5-1996

# On Quaternionic Pseudo-Random Number Generators

Gary R. Greenfield

# On Quaternionic Pseudo-Random Number Generators

Gary R. Greenfield
Department of Mathematics and Computer Science
University of Richmond
Richmond, Virginia 23173
email: grg5l@mathcs.urich.edu

May, 1996

**TR-96-01**

# On Quaternionic Pseudo-Random Number Generators

Gary R. Greenfield
Department of Mathematics and Computer Science
University of Richmond
Richmond, Virginia 23173

May, 1996

E-mail: grg5l@mathcs.urich.edu

**TR-96-01**

# 1 Introduction.

There is no dearth of published literature on the design, implementation, analysis, or use of pseudo-random number generators or PRNGs. For example, [6] [7] [14] and the references therein, provide a broad overview and firm grounding for the subject. This report complements and elaborates upon the work of McKeever [9], who investigated PRNGs constructed in a *non-commutative* setting with the target application being so-called cryptographically secure PRNGs as discussed in [12] or [13]. Novel "solutions" to the problem of designing cryptographically secure PRNGS continue to be proposed [1] [2] [10] [15], so despite the caution and skepticism required, the area remains active. The concept elaborated upon here is computation in a finite non-commutative object which is more than a matrix ring over a finite field. Specifically, we consider computation in a homomorphic image of a maximal order of an ordinary quaternion algebra. In Section Two we develop the necessary algebraic machinery. In Section Three we consider PRNG design in this computational setting. In Section Four we attempt some preliminary analysis of the PRNGs described. In Section Five we offer some final remarks and conclusions.

# 2 The Ring $H_p$.

Some of the material in this section is a more leisurely paced and more complete version of results which can be found in [4, Chapter 7]. Let $U$ be the division ring of ordinary quaternions over the field of rational numbers $Q$,

$$U = \{a + bi + cj + dk : a, b, c, d \in Q\},$$

and consider the distinguished element $\zeta \in U$ given by

$$\zeta = \frac{1}{2}(1 + i + j + k).$$

Let $Z$ be the ring of integers, and define

$$H = \{a\zeta + bi + cj + dk : a, b, c, d \in Z\}.$$

LEMMA 2.1. *$H$ is a maximal order in $U$.*

*Proof.* To see that $H$ is a subring of $U$ we need the relations:

$$\begin{aligned} i\zeta &= -\zeta + i + k, \\ j\zeta &= -\zeta + i + j, \\ k\zeta &= -\zeta + j + k, \end{aligned}$$

and

$$\zeta i = j\zeta,\ \zeta j = k\zeta,\ \zeta k = i\zeta,$$

in addition to

$$i^2 = j^2 = k^2 = -1,\ \zeta^2 = -\zeta + i + j + k.$$

Next we observe:

$$\begin{aligned} 1 &= 2\zeta - i - j - k, \\ i &= 2\zeta - 1 - j - k, \\ j &= 2\zeta - 1 - i - k, \\ k &= 2\zeta - 1 - i - j. \end{aligned}$$

Now, if $u \in U$ is integral over $Z$, then $mu \in W = Z[i, i, j, k]$ for some $m \in Z$, so the elements $\zeta, i, j, k, u$ are linearly dependent. This proves $H$ is a maximal order in $U$.

We recall that if $\alpha = a + bi + cj + dk \in U$ then the **conjugate** of $\alpha$ is $\overline{\alpha} = a - bi - cj - dk$ and the **norm** of $\alpha$ is

$$N(\alpha) = \alpha\overline{\alpha} = \overline{\alpha}\alpha = a^2 + b^2 + c^2 + d^2.$$

It is routine to verify that $\overline{\alpha\beta} = \overline{\beta}\overline{\alpha}$, and therefore $N(\alpha\beta) = N(\alpha)N(\beta)$. It is equally obvious that conjugation is an involution.

LEMMA 2.2. If $\alpha \in H$, then $N(\alpha) \in Z$.

*Proof.* Write $\alpha = a\zeta + bi + cj + dk$, and compute

$$\begin{aligned} N(\alpha) &= (\frac{a}{2})^2 + (\frac{2b+a}{2})^2 + (\frac{2c+a}{2})^2 + (\frac{2d+a}{2})^2 \\ &= \frac{a^2 + 4b^2 + 4ab + a^2 + 4c^2 + 4ac + a^2 + 4d^2 + 4ad + a^2}{4} \\ &= a^2 + b^2 + c^2 + d^2 + a(b + c + d). \end{aligned}$$

LEMMA 2.3. If $\alpha \in H$, then $\overline{\alpha} \in H$.

