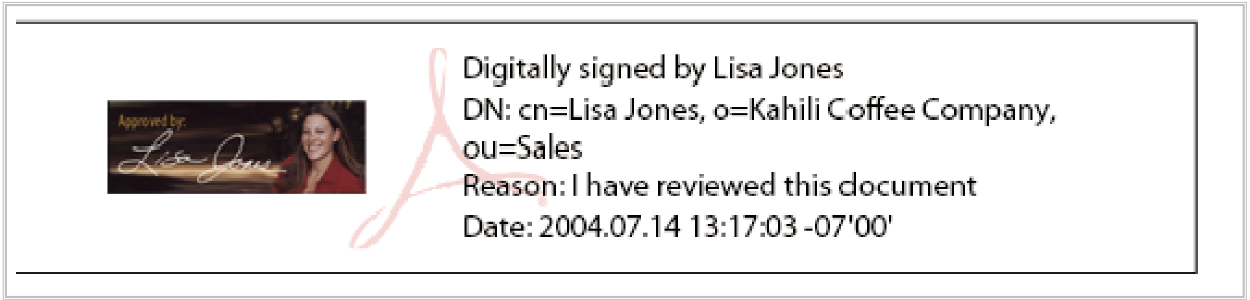
INFORMATION SECURITY

Cryptography[1]

```
What is the secret meaning of the following phrase,
 knowing that it was built with "Caesar's cipher”?

  «Sxw#pruh#frgh/#jhw#pruh#exjv1»

Note: cipher adapted to Latin 1 (ISO 8859-1) table;
      delimiting quotes are not part of code.
```



1 However, keep in mind: «*Cryptography is rarely ever the solution to a security problem*. (D. Gollmann, Computer Security, p. 203)»

Basics

- Originally:
  - science (and art) of secret writing
  - aimed at hinder the knowledge of sensitive information
- Currently:
  - science (and art) of providing mechanisms to ensure security properties (confidentiality, integrity...)
  - aims to control of access to information
- Relevant types of professionals:
  - *cryptographers* - try to master and enhance that access control
  - *cryptanalysts* - try to break the enabled access control

Practical uses

- Traditional
  - control access to information by **concealing** it, i.e. making it unintelligible
- Modern:
  - the traditional, plus
  - control access to information by **identifying** it with a *fingerprint* (or *hash*[1])
  - **support** all above uses and produce "random" numbers, e.g. keys[2]

Cryptography usage:

Traditional	***New***	
P		
ciphering:	fingerprinting:	transmission:
C	*h*	*P'*
deciphering:	verification:	
P	*P = P' ?*	

1 PT: *síntese*, *sumário*
2 pieces of data necessary for using cryptographic security mechanisms

Traditional usage of Cryptography

- confidentiality protection:
  - conceal information, by making it unintelligible
  - *elsewhere or later*, retrieve original information

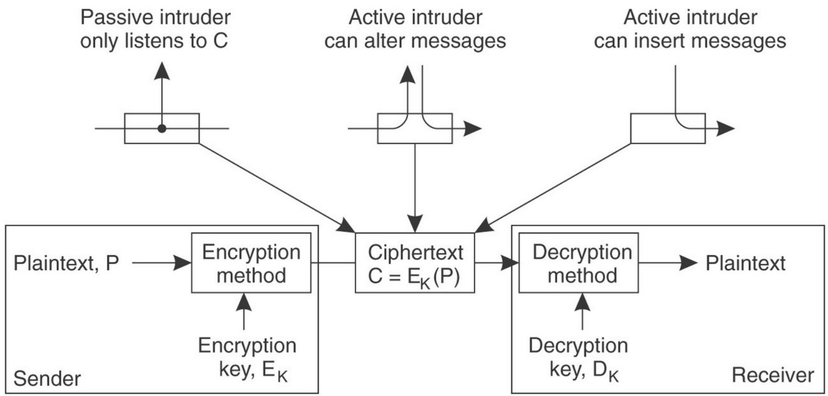


Fig. Original Cryptography: basic model of concealment and recovery of info with examples of attacks (in several of Tanenbaum's books).

