Stream Cipher: One Time Pad (OTP)

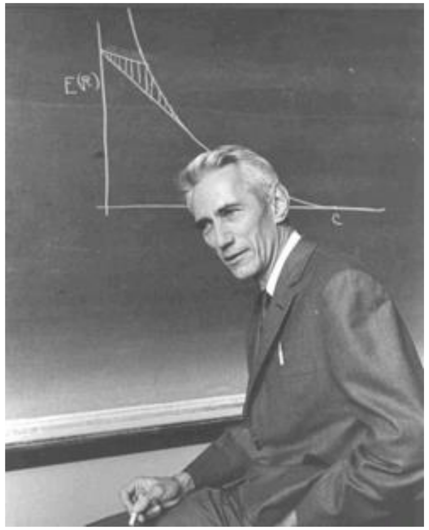
- Stream ciphers encrypt one bit or one byte at a time
- Gilbert S. Vernam (1917)
- OTP: Key is only used to encrypt one message

Key:	0	1	0	1	1	1	0	0	1	0	$\oplus$
Plaintext:	1	1	0	0	0	1	1	0	0	0	
Ciphertext:	1	0	0	1	1	0	1	0	1	0	

Encryption: $c = E(k, m) = m \oplus k$

Decryption: $D(k, c) = c \oplus k = (m \oplus k) \oplus k = m$

OTP Security

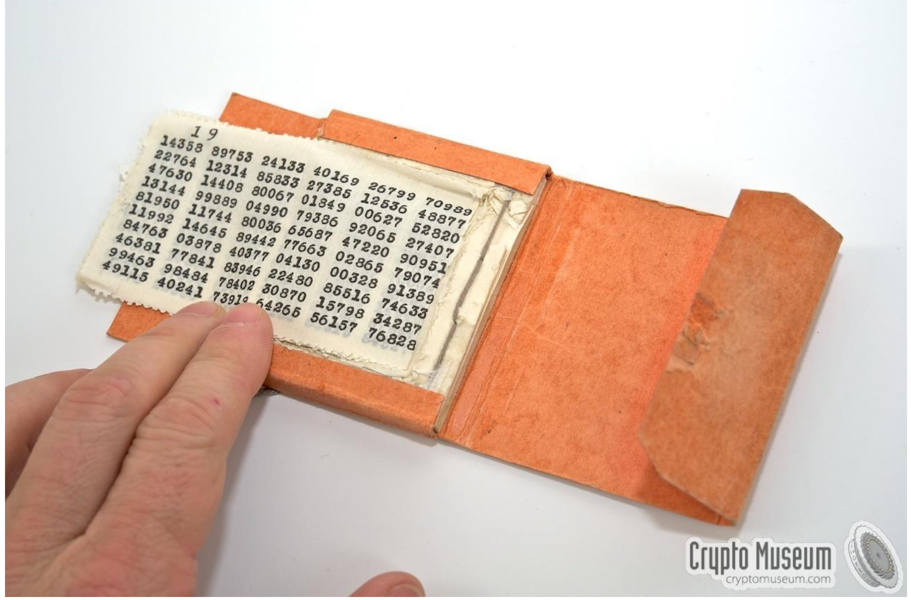


- *Shannon’s Perfect Secrecy (1949)*
- We say a cryptosystem has perfect secrecy if
  - Pr (P=x | C=y) = Pr (P=x) for all x,y
  - P: plaintex, C: ciphertext

- The probability that the plaintext is *x* given that you have observed ciphertext *y is the same as* the probability that the plaintext is x (without seeing the ciphertext)

- OTP is *“perfectly secure”*

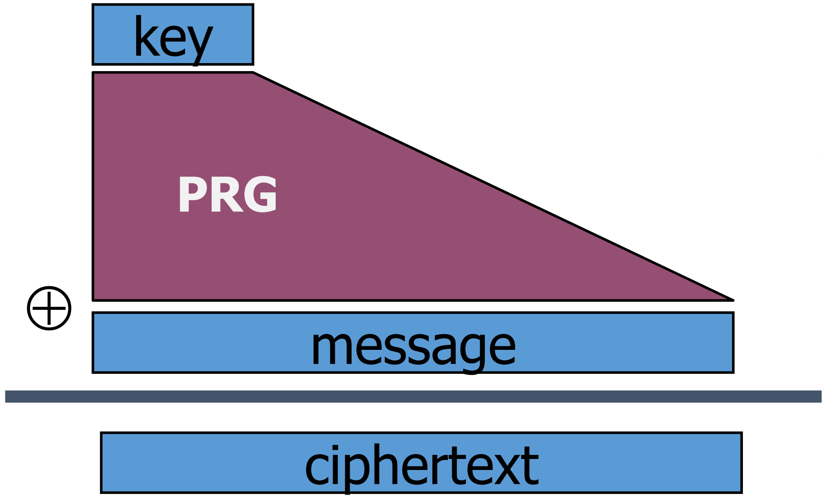
OTP Drawbacks



- Perfectly secure but impractical…
  - Require truly random one-time pads keys
  - Truly random value is difficult to generate
- Very long keys
  - *Need to be the same length of the message*
- Need a new key each time, high key exchange overhead
  - How to securely exchange the key? Bible? ☺
  - How to make sure the key is not (partially) repeated each time?

