Enigma

Protecting information – From Classical Error Correction to Quantum Cryptography, Chapter 1.2

30/01/2008

Brief Background and History

- Enigma is probably the most famous encryption cipher
  - For most it is known as the Enigma Machine
- Originally it was invented by German electric- and mechanics engineer Albert Scherbius
  - He wanted to find a replacement to the WWI encryption tools / booklets with something that uses modern technology
  - Basically the first implementations where electronically working Leon Alberti's encryption discs (the oldest encryption machine from the 15th century)
- German Navy took the Enigma Machines in use in around 1925
  - Started to modify it into a more complex form

What is the Enigma?

- Can be described with purely mathematical terms but the cipher is usually tied to a machine
- The main cryptographic components of the machine are:
  - *A Plugboard*
  - *Three+ Rotors*
  - *A Reflector*
- Each of these components has the effect of permuting the alphabet
  - Electrical wires that "connect" the input letter to the output letter
- Basic idea – Decrypting a message with the same machine settings as it was encrypted provides us with the plaintext

How does the machine work? - Plugboard

- Includes an array of 26 jacks, one for each letter, and six electrical cables
- Each cable can be plugged into two jacks / two letters
  - This will do an interchange of those two letters
    - E.g. A ⇔ Y
- Letters that are not "plugged" are left unchanged
- Left alone the plugboard is just a simple single substitution cipher and can be cracked easily using frequency analysis

How does the machine work? - Rotors

- Each *rotor* is a (circle) disk with 26 input locations on one side and 26 output locations on the other side
  - Identical on both sides
- Inside a rotor wires go from input location to an output location
  - A permutation without any special symmetries on how the wires go
    - To decrypt you need to know the order of the rotors (*rotor numbers)* as each of them had a different permutation
- Output from the plugboard goes as the input of the first rotor and the output from it goes as the input to the second rotor and so on…
- Permutation was one of the key elements, but the most important feature of the machine are
  - The starting orientation of the rotors can be changed
  - A rotors orientation can change between key strokes with respect to the other rotors
- Rotors don’t provide that many keys, but make the system unbreakable by frequency analysis

How does the machine work? - Reflector

- Reflector acts on the output of the last rotor and swaps the letters in pairs as new output to the rotors
- Permutation of the reflector is fixed
  - Some “versions” of the Enigma also allow adjusting the permutation of the reflector
- Permutation not limited to six pairs of letters as with the plugboard
- Every letter is sent to a different letter
  - Lead to the fact with the machine that no letter was ever encrypted as it self
    - This fact actually helped crack the enigma

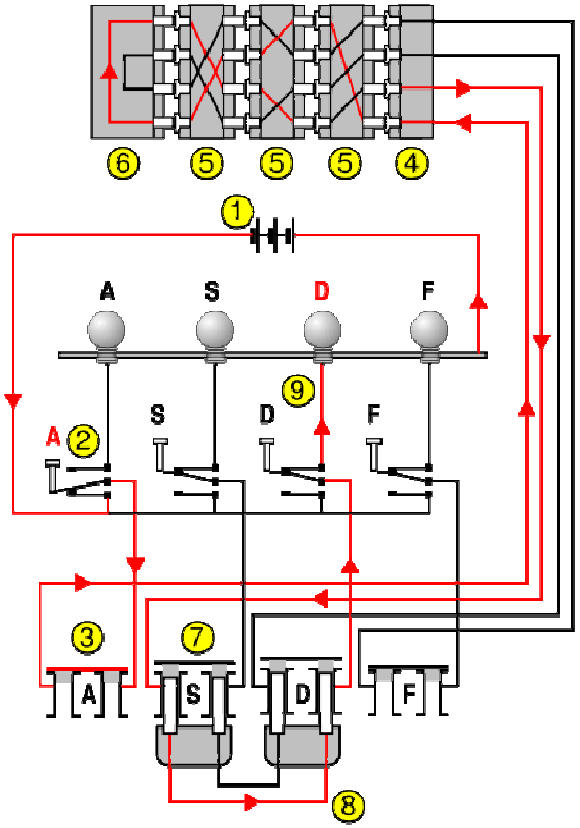
The Key = The Settings

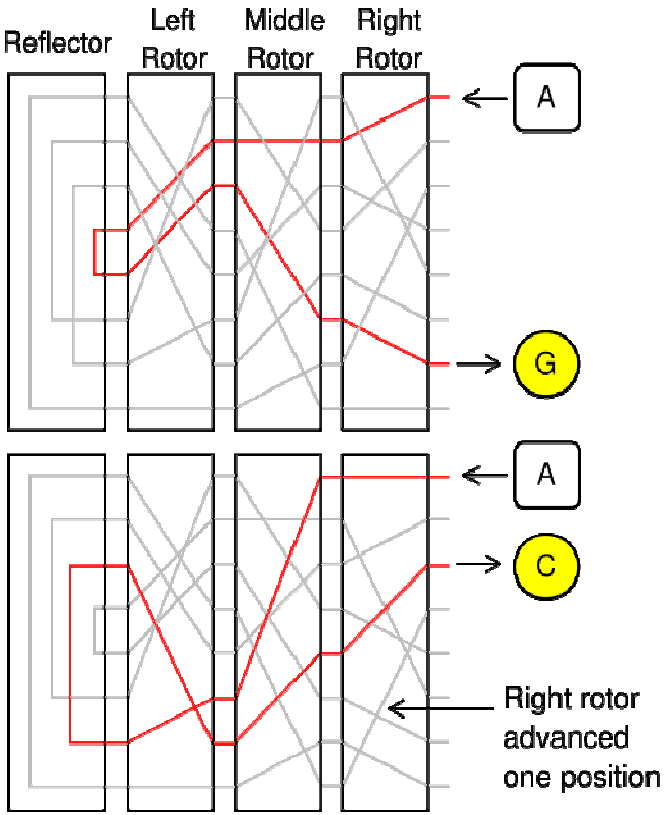
- Basically the “key” = “settings”
  - Plugboard settings
  - Order of rotors + which rotors are used
  - Starting position of the rotors
    - This was adjusted also on per message basis
- Each of the *rotors* can but to 26 different positions
  - The most common three rotor Enigma had 17576 possibilities
- Each of the rotors can be in a different order, so with three different rotors 6 alternatives (if only three rotors to pick from)
  - The number of rotors to chose from was increased later on e.g. Naval enigma had 8 rotors to pick from => 336 alternatives

