

CS458: Introduction to Information Security

Notes 2: Historical Crypto

Yousef M. Elmehdwi

Department of Computer Science

Illinois Institute of Technology

yelmehdwi@iit.edu

January 17th 2024

Slides: Adopted from Ewa Syta, [Yale University](#), Computer Security: Principles and Practice, 4th Edition & Ian Goldberg, Florian Kerschbaum [University of Waterloo](#)

- Cryptography
- Cryptanalysis
- Classical Cryptography
 - Substitution Ciphers
 - Transposition Ciphers

The basic terminology of crypto includes the following:

- **Cryptography**: (“secret writing”): Making secret messages
 - The practice of creating secret messages by transforming **plaintext** (an ordinary readable message) into **ciphertext** (secret messages that are difficult to interpret).
 - i.e., the science of secret writing, aiming to conceal the meaning of a message.
- **Cryptanalysis**: Breaking secret messages
 - The process of deciphering secret messages, involving the recovery of plaintext from ciphertext without possessing the key.
 - Also called code breaking
- **Cryptology**: The combined art and science of both creating and deciphering “secret codes”.
 - It encompasses cryptography and cryptanalysis, representing the broader field of secure communication techniques.
- **Crypto**: A term that serves as a synonym for any or all of the aforementioned concepts, with the precise meaning determined by the context in which it is used.
- The point of cryptography is to send secure messages over an insecure medium (like the Internet)

Characterizing Cryptographic Systems

- Cryptographic systems are generally classified along three independent dimensions:
 - **Type of operations used for encryption**
 - **Substitution**: each element in the plaintext is mapped into another element
 - **Transposition**: elements in plaintext are rearranged
 - **Product systems/ciphers**: multiple stages/combinations of substitutions and transpositions
 - **Number of keys used**
 - **Symmetric**: sender and receiver use same key
 - **Asymmetric**: sender and receiver each use a different key
 - **Processing of plaintext**
 - **Block cipher**: Operates on input data in fixed-size blocks. Processes each block independently, generating an output block.
 - **Stream cipher**: process the input elements continuously, producing output element one at a time, as it goes along.
- **Key**
 - The key, which is known only to the sender and receiver, is a critical component of the encryption/decryption processes.

Terminology

The basic terminology of crypto includes the following:

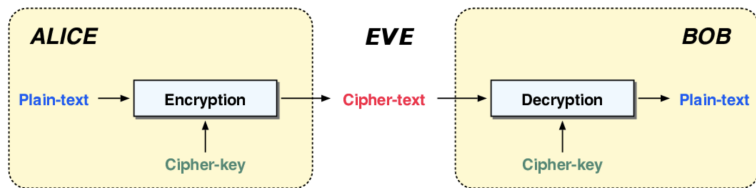
- **Cipher**: a particular algorithm (cryptographic system)
- **Plaintext**: original message
- **Ciphertext**: encrypted or coded message
- **Encryption**: convert from plaintext to ciphertext (enciphering)
- **Decryption**: restore the plaintext from ciphertext (deciphering)
- **Key**: critical information used in cipher known only to sender/receiver
- **Symmetric key** cryptosystem uses the same key to encrypt as to decrypt
- **Public/Asymmetric key** cryptosystem uses a **public key** to encrypt and a **private key** to decrypt

Secret Message Transmission Problem

- Alice wants to send Bob a private message \mathbf{m} over the internet.
- Eve is an eavesdropper who listens in and wants to learn \mathbf{m} .
- Alice and Bob want \mathbf{m} to remain private and unknown to Eve.

Solution using encryption¹

- A **symmetric cryptosystem** (sometimes called a private-key or one-key system) is a pair of efficiently-computable functions E and D such that
 - $E(k, m)$ **encrypts** plaintext message m using key k to produce a ciphertext c .
 - $D(k, c)$ **decrypts** ciphertext c using k to produce a message m .
- **Requirements:**
 - **Correctness** $D(k, E(k, m)) = m$ for all keys k and all messages m .
 - **Security** Given $c = E(k, m)$, it is hard to find m without knowing k



¹ image credit: Derived from https://iis-people.ee.ethz.ch/~kgf/acacia/fig/alice_bob.png

Symmetric Cryptosystem Components

- Plaintext (m): original message
- Ciphertext (c): encrypted message
- Key (k): private information
- Encryption algorithm: $c = E(k, m) = E_k(m)$
- Decryption algorithm: $m = D(k, c) = D_k(c)$

The protocol

- **Protocol:** Composition and ordering of the exchanged messages
- **Protocol**
 1. Alice and Bob share a common secret key k .
 2. Alice computes $c = E_k(m)$ and sends c to Bob.
 3. Bob receives c' . Computes $m' = D_k(c')$, and assumes m' to be Alice's message.
- **Assumptions**
 - Eve learns nothing except for c during the protocol.
 - The channel is perfect, so $c' = c$.
 - Eve is a **passive eavesdropper** who can read c but not modify it.

