Handycipher: a Low-tech, Randomized, Symmetric-key Cryptosystem

Bruce Kallick
Curmudgeon Associates
Winnetka, IL 60093
curmudgeon@rudegnu.com

Handycipher is a low-tech, randomized, symmetric-key, stream cipher, simple enough to permit pen-and-paper encrypting and decrypting of messages, while providing a significantly high level of security. It combines a simple 31-character substitution cipher with a 3,045-token nondeterministic homophonic substitution cipher, and employs the insertion of randomly chosen decoy characters at random locations in the ciphertext.

1. Introduction

For several thousand years cryptography was concerned largely with developing various kinds of substitution and transposition ciphers which, through sharing a manageably sized secret key, permitted easy encryption and decryption of messages using nothing more than pen and paper. This has all changed, of course, within our lifetime and now with public key cryptosystems, employing massively powerful computers, so-called hand ciphers are for the most part interesting only to historians and hobbyists.

Yet one can conceive of circumstances in which a highly secure pen-and-paper cipher would be invaluable; for example, someone needing to send or receive a secret message might not have access to a secure computer, or might need to refrain from using one to avoid arousing suspicion that messages are being exchanged secretly. Indeed, Bruce Schneier, a cryptographer and fellow at Harvard's Berkman Center, designed the Solitaire cipher [6] used in the novel *Cryptonomicon* for such a scenario.

Moreover, apart from any consideration of potential real-world applications, it is an interesting challenge to explore how much security against a large-scale computer-based cryptanalytic attack can be achieved using nothing more than a few hours of effort with pen and paper. The problem of designing such a cipher has received little attention in the recent cryptographic literature, and Schneier's Solitaire is widely regarded as the

best serious attempt to deal effectively with this problem yet to have been devised. In this paper we describe a cipher which compares favorably in that it is a good deal easier to implement by hand, is less subject to error propagation, and needs no additional equipment besides pen and paper (unlike Solitaire which requires an ordered deck of cards).

In a seminal 1949 paper which heralded the emergence of modern cryptography, Shannon [7] observed:

> ...we can frame a test of ciphers which might be called the acid test. It applies only to ciphers with a small key (less than, say, 50 decimal digits), applied to natural languages, and not using the ideal method of gaining secrecy. The acid test is this: How difficult is it to determine the key or a part of the key knowing a small sample of message and corresponding cryptogram? ... Note that the requirement of difficult solution under these conditions is not, by itself, contradictory to the requirements that enciphering and deciphering be simple processes.

In this spirit, then, the cipher described in this paper is proposed as a candidate for a modern formulation of Shannon's acid test. Using a relatively small key (16 decimal digits more than Shannon's suggestion but still 36 bits less than the Advanced Encryption Standard 256-bit key size), Handycipher incorporates a nondeterministic encryption procedure along the lines described by Rivest and Sherman [5], and employs multiple encryption as suggested by Merkle and Hellman [3]. Combining a simple 31-character substitution cipher with a 3,045-token nondeterministic homophonic substitution cipher results in a novel system which, while easy to implement by hand, confers enough complexity to the relationship between ciphertext and plaintext and that between ciphertext and key to achieve a significant level of computational security against both statistical analysis and known-plaintext, chosen-plaintext, and chosen-ciphertext attack models.

The basic approach of the cipher is to take each plaintext character, convert it to a key-defined pattern of length five and, using this pattern as a template with one to five holes, select certain ciphertext characters from a 5 x 5 key-defined grid.

2. The core-cipher

Handycipher is based on a core-cipher which operates on plaintext strings over the 31-character alphabet $\mathcal{A}$ comprising the 26 uppercase letters {A-Z} together with the five symbols { , . - ? ^}, and generates ciphertext strings over the 50-character alphabet $\mathcal{A}'$ comprising the 50 uppercase and lowercase letters {A-Y} and {a-y}.

Some permutation of these 50 characters plus the space symbol ^ is chosen as a secret shared 51-character key K, as for example:

QjufGCtwbUSNLqHAgVDOoansIhyBKJWFdxvPk^peXMTlirYRmcE

The 50 non-space characters of K (i.e., all but ^) are displayed as a 5 x 10 table T_K by writing successive groups of ten characters into the five rows of the table, as, continuing with the example:

Q	j	u	f	G	C	t	w	b	U
S	N	L	q	H	A	g	V	D	O
o	a	n	s	I	h	y	B	K	J
W	F	d	x	v	P	k	p	e	X
M	T	l	i	r	Y	R	m	c	E

A 31-plaintext-character sub-key $\bar{K}$ is derived from K by omitting the 20 lowercase letters {f-y} and substituting {Z , . ? -} for the letters {a b c d e}, respectively

QGC,USNLHAVDOZIBKJWF?P^-XMTYR.E

and is displayed as a substitution table, $\xi_{\bar{K}}$

m:	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	,	.	-	?	^
$\xi_{\bar{K}}(m)$:	10	16	3	12	31	20	2	9	15	18	17	8	26	7	13	22	1	29	6	27	5	11	19	25	28	14	4	30	24	21	23

Then, simply by referring to T_K and $\xi_{\bar{K}}$, plaintext characters are encrypted into k-tuples of ciphertext characters by means of the following scheme:

> Regarding the first five columns of T_K as a 5 x 5 matrix comprising five rows, five columns, and ten diagonals, each plaintext character m is encrypted by first expressing $\xi_{\bar{K}}(m)$ as a five digit binary number $b_1b_2b_3b_4b_5$ and by using the position of the 1's in this number as a pattern, associating the plaintext character m with a subset of the ciphertext characters comprising a randomly chosen row, column, or diagonal. Then a randomly chosen permutation of that subset is taken as the corresponding k-tuple of ciphertext characters.
>
> For example, the plaintext character ? occupying position 21 = 10101 is encrypted into one of the six permutations of one of the twenty 3-tuples
>
> {QuG SLH onI Wdv Mlr QoM jaT unl fsi GIr Qnr jsM uIT fol Gai QsT jIl uoi far GnM}
>
> whereas the plaintext character A occupying position 10 = 01010 is encrypted into one of the two permutations of one of the twenty 2-tuples
>
> {jf Nq as Fx Ti SW NF Ld qx Hv Nx Lv qW HF Sd Hd Sx Nv LW qF}

This roughly sketched scheme is now defined more precisely as follows.

A plaintext message M is encrypted into a ciphertext cryptogram C using a 51-character key K by means of the encryption algorithm E defined as follows:

Core-cipher encryption algorithm: C ⇐ E(K,M)

First, omitting ^ the remaining 50 characters of K are displayed as a 5 x 10 ***key-table*** T_K by writing successive groups of ten characters into the five rows of the table.

The first five columns of T_K comprise a 5 x 5 ***key-matrix*** M_K and the rows, columns, and diagonals of M_K are designated R_1–R_5, C_1–C_5, and D_1–D_{10}, respectively. We refer to them collectively as ***lines***, and call two characters ***colinear*** if they lie in the same line. The 25 characters comprising columns C_6–C_{10} are said to be ***null characters***.

Also, a 31-character ***plaintext sub-key*** $\bar{K}$ is derived from K by omitting the 20 lowercase letters f-y and substituting {Z , . ? -} for the letters {a b c d e} respectively, and a simple (numerical coding) substitution $\xi_{\bar{K}}$ is applied, transforming each character m of M into the number $\xi_{\bar{K}}(m)$ representing its position in $\bar{K}$ (i.e., if $\bar{K} = p_1p_2 \ldots p_{31}$ then $\xi_{\bar{K}}(m) = i$ where $m = p_i$).

Then the following four steps are applied in turn to each character m of M.

1. A random choice is made (with equal probability of each of the 20 possible rows, columns or diagonals) between:
   - 1.1. ***Column-encryption*:** One of the five columns in M_K, say C_j, is randomly chosen (with equal probability), or
   - 1.2. ***Row-encryption*:** One of the five rows in M_K, say R_j, is randomly chosen (with equal probability) subject to the following three restrictions, where $\hat{m}$ denotes the character immediately following m in M,
     $\xi_{\bar{K}}(m) \neq 1, 2, 4, 8,$ or 16
     $\xi_{\bar{K}}(\hat{m}) \neq 2^{5-j}$, if the position of the character $\hat{m}$ in M is an odd number
     $\xi_{\bar{K}}(\hat{m}) \neq 2^{j-1}$, if the position of the character $\hat{m}$ in M is an even number
     or
   - 1.3. ***Diagonal-encryption*:** One of the ten diagonals in M_K, say D_j, is randomly chosen (with equal probability) subject to the restriction that $\xi_{\bar{K}}(m) \neq 1, 2, 4, 8,$ or 16.
2. $\xi_{\bar{K}}(m)$ is expressed as a five digit binary number, $b_1b_2b_3b_4b_5$, and if the position of the character m in M is an odd number, then
   - 2.1. If 1.1 was chosen in step 1, then for each i such that $b_i = 1$, the i-th element of C_j is chosen, yielding a subset of the five characters comprising C_j, or
   - 2.2. If 1.2 was chosen in step 1, then for each i such that $b_i = 1$, the i-th element of R_j is chosen, yielding a subset of the five characters comprising R_j, or
   - 2.3. If 1.3 was chosen in step 1, then for each i such that $b_i = 1$, the i-th element of D_j is chosen, yielding a subset of the five characters comprising D_j.

   but if the position of the character m in M is an even number, then
   - 2.4. If 1.1 was chosen in step 1, then for each i such that $b_i = 1$, the (6-i)-th element of C_j is chosen, yielding a subset of the five characters comprising C_j, or
   - 2.5. If 1.2 was chosen in step 1, then for each i such that $b_i = 1$, the (6-i)-th element of R_j is chosen, yielding a subset of the five characters comprising R_j, or

2.6. If 1.3 was chosen in step 1, then for each i such that $b_i = 1$, the (6-i)-th element of D_j is chosen, yielding a subset of the five characters comprising D_j.

