Dispersed Cryptography and the Quotient Ring Transform

Anna M. Johnston
drannajohnston at gmail dot com

February 14, 2017

Abstract

This paper describes a radically different privacy, security and integrity solution. Dispersed Cryptography converts the cloud from a security threat into a security asset by combining a standard stream cipher and the Quotient Ring Transform (QRT). The result is an integrated error correction/encryption algorithm. This encoding disperses data, breaking it into many smaller pieces and scattering them to different sites. No single site is critical; any can be lost without losing data. No single site can access data, even if the cryptovariable (secret key) is compromised.

The resulting system is more flexible and seamlessly adds both data integrity and security. The underlying codes are linear, and therefore have homomorphic properties and may be used in coding based quantum resistant cryptography.

Keywords: encryption, encoding

1 Introduction

Dispersed Cryptography integrates encryption and error correction capabilities into a single cipher-coding system. This system disperses and decentralizes both security and integrity.

Assume a message m needs to be transmitted from user 1 to user 2. Messages are not usually sent directly (figure 1). For example, email is sent through a host, cloud storage is stored on a host before being retrieved.

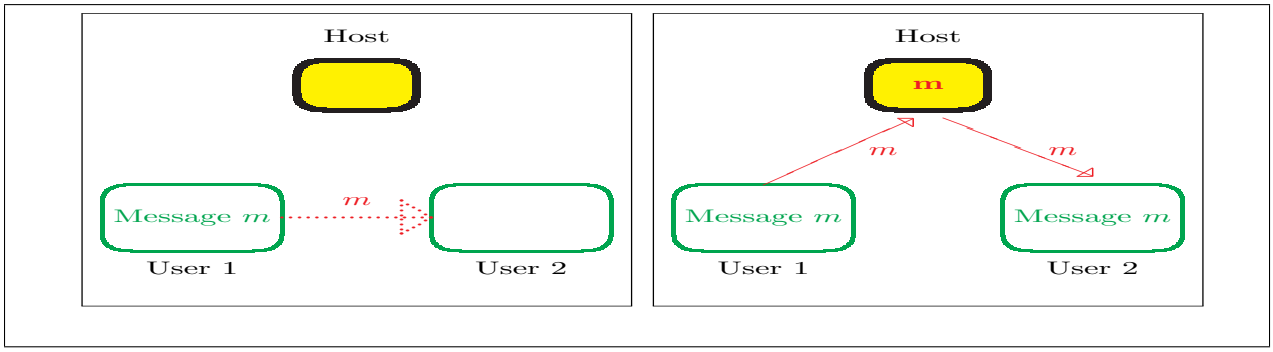


Figure 1: User 1 wants to send user 2 message m. Data passes through host.

Dispersed cryptography takes a different view. Instead of sending the entirety of the message through a single host, it encodes the message into n smaller pieces which are transmitted to n hosts. These hosts then transmit their piece to user 2 (figure 2). This decentralizes both security and integrity.

Integrity: Data at individual hosts can be lost without losing the integrity of the data (figure 3). This also helps stop ransomware attacks, as an attacker would have to destroy or encrypt data at many sites to prevent recovery.

Security: The underlying cryptography is the main security component of the system. However, if the cryptovariable (key) was compromised an attacker would still have to access data from multiple hosts to recover the data.

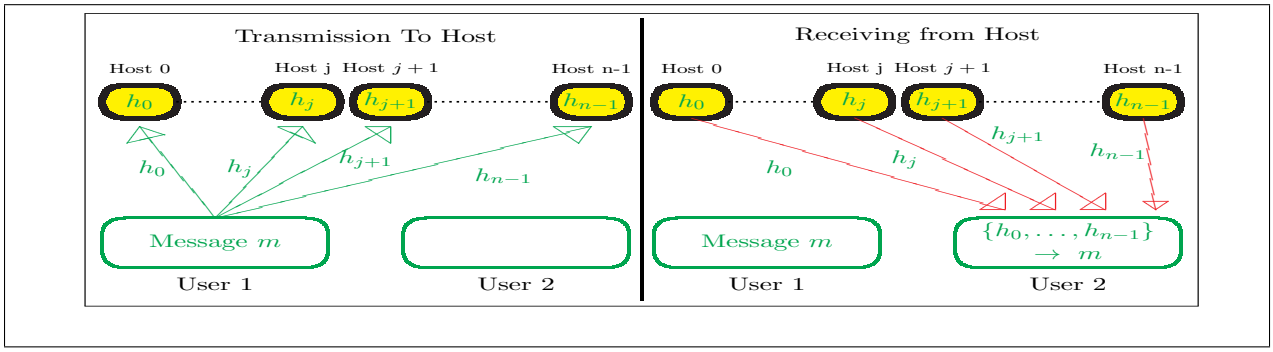


Figure 2: User 1 encodes m into n smaller pieces and transmits to n hosts.

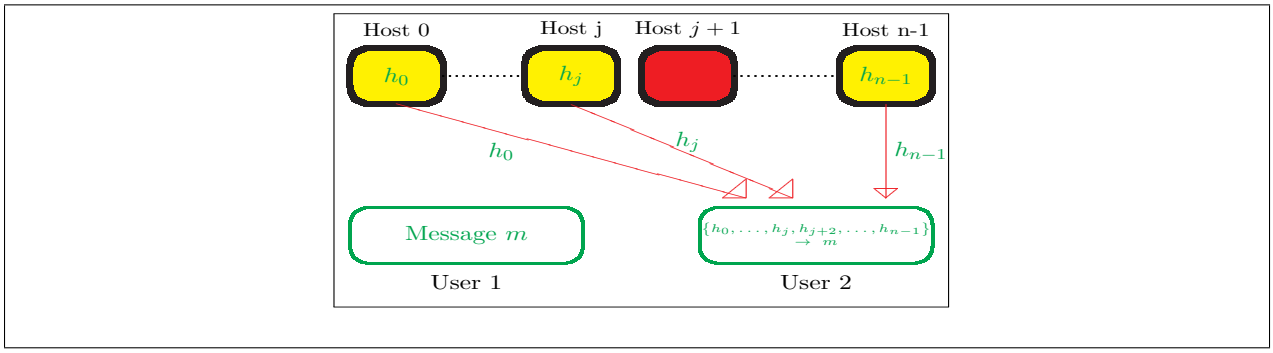


Figure 3: User 2 can still get message even if some number of hosts are lost

Dispersed Cryptography's integrated cipher-coding uses a standard cryptographic stream cipher[1] and a new coding process, the Quotient Ring Transform - QRT (definition 4.1). The stream cipher determines which QRT code to use, adding security, integrity, and dispersing data in a single encoding process.

Dispersal could also be achieved using a non-integrated process: use a standard encryption technique followed by an error correction code with similar dispersal properties. However, integrating the process adds important properties which are otherwise missing:

- Diffusion: Input in a QRT block is diffused across the output, with each output piece equivalent to a key dependent hash of the input.
- Combined algorithms deter many practical cryptanalysis techniques: Attacks against modern cryptography requires cipher text and some information about the matching plain text. Non-integrated systems reveal non-encoded cipher text, allowing the cryptography to be analyzed on its own. Integrating code and cryptography minimizes leakage of straight cipher text, preventing cryptanalysis without first recovering the full code.
- Lack of flexibility: Most encoding techniques have size requirements. Quotient Ring Transform based coding has a wide range of block and word sizes.

The remainder of this paper describes the Quotient Ring Transform, and how it can be used to encode, decode, and correct errors in coded data. There are a few things about the QRT that should be kept in mind:

[1]Examples of stream ciphers are AES in counter mode or one of the eStream ciphers[4].

1. The QRT is an extension of the iterative da yen (Chinese remainder theorem)[2].
2. The QRT enables a very flexible linear error correction code.
3. Encoding with the QRT is not Reed-Solomon encoding or secret sharing. However, restricting the QRT to quotient rings of the form $\mathbb{F}[x]/(x-c)$, where $c \in \mathbb{F}$, reduces it to a non-cyclic, non-BCH code similar to Reed-Solomon codes. It converts to I Restricting data further to the the ring $\mathbb{F}[x]/(x-0)$, using random input for the remaining the quotient rings, $\mathbb{F}[x]/(x-j)$ for $1 \leq j < k$, and using the QRT to find the values for $\mathbb{F}[x]/(x-j)$ for $k \leq j \leq n$ reduces the QRT to a k out of n secret sharing variant[3].

Algorithm CookBook in Appendix
This paper describes algorithms required to implement
dispersed cryptography, including the QRT.
More detailed, cookbook versions of the algorithms are listed in the appendix.

