1 The Rectangular Key

The key for this symmetric system is a rectangle of bits of variable size. For instance, a key might be 64 rows of 64 bits each, for a 4096 bit key. A smaller key might be 16 rows of 16 bits each, for a more usual size of 256 bits.

2 The Alphabet

The alphabet is also variable. Larger keys make larger alphabets possible. For instance, a 16 X 16 key is sufficient for encoding and decoding sequences of octal digits. A 32 X 32 bit key is sufficient for hex. A 64 X 64 bit key is sufficient for 5 bit symbols (base 32), etc.

3 The General Idea

As the plaintext is processed, the rows and columns of this key are rotated cyclically, each row and column by a calculated amount which is a function of the key's current state and the plaintext symbol being processed.

4 The Details, Some of Them

At a given moment, we can calculate permutations f and g by reading them off the key. The matrix is flattened so that the rows are concatanated, with the top row first and the bottom row last. To calculate f, we read forward from the beginning, taking a set number of bits that depends on the base (3 bits for octal encoding, for instance) and interpreting these bits as a number. For instance, we might find the (translated) octal values of 3,4,5,0,5,2,.... So far we have $f(0) = 3, f(1) = 4, f(2) = 5, f(3) = 0, ..., f(4) = 2, ...$ Note that we have to skip the second 5, since we need f to be invertible. To get g, we read the 'tape' (the flattened matrix which is now a list) backwards. We start from the end and read the bits backwards. So if the list ends with 010100, we know that $g(0) = 1, g(1) = 2$. Note that the pattern 100 is read as 1 and not 4, as if the reader were upside down.

Next the composition $h = f \circ g$ is formed, and this is used to process the plaintext symbol x. Then f and g are 'shifted' by x. Finally f and g are used equally to rotate first the rows and then columns, hopefully messing things up in a way that minimizes loops and maximizes the number of unique compositions h.

So far we've sketch what's called 'encoding' in the source code. Encryption involves rounds of encoding with the addition of reversing the text between rounds and 'autospinning' the key so that its initial state is not overused.