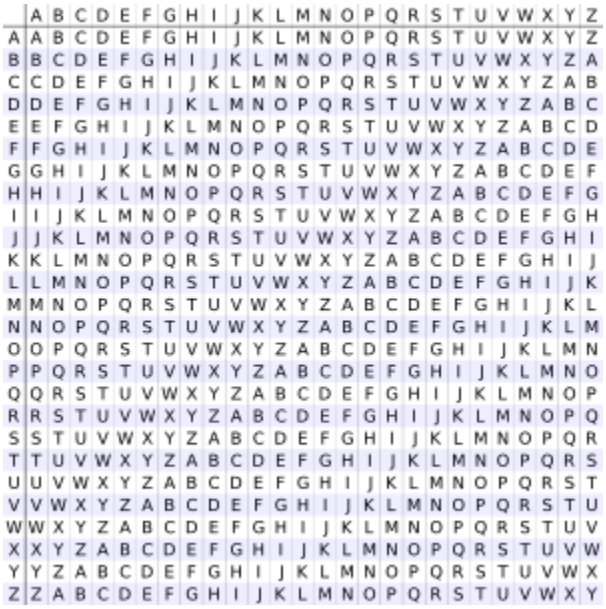
WIKIPEDIA

Autokey cipher

An **autokey cipher** (also known as the **autoclave cipher**) is a cipher that incorporates the message (the plaintext) into the key. The key is generated from the message in some automated fashion, sometimes by selecting certain letters from the text or, more commonly, by adding a short *primer key* to the front of the message.

There are two forms of autokey cipher: *key-autokey* and *text-autokey* ciphers. A key-autokey cipher uses previous members of the keystream to determine the next element in the keystream. A text-autokey uses the previous message text to determine the next element in the keystream.

In modern cryptography, self-synchronising stream ciphers are autokey ciphers.



A tabula recta for use with an autokey cipher

Contents

History

This cipher was invented in 1586 by Blaise de Vigenère with a reciprocal table of ten alphabets. Vigenère's version used an agreed-upon letter of the alphabet as a primer, making the key by writing down that letter and then the rest of the message.[1]

More popular autokeys use a tabula recta, a square with 26 copies of the alphabet, the first line starting with 'A', the

next line starting with 'B' etc. Instead of a single letter, a short agreed-on keyword is used, and the key is generated by writing down the primer and then the rest of the message, as in Vigenère's version. To encrypt a plaintext, the row with the first letter of the message and the column with the first letter of the key are located. The letter in which the row and the column cross is the ciphertext letter.

Method

The autokey cipher, as used by members of the American Cryptogram Association, starts with a relatively-short keyword, the *primer*, and appends the message to it. If, for example, the keyword is "QUEENLY" and the message is "ATTACK AT DAWN", the key would be "QUEENLYATTACKATDAWN".[2]

```
Plaintext:  ATTACK AT DAWN...
Key:        QUEENL YA TTACK AT DAWN....
Ciphertext: QNXEPV YT WTWP...
```

The ciphertext message would thus be "QNXEPVYTWTWP".

To decrypt the message, the recipient would start by writing down the agreed-on key again.

```
QUEENLY
```

The first letter of the key, Q, would then be taken, and that row would be found in a tabula recta. That column for the first letter of the ciphertext would be looked across, also Q in this case, and the letter to the top would be retrieved, A. Now, that letter would be added to the end of the key:

```
QUEENLYA
```

Then, since the next letter in the key is U and the next letter in the ciphertext is N, the U row is looked across to find the N to retrieve T:

```
QUEENLYAT
```

That continues until the entire key is reconstructed, when the primer can be removed from the start.

Cryptanalysis

Autokey ciphers are somewhat more secure than polyalphabetic ciphers that use fixed keys since the key does not repeat within a single message. Therefore, methods like the Kasiski examination or index of coincidence analysis will not work on the ciphertext, unlike for similar ciphers that use a single repeated key.[3]

A key weakness of the system, however, is that the plaintext is part of the key. That means that the key will likely contain common words at various points. The key can be attacked by using a dictionary of common words, bigrams, trigrams etc. and by attempting the decryption of the message by moving that word through the key until potentially-

readable text appears.

Consider an example message "MEET AT THE FOUNTAIN" encrypted with the primer keyword "KILT":[4] To start, the autokey would be constructed by placing the primer at the front of the message:

```
plaintext:  MEETATTHEFOUNTAIN
primer:     KILT
autokey:    KILTMEETATTHEFOUN
```

The message is then encrypted by using the key and the substitution alphabets, here a tabula recta:

```
plaintext:  MEETATTHEFOUNTAIN
key:        KILTMEETATTHEFOUN
ciphertext: WMPMMXXAEYHBRYOCA
```

The attacker receives only the ciphertext and can attack the text by selecting a word that is likely to appear in the plaintext. In this example, the attacker selects the word "THE" as a potential part of the original message and then attempts to decode it by placing THE at every possible location in the ciphertext:

```
ciphertext: WMP MMX XAE YHB RYO CA
key:        THE THE THE THE THE ..
plaintext:  DFL TFT ETA FAX YRK ..

ciphertext: W MPM MXX AEY HBR YOC A
key:        . THE THE THE THE THE .
plaintext:  . TII TQT HXU OUN FHY .

ciphertext: WM PMM XXA EYH BRY OCA
key:        .. THE THE THE THE THE
plaintext:  .. WFI EQW LRD IKU VVW
```