2 Data as Relations

Encrypted or transformed data is represented as a set of equivalence relations in reduced form. In this paper, a relation is a quotient ring element over a Euclidean domain $\mathcal{E}$ such as the integers or a polynomial ring. A relation (μ) consists of an value ($\dot{\mu}$) and modulus or ideal generator ($\widehat{\mu}$), both in $\mathcal{E}$.

See example C.1

Size of data must be carefully controlled for any sort of encoding. Dispersed cryptography requires that the size of a relations value is non-negative and bounded by the modulus. For the integers this size boundary implies that values are non-negative and less than the modulus (generator of the ideal). For polynomial rings it implies that the degree of the polynomial is non-negative and less than the degree of the polynomial modulus.

These size restricted values paired with their moduli will be called *q-relations*.

[2]Ta-yen/Da-yan, translates to 'Great Extension', the name given by Qin Jiushao (Shushu Jiuzhang - Mathematical Treatise in Nine Sections in 1247) to the indeterminate analysis solution, also known as the Chinese remainder theorem.

[3]Equivalently, data is the constant of a polynomial with all other coefficients random.

Definition 2.1 (q-Relation)**:** Let $\mathcal{E}$ be a Euclidean domain with function (norm) $\mathcal{N}:\mathcal{E}\rightarrow\mathbb{N}$. A q-relation, μ, is a pair of elements in $\mathcal{E}$ with $\mu=(\dot{\mu},\widehat{\mu})$ representing $(\dot{\mu} \bmod \widehat{\mu})$ with

1. $\widehat{\mu}>0$ and a non-unit;
2. $\dot{\mu}\geq 0$ and $\dot{\mu}=0$ or $\mathcal{N}(\dot{\mu})<\mathcal{N}(\widehat{\mu})$.

Defining q-relations as a pair of elements simplifies the description but neglects a few important details. q-Relations will be compared to other q-relations or to other ring elements. Relative primality between q-relations will be discussed as will operations between q-relations. The following definitions are given to clarify what these comparisons and statements imply:

Definition List 2.2: (q-Relationships)

Let μ_j,μ_k be two q-relations.

2.2.1: μ_j,μ_k are *relatively prime* if their respective moduli, $\widehat{\mu_j},\widehat{\mu_k}$, are relatively prime. A set of relations R is *relatively prime* if all of its elements are pair-wise relatively prime.

2.2.2: μ_j,μ_k are *equivalent* $(\mu_j\equiv\mu_k)$ if respective moduli are the same size $(\mathcal{N}(\widehat{\mu_j})=\mathcal{N}(\widehat{\mu_k}))$ and $\dot{\mu_j}=\dot{\mu_k}$.

2.2.3: A set of relations R is *regular* if every relation in R has the same size: $\mathcal{N}(\mu_i)=\mathcal{N}(\mu_j)$ for all $\mu_i,\mu_j\in R$.

2.2.4: For all $x\in\mathcal{E}$, $x\equiv\mu$ implies $x\equiv\dot{\mu} \bmod \widehat{\mu}$

2.2.5: For all $x\in\mathcal{E}$, $\mu+x=x+\mu$ represents the q-relation $(x+\dot{\mu} \bmod \widehat{\mu})$.

3 The Da Yen

In simple terms, the da yen equates a set of relations to a larger relation (figure 4). It states that the direct product of the set of relations is isomorphic to the larger relation. Errors in an overdetermined set of relations can be detected and corrected using the independence of the individual relations. Both Reed-Solmon and the QRT use the da yen, but the QRT uses it in a slightly different way. The transform converts a set of relations to another equivalent (definition 2.2.2) set of relations without computing the larger intermediate relation (figure 5).

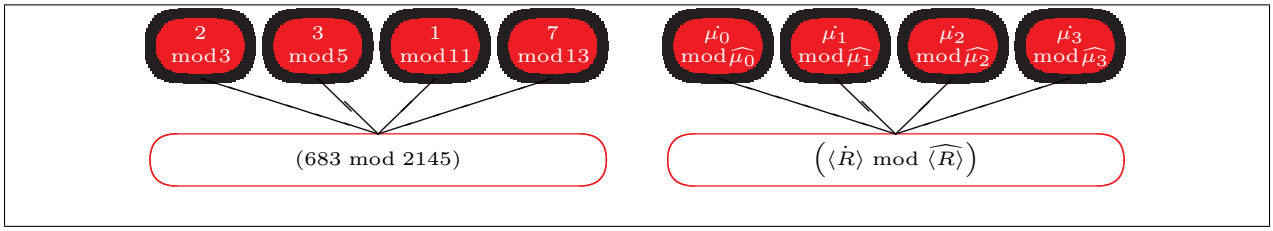


Figure 4: The da yen modulo $3,\ 5, 11,\ 13$

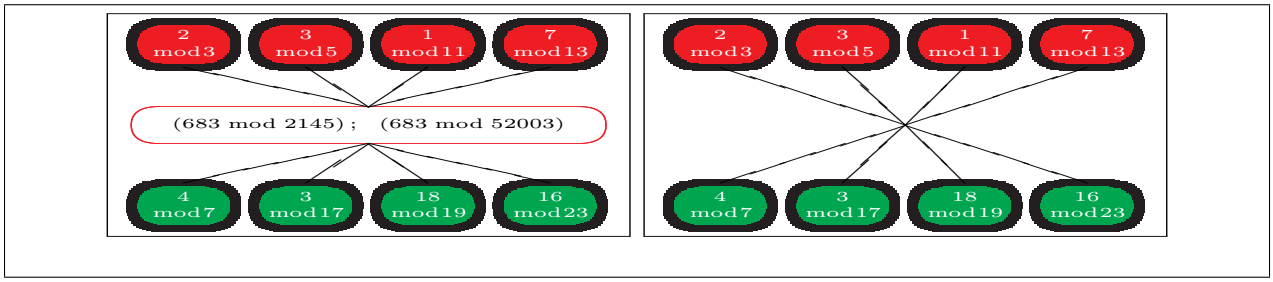


Figure 5: Transforming one relation set to another

Notation: If $R = \{\mu\}$ is a set of co-prime q-relations, then $\langle R \rangle = \left(\langle \dot{R} \rangle, \widehat{\langle R \rangle}\right)$ represents the q-relation, value $\langle \dot{R} \rangle$ and modulus $\widehat{\langle R \rangle}$, with:

$$\widehat{\langle R \rangle} = \prod_{\mu \in R} \widehat{\mu} \qquad\qquad \langle \dot{R} \rangle \equiv \mu \ \forall \ \mu \in R.$$

$$\mathcal{N}(R) = \mathcal{N}\left(\widehat{\langle R \rangle}\right)$$

The existence of $\langle R \rangle$ is guaranteed by the da yen.

Computationally, working in R (independent set of smaller moduli) is much more efficient than in $\langle R \rangle$ (modulo the product of all the small moduli). The QRT uses an intermediate transform called a weave (section 5) which converts $R = \{\mu_j \mid 0 \leq j < n\}$ into a modified set of q-relations $\overline{R} = \{\omega_j \mid 0 \leq j < n\}$ with matching moduli $\widehat{\mu_j} = \widehat{\omega_j}$. The woven set $\overline{R}$ enables computation of $\langle \dot{R} \rangle$ reduced modulo any other modulus using only the smaller modular operations. The full value $\langle \dot{R} \rangle$ is never computed. The woven set also streamlines the detection and correction of errors.

The general ring version of the da yen can be found in [1][3]. Here is the da yen using q-relations.

Theorem 3.1 (Da Yen Isomorphism)**:** Let R be a set of relatively prime q-relations. Then there exists an isomorphism from R to a q-relation $\langle R \rangle$ such that

1. $\widehat{\langle R\rangle} = \prod_{\mu\in R} \widehat{\mu}$ and $\langle \dot{R}\rangle \equiv \mu$ for all $\mu \in R$.

2. (homomorphic) For any two q-relation sets $R = \{\mu_j \mid 0 \le j < n\}, Q = \{\nu_k \mid 0 \le k < n\}$ with equal moduli (i.e., $\widehat{\mu_j} = \widehat{\nu_j}$ for $0 \le j < n$),

$$\langle R\rangle + \langle Q\rangle = \langle \{\dot{\mu_j} + \dot{\nu_j} \bmod \widehat{\mu_j} \mid 0 \le j < n\}\rangle$$
$$\langle R\rangle \langle Q\rangle = \langle \{\dot{\mu_j}\dot{\nu_j} \bmod \widehat{\mu_j} \mid 0 \le j < n\}\rangle$$

See example C.2

4 Iterative Da Yen Algorithm

The weave starts with the iterative version of the da yen (see [2]). This technique maps a set of q-relations R to the q-relation $\langle R\rangle$ by adding one q-relation at a time. Let $R_t = \{\mu_j \mid 0 \le j < t\}$.

