ॐ भूर्भुवस्व तत् सवितूर्वरेण्यम् भर्गो देवस्य धीमहि धियो यो न प्रचोदयात्

VEST
Hardware-Dedicated Stream Ciphers

Sean O'Neil, Benjamin Gittins, Howard A. Landman
CB Capital Management S.A.

Abstract: VEST hardware-dedicated stream ciphers are based on bijective non-linear parallel feedback shift registers assisted by non-linear Remainder Number System (RNS) based counters. Three VEST cipher family trees are introduced: VEST-4, VEST-16 and VEST-32. VEST-4 returning 4 bits of output per clock cycle occupying ~4K ASIC gates offers 80-bit security, perfect for low-area low-cost applications such as RFID and smartcards. VEST-16 returning 16 bits of output per clock cycle occupying ~12K ASIC gates offers 160-bit security, perfect for high-speed low-area low cost applications. VEST-32 returning 32 bits of output per clock cycle occupying ~20K ASIC gates offers 256-bit security, perfect for high-speed high security applications such as processor-bus encryption. All VEST ciphers support variable key sizes and instant re-keying, and all VEST ciphers release output on every clock cycle.

1. Introduction

Most modern hardware-dedicated stream ciphers and hash functions are built on well-understood linear feedback shift registers (LFSRs) with non-linearity introduced by stateless non-linear combiners and/or irregular clocking. Purely non-linear feedback shift registers (NLFSRs), especially the ones with parallel feedback including word-based NLFSRs, are an attractive choice for makers of stream ciphers who unfortunately have to face the open problems of constructing NLFSRs with known or predictable periods and of determining periods of NLFSRs to ensure a guaranteed minimum period length of their cipher. Only the average period of a large NLFSR can be easily calculated. Contrary to that, it is trivial to construct an LFSR with a known and predictable maximal period length. However, LFSRs suffer from high predictability. Because of that, all linear components are nearly completely dismissed by the attackers as not contributing much to the ciphers' security. Thus, the marriage of both components has become a popular construction for stream ciphers: LFSRs are used to guarantee a minimal period length, and sufficiently iterated NLFSR combiners introduce the essential non-linearity and high algebraic degree into the relationships between keystream bits.

A new generation of counter-assisted NLFSR ciphers is introduced here with two major design improvements achieved by adding parallel feedback to the main bijective NLFSR combiner (core accumulator) and by replacing the linear counter with a group of smaller NLFSR-based counters with a guaranteed long combined period.

Careful addition of parallel feedback to the NLFSR combiner (core accumulator) significantly accelerates diffusion while maintaining NLFSR's bijective (reversible) operation and the extra non-linearity in the counters added at no cost allows us to rely on the counter state as a set of independent variables that cannot be easily predicted or approximated with a reasonably short LFSR.

To date construction of an NLFSR with a known, or a predictable period or calculation of the period of a given NLFSR remains an open problem and remains computationally infeasible for sufficiently large NLFSRs (80-bit or larger). As a solution, a number of easy to construct small (40-bit or smaller) non-bijective NLFSRs with guaranteed periods can be combined into a single set with a combined period calculated the same way as the period of any RNS-based counter. If one or some of the bits of the initial NLFSR counter state are initialised with a predetermined constant, every single starting value can be automatically placed inside the NLFSR loop thus restoring its bijective operation with a guaranteed fixed period of a sufficiently large length (desirably prime or having only large divisors). Thus a non-linear counter with a guaranteed arbitrarily long period can be easily constructed out of a small number of shorter NLFSRs at a very low cost.

The simplest and the most popular way to construct bijective (reversible) bitwise NLFSRs is by linearly combining each bit with a non-linear Boolean function of a number of previous bits in the stream stored in the NLFSR state. Such feedback of a single Boolean function into itself often results in exploitable patterns in the polynomial relationships between output bits at certain distances from each other. Such patterns are eliminated by allowing different carefully chosen Boolean feedback functions to be combined with every bit in the NLFSR state simultaneously, in other words by adding parallel feedback. In order for such NLPFSR to remain bijective, only one condition must be satisfied: only the previous bits in the stream stored in the NLPFSR state can be used as inputs into each non-linear Boolean function, except for the parts where bijectiveness is achieved by other means.

The extra non-linearity in the counters responsible for the guaranteed minimum period prevents the attacker from dismissing the counters as easily predictable, and the parallel feedback in a sufficiently wide NLPFSR accumulator allows the cipher to release output on every clock cycle thanks to the significantly increased diffusion rate.

VEST ciphers are conservatively engineered to address a pressing immediate need for secure ciphers and hashes in the high-volume commodity hardware market that can not be currently serviced using existing ciphers such as A5x [3], DST [4], E0 [5], HDCP [6], ORYX [7], VSC [8], [9] or MD5 [10], which are already seriously compromised.

Before describing the VEST cipher families, let us recall the famous Auguste Kerckhoffs' six desiderata for cryptographic systems presented in his 1883 work La Cryptographie Militare [1] to clarify the common misconception regarding what should and what should not be kept secret in ciphers. These six desiderata are often misquoted by cryptographers in form of a "security through obscurity" slur derived from a common misinterpretation of the second desideratum while ignoring the other five and without a deep understanding of the concept. To avoid any unnecessary arguments, these six desiderata are included below, slightly paraphrased (translated to the modern cryptographic terminology). They remain essential after over 120 years of collective cryptographic research:

1. A cipher should be unbreakable, at least in practice.
2. Compromise of one system should not affect [users of] other systems.
3. Keys should be as short as possible and re-keying should be as fast as possible.

4. Encrypted messages should be in the most conveniently aligned binary form.
5. The cipher should occupy as little area and as little memory as possible.
6. The cipher implementation and usage should not cause any mental strain.

Note: the word “system” in the second desideratum has a broader meaning than just “a device”. Besides physical devices and microchips, the word “system” should be understood as also including in its meaning software implementations, libraries, hybrid systems and communication protocols.

In addition, the second desideratum does not oppose keeping details of the exact wiring and Boolean logic used in a cryptographic device secret. It only implies that knowledge of those details by an attacker should not affect other devices and other implementations of the same cipher. In other words if such information is to be kept secret, it should be treated as a part of the secret key unique for each device.

The fourth desideratum has been translated to match the terminology of modern communications that have progressed from telegraph to high-speed bus and fibre optic links and from telegraph machines to network controllers and other microchips operating on binary messages.

The fifth desideratum has been translated to match the terminology of modern hardware and software cipher implementations.

The original word “operator” in the sixth desideratum should be understood as including both the user of the encryption product and the person implementing the cipher who can be seen as the “operator” of the cryptographic library. In other words, the cipher should be headache-free, for both the end-user and the product developer.

VEST ciphers are created addressing all of the above desiderata. Three family trees of VEST ciphers are introduced in this paper: 80-bit secure VEST-4 releasing up to four bits per clock cycle, 160-bit secure VEST-16 releasing 16 bits per clock cycle and 256-bit secure VEST-32 releasing 32 bits per clock cycle. These ciphers target three major applications: 1) minimum-area medium-security low-cost RFID encryption and authentication, 2) minimum-area high-security medium-cost microprocessor encryption, and 2) high-speed high-security bus channel encryption and authentication. Although other family trees of VEST ciphers with the same properties as proposed here can be tailored for each particular application by carefully adjusting the design variables by a highly skilled cryptologist, it should not be attempted by end-users or cryptographic developers.

1.1 Keyed Family Variants

The three root cipher families presented here are referred to as VEST-4, VEST-16 and VEST-32. Each of the three family trees of VEST ciphers supports family keying to generate other independent cipher families of the same size. The family-keying process provides a standard method to generate cipher families with unique substitutions and unique counters with different prime periods.

Family keying has several important cryptographic applications including support for unique transformations for protocols employing strict Fail-Stop policies and for supporting proprietary cipher implementations.

VEST ciphers have been designed so that each cipher family generated using a static family key can be efficiently synthesised in hardware. In some applications, the hardware structures supporting a VEST cipher may be reprogrammable, for example by using SRAM-based look-up tables for logic functions present in some FPGAs. In that case, the family key could be changed as desired generating arbitrary cipher families. In other applications, the hardware structures would be immutable, for example standard cells and wires in an ASIC. In that case the family key would be considered hardwired into the device and could not be changed. There are also intermediate cases such as via-programmable routing, which could be customized on a per-chip basis by direct e-beam write. Several manufacturers offer this kind of technology, including ST Microelectronics, Toshiba, UMC, and eASIC. In general, the hardwired approaches will have higher performance, lower area, and lower power than the reprogrammable ones.

On systems designed to communicate with more than one cipher family, the systems must store the family key and member key together with the unique device identifier. The family key may be a known public variable or considered part of the cipher's symmetric secret key. If the family key is considered a secret, it leaves the attacker with the only options to either gain access to both secret keys stored in other devices, to recover both keys by brute-force or to extract both keys out of the microchip by reverse engineering it. The security of the VEST ciphers is rated assuming that the family key is known to the attacker. The family-keying process, as described in section 3.1 and 3.2 ensures that the successful reverse engineering of one cipher family does not compromise cipher families generated with other family keys.

2. VEST Structure

VEST ciphers consist of four components: a counter, a counter diffusor, an accumulator and an output combiner. The RNS counter consists of sixteen NLFSRs with prime period lengths, the counter diffusor is a set of 5-to-1 linear combiners with feedback compressing outputs of the 16 counters into 10 bits, the core accumulator is an NLPFSR accepting 10 bits of the counter diffusor as input, and the output combiner is a set of 6-to-1 linear combiners. There are no known or intentional weaknesses, backdoors or shortcuts in VEST ciphers' structure, permutations or Boolean functions.

NLPFSR accumulators in VEST ciphers can be seen as individual bijective substitution-permutation networks constructed using non-linear 6-to-1 feedback functions, one for each bit, all updated simultaneously. Bijective NLPFSRs with parallel feedback benefit from significantly faster diffusion than their bitwise NLFSR and small S-box based or word-based counterparts. Properly constructed bijective NLPFSRs of up to 1296 bits in size can achieve complete diffusion in only four rounds with such small feedback functions.

Sizes of the input combiner and of the core accumulator are fixed for each VEST family tree.

The only exception in the accumulator state is a group of its least significant five bits updated by a bijective 5x5 S-Box and linearly combined with five input bits on each round. All other bits in the VEST accumulator state are updated by 6-to-1 feedback functions in which one of the bits of the accumulator state is linearly combined with a non-linear function of five other bits in the accumulator. In five of those feedback functions, one of the input bits is linearly combined with one of the bits of the accumulator state and with a non-linear function of four other bits in the accumulator. Appendixes A, B, C and D provide bit permutations and non-linear feedback functions for VEST-4, VEST-16 and VEST-32 cipher family trees.

The following parameters and variables are used in VEST cipher structure and operation:

B_i – sizes of RNS counters
C – combined size of all 16 RNS counters
c_i – 16 RNS counters
d – 10-bit linear counter diffusor
f – set of non-linear feedback functions for the core accumulator
g_i – set of non-linear feedback functions for the RNS counters
i – RNS counter index
j – bit index
k – input bits: key, IV or data
L – message length
M – cipher output width in bits
N – total IV length in bits
o – output bits
p – core accumulator bit permutation:
$p_{j0}..p_{j4}$ – indexes of 5 input bits to core accumulator feedback function f_j
p_{j5} – f_j output bit index
R_j – number of rounds in keying step j
Ri – number of rounds in cipher initialisation
r – round number
s – Boolean function index
t – permutation index
Va – 16 indexes of 10-bit RNS counters
Vb – 16 indexes of 11-bit RNS counters
Vf – 4096 accumulator feedback functions
Vp – 128 input bit permutations for accumulator feedback functions
W – width of the core accumulator in bits
w – accumulator feedback function index
x – core accumulator
xa, xb, xc, xd, xe, xf – indexes of accumulator output bits

2.1 RNS Counters

C-bit wide VEST cipher counter consists of sixteen separate 10-bit and 11-bit wide RNS counters *c* implemented as bijective NLFSRs, each updated with its own non-linear Boolean function with six inputs, chosen so that each of these smaller counters consists of two or four distinct loops with their lengths being unique primes. The period of each of the chosen NLFSRs is guaranteed to be equal to a predetermined prime number for any starting value. Such prime-period NLFSRs when combined together result in a counter with a total period being a multiple of the individual periods of all 16 chosen NLFSRs. In VEST-4 ciphers, 13 of the NLFSRs are 10-bit wide and 3 are 11-bit wide, forming a 163-bit counter state. In VEST-16 and in VEST-32 ciphers, 5 of the NLFSRs are 10-bit wide and 11 are 11-bit wide, forming a 171-bit counter state. For any given keyed combination of such NLFSRs, total period of the combined counter is between 2^{128} and 2^{153} for VEST-4 ciphers and between 2^{138} and 2^{162} for VEST-16 and VEST-32 ciphers.

The 16 RNS counters have B_i-bit (10-bit or 11-bit) states c_i updated by their respective non-linear feedback functions g_i. The input bit indexes for all the counters feedback functions are fixed at 0, 1, 2, 6 and 7. Each bit $B_{i\text{-}1}$ is also linearly combined with the output of each g_i and the result is stored in bit *0* while the remaining state is shifted by one bit to the left. The list of RNS counters feedback functions is provided in the appendix E, including all possible prime period lengths for each NLFSR. The 16 RNS counters feedback functions are chosen from that list depending on the family key. Counters with the longest shorter periods are chosen for all the root family ciphers.

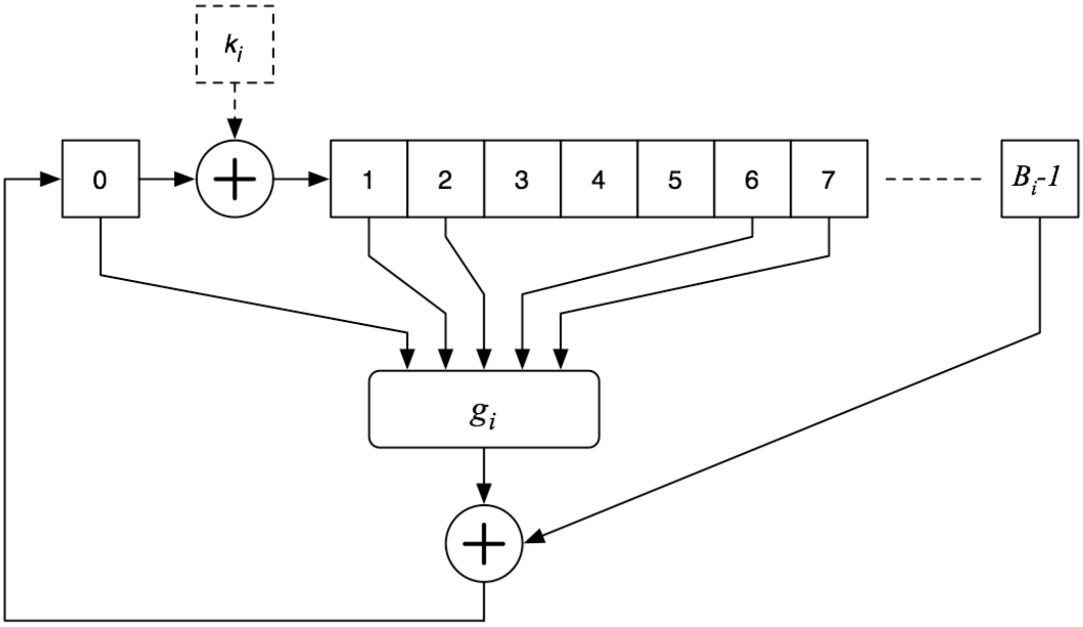

Fig 1. NLFSR counter

In keying mode, all sixteen counters accept sixteen key bits as their inputs linearly combined with bit *0* of each counter. In hashing mode, eight of the sixteen counters accept eight key bits as their inputs linearly combined with bit *0* of each counter, leaving eight of the counters out of control by the input stream.

