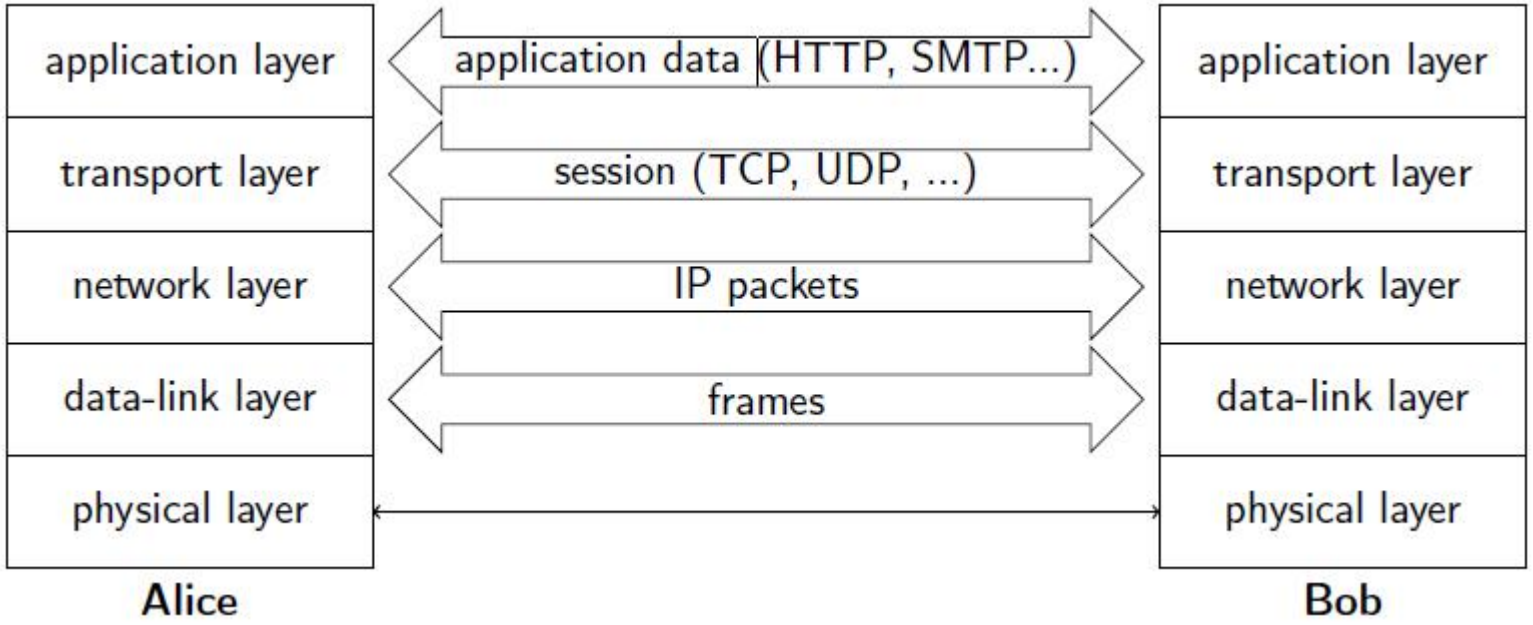
第12章 无线局域网安全



- APplication layer security (SSH, S-MIME, PGP, ……)
- transport layer security (TLS/SSL, ……)
- network layer security (IPsec, ……)
- data-link layer security (有线等效保密(WEP)协议, WPA, WPA2, ……)

主要内容

- 无线局域网简介
- 有线等效保密(WEP)协议
- WPA/WPA2原理
- 无线局域网攻防

无线局域网简介

WLAN组件：由站(Station，STA（站点）)、接入点(Access Point，AP（无线接入点）)、无线介质(Wireless Medium，WM)和分布式系统(Distribution System，DS（分布式系统）)组成。

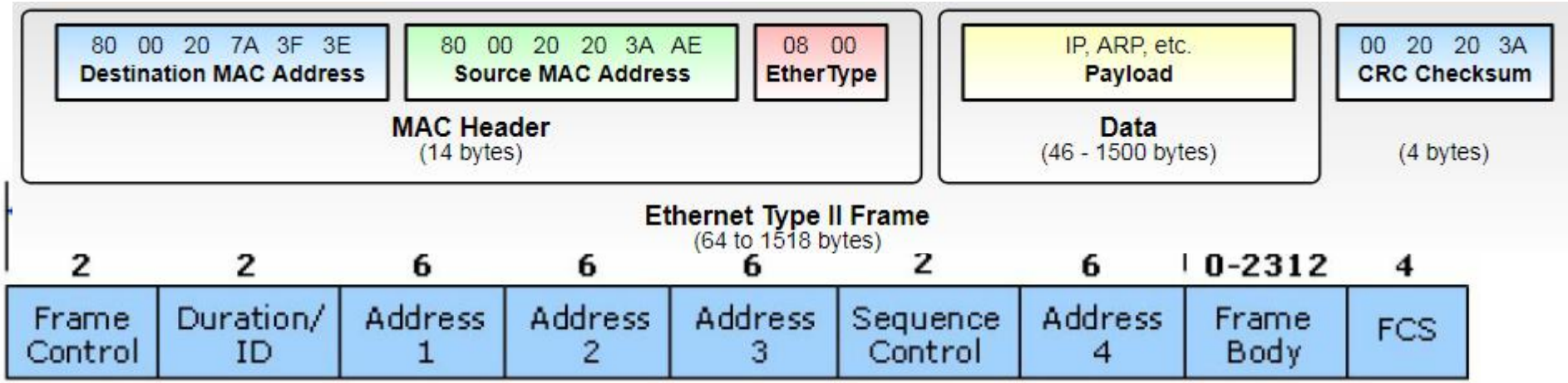
用于连接一系列基本服务集（Basic Service Sets，**BSSs**）和局域网，从而创建一个扩展服务集（Extended Service Set，**ESS**）的系统。

WLAN的基本工作原理

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

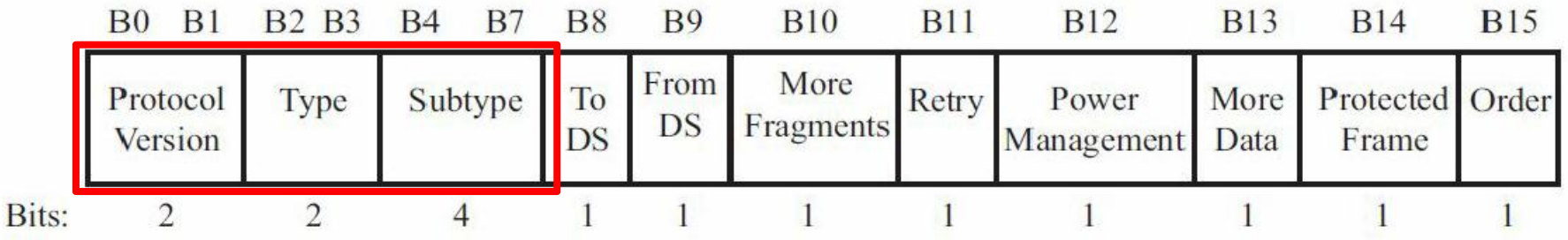
管理帧　　控制帧　　数据帧

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

Frame Control域



- 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。

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应答，借此促进工作站间数据传输的可靠性。

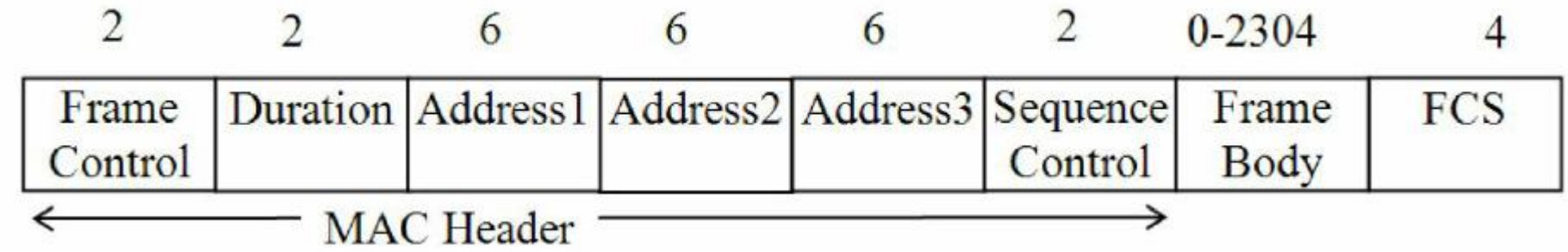
802.11 数据帧

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。
- 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（站点）互相认证的方法，常用于Mesh BSS网络。
  - 65535：代表厂商自定义算法。

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（无线接入点）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（站点）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）。一般该字段会设置为100个TU。

802.11管理帧：定长字段（3）

- CAP（无线接入点）ability Information（能力信息）：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

802.11常用管理帧：Beacon帧

- AP（无线接入点）通过定时发送Beacon帧来声明某个无线网络，STA（站点）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：TimeSTA（站点）mp、Beacon Interval、CAP（无线接入点）ability、SSID

802.11常用管理帧：Probe Request/Response帧

- STA（站点）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（无线接入点）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.

802.11常用管理帧：Association Request帧

- 当STA（站点）要关联某个AP（无线接入点）时，发送Association Request帧。该帧携带的主要信息如下：
  - CAP（无线接入点）ability：AP（无线接入点）将检查该字段来判断STA（站点）是否满足要求
  - Listen Interval：指STA（站点）两次苏醒之间，跳过多少beacon帧
  - SSID：AP（无线接入点）将检查SSID是否为自己所在网络
  - Supported Rates: AP（无线接入点）将检查该字段是否满足要求

802.11常用管理帧：Association Response帧

- 针对Association Request帧，AP（无线接入点）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：
  - CAP（无线接入点）ability: AP（无线接入点）设置的CAP（无线接入点）ability
  - STA（站点）tus Code: AP（无线接入点）返回的关联请求处理结果
  - AID：AP（无线接入点）返回关联ID给STA（站点）
  - Supported Rates: AP（无线接入点）支持的传输速率

802.11管理帧：authentication确认帧

- 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
  - STA（站点）tus 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

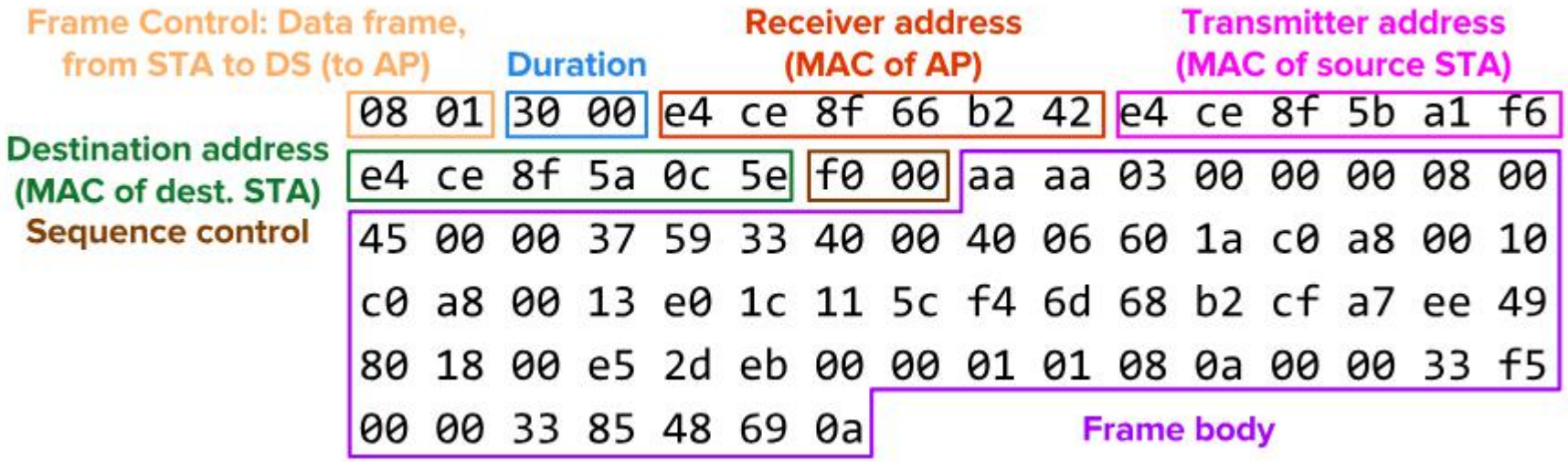
802.11管理帧：总结

- 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。

地址字段

- Receiver Address（RA）：用于描述接收MAC数据帧的接收者地址，可以是STA（站点）或者AP（无线接入点）。
- 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（站点）或者AP（无线接入点）。
- 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（站点）地址。一般情况下都是单播地址。

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

WLAN安全

WLAN的安全发展过程

- 有线等效保密(WEP)协议（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，……

有线等效保密(WEP)协议

Wired Equivalent Privacy

有线等效保密

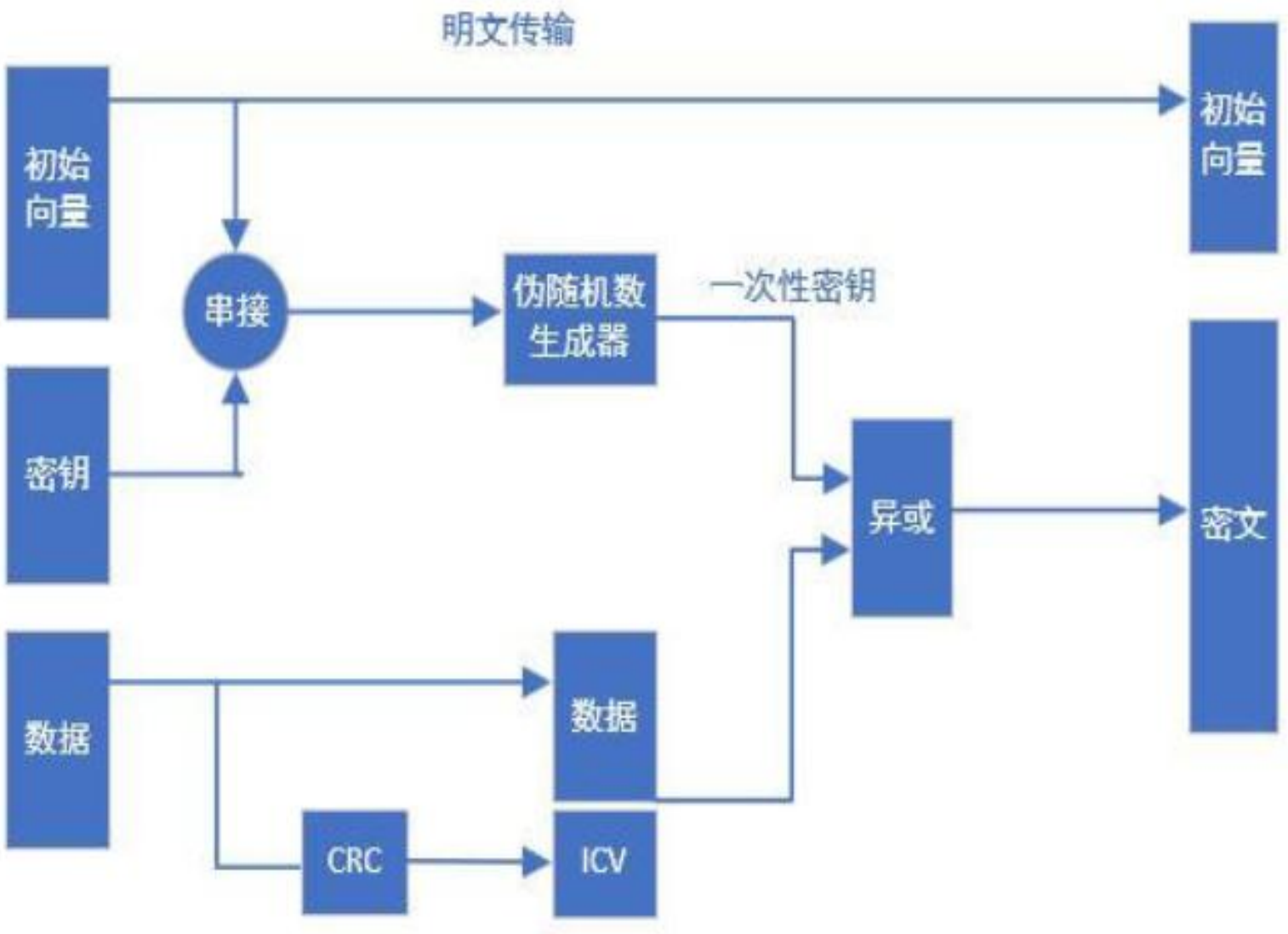
有线等效保密(WEP)协议的安全服务

- 身份认证
- 保密性
- 完整性

有线等效保密(WEP)协议加密封装

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

有线等效保密(WEP)协议：WEP加密过程



1，先串接，构成64位的随机数种子。

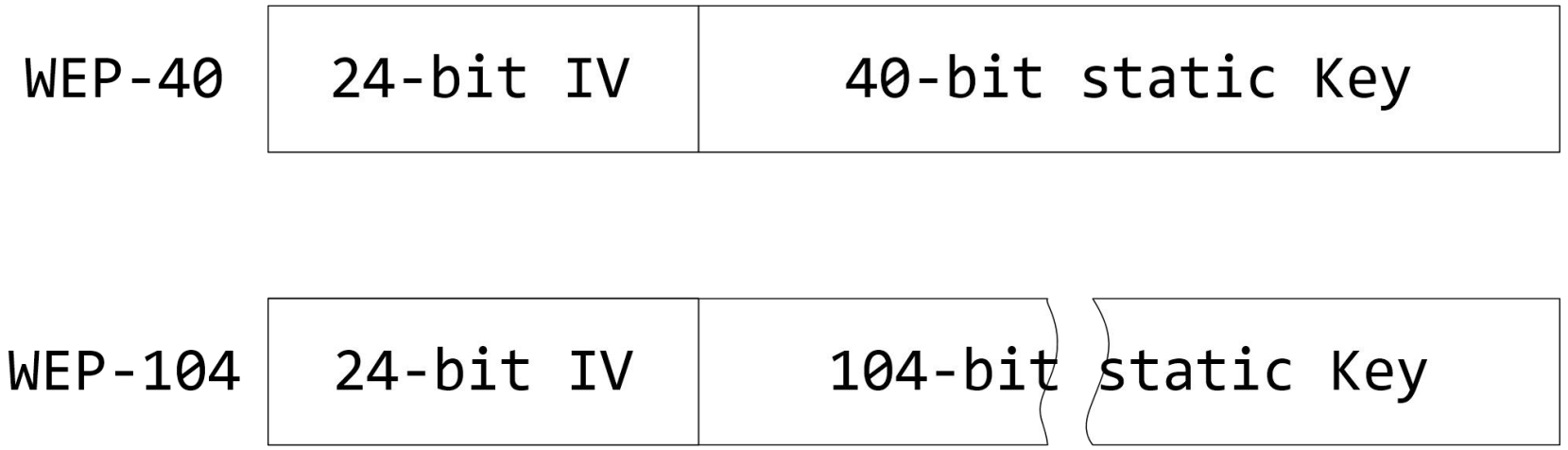
2，产生一次性密钥，一次性密钥的长度=数据长度+4，增加的4字节是用于完整性检测的32位循环冗余检验码。

