Code Structures for Quantum Encryption and Decryption

Eric Sakk
Department of Computer Science
Morgan State University
Baltimore, USA
eric.sakk@morgan.edu

Shuangbao Paul Wang
Department of Computer Science
Morgan State University
Baltimore, USA
shuangbao.wang@morgan.edu

***Abstract*—The paradigm of quantum computation has led to the development of new algorithms as well variations on existing algorithms. In particular, novel cryptographic techniques based upon quantum computation are of great interest. Many classical encryption techniques naturally translate into the quantum paradigm because of their well-structured factorizations and the fact that they can be phased in the form of unitary operators. In this work, we demonstrate a quantum approach to data encryption and decryption based upon the McEliece cryptosystem using Reed-Muller codes. This example is of particular interest given that post-quantum analyses have highlighted this system as being robust against quantum attacks. Finally, in anticipation of quantum computation operating over binary fields, we discuss alternative operator factorizations for the proposed cryptosystem.**

Keywords—quantum transforms, quantum computing, transform methods, signal processing

I. Introduction

Post-quantum computational techniques are those that anticipate the development of practical quantum computers. Along these lines, cryptography is an area where several quantum algorithms have been proposed having the potential to challenge classical cryptographic methodologies. In the era of post-quantum cryptography there appear to be two basic areas of focus [1,2]. First, it is important to characterize which existing cryptosystems would be powerful enough to withstand quantum attacks. Second, the development of novel quantum-based cryptographic algorithms must also be considered. This work reviews and addresses, in part, both of these areas.

II. Background

In anticipation of the post-quantum era, various analyses have led to a shortlist of cryptosystems that have the potential to withstand quantum attacks [2]. One of these candidates is the McEliece public key cryptosystem which originally was derived using the theory of error control codes [3,4]. Specifically, a $k \times n$ generator matrix G from a system that can correct up to t errors for which an efficient decoding scheme exists is chosen. A $k \times k$ nonsingular random matrix R and an $n \times n$ permutation matrix P are then chosen to form the matrix

$$\hat{G} = RGP \tag{1}$$

This leads to the cryptosystem parameters

$$\begin{aligned} &\text{Public Key: } (\hat{G}, t) \\ &\text{Private Key: } (R, G, P) \end{aligned} \tag{2}$$

Encryption of a $1 \times k$ bit message vector μ using the public key is accomplished via the prescription

$$c = \mu\hat{G} + v \tag{3}$$

where v is random binary error vector containing at most t ones. The encryption process is illustrated in Figure 1. Determining μ from c is known to be an NP hard problem.

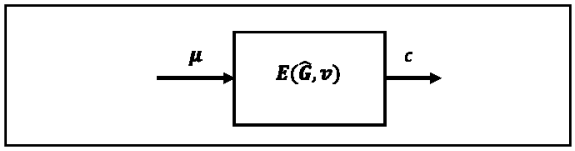


Fig. 1. McEliece encryption using (1) and (3).

The decryption process uses the private key to compute

$$\hat{c} = cP^{-1} \tag{4}$$

Then, *assuming an efficient decoding algorithm exists for the code generated by* G, $\hat{\mu}$ is decoded from $\hat{c}$. Finally, the original message can be recovered

$$\mu = \hat{\mu}R^{-1} \tag{5}$$

While this system is considered to be computationally secure, practical classical objections to the system reside mainly in the size of the key. However, in the post-quantum era, the large key size is precisely what leads to the system being considered to be quantum resistant [4], [5].

