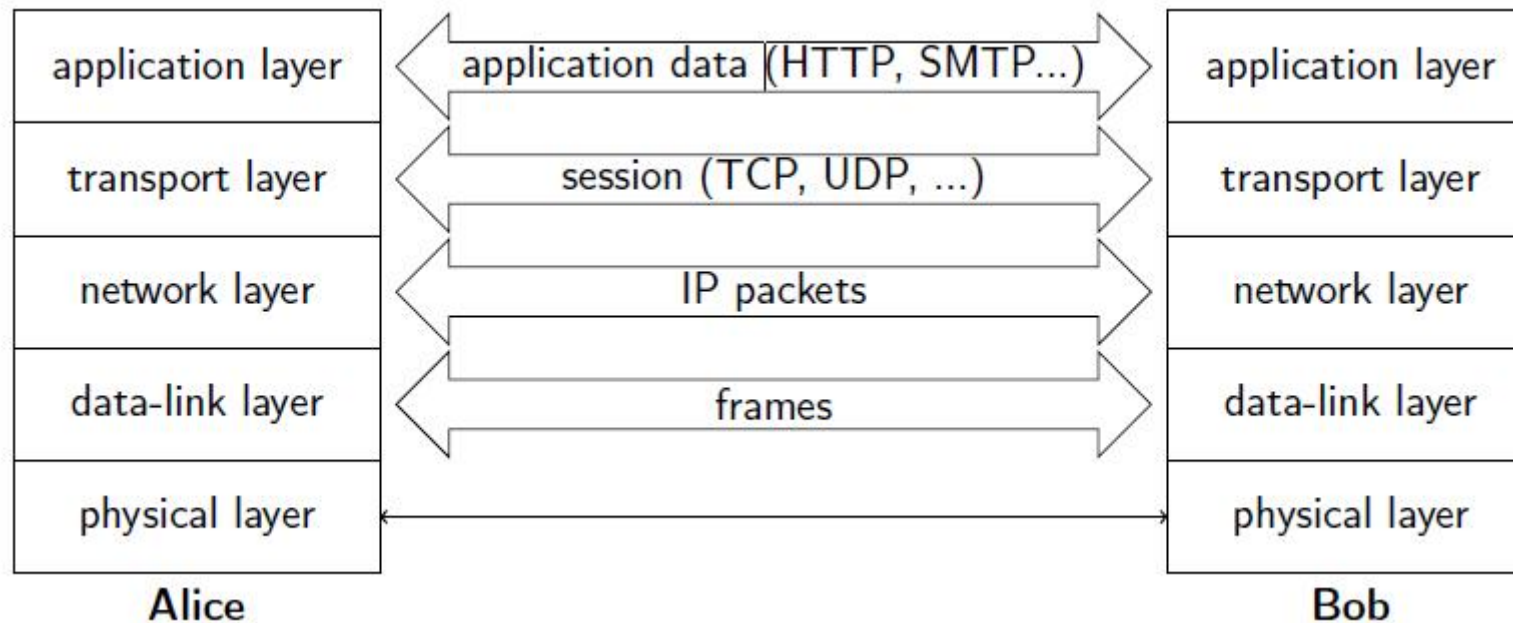


第5章 无线局域网安全



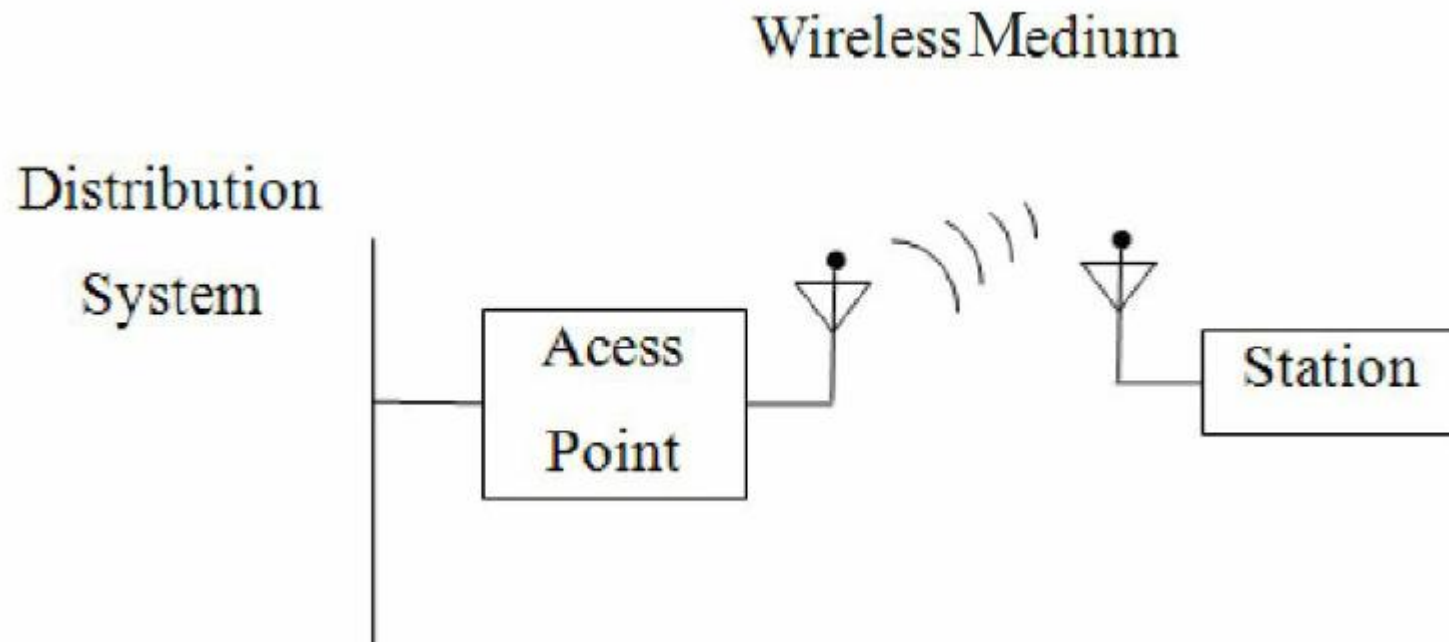
- application layer security (SSH, S-MIME, PGP,)
- transport layer security (TLS/SSL,)
- network layer security (IPsec,)
- data-link layer security (WEP, WPA, WPA2,)

主要内容

- 无线局域网简介
- WEP
- WPA/WPA2原理
- 无线局域网安全措施

无线网络简介

无线网络组件



□无线媒介 (Wireless Medium)：传输无线MAC帧的媒介，主要包括射频和红外两种，目前主要指射频

无线网络组件

□ STA(Station)

- ◆ 具有无线网卡的设备，比如：笔记本电脑、智能手机等

□ AP(Access Point)

- ◆ 无线接入点，能为已经关联的STA提供DS服务

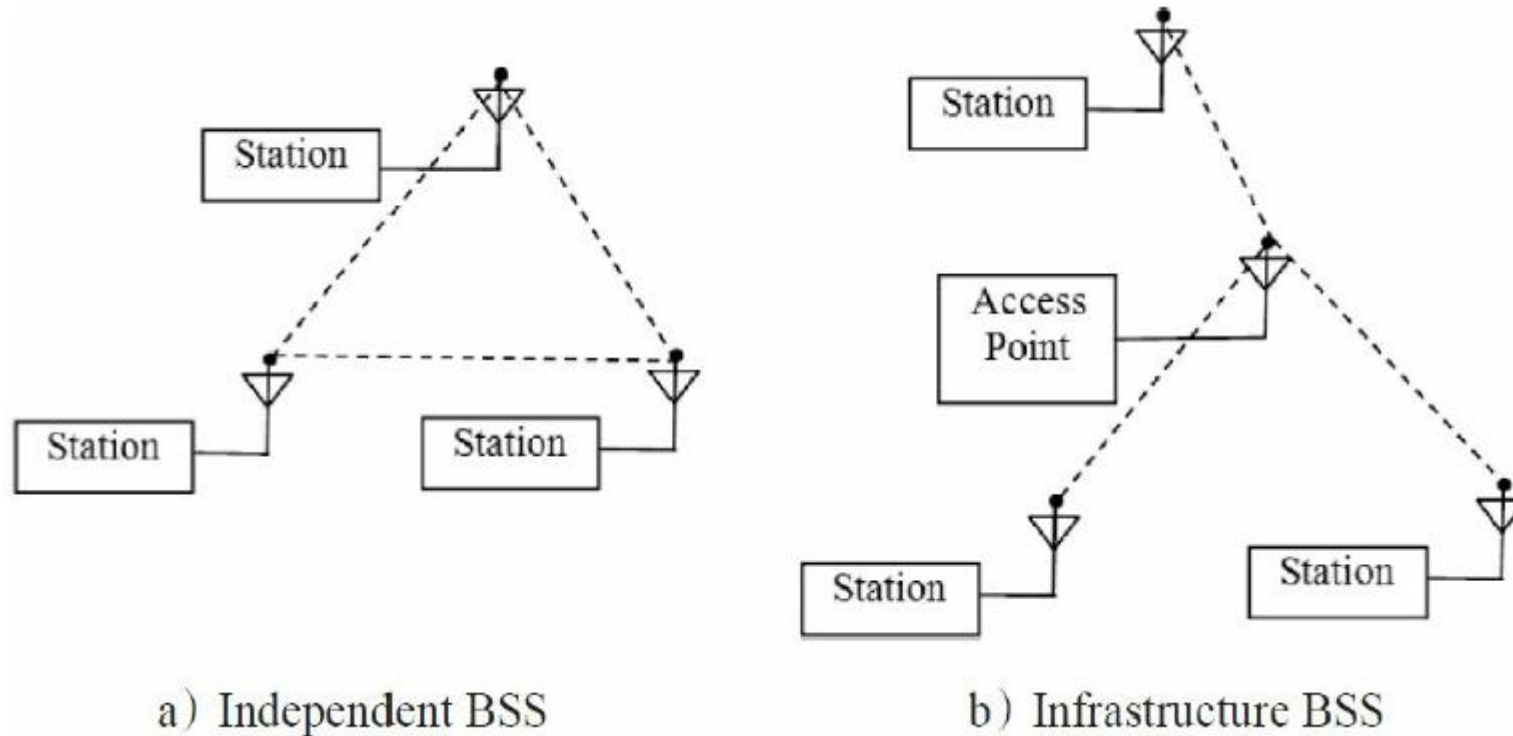
□ DS (Distributed System)

- ◆ A system used to interconnect a set of basic service sets (**BSSs**) and integrated local area networks (LANs) to create an extended service set (**ESS**) .

无线网络的构建

- **SS** (Service set) , 服务集
 - ◆ Service set is a group of wireless network devices that are operating with **the same networking parameters**.
- **BSS**(Basic Service Set), 基本服务集
 - ◆ Basic service sets are units of devices operating with **the same medium access characteristics** (i.e. radio frequency, modulation scheme etc.),
 - ◆ 包括: Independent BSS和Infrastructure BSS

无线网络的构建



Independent BSS: 简称**IBSS**，联网无需AP参与，又称为ad-hoc BSS或者自组网络。

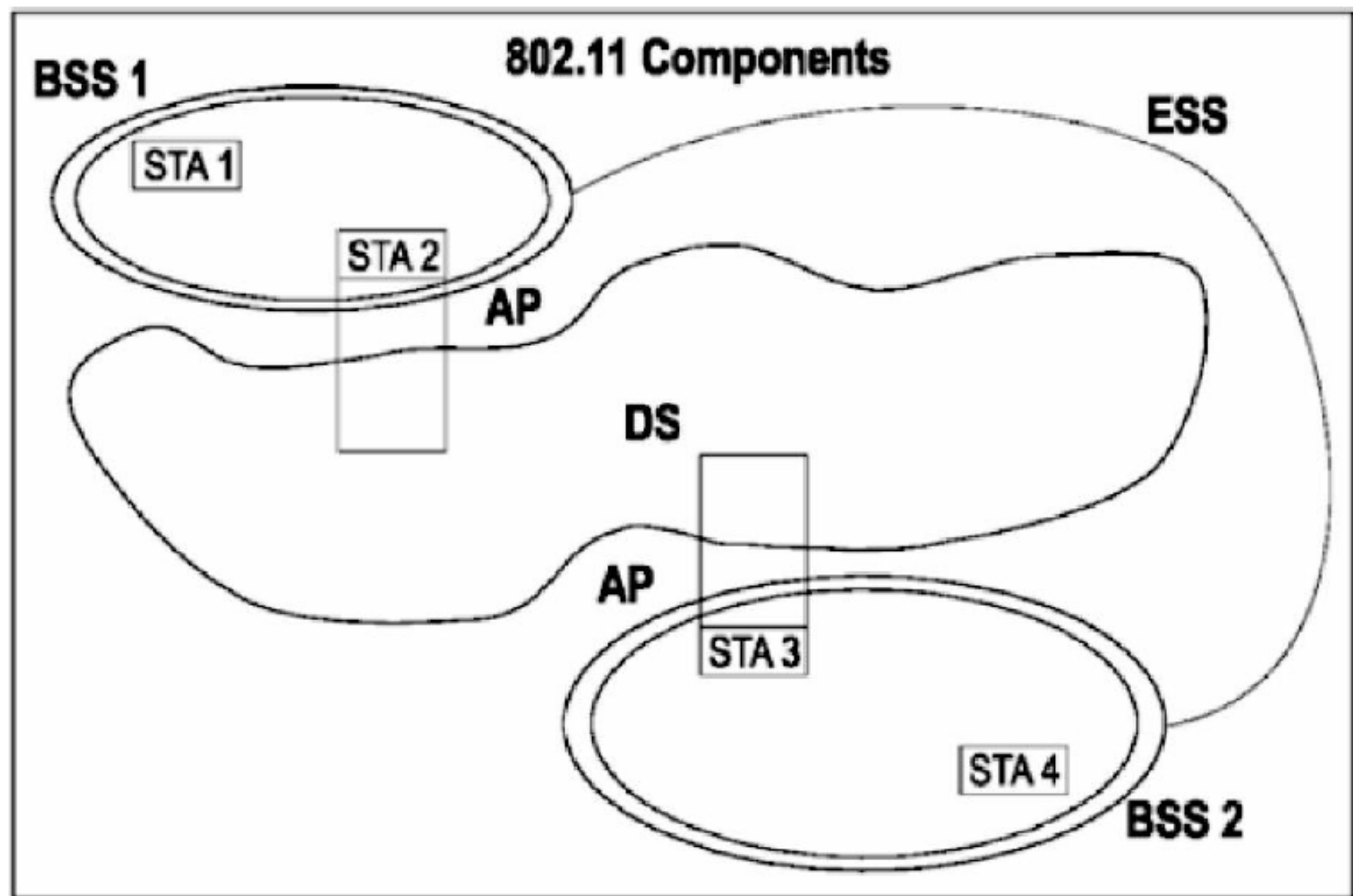
Infrastructure BSS: 一般来说，**BSS**就是指Infrastructure BSS，需要AP参与来构建网络。

无线网络的构建

□ 扩展服务集 (Extended Service Set, ESS)

- ◆ Extended service sets (ESS) are logical units of one or more basic service sets on **the same logical network segment** (i.e. IP subnet, VLAN etc.).
- ◆ Logical networks (including extended service sets) are identified by **SSIDs**, which serve as "network names" and are typically natural language labels.

扩展服务集示例



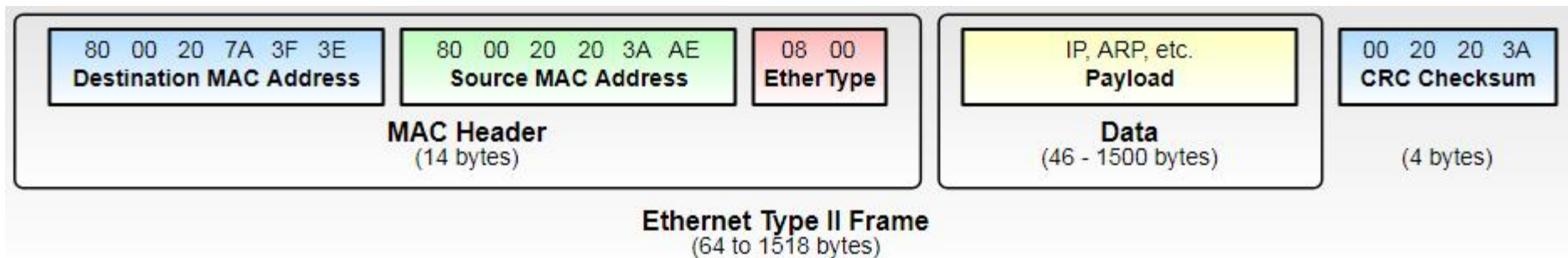
WLAN的基本工作原理

- AP周期性发送Beacon帧，用于宣布其网络的存在和网络参数
- STA发送Probe Request帧主动探测某个AP
- STA发送认证请求，AP回复认证响应
- 如果通过认证，STA发送关联请求，AP回复关联响应，实现STA和AP的关联
- STA和AP之间传输数据帧

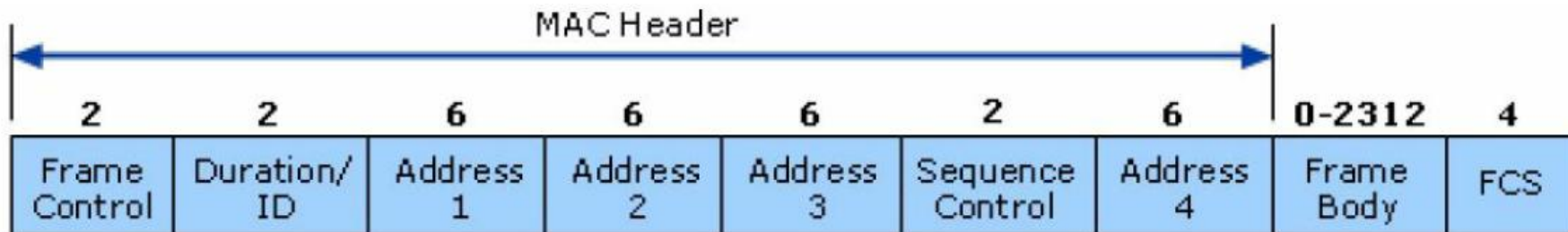
□WLAN由 IEEE 802.11定义，规范了MAC子层和PHY层

□MAC子层体现为帧，因此需要了解802.11的帧格式

对比Ethernet II帧格式

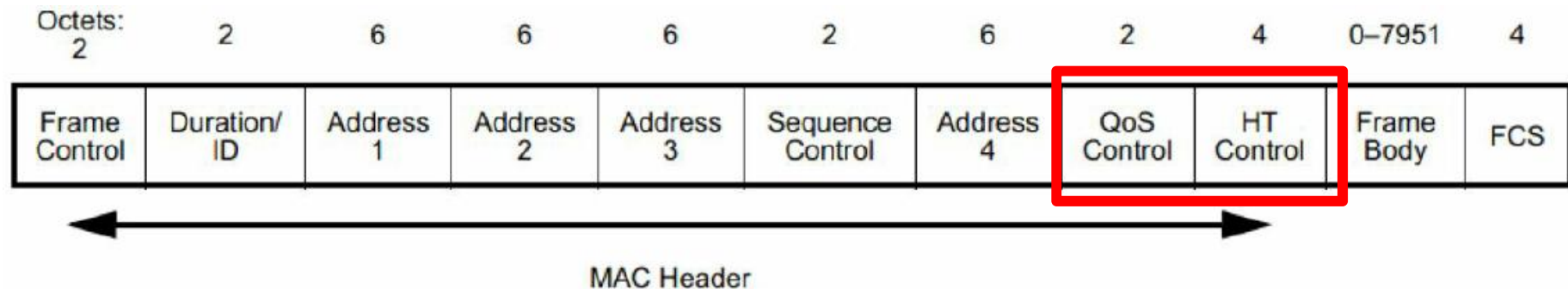


802.11 MAC帧格式



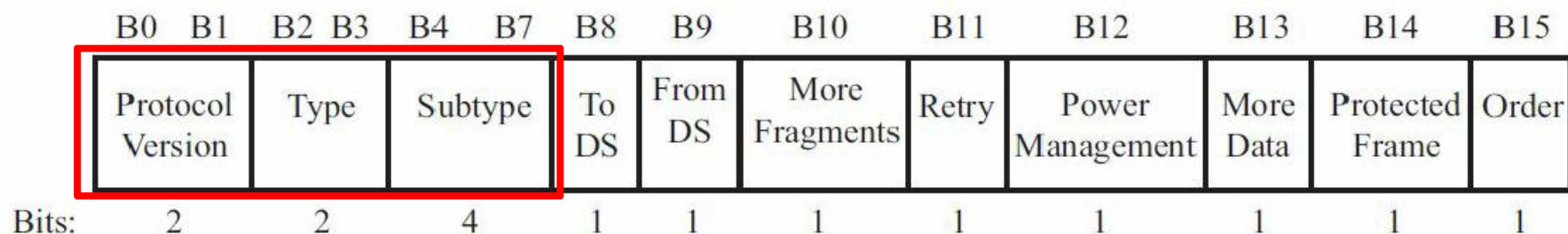
- **MAC Header**: 包括帧控制 (Frame Control)、时长 (Duration)、地址 (Address) 等
- **Frame Body**: 代表数据域。这部分内容的长度可变, 其具体存储的内容由帧类型 (type) 和子类型 (subtype) 决定
- **FCS**: (Frame Check Sequence, 帧校验序列) 用于保障帧数据的完整性

完整数据帧格式



- 如果是QoS数据帧，还需要附加QoS Control字段。
- 如果是HT（High Throughput，一种用于提高无线网络传输速率的技术）数据帧，还需要附加HT Control字段。

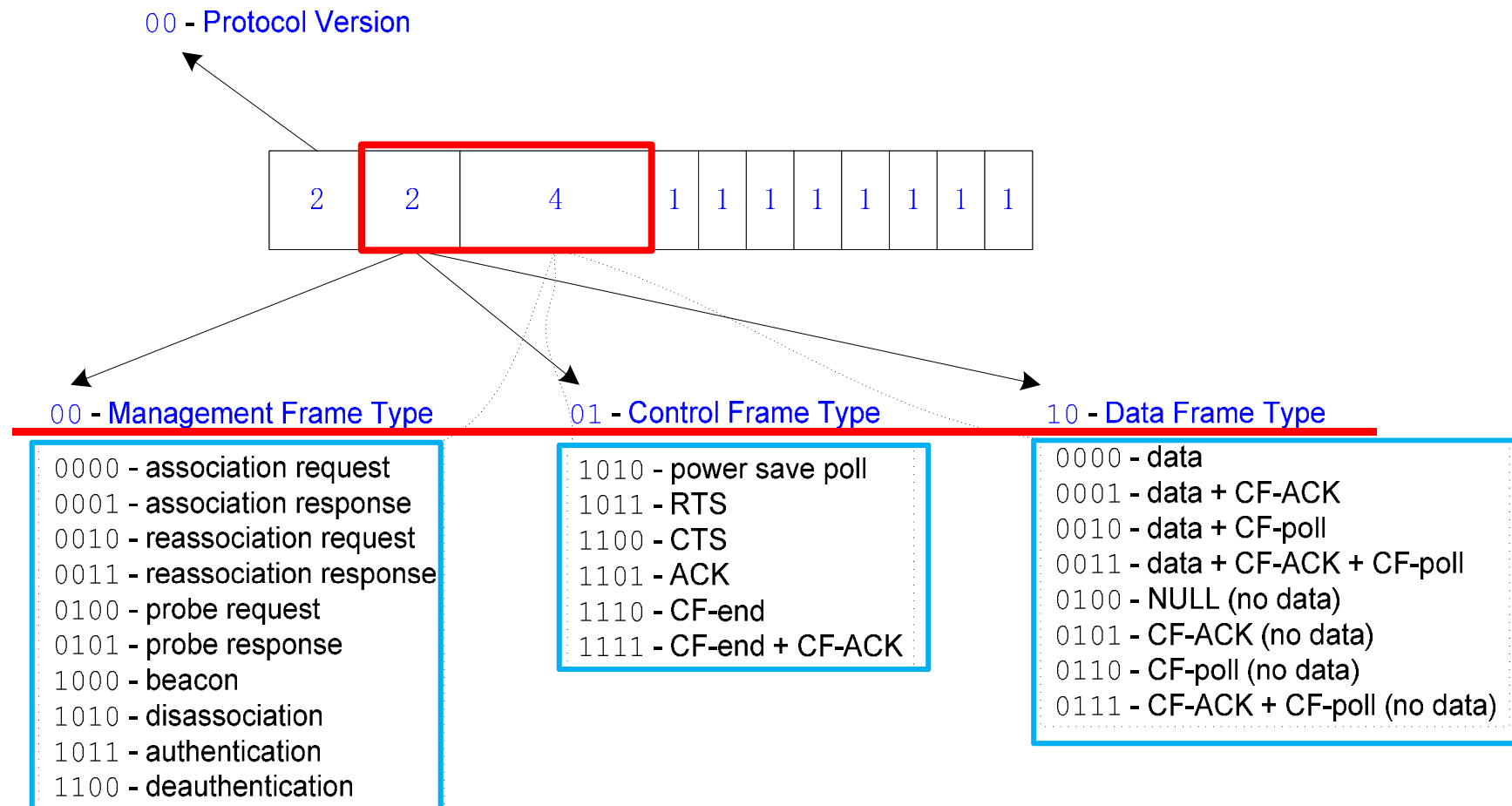
Frame Control字段



- Protocol Version: 代表802.11 MAC帧的版本号。目前的值是0。
- Type和Subtype: 这两个字段用于指明MAC帧的类型。802.11中MAC帧可划分为三种类型，分别是control、data和management，每种类型的帧用于完成不同功能。

帧控制域: types and subtypes

802.11 Types and Subtypes



帧控制域

- To DS和From DS: 只用在数据类型的帧中。
- More Fragments: 表明数据是否分片。只支持data和management帧类型。
- Retry: 如果该值为1, 表明是重传包。
- Power Management: 表明发送该帧的STA处于活跃模式还是处于省电模式。

帧控制域

- **More Data**: 和省电模式有关。AP会为那些处于省电模式下的STA缓冲一些数据帧，而STA会定时查询是否有数据要接收。该参数表示AP中还有缓冲的数据帧。如果该值为0，表明STA已经接收完数据帧了。
- **Protected Frame**: 表明数据是否加密。
- **Order**: 指明接收端必须按顺序处理该帧。

地址域

- MAC 帧头中包含四个地址域，其用法与 Frame Control 域中 **To/From DS flags** 相关。
- 原则是 Address 1 表示 Receiver Address (RA), Address 2 表示 Transmitter Address (TA), Address 3 辅助用, Address 4 用于无线桥接或 Mesh BSS 网络中。

地址类型

- Destination Address (DA) : 用来描述MAC数据帧**最终**接收者 (final recipient) , 可以是单播或组播地址。
- Source Address (SA) : 用来描述**最初**发出MAC数据帧的STA地址。一般情况下都是单播地址。
- Transmitter Address (TA) : 用于描述将MAC数据帧发送到无线媒介的实体的地址, 可以是STA或者AP。
- Receiver Address (RA) : 用于描述接收MAC数据帧的接收者地址, 可以是STA或者AP。