Stream ciphers using PRG

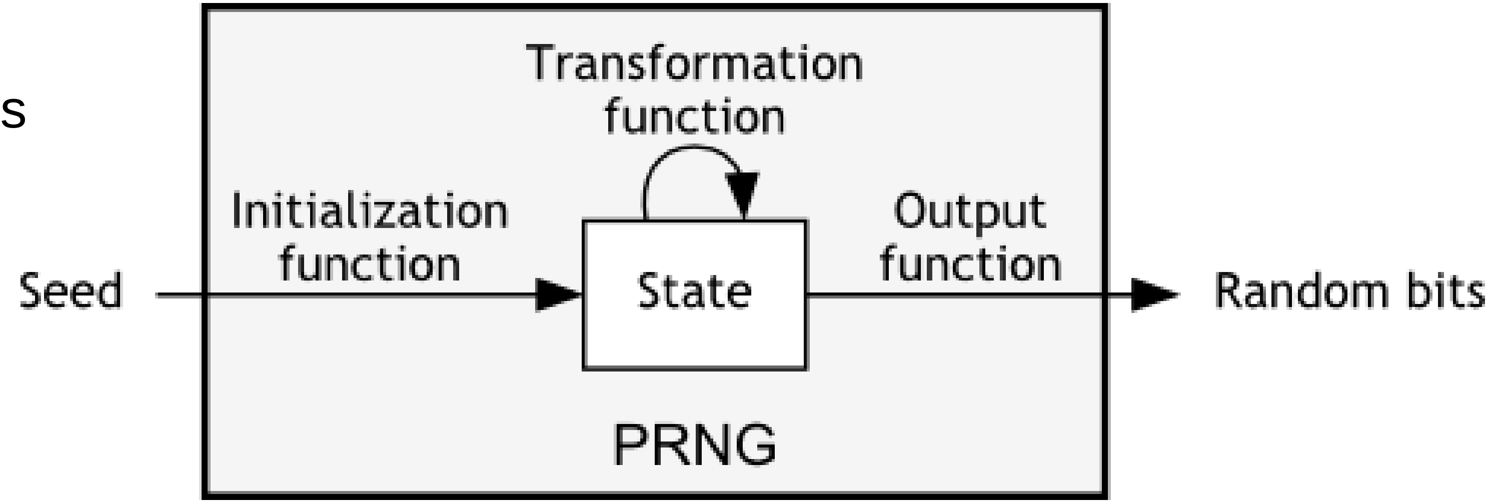
- Problem: OTP key is as long as the message
- Solution: *Pseudo random key* -- stream ciphers
- Examples: **ChaCha, Sosemanuk, RC4, …**



$c \leftarrow \textbf{PRG}(k) \oplus m$

Pseudorandom generator (PRG)

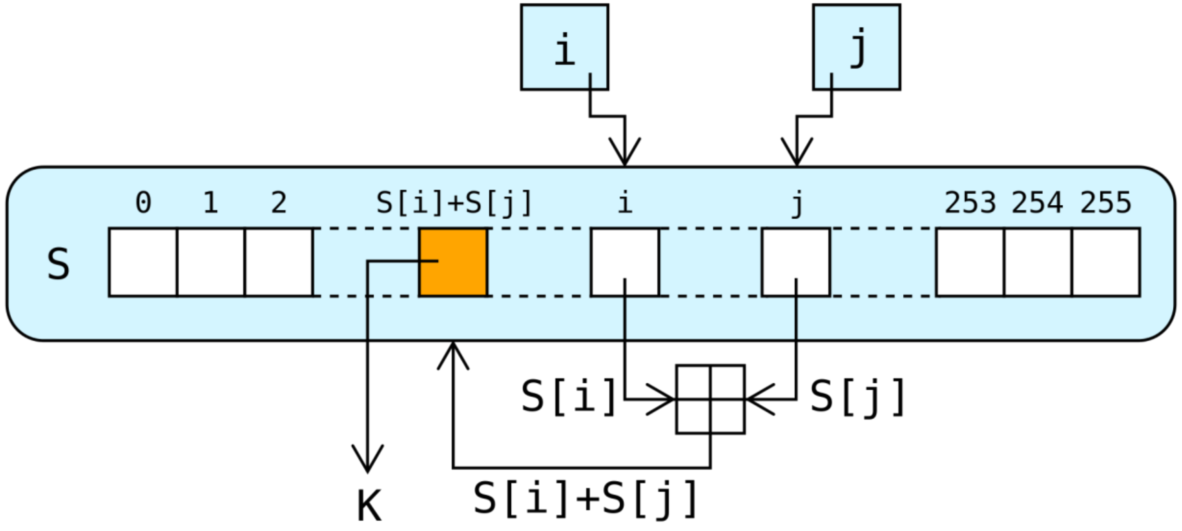
- PRG takes one number (called the *seed*) and produces a sequence of bits
- Sequence of bits is fully determined by the seed.
- This sequence is said to be pseudo-random if it passes a number of *statistical tests*, and thus *appears* random.
- Sample test
  - Measuring the frequency of bits and bit sequences
  - Evaluating entropy by trying to compress the sequence



