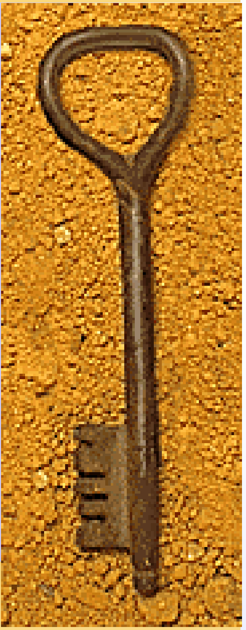
Graduate Course on **Computer Security**

Lecture 2: Shared-Key Cryptography

Iliano Cervesato **iliano@itd.nrl.navy.mil**

ITT Industries, Inc @ NRL - Washington DC

http://www.cs.stanford.edu/~iliano/

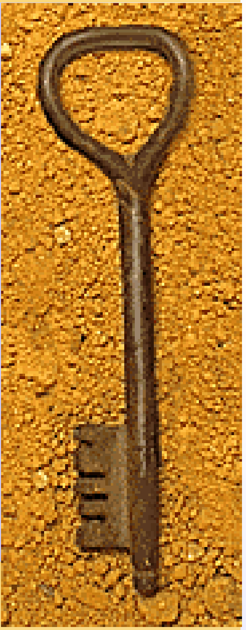


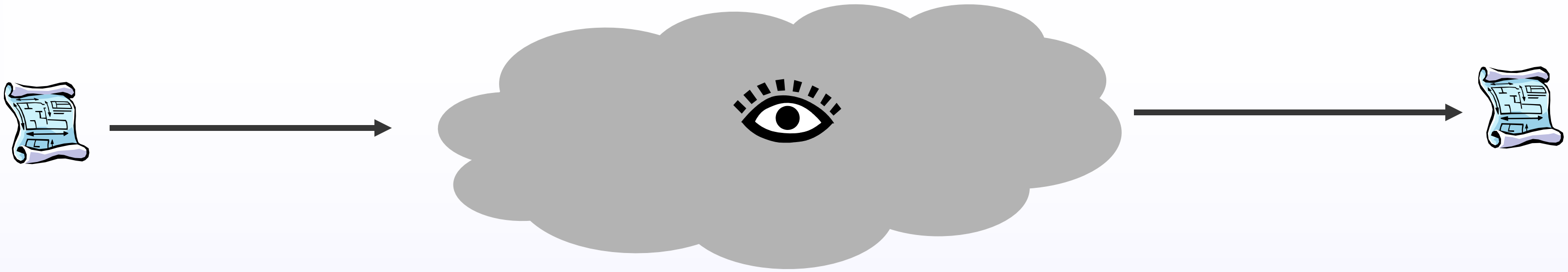
Outline

- Goals of cryptography
- History
- Symmetric ciphers
  - Attacks
  - Block ciphers
  - Stream ciphers
  - Data Encryption Standard (DES)
- What is a secure cipher?

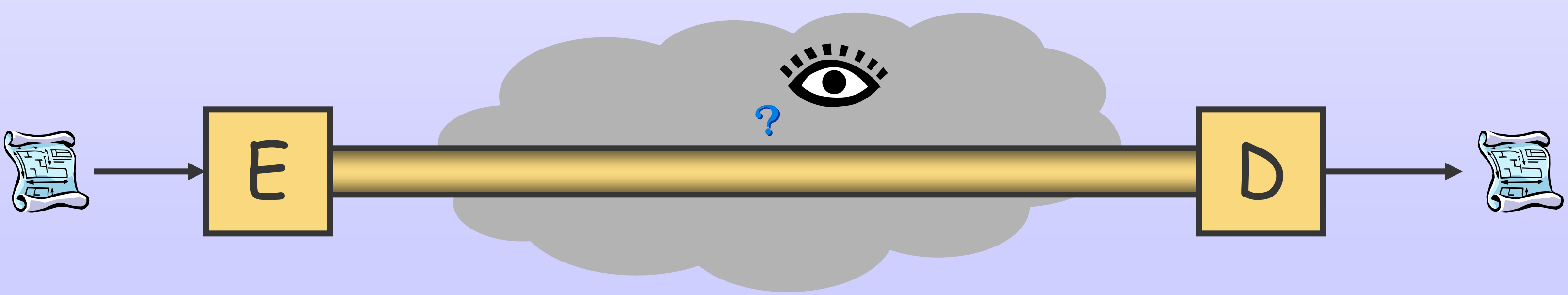
Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

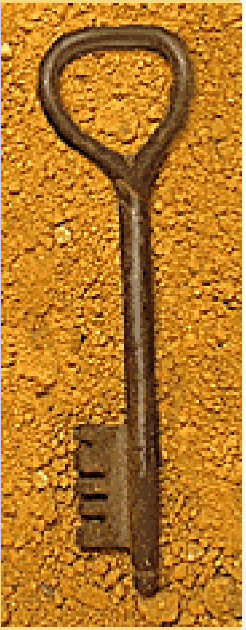
Confidentiality





Implement a virtual trusted channel over an insecure medium





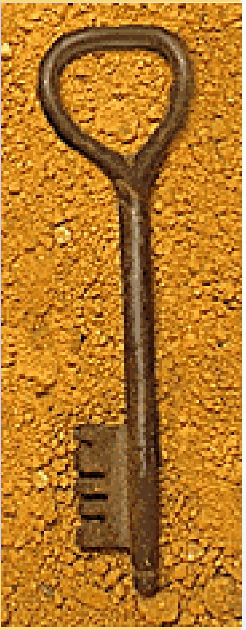
Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