(Thus for each successive plaintext character the process alternates between reading rows left-to-right or right-to-left and between reading columns and diagonals top-down or bottom-up.)

3. The elements of the subset specified in Step 2 are concatenated in a randomly chosen order. If this string, composed of 1 to 5 ciphertext characters, satisfies both of the following two restrictions, where $\bar{m}$ denotes the character immediately preceding m in M, then it is taken as $\sigma(m)$. Otherwise, Step 1 is restarted.[1]

   3.1. The first character of $\sigma(m)$ must not lie in the line used to encrypt $\bar{m}$.

   3.2. The first character of $\sigma(m)$ must be colinear with the last character of $\sigma(\bar{m})$, unless $\xi_K(\bar{m}) = 1, 2, 4, 8,$ or 16 in which case the first character of $\sigma(m)$ must be non-colinear with the single character of $\sigma(\bar{m})$.

4. So-called ***noise characters*** are randomly inserted into each $\sigma(m)$ produced in Step 3 in the following manner: following each character of $\sigma(m)$ except the first, any one of the eight characters non-colinear with that character[2] may optionally be inserted (i.e., one such character may or may not be inserted).

Finally, the strings produced in Step 4 for each character of M are concatenated forming C.

As a result of the restrictions contained in Steps 1 and 3, the resulting ciphertext cryptogram C, consisting of the string $\sigma(m_1)\sigma(m_2)\sigma(m_3)\ldots$ can be unambiguously decrypted into the plaintext message $M = m_1m_2m_3\ldots$ by means of the decryption algorithm D defined as follows:

Core-cipher decryption algorithm: $\mathbf{M \Leftarrow D(K,C)}$

C is divided into contiguous groups of characters, proceeding from left to right, discarding noise characters, at each stage grouping as large an initial segment of the remaining ciphertext as possible composed of colinear characters of M_K, then inverting the association between binary numbers and subsets of column, row, or diagonal elements invoked in step 2 of the encryption algorithm, and finally decoding that number by inverting the substitution ξ_K.

Thus each plaintext character m is encrypted by randomly choosing a line of the key-matrix M_K and representing that character's numerical code $\xi_K(m)$ by an n-tuple $\sigma(m)$ of characters lying in the chosen line, but the randomly chosen line is required to be a column in case $\xi_K(m) = 1, 2, 4, 8,$ or 16. So that in decryption it will be possible to tell where

---

[1] It's fairly straightforward to show that some combination of choices made in Steps 1 and 3 satisfying all restrictions must exist unless $\xi_K(m) \times \xi_K(\bar{m}) = 16$ for two consecutive plaintext characters, which would require the two consecutive ciphertext characters to lie in the same row. Accordingly, for each key there will be five bigrams which cannot be encrypted by the algorithm; such bigrams can be handled by hyphenating them. (See Appendix 1.)

[2] See implementation note 10.3

one encrypted character ends and the next begins, σ(m) is not allowed to begin with any character lying in the line chosen for σ($\bar{m}$). Noise characters are recognizable because although they lie outside of the line determined by the first two characters of the σ(m) being decrypted, being non-colinear with the previous character, they cannot be the beginning of the next encrypted character.

3. Example encryption with the core-cipher

Using the key-table T_K from Section 2:

Q	j	u	f	G	C	t	w	b	U
S	N	L	q	H	A	g	V	D	O
o	a	n	s	I	h	y	B	K	J
W	F	d	x	v	P	k	p	e	X
M	T	l	i	r	Y	R	m	c	E

and its associated substitution table, $\xi_{\bar{K}}$:

m:	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	,	.	-	?	^
$\xi_{\bar{K}}$(m):	10	16	3	12	31	20	2	9	15	18	17	8	26	7	13	22	1	29	6	27	5	11	19	25	28	14	4	30	24	21	23

the plaintext CATS AND DOGS could be encrypted as follows[3]:

m	$\xi_{\bar{K}}$(m)		C/R/D	σ(m)	σ(m) + noise
C	3	00011	R5	ri	rin
A	10	01010	R2	qN	qN
T	27	11011	R4	xFvW	xFvqW
S	6	00110	C3	Ln	Ln
^	23	10111	D10	GnMF	GnMF
A	10	01010	D1	Nx	Nx
N	7	00111	D6	sdT	sdT
D	12	01100	C2	Fa	Fa
^	23	10111	D4	oFfl	oFLfNl
D	12	01100	C4	xs	xs
O	13	01101	D1	nNr	nNr
G	2	00010	C3	L	L
S	6	00110	C2	Fa	Fa

yielding the ciphertext

rinqNxFvaWLnGnMFNxsdTFaoFLfNlxsnNrLFa

[3] In the third column $\xi_{\bar{K}}$(m) is expressed in binary; in the fourth column the row, column, or diagonal chosen in Step 1 is indicated.

This ciphertext would be decrypted by scanning it from left to right, deleting noise characters, and dividing it, according to the table T_K, into its constituent k-tuples and then finding each group's associated binary number, converting to decimal, and decoding by inverting the substitution ξ_K

n:	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
$\xi_K^{-1}(n)$:	Q	G	C	,	U	S	N	L	H	A	V	D	O	Z	I	B	K	J	W	F	?	P	^	-	X	M	T	Y	R	.	E

Thus, `rin` is recognized as the bigram `ri` because `n` is non-colinear with `i` (and so is a noise character) while the following `q` is colinear with `i` but lies outside the line determined by `ri`, and `ri` is associated with 00011 = 3 which is decoded as `C`.

Then `qN` is recognized as the bigram `qN` because the following `x` is colinear with `N` but lies outside the line determined by `qN`, and `qN` is associated with 01010 = 10 which is decoded as `A`.

Then `xFvqW` is recognized as the 4-tuple `xFvW` because `q` is non-colinear with `v` (and so is a noise character) while the following `L` is colinear `W` with but lies outside the line determined by `xFvW`, and `xFvW` is associated with 11011 = 27 which is decoded as `T`, and so forth.

For a slightly larger example consider the 229-character plaintext:

```
IT HAUNTS ME, THE PASSAGE OF TIME. I THINK TIME IS A MERCILESS THING. I
THINK LIFE IS A PROCESS OF BURNING ONESELF OUT AND TIME IS THE FIRE THAT
BURNS YOU. BUT I THINK THE SPIRIT OF MAN IS A GOOD ADVERSARY.  -- TENNESSEE
WILLIAMS
```

which can be encrypted by the core-cipher as, for example, this 873-character ciphertext:

```
isWxq LfdWr lMriH udfTi aGnld HSNqx soHlf LWflr MiTSs FdIvW MSWQf xWoTQ
SnMTs rQLHL STqFN nFNLf jSriT MlouM iFNnN rTaGQ fuFiq xGfrN fxnsa HIFSW
lMoGQ LGjqu fiGad ILjvs ndflQ uWusT nqMLu dMLnx sNiMQ LGuSQ fLMuv jqlFo
HqxiM FGnfq lNjaT MLvsI qFGLM nvasi adGrT invNu MoTiT jWaFv iLGqd FnLQu
Gsfja FnaHi sfxGW fLIrl SdHjQ saFTq dunLH folLl Qdnan oIxrN QjxuF nMqSI
jflqr fLaSs aoGnQ ondLG ulIxj lauIL Wqsdu Lolnj qNiqs fliMo uxivL lnMTH
TdfsQ NLvsT qIuiS anMos FnIlv dGniM Fdxvo WQISa WrNsv jMnGF iWoMN lusLQ
ndFnf NvnoG iLuLT IoWMu QjsWv HLdHQ aTxuf SGsQH Fxsio WFofl GxrNQ TdHas
iSdru laLGn dMaiQ SnGnl dLaSQ Gurfl LuHdG jMLsv joxdW svITN xnrGu fQWSM
ojQnG TqMSM WjQjx vNTir oMnMF GinrT aFsjM vfsro Hvdfx WfGuM NLFHo uFQWS
oIjlG SQafl xHSja fruLH qLNif qsQNx rGdFN alTHo fulHT QsMvL jFfaG SiFdu
jQiGa dsiqS xGIlQ iaGdL WlraI soaid IaqGn dufoQ ujiGd QFHvN xiIsT sSQHo
WSMdq ISqHn LvNQr nxqaS ifoqF inaGj oQiuN qxdLd lTiQr lMLNS HsqNv xsfiQ
lTvML svMdr xQInj oSQMq fdrWu GfQLT riQMj aNfFT uNons Iqisx fWaTl xvdfF
WnflH oFMru ilSGi vsMHL WNsdQ HqNFM qWI
```

