Cryptography

Cryptography: terminology (1/2)

- Cryptography
  - Art or science of hidden writing
    - from Gr. *kryptós*, hidden + *graph*, r. of *graphein*, to write
  - It was initially used to maintain the confidentiality of information
  - Steganography
    - from Gr. *steganós*, hidden + *graph*, r. of *graphein*, to write
- Cryptanalysis
  - Art or science of breaking cryptographic systems or encrypted information
- Cryptology
  - Cryptography + cryptanalysis

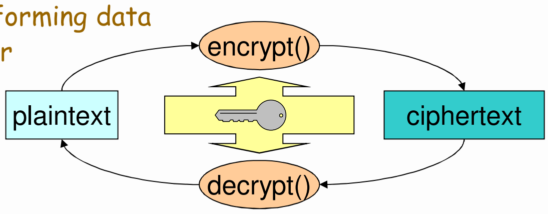
Cryptography: terminology (2/2)

- Cipher
  - Specific cryptographic technique

- Cipher operation

  Encryption: plaintext (or cleartext) → ciphertext (or cryptogram)
  Decryption: ciphertext → plaintext

  Algorithm: way of transforming data
  Key: algorithm parameter



Use cases (symmetric cryptography)

- Self-protection with key K
  - Alice encrypts plaintext P with key K

    A: $C = \{P\}_K$
  - Alice decrypts cryptogram C with key K

    A: $P' = \{C\}_K$
  - P' should be equal to P (requires checking)

- Secure communication with key K
  - Alice encrypts plaintext P with key K

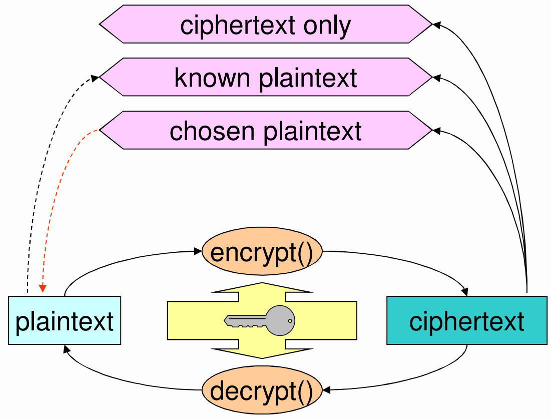
    A: $C = \{P\}_K$
  - Bob decrypts C with key K

    B: $P' = \{C\}_K$
  - P' should be equal to P (requires checking)

Cryptanalysis: goals

- Discover original plaintext
  - Which originated a given ciphertext
- Discover a cipher key
  - Allows the decryption of ciphertexts created with the same key
- Discover the cipher algorithm
  - Or an equivalent algorithm
  - Usually algorithms are not secret, but there are exceptions
    - Lorenz, A5 (GSM), RC4 (WEP), Crypto-1 (Mifare)
    - Algorithms for DRM (Digital Rights Management)
  - Reverse engineering

Cryptanalysis attacks: approaches



ciphertext only

known plaintext

chosen plaintext

encrypt()

plaintext

ciphertext

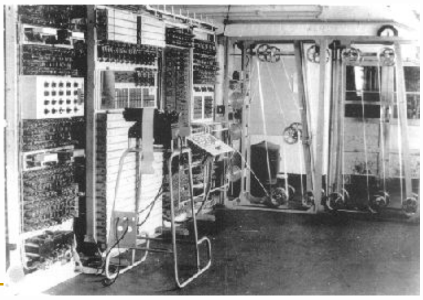
decrypt()

Cryptanalysis attacks: approaches

- Brute force
  - Exhaustive search along the key space until finding a suitable key
  - Usually infeasible for a large key space
    - e.g. 2^{128} random keys (or keys with 128 bits)
    - Randomness is fundamental!
- Cleaver attacks
  - Reduce the search space to a smaller set of potential candidates

Ciphers: evolution of technology

- Manual
  - Simple transposition or substitution algorithms
- Mechanic
  - From XIX cent.
    - Enigma machine
    - M-209 Converter
  - More complex substitution algorithms
- Informatics
  - Appear with computers
  - Highly complex substitution algorithms
  - Mathematical algorithms

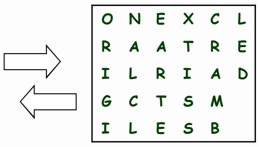


Ciphers: basic types (1/3)

- Transposition
  - Original cleartext is scrambled

    `Onexcl raatre ilriad gctsm ilesb`
  - Block permutations

    (13524) → `boklc pruem ttoai ns`
- Substitution
  - Each original symbol is replaced by another
    - Original symbols were letters, digits and punctuation
    - Actually they are blocks of bits
  - Substitution strategies
    - Mono-alphabetic (one→one)
    - Polyalphabetic (many one→one)
    - Homophonic (one→many)



O	N	E	X	C	L
R	A	A	T	R	E
I	L	R	I	A	D
G	C	T	S	M	
I	L	E	S	B	

Ciphers: basic types (2/3): Mono-alphabetic

- Use a single substitution alphabet
  - With #α elements
- Examples
  - Additive (translation)
    - crypto-symbol = (symbol + key) mod # α
    - symbol = (crypto-symbol – key) mod # α
    - Possible keys = #α
    - Caesar Cipher (ROT-x)
  - With sentence key

    ABCDEFGHIJKLMNOPQRSTUVWXYZ

    QRUVWXZSENTCKYABDFGHIJLMOP
    - Possible keys = # α ! → 26! ≈ 2^{88}
- Problems
  - Reproduce plaintext pattern
    - Individual characters, digrams, trigrams, etc.
  - Statistical analysis facilitates cryptanalysis
    - "The Gold Bug", Edgar Alan Poe

```
53‡‡†305))6*;4826)4‡.)
4‡);806*;48†860))85;1‡
(;:‡*8†83(88)5*†;46(;8
8*96*?;8)*‡(;485);5*†2
:*‡(;4956*2(5*—4)88*;4
069285);)6†8)4‡‡;1(‡9;
48081;8:8‡1;48†85;4)48
5†528806*81(‡9;48;(88;
4(‡?34;48)4‡;161;:188;
‡?;
```

```
A good glass in the
bishop's hostel in the
devil's seat fifty-one
degrees and thirteen
minutes northeast and
by north main branch
seventh limb east side
shoot from the left eye
of the death's-head a
bee line from the tree
through the shot forty
feet out
```

Ciphers: basic types (3/3): Polyalphabetic

- Use N substitution alphabets
  - Periodical ciphers, with period N
- Example
  - Vigenère cipher
- Problems
  - Once known the period, are as easy to cryptanalyze as N mono-alphabetic ones
    - The period can be discovered using statistics
    - Kasiski method
      - Factoring of distances between equal ciphertext blocks
    - Coincidence index
      - Factoring of self-correlation offsets that yield higher coincidences

Vigenère cipher (or the Vigenère square)

	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 of encryption of character M with key S, yielding cryptogram E
  - Decryption is the opposite, E and S yield M

Cryptanalysis of a Vigenère cryptogram: Example (1/2)

- Plaintext:

  Eles não sabem que o sonho é uma constante da vida
  tão concreta e definida como outra coisa qualquer,
  como esta pedra cinzenta em que me sento e descanso,
  como este ribeiro manso, em serenos sobressaltos
  como estes pinheiros altos

- Cipher with the Vigenère square and key “poema”

```
plaintext  elesnaosabemqueosonhoeumaconstantedavidataoconcretaedefinida
key        poemapoemapoemapoemapoemapoemapoemapoemapoemapoemapoemapoema
cryptogram tzienpcwmbtaugedgszhdsyyarcretpbxqdpjmpaiosoocqvqtpshqfxbmpa
```

- Kasiski test
  - With text above:

mpa	$20 = 2\times\ 2\times 5$
tp	$20 = 2\times\ 2\times 5$

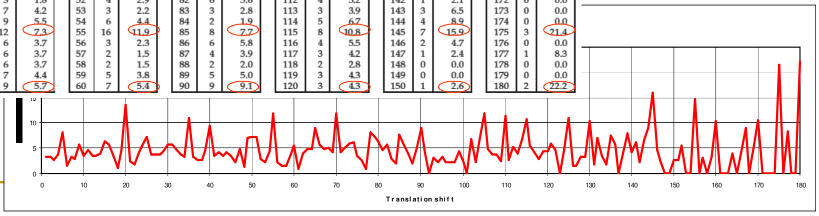
  - With the complete poem:

$175 = 5\times\ 5\times 7$	1
$105 = 3\times\ 5\times 7$	3
$35 = 5\times 7$	1
$20 = 2\times 2\times 5$	4

Cryptanalysis of a Vigenère cryptogram: Example (2/2)

- Coincidence index (with full poem)

D	I	P (%)
1	6	3.2
2	6	3.2
3	5	2.7
4	7	3.8
5	15	8.2
6	3	1.6
7	6	3.3
8	5	2.8
9	10	5.6
10	6	3.4
11	8	4.5
12	6	3.4
13	6	3.4
14	7	4.0
15	11	6.3
16	10	5.8
17	6	3.5
18	2	1.2
19	8	4.7
20	23	13.6
21	4	2.4
22	3	1.8
23	7	4.2
24	9	5.5
25	12	7.3
26	6	3.7
27	6	3.7
28	6	3.7
29	7	4.4
30	9	5.7
31	9	5.7
32	7	4.5
33	6	3.8
34	5	3.2
35	17	11.0
36	5	3.3
37	4	2.6
38	4	2.6
39	7	4.7
40	14	9.4
41	5	3.4
42	6	4.1
43	5	3.4
44	6	4.1
45	5	3.5
46	3	2.1
47	7	4.9
48	2	1.4
49	10	7.1
50	10	7.2
51	10	7.2
52	4	2.9
53	3	2.2
54	6	4.4
55	16	11.9
56	3	2.3
57	2	1.5
58	2	1.5
59	5	3.8
60	7	5.4
61	1	0.8
62	5	3.9
63	6	4.8
64	6	4.8
65	11	8.9
66	7	5.7
67	6	4.9
68	6	5.0
69	5	4.2
70	14	11.8
71	5	4.2
72	6	5.1
73	7	6.0
74	7	6.1
75	4	3.5
76	3	2.7
77	1	0.9
78	9	8.1
79	8	7.3
80	7	6.4
81	5	4.6
82	6	5.6
83	3	2.8
84	2	1.9
85	8	7.7
86	6	5.8
87	4	3.9
88	2	2.0
89	5	5.0
90	9	9.1
91	4	4.1
92	0	0.0
93	3	3.1
94	2	2.1
95	3	3.2
96	2	2.2
97	2	2.2
98	2	2.2
99	4	4.4
100	2	2.2
101	0	0.0
102	6	6.9
103	2	2.3
104	6	7.1
105	10	11.9
106	4	4.8
107	3	3.7
108	3	3.7
109	2	2.5
110	9	11.4
111	2	2.6
112	4	5.2
113	3	3.9
114	5	6.7
115	8	10.8
116	4	5.5
117	3	4.2
118	2	2.8
119	3	4.3
120	3	4.3
121	4	5.9
122	3	4.5
123	0	0.0
124	3	4.6
125	7	10.9
126	1	1.6
127	1	1.6
128	2	3.3
129	2	3.3
130	6	10.2
131	1	1.7
132	4	7.0
133	2	3.6
134	1	1.8
135	4	7.4
136	3	5.7
137	0	0.0
138	2	3.9
139	4	8.0
140	2	4.1
141	3	6.2
142	1	2.1
143	3	6.5
144	4	8.9
145	7	15.9
146	2	4.7
147	1	2.4
148	0	0.0
149	0	0.0
150	1	2.6
151	1	2.6
152	2	5.4
153	0	0.0
154	0	0.0
155	5	14.7
156	0	0.0
157	1	3.1
158	0	0.0
159	1	3.3
160	3	10.3
161	0	0.0
162	0	0.0
163	0	0.0
164	1	4.0
165	0	0.0
166	1	4.3
167	2	9.1
168	0	0.0
169	1	5.0
170	2	10.5
171	0	0.0
172	0	0.0
173	0	0.0
174	0	0.0
175	3	21.4
176	0	0.0
177	1	8.3
178	0	0.0
179	0	0.0
180	2	22.2



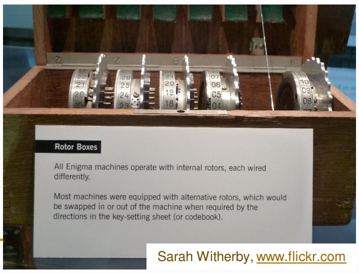
Rotor Machines (1/3)

THE ENIGMA MACHINE ROTOR
A DISMANTLED ROTOR SHOWING THE WIRING BETWEEN THE CONTACT STUDS AND THE SPRING LOADED CONTACTS

David J Morgan, www.flickr.com

Rotor machines (2/3)

- Rotor machines implement complex polyalphabetic ciphers
  - Each rotor contains a permutation
    - Same as a set of substitutions
  - The position of a rotor implements a substitution alphabet
  - Spinning of a rotor implements a polyalphabetic cipher
  - Stacking several rotors and spinning them at different times adds complexity to the cipher
- The cipher key is:
  - The set of rotors used
  - The relative order of the rotors
  - The position of the spinning ring
  - The original position of all the rotors
- Symmetrical (two-way) rotors allow decryption by "double encryption"
  - Using a reflection disk (half-rotor)


Rotor Boxes

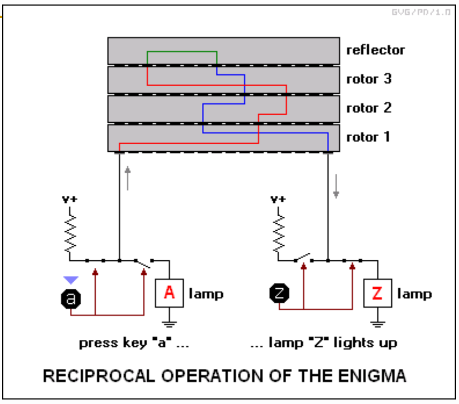
All Enigma machines operate with internal rotors, each wired differently.