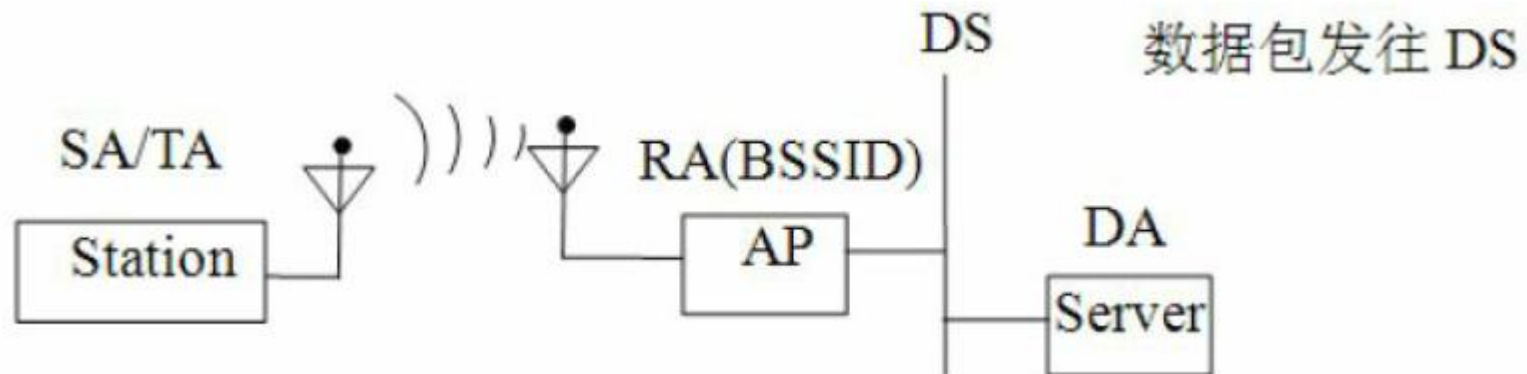
To/From DS flags

To DS	From DS	含义
0	0	在同一IBSS中, 从一个STA发给另一个STA的数据帧、管理帧或者控制帧
0	1	从一个DS出去的数据帧
1	0	发往一个DS的数据帧
1	1	从一个AP发往另一个AP的无线分布式系统帧 (无线桥接模式)

地址域用法

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	RA=DA	TA=SA	BSSID	N/A
0	1	RA=DA	TA=BSSID	SA	N/A
1	0	RA=BSSID	TA=SA	DA	N/A
1	1	RA	TA	DA	SA

地址: case 1



数据包发往DS的情况下，SA和TA一致，而RA和BSSID一致。

实例演示

IEEE 802.11 Data, Flags: .p.....TC

Type/Subtype: Data (0x0020)

Frame Control Field: 0x0841

.... ..00 = Version: 0

.... 10.. = Type: Data frame (2)

0000 = Subtype: 0

Flags: 0x41

.... ..01 = DS status: Frame from STA to DS via an AP (To DS: 1 From DS: 0) (0x1)

.... .0.. = More Fragments: This is the last fragment

.... 0... = Retry: Frame is not being retransmitted

...0 = PWR MGT: STA will stay up

..0. = More Data: No data buffered

.1.. = Protected flag: Data is protected

0... = Order flag: Not strictly ordered

.000 0000 0010 0100 = Duration: 36 microseconds

Receiver address: Cisco-Li_f5:c2:c6 (00:18:f8:f5:c2:c6)

Transmitter address: Cisco-Li_74:95:92 (00:25:9c:74:95:92)

Destination address: SurecomT_94:24:7b (00:02:44:94:24:7b)

Source address: Cisco-Li_74:95:92 (00:25:9c:74:95:92)

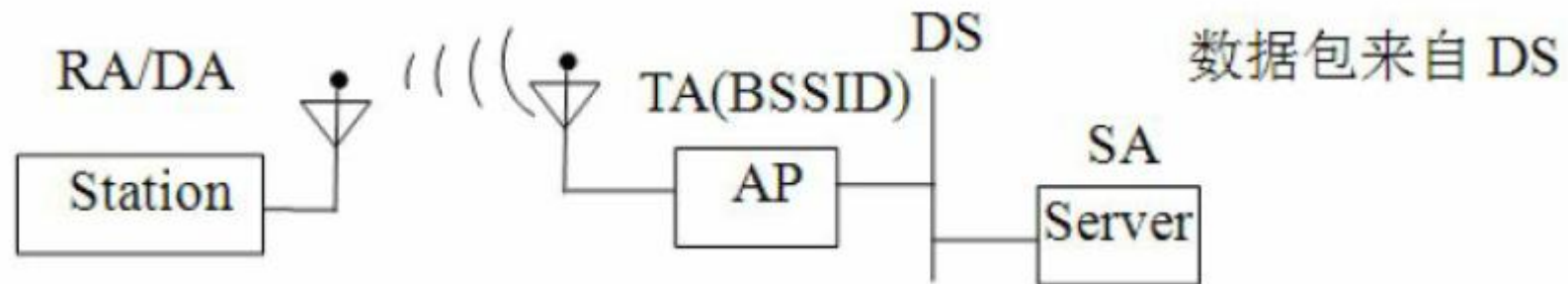
BSS Id: Cisco-Li_f5:c2:c6 (00:18:f8:f5:c2:c6)

STA address: Cisco-Li_74:95:92 (00:25:9c:74:95:92)

.... 0000 = Fragment number: 0

1000 0011 0010 = Sequence number: 2098

地址域: case 2



数据包来自DS的情况下，RA和DA一致，而TA和BSSID一致。

实例分析

IEEE 802.11 Data, Flags: .p....F.C

Type/Subtype: Data (0x0020)

Frame Control Field: 0x0842

.... ..00 = Version: 0

.... 10.. = Type: Data frame (2)

0000 = Subtype: 0

Flags: 0x42

.... ..10 = DS status: Frame from DS to a STA via AP (To DS: 0 From DS: 1) (0x2)

.... .0.. = More Fragments: This is the last fragment

.... 0... = Retry: Frame is not being retransmitted

...0 = PWR MGT: STA will stay up

..0. = More Data: No data buffered

.1.. = Protected flag: Data is protected

0... = Order flag: Not strictly ordered

.000 0000 0010 1100 = Duration: 44 microseconds

Receiver address: Cisco-Li 74:95:92 (00:25:9c:74:95:92)

Transmitter address: Cisco-Li f5:c2:c6 (00:18:f8:f5:c2:c6)

Destination address: Cisco-Li 74:95:92 (00:25:9c:74:95:92)

Source address: SurecomT_94:24:7b (00:02:44:94:24:7b)

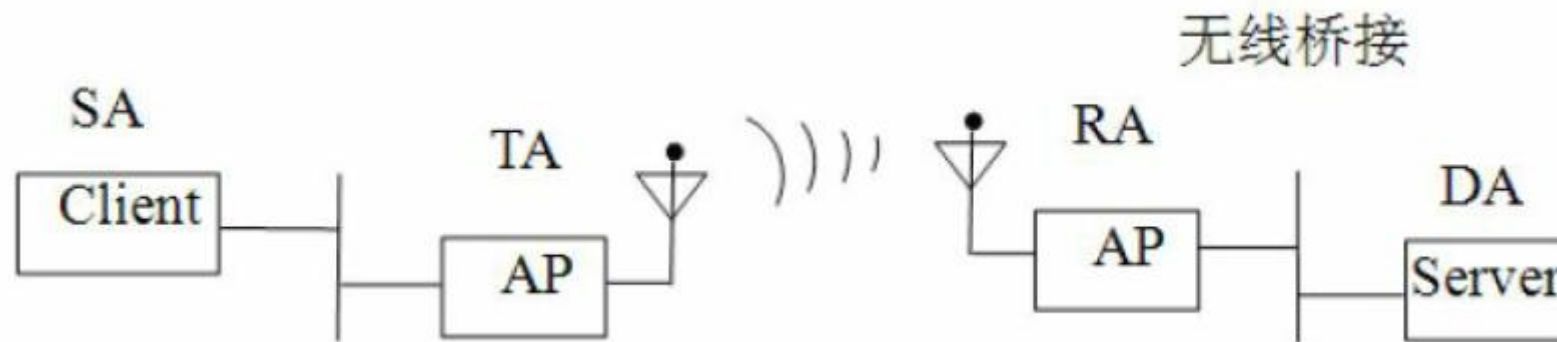
BSS Id: Cisco-Li f5:c2:c6 (00:18:f8:f5:c2:c6)

STA address: Cisco-Li 74:95:92 (00:25:9c:74:95:92)

.... 0000 = Fragment number: 0

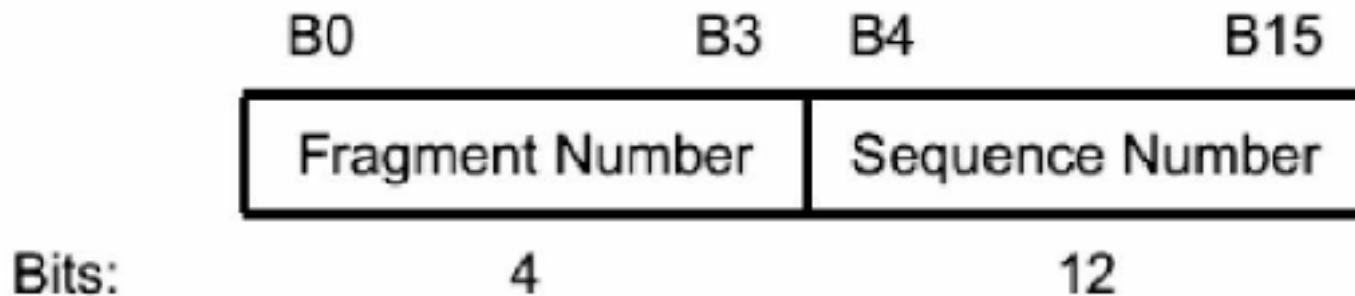
0101 1000 1000 = Sequence number: 1416

地址域: case 3



无线桥接的情况下，SA、TA、RA和DA四种地址域都有效。

Sequence Control



❑ Sequence Control域长16位，前4位表示分片编号FN，后12位为帧顺序编号SN

❑ Sequence Number: STA每次发送数据帧时都会设置一个帧顺序编号，控制帧没有帧顺序编号，重传帧不使用新的帧顺序编号。

❑ Fragment Number: 用于控制分片帧。如果数据量过大，则MAC层会将其分片发送。每个分片帧都有对应的分片编号。

Sequence Number

	Time	Source	Destination	Protocol	Length	Info
25	2.150628	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=3999, FN=0, flags=.....C,
26	2.151616	Cisco-Li_82:b2:55	Spanning-tre...	802.11	118	Data, SN=4000, FN=0, Flags=.p...F.C
27	2.253468	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4001, FN=0, flags=.....C,
28	2.355534	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4002, FN=0, flags=.....C,
29	2.458576	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4003, FN=0, flags=.....C,
30	2.560555	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4004, FN=0, flags=.....C,
31	2.663481	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4005, FN=0, flags=.....C,
32	2.765524	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4006, FN=0, flags=.....C,
33	2.867455	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4007, FN=0, flags=.....C,
34	2.970429	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4008, FN=0, flags=.....C,
35	3.072406	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4009, FN=0, flags=.....C,
36	3.175408	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4010, FN=0, flags=.....C,
37	3.277398	Cisco-Li_82:b2:55	Broadcast	802.11	168	Beacon frame SN=4011, FN=0, flags=.....C,

帧格式实例分析

Frame Control: Data frame, from STA to DS (to AP)	Duration	Receiver address (MAC of AP)	Transmitter address (MAC of source STA)
08 01	30 00	e4 ce 8f 66 b2 42	e4 ce 8f 5b a1 f6
Destination address (MAC of dest. STA)	e4 ce 8f 5a 0c 5e	f0 00	aa aa 03 00 00 00 08 00
Sequence control	45 00 00 37 59 33 40 00	40 06 60 1a c0 a8 00 10	c0 a8 00 13 e0 1c 11 5c f4 6d 68 b2 cf a7 ee 49
	80 18 00 e5 2d eb 00 00	01 01 08 0a 00 00 33 f5	
	00 00 33 85 48 69 0a	Frame body	

控制帧、管理帧、数据帧

802.11 控制帧

控制帧得名于媒体访问控制（Media Access Control, MAC），用来控制对通信媒体的访问。控制帧通常与数据帧搭配使用，负责区域的清空、信道的取得以及载波监听的维护，并于收到数据时予以的应答，借此促进工作站间数据传输的可靠性。

802.11控制帧

□ 控制帧的作用包括协助数据帧的传递、管理无线媒介的访问等，简介如下4种。

◆ RTS (Request To Send)

◆ CTS (Clear To Send)

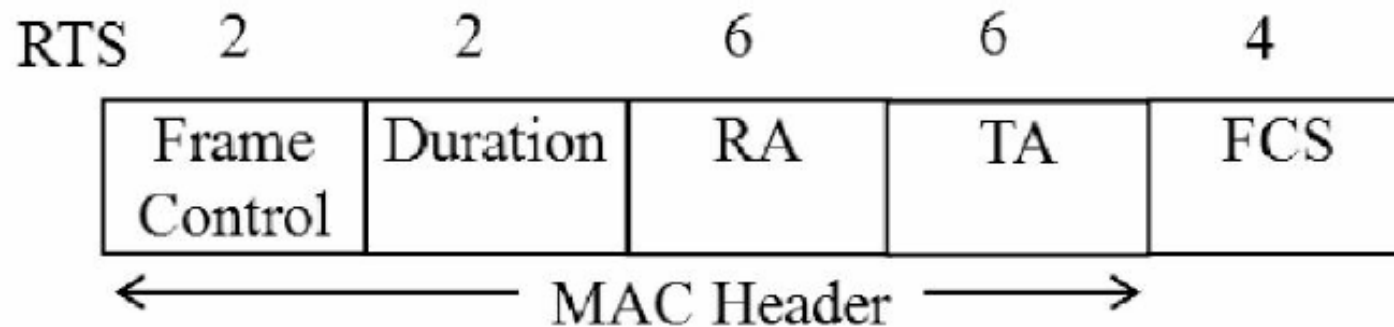
◆ ACK

◆ PS-POLL

控制帧都和CSMA/CA有关，无线网络是冲突避免，以太网是冲突检测。

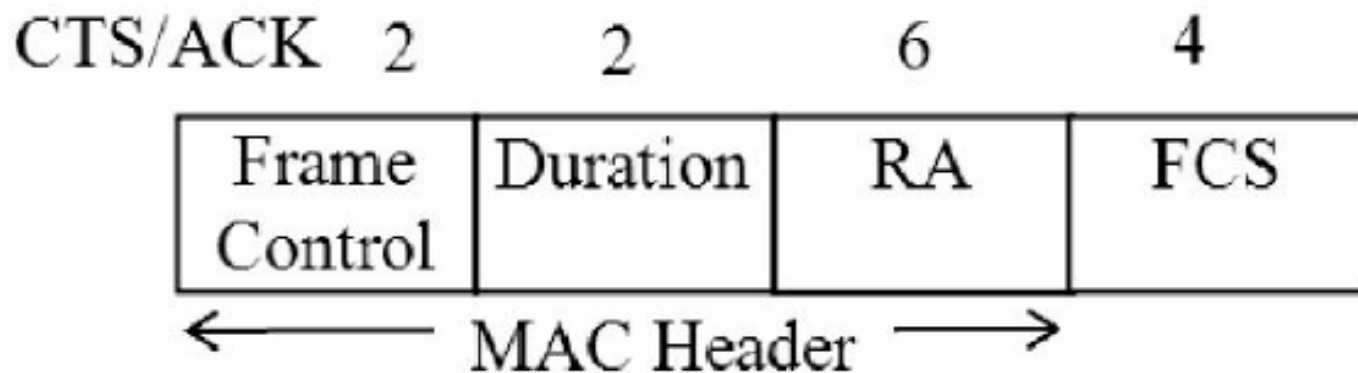
控制帧还包括CF-End、CF-End+CF-Ack等

802.11控制帧：RTS



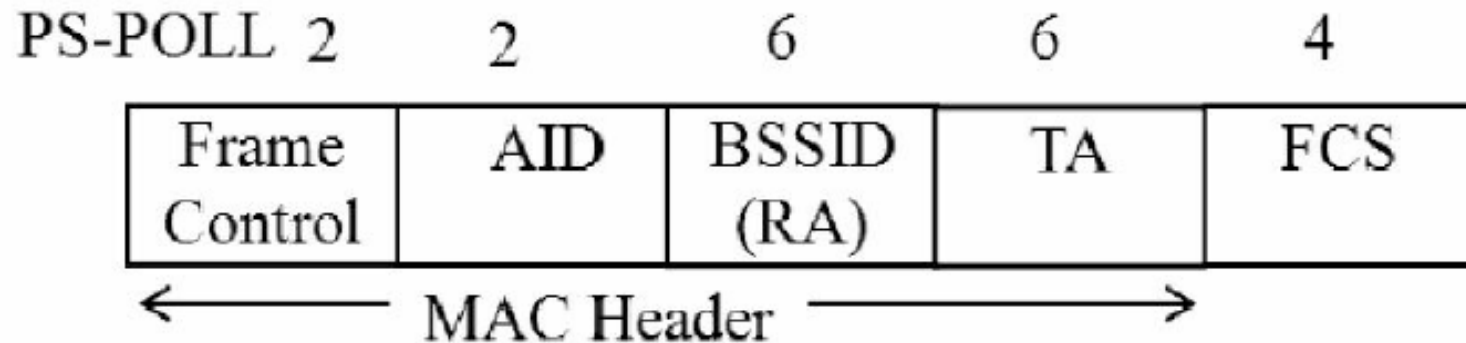
- RTS (Request To Send) : RTS用于申请无线媒介的使用时间，值为Duration，单位为微秒。

802.11控制帧：CTS/ACK帧



- CTS (Clear To Send) : 用于回复RTS帧。另外它被802.11g保护机制用来避免干扰旧的STA
- ACK: 802.11中, MAC以及任何数据的传输都需要得到确认。这些数据包括普通的数据传输、RTS/CTS交换之前帧以及分片帧。

802.11控制帧： PS-POLL



- PS-POLL: 该控制帧被STA用于从AP中获取因省电模式而缓存的数据。当一部移动工作站从省电模式中苏醒，便会发送一个 PS-Poll 帧给基站，以取得任何暂存帧。其中AID的值是STA和AP关联时，由AP赋给该STA的。

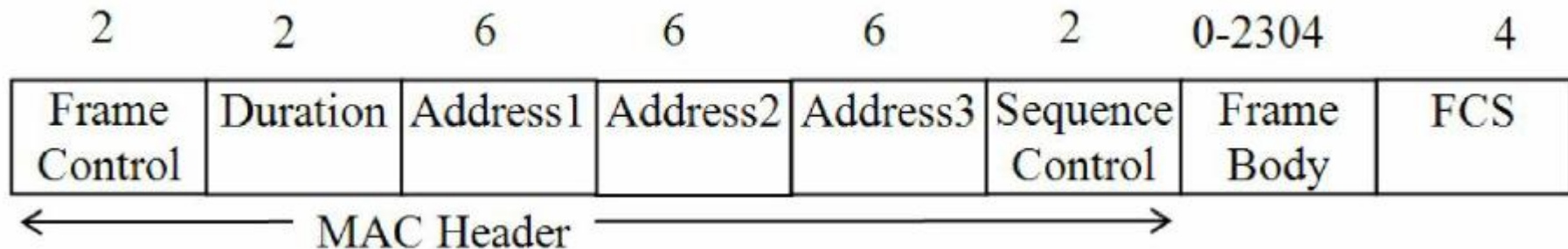
802.11 管理帧

用于管理无线网络，如节点的加入和退出无线网络等。

802.11管理帧

- 管理帧主要用于无线网络的管理，主要包括：
 - ◆ Beacon（信标）帧
 - ◆ Association Request/Response（关联请求/回复）帧
 - ◆ Probe Request/Response（探测请求/回复）帧
 - ◆ Authentication/Deauthentication（认证/取消认证）帧。

802.11管理帧格式



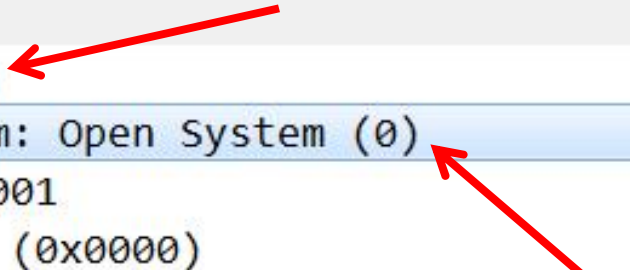
- 管理帧包括 MAC Header （6个字段），Frame Body和FCS，其中Frame Body携带具体的管理信息数据。
- 管理信息数据包括：
 - 定长字段（Fixed Field）
 - 信息元素（Information Element）

802.11管理帧：定长字段（1）

- Authentication Algorithm Number: 2个 byte, 用于说明认证过程中所使用的认证类型
 - ◆ 0: 代表开放系统身份认证 (Open System Authentication)。
 - ◆ 1: 代表共享密钥身份认证 (Shared Key Authentication)。
 - ◆ 2: 代表快速BSS切换 (Fast BSS Transition)。
 - ◆ 3: 代表SAE (Simultaneous Authentication of Equals)。用于两个STA互相认证的方法, 常用于Mesh BSS网络。
 - ◆ 65535: 代表厂商自定义算法。

Fixed Parameters: 示例

```
▷ Frame 78: 58 bytes on wire (464 bits), 58 bytes captured (464 bits)
▷ Radiotap Header v0, Length 24
  ▲ 802.11 radio information
    PHY type: 802.11b (4)
    Short preamble: False
    Data rate: 1.0 Mb/s
    Channel: 1
    Frequency: 2412MHz
    ▷ [Duration: 464µs]
  ▷ IEEE 802.11 Authentication, Flags: .....C
  ▲ IEEE 802.11 wireless LAN
    ▲ Fixed parameters (6 bytes)
      Authentication Algorithm: Open System (0)
      Authentication SEQ: 0x0001
      Status code: Successful (0x0000)
```



802.11管理帧：定长字段（2）

- Beacon Interval field: 该字段占2字节。每隔一段时间AP就会发出Beacon信号用来宣布无线网络的存在。该信号包含了BSS参数等重要信息。所以STA必须要监听Beacon信号。
- Beacon Interval field字段用来表示Beacon信号之间间隔的时间，其单位为Time Units（规范中缩写为TU。注意，一个TU为1024微秒。这里采用2作为基数进行计算）。一般该字段会设置为100个TU。

Fixed Parameters: 示例

- ▷ Frame 75: 168 bytes on wire (1344 bits), 168 bytes captured (1344 bits)
- ▷ Radiotap Header v0, Length 24
- ▷ 802.11 radio information
- ▷ IEEE 802.11 Beacon frame, Flags:C
- ▲ IEEE 802.11 wireless LAN
 - ▲ Fixed parameters (12 bytes)
 - Timestamp: 0x000000011c27c18b
 - Beacon Interval: 0.102400 [Seconds]
 - ▷ Capabilities Information: 0x0411
 - ▲ Tagged parameters (104 bytes)
 - ▲ Tag: SSID parameter set: Coherer
 - Tag Number: SSID parameter set (0)
 - Tag length: 7
 - SSID: Coherer
 - ▲ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 18, 24, 36, 54, [Mbit/sec]
 - Tag Number: Supported Rates (1)

802.11管理帧：定长字段（3）

- ▣ Capability Information（性能信息）：该字段长2字节，一般通过 Beacon 帧、Probe Request和Response帧携带它。该字段用于宣告此网络具备何种功能。2字节中的每一位（共16位）都用来表示网络是否拥有某项功能

Fixed Parameters: 示例

- ▷ IEEE 802.11 Beacon frame, Flags:C
- ▲ IEEE 802.11 wireless LAN
 - ▲ Fixed parameters (12 bytes)
 - Timestamp: 0x000000011c27c18b
 - Beacon Interval: 0.102400 [Seconds]
 - ▲ Capabilities Information: 0x0411
 -1 = ESS capabilities: Transmitter is an AP
 -0. = IBSS status: Transmitter belongs to a BSS
 -0. 00.. = CFP participation capabilities: No point coordinator at AP (0x00)
 -1 = Privacy: AP/STA can support WEP
 -0. = Short Preamble: Not Allowed
 -0.. = PBCC: Not Allowed
 - 0... = Channel Agility: Not in use
 -0 = Spectrum Management: Not Implemented
 -1.. = Short Slot Time: In use
 - 0... = Automatic Power Save Delivery: Not Implemented
 - ...0 = Radio Measurement: Not Implemented
 - ..0. = DSSS-OFDM: Not Allowed
 - .0.. = Delayed Block Ack: Not Implemented
 - 0... = Immediate Block Ack: Not Implemented
 - ▲ Tagged parameters (104 bytes)

802.11常用管理帧： Beacon帧

- AP通过定时发送Beacon帧来声明某个无线网络， STA通过接收到的Beacon帧来感知当前存在的无线网络。 Beacon帧就是某个无线网络的心跳帧， 主要携带如下信息： Timestamp、 Beacon Interval、 Capability、 SSID

Beacon帧示例

IEEE 802.11 wireless LAN

Fixed parameters (12 bytes)

Timestamp: 0x00000011bf75187

Beacon Interval: 0.102400 [Seconds]

Capabilities Information: 0x0411

.... 1 = ESS capabilities: Transmitter is an AP

.... 0 = IBSS status: Transmitter belongs to a BSS

.... 00.. = CFP participation capabilities: No point coordinator at AP (0x00)

.... 1 = Privacy: AP/STA can support WEP

.... 0 = Short Preamble: Not Allowed

.... 0 = PBCC: Not Allowed

.... 0 = Channel Agility: Not in use

.... 0 = Spectrum Management: Not Implemented

.... 1 = Short Slot Time: In use

.... 0 = Automatic Power Save Delivery: Not Implemented

.... 0 = Radio Measurement: Not Implemented

.... 0 = DSSS-OFDM: Not Allowed

.... 0 = Delayed Block Ack: Not Implemented

.... 0 = Immediate Block Ack: Not Implemented

Tagged parameters (104 bytes)

▷ Tag: SSID parameter set: Coherer

▷ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 18, 24, 36, 54, [Mbit/sec]

802.11常用管理帧：Probe Request/Response帧

- STA用Probe Request帧来搜索周围的无线网络，包括的信息：SSID、Supported Rates、Extended Supported Rates.
- AP收到Probe Request帧后，会以Probe Response帧进行响应，该帧携带的信息和Beacon帧类似.