parsed as:

```
I     T     ^     H   A  U  N   T    S  ^    M   E       , ^     T    H
isWxq LfdWr lMriH udf Ti aG nld HSNq xs oHlf LWf lrMiTS s FdIvW MSWQ fx

E       ^      P    A  S  S  A  G E     ^      O   F  ^     T     I
WoTQSnM TsrQLH LSTq FN nF NL fj S riTMl ouMiFN nNr Ta GQfuF iqxGf rNfxn
```

```
M     E         .      ^     I     ^      T      H  I     N     K   ^
saHIF SWlMoGQL Gjquf iGadI Ljvs ndflQu WusTnq ML udMLn xsNi MQL GuSQf

T      I     M   E       ^     I     S  ^      A  ^     M     E       R
LMuvjq lFoH qxi MFGnfql NjaT MLvs Iq FGLMnv as iadG rTin vNuMoTi TjWaF

C   I     L E      S  S   ^     T     H  I     N   G .     ^     I
viL GqdFn L QuGsfj aF naH isfxG WfLIr lS dHjQs aFT q dunL Hfol LlQdn

^    T    H   I    N   K   ^    L I     F  E      ^     I     S  ^
anoI xrNQ jxu FnMq SIj flq rfLa S saoGn Qo ndLGul Ixjla uILWq sd uLolnj

A  ^    P   R     O   C  E      S  S  ^    O   F  ^     B U  R     N
qN iqsf liM ouxiv Lln MT HTdfsQ NL vs TqIu iSa nM osFnI l vd GniMF dxv

I     N   G ^    O    N    E       S   E        L F  ^     O    U
oWQIS aWr N svjM nGFi WoMN lusLQnd Fnf NvnoGiLu L TI oWMuQ jsWv HL

T      ^      A  N   D  ^     T    I     M    E        ^      I    S
dHQaTx ufSGsQ HF xsi oW FoflG xrNQ TdHas iSdr ulaLGndM aiQSnG nldL aS

^     T     H  E      ^     F  I    R    E      ^     T     H  A  T
QGurf lLuHd Gj MLsvjo xdWsv IT Nxnr GufQ WSMojQ nGTqM SMWjQ jx vN TiroM

^    B U  R     N     S  ^     Y    O    U   .    ^     B U  T
nMFG i nr TaFsj Mvfsr oH vdfxW fGuM NLFH ouF QWSo IjlGS Q af lxHSj

^     I    ^    T    H  I     N   K  ^    T    H  E      ^     S  P
afruL HqLN ifqs QNxr Gd FNalT Hof ul HTQs MvLj Ff aGSiFd ujQiG ad siq

I      R    I     T    ^      O   F  ^     M    A  N    ^     I     S
SxGIlQ iaGd LWlra Isoa idIaqG ndu fo QujiG dQFH vN xiIs TsSQH oWSMd qI

^     A  ^    G O   O     D  ^      A  D  V   E      R    S  A  R
SqHnL vN Qrnx q aSi foqFi na GjoQiu Nq xd Ldl TiQrlM LNSH sq Nv xsfiQ

Y    .     ^      ^    -   -  ^     T     E      N   N   E     S  S
lTvM LsvMd rxQInj oSQM qfd rW uGfQL TriQM jaNfFT uNo nsI qisxf Wa Tl

E       E     ^     W   I     L L I    A  M   S
xvdfFWn flHoF Mruil SGi vsMHL W N sdQH qN FMq WI
```

4. Handycipher

While the core-cipher itself has proven to be remarkably robust when encrypting relatively short plaintexts (less than a few hundred characters), with increasing message length it becomes more vulnerable to statistically based hill-climbing attacks along the lines described by Dhavare, et al. [2], even though the random insertion of noise characters is designed to interfere with such attacks. Therefore Handycipher is further strengthened by randomly salting the ciphertext with additional decoys chosen from the 25 null characters of columns C6–C10 of the key-matrix.

After a message has been encrypted by the core-cipher using the key K into a ciphertext C′ (composed of non-nulls), a slightly longer string N of nulls is generated by randomly choosing from K's 25 nulls, and then N is randomly interleaved with C′ by means of the following procedure.

Salting algorithm: **R ⇐ salt(S,T)**

Given two strings $S = s_1\, s_2 \ldots s_{|S|}$ and $T = t_1\, t_2 \ldots t_{|T|}$, S is said to be ***salted*** with T by randomly interleaving the two strings, resulting in a string $R = r_1\, r_2 \ldots r_{|R|}$ constructed by the following process, which is designed to incorporate all of S while randomly interleaving successive characters of T (repeating from the beginning of T in case the end of T is reached before S is exhausted).

```
i←0   j←0   k←0
WHILE j ≤ |S|
    flip a fair coin
    IF heads
            increment i
            increment j
            IF j ≤ |S|
                ri←sj
            ENDIF
    ELSE
        increment i
        increment k
        ri←tk
        IF k = |T|
            k←0 (Issue a warning that T should be made longer.)
        ENDIF
    ENDIF
ENDWHILE
```

So that the interleaved null characters will statistically resemble the non-nulls, they are chosen in a manner designed to make close repetition of identical nulls unlikely. To accomplish all of this, Handycipher is defined as the following extension of the core-cipher:

Handycipher encryption algorithm: **C ⇐ E†(K,M)**

1. C′ ⇐ E(K,M)
2. Generate a somewhat longer string N of nulls[4] by sequentially choosing nulls at random from the set of 25, but potentially rejecting each choice with probability (6-k)/5 if that choice would duplicate the kth previous choice, for $1 \le k \le 5$.[5]
3. C ⇐ salt(C′,N)

---

[4] The exact number of nulls called for by the salting algorithm will depend on K, M, and C′ but typically will not exceed the length of C′ by more than 16%. In case N is too short, the algorithm will recycle it from the beginning (which introduces a vulnerability) and issue a warning but, of course, when encrypting by hand the required nulls can be selected as needed rather than in advance as the null-string N.

[5] This leads to 100% rejection at n=1 (i.e., consecutive identical nulls are not allowed) and lessens the probability of identical nulls occurring close to each other in N.

and the corresponding decryption is simply accomplished as:

Handycipher decryption algorithm: $M \Leftarrow D^{\dagger}(K,C)$

This algorithm is identical to the core-cipher decryption algorithm except that the phrase

proceeding from left to right,

is amended to read:

proceeding from left to right and omitting null characters,

5. Complementary keys

Regarding the last five columns of the 5 x 10 key-table T_K as a 5 x 5 ***null-matrix*** N_K we can write $T_K = M_K N_K$ where M_K is composed of the non-nulls and N_K is composed of the nulls. If for two keys K1 and K2, M_{K2} is some permutation of N_{K1} and N_{K2} is some permutation of M_{K1} (i.e., each key's nulls are the other's non-nulls) we can call these two keys ***complementary***. So for any given key there are $51 \times (25!)^2 \approx 1.23 \times 10^{52}$ complementary keys.

If $C1 \Leftarrow E(K1,M1)$ and $C2 \Leftarrow E(K2,M2)$ are core-cipher encryptions of two messages M1 and M2 using complementary keys K1 and K2, respectively, then C1 and C2 will each consist entirely of nulls if decrypted by $D^{\dagger}$ using the wrong key. This leads to several interesting consequences. If the characters of C1 and C2 are randomly interleaved then the resulting string C will be decrypted by $D^{\dagger}$ to M1 using the key K1 but decrypted to M2 using the key K2; i.e., $M1 \Leftarrow D^{\dagger}(K1,C)$ and $M2 \Leftarrow D^{\dagger}(K2,C)$.

As an amusing example, taking the previous example key as K1

`QjufGCtwbUSNLqHAgVDOoansIhyBKJWFdxvPk^peXMTlirYRmcE`

and taking a randomly constructed complementary key as K2:

`eUmDpQnTrsbRCJwIdioxhEkXOfjHLvPgVtBaS^GlFKcAYyMuWqN`

with key-table:

e	U	m	D	p	Q	n	T	r	s
b	R	C	J	w	I	d	i	o	x
h	E	k	X	O	f	j	H	L	v
P	g	V	t	B	a	S	G	l	f
K	c	A	Y	y	M	u	W	q	N

