Computer Science and Information Engineering
National Chi Nan University

The Principle and Application of Secret Sharing

Dr. Justie Su-Tzu Juan

Lecture 2. Fundamental and Technology of Cryptography

§ 2.3 Symmetric Cryptosystem

Slides for a Course Based on the Text
密碼學與網路安全
by 王旭正、柯宏叡

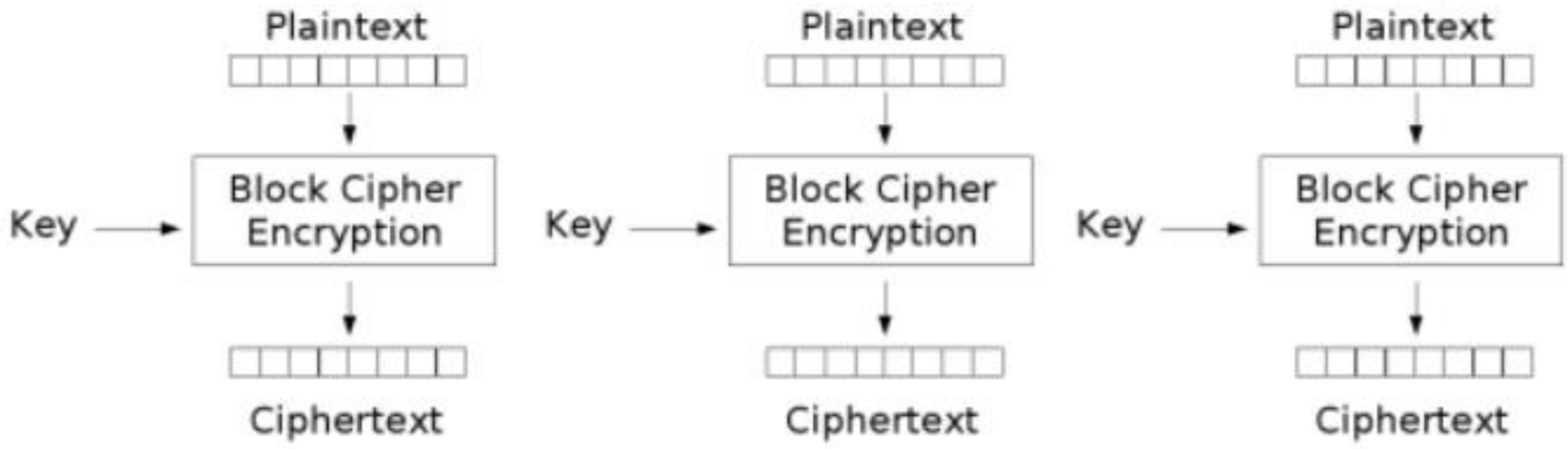
Symmetric Cryptosystem

- Symmetric Cryptosystem (對稱式密碼系統)
- Mode of Operation for Block Cipher (區塊密碼的工作模式)
  - ECB (Electronic Codebook Mode) (電子密碼本)
  - CBC (Cipher Block Chaining Mode) (密碼塊連結)
  - PCBC (Propagating Cipher-Block Chaining) (填充密碼塊連結)
  - CFB (Cipher Feedback Mode) (密文回饋)
  - OFB (Output Feedback Mode) (輸出回饋)

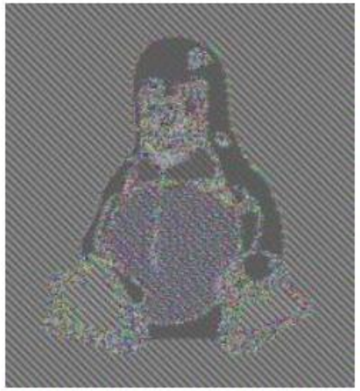
  - 參考：維基百科
    https://zh.wikipedia.org/wiki/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81%E5%B7%A5%E4%BD%9C%E6%A8%A1%E5%BC%8F

Symmetric Cryptosystem

- **ECB** **(Electronic Codebook Mode) (電子密碼本)**



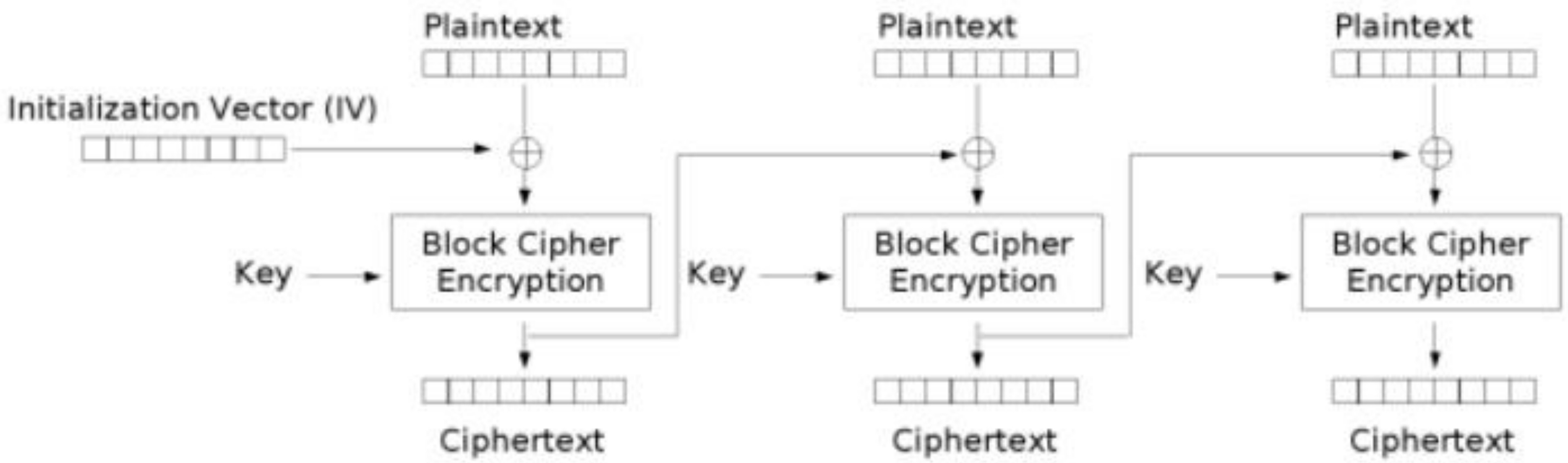
  - Drawback：same Plaintext get the same Ciphertext:



- https://zh.wikipedia.org/wiki/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81%E5%B7%A5%E4%BD%

Symmetric Cryptosystem

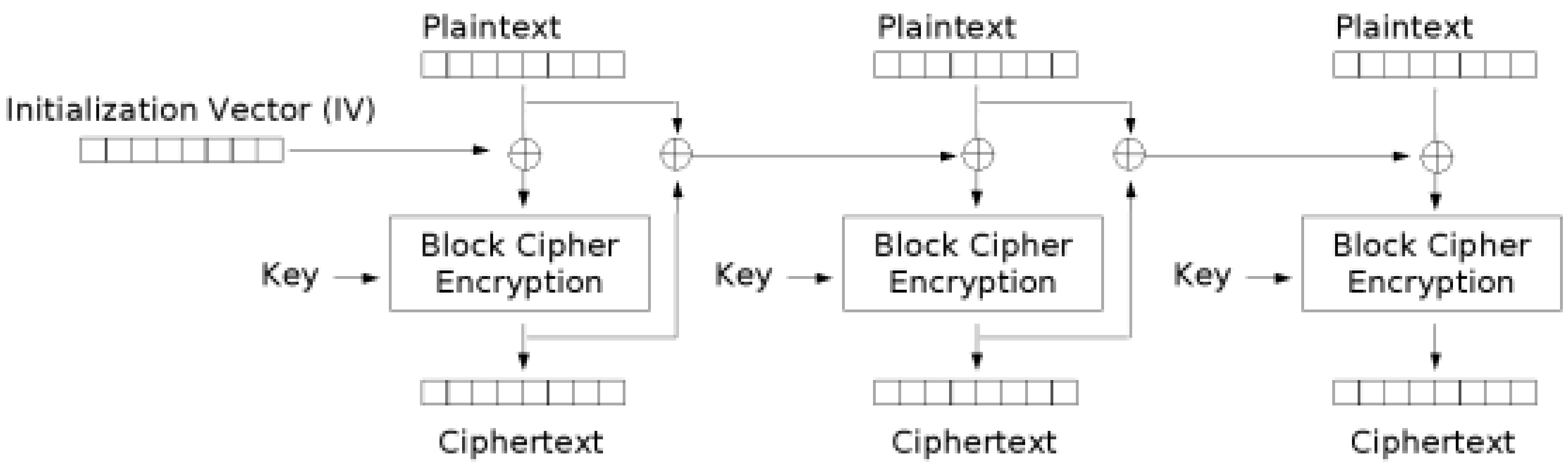
- CBC (Cipher Block Chaining Mode) (密碼塊連結)
  - IV: initialization Vector (起始向量)