Probe Request示例

- ▷ IEEE 802.11 Probe Request, Flags:C
 - ▲ IEEE 802.11 wireless LAN
 - ▲ Tagged parameters (25 bytes)
 - ▲ Tag: SSID parameter set: Coherer
Tag Number: SSID parameter set (0)
Tag length: 7
SSID: Coherer
 - ▲ Tag: Supported Rates 1, 2, 5.5, 11, 18, 24, 36, 54, [Mbit/sec]
Tag Number: Supported Rates (1)
Tag length: 8
Supported Rates: 1 (0x02)
Supported Rates: 2 (0x04)
Supported Rates: 5.5 (0x0b)
Supported Rates: 11 (0x16)
Supported Rates: 18 (0x24)
Supported Rates: 24 (0x30)
Supported Rates: 36 (0x48)
Supported Rates: 54 (0x6c)
 - ▲ Tag: Extended Supported Rates 6, 9, 12, 48, [Mbit/sec]
Tag Number: Extended Supported Rates (50)
-

Probe Response示例

- ▷ IEEE 802.11 Probe Response, Flags:C
- ▲ IEEE 802.11 wireless LAN
 - ▲ Fixed parameters (12 bytes)
 - Timestamp: 0x000000011c23ff61
 - Beacon Interval: 0.102400 [Seconds]
 - ▲ Capabilities Information: 0x0411
 - 1 = ESS capabilities: Transmitter is an AP
 - 0. = IBSS status: Transmitter belongs to a BSS
 -0. 00.. = CFP participation capabilities: No point coordinator at AP (0x00)
 - 1 = Privacy: AP/STA can support WEP
 -0. = Short Preamble: Not Allowed
 -0.. = PBCC: Not Allowed
 - 0... = Channel Agility: Not in use
 -0 = Spectrum Management: Not Implemented
 -1.. = Short Slot Time: In use
 - 0... = Automatic Power Save Delivery: Not Implemented
 - ...0 = Radio Measurement: Not Implemented
 - ..0. = DSSS-OFDM: Not Allowed
 - .0.. = Delayed Block Ack: Not Implemented
 - 0... = Immediate Block Ack: Not Implemented
 - ▲ Tagged parameters (98 bytes)
 - ▲ Tag: SSID parameter set: Coherer

802.11常用管理帧：Association Request帧

- 当STA要关联某个AP时，发送Association Request帧。该帧携带的主要信息如下：
 - ◆ Capability: AP将检查该字段判断STA是否满足要求
 - ◆ Listen Interval: AP将根据该值分配PS时所需的缓冲区
 - ◆ SSID: AP将检查SSID是否为自己所在网络
 - ◆ Supported Rates: AP将检查该字段是否满足要求

Association Request示例

- ▷ IEEE 802.11 Association Request, Flags:C
- ▲ IEEE 802.11 wireless LAN
 - ▲ Fixed parameters (4 bytes)
 - ▲ Capabilities Information: 0x0431
 -1 = ESS capabilities: Transmitter is an AP
 -0. = IBSS status: Transmitter belongs to a BSS
 -0. 00.. = CFP participation capabilities: No point coordinator at AP (0x00)
 -1 = Privacy: AP/STA can support WEP
 -1. = Short Preamble: Allowed
 -0.. = PBCC: Not Allowed
 - 0... = Channel Agility: Not in use
 -0 = Spectrum Management: Not Implemented
 -1.. = Short Slot Time: In use
 - 0... = Automatic Power Save Delivery: Not Implemented
 - ...0 = Radio Measurement: Not Implemented
 - ..0. = DSSS-OFDM: Not Allowed
 - .0.. = Delayed Block Ack: Not Implemented
 - 0... = Immediate Block Ack: Not Implemented
 - Listen Interval: 0x000a
 - ▲ Tagged parameters (47 bytes)
 - ▲ Tag: SSID parameter set: Coherer
 - Tag Number: SSID parameter set (0)
 - Tag length: 7

802.11常用管理帧：Association Response帧

- ▣ 针对Association Request帧，AP会回复一个Association Response帧来通知关联请求的处理结果，主要包括如下信息：
 - ◆ Capability: AP设置的Capability
 - ◆ Status Code: AP返回的关联请求处理结果
 - ◆ AID: AP返回关联ID给STA
 - ◆ Supported Rates: AP支持的传输速率

802.11管理帧： authentication帧

□ Authentication帧用于进行身份认证，主要包括如下信息：

- ◆ Authentication Algorithm Number: 认证算法类型
- ◆ Authentication Transaction Sequence Number: 认证过程可能需要好几次帧交换，所以每个帧都有自己的编号
- ◆ Status Code: 有些类型的认证会使用该值返回结果
- ◆ Challenge Text: 有些类型的认证会使用该字段

Authentication帧示例

- ▷ IEEE 802.11 Authentication, Flags:C
- ▲ IEEE 802.11 wireless LAN
 - ▲ Fixed parameters (6 bytes)
 - Authentication Algorithm: Open System (0)
 - Authentication SEQ: 0x0002
 - Status code: Successful (0x0000)
 - ▲ Tagged parameters (8 bytes)
 - ▲ Tag: Vendor Specific: Broadcom
 - Tag Number: Vendor Specific (221)
 - Tag length: 6
 - OUI: 00:10:18 (Broadcom)
 - Vendor Specific OUI Type: 2
 - Vendor Specific Data: 020004

802.11管理帧：总结

- ▣ 802.11规范里面共定义了15种管理帧，携带的信息很复杂，其中定长字段有42种，信息元素有120种。

802.11 数据帧

用来携带上层协议数据（如IP数据包），负责在工作站之间传输数据。

802.11数据帧

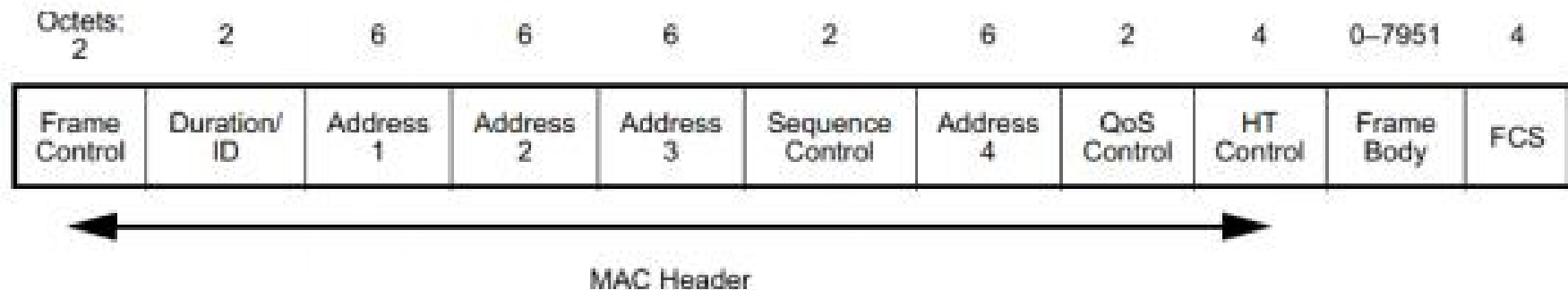


图 3-24 数据帧格式

其中，QoS Control和HT (High Throughput) Control字段只在QoS和HT的情况下出现

WLAN安全

WLAN的安全发展过程

- WEP (Wired Equivalent Privacy) , 即有线等效保密, 目的是达到和有线网络相同的安全性。
- WPA (Wi-Fi Protected Access) , 实现了802.11i草案的一个子集, 只需要更新固件, 不需要更新硬件即可实现
- WPA2 (Wi-Fi Protected Access II) , 实现了802.11i规范
- WPA3 (Wi-Fi Protected Access III) , 更强的安全算法, GCMP, ECDH,

WEP

Wired Equivalent Privacy

有线等效保密

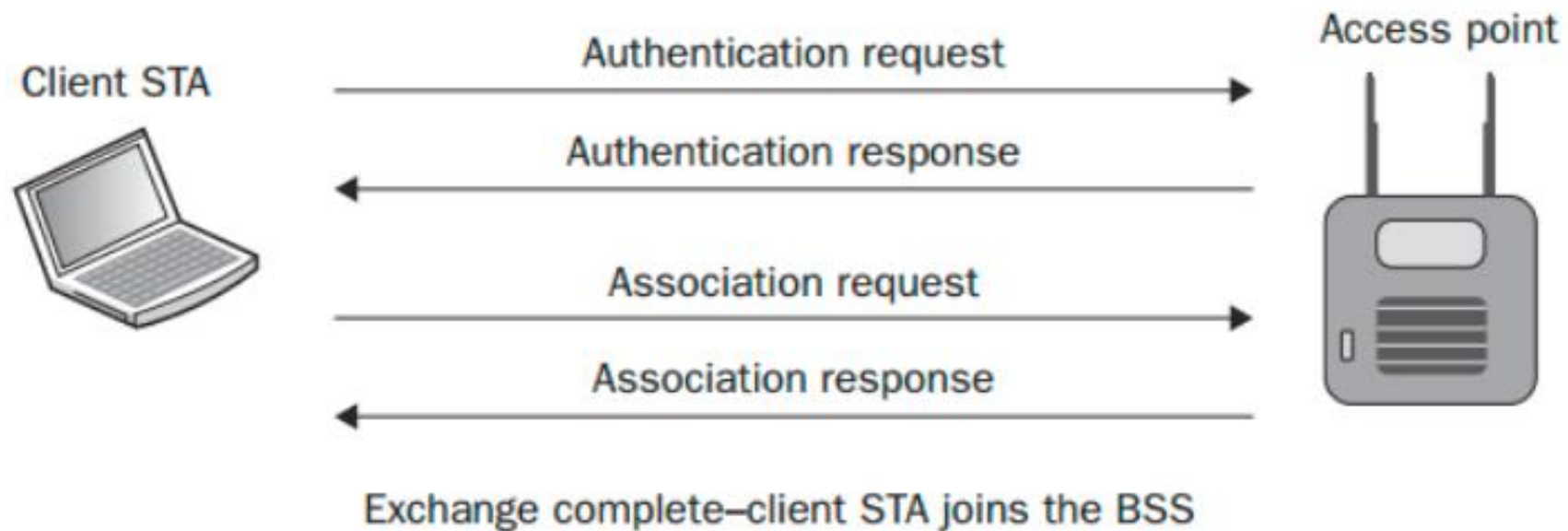
WEP的安全服务

- 身份认证
- 保密性
- 完整性

WEP: 身份认证

- 开放系统认证
- 基于PSK的身份认证

WEP: 开放系统认证



WEP: PSK身份认证

Static WEP key =
0123456789



Client STA

Client station sends an authentication request frame



Static WEP key =
0123456789



AP

Access point sends a cleartext challenge to the client station in an authentication response frame



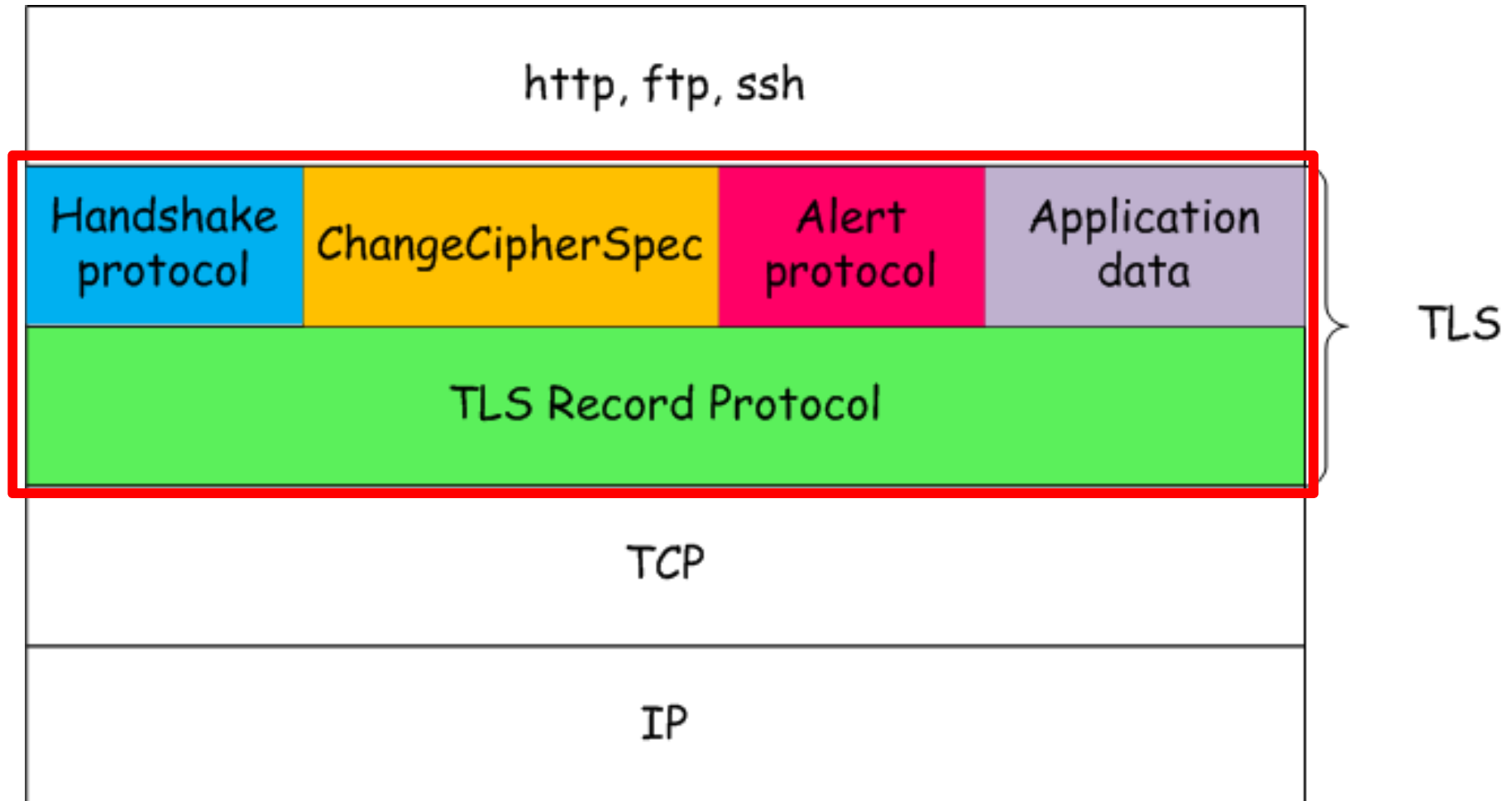
Client station encrypts the cleartext challenge and sends it back to the access point in another authentication request frame



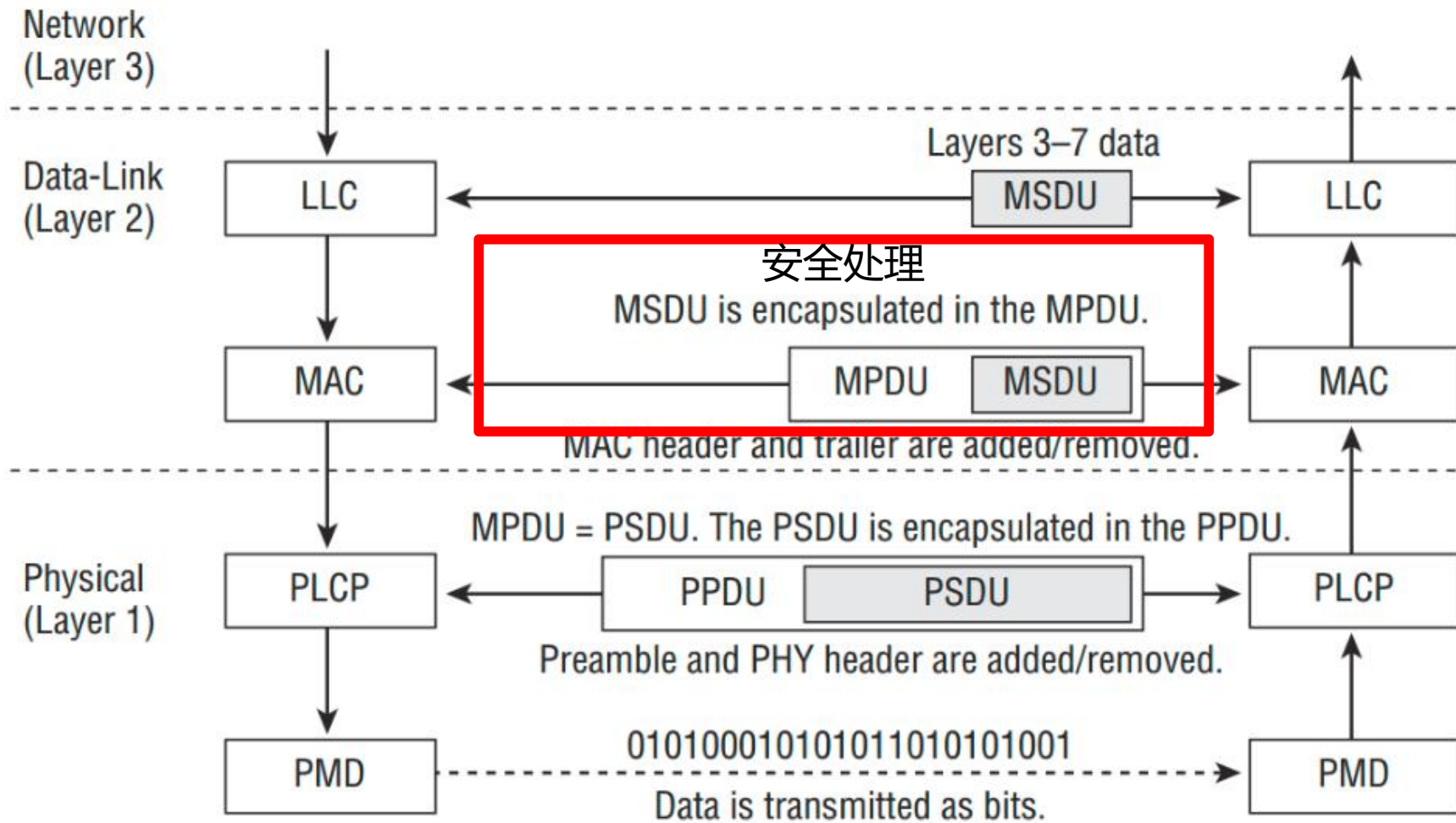
If the access point is able to decrypt the frame, and it matches the challenge text, it will reply with an authentication frame indicating that the authentication is successful



回顾TLS的安全处理



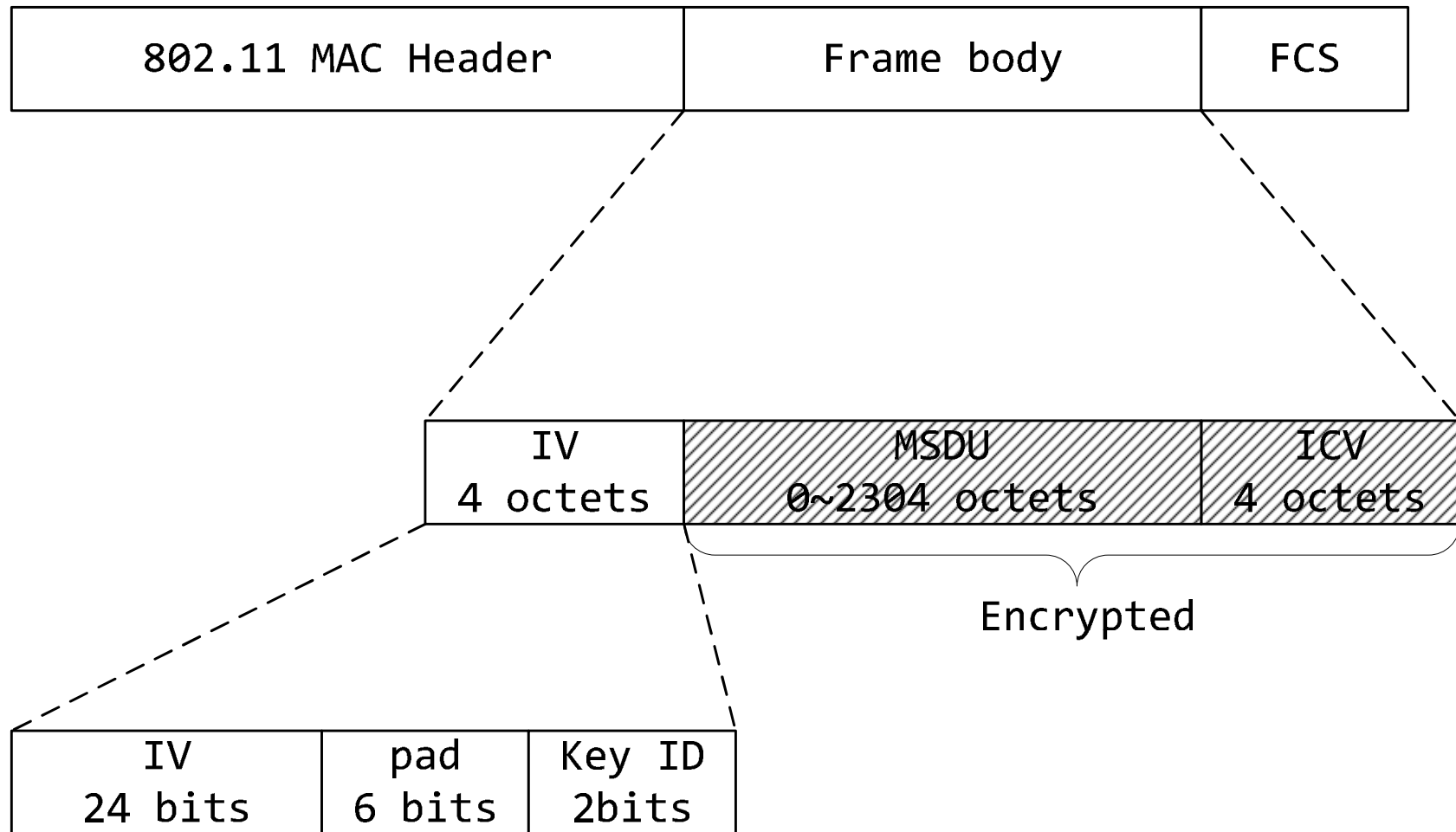
WLAN的安全处理功能



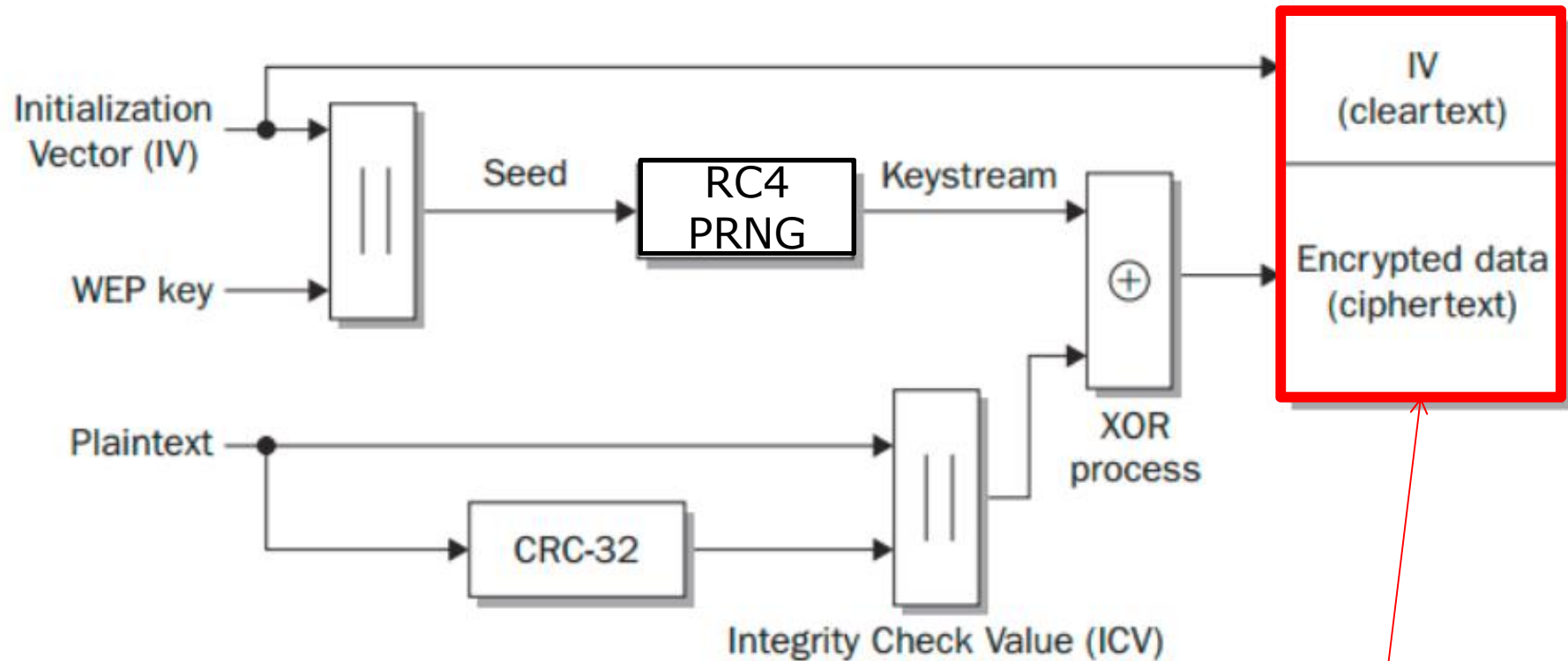
WEP加密封装

- 仅对数据帧进行加密
- 无MSDU的帧无需加密，如：
 - ◆ 管理帧
 - ◆ 控制帧
 - ◆ 空帧（无数据字段）

加密后的MPDU格式



WEP加密过程



Frame body

WEP加密流程

1. 生成24bit随机数作为IV，与WEP Key组合作为RC4算法的输入，产生密钥流；IV随数据帧以明文方式发送。
2. 对需要加密的数据应用CRC-32生成ICV，追加该ICV到明文数据后面。
3. 将2的数据与1的密钥流做XOR运算生成密文。
4. 将IV添加到密文前面作为Frame body封装成帧并发送。

WEP Key

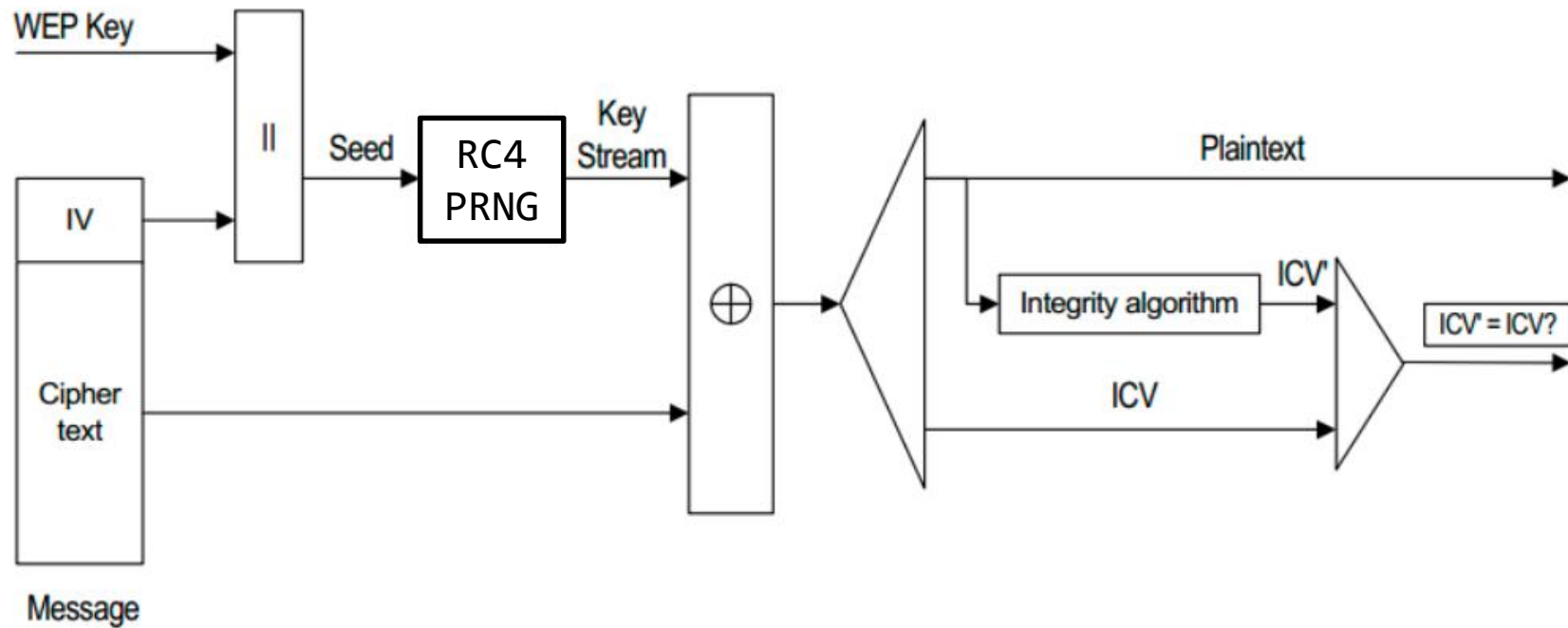
WEP-40	24-bit IV	40-bit static Key
WEP-104	24-bit IV	104-bit static Key

802.11标准定义了两个WEP版本：

◆ WEP-40, 24-bit IV || 40-bit static Key (10 HEX characters or 5 ASCII characters)构成64bit ARC4的种子, 生成密钥流

◆ WEP-104, 24-bit IV || 104-bit static Key (26 HEX characters or 13 ASCII characters)构成128-bit ARC4的种子, 生成密钥流

WEP解密过程



WEP解密流程

1. 提取Frame body中的IV，与WEP Key组合作为RC4算法的输入，产生密钥流。
2. 将密钥流与密文做XOR运算得到数据明文和ICV。
3. 对数据明文计算ICV，并与2中的ICV进行对比以确定数据完整性。

WEP: weakness

- RC4: FMS attack

- ◆ Scott Fluhrer, Itsik Mantin, and Adi Shamir

- ◆ Attackers can recover the RC4 key after eavesdropping on the network.