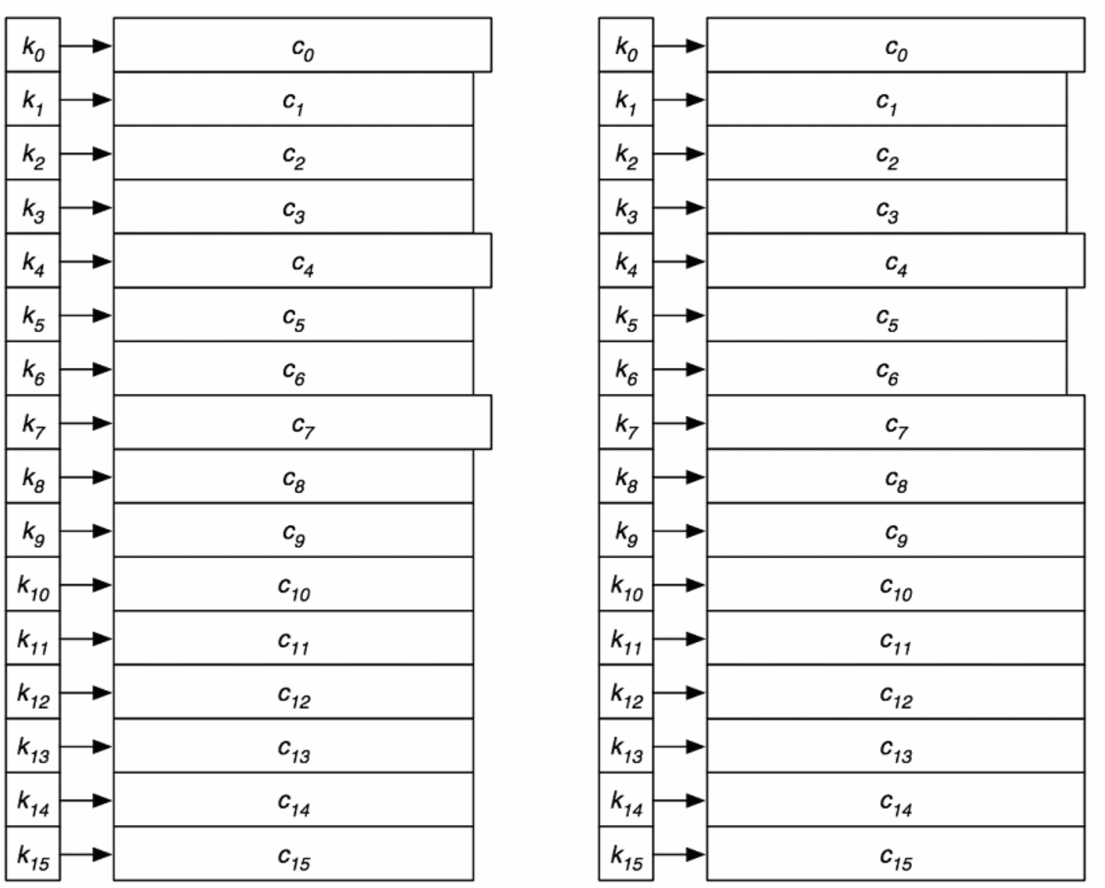

Fig 2. RNS counters (VEST-4 to the left, VEST-16 and VEST-32 to the right)

The following is a summary of the RNS counters operation in VEST ciphers, where *r* is the round number. The feedback functions *g* are family key dependent. The feedback functions *g* for the three root cipher families are provided in the appendixes A, B and C.

In keying mode:

$$c_{i\,0}^{r+1} = g_i\,(c_{i\,0}^{r}, c_{i\,1}^{r}, c_{i\,2}^{r}, c_{i\,6}^{r}, c_{i\,7}^{r}) + c_{i\,B[i]-1}^{r},$$
$$c_{i\,1}^{r+1} = c_{i\,0}^{r} + k_{i}^{r},$$
$$c_{i\,j}^{r+1} = c_{i\,j-1}^{r},\ 2 \le j < B_i,\ 0 \le i < 16;$$

In hashing mode:

$$c_{i\,0}^{r+1} = g_i\,(c_{i\,0}^{r}, c_{i\,1}^{r}, c_{i\,2}^{r}, c_{i\,6}^{r}, c_{i\,7}^{r}) + c_{i\,B[i]-1}^{r},$$
$$c_{i\,1}^{r+1} = c_{i\,0}^{r} + k_{i}^{r},$$
$$c_{i\,j}^{r+1} = c_{i\,j-1}^{r},\ 2 \le j < B_i,\ 0 \le i < 8;$$
$$c_{i\,0}^{r+1} = g_i\,(c_{i\,0}^{r}, c_{i\,1}^{r}, c_{i\,2}^{r}, c_{i\,6}^{r}, c_{i\,7}^{r}) + c_{i\,B[i]-1}^{r},$$
$$c_{i\,j}^{r+1} = c_{i\,j-1}^{r},\ 1 \le j < B_i,\ 8 \le i < 16;$$

In counter mode:

$$c_{i\,0}^{r+1} = g_i\,(c_{i\,0}^{r}, c_{i\,1}^{r}, c_{i\,2}^{r}, c_{i\,6}^{r}, c_{i\,7}^{r}) + c_{i\,B[i]-1}^{r},$$
$$c_{i\,j}^{r+1} = c_{i\,j-1}^{r},\ 1 \leq j < B_i,\ 0 \leq i < 16;$$

2.2 Counter Diffusor

Bits of different counters are combined together before entering the core accumulator as follows:

$$d_0^{r+1} = d_1^r + c_{1\,1}^r + c_{4\,1}^r + c_{5\,1}^r + c_{11\,1}^r + c_{13\,1}^r + 1;$$
$$d_1^{r+1} = d_2^r + c_{0\,1}^r + c_{2\,1}^r + c_{6\,1}^r + c_{8\,1}^r + c_{14\,1}^r;$$
$$d_2^{r+1} = d_3^r + c_{3\,1}^r + c_{4\,1}^r + c_{7\,1}^r + c_{10\,1}^r + c_{15\,1}^r;$$
$$d_3^{r+1} = d_4^r + c_{0\,1}^r + c_{3\,1}^r + c_{5\,1}^r + c_{9\,1}^r + c_{12\,1}^r;$$
$$d_4^{r+1} = d_5^r + c_{1\,1}^r + c_{4\,1}^r + c_{6\,1}^r + c_{12\,1}^r + c_{15\,1}^r + 1;$$
$$d_5^{r+1} = d_6^r + c_{0\,1}^r + c_{7\,1}^r + c_{9\,1}^r + c_{13\,1}^r + c_{14\,1}^r;$$
$$d_6^{r+1} = d_7^r + c_{1\,1}^r + c_{8\,1}^r + c_{11\,1}^r + c_{14\,1}^r + c_{15\,1}^r;$$
$$d_7^{r+1} = d_8^r + c_{2\,1}^r + c_{5\,1}^r + c_{6\,1}^r + c_{10\,1}^r + c_{12\,1}^r + 1;$$
$$d_8^{r+1} = d_0^r + c_{0\,1}^r + c_{3\,1}^r + c_{7\,1}^r + c_{8\,1}^r + c_{9\,1}^r + 1;$$
$$d_9^{r+1} = d_9^r + c_{8\,1}^r + c_{10\,1}^r + c_{12\,1}^r + c_{13\,1}^r + c_{15\,1}^r + 1;$$

2.3 Accumulator

In the accumulator state x, the least significant five bits x_0 to x_4 are used as inputs into five non-linear 5-to-1 Boolean functions f_0 to f_4 forming a 5x5 S-Box with its outputs linearly combined with five counter diffusor bits d_0 to d_4 and fed back into bits x_{p0} to x_{p4} respectively. Bits x_5 to x_9 are linearly combined with outputs of the next five non-linear 4-to-1 feedback functions f_5 to f_9 and with five counter diffusor bits d_5 to d_9 and fed back into bits x_{p5} to x_{p9} respectively. All other bits x_j in the VEST accumulator state are linearly combined with outputs of non-linear 5-to-1 Boolean functions f_j using bits $x_{nj[0]}$ to $x_{nj[4]}$ as inputs and fed back into bits x_{pj}.

To summarise the core accumulator operation:

$$x_{pj[5]}^{r+1} = f_j\,(x_0^r, x_1^r, x_2^r, x_3^r, x_4^r) + d_j^r,\ 0 \leq \mathrm{j} < 5;$$
$$x_{pj[5]}^{r+1} = f_j\,(x_{pj[0]}^r, x_{pj[1]}^r, x_{pj[2]}^r, x_{pj[3]}^r, x_{pj[4]}^r) + x_j^r + d_j^r,\ 5 \leq \mathrm{j} < 10;$$
$$x_{pj[5]}^{r+1} = f_j\,(x_{pj[0]}^r, x_{pj[1]}^r, x_{pj[2]}^r, x_{pj[3]}^r, x_{pj[4]}^r) + x_j^r,\ 10 \leq j < W;$$

Appendixes A, B, C and D provide the bit permutations p and the non-linear feedback functions f for VEST-4, VEST-16 and VEST-32 family tree ciphers.

In order for the above structure to be bijective, all feedback function indexes j must be greater than their corresponding input bit indexes pj_0, pj_1, pj_2, pj_3 and pj_4, and functions f_0, f_1, f_2, f_3 and f_4 must form a bijective (reversible) substitution operation. There are no restrictions on any of the other feedback functions f_j, which can be arbitrary, although a set of linearly independent strong balanced non-linear Boolean functions is carefully chosen for all VEST ciphers. The subsequent permutation of the accumulator bits also does not affect its bijectiveness. These relaxed conditions allow a heuristic search to find a permutation *p* that would result in a complete diffusion of a single bit change in the accumulator state as quickly as possible.

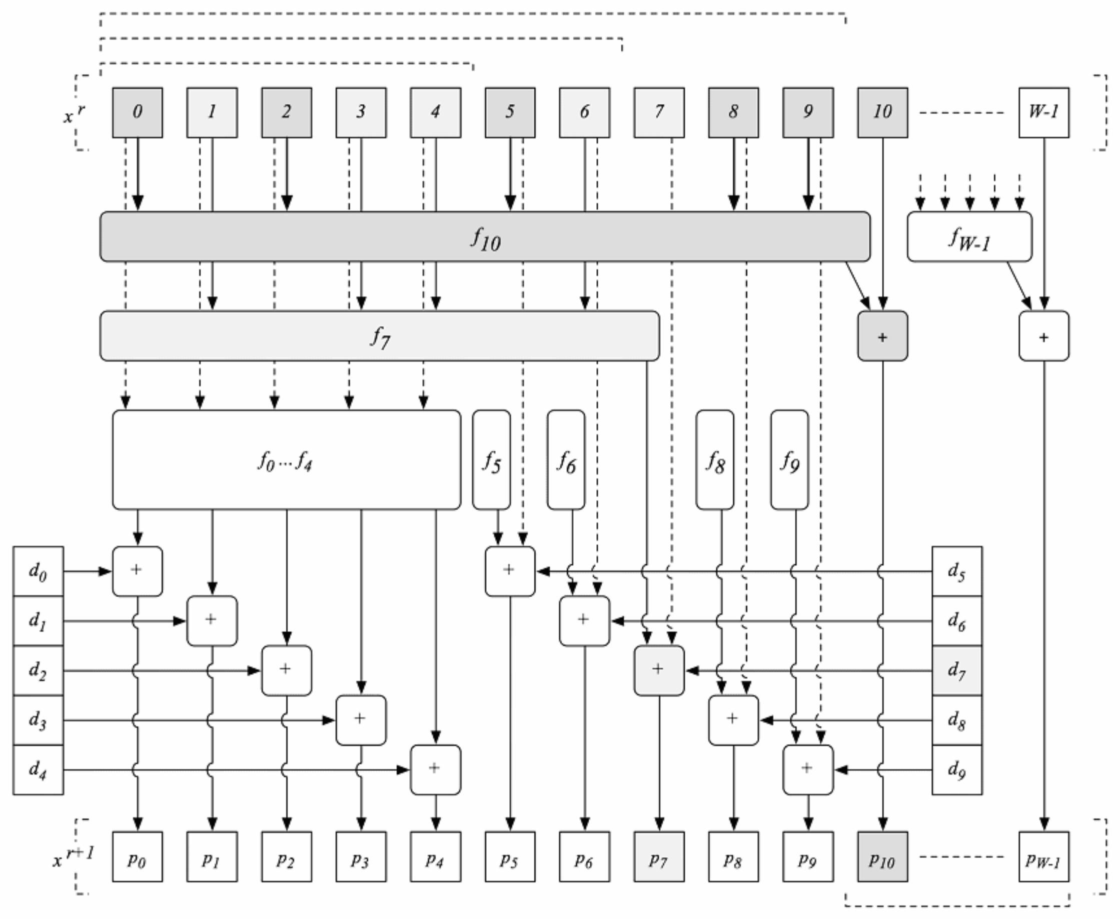

Fig 3. Partial illustration of the core accumulator

2.4 Output Combiner

Bits of the accumulator state are not released directly as output. For each of the *M* bits of output, six strongest bits of the accumulator state are linearly combined. To summarize the operation of the linear output combiner where bits o^{r*M} to $o^{(r+1)*M-1}$ are the *M* bits of output of round *r* after R_i rounds of cipher initialisation:

$$o^{r*M+j} = x^{Ri+r}{}_{xaj} + x^{Ri+r}{}_{xbj} + x^{Ri+r}{}_{xcj} + x^{Ri+r}{}_{xdj} + x^{Ri+r}{}_{xej} + x^{Ri+r}{}_{xfj},\ 0 \le j < M;$$
$$0 \le r < (L+M-1)/M.$$

The choice of all 6**M* input bit indexes *xa*, *xb*, *xc*, *xd*, *xe* and *xf* for all *M* output bits is fixed for the entire VEST cipher family tree of each size and is provided in appendix G.

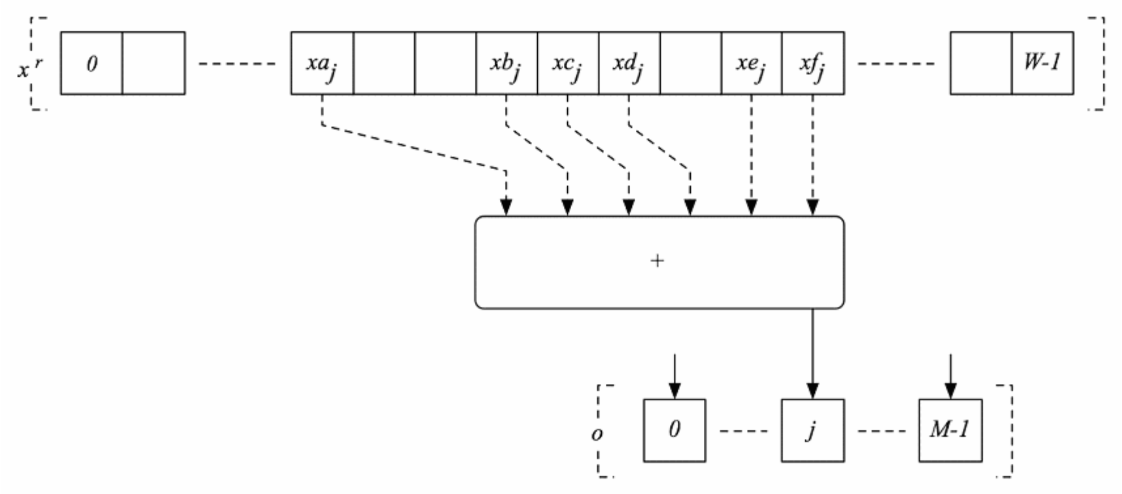

Fig 4. Output combiner

3. VEST Process

VEST cipher families are identified according to the width of the accumulator *W* and the width of the output *M*. This paper presents three VEST families of ciphers: a family with an 83-bit wide accumulator with a four-bit wide output, a family with a 331-bit wide accumulator with a 16-bit wide output, and a family with a 587-bit wide accumulator with a 32-bit wide output. As this is a preliminary ECRYPT submission, these preliminary variants are called ProVEST: ProVEST-4, ProVEST-16 and ProVEST-32. ProVEST ciphers are expected to resist all known attacks and only minor improvements are anticipated in the second round in response to the ongoing research and feedback of the cryptographic community. The improved second round variants recommended for wide spread use will be called VEST ciphers.

The included ProVEST-4, ProVEST-16 and ProVEST-32 ciphers are the root cipher families of their respective family trees. Root ciphers can be used on their own to generate keystream or to hash messages as default cipher families. Each of the root cipher families can also be used to generate a vast number of families of independent VEST ciphers. The three root VEST cipher families enable unique cipher families to be routinely generated in a standard interoperable manner if necessary. Implementations choosing to use a unique family key per cipher instance will increase the security of the cipher IF the cipher's family key is as securely protected as the cipher's secret member key used to initialise it. Evaluation of security of cipher families with family keys kept secret is outside the scope of this document.

If a more standardised cipher implementation is required efficient on the widest range of platforms, it is recommended to implement one of the root VEST cipher families directly as authenticated or unauthenticated stream ciphers or hash functions.

3.1 Cipher Family Generation

Besides the three fixed root families, the hardware structure (feedback functions) of each VEST cipher family is defined by the family key. Members of each family differ only in their symmetric member keys. The pseudo-random bits needed to generate the hardware structure for the new cipher family are produced by executing the root cipher family of the same size in counter mode. The root family cipher must first be initialised with the given family key as its member key and an optional IV. A number of first output bits are then used to choose the hardware structure elements for the new cipher family.