- https://zh.wikipedia.org/wiki/%E5%88%86%E7%BB%84%E5%AF%86%E7%A0%81%E5%B7%A5%E4%BD%9C%E6%A8%A1%E5%BC%8F

Symmetric Cryptosystem

- **PCBC (Propagating Cipher-Block Chaining) (填充密碼塊連結)**

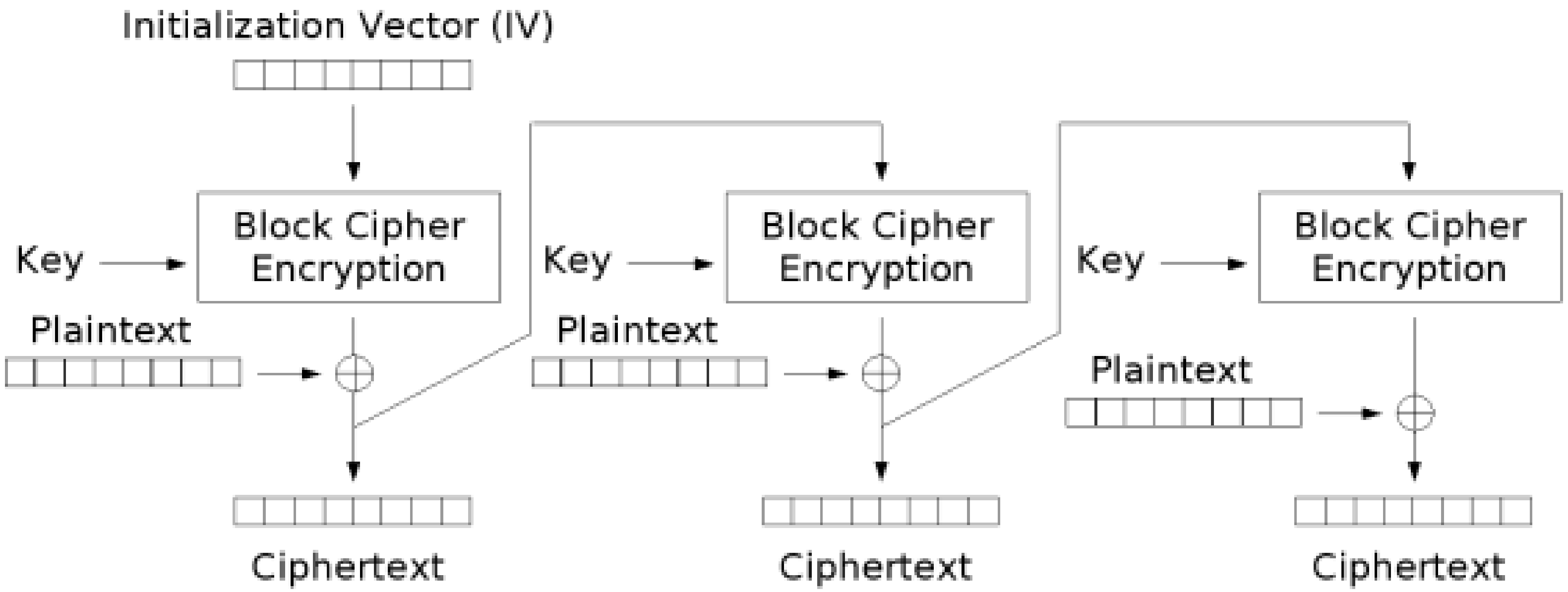


Propagating Cipher Block Chaining (PCBC) mode encryption

- **由 Loadmaster (David R. Tribble) - derived from File:Cbc encryption.png, 公有領域,** https://commons.wikimedia.org/w/index.php?curid=5715288

Symmetric Cryptosystem

- **CFB (Cipher Feedback Mode) (密文回饋)**

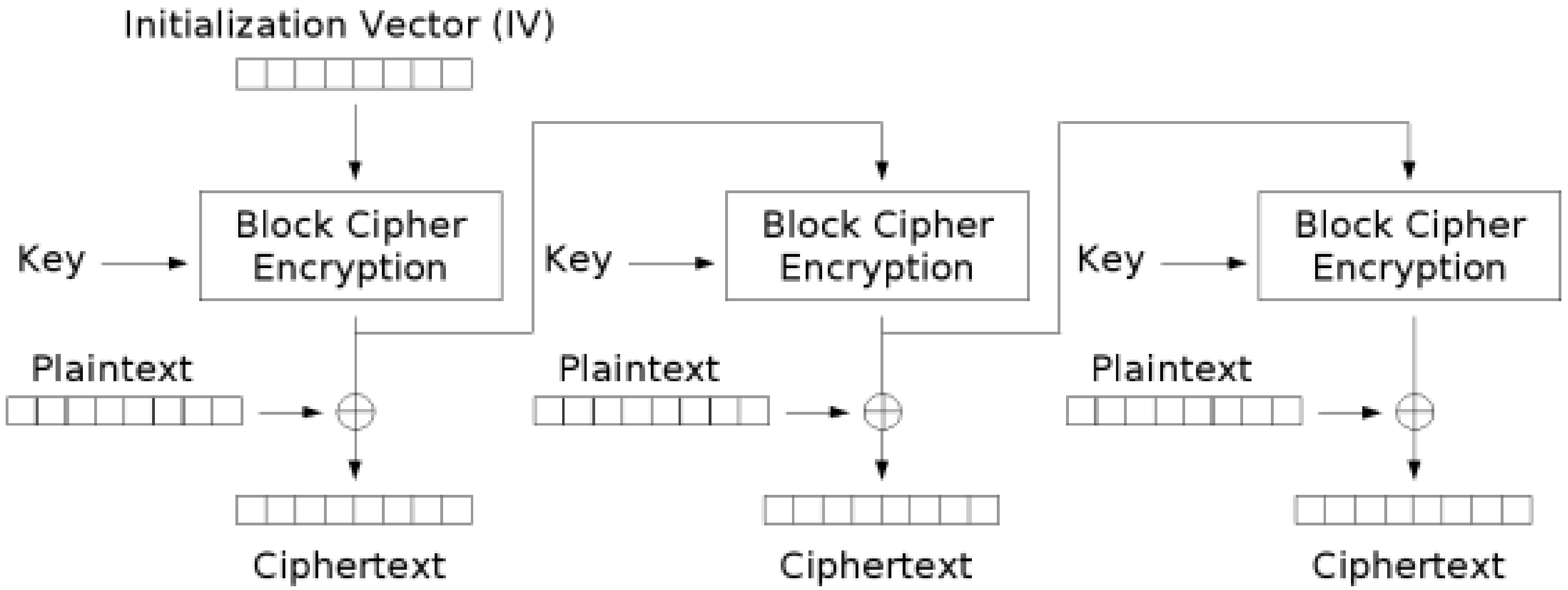


Cipher Feedback (CFB) mode encryption

- **由 Gwenda - English Wikipedia, 公有領域, https://commons.wikimedia.org/w/index.php?curid=1165191**