3，数据和4B的完整性检验值（ICV）与长度相同的一次性密钥经过异或运算得到密文。

有线等效保密(WEP)协议加密流程

1. 生成24bit随机数作为IV，与有线等效保密(WEP)协议 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。

RC4 种子

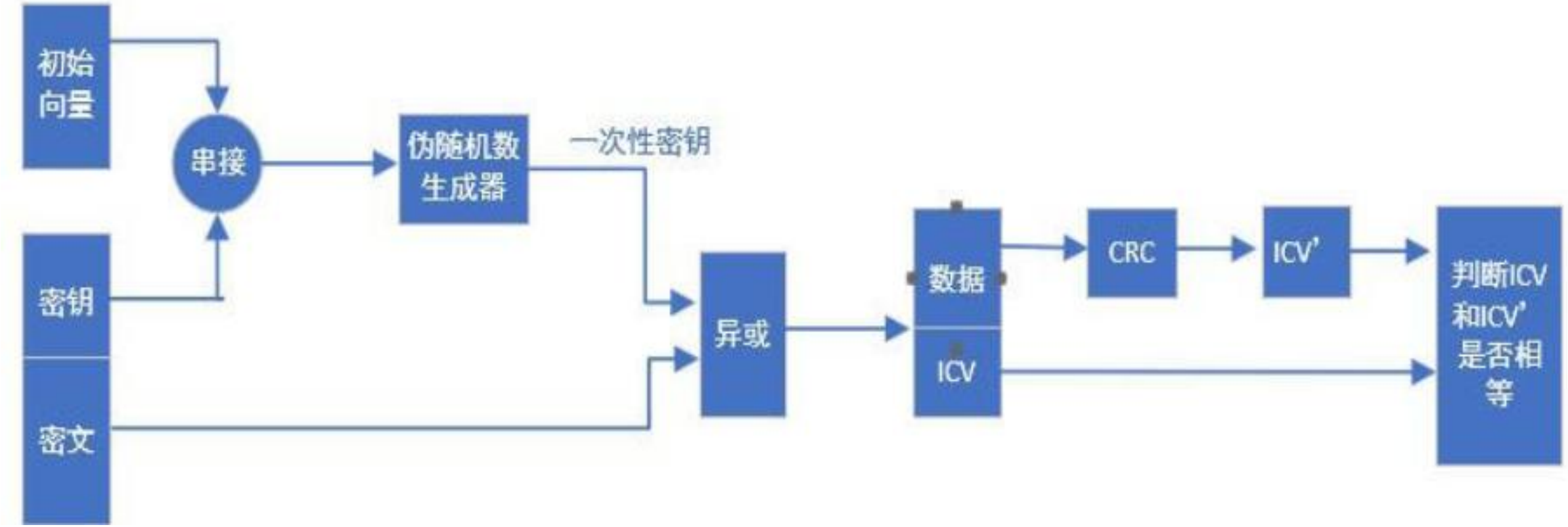


802.11标准定义了两个有线等效保密(WEP)协议版本：

◆有线等效保密(WEP)协议-40，24-bit IV || 40-bit STA（站点）tic Key (10 HEX characters or 5 ASCII characters)构成64bit ARC4的种子，生成密钥流

◆有线等效保密(WEP)协议-104，24-bit IV || 104-bit STA（站点）tic Key (26 HEX characters or 13 ASCII characters)构成128-bit ARC4的种子，生成密钥流

有线等效保密(WEP)协议：完整性验证过程



1，密文和一次性密钥异或运算得到数据和4位完整性检验值；

2，根据数据和生成函数G（x）计算出数据的32位循环冗余检验码，并把计算的结果和MAC携带的完整性检验值进行对比，如果相等，则通过完整性检测。

有线等效保密(WEP)协议：鉴别机制

1，开放系统鉴别机制并不对终端进行鉴别，只要终端向AP发送鉴别请求帧，AP一定向终端回送表示鉴别成功的鉴别响应帧。

2，共享密钥鉴别机制确定终端是否为授权终端的唯一依据就是终端是否拥有和AP一样的共享密钥

有线等效保密(WEP)协议：鉴别机制

终端

AP

认证请求

认证响应（challenge）

认证请求（加密后密文）

认证响应（成功）

1 challenge是128字节长度的随机数。

2 密文＝（challenge||ICV）⊕ K；

K是一次性密钥，以IV和密钥为伪随机数种子生成的伪随机数，其长度＝128＋4（单位字节）。

有线等效保密(WEP)协议：鉴别机制

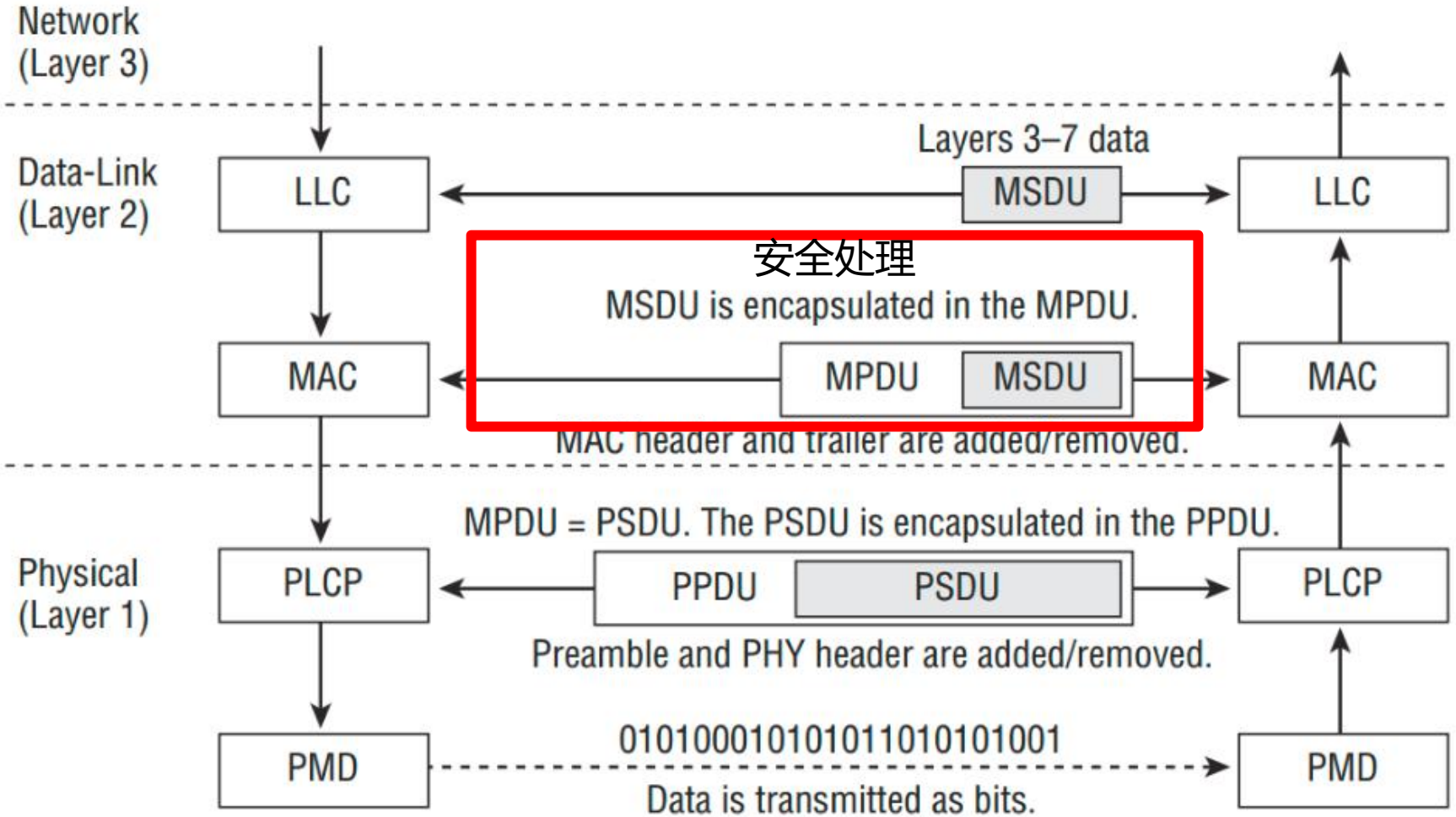
AP事先建立访问控制列表，表中给出授权终端的MAC地址。当某个终端接入AP前，先向AP发送鉴别请求帧，鉴别帧中指明鉴别机制时MAC地址鉴别。

AP用该MAC地址的源MAC地址去**检索**访问控制列表，如果在访问控制列表中找到匹配的MAC地址，发送鉴别成功的鉴别响应帧。

有线等效保密(WEP)协议：关联的接入控制功能

列表：该项内容包含终端的MAC地址、身份鉴别方式、是否支持查询、支持的物理层标准、数据传输速率和关联寿命等信息。

WLAN的安全处理功能



有线等效保密(WEP)协议：WEP的安全缺陷

1.共享密钥鉴别机制的安全缺陷

2.完整性检测缺陷

3.静态秘钥管理的缺陷

有线等效保密(WEP)协议：WEP的安全缺陷

1.共享密钥鉴别机制的安全缺陷

入侵终端通过侦听得到随机数P（这是前面的共享秘钥鉴别机制），授权终端发送给AP的密文，入侵终端以此可以得出此次加密的一次性密钥和初始向量IV。

入侵终端共享秘钥鉴别机制时，发起鉴别过程，因它侦听得到的一次性秘钥和初始向量都是有效的，所以能通过AP对它的**身份鉴别**。

有线等效保密(WEP)协议：鉴别机制

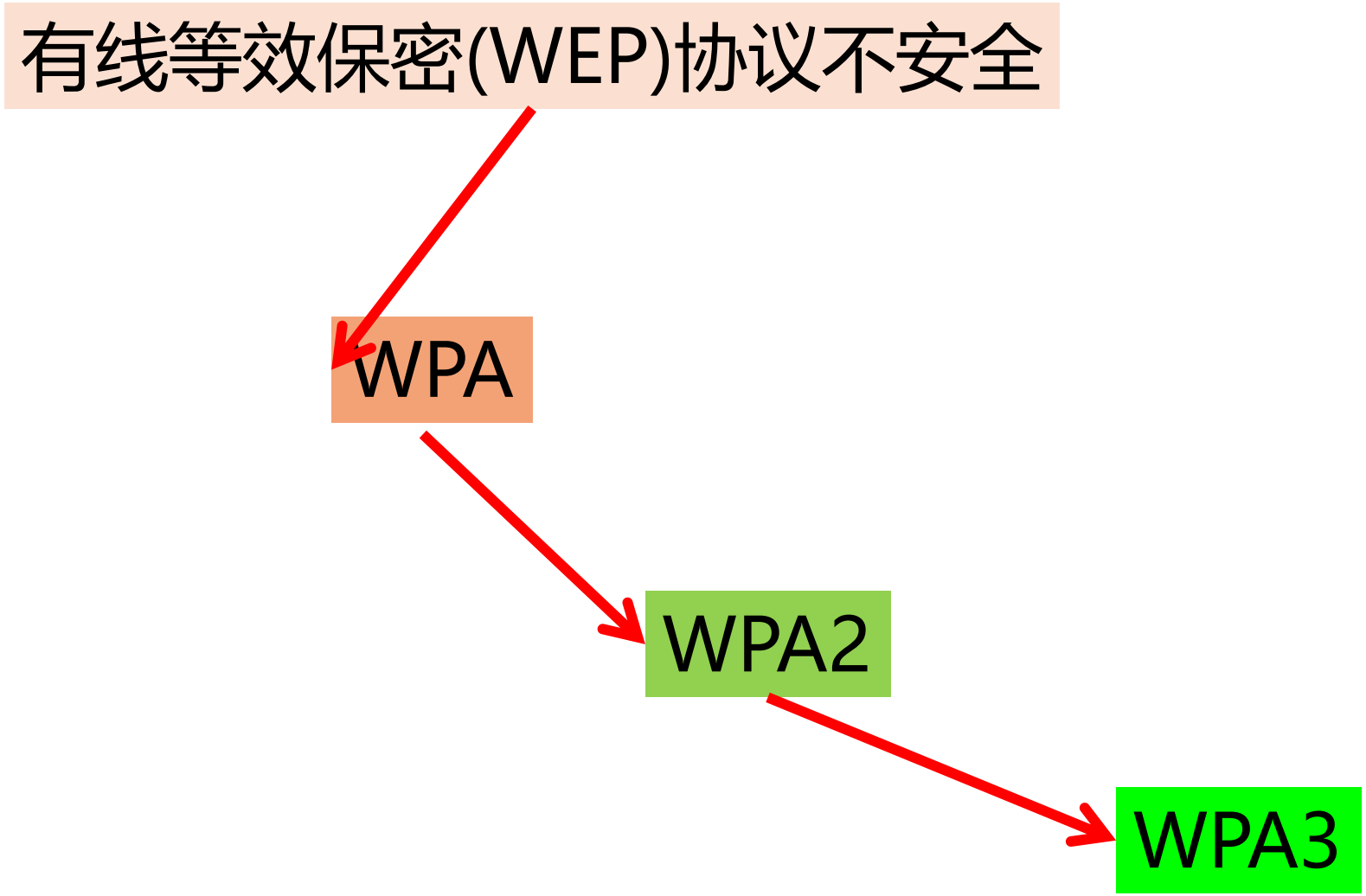
2.完整性检测缺陷

入侵终端将侦听到的密文和自己篡改的密文数据（4位ICV用篡改数据除以和终端使用的生成函数得到）**进行异或**，得到篡改后的密文，然后用一次性密钥和篡改后的密文进行异或，得到数据明文和ICV

有线等效保密(WEP)协议：WEP的安全缺陷

3.静态秘钥管理的缺陷

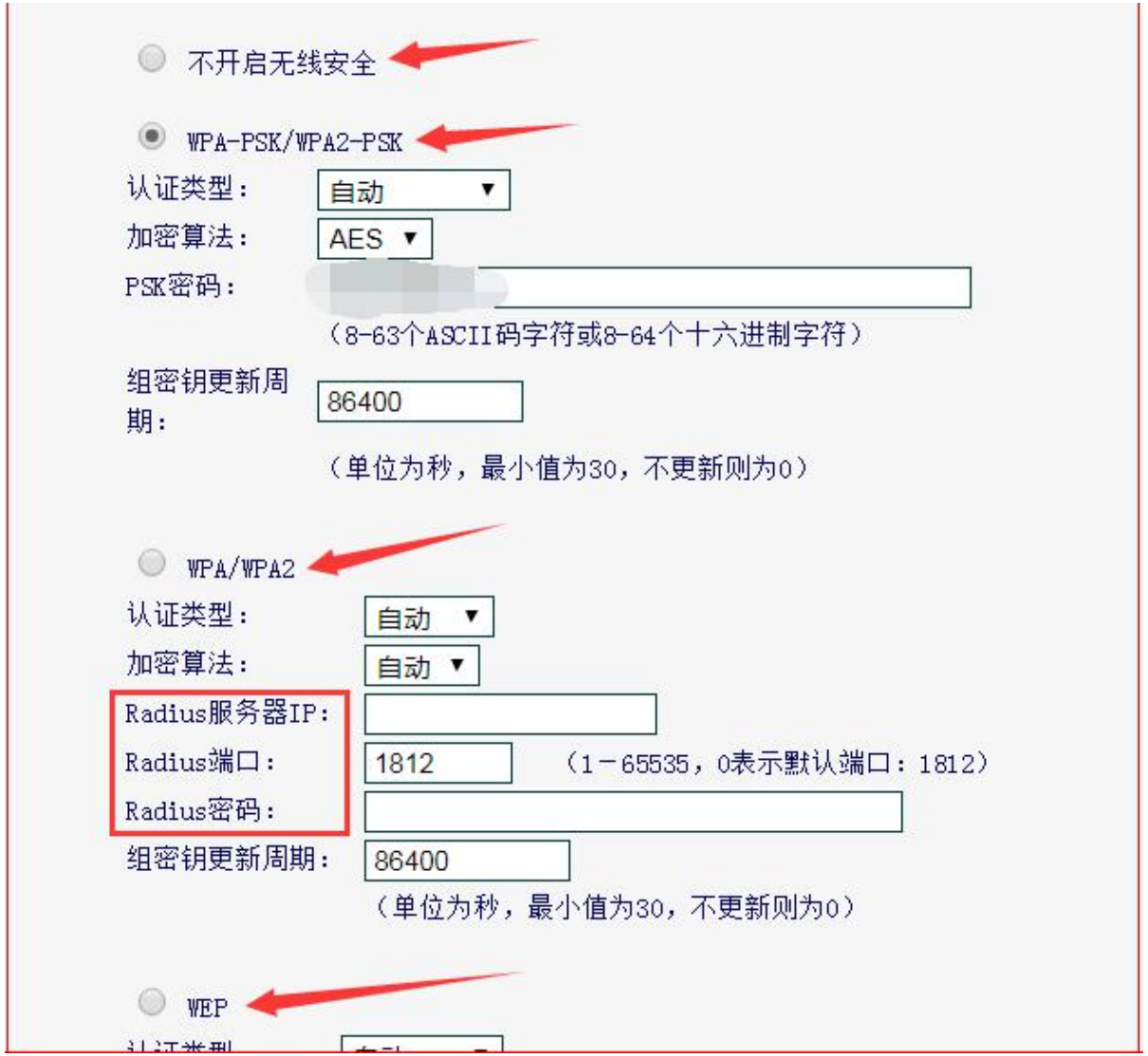
WEP要求属于同一BSS的所有终端**共享同一密钥**，由于一次性密钥只与IV相关，所以属于同一BSS的终端，在保持密钥不变的情况下，只能共享2^24个一次性密钥，导致**重复使用一次性密钥的几率大大增加**，影响数据的安全传输。


