Cryptography

Lecture 9: Authenticated Encryption

December 10, 2024

Overview

Contents

1. Classical cryptography
   (Shift & Vigenère cipher, one-time pad, perfect secrecy)
2. Security definitions & threat models
   (Computational security, CPA & CCA)
3. Private-key cryptography
   (Message authentication, hash functions, primitives, relevant ciphers)
4. Public-key cryptography
   (Assumptions, key management, digital signatures, relevant ciphers)

Message Authentication Codes

Alice

Eve

Message Authentication Codes

Alice
$k \leftarrow \text{gen}(1^n)$

Security parameter 1^n

Eve

Message Authentication Codes

Alice

$k \leftarrow \text{gen}(1^n)$

Security parameter 1^n

Message m & code c

Eve

Message Authentication Codes

Alice

$k \leftarrow \mathsf{gen}(1^n)$

$b = \mathsf{val}_k(m, c)$

$b' = \left((m, c) \overset{?}{\notin} Q\right)$

Security parameter 1^n

Message m & code c

Eve

Message Authentication Codes

Alice

$k \leftarrow \text{gen}(1^n)$

$b = \text{val}_k(m, c)$

$b' = \left((m, c) \overset{?}{\notin} Q\right)$

Return $b \wedge b'$

Security parameter 1^n

Message m & code c

Eve

Definition (§7.4 Secure MAC)

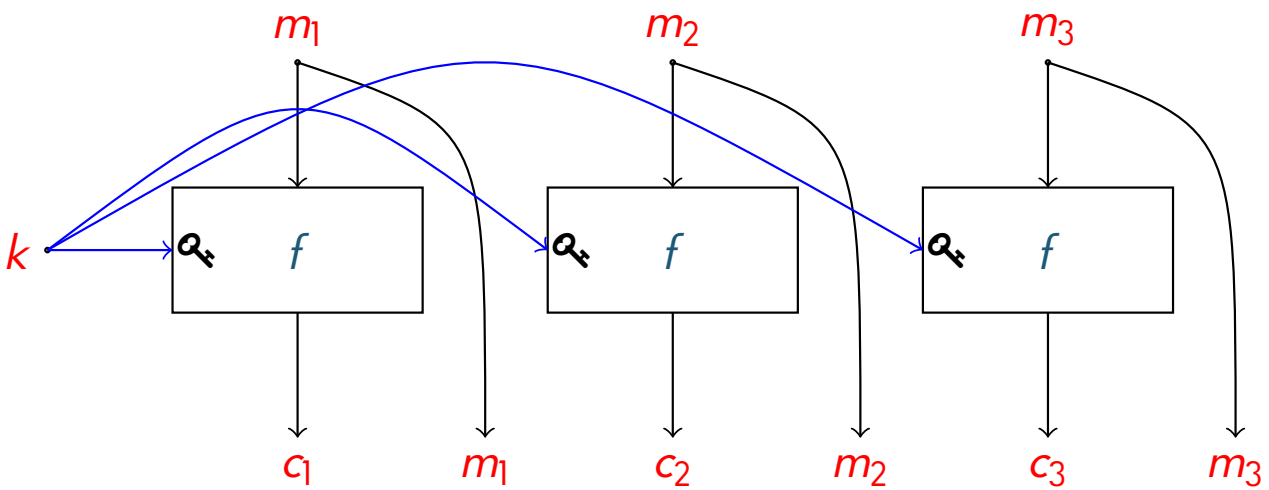
Message authentication scheme $\mathcal{E} = (\text{gen}, \text{mac}, \text{val})$ is secure
if for every PPT algorithm $\mathcal{A}$ with access to mac-oracle (encryption oracle)

$$\mathrm{P}\left[\text{Forge}^{\text{MAC}}_{\mathcal{E},\mathcal{A}}(n)\right] \simeq 0$$

Message Authentication Codes

Pseudorandom function f and message $m = m_1 m_2 m_3$

Electronic Code Book (ECB)



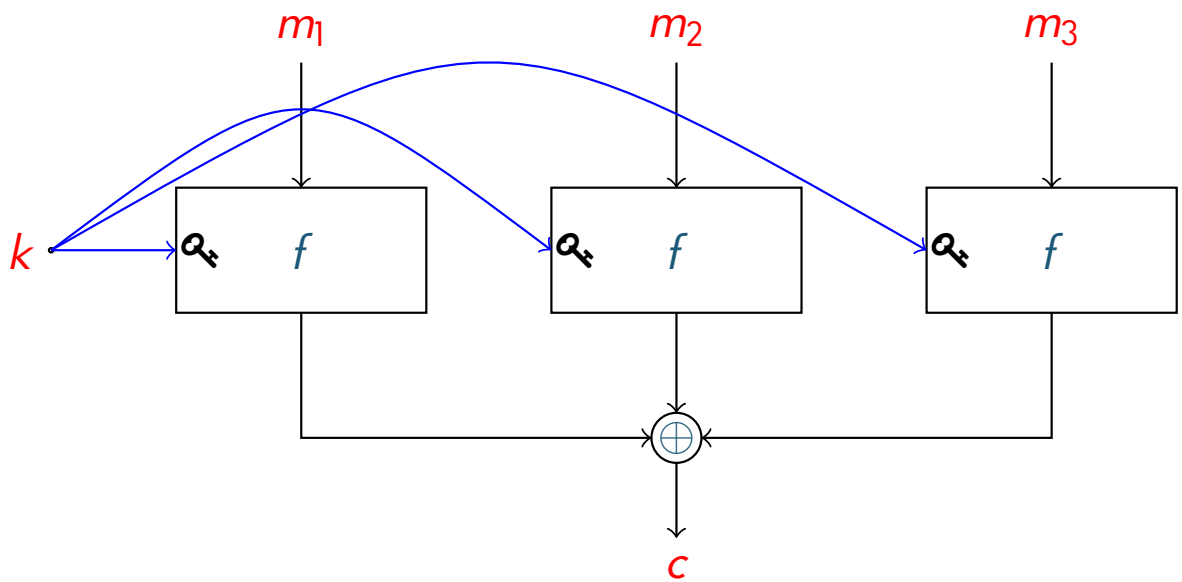
Secure MAC
✗

Messages will always be output by MACs (will not be specially indicated)

