CS361: Introduction to Computer Security

Cryptography I

Dr. Bill Young
Department of Computer Sciences
University of Texas at Austin

Last updated: February 25, 2020 at 12:03

Elementary Cryptography

This is not a course in cryptography. The department offers one and you are advised to take it, if you plan to work in the security field.

Our point here will be to give some intuitions about:

- what are the key concepts of cryptography;
- how is it used as a tool for security;
- how effective is it in that regard.

A Thought Experiment

Suppose you're confronted with a text that you believe to be the encryption of some message. You'd like to apply your cryptanalytic skills. How do you get started? What questions should you ask?

A Thought Experiment

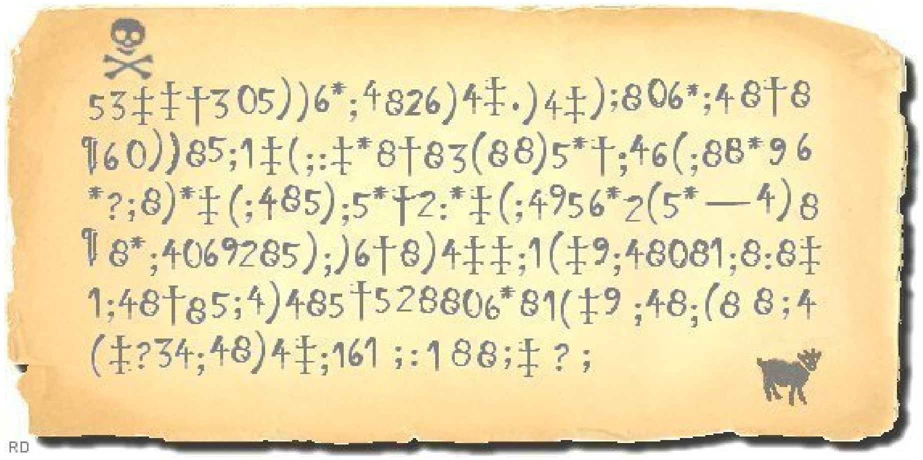
Suppose you're confronted with a text that you believe to be the encryption of some message. You'd like to apply your cryptanalytic skills. How do you get started? What questions should you ask?

- What is the likely underlying language of the plaintext?
- What characteristics of the probable source text are relevant?
- What characteristics of the source language are relevant?
- Have any transformations/compressions been applied prior to encryption?
- What is the likely nature/complexity of the encryption algorithm?
- Anything else?

Thought Experiment: The Gold Bug

The setting: In the early 1800's, a man named William Legrand finds a scrap of parchment on a South Carolina beach. The parchment appears blank, but when he holds it close to a candle flame to examine it, a strange encoded message appears. In one corner is a drawing of a goat. Legrand wonders if the message could be directions to the location of a treasure buried by the infamous pirate Captain Kidd.

The Ciphertext



53‡‡†305))6*;4826)4‡.)4‡);806*;48†8
¶60))85;1‡(;:‡*8†83(88)5*†;46(;88*96
?;8)‡(;485);5*†2:*‡(;4956*2(5*—4)8
¶8*;4069285);)6†8)4‡‡;1(‡9;48081;8:8‡
1;48†85;4)485†528806*81(‡9;48;(88;4
(‡?34;48)4‡;161;:188;‡?;

An Aside: Talk Like a Pirate

A useful Pirate to English translator can be found at:
`http://www.talklikeapirate.com/translator.html`

Information Theory and Cryptography

Information theory vitally informs cryptography in a number of ways:

- What effect does encoding a message have on the information content of the file?
- An attempt to decrypt a message is really an attempt to recover a message from a (systematically) noisy channel.
- How can redundancy in the source give clues to the decoding process?
- Is a perfect encryption possible (i.e., one that is theoretically unbreakable) ?

Message Transmission

Consider the steps in sending messages from sender S to recipient R. Suppose S entrusts the message to T, who delivers it to R. We call T a **transmission medium**.

If an outsider O wants to access the message (to read, change, or destroy it), we call O an **interceptor** or **intruder**. This might take different forms:

block it	by preventing it reaching R;
intercept it	by reading or listening to the message;
modify it	by seizing the message and changing it;
fabricate	an authentic looking message and cause it to be delivered.

For which of these would encryption help?

