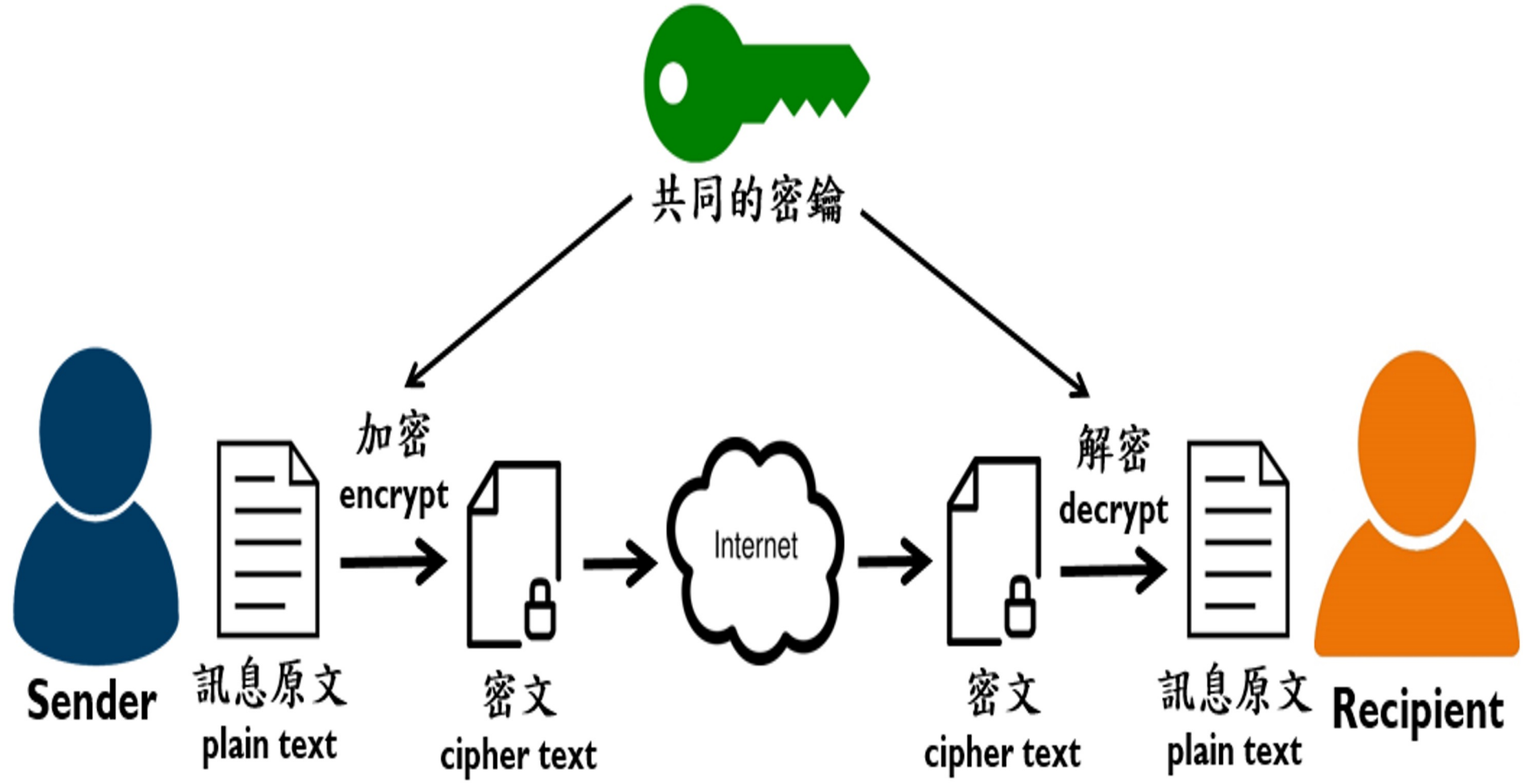
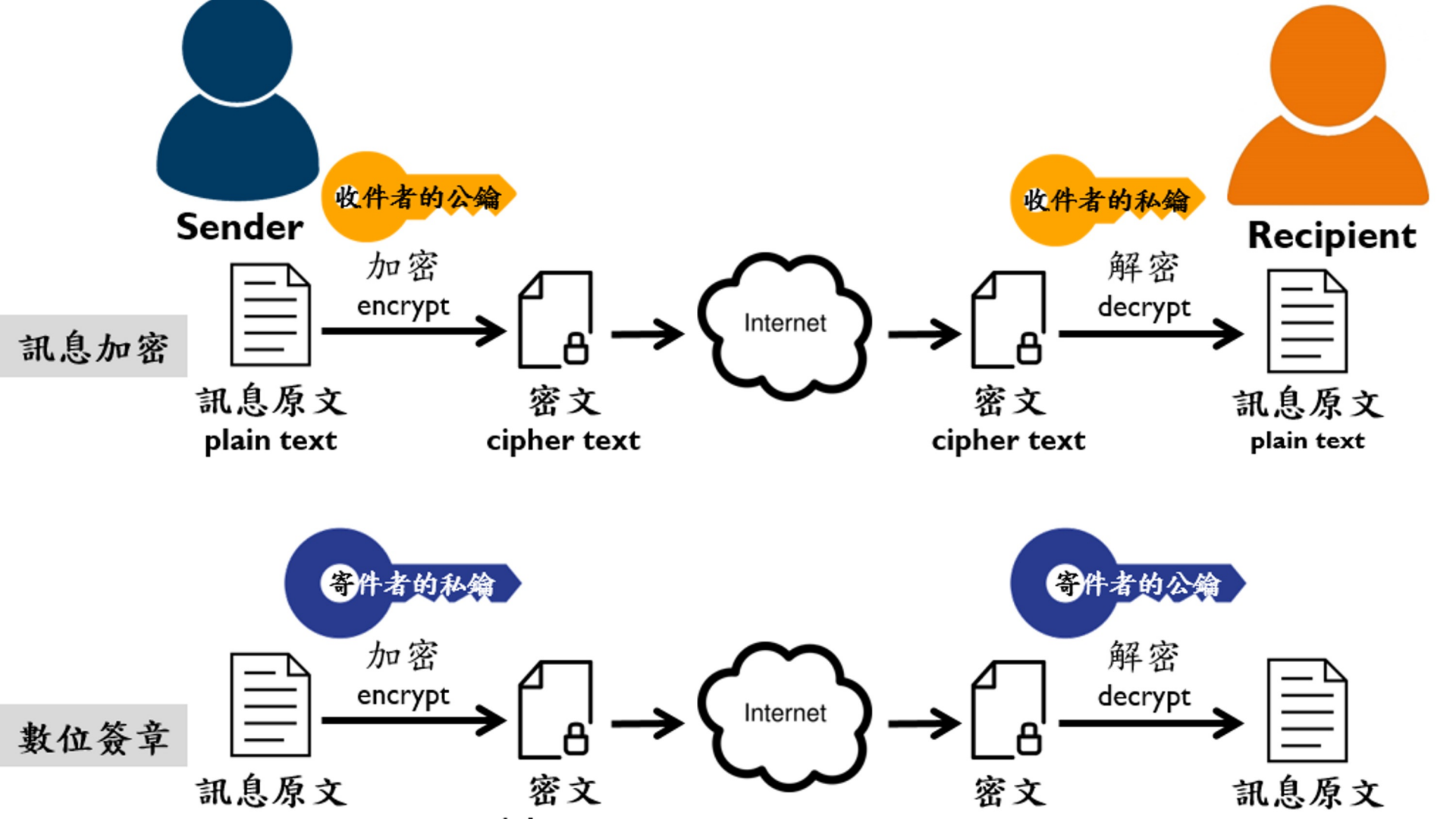
# 接觸史文件