Symmetric Cryptosystem

- **OFB (Output Feedback Mode) (輸出回饋)**

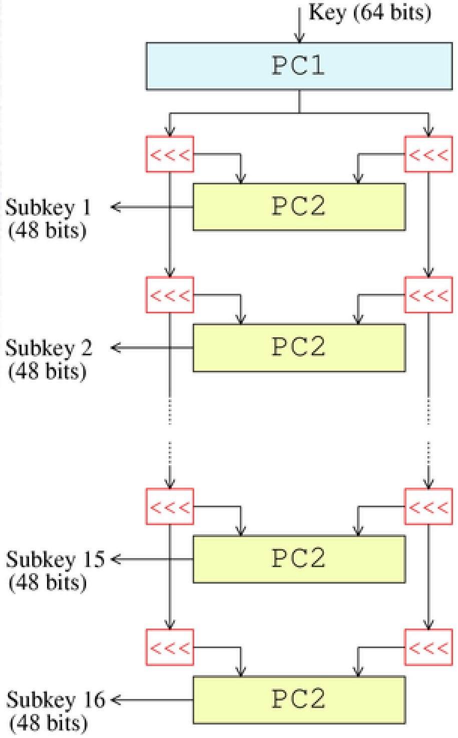


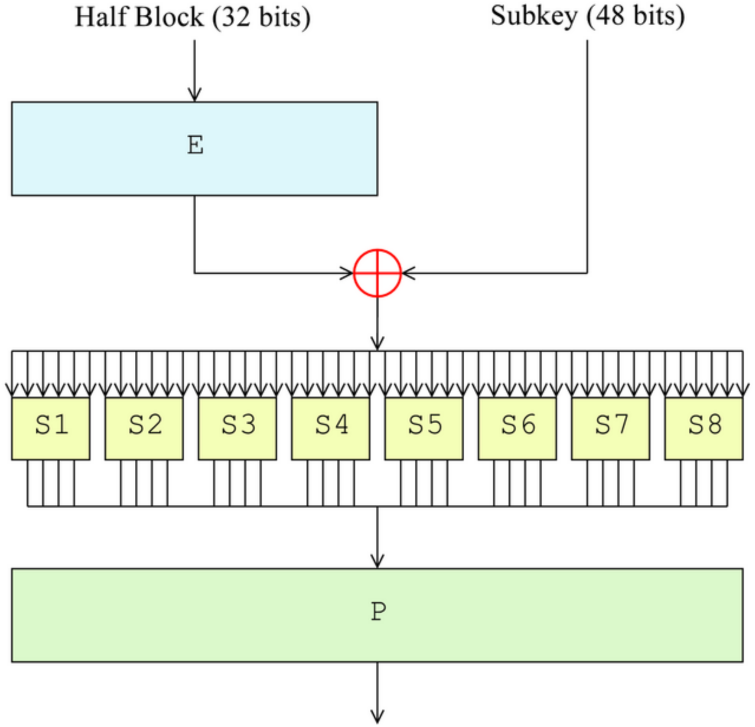
Output Feedback (OFB) mode encryption

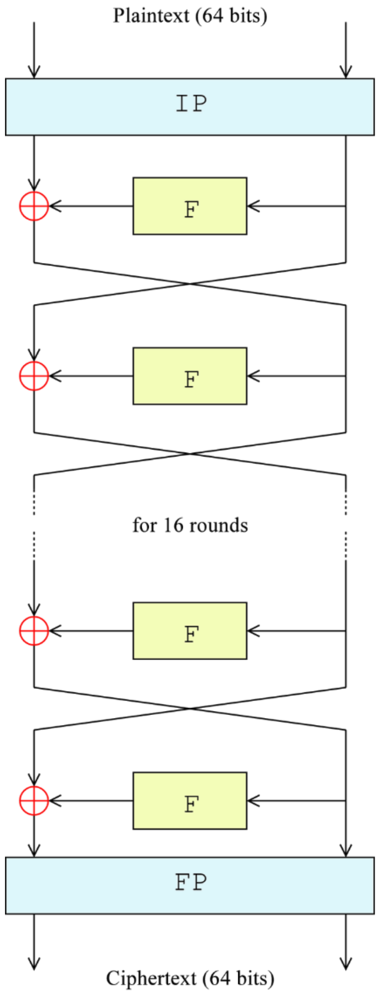
- **由 Gwenda - English Wikipedia, 公有領域, https://commons.wikimedia.org/w/index.php?curid=1165209**

Symmetric Cryptosystem

- **DES: 1970, Horst Feistel** (Lucifer, for IBM): using 56 bits key to encrypt 64 bits plaintext.→ Triple-DES (before 2030).
- Confusion (混淆)
- Diffusion (擴散)



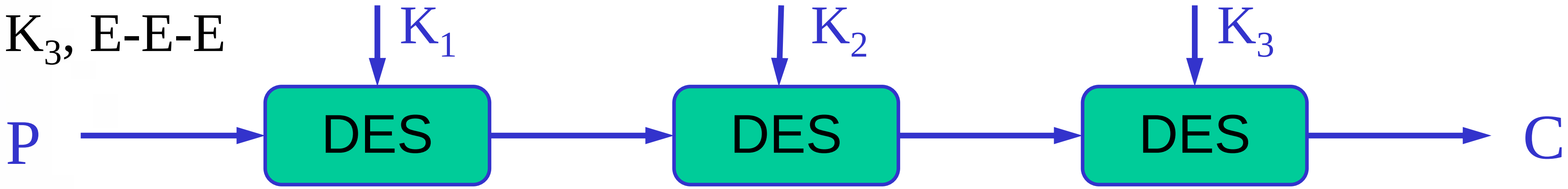




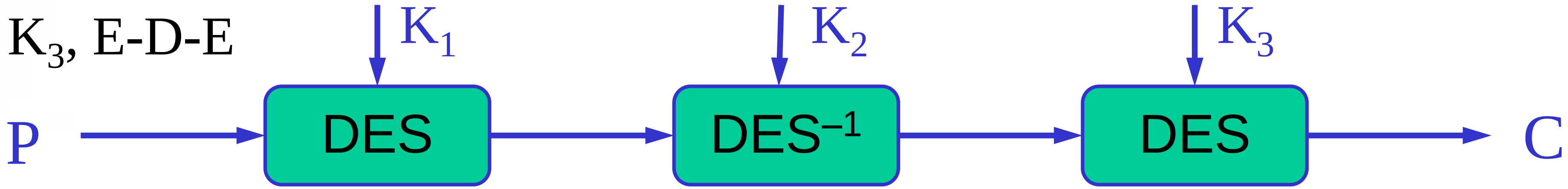
圖片來源：https://zh.wikipedia.org/wiki/%E8%B3%87%E6%96%99%E5%8A%A0%E5%AF%86%E6%A8%99%E6%BA%96

Symmetric Cryptosystem

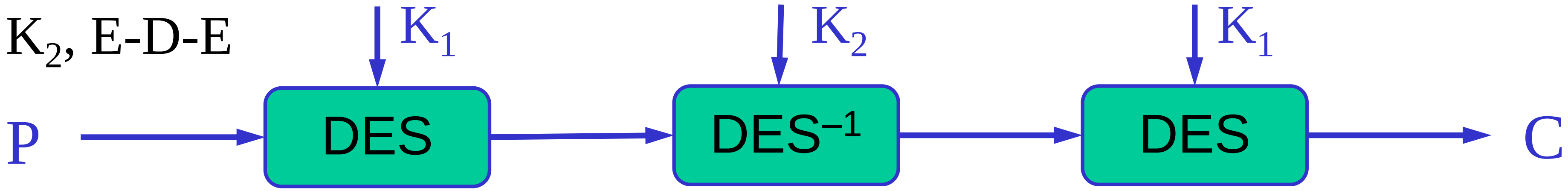
- **Triple-DES** (TDES, 3DES)
  - (a) $K_1 \neq K_2 \neq K_3$, E-E-E



