暗 GO の暗号化法について

暗 GO の暗号化は、①ストリーム暗号　②256bit ブロック暗号　のハイブリット方式です。
① 初期化乱数とパスワードを初期化ベクトルとした擬似乱数を生成し、それをデータに、排他的論理和で被せます。
② 256bit を、さらに、64bit ずつの 4 つのブロックに分けてブロック暗号化を行います。

②の部分は次のように行います。
まず、初期化乱数とパスワードを初期化ベクトルとして 64×4×5＝1280bit の乱数を生成します。この乱数は、すべての 256bit ブロックに対して共通に使用します。この乱数によって，64bit の 4 ブロックに対して 5 回、次のような方法で bit の攪拌を行います。

64bit の 0,1 の列のさらに 1bit 下位に 1 を継ぎ足して、65bit の 0,1 の列を考えます。これを数と見做したとき、最下位 bit が 1 ですから、奇数となります。奇数と 2^{65} は互いに素ですから、ユークリッドの互除法により、任意の奇数 x に対して

$$k \cdot x + l \cdot 2^{65} = 1 \quad (\text{k は} 0 < k < 2^{65} \text{である奇数})$$

なる k,l が存在します。この式は、2^{65} の剰余類で考えたとき、乗法についての x の逆元が k であること、すなわち、$k = x^{-1}$ であることを示しています。
したがって、64bit のデータに対して 65bit の奇数 m を考え、m に 65bit の乱数 x を掛けて $y = x \cdot m$ を作ったとき、x^{-1} を用いて、

$$x^{-1} \cdot y = x^{-1} \cdot (x \cdot m) = (x^{-1} \cdot x) \cdot m = 1 \cdot m = m$$

と、m に戻すことができます。これは、m に x をかけて y に暗号化したものを、y に x^{-1} を掛けて m に復号化できることを意味します。
実際のデータは 64bit ですから、64bit 同士の積は次のように演算します。
奇数 x,y,m の最下位の 1bit を除いた 64bit の数をそれぞれ x_0, y_0, m_0 とするとき

$$x = 2x_0 + 1, \quad y = 2y_0 + 1, \quad m = 2m_0 + 1$$

となりますから、$y = x \cdot m$ より

$$2y_0 + 1 = (2x_0 + 1)(2m_0 + 1)$$

したがって、

$$y_0 = 2x_0 m_0 + x_0 + m_0$$

さらに変形して

$$y_0 = m_0(2x_0 + 1) + x_0$$

となります。実際にコード化するときは

$$y_0 = m_0(x_0 << 1 \quad or \quad 1) + x_0$$

として演算しました。
ただし、この方法では、bit の攪拌は整数の積によって行われるため、下位の bit から上位の bit に向かって bit の攪拌が行われます。したがって、これだけでは下位 bit は攪拌されません。そのため、この演算の後、上位の bit を下位にシフトし、排他的論理和によって下位の bit に被せます。
さらに、この攪拌された 64bit のブロックを、排他的論理和によって隣の 64bit のブロックに被せます。このことを 64bit の 4 つのブロックに対して行うことを 1 ターンとすると、実装では、これを 5 ターンの繰り返しました。