

Appendix 1 Single Track Feedback - Proof of Preclusion of Message Modification

Although the ZK-Crypt II has been replaced by the NIST Candidacy ZK-Crypt, understanding the failed attack on ZK-Crypt II; is immensely important. We will use the attack strategy to prove the robustness of any parallelized Dual Track implementation.

A (Repulsed) Attack Regime in the Data Authentication Function

The classic attack on a MAC algorithm is simple. Flip bits in a Message in one clocked cycle and then in a subsequent clock cycle flip the same index bit – it can happen that the second flipped bit will reconcile the falsified bit; without leaving a trace in any of the engine variables. Attacking the ZK-Crypt is more complicated, as the attacker is shooting at a moving target; the clocked tiers in the Register Bank that are falsified, right shift one cell at every clock. If the attacker complements bit(s) in a shift register, she must reconcile the bit(s) a cycle (or a small number of cycles) later in the new shifted position.

An attack of this type on the ZK-Crypt is successful if all three of the following conditions can be met:

- a) (one) falsified bit(s) complemented Message Word can be inserted followed by a second reconciliation (falsified bits tailored to new position re-complemented) Message Word in a way that a following sequence of Messages can be contrived that will reconcile all falsified bits in the device, thereby assuring that a true tag can be generated.
- b) the adversary can choose a likely candidate Message Word to falsify and subsequently generate an auspicious word containing only bits that will not cause subsequent propagation of false signals into the Random Controller, or leave traces (irreconcilable falsified bits) in the Register Bank, the Data Churn or the Result/Feedback Processor.
- c) after generation of the first falsifying/reconciling Message Word pair (or short sequence), subsequent Message Words can be generated which cause valid feedback (the same feedback sequence generated in the original Message string digest) to first reconcile and then maintain components that normally retain "historical evidence of false words" (the Store & XORs) in a valid condition, so that at the end of the Message Word string digest the binary variables will be in the true unextended condition and can generate a true Tag.

The attacker has the best chance of success, if she reconciles the falsified bit(s) on the immediately following clock cycle. For example, assume that she has falsified the LS bits in the Register Bank, and she waited 16 cycles to insert a reconciling word. On the 12th cycle the falsified bit would have corrupted the Top Left nLFSR as the moving false bit corrupted the MS nLFSR feedback bit; on the 14th cycle it would have corrupted the Bottom Left nLFSR; and on the 15th cycle it would have corrupted the Left Super Tier nLFSR. It is also mandatory that the corrupted tiers shift together for the reconciliation bit to be able to recomplement all falsified bits. The same tiers rarely rotate together for more than 5 consecutive Primary Clock cycles.

Most Message Words are valid candidates to enable two step falsifying and rectification of the Register Bank without affecting the Random Controller in the ZK-Crypt II. In the two step sequence, up to 28 bits of candidate words can be falsified without complementing the MS (internal feedback bits) of the Register Bank nLFSRs. Many combinations of the 27 or 28 candidate bits may cause irreconcilable disruption of the Splash Selector sequence. In this analysis, we assume that the adversary has chosen a most auspicious word that will corrupt the Register Bank, the Data Churn and the Result Store, on the first cycle, and reconcile the Register Bank on the next cycle. (We want to prove, later, that even if she "guessed" the best of all possible words, the attack will not work.)

The ZK-Crypt II feedback tracks are linear. In MAC mode, a complemented bit in a valid Message Word complements the same indexed bit in the clocked tiers of the Register Bank two clock cycles later. (Flipped Message Word bits are inserted into the Feedback Store, on the next clock. Two clocks later the flipped bits affect the Register Bank and the Data Churn.) Only tiers that are clocked are affected by feedback. For simplicity we assume that all four Tiers are clocked together. If the i 'th bit is complemented, at the next clock the i 'th bit is

shifted into the $i+1$ 'th cell(s). As the complemented bit is shifted into the $i+1$ 'th cell, it can simultaneously be re-complemented by a false complemented feedback bit in the next clock cycle. In this most efficient method, the second false reconciling Message Word reconciles the Register Bank immediately.

In Fig. Appx-1 we show the process of falsifying and reconciling a single moving bit. T denotes true binary values, and symbol F signifies false bit values.