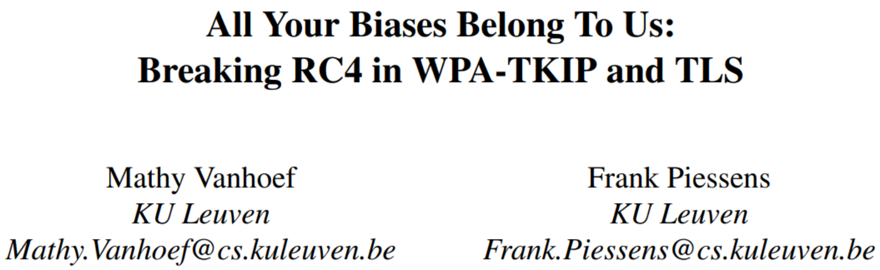
http://pit-claudel.fr/clement/blog/how-random-is-pseudo-random-testing-pseudo-random-number-generators-and-measuring-randomness/

RC4

- Rivest Crypto 4, invented in 1987 as secret of RSA (company), leaked in 1994
- Used in WEP, WPA, BitTorrent, SSL, SSH, Remote Desktop Protocol
- Key size (typical): 64, 128, 256
- Key => Key-scheduling algorithm (KSA) => State Vector => PRG => Key stream



PRG=> Key Stream



All Your Biases Belong To Us:
Breaking RC4 in WPA-TKIP and TLS

Mathy Vanhoef
KU Leuven
Mathy.Vanhoef@cs.kuleuven.be

Frank Piessens
KU Leuven
Frank.Piessens@cs.kuleuven.be

Not secure any more

https://blog.cryptographyengineering.com/2011/12/15/whats-deal-with-rc4/

Dangers in using stream ciphers

One time key !!   "Two time pad" is insecure:

$$c_1 \leftarrow m_1 \oplus PRG(k)$$
$$c_2 \leftarrow m_2 \oplus PRG(k)$$

Eavesdropper does:

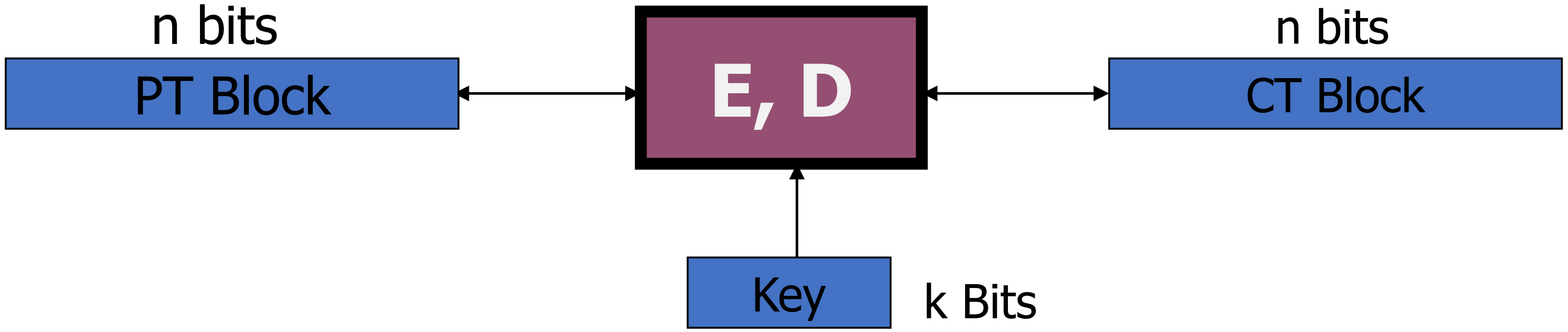
$$c_1 \oplus c_2 \quad \rightarrow \quad m_1 \oplus m_2$$

Enough redundant information in English that:

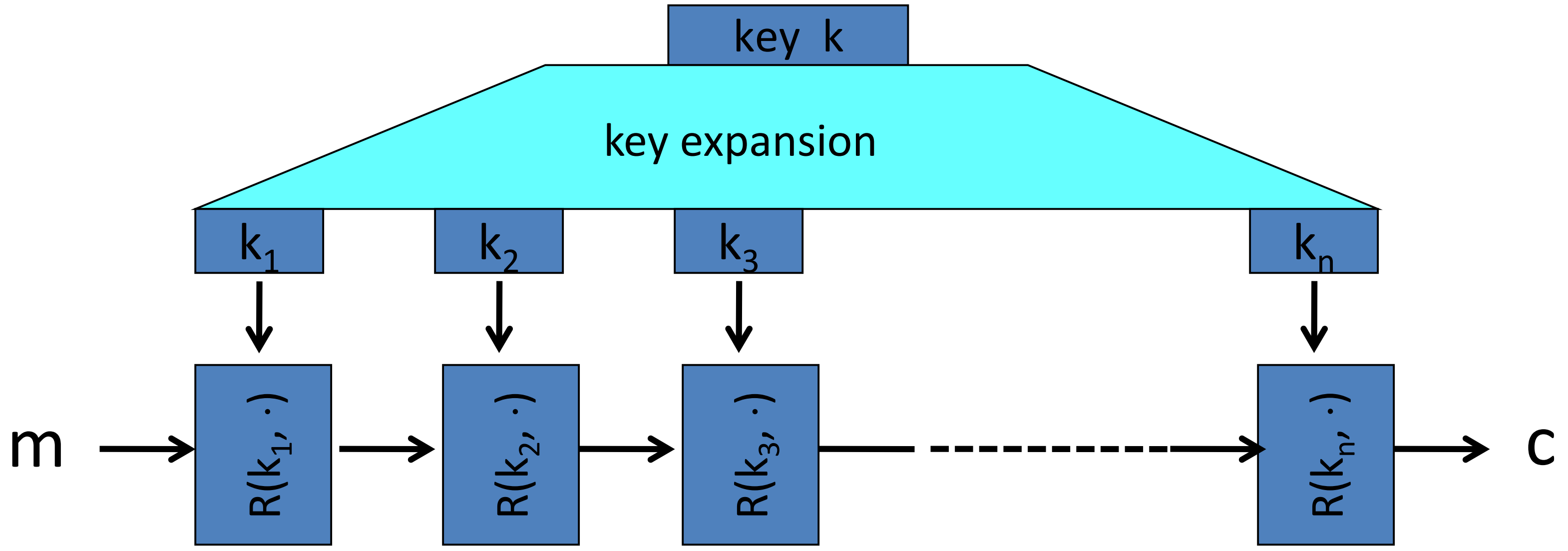
$$m_1 \oplus m_2 \quad \rightarrow \quad m_1 , \; m_2$$

