Lessons learned from “RLC FEC Scheme” spec. @ TSVWG

Vincent Roca, Inria PRIVATICS, vincent.roca@inria.fr

Avril 2019

A long process…

- Two reasons:
  1. we first missed the Pseudo-Random Number Generator issue
     - ***Park Miller Linear Congruential PRNG is broken!***
     - see my IETF102 NWCRG slides
  2. specifying a PRNG **through its C reference implementation is complex**
     - **copyright** and **license** issues if C ref. implementation are not I-D authors

```
Copyright Notice

   Copyright (c) 2019 IETF Trust and the persons identified as the
   document authors.  All rights reserved.

   This document is subject to BCP 78 and the IETF Trust's Legal
   Provisions Relating to IETF Documents
```

     - C specification raises **interoperability** concerns
     - …sometimes rather subtle (neg. value representation is not standardized)
     - pseudo-code is preferred but sometimes non available

TinyMT32 PRNG

- Tiny Mersenne Twister, 32-bit version
  - compact version of the renown/widely used Mersenne Twister PRNG
    - see https://en.wikipedia.org/wiki/Mersenne_Twister)
  - provable quality ☺
  - comes with a reference C implementation

  - fixed 3 internal parameters for simplicity

TINYMT32_MAT1_PARAM	0x8f7011ee
TINYMT32_MAT2_PARAM	0xfc78ff1f
TINYMT32_TMAT_PARAM	0x3793fdff

    - those are good official values
    - many other triples could be used but lead to different pseudo-random number sequences

TinyMT32 PRNG (2)

- determinism is a MUST. We provide **tables of 50 first 32-bit PRNG values**
- we checked on various 64-bit, 32-bit, 16-bit and 8-bit platforms (MacOS, Linux, RIOT)

```
2545341989  981918433 3715302833 2387538352 3591001365
3820442102 2114400566 2196103051 2783359912  764534509
 643179475 1822416315  881558334 4207026366 3690273640
3240535687 2921447122 3984931427 4092394160   44209675
2188315343 2908663843 1834519336 3774670961 3019990707
4065554902 1239765502 4035716197 3412127188  552822483
 161364450  353727785  140085994  149132008 2547770827
4064042525 4078297538 2057335507  622384752 2041665899
2193913817 1080849512   33160901  662956935  642999063
3384709977 1723175122 3866752252  521822317 2292524454

           Figure 2: First 50 decimal values returned by
tinymt32_generate_uint32(s) as 32-bit unsigned integers, with a seed
                              value of 1.
```

- Not included: scaling the 32-bit pseudo-random number to a sub-space
  - e.g., RLC (but not TinyMT32) specifies two functions to map to 4-bit and 8-bit spaces (needed by coding coefficient generation)

The good news

- I-D has been mostly reviewed by IESG, should be published by next IETF

TSVWG
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2019

M. Saito
M. Matsumoto
Hiroshima University
V. Roca (Ed.)
E. Baccelli
INRIA
March 5, 2019

TinyMT32 Pseudo Random Number Generator (PRNG)
draft-roca-tsvwg-tinymt32-01

Abstract

This document describes the TinyMT32 Pseudo Random Number Generator (PRNG) that produces 32-bit pseudo-random unsigned integers and aims at having a simple-to-use and deterministic solution. This PRNG is a small-sized variant of Mersenne Twister (MT) PRNG, also designed by M. Saito and M. Matsumoto. The main advantage of TinyMT32 over MT is the use of a small internal state, compatible with most target platforms including embedded devices, while keeping a reasonably good randomness.