A Rough Guide to Modern Cryptology

ralf-philipp weinmann
<ralf-philipp.weinmann@uni.lu>

SXXM seminar, 2010-06-30

Overview: Part I

- Symmetric cryptography (secret-key)
  - Block ciphers
  - Hash functions
  - Message authentication codes
  - Stream ciphers
  - Attacks

Goals and how to achieve them

- Confidentiality: Block ciphers, Stream ciphers, Public-key encryption schemes
- Integrity: Message authentication codes (MAC), Cryptographic hash functions, Digital signature schemes
- Key agreement

The standard toolchest

- Block ciphers: DES, 3DES, AES
- Stream ciphers: RC4, block cipher in CFB/OFB mode
- Message authentication codes: HMAC, CBC-MAC
- Cryptographic hash functions: MD5, SHA-1, SHA-256/384/512
- Digital Signature: RSA, (EC)DSA, ElGamal
- Public-key encryption: RSA, EC(IES)
- Key agreement: (EC)DH

Kerckhoffs & Shannon

"Il faut qu'il n'exige pas le secret, et qu'il pulsse sans inconvénient tomber entre les mains de l'ennemi"

Auguste Kerckhoffs, *La cryptographie militaire*, Journal des sciences militaires, vol. IX, pp. 5–83, Jan. 1883, pp. 161–191, Feb. 1883

"The enemy knows the system [being used]."

Claude Shannon, *Communication Theory of Secrecy Systems,* Bell Systems Technical Journal, Vol. 28, pp. 656–715, 1948.

Kerckhoffs & Shannon

“It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience”

Auguste Kerckhoffs, *La cryptographie militaire*, Journal des sciences militaires, vol. IX, pp. 5–83, Jan. 1883, pp. 161–191, Feb. 1883

“L'ennemi sait que le système (utilisé).”

Claude Shannon, *Communication Theory of Secrecy Systems,* Bell Systems Technical Journal, Vol. 28, pp. 656–715, 1948.

Block ciphers

- Formal definition: family of permutations

$$E_k : P \to C$$

with

$$K = \{0,1\}^l,\ \mathcal{P} = \{0,1\}^n,\ \mathcal{C} = \{0,1\}^m,\ K \in \mathcal{K}$$

and

$$D_k := E_k^{-1}$$

Confusion / Diffusion

- Terminology introduced by Shannon
- Confusion: local changes
- Diffusion: Spread changes widely

Iterated ciphers

- Make a weak function strong by applying it many times
- Usually function uses different key in each round
- Round key generated using *key schedule*
- Typical round count in modern ciphers: 8-80 rounds
- DES: 16 rounds
- AES-128/192/256: 10/12/14 rounds
- 3DES does not have 48 rounds!
  $3DES(P, (K_1||K_2||K_3) := DES(DES^{-1}(DES(P, K_1), K_2), K_3)$

