatrix} 0 & \mathsf{H}_f^{\top} \\ \mathsf{B}_1 & \mathsf{B}_2 \end{bmatrix} \cdot \left(\mathsf{T}^{-1}\right)^{\top} \in \mathbb{F}_{q^m}^{(2r-f+\ell)\times n}, \tag{20}$$

where $\mathsf{H}_f \in \mathbb{F}_{q^m}^{(n-t)\times(2r-f)}$ is the check matrix of a $k+f$ dimensional Gabidulin code $\mathcal{G}_f$ of length $n - t$, B_1 is some $\ell \times t$ matrix with $0 \leq \ell \leq t$ and B_2 is some $\ell \times (n - t)$ matrix.

Proof. First, we assume, that T and S are the identity matrix. The proof is analogous, if this is not the case. We may write

$$\Lambda_f(\mathsf{G}^{\mathrm{pub}}) = [\, \underbrace{\Lambda_f\left(\mathsf{G}_{\bullet\{1,\cdots,t\}} + \mathsf{X}\right)}_{t} \mid \underbrace{\Lambda_f\left(\mathsf{G}_{\bullet\{t+1,\cdots,n\}}\right)}_{n-t} \,] \in \mathbb{F}_{q^m}^{(k(f+1))\times n}$$

By lemma 3.3, the last $n - t$ columns of $\Lambda_f(\mathsf{G}^{\mathrm{pub}})$ define an $[n - t, k + f]$ Gabidulin code $\mathcal{G}_f$. Thus the subvectorspace spanned by the rows of

$$[\ 0 \mid \mathsf{H}_f^{\top}\ ] \in \mathbb{F}_{q^m}^{(2r-f)\times n},$$

where $\mathsf{H}_f \in \mathbb{F}_{q^m}^{(n-t)\times(2r-f)}$ is the check matrix of $\mathcal{G}_f$, is in the dual space of $\Lambda_f(\mathsf{G}^{\text{pub}})$. To get a matrix which defines the whole dual space of $\Lambda_f(\mathsf{G}^{\text{pub}})$, we might have to add some more rows to $\left[\begin{array}{c|c} \mathsf{0} & \mathsf{H}_f^\top \end{array}\right]$, which already has rank $n-t$ over $\mathbb{F}_q$. However, it is clear, that there will be at most t rows missing, as $\Lambda_f(\mathsf{G}^{\text{pub}})$ has at least rank $k+f$. This proves the theorem. ■

The lemma yields the existence of a vector $\mathbf{h}$ of rank norm $N \geq n-t$ in $\Lambda_f(\mathsf{G}^{\text{pub}})^\perp$ for $f = 2r-1$. (Such a $\mathbf{h}$ is easy to recover over an extension field $\mathbb{F}_{q^{am}}$ of $\mathbb{F}_{q^m}$ with $a < n$.) Consequently the second step of our attack can be done in $\mathcal{O}(n^3)$ operations and does not fail. It remains to show, that every choice of $\mathbf{h}$ leads to a valid alternative secret key for the public GGPT key G^{pub}.

Theorem 3.24 *Let $f = 2r-1$, $\mathbf{h} \in \Lambda_f(\mathsf{G}^{\text{pub}})^\perp$ be of rank norm $N \geq n-t$. Further, let $\widehat{\mathsf{T}} \in \mathbb{F}_q^{n\times n}$ such that for $\widehat{\mathbf{h}} = \mathbf{h}\widehat{\mathsf{T}}^\top$: $\widehat{\mathbf{h}}_{1,\cdots,n-N} = 0$. Then, $\widehat{\mathbf{h}}$ is the check vector of the Gabidulin (sub-)code defined by the last N columns of $\mathsf{G}^{\text{pub}}\widehat{\mathsf{T}}^{-1}$. Thus, algorithm 3.5.1 returns a valid secret key.*

Proof. As in the proof of theorem 3.22, we use again the fact, that for all $\mathbf{a}, \mathbf{b} \in \mathbb{F}_{q^m}^n$:

$$(\mathbf{a}\mathbf{b}^\top = 0 \wedge \mathbf{a}^{[q]}\mathbf{b}^\top = 0) \Leftrightarrow (\mathbf{a}^{[q]}(\mathbf{b}^{[q]})^\top = 0 \wedge \mathbf{a}^{[q]}\mathbf{b}^\top = 0).$$