Added newer usage of Cryptography

- integrity protection:
  - information is *fingerprinted*,[1] by calculating its *hash (or digest)*
  - *elsewhere or later*, the hash will be used to detect the adulteration of the original information

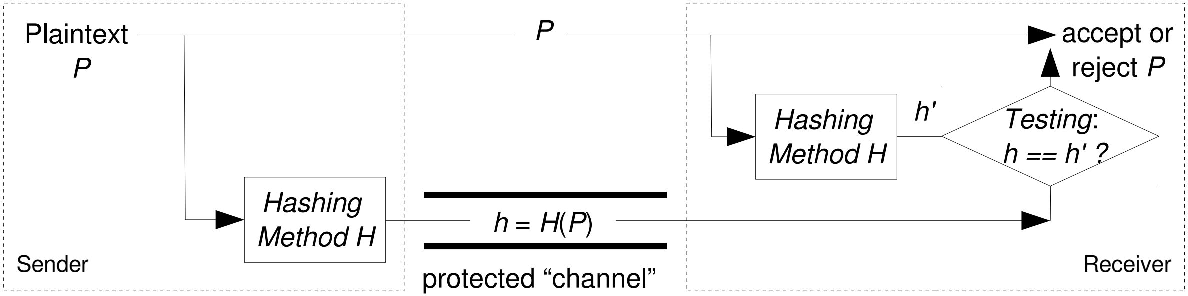


Fig. Modern Cryptography: basic model for the validation of info (e.g. integrity protection). Note the need for a protected channel!

1 small array of bytes that represents the original information

Notation

Symbol	*Name of symbol*	*Meaning of symbol*
$\boldsymbol{P}$	plaintext[1]	original, uncovered information
$\boldsymbol{E}$	enciphering algorithm	method to conceal the info
$\boldsymbol{K_e}$	enciphering key	parameter of the concealment methods
$\boldsymbol{C}$	ciphertext	hidden information
$\boldsymbol{D}$	deciphering algorithm	method to recover the original info
$\boldsymbol{K_d}$	deciphering key	parameter of the recovering methods
$\boldsymbol{H, h}$	hash algorithm, hash value	method to transform (hash) the info, transformed info
$\boldsymbol{F}$	fingerprint, hash value	transformed info

1 PT: texto inteligível

...Notation (cont.)

Operation	Symbolic representation
ciphering	$C = E_{Ke}(P)$ $C = E(P, K_e)$ $C = K_e(P)$
deciphering	$P = D_{Kd}(C)$ $P = D(\mathrm{C}, K_d)$ $P = K_d(C)$
(cyptographic) hashing[1]	$h = H(P)$ $F = H(P)$ $F = h(P)$
reversing	$D_{Kd}(E_{Ke}(P)) = P$

If...	Cryptography type
$K_e = K_d$	symmetric
$K_e \neq K_d$	asymmetric
$K_e = K^+$ $K_d = K^-$	public-key (asymmetric)

Advance notice for Digital Signature:

$[Doc]_E \Longleftrightarrow K_E^-(\mathrm{Doc}) \Longleftrightarrow K_E^-(H(\mathrm{Doc}))$

1 Note: *cryptographic* hashing is different from *database* hashing.

Breaking cryptographic systems

- Professionals: cryptanalysts, random crackers
- Methods: mathematics, statistics, intuition[1]
- Goals: depend on type of usage

Attacks in traditional use

- Goal: grasp the deciphering key! Sometimes, at least, grasp plaintexts.
- Approaches (in descending order of difficulty):
  - normal
    - only ciphertexts are available
  - known original text (“passively” obtained)
    - both some original texts and their enciphered counterparts are available
  - planned original text (“actively” prepared)
    - specific original texts are made to be enciphered

1 For an example, see Bishop: "*Introduction*", Chap.8; "*Art & Science*", chap.9.

Attacks in added recent usage

- Goal: break integrity protection
- Approaches[1] (in descending order of difficulty):
  - find collisions[2]
    - produce chosen document pairs (*birthday attack*[3])
    - produce another document for a specific original

Ideal cryptographic system:

- *hard to break* - in a reasonable future horizon
- *easy to use* - otherwise will be rejected or bypassed by users
- *if broken, easily replaceable* - this should be a must, as systems **will** be broken!