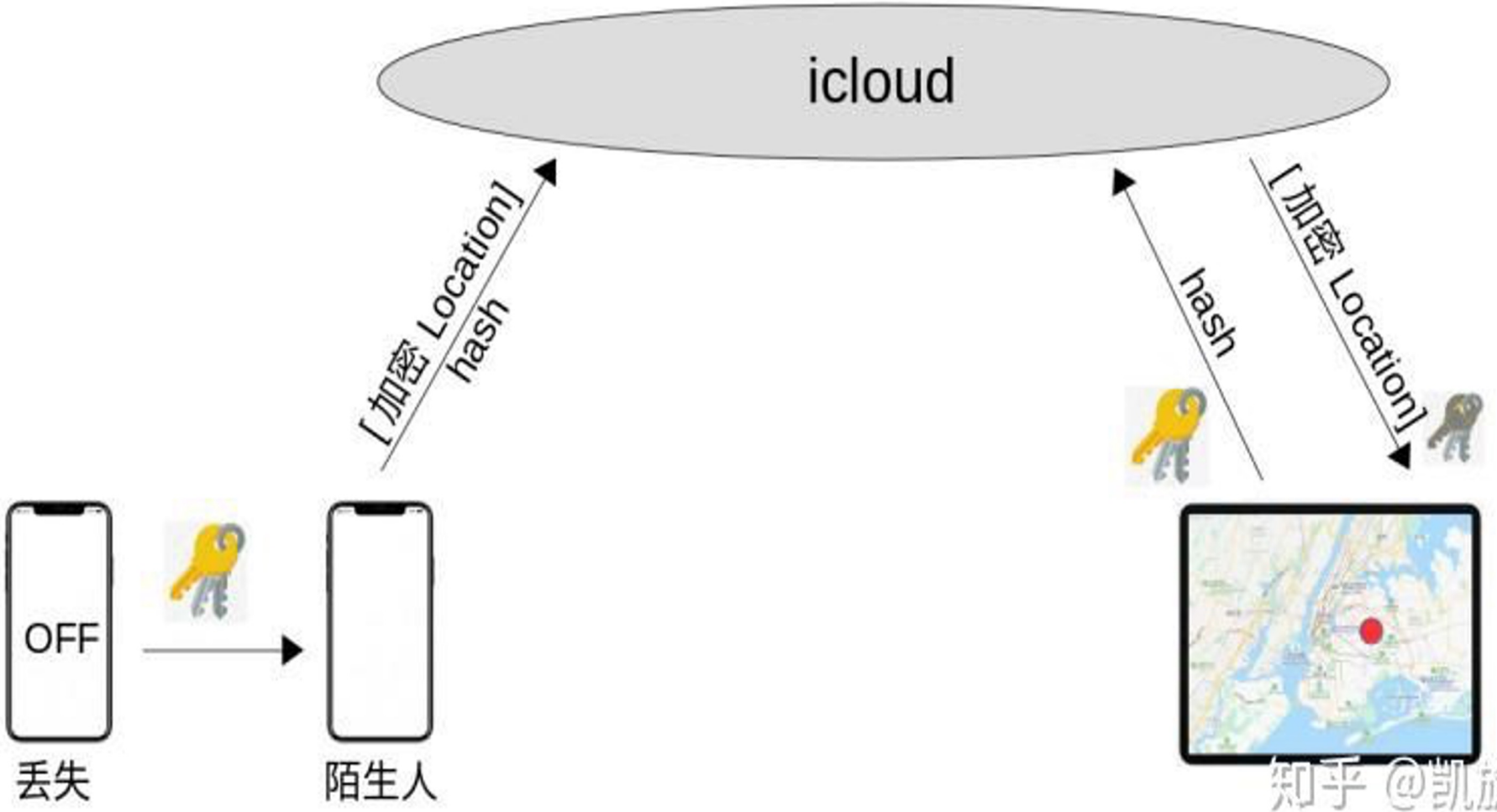
# 對稱加密


共同的密鑰
加密
encrypt
解密
decrypt
Internet
Sender
訊息原文
plain text
密文
cipher text
密文
cipher text
訊息原文
plain text
Recipient

# 非對稱加密


Sender
收件者的公鑰
加密
encrypt
Internet
收件者的私鑰
解密
decrypt
Recipient
訊息加密
訊息原文
plain text
密文
cipher text
密文
cipher text
訊息原文
plain text
寄件者的私鑰
加密
encrypt
寄件者的公鑰
解密
decrypt
數位簽章
訊息原文
密文
密文
訊息原文

# Apple -
# Find My Phone


icloud
[加密 Location]
hash
hash
[加密 Location]
OFF
丢失
陌生人
知乎 @凯旋

# 回到正題...

# TK (Tracing Key)

## TK (Tracing Key)

1. 每個手機只生成一次，生成後固定不變

1. 32字節長度，通過算法保證每個用戶唯一

1. 該密鑰由操作系統組件生成

$$tk \leftarrow CRNG(32)$$

# DTK(Daily Tracing Key)

## DTK (Daily Tracing Key)

1. 每天變化一次
2. 16字節長度

$$dtk_i \leftarrow HKDF(tk, NULL, (\text{UTF8("CT-DTK")} \,||\, D_i), 16)$$

## Di (DayNumber)

從1970年1月1日到當前的天數，其算法如下:

$$\text{DayNumber} \leftarrow \frac{\text{Number of Seconds since Epoch}}{60 \times 60 \times 24}$$

# RPI
# (Rolling Proximity Identifier)

## RPI (Rolling Proximity Identifier)

1. 每10分鐘變化一次
2. 為藍牙向外廣播的內容資料

$$RPI_{i,j} \leftarrow \text{Truncate}(HMAC(dkt_i, (\text{UTF8("CT-RPI")} \,||\, TIN_j)),16)$$

## TIN (TimeIntervalNumber)

1. 從每一天的0點開始初始化為0
2. 每過10分鐘累加1
3. 取值範圍[0-143]

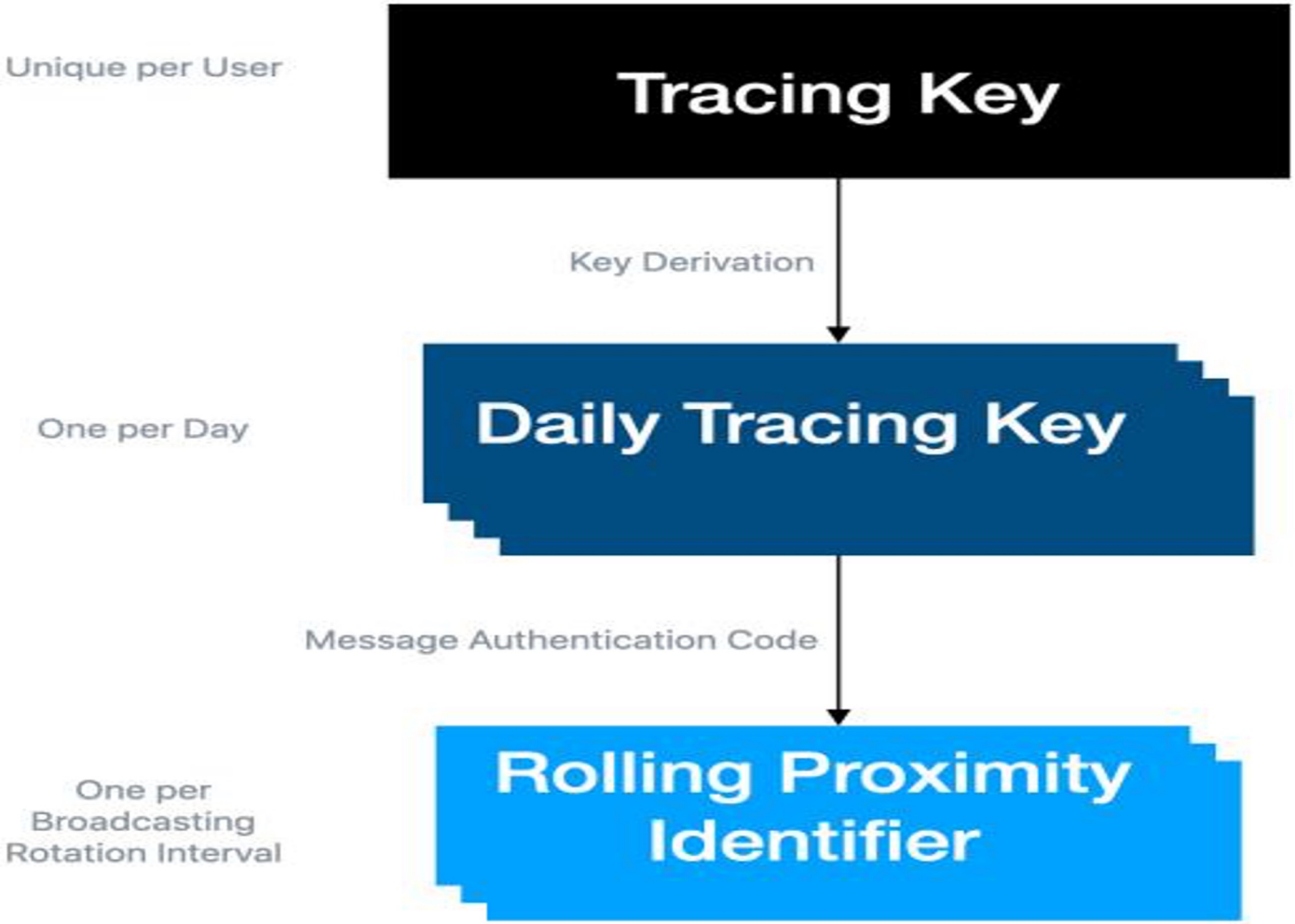
$$\text{TimeNumberInterval} \leftarrow \frac{\text{Seconds Since Start of DayNumber}}{60 \times 10}$$

