K.E.G. Card Cipher

(Kolor Encryption Gate)

by Karl Zander

karl.c.zander@gmail.com

Abstract. KEG is a playing card cipher designed for use with ordinary playing cards. It's motions resemble a rotor machine but play resembles a card game. It was designed to be resistant to modern attacks both by human and machine. It's key length and strength is 52 factorial (52!). Keg's key stream is designed to be non-periodic.

Design:

KEG is a non-linear stream cipher that fits in the palm of your hand. The discard of cards and pick up upon encountering the gate color provides non-linearity to what would seem like an ordinary rotor machine construction. KEG was designed to be easy to use and easy to remember for users who want strong encryption for A-Z messages.

KEG's output resembles an ideal cipher and in testing is not distinguishable from a truly random A-Z sequence.

Specification:

State = 52 cards
Key Length = 52 cards
Gate Color = color of the first card in the keyed deck

KEG is a hand held rotor machine that feels like a card game. It consists of an encryption deck pile and discard pile.

Operational setup:

Start by creating a lookup dictionary that converts a card into it's assigned number. Such a lookup dictionary should look like this:

AC	2C	3C	4C	5C	6C	7C	8C	9C	10C	JC	QC	KC	AS	2S	3S	4S	5S	6S	7S	8S	9S	10S	JS	QS	KS
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

AH	2H	3H	4H	5H	6H	7H	8H	9H	10H	JH	QH	KH	AD	2D	3D	4D	5D	6D	7D	8D	9D	10D	JD	QD	KD
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Card values are used for modular addition of plaintext letters.

Stepping values

AC	2C	3C	4C	5C	6C	7C	8C	9C	10C	JC	QC	KC	AS	2S	3S	4S	5S	6S	7S	8S	9S	10S	JS	QS	KS
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

AH	2H	3H	4H	5H	6H	7H	8H	9H	10H	JH	QH	KH	AD	2D	3D	4D	5D	6D	7D	8D	9D	10D	JD	QD	KD
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51

Encryption Pile Stepping:

Before any encryption of plain text is done, the encryption pile is stepped.

The stepping card is always the second card in the encryption pile.

1. The stepping card’s value is looked up in the key and the color is noted.

2. If the color of the card matches the gate color then the discard pile is picked up and placed at the rear of the encryption pile. (Last card face down should be the first card meeting the last card in the encryption pile.)

3. Discard the stepping card to the discard pile facing down.

4. Step the encryption pile by removing the first card and placing it at the rear.

Finally, step the encryption pile by the noted stepping card’s value (0-51). This means repeating step #4 by the value of the stepping card.

Encryption:

Step the encryption deck by using the Encryption Pile Stepping instructions.

Encryption of the first letter in the plaintext is performed by modular addition of the letter (0-25) with the card value of the first card in the encryption pile.

Decryption:

Step the encryption deck by using the Encryption Pile Stepping instructions.

Decryption of the first letter in the ciphertext is performed by modular subtraction of the letter (0-25) with the card value of the first card in the encryption pile.

Test Vector:

Key: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51

Example message: LETUSPLAYKEGTOGETHER
Example cipher text: ONONIANLXQHEYCNUGIAA

Statistical Analysis:

kmstats26 was used in generating the following statistical information. No statistical bias was discovered and no repeating sequence of the keystream was detected.

Results from 1 million A’s encrypted by a single key:

```
Value  Occurrences  Fractions  Probability
  0        38324     0.038324   0.996424
  1        38599     0.038599   1.003574
  2        38474     0.038474   1.000324
  3        38338     0.038338   0.996788
  4        38569     0.038569   1.002794
  5        38609     0.038609   1.003834
  6        38451     0.038451   0.999726
  7        38670     0.038670   1.005420
  8        38131     0.038131   0.991406
  9        38462     0.038462   1.000012
 10        38430     0.038430   0.999180
 11        38328     0.038328   0.996528
 12        38788     0.038788   1.008488
 13        38267     0.038267   0.994942
 14        38432     0.038432   0.999232
 15        38425     0.038425   0.999050
 16        38329     0.038329   0.996554
 17        38809     0.038809   1.009034
 18        38698     0.038698   1.006148
 19        38470     0.038470   1.000220
 20        38275     0.038275   0.995150
 21        38531     0.038531   1.001806
 22        38230     0.038230   0.993980
 23        38549     0.038549   1.002274
 24        38327     0.038327   0.996502
 25        38485     0.038485   1.000610
Entropy 4.700426
Average 12.498564
IC 25.999836
Serial Correlation 0.000059
Chi-Squared Distribution 24.946409
```

Results from 1 billion A’s encrypted by a single key:

Value	Occurrences	Fractions	Probability
0	38450685	0.038451	0.999718
1	38458409	0.038458	0.999919
2	38457594	0.038458	0.999897
3	38456978	0.038457	0.999881
4	38453772	0.038454	0.999798
5	38468577	0.038469	1.000183
6	38466932	0.038467	1.000140
7	38470261	0.038470	1.000227
8	38465292	0.038465	1.000098
9	38450095	0.038450	0.999702
10	38460335	0.038460	0.999969
11	38460718	0.038461	0.999979
12	38477047	0.038477	1.000403
13	38461882	0.038462	1.000009
14	38462045	0.038462	1.000013
15	38459673	0.038460	0.999951
16	38464065	0.038464	1.000066
17	38452468	0.038452	0.999764
18	38462209	0.038462	1.000017
19	38470300	0.038470	1.000228
20	38456837	0.038457	0.999878
21	38460562	0.038461	0.999975
22	38461043	0.038461	0.999987
23	38465182	0.038465	1.000095
24	38452935	0.038453	0.999776
25	38474104	0.038474	1.000327

Entropy 4.700440
Average 12.500279
IC 25.999998
Serial Correlation 0.000000
Chi-Squared Distribution 25.000149

Bigraph and Trigraph Analysis:

Bigraphs and trigraphs were compared between 100 sets of 1 million encrypted A’s under different keys including the all neutral key and results were consistent with the A-Z output sampled from /dev/urandom (CSPRNG using ChaCha).