Message Authentication Codes

Pseudorandom function f and message $m = m_1 m_2 m_3$

XORed ECB



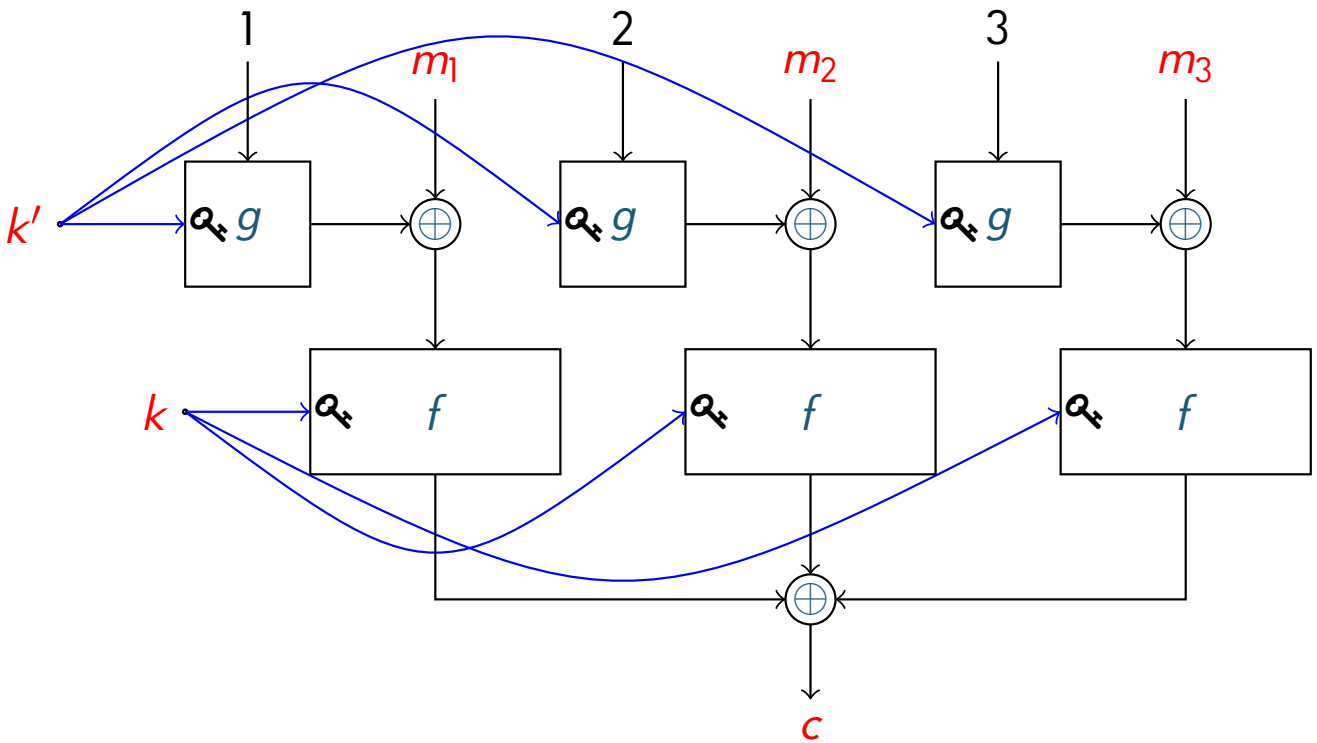
Secure MAC
✗

Adversary can extend message to $m_1 m_2 m_3 mm$ with valid code c

Message Authentication Codes

Pseudorandom functions f, g and message $m = m_1 m_2 m_3$

Padded XORed ECB



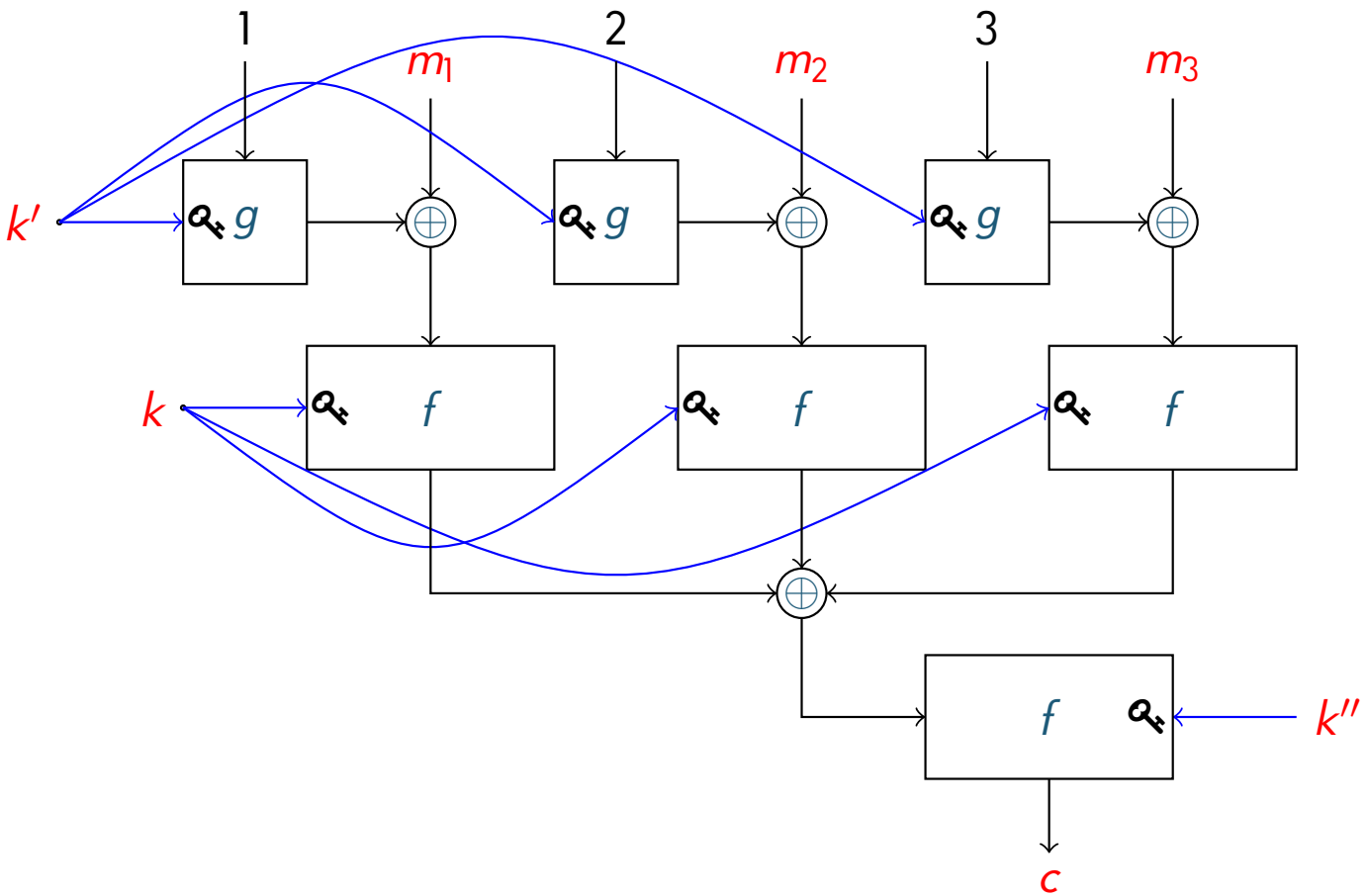
Secure MAC
✗

Adversary queries for code c' of m' and code c_1 of m_1.
Then $c \oplus c_1 \oplus c'$ is valid for message to $m' m_2 m_3$

Message Authentication Codes

Pseudorandom functions f, g and message $m = m_1 m_2 m_3$

Parallel MAC (PMAC)

