Advance Notice of AHS-RNG Version 2.0

The release of the reference programs has been delayed. The technical reason for this is that we want to incorporate three important improvements (see below) and avoid releasing version 2.0 only to follow up with version 3.0 just three months later. The personal reason is that an individual complaint we filed with the European Court of Human Rights (Application No. 12618/23) has led to a ruling that we consider seriously flawed, both in terms of procedural law and in terms of the characterization of our work. We are preparing a separate publication in which the relevant documents and analyses will be presented. In the meantime, development of AHS-RNG will continue at a reduced capacity. The following technical announcements describe the improvements in version 2.0 and the results of the avalanche tests that validate them.

If anyone would like to test the 2-lane 1.0 variant now, he can send a request by email to sicapas@pt.lu . We will send him the program for generating a random number file and a parameter file with 500 run numbers.

Introduction of randomly generated FAAPs and new layout of the parameter file

In the first version, we used a hand-selected FAAP table as a constant in the commands. We have found that the use of stored constants does not result in any speed disadvantage, at least not in our test configuration with Zen2 7702P CPUs.

The introduction of individual FAAPs per run requires a change to the parameter file. Additionally, the other extensions need to be prepared, so we have defined a new layout for the parameter file. Seeds for XOshiro have now also been added (4 x 64 bits, the first one for both LCG and XOshiro). We then inserted a randomly generated FAAP table, 16 x 16 bits, i.e., 256 bits. During generation, we verify that at least 3 bits are “1” per 16-bit row, which is why only one table out of approximately four randomly generated FAAPs is accepted. Nevertheless, the number of possible FAAP tables remains over 10^18. Since four different FAAPs are used in 4-lane, the number of possible combinations is over 10^72, and in 2-lane it is still over 10^36.

Following the FAAP, a 64-byte table for the BAP parameters is appended.

Introduction of 4 parallel streams, called 4-lane

Since extending the algorithm to individual FAAPs significantly increases randomness, we investigated whether it is possible to generate 4 parallel streams—that is, 4 bits per 64-bit cycle. This, of course, precludes the possibility of extending the BFT to 64 or 256 KB. In our view, this is not a disadvantage if different FAAPs are used. For an 8 KB BFT, there are 10^19725 variants. In initial tests, we measured a speedup of over 23% for the “4-lane” variant compared to the “2-lane” variant. At 75% CPU load with 64 cores, we achieve over 900 Mbit/s with the 4-lane variant. We want to retain the distinction between 1-lane, 2-lane, and 4-lane, as this clearly explains the unique characteristics of the different variants. For example, imagine three car factories producing 1, 2, or 4 cars at the same clock speed on different production lines. When the gates open, the car from the first assembly line is the first to enter the single-lane exit. With 1-lane, that is the entire output for this cycle.

With 2-lane and 4-lane, a second car follows the first, entering the single-lane exit after the first car. With 4-lane, a third car follows, and finally the fourth. Then a new cycle begins.

Introduction of BAP – (Bit Assembling Permutations)

In the parameter table, we have also introduced a 64-byte area to allow the permutation of the order in which the 64-bit random numbers are assembled. We refer to this new method as BAP (Bit Assembling Permutations). During the assembly of the individual bits within half a 64-bit word (i.e., 32 bits), the order in which the "fished" bits are inserted can be freely chosen, independently for the left and for the right half of the 64-bit word. The number of possible permutations thus amounts to 32! * 32! = 6,923… * 10^70. This increases the guaranteed security period against quantum computers from 100,000 years to 100 million years, representing another significant advancement 😊 . This statement is, of course, merely a metaphor, because scientifically speaking, the state space actually exceeds the number of possible quantum operations over the entire lifetime of the universe by many orders of magnitude. Decrypting a message encoded with AHS-RNG and RPP-OTP is not computationally feasible within the time span until the Earth's demise, by using any known method—whether classical or quantum-based—on any conceivable hardware.

However, this option has the disadvantage that the speed in 4-lane mode decreases by approximately 8%, as measured with the first, not yet optimized version of the code. In the 4-lane variant with BAP, AHS-RNG still generates 828 Mbit/s, compared to 733 Mbit/s in the standard 2-lane variant. In cryptography, this opens up the possibility of potentially using this as a 320-bit additional key. These permutations can be viewed as a 7th mixer. It is not yet integrated into the AHS demo on sicap.lu.

