



ZK-Crypt Security Analysis & Related Issues -

Preliminary Draft for January 2009 Launch

Security Analysis of the ZK-Crypt

Basic Tenets & Rationale for Major Enhancements

Dual Track Orthogonal Feedback in all Modes Precludes Message Modification

1st Round Massive Diffusion of Feedback & Random Control Variables

Hybrid and Correlation Filters – 2 of 3 Majority/3XOR & Store & XOR

Correlation Anomalies

Intractability of Algebraic Cryptanalytic Attacks

Accelerated HMAC & Cipher Initialization

Concatenating/Pairing 9.5K Gate Engines -Highest Security & Increased Speed

Parallel Configurations – for Transparent Simultaneous Decryption & Hash Digest

Double Key Validation on Hash Value

Exhaustive Statistics – Differentials, Correlation & Permutations

Accelerated Deterministic & TRNG AIS 31 (FM Random) All-Digital Noise Sources

A Reference for –

"Understanding the ZK-Crypt a Hash/Stream Cipher/TRNG for (almost) all Reasons", www.FortressGB.com,

CONTRIBUTORS:

CARMI GRESSEL

AVI HECHT

NICOLAS T. COURTOIS

GREGORY BARD

RAN GRANOT

ORR DUNKELMAN (eSTREAM SECURITY ANALYSIS)

January 2009

Introduction

The present hardware optimized, triple function, cryptographic engine was launched to be a single step 32 bit word random number generator/stream cipher. We massively increased the amount of diffusing random logic filters, so that the number of intermediary variables is more than three times the number of state variables (flip-flops), with exponential increase of algebraic cryptocomplexity. Incremental improvements led to hashed loading of secret keys and finally to robust data authentication with proved resistance to message modification and collision. The process was exceedingly difficult as we followed our since vindicated, non-conventional approach that both stream ciphers and hash functions should be based on massively diffusing feedback and non-linear permutations to "get the most out of" each binary state variable.

The following tenets explained in [undst] are also valid propositions in the security analysis. The tenets are the backbone of the security of the ZK-Crypt. In Chapter 1, we give the rationales for the design decisions; based on the four tenets:

- 1) Large scale diffusion, wherein each binary state variable is an immediate function of at least three preferably disparate state variables, leads to intractable algebraic cryptocomplexity; unbiased binary and word state variables; and to a stark reduction of state variable autocorrelation.
- 2) XOR summing of two or more slightly biased binary structures; i.e., bits, nibbles bytes or 32 bit words, results in less predictable, less biased output; if and only if there is no or very little detectable cross correlation between the structures; i.e., one structure may show strong signs of autocorrelation.
- 3) Judicious use is made of conventional and tailored non-linear and linear artifacts to simultaneously control: the levels of autocorrelation and bias within binary structures, and to cause an acute reduction of cross correlation between linear combined binary structures.
- 4) The Dual (2x32 bit) Track Orthogonal Feedback permuted streams are linearly combined into interacting state variables thereby amplifying the affects of both the randomized increments of the Message counter and the irreconcilable modifications of Message Words; the resulting increased diffusion effects proven preclusion of Herded Collisions and Message Modification attacks and effective resistance to Pre-Image attacks.

Analysts generally prefer simple "easily analyzed" algorithms which we have replaced with pseudorandom statistically and mathematically proved constructs. The analyses relate to security in keyed (proprietary) and unkeyed hashing, stream ciphering and FIPS 198-1 HMAC hashing.

2 The ZK-Crypt Tenets Briefly Explained

The tenets with brief explanatory examples:

- 1) **Large scale diffusion, wherein each binary state variable is an immediate function of at least three preferably disparate state variables, leads to intractable algebraic cryptocomplexity; unbiased binary and word state variables; and to a stark reduction of state variable autocorrelation.**

A device with massive immediate first degree linear and non-linear diffusion, wherein each variable is affected by 3 or more state variables; where there is no sensible bias, no sensible end effect, and no sensible auto or cross correlation; where said device embraces a large state space of over 400 binary state variables, and more than double intermediary logic variables with tiers permuted by random displacements and jittered clocks; and where pseudorandom words are shifted and randomly linear and non-linearly combined; together cause a condition where an algebraic or quantum computing cryptanalysis is inherently intractable as a consequence of the huge diffused number (>50 million) of unbiased monomials in irreducible equations. [Chapter 2]

- 2) **XOR summing of two or more slightly biased binary structures; i.e., bits, nibbles bytes or 32 bit words, results in less predictable, less biased output; if and only if there is no or very little detectable cross**

correlation between the structures; i.e., one structure may show strong signs of autocorrelation.

Examples:

The output of the Least Significant, LS, cell of an up-counter that randomly counts even and odd length pulse streams and thereby generates a random toggle (jittered oscillator) variable. This "irregular" oscillator output is XOR summed to the unpredictable outputs of randomized pseudo-Linear Feedback Shift Registers, nLFSRs, and to random data bits from unbiased word structures, to generate unpredictable unbiased permutation drivers. Such unbiased signals also modulate conventional random logic to generate "occasional" permutating pulses; e.g., to generate missing clock pulses; and to generate complemented internal feedback (Slip) pulses on one or more of the 12 ZK-Crypt nLFSRs. [zk-ccc R04-09 & P02-04].

The permuted 2 of 3 Majority Splash Output filter is forced into a strongly correlated condition, in which at every clock every fourth output bit is highly biased to the same random value. We prove statistically that the output is unbiased and strongly correlated. When linearly combined to a slightly correlated (even number of ones and zeroes) word, the output is unbiased, with trace autocorrelation.

3) Judicious use of conventional and tailored non-linear and linear artifacts to simultaneously control: the levels of autocorrelation and bias within binary structures; and the acute reduction of cross correlation between linear combined binary structures.

- a) Conventional correlation immunizers, e.g., a summation of present and past results; and pseudo-random linear displacements, reduce auto correlation; and,
- b) A non-linear transformation can force a biased input into an orderly correlated word, e.g., every n'th bit is forced, with high probability to be of same value. [zk-ccc T03-T04]

4) The Dual (2x32 bit) Track Orthogonal Feedback permuted streams are linearly combined into interacting state variables thereby amplifying the affects of both the randomized increments of the HAIFA Message counter [zk-ccc H0-3A] and the irreconcilable modifications of Message Words; the resulting increased diffusion effects proven preclusion of Collision and Message Modification attacks and effective resistance to Pre-Image attacks. Swapping orthogonal feedback between concatenated ZK-Crypt units insures that a modified Message in one unit will irreconcilably corrupt the state variables in both units.