Most machines were equipped with alternative rotors, which would be swapped in or out of the machine when required by the directions in the key-setting sheet (or codebook).

Sarah Witherby, www.flickr.com

Rotor machines (3/3)

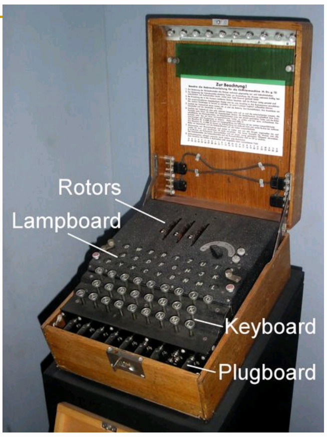
Andrew Magill, www.flickr.com



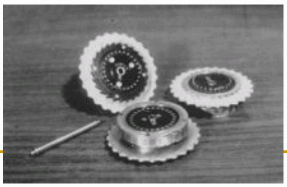
RECIPROCAL OPERATION OF THE ENIGMA

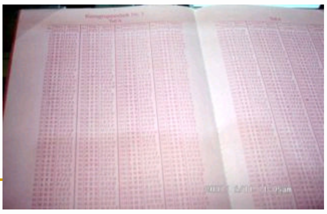
- Reciprocal operation with reflector
  - Sending operator types "A" as plaintext and gets "Z" as ciphertext, which is transmitted
  - Receiving operator types the received "Z" and gets the plaintext "A"
  - No letter could encrypt to itself !

Enigma



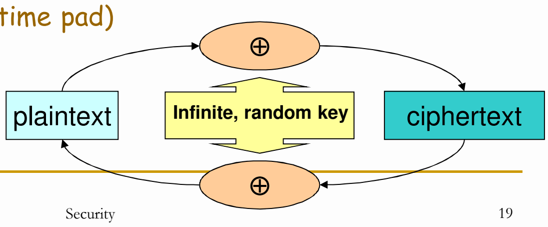
- WWII German rotor machine
  - Many models used
- Initially presented in 1919
  - Enigma I, with 3 rotors
- Several variants where used
  - With different number of rotors
  - With patch cord to permute alphabets
- Key settings distributed in codebooks





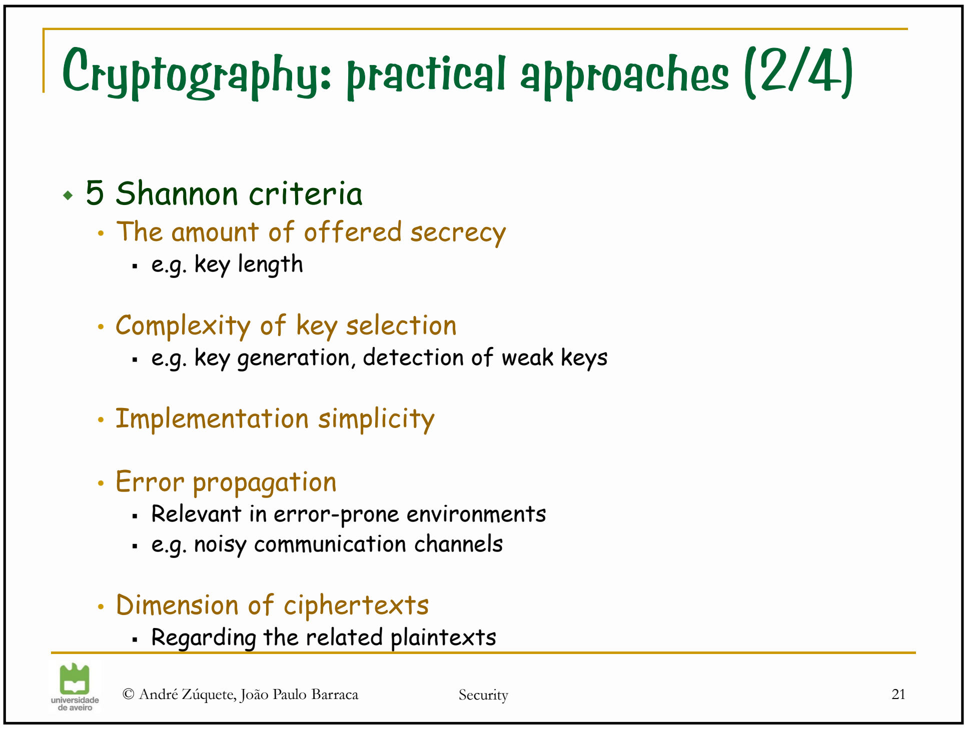
Cryptography: theoretical analysis

- Plaintext space
  - Set of all possible plaintext messages (*M*)
- Ciphertext space
  - Set of all possible ciphertext values (*C*)
- Key space
  - Set of all possible key values for a given algorithm (*K*)
- Perfect (information-theoretical) security
  - Given $c_j \in C$, $p(m_i, k_j) = p(m_i)$
  - $\#K \geq \#M$
  - Vernam cipher (one-time pad)



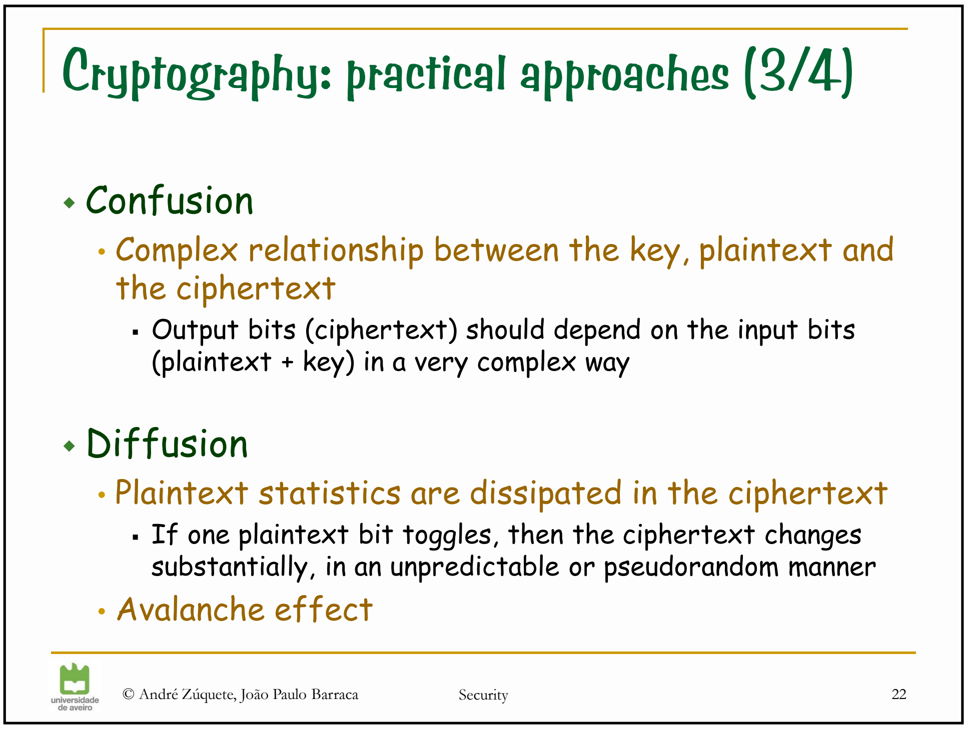
Cryptography: practical approaches (1/4)

- Theoretical security vs. practical security
  - Expected use ≠ practical exploitation
  - Defective practices can introduce vulnerabilities
    - Example: re-use of one-time pad key blocks
- Computational security
  - Security is measured by the computational complexity of break-in attacks
    - Using brute force
  - Security bounds:
    - Cost of cryptanalysis
    - Availability of cryptanalysis infra-structure
    - Lifetime of ciphertext



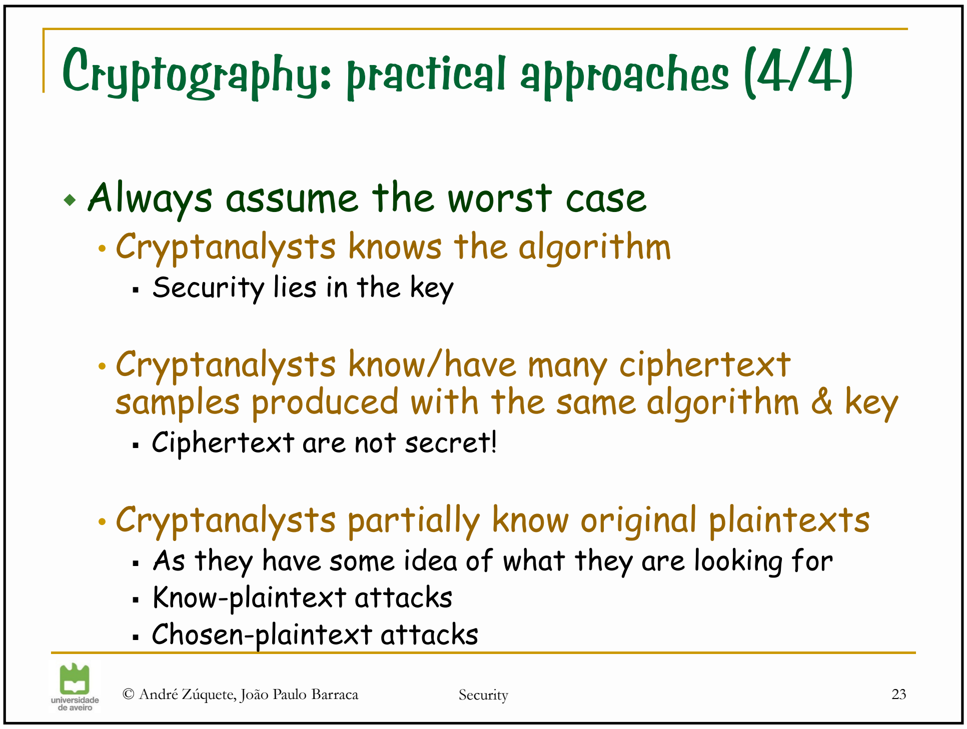
Cryptography: practical approaches (2/4)

- 5 Shannon criteria
  - The amount of offered secrecy
    - e.g. key length
  - Complexity of key selection
    - e.g. key generation, detection of weak keys
  - Implementation simplicity
  - Error propagation
    - Relevant in error-prone environments
    - e.g. noisy communication channels
  - Dimension of ciphertexts
    - Regarding the related plaintexts