Block ciphers

- Block ciphers encrypt a fixed number of bits as a single chunk.
- *Padding* needed when bits can't fill a block
- Two prominent algorithms: *AES and DES*

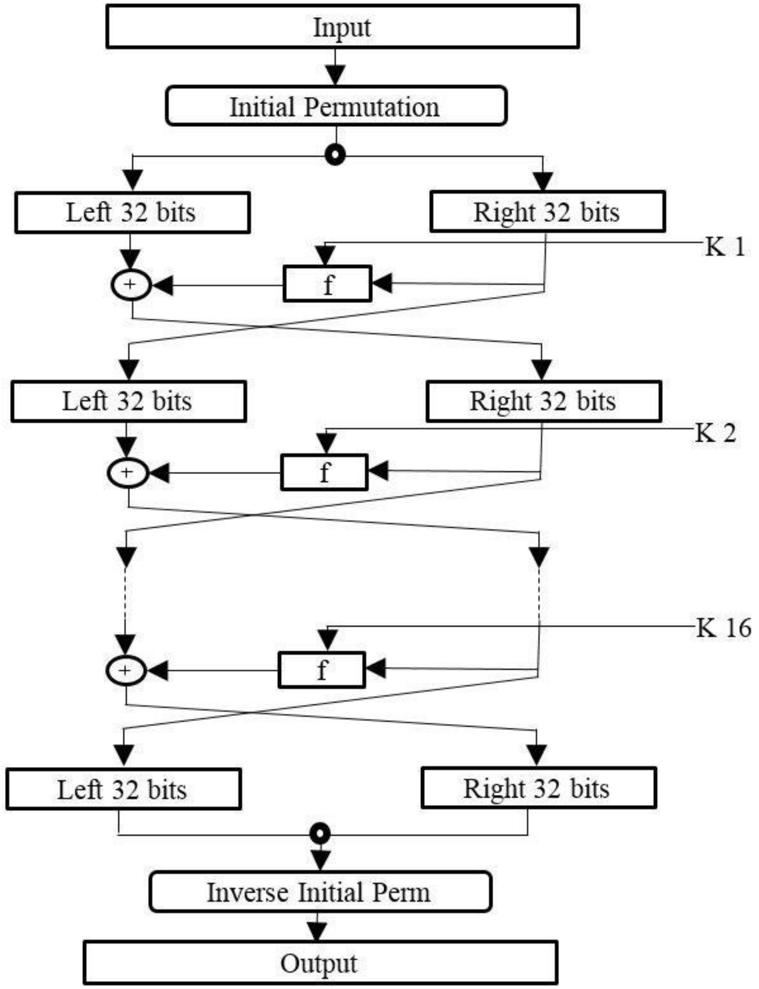


Block Ciphers Built by Iteration



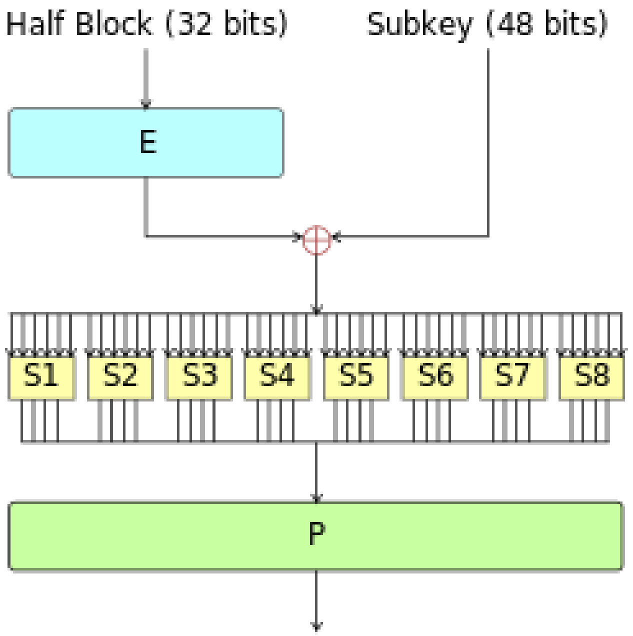
R(k,m): round function

DES: The Data Encryption Standard



Feistel Structure

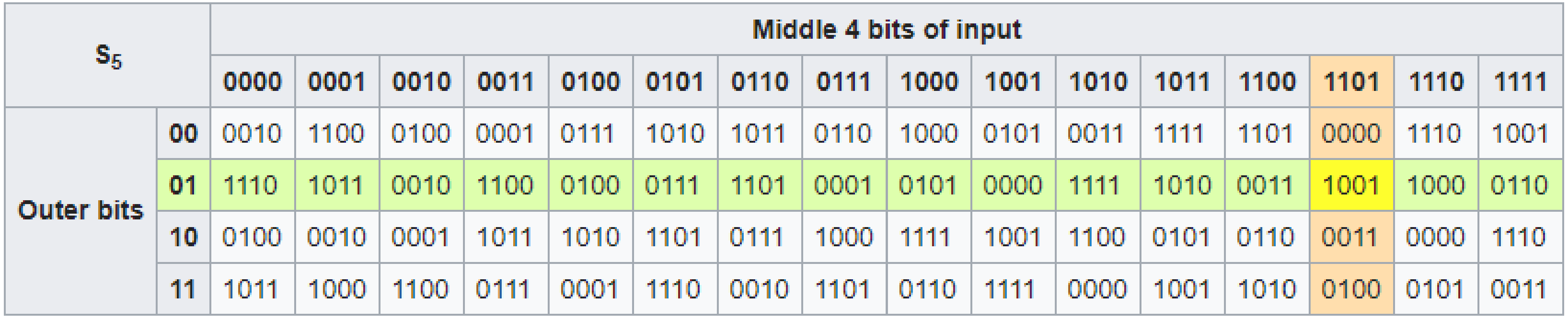