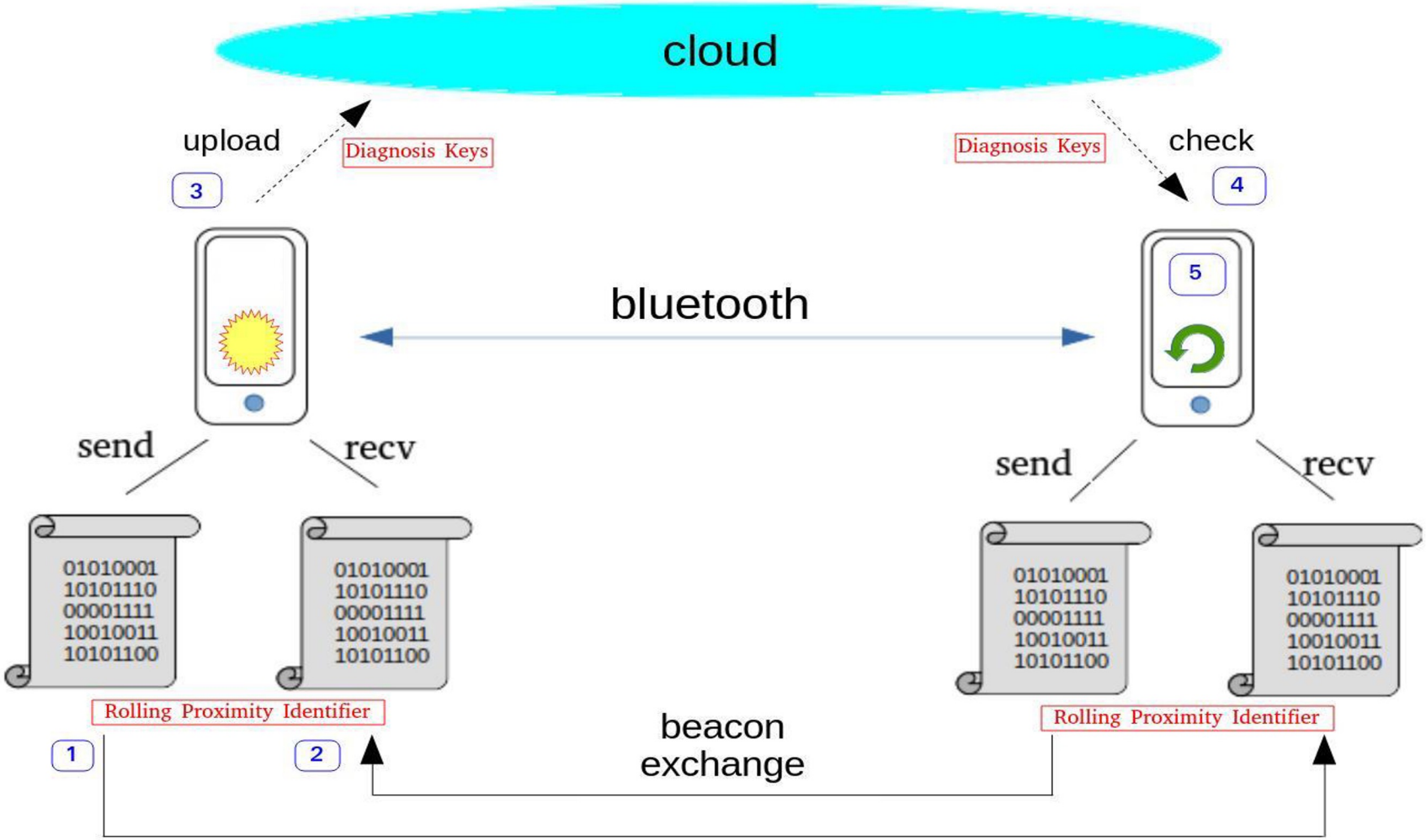
# DK (Diagnosis Keys)

# DK (Diagnosis Keys)

1. 一組每日追蹤密鑰的集合

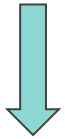
1. [Daily Tracing Key, 產生日期]，一般是14組。


Unique per User
Tracing Key
Key Derivation
One per Day
Daily Tracing Key
Message Authentication Code
One per Broadcasting Rotation Interval
Rolling Proximity Identifier

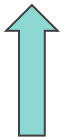

cloud
upload
Diagnosis Keys
3
Diagnosis Keys
check
4
5
bluetooth
send
recv
send
recv
01010001
10101110
00001111
10010011
10101100
01010001
10101110
00001111
10010011
10101100
01010001
10101110
00001111
10010011
10101100
01010001
10101110
00001111
10010011
10101100
Rolling Proximity Identifier
Rolling Proximity Identifier
1
2
beacon
exchange

0x03

透過ServiceUUID告知此設備所支持的Service



| Flags | | | Complete 16-bit ServiceUUID | | | Service Data - 16 bit UUID | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Length** | **Type** | **Flags** | **Length** | **Type** | **ServiceUUID** | **Length** | **Type** | **ServiceData** | |
| 0x02 | 0x01 (Flag) | 0x1A | 0x03 | 0x03<br><br>(Complete 16-bit ServiceUUID) | 0xFD6F<br><br>(Contact Detection Service) | 0x13 | 0x16<br><br>(Service Data - 16 bit UUID) | 0xFD6F<br><br>(Contact Detection Service) | **16 bytes**<br><br>**Rolling Proximity Identifier** |



0x16

前兩個byte 為 Service UUID

後面為自定義資料

# Advertising Packet : Eddystone Beacon

| 1 byte | 4 bytes | ≤ 39 bytes | 3 bytes |
|---|---|---|---|
| Preamble | Access Address | Protocol Data Unit (PDU) | CRC |
| 0xAA | 0x8E89BED6 | | |

| | 2 bytes | ≤ 37 bytes |
|---|---|---|
| Advertising Channel PDU | Header | Advertising payload |

| 4 bits | 2 bits | 1 bit | 1 bit | 6 bits | 2 bits |
|---|---|---|---|---|---|
| PDU type | RFU | TxAdd | RxAdd | Length | RFU |

| 6 bytes | ≤ 31 bytes | | |
|---|---|---|---|
| ADV Address | Advertising Data | | |
| | AD0 | AD1 | AD2 |

| 1 byte | 1 byte | 1 byte |
|---|---|---|
| AD Length | AD Type | Flags |
| 0x02 | 0x01 | 0x06 |

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| AD Length | AD Type | Eddystone UUID |
| 0x03 | 0x03 | 0xFEAA |

| 1 byte | 1 byte | 2 bytes | ≤ 20 bytes |
|---|---|---|---|
| AD Length | AD Type | Eddystone UUID | Eddystone Frame |
| | 0x16 | 0xFEAA | |

- UID frame
- URL frame
- TLM frame

```
private List<ScanFilter> generateScanFilters() {
    List<ScanFilter> filters = new ArrayList<>();
    ScanFilter filter = new ScanFilter.Builder()
            .setServiceUuid(ParcelUuid.fromString("0xFD6F"))
            .build();
    filters.add(filter);
    return filters;
}
```

# HMAC

# HMAC (Keyed-hash message authentication code)

1. 避免使用同樣數據總是得到相同Hash 值

1. 額外加入一個Key

1. 使用不同的key 就能得出不同的Hash值

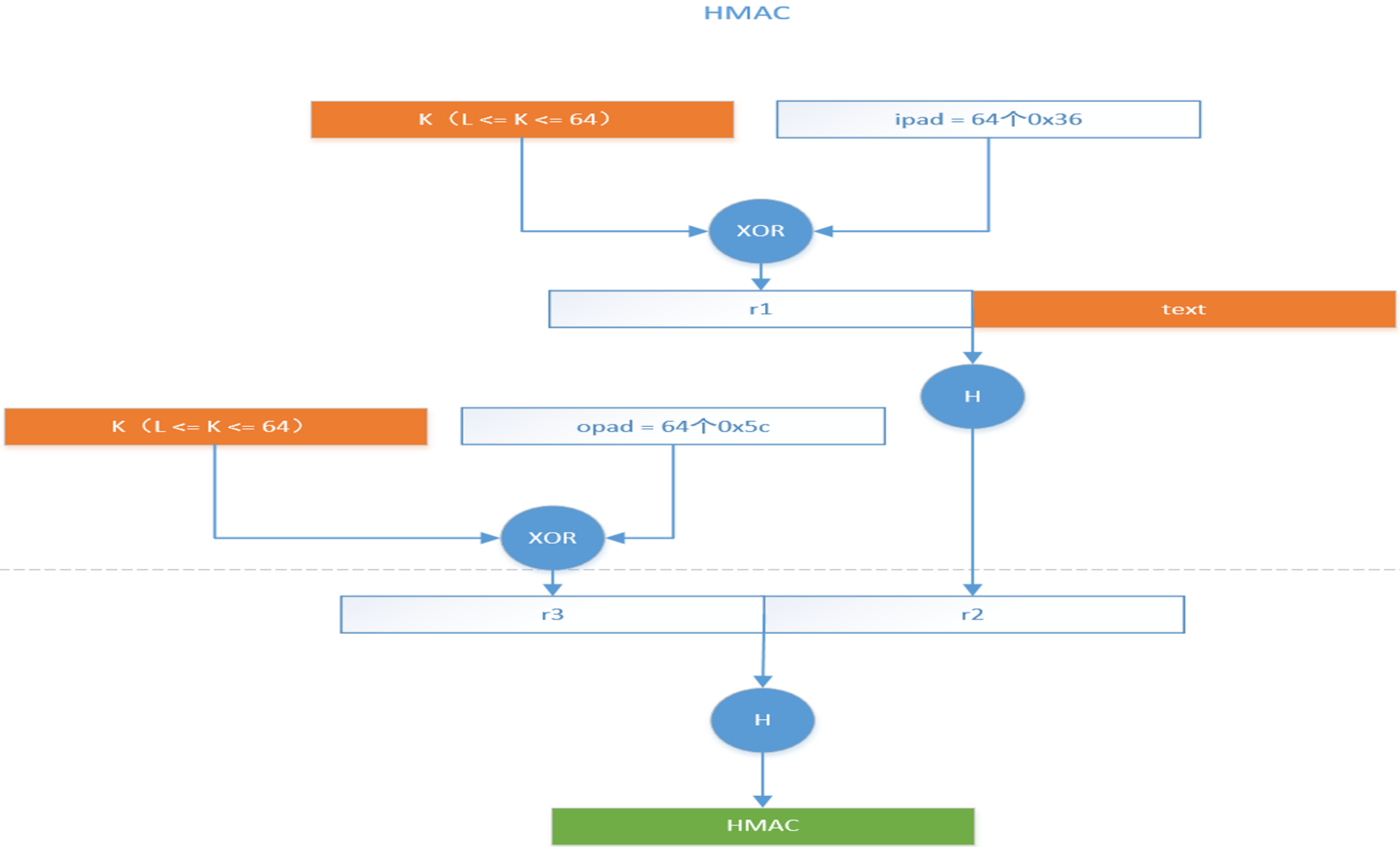
$$\text{Output} \leftarrow HMAC(\text{Key, Data})$$