Requirements

- What do we require of E , D , and the computing environment?
 - Given c , it is hard to find m without also knowing k .
 - k is not initially known to Eve.
 - Eve can guess k with at most negligible success probability.
 - k must be chosen randomly from a large key space.
 - Alice and Bob successfully keep k secret.
 - Their computers have not been compromised; Eve can't find k on their computers even if she is a legitimate user, etc.
 - Eve can't obtain k in other ways, e.g., by social engineering, using binoculars to watch Alice or Bob's keyboard, etc.

Eve's side of the story ²

I'M SURE YOU'VE HEARD ALL ABOUT THIS SORDID AFFAIR IN THOSE GOSSIPY CRYPTOGRAPHIC PROTOCOL SPECS WITH THOSE BUSYBODIES SCHNEIER AND RIVEST, ALWAYS TAKING ALICE'S SIDE, ALWAYS LABELING ME THE ATTACKER.



YES, IT'S TRUE. I BROKE BOB'S PRIVATE KEY AND EXTRACTED THE TEXT OF HER MESSAGES. BUT DOES ANYONE REALIZE HOW MUCH IT HURT?



HE SAID IT WAS NOTHING, BUT EVERYTHING FROM THE PUBLIC-KEY AUTHENTICATED SIGNATURES ON THE FILES TO THE LIPSTICK HEART SMEARED ON THE DISK SCREAMED "ALICE."



I DIDN'T WANT TO BELIEVE. OF COURSE ON SOME LEVEL I REALIZED IT WAS A KNOWN-PLAINTEXT ATTACK. BUT I COULDN'T ADMIT IT UNTIL I SAW FOR MYSELF.



SO BEFORE YOU SO QUICKLY LABEL ME A THIRD PARTY TO THE COMMUNICATION, JUST REMEMBER: I LOVED HIM FIRST. WE HAD SOMETHING AND SHE TORE IT AWAY. SHE'S THE ATTACKER, NOT ME. NOT EVE.



²image credit: <https://xkcd.com/177/>

- Basic assumptions
 - The system is completely known to the attacker
 - Only the key is secret
 - That is, crypto algorithms are not secret
- This is known as **Kerckhoffs' Principle**
 - i.e., one should always assume that the adversary knows the encryption/decryption algorithms and the resistance of the cipher to attacks must be based on only the secrecy of the key.
 - This has a number of implications
 - System security is limited by the number of possible keys.
 - Attackers can just attempt all keys to find the correct one.
 - A **strong cryptosystem** is one where that's the best Attackers can do.
 - with weaker systems, there are shortcuts to finding the key.
- Why do we make such an assumption?
 - History shows secret algorithms often have weaknesses.
 - Secret algorithms never remain secret
 - Public exposure helps identify weaknesses early (better to find weaknesses beforehand)

Eve's goals

- Eve wants learn *something*. Eve is not bound by any rules. She can do as she wishes with the information she has available.
- We don't want her to be able to:
 - Recover the key.
 - Find the plaintext to a ciphertext.
 - Determine any character to the plaintext.
 - Derive any meaningful information about the plaintext.

A CRYPTO NERD'S IMAGINATION:

HIS LAPTOP'S ENCRYPTED.
LET'S BUILD A MILLION-DOLLAR
CLUSTER TO CRACK IT.

NO GOOD! IT'S
4096-BIT RSA!

BLAST! OUR
EVIL PLAN
IS FOILED!



WHAT WOULD ACTUALLY HAPPEN:

HIS LAPTOP'S ENCRYPTED.
DRUG HIM AND HIT HIM WITH
THIS \$5 WRENCH UNTIL
HE TELLS US THE PASSWORD.

GOT IT.



Eve's information

- We assumed that Eve has no information about the cryptosystem except for the encryption/decryption methods and the ciphertext c .
- In practice, Eve might know much more.
 - She probably knows (or has a good idea) of the message distribution¹.
 - She might have obtained several other ciphertexts.
 - She might have learned the decryptions of earlier ciphertexts.
 - She might have even chosen the earlier messages or ciphertexts herself
- This leads us to consider several attack scenarios.