We show that any change of a valid message word aberrates orthogonal feedback streams fed into the Register Bank and the Data Churn, such that different parts of the Bank and the Churn are corrupted differently so that any reconciliation of the corruption caused by one stream will further corrupt (and continue to corrupt) parts corrupted by the second stream. Formally, we define 2 hash/MAC feedback streams as orthogonal in a tiered word manipulation architecture where: any non-valid sequence of message words causes a first feedback stream to corrupt at least one first bit in at least one word in a first tier on a first clock; and on a second clock a second false message word activates a reconciliation (a flip back) of the corrupted at least one bit in said first tier's corrupted word to the valid previously unmodified second clock condition; and whereas simultaneously, the at least one corrupting bit in the first false message word of the message sequence corrupts at least one bit in the second feedback stream which simultaneously first corrupts at least one bit in at least one other tier simultaneously, such that the said second message word cannot reconcile the said corrupted bits in the second tier,

thereby modifying the state variables of said tiered word manipulation architecture from the valid condition generated by a valid message sequence.

E.g., in an attempted message modification wherein an adversary changes a "\$100", USD value to a "£100", UK Pounds Sterling value in a first message word in a dual track orthogonal feedback configuration:

On a first clock, flipping false message bits wherein said flips cause an equivalent false "£" bit value in at least a first word in one first feedback stream affecting one specific tier; and in a second clock in a second sequence message, via said first feedback stream, flipping bits operative to reconcile relevant bits in said first tier; thereby to restore the "£" related bits back to the valid "\$" representation bits in said first tier, hiding the first word modification in said at least first tier; while simultaneously,

said flipping on the first clock false message bits causes an equivalent false "£" bit value in the second feedback stream causing an equivalent false "£" bit value in at least one second word in one second tier; wherein said second clock first restoration "\$" message bits cannot restore said second word in said second tier to a valid condition.[Appendix 2]

A Result is defined as an intermediate hash digesting value which is a linear function of an enciphered mask output of the engine core XOR summed to a message word.

We prove generically, that if one feedback sequence is a function of a first clocked Result, and a second feedback sequence is a function of the (same) first clocked Result XOR summed to the previous clocked Result, the two feedback sequences are orthogonal, and provably preclude message modification.

The Dual Track Orthogonal Feedback streams when XOR summed to 64 disbursed output bits of an enlarged Biham-Dunkelman HAIFA counter, serve to prevent hash collisions for the first (more than) 2^{61} message word sequences. The only degree of freedom available to an adversary in a secured environment is to modify message inputs. To engineer a collision, the attacker would have to forge message words that would complement the disbursed counter bits in both feedback streams. The 64 bit pseudo-random sequence generated by the Hybrid Mersenne Prime LFSR/binary HAIFA Counter XOR summed to the two orthogonal Feedback Stores obviously is non-malleable, and the 64 bits of which cannot be reconciled to a new herded colliding sequence via a sequence of 32 bit Message Words.

We use hardware parlance when referring to logic variables. A signal is a variable that serves to start or cause some action; e.g., a Slip which aberrates an nLFSR¹ feedback, a Splash Select signal which routes displacement bit in the Splash Matrix, or a clock signal. We are more explicit, however, when referring to clock signals. We generally call clock signals pulses, as they are reserved for "Sampling" memory cells, and are only effectively active when "rising" from zero to logic '1'. A clock pulse remains at logic '1', for only the first half of a clocking cycle. State variables are the outputs of memory cells, and they drive random logic gates, e.g., XOR, AND, and OR. During the initial less than 1 nanosecond of rise time of a clock signal, the input value to the memory cell (a flip-flop) becomes or continues to be the output value for the remainder, at least, of the clock cycle.

¹ Most of the so called "nLFSR" shift registers in the ZK-Crypt linearly receive external feedback; and may better be referred to as "pseudo linear feedback shift registers", pLFSRs.

Chapter 1: Rationale of Enhancements & Features of the Original ZK-Crypt nLFSR Secret Vault

In the introductory chapter, and in the "Understanding the ZK-Crypt" document, we stated the tenets that lead to the compact, highly diffused, very fast triple function random number core. We welcomed NIST's request to supply a rationale for the development stages of the SHA3 candidates, which we use as a preface to the more rigorous component analyses. The following 19 point rationale and sectional overview is meant to bridge the comprehension gaps between understanding the concept, the security analysis and the whys and wherefores which make for a trusted realization on silicon.

Referenced archived circuitry, procedural and algorithmic flow charts, and concept diagrams are found in [zk-ccc]; referenced chapters; e.g., [Chap.8] refer to chapters in this SHA3 security analysis. [undst] is the latest overview of the ZK-Crypt, which we suggest reading prior to following explicit proofs; the [zk- a-z glos] guide and glossary provides succinct explanations of ZK-Crypt concepts for easy reference. The design of this cipher engine revolved around designing and testing components and their combinations, in each case, following the four basic tenets. The analyses relate to security in keyed (proprietary) and unkeyed hashing, stream ciphering and FIPS 198-1 HMAC hashing.

DOC Claim: All documents are being and have been reviewed by submitters and third parties. We have prefaced proofs with intuitive proofs and basic tenets and supplied the Fig. 1 "Pathfinder" to focus the reader on the location of the claims.

Rationale: We realize that first time reviewers are perhaps daunted by the seeming complexity of the ZK-Crypt; and were unable to grasp the basic simple tenets of the massively diffusing ZK-Crypt functions. Prominent cryptographers welcomed the non-conventional highly diffused stream-cipher approach, fine tuned with exhaustive standard and proprietary testing; others claimed it was impossible to properly analyze. Some requested easier to understand rationale [robshaw].

