Gmail **Bill Ricker <bill.n1vux@gmail.com>**

Vicksburg Re: One of the hazards of the internet

1 message

Bill Ricker <bill.n1vux@gmail.com> Mon, Jul 13, 2020 at 6:07 PM
To: Stephen Ronan <sronan@gmail.com>
Cc: Kurt Keville <kurt.keville@gmail.com>, Brian DeLacey <bdelacey@gmail.com>

On Mon, Jul 13, 2020 at 3:12 PM Stephen Ronan <sronan@gmail.com> wrote:

> no I missed that one... had to cheat now to learn that reinforcements were not on the way...

Vicksburg fell the same time as Gettysburg ... July 4, 1863 ... arguably more important to Union eventually victory over Slaver Rebels than Gettysburg.
(OTOH as a proud son of Maine, I'm all for Chamberlain "Bayonets!" getting excess credit.)

Ok, here's what SHOULD be on the screen for that video ...

I demonstrated using Kasiski Examination or Friedman Incident of Coincidence to find the most propbably length of repeating key for a polyalphabetic cipher, using the language formerly known as Perl6 (now called "raku")

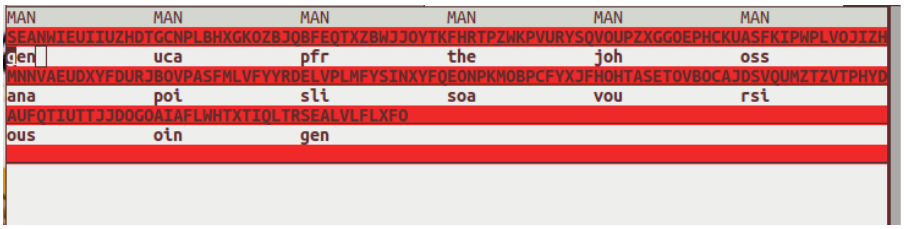
```
raku ./vicksburg.p6
(S| E| A| N| W| I| E| U| I| I| U| Z| H| D| T| G| C| N| P| L| B| H| X| G| K| O| Z| B| J| Q|
B| F| E| Q| T| X| Z| B| W| J| J| O| Y| T| K| F| H| R| T| P| Z| W| K| P| V| U| R| Y| S| Q|
V| O| U| P| Z| X| G| G| O| E| P| H| C| K| U| A| S| F| K| I| P| W| P| L| V| O| J| I| Z| H|
M| N| N| V| A| E| U| D| X| Y| ...)
15 => <273/103>=2.650485
14 => <286/207>=1.381643
7 => <130/107>=1.214953
4 => <260/217>=1.198157
16 => <234/205>=1.141463
10 => <234/211>=1.109005
12 => <208/209>=0.995215
8 => <208/213>=0.976526
3 => <104/109>=0.954128
2 => <208/219>=0.949772
11 => <13/15>=0.866667
9 => <91/106>=0.858491
5 => <91/108>=0.842593
1 => <91/110>=0.827273
13 => 0.5=0.5
6 => <104/215>=0.483721
15
```

Isn't that cute, it has real Rational numbers as a first class class.

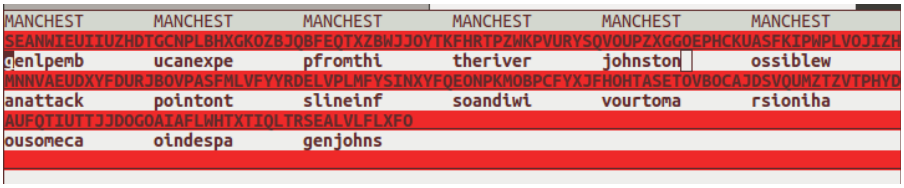
This cryptogram is too small for automated guessing of hot plaintext letters ETAONRISH for the hottest letters in each key position 0..14, and we don't yet know that the key repeats some key letters (bad crypto opsec, but then using the same key for 4 years was worse).

Second, using that assumed key repetition rate, we'll try the probable word "GEN" (Prefix or abbreviation of "General") to see what happens. Normally this would be trial and error, "dragging" the "crib" through to see if the assumption in a position gives a key letter than brings plausible natural language in other repetitions of the key letter. (Which is why we needed the repetition length first!)

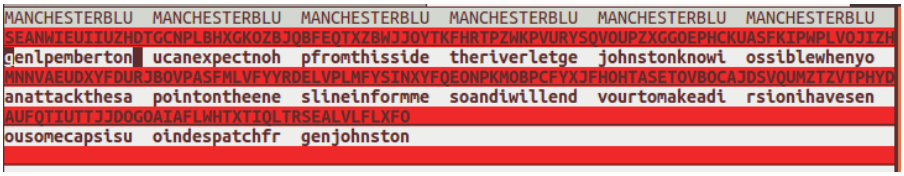
As luck would have it, if we guess GEN right at the start, we get quite a few english trigrams, including a second "GEN", a "THE" and a "JOH", which latter is the beginning of one of the 3 Rebel generals in the area according to our intel brief, and the other sequences look plausibly English.



