COMP 8920: Cryptography Winter 2025

Lecture 3 — Jan 14, 2025

Prof. Curtis Bright *Scribe: Aaron Barnoff*

1 Overview

This lecture continues with classical ciphers from Chapter 2.1. Previously covered were Shift, Substitution, and Affine ciphers.

2 Vigenère Cipher

The previous ciphers mapped each character to another character (fixed throughout the plaintext).

- These are called **monoalphabetic** ciphers.

The Vigenère cipher extends this idea using a key of length m:

- Encrypt m characters at a time.
- Add each plaintext character to the corresponding character in the key, for $1 \leq i \leq m$.
- This process is based on modular arithmetic.

Example: Let $k =$ 'CAT' and $P =$ 'HELLO'.

Align the key repeatedly above the plaintext and add character-wise:

$$\begin{array}{rcccc} C & A & T & C & A \\ +\,H & E & L & L & O \\ \hline J & E & N & N & O \end{array}$$

Expressing numerically (where $A = 0, B = 1, \ldots, Z = 25$):

$$(2, 0, 19, 2, 0) + (7, 4, 11, 11, 14) = (9, 4, 4, 13, 14).$$

Formally, define:

$$\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}^m.$$

For $k = (k_1, \ldots, k_m)$, encryption and decryption are:

$$e_k(x) = (x_1 + k_1, \ldots, x_m + k_m) \mod 26,$$

$$d_k(y) = (y_1 - k_1, \ldots, y_m - k_m) \mod 26.$$

The keyspace is of size $|\mathcal{K}| = 26^m$.

- Thus, even a moderate key length makes brute-force infeasible.

3 Hill Cipher

The Hill cipher encrypts messages using a **linear transformation**:

- Let A be an $m \times m$ invertible matrix over $\mathbb{Z}_{26}$.
- The plaintext is represented as an m-dimensional row vector.

Encryption: The transformation is:

$$x \mapsto xA \mod 26.$$

Decryption: Requires the matrix inverse:

$$y \mapsto yA^{-1} \mod 26.$$

The keyspace consists of all invertible $m \times m$ matrices:

$$\mathcal{K} = \{A \in \mathbb{Z}_{26}^{m\times m} \mid \det(A) \not\equiv 0 \mod 26\}.$$

For A^{-1} to exist, $\gcd(\det A, 26) = 1$ must hold.

Finding A^{-1}: The inverse of A is computed as:

$$A^{-1} = (\det(A))^{-1} \cdot \text{Adj}(A).$$

The adjugate (a of a matrix A is the transpose of its cofactor matrix:

$$\text{Adj}(A) = (\text{Cof}(A))^T.$$

The cofactor C_{ij} of an element a_{ij} in A is given by:

$$C_{ij} = (-1)^{i+j} M_{ij},$$

where M_{ij} is the minor of a_{ij}, and is the determinant of the submatrix obtained by removing the ith row and jth column of A.

Example: For a matrix $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$

The minors are:

$$M_{11} = a_{22}, \quad M_{12} = a_{21}, \quad M_{21} = a_{12}, \quad M_{22} = a_{11}$$

The cofactor matrix is:

$$\text{Cof}(A) = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} (-1)^{1+1}M_{11} & (-1)^{1+2}M_{12} \\ (-1)^{2+1}M_{21} & (-1)^{2+2}M_{22} \end{pmatrix} = \begin{pmatrix} a_{22} & -a_{21} \\ -a_{12} & a_{11} \end{pmatrix}$$

The adjugate matrix is:

$$\text{Adj}(A) = (\text{Cof}(A))^T = \begin{pmatrix} a_{22} & -a_{21} \\ -a_{12} & a_{11} \end{pmatrix}^T = \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}$$

The inverse is:

$$A^{-1} = \frac{1}{\det(A)} \cdot \text{Adj}(A) = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}$$

Example: Consider the matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$

Compute the determinant:

$$\det(A) = ad - bc = 11 \cdot 7 - 8 \cdot 3 = 77 - 24 = 53.$$

Since $\gcd(53, 26) = 1$, A is invertible modulo 26.

