ॐ भूर्भुवस्व तत् सवितूर्वरेण्यम् भर्गो देवस्य धीमहि धियो यो न प्रचोदयात्

VEST Ciphers

(eSTREAM Phase II submission)

Sean O'Neil [sean@cryptolib.com], Benjamin Gittins, Howard A. Landman

January 2007

Abstract: VEST (Very Efficient Substitution-Transposition) is a set of families of counter-assisted substitution-transposition ciphers designed and optimised specifically for ASIC and FPGA hardware. VEST ciphers provide fast scalable keystream generation, authenticated encryption and collision-resistant hashing at a very low cost in area and power consumption. All VEST ciphers support variable-length keys and IVs and are naturally very slow in software. Cores of VEST ciphers can be viewed as light-weight T-functions or large bijective nonlinear feedback shift registers (NLFSRs) with massively parallel feedback, assisted by a nonlinear residue number system (RNS) based counter with a very long period. Four VEST cipher family trees are introduced: 80-bit secure VEST4-80, 128-bit secure VEST8-128, 160-bit secure VEST16-160 and 256-bit secure VEST32-256, returning 4 to 32 bits of output per clock cycle while occupying ~3K to ~28K ASIC gates.

1. Introduction

Most modern hardware-dedicated stream ciphers and hash functions are built on well-understood linear feedback shift registers (LFSRs) with nonlinearity introduced only by stateless nonlinear combiners and/or irregular clocking. Purely nonlinear feedback shift registers (NLFSRs), especially the ones with parallel feedback including word-based NLFSRs, are an attractive choice for makers of stream ciphers. Unfortunately, designers have to face the open problems of constructing large NLFSRs with known or predictable periods or determining the shortest, the longest or the average periods of the existing large NLFSRs, the problems that are also computationally infeasible. On the contrary, it is trivial to construct an LFSR with a known and predictable period length. Output of LFSRs is highly predictable and the history shows that linear components are nearly completely dismissed by the attackers as not contributing much to the ciphers' security. Therefore a careful code-maker would exclude bits of the cipher state updated linearly from the calculation of the amount of entropy allowed to be returned as output on every round and would try to maximise the size of the cipher state updated nonlinearly. The marriage of both components has become a popular construction for stream ciphers: large LFSRs are used to guarantee a certain minimal period length, and sufficiently iterated NLFSRs or stateless nonlinear combiners introduce the essential nonlinearity and high algebraic degree into the relationships between keystream bits. In that regard VEST ciphers have nearly the opposite structure: they rely on a large nonlinear counter, a large nonlinear core and they have a small linear output combiner.

VEST ciphers rely primarily on their large core accumulators updated by a massively parallel nonlinear substitution-transposition network constructed carefully enough to maintain its bijective (reversible) operation while ensuring very high diffusion rate. The core accumulators are supported by a set of small NLFSR-based counters with a long combined period. A number of such easy to construct small (40-bit or smaller) NLFSRs with guaranteed periods can be combined into a single set with its total period calculated the same way as the period of any RNS counter. A small number of such short NLFSRs with co-prime periods can be used to construct nonlinear RNS counters with arbitrarily long periods at a very low cost. Such RNS counters are as cheap in

hardware as LFSRs, but their nonlinearity allows the cipher to rely on the counter state as a set of independent variables that cannot be easily predicted or approximated with reasonably short LFSRs.

The simplest and the most popular way to construct bijective (reversible) bitwise NLFSRs is by linearly combining each bit with a nonlinear Boolean function of a number of previous bits in the stream stored in the NLFSR state. Such structured feedback with a single Boolean function results in exploitable patterns in the polynomial relationships between output bits at certain distances from each other. Those patterns can be eliminated by updating each bit of the NLFSR state with a different Boolean function with a different set of inputs and by updating them all at the same time, in other words by adding parallel feedback. Such NLPFSRs must satisfy only one condition to be bijective: only the previous bits in the stream stored in the NLPFSR state can be used as inputs into each nonlinear Boolean function except for the parts where bijectivity is achieved by other means (via bijective NxN S-boxes, arithmetic operations, etc) [22].

A large number of chosen and desired factors must be considered by the code-maker when constructing stream ciphers. Balanced proportions must be maintained between cipher security, its security margin, size of its state, its input width, its output width, amount of feedback, diffusion rate, perfect diffusion rate, width of the output combiner functions, amount of state reserved for the counter to ensure a certain period or to be used in the counter mode or both, numbers of sealing rounds after keying, after loading an IV, before producing a MAC and between outputs, and the algebraic degree, nonlinearity, correlation immunity, density and logic depth of all the functions. Convenience of the key, the IV, the input, the output and the cipher state sizes must also be taken into account, preferably also allowing secure use of variable-size and related keys and IVs and low-entropy IVs. Other factors such as scalability, personalisation, potential need for high clock speed to match other parts of the circuit, low register fan-out, low wire latency, uniform logic depth, efficiency on 4-to-1 and 6-to-1 FPGA architectures, logic-to-register ratio in FPGA, efficient D-Type register handling particularly during initialisation, balancing power consumption between registers and combinatorial logic, potential unavailability or high cost of RAM, complexity of the finite state machine logic, potential need for other cryptographic components implemented on the same circuit, potential need to unroll the cipher logic to reduce clock speed and sensitivity to glitch attacks and side channel analysis must also be considered when constructing hardware ciphers.

VEST ciphers are addressing an immediate pressing need for secure authenticated ciphers in the high-volume commodity hardware market that can not be currently serviced using the existing seriously compromised primitives such as A5x [3], DST [4], E0 [5], HDCP [6], ORYX [7], VSC [8], [9] or MD5 [10].

Before describing the VEST cipher families, let us recall the famous Auguste Kerckhoff's 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 in the 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 more than 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 rekeying 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.

The second desideratum is a call for secure family keying. It does not oppose keeping the details of the exact substitutions and transpositions in a cryptographic device secret. It states that knowledge of those details by an attacker should not affect other devices and other products implementing other cipher families. In the same way, knowledge of the secret key unique to one device should not affect other devices and their users.

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 or the API. 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. VEST ciphers are too large to type by hand and they were designed with uniform structure to allow for easy automated creation of the source code.

VEST ciphers cover all major hardware (FPGA and ASIC) applications, from minimum-area medium-security low-cost RFID encryption and authentication to high-speed high-security encryption and authentication applications. Other family trees of VEST ciphers with the same properties as proposed here can be tailored for each particular application. A highly skilled cryptologist can do it fairly quickly by carefully adjusting the design variables, but it should not be attempted by end-users or cryptographic developers.

1.1 Keyed Family Variants

The four root cipher families presented here are referred to as VEST4-80, VEST8-128, VEST16-160 and VEST32-256. Each of the four family trees of VEST ciphers supports family keying to generate other independent cipher families. Family keying has several important cryptographic applications that include support for applications requiring

anti-cloning and anti-emulation protection, support for unique transformations for protocols employing strict Fail-Stop policies, and for applications requiring proprietary ciphers. The difference between any two ciphers generated with different family keys must be sufficient to ensure that successful reverse engineering of one cipher family does not compromise cipher families generated with other family keys in any way.

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 can be made 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 by using standard cells and wires. 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 customised on a per-chip basis by direct eBeam write. Several manufacturers offer this kind of technology, including eASIC, Toshiba, UMC and ST Microelectronics. In general, the hardwired approaches will have higher performance, lower area, and lower power than the reprogrammable ones.

The systems designed to communicate with more than one cipher family must store the family key and the member key for each device. The family key may be a publicly known variable or a part of the cipher's symmetric secret key. If the family key is kept secret, it leaves the attacker with the only options either to 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 described in sections 3.1 and 3.2 provides a standard method to generate cipher families with unique substitutions and unique counters with different co-prime periods. Any chosen VEST cipher with any desired security rating can be used to generate keyed families for VEST ciphers of other sizes and security ratings by following the family keying process.

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 16 NLFSRs with co-prime period lengths. The counter diffusor is a set of 6-bit wide linear combiners with feedback that compress outputs of the 16 counters into 10 bits. The core accumulator is a balanced light-weight T-function accepting 10 bits of the counter diffusor as input. The output combiner is a set of 6-bit wide linear combiners. There are no known or intentional weaknesses, backdoors or shortcuts in VEST ciphers' structure, transpositions, combiners or update functions.

The following parameters and variables are used in VEST 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 – key length in bits
f – set of nonlinear feedback functions for the core accumulator
g_i – set of nonlinear feedback functions for the RNS counters
H – hash width in bits
i – RNS counter index
j – bit or IV index
k – input bits: key, IV or data
L – message length
M – cipher output width in bits
n – keying (IV loading) step index
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
Ra – first round of the current mode of operation
Rb – first round of the next mode of operation
R_n – number of rounds after keying or IV loading step n
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 – 1024 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

NLPFSR accumulators in VEST ciphers are easier viewed as substitution-transposition networks. Their substitution layer is a set of W nonlinear 6-to-1 functions forming a W-bit wide S-box. It's followed by a transposition chosen to maximise diffusion rate.

In the substitution layer, the least significant five bits of the accumulator state are updated by a bijective 5x5 S-box and are linearly combined with the first five of the ten accumulator input bits on each round. The next five bits of the accumulator state are linearly combined with one of the remaining five accumulator input bits and with a nonlinear function of four other bits of the accumulator state. In authenticated encryption (AE) mode of operation, the next M bits of the accumulator state are linearly combined with bits of the ciphertext feedback state and with a nonlinear function of four other bits of the accumulator state. All the 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 nonlinear function of five other bits in the accumulator.

Appendixes A to E provide bit permutations and nonlinear feedback functions for VEST4-80, VEST8-128, VEST16-160 and VEST32-256 cipher family trees. Additional pictures clarifying the structure of VEST ciphers can be found in appendix I.

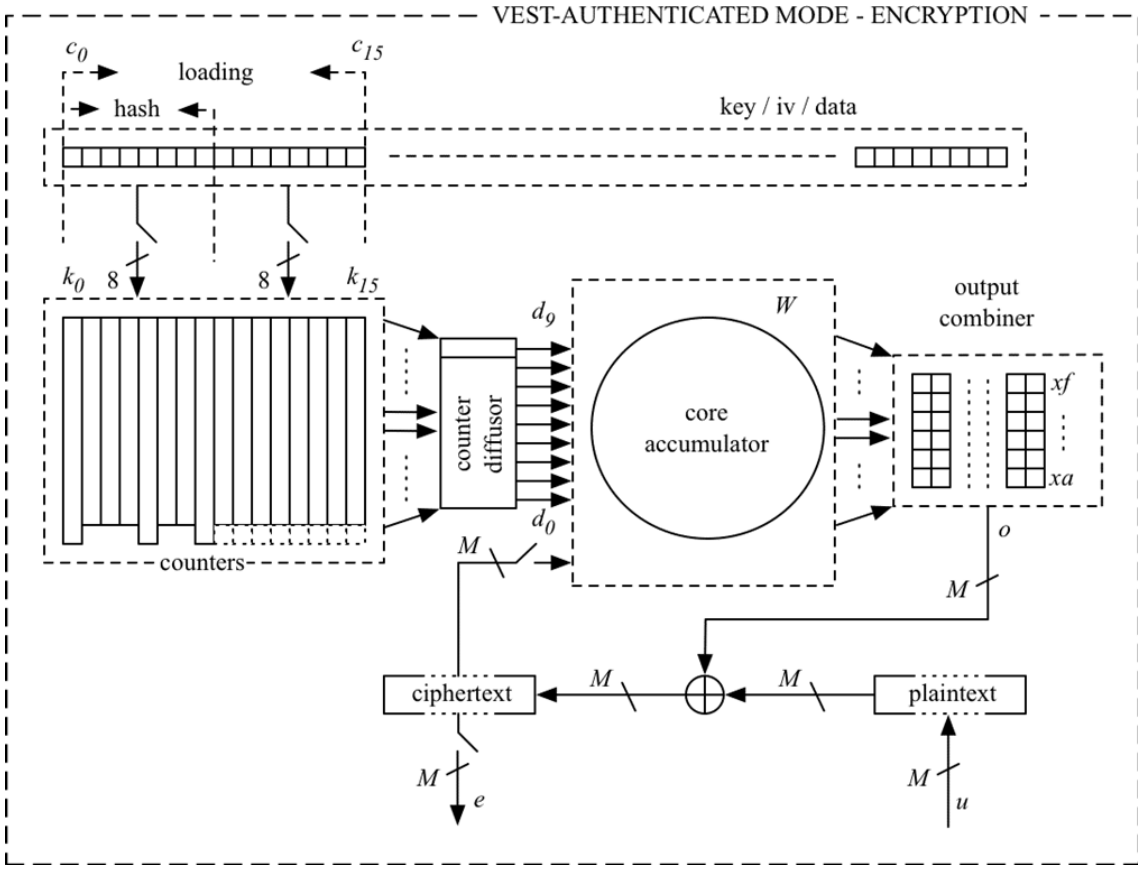


Fig 1. High-level VEST structure

2.1 RNS Counters

C-bit wide VEST cipher counter consists of 16 separate 10-bit and 11-bit wide RNS counters *c* implemented as bijective NLFSRs. Each of the NLFSRs is updated with a different nonlinear feedback function with six inputs chosen so that it produces two distinct loops with unique period lengths that have no common divisors with any other counter. Such NLFSRs when bijectively combined together form a counter with a total period being a multiple of the individual periods of all 16 NLFSRs.

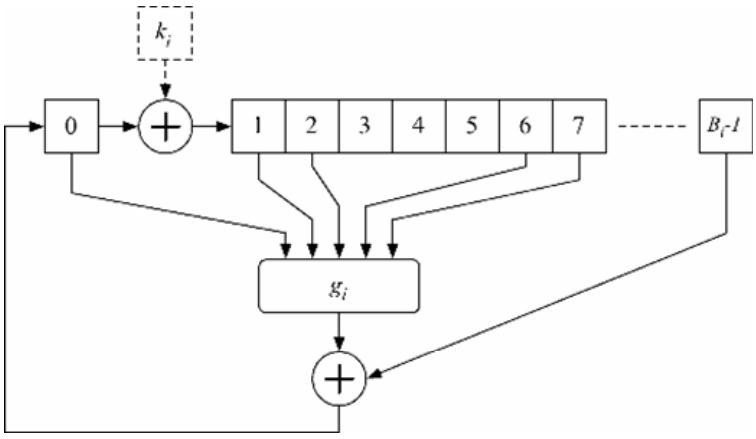


Fig 2. NLFSR counter

The 16 RNS counters have B_i-bit (10-bit or 11-bit) states c_i updated by their respective nonlinear 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-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. See appendix F for the list of RNS counters feedback functions and their period lengths. 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 cipher families.

In VEST4-80 and VEST8-128 ciphers, 13 counters are 10-bit wide and 3 are 11-bit wide, forming a 163-bit counter state. In VEST16-160 and VEST32-256 ciphers, 5 counters are 10-bit wide and 11 are 11-bit wide, forming a 171-bit counter state.