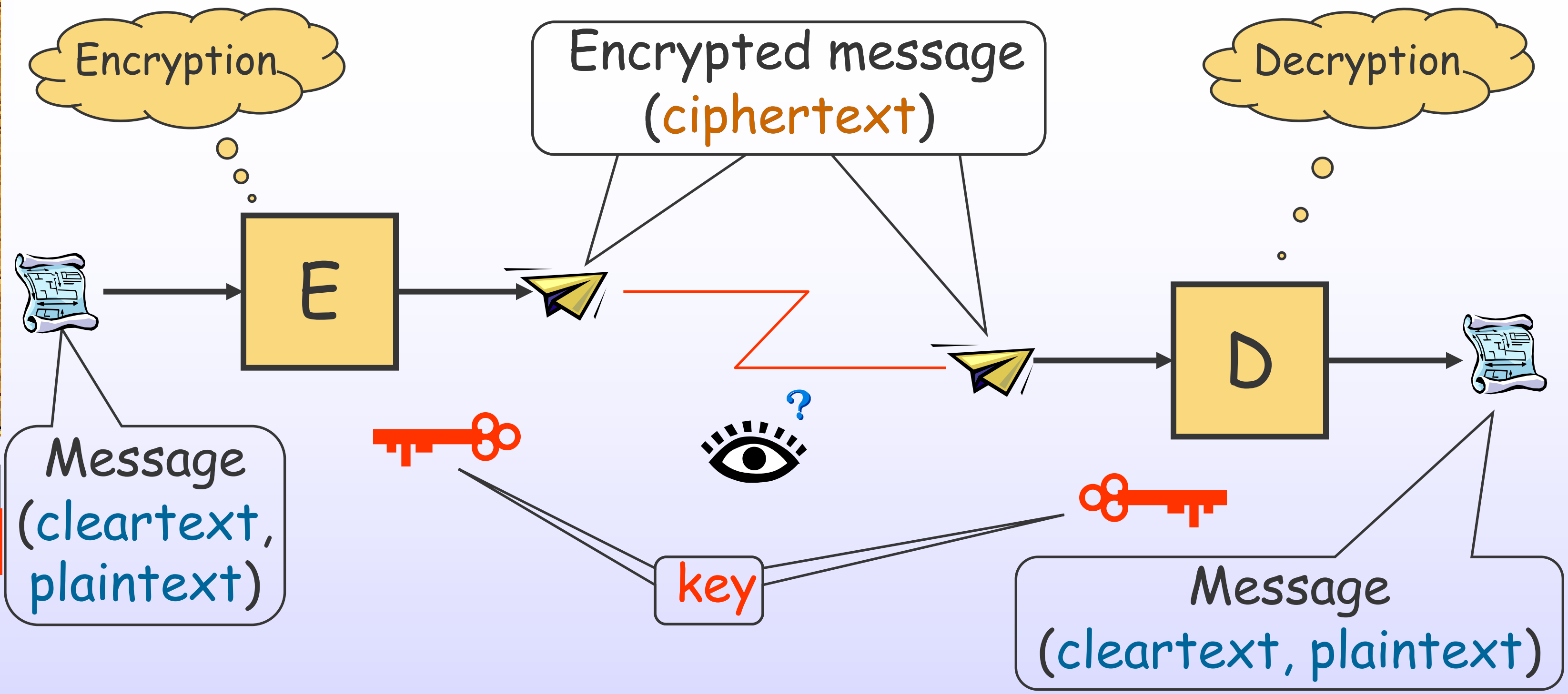
Insecure Channels

External observer can

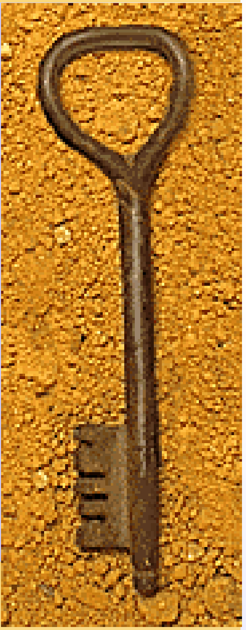
- Read traffic
- Inject new traffic
- Erase traffic … sometimes
- Modify traffic … sometimes

Classical Goals of Cryptography





E, D realize a virtual trusted channel, given key



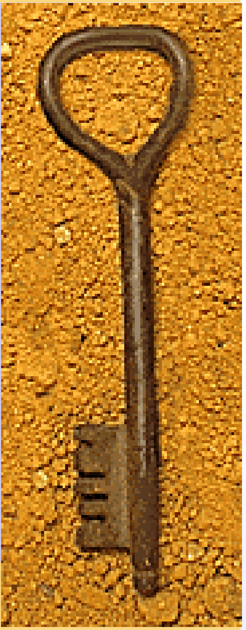
Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

Modern Cryptography

Not just about confidentiality!

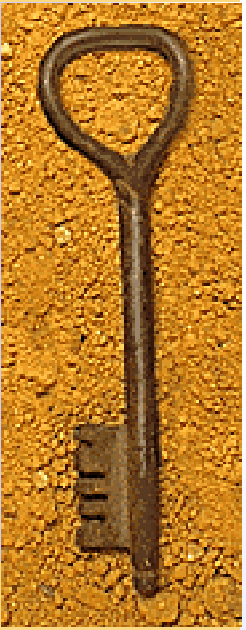
- Integrity
  - Digital signatures
  - Hash functions
- Fair exchange
  - Contract signing
- Anonymity
  - Electronic cash
  - Electronic voting
- ...

A Brief History of Cryptography



- ~2000 years ago: *Substitution ciphers*
- A few centuries later: *Permutation ciphers*
- Renaissance: *Polyalphabetic ciphers*
- 1844: *Mechanization*
- 1976: *Public-key cryptography*

Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.



Substitution Ciphers

Caesar's cipher:

A	→	C
B	→	E
D	→	F
	…	
X	→	A
Y	→	B
Z	→	C

Replace each letter with another

- Key: substitution table
- How to break it?
  - Brute force? 26! possibilities (= 4×10^{26})
  - Count the frequencies of letters, pairs, …
    - Arabs had tabulated the Koran by 1412
  - Ciphertext is enough: ciphertext-only attack
- Example:

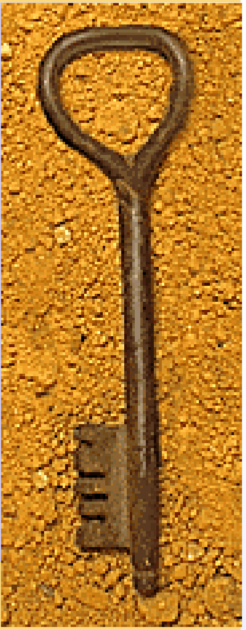
  QVAQBCWZQRLWDVEFW

  IAMINDECIPHERABLE

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

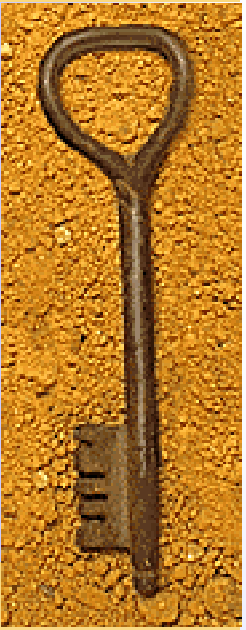
Permutation Ciphers

$$k = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 1 & 2 \end{pmatrix}$$



Switch letters around by a permutation

- Example: HELLOWORLD → LOLHERDLWO
- Key: permutation
- Breakable with ciphertext-only attack



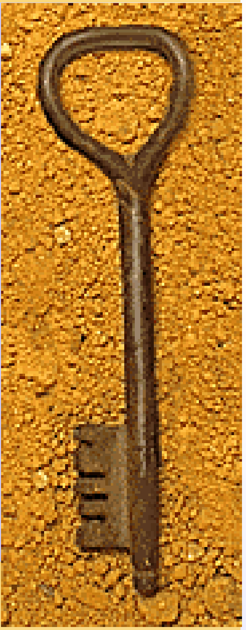
Renaissance Ciphers

Use message and key letters for cipher

- Key: a word (CRYPTO)
- Example:

```
    WHATANICEDAYTODAY
  + CRYPTOCRYPTOCRYPT   (mod 26)
  ---------------------
    ZZZJUCLUDTUNWGCQS
```

- Polyalphabetic cipher:
  - Encryption of letter is context-dependent
- Seed of modern cryptography



Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

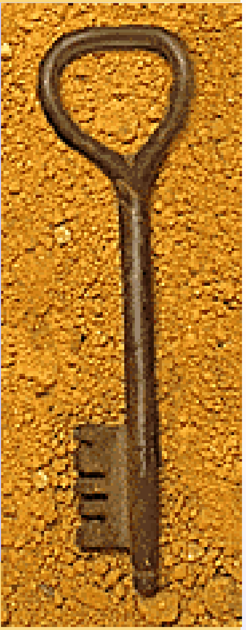
Mechanization

The Enigma

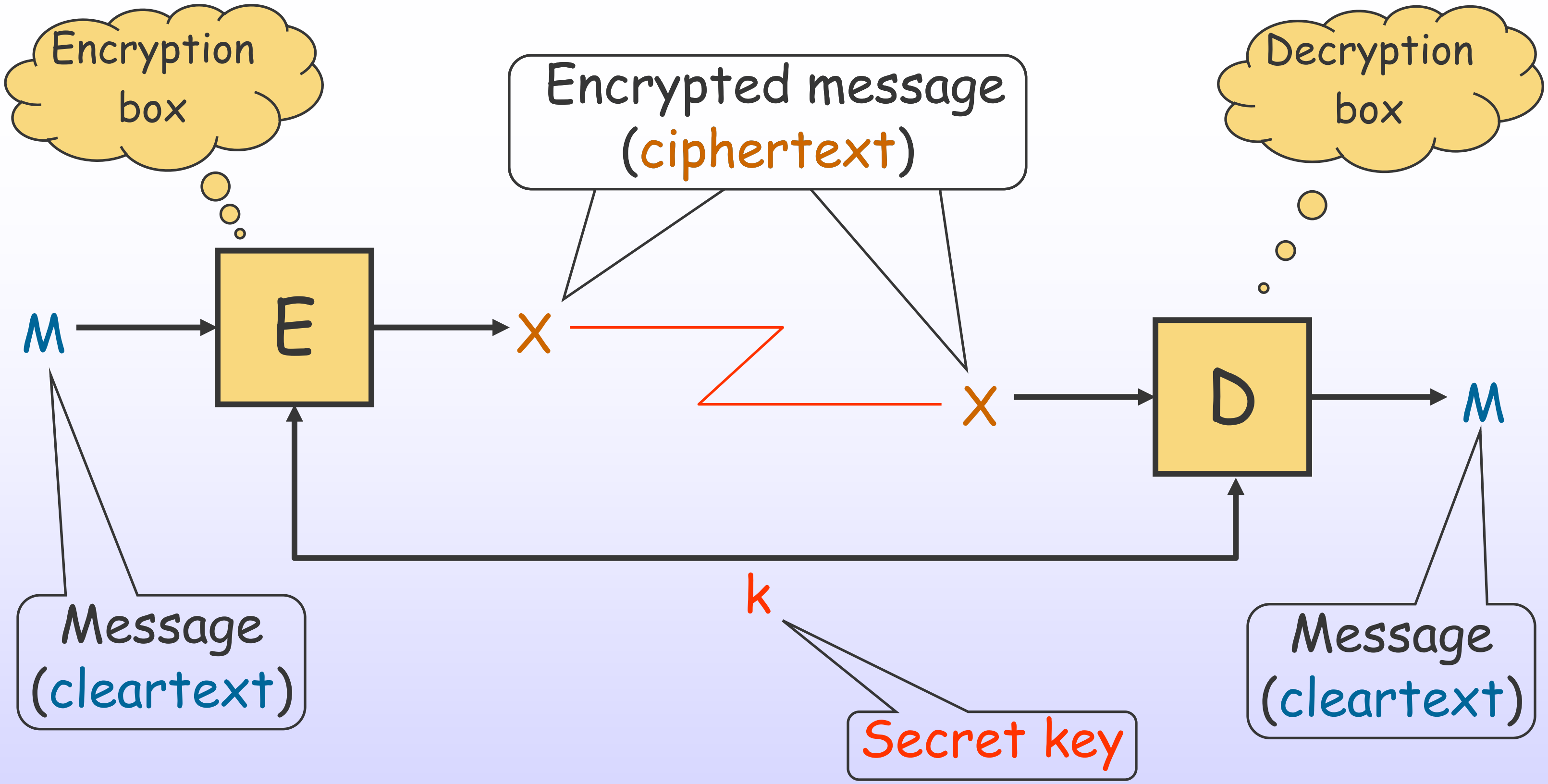
- 1844: invention of telegraph
  - Beginning of civilian crypto
- Rotor machines
  - Key: initial position of rotors
  - Culminate in WW II
- 1975: DES
  - 1996-2000 AES
- 1976: Public key cryptography

We will examine in some detail

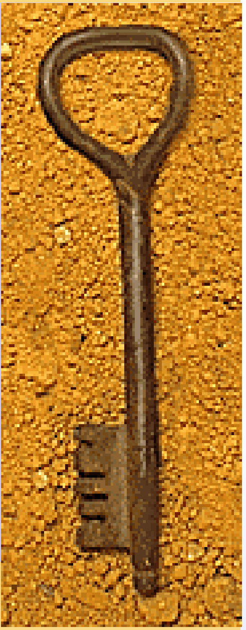
Symmetric Ciphers



- Goals
- History
- Shared-Key
- Attacks
- Block C.
- Stream C.
- DES
- Secure C.



$$D_k(E_k(m)) = m$$

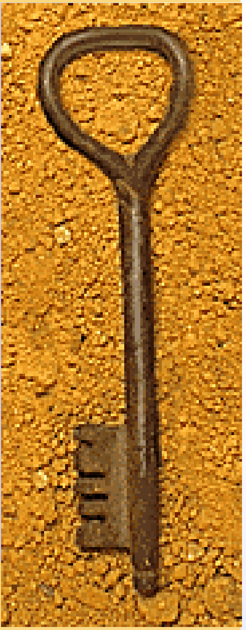


Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

Properties of a Good Cipher

$$E, D : \{0,1\}^n \times \{0,1\}^l \rightarrow \{0,1\}^n$$

- $D_k(E_k(m)) = m$
  - For every k, E_k is an injection with inverse D_k
- $E_k(m)$ is <u>easy</u> to compute, given m and k
- $D_k(x)$ is <u>easy</u> to compute, given x and k
  - <u>Polynomial</u> in $\max\{n,l\}$ - often linear
- If $x = E_k(m)$, it is **hard** to find m without k
  - **Exponential** in $\min\{n,l\}$



Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

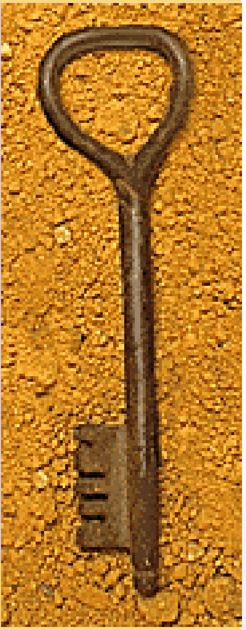
Open Design

Kerchoff's Principle (1883)

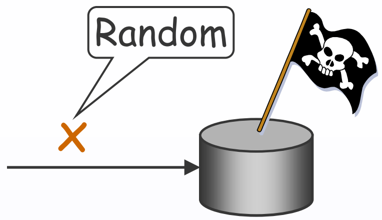
> The security of a cryptosystem must not depend on keeping the algorithm secret

No security by obscurity

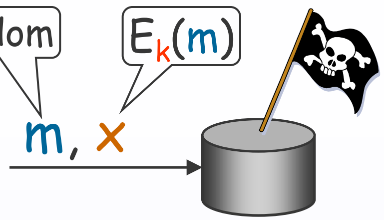
- Better
  - Lots of smart but innocuous people dissect it
  - Than a single smart malicious