1. Let $\widehat{\omega_i} = \widehat{\mu_i}$ for $0 \le i < n$ and $\dot{\omega_0} = \dot{\mu_0}$. $R_1 = \{\mu_0\}$, and $\langle \dot{R_1}\rangle = \dot{\mu_0}$.

2. For $t = 1$ to $n-1$, let

$$\dot{\omega_t} = \left(\widehat{\langle R_t\rangle}^{-1}\left(\dot{\mu_t} - \langle \dot{R_t}\rangle\right) \bmod \widehat{\mu_t}\right) \tag{4.1}$$
$$\langle \dot{R_{t+1}}\rangle = \langle \dot{R_t}\rangle + \widehat{\langle R_t\rangle}\dot{\omega_t}. \tag{4.2}$$

3. $R_n = R$, and $\langle R\rangle = \left(\langle \dot{R_n}\rangle \bmod \widehat{\langle R_n\rangle}\right)$.

Note that $\langle \dot{R_t}\rangle < \widehat{\langle R_t\rangle}$ without any further reduction.

See example C.3

The iterative da yen creates these woven q-relations $\omega_t = (\dot{\omega_t} \bmod \widehat{\mu_t})$ (equation 4.1). Notice that $\langle \dot{R}\rangle$ can be computed with just the set of $\{\omega_j\}$ q-relations:

$$\langle \dot{R}\rangle = \dot{\omega_0} + \widehat{\omega_0}\left(\dot{\omega_1} + \widehat{\omega_1}\left(\ldots\left(\dot{\omega_{n-2}} + \widehat{\omega_{n-2}}\dot{\omega_{n-1}}\right)\ldots\right)\right). \tag{4.3}$$

The weave transform (section 5) computes these woven q-relations without computing the large intermediate q-relations $\langle R_t\rangle$. Once the ω_t have been computed, the full (or any intermediate $\langle \dot{R_t}\rangle$) q-relation value can be computed outright, with the result already in reduced form (equation 4.3). This avoids all large quotient ring computations and parallelizes much of the computation.

Definition 4.1 (Woven Relations and the Quotient Ring Transform (QRT))**:** Let

$$R = \{\mu_j \mid 0 \le j < n\} \qquad\qquad Q = \{\alpha_j \mid 0 \le j < m\}$$

be two sets of q-relations with $\gcd(\widehat{\mu_j}, \widehat{\mu_i}) = 1$ and $\gcd(\widehat{\alpha_j}, \widehat{\alpha_i}) = 1$ for all $i \neq j$.

- The Woven Set of R is: $\overline{R} = \{\omega_t = (\dot{\omega_t} \bmod \widehat{\mu_t}) \mid 0 \le t < n\}$ with $\dot{\omega_t}$ from equation (4.1)
- R evaluated at a relation α is

$$R(\alpha) = \dot{\omega_0} + \widehat{\omega_0}\left(\dot{\omega_1} + \widehat{\omega_1}\left(\ldots\left(\dot{\omega_{n-2}} + \widehat{\omega_{n-2}}\dot{\omega_{n-1}}\right)\ldots\right)\right) \bmod \widehat{\alpha}. \tag{4.4}$$

  (cookbook algorithm B.2)

- The Quotient Ring Transform (QRT) from R to Q (also called the encoding of R) is:

$$R(Q) = \{R(\alpha) \mid \alpha \in Q\}. \tag{4.5}$$

See example C.4

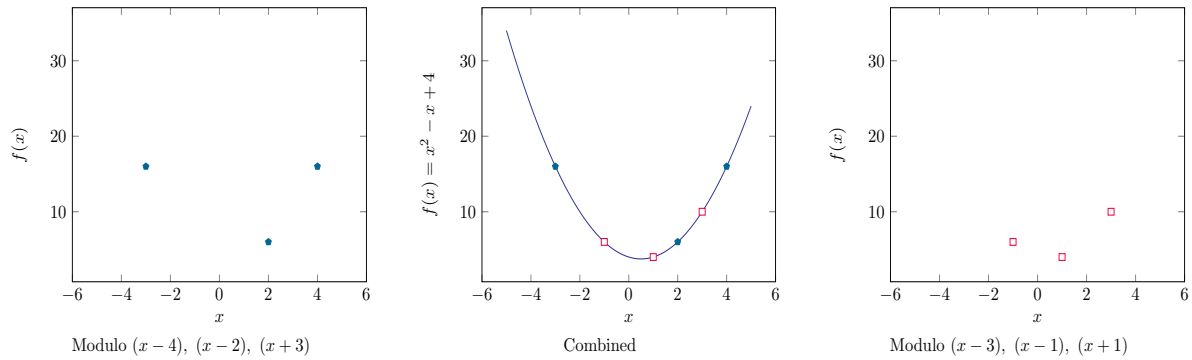
If the size of the relation sets and their values are equal, i.e., $\mathcal{N}(R) = \mathcal{N}(Q)$ and $\langle \dot{R} \rangle = \langle \dot{Q} \rangle$, then R and Q contain the same information even though the individual relations in R and Q appear quite different. If $\mathcal{N}(Q) > \mathcal{N}(R)$, errors in Q can be detected. If $\mathcal{N}(Q)$ is sufficiently larger than $\mathcal{N}(R)$, errors can also be corrected (section 8).

4.1 Integers vs Polynomials

Note that distinct ($\widehat{\mu_j} \neq \widehat{\mu_k}$) yet equivalent q-relations are not possible in $\mathcal{E} = \mathbb{Z}$. For this reason Dispersed Cryptography is not practical over $\mathbb{Z}$. The QRT over the integers is still interesting and may have applications for large integer operations.

The choice of the polynomial domain type of moduli used effect the efficiency of the QRT and dispersed cryptography.

- The most efficient and practical Euclidian domains for dispersed cryptography are polynomial rings $\mathbb{F}_q[x]$ where $q = 2^n$.
- Degree one polynomials in $\mathbb{F}[x]$ (i.e., $\widehat{\mu} = (x - c)$ where $c \in \mathbb{F}$) simplify the QRT even further.
- Note that using degree one polynomials as moduli reduces the QRT to converting a set of points on a polynomial to another set of points on the same polynomial (figure 6)



$(\dot{\mu}, \widehat{\mu})$	$\dot{\mu} \bmod \widehat{\mu}$	(x, y)
$(16, (x-4))$	$16 \bmod (x-4)$	$(4, 16)$
$(6, (x-2))$	$6 \bmod (x-2)$	$(2, 6)$
$(16, (x+3))$	$16 \bmod (x+3)$	$(-3, 16)$
$x^2 - x + 4 \bmod (x^3 - 3x^2 - 10x + 24)$		

$(\dot{\mu}, \widehat{\mu})$	$\dot{\mu} \bmod \widehat{\mu}$	(x, y)
$(10, (x-3))$	$10 \bmod x$	$(3, 10)$
$(4, (x-1))$	$4 \bmod (x-1)$	$(1, 4)$
$(6, (x+1))$	$6 \bmod (x+1)$	$(-1, 6)$
$x^2 - x + 4 \bmod (x^3 - 3x^2 - x + 3)$		

Figure 6: Relations as Points

5 QRT Weave

The QRT weave converts a co-prime set of q-relations, $R = \{\mu_j \mid 0 \leq j < n\}$, to their iterative components. Computation of $\overline{R} = \{\omega_j \mid 0 \leq j < n\}$ requires $(n-1)$ parallel steps (theorem 5.1) or $\binom{n}{2}$ in serial.

Lemma 5.1 (Weave Terms)**:** Let $R = \{\mu_k \mid 0 \leq k < n\}$ be a set of n relatively prime q-relations. If

$$\begin{aligned} \omega_k^{(0)} &= \mu_k && k = 0, 1, \ldots, (n-1) \\ \omega_k^{(j)} &= \left(\widehat{\mu_{j-1}}^{-1} \left(\omega_k^{(j-1)} - \dot{\omega_{j-1}} \right) \bmod \widehat{\mu_k} \right) && 1 \leq j \leq k < n \\ \omega_k &= \omega_k^{(k)}. \end{aligned}$$

then ω_k are the weaving q-relations needed in the iterative da yen (equation 4.1).