The idea of applying linear code-based encryption schemes is regularly revisited [7]-[9]. If the McEliece cryptosystem is a worthwhile candidate for post-quantum techniques, the question arises as to which generator matrices with efficient decoding algorithms can be used for the construction of quantum resistant systems. Prior to quantum computation coming into the picture, Goppa codes were originally suggested as the McEliece cryptosystem generator. However, in response to more recent cryptanalyses, other families of codes have been proposed [4,5]. Reed Muller codes have appeared as one of these candidates,

978-1-7281-8620-7/21/$31.00 ©2021 IEEE

and it is this family that we wish to explore [6]. However, our goal is not to address their classical implementation. Instead, we wish to consider the development of novel quantum-based cryptographic algorithms. As a stepping stone toward this goal, we will focus on the implementation of an efficient quantum Reed Muller decoder for the McEliece system.

III. Quantum Reed Muller Codes

Much groundwork for quantum Reed Muller codes has been established. In this section, we will discuss applying Reed Muller codes as the basis for a quantum-based decoder (QD) (see Figure 1) as well as a quantum-based encoder (QE) for the McEliece system. Decoding of these codes is usually presented by transforming typical parity check rules into operations involving universal quantum gates [10], [11].

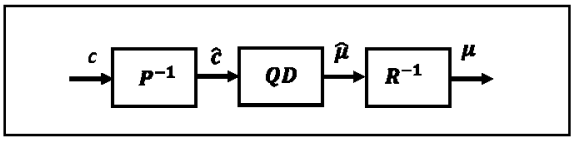


Fig. 2. McEliece decryption using (4) and (5).

A. RM(1,m) Quntum Decoder (QD) Implementation

As long as the binary codewords $\hat{c}$ can be encoded as qubits that are input to QD in Figure 1, we propose an efficient decoder implementation that will not require directly referring to the parity check space [17]. First order Reed Muller codes RM(1,m) have codeword lengths $n = 2^m$ and are known to be decodable using to the Hadamard transform. The generator matrix for such a code is quite straightforward. For example,

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

is an example of a generator matrix for an RM(1,3) code. Notice that the columns for a 1st order RM code count in binary from 0 to 2^{m-1} with the top rows equal to all ones. Obviously, the generator matrix for the McEliece cryptosystem requires a very large value of m.

Given an n-bit binary codeword row vector

$$c = (c_0, c_1, \cdots, c_{n-1})$$

define

$$\mathcal{F}(c) = (-1)^c$$

where the new vector is formed using element-by-element exponentiation yielding the rule

$$0 \rightarrow 1$$
$$1 \rightarrow -1$$

for each component of the codeword c.

Next, let

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

be a 2x2 Hadamard matrix. This operator is unitary

$$H^{\dagger}H = I$$

which implies that it can be used for quantum computational operations [21]. It is, in fact, a valid circuit element typically applied to single qubit systems in quantum computation. For example,

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

and

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

represent the action of H on single qubits states $|0\rangle$ and $|1\rangle$. To extend operators to work with multiple qubit systems, tensor products of qubit states must be constructed using the Kronecker product. Specifically, the Kronecker product of an mxn matrix A and a pxq matrix B is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}$$

Quantum computation builds up qubit spaces using this operation. This tensor product construction along with the time evolution of quantum systems is effectively what enables a quantum computer to act as a highly parallel processor.

The operator iteratively defined by

$$H_m = H_{m-1} \otimes H$$

(where H_m is the Kronecker product of 'm' Hadamard matrices) is highly relevant to the quantum implementation of a Reed Muller decoder. For the RM(1,m) code, calculating the Hadamard transform of a codeword

$$\mathcal{H}(c) = \mathcal{F}(c)H_m \tag{6}$$

is equivalent to correlating a codeword with each column of H_m. Furthermore, and most importantly for this work, up to a reordering of the codewords and a multiplicative constant, it is also true that

$$\mathcal{F}(RM(1,m)) = \begin{bmatrix} H_m \\ -H_m \end{bmatrix} \tag{7}$$

when $\mathcal{F}(c)$ is applied to every codeword in RM(1,m). Under these circumstances, the Hadamard transform of all RM(1,m) codewords reduces to

$$\mathcal{H}(RM(1,m)) = \mathcal{F}(RM(1,m))H_m = \begin{bmatrix} I_m \\ -I_m \end{bmatrix} \tag{8}$$

(up to a multiplicative constant) where I_m is the Kronecker product of 2x2 identity matrices. Efficient maximum likelihood decoding of an R(1,m) codeword $\hat{c}$ to the message $\hat{\mu}$ can be performed by analyzing $\mathcal{H}(\hat{c})$. Specifically, the component of $\mathcal{H}(\hat{c})$ with the greatest magnitude can directly related back to the to the message $\hat{\mu}$.