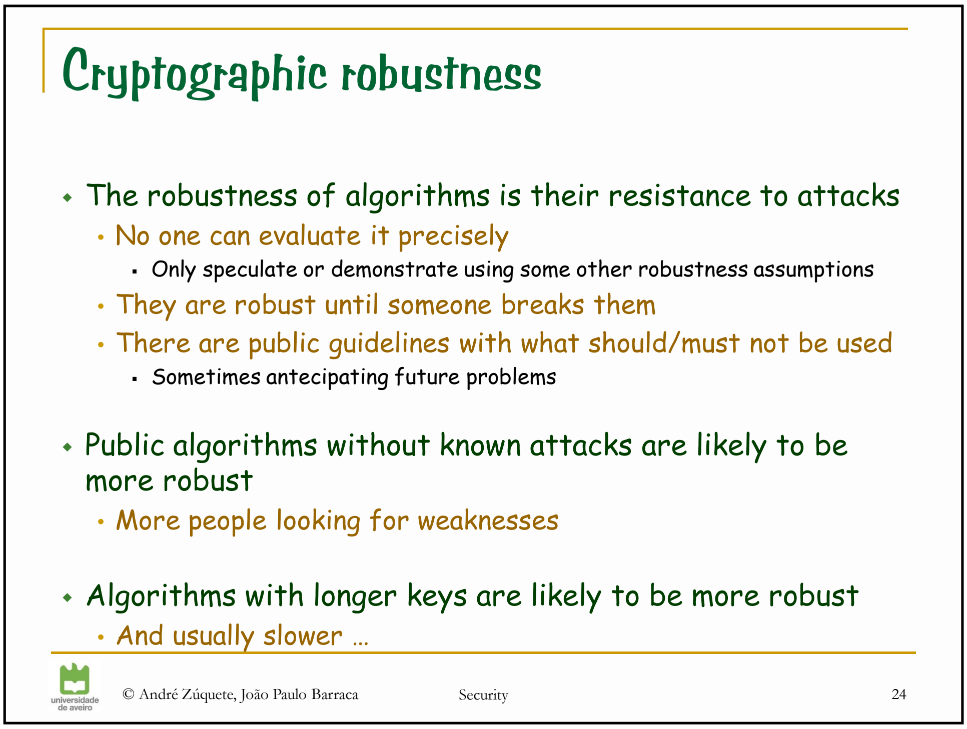
Cryptography: practical approaches (3/4)

- Confusion
  - Complex relationship between the key, plaintext and the ciphertext
    - Output bits (ciphertext) should depend on the input bits (plaintext + key) in a very complex way
- Diffusion
  - Plaintext statistics are dissipated in the ciphertext
    - If one plaintext bit toggles, then the ciphertext changes substantially, in an unpredictable or pseudorandom manner
  - Avalanche effect



Cryptography: practical approaches (4/4)

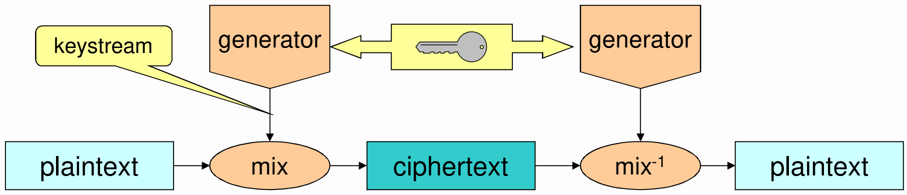
- Always assume the worst case
  - Cryptanalysts knows the algorithm
    - Security lies in the key
  - Cryptanalysts know/have many ciphertext samples produced with the same algorithm & key
    - Ciphertext are not secret!
  - Cryptanalysts partially know original plaintexts
    - As they have some idea of what they are looking for
    - Know-plaintext attacks
    - Chosen-plaintext attacks



Cryptographic robustness

- The robustness of algorithms is their resistance to attacks
  - No one can evaluate it precisely
    - Only speculate or demonstrate using some other robustness assumptions
  - They are robust until someone breaks them
  - There are public guidelines with what should/must not be used
    - Sometimes antecipating future problems
- Public algorithms without known attacks are likely to be more robust
  - More people looking for weaknesses
- Algorithms with longer keys are likely to be more robust
  - And usually slower …

Stream ciphers (1/2)



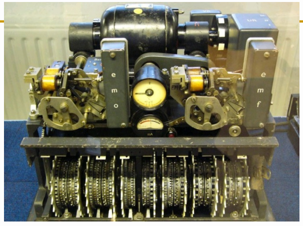
- Mixture of a keystream with the plaintext or ciphertext
  - Random keystream (Vernam's one-time pad)
  - Pseudo-random keystream (produced by generator using a finite key)
- Reversible mixture function
  - e.g. bitwise XOR
  - $C = P \oplus ks$ $P = C \oplus ks$
- Polyalphabetic cipher
  - Each keystream symbol defines an alphabet

Stream ciphers (2/2)

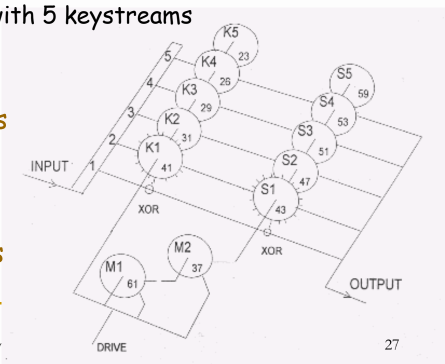
- Keystream may be infinite but with a finite period
  - The period depends on the generator
- Practical security issues
  - Each **keystream** should be used only **once**!
    - Otherwise, the sum of cryptograms yields the sum of plaintexts

      $C1 = P1 \oplus Ks$, $C2 = P2 \oplus Ks$ → $C1 \oplus C2 = P1 \oplus P2$
  - **Plaintext length** should be **smaller** than the **keystream period**
    - Keystream exposure is total under know/chosen plaintext attacks
    - Keystream cycles help the cryptanalysts knowing plaintext samples
  - Integrity control is mandatory
    - No diffusion! (only confusion)
    - Ciphertexts can easily be changed deterministically

Lorenz (Tunny)

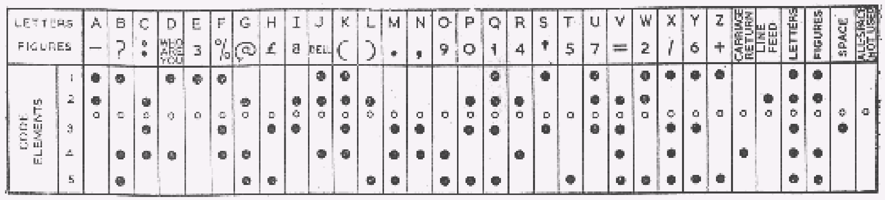


- 12-Rotor stream cipher
  - Used by the German high-command during the 2nd WW
  - Implements a stream cipher
    - Each 5-bit character is mixed with 5 keystreams
- Operation
  - 5 regularly stepped (χ) wheels
  - 5 irregularly stepped (ψ) wheels
    - All or none stepping
  - 2 motor wheels
    - For stepping the ψ wheels
  - Number of steps in all wheels is relatively prime



Cryptanalysis of Tunny in Bletchley Park (1/4)

- They didn't know Lorenz internal structure
  - They observed one only at the end of the war
  - They knew about them because they could get 5-bit encrypted transmissions
    - Using the 32-symbol Baudot code instead of Morse code



LETTERS		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	CARRIAGE RETURN	LINE FEED	LETTERS	FIGURES	SPACE	ALL-SPACES NOT USED
FIGURES		—	?	:	WHO ARE YOU	3	%	@	£	8	BELL	(	)	.	,	9	0	1	4	'	5	7	=	2	/	6	+						
CODE ELEMENTS	1	●	●		●	●	●				●	●						●		●		●		●	●	●	●			●	●		
	2	●		●				●		●	●	●	●				●	●	●			●	●	●					●	●	●		
		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	3			●			●		●	●		●		●	●		●	●		●		●	●		●	●				●		●	
	4		●	●	●		●	●			●	●		●	●	●			●				●		●			●		●	●		
	5		●					●	●				●	●		●	●	●			●		●	●	●	●	●			●	●		

Cryptanalysis of Tunny in Bletchley Park (2/4)

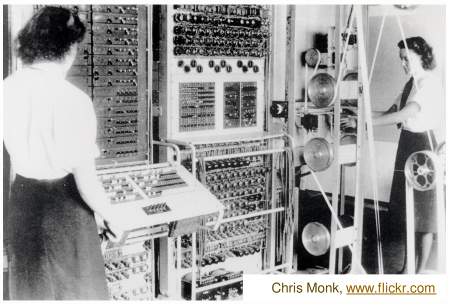
- The mistake (30 August 1941)
  - A German operator had a long message (~4,000) to send
    - He set up his Lorenz and sent a 12 letter indicator (wheel setup) to the receiver
    - After ~4,000 characters had been keyed, by hand, the receiver said "send it again"
  - The operator resets the machine to the same initial setup
    - Same keystream! Absolutely forbidden!
  - The sender began to key in the message again (by hand)
    - But he typed a slightly different message!
    - $C = M \oplus Ks$
    - $C' = M' \oplus Ks$ → $M' = C \oplus C' \oplus M$ → text variations
    - If you know part of the initial text, you can find the variations

Cryptanalysis of Tunny in Bletchley Park (3/4)

- Breakthrough
  - Messages began with a well known SPRUCHNUMMER — "msg number"
    - The first time the operator keyed in S P R U C H N U M M E R
    - The second time he keyed in S P R U C H N R
    - Thus, immediately following the N the two texts were different!
  - Both messages were sent to John Tiltman at Bletchley Park, which was able to fully decrypt them using an additive combination of the messages (called *Depths*)
    - The 2nd message was ~500 characters shorter than the first one
    - Tiltman managed to discover the correct message for the 1st ciphertext
  - They got for the 1st time a long stretch of the Lorenz keystream
    - They did not know how the machine did it, …
    - … but they knew that this was what it was generating!

Cryptanalysis of Tunny in Bletchley Park (4/4): Colossus

- The cipher structure was determined from the keystream
  - But deciphering it required knowing the initial position of rotors
- Germans started using numbers for the initial wheels' state
  - Bill Tutte invented the double-delta method for finding that state
  - The Colossus was built to apply the double-delta method
- Colossus
  - Design started in March 1943
  - The 1,500 valve Colossus Mark 1 was operational in January 1944
  - Colossus reduced the time to break Lorenz from weeks to hours

