Dear all:
Rereading my last email, I've noticed that I made a mistake. The byte that is always even is not in the ciphertext, but in the keystream. The last bit of that byte in the ciphertext is always equal to the last bit of the corresponding byte of the plaintext.
Best regards

Miguel Montes

On Sat, Apr 27, 2019 at 5:15 PM Miguel Montes <miguel.montes@gmail.com> wrote:
Dear all:
There is a problem with the implementation of the finite field multiplication used in ORANGE-Zest. The first byte of the result is always even.
Because of this, when the length of the last block of plaintext is greater than 16, byte 16 of the last block of ciphertext is always even.

Also I think the implementation of mult does not comply with the specs. The function mult, as specified, seems to swap both halves of its input. It receives V^b, V^t and returns V^t || α^c . V^b
As implemented, it simply multiplies its second half of its input.
Am I misunderstanding the specs?.
Best regards
Miguel Montes