Crypto Lecture 3

§ 2.9 The Hill Cipher.

The Hill cipher uses linear algebra to encrypt messages. First, a brief refresher:

If A is a 2×2 matrix and B is a 2×2 matrix: $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, $B = \begin{pmatrix} e & f \\ g & h \end{pmatrix}$, then their product is

$$AB = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{pmatrix}.$$

When $\vec{v} = \begin{pmatrix} x \\ y \end{pmatrix}$ is a vector, we multiply

$$A\vec{v} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax+by \\ cx+dy \end{pmatrix}.$$

The matrix $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ is the identity,

and if a matrix C satisfies $CA = AC = I$, then $C = A^{-1}$ is "the inverse of A". The formula is:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}, \quad ad-bc \neq 0.$$

In general, if A and B are $n\times n$ matrices,

say $A = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}$ (n row vectors)

and $B = (v_1\ v_2 \cdots v_n)$ (n column vectors), then

$$AB = \begin{pmatrix} r_1 \cdot v_1 & r_1 \cdot v_2 & \cdots & r_1 \cdot v_n \\ r_2 \cdot v_1 & r_2 \cdot v_2 & & \vdots \\ \vdots & & & \\ r_n \cdot v_1 & \cdots & \cdots & r_n \cdot v_n \end{pmatrix} \quad (n \times n \text{ again}).$$

where $\vec{r} \cdot \vec{v}$ is the usual dot product of vectors:

$$(a_1\ a_2 \cdots a_n) \cdot \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n.$$

Then $A\vec{v} = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix} v = \begin{pmatrix} r_1 \cdot v \\ r_2 \cdot v \\ \vdots \\ r_n \cdot v \end{pmatrix}$ if v is an n-dim column vector.

As before, $I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{pmatrix}$ is the identity matrix,

and if C satisfies $AC = CA = I$, then $C = A^{-1}$ is the inverse of A.

The algorithm: (Let's do the 2×2 case)

- Choose a key, which is an $n \times n$ matrix A that has an inverse A^{-1}, note that both A and A^{-1} must have integer entries!

(so if A is 2×2, $ad-bc$ must be invertible mod 26)

E.g. $A = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}$.

- Break your plaintext into blocks of length n, padding if necessary. Multiply each by A.

E.g. If our plaintext is "help" then

plaintext: he lp

as vectors: $\begin{pmatrix} 7 \\ 4 \end{pmatrix}$ (H, e) $\quad \begin{pmatrix} 11 \\ 15 \end{pmatrix}$ (L, P)

multiply by $A \pmod{26}$:

$$\begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}\begin{pmatrix} 7 \\ 4 \end{pmatrix} = \begin{pmatrix} 33 \\ 34 \end{pmatrix} \pmod{26} = \begin{pmatrix} 7 \\ 8 \end{pmatrix} \bmod 26.$$

$$\begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}\begin{pmatrix} 11 \\ 15 \end{pmatrix} = \begin{pmatrix} 78 \\ 97 \end{pmatrix} \pmod{26} = \begin{pmatrix} 0 \\ 19 \end{pmatrix}$$

ciphertext: 7 8 0 19

h i a t

To decrypt, we need to find A^{-1}. Use the formula:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

$$\frac{1}{15-6}\begin{pmatrix}5 & -3\\ -2 & 3\end{pmatrix}$$ (but what is this mod 26?)

$$= \frac{1}{\cancel{15-6}\ 9}\begin{pmatrix}5 & 23\\ 24 & 3\end{pmatrix}$$

But what is $\frac{1}{9}$ mod 26? Here, we want a number 'x' so that $9 \cdot x \equiv 1 \bmod 26$. This is what we mean by "$\frac{1}{9}$ mod 26". So start testing; and note:

