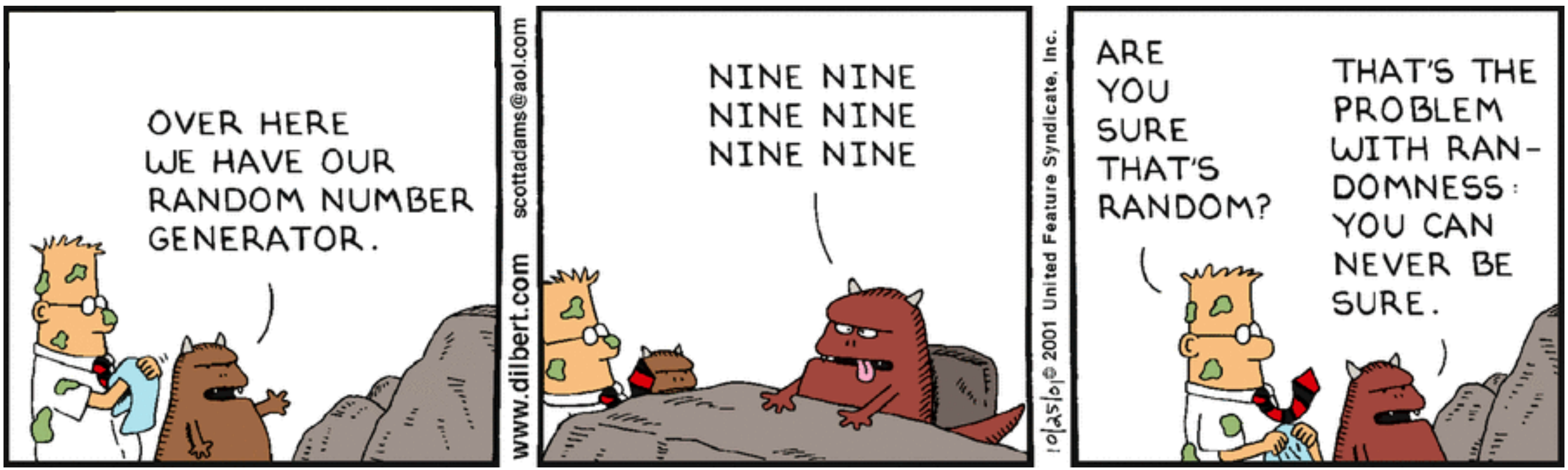
Block Ciphers

January 27, 2022



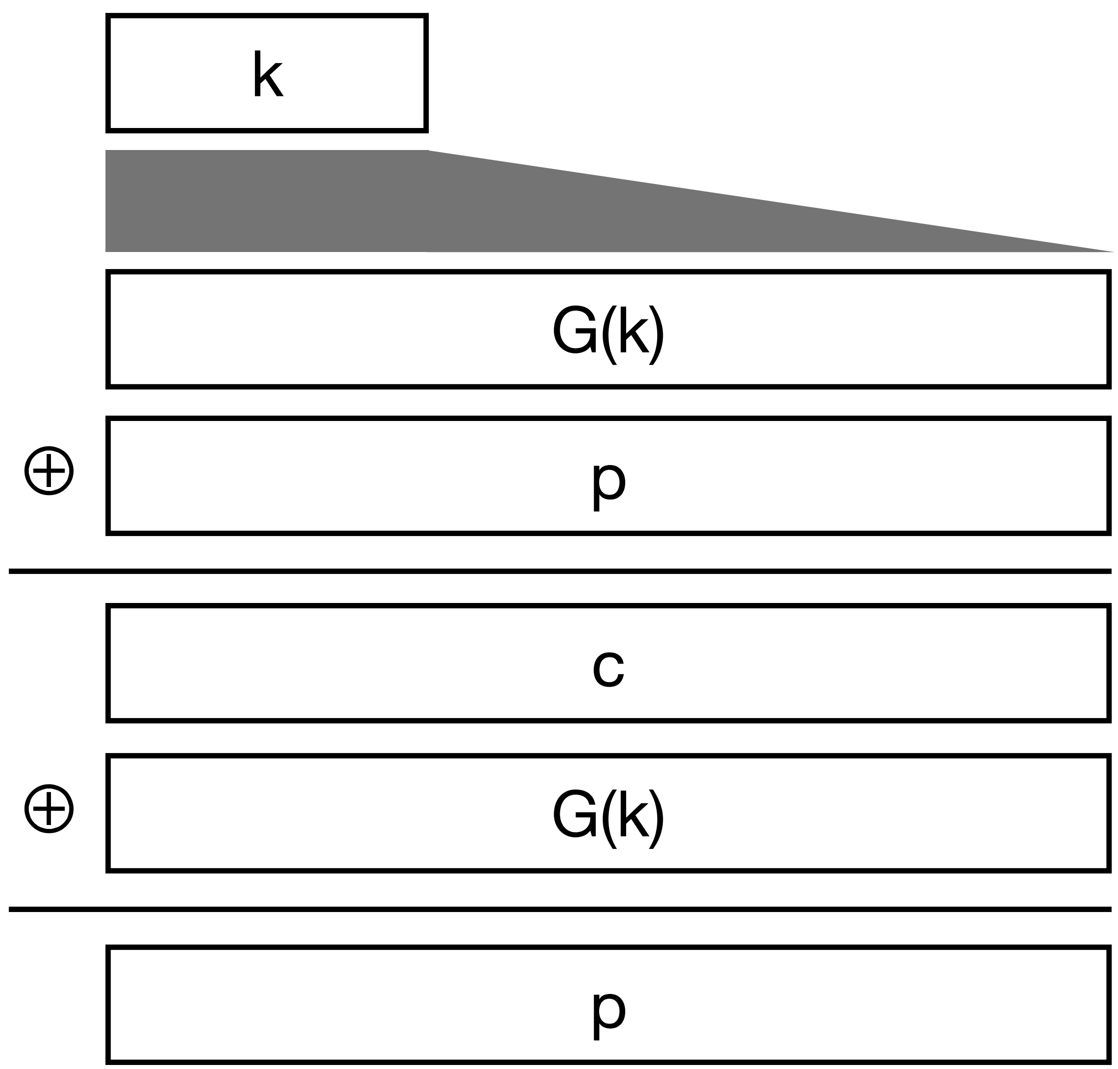
Today

- Stream cipher examples: RC4, **Salsa20 / ChaCha20**
- Block ciphers
  - *What is a block cipher?*
  - practical block ciphers: DES, **AES**

very important!

Feedback: `https://forms.gle/JGbNCmCsU69iWaTv8`

Reminder:
Stream Ciphers



- **encrypt** plaintext by **XORing** it to the pseudorandom sequence **bit-by-bit**
- **decrypt** ciphertext by **XORing** it to the pseudorandom sequence **bit-by-bit**

same as encryption

RC4 Cipher:
Old WiFi and Web Security

- Designed in 1987 by Ron Rivest for RSA Security, a security company
  - originally, it was kept a trade secret, but someone leaked it in 1994
- Advantages
  - **variable key length** (from 8 to 2048 bits)
  - very simple, based on **byte-oriented operations**:
    only eight to sixteen machine operations are required per output byte
- Applications
  - WiFI security: **WEP** (1997) and **WPA** (2003)

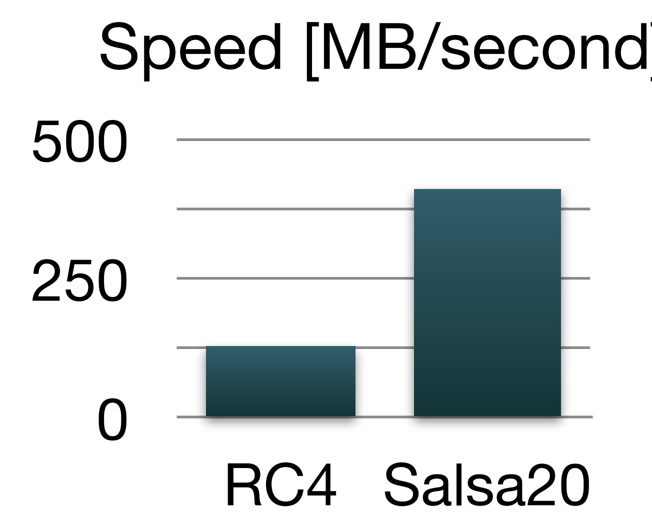
    ⚠ very practical attack found in 2001, WEP and WPA **deprecated in 2004**
  - web security (HTTPS): **SSL** (1995) / **TLS** (1999)

    ⚠ practical attack found in 2013, RC4 in SSL/TLS **deprecated in 2015**

RC4 had a good run, but it has been retired…

Salsa20 / ChaCha20 Cipher:
"State-of-the-Art" Stream Cipher

- Designed by Daniel Bernstein in 2005 (Salsa20) and 2008 (ChaCha20)
  - not patented, several public domain implementations
  - **ChaCha20** variant: more secure, more efficient
- Key length: **128 or 256 bits**
- Advantages
  - **fast software implementation** (simple 32-bit operations)
  - can seek to any position in the output sequence
  - **64-bit nonce is part of the algorithm** (to prevent key-reuse issues)
- <u>Security:</u> no significantly stronger attacks than brute force *(yet)*
- Adoption
  - Google implemented it in OpenSSL as a replacement for RC4
  - Linux (and some other operating systems) use it for random number generation
  - …