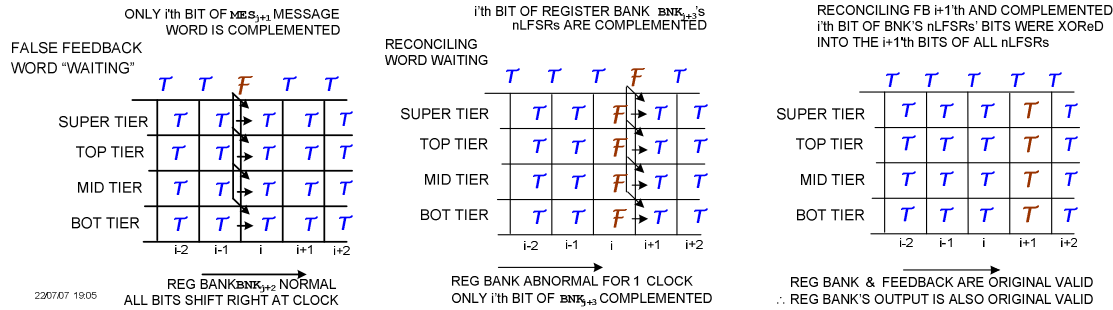


Fig. Appx-1[15]: Reconciling the Falsely Complemented Register Bank

At clock interval $j+1$, the i 'th bit of the superimposing feedback word is F , false, ready to be XOR summed with the $i-1$ 'th true, T , bits in the tiers' shift registers so that at the $j+2$ 'th clock cycle the i 'th bit of the four tiers will be F . We assume for simplicity that all tiers are clocked (will shift together and accept the superimposed feedback bits) simultaneously. The only constraint is that false Message feedback bits are generated to the same combination of clocked (shifting) TMB Tiers.

At clock interval $j+2$, the $i+1$ 'th bit of the superimposing feedback word is F , false, ready to be XOR summed with the i 'th false, F , bit in the tiers' shift registers so that at the $j+3$ 'th clock cycle the $i+1$ 'th bit of the four tiers will be T . Recall that $F \oplus F = T$.

Up to 28 bits may be complemented and reconciled in a Message, under auspicious circumstances, e.g., if the MS bit of a feedback shift register, an nLFSR, is complemented; it uniquely falsifies one nLFSR in the Register Bank in a way that defies reconciliation. If the MS bit of all tiers is complemented, four nLFSRs will be uniquely falsified; and all four uniquely falsified nLFSRs must be reconciled in a subsequent clock cycle. As all falsified feedback words can not be simultaneously reconciled, the attack fails.

In Fig. Appx-2, MAC feedback, LFB_j , generated at cycle j affects the $j+2$ 'th output of RBC, the output of the Register Bank Combiner. Single complemented feedback bits complement two to four bits in the RBC output, because of the XORed projected Images of the tiers.

A single complemented feedback bit in the Data Churn, as it descends down, diffuses with other bits and appears in the binary equations of between 20 and 28 bits of the Cipher Mask [zk-ccc Appx A] at the next clock.

At the $j+1$ 'th clock, if there are any false bits in a Register Bank nLFSRs, with a very high level of probability the RBC output will be false. This is an assumption, which has no effect on the ultimate success or failure of the attack. At the $j+2$ 'th and subsequent clocks, if the Register Bank is true, and the Random Controller is not corrupted, the RBC word is deterministically true. If at any x 'th clock the generated feedback is false, at the $x+2$ 'th clock, the Top, Intermediate and Bottom Store & XORs outputs, TSX, ISX, and CIP are, in all probability, false. This is another assumption, which has no effect on the success or failure of the attack. We know that a first falsifying word leaves a trace in at least three nLFSRs, and in at least two of the three Data Churn Stores & XORs. This knowledge has no effect on the success or failure of the attack. We know that the Result Store is deterministically corrupted when a Previous Result is false. We will prove that if valid feedback

is generated after insertion of a first false Message Word or Words in the sequence, subsequent Result Store values cannot be reconciled.

The Bottom Store & XOR output is the Cipher Mask, CIP, in the Result/Feedback Processor.

