From: Raghvendra Rohit <iraghvendrarohit@gmail.com>
Sent: Tuesday, September 17, 2019 11:10 AM
To: lightweight-crypto
Cc: lwc-forum@list.nist.gov; Sumanta Sarkar; rsrohit@uwaterloo.ca
Subject: ROUND 2 OFFICIAL COMMENT: Oribatida
Attachments: main.pdf

Dear all,

We would like to point out that the secondary version of Oribatida i.e., Oribatida-192-96 does not achieve 128 bit security level as claimed by designers (Page 15, Table 2 of specification document), and does NOT even achieve 112 bit level security which is the primary requirement for NIST-LWC's consideration. There is a trivial key recovery attack (short description provided in the attached document) which can recover the 128 bit secret key with data complexity 2 encryption queries and time complexity 2^{96}.

We reached out to the designers with our findings, and they have confirmed the attack. Following this, they have updated the document (sent through private communication), where they corrected their security bound from 128 to 89 bits, which still does not meet the 112 bit security requirement of NIST-LWC.

Thanks and regards,
Raghvendra Rohit and Sumanta Sarkar

Trivial Key Recovery Attack on Oribatida-192-96

Raghvendra Rohit and Sumanta Sarkar

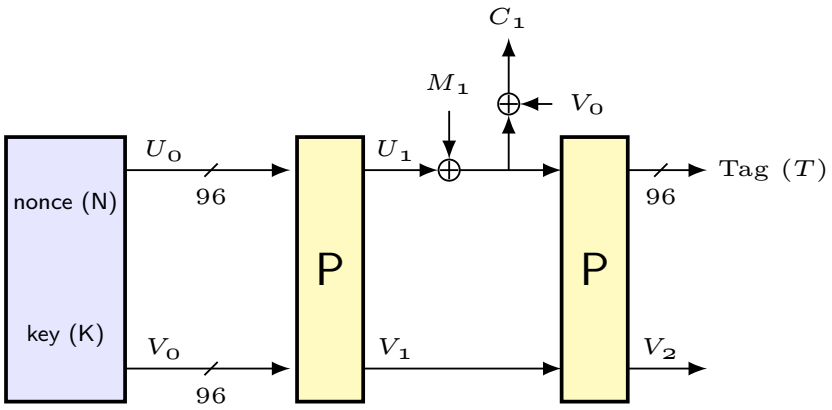

Figure 1: Oribatida-192-96 with empty AD and 1 message block

128 bit key $K = K_0\|K_1\|K_2\|K_3$ and 64-bit nonce N. Intial state $U_0 = N\|K_0$ and $V_0 = K_1\|K_2\|K_3$.

Attack details. For all 2^{96} values of V_2

1. Compute $P^{-1}(T, V_2) = (U_1 \oplus M_1, V_1)$, and U_1 (as M_1 is known)
2. Compute $P^{-1}(U_1, V_1) = (U_0, V_0)$ and match with 64 bits of N. If the match occurs, the remaining 128 bit of state is one of the possible key candidate.

The 128-bit key space is reduced to 2^{32} keys after $2^{96}\times 2$ evaluations of Oribatida-192-96 permutation. The master key can then be filtered out by doing an exhaustive search on remaining 2^{32} keys (To do this step, we need another ciphertext and tag values).

Attack complexities. Data : 2 and Time : 2^{96}

Note: The attack works for empty/non-empty AD and/or empty/non-empty message, and is independent of domain separator values.