An equivalent quantum-based decoder immediately follows from this observation. If, in Figure 2, $\hat{c}$ is presented to QD in the form of qubits, (7) then tells us that $\mathcal{F}(\hat{c})$ can be generated via the Hadamard transform. It then follows from (8) that the message $\hat{\mu}$ can be recovered. One issue with this approach is that (8) involves negating the identity. Further research of this approach will involve integrating the unitary operator

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

in order to integrate the negation operation. For this work, we simply consider half the set of codewords so that (8) reduces to

$$\mathcal{F}\left(\widetilde{RM}(1,m)\right)H_m = I_m \qquad (9)$$

where $\widetilde{RM}(1,m)$ refers to the reduced code. Table I provides a list of parameters that easily extend to any number bits.

TABLE I PARAMETERS FOR QUANTUM $R(1,M)$ DECODING

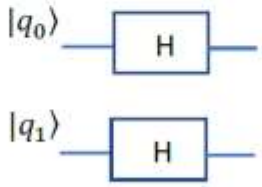
m	k=m+1	n=2^m	# qubits, m	# codewords, 2^{m-1}
2	3	4	2	2
3	4	8	3	4
4	5	16	4	8
5	6	32	5	16
6	7	64	6	32
7	8	128	7	64
8	9	256	8	128

B. *Quntum Circuit for Quantum Decoder (QD) Implementation*

Quantum circuits are schematic representations of quantum operations on qubits. The identity

$$(A\otimes B)(C\otimes D) = (AC)\otimes(BD)$$

(where the matrix multiplication is well-defined for matrices A,B,C,D) is of great importance in understanding how to build up multiple qubit circuits. For example, the H operation applied to two distinct qubits $|q_0\rangle$ and $|q_1\rangle$ is depicted in the following quantum circuit



The above identity can be used to mathematically characterize its schematic representation

$$H|q_1\rangle\otimes H|q_0\rangle = (H\otimes H)(|q_1\rangle\otimes|q_0\rangle|)$$

Furthermore, tensor products of qubits are abbreviated using the following notation

$$|q_1\rangle\otimes|q_0\rangle| = |q_1q_0\rangle$$

leading to the more compact circuit description

$$H|q_1\rangle\otimes H|q_0\rangle = (H\otimes H)|q_1q_0\rangle.$$

Taking successive Kronecker products to build up the Hadamard decoder naturally follows. Figure 3 shows a quantum circuit for RM(1,4). The M operations in this figure depicts quantum measurements in order to determine the decoded bit stream. Finally, Figure 4 shows that a message 0110 has been recovered when QD is presented with the associated binary codeword.