有线等效保密(WEP)协议不安全
WPA
WPA2
WPA3

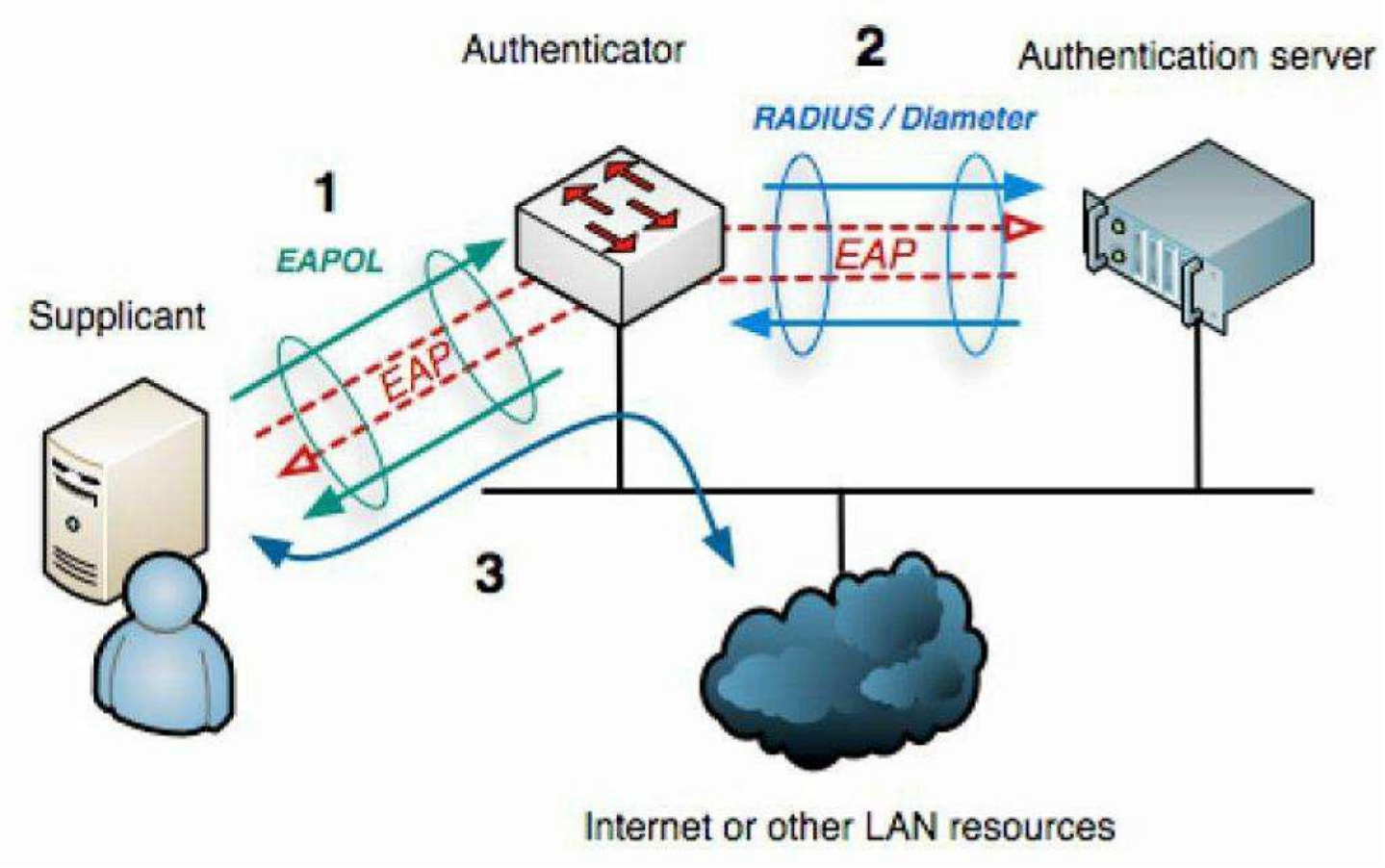
WPA

(Wireless Protected Access)

无线安全设置界面

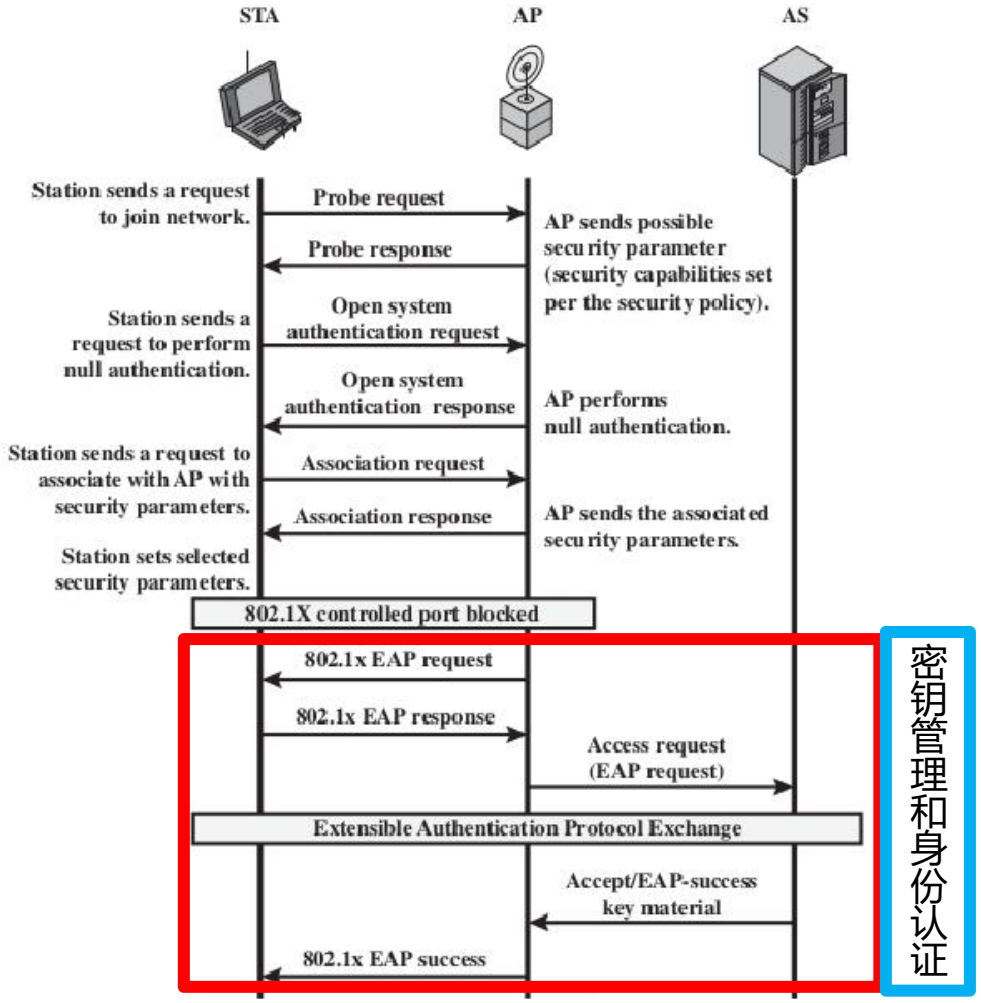

不开启无线安全
WPA-PSK/WPA2-PSK
认证类型：
自动
加密算法：
AES
PSK密码：
（8-63个ASCII码字符或8-64个十六进制字符）
组密钥更新周期：
86400
（单位为秒，最小值为30，不更新则为0）
WPA/WPA2
认证类型：
自动
加密算法：
自动
Radius服务器IP：
Radius端口：
1812
（1－65535，0表示默认端口：1812）
Radius密码：
组密钥更新周期：
86400
（单位为秒，最小值为30，不更新则为0）
WEP

802.1X/EAP架构



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

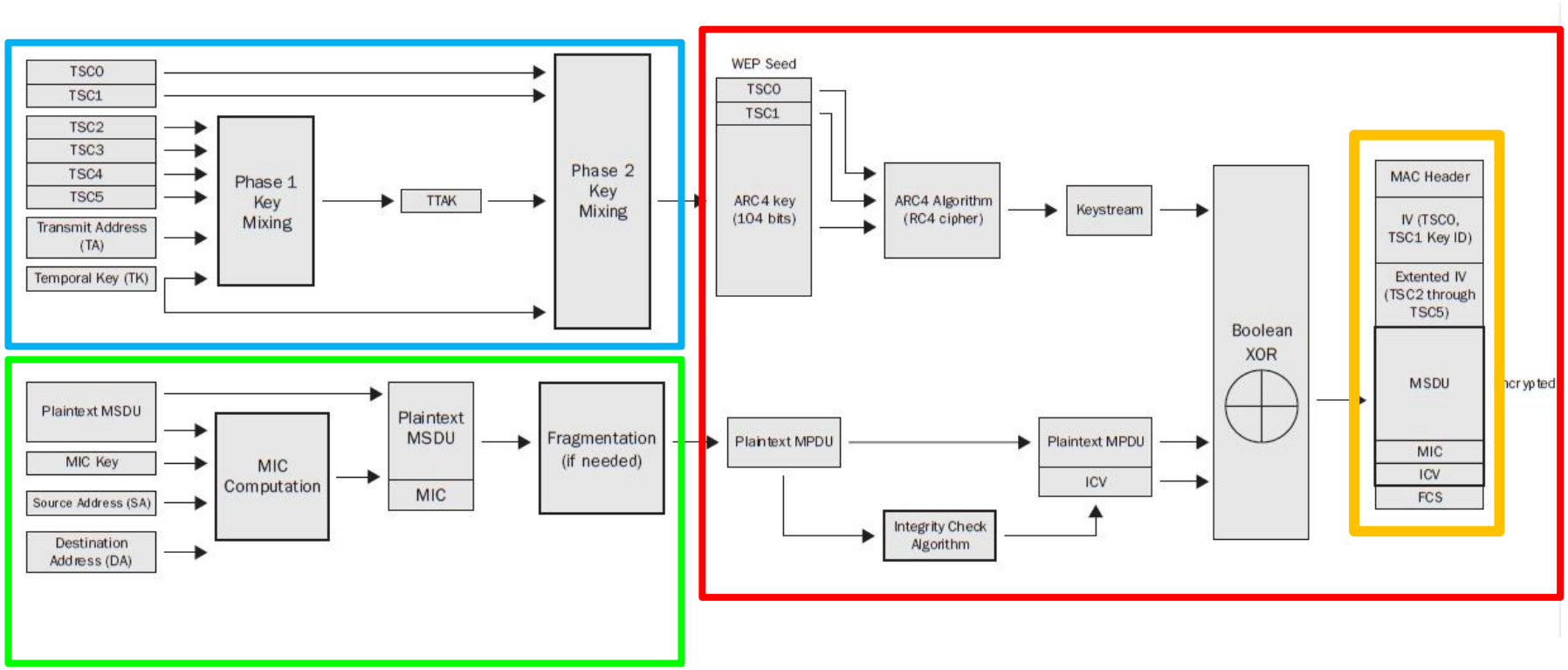
802.11i运行过程


STA
AP
AS
Station sends a request to join network.
Probe request
Probe response
AP sends possible security parameter (security capabilities set per the security policy).
Station sends a request to perform null authentication.
Open system authentication request
Open system authentication response
AP performs null authentication.
Station sends a request to associate with AP with security parameters.
Association request
Association response
AP sends the associated security parameters.
Station sets selected security parameters.
802.1X controlled port blocked
802.1x EAP request
802.1x EAP response
Access request (EAP request)
Extensible Authentication Protocol Exchange
Accept/EAP-success key material
802.1x EAP success
密钥管理和身份认证

WPA: TKIP

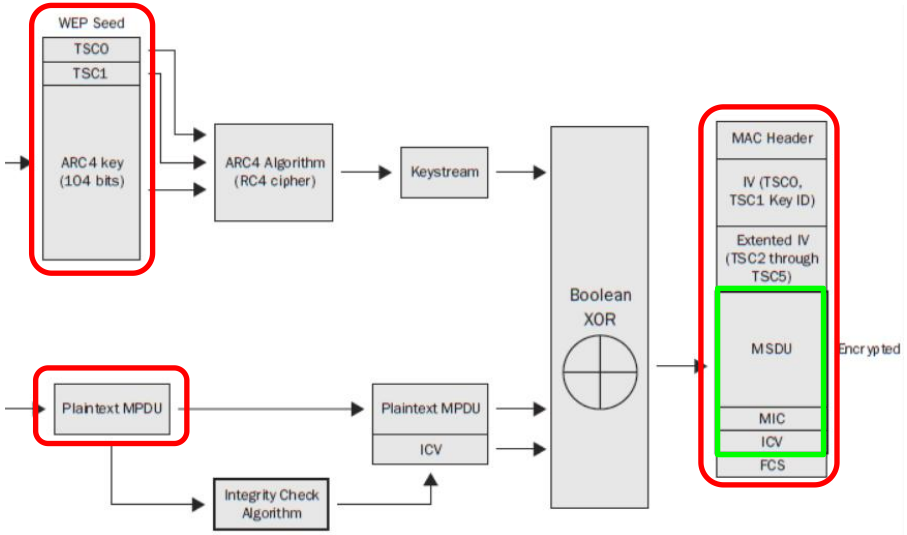
- TKIP：Temporal Key Integrity Protocol
- 基于802.11i draft
- 对有线等效保密(WEP)协议的改进
  - 只需更新固件

TKIP Encryption & Data Integrity Process



可划分为三部分

TKIP加密过程



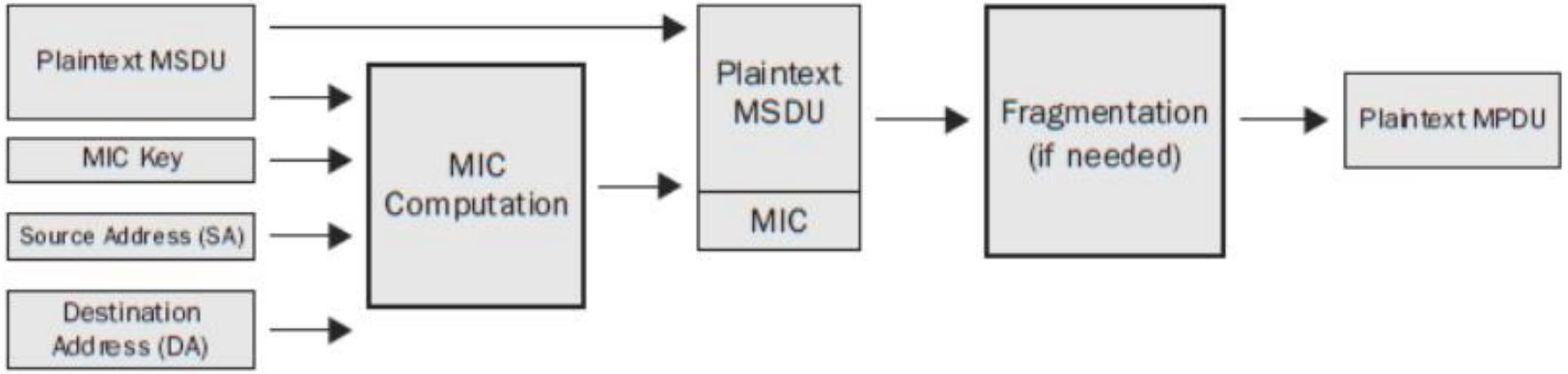
对照有线等效保密(WEP)协议，改变的地方包括：

1、有线等效保密(WEP)协议 Seed的生成

2、Plaintext MPDU(MAC协议数据单元)

3、帧封装。有线等效保密(WEP)协议在MAC Header后面紧随4-octet的IV字段，然后是加密的MSDU(MAC服务数据单元)||ICV；TKIP的MAC Header后面紧随8个octet的(IV||Extended IV)，然后是加密的MSDU(MAC服务数据单元)||MIC||ICV。

TKIP：Plaintext MPDU(MAC协议数据单元)生成



输入：

- Plaintext MSDU(MAC服务数据单元)：未加密的MSDU(MAC服务数据单元)
- 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