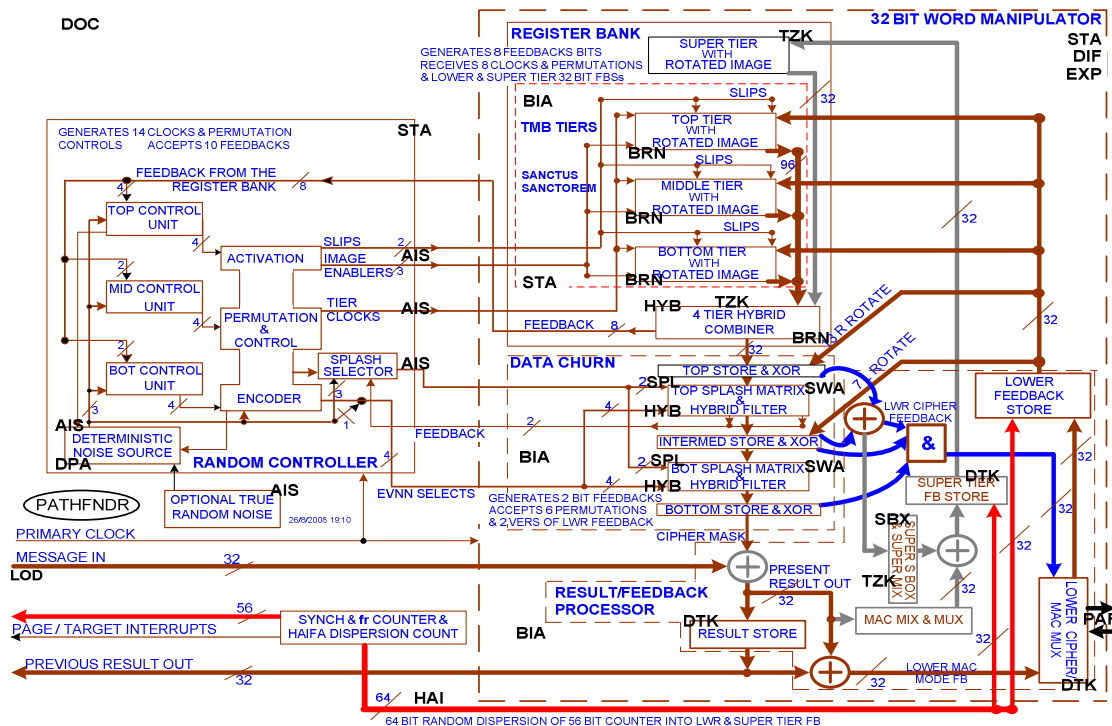


Fig. 1: A Pathfinder for Pinpointing Claim Location

Solution: In new documents [undst] we explain the simple tenets on which the ZK-Crypt unpredictability is based; to which we add support with short intuitive explanations, supporting

statistical tests, extensive proofs and improved reference documents. In the following chapters, which we will edit for the final submission, we assembled supporting graphical demonstrations, accrued statistics, in house explanations, proofs and supporting articles. We have updated the A-Z Guide, an annotated glossary with short explanations of concepts and circuits to include recent tweaks and additions.

SPL Claim: The 4 Rule Splash virtual cost free displacement Matrices present highest quality disparate quartets of "new" near neighbor inputs to each of the 32 slices of hybrid linear/nonlinear hybrid filters. The Random Rule Vectors are one of the vital building blocks that assure the graphic dispersions seen in Appendix 4. Bringing four bits from different parts of a random word serves to fragment potential string extensions shown in Chapter 3, generated by 2 of 3 Majority Filters; and also break normal correlations caused by LFSR type movement, where bit **i** is shown to have a strong correlation to the **i+1** bit from the previous clock cycle.

Rationale: We decided to include the Identity Rule Vector, where input equals output, as a device to accelerate legacy software implementations. We chose 3 random Spread Spectrum multi tapped LFSR sequences, which we would then manually repair so that in no case would the **j** slice input "into" two vectors direct to the same filter; e.g., the **I25** input into the **B** and **C** Rule Vectors would not both be displaced to the **O12** output, aka a collision.

Solution: We took 3 different "One to Many" 5 bit maximum length LFSR sequences. We manually aligned them in several instances so that visually we found only a few collisions, and then swapped values in one vector to remove the collision, preferably in a way that near neighbors did not appear; e.g., inputs **I15** and **I16** would not become output neighbors **O15** and **O16**. After a few iterations we settled on the final matrices [zk-ccc figs. 16 and 17]. Choice of which pair of vectors are chosen at each machine cycle, is dependent [zk-ccc fig. 18] on present and previous values of random binary values from the Data Churn, and one Deterministic Noise Bit generated in the Random Controller.

EXP Claims: At every clocked step of a single 9K Gate ZK-Crypt configured for data authentication; a 32 bit Message Word undergoes a random one-way expansion into an immediate average of 288 or 320 state variables in the 32 bit Word Manipulator (96 bits more in the final submission); and, simultaneously, 10 bits from the Data Churn expand into the Random Controller. The chaining value at each stage consists of the unpredictable 320 state variables of the Word Manipulator, the 62 binary variables of the Random Controller and the 56 bits of the HAIFA Counter (which are known variables, mostly zeroes, dispersed into the Lower and Super Tier Feedbacks). The last 2 phase Hash/Tag generation consists of an expanding 1024 bit Scramble (optionally with add-on second key) and a 224, 256 or 384 bit Hash/Tag expansion of the expanded 404 bit final chaining value. There is no truncation in the ZK-Crypt Hash/Tag operation.

Rationale: Truncation reduces entropy. We were determined to expand (as opposed to conventional compression and truncation) at every stage, while compressing the total Message into a 224, 256 or 384 Hash/Tag for a single engine (512 and 768 Hash/Tags for a 2 engine ZK-Crypt concatenation). Each ZK-Crypt digesting step naturally expands the unpredictability of each and every 32 bit Message Word. We decided to increase the expansion of the initial IV with an additional expansion Scramble, and likewise expand the hash digest with a second Scramble, previous to the generation of the final Hash/Tag Value.

Solution: We added two Scramble expansions; a 256 bit expansion of the final IV chaining value, and a 512 bit Scramble expansion of the final Tail Words. On the second Scramble, we allow an optional XOR Sum of a second IV or Key. The final Hash/Tag generation is also an expansion process, where the Hash/Tag generation includes an expansion of the previous word Hash Value.

BIA Claim: We found that there was no detectable bias on any state variable in the 32 bit Word Manipulator, and surprisingly, no out of range bias on bits in the intermediary word variables.

Intermediary variables with huge numbers of repeated words and no detectable bias combined nicely (no sensed cross correlation) with above mentioned state variables. Our tests show immunity to known Differential Cryptanalytic attacks [biham2], as there are no detectable differentials (biases on any state variable) and no detectable auto correlation in the sequence of clocked state variables.