*Proof.* $H$ is the disjoint union of the two sets $W = \{a + bi + cj + dk : a, b, c, d \in Z\}$ and $\{\frac{2a+1}{2} + \frac{2b+1}{2}i + \frac{2c+1}{2}j + \frac{2d+1}{2}k : a, b, c, d \in Z\}$, each of which is closed under the operation of conjugacy.

For completeness, we include two additional lemmas which aid in our understanding of the structure of $H$.

LEMMA 2.4. Let $\alpha \in H$. Then $\alpha$ is invertible if and only if $N(\alpha) = 1$.

*Proof.* If $\alpha$ is invertible, then $\alpha\beta = 1$ for some $\beta \in H$, so $N(\alpha)N(\beta) = 1$. Since $N(\alpha)$ is a non-negative integer, $N(\alpha) = 1$. If $N(\alpha) = 1$, then $\alpha\overline{\alpha} = 1$. By the previous lemma, $\overline{\alpha} \in H$ so $\alpha$ is invertible.

The following lemma is a trivial consequence of the classification of the finite subgroups of $U$, but we include here an elementary proof based upon first principles.

LEMMA 2.5. The invertible elements of $H$ are: $\pm 1, \pm i, \pm j, \pm k, \frac{\pm 1 \pm i \pm j \pm k}{2}$.

*Proof.* We again appeal to the realization of $H$ as a union of sets of elements with either integer or "half-odd" integer coefficients. If $\alpha = a + bi + cj + dk$ has integer coefficients and $N(\alpha) = a^2 + b^2 + c^2 + d^2 = 1$, then it is clear that $\alpha = \pm 1, \pm i, \pm j, \pm k$. If $\alpha = \frac{a}{2} + \frac{b}{2}i + \frac{c}{2}j + \frac{d}{2}k$ where $a, b, c, d$ are all *odd* integers and $N(\alpha) = 1$, then $a^2 + b^2 + c^2 + d^2 \leq 4$ whence $|a|, |b|, |c|, |d| \leq 2$ and the remaining units are readily obtained.

DEFINITION 2.6. Let $p$ be an odd prime. We define $H_p = H/pH = \{a\zeta + bi + cj + dk : a, b, c, d \in Z_p\}$.

Observe that $H_p$ is a *finite* non-commutative ring, whence it must have zero-divisors. Zero divisors are easily constructed by first writing $4p$ as a sum of four squares and then constructing $\alpha \in H_p$, with $\alpha\overline{\alpha} = N(\alpha) = p = 0$ in $Z_p$. For example, with $p = 7$, $28 = 4 \cdot 7 = 3^2 + 3^2 + 3^2 + 1^2$, gives $\alpha = \frac{3}{2} + \frac{3}{2}i + \frac{3}{2}j + \frac{3}{2}k = 3\zeta - k$. The key result that follows shows that we are able to characterize *all* the zero divisors in $H_p$.

PROPOSITION 2.7. Let $\alpha \in H_p$. Then $\alpha$ is a zero divisor if and only if $N(\alpha) = 0$ in $Z_p$.

*Proof.* If $N(\alpha) = 0$ in $Z_p$, then $\alpha\overline{\alpha} = 0$ in $H_p$. Conversely, if $\alpha$ is a

zero divisor, let $\beta$ satisfy $\beta\alpha = 0$. Suppose $N(\alpha) \neq 0$. Then there exists an integer $k > 0$ such that $N(\alpha)^k = 1$ in $Z_p$. We have

$$0 = (\beta\alpha)(\alpha^{k-1}\overline{\alpha}^k) = \beta N(\alpha)^k = \beta,$$

which gives a contradiction and completes the proof.