- Symmetric block cipher
- Developed in 1976 by IBM for the US National Institute of Standards and Technology (NIST)



The Feistel function (F-function)

S-box

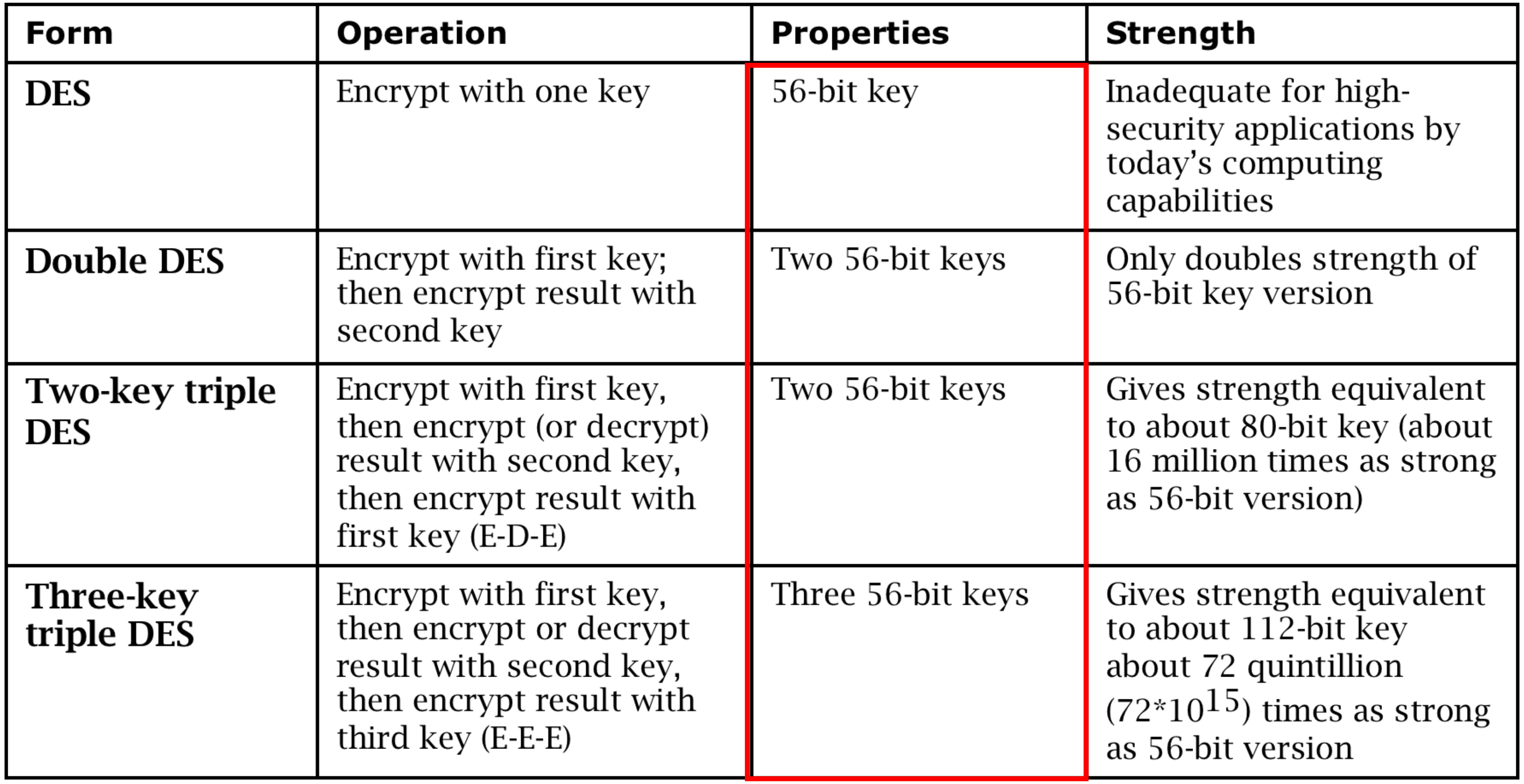
- Used to obscure the relationship between the key and the ciphertext (confusion)
- 6-bit input (middle + outer) => 4-bit output
- Biham and Shamir found that even small modifications to an S-box could significantly weaken DES