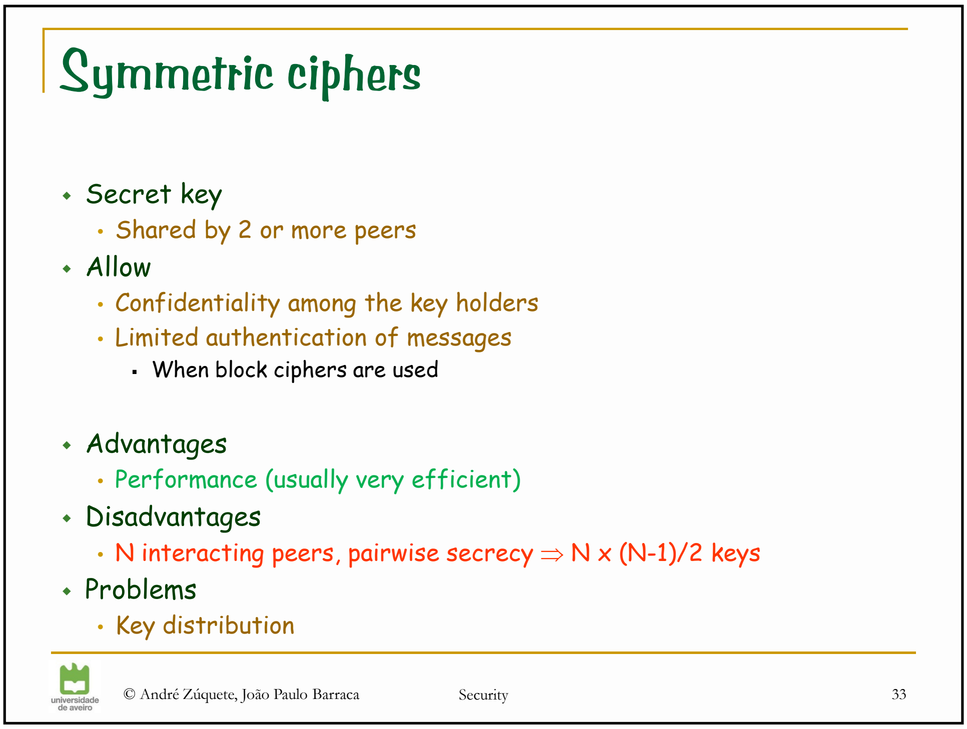


Chris Monk, www.flickr.com

Modern ciphers: types

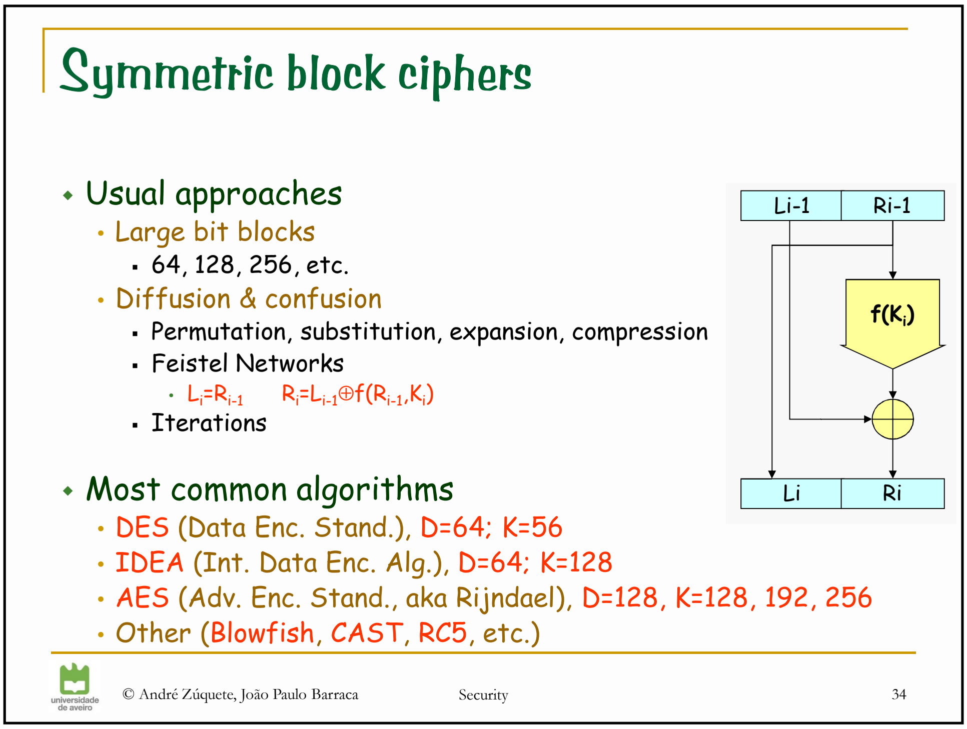
- Concerning operation
  - Block ciphers (mono-alphabetic)
  - Stream ciphers (polyalphabetic)
- Concerning their key
  - Symmetric ciphers (secret key or shared key ciphers)
  - Asymmetric ciphers (or public key ciphers)
- Arrangements

	Block ciphers	Stream ciphers
Symmetric ciphers	🟩	🟩
Asymmetric ciphers	🟩	🟥



Symmetric ciphers

- Secret key
  - Shared by 2 or more peers
- Allow
  - Confidentiality among the key holders
  - Limited authentication of messages
    - When block ciphers are used
- Advantages
  - Performance (usually very efficient)
- Disadvantages
  - N interacting peers, pairwise secrecy ⇒ N x (N-1)/2 keys
- Problems
  - Key distribution



Symmetric block ciphers

- Usual approaches
  - Large bit blocks
    - 64, 128, 256, etc.
  - Diffusion & confusion
    - Permutation, substitution, expansion, compression
    - Feistel Networks
      - $L_i=R_{i-1}$ $R_i=L_{i-1} \oplus f(R_{i-1},K_i)$
    - Iterations
- Most common algorithms
  - DES (Data Enc. Stand.), D=64; K=56
  - IDEA (Int. Data Enc. Alg.), D=64; K=128
  - AES (Adv. Enc. Stand., aka Rijndael), D=128, K=128, 192, 256
  - Other (Blowfish, CAST, RC5, etc.)

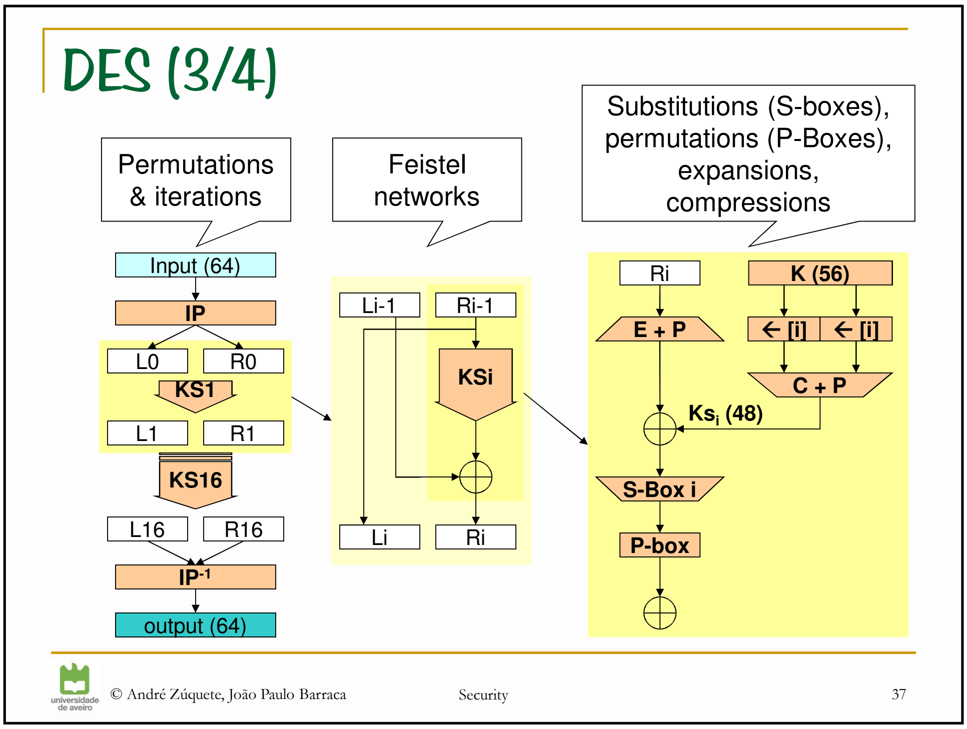
Li-1
Ri-1
f(K$_i$)
Li
Ri

DES (Data Encryption Standard) (1/4)

- 1970: the need of a standard cipher for civilians was identified
- 1972: NBS opens a contest for a new cipher, requiring:
  - The cryptographic algorithm must be secure to a high degree
  - Algorithm details described in an easy-to-understand language
  - The details of the algorithm must be publicly available
    - So that anyone could implement it in software or hardware
  - The security of the algorithm must depend on the key
    - Not on keeping the method itself (or part of it) secret
  - The method must be adaptable for use in many applications
  - Hardware implementations of the algorithm must be practical
    - i.e. not prohibitively expensive or extremely slow
  - The method must be efficient
  - Test and validation under real-life conditions
  - The algorithm should be exportable

DES (2/4)

- 1974: new contest
  - Proposal based on Lucifer from IBM
  - 64-bit blocks
  - 56-bit keys
    - 48-bit subkeys (key schedules)
  - Diffusion & confusion
    - Feistel networks
    - Permutations, substitutions, expansions, compressions
    - 16 iterations
  - Several modes of operation
    - **ECB** (Electronic Code Book), **CBC** (Cypher Block Chaining)
    - **OFB** (Output Feedback), **CFB** (Cypher Feedback)
- 1976: adopted at US as a federal standard

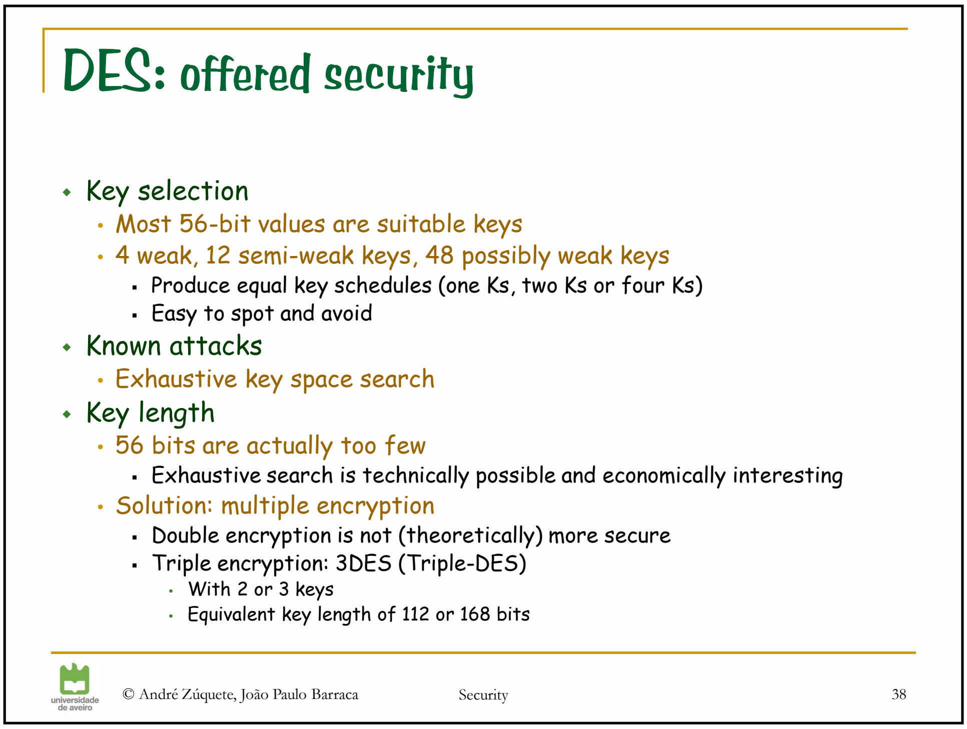


DES (3/4)

Permutations & iterations

Feistel networks

Substitutions (S-boxes), permutations (P-Boxes), expansions, compressions



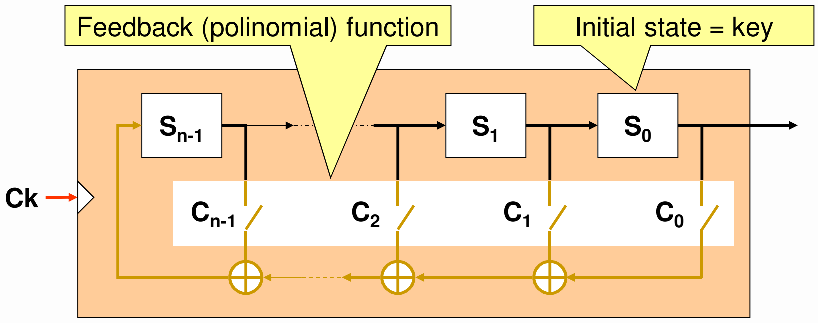
DES: offered security

- Key selection
  - Most 56-bit values are suitable keys
  - 4 weak, 12 semi-weak keys, 48 possibly weak keys
    - Produce equal key schedules (one Ks, two Ks or four Ks)
    - Easy to spot and avoid
- Known attacks
  - Exhaustive key space search
- Key length
  - 56 bits are actually too few
    - Exhaustive search is technically possible and economically interesting
  - Solution: multiple encryption
    - Double encryption is not (theoretically) more secure
    - Triple encryption: 3DES (Triple-DES)
      - With 2 or 3 keys
      - Equivalent key length of 112 or 168 bits

(Symmetric) stream ciphers

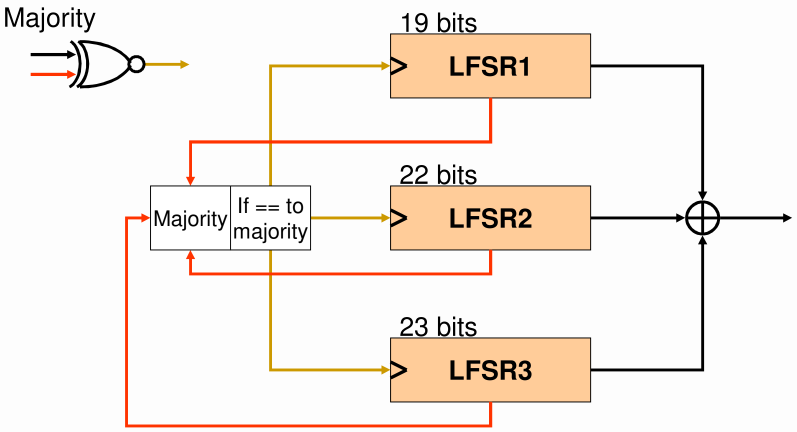
- Approaches
  - Cryptographically secure pseudo-random generators (PRNG)
    - Using linear feedback shift registers (LFSR)
    - Using block ciphers
    - Other (families of functions, etc.)
  - Usually not self-synchronized
  - Usually without uniform random access
- Most common algorithms
  - A5/1 (US, Europe), A5/2 (GSM)
  - RC4 (802.11 WEP/TKIP, etc.)
  - E0 (Bluetooth BR/EDR)
  - SEAL (w/ uniform random access)