As we will show that the adversary can generate true feedback from the $j+2$ 'th clock onward, and that if the adversary has initiated the attack with auspicious false Message Words, we can assume that the Random Controller will generate true clock and permutation signals for the duration of the attack. Now, we can rightfully assume that the attack only affects the Data Churn, Register Bank and the Result/Feedback Processor, and that the Adversary's only device for implementing the attack is insertion of false Message Words in the sequence.

Aberrations from the $j+1$ 'th clocked Message Word, MES_j , appear in the outputs of the $j+3$ Cipher Mask, CIP_2 . (One clock delay waiting to be clocked into the Feedback Store- a second one clock delay waiting to be clocked into the tiers.)

If the adversary chose a word that caused an original false feedback from the Top Splash Displacement Matrix' filter; the Splash Selector, at a given clock cycle, may, with a less than 0.5 probability be in a valid state, and the attack process may succeed to proceed to the next step, or may not be in a valid state and the attack process will return to a previous state. In most instances, the Register Bank will be in a true state, and the attack process may continue. If the attack cannot continue, the attack has failed, and needs no further proof. We will always allow the attack to continue until we can prove that the attack is futile, and can subsequently prove that a valid Tag value cannot be generated with any reasonable chance of success.

In MAC mode the CIPHER/MAC MUX, outputs the MAC feedback on LFB. A MAC feedback on cycle x , LFB_x , is the XORed sum of the Present Result, NWR_x and the stored Previous Result, PVR_x .

$$LFB_x = NWR_x \oplus PVR_x = CIP_x \oplus MES_x \oplus CIP_{x-1} \oplus MES_{x-1}.$$

Generated FB Present Result NWR Previous Result PVR

Each Result is the XORed sum of the Cipher Mask and the Message Word. An adversary has one degree of freedom, the "Present" Message Word. He can change the MES_x word to reconcile the LFB_x feedback to a known value, compensating for any aberrations in any of the other three words.

An Attempted Fraudulent Message Word Attack on the ZK-Crypt II:

In the following section we outline a Message modification attack, where the MAC adversary must input a series of false Message Words, MES_{j+x} , according to the following equation, to be compliant with the classic attack.

$MES_{j+x} = CIP_{j+x} \oplus LFB_{j+x} \oplus CIP_{j+x-1} \oplus MES_{j+x-1}$ where $j > 2$, $x \geq 0$ and LFB_{j+x} is known or successfully guessed.

We show how the Message Words serve to contrive the LFB feedback words. In all cases the feedback is a fixed value. The first two feedback words falsify and reconcile, and all following feedback words must be the original true feedback word, in order to sustain the Register Bank in a valid state. In each case, one or a combination of more than one of the variables is provably invalid. In all cases, a Message Word can compensate, in order to generate valid feedback.