The reader will wonder, however, why we choose to work in $H_p$ rather than the seemingly more "natural" ring $W_p = \{a + bi + cj + dk : a, b, c, d \in Z_p\}$. After all, while it seems instinctively better to work in a maximal object, reducing modulo $p$ cannot change the *size* of the object. Indeed, an explicit bijection from $H_p$ to $W_p$, which is just a change of *representation*, is obtained by defining the function

$$\Theta : a\zeta + bi + cj + dk \longmapsto a' + b'i + c'j + d'k,$$

where $a' = av, b' = av + b, c' = av + c, d' = av + b$ and $v$ is the multiplicative inverse of two in $Z_p$. The answer to the question we posed — and we cannot overstress this point — is that we want multiplication to *appear* as non-linear as possible. The multiplicative relations on the basis $\zeta, i, j, k$ and the norm form on $H$ will help "disguise" the computations considered in the next section.

## 3 Generators over $H_p$.

We recall that a *cryptographically secure* PRNG is one for which it is computationally infeasible to predict the next random number based on observations made from previous random numbers in the sequence. Here, we will take this to mean that, knowing the method of generation, it is computationally infeasible to determine the *parameters* that were selected to initialize the generator. Discovering these parameters constitutes one form of "breaking" the generator. A simple example will illustrate what we mean.

EXAMPLE 3.1. **Quaternionic Linear Congruence Generators (LCGs).** The ordinary linear congruence generator over $Z_p$ is defined in terms of the parameters $\alpha, \beta \in Z_p$ by the recurrence

$$x_{n+1} = \alpha x_n + \beta.$$

From a *seed* $x_0 \in Z_p$ we generate a pseudo-random sequence $x_1, x_2, \ldots$ However, from any three consecutive outputs $x, y, z$ we have $z - y = \alpha(y - x)$ which allows us to determine first $\alpha$ and then $\beta = y - \alpha x$. If we now take $\alpha, \beta \in H_p$, and a seed $x_0$ from $H_p$, the algorithm just given will still break the generator *provided* $y - x$ is invertible in $H_p$. We also remark that an easy induction shows that for all $k > 0$,

$$x_k = \alpha^k x_0 + \frac{\alpha^k - 1}{\alpha - 1}\beta,$$

which shows that the *period* of the generator is the order of $\alpha$ in $Z_p$ (respectively $H_p$).

Our design problem can now be more clearly stated: Using the $H_p$ operations of addition, multiplication, and conjugacy what potential (cryptographically secure) PRNGs might be constructed and studied? The reason it is necessary to ask which generators can be constructed is because there do not exist non-commutative analogs for those commutative generators which require exponentiation to be a binary operation on $H_p$ *i.e.*, those which would rely on a definition of $\alpha^\beta$ where $\alpha, \beta \in H_p$. (A concrete example is given by the Blum-Micali generator $x_{n+1} = g^{x_n}$ where $g$ is a primitive root in $Z_p$.) Another difficulty is that we must prohibit the use of logical bit string operators, for we would be hard pressed to define, for example, either ($\alpha$ OR $\beta$) or ($\alpha$ AND $\beta$) over $H_p$. With these caveats we shall establish four "direct" generalizations of commutative generators.

- **Quaternionic Polynomial Generators (QPGs).**

  Fix $m > 0$ and a polynomial

  $$F(x) = \sum_{s=0}^{m} \alpha_s x^s,$$

  where $\alpha_s \in H_p$ for $s = 0, \ldots, m$. Define the quaternionic polynomial generator using the recurrence $x_{n+1} = F(x_n)$. Note that when $m = 1$ this is the ordinary quaternionic linear congruence generator.

- **Quaternionic Linear Recurrence Generators (LRGs).**

  Let the seed consist of $m > 0$ elements $x_0, x_1, \ldots, x_{m-1} \in H_p$. Fix $\alpha_s \in H_p$ for $s = 0, \ldots, m - 1$. For $n > 0$, let

  $$x_{n+m} = \sum_{s=0}^{m-1} \alpha_s x_{n+s}.$$

- **Quaternionic Cellular Automata (CAs).**

  The CA generators load $m$ cells with initial values (the seed) $x_1^0, \ldots, x_m^0$. This represents stage zero. To compute the $(n+1)$st stage from the $n$th stage, we consider two possible generators, one defined using the rule

  $$x_s^{n+1} = x_{s-1}^n \cdot x_s^n \cdot x_{s+1}^n$$

  and the other using the rule

  $$x_s^{n+1} = x_{s-1}^n \cdot x_{s+1}^n.$$

  The arithmetic on the subscripts takes place in $Z_m$. Since the second rule uses fewer operations, it has efficiency advantages. Regardless of which rule is used, it is clear that the presence of zero in any cell would cause zero propagation, thus we *further stipulate that the seed consist entirely of invertible elements.* The output at each stage is the element of a fixed, but arbitrary, cell.

