Unit 43
Attacking the Fuer GOD cipher with cribs

We have seen crib-based attacks before, but polyalphabetic ciphers add some complications. There are three-ish overlapping and interrelated tasks of such an attack, and each of them is affected by the nature of periodic polyalphabetic ciphers:

0. finding the period of the cipher
1. placement of the crib against the ciphertext
2. extracting information about the key from the crib and the matched segment of ciphertext
3. expanding the information that we have about the key and/or plaintext

Finding the period

We have studied this already, and recommend the Sinkov method.

Placement of the crib

We try placing the crib at every possible position against the ciphertext and checking for consistency. In other words, we check to see that for each key alphabet there is only one letter for each of the twenty-six slots and that no letter is duplicated. We can also check that a partial key matches any row of the cipher's tableau. For other ciphers there can be more or different constraints. There will often be several possible placements of the crib that are consistent and have to be tried. Ultimately, textual fitness will determine success.

Extracting information

This step overlaps the previous one. By comparing the crib to the corresponding segment of the ciphertext, we find some of the letters in the key table. Using those letters, we can also find the plaintext letters for parts of the ciphertext outside the segment we used.

Expanding the information

There are two ways to increase what we know about the keys and plaintext. One is to look for probable words in the text where some letters are known. If we are correct, then we can add more letters to the key table. The second way is to use the structure of the cipher to deduce more information about the keys from what we have. With the Fuer GOD cipher, since we know the tableau, once we know one or two letters in a key alphabet, then we can fill in the rest of that row in the key table.

To summarize: The placement of a crib can fail for any of these reasons:

- the key table gets two of the same letter in different positions on the same row
- we try to put two different letters into the same position in the same row of the key table
- a row of the key table is not consistent with any of the rows of the tableau
- the resulting decrypted plaintext does not resemble text in a human language

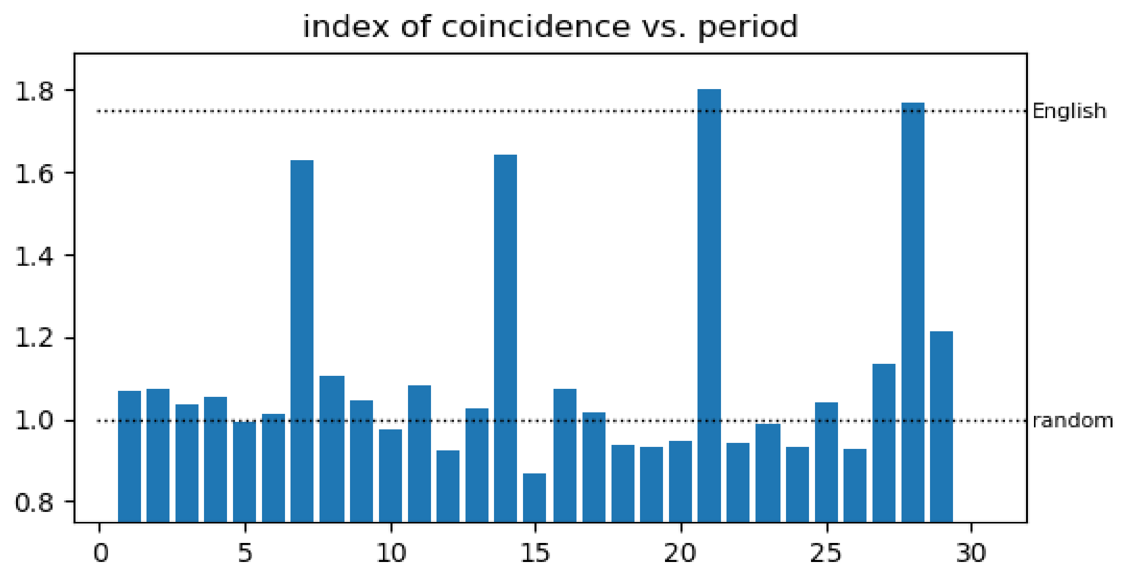
An example would be nice, eh? Consider this ciphertext:

```
JROQLDBCVZTQRSGMXQHHZGESKLOSOTLOHEIFZOFOYJLENIFTHQFFBPHOHFC
FDUSFDNKBGQLZBDIRHMKXTMDQSQYELDBHRYCHEFJRMTFCJCXRBUGTRCTHCV
BSQJWEUIGNJWOWQCUYIHGPJPMFLYQJPNBTMHMXXTBEOSQYPMDCXFNIFUOMX
SBMWGUIXMCHIGNXMCCHCVZQIVDRWUBMWKLKOIFUABDYQIVSHPVFBZAFVOQQ
UHPPGOBEBFFRLWCBOFUFDUSFDNKBGASPGOIVQJCLEIHVFFCLWHQJRI
```