The adjugate of A is then:

$$\text{Adj}(A) = \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \begin{pmatrix} 7 & -8 \\ -3 & 11 \end{pmatrix}$$

And working in $\mathbb{Z}_{26}$, we have:

$$A^{-1} = \frac{1}{53} \cdot \begin{pmatrix} 7 & -8 \\ -3 & 11 \end{pmatrix} = 1 \cdot \begin{pmatrix} 7 & 17 \\ 23 & 11 \end{pmatrix} \mod 26.$$

Thus, in $\mathbb{Z}_{26}$, the inverse of A is:

$$A^{-1} = \begin{pmatrix} 7 & 17 \\ 23 & 11 \end{pmatrix}.$$

4 Permutation Cipher

Unlike previous ciphers, the permutation cipher does not substitute characters but only changes their positions.

Define:

$$\mathcal{K} = \text{Perm}(\{1, \ldots, m\}).$$

Encryption: Reorders the characters according to permutation π:

$$e_\pi(x) = (x_{\pi(1)}, \ldots, x_{\pi(m)}).$$

Decryption: Uses the inverse permutation:

$$d_\pi(y) = (y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(m)}).$$

Example: Consider the permutation $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 5 & 1 & 6 & 4 & 2 \end{pmatrix}$

This can also be written as:

x	1	2	3	4	5	6
$\pi(x)$	3	5	1	6	4	2

or in **cycle notation** as (1 3) (2 5 4 6).

Let P = 'HELLOTHERE'.

- The degree of the permutation is 6, and there are 10 plaintext characters.

- We want to split the plaintext evenly into blocks of size 6.

Thus, we pad the plaintext with two padding characters, #, giving:

$$P = \text{`HELLOTHERE\#\#'}$$

Applying the permutation to the indices, we obtain:

x	1	2	3	4	5	6	1	2	3	4	5	6
Plaintext	H	E	L	L	O	T	H	E	R	E	#	#
π_x	3	5	1	6	4	2	3	5	1	6	4	2
Ciphertext	L	O	H	T	L	E	R	#	H	#	E	E

Thus, the ciphertext is:

$$C = \text{`LOHTLER\#H\#EE'}$$

4.1 Permutation Matrix

The Permutation Cipher is a special case of the Hill Cipher.

We can model the permutation $\pi = \{1, \ldots, m\}$ as an $m \times m$ permutation matrix $K_\pi = (k_{i,j})$, such that:

$$k_{i,j} = \begin{cases} 1 & \text{if } i = \pi(j) \\ 0 & \text{otherwise} \end{cases}$$

For example, if $\pi = (1\ 3)(2\ 5\ 4\ 6)$, then:

$$K_\pi = K_{(1\ 3)(2\ 5\ 4\ 6)} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

For example, for $i = 5$, since $\pi(5) = 4$, we place a 1 at position $(\pi(i), i) = (\pi(5), 5) = (4, 5)$.

5 Stream Ciphers

The cryptosystems we have seen so far are block ciphers:

- Successive elements of plaintext are encrypted using the same key, K.
- The ciphertext string $\mathbf{y}$ is computed blockwise:

$$\mathbf{y} = y_1 y_2 \cdots = e_k(x_1) e_k(x_2) \ldots$$

Stream ciphers use a "keystream" instead:

- Generate a keystream $\mathbf{z} = z_1 z_2 \cdots$
- Use it to encrypt a plaintext string $\mathbf{x} = x_1 x_2 \cdots$
- To produce a ciphertext string $\mathbf{y} = y_1 y_2 \cdots = e_{z_1}(x_1) e_{z_2}(x_2) \cdots$

5.1 Synchronous Keystream

A synchronous keystream only depends on the key k, and not the plaintext $\mathbf{x}$.