Rationale: Differentials aka biased bits are the "hooks" that analysts prey on. As Biham wrote, "A stream cipher which has no high probability differentials (or even impossible differential) is expected to be immune to resynchronization attacks, related key attacks, and re-keying attacks. The last statement is even stronger for ciphers that propose authenticated encryption, where the attacker can introduce differences through the plain text or cipher text as well." The SHA-3 directives call for authentication of encrypted data, and the ZK-Crypt loading procedures hash-in data provided by users and hackers.

The Repeated Word test accurately detected biased bits in variables wherein there was no internal correlation of words. We had to know if there were distinguishing differentials, which might be caused by hashed in Messages, Keys or IVs. We wanted to see if hashing in a constant value or random data would affect bias or correlation.

Solution: We exhaustively counted the '1's in literally every state variable and in all major intermediary variables in 4 incremental test versions of the ZK-Crypt's Data Manipulator. Consistently the results were with low standard deviation. See sets of statistics in Appendix 5. We tested the effect of constant inputs and random data inputs. Tests with constant input data tested marginally better than our normal testing (with excellent results) with the Message input locked on '0'. Running the same tests on random input, not surprisingly, showed again, a miniscule improvement.

Bard showed [Chapter 2] that all binary state variables in the Register Bank and Data Churn are functions of at least 3 state variables. Bard's model showed over 55,000 monomials in "one trip through the Data Churn", and we extrapolated more than 5 million monomials if the "trip" includes the Register Bank and the Result/Feedback Processor.

Bard's analysis of the diffusion shows that if you know 319 of the 320 state variables in the Word Manipulator at a specific time, but don't know which bit you do not know; two rounds later you will, with close to certainty, not know the value of any bit with more than a probability of 0.5.

HAI Claim: The HAIFA Counter's dispersion 64 unique count bits into both Lower and Super Tier Feedback precludes herding a pseudo collision from any one of the first more than 2^{61} Message Words.

The HAIFA Counter's dispersion 64 pseudo random count bits into both the Lower and Super Tier Feedback; precludes a sequence of less than 2^{61} Cipher Mask words..

Rationale: Using the HAIFA Counter, designed to allay collision, develop designs and proofs that can both maximize the secured size of a provable hash digest sequence and also give evidence supporting the lemma that a Stream Cipher and a Hash should be "good for" safely processing more than 2^{66} binary bits.

Solution: We increased the originally proposed 24 bit HAIFA up counter [dnklmn] to the Hybrid Mersenne Prime LFSR/binary count 64 bit HAIFA Counter. The pseudo-random dispersion of the bits into the Lower and Super Tier Feedback Stores assure non-malleability of the HAIFA "whitener" and/or "salt", depending on how the viewer assesses the value of the 64 addition to the ZK-Crypt chaining value. Obviously, there can be no identical chaining values for two separate HAIFA Counts. The attacker has only one degree of freedom, the 32 bit Message, to compensate for

changes in 64 bits of two streams. A changed Message bit immediately changes the Result Store, obviating future compensation.

HYB Claim: The proprietary combination of our Hybrid MAJ/3XOR pre-filter linked to the Store & XOR correlation immunizer proved to provide a reliable non-linear barrier whose output consistently generates optimal test results; and deterministically multiplies diffusion. Each of the three filter elements is preceded by disparate randomizers; i.e., the 4 tiers of the Register Bank which randomize the Lower and Super Tier Feedback, and the Top and Bottom Splash Matrices which diffuse and amplify the diffusion of the Hybrid filter. Because of the disparate functions and the massive diffusion, we claim that the ZK-Crypt is immune to Sliding Attacks.

Rationale: Develop filtering mechanisms to integrate the 2 of 3 Majority (MAJ) functions into the Register Bank and Data Churn, after none of the standalone non-linearization methods; e.g., carry save, n of n+1 majority functions (sic 4 of 7 Majority); suggested in the literature or by reviewers, tested out satisfactorily. [diehrd][fips140] [maurer-92]. On our first cut MAJ function analyses we showed that on an average, the MAJ function amplifies bias. Subsequent more serious analyses showed that if a balanced pseudo random word is input to 32 MAJ side by side gates, natural "string extensions" of same literals occur; e.g., if a bit has value '1', with a probability of about 70%, the next bit will have value '1'. (In a balanced pseudorandom string, half of the bits are lone '1's or '0's.)

Solution: We developed non-linear (2 of 3 Majority) based filters with designed to be relatively uncorrelated inputs [zk-ccc figs. T03-T04] linked to the proprietary Store & XOR buffers, such that the combined words may show some sign of correlation, but the final output would test out with no auto-correlation.

DTK Claim: Dual Track Orthogonal Feedback in single engine or in configurations of two or more concatenated ZK-Crypt engines provably precludes Message Modification.
[Chapter 5; Appendices 1 & 2]

Inputting all 64 bit output bits of the HAIFA Counter into the two tracks of orthogonal feedback we are sure that we have generated a minimum length unique sequence of more than 2^{61} chaining value sequences.

Rationale: With the original single track of feedback, we found that, in certain circumstances, up to 28 bits could be modified and the Register Bank and Data Churn state variables reconciled to valid states with false messages, i.e., classical Message modification. This attack did not enable generation of a Hash Value or Tag, because the single feedback track was a function of the last two Result words, causing the Result Store to retain a "history" of false Messages necessary to maintain true feedback.

Therefore, while a feedback series exists in which any first false Message which corrupts the Register Bank; followed by a second false feedback word which may reconcile Register Bank and Data Churn state values; followed by any length of contrived false Message feedback words which generate valid feedback (original feedback for true Messages) which maintain the Register Bank and the Data Churn in a valid state for the remaining full length Message and Tail input; the engine is in a state that cannot generate a valid Hash Value;

the fact that false Messages could control the Register Bank and the Data Churn was a weakness, and called for a system wherein false Messages could never gain control of any part of the engine.
[Appendix 1]

An additional rationale for dual track orthogonal feedback generation could have been to create a provable links joining 2 or more ZK-Crypt engines.

We needed to find what we call orthogonal feedbacks, where, provably, if one feedback stream would corrupt and reconcile the TMB tiers, the second feedback stream would be, and continue to be corrupted by the ensuing Messages, provably maintaining the Register Bank and Data Churn in a corrupt state from the fourth corrupting Primary Clock.

We had to prove that a HAIFA Counter could be integrated in the feedback streams to provably prevent herding (finding a collision and replicating at least one more sequence), where the only degree of freedom that an attacker has is contrived Messages. We had to prove that as each chaining value is unique, with an indexed HAIFA word; the collided sequence would have to be a second sequence wherein the Message Word would have to compensate for the at least the one bit difference in the second HAIFA count.