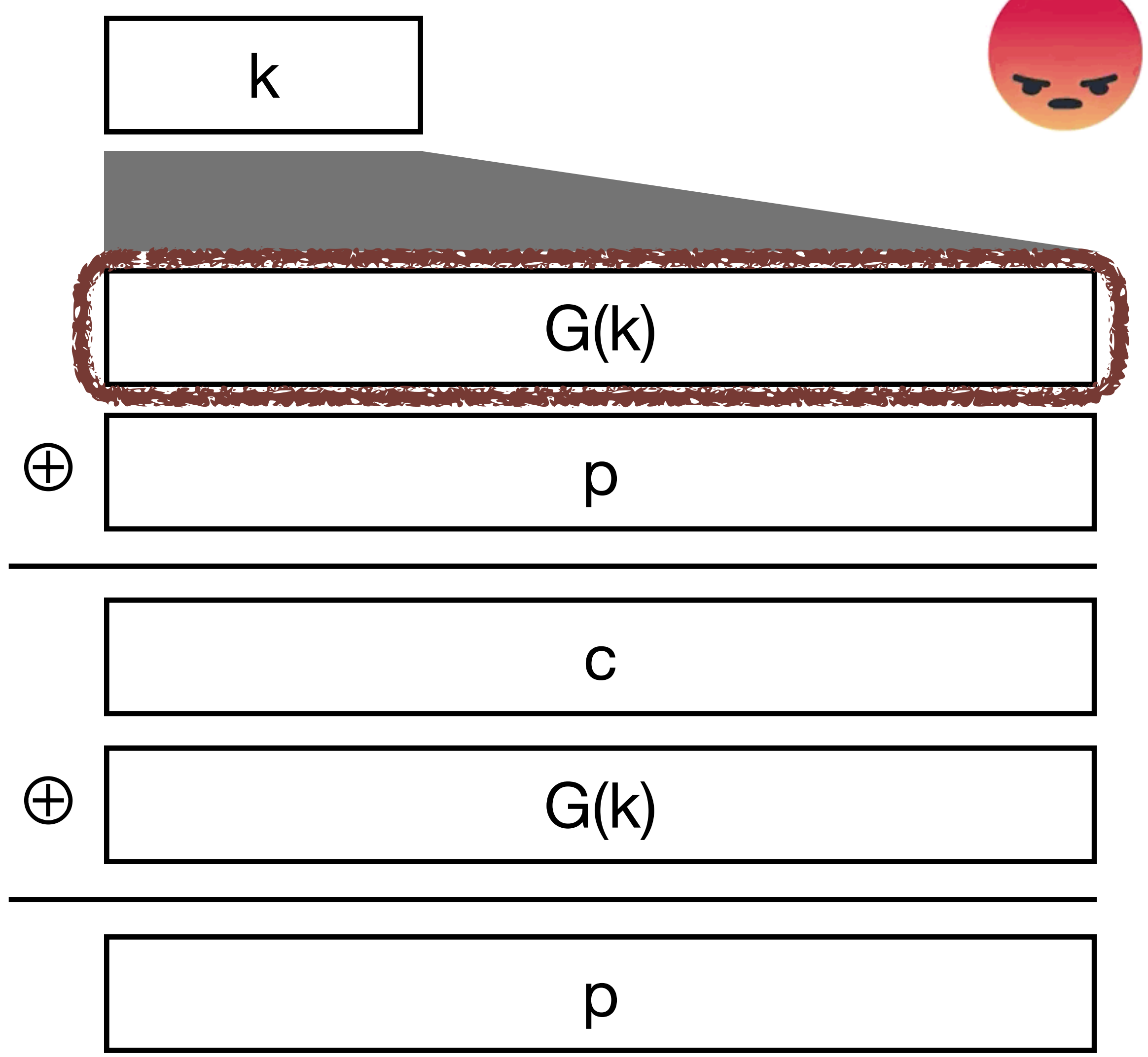


(Intel Core 2 1.83 GHz)

Salsa20 Cipher
Algorithm

- Generates its output in blocks of 16 x 32 bits
- Internal state: 16 x 32 bits
  - initialized using the key, the nonce (64 bits), and seek position (64 bits)
- Operations for updating the state:
  XOR, 32-bit addition mod 2^{32}, and rotating 32 bit values
- Salsa20 performs 20 rounds of XOR-add-rotate, each of which updates all values in the state
  - Salsa20/8 and Salsa20/12 perform only 8 and 12 rounds
- Finally, the state is added to the original state to obtain the output

Reminder:
Stream Ciphers



 generator does not need to be invertible

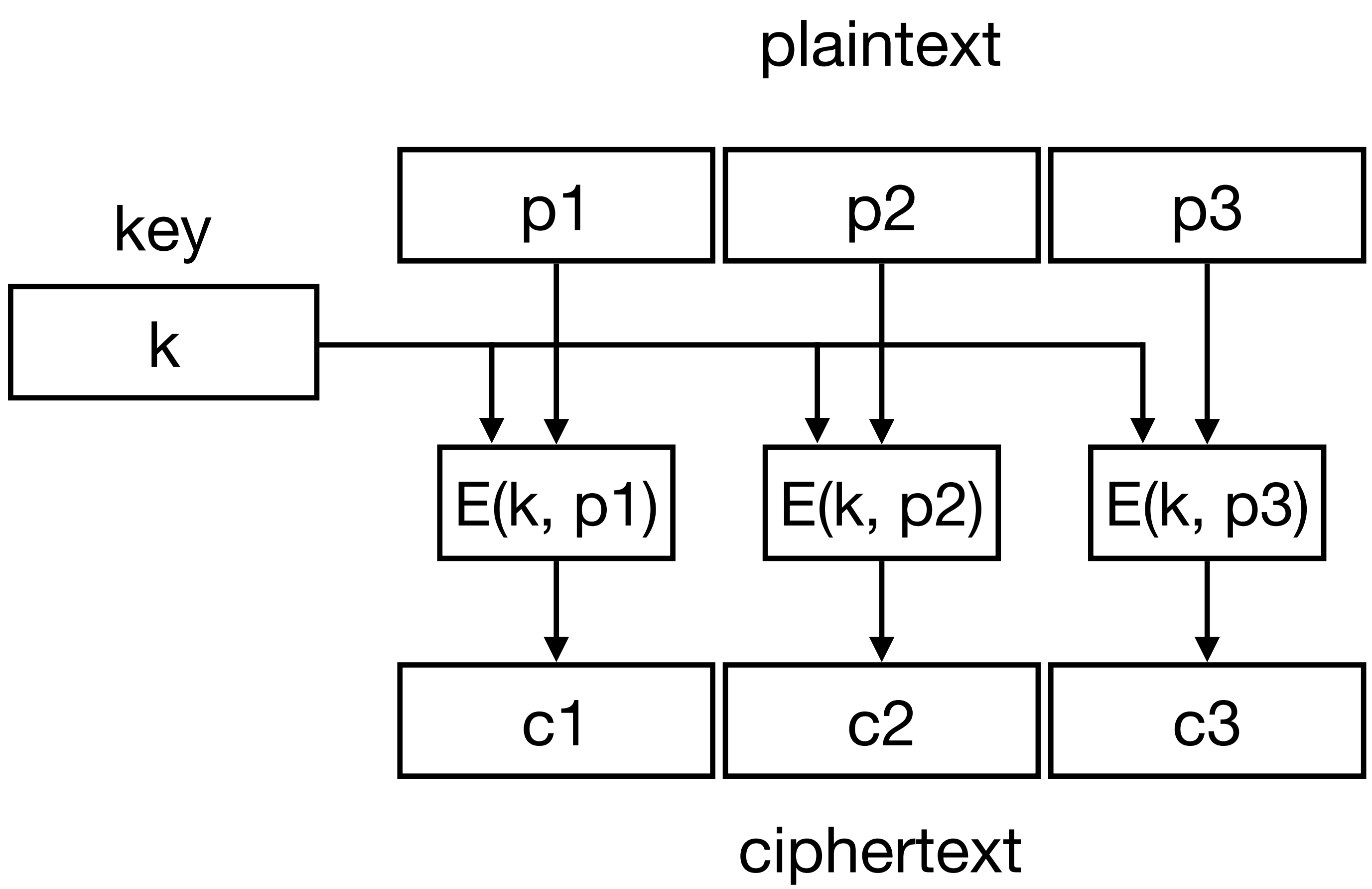
each sequence can be used only once

- **encrypt** plaintext by **XORing** it to the pseudorandom sequence **bit-by-bit**
- **decrypt** ciphertext by **XORing** it to the pseudorandom sequence **bit-by-bit**