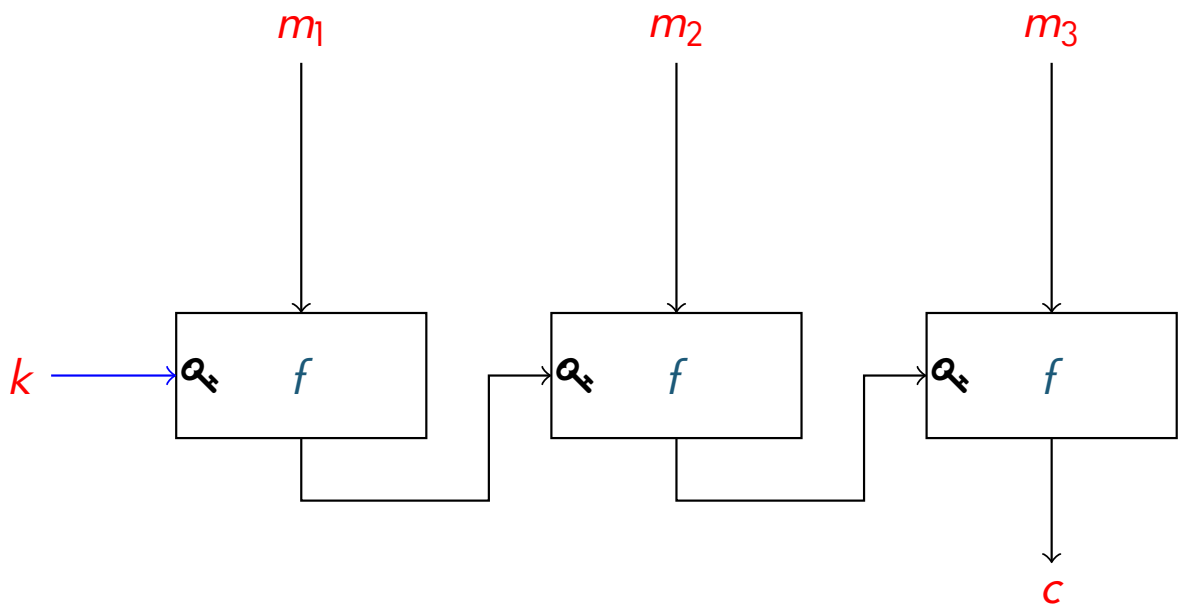


Secure MAC
✓

Message Authentication Codes

Pseudorandom function f and message $m = m_1 m_2 m_3$

Cascade



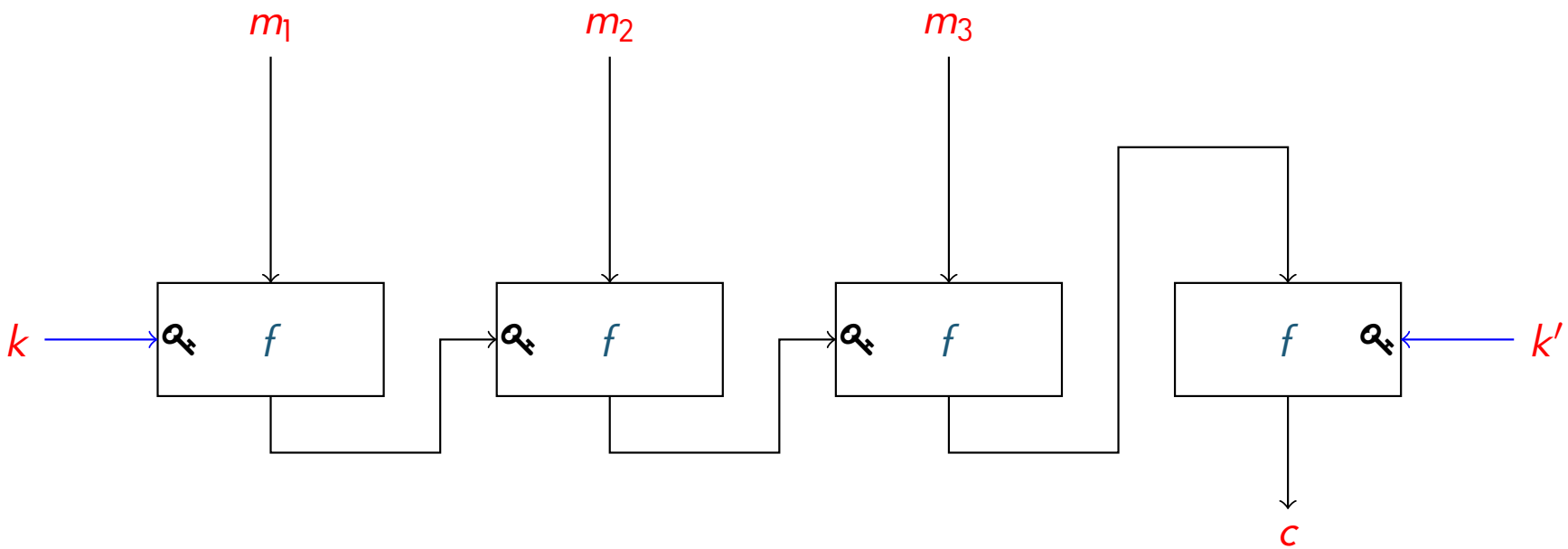
Secure MAC
✗

Adversary can extend message to $m_1 m_2 m_3 m$ with valid code $f_c(m)$

Message Authentication Codes

Pseudorandom function f and message $m = m_1 m_2 m_3$

Nested MAC (NMAC)

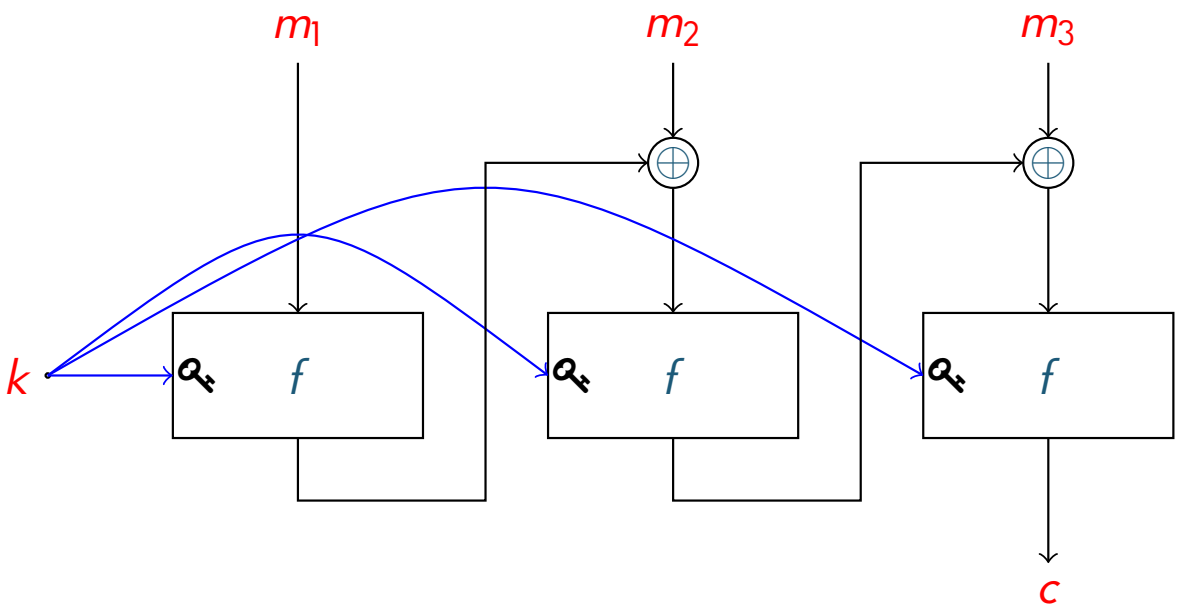


Secure MAC
✓

Message Authentication Codes

Pseudorandom function f and message $m = m_1 m_2 m_3$

Cipher Block Chaining MAC (CBC-MAC)