Solution: We show intuitively, and in an exhaustive search of all of the 2^{31} possible false Message Words, that with one feedback stream where the feedback is a function of two consecutive Results (where the Result is a linear function of the Message): a) the Result Store could never be rectified to a valid state, while false Messages compensate for faulty Result Store values operative to generate valid feedback, necessary to sustain the Register Bank in a true state; thereby proving immunity to classical Message modification; b) a valid tag/hash could not be generated following at least one aberrated Message bit; c) the weakness would only appear in a hash environment with a known IV where the attacker knows the valid feedback; else the attacker would have to "guess" each and every subsequent valid feedback that would be compensated for with a false contrived Message Word.

We proved that feedback streams are orthogonal if one stream is a linear function of a present Result, and the second stream is a linear function of both a present Result and the previous Result. We further proved that with orthogonal dual feedback where one of the feedbacks is masked by a function of the Data Churn, the corruption of the Super Tier, the Data Churn and the Result/Feedback Processor is provable. The Lower Feedback can only reconcile the TMB Tiers, and can, under very limited conditions maintain the TMB tiers in a valid condition for a short time.

We prove that the value in the Result and the value in the Result Store are invalid; i.e., all Results from the first false Message, as Messages are now random words that are linear functions of the false Results and the true Lower Feedbacks (necessary to maintain the TMB in a true condition). See the graphical proofs of diffusion in Appendix 4.

The Super Tier feedback's being a linear function of the false Results means that the MAC MIX output is always false, from the first false Message Word. We prove that had the attacker been very lucky, and that the two false values reconciled the state variables in the Super Tier, then the output of the SuperMIX would be true. That cannot happen, because from that cycle onward the Super Tier Feedback must be true. If the MAC MIX is false, and the SuperMIX is true, then the Super Tier Feedback cannot be true.

If the Super Tier feedback is always false, the Register Bank Output will be false, and the Data Churn will be corrupted. The randomized 8 binary signals from the Register Bank output which regulate the TMB Tier Clocks, Brown Images and Slips will undoubtedly corrupt the TMB tiers, completing the rapid diffusion of errors.

Similarly, if we attempt to herd "colliding group" sections, where the chaining values are identical, except for the Message and the HAIFA Count which is XOR summed into the Feedback Stores, the attacker would have to reconcile both feedback streams with false Messages, an impossible task, as shown in Chapter 8.

LOD Claim: Robust initialization regimes were developed which increased security and virtually extended the effective bit lengths of Keys and IVs and precluded any tractable contrived attack.

Rationale: Loading of the first 128 bits of Key or IV directly into the Register Bank and the Random Controller, was a costly procedure, demanding 24 added scrambling procedures, as a worst case manual analysis of the "wake up" (initial activation) of all elements of the previous ZK-Crypt engines showed us. It was found that an adversary could initially load all '0' keys that could stall activation of several Random Controller non-essential elements for up to 16 clock cycles. Having a second "degree of freedom" presented a potential vulnerability.

Solution: The Global (Re)set routine was adapted to insure that no adversarial key could delay wake up of any internal element of the ZK-Crypt engine.

All Key and IV words are "hashed into" the engine via the Message Word.

TZK Claim: Assuming that the Initial Condition of the TMB is unknown; the Register Bank Super Tier masked output is a minimized 32 bit only observable output, we claim no information from the TMB tiers' Sanctus Sanctorum is leaked and that the TMB Tiers are unobservable in a keyed hash environment.

Rationale: At every stage of development we incrementally decided to enhance the unobservability of the contents of the Secret Key initiated TMB entropy reservoir- our Sanctus Sanctorem, the core of the ZK-Crypt family.

Solution: We added the Super Tier to the Register Bank, unaffected by the Random Controller, as a random mask of the TMB output.

We developed the concept of dual track orthogonal feedbacks, with unique permutations, and (in MAC mode) which is a function of a single Result, whereas the Lower Feedback is a function of a sequence of two last Results, with no detected trace correlation.

We developed a random clocking sequence of the TMB Tiers, and a non-correlated random Imaging combining sequence to augment the LFSR type feedbacks. We replaced the 3XOR combiner with the 2 of 3 Majority filter. We Imaged the combined output (rotate right 5 steps) and XORed the Image to the combined output, to fragment the long literal string outputs from the majority filter, and XOR summed the result with the Super Tier mask.

We source the "deterministic noise" to the Random Controller from the Register Bank output instead of from the Register Bank's nLFSRs, so that the only window to the TMB is from the Register Bank output.

We augmented the permutation on the Super Tier Feedbacks (Cipher and MAC mode), with Biham's Serpent bijective S Box to mask and delinearize (without loss of information) the MAC MIX permutation.

We claim to have achieved unpredictable (we like to say entropy) maximization. In other words; XORing any signal with a fair coin results in a signal with entropy/unpredictability equal to the fair coin (i.e. maximal). Likewise, the XOR between a bias and unbiased bit creates a much less biased bit, but does not lower the algebraic degree of the polynomial, thus getting around the normal tradeoff of bias (resisting differential cryptanalysis) and algebraic degree (resisting algebraic cryptanalysis).

BRN Claim: Rotating a word and sampling the XOR of the word to itself, typically provides for a cheap, effective decorrelated and debiased result, especially for outputs with correlated movement, i.e., shift register outputs. The function typically fragments long literal ('1's and '0's) strings into better statistical length strings, e.g., 1/4 of the bits in good distributions are single bit literal strings.

Rationale: Reduce left right correlation emanating from bits moving unchanged in shift registers. The linear combined TMB tier outputs showed blatant signs of auto correlation. Intuitively this is easy to understand, as close to $\frac{3}{4}$ of the bits advancing from left to right in an nLFSR move without change in the next activated clock. Cells between taps in One to Many nLFSRs [undst] always retain the same value, and tapped cells move without change on one half of the clock signals, as feedback has a probability of $\frac{1}{2}$. DieHard and other tests sensed this correlation, before we were aware of the prowess of the Repeated Word tests.