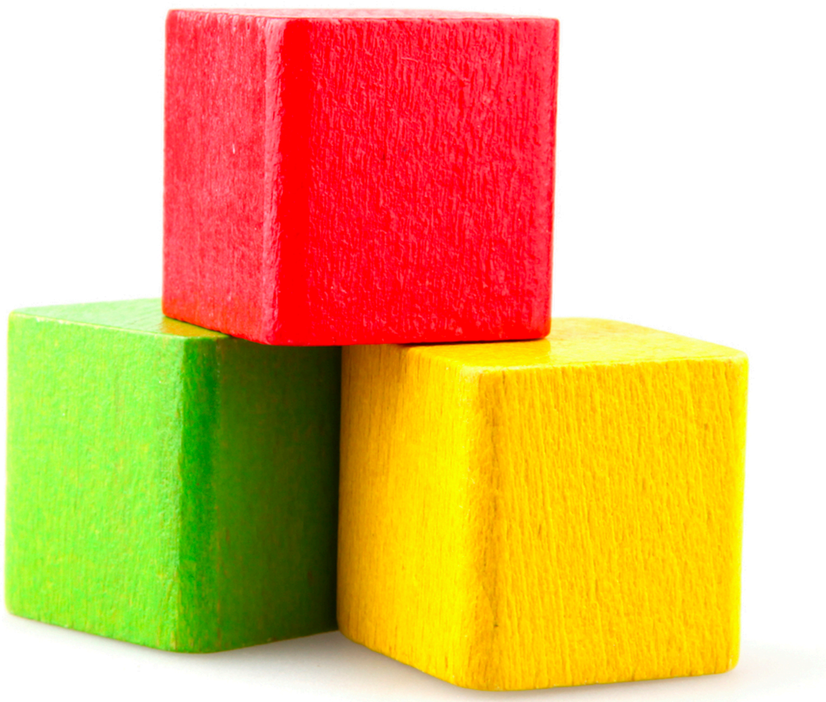
Block Ciphers

Block Ciphers

- Encrypt plaintext in **fixed-size blocks**

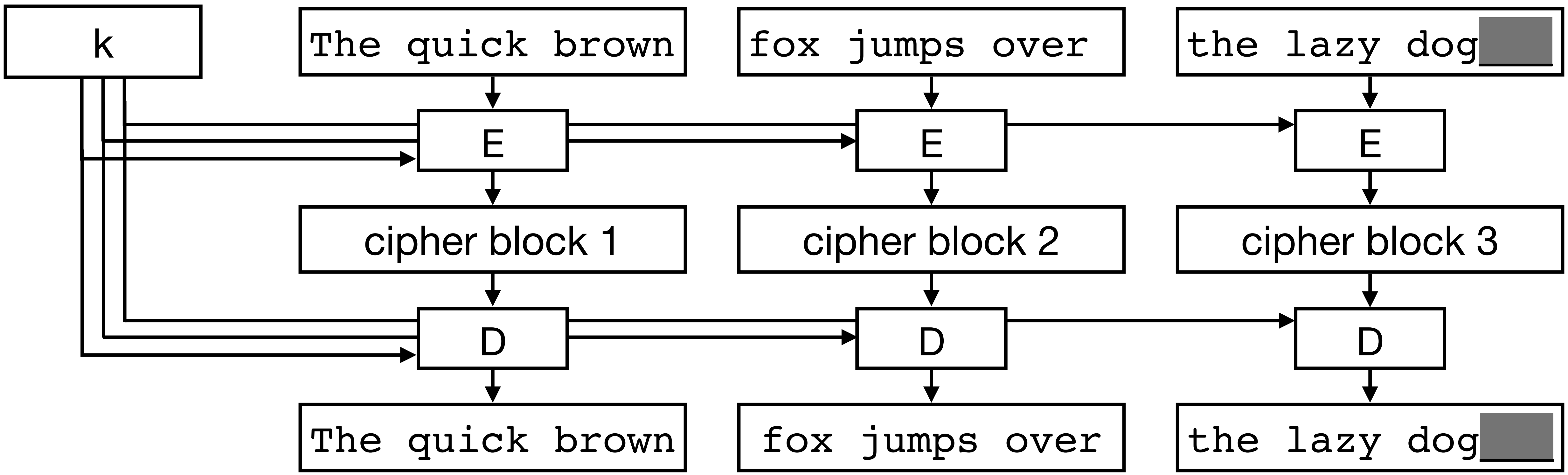


- Encryption/decryption are **different operations**



Block Cipher Example

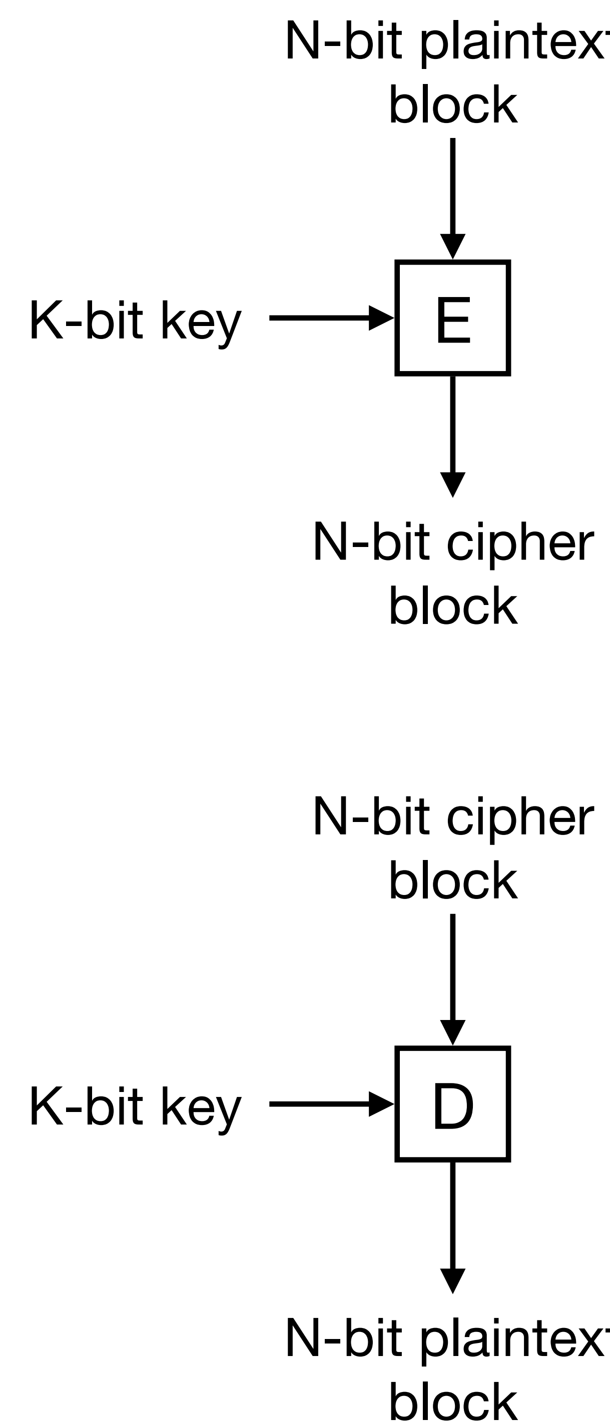
Example plaintext: “`The quick brown fox jumps over the lazy dog`”



- Key size K depends on the cipher
  - *example*: DES works with K = 56 bits
- Block size N depends on the cipher
  - *examples*: DES works with N = 64 bits, AES works with N = 128 bits
  - size of plaintext and ciphertext blocks is the same

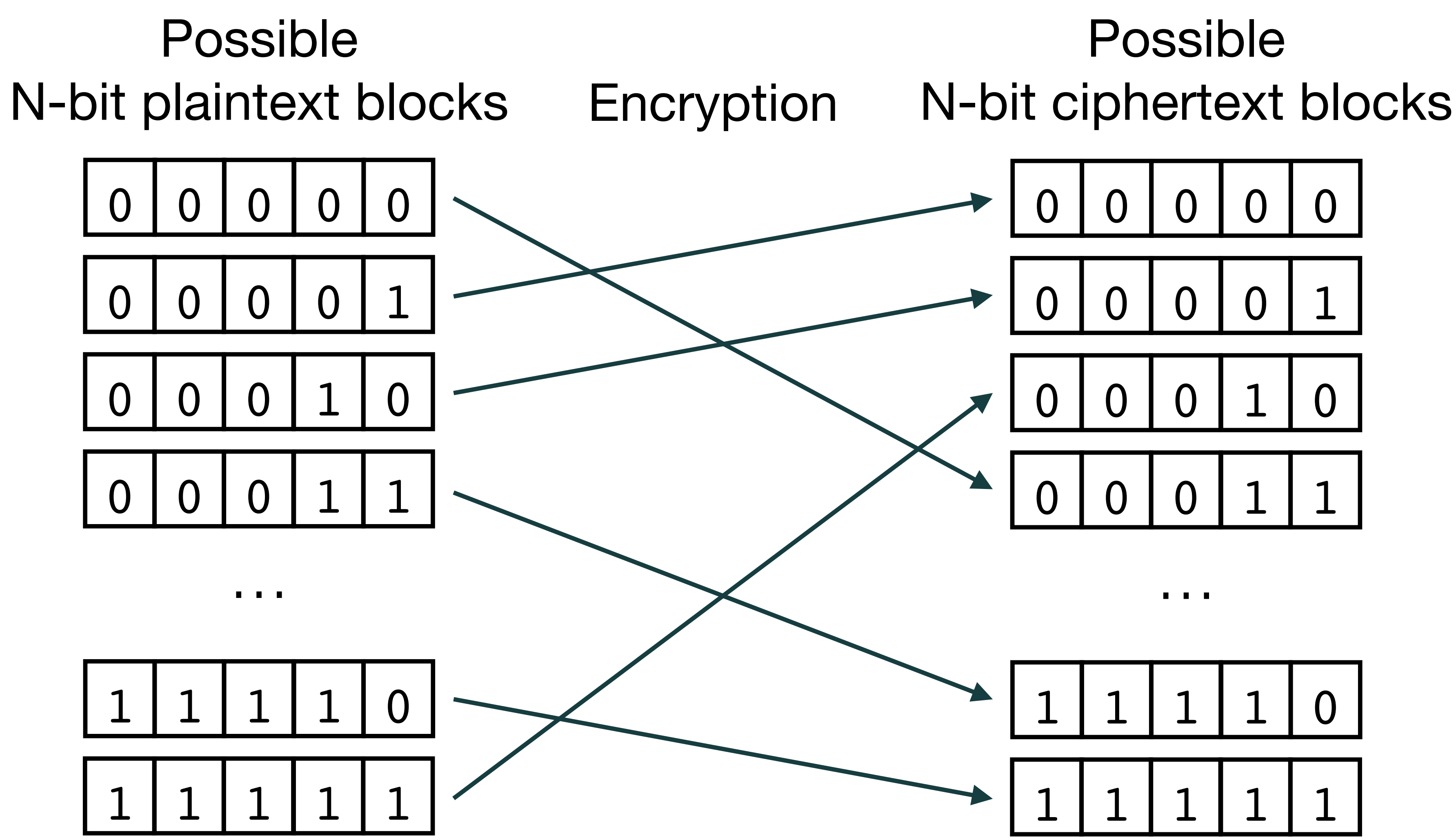
Block Cipher Design Considerations

- Key size
  - **number of possible keys** with K-bit key = 2^K (must prevent brute force attacks)
- Block size
  - **too short** → does not hide patterns in the plaintext
    - *example*: N = 8 bits (1 character in ASCII)
      → same as classic substitution cipher
  - **too long** → impractical, wasteful
- Encryption must be invertible
  - different input blocks must be transformed by the encryption to different output blocks
  - encryption can be viewed as a **permutation over all possible N-bit blocks**



Secure Block Cipher

- An N-bit block cipher can be viewed as a permutation over all possible N-bit blocks



mapping depends on the
cipher algorithm and **random secret key**

Secure Block Cipher

- An N-bit block cipher can be viewed as a permutation over all possible N-bit blocks
  - number of possible permutations with N-bit blocks = $2^N!$
- An N-bit block cipher is **secure** if it is **indistinguishable from a random permutation** of N-bit blocks (for a computationally bounded attacker)

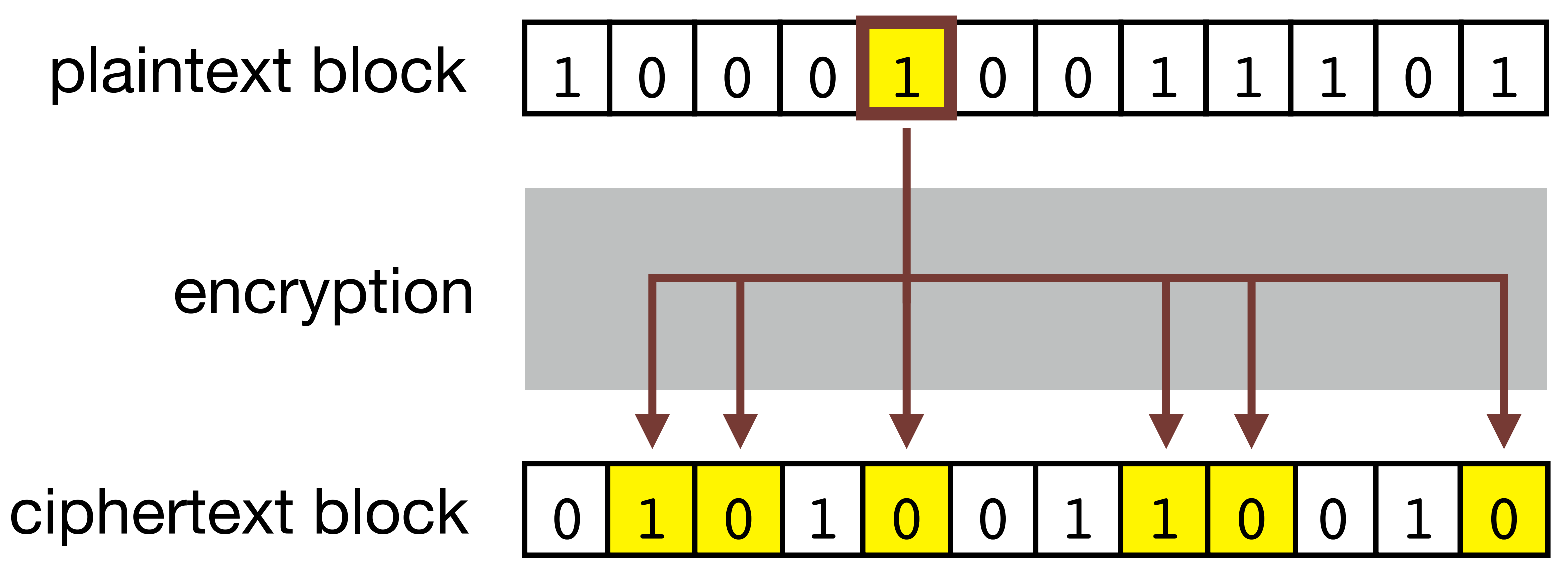
We need more practical goals to design a practical cipher…