Let’s see if we can break it with this crib:

```
IVYANDTHECAN
```

The zeroeth step is to find that the period is seven:



So we will build a key table with seven alphabets:

	plaintext alphabet ABCDEFGHIJKLMNOPQRSTUVWXYZ
0	....
1	....
2	....
3	....
4	....
5	....
6	....

Next, we must find the correct placement of the crib. We will show two positions that fail, and then move on to the positions that give consistent key tables, one of which fails in comparison to the tableau and one of which succeeds. Start at position zero, i.e., at the beginning of the ciphertext. Is this placement possible?

```
0123456012345601234560123450...
IVYANDTHECAN
JROQLDBCVZTQRSGMXQHHZGESKLO...
```

If we fill in the key table, we get to the following state. But at position 10, we need to put a `T` under the plaintext `A` in row 3, yet that spot is already occupied by `Q`. Therefore we must reject this placement.

	plaintext alphabet ABCDEFGHIJKLMNOPQRSTUVWXYZ
0	.......CJ.................
1	....V................R....
2	..Z.....................O.
3	Q.........................
4	.............L............
5	...D......................
6	...................B......

We get a similar inconsistency until we reach position 12:

```
...34560123456012345601234560 1...
   IVYANDTHECAN
...RSGMXQHHZGESKLOSOTLOHEIFZOF...
```

We fill in the table until the following state. But then at position 21 we want to put a `G` under the plaintext `C` in row 5. But there is already a `G` under `Y` in row 5. So we must reject this placement of the crib.

	plaintext alphabet ABCDEFGHIJKLMNOPQRSTUVWXYZ
0	.............X............
1	...Q......................
2	...................H......
3	.......HR.................
4	....Z................S....
5	........................G.
6	M.........................

At position 118 we find a placement that passes the tests of internal consistency of the key table.

```
...60123456012345601234560123 4...
```

```
...6012345601234560123456012 34...
   IVYANDTHECAN
...BSQJWEUIGNJWOWQCUYIHGPJPMFL...
```

	plaintext alphabet ABCDEFGHIJKLMNOPQRSTUVWXYZ
0	....G................S....
1	..N.....................Q.
2	J.........................
3	.............W............
4	...E......................
5	...................U......
6	.......IB.................

The problem now is that there is no row of the Fuer GOD tableau that is consistent with row 0 of our key table. If G comes under E, then S must be under T, P, or M in rows C, W, and Z of Table 41.2. But here it appears under V. So, once again, we reject this placement of the crib. We have better luck at position 181:

```
...60123456012345601234560 1...
   IVYANDTHECAN
...GUIXMCHIGNXMCCHCVZQIVDRW...
```

We get this partial key table:

	plaintext alphabet `ABCDEFGHIJKLMNOPQRSTUVWXYZ`
0	`....G................U....`
1	`..N.....................I.`
2	`X.........................`
3	`.............M............`
4	`...C......................`
5	`...................H......`
6	`.......IG.................`

