Hamming Quasi-Cyclic Encryption Schemes

Jacob Ryder, Rochester Institute of Technology

Abstract

This paper is a review of the Hamming Quasi-Cyclic schemes submitted by Gaborit, Deneuville, et al. in the National Institute for Standards and Technology competition for post-quantum public key encryption. The schemes are based upon coding theory and rely on the hardness of the syndrome decoding problem.

Keywords: HQC, NIST, Post-Quantum, Coding Theory

1. Introduction

Quantum computing has been an area of interest in computer science for many decades. Quantum algorithms would be capable of non-deterministic analysis, evaluating an exponentially growing set of possibilities simultaneously and solving traditionally NP problems in polynomial time. Grover's Algorithm, published in 1996, is a quantum-based general database search which runs in $O(\sqrt{n})$ as opposed to the traditional $O(n)$, and other quantum algorithms tailored to specific tasks (such as Shor's for integer factorization) achieve even higher efficiency. Since the most common asymmetric schemes in modern cryptography rely on the computational hardness of integer factorization and discrete logarithms, the growing capabilities of strong attacks against these NP problems are fascinating yet concerning.

In 2016, the National Institute of Standards and Technology began accepting proposals for a post-quantum cryptography standard, not dissimilar from the competition for the Advanced Encryption Standard starting in 1997. Hamming Quasi-Cyclic (HQC) is one of these submissions, proposed by Aguilar Melchor, Arahon, Bettaieb, Bidoux, Blazy, Deneuville, Gaborit, Persichetti and Zémor. These authors have a great deal of experience in the field of code-based cryptography, and several contributed to the related Ouroboros encryption scheme.

The security of HQC is reliant on the computational hardness of the syndrome decoding problem, and its Public Key Encryption (HQC.PKE) and Key Encapsulation Mechanism (HQC.KEM) schemes achieve security and efficiency through computations in the context of systematic quasi-cyclic Bose-Chaudhrui-Hocquenghem (BCH) codes. Notable advantages of this model include precise estimations of its security, quantum resistance comparable to lattice-based cryptography, and efficient key sizes relative to most code-based cryptography.

Though this report aims to summarize and analyze the HQC schemes, the original publication does an excellent job of this as well. The complete specification for HQC is published at *http://pqc-hqc.org*, and provides more comprehensive explanations in some areas.

2. Definitions

To explain an encryption scheme based on coding theory, it is prudent to first review coding theory. The following definitions describe relevant aspects of linear codes, the specific code structures in HQC, and the syndrome decoding problem. Information is drawn from documentation on HQC and Ouroboros.

2.1. Linear Codes

The primary domain for this system is a finite binary Galois field $R = F_2^k[X]/(x^k - 1)$, where k denotes the dimensionality of the field. A Linear Code C is a subspace of R with length n and dimensionality k, and is denoted $[n, k]$. The n different vectors $c \in R$ of C are referred to as codewords of C.

A matrix $G \in F_2^{k \times n}$ is considered a Generator Matrix for the $[n, k]$ code C if the product of any vector m and G yields a codeword c, or more formally if $C = \{mG, \forall m \in F_2^k\}$. The codewords of C span the rows of G.

Multiplying a vector m by G is called encoding m; resolving codeword c back into m is called decoding.

A matrix $H \in F_2^{n-k \times n}$ is considered a Parity Check Matrix for the $[n, k]$ code C if the product of H and any codeword $c \in C$ is 0, i.e. $Hc^T = 0$, $\forall c \in C$. This is equivalent to stating that parity check matrix H is the generator matrix for the dual of C.

The product of H and a vector $v \in F_2^k$ is called the Syndrome of v, and the syndrome of v is 0 iff $v \in C$.

For a code C and a norm function ω over R, the Minimum Distance of C is given by $d = \min \omega(u - v)$ for $u, v \in C, u \neq v$, or the smallest distance between two vectors. A code with minimum distance d is capable of uniquely decoding a pattern within $\delta = \lfloor \frac{d-1}{2} \rfloor$ of a codeword, i.e. a codeword with up to δ errors.

2.2. Systematic Quasi-Cyclic, and BCH Codes

Each codeword c in an $[sn, k]$ linear code C can be represented as a set of s n-tuples, or blocks. This code is considered Quasi-Cyclic of index s iff a simultaneous circular shift on every block in the codeword c yields another codeword in C. Using this property, we can represent cyclic matrices in $1/s$ the space without loss of information; since code-based cryptography traditionally suffers from inordinate key lengths, shortening the code can be a substantial advantage.