The Key = The Settings (2)

- The *plugboard* provides us over 10^{11} different ways of matching six different pairs of letters with one and other
  - Number of cables was increased later on
- So as a total the Enigma already provided over 10^{16} different possibilities for a key
- With the more complex versions of the Enigma you could also
  - adjust the *reflector* and
  - adjust the *rotors* with *rings*
  - These provided additional security, but not additional crypto graphical challenges

How does the machine work? – Putting it all together





Mathematics behind the Enigma Cipher

- Let A be the plugboard's permutation, and we note that $A = A^{-1}$
- Let B be the reflector's permutation, and we note that $B = B^{-1}$
- Let R_i be the permutation executed by the *i*th rotor in "standard" orientation and
- Let S be the simple permutation A→B→C→…→Z→A and S^{-1} its inverse

- Rotors *i* permutation when in "standard" format is

$$S^{-1}R_iS$$

- when rotated *n* steps, meaning n/26 of a complete cycle the permutation is

$$S^{-n}R_iS^n$$

Mathematics behind the Enigma Cipher (2)

- After n steps
  - The rotor's indexes n_i = [0 , 25].
  - For n_1 = n mod 26
  - When n_1 goes from 25 to 0, n_2 moves a step: $n_2 = n_1$ mod 26
  - Same applies to n_3
  - Note that the starting index with each rotor in comparison to the index that the neighboring rotors moves a step might not be zero
- Putting these all together the mapping that the machine implements at every keystroke is (read from right to left)

$$A(S^{-n_1}R_1S^{n_1})^{-1}(S^{-n_2}R_2S^{n_2})^{-1}(S^{-n_3}R_2S^{n_3})^{-1}B(S^{-n_1}R_1S^{n_1})(S^{-n_2}R_2S^{n_2})(S^{-n_3}R_3S^{n_3})A$$

$$\Leftrightarrow \quad AS^{-n_1}R_1^{-1}S^{(n_1-n_2)}R_2^{-1}S^{(n_2-n_3)}R_3^{-1}S^{n_3}BS^{-n_3}R_3S^{(n_3-n_2)}R_2S^{(n_2-n_1)}R_1S^{n_1}A$$

How was the Enigma used during WWII

Message exchange – encrypt

1. Set machine according to the daily key
2. Type the random message key *twice* and send the resulting six (eight) encrypted letters
3. Reset the rotors according to the selected message key
4. Encrypt the message and send it

Message exchange - decrypt

1. Set machine according to the daily key
2. Type in the first six (eight) letters of the cipher text to read the message key
3. Reset the rotors according to the decrypted message key
4. Decrypt the rest of message

Cracking Enigma

- Enigma was cracked for the first time many years before the war started by the young Polish Crypto analyst Marian Rejewski
- The Poles had managed to built an exact copy of the German Military Enigma
  - Help from the French Intelligence and a bitter German informant
  - They had already had the commercial copy of the machine – did not help at all
- Even having an exact copy of the machine was not enough - needed to know the key

- Q Z A E L L
- R S Z J J Q
- E X T S I N
- S R W Q Y K
- Q P C E B D
- J P T R B N

Cracking Enigma (2)

- Rejewski focused to finding repetition from the ciphers
  - The most obvious repeated text was the *message key* sent in the beginning of each message twice
- Rejewski managed to find cycles from the messages e.g. Q → E → S → Q had a cycle of length three
- Having enough messages one could write down all cycles produced by the daily key
- These lengths are not depended on the plugboard settings → separation of the plugboard and the rotor settings
  - The number of keys to find dropped to about 100 000
- Poles build a catalog of each cycle length = basically a different representation of the rotor settings of the daily key
- What was left was to cracking the plugboard settings
  - Not quite but almost as easy as cracking a substitution cipher

Cracking Enigma (3)

- Germans made the machine a bit more secure and the war was imminent
  - Poles couldn't crack it anymore
  - Information was given to the British and the French
- The British continued the Poles work
- Alan Turing, built a machine to crack the Enigma called the Turing Bombe
  - Gave the foundation for modern computing
  - Could crack the Enigma in about 15 hours
- Basically Germans had some weaknesses on how they used the enigma
  - E.g. messages sent on the same weather station contained predictable information and words – no need to encrypt known information…

Summary

- A relatively famous cipher which is usually tide up to a machine
- Math behind the cipher is not complex, but even with modern computers it isn’t the quickest thing to crack
- Misuse of the system made it vulnerable
- Gave the bases to modern computer science, if not directly, through the work of Alan Turing on cracking the system