Now comes the the time to expand our knowledge. This will be very easy, since we know the tableau of the Fuer GOD cipher. By comparing the zeroth row of the table above to the tableau, we see that it matches key letter `C`, the first row matches `R`, etc. Row 5 matches tableau rows `L`, `N`, and `P`, so we will have to do more work. Row 6 matches `A` and `E`. The others have a single match each. Suppose we decrypt with key `CRIBB..`; we get (after inserting the missing bits of the crib)

```
HESTO..ANDST..EDATI..ERYMO..NFULI..IMSEL..ORTHE..STPAR..FTH
EC..ISTMA..OMICH..LWASN..THEGO..FEEDI..HERSE..ALWAY..ROVID.
.THOUG..ECOUL..AKEHI..HAREO..HATAS..LLASA..THERN..UTTHE..LL
YAN.IVYANDTHECAND..ANDTH..RIBAN..OWSHE..DSETH..FACEA..INSTT
..MALLA..ITWOU..NTBEC..ISTMA..TALLH..HOUGH..ITHOU..HEM
```

We can see some probable words peeking through, such as `ALWAY.` which looks like `ALWAYS`. If we look at the ciphertext, we see that that `S` was enciphered to `G` in row 5. Now we have

	plaintext alphabet `ABCDEFGHIJKLMNOPQRSTUVWXYZ`
5	`..................GH......`

This breaks the ambiguity for row 5, since now it can only match row `L` of the tableau. With this improvement, the plaintext has fewer holes:

```
HESTOO.ANDSTA.EDATIT.ERYMOU.NFULIN.IMSELF.ORTHEB.STPART.FTH
ECH.ISTMAS.OMICHA.LWASNO.THEGOO.FEEDIN.HERSEL.ALWAYS.ROVIDE
.THOUGH.ECOULD.AKEHIS.HAREOF.HATASW.LLASAN.THERNO.UTTHEH.LL
YANDIVYANDTHECANDL.ANDTHE.RIBAND.OWSHEH.DSETHE.FACEAG.INSTT
H.MALLAN.ITWOUL.NTBECH.ISTMAS.TALLHE.HOUGHT.ITHOUT.HEM
```

It looks like `.ROVIDE` could be `PROVIDE`. That adds a letter to row 6:

	plaintext alphabet `ABCDEFGHIJKLMNOPQRSTUVWXYZ`
6	`.......IG......T..........`

This fixes the last letter of the key to be `E`, and the key is `CRIBBLE`. We can now completely decrypt the text, which comes from *Candle and Crib* by Katherine Frances Purdon:

> He stood and stared at it, very mournful in himself. For the best part of the Christmas to Michael was not the good feeding Herself always provided, though he could take his share of that, as well as another; no, but the holly and ivy and the Candle and the Crib; and now she had set her face against them all. And it wouldn't be Christmas at all, he thought, without them!

Programming tasks

1. Implement the attack. The function you wrote in Exercise 2 of Unit 37 should be useful in checking for consistency and building the partial key table.

Exercises

Break these ciphertexts. Plaintexts are in English. What are the keywords?

1. Crib: `UNMARRIEDGERMAN`

```
KXQQRMXXHXWXDNRZXQQCFXOHNQXXFRCWZIRNMKCPVLQLPEMKHMSYXLPVFXH
TRMXHMCHHVQCVEKNCIHVRXGXHNCIRHVMYGAVMAKZROJQBTCMFAWFCFYXAYX
JCEIEKNRJFBJFSYGXXMHDRWTHAXWUHSPXOENPESETAXSKLZRZKAFRGAQCFC
HCCPMGOTALDNWNRJSUDFWLAWHLBRAXXQZRHZ
```

2. Crib: `THREEOREVENFOUR`

```
AQWAYEXHFTAQBOTSFKIBCEAHHMEUTMKNCWRQMRZOXJQGJOTXCKDKYAKNCWR
HYWGIRARKNAMHOFCZNSYSQMVRBNGIGKPUJAJDCTRUTMKNCWWPCEXAFNDUJX
DAQEFYAGIQ
```