Description of Random Latin Square by Cipher M. J. Cowan.

The size of the character space is 127, from chr[0] to chr[126]. Characters with a number less than 32 are control characters. The printable characters are:

!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~

Use of the control characters ensures that line feed and carriage return are picked up from the plaintext, enciphered and reproduced in the decipherment.

To encipher, the master key and plaintext are entered into the form. The computer makes a number from the microtime (which is a six-figure decimal between 0 and 1). This seeds a RNG, which then outputs three hex numbers to provide the salt. The chance of repeating the same salt is 1 in a million – low enough for my purposes.

The purpose of the salt id to ensure that the key is different in every message.

The key is concatenated to the salt and hashed with the SHA512 algorithm, which produces a hash of 64 hex numbers. This hash is concatenated to the salt and the process is repeated 1,000 times. The purpose is to load-up the time for any hacker who tries to find the key by brute force.

A regular 127x127 Latin Square is generated. The first row contains the integers 0 to 126 in order. Each row beneath is left-shifted one place.

Then 127 random numbers are made from two rounds of SHA512 hash. Repeated integers are eliminated and missing integers are inserted into the gaps in a random order determined by the RNG, seeded as before. The rows of the Latin Square are shuffled with these random numbers.

The columns are shuffled in a similar way.

The seed is converted to 3 hex numbers which are placed at the start of the ciphertext.

Now a stream of random numbers is made by further rounds of hashing.
To encipher each plaintext letter:
row = next random number in the stream;
column = asc11 code of plain letter
cipher character = Latin Square[row][column] expressed as a hex number.

The cipher character is appended to the ciphertext, and the completed ciphertext is displayed.

To decipher, the master key and ciphertext are entered by the user.

The first 3 bytes are converted to a decimal seed. The salt is created as described above, together with the shuffled Latin Square and the stream of random numbers.

Decipherment is the reverse of encipherment. The random number selects the row and the decimal value of the cipher byte is sought in that row. When found, the column represents the ascii code of the plaintext.

Examples :

Note: the texts below can be downloaded from
    http://www.cryptoden.com/articles/LatinSquareApp.doc

Input Key: Latin Square
Input plaintext:
He thought he saw
His mother's Aunt
descending from a bus.
He looked again and saw it was a Hippopotomus!

Output encipherment:
indicator = 03 e6 f2
seed from indicator = 255730
salt = 362b111d153b

full ciphertext below:

```
03 e6 f2 49 67 6b 07 69 3f 57 25 4c 38 0e 0f 2d 7e 36 36 51 68 45 15
67 1d 6b 73 64 3c 23 55 6b 5c 00 7a 10 7d 54 4f 13 74 01 11 61 60 57
15 6a 58 7c 20 7e 70 1c 50 50 4d 16 47 4e 59 33 33 1f 75 65 2d 3d 1a
05 39 4f 27 55 35 47 69 38 04 3c 6c 35 24 1e 21 26 57 36 3e 5e 3d 0b
1e 13 15 11 09 1c 2d 05 45 5a 23 6c 21 42 39 1f 37 7a 08 15 70 4d
```

Note that the indicator is shown as the first 3 bytes of the ciphertext.

Repeating the encipherment with the same key a few moments later gives a completely different ciphertext because the salt has changed.

indicator = 00 f3 b7
seed from indicator = 62391
salt = 12583d6c3b79
full ciphertext below:

```
00 f3 b7 71 3e 28 7d 25 54 75 51 64 30 0d 66 50 20 4b 57 45 19 0e 69
45 28 3c 05 20 0c 05 52 21 56 25 49 68 36 02 5d 50 35 3c 1d 2c 51 25
61 30 46 7c 5f 60 20 0c 01 19 44 16 6f 6d 7c 70 11 4e 46 60 08 56 58
52 08 4d 6e 1d 4d 01 50 6c 44 49 2d 7e 12 3d 2e 66 45 42 05 0e 18 29
31 1d 46 5d 0d 2d 7d 1f 0b 18 24 7d 66 5f 30 0d 1c 5d 2e 56
```

Inserting either ciphertext into the solver with the key ‘Latin Square’ gives back the original plaintext.