$$\mathrm{HMAC}(K, m) = \mathrm{H}\left( (K' \oplus opad) \parallel \mathrm{H}\left( (K' \oplus ipad) \parallel m \right) \right)$$

$$K' = \begin{cases} \mathrm{H}(K) & K \text{ is larger than block size} \\ K & \text{otherwise} \end{cases}$$

ipad -> 64個0x36

opad-> 64個0x5c


HMAC
K （L <= K <= 64）
ipad = 64个0x36
XOR
r1
text
H
K （L <= K <= 64）
opad = 64个0x5c
XOR
r3
r2
H
HMAC

# HKDF

# HKDF (HMAC-based Key Derivation Function)

1. 將較短的key擴展成較長的key

1. 保證隨機性

1. 有以下兩個步驟
    (a) HKDF-Extract

    (a) HKDF-Expand

Output $\leftarrow HKDF$(Key, Salt, Info, OutputLength)

# Salt

```php
<?php
function hash($a) {
    $salt="WIKIPEDIA"; //定義一個加鹽字串(WIKIPEDIA)
    $b=$a.$salt; //把密碼與加鹽結合
    $b=sha($b);
    return $b;
}
?>
```

# HKDF-Extract

HKDF-Extract(salt, IKM) => PRK

1. SHA-256 => Length = 32 bytes

1. 如果沒有salt，默認是32個的0

1. salt => key , IKM => original key => data

   HKDF- Extract (H, salt, IKM)

   HMAC- Hash (H, salt, IKM)

1. 輸出結果結果為 PRK

# HKDF-Expand

HKDF-Expand(PRK, info, L) => OKM

1. SHA-256 => Length = 32 bytes

1. HKDF-Extract 所生成的PRK

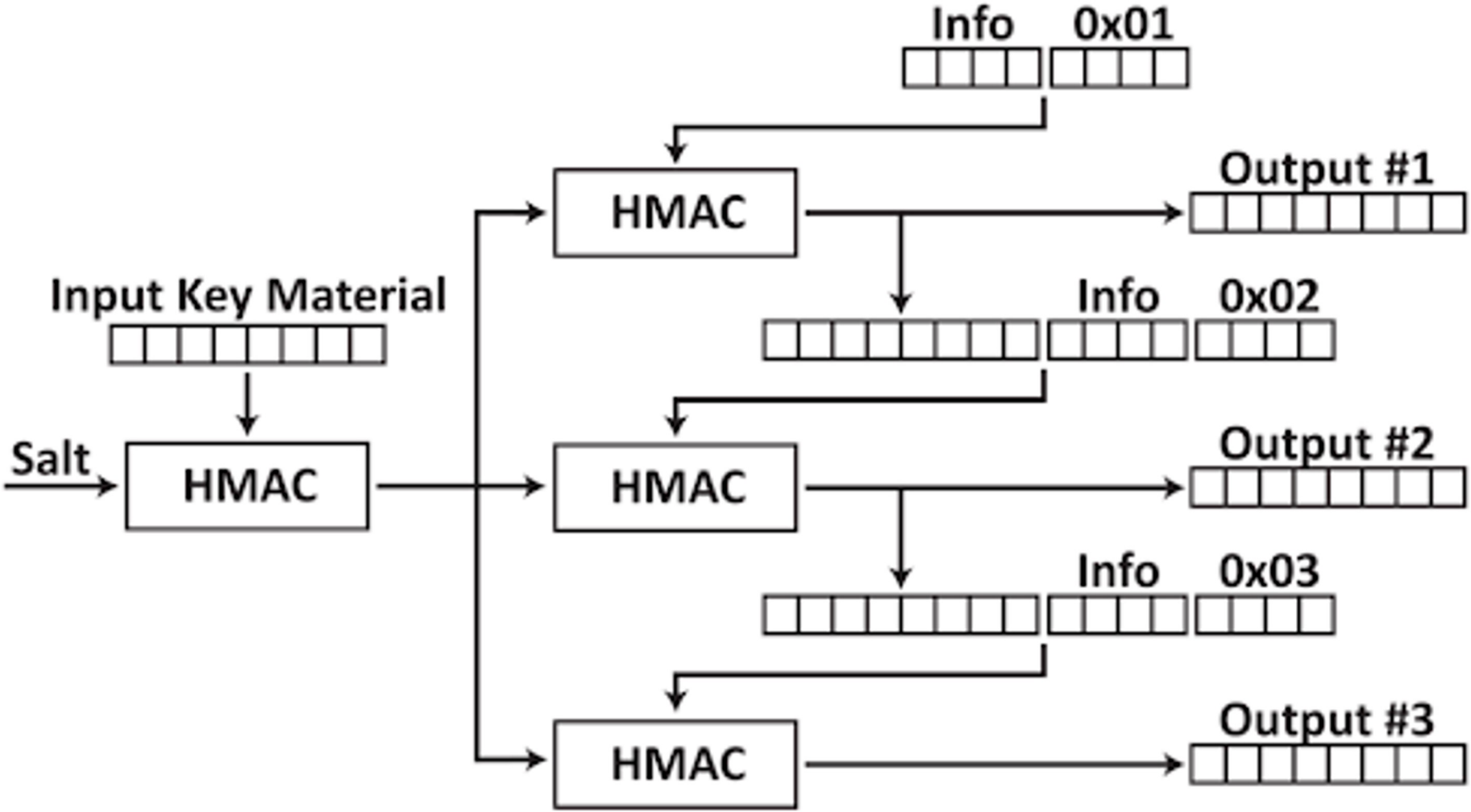
1. 隨機元info，可以為空

1. 指定密鑰長度

1. 輸出結果為 OKM

```
N = ceil(L/HashLen)
T = T(1) | T(2) | T(3) | ... | T(N)
OKM = first L octets of T

where:
T(0) = empty string (zero length)
T(1) = HMAC-Hash(PRK, T(0) | info | 0x01)
T(2) = HMAC-Hash(PRK, T(1) | info | 0x02)
T(3) = HMAC-Hash(PRK, T(2) | info | 0x03)
...
```


Info
0x01
HMAC
Output #1
Input Key Material
Info
0x02
Salt
HMAC
HMAC
Output #2
Info
0x03
HMAC
Output #3

# 新版

## TEK (Temporary Exposure Key)

- 每天生成一次
- 16字節長度，通過算法保證每個用戶唯一
- 該密鑰由操作系統組件生成

$$tek_i \leftarrow CRNG(16)$$

EKRollingPeriod

一天當中有幾個10分鐘 => 144

ENIntervalNumber

從1970年1月1日到當前有幾個十分鐘，其算法如下:

$$\text{ENIntervalNumber}(\text{Timestamp}) \leftarrow \frac{\text{Timestamp}}{60 \times 10}$$

## RPIK (Rolling Proximity Identifier Key)

$$\text{RPIK}_i \leftarrow HKDF(tek_i, NULL, \text{UTF8("EN-RPIK")}, 16)$$

## RPI (Rolling Proximity Identifier)

其中PaddedData :

- PaddedData [0-5] = UTF-8( “EN-RPI” )
- PaddedData [6-11] = 0x0000000000
- PaddedData [12-15] = ENIntervalNumber

$$\text{RPI}_{i,j} \leftarrow AES_{128}(RPIK_i, \text{PaddedData}_j)$$

## AEMK (Associate Encrypted MetaData Key)

$$AEMK_i \leftarrow HKDF(tek_i, NULL, \text{UTF8("CT-AEMK")}, 16)$$

## AEM (Associate Encrypted MetaData)

其中MetaData :

- MetaData [0] = 藍牙版本號
- MetaData [1] = 信號強度(Rssi)
- MetaData [2-3] = 暫時保留

$$\text{Associated Encrypted Metadata}_{i,j} \leftarrow AES_{128} - CTR(AEMK_i, RPI_{i,j}, \text{Metadata})$$

| V1.1 | V1.2 |
| --- | --- |
| TK (Tracing Key)<br><br>DTK (Daily Tracing Key) | TEK (Temporary Exposure Key) |
| DayNumber<br><br>TimeIntervalNumber | EKRollingPeriod<br><br>EnIntervalNumber |
| RPI (Rolling Proximity Identifier) | RPIK(Rolling Proximity Identifier key)<br><br>RPI (Rolling Proximity Identifier) |
| | AEMK (Associated Encrypted Metadata Key)<br><br>AEM (Associated Encrypted Metadata) |