3.1.1) Initialisation

First, the root variant of the cipher family tree is initialised by filling the sixteen RNS counters, the core accumulator and the counter diffusor with zeroes:

$c_{i\,j}^{0} = 0,\ 0 \le i < 16,\ 0 \le j < B_i;$
$d^{0}_{j} = 0,\ 0 \le j < 10;$
$x^{0}_{j} = 0,\ 0 \le j < W.$

3.1.2) Loading the key

The cipher is executed in keying mode for $R_0 = F$ rounds, where F is the length of the family key in bits, loading the family key bits 16 bits per round beginning with the least significant bit, sliding the 16-bit window by one bit on each round:

$c_{i\,0}^{r+1} = g_i\,(c_{i\,0}^{r}, c_{i\,1}^{r}, c_{i\,2}^{r}, c_{i\,6}^{r}, c_{i\,7}^{r}) + c_{i\,B[i]-1}^{r},$
$c_{i\,1}^{r+1} = c_{i\,0}^{r} + k^{r}_{i},$
$c_{i\,j}^{r+1} = c_{i\,j-1}^{r},\ 2 \le j < B_i,\ 0 \le i < 16;$
$x^{r+1}_{pj[5]} = f_j\,(x^{r}_{0}, x^{r}_{1}, x^{r}_{2}, x^{r}_{3}, x^{r}_{4}) + d^{r}_{j},\ 0 \le j < 5;$
$x^{r+1}_{pj[5]} = f_j\,(x^{r}_{pj[0]}, x^{r}_{pj[1]}, x^{r}_{pj[2]}, x^{r}_{pj[3]}, x^{r}_{pj[4]}) + x^{r}_{j} + d^{r}_{j},\ 5 \le j < 10;$
$x^{r+1}_{pj[5]} = f_j\,(x^{r}_{pj[0]}, x^{r}_{pj[1]}, x^{r}_{pj[2]}, x^{r}_{pj[3]}, x^{r}_{pj[4]}) + x^{r}_{j},\ 10 \le j < W;$
$0 \le r < R_0.$

3.1.3) Hashing the key

The process switches to hashing mode for the next $R_1 = F/8$ rounds:

$c_{i\,0}^{r+1} = g_i\,(c_{i\,0}^{r}, c_{i\,1}^{r}, c_{i\,2}^{r}, c_{i\,6}^{r}, c_{i\,7}^{r}) + c_{i\,B[i]-1}^{r},$
$c_{i\,1}^{r+1} = c_{i\,0}^{r} + k^{r}_{i},$
$c_{i\,j}^{r+1} = c_{i\,j-1}^{r},\ 2 \le j < B_i,\ 0 \le i < 8;$
$c_{i\,0}^{r+1} = g_i\,(c_{i\,0}^{r}, c_{i\,1}^{r}, c_{i\,2}^{r}, c_{i\,6}^{r}, c_{i\,7}^{r}) + c_{i\,B[i]-1}^{r},$
$c_{i\,j}^{r+1} = c_{i\,j-1}^{r},\ 1 \le j < B_i,\ 8 \le i < 16;$
$x^{r+1}_{pj[5]} = f_j\,(x^{r}_{0}, x^{r}_{1}, x^{r}_{2}, x^{r}_{3}, x^{r}_{4}) + d^{r}_{j},\ 0 \le j < 5;$
$x^{r+1}_{pj[5]} = f_j\,(x^{r}_{pj[0]}, x^{r}_{pj[1]}, x^{r}_{pj[2]}, x^{r}_{pj[3]}, x^{r}_{pj[4]}) + x^{r}_{j} + d^{r}_{j},\ 5 \le j < 10;$
$x^{r+1}_{pj[5]} = f_j\,(x^{r}_{pj[0]}, x^{r}_{pj[1]}, x^{r}_{pj[2]}, x^{r}_{pj[3]}, x^{r}_{pj[4]}) + x^{r}_{j},\ 10 \le j < W;$
$R_0 \le r < R_0 + R_1.$

3.1.4) Finalising the hash

In hashing mode, each bit of input is used only once. To finalise the hashing process it's followed by one more round executed in hashing mode, hashing in 0x4E (0,1,1,1,0,0,1,0) as bits $k_F..k_{F+7}$ if the value being hashed is a key used for cipher family generation, hashing in 0x2B (1,1,0,1,0,1,0,0) as bits $k_F..k_{F+7}$ if the value being hashed is a key used for keystream generation, hashing in 0xB2 (0,1,0,0,1,1,0,1) as bits $k_F..k_{F+7}$ if the value being hashed is a key used for hashing, or hashing in 0xE4 (0,0,1,0,0,1,1,1) as bits $k_N..k_{N+7}$ if the value being hashed is an IV, in which case N = ∑ *width(*IV$_j$*)*, otherwise by hashing in 0xFF (1,1,1,1,1,1,1,1) as bits $k_L..k_{L+7}$ for all other data inputs. The input length in bits must also be a multiple of 8.

The process then switches to counter mode for R_2 = 31 more rounds. This step is called "sealing" the cipher:

$$
\begin{aligned}
&c_i{}^{r+1}{}_0 = g_i\,(c_i{}^r{}_0, c_i{}^r{}_1, c_i{}^r{}_2, c_i{}^r{}_6, c_i{}^r{}_7) + c_i{}^r{}_{B[i]-1},\\
&c_i{}^{r+1}{}_j = c_i{}^r{}_{j-1},\ 1 \le j < B_i,\ 0 \le i < 16;\\
&x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_0, x^r{}_1, x^r{}_2, x^r{}_3, x^r{}_4) + d^r{}_j,\ 0 \le \mathrm{j} < 5;\\
&x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j + d^r{}_j,\ 5 \le \mathrm{j} < 10;\\
&x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j,\ 10 \le j < W;\\
&R_0 + R_1 + 1 \le r < R_0 + R_1 + 1 + R_2.
\end{aligned}
$$

At this stage the entire internal state of the VEST cipher including the accumulator and all the counters can be saved as a "processed" (W+C+10) bit wide family key that can be stored and loaded into VEST registers for instant re-keying of the family tree cipher at any time.

Optionally, an IV or multiple IVs that can include a vendor ID, a product ID, its expiry date, etc., can be loaded into the cipher state:

3.1.5) Hashing the IVs

The cipher is executed in hashing mode for *width*(IV$_j$) rounds for each of the IVs as in step 3.1.3.

3.1.6) Finalising the hash

The cipher is "sealed" for the second time as in step 3.1.4.

From this point, i.e. after *Ri* cipher initialisation rounds, the process begins to return output keystream while executed continuously in counter mode releasing *M* bits of output on every round. The output keystream is used as follows:

3.1.7) Choosing 10-bit counters

For each i^{th} counter index i in the array of 16 available 10-bit counters Va specified in the appendix E, a 4-bit cipher output block (least significant bit first) is used as a 4-bit index s swapping Va_i with Va_s.

$s = \sum (o^{i*4+j} << j), 0 \leq j < 4;$
swap $(Va_i, Va_s);$
$0 \leq i < 16.$

3.1.8) Choosing 11-bit counters

For each i^{th} counter index i in the array of 16 available 11-bit counters Vb specified in the appendix E, a 4-bit cipher output block is used as a 4-bit index s swapping Vb_i with Vb_s.

$s = \sum (o^{64+i*4+j} << j), 0 \leq j < 4;$
swap $(Vb_i, Vb_s);$
$0 \leq i < 16.$

3.1.9) Choosing feedback functions

For each w^{th} Boolean function of the W required Boolean functions, a 10-bit cipher output block is selected as a 10-bit index s in the array of Boolean functions Vf provided in the appendix D swapping each Boolean function in the array identified by the index w with the function identified by the index s:

$s = \sum (o^{128+w*10+j} << j), 0 \leq j < 10;$
swap $(Vf_w, Vf_s);$
$0 \leq w < W.$

The first W elements of the rearranged array Vf become W accumulator feedback functions f for the family cipher being generated:

$\boldsymbol{f}_w = Vf_w, 0 \leq w < W.$

3.1.10) Permuting the inputs of feedback functions

For each of the W Boolean functions, from function *0* to function *W-1* in the array rearranged in step 3.1.8, a 7-bit output block is selected as a 7-bit input permutation index t. The five inputs to each accumulator feedback function for the family cipher being generated are rearranged in the order specified by t selecting one of the 128 input pin permutations listed in the appendix F:

$t = \sum (o^{128+W*10+w*7+j} << j), 0 \leq j < 7;$
$\boldsymbol{p}_w = Vp_t;$
$0 \leq w < W.$

Thus the family variant is generated using (W *17 + 128) bit of keystream output.

3.2 Family Member Initialisation

3.2.1) Initialisation

Each family cipher generated as described in 3.1 is first initialised as shown in 3.1.1.

3.2.2) Keying

Before switching to hashing mode or to counter mode, the family member must be initialised with a unique member key at runtime with a procedure identical to steps 3.1.2 to 3.1.4.

For devices storing fixed or rarely re-programmed long-term keys it is recommended that the above steps are executed off-line and the entire (W+C+10) bit initial secret state stored on the device for its instant loading into the cipher state at runtime. The entire secret internal state of VEST-4 ciphers is conveniently and conservatively 256 bits in size. The entire secret internal state of VEST-16 ciphers is conveniently and conservatively 512 bits in size. The entire secret internal state of VEST-32 ciphers is conveniently and conservatively 768 bits in size.

3.3 Encryption

3.3.1) Initialisation, keying and IV setup

The cipher is first initialised as shown in 3.2. Optional IVs may also be loaded as shown in steps 3.1.5 and 3.1.6. The cipher does not release any output for the first *Ri* rounds.

3.3.2) Keystream generation

To generate L bits of keystream or to encrypt an L-bit message, the cipher is executed continuously in counter mode for L/M rounds, where L should be a multiple of M:

$$
\begin{aligned}
&c_i{}^{r+1}{}_0 = g_i\,(c_i{}^r{}_0, c_i{}^r{}_1, c_i{}^r{}_2, c_i{}^r{}_6, c_i{}^r{}_7) + c_i{}^r{}_{B[i]-1},\\
&c_i{}^{r+1}{}_j = c_i{}^r{}_{j-1},\ 1 \le j < B_i,\ 0 \le i < 16;\\
&x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_0, x^r{}_1, x^r{}_2, x^r{}_3, x^r{}_4) + d^r{}_j,\ 0 \le \mathrm{j} < 5;\\
&x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j + d^r{}_j,\ 5 \le \mathrm{j} < 10;\\
&x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j,\ 10 \le j < W;\\
&o^{(r-Ri)*M+j} = x^r{}_{xaj} + x^r{}_{xbj} + x^r{}_{xcj} + x^r{}_{xdj} + x^r{}_{xej} + x^r{}_{xfj},\ 0 \le j < M;\\
&Ri \le r < Ri + L/M.
\end{aligned}
$$

The cipher output bits o are applied to the message according to the protocol in place, usually using an XOR operation. Careful attention should be paid to the electrical isolation of the cipher signals to minimise leakage of information to the adjacent wires or circuits. In some cases, especially in low-power applications, it maybe necessary to balance the power consumption of the cipher very carefully, so that information about the key cannot be extracted by analysis of the power consumption of the chip.

3.4 Hashing

3.4.1) Initialisation

The cipher is first initialised as shown in 3.2. Optional IVs may also be loaded as shown in steps 3.1.5 and 3.1.6. The cipher does not release any output for the first *Ri* rounds.

3.4.2) Message processing

The cipher is executed in hashing mode for *L*/8 rounds, where *L* is the message length in bits (which must be a multiple of 8):

$$\begin{array}{l}
c_{i\;0}^{r+1} = g_i\,(c_{i\;0}^{r}, c_{i\;1}^{r}, c_{i\;2}^{r}, c_{i\;6}^{r}, c_{i\;7}^{r}) + c_{i\;B[i]-1}^{r}, \\
c_{i\;1}^{r+1} = c_{i\;0}^{r} + k^{r}_{i}, \\
c_{i\;j}^{r+1} = c_{i\;j-1}^{r},\ 2 \le j < B_i,\ 0 \le i < 8; \\
c_{i\;0}^{r+1} = g_i\,(c_{i\;0}^{r}, c_{i\;1}^{r}, c_{i\;2}^{r}, c_{i\;6}^{r}, c_{i\;7}^{r}) + c_{i\;B[i]-1}^{r}, \\
c_{i\;j}^{r+1} = c_{i\;j-1}^{r},\ 1 \le j < B_i,\ 8 \le i < 16; \\
x^{r+1}_{pj[5]} = f_j\,(x^{r}_{0}, x^{r}_{1}, x^{r}_{2}, x^{r}_{3}, x^{r}_{4}) + d^{r}_{j},\ 0 \le \mathrm{j} < 5; \\
x^{r+1}_{pj[5]} = f_j\,(x^{r}_{pj[0]}, x^{r}_{pj[1]}, x^{r}_{pj[2]}, x^{r}_{pj[3]}, x^{r}_{pj[4]}) + x^{r}_{j} + d^{r}_{j},\ 5 \le \mathrm{j} < 10; \\
x^{r+1}_{pj[5]} = f_j\,(x^{r}_{pj[0]}, x^{r}_{pj[1]}, x^{r}_{pj[2]}, x^{r}_{pj[3]}, x^{r}_{pj[4]}) + x^{r}_{j},\ 10 \le j < W; \\
Ri \le r < Ri + L/8.
\end{array}$$

The hashing protocol may also include hashing in the total message length and other parameters such as current date, time, etc.

3.4.3) Finalising the hash

One more round is executed in hashing mode, hashing in all ones as bits $k_L..k_{L+7}$ as described in 3.1.4:

$$\begin{array}{l}
c_{i\;0}^{r+1} = g_i\,(c_{i\;0}^{r}, c_{i\;1}^{r}, c_{i\;2}^{r}, c_{i\;6}^{r}, c_{i\;7}^{r}) + c_{i\;B[i]-1}^{r}, \\
c_{i\;1}^{r+1} = c_{i\;0}^{r} + 1, \\
c_{i\;j}^{r+1} = c_{i\;j-1}^{r},\ 2 \le j < B_i,\ 0 \le i < 8; \\
c_{i\;0}^{r+1} = g_i\,(c_{i\;0}^{r}, c_{i\;1}^{r}, c_{i\;2}^{r}, c_{i\;6}^{r}, c_{i\;7}^{r}) + c_{i\;B[i]-1}^{r}, \\
c_{i\;j}^{r+1} = c_{i\;j-1}^{r},\ 1 \le j < B_i,\ 8 \le i < 16; \\
x^{r+1}_{pj[5]} = f_j\,(x^{r}_{0}, x^{r}_{1}, x^{r}_{2}, x^{r}_{3}, x^{r}_{4}) + d^{r}_{j},\ 0 \le \mathrm{j} < 5; \\
x^{r+1}_{pj[5]} = f_j\,(x^{r}_{pj[0]}, x^{r}_{pj[1]}, x^{r}_{pj[2]}, x^{r}_{pj[3]}, x^{r}_{pj[4]}) + x^{r}_{j} + d^{r}_{j},\ 5 \le \mathrm{j} < 10; \\
x^{r+1}_{pj[5]} = f_j\,(x^{r}_{pj[0]}, x^{r}_{pj[1]}, x^{r}_{pj[2]}, x^{r}_{pj[3]}, x^{r}_{pj[4]}) + x^{r}_{j},\ 10 \le j < W; \\
Ri + L/8 \le r < Ri + L/8 + 1.
\end{array}$$

The cipher switches to counter mode executing 31 sealing rounds:

$$\begin{array}{l}
c_{i\;0}^{r+1} = g_i\,(c_{i\;0}^{r}, c_{i\;1}^{r}, c_{i\;2}^{r}, c_{i\;6}^{r}, c_{i\;7}^{r}) + c_{i\;B[i]-1}^{r}, \\
c_{i\;j}^{r+1} = c_{i\;j-1}^{r},\ 1 \le j < B_i,\ 0 \le i < 16; \\
x^{r+1}_{pj[5]} = f_j\,(x^{r}_{0}, x^{r}_{1}, x^{r}_{2}, x^{r}_{3}, x^{r}_{4}) + d^{r}_{j},\ 0 \le \mathrm{j} < 5; \\
x^{r+1}_{pj[5]} = f_j\,(x^{r}_{pj[0]}, x^{r}_{pj[1]}, x^{r}_{pj[2]}, x^{r}_{pj[3]}, x^{r}_{pj[4]}) + x^{r}_{j} + d^{r}_{j},\ 5 \le \mathrm{j} < 10; \\
x^{r+1}_{pj[5]} = f_j\,(x^{r}_{pj[0]}, x^{r}_{pj[1]}, x^{r}_{pj[2]}, x^{r}_{pj[3]}, x^{r}_{pj[4]}) + x^{r}_{j},\ 10 \le j < W; \\
R_i + L/8 + 1 \le r < R_i + L/8 + 32.
\end{array}$$

3.4.4) Returning the hash value

The subsequent *H*/*M* rounds of the cipher executed in counter mode return a *H*-bit wide (least significant bit first) hash of the message:

$$
\begin{aligned}
&c_i{}^{r+1}{}_0 = g_i\,(c_i{}^r{}_0, c_i{}^r{}_1, c_i{}^r{}_2, c_i{}^r{}_6, c_i{}^r{}_7) + c_i{}^r{}_{B[i]-1},\\
&c_i{}^{r+1}{}_j = c_i{}^r{}_{j-1},\ 1 \le j < B_i,\ 0 \le i < 16;\\
&x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_0, x^r{}_1, x^r{}_2, x^r{}_3, x^r{}_4) + d^r{}_j,\ 0 \le \mathrm{j} < 5;\\
&x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j + d^r{}_j,\ 5 \le \mathrm{j} < 10;\\
&x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j,\ 10 \le j < W;\\
&o^{(r-Ri-L/8-32)*M+j} = x^r{}_{xaj} + x^r{}_{xbj} + x^r{}_{xcj} + x^r{}_{xdj} + x^r{}_{xej} + x^r{}_{xfj},\ 0 \le j < M;\\
&R_i + L/8 + 32 \le r < R_i + L/8 + 32 + H/M.
\end{aligned}
$$

Note: The hash length *H* must be a multiple of *M*, but the resulting hash value can be truncated if necessary.

4. VEST Design Principles

VEST ciphers are designed targeting exclusively FPGA and ASIC platforms with a goal of achieving the highest possible performance while occupying minimal area at the expense of software performance. The three proposed sizes target three major applications: high volume low cost, medium volume medium cost and high-speed high security products. VEST-4 design is optimized to occupy the smallest possible area, VEST-32 design is optimised for the best possible area/performance ratio and VEST-16 design is optimised to achieve balance between security and cost.

VEST-4 family tree is designed to achieve at least 80-bit security while occupying an area of ~4K gates in ASIC or ~512 logic cells and ~256 register cells in the Altera Stratix-II FPGA.

VEST-16 family tree is designed to achieve at least 160-bit security while occupying an area of ~12K gates in ASIC or ~1K logic cells and ~512 register cells in the Altera Stratix-II FPGA.

VEST-32 family tree is designed to achieve at least 256-bit security while returning 32 bits of output per clock cycle and occupying an area of ~20K gates in ASIC or ~2K logic cells and ~1K register cells in the Altera Stratix-II FPGA.