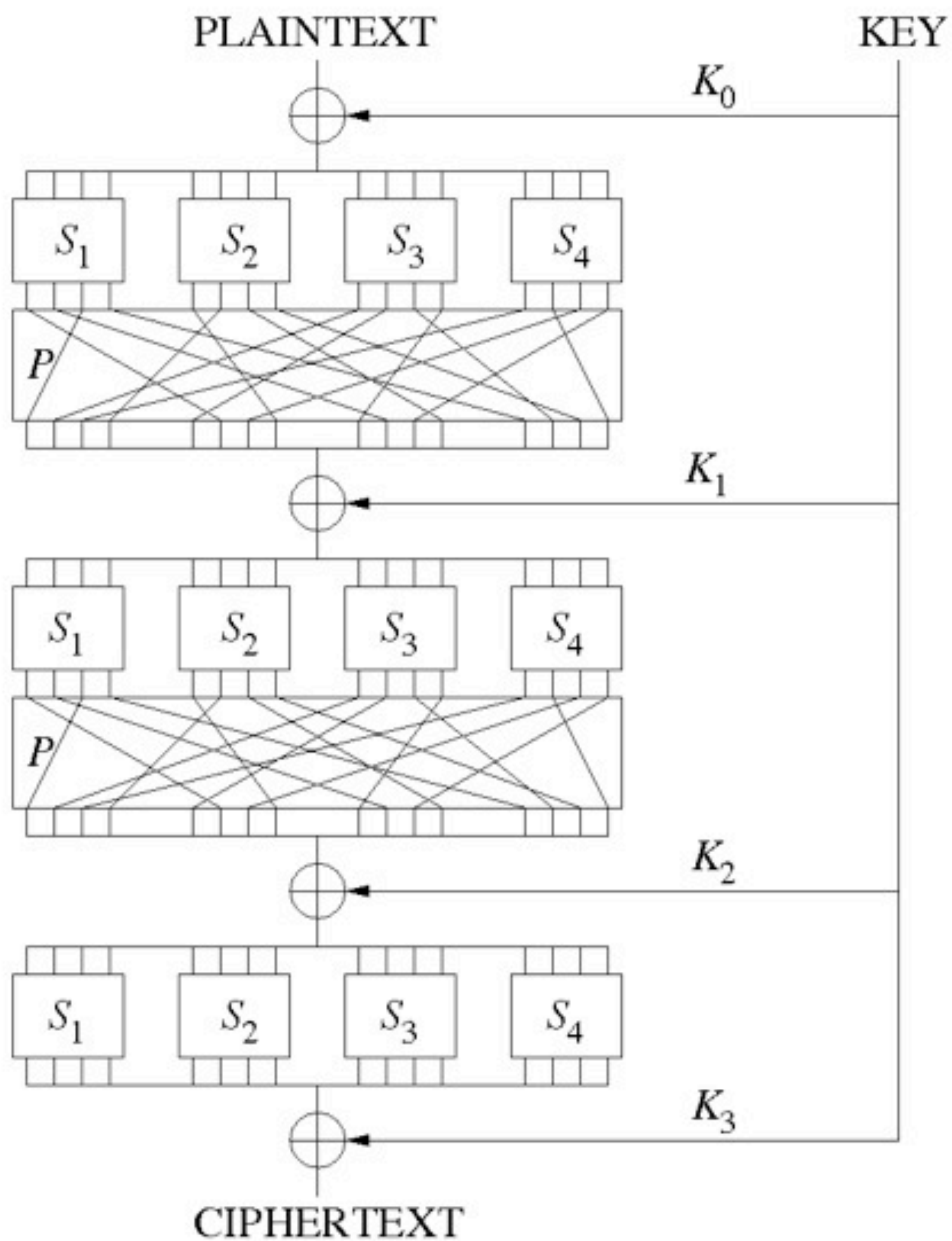
Rounds vs. cycles

- ***Round***: one iteration
- ***Cycle***: number of rounds necessary for each bit of the block to have been involved in target **and** source at least once
- Notion of cycle useful to compare strength of UFNs with other constructions

Common construction types

- Substitution-Permutation Networks (SPN)
  - oftentimes misnomer for: Substitution Linear Network
- Feistel Networks
- Unbalanced Feistel Networks (UFN)
- (Lai-Massey)