$9 \cdot 3 = 27 \equiv 1 \bmod 26$. So

$$A^{-1} = 3\begin{pmatrix}5 & 23\\ 24 & 3\end{pmatrix} \bmod 26 = \begin{pmatrix}15 & 17\\ 20 & 9\end{pmatrix}.$$

So to decrypt:

ciphertext : h i a t

as vectors $\begin{pmatrix}7\\ 8\end{pmatrix}$ $\begin{pmatrix}0\\ 19\end{pmatrix}$

multiply by A^{-1} mod 26

$$\begin{pmatrix}15 & 17\\ 20 & 9\end{pmatrix}\begin{pmatrix}7\\ 8\end{pmatrix} = \begin{pmatrix}7\\ 4\end{pmatrix} \bmod 26$$

$$\begin{pmatrix}15 & 17\\ 20 & 9\end{pmatrix}\begin{pmatrix}0\\ 19\end{pmatrix} = \begin{pmatrix}11\\ 15\end{pmatrix} \bmod 26$$

plaintext 7 4 11 15

h e l p

Remarks: • This can obviously be done by using any block size n.

• This overcomes the "position dependence" of the earlier ciphers that used blocks. The position of a letter (mod the block size) does not, in any obvious way, determine how it is encrypted. It is encoded as a linear combination of other letters in the block.

Chapter 3 The Enigma machine and Ultra.

Goal: Describe the encryption algorithm used by the Germans in WW II.

However, because the algorithm was implemented on a specialized machine (the enigma), we cannot realistically describe the algorithm without explaining the physical functioning of the machine.

So, first: The physical machine.
Second: The enigma algorithm.

The machine:

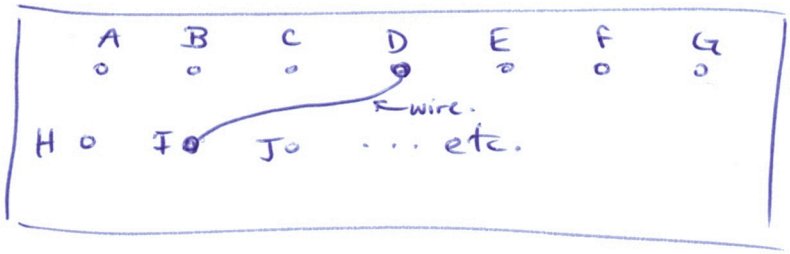
The machine has 3 basic parts we need to understand:

1. The plugboard
2. The rotors and drum
3. The reflector plate.

The plugboard:

Literally a board with one plug for each letter. The plugs can be connected by wires. The placement of the wires will determine how letters (signals) that pass through the plugboard are scrambled:

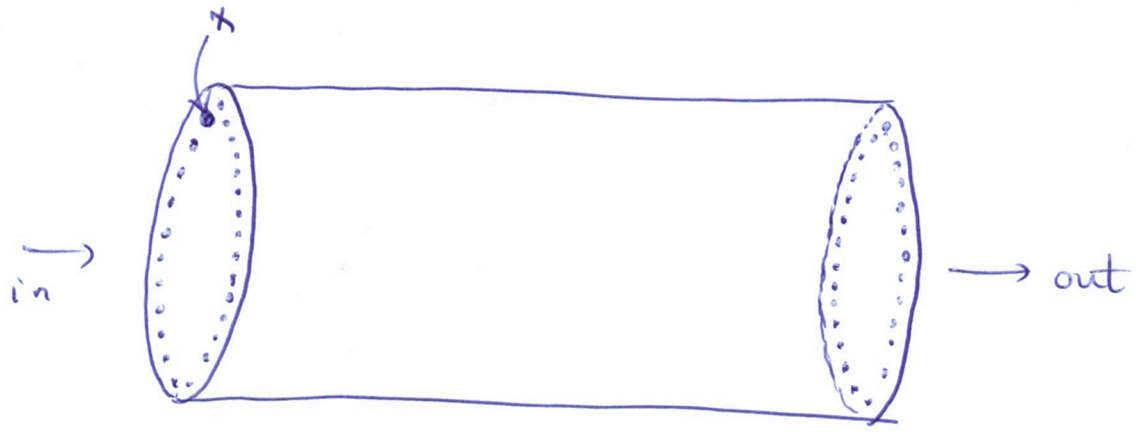
E.g.