Secure MAC
✗

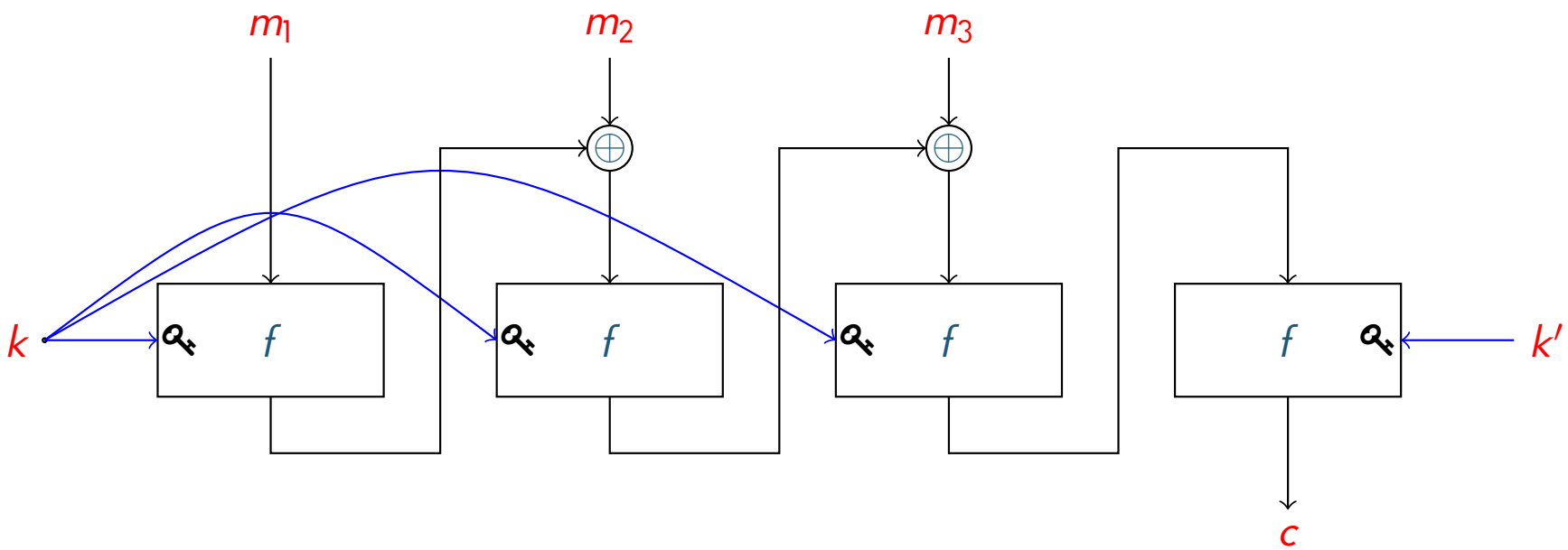
(Secure if number of message parts is universally fixed)

Adversary queries for code c_1 of m_1 and
forges valid code c for message $(m_2 \oplus c_1)m_3$

Message Authentication Codes

Pseudorandom function f and message $m = m_1 m_2 m_3$

Encrypted Cipher Block Chaining (ECBC)



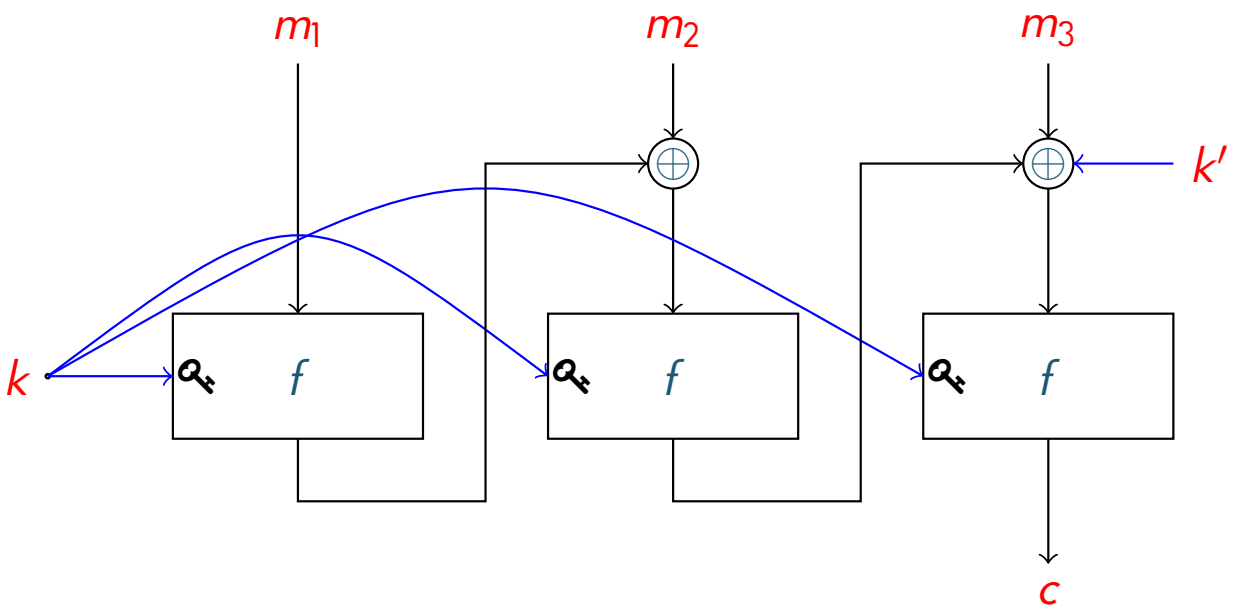
Secure MAC
✓

Message Authentication Codes

Pseudorandom function f and message $m = m_1 m_2 m_3$

Variant of CBC-MAC (CMAC)

(NIST standard)



Secure MAC
✓

Message Authentication Codes

Common attacks against MACs

- Block removal, reordering, duplication, or addition
- "Mix-&-Match" blocks from multiple messages

Notes

- Use known secure modes
- MACs based on collision-resistant hash functions later

Unconditional Message Authentication Codes

One-time MAC

- Analogue of one-time pad
- Secure against any adversary
- Key again only used for single MAC (new key for each MAC)

§8.2 Construction (One-time MAC)

For every n let $q_n \in \mathbb{N}$ be prime such that $2^n < q_n < 2^{n+1}$. Construct message authentication scheme $(\mathrm{gen}, \mathrm{mac}, \mathrm{val})$ with canonical validation