- 同一网络下的STA可相互窃听流量

- ◆ 相同的WEP Key

- ◆ 明文传输 IV

-

WEP不安全

WPA

WPA2

WPA3

WPA

(Wireless Protected Access)

无线安全设置界面

☐ 不开启无线安全

☒ WPA-PSK/WPA2-PSK

认证类型: 自动

加密算法: AES

PSK密码:

(8-63个ASCII码字符或8-64个十六进制字符)

组密钥更新周期: 86400

(单位为秒, 最小值为30, 不更新则为0)

☐ WPA/WPA2

认证类型: 自动

加密算法: 自动

Radius服务器IP:

Radius端口: 1812 (1-65535, 0表示默认端口: 1812)

Radius密码:

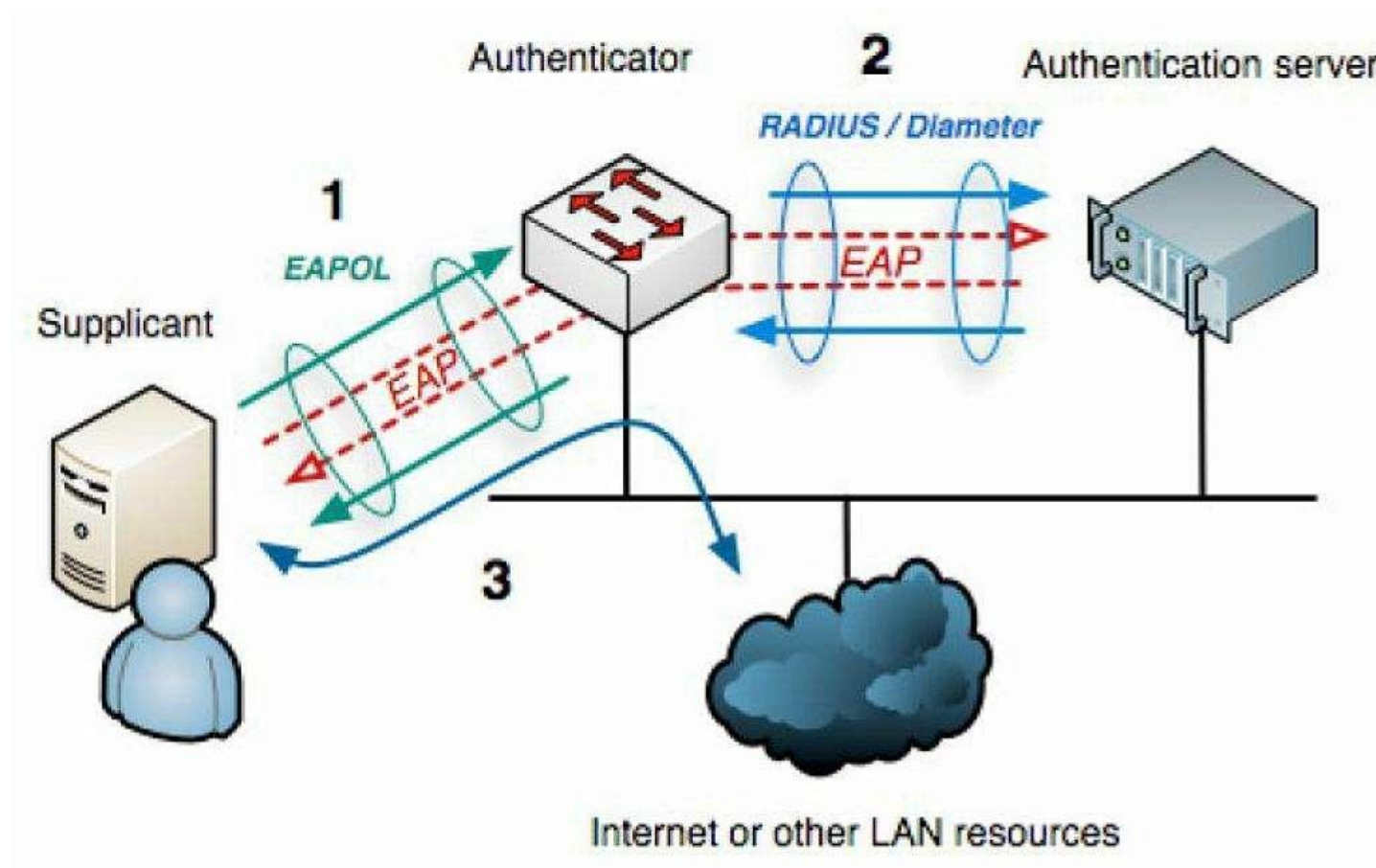
组密钥更新周期: 86400

(单位为秒, 最小值为30, 不更新则为0)

☐ WEP

认证类型: 自动

EAP架构



RADIUS: Remote Authentication Dial In User Service, 远程用户拨号认证系统

802.11i运行过程

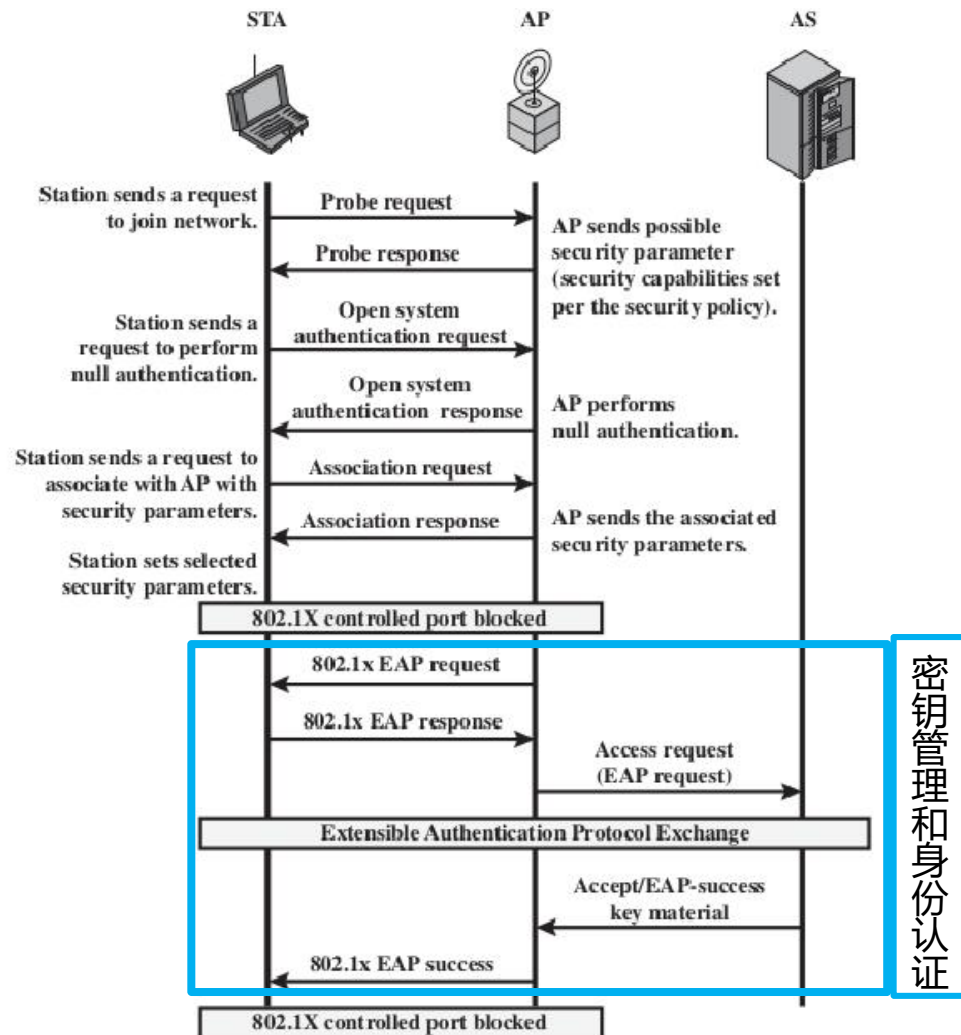


Figure 17.6 IEEE 802.11i Phases of Operation: Capability Discovery, Authentication, and Association

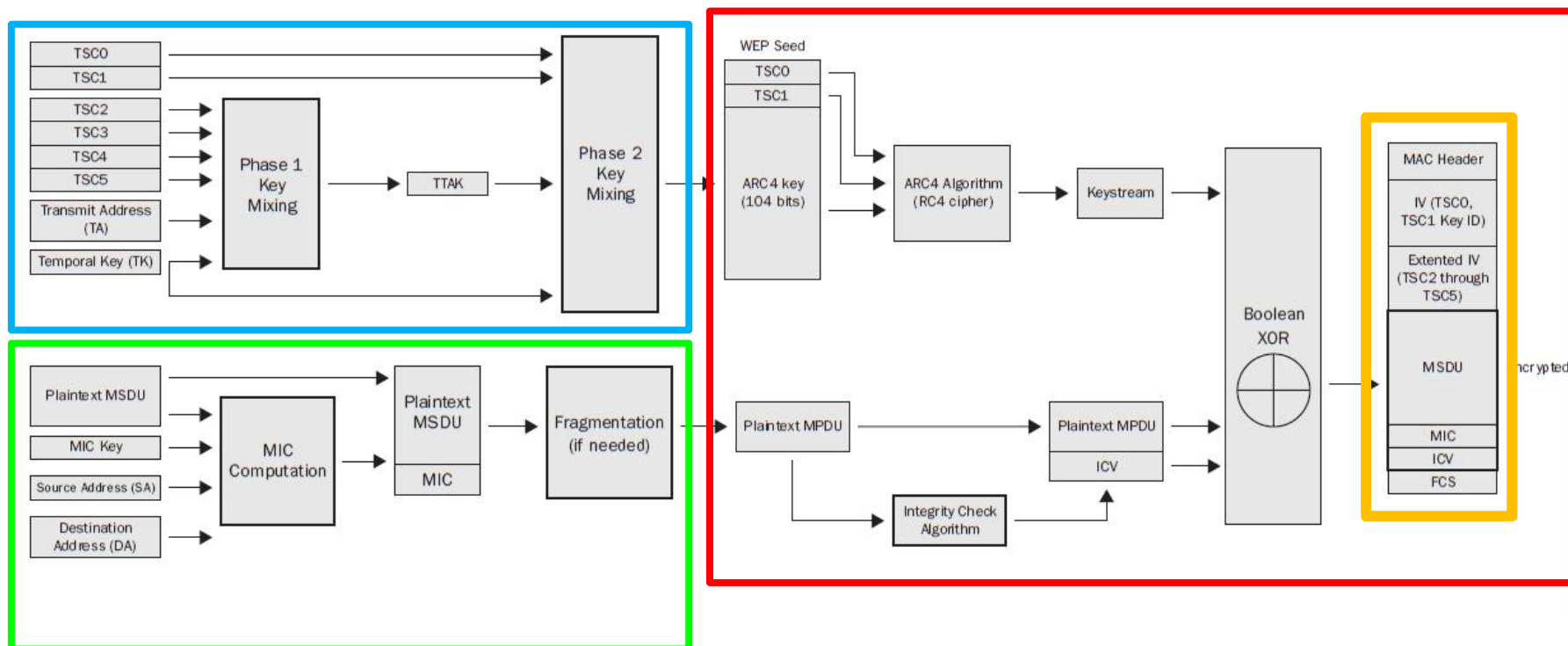
EAPOL示例

No.	Time	Source	Destination	Protocol	Length	Info
86	5.648961		Cisco-Li_82:b...	802.11	38	Clear-to-send, Flags=.....C
87	5.649953	Cisco-Li...	Apple_82:36:3a	EAPOL	181	Key (Message 1 of 4)
88	5.649964		Cisco-Li_82:b...	802.11	38	Acknowledgement, Flags=.....C
89	5.650959	Apple_82...	Cisco-Li_82:b...	EAPOL	181	Key (Message 2 of 4)
90	5.650970		Apple_82:36:3...	802.11	38	Acknowledgement, Flags=.....C
91	5.654947		Cisco-Li_82:b...	802.11	38	Clear-to-send, Flags=.....C
92	5.655957	Cisco-Li...	Apple_82:36:3a	EAPOL	239	Key (Message 3 of 4)
93	5.655968		Cisco-Li_82:b...	802.11	38	Acknowledgement, Flags=.....C
94	5.655973	Apple_82...	Cisco-Li_82:b...	EAPOL	159	Key (Message 4 of 4)
95	5.656951		Apple_82:36:3...	802.11	38	Acknowledgement, Flags=.....C
96	5.734961	Cisco-Li...	Broadcast	802.11	168	Beacon frame, SN=4045, FN=0, Flags=....
97	5.837942	Cisco-Li...	Broadcast	802.11	168	Beacon frame, SN=4046, FN=0, Flags=....
98	5.842998		Apple 82:36:3...	802.11	38	Clear-to-send, Flags=.....C

WPA: TKIP

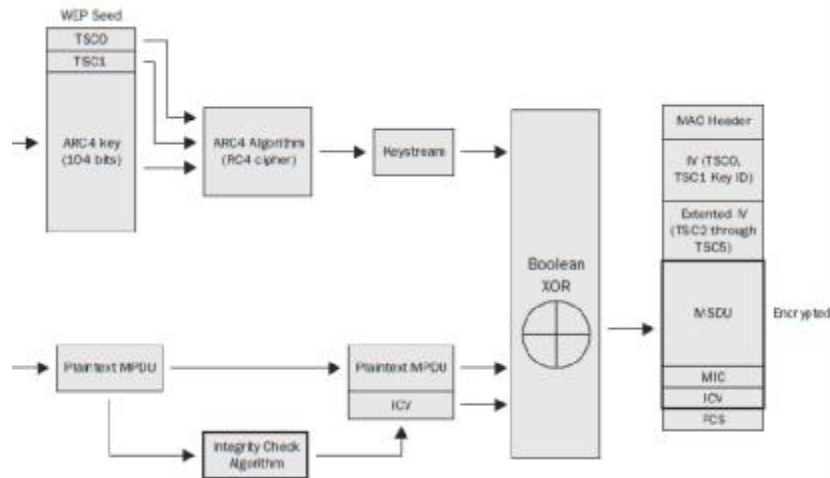
- 基于802.11i draft
- 对WEP的改进
 - ◆ 只需更新固件

TKIP Encryption & Data Integrity Process



可划分为三部分

TKIP加密过程



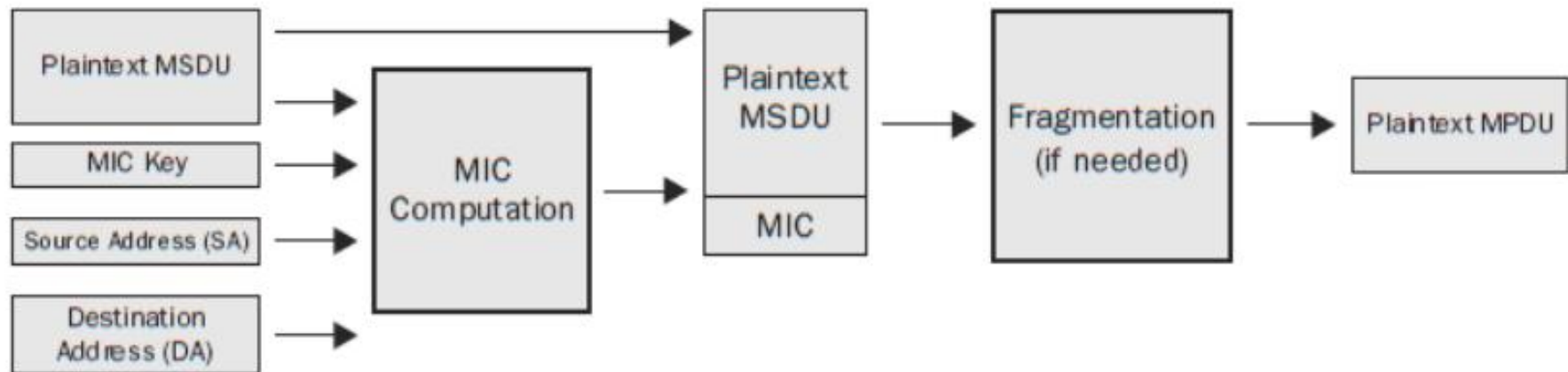
对照WEP，改变的地方包括：

1、WEP Seed的生成

2、Plaintext MPDU

3、帧封装。WEP在MAC Header后面紧随4-octet的IV字段，然后是加密的MSDU||ICV；TKIP的MAC Header后面紧随8个octet的(IV||Extended IV)，然后是加密的MSDU||MIC||ICV。

TKIP: Plaintext MPDU生成



输入:

- Plaintext MSDU: 未加密的MSDU
- MIC Key: 它是从TK中取出来的指定64位
- SA (source address) : 指发送端的MAC地址
- DA(destination address):指接收端的MAC地址

MIC计算

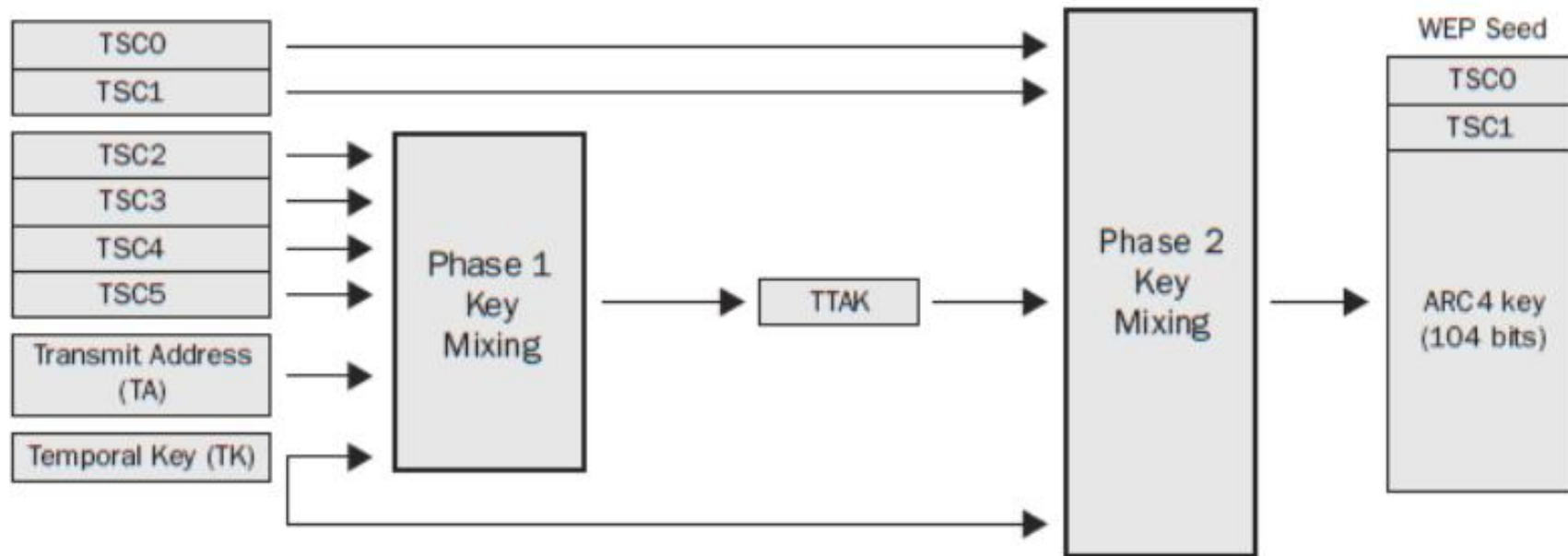
TKIP 使用 称为 Michael 算法的 Keyed Hash function来生成MIC。Michael的输入为64比特key和任意长度的消息，输出为64比特的Michael值。

MIC Key:

如果是AP发送给STA，则为TK的128 - 191比特

如果是STA发送给AP，则为TK的192 - 255比特

TKIP: WEP Seed生成



TSC0-TSC5: TSC0-TSC5 分别代表 TSC (TKIP Sequence Counter) 的每个字节。TSC 是 TKIP 中的一个计数生成器，它会为每一个 MPDU 递增的生成一个 6 字节的 TSC 序列号，用于抗重放攻击。

TK (Temporal Key) : 临时密钥，它是从 PTK 或者 GTK 派生而来的

TA (Transmitter Address) : 一般是指 AP 的 MAC 地址

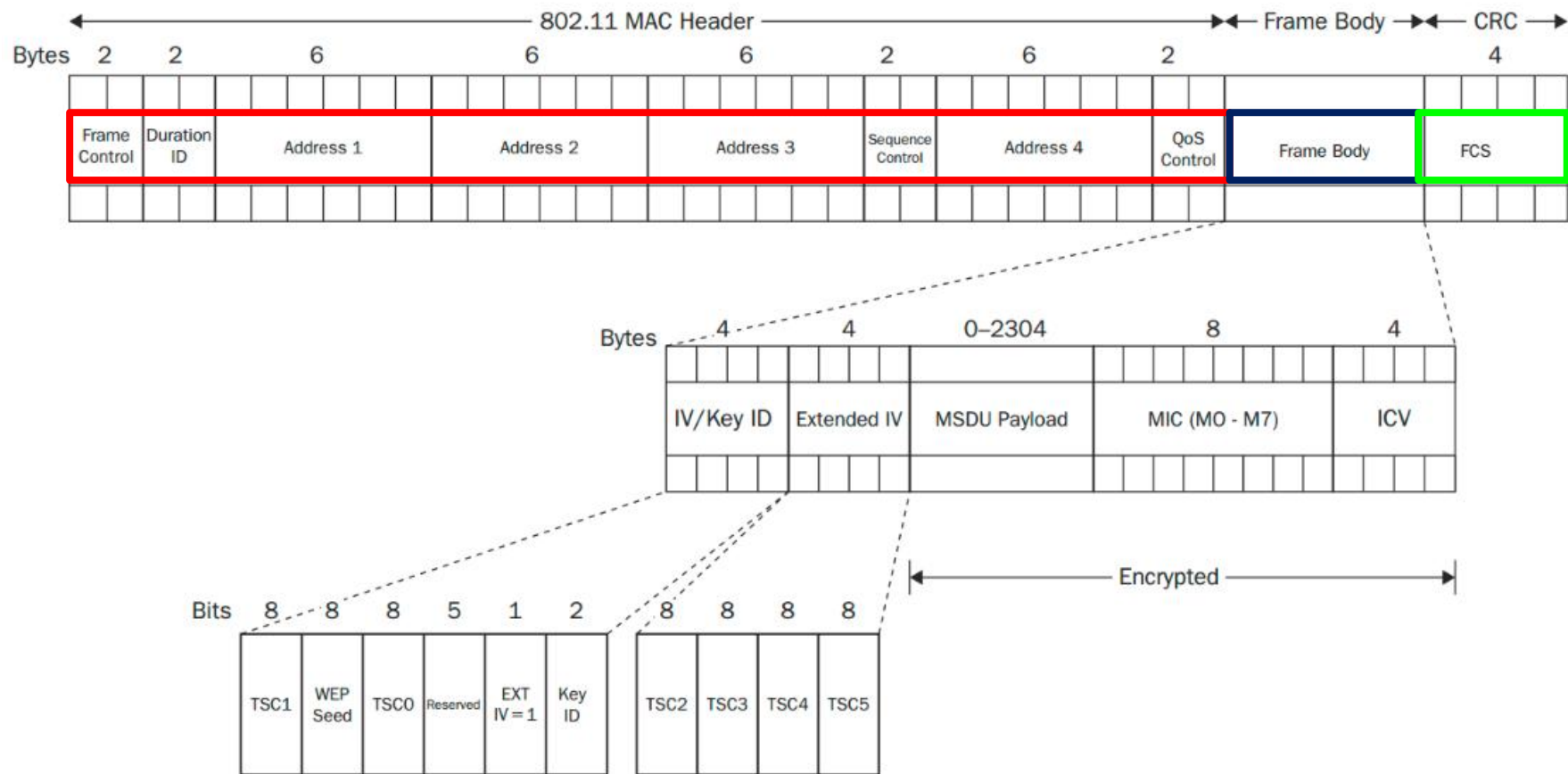
TKIP: WEP Seed生成

TKIP在加密数据时，采用密钥混合的方式来产生加密密钥。为了降低开销，混合过程分为两个阶段。

第一阶段生成 TKIP-mixed Transmit Address and Key (TTAK)，表示为：TTAK = phase1 (TK, TA, TSC)，其中 TK 是 128bit，TA，和 TSC 的 TSC2~TSC5 字节。第一阶段的输出在整个会话过程中保持不变。

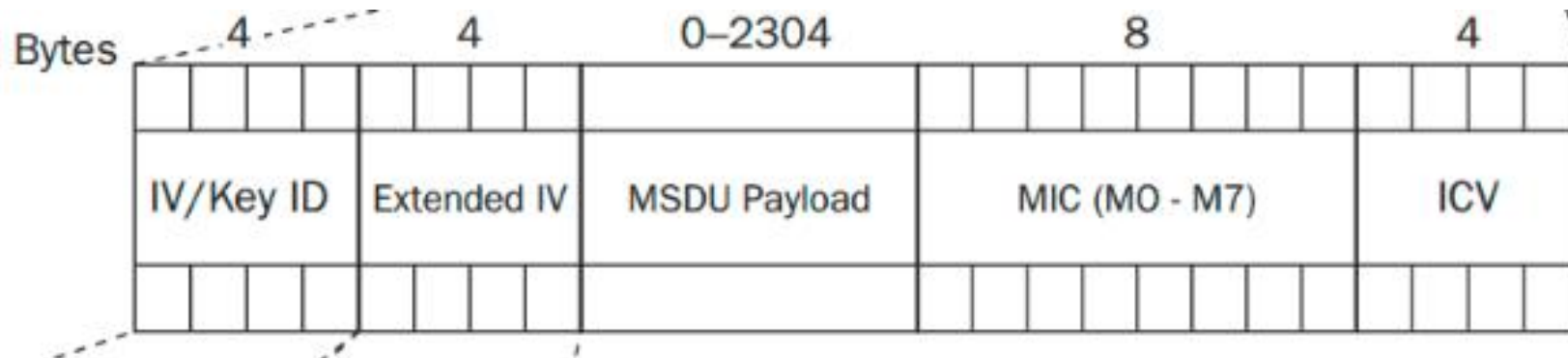
第二阶段，对每个包都需要执行，把 IV 和阶段 1 的结果进行混合，生成 WEP seed = phase2(TTAK, TK, TSC)，128bit，其中又分为 IV 和 104bit 的 WEP key。

TKIP加密处理后的MPDU



安全处理后的帧也是由：
MAC Header、Frame Body、FCS三部分组成

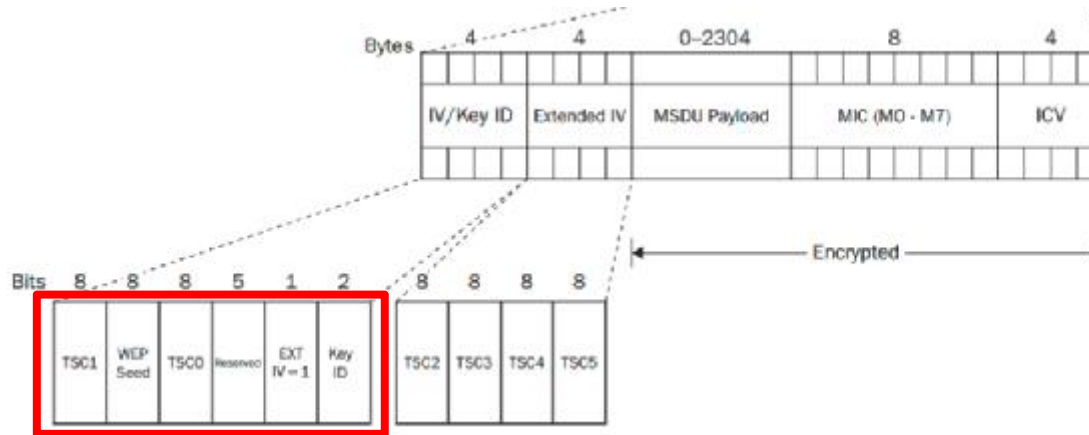
重新封装的Frame Body



五部分构成:

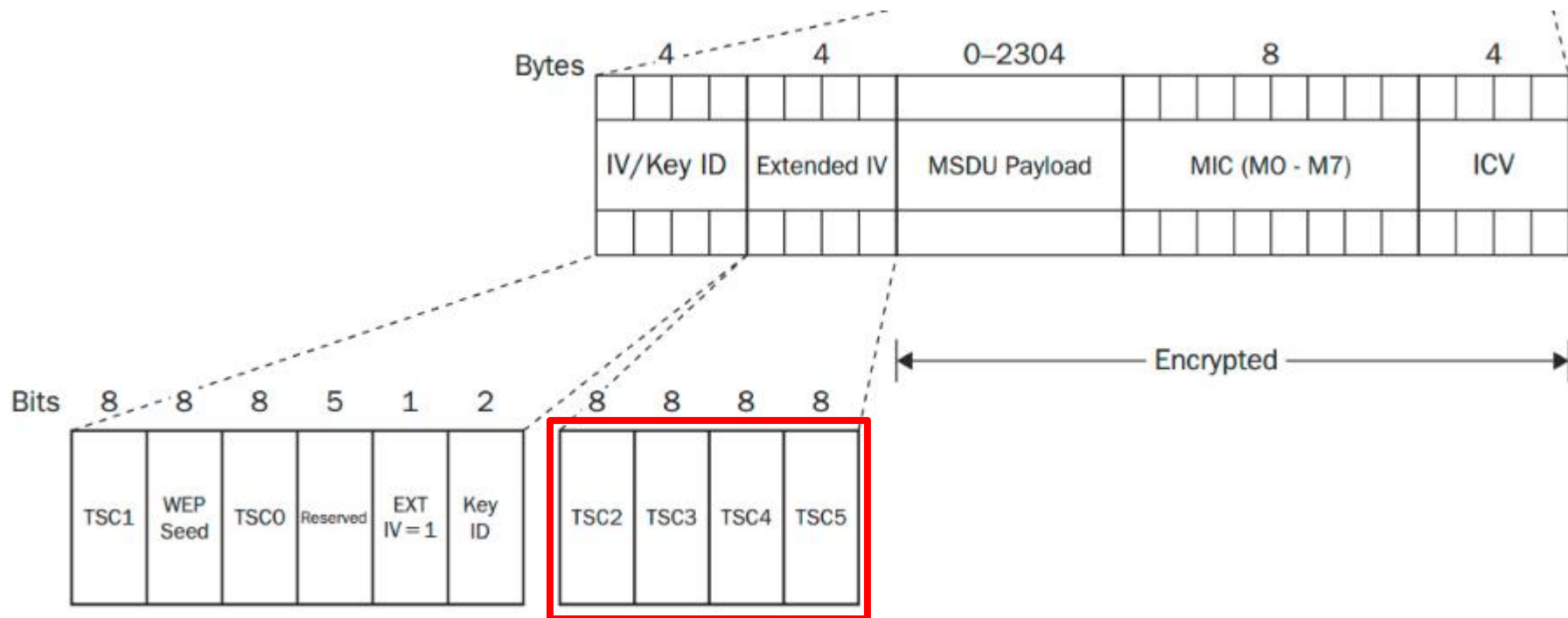
- ◆ IV/Key ID
- ◆ Extended IV
- ◆ MSDU payload
- ◆ MIC
- ◆ ICV

Frame Body: IV/Key ID



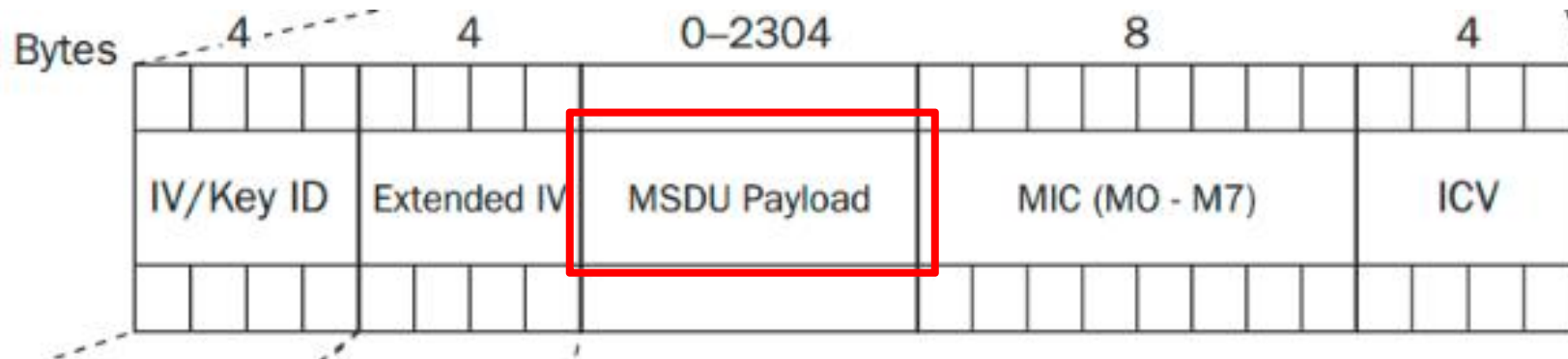
1、IV/Key ID: 它的长度4个字节。WEP Seed是 $(TSC1 | 0x20) \& 0x7f$ 的结果。EXT IV = 1表示后面有Extended IV字段。WEP不需要，因此设置为0；TKIP需要，设置为1。

Frame Body: Extended IV



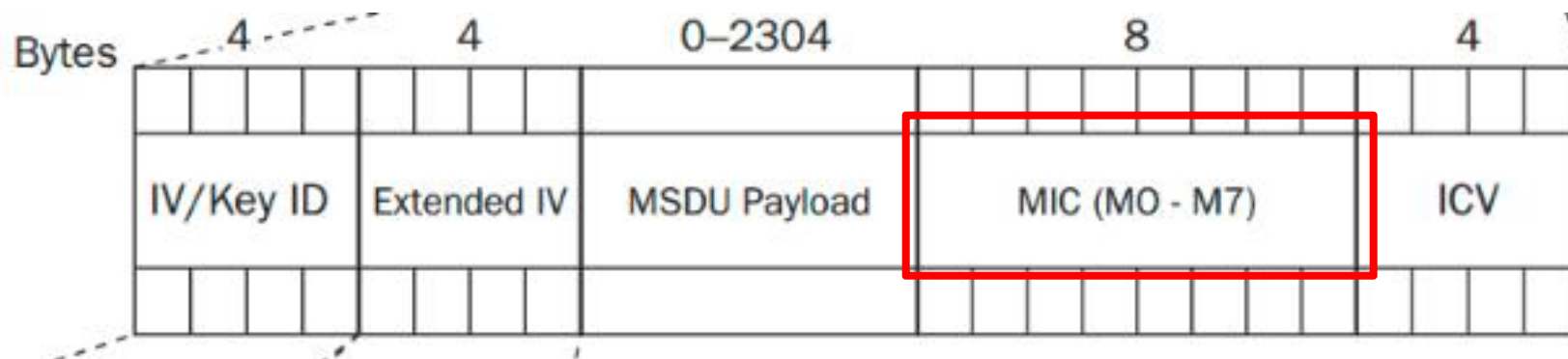
2、Extended IV: 长度4个字节, 是从48-bit TSC导出, 取其中的TSC2到TSC5

Frame Body: MSDU Payload



3、MSDU Payload: 0~2304字节

Frame Body: MIC



4、MIC: MSDU Payload后面是MIC, 8个字节, 当它追加到MSDU后面时, 就成了MSDU的一部分

TKIP: 解密过程

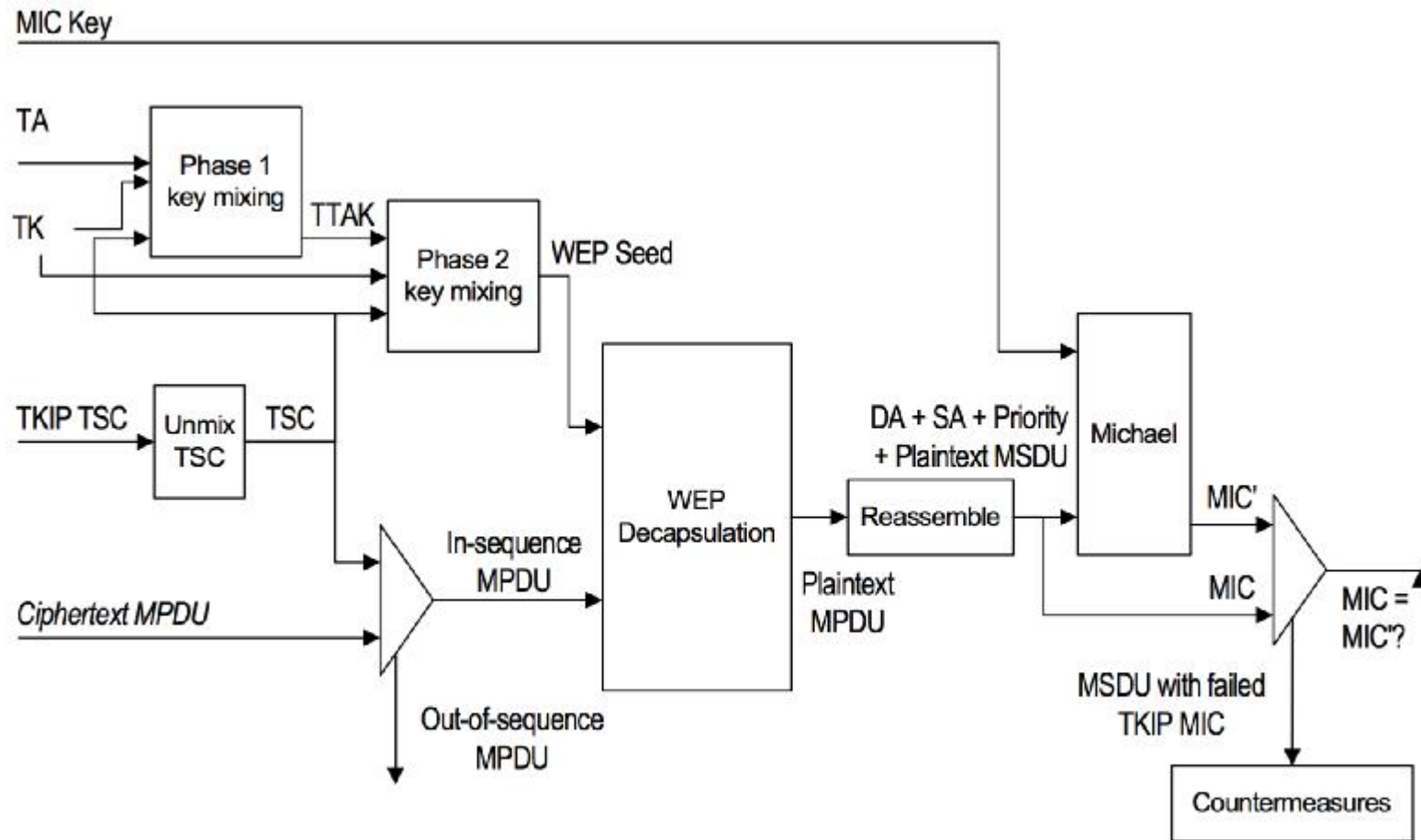


Figure 11-6—TKIP decapsulation block diagram

WPA2

(Wi-Fi Protected Access II)

CCMP

CCM

RFC3610

CCM: Counter with CBC-MAC

CCM是一种通用的**认证加密**分组密码模式，仅定义为使用128比特的分组长度，比如采用AES。但是，根据CCM的设计原理，CCM也可以用于其他分组长度。

对于通用的CCM模式来说，需要两个参数选择。

第一个选择是M，指认证字段的长度，有效值包括4、6、8、10、12、14和16个octets。CCMP选择M为8。

第二个选择是L，指length字段的长度，L的有效值介于2到8个octets（L=1保留用）。CCMP选择L为2。

CCM: 输入

- (1) 适合分组密码的加密密钥K
- (2) 长度为 $15-L$ 的Nonce N。在任何加密密钥K的使用期限内, Nonce不能重复使用。
- (3) 消息m, 包含 $l(m)$ 个octets, $0 \leq l(m) < 2^{(8L)}$ 。长度限制确保了 $l(m)$ 能被编码到L个octets的字段中。
- (4) 附加认证数据a, 由 $l(a)$ 个octets组成, $0 \leq l(a) < 2^{64}$ 。附加认证数据会被认证但是不加密, 附加数据也不包括在这种模式的输出。可以用于认证包头的明文字段, 对理解消息有影响的上下文信息。

CCM: 输入总结

Name	Description	Size
K	Block cipher key/分组加密密钥	取决于分组密码
N	Nonce	15-L octets
m	要认证加密的消息	l(m) octets
a	附加认证数据	l(a) octets

CCM: authentication

首先需要生成以下分组模式:

B_0 || 附加认证数据分组 || 明文数据分组

然后应用CBC-MAC处理上述分组序列:

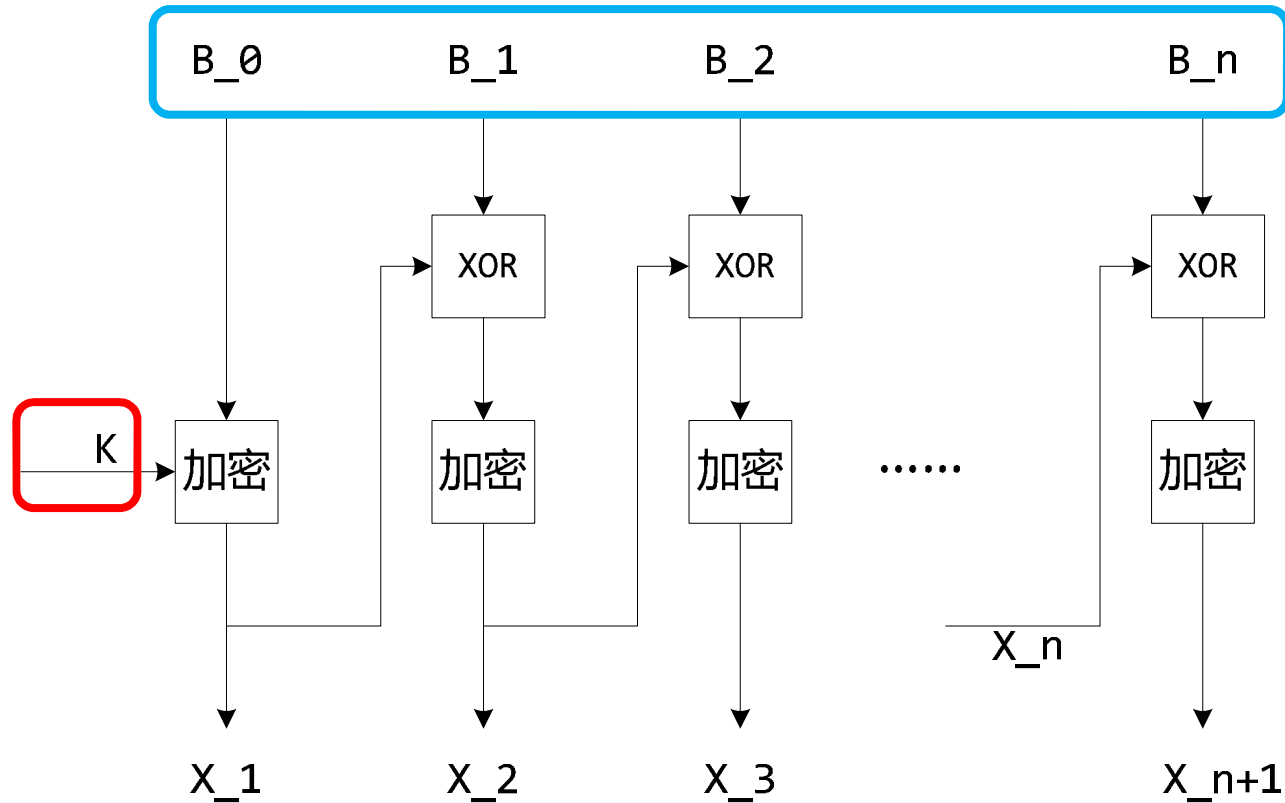
$X_1 := E(K, B_0)$

$X_{i+1} := E(K, X_i \text{ XOR } B_i)$ for $i=1, \dots, n$

$T := \text{first-M-bytes}(X_{n+1})$

T即为MAC值

CCM: authentication

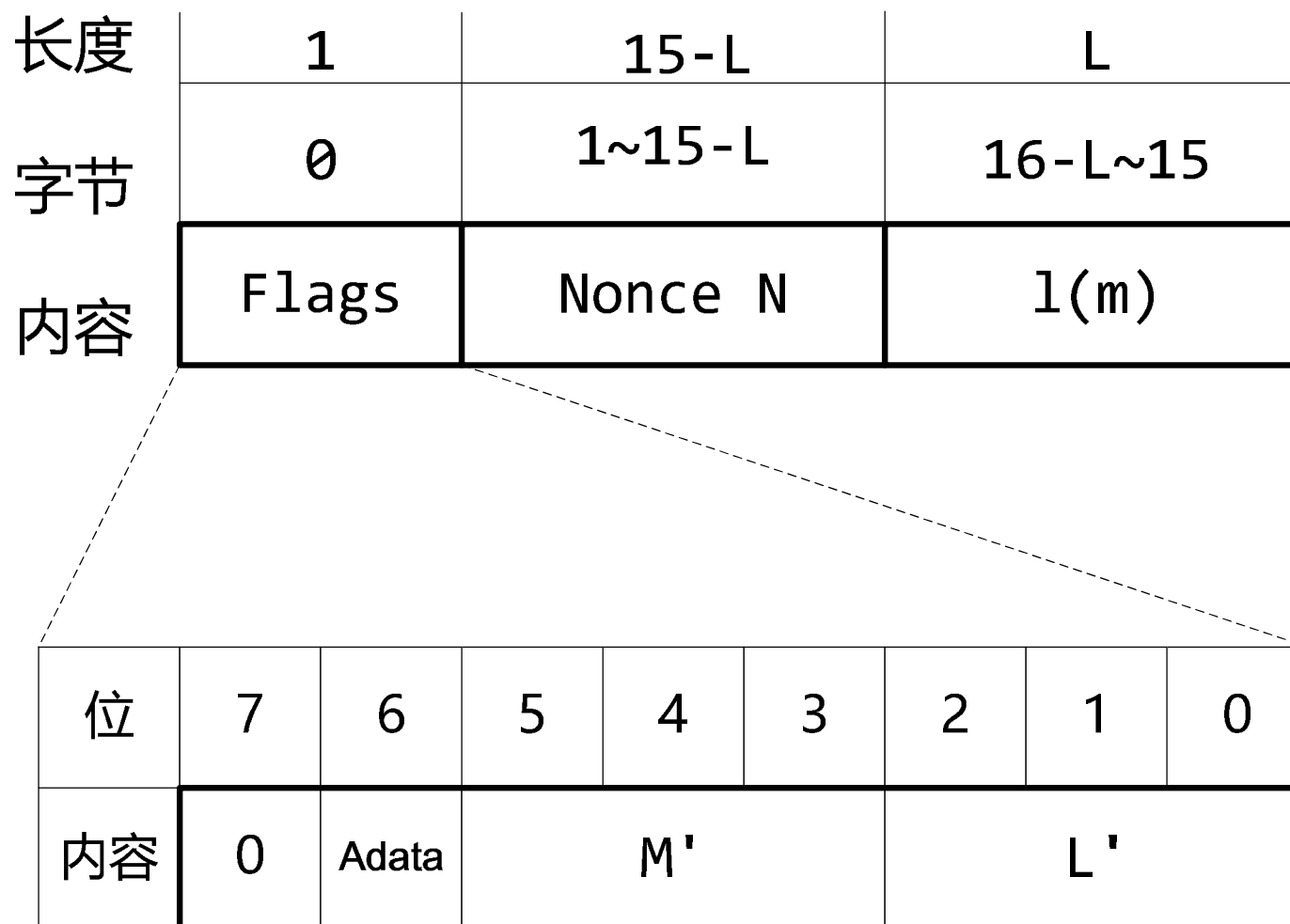


算法的输入:

分组: $B_0, B_1, B_2, \dots, B_n$

密钥: k

CCM authentication: B_0



CCM authentication: B_0 flags

位	7	6	5	4	3	2	1	0
内容	0	Adata	M'			L'		

第7比特保留为将来扩展用，设置为0。第6比特为Adata比特，如果 $1(a) = 0$ ，则Adata为0，表示没有附加认证数据；反之，如果 $1(a) > 0$ ，则Adata为1，表示有附加认证数据。第5到3比特为M'，设置为 $(M-2)/2$ 。第2到0比特为L'，设置为 $L-1$ 。

CCM authentication: 附加认证数据

如果Adata为1，则表示有附加认证数据，其构成为：

$1(a) || a$

也就是 $1(a)$ 与 a 的拼接，然后对拼接的结果按一个分组16字节进行分割，必要时需对最后一个分组添加0x00以补齐16字节。这些分组分别为 B_1, B_2, \dots ，也就是 B_0 后面的分组。注意，这里的 $1(a)$ 部分是对 $1(a)$ （也就是 a 的长度）本身的编码。

CCM authentication: 分组构造

在附加认证数据分组后是消息明文分组。对 m 按一个分组16字节进行划分，必要时添加0x00补齐，从而构成消息明文分组。如果 m 是一个空串，则无需此步。

最后，得到 B_0, B_1, \dots, B_n 分组序列，然后应用CBC-MAC处理上述分组序列

CCM encryption: 密钥流生成

加密过程是采用CTR模式，首先定义密钥流分组为：

$S_i := E(K, A_i)$ ，其中 i 为 $0, 1, 2, \dots$

A_i 的构成

长度	1	15-L	L
字节	0	1...15-L	16...L~15
内容	Flags	Nonce N	Counter i

位	7	6	5	4	3	2	1	0
内容	0	0	0			L'		

CCM encryption: A_i flags构成

长度	1	15-L	L
字节	0	1...15-L	16...L~15
内容	Flags	Nonce N	Counter i

位	7	6	5	4	3	2	1	0
内容	0	0	0			L'		

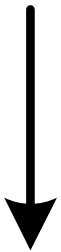
第7和6比特预留为将来扩展用，必须设置为0。第5到3比特全部设置为0，这样就确保了所有的A分组都与B₀（其中M'不为0）不一样。L'与B₀一致。

CCM encryption

密钥: $S_1 \mid S_2 \mid S_3 \mid \cdots \mid S_n$

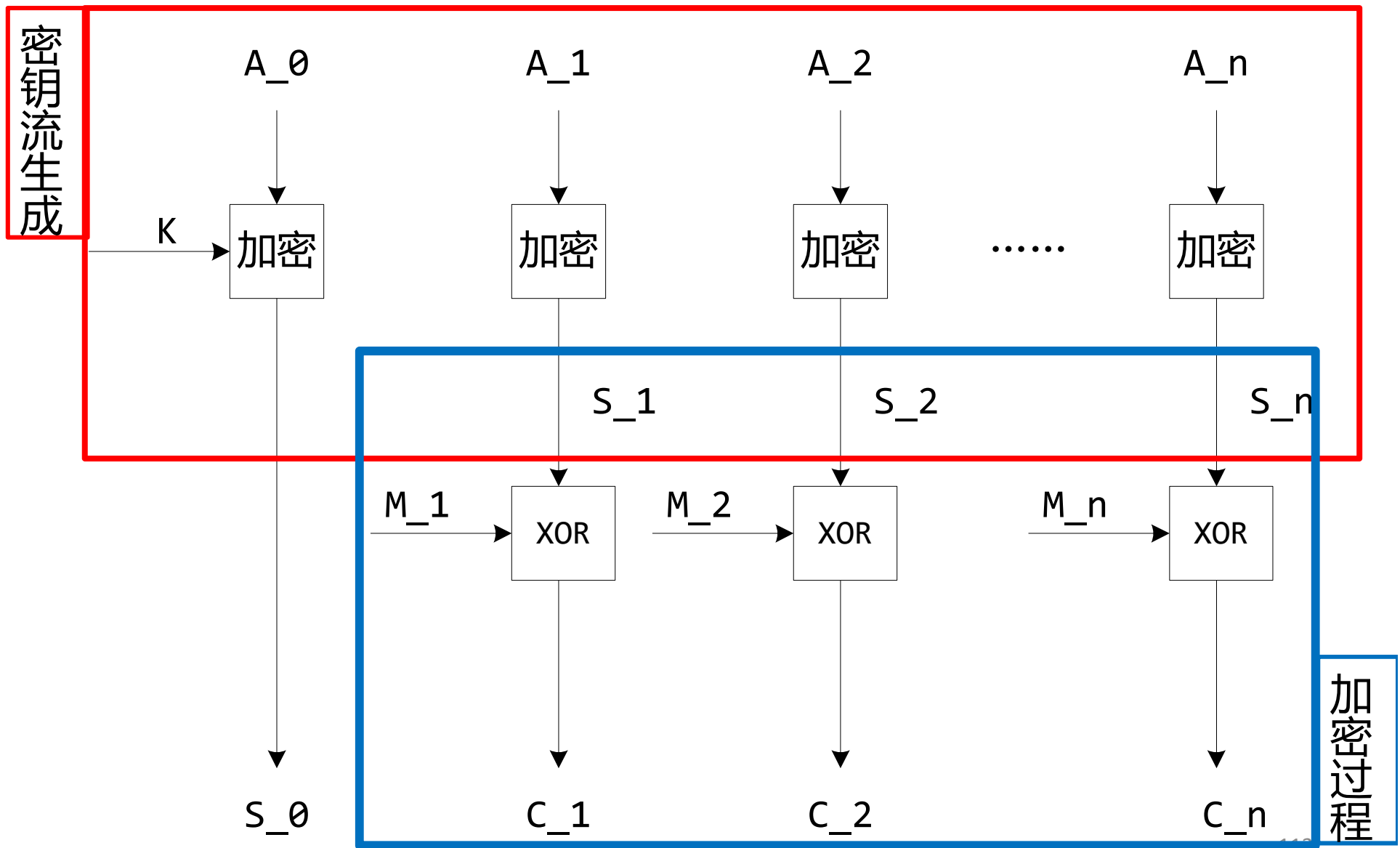
XOR

明文: $M_1 \mid M_2 \mid M_3 \mid \cdots \mid M_n$



密文: $C_1 \mid C_2 \mid C_3 \mid \cdots \mid C_n$

CCM encryption



CCM 计算认证值

注意： s_0 没有用于加密消息， s_0 用于计算认证值。

计算认证值 U ：

$$U := T \text{ XOR first-M-bytes}(s_0)$$

即对 T 用 s_0 的前 M 个字节进行 XOR 运算。

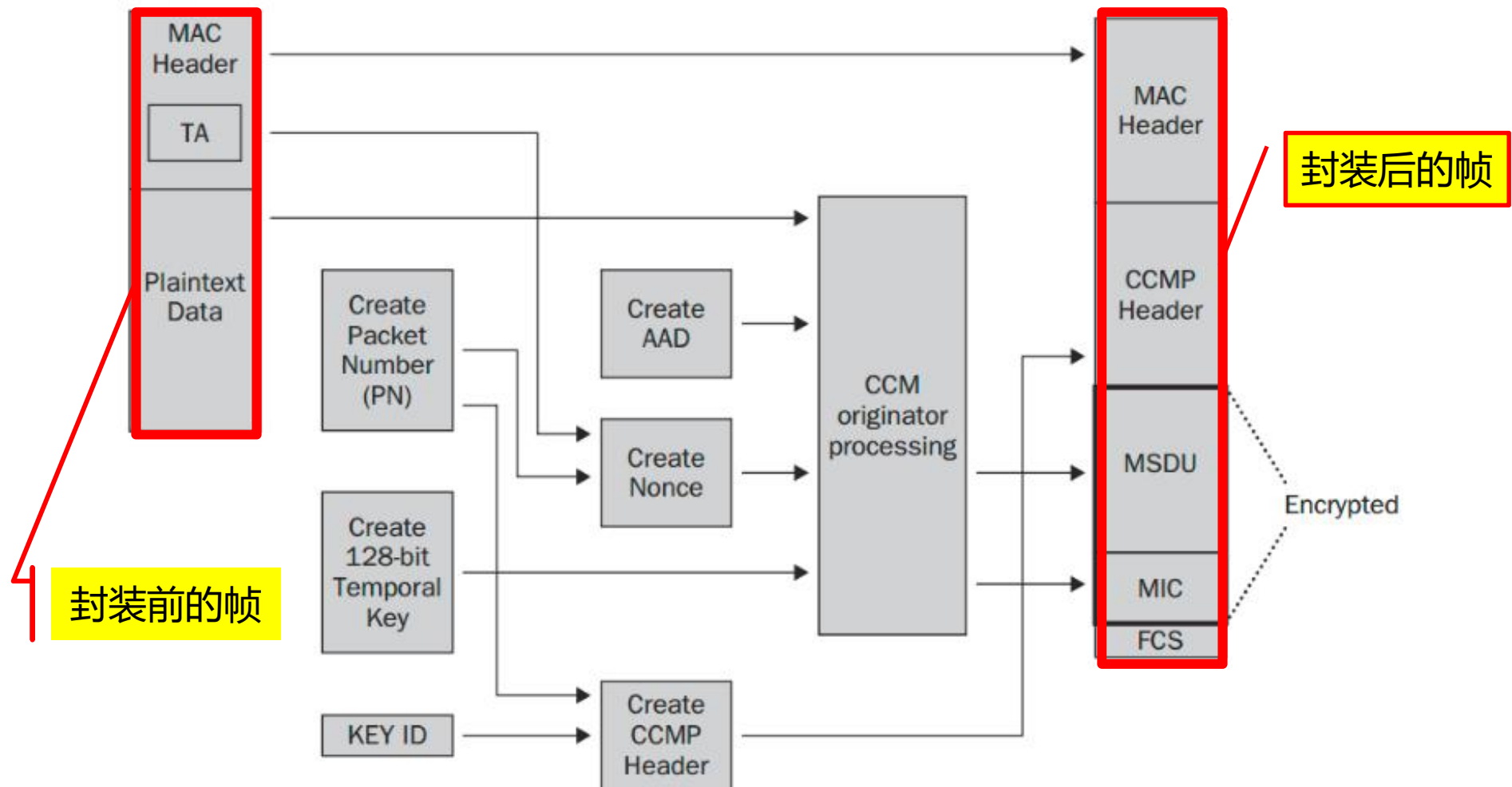
T 是 CBC-MAC 的输出。

最后的输出 c ：加密消息和紧随其后的加密的认证值 U

CCMP

- 基于CCM
- AES, 128bit
- Counter mode with CBC-MAC
 - ◆ Authentication-then-Encryption
 - ◆ CBC-MAC
 - ◆ CTR-AES