The wire connecting I to D guarantees that an electrical signal entering as "I" exits as "D" and vice versa. You can place up to 13 wires.

The rotors and drum:

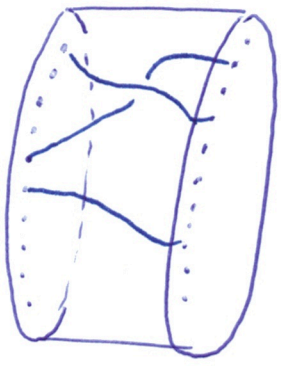
The drum is a cylindrical cavity in the machine with 26 electrodes on each end, spaced evenly as follows:



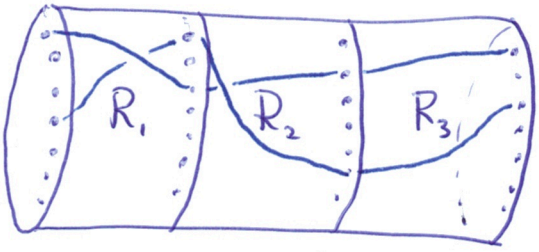
Electrical signals enter one side, and will come out the other. The signal entering at the position marked "x" will potentially come out of any of the 26 outputs on the other side, depending on how the signals are "scrambled" as they pass through the drum.

The drum is filled with rotors, 3 of them, which do the scrambling.

A rotor is a cylindrical object with 26 electrodes on each side. Inside the rotor is wiring that connects the electrodes on one side to those on the other in some way:



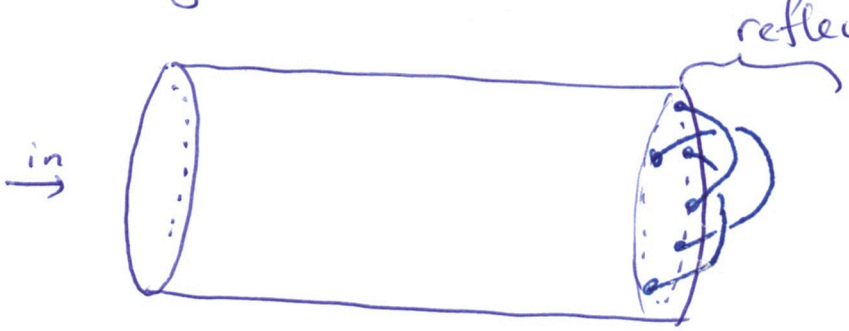
3 of them fit in the drum, and they can be switched in/out easily:



A machine would come with a large collection of rotors to swap in/out.

The reflector plate:

At one end of the drum (I'll draw it on the left), signals are turned around and sent back through the drum! This is because the outputs are hardwired to themselves in some predetermined way (ie this can't be changed).



How the machine works:

You push a button. The signal:

- passes through the plugboard
- through the rotors
- through the reflector plate
- back through the rotors

- back through the plugboard
- a light goes on next to the "output letter", telling you what your input is to be replaced with in order to make the ciphertext.
- The ~~drums~~ rotors rotate to prepare for the next input!!!

Additional rotor information:

- Each rotor rotates in the drum.
- R_1 turns one "tick" each time you encode a letter.
- R_2 turns one "tick" for each complete turn of R_1
- R_3 turns one "tick" " " " " " R_2.

How? There are "notches" on adjacent rotors. When the notch on R_1 passes by the notch on R_2, it turns it 1 "click".

- So after inputting a letter, the plugboard & reflector plates are the same, but the drum scrambles differently!