  - (b) $K_1 \neq K_2 \neq K_3$, E-D-E



  - (c) $K_1 = K_3 \neq K_2$, E-D-E



Computer Science and Information Engineering
National Chi Nan University

The Principle and Application of Secret Sharing

Dr. Justie Su-Tzu Juan

Lecture 2. Fundamental and Technology of Cryptography

§ 2.4 Digital Signature

Slides for a Course Based on the Text
密碼學與網路安全
by 王旭正、柯宏叡

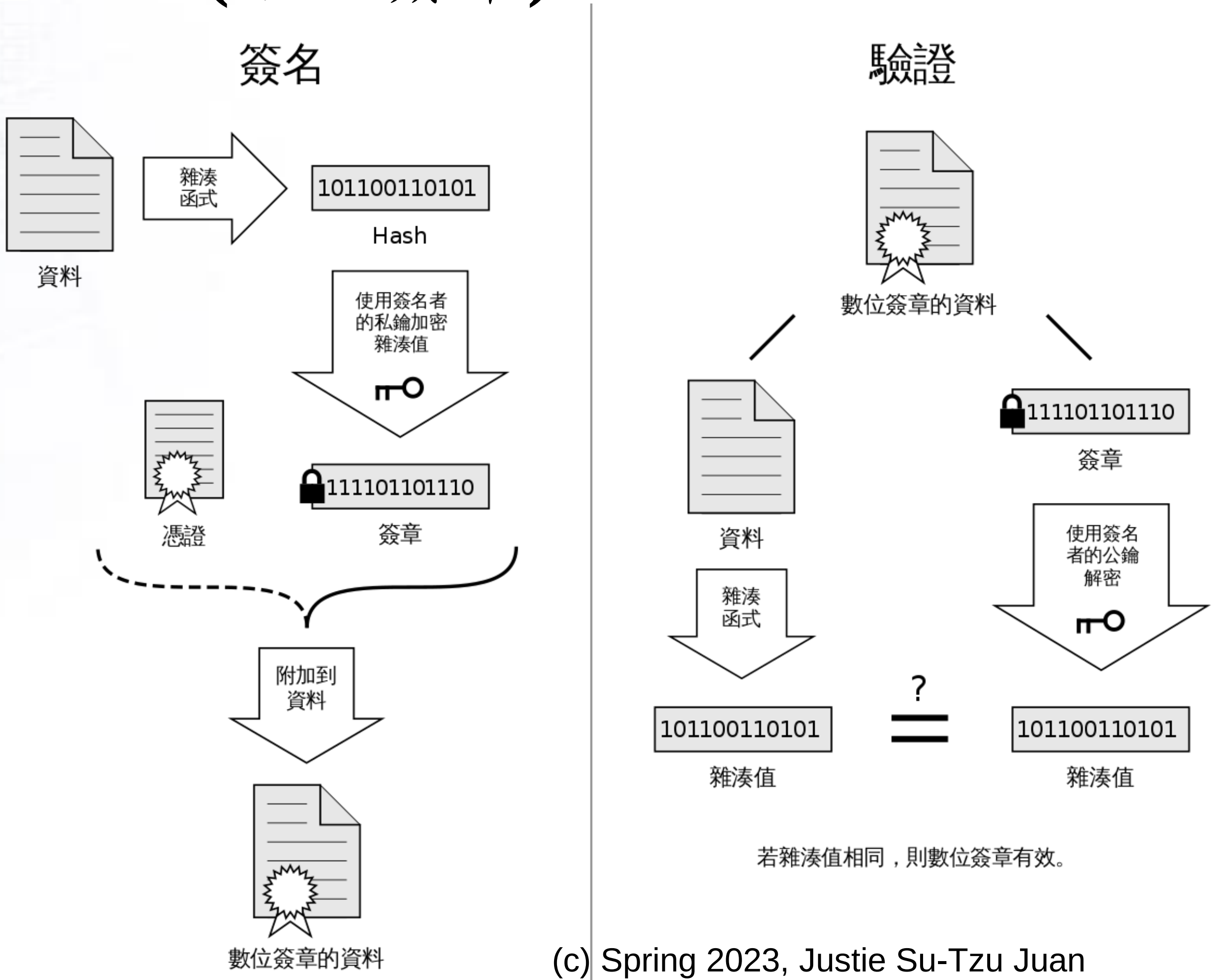
Digital Signature

- **Digital Signature (數位簽章):**
  - **Authentication**
  - **Integrity**
  - **Non-repudiation**

- **Def**: A **digital signature scheme** is a triple of probabilistic polynomial time algorithms, (G, S, V), satisfying:
  - G (key-generator) generates a public key (pk), and a corresponding private key (sk).
  - S (signing) returns a tag (t), on the inputs: the private key (sk), and a string (x).
  - V (verifying) outputs *accepted* or *rejected* on the inputs: the public key (pk), a string (x), and a tag (t).

Digital Signature

Digital Signature (數位簽章):



由 Acdx - https://en.wikipedia.org/wiki/File:Digital_Signature_diagram.svg, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=109180555

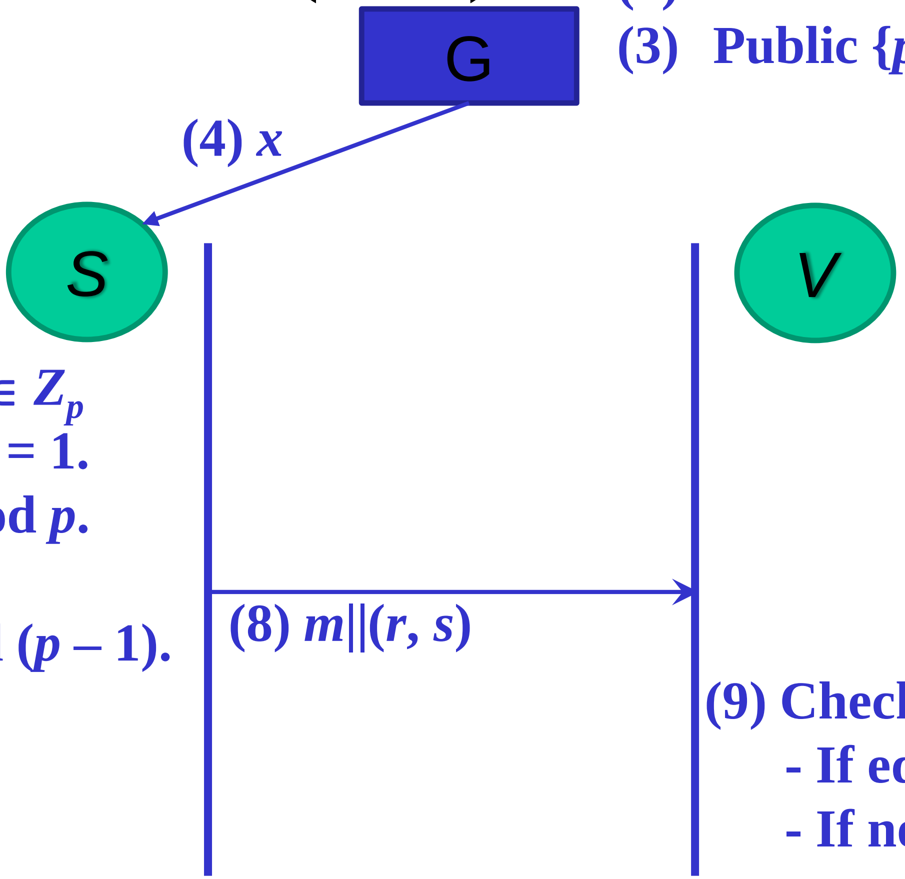
Digital Signature

- **ElGamal Signature Scheme: (1985)**

(1) Find big prime p and a generator g

(2) Choose $x \in [1, p - 1]$ and comp. $y = g^x \bmod p$

(3) Public $\{p, g, y\}$



(5) Pick random $k \in Z_p$ s.t. $\gcd(k, p - 1) = 1$.

(6) Comp. $r = g^k \bmod p$.

(7) Comp. s s.t.
$m = xr + ks \bmod (p - 1)$.

