Introduction to Cryptography

Michael Clear

4 Feb 2015

Cryptography

- = "Secret writing" throughout most of its history.
  - More precisely, "writing" with a hidden meaning - as opposed to steganography where the existence of the "writing" itself is hidden.
- The idea is to make a message unintelligble except to the intended receiver.

- Up until the 1970's, the typical pattern was ([1])
  - Somebody creates a cipher.
  - They claim (or assume) the cipher is unbreakable.
  - Their enemy breaks the cipher using cryptanalysis.

Two Periods: BDH and ADH

- BDH - Before Diffie-Hellman < 1976
- ADH - After Diffie-Hellman > 1976

Two big changes in 1976

- Selection of Data Encryption Standard (DES) block cipher.
- Public-key cryptography - Diffie-Hellman.

BDH: Symmetric Cryptography

- A symmetric cipher uses the same key for encryption and decryption.
- Two main types:
  - Stream cipher.
  - Block cipher.
- Prior to 1970's, most ciphers were stream ciphers.
- A symmetric cipher consists of three algorithms - G, E and D:
  - G generates a secret key k.
  - E takes key k and plaintext m and ouputs a ciphertext c i.e. $c = E(k, m)$.
  - D takes a key k and a ciphertext c and ouputs a plaintext m i.e. $m = D(k, c)$.

Entropy

- Measure of uncertainty in a given source of information.

Entropy

- Measure of uncertainty in a given source of information.
- It measures the average amount of information contained in a given message, where a message is drawn from some distribution.

Entropy

- Measure of uncertainty in a given source of information.
- It measures the average amount of information contained in a given message, where a message is drawn from some distribution.
- Shannon's definition of entropy H:

Entropy

- Measure of uncertainty in a given source of information.
- It measures the average amount of information contained in a given message, where a message is drawn from some distribution.
- Shannon's definition of entropy H:
  - Let X be a discrete random variable.

Entropy

- Measure of uncertainty in a given source of information.
- It measures the average amount of information contained in a given message, where a message is drawn from some distribution.
- Shannon's definition of entropy H:
  - Let X be a discrete random variable.
  - Let $P(X)$ denote the probability mass function of X.

Entropy

- Measure of uncertainty in a given source of information.
- It measures the average amount of information contained in a given message, where a message is drawn from some distribution.
- Shannon's definition of entropy H:
  - Let X be a discrete random variable.
  - Let $P(X)$ denote the probability mass function of X.
  - Define H as $H(X) = -\sum_i P(x_i) \log_2 P(x_i)$.

Entropy

- Measure of uncertainty in a given source of information.
- It measures the average amount of information contained in a given message, where a message is drawn from some distribution.
- Shannon's definition of entropy H:
  - Let X be a discrete random variable.
  - Let $P(X)$ denote the probability mass function of X.
  - Define H as $H(X) = -\sum_i P(x_i) \log_2 P(x_i)$.
- H gives the average number of bits of information contained in some message, which we call the amount of *entropy*.

Unicity Distance

- Minimum length of ciphertext needed for a computationally unbounded adversary to recover the (unique) encryption key.

Unicity Distance

- Minimum length of ciphertext needed for a computationally unbounded adversary to recover the (unique) encryption key.
- In a brute force attack where every key is tried, there should be just one key that decrypts to a sensible plaintext.

Unicity Distance

- Minimum length of ciphertext needed for a computationally unbounded adversary to recover the (unique) encryption key.
- In a brute force attack where every key is tried, there should be just one key that decrypts to a sensible plaintext.
- The keys other than the correct key that yield a sensible plaintext on decryption are called *spurious keys*.

Unicity Distance

- Minimum length of ciphertext needed for a computationally unbounded adversary to recover the (unique) encryption key.
- In a brute force attack where every key is tried, there should be just one key that decrypts to a sensible plaintext.
- The keys other than the correct key that yield a sensible plaintext on decryption are called *spurious keys*.
- The unicity distance is the needed length of ciphertext for the number of spurious keys to be zero.

Unicity Distance

- Minimum length of ciphertext needed for a computationally unbounded adversary to recover the (unique) encryption key.
- In a brute force attack where every key is tried, there should be just one key that decrypts to a sensible plaintext.
- The keys other than the correct key that yield a sensible plaintext on decryption are called *spurious keys*.
- The unicity distance is the needed length of ciphertext for the number of spurious keys to be zero.
- We calculate the unicity distance as:

Unicity Distance

- Minimum length of ciphertext needed for a computationally unbounded adversary to recover the (unique) encryption key.
- In a brute force attack where every key is tried, there should be just one key that decrypts to a sensible plaintext.
- The keys other than the correct key that yield a sensible plaintext on decryption are called *spurious keys*.
- The unicity distance is the needed length of ciphertext for the number of spurious keys to be zero.
- We calculate the unicity distance as:
  - Let k be the number of keys.