the following ciphertext, created by such a random interleaving of two core-encryptions:

```
isAhR BWxqT ehmPb qkoYI NEpKY AVwuQ hxDAr jWSKh rdabM AOBUi XsChV dgEuO klnjK cgLVt kKdHq DgBoS XkKrL
jpUED hIgUS xKAlg nhdPM HbqYv LmSWi qsnhR uCHJL mfqBi xwsDF dgOMl TNATj UYdbV LtvUO ohSxf ekVAi QCEsT
idxoV kWetv jAVQS Fjdka omPMX ilwne hOkER MKPhR FSJXJ GjYTD BxtAm NFMSC ekHdp fVEye VSlwR Nyejx Gkstf
JiqLX OUASX WsLgM FhNDi ctGbA TdSUK cTYoy ACDUH Egshf imAxw gvhkg hBmEJ utXHo DEndj klStQ BMPfV yoNmE
rdWLx XfkCh XOdUc FipEO JMXkh KCUxG MFndu RCYUC BJKnh YWGBC UgSVc KqGEW BMhnf FmJtY NXwVS HPLGV lkyfR
oGHrx tNSBV xNgks QHmaA TdCaF Rgfqi EhYsm ULvHs cECQF uJcjT QyHAJ sMSUa KBCPh KbLoe fdWqd IVxnC wRjEX
lYIxi NcuAK yLlFe DNWmT aOjbx vAdUC KUBJh waDCS iGjRw byhPp fxcmE OLiJS GTaPN emvcO mkOEn PHyfF DEbeA
uluiL vnhXo lCMEP iBdln QNiSG Patdh gQTsG HIVAs cgDoh nNySu LbHVc XeVdx scaiwx afbNe tRlBV nPdWX uCVsY
LNcFy MKjLA JgXsK bSOTA dCPlj hxbIK eugPG VjLjr XeMnk hANVC akQTs dEPWt PVLSL FKHqm OkFft aReTN WnwgU
VCDNT gFANh lkRjs GFtiM nevac OAcmj FouGk qMndV IcjnE xNUsp mxCjU BKAcy uYfKd GexbQ SQpoq adgkq ABGiC
DgAVn vogSJ YXKWd vgjaD JXVRv ngKsl PoSmo hFYVk fHoTS mJCRm sJwHv iCLuo lFRgq VctHY ECXhk KLGSJ kglGQ
VCfPV IjqAO utbLK CaXtD iCTXB txbPl kEVrv PBhsg dulnL DNajU cqTWE MShOX EQPsg UEgoJ tlYMr TioND uiEqC
dSyMr PHDUs RdhTm BQpoY hPKXC GnfBq MQYpw iVXve hRouN xkmmU DgrXl BHhaF AwccO mfqek CWfFH AlFmR tTHyo
MdkQe SVLYW frvuR jlEIc edxgR SyCDO gkEHN VLSpi Ggstd VPcyS KYHDC EScUh tqVrc LdvgE kyWem PFxnU IOhso
jaLeq vnWEI bTLPC JRPUu GupMN DleMq OHaNS VfYEd kJGiS PtLar fWMMl bYEir xrVCP KACBN KXhbK AGnVm aNJjC
rmCxu AVXJT DxIpD cqhFH DLnAk euqIt DXOkN EwlXe SDpUO AjIan YiFJX NPkny ldHtf sJCdT KygAm SHkpS EQVCV
kJogF IuqkT HTnsd oKpPw RCLKh jsQXB GUfCJ uOdij tGRSh qBYij dAkCL PlNyD ugERm nkAhP Mbjmw DvAhJ EsiUc
nVIFa opuGL bfjHQ aHNqa SLiWS oTnQi rxNTN GaFL
```

is decrypted using K1 as:

```
I WANDERED LONELY AS A CLOUD THAT FLOATS ON HIGH O-ER VALES AND HILLS, WHEN
ALL AT ONCE I SAW A CROWD, A HOST, OF GOLDEN DAFFODILS - BESIDE THE LAKE,
BENEATH THE TREES, FLUTTERING AND DANCING IN THE BREEZE.
```

but the same ciphertext when decrypted using K2 yields:

```
THIS IS THE FOREST PRIMEVAL. THE MURMURING PINES AND THE HEMLOCKS, BEARDED
WITH MOSS, AND IN GARMENTS GREEN, INDISTINCT IN THE TWILIGHT, STAND LIKE
DRUIDS OF ELD, WITH VOICES SAD AND PROPHETIC, STAND LIKE HARPERS HOAR, WITH
BEARDS THAT REST ON THEIR BOSOMS.
```

Another consequence of the interrelationship between complementary keys is the possibility of using such a key pair with the core-cipher instead of randomly generating the null characters called for in the Handycipher encryption algorithm. This can be accomplished by using a complementary key pair K1 and K2 to encrypt a message M in the following way.

First split M into two parts, M1 and M2, and encrypt each with the core-cipher using K1 and K2, generating ciphertexts C1 and C2 respectively. Then randomly salt C1 with C2 with the salting procedure as C ⇐ salt(C1,C2) and take the resulting character string C as the encryption of M. In essence, C is the string that would result if C1 were to be encrypted with Handycipher using K1, but using C2 in place of the random string of nulls generated in Step 2 of the encryption algorithm, and likewise for C2 and K2, *mutatis mutandis*. Now C will be decrypted to M1 by Handycipher using K1, and to M2 using K2, and then joining M1 and M2 will yield M.

A slight complication arises because, due to the random nature of the salting algorithm, one cannot predict how much of C2 will actually be used in salting C1. If C2 is too long and not entirely consumed in salting C1, then some of M2 will be lost in decryption; if C2 is too short then additional K1-nulls will have to be spliced on at its end, which will be decrypted as gibberish added at the end of M2. Therefore, by the choice of where to split M and by adjusting the amount of noise inserted in the two core-cipher encryptions, the length of C2 should first be arranged not to exceed 84% of C1's length, and then it should be lengthened by adding enough randomly chosen K1-nulls to make it at least 16% longer than C1. Also, some distinctive marker such as "finis" should be added at the end of M2 to demarcate unambiguously the beginning of the added gibberish.

Modifying the Handycipher encryption algorithm in this way has two beneficial effects (although at the cost of requiring two 51-character keys instead of one): first, it reduces the cipher's expansion factor by requiring many fewer nulls to be inserted than would otherwise be in Step 2, and second, it avoids the potential vulnerability caused by the nulls (distributed evenly in terms of aggregate numbers) tending towards their expected frequency value, while the non-nulls diverge from their expected value to a statistically significant degree.

Encrypting the 229-character Williams quote example from Section 3 in this way, using the same two complementary keys, yields the following 885-character ciphertext:

```
islxq LfWir GQfTu WurdH onWLI xQxNr Wofiq sLdou vsFjM dLalI jSWSM QudSd Gaviu oNifx sdLWo FdsaL TajiN
MFluL QnIqT aisIn orudL lMWho SxSlL ldniu xolsH FdiaG TvouN QoxWM alLus diSoS WsQuG fljQx MnrfL Wrxqo
isLdl xrjQn NjuQF GilTv rjHsf odFlx FarfM WGxrN TMirl xQnrq ldQfj uLQSo MWqLj uivQo uqxif TLMoa srnxd
vLGFu qnIan oLFTa NQjGN uWVQl SjFld NLnGn qiulL nuHqa GdoSQ ulSIj xQuGs frWoS QsnLf arvNa jNTMi ljFaT
GsLMS juIWh qTHsF IWNMq GnjdM SIlno IsiIn MFGLn xvndu ovNri lSfis rxvoL ufGul SIxja WdMdS QWHLo lqjxI
lsFnS GaTWx FLvuQ fvGFx ldnov xrQnW GfjQJ tPyhD EXCEk OBRcX meEKA EYXDc hKtPb VEKQm PYgKw PbCRb UmkKm
JcOUR EgwgA DUXED PpcmJ tGmkg pkKtJ cXPeC tekhA gRhBE CtPgX AObeP kKJyE OAbJP bOgAm pEgbR AUXER JwbCm
AECJc pOhAk eYthR JOykb REcCU YOhgR whDCP hmpkg VEJbc CUECy YXgJD hOXEB DmpeU cREPC XKDUy mPOUg bRcAY
KXKCO UAYcE gRyPO AbKeY bPpcK pVtVE cgkAJ mUgRc OJmUm ewVcp PtVyb YgEKc DVPtD hbEJh bKePt OUVbA hPEDy
OhykY CRmDc tVPJO PcBEX KmwVO YJtDe mDpkO hPcRB UBXPC KgtyK AXtmy bJVeh YtBPU gymJC YCDtE ghpRJ cAKep
XOgmV BAYUm EkhwX bchPt KYbhk tVDAc ytKJP
```

which is decrypted by the first key as

```
IT HAUNTS ME, THE PASSAGE OF TIME. I THINK TIME IS A MERCILESS THING. I
THINK LIFE IS A PROCESS OF BURNING ONESELF OUT AND T
```

and by the second as

```
IME IS THE FIRE THAT BURNS YOU. BUT I THINK THE SPIRIT OF MAN IS A GOOD AD-
VERSARY.  -- TENNESSEE WILLIAMSFINISU--JDBIJ,IYPQCVC,-Y^VVZEXCJ
```

Thus dividing the plaintext into two parts and using complementary core-encryptions of each in place of random null-strings for the other, yields a ciphertext only 12 characters longer than the 873-character core-cipher encryption of the same plaintext in Section 3, yet with no less security than that achieved by a full Handycipher encryption which would require more than 1,600 characters.

6. Deniable encryption

The ploy, considered in the previous section, of using a pair of complementary keys and randomly interspersing the two ciphertexts produced by encrypting a different message with each key can be exploited to achieve a ***deniable encryption scheme*** as defined by Canetti, et al [1]

> Consider a situation in which the transmission of encrypted messages is intercepted by an adversary who can later ask the sender to reveal the random choices (and also the secret key, if one exists) used in generating the ciphertext, thereby exposing the cleartext. An encryption scheme is deniable if the sender can generate 'fake random choices' that will make the ciphertext 'look like' an encryption of a different cleartext, thus keeping the real cleartext private.

A deniable encryption scheme can be based on Handycipher in the following way. Let us suppose that M2 is a real message to be encrypted as a ciphertext C, using a real secret shared key K2. However, in case an adversary might be able to intercept C and force the sender (or the recipient) to reveal both message and key, it would be desirable to be able to provide an "innocent" fake message M1 and fake key K1 such that M1 would also be encrypted as the same ciphertext C, using the fake secret shared key K1.