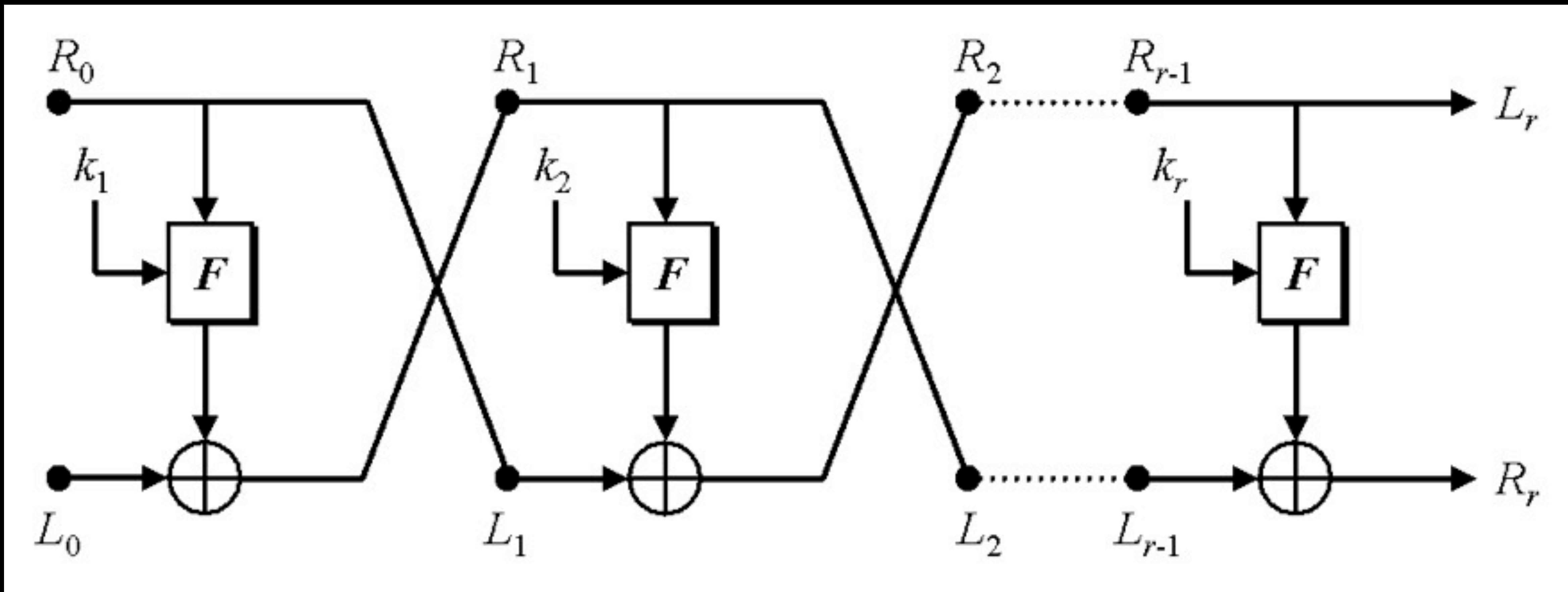
SP-Network



Picture credit: Wikipedia

SL-Network

Feistel network

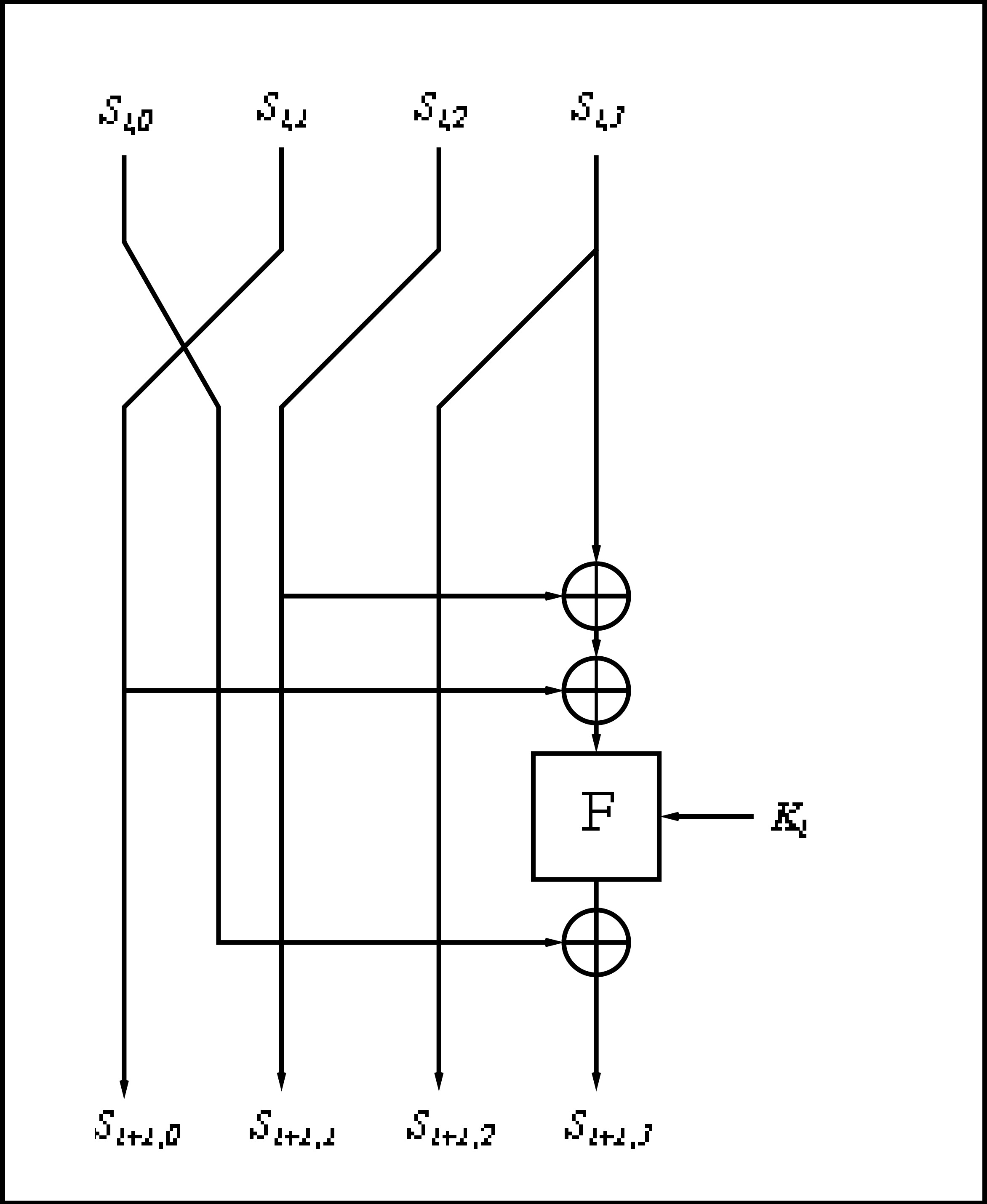


Picture credit: http://www.rsa.com/rsalabs/faq/images/feistel.gif

- Split block into left and right half (equal size)
  [tilt your head to the left for optimal viewing]

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

Unbalanced Feistel example



- One round of SMS4 (SMS4: 32 rounds)
- Source-heavy (96:32) UFN
- Complete
- Homogeneous

... from block ciphers
to
cryptographic hash functions ...

Cryptographic hash functions

- What properties do we expect from a CHF?
  - Preimage resistance (effort to compute 2^n)
  - Second-preimage resistance (effort to compute 2^n)
