BLOCK CIPHERS

MATH 195

Some other groups that we shall encounter in the following:

- $\mathbb{Z}/2\mathbb{Z}$, with group law + or $\oplus$ or XOR.
- $(\mathbb{Z}/2\mathbb{Z})^3 = \{(a_1, a_2, a_3) : a_i \in \mathbb{Z}/2\mathbb{Z}\}$. In general, the set $(\mathbb{Z}/n\mathbb{Z})^k$ is an additive abelian group with respect to componentwise addition modulo n. This has the number of elements $\#(\mathbb{Z}/n\mathbb{Z})^k = n^k$. For example, in $(\mathbb{Z}/2\mathbb{Z})^3$ we have $(1,0,1) + (0,1,1) = (1,1,0)$.

Block Cipher

[Block ciphers are discussed in §3.2.]

Say your alphabet has n elements: e.g. $n = 2$ (bits), $n = 26$ (letters), $n = 10$ (digits). The message is a finite sequence of symbols from the alphabet. Since any given message is finite but they can be infinitely long, we pick a positive integer k (e.g. $k = 1, 3, 64, 2000$), and chop up your message in blocks of length k. For example, we might have `LET|USM|EET|TOM|ORR|OWX`.

We then let our plaintext space be $\mathcal{P}$ be words of length k on your alphabet: we identify this with $\mathcal{P} = (\mathbb{Z}/n\mathbb{Z})^k$. Each (plaintext) message is now a finite sequence $(X_1, X_2, \ldots, X_\ell)$ of elements of $\mathcal{P}$. Encrypt them using the same key $K \in \mathcal{K}$, so that $\mathcal{C} = \mathcal{P}$. Then Alice sends $(E_K(X_1), \ldots, E_K(X_\ell))$ to Bob, and Bob applies D_K to each of the $E_K(X_i)$ to recover the X_i.

As an example, we consider the *Vigenère cipher* [§2.3]. It has $n = 26$,

$$A = 0, B = 1, \ldots, Z = 25,$$

$k = 9$, $\mathcal{P} = \mathcal{C} = (\mathbb{Z}/26\mathbb{Z})^9 = \mathcal{K}$, and $E_K(X) = X + K$ (in the group $(\mathbb{Z}/26\mathbb{Z})^9$). With

$$K = \texttt{DISCOVERY} = (3, 8, 18, 2, 14, 21, 4, 17, 24),$$

and

$$X = \texttt{WEAREDISC} = (22, 4, 0, 17, 4, 3, 8, 18, 2),$$

we have

$$E_K(X) = (25, 12, 18, 19, 18, 17, 24, 12, 9, 0) = \texttt{ZMSTSRYMJA}.$$

In this case, $D_K(X) = X - K$.

We can also use stream ciphers. [See §3.2.]

This is some of the material covered January 31, in Math 195: Cryptography, taught by Hendrik Lenstra, prepared by John Voight `jvoight@math.berkeley.edu`.