Cryptanalysis of the Affine Cipher

Ciphertext-only attack:
- Brute-force (try possible keys)
- Frequency analysis

Any other ideas?

Suppose we know two symbols and what they map to.

Example: \( 0 \rightarrow 3 \) \( \alpha = 1 \)
\( 7 \rightarrow 10 \) \( \beta = 3 \)
Cryptanalysis of the Affine Cipher

More challenging examples:

\[
\begin{align*}
4 & \rightarrow 17 & \alpha \cdot 4 + \beta & = 17 \pmod{26} \\
19 & \rightarrow 3 & \alpha \cdot 19 + \beta & = 3 \pmod{26}
\end{align*}
\]

for \( m = 26 \)

Subtracting the first congruence from the second:

\[
\begin{align*}
15\alpha & = -14 \equiv 12 \pmod{26} \\
15\alpha & \equiv 12 \pmod{26} \\
5\alpha & \equiv 4 \pmod{26}
\end{align*}
\]

\( \alpha \neq 6 \) not allowed since \( \gcd(\alpha, 26) \neq 1 \)

\[
\begin{align*}
4 & \rightarrow 17 & 15\alpha & \equiv -7 \equiv 19 \pmod{26} \\
19 & \rightarrow 10 & \alpha & \equiv 17/15 \pmod{26} \\
\alpha = 3 & \quad \text{(bec. } 15 \cdot 3 = 45 \equiv 19 \pmod{26})
\end{align*}
\]

\[
\begin{align*}
\beta & = 5
\end{align*}
\]
Cryptanalysis of the Affine Cipher

Remarks:

- For which attacks do we have two pairs of symbols and their maps?
  
  chosen plaintext
  chosen ciphertext
  known plaintext - assuming the plaintext includes 2 different symbols

- Can we use this idea for the ciphertext-only attack?
  
  No, but we can still brute-force / freq. analysis
  also can use frequency analysis to find the pairs (e.g. most common is e, we have its map from the ciphertext)

- What if we have only one pair of a symbol and its map?
  
  E.g.: 4 \rightarrow 17 then we brute force a's (12 possibilities),
  \beta follows from \quad 4a + \beta = 17 \pmod{26}

- Read Section 3.3 to learn more about congruences.
Recall that there are 12 elements $a \in \mathbb{Z}_{26}$ such that $\gcd(a,26)=1$.

In general, for $m>0$, the number of elements of $\mathbb{Z}_m$ that are relatively prime to $m$ is denoted by $\phi(m)$ and it is usually referred to as the Euler phi function.

Examples:

$\phi(26) = 12$

$\phi(p) = p-1$ if $p$ is a prime

$\phi(p^i) = p^i - p^{i-1} = (p-1)p^{i-1}$ if $p$ is a prime

E.g.

$p = 2 \quad k = 3$

$\phi(2^3) = \phi(8) = 4$

$\mathbb{Z}_8 = \{0,1,3,5,6,7\}$

$\phi(2^7) = \phi(3^3) = 18$
For non-prime numbers:

Suppose \[ m = \prod_{i=1}^{n} p_i^{e_i} \]

where the \( p_i \)'s are distinct primes and \( e_i > 0 \) for \( i \in \{1,2,\ldots,n\} \).

Then,

\[
\phi(m) = \prod_{i=1}^{n} (p_i^{e_i} - p_i^{e_i - 1})
\]

\[
\phi(26) = (p_1^{e_1} - p_1^{e_1 - 1}) \cdot (p_2^{e_2} - p_2^{e_2 - 1}) = \begin{cases} p_1 = 2, & p_2 = 13, \ e_1 = 1, \ e_2 = 1 \\
\end{cases}
\]

\[
= (2^{1} - 2^{0}) \cdot (13^{1} - 13^{0}) = (2^{1})(13^{1}) = 12
\]

How many keys do we have for the affine cipher over \( \mathbb{Z}_m \) ?

\[
\phi(m) \cdot m
\]

# choices for \( \alpha \) # choices for \( \beta \)
Some More Number Theory

Computing $a^{-1} \pmod{m}$:
- Option 1: brute-force
  - advantages / disadvantages?

  takes a really long time, e.g. $m = 10^{200}$

- Option 2: ???