Proof. The following will prove that $\omega_k^{(j)} = \left(\widehat{\langle R_j \rangle}^{-1} \left(\mu_k - \langle \dot{R}_j \rangle \right) \bmod \widehat{\mu_k} \right)$ for all $0 \leq j \leq k < n$, and $\omega_k = \omega_k^{(k)}$. By definition, $\omega_k^{(0)} = \mu_k$ and $\omega_0 = \omega_0^{(0)} = \mu_0$, and $R_1 = \{\mu_0\}$. Using induction on t

- For $t = 1$: $\langle \dot{R_1} \rangle = \dot{\mu_0} = \omega_0 = \omega_0^{(0)}$, therefore

$$\omega_k^{(1)} = \widehat{\mu_0}^{-1} \left(\omega_k^{(0)} - \omega_0 \right) \equiv \widehat{\langle R_1 \rangle}^{-1} \left(\dot{\mu_k} - \dot{\mu_0} \right) \equiv \widehat{\langle R_1 \rangle}^{-1} \left(\dot{\mu_k} - \langle \dot{R_1} \rangle \right) \bmod \widehat{\mu_k}$$

  for all $1 \leq k < n$. Furthermore, $\omega_1^{(1)} \equiv \widehat{\langle R_1 \rangle}^{-1} \left(\dot{\mu_1} - \langle \dot{R_1} \rangle \right) \equiv \omega_1 \bmod \widehat{\mu_1}$.

- For $t > 1$, assume

$$\omega_k^{(t-1)} = \widehat{\langle R_{(t-1)} \rangle}^{-1} \left(\dot{\mu_k} - \langle \dot{R_{t-1}} \rangle \right) \bmod \widehat{\mu_k}$$
$$\omega_{t-1} = \omega_j^{(t-1)}$$

  for all $j \leq k < n$.

- Then $\omega_k^{(t)}$ for $t \leq k < n$ is:

$$\begin{aligned} \omega_k^{(t)} &\equiv \widehat{\mu_{t-1}}^{-1} \left(\omega_k^{(t-1)} - \omega_{t-1} \right) \bmod \widehat{\mu_k} \\ &\equiv \widehat{\mu_{t-1}}^{-1} \left(\widehat{\langle R_{t-1} \rangle}^{-1} \left(\dot{\mu_k} - \langle \dot{R_{t-1}} \rangle \right) - \omega_{t-1} \right) \bmod \widehat{\mu_k} \\ &\equiv \widehat{\langle R_t \rangle}^{-1} \left(\dot{\mu_k} - \left(\langle \dot{R_{t-1}} \rangle + \widehat{\langle R_{t-1} \rangle} \omega_{t-1} \right) \right) \bmod \widehat{\mu_k} \end{aligned}$$

  We know (equation 4.2) that $\langle \dot{R_t} \rangle = \langle \dot{R_{t-1}} \rangle + \widehat{\langle R_{t-1} \rangle} \omega_{t-1}$, therefore

$$\omega_k^{(t)} \equiv \widehat{\langle R_t \rangle}^{-1} \left(\dot{\mu_k} - \langle \dot{R_t} \rangle \right) \bmod \widehat{\mu_k}$$

  and $\omega_t = \omega_t^{(t)} \equiv \widehat{\langle R_t \rangle}^{-1} \left(\dot{\mu_t} - \langle \dot{R_t} \rangle \right) \bmod \widehat{\mu_t}$

□

(cookbook algorithm B.1)

See example C.5

6 Weave-Swap

If $R = \{\mu_j \mid 0 \leq j < n\}$, the weave transform $\overline{R}$ lets us extract not only $\langle R \rangle$, but the relation $\langle R_t \rangle$, where $R_t = \{\mu_0, \mu_1, \ldots, \mu_{t-1}\}$, for every $0 < t \leq n$. It does not let us extract the relations for other subsets, such as $\{\mu_1, \mu_3, \mu_5\}$.

The order of q-relations in R does not matter for $\langle R \rangle$, but it matters in the woven set $\overline{R}$. Lower indexed woven relations are not dependent on higher indexed relations. More precisely, the j-th weave relation ω_j (equation 4.1) depends only on R_j and μ_j. In other words, only on the unordered set of relations μ_k with $0 \leq k \leq j$.

Errors are detected and corrected in an overdetermined set of q-relations by extracting data from different minimal subsets, and comparing computed values to the remaining q-relations. Re-computing the weave for each subset would be computationally prohibitive. Fortunately, the order of the transform can be easily permuted with a 'weave swap'.

Notation (Relation Weave and Permutations): Let π be a permutation on the integers $0 \leq j < n$: $\pi(j) = j$ for $0 \leq j < n$. Then for $R = \{\mu_j \mid 0 \leq j < n\}$, any $0 < t \leq n$,

$$\overline{R_t^{\pi}} = \left\{\omega_{\pi(j)} \mid 0 \leq j < t\right\}$$

is the woven q-relation set ordered by π. q-Relation ω_j has modulus $\widehat{\omega_j} = \widehat{\mu_j}$ and values $\dot{\omega_j}$ such that:

$$\langle \dot{R_t^{\pi}} \rangle = \dot{\omega_{\pi(0)}} + \widehat{\omega_{\pi(0)}} \left(\dot{\omega_{\pi(1)}} + \widehat{\omega_{\pi(1)}} \left(\ldots \left(\dot{\omega_{\pi(t-2)}} + \widehat{\omega_{\pi(t-2)}} \dot{\omega_{\pi(t-1)}}\right) \ldots\right)\right) \tag{6.1}$$

Weave swapping switches the order of two adjacent q-relations in a woven transform with minimal work. Only one multiply and one 'divide' are required. If $\mathcal{E} = \mathbb{F}_{2^v}$ and moduli are all degree one, only the single multiply is required (equation 6.5, 6.4).

Recall that the weave transform (theorem 5.1) has intermediate results

$$\omega_{\pi(j)}^{(k)} = \left(\dot{\omega_{\pi(j)}} - \dot{\omega_{\pi(k)}}\right) \widehat{\omega_{\pi(k)}}^{-1} \bmod \widehat{\mu_{\pi(j)}}$$

with

$$\omega_{\pi(j)} = \omega_{\pi(j)}^{(j)}.$$

If the initial permutation is π and the $(j, (j+1))$ terms need to be swapped, with σ as the resulting permutation, then:

1. $\dot{\omega_{\sigma(j)}} = \dot{\omega_{\pi(j+1)}^{(j)}}$, therefore just the (j)-th step of the weave must be undone:

$$\dot{\omega_{\sigma(j)}} = \dot{\omega_{\pi(j+1)}} \widehat{\omega_{\pi(j)}} + \dot{\omega_{\pi(j)}} \bmod \widehat{\omega_{\sigma(j)}}. \tag{6.2}$$

2. $\omega_{\sigma(\dot{j+1})}{}^{(j)} = \omega_{\pi(\dot{j})}$ and we need $\omega_{\sigma(\dot{j+1})} = \omega_{\sigma(\dot{j+1})}{}^{(j+1)}$, so to move it one term higher:

$$\omega_{\sigma(\dot{j+1})} = \left(\omega_{\pi(\dot{j})} - \omega_{\sigma(\dot{j})}\right) \widehat{\omega_{\sigma(j)}}^{-1} \bmod \widehat{\omega_{\sigma(j+1)}}. \tag{6.3}$$

The second half of the swap above (equation 6.3) disappears completely if moduli are degree one monic. Let $\mathcal{E} = \mathbb{F}[x]$ for some field $\mathbb{F}$ and moduli be $\widehat{\mu_j} = (x - c_j)$, with $c_j \in \mathbb{F}$. Then:

$$\widehat{\mu_j} \equiv (c_k - c_j) \bmod \widehat{\mu_k}.$$

Replacing the moduli with the sum of the constants also eliminates any need for reduction – every term in the equation is now an element of $\mathbb{F}$.

$$\omega_{\sigma(\dot{j})} = \omega_{\pi(\dot{j+1})} \left(c_{\pi(j+1)} - c_{\pi(j)}\right) + \omega_{\pi(\dot{j})} \tag{6.4}$$

$$\begin{aligned}
\omega_{\sigma(\dot{j+1})} &= \left(\omega_{\pi(\dot{j})} - \omega_{\sigma(\dot{j})}\right) \left(c_{\pi(j)} - c_{\pi(j+1)}\right)^{-1} \\
&= \left(\omega_{\pi(\dot{j})} - \left(\omega_{\pi(\dot{j+1})} \left(c_{\pi(j+1)} - c_{\pi(j)}\right) + \omega_{\pi(\dot{j})}\right)\right) \left(c_{\pi(j)} - c_{\pi(j+1)}\right)^{-1} \\
&= \left(\omega_{\pi(\dot{j+1})} \left(c_{\pi(j)} - c_{\pi(j+1)}\right)\right) \left(c_{\pi(j)} - c_{\pi(j+1)}\right)^{-1} \\
&= \omega_{\pi(\dot{j+1})}
\end{aligned} \tag{6.5}$$

Algorithms detailing the general case (cookbook algorithm B.3) and the special case (cookbook algorithm B.4): $\mathcal{E} = \mathbb{F}[x]$ using degree one, monic polynomials as moduli are in the appendix.

