Tweak on TIB3

NIST has indicated that it will allow the 15 candidates that go to the second round to do some minor tweak on their submissions. We have decided to announce now what our tweak would be if we are selected, to give ample time for analysis. The tweak changes just one part of the underlying block cipher of TIB3, in fact just the macro "DIFFUSION" and in fact just one part of it.

We have decided to do a tweak to TIB3 because we realized that we made a mistake when calculating some of the properties of the diffusion of the cipher. This makes the underlying block cipher of TIB3 to be weak, a fact that has been shown by Mendel and Schlaffer. While this does not translate directly into a lethal attack (the Mendel-Schlaffer attack against TIB3-the-hash has complexity $2^{122.5}$ on both time and memory) it does show an undesirable property.

The basic problem with the block cipher of TIB3 is that the Sbox operates on a bit by bit basis, so one needs another way to obtain inter-bit diffusion. This was the purpose of the PHTXs and of the diffusion matrix implicit in the operations $A = A\tilde{+}G$, $G = G\tilde{+}E$ and (at the beginning of the round) $G = G \oplus C$. (in the 256 version $\tilde{+}$ meant parallel 32-bit addition). But we made a mistake with it, in that there are some differential paths with high probability that manage to skip all PHTXs, thus they are differential paths on only one bit (when the four registers are aligned one above the other), and their probability is high.

We have decided to change just the instruction $A = A\tilde{+}G$. We are replacing this with the instruction $A = (G \lll 37) \oplus (A \lll 12) \oplus (A \gg 1)$, where $\lll$ indicates rotation to the left of the 64-bit word and $\gg$ is shift to the right.

This change will ensure that one cannot find a path restricted to just one bit. We obtain a much better diffusion with this single change, although we pay a minor price in speed. For the 224/256 version, the speeds changes in the 64-bit implementation from 7.68 cycles per byte on the original version to 8.08 in the tweaked version. In the case of the 32-bit implementation, the speed changes from 12.96 cycles per byte to 14.24.

For the 384/512 case, we do an analogous thing. In this case in the instruction $A = A\tilde{+}G$, the registers are now 128-bits and $\tilde{+}$ meant the parallel 64-bit addition. We replace this with something similar to the 256 case, namely: $A = (G \lll\lll 64)^{\lll\lll 37} \oplus A^{\lll\lll 12} \oplus A^{\gg\gg 1}$, where $\lll\lll$ means rotation of the 128 bit register, while the notation $X^{\lll\lll r}$ means to rotate both 64-bits halves of the 128-bit register X by r. Similarly $A^{\gg\gg 1}$ means to shift by 1 both 64-bits halves of the 128 bit register A.

There is also a speed price to pay: the 64-bit optimized implementation goes down from 6.24 cycles per byte to 7.20 cycles per byte. The 32-bit implementation (in ANSI C) goes down from 17.68 cycles per byte to 19.36 cycles per byte. (there is an SSE implementation of the original version due to Wei Dai that achieves speed of about 5 cycles per byte. For the moment we are limiting ourselves to ANSI C implementations).

This tweak makes the underlying block cipher of TIB3 to be much stronger. Every differential path will have to pass through some PHTX or a rotation, and hence there cannot longer be a path that is restricted to only one bit. The number of active bits multiply very rapidly, and the threshold of probability 2^{-128} in the 256 case or 2^{-256} in the 512 case is quickly reached.

An attacker on TIB3 may want to choose deliberately the differences in the blocks to get something useful, but it is very difficult to deal with the long pipe structure of TIB3, because every message block is used twice, in a different way. In spite of the shown weakness of the underlying cipher, no attack better than the Mendel-Schlaffer attack has been found for TIB3, which we think shows the strength of the general structure of TIB3. The new version, with a stronger cipher, should be very secure. (there has been other attacks against the underlying block cipher, when the attacker is allowed to also choose differences in the key. These attacks have not been translated into an attack on TIB3-the-hash, again because the long pipe design protects TIB3 from them). The Mendel-Schlaffer attack does not use this approach, but rather a birthday attack to find one of many useful differences in the state. Mendel and Schlaffer show that this difference has a high probability of being canceled if no further change is made on the blocks. The tweaked version does not have this weakness.

Property: Every differential path in the underlying block cipher of the tweaked version must pass through one of the PHTXs or the rotation of A, and thus it cannot be restricted to a single bit.

Proof:

We assume the reader familiar with the round structure of TIB3.

Assume first the case in which there is no difference in the keys, only on the state.

If, after the $G = G \oplus C$ operation and mix with the key there is a difference remaining in G, it will have to pass through the PHTX that is applied to G. So we can assume that the differences at the start of the Sbox step are contained only on A, C, E, which will pass through the Sbox. It at the exit of the Sbox there is a difference on C, it will pass through the PHTX that is applied to C. If there is a difference in A, it will pass through the rotation. So, the only possible output of the Sbox is to have non zero difference only on E. That is, the difference at the output must be in each bit of the form 0010 or 0000, when viewed with respect to the words A, C, E, G. This will get transformed into a difference of type 0011 on at least one bit by the $G = G \tilde{+} E$ operation, (and maybe 0001 on other bits, because of the carry, but the attacker is trying to contain things within one bit, so assume that there is no carry). The bit with difference 0011 will be 0110 after the permutation of the words at the end of the round. Since there are no difference in the keys, this will be transformed at the start of the round in a difference of type 0111 by the $G = G \oplus C$ operation, and thus it will go through the PHTX of G. Notice that in this case we have proved that there is a passage through a PHTX or rotation at least once every two rounds.