Secure Block Ciphers in Practice

- Two design principles introduced by Claude Shannon in 1949

1. Diffusion

- *Goal*: dissipate the statistical structure of the plaintext over long-range statistics of the ciphertext
- **Each plaintext bit should affect the value of many ciphertext bits**



relationship between plaintext and ciphertext is "chaotic"

Secure Block Ciphers in Practice

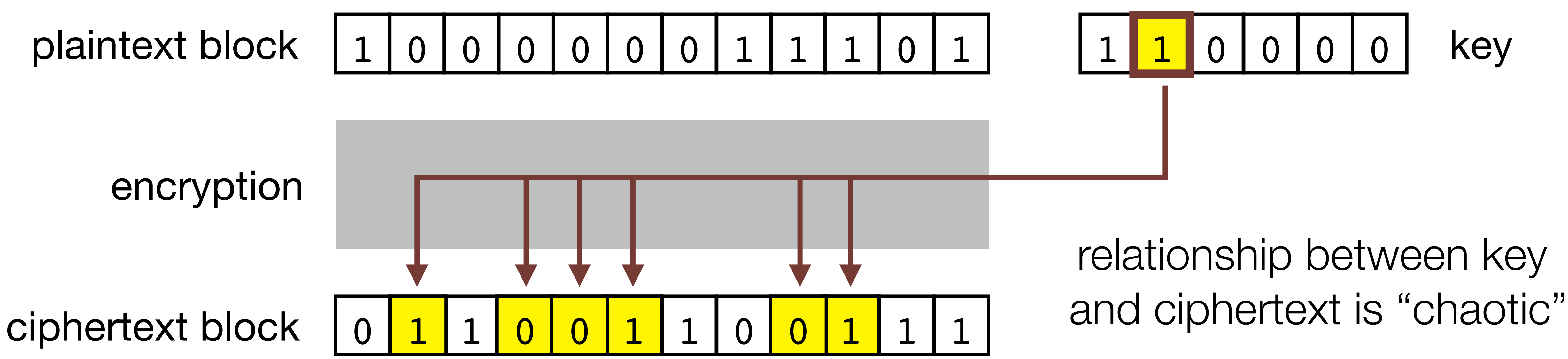
- Two design principles introduced by Claude Shannon in 1949

1. Diffusion

- *Goal*: dissipate the statistical structure of the plaintext over long-range statistics of the ciphertext
- Each plaintext bit should affect the value of many ciphertext bits

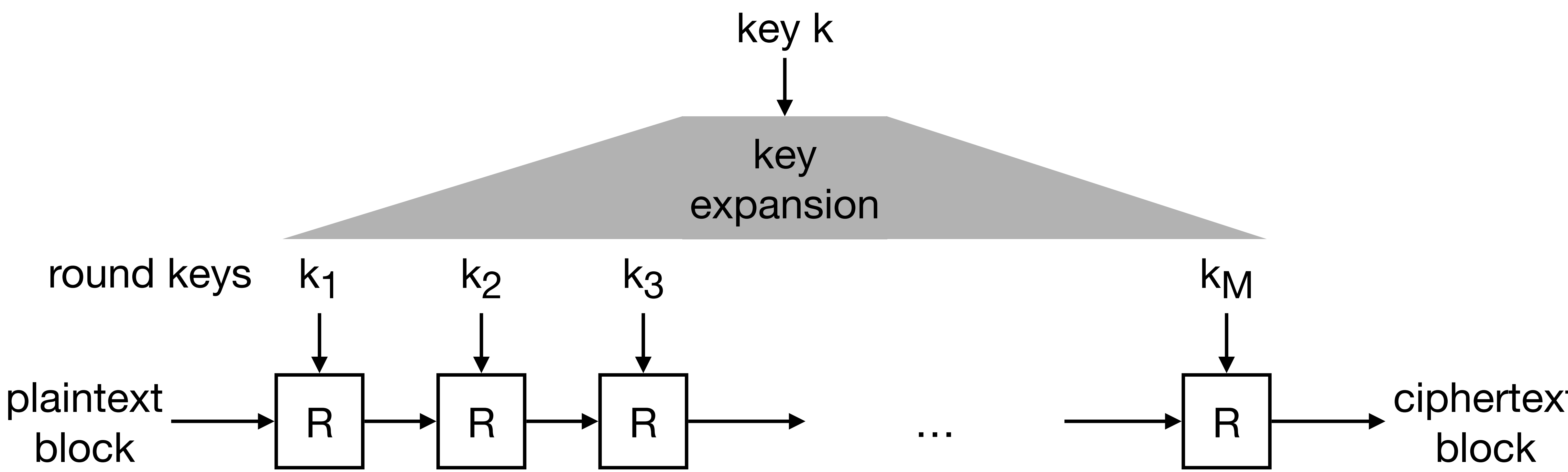
2. Confusion

- *Goal*: make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible
- **Each bit of the ciphertext should depend on many bits of the key**



Iterated Block Ciphers

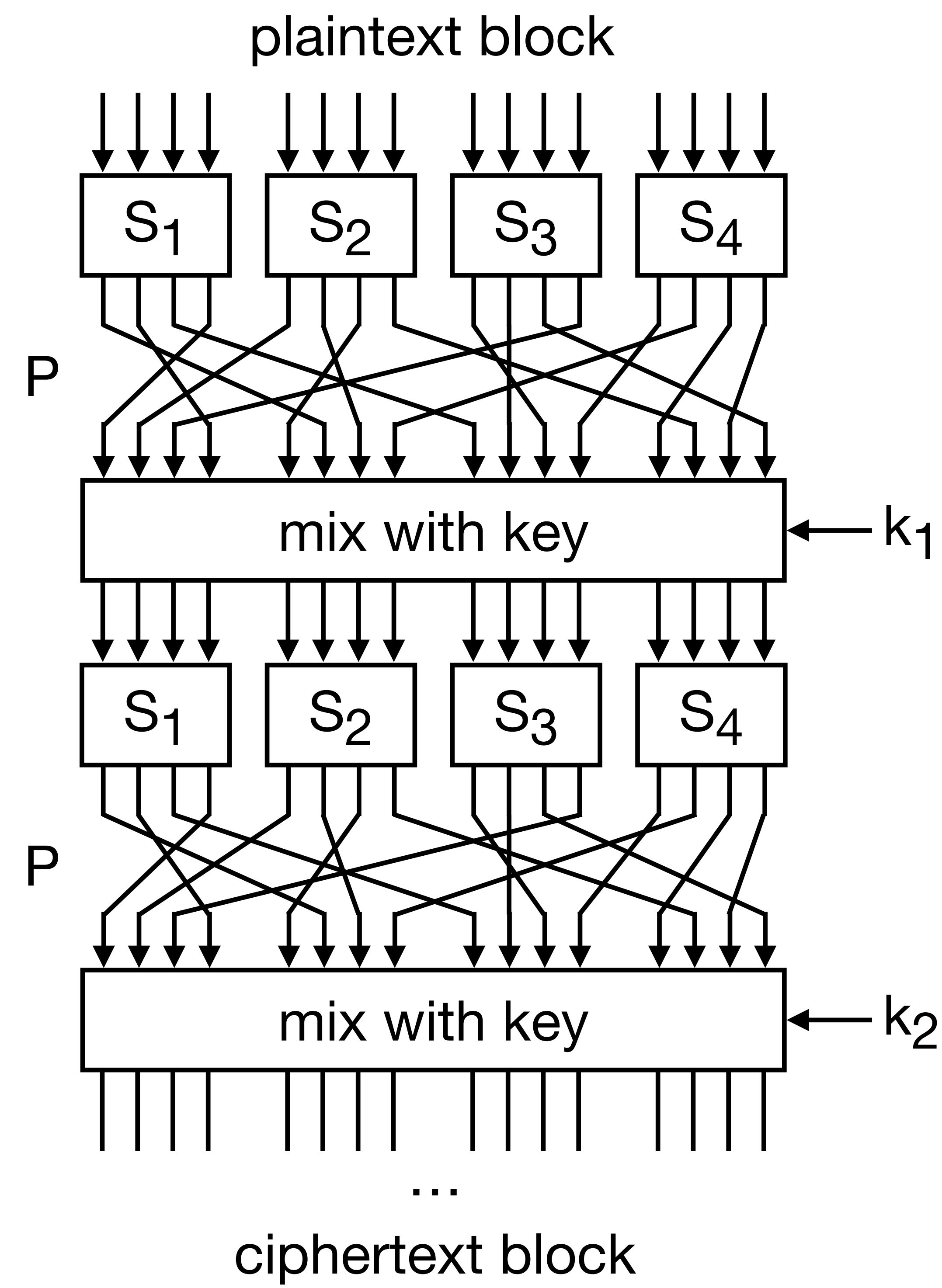
- It is difficult to design a single **invertible** transformation that satisfies both the **diffusion** and **confusion** properties



- R: round function
  - relatively "weak" transformation, which introduces some diffusion and confusion
  - by combining a large number of rounds, we can build a strong block cipher

Substitution-Permutation Ciphers

- A very common subtype of iterated block ciphers
- Each round R consists of two steps
  - **Substitution S**
    - substitutes a small block of bits with another small block
    - ideally, **changing one input bit changes half of the output bits**
  - **Permutation P**
    - permutation of all the bits

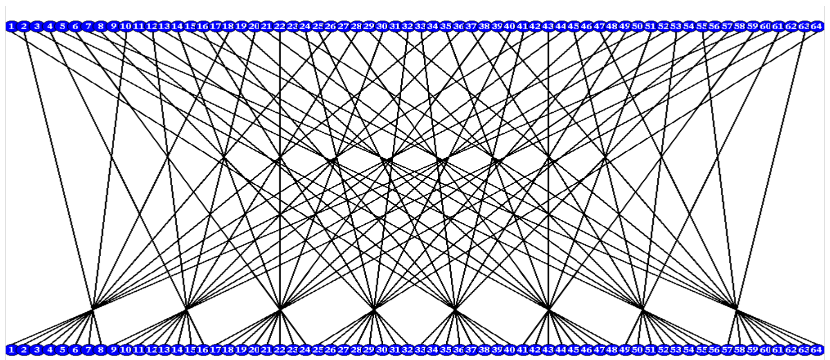


Data Encryption Standard (DES)