1 The special case of "digital signatures" will be seen elsewhere.
2 meaning: different documents with same fingerprint
3 https://en.wikipedia.org/wiki/Birthday_attack

Classification of cryptographic systems

Perspective	*Variant*	*Sub-variant*	*Examples*
on the secret	secret algorithm	-	RC4 (originally)
	secret key(s)	single key, shared-key, symmetric	AES
		two-key, public key, asymmetric	RSA
on the method	stream[1]	-	RC4
	block	-	AES, RSA[2]
on the purpose	bidirectional, reversible, two-way	confidentiality[3]	AES
		authentication[4]	RSA
	unidirectional, irreversible, one-way	-	MD5, SHA-2

1 PT: *contínuo*
2 Some texts do not consider this to be a "block" cipher, just because of its comparative inefficiency...
3 Keys are temporary and efficient
4 Keys are personal and durable (long-lasting)

Classification of cryptographic systems: *on the secret*

Types of secret

- secret(s) algorithm(s)
- secret(s) key(s)

Secret algorithm systems

Example:

- Discover the algorithm[1] that turns the phrase (quotes not included):

  «`Put more code, get more bugs.`»

  into

  «`Wklt!jx%f){xm ~v”cojrvmx|!54w`»

Usage:

- typically in military systems; also in commercial ones

---

1 and then tell me about it, because I have forgotten the algorithm!

Secret key's systems

- single key
- two-key

Example:

- Knowing that a variant of "Caesar's cipher"[1] is being used (adapted to Latin 1, ISO 8859-1, table), find the "key" that turns the phrase[2]

  «`Put more code, get more bugs.`»

  into

  «`Sxw#pruh#frgh/#jhw#pruh#exjv1`»

Usage:

- common in many military, commercial and personal applications

1 apparently, the original Caesar's cipher used a simple "3" as key
2 French quotes are just delimiters

Enciphering systems with key

Symmetric, secret key, or shared key

- $K_e = K_d = K$
- very efficient computation; so, very suitable for large amounts of data
- difficult combination and sharing of key, so, preferred for closed environments
- e.g. AES (*Advanced Encryption Standard*)

Asymmetric, public key, or double-key

- $K_e = K^+ \neq K_d = K^-$
- very heavy computation, so, not suitable for large amounts of data
- easy combination and exchange of keys, so, ideal for open environments
- e.g. RSA (Rivest-*Shamir-Adleman*)

Classification of cryptographic systems: *on the method*

Enciphering methods for “long” texts

- Encipher (and decipher) operations have to be done in pieces (blocks)
  - pieces could be of 1 b, 1 B, 8 B,…
    - typical: 8 B (64 b) and 16 B (128 b)
- So, *plaintext P* is divided into parts of equal size:
  - $P = P_1 P_2 ...$
  - each, is separately enciphered (and later deciphered) by one of the methods:
    - stream[1]
    - block
    - “mix” of previous...

Exercise:

- In practice, almost any text is "long". Why?