The algorithm:

- Choose a way of wiring the plugboard, ie a permutation P of the alphabet.
- Choose 3 rotors, ie choose permutations R_1, R_2, R_3 of the alphabet.
- Choose the intial positions of the rotors,

- That is, choose how many "ticks" R_1 has to turn before it causes R_2 to turn for the first time. Ditto for R_2 causing R_3 to turn.
- Let "R" (no subscript) denote the permutation of the alphabet done by the reflector plate (always the same).
- Encrypt the first letter by applying the permutation

$$P^{-1} \cdot R_1^{-1} \cdot R_2^{-1} \cdot R_3^{-1} \cdot R \cdot R_3 \cdot R_2 \cdot R_1 \cdot P.$$

Now let "S" denote the permutation of the alphabet corresponding to $S(x) = x+1 \pmod{26}$.

- Encrypt the second letter by applying

$$P^{-1} S R_1 S^{-1} R_2^{-1} R_3^{-1} R R_3 R_2 S R_1 S^{-1} P$$

- Encrypt the next letter with $S^2 R_1 S^{-2}$ in place of R_1, then $S^3 R_1 S^{-3}$, etc.
- Suppose after k "ticks" of R_1, the second rotor finally turns one tick. Then we use

$$P^{-1} S^{k+1} R_1 S^{-(k+1)} S R_2 S^{-1} R_3^{-1} R R_3 S R_2 S^{-1} S^{k+1} R_1 S^{-(k+1)} P$$

on the (k+1)-letter.

- Continuing in this way, we put higher and higher powers of "S", with powers reduced mod 26, as we encode our entire message.

To decrypt:

Set up your machine in the same way as for encrypting. Run your ciphertext through it. ~~It~~ This applies

$$P^{-1} R_1^{-1} R_2^{-1} R_3^{-1} R R_3 R_2 R_1 P$$

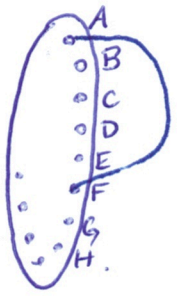
to the first letter of your ciphertext. So overall, you're done:

$$\left(P^{-1} R_1^{-1} R_2^{-1} R_3^{-1} R R_3 R_2 R_1 P\right)^2$$

$$= P^{-1} R_1^{-1} R_2^{-1} R_3^{-1} R^2 R_3 R_2 R_1 P$$

to ~~your~~ the first letter of the plaintext.

But: Look at the permutation R from the reflector plate:



If it sends A to F, then doing it again sends F back to A! I.e., R^2 = identity.

So $P^{-1} R_1^{-1} R_2^{-1} R_3^{-1} \underbrace{R^2}_{\text{id}} R_3 R_2 R_1 P$ = identity.

So running the enigma machine a second time on your ciphertext will decode it.

Lecture 4: Permutations, combinations, and the "expected" security of the Enigma machine.

We first review some basic counting arguments.

Permutations: We already saw that the number of permutations of the alphabet is 26!, and more generally, the number of permutations of a set of "n" things is n!.

Such a permutation was a choice of 26 letters, in order. First we had 26 choices, then 25, then 24, etc. If we had instead only wanted to choose, say, 3 letters in order, then we would want to stop after 3 steps:

$$26 \cdot 25 \cdot 24 \cdot 23 \cdot 22 \cdot 21 \cdot \ldots \cdot 1$$

↑ first choice; ↑ second choice; ↑ third choice STOP HERE; ↑ fourth choice ....

We can write the result $26 \cdot 25 \cdot 24$ as

$$\frac{26!}{(26-3)!} = \frac{26 \cdot 25 \cdot 24 \cdot 23 \cdot \ldots \cdot 1}{23 \cdot 22 \cdot \ldots \cdot 1}.$$

This formula holds in a more general setting: The number of ways of choosing k things

In order from a collection of n things is

$$P(n,k) = \frac{n!}{(n-k)!}$$, this counts

the number of permutations of k things chosen from n

Combinations:

A combination of k objects, chosen from a collection of n total objects, is a way of choosing k objects when order doesn't matter.