- **Quaternionic Linear Feedback Shift Registers (LFSRs).**

  The LFSR generator associates coefficients $\alpha_0, \ldots, \alpha_{m-1}$ in $H_p$ to each of its $m$ cells respectively. At stage zero the seed $x_0^0, x_1^0, \ldots, x_{m-1}^0$ is loaded into the cells. To pass from the $n$th stage to the $(n+1)$st stage we set

  $$x_s^{n+1} = x_{s+1}^n,$$

  for $s < m-1$, and

  $$x_{m-1}^{n+1} = \sum_{s=0}^{m-1} \alpha_s x_s^n.$$

  The output at stage $n+1$ is the the pseudo-random number $x_0^n$.

These families do not take full advantage of the rich computational environment provided by non-commutativity. We introduce two mild variations obtained by "twisting." For convenience, we twist at the position of the zero subscripted coefficient.

- **Single Twist LFSRs.** These generators use the recurrence relation

  $$x_{m-1}^{n+1} = x_0^n \alpha_0 + \sum_{s=1}^{m-1} \alpha_s x_s^n.$$

- **Single Twist LRGs.** These generators use the recurrence relation

$$x_{n+m} = x_n\alpha_0 + \sum_{s=1}^{m-1} \alpha_s x_{n+s}.$$

It is now a simple matter to initiate twisting at other locations by formulating *two-sided* generators. For example, we could consider the family of **Two Sided Quaternionic LRGs** defined by

$$x_{n+m} = \sum_{s=0}^{m-1} \alpha_s x_{n+s}\beta_s.$$

This leads naturally to the consideration of a generalized monomial of degree $s$ which takes the form

$$f_s(x) = \alpha_0 x \alpha_1 x \alpha_2 x \cdots \alpha_{s-1} x \alpha_s,$$

and the family of **Generalized Quaternionic Polynomial PRNGs** of degree $m$ defined using the generalized polynomials

$$F(x) = \sum_{s=0}^{m} f_s(x),$$

where each $f_s$ is a generalized monomial of degree $s$, or the family of **Generalized Homogeneous Quaternionic PRNGs** defined by

$$G(x) = \sum_{s=0}^{m} g_s(x),$$

where each $g_s$ is a generalized monomial of *fixed* degree $t$. Finally, we can avail ourselves of the conjugacy operator by denoting $\overline{\alpha}$ by $\tau(\alpha)$ and constructing, for example, **Conjugate Quaternionic LRGs** via

$$x_{n+m} = \sum_{s=0}^{m-1} \alpha_s \tau^{\varepsilon_s}(x_{n+s})\beta_s,$$

where each $\varepsilon_s$ is zero or one. Other variations and modifications may suggest themselves to the reader, but by now our point has been made: a plethora of available examples are to be found over $H_p$. It is time to turn our attention to their analysis.

# 4 On Breaking Quaternionic Generators.

In this section we attempt to offer some insight into algorithms, and some indication of the computational resources required, for breaking a sampling of the quaternionic generators introduced in the previous section. Generators whose coordinate functions are *linear* in the coordinates of their *parameters* — the quaternionic PGs, LRGs, LFSRs and the single twist generators — can, of course, be broken by collecting enough observations to set up linear systems in the $\zeta, i, j$ and $k$ coordinates of the parameters. Surprisingly, as demonstrated by the small order — those which minimize the number of adds and multiplies — examples below, elimination of variables may also be a viable technique.

EXAMPLE 4.1. Consider the single twist LRG

$$x_{n+2} = \alpha x_{n+1} + x_n \beta.$$

Assume we have three consecutive values, $x_s, x_{s+1}, x_{s+2}$. If $x_{s+1}$ is invertible, then

$$\alpha = (x_{s+2} - x_s\beta)x_{s+1}^{-1} = x_{s+2}x_{s+1}^{-1} - x_s\beta x_{s+1}^{-1}.$$