Linear Feedback Shift Register (LFSR)



- 2^n-1 non-null sequences
  - If one of them has a 2^n-1 period length, then all have it
- Primitive feedback functions (primitive polynomials)
  - All non-null sequences have a 2^n–1 period length

Generators using many LFSR: A5/1 (GSM)



Deployment of (symmetric) block ciphers: Cipher modes

- Initially proposed for DES
  - ECB (Electronic Code Book)
  - CBC (Cipher Block Chaining)
  - OFB (Output Feeback)
  - CFB (Cipher Feedback)
- Can be used with other block ciphers
  - In principle ...
- Some other modes do exist
  - CTR (Counter Mode)
  - GCM (Galois/Counter Mode)

Block cipher modes: ECB and CBC

Electronic Code Book

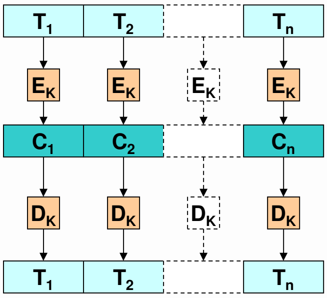
$C_i = E_K(T_i)$

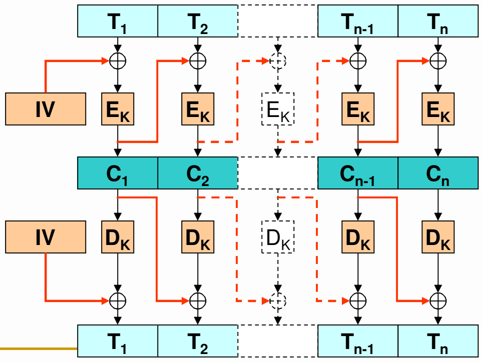
$T_i = D_K(C_i)$

Cipher Block Chaining

$C_i = E_K(T_i \oplus C_{i-1})$

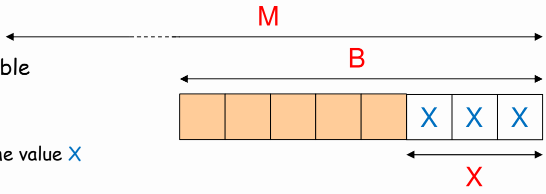
$T_i = D_K(C_i) \oplus C_{i-1}$

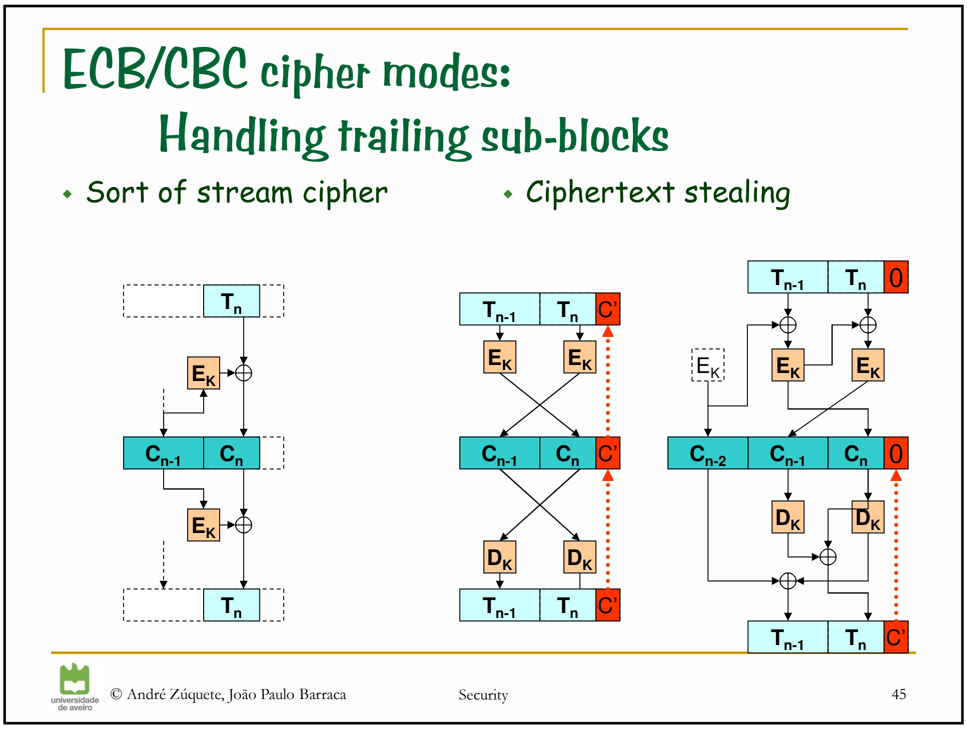




ECB/CBC cipher modes: Trailing sub-block issues

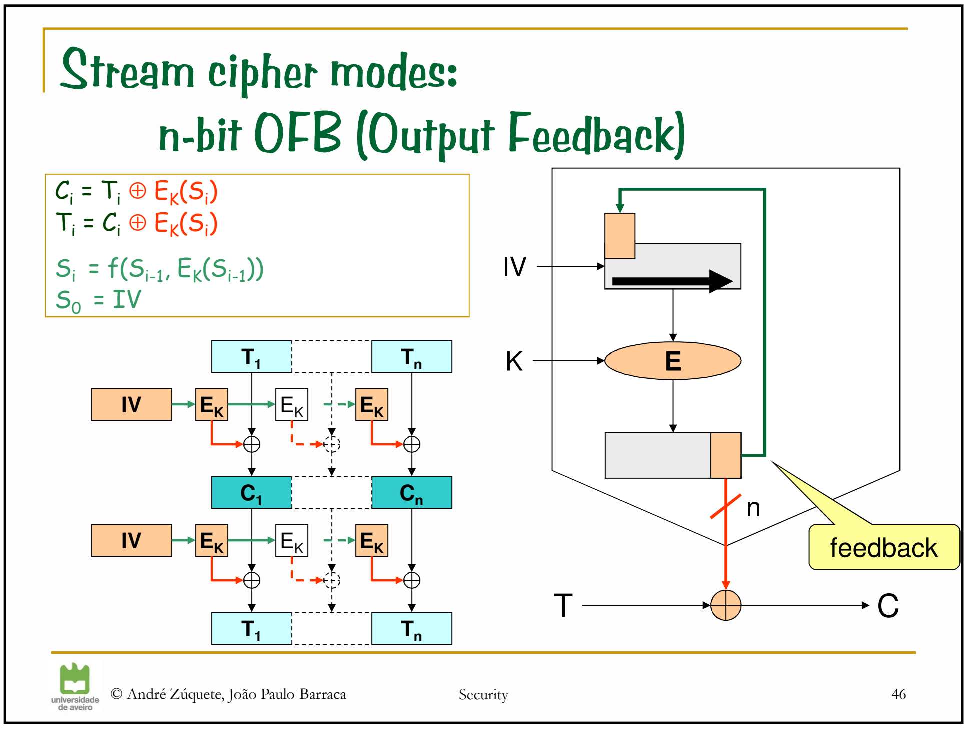
- Block cipher modes ECB and CBC require block-aligned inputs
  - Trailing sub-blocks need special treatment
- Alternatives
  - Padding
    - Of last block, identifiable
    - PKCS #7
      - X = B – (M mod B)
      - X extra bytes, with the value X
    - PKCS #5
      - Equal to PKCS #7 with B = 8
  - Different processing for the last block
    - Adds complexity





ECB/CBC cipher modes: Handling trailing sub-blocks

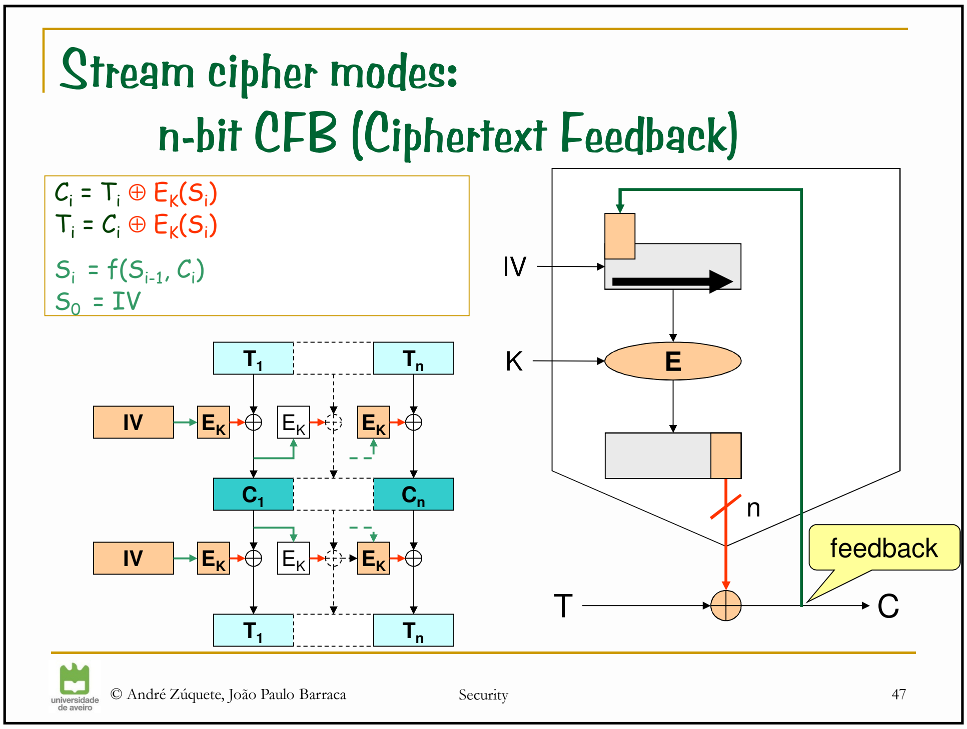
- Sort of stream cipher
- Ciphertext stealing



Stream cipher modes: n-bit OFB (Output Feedback)

$C_i = T_i \oplus E_K(S_i)$
$T_i = C_i \oplus E_K(S_i)$

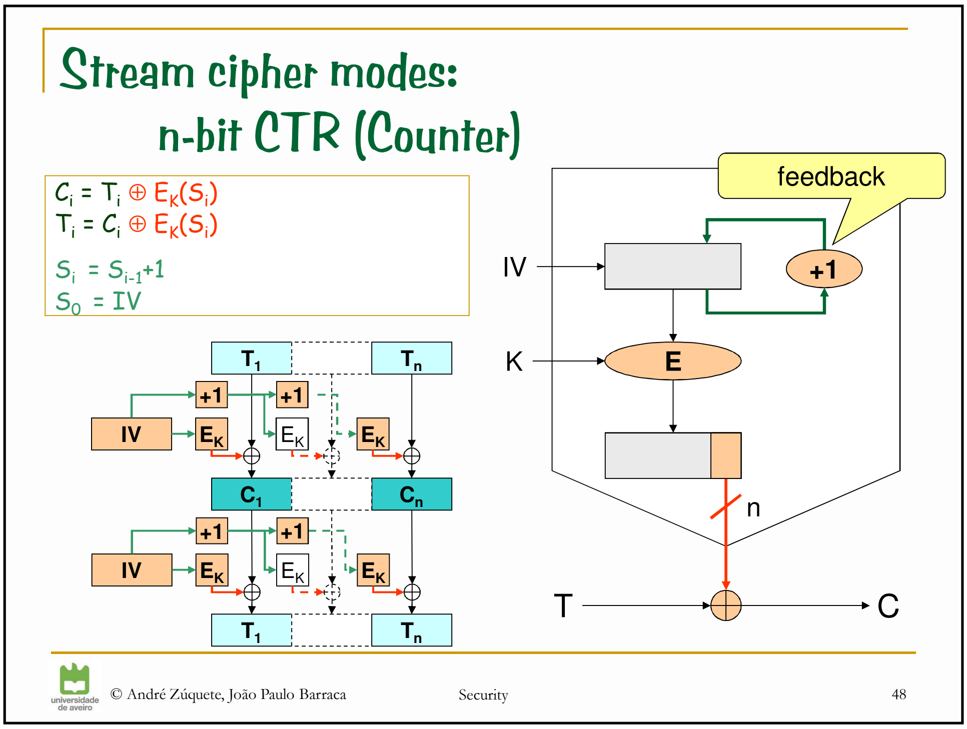
$S_i = f(S_{i-1}, E_K(S_{i-1}))$
$S_0 = IV$