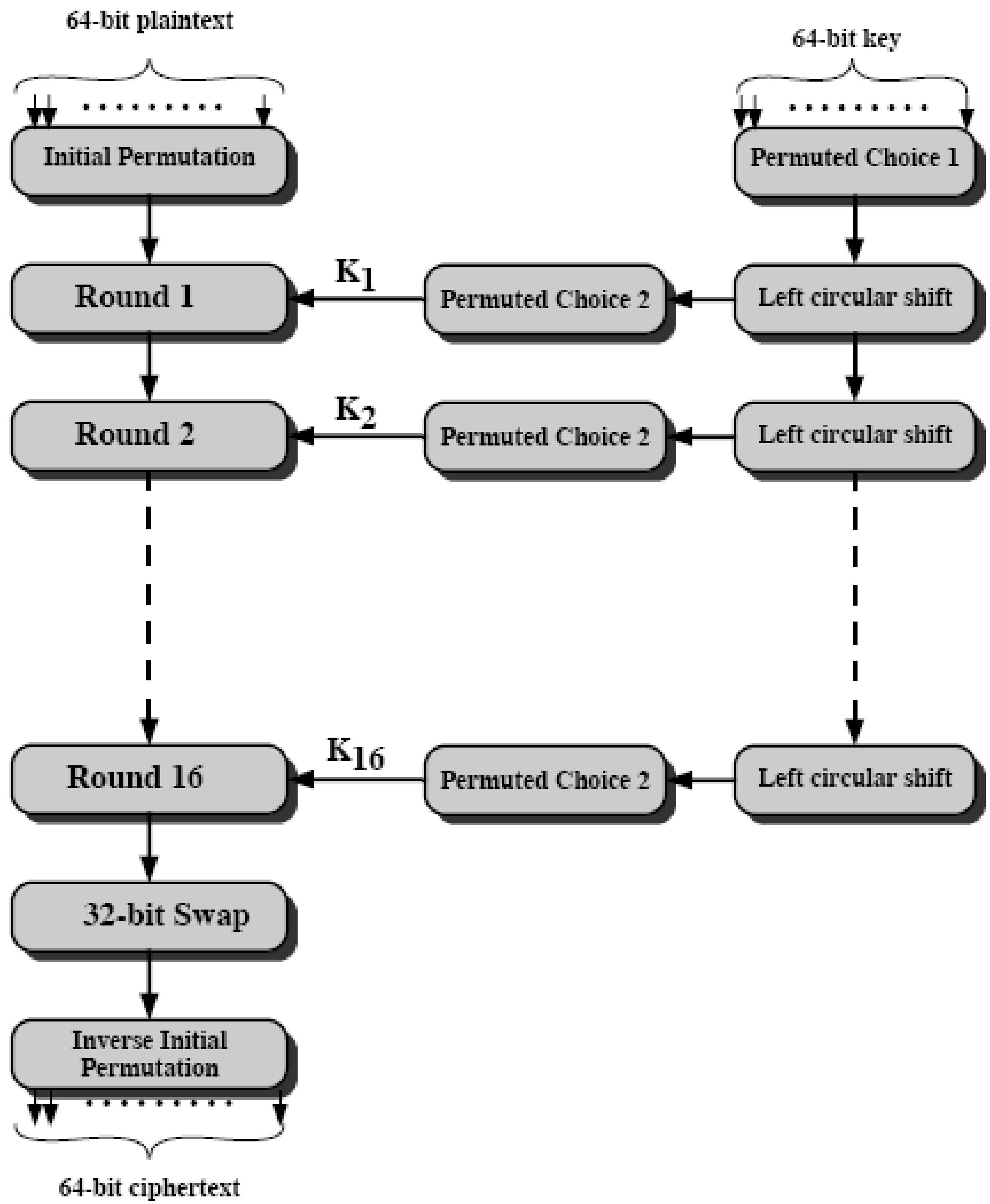
- In the early 1970s, Horst **Feistel** developed the Lucifer cipher at IBM with his colleagues
  - multiple variants with key and block sizes from 48 bits to 128 bits
- In 1973, the National Bureau of Standards (now named NIST) solicited proposals for a **government-wide standard encryption**
- In 1974, IBM submitted a cipher based on Lucifer
- In 1976, DES was approved as a **federal standard** by the NBS
  - block size: **64 bits**
  - key size: **56 bits**
  - iterated substitution-permutation cipher with 16 rounds

DES Structure

- Key
  - 56 bit random
  - 8 bit parity check
- Initial Permutation
  - no cryptographic significance
  - facilitated loading blocks in and out of 8-bit hardware



- Key permutation
  - discards the parity bits
  - no cryptographic significance



Feistel Network

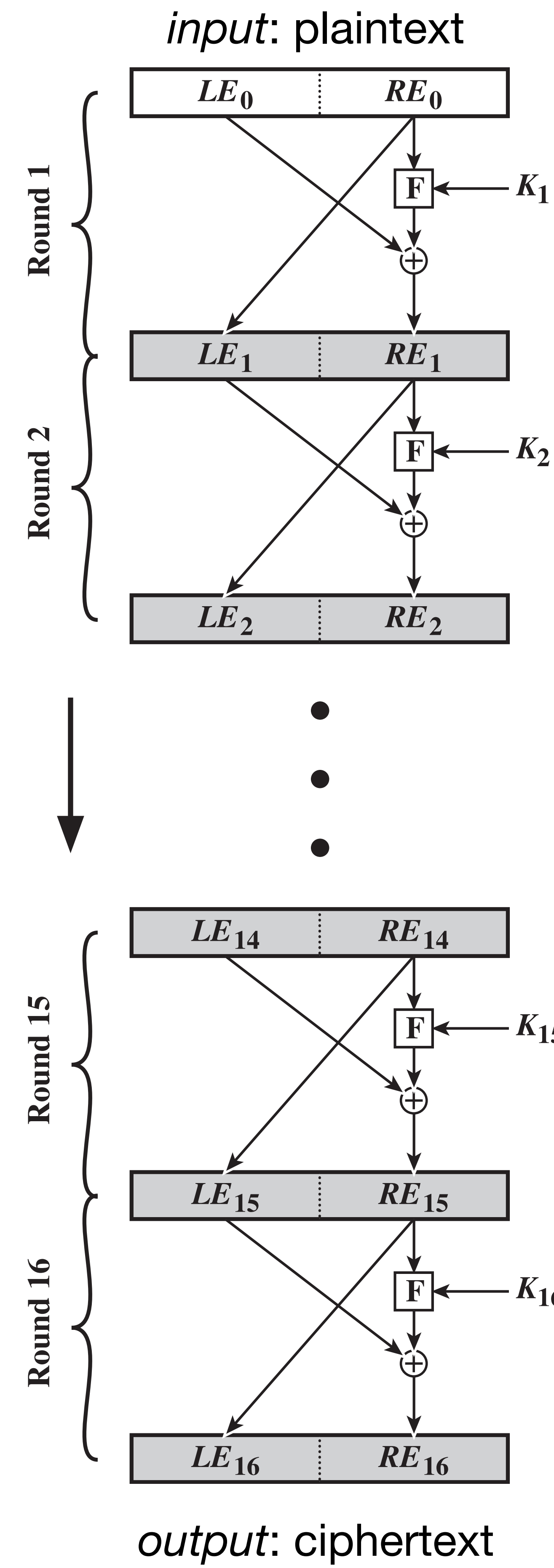
- Encryption round
  - *input*: block from previous round (or the plaintext)
  - divide input into two halves $\mathbf{L_i}$ and $\mathbf{R_i}$
  - derive round key $\mathbf{K_i}$ from the secret key (different for each round)
  - *output*:

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus F(K_i, R_i)$$

- Decryption round
  - we can invert the encryption without inverting **F**:

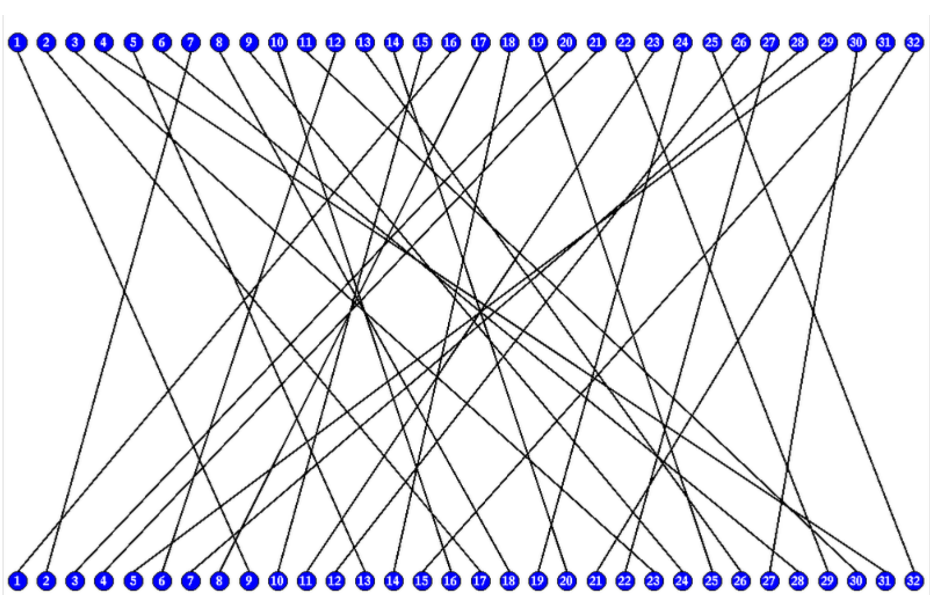
$$R_i = L_{i+1}$$
$$L_i = R_{i+1} \oplus F(K_i, L_{i+1})$$
$$= R_{i+1} \oplus F(K_i, R_i) = L_i \oplus F(K_i, R_i) \oplus F(K_i, R_i) = L_i$$