For *each* distinct trio of consecutive values, $x_t, x_{t+1}, x_{t+2}$ where $t > s+2$, and $x_{t+1}$ is invertible,

$$\begin{aligned} x_{t+2} &= \alpha x_{t+1} + x_t\beta \\ &= (x_{s+2}x_{s+1}^{-1} - x_s\beta x_{s+1}^{-1})x_{t+1} + x_t\beta, \end{aligned}$$

which leads to the equation

$$x_{t+2} - x_{s+2}x_{s+1}^{-1}x_{t+1} + x_s\beta x_{s+1}^{-1}x_{t+1} - x_t\beta = 0,$$

or

$$x_{t+2} - x_{s+2}x_{s+1}^{-1}x_{t+1} + x_s(x_{s+1}^{-1}x_{t+1}x_{t+1}^{-1}x_{s+1})\beta x_{s+1}^{-1}x_{t+1} - x_t\beta = 0.$$

If $x_t$ is invertible, then

$$(-x_t^{-1}x_{t+2}+x_t^{-1}x_{s+2}x_{s+1}^{-1}x_{t+1})-(x_t^{-1}x_s x_{s+1}^{-1}x_{t+1})(x_{t+1}^{-1}x_{s+1})\beta(x_{s+1}^{-1}x_{t+1})+\beta = 0.$$

Notice the key step in our reduction: we introduce extraneous constants to "balance" the variable beta so that it can be replaced by a (group-theoretic)

conjugate. Using constants from $H_p$ ensures that this conjugate is still *linear* in the coordinates of beta. We shall make reference to this technique in subsequent examples.

Returning to the derivation, setting

$$\begin{aligned} a_t &= x_t^{-1}x_{s+2}x_{s+1}^{-1}x_{t+1} - x_t^{-1}x_{t+2}, \\ b_t &= x_t^{-1}x_s x_{s+1}^{-1}x_{t+1}, \\ c_t &= x_{t+1}^{-1}x_{s+1}, \end{aligned}$$

the previous equation simplifies to

$$a_t - b_t c_t \beta c_t^{-1} + \beta = 0.$$

Switching to coordinates, we expand this last equation to

$$\begin{aligned} 0 &= (a_{t,\zeta} + \beta_\zeta - f_\zeta(b_t, c_t, \beta))\zeta \\ &\quad + (a_{t,i} + \beta_i - f_i(b_t, c_t, \beta))i \\ &\quad + (a_{t,j} + \beta_j - f_j(b_t, c_t, \beta))j \\ &\quad + (a_{t,k} + \beta_k - f_k(b_t, c_t, \beta))k, \end{aligned}$$

where the coordinate functions $f_\zeta, f_i, f_j, f_k$ are *linear* in $\beta_\zeta, \beta_i, \beta_j, \beta_k$. The strategy is now clear. We collect sufficient $x_t, x_{t+1}, x+t+2$ trios to determine the coordinates for $\beta$, and then solve for $\alpha$.

We should mention that in [9] the method for breaking the closely related quaternionic homogeneous generator $x_{n+1} = \alpha x_n + x_n\beta$, by reverting to coordinates is outlined. Our second example points to the difficulty of scaling-up when using the previous technique.

EXAMPLE 4.2. Consider the single twist LRG

$$x_{n+3} = \alpha x_{n+2} + \beta x_{n+1} + x_n\gamma.$$

We shall not explicitly draw attention to the necessary invertibility assumptions on the subsequences of observations of length four that we will be working with. Start with four consecutive values, $x_s, x_{s+1}, x_{s+2}, x_{s+3}$. Then

$$\begin{aligned} \alpha &= (x_{s+3} - \beta x_{s+1} - x_s\gamma)x_{s+2}^{-1} \\ &= x_{s+3}x_{s+2}^{-1} - \beta x_{s+1} - x_s\gamma x_{s+2}^{-1}. \end{aligned}$$

From another quadruple $x_t, x_{t+1}, x_{t+2}, x_{t+3}$,

$$\begin{aligned} x_{t+3} &= (x_{s+3}x_{s+2}^{-1} - \beta x_{s+1} - x_s\gamma x_{s+2}^{-1})x_{t+2} + \beta x_{t+1} + x_t\gamma \\ &= x_{s+3}x_{s+2}^{-1}x_{t+2} - x_s\gamma x_{s+2}^{-1}x_{t+2} - \beta(x_{s+1}x_{t+2} - x_{t+1}) + x_t\gamma, \end{aligned}$$

whence

$$\beta = [x_{s+3}x_{s+2}^{-1}x_{t+2} - x_s\gamma x_{s+2}^{-1}x_{t+2} + x_t\gamma](x_{s+1}x_{t+2} - x_{t+1})^{-1}.$$