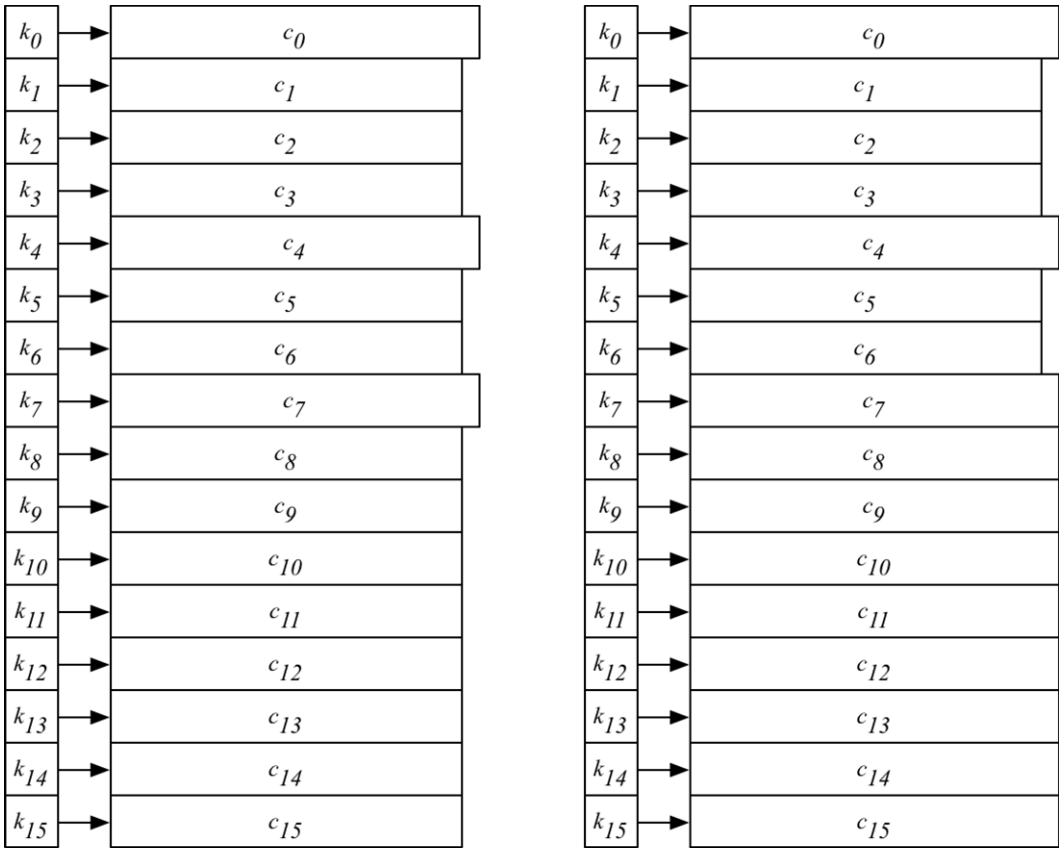


Fig 3. RNS counters (left: 163-bit, VEST4/VEST8; right: 171-bit, VEST16/VEST32)

In keying mode, all 16 counters accept 16 key bits as their inputs linearly combined with bit *0* of each counter and stored in bit *1*. In hashing mode, eight of the sixteen counters accept eight key bits as their inputs linearly combined with bit *0* of each counter, which leaves eight of the counters impossible to control with the input stream.

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

In keying mode:

$$c_i^{r+1}{}_0 = g_i\,(c_i^r{}_0, c_i^r{}_1, c_i^r{}_2, c_i^r{}_6, c_i^r{}_7) + c_i^r{}_{B[i]-1},$$
$$c_i^{r+1}{}_1 = c_i^r{}_0 + k^r{}_i,$$
$$c_i^{r+1}{}_j = c_i^r{}_{j-1},\ 2 \le j < B_i,$$
$$0 \le i < 16;$$

In hashing mode:

$$c_i^{r+1}{}_0 = g_i\,(c_i^r{}_0, c_i^r{}_1, c_i^r{}_2, c_i^r{}_6, c_i^r{}_7) + c_i^r{}_{B[i]-1},$$
$$c_i^{r+1}{}_1 = c_i^r{}_0 + k^r{}_i,$$
$$c_i^{r+1}{}_j = c_i^r{}_{j-1},\ 2 \le j < B_i,$$
$$0 \le i < 8;$$
$$c_i^{r+1}{}_0 = g_i\,(c_i^r{}_0, c_i^r{}_1, c_i^r{}_2, c_i^r{}_6, c_i^r{}_7) + c_i^r{}_{B[i]-1},$$
$$c_i^{r+1}{}_j = c_i^r{}_{j-1},\ 1 \le j < B_i,$$
$$8 \le i < 16;$$

In sealing, keystream generation and authenticated encryption modes:

$$c_i^{r+1}{}_0 = g_i\,(c_i^r{}_0, c_i^r{}_1, c_i^r{}_2, c_i^r{}_6, c_i^r{}_7) + c_i^r{}_{B[i]-1},$$
$$c_i^{r+1}{}_j = c_i^r{}_{j-1},\ 1 \le j < B_i,$$
$$0 \le i < 16;$$

2.2 Counter Diffusor

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

$$d^{r+1}{}_0 = d^r{}_1 + c_1^r{}_1 + c_4^r{}_1 + c_5^r{}_1 + c_{11}^r{}_1 + c_{13}^r{}_1 + 1;$$
$$d^{r+1}{}_1 = d^r{}_2 + c_0^r{}_1 + c_2^r{}_1 + c_6^r{}_1 + c_8^r{}_1 + c_{14}^r{}_1;$$
$$d^{r+1}{}_2 = d^r{}_3 + c_3^r{}_1 + c_4^r{}_1 + c_7^r{}_1 + c_{10}^r{}_1 + c_{15}^r{}_1;$$
$$d^{r+1}{}_3 = d^r{}_4 + c_0^r{}_1 + c_3^r{}_1 + c_5^r{}_1 + c_9^r{}_1 + c_{12}^r{}_1;$$
$$d^{r+1}{}_4 = d^r{}_5 + c_1^r{}_1 + c_4^r{}_1 + c_6^r{}_1 + c_{12}^r{}_1 + c_{15}^r{}_1 + 1;$$
$$d^{r+1}{}_5 = d^r{}_6 + c_0^r{}_1 + c_7^r{}_1 + c_9^r{}_1 + c_{13}^r{}_1 + c_{14}^r{}_1;$$
$$d^{r+1}{}_6 = d^r{}_7 + c_7^r{}_1 + c_8^r{}_1 + c_{11}^r{}_1 + c_{14}^r{}_1 + c_{15}^r{}_1;$$
$$d^{r+1}{}_7 = d^r{}_8 + c_2^r{}_1 + c_5^r{}_1 + c_6^r{}_1 + c_{10}^r{}_1 + c_{12}^r{}_1 + 1;$$
$$d^{r+1}{}_8 = d^r{}_0 + c_0^r{}_1 + c_3^r{}_1 + c_7^r{}_1 + c_8^r{}_1 + c_9^r{}_1 + 1;$$
$$d^{r+1}{}_9 = d^r{}_9 + c_8^r{}_1 + c_{10}^r{}_1 + c_{12}^r{}_1 + c_{13}^r{}_1 + c_{15}^r{}_1 + 1;$$

Important Note: The original cipher specification and source code contained a typo in the feedback of d_6, which has been corrected in this version of the paper: it must be updated by the output of counters 7, 8, 11, 14 and 15, not 1, 8, 11, 14 and 15.

2.3 Accumulator

In the accumulator state x, the least significant five bits x_0 to x_4 are used as inputs into five nonlinear 5-to-1 Boolean functions f_0 to f_4 that form a 5x5 S-box. Outputs of those functions are 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 nonlinear 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. Bits x_{10} to x_{M+9} are linearly combined with outputs of the next M nonlinear 4-to-1 feedback functions f_{10} to f_{M+9} and in AE mode also with five ciphertext feedback bits and fed back into bits x_{p10} to $x_{p[M+9]}$ respectively. All other bits x_j in the VEST accumulator state are linearly combined with outputs of nonlinear 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}.

For the accumulator feedback process 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 completely arbitrary, although a set of linearly independent strong balanced nonlinear Boolean functions is carefully chosen for all VEST ciphers. The subsequent permutation of the accumulator bits also does not affect its bijectivity. 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.

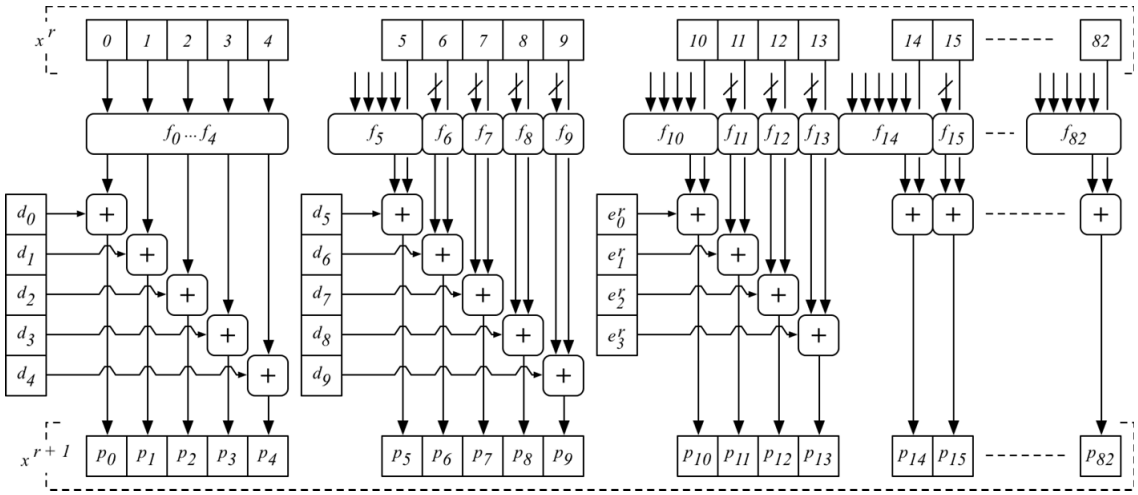


Fig 4. Partial illustration of the core accumulator

To summarise operation of the core accumulator without ciphertext feedback:

$$x^{r+1}_{pj[5]} = f_j\,(x^r_0, x^r_1, x^r_2, x^r_3, x^r_4) + d^r_j,\ 0 \le j < 5;$$
$$x^{r+1}_{pj[5]} = f_j\,(x^r_{pj[0]}, x^r_{pj[1]}, x^r_{pj[2]}, x^r_{pj[3]}, x^r_{pj[4]}) + x^r_j + d^r_j,\ 5 \le j < 10;$$
$$x^{r+1}_{pj[5]} = f_j\,(x^r_{pj[0]}, x^r_{pj[1]}, x^r_{pj[2]}, x^r_{pj[3]}, x^r_{pj[4]}) + x^r_j,\ 10 \le j < W;$$

Appendixes A, B, C, D and E provide the bit permutations p and the nonlinear feedback functions f for the VEST family tree ciphers.

2.4 Output Combiner

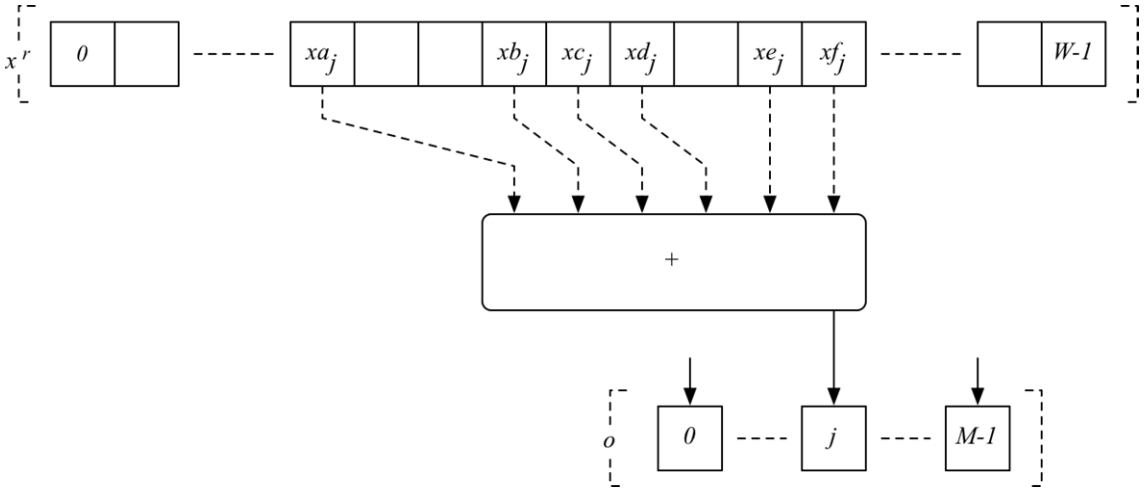


Fig 5. Partial illustration of the output combiner

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 summarise 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 initialisation rounds:

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

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 H.

3. VEST Process

VEST cipher families can be identified according to their security rating, width of the core accumulator W or the width of the output M. This paper presents four VEST families of ciphers: an 80-bit secure family with an 83-bit wide accumulator and 4-bit wide output, a 128-bit secure family with a 211-bit wide accumulator and 8-bit wide output, a 160-bit secure family with a 331-bit wide accumulator and 16-bit wide output and a 256-bit secure family with a 587-bit wide accumulator and 32-bit wide output.

The included VEST4-80, VEST8-128, VEST16-160 and VEST32-256 ciphers are the root cipher families of their respective family trees. Root cipher families can be used to generate keystream, to encrypt or hash messages, or to generate a vast number of keyed families of VEST ciphers routinely in a standard interoperable manner.

If a more standardised cipher implementation is required, efficient on the widest range of platforms, we recommend implementation of one of the root VEST cipher families according to this specification. A root cipher family can be implemented directly as authenticated or unauthenticated stream ciphers or [keyed or unkeyed] collision-resistant cryptographic hash functions, preferably executed in CTR or CTR-AE mode.

3.1 Modes of Operation

VEST ciphers operate in five different base modes described below: keying, hashing, sealing, keystream generation and authenticated encryption. The higher-level processes of generation of cipher families, encryption and hashing of data streams (or data blocks for CTR, CTR-AE or CTR-HASH modes) are described in sections 3.2 to 3.6.

3.1.1) Keying Mode

Prior to all keystream generation, data encryption or hashing, the cipher is executed in keying mode for $R_0 = F+16$ rounds, where F is the length of the key in bits, loading the key bits 16 bits per round, thus loading a key prepended with 15 zero bits and followed by a single bit 1 and 15 zero bits. Key loading begins with the least significant bit of the key loaded into the counter 15, sliding the 16-bit window by one bit on each round until the single bit 1 attached at the end of the key is loaded into the counter 0:

$$c_i^{r+1}{}_0 = g_i\,(c_i^r{}_0, c_i^r{}_1, c_i^r{}_2, c_i^r{}_6, c_i^r{}_7) + c_i^r{}_{B[i]-1},$$
$$c_i^{r+1}{}_1 = 0, \text{ if } r+i < 15,$$
$$c_i^{r+1}{}_1 = c_i^r{}_0 + k_{r+i-15}, \text{ if } 15 \le r+i < F+15,$$
$$c_i^{r+1}{}_1 = 1, \text{ if } r+i = F+15,$$
$$c_i^{r+1}{}_1 = 0, \text{ if } F+15 < r+i,$$

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

The streamlined keying process dramatically simplifies the control logic for both the key storage and the cipher operation. It also reduces the circuit area and the complexity involved in developing regression test suites. Special attention should be paid to the key storage and key interface circuitry to ensure minimal information leakage.