  - Collision resistance (b-day bound, effort $2^{n/2}$)
  - behaves like a *Random Oracle*

Aside: Random oracles

- Idealized theoretical abstraction:
  - For every query *Q* in input domain return **uniformly distributed random** response *R* from output domain.
  - For sequence of queries (q_1,..., q_n) with associated responses (r_1,...,r_n) whenever $q_i = q_j$ it is required that also $r_i = r_j$ holds.
- And yes, we also believe in Santa Claus.

Construction hash functions

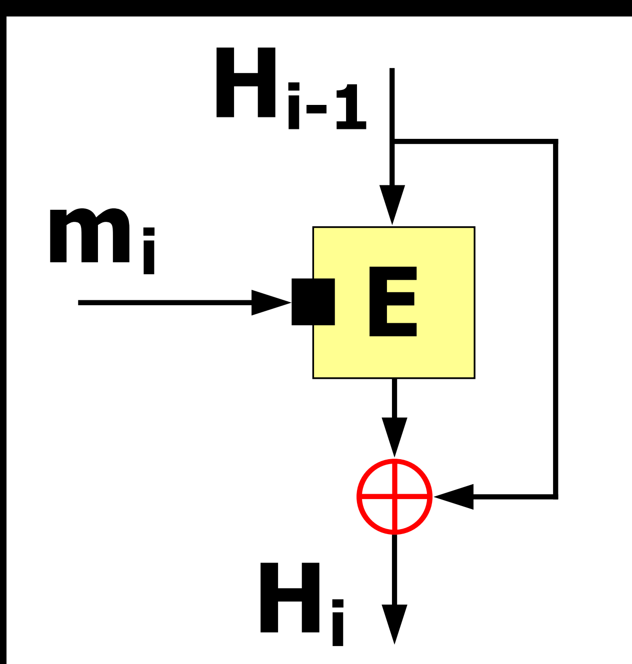
- Traditionally: Merkle-Damgåard
- Pad message to block multiple, append length (MD strengthening), chop message M into blocks $m_1, \ldots, m_k$, apply compression function repeatedly:

$$H = C(m_k, \ldots C(m_2, C(m_1, C(IV)))$$

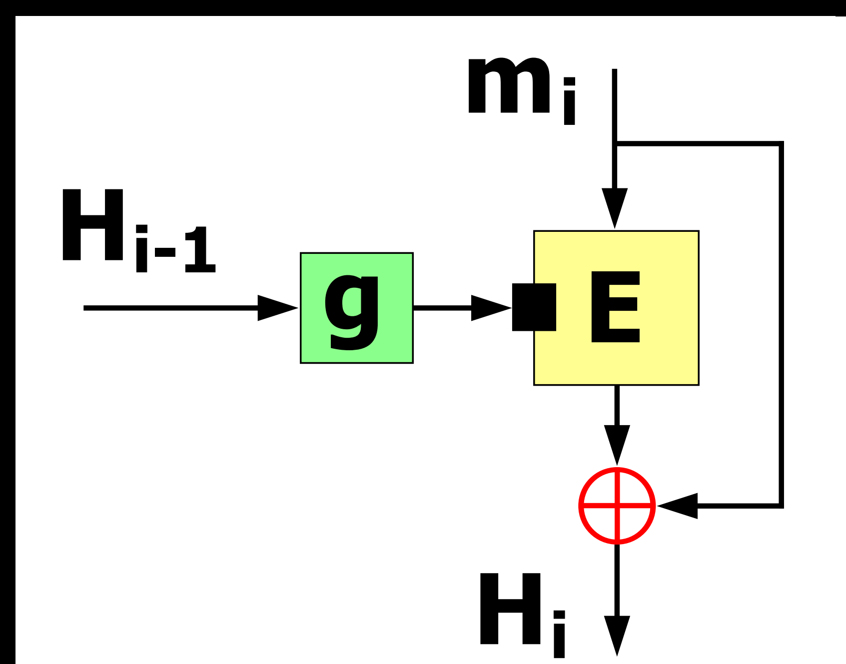
- Why? Security proof:
  compression function secure => CHF secure
- Many SHA-3 candidates use other construction principles: Double/Wide-pipe, Sponge

Compression functions

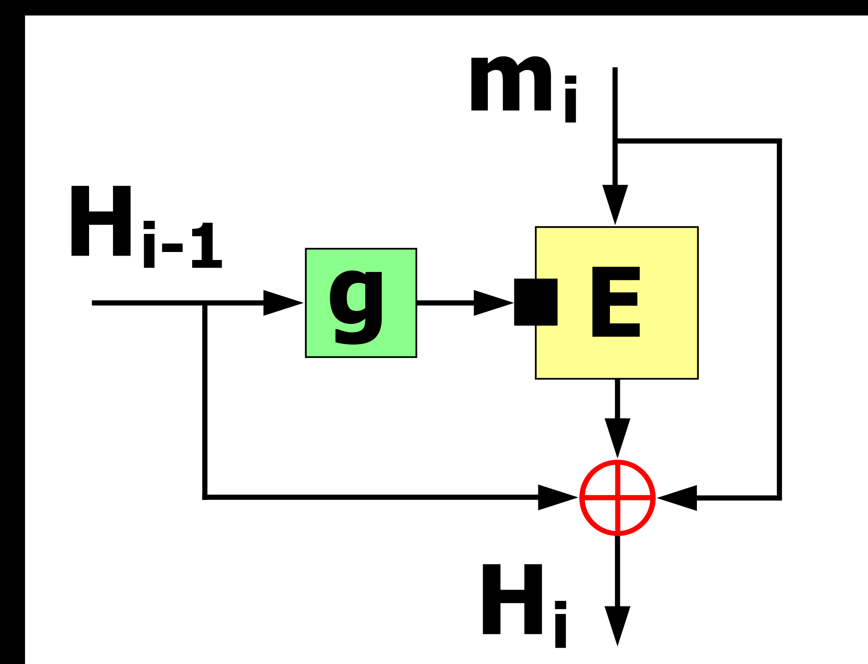
- can be built from block ciphers
- Davies-Meyer construction most common



Davies-Meyer



Matyas–Meyer–Oseas



Miyaguchi–Preneel

Picture credit: Wikipedia

Attacks on block ciphers

- Goals: Key-recovery,
  distinguish from pseudo-random permutation
- Level of access:
  Ciphertext-only < Known text < Chosen text <
  Adaptive chosen-text