(9) Check $g^m =? y^r r^s \bmod p$
- **If equal, accept the signature (valid)**
- **If not equal, reject the signature (invalid)**

Digital Signature

- **Digital Signature Algorithm (DSA) : from DSS, SHA and Elgamal (1996)**

G

(1) Find big prime $p \in (2^{t-1}, 2^t)$, $64|t$ and $t \in [512, 1024]$.
(2) Find prime q, s.t. $q|p-1$, $q \in (2^{159}, 2^{160})$.
(3) Find $g = h^{(p-1)/q} \bmod p > 1$, commonly $h = 2$ is used.
(4) Choose $x \in [1, q-1]$ and comp. $y = g^x \bmod p$
(5) Public $\{p, q, g, y\}$

(4) x

S

V

(5) Pick random $k \in Z_q - \{0\}$.
(6) Comp. $r = (g^k \bmod p) \bmod q$.
(7) Comp. $s = k^{-1}[h(m) + xr] \bmod q$.

(8) $m||(r, s)$

(9) Comp. $\alpha = s^{-1} \bmod q$
$\beta = [(g^{\alpha h(m)})(y^{\alpha r}) \bmod p] \bmod q$
(10) Check $\beta =? r$
- If equal, accept the signature (valid)
- If not equal, reject the signature (invalid)

Computer Science and Information Engineering
National Chi Nan University

The Principle and Application of Secret Sharing

Dr. Justie Su-Tzu Juan

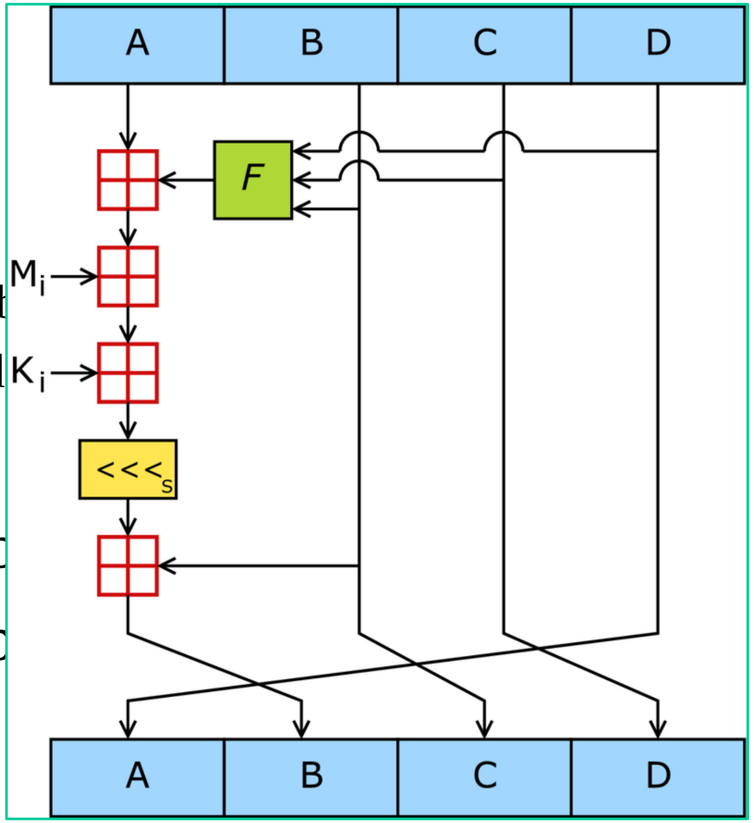
Lecture 2. Fundamental and Technology of Cryptography

§ 2.5 Hash Function

Slides for a Course Based on the Text
密碼學與網路安全
by 王旭正、柯宏叡

Hash Function

- **Hash Function (雜湊函數):** any function that can be used to map data of arbitrary size to fixed-size values. The output is the digest (訊息摘要) of the input.
  - **Message Digest Algorithm (MD5):** (Rivest, 1992) → MD6 (2009,2011)
    - Input: any length; output: 128 bits (16 bytes)
    - 由 Matt_Crypto (talk) (Uploads) - original illustration for Wikipedia, created in Dia., 公有領域, https://commons.wikimedia.org/w/index.php?curid=214963
    - Advantage:
      - MD5("The quick brown fox jumps over the lazy dog") = 9e107d9d372b
      - MD5("The quick brown fox jumps over the lazy dog.") = e4d909c290d
    - Disadvantage:
      - 1996: MD5 is proven weak and can be cracked (Dobb
      - 2011: an informational RFC 6151 was approved to up considerations in MD5 and HMAC-MD5.

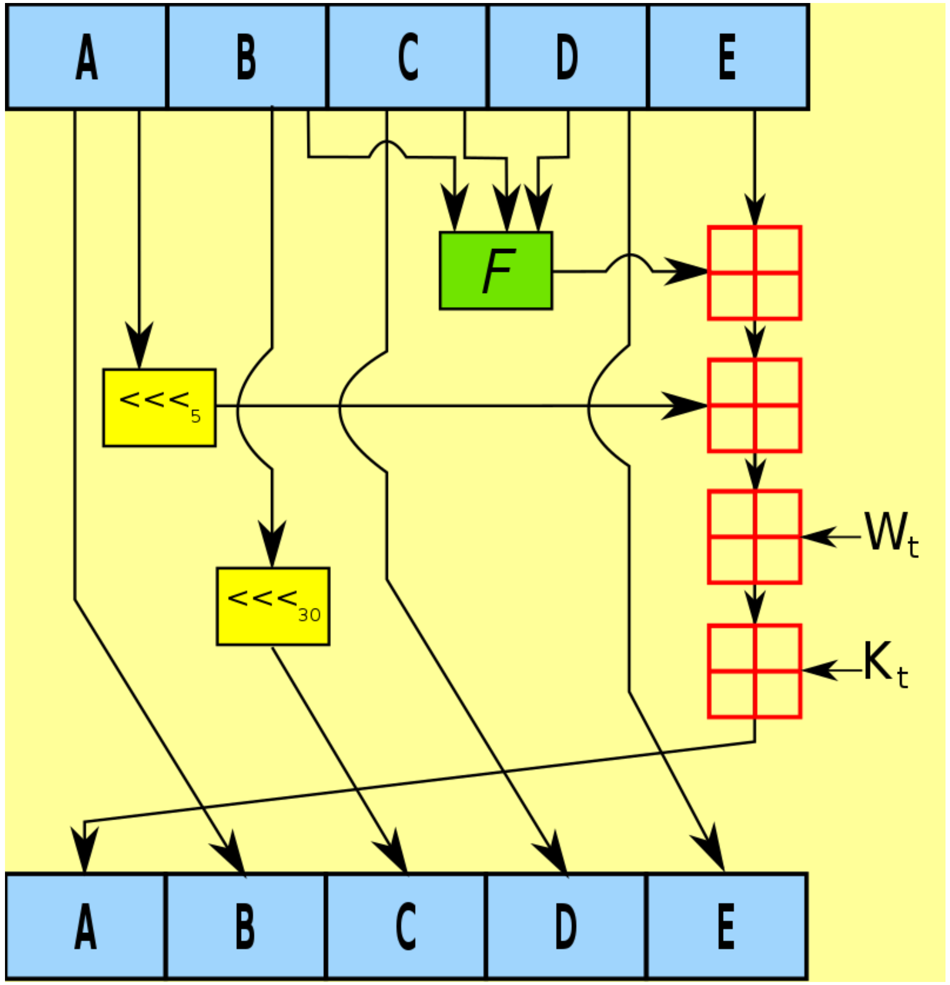


Hash Function

– **Secure Hash Algorithm (SHA)**: (NIST, 1993(SHA-0), 1995-2010(SHA-1))

- Input: any length; output: 160 bits (20 bytes)
-
- 2005: Effective Attack Method Discovered.
- 2020: Collision attacks are already practical.

  It is not recommended to use it anymore.