3.1.2) Hashing Mode

This mode is used for keyed and unkeyed hashing of data and for loading IVs. In this mode executed between rounds R_a and R_b, the input data is hashed 8 bits per round:

$c_i^{r+1}{}_0 = g_i\,(c_i^r{}_0, c_i^r{}_1, c_i^r{}_2, c_i^r{}_6, c_i^r{}_7) + c_i^r{}_{B[i]-1}$,
$c_i^{r+1}{}_1 = c_i^r{}_0 + k_{(r-Ra)*8+i}$,
$c_i^{r+1}{}_j = c_i^r{}_{j-1}$, $2 \le j < B_i$, $0 \le i < 8$;
$c_i^{r+1}{}_0 = g_i\,(c_i^r{}_0, c_i^r{}_1, c_i^r{}_2, c_i^r{}_6, c_i^r{}_7) + c_i^r{}_{B[i]-1}$,
$c_i^{r+1}{}_j = c_i^r{}_{j-1}$, $1 \le j < B_i$, $8 \le i < 16$;
$x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_0, x^r{}_1, x^r{}_2, x^r{}_3, x^r{}_4) + d^r{}_j$, $0 \le \mathrm{j} < 5$;
$x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j + d^r{}_j$, $5 \le \mathrm{j} < 10$;
$x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j + e^{(r-Ri-1)*M+j-10}$, $10 \le j < 10+M$;
$x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j$, $10+M \le j < W$;
$R_a \le r < R_b$.

The ciphertext feedback state e in the above process is used only during hashing of the 0xFF sealing constant after AE mode (see 3.6.3) and must be set to all 0s otherwise.

3.1.3) Sealing Mode

Cipher sealing is identical to hashing a non-zero constant followed by 31 zero bytes. It is executed after loading a key (3.1.1), after loading IVs, and before producing a hash or a message authentication code. The non-zero constant is selected from the list below:

0x4E (0,1,1,1,0,0,1,0) as bits $k_F..k_{F+7}$
after loading a key used for cipher family generation,
0x2B (1,1,0,1,0,1,0,0) as bits $k_F..k_{F+7}$
after loading a key used for keystream generation or authenticated encryption,
0xB2 (0,1,0,0,1,1,0,1) as bits $k_F..k_{F+7}$
after loading a key used for hashing,
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 $N = \sum width(\mathrm{IV}_j)$, otherwise
0xFF (1,1,1,1,1,1,1,1) as bits $k_L..k_{L+7}$
for all other data inputs.

If sealing the state after the AE mode, the ciphertext feedback state is now reset to all 0s. The cipher is executed for 31 more rounds between rounds R_a+1 and R_a+32 in sealing mode, during which the cipher receives no input and produces no output:

$$c_i^{r+1}{}_0 = g_i\,(c_i^r{}_0, c_i^r{}_1, c_i^r{}_2, c_i^r{}_6, c_i^r{}_7) + c_i^r{}_{B[i]-1},$$
$$c_i^{r+1}{}_j = c_i^r{}_{j-1},\ 1 \le j < B_i,\ 0 \le i < 16;$$
$$x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_0, x^r{}_1, x^r{}_2, x^r{}_3, x^r{}_4) + d^r{}_j,\ 0 \le \mathrm{j} < 5;$$
$$x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j + d^r{}_j,\ 5 \le \mathrm{j} < 10;$$
$$x^{r+1}{}_{pj[5]} = f_j\,(x^r{}_{pj[0]}, x^r{}_{pj[1]}, x^r{}_{pj[2]}, x^r{}_{pj[3]}, x^r{}_{pj[4]}) + x^r{}_j,\ 10 \le j < W;$$
$$R_a{+}1 \le r < R_a{+}32.$$

3.1.4) Keystream Mode

During keystream generation executed between rounds R_a and R_b, the cipher receives no input, and its output bits o are either returned as the generated keystream or applied to the message according to the protocol in place, usually using an XOR operation:

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

3.1.5) Authenticated Encryption (AE) Mode

During AE mode, the ciphertext is fed back into the core accumulator as follows:

[encryption]: $e^{(r-Ri)*M+j} = u^{(r-Ri)*M+j} + o^{(r-Ri)*M+j},\ 0 \le j < M;$
[decryption]: $u^{(r-Ri)*M+j} = e^{(r-Ri)*M+j} + o^{(r-Ri)*M+j},\ 0 \le j < M;$

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

Operation of VEST ciphers in their native authenticated encryption mode has not been changed since its first publication in [12].

3.2 Cipher Family Generation

Besides the four fixed root families, a vast number of keyed families of VEST ciphers can be generated by choosing different feedback functions for each VEST cipher family according to its family key. To generate a keyed cipher family, the corresponding root cipher family is initialised with the given family key and optional IVs, sealed and executed in keystream mode. The first W*17+128 bits of keystream are then used to choose the feedback functions for the new cipher family. Members of each family differ

only by their symmetric member keys. It is also possible to generate sub-families of ciphers by using a keyed cipher family instead of the root cipher family in this process.

3.2.1) Initialisation

First, the cipher state is initialised by filling the core accumulator, the counter diffusor and the sixteen RNS counters with bits 1:

$x^0_j = 1,\ 0 \le j < W.$
$d^0_j = 1,\ 0 \le j < 10;$
$c_i{}^0_j = 1,\ 0 \le i < 16,\ 0 \le j < B_i;$

Cipher initialisation with ones instead of zeroes is chosen to reduce circuit area when synthesised in some ASIC standard cell libraries. The implementations including AE mode must set the "previous round ciphertext" state to all zeroes.

3.2.2) Loading the key

The cipher is executed in keying mode for $R_0 = F+16$ rounds as described in 3.1.1.

3.2.3) Sealing the cipher state after keying

The cipher state is sealed using 0x4E sealing constant as described in 3.1.3 between rounds $R_a=R_0$ and $R_1=R_0+32$. Now 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. This keyed state can be stored and loaded into VEST registers for instant re-keying of the family tree cipher at any time.

3.2.4) Hashing the IVs

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 as described in 3.1.2. The cipher is executed in hashing mode beginning with round R_1 between rounds $Ra=R_n$ and $Rb=R_{n+1}=R_n+width(\mathrm{IV}_{n-1})$ for each of the IVs.

3.2.5) Sealing the cipher state after IV loading

The IV hashing process is finalised by sealing the cipher state with a constant 0xE4 between rounds $R_a=Ri-32$ and Ri as described in 3.1.3.

3.2.6) Producing Keystream

After the total of Ri cipher initialisation rounds, the process begins to return output keystream while executed continuously between rounds $Ra=Ri$ and Rb=W *17+128 in keystream generation mode as described in 3.1.4 releasing M bits of output on every round. The output keystream is used as follows:

3.2.7) Choosing 10-bit counters

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

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

3.2.8) Choosing 11-bit counters

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

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

3.2.9) 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 E. Each Boolean function identified by the index w is swapped with the function identified by the index s:

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

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

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

3.2.10) 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 3.2.8, 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 new cipher family are rearranged in the order specified by t selecting one of the 128 input pin permutations listed in the appendix G:

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

Thus the new cipher family is generated using (W *17 + 128) bits of keystream output. Alternatively, W*17+128 bits of entropy from another source can be used.

3.3 Family Member Initialisation and Keying

Each VEST cipher has its own maximal proven security level, but VEST ciphers can accept keys and IVs of any size. The keying process in VEST ciphers is designed to resist attacks against all kinds of implementations that may use poorly generated, related or shortened keys padded with known information. It is also designed as a fast method of compressing larger keys, so no additional processing of key material is required. For example, a 4096-bit result of a Diffie-Hellman key exchange can be safely loaded into

any VEST cipher directly as it is. It gets hashed and compressed automatically during the keying process. Of course, keys larger than the size of the internal state of the cipher would lose some of their entropy during compression and such cryptosystems cannot be expected to provide more security than the actual security rating of the cipher.

Although all VEST ciphers accept keys of any size, we strongly recommend use of keys at least twice the security rating of the chosen cipher to prevent parallel brute-force [13] and time-memory-data (TMD) trade-off attacks [14] and possible future attacks by quantum computers. That is, VEST4-80 should be initialised with 160-bit keys, VEST8-128 with 256-bit keys, VEST16-160 with 320-bit keys and VEST32-256 with 512-bit keys. If the cryptosystem absolutely must use keys of the same length as the security rating of the cipher, IVs providing the same amount of entropy must be used.

Although block cipher theory seems to ignore the TMTO (TMD trade-off) attacks and parallel brute-force attacks as somehow irrelevant to security, designers of secure crypto-systems must take TMTO attacks into serious consideration. It means loading at least twice as many bits of entropy into the cipher state as the required security rating and choosing a cipher that can support such amount of entropy securely.

Use of keys that are twice as long does not affect circuit area or performance of VEST ciphers and the increase in the number of initialisation rounds is a negligible expense. We believe that 80-bit secure authenticated stream ciphers initialised with 160-bit keys are perfectly affordable in constrained environments such as passive RFID tags.

3.3.1) Initialisation

A keyed cipher family or a root cipher family is first initialised as shown in 3.2.1.

3.3.2) Keying

Before generation of keystream or hashing or authenticated encryption of data, the family member must be initialised with a unique member key at runtime with a procedure identical to steps 3.2.2 and 3.2.3. The only difference is the sealing constant that is chosen according to the intended purpose: 0x2B if the cipher will be used for authenticated and unauthenticated encryption or 0xB2 if the cipher will be used for hashing. No output is produced on this stage.

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 is stored on the device for its instant loading into the cipher state at runtime.

The entire secret internal state of VEST4-80 ciphers is 256 bits in size.
The entire secret internal state of VEST8-128 ciphers is 384 bits in size.
The entire secret internal state of VEST16-160 ciphers is 512 bits in size.
The entire secret internal state of VEST32-256 ciphers is 768 bits in size.

3.3.3) Instant Keying

If the key generation or key exchange process can provide as much entropy as the cipher state size, the entire cipher state can be initialised instantly with the supplied key value as shown below. In this case steps 3.1.1 and 3.1.2 are omitted and the protocol must reflect it. All VEST ciphers are expected to maintain their advertised security

ratings. VEST4-80 ciphers require 256-bit keys, VEST8-128 – 384-bit keys. VEST16-160 – 512-bit keys and VEST32-256 ciphers require 768-bit keys for instant keying.

The cipher state is initialised with the supplied *W*-bit key value as follows:

$x^0_j = k_j$, $0 \le j < W$.
$d^0_j = k_{j+W}$, $0 \le j < 10$;
$c_i{}^0_j = k_{ij+W+10}$, $0 \le i < 16$, $0 \le j < B_i$;

The implementations including AE mode must also set the "previous round ciphertext" state to all 0s.

3.4 Hashing

3.4.1) Initialisation

Prior to hashing data, a keyed cipher family or a root cipher family is first initialised and keyed as described in 3.3, where the sealing constant 0xB2 must be used during the key sealing process. Optional IVs may also be loaded as described in 3.2.4 and 3.2.5. The cipher does not release any output for the first *Ri* rounds. This step is identical to 3.5.1.

3.4.2) Message processing

To hash an *L*-bit long stream of data, where *L* is a multiple of 8, the cipher is executed continuously in hashing mode for *L*/8 rounds (between rounds *Ra*=*Ri* and *Rb*=*Ri*+*L*/8) as described in 3.1.2.

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

3.4.3) Finalising the hash

After the hashing process the cipher state is sealed by executing the cipher in sealing mode between rounds *Ra*=*Ri*+*L*/8 and *Rb*=*Ri*+*L*/8+32 as described in 3.1.3 using the sealing constant 0xFF.

3.4.4) Returning the hash value

To produce a hash value, the cipher is executed in keystream generation mode for *H/M* rounds (between rounds *Ra*=*Ri*+*L*/8+32 and *Rb*=*Ri*+*L*/8+32+*H/M*) as described in 3.1.4. The returned keystream is an *H*-bit wide (least significant bit first) hash of the message. The hash length *H* is a multiple of *M*, but the hash value can be truncated.

3.5 Encryption without Error Propagation

3.5.1) Initialisation, keying and IV setup

Prior to data encryption, a keyed cipher family or a root cipher family is first initialised and keyed according to 3.3. The sealing constant 0x2B must be used during the key sealing process. Optional IVs may also be loaded as described in 3.2.4 and 3.2.5. The cipher does not release any output for the first *Ri* rounds. This step is identical to 3.4.1.

3.5.2) Keystream generation

To generate *L* bits of keystream or to encrypt an *L*-bit message, where *L* is a multiple of *M*, the cipher is executed continuously in keystream generation mode for *L/M* rounds (between rounds *Ra=Ri* and *Rb=Ri+L/M*) as described in 3.1.4.

The output bits *o* are applied to the plaintext or ciphertext bits according to the protocol, 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. Additional storage or processing of the ciphertext may also be required to prevent recovery of the plaintext or keystream bits from the shape of the output signal.

3.6 Authenticated Encryption

3.6.1) Initialisation, keying and IV setup

Prior to authenticated encryption, a keyed cipher family or a root cipher family is first initialised and keyed as shown in 3.3, where the sealing constant 0x2B must be used during the key sealing process. Optional IVs may also be loaded as described in 3.2.4 and 3.2.5. The cipher does not release any output for the first *Ri* rounds.

3.6.2) Encryption and decryption of data

To encrypt or decrypt an *L*-bit message, where *L* is a multiple of *M*, the cipher is executed continuously in AE mode for *L/M* rounds (between rounds *Ra=Ri* and *Rb=Ri+L/M*) as described in 3.1.5, feeding the ciphertext back into the core accumulator on every round.

3.6.3) Sealing the cipher state

After processing the data stream, the cipher state is sealed by executing the cipher in sealing mode between rounds *Ra=Ri+L/M* and *Rb=Ri+L/M*+32 as described in 3.1.3 using the sealing constant 0xFF.

3.6.4) Returning the message authentication code (MAC)

To produce a MAC, the cipher is executed in keystream generation mode for *H/M* rounds (between rounds *Ra=Ri+L/M*+32 and *Rb=Ri+L/M*+32+*H/M*) as described in 3.1.4, returning the keystream as an *H*-bit wide (least significant bit first) message authentication code. The length *H* of the message authentication code must be a multiple of *M*, but the resulting value can be truncated if necessary.

3.7 Operation in Counter Mode

Counter mode is a well understood mode of operation primarily used in block ciphers to provide scalability. It applies to all modes of operation of VEST ciphers. Systems implementing counter mode of VEST ciphers can provide high (up to Tbps) throughput for unauthenticated and authenticated encryption and collision-resistant hashing on mature ASIC geometries while being compatible with the smallest implementations.

The scalable counter mode of operation for message authentication codes and collision resistant hash functions is described in [21]. The formal specification of the CTR, CTR-

AE and CTR-HASH modes of operation of VEST ciphers utilising the built-in RNS counters will be published in the very near future.

4. VEST Design Principles

The update functions used in VEST accumulators are selected satisfying a number of cryptographic criteria that can be summarised as a requirement for all update functions and all their linear combinations to achieve the most balanced distribution of monomials of each algebraic degree in their AND-XOR, AND-OR and other algebraic forms. The Boolean functions used in VEST ciphers may not demonstrate the highest possible nonlinearity or the highest possible algebraic degree. A balanced compromise with other important cryptographic properties is chosen to ensure balanced resistance to both known and unknown attacks. Choice of the size of update functions for VEST ciphers was limited by the requirement for efficiency on most FPGA platforms. Most FPGA architectures only implement arbitrary 4-to-1 Boolean functions efficiently. Some FPGA architectures like Altera Stratix II or Xilinx Virtex-5 implement arbitrary 6-to-1 Boolean functions very efficiently. Wider update 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 update functions would significantly increase probability of exploitable linearities, correlations and other undesirable properties.