MIC计算

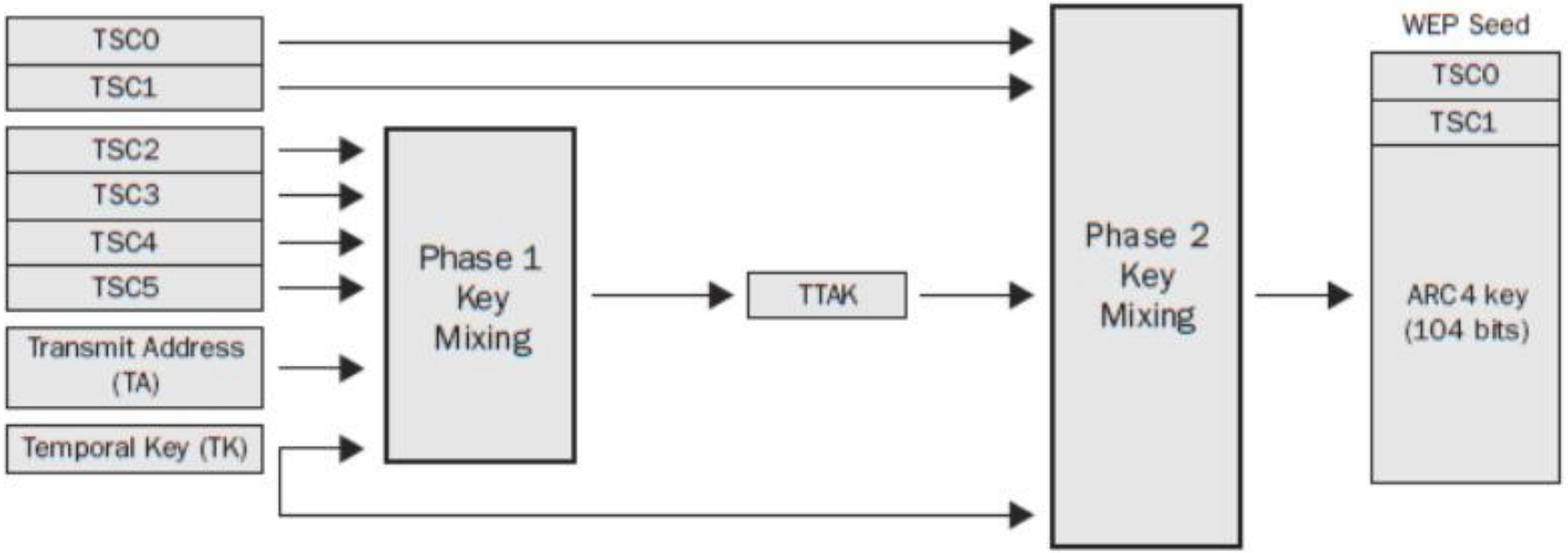
TKIP使用称为Michael算法的Keyed Hash function来生成MIC。Michael的输入为64比特key和任意长度的消息，输出为64比特的Michael值。

MIC Key:

如果是AP（无线接入点）发送给STA（站点），则为TK的128 - 191比特

如果是STA（站点）发送给AP（无线接入点），则为TK的192 - 255比特

TKIP：有线等效保密(WEP)协议 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(MAC协议数据单元)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):Transmitter的MAC地址

TKIP：有线等效保密(WEP)协议 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，生成有线等效保密(WEP)协议 seed = phase2(TTAK, TK, TSC)，128bit，其中又分为IV和104bit的有线等效保密(WEP)协议 key。

TKIP：解密过程

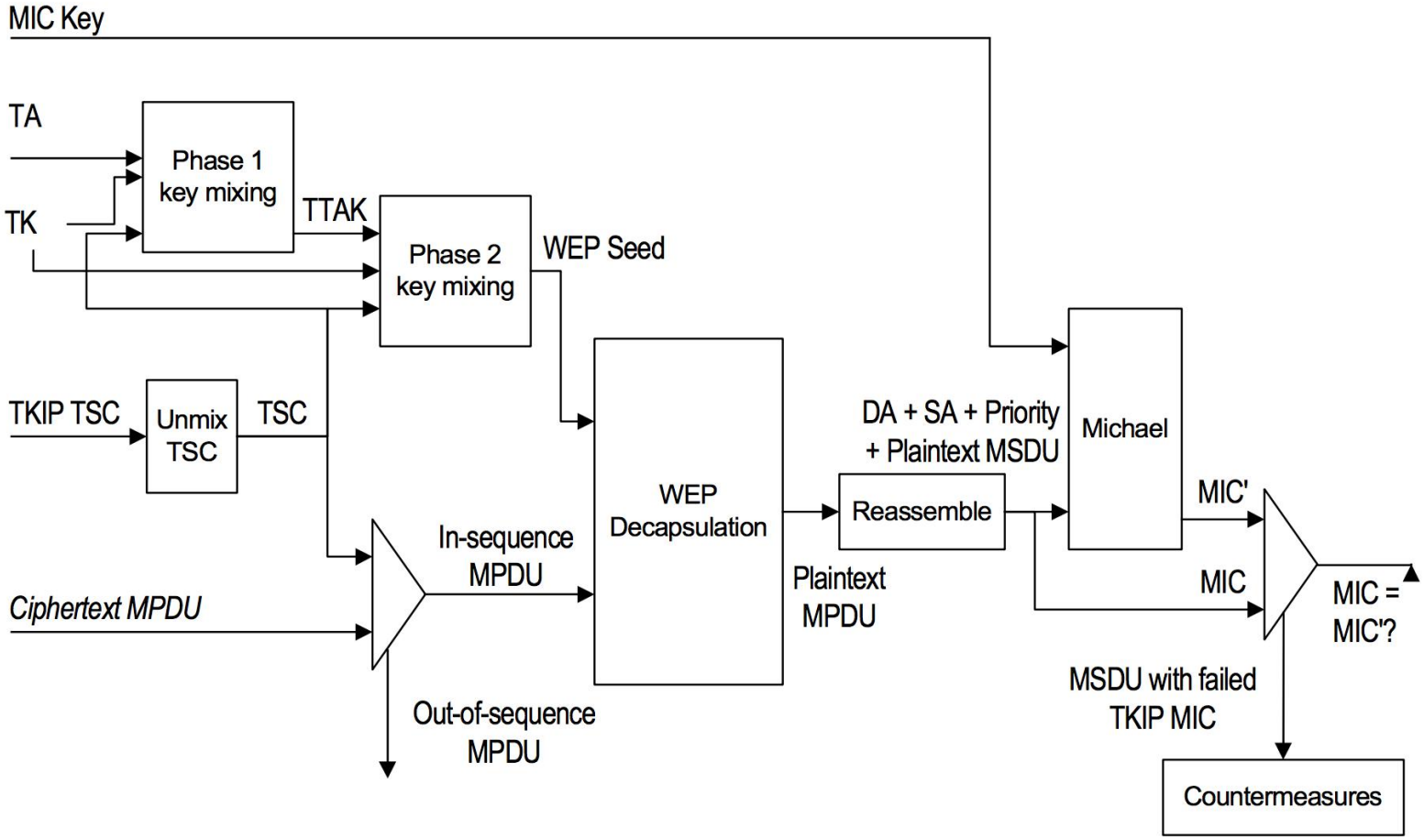


Figure 11-6—TKIP decapsulation block diagram

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 AP（无线接入点）proved, interoperable implementation of the full 802.11i as WPA2, also called RSN (Robust Security Network)
- 有线等效保密(WEP)协议: Pre-RSNA
- WPA/WPA2: RSNA

密钥管理

- 有线等效保密(WEP)协议
  - 所有STA（站点）使用相同的有线等效保密(WEP)协议 Key进行数据加密
  - 安全性较差
- RSNA 密钥管理
  - 关联后，不同STA（站点）和AP（无线接入点）之间使用不同Key进行数据加密，即：Pairwise Key

密钥用途

- KCK: 用于计算WPA EAP（无线接入点）OL-Key消息中的MIC
- KEK: AP（无线接入点）用KEK加密发送给STA（站点）的附加数据，在Key Data字段，比如EAP（无线接入点）OL-key消息中的GTK信息字段
- TK： 用于加/解密单播数据帧
- Tx: 用于计算AP（无线接入点）发送的单播数据帧的MIC
  - Michael MIC Key
- Rx: 用于计算STA（站点）发送的单播数据帧的MIC
  - Michael MIC key

用于TKIP

WPA/WPA2-PSK: PMK

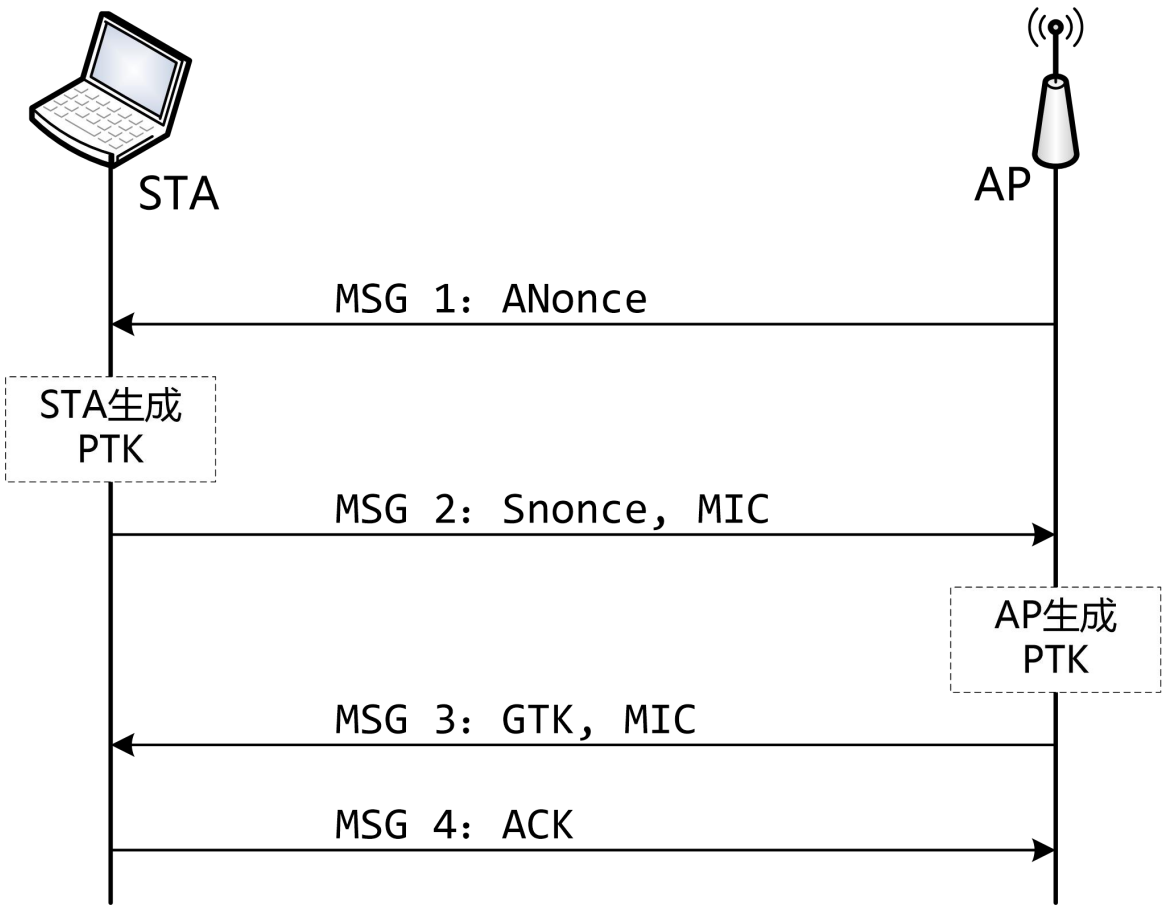
PMK = **PBKDF2**(HMAC–SHA1, PWD, SSID, 4096, 256)