Unicity Distance

- Minimum length of ciphertext needed for a computationally unbounded adversary to recover the (unique) encryption key.
- In a brute force attack where every key is tried, there should be just one key that decrypts to a sensible plaintext.
- The keys other than the correct key that yield a sensible plaintext on decryption are called *spurious keys*.
- The unicity distance is the needed length of ciphertext for the number of spurious keys to be zero.
- We calculate the unicity distance as:
  - Let k be the number of keys.
  - Let s be the number of "sensible" (readable) plaintexts.

Unicity Distance

- Minimum length of ciphertext needed for a computationally unbounded adversary to recover the (unique) encryption key.
- In a brute force attack where every key is tried, there should be just one key that decrypts to a sensible plaintext.
- The keys other than the correct key that yield a sensible plaintext on decryption are called *spurious keys*.
- The unicity distance is the needed length of ciphertext for the number of spurious keys to be zero.
- We calculate the unicity distance as:
  - Let k be the number of keys.
  - Let s be the number of "sensible" (readable) plaintexts.
  - Let p be the total number of possible plaintexts.

Unicity Distance

- Minimum length of ciphertext needed for a computationally unbounded adversary to recover the (unique) encryption key.
- In a brute force attack where every key is tried, there should be just one key that decrypts to a sensible plaintext.
- The keys other than the correct key that yield a sensible plaintext on decryption are called *spurious keys*.
- The unicity distance is the needed length of ciphertext for the number of spurious keys to be zero.
- We calculate the unicity distance as:
  - Let k be the number of keys.
  - Let s be the number of "sensible" (readable) plaintexts.
  - Let p be the total number of possible plaintexts.
  - Then the unicity distance is the length of ciphertext U such that $k \cdot \frac{s}{p} = 1$.

Unicity Distance (Cont'd)

- We can alternatively define the unicity distance:

Unicity Distance (Cont'd)

- We can alternatively define the unicity distance:
  - Let K be the key space. The entropy of K is $H(K)$.

Unicity Distance (Cont'd)

- We can alternatively define the unicity distance:
  - Let K be the key space. The entropy of K is $H(K)$.
  - Let $H(M)$ be the entropy of a given "sensible" (e.g: English language) plaintext character.

Unicity Distance (Cont'd)

- We can alternatively define the unicity distance:
  - Let K be the key space. The entropy of K is $H(K)$.
  - Let $H(M)$ be the entropy of a given "sensible" (e.g: English language) plaintext character.
  - Let n be the number of bits per plaintext character.

Unicity Distance (Cont'd)

- We can alternatively define the unicity distance:
  - Let K be the key space. The entropy of K is $H(K)$.
  - Let $H(M)$ be the entropy of a given "sensible" (e.g: English language) plaintext character.
  - Let n be the number of bits per plaintext character.
  - Then we have $U = H(K)/(n - H(M))$.

Unicity Distance (Cont'd)

- We can alternatively define the unicity distance:
  - Let K be the key space. The entropy of K is $H(K)$.
  - Let $H(M)$ be the entropy of a given "sensible" (e.g: English language) plaintext character.
  - Let n be the number of bits per plaintext character.
  - Then we have $U = H(K)/(n - H(M))$.
  - The value $D = n - H(M)$ is the redundancy of the plaintext.

Perfect Secrecy

- A cipher is perfectly secure if the entropy of the plaintext *given* the ciphertext is equivalent to the entropy of the plaintext.

Perfect Secrecy

- A cipher is perfectly secure if the entropy of the plaintext *given* the ciphertext is equivalent to the entropy of the plaintext.
  - Let $H(M)$ be the entropy of the plaintext.

Perfect Secrecy

- A cipher is perfectly secure if the entropy of the plaintext *given* the ciphertext is equivalent to the entropy of the plaintext.
  - Let $H(M)$ be the entropy of the plaintext.
  - Let C be the probability distribution of the ciphertext.

Perfect Secrecy

- A cipher is perfectly secure if the entropy of the plaintext *given* the ciphertext is equivalent to the entropy of the plaintext.
  - Let $H(M)$ be the entropy of the plaintext.
  - Let C be the probability distribution of the ciphertext.
  - Then
$$H(M) = H(M \mid C)$$
    where $H(M|C)$ the *conditional entropy* of M given a ciphertext from C.