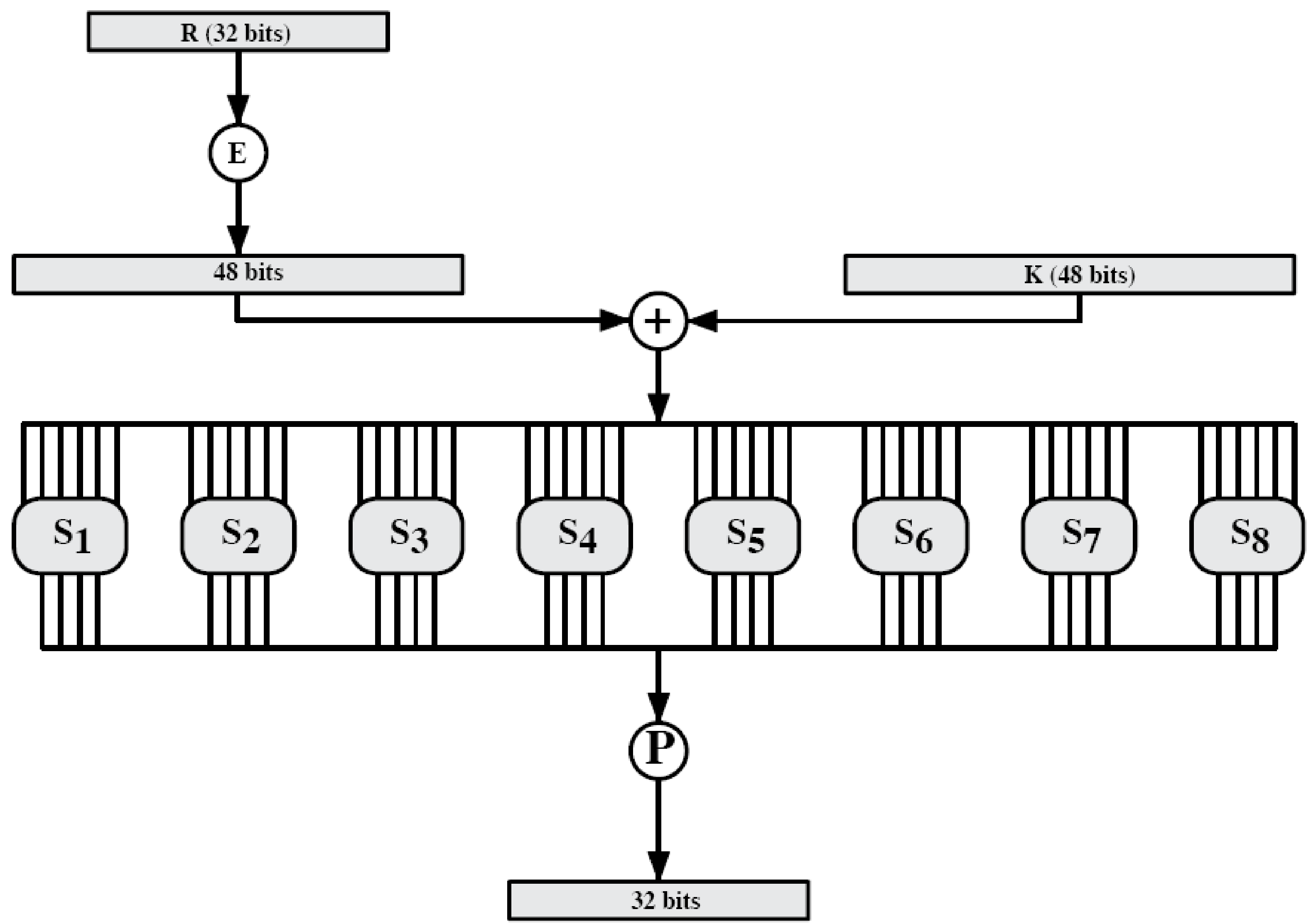
use the same implementation with round keys in reverse order



DES F-Function

- **Expansion**:
  duplicates half of the bits
- **Substitution** (S-boxes):
  maps 6 bit block into 4 bit block based a lookup table
- **Permutation** (P-box):
  fixed permutation





DES S-Boxes

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

- Each S-box S_i is different
  - tables are specified by the standard
- S-boxes (and P-box) were be carefully designed
  - randomly chosen boxes would result in an insecure cipher

Security of DES

- Cryptanalysis
  - **best known attack:** linear cryptanalysis, which requires 2^{43} known plaintexts and ciphertexts, and finds a key in $\mathbf{2^{39}}$ steps
- Vulnerable to brute-force attacks: **key length = 56 bits**
  - *in **1977**, Diffie and Hellman proposed a parallel machine with 1 million encryption devices (**~$20 million**), which would have found a DES key in **10 hours***
  - *in 1997, RSA Security sponsored a contest for breaking DES: DESCHALL Project utilized thousands of Internet-connected computers run by volunteers to find DES key in 3 months*
  - *in **1998**, the Electronic Frontier Foundation built a machine for less than **$250,000**, which found a DES key in **56 hours***
  - *in **2008**, SciEngines designed RIVYERA, which can find a DES key in **less than a day** and costs around **$10,000***
- Since 1999, DES is permitted by NIST only in legacy systems

Advanced Encryption Standard

Advanced Encryption Standard (AES)

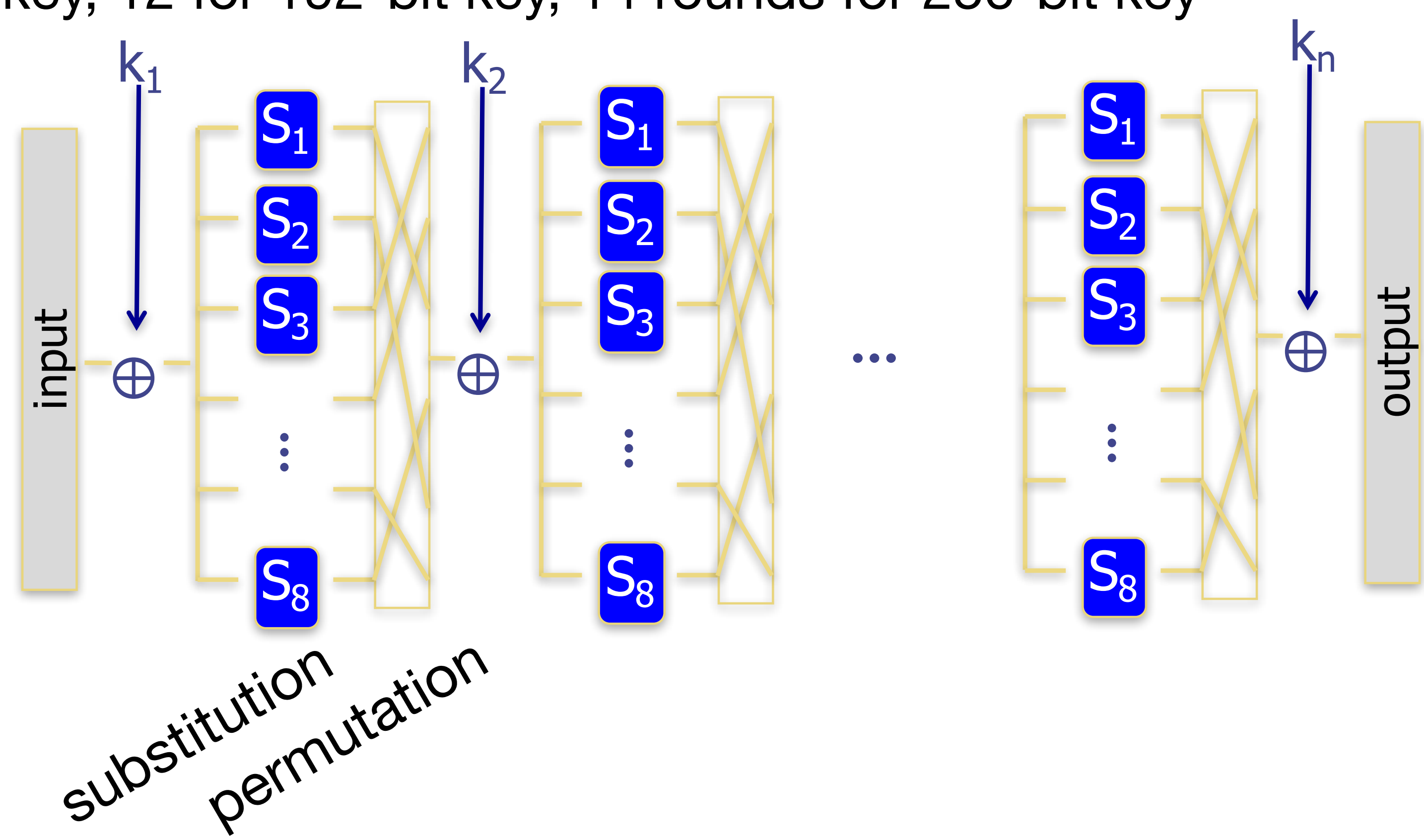
- In 1997, NIST announced a **request for proposal to replace DES**
- Based on initial feedback, NIST announced a call for ciphers
  - requirements: **128-bit block size**, and **128, 192, 256-bit key size**
- 15 submissions were received in 9 months
  - ciphers were evaluated based on both their strength against cryptanalytic attacks as well as performance
- In 1999, the list was narrowed down to five “AES finalists”
- In 2000, NIST announced the winning cipher: Rijndael
  - developed by Belgian cryptographers Joan Daemen and Vincent Rijmen
- Standard: **FIPS PUB 197: Advanced Encryption Standard** (2001)

AES Applications

- WiFi security
  - **WPA2 / WPA3**: current standards
- Web security (HTTPS)
  - **SSL/TLS**: supported since 2008, one of the most widely used ciphers today
- Other protocols
  - IPSec, SSH
- Disk encryption
  - FileVault (Mac OS X), BitLocker (Windows)
- Compressed archives
  - 7z, WinZIP
- …

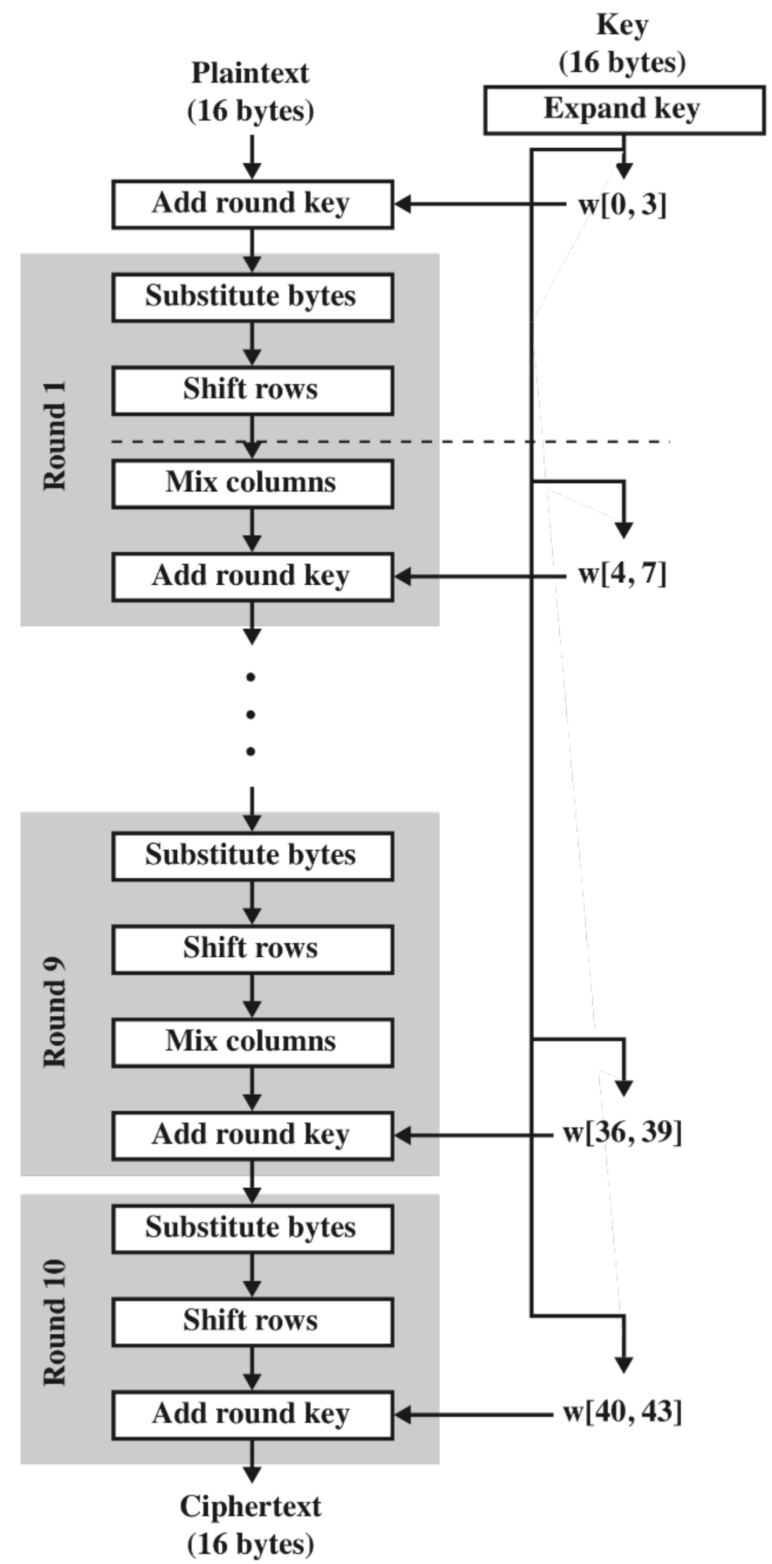
AES Structure

- Substitution-permutation cipher
  - but **not** a Feistel network
- Each round must be invertible for decryption
- Key expansion and schedule: generates a different “round key” for each round
- Number of rounds depends on the key size
  - 10 for 128-bit key, 12 for 192-bit key, 14 rounds for 256-bit key