To accomplish this a key complementary to K2 is randomly chosen as K1, and each message is core-cipher encrypted using the corresponding key. I.e.,

$C1 \Leftarrow E(M1,K1)$ and $C2 \Leftarrow E(M2,K2)$.

Then the characters of C1 and C2 are randomly interleaved by salting C1 with C2,

$C \Leftarrow salt(C1,C2)$

and C will be decrypted as the real message by Handycipher using the real key, but decrypted as the fake message using the fake key. I.e.,

$M2 \Leftarrow D^{\dagger}(C,K2)$ and $M1 \Leftarrow D^{\dagger}(C,K1)$

Of course, the same concerns about adjusting the relative lengths of C1 and C2 considered in Section 5 apply here as well, so that the core-encryption of the fake message must be salted with that of the real message to which enough fake key nulls have been added.

It should be noted that disclosing the fake key would reveal the null set of the real key so that all the real key nulls could then be dropped from the ciphertext, reducing the security of the real encryption to that afforded by the core-cipher alone.

7. Extended Handycipher

Extended Handycipher operates with the same plaintext and ciphertext alphabets, and encrypts a message M using a key K by first generating a random session key K′, and encrypting M with Handycipher using K′ to produce an intermediate ciphertext C′. K′ is then encrypted with Handycipher using K, and then embedded in C′ at a location based on K and the length of M, producing the final ciphertext C.

Extending Handycipher in this way confers several advantages in security at little computational cost. Because each plaintext message is encrypted with a different randomly

generated session key, the primary secret key is less exposed to any attack that depends on having a lot of ciphertext to work with, and the security of the cipher is less compromised by encrypting multiple messages with the same key.

Extended Handycipher encryption algorithm: **$C \Leftarrow E^*(K,M)$**

1. Generate a random 51-character *session key* K' with associated table $T_{K'}$ and coding substitution $\xi_{\bar{K}'}$.
2. Transcribe K' into plaintext characters by inserting a "-" before each run of uppercase letters and a "." before each run of lowercase letters, and then changing all lowercase letters to uppercase.[6]
3. Encrypt the transcribed K' with Handycipher and K, yielding K''. Adjust K'' if necessary ensuring that for the last character m of the transcribed K' to be encrypted, no null characters are interspersed with $\sigma(m)$ and that K'' terminate with exactly one null character.[7]
4. Encrypt M with Handycipher and K', yielding C'.
5. Adjust C' if necessary, by inserting more nulls, ensuring that $|C'| + |K''| \geq 700$ and also that $N \geq 30 - R$ where $|C'| = 31 \cdot N + R,\ 0 \leq R < 31$.
6. Calculate $j = \lfloor (|C'| + |K''| - 700) / 31 \rfloor \cdot \{[\xi_{\bar{K}}(A) + \xi_{\bar{K}}(B) + \xi_{\bar{K}}(C)] \bmod 31\} + [\xi_{\bar{K}}(D) + \xi_{\bar{K}}(E) + \xi_{\bar{K}}(F)] \bmod 31$.[8]
7. Insert K'' into C' immediately following position j as calculated in step 6, yielding C.

Extended Handycipher decryption algorithm: ***$M \Leftarrow D^*(K,C)$***

1. Calculate $j = \lfloor (|C| - 700) / 31 \rfloor \cdot \{[\xi_{\bar{K}}(A) + \xi_{\bar{K}}(B) + \xi_{\bar{K}}(C)] \bmod 31\} + [\xi_{\bar{K}}(D) + \xi_{\bar{K}}(E) + \xi_{\bar{K}}(F)] \bmod 31$

   and begin decrypting the substring of C immediately following position j with Handycipher and K.
2. Transcribe the letters between "." and "-" into lowercase and delete those symbols.
3. Continue until 51 such characters have been decrypted, yielding the session key, K'.
4. Remove the decrypted substring from C, leaving C'.
5. Decrypt C' with Handycipher and K', yielding M.

8. Cryptanalytic vulnerability

The way that the random choices are made in Steps 1 and 3 of the core-cipher encryption algorithm, and also in the null character insertion process of the Handycipher encryption algorithm, will have a significant effect on the cipher's vulnerability to statistically based attacks. In Step 1, the choices of R_1–R_5, C_1–C_5, and D_1–D_{10} should all be

---

[6] E.g., the Section 2 key `QjufGCtwbUSNLqHAgVDOoansIhyBKJWFdxvPk^peXMTlirYRmcE` is transcribed as `-Q.JUF-GC.TWB-USNL.Q-HA.G-VDO.OANS-I.HY-BKJWF.DXV-P.K^PE-XMT.LIR-YR.MC-E`

[7] This is necessary so that in Step 3 of the decryption algorithm the end of K'' can be recognized.

[8] Here $\lfloor x \rfloor$ denotes the integer part of x and |C| denotes the length of C. The formula is designed merely to make the value of j depend on K (and its sub-key $\bar{K}$) and |C|. The adjustments in Step 5 ensure that $j \leq |C|$. (See Appendix 2.)

equally probable, and in Step 3, each permutation of the string σ(m) should be equally probable. This can be accomplished with the use of a single six-sided die (as described, for example, by Reinhold [4]) or one can improve one's skill at behaving randomly by visiting Chris Wetzel's website [8]. For very short messages, it might be sufficient for these choices merely to be made nondeterministically, but as message length increases any departure from choosing randomly is likely to compromise the cipher's security against statistically based attacks.

The purpose of randomly inserting noise and null characters into the ciphertext is to defeat potential hill-climbing attacks against the otherwise undisguised ciphertext which would be vulnerable without that insertion. Although it might be sufficient for shorter messages merely to insert null characters nondeterministically, as the message length increases it becomes more important that they be inserted in the statistically balanced way described in the encryption algorithm to avoid their being detectable by statistical analysis.

It seems reasonable that any successful attack method will involve identifying the set of null characters for a given key, which is likely to be difficult since this set can be any of the approximately 1.3×10^{14} 25-character subsets of a 50-character ciphertext alphabet. Moreover, even if the set of nulls were to be discovered this would give no information about where the divisions occur in the ciphertext, and when removed the remaining ciphertext still remains quite secure through the insertion of noise characters and because of the homophonic nature of the cipher, using a great many cipher tokens. With any key, of the 31 characters comprising the plaintext alphabet $\mathcal{A}$: five are mapped into one of 5 length-1 ciphertext unigrams, ten are mapped into one of $20 \times 2! = 40$ length-2 ciphertext bigrams, ten are mapped into one of $20 \times 3! = 120$ length-3 ciphertext trigrams, five are mapped into one of $20 \times 4! = 480$ length-4 ciphertext 4-grams, and one is mapped into one of $20 \times 5! = 2400$ length-5 ciphertext 5-grams; thus there are a total of 3,045 cipher tokens available for use in the cipher's homophonic substitution before (and an unlimited number after) the insertion of noise or null characters.

The alternating reversal of coding direction might as well only be necessary for longer messages; on the other hand, it could be strengthened by building into the key an indication of which one of some other arbitrary patterns of alternation is to be followed—for example, the choice of null character used to mark the end of the embedded session key in Extended Handycipher could be so used. However, the most secure way of encrypting a very long message would seem to be to divide it into shorter ones and encrypt each using Extended Handycipher so that none of the randomly generated session keys will be exposed by encrypting a very long plaintext.

With respect to known-plaintext and chosen-plaintext resistance, the homophonic nature of the cipher together with the fact that each token is composed of a variable length string of symbols, is a very strong counter to such attacks. In effect an attacker must try all possible symbol lengths to try to synchronize with the text he knows. Also, the use of session keys would further limit the benefits of chosen or known-plaintext as such text only betrays itself. Similarly, the risk of the same message being encrypted twice with different keys is reduced.

The cipher would clearly be vulnerable to a chosen text of long repetitions of characters (e.g., the five singletons would ultimately reveal the five rows of the session-key-matrix) but it seems unlikely a hand cipher user would be trapped in this way. However it does imply that Handycipher would be a poor choice to implement in a micro controller with a fixed key.

9. Challenge cryptograms

Two plaintext messages M1 and M2 have each been encrypted with Extended Handycipher using the same key K, yielding the two cryptograms C1 and C2 contained in Appendix 3, not necessarily in that order. The first 229 characters of M1 consist of the Williams quotation in Section 3. The combined length of the two messages is between 1700 and 1800 characters.

The Handycipher variation described in Section 5 has been used, whereby each message was divided into two parts, and a randomly chosen complementary pair of session keys was used to core-encrypt the two parts, producing two ciphertexts which were then randomly interleaved to yield an intermediate cryptogram C′ for each message.

Instead of just one 51-character random session key K′, the 102-character concatenation of the random complementary pair of session keys used to core-encrypt the two parts of each message was transcribed, encrypted, and inserted into C′ in the Extended Handycipher encryption.

Four challenges in increasing order of difficulty are offered:

1. Determine whether C_1 is the encryption of M_1 or of M_2.
2. Reveal the plaintext following the first 229 characters of M_1.
3. Reveal M_2.
4. Reveal K.

10. Implementation notes

10.1. Although the process is tedious, with a bit of practice one can reasonably expect to encrypt or decrypt messages with Handycipher at a rate of approximately three plaintext characters per minute. At that rate the 229 character Williams quotation takes about an hour and a quarter to encrypt and perhaps an additional 30 minutes to generate, encrypt, and insert a session key.

