Block ciphers

Elena Kirshanova

Course “Information and Network Security”
Lecture 3
10 марта 2020 г.

Recap: PRG

- 'Random' in crypto may come from two sources:
  - A 'true' random number generator (or entropy generator)
  - An algorithmic 'pseudorandom number generator' (PRNG)

- **Pseudorandom generator (PRG)** – efficient algorithm taking on input truly random bits (seem) and outputting bits that are indistinguishable from random by ppt adversaries

- Examples: in Linux: /dev/random, /dev/urandom (not recommended for crypto applications); both use the same PRG (SHA-1); on Windows: CryptoAPI's CryptGenRandom

- Better use established cryptographic PRGs, e.g. ChaCha, Salsa, HMAC-SHA1 or CBC-AES

- Be aware of backdoored PRGs: Dual EC DRBG

Recap: PRG

- Statistical Tests: Diehard, NIST's SP 800-22
- Known Answer Tests : BETTER NOT IN CODE RELEASE

Recap: Stream ciphers

- Symmetric key is a seed to a PRG
- Encrypt: $\mathsf{Enc}(m, s) = \mathsf{PRG}(s) \oplus m = c$
  Decrypt: $\mathsf{Dec}(c, s) = \mathsf{PRG}(s) \oplus c$
- "Primitive"(= OTP) stream ciphers are typically very fast and simple
- ... but inappropriate (= insecure) for many scenarios: broken by key re-use, require integrity checking
- Examples of practical stream ciphers: RC4, Trivium, A5/1 Generator

Formal definition

A **Block cipher** is a deterministic cipher with $\mathcal{X} := \mathcal{M} = \mathcal{C}$ and a function

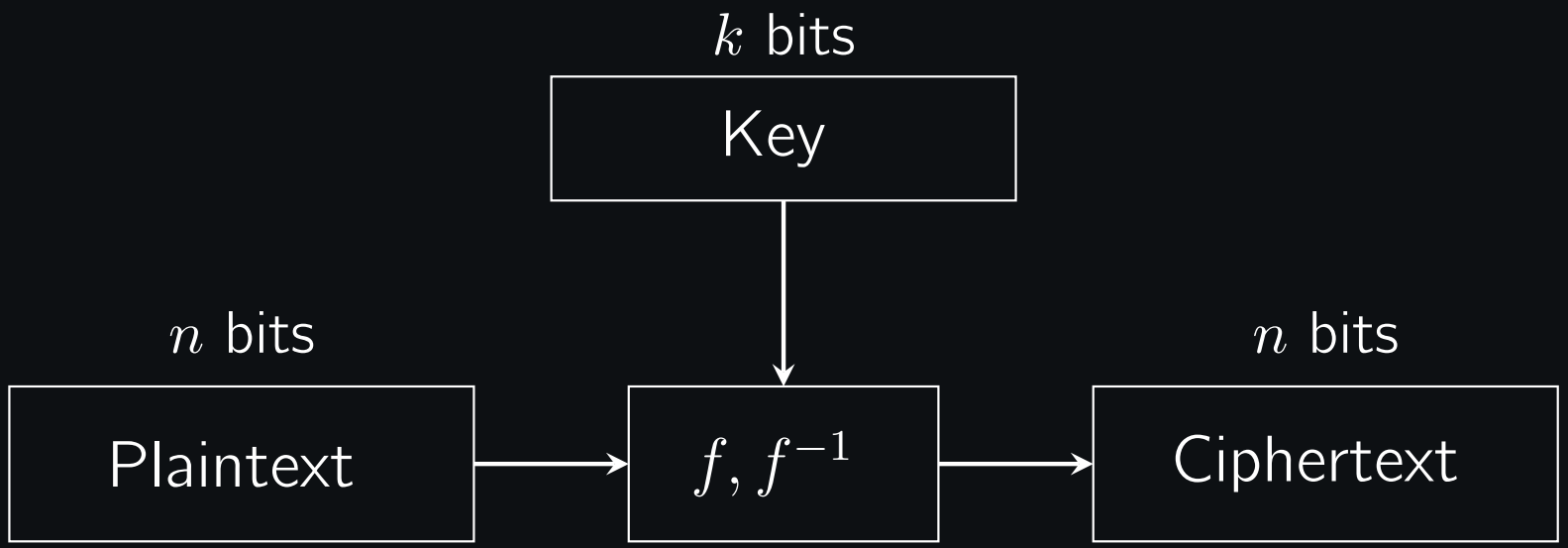
$$f(k, \cdot) : \mathcal{X} \to \mathcal{X}$$