By introducing the appropriate balancing coefficients, this describes $\beta$ as a linear equation in *two* conjugates of $\gamma$ and back substitution allows us to rewrite $\alpha$ as a linear equation in *three* conjugates of $\gamma$. Any subsequent quadruple generated by the sequence say, $x_u, x_{u+1}, x_{u+2}, x_{u+3}$, gives

$$\begin{aligned} x_{u+3} &= \alpha x_{u+2} + \beta x_{u+1} + x_u\gamma \\ &= (\text{three } \gamma\text{-conjugates})x_{u+2} + (\text{two } \gamma\text{-conjugates})x_{u+1} + x_u\gamma, \end{aligned}$$

and therefore expresses the zero element of $H_p$ as a linear equation in *seven* conjugates of $\gamma$. Reverting to coordinates yields a linear system to be solved for $\gamma$, though several quadruples may be necessary to satisfy the invertibility requirements and to determine all coordinates.

Though a complete, detailed solution appears awkward to describe, it seems clear that the "elimination methods" above can be used to break any single twist LRG. For our final examples, we turn our attention to a low order polynomial generator.

EXAMPLE 4.3. Recall the quadratic quaternionic polynomial generator,

$$x_{n+1} = \alpha x_n^2 + \beta x_n + \gamma.$$

From an output pair $x_s, x_{s+1}$ with $x_s$ invertible, we obtain

$$\alpha = (x_{s+1} - \beta x_s - \gamma)x_s^{-2}.$$

From a subsequent pair $x_t, x_{t+1}$ we get

$$x_{t+1} = (x_{s+1} - \beta x_s - \gamma)x_s^{-2}x_t^2 + \beta x_t + \gamma$$

which allows us to solve for beta,

$$\beta = [(x_{t+1} - x_{s+1}x_s^{-2}x_t^2) - \gamma(1 - x_s - 2x_t^2)](x_t - x_s^{-1}x_t^2)^{-1}.$$

As before, we can back substitute to find $\alpha$ in terms of $\gamma$ and use additional pairs to solve for $\gamma$.

EXAMPLE 4.4. To break the twisted quadratic quaternionic polynomial generator,

$$x_{n+1} = \alpha x_n^2 + x_n\beta + \gamma,$$

first use $x_s, x_{s+1}$ to get

$$\alpha = (x_{s+1} - x_s\beta - \gamma)x_s^{-2},$$

then $x_t, x_{t+1}$ to find

$$\begin{aligned} x_{t+1} &= (x_{s+1}x_s^{-2} - x_s\beta x_s^{-2} - \gamma x_s^{-2})x_t^2 + x_t\beta + \gamma \\ &= x_{s+1}x_s^{-2}x_t^2 - x_s\beta x_s^{-2}x_t^2 + x_t\beta - \gamma(x_s^{-2}x_t^2 - 1). \end{aligned}$$

Therefore

$$\gamma = (x_{s+1}x_s^{-2}x_t^2 - x_{t+1} - x_s\beta x_s^{-2}x_t^2 + x_t\beta)(x_s^{-2}x_t^2 - 1)^{-1}.$$

As in the previous example, back substitution into $\alpha$, and additional output pairs, allow one to solve for $\beta$.

Some of the other small order generators can be easily transformed into generators that would succumb to the above methods. For example, for the purposes of recovering parameters, we observe that

$$x_{n+1} = \alpha x\beta + \gamma,$$

is equivalent to

$$x_{n+1}\overline{\beta} = N(\beta)\alpha x_n + \gamma\overline{\beta},$$

or

$$x_{n+1}\beta' = \alpha' x_n + \gamma',$$

where $\beta' = \overline{\beta}$, $\alpha' = N(\beta)\alpha$, $\gamma' = \gamma\overline{\beta}$. The latter, when broken into components, has linear coordinate functions.

The use of conjugation does not have any significant effect on the examples and methods considered previously. In the simplest case,

$$x_{n+1} = \alpha\overline{x_n} + \beta,$$

it is routine to establish that from a suitably invertible recurrence triple $x, y, z$ one can solve $\alpha = (z - y)(\overline{y} - \overline{x})^{-1}$, and then recover $\beta$.