7 Encoding

The QRT allows for error correction by transforming R into a larger set Q. Data is stored as values, $\dot{\mu_j}$, in a set of relatively prime regular q-relations (definition 2.2), R, over a polynomial ring $\mathbb{F}[x]$. If there are $2r$ data q-relations, then transforming that to Q, a distinct set of $2(r + s)$ q-relations, allows for s errors to be detected and corrected (theorem A.1). Moduli of the q-relation sets $(R \bigcup Q)$ may be chosen at random or be fixed, as long as the relations are regular, pair-wise relatively prime (definition 2.2.3, 2.2.1).

Salt is random data used in some cryptographic systems to insure input is sufficiently random. Dispersed cryptography adds salt q-relations to R to improve encryption on repeated data, such as all zeros. The salt q-relations, both value and moduli, will be generated by the cryptographic stream cipher and will be known to both encoder and decoder. Salt helps mask constant data and is used along with the data to create the coded q-relations.

Here is a short description of the encoding process. The full encoding algorithm is in the appendix (cookbook algorithm B.5).

- Let the data and salt q-relation set be: $R = \{\mu_j \mid 0 \le j < 2r + l\}$. Data will be the first $2r$ relations, salt the remaining l relations.
- Encoded data will be $Q = \{\nu_j \mid 0 \le j < 2(r + s)\}$ with $\dot{\nu_j} \equiv \langle \dot{R} \rangle \bmod \widehat{\nu_j}$.

Note that the amount of salt added has no effect on the number of encoded relations. It also has minimal effect on the error correction process.

8 Error Correction

The decoding process is identical to encoding if there are no errors and at least $2r$ q-relations were received. If $2(r+t)$ q-relations were received with at most t errors, an error free set of $2r$ q-relations can be found and error free plain text extracted.

The decoded values are determined by $2r$ data and l salt q-relations. Let the number of received q-relations be $2(r + t)$: $Q = \{\nu_j \mid 0 \le j < 2(r + t)\}$. We know that if Q has at most t errors, then there exist subsets of Q of size $2r$ which are error free. Furthermore, if a subset $Q' \subset Q$ contains errors, more than t of the q-relations in $Q \setminus Q'$ will contain errors (theorem A.1). This gives us an easy way to check if a given subset is error free.

1. Weave salt $\{\mu_j \mid 2r \le j < 2r + l\}$ and $Q = \{\nu_j \mid 0 \le j < 2(r + t)\}$ to form

   $$\overline{Q} = \{\omega_j \mid 0 \le j < l + 2(r + t)\}.$$

   Salt is error free, so should be fixed as the lower l terms: $\widehat{\mu_{2r+j}} = \widehat{\omega_j}$ for $0 \le j < l$.

2. Check that $Q_{2r+l}(\nu_j) = \nu_j$ for all $l + 2r \le j < l + 2(r + t)$.

3. If all the upper $2t$ q-relations check (i.e., equal their computed values), then there are no errors;

4. If more than t q-relations fail, then there is an error in the set $\{\omega_j \mid l \le j < l + 2r\}$.

5. Otherwise (less than or equal to t failed q-relations) the errors are in the upper q-relations which fail the check.

Brute force decoding requires at most $\binom{2r+2t}{2r}$ tests and associated weave swaps (section 6). This number is reduced to at most $\binom{r+t}{r}$ tests with a simple pairing of relations. q-Relations are paired with their adjacent q-relation: pair j is $(\nu_{l+2j}, \nu_{l+2j+1})$. Of the $(r + t)$ q-relation pairs, r

are needed to extract data for the test, and t are used for the check. In the worst case recoverable scenario – t errors – there are at most t pairs which could contain errors and at least one set of r pairs which is error free. In other words, one of these $(r + t)$ choose r pair subsets is an error free transformation. This reduces the number of subsets to $\binom{r+t}{r}$. Pairs of q-relations will always be swapped together (cookbook algorithm B.7).

The decoding algorithm (cookbook algorithm B.8) intelligently exhausts over subsets of received q-relations by pairing adjacent q-relations. This reduces computation time per subset by replacing the full transform with a few pair q-relation pair swaps. The full transform only once, at the beginning of the decoding process. If a set of $2r + l$ q-relations contains errors, the woven data is modified to a new subset by doing pair swaps (cookbook algorithm B.7). Notice that the addition of salt equations effects the initial weave, the subset checks, and final extraction, but does not increase the number of swaps required.

References

[1] Thomas W. Hungerford, *Algebra*, ch. III, pp. 131–133, Springer-Verlag, 175 Fifth Avenue, New York, New York 10010, U.S.A., 1974.

[2] D.E. Knuth, *The art of computer programming: Fundamental algorithms*, 3 ed., vol. 1, ch. 4.3.2, pp. 289–290, Addison-Wesley Publishing Company, Reading, Massachusetts, 1997.

[3] Serge Lang, *Algebra*, third ed., ch. [II,§2], pp. 63–64, Addison-Wesley Publishing Co., Reading, Massachusettes, 1971.

[4] Various, *estream: the ecrypt stream cipher project*, Website: EU ECRYPT Network, 2008, http://www.ecrypt.eu.org/stream/.

A Proofs for Woven Transform and Encoding

A.1 Proof of Error Correction Capabilities

Forcing the number of words used in the encoding to be even reduces computational cost, so in this description there will be $2r$ q-relations for plain text and $2s$ correction q-relations: $R = \{\mu_j \mid 0 \leq j < 2(r + s)\}$. Any subset $I \subset R$ with $|I| \geq 2r$ determines all $2(r + s)$ q-relations in R. In other words, $I(\mu) = \mu$ for all $\mu \in R$.

Let the corrupted set be $R' = \{\mu' = \mu_j + e_j \mid 0 \leq j < 2(r+s)\}$ with at most s of the $e_j \in \mathcal{E}$ not zero, and $I \subset R$ with $|I| = 2r$. The following proof will show that if there are at most s errors in R' and $|I(R) \setminus R'| \leq s$ (i.e., $I(R)$ and R' are equal in all but at most s q-relations), then I contains no errors and $R = I(R)$.

The following set definitions will solidify the concepts and simplify the proof. The sets G, B contain the actual correct (good) and corrupted (bad) q-relations, while the sets G_I, B_I contain the observed correct and corrupted q-relations derived with a subset $I \in R'$.

$$G = R' \cap R = \{\mu'_j \in R' \mid e_j \equiv 0\} \qquad B = R' \setminus R = \{\mu'_j \in R' \mid e_j \not\equiv 0\} \tag{A.1}$$

$$G_I = R' \cap I(R') = \{\mu' \in R' \mid I(\mu') = \mu'\} \qquad B_I = R' \setminus I(R') = \{\mu' \in R' \mid I(\mu') \neq \mu'\} \tag{A.2}$$

Notice that:

- Both pairs, G, B and G_I, B_I partition the set of observed q-relations R':

$$R' = G \cup B \qquad \emptyset = G \cap B$$
$$R' = G_I \cup B_I \qquad \emptyset = G_I \cap B_I$$

- The sets G, B depend only on R and R'.
- The sets G_I, B_I depend only on the results of $\langle I \rangle$, therefore if I, J are equivalent, $\langle I \rangle \equiv \langle J \rangle$ and $G_I = G_J$ and $B_I = B_J$.

This encoding scheme is essentially an application of the da yen[4][1], as is Reed-Solomon coding. The proofs rely on this theorem.

Theorem A.1 (Error Correction Capabilities)**:** Let $R = \{\mu_j \mid 0 \leq j < 2(r+s)\}$ be a set of relatively prime q-relations uniquely determined by any subset of $2r$ elements. Let the set with added errors be

$$R' = \{\mu'_j = \mu_j + e_j \mid e_j \in \mathcal{E},\ 0 \leq j < 2(r+s)\}$$

with at most s non-zero e_j. If $I \subset R'$ with $|I| = 2r$, then $|B_I| \leq s$ if and only if $I \subset G$.