S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

https://en.wikipedia.org/wiki/S-box

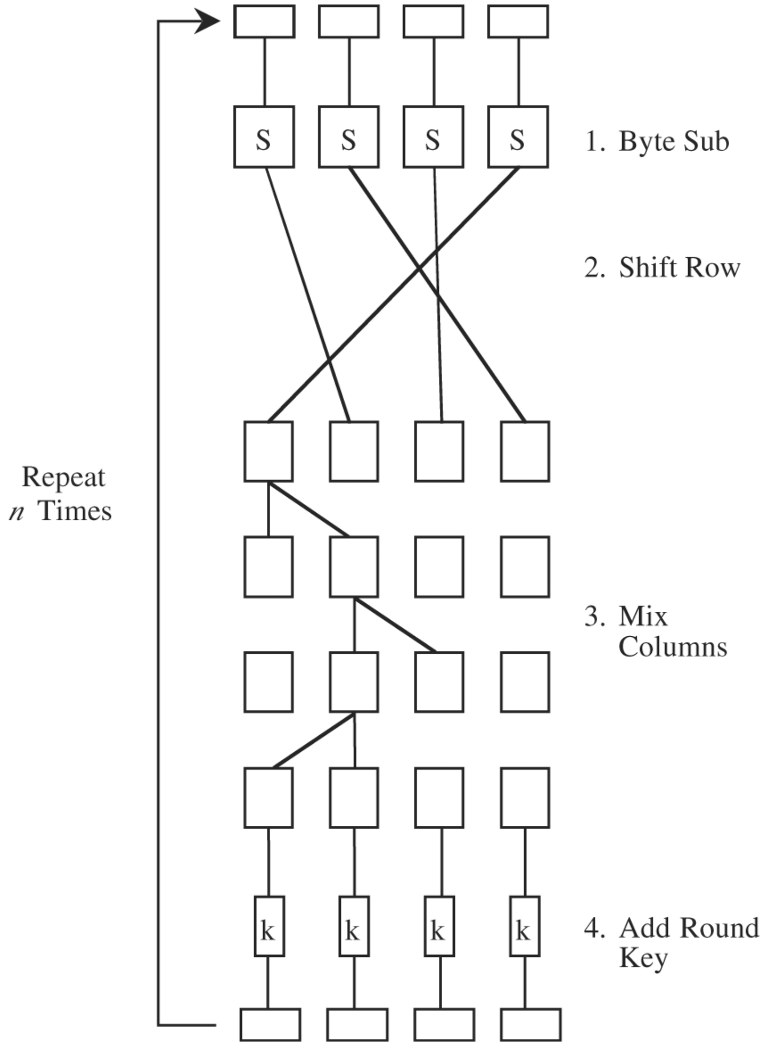
DES versions



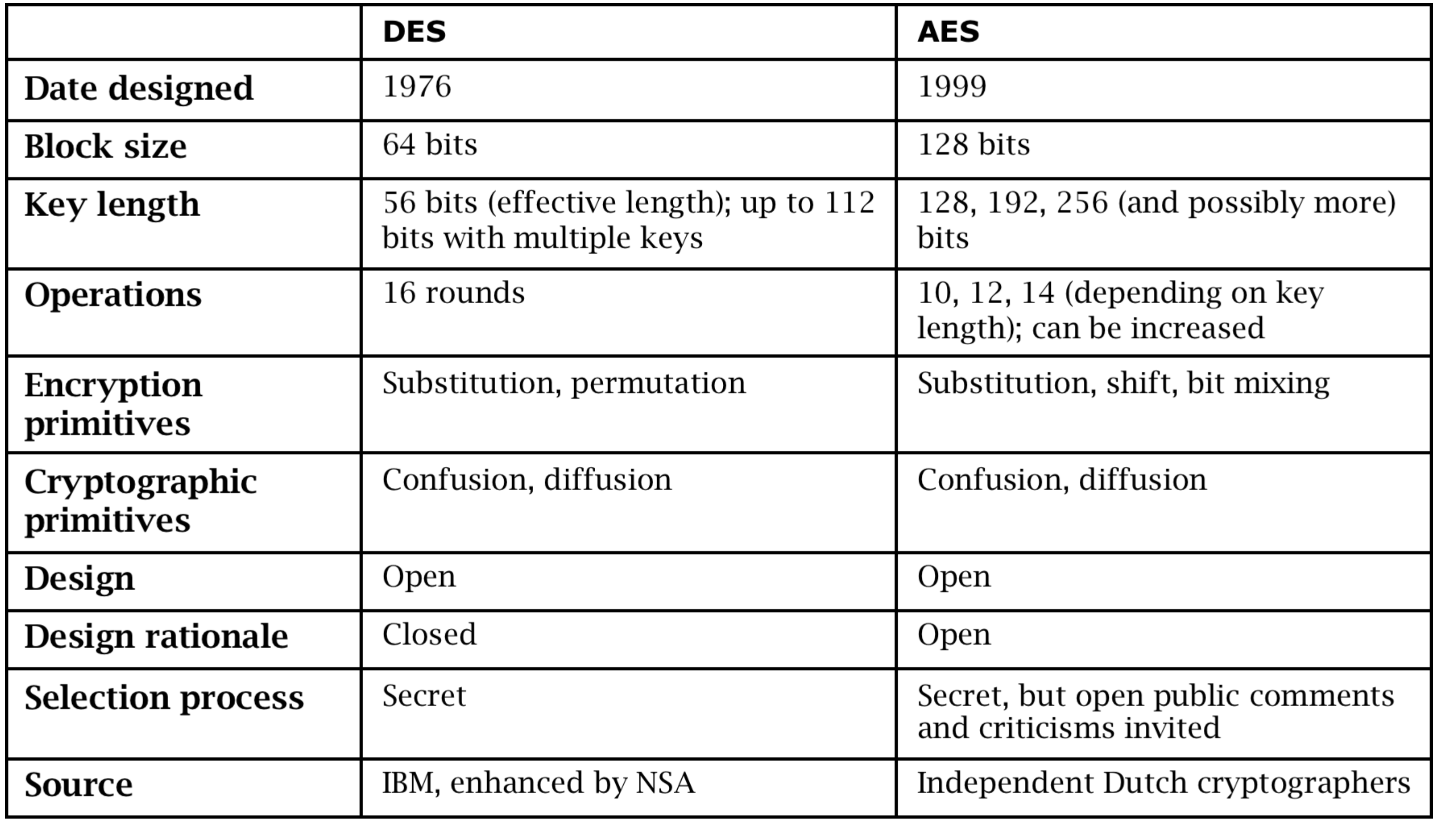
Form	Operation	Properties	Strength
DES	Encrypt with one key	56-bit key	Inadequate for high-security applications by today's computing capabilities
Double DES	Encrypt with first key; then encrypt result with second key	Two 56-bit keys	Only doubles strength of 56-bit key version
Two-key triple DES	Encrypt with first key, then encrypt (or decrypt) result with second key, then encrypt result with first key (E-D-E)	Two 56-bit keys	Gives strength equivalent to about 80-bit key (about 16 million times as strong as 56-bit version)
Three-key triple DES	Encrypt with first key, then encrypt or decrypt result with second key, then encrypt result with third key (E-E-E)	Three 56-bit keys	Gives strength equivalent to about 112-bit key about 72 quintillion ($72*10^{15}$) times as strong as 56-bit version

AES: Advanced Encryption System

- Other name: Rijndael
- Developed in 1999 by independent Dutch cryptographers