It would therefore seem that the two simplest nonlinear generators which our methods do not address are

$$x_{n+1} = \alpha x_n \beta x_n \gamma,$$

and

$$x_{n+1} = \alpha x_n \beta + \gamma x_n \delta.$$

The source of record concerning (generalized) polynomials over a division ring [5] is concerned with their zeros, and more recent work [3] has focused on their factorization. We are not aware of any significant results about their reconstruction based on a set of their values, or the properties of their iterates.

# 5 Concluding Remarks.

The hallmarks of a "good" PRNG are that it exhibits the following three properties:

1. All seeds should give rise to long sequences without repetition;
2. Outputs should satisfy acceptable statistical or theoretical "randomness" criteria;
3. Fast, efficient implementations should be possible.

Regarding long sequences without repetition, or "loops," we have already indicated that the loop length of the ordinary quaternion LCG $x_{n+1} = \alpha x_n + \beta$ is determined by the order of $\alpha$ in $H_p$. Similarly, for the generator $x_{n+1} = \alpha x_n + x_n \beta$, routine induction gives $x_n = \sum_{s=0}^{n} \binom{n}{s} \alpha^{n-s} x_0 \beta^s$, which shows that this generator too is sensitive to the orders of $\alpha$ and $\beta$ in $H_p$. In general, iterates of the remaining non-commutative recurrence generators seems difficult to evaluate in this regard.

This paper has not investigated the randomness characteristics of any of the generators discussed. In [9] on the basis of data collected from the generators $x_{n+1} = \alpha x_n \beta + \gamma$, $x_{n+1} = \alpha x_n + x_n \beta$, and a five-cell quaternionic CA, it was suggested that only quaternionic CAs held any promise for exhibiting acceptable randomness. It is important to note, however, that the data collected in those experiments consisted of "taps" from our prescribed

outputs. This simply means that only a fixed *coordinate* of the output was recorded. The rationale for this is exquisite: While we may have methods for breaking (linear) generators given the normal (default) outputs, we are hopelessly stymied when we are confronted by nothing more that a sequence of "taps" from these outputs.

There is some inherent efficiency to be gained from implementing multiplication in $H_p$ through $4 \times 4$ matrix multiplication (see Appendix A below). The standard speed-up for computing polynomial values over a field is Horner's Method. Here is a non-commutative variation, employing "twisting" at each iteration according to a boolean array `twist`, whose consequences we have not studied:

```
F = alpha[n]}
for s = n-1 downto 0
   if (left[s])
     F =  F * x + alpha[s]
   else
     F =  x * F + alpha[s]
```

The reader will no doubt think of other possibilities along these lines.

There are no known "attacks" for breaking CAs with or without "taps." The CA loop structure also remains obscure, even in the commutative case [16]. Their chief drawback remains their excessive computational overhead. Given the fact that both loop length and randomness characteristics have been successfully determined for commutative LCGs [7], and the implementation issues for these LCGs have been carefully researched [14], we have every reason to believe that in the future more complete and equally satisfactory results for non-commutative generators might be obtained.

# A The Group of Units of $H_p$.

It was conjectured in [9] that the cardinality of the set of zero divisors in $H_p$ is

$$\begin{aligned} p^3 + p^2 - p - 1 &= (p^3 - p) + (p^2 - 1) \\ &= p(p^2 - 1) + (p^2 - 1) \\ &= (p+1)(p^2 - 1) \\ &= (p-1)(p+1)^2. \end{aligned}$$

This would imply that the set of non-invertible elements — the zero divisors together with the zero element — has cardinality $p^3 + p^2 - p$, and therefore that the cardinality of the set of invertible elements is

$$\begin{aligned} p^4 - (p^3 + p^2 - p) &= p(p^3 - p^2 - p + 1) \\ &= p(p^2(p-1) - (p-1)) \\ &= p(p^2 - 1)(p-1) \\ &= p(p+1)(p-1)^2. \end{aligned}$$

This is significant for two reasons. First is predicts that for *large* $p$ the number of invertible elements is effectively $p^4$. Second is suggests that the group of units of $H_p$ might be isomorphic to the group $GL(2,p)$, since the latter is known to be of this order [11, Theorem 8.13].[1]

Recall that there is an isomorphic embedding $U \hookrightarrow M_2(Q(i))$ given by

$$a + bi + cj + dk \longmapsto \begin{pmatrix} a+bi & c+di \\ -c+di & a-bi \end{pmatrix}.$$

(Replacing the complex entries of this matrix by the $2 \times 2$ matrix equivalents with rational entries gives the embedding of $U$ into $M_4(Q)$.) We will attempt to carry this construction over to $H_p$.