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)
```

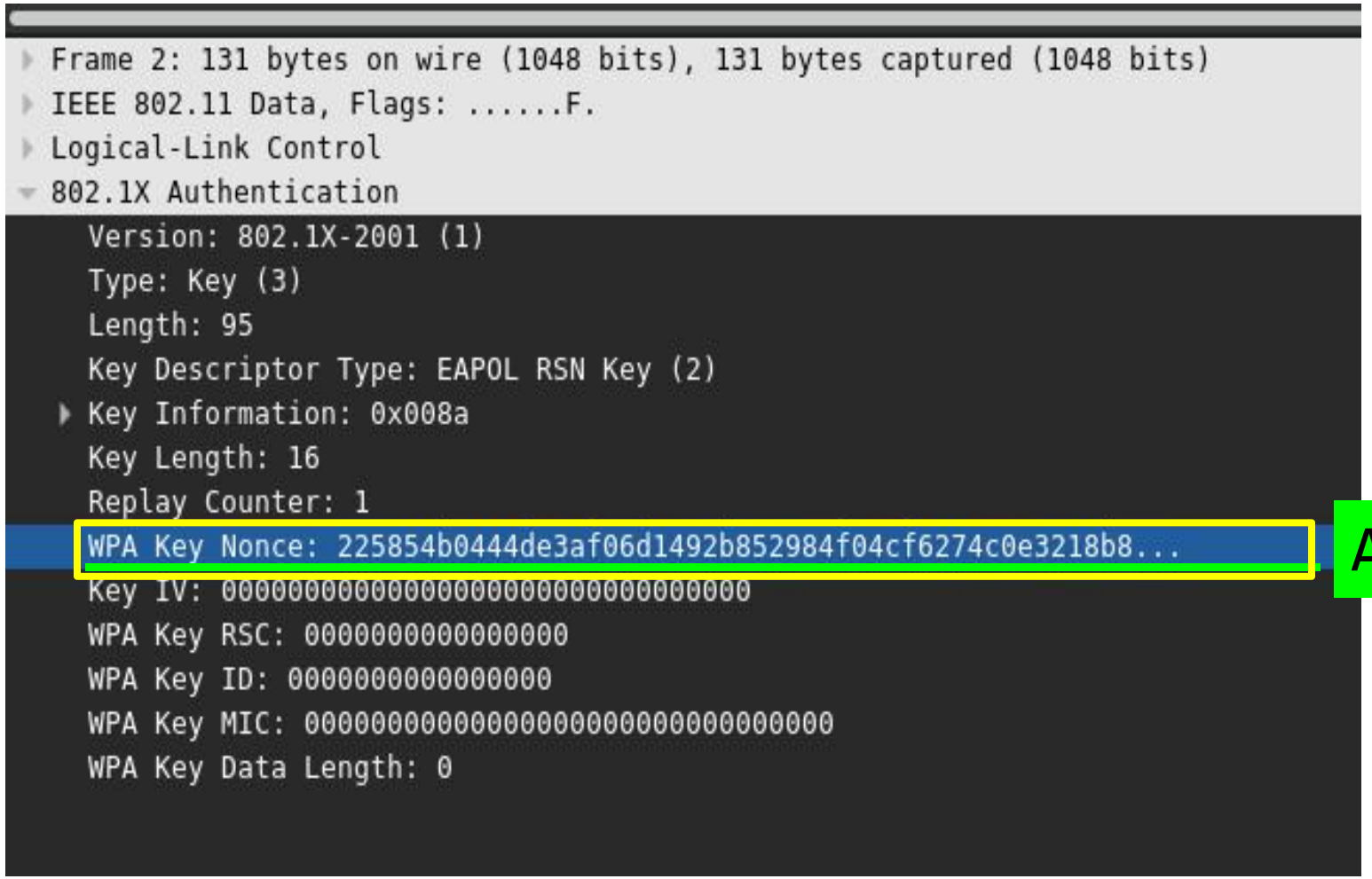
4-WAY HANDS（分布式系统）HAKE

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 HANDS（分布式系统）HAKE: MSG 1

MESSAGE 1: AP（无线接入点）→STA（站点）



ANonce（随机数）

第一条消息后

STA（站点）拥有的信息：

PMK　SNonce（随机数）　ANonce（随机数）　SPA

PTK<—PRF-X(PMK, "Pairwise key expansion", Min(AA,SPA) || Max(AA,SPA) || Min(ANonce（随机数）,SNonce（随机数）) || Max(ANonce（随机数）,SNonce（随机数）))

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

TKIP: TK_bits=256　　CCMP(计数器模式密码块链消息完

PRF() for PTK

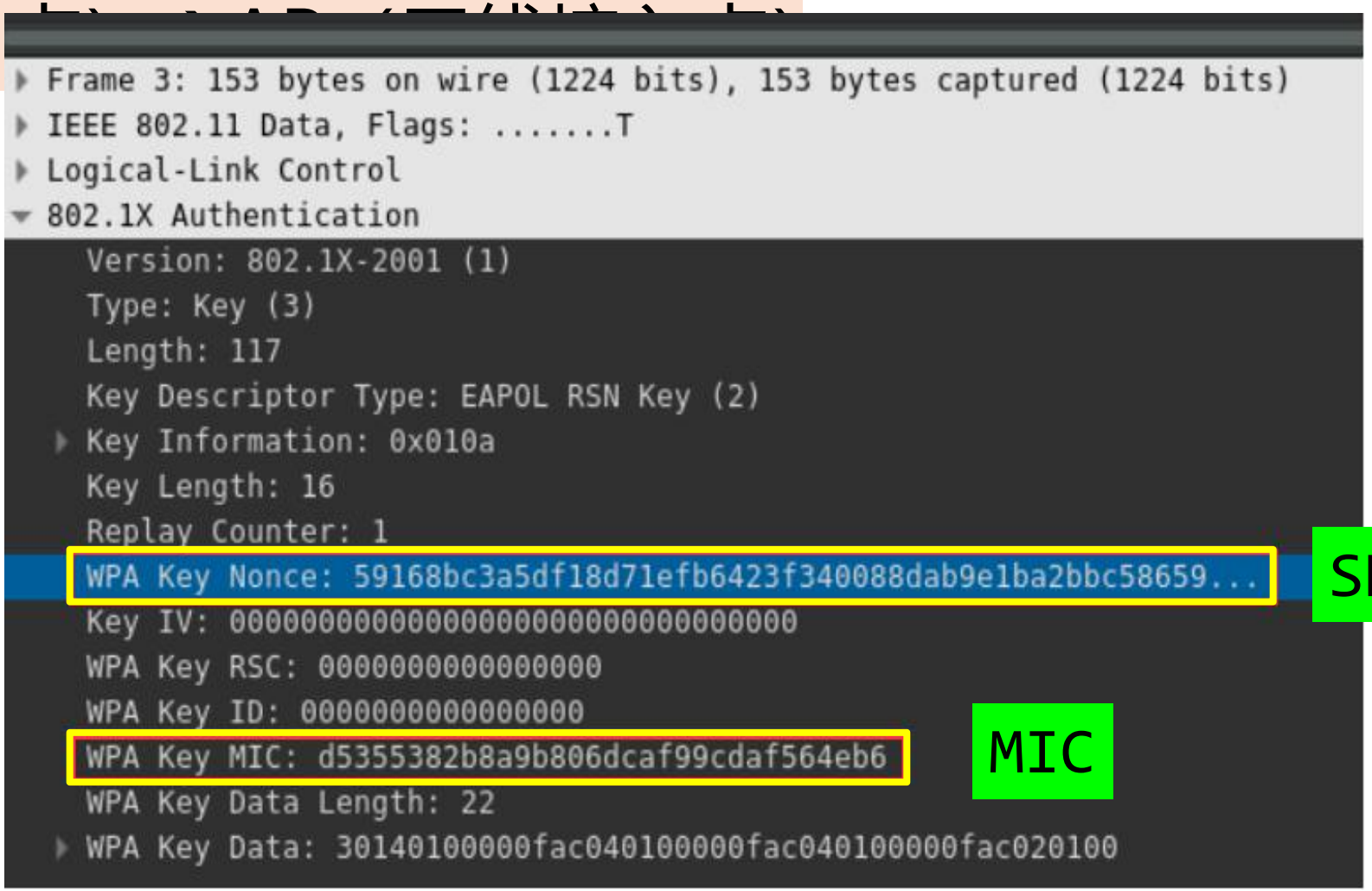
```
#Pseudo-random function for generation of
#the pairwise transient key (PTK)
#key:        The PMK
#A:          b'Pairwise key expansion'
#B:          The AP（无线接入点）Mac, cliMac, aNonce（随机数）, and sNonce（随机数）
concatenated
#            like mac1||mac2||Nonce（随机数）1||Nonce（随机数）2
#            such that mac1 < mac2 and Nonce（随机数）1 < Nonce（随机数）2
#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（随机数）:        The aNonce（随机数） from the 4-way hanDS（分布式系统）hake
#sNonce（随机数）:        The sNonce（随机数） from the 4-way hanDS（分布式系统）hake
#AP（无线接入点）Mac:          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（随机数）, sNonce（随机数）, AP（无线接入点）Mac, cliMac):
    A = b"Pairwise key expansion"
    B = min(AP（无线接入点）Mac, cliMac) + max(AP（无线接入点）Mac, cliMac) + min(aNonce（随
机数）, sNonce（随机数）) + max(aNonce（随机数）, sNonce（随机数）)
    return (A, B)
ptk = PRF(pmk, A, B)
```

4-WAY HANDS（分布式系统）HAKE: MESSAGE 2

MESSAGE 2: STA（站



第二条消息后

AP（无线接入点）拥有的信息：

PMK　　ANonce（随机数）　SNonce（随机数）　PA

AP（无线接入点）采用与STA（站点）相同的算法生成PTK:

PTK<—PRF-X(PMK, "Pairwise key expansion", Min(AA,SPA) || Max(AA,SPA) || Min(ANonce（随机数）,SNonce（随机数）) || Max(ANonce（随机数）,SNonce（随机数）))

Message Integrity Check (MIC)

MIC: 128-bits。用于验证STA（站点）拥有正确的PTK，从而拥有正确的PMK，也就是验证了STA（站点）的合法性

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)

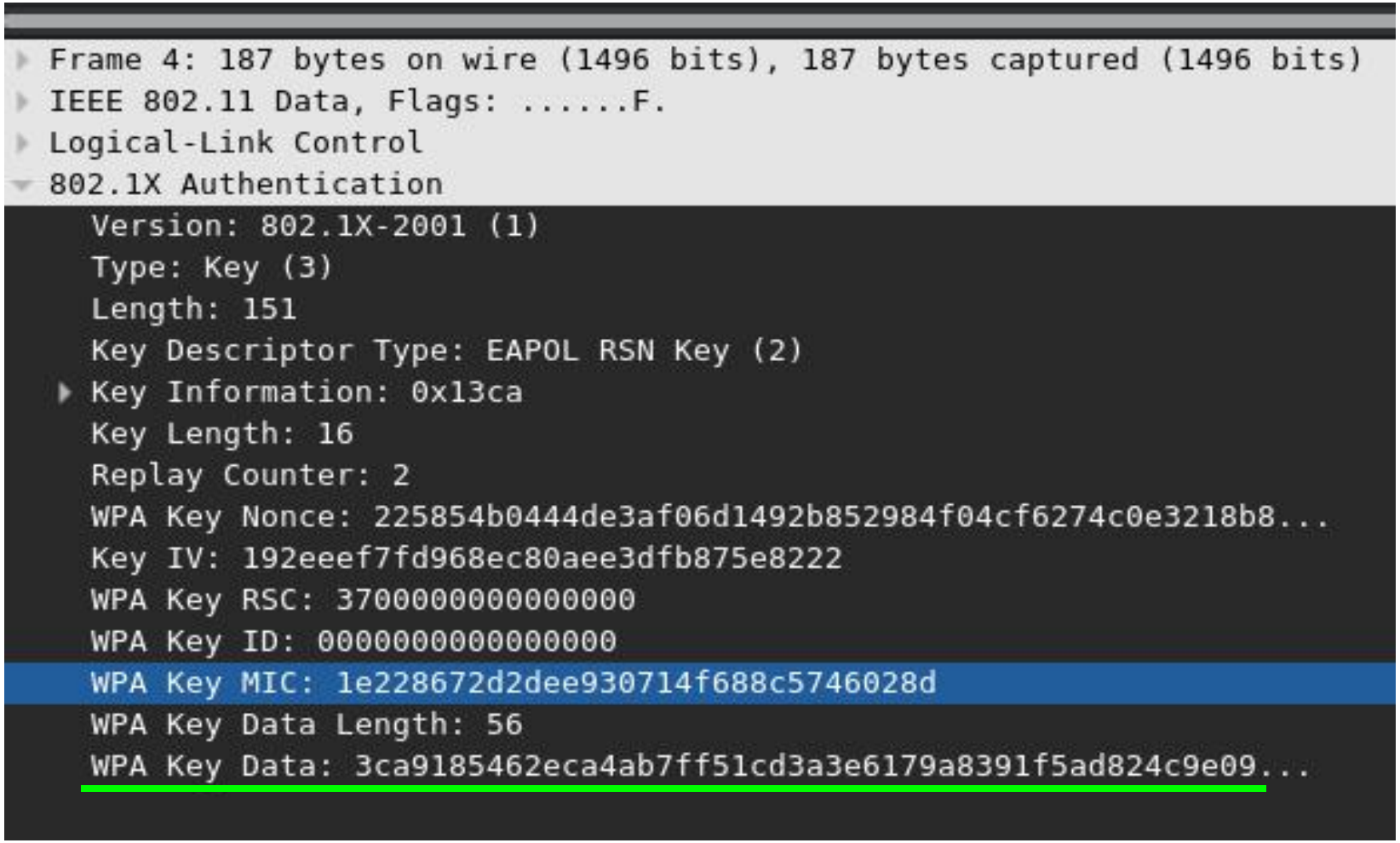
注：为了兼容：wpa2也取前128bits

计算MIC

