

# Bit Balancing (or not) by table lookup

*Source:* <http://www.derkeiler.com/Newsgroups/sci.crypt/2003-03/2167.html>

---

*From:* John E. Hadstate ([nospam@null.nil](mailto:nospam@null.nil))

*Date:* 03/27/03

From: "John E. Hadstate" <nospam@null.nil>

Date: Thu, 27 Mar 2003 06:24:07 -0500

(1) Create a table of the 12,870 unique 16-bit words that have exactly 8 bits equal to 1.

(2) From this table, randomly draw 256 words and store them in a second table.

(3) Lookup incoming plaintext bytes in the table and replace the plaintext bytes with 16-bit, bit-balanced, words.

The obvious advantage is high speed (not considering the potentially slow setup time), assuming you have some reason for bit-balancing. The obvious disadvantages are inefficient use of bandwidth and the computational cost of inverting step (3).

Now, consider this variation. In step (1), instead of creating a table of the 12,870 combinations of 16 bits taken 8 at a time, create a table of the 8,008 combinations of 16 bits taken 6 at a time and use it for input to step (2).

I claim that even though the output stream will be heavily unbalanced in favor of 0's (5/8 0's, 3/8 1's) it will be impossible to exploit this fact because every output word shows exactly the same statistics.