Thinking of the alphabet again: How many sets of three letters (distinct letters) can we make from the English alphabet?

We already know there are

$$\frac{26!}{(26-3)!}$$

ways of choosing 3 letters when order matters. So this formula will count things like

(a, b, c), (a, c, b), (b, a, c), ... etc

separately. We want all of these choices to "be the same". There are $3!$ such choices, so we divide and get

$$\frac{26!}{3!(26-3)!}$$ subsets of size 3 in a 26-letter alphabet.

So in general: The number of ways of choosing k distinct things from a collection of n things when order does not matter is

$$C(n,k) = \frac{n!}{k!\,(n-k)!}$$

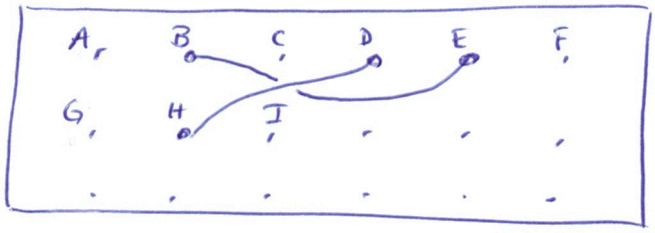
, which counts the number of combinations of k things chosen from n.

The number $C(n,k)$ is often written as $\binom{n}{k}$, is called a binomial coefficient and is read ~~else~~ as "n choose k".

Security of the enigma:

(1) The plugboard.

The plugboard could have anywhere from zero to thirteen cables plugged into it:



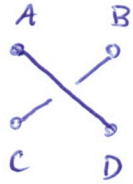
Plugboard with two cables.

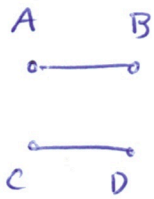
∴ Plugging in one cable amounts to selecting two letters to connect, so the number of ways to configure the plugboard using 1 cable is

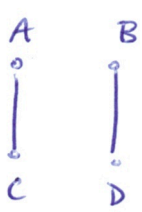
$$C(26,2) = \frac{26!}{2!\,(26-2)!}$$

- Two cables is more complicated. We start by choosing four letters, there are $C(26, 4)$ ways to do this.

Now there a few different ways to connect 4 letters with 2 cables:







(that's all)

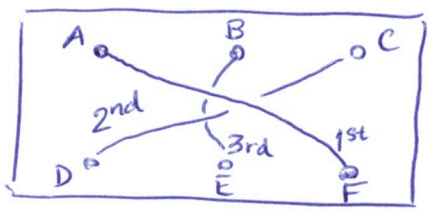
So there are $C(26, 4) \cdot 3$ ways to configure the plugboard with 2 cables.

- Three cables. We start with $C(26, 6)$, but now need to count the number of ways to connect 6 letters with 3 cables.

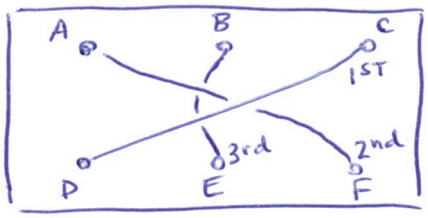
To count this, note that:

- There are $C(6, 2)$ ways to ~~choose~~ connect the first cable
- There are then only 4 letters left, so now $C(4, 2)$ ways to connect the second cable.
- Only $C(2, 2) = 1$ way to connect the last cable.

Problem with this argument: The two configurations below are counted separately because the order of cables mattered in our argument:



vs



So we have to divide by the number of ways of ordering 3 things (3 pairs of ~~ALEX~~ letters, in this case) to correct for overcounting. We get

$$\frac{C(26,6)\cdot C(6,2)\cdot C(4,2)\cdot C(2,2)}{3!}$$

ways to use 3 cables.

