

Help me sort though some complex math

Source: <http://sci.tech-archive.net/Archive/sci.math/2008-04/msg01700.html>

- *From:* grocery_stocker <cdalten@xxxxxxxx>
 - *Date:* Wed, 9 Apr 2008 20:21:31 -0700 (PDT)
-

The questions stem from the following URL

<http://64.233.167.104/search?q=cache:R1ERKgiy5L8J:www.phrack.org/issues.html%3Fid%3D10%26issue%3D64+p>

"--[4 - Veins' DPA-128 description

DPA-128 is a 16 rounds block cipher providing 128 bits block encryption using an n bits key. Each round encryption is composed of 3 functions which are rbytechain(), rbitshift() and S_E(). Thus for each input block, we apply the E() function 16 times (one per round) :

```
void E (unsigned char *key, unsigned char *block, unsigned int shift)
{
  rbytechain (block);
  rbitshift (block, shift);
  S_E (key, block, shift);
}
```

where:

- block is the 128b input
- shift is a 32b parameter dependent of the round subkey
- key is the 128b round subkey

Consequently, the mathematical description of this cipher is:

$f: |P \times |K \rightarrow |C$

where:

- |P is the set of all plaintexts
- |K is the set of all keys
- |C is the set of all ciphertxts

For p element of |P, k of |K and c of |C, we have $c = f(p,k)$ with $f = E'E...E'E = E'16$ and ' meaning the composition of functions.

We are now going to describe each function. Since we sometimes may need

Help me sort though some complex math

mathematics to do so, we will assume that the reader is familiar with basic algebra ;>

rbytechain() is described by the following C function:

```
void rbytechain(unsigned char *block)
{
int i;
for (i = 0; i < DPA_BLOCK_SIZE; ++i)
block[i] ^= block[(i + 1) % DPA_BLOCK_SIZE];
return;
}
```

where:

- block is the 128b input
- DPA_BLOCK_SIZE equals 16

Such an operation on bytes is called linear mixing and its goal is to provide the diffusion of information (according to the well known Shannon theory). Mathematically, it's no more than a linear map between two $GF(2)$ vector spaces of dimension 128. Indeed, if U and V are vectors over $GF(2)$ representing respectively the input and the output of rbytechain() then $V = M.U$ where M is a 128x128 matrix over $GF(2)$ of the linear map where coefficients of the matrix are trivial to find. Now let's see rbitshift(). Its C version is:"

What is a liner map between two $GF(2)$ vector spaces of 128 dimensions? Actually, I also don't see how they get 128 dimensions.

.