# V1.1

| Flags | | | Complete 16-bit ServiceUUID | | | Service Data - 16 bit UUID | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Length** | **Type** | **Flags** | **Length** | **Type** | **ServiceUUID** | **Length** | **Type** | **ServiceData** | |
| 0x02 | 0x01 (Flag) | 0x1A | 0x03 | 0x03<br><br>(Complete 16-bit ServiceUUID) | 0xFD6F<br><br>(Contact Detection Service) | 0x13 | 0x16<br><br>(Service Data - 16 bit UUID) | 0xFD6F<br><br>(Contact Detection Service) | **16 bytes**<br><br>**Rolling Proximity Identifier** |

# V1.2

| Flags | | | Complete 16-bit Service UUID | | | Service Data - 16 bit UUID | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Length** | **Type** | **Flags** | **Length** | **Type** | **Service UUID** | **Length** | **Type** | **Service Data** | | |
| 0x02 | 0x01 (Flag) | 0x1A | 0x03 | 0x03<br><br>(Complete 16-bit Service UUID) | 0xFD6F<br><br>(Exposure Notification Service) | 0x17 | 0x16<br><br>(Service Data - 16 bit UUID) | 0xFD6F<br><br>(Exposure Notification Service) | **16 bytes**<br><br>**Rolling Proximity Identifier** | **4 bytes**<br><br>**Associated Encrypted Metadata** |

# AES 128

# AES 128 (Advanced Encryption Standard)

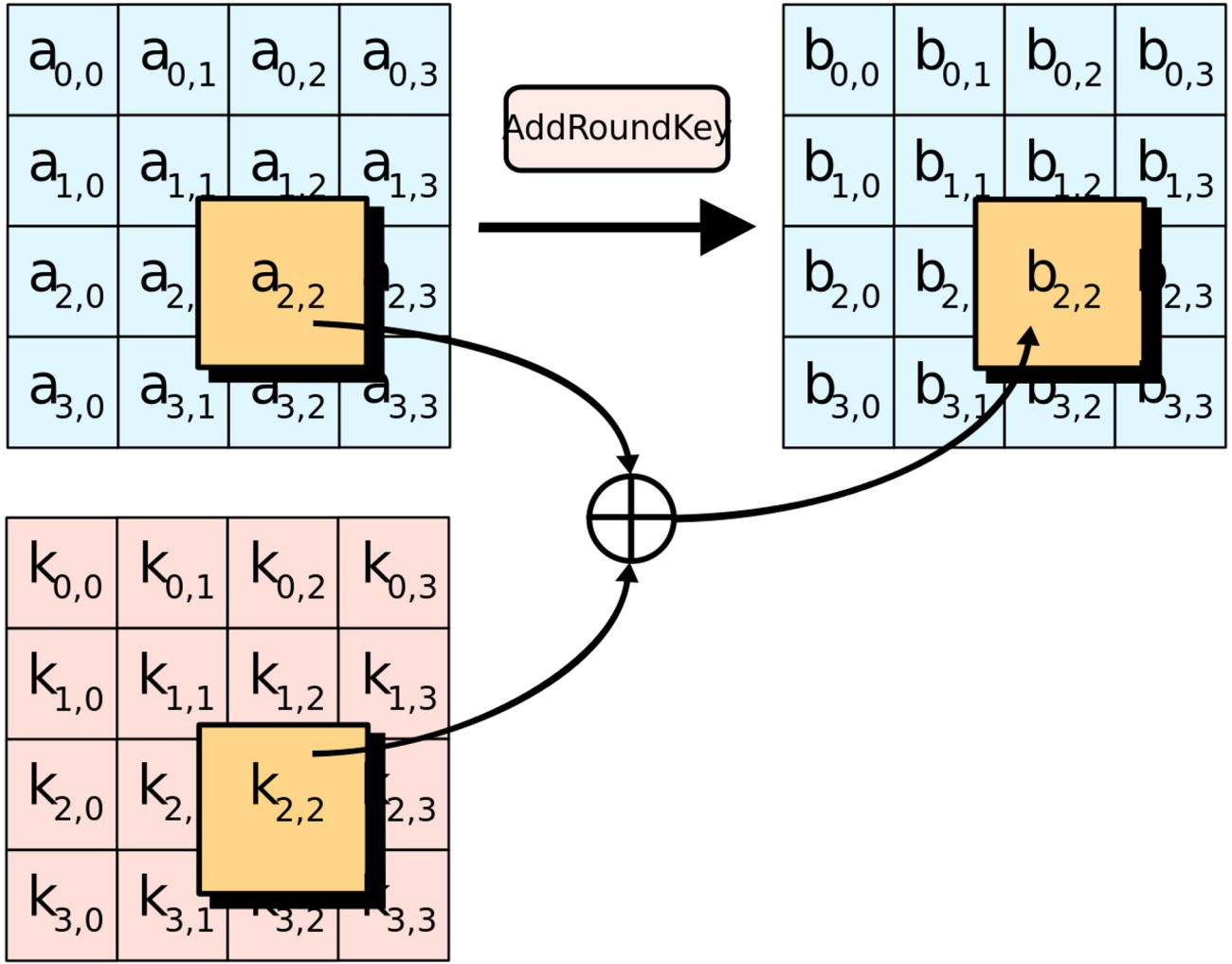
1. AddRoundKey

1. SubBytes

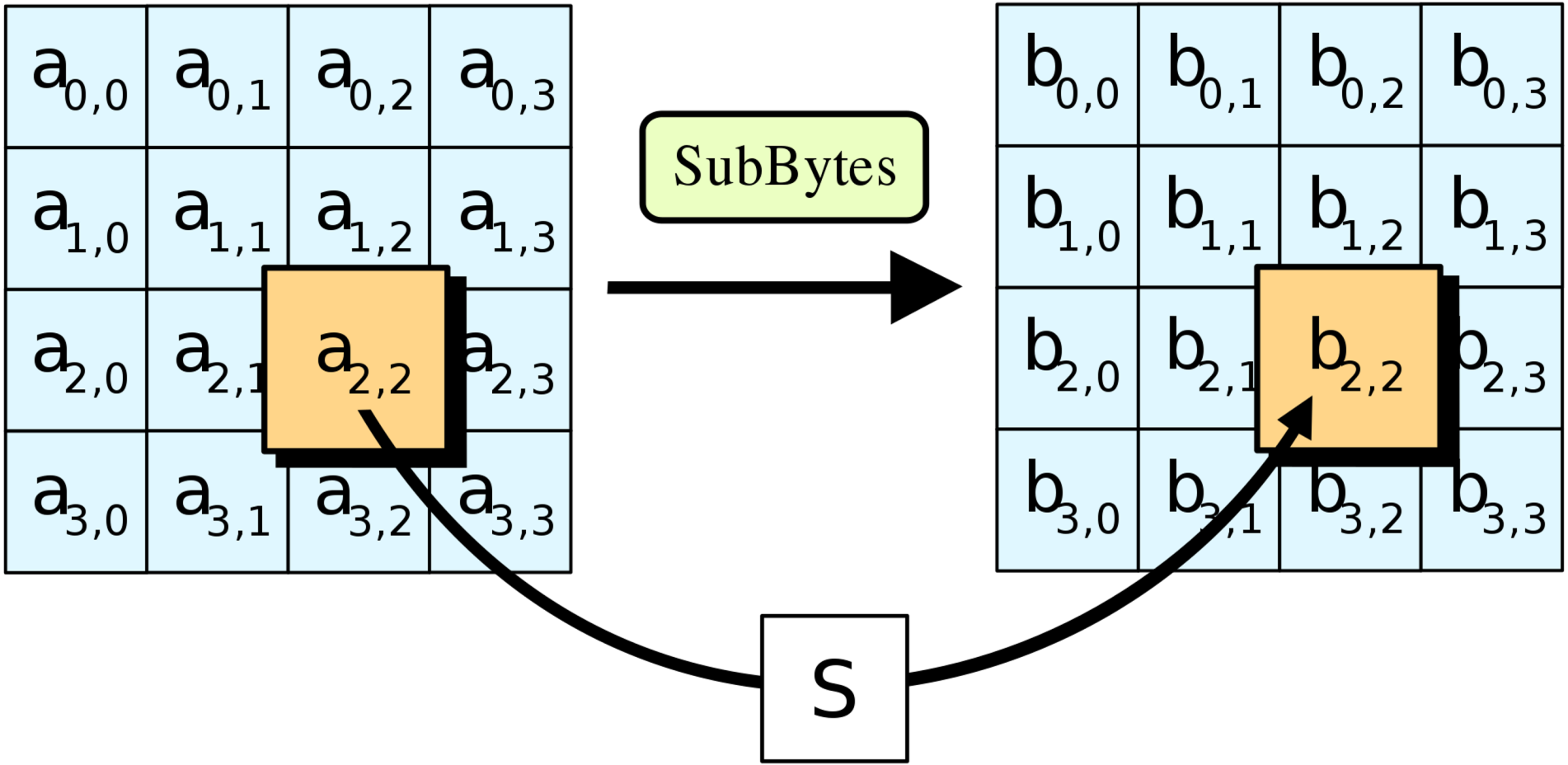
1. ShiftRows

1. MixCoulmns

# AddRoundKey


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,}$ $a_{2,2}$ $2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
AddRoundKey
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,}$ $b_{2,2}$ $2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
$k_{0,0}$ $k_{0,1}$ $k_{0,2}$ $k_{0,3}$
$k_{1,0}$ $k_{1,1}$ $k_{1,2}$ $k_{1,3}$
$k_{2,0}$ $k_{2,}$ $k_{2,2}$ $2,3}$
$k_{3,0}$ $k_{3,1}$ $k_{3,2}$ $k_{3,3}$