Stream cipher modes: n-bit CFB (Ciphertext Feedback)

$C_i = T_i \oplus E_K(S_i)$
$T_i = C_i \oplus E_K(S_i)$

$S_i = f(S_{i-1}, C_i)$
$S_0 = IV$



Stream cipher modes: n-bit CTR (Counter)

$C_i = T_i \oplus E_K(S_i)$
$T_i = C_i \oplus E_K(S_i)$

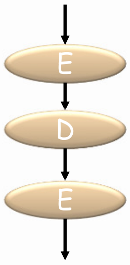
$S_i = S_{i-1}+1$
$S_0 = IV$

Cipher modes:
Pros and cons

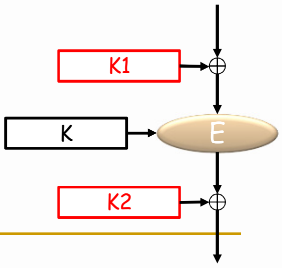
	Block		Stream		
	ECB	CBC	OFB	CFB	CTR
Input pattern hiding		✓	✓	✓	✓
Confusion on the cipher input		✓		✓	Secret counter
Same key for different messages	✓	✓	other IV	other IV	other IV
Tampering difficulty	✓	✓ (...)		✓	
Pre-processing			✓	...	✓
Parallel processing	✓	Decryption Only	w/ pre-processing	Decryption only	✓
Uniform random access					
Error propagation	Same block	Same block Next block		Some bits afterwards	
Capacity to recover from losses	Block Losses	Block Losses		✓	

Cipher modes:
Security reinforcement

- Multiple encryption
  - Double encryption
    - Breakable with a meet-in-the-meddle attack in 2^{n+1} attempts
      - With 2 or more known plaintext blocks
      - Using 2^n blocks stored in memory ...
    - Not secure enough (theoretically)
  - Triple encryption (EDE)
    - $C_i = E_{K1}(D_{K2}(E_{K3}(T_i)))$ $P_i = D_{K3}(E_{K2}(D_{K1}(C_i)))$
    - Usually $K_1 = K_3$
    - If $K_1 = K_2 = K_3$,then we get simple encryption



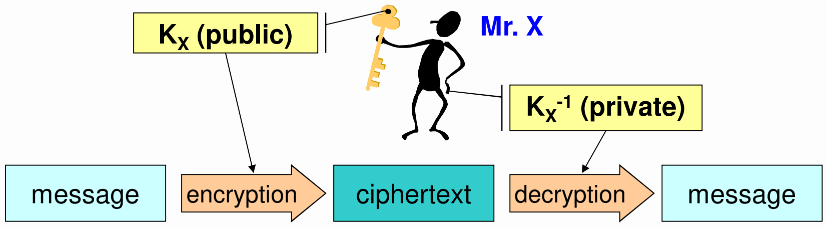
- Whitening (DESX)
  - Simple and efficient technique to add confusion
  - $C_i = E_K(K_1 \oplus T_i) \oplus K_2$
  - $T_i = K_1 \oplus D_K(K_2 \oplus C_i)$



Asymmetric (block) ciphers

- Use key pairs
  - One private key (personal, not transmittable)
  - One public key
- Allow
  - Confidentiality without any previous exchange of secrets
  - Authentication
    - Of contents (data integrity)
    - Of origin (source authentication, or digital signature)
- Disadvantages
  - Performance (usually very inefficient and memory consuming)
- Advantages
  - N peers requiring pairwise, secret interaction ⇒ N key pairs
- Problems
  - Distribution of public keys
  - Lifetime of key pairs

Confidentiality w/ asymmetric ciphers



K_X (public)

Mr. X

K_X^{-1} (private)

message

encryption

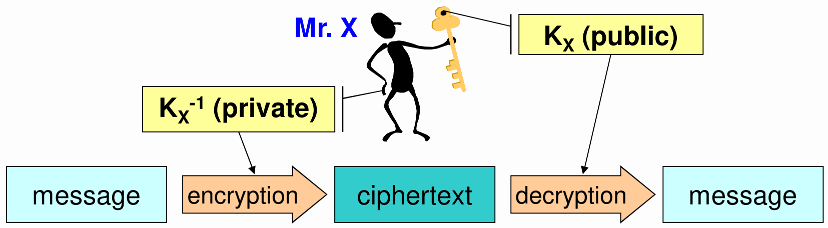
ciphertext

decryption

message

- Only the key pair of the recipient is involved
  - C = E(K, P)   P = D(K^{-1}, C)
  - To send something with confidentiality to X is only required to know X's public key (K_X)
- There is no source authentication
  - X has no means to know who produced the ciphertext
  - If K_X is really public, then everybody can do it

Source authentication w/ asymmetric ciphers



Mr. X

K_X^{-1} (private)

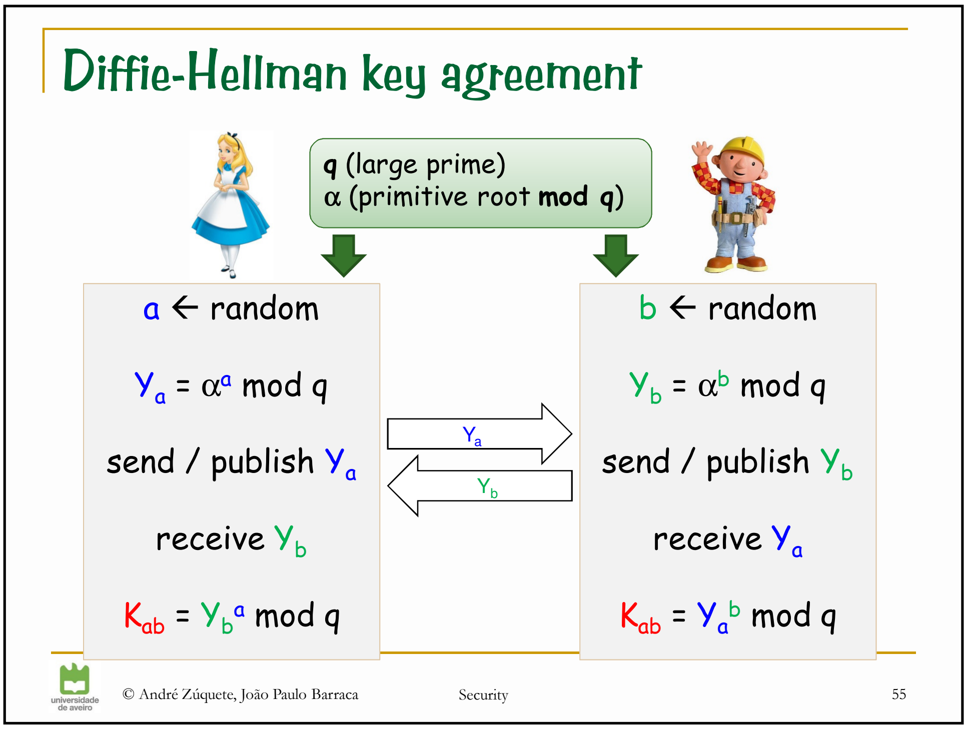
K_X (public)

message → encryption → ciphertext → decryption → message

- Only the key pair of the originator is involved
  - $C = E(K^{-1}, P)$ $P = D(K, C)$;
  - Only X knows K_X^{-1} that produced C
- There is no confidentiality
  - Anyone knowing the public key of the originator (K_X) can decrypt C
  - If K_X is really public, then everybody can do it

Asymmetric (block) ciphers

- Approaches: complex mathematic problems
  - Discrete logarithms of large numbers
  - Integer factorization of large numbers
  - Knapsack problems
- Most common algorithms
  - RSA
  - ElGamal
  - Elliptic curves (ECC)
- Other techniques with asymmetric key pairs
  - Diffie-Hellman (key agreement)



Diffie-Hellman key agreement

q (large prime)
α (primitive root **mod q**)

$a \leftarrow$ random

$Y_a = \alpha^a \bmod q$

send / publish Y_a

receive Y_b

$K_{ab} = Y_b^{\,a} \bmod q$

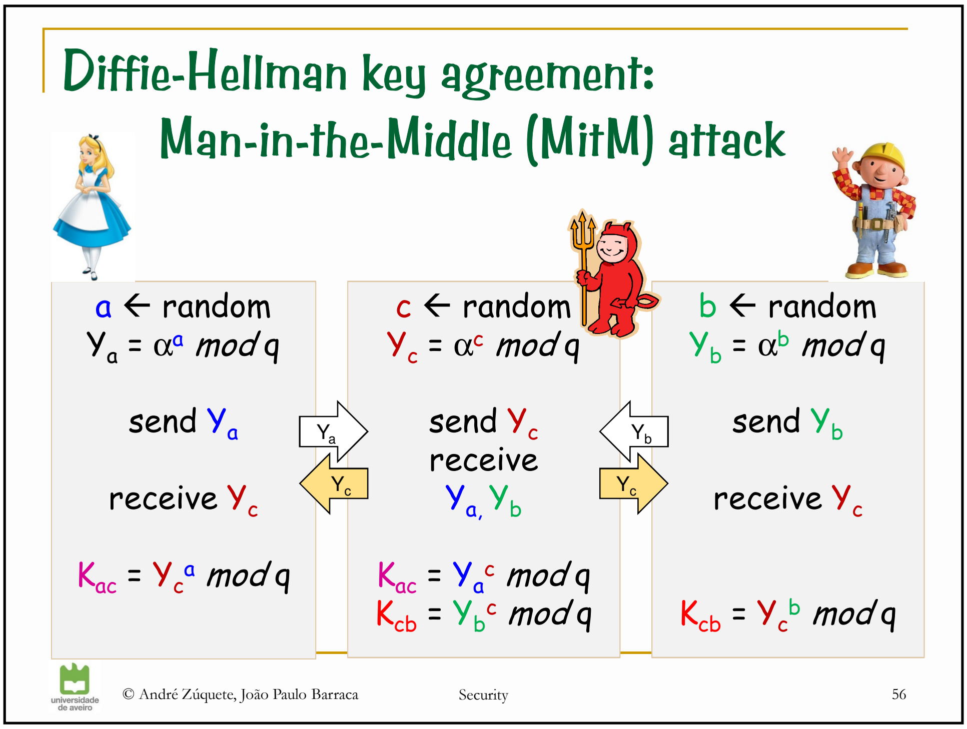
$b \leftarrow$ random

$Y_b = \alpha^b \bmod q$

send / publish Y_b

receive Y_a

$K_{ab} = Y_a^{\,b} \bmod q$



Diffie-Hellman key agreement: Man-in-the-Middle (MitM) attack

$a \leftarrow$ random
$Y_a = \alpha^a$ *mod* q

send Y_a

receive Y_c

$K_{ac} = Y_c^{\,a}$ *mod* q

$c \leftarrow$ random
$Y_c = \alpha^c$ *mod* q

send Y_c
receive
Y_a, Y_b

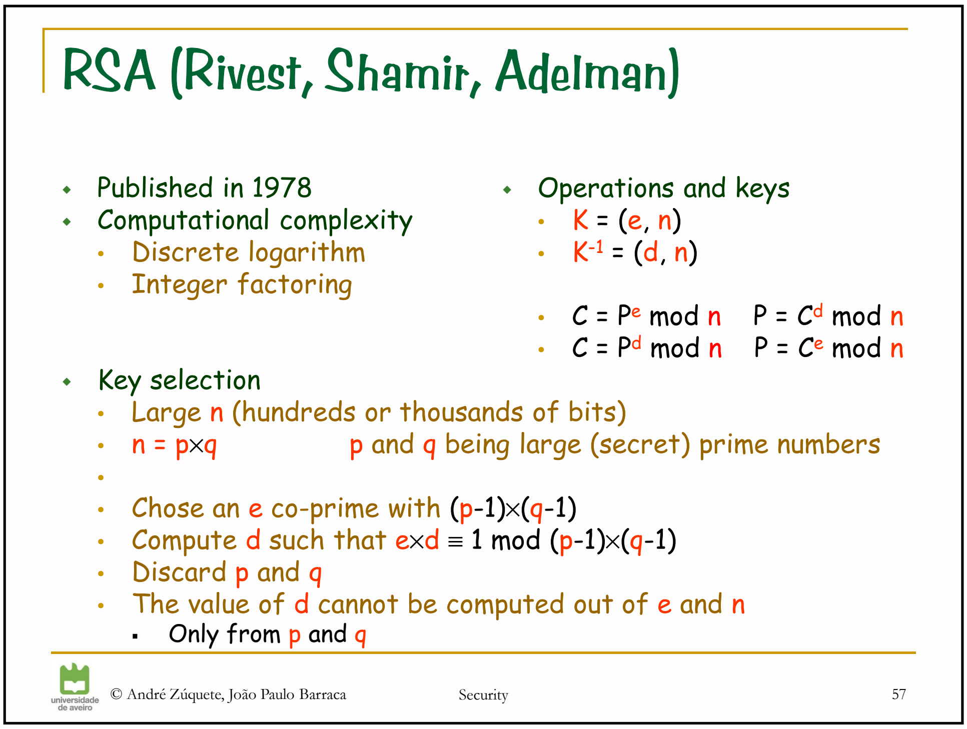
$K_{ac} = Y_a^{\,c}$ *mod* q
$K_{cb} = Y_b^{\,c}$ *mod* q