Encryption / Decryption

The purpose of encryption is to render the message less useful / meaningful to the intruder. Conceptually, the process of encryption is quite simple:

$$\begin{tikzcd} {} \arrow[r, "\text{plaintext}"] & \boxed{\text{Encrypt}} \arrow[r, "\text{ciphertext}"] & {} \\ & \text{key}_e \text{ (optional)} \arrow[u] & \end{tikzcd}$$

as is the process of decryption:

$$\begin{tikzcd} {} \arrow[r, "\text{ciphertext}"] & \boxed{\text{Decrypt}} \arrow[r, "\text{plaintext}"] & {} \\ & \text{key}_d \text{ (optional)} \arrow[u] & \end{tikzcd}$$

Some Terminology

Encryption is the process of encoding a message so that its meaning is not obvious.

Decryption is the reverse process, transforming an encrypted message back to its original form.

The terms *encrypt*, *encode*, and *encipher* are used interchangeably, as are *decrypt*, *decode*, and *decipher*.

A system for encryption and decryption is called a *cryptosystem*.

The original form of a message is called *plaintext* and the encrypted form called *ciphertext*.

More Terminology

Encryption and decryption are functions which transform one text into another. In functional notation:

$$C = E(P) \quad \text{and} \quad P = D(C)$$

where C denotes ciphertext, E is the encryption rule, D is the decryption rule, P is the plaintext. In this case, we also have:

$$P = D(E(P))$$

It is obviously important to be able to recover the original message from the ciphertext.

Keyed Algorithms

Often the encryption and decryption algorithms use a *key* K. The key selects a specific algorithm from the family of algorithms defined by E.

We write this dependence as:

$$C = E(P, K_E) \quad \text{and} \quad P = D(C, K_D)$$

If $K_E = K_D$, then the algorithm is called *symmetric*. If not, then it is called *asymmetric*. In general,

$$P = D(E(P, K_E), K_D)$$

An algorithm that does not use a key is called a *keyless cipher*.

Some Notation

Often the notation $E(P, K)$ and $D(C, K)$ becomes cumbersome. An alternative notation is often used, particularly in cryptographic protocols.

We'll often use $\{M\}_K$ to denote $E(M, K)$, and sometimes to denote $D(M, K)$. For example,

$$P = D(E(P, K_E), K_D) = \{\{P\}_{K_E}\}_{K_D}.$$

This is usually appropriate since, in many of the most important commercial crypto systems, the same algorithm is used for both encryption and decryption (i.e., the algorithm is its own inverse).

Some More Terminology

The word *cryptography* means "secret writing." It refers to the practice of using encryption to conceal text.

Cryptanalysis is the attempt to extract the meaning of encrypted messages.

Cryptology is the research into and study of encryption and decryption; it includes both cryptography and cryptanalysis.

Cryptanalysis

A cryptanalyst can attempt to do any or all of the following:

- to break a single message;
- to recognize patterns in encrypted messages;
- to infer some meaning without breaking the algorithm;
- to deduce the key;
- to find weaknesses in the implementation or environment or the use of encryption;
- to find weaknesses in the algorithm, without necessarily having intercepted any messages.

Cryptanalysis (Cont.)

The analyst works with:

- encrypted messages,
- known encryption algorithms,
- intercepted plaintext,
- data items known or suspected to be in a ciphertext message,
- mathematical and statistical tools and techniques,
- properties of languages,
- computers,
- ingenuity and luck.

Breakable Encryption

An encryption algorithm is called *breakable* if, given enough time and data, an analyst can recover the plaintext.

However, just because an algorithm is breakable doesn't mean that it is feasible to break it.

Example: consider a simple substitution algorithm on the 26 characters of English. There are something like 26! different possible encipherments. Checking 10^{10} per second, this would still require approximately millenia to check them all.

This is infeasible, and obviously unnecessary. No-one would attempt to break a simple substitution that way.

Breakability Evolves

Suppose we use a more ingenious approach that reduces this to 10^{15} operations. An exhaustive approach would require only about one day. (But still not be needed, probably!)

Because of advances in computer technology, algorithms that were considered *strong enough* 20 years ago, can be effectively broken today.

You see the result in current discussion of increasing the *key length* for standard algorithms such as DES and RSA. We'll consider this issue later.

Strong Encryption

A cryptosystem is *strong* if there are no "short cuts" to breaking it. That is, there is no cryptoanalytic approach that is substantially faster than brute force—i.e., trying all of the keys one by one. *Most strong algorithms are still breakable.*

For an n-bit block cipher with k-bit key, given a small number of plaintext/ciphertext pairs encrypted under key K, K can be recovered by exhaustive search in an expected time on the order of 2^{k-1} operations.