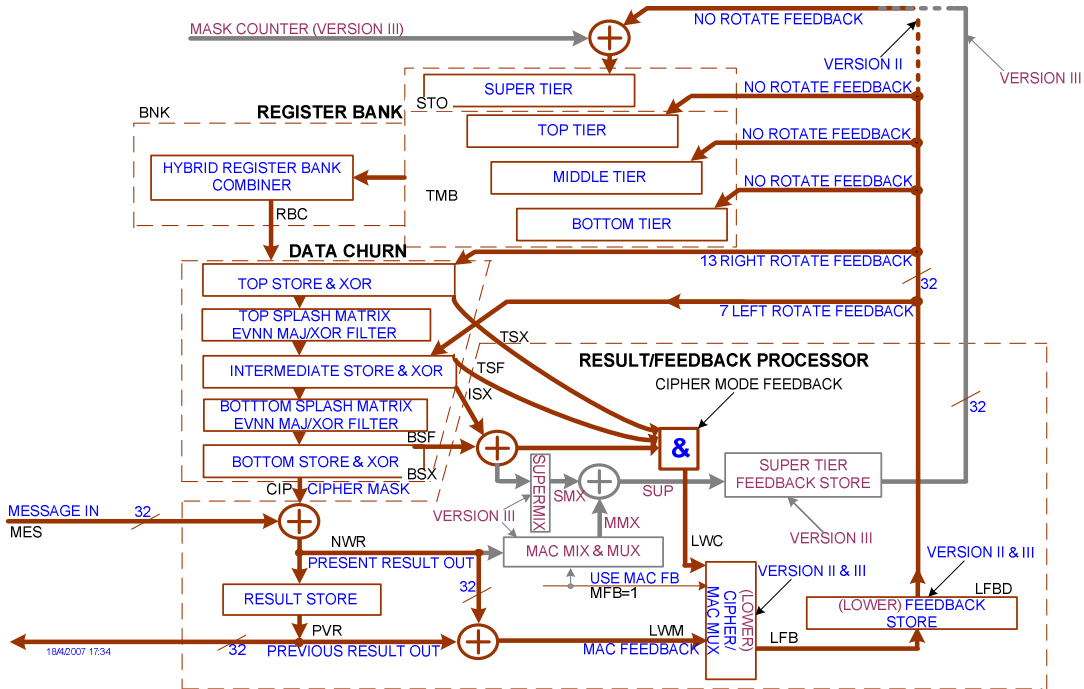


Fig. Appx-xxx2[15]: The ZK-Crypt II Feedback Strategy (SHA III-ZK-Crypt enhancements in light gray)

Start– All is well– the Engine is in a valid state– The Register Bank BNK_j is true, therefore RBC_i is true. Top Store out TSX_j is true; Intermediate Store output ISX_i is true; Bottom Store output, CIP_j , is true. The Message Word = MES_j is true; the Present Result = NWR_j is true; the Previous Result = PRV_i is true; the generated & Stored Feedback $LFB_j=LWM_j$ and LFB_j are true original "historic" values.

We denote-

- 1) All false or most probably false variable words are designated in **Bold**, e.g., CIP_{j+1} . We underline provably false variables, e.g., MES_{j+1} .
- 2) Often we cannot prove that a single word variable is false (or true), but we can prove that the expression is false, where we underline the expression, e.g., $CIP_x \oplus MES_x$.

Step I – The Adversary contrives an auspicious falsifying Message Word, MES_{j+1} , wherein between 1 and 28 bits have been auspiciously complemented as described above.

The generated feedback is false-

$$\text{Generated Feedback } \underline{LFB_{j+1}} = \text{Present Result } CIP_{j+1} \oplus \text{Previous Result } \underline{MES_{j+1}} \oplus CIP_j \oplus MES_j, \text{ and the other relevant variables-}$$

$$LFB_{j+1}, BNK_{j+1}, RBC_{j+1}, TSX_{j+1}, ISX_{j+1}; CIP_{j+1} \text{ are true.}$$

The first false feedback is "waiting to" be stored in Feedback Store LFB .

Step II – The adversary calculates a second reconciling Message Word, MES_{j+2} that generates Feedback to complement the one-bit-rotated to the right fraudulent bits that will be in the Register Bank, two clocks hence.

The reconciliation word will rectify the Register Bank (true state)-in another 2 steps. See Fig. Appx-1.

The generated feedback-

$$\text{Generated Feedback } \underline{LFB_{j+2}} = \text{Present Result } CIP_{j+2} \oplus \text{Previous Result } \underline{MES_{j+2}} \oplus CIP_{j+1} \oplus \underline{MES_{j+1}} \text{ is false other relevant variables-}$$

We know from Fig. Appx-1 that \underline{LFB}_{j+2} is false, if it must rectify false bits from \underline{MES}_{j+1} .

We know that at least one false bit in \underline{MES}_{j+2} reconciles the left most false bit in \underline{PRV}_{j+1} and one right most bit \underline{MES}_{j+2} is necessary to complement the new false rightmost position bit the Register Bank. (See later examples showing generation of Message Words that satisfy LWM_1 -)

$$LWM_1 = (NWR_i \oplus PVR_i) = (CIP_i \oplus MES_i) \oplus (CIP_{i-1} \oplus MES_{i-1}).$$

BNK_{j+2} , RBC_{j+2} , TSX_{j+2} , ISX_{j+2} ; CIP_{j+2} are still true; and,

\underline{LFB}_{j+2} is false as \underline{LFB}_{j+1} was false.