$b \leftarrow$ random
$Y_b = \alpha^b$ *mod* q

send Y_b

receive Y_c

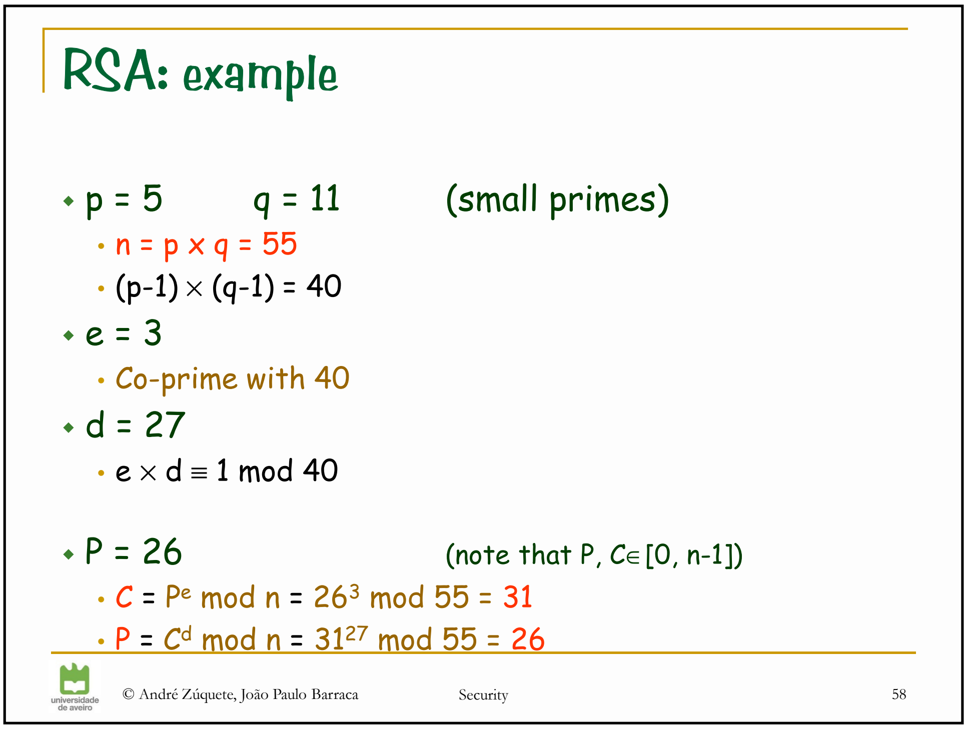
$K_{cb} = Y_c^{\,b}$ *mod* q



RSA (Rivest, Shamir, Adelman)

- Published in 1978
- Computational complexity
  - Discrete logarithm
  - Integer factoring
- Operations and keys
  - $K = (e, n)$
  - $K^{-1} = (d, n)$

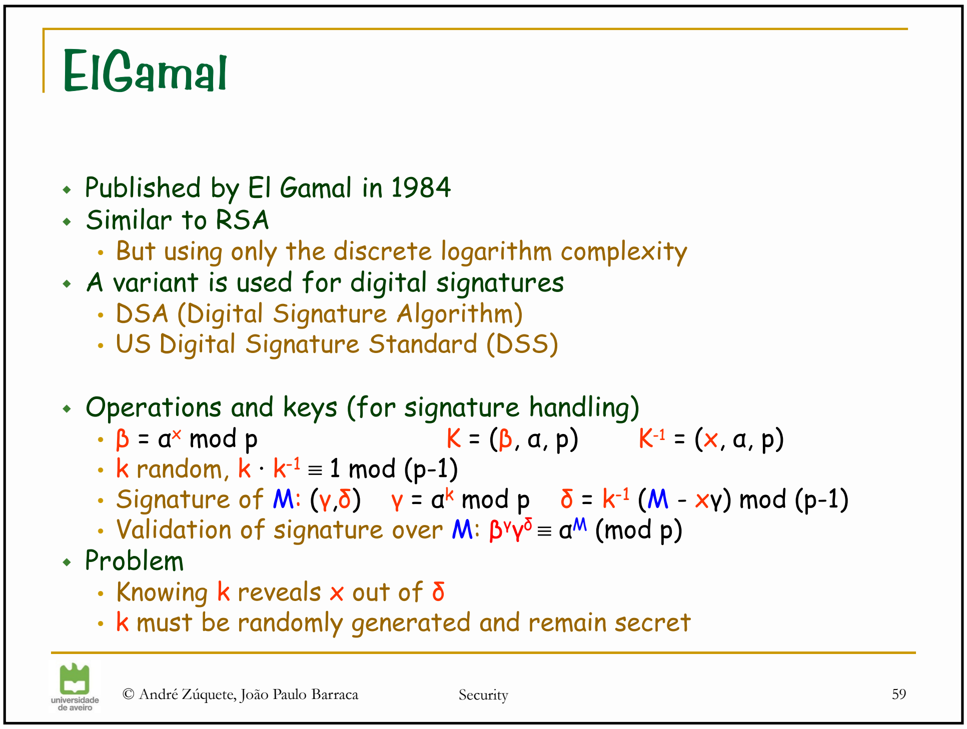
  - $C = P^e \bmod n \quad P = C^d \bmod n$
  - $C = P^d \bmod n \quad P = C^e \bmod n$
- Key selection
  - Large n (hundreds or thousands of bits)
  - $n = p \times q$ p and q being large (secret) prime numbers
  - Chose an e co-prime with $(p-1)\times(q-1)$
  - Compute d such that $e \times d \equiv 1 \bmod (p-1)\times(q-1)$
  - Discard p and q
  - The value of d cannot be computed out of e and n
    - Only from p and q



RSA: example

- p = 5 q = 11 (small primes)
  - $n = p \times q = 55$
  - $(p-1) \times (q-1) = 40$
- e = 3
  - Co-prime with 40
- d = 27
  - $e \times d \equiv 1 \bmod 40$

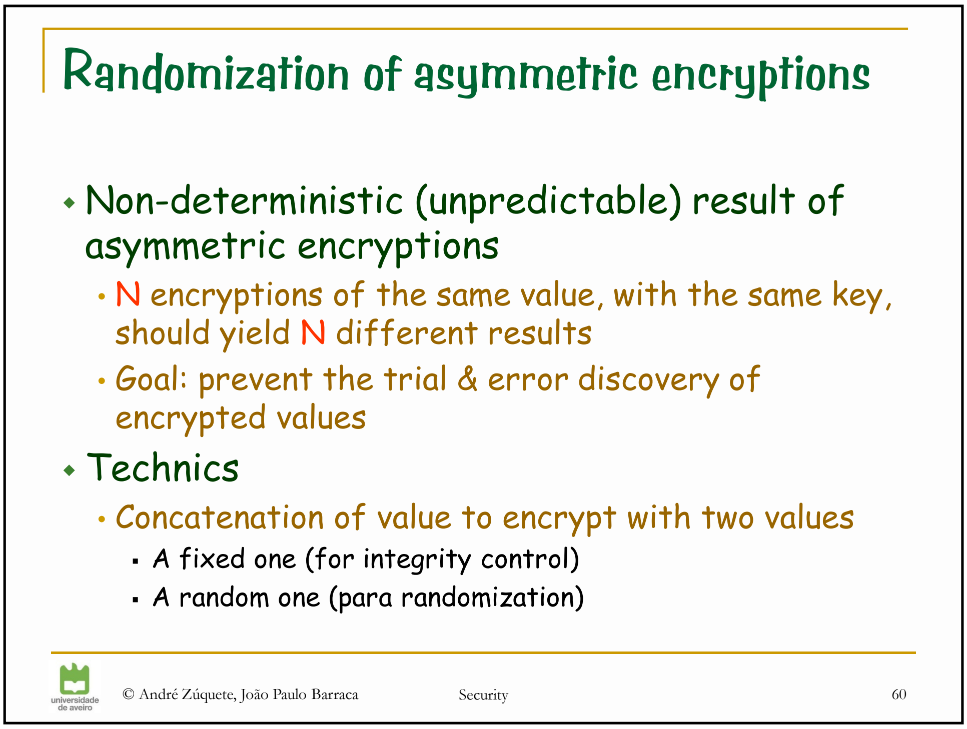
- P = 26 (note that $P, C \in [0, n-1]$)
  - $C = P^e \bmod n = 26^3 \bmod 55 = 31$
  - $P = C^d \bmod n = 31^{27} \bmod 55 = 26$



ElGamal

- Published by El Gamal in 1984
- Similar to RSA
  - But using only the discrete logarithm complexity
- A variant is used for digital signatures
  - DSA (Digital Signature Algorithm)
  - US Digital Signature Standard (DSS)

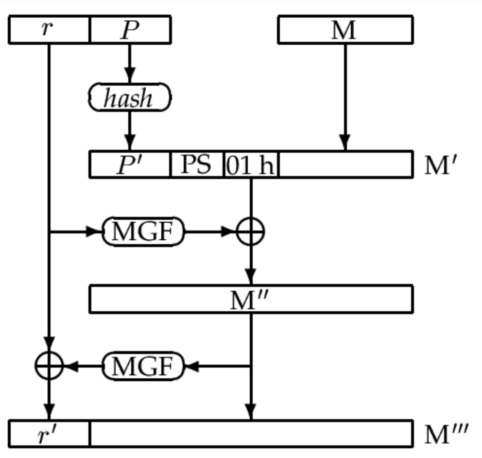
- Operations and keys (for signature handling)
  - $\beta = \alpha^{x} \bmod p$ $\quad K = (\beta, \alpha, p)$ $\quad K^{-1} = (x, \alpha, p)$
  - k random, $k \cdot k^{-1} \equiv 1 \bmod (p-1)$
  - Signature of M: (γ,δ) $\quad \gamma = \alpha^{k} \bmod p$ $\quad \delta = k^{-1}(M - x\gamma) \bmod (p-1)$
  - Validation of signature over M: $\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{M} \pmod p$
- Problem
  - Knowing k reveals x out of δ
  - k must be randomly generated and remain secret



Randomization of asymmetric encryptions

- Non-deterministic (unpredictable) result of asymmetric encryptions
  - N encryptions of the same value, with the same key, should yield N different results
  - Goal: prevent the trial & error discovery of encrypted values
- Technics
  - Concatenation of value to encrypt with two values
    - A fixed one (for integrity control)
    - A random one (para randomization)

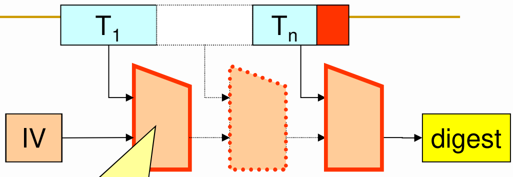
Randomization of asymmetric encryptions: OAEP (Optimal Asymmetric Encryption Padding)



Digest functions

- Give a fixed-length value from a variable-length text
  - Sort of text "fingerprint"
- Produce very different values for similar texts
  - Cryptographic one-way hash functions
- Relevant properties:
  - Preimage resistance
    - Given a digest, it is infeasible to find an original text producing it
  - 2nd-preimage resistance
    - Given a text, it is infeasible to find another one with the same digest
  - Collision resistance
    - It is infeasible to find any two texts with the same digest
    - Birthday paradox