The larger the *keyspace*, the longer to find the key by search. Thus, an important question for any cryptosystem: What is the size of the keyspace? How does this relate to the size of the key?

Types of Ciphers

The simplest building blocks of encryption are:

substitution: in which each symbol is exchanged for another (not necessarily uniformly), and

transposition: in which the order of symbols is rearranged.

It might seem that these are too naive to be effective. *But almost all modern commercial symmetric ciphers use some combination of substitution and transposition for encryption.* The same cannot be said for asymmetric ciphers such as RSA.

Substitution Ciphers

A substitution cipher is one in which each symbol of the plaintext is exchanged for another symbol. If this is done uniformly (eg. every occurrence of X is replaced by Y) this is called a *monoalphabetic cipher* or *simple substitution cipher*. An example is the Caesar cipher.

If different substitutions are made for a letter depending on where in the plaintext the letter occurs, this is called a *polyalphabetic substitution*. An example is the Vigenère cipher.

Caesar Cipher

The idea of the Caesar Cipher is that each letter is replaced in the encryption by another letter a fixed “distance” away in the alphabet, circularly. For example, A is replaced by C, B by D, ..., Y by A, Z by B, etc.

This encryption scheme is said to have been used by Julius Caesar.

Like all early schemes, simple substitution had to be easy to perform in the field. Simplicity did not substantially compromise the safety of the encryption since few people could read, anyway.

What is the size of the keyspace? Is the algorithm strong?

Substitution Ciphers

In general, a simple substitution cipher is an injection (1-1 mapping) of the alphabet into itself or another alphabet. The key is a table or other scheme that exhibits the mapping.

Breaking the cipher theoretically can be done via brute force. However, there are $k!$ permutations of an alphabet of k characters. Thus, we generally rely on redundancy in the source language for clues that speed the decryption.

Note that not all substitution ciphers are simple substitution ciphers.

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

Running Key Ciphers

One source of keys is a book, poem, or other text available to both sender or receiver. Suppose both agree to use text from page 851 of *Computer Security: Art and Science* to encode the following text: "four score and seven years ago." Align the two texts, possibly removing spaces:

plaintext:	`fours corea ndsev enyea rsago`
key:	`monit orsto gotot hebat hroom`
ciphertext:	`rcizl qfkxo trlso lrzet yjoua`

Then use the letter pairs to look up an encryption in a table (called a *Vigenère Tableau* or *tabula recta*). For example, look up the pair (`f`, `m`) by consulting row `f` and column `m` to obtain `r`, etc.
What is the corresponding decryption algorithm?

Running Key Ciphers (Cont.)

Using the Vigenère Tableau means that you are using one of twenty-six different Caesar Ciphers at each position, depending upon the corresponding letter in the key.

Some people use as a key a short word or phrase repeated over and over again.

key:	`light light light light light ...`
inverse key:	`psuth psuth psuth psuth psuth ...`

Though easy to remember, it is weak because every fifth letter is encoded with the same Caesar Cipher. If you knew the key period (five letters) and had enough ciphertext, it would be easy to decipher with a bit of statistical analysis and trial and error.

Running Key Ciphers (Cont.)

In general, running key ciphers are susceptible to statistical analysis. Notice both the key and the plaintext are English language strings and so have the entropy characteristics of English. In particular, the letters A, E, O, T, N, I make up approximately 50% of English text. Thus, at approximately 25% of indices, these can be expected to coincide.

Then we work backwards from the tableau, looking for those letters in the ciphertext that correspond to the row-column intersection of combinations of these common letters.

This is an example of a *regularity* in the ciphertext that would not be expected merely from chance. It provides clues that can be used by the cryptanalyst.

Aside: Complexity of the Algorithm

As with any other algorithm in computer science, you can ask about the computational complexity of an encryption algorithm. The input is the plaintext, so the complexity should be a function of the *length* of the plaintext. Time and space complexity are both important.

For any simple substitution cipher, the time complexity of the encryption and decryption is obviously $O(n)$ since constant time is required for each symbol of the plaintext / ciphertext.

The space complexity is clearly $O(1)$ since the only space required is to store the translation table and the current symbol.

Can you think of encryption algorithms for which the time complexity would not be linear and/or for which the space complexity would not be constant?

Substitution Ciphers

Suppose you have an alphabet of 26 letters and your encryption algorithm is a bijection of the alphabet onto itself.

What is the size of the keyspace? What is the effective size of the key? Is the algorithm strong?