10.2. In pen-and-paper work with strings of upper and lowercase letters the author has found it convenient to adopt a pronunciation convention as suggested by this example: the 10-tuple AbcDEFghiJ is read as

“cap-a *pause* bc caps-def *pause* ghi cap-j.”

It’s also helpful to read “w” as “dub,” “.” as “dot,” and “?” as “query.”

10.3. In order to facilitate visualizing the extended diagonals it’s helpful to think of the key-matrix as a 5 x 5 chess board (where the rows, columns, and diagonals wrap around the edges) and recognize that for any given square, 16 of the remaining 24 squares are colinear while just 8—those that are a knight’s move away—are non-colinear.

10.4. Although there is little propagation of errors in both encrypting and decrypting (except for possibly disturbing synchrony as discussed in 10.5 below) special care should be taken when processing the session key K′ since any error introduced into a key obviously will be propagated.

10.5. If an error is made in keeping track of the alternating direction of encrypting plaintext characters or decrypting groups of ciphertext characters (or if some other error causes such a disruption), the receiver will immediately notice what has happened and can adjust to keep in synchrony. (It might even be a useful ploy to do this intensionally several times to thwart some types of attack.)

10.6. Noise and null characters should certainly be introduced in encrypting the session key so that its length is less predictable, and the encryption of the 51st session key character must contain only a single null character at its end to ensure that its boundary is demarcated in the decryption process.

10.7. The frequency distribution of ciphertext characters can be further randomized by modifying K′, in Step 1 of the Extended Handycipher encryption algorithm, so that the most common plaintext letters will not be encrypted by K′ into unigrams, 4-grams, or 5-grams.

10.8. Similarly, to avoid having to insert many hyphens into the plaintext, K′ can be modified so that no very common plaintext bigram is among the five that cannot be encrypted by K′.

10.9. Before proceeding to Step 3 of the Extended Handycipher encryption algorithm, K′ should also be modified, if necessary, so that the transcribed K′ generated in Step 2 contains none of the five plaintext bigrams that cannot be encrypted by K.

10.10. Source code for a full Python implementation of Handycipher and Extended Handycipher, as well as several additional Handycipher-based challenges, are available at: https://www.mysterytwisterc3.org/ (MTC3).

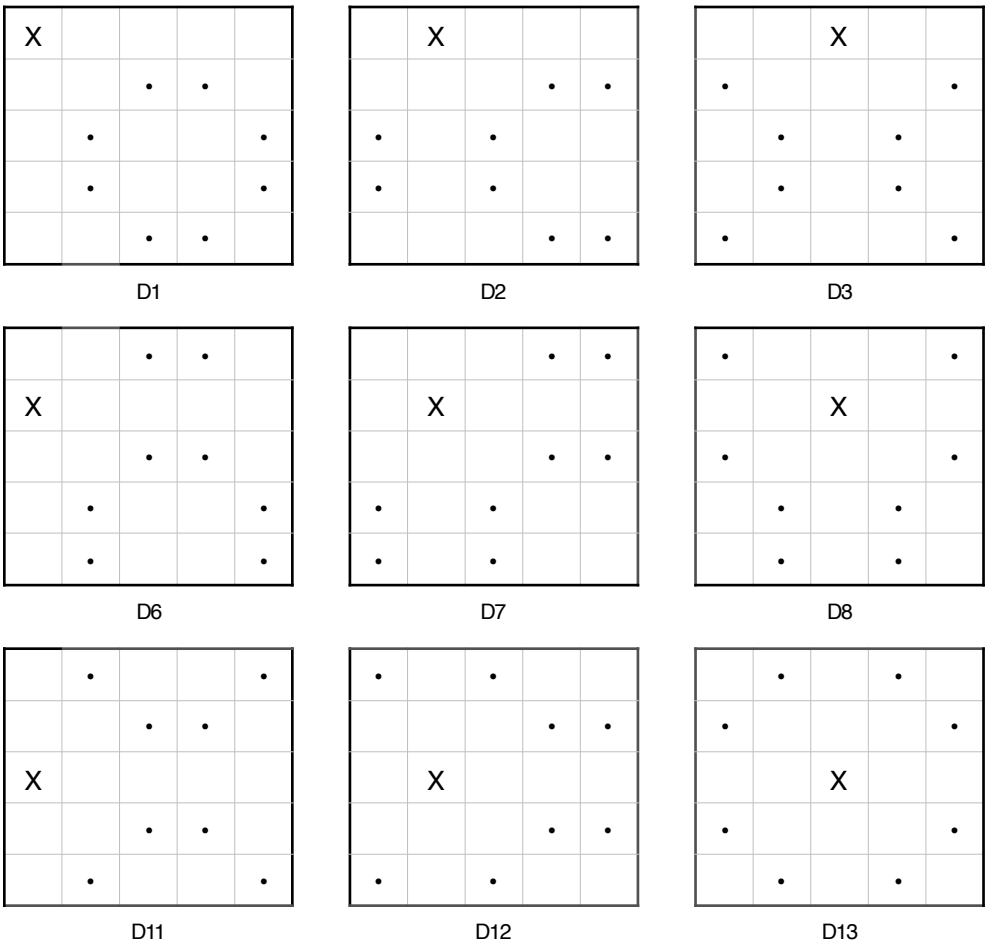
References

1. R. Canetti, C. Dwork, M. Naor, R, Ostrovsky, Deniable Encryption, in *Advances in Cryptology: Proceedings of Crypto 97* (1997), available at http://link.springer.com/chapter/10.1007/BFb0052229#..
2. A. Dhavare, R. M. Low, M. Stamp, Efficient cryptanalysis of homophonic substitution ciphers, Cryptologia 37 (2013) 250-281.
3. R. Merkle, M. Hellman, On the security of multiple encryption, CACM 24 (1981) 465-467.
4. A. G. Reinhold, How do I use dice to create random character strings?(1995), available at http://world.std.com/%7Ereinhold/dicewarefaq.html#randomstrings.
5. R. L. Rivest, A. T. Sherman, Randomized encryption techniques, in Advances in Cryptology: Proceedings of Crypto 82 (1982).
6. B. Schneier, The Solitaire encryption algorithm (1999), available at https://www.schneier.com/solitaire.html.
7. C. E. Shannon, Communication theory of secrecy systems. Bell System Technical Journal 28 (1949) 659-715, available at http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf.
8. C. Wetzel, Can you behave randomly? (1999) available at http://faculty.rhodes.edu/wetzel/random/intro.html.

Appendix 1

In processing the nth character m_n of a plaintext message $m_1 \ldots m_{n-1} m_n m_{n+1} \ldots m_N$ some combination of choices made in Steps 1 and 3 of the core encryption algorithm will generate a $\sigma(m_n)$ satisfying all the restrictions of Steps 1 and 3, provided that neither $\xi_{\bar{K}}(m_{n-1}) \times \xi_{\bar{K}}(m_n)$ nor $\xi_{\bar{K}}(m_n) \times \xi_{\bar{K}}(m_{n+1})$ equal 16.

It is fairly straightforward, although somewhat tedious, to show this by considering the distribution of colinear and non-colinear characters with respect to any given character in the 5 x 5 matrix M_K. For any such character the remaining 24 characters comprise 16 colinear and 8 non-colinear characters which can be diagramed as follows, where the symbol • indicates the position of a character non-colinear with the character located at X (see implementation note 10.3):



The other 16 possible locations for X result in six diagrams (D4, D5, D9, D10, D14, and D15) horizontally symmetric to six of these nine (D1, D2, D6, D7, D11, and D12), six diagrams (D16, D17, D18, D21, D22, and D23) vertically symmetric to six of these nine (D1, D2, D3, D6, D7, and D0), and four diagrams (D19, D20, D24, and D25) centrally symmetric to four of these nine (D1, D2, D6, and D7).

We prove the assertion inductively by describing an iterative process which chooses a $\sigma(m_n)$ satisfying all restrictions and also "looks ahead" eliminating the choice of any row which would make it impossible to encrypt m_{n+1} when $\xi_{\bar{K}}(m_{n+1})$ is a power of 2.

Initially, for n = 1, any of the 20 lines of M_K can be chosen to encrypt m_1, unless $\xi_{\bar{K}}(m_2) = 2^k$ for some k, in which case row R_{k+1} is eliminated, or unless $\xi_{\bar{K}}(m_1) = 2^j$ for some j, in which case $\sigma(m_1)$ must be chosen as one of the five characters in row R_{5-j} and clearly this is always possible unless k+1 = 5-j or j+k = 4, i.e., $\xi_{\bar{K}}(m_1) \times \xi_{\bar{K}}(m_2) = 16$.

In general, for n > 1, taking X as the location of the last character of $\sigma(m_{n-1})$ in each of the nine diagrams, it can be seen by inspection that for any of the 31 possible values of $\xi_{\bar{K}}(m_n)$ some permutation of the required characters in some line can be chosen to encrypt m_n satisfying all restrictions.