Perfect Secrecy

- A cipher is perfectly secure if the entropy of the plaintext *given* the ciphertext is equivalent to the entropy of the plaintext.
  - Let $H(M)$ be the entropy of the plaintext.
  - Let C be the probability distribution of the ciphertext.
  - Then
    $$H(M) = H(M \mid C)$$
    where $H(M|C)$ the *conditional entropy* of M given a ciphertext from C.
  - Put another way, for all plaintexts $m \in M$ and all ciphertexts $c \in C$, we have
    $$\Pr(x) = \Pr(m \mid c)$$

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.
- A stream cipher in which the key is the same length as the plaintext, and is chosen to be truly random.

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.
- A stream cipher in which the key is the same length as the plaintext, and is chosen to be truly random.
- The random key is called a pad.

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.
- A stream cipher in which the key is the same length as the plaintext, and is chosen to be truly random.
- The random key is called a pad.
- The cipher is unbreakable as long as the key is

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.
- A stream cipher in which the key is the same length as the plaintext, and is chosen to be truly random.
- The random key is called a pad.
- The cipher is unbreakable as long as the key is
  - truly random

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.
- A stream cipher in which the key is the same length as the plaintext, and is chosen to be truly random.
- The random key is called a pad.
- The cipher is unbreakable as long as the key is
  - truly random
  - kept secret

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.
- A stream cipher in which the key is the same length as the plaintext, and is chosen to be truly random.
- The random key is called a pad.
- The cipher is unbreakable as long as the key is
  - truly random
  - kept secret
  - used only once

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.
- A stream cipher in which the key is the same length as the plaintext, and is chosen to be truly random.
- The random key is called a pad.
- The cipher is unbreakable as long as the key is
  - truly random
  - kept secret
  - used only once
  - the same length as the plaintext.

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.
- A stream cipher in which the key is the same length as the plaintext, and is chosen to be truly random.
- The random key is called a pad.
- The cipher is unbreakable as long as the key is
  - truly random
  - kept secret
  - used only once
  - the same length as the plaintext.
- The one-time pad has perfect secrecy.

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.
- A stream cipher in which the key is the same length as the plaintext, and is chosen to be truly random.
- The random key is called a pad.
- The cipher is unbreakable as long as the key is
  - truly random
  - kept secret
  - used only once
  - the same length as the plaintext.
- The one-time pad has perfect secrecy.
- The n-th character is enciphered by "adding" the n-the character of the key to the n-th character of the plaintext.

One-time Pad

- Described by Gilbert Vernam in 1917. Known as the Vernam Cipher.
- A stream cipher in which the key is the same length as the plaintext, and is chosen to be truly random.
- The random key is called a pad.
- The cipher is unbreakable as long as the key is
  - truly random
  - kept secret
  - used only once
  - the same length as the plaintext.
- The one-time pad has perfect secrecy.
- The n-th character is enciphered by "adding" the n-the character of the key to the n-th character of the plaintext.
- Example: using addition modulo 2 (XOR) when encrypting a binary message.

Confusion and Diffusion

- Encryption is based on the principles of confusion and diffusion.

Confusion and Diffusion

- Encryption is based on the principles of confusion and diffusion.
- **Confusion:** Making the relationship between the ciphertext and the key complex. Each ciphertext character should depend on several different parts of the key.

Confusion and Diffusion

- Encryption is based on the principles of confusion and diffusion.
- **Confusion:** Making the relationship between the ciphertext and the key complex. Each ciphertext character should depend on several different parts of the key.
  - Means that drastic changes are made from the input to the output.

Confusion and Diffusion

- Encryption is based on the principles of confusion and diffusion.
- **Confusion:** Making the relationship between the ciphertext and the key complex. Each ciphertext character should depend on several different parts of the key.
  - Means that drastic changes are made from the input to the output.
  - Can be achieved via the technique of substitution.

Confusion and Diffusion

- Encryption is based on the principles of confusion and diffusion.
- **Confusion:** Making the relationship between the ciphertext and the key complex. Each ciphertext character should depend on several different parts of the key.
    - Means that drastic changes are made from the input to the output.
    - Can be achieved via the technique of substitution.
- **Diffusion:** Changing a single character of the plaintext changes many characters of the ciphertext.