In each case, the resulting plaintext appears almost random because the key is not aligned for most of the ciphertext. However, examining the results can suggest locations of the key being properly aligned. In those cases, the resulting decrypted text is potentially part of a word. In this example, it is highly unlikely that "DFL" is part of the original plaintext and so it is highly unlikely either that the first three letters of the key are THE. Examining the results, a number of fragments that are possibly words can be seen and others can be eliminated. Then, the plaintext fragments can be sorted in their order of likelihood:

```
unlikely <------------------> promising
EQW DFL TFT ... ... ... ... ETA OUN FAX
```

A correct plaintext fragment is also going to appear in the key, shifted right by the length of the keyword. Similarly, the guessed key fragment ("THE") also appears in the plaintext shifted left. Thus, by guessing keyword lengths (probably between 3 and 12), more plaintext and key can be revealed.

Trying that with "OUN", possibly after wasting some time with the others, results in the following:

```
shift by 4:
```

```
ciphertext: WMPMMXXAEYHBRYOCA
key:        ......ETA.THE.OUN
plaintext:  ......THE.OUN.AIN

by 5:
ciphertext: WMPMMXXAEYHBRYOCA
key:        .....EQW..THE..OU
plaintext:  .....THE..OUN..OG

by 6:
ciphertext: WMPMMXXAEYHBRYOCA
key:        ....TQT...THE...O
plaintext:  ....THE...OUN...M
```

A shift of 4 can be seen to look good (both of the others have unlikely Qs) and so the revealed "ETA" can be shifted back by 4 into the plaintext:

```
ciphertext: WMPMMXXAEYHBRYOCA
key:        ..LTM.ETA.THE.OUN
plaintext:  ..ETA.THE.OUN.AIN
```

A lot can be worked with now. The keyword is probably 4 characters long ("..LT"), and some of the message is visible:

```
M.ETA.THE.OUN.AIN
```

Because the plaintext guesses have an effect on the key 4 characters to the left, feedback on correct and incorrect guesses is given. The gaps can quickly be filled in:

```
MEETATTHEFOUNTAIN
```

The ease of cryptanalysis is caused by the feedback from the relationship between plaintext and key. A three-character guess reveals six more characters, which then reveal further characters, creating a cascade effect. That allows incorrect guesses to be ruled out quickly.

In modern ciphers

Modern autokey ciphers use very different encryption methods, but they follow the same approach of using either key bytes or plaintext bytes to generate more key bytes. Most modern stream ciphers are based on pseudorandom number generators: the key is used to initialize the generator, and either key bytes or plaintext bytes are fed back into the generator to produce more bytes.

Some stream ciphers are said to be "self-synchronising" since the next key byte usually depends only on the previous *N* bytes of the message. If a byte in the message is lost or corrupted, then the key-stream will also be corrupted, but only until *N* bytes have been processed. At that point, the keystream goes back to normal, and the rest of the message will decrypt correctly.

See also

- Chaocipher
- Cipher Block Chaining

Notes

1. "Vigenère Cipher" (http://crypto.interactive-maths.com/vigenegravere-cipher.html). *Crypto Corner*. Retrieved 2018-08-13.
2. "Autokey Calculator" (https://web.archive.org/web/20131202225102/http://asecuritysite.com/security/Coding/autokey?word=attackatdawn%2Cqueenly). Asecuritysite.com. Archived from the original (http://asecuritysite.com/security/Coding/autokey?word=attackatdawn%2Cqueenly) on 2013-12-02. Retrieved 2012-12-26.
3. Hoffstein, Jeffrey; Pipher, Jill; Silverman, Joseph (2014). *An Introduction to Mathematical Cryptography* (https://books.google.ca/books?id=cbl_BAAAQBAJ). Springer. p. 288.
4. "Autokey Calculator" (https://web.archive.org/web/20131203104209/http://asecuritysite.com/security/Coding/autokey?word=meetatthefountain%2Ckilt). Asecuritysite.com. Archived from the original (http://asecuritysite.com/security/Coding/autokey?word=meetatthefountain%2Ckilt) on 2013-12-03. Retrieved 2012-12-26.

References

- Bellaso, Giovan Battista, *Il vero modo di scrivere in cifra con facilità, prestezza, et securezza di Misser Giovan Battista Bellaso, gentil'huomo bresciano*, Iacobo Britannico, Bressa 1564.
- Vigenère, Blaise de, *Traicté des chiffres ou secrètes manières d'escrire*, Abel l'Angelier, Paris 1586. ff. 46r-49v.
- LABRONICUS (Buonafalce, A), *Early Forms of the Porta Table*, "The Cryptogram", vol. LX n. 2, Wilbraham 1994.
- Buonafalce, Augusto, *Bellaso's Reciprocal Ciphers*, "Cryptologia" 30 (1):39-51, 2006.
- LABRONICUS (Buonafalce, A), *Vigenère and Autokey. An Update*, "The Cryptogram", vol. LXXIV n. 3, Plano 2008.

External links

- Secret Code Breaker (http://www.secretcodebreaker.com/autokey-cipher.html) - AutoKey Cipher Decoder and Encoder
- A Javascript implementation of the Autokey cipher (http://www.practicalcryptography.com/ciphers/autokey-cipher/#javascript-example)