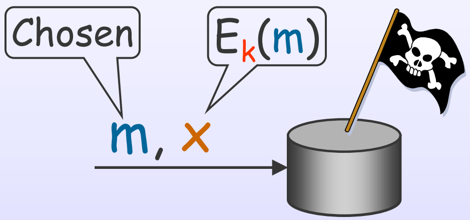
Attack Models



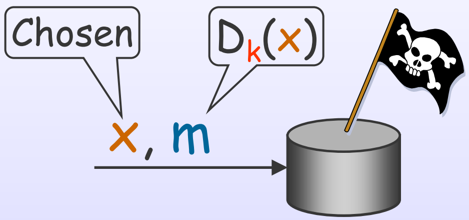
Ciphertext Only



Known Plaintext

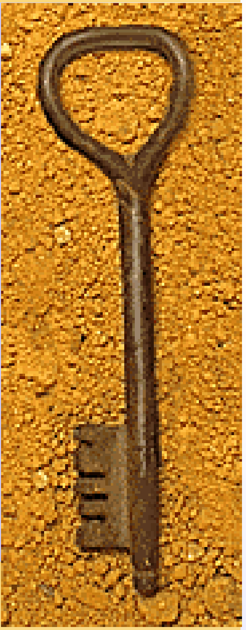


Chosen Plaintext



Known Plaintext

Good ciphers resist <u>all</u> attack models

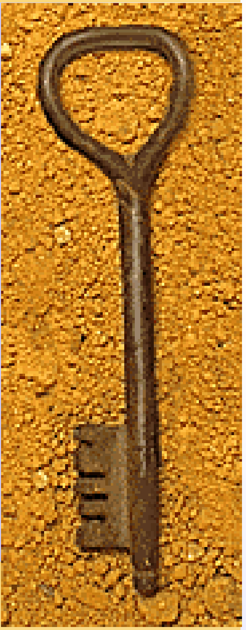


Successful Attacks

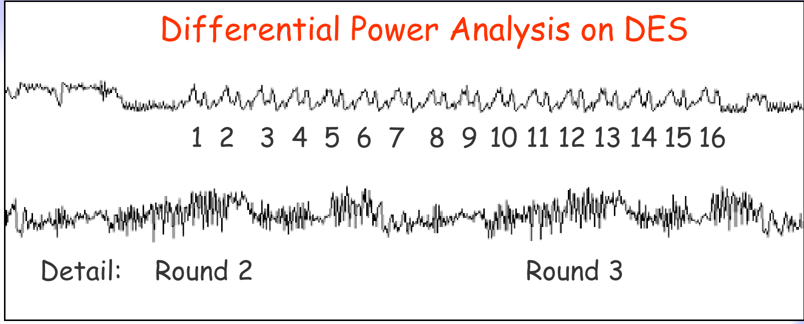
Decrypt future messages coded with k

- Recover k
  - Hard

- Often not needed!
  - Exploit properties of the cipher
  - See Lecture 5 (WEP)



Sneaky Attacks



From http://www.cryptography.com/dpa/technical

- Obtain the key somehow
  - Network sniffers, worms, backup tapes, …
  - Blackmail, bribery, torture, …

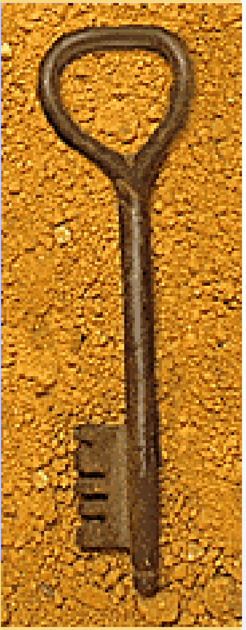
  Be careful!

- Side-channel cryptanalysis
  - Power consumption ⇒ off-peak computation
  - Encryption time ⇒ random noise
  - Radiation ⇒ physical shielding

  Better implementation and design

Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

Encrypting Long messages

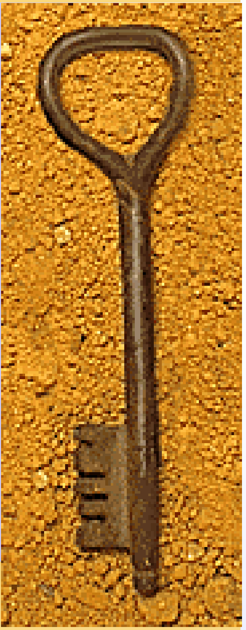


Most algorithms operate on fixed sizes

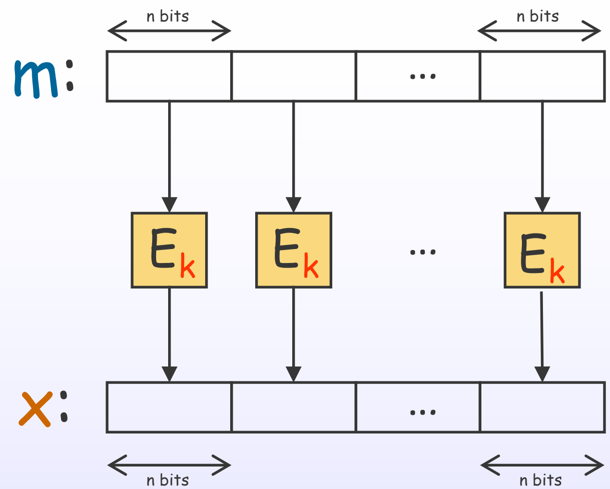
- E.g. 64 bits for DES

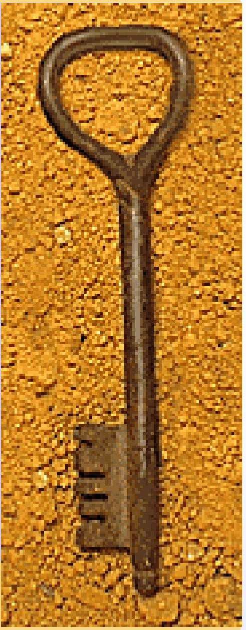
- Block ciphers
  - Slice m into $m_1, \ldots, m_n$
    - Add padding to last block
  - Use E_k to produce $x_1, \ldots, x_n$
  - Use D_k to recover $m_1, \ldots, m_n$
- Stream ciphers
  - Rely on pseudo-random sequence

Electronic Codebook Mode – ECB



- Any identical block encrypted identically
- Lots of ciphertext with the same k
- Dictionary attack
  - Attacker records blocks
  - Substitute them back when appropriate
    - Encryption guarantees secrecy, not integrity





Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

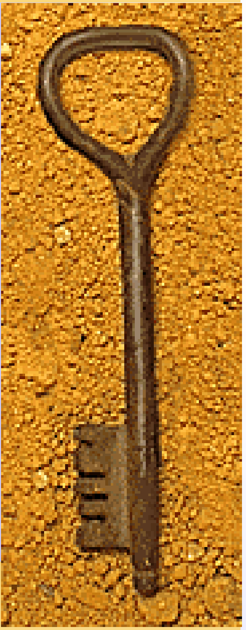
Exclusive OR

Fundamental operation of many ciphers

y	z	y ⊕ z
0	0	0
0	1	1
1	1	0
1	0	1

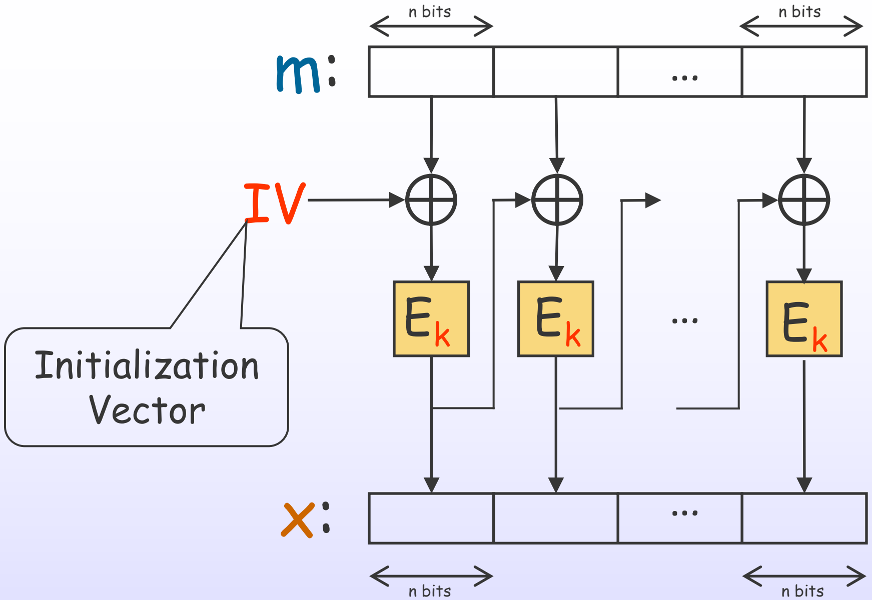
- Properties
  - $y \oplus y = 0$
  - $y \oplus 0 = y$
  - $y \oplus 1 = \overline{y}$
  - $y \oplus z \oplus z = y$