```
#The entire 802.1x frame with the MIC field set to all zeros
data1 =
a2b_hex("0103007502010a00100000000000000159168bc3a5df18d71efb6423f340088dab9e1
ba2bbc58659e07b3764b0de85700000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000163014010000fac040100000fac0401000
00fac020100")
#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 shoud be mic1 = "d5355382b8a9b806dcaf99cdaf564eb6"
```

4-WAY HANDS（分布式系统）HAKE: MESSAGE 3

MESSAGE 3: AP（无线接入点）→STA（站点）



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（无线接入点）的真实性（不是流氓AP（无线接入点）），一旦STA（站点）验证了MIC，则STA（站点）能够确定该AP（无线接入点）拥有正确的PTK，从而拥有正确的PMK，而只有正确的AP（无线接入点）才能知道PMK，因此验证了AP（无线接入点）的真实性

WPA Key Data：包含了加密的GTK

4-WAY HANDSHAKE: MESSAGE 4

MESSAGE 4: STA（站点）→AP（无线接入点）

用于确认STA（站点）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：

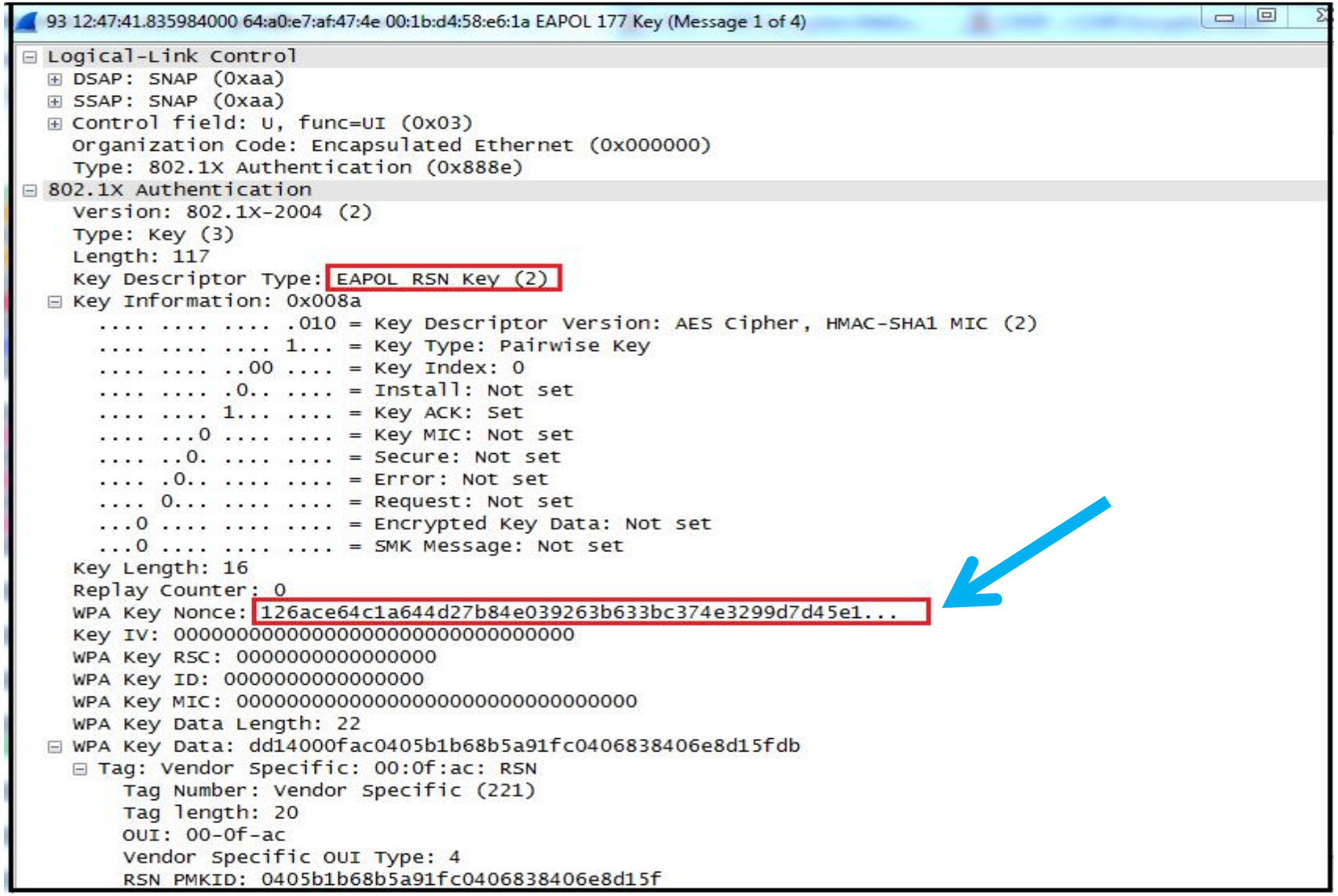
ANonce（随机数）是以明文方式传送的，如果ANonce（随机数）被攻击者篡改了，会有什么后果？密钥建立过程会成功吗？

■如果ANonce（随机数）被修改了，会导致STA（站点）和AP（无线接入点）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（随机数），但是握手过程不会成功。

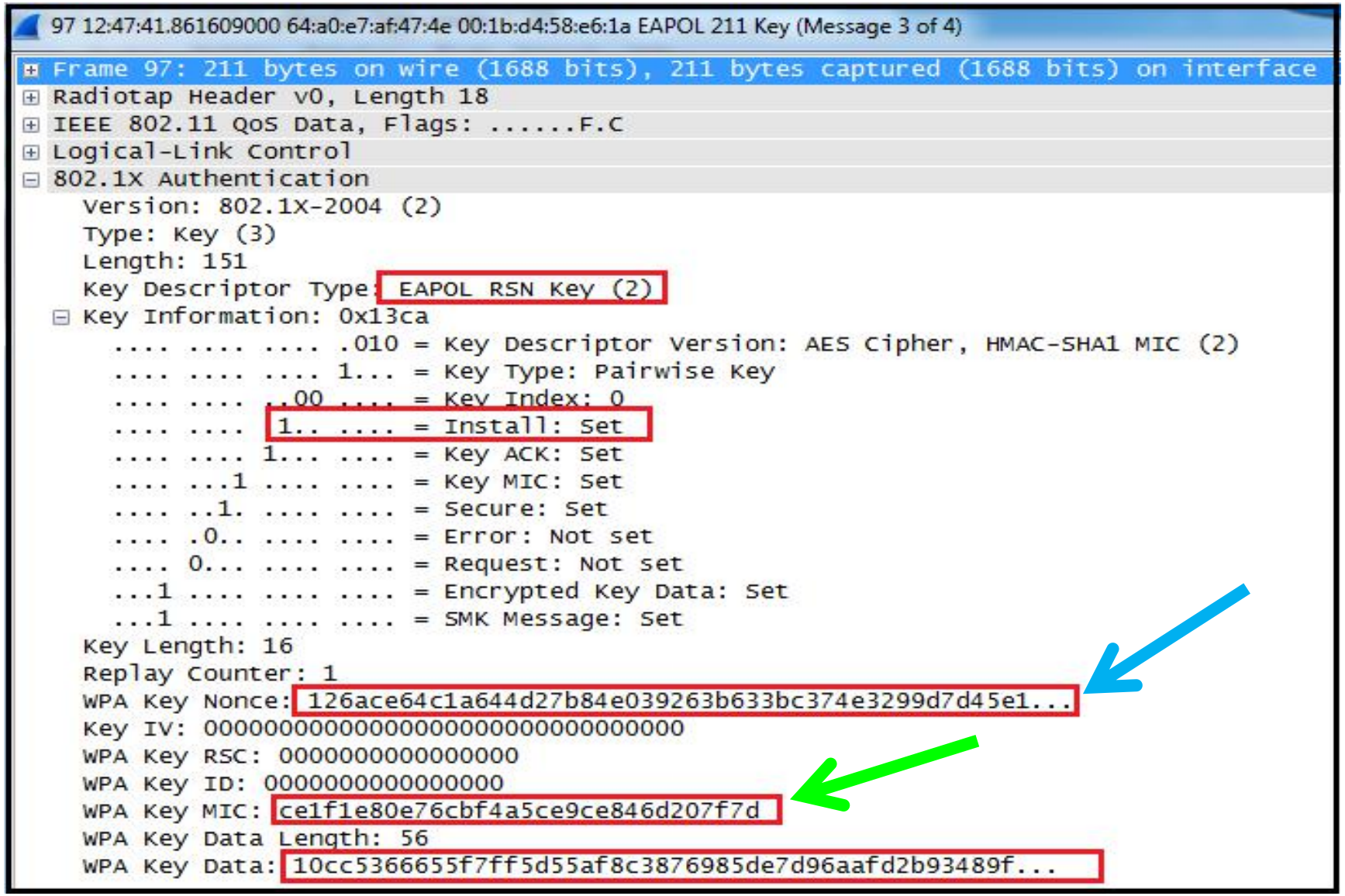
问题讨论1：

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

Message 1: WPA Key Nonce（随机数）

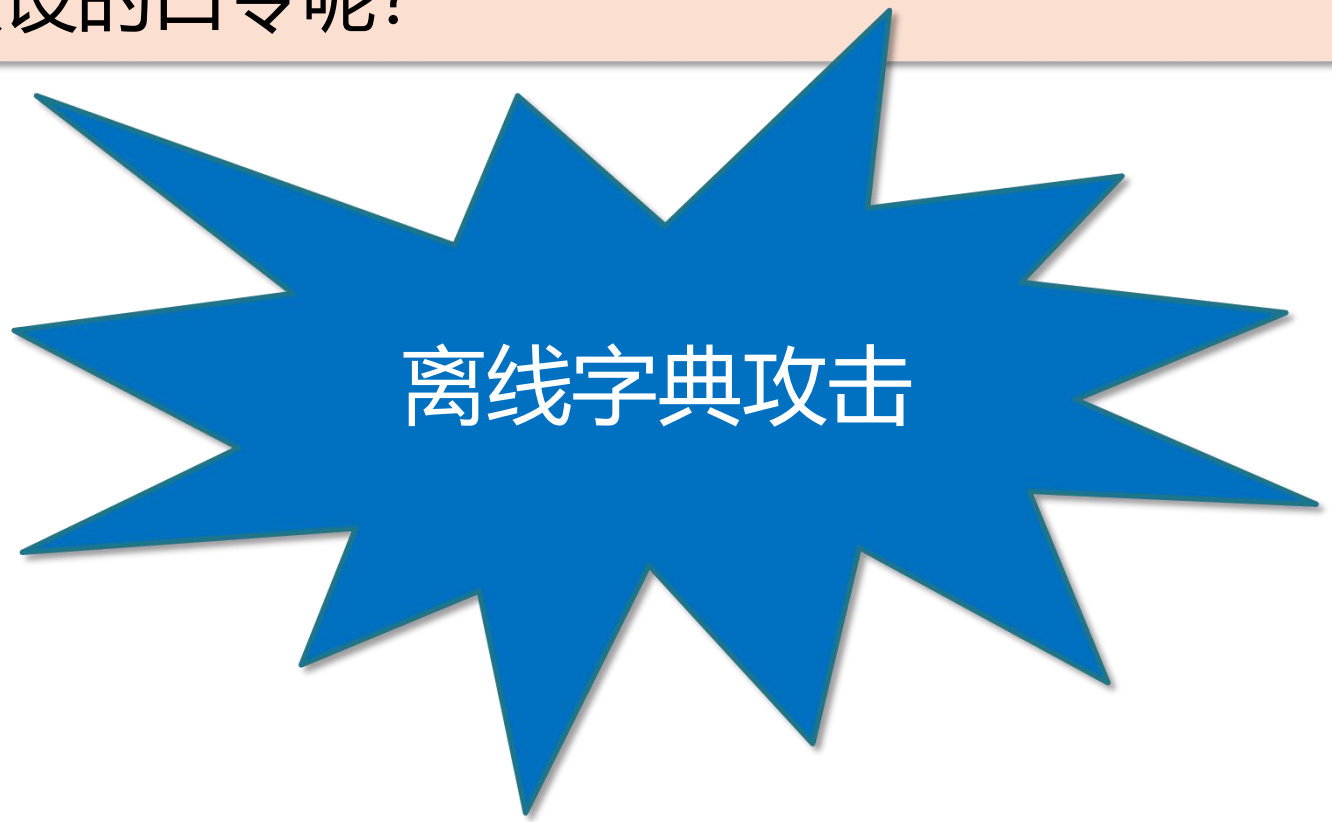


Message 3: WPA Key Nonce（随机数）



问题讨论2：

有没有可能破解AP（无线接入点）和STA（站点）上预设的口令呢?



WPA2

(Wi-Fi Protected Access II)

CCMP(计数器模式密码块链消息完整码协议)

CCM
——COUNTER MODE WITH CBC-MAC

RFC3610

CCM: authentication

首先需要构造以下分组模式:

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:

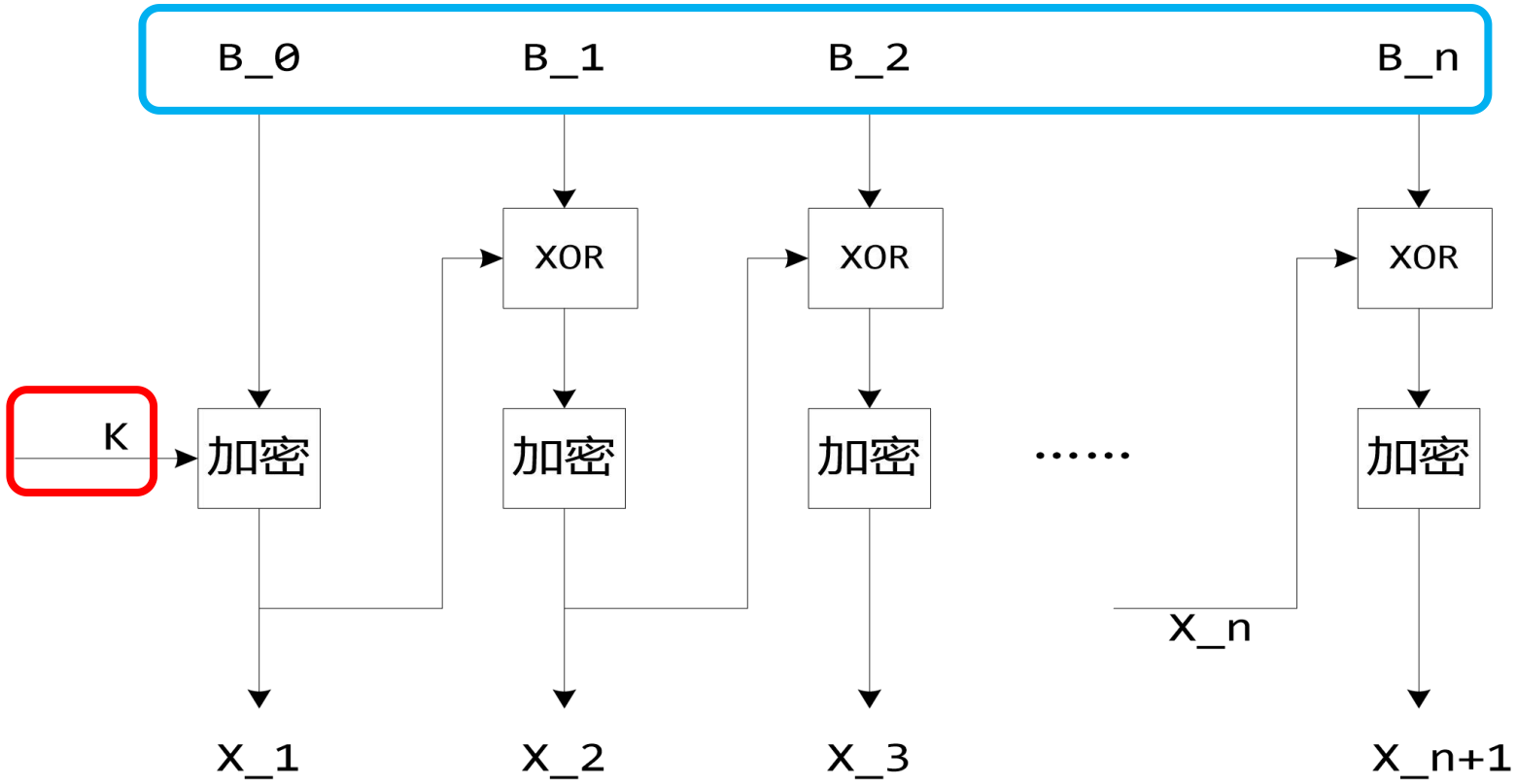
X_1 := E(K, B_0)

X_i+1 := E(K, X_i XOR B_i) for i=1,...,n

T := 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, ……, B_n
密钥: k

CCM encryption: 密钥流生成

加密过程是采用CTR模式，首先构造A_i，然后从A_i生成密钥流分组为：

S_i ：= E(K,A_i)，其中i为0，1，2，……

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

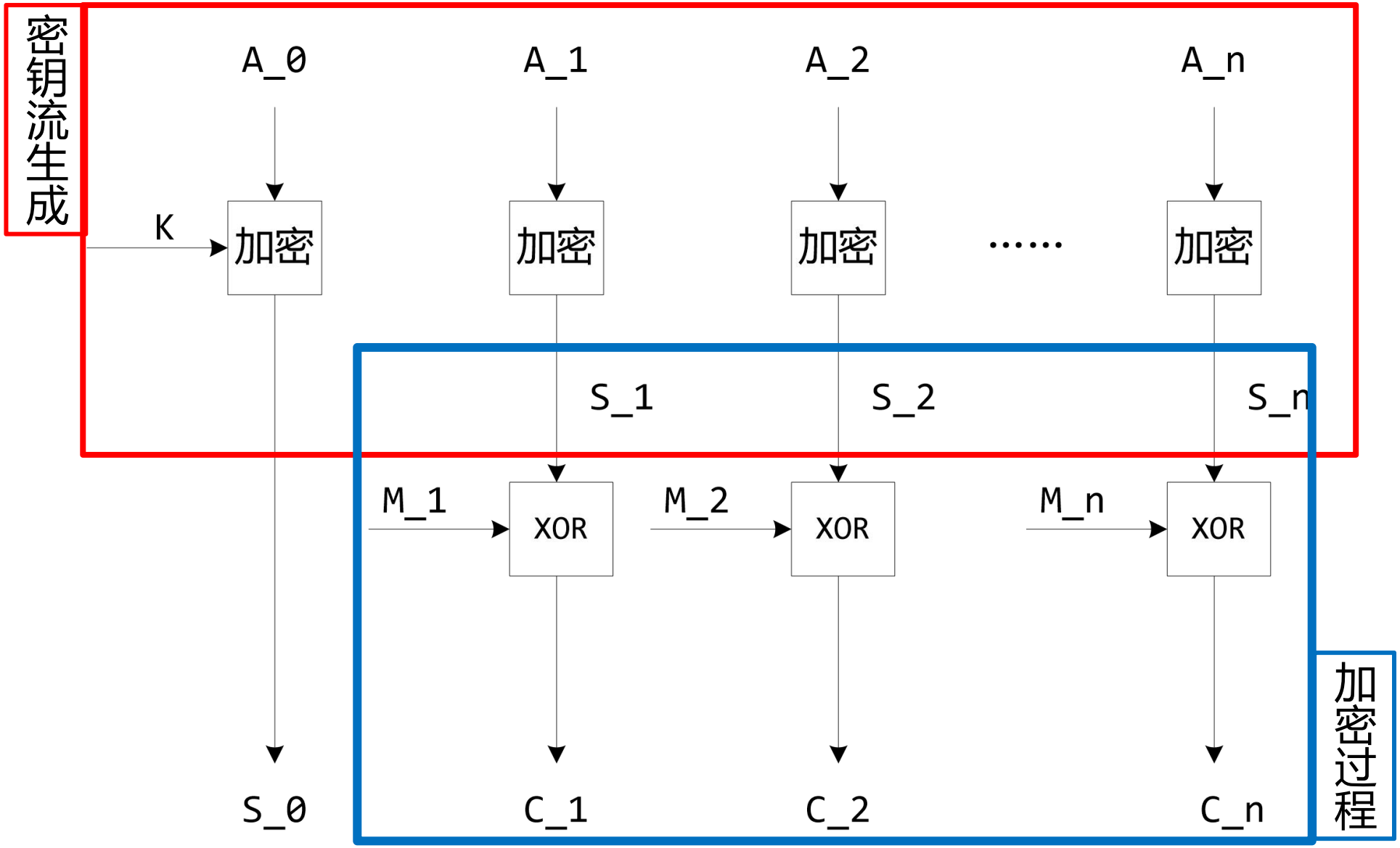
密钥: S_1 | S_2 | S_3 | … | S_n

XOR

明文: M_1 | M_2 | M_3 | … | M_n

密文: C_1 | C_2 | C_3 | … | 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 XOR first-M-bytes(S_0)

即对T用S_0的前M个字节进行XOR运算。

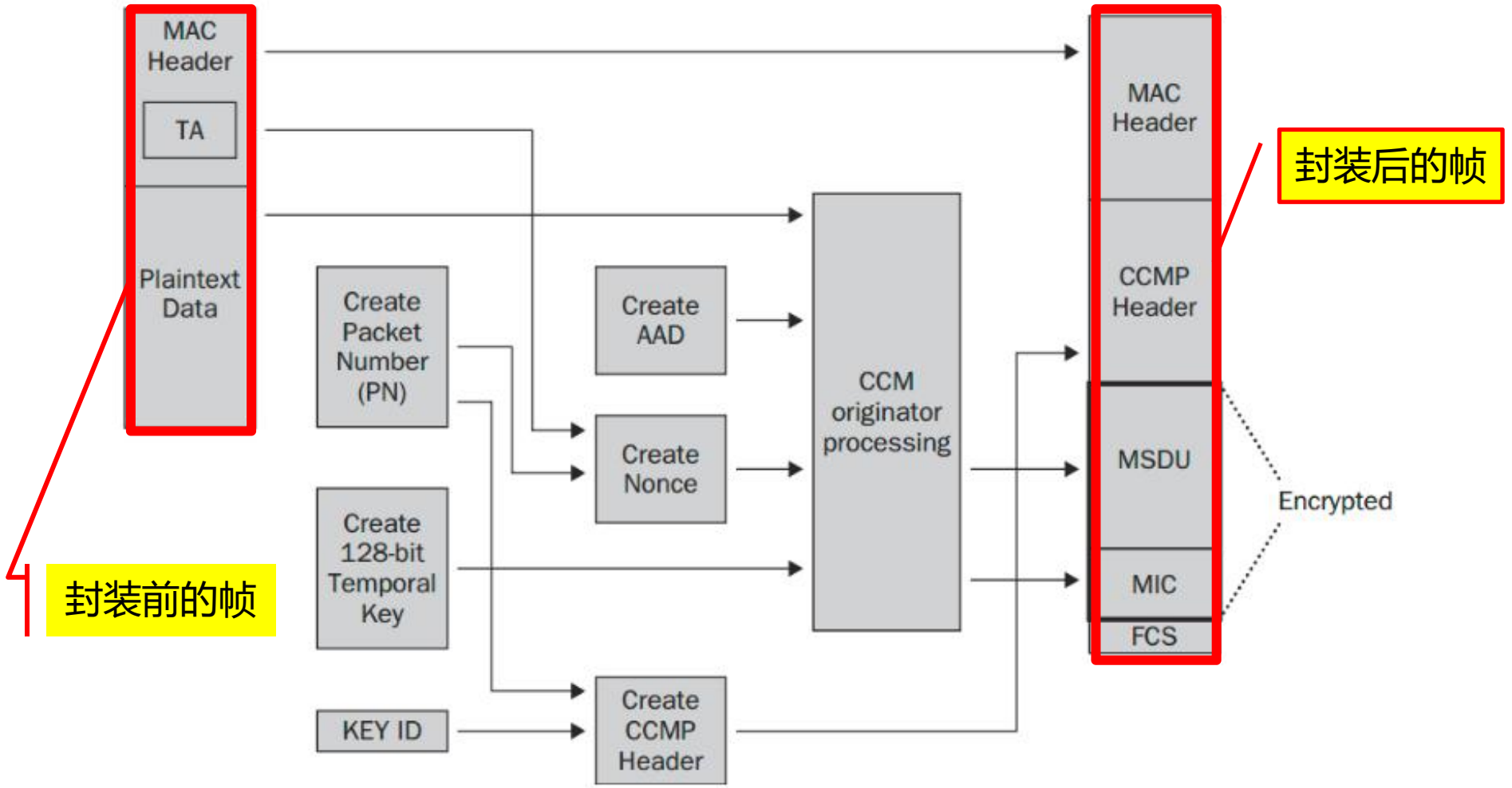
T是CBC-MAC的输出。

最后的输出C：加密消息和紧随其后的加密的认证值U

CCMP(计数器模式密码块链消息完整码协议)

- 基于CCM
- AES，128bit
- **C**ounter mode with **C**BC-**M**AC