¹ *Message distribution refers to the probability of different messages appearing in a communication system.*

Attack scenarios

- **Cryptanalysis**: Process of attempting to discover the plaintext or key.
- **Ciphertext-only attack**
 - Eve knows only the ciphertext to be decoded c and tries to recover m .
- **Known plaintext attack**
 - Eve knows the ciphertext to be decoded c and a sequence of plaintext-ciphertext pairs $(m_1, c_1), \dots, (m_r, c_r)$ where $c \notin \{c_1, \dots, c_r\}$.
 - She tries to recover m .

Known plaintext attacks

- A **known plaintext attack** can occur when
 1. Alice uses the same key to encrypt several messages;
 2. Eve later learns or successfully guesses the corresponding plaintexts.
- Some ways that Eve learns plaintexts.
 - The plaintext might be publicly revealed at a later time, e.g., sealed bid auctions.
 - The plaintext might be guessable, e.g., an email header.
 - Eve might later discover the decrypted message on Bob's computer.

Chosen text attack scenarios

- Still stronger attack scenarios allow Eve to choose one element of a plaintext-ciphertext pair and obtain the other
- Chosen plaintext attack
 - Like a known plaintext attack, except that Eve chooses messages m_1, \dots, m_r before getting c and Alice (or Bob) encrypts them for her.
- Chosen ciphertext attack
 - Like a known plaintext attack, except that Eve chooses ciphertexts c_1, \dots, c_r before getting c and Alice (or Bob) decrypts them for her.
- Mixed chosen plaintext/chosen ciphertext attack
 - Eve chooses some plaintexts and some ciphertexts and gets the corresponding decryptions or encryptions.

Why would Alice cooperate in a chosen plaintext attack?

- Eve might be authorized to generate messages that are then encrypted and sent to Bob, but she isn't authorized to read other people's messages.¹
- Alice might be an internet server, not a person, that encrypts messages received in the course of carrying out a more complicated cryptographic protocol.
- Eve might gain access to Alice's computer, perhaps only for a short time, when Alice steps away from her desk.

¹Nothing we have said implies that Eve is unknown to Alice and Bob or that she is not also a legitimate participant in the protocol.

Adaptive chosen text attack scenarios

- Adaptive versions of chosen text protocols are when Eve chooses her texts one at a time after learning the response to her previous text
- Adaptive chosen plaintext attack
 - Eve chooses the messages m_1, m_2, \dots one at a time rather than all at once.
 - Thus,
 - m_2 depends on (m_1, c_1)
 - m_3 depends on both (m_1, c_1) and (m_2, c_2) , etc.
- Adaptive chosen ciphertext and adaptive mixed attacks
 - are defined similarly

Computationally Secure Encryption Schemes

- Only relatively weak algorithms fail to withstand a **ciphertext-only attack**.
- Some ciphers are easy to crack, while others are extremely challenging.
- In practice, **perfect security** is impossible to achieve¹.
 - *Perfect security, or information-theoretic security, ensures that encrypted data reveals no hints about the original message, regardless of the attacker's computational capabilities.*
- Instead, most ciphers in use today rely on **computational security**.
 - This means while the encryption doesn't guarantee perfect security, it's designed to be secure against attacks using currently known computational methods and technologies.
 - The underlying assumption is that breaking the encryption within a practical timeframe using current computing resources is not feasible.
 - *These ciphers essentially rely on the fact that there is no computer system powerful enough to crack them in a reasonable amount of time.*

¹ The one-time pad (OTP) is a unique encryption technique that offers perfect security in theory. However, its practical implementation imposes significant limitations, making it less feasible for widespread use in modern cryptography.

Computationally Secure Encryption Schemes

- *Computationally secure means that a system or algorithm is secure as long as an attacker does not have enough computational resources or time to break it. However, it's worth noting that even computationally secure systems may still be vulnerable to attacks that exploit implementation errors or other weaknesses.*

Attacks

- Goal of the Attacker

- Discover the plaintext (good)
- Discover the key (better)

- Assumed Attacker Knowledge

- Ciphertext
- Algorithm
- Other pairs of (plaintext, ciphertext) using same key

- Attack Methods

- Brute-force attack

- Try every possible key on ciphertext until readable text is obtained from the ciphertext
- On average, number of guesses is half the key space

- Cryptanalysis

- Use knowledge of algorithm and/or plaintext patterns to “intelligently” decipher the ciphertext
- Exploit characteristics of algorithm to deduce plaintext or key
- Attacks differ based on amount of information known to attacker

- Assumption: attacker can recognize correct plaintext

- *Cryptanalyst or attacker tries to break the system*

Exhaustive Key Search

- Exhaustive key search

- Eve can simply try all possible keys and test each to see if it is correct.
 - Remember, she has some ciphertexts so she knows when she found the right key.
- To prevent an exhaustive key search, a cryptosystem must have a large **keyspace**.
 - The set of all possible keys that can be used to generate a key.
 - Must be too many keys for Eve to try them all in any reasonable amount of time.