---

1 PT: contínuo

Stream method

- each part is enciphered with a different key $K = K_1\ K_2...$
- $C = K(P) = K_1(P_1)\ K_2\,(P_2)...$
- Examples: Ronald Rivest's RC4 (ARC4), *one-time pad*

Example:

P:	Put	mo	re	cod	e,	get	mo	re	bug	s.
key:	3	1	5	9	11	2	3	3	8	2
C:	Sxw	!np	wj%	lxm	p7+	igv	#pr	uh#	j}o	u0

Example of cryptographic technique: *One-time Pad*

- stream-type system
- random key (or cryptographically secure pseudo-random...)
- size of key equal to the the original text's
- key used only once
- Advantages: proved unbreakable[1]
- Disadvantages: exercise!

enciphering:

- original text: P
- key: K
- enciphered text: $C = P \oplus K$

deciphering:

- $P = C \oplus K$

Exercises:

- Show that the system is bidirectional. Why is it not very much used?

1 If...

...Classification of cryptographic systems: *on the method*

Block Method

- each part of text is enciphered with the same key K: ECB[1] mode
- $C = K(P) = K(P_1)\ K(P_2)...$
- Examples: *AES, RSA*[2]

Example and exercise (complete the blank boxes):

P:	Put	mo	re	cod	e,	get	mo	re	bug	s.
key:	3	3	3	3	3	3	3	3	3	3
C:	Sxw	#pr	uh#	frg						

1 Electronic Code Book
2 Many texts do not consider RSA to be a block cipher, as it is not efficient to use consecutively (block after block) in long documents.

...Classification of cryptographic systems: *on the method - block (cont.)*

Serious problem:

- with this method, identical blocks give identical codes!
- visual example:

Fig. danger of using the basic block method: plaintext is exposed.

Solutions to the problem:

- mixing additional (and different) information per block!
  - **but** several of the "solutions" are still vulnerable!

1 here, with algorithm AES 256b, ECB

Classification of cryptographic systems: on the *purpose*

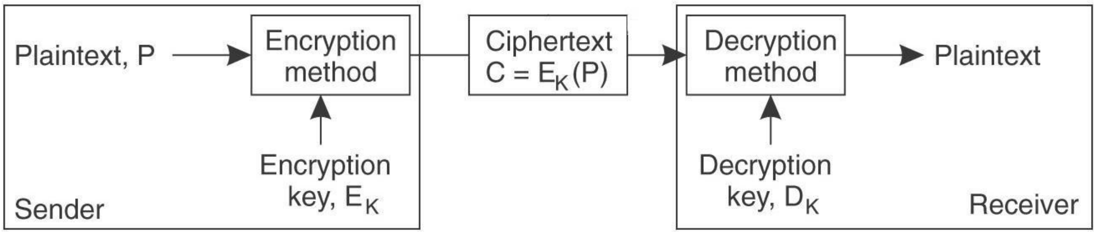
Purpose types

- Confidentiality: bidirectional, reversible (*two-way*)
- Integrity: unidirectional, irreversible (*one-way*)

Reversible (or bidirectional, *two-way*) encipherment:

Usage area

- Confidentiality
- (Authentication)
- (Integrity checking)



(Desired) properties of the bidirectional algorithm:

Simplicity:

- the enciphering of the *plaintext* P (with K_e) is (relatively) easy;
- the deciphering of the *ciphertext* C (with K_d) also is.

Resistance:

- given a *plaintext* P and its ciphered counterpart C, it is impractical to compute the key K, used to produce $C = E_K(P)$

Uniqueness:

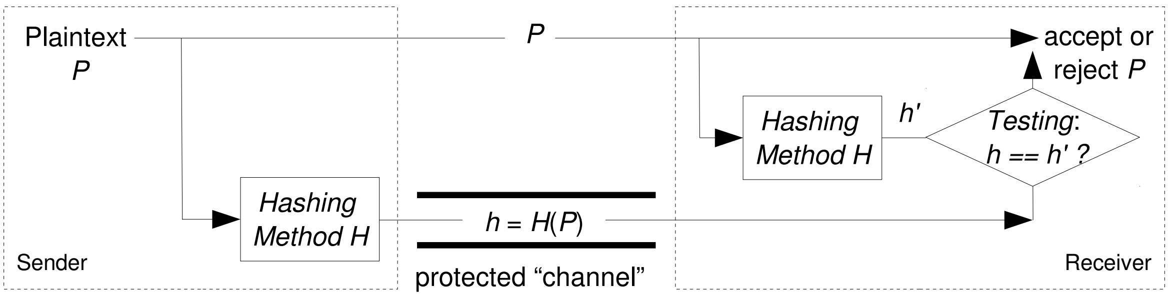
- given a *plaintext* P and a key K, it is impractical to compute another key K' such as $E_K(P) = E_{K'}(P)$