- Standardized in 2001 by NIST
- Still in common use
- 10, 12, 14 cycles for keys of 128, 192 and 256 bits



DES vs. AES



	DES	AES
Date designed	1976	1999
Block size	64 bits	128 bits
Key length	56 bits (effective length); up to 112 bits with multiple keys	128, 192, 256 (and possibly more) bits
Operations	16 rounds	10, 12, 14 (depending on key length); can be increased
Encryption primitives	Substitution, permutation	Substitution, shift, bit mixing
Cryptographic primitives	Confusion, diffusion	Confusion, diffusion
Design	Open	Open
Design rationale	Closed	Open
Selection process	Secret	Secret, but open public comments and criticisms invited
Source	IBM, enhanced by NSA	Independent Dutch cryptographers

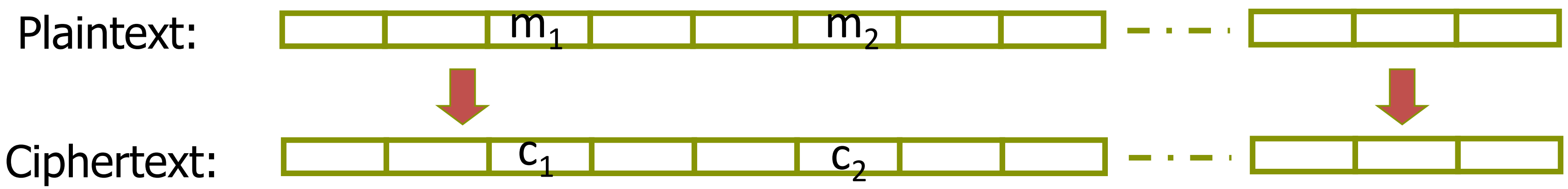
Modes of Operation

- Direct use of block ciphers is not very useful
  - Message typically takes multiple blocks
- How to repeatedly apply a block cipher to securely encrypt/decrypt arbitrary inputs
- Five standard modes
  - ECB: Electronic Code Book
  - CBC: Cipher Block Chaining
  - CFB: Cipher Feedback
  - OFB: Output Feedback
  - CTR: Counter