Beyond Exhaustive Search

- A large key space is **necessary** for security.
- But a large key space is **not sufficient**.
 - Shortcut attacks might exist.
 - In cryptography, we can (almost) never prove that no shortcut attack exists

- For now, we will focus on classical crypto.
- All classical ciphers are symmetric.

Substitution Ciphers

- Probably the most common form of cipher.
- They work by replacing each letter of the plaintext with another letter (or possibly even a random symbol).
- The most famous one comes after the [Caesar cipher](#).

Caesar Cipher/shift

- **Idea:** substitute one letter for another one.
- Julius Caesar shifts each letter by three positions in the alphabet:
- Plaintext: **fourscoreandsevenyearsago**
- Key: how we substitute

Plain	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
Cipher	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

- Ciphertext: **IRXUVFRUHDQGVHYHQBHDUVDJR**
- *Replace each letter with one 3 letters later in the alphabet.*
- The replacement remains the same throughout the message.
- The cipher is classified as a type of **Mono-alphabetic substitution** (fixed replacement structure)
 - Same letter is replaced with only one other (always the same for given cipher message).

Caesar's Cipher Decryption

- Caesar code decryption

- Deciphering is done in reverse, with a left shift of 3.
- Replaces a letter another with an inverse alphabet shift: a previous letter in the alphabet.

- Caesar's cipher using a right shift of three places

Plain	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
Cipher	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

- Given ciphertext:

JHW D YDFFLQH, ZH DU D PDVN, VRFLDL SKBVLFD O
GLVWDQFH

- Plaintext:

get a vaccine, wear a mask, social distance

Shift Cipher

- We can shift by any number of positions:
 - shift by k for some $k \in \{0, 1, 2, \dots, 25\}$
- Then, key is k
- Example: key $k = 7$.

Plain	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
Cipher	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

Shift Cipher Encryption/Decryption

- Another way to encrypt/decrypt, more mathematical
 - Let, M: plaintext; K: key; E: encryption function; D: decryption function
 - $M = \{\text{sequences of letters}\}$
 - The encryption can also be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme,
 - $A \rightarrow 0, B \rightarrow 1, \dots, Z \rightarrow 25$
 - Encryption of a letter m by a shift k can be described mathematically as
 - $K = \{k \mid k \text{ is an integer and } 0 \leq k \leq 25\}$
 - $E = \{E \mid k \in K, E_k(m) = (m+k) \bmod 26\}$
 - Decryption of a letter c by a shift k can be described mathematically as
 - $D = \{D \mid k \in K, D_k(c) = (26+c-k) \bmod 26\}$

Breaking the Shift Cipher

- **Cryptanalysis** attempts to discover the key or the plaintext of an encrypted message
- Imagine you have the ciphertext. How to find the key?
- A simple substitution (shift by k) is used.
 - But the key is unknown
- Given ciphertext: **CSYEVIXIVQMREXIH**
- **Exhaustive key search**
 - If the key space is small enough, try them all approach.
 - Only 26 possible keys.
 - Solution:
 - key is $k = 4$
 - Plaintext: **youareterminated**

Coding assignment I

- The objective of this assignment is to implement a simple encryption and decryption program in any programming language.
- Due in Blackboard Assignment by Thursday, February 1st at 11:59pm

Permutation Cipher

- A **permutation cipher** is a type of simple substitution cipher in which the mapping of plaintext letters to ciphertext letters is an arbitrary permutation of the alphabet.
- The secret key for this cipher is a randomly chosen permutation of the numbers from 0 to 25, which correspond to the letters of the alphabet.
- How many keys are possible?
- For example:

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

- Then $26 \times 25 \times 24 \times 23 \dots \times 3 \times 2 \times 1 = 26! = 403291461126605635584000000$ possible keys!
 - *It would likely take millions of years to try every possible key*
- *Since the permutation key is a mapping of all 26 letters of the alphabet, there are $26!$ (26 factorial) possible permutations.*