\underline{LFB}_{j+2} is "waiting" to falsely complement the Register Bank and the Data Churn.

\underline{LFB}_{j+2} is "waiting" to follow \underline{LFB}_{j+2} to reconcile the Register Bank to a true value.

Step III – In the following steps a MAC adversary must guess words (\underline{MES}_{es}) that will compensate for a false Previous Result and/or a false Cipher Mask.

In this step, \underline{LFB}_{j+2} is XORed into the \underline{BNK} and Data Churn, thereby corrupting-

\underline{BNK}_{j+3} , \underline{RBC}_{j+3} , \underline{TSX}_{j+3} , \underline{ISX}_{j+3} , & \underline{CIP}_{j+3} - (\underline{TSX}_{j+3} and \underline{ISX}_{j+3} were corrupted by \underline{LFB}_{j+2}).

The generated feedback-

$$\underline{LFB}_{j+3} = \frac{\underline{CIP}_{j+3} \oplus \underline{MES}_{j+3}}{\text{Present Result}} \oplus \frac{\underline{CIP}_{j+2} \oplus \underline{MES}_{j+2}}{\text{Previous Result}} \text{ is true.}$$

Actually, we proved that the Previous Result was false, therefore the Present Result must be false. In which case either \underline{CIP}_{j+3} or \underline{MES}_{j+3} is false or both are false.

\underline{LFB}_{j+3} is false as \underline{LFB}_{j+2} was false.

\underline{LFB}_{j+3} is "waiting" to reconcile the variables in the Register Bank to a true state.

\underline{LFB}_{j+3} is "waiting" with true Feedback, to "sustain" the Register Bank in a true state.

Step IV – In this step, reconciling feedback is XORed into the Register Bank, thereby recovering all Register Bank variables into a true state. The reconciling feedback further corrupts the Data Churn. The Register Bank Combiner is now true. The MAC adversary will continue guessing compensating words to generate "historic" original Feedback. The Combiner output is "waiting" to help reconcile the Top Store & XOR's memory. There is an increasingly lower probability of reconciliation if this step is delayed. The longest delay possible is 12 clock cycles; else one (moving) false bit will corrupt one nLFSR MS bit.

Steps III & IV can theoretically be a multi-step process that can be repeated, as described in the text.

\underline{TSX}_{j+4} , \underline{ISX}_{j+4} , & \underline{CIP}_{j+4} are still false, with a probability asymptotally approaching one.

\underline{BNK}_{j+4} , \underline{RBC}_{j+4} are now true.

The generated feedback-

$$\underline{LFB}_{j+4} = \frac{\underline{CIP}_{j+4} \oplus \underline{MES}_{j+4}}{\text{Present Result}} \oplus \frac{\underline{CIP}_{j+3} \oplus \underline{MES}_{j+3}}{\text{Previous Result}} \text{ is "once" again true.}$$

the Message Word \underline{MES}_{j+4} probably compensates three false variables.

We proved that the Previous Result was false, therefore the Present Result must be false. In which case either \underline{CIP}_{j+4} or \underline{MES}_{j+4} is false or both are false.

\underline{LFB}_{j+4} is true as \underline{LFB}_{j+3} was true.

\underline{LFB}_{j+4} is "waiting" to sustain the variables in the Register Bank in a true state.

\underline{LFB}_{j+4} is also "waiting" with true Feedback to sustain the Register Bank in a true state.

True \underline{RBC}_{j+4} , and true feedback are waiting to reconcile the Top Store & XOR.

Step V – In this step true sustaining feedback is XORed into the Register Bank. The true feedback will not adversely affect the corrupted variables in the Data Churn. True feedback and true \underline{RBC} XORed into the Top Store & XOR cause \underline{TSX} to be true. \underline{ISX}_{j+5} and \underline{CIP}_{j+5} states remain false with a probability of close to one. The MAC adversary will continue guessing compensating words to generate "historic" true original Feedback. The Combiner output is "waiting" to help reconcile the Intermediate Store & XOR's memory.

\underline{ISX}_{j+5} & \underline{CIP}_{j+5} are still false with a probability of close to 1.

\underline{BNK}_{j+5} , \underline{RBC}_{j+5} , & \underline{TSX}_{j+5} are now provably true.