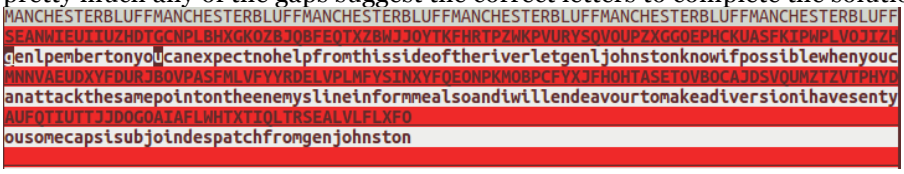
Filling out his name JOHNSTON gets us half done ...



at which point recipient's name is poking out at us: GENL PEMBERTON



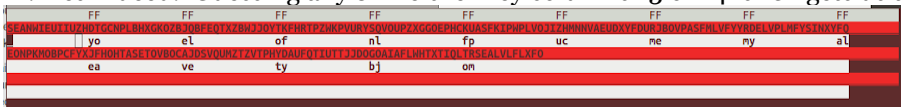
pretty much any of the gaps suggest the correct letters to complete the solution -



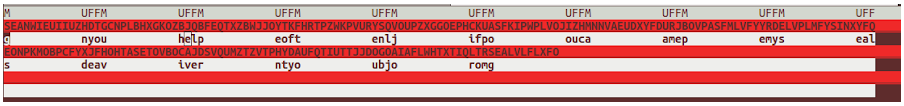
As we can see, leaving out word divisions only slows the unauthorized decryptor down slightly.

(I should have included screenshots like this in my notes file for JABR ...)

We *in theory* could have used a variation of Freidman Kappa or IC again to compare the mod 0..14 position subsequences to detect any common key letter, but wouldn't work very well for this short a message - columns 13 and 14 have the same key, but only score slightly more similar than columns 2 and 4. And even if was more obvious, or we took it as suggestive, the merged buckets wouldn't be big enough for "law of large numbers" to make them divulge much of ETAONRISH. The buckets that *should* merge have plaintext `aabceeeeffjlllmmmnoooptuvyyy` which is missing R,I,S,H and deficient in T, A, with excess of O, Y, M, so would really only point out e=>J and 2 candidates for EAON. Better than nothing, would probably get us in? Yes indeed. Guessing any **J** in either key columns 13 or 14 is "***e"*** gets us the following,



in which most glaring is an "**yo**" that we tentatively guess is ***you***,
and from that the "***el***" becomes an "***elp***" that we guess is "***help***", which assures us we're on the right track.



Once we see "***enlj"*** and remember our situational awareness brief, the generals on rebel side are Pemberton besieged (where message was going *to*), Walker on far bank (where message came *from*), and Johnston in overall command but not on scene, **GENL JOHNSTON** becomes the probable continuation and we're in, half done, same as above where we entered with GEN.

Had they used secret tactical codenames for the commanders and not used a key with reused letters, this would be a bit harder ... but crib-dragging the contextually probable words for a Mississippi River city-fortress siege works too: the first one of "RIVER" "ATTACK" "ENEMY" that we think of (or find in our thesaurus) will get in quick enough.

(And the President's Telegraphers had by now learned the key was MANCHESTER BLUFF anyway.)

This message wasn't cracked at the time, it wasn't even opened until recently. Another intercepted message at the same siege, direct from Johnston to Pemberton, was forwarded by telegraph to Washington and cracked. It used the same Key, and didn't even encipher every word, which is /worse/ OPSEC, and contained mistakes.

I have yet to determine what "MANCHESTER BLUFF" meant to the Rebels; their later key phrases were meaningful slogans (poor OPSEC). One suggestion is regards to trade with UK cotton mill town of same name, which engaged in trade via the port of Liverpool. (The cotton merchants there quickly pivoted to planting cotton in Egypt, so any bluff of Cotton merchants forcing Parliament to send the RN to break the blockade would indeed be a bluff. I am unconvinced.) OTOH I find no such placename either?

The above demo was an interactive text program, using "curses" library in Perl.

```
Usage: CursesSimp_1.pl [options]
       CursesSimp_1.pl -help
       CursesSimp_1.pl -version

Options:

 -keylen <keylen>       Presumed keylength is <keylen>. Required.
 -key <key>             Key initialized to <key>
                                or <key>x<keylen> if single char, both given
 -nobreaks              remove word breaks from cipher

 TBD to load key, cipher or plaintext by arg

 F4 to quit. HOME to Begining. PGUP to Key, PGDN to text.
  arrows to move.  INS to toggle Overstrike mode.
  letters on key or plaintext to guess. BackSpace or Space to erase.
```

(I need to add a TAB function to slide a trial word along as well as arguments to load ciphertext from commandline or file sometime, but not today.)