Transpositions in the VEST core accumulators are selected at random and heuristically refined minimising redundancy in the polynomial relationships between accumulator bits while maximising the number of variables introduced into those relationships on each round to ensure the highest possible diffusion rate. The accumulator cores in VEST4-80 achieve complete diffusion in 4 rounds and in VEST8-128, VEST16-160 and VEST32-256 in 5 rounds. All the feedback functions are also chosen of degree 4 to ensure that the algebraic degree of each polynomial also reaches its maximum possible value for each core after 4-5 rounds.

The key components of VEST ciphers have prime sizes and prime or co-prime periods wherever possible to avoid exploitable patterns in their combinations and to prevent decimation attacks. The RNS counters in VEST ciphers are chosen to be of practical size, small enough to allow for parallel 8-bit or 16-bit wide inputs of data and key material into the counter state, but large enough to combine into a period of at least 2^{128}.

Sizes of the entire secret internal state of all the VEST cipher families are chosen to be conservatively large and conveniently sized to be stored in popular word-aligned memory storages. The core accumulators of all VEST ciphers are also chosen to be conservatively large in proportion to the sizes of their outputs [over 18 to 1].

The authors are unaware of any issued patents covering the design or structure of VEST ciphers. Although Synaptic Laboratories Ltd BVI has filed a number of patent applications in an attempt to cover common cryptographic techniques used in such ciphers as Achterbahn, Salsa20, Trivium, TSC, VEST and XOR MACs, we are not qualified to assess scope or validity of those patent applications. In the design of VEST ciphers we used only well-studied traditional cryptographic techniques found in the cryptographic literature. The submitted source code remains copyrighted property of VEST Corporation SARL and is free for research purposes, educational and non-commercial use.

5. VEST Security

We belong to the category of cryptographers who believe that security of a cipher is different to the supported key length and cannot be measured only by the entropy provided by the key. A stream cipher compressing 4096-bit keys into its 160-bit state cannot possibly provide 4096-bit security. The same stream cipher relying exclusively on 80-bit keys cannot possibly provide 80-bit security either. Requirements for IVs in crypto-systems are relaxed sufficiently to reduce the entropy added by the IVs to an insignificantly small value that is practically ignored by the TMD trade-off attacks. Therefore we strongly recommend to use either instant keying of VEST ciphers or to initialise them with keys at least twice as long as the security rating of the cipher.

VEST4-80 cipher family tree is designed to offer 80-bit or higher short-term security. VEST8-128 cipher family tree is designed to offer 128-bit or higher medium-term security. VEST16-160 cipher family tree is designed to offer 160-bit or higher medium-term security. VEST32-256 cipher family trees are designed to offer 256-bit or higher long-term security. Keeping family keys secret also offers increased protection against power analysis and other side channel attacks.

VEST ciphers executed in [keyed or unkeyed] hashing mode can be used as collision-resistant hash functions with their security matching that of the cipher. We recommend use of hash values that are at least 160-bit long to provide 80-bit security, at least 256-bit long to provide 128-bit security, at least 320-bit long to provide 160-bit security, and at least 512-bit long to provide 256-bit security.

There are no known attacks against VEST ciphers or hash functions that are faster than serial brute-force of the key space or of the internal state. Parallel brute-force attacks are prevented by the use of keys at least twice the security rating of the cipher or by using IVs with a sufficient amount of entropy.

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. Due to the short period length, individual VEST counters cannot pass automated randomness tests, but linear combinations of 3 or 4 of them do pass all automated randomness tests.

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 input and output bits in any algebraic form being indistinguishable from random after six rounds. For comparison, five rounds of the AES (Rijndael) are required to make a controlled change in its block/key pair indistinguishable from random by automated tools, and LFSR-based ciphers like LILI-128 or LILI-2 and some simple NLFSR-based ciphers like KeeLoq or Trivium fail ASD the tests perpetually.

5.3 Periods

VEST ciphers are assisted by a nonlinear counter with a very long period. Contrary to the popular belief that periods of NLFSRs are all equiprobable averaging at $2^{W/2}$, they are not. The analysis of periods of NLFSRs as well as construction of NLFSRs and NLPFSRs with predictable periods remains an open problem. For example, maximal period is possible only in NLFSRs in which all the bits of the state are used to update at least one bit. The widths of feedback functions and other parameters like their number, algebraic degrees, bijectivity, etc. significantly affect the range and distribution of the shortest, the longest and the average periods of a random set of NLFSRs. Therefore we only present here the shortest and the longest theoretically possible periods of VEST ciphers and avoid making any claims regarding their average period lengths:

Period:	VEST4	VEST8	VEST16	VEST32
Shortest Possible:	2^{134}	2^{134}	2^{143}	2^{143}
Shortest Possible, Root:	2^{141}	2^{141}	2^{152}	2^{152}
Longest Possible Keystream, Root:	2^{244}	2^{372}	2^{498}	2^{754}
Longest Possible Keystream:	2^{247}	2^{375}	2^{503}	2^{759}
Longest Possible AE Mode, Root:	2^{248}	2^{380}	2^{514}	2^{786}
Longest Possible AE Mode:	2^{251}	2^{383}	2^{519}	2^{791}

Table 2. Period lengths of VEST ciphers

Determining average periods of VEST ciphers or probabilities of the shortest periods of VEST16-160 and VEST32-256 falling below their advertised security ratings for some keys is an open problem and is computationally infeasible. We believe that these probabilities are below 2^{-160} for VEST16-160 and below 2^{-256} for VEST32-256. The shortest theoretically possible periods of VEST4-80 and VEST8-128 are above their security ratings.

5.4 Weak Keys, Related Keys and IV Attacks

There are no fixed points and no collisions in any of the VEST ciphers' components. Although the output of 16 counters is compressed into 10 bits on every round, every single bit of the key is loaded into every single counter ensuring collision-free keying. The 8 IV/data input bits are bijectively expanded into 9 bits in the counter diffusor with control over at least one bit of the keyed counters 8 to 15 required to create an exploitable collision. Every bit of the key is loaded sequentially through all the counters and then thoroughly mixed in the core accumulator with all other key and IV bits. Six rounds are required for the diffusion of a controlled change in the core accumulator to turn the entire accumulator state into indistinguishable from random by automated tools. The 32 sealing rounds make it very hard to determine what changes in the key, IV or any other data input could result in an identifiable change in the internal state.

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 nonlinear and cannot be dismissed from the attacks. All the feedback

functions in the core accumulator are dense degree 4 polynomials. In 4-5 rounds, every bit in the accumulator state of VEST ciphers will depend on all other accumulator bits, also depending on the bits of the counters and the counter diffusor. Even assuming that the counter state is guessed by the attacker, the core accumulators of all VEST ciphers are conservatively chosen 19-27 times the width of the output. In the at least 10 rounds required to define a half of the accumulator state, these relationships grow sufficiently large and sufficiently dense 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: VEST4-80 ciphers have a 256-bit internal state, VEST8-128 ciphers have a 384-bit internal state, VEST16-160 ciphers have a 512-bit internal state, and VEST32-256 ciphers have a 768-bit internal state. This proportion is chosen to make TMD trade-off attacks require more resources than brute-force of the cipher key space.

5.7 Guess-and-Determine Attacks

For instance, to calculate the next 32 bits of keystream of a VEST32-256 cipher, the attacker must guess over 128 bits of the accumulator. Values of each of those bits depend on five to six other bits of the accumulator. 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 and Differential Attacks

The low amount of redundancy in the inputs into the feedback functions and the rapid growth of their widths and algebraic degrees with each round inherent in the design of all VEST cipher cores naturally make them tolerant to large amounts of linearity in their feedback functions. Although it is possible to choose Boolean functions with better nonlinearity for the core S-box and the core feedback function set, compromises would have to be made on the cipher's resistance to correlation and algebraic attacks. We preferred a reasonable balance between them. There are no exploitable linear approximations in VEST feedback functions or their combinations.

5.9 Distinguishing Attacks

VEST ciphers have a very high diffusion rate, they release only a small portion of the accumulator state on every round (less than 1/18), all their components are bijective and balanced, and the entire accumulator state is updated on every round with nonlinear feedback. These properties make VEST ciphers not susceptible to distinguishing attacks up to their period length that is guaranteed to be no less than 2^{128}.

5.10 Entropy Attacks

Parallel brute-force attacks [13] and quantum computing attacks can be prevented by choosing keys at least twice the security rating of the cipher. We strongly recommend using keys and IVs twice the security rating of the chosen cipher.

Authors of ciphers that do not support keys larger than the advertised security rating of those ciphers are forced to restrict the definition of a valid attack. A cipher meant for

use as a global standard cannot afford such artificial limitations and must be flexible enough to address the needs of all kinds of users by supporting variable-sized keys up to at least twice the security rating of the cipher.

5.11 Side-channel Attacks

VEST ciphers do not use any of the operations exploited by side-channel attacks:

- Key or data dependent branching operations
- Key or data dependent arithmetic operations
- Key or data dependent word-based rotation operations
- Unbalanced Boolean functions
- Memory access
- Use of subkeys or linear combinations of key bits
- Separate iterations processing only key bits

VEST ciphers use operations with several highly desirable properties:

- Shallow and near-uniform logic depth for all cryptographic operations
- Massive parallelism updating almost the entire state on every clock cycle
- Complex substitution-transposition network resulting in any glitch affecting the entire core of the cipher in a very complex way quicker than enough information can be collected to exploit it
- Round function operating with almost identical behavioural characteristics during keying, IV hashing, sealing and key compression and expansion
- High power efficiency and low signal emissions to ciphertext output ratio

The latter properties hamper the possible success of side-channel and invasive attacks and are inherent in VEST cipher design. Thanks to these properties, VEST ciphers do not include any additional exhaustive defences against side-channel attacks that other more exposed designs require. Complete implementations must still be reviewed to ensure that each component exhibits balanced power consumption so that information about the key cannot be extracted by analysis of other parts of the chip. Key storage and key interface circuitry require special attention to ensure minimal information leakage. Careful attention should also be paid to the electrical isolation of cryptographically sensitive signals to minimise leakage of information to the adjacent wires or circuits. Additional storage or processing of the ciphertext may also be required to prevent recovery of the plaintext or keystream bits from the shape of the output signal.

5.12 Timing Attacks

VEST ciphers have uniform structure and use only parallel carry-free combinatorial logic operations. It naturally makes their execution time constant in hardware.

In software, VEST ciphers can be implemented using either pedagogical single bit manipulations or the bitslice technique. Single bit manipulations require either table look-ups or key/IV/data-dependent shift operations that make them vulnerable to timing attacks. Therefore we recommend the faster bitslice implementations that naturally have constant execution time. Reconfigurable software implementations of VEST ciphers supporting family keying have to use look-up arrays, but those table look-ups remain constant for each cipher family regardless of the session key.

5.13 Security of Family Keying

Family keying became known to be hard to implement securely only due to the previous cipher designs not accommodating for it. In contrast to those designs, VEST ciphers are updated by small functions with minimal or no redundancy in their inputs. This uniform structure allows any choice of feedback functions from the same group to provide roughly the same level of security. The family keying process in VEST ciphers thus alters only the parts not essential to security of the cipher so that there is no possibility of generating an insecure family member either at random or deliberately [as a proof of such a possibility by an attacker]. Any possible minor differences in resistance against various cryptanalytic attacks between different family-keyed VEST ciphers are covered by their large security margins.

Use of a unique family key for each cipher instance naturally increases its security if the family key is kept as secret as the member key used to initialise that cipher. Evaluation of security of secret cipher families is outside the scope of this document.

5.14 Other Non-Cryptographic Attacks

Smartcard cloning, content piracy, modchip attacks against gaming devices and relay attacks against RFID and NFC devices are good examples of non-cryptographic attacks breaking security of otherwise tamper-resistant devices and cryptographically secure systems. Cryptographers have only recently begun to address these issues with new types of protocols and by designing new types of ciphers. VEST is one of such new types of ciphers designed with features that can be used in preventing some of these attacks. These VEST features are outside the scope of this specification and will be the subject of a separate publication in the near future. However, we insist that these types of attacks are given closer attention when designing new ciphers, new communication protocols and complete crypto systems. We also hope that these additional unique features allowing secure ciphers to prevent such application-specific attacks become reflected in the new standards available to the manufacturers.

6. VEST Performance

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

VEST ciphers can be re-keyed either instantly by filling the entire internal state with a 'processed' secret key, or by following the keying procedure described in 3.2, in which case initialisation of a VEST4-80 cipher with a 160-bit key without an IV takes 208 rounds, VEST8-128 with a 256-bit key – 304 rounds, VEST16-160 with a 320-bit key – 368 rounds, and VEST32-256 with a 512-bit key – 560 rounds. Loading a 64-bit IV into a keyed cipher state takes 40 rounds, a 128-bit IV – 48 rounds, and a 256-bit IV – 64 rounds. In hashing mode, all VEST ciphers hash data in one pass accepting eight bits of data per round and returning the hash value in $H/4$, $H/8$, $H/16$, or $H/32$ rounds after the final sealing of the cipher state. The time it takes to load an IV and to produce a hash or a MAC affects performance of the counter mode for individual small packets.

6.1 ASIC Performance Measurements

Synthesis and power consumption measured by LSI RapidChip processes are expected to map very closely to the worst-case standard-cell ASIC performance because they are based on similar technology processes and optimisations. The G12-r [15] (180ηm) and Gflx-r [16] (110ηm) R-cell count is a close approximation to the standard low-power standard-cell gate count usage such as the MOSIS library [17].

Based on the conservative standard RapidChip design front-end sign-off process, a full-featured implementation of VEST32-256 in AE mode satisfies a demand for 256-bit secure 10Gbps authenticated encryption @ 167 MHz on 180ηm LSI Logic RapidChip platform ASIC technologies in less than 45K Gates and zero SRAM. On the 110ηm RapidChip technologies, a full-featured implementation of VEST32-256 offers 20Gbps authenticated encryption @ 320 MHz in less than 45K gates. The circuit area of application-specific VEST ciphers restricted in their modes of operation or implementing instant keying is certainly lower than the circuit area of full-featured implementations.

Description	Unroll	Chip	Geo ηm	Tgt MHz	Actual MHz	Mb/s	Capped Mb/s	Total R-Cell	DFF R-Cell	Die Area micron^2
VEST32-AREA	1x	G12-r	180	1	204.7	13,100	**10,688**	38,825	37.0%	900x900
VEST32-SPEED	2x	G12-r	180	303	256.6	16,422	**10,688**	64,230	12.6%	1100x1100
VEST4-AREA	1x	Gflx-r	110	1	556.5	2,262	**1,248**	5,392	49.5%	215x215
VEST8-AREA	1x	Gflx-r	110	1	513.1	4,104	**2,496**	9,321	41.2%	284x284
VEST16-AREA	1x	Gflx-r	110	1	506.2	8,099	**4,992**	13,599	37.6%	350x350
VEST32-AREA	1x	Gflx-r	110	1	473.8	15,161	**9,984**	23,500	40.2%	465x465
VEST32-SPEED	1x	Gflx-r	110	644	642.4	20,556	**9,984**	28,208	30.2%	500x500
VEST32-SPEED	2x	Gflx-r	110	313	328.6	21,030	**19,968**	40,038	20.9%	800x800

Table 3. Lower-bound synthesis and static timing for full-featured implementations of VEST ciphers

Static Power analysis was performed using the RapidChip-PowerStation spreadsheet provided to us under a non-disclosure agreement. Theta Junction-to-Ambient thermal resistance was set to 12.8°C/W with a commercial ambient temperature of 55.0°C. Logic switching activity was set at 50%.