CCMP处理过程



体现为如何重新封装帧，以获得安全保护！

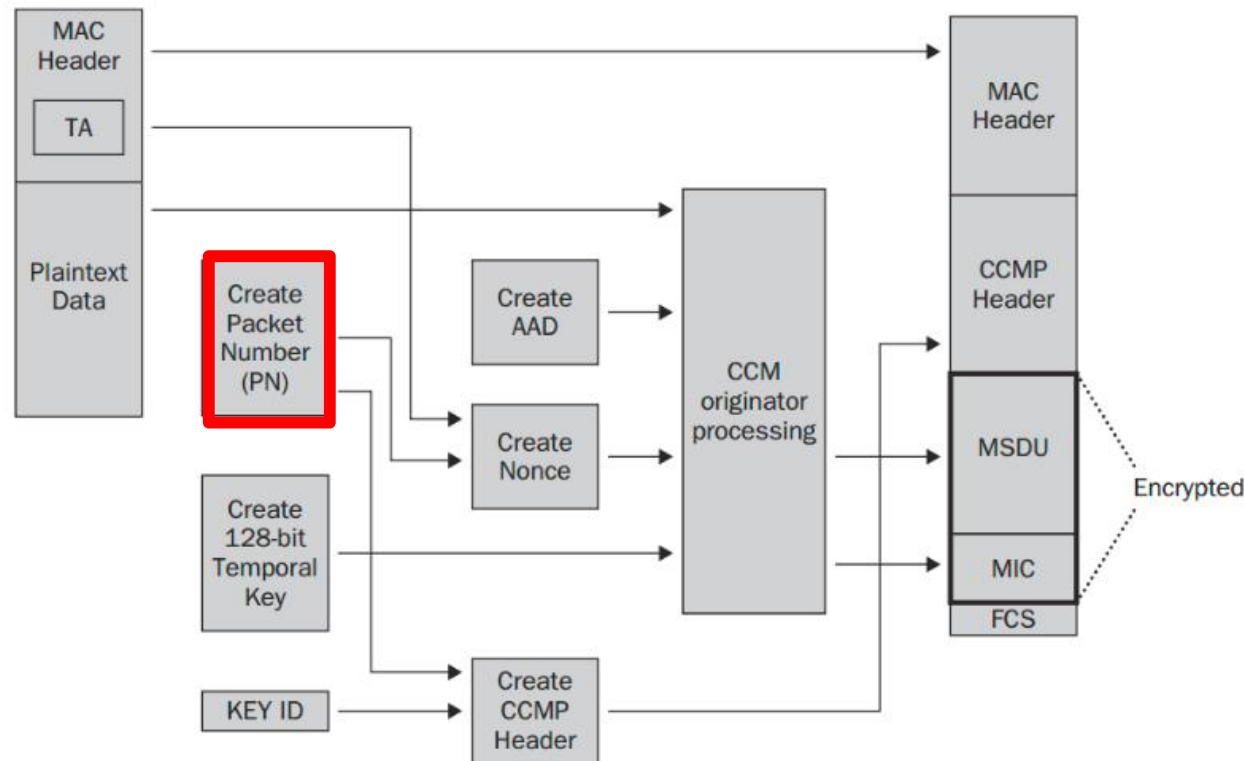
CCMP的输入 (1)

- MAC header: 802.11 MAC 头部
- plaintext Data(MSDU): 需要发送的payload
- PN(packet number): 长度48bit, 与TKIP中的TSC (TKIP Sequence Counter) 相似, 是每个帧的标识, 它会随着帧的发送过程不断递增, 用于抗重放攻击。
- TK(Temporal Key): 和TKIP加密一样, CCMP也有一个128bit的TK。 (在后面密钥管理部分详述)

CCMP的输入 (2)

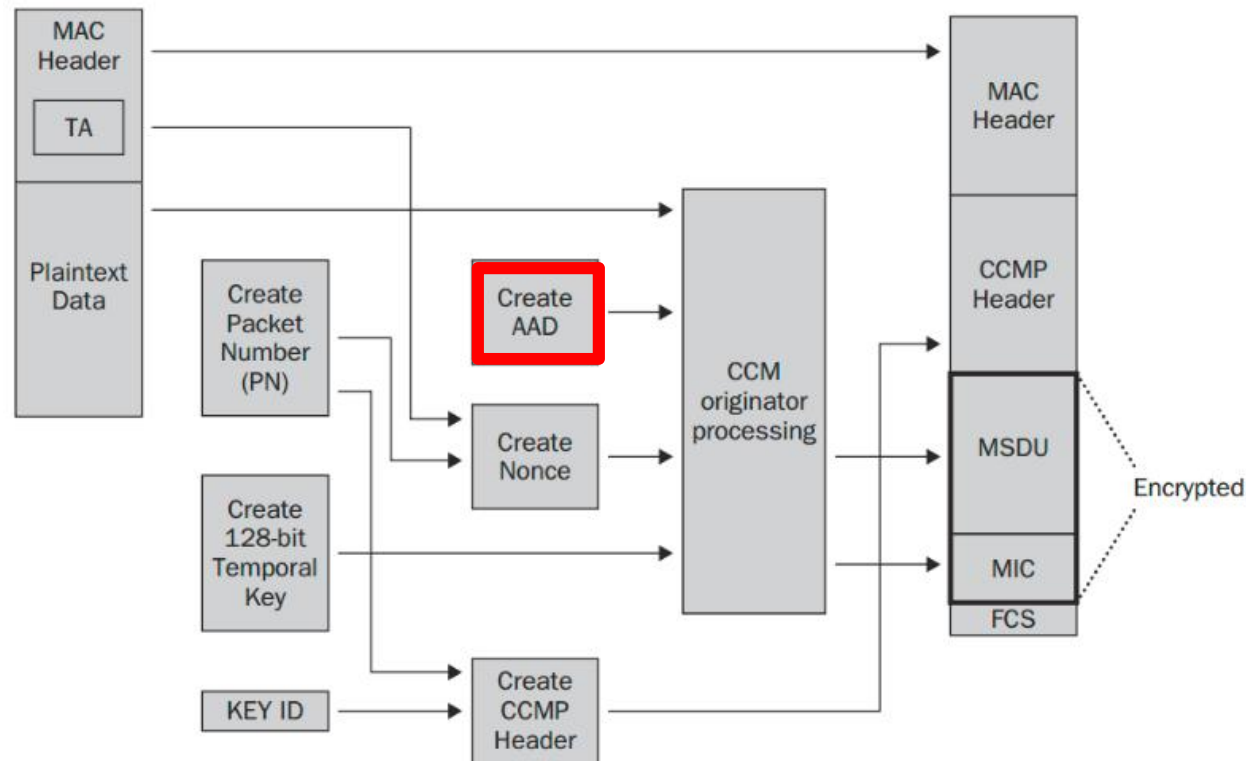
- Key ID: 和TKIP中的一样，用于指定加密用的key，这个ID是index的缩写。
- Nonce：是一个随机数，长104bit，是由PN (packet number, 48bit), Qos中的优先级字段 (8bit) 和 TA(Transmitter Address, 48bit)这三个字段组合来。
- AAD (Additional Authentication Data)：由MPUD的头部构建而来，用于确保MAC头部的数据完整性，接收端会使用这个字段来校验MAC头部。

CCMP: packet number



1、需要发送一个新的MPDU时，会重新创建一个48bit的PN；如果是重传的MPDU，则使用原来发送MPDU的PN。

CCMP: 构建AAD



2、使用 MPDU 的头部构建 AAD (Additional Authentication Data)

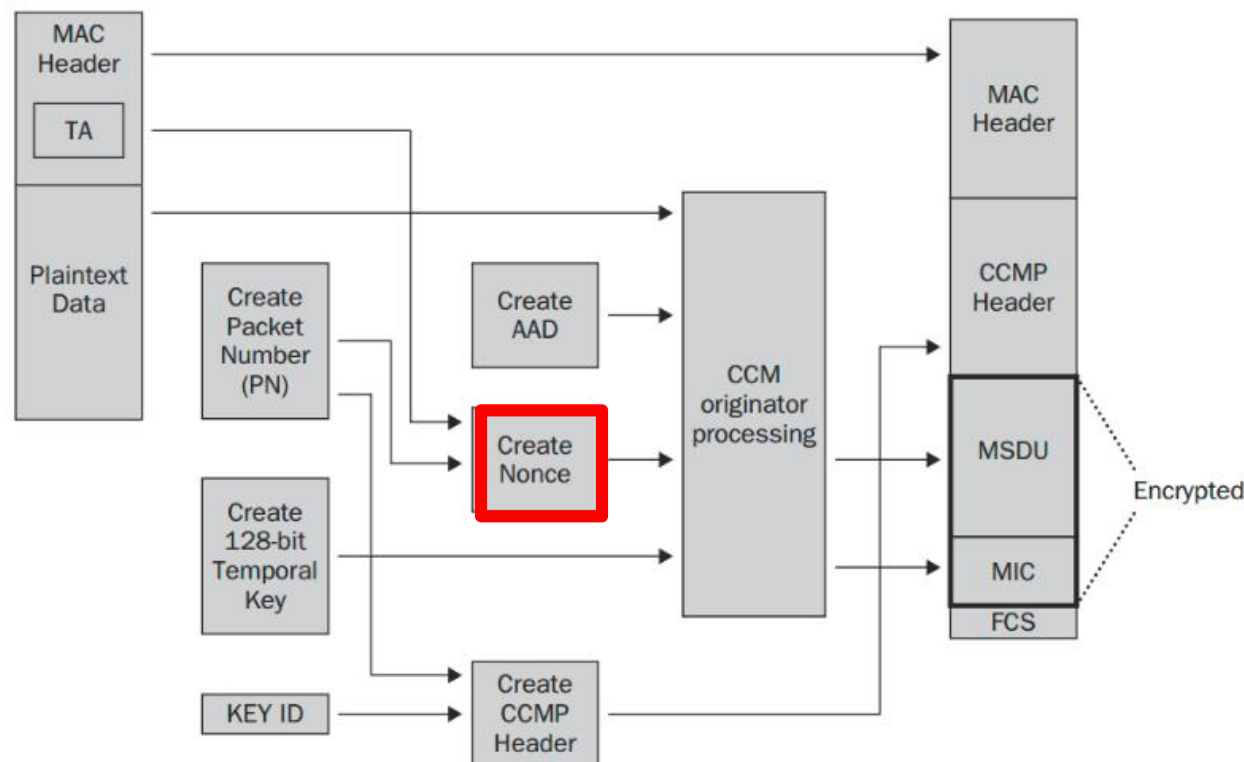
CCMP: AAD构建

	FC	A1	A2	A3	SC	A4	QC
Octets:	2	6	6	6	2	6	2

AAD由MAC Header的上述字段构成，其中部分字段的
部分比特可能设置为0。根据帧的类型不同，A4和QC
字段可能没有，比如管理帧是没有QC字段的。

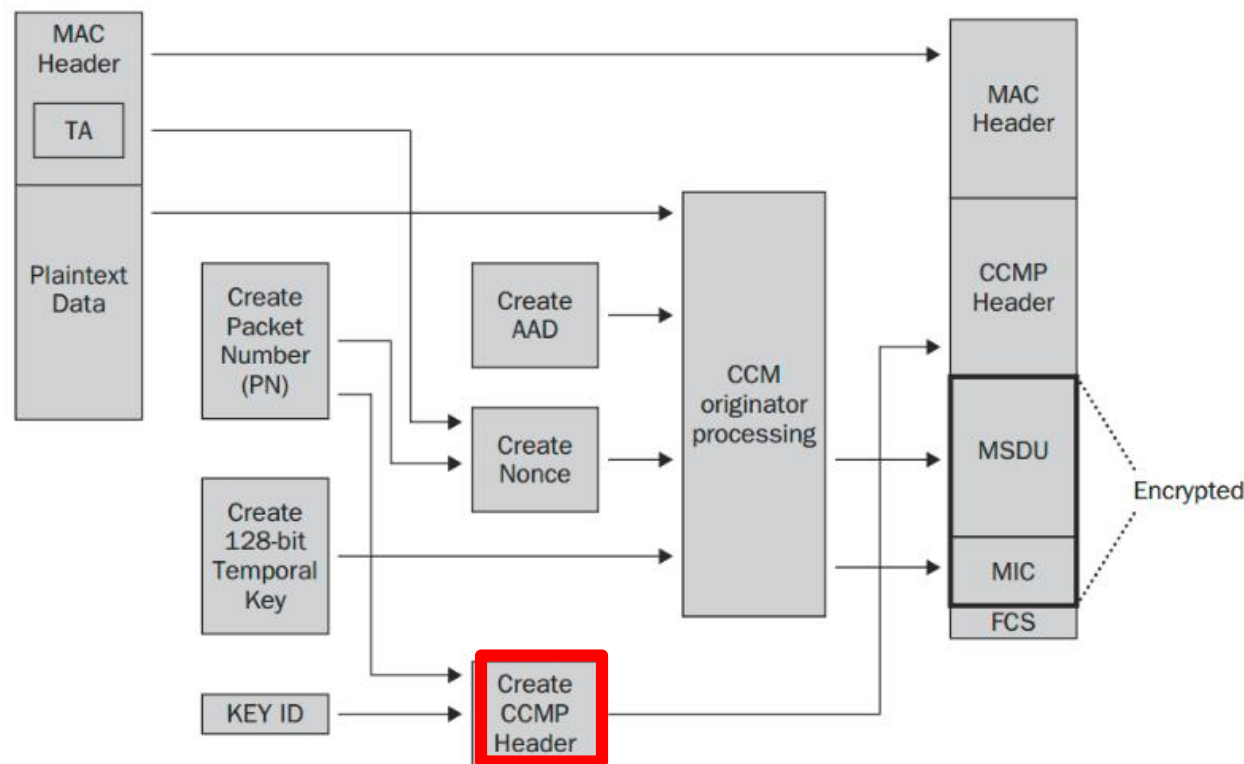
因为AAD作为计算MIC值的输入，因此确保了MAC
Header部分字段的完整性。

CCMP: 构建Nonce



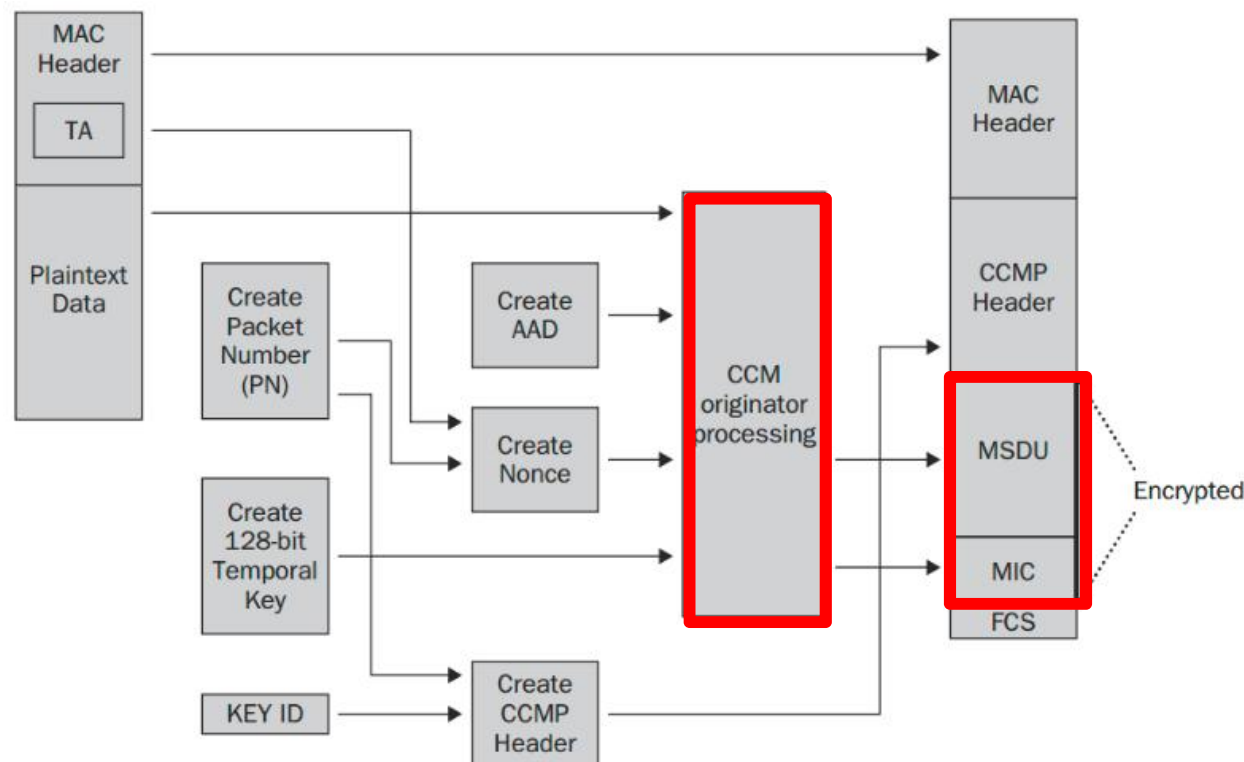
3、由PN(packet number, 48bit), Qos中的优先级字段 (8bit) 和TA(transmitter address , 48bit, Address 2)这三个字段组合生成一个Nonce。

CCMP: 构建CCMP头部



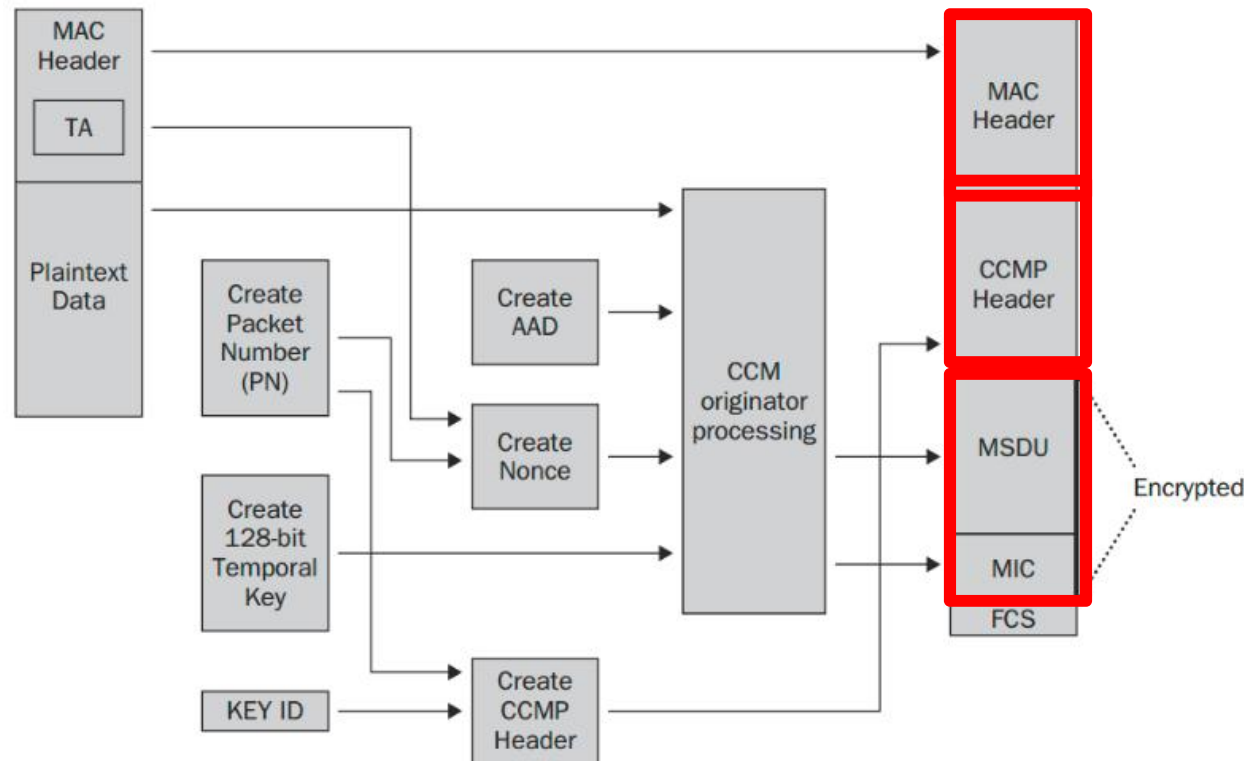
4、构建8-octet CCMP 头部，这个头部由Key ID和PN构成，PN又被分成6个字段。

CCMP: CCM处理



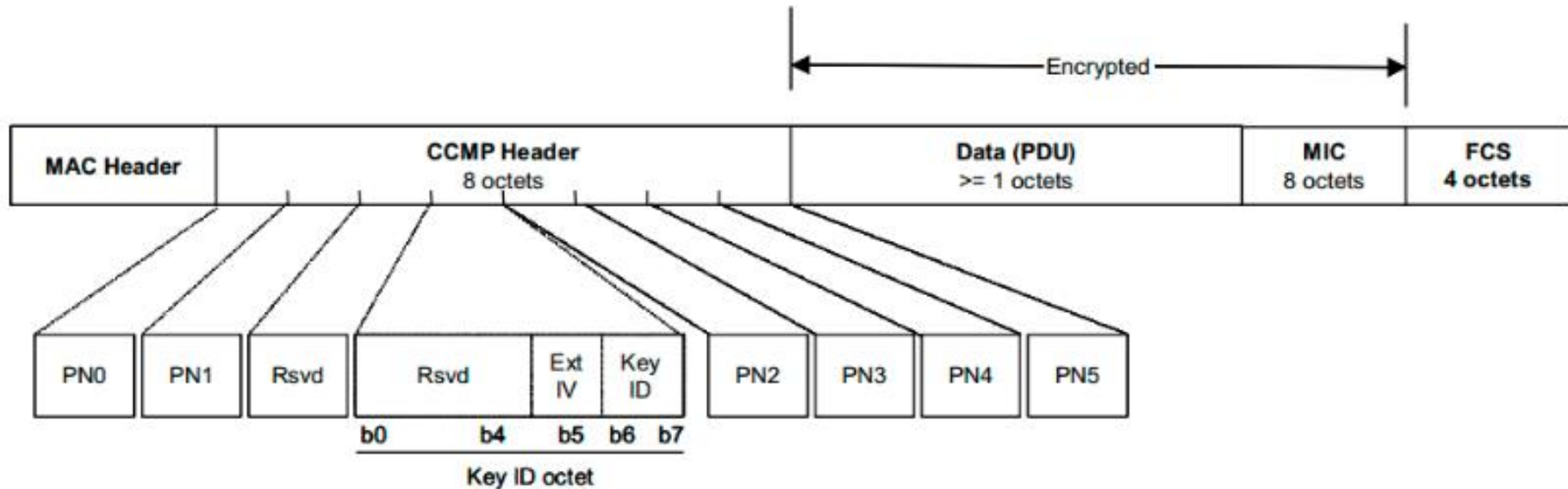
5、使用Temporal Key, AAD, Nonce, 和MPDU data 作为CCM算法输入, 生成8个字节的MIC和加密的MSDU。

CCMP: 封装安全处理过后的帧



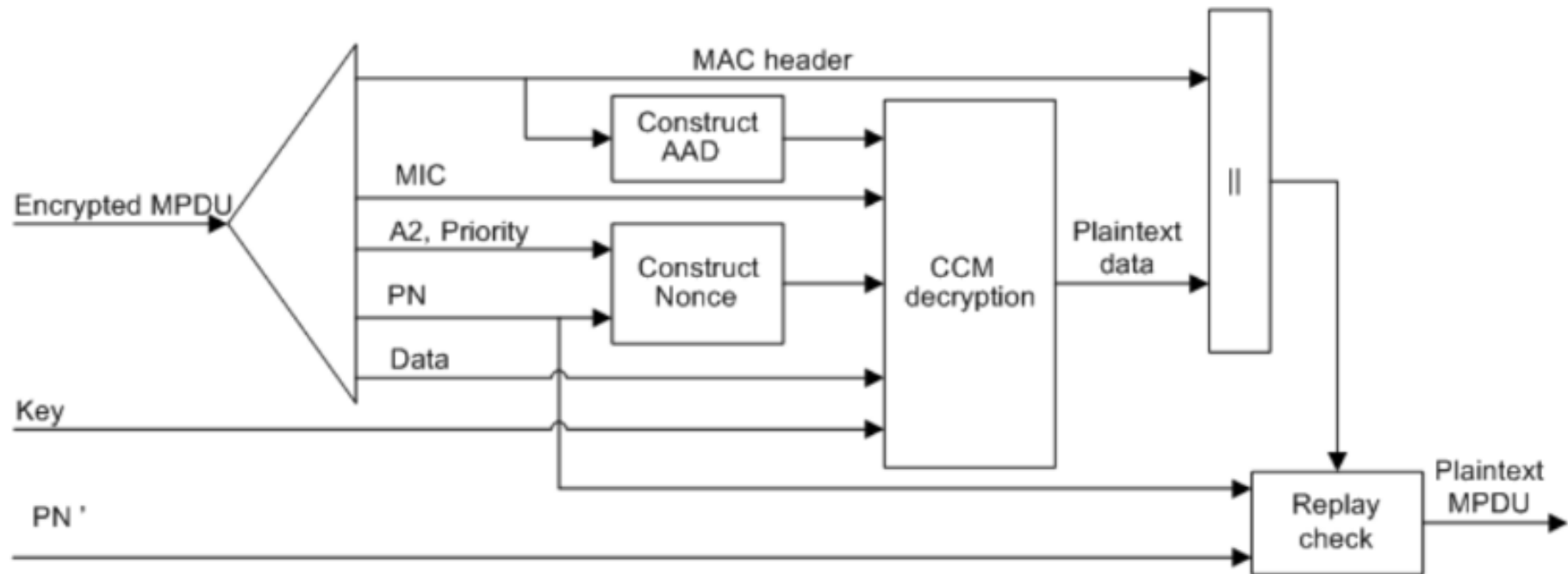
6、在MAC Header后面追加CCMP Header，然后是加密的MSDU和MIC，接下来的是FCS，构成安全处理过后的帧。