- $\mathrm{P}^n_K(\langle k, k' \rangle) = q_n^{-2}$ for all $k, k' \in \{0, \ldots, q_n - 1\}$ (each key length $n+1$)
- $\mathrm{mac}_{\langle k,k' \rangle}(m) = (mk + k') \bmod q_n$ for all $k, k' \in K$ and $m \in M$ with $|k| = n+1 = |k'|$ and $|m| = n$
  (we freely use binary sequences & number they represent)

Unconditional Message Authentication Codes

§8.3 Theorem (Unconditionally secure one-time MAC)

Let $\mathcal{E}$ message authentication scheme of Construction §8.2.

$$\mathrm{P}\left[\mathrm{Forge}^{\mathrm{MAC}}_{\mathcal{E},\mathcal{A}}(n)\right] \leq 2^{-n}$$

for all $n \in \mathbb{N}$ and any adversary that may query mac oracle only once (even non-PPT adversaries)

Proof

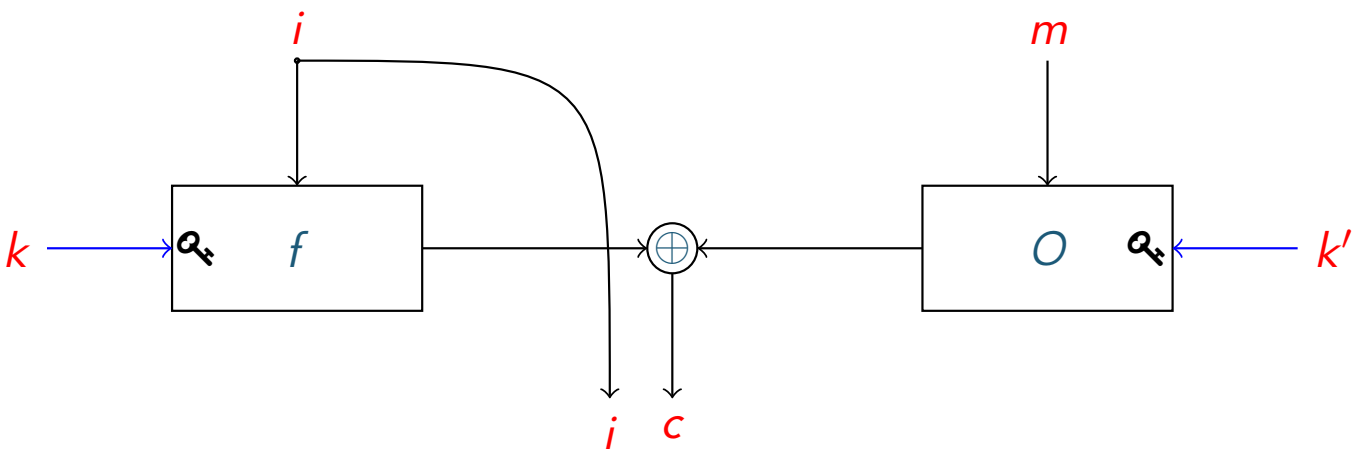
Relatively straightforward exercise

Show that two queries completely break security □

Message Authentication Codes

Pseudorandom function f, secure one-time MAC O, message m, and random i

Carter-Wegman MAC (CWMAC)



Secure MAC
✓

Authenticated Encryption

Authenticated encryption

- Schemes that achieve both: secrecy & integrity
- Notion of secrecy: CCA-secure
- Notion of integrity: Unforgeability

Authenticated Encryption

§8.4 Definition (Unforgeable encryption game)

Let $n \in \mathbb{N}$, $\mathcal{E} = (\mathrm{gen}, \mathrm{enc}, \mathrm{dec})$ efficient private-key encryption scheme and $\mathcal{A}$ stateful algorithm. Unforgeable encryption game $\mathrm{Forge}^{\mathrm{ENC}}_{\mathcal{E},\mathcal{A}}(n)$ is

1. Security parameter 1^n sent to adversary $\mathcal{A}$ and $k \leftarrow \mathrm{gen}(1^n)$

   $\mathcal{A}$ selects ciphertext $c \in C$ 2

3. $m = \mathrm{dec}_k(c)$ and return $\Big(m \neq \bot\Big) \wedge \Big(m \notin Q\Big)$

   where Q set of messages forwarded to encryption oracle by $\mathcal{A}$

Alice

$k \leftarrow \mathrm{gen}(1^n)$ $\quad m = \mathrm{dec}_k(c)$

Return $(m \neq \bot) \wedge (m \notin Q)$

Security parameter 1^n

Ciphertext c

Eve

Authenticated Encryption

§8.5 Definition (Unforgeable)

Efficient private-key encryption scheme $\mathcal{E} = (\mathrm{gen}, \mathrm{enc}, \mathrm{dec})$ is unforgeable if for every PPT algorithm $\mathcal{A}$ with access to encryption oracle

$$\mathrm{P}\left[\mathrm{Forge}^{\mathrm{ENC}}_{\mathcal{E},\mathcal{A}}(n)\right] \simeq 0$$

§8.6 Definition (Authenticated encryption scheme)

Efficient private-key encryption scheme is authenticated encryption scheme if it is CCA-secure and unforgeable

Authenticated Encryption

Components

- CPA-secure private-key encryption scheme $\mathcal{E} = (\mathsf{gen}, \mathsf{enc}, \mathsf{dec})$
- Secure message authentication scheme $\mathcal{E}' = (\mathsf{gen}', \mathsf{mac}, \mathsf{val})$
- Construct authenticated encryption scheme $\underline{\mathcal{E}} = (\underline{\mathsf{gen}}, \underline{\mathsf{enc}}, \underline{\mathsf{dec}})$ with independent key generation $\mathsf{P}_{K \times K'}(\langle k, k' \rangle) = \mathsf{P}_K(k) \cdot \mathsf{P}_{K'}(k')$

Three generic approaches for encryption $\underline{\mathsf{enc}}_{\langle k,k' \rangle}(m)$

1. ~~Encrypt-and-authenticate:~~ $c \leftarrow \mathsf{enc}_k(m)\,;\, c' \leftarrow \mathsf{mac}_{k'}(m)\,;\, \text{return } \langle c, c' \rangle$
2. Authenticate-then-encrypt: $c' \leftarrow \mathsf{mac}_{k'}(m)\,;\, c \leftarrow \mathsf{enc}_k(mc')\,;\, \text{return } c$
3. Encrypt-then-authenticate: $c \leftarrow \mathsf{enc}_k(m)\,;\, c' \leftarrow \mathsf{mac}_{k'}(c)\,;\, \text{return } \langle c, c' \rangle$

Authenticated Encryption

Decryption for approaches ❷ and ❸

❷ Authenticate-then-encrypt: $\underline{\mathrm{dec}}_{\langle k,k'\rangle}(c)$ (e.g. SSL)

$$mc' = \mathrm{dec}_k(c)\ ;\ \text{if}\ \mathrm{val}_{k'}(m, c')\ \text{then return}\ m\ \text{else return}\ \perp$$

❸ Encrypt-then-authenticate: $\underline{\mathrm{dec}}_{\langle k,k'\rangle}(\langle c, c'\rangle)$ (e.g. IPsec)

$$\text{if}\ \mathrm{val}_{k'}(c, c')\ \text{then return}\ \mathrm{dec}_k(c)\ \text{else return}\ \perp$$

§8.7 Theorem (Authenticated encryption scheme)