Boolean functions used in VEST accumulators and input combiners are selected satisfying a number of cryptographic criteria which can be summarised as a requirement for each Boolean function to achieve the most balanced distribution of monomials of each algebraic degree in their AND-XOR, AND-OR and other algebraic forms, also applicable to their linear combinations. While Boolean functions used in VEST ciphers may not demonstrate the highest possible non-linearity or the highest possible algebraic degree, a balanced compromise with other important cryptographic properties is achieved ensuring VEST ciphers balanced resistance to both known and unknown attacks. Choice of Boolean functions for VEST ciphers was limited by high performance requirements on most FPGA platforms. Most FPGA architectures only implement arbitrary 4-to-1

Boolean functions efficiently. Some FPGA architectures like Altera Stratix II implement efficiently arbitrary 6-to-1 Boolean functions. Wider Boolean functions would significantly increase the cipher's size and significantly reduce its performance on most FPGA platforms reducing the potential for the cipher's [hardware] cross-platform interoperability. Smaller Boolean functions significantly increase probability of exploitable linearity, correlation and other undesirable properties, dramatically reducing cipher's security.

Permutations in VEST accumulators and input combiners are selected at random and heuristically refined to minimise redundancy in polynomial relationships between different bits of the accumulator while maximising the number of variables introduced into those polynomial relationships with each round ensuring the highest possible diffusion rate.

The key components of VEST ciphers have prime sizes and prime periods to avoid exploitable patterns in their combinations and to thwart decimation attacks. The RNS counters are chosen to be of practical size, small enough to allow for parallel 8-bit or 16-bit inputs into the counter state intended to guarantee a minimum period of at least 2^{128}.

The entire secret internal state of VEST-4 ciphers is chosen to be conservatively large and conveniently sized to 256 bits. The entire secret internal state of VEST-16 ciphers is chosen to be conservatively large and conveniently sized to 512 bits. The entire secret internal state of VEST-32 ciphers is chosen to be conservatively large and conveniently sized to 768 bits. The core accumulators of VEST-16 and VEST-32 ciphers are also chosen to be conservatively large in proportion to the size of the output (over 18 to 1).

5. VEST Security

VEST-32 cipher family tree is designed to offer 256-bit or higher long-term security. VEST-16 cipher family tree is designed to offer 160-bit or higher medium-term security. VEST-4 cipher family tree is designed to offer short-term 80-bit security. Keeping family keys secret also offers increased protection against power analysis and other side channel attacks.

There are no known attacks against VEST ciphers faster than the brute-force of the key space or of the internal state.

5.1 Randomness Tests

Each component of VEST ciphers has been thoroughly tested with the best existing randomness tests. Individual streams of any of the VEST accumulators outputs, combined VEST counters outputs, and outputs of complete VEST ciphers are unbiased and are indistinguishable from random.

5.2 Algebraic Structure Defectoscopy Tests

Tests of the algebraic structure of VEST ciphers by a proprietary set of ASD tools show that any controlled change in the accumulator state results in the distribution of monomials in the polynomial relationships between the output bits in any algebraic form being indistinguishable from random (achieving complete diffusion) after four rounds.

For comparison, four rounds of Rijndael are required to achieve complete diffusion of a controlled change in its 128-bit block, and LFSR-based ciphers like LILI-128 never pass ASD tests after any number of rounds.

5.3 Period

VEST ciphers are assisted by a non-linear counter with a very long period. The core accumulator has an average period of $2^{W/2}$. Combined with the period of the counter, the average period length of VEST-4 ciphers executed in counter mode is over 2^{182}, the average period length of VEST-16 ciphers is over 2^{315}, and the average period length of VEST-32 ciphers is over 2^{443}. When executed in counter mode, VEST-4 ciphers have a guaranteed minimum period of no less than 2^{128} and VEST-16 and VEST-32 ciphers have a guaranteed minimum period of no less than 2^{138}.

5.4 Weak Keys, Related Keys and IV attacks

There are no fixed points and no collisions in any of the VEST ciphers' components. Key material is loaded into the internal state sequentially through the bijective input combiner, then expanded and thoroughly mixed in the core accumulator with all other key and IV bits. Four rounds are required for a complete diffusion of every single bit loaded into the core accumulator, and the 32-round key-sealing and IV-sealing steps make it very hard to determine what changes in the key or IV material could result in a predetermined change in the internal state after the keying process.

5.5 Algebraic Attacks

Algebraic attacks are most effective against ciphers with linear or quadratic components and against ciphers with easily reducible polynomials defining relationships between output bits and bits of internal state or key material. All key components in VEST ciphers are non-linear and cannot be dismissed from the attacks. All the feedback functions in the core accumulator and in the input combiner are dense degree 4-5 polynomials. In four rounds, every bit in the 587-bit accumulator state will depend on all other accumulator bits, as well as bits of the counters and input combiner – a very large number of variables, which cannot be significantly reduced by knowledge of output bits. Even assuming that the counter state is guessed by the attacker, the core accumulators of all VEST ciphers are conservatively chosen to be at least 18 times the width of the output. In the 18 rounds required to define the internal state of the accumulator, these relationships grow sufficiently large to render algebraic attacks infeasible.

5.6 Time-Memory-Data Trade-off Attacks

Internal states of all VEST ciphers families are conservatively chosen to be at least 3 times the size of their expected security in bits: 80-bit secure VEST-4 ciphers have a 256-bit internal state, 160-bit secure VEST-16 ciphers have a 512-bit internal state, and 256-bit secure VEST-32 ciphers have a 768-bit internal state. Such proportion is chosen to make TMD attacks including possible shortcuts require more resources than brute-force of the cipher keyspace.

5.7 Guess-and-Determine Attacks

To calculate the next 32 bits of keystream of a VEST-32 cipher, the attacker must guess over 128 bits of the accumulator. To calculate the next 16 bit of keystream of a VEST-16 cipher, the attacker must guess over 64 bit of the accumulator. Values of each of those bits depend on six other bits of the accumulator, with at most three input bits in common between any pair of bits. The rapid increase in the number of variables in inter-bit relationships between rounds does not allow the attacker to reduce the number of unknowns below the exhaustive search after any number of rounds.

5.8 Linear Attacks

No exploitable linear approximations were found for VEST feedback functions.

5.9 Distinguishing Attacks

Since VEST ciphers release only a small portion of the bijective non-linear accumulator, they are also not susceptible to distinguishing attacks up to the period length of the counter, which is guaranteed to be no less than 2^{128}.

6. VEST Performance

A number of flexible parameters influence performance of the cipher. These variables include:

- Widths of the Boolean functions
- Minimum guaranteed period length
- Width of the input
- Number of bits of output per clock cycle
- Selection of wiring potentially unique in each cipher instance
- Maximum allowed distance between feedback functions inputs and outputs
- The hardware architecture the cipher is implemented in

All VEST ciphers components are designed with needs of FPGA and ASIC designers in mind, considering the cipher hardware efficiency to be the second top priority after its security.

VEST ciphers can be re-keyed instantly by filling the entire internal state with a slightly larger ‘processed’ secret key, or with a keying procedure described in 3.2, in which case initialisation of a VEST-4 cipher with an 80-bit key without an IV takes 122 rounds, initialisation of a VEST-16 cipher with a 160-bit key without an IV takes 212 rounds, and initialisation of a VEST-32 cipher with a 256-bit key without an IV takes 320 rounds.

Loading a 64-bit IV into a pre-keyed cipher state takes 40 rounds. Loading a 128-bit IV into a pre-keyed cipher state takes 48 rounds.

In counter mode, VEST-4 ciphers release four bits of output per clock cycle, VEST-16 ciphers release 16 bits of output per clock cycle, and VEST-32 ciphers release 32 bits of output per clock cycle.

In hashing mode, all VEST ciphers hash the message in one pass accepting eight bit of the message per clock cycle and returning the hash value in *H*/4, *H*/16 or *H*/32 clock cycles after the final "sealing" of the cipher state.

VEST-4 ciphers are recommended for generation of at most 80-bit secure 160-bit hashes. VEST-16 ciphers are recommended for generation of at most 160-bit secure 320-bit hashes. VEST-32 ciphers are recommended for generation of at most 256-bit secure 512-bit hashes.

Estimated FPGA and ASIC performance figures are provided below. Of course, the exact area and performance figures depend on the implementation requirements and on the additional functionality of the cipher module such as optional support for fast packet keying, area minimisation, etc., the target hardware architecture mapping, placement and routing as part of a larger circuit, the speed grading of the chip and a myriad of other variables. Variations in cipher module functionality significantly affect area figures and some complex multi-function modules cannot be efficiently implemented in some FPGA architectures due to pre-built registers macros, thus affecting performance figures. Exact performance figures for a wide range of cipher module implementations on a number of hardware architectures are outside the scope of this document.

6.1 Overview of FPGA Performance on Altera Stratix II

VEST cipher design is well optimised for 6-to-1 lookup table architectures. Generally in FPGA implementations 6-to-1 Boolean functions synthesise as three 4-to-1 Boolean functions.

In Altera Stratix-II architecture with a top speed of 500 MHz, synthesised circuit of the core accumulator of a VEST-32 cipher using 6-to-1 Boolean functions requires 587 6-to-1 ALUT cells and achieves clock speeds of 430+MHz resulting in processing speed of over 13 Gigabit per second. The interconnect delay on the slowest pin of the accumulator is approximately 80% of the total register-to-register delay.

The same circuit synthesised on Altera Stratix architecture with a top speed of 422 MHz requires 1730 4-to-1 Logic Elements and achieves clock speeds of 230+MHz resulting in processing speed of over 7 Gigabit per second. The interconnect delay on the slowest pin of the accumulator is approximately 85% of the total register-to-register delay.

In Altera Stratix-II architecture with a top speed of 500 MHz, synthesised circuit of the core accumulator of a VEST-4 cipher using 6-to-1 Boolean functions requires 83 6-to-1 ALUT cells. The register-to-register delay on the slowest path is less than 1.396 ns and the register delay is ~176ps, A total combined delay of ~1.572 ns corresponds to ~630 MHz clock speed limited to 500 MHz on the device in practice.

6.2 Heuristic Estimates of ASIC Performance

The static timing performance of VEST ciphers in ASIC remains an open question. The optimal choice of Boolean functions is also currently an open question. The general rule of thumb is that ASIC designs are approximately 4 times faster than FPGA designs. According to these estimates, ASIC implementations of VEST-4 ciphers are expected to operate at speeds of ~10 Gigabit per second, and VEST-32 ciphers at ~52 Gigabit per second.

All VEST-4 ciphers are expected to occupy ~4K gates in ASIC. All VEST-16 ciphers are expected to occupy ~12K gates in ASIC. All VEST-32 ciphers are expected to occupy ~20K gates in ASIC.

6.3 Software

VEST ciphers are relatively slow on software platforms when compared with modern general-purpose and software-dedicated ciphers. Various optimisation techniques such as bit slicing may improve software performance, but competitive software performance is not expected of VEST ciphers and therefore only a reference software implementation is provided. VEST ciphers are designed for dedicated hardware applications that do not require exceptionally high software speeds.

7. Conclusion

This paper presents VEST families of hardware-dedicated stream ciphers based on a number of innovative techniques. Designed with hardware efficiency in mind, VEST ciphers achieve high speed and high security occupying the smallest possible area and requiring minimum power while generally disregarding software performance.

As a new construction, VEST ciphers should be thoroughly studied and attacked, and potentially tweaked before they can be widely accepted as cryptographically secure primitives. All critical feedback of the cryptographic community is welcome. Further research should also be conducted to review the relative merits of per-cipher family-keying in respect to their security and hardware efficiency.

8. Authors Contributions and Acknowledgments

Sean O’Neil is the principal author of this paper and cipher designer. Howard A Landman has contributed on FPGA and ASIC issues related to the cipher design and significantly improved overall clarity of the paper. Benjamin Gittins has contributed the drawings, the FPGA and ASIC performance figures and his testing and feedback during the VEST cipher development and debugging stages resulted in a number of corrections to the cipher and its implementation. The authors would also like to thank the investors, the patenting and management teams for their ongoing support and contribution over the last 24 months.

References

[1] Auguste Kerckhoffs, La Cryptographie Militare, 1883.

[2] Oliver Kömmerling, Markus G. Kuhn, Design Principles for Tamper-Resistant Processors, 1999.

[3] E. Barkan, E. Biham, N. Keller, “Instant Ciphertext-only Cryptanalysis of GSM Encrypted Communication”, Crypto 2003.

[4] S. Bono, M. Green, A. Stubblefield, A. Rubin, “Analysis of the Texas Instruments DST RFID”.

[5] F. Armknecht, “A Linearization Attack on the Bluetooth Key Stream Generator”, Eprint 2002.

[6] N. Ferguson, “Censorship in action: why i don't publish my HDCP results”, http://www.macfergus.com/niels/dmca/cia.html

[7] D. Wagner, L. Simpson, E. Dawson, J. Kelsey, W. Millan, B. Schneier, “Cryptanalysis of ORYX”, SAC 98.

[8] Y. Tsunoo et at, “Distinguishing Attack with Chosen Initialisation Vector Against VSC128”, ECRYPT - The State of the Art of Stream ciphers, October 2004.

[9] S. O'Neil, “Vector Stream Cipher instant key recovery”, Synaptic Labs, September 2004.

[10] X. Wang, D. Feng, X. Lai, H. Yu, “Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD”, August 2004.

Appendix A

VEST-4 core accumulator bit permutation

j	$p_{j[0]}$	$p_{j[1]}$	$p_{j[2]}$	$p_{j[3]}$	$p_{j[4]}$	$p_{j[5]}$
0	0	1	2	3	4	52
1	0	1	2	3	4	12
2	0	1	2	3	4	65
3	0	1	2	3	4	77
4	0	1	2	3	4	55
5	4	4	1	3	2	80
6	5	5	0	4	3	70
7	1	1	3	2	6	54
8	2	2	7	0	6	49
9	3	3	7	2	6	21
10	5	6	2	0	3	37
11	4	5	1	7	8	20
12	9	8	6	5	7	42
13	10	5	6	4	7	22
14	8	12	7	5	10	73
15	7	11	14	10	5	74
16	13	4	8	3	1	31
17	5	13	9	16	15	58
18	8	17	6	7	9	82
19	18	5	14	8	11	13
20	2	16	18	9	3	11
21	15	3	13	10	16	36
22	7	18	15	9	5	0
23	22	5	19	11	14	38
24	18	15	17	5	20	32
25	24	21	18	14	8	50
26	9	16	17	15	22	71
27	25	13	21	15	10	45
28	27	22	25	15	11	79
29	16	19	22	1	23	43
30	26	18	24	16	20	61
31	10	30	15	0	14	24
32	31	27	25	28	15	23
33	20	17	26	24	16	6
34	18	24	21	25	27	53
35	30	24	20	26	18	29
36	35	34	29	25	19	7
37	29	5	31	28	19	62
38	19	18	36	35	6	75
39	26	33	5	17	37	14
40	39	20	37	29	23	68

41	40	30	31	33	32	60
42	19	34	28	36	25	30
43	33	39	42	32	26	69
44	24	25	4	28	42	59
45	44	41	38	12	9	17
46	36	42	39	40	33	27
47	10	43	46	32	30	2
48	32	43	41	37	39	35
49	48	47	45	43	32	28
50	37	46	40	43	47	64
51	48	27	31	37	28	9
52	38	51	26	35	41	81
53	35	52	46	42	33	16
54	27	24	48	17	25	72
55	23	51	48	44	38	10
56	41	40	49	45	39	3
57	56	39	29	36	42	67
58	50	54	40	57	56	19
59	33	36	18	40	39	44
60	21	47	59	40	13	26
61	28	57	27	58	54	4
62	61	55	56	45	48	76
63	54	62	56	61	46	41
64	1	49	5	38	59	34
65	64	53	51	58	48	39
66	53	22	44	41	11	57
67	60	63	34	66	49	78
68	56	52	50	57	63	33
69	58	50	57	65	68	47
70	67	64	61	57	63	15
71	70	34	67	60	66	48
72	65	48	13	51	69	63
73	72	69	66	62	55	18
74	17	69	52	51	48	1
75	59	71	68	64	63	8
76	66	60	73	71	59	46
77	76	74	53	51	60	25
78	75	74	72	12	76	40
79	76	75	58	68	77	66
80	75	68	79	73	78	51
81	68	79	73	47	77	56
82	81	76	78	79	80	5

ProVEST-4 root family

RNS counters indexes:

0, 16, 17, 18, 1, 19, 20, 2, 21, 22, 23, 24, 25, 26, 27, 28

Feedback function indexes:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77

Input bit permutation indexes:

0, 86, 69, 83, 84, 45, 52, 32, 67,105,112,104,101,114, 32, 70, 97,109,105,108, 121, 32, 84,114,101,101, 10, 68,101,115,105,103,110,101,100, 32, 98,121, 32, 83, 101, 97,110, 32, 79, 39, 78,101,105,108, 10, 67, 66, 32, 67, 97,112,105,116, 97, 108, 32, 77, 97,110, 97,103,101,109,101,110,116, 32, 83, 46, 65, 46, 0