Cryptanalysis: Be Clever

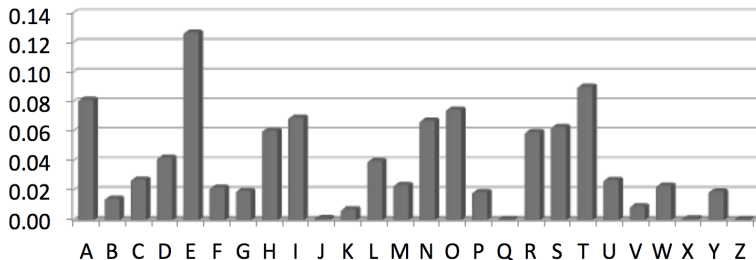
- Cannot try all $26!$ simple substitution keys. (would not be practical)
- Can we be more clever?
- But knowledge of language statistics makes it easy to break
 - If attacker knows the message is in plain English can use known patterns in English language:
 - Frequency of letters
 - Frequency of pairs of letters (bigram/digram) and triples of letters (trigram)
 - Known or expected words in plaintext
 - *Break it by guessing words from their letter patterns, or by using the relative frequency of individual letters. E is most frequent, T is second, etc.*

Cryptanalysis: Frequency Analysis

- **Frequency analysis** is the study of the frequency of letters or groups of letters in a ciphertext.
- **Frequency analysis** is a technique based on the idea that certain letters or letter combinations appear more frequently in a given language
 - For example, in English, the letter **e** is the most common letter, while **q** and **z** are the least common
- By analyzing the frequency of letters in a ciphertext, an attacker can make educated guesses about the mapping of letters in the key and eventually decrypt the message.

Cryptanalysis: Frequency Analysis

- The following chart shows the frequency of each letter in the English alphabet.



Cryptanalysis: Frequency Analysis

1. Count the frequency of each letter in the ciphertext.
2. Compare the frequency of the letters in the ciphertext to the known frequency of letters in the English language.
3. Use this information to make educated guesses about which letters in the ciphertext correspond to which letters in the English language.
4. Substitute the letters in the ciphertext with your guesses.
4. Repeat the above steps and refine your guesses until you have deciphered the message.

Cryptanalysis of substitution ciphers

- Frequency analysis works well with substitution ciphers.
- We replace one letter with another one but it doesn't affect the frequency distributions.
 - Calculate the frequency table.
 - Try to guess the most popular letters.
 - Try to find pairs and triples of letters.
 - Fill in the blanks.

Transposition

- Let's try another approach to hide information.
- What else can we do with the plaintext message?
- Instead of replacing letters, focus on their positions.
- **Transposition ciphers**
 - The name given to any encryption that involves rearranging the plaintext letters in a new order
 - i.e., rearrange characters of plaintext to produce ciphertext.

Simple Transposition

- The key to the cipher is the number of rows and columns of a matrix.
- Encipher a message by writing the plaintext into the matrix by rows and reading the ciphertext out of the matrix by columns.
- Plaintext: **attackxatxdawn**

a	t	t	a
c	k	x	a
t	x	d	a
w	n	x	x

- Ciphertext: **ACTWTKXNTXDXAAAX**

Double Transposition

- Can we do better?
- We could put the plaintext into an array and permute the rows and columns
- Plaintext: **attackatdawn**

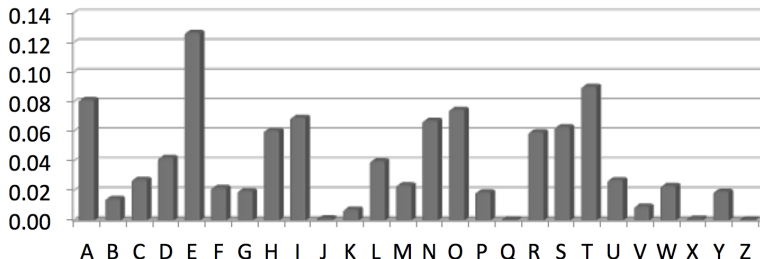
$$\begin{bmatrix} a & t & t & a \\ c & k & a & t \\ d & a & w & n \end{bmatrix} \longrightarrow \begin{bmatrix} d & a & w & n \\ c & k & a & t \\ a & t & t & a \end{bmatrix} \longrightarrow \begin{bmatrix} n & a & d & w \\ t & k & c & a \\ a & t & a & t \end{bmatrix}$$

- Key is the matrix size and the row and column permutations: (3,2,1) and (4,2,1,3).
- Ciphertext: **NADWTKCAATAT**

- You are given the ciphertext **xtawxnattxadakc**. How do you find the plaintext?
- Assume you know a transposition cipher was used.
 - You need to reconstruct the matrix and figure out the scrambling method.
 - Single transposition: guess the number of columns.
 - Double transposition: also need the column and row ordering.

Cryptanalysis

- We learned about frequency analysis! Why can't we use it here?
- Well, we can. But will it do us any good?
- This is what you will get.



- Q: What is going on here?