# SubBytes


a0,0 a0,1 a0,2 a0,3
a1,0 a1,1 a1,2 a1,3
a2,0 a2,1 a2,2 a2,3
a3,0 a3,1 a3,2 a3,3
SubBytes
b0,0 b0,1 b0,2 b0,3
b1,0 b1,1 b1,2 b1,3
b2,0 b2,1 b2,2 b2,3
b3,0 b3,1 b3,2 b3,3
S

# 轉換-取代(SubBytes)

- 範例 3.2 展示一組狀態如何使用SubBytes進行轉換。

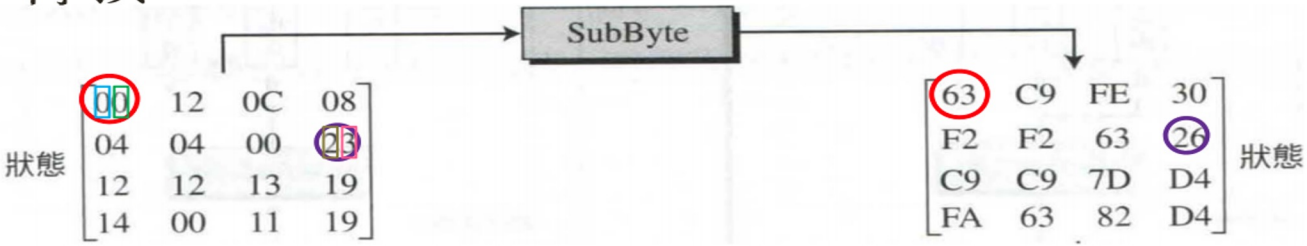


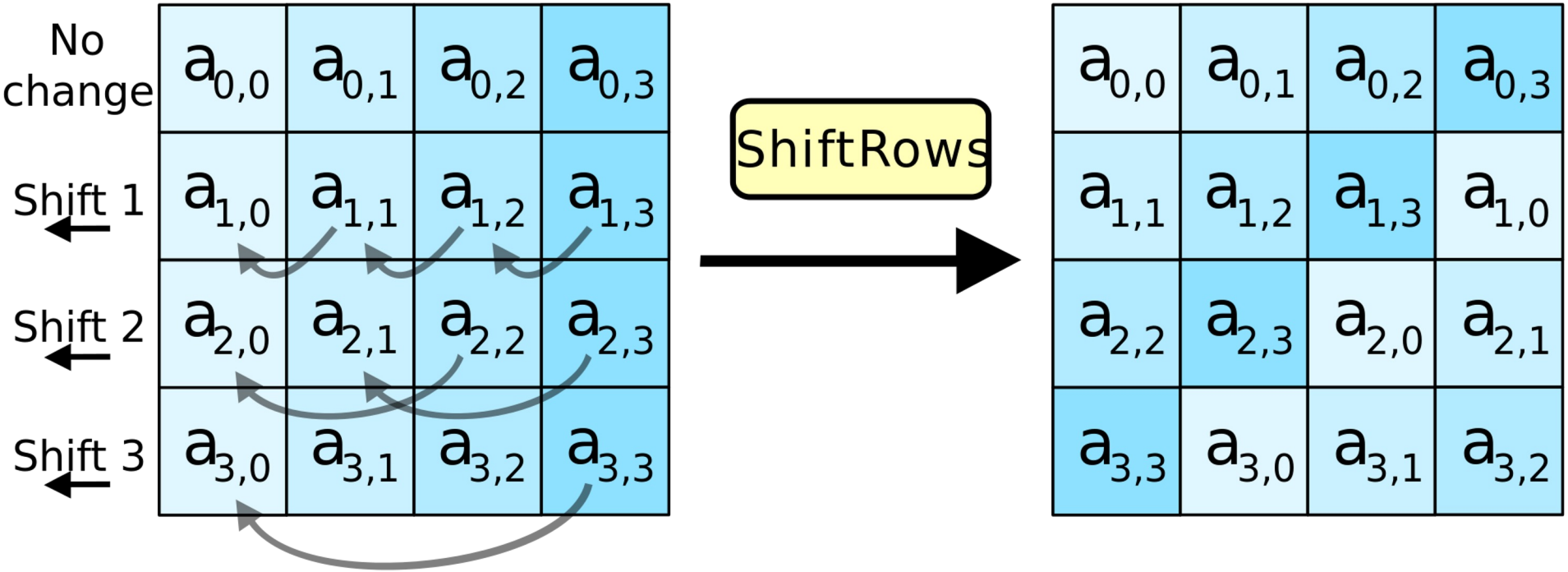

表 7.1 SubBytes 轉換表

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 7C | 77 | 7B | F2 | 6B | 6F | C5 | 30 | 01 | 67 | 2B | FE | D7 | AB | 76 |
| 1 | CA | 82 | C9 | 7D | FA | 59 | 47 | F0 | AD | D4 | A2 | AF | 9C | A4 | 72 | C0 |
| 2 | B7 | FD | 93 | 26 | 36 | 3F | F7 | CC | 34 | A5 | E5 | F1 | 71 | D8 | 31 | 15 |
| 3 | 04 | C7 | 23 | C3 | 18 | 96 | 05 | 9A | 07 | 12 | 80 | E2 | EB | 27 | B2 | 75 |
| 4 | 09 | 83 | 2C | 1A | 1B | 6E | 5A | A0 | 52 | 3B | D6 | B3 | 29 | E3 | 2F | 84 |
| 5 | 53 | D1 | 00 | ED | 20 | FC | B1 | 5B | 6A | CB | BE | 39 | 4A | 4C | 58 | CF |
| 6 | D0 | EF | AA | FB | 43 | 4D | 33 | 85 | 45 | F9 | 02 | 7F | 50 | 3C | 9F | A8 |
| 7 | 51 | A3 | 40 | 8F | 92 | 9D | 38 | F5 | BC | B6 | DA | 21 | 10 | FF | F3 | D2 |
| 8 | CD | 0C | 13 | EC | 5F | 97 | 44 | 17 | C4 | A7 | 7E | 3D | 64 | 5D | 19 | 73 |
| 9 | 60 | 81 | 4F | DC | 22 | 2A | 90 | 88 | 46 | EE | B8 | 14 | DE | 5E | 0B | DB |
| A | E0 | 32 | 3A | 0A | 49 | 06 | 24 | 5C | C2 | D3 | AC | 62 | 91 | 95 | E4 | 79 |
| B | E7 | CB | 37 | 6D | 8D | D5 | 4E | A9 | 6C | 56 | F4 | EA | 65 | 7A | AE | 08 |
| C | BA | 78 | 25 | 2E | 1C | A6 | B4 | C6 | E8 | DD | 74 | 1F | 4B | BD | 8B | 8A |
| D | 70 | 3E | B5 | 66 | 48 | 03 | F6 | 0E | 61 | 35 | 57 | B9 | 86 | C1 | 1D | 9E |
| E | E1 | F8 | 98 | 11 | 69 | D9 | 8E | 94 | 9B | 1E | 87 | E9 | CE | 55 | 28 | DF |
| F | 8C | A1 | 89 | 0D | BF | E6 | 42 | 68 | 41 | 99 | 2D | 0F | B0 | 54 | BB | 16 |