Note that one hallmark of a *simple* substitution algorithm is that the symbol frequencies of the plaintext are preserved, but associated with other symbols. I.e., in the ciphertext, there will be *some* symbol with the frequency of E in the plaintext, etc.

A Proviso

The goal of one common task of the cryptanalyst is, given specific ciphertext, to reduce as much as possible the uncertainty in the plaintext from which it was produced.

That's why a successful cryptanalysis benefits from a larger quantity of ciphertext on which to operate.

Using Information

Suppose you know that the following is an encoding of a string over the English alphabet (26 letters) using a substitution cipher: `xyy`. How many decryptions are possible?

Using Information

Suppose you know that the following is an encoding of a string over the English alphabet (26 letters) using a substitution cipher: `xyy`. How many decryptions are possible?

With no additional information: $26^3 = 17576$

Using Information

Suppose you know that the following is an encoding of a string over the English alphabet (26 letters) using a substitution cipher: `xyy`. How many decryptions are possible?

With no additional information: $26^3 = 17576$

If you know a simple substitution cipher was used: $26 \times 25 = 650$. (Reduce search space by a factor of 27.)

Using Information

Suppose you know that the following is an encoding of a string over the English alphabet (26 letters) using a substitution cipher: `xyy`. How many decryptions are possible?

With no additional information: $26^3 = 17576$

If you know a simple substitution cipher was used: $26 \times 25 = 650$. (Reduce search space by a factor of 27.)

and you know the plaintext is an English word: around 40. (Reduce original search space by a factor of 439.)

Add Context

Now suppose you know that the following is an encoding of a common English phrase using a simple substitution cipher: `xyy rqk`.

What can you infer? How many different potential decryptions come to mind?

Length Preserving Ciphers

Consider the set of strings of length n over an alphabet A, this is sometimes denoted as A^n. A *length preserving cipher* is a mapping from $A^n \to B^n$.

Most substitution ciphers are length preserving.

Perfect Ciphers

A perfect cipher would be one in which having access to the cyphertext doesn't allow you to reduce the search space at all.

Very roughly speaking, given a plaintext string $P \in A^n$ and an encryption algorithm E_k for key k such that $E_k(P) = C$, then (abusing notation) the encryption is *perfect* if,

$$h(P|C) = h(P).$$

That is, the uncertainty (the likelihood of guessing the plaintext) of the message is exactly the same *whether or not you know the ciphertext.*

Perfect Ciphers

"Perfect Secrecy" is defined by requiring of a system that after a cryptogram is intercepted by the enemy the a posteriori probabilities of this cryptogram representing various messages be identically the same as the a priori probabilities of the same messages before the interception. It is shown that perfect secrecy is possible but requires, if the number of messages is finite, the same number of possible keys. If the message is thought of as being constantly generated at a given "rate" R (to be defined later), key must be generated at the same or a greater rate. (Claude Shannon)

One Time Pad

A *one-time pad*, invented in 1917 by Vernam and Mauborgne, is theoretically considered a perfect cipher. This was proved by Claude Shannon.

The idea is to use a key that is the same length as the plaintext, and to use it only once. The key is XOR'd with the plaintext.

Is this theoretically unbreakable? Why or why not? Does it depend on the characteristics of the input language?

There are two practical problems with the one-time pad scheme:

1. It requires absolute synchronization between sender and receiver, and
2. it requires an unlimited amount of key material.

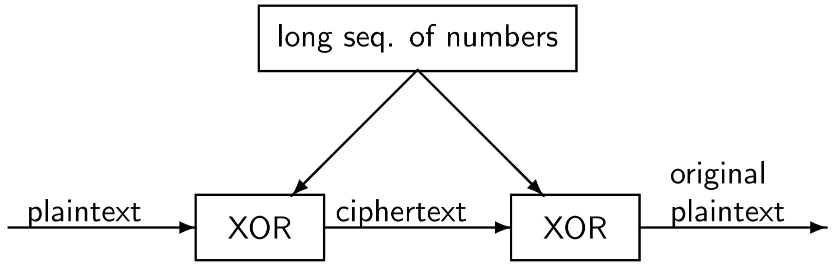
Key Distribution

The main problem with the one-time pad is practical, rather than theoretical, and is a problem with all symmetric encryption algorithms. It is the *key distribution* problem, to which we will return.

That is, given the need to communicate securely, how do the sender and receiver agree on a *secret* (key) that they can use in the algorithm. If the keys are as long as the message, distribution of the keys becomes a significant challenge that faces many modern cryptographic algorithms.

Vernam Cipher

The *Vernam cipher* is a type of one-time pad suitable for use on computers.



