Introduction to Cryptography CS 355

Lecture 7

Mini Review & Enigma Machine

Answers to Quiz 1 problems

- What is the Caesar Cipher?
  - Shift cipher with shift 3
- Types of classical ciphers:
  - Transposition ciphers, e.g., Scytale cipher
  - Mono-alphabetical substitution ciphers
  - Poly-alphabetical substitution ciphers, e.g., the Vigenère cipher
  - Substituting more than one letter at a time, e.g., Playfair and ADFGX

Attacking Substitution Cipher

- Attacking substitution cipher under different adversary model
  - ciphertext only: frequency analysis
  - known plaintext: derive partial key, then use frequency analysis if necessary
  - chosen plaintext:
    - if can choose a string of 26+ letters, enumerate the alphabet
    - if not, then choose the ones that help cryptanalysis the most
  - chosen ciphertext: ?

The Vigenère Cipher

- Two phases in (ciphertext only) cryptanalysis
- Phase 1: finding out the key length
  - The Kasiski attack
  - Using index of coincidence to verify guesses of key length
- Phase 2: finding out the key (and thus the plaintext)
  - How to do this?
- What about known plaintext attack?

Review of Number Theory

- The extended Euclidian algorithm
  - given a, n, computes d=gcd(a,n) and integers s, t such that a?s+n?t = d

- Solving linear congruence $ax \equiv b \bmod n$
  - when gcd(a,n)=1, compute s such that a?s+n?t = 1, the solution is x=s?b mod n

Chinese Reminder Theorem (CRT)

Theorem

Let n_1, n_2, ,,, n_k be integers s.t. $\gcd(n_i, n_j) = 1$ for any $i \neq j$.

$$x \equiv a_1 \bmod n_1$$

$$x \equiv a_2 \bmod n_2$$

$$\ldots$$

$$x \equiv a_k \bmod n_k$$

There exists a unique solution modulo

$n = n_1 n_2 \ldots n_k$

Proof of CRT

- Consider the function χ: $Z_n \rightarrow Z_{n1} \times Z_{n2} \times \ldots \times Z_{nk}$
  $\chi(x) = (x \bmod n_1, \ldots, x \bmod n_k)$
- We need to prove that χ is a bijection.
- For $1 \leq i \leq k$, define $m_i = n / n_i$, then $\gcd(m_i, n_i) = 1$
- For $1 \leq i \leq k$, define $y_i = m_i^{-1} \bmod n_i$
- Define function $\rho(a1,a2,\ldots,ak) = \Sigma\ a_i m_i y_i \bmod n$, this function inverts χ
  - $a_i m_i y_i \equiv a_i \pmod{n_i}$
  - $a_i m_i y_i \equiv 0 \pmod{n_j}$ where $i \neq j$

Example of CRT:

$x \equiv 5 \pmod{7}$

$x \equiv 3 \pmod{11}$

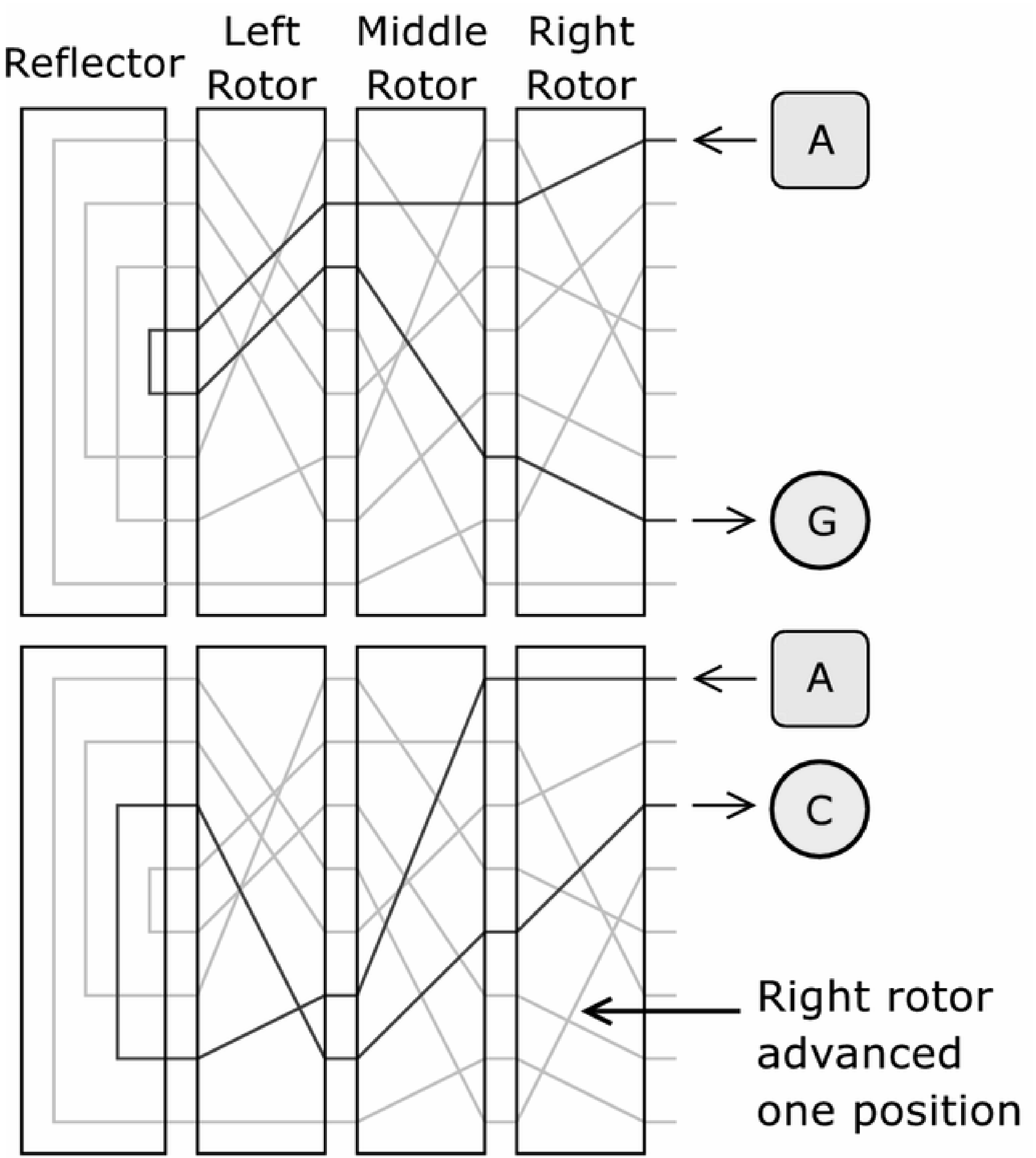
$x \equiv 10 \pmod{13}$

- $n_1=7$, $n_2=11$, $n_3=13$, $n=1001$
- $m_1=143$, $m_2=91$, $m_3=77$
- $y_1=143^{-1} \bmod 7 = 3^{-1} \bmod 7 = 5$
- $y_2=91^{-1} \bmod 11 = 3^{-1} \bmod 11 = 4$
- $y_3=77^{-1} \bmod 13 = 12^{-1} \bmod 13 = 12$

- $x = (5\times143\times5 + 3\times91\times4 + 10\times77\times12) \bmod 1001$
  $= 13907 \bmod 1001 = 894$

History of the Enigma Machine

- Patented by Scherius in 1918
- Widely used by the Germans from 1926 to the end of second world war
- First successfully broken by Polish in the thirties by exploiting the repeating of the message key
- Then broken by the UK intelligence during the WW II


Reflector
Left Rotor
Middle Rotor
Right Rotor
A
G
A
C
Right rotor advanced one position

Using Enigma Machine

- A day key has the form
  - Plugboard setting: A/L–P/R–T/D–B/W–K/F–O/Y
  - Scrambler arrangement: 2-3-1
  - Scrambler starting position: Q-C-W
- Sender and receiver set up the machine the same way for each message
- Use of message key: a new scrambler starting position, e.g., PGH
  - first encrypt and send the message key, then set the machine to the new position and encrypt the message
  - initially the message key is encrypted twice

Permutations

- A **permutation** is a bijection from a finite set *X* onto itself.
- Each permutation has an inverse
- Given permutations P_1, P_2, their concatenation P_1P_2 is also a permutation; it is the permutation of first applying P_1, then applying P_2
- The inverse of P_1P_2 is $P_2^{-1}P_1^{-1}$
- E.g., P_1 = CBDEA, P_2=DAEBC, then
  P_1^{-1}=?, P_1P_2=?

Mathematical Description

- Let P denote the plugboard transformation
- Let L,M,R denote the three motors
- Let U denote the reflector,
- Then the encryption function
  $$E = PRMLUL^{-1}M^{-1}R^{-1}P^{-1}$$
- Fact 1: $E[x] \neq x$
- Fact 2: $E[E[x]] = x$

How to break the Enigma machine?

- Recover 3 secrets
  - Internal connections for the 3 motors
  - Daily keys
  - Message keys
- Exploiting the repetition of message keys
  - In each ciphertext, letters in positions 1 & 4 are the same letter encrypted under the day key
  - One can thus write equations in which the variables
  - With 2 months of day keys and Enigma usage instructions, the Polish mathematician Rejewski to reconstruct the internal wiring

How to recover the day key?

- Catalog of “characteristics”
  - Main idea: separating the effect of the plugboard setting from the starting position of motors
  - determine the motor positions first
  - then attacking plugboard is easy
  - plugboard does not affect chain lengths in the permutation
- Using known plaintext attack
  - stereotypical structure of messages, easy to predict standard reports, retransmission of messages between multiple networks,

Lessons Learned From Breaking Engima

- Keeping a machine (i.e., a cipher algorithm) secret does not help
  - The Kerckhoff’s principle
- Large number of keys are not sufficient
- Known plaintext attack was easy to mount
- Key management was the weakest link
- People were also the weakest link
- Never underestimate the opponent

Coming Attractions …

- One-time Pad, Pseudo Random Number Generator
- Recommended reading for next lecture:
  - Trappe & Washington: 2.9, 2.10