Construction ❸ is authenticated encryption scheme
for every CPA-secure encryption scheme and
secure message authentication scheme

Authenticated Encryption

Construction 2

- Can yield authenticated encryption scheme under same assumptions
  (multiple points of failure → padding-oracle attack)
  → careful, use established combinations
- Emotionally preferred since integrity provided directly on message

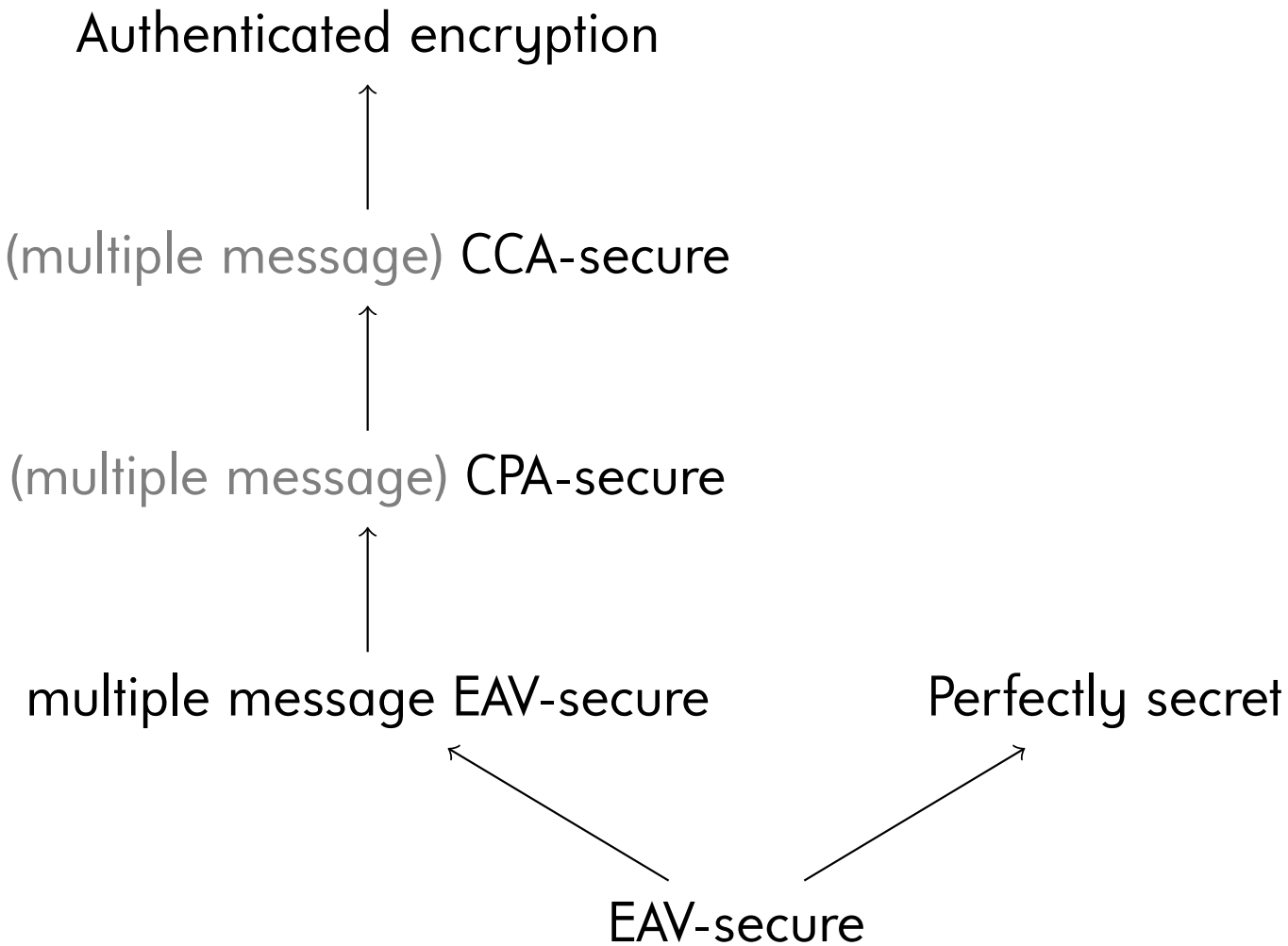
Standards

- CCM (Counter with CBC-MAC) — Construction 2
  CBC-MAC then CTR mode encryption (e.g. TLS 1.2, 802.11i)
- GCM (Galois/Counter mode) — Construction 3
  CTR mode encryption then CW-MAC (e.g. TLS 1.3, 802.1AE, 802.11ad)
- EAX (Encrypt-then-authenticate-then-translate) — Construction 3
  CTR mode encryption then OMAC (≈ CMAC)
  (potential successor of CCM; ANSI C12.22; see literature for details)

Summary of Encryption Standards

Standard	Typical primitives (implementation)	Attacker model
Perfectly secret	One-time pad	Unrestricted but passive
EAV-secure	Pseudorandom generator	passive PPT
CPA-secure	Pseudorandom function	PPT with access to encryption
CCA-secure	CPA-secure + secure MAC	PPT with access to encryption & decryption
Authenticated encryption	CCA-secure + secure MAC	PPT with access to encryption & decryption

Summary of Encryption Standards



Proposals for Pseudorandom Functions — DES

DES

- Block cipher “Data Encryption Standard” (successor of IBM’s Lucifer)
- Was extremely popular due to its speed & simplicity
- Replaced by AES in 2001 because broken in 1997 by brute force
- Utilizes Feistel network construction
  (inspired by Shannon’s “confusion & diffusion”)

Horst Feistel (* 1915; † 1990)

- US-German cryptographer
- Head of cryptography at IBM
- Previously developed IFF systems