Note: impractical = currently, computationally infeasible

Irreversible (or unidirectional, *one-way*) "encipherment":

Usage area

- Integrity checking
- Authentication



Basic idea:

- from an original text, compute a number that is characteristic of the text:
  - *hash value*, *digest*[1], *fingerprint* [, *checksum*]
- (The original text is not recoverable from the hash!)

Usually,

- a key is not necessary: $C = H(P)$
- the “number" has a fixed length
- the hashing function is somewhat akin to database dispersion functions, but has very different features and purpose

1 PT: sumário

(Desired) properties of the unidirectional algorithm:

Simplicity:

- the encoding of the original text is easy

No reversibility:

- it is impractical to invert the function H: $P \neq H^{-1}(C)$

Uniqueness (or collision resistance):

- it is <u>impractical</u> to find two texts *P1* and *P2* such that $H(P1) = H(P2)$
- Note:
  - a variant[1] of this property, says that, for a given specified text *P1*, it is impractical to find a text *P2* such that $H(P1) = H(P2)$.

1 This variant is even "more impractical" than the first!

Weaknesses of irreversible systems

Problem:

- The number produced by the hashing operation is usually fixed (and finite)!
  - So, there **have to be** collisions, in an infinite universe of inputs[1]
  - Will they be likely or easy to cause?

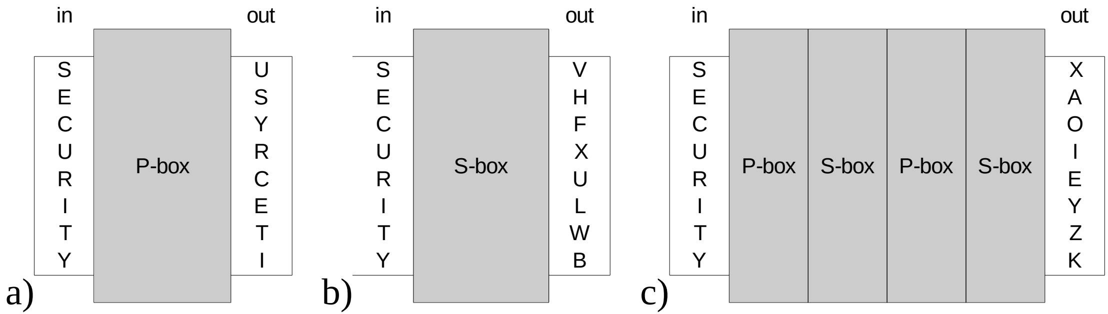
Answer:

- that depends
  - on the randomness of the numbers resulting from the operation
  - on the size of those numbers (number of bits)
  - on the intended application

1 Let us suppose you use a specific hash with 3 decimal digits; the possible values will be: 000, 001, … 567, … 998, 999, a total of 1000 (=10^3) possibilities. If you calculate the hash of 1001 documents, two of then will have the same hash – a collision!

Cryptographic transformations

- Transposition – exchange or swapping of positions of elements – *P-box*
- Substitution – exchange of elements (e.g. Caeser's cipher) – *S-box*
- Combination - transposition and substitution cascade – *product cipher*



Cryptographic transformations: a) permutation box; b) substitution box; “complete” system.
Exercise: find out the algorithms for P- and S- boxes and validate them with c).