It follows, that $\widehat{\mathbf{h}}^{[1/q^{f-i}]} \in \lambda_i(\mathsf{G}^{\text{pub}}\widehat{\mathsf{T}}^{-1})^\perp$ for all $i = f, f-1, \cdots, 1$ and thus

$$\widehat{\mathsf{H}}^\top := \lambda_f\left(\widehat{\mathbf{h}}^{[1/q^{2r-1}]}\right) \subseteq (\mathsf{G}^{\text{pub}}\widehat{\mathsf{T}}^{-1})^\perp.$$

As the N last rows of $\widehat{\mathsf{H}}^\top$ form a check matrix of an $[N, N-(n-k)]$ Gabidulin code $\widehat{\mathcal{G}}$, the matrix $(\mathsf{G}^{\text{pub}}\widehat{\mathsf{T}}^{-1})_{\bullet\{n-N+1,\cdots,n\}}$ generates (a subcode of) $\widehat{\mathcal{G}}$. ■

From the theorem, one can see, that $(\mathsf{G}^{\text{pub}}\widehat{\mathsf{T}}^{-1})_{\bullet\{n-N+1,\cdots,n\}}$ is indeed a (sub-)code of a Gabidulin code with generator matrix $\widehat{\mathsf{G}}$ and known error correction algorithm $\mathcal{D}_{\widehat{\mathcal{G}}}$. Thus, an matrix $\widehat{\mathsf{S}} \in \mathbb{F}_{q^m}^{k\times N-(n-k)}$, which is the remaining part for an alternative secret key $(\mathcal{D}_{\widehat{\mathcal{G}}}, \widehat{\mathsf{S}}, \widehat{\mathsf{T}})$ may be easily recovered. Note that the presented attack may be performed in $\mathcal{O}(n^3)$ operations over $\mathbb{F}_{q^m}$ and works for GGPT, all simple variants and for any subfield subcode version of GPT using the error correction algorithm of the original Gabidulin code. Thus, this modified version from the attack from [38] is more powerful than the original one.

Some timings for the attack may be found in the table 3.10. Again, we viewed random instances with uniformly distributed $\mathsf{G}, \mathsf{S}, \mathsf{T}$ and X. The differences to the values in [38] are due to the faster computer used.

Parameters				average runtime	WF best of
m	k	t	s	algorithm 3.5.1	Gibson's attacks
48	10	16	3	17 minutes	2^{139}
48	16	18	4	20 minutes	2^{200}
64	8	40	1	7 minutes	2^{111}

Table 3.10: Attacking the GGPT cryptosystem

3.5.5 A Structural Attack for "RRC-GPT"

In [39] the author presented a security reduction for GPT with reducible rank codes. The main idea is, to view only parts of the public generator matrix, which define public generator matrices of the CS-GPT cryptosystem. We will limit ourselves to the case, where the secret RRC is build from two Gabidulin codes. Proofs are analogous for all other cases.

Let $\left(\mathsf{G}^{\text{pub}}, r\right)$ be the public key of an instance of the RRC-GPT cryptosystem as given in equation (16) with parameters q, m and n_i, k_i, t_i, $i = 1, 2$. To attack RRC-GPT we first rewrite the public generator matrix:

Lemma 3.25 *Let* $(\mathsf{G}, \mathsf{S}, \mathsf{T})$ *be the secret key corresponding to the RRC-GPT public key* $\left(\mathsf{G}^{\text{pub}}, r\right)$. *Then there exists an invertible matrix* $\widehat{\mathsf{T}} \in \mathbb{F}_q^{(n_1+n_2)\times(n_1+n_2)}$ *such that*

$$\mathsf{G}^{\text{pub}}\widehat{\mathsf{T}}^{-1} = \mathsf{S}\left[\begin{array}{c|c|c} \mathsf{Z}_1 & \mathsf{G}_1 & 0 \\ \hline \mathsf{Z}_2 & \mathsf{Y} & \mathsf{G}_2 \end{array}\right], \tag{21}$$

where the matrices G_i *are generator matrices of* $[n_i - t_i, k_i]$ *Gabidulin codes and* $\mathsf{Z}_i \in \mathbb{F}_{q^m}^{k_i\times(t_1+t_2)}$ *as well as* $\mathsf{Y} \in \mathbb{F}_{q^m}^{k_2\times(n_1-t_1)}$ *are arbitrary matrices.*