Remark: $C(6,2)\cdot C(4,2)\cdot C(2,2)\cdot \frac{1}{3!}$

$$= \frac{6!}{2!(6-2)!}\cdot\frac{4!}{2!(4-2)!}\cdot\frac{2!}{2!(2-2)!}\cdot\frac{1}{3!}$$

$$= \frac{6\cdot5\cdot4\cdot3\cdot2\cdot1}{2!\cdot2!\cdot2!\cdot3!} = \frac{6\cdot5\cdot4\cdot3\cdot2\cdot1}{(2\cdot3)\cdot(2\cdot2)\cdot(2\cdot1)} = 5\cdot3\cdot1.$$

New notation: $n!!$ will be the product:

$n!! = n\cdot(n-2)(n-4)\cdot\ldots\cdot 2$ if n even

$n!! = n(n-2)(n-4)\cdot\ldots\cdot 1$ if n odd.

So the number of ways of ~~pair~~ grouping 6 letters into 3 pairs is

$$\frac{C(6,2)\cdot C(4,2)\cdot C(2,2)}{3!} = 5!!$$

In general (Theorem): The number of ways of grouping p things (p an even integer) into $\frac{p}{2}$ pairs will be

$$(p-1)!!$$

Proof of Theorem, by induction:

We know the claimed formula holds for $p=6$.
Now suppose it is true for some even number $p=2k$.
(ie suppose the number of ways of pairing up $2k$ things is $(2k-1)!!$). Consider the case of $2k+2$ things.

Choose one object from $2k+2$ of them. There are $2k+1$ choices for a second element to make a pair. After this first choice, there are $2k$ objects remaining to be paired up, so $(2k-1)!!$ possibilities. Overall this gives

$$(2k+1)\cdot(2k-1)!! = (2k+1)!!$$

So the theorem holds.

Remark: What is wrong with the following argument:
There are $C(2k+2, 2)$ ways of choosing the first pair, and $(2k-1)!!$ ways of pairing up the remaining objects. So the number of ways of pairing up $2k+2$ objects is

$$C(2k+2, 2)\cdot(2k-1)!!$$

(It is wrong, but why?)

Conclusion: The number of ways of configuring the plugboard using k cables is:

$$\binom{26}{2k}(2k-1)!!$$

$\binom{26}{2k}$: # ways of choosing $2k$ letters to connect with cables

$(2k-1)!!$: # of ways to connect $2k$ letters.

So the total number of plugboard configurations is:

$$1 + \binom{26}{2}1!! + \binom{26}{4}3!! + \binom{26}{6}5!! + \binom{26}{8}7!! \ldots + \binom{26}{26}25!!$$

$$\approx 5.32 \times 10^{14}.$$

The rotors and drum, pluse reflector plate:

- Each rotor encoded a permutation of the alphabet, there are $26!$ such permutations, and 3 rotors. So $(26!)^3$ possibilities for choices of rotors.
- Each rotor also had an initial position, and there were 26 possible initial positions. So overall, 26^3 initial positions.

- The reflector plate was basically a plugboard with 13 cables (every letter is connected to another). There are $25!!$ possible configurations.
- The positions of the notches on the rotors. This is encoded by two ~~letters~~ numbers $k, \ell \in \{1, \dots, 26\}$. So 26^2 variable positions of the notches.

Overall, there are

$(26!)^3 \cdot 26^3 \cdot 26^2 \cdot 25!! \approx 6.16 \times 10^{99}$ configurations of the drum and reflector plate.

Total number of Enigma states:

$\sim 6.16 \times 10^{99} \times 5.32 \times 10^{14} \simeq 5 \times 10^{114}$.

Consequently, it was believed that the allies had no hope of cracking the enigma (e.g. even with astronomical computing power: one supercomputer for each atom in the universe we would only have a 1% chance of cracking the code by brute force if we ran all computers for 14 billion years (age of universe).