- Correctness $\implies$ $f(k, \cdot)$ is one-to-one for all k.
- $|\mathcal{X}| < \infty$.

Thus, $f(k, \cdot)$ is a permutation on $\mathcal{X}$.

Security (informal) : $f(k, \cdot)$ "looks like" a random permutation

Block cipher in pictures



Examples:

- AES: $n = 128$, $k = 128, 192, 256$
- ГОСТ 34.12-2018: $n = 128$, $k = 256$ (Кузнечик)

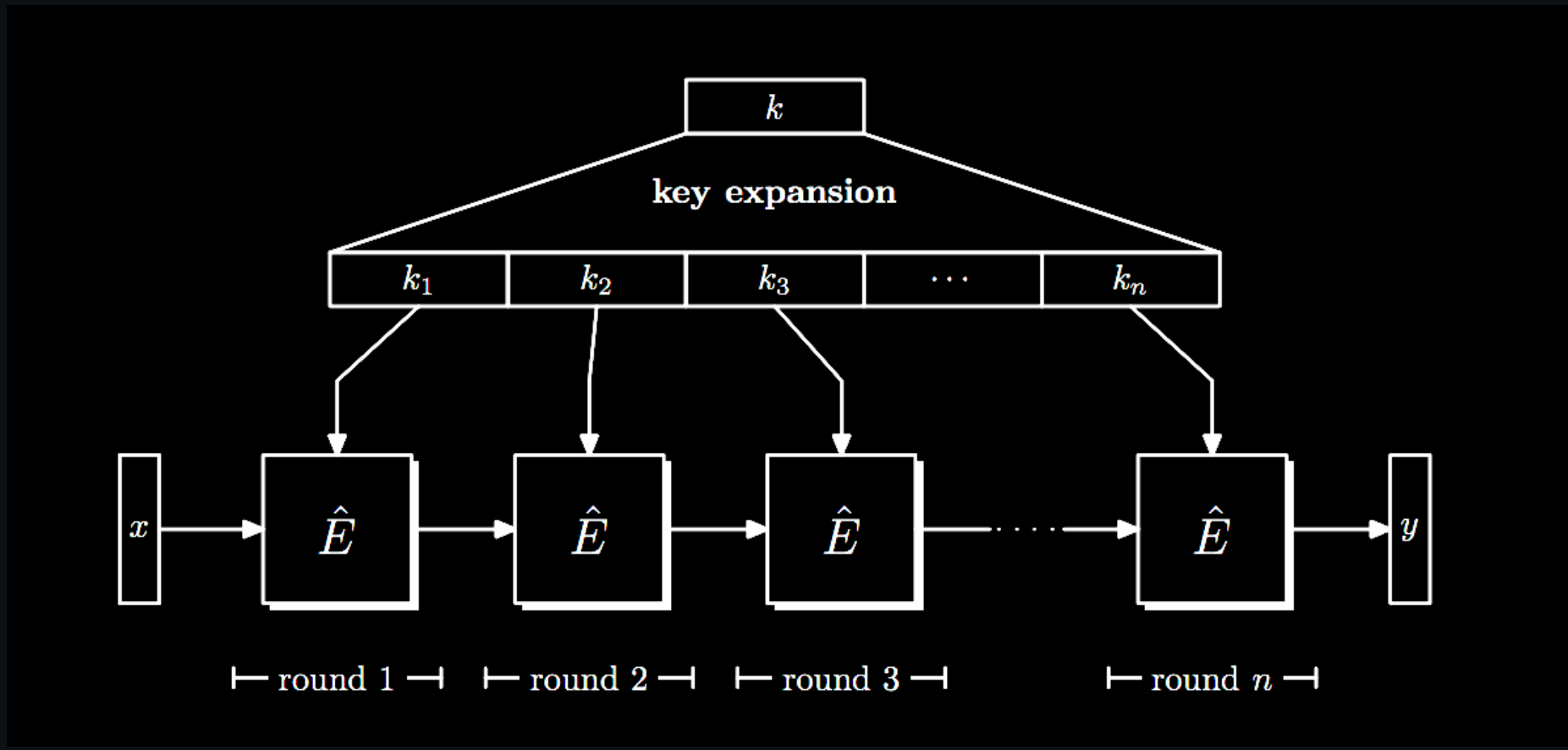
A bit of history

- 70's: IBM designs Lucifer. $k = 128, n = 128$
- '76: DES is standardised $k = 56, n = 64$
- '98: 3DES is standardised $k = 168, n = 64$
- '00: AES winner Rejndael $k = \{128, 192, 256\}, n = 128$

Russian standards:

- '89: ГОСТ 28147-89 $k = 256, n = 64$
- '15 : ГОСТ Р 34.12-2015, RFC 7801 $k = 256, n = 128$