For example, consider the case in which location X is as in diagram D7. Expressing $\xi_{\bar{K}}(m_n)$ in binary as abcde, and assuming n is even, in order to make the first character of $\sigma(m_n)$ non-colinear with the last character of $\sigma(m_{n-1})$:

I. if a = 1, one can identify these eight lines: R_1, R_3, C_1, C_3, D_2, D_4, D_7, or D_{10}

II. if b = 1, one can identify these eight lines: R_1, R_3, C_1, C_3, D_3, D_5, D_6, or D_9

III. if c = 1, one can identify these eight lines: R_4, R_5, C_4, C_5, D_2, D_4, D_6, or D_7

IV. if e = 1, one can identify these eight lines: R_4, R_5, C_4, C_5, D_4, D_5, D_9, or D_{10}

If two or more of I – IV are true then some permutation of the required characters in any of those identified lines (except any row which is eliminated by looking ahead at m_{n+1}) can begin with a non-colinear character and therefore satisfy all restrictions.

If just one of I – IV is true then $\xi_{\bar{K}}(m_n) = 16, 8, 4$, or 1 and the required character in either of the corresponding identified columns is non-colinear and can be chosen for $\sigma(m_n)$ satisfying all restrictions.

If none of I – IV is true then $\xi_{\bar{K}}(m_n) = 2$ and, by induction, it can be assumed that R2 was not used to choose $\sigma(m_{n-1})$ and so any character in R2 other than the one at location X can be chosen for $\sigma(m_n)$ satisfying all restrictions.

Appendix 2

Given that $|C'| + |K''| \geq 700$ and also $N \geq 30 - R$ where $|C'| = 31 \cdot N + R$, $0 \leq R < 31$ we show that $j \leq |C'|$.

$|K'| \leq 101$ after transcription, therefore it may be safely assumed that $|K''| \leq 700$, and so $|C'| + |K''| - 700 \leq |C'|$.

Therefore

$$j = \lfloor (|C'| + |K''| - 700) / 31 \rfloor \cdot \{[\xi_{\bar{K}}(A) + \xi_{\bar{K}}(B) + \xi_{\bar{K}}(C)] \bmod 31\} + [\xi_{\bar{K}}(D) + \xi_{\bar{K}}(E) + \xi_{\bar{K}}(F)] \bmod 31$$

$$\leq \lfloor |C'| / 31 \rfloor \cdot 30 + 30 = N \cdot 30 + 30$$

$$\leq N \cdot 30 + N + R = N \cdot 31 + R = |C'|$$

Appendix 3

C_1

```
bxSvO HVLlL acTns tmjRC tYlHy JBymE VKwOQ YHYUc skodn laofP VeQib AFUaw WNKpf
olsYr RIiXO vfRAE nDUIy VMGxR imBuX TkKOA fnHlV OnAPJ eoNdQ tMaBr CJvEm HyJSx
hqfKt DlOGw FCXLy MQVAO sPEne lqXvf KOiNc XTGKm nrIBd AmuUm PIRPe KgFrD FvnIy
GAVvX FGigH oWXCB qDePg qYBmH BgJTp UfayN wmYDd nTrEA NIxmU VaEcv sxBLS WpcLi
HuSrS UVgKj TaqoE RVBwY LMqnm QUSnY gedlR tEJMk GSDog nhqFw ICbUb HBnCd WYDpf
ILbTO yHSEB LreIm IMkUg WLRAs THnjU VDbFG qQnVx JWebR DCEBd nBbft DMVdt WQODw
ofLke rcYVH UAyjn QqOeT LFfog CNvje DwETa mneFV STUGo taRJS kAxOU jbMgh KkRGC
IVEVo NvNru VYTDI FfNrg RlbHw qheBK mcwLR WMfvK OXoeP GKRIP NqxVW eivuD RjwRA
DHJxW TrLyM gFPof hDvwn Iikmw Clayf mVagD JtEpM KTAgR olmer AGByv NewBj GmHgL
FNEub SCdRT UMxjv XNDdT qAlhk jgRAo LckgH JUjLM yorKv ERkAh pixDe NEvPA oYabi
SqQpT fLevB jksCH SKXDr dvCDk NeXFC HBsuT vhjVU yeILb JAUtg jICdK QIirL sMJCe
dFgIQ mCDvc qdtbK QRVrD YdSxg CEJRh GEwQN Sexkg hMVGT xYeNv QiXGn TjCvO YUnjO
FdCYf PIxFK ALGBI mlvyw HAjLO beTgh lIhGf ysPHe jdbaB wCQTU WBuEG HAyIE gUbwO
YyIFi rNDMd sFMVK NGkvJ Pbcxe RJdmC oTPtR vjedc VfxWk lctRo MAItc EoDSF fVehA
NgVDo sOYyT AYRrt qGtCu UahBP JEqga WOVRE IAvkS NrNDQ TJHWd oDvoB wFDPc VemOD
WgGqK MmEVL OPvBN tbdxJ aXeLi UFuKa gULsw YDOgK NiWvU emDIF hOBdS DQcgn NuAnI
vcoRj PvbNE vXjTO wFCdj KQsIw KYxpJ FLTBt qlrRh jmAwC bgVYX UNCxt dUGDb EvSCG
cfMHe uPejv YEoHI oyDKv jHaeN WATwY AaHyq VWYbS JIEHC yUmFB cuOHK mjRDJ loDMb
dbPxO tMAHY DdrFL kmFgf lOFqa xFRWO jXNTv eNhmr yiGtH TRqOi GybNC dsnAm bIlgG
LvCJb gQRir eqnAM oSvBm DKXsI KuHgy tFCLY rhDTd ryQaQ RPHOT oDBRk hjAsj pgGAD
rEmWq HidDX RAxIu Jgpih xaEnF qsekf PlsXG OtFps dYgAt gXDOA FJtmG fhBbE VdrPJ
hSToj HxiNe uyINH KDJcS THSWd xJfQI QtwbV gwGcM Strqk fENFK nyAau wJtXE rHyqa
eCRqe ocuLd DmcsQ jhwWL QnhDX SIYif gLAbs dPVWD lMXtu NyqdV Qxlvh anbyL WDOwj
nFdfy CjKYQ XdiBm jTKjt IEGnm QqDKB jOemy Wlsux SARge IQPKp fNGVd ofPeE NFIlr
LJtoq rbkhY fWBLx oICdt lhMvR jbIQI vnmPT XvQhI usdBS APICD kbqwL TrBNh DnWQT
UMegN lxHcv uvxTO LNbYX RlYtT sSDNU IumJc UCVkE hrfRF jAXxS ajgJm VGkDI oyYjC
vYbFD yAajt UgRaf FAhjt WOaWJ NsliB eSMms XNFXj nstiY ycjTO KxFvM LCcbW BxelN
LhbxD paCyo vlInH VBmho kMbrj BgnpT nLdFf Rkvcf roNCe VLehF xOSJe JkscA WKwcM
ykmcC YXxgN IoSeq DbCfi dqkCG FIVCO YmSUs JuYkl Jxqec jtKBc sWKVx eMulH tsjIf
pMKwF UQGyc UjOxE DFuQf FbOlG vpatg LFQkO GgFsj GUvQO gplFJ usOUt jvelY VOlmQ
UapfJ YFfrl stYaQ UGpfb VfQjJ ULVvs lUgJF kQfLm UgFtG glQfJ QAler FfkFL uvQJU
kGYGv OUpep jagGs aYpFr Afpbt LaJgj Fvgem sfQlA sfeJu msUsQ lLuaJ FtOae mOYkA
tLgVL jtAmg vletF YfraV FOmeF Ykvsl pFver UAUGL mjfJp aGeVm saQkL OFlrV fbpYu
Ugvbg rFleb rYrtg fbJFl pYUjt QLrJp kjvlY GAQVm AjJUL tbavp QeUFY mFOLt ALmLV
fgYbU stLtA fAeug UAFeG LQFGt pgkrJ mFsbu kpOGs atLAQ vYGsg vfbLO lFGUm kgGeA
FeUbg QsUpg LrGUb emsuJ GrOvl VaUQu gFbrf vOvjr lGOra mYgbp mJsml uvUmG OmQVg
UbUre pkUjv gmjeU bLOJl Grbvj lYJFp VluUg pseTh rwSFQ LoROS bfMlh YykjM ElvXP
IKkfw LjRWG TSuRQ iuade bvCLT pymoE ICbnq MekWL CbrEA ntTGn yKquW Qfexb piVHj
PRWTD ONnjK nQGAD Rqisl iWKbl VHyhn Bpaot QCmak RfYiT hycrx EmQUe BARbh Tfvea
NMsrp ivECn NBVqI eFlgj wOFIQ csELU OCLbp KyFOv HTDPC jADpd lwoDT rAMJI UgpOk
DbUbi aRAtO wGhDF LASdT FYXxK FHatV hPpbo nKlxN mvwRu tAMeE ipVDW wxYua YKJtr
iOUcR PgSdD aQWlf sOUxy TjLoc LvALI SHXia hVgNB PMwaU aTuEd gVrhm fGRgq FHaLh
dahnP yRemt Ajubn ORoGB tNwJT yptAU jlxbg ednWh DWaXW hUFcC YHNtT JWpiL MBPEf
oYSVT FBmrp JvDuS UOETE mHMRB Ppdug vxSNJ Dqrdb PxKIm CDGbE WJuMR ilyTp mGBOc
vQnHo bwhsN OAuDT bayTA XYRqD cWdu
(2889 characters)
```