Set $v = (p+1)/2$ in $Z_p$, so $2v = 1$ in $Z_p$, and let $y$ be a "symbol" satisfying $y^2 = -1$ in $Z_p$. Of course, if $p = 1 \pmod 4$, we must take $y = g^{(p-1)/4}$ where $g$ is a generator for the group of mon-zero elements in $Z_p$. Define $\Psi : H_p \hookrightarrow M_2(Z_p[y])$ via

$$a\zeta + bi + cj + dk \longmapsto \begin{pmatrix} av + (av+b)y & (av+c) + (av+d)y \\ -(av+c) + (av+d)y & av - (av+b)y \end{pmatrix}.$$

The map $\Psi$ "unpacks" $a\zeta + bi + cj + dk$ to $(a/2) + (a/2+b)i + (a/2+c)j + (a/2+d)k$ and then uses $v$ in place of 2, and $y$ in place of $i$, so that the new embedding is well-defined. It is trivial to prove that $\Psi(h_1 + h_2) = \Psi(h_1) + \Psi(h_2)$ and that $\Psi$ is one-to-one. It is harder to check that $\Psi(h_1 h_2) = \Psi(h_1)\Psi(h_2)$. Now, if $p = 1 \pmod 4$, then $Z_p[y] = Z_p$, so $\Psi$ is onto and we have established an isomorphism between the units of $H_p$ and $GL(2,p)$. But, if $p = 3 \pmod 4$, then $Z_p[y] \simeq GF(p^2)$, and we have established an isomorphism from $H_p$ into a subgroup of $GL(2, GF(p^2))$. We are unable to determine the image of the map $\Psi$ in this case.

[1] I thank Dan Frohardt-Lane for reminding me of this fact.

## References


[1] L. Blum, M. Blum and M. Shub, A simple unpredictable pseudo-random number generator, *SIAM Journal of Computing*, Volume 15, Number 2, May 1986, 364–383.

[2] D. Coppersmith, H. Krawczyk and Y. Mansour, The shrinking generator, in *Advances in Cryptology — CRYPTO '93*, 13th Annual Cryptology Conference Santa Barbara, California, USA, August 22–26 1993 Proceedings, D. Stinson (ed), Springer-Verlag, 1994.

[3] D. Haile and L. Rowen, Factorizations of polynomials over division algebras, *Algebra Colloquium*, Volume 2, Number 2, 1995, 145–156.

[4] I. Herstein, *Topics in Algebra* 2nd edition, Xerox Corporation, 1975.

[5] B. Gordon and T. Motzkin, On the zeros of polynomials over division rings, *Transactions of the American Mathematical Society*, Volume 116, 1965, 218–226; correction Volume 122, 1966, 547.

[6] P. L'Ecuyer, Random numbers for simulation, *Communications of the ACM*, Volume 33, Number 10, October 1990, 85–97.

[7] D. Knuth, *The Art of Computer Programming* Volume 2 *Seminumerical Algorithms*, Addison-Wesley, 1981.

[8] G. Marsaglia, Remarks on choosing and implementing random number generators, *Communications of the ACM*, Volume 36, Number 7, July 1993, 105–110.

[9] B. McKeever, The Use of Non-Commutative Algebra in Cryptographically Secure Pseudo-Random Number Generators, Honors thesis, University of Richmond, 1996.

[10] D. Mitchell, Nonlinear key generators, *Cryptologia*, Volume 14, Number 4, October 1990, 350–354.

[11] J. Rotman, *The Theory of Groups*, Allyn and Bacon, 1965.

[12] R. Rueppel, *Analysis and Design of Stream Ciphers*, Springer-Verlag, Berlin, 1986.

[13] B. Schneier, *Applied Cryptography : protocols, algorithms, and source code in C*, Wiley, 1994.

[14] S. Park and K. Miller, Random number generators : Good ones are hard to find, *Communications of the ACM*, Volume 31, Number 10, October 1988, 1192–1201.

[15] K. Zeng *et al*, Pseudorandom bit generators in stream-cipher cryptography, *IEEE Computer*, February 1991, 8–17.

[16] S. Wolfram, *Cellular automata and complexity : collected papers Stephen Wolfram*, Addison-Wesley, 1994.