Further, if the matrix S_{JK_2} *is invertible for a subset* $J \subseteq \{1, \cdots, k_1 + k_2\}$ *and* $K_2 := \{k_1 + 1, \cdots, k_1 + k_2\}$, *then* $\mathsf{G}^{\text{pub}}_{J\bullet}$ *is an instance of the CS-GPT cryptosystem.*

Proof. As the matrices X_1 and X_2 used on key generation are of column rank smaller than t_i over $\mathbb{F}_q$, we may assume without loss of generality, that only their first t_i columns contain non-zero entries. Thus, by exchanging the $(t_1 + i)$-th column of $\mathsf{G}^{\text{pub}}\mathsf{T}^{-1}$ with the $(n_1 + i)$-th column for $i = 1, \cdots, t_2$

and modifying T accordingly, we get a matrix $\widehat{\mathsf{T}}$ with the desired properties. The fact that $\mathsf{G}^{\mathrm{pub}}_{J\bullet}$ forms a instance of the CS-GPT follows from the observations above. ■

The representation of the private key as in equation (21) suggests an attack on RRC-GPT as sketched in algorithm 3.5.2, which is a modification of the attack presented by the author in [37]. We prove the correctness of the algorithm in the case where $w = 2$:

Algorithm 3.5.2 Structural Attack for RRC-GPT

```
Input: A RRC-GPT public key $(\mathsf{G}^{\mathrm{pub}}, r)$.
Output: The row and column scrambler of a secret key for $(\mathsf{G}^{\mathrm{pub}}, r)$.

$\widehat{\mathsf{G}^{\mathrm{pub}}} = \mathsf{G}^{\mathrm{pub}}$, $N = n$ and $K = k$.
for $i = w$ down to 1 do
  Recover a (partial) column scrambler $\mathsf{T}_i \in \mathbb{F}_q^{N\times N}$ from the matrix $\lambda_f(\widehat{\mathsf{G}^{\mathrm{pub}}})$, where $f = n_w - k_w - t_w - 1 \leq 2r - 1$ as in algorithm 3.5.1.

  Verify that the $N_i$ rightmost columns of $\widehat{\mathsf{G}^{\mathrm{pub}}}\mathsf{T}_i^{-1}$ define an $[N_i, K_i]$ Gabidulin (sub-)code, where $N_i$ is maximal.

  Compute a (partial) row scrambler $\mathsf{S}_i \in \mathbb{F}_{q^m}^{K\times K}$, such that $(\mathsf{S}_i\widehat{\mathsf{G}^{\mathrm{pub}}}\mathsf{T}_i^{-1})_{\{1,\cdots,K-K_i\}\{N-N_i+1,\cdots,n\}} = \mathsf{0}$.

  Set $\mathsf{T}_i = \left[\begin{array}{c|c} \mathsf{T}_i & \mathsf{0} \\ \hline \mathsf{0} & \mathrm{Id}_{n-N} \end{array}\right]$ and $\mathsf{S}_i = \left[\begin{array}{c|c} \mathsf{S}_i & \mathsf{0} \\ \hline \mathsf{0} & \mathrm{Id}_{k-K} \end{array}\right]$.
  Set $N = N - N_i$, $K = K - K_i$ and $\widehat{\mathsf{G}^{\mathrm{pub}}} = (\mathsf{S}_i\mathsf{G}^{\mathrm{pub}}\mathsf{T}_i^{-1})_{\{1,\cdots,K\}\{1,\cdots,N\}}$ .

return $S = S_w \cdots S_1$ and $T = T_1 \cdots T_w$.
```

Theorem 3.26 *Algorithm 3.5.2 returns a pair of row- and column scrambler belonging to a valid secret key for* $(\mathsf{G}^{\mathrm{pub}}, r)$*.*

Proof. For the correctness of the first step of the loop: If $\mathbf{h}_2$ is the check vector of G_2, then $\left(\begin{array}{c|c|c} \mathbf{0} & \mathbf{0} & \mathbf{h}_2 \end{array}\right)$ will be in $\lambda_f(\mathsf{G}^{\mathrm{pub}}\widehat{\mathsf{T}}^{-1})^{\perp}$, where $\widehat{\mathsf{T}}$ is as in equation (21). Let $\mathbf{h} \in \mathbb{F}_{q^{am}}^n$ be a vector of maximal rank $N_2 \geq n_2 - t_2$ in the dual of $\lambda_f(\mathsf{G}^{\mathrm{pub}})^{\perp}$. Note, that we do not restrict $\mathbf{h}$ to $\mathbb{F}_{q^m}^n$, as to assure, that its norm will not be limited by m. Further, as the norm of $\mathbf{h}$ is maximal, the