Big Crypto Ideas

- So, what have we learned so far?
- **3 Big Ideas:**
 - Big Idea #1: Confusion
 - Big Idea #2: Diffusion
 - Big Idea #3: Key secrecy

Confusion & Diffusion

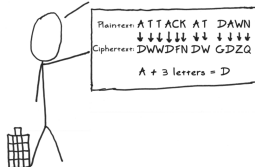
- **Confusion** and **Diffusion** are two properties of the operation of a secure cipher which were identified by Claude Shannon in his paper *Communication Theory of Secrecy Systems*¹.
- DES, AES and many block ciphers are designed using Shannon's idea of confusion and diffusion.

¹<http://pages.cs.wisc.edu/~rist/642-spring-2014/shannon-secrecy.pdf>

Confusion

Big Idea #1: Confusion

It's a good idea to obscure the relationship between your real message and your 'encrypted' message. An example of this 'confusion' is the trusty ol' Caesar Cipher:



www.mosware.com/2009/09/stick-figure-guide-to-advanced.html

- **Confusion** is a fundamental property of secure ciphers that aims to obscure the relationship between the ciphertext and the key.
- It's designed to make it extremely challenging for an attacker to extract any meaningful information about the plaintext or the key from the ciphertext alone.
- *No leaks of information. You should not be able to find patterns*

Big Idea #2: Diffusion

It's also a good idea to spread out the message. An example of this 'diffusion' is a simple column transposition:

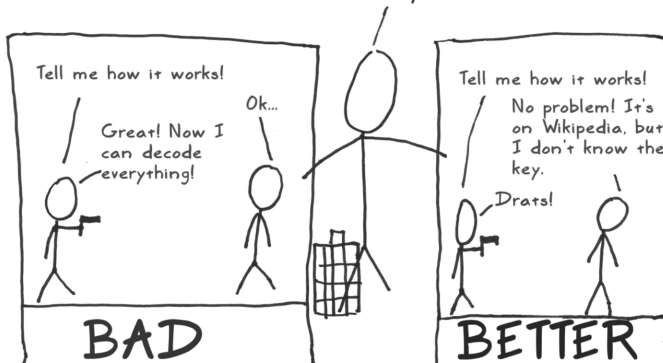


www.moservare.com/2009/09/stick-figure-guide-to-advanced.html

- **Diffusion**: Refers to the process of spreading the influence of each plaintext element over a large portion of the ciphertext
- This ensures that even minor modifications to the plaintext trigger widespread, unpredictable changes within the ciphertext.
- It helps in preventing patterns from being preserved in the encryption process, making the cipher more resistant to statistical analysis and attacks.

Big Idea #3: Secrecy Only in the Key

After thousands of years, we learned that it's a bad idea to assume that no one knows how your method works. Someone will eventually find that out.



www.moserware.com/2009/09/stick-figure-guide-to-advanced.html

This is known as Kerckhoffs Principle.

Combining Ciphers

- Confusion (substitution) and diffusion (transposition) on their own are not enough.
- What if we combine multiple substitution **or** multiple transposition ciphers?
 - Two (or more) substitutions are really only one more complex substitution.
 - Two (or more) transpositions are really only one more complex transposition.
- But: it makes sense to combine substitution and transposition!
- You get the best of both worlds!

Mono-alphabetic cipher

- A **shift cipher** uses a letter substitution defined by a rotation of the alphabet.
- Any cipher that uses a substitution to replace a plaintext letter by a ciphertext letter is called a **substitution cipher**.
 - A shift cipher is a special case of a substitution cipher.
- Any cipher that encrypts a message by applying the same substitution to each letter of the message is called a **mono-alphabetic cipher**.
- A **mono-alphabetic substitution cipher** relies on a fixed replacement structure, where each plaintext letter is replaced by a fixed ciphertext letter.
 - For example, if “a” is encrypted to “S”, then every time we see the letter “a” in the plaintext, we replace it with the letter “S” in the ciphertext.

Poly-alphabetic ciphers

- Poly-alphabetic ciphers are another way to strengthen substitution ciphers by using different substitutions for different letter positions.
 - Choose sequence of r different alphabet permutations π_1, \dots, π_r for some number r .
 - Each permutation is used for a specific letter position:
 - Use π_1 for the first letter, π_2 for the second letter, etc.
 - The sequence is repeated after every r letters.
- This makes it much harder to break than mono-alphabetic ciphers, but letter frequency analysis can still be used
- As every r^{th} letter is encrypted using the same permutation, the sub-message consisting of just those letters still exhibits normal English language letter frequencies.
- *The permutations are applied cyclically, meaning the first permutation is used for the first letter, the second for the second letter, and so on, repeating the sequence for every r letters.*

Vigènere Cipher

- The **Vigènere cipher** is a **polyalphabetic cipher** in which the number of different substitutions r is also part of the key.
- The adversary must determine r as well as discover the different substitutions in order to break the encryption.
 - *This makes it more secure than monoalphabetic ciphers.*
- All polyalphabetic ciphers can be broken using letter frequency analysis, but they are secure enough against manual attacks to have been used in the past.
- The German Enigma encryption machine used in the second world war is also based on a polyalphabetic cipher.