Proof. Let R, R' be defined as above with $I \subset R'$ be a subset with $2r$ elements and $|B| \leq s$. Because any $2r$ q-relations in R uniquely determine R and $G \subset R$, and subset $I \subset G$ with $|I| = 2r$ generates R: $I(R) = I(R') = R$. Any subset $J \subset I(R)$ with $|J| \geq |I|$ contains , and $G_I \subset I(R)$, therefore any subset $J \subset G_I$ is equivalent to I:

[4]aka: Chinese Remainder Theorem

1. If $I \subset G$, then $I(\mu) = \mu$ for all $\mu \in R$ and $|B_I| = |B| \leq s$.
2. If $|B_I| \leq s$:
   - $|G_I| = |R'| - |B_I| \geq 2(r+s) - s = 2r + s$.
   - Since $|G_I \cap B| \leq s$, we know $|G_I \cap G| = |G_I| - |G_I \cap B| \geq 2r + s - s = 2r$.
   - Therefore there exists $J \subset (G_I \cap G) \subset G$ with $|J| = 2r$ and $J \equiv I$.
   - Finally, since $J \subset G$, $|J| = 2r$ we know that $J(R') = R$. Since $J \equiv I$, $I(R') = R$ therefore $I \subset G$.

□

B Algorithms

B.1 Basic QRT Weave (section 5)

Algorithm B.1: Weave Transform (section 5)

Input: $R = \{\mu_j \mid 0 \leq j < n\}$, a set of relatively prime q-relations

Output: $\overline{R} = \{\omega_j \mid 0 \leq j < n\}$ an ordered set of relatively prime q-relations satisfying equation (4.1).

I: Initialize:

A: $\omega_0 = \mu_0$

B: for $k = 1$ to $(n-1)$: $\omega_k^{(0)} = \mu_k$;

II: For $j = 1$ to $(n-1)$:

A: $\omega_j = \omega_j^{(j)}$

B: For $k = (j+1)$ to $(n-1)$: $\omega_k^{(j)} = \left(\widehat{\omega_j}^{-1}\left(\omega_k^{(\dot{j-1})} - \dot{\omega_j}\right) \bmod \widehat{\omega_k}\right)$

End of Algorithm B.1

B.2 Reduction with Woven Values (equation 4.4)

Algorithm B.2: Simplified Reduction (section 4)

Input: $\overline{R} = \{\omega_j \mid 0 \leq j < n\}$ and q-relation α

Output: q-relation $R(\alpha) = \left(\langle \dot{R} \rangle \bmod \widehat{\alpha}\right)$

I: Set $v = \dot{\omega_{n-1}} \bmod \widehat{\alpha}$

II: for $j = (n-2)$ down to 0: $v = v\widehat{\omega_j} + \dot{\omega}_j \bmod \widehat{\alpha}$

III: return $R(\alpha) = (v \bmod \widehat{\alpha})$

End of Algorithm B.2

B.3 Weave Swapping (section 6)

Algorithm B.3: Weave Swap, General (section 6)

Input: $\overline{R^\pi}$, $0 \leq j < (n-1)$

Output: $\overline{R^\sigma}$ (equation 6.2, 6.3)

I: $\omega_{\dot{\pi(j+1)}} = \omega_{\dot{\pi(j+1)}}\widehat{\omega_{\pi(j+1)}} + \dot{\omega_{\pi(j)}} \bmod \widehat{\omega_{\pi(j+1)}}$

II: $\dot{\omega_{\pi(j)}} = \left(\dot{\omega_{\pi(j)}} - \omega_{\dot{\pi(j+1)}}\right)\widehat{\omega_{\pi(j+1)}}^{-1} \bmod \widehat{\omega_{\pi(j)}}$

III: $\sigma = \pi$

IV: $\sigma(j) = \pi(j+1)$

V: $\sigma(j+1) = \pi(j)$

VI: $\overline{R^\pi}$ is now $\overline{R^\sigma}$

End of Algorithm B.3

Algorithm B.4: **Weave Swap** $\mathcal{E} = \mathbb{F}[x],\ |\widehat{\mu}| = 1$ **(section 6)**

Input: $\overline{R^{\pi}}$, $0 \le j < (n-1)$ with $\widehat{\omega_j} = (x - c_j)$

Output: $\overline{R^{\sigma}}$ (equation 6.4, 6.5)

I: $tmpVal = \dot{\omega_{\pi(j+1)}}$

II: $\dot{\omega_{\pi(j+1)}} = \dot{\omega_{\pi(j+1)}} \left(c_{\pi(j+1)} - c_{\pi(j)}\right) + \dot{\omega_{\pi(j)}}$

III: $\dot{\omega_{\pi(j)}} = tmpVal$

IV: $\sigma = \pi$

V: $\sigma(j) = \pi(j+1)$

VI: $\sigma(j+1) = \pi(j)$

VII: $\overline{R^{\pi}}$ is now $\overline{R^{\sigma}}$

End of Algorithm B.4

B.4 QRT Encoding (section 7)

Algorithm B.5: Woven Encoding With Salt (section 7)

Input:

$\mathcal{E}$	For efficiency $\mathcal{E} = \mathbb{F}_{2^n}[x]$ though any polynomial ring will work
v	the fixed degree of moduli;
$2r$	the number of data q-relations per block
l	the number of salt q-relations ($l \ge 0$)
$2s$	the number of spare q-relations
$R = \{\mu_j \mid 0 \le j < 2r + l\}$	μ_j are q-relations over $\mathcal{E}$ containing data ($0 \le j < 2r$) and salt ($2r \le j < 2r + l$).
$Q = \{\nu_j \mid 0 \le j < 2(r+s)\}$	output q-relation moduli (i.e., $\dot{\nu_j}$ is not set).

Output: $\{\dot{\nu_j} \mid 0 \leq j < 2(r+s)\}$, encoded words in $\mathcal{E}$

I: Return the set of elements: $R(\dot{Q})$ (definition 4.1) using algorithm B.2

End of Algorithm B.5

B.5 QRT Decoding (section 8)

Algorithm B.6: Check Decoding Weave (section 8)

Input:

$Q = \{\mu_j \mid 0 \leq j < d+2t\}$	A set of q-relations
$\overline{Q^\sigma} = \{\omega_j \mid 0 \leq j < d+2t\}$	the transformed version of Q using permutation σ
d	d q-relations determine Q
t	$2t$ check q-relations in Q

Output: true, if weave is correct; false is fails correct check

I: `numPass` $= 0$

II: for $j = 0$ to $2t-1$, and `numPass` $< t$:

A: Extract $Q_d^\sigma(\mu_{\sigma(d+j)}) = \tau$ using $\overline{Q_d^\sigma}$ (cookbook algorithm B.2).

B: If τ equals $\mu_{\sigma(d+j)}$: `numPass` $=$ `numPass` $+ 1$.

III: return: if `numPass` $\geq t$, return true; else return false

End of Algorithm B.6

Algorithm B.7: Pair Swapping (section 8)

Input:

$\overline{D^\sigma} = \{\omega_{\sigma(j)} \mid 0 \leq j < l+2(r+t)\}$	set of $l+2(r+t)$ woven q-relations ordered by σ
j	pair index $0 < j < (r+t)$ to swap with $(j-1)$
l	number of salt q-relations in D

Output: $\overline{D^{\pi}}$ with altered permutation

I: swap (cookbook algorithm B.3, B.4) $\overline{D^{\sigma}}$ at $l+2j$ and $l+(2j-1)$;

II: swap (cookbook algorithm B.3, B.4) $\overline{D^{\sigma}}$ at $l+(2j-1)$ and $l+(2j-2)$;

III: swap (cookbook algorithm B.3, B.4) $\overline{D^{\sigma}}$ at $l+(2j+1)$ and $l+2j$

IV: swap (cookbook algorithm B.3, B.4) $\overline{D^{\sigma}}$ at $l+2j$ and $l+(2j-1)$

End of Algorithm B.7

Algorithm B.8: Decoding (section 8)

Input:

$\mathcal{E}$	For efficiency $\mathcal{E}=\mathbb{F}_{2^n}[x]$ though any polynomial ring will work
v	the fixed degree of moduli; for efficiency $v=1$ (monic degree one polynomials).
r	$2r$ data words per block
t	$2t$ spare words, $t \le s$
l	l known salt words $l \ge 0$
$Q^{\pi}_{2(r+t)} = \{\nu_{\pi(j)} \mid 0 \le j < 2(r+t)\}$	received code words and initial permutation
$S = \{\kappa_j \mid 0 \le j < l\}$	known salt q-relations
$R = \{\mu_j \mid 0 \le j < 2r\}$	set of output q-relations, moduli only

Output: $R = \{\mu_j \mid 0 \le j < 2r\}$ or Error

I: $\mathtt{L} = \mathtt{z} = 0$, $\mathtt{nVet} = 0$, $\mathtt{LVet}[j] = 0 \qquad 0 \le j < t$

II: Set $\delta_j = \kappa_j$ for $0 \le j < l$ and $\delta_j = \nu_{\pi(j-l)}$ for $l \le j < 2(r+t)$.