$n_2 - t_2$ last positions of $\mathbf{h}\widehat{\mathsf{T}}^\top$ are non-zero. Let $\mathsf{T}_2 \in \mathbb{F}_{q^m}^{n\times n}$ be an invertible matrix s.t. the first $n - N_2$ positions of $\mathbf{h}\mathsf{T}_2^\top$ are zero. As in the previous section, $\mathsf{G}^{\text{pub}}\mathsf{T}_2^{-1} \subseteq \lambda_f(\mathbf{h}^{[1/q^{1/f}]}\mathsf{T}_2^\top)$, thus we can already correct errors of rank up to r in the last N_2 positions of $\mathsf{G}^{\text{pub}}\mathsf{T}_2^{-1}$. Thus, the verification in the second step of the loop never fails.

It remains to show that step three to five of the loop generate a public key of the GGPT cryptosystem, with minimum distance $\geq (2r+1)$: Let K_2 be the rank of the submatrix of $\mathsf{G}^{\text{pub}}\mathsf{T}_2^{-1}$, which consists of the last N_2 columns. If $K_2 = k$, we are done. Otherwise $k > K_2 \geq k_2$. In this case, it is easy to compute an row scrambler $\mathsf{S}_2 \in \mathbb{F}_{q^m}^{k\times k}$, such that $(\mathsf{S}_2\mathsf{G}^{\text{pub}}\mathsf{T}_2^{-1})_{\{1,\cdots,k-K_2\}\{n-N_2+1,\cdots,n\}} = \mathsf{0}$. Thus, the first $k - K_2$ rows of $\mathsf{S}_2\mathsf{G}^{\text{pub}}$ have no influence from G_2. These rows form a subcode of some public code of the GGPT cryptosystem which can correct errors of rank up to r and thus has minimum distance $\geq (2r+1)$. This property does not change, if we remove the last N_2 columns, as they are zero. ■

It follows, that we can recover a alternative secret key in $\mathcal{O}(n^3)$ operations over $\mathbb{F}_{q^{am}}$ with $a < n \leq 2m$. Thus, unlike stated by the author in [37], there are no instances of RRC-GPT, which are secure against structural attacks. Again, we omit giving timings for more than the initially proposed parameter set from [17], whose random instances can be broken by algorithm 3.5.2 in about 10 minutes.

4 Conclusions and Perspectives

For the McEliece PKC, to our knowledge, the best attack is the one proposed by Canteaut and Chabaud [7]. Despite the improvement achieved in this thesis, the current versions of the statistical decoding algorithm have no advantage over this attack for any reasonable parameter set. Consequently, parameter sets for the McEliece cryptosystem remain unchanged by the results of this thesis.

However, the presented methods to improve statistical decoding can certainly be transfered to iterative decoding. Further, one might try to weight the information obtained by the different sets of check vectors in an other way. It would be interesting to check whether one can get a significant improvement by choosing a different transformation of the random variables than to $\mathcal{N}(0,1)$.

The larger part of this thesis was dedicated to attacks on GPT-like PKCs. We gathered up proposed techniques to prevent such attacks and have shown that none of the existing GPT variants is secure. Neither the addition of random redundancy, distortion matrices or supplementary check vectors, nor the employment of reducible codes is sufficient to allow the use of Gabidulin codes in cryptography. Additionally, we proved that (unlike in the proposal from Niederreiter) the use of subfield subcodes does not lead to a secure GPT variant. Because of their highly structured generator matrix, Gabidulin codes can not be used for cryptographic applications.

Again, a class of codes which can easily be distinguished from a random code has been proven unsuitable for cryptographic applications. This corroborates the evidence, that the existence of a distinguisher indicates the insecurity of a cryptosystem, which is e.g. the case for quasi-cyclic codes [19].

Promising to our opinion could be research on the connection of Goppa and Gabidulin codes via (generalized) Srivastava codes [32]. Both classes have an intersection with the class of Srivastava codes and thus the latter might be used either to lead way to an attack on McEliece's cryptosystem or to a codebased cryptosystem with smaller key sizes.