Description Low-AREA	Unroll	Cap MHz	Cap Mb/s	Total R-Cell	DFF R-Cell	mW @ 1 MHz	mW @ 10 Mb/s	mW @ 100 Mb/s	mW @ 1000 Mb/s	mW @ Cap MHz
VEST4	1x	312.0	**1,248**	5,392	49.5%	0.11	0.21	1.72	16.77	20.26
VEST4	2x	312.0	**2,496**	7,336	34.6%	0.16	0.18	1.16	11.02	27.41
VEST8	1x	312.0	**2,496**	9,321	41.2%	0.20	0.23	1.49	14.18	35.28
VEST8	2x	312.0	**4,992**	15,233	25.1%	0.33	0.26	1.26	11.21	55.34
VEST16	1x	312.0	**4,992**	13,599	37.6%	0.29	0.23	1.14	10.27	50.91
VEST16	2x	294.8	**9,433**	22,099	20.9%	0.48	0.30	1.01	8.15	74.98
VEST32	1x	312.0	**9,984**	23,500	40.2%	0.50	0.30	1.10	9.03	88.28
VEST32	2x	264.1	**16,902**	39,496	19.5%	0.85	0.47	0.54	7.45	141.09

Table 4. Lower-bound static power consumption for low-area
full-featured implementations VEST ciphers on Gflx-r 110ηm technology

Tables 3 and 4 present measurements of the synthesis, place-and-route, static timing and static power consumption of the complete data-path of full-featured implementations of VEST ciphers without control logic. For a detailed excursion on hardware and software performances of VEST ciphers, please see our companion documents [18] and [19].

6.2 Software Performance

Massive parallelism of VEST ciphers feedback and minimal redundancy in inputs into their update functions naturally makes them fast only in ASIC and FPGA hardware and prohibits their fast software implementations. Comparing ASIC speeds with the fastest possible bitslice software implementations processing a single data stream at the same clock speed, VEST4-80 is over 1000 times slower in software, VEST8-128 is over 2300 times slower in software, VEST16-160 is over 3500 times slower in software, and VEST32-256 is over 6000 times slower in software. Large word width in bitslice software implementations can speed up the counter mode and packet-based server applications, but it cannot accelerate keystream generation, hashing or authenticated encryption of a single stream. Use of pipelining can additionally improve VEST speed on some processors up to 20%.

7. Improvements

A number of minor improvements have been made since the publication of the original version of this document [11]. These improvements include:

- ✓ Better choice of feedback functions for the RNS counters resulting in a stronger more balanced counter output [20] and longer periods
- ✓ New bit permutations with better fan-out distribution for all VEST cores
- ✓ Improved streamlined keying process providing a significant reduction in hardware implementation complexity and circuit area
- ✓ Reset of the state to all binary 1 bits reducing ASIC circuit area
- ✓ Native authenticated encryption mode of operation [12]
- ✓ CTR and CTR-AE modes of operation
- ✓ A fast bitslice software implementation of all the root cipher families

No changes were made to the fundamental structure or operation of the cipher.

During the first two weeks of Phase II of the eSTREAM competition, a bug was discovered in our fan-out counter implementation that has resulted in all the new core accumulators not having the claimed limited fan-out. We have corrected this error and generated new bit permutations for the core accumulators with the promised limited fan-out on all the registers (found in the appendixes A, B, C and D). It has subsequently also affected the choices of bits for the output combiners that are found in the appendix H, and the attached test vectors. No other changes were made. The new core accumulators also support the unpublished NAE mode with ciphertext feedback into their M most significant bits (into bits $W–M$ to $W–1$ instead of the bits 10 to $M+9$ as in the AE mode).

In January 2007, Antoine Joux inspired us to check the counter diffusor where we noticed a typo that made its way into the cipher specification and into the source code. This typo was corrected in the section 2.2 of this paper.

We sincerely apologise for any inconvenience caused by these changes.

8. Summary

Family tree:	VEST4	VEST8	VEST16	VEST32
Output, bits per clock:	4	8	16	32
Expected security, bits:	80	128	160	256
Advised key length, bits:	160	256	320	512
State Size, bits:	256	384	512	768
Core Size, bits:	83	211	331	587
Shortest Possible Period:	$>2^{134}$	$>2^{134}$	$>2^{143}$	$>2^{143}$
Longest Possible Period:	$>2^{251}$	$>2^{383}$	$>2^{519}$	$>2^{791}$
Counter Size, bits:	163	163	171	171
Min software clocks per round:	1074	2378	3626	6282
Software clocks per byte, 32x:	~67	~64	~47	~42
Stratix I AE Speed, Gbps:	~1	~2	~4	~7
Stratix II AE Speed, Gbps:	~2	~4	~8	~13
110 ηm Min-Area AE, Gbps:	~2	~4	~8	~16
110 ηm Min-Area AE, gates:	<6K	<11K	<15K	<24K
110 ηm Max-Speed AE, Gbps:	–	–	–	~20
110 ηm Max-Speed AE, gates:	–	–	–	<28K

Table 1. Summary of the key properties of VEST ciphers at time of publication (area figures include control logic for full-featured implementations)

9. Authors Contributions and Acknowledgments

Sean O'Neil is the principal author of this paper and is the author, designer and owner of VEST ciphers. Benjamin Gittins has contributed the drawings and the FPGA and ASIC performance figures. Howard A. Landman has contributed on FPGA and ASIC issues related to the cipher design and significantly improved the overall clarity of the original version of this paper. We would also like to thank Philip Hawkes for advising us to improve the quality of the RNS counters, Antoine Joux for inspiring us to double check the counter diffusor and everyone else who provided their invaluable feedback that helped us improve the quality of this paper and cipher design.

10. References

[1] Auguste Kerckhoff, La Cryptographie Militaire, 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", Cryptology ePrint Archive, 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 Laboratories Ltd., September 2004.

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

[11] S. O’Neil, B. Gittins, H. A. Landman, “VEST Hardware-Dedicated Stream Ciphers”, eSTREAM Archive Report 2005/032, April 2005.

[12] S. O’Neil, B. Gittins, “Authenticated Encryption Mode of VEST Ciphers”, eSTREAM Archive Report 2005/077, November 2005.

[13] D. J. Bernstein, “Understanding brute force”, April 2005.

[14] J. Hong, P. Sarkar, “Rediscovery of Time Memory Tradeoffs”, Cryptology ePrint Archive, Report 2005/090, March 2005.

[15] LSI Logic, “G12-r RapidChip Cell Technology – Databook”, July 2004.

[16] LSI Logic, “Gflx-r RapidChip Cell Technology – Databook”, May 2005.

[17] Tanner CES, “Digital Low Power Standard Cell Library for MOSIS TSMC CMOS 0.25 Process. Deep Sub-Micron Technology”, 1999.

[18] B. Gittins, H. A. Landman, S. O'Neil and R. Kelson, “A Presentation on VEST Hardware Performance, Chip Area Measurements, Power Consumption Estimates and Benchmarking in Relation to the AES, SHA-256 and SHA-512”, Cryptology ePrint Archive Report 2005/415, November 2005.

[19] B. Gittins, “VEST-32, 256-bit secure, Single-Pass Authenticated Encryption. 20 Gigabit/s @ 312MHz on 110ηm LSI Logic RapidChip Platform ASIC Technology <45K Gates, Zero SRAM and <150mW”, March 2006.

[20] Private discussion with Philip Hawkes.

[21] M. Bellare, R. Guerin and P. Rogaway, “XOR MACs: New methods for message authentication using finite pseudorandom functions”, Crypto 1995.

[22] A. Klimov and A. Shamir, “A New Class of Invertible Mappings”, 2002.

Appendix A

VEST4-80 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	58
1	0	1	2	3	4	47
2	0	1	2	3	4	36
3	0	1	2	3	4	27
4	0	1	2	3	4	11
5	4	4	1	3	2	20
6	5	5	2	4	3	78
7	6	6	3	0	4	70
8	7	7	0	1	5	77
9	8	8	5	2	6	56
10	9	9	6	8	7	35
11	9	9	8	10	0	62
12	11	11	8	5	6	64
13	11	11	12	6	10	44
14	13	10	7	5	9	26
15	9	11	8	13	12	10
16	15	12	14	5	11	37
17	11	14	9	6	13	76
18	17	14	11	7	5	8
19	18	15	12	16	7	32
20	19	16	14	9	8	40
21	20	17	14	10	12	71
22	21	18	15	13	20	2
23	1	19	20	10	6	13
24	19	20	17	13	7	34
25	23	21	18	14	8	51
26	25	24	19	15	9	68
27	26	17	23	16	15	59
28	22	24	18	10	27	63
29	28	25	26	18	12	14
30	29	28	23	16	11	1
31	30	29	24	27	26	6
32	17	27	25	21	22	46
33	32	29	15	22	16	38
34	33	27	22	13	31	54
35	19	31	12	24	34	0
36	34	32	18	25	19	60
37	31	33	30	35	36	73
38	23	34	31	30	10	21
39	17	20	32	23	31	74
40	15	37	21	30	23	39
41	33	39	40	30	22	42
42	41	38	35	14	25	55
43	29	23	36	32	35	28
44	43	24	18	42	36	65

45	40	41	19	31	28	69
46	45	42	39	28	37	57
47	38	41	40	43	44	17
48	38	21	45	37	47	19
49	48	45	43	35	16	30
50	49	46	43	39	33	9
51	49	47	50	38	31	67
52	51	48	45	41	32	5
53	52	49	46	42	36	12
54	49	50	45	42	44	4
55	37	33	47	53	41	29
56	40	52	50	45	39	61
57	29	53	50	27	56	18
58	53	34	57	43	42	72
59	58	44	52	48	53	22
60	59	54	58	26	29	49
61	60	57	54	50	51	31
62	38	48	55	34	61	43
63	25	60	59	53	62	3
64	48	39	46	53	57	52
65	64	61	59	60	48	66
66	50	62	65	61	46	45
67	66	41	62	38	57	75
68	57	64	61	63	33	16
69	68	67	50	16	58	53
70	61	66	63	56	29	33
71	24	70	67	65	54	15
72	61	64	70	71	69	7
73	72	69	66	59	56	23
74	45	68	33	49	69	82
75	67	66	72	7	70	80
76	25	67	73	69	63	81
77	52	68	60	53	74	79
78	77	55	63	75	74	25
79	76	76	61	78	51	48
80	79	79	76	73	68	24
81	80	80	77	74	79	41
82	81	81	80	75	71	50

VEST4-80 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, 45, 56, 48, 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, 40, 99, 41, 32, 83, 121, 110, 97, 112, 116, 105, 99, 32, 76, 97, 98, 115, 32, 76, 116, 100, 46, 10, 0

Appendix B

VEST8-128 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	19
1	0	1	2	3	4	82
2	0	1	2	3	4	167
3	0	1	2	3	4	58
4	0	1	2	3	4	59
5	4	4	1	3	2	172
6	5	5	2	4	3	129
7	6	6	1	0	5	104
8	7	7	0	5	6	153
9	8	8	5	7	6	49
10	9	9	6	7	8	184
11	10	10	7	9	5	157
12	11	11	8	5	7	169
13	10	10	9	6	8	34
14	13	13	10	7	9	98
15	14	14	12	8	6	53
16	15	15	12	9	11	29
17	16	16	13	10	15	61
18	17	14	11	13	12	168
19	18	15	12	8	17	43
20	19	16	13	9	18	160
21	20	17	14	10	19	121
22	21	18	15	11	20	70
23	22	19	16	12	21	68
24	23	20	17	13	22	52
25	24	21	18	14	16	96
26	25	22	19	15	14	180
27	26	23	20	16	10	76
28	27	24	21	17	11	120
29	28	25	22	18	12	100
30	29	26	24	19	13	66
31	30	27	24	20	14	199
32	27	23	25	31	15	155
33	32	29	26	22	16	170
34	33	30	27	23	28	94
35	34	31	28	24	18	150
36	35	32	29	25	19	24
37	36	33	30	26	20	78
38	37	34	31	27	21	13
39	37	35	26	28	22	185
40	25	11	33	35	23	202
41	39	37	38	40	30	197
42	41	38	35	31	25	107
43	42	34	36	32	26	40

44	43	40	36	33	27	99
45	44	41	38	34	28	164
46	45	42	40	39	35	137
47	46	43	29	36	30	191
48	47	44	41	37	31	141
49	48	45	42	38	32	95
50	49	24	36	39	33	136
51	50	47	44	40	34	36
52	29	48	45	41	35	93
53	21	52	48	42	36	1
54	23	50	47	53	39	165
55	54	51	48	53	38	81
56	55	52	49	45	39	171
57	55	53	52	46	40	108
58	57	54	51	47	41	109
59	50	47	46	41	32	10
60	30	17	57	56	33	134
61	49	57	60	50	44	166
62	32	58	55	51	45	5
63	62	44	56	61	46	146
64	59	60	57	53	47	147
65	49	37	52	64	31	73
66	65	62	63	55	49	23
67	66	63	60	56	50	106
68	67	64	51	57	43	26
69	67	54	62	59	61	193
70	46	66	63	59	69	84
71	70	67	64	42	54	126
72	71	68	49	56	65	125
73	72	69	66	57	62	17
74	65	70	67	63	34	190
75	74	71	68	64	66	44
76	75	72	69	65	59	117
77	58	51	70	66	60	161
78	76	48	28	74	61	69
79	40	75	72	68	73	67
80	79	58	76	69	63	102
81	80	77	74	70	64	101
82	53	60	75	71	43	103
83	82	79	81	72	80	149
84	68	62	37	73	67	38
85	84	56	78	74	68	92
86	85	82	79	75	69	55
87	86	83	80	44	70	72
88	87	73	81	77	71	25
89	43	85	82	78	72	33
90	46	86	83	79	89	123
91	89	87	84	80	74	110
92	54	78	88	90	75	145
93	92	89	86	52	76	37

94	93	90	91	83	88	115
95	63	91	53	84	78	51
96	95	91	89	85	92	42
97	61	42	96	86	80	4
98	61	97	55	87	95	182
99	74	95	79	88	77	9
100	94	45	82	65	92	179
101	100	97	94	58	99	132
102	81	88	95	101	91	162
103	102	99	96	92	86	142
104	103	94	97	93	87	77
105	79	101	98	94	88	3
106	105	104	100	90	81	47
107	50	103	106	86	91	173
108	56	107	104	97	83	12
109	98	108	90	58	43	62
110	77	88	103	109	102	118
111	94	70	110	107	65	148
112	97	83	105	101	93	151
113	112	73	93	102	96	85
114	106	85	93	110	89	178
115	60	111	114	97	98	27
116	94	112	113	115	105	195
117	116	113	110	106	100	15
118	117	114	110	109	104	143
119	115	83	112	89	87	97
120	119	111	101	107	98	16
121	72	102	100	107	85	175
122	92	104	115	108	90	48
123	122	119	116	112	117	64
124	123	99	117	121	100	201
125	39	96	118	114	85	83
126	125	113	119	95	122	0
127	99	123	120	116	67	91
128	96	93	125	69	127	187
129	117	121	126	128	112	8
130	116	126	78	111	113	176
131	130	127	124	76	114	22
132	71	55	126	121	48	124
133	110	108	126	122	107	200
134	133	130	127	117	101	113
135	125	131	128	132	118	88
136	73	134	129	123	119	135
137	129	133	128	105	120	156
138	110	100	131	128	123	46
139	138	111	81	71	122	186
140	80	139	133	136	130	181
141	140	137	131	130	103	57
142	140	106	118	131	96	154
143	142	101	136	129	126	90