Some famous cryptographic algorithms

- RC4: stream key generation (1987, survives with medication)
- DES[1]: reversible system, secret key (1975, defunct)
- AES: reversible system, secret key (1998, still healthy)
- RSA[2]: reversible system, public key (1977, still healthy)
- MD5[3]: irreversible system (1992, defunct)
- SHA-1[4]: irreversible system (1995, defunct)
- SHA-2: irreversible system (2001, still healthy)
- SHA-3[5]: irreversible system (2015, yet in phase of wide adoption)

1 Data Encryption Standard, a landmark of cryptography
2 another landmark of (public-key) cryptography
3 yet another landmark of cryptography
4 about SHA-1 end of life, see sha-mbles.github.io
5 based on new paradigm - sponge construction (keccak.team/sponge_duplex.html)

Case study (simplified): RSA (*Rivest-Shamir-Adleman*)

- reversible, public key system
- published in 1977; protected by patent in USA until 2000
- unbroken, so far:[1]
    - if keys are well chosen (e.g. made with very large prime numbers, > 10^{150})
    - if *padding* of *P* receives appropriate consideration!
- base of operation:
    - modular arithmetic
    - *P* and *C* will be used as numbers!

1 before the advent of serious quantum computers

...Case study: RSA (cont.)

(Very simple) example of using of RSA:

- $K^+ = K_e = (3, 33)$; $K^- = K_d = (7, 33)$;
- P = "S U Z A N N E" : P_1 = “S” = 19 (< 33) ; P_2 = “U” = 21 (< 33) ...
- C = "28 21 20 1 5 5 26" : C_1 = 28 (< 33) ; C_2 = 21 (< 33) ...

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

Sender's computation (Symbolic through P^3 (mod 33)) — Receiver's computation (P^3 (mod 33) through Symbolic)

Exercise: Redo the demo, but now replacing each letter by its ASCII equivalent.

...Case study: RSA (cont.)

Ciphering with RSA [1]

- Key: $K^{+} = K_{e} = (e, n)$
- $C = P^{e} \pmod{n}$

Deciphering with RSA

- Key: $K^{-} = K_{d} = (d, n)$
  - with $e.d = 1 \pmod{\phi(n)}$ [2]
- $Q = C^{d} \pmod{n} = (P^{e} \pmod{n})^{d} = P^{ed} \pmod{n} = P$
  $Q = P$!

Notes:

- n is a very large prime number (e.g. $\approx 10^{300} \approx 2^{1024}$)
- e is usually much smaller than d (typical values of e: 3, 65537...)

---

1 Note that C and P are interpreted as (binary) numbers; the exponentiation operation is the usual, e.g. $a^3 = a \times a \times a$.
2 ϕ is a well known mathematical function.

Some numbers...

- $2^{8} = 256$ number of values represented by a byte
- $2^{32} = 4\ 294\ 967\ 296$ maximum number of IPv4 addresses
  $\simeq$ 0,5 * number of people on Earth in 2023
- $2^{56} = 72\ 057\ 594\ 037\ 927\ 936$ number of different keys for DES algorithm
- $2^{64} = 18\ 446\ 744\ 073\ 709\ 551\ 616$
  1+ number of grains of wheat in chess board (from 1, doubled in each square)
- $2^{76} \simeq 10^{23}$ mass of the Moon in kg
- $2^{79} \simeq 10^{24}$ Avogadro's constant
- $2^{82} \simeq 10^{25}$ mass of the Earth in kg
- $2^{101} \simeq 10^{30}$ mass of the Sun in kg
- $2^{128} = 340\ 282\ 366\ 920\ 938\ 463\ 463\ 374\ 607\ 431\ 768\ 211\ 456$
  $\simeq 10^{38}$ maximum number of IPv6 addresses
- $2^{256} \simeq 10^{77}$ number of values of SHA-256 hash
- $2^{280} \simeq 10^{84}$ number of fundamental particles in the observable universe