Appendix B

VEST-16 core accumulator bit permutation

j	$p_{j[0]}$	$p_{j[1]}$	$p_{j[2]}$	$p_{j[3]}$	$p_{j[4]}$	$p_{j[5]}$
0	0	1	2	3	4	326
1	0	1	2	3	4	34
2	0	1	2	3	4	122
3	0	1	2	3	4	209
4	0	1	2	3	4	180
5	4	4	1	3	2	266
6	5	5	2	1	3	178
7	1	1	3	5	4	189
8	7	7	4	1	0	85
9	8	8	5	2	6	118
10	2	6	3	4	5	223
11	10	8	6	5	9	243
12	11	8	5	10	9	200
13	5	9	6	2	3	86
14	1	10	7	2	4	301
15	1	11	14	4	8	117
16	1	0	3	4	7	143
17	16	13	7	6	0	233
18	17	13	9	5	6	123
19	17	13	12	3	7	114
20	19	17	13	12	16	296
21	4	3	11	10	14	39
22	21	18	15	11	9	265
23	20	19	13	12	15	163
24	15	20	9	13	23	218
25	15	9	18	14	11	293
26	20	22	19	15	9	92
27	8	11	12	25	10	64
28	15	8	12	16	11	227
29	26	20	28	18	12	175
30	26	20	29	19	24	283
31	23	25	24	10	14	245
32	12	28	25	21	29	204
33	32	19	26	22	20	201
34	26	14	19	21	30	231
35	12	20	13	17	18	244
36	33	11	31	25	14	267
37	36	33	30	26	20	10
38	24	37	25	31	27	309
39	6	0	16	13	8	255

40	25	31	14	28	10	330
41	40	31	37	30	24	18
42	34	27	38	31	25	146
43	42	39	36	32	26	54
44	43	26	22	11	27	186
45	42	41	35	20	37	100
46	45	42	36	39	30	202
47	46	43	45	42	30	104
48	40	11	25	10	31	167
49	47	43	42	32	10	166
50	47	46	42	17	34	11
51	50	47	44	43	22	134
52	51	36	45	41	35	211
53	50	30	46	42	43	217
54	53	50	44	28	51	19
55	23	53	43	52	29	297
56	55	52	32	45	39	31
57	42	43	19	56	26	298
58	57	16	51	13	12	37
59	58	53	55	48	54	232
60	17	59	47	49	43	274
61	29	28	32	23	5	311
62	54	58	8	6	9	289
63	49	45	43	47	36	188
64	50	51	44	53	47	230
65	45	57	58	54	48	147
66	20	38	24	55	23	216
67	65	63	45	47	49	197
68	44	64	65	31	49	312
69	68	65	62	49	45	119
70	2	65	68	69	22	170
71	61	40	31	60	62	16
72	58	70	69	62	65	288
73	58	61	35	51	56	220
74	73	51	61	50	57	179
75	63	7	1	38	74	271
76	75	72	63	47	44	125
77	29	53	68	64	51	273
78	40	65	24	69	61	131
79	46	75	72	52	62	176
80	46	76	73	16	64	276
81	65	77	74	70	64	30
82	63	78	75	38	66	295
83	82	33	76	47	73	77
84	40	80	30	44	67	193
85	72	81	23	29	75	130

86	54	80	59	77	85	169
87	73	83	57	76	56	172
88	48	78	55	66	59	150
89	60	86	80	70	79	242
90	89	75	83	72	73	25
91	90	79	84	80	74	325
92	38	88	43	68	63	40
93	68	89	69	82	24	0
94	73	76	90	36	28	66
95	94	91	88	8	78	56
96	15	21	89	82	79	194
97	21	59	93	86	66	154
98	97	94	91	87	95	238
99	98	78	92	88	63	165
100	82	7	93	89	83	268
101	91	97	94	90	53	162
102	14	37	48	40	69	205
103	102	99	96	60	45	76
104	103	29	59	93	60	38
105	79	84	100	94	28	27
106	90	100	56	71	72	212
107	82	2	98	85	27	262
108	71	89	72	95	96	284
109	108	105	33	98	5	15
110	108	104	92	99	33	58
111	87	95	21	14	28	279
112	94	108	55	101	107	29
113	53	9	83	33	91	141
114	4	111	81	69	106	63
115	107	88	108	77	98	126
116	96	102	108	48	99	305
117	116	79	110	100	86	32
118	112	114	104	80	59	249
119	71	118	22	110	115	304
120	59	77	108	93	97	127
121	81	117	115	110	106	151
122	79	95	121	111	74	152
123	122	104	113	59	109	226
124	123	34	97	113	107	299
125	54	101	33	122	109	254
126	125	43	119	87	61	157
127	40	62	69	114	118	264
128	74	86	64	71	110	303
129	24	76	114	106	112	50
130	106	49	16	121	92	116
131	130	121	124	127	118	53

132	131	117	110	71	115	236
133	78	120	126	132	59	2
134	83	103	60	93	113	307
135	108	18	128	124	71	158
136	115	89	131	70	127	49
137	59	133	130	126	120	26
138	41	112	37	107	119	161
139	18	135	92	128	80	308
140	94	55	52	83	130	324
141	140	99	134	42	17	23
142	141	138	135	131	125	105
143	64	128	136	132	68	99
144	142	95	122	7	62	137
145	96	93	89	82	21	135
146	44	131	139	135	124	88
147	116	123	99	102	139	133
148	147	117	61	137	100	129
149	62	145	142	138	77	48
150	70	136	143	145	115	51
151	150	28	57	95	111	132
152	41	148	1	142	37	87
153	138	149	121	35	145	234
154	128	104	113	112	68	323
155	154	61	151	144	40	72
156	138	87	149	133	126	199
157	103	102	132	22	15	317
158	157	154	151	17	141	74
159	116	102	41	148	142	214
160	159	136	153	150	70	93
161	69	128	98	127	67	278
162	161	158	155	151	145	8
163	153	136	149	133	159	196
164	112	116	107	139	147	185
165	147	32	61	163	155	113
166	165	94	91	155	90	221
167	162	137	74	111	113	198
168	66	59	78	133	126	148
169	135	112	162	155	165	252
170	169	109	163	159	153	195
171	131	167	151	98	154	257
172	161	168	155	171	67	139
173	138	169	149	162	156	300
174	167	101	107	21	164	313
175	76	152	174	168	36	206
176	175	35	169	165	87	81
177	176	173	170	166	160	95

178	177	174	171	167	97	84
179	173	109	141	117	103	203
180	179	176	149	136	173	6
181	171	49	174	155	132	101
182	178	177	166	176	165	145
183	109	170	176	98	166	83
184	56	156	163	53	119	258
185	0	96	102	60	168	110
186	112	182	165	153	169	79
187	186	183	52	109	105	97
188	130	172	92	161	182	306
189	110	188	181	128	160	328
190	115	174	183	186	173	256
191	124	171	167	97	174	22
192	159	157	185	177	123	46
193	11	141	175	177	182	320
194	183	190	157	118	177	215
195	35	191	101	125	124	14
196	143	187	175	134	152	263
197	196	194	68	186	182	106
198	197	194	191	187	181	12
199	198	195	192	188	164	44
200	119	196	134	146	183	235
201	200	197	194	190	184	1
202	120	80	164	128	84	41
203	202	123	196	182	186	275
204	203	196	152	193	190	55
205	172	201	198	192	202	281
206	205	111	146	195	160	5
207	172	198	201	154	191	259
208	199	86	201	202	85	103
209	172	205	202	198	192	121
210	127	206	203	199	208	35
211	160	84	158	146	157	270
212	211	168	67	155	46	153
213	210	143	207	81	196	174
214	213	210	207	203	140	96
215	214	128	208	203	160	239
216	215	52	140	134	48	322
217	208	66	88	201	48	181
218	217	184	192	161	158	246
219	213	184	212	126	210	237
220	219	216	162	75	137	80
221	129	217	84	152	204	69
222	203	160	215	214	167	294
223	191	94	7	212	38	321

224	149	129	217	138	145	269
225	224	188	168	214	81	156
226	185	200	184	190	194	171
227	185	140	223	216	210	111
228	182	175	195	146	211	57
229	228	225	147	223	212	75
230	142	189	41	110	203	319
231	213	173	229	222	207	70
232	125	195	23	56	158	208
233	220	219	215	54	27	108
234	144	113	224	83	228	315
235	139	231	234	224	218	17
236	235	232	229	225	228	187
237	152	37	221	92	158	329
238	237	185	231	227	216	138
239	237	177	235	228	151	292
240	239	236	235	229	228	149
241	157	171	234	183	224	78
242	236	238	235	232	225	21
243	174	239	236	219	142	192
244	241	240	215	156	77	60
245	154	170	227	71	185	285
246	188	129	239	242	241	260
247	206	73	229	180	144	91
248	207	239	240	237	241	52
249	207	222	192	238	213	7
250	241	240	229	239	125	107
251	209	226	181	191	234	287
252	234	134	146	200	26	327
253	232	225	246	242	236	302
254	144	129	247	243	237	13
255	82	251	237	27	144	164
256	252	226	230	180	222	248
257	241	256	249	246	240	210
258	202	254	234	247	251	207
259	258	254	238	179	67	253
260	205	256	253	249	213	73
261	260	253	244	250	177	280
262	255	142	261	251	181	182
263	222	259	256	252	246	224
264	263	260	257	253	261	128
265	179	261	258	254	247	47
266	156	208	67	255	249	65
267	105	263	260	256	250	62
268	215	263	222	250	96	251
269	230	267	180	21	218	168

270	269	248	258	247	268	28
271	270	106	269	218	124	291
272	156	262	220	181	255	183
273	272	269	206	143	208	89
274	268	193	204	103	190	9
275	134	141	236	264	52	316
276	275	272	262	197	259	219
277	265	255	245	251	214	90
278	173	241	271	267	277	191
279	278	276	272	101	275	290
280	264	261	245	144	263	190
281	252	277	256	95	213	261
282	230	252	226	189	281	71
283	260	279	276	272	84	228
284	55	246	148	259	271	155
285	252	281	278	274	268	229
286	223	218	150	206	269	250
287	286	283	280	276	272	124
288	274	241	257	277	270	222
289	266	273	282	219	269	277
290	289	286	244	226	273	142
291	281	265	284	283	274	120
292	250	279	288	282	275	318
293	292	285	221	282	209	20
294	262	290	120	245	277	61
295	235	291	121	284	278	45
296	267	292	289	285	193	82
297	294	105	290	286	280	3
298	294	290	280	287	240	67
299	273	140	266	288	282	240
300	264	224	293	166	135	159
301	300	211	294	290	264	310
302	281	298	295	278	284	140
303	302	293	296	292	286	115
304	298	271	90	300	127	286
305	299	282	140	279	288	314
306	305	302	299	248	295	24
307	306	303	187	233	220	4
308	307	304	301	297	291	98
309	292	302	221	137	275	109
310	296	306	257	299	305	94
311	228	301	304	300	307	112
312	253	308	305	248	295	173
313	311	289	184	238	217	136
314	313	291	307	303	297	177
315	219	311	308	304	298	36

316	243	280	309	214	242	247
317	225	211	212	137	159	282
318	306	287	198	114	187	160
319	318	315	309	308	301	68
320	310	216	317	204	123	213
321	320	317	302	310	285	43
322	321	318	315	312	320	225
323	139	164	316	314	184	272
324	270	287	319	179	258	144
325	178	321	323	314	316	102
326	325	199	321	315	322	184
327	251	314	246	319	300	33
328	327	324	233	325	279	241
329	328	325	322	318	312	42
330	329	314	323	319	313	59

ProVEST-16 root family

RNS counters indexes:

0, 16, 17, 18, 1, 19, 20, 2, 3, 4, 5, 6, 7, 8, 9, 10

Feedback function indexes:

78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403

Input bit permutation indexes:

83, 84, 53, 38, 25, 42, 87, 66, 29, 55, 21, 114, 98, 111, 5, 110, 46, 115, 117, 104, 107, 76, 1, 44, 3, 113, 69, 71, 102, 41, 85, 4, 19, 108, 80, 9, 79, 77, 65, 48, 23, 14, 82, 109, 6, 17, 75, 0, 105, 64, 61, 74, 24, 100, 16, 90, 10, 2, 78, 116, 106, 27, 68, 33, 60, 12, 13, 52, 49, 15, 50, 94, 119, 92, 112, 40, 58, 103, 99, 54, 30, 37, 73, 118, 62, 35, 101, 88, 97, 22, 93, 20, 91, 96, 39, 8, 51, 43, 81, 95, 28, 18, 7, 57, 11, 63, 70, 26, 59, 32, 56, 72, 47, 36, 31, 86, 89, 34, 45, 67, 0, 86, 69, 83, 84, 45, 49, 54, 32, 67, 105, 112, 104, 101, 114, 32, 70, 97, 109, 105, 108, 121, 32, 84, 114, 101, 101, 10, 68, 101, 115, 105, 103, 110, 101, 100, 32, 98, 121, 32, 83, 101, 97, 110, 32, 79, 39, 78, 101, 105, 108, 10, 67, 66, 32, 67, 97, 112, 105, 116, 97, 108, 32, 77, 97, 110, 97, 103, 101, 109, 101, 110, 116, 32, 83, 46, 65, 46, 0, 45, 42, 103, 106, 71, 82, 29, 86, 90, 4, 33, 70, 31, 114, 9, 99, 15, 72, 46, 79, 61, 75, 97, 78, 63, 115, 34, 67, 96, 116, 51, 5, 55, 50, 21, 112, 118, 40, 104, 26, 98, 14, 60, 91, 65, 92, 49, 6, 47, 24, 23, 20, 57, 18, 80, 2, 54, 19, 74, 64, 59, 111, 48, 108, 37, 30, 35, 17, 39, 62, 85, 53, 94, 52, 16, 38, 7, 36, 3, 102, 44, 100, 13, 76, 89, 66, 27, 88, 32, 58, 81, 8, 119, 1, 69, 110, 87, 77, 101, 22, 107, 0, 105, 73, 28, 117, 12, 93, 84, 43, 83, 113, 95, 10, 56, 109, 11, 68, 41, 25, 2, 3, 5, 7, 11, 13, 17

Appendix C

VEST-32 core accumulator bit permutation

j	$p_{j[0]}$	$p_{j[1]}$	$p_{j[2]}$	$p_{j[3]}$	$p_{j[4]}$	$p_{j[5]}$
0	0	1	2	3	4	505
1	0	1	2	3	4	565
2	0	1	2	3	4	209
3	0	1	2	3	4	238
4	0	1	2	3	4	378
5	4	4	1	3	2	157
6	5	5	2	4	3	495
7	1	1	3	0	4	364
8	7	7	4	1	5	103
9	8	8	5	2	6	514
10	9	0	7	3	5	414
11	10	7	4	0	6	345
12	11	6	5	9	7	518
13	12	9	6	10	8	267
14	13	10	7	3	9	124
15	13	11	8	4	5	516
16	15	2	9	5	11	339
17	16	13	10	6	0	248
18	17	14	11	6	15	372
19	14	15	12	8	2	404
20	19	16	13	9	3	100
21	17	16	8	10	9	391
22	18	8	15	11	5	318
23	22	9	2	12	6	440
24	23	22	17	9	7	494
25	0	5	24	21	8	520
26	0	25	1	2	4	359
27	6	9	20	16	10	365
28	22	24	21	23	11	371
29	28	23	22	18	12	300
30	4	1	27	7	10	19
31	21	27	24	17	14	227
32	31	16	28	21	24	247
33	32	29	26	20	7	81
34	33	29	12	23	17	469
35	34	31	28	24	21	13
36	33	32	35	25	19	187
37	36	18	6	11	20	477
38	35	15	28	27	21	249
39	22	12	15	14	18	259

40	39	17	33	29	22	425
41	40	28	34	30	14	149
42	23	38	35	41	25	389
43	42	39	15	27	31	203
44	43	40	14	31	21	221
45	44	41	38	34	28	158
46	45	42	39	35	29	195
47	46	43	40	44	30	463
48	47	18	41	37	31	448
49	48	45	37	39	32	388
50	42	36	43	39	33	399
51	3	47	50	40	44	268
52	37	36	29	32	35	461
53	52	35	46	42	36	539
54	53	48	47	16	37	200
55	46	51	42	45	40	314
56	8	52	49	45	39	279
57	56	53	50	46	40	168
58	57	43	33	44	23	317
59	10	46	44	48	34	336
60	50	24	53	49	56	320
61	49	20	41	3	26	332
62	15	61	39	11	18	258
63	26	31	40	10	20	289
64	63	49	57	53	47	347
65	55	22	24	61	48	397
66	52	50	19	55	49	230
67	45	63	60	56	38	90
68	55	60	56	16	64	534
69	68	65	62	58	52	12
70	15	66	27	1	11	316
71	70	67	64	60	54	121
72	54	17	71	61	55	43
73	25	19	66	62	56	466
74	31	71	50	44	57	554
75	24	21	68	64	58	529
76	75	68	69	64	59	384
77	76	51	70	66	60	0
78	76	62	34	28	61	464
79	29	37	72	68	62	101
80	79	64	25	69	73	262
81	44	14	8	55	80	222
82	18	69	75	78	65	170
83	11	79	76	19	66	482
84	83	80	77	49	67	472
85	44	74	3	65	82	538

86	85	36	83	75	82	184
87	86	83	80	76	71	310
88	67	84	81	77	71	288
89	55	52	20	59	72	475
90	47	86	31	79	73	22
91	90	86	84	80	74	139
92	72	38	41	42	75	215
93	92	60	86	82	76	224
94	72	16	29	33	68	1
95	54	58	89	13	78	574
96	82	93	65	39	90	486
97	54	93	92	86	80	400
98	97	63	91	87	35	271
99	36	18	92	88	82	264
100	87	96	64	44	45	583
101	36	97	94	90	91	110
102	101	97	58	83	85	128
103	88	99	25	92	96	206
104	85	30	97	78	87	119
105	73	101	98	99	82	321
106	4	10	63	2	1	218
107	58	71	100	18	101	569
108	107	85	36	97	91	323
109	108	105	102	98	61	74
110	109	106	59	99	38	177
111	105	20	55	89	26	441
112	111	61	105	55	94	350
113	104	17	29	16	96	164
114	113	64	91	108	97	335
115	73	114	108	91	98	508
116	36	35	109	83	99	325
117	26	12	113	98	100	542
118	98	38	112	13	14	541
119	118	70	112	108	102	197
120	63	107	97	118	94	506
121	59	42	7	102	61	552
122	102	109	39	111	105	515
123	122	119	116	112	53	188
124	123	91	118	71	107	485
125	73	19	118	114	108	563
126	89	57	87	70	109	79
127	113	123	81	116	110	311
128	29	95	121	2	73	252
129	85	101	104	56	5	576
130	81	129	123	110	113	499
131	96	83	124	102	114	192

