December 17, 2011

Dean Wiley
Dunn, NC (USA)
http://www.mldragon.com

More Kryptos (K4) Thinking

The most (and most useful) Kryptos info: http://elonka.com/kryptos/
The official solution tester: http://kryptosclue.com/clue/clue.html
(Also stops the FAQ/Forum/Group/Buzz over "96 or 97 length of K4" – it says "…10 of the 97…")
My own K4 php decoder-ring: http://www.mldragon.com/k4.php
The nicely colored "Oh, jeez, it's like that!" page that got me decoding against a "correct" tableau (boxed, not offset) – from near the bottom of Elonka Dunin's kryptos page:
http://web.archive.org/web/20071116100808/http://filebox.vt.edu/users/batman/kryptos.html
("How to solve the KRYPTOS sculpture encryption" – Bill Houck)
(In the event that web.archive.org (or other "internet way back machines") vanish – I'm giving credit here for blatantly stealing Mr. (or Dr.) Houck's colored tables. In all of the Kryptos stuff I've seen, only his had a "correct" – if partial – encoding chart for K1/K2.)
Here's one through the magic of Copy/Paste from Internet Explorer to MSWord:

	K	**R**	**Y**	**P**	**T**	**O**	**S**	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z
1	**P**	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y
2	**A**	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S
3	**L**	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J
4	**I**	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H
5	**M**	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L
6	**P**	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y
7	**S**	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O
8	**E**	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D
9	**S**	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O
10	**T**	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P

There are many "important" things about the "Houck Chart(s)" but the "really important" part to me was that it visually illustrates how the K1/K2 "Key Alphabets" are pushed under the 1st letter of the "(Shifted/Caesar) Plain Text" alphabet.

This visual aid was absolutely necessary to correct the computer code because using normal (offset one left) table making my code (php) never produced a correct solution (always one letter wrong).

Because in the 6 months or so since Kryptos became a "way back burner" interest for me, I only had time to occasionally read the Yahoo Group messages and do some searching/reading. In my "very little" explorations – I have not found a complete "re-assembly" of the table (tableau) originally used to encode K1/K2. Houck's Charts only show the two known keyword portions.

Without further “to-do” here is a complete “Houck Chart” or “Sanborn Tableau” for K1/K2:

K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z
A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S
B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A
C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B
D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C
E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D
F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E
G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F
H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G
I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H
J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I
K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z
L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J
M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L
N	Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M
O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T
P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y
Q	U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N
R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K
S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P	T	O
T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R	Y	P
U	V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q
V	W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U
W	X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V
X	Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W
Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X	Z	K	R
Z	K	R	Y	P	T	O	S	A	B	C	D	E	F	G	H	I	J	L	M	N	Q	U	V	W	X

Now, having it all drawn out makes it easier to point out the important (and missed by me first time around) offset. The tableau (tabled out above) is 27 characters “high” (English alphabet plus Sanborn Key on top) by 26 characters “wide” (only enough for a standard English alphabet). The why of this importance is because the “99%-ers” (me included) usually would make a 26x26 table (or a 27x27). If you take the “starts with ABC and ends with KRYPTOS” line as the “top” (26x26) you cannot encode K1/K2 the way they were/are – you end up with a completely different cipher-text.

B+P (K1 1st characters) 26x26 = T (wrong)

B+P (K1 1st characters) 27x26 = E (correct)

Why this matters to K4

Well, it might not matter at all to K4. But "we" cannot eliminate the K1/K2 method without a proper trial (or assertion for you IT/programmer geeks).

Having the correct K1/K2 table/chart to work against allows us to "back out" the November, 2010, clue and see if we get something resembling a key.

You have to read the "reports" (NY Times and Wired) very lawyerly, essentially (how I interpret it):

- BERLIN in 64-69 boxes (where NYPVTT live now)
- It does not have to be a direct decode, it could be encoded and transposed (NY Times = "…the other 91 characters and their proper order are yet to be determined."

Using my new chart (above) here is the breakdown (cipher-character "found" in the top or left [they interchange]):

- N in B = E
- Y in E = L
- P in R = Y
- V in L = O
- T in I = I
- T in N = E

ELYOIE – been "worked to death" in the group and a couple of web pages. If it is a correct decode then it indicates: (1)a one-time-pad (non-repeating key); (2)multiple-pass-encryption*.

*= If K4 is multi-encrypted it is going to take a lot more brain and computer power to break it. For example, if "we" use the chart above and encode the plain text **ABC** against the key **ABC** we get **HJM**. Then, if we use **ABC** as key again (against **HJM**) we get **UXR**. So, knowing that **UXR=ABC** (only) does not help the potential decoder.