CCMP MPDU



- ◆MAC Header (32字节) 没有任何变化,
- ◆帧体由CCMP Header || 加密的 (MSDU Payload || MIC) 构成
- ◆CCMP Header由Key ID和PN构成(PN被分为6个字段, 分别放置)

CCMP解密过程



CCMP总结

- ▣ CCMP针对MPDU进行安全处理，对MSDU提供保密性，同时对MSDU和MAC Header的部分字段做完整性保护。

RSNA 密钥管理

RSNA

- Robust Security Network Association

 - ◆ 强健安全网络关联

 - ◆ The Wi-Fi Alliance refers to their approved, interoperable implementation of the full 802.11i as WPA2, also called RSN (Robust Security Network)

- WEP: Pre-RSNA

- WPA/WPA2: RSNA

密钥管理

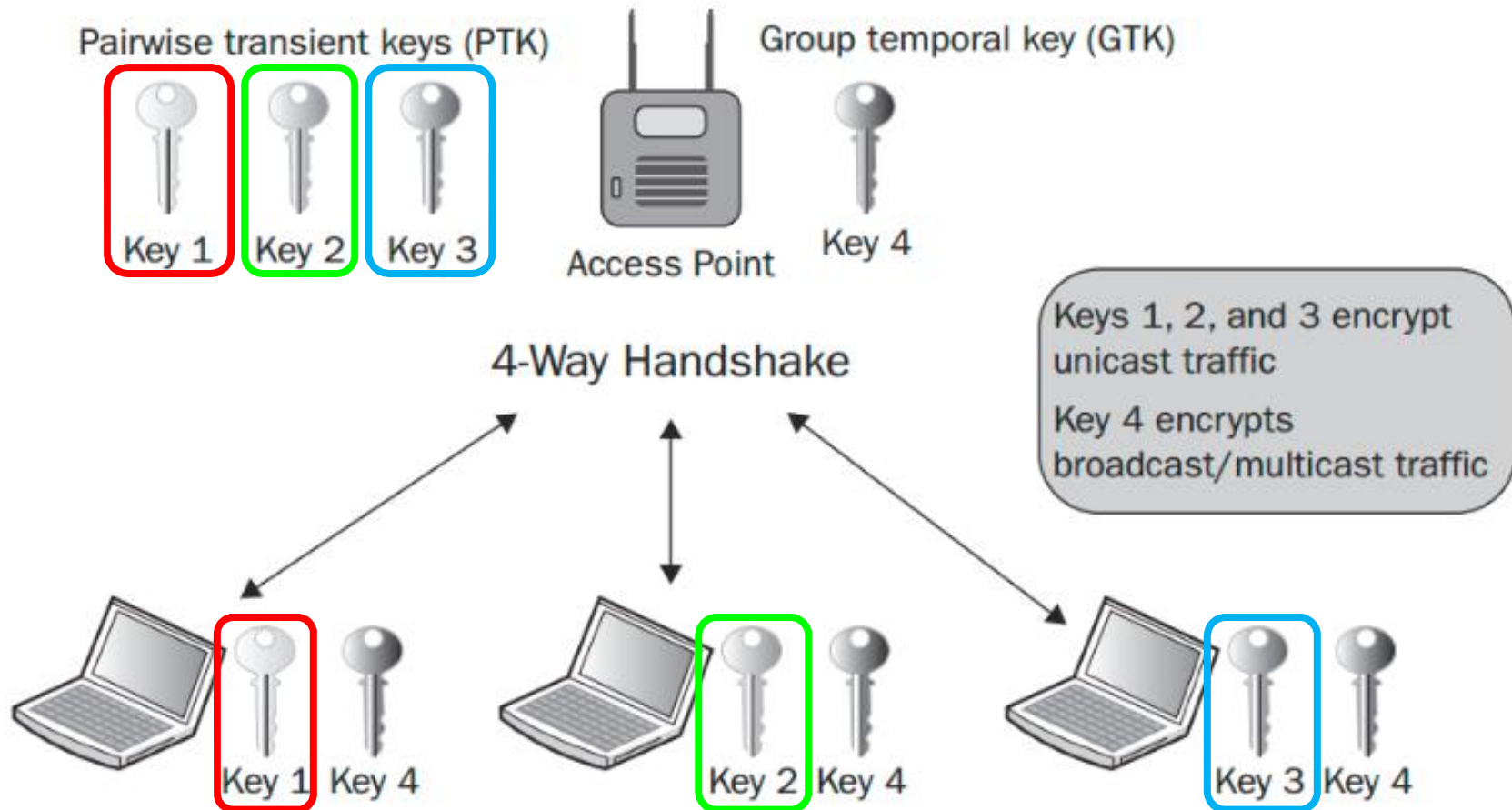
□ WEP

- ◆所有STA使用相同的WEP Key进行数据加密
- ◆安全性较差

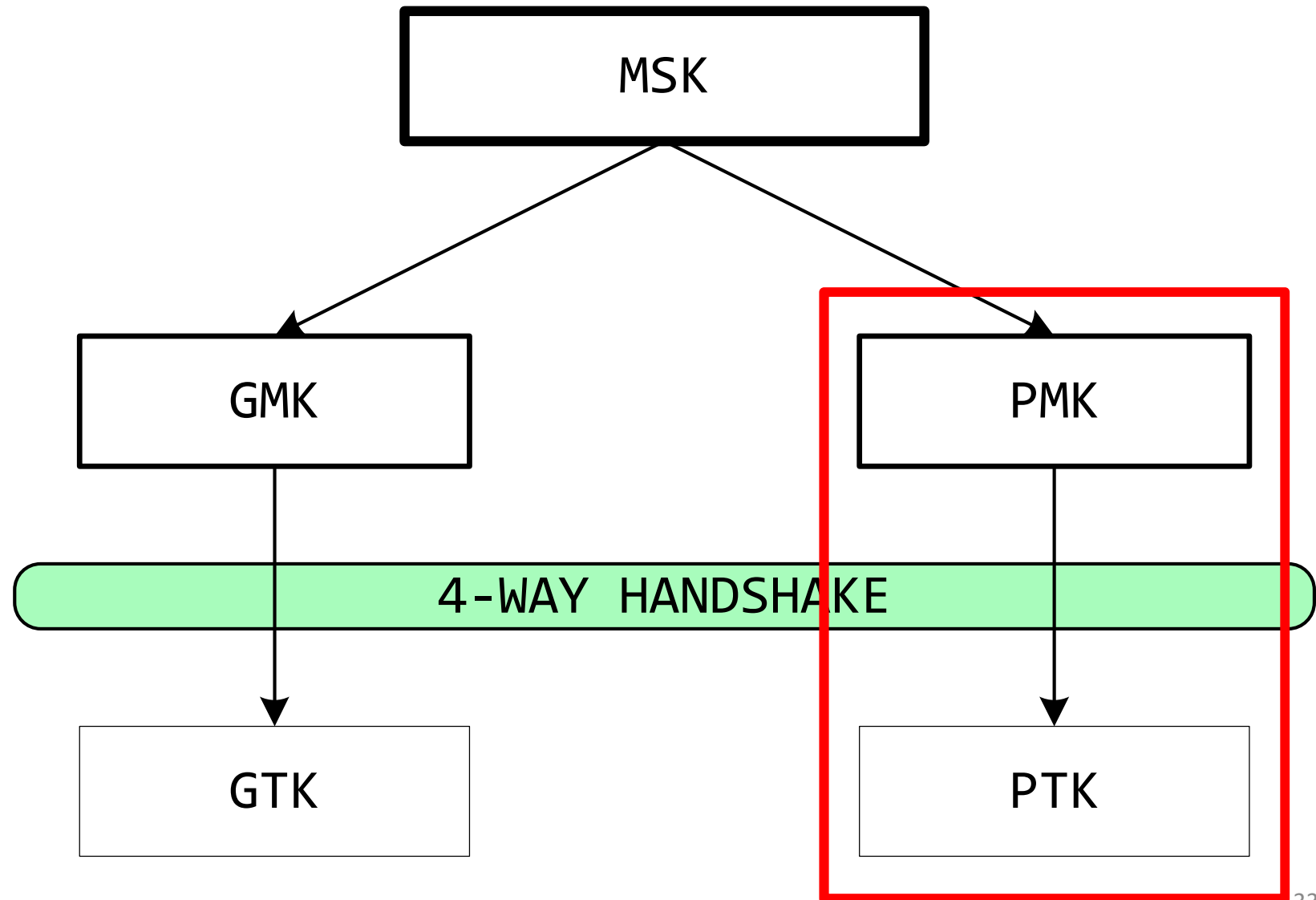
□ RSNA

- ◆关联后，不同STA和AP之间使用不同Key进行数据加密，即：Pairwise Key

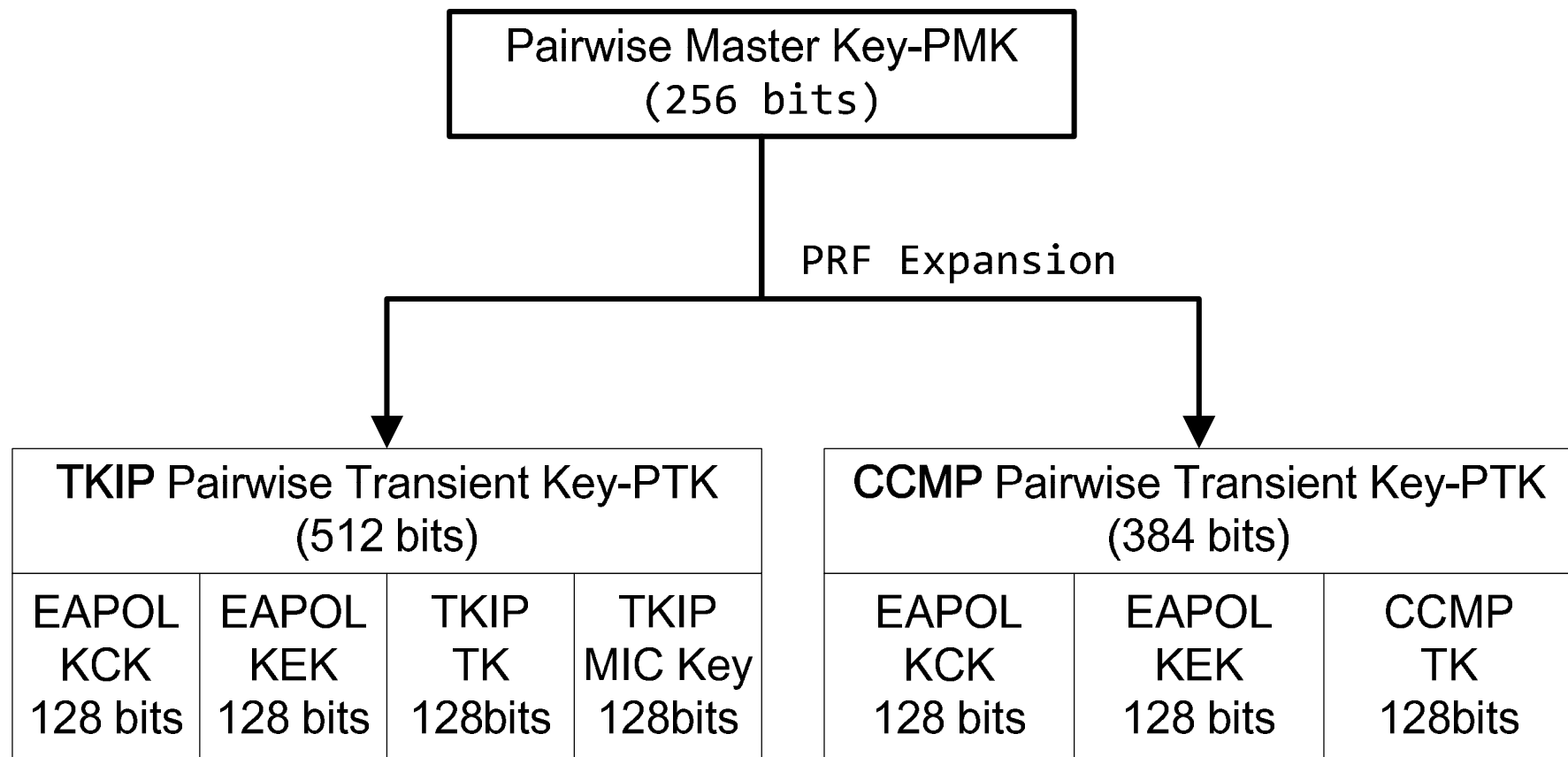
RSNA Pairwise Key



RSNA密钥层次



密钥导出



密钥用途

- KCK: 用于计算WPA EAPOL-Key的MIC
- KEK: AP用KEK加密发送给STA的附加数据, 在Key Data字段, 比如GTK信息 (EAPOL-key)
- TK: 用于加/解密单播数据帧
- Tx: 用于计算AP发送的单播数据帧的MIC
 - ◆ Michael MIC Key
- Rx: 用于计算STA发送的单播数据帧的MIC
 - ◆ Michael MIC key

用于TKIP

RSNA过程

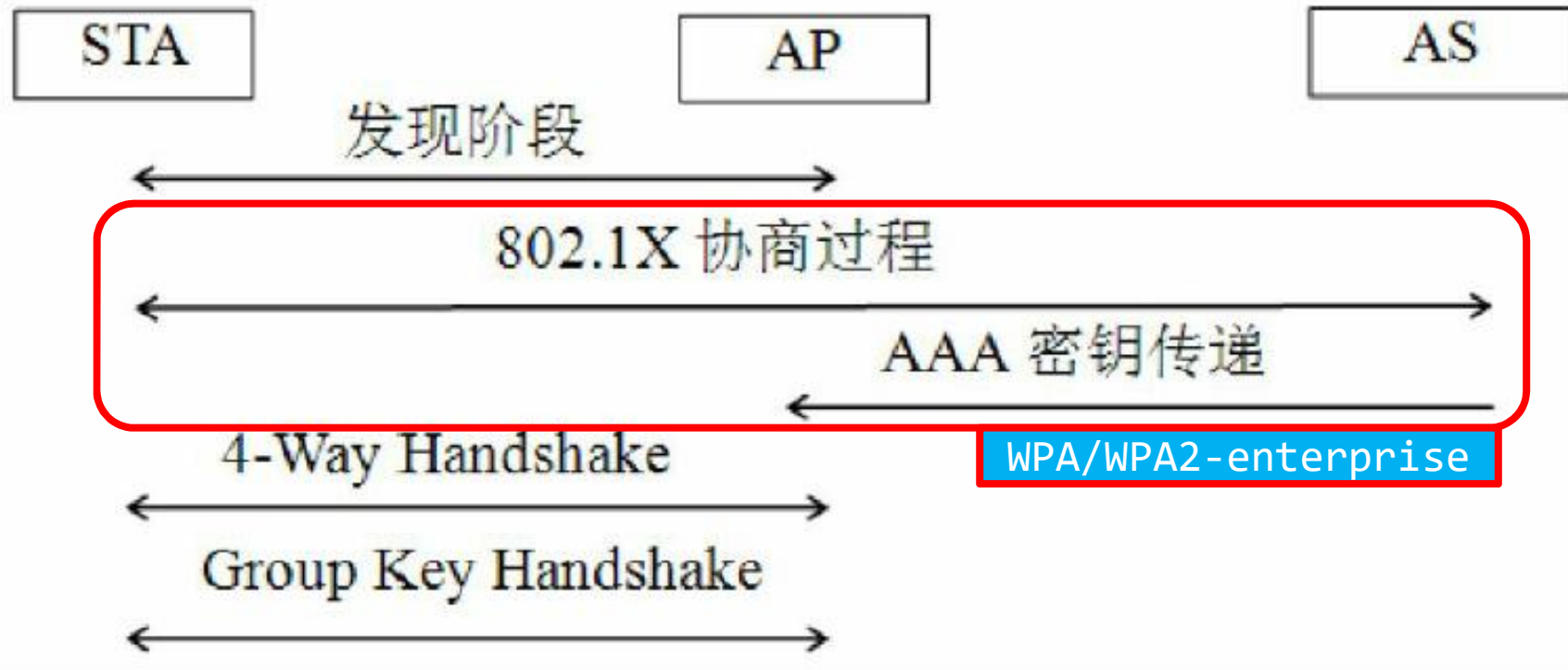


图3-45 RSNA过程

参考：《深入理解Android：WiFi模块 NFC和GPS卷》

PMK导出

□ WPA/WPA2-Enterprise

- ◆ $PMK = L(MSK, 0, 256)$
- ◆ MSK的通过EAPOL交互过程导出的AAA Key

□ WPA/WPA2-PSK

- ◆ PMK 就是PSK
- ◆ 通过Password和SSID导出

WPA/WPA2-PSK: PMK

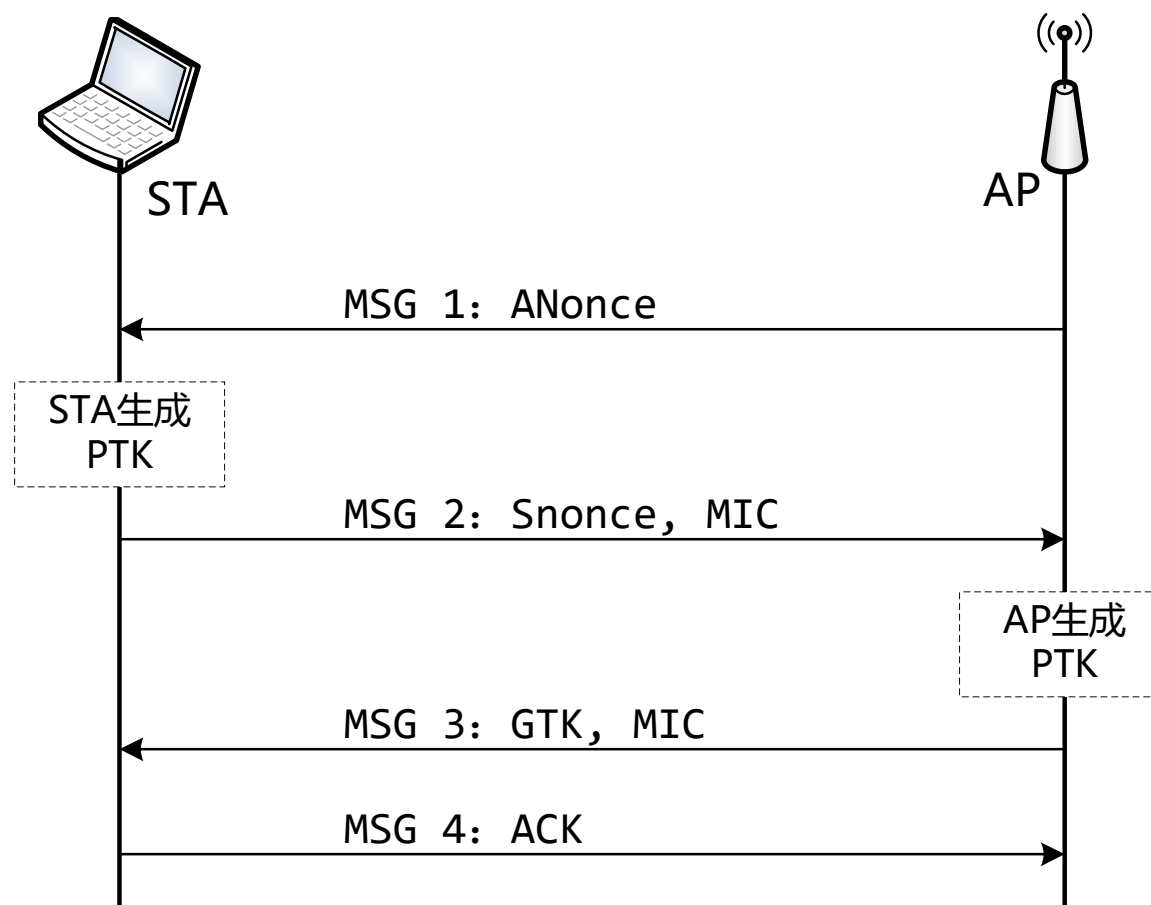
PMK = **PBKDF2**(HMAC-SHA1, PWD, SSID, 4096, 256)

RFC 8018

```
from binascii import b2a_hex
from hashlib import pbkdf2_hmac
pwd = '12345678'
ssid = 'Harkonen'
pmk =
pbkdf2_hmac('sha1',pwd.encode('ascii'),ssi
d.encode('ascii'),4096,32)
pmkStr = b2a_hex(pmk).decode().upper()
print(pmkStr)
```

4-WAY HANDSHAKE

4次握手用于鉴别通信两端的真实性和建立加密密钥



4-WAY HANDSHAKE: MSG 1

MESSAGE 1: AP→STA

```
▶ Frame 2: 131 bytes on wire (1048 bits), 131 bytes captured (1048 bits)
▶ IEEE 802.11 Data, Flags: .....F.
▶ Logical-Link Control
▼ 802.1X Authentication
  Version: 802.1X-2001 (1)
  Type: Key (3)
  Length: 95
  Key Descriptor Type: EAPOL RSN Key (2)
  ▶ Key Information: 0x008a
  Key Length: 16
  Replay Counter: 1
  WPA Key Nonce: 225854b0444de3af06d1492b852984f04cf6274c0e3218b8...
  Key IV: 00000000000000000000000000000000
  WPA Key RSC: 0000000000000000
  WPA Key ID: 0000000000000000
  WPA Key MIC: 00000000000000000000000000000000
  WPA Key Data Length: 0
```

第一条消息后

STA拥有的信息:

PMK

SNonce

ANonce

AA

SPA

$$PTK \leftarrow \text{PRF-X}(\text{PMK}, \text{"Pairwise key expansion"}, \text{Min}(\text{AA}, \text{SPA}) || \text{Max}(\text{AA}, \text{SPA}) || \text{Min}(\text{ANonce}, \text{SNonce}) || \text{Max}(\text{ANonce}, \text{SNonce}))$$

PRF-X是伪随机函数, $X = 256 + \text{TK_bits}$, TK_bits与具体的密码套件有关。

TKIP: TK_bits=256

CCMP: TK_bits=128

PRF() for PTK

```
#Pseudo-random function for generation of
#the pairwise transient key (PTK)
#key:      The PMK
#A:        b'Pairwise key expansion'
#B:        The apMac, cliMac, aNonce, and sNonce concatenated
#          like mac1||mac2||nonce1||nonce2
#          such that mac1 < mac2 and nonce1 < nonce2
#return:    The ptk
def PRF(key, A, B):
    #Number of bytes in the PTK
    nByte = 64
    i = 0
    R = b''
    #Each iteration produces 160-bit value and 512 bits are required
    while(i <= ((nByte * 8 + 159) / 160)):
        hmacsha1 = hmac.new(key, A + chr(0x00).encode() + B +
chr(i).encode(), sha1)
        R = R + hmacsha1.digest()
        i += 1
    return R[0:nByte]
```

Parameters for PTK

```
#Make parameters for the generation of the PTK
#aNonce:      The aNonce from the 4-way handshake
#sNonce:      The sNonce from the 4-way handshake
#apMac:       The MAC address of the access point
#cliMac:      The MAC address of the client
#return:      (A, B) where A and B are parameters
#             for the generation of the PTK
def MakeAB(aNonce, sNonce, apMac, cliMac):
    A = b"Pairwise key expansion"
    B = min(apMac, cliMac) + max(apMac, cliMac) + min(aNonce, sNonce)
    + max(aNonce, sNonce)
    return (A, B)

ptk = PRF(pmk, A, B)
```

4-WAY HANDSHAKE: MESSAGE 2

MESSAGE 2: STA→AP

```
▶ Frame 3: 153 bytes on wire (1224 bits), 153 bytes captured (1224 bits)
▶ IEEE 802.11 Data, Flags: .....T
▶ Logical-Link Control
▼ 802.1X Authentication
```

```
Version: 802.1X-2001 (1)
Type: Key (3)
Length: 117
Key Descriptor Type: EAPOL RSN Key (2)
▶ Key Information: 0x010a
Key Length: 16
Replay Counter: 1
```

```
WPA Key Nonce: 59168bc3a5df18d71efb6423f340088dab9e1ba2bbc58659...
```

SNonce

```
Key IV: 00000000000000000000000000000000
```

```
WPA Key RSC: 0000000000000000
```

```
WPA Key ID: 0000000000000000
```

```
WPA Key MIC: d5355382b8a9b806dcaf99cdaf564eb6
```

MIC

```
WPA Key Data Length: 22
```

```
▶ WPA Key Data: 30140100000fac040100000fac040100000fac020100
```


第二条消息后

AP拥有的信息:

PMK

ANonce

SNonce

AA

SPA

AP采用与STA相同的算法生成PTK:

$$\text{PTK} \leftarrow \text{PRF-X}(\text{PMK}, \text{"Pairwise key expansion"}, \text{Min}(\text{AA}, \text{SPA}) \parallel \text{Max}(\text{AA}, \text{SPA}) \parallel \text{Min}(\text{ANonce}, \text{SNonce}) \parallel \text{Max}(\text{ANonce}, \text{SNonce}))$$

Message Integrity Check (MIC)

MIC: 128-bits。用于验证STA知道PTK，从而知道正确的PMK，也就是验证了STA的合法性（真实性）。

计算MIC:

以802.1x的所有字段作为HMAC函数的输入，计算时MIC字段设置为0。

对WPA，所采用的哈希函数是MD5(128bits)

对WPA2，所采用的哈希函数是SHA1(160bits)

计算MIC

```
#The entire 802.1x frame with the MIC field set to all zeros
data1 =
a2b_hex("0103007502010a00100000000000000000159168bc3a5df18d71efb6423f
340088dab9e1ba2bbbc58659e07b3764b0de857000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
01630140100000fac040100000fac040100000fac020100")
#WPA uses md5 to compute the MIC while WPA2 uses sha1
wpa = false #treat it as WPA2
hmacFunc = md5 if wpa else sha1
#Create the MICs using HMAC-SHA1 of data and return all computed
values
mic1 = hmac.new(ptk[0:16], data1, hmacFunc).digest()
#the result should be mic1 = "d5355382b8a9b806dc9f99cdaf564eb6"
```

4-WAY HANDSHAKE: MESSAGE 3

MESSAGE 3: AP→STA

```
▶ Frame 4: 187 bytes on wire (1496 bits), 187 bytes captured (1496 bits)
▶ IEEE 802.11 Data, Flags: .....F.
▶ Logical-Link Control
▼ 802.1X Authentication
  Version: 802.1X-2001 (1)
  Type: Key (3)
  Length: 151
  Key Descriptor Type: EAPOL RSN Key (2)
▶ Key Information: 0x13ca
  Key Length: 16
  Replay Counter: 2
  WPA Key Nonce: 225854b0444de3af06d1492b852984f04cf6274c0e3218b8...
  Key IV: 192eeef7fd968ec80aee3dfb875e8222
  WPA Key RSC: 3700000000000000
  WPA Key ID: 0000000000000000
  WPA Key MIC: 1e228672d2dee930714f688c5746028d
  WPA Key Data Length: 56
  WPA Key Data: 3ca9185462eca4ab7ff51cd3a3e6179a8391f5ad824c9e09...
```

