Implementation of the QC-MDPC cryptosystem over GF(4)

Author: Ing. Tomáš Vavro Supervisor: Mgr. Tomáš Fabšič, PhD.

Faculty of Electrical Engineering and Information Technology of STU in Bratislava

Goals and motivation

The advent of quantum computers poses a substantial threat to asymmetric cryptosystems based on computational hardness assumption of the prime factorization and the discrete logarithm problems. The vulnerable cryptosystems are present in many aspects of everyday life: communication with your bank via internet banking services, online authentication, digital documents signage, etc.

There are many proposals of new cryptosystems based on a wide range of problems which are assumed to be computationally hard. One in particular is the subject of this thesis: QC-MDPC McEliece cryptosystem [4], specifically its GF(4) variant conceived by Baldi et al. [1]

The goals of this thesis were to:

- implement the cryptosystem
- design faster decoding algorithms for use in the decryption step
- evaluate their performance

One of the outcomes was a library which is currently used for further research of this cryptosystem and possible attacks which could be mounted against it.

Preliminaries

- **Hamming weight of a vector** v (denoted as $w(v)$) is the count of its nonzero entries.
- **Linear codes** have a generator matrix G and a control matrix H such that $GH^T = 0$.
- A vector v can be **encoded** using the matrix G and **decoded** using a suitable decoder and the matrix H.
- The **syndrome of a vector** v is also a vector (denoted as s). It holds that s is zero if and only if v is a codeword.
- **QC-MDPC McEliece cryptosystem** (both the binary and GF(4) versions) is asymmetric and uses a randomly generated QC-MDPC linear code with certain properties. G serves the purpose of a public key whilst H is the private key.
- The **encryption** of a plaintext is done in two steps: first this message is encoded using the matrix G and then t random errors are intentionally introduced in the encoded message thus creating a ciphertext.
- The **decryption** relies on a suitable decoder to correct these errors. Once corrected, the plaintext can be simply extracted from the message. The decoders tend to be probabilistic in nature and have a nonzero **decoding failure rate (DFR)**, i.e. a successful decoding is not guaranteed.

Current state of decoders for QC-MDPC over GF(4)

Baldi et al. propose a **symbol-flipping (SF)** decoding algorithm for QC-MDPC McEliece over GF(4). First, this decoder calculates the syndrome of a message. Then, it iteratively corrects errors.

During each iteration, for all positions j in the message the value σ_j is calculated as follows: $\sigma_j = w(s) - w(s - aH_{*,j})$, where $H_{*,j}$ is the j^{th} column of H and a is the symbol used for correction. It expresses how much closer the syndrome would be to being a zero vector if the j^{th} position in the message is corrected. In simpler words, it can be conceptualized as the amount of "correction" gained by modifying the position j of the message.

Every iteration, only the position with the highest value σ_j is corrected. It must be emphasized that this does not mean that the j^{th} position truly was erroneous. In fact, should a correct position have the highest value σ_j, the decoder will "correct" it anyways, creating an additional error. At the end of the iteration, the syndrome is updated.

Proposed decoders for QC-MDPC GF(4)

This thesis proposed two new decoders for QC-MDPC over GF(4). First, let us clarify the reason for which more decoders are needed. The SF decoder has a very good performance from the point of view of DFR (as shown in both its author's experiments and ours). This is due to the fact that it only corrects one position each iteration. For the very same reason however, it is very slow. In order to correct t errors, it must run for at least t iterations. The proposed decoders were designed to be much faster in this regard.

The SF decoder is conceptually similar to the bit-flipping (BF) decoder. The BF decoder evaluates each position j in each iteration and calculates the number of unsatisfied parity-check equations upc_j. It corrects the position with the highest upc_j. This decoder also has similar properties to SF, i.e. it is slow with good DFR.

There are other binary decoders which are of interest to this thesis. First, there is a decoder which utilizes an integer parameter δ to correct all positions which have upc_j close to the maximum value in the given iteration. [4] Second, there is a decoder which calculates a threshold based of the hamming weight of the syndrome (as described in [2] and flips all positions with upc_j higher than this threshold.

These decoders serve as an inspiration for our proposed decoders in the same way BF served as inspiration for SF. First, we propose SF decoder with δ. In each iteration, the maximum value σ_j is found. We denote this value σ_{max}. Then, the decoder corrects all positions j for which $\sigma_j \geq \max\{\sigma_{max} - \delta, 0\}$.

Second, we propose a decoder with threshold calculated as a function of syndrome, denoted $T(s)$. In each iteration, the decoder calculates the new value of threshold and corrects all positions j for which $\sigma_j \geq T(s)$. Therefore, there is no need to find σ_{max} in advance, i.e. a single pass over the entire message is needed in each iteration.