Vigènere Cipher

- Like Caesar cipher, but use a phrase as key
- Example
 - **Message:** THE BOY HAS THE BALL
 - **Key:** VIG
 - **Encipher:** using shift cipher for each letter:

key	VIGVIGVIGVIGVIGV
plain	THEBOYHASTHEBALL
cipher	OPKWECIYOPKWIRG
- The **Vigènere cipher** uses a 26×26 table with A to Z as the row heading and column heading.
- This table is usually referred to as the **Vigènere Tableau**, **Vigènere Table** or **Vigènere Square**.

The Vigenère Tableau

	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

Vigenère Cipher: Encrypt

1. Choose a keyword (e.g., **VIG**).
2. Place the keyword across the top of the text you want to encrypt, repeating it as needed.
3. For each letter in the text, locate the corresponding letter in the keyword row of the [Vigenère table](#).
 - if it was 'V', then you would go to the row that starts with an 'V'
4. Identify the column of your plaintext letter (e.g., 't' corresponds to column 20).
5. Follow the column down to the row determined by the keyword, and note the intersecting letter (e.g., 'o').
6. Repeat this process for each letter in the plaintext to encrypt the entire message.

Vigènere Cipher

	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

- Example
- key V, letter T: follow V column down to T row (giving "O")
- key I, letter H: follow I column down to H row (giving "P")

key	VIGVIGVIGVIGVIGV
plain	THEBOYHASTHEBALL
cipher	OPKWWECIYOPKWIRG

Vigenère Cipher: Decrypt

1. Choose a keyword (e.g., VIG).
2. Place the keyword across the top of the text you want to decrypt, repeating it as needed.
3. For each letter in the ciphertext, identify the corresponding letter in the keyword row of the [Vigenère table](#).
4. Use this row to find the letter heading of the column containing the ciphertext letter. This identified letter is the corresponding plaintext letter (e.g., finding 't' in this case).
 - *The letter at the top of that column is the corresponding plaintext letter.*
5. Repeat this process for each letter in the ciphertext to decrypt the entire message.

Security of Vigenere Cipher

- Vigenere masks the frequency with which a character appears in a language:
 - one letter in the ciphertext corresponds to multiple letters in the plaintext.
- Makes the use of frequency analysis more difficult.
- Any message encrypted by a Vigenere cipher is a collection of as many shift ciphers as there are letters in the key.

Attacking the Cipher

1. Establish the period (length) of the key by finding repeated patterns in the ciphertext and analyzing the distance between them.
2. Break the ciphertext into r parts, where r is the period determined in step 1. Each part is enciphered using the same key letter.
3. Solve each part individually by using a technique such as frequency analysis to determine the most likely plaintext letters corresponding to the ciphertext letters.

Cryptanalysis: Terminology

- Cryptosystem is **secure** if best known attack is to try all keys.
 - Exhaustive key search, that is.
- Cryptosystem is **insecure** if any shortcut attack is known.
- **Q: Are there any completely secure ciphers?**

Perfect Secret-key Encryption

- Is it possible to make a completely unbreakable cryptosystem?
- Yes: The Vernam cipher (one-time pad)

Vernam cipher (One-time pad)

- One-time pad is a cipher that cannot be broken if it is used correctly.
- Rules:
 - The key is as long as the message.
 - The key is random.
 - The key is never reused.

Exclusive-or on bits

- Vernam cipher is based on exclusive-or (XOR), which we write as \oplus
 - $x \oplus y$ is **true** when exactly one of x and y is **true**.
 - $x \oplus y$ is **false** when x and y are both **true** or both **false**.
- Exclusive-or is just sum modulo two if 1 represents **true** and 0 represents **false**.

$$x \oplus y = (x + y) \bmod 2$$

- XOR is associative and commutative. 0 is the identity element.

$$k \oplus 0 = 0 \oplus k = k$$

- XOR is its own inverse.

$$k \oplus k = 0$$

One-Time Pad: Informal description

- The one-time pad encrypts a message m by XORing it with the key k , which must be as long as m .
- Assume both m and k are represented by strings of bits. Then ciphertext bit $c_i = m_i \oplus k_i$.
- Note that $c_i = m_i$ if $k_i = 0$, and $c_i = \neg m_i$ if $k_i = 1$.
- Decryption is the same, i.e., $m_i = c_i \oplus k_i$