Solution: Each tier of the Register Bank was uniquely left rotated (i.e., 1, 3, 5 and 7 bits). Each of the TMB images was XOR summed to the nLFSR pair of the tier randomly about 68% of the clocks. The Super Tier's 7 bit left rotated Image is XOR summed to the outputs of the tier's nLFSRs on every clock, ensuring an excellent masking distribution to the output of the 3 Tier Combiner. The 2 of 3 Majority combined 3 tier output's Image is a 5 bit right rotated which is XOR summed to the Majority filter output. This fragments the "unruly" extended literal string outputs of the Majority filter. Single bit left rotations on Splash Matrix outputs, serve to enhance the Matrix random displacements. [zk-ccc figs. R13, T03-4 & R01-R12]

SBX Claim: By prefacing the SuperMIX twiddle transformation with the Biham Serpent bijective S Box, we delinearize the masking effect of the Super Tier MAC Mode feedback.

Rationale: The Lower and Super Tier Feedbacks are linear functions of the Messages. If a non-linear mask is applied on the Super Tier, it serves to delinearize the Register Bank output.

Solution: The full Serpent S Box delinearizes the XORed Top Store & XOR output with the output of the Top Matrix output filter which is input to the Super Tier "twiddle".

DIF Claim: Each binary state variable in the 32 Bit Word Manipulator is a diffusion of at least 3, other state variables. Each Random Controller output variable is a diffusion of internal Random Controller variables and remote processed Data Churn binary logic (not state) variables. One changed Hash Message (or Cipher Mask Cipher) bit on the first round affects at least 140 32 bit Word Manipulator state variables, and 30 out of 32 Hash Result/Cipher Mask bits in the first round. (In the final tweak, with the triple register Bottom store configuration, the equations of over 200 bits in the Word Manipulator are affected in the first round, and all 32 bits of the Cipher Mask are likewise affected.

Rationale: Our original tenets for developing a highly diffused delinearized random number generator core useful to generate a highly diffused data authenticator, cipher and true random number motivated our effort to develop crypto-complex blocks of disparate functionality with multiple inputs to each Word Manipulator state variable.

Solution: We developed Combinations of hybrid pseudo Linear Feedback Shift Registers with external word feedback and permutation control signals; random matrix displacements driving hybrid majority function/3XOR filters feeding Store & XOR outputs, and 26 interacting signals between the Random Controller and the Register Bank/Data Churn complex.

STA Claim: Using standard randomness testing on final Results; exhaustive 32 bit Repeated Word exercising on all parts of the 32 Bit Word Manipulator; statistical testing of auto correlation; exhaustive testing for trace correlation between neighboring state (word) variables; and exhaustive search for binary bias of all state variables in the Manipulator or Random Controller we have shown:

No bias in any state variable in 32 Bit Word Manipulator
No internal (auto) correlations in word state variables
No distinguishing correlations between near neighbor word state variables

No bias in state variables in the Random Controller; excepting for activators of occasional random activations; e.g., missing pulses;
No distinguishing effect of a constant input Message bit or combination of bits; on the contrary, in trillions of tests, any non-zero Message input conferred slightly better than expected expectation found with all zero Messages
No distinguishing effects on final Results when word state variables were forced into highly correlated and/or highly biased conditions.

Rationale: Previous to being introduced to the full 32 bit Repeated Word test, we relied on the industry standard [maurer-92], [fips 140] and our own automated [diehrd] random tests. None of these tests, we learned, effectively detected internal correlation of the elements of the statistical output. None of the tests adequately detected the correlation between intermediate word variables.

Solution: In 32 Bit Repeated Word tests [Chap. 2] where each of the 100 subtests is run with one of the set of 100 different random keys; each subtest generates 10 million samplings. With the original half size Data Churn, close to 12,500 (out of the 10 million) words were repeated. Whereas we learned that good statistical output would generate a range of about 11,600 repeated words, typically better than the expected statistics. (Note- if one bit is stuck on '1' or '0', the pool of possible words is halved and the number of repeated words is doubled to over 23000 repeated words.) We ran and analyzed Repeated Word tests on all word variables of the ZK-Crypt.

Random stalling of a variable, e.g., a TMB tier, causes millions of expected repeated words; we disabled "stalls" and proved virtually perfect distributions. We also checked bias, before and after disabling stalls. Repeated Word output counts detect correlation and bias, but cannot detect the cause which may be one or more bits with simple bias or one or more repeated patterns. We carefully analyzed any discrepancies in intermediate (not state – but logic forced correlation) variables. No statistic varied from a manual estimate.

As a result of the above studies, the causes of the repeated word discrepancies later led to improvements of the filters, feedback permutations, and gave additional a motivation for the dual track orthogonal feedback concept. [undst][zk-ccc figs T04 & T05][Chap.2]

PAR Claim: An error or false one bit in a single Message Bit (for hashing) or a single Register Bank output bit (for ciphering) causes next round irreconcilable massive crypto-complex errors in the origin engine and in the linked engine in concatenated ZK-Crypt implementations. On the next round, the receiving corrupted engine provably sends false word feedbacks to either the original false generator, in two engine swap configurations or to the $x + 1 \bmod n$ engine with a parallelized n configuration. The use of "shared" orthogonal feedbacks to join ZK-Crypt engine feedbacks amalgamates multi-engine configurations effectively.

Rationale: Design a provable link to support effective parallelization of n , $1 < n$, concatenated ZK-Crypt engines.

Solution: In an n engine concatenated parallelization, the x 'th mod n ($0 \leq x \leq n-1$) engine's generated Lower Feedback is solely linked to (input into) the $x+1$ 'th mod n engine's Lower Feedback Store. (Said differently, in a paired concatenation, the Lower Feedbacks are swapped.) Assuming that a L/H engine Message Word is modified, the L/H Result Store is modified, and provably cannot be reconciled; at the first round, wherein the R/H received Lower Feedback corrupts the R/H engine; any attempt to insert a false Message Word in the R/H engine to sustain valid feedback would irreconcilably corrupt the R/H Result Store.

AIS Claim: We developed a deterministic and random Noise Source generator, with a secondary multiple random and deterministic noise generator which provably (statistically) provides three unbiased

uncorrelated binary primary control signals which internally regulate permutations in the Random Controller, such that all permuting functions have exact statistics, enhancing the normal debiasing, decorrelating functions in the Register Bank and the Data Churn. When the random Noise Source is switched in, the ZK-Crypt is an ideal compact minimal power True Random Number generator, made of digital components only. As all security devices need a random number generator, the ZK-Crypt grants the designer free silicon, by replacing an analog device. We claim that the ZK-Crypt is faster, more robust and cheaper as a result.