A linear code which is Systematic requires a parity check matrix with a specific format: for an $[sn, n]$ code C which is quasi-cyclic of index s, the parity check matrix is an $(s-1)n \times sn$ matrix formed of an identity matrix $I_{(s-1)n}$ and the last n columns populated by $n \times n$ circulant matrices (which are represented of a single $1 \times n$ vector with all possible shifts). This format allows the parity check matrix to be conveyed in terms of $s - 1$ $1 \times n$ vectors without loss of informations, and similarly offers substantial advantages in overall key length. For $s = 2$, the parity check matrix can be conveyed with a single vector.

BCH codes are a well-studied type of self-correcting codes with can be constructed with controls over the error limit δ that they can correct, as well as relatively simple methods for decoding codewords into vectors. These properties allow for precise evaluation on the failure rates of decoding, which in turn allows for formal demonstrations of each scheme's security.

The Tensor Product of two codes C_1, C_2 is represented as $C_1 \bigotimes C_2$ and defined as the set of all $n_2 \times n_1$ matrices whose rows are codewords of C_1 and whose columns are codewords of C_2. For a specific tensor product, encoding a message as the tensor is a matter of encoding as each code in series, while decoding is the inverse. This allows for further tuning of a code's parameters while maintaining its decoding properties.

The HQC schemes are based on Systematic Quasi-Cyclic BCH codes tensored with simple repetition codes. These products take the form $C = BCH(n_1, k, d) \bigotimes 1_{n_2}$, and though they are not proven to be the most efficient from a decoding perspective, they do lend themselves to very accurate estimations of their decoding failure rates and subsequently to very accurate security models for each scheme.

2.3. The Syndrome Decoding Problem

The Hamming Weight $\omega(x)$ of a vector x is equal to the number of non-zero coefficients it contains. Within a binary field, this is a simple yet meaningful metric.

Given an $[n, k, \delta]$ code C with parity check matrix H, a codeword $c \in C$, and an error pattern $e \in F_2^k$, the combination of c and e is the word $v = c + e$. Since the syndrome of c is 0 by definition, the syndrome of v is equal to the syndrome of e. v can be successfully decoded as c if $\omega(Hv^T) \leq \delta$. The process of correcting v to c and decoding the message is referred to as Syndrome Decoding.

The Syndrome Decoding (SD) problems are defined as follows: A $\mathrm{SD}(n, k, w)$ Distribution produces a parity check matrix $H \in F_2^{(n-k)\times k)}$ for an [n,k] code C and a word $x \in F_2^k$ with $\omega(x) = w$. The matrix H and the syndrome $y = Hx^T$ are published. The Search SD Problem is to find the vector $x \in F_2^k$ such that $Hx^T = y^T$ and $\omega(x) = w$, while the Decision SD Problem is to determine with advantage whether the pair (H, y^T) came from the SD distribution or the uniform distribution over $F_2^{(n-k)\times k}$.

The translation from the full. SD problem to the case of Systematic Quasi-Cyclic codes (s-QCSD) is not entirely equivalent, but these constraints have been demonstrated to only minimally affect generality. No known attacks utilizing the cyclic structure of these codes reduce the overall practical complexity of these problems, while the performance advantages they offer are substantial.

The Syndrome Decoding problems have been shown to be NP-complete, as the best general solutions require searching over all 2^k possible vectors, requiring a runtime based exponentially on the length of the input. This assertion is verified through comparison to the Learning Parity with Noise problem, which is believed to be computationally hard. The quantum resistance of this family of problems is rooted in the complexity of their underlying mathematics; even with the non-deterministic runtime advantages that quantum computing would offer, the time required to solve these problems is deemed sufficiently intractable.

3. Hamming Quasi-Cyclic Public Key Encryption

As explained in section 2.2, the HQC schemes operate using Systematic Quasi-Cyclic BCH codes tensored with repetition codes. This allows for efficient decoding of syndromes with $\omega(v) \leq \delta$, but when the weight of the error pattern exceeds this threshold the problem becomes intractable.

The Public Key Encryption scheme (HQC.PKE) relies on masking encoded messages with non-deterministic error vectors based on a public/private key pair. The code C being utilized and its corresponding generator matrix G are published; the security of the system does not rely on their secrecy, only on the key pair.

The PKE scheme is made of four polynomial-time operations: Setup, KeyGen, Encrypt, and Decrypt.

3.1. Setup

- The global parameter tuple $\{n, k, \delta, w, w_r, w_e\}$ is selected and published.
- The first three parameters define the $[n, k, \delta]$ code being used.
- The latter three are weights required for later vector sampling.

3.2. KeyGen

Goal: generate keypair (pk, sk) and publish pk.

- The generator matrix G for the $[n, k, \delta]$ code C is created and published.
- Secret key $sk = (x, y)$, where $x, y \leftarrow R^2$ with $\omega(x) = \omega(y) = w$.
- Public key $pk = (h, s)$, where nonce $h \leftarrow R$ and value $s = x + h \cdot y$.