In an extensive series of tests, we will also examine whether BAP can better meet the requirements for AHS-secret. As is well known, an external random source is incorporated into the program's flow via nanosecond-precise time variability, which cannot be traced later. This may be easier and better to implement by swapping two or four BAP values at each interrupt. This approach seems promising to us and naturally requires an extensive study. We will report on the details.

FAAP tests to demonstrate the avalanche effect

To demonstrate that the FAAP indeed exhibits the expected avalanche effect, we have done 5,000 Bigcrush TestU01 runs three times to test the FAAP avalanche effect. The FAAP consists of 256 bits (top left in the AHS demo) and contains 64 "1" bits to determine the selection of bits from four BRVs. Each position in the 16-bit row can be visualized as a small 4x4 mini-chessboard. The rows A1 through A4, B1 through B4, C1 through C4, and D1 through D4 yield the 16 values, for one specific bit-position. One must now place a rook on each of these squares per row, such that no rook attacks another. There are 24 permutations that satisfy this requirement. We now prefer to use the analogy of rooks instead of the term "queens," which we mentioned in the 2006 white paper. Rooks better illustrate the possible positions, as the rule about not attacking is easy to understand.

In each of the three test-series, two streams of random numbers are generated 5,000 times, each spanning over 11 trillion bits for the Bigcrush test. The first stream is based on a run number from the new parameter table, with a random FAAP. For the second stream per test,

we use all values from the identical parameter table, meaning BFT, Seeds, and BM are identical. We have shifted only in the FAAP 2, 4, or 8 rooks. Shifting 2 rooks is the absolute minimum, since the condition of not attacking must always be met. Since two positions are set from “1” to “0” and two positions from “0” to “1,” we call the first test Modi4. Accordingly, Modi8 and Modi16 for the second and third. Modi 4 and Modi 8 refer to a randomly selected position out of the 16 possible ones, Modi 16 to two randomly selected positions. To account for all possibilities, the changes are selected cyclically from the possible variants.

The results are astonishing and fully confirm the avalanche effect of FAAP. Even with the first 64-bit random number, there was a 12-bit difference between the two variants in modi4, due only to the effect of two shifted bits. After the first 64-bit generation, the difference between the two streams is not solely due to the different composition of the “Final Address” by the two different FAAPs, but the feedback of the most recently generated random number to the four feedback modifiers (FBMs) accelerates the separation into independent streams. Statistically, random results can occur in the various 16-bit feedback loops, resulting in a slight increase in the measured p-values < 0.001 and p-values > 0.999. The average across the 250,000 Bigcrush tests is 0.2078% of all p-values. For modi4, we have a value of 0.354…%, for modi8 of 0.2611…%, and for modi16 only 0.2181…%. It is very important to note that there are no real statistical outliers, such as those observed, for example, in the two TRNG (True Random Number Generator) BigCrush tests of the IDQuantique generator. In these two tests, 67 p-values out of 508 were marked with *****, corresponding to a percentage of 13.1889…%. However, 11 of these p-values are smaller than 10^-300 (eps) and another 17 are smaller than 10^-20. We analyzed the p-values we got from the three test campaigns modi4, modi8 and modi16. Out of always 5000 Bigcrush runs we got 1’270’000 p-values per campaign. We noticed the most extreme p-value for the left tail as 1.3e-12, 6.6e-8 and 1.9-7 for modi4, modi8 and modi16. On the right tail the most extreme p-values out of always 1’270’000 is 1 – 1.4e-14, 1 – 2.8e-8 and 1-1.0e-7, for modi4, modi8 and modi16.

In the FAAP avalanche effect tests, we see indeed a slight increase of the number of p-values in the one-per-thousand range at the left and right tails compared to the average of the 250,000 BigCrush runs with 5 RNGs. A calculation of the frequencies dispels all concerns regarding this statistical clustering. First, this is an extreme test designed to investigate the contribution of FAAP to achieving perfect randomness. Mathematically speaking, this case of statistical anomaly occurs extremely rarely: once every 30,608 billion bits for modi4, once every 84,206 billion bits for modi8, and for modi16, once every 431,818 billion bits. Everything points to a minor statistical effect that decreases with the number of shifted “rooks.” When 8 of the 64 bits of the FAAP are changed, the effect has almost completely disappeared. The difference between the two streams is a very good approximation of a true random sequence. Should anyone be interested in a mathematical analysis of the FAAP and the test results, we will of course make all raw data available.