Formally, a synchronous stream cipher includes in its description:

- $\mathcal{L}$, the keystream alphabet.
- The keystream generating function, $g\colon \mathcal{K} \to \mathcal{L}^{\mathbb{N}}$.
- g takes a key $K \in \mathcal{K}$ as input and generates an infinite string $z_1 z_2 \cdots$, where $z_i \in \mathcal{L}, \forall i \geq 1$.

5.2 The Vigenere Cipher

The Vigenère Cipher can be described as a synchronus stream cipher by defining:

$$\mathcal{K} = \mathbb{Z}_{26}^m, \quad \mathcal{P} = \mathcal{C} = \mathcal{L} = \mathbb{Z}_{26}$$

$$e_z(x) = (x + z) \bmod 26, \quad d_z(y) = (y - z) \bmod 26$$

$$z_i = \begin{cases} k_i & \text{if } 1 \leq i \leq m \\ z_{i-m} & \text{otherwise} \end{cases}$$

where $K = (k_1, \ldots, k_m)$.

This generates the keystream:

$$k_1 k_2 \cdots k_m k_1 k_2 \cdots k_m k_1 k_2 \cdots$$

from the key $K = (k_1, k_2, \ldots, k_m)$.

Ideally, we want a short key to generate a long keystream, and it should be unpredictable and seemingly random.

The Vigenère Cipher (which has keyword length m) is a periodic stream cipher with period m.

- This means the keystream repeats after only the first m elements.
- Since the period is only linear in m, it is a poor stream cipher.

5.3 Binary Stream Ciphers

Stream ciphers are often bitwise, such that $\mathcal{P} = \mathcal{C} = \mathcal{L} = \mathbb{Z}_2$. Encryption and decryption are simply addition modulo 2:

$$e_z(x) = (x + z) \bmod 2, \quad d_z(y) = (y + z) \bmod 2$$

Addition modulo 2 (bitwise addition) corresponds to the XOR operation ($\oplus$), which allows encryption and decryption to be implemented efficiently in hardware.

A common way of generating a synchronous keystream is by using a **linear recurrence**:

$$z_{i+m} = \sum_{j=0}^{m-1} c_j z_{i+j} \bmod 2, \quad \forall i > 0$$

where $c_0, \ldots, c_{m-1} \in \mathbb{Z}_2$ are given constants.

- This recurrence has **degree** m since each term depends on m previous terms.
- It is linear because z_{i+m} is a linear function of previous terms.
- The key K is defined by the $2m$ values: $k_1, \ldots, k_m$ and $c_0, \ldots, c_{m-1}$.

To maximize the keystream period, we choose c_i carefully so that the period is as large as $2^m - 1$.

- It is not 2^m because the keystream $(k_1, \ldots, k_m) = (0, 0, \ldots, 0)$ does not encrypt the plaintext and is never used.

Example: Let $m = 4$, and let the keystream be generated using the linear recurrence:

$$z_{i+4} = (z_i + z_{i+1}) \bmod 2, \quad \forall i \geq 1$$

and let the starting values be:

$$(z_1, z_2, z_3, z_4) = (k_1, k_2, k_3, k_4) = (1, 0, 0, 0).$$

Then, for $i = 1$ we have:

$$z_5 = (z_1 + z_2) \bmod 2 = 1 + 0 \bmod 2 = 1,$$

and for $i = 2$:

$$z_6 = (z_2 + z_3) \bmod 2 = 0 + 0 \bmod 2 = 0,$$

and so on.

Computing the rest of the values gives the keystream:

$$\mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{1}, 1, 0, 0, 0, \ldots$$

where the first $2^4 - 1 = 15$ elements make up the period, after which the keystream repeats.

5.4 Linear Feedback Shift Register (LFSR)

Keystream generation via bitwise arithmetic can be implemented very efficiently by encoding it as a hardware circuit (LFSR).

The LFSR is a shift register that contains m consecutive keystream elements (stages), and is initialized by the vector $(k_1, \ldots, k_m)$.

The register uses XOR addition with left-shifts (see Fig 2.2 in text).

Finally, there are non-synchronous ciphers in which the keystream depends on both the key, as well as previous plaintext or ciphertext elements. The autokey cipher is an example.