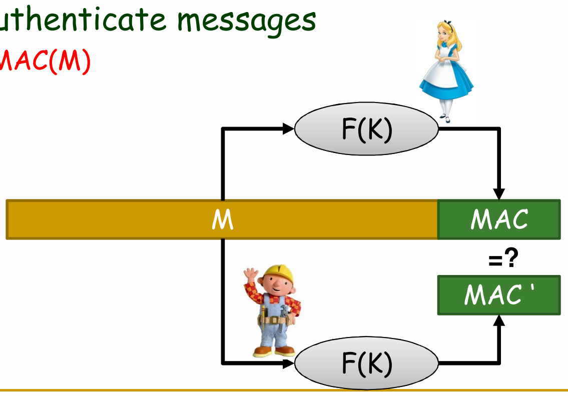
Digest functions

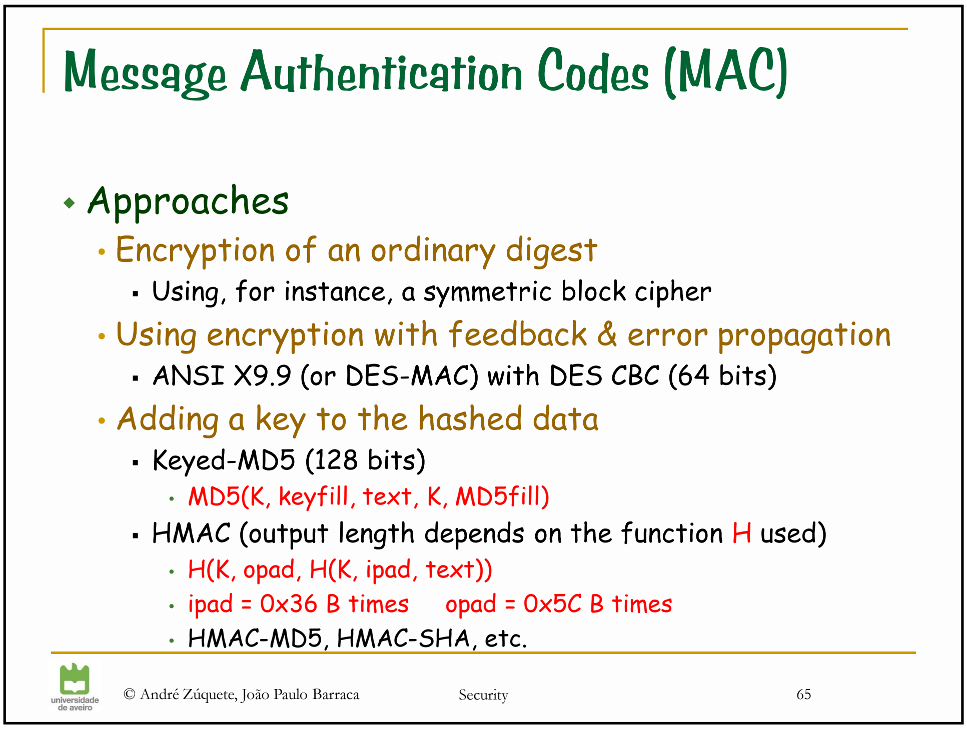


- Approaches
  - Collision-resistente, one-way compression functions
  - Merkle-Damgård construction
    - Iterative compression
    - Length padding
- Most common algorithms
  - MD5 (128 bits)
    - No longer secure! It's easy to find collisions!
  - SHA-1 (Secure Hash Algorithm, 160 bits)
    - Also no longer secure … (collisions found in 2017)
  - Other
    - SHA-2, aka SHA-256/SHA-512, SHA-3, etc.

Message Authentication Codes (MAC)

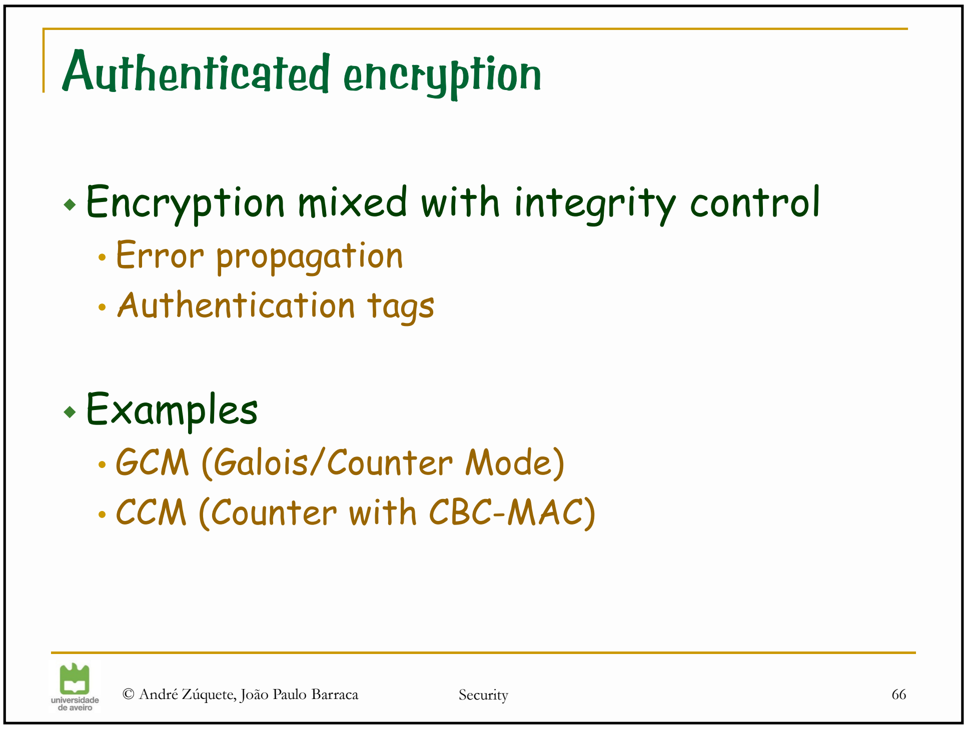
- Hash, or digest, computed with a key
  - Only key holders can generate/validate the MAC
- Used to authenticate messages
  - M' = M | MAC(M)





Message Authentication Codes (MAC)

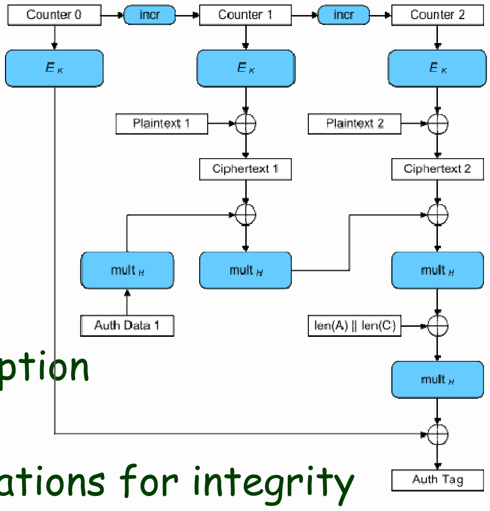
- Approaches
  - Encryption of an ordinary digest
    - Using, for instance, a symmetric block cipher
  - Using encryption with feedback & error propagation
    - ANSI X9.9 (or DES-MAC) with DES CBC (64 bits)
  - Adding a key to the hashed data
    - Keyed-MD5 (128 bits)
      - MD5(K, keyfill, text, K, MD5fill)
    - HMAC (output length depends on the function H used)
      - H(K, opad, H(K, ipad, text))
      - ipad = 0x36 B times   opad = 0x5C B times
      - HMAC-MD5, HMAC-SHA, etc.



Authenticated encryption

- Encryption mixed with integrity control
  - Error propagation
  - Authentication tags
- Examples
  - GCM (Galois/Counter Mode)
  - CCM (Counter with CBC-MAC)

GCM



- CTR mode for encryption
- Successive multiplications for integrity control
  - Multiplications in GF(2^n)

Encryption + authentication

- Encrypt-then-MAC
  - MAC is computed from cryptogram
- Encrypt-and-MAC
  - MAC is computed from plaintext
  - MAC is not encrypted
- MAC-then-Encrypt
  - MAC is computed from plaintext
  - MAC is encrypted

Digital signatures

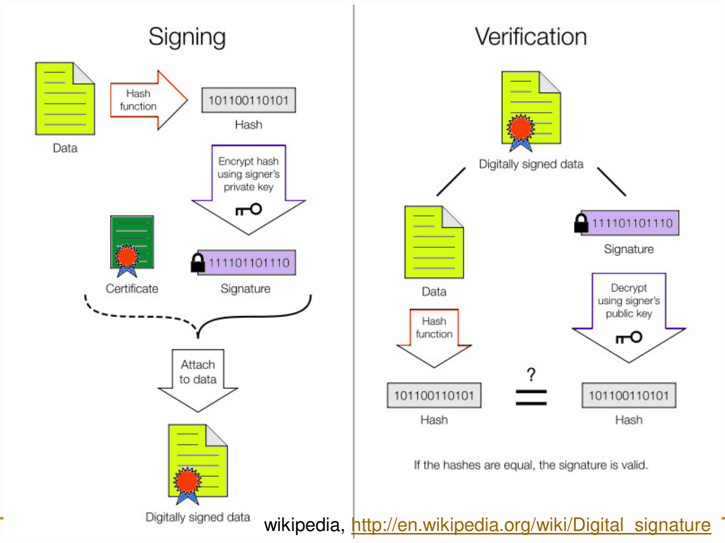
- Goal
  - Authenticate the contents of a document
    - Ensure its integrity
  - Authenticate its author
    - Ensure the identity of the creator/originator
  - Prevent origin repudiation
    - Genuine authors cannot deny authorship
- Approaches
  - Asymmetric encryption
  - Digest functions (only for performance)
- Algorithms

  Signing: $A_x(doc) = info + E(K_x^{-1}, digest(doc+info))$

  Verification: $info \rightarrow K_x$

  $D(K_x, A_x(doc)) \equiv digest(doc + info)$

Signing / verification diagrams



wikipedia, http://en.wikipedia.org/wiki/Digital_signature

Digital signature on a mail: Multipart content, signature w/ certificate

```
From - Fri Oct 02 15:37:14 2009
[…]
Date: Fri, 02 Oct 2009 15:35:55 +0100
From: =?ISO-8859-1?Q?Andr=E9_Z=FAquete?= <andre.zuquete@ua.pt>
Reply-To: andre.zuquete@ua.pt
Organization: IEETA / UA
MIME-Version: 1.0
To: =?ISO-8859-1?Q?Andr=E9_Z=FAquete?= <andre.zuquete@ua.pt>
Subject: Teste
Content-Type: multipart/signed; protocol="application/x-pkcs7-signature"; micalg=sha1; boundary="------------ms050405070101010502050101"

This is a cryptographically signed message in MIME format.

--------------ms050405070101010502050101
Content-Type: multipart/mixed;
 boundary="------------060802050708070409030504"

This is a multi-part message in MIME format.
--------------060802050708070409030504
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: quoted-printable

Corpo do mail

--------------060802050708070409030504–
--------------ms050405070101010502050101
Content-Type: application/x-pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"
Content-Description: S/MIME Cryptographic Signature

MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMIAGCSqGSIb3DQEHAQAAoIIamTCC
BUkwggSyoAMCAQICBAcnIaEwDQYJKoZIhvcNAQEFBQAwdTELMAkGA1UEBhMCVVMxGDAWBgNV
[…]
KoZIhvcNAQEBBQAEgYCofks852BV77NVuww53vSxO1XtI2JhC1CDlu+tcTPoMD1wq5dc5v40
Tgsaw0N8dqgVLk8aC/CdGMbRBu+J1LKrcVZa+khnjjtB66HhDRLrjmEGDNttrEjbqvpd2QO2
vxB3iPTlU+vCGXo47e6GyRydqTpbq0r49Zqmx+IJ6Z7iigAAAAAAAA==
--------------ms050405070101010502050101--
```

Blind signatures

- Signatures made by a "blinded" signer
  - Signer cannot observe the signed contents
  - Similar to a handwritten signature on an envelope containing a document and a carbon-copy sheet
- They are useful for ensuring anonymity of the signed information holder, while the signed information provides some extra functionality
  - Signer X knows who requires a signature (Y)
  - X signs T_1, but Y afterwards transforms it into a signature over T_2
    - Not any T_2, a specific one linked to T_1
  - Requester Y can present T_2 signed by X
    - But it cannot change T_2
    - X cannot link T_2 to the T_1 that it observed when signing

Chaum Blind Signatures

- Implementation using RSA
  - Blinding
    - Random blinding factor K
    - $k \times k^{-1} \equiv 1 \pmod{N}$
    - $m' = k^e \times m \bmod N$
  - Ordinary signature (encryption w/ private key)
    - $A_x(m') = (m')^d \bmod N$
  - Unblinding
    - $A_x(m) = k^{-1} \times A_x(m') \bmod$