One-Time Pad: Encryption

- Let $a=000$, $h=001$, $i=011$, $k=100$, $p=101$, $y=111$
- **Encryption:** Plaintext \oplus Key = Ciphertext

	h	a	p	p	y
Plaintext	001	000	101	101	111
Key	101	111	110	101	011
Ciphertext	100	111	011	000	100
	k	y	i	a	k

One-Time Pad: Decryption

- Let $a=000$, $h=001$, $i=011$, $k=100$, $p=101$, $y=111$
- **Decryption:** **Ciphertext \oplus Key = Plaintext**

	k	y	i	a	k
Ciphertext	100	111	011	000	100
Key	101	111	110	101	011
Plaintext	001	000	101	101	111
	h	a	p	p	y

The one-time pad cryptosystem formally defined

- $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0, 1\}^r$ for some length r .
- $E_k(m) = k \oplus m$, where \oplus is applied to corresponding bits of k & m .
- $D_k(c) = k \oplus c$, where \oplus is applied to corresponding bits of k & c .
- It works because

$$D_k(E_k(m)) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = 0 \oplus m = m$$

One-time pad: Security

- Like the 1-letter Caesar cipher, for given m and c , there is exactly one key k such that $E_k(m) = c$ (namely, $k = m \oplus c$)
- For fixed c , m varies over all possible messages as k ranges over all possible keys, so c gives no information about m .
- It will follow that the one-time pad is information-theoretically secure¹

¹ Information-theoretic security is a cryptosystem whose security derives purely from information theory. In other words, it cannot be broken even if the adversary had unlimited computing power. The adversary simply does not have enough information to break the encryption and so the cryptosystems are considered cryptanalytically-unbreakable

Importance of the Vernam cipher

- It is important because
 - it is sometimes used in practice;
 - it is the basis for many [stream ciphers](#), where the truly random key is replaced by a pseudo-random bit string.

Attraction of one-time pad

- The one-time pad would seem to be the perfect cryptosystem.
 - It works for messages of any length (by choosing a key of the same length).
 - It is easy to encrypt and decrypt.
 - It is information-theoretically secure.
- In fact, it is sometimes used for highly sensitive data.
 - *It's still in use today, particularly in military and diplomatic communication, where the highest level of security is required. However, it is relatively rare in general use due to the difficulty and expense of generating and distributing truly random keys of sufficient length, and as a result, it's usually not practical for most everyday uses.*

Drawbacks of one-time pad

- It has two major drawbacks:
 1. The key k must be as long as the message to be encrypted.
 2. The same key must never be used more than once. (Hence the term “one-time”.)
- Together, these make the problem of key distribution and key management very difficult.

Why the key cannot be reused

- If Eve knows just one plaintext-ciphertext pair (m_1, c_1) , then she can recover the key $k = m_1 \oplus c_1$.
- This allows her to decrypt all future messages sent with that key.
- Even in a ciphertext-only situation, if Eve has two ciphertexts c_1 and c_2 encrypted by the same key k , she can gain significant partial information about the corresponding messages m_1 and m_2 .
 - In particular, she can compute $m_1 \oplus m_2$ without knowing either m_1 or m_2

$$c_1 \oplus c_2 = (m_1 \oplus k) \oplus (m_2 \oplus k) = m_1 \oplus m_2$$

How knowing $m_1 \oplus m_2$ might help an attacker

- Fact (important property of \oplus)
 - For bits b_1 and b_2 , $b_1 \oplus b_2 = 0$ if and only if $b_1 = b_2$
 - Hence, blocks of 0's in $m_1 \oplus m_2$ indicate regions where the two messages m_1 and m_2 are identical.
 - That information, together with other information, Eve might have about the likely content of the messages, may be enough for her to seriously compromise the secrecy of the data.

One-Time Pad Summary

- Ciphertext provides no info about plaintext.
- But, only when used correctly!
 - Pad must be random, used only once.
 - Pad is known only to sender and receiver.
- Note: pad (key) is same size as message.
 - So, why not distribute message instead of pad?

Key Points

- Two basic types of ciphers
 - Transposition ciphers and substitution ciphers
- Caesar cipher uses one key
- Vigenère cipher uses a sequence of keys
- One-time pad is a provably secure cryptosystem but not practical to use in modern communication
- Cryptanalysis
 - Exhaustive search
 - Statistical analysis

- Information Security: Principles and Practice, 2nd edition (available online)
 - Chapter 2 (Till 2.3.5)
- Computer Security: Principles and Practice
 - Chapter 20 (Only 20.1)

- Next: Modern Cryptography