Approximation of the threshold function

We identified the threshold function $T(s)$ using numerical simulations. We collected all of the calculated σ_j values across a thousand instances of decoding with the SF decoder. Then, we aggregated them per syndrome weights and separated the data fo each weight into two categories: position "should be" and position "should not be" corrected. Then, we found suitable σ_j values separating the categories. This resulted in a set of points which we used as input for a linear regression algorithm to approximate a linear function $f_0(s) = 0.0248577875w(s) - 29.1143817$. Using this function, we defined multiple threshold functions to use in our experiments in the form of

$$T_i(s) = max\{\lfloor f_0(s) + i \rfloor, 0\}, \text{ where } i \in \{1, 2, 3, 4\}$$

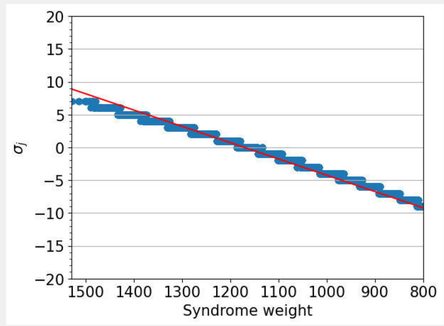
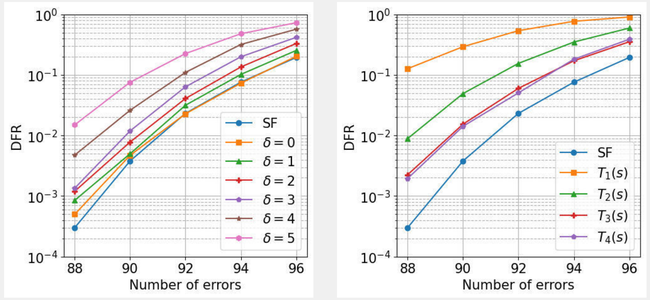


Figure 1. The dividing σ_j values per syndrome weight and the approximated function f_0

Results of experiments

We tested the DFR using a similar methodology to the one used by Baldi et al. in [1]. We used several settings of number of errors t to see the evolution of DFR. For $t = 88$, we decoded 20 000 messages and for the remaining values of t, we decoded 10 000 messages. For the SF decoder with δ, we used $\delta \in \{0, 1, 2, 3, 4, 5\}$. We summarize the results in the figure 2.



(a) DFR of the SF decoder with δ (b) DFR the SF decoder with $T_i(s)$

Figure 2. DFR of proposed decoders

As is to be expected, the decoders perform worse than the SF decoder. However, they are much faster. We measured the number of elapsed iterations for each decoder over 2 000 messages with $t = 84$ errors. We summarize the results in the table 1.

	SF	$\delta = 0$	$\delta = 1$	$\delta = 2$	$\delta = 3$	$\delta = 4$	$\delta = 5$	$\mathbf{T_1(s)}$	$\mathbf{T_2(s)}$	$\mathbf{T_3(s)}$	$\mathbf{T_4(s)}$
Q1	84	50	26	17	13	10	8	3	3	3	4
Q2	84	52	28	19	14	10	9	4	3	4	4
Q3	86	55	31	20	15	11	9	4	4	4	4

Table 1. The 1st, 2nd and 3rd quartile for the measured numbers of iterations of the proposed decoders with $t = 84$

Conclusion

The proposed decoders are substantially faster than the current state-of-the-art decoder for QC-MDPC codes over GF(4) (SF). They can be used as a stepping stone in designing decoders which may be both faster and provide better DFR than the SF. Furthermore, the implementation of this cryptosystem resulting from our work is currently being used in research of GJS attack [3] against it.

References

[1] Marco Baldi, Giovanni Cancellieri, Franco Chiaraluce, Edoardo Persichetti, and Paolo Santini.
Using non-binary ldpc and mdpc codes in the mceliece cryptosystem.
In *2019 AEIT International Annual Conference (AEIT)*, pages 1–6, 2019.

[2] Julia Chaulet.
Etude de cryptosystèmes à clé publique basés sur les codes MDPC quasi-cycliques.
Theses, Université Pierre et Marie Curie - Paris VI, March 2017.

[3] Qian Guo, Thomas Johansson, and Paul Stankovski.
A key recovery attack on mdpc with cca security using decoding errors.
In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, pages 789–815, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[4] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto.
Mdpc-mceliece: New mceliece variants from moderate density parity-check codes.
In *2013 IEEE International Symposium on Information Theory*, pages 2069–2073, 2013.

tomas.vavro@stuba.sk tomas.fabsic@stuba.sk