This relies on the fact that $(A \oplus B) \oplus B = A$, where $\oplus$ denotes XOR.

One Time Pad Approximation

A close approximation to the one-time pad for use on computers is a pseudo-random number generator. This generates a long sequence of numbers that can be used as the key.

Another computer running the same random number generator function can produce the key simply by knowing the *seed*. Obviously this would not work if the state of the PRNG is refreshed with randomness from the environment, as is often done in *cryptographic* PRNGs.

This works well because a pseudorandom sequence may have a very long *period*. However, notice that it is susceptible to compromise by someone who knows the algorithm and the seed.

Confusion and Diffusion

Two desirable characteristics for any encryption scheme are the following:

Confusion: transforming information in plaintext so that an interceptor cannot readily extract it.

Diffusion: spreading the information from a region of plaintext widely over the ciphertext.

For example, the Caesar Cipher is poor at confusion; decryption of a few letters leads to decryption of many others. It is also poor at diffusion, since all of the information in a plaintext letter is contained in the corresponding ciphertext letter. *Analyze the one-time pad with respect to these criteria.*

Transposition Ciphers

The goal of substitution is *confusion*. The goal of transposition is *diffusion*.

Columnar transposition is a simple variety of transposition algorithm. Write the plaintext characters in a number of fixed length rows such as the following:

c_1	c_2	c_3	c_4	c_5
c_6	c_7	c_8	c_9	c_{10}
c_{11}	c_{12}	etc.		

Form the ciphertext by reading down the columns: $c_1 c_6 c_{11} c_2 \ldots$. If the message length is not a multiple of the number of columns, pad the final row with any character.

Complexity

The algorithm involves no additional work beyond rearranging the characters, so has *time* complexity linear in the length of the message.

Unlike simple substitution, it has greater *space* complexity since the message can't be decrypted until it has been read in its entirety. This may be an issue for very long messages, and causes a delay in the decryption.

Cryptanalysis of Transpositions

By rearranging the order of characters, the first-level entropy of the text is maintained, but higher levels are disrupted. That is, letter frequencies are preserved in the ciphertext, but the frequencies of digrams, trigrams, etc. are not.

Hence, if an analysis shows that the first level entropy of the ciphertext is that of the source language, a transposition may be in use. Then a systematic approach is called for.

In a columnar transposition with rows of length n, adjacent characters in the plaintext are at c_1 and c_{n+1}, c_2 and c_{n+2}, etc. We hypothesize a distance of n and check these pairs to see if they contain common digrams, as would be expected for the language. If so, we may have found the key to the cipher. Otherwise, try a distance of $n+1$.

Combinations of Approaches

Substitutions and transpositions can be regarded as building blocks for encryption. Some important commercial algorithms use these, in concert with other approaches.

A combination of two or more ciphers is called a *product cipher* or sometimes a *cascade cipher*. These are typically performed one after another.

$$E_2(E_1(P, k_1), k_2)$$

Note that a combination is not necessarily stronger than either cipher individually. It may be stronger, but it may even be weaker.

What is Good Encryption?

The value of an encryption algorithm depends on the use. An algorithm that is appropriate for computers might not be appropriate for use by an agent in the field.
Shannon (1949) listed characteristics of good ciphers.

1. The degree of secrecy required should determine the effort expending in encryption / decryption.
2. The keys and algorithm should be free from complexity.
3. The implementation should be as simple as possible.
4. Errors in encryption should not propagate.
5. The size of the ciphertext shouldn't be more than the size of the plaintext.

Which of these still apply to modern cryptography?

Other Criteria

Some of Shannon's criteria don't really apply to modern computer-based cryptography.

The following are suggested as other tests of worth for current cryptographic practice:

- is based on sound mathematics;
- has been analyzed by competent experts and found to be sound;
- has stood the test of time.

We'll consider three important modern algorithms: DES, RSA, and AES.

Symmetric and Asymmetric Systems

Recall that there are two basic types of encryption:

symmetric algorithms (also called "secret key") use a single key for both encryption and decryption;

asymmetric algorithms (also called "public key") use different keys for encryption and decryption.

For any encryption approach, there are two major challenges:

Key distribution: how do we convey keys to those who need them to establish secure communication.

Key management: given a large number of keys, how do we preserve their safety and make them available as needed.

How Many Keys Are Needed

A symmetric system provides a two-way channel for users. It requires a key for each pair of individuals for whom a secure channel is needed. Thus, for n users needing pairwise communication, $n(n-1)/2$ keys are required.