Now, consider the case in which the difference in the state is zero and there is a difference in the keys only. Consider the first round in which there is a non-zero difference. Notice that the rounds keys of TIB3 are all of the form $\alpha\beta\gamma\beta$, thus in order not to have any difference in G (to avoid its PHTX), the difference must be of the form $\alpha 0 \gamma 0$. This mean the difference in the inputs to the 3 by 3 Sbox is of the form $\alpha 0 \gamma$, i.e., 0,1,4 or 5, but since we are assuming a non zero difference, at least one of α or γ is not zero, i.e., the Sbox has input difference 1,4 or 5. If one looks at the difference table of the Sbox (page 15 of the TIB3 submission) one sees that that 1,4 or 5 cannot go to 1, that is they all go to differences with either a component on C (thus passing through a PHTX) or in A (thus going through a rotation).

The only possibility remaining is to have both a difference in the state and in the keys. If the difference in the key cancel the one on the state in the first round, then we are as before, but with one round shifted. The fact that the first round key is not null implies necessarily that some other round key will be non null, and since we have now a null difference in the state after the first round, this reduces to the previous case.

If the first round key does not cancel the difference in the state, then we are as in the first part of the analysis of the first case: after the first round, we must have the difference in one bit of 0011, which at the start of the next round gets transformed first into 0110 and then into 0111. In order not to pass through G we must nullify it with the key. Again, since the key differences are of the form $\alpha\beta\gamma\beta$, we need a key difference of type $\alpha 1\gamma 1$, thus obtaining a difference at the start of the Sbox of the form $\alpha 0\overline{\gamma}0$. As explained above, any non zero such difference will go through a rotation or a PHTX, so both α and $\overline{\gamma}$ must be zero, (thus the key difference was 0111) and the difference at the end of the second round must be all zero. But then we are now in the following situation: the difference in the state is zero, and since the key difference in the second round is 0111, this implies that $\Delta D_3 \neq 0$, and not all following key differences can be zero. (If $\Delta RK_3 = 0$, then $\Delta D_3 \neq 0$ implies $\Delta LK_3 \neq 0$, and round 3 has a non zero key difference. If $\Delta RK_3 = 0$, then round 10 has a non zero key difference). Thus, we are in the same situation of the second case.

QED

Let's take a closer look at the case when there is no difference in the keys, just in the state. Note that in fact, any difference that goes through G will not only pass through a PHTX, but also through a rotation, since G is rotated when xored to A. This difference will remain in both G (PHTXd only) and in A (PHTXd and rotated), and at the beginning of the next round it will be in E (PHTXd) and in G (PHTXd and rotated) again. So in the case with no key differences, after any difference enters G, it becomes an engine of spreading differences and the difference in the state should grow rapidly. Any difference in A at the exit from the Sbox will get rotated and then moved to G in the next round. Any difference in E at the exit from the Sbox will be moved to C and then to G at the start of the next round. Any difference in C at the exit from the Sbox will pass through a PHTX. Any small weight difference should multiply its weight in a short number of rounds.

Just in case, we did several tests that seems to indicate that indeed this is the case. We describe here more in detail one such type of test.

We called it the Greedy Test. The test takes a set of differences and for each difference it will look for the best next difference, best in the sense of highest probability. We measured this by lower bounding the probability:

Given a difference $\Delta A, \Delta C, \Delta E, \Delta G$, then each non zero bit in the OR of $\Delta A, \Delta C$ and ΔE corresponds to an active Sbox, so each such bit contributes 2^{-2} to the probability. Also, after doing the $G = G \oplus C$ operation, the difference in G will be $\Delta G \oplus \Delta C$. This difference will be transformed by the PHTX, which is not linear. If one does the OR of the high bits with the low bits, every "1" (except for the highest bit) in the result will contribute with a probability of 2^{-1} to produce or not produce a carry in the PHTX operation. (Actually, some may contribute more because there are two sums, but lets ignore one). The result with the highest probability is the one with no carries. If we estimate that the PHTX acts in a fully linear way then we will have a certain difference

in G, say $\tilde{\Delta}G$. Each non zero bit (except for bits 63 and 31) in this difference will have a probability of 2^{-1} to produce or not produce a carry when the $G = G\tilde{+}E$ operation is used. So adding this number with the previous one, we get a number N that will help us in bounding the probability: if we take the number of non zero bits of the OR of $\Delta A, \Delta C$ and ΔE, multiply this number by 2, and add to it N, we get a number M such that 2^{-M} is a lower bound on the probability of the transition to any difference. (something similar in the 512 case).

With that metric, we evaluate the "best" next difference, i.e., in round 1 we search for the difference that will make the transition on round 2 to have the smallest difference, etc.

The differences that we tested are all the differences that will produce just one active Sbox and with a difference in G equal to that of C, so it will get canceled and not pass through the PHTX. We show how this differences evolve, and cut the computation the round before the total probability would go below 2^{-300} in the 256 case. (Recall than in that case the birthday bound is 2^{-128}). We see that at most by round 7 we get a probability of 2^{-200}. The total probability for the 16 rounds is below 2^{-1000}. We then tested for all such differences that will produce two active Sboxes and then three, although to obtain shorter files we do no show how the differences evolve but rather when we reach a probability of 2^{-140}. Again this is reached at most on round 7.

For the 512 version the test was similar, but we were able to test only all differences with one active Sbox and with 2. In this case we went further and asked for probability below 2^{-500}, although we only require 2^{-256}. Again, by round 7 at most we achieve these low probabilities.