3.3. Encrypt - $E(pk, m, \theta) = c$

Goal: using $pk = (h, s)$, convert plaintext message $m \in R$ into ciphertext c.

- Nonces $r_1, r_2 \leftarrow R$ with $\omega(r_1) = \omega(r_2) = w_r$.
- Nonce $e \leftarrow R$ with $\omega(e) = w_e$.
- Value $u = r_1 + h \cdot r_2$.
- Value $v = mG + s \cdot r_2 + e$.
- Ciphertext $c = (u, v)$.

3.4. Decrypt - $D(sk, c) = m$

Goal: using $sk = (x, y)$, convert ciphertext $c = (u, v)$ into original message m.

- $m = \text{C.Decode}(v - u \cdot y)$

Decryption is possible only with knowledge of k. The expression $v - u \cdot y$ expands to $mG + x \cdot r_2 - r_1 \cdot y + e$, and can be decoded iff $\omega(x \cdot r_2 - r_1 \cdot y + e) \leq \delta$. The failure rate of this decryption depends on the parameters of the system, and is presented in section 5.

The authors of HQC prove through an analysis of adversarial games that the PKE scheme achieves Indistinguishability under Chosen Plaintext Attacks (IND-CPA), i.e. that an attacker has negligible advantage in guessing the plaintext corresponding to a ciphertext even between two known plaintexts.

4. Hamming Quasi-Cyclic Key Encapsulation Mechanism

The HQC.PKE scheme is constructed to perfectly fit the requirements of the Fujisaki-Okamoto transformation, which converts an IND-CPA PKE scheme into an Indistinguishable under Adaptive Ciphertext Attack (IND-CCA2) Key Encapsulation Mechanism (KEM).

Most of the KEM relies directly on the functions detailed in the PKE scheme, but also includes a set of secure hash functions H_1, H_2, and H_3. The purpose of the KEM is to have both parties derive a secret key from a shared seed and the encrypted value of the seed, so the secure transmission of this seed is vital.

The Setup and KeyGen functions are as in PKE. Encapsulation and Decapsulation are defined as follows:

4.1. Encapsulation

Goal: using pk, securely transmit a seed m for a shared key.

- Seed $m \leftarrow R$ is selected.
- Randomness $\theta = H_1(m)$ is used to derive e, r_1, r_2.
- The seed is encrypted as $c = E(pk, m, \theta)$.
- The shared key is derived as $k = H_3(m, c)$.
- The values $(c, d = H_2(m))$ are shared.

Using $H_1(m)$ to generate the security nonces limits some of the non-deterministic elements of the encryption, but does allow both parties to compute the same ciphertext. Information may be leaked if the same hashing algorithm are used for H_1 and H_2, but otherwise the nonces will remain secret.

4.2. Decapsulation

Goal: using sk, retrieve seed m and derive the shared key.

- The ciphertext is decrypted as $m' = D(sk, c)$.
- The seed is reencrypted as $c' = E(pk, m', H_1(m'))$ and $d' = H_2(m')$.
- If $c \neq c'$ or $d \neq d'$, the seed is rejected.
- Otherwise, the shared key is derived as $k = H_3(m', c')$.

As previously mentioned, HQC.KEM is provably secure against chosen plaintext and adaptive ciphertext attacks, giving it a great deal of overall security. Though the addition of random noise in PKE and KEM does introduce a chance to fail decryption, this chance is well-controlled by the parameters offered and will rarely if ever pose an issue to the scheme's use.

5. Parameters and Performance

Table 1 shows the sets of recommended parameters published by the authors of the HQC schemes. The parameters n_1 and n_2 refer to the sizes of the BCH and Repetition codes, and n is the smallest primitive prime greater than $n_1 \cdot n_2$. While n_1 stays mostly consistent, changes in n_2 and weights w, w_e, w_r create differing security levels and worst-case decryption failure rates. The dimensionality k is consistent across all parameter sets. Values are in bits.

Instance	n_1	n_2	n	k	δ	w	w_r, w_e	Security	p_{fail}
Basic-I	766	29	22229	256	57	67	77	128	$< 2^{-64}$
Basic-II	766	31	23747	256	57	67	77	128	$< 2^{-96}$
Basic-III	796	31	24677	256	60	67	77	128	$< 2^{-128}$
Advanced-I	796	51	40597	256	60	101	117	192	$< 2^{-64}$
Advanced-II	766	57	43669	256	57	101	117	192	$< 2^{-128}$
Advanced-III	766	61	46747	256	57	101	117	192	$< 2^{-192}$
Paranoiac-I	766	77	59011	256	57	133	153	256	$< 2^{-64}$
Paranoiac-II	766	83	63587	256	57	133	153	256	$< 2^{-128}$
Paranoiac-III	796	85	67699	256	60	133	153	256	$< 2^{-192}$
Paranoiac-IV	766	89	70853	256	60	133	153	256	$< 2^{-256}$

Table 2 shows the sizes of the private keys, secret keys, ciphertexts, and shared secrets in the HQC.KEM model. The first (pk, sk) pair is for the base implementation, the second utilizes NIST's seed expander with 40-byte seeds. Ciphertext and shared secret lengths are the same in both cases. All values are given in bytes.