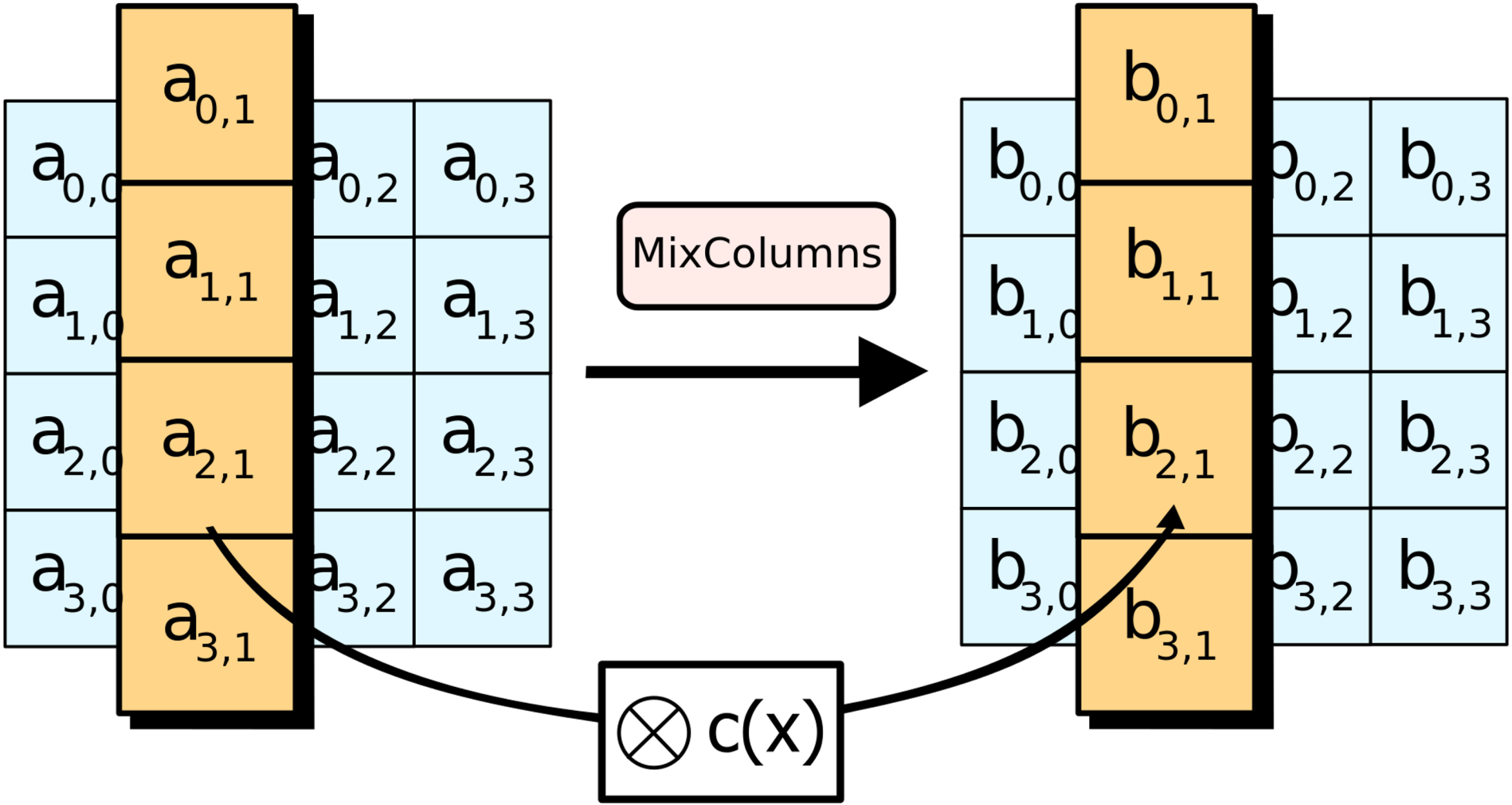
表 7.2 InvSubBytes 轉換表

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 7C | 77 | 7B | F2 | 6B | 6F | C5 | 30 | 01 | 67 | 2B | FE | D7 | AB | 76 |
| 1 | CA | 82 | C9 | 7D | FA | 59 | 47 | F0 | AD | D4 | A2 | AF | 9C | A4 | 72 | C0 |
| 2 | B7 | FD | 93 | 26 | 36 | 3F | F7 | CC | 34 | A5 | E5 | F1 | 71 | D8 | 31 | 15 |
| 3 | 04 | C7 | 23 | C3 | 18 | 96 | 05 | 9A | 07 | 12 | 80 | E2 | EB | 27 | B2 | 75 |
| 4 | 09 | 83 | 2C | 1A | 1B | 6E | 5A | A0 | 52 | 3B | D6 | B3 | 29 | E3 | 2F | 84 |
| 5 | 53 | D1 | 00 | ED | 20 | FC | B1 | 5B | 6A | CB | BE | 39 | 4A | 4C | 58 | CF |
| 6 | D0 | EF | AA | FB | 43 | 4D | 33 | 85 | 45 | F9 | 02 | 7F | 50 | 3C | 9F | A8 |
| 7 | 51 | A3 | 40 | 8F | 92 | 9D | 38 | F5 | BC | B6 | DA | 21 | 10 | FF | F3 | D2 |
| 8 | CD | 0C | 13 | EC | 5F | 97 | 44 | 17 | C4 | A7 | 7E | 3D | 64 | 5D | 19 | 73 |
| 9 | 60 | 81 | 4F | DC | 22 | 2A | 90 | 88 | 46 | EE | B8 | 14 | DE | 5E | 0B | DB |
| A | E0 | 32 | 3A | 0A | 49 | 06 | 24 | 5C | C2 | D3 | AC | 62 | 91 | 95 | E4 | 79 |
| B | E7 | CB | 37 | 6D | 8D | D5 | 4E | A9 | 6C | 56 | F4 | EA | 65 | 7A | AE | 08 |
| C | BA | 78 | 25 | 2E | 1C | A6 | B4 | C6 | E8 | DD | 74 | 1F | 4B | BD | 8B | 8A |
| D | 70 | 3E | B5 | 66 | 48 | 03 | F6 | 0E | 61 | 35 | 57 | B9 | 86 | C1 | 1D | 9E |
| E | E1 | F8 | 98 | 11 | 69 | D9 | 8E | 94 | 9B | 1E | 87 | E9 | CE | 55 | 28 | DF |
| F | 8C | A1 | 89 | 0D | BF | E6 | 42 | 68 | 41 | 99 | 2D | 0F | B0 | 54 | BB | 16 |

# ShiftRows


No change
Shift 1
Shift 2
Shift 3
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

# MixColumns


a0,0
a0,1
a0,2
a0,3
a1,0
a1,1
a1,2
a1,3
a2,0
a2,1
a2,2
a2,3
a3,0
a3,1
a3,2
a3,3
MixColumns
b0,0
b0,1
b0,2
b0,3
b1,0
b1,1
b1,2
b1,3
b2,0
b2,1
b2,2
b2,3
b3,0
b3,1
b3,2
b3,3
c(x)

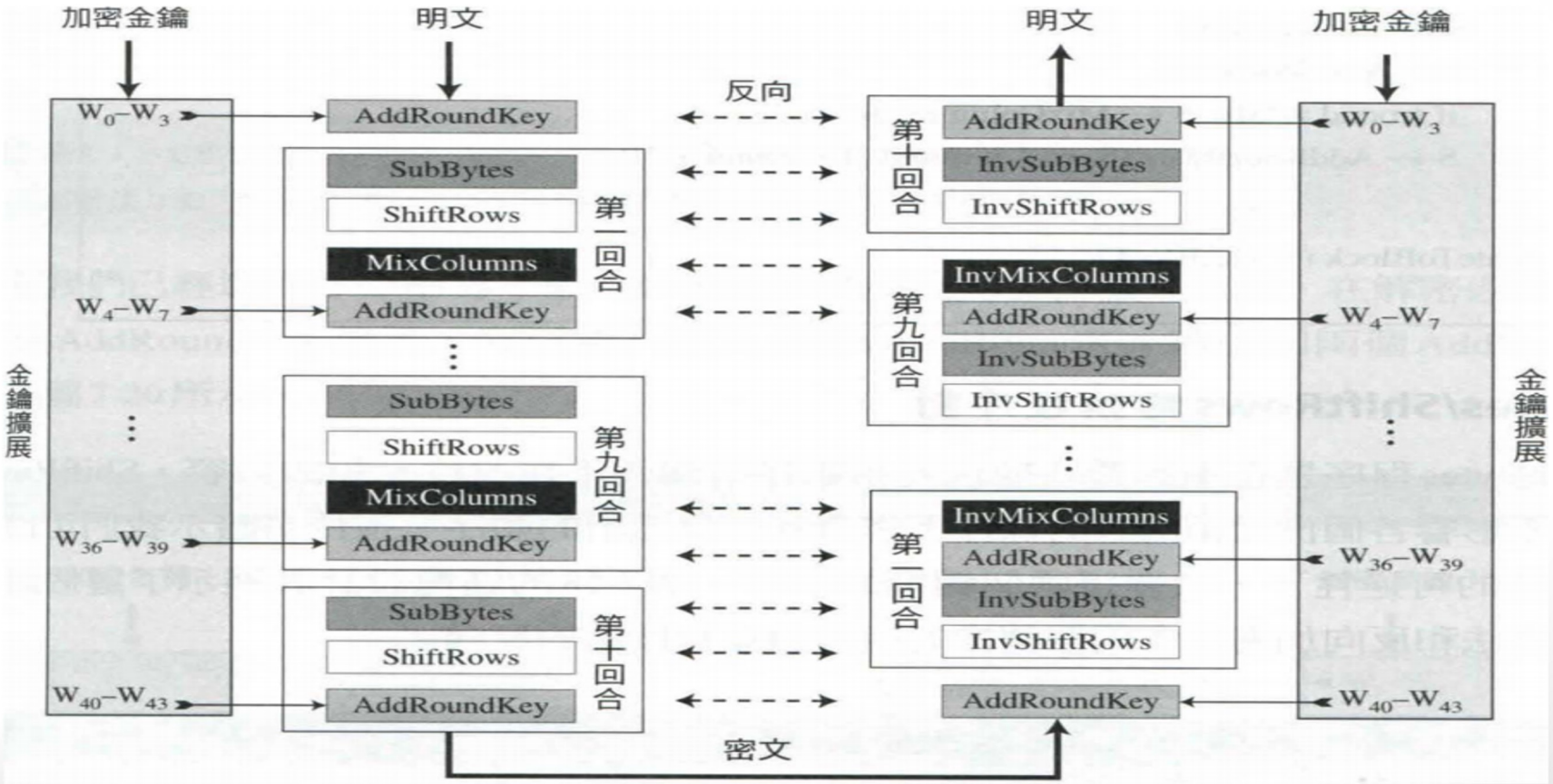
# 轉換-混合(MixColumns)

