Second preimages on MCSSHA-3

Jean-Philippe Aumasson[1] and María Naya-Plasencia[2]

[1] FHNW, Windisch, Switzerland
[2] INRIA project-team SECRET, France

Abstract. MCSSHA-3 is a hash function submitted to the NIST Hash Competition. This paper shows how to find second preimages for MCSSHA-3 with h-bit digest in $2^{3h/4}$, and collisions in $2^{3h/8}$. These observations make MCSSHA-3 ineligible for SHA-3.

MCSSHA-3 is a hash function submitted to the NIST Hash Competition. It computes a h-bit digest by

1. initializing a h-bit nonlinear feedback shift register (NFSR) with $h/8$ byte elements
2. clocking the register once with input of a message byte
3. clocking the register three times with input of the zero byte
4. repeating steps 2 and 3 for each message byte (so that each one is input only once)
5. making a finalization that consists in clocking the NFSR $h/2$ times with no message input

MCSSHA-3 uses no message length padding. Details of the specification can be found in

`http://registercsp.nets.co.kr/MCSSHA/MCSSHA-3.pdf`

Below we describe a second-preimage attack on MCSSHA-3 with 256-bit digest (it applies as well to the versions with 224-, 384-, and 512-bit digests).

The key observation is that each input of a message byte allows one to "choose" a byte of the 256-bit state state (the 32-byte NFSR); since a message byte is input every 4 clocks, one can choose $32/4 = 8$ bytes of the state after 32 clocks. Starting from a random state, with 8 message bytes one can thus choose 8 bytes of the state and gets $32 - 8 = 24$ random bytes in the other cells.

To find a second preimage of some message, one can thus

1. determine the 256-bit value of the internal state T after the message is processed (and before finalization)
2. pick a random message of at least 32 bytes and process it with MCSSHA-3 steps to get a random 32-byte state S
3. starting from S, find the 8 subsequent message bytes that give 8 register cells equal to T's after 32 clocks
4. if the state obtained after 32 clocks is not T, repeat steps 2 to 3

After step 2 the state obtained equals the target T with probability $2^{-(256-8\times8)} = 2^{-192}$, because one can choose 8 of the 32 state bytes. Hence about 2^{192} trials are necessary to find 8 message bytes that lead to T. Since finalization is message-indendent, one gets the same digest as with the first message.

Similarly, one can find collisions in $2^{192/2} = 2^{96}$, instead of 2^{128}. Both attacks apply to versions of MCSSHA-3 with 224-, 384-, and 512-bit digest: for a h-bit digest, one finds second preimages in $2^{3h/4}$, and collisions in $2^{3h/8}$ (instead of 2^h and $2^{h/2}$ ideally).