Instance	pk size	sk size	pk size	sk size	ct size	ss size
Basic-I	5558	252	2819	40	5622	64
Basic-II	5938	252	3009	40	6002	64
Basic-III	6170	252	3125	40	6234	64
Advanced-I	10150	404	5115	40	10214	64
Advanced-II	10918	404	5499	40	10982	64
Advanced-III	5884	404	5884	40	11752	64
Paranoiac-I	14754	532	7417	40	14818	64
Paranoiac-II	15898	532	7989	40	15962	64
Paranoiac-III	16926	566	8503	40	16990	64
Paranoiac-IV	17714	566	8897	40	17778	64

Table 3 shows the timing data for the reference implementation of HQC.KEM, collected on the benchmark computing platform defined for the NIST application. Values are given first in milliseconds, then in millions of CPU cycles. The results here seem fairly promising, with lighter security options executing quickly and heavier security only having a moderate impact.

Instance	KeyGen	Encaps	Decaps	KeyGen	Encaps	Decaps
Basic-I	0.17	0.36	0.57	0.57	1.22	1.95
Basic-II	0.18	0.38	0.63	0.61	1.28	2.07
Basic-III	0.19	0.40	0.61	0.63	1.35	2.15
Advanced-I	0.37	0.77	1.13	1.26	2.61	3.82
Advanced-II	0.40	0.83	1.21	1.37	2.81	4.11
Advanced-III	0.43	0.89	1.28	1.47	3.02	4.35
Paranoiac-I	0.65	1.38	1.96	2.21	4.67	6.67
Paranoiac-II	0.76	1.60	2.22	2.52	5.37	7.51
Paranoiac-III	0.78	1.65	2.35	2.66	5.62	8.03
Paranoiac-IV	0.82	1.76	2.50	2.81	5.95	8.46

6. Conclusions

The Hamming Quasi-Cyclic encryption schemes present an efficient and secure model for public key cryptography, and offer an interesting perspective into the domain of coding theory. Though code-based cryptography holds many similarities to lattice-based cryptography (and correspondingly, the Syndrome Decoding problem is similar to the Learning With Errors problem), the optimizations enabled through the specific code structure presented here offer promising performance advantages.

A great many lattice-based encryption schemes were submitted for NIST to consider. Though lattice-based cryptography is not a new field, it has not seen much attention until recently. The smaller key spaces exemplified by integer factorization and discrete logarithm schemes are more desirable for traditional models, but may therefore be insufficient in a post-quantum landscape.

At the moment, the solid balance between security and efficiency makes it seem likely that one of these lattice-based (or code-based) solutions is likely to emerge victorious, but it is yet unclear which is superior overall. Then again, there is a chance that the increased attention may lead to the discovery of new and powerful attacks against lattices, which would also be fascinating yet concerning.

The cryptographers involved in this proposal have done an excellent job not only in designing an effective and efficient scheme, but also in thoroughly documenting their designs from high-level theory down to technical implementations. With any luck, this interesting work by Deneuville, Gaborit, Zémor and their colleagues will soon be recognized and ratified, and these researchers will be even better known.

References

Gaborit, J.-C. D. P., & Zmor, G. (2017). Ouroboros: a simple, secure and efficient key exchange protocol based on coding theory. 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings. `http://www.unilim.fr/pages_perso/deneuville/files/ba43bf8d80cef2999dbf4308828213ec.pdf`.

Hofheinz, D., Hvelmanns, K., & Kiltz, E. (2017). A modular analysis of the fujisaki-okamoto transformation. Cryptology ePrint Archive, Report 2017/604. `https://eprint.iacr.org/2017/604`.

McEliece, E. R. B. R. J., & van Tilborg, H. C. (1978). On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory. `https://authors.library.caltech.edu/5607/1/BERieeetit78.pdf`.

Persichetti, A. M. A. B. B. B. D. G., & Zémor (2017). Hamming quasi-cyclic (hqc). NIST Post-Quantum Cryptography Standardization Project. `http://pqc-hqc.org/doc/hqc-spec.pdf`.

Stinson, D. R. (2006). *Cryptography: Theory and Practice*. (3rd ed.). Taylor & Francis Group, LLC.