- 例題3.7 展示一組狀態如何使用MixColumns進行轉換。


$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} 63 \\ F2 \\ 7D \\ D4 \end{bmatrix} = \begin{bmatrix} s_{0,0} \\ s_{1,0} \\ s_{2,0} \\ s_{3,0} \end{bmatrix} \Rightarrow \begin{cases} s_{0,0}(new) = (\{02\} \cdot 63) \oplus (\{03\} \cdot F2) \oplus 7D \oplus D4 \\ s_{1,0}(new) = (\{02\} \cdot F2) \oplus (\{03\} \cdot 7D) \oplus D4 \oplus 63 \\ s_{2,0}(new) = (\{02\} \cdot 7D) \oplus (\{03\} \cdot D4) \oplus 63 \oplus F2 \\ s_{3,0}(new) = (\{02\} \cdot D4) \oplus (\{03\} \cdot 63) \oplus F2 \oplus 7D \end{cases}$$

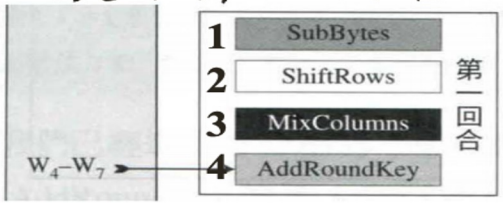
# AES進階加密法

- 進階加密法的加密



# AES範例

- 展示第一回合的狀態值變化。




| 回合 | 輸入狀態 | 輸出狀態 | 回合金鑰 |
|---|---|---|---|
| 1 | 24 26 3D 1B<br>71 71 E2 89<br>B0 44 01 4D<br>A7 88 11 9E | 6C 44 13 BD<br>B1 9E 46 35<br>C5 B5 F3 02<br>5D 87 FC 8C | 89 BD 8C 9F<br>55 20 C2 68<br>B5 E3 F1 A5<br>CE 46 46 C1 |

- 1. 首先將輸入狀態對應SubByte轉換表

| | | | |
|---|---|---|---|
| 24 | 26 | 3D | 1B |
| 71 | 71 | E2 | 89 |
| B0 | 44 | 01 | 4D |
| A7 | 88 | 11 | 9E |

輸入狀態

表 7.1 SubBytes 轉換表

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 7C | 77 | 7B | F2 | 6B | 6F | C5 | 30 | 01 | 67 | 2B | FE | D7 | AB | 76 |
| 1 | CA | 82 | C9 | 7D | FA | 59 | 47 | F0 | AD | D4 | A2 | AF | 9C | A4 | 72 | C0 |
| 2 | B7 | FD | 93 | 26 | 36 | 3F | F7 | CC | 34 | A5 | E5 | F1 | 71 | D8 | 31 | 15 |
| 3 | 04 | C7 | 23 | C3 | 18 | 96 | 05 | 9A | 07 | 12 | 80 | E2 | EB | 27 | B2 | 75 |
| 4 | 09 | 83 | 2C | 1A | 1B | 6E | 5A | A0 | 52 | 3B | D6 | B3 | 29 | E3 | 2F | 84 |
| 5 | 53 | D1 | 00 | ED | 20 | FC | B1 | 5B | 6A | CB | BE | 39 | 4A | 4C | 58 | CF |
| 6 | D0 | EF | AA | FB | 43 | 4D | 33 | 85 | 45 | F9 | 02 | 7F | 50 | 3C | 9F | A8 |
| 7 | 51 | A3 | 40 | 8F | 92 | 9D | 38 | F5 | BC | B6 | DA | 21 | 10 | FF | F3 | D2 |
| 8 | CD | 0C | 13 | EC | 5F | 97 | 44 | 17 | C4 | A7 | 7E | 3D | 64 | 5D | 19 | 73 |
| 9 | 60 | 81 | 4F | DC | 22 | 2A | 90 | 88 | 46 | EE | B8 | 14 | DE | 5E | 0B | DB |
| A | E0 | 32 | 3A | 0A | 49 | 06 | 24 | 5C | C2 | D3 | AC | 62 | 91 | 95 | E4 | 79 |
| B | E7 | CB | 37 | 6D | 8D | D5 | 4E | A9 | 6C | 56 | F4 | EA | 65 | 7A | AE | 08 |
| C | BA | 78 | 25 | 2E | 1C | A6 | B4 | C6 | E8 | DD | 74 | 1F | 4B | BD | 8B | 8A |
| D | 70 | 3E | B5 | 66 | 48 | 03 | F6 | 0E | 61 | 35 | 57 | B9 | 86 | C1 | 1D | 9E |
| E | E1 | F8 | 98 | 11 | 69 | D9 | 8E | 94 | 9B | 1E | 87 | E9 | CE | 55 | 28 | DF |
| F | 8C | A1 | 89 | 0D | BF | E6 | 42 | 68 | 41 | 99 | 2D | 0F | B0 | 54 | BB | 16 |

| | | | |
|---|---|---|---|
| 36 | F7 | 27 | AF |
| A3 | A3 | 98 | A7 |
| E7 | 1B | 7C | E3 |
| 5C | C4 | 82 | 0B |

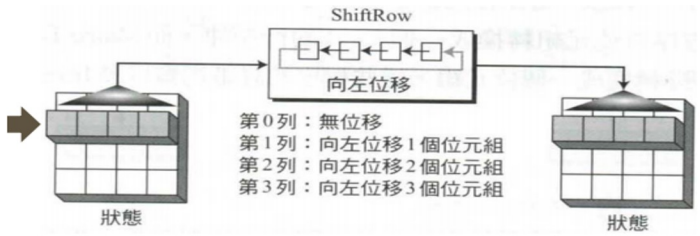
SubByte程序後

# AES範例

- 2.經ShiftRows左移位元組

| | | | |
|---|---|---|---|
| 36 | F7 | 27 | AF |
| A3 | A3 | 98 | A7 |
| E7 | 1B | 7C | E3 |
| 5C | C4 | 82 | 0B |

SubByte程序後



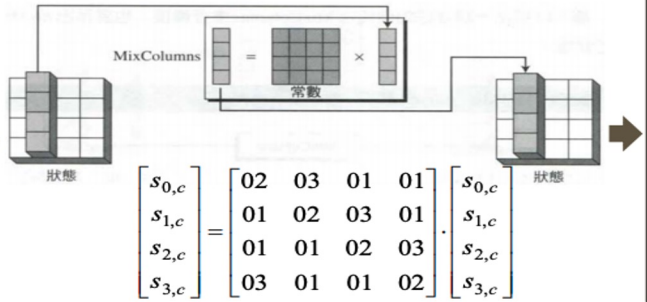

| | | | |
|---|---|---|---|
| 36 | F7 | 27 | AF |
| A3 | 98 | A7 | A3 |
| 7C | E3 | E7 | 1B |
| 0B | 5C | C4 | 82 |

ShiftRows程序後

- 3.使用MixColumns計算值

| | | | |
|---|---|---|---|
| 36 | F7 | 27 | AF |
| A3 | 98 | A7 | A3 |
| 7C | E3 | E7 | 1B |
| 0B | 5C | C4 | 82 |

ShiftRows程序後

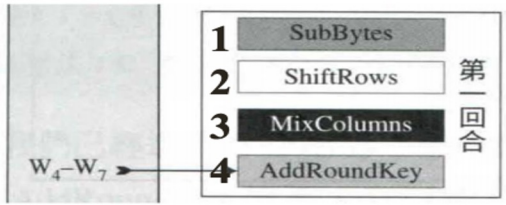



| | | | |
|---|---|---|---|
| E5 | F9 | 9F | 22 |
| E4 | BE | 84 | 5D |
| 70 | 56 | 02 | A7 |
| 93 | C1 | BA | 4D |

MixColumns程序後

# AES範例

- 4. 最後加入回合金鑰(第一回合金鑰$w_4$-$w_7$)，且第一回合的密文輸出。




| | | | |
|---|---|---|---|
| E5 | F9 | 9F | 22 |
| E4 | BE | 84 | 5D |
| 70 | 56 | 02 | A7 |
| 93 | C1 | BA | 4D |

MixColumns程序後

⊕ (第一回合金鑰$w_4$-$w_7$)

| 回合 | 回合金鑰 |
|---|---|
| 1 | 89 BD 8C 9F<br>55 20 C2 68<br>B5 E3 F1 A5<br>CE 46 46 C1 |

| 輸出狀態 |
|---|
| 6C 44 13 BD |
| B1 9E 46 35 |
| C5 B5 F3 02 |
| 5D 87 FC 8C |

$$E5 \oplus 89 = 11100101 \oplus 10001001 = 01101100$$

$$56 \oplus E3 = 01010110 \oplus 11100011 = \underbrace{1011}_{B}\underbrace{0101}_{5}$$

# END