  - Authentication-then-Encryption
  - CBC-MAC
  - CTR-AES

CCMP(计数器模式密码块链消息完整码协议)处理过程



体现为如何重新封装帧，以获得安全保护！

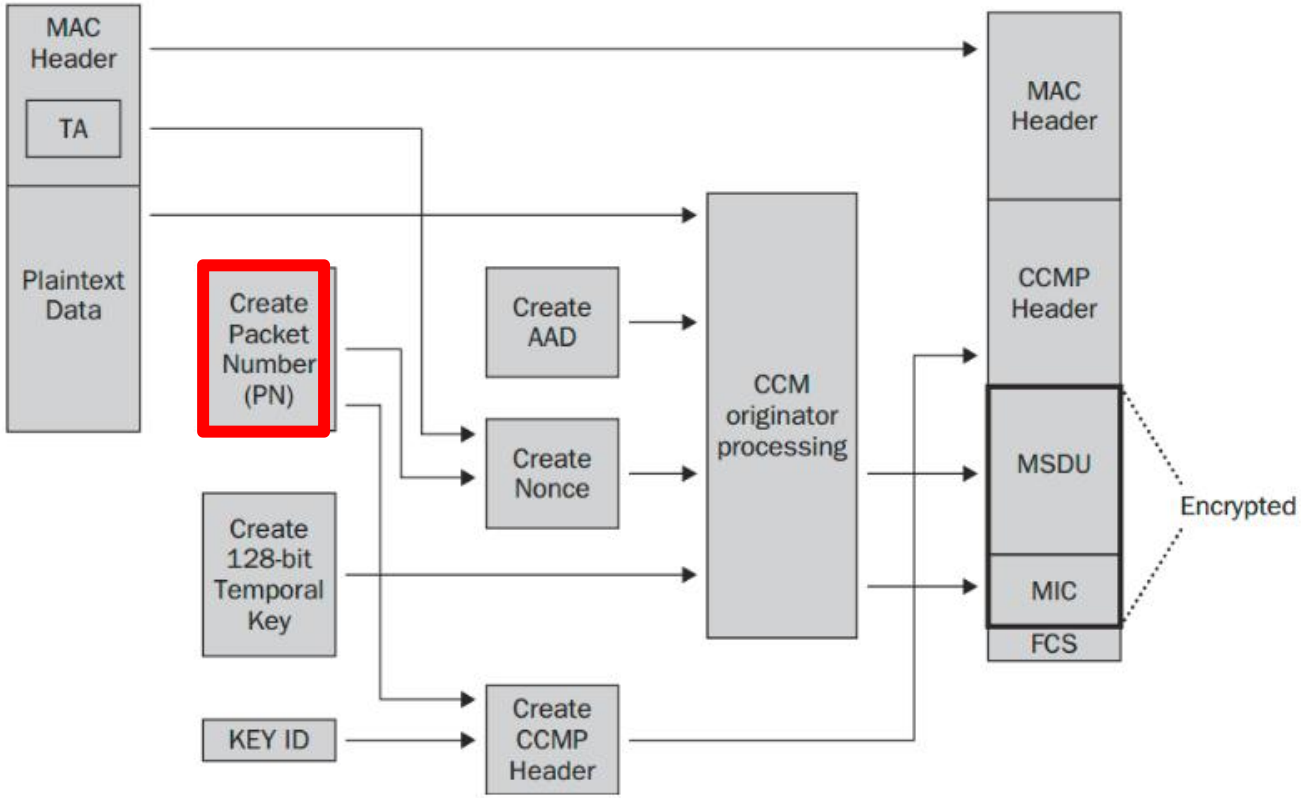
CCMP(计数器模式密码块链消息完整码协议)的输入（1）

- MAC header：802.11 MAC 头部
- plaintext Data(MSDU(MAC服务数据单元))：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(计数器模式密码块链消息完整码协议)也有一个128bit的TK。

CCMP(计数器模式密码块链消息完整码协议)的输入（2）

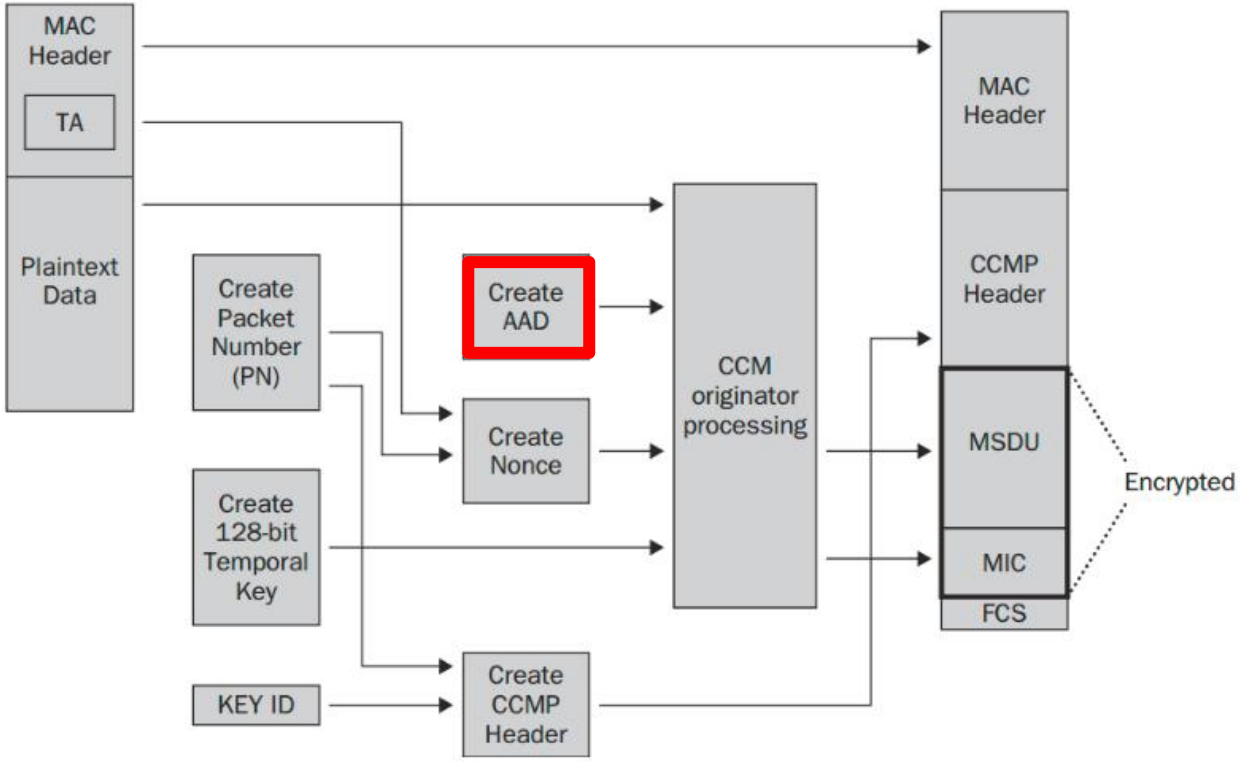
- Key ID: 和TKIP中的一样，用于指定加密用的key，这个ID是index的缩写。
- Nonce（随机数）：长104bit，是由PN (packet number，48bit), Qos中的优先级字段（8bit）和TA(Transmitter Address，48bit)这三个字段组合来。
- AAD（附加认证数据）（Additional Authentication Data）：由MPUD头部的部分字段构建而来，用于确保MAC头部的数据完整性，接收端会使用这个字段来校验MAC头部。

CCMP(计数器模式密码块链消息完整码协议)：packet number



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

CCMP(计数器模式密码块链消息完整码协议)：构建AAD(附加认证数据)



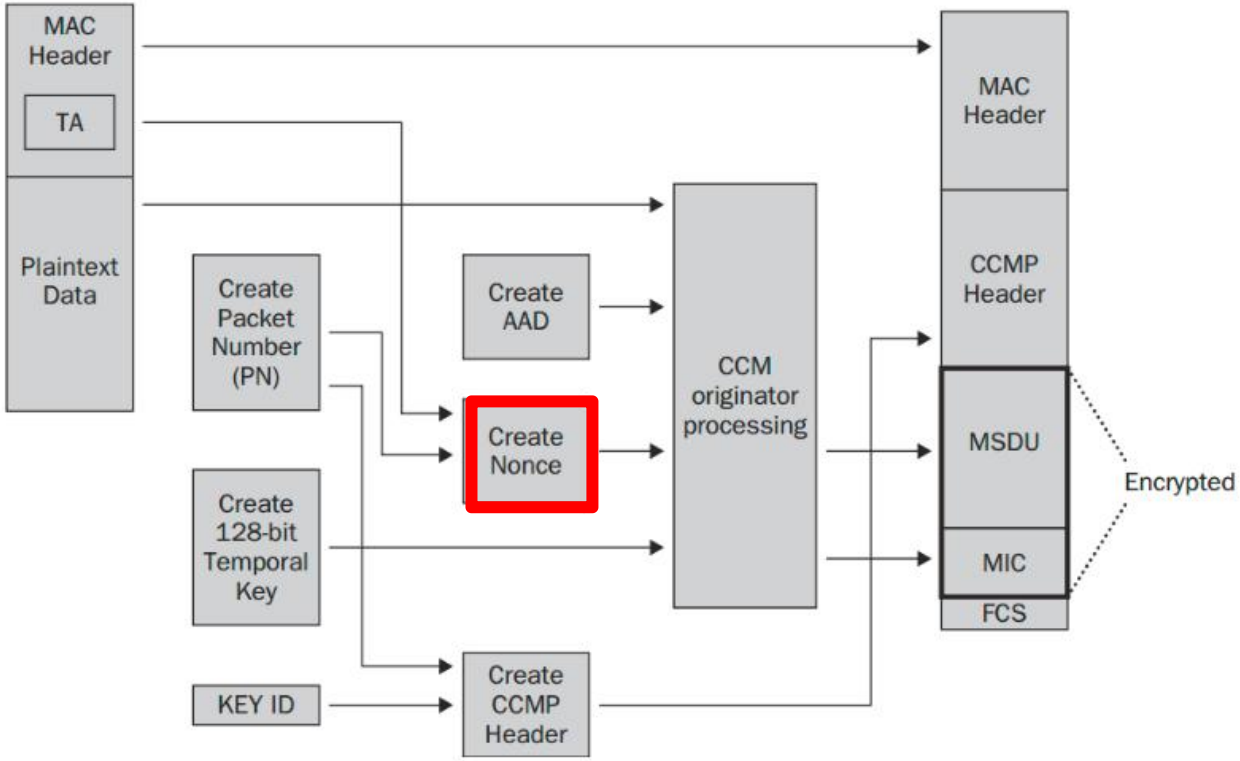
2、使用MPDU(MAC协议数据单元)的头部构建AAD（附加认证数据）（Additional

CCMP(计数器模式密码块链消息完整码协议): AAD（附加认证数据）构建

AAD（附加认证数据）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

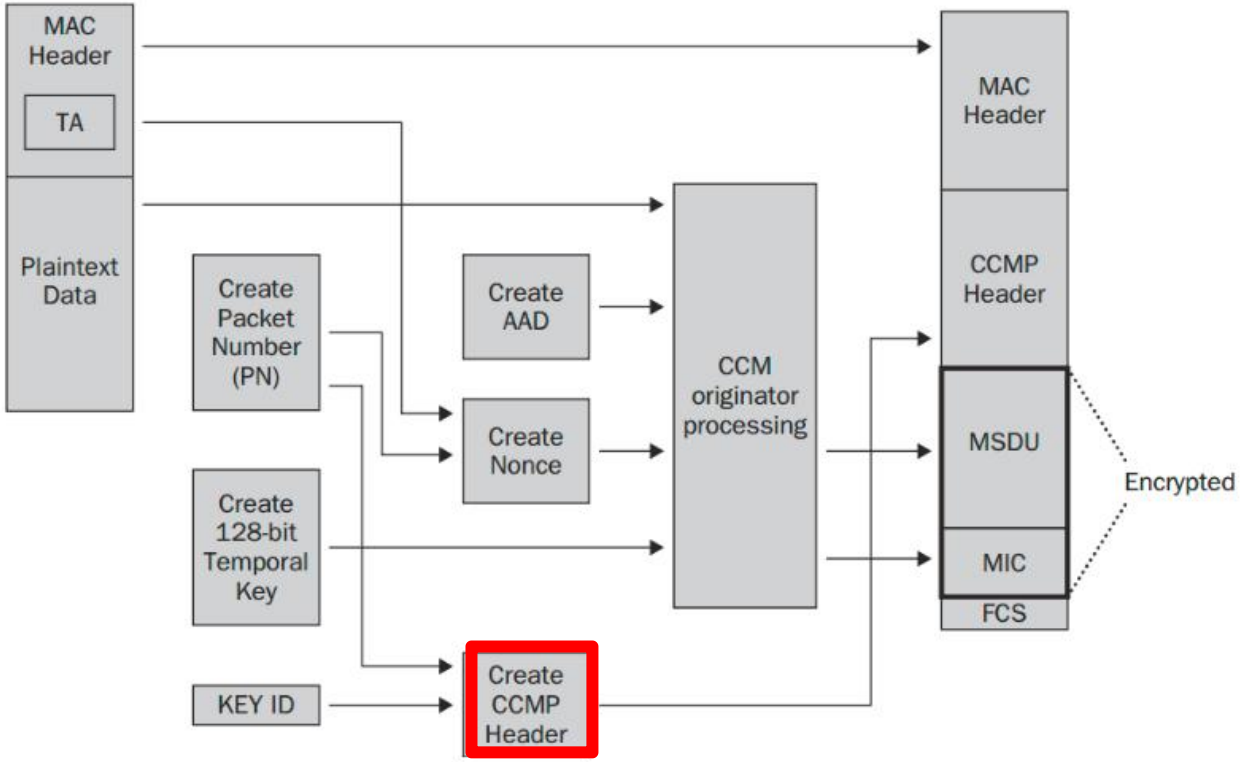
因为AAD（附加认证数据）作为计算MIC值的输入，因此确保了MAC Header部分字段的完整性。

CCMP(计数器模式密码块链消息完整码协议)：构建Nonce（随机数）



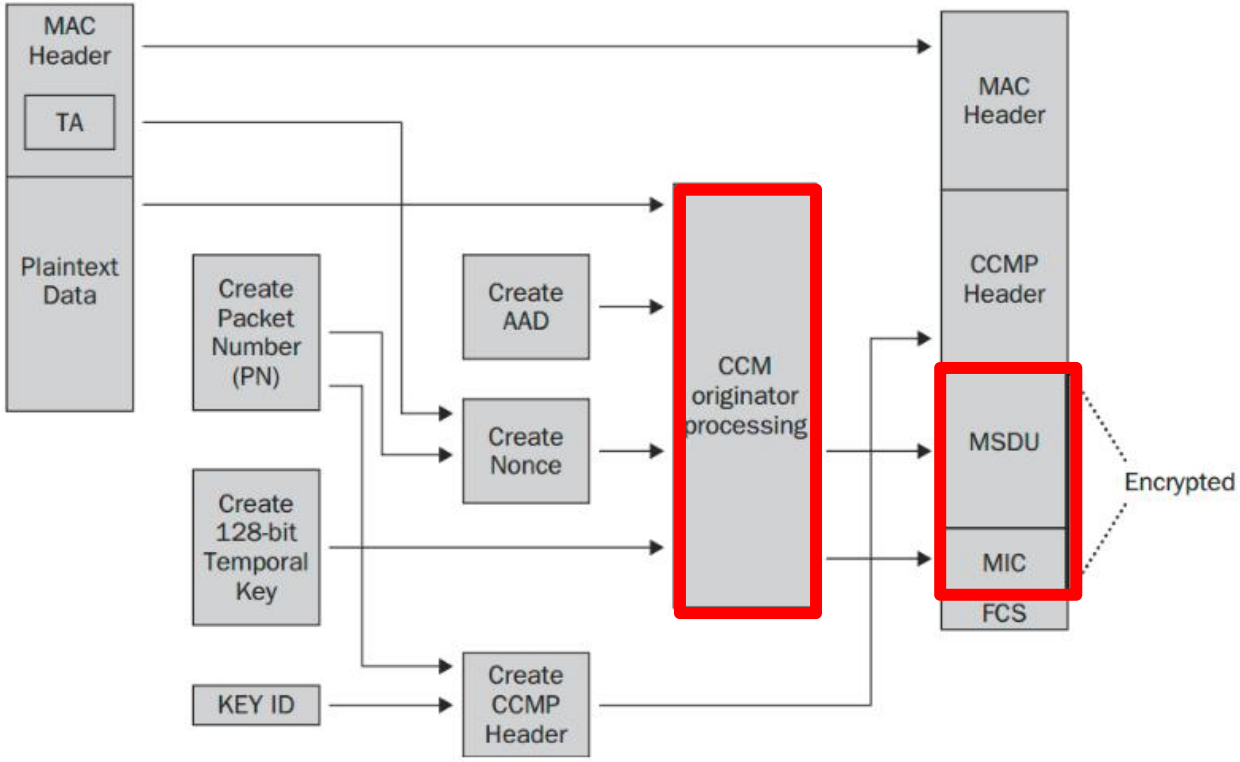
3、由PN(packet number，48bit), Qos中的优先级字段（8bit）和TA(transmitter address , 48bit, Address 2)这三个字段组合生成一个Nonce（随机数）。

CCMP(计数器模式密码块链消息完整码协议)：构建CCMP(计数器模式密码块链消息完整码协议)头部



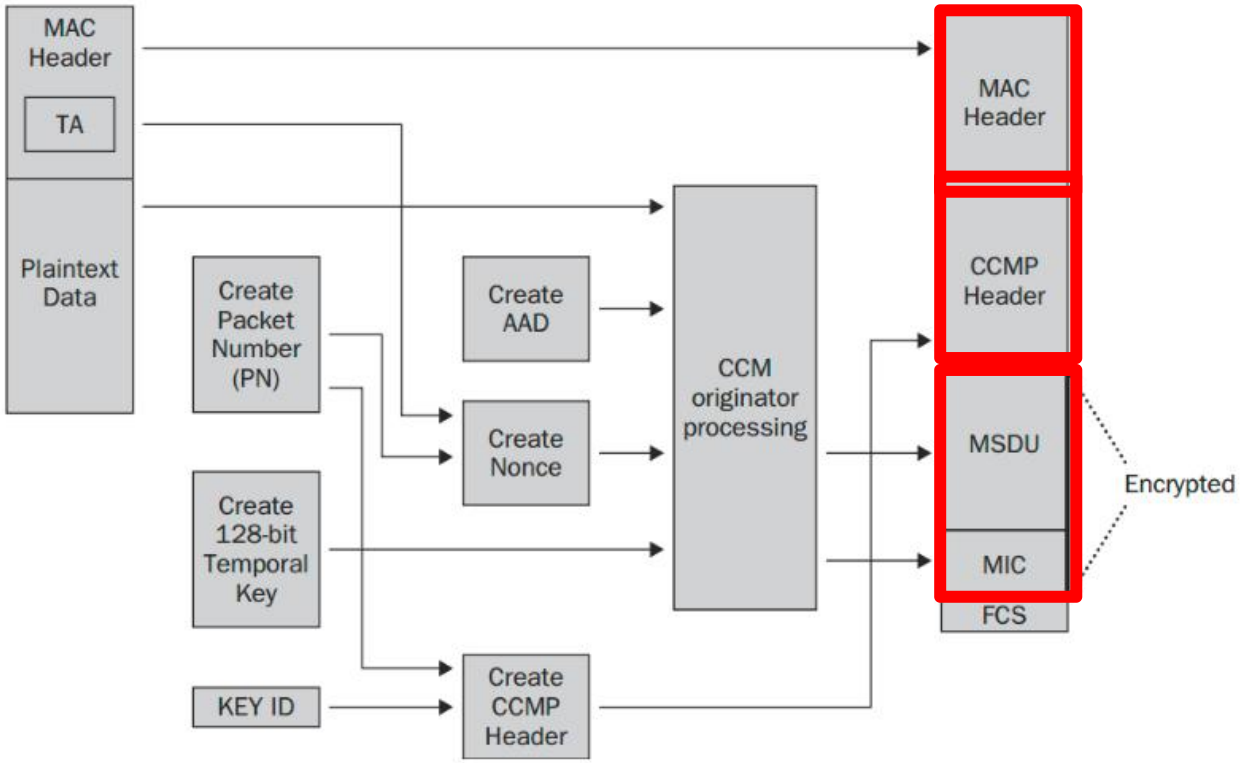
4、构建8-octet CCMP(计数器模式密码块链消息完整码协议) 头部，这个头部由Key ID 和PN构成，PN又被分成6个字段

CCMP(计数器模式密码块链消息完整码协议)：CCM处理



5、使用Temporal Key, AAD（附加认证数据），Nonce（随机数），和MPDU(MAC协议数据单元) data 作为CCM算法输入，生成8个字节的MIC和加密的MSDU(MAC服务数据单元)。

CCMP(计数器模式密码块链消息完整码协议)：封装安全处理过后的帧



6、 在MAC Header后面追加CCMP(计数器模式密码块链消息完整码协议) Header，然后是加密的MSDU(MAC服务数据单元)和MIC，接下来的是FCS，构成安全处理过后的帧。

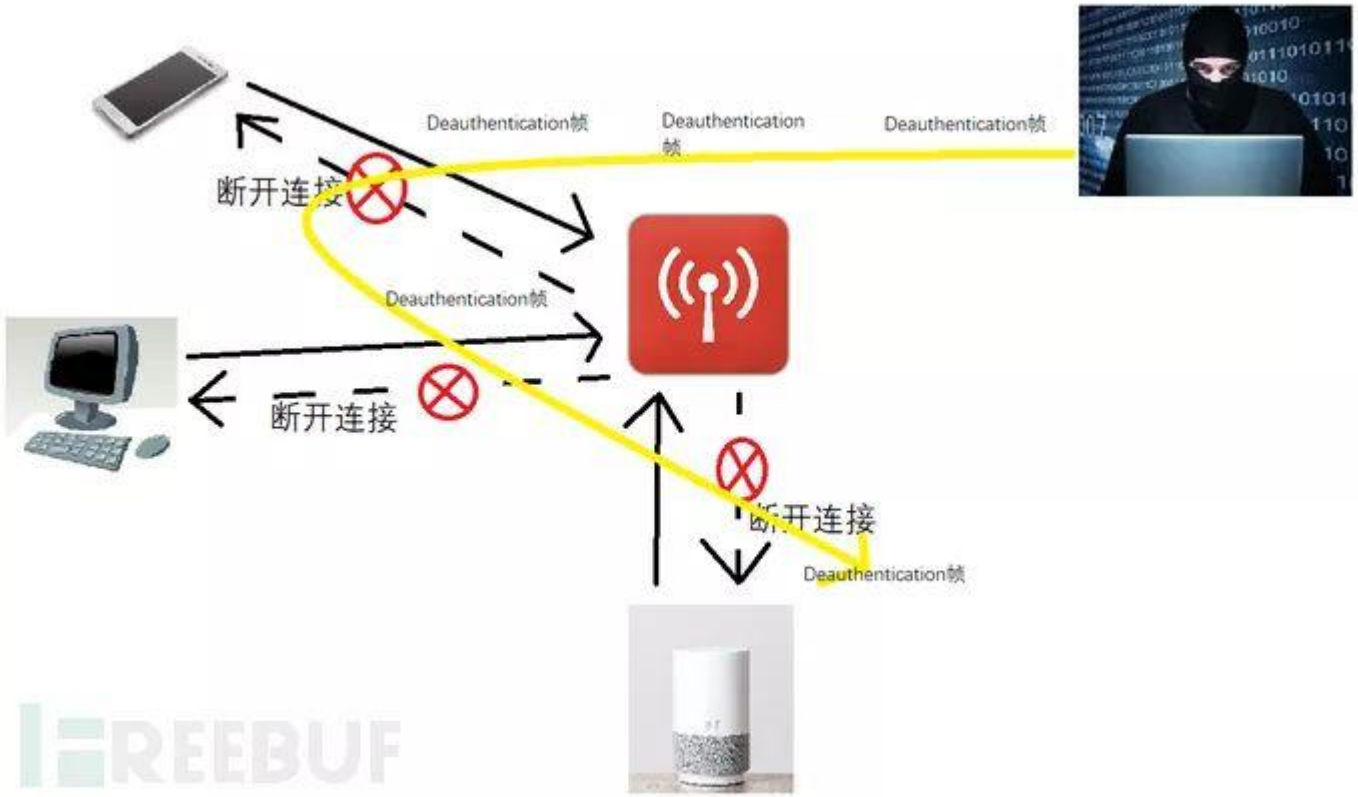
CCMP(计数器模式密码块链消息完整码协议)总结

- CCMP(计数器模式密码块链消息完整码协议)针对MSDU(MAC服务数据单元)进行安全处理，对MSDU(MAC服务数据单元)提供保密性，同时对MSDU(MAC服务数据单元)和MAC Header的部分字段做完整性保护。

无线网络攻击

攻击1：DeAuth

- 伪装成目标AP（无线接入点）已关联的设备，向AP（无线接入点）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攻击的用途

- Evil twin access point
  - 强制STA（站点）连接到一个假冒的AP（无线接入点）
- Password attack
  - 抓取4-way hanDS（分布式系统）hake帧

Deauth实施工具

命令:

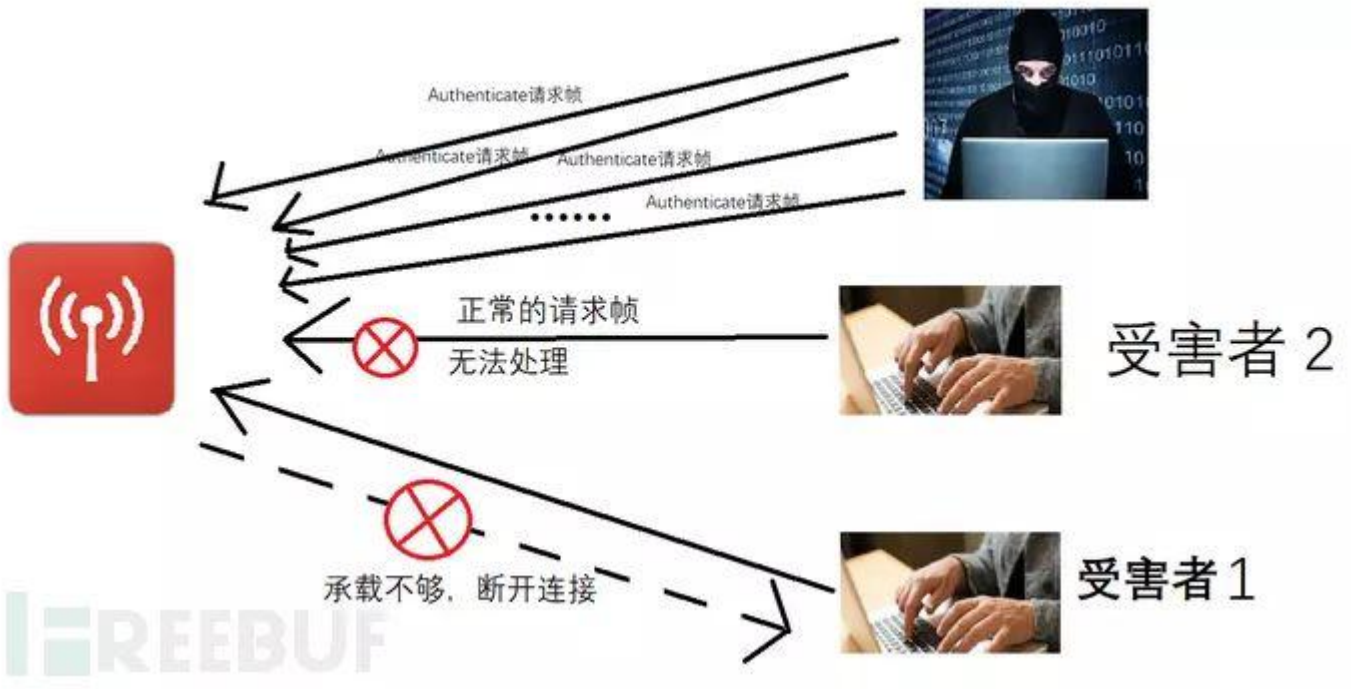
```
aireplay-ng -0 1 -a xx:xx:xx:xx:xx:xx -c yy:yy:yy:yy:yy:yy