Cipher Block Chaining – CBC

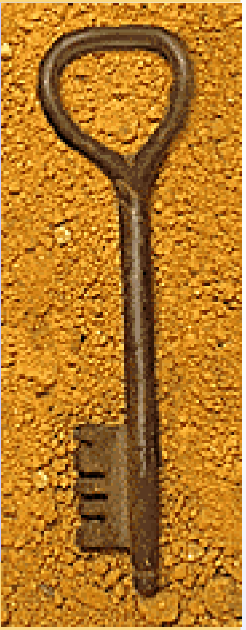


Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

- Encryption
  - $x_1 = E_k(m_1 \oplus IV)$
  - $x_i = E_k(m_i \oplus x_{i-1})$
- Decryption
  - $m_1 = D_k(x_1) \oplus IV$
  - $m_i = D_k(x_i) \oplus x_{i-1}$
- Widely used
  - E.g IPSec

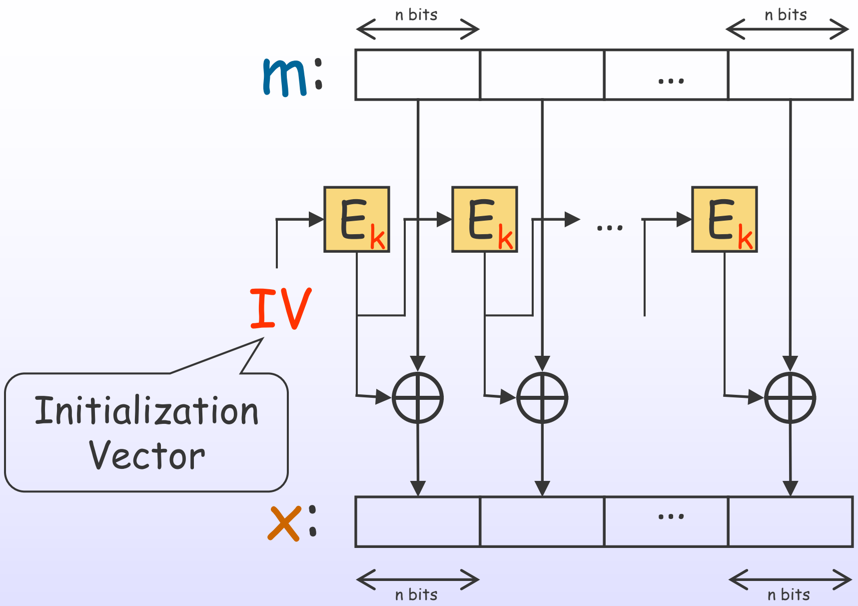


Output Feedback Mode – OFB

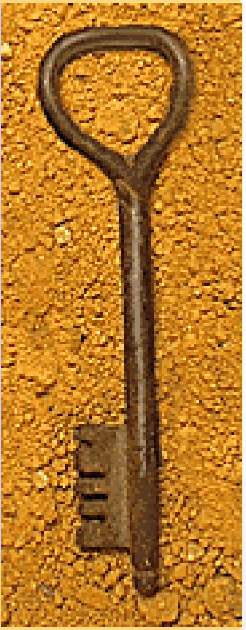


Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

- Encryption
  - $x_i = m_i \oplus E_k(IV)^i$
- Decryption
  - $m_i = x_i \oplus D_k(IV)^i$



NB: encryption is never applied to m



Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

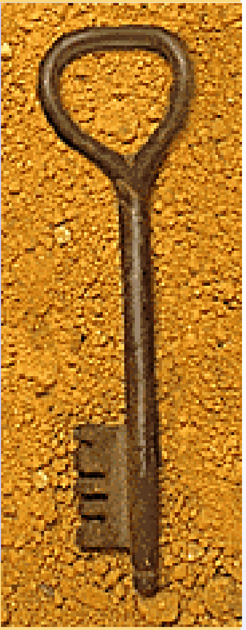
One-Time Pad

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

- $D_k(x) = x \oplus k$
- Requires $|m| = |k|$
- Very fast
- Perfect secrecy
  - Prob[*guessing* m] = Prob[*guessing* m|x]
- k should never be reused again!
  - $x_1 = m_1 \oplus k$
  - $x_2 = m_2 \oplus k$

  $\Rightarrow x_1 \oplus x_2 = m_1 \oplus m_2$
- k very large for long messages
  - How to distribute it?

Pseudo-Random Bit Generators



Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

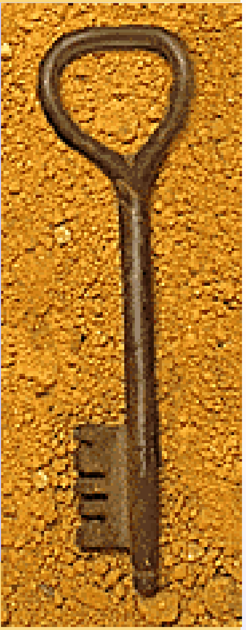
- Deterministic functions
  - RNG : $\{0,1\}^n \rightarrow \{0,1\}^\infty$
- Stretch fixed-size seed to an unbounded sequence that looks random
- Computable approximation of one-time pad
- Example: RC4

Example:

```
i := 0
i := 0
do forever
    i := i+1 mod 256
    j := j+s[I] mod 256
    swap s[i],s[j]
    t := s[i]+s[j] mod 256
    output s[t]
```

Seed: initial value of s

Size of state: $(2^{256})^{256}$



Stream Ciphers

One-time pad using a RNG

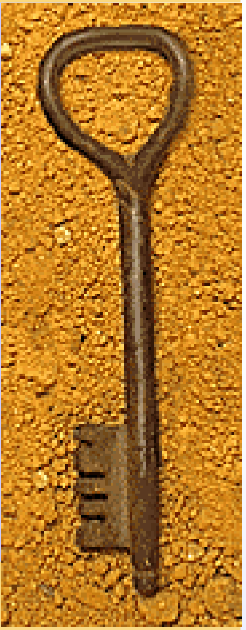
- Use k as seed? $E_k(m) = m \oplus RNG(k)$
  - Reuse problem!
- Typical usage (e.g., with DES)

$$E_k(m) = \underbrace{DES_k(s)}_{\text{strong}} \, , \, \underbrace{m \oplus RNG(s)}_{\text{fast}}$$

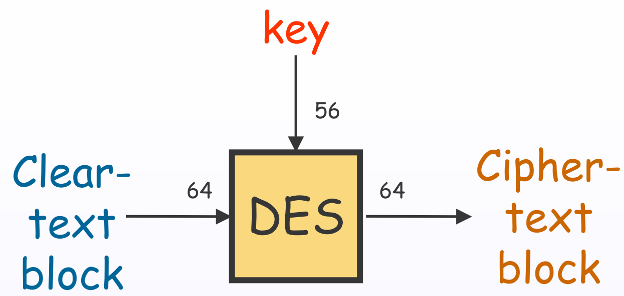
  - Chose new s each time

DES - Data Encryption Standard

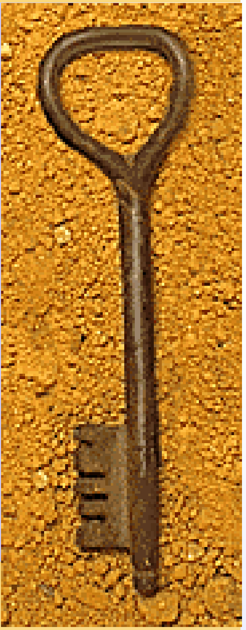
[NIST/IBM/NSA, released 1975]



- Message blocks: 64 bits
- Keys: 56 bits



- Speed
  - Software: 43,000 block/sec ~ 2.7 Mbit/sec
    - Measured on an old 80486 at 66MHz
    - OK for files and web pages
    - Too slow for sound and video
  - Hardware: 16.8 million block/sec ~ 1 Gbit/sec
    - High speed Ethernet: 100 Mbit/sec
    - Modem: 56 Kbit/sec