Nevertheless, Gabidulin codes are interesting and offer a variety of different applications. In this thesis we developed a cubic time error correction algorithm which corrects rank errors up to the amount of redundancy in an interleaved Gabidulin code with overwhelming probability.

As mentioned in section 3.3, we can view an erroneous codeword of an interleaved Gabidulin code as the public key of the GGPT cryptosystem.

The fact that we are able to recover partial information about the used secret key might open the way to correct errors in received codewords of an interleaved Gabidulin code without the knowledge of the underlying code. The property of codewords to allow error correction without knowledge of the underlying code is new and might be interesting. To our knowledge, there exists no other code offering this property.

References

[1] D. Augot, M. Finiasz, and N.Sendrier. A family of fast syndrome based cryptographic hash functions. In *Proc. of Mycrypt 2005*, volume 3715 of *LNCS*, pages 64–83, 2005.

[2] Ch.H. Bennet and G. Brassard. Quantum cryptography: public key distribution and coin tossing. *Proc. of IEEE Int. conf. Computers, Systems & Signal Processing, Bangalore, India*, pages 175–179, 1984.

[3] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani. Strengths and weaknesses of quantum computing, 1994. Preprint at http://vesta.physics.ucla.edu/cgi-bin/uncompress_ps_cgi?bbbv94h.ps.

[4] 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.

[5] E.R. Berlekamp. *Algebraic coding theory.* McGraw-Hill, New York, 1968.

[6] D. Bleichenbacher, A. Kiayias, and M. Yung. Decoding of interleaved Reed Solomon codes over noisy data. In *Proc. of ICALP 2003*, volume 2719 of *LNCS*, pages 97–108, 2003.

[7] A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEETIT: IEEE Transactions on Information Theory*, 44, 1998.

[8] N. Courtois, M. Finiasz, and N.Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248, pages 157–174. Springer-Verlag, 2001.

[9] G. Doolen and R. Hughes et al. A quantum information science and technology roadmap. In *Technical report LA-UR-02-6900*, 2002. available at http://qist.lanl.gov.

[10] D. Engelbert, R. Overbeck, and A. Schmidt. A summary of McEliece-type cryptosystems and their security. *Journal of Mathematical Cryptology*, 1(2):151–199, 2007.

[11] S. R. Finch. *Mathematical Constants.* Encyclopedia of Mathematics and Applications. Cambridge, 2003. (see http://mathworld.wolfram.com/InfiniteProduct.html).

[12] J.-B. Fischer and J. Stern. An eficient pseudo-random generator provably as secure as syndrome decoding. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *LNCS*, pages 245–255. Springer-Verlag, 1996.

[13] M. Fossorier, H. Imai, and K. Kobara. Modeling bit flipping decoding based on non orthogonal check sums and application to iterative decoding attack of McEliece crypto-system. In *Proc. of 2004 International Symposium on Information Theory and its Applications, Parma, Italy (ISITA'04)*, October 2004.

[14] E. M. Gabidulin and A. V. Ourivski. Column scrambler for the GPT cryptosystem. *Discrete Applied Mathematics*, 128(1):207–221, 2003.

[15] E.M. Gabidulin. Theory of codes with maximum rank distance. *Problems of Information Transmission*, 21, No. 1, 1985.

[16] E.M. Gabidulin and P. Loidreau. Subfield subcodes of maximum-rank distance codes. In *Seventh International Workshop on Algebraic and Combinatorial Coding Theory*, volume 7 of *ACCT*, pages 151–156, 2000.

[17] E.M. Gabidulin, A.V. Ourivski, B. Honary, and B. Ammar. Reducible rank codes and their applications to cryptography. *IEEE Transactions on Information Theory*, 49(12):3289–3293, 2003.

[18] E.M. Gabidulin, A.V. Paramonov, and O.V. Tretjakov. Ideals over a non-commutative ring and their applications to cryptography. In *Proc. Eurocrypt '91*, volume 547 of *LNCS*. Springer Verlag, 1991.

[19] P. Gaborit. Shorter keys for code based cryptography. In *Proc. of WCC 2005*, pages 81–90, 2005.

[20] J. K. Gibson. Severely denting the Gabidulin version of the McEliece public key cryptosystem. *Designs, Codes and Cryptography*, 6(1):37–45, July 1995.

[21] K. Gibson. Equivalent Goppa codes and trapdoors to McEliece's public key cryptosystem. In D. W. Davies, editor, *Advances in Cryptology - Eurocrypt'91*, volume 547 of *LNCS*, pages 517–521. Springer Verlag, 1991.