III: Set $\sigma(j) = j$ and $D = \{\delta_j \mid 0 \le j < 2(r+t)+l\}$

IV: Weave (cookbook algorithm B.1) D^{σ} into $\overline{D^{\sigma}}$.

V: `pass`=check weave (cookbook algorithm B.6) on D^{σ} and $\overline{D^{\sigma}_{2r}}$ with $d = 2r + l$ and t

VI: while (pass is false) and $\mathtt{L} < t$

A: for $j = (r - \mathtt{z})$ to r: pair swap (cookbook algorithm B.7) $\overline{D^{\sigma}}$ at j

B: `pass`=check weave (cookbook algorithm B.6) on D^{σ} and $\overline{D^{\sigma}_{l+2r}}$

C: $++\mathtt{z}$; $++\mathtt{LVet}[0]$; $++\mathtt{nVet}$

D: if (pass is false) and $\mathtt{nVet} \geq r$

1: While (pass is false) and $\mathtt{L} < t$ and $\mathtt{nVet} \geq r$:

i: $\mathtt{nVet} = \mathtt{nVet} - \mathtt{LVet}[\mathtt{L}]$

ii: $\mathtt{LVet}[\mathtt{L}] = 0$

iii: $++\mathtt{L}$

iv: for $j = (r + \mathtt{L})$ down to r: pair swap (cookbook algorithm B.7) $\overline{Q^{\sigma}}$ at j.

v: `pass` =check weave (cookbook algorithm B.6) on Q^{σ} and $\overline{Q^{\sigma}_{l+2r}}$

vi: $++\mathtt{LVet}[\mathtt{L}]$; $++\mathtt{nVet}$

2: if (pass is false) and $\mathtt{nVet} < r$: $\mathtt{z} = 1$; $\mathtt{L} = 0$

VII: if (pass is false) return fail;

VIII: else return $Q^{\sigma}_{l+2r}(R)$.

End of Algorithm B.8

C Examples

Example C.1: q-Relations

Domain $\mathcal{E}$	relation μ	value $\dot{\mu}$	modulus $\widehat{\mu}$	Domain $\mathcal{E}$	relation μ	value $\dot{\mu}$	modulus $\widehat{\mu}$
$\mathbb{Z}$	$(2 \bmod 3)$	2	3	$\mathbb{F}_{11}[x]$	$(5 \bmod (x-2))$	5	$(x-2)$
$\mathbb{Z}$	$(1 \bmod 5)$	1	5	$\mathbb{F}_{11}[x]$	$(7 \bmod (x-1))$	7	$(x-1)$
$\mathbb{Z}$	$(3 \bmod 7)$	3	7	$\mathbb{F}_{11}[x]$	$(2 \bmod (x-4))$	2	$(x-4)$

Domain $\mathcal{E}$	relation μ	value $\dot{\mu}$	modulus $\widehat{\mu}$
$\mathbb{F}_2[x]$	$(x^3+x+1 \bmod x^4+x^3+1)$	x^3+x+1	x^4+x^3+1
$\mathbb{F}_2[x]$	$(x^2+x+1 \bmod x^5+x^2+1)$	x^2+x+1	x^5+x^2+1

End of Example C.1

Example C.2: Da Yen

Continued from (example C.1). Note that polynomials over $\mathbb{F}_2$ will be written in hexadecimal notation. For example $\texttt{0xB} = x^3 + x + 1$.

$\mathcal{E}$	Relation Set R	Combined Relation $\langle R \rangle$
$\mathbb{Z}$	$\{(2,3),(1,5),(3,7)\}$	$(101, 105)$
$\mathbb{Z}$	$\{(1,2),(2,11),(10,13)\}$	$(101, 286)$
$\mathbb{F}_{11}[x]$	$\{(5,(x-2)),\ (7,(x-1)),\ (2,(x-4))\}$	$\left(\left(2x^2+3x+2\right), \left(x^3+4x^2+3x+3\right)\right)$
$\mathbb{F}_{11}[x]$	$\{(1,(x-5)),\ (0,(x-7)),\ (7,(x-3))\}$	$\left(\left(2x^2+3x+2\right), \left(x^3+7x^2+5x+5\right)\right)$
$\mathbb{F}_2[x]$	$\{(\texttt{0xb}, \texttt{0x19}),\ (\texttt{0x7}, \texttt{0x25})\}$	$(\texttt{0x1d4}, \texttt{0x35d})$
$\mathbb{F}_2[x]$	$\{(\texttt{0x4}, \texttt{0xb}),\ (\texttt{0x1d}, \texttt{0x43})\}$	$(\texttt{0x1d4}, \texttt{0x2dd})$

End of Example C.2

Example C.3: Iterative Da Yen

Continued from (example C.2): $\omega_{t-1}^{\cdot}$ in bold and :

1. Over $\mathbb{Z}$: $R = \{(2,3),(1,5),(3,7)\}$

$$\langle R_1 \rangle = \mathbf{2} \bmod 3$$
$$\langle R_2 \rangle = 2 + 3\left(3^{-1}(1-2) \bmod 5\right) = 2 + 3(\mathbf{3}) = 11 \bmod 15$$
$$\langle R_3 \rangle = 11 + 15\left(15^{-1}(3-11) \bmod 7\right) = 11 + 15(\mathbf{6}) = 101 \bmod 105$$
$$\overline{R} = \{2 \bmod 3,\ 3 \bmod 5,\ 6 \bmod 7\}.$$

2. Over $\mathbb{F}_{11}[x]$: $R = \{(5,(x-2))\,,\ (7,(x-1))\,,\ (2,(x-4))\}$

$$\begin{aligned}
\langle R_1\rangle &= \mathbf{5} \bmod (x-2)\\
\langle R_2\rangle &= 5 + (x-2)\left((x-2)^{-1}(7-5) \bmod (x-1)\right) = 5 + (x-2)\,(\mathbf{9})\\
&= 9x + 9 \bmod (x^2 + 8x + 2)\\
\langle R_3\rangle &= 9x + 9 + (x^2+8x+2)\left((x^2+8x+2)^{-1}(2-(9x+9)) \bmod (x-4)\right)\\
&= 9x + 9 + (x^2+8x+2)\left(6^{-1}(1)\right) = 9x + 9 + (x^2+8x+2)(\mathbf{2})\\
&= 2x^2 + 3x + 2 \bmod (x^3 + 4x^2 + 3x + 3)
\end{aligned}$$

$$\overline{R} = \{5 \bmod (x-2),\ 9 \bmod (x-1),\ 2 \bmod (x-4)\}\,.$$

3. Over $\mathbb{F}_2[x]$: $R = \{(\texttt{0xb}, \texttt{0x19})\,,\ (\texttt{0x7}, \texttt{0x25})\}$.

$$\begin{aligned}
\langle R_1\rangle &= \mathbf{0xb} \bmod \texttt{0x19}\\
\langle R_2\rangle &= \texttt{0xb} \oplus \texttt{0x19}\left(\texttt{0x19}^{-1}\,(\texttt{0x7} \oplus \texttt{0xb}) \bmod \texttt{0x25}\right)\\
&= \texttt{0xb} \oplus \texttt{0x19}\,(\texttt{0xa0xc} \bmod \texttt{0x25})\\
&= \texttt{0xb} \oplus \texttt{0x19} \cdot \mathbf{0x17}\\
&= \texttt{0x1d4}
\end{aligned}$$

End of Example C.3

Example C.4: Simplified Reduction, from (example C.3)

1. Over $\mathbb{Z}$: from (example C.3), $R = \{(2,3),(1,5),(3,7)\}$ and the woven set is $\overline{R} = \{(2,3),\ (3,5),\ (6,7)\}$. If $\widehat{\alpha} = 11$, compute $R(\alpha)$:

$$R(\alpha) \equiv 2 + 3(3 + 5(6)) \equiv 2 + 3(3+8) \equiv 2 + 3(0) \equiv 2 \bmod 11$$

2. Over $\mathbb{F}_{11}[x]$: from (example C.3), $R = \{(5,(x-2))\,,\ (7,(x-1))\,,\ (2,(x-4))\}$, and the woven set is $\overline{R} = \{(5,(x-2)),\ (9,(x-1)),\ (2,(x-4))\}$. If $\widehat{\alpha} = (x-3)$, compute $R(\alpha)$:

$$\begin{aligned}
R(\alpha) &\equiv 5 + (x-2)\,(9 + (x-1)2) \bmod (x-3)\\
&\equiv 5 + (3-2)\,(9 + (3-1)2) \equiv 7 \bmod (x-3)
\end{aligned}$$

3. Over $\mathbb{F}_2[x]$: from (example C.3), $R = \{(\texttt{0xb}, \texttt{0x19}),\ (\texttt{0x7}, \texttt{0x25})\}$, and the woven set is $\overline{R} = \{(\texttt{0xb}, \texttt{0x19}), (\texttt{0x17}, \texttt{0x25})\}$. If $\widehat{\alpha} = \texttt{0xb}$, compute $R(\alpha)$:

$$\begin{aligned} R(\alpha) &= \texttt{0xb} \oplus \texttt{0x19} \cdot \texttt{0x17} \bmod \texttt{0xb} \\ &= \texttt{0x0} \oplus \texttt{0x4} \cdot \texttt{0x1} \bmod b \\ &= 4 \end{aligned}$$

End of Example C.4

Example C.5: Integer Weave Transform

Let $R = \big\{(2 \bmod 3)(3 \bmod 5)(1 \bmod 11)(7 \bmod 13)\big\}$, find $\langle \dot{R} \rangle \bmod 7$.