An asymmetric system provides a means for *anyone* to communicate with an individual securely. The receiver maintains a *private key* to be used for decryption, and publicizes a *public key* that can be used by anyone to encrypt. Thus, each individual requires two keys. For n individuals, $2n$ keys are required in the system.

Authentication

In a symmetric system, the key is a *shared secret* that can be used for *authentication*.

The receipt of an encrypted message that correctly decrypts with the key is *de facto* proof that the sender shares the key. This authenticates the identity of the sender, assuming that the key distribution and management practices are secure. Does it provide confidentiality?

For an *asymmetric* system, does receipt of a message encrypted with the user's public key provide authentication? How about the private key? Does either provide confidentiality? *Note that not all public key systems allow encryption with the private key. RSA does, but most others do not.*

Stream and Block Ciphers

Another important distinction in cryptographic algorithms is between stream and block ciphers.

Stream ciphers convert one symbol of plaintext directly into a symbol of ciphertext.

Block ciphers encrypts a group of plaintext symbols as one block.

Simple substitution is an example of a stream cipher. Columnar transposition is a block cipher. Most modern symmetric encryption algorithms are block ciphers. Can stream ciphers ever excel in diffusion?

Stream Encryption

Advantages:

- *Speed of transformation:* since symbols are encrypted individually, the algorithms are linear (in time).
- *Low error propogation:* an error in encrypting one symbol likely will not affect subsequent symbols.

Disadvantages:

- *Low diffusion:* all information of a plaintext symbol is contained in a single ciphertext symbol.
- *Susceptibility to insertions/ modifications:* an active interceptor who breaks the algorithm might insert spurious new messages that may look authentic.

Block Encryption

Advantages:

- *High diffusion:* information from one plaintext symbol is diffused into several ciphertext symbols.
- *Immunity to tampering:* it is difficult to insert symbols without detection.

Disadvantages:

- *Slowness of encryption:* an entire block must be accumulated before encryption / decryption can begin.
- *Error propogation:* An error in one symbol may corrupt the entire block.

Malleability

An encryption algorithm is said to be *malleable* if transformations on the ciphertext produce meaningful changes in the plaintext.

That is, given a plaintext P and the corresponding ciphertext $C = E(P)$, it is possible to generate $C_1 = f(C)$ so that

$$D(C_1) = P_1 = f'(P)$$

with arbitrary, but known, functions f and f'.

An algorithm that is not malleable is called *non-malleable*. Stream ciphers are often malleable encryption algorithms.

Homomorphic Encryption

Homomorphic encryption is a form of encryption where a specific algebraic operation performed on the plaintext is equivalent to another (possibly different) algebraic operation performed on the ciphertext.

Homomorphic encryption schemes are malleable by design. The homomorphic property of various cryptosystems can be used to create secure voting systems, collision-resistant hash functions, and private information retrieval schemes.

Cryptanalysis

Attacks on an encryption algorithm can be classified according to what information is available to the attacker.

Ciphertext-only attack: decryption is based on probabilities, distributions, characteristics of the available ciphertext, plus publicly available information. Any encryption scheme susceptible to this is deemed completely insecure.

Known plaintext: attacker has a quantity of ciphertext and corresponding plaintext.

Chosen plaintext attack: the attacker has infiltrated the sender's transmission process and can cause messages of his choosing to be encrypted.

Cryptanalysis

Adaptive chosen plaintext attack: chosen plaintext attack where the choice of plaintext may depend on the ciphertext from earlier attempts.

Chosen ciphertext attack: the attacker selects a ciphertext and is given the corresponding plaintext. E.g., attacker gains access to the decryption device but not the key.

Recall *the Principle of Easiest Penetration*. Often it is more effective to attack the human users rather than the cryptographic algorithms. Many successful attacks succeed because the users are hurried, lazy, careless, naive or uninformed. Sometimes users can be bribed or coerced.

Kerckhoff's Law

Kerckhoff's law is one expression of our *no security through obscurity* principle.

Kerckhoff's Law: a cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

An equivalent formulation was given by Claude Shannon.
Shannon's Maxim: the enemy knows the system.

Kerckhoff's Law

Every security system depends on keeping *some* things secret. But every secret provides a potential failure point. The things to keep secret should be the things that are easiest and least costly to change if they are compromised.

Changing an algorithm or its implementation is costly. Therefore, the system is *brittle* if its security depends on keeping the algorithm secret.

Relatively speaking, changing a key is easy. Simply generate and distribute a new key.