# One-Time Pad Encryption

Many simple encryption schemes are based upon letter shifting. The Caesar cipher, for example, shifts all letters of the alphabet by a fixed amount. A shift of 3 converts A → D, B → E, C → F, …, W → Z, X → A, Y → B, Z → C. (Note how the shift wraps around the alphabet at the end.) Shifts of this sort are the basis for the *decoder rings* once included as prizes in breakfast cereal boxes. The image at right shows a decoder ring set to a shift of 7. To encode a message, you substitute characters on the outer ring for the ones on the inner ring. Decoding, you do the opposite.


If we assign numeric values to the letters, with A = 0, B = 1, …, Z = 25, we can see that the Caesar cipher is a form of modular arithmetic. Shifting character $c$ by amount $s$ results in character $(c + s)$ **mod** 26.

To practice, can you decode the name of the important person below, encoded with a shift of 14?

**Gcdvwo Gawhv**

A Caesar cipher is easy to break because all letters use the same shift. More difficult schemes use shifts that change with each character. To encrypt or decrypt messages using a one-time pad, you manipulate individual letters according to a random sequence of shifts (called the pad), itself represented as a sequence of letters. Each element of the random sequence specifies the amount to shift each letter. For example:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Message: | D | o | g | | 3 | 14 | 6 | |
| | | | | As numbers: | + (mod 26) | | | |
| Pad: | Q | X | D | | 16 | 23 | 3 | |
| | | | | | = | | | |
| | | | | Result: | 19 | 11 | 9 | As letters: T l j |

*(A note on capitalization: In a computer, letters are stored using the standard ASCII code. This represents the capital letter alphabet using the numbers from 65 to 90, and lowercase letters using the numbers from 96 to 121. So to perform a shift on a letter in ASCII code, you first subtract either 65 or 96, perform the shift as above, then add 65 or 96 again to return it to ASCII while preserving the case. The caesar.py example program shows one way to do this.)*

To practice your cipher skills on a one-time pad, see if you can decode this exciting message:

Message: **Hwpcr Bug!**

Pad: **QWERTYUIOP**

*Notes: remember that punctuation and spaces don't get encoded, and don't use up a pad character. If the pad is longer than the message, you won't use all of it.*