Confusion and Diffusion

- Encryption is based on the principles of confusion and diffusion.
- **Confusion:** Making the relationship between the ciphertext and the key complex. Each ciphertext character should depend on several different parts of the key.
  - Means that drastic changes are made from the input to the output.
  - Can be achieved via the technique of substitution.
- **Diffusion:** Changing a single character of the plaintext changes many characters of the ciphertext.
  - Distributing the statistical structure of the plaintext across much larger structures in the ciphertext.

Confusion and Diffusion

- Encryption is based on the principles of confusion and diffusion.
- **Confusion:** Making the relationship between the ciphertext and the key complex. Each ciphertext character should depend on several different parts of the key.
  - Means that drastic changes are made from the input to the output.
  - Can be achieved via the technique of substitution.
- **Diffusion:** Changing a single character of the plaintext changes many characters of the ciphertext.
  - Distributing the statistical structure of the plaintext across much larger structures in the ciphertext.
  - Can be achieved via the technique of permutation (aka transposition).

Substitution

- Replace each character of the plaintext with a potentially different character determined by the key.

Substitution

- Replace each character of the plaintext with a potentially different character determined by the key.
- In a *monoalphabetic* substitution cipher, the key decides the particular susbtitution table that is used.

Substitution

- Replace each character of the plaintext with a potentially different character determined by the key.
- In a *monoalphabetic* substitution cipher, the key decides the particular susbtitution table that is used.
- Example - Caesar Cipher: each character is shifted by k places in the alphabet where k is the key.

Substitution

- Replace each character of the plaintext with a potentially different character determined by the key.
- In a *monoalphabetic* substitution cipher, the key decides the particular susbtitution table that is used.
- Example - Caesar Cipher: each character is shifted by k places in the alphabet where k is the key.
  - For $k = 3$, the string "HELLO" is encrypted as "KHOOR".

Substitution

- Replace each character of the plaintext with a potentially different character determined by the key.
- In a *monoalphabetic* substitution cipher, the key decides the particular susbtitution table that is used.
- Example - Caesar Cipher: each character is shifted by k places in the alphabet where k is the key.
  - For $k = 3$, the string "HELLO" is encrypted as "KHOOR".
  - Encryption of m is $c = m + k \mod 26$.

Substitution

- Replace each character of the plaintext with a potentially different character determined by the key.
- In a *monoalphabetic* substitution cipher, the key decides the particular susbtitution table that is used.
- Example - Caesar Cipher: each character is shifted by k places in the alphabet where k is the key.
  - For $k = 3$, the string "HELLO" is encrypted as "KHOOR".
  - Encryption of m is $c = m + k \mod 26$.
- A monoalphabetic substitution cipher can have $P!$ different keys where P is the number of plaintext characters.

Substitution

- Replace each character of the plaintext with a potentially different character determined by the key.
- In a *monoalphabetic* substitution cipher, the key decides the particular susbtitution table that is used.
- Example - Caesar Cipher: each character is shifted by k places in the alphabet where k is the key.
  - For $k = 3$, the string "HELLO" is encrypted as "KHOOR".
  - Encryption of m is $c = m + k \mod 26$.
- A monoalphabetic substitution cipher can have $P!$ different keys where P is the number of plaintext characters.
  - So English language text with 26 characters results in $26! = 403291461126605635584000000$ possible keys.

Substitution

- Replace each character of the plaintext with a potentially different character determined by the key.
- In a *monoalphabetic* substitution cipher, the key decides the particular susbtitution table that is used.
- Example - Caesar Cipher: each character is shifted by k places in the alphabet where k is the key.
  - For $k = 3$, the string "HELLO" is encrypted as "KHOOR".
  - Encryption of m is $c = m + k \mod 26$.
- A monoalphabetic substitution cipher can have $P!$ different keys where P is the number of plaintext characters.
  - So English language text with 26 characters results in $26! = 403291461126605635584000000$ possible keys.
  - The entropy per character of English is $H(M) \approx 1.5$.

Substitution

- Replace each character of the plaintext with a potentially different character determined by the key.
- In a *monoalphabetic* substitution cipher, the key decides the particular susbtitution table that is used.
- Example - Caesar Cipher: each character is shifted by k places in the alphabet where k is the key.
  - For $k = 3$, the string "HELLO" is encrypted as "KHOOR".
  - Encryption of m is $c = m + k \mod 26$.
- A monoalphabetic substitution cipher can have $P!$ different keys where P is the number of plaintext characters.
  - So English language text with 26 characters results in $26! = 403291461126605635584000000$ possible keys.
  - The entropy per character of English is $H(M) \approx 1.5$.
  - The unicity distance is $U = H(K)/(n - H(M)) = \log_2 26!/(\log_2 26 - 1.5) \approx 28$.