Block ciphers are iterative



x – plaintext, y – ciphertext

picture is taken from D.Boneh, V.Shoup A Graduate Course in Applied Cryptography

Two main paradigms in block cipher designs

- Feistel cipher
  Сеть Фейстеля

  Examples: DES, ГОСТ 28147-89

- Substitution-Permutation Network (SPN).
  Подстановочно-перестановочная сеть

  Examples: AES, ГОСТ 34.12-2018

Feistel Cipher

Provides a generic way to build invertible functions from arbitrary functions.

Given $f_1, \ldots, f_r : \{0,1\}^n \to \{0,1\}^n$

construct an invertible $F : \{0,1\}^{2n} \to \{0,1\}^{2n}$

Feistel Cipher

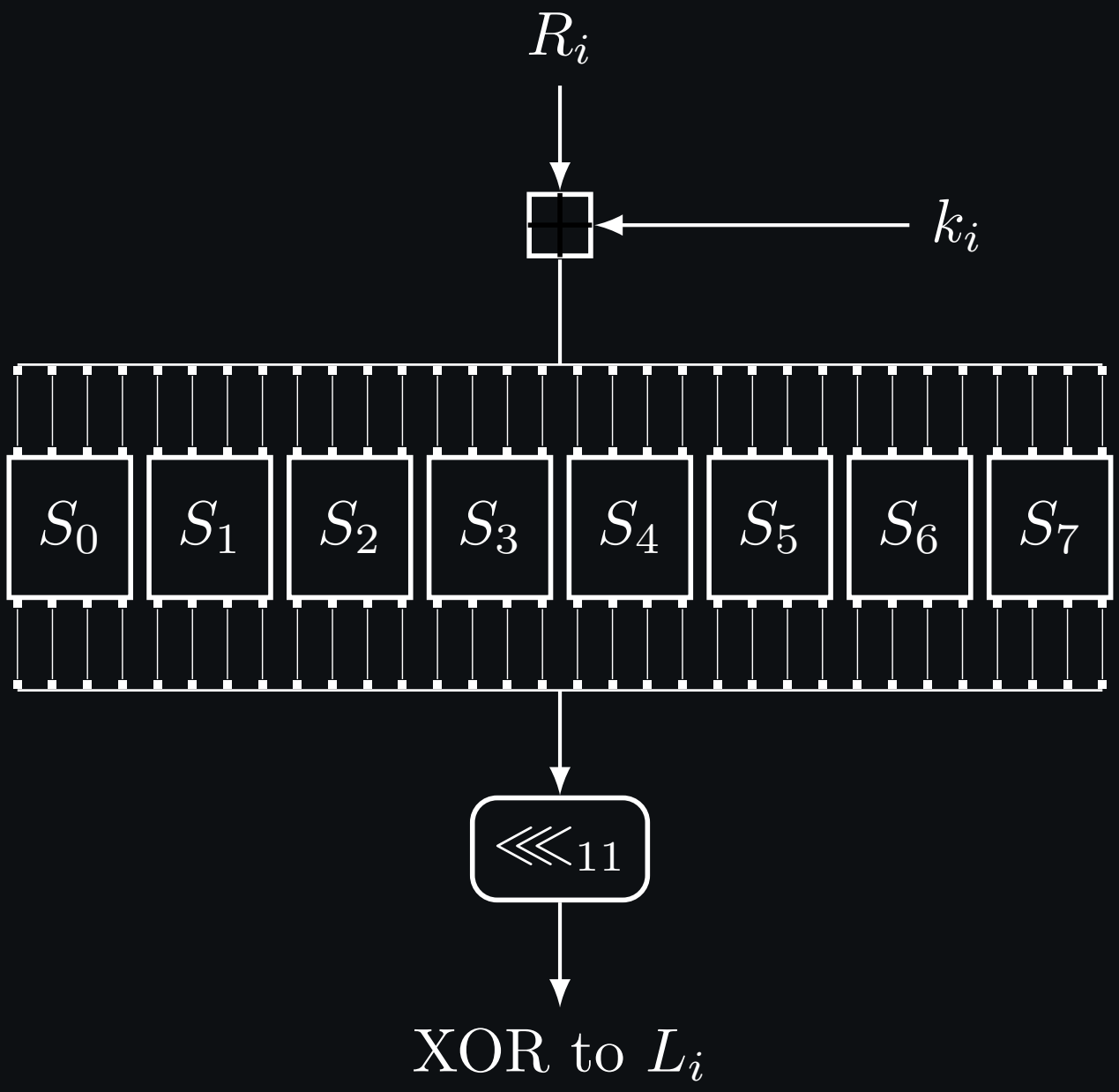
Provides a generic way to build invertible functions from arbitrary functions.