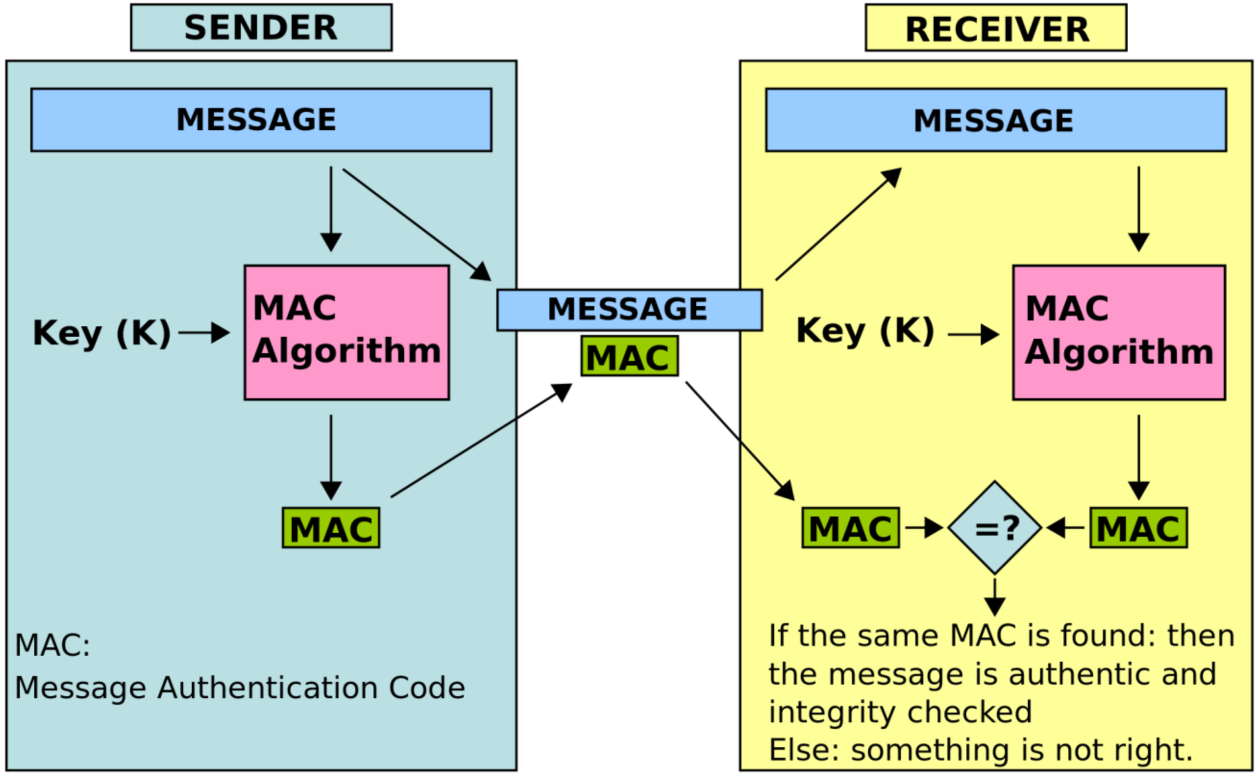
Hash Function

– **Secure Hash Algorithm (SHA)**: Comparison of SHA functions (https://en.wikipedia.org/wiki/Secure_Hash_Algorithms)

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Rounds	Operations	Security against collision attacks (bits)	Security against length extension attacks (bits)	Performance on Skylake (median cpb)[1]		First published
									Long messages	8 bytes	
MD5 (as reference)		128	128 (4 × 32)	512	4 (16 operations in each round)	And, Xor, Or, Rot, Add (mod 2^{32})	≤ 18 (collisions found)[2]	0	4.99	55.00	1992
SHA-0		160	160 (5 × 32)	512	80	And, Xor, Or, Rot, Add (mod 2^{32})	< 34 (collisions found)	0	≈ SHA-1	≈ SHA-1	1993
SHA-1							< 63 (collisions found)[3]		3.47	52.00	1995
SHA-2	*SHA-224* *SHA-256*	224 256	256 (8 × 32)	512	64	And, Xor, Or, Rot, Shr, Add (mod 2^{32})	112 128	32 0	7.62 7.63	84.50 85.25	2004 2001
	SHA-384	384	512 (8 × 64)	1024	80	And, Xor, Or, Rot, Shr, Add (mod 2^{64})	192	128 (≤ 384)	5.12	135.75	2001
	SHA-512	512					256	0[4]	5.06	135.50	2001
	SHA-512/224 *SHA-512/256*	224 256					112 128	288 256	≈ SHA-384	≈ SHA-384	2012
SHA-3	*SHA3-224* *SHA3-256* *SHA3-384* *SHA3-512*	224 256 384 512	1600 (5 × 5 × 64)	1152 1088 832 576	24[5]	And, Xor, Rot, Not	112 128 192 256	448 512 768 1024	8.12 8.59 11.06 15.88	154.25 155.50 164.00 164.00	2015
	SHAKE128 *SHAKE256*	*d* (arbitrary) *d* (arbitrary)		1344 1088			min(*d*/2, 128) min(*d*/2, 256)	256 512	7.08 8.59	155.25 155.50	

Hash Function

- **Message Authentication Code (訊息鑑定碼):** a short piece of information used for authenticating a message.



- 由 Twisp, based on diagram by w:User:Smilerpt - 本vector image使用Inkscape創作 ., 公有領域, https://commons.wikimedia.org/w/index.php?curid=3410890

Hash Function

- **Message Authentication Code (訊息鑑定碼):** a short piece of information used for authenticating a message.
  - **Hash-based Message Authentication Code (HMAC)**: (keyed-hash message authentication code)
  - Def: (from RFC 2104) By Gdrooid - Own work, CC0, https://com

$HMAC(K, m) = H((K' \oplus opad) \,||\, H((K' \oplus$

where H is a cryptographic hash function.

m is the message to be authenticated.

K is the secret key.

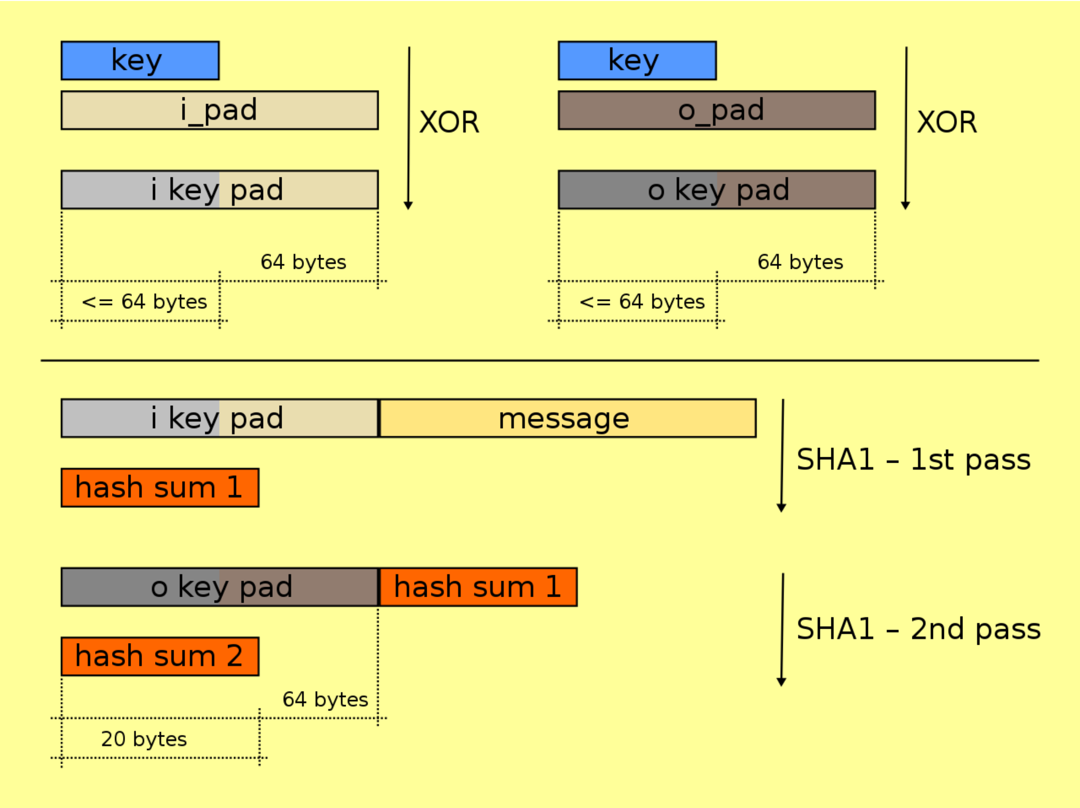
$K' = H(K)||00…0$, if K is larger than block

$||$ denotes concatenation.