132	131	75	125	128	106	25
133	109	57	129	122	63	117
134	92	122	99	123	116	285
135	113	104	101	124	41	446
136	135	132	129	125	3	307
137	94	133	112	126	3	278
138	47	65	107	127	54	272
139	136	135	23	54	122	402
140	139	136	133	129	127	556
141	140	53	134	130	124	245
142	141	138	122	121	125	226
143	65	135	136	132	126	360
144	109	105	87	99	89	299
145	42	36	91	27	39	381
146	63	32	139	138	129	277
147	146	143	139	136	130	423
148	134	144	141	137	131	275
149	148	47	131	138	132	408
150	149	27	146	93	142	282
151	17	147	144	140	134	386
152	85	151	145	141	135	557
153	131	51	136	138	78	291
154	38	134	153	143	137	562
155	154	137	34	40	138	390
156	23	143	121	100	150	75
157	142	153	150	68	152	309
158	144	96	151	105	141	488
159	113	155	152	104	153	330
160	159	121	95	149	143	487
161	147	134	154	150	144	329
162	82	158	155	123	145	362
163	28	161	156	43	147	301
164	163	112	137	139	53	302
165	122	123	145	63	128	159
166	121	162	122	163	119	543
167	51	35	160	153	150	409
168	120	0	49	112	161	260
169	26	94	120	115	84	231
170	144	123	163	165	164	86
171	23	170	149	160	84	261
172	171	149	151	161	10	70
173	46	169	126	162	156	108
174	78	171	75	173	89	531
175	14	98	168	89	158	92
176	175	172	169	162	159	167
177	67	133	146	147	160	528

178	156	67	171	33	161	403
179	178	170	172	151	147	292
180	76	176	93	133	158	242
181	45	177	174	170	164	95
182	150	131	175	37	165	61
183	134	116	123	164	166	217
184	140	180	28	173	167	214
185	33	181	177	77	43	52
186	113	106	179	140	165	415
187	72	183	98	175	121	567
188	174	100	181	5	171	270
189	86	185	182	178	172	273
190	145	186	183	179	173	93
191	118	187	112	180	163	64
192	50	188	190	134	88	411
193	159	76	48	157	176	582
194	71	176	81	183	174	6
195	98	72	115	188	105	585
196	24	79	189	69	59	156
197	189	20	88	48	59	533
198	132	194	191	161	58	50
199	166	128	22	80	171	426
200	199	80	193	178	183	153
201	200	197	194	190	179	67
202	194	198	146	191	185	48
203	95	149	89	101	2	294
204	203	140	197	193	99	26
205	204	201	198	194	188	72
206	205	141	198	64	73	104
207	187	88	142	196	190	296
208	32	204	7	197	191	136
209	77	74	84	48	56	379
210	115	204	119	199	106	34
211	196	107	94	193	120	476
212	211	208	205	156	195	31
213	181	152	206	209	160	57
214	213	166	116	203	186	37
215	180	115	169	5	128	77
216	126	62	77	114	206	447
217	216	213	154	206	200	66
218	217	214	211	207	201	233
219	109	215	212	208	202	571
220	155	186	201	111	132	555
221	220	217	214	210	204	69
222	202	211	167	78	107	132
223	222	219	216	154	200	28

224	142	196	110	200	190	383
225	224	168	203	84	167	491
226	100	222	220	66	96	298
227	199	206	200	40	190	368
228	144	200	227	130	196	545
229	228	225	8	218	221	83
230	229	226	223	219	213	5
231	147	227	224	196	230	122
232	111	98	225	228	215	269
233	74	203	48	181	166	431
234	216	79	222	189	233	342
235	103	234	161	125	37	577
236	235	232	229	131	219	46
237	118	21	134	65	191	420
238	188	106	30	234	221	91
239	120	175	135	188	115	559
240	219	41	154	229	223	305
241	240	237	234	163	224	88
242	241	154	235	231	225	127
243	88	218	92	236	226	204
244	196	229	237	233	85	352
245	244	241	232	239	182	308
246	245	242	239	235	229	444
247	246	243	190	236	168	62
248	73	196	233	237	80	547
249	60	245	242	239	171	284
250	27	93	195	221	38	198
251	174	52	244	127	32	41
252	142	57	43	241	207	171
253	245	19	246	242	152	458
254	130	253	144	247	208	213
255	46	34	222	244	98	490
256	255	156	249	245	239	142
257	179	115	253	173	240	193
258	231	244	103	117	98	59
259	210	184	78	151	242	120
260	166	256	253	72	187	370
261	133	252	216	198	204	427
262	261	258	239	255	245	162
263	150	259	97	186	246	138
264	256	105	252	260	44	114
265	243	159	263	259	211	568
266	265	262	87	255	249	7
267	102	178	254	125	162	76
268	267	264	23	257	230	208
269	268	265	262	258	60	181

270	210	232	16	262	218	250
271	169	267	264	260	254	326
272	211	254	263	244	243	578
273	228	219	229	247	272	324
274	266	260	264	236	257	550
275	171	271	268	264	274	176
276	169	42	46	265	259	328
277	206	268	141	170	205	237
278	277	255	239	183	261	465
279	212	243	110	249	58	173
280	135	235	165	118	172	306
281	195	267	275	270	264	228
282	234	66	275	271	194	135
283	252	197	269	272	263	451
284	283	241	277	280	273	105
285	248	96	143	142	183	396
286	202	282	51	133	271	511
287	241	283	280	243	277	155
288	287	284	281	274	185	327
289	164	87	261	151	272	338
290	289	228	280	201	273	382
291	290	148	284	65	274	68
292	218	145	285	281	107	255
293	275	208	101	282	20	480
294	270	290	287	283	277	179
295	66	291	288	167	278	468
296	164	83	147	231	285	186
297	232	121	157	166	150	548
298	102	288	188	294	166	438
299	298	295	292	215	290	254
300	299	281	230	289	137	413
301	189	281	87	294	284	546
302	172	298	108	291	251	361
303	180	161	171	262	286	449
304	303	300	297	293	287	357
305	304	214	220	227	181	416
306	231	272	224	247	304	525
307	282	133	12	197	263	219
308	248	79	78	300	82	398
309	263	234	140	186	305	424
310	189	70	253	291	235	146
311	251	307	304	187	181	84
312	157	308	210	311	178	54
313	305	253	147	144	306	561
314	310	179	190	240	120	462
315	310	114	308	304	303	498

316	80	193	314	211	312	337
317	30	196	172	302	130	385
318	251	207	293	100	271	497
319	199	283	312	306	137	85
320	274	148	288	309	303	191
321	304	317	314	310	205	313
322	313	266	318	311	276	44
323	322	306	319	141	316	303
324	308	323	258	313	307	143
325	297	180	289	230	95	443
326	320	325	319	1	298	513
327	165	160	300	171	174	395
328	129	185	321	279	202	87
329	315	227	190	226	267	322
330	84	148	120	319	220	212
331	330	192	324	288	314	239
332	331	312	146	328	246	180
333	125	323	319	322	309	523
334	313	285	327	307	145	161
335	334	56	327	61	272	558
336	308	335	111	333	95	586
337	274	333	246	74	329	417
338	276	335	90	327	321	349
339	338	335	332	328	327	433
340	212	106	333	329	250	517
341	340	337	334	330	324	20
342	315	205	152	319	333	178
343	342	339	336	332	326	29
344	186	183	337	334	327	232
345	344	341	338	335	328	165
346	299	332	339	335	284	380
347	346	330	209	336	170	412
348	301	190	217	49	323	481
349	53	193	247	312	332	455
350	348	346	254	293	201	18
351	315	347	344	340	334	130
352	139	300	220	351	306	60
353	266	349	346	251	336	183
354	269	341	193	188	162	564
355	117	351	348	344	86	202
356	167	281	320	182	318	502
357	348	353	74	346	340	148
358	155	354	351	347	341	58
359	180	355	213	349	342	484
360	317	228	353	349	172	115
361	32	257	26	111	311	134

362	361	358	355	263	345	35
363	362	356	340	352	346	434
364	71	50	339	192	347	151
365	301	361	286	285	348	240
366	365	362	359	355	356	65
367	340	363	297	356	350	73
368	367	363	295	357	239	32
369	343	126	149	331	310	21
370	117	366	363	359	353	14
371	176	169	365	360	159	30
372	244	178	263	358	338	56
373	102	313	296	318	356	36
374	261	328	311	250	320	293
375	94	148	371	228	273	145
376	199	236	155	54	302	8
377	6	130	226	154	323	367
378	368	292	293	210	243	522
379	182	244	372	301	362	297
380	251	113	50	106	38	467
381	207	377	254	202	373	210
382	295	163	375	119	137	109
383	382	379	376	227	366	253
384	207	317	186	299	175	376
385	384	305	192	374	368	55
386	385	282	325	258	215	457
387	80	255	312	350	370	185
388	380	384	387	114	371	510
389	388	385	165	382	316	236
390	389	386	383	379	298	107
391	390	387	384	380	374	53
392	127	153	69	266	179	401
393	392	389	256	382	376	251
394	303	250	387	286	297	144
395	394	391	305	384	182	129
396	109	389	276	194	201	526
397	291	396	393	381	380	560
398	167	191	397	341	381	489
399	376	395	392	375	382	118
400	234	260	393	275	115	493
401	176	320	354	361	331	205
402	182	390	386	259	385	343
403	393	338	313	258	387	418
404	225	400	212	301	393	355
405	361	250	329	351	279	581
406	405	51	404	395	389	340
407	371	354	367	358	146	225

408	117	349	401	364	391	356
409	408	405	399	270	283	116
410	409	406	333	403	298	452
411	410	342	225	400	394	346
412	407	324	265	268	299	287
413	169	223	265	392	396	492
414	413	386	407	403	397	445
415	414	411	408	404	398	199
416	136	130	409	415	399	82
417	375	296	252	269	63	223
418	249	223	414	388	91	211
419	266	344	337	127	383	503
420	419	416	413	383	403	131
421	304	324	181	323	235	459
422	269	208	418	262	405	537
423	333	419	416	412	406	38
424	423	420	345	413	326	89
425	422	377	402	382	408	333
426	316	353	367	360	125	113
427	309	423	407	416	413	373
428	198	424	421	417	411	160
429	428	425	422	418	412	229
430	0	411	279	371	407	572
431	357	394	424	330	286	169
432	431	428	207	422	415	566
433	432	429	426	422	240	23
434	433	430	427	90	417	125
435	434	431	428	424	418	111
436	435	116	295	278	419	470
437	280	407	290	273	294	570
438	358	143	300	415	160	584
439	438	435	432	428	422	78
440	127	439	433	429	377	166
441	440	437	434	430	243	49
442	385	233	417	215	425	540
443	442	343	257	111	426	71
444	290	299	437	341	427	196
445	173	265	160	434	406	454
446	350	442	325	369	248	579
447	423	110	444	65	345	519
448	447	309	441	405	357	483
449	448	445	435	438	432	532
450	223	256	446	436	203	112
451	403	447	444	373	410	509
452	436	448	278	441	435	175
453	452	408	446	442	436	286

454	421	449	420	315	415	256
455	354	451	448	444	297	106
456	310	360	152	410	434	575
457	360	429	434	280	440	189
458	184	365	451	447	388	530
459	458	454	394	354	88	453
460	450	234	275	405	427	266
461	372	437	453	445	322	504
462	381	448	115	451	445	351
463	462	459	456	452	446	4
464	420	431	369	385	436	319
465	190	461	458	454	464	42
466	465	462	459	455	118	17
467	405	400	460	447	450	140
468	467	352	396	343	334	375
469	348	179	397	387	452	126
470	166	256	212	249	383	580
471	451	467	468	457	402	437
472	471	196	468	354	455	265
473	326	128	466	99	266	150
474	363	303	108	329	443	94
475	286	471	393	464	421	334
476	475	450	460	465	459	97
477	476	294	470	366	466	331
478	445	390	214	381	440	147
479	419	110	292	353	475	551
480	437	418	248	217	476	512
481	428	305	474	250	413	315
482	455	276	475	471	439	133
483	398	465	476	159	466	521
484	177	401	477	473	195	394
485	484	481	471	474	468	40
486	225	473	482	475	469	194
487	335	461	480	476	470	478
488	487	369	379	276	439	33
489	488	485	287	331	472	98
490	489	486	273	479	478	11
491	490	441	124	96	474	524
492	397	293	362	481	365	442
493	492	489	406	482	469	241
494	493	177	487	483	477	15
495	494	491	488	202	478	421
496	379	492	142	485	478	216
497	464	493	327	486	496	363
498	158	484	491	474	481	544
499	296	452	492	203	482	527

500	480	309	248	185	130	460
501	500	487	494	490	370	137
502	501	498	161	292	485	174
503	5	483	378	479	174	163
504	503	500	497	493	487	24
505	46	11	367	453	443	220
506	352	479	499	468	355	405
507	412	480	251	182	501	430
508	507	504	501	497	491	80
509	443	474	502	322	57	263
510	304	509	506	499	493	290
511	429	430	433	498	419	536
512	511	508	450	501	261	96
513	512	509	506	435	474	276
514	513	259	507	503	233	535
515	373	511	455	504	508	201
516	412	515	509	386	499	436
517	507	513	279	242	421	172
518	325	474	481	471	369	479
519	469	435	512	498	502	354
520	457	286	395	233	432	377
521	308	184	45	510	504	3
522	521	463	515	511	252	358
523	512	454	358	472	237	182
524	453	509	517	513	507	9
525	308	437	431	326	496	393
526	525	522	423	502	366	51
527	396	168	523	358	484	500
528	462	524	521	502	449	280
529	528	250	457	216	524	274
530	496	526	59	246	438	344
531	525	458	394	232	281	369
532	421	104	133	521	516	257
533	532	530	265	522	499	553
534	480	454	401	447	303	422
535	534	531	523	463	521	496
536	535	532	529	224	519	244
537	522	533	530	492	520	353
538	446	351	531	302	430	473
539	456	132	445	57	488	573
540	471	536	525	477	523	283
541	540	537	439	530	496	190
542	453	510	535	400	463	435
543	542	539	536	532	526	152
544	518	430	357	368	291	207
545	531	436	538	534	528	10

546	295	402	539	535	429	450
547	546	409	516	384	399	154
548	467	544	540	85	104	366
549	517	533	479	294	519	456
550	461	527	522	369	532	63
551	537	392	373	544	199	387
552	473	388	542	452	519	501
553	535	488	197	542	536	27
554	553	550	541	543	233	312
555	417	478	410	544	520	341
556	555	552	549	545	551	474
557	556	553	549	546	540	507
558	392	500	470	547	132	2
559	510	321	456	548	459	406
560	370	339	553	343	503	410
561	560	398	557	353	554	374
562	530	407	555	551	545	429
563	562	559	556	552	546	243
564	335	548	557	553	560	281
565	564	561	558	554	553	295
566	449	529	192	543	307	39
567	566	529	524	374	550	419
568	550	444	460	538	472	348
569	559	417	473	527	544	407
570	569	566	563	552	374	234
571	570	390	564	560	554	45
572	551	571	398	561	500	246
573	572	569	566	562	556	16
574	457	570	567	563	557	47
575	574	571	568	564	558	141
576	456	470	528	572	396	304
577	576	524	456	559	560	439
578	577	574	571	567	561	123
579	549	517	484	568	562	102
580	579	570	567	458	506	428
581	580	344	576	123	575	392
582	581	283	148	539	578	432
583	492	466	462	572	578	549
584	372	580	583	573	581	471
585	483	578	410	574	579	235
586	575	582	207	377	344	99

ProVEST-32 root family

RNS counters indexes:

0, 16, 17, 18, 1, 19, 20, 2, 3, 4, 5, 6, 7, 8, 9, 10

Feedback function indexes:

404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 780, 981, 982, 983, 984, 985

Input bit permutation indexes:

15, 60, 99, 94, 56, 42, 89, 46, 96, 114, 36, 92, 106, 85, 66, 76, 93, 31, 105, 44, 25, 112, 113, 103, 111, 58, 78, 12, 34, 0, 32, 54, 80, 82, 118, 72, 49, 20, 115, 81, 39, 38, 73, 90, 77, 16, 109, 10, 84, 14, 83, 21, 11, 4, 41, 98, 6, 117, 19, 74, 70, 101, 95, 86, 119, 59, 69, 110, 75, 8, 5, 104, 27, 71, 48, 13, 37, 30, 35, 108, 53, 50, 45, 61, 47, 52, 33, 102, 67, 18, 116, 55, 65, 51, 24, 97, 107, 63, 3, 68, 40, 43, 9, 1, 57, 88, 87, 2, 7, 17, 29, 23, 26, 100, 62, 22, 79, 64, 91, 28, 101, 92, 24, 53, 97, 78, 108, 116, 80, 86, 81, 71, 114, 50, 40, 89, 47, 42, 59, 77, 69, 105, 72, 57, 19, 36, 17, 16, 55, 99, 56, 106, 26, 110, 85, 13, 119, 82, 88, 70, 102, 22, 84, 32, 30, 3, 64, 74, 25, 118, 6, 37, 43, 46, 100, 79, 1, 73, 12, 9, 68, 23, 93, 66, 27, 60, 95, 31, 112, 90, 33, 94, 87, 45, 18, 21, 96, 38, 63, 117, 5, 48, 8, 103, 107, 0, 76, 62, 111, 115, 44, 20, 39, 83, 15, 10, 7, 2, 29, 58, 61, 67, 104, 51, 4, 91, 113, 49, 98, 54, 34, 35, 11, 75, 14, 52, 109, 65, 28, 41, 0, 86, 69, 83, 84, 45, 51, 50, 32, 67, 105, 112, 104, 101, 114, 32, 70, 97, 109, 105, 108, 121, 32, 84, 114, 101, 101, 10, 68, 101, 115, 105, 103, 110, 101, 100, 32, 98, 121, 32, 83, 101, 97, 110, 32, 79, 39, 78, 101, 105, 108, 10, 67, 66, 32, 67, 97, 112, 105, 116, 97, 108, 32, 77, 97, 110, 97, 103, 101, 109, 101, 110, 116, 32, 83, 46, 65, 46, 0, 90, 102, 100, 73, 75, 60, 17, 26, 84, 67, 19, 0, 61, 57, 85, 32, 115, 40, 4, 76, 46, 2, 83, 77, 36, 116, 12, 81, 103, 11, 16, 113, 55, 27, 94, 106, 51, 79, 3, 56, 30, 39, 112, 29, 66, 69, 74, 10, 101, 43, 28, 107, 98, 91, 80, 88, 87, 47, 21, 118, 114, 8, 37, 68, 93, 70, 49, 33, 50, 58, 52, 22, 38, 14, 45, 62, 25, 54, 105, 119, 6, 18, 34, 1, 23, 35, 53, 109, 96, 117, 64, 63, 111, 108, 24, 48, 78, 5, 15, 89, 42, 92, 97, 41, 20, 86, 7, 13, 71, 65, 99, 59, 9, 82, 110, 72, 95, 44, 31, 104, 13, 89, 112, 37, 17, 106, 55, 81, 5, 107, 29, 103, 11, 50, 66, 22, 93, 41, 47, 62, 88, 79, 60, 82, 49, 18, 87, 2, 23, 99, 25, 76, 45, 16, 57, 68, 64, 116, 58, 97, 102, 36, 65, 92, 30, 48, 19, 118, 117, 85, 96, 69, 95, 86, 9, 34, 108, 98, 28, 72, 70, 80, 74, 33, 15, 94, 31, 8, 21, 56, 12, 1, 67, 84, 63, 59, 113, 14, 100, 24, 78, 27, 43, 110, 39, 0, 111, 104, 53, 32, 73, 101, 35, 115, 7, 3, 77, 44, 105, 91, 54, 83, 42, 75, 119, 90, 20, 6, 26, 10, 114, 4, 109, 38, 52, 71, 40, 51, 61, 46, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83

Appendix D

VEST core accumulator feedback functions

```
vest_f[1024] = {
     0xDA46704F, 0x41F68DE1, 0x265DE859, 0x892FCC1E, 0xBA526D26,
     0xA96B11BC, 0x9CE447C6, 0xA265B61E, 0x4790CDB6, 0x10F8DF1C,
     0xC47CD24B, 0x97C62572, 0x1DC65D25, 0x38BC364B, 0x89F83336,
     0xA7654CB4, 0x631BCA96, 0x636A4BC3, 0x574C85DC, 0xC7052AEE,
     0x3539E70C, 0x13BFC18A, 0x3D5152D9, 0x72F5412E, 0xA1FE2385,
     0x3B6153AA, 0xB4625F25, 0x512AEE36, 0x9A8E18AF, 0x07C4CFC6,
     0x9B259B4A, 0x3E0E6276, 0x64A73A96, 0x19B66B23, 0xD7402DBC,
     0x85E23E36, 0x9F22D19A, 0x9F1530BC, 0x23F15DC2, 0xC0733AE6,
     0xB071ADA6, 0xB69D1C94, 0xA42BB656, 0x9C31A9CE, 0x972F8B21,
     0x4D2B6C2E, 0x838F74A6, 0xB9255798, 0x2E9A0DD6, 0x34F43633,
     0xF44C342F, 0x92E31A5E, 0x894CBEA5, 0x95A83C3B, 0xB62F8469,
     0xA87F05A6, 0x7109AF9A, 0x92D3BA94, 0x0DEF20E6, 0xF23329B4,
     0x72664B2E, 0x2A732E8E, 0x34D5D496, 0x5B237265, 0x8D229CF6,
     0xC7660DA5, 0x8C0BE772, 0xB96D518C, 0x356FA232, 0x8FD16496,
     0x4C5D59E4, 0x52AB627C, 0xB27E5065, 0xCA3B1D26, 0x9C7431AD,
     0x534EAA36, 0x5F418EE1, 0x91A4DF1C, 0x44E7729C, 0x351BE0AE,
     0xB5A3299C, 0xD543D52C, 0xD453D8A6, 0x8A64BF94, 0x70D6722D,
     0x9BA55166, 0xA372E643, 0xC6B81BC6, 0x13CBCD58, 0x36CA437A,
     0xD26A4B36, 0x8BF25516, 0x82DA6A9E, 0x45E18CDE, 0xBB1C2E43,
     0xC5239AB6, 0x446FCE46, 0x7E271946, 0x2D726A1E, 0x9C047BD9,
     0x462ABB2E, 0x723532D9, 0x96F709C4, 0x93258EF2, 0xA37A11CE,
     0xC1E629AD, 0x97893746, 0xF271133C, 0x547C95D2, 0x119AF1D9,
     0x364EC0F5, 0x6476598E, 0x47E20D6D, 0x867192FA, 0xC7F42265,
     0x8F5C614B, 0x4D2BB836, 0x96D50AE5, 0x9646C37C, 0xF1624676,
     0x90FC8E1B, 0x9849DACE, 0x079B9AC3, 0x12DDCC4E, 0x993BD416,
     0xF07D189C, 0x2F2937B0, 0xD2840FFA, 0xA65D4BD0, 0xF2261C9B,
     0x6179691E, 0xC1BC705E, 0xC7E64551, 0x26C165DE, 0x9769D522,
     0x39567A49, 0x0B98FE85, 0x0E25F4F4, 0x24FC6476, 0x8F813EC9,
     0xDC09759A, 0xE2336A8E, 0x9185F1EC, 0xC5866976, 0x8B684BF4,
     0xE55D0D16, 0xC07A2E5E, 0x50C37DAC, 0x4623BF98, 0xC1B3299E,
     0x94E17F30, 0xA74F0D1C, 0x8BE13AA6, 0x7334E949, 0x7B096A69,
     0xA1655BF0, 0xD0A76A69, 0x1A3DFC11, 0x4760F873, 0x17F3C50A,
     0x347F82A6, 0x2EE7191A, 0x721E43E5, 0x74956B26, 0x5330EDC3,
     0x3D0DAC56, 0xA5478DCC, 0x328575E9, 0x865FA744, 0xC42D2F6A,
     0xE3256B98, 0xC42A7B4B, 0x73865A59, 0x1F16A1E5, 0xB75700BC,
     0xB1EB590A, 0xA37E8D03, 0x3E624C3B, 0x56993E52, 0xA54566BC,
     0x8F99455A, 0x6A5A631E, 0x729945CE, 0x0CFAA51E, 0xD1AA33B1,
     0x712B6A72, 0x3627A69C, 0x879B1BA4, 0x16C2DF46, 0x51A1D8DE,
     0x47CC709B, 0x2C7A474D, 0x55BD43C4, 0x1AF22D66, 0x2B5A4C3B,
     0xCD1A6D19, 0x2BED1836, 0x4F5B6252, 0x25D6D0CB, 0x94BACB46,
     0xC07A55E6, 0x44EB19D6, 0x8276E1B9, 0xD525D9D0, 0x0ED04FDC,
     0x0E9EF825, 0x1A76E174, 0xC2539DC6, 0xDE053B94, 0x5A0C2FDA,
     0xA63D4BB0, 0x4E6F1958, 0x52B36D64, 0x21EDB19A, 0xB05CD6C3,
     0x784D661E, 0x1D884DFC, 0xF06B6552, 0x9E64897A, 0xE43F2986,
     0x2352EF45, 0x9727D30C, 0x927783AC, 0x737306C3, 0xA53D6652,
     0x61AE69C9, 0x77905D16, 0x568D0CFC, 0xE51D0D9A, 0x1EB233B4,
     0x186FE272, 0x4AD5790E, 0x4FA419D6, 0x3E2A4C73, 0x73A1392E,
     0x857D92C6, 0xD2AA12AF, 0xCC615C9E, 0x26A1B1FC, 0x741D4BCA,
     0x2C43E7D4, 0x77C511D2, 0x2FF42E03, 0x2B4D6DC1, 0x1C6E8B5A,
     0x9813F87A, 0x4E2EB316, 0x9D507374, 0x507BDA34, 0x747243E6,
     0x24A63AF5, 0x9271F4B4, 0xD7C5170C, 0xD63489AE, 0x9B7944D2,
     0x743ACD49, 0x2A29F71C, 0x560FC6B4, 0x1F8FC11C, 0x98714F6C,
     0x73376B04, 0xB833568E, 0xD7417D12, 0xA4532FB8, 0xA661B1DA,
     0x82E5774C, 0x2E722B5C, 0x9757C074, 0x9F9222E5, 0xD1643C8F,
     0xE46B09E6, 0xA25F38C3, 0xA74F5066, 0x4D2854FB, 0x3B236D38,
     0xBB2311E6, 0x5258DCA7, 0x06AFB85C, 0x1AFD470C, 0x46D55D8C,
     0xA531BD46, 0x9579F01A, 0xA52A4D6E, 0x305CD9DA, 0x99E90CB6,
     0x8497ED52, 0x1EB720E6, 0xD42A0FAD, 0x85A2BA3E, 0x897CB0B3,
     0x71B246E5, 0x1B8B792C, 0x9551F652, 0xA61D6BA4, 0xA6E5631A,
     0xC65ED225, 0x4C5D6D58, 0x9931A96E, 0xDB507159, 0xD40DC9BC,
     0x8753B0BA, 0x2E1EB463, 0xD6173D50, 0x9323F07A, 0x9501F6CE,
     0xA786703E, 0x14F990FC, 0xC7DE01E1, 0x9381DB6A, 0xB61381FA,
     0x52FB305C, 0x1938F52D, 0x6149D9DA, 0xAD3D254C, 0x082FAAED,
     0xA4D5339C, 0x36A57646, 0x534D8CF2, 0x5F14C8E3, 0x85498FBA,
     0xC259AB9C, 0x879538F4, 0x6C661D96, 0xF20E457C, 0x2FD04D1E,
     0x0D79D459, 0x4D617BA1, 0xB073F854, 0x1585EF98, 0x45B06F63,
     0x1A32EB3A, 0x9CD55616, 0xD651BC0E, 0xD0DC583B, 0x1D884BFA,
     0x871FE446, 0xC86913EE, 0x04B9B2F9, 0x92B670AD, 0x279970AE,
     0x94AB8DC6, 0x13E3C69A, 0x5365D94C, 0xC4636EA6, 0x1949EED1,
     0x774D13D0, 0x41DB7C23, 0xCF184D63, 0x24CF70F4, 0x8AEF4A26,
     0x22699DB6, 0x9853CB5C, 0xB526C23E, 0x7A6246B9, 0xA49F31A6,
     0x34B371F0, 0x8D3927A6, 0x0AC9DB5A, 0x9C17D49A, 0x2E4DD31C,
     0xB403AD6E, 0x82F72B64, 0x0DA24EBD, 0x9918AADD, 0x5C3A1B9A,
     0x1CF25B52, 0xAD3B07A4, 0x228DABF2, 0xDD051DCC, 0x54C74CF4,
     0x1C6EC54E, 0xB507D3A4, 0x4D59AD1A, 0x977489C6, 0x85A7F485,
     0x31BA43E3, 0xA529D23E, 0xB5239D64, 0x7218746F, 0x2765DA34,
     0x4F630ED4, 0x307DD651, 0x73BC30A3, 0x8E39D91A, 0xA6756A1A,
     0x0FBD3245, 0xBF510879, 0x1AF154CE, 0x13E0F746, 0x2D5A6A36,
     0x3749919E, 0x763470B3, 0xB58B5256, 0x9A4335EA, 0x0DDC6F12,
     0x26EA533C, 0xC719DA31, 0x4570AC6F, 0x5770A1CB, 0xAF284763,
     0x5366AA1E, 0x8C354FCC, 0x1E9A339C, 0x984E25FA, 0x4E6926AD,
     0x671C4AD6, 0x717D5C12, 0xA443ECCE, 0x80D0BFE6, 0x52AB5A66,
     0x0978ABBA, 0x3656E346, 0x9BBC409E, 0x8F8067CB, 0x2B7B43B0,
     0x1B8DE236, 0x88FF289A, 0x56C3A572, 0x86AF4666, 0x88FA398E,
     0xA1507FB4, 0x8D615A7C, 0xD7724594, 0xA71925AE, 0x4C52B5E9,
     0xE6163D34, 0x8691AE7C, 0x8378AB9A, 0x07B3DA86, 0x1D21FD62,
     0xB625543E, 0x5D25A3B4, 0x9E095DF0, 0x832D9E9C, 0x65CA7C19,
     0x93378E86, 0x904D9ABE, 0x4389E2DE, 0x195E8A8F, 0x3B585999,
     0x5C6F445C, 0x760B7A45, 0x5273D85A, 0x947D89CC, 0xD32B4674,
     0x928BE59C, 0xC4D6730E, 0x9453F70A, 0x7063B95C, 0x7354E929,
     0xA4BA3B86, 0x252AE72D, 0x949B549E, 0x468A5E75, 0xDF0A4596,
     0x924BF4D4, 0x27DB5582, 0x0F71AF50, 0x5D4C7706, 0x8377BC14,
     0xC6D8446F, 0xE05D68BA, 0x837EC456, 0x2C4665F9, 0x97368F14,
     0x9A7E12C3, 0x0C78CACF, 0x8D192EBC, 0x905CE71E, 0x5FA3119C,
     0xD0F549A6, 0x2AD92B2E, 0x5B20764F, 0x37D321AA, 0xC37324AE,
     0x991DEE05, 0x8B996378, 0xA3934DB8, 0x57D03F14, 0xB371523C,
     0x79642F16, 0xA7C70A1E, 0xB83D216E, 0x19F247C3, 0xC405FB66,
     0x42EB7516, 0x92ADBA26, 0xC1F158AD, 0x20EA7D4E, 0x15DCC44F,
     0x3D752B0C, 0x1FD06D45, 0xD83C3A27, 0x2350EA6F, 0x3D10D9B6,
     0x472DAB38, 0xA13B49DA, 0x2F3D710C, 0xA1BC3876, 0x717E3192,
     0xC43E8AB5, 0x992972AD, 0x8E254EB9, 0x7435D1D2, 0x757A2847,
     0x9564D82F, 0x1D4DA91E, 0x86FE8661, 0x06B1BFC2, 0x192CCBDA,
     0x0D4FE92A, 0xA6831BF4, 0x54C17D1E, 0x2FA21E59, 0x39605BAE,
     0xE71C7225, 0x0BDA649E, 0x1B3CCD85, 0x1B8D4CBA, 0x665C1DA6,
     0xA32A1E9E, 0x8E854B7A, 0x9C1AD2AD, 0x92F03B63, 0x7E1270E5,
     0x8767A52A, 0x1CB5CD26, 0x868B1FD8, 0x309ABE65, 0x320DFA66,
     0x4D1DE296, 0xDF24491E, 0x87A5C9B8, 0x0DE1A4BE, 0xE7690671,
     0x30F01EAF, 0xD321C75A, 0xF70B5D02, 0xD3C25336, 0x55E42E1E,
     0x34E3562D, 0x96E5634C, 0x51E958AD, 0x84A67B36, 0xC2D6750E,
     0x3DA74394, 0x516DE632, 0x8E8E0CF5, 0x8361CD7C, 0x0A69BBA9,
     0xC13A5BA6, 0x73575984, 0xC0D82EF3, 0xB3199D64, 0xB034BAC7,
     0x9292CD6E, 0x6F186356, 0x18E1D4F6, 0xA44DB99A, 0x1AA23BCE,
     0xB6097A66, 0xBA5D053C, 0x09DEC5C9, 0x58CD721E, 0x8AD062FD,
     0x986F5D90, 0xAD663156, 0x732A522F, 0xE75909E4, 0xC7D11C56,
     0x9E17705A, 0x8E06D1FA, 0x9A4336E6, 0xF34816D9, 0xD3C43556,
     0x968977C4, 0x85D2679C, 0x2D666359, 0x4F1569CA, 0xA2529DCE,
     0xD27196C3, 0x907AD752, 0x62754CDA, 0xA4BF412E, 0x706136ED,
     0xD3A83396, 0x22EDB13C, 0x77812EA9, 0xA1BC703E, 0x2E4B707A,
     0x985937F0, 0x95C17E86, 0x39C94B5C, 0x5B7CD203, 0x9F6182B6,
     0x53B81A9E, 0x068DFC5C, 0xD3124FB4, 0x5745D94A, 0x4543FDD0,
     0x8E47BD82, 0x5A276A3C, 0x5189BBA6, 0xAF4A3619, 0x83C5CD72,
     0x5743DD0C, 0x0C7E9B23, 0x13D9BC52, 0xC72D2AC6, 0x8B2DC36C,
     0xB2B91E49, 0x84DF1D8C, 0x0B5BA8DC, 0x354DA69C, 0x51C7AC96,
     0x8A2ADF86, 0x750471F9, 0xC766254E, 0x9D730A3A, 0x4427E5E9,
     0x70618DFC, 0x63AC3E91, 0x991D6E34, 0x4BC34DB4, 0x48E16E3E,
     0x1AEA0F3A, 0x84F5CCB4, 0xD4292BB6, 0x42E27CB3, 0x451CFD43,
     0x18C64EDD, 0x3AC54F1C, 0x44D2EF16, 0x84CFE4A6, 0x0A3ADF43,
     0xD24F28DA, 0xC07459BE, 0x824AEBBC, 0x84C76F4C, 0x42DD72C6,
     0x29DA0ED9, 0xCA1B5D52, 0x0F41DAF4, 0x9D6034F3, 0xBF076425,
     0x822BB1EE, 0x3E62705D, 0xA1714FF0, 0x8672E5B2, 0x7703C1DA,
     0x3176E40F, 0xB7A506D1, 0x44B46DF2, 0x8F917A51, 0x78656636,
     0x3438D3F4, 0x80DD7F42, 0xD276A561, 0x866FB90C, 0x45DA6B85,
     0xD3B951C2, 0x1E8E4F26, 0x617C670B, 0x56D86723, 0x2E7EB403,
     0x0EFE8E11, 0xD46D9926, 0x4A996B4E, 0x0C78C5FC, 0x1DB2719C,
     0xB9547951, 0x9F213572, 0x3EE6031E, 0x755D5B02, 0x991BD196,
     0x8E46BE19, 0x6C036DF4, 0xAA2D6A4E, 0x0D629EB3, 0x07A896F3,
     0x4A7C35C6, 0x886AD73C, 0x736E231A, 0x4DB841F6, 0xE35D49C2,
     0x19D072E7, 0xB5B515D0, 0x455DE0BA, 0xDE0C561B, 0xB256C54E,
     0x96DF7102, 0x9C05D4EE, 0xA74445FA, 0x742B5A74, 0x06F588FA,
     0x84AD997A, 0x06C6BE55, 0x53D2750D, 0x1A5DF072, 0x42DBCB54,
     0x2F0E18F3, 0x44A9D3F4, 0x4E41D97A, 0x97803F5C, 0x574FC614,
     0x92BD8366, 0x88C63FD2, 0xB58B29B4, 0x43F62A69, 0x917B85CC,
     0x0DB3E1A6, 0x496BC794, 0x91759B86, 0x48F93E89, 0x0A9678CF,
     0x2F374D90, 0x3702F635, 0x97C52AB2, 0x471DE49A, 0xA7E64531,
     0xC119CDEC, 0x96D93C1C, 0x929A5D3A, 0x244DF978, 0xA3EB09B8,
     0xE2127B4E, 0xA13DE9B0, 0x469D6F42, 0xA73D185C, 0x9BF21945,
     0x87CF05B4, 0xC46967CC, 0x0F30E6B6, 0x2E7D04AE, 0x8DB127B8,
     0xB74314E6, 0x7109EB74, 0xA427B966, 0x54C1DA3E, 0xB34C1E95,
     0x4C6D723C, 0x3B9E0A65, 0x1934BAB3, 0x495E4B99, 0x3457E962,
     0x94F1DC1A, 0xA70B6AB2, 0x317FA1C4, 0x5E1073DA, 0x82AFB546,
     0x8683EB9A, 0xF043657A, 0x0DECB299, 0x2774F154, 0xA17349F4,
     0x96FB03A4, 0x57D5640E, 0x829CCABB, 0xA1AD43B6, 0x291CDAAD,
     0x1859BE8E, 0x86F4B90E, 0x2ACB1D9C, 0x9CD54752, 0x185AF44F,
     0x2B276BC4, 0xA17361DC, 0xC3698BE4, 0xB56BB504, 0xD9615176,
     0x53DE1951, 0xD40CB2AF, 0x94C78EA9, 0x178BBE06, 0x919DB970,
     0x4F674A85, 0x1A3DCF44, 0x1BC5BD12, 0x90ACED72, 0x889E6F16,
     0xAE165D1C, 0x16D0AE8F, 0x1E907C5E, 0x43986E9E, 0xB5681DB4,
     0xCF19149E, 0x73734D22, 0x706D43B6, 0xD5293B94, 0x354D9AC6,
     0x532EAA56, 0x1D9AA9C9, 0xB3DA510D, 0x1F589627, 0x855CB95A,
     0x9129AF36, 0x8FCA522D, 0x8B21B2EE, 0x987FD094, 0x9DE5206D,
     0xE6103B6E, 0x3674A0AF, 0xC8212FFC, 0x6D70613E, 0xC64A5CD6,
     0x91A93B6C, 0x226FAA66, 0xB319D52C, 0x23EF630C, 0x8DC76616,
     0x95B342AE, 0x50C57DCC, 0x33ED1B44, 0x1B4BF164, 0x5E451EA6,
     0xD68F41C6, 0x3E464CD3, 0x229AE1BE, 0x927BF122, 0x1D77E026,
     0xB5AC2396, 0x38E54D3C, 0xD4292EBC, 0xC72B236C, 0x1DA9A35A,
     0xAA6D41E6, 0x9721D56A, 0xCF495629, 0x0A74CBDA, 0xC322EB65,
     0x447DD4C6, 0x96498BF8, 0x9D47893C, 0x9A19A57A, 0x2E4BD456,
     0x3FBE0843, 0x1586EB8E, 0xB5091FCC, 0x0AA5FC1E, 0x8C423EF6,
     0x4F1C6273, 0x4C584EF6, 0x1844DEF3, 0x3DCD241E, 0xC1E649F1,
     0x2C6B8EA9, 0xB58B6296, 0x707A39C5, 0x6B1A69E1, 0x8D1B63A6,
     0x1DA57561, 0x70CB7A29, 0x96E32E34, 0x0A9CB5E9, 0x0BFB20F4,
     0x5C47A696, 0x13ADBC26, 0xA2536E3C, 0xDF0D41B2, 0x07B9B94A,
     0x4C4F7398, 0x95338BD4, 0xC12FE998, 0x15AE85D9, 0x85ED915A,
     0x936FA486, 0xC65F0696, 0x0CA3D9DA, 0x9D3B1586, 0xAB372916,
     0x50E760FC, 0x1D21ABF8, 0x5A3C7076, 0xF4497632, 0x72D274A5,
     0x2C732EB4, 0xB319FD04, 0xA652BE25, 0x53CF7510, 0x1C97E1C6,
     0x93C9DD42, 0x3CAC34CB, 0x3D31AD46, 0x996F0B89, 0x3655BC1A,
     0x8E47ACC6, 0xB041D6DE, 0xAD5B6161, 0x1979B0F4, 0x84CD6F8A,
     0x4733A9AC, 0x83D9F431, 0x77C3059A, 0x95CE0F52, 0x3B2F0CD2,
     0xAE790E1C, 0x9BF602A5, 0x496D4BCC, 0xB2A71B34, 0x2F1A296D,
     0xC50B9A9E, 0xAB196BD0, 0x1E954AAE, 0x87932D9A, 0x9EA134F2,
     0x11CC9FC3, 0x714438FB, 0xD713B216, 0xA350B93E, 0x6A3A4B56,
     0x3E4A7075, 0x85B9E5D0, 0x9E4338B6, 0x387F50B4, 0x42C56FCC,
     0x57917694, 0x1586DEE1, 0x741B29EA, 0xA32B0EF2, 0x8B599C8B,
     0xCC5E7129, 0x961BFB02, 0x27F9821E, 0x96FD5302, 0x41EE83E9,
     0x55A5D469, 0x5B5D3419, 0xA66AD172, 0x32EA473C, 0x9ED05369,
     0x0ECD91BA, 0x720A5F65, 0x1CA73BA4, 0x1DA534E3, 0xAB79072C,
     0x4C5EB643, 0x8D5D9926, 0xF0250FBA, 0x1795DE11, 0x0E299DF8,
     0xC74D2AA6, 0x3F41578A, 0xB06EE50E, 0x51BC486F, 0x76230BB6,
     0x26F23792, 0x716964B9, 0x1DFA9485, 0xAC2E6A1E, 0x30EB3C66,
     0xF0305DE9, 0x1877BD90, 0x917DB066, 0x971AA0F3, 0x17C69E49,
     0x3A776446, 0x0BBC25E3, 0x8F832F2A, 0xD0099EBE, 0x5D317A32,
     0x096ADB9C, 0x571C868F, 0x926CAD5A, 0xA4391CEE, 0xD1E86656,
     0x73890DBA, 0x70E67E11, 0xC6612DEA, 0x8799654E, 0xCE4C5786,
     0x9E4358D6, 0xA74D4C9C, 0xA549D25E, 0x74E1473A, 0xF27023F2,
     0x9D0D8CD9, 0x04ECD1FC, 0xD41978BC, 0x3D674C54, 0x71A94D72,
     0x8D7E40C7, 0x978E253A, 0x78271AE9, 0x74355C9C, 0xC343F31C,
     0x764609FC, 0x25983BCB, 0x742B8E36, 0xD48B65CC, 0x25EC1A3B,
     0xA5CE093E, 0x9E6531CC, 0x12B1FC2E, 0x428BAE9E, 0x53F5620E,
     0x1F34D983, 0x57CD0A3C, 0x28697E72, 0x93BC0F1A, 0x384C4DFC,
     0x4EDE7205, 0x4E1D4CBC, 0x3C2E895E, 0x6609F752, 0x5F181F92,
     0x90DF53B0, 0x3D264CD6, 0x86D6E075, 0x42DB65D8, 0xD6D0616E,
     0x96D53570, 0x94B0CBE6, 0x83B2F4A5, 0x5969B346, 0x8C276E99,
     0x37B531C2, 0x6A477592, 0x2A7A2C3E, 0xB54EC259, 0xB151A5E6,
     0x1385FF90, 0x7141F83E, 0x5A122FCB, 0xA5F4059E};
```