Frequency Analysis

- A monoalphabetic substitution cipher can be defeated with frequency analysis.

Frequency Analysis

- A monoalphabetic substitution cipher can be defeated with frequency analysis.
- Invented by Al-Kindi in the 9th century.

Frequency Analysis

- A monoalphabetic substitution cipher can be defeated with frequency analysis.
- Invented by Al-Kindi in the 9th century.
- Involves counting the number of occurences of each ciphertext character and matching against the frequency distribution of plaintext characters.

Frequency Analysis

- A monoalphabetic substitution cipher can be defeated with frequency analysis.
- Invented by Al-Kindi in the 9th century.
- Involves counting the number of occurences of each ciphertext character and matching against the frequency distribution of plaintext characters.
- In English, the most frequently occuring letters are (in order) E, T, A, O, I, N, S, H, R, D, L, U...

Frequency Analysis

- A monoalphabetic substitution cipher can be defeated with frequency analysis.
- Invented by Al-Kindi in the 9th century.
- Involves counting the number of occurences of each ciphertext character and matching against the frequency distribution of plaintext characters.
- In English, the most frequently occuring letters are (in order) E, T, A, O, I, N, S, H, R, D, L, U. . .
- So given a ciphertext generated from an English plaintext, the most frequently occuring character likely corresponds to E etc.

Polyalphabetic Substitution Cipher

- Uses multiple substitution alphabets.

Polyalphabetic Substitution Cipher

- Uses multiple substitution alphabets.
- The main idea is to change the substitution alphabet with each plaintext character, so the first letter is encrypted according to one alphabet, the second according to a different alphabet and so on (note the alphabets may repeat after a certain period).

Vigenère Cipher

- Example of a polyalphabetic substitution cipher.

Vigenère Cipher

- Example of a polyalphabetic substitution cipher.
- Works as follows:

Vigenère Cipher

- Example of a polyalphabetic substitution cipher.
- Works as follows:
  - Let k be a keyword such as "BLAZE", with $n = 5$ letters.

Vigenère Cipher

- Example of a polyalphabetic substitution cipher.
- Works as follows:
  - Let k be a keyword such as "BLAZE", with $n = 5$ letters.
  - Let k_i denote the numeric value (modulo 26) of the i-th letter of the keyword k.

Vigenère Cipher

- Example of a polyalphabetic substitution cipher.
- Works as follows:
  - Let k be a keyword such as "BLAZE", with $n = 5$ letters.
  - Let k_i denote the numeric value (modulo 26) of the i-th letter of the keyword k.
  - The i-th letter of plaintext m_i is encrypted as $c_i = m_i + k_{i \pmod{\text{lengthofthekey}}} \pmod{26}$ (when represented modulo 26)

Vigenère Cipher

- Example of a polyalphabetic substitution cipher.
- Works as follows:
  - Let k be a keyword such as "BLAZE", with $n = 5$ letters.
  - Let k_i denote the numeric value (modulo 26) of the i-th letter of the keyword k.
  - The i-th letter of plaintext m_i is encrypted as $c_i = m_i + k_{i \pmod{lengthofthekey}} \pmod{26}$ (when represented modulo 26)
- Encrypting the text "HELLO" with keyword "BLAZE" yields the ciphertext "IPLKS".

Viginère Cipher - Tabula Recta

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Source: Wikipedia

Transposition

- Idea: rearrange the plaintext (change the order) to produce the ciphertext.
- The positions of the plaintext characters are shifted.
- Also known as *permutation*.
- Examples:
  - Rail Fence
  - Route Cipher
  - Columnar Transposition

Example: Columnar Transposition

- Write plaintext along rows whose length is determined by the key
- Example (here X denotes a null character):

T	H	E	R	E
M	U	S	T	B
E	S	O	M	E
K	I	N	D	O
F	W	A	Y	O
U	T	O	F	H
E	R	E	X	X

- Suppose the key specifies the row length as 5 and the order of columns to write out as 4, 2, 5, 1, 3.
- Then we get the ciphertext by writing out the columns in the specified order:
  - we obtain: RTMDYFXHUSTWTREBEOOHXTMEKFUEESONAOE

Example: Columnar Transposition (Cont'd)

- The key could be alternatively given as a keyword such as TOWER
  - the length of the keyword represents the row length.
  - the alphabetical order of the letters in the keyword gives the order of the columns to be written out.

References

Barak, B.:
Cos 433: Cryptography.
www.cs.princeton.edu/courses/archive/fall07/cos433/.../lec1-intro.ppt (2007)