The generated feedback-

$$\text{Generated Feedback } LFB_{j+5} = \underbrace{(CIP_{j+5} \oplus MES_{j+5})}_{\text{Present Result}} \oplus \underbrace{(CIP_{j+4} \oplus MES_{j+4})}_{\text{Previous Result}} . \text{ The feedback } LFB \text{ is again true.}$$

The Message Word again with a probability close to 1 compensates three false variables.

We proved that the Previous Result was false and we define the feedback as true, therefore the Present Result must be false. In which case either CIP_{j+5} or MES_{j+5} is false or both are false.

LFB_{j+5} is true as LFB_{j+4} was true.

LFB_{j+5} is "waiting" to sustain the variables in the Register Bank in a true state.

LFB_{j+5} is also "waiting" with true Feedback to sustain the Register Bank in a true state.

Step VI – If the $j+6$ 'th Splash Select is false, Step VI will fail. (We assume a poorly chosen falsifying Message Word.)

In this step true sustaining feedback is again XORed into the Register Bank. The true feedback will not adversely affect the Data Churn. True feedback and true RBC & TSX are XORed into the Intermediate Store & XOR, such that the ISX is now true. The CIP_{j+6} state is still probably false with a probability of close to 1. The MAC adversary will continue guessing compensating words to generate "historic" true original Feedback. The RBC output is "waiting" to help reconcile the CIP.

CIP_{j+6} is still false with a probability of close to 1.

$BNK_{j+6}, RBC_{j+6}, TSX_{j+6}$ & ISX_{j+6} are now true.

The generated feedback-

$$\text{Generated Feedback } LFB_{j+6} = \underbrace{(CIP_{j+6} \oplus MES_{j+6})}_{\text{Present Result}} \oplus \underbrace{(CIP_{j+5} \oplus MES_{j+5})}_{\text{Previous Result}} \text{ is again true.}$$

the Message Word again probably compensates three false variables.

We proved that the Previous Result was false, therefore the Present Result must be false. In which case either CIP_{j+6} or MES_{j+6} is false or both are false.

LFB_{j+6} is true as LFB_{j+5} was true.

LFB_{j+6} is "waiting" to sustain the variables in the BNK in a true state.

LFB_{j+6} is also "waiting" with true feedback to continue sustaining the BNK in a true state.

Step VII – In this step true sustaining feedback is again XORed into the Register Bank. The true feedback will not adversely affect the Data Churn. True feedback and true RBC, TSX & ISX are XORed into the Bottom Store & XOR, such that the CIP_{j+7} 's state is now true. The adversary has probably had to guess four compensating Message Words, and has been able to reconcile both the Register Bank and the Data Churn. This step removes all traces of a falsified Cipher Mask in the contrived true LFB feedback equation.

$BNK_{j+7}, RBC_{j+7}, TSX_{j+7}, ISX_{j+7}$ & CIP_{j+7} are all true.

The generated feedback-

$$\text{Generated Feedback } LFB_{j+7} = CIP_{j+7} \oplus \underbrace{MES_{j+7}}_{\text{Present Result}} \oplus \underbrace{(CIP_{j+6} \oplus MES_{j+6})}_{\text{Previous Result}} \text{ is again true.}$$

as the Cipher Mask is true, and the Previous Result is false, the new Message Word MES_{j+7} is provably false, if we are to sustain the Register Bank and Data Churn in a true state.

LFB_{j+7} is true as LFB_{j+6} was true.

Step VIII – Now -

$BNK_{j+8}, RBC_{j+8}, TSX_{j+8}, ISX_{j+8}$ & CIP_{j+8} are all true.

The generated feedback-

$$\text{Generated Feedback } LFB_{j+8} = CIP_{j+8} \oplus \underbrace{MES_{j+8}}_{\text{Present Result}} \oplus \underbrace{CIP_{j+7} \oplus MES_{j+7}}_{\text{Previous Result}} \text{ is again true.}$$

We assume that the adversary could continue guessing the illusive Message Word that would generate valid feedback. The Register Bank would be true, and the present and previous Cipher Masks would be true; e.g., CIP_{j+x} and CIP_{j+x-1} would be true, $x > 8$.