	$\widehat{\mu_0} = 3$	$\widehat{\mu_1} = 5$	$\widehat{\mu_2} = 11$	$\widehat{\mu_3} = 13$
$\omega_k^{(0)}$	2	3	1	7
$\omega_k^{(1)}$	– – –	$3^{-1}(3-2)$ $2(1)$ 2	$3^{-1}(1-2)$ $4(-1)$ 7	$3^{-1}(7-2)$ $9(5)$ 6
$\omega_k^{(2)}$	– – –		$5^{-1}(7-2)$ $9(5)$ 1	$5^{-1}(6-2)$ $8(4)$ 6
	– – –			$11^{-1}(6-1)$ $6(5)$ 4
$\overline{R}\pi$	$(2 \bmod 3)$	$(2 \bmod 5)$	$(1 \bmod 11)$	$(4 \bmod 13)$

Using the transformed set it is easy to compute the value modulo 7:

$$\begin{aligned} \langle \dot{R} \rangle &\equiv 2 + 3(2 + 5(1 + 11(4))) \bmod 7 \\ &\equiv 2 + 3(2 + 5(1 + 4(4))) \bmod 7 \\ &\equiv 2 + 3(2 + 5(3)) \bmod 7 \\ &\equiv 2 + 3(3) \bmod 7 \\ &\equiv 4 \bmod 7 \end{aligned}$$

or the value modulo 2:

$$\langle \dot{R} \rangle \equiv 2 + 3(2 + 5(1 + 11(4))) \bmod 2 \equiv 0 + 1(0 + 1(1 + 0)) \equiv 1 \bmod 2$$

Notice that the full value of $\langle R\rangle$ would be $\langle R\rangle = (2+3(2+5(1+11(4)))) = 683 \bmod 2145$, which is equivalent to $\langle \dot{R}\rangle \equiv 4 \bmod 7$.

End of Example C.5

Example C.6: Weave Coding/Decoding– $\mathcal{E} = \mathbb{F}_{19}[x]$ using monic degree 1

In this example I'll be using all ones for my input data to see the effect of using known data to encode. Let $R = \{\mu_j = (1 \bmod (x - c_j)) \mid c_0 = 5,\ c_1 = 1,\ c_2 = 2,\ c_3 = 7\}$, encoding moduli set be $\widehat{Q} = \{(x - b_i) \mid b_0 = 3,\ b_1 = 4,\ b_2 = 8,\ b_3 = 11,\ b_4 = 6,\ b_5 = 14\}$.

The first encoding uses only R. Because all the data is constant, the returned value will also be constant: $\nu_i = (1 \bmod (x - b_i))$. Lets say an error occurs in q-relation ν_1, changing it to $\nu_1' = (2 \bmod (x - 4))$. Here's the woven transform:

$\widehat{\nu_j}$	$(x-3)$	$(x-4)$	$(x-8)$	$(x-11)$	$(x-6)$	$(x-14)$
$\dot{\omega}_j^{(0)}$	1	2	1	1	1	1
$\dot{\omega}_j^{(1)}$	$--$	1	0	0	0	0
$\dot{\omega}_j^{(2)}$	$--$		14	8	9	17
$\dot{\omega}_j^{(3)}$	$--$			17	12	10
$\dot{\omega}_j^{(4)}$	$--$				1	4
$\dot{\omega}_j^{(5)}$	$--$					17
$\dot{\omega}_j$	1	1	14	17	1	17

To decode we'll use the set $\overline{Q_4^\pi}$ to compute $Q_4^\pi(\nu_{\pi(4)})$ and $Q_4^\pi(\nu_{\pi(5)})$. If at least one of these matches then we've found a uncorrupted set and can decode. Let $\widehat{\nu_j} = (x - c_j)$

	$\overline{Q^\pi}$						$Q_4^\pi(\nu_0) \stackrel{?}{=} 1$	$Q_4^\pi(\nu_1) \stackrel{?}{=} 2$	$Q_4^\pi(\nu_2) \stackrel{?}{=} 1$	$Q_4^\pi(\nu_3) \stackrel{?}{=} 1$	$Q_4^\pi(\nu_4) \stackrel{?}{=} 1$	$Q_4^\pi(\nu_5) \stackrel{?}{=} 1$
$b_{\pi(j)}$	3	4	8	11	6	14	1	2	1	1	17	4
$\dot{\omega}_{\pi(j)}$	1	1	14	17	1	17						
$b_{\pi(j)}$	3	4	6	14	8	11	1	2	17	14	1	1
$\dot{\omega}_{\pi(j)}$	1	1	9	1	14	17						
$b_{\pi(j)}$	6	14	8	11	3	4	1	1	1	1	1	1
$\dot{\omega}_{\pi(j)}$	1	0	0	0	0	17						

The first two pair swaps creates mismatches in every spare element, but the third pair swap weave has only one error. Since no more than half the spares are errors, the weave creates the original set of q-relations, correcting the error.

This example encodes the exact same data using two known q-relations, $\kappa_0 = (13 \bmod (x-9))$ and $\kappa_1 = (6 \bmod (x-17))$

End of Example C.6

- Example C.7: Weave Coding/Decoding – $\mathcal{E} = \mathbb{F}_{19}[x]$ using degree 1 moduli and salt -

The data used in example C.6 was very repetitive. This example encodes the exact same data using two salt q-relations, $S = \{\kappa_0 = (13 \bmod (x-9)), \kappa_1 = (6 \bmod (x-17))\}$, to mask the data. Adding salt data q-relations adds a little work to the extraction process but does not increase the number of trials or pair swaps required for detecting and correcting errors.

Using the known q-relations S and R (example C.6) produces the coded data

$$Q = \left[\begin{array}{c|cccccc} b_i & 3 & 4 & 8 & 11 & 6 & 14 \\ \dot{\nu}_i & 4 & 15 & 12 & 7 & 15 & 0 \end{array}\right]$$

We'll introduce an error in the q-relation $(15 \bmod (x-4))$ by changing 15 to 10. Here's the weave of the known data S and the received data with the error:

$\widehat{\nu_j}$	$(x-9)$	$(x-17)$	$(x-3)$	$(x-4)$	$(x-8)$	$(x-11)$	$(x-6)$	$(x-14)$
$\omega_j^{(0)}$	13	6	4	10	12	7	15	0
$\omega_j^{(1)}$	$--$	11	11	12	1	16	12	5
$\omega_j^{(2)}$	$--$		0	16	18	15	12	2
$\omega_j^{(3)}$	$--$			16	15	9	4	14
$\omega_j^{(4)}$	$--$				14	18	13	15
$\omega_j^{(5)}$	$--$					14	10	16
$\omega_j^{(5)}$	$--$						16	7
$\omega_j^{(5)}$	$--$							6
$\dot{\omega}_j$	13	11	12	13	8	8	0	6

The known q-relations in the weave, $\overline{S}$, remain in the lowest positions of the weave, never change, and will always be used for checking. The other q-relations will be swapped into the lower positions

as needed.

	$\overline{Q^\pi}$								$Q^\pi_{2+4}(\nu_{\pi(6)})$	$Q^\pi_{2+4}(\nu_{\pi(7)})$
$b_{\pi(j)}$	9	17	3	4	8	11	6	14	6	14
$\dot{\omega_{\pi(j)}}$	13	11	0	16	14	14	16	6	$8 \not\equiv 15$	$2 \not\equiv 18$
$b_{\pi(j)}$	9	17	3	4	6	14	8	11	8	11
$\dot{\omega_{\pi(j)}}$	13	11	0	16	13	5	15	6	$17 \not\equiv 10$	$0 \not\equiv 18$
$b_{\pi(j)}$	9	17	6	14	8	11	3	4	3	4
$\dot{\omega_{\pi(j)}}$	13	11	12	13	8	8	0	6	4	$15 \not\equiv 10$

End of Example C.7