Given $f_1, \ldots, f_r : \{0,1\}^n \to \{0,1\}^n$

construct an invertible $F : \{0,1\}^{2n} \to \{0,1\}^{2n}$

Security (informal) : if $f : \mathcal{K} \times \{0,1\}^n \to \{0,1\}$ "looks like" a random function, then 3-Round Feistel
$F : \mathcal{K}^3 \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ is a pseudorandom permutation.

Example: GOST'89 Round function



What's S-box?

$$S := \{0,1\}^n \rightarrow \{0,1\}^m$$

- Implemented as a look-up table
- There can be several S-boxes in one block-cipher
- Designed to be resistant to linear and differential cryptanalysis
- Must not contain any fixed points:
  $S(x) \neq x$, $S(x) \neq \bar{x}\ \forall x$
- an S-box is **perfect** if it's a *bent* function (i.e., as "far way" from linear of affine boolean function as possible)

Example: S-box in DES

$$S := \{0,1\}^6 \to \{0,1\}^4$$

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

picture taken from Wikipedia

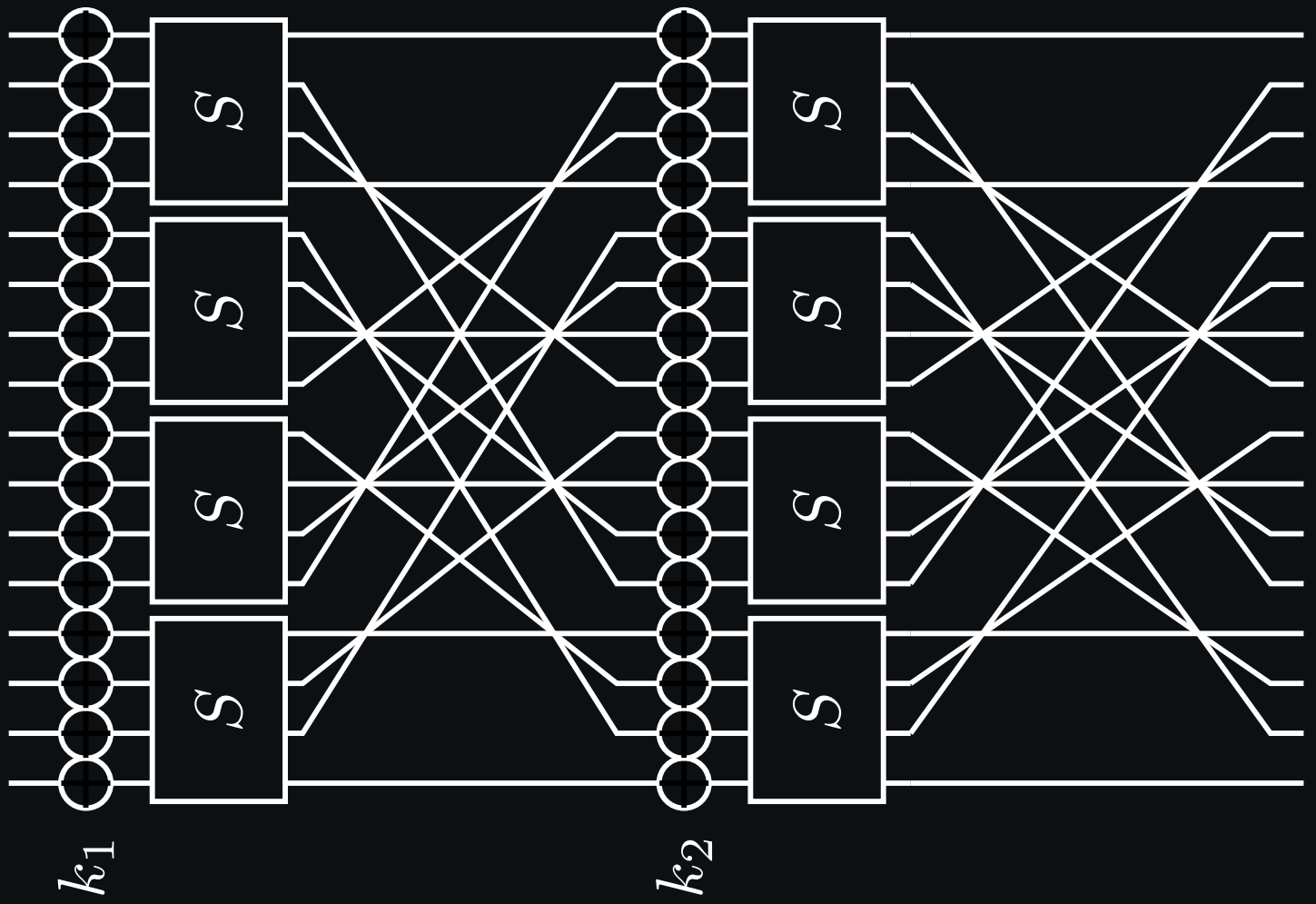
Example: S-box in AES

$$S := \{0,1\}^8 \to \{0,1\}^8$$

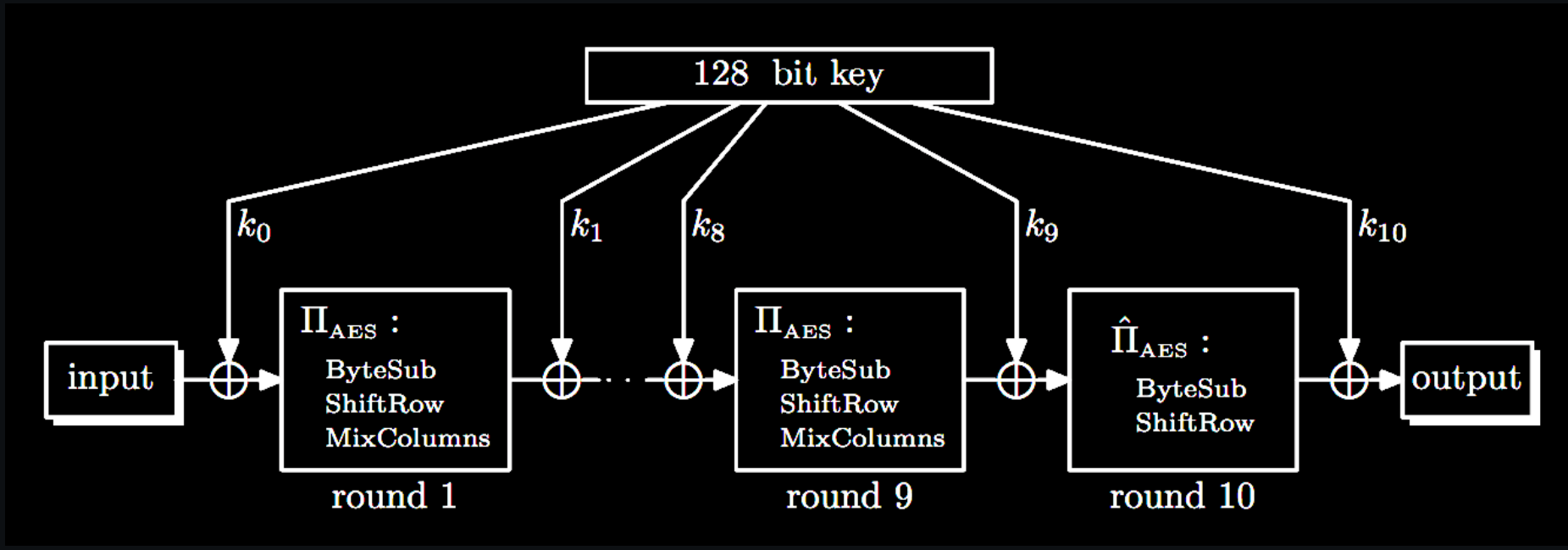
