Introduction

This paper presents a potential 'one way' mathematical function mediated through iterative forward-coding of two scaled, fractional variables. These variables are derived from the quantized step-error of the inverse function. The equation's modulated output constitutes a binary walk through an information space, maintaining discrete parameterizations which enable the enfolding conservation of a single 'bit' of information in addition to the function product. As the delta of the function and it's inverse describe a ternary probability inheritance tree, it is shown that such a function cannot be self-contained in terms of total entropy production when the 'spare bit' is used to operate upon a separate data set and as such, necessitates a 'trap-door' table to preserve an additional single 'bit' of information required to correctly derive the inverse function.
Furthermore, such a trap-door table represents an essential index in order for the equation to operate as a collision free hash function. The splitting of the equation's output between a cumulative, discretely iterated, binary hash function and a separate binary trapdoor table means that without benefit of access to both information sources, the inverse function is consequently encrypted in exponential time.

Theorem

There exists a (relatively) well-conditioned, iterative, two-step equation

$f(x) = (x^2 - c)^2 - c$

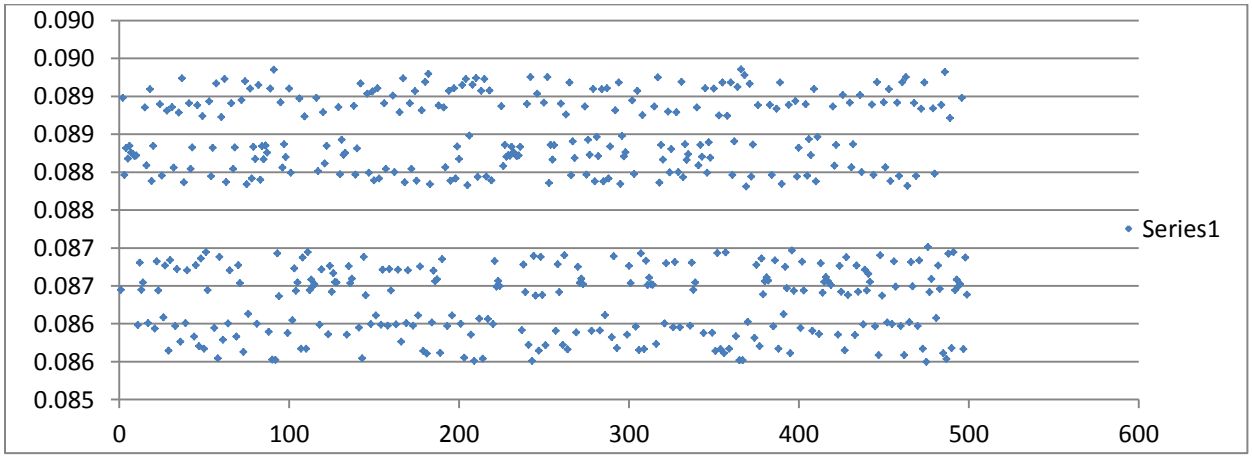
For expository purposes, assuming: $i = .001$, $c = 1.095 \pm i$

And $P[c_1 \cup c_2] = 1$

Such that the $Var[c_1, c_2]$ exhibits some, pseudo-random, binomial distribution.

Then $f(x_n) \rightarrow f(x_{n+1})$ traverses a (pseudo-random), binary decision tree.

Where functional parameterization of $[c_1, c_2]$ is constrained within four discrete bands, $|x1_{u,v}|x2_{y,z}|$



*Pseudo-random distribution of function $f(x)$ variables $[c_1, c_2]$ over 500 iterations

**Insert limit function here...*

For the regression $x_n \to x_{n-1}$ of the inverse function, $f^{-1}(x)$

Where, $f^{-1}(x_n) = \sqrt{\sqrt[-1]{(x_{n+}c_{z)}} + c_z}$

With reliable machine operation, at some specified floating point precision, $r \geq 5, \leq 15\ U.P.$

Where x_n is a rounded, truncated, floating point integer of precision r.

Then, within the prescribed parameters, the inverse function is well enough conditioned that

$f^{-1}(x_n) \to f^{-1}(x_{n-1})$ rests within$\pm$ one u.l.p of $f(x_{n-1})$

And as such, the inverse function maps a surjective, ternary, probability inheritance node, $T \Rightarrow T_{l,m,r}$

Such that, $f^{-1}(x_n) - f(x_{n-1}) \overset{\Delta}{\to} T_{l,m,r}$

Where the inverse function's default is always T_m but the 'child' leaf with true correspondence to $f(x_{n-1})$ may be any one of $T_{l,m,r}$.

Returning to the expository assumption of random allocation of variables, $[c_1, c_2]$

Assuming that for function $f(x_{n+1})$, we may forward cast either of the variables $[c_1, c_2]$ in terms of the preceding node's ΔT…

Such that, $T_l = c_1, T_r = c_2$ and allowing $T_{m_{[c_1,c_2]}}$ to operate on some external binary set, S.

As $P[T_l, T_m, T_r] = 1$, then the '*free*' binary coding opportunity afforded in the case $T_{m_{[c_1,c_2]}}$ has $\sim P\left[\frac{1}{3}\right]$

Because of the default behaviour, $f^{-1}(x) \Rightarrow T_m$, there is no way to verify if $T_m \triangleq f(x_{n-1})$

Unless the inheritance identity of $\Delta T f(x_{n-1})$ has been separately recorded in a binary trapdoor file.

Discussion

Although the resultant 'trapdoor-table' output is larger than the external data set operated upon, the lower entropy of this file indicates it should remain compressible by conventional means.

With function precision of an indicated 14 decimal places, given that the leading decimal place is zero and the second place-holder of significance is also unchanging, this only yields an ostensible '96 bits' of encryption strength. Even that is predicated upon a reduced range, the function 'key-set' as presented being restricted to decimal numbers. By encryption standards, in this most basic form, the function is weak in terms of 'worst case performance' predicated upon a 'lucky guess'. These issues are addressed in the accompanying spreadsheet model via the sequential iteration of an additional static keyset, spreading encryption robustness across multiple iterative steps.

The described operation of the function and its inverse could be characterized as a semi -deterministic Turing machine. Iterative function nodes identified by the inverse function as being 'truly' $'T_m'$ may be freely forward cast according to some external binary data set. Those inverse regressions identified $'T_l, T_r'$, must be forward cast with their corresponding binary variable identity.
As such, the function output is split between the enfolding cumulative binary identity of the hash function and the corresponding binary indexing function of the separately maintained 'trapdoor' table.

A 'two-bit' equation…

A continuous, boolean, fractional equation, cannot function as a deterministic Turing machine with singular binary entropy conservation unless the output constitutes some random 'Oracle' sequence. This is impossible over a dismally finite number field, unless the 'Oracle' is also a finite loop.
Without a trapdoor file, the equation alone would also have to operate as a collision free hash function. The surjective, ternary error relationship between this equation and its inverse would therefore appear to represent a functional limit upon entropy conservation in continuous fractional equations.

In conclusion, without access to the hash key(s), the trapdoor index is encrypted to the order of, $O(2^n)$

somerville/unitambo/24/04/2014