  <

  Related-key attacks < Related subkey attacks

  <

  Known-key attacks < Chosen-key attacks

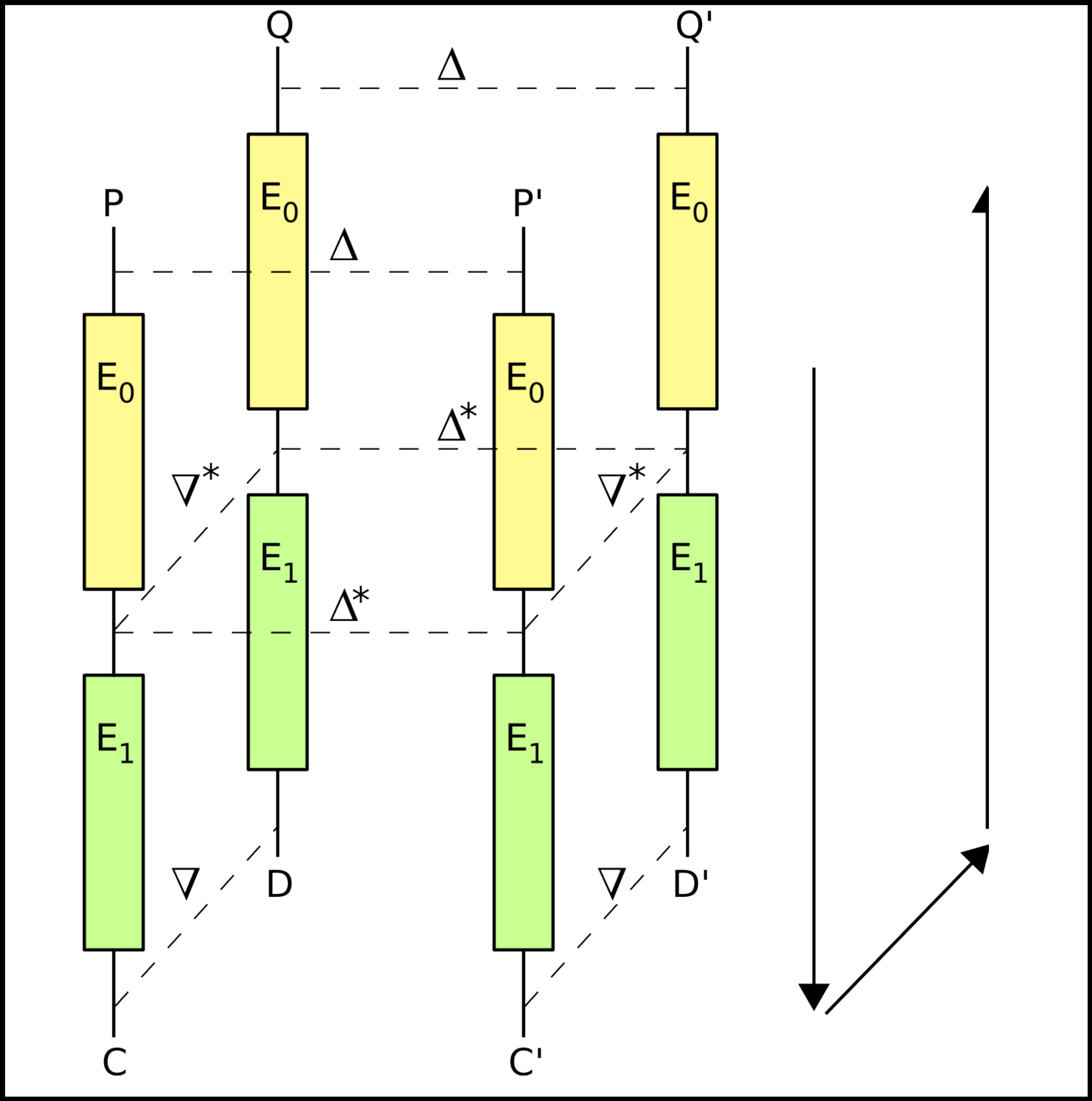
Attack methods: DC

- Most powerful tool still is *differential cryptanalysis* (chosen plaintext attack)
- <u>**Idea**</u>: | **Prob(** $\boldsymbol{F(X) - F(X + \Delta_I) = \Delta_O}$ **) - 0.5 |** is large
- Pair **(**Δ_I, Δ_O**)**: *differential characteristic*
- Characteristics can be chained
- *Differential (trail)*: multiple interfering characteristics
- <u>**Attack**</u>: build trail over *N-R* rounds, recover round keys for *R* rounds by distinguishing using many pairs