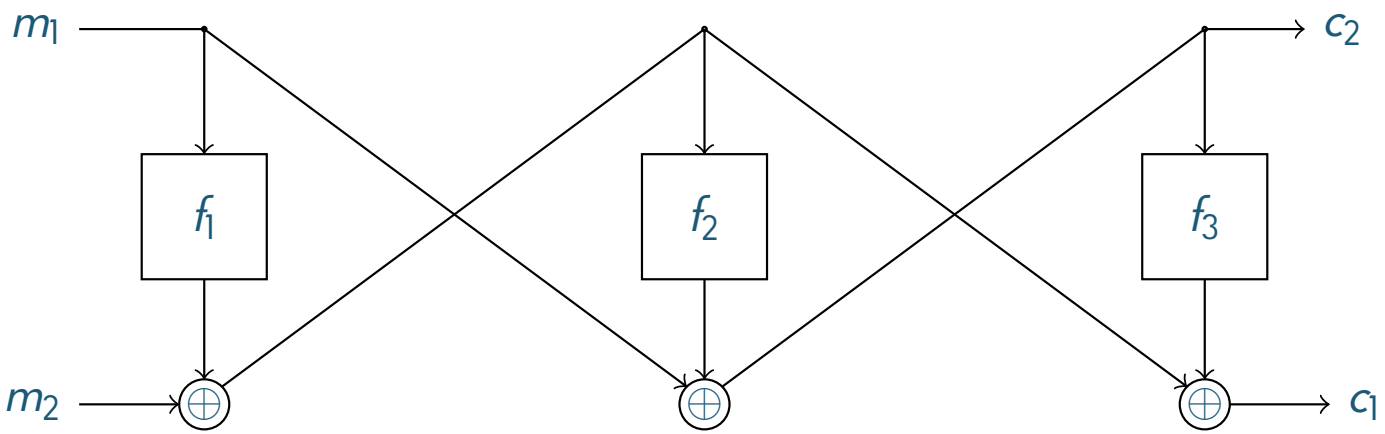
© IBM

Proposals for Pseudorandom Functions — DES

Feistel networks

- ℓ rounds using computable functions $f_1, \ldots, f_\ell \colon \{0,1\}^n \to \{0,1\}^n$
- Round function $r_i(m_1, m_2) = \langle f_i(m_1) \oplus m_2,\ m_1 \rangle$ for $1 \le i \le \ell$
- Computes $\mathrm{FN}(f_1, \ldots, f_\ell) \colon \{0,1\}^{2n} \to \{0,1\}^{2n}$ with $m \mapsto (r_1 ; \cdots ; r_\ell)(m)$

Illustration of $\mathrm{FN}(f_1, f_2, f_3)$ for input $m_1 m_2$ and output $c = c_1 c_2$



§9.1 Theorem (Feistel property)

Function $\mathrm{FN}(f_1, \ldots, f_\ell)$ computed by Feistel network is effectively invertable (bijective & inverse effectively computable)

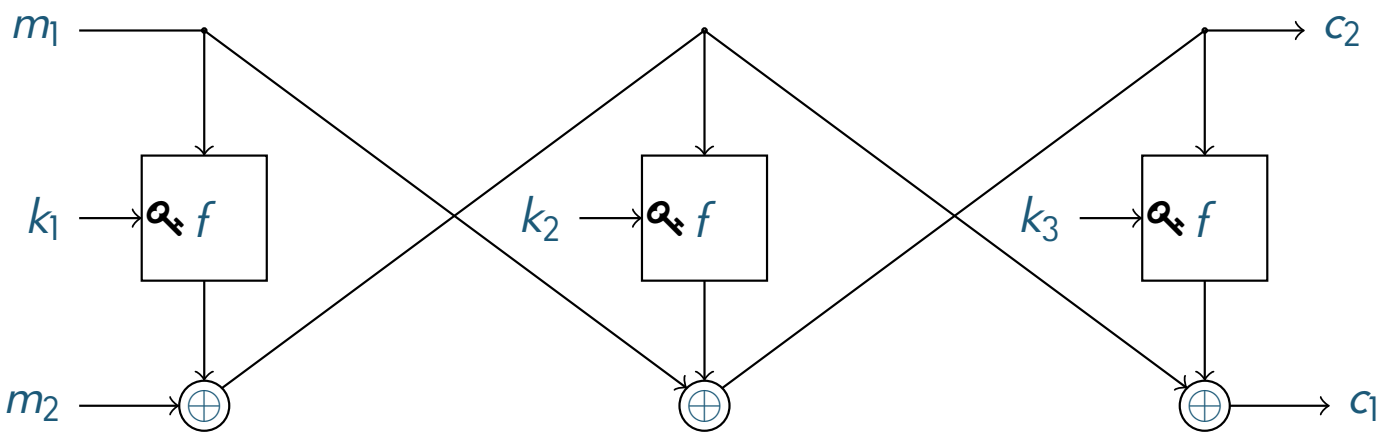
Proposals for Pseudorandom Functions — DES

(wrong type according to our definitions)

§9.2 Theorem [Luby & Rackoff 1988]

Given pseudorandom function f and 3 independent keys k_1, k_2, k_3,
Feistel network $\mathrm{FN}(f_{k_1}, f_{k_2}, f_{k_3})$ computes pseudorandom permutation
$g\colon \{0,1\}^{5n} \to \{0,1\}^{2n}$ given by $g(k_1, k_2, k_3, w) = \mathrm{FN}(f_{k_1}, f_{k_2}, f_{k_3})(w)$

Illustration of $\mathrm{FN}(f_{k_1}, f_{k_2}, f_{k_3})$ for input $m_1 m_2$ and output $c = c_1 c_2$



Proposals for Pseudorandom Functions — DES

Characteristics of DES

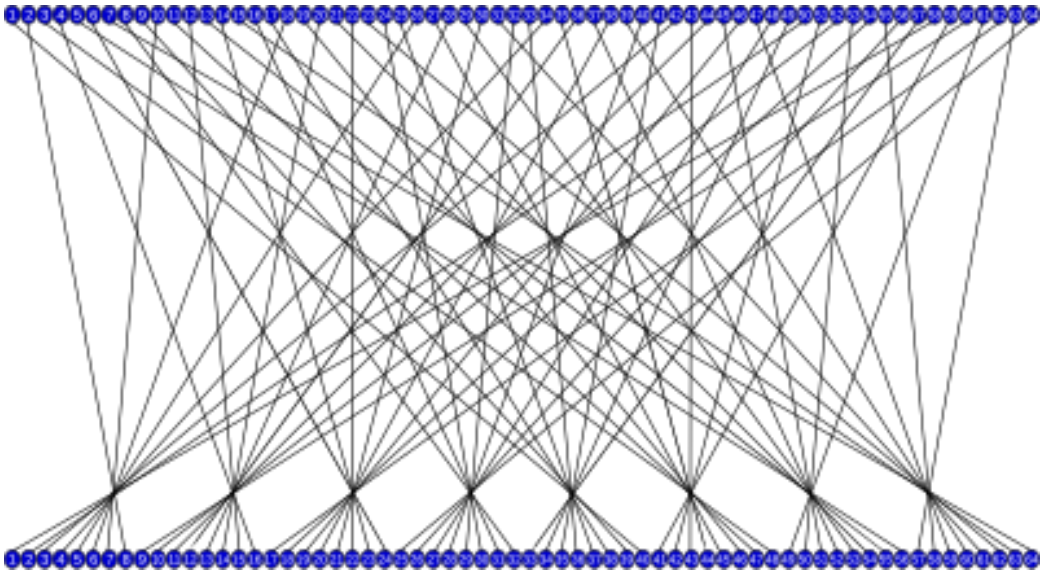
- Block size 64 bits and key size 56 bits
- Initial permutation IP (1st bit becomes 40th bit)
- 16-round Feistel network with same round function $f_{k_1}, \ldots, f_{k_{16}}$

```
IP = (40,  8,48,16,56,24,64,32,
      39,  7,47,15,55,23,63,31,
      38,  6,46,14,54,22,62,30,
      37,  5,45,13,53,21,61,29,
      36,  4,44,12,52,20,60,28,
      35,  3,43,11,51,19,59,27,
      34,  2,42,10,50,18,58,26,
      33,  1,41,  9,49,17,57,25)
```