AES Round

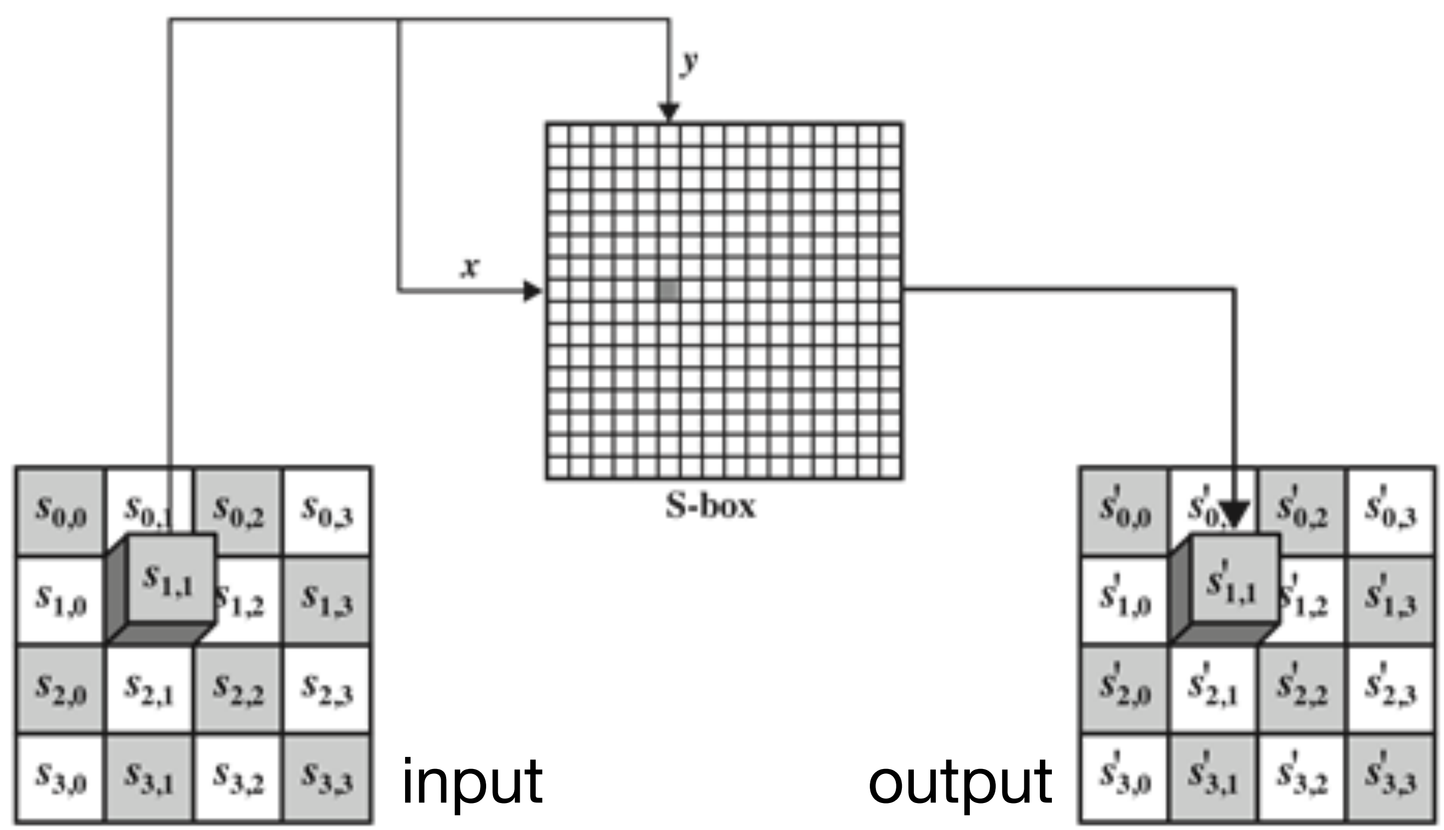
- Input:
  - 128-bit “state” from previous round (or the plaintext) represented as a 4 x 4 byte matrix
  - 128-bit round key (from key schedule)
- Output: 128-bit state
- Each round consists of multiple steps:
  - `AddRoundKey`: XOR round key to the state
  - substitution and permutation:
    - `SubBytes`
    - `ShiftRows`
    - `MixColumns`



`SubBytes` Step

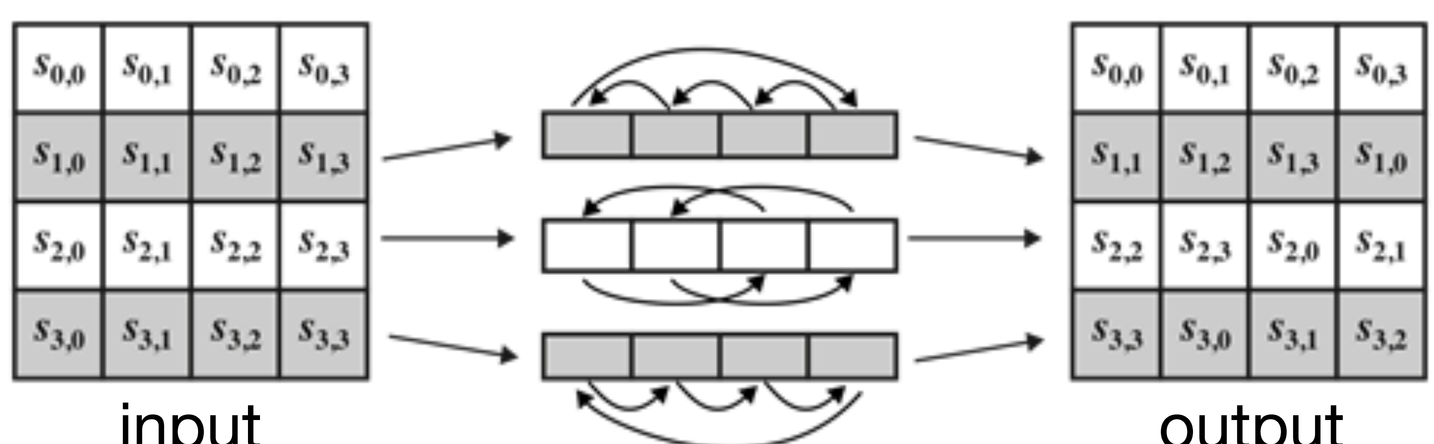
- Each byte is replaced using an **8-bit substitution box** (S-box)
  - defined using **mathematical operations:** multiplicative inverse over a finite field + affine transformation
- Designed to be resistant to cryptanalysis
  - minimize correlation to linear functions
  - minimize difference propagation

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16



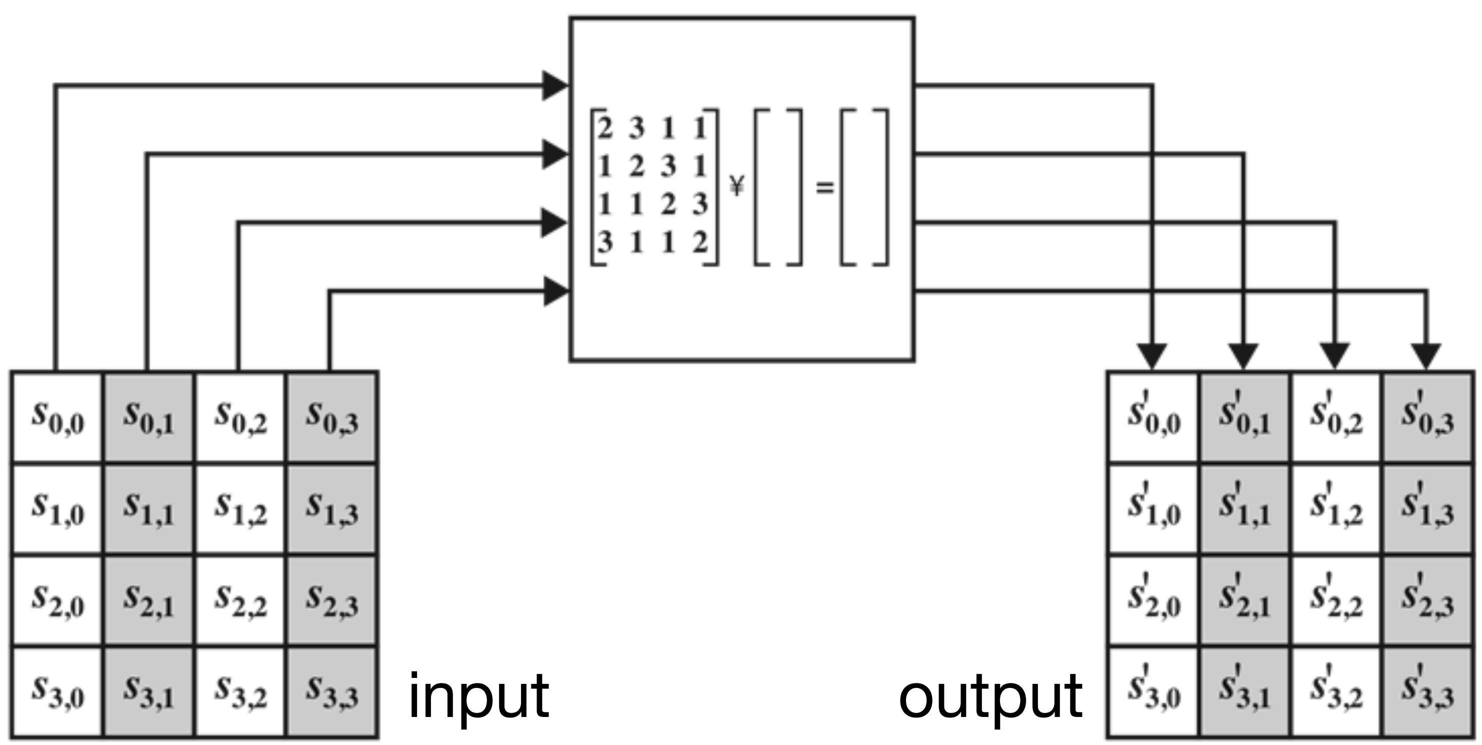
`ShiftRows` Step

- **Cyclically shifts** the second, third, and fourth rows to the left
  - second row is shifted one byte
  - third row is shifted two bytes
  - forth row is shifted three bytes
- Ensures that the 4 bytes of each column are spread out to four different columns → provides diffusion
  - without this step, each input byte would affect only a single column