$\oplus$ denotes bitwise exclusive or (XOR).

opad is the block-sized outer padding, con

ipad is the block-sized inner padding, cons



Computer Science and Information Engineering
National Chi Nan University

The Principle and Application of Secret Sharing

Dr. Justie Su-Tzu Juan

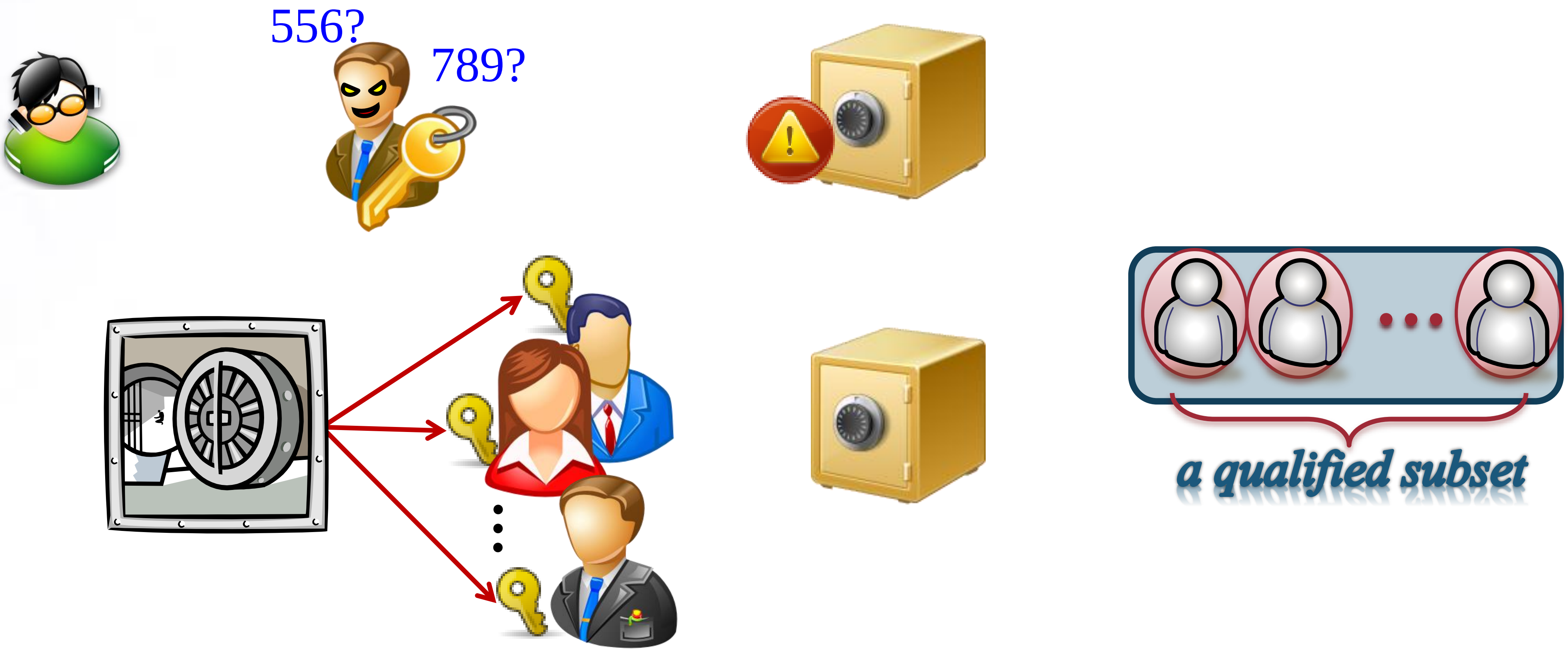
Lecture 2. Fundamental and Technology of Cryptography

§ 2.6 Secret Sharing Scheme

Slides for a Course Based on the Text
密碼學與網路安全
by 王旭正、柯宏叡

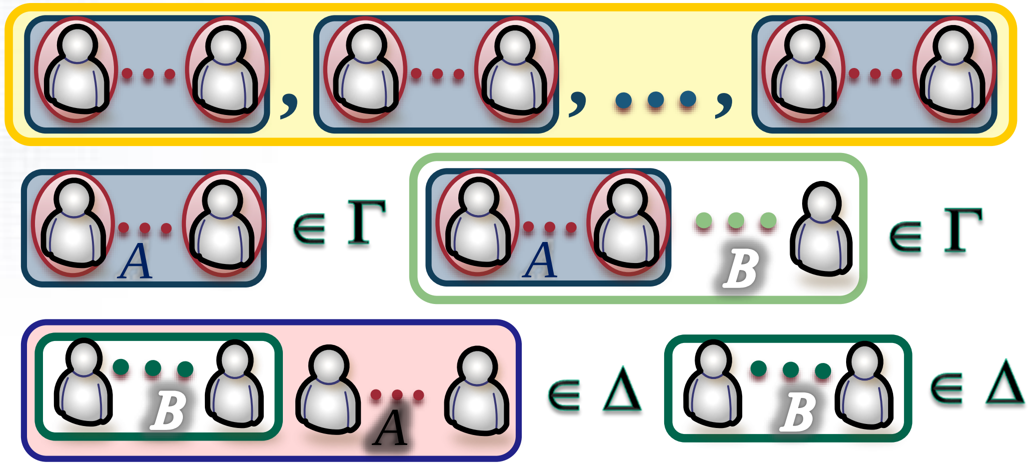
Secret Sharing Scheme

Secret Sharing Scheme (秘密分享機制): Shamir, 1979 (Blakley, 1979)

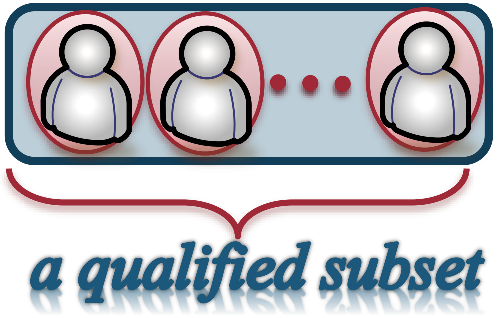


Secret Sharing Scheme

- (t, n)-threshold access structure
  - $\geq t$ participants in any qualified subset
- General access structure
  - Access structure Γ — ***monotone increasing property***
  - Prohibited structure Δ — ***monotone decreasing property***
- Γ =