144	143	140	125	133	127	131
145	132	141	138	124	123	159
146	145	120	139	135	129	39
147	128	145	127	142	124	2
148	102	144	141	137	147	80
149	109	131	105	138	132	14
150	116	146	143	137	122	63
151	121	84	144	140	135	20
152	150	148	145	141	135	183
153	118	149	139	142	136	111
154	120	131	147	143	134	119
155	154	151	145	144	138	50
156	148	152	111	146	126	89
157	156	121	107	145	140	65
158	130	136	151	154	134	139
159	153	155	152	148	130	127
160	137	151	125	141	145	198
161	124	153	154	150	144	7
162	158	161	103	137	148	196
163	158	159	156	152	68	11
164	162	160	120	113	147	138
165	124	117	158	149	163	31
166	164	165	142	155	143	177
167	103	163	99	156	150	54
168	167	164	161	157	124	116
169	144	134	155	158	152	194
170	143	168	121	134	139	133
171	140	138	160	151	112	28
172	135	168	166	153	155	56
173	171	157	139	162	156	128
174	173	158	154	127	157	60
175	174	143	171	109	173	189
176	175	170	87	165	159	6
177	144	151	176	166	160	158
178	177	174	176	167	169	79
179	169	54	177	159	162	18
180	151	95	142	174	169	206
181	168	172	174	170	175	87
182	164	180	173	172	165	130
183	122	178	163	172	166	32
184	179	180	163	64	183	192
185	141	106	75	174	179	210
186	171	180	183	92	165	209
187	185	183	173	181	170	163
188	187	184	181	177	186	203
189	188	185	182	178	184	174
190	152	186	187	164	146	188
191	188	114	152	178	167	207
192	171	176	120	191	178	205
193	192	179	150	188	176	105

194	174	190	188	163	184	122
195	194	180	175	189	188	41
196	132	192	189	185	193	30
197	190	195	182	179	186	204
198	197	149	191	187	196	144
199	198	188	192	195	182	74
200	199	178	190	194	195	208
201	200	197	194	198	184	86
202	189	183	161	191	185	71
203	199	199	201	200	114	35
204	196	196	203	195	193	140
205	204	204	202	186	196	45
206	205	205	105	199	195	114
207	206	206	205	203	204	75
208	207	207	203	193	181	152
209	208	208	207	194	202	21
210	209	209	201	200	158	112

VEST8-128 root family

RNS counters indexes:

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

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

Input bit permutation indexes:

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

Appendix C

VEST16-160 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	143
1	0	1	2	3	4	189
2	0	1	2	3	4	35
3	0	1	2	3	4	88
4	0	1	2	3	4	307
5	4	4	1	3	2	273
6	5	5	2	4	3	204
7	6	6	5	0	1	2
8	7	7	6	0	5	84
9	8	8	5	7	6	158
10	9	9	6	8	7	272
11	10	10	7	5	8	181
12	11	11	8	5	9	41
13	12	12	9	6	8	51
14	13	13	10	7	8	50
15	13	13	11	12	9	36
16	15	15	12	9	10	205
17	16	16	15	10	6	111
18	17	17	14	11	7	260
19	18	18	11	12	16	89
20	19	19	16	14	9	303
21	20	20	17	14	10	97
22	21	21	18	13	11	311
23	22	22	19	16	21	229
24	22	22	20	17	13	199
25	13	13	20	18	14	292
26	25	22	19	15	20	234
27	26	23	20	16	10	231
28	27	24	21	17	11	69
29	28	25	22	18	12	168
30	29	26	23	19	28	13
31	30	27	24	20	14	252
32	31	28	25	21	15	75
33	32	29	26	22	16	203
34	33	30	27	23	17	302
35	34	31	15	24	18	125
36	35	32	29	25	19	312
37	12	17	30	26	14	136
38	37	34	31	27	21	266
39	13	34	32	33	28	251
40	39	36	33	29	23	271
41	40	37	34	30	24	58
42	41	38	35	31	25	105
43	42	39	36	32	26	57
44	35	40	31	39	27	26

45	44	36	38	34	28	249
46	45	42	43	35	29	280
47	46	43	40	36	30	207
48	47	44	41	37	31	298
49	48	23	42	38	32	96
50	49	24	43	28	46	288
51	50	47	44	40	34	196
52	24	44	32	48	22	264
53	43	38	33	48	36	113
54	53	50	47	43	37	145
55	54	51	48	44	38	216
56	55	52	21	45	39	180
57	56	53	50	46	40	182
58	57	54	30	47	41	16
59	58	55	52	48	42	282
60	59	56	53	49	48	150
61	60	57	41	50	44	314
62	47	58	55	50	45	61
63	62	59	56	52	46	10
64	63	60	57	53	47	95
65	23	18	58	54	37	166
66	65	62	59	55	49	241
67	66	63	60	56	41	98
68	67	64	66	57	51	236
69	68	65	62	58	52	29
70	69	66	63	59	53	135
71	70	67	51	60	54	245
72	71	68	65	61	43	90
73	39	69	67	62	56	290
74	73	33	67	63	57	49
75	74	71	72	64	58	109
76	75	72	41	29	35	187
77	76	73	52	66	55	137
78	77	76	71	67	61	161
79	78	75	72	63	62	238
80	27	76	73	69	63	219
81	80	77	74	70	64	8
82	81	78	73	36	65	173
83	82	79	76	72	66	281
84	83	54	77	80	61	276
85	66	25	75	74	61	149
86	85	82	60	49	69	309
87	75	83	80	86	70	78
88	87	69	81	77	71	215
89	26	88	85	38	52	151
90	40	55	83	79	73	250
91	90	87	76	80	39	21
92	58	87	85	81	90	253
93	49	81	37	82	79	20
94	92	61	87	49	77	194

95	83	91	88	82	78	68
96	95	92	89	85	79	120
97	96	93	90	86	80	289
98	97	94	91	67	64	202
99	72	95	51	84	82	224
100	99	35	93	98	89	144
101	73	76	82	90	88	116
102	59	93	80	98	91	40
103	102	99	57	84	86	223
104	103	100	84	93	77	261
105	61	101	104	94	84	291
106	105	102	99	95	89	232
107	106	33	60	96	83	177
108	107	104	101	97	91	277
109	74	105	102	108	92	4
110	109	106	103	99	93	123
111	110	91	51	107	88	141
112	111	108	105	101	95	103
113	112	109	106	102	96	255
114	84	110	107	109	97	221
115	95	56	70	114	111	269
116	94	97	109	105	99	28
117	114	113	74	106	100	211
118	117	114	115	107	101	52
119	118	94	113	108	102	67
120	45	100	96	119	103	191
121	98	117	113	81	107	304
122	64	118	115	83	121	206
123	122	119	105	112	106	37
124	87	116	96	123	62	305
125	88	110	124	118	108	209
126	89	117	119	96	121	172
127	126	68	120	116	113	44
128	127	100	121	112	117	146
129	128	108	122	118	111	72
130	116	113	123	129	104	293
131	130	127	124	111	114	248
132	116	128	124	121	115	110
133	50	92	42	98	94	262
134	86	130	127	123	117	226
135	97	115	134	124	118	92
136	135	132	129	125	99	184
137	136	129	130	126	120	208
138	81	134	131	127	121	154
139	138	137	132	136	122	233
140	134	139	133	64	115	285
141	128	137	134	130	124	32
142	85	139	120	141	131	301
143	142	45	136	132	126	131
144	125	140	137	128	45	81

145	74	114	138	137	94	274
146	65	142	120	145	86	244
147	127	78	140	136	130	200
148	147	144	141	137	131	3
149	148	79	110	138	132	18
150	131	146	51	102	78	284
151	69	147	84	137	114	54
152	151	148	108	103	75	164
153	70	109	98	131	125	313
154	123	150	147	143	118	139
155	154	151	148	112	138	183
156	155	154	149	145	139	108
157	156	93	150	19	146	31
158	157	154	95	140	141	114
159	142	151	135	153	129	296
160	103	155	140	149	126	43
161	160	86	152	132	144	63
162	161	158	155	105	54	129
163	162	159	156	92	135	17
164	163	160	157	98	153	22
165	164	161	158	154	148	212
166	165	145	159	155	162	198
167	116	163	160	156	150	239
168	167	146	159	139	164	34
169	168	165	162	158	152	270
170	113	166	163	142	153	82
171	170	167	164	143	154	257
172	112	150	151	168	103	286
173	149	169	101	158	156	256
174	170	136	152	149	157	287
175	152	110	174	128	165	160
176	175	122	169	161	126	77
177	176	126	173	139	89	228
178	155	71	157	167	147	295
179	153	175	172	106	171	38
180	88	160	179	170	163	107
181	153	138	174	170	164	119
182	173	178	175	171	165	0
183	133	179	176	161	166	294
184	134	135	183	173	180	195
185	183	130	172	143	168	93
186	185	182	179	175	169	258
187	158	146	90	185	170	147
188	187	184	167	177	171	162
189	168	72	150	178	165	56
190	131	117	183	179	173	152
191	140	68	184	149	171	259
192	144	188	169	181	175	133
193	159	143	185	182	176	48
194	193	185	187	129	190	70

195	180	184	146	186	100	246
196	115	147	144	166	180	112
197	196	146	190	186	151	267
198	197	194	191	166	181	148
199	119	136	192	188	198	128
200	199	189	193	121	150	201
201	135	197	194	190	184	101
202	161	185	178	183	191	176
203	119	122	196	192	142	86
204	177	167	197	153	190	178
205	173	201	198	194	188	217
206	205	148	198	100	59	227
207	206	156	200	111	190	23
208	116	204	207	197	171	300
209	143	205	202	174	192	156
210	189	206	120	203	193	235
211	193	182	199	210	194	159
212	178	184	202	207	195	297
213	188	196	162	211	125	306
214	213	205	200	199	197	167
215	214	79	208	204	198	85
216	71	174	104	68	212	14
217	170	213	169	190	192	47
218	186	214	211	185	201	132
219	218	216	212	160	195	171
220	209	202	176	217	164	240
221	78	188	189	201	178	275
222	125	197	216	212	134	60
223	203	163	216	222	206	94
224	201	220	217	213	207	140
225	165	189	221	214	208	65
226	145	180	219	215	177	99
227	199	226	163	223	210	192
228	186	220	148	53	132	9
229	194	208	90	125	212	268
230	176	206	123	219	155	79
231	230	227	224	220	70	80
232	200	189	177	104	215	122
233	101	229	227	104	177	124
234	226	230	227	159	171	283
235	203	217	172	221	218	247
236	200	123	91	225	219	179
237	236	233	230	221	220	213
238	189	188	142	111	237	24
239	154	235	75	228	222	55
240	174	195	233	229	225	46
241	191	161	204	177	205	308
242	221	220	167	231	145	117
243	46	151	236	209	133	155
244	172	240	223	229	227	100

245	244	223	238	234	213	33
246	245	119	203	235	68	6
247	222	225	213	224	230	193
248	247	233	238	237	222	265
249	245	241	242	248	232	91
250	236	246	234	225	233	76
251	222	157	122	240	234	1
252	232	240	239	172	235	153
253	220	216	223	242	195	218
254	202	250	210	209	241	53
255	254	251	248	244	245	11
256	225	252	249	255	208	237
257	252	247	240	133	243	169
258	247	254	251	127	228	220
259	258	186	196	248	242	74
260	207	256	253	249	243	62
261	149	256	254	250	229	222
262	261	258	215	247	158	39
263	230	255	257	252	261	73
264	263	260	193	257	247	157
265	264	228	258	224	230	115
266	265	206	259	255	256	121
267	266	198	207	256	250	320
268	147	244	160	87	15	324
269	249	265	253	195	252	327
270	269	266	263	259	253	319
271	179	267	242	260	244	315
272	169	252	218	257	182	329
273	178	246	259	269	261	254
274	143	270	267	263	257	87
275	273	210	268	271	267	27
276	210	246	269	265	260	25
277	258	265	210	267	254	59
278	277	274	271	267	261	188
279	175	191	272	268	262	142
280	194	268	279	243	236	134
281	266	239	274	270	264	186
282	249	242	238	226	279	326
283	239	273	245	181	258	325
284	259	266	264	204	227	317
285	226	65	248	277	281	328
286	214	244	279	219	282	263
287	217	207	239	276	270	316
288	237	284	281	277	271	71
289	260	285	282	278	288	12
290	289	264	283	279	273	174
291	191	281	282	289	239	243
292	232	285	277	286	275	318
293	262	291	286	275	241	5
294	291	284	133	172	285	330

295	273	288	294	243	278	310
296	295	255	274	288	231	323
297	266	293	290	286	280	214
298	297	293	291	287	281	210
299	226	295	292	288	282	138
300	284	249	292	246	285	279
301	300	211	294	290	268	190
302	233	300	278	294	226	321
303	301	302	293	298	174	15
304	291	237	251	258	235	299
305	292	283	301	269	276	45
306	304	305	297	292	296	102
307	306	303	300	296	278	185
308	307	289	301	297	238	19
309	308	280	302	253	272	66
310	309	242	299	306	287	322
311	305	307	285	300	310	225
312	311	308	231	299	295	42
313	312	309	306	302	296	126
314	302	276	284	310	312	170
315	263	263	244	283	306	64
316	218	218	312	310	299	118
317	304	304	275	303	294	165
318	107	107	272	290	42	127
319	243	243	310	315	225	104
320	318	318	313	312	314	278
321	298	298	316	278	315	7
322	321	321	317	315	262	242
323	304	304	133	320	280	163
324	262	262	320	317	311	106
325	321	321	313	318	322	230
326	325	325	324	287	323	83
327	223	223	241	314	316	130
328	327	327	324	322	325	197
329	328	328	311	322	319	30
330	329	329	326	323	319	175

VEST16-160 root family

RNS counters indexes:

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

Feedback function indexes:

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, 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

Input bit permutation indexes:

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

Appendix D

VEST32-256 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	131
1	0	1	2	3	4	224
2	0	1	2	3	4	17
3	0	1	2	3	4	150
4	0	1	2	3	4	354
5	4	4	1	3	2	113
6	5	5	2	4	3	19
7	6	6	0	1	5	109
8	7	7	0	6	5	191
9	8	8	5	7	6	370
10	8	8	6	5	7	461
11	10	10	7	9	5	100
12	11	11	8	7	10	342
13	12	12	9	6	7	542
14	13	13	10	11	8	47
15	14	14	11	8	9	478
16	15	15	12	9	11	448
17	12	12	13	10	6	329
18	17	17	14	11	10	179
19	18	18	15	12	8	117
20	19	19	16	13	9	132
21	9	9	17	14	13	279
22	20	20	18	15	11	197
23	22	22	18	16	12	398
24	22	22	15	23	13	389
25	24	24	21	18	14	532
26	25	25	21	16	15	195
27	26	26	12	20	16	73
28	10	10	23	22	17	465
29	14	14	28	22	24	244
30	28	28	25	23	16	213
31	30	30	27	24	20	94
32	31	31	30	25	26	105
33	32	32	27	20	22	71
34	17	17	30	29	28	537
35	34	34	31	28	24	216
36	30	30	27	18	29	151
37	36	36	35	27	16	6
38	26	26	33	31	37	274
39	13	13	24	14	35	184
40	35	35	23	34	17	411
41	29	29	21	34	33	452
42	41	38	35	31	25	336
43	42	39	36	32	26	269