C_2

```
Ulphk NrYoO tbAew sXPvu rNtiC lcgWx eRDJU TwCjV BcnWK OXoJI AMaLu yEcXu UlRwQ
DJied KorFX lGeDA KuOjg CpdBw SEICN gtVck YFvOH ueYUw dbPWk wlIyA oTNlU FhNHr
uNcUv YtXru UniIh PfvRW LjNAh cgCTa yIphy lSFkq vrOXJ YDxnk bJMkv YGuPy aplqx
rWvwi RLHVK ynNjJ TViIk dFqaN IJHeP GfDKi uOhcq lrxdu NfiWE MUmRA lNqGe nlujF
rsMlp fLUWw xEFGA LeVuF RsgSY XoCmH GgUIN SKPeY iAoUO gueGX jYEbR vfSoW ohJFX
VoQKo NCSxO YlMOy jxWjQ ajPXr emAiq wntOL MVSNB wtHQd ukeDI pOVtA almVT IQVbt
```

```
UqyDF xQShe GOisP XhaKp UYUGq YlAFi tVmTj uPhyM vCuBy xavhN QGFYj TOAac bXBMU
nSPwW iLxKh BdPRO tawin gcEwO dXPfa yRVUM avUKR YBtAq HVBXc eiSFD MgPCu pxfMJ
OPNeX rwYKB IfnWa cqMOe YPqnt DUEiR OBoUW eCdCL DTGQB coJpm YCIFp LiWXS MESQj
XhlHu SXWvH nFwFT ACfQw Bjaub NXwjy aQGEx CYpGd hUNtM gHfaR rMSYL dFxyc Jeqfp
rFohr XDvHG xEpRl wqHXM kPWct rxeuR lOSmi AXLxu FMDGE wcQLq xGjyU VcSFW wMwth
aEncD bJVst TQvYD IlwNb StNrk bhSKX BHcjv FNpbJ qdPMC FhmdC yeFls vTQrf AoHdW
UOEgI GcpoY obhyX RqxeD xGrXI yNMVX jnbeX lrEgh MIoYa nQulG JkTvs KReUP fnYGU
MrfUs PHGFY dcKhJ AsXvB yGQPH KxoVa XpIwC FHeOY NkPtE WyhwD FOJTQ wqnPu GBMbE
CjhnD lGAeF QlArN xKlOx uXQUY BiPyR IUMLw WygqA slKDY giKuV RYSyX bktVW Cmyub
XJpxX HQTmo tEFaw CRVxC QwBXv btFyu YcjMU PWfUD KhIet AoykY UcYkx toOIr dgEAO
CkYTe UPWxp vlwFX ubfap ntxqD ueriT kUlGx CIhNO KYIJV atsvr ViXgh QXNQR vixGw
yVNBl hEMhA lKnBt VeQjP ldnhs ufNOg JfHPl hkCJM AVmpO vEXEk ICnet YgBqS auyma
UXJgG wCnAN UQdXk UWtyD uVovO iQdyQ HhCdB AscGh wALFl PilJx peILc GMwXR WfWQx
UFaHn YtdhA CeNXg UhrJa kUCxu tmTGL HXJIg pGjBu SdnVR hkmqg bcaNt RAXFw lopXI
xCWUr swqne AOvbg lnwBO FJHCG MPSyu aomKv yxVAM uCGnP OhaUQ bwxYS nqktl vuNXT
OFCtd xrePT FKUeM VOdoe xNAyV eHoXJ YpVQB EGMuN hdHAI EhMcO lhVQi ITKdP hBTxb
voerx jGwQy cYGhk MJfyG UCGjB HcmwD XRqcC OoKWU nixBd LTvYN OAIwu fMILG YcjUg
xXQDr OluKt PnwaU dMjsl asMCy uvEXR MDhyA nqlLw icejC fyGkg chdeq nyYlF wIGJH
RFwIm xWktN bYyji IbEVd JaxHN BRktY wenGr vBUYW gKOtM RVcuG unTUY AiGxu cMotF
bPFah XLHWv xsrHc VToei rQimN FMGwo JejxY OIwVA XFuHc Pbsrg GdjiO CgnYD oredI
ihjJD dQDCd bjluD UonSm uOjnv giUSD BHFRN CrIde AKBSe dONxV kpXxl JuRnO fvTcl
UXQxW HGETX FJjLg NnkDf tBIHg Rkids wnWXu iaTfV vuinS wtMqX ibdiv iuHIo OVtar
KkRVU eTBEL XIimC WQDtv oBqEm WDikt hnMUA PlAGV TNxht KCWIG AqQlA vienr miAkJ
LitVj EOrAQ tmnBg UwFIM oAtIU JCVxk UBkGd huFex cSiyR YqQyG mTniP dvNUS mqWiX
hIaid vuRAU OySrg PtXnu RsHDT vglBK OSicV ICLWP vmMAu hqFRX PhiIT YPOQd MHgae
cUGYP TvnlJ ADqeE XDWUP oxKBa Sndrx YyoBX RMqal Idwum Trlgm CMDhC RKFOP uUkHG
YoGbX MiDcO qakLo AcfSe KOVyS jEpIO cYkeP RBiOu UvGck WEXAt sIdDV PbtqY tHXBE
eAvqW LnxTE uPjtw qnsAG woFKB aevdT LTJEa FeKuB XvSdB tIjCQ GARyc XRVJw GESFH
FnfOJ iDRfA LxHYb pyvuO PGUqo Chxyb mPXaW SgUeo HOWtQ fqPch CKYMI hLOPU HYFki
AcedB xSQGe nFJYP bicXb tOCuR kWtRe LBrvm QMLDi yVrQY MxMPt IGlKM AkeHJ RwiLq
wlpna wkPVM lHLSY wPanf iUXuJ bloMb wspHw WPxlu hbYjL OAlfo WjHpI JxXFV odPbp
BPvou WGmEn tnNay CxHqK NIDAk DcxQa MXUPx BxnjD OgpEL jDdvE wAieq TVRCm xXtFP
HLIHS WYwks EpVDP rnhQY EABWc kaKvV UypIm cfVUW nKCAX DPhmy FKsHa LqxYm IWSDx
tEkHN lTrLr fovKU VYOEd yPIwl WiLdC tINDL pxGoJ hQnHF TPaUw WjIEH PThif FKsvo
Wdist XNkxT Hbeki Ycwqd TsWXQ hDrjv BnJEm Avwlo mgPfU HGhTv DQNbt rqJtO tvihI
wsWgD sNikJ oMXhA NxHhB YduOx KpRTD LIHoa JmXhp txPDk GEdhQ dNLDt MlXdy FwElj
YvGdn sWASk tLwmc BwFSf TJloi elHQO KtmYr yREGP xMVne SoJXv RNdEH lmFMF XAeGE
gXcdj Dnxuw LmBpV UhPnw WVijI Ctqdl rSpOc QYMNl bgsxo yTIiJ KBXWx ovehM Kxfse
rApBH neGkn YbhFu Qvgkm LYsAk rUsQO GJLUg rYuep rgQFb usaVe muFLb vOLfp LbkFY
lQVOl sgpua Fvrsu JQrke pmAFU fvbfp ULYJv QOFQJ vlasr LpQbj fFYsA upGgY gQVrt
jAteV UmGlJ OYkmA GptLA malFt rGaQl VtjAf gQvsa QGOlp OfGkA erYft kpbaj QUapF
rgtvg mUugv fgVAf JYpLJ AlAVY euvbG QUGfA pFrJu rgupt fFrGv YtaLg vrabF UluAp
bUYjQ mUmrU sUgvr bfFsg emjOj uLYrf gtFGY OGvgf VrajO lGgYF fjYlf mvVgA uQJUl
Ybfvm UluFf YGVem fVOme lFfuk vmFLp gAYrL gvstr LAtra bpfuJ sUQlV FQkfA rvUbp
QrtGp blapL bYusA rbJaA pgmJA LFGUY LaeGp fbrJg mAkjm vgjsQ vbQrg raVlU pkGUV
saQOJ QGmLp flOgG UAQUs ptsvl pabrm JfFYe YFQYL OlefU lvrer pJGFv Rmjna VvKCF
tirIj IEepy AXquR HdWSJ NFwlp tgHbA RYPfy MwfOk NtEpN nGciW SxoyH MXbPm wrkAt
TgUDy mVasN YDnAo XqMEI SJExd XkBpq OJHsP caKiL XJYrV MpBop QCfyF IxYEI TOnRM
JkySs auNXb gJGIm fdNqx SLvhd mTyAB fCaYE jhNAS LViQv BvhwE qrWKm RTvSu aCfYI
OuhiJ DxLBw VPqJp fasYP qSfcJ dYmDn jKAcg SyxBe Xrhjw IkKVR iMeaq mDhfy HnGuG
KTwna PnRYp jOSie CLswH JlkvR pmeXM cnCmg SCAui tUVQv HITos NfSyI qsjxd QpJaU
ykswA TReSa BJwIk DhnFi MVxEt mNsfH gNqYd qBlaS PRdrU FMxlu yVuMs ODGeu CIiHE
KplDO JXyLG UXneB aMNbl ShVtQ amtbg oOrJq uWsfA YpUjC VNuxd yqxfJ DUEoF PXsGk
IXTKW cpCKS NYXMU gQsro mVpaV jMJId Wkslb iBYNF jISUO pqgKA ExSXr MtVQe YiyEH
KNuKB q
(3906 characters)
```