Rationale: Every security chip needs a random number generator. Most generators are slow and are based on sampling analog components which most likely suffer from statistically poor outputs, cross talk with binary signals and ageing components. The demand for unpredictable challenges and prime numbers has lead to new stringent tests of entropy of the random signals. Apparently very few, if any popular noise sources pass the long term new AIS 31 standard, and probably none with the speed necessary for deterministic ciphering. Replacing a present random number generator with the ZK-Crypt, means the silicon devoted to random number generation on a security chip, is for free, if it replaces both the random number generator and the firmware devoted to less secure stream ciphers and data authenticators.

Solution: We derive random unpredictable noise from our proprietary random frequency modulated oscillator. We post process the Primary Clocked sampling of the oscillator driven randomizing functions with delay circuits and debiasing signals to generate 3 unbiased uncorrelated signals and one random missing clock. Deterministic randomness for the noise circuit is indirectly derived from logic variables from the Register Bank and the Data Churn. [zk-ccc N02-3 & P02-4]

DPA Claim: The ZK-Crypt is inherently resistant to Side Channel Attack aka DPA. At the rising Primary Clock signal, most of the parallel activity is immediately activated, with one or two immediate levels of power consumption, caused by random clocking of two or three of the TMB Tiers. We have made provision for maintaining constant current consumption for the period of "random stalling" of the Random Controller so that hardware implementers can reinstate the same two levels, deterring simple power attacks.

Rationale: Make a "current waster" to compensate for the lowering of power when the missing (P)Random Clock pulses idle the TMB Control Units of the Random Controller; causing an easily detectable lowering of power (Simple Power Analysis).

Solution: During the period of the stalled Random Controller, we assure that all TMB Tiers are activated; 2 of 3 TMB Images are active; and exercise flip flops in the unused TRNG Noise Controller. The final implementer will adjust the "current wasting apparatus [zk-ccc fig N06].

SWA Claim: We enable "friendlier" deployment of the ZK-Crypt to legacy systems by "locking in" the Top and Bottom Matrices D Identity Rule Vector (eliminating the random displacements). This saves time and marginal effect on statistical results. Message Diffusion is lowered by 7%.

In addition, we retain an option of random clocking of only one TMB Tier, instead of random activation of 2 or 3 tiers. This marginally increases speed, and considerably lowers power consumption. This decreases the cryptocomplexity, but does not lower the efficacy of the TMB Tiers.

Rationale: Implementing the Splash Matrix displacements is the most time consuming software function in a ZK-Crypt memory embedded realization.

Solution: We provided an optional hardware lock eliminating the random displacement in the Splash Filters, to allow accelerated software/hardware compliance.



We formally reinstated the option of single tier clocking, as opposed to 2 or 3 TMB tier activation for software implementations. [zk-ccc fig. P08]

Assorted Security Issues as Dealt with in eSTREAM.

Attacks Divide & Conquer

Immunity to divide-and-conquer attacks – Treating the cipher as a directed graph (where the cycles are all through some flip-flop, and thus, for a given cycle, the graph is acyclic), any division of the cipher into two (or more) components large enough for divide-and-conquer attacks induce a cut in the graph. It is easy to see that all cuts have information flowing from one side to the other, thus, requiring an attacker that performs divide-and-conquer attack on the cipher to guess (at least) also the information that enters the component she has chosen to attack first. By requiring that each such cut have at least ten input bits, the attacker has to guess ten new bits on every clock cycle.

Non-Linear Clocking - Tier Clocks, Slips in CUs, Slips in TMB nLFSRs, (P)Random Clock

Nonlinear clocking – The nonlinear feedback registers in the controls are clocked in an irregular way. Excluding the effect of the average 1 in 12'th missing (P)Random pulse, at each clocked cycle, with a probability of 0.5, two (of the three Top, Middle or Bottom- TMB) tiers will be activated, else all 3 are activated. The choice of which of the 3 is not activated is now an uncoloured random function, wherein all Tiers are equiprobably activated [zk-ccc P06-8]. To balance partial inactivity of the Control Units during missed (P)Random clock pulses, all three TMB tiers are clocked simultaneously.

Primitiveness

All nLFSRs are such that there is one pseudo-random sequence that covers all possible different 2^{n-1} stages.

Diffusion

In Appendix 4 we deal with massive first order diffusion – A change of a single bit in the Result buffer; e.g., fraudulently changing one bit of a Message word, will cause a massive change in the State of the machine within a single clock cycle and will affect the output of at least 27 bits (out of 32) in the cipher mask. In most scenarios all 32 Result bits are affected. The number of state variables affected varies from 145 to 174. Second order changes (after 2 cycles) will cause a full (32 bit) cipher mask coverage and a full coverage of the 32 BitWord Manipulator. (In the final submission with 3 additional correlation immunizers, diffusion is increased commensurately.)

Sparse Cipher Feedback Non-Linear & Linear

Combination of sparse and dense feedback – The sparse FB (an average 4 '1' bits in a 32 bit word recirculated in three correlated versions into 5 32 bit variables) has been shown (DieHard) to be the maximum density linear feedback which does not degrade correlation statistics in the TMB tiers. This reduces internal correlation while providing internal data flow to secure the Register Bank against divide and conquer attacks. The Super Tier receives a dense Markovian feedback, with an average 16 '1' bits in the 32 bit feedback vector. The Super Tier's dense non-linear feedback increases the complexity of data flow between the components, thus greatly increasing the protection against divide and conquer attacks. Conversely, the non-linear Super Tier Cipher feedback served to enhance the DieHard statistics and no less important obviated the contrived "Fraudulent Word" collision attack.

We have proved [Appendix 1] the generic efficacy of preventing message modification of a simple dual track feedback stream, each feedback affecting tandem variables in a word manipulator, wherein one

feedback stream is a non-linear derivation of a present Result, and the second feedback stream is simply a present Result XORed to the previous Result.

(In the final submitted version, we have removed vestiges of sparse feedback with the HAIFA Counter Whitening, Salting affect on the two feedback streams.)

Repeated Words

In [Chap. 2] we judge the ZK-Crypt against the various benchmarks.. Using Bernstein's test, we compared the results of several hundred runs on the ZK-Crypt and compared our results to the "perfect" random stream, and to a stream produced by RC4 (with 256 bit keys). We found the generated statistics to fall comfortably in the range of the published "perfect cipher". Bernstein's results averaged about 11,626 repeating DWORDS in a run of 10 million 32 bit words, the RC4 averaged 11,625 words, the ZK-Crypt II 11,628 and the enhanced ZK-Crypt III on the Result in MAC mode test around 11,615, with excellent results on state words in the Register Bank and the Data Churn.