Appendix E

RNS counters

Counter	Width	Feedback Function	All Possible Period Lengths
0	11	0xA1705DB9	1009, 1039
1	11	0x947A15CB	997, 1051
2	11	0x85A1BDC3	919, 1129
3	11	0xD0C274F5	877, 1171
4	11	0xEA4308DF	811, 1237
5	11	0xF031C87D	769, 1279
6	11	0xB8CA56E1	757, 1291
7	11	0xD384CB59	751, 1297
8	11	0xC479F503	727, 1321
9	11	0xA6945DE1	619, 1429
10	11	0xBA8E0397	601, 1447
11	11	0xE378A90D	577, 1471
12	11	0xFA612475	499, 1549
13	11	0xD18453AF	463, 491, 523, 571
14	11	0xC5A026F7	443, 449, 557, 599
15	11	0x86E4127F	439, 1609
16	10	0xF4AC89E1	503, 521
17	10	0x963E41A7	467, 557
18	10	0xBA043D8F	461, 563
19	10	0xA1FE08D3	431, 593
20	10	0xC4D835E9	383, 641
21	10	0xF90E8C63	347, 677
22	10	0xE8F463A1	281, 743
23	10	0xBF2C980D	263, 761
24	10	0x971D843B	251, 773
25	10	0xC1B0F547	227, 797
26	10	0x8AF17435	197, 827
30	10	0xE6247CB1	191, 239, 241, 353
27	10	0xD270F389	167, 857
28	10	0xF7C43681	163, 211, 313, 337
29	10	0xD34C0FA9	151, 223, 283, 367
31	10	0xE6AD9C81	149, 233, 293, 349

Appendix F

Input bit permutations

```
Vp[128][5] = {
     {0,1,2,3,4}, {1,0,2,3,4}, {0,2,1,3,4}, {2,0,1,3,4}, {1,2,0,3,4},
     {2,1,0,3,4}, {0,1,3,2,4}, {1,0,3,2,4}, {0,3,1,2,4}, {3,0,1,2,4},
     {1,3,0,2,4}, {3,1,0,2,4}, {0,2,3,1,4}, {2,0,3,1,4}, {0,3,2,1,4},
     {3,0,2,1,4}, {2,3,0,1,4}, {3,2,0,1,4}, {1,2,3,0,4}, {2,1,3,0,4},
     {1,3,2,0,4}, {3,1,2,0,4}, {2,3,1,0,4}, {3,2,1,0,4}, {0,1,2,4,3},
     {1,0,2,4,3}, {0,2,1,4,3}, {2,0,1,4,3}, {1,2,0,4,3}, {2,1,0,4,3},
     {0,1,4,2,3}, {1,0,4,2,3}, {0,4,1,2,3}, {4,0,1,2,3}, {1,4,0,2,3},
     {4,1,0,2,3}, {0,2,4,1,3}, {2,0,4,1,3}, {0,4,2,1,3}, {4,0,2,1,3},
     {2,4,0,1,3}, {4,2,0,1,3}, {1,2,4,0,3}, {2,1,4,0,3}, {1,4,2,0,3},
     {4,1,2,0,3}, {2,4,1,0,3}, {4,2,1,0,3}, {0,1,3,4,2}, {1,0,3,4,2},
     {0,3,1,4,2}, {3,0,1,4,2}, {1,3,0,4,2}, {3,1,0,4,2}, {0,1,4,3,2},
     {1,0,4,3,2}, {0,4,1,3,2}, {4,0,1,3,2}, {1,4,0,3,2}, {4,1,0,3,2},
     {0,3,4,1,2}, {3,0,4,1,2}, {0,4,3,1,2}, {4,0,3,1,2}, {3,4,0,1,2},
     {4,3,0,1,2}, {1,3,4,0,2}, {3,1,4,0,2}, {1,4,3,0,2}, {4,1,3,0,2},
     {3,4,1,0,2}, {4,3,1,0,2}, {0,2,3,4,1}, {2,0,3,4,1}, {0,3,2,4,1},
     {3,0,2,4,1}, {2,3,0,4,1}, {3,2,0,4,1}, {0,2,4,3,1}, {2,0,4,3,1},
     {0,4,2,3,1}, {4,0,2,3,1}, {2,4,0,3,1}, {4,2,0,3,1}, {0,3,4,2,1},
     {3,0,4,2,1}, {0,4,3,2,1}, {4,0,3,2,1}, {3,4,0,2,1}, {4,3,0,2,1},
     {2,3,4,0,1}, {3,2,4,0,1}, {2,4,3,0,1}, {4,2,3,0,1}, {3,4,2,0,1},
     {4,3,2,0,1}, {1,2,3,4,0}, {2,1,3,4,0}, {1,3,2,4,0}, {3,1,2,4,0},
     {2,3,1,4,0}, {3,2,1,4,0}, {1,2,4,3,0}, {2,1,4,3,0}, {1,4,2,3,0},
     {4,1,2,3,0}, {2,4,1,3,0}, {4,2,1,3,0}, {1,3,4,2,0}, {3,1,4,2,0},
     {1,4,3,2,0}, {4,1,3,2,0}, {3,4,1,2,0}, {4,3,1,2,0}, {2,3,4,1,0},
     {3,2,4,1,0}, {2,4,3,1,0}, {4,2,3,1,0}, {3,4,2,1,0}, {4,3,2,1,0},
     {3,4,0,1,2}, {2,4,1,0,3}, {1,4,2,0,3}, {4,3,2,1,0}, {3,2,1,0,4},
     {2,1,0,4,3}, {1,0,4,3,2}, {0,4,3,2,1}};
```

Appendix G

Output Combiners

VEST-4 output combiner:

o	*xa*	*xb*	*xc*	*xd*	*xe*	*xf*
0	18	25	33	67	79	81
1	29	41	46	51	63	64
2	15	16	19	23	40	48
3	35	47	57	72	76	78

VEST-16 output combiner:

o	*xa*	*xb*	*xc*	*xd*	*xe*	*xf*
0	42	60	79	147	162	314
1	29	109	128	144	156	211
2	30	159	206	225	260	318
3	14	45	77	210	275	322
4	22	33	43	81	305	324
5	66	138	140	182	240	313
6	25	36	185	224	232	316
7	19	20	107	133	212	257
8	52	88	101	115	261	325
9	28	98	125	214	221	259
10	41	191	263	284	302	312
11	35	37	241	252	291	292
12	50	168	198	228	253	288
13	47	53	71	84	276	323
14	55	131	229	256	287	308
15	17	106	130	172	181	277

VEST-32 output combiner:

o	*xa*	*xb*	*xc*	*xd*	*xe*	*xf*
0	235	269	293	319	344	375
1	159	182	312	376	418	437
2	55	98	193	266	281	551
3	177	244	349	374	432	457
4	27	94	125	354	560	561
5	210	346	347	363	424	428
6	11	277	285	313	334	483
7	31	117	191	194	256	366
8	63	99	196	241	362	540
9	21	42	144	405	536	557
10	239	373	415	422	450	515
11	40	254	338	429	486	491
12	10	133	292	335	407	481
13	39	431	471	489	525	575
14	126	263	323	343	519	553
15	106	139	147	382	499	571
16	104	134	306	341	348	357
17	102	216	361	416	454	533
18	53	146	178	234	443	531
19	107	152	198	261	417	498
20	37	129	170	190	351	355
21	69	92	243	395	419	508
22	110	115	464	530	543	573
23	22	212	467	526	532	548
24	108	233	315	468	574	579
25	17	84	321	408	449	555
26	127	128	438	445	487	496
27	89	207	273	369	409	563
28	73	131	138	336	480	562
29	26	119	169	189	214	413
30	75	97	205	232	448	501
31	12	20	290	305	488	521