Attack methods: LC

- Linear cryptanalysis: known plaintext attack
- **Idea**: approximate non-linear functions with linear ones
- bias of a linear function *L* for target function *F*:
  $|Pr(L(X) = F(X)) - 0.5|$
- find best approximation (highest bias) locally, chain approximations over many rounds
- **Attack**: assume linear approximation holds over N-R rounds, recover key for last R rounds using counters

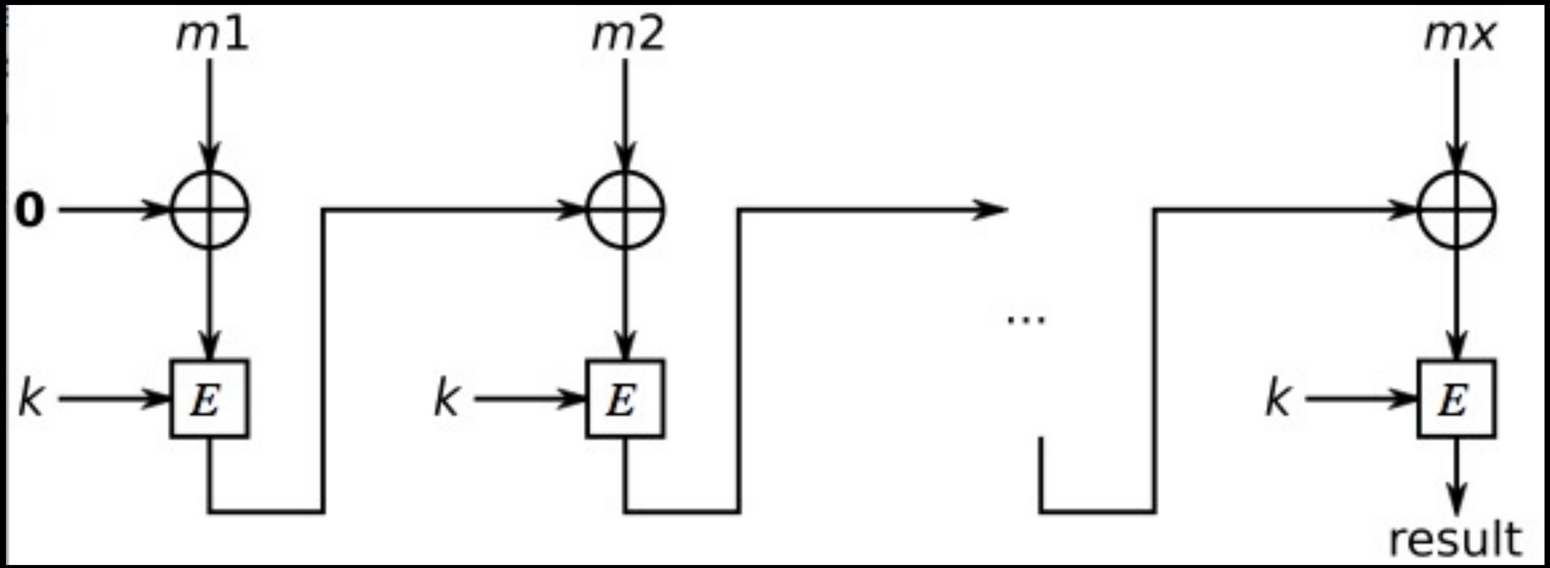
Boomerang attacks



- adaptive chosen plaintext/ciphertext attack
- divide and conquer principle (break cipher into two parts)
- needs access to encryption ***and*** decryption oracle!