Chaining

ECB

Electronic Code Book (ECB):



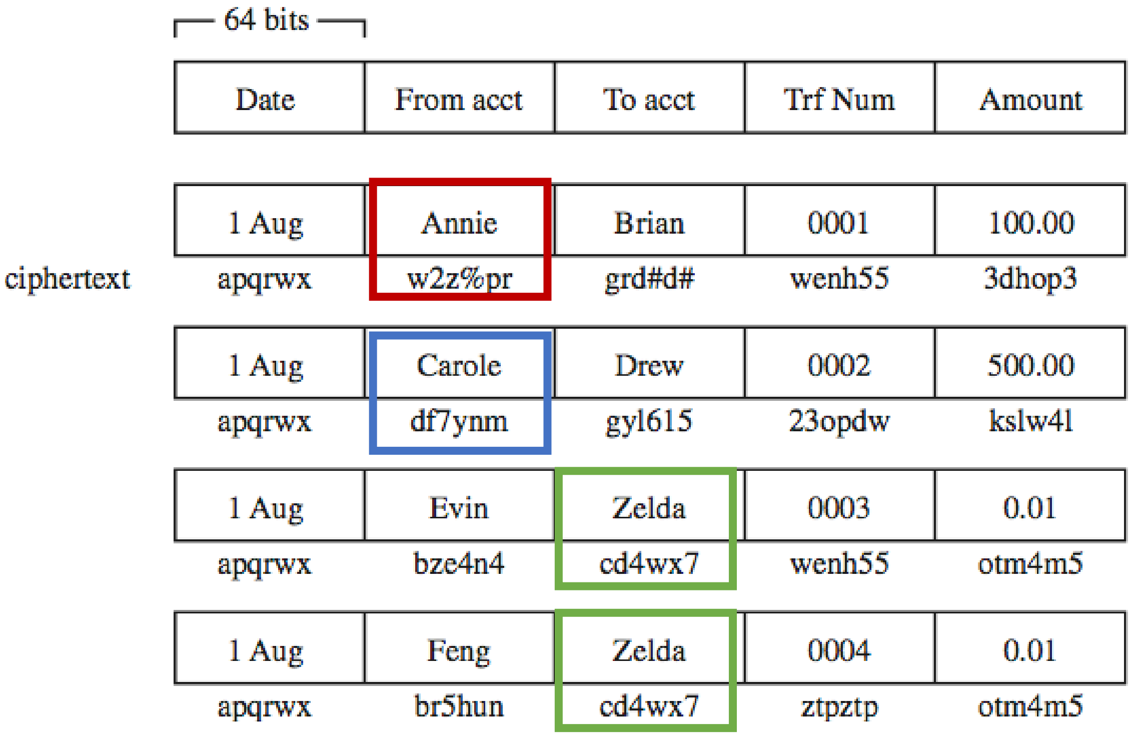
Problem:

- if $m_1=m_2$ then $c_1=c_2$

Don’t use ECB!

ECB weakness

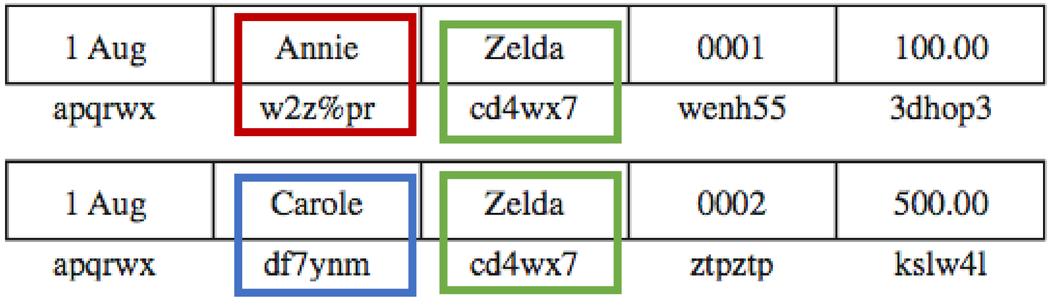
Data patterns may remain visible
Susceptible to replay attacks, block insertion/deletion



64 bits

	Date	From acct	To acct	Trf Num	Amount
	1 Aug	Annie	Brian	0001	100.00
ciphertext	apqrwx	w2z%pr	grd#d#	wenh55	3dhop3
	1 Aug	Carole	Drew	0002	500.00
	apqrwx	df7ynm	gyl615	23opdw	kslw4l
	1 Aug	Evin	Zelda	0003	0.01
	apqrwx	bze4n4	cd4wx7	wenh55	otm4m5
	1 Aug	Feng	Zelda	0004	0.01
	apqrwx	br5hun	cd4wx7	ztpztp	otm4m5

Ciphertext of bank transfer message



1 Aug	Annie	Zelda	0001	100.00
apqrwx	w2z%pr	cd4wx7	wenh55	3dhop3
1 Aug	Carole	Zelda	0002	500.00
apqrwx	df7ynm	cd4wx7	ztpztp	kslw4l

Zelda (adversary) fabricates messages to ask bank transfer money