```

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 (无线接入点) (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 (无线接入点)
- wlan0 is the NIC (Network Interface Card)

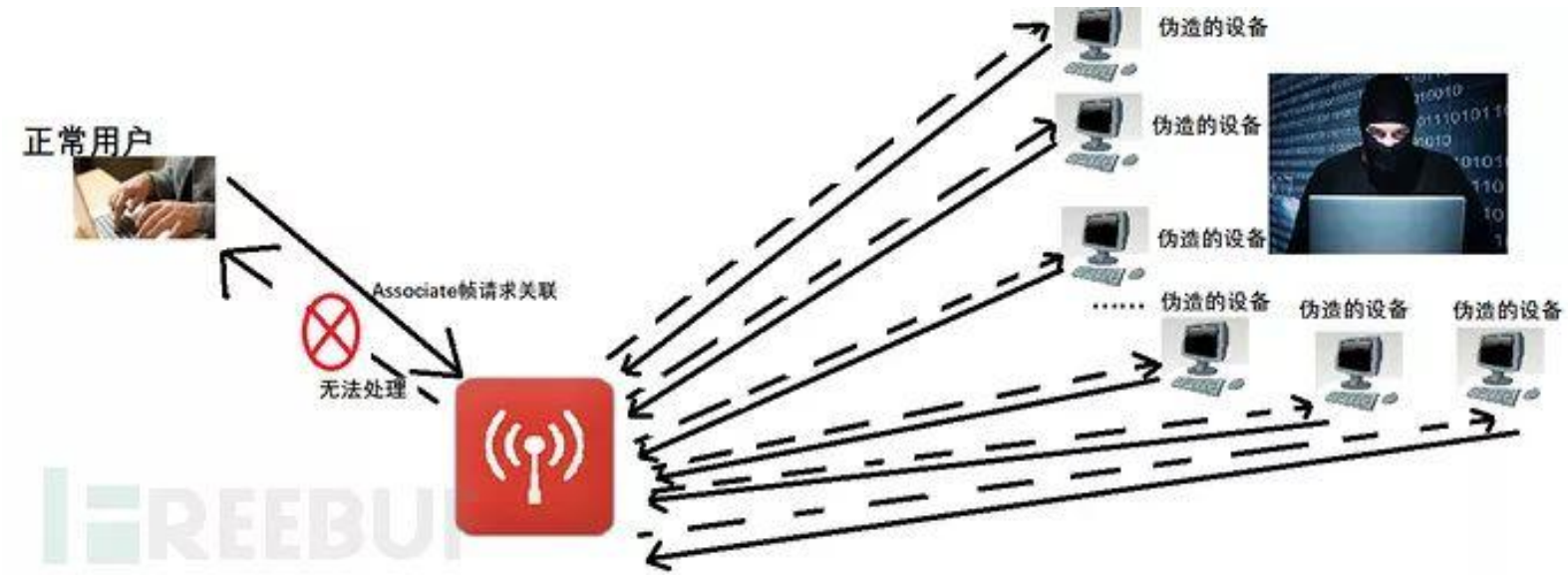
攻击手段2：AuthDos

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



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（无线接入点）的关联表，使正常用户无法与AP（无线接入点）建立关联



攻击4：RF jammingAttack

- 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

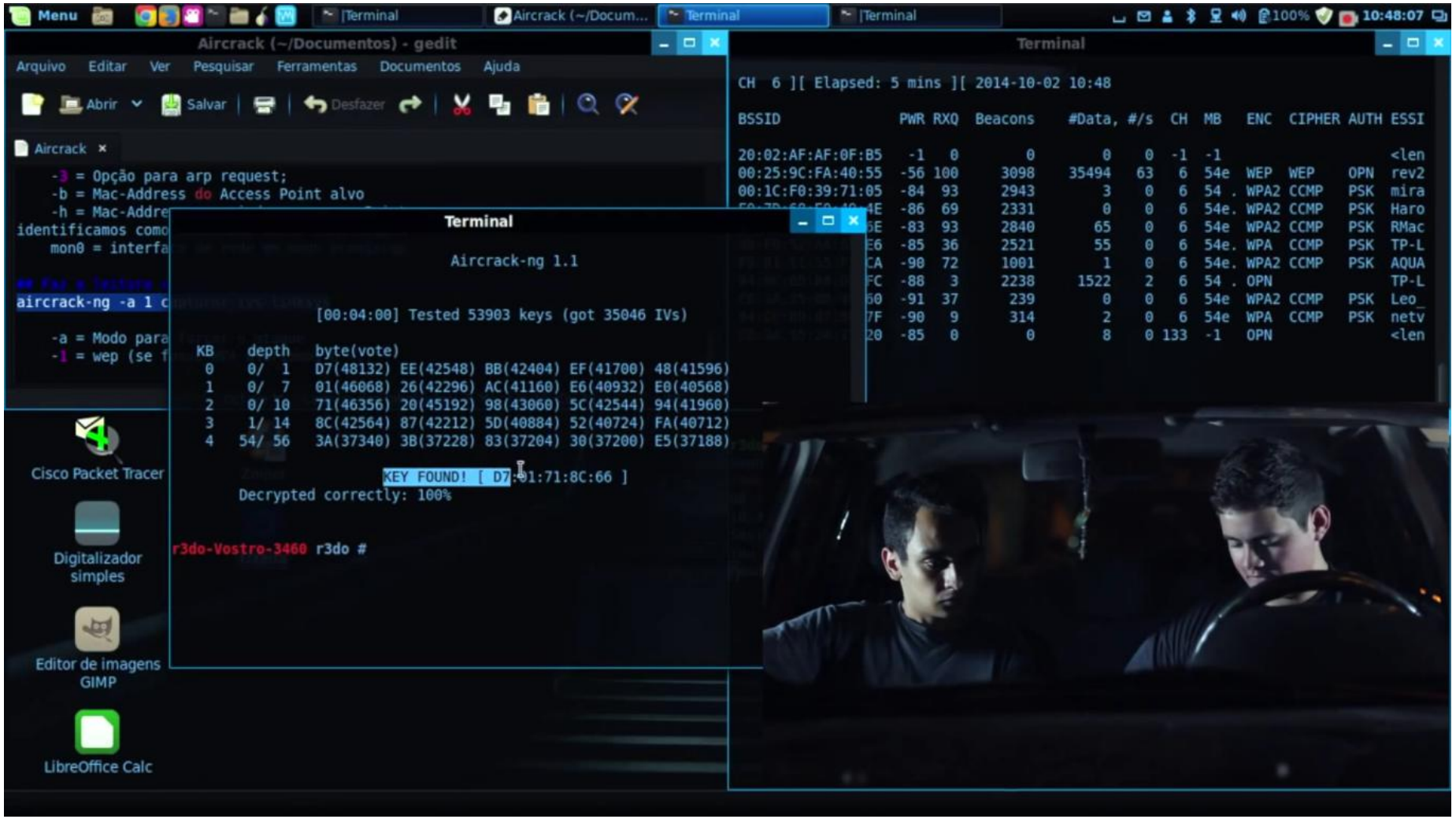
无线入侵检测工具

- WAIDPS
  - Wireless Auditing, Intrusion Detection & Prevention System

更多信息:

https://github.com/SYWorks/waidps

Movies


Menu
Terminal
Aircrack (~/Documentos) - gedit
Arquivo Editar Ver Pesquisar Ferramentas Documentos Ajuda
Abrir Salvar Desfazer
Aircrack ×
-3 = Opção para arp request;
-b = Mac-Address do Access Point alvo
-h = Mac-Addre
identificamos como
mon0 = interfa
aircrack-ng -a 1 c
-a = Modo para
-1 = wep (se f
Terminal
Aircrack-ng 1.1
[00:04:00] Tested 53903 keys (got 35046 IVs)
KB depth byte(vote)
0 0/ 1 D7(48132) EE(42548) BB(42404) EF(41700) 48(41596)
1 0/ 7 01(46068) 26(42296) AC(41160) E6(40932) E0(40568)
2 0/ 10 71(46356) 20(45192) 98(43060) 5C(42544) 94(41960)
3 1/ 14 8C(42564) 87(42212) 5D(40884) 52(40724) FA(40712)
4 54/ 56 3A(37340) 3B(37228) 83(37204) 30(37200) E5(37188)
KEY FOUND! [D7:01:71:8C:66]
Decrypted correctly: 100%
r3do-Vostro-3460 r3do #
Terminal
CH 6][Elapsed: 5 mins][2014-10-02 10:48
BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSI
20:02:AF:AF:0F:B5 -1 0 0 0 0 -1 -1 <len
00:25:9C:FA:40:55 -56 100 3098 35494 63 6 54e WEP WEP OPN rev2
00:1C:F0:39:71:05 -84 93 2943 3 0 6 54 . WPA2 CCMP PSK mira
-86 69 2331 0 0 6 54e. WPA2 CCMP PSK Haro
-83 93 2840 65 0 6 54e WPA2 CCMP PSK RMac
-85 36 2521 55 0 6 54e. WPA CCMP PSK TP-L
-90 72 1001 1 0 6 54e. WPA2 CCMP PSK AQUA
-88 3 2238 1522 2 6 54 . OPN TP-L
-91 37 239 0 0 6 54e WPA2 CCMP PSK Leo_
-90 9 314 2 0 6 54e WPA CCMP PSK netv
-85 0 0 8 0 133 -1 OPN <len
Cisco Packet Tracer
Digitalizador simples
Editor de imagens GIMP
LibreOffice Calc

更多信息:

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

作业

- 对TKIP和有线等效保密(WEP)协议做安全性比较
- 对CCMP(计数器模式密码块链消息完整码协议)和TKIP做安全性比较