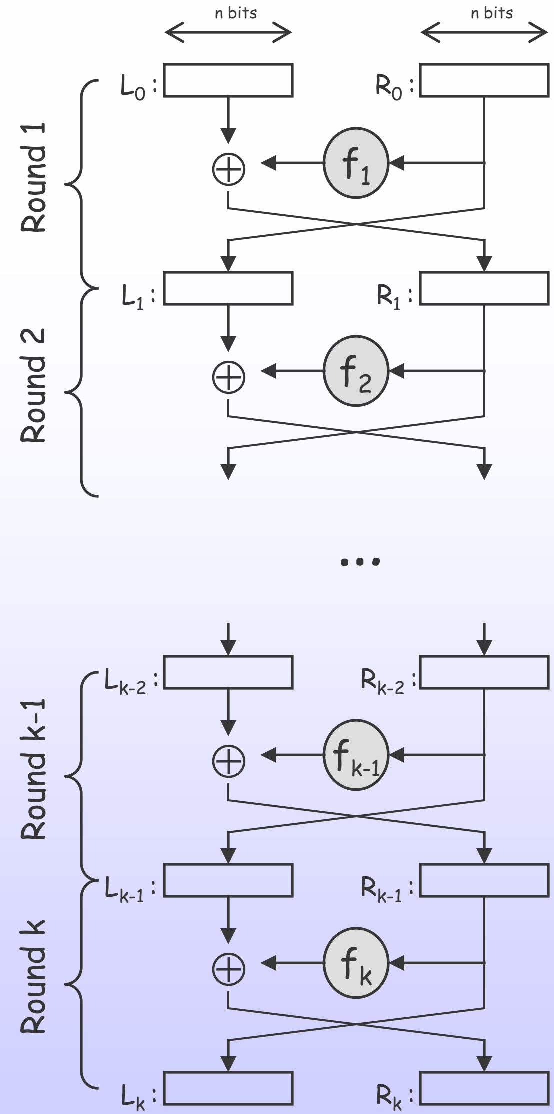
Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

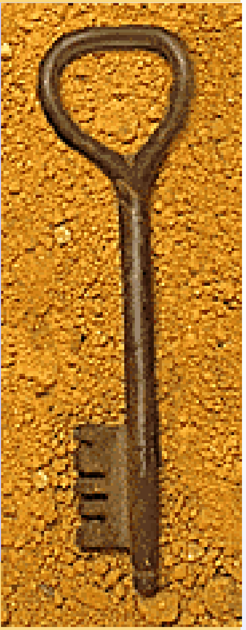
Feistel Networks

$$f_1, \ldots, f_k : \{0,1\}^n \rightarrow \{0,1\}^n$$

- Arbitrary functions
- Not necessarily invertible

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus f_i(R_{i-1}) \end{cases}$$





Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

Inverting a Feistel Network

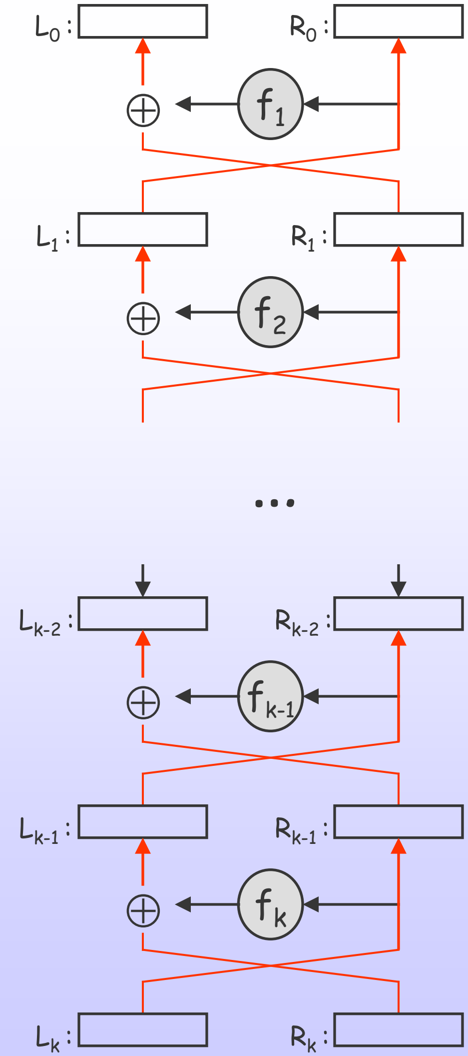
Theorem

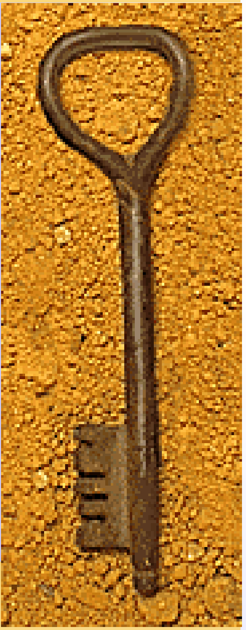
For any $f_1, \ldots, f_k : \{0,1\}^n \to \{0,1\}^n$,
a Feistel network computes a
permutation $\pi : \{0,1\}^n \to \{0,1\}^n$

Inverse: $\begin{cases} L_{i-1} = R_i \oplus f_i(L_i) \\ R_{i-1} = L_i \end{cases}$

Feistel networks convert

- generic functions
- into permutations





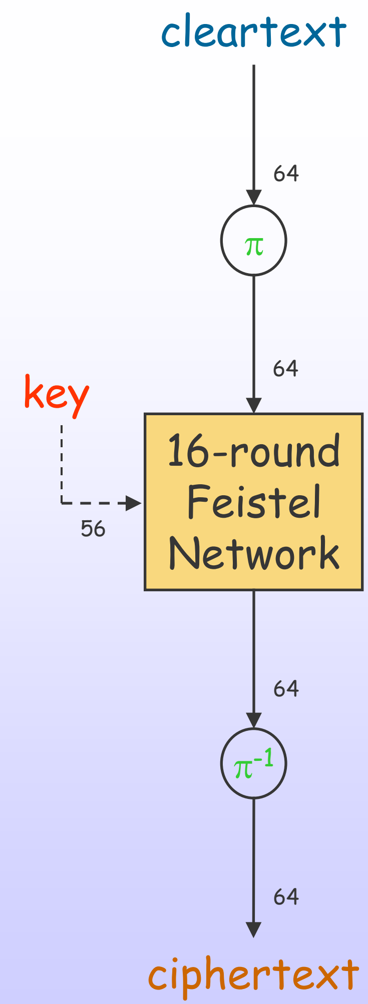
Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

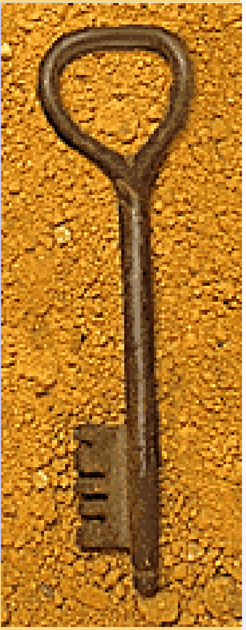
Inside DES

DES is a Feistel network with

- 16 rounds
- 64 bit cleartext blocks
- 56 bits key
- $f_1, \ldots, f_{16}$ derived from key
- Initial permutation π (public)