2 of 3 Majority Function

In Chap. 3 we show why the 2 of 3 MAJ function lengthens literal strings, thereby proving auto correlations. When running the MAJ function on a string, you know that a specific bit is '1', then you know that its near right hand neighbor is a '1' with an estimated probability of 0.73.

Correlation

The ZK-Crypt has passed all the DieHard tests successfully, with very few p values closely approaching zero or one, and never any DieHard failures; and the passes the Repeated Word test with grades almost identical to the RC4. We show that we could achieve an apparently optimal number of occurrences of repeated words, (17 fewer repeated words) at a cost of 234 additional memory cells, equivalent to about 2000 gates. This of course does not prove it is secure, but rather implies that there are no easy-to-identify weaknesses. Still, as each output word depends on the last four cycles of the nLFSRs, i.e., on 12 values of the registers, that is permuted differently through several distinct permutations and nonlinear combiners. Thus, we are not aware of any correlation that can be used to distinguish the ZK-Crypt from other product given 2^{64} output words and a 2^{128} work load from any other product with similar Repeated Word sequences.

Massive first order diffusion

A change of a single bit in the Result buffer; e.g., fraudulently changing one bit of a Message word, will cause a massive change in the State of the machine within a single cycle and will affect the output of at least 27 bits (out of 32) in the cipher mask. In most scenarios all 32 Result bits can be affected. The number of state variables affected varies from 145 to 174 (over 200 in the final submission). These figures depend on which Splash vectors are selected in the cycle. Second order changes (after 2 cycles) will cause a full (32 bit) cipher mask coverage and a full coverage of the 32 bit Word Manipulator.

Bias

In Chap. 2 we review results of our exhaustive tests on the internal correlation of ZK-Crypt Result words. We also show that the Repeated Word count cannot be the only measure of merit for judging internal variables.

MAC Mode Security Features in Cipher and TRNGxxx

At the initial stages of developing the ZK-Crypt a chosen word attack was suggested against the MAC mode, see Appendix C. This has subsequently shown to be a false alarm, but prompted the designers to introduce the basis for the present ZK-Crypt with a Dual Track feedback mechanism. In short time the enhancement generated better statistics, proof of the inability of an attacker to gain partial control of the Register Bank, quick proof of resistance against Message Modification, providing the back bone for precluding collisions with the 56 bit HAIFA counter.

In earlier ZK-Crypt developments, ultimate protection against fraudulent word attack relied on the non-linear complexity of the internal state, and the fact that the six nLFSR feedback streams were an XORed combination of a Present and a Previous Message affected Result. Once changed, there is no

rectification or reconciliation, as the Result Store keeps a record of all fraudulent Message words, used to rectify a first untrue Message Word; assuming that the attacker must maintain valid feedback to sustain any reconciled condition.

Changing one plaintext bit can affect up to 170 (over 200 bits in the present submission) bits in the internal state. The problem an attacker faces when he (or she) tries to complement one (or several) bits in a ciphertext word is the fact that the effects of his (or her) actions are completely unpredictable. This follows from the fact that any change in the internal state keeps affecting the machine in the following rounds, and due to the nonlinear nature of the effect, predicting these effects cannot be done with probability larger than random.

We note that it is easy to see that changing only one word of ciphertext cannot lead to the same internal state. Thus, each attempted attack must use at least two ciphertext words. As we noted before, due to the nonlinear nature of the development of the effects of the previous message word on the internal state, it is impossible to predict the effect, and thus, to find the correct sequence of words that compensate for the first change.

The Dual Track feedback is used in all functions. Secret Cipher and MAC keys which are longer than 128 bits are "digested" using the MAC mode Initial secret Key and IV are hashed into the ZK-Engine in MAC mode. It is used in the MAC function, for which the Dual Track was designed, to preclude "Fraudulent Word" insertion. Conversely, the added randomizing functions have shortened the TRNG random initialization process.

The ZK-Crypt - MAC mode incorporates two types of feedback. The simple FB transfers the whole incoming word combined with the ciphers resultant mask and the previous word back into the cipher. The complex FB transposes the individual nibbles in the simple FB and combines them with the Super Tier's complex FB. The combination of the dual feedbacks ensures that tailored words sent into the MAC will affect, in a single round, many neighbouring bits both near and far making the calculation of the tailored false feedback impossible. See explanations and explicit proofs in Appendices C & D.

Declaration of the Submitters:

The submitters have diligently tested and searched for algebraic and logical weaknesses in the proposed design. We believe that compact highly diffused design is exponentially more robust than designs with simpler algebraic complexity. As a group we have an estimated more than 30 man years of experience in defending a hardware design from side channel attacks; and are confident that we can harden a design to be immune to future non-invasive and simple probe attacks.

References:

- [diehrd] DieHard Tests, to be found at <ftp://ftp.csis.hku.hk/pub/random/source>, 2004.
- [fips 140] Federal Information Processing Standard Publication, FIPS 140-2, NIST, Washington, May, 2001.
- [biham0] E. Biham, Due Diligence for Potential Sponsor, Haifa, Sept. 16, 2005,
- [biham1] E. Biham & O. Dunkelman, A Framework for Iterative Hash Functions, NIST Hash Forum 2006, Santa Barbara.
- [biham2] E. Biham & O. Dunkelman, Differential Analysis in Stream Ciphers, Tech Report CS2007-10-1007, Technion, Haifa.
- [dnklmn] O. Dunkelman, Personal Correspondence on HAIFA Counter, March 2008.
- [maurer-92] U.M. Maurer, "A Universal Statistical Test for Random Bit Generators", Journal of Cryptography, vol. 5 Springer-Verlag, Heidelberg, 1992.
- [maurer-03] U.M. Maurer, Advanced Technology Seminar, Engelberg, October 2003.
- [repwrdr] D.J. Bernstein, "Does the ZK-Crypt I flunk the repetition test", eSTREAM website, March, 2006.
- [robshaw] M. Robshaw, Personal Correspondence, May, 2008.
- [trichina] E. Trichina, Personal Conversations, 2005/6.
- [undst] C. Gressel et al, Understanding the ZK-Crypt, www.fortressgb.com, London & Omer, July 2008.
- [zk-a-z glos] The A-Z Guide to the ZK-Crypt, An annotated glossary, www.fortressgb.com, vers 3,
- [zk-ccc] C. Gressel, ZK-Crypt Circuit Concept Drawings, www.fortressgb.com, London & Omer, July 2008.