First, some math analysis:
For which $a \in \mathbb{Z}_m$ the inverse $a^{-1}$ does not exist in $\mathbb{Z}_m$?

all a's s.t. $a$ is not relatively prime w. $m$
(then for every $x$, $a \cdot x$ will be divisible by the gcd$(a,m) = \text{never } 1$)
Some More Number Theory

Computing gcd(a,b):

Euclidean algorithm

\[ a_1 = 62 \quad b_1 = 14 \]

remainder of \( a/b \) here: 6

\[ a_2 = 14 \quad b_2 = 6 \]
\[ a_3 = 6 \quad b_3 = 2 \]
\[ a_4 = 2 \quad b_4 = 0 \]

\( \text{gcd} \)
Euclidean Algorithm

Def EuclideanAlgorithm (a,b): // a,b>0, integers

\[ r_0 = a, \ r_1 = b \]
\[ m = 0 \]

while \( r_{m+1} \neq 0 \):

\[ m++ \]
\[ q_m = \lfloor \frac{r_{m-1}}{r_m} \rfloor \]
\[ r_{m+1} = r_{m-1} - q_m r_m \]

return \( r_m \)

### Example

\[ a = 62, \ \ b = 14 \]
\[ r_0 = 62, \ r_1 = 14, \ r_2 = 6, \ r_3 = 2, \ r_4 = 0 \]

Extended Euclidean Alg:

we will find \( s, t \) s.t.

\[ a \cdot s + b \cdot t = \gcd(a,b) \]
Extended Euclidean Algorithm

Given integers $a, b > 0$, it computes $r, s, t$ such that:

$$ r = \gcd(a, b) $$
$$ sa + tb = r $$

How is it helpful for computation of $a^{-1}$?

We first need to check that

$$ \gcd(a, m) = 1 $$

we get:

$$ s \cdot a + t \cdot m = 1 $$

$$ s \cdot a \equiv 1 \pmod{m} \quad \Rightarrow \quad s \cdot a \equiv 1 \pmod{m} $$

\[ a^{-1} \pmod{m} \]

i.e. find $a^{-1} \in \mathbb{Z}_m$

s.t. $a \cdot a^{-1} \equiv 1 \pmod{m}$

e.g.

$$ 28^{-1} \pmod{75} $$

run Extended Euclidean Algorithm

w. $a = 28$ (a)

$b = 75$ (m)

we are looking for $s$

(we did the computation on the board: $s = -8$ $r = 1$)

$$ t = 3 $$

then

$$ 28^{-1} \equiv -8 \equiv 67 \pmod{75} $$

$s$ is the multiplicative inverse
Extended Euclidean Algorithm

Def ExtendedEuclideanAlgorithm (a,b): // a,b>0, integers

r₀ = a, r₁ = b, s₀ = 1, s₁ = 0, t₀ = 0, t₁ = 1

m = 0

while rₘ₊₁ ≠ 0:

  m++

  qₘ = ⌊ rₘ₋₁ / rₘ ⌋

  rₘ₊₁ = rₘ₋₁ - qₘ rₘ

  tₘ₊₁ = tₘ₋₁ - qₘ tₘ

  sₘ₊₁ = sₘ₋₁ - qₘ sₘ

return rₘ, sₘ, tₘ

We will compute a sequence of sᵢ's and tᵢ's such that:

sᵢ · a + tᵢ · b = rᵢ

∀ i ∈ {0,1,...,m}
Extended Euclidean Algorithm

Remarks:
- By induction on $j$ we can show that for $j \in \{0, 1, \ldots, m\}$:
  $$r_j = s_ja + t_jb$$
  Hence, the algorithm is correct.

- We can save space by using many fewer variables.

- Running time?
  
  Suppose $n \geq a, b$
  
  $\leq 2\log n$ iterations, bec. if $a > b$ then $r_1 \geq 2r_{i+2}$
  
  (and if $a \leq b$ we need only one extra iteration)
  
  math operations (addition, multiplication, etc.) take $O(\log n \log \log n)$ steps
  
  usually we consider math operations to be $O(1)$
  
  but here we will work with very large numbers so we need to consider this running time as well.

OVERALL: $O(\log^2 n)$

when working a little harder on the analysis
Solving $ax \equiv c \pmod{m}$

- useful for cryptanalysis of e.g. the affine cipher

Possibilities:

- if $\gcd(a,m)=1$, then:
  \[ ax \equiv c \pmod{m} \]
  \[ x \equiv a^{-1}c \pmod{m} \]

- if $\gcd(a,m)=d>1$ and $d$ does not divide $c$, then:
  \[ ax \equiv c \pmod{m} \]
  \[ e.g. \quad 2x \equiv 3 \pmod{8} \]
  \[ \text{NO SOLUTION for } x \]

- if $\gcd(a,m)=d>1$ and $d$ divides $c$, then:
  \[ e.g. \quad 2x \equiv 6 \pmod{8} \]
  \[ x=3 \]
  \[ x=7 \]

Solve:

\[ \frac{a}{d} \cdot y \equiv \frac{c}{d} \pmod{\frac{m}{d}} \]

Then

\[ x \in \{ y, y \cdot \frac{m}{d}, y \cdot 2 \cdot \frac{m}{d}, \ldots, y \cdot \left(\frac{d-1}{d}\right) \cdot \frac{m}{d} \} \]