Second preimage attack on Ponic

María Naya-Plasencia

INRIA project-team SECRET, France

Abstract. Ponic is a candidate to the SHA-3 NIST competition. We present here a second preimage attack against the 512-bit version with time complexity 2^{265}. The attack exploits the fact that the round function is invertible and it works for any number of rounds.

1 Brief description of Ponic

Ponic is a shift-register-based hash function. We first briefly describe the main characteristics of Ponic which are involved in the attack. The internal state of Ponic is composed of 6 ”circular” shift registers of 128 bits each. The message blocks consist of 256 bits. The message insertion is made by xoring the 128 least significant bits of the message block to register 0 and the 128 most significant bits to register 3. After each message insertion, the round function is applied ROUNDS times. We will not detail here this function but we point out that it is invertible. Once all message blocks have been inserted, the round function is applied POST-ROUNDS times. Then, the output of the round function is used for computing the hash value.

2 Second preimage attack on Ponic

We describe here a second preimage attack that needs 2^{265} calls to the round function. This attack applies to all versions of Ponic. Then, it is faster than the generic attack for all the proposed versions with a hash value longer than 256 bits.

The attack exploits the fact that the round function is invertible.

Let M be a given message and let h be its hash value.

From the initial state, we first insert 2^{256} different messages composed of three blocks. Inserting each of these messages lead to 3 internal states. All the 3×2^{256} resulting internal states are then stored in a set $\mathcal{S}_1$.

Now, a similar computation can be performed backwards. We compute the internal state obtained for M before the last POST-ROUNDS. We then invert ROUNDS rounds until we arrive to the instant where a message block has been inserted. We then choose 2^{256} random messages of 3 blocks. We insert the 2^{256} first blocks, invert the round function until another message block has to be inserted. This must be done for all 3-block messages and leads to 3×2^{256} internal states, which form the set $\mathcal{S}_2$.

The aim is to find a pair (S_1, S_2) of internal states in $\mathcal{S}_1 \times \mathcal{S}_2$ which collide on all registers that are not controlled by the message insertion, i.e. on all registers except registers 0 and 3. These 4 registers consist of $128 \times 4 = 512$ bits, implying that a collision between both sets can be found. Let us denote by M_1 and M_2 the two messages that generate those two colliding states. We can now force the other 128×2 bits to collide by inserting an appropriate message block, m. More precisely, the 128 least significant bits (resp. most significant bits) of m correspond to the value of $S_1 \oplus S_2$ in Register 0 (resp. Register 3).

Then, the message $M_1||m||M_2$ is a second preimage of the given message M.