Using the Screen (The letters cut into the "right side" of Kryptos)

My original PHP code and by-hand scribbles were done against the tableau on Kryptos. It is very different from the Sanborn/Houck table I created above. See all the links above or the CIA's own breakdown at: https://www.cia.gov/about-cia/headquarters-tour/kryptos/index.html

There are extra letters (ABCD) in the top/bottom (they are the same) lines, there is an extra "L" in the first line of (what the CIA calls) "Panel 4" and the "key" alphabets (top, bottom and left-side) are all standard A-to-Z alphabets. All of these things make encoding/decoding using "the screen" very different than using the "…like the screen…" (-Sanborn) tableau (above) of K1/K2.

Using the screen with the "left-side key" (the transient L may indicate a "right-side key" – below):

N/B=S; Y/E=Y; P/R=M; V/L=L; T/I=W; T/N=R (**SYMLWR**)

(Does not look like a good key candidate.)

Using the screen with the "right-side key" (very complicated because no alignment and two "L" rows):

N/B=P; Y/E=V; P/R=F; V/L=H or I; T/I=T; T/N=P (**PVFH(I)TP**)

(Does not look like a good key candidate.)

Transposition ("Simple Transposition") Discussion

There is (still) argument to be made for the "mixed up letters" solution possibility for K4. In spite of the 4 "Z" characters, there are 6 "U" to match with the 4 "Q" (very few English [another assumption there] words contain Q without U). Because K4 is *only* 97 letters, using a form of "sift" is the easiest for trying to "un-mix" them. Trying to figure out "numerical" or "geometric" stacks (or matrices) is very difficult without a lot more letters to work with.

Here is K4 "straight-line" (Sorry, font reduced to fit the line):
OBKRUOXOGHULBSOLIFBBWFLRVQQPRNGKSSOTWTQSJQSSEKZZWATJKLUDIAWINFBNYPVTTMZFPKWGDKZXTJCDIGKUHUAUEKCAR

So, if I "pull out" BERLIN and all of the QU pairs I get:
OKOXOGHBSOLFBBWFLRVPRGKSSOTWTSJSSKZZWATJKLDIAWINFBNYPVTTMZFPKWGDKZXTJCDIGKHAUEKCAR

BERLIN
QU
QU
QU
QU

Next, I organize the remainder, because I like organized (numeric count tabbed out to make sure):

AAAA	4	
BBBB	4+1	
CC	2	
DDD	3	
E	1+1	
FFFF	4	
GGGG	4	
HH	2	
III	3+1	
JJJ	3	
KKKKKK KK	8	
LLL	3+1	
M	1	
NN	2+1	
OOOOO	5	
PPP	3	(Q=+4)
RRR	3+1	
SSSSSS	6	
TTTTTT	6	
UU	2+4	
VV	2	
WWWWW	5	
XX	2	
Y	1	
ZZZZ	4	(97)

BERLIN
QU
QU
QU
QU

Next, I take the "other side" of the argument and say, "That's more confusing than cryptography." Maybe someone out there more "anagram-tastic" than myself could assemble QUICK and BUZZBOMB and VX and SHOT into something. It does not seem like enough vowels to me, but I am very English language challenged. FYI: DOODLEBUG is another name for buzzbomb. *But*, I ask myself, *what about all those Ks and Ws*?

Machine Based *Destruction* (De-Construction)

"It *should* be easy!" -Every person who ever designed (thought up) a computer program and is **not** actually doing the programming.

The semi-programmatically-minded outline would look like **:

- Start with two A-Z arrays (characters)
- Crunch them against each other to find all the letter possibilities for BERLIN=NYPVTT (you do not have to use Sanborn's shifted KRYPTOSABC… alphabet because we are looking for a machine/math based "key"
- With a new (probably large) list of 6 character combinations "decode" the full 97 letters
- If "very lucky" the output results will have enough "almost" **human readable** letterings to pick "the best" of the new (probably large) list of 6 character combinations list as a "partial key" and just "brute-out" or "think-out" the rest of the key. **

**= I have not coded such a thing because:

- It takes time, of which I have none.
- It takes "spare" computer(s) or VMs, of which I have none.
- It relies on several assumptions (guesses): (1)2-key poly-coding; (2)repeating key; (3)**not** "true" poly-crypt-o (multi-pass poly-key example on page 3 of this *thing*); (4) original (plain-text) is English; etc. and so on, 5, 6, 7… unlimited assumptions and guesses that you cannot really write computer software to un-guess.
- Another "assumption" (that I do not hold to): BERLIN=NYPVTT – it is not what the article says so de-coding without transposition may be futile.
- The output of a single-character 2-key-poly de-crack (brute force against only the 6 letters looking for BERLIN) would be approximately 6.156119580207157310796674288400 2e+36 (according to Windows Calculator 26^26 (26 to the 26^{th} power)) lines of text to read, times 6 (for the 6 letters)… that would require a lot of spare humans, of which I have none.

Finally, for K4, for this year (2011) and probably next – unless "divine inspiration" happens:

Someone hurry up and solve K4 so I do not have to think about it ever again.