[22] K. Gibson. The security of the Gabidulin public key cryptosystem. In *Proc. of Eurocrypt'96*, volume 1070 of *LNCS*, pages 212–223. Springer Verlag, 1996.

[23] A Kh. Al Jabri. A statistical decoding algorithm for general linear block codes. In *Cryptography and Coding 2001*, volume 2260 of *LNCS*, pages 1–8. Springer Verlag, 2001.

[24] T. Johansson and A.V. Ourivski. New technique for decoding codes in the rank metric and its cryptography applications. *Problems of Information Transmission*, 38, No. 3:237–246, 2002.

[25] D. Kielpinski, C.R. Monroe, and D.J. Wineland. Architecture for a large-scale ion-trap quantum computer. *Nature*, 417:709–711, 2002. available at http://www.boulder.nist.gov/.

[26] K. Kobara and H. Imai. Semantically secure McEliece public-key cryptosystems - conversions for McEliece PKC. In *Practice and Theory in Public Key Cryptography - PKC '01 Proceedings*. Springer Verlag, 2001.

[27] P.J. Lee and E.F. Brickell. An observation on the security of McEliece's public key cryptosystem. In *Advances in Cryptology-EUROCRYPT'88*, volume 330 of *LNCS*, pages 275–280. Springer Verlag, 1989.

[28] Arjen K. Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 14(4):255–293, 2001.

[29] Y.X. Li, R.H. Deng, and X.M. Wang. the equivalence of McEliece's and Niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, Vol. 40, pp. 271-273, 1994.

[30] P. Loidreau and R. Overbeck. Decoding rank errors beyond the error-correction capability. In *Proc. of ACCT-10, Zvenigorod*, 2006.

[31] P. Loidreau and N. Sendrier. Weak keys in the McEliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47, No. 3:1207 –1211, March 2001.

[32] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correctiong Codes*. North-Holland Amsterdam, 7 edition, 1992.

[33] R.J. McEliece. A public key cryptosystem based on algebraic coding theory. *DSN progress report*, 42-44:114–116, 1978.

[34] L. Minder. Breaking the Sidelnikov cryptosystem. In *Proc. of Eurocrypt'07.* to appear.

[35] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Probl. Control and Inform. Theory*, 15:19–34, 1986.

[36] A.V. Ourivski. Recovering a parent code for subcodes of maximal rank distance codes. In *Proc. of WCC 03*, pages 357–363, 2003.

[37] R. Overbeck. Structural attacks for public key cryptosystems based on Gabidulin codes. *Journal of Cryptology.* accepted for publication.

[38] R. Overbeck. A new structural attack for GPT and variants. In *Proc. of Mycrypt 2005*, volume 3715 of *LNCS*, pages 50–63. Springer Verlag, 2005.

[39] R. Overbeck. Extending Gibson's attacks on the GPT cryptosystem. In *Proc. of WCC 2005*, volume 3969 of *LNCS*, pages 178–188. Springer Verlag, 2006.

[40] R. Overbeck. Statistical decoding revisited. In *Proc. of ACISP 2006*, volume 4058 of *LNCS*, pages 283–294. Springer Verlag, 2006.

[41] N. Sendrier. On the security of the McEliece public-key cryptosystem. In M. Blaum, P.G. Farrell, and H. van Tilborg, editors, *Proceedings of Workshop honoring Prof. Bob McEliece on his 60th birthday*, pages 141–163. Kluwer, 2002.

[42] P.W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. of Computing*, 26(5):1484–1509, 1997.

[43] V.M. Sidelnikov. A public-key cryptosystem based on binary Reed-Muller codes. *Discrete Mathematics and Applications*, 4 No. 3, 1994.

[44] V.M. Sidelnikov and S.O. Shestakov. On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 2, No. 4:439–444, 1992.

[45] J. Stern. A method for finding codewords of small weight. *Coding Theory and Applications*, 388:106–133, 1989.

[46] M. Ogle T. Migler, K.E. Morrison. Weight and rank of matrices over finite fields, 2003. available at http://www.calpoly.edu/˜kmorriso/Research/research.html.

[47] Christian Wieschebrink. An attack on a modified Niederreiter encryption scheme. In *Public Key Cryptography*, volume 3958 of *LNCS*, pages 14–26, 2006.