All equations, after Step VIII would have the same form:

$$\underset{\text{Generated Feedback}}{LFB_{j+x}} = \underset{\text{Present Result}}{CIP_{j+x}} \oplus \underset{\text{Previous Result}}{MES_{j+x}} \oplus \underset{\text{Previous Result}}{CIP_{j+x-1}} \oplus \underset{\text{Previous Result}}{MES_{j+x-1}} \text{ where } x > 8.$$

Note that if the j+8'th Message Word, MES_{j+8} , were true (the real original j+8'th Message Word), then MES_{j+7} would also have to have been true, and each Previous Result up to the j+2'th Previous Result would have had to be true. As we defined the j+1'th Message and Result to be false, we prove that from Step VII, all Message Words must be false to sustain the Register Bank and Data Churn in a valid state.

Said differently, we have proved that an adversary can reconcile the Register Bank and the Data Churn. From Step IV to Step VII, we can only prove that the Previous Result is false. From Step VII, we prove that both the Previous Result and the Message Word are false, if we are to sustain the Register Bank in a true state.

The T'th Message Word should be a meaningful Tail not a random MES_T , necessary to compensate for false MES_{T-1} .

$$\underset{\text{Generated Feedback}}{LFB_T} = \underset{\text{Present Result}}{CIP_T} \oplus \underset{\text{Previous Result}}{MES_T} \oplus \underset{\text{Previous Result}}{CIP_{T-1}} \oplus \underset{\text{Previous Result}}{MES_{T-1}} \text{ where } T > j+7.$$

A true Tail word would obviously have generated:

$$LFB_T = CIP_T \oplus MES_T \oplus CIP_{T-1} \oplus MES_{T-1}, \text{ a false feedback;}$$

Remember, the feedback corrupts the Word Manipulator 2 clock cycles later. Therefore this would corrupt the third and subsequent MAC Feedback Scramble states and all subsequent Tag states of the Register Bank, Data Churn and Previous Result.

In the tag process (see Appendix C) all Messages Words after the T'th word are, by definition, "all zeroes". The adversary has no degree of freedom.

The first MAC Feedback Scramble for a provably false Tail Word is false-

$$\underset{\text{Generated Feedback}}{LFB_{T+1}} = \underset{\text{Present Result}}{CIP_{T+1}} \oplus [00\dots0] \oplus \underset{\text{Previous Result}}{CIP_{T-1}} \oplus \underset{\text{Previous Result}}{MES_T} = \underset{\text{Present Result}}{CIP_{T+1}} \oplus \underset{\text{Previous Result}}{PRV_T},$$

because the Tail word was false;

but the second MAC Feedback Scramble will be true, as false feedback corrupts two cycles later-

$$\underset{\text{Generated Feedback}}{LFB_{T+2}} = \underset{\text{Present Result}}{CIP_{T+2}} \oplus \underset{\text{Previous Result}}{CIP_{T+1}}.$$

now $LFB_{T+2} = LFB_{T+1}$ is false,

the third MAC Feedback Scramble feedback is false, as LFB_{T+1} is inserted into BNK_{T+3} , corrupting RBC_{T+2} and the Data Churn-

$$\underset{\text{Generated Feedback}}{LFB_{T+3}} = \underset{\text{Present Result}}{CIP_{T+3}} \oplus \underset{\text{Previous Result}}{CIP_{T+2}}.$$

at this stage, BNK_{T+4} remains false as true LFB_{T+2} feedback cannot reconcile a false Register Bank.

Conclusion:

This repulsed attack shows a "weakness" in MAC mode ZK-Crypt II, despite the fact that the attack cannot succeed.

The classic fraudulent Message Word attack, as defined in the preamble of where the Register Bank and Data Churn are reconciled to a valid state after a short number of cycles does not succeed.

It is possible to sustain the Register Bank and the Data Churn in a true state during a Message Digest process with contrived Message Words. We consider this a weakness.



In Steps I & II and from Step VII of the attack, all inserted Message Words are provably false.

If the Register Bank is sustained in a true sequence following a false Message Word, the Register Store cannot be reconciled, and a valid Tag cannot be generated.