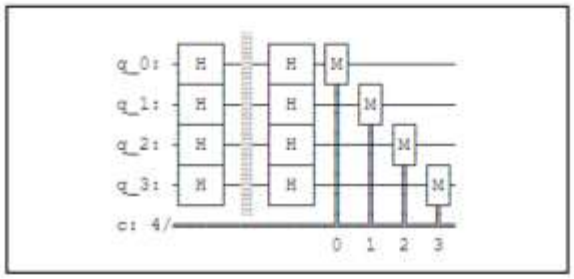


Fig. 3. Qiskit Implementation of RM(1,4) quantum decoder QD

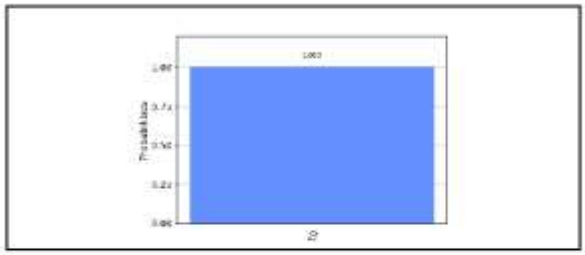


Fig. 4. Input codeword decoded to message 0110

C. *Quantum-based Encoder (QE) for Reed Muller Codes*

Quantum gates for quantum circuits are generally implemented using the typical universal set of unitary transformations necessary for quantum computation [21]. Such calculations necessarily involve complex fields given the nature of quantum calculations. This implies, more often than not, computations must be 'coaxed' into an answer that reflects the mathematics of a specific problem.

We would suggest that a worthy pursuit would be to insulate the quantum programmer from this final step. One possible solution might be to create a computational environment that reflects a binary field so that computations involving binary numbers could readily be phrased on a quantum computer. Excellent work along these lines has already been established [12-14]. We project that this will gradually become the norm for quantum programming. If this is the case, then Reed Muller codes could be constructed from fundamental quantum operations involving tensor products involving finite fields. In anticipation of this result, we point out how to generate such codes in a vector space over binary numbers [17].

Consider the following definitions

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

such that

$$P_m = P \otimes P \cdots \otimes P \mod 2$$

Is the Kronecker product of P with itself m times taken mod 2. This type of calculation is quite common within quantum computation when operators are unitary. Over the binary field, it should be clear that

$$P = P^T$$
$$PP^T = I \mod 2$$

Making P an excellent candidate gate operation for quantum computation over binary fields. Under these circumstances, Reed Muller codes can be generated from P_m. Specifically, a generator matrix for a code with minimum distance d_{min} can be constructed by extracting all rows from P_m having a Hamming weight $w_H \geq d_{min}$ [18]. In terms of a quantum circuit, this amounts to extracting the appropriate qubits after the tensor product P_m mod 2 has been applied.

In addition, in anticipation of quantum computation operating over binary fields, we also briefly point out that interesting wavelet packet formulations of Reed Muller codes can be generated via various 'perfect shuffle' factorizations. Many such factorizations have been proposed in the literature since the inception of quantum computation [15-20]. We briefly present one that is useful for encoding and decoding Reed Muller codes based upon tensor products over finite fields. For the ith scale in a wavelet packet decomposition, letting S^i represent a perfect shuffle, the decomposition T_m can be formulated as

$$T_m = P_m \Phi_m$$

where

$$\Phi_m = \Phi_m^{(0)} \Phi_m^{(1)} \ldots \Phi_m^{(m-1)}$$
$$\Phi_m^{(i)} = I_{m-i-1} \otimes S^i.$$

IV. Conclusions

In the era of post-quantum cryptography, it is important to (i) characterize cryptosystems powerful enough to withstand quantum attacks and (ii) consider novel approaches to quantum-based cryptography. The McEliece system is considered to have the potential to withstand quantum attacks. In light of (ii), it is natural to pursue whether or not there exists a novel approach where this system can be implemented completely on a quantum computer. The random and permutation matrices within the system, in principal, could be integrated. Given the results of this work, efficient encoding and decoding using a linear code-based encryption scheme appears to be plausible as well. Specifically, we have applied quantum-based Reed Muller codes to demonstrate feasibility. An efficient decoding approach has been introduced involving RM(1,m). Furthermore, if extensions to finite field quantum computations can be realized, we have introduced efficient encoding process for this family of codes as well. The field of quantum computation appears to be in a state similar to when analog computers were eclipsed by digital computers. One possible key to this transition may be to define and generate unitary systems that are useful over finite fields.

Acknowledgment

This research is funded by a grant from National Science Foundation #1560214 (2020).