44	43	40	37	33	27	199
45	44	41	38	34	28	128
46	45	42	39	35	29	163
47	46	43	40	36	21	61
48	47	44	41	37	31	348
49	48	45	42	38	32	247
50	49	46	43	39	33	327
51	50	47	44	40	34	366
52	51	48	45	44	35	464
53	52	49	46	42	36	97
54	53	50	47	43	37	535
55	33	54	50	51	26	145
56	55	52	49	45	39	519
57	56	53	50	46	40	466
58	57	54	51	47	41	321
59	58	55	52	48	42	107
60	59	56	53	49	38	220
61	59	60	55	37	54	547
62	61	58	55	51	45	238
63	62	59	47	57	46	409
64	45	63	38	53	47	377
65	64	61	58	54	48	422
66	19	62	59	55	65	360
67	42	63	60	56	50	301
68	67	64	43	57	51	66
69	68	65	62	58	52	264
70	69	66	63	59	53	266
71	70	67	64	60	54	486
72	71	68	32	61	55	317
73	72	69	66	49	56	553
74	73	70	61	63	57	241
75	74	56	29	53	68	175
76	75	72	69	65	59	340
77	76	73	70	66	60	338
78	77	74	71	67	61	303
79	78	75	72	68	21	441
80	76	62	71	69	63	92
81	80	66	29	77	26	275
82	81	78	75	71	65	21
83	82	79	76	72	66	146
84	83	80	77	73	67	516
85	25	81	77	58	46	459
86	80	85	82	75	41	193
87	86	83	80	76	70	207
88	87	84	81	77	71	424
89	88	85	82	78	72	432
90	89	86	83	79	73	414
91	90	87	84	80	74	65
92	91	19	85	81	75	248
93	92	89	86	82	76	353

94	93	61	87	83	77	158
95	94	91	88	84	78	54
96	95	92	89	19	94	40
97	96	60	69	54	90	313
98	69	97	91	87	81	23
99	98	95	92	88	82	403
100	99	96	93	89	83	215
101	100	97	94	90	84	123
102	101	98	95	91	85	462
103	62	99	96	102	86	444
104	103	100	97	93	87	404
105	104	101	58	62	88	388
106	64	23	48	93	89	234
107	106	103	100	96	90	510
108	107	104	101	97	91	511
109	108	105	102	98	92	141
110	98	106	103	99	93	31
111	31	107	110	100	94	552
112	106	108	105	101	95	287
113	39	88	44	106	96	539
114	113	110	107	103	97	489
115	114	111	108	104	98	467
116	24	20	19	102	81	451
117	116	40	110	106	100	358
118	71	114	111	107	117	142
119	94	115	112	108	102	228
120	66	119	97	109	103	80
121	99	120	114	109	104	203
122	121	118	115	111	105	36
123	122	119	116	112	106	161
124	123	120	117	113	107	484
125	124	121	118	114	108	439
126	125	122	119	115	109	368
127	126	123	120	116	110	237
128	127	124	121	117	111	85
129	128	125	122	118	83	288
130	91	49	123	116	129	169
131	79	127	82	120	115	330
132	33	102	114	121	103	285
133	105	129	98	122	116	505
134	133	79	127	123	73	122
135	134	131	128	124	118	48
136	135	117	129	125	119	312
137	136	133	130	126	120	181
138	137	134	131	127	121	297
139	110	135	132	128	138	177
140	139	137	105	129	123	527
141	140	137	134	130	124	29
142	141	138	135	131	125	202
143	63	57	22	85	40	347

144	135	140	17	136	38	271
145	144	141	138	134	128	361
146	145	142	139	135	129	517
147	146	115	140	136	130	98
148	147	133	141	137	131	442
149	148	145	142	138	132	124
150	149	146	143	139	133	419
151	150	147	144	41	134	235
152	148	43	25	141	128	318
153	152	149	146	142	136	250
154	122	153	147	143	150	373
155	154	151	148	144	138	320
156	155	152	149	117	139	400
157	156	153	132	146	109	343
158	157	154	151	147	118	262
159	158	155	152	148	142	64
160	88	99	144	80	87	227
161	160	157	154	150	144	55
162	132	158	136	18	130	291
163	56	159	156	162	132	232
164	120	68	104	65	67	106
165	164	161	34	154	148	529
166	165	162	159	155	149	385
167	114	73	141	156	131	497
168	140	164	165	157	167	84
169	168	165	162	158	152	405
170	125	166	128	117	159	157
171	163	167	164	160	154	334
172	171	168	165	161	155	13
173	172	169	166	162	156	115
174	173	170	167	163	157	186
175	174	171	168	164	158	332
176	112	94	169	165	159	208
177	176	174	170	166	160	152
178	177	174	171	167	161	99
179	178	147	172	105	162	305
180	179	176	173	169	163	425
181	113	180	174	170	84	309
182	145	178	74	171	156	522
183	182	179	176	172	166	78
184	183	180	23	173	15	463
185	184	181	178	174	168	108
186	185	182	179	175	169	351
187	178	130	184	20	133	541
188	187	184	104	177	171	140
189	188	185	118	178	172	2
190	189	186	183	179	173	493
191	152	92	184	180	190	420
192	191	188	185	181	175	111
193	93	190	72	151	113	365

194	193	190	187	183	161	125
195	194	176	188	184	178	359
196	195	192	189	185	179	62
197	85	193	196	155	112	233
198	158	185	37	188	130	72
199	198	195	192	188	182	483
200	199	79	193	189	187	475
201	200	124	126	187	127	103
202	90	201	145	182	67	378
203	167	200	196	168	186	387
204	186	143	197	193	187	3
205	141	169	131	194	195	249
206	205	202	199	195	189	524
207	167	171	112	196	190	546
208	153	177	194	197	109	53
209	208	205	202	198	192	26
210	209	206	203	199	193	245
211	210	207	204	200	194	503
212	116	179	205	181	195	183
213	164	176	206	189	204	364
214	213	210	207	203	197	295
215	150	126	208	74	198	502
216	215	212	168	205	209	18
217	216	213	210	206	200	112
218	150	214	211	148	86	168
219	218	215	212	208	202	144
220	52	170	161	194	113	137
221	99	217	27	32	204	453
222	221	218	215	211	205	118
223	222	219	204	183	206	294
224	201	220	217	213	207	429
225	222	191	221	214	101	251
226	225	222	219	215	209	549
227	211	51	214	163	195	438
228	78	224	221	227	211	391
229	228	225	222	218	212	363
230	229	226	223	219	213	43
231	184	153	207	220	151	79
232	231	228	225	221	215	91
233	232	146	113	223	182	328
234	157	230	227	223	217	504
235	216	214	212	203	218	446
236	75	84	192	222	44	201
237	236	233	230	226	220	346
238	237	234	231	227	221	219
239	238	189	175	191	222	544
240	227	236	186	21	221	162
241	102	160	216	162	169	187
242	241	238	235	231	225	394
243	242	223	236	165	226	428

244	201	240	159	233	210	185
245	197	204	241	243	115	246
246	245	242	239	235	229	25
247	239	136	240	236	230	518
248	247	244	235	237	231	477
249	108	129	237	234	232	525
250	173	36	243	242	233	406
251	160	230	19	240	181	267
252	251	248	245	241	235	38
253	252	249	246	242	236	554
254	253	250	217	243	237	58
255	57	139	79	156	247	352
256	255	247	252	132	239	196
257	177	253	181	246	240	418
258	257	254	230	232	241	479
259	107	126	252	177	110	426
260	142	245	253	236	243	344
261	260	251	254	244	224	226
262	204	137	255	159	245	455
263	262	259	256	252	246	392
264	219	232	202	253	257	178
265	264	249	119	112	215	383
266	265	262	201	255	228	160
267	224	145	263	256	250	468
268	240	264	261	257	251	324
269	268	267	229	258	252	104
270	238	208	263	259	253	211
271	270	267	264	260	254	550
272	271	172	265	180	262	194
273	260	203	249	266	256	9
274	201	76	185	100	225	449
275	274	271	268	264	258	443
276	232	65	183	140	242	300
277	252	213	192	203	260	492
278	277	274	190	267	261	272
279	278	275	272	268	262	27
280	119	197	273	271	263	417
281	235	277	274	270	249	490
282	233	278	275	271	265	512
283	111	279	276	272	266	514
284	283	280	277	232	225	24
285	284	281	278	191	187	173
286	285	282	279	275	269	156
287	144	210	219	276	270	376
288	287	284	281	277	271	136
289	288	285	282	278	272	223
290	289	276	140	158	283	315
291	245	287	284	280	274	349
292	259	153	285	281	198	435
293	292	289	286	282	276	362

294	293	229	279	142	239	401
295	294	231	271	284	274	482
296	290	292	289	183	244	434
297	296	293	211	286	280	457
298	297	294	291	287	281	225
299	298	295	292	288	282	367
300	299	296	293	289	283	471
301	170	207	285	270	259	59
302	301	298	295	291	285	325
303	302	299	296	292	286	11
304	303	300	297	293	287	256
305	304	238	298	286	288	399
306	191	302	299	295	289	469
307	306	303	300	296	290	416
308	190	230	157	297	291	10
309	308	305	302	298	292	526
310	303	199	250	234	293	119
311	310	302	247	307	266	221
312	206	155	243	308	268	212
313	312	309	192	208	296	548
314	262	310	231	291	307	292
315	249	299	234	308	298	270
316	237	127	288	247	298	200
317	286	109	310	306	300	335
318	317	314	311	307	301	176
319	202	277	312	199	251	339
320	295	263	273	309	278	57
321	320	317	314	310	304	229
322	320	145	315	311	305	545
323	180	319	316	312	306	159
324	290	262	266	220	312	415
325	324	265	321	314	308	34
326	325	322	319	315	309	498
327	318	323	324	316	310	536
328	212	324	321	327	311	60
329	300	325	286	318	313	127
330	126	50	238	200	208	445
331	330	163	143	305	324	192
332	182	122	325	321	283	255
333	332	329	326	322	316	254
334	260	300	194	228	247	280
335	219	164	328	324	318	384
336	294	277	331	175	137	88
337	314	319	330	326	30	49
338	320	302	205	318	280	456
339	196	335	207	223	322	174
340	339	336	333	329	323	430
341	340	188	258	334	307	304
342	341	338	335	331	325	0
343	342	339	336	332	326	520

344	343	340	337	333	327	319
345	344	283	327	334	328	8
346	281	301	172	335	216	198
347	334	344	340	336	250	534
348	255	294	341	304	331	413
349	348	345	342	338	332	277
350	349	336	347	290	308	68
351	229	266	333	339	152	133
352	227	269	246	287	304	447
353	352	349	346	342	336	379
354	353	350	347	343	337	268
355	354	351	348	344	338	290
356	201	170	151	345	275	393
357	139	284	267	317	290	50
358	345	327	218	347	331	322
359	358	355	352	348	342	134
360	357	315	350	341	353	242
361	323	336	224	350	343	70
362	347	196	355	351	323	499
363	361	360	332	250	346	170
364	349	254	210	353	347	286
365	364	361	358	354	348	282
366	357	276	263	355	282	16
367	273	363	297	233	366	473
368	68	86	311	357	244	436
369	314	331	305	317	352	500
370	369	366	363	359	353	32
371	64	32	306	319	354	507
372	304	368	348	365	355	316
373	355	218	366	362	356	431
374	373	356	272	288	316	120
375	360	305	368	352	358	421
376	346	359	345	343	348	89
377	376	373	319	366	360	41
378	191	374	371	306	372	39
379	378	375	372	368	362	143
380	379	376	377	356	363	427
381	380	377	374	370	364	311
382	381	378	375	371	365	77
383	338	163	376	372	366	135
384	280	380	321	373	367	260
385	256	381	378	153	372	205
386	385	382	379	375	369	474
387	386	383	380	376	370	74
388	387	384	381	337	371	22
389	246	248	346	378	372	496
390	389	317	383	198	334	326
391	353	289	384	380	374	481
392	199	368	177	381	375	222
393	392	273	253	338	329	306

394	211	313	311	387	388	423
395	394	391	388	384	378	180
396	395	327	226	256	379	217
397	396	393	390	386	380	102
398	397	394	391	387	381	82
399	241	395	297	383	340	487
400	392	354	273	313	242	331
401	385	397	373	390	328	281
402	358	401	395	95	374	369
403	345	297	396	392	386	523
404	403	400	397	393	387	538
405	341	401	398	394	388	345
406	272	322	399	395	326	437
407	406	403	400	396	390	209
408	378	295	288	397	401	501
409	408	405	402	398	392	56
410	409	406	403	399	393	153
411	350	407	408	337	238	5
412	342	402	166	28	70	488
413	412	362	393	198	396	87
414	306	412	406	248	388	460
415	337	147	89	323	405	189
416	414	412	92	405	362	101
417	416	413	410	406	400	33
418	228	394	411	407	401	261
419	270	415	257	90	264	278
420	366	323	347	341	382	110
421	420	417	414	410	404	333
422	315	418	402	304	329	76
423	307	197	416	412	406	528
424	423	420	417	413	407	509
425	212	303	392	414	389	28
426	344	403	328	416	425	167
427	353	401	345	416	422	472
428	320	260	330	424	282	148
429	389	143	237	409	362	454
430	429	426	423	419	413	375
431	325	240	424	420	414	356
432	431	428	150	421	303	374
433	432	429	426	422	416	182
434	433	430	427	423	371	30
435	373	356	430	180	258	491
436	397	244	36	408	111	543
437	436	433	430	426	420	15
438	437	368	431	346	374	407
439	438	435	417	375	422	116
440	418	275	326	333	423	138
441	440	437	411	408	52	506
442	281	428	200	30	364	35
443	442	350	436	338	400	93

444	443	359	437	351	427	402
445	322	441	438	434	428	533
446	445	442	385	432	429	257
447	446	421	356	436	294	190
448	447	444	441	437	431	307
449	322	327	254	320	313	386
450	356	404	440	439	419	515
451	450	447	444	440	434	166
452	267	448	369	246	301	299
453	365	404	446	396	436	51
454	149	263	321	443	445	14
455	408	365	364	441	438	252
456	455	452	449	445	439	164
457	456	333	451	382	453	63
458	457	454	451	447	441	433
459	351	239	160	453	446	560
460	293	349	448	313	268	42
461	396	457	251	380	444	574
462	461	458	455	451	445	568
463	399	321	456	330	285	575
464	334	258	434	311	428	579
465	464	161	175	454	290	558
466	444	452	459	461	381	580
467	384	463	460	367	450	556
468	450	395	459	121	440	583
469	434	447	461	458	403	12
470	440	334	411	243	313	562
471	470	319	464	460	467	172
472	371	468	443	461	455	559
473	234	425	214	462	468	188
474	473	470	467	463	457	323
475	404	471	468	460	220	239
476	331	369	344	465	377	90
477	444	229	341	96	461	582
478	275	477	369	418	461	530
479	474	291	472	468	393	390
480	464	459	292	478	470	570
481	375	250	449	476	453	408
482	481	478	475	471	465	139
483	400	480	476	465	466	214
484	419	416	359	473	467	302
485	154	471	332	361	440	578
486	485	482	479	475	469	283
487	244	483	357	453	470	495
488	487	436	481	477	413	114
489	269	485	482	478	472	1
490	489	486	483	479	473	259
491	490	487	484	480	474	75
492	409	149	481	445	349	355
493	414	411	332	435	476	572