- Decryption
  - Apply $f_{16}, \ldots, f_1$ (in reverse order)
  - Same chip



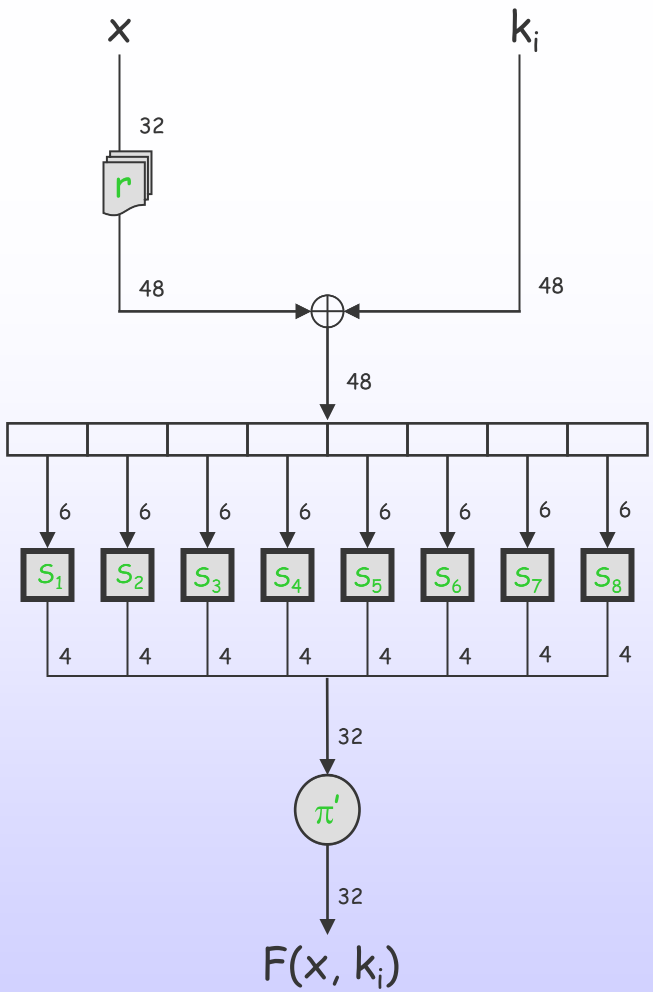


Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

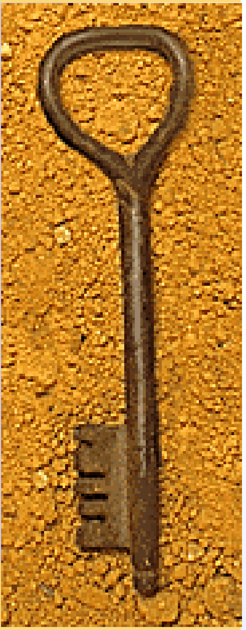
The Functions f_i

$$f_i(x) = F(x, k_i)$$

- 48 bits: k_i derived from k (56 bits)
  - Public key schedule
- $F: \{0,1\}^{32} \times \{0,1\}^{48} \rightarrow \{0,1\}^{32}$ is public
  - ½ block x (32 bits) expanded to x' (48 bits)
    - Public replicator r
  - S-boxes S_j (6 bits → 4 bits) are public
    - … where the magic happens
    - Rationale was kept secret
  - Final permutation π' is public
    - Shuffles input for next round



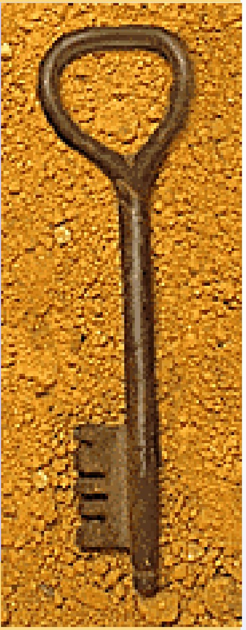
Attacks on DES



Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

- Exhaustive search
  - Given plaintext m and ciphertext x, *with high probability* there is a single key k s.t.
    x = DES(m,k)
  - Trying 10^6 keys/sec, it takes 2,000 years
- However …
  - 1993, $\$10^6$ homemade supercomputer breaks DES in 7 hours (CPA)
- More sophisticated attacks
  - Use properties (e.g. $DES(\overline{m},\overline{k}) = \overline{DES(m,k)}$)
  - Linear / differential crypto-analysis

Avoiding Exhaustive Search–3DES



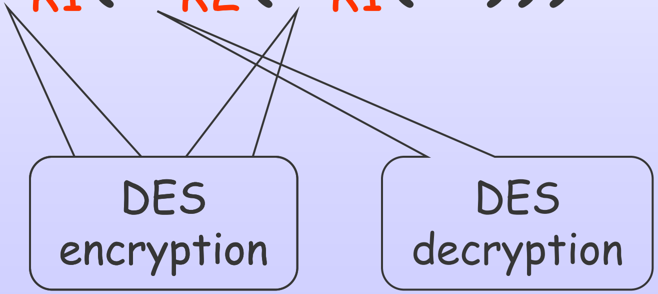
Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

DES is not a group

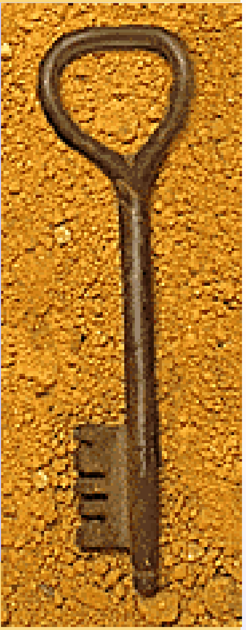
- Given k1, k2, with high probability there is no k3 s.t.

$$E_{k1}(E_{k2}(m)) = E_{k3}(m) \text{ for every } m$$

$$3DES_{k1,k2}(m) = E_{k1}(D_{k2}(E_{k1}(m)))$$



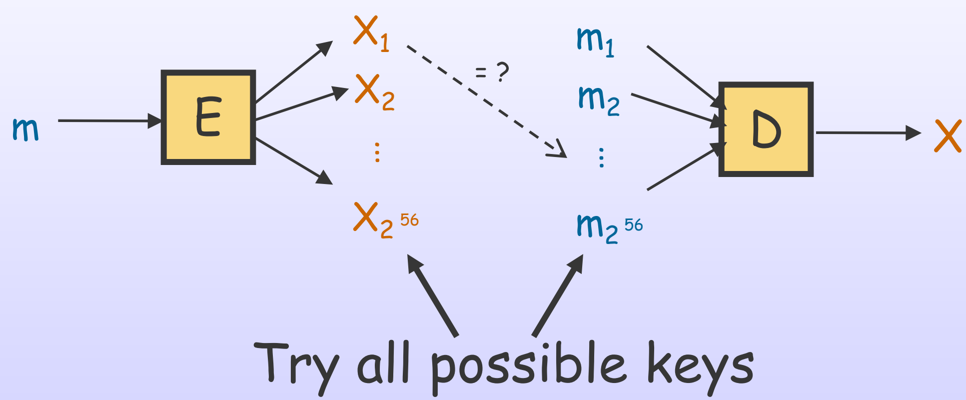
- Key length: 112 bits
- Very popular



How about a 2DES?

$$2DES_{k1,k2}(m) = E_{k1}(\,E_{k2}(m))\ \ ??$$

- Meet-in-the-middle attack!



For key length n, total work is "only" $2^n + 2^n = 2^{n+1}$

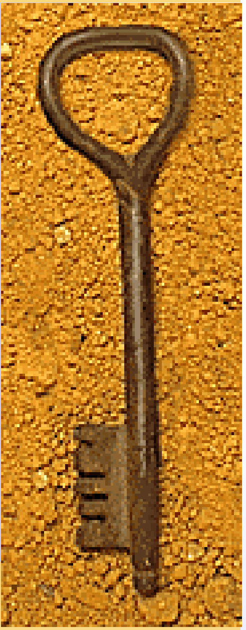
- Effective key length is just 57 bits!
- Applies to any encryption algorithm

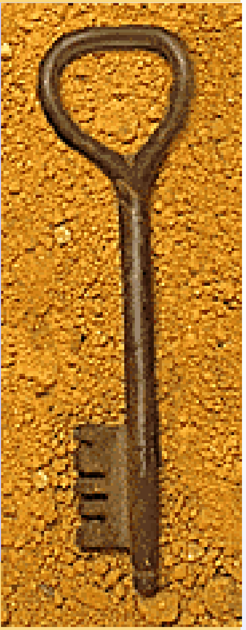
DESX

DES encryption

$$DESX_{k1,k2,k3}(m) = k1 \oplus E_{k2}(m \oplus k3)$$

- Key length: 56 + 2*64 = 184 bits
- However, effective key length is only about 100 bits