References

[1] D.J. Bernstein and T. Lange, Post-quantum cryptography, Nature, 549, p188-194, (2017).

[2] D. Augot, et al. Initial recommendations of long-term secure post-quantum systems. PQCRYPTO: Post-Quantum Cryptography for Long-Term Security (7 September 2015).

[3] J.D. Bernstein. Grover vs. McEliece. Post-quantum cryptography 2010. Lecture Notes in Computer Science. 6061. Sendrier, Nicolas (ed.). Berlin: Springer. p73–80.

[4] H. Dinh et al. Quantum Fourier Sampling, Code Equivalence, and the Quantum Security of the McEliece and Sidelnikov Cryptosystems, ArXiv, 2018, arxiv.org/pdf/1111.4382.pdf.

[5] Y. Wang. Quantum resistant random linear code based public key encryption scheme RLCE. 2016 IEEE International Symposium on Information Theory (ISIT), p2519-2523 (2016).

[6] J. Elder, Quantum resistant Reed Muller codes on McEliece cryptosystem, Ph.D. Thesis, University of North Carolina at Charlotte, 2020.

[7] K. Khathuria, J. Rosenthal and V. Weger, Encryption scheme based on expanded Reed-Solomon codes, Advances in Mathematics of Communications, doi: 10.3934/amc.2020053, 2019.

[8] C. T. Gueye and E. Mboup, Secure Cryptographic Scheme based on Modified Reed Muller Codes, International Journal of Security and Its Applications, Vol. 7, No. 3, May, 2013.

[9] U. Neelima and F. Noorbasha, Data encryption and decryption using Reed-Muller techniques, International Journal of Engineering and Technology, 8(1), p83-91, Feb-Mar 2016.

[10] A.M. Steane, Quantum Reed–Muller Codes, IEEE Transactions on Information Theory, 45(5), p1701-1702, 1999.

[11] J. T. Anderson, G. Duclos-Cianci, and D. Poulin. Fault-Tolerant Conversion between the Steane and Reed-Muller Quantum Codes. Phys. Rev. Letters, 113, August 2014.

[12] C. Dawson, H. Haselgrove, A. Hines, D. Mortimer, M. Nielsen, and T. Osborne. Quantum computing and polynomial equations over the finite field Z_2. Quantum Inf. Comput., 5(2):102-112, 2005.

[13] A. Montanaro., Quantum circuits and low-degree polynomials over F2, Journal of Physics A: Math. and Theor. 50, 2017.

[14] S. Gangopadhyay, V. S. Poonia, D.a Aggarwal and R. Parekh. Generalized Boolean Functions and Quantum Circuits on IBM-Q, 10th ICCCNT, July 6 -8, 2019 - IIT, Kanpur.

[15] P. Hoyer, P. Efficient quantum transforms. arXiv preprint quant-ph/9702028 (1997).

[16] A. Fijany and C.P. Williams, Quantum wavelet transforms: Fast algorithms and complete circuits. Lect. Notes Comput. Sci. 1509, 10-33 (1998)..

[17] E. Sakk, Wavelet packet formulation of generalized Reed-Muller codes, Ph.D. Thesis, Cornell University, Ithaca, NY, USA, 2002.

[18] E. Sakk and S. B. Wicker, Wavelet packets for error control coding, Proc. SPIE 5207, Wavelets: Applications in Signal and Image Processing X, (13 November 2003).

[19] H.-S. Li, P. Fan, H.-Y. Xia, S. Song and X. He The multi-level and multi-dimensional quantum wavelet packet transforms. Sci Rep 8, 13884 (2018).

[20] S. Wang, M. Rohde & A. Ali. Quantum Cryptography and Simulation: Tools and Techniques. ACM Proc. of International Conference of Cryptography, Security and Privacy (ICCSP). pp. 36-41. 2020.

[21] A. Nielsen and I.L. Chuang. Quantum Computation and Quantum Information., Cambridge University Press, 2011.