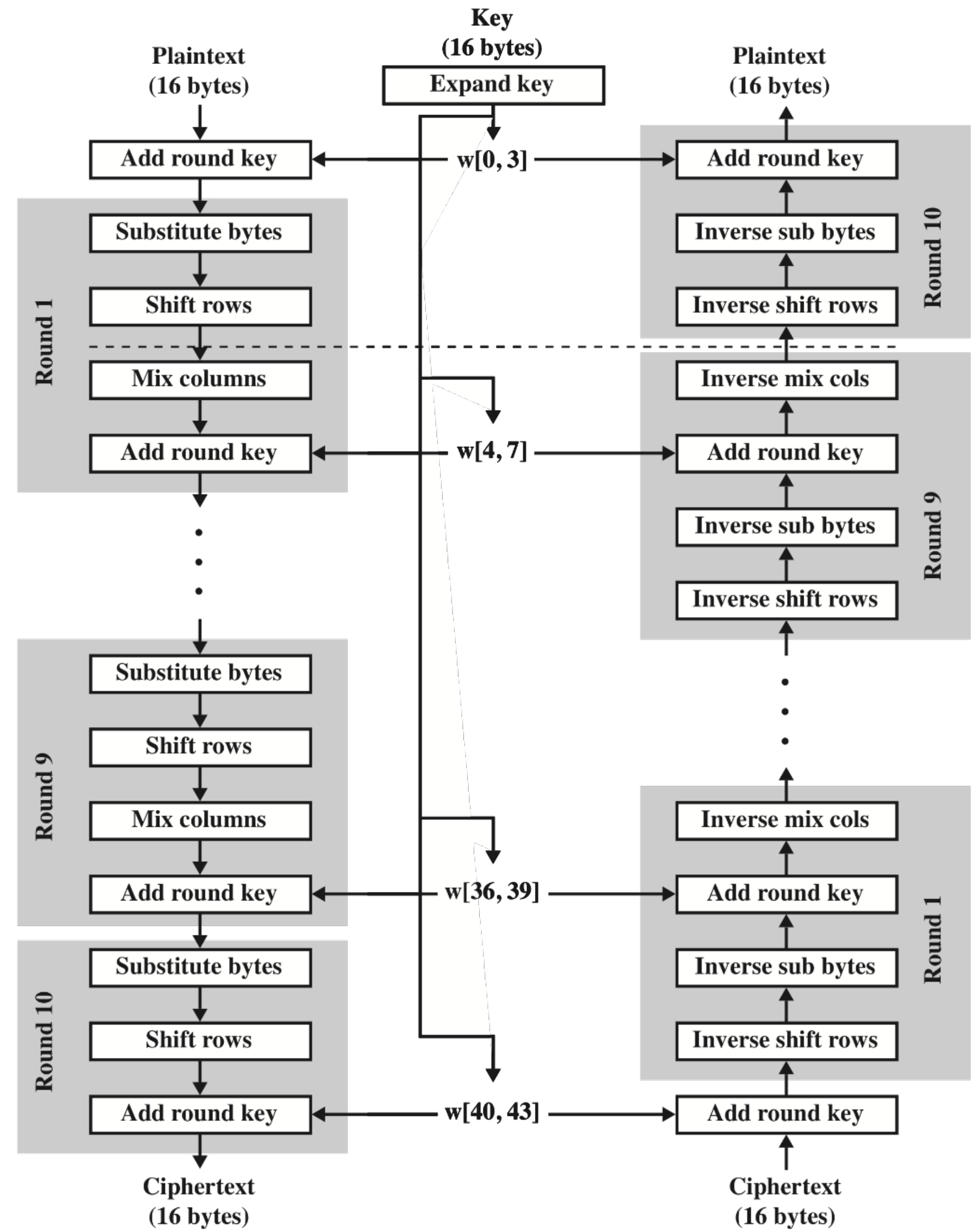
`MixColumns` Step

- Each column is multiplied by a fixed matrix
  - invertible linear transformation
- Good mixing among the bytes of each column → provides diffusion
  - combined with `ShiftRows`, ensures that each output bit depends on every input bit after a few rounds



AES Decryption

- Each step is invertible
  - `InvertMixColumns`: multiply by matrix inverse
  - `InvertShiftRows`: shift rows cyclically to the right
  - `InvertSubBytes`: invert affine transformation and multiplicative inverse
  - `InvertAddRoundKey`: XOR round key to state
- For decryption, round keys are used in reverse order



AES Performance and Security

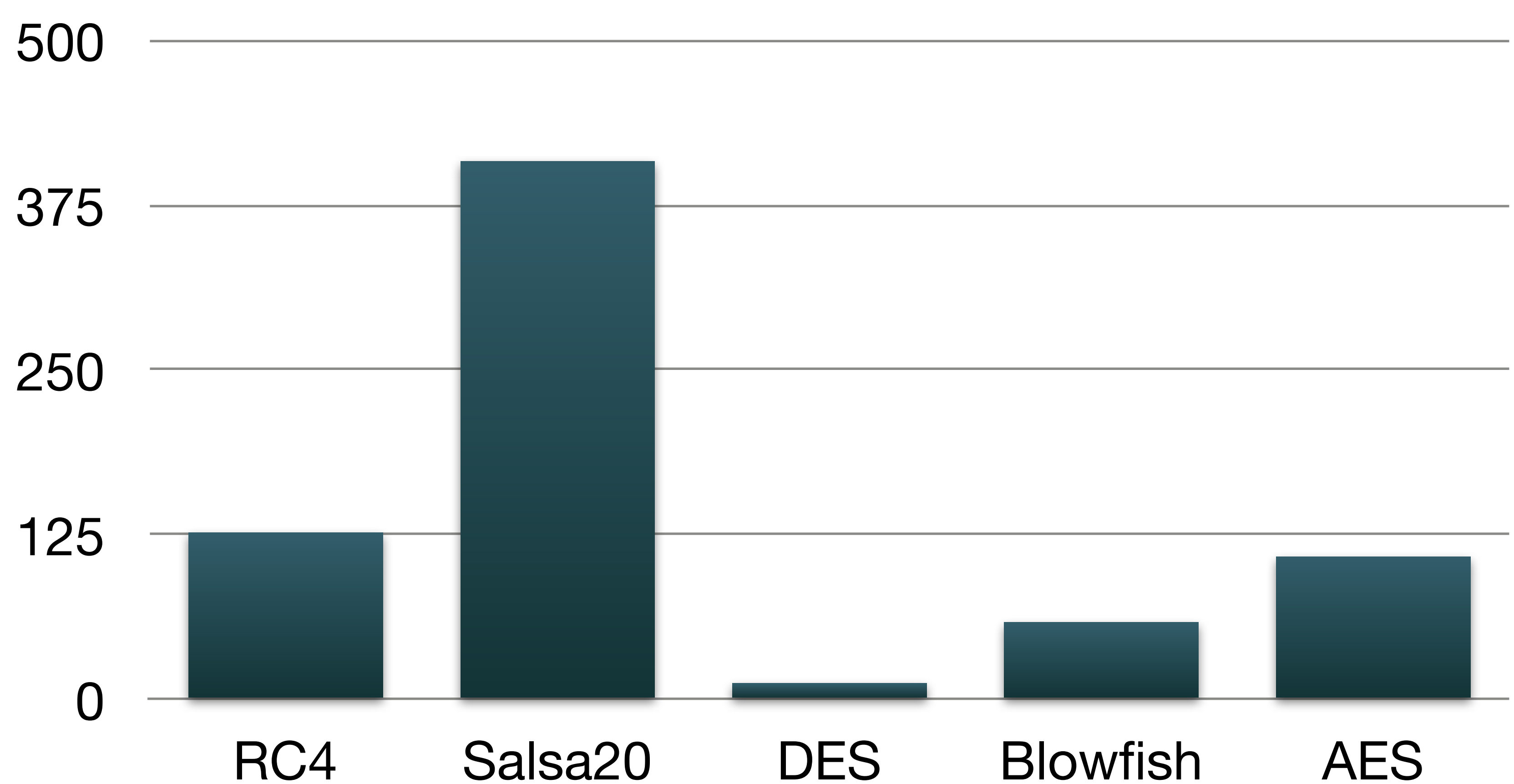
- Operations on bytes and 32-bit words
  - most operations can be precomputed (*e.g.*, 256-byte substitution table for `SubBytes`)
- Hardware support: AES instruction set for CPUs
  - introduced for x86 by Intel in 2008, supported by newer Intel and AMD CPUs
  - other architectures also provide support (*e.g.*, ARM, IBM Power, SPARC)
  - instructions for computing a round of encryption/decryption, key generation, etc.
  - supported by many software (*e.g.*, Java, Linux cryptography API, OpenSSL)
- Best attack against arbitrary keys
  - in 2015, it was shown that 128-bit AES keys can be recovered in 2^{126} steps (only four times faster than brute-force search over the entire key space)
- There are no publicly known practical attacks

Other Notable Block Ciphers

- KASUMI
  - block cipher used UMTS (3G) cell phone networks (also in GSM as A5/3)
  - derived from MISTY1, a block cipher developed by Mitsubishi Electric in 1995
  - 64-bit blocks, 128-bit key, based on Feistel structure with 8 rounds
  - in 2010, a very efficient related-key attack was published; however, it is not applicable to how KASUMI is used in 3G networks
- Blowfish
  - designed in 1993 by Bruce Schneier
  - 64-bit blocks, 32 - 448-bit key, based on Feistel structure with 16 rounds (similar to DES)
  - small block size may be exploited if large amount of data is encrypted
- Twofish
  - based on Blowfish
  - one of the “AES finalists”, no practical attacks are known
- Serpent
  - substitution-permutation cipher with 32 rounds, operating on 32-bit words
  - one of the “AES finalists”, no practical attacks are known

Block Ciphers vs. Stream Ciphers

- Stream ciphers
  - can encrypt one bit at a time
  - are typically faster and use less memory
- Block ciphers
  - can be used to build various other cryptographic primitives
  - leak less information with key reuse



Speed of ciphers [MB / second] measured on an Intel Core 2 1.83 GHz using the Crypto++ 5.6 library

Next lecture:

Block Cipher Modes of Operation