MESSAGE 3: 关键信息

MIC: 计算方法与第二个握手包相同。用于鉴别AP的真实性（不是流氓AP），一旦STA验证了MIC，则STA能够确保该AP知道正确的PTK，从而知道正确的PMK，而只有正确的AP才能知道PMK，因此验证了AP的真实性。

WPA Key Data: 包含了加密的GTK

4-WAY HANDSHAKE: MESSAGE 4

MESSAGE 4: STA→AP

用于确认STA已经收到正确的keys并且加密通信即将开始。第四个握手包也包含了MIC字段，计算方法同前。

问题讨论1:

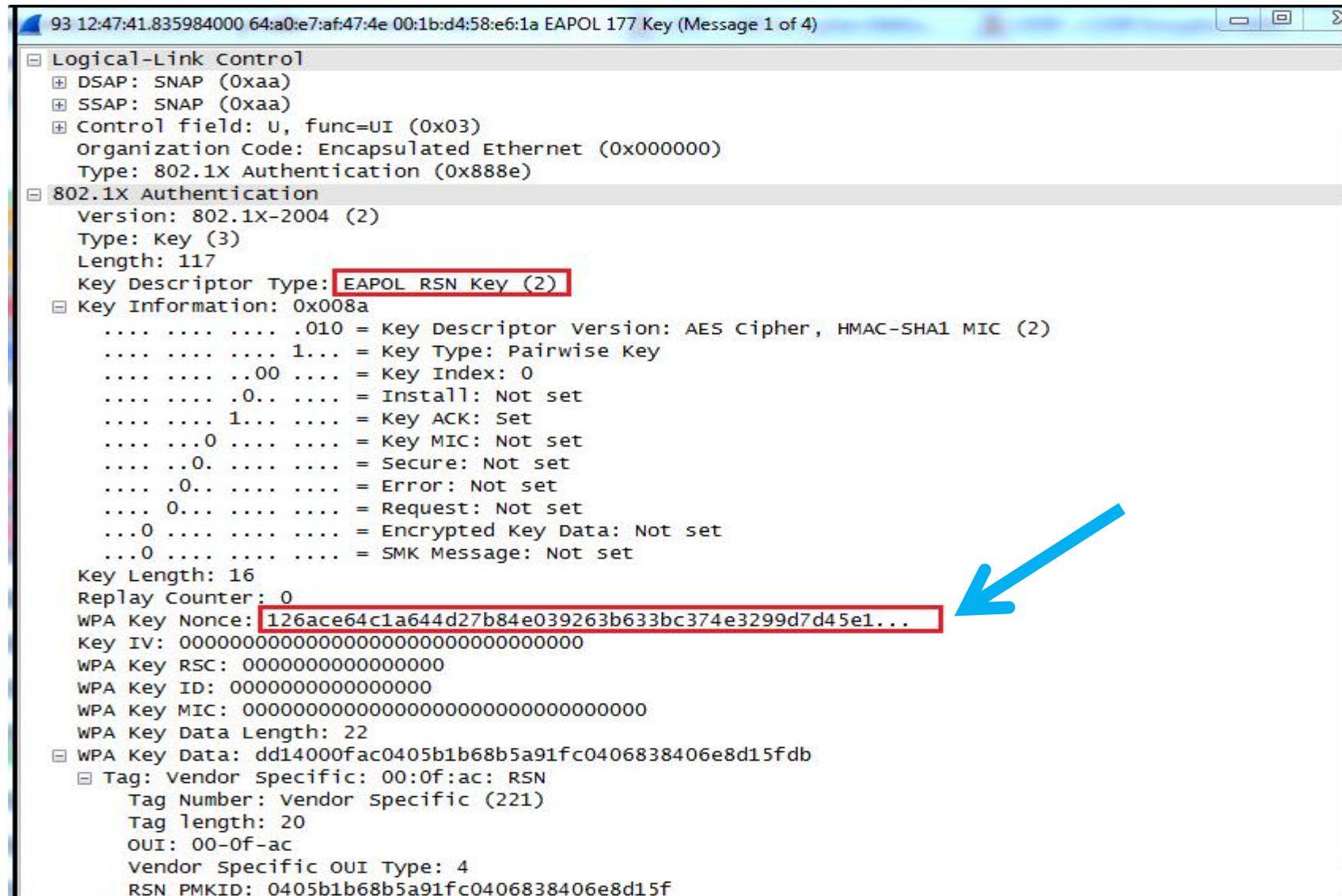
ANonce是以明文方式传送的，如果ANonce被攻击者替换了，会有什么后果？密钥建立过程会成功吗？

■如果ANonce被修改了，会导致STA和AP计算出的PTK不一致，从而KCK不一致，最终导致计算出的MIC不能通过验证。标准规定，如果接收方的MIC验证没有通过，则对应的消息必须丢弃。因此，虽然攻击者能够修改ANonce，但是握手过程不会成功。

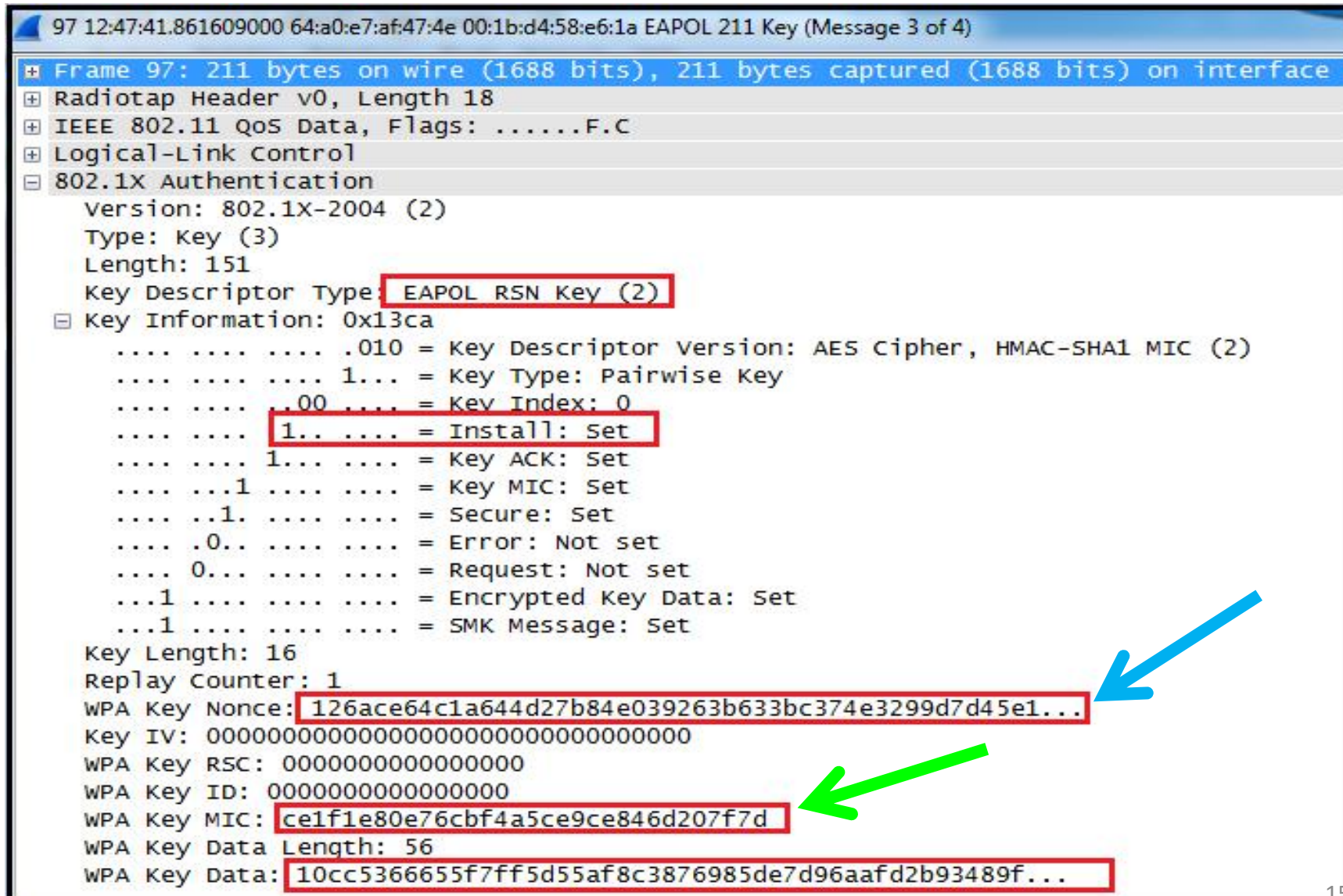
问题讨论1:

■此外，作为一种额外的防御措施，第三个握手包也包含了ANonce。接收方必须验证在第三个握手包中的ANonce和第一个握手包里面的ANonce是一样的，而第三个握手包是通过MIC保护的EAPOL-key帧，因此如果第三个握手包通过MIC验证则表示传输的ANonce是没有被篡改，进而如果第三个握手包中的ANonce和第一个握手包中的ANonce一致，则表示第一个握手包中的ANonce是正确的。

Message 1: WPA Key Nonce



Message 3: WPA Key Nonce



问题讨论2:

有没有可能破解AP和STA上预设的口令呢?



离线字典攻击

MIC的计算过程

- PWD:预设 (WPA/WPA2-PSK)
- $PMK = PBKDF2(HMAC-SHA1, PWD, SSID, 4096, 256)$
- $PTK = PRF-X(PMK, "Pairwise key expansion",$
 $Min(AA, SPA) || Max(AA, SPA) ||$
 $Min(ANonce, SNonce) || Max(ANonce, SNonce))$
- MIC-KEY = PTK的前16字节, 即: KCK
- $MIC = hmac.new(MIC-Key, 802.1x-data,$
 $hmacFunc).digest()$

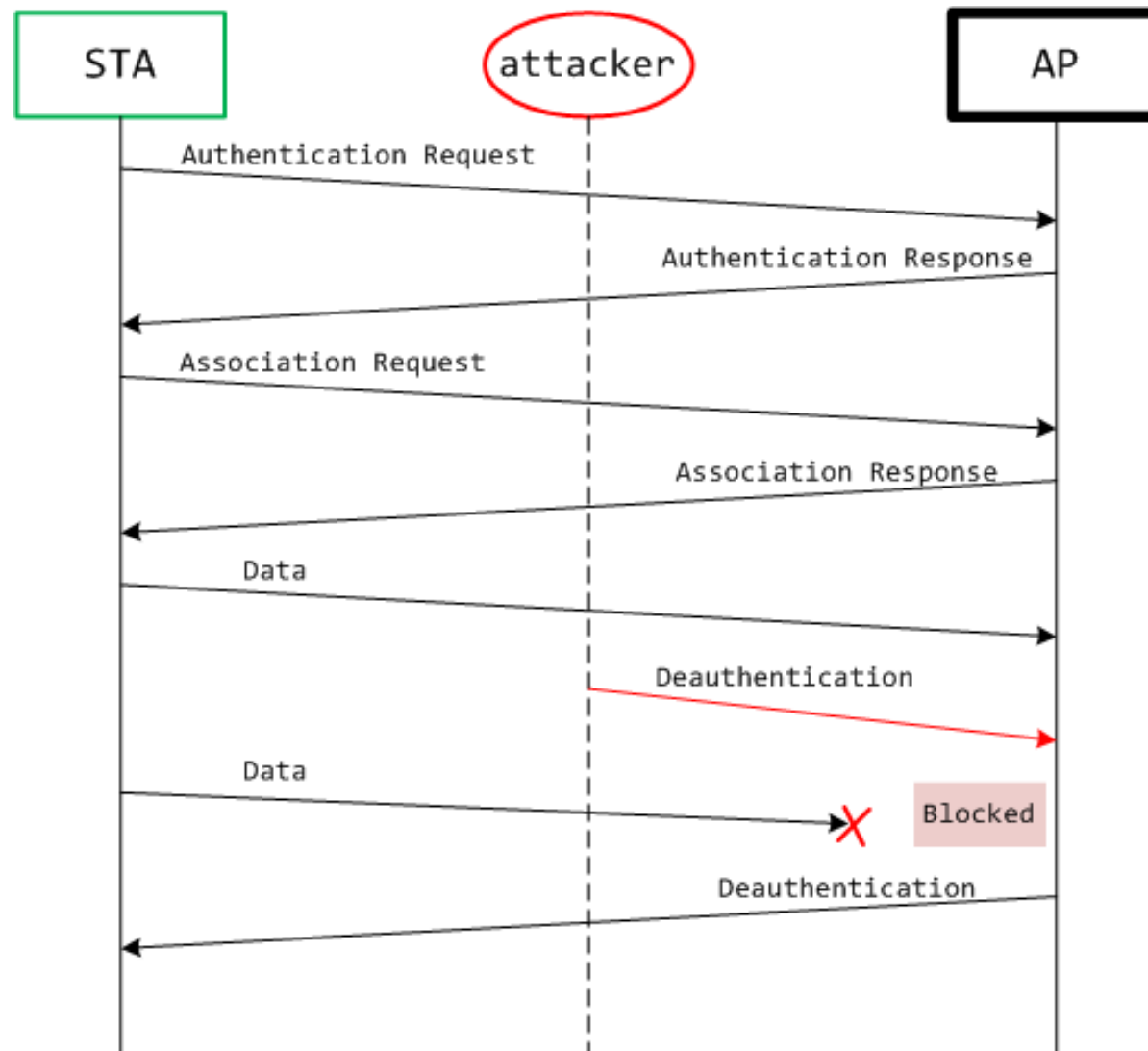
无线网络攻击

攻击1: DeAuth

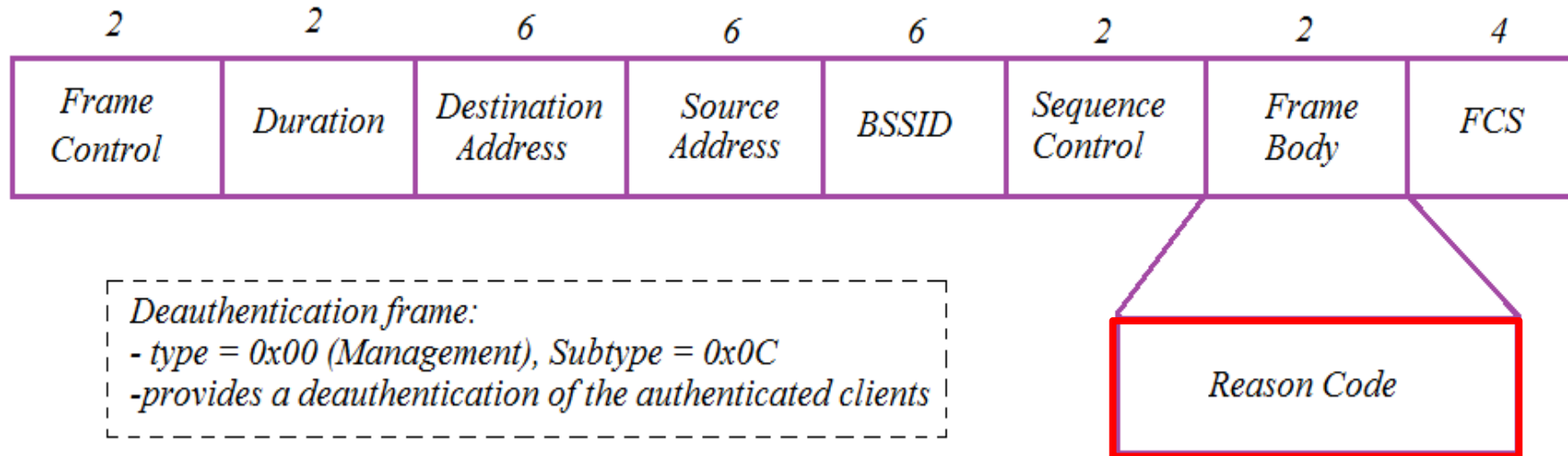
- 伪装成目标 AP 已关联的设备，向 AP 发送 Deauthentication解除认证帧，造成设备掉线，从而达到拒绝服务的目的。



Deauth流程



Deauthentication Frame



- 0 – Reserved
- 1 – Unspecified reason
- 2 – Previous authentication no longer valid
- 3 – Deauthenticated because sending station is leaving (or has left) IBSS or ESS
- 4 – Disassociated due to inactivity

.....

Deauth攻击的用途

- Evil twin access point

- ◆ 强制STA连接到一个假冒的AP

- Password attack

- ◆ 抓取4-way handshake帧

Deauth实施工具

命令:

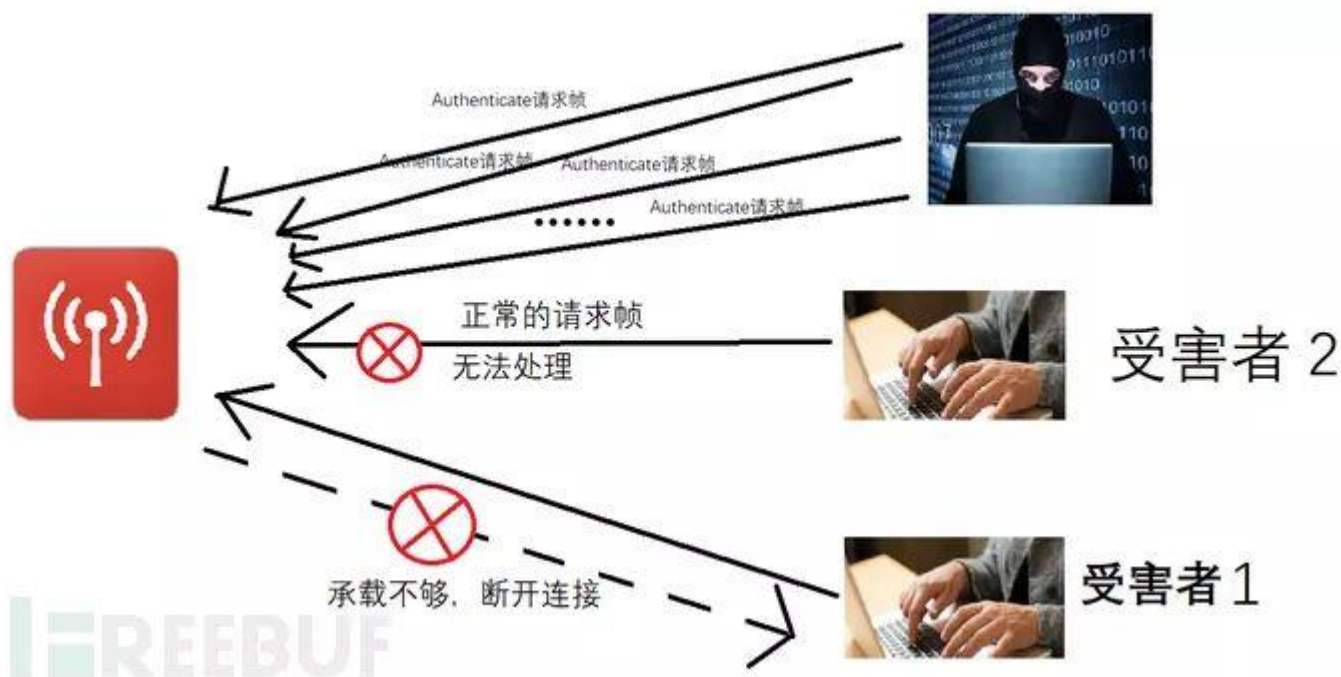
```
aireplay-ng -0 1 -a xx:xx:xx:xx:xx:xx -c yy:yy:yy:yy:yy:yy wlan0
```

参数说明:

- ❑ -0 arms deauthentication attack mode
- ❑ 1 is the number of deauths to send; use 0 for infinite deauths
- ❑ -a xx:xx:xx:xx:xx:xx is the AP (access point) MAC (Media Access Control) address
- ❑ -c yy:yy:yy:yy:yy:yy is the target client MAC address; omit to deauthenticate all clients on AP
- ❑ wlan0 is the NIC (Network Interface Card)

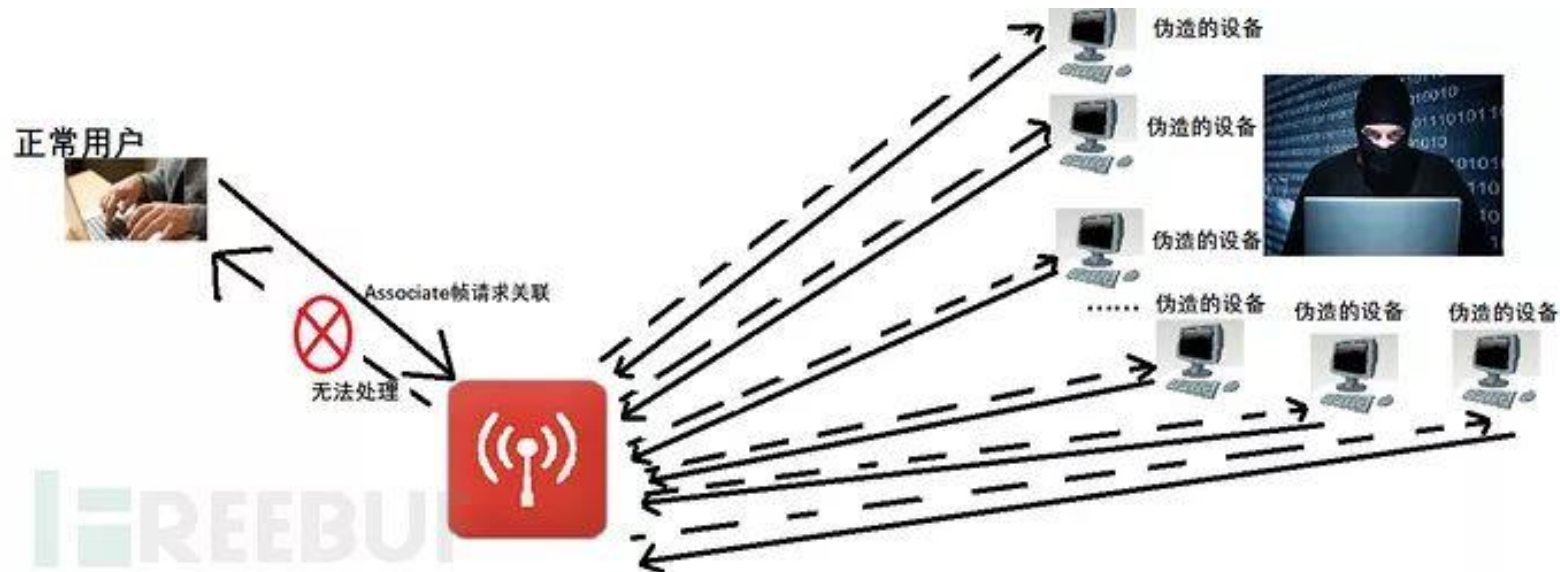
攻击手段2: AuthDos

- 通过随机生成大量mac地址，伪装设备向AP发送大量Authenticattion请求帧，使请求数量超出AP承载能力，从而造成拒绝服务攻击，使正常用户无法连接AP。



攻击3：关联洪水攻击

- 关联洪水攻击，又称asso攻击，主要针对空密码或已破解密码的WLAN，伪造大量设备关联请求，淹没AP的关联表，使正常用户无法与AP建立关联



攻击4: RF jamming Attack

- 不针对管理帧的漏洞，而是对无线信号进行干扰，用噪声信号淹没射频信号导致系统失效。这种干扰会影响到一片区域内指定频带范围的信号。



无线网络防御

无线入侵检测工具

□ WAIDPS

◆ Wireless Auditing, Intrusion Detection & Prevention System

更多信息:

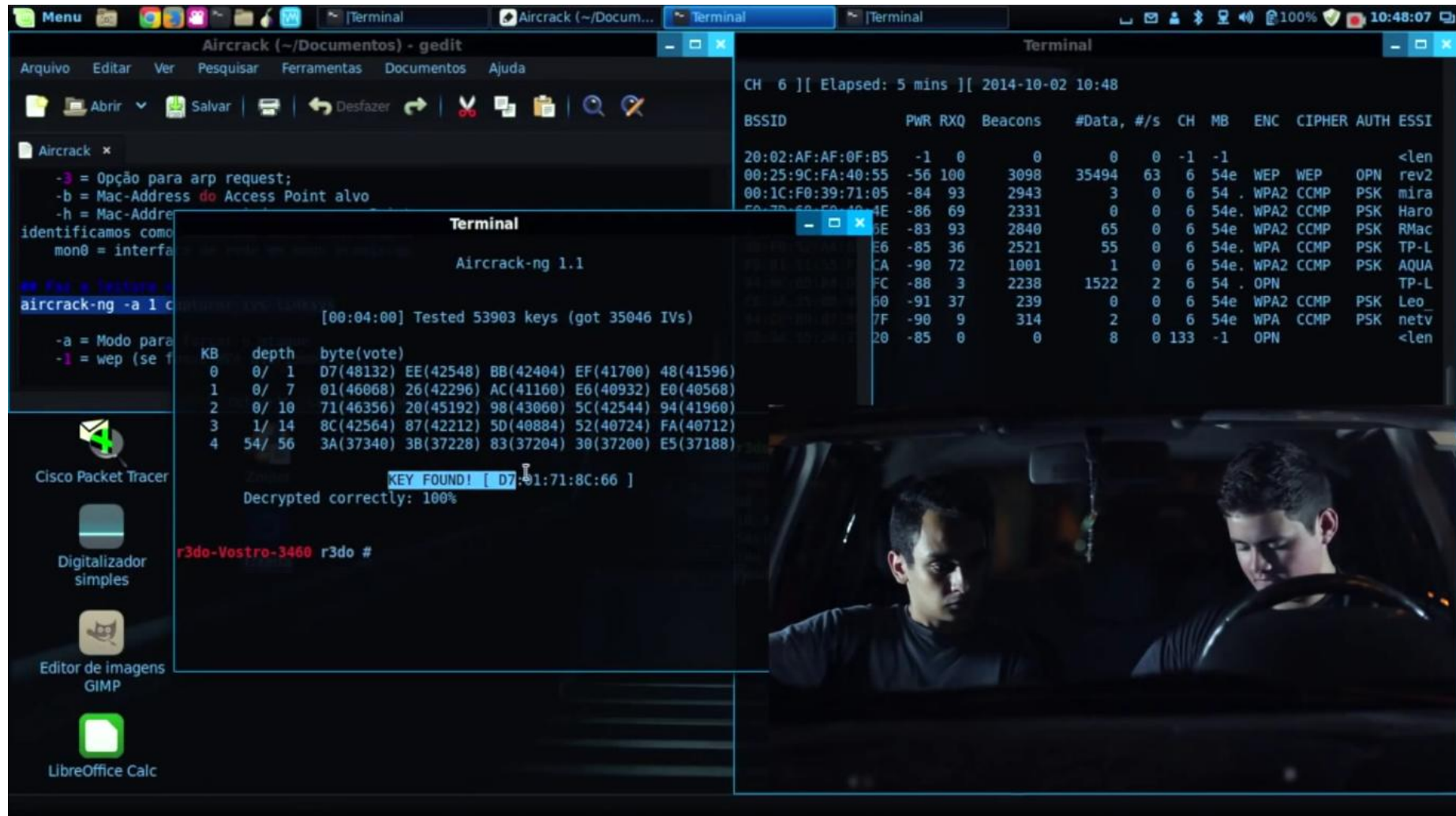
<https://github.com/SYWorks/waidps>

渗透测试工具: AirCRACK-NG

- ❑ A complete suite of tools to assess WiFi network security.
- ❑ Monitoring: Packet capture and export of data to text files for further processing by third party tools
- ❑ Attacking: Replay attacks, deauthentication, fake access points and others via packet injection
- ❑ Testing: Checking WiFi cards and driver capabilities (capture and injection)
- ❑ Cracking: WEP and WPA PSK (WPA 1 and 2)



Movies



更多信息:

<http://www.aircrack-ng.org/index.html>

作业

- 对TKIP和WEP做安全性比较
- 对CCMP和TKIP做安全性比较