494	493	490	487	483	477	371
495	467	491	134	458	472	563
496	495	492	489	485	479	129
497	480	496	490	486	477	95
498	463	443	418	450	355	67
499	423	451	173	344	273	147
500	488	496	291	386	410	276
501	500	497	494	490	484	293
502	386	473	495	499	254	284
503	502	499	496	492	486	52
504	480	124	497	425	431	458
505	398	459	488	443	469	236
506	476	395	494	493	470	585
507	420	151	456	436	466	397
508	474	414	308	478	493	576
509	458	418	368	498	435	567
510	251	485	503	499	329	480
511	444	433	504	500	494	149
512	511	508	505	501	495	380
513	310	296	508	512	335	584
514	506	510	472	335	491	357
515	514	511	508	504	498	565
516	314	512	257	431	402	555
517	382	258	510	506	427	230
518	515	424	220	464	391	210
519	456	402	449	235	422	155
520	519	516	513	509	503	337
521	392	508	452	166	476	557
522	506	463	521	459	505	540
523	417	449	510	411	508	586
524	513	521	506	417	511	372
525	524	521	462	514	508	314
526	357	523	519	515	417	289
527	468	437	382	516	510	561
528	527	503	494	418	451	571
529	528	525	522	518	512	485
530	379	397	523	265	513	521
531	513	501	350	520	527	86
532	531	528	525	521	515	44
533	532	529	526	522	516	126
534	174	530	491	402	517	573
535	457	531	452	497	524	569
536	358	401	529	525	519	412
537	318	533	530	526	520	494
538	505	533	531	248	450	263
539	509	457	538	372	522	69
540	133	466	186	320	523	4
541	540	537	534	530	398	450
542	487	337	535	531	438	154
543	542	539	536	532	526	566

544	515	507	451	533	526	350
545	544	541	538	534	528	206
546	545	542	385	535	529	243
547	509	385	541	335	390	382
548	513	535	538	435	531	581
549	548	545	430	523	493	7
550	549	546	543	539	533	265
551	550	547	544	540	534	231
552	513	518	535	536	527	577
553	552	549	546	529	536	564
554	553	550	547	543	537	165
555	554	554	551	548	544	130
556	555	555	552	217	545	440
557	556	556	491	553	546	45
558	545	545	554	551	547	81
559	427	427	539	541	470	298
560	464	464	556	489	433	218
561	485	485	541	552	454	341
562	259	259	558	529	551	83
563	551	551	125	471	553	258
564	563	563	560	498	468	396
565	382	382	561	558	554	476
566	551	551	193	485	562	508
567	504	504	441	123	48	410
568	78	78	373	146	135	204
569	531	531	514	560	517	20
570	176	176	245	405	559	470
571	570	570	567	564	518	37
572	524	524	568	565	521	240
573	544	544	569	566	572	171
574	60	60	257	527	563	513
575	175	175	555	239	265	273
576	575	575	502	415	565	308
577	529	529	557	280	568	381
578	577	577	574	560	567	96
579	578	578	577	559	566	46
580	566	566	259	573	101	121
581	580	580	571	579	564	296
582	581	581	578	318	571	395
583	582	582	579	576	555	310
584	583	583	464	370	269	253
585	584	584	387	549	39	551
586	585	585	581	476	573	531

VEST32-256 root family

RNS counters indexes:

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

Feedback function indexes:

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, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1023, 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, 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

Input bit permutation indexes:

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

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

Appendix E

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 F

RNS counters

Counter	Width	Feedback Function	Possible Period Lengths	
0	11	0xDD1B4B41	1021	1027=13*79
1	11	0xEE72650D	1009	1039
2	11	0x93A1E709	997	1051
3	11	0x9C550E3F	979=11*89	1069
4	11	0xB23F8963	961=31*31	1087
5	11	0xCB0E5AAB	877	1171
6	11	0x9AE3132F	859	1189=29*41
7	11	0x8DC509C7	817=19*43	1231
8	11	0xD70D09C9	811	1237
9	11	0xD6A21F59	799=17*47	1249
10	11	0xE561B0CB	769	1279
11	11	0xC5F84439	757	1291
12	11	0xB2E5C68B	751	1297
13	11	0xD282E3BD	727	1321
14	11	0xD6DD04AB	619	1429
15	11	0xECCA941F	601	1447
16	10	0x94E74373	509	515=5*103
17	10	0x8666525D	503	521
18	10	0xD4054B4F	487	537=3*179
19	10	0xA3660FAD	467	557
20	10	0xC69C15F9	461	563
21	10	0xB83A3EC9	443	581=7*83
22	10	0xF865319D	431	593
23	10	0xE0CE9AC7	383	641
24	10	0x8D18BD6D	347	677
25	10	0xCD0AF563	281	743
26	10	0xEEB5B411	263	761
27	10	0xA59C289B	251	773
28	10	0xD15B7893	227	797
29	10	0x842BB1D3	197	827
30	10	0x85F4AB17	173	851=23*37
31	10	0xD1E122EF	167	857

Appendix G

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 H

Output Combiners

VEST4-80 output combiner:

o	*xa*	*xb*	*xc*	*xd*	*xe*	*xf*
0	15	23	29	43	57	67
1	33	49	52	55	69	73
2	17	22	45	60	61	72
3	18	19	39	46	65	66

VEST8-128 output combiner:

O	*xa*	*xb*	*xc*	*xd*	*xe*	*Xf*
0	38	89	115	137	159	177
1	31	51	56	99	128	148
2	25	60	65	83	108	176
3	63	85	111	138	141	145
4	88	113	139	142	149	165
5	20	26	116	136	143	162
6	39	47	67	124	154	173
7	81	118	131	132	135	166

VEST16-160 output combiner:

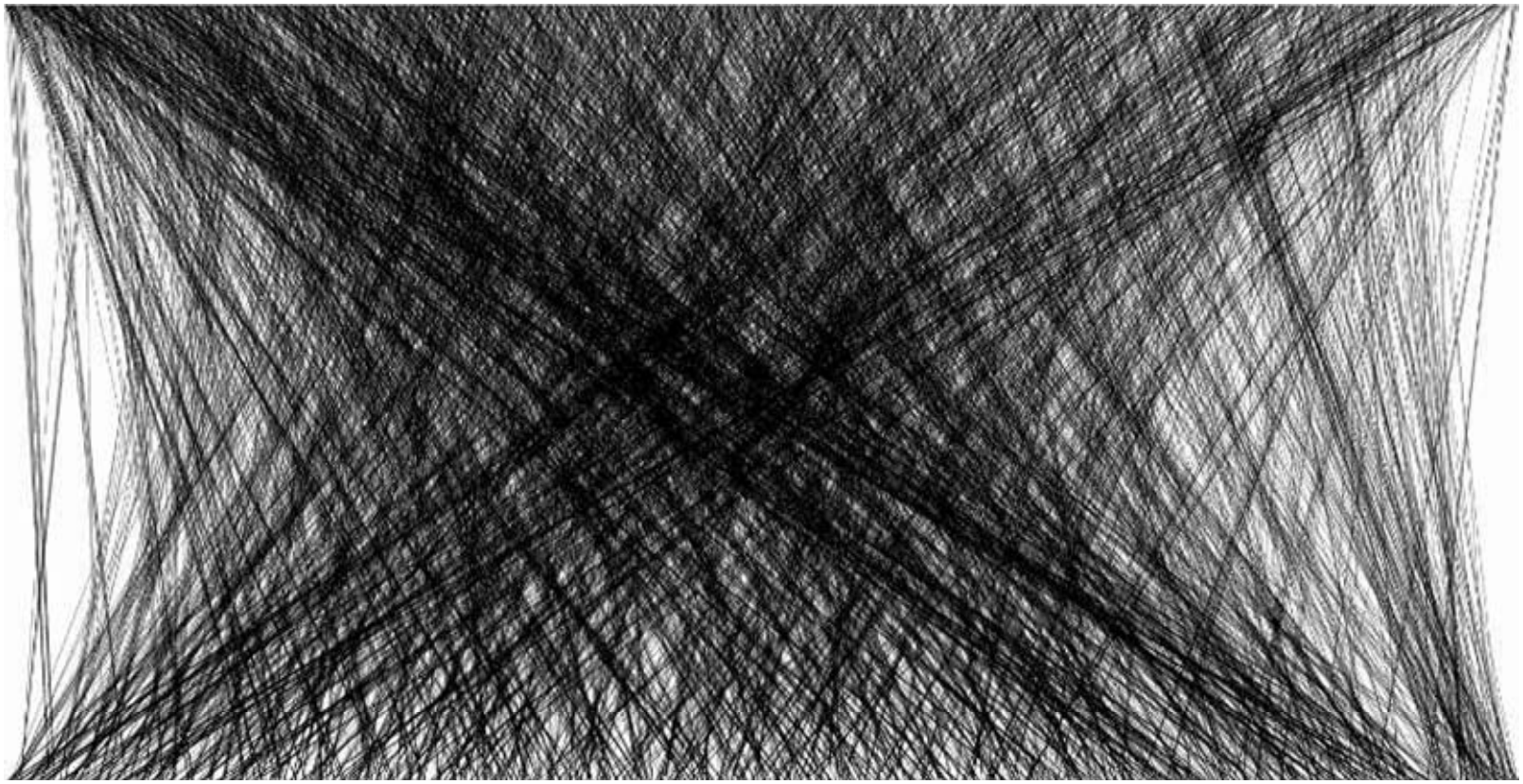
O	*xa*	*xb*	*xc*	*xd*	*xe*	*xf*
0	91	99	110	140	176	194
1	39	62	63	137	211	213
2	34	52	76	121	129	218
3	153	195	212	237	244	250
4	74	108	150	152	240	253
5	115	145	155	162	173	226
6	55	85	90	146	156	182
7	113	223	227	247	259	265
8	38	49	117	122	183	261
9	53	123	177	235	238	257
10	68	80	179	191	207	248
11	65	79	144	161	164	196
12	26	114	193	239	245	262
13	32	56	103	202	216	220
14	28	222	224	233	236	246
15	40	44	95	131	200	221

VEST32-256 output combiner:

o	*xa*	*xb*	*xc*	*xd*	*xe*	*xf*
0	148	174	227	257	402	455
1	125	133	182	237	407	423
2	106	116	118	167	189	190
3	53	160	374	375	384	444
4	89	98	164	192	294	404
5	91	93	159	249	427	437
6	80	101	170	256	300	386
7	144	254	261	330	379	415
8	103	266	277	328	331	367
9	43	122	138	246	338	399
10	120	280	383	409	416	430
11	63	76	108	356	391	435
12	107	155	178	194	387	403
13	51	87	319	363	388	425
14	204	245	252	262	304	428
15	42	198	307	345	373	449
16	205	251	278	312	352	433
17	119	222	343	346	361	422
18	185	333	376	405	432	454
19	59	146	187	200	303	442
20	84	143	250	264	270	394
21	115	202	217	241	297	421
22	48	57	79	238	401	434
23	127	223	295	315	420	424
24	64	78	104	290	362	446
25	55	136	158	203	208	426
26	147	177	209	215	282	309
27	85	186	286	316	344	429
28	77	124	183	255	340	436
29	281	299	306	311	313	359
30	56	326	332	351	364	418
31	60	65	152	207	268	447

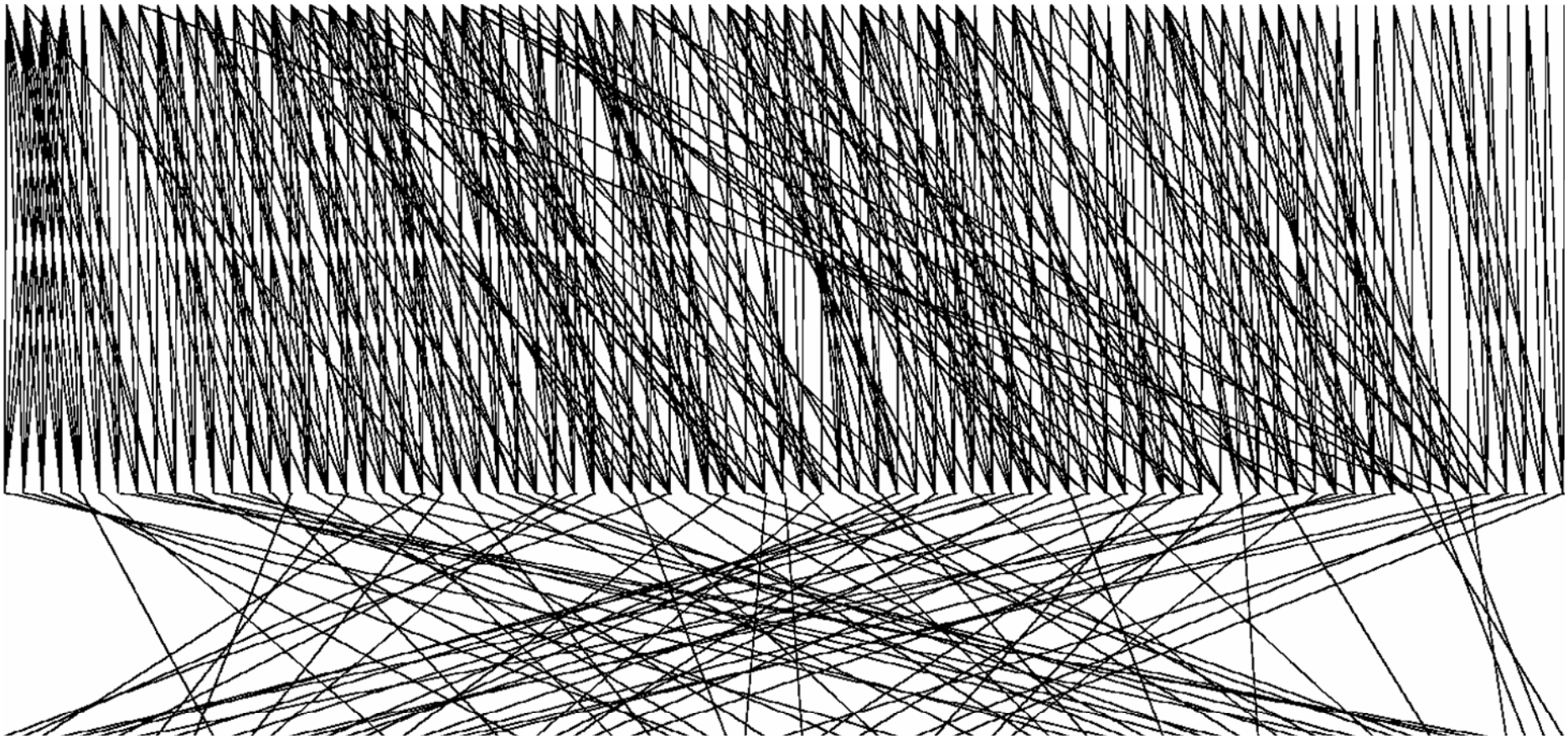
Appendix I

Pictures



Picture 1. Structure of one round of feedback in VEST32-256 core accumulator

The structure of VEST ciphers is extremely complex for mental comprehension, but due to its uniformity they are easy to implement in hardware by generating the source code with simple automated tools, which can be seen from the attached bitslice source code.



Picture 2. Structure of one round of feedback in VEST4-80 core accumulator

The above picture shows the substitution and the transposition layers separately. The structure of the substitution layer becomes clear: the vertical lines are showing the linearly added bits. It also shows how only the “previous” bits are used as inputs into all the accumulator’s feedback functions to ensure its bijectivity. The subsequent transposition hides that structure.