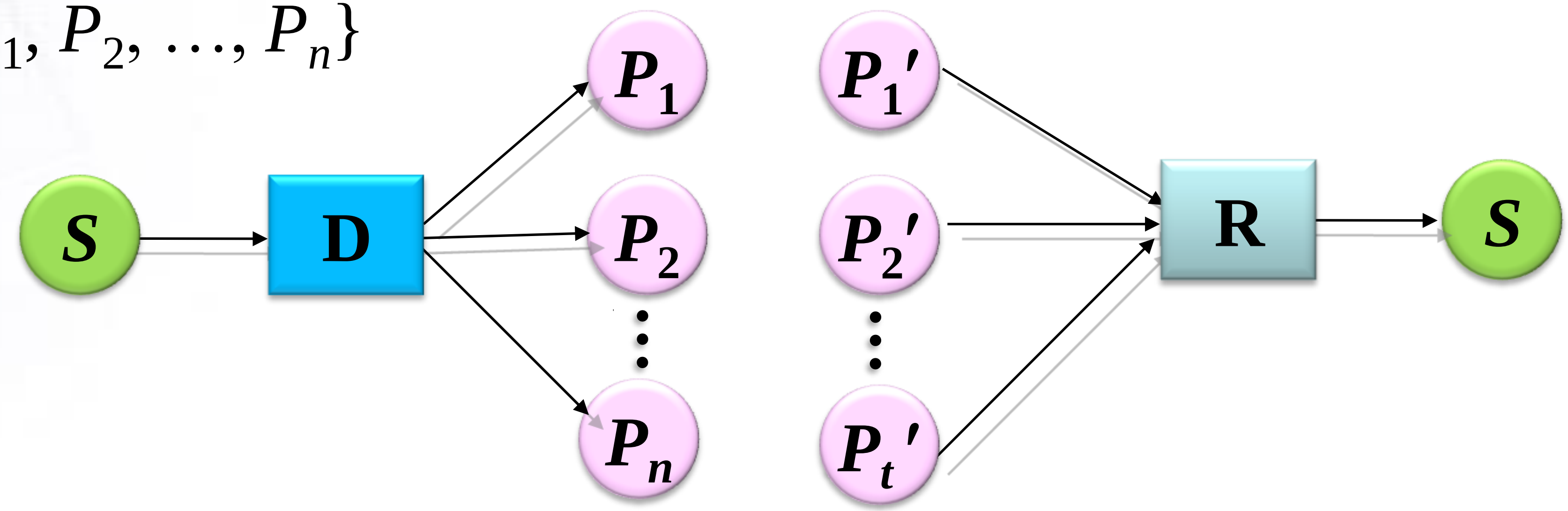


- minimal access structure Γ′



Secret Sharing Scheme

- Dealer
- Participants
  - $P = \{P_1, P_2, \ldots, P_n\}$



D : Distribution Algorithm

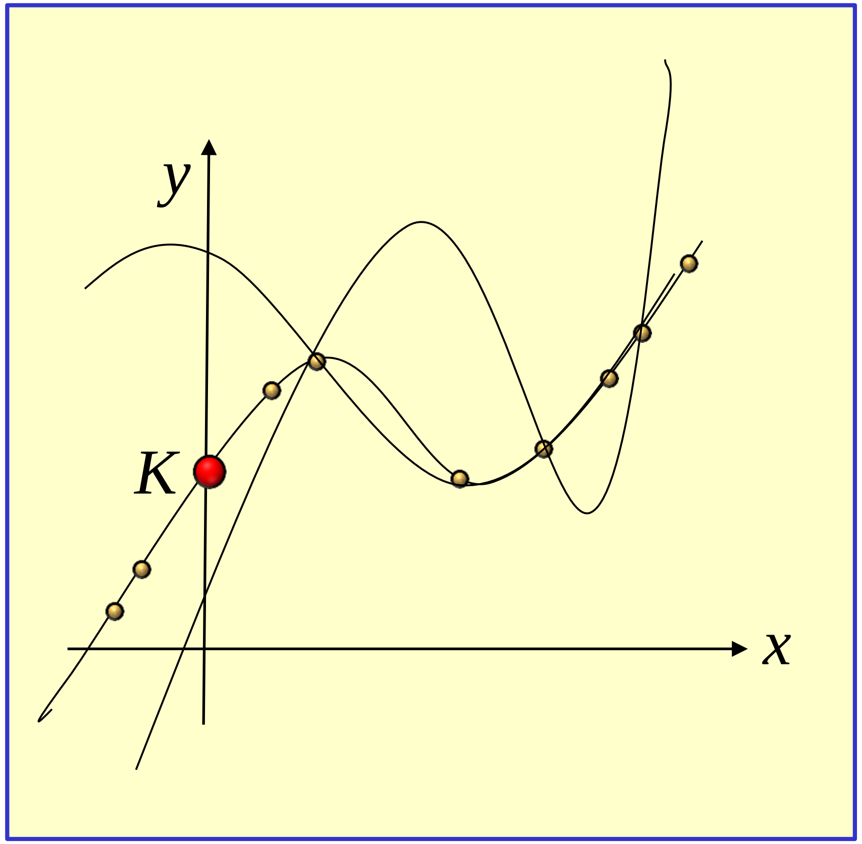
R : Reconstruction Algorithm

Secret Sharing Scheme

- **Shamir's (t, n)-threshold Secret Sharing Scheme:** 1979 (Perfect Secrecy)
  - **Distribution Algorithm**
    - $f(x) = K + a_1x + a_2x^2 + \ldots + a_{t-1}x^{t-1} \pmod{q}$
    - Send $K_i = f(x_i)$ to P_i; x_i is the ID of P_i.
  - **Reconstruction Algorithm**
    - Collect t pairs of $(x_i, f(x_i))$ to recover $f(x)$
    - Using Lagrange Interpolation Formula:

$$f(x) = \sum_{i=1}^{t} f(x_i) \prod_{j=1, j \neq i}^{t} \frac{x - x_i}{x_i - x_j} \bmod p$$

    - Get $K = f(0)$



Secret Sharing Scheme

- **Shamir's (t, n)-threshold Secret Sharing Scheme:** 1979

  Ex: K = 13, $(t, n) = (3, 5)$, $p = 17$, $f(x) = 13 + 10x + 2x^2 \pmod{17}$, and ID$i$ = i:

  – **Distribution:**

  - $K_1 = f(1) = 8$; $K_2 = f(2) = 7$; $K_3 = f(3) = 10$; $K_4 = f(4) = 0$; $K_5 = f(5) = 11$.

  – **Reconstruction:**

  - Collect 3 pairs of (i, K_i), for example, $(1, K_1)$, $(3, K_3)$, $(5, K_5)$ to recover $f(x)$ as

$$f(x) = [8\frac{(x-3(x-5)}{(1-3)(1-5)} + 10\frac{(x-1)(x-5)}{(3-1)(3-5)} + 11\frac{(x-1)(x-3)}{(5-1)(5-3)}] \bmod 17$$

$$= [8 \cdot (8^{-1})(x^2 - 8x + 15) + 10 \cdot (-4^{-1})(x^2 - 6x + 5) + 11 \cdot (8^{-1})(x^2 - 4x + 3)] \bmod 17$$

$$= [8 \cdot 15 \cdot (x^2 - 8x + 15) + 10 \cdot 4 \cdot (x^2 - 6x + 5) + 11 \cdot 15 \cdot (x^2 - 4x + 3)] \bmod 17$$

$$= [x^2 - 8x + 15 + 6(x^2 - 6x + 5) + 12(x^2 - 4x + 3)] \bmod 17$$

$$= [2x^2 + 10x + 13] \bmod 17$$

  - $K = f(0) = 13$

Secret Sharing Scheme

- **Harn's Generalized Secret Sharing Scheme:** 1994 (based on Shamir's SSS)
  - **Distribution Algorithm**

    Input: prime p, K ($< p$), $\Gamma = \{A_1, A_2, \ldots, A_{|\Gamma|}\}$, where $A_i = \{P_{i,1}, P_{i,2}, \ldots, P_{i,|A_i|}\}$, $P_{i,j} \in P$

    1. Randomly select $r_i < p$ for P_i with ID_i.
    2. For any A_i: using Lagrange's formula to form $f_{A_i}(x)$ by (ID_j, r_j) for all $j \in A_i$

       calculate $V_{A_i} = K - f_{A_i}(0) \bmod p$
    3. Send r_i to P_i; Public $\{V_{A_i} \mid A_i \in \Gamma\}$.
  - **Reconstruction Algorithm**

    Input: (ID_j, r_j) for all $j \in A_i$ for some $A_i \in \Gamma$

    1. Using Lagrange Interpolation Formula to get $f_{A_i}(x)$
    2. Get $K = V_{A_i} + f_{A_i}(0) \bmod p$

Public-Key Cryptosystem - RSA

- **Programming Homework #2**: (4/18) Implement **Harm's Generalized Secret Sharing Scheme**.
- 1. $p > 10^9$
- 2. Using your student's ID as IDi
- 3. $|A_i|$ may = 11
- 4. $10^5 < K < p$