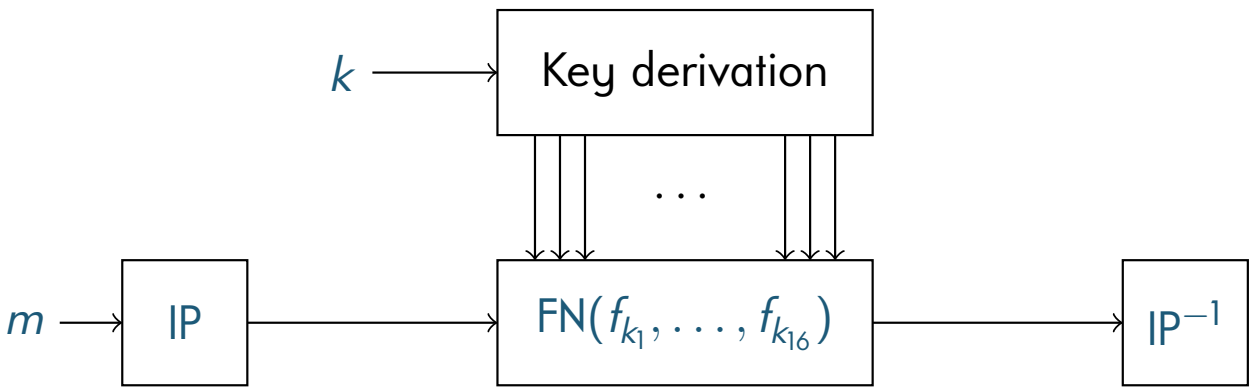
Proposals for Pseudorandom Functions — DES

Characteristics of DES

- Block size 64 bits and key size 56 bits
- Initial permutation IP (1st bit becomes 40th bit)
- 16-round Feistel network with same round function $f_{k_1}, \ldots, f_{k_{16}}$



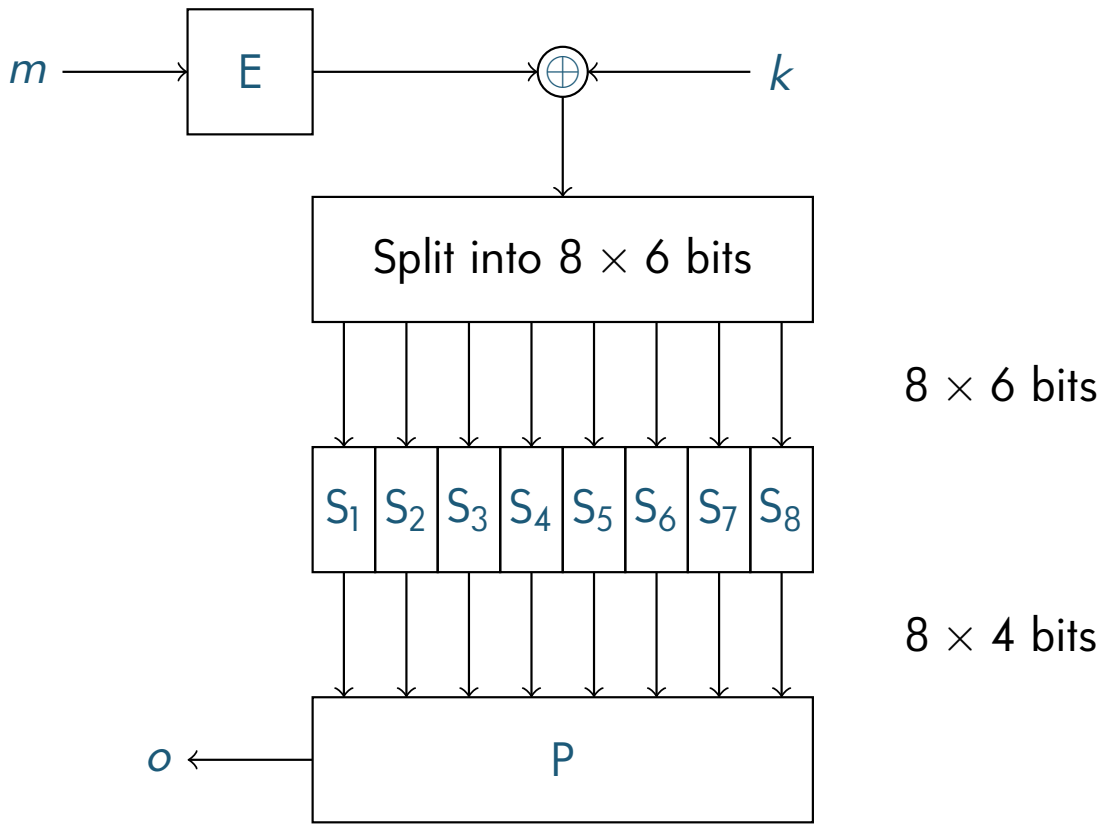
© Bourrichon



Proposals for Pseudorandom Functions — DES

Round function f_k with 48 bit key k and 32 bit input m

- Expansion E from 32 to 48 bits (see appendix)
- Substitution boxes $S_1, \ldots, S_8$ mapping 6 to 4 bits (see appendix)
- Permutation P of 32 bits (see appendix)



Proposals for Pseudorandom Functions — DES

Final notes on DES

- Use strongly discouraged (key length 56 bits inadequate)
  $\rightarrow$ use more modern block ciphers like AES with at least 256 bit keys
- If use cannot be avoided, then use Triple DES (3DES)
  (triple DES encryption with 3 indepedent keys,
  which effectively increases key size to 112 bits)

Summary

- Practical message authentication schemes
- Authenticated encryption
- DES

Appendix — DES

Expansion E of 32 to 48 bits

```
(32, 1, 2, 3, 4, 5,
 4, 5, 6, 7, 8, 9,
 8, 9,10,11,12,13,
12,13,14,15,16,17,
16,17,18,19,20,21,
20,21,22,23,24,25,
24,25,26,27,28,29,
28,29,30,31,32, 1)
```

(read: first bit of output
is 32nd bit of input)

Key generation & substitution boxes

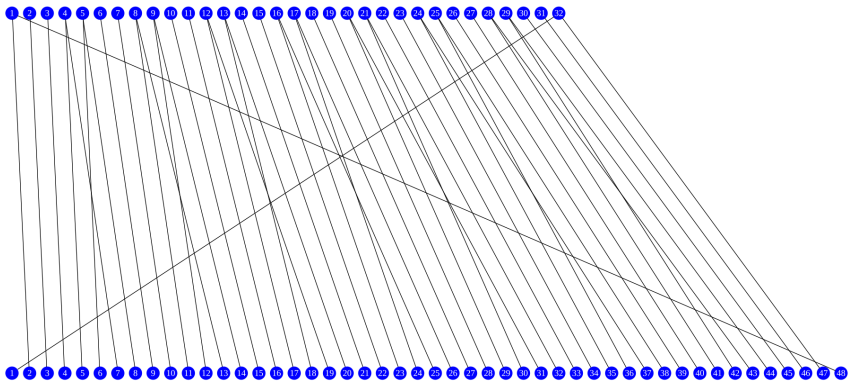
▸ Wikipedia

Permutation P of 32 bits

```
( 9,17,23,31,13,28, 2,18,
24,16,30, 6,26,20,10, 1,
 8,14,25, 3, 4,29,11,19,
32,12,22, 7, 5,27,15,21)
```

Appendix — DES

Expansion E of 32 to 48 bits



Key generation & substitution boxes

▸ Wikipedia

Permutation P of 32 bits