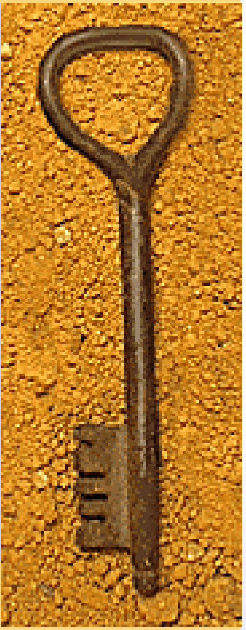


Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

AES – a Successor to DES

Advanced Encryption Standard

- 1996: NIST issues public call for proposal
  - Secure for next 50-100 years
  - Block cipher faster than 3DES
  - Variable key lengths (128, 192, 256, … bits)
  - Open design
- 15 algorithms submitted
  - Public (and private) crypto-analysis for 4 years
  - 5 finalists



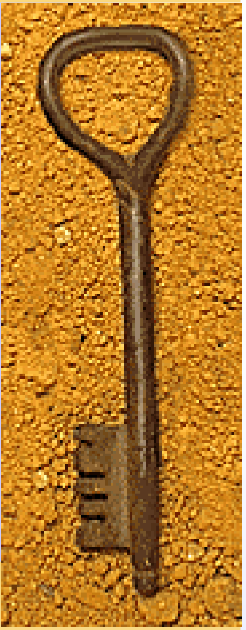
Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

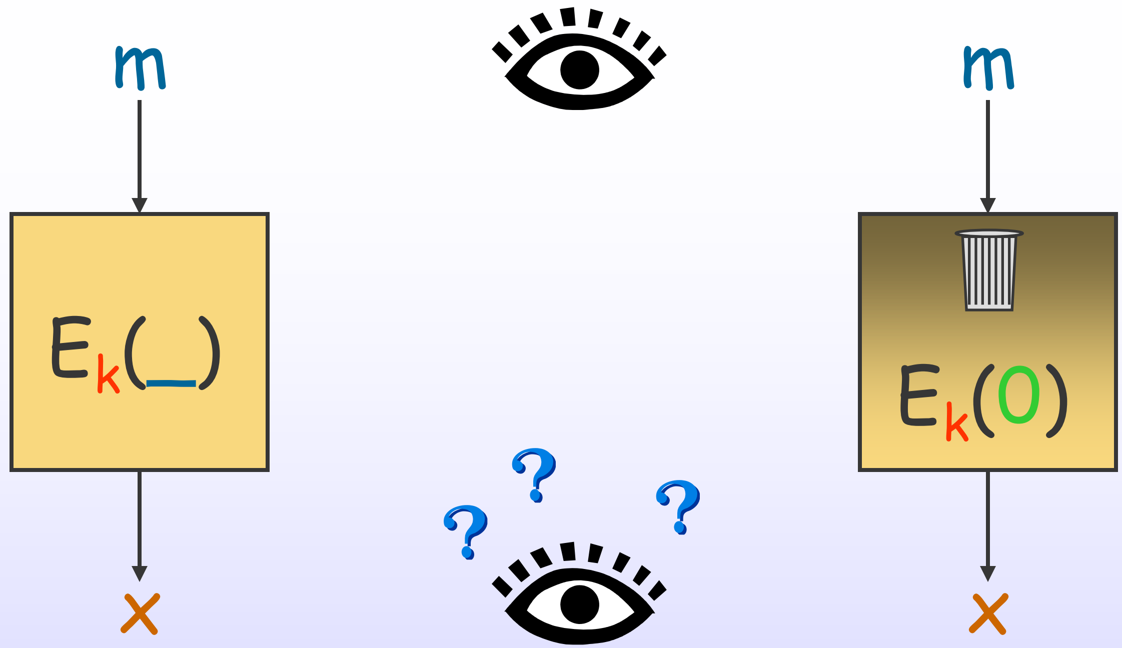
Oct. 2000: AES Contest Winner

Rijndael, by J. Daemen and V. Rijmen

- Fast (~18-20 cycles to encrypt a byte)
- Small (98 Kb)
- Well understood characteristics
  - Bit operations: ⊕, shift, …
- Provides good safety (1.33 safety factor)

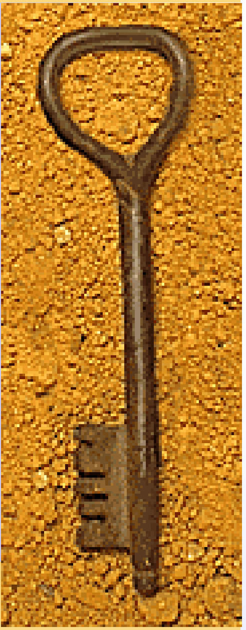
When is a Cipher Secure?





Polynomial adversary cannot tell a real encryption box from a fake one

Formal Definition



Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

Let

- $E: \{0,1\}^n \times \{0,1\}^l \rightarrow \{0,1\}^n$
- $A(x \leftrightarrow m) = 1$ iff $x = E_k(m)$
  - A algorithm polynomial in key length l
- $x_m = E_k(m)$

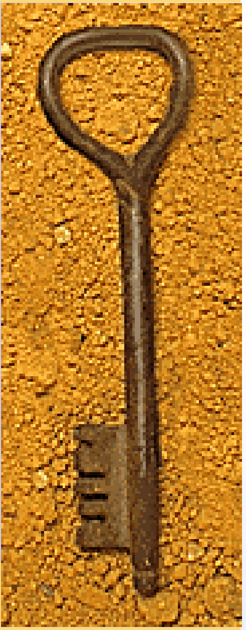
E is a secure encryption scheme if

$\forall$ polynomial p(_)

$\exists$ L s.t. $\forall$ $l > L$

$\forall$ $k \in \{0,1\}^l$

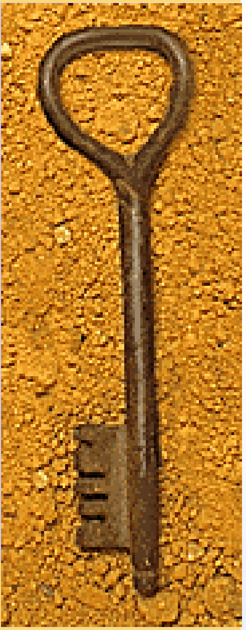
$$Pr[A(x_m \leftrightarrow m) = 1] - Pr[A(x_0 \leftrightarrow m) = 1] < 1/p(l)$$



Readings

- Andrea Sgarro, *Codici Segreti*, 1989
- David Kahn, *The Code-Breakers*, 1996
- A. Menezes, P. van Oorschot and S. Vanstone, *The Handbook of Applied Cryptography*, 1996

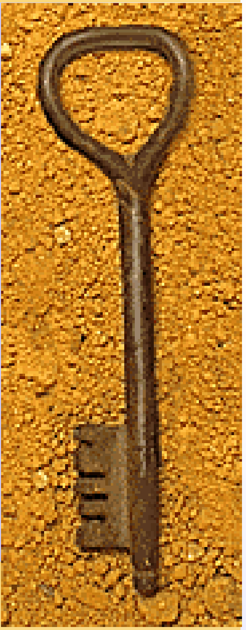
"The comprehensive History of Secret Communication from Ancient Times to the Internet"



Exercises for Lecture 2

Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

- Find a way to measure the redundancy in the ASCII rendering of English (or Italian) text
- Prove the invertibility of a Feistel network
- Why is 3DES immune from the meet-in-the-middle attack?
  - Can you explain why 3DES uses only 2 keys?
  - What is the cost of breaking y iterated encryptions with different keys?



Goals
History
Shared-Key
Attacks
Block C.
Stream C.
DES
Secure C.

Next ...

- Public-Key Cryptography