MACs

- Objective: using a shared key, authenticate a message using a *tag*
- CBC-MAC: use block cipher in CBC mode



- HMAC: use CHF
  $HMAC(K,m) = H((K \oplus opad) \| H((K \oplus ipad) \| m))$
  $ipad = 0x5c5c5c...5c5c$, $opad = 0x363636...3636$

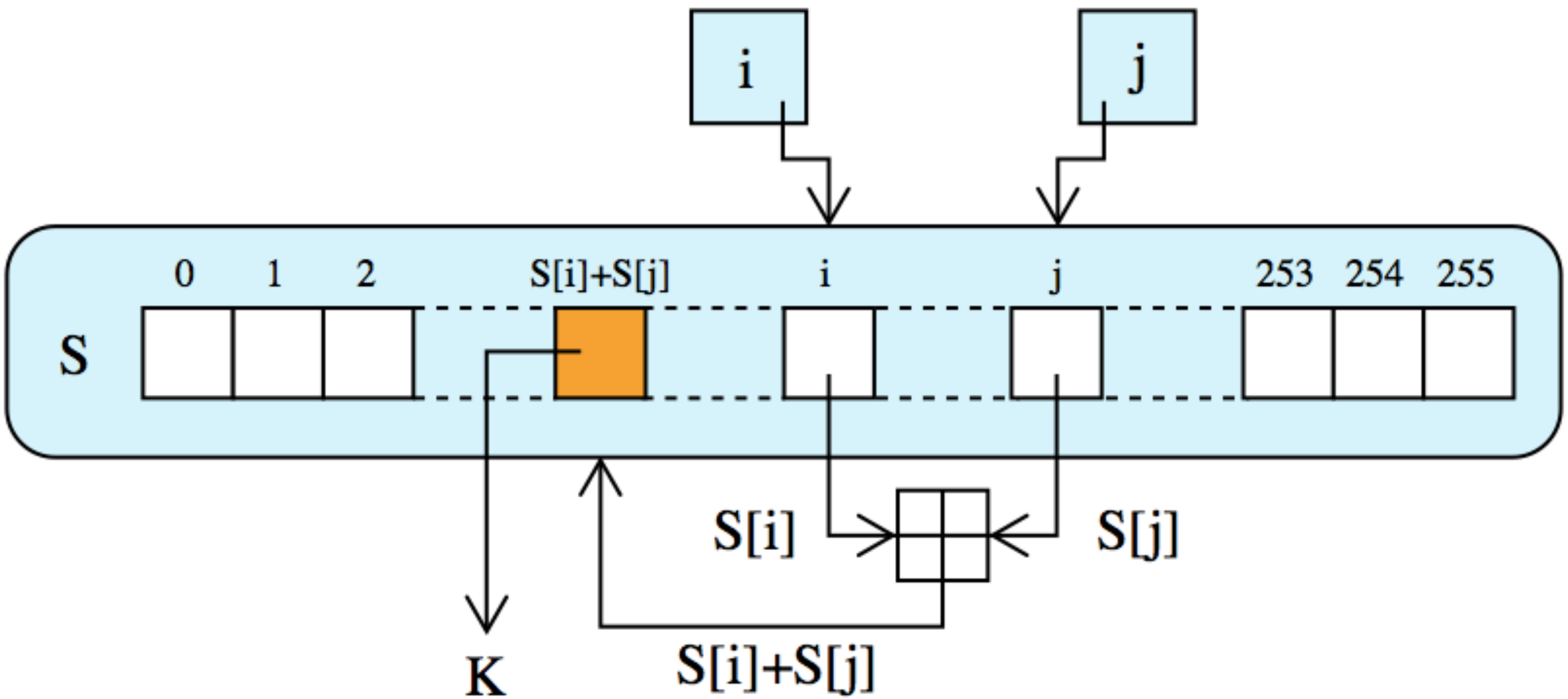
Stream ciphers

- Synchronous: generate stream of pseudo-random bits independently of plaintext/ciphertext, combine stream & text (using XOR usually)
- Self-synchronizing (asynchronous, auto-key): use plaintext to generate keystream, synchronizes if bits are deleted/added to stream

Most common example

- Historically: based on LFSRs (irregularly clocked, combination generator, filter generator)
- Examples: A5/1 (GSM), E0 (Bluetooth), DSC (DECT)
- widespread in software: RC4
- Used in SSL, WEP, TKIP

RC4



Overview: Part II

- Asymmetric cryptography (public-key)
  - RSA
  - Diffie-Hellmann, ElGamal, Schnorr, DSA
  - Elliptic Curve Cryptography
  - Pairings, Identity-based cryptography
  - Attacks