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
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

picture taken from Wikipedia

Substitution-Permutation Network (SPN)



AES: an SPN cipher



$$\Pi_{\text{AES}} = \{0,1\}^{128} \to \{0,1\}^{128} - \text{ invertible permutation}$$

picture is taken from D.Boneh, V.Shoup A Graduate Course in Applied Cryptography

Attacks on block ciphers

Exhaustive search for block cipher key.

For DES/AES/GOST: **two** plaintext/ciphertext pairs
$(m_1, c_1 = \mathsf{Enc}(k, m_1)), (m_2, c_2 = \mathsf{Enc}(k, m_2))$
determine k with sufficiently high probability

Example : For DES find $k \in \{0,1\}^{56}$ s.t. $c_i = \mathsf{Enc}(m_i, k)$.

Cryptanalytic efforts:

- **In '99** 22h on DeepCrack + distributed.net (a bit expensive hardware)
- **In '07** 13 days COPACOBANA (cheaper)

Advanced attacks on block ciphers

- Design attacks: linear & differential cryptnalalysis
  Target: find a linear relation in bit positions

$$\Pr\left[m[S_0] \oplus \mathsf{Enc}(k,m)[S_1] = k[S_2]\right] \geq 1/2 + \varepsilon$$
$$S_i \subset \{0,\ldots,n-1\} \quad \forall k \text{ and random } m$$

Advanced attacks on block ciphers

- Design attacks: linear & differential cryptnalalysis
  Target: find a linear relation in bit positions

$$\Pr\left[m[S_0] \oplus \mathsf{Enc}(k,m)[S_1] = k[S_2]\right] \geq 1/2 + \varepsilon$$
$$S_i \subset \{0,\ldots,n-1\} \quad \forall k \text{ and random } m$$

- Side-channel attacks: measure **time** or **power** needed for Enc, Dec

Advanced attacks on block ciphers

- Design attacks: linear & differential cryptnalalysis
  Target: find a linear relation in bit positions

$$\Pr\left[m[S_0] \oplus \mathsf{Enc}(k,m)[S_1] = k[S_2]\right] \geq 1/2 + \varepsilon$$
$$S_i \subset \{0,\ldots,n-1\} \quad \forall k \text{ and random } m$$

- Side-channel attacks: measure **time** or **power** needed for Enc, Dec
- Fault-injection attacks: cause the hardware to introduce errors at runtime (heat, EM interference)

Take-home message

- **DON'T** design **YOUR OWN** block-cipher
- **TRY NOT TO** implement cryptoprimitives yourself if good implementations exist
- Choose key-sizes wisely

Further reading