ICS 35.110
L 78

中华人民共和国国家标准

GB 15629.11—2003/XG1—2006

信息技术　系统间远程通信和信息交换　局域网和城域网　特定要求　第11部分：无线局域网媒体访问控制和物理层规范

第1号修改单

Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications

(ISO/IEC 8802-11:1999, MOD)

Amendment 1

2006-01-27 发布

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

信息技术　系统间远程通信和信息交换 局域网和城域网　特定要求　第11部分：无线局域网媒体访问控制和物理层规范

第1号修改单

前　言

本修改单的7.3.2.8和第8章为强制性的，其余为推荐性的。

考虑到不同安全体制的并存，本修改单对国家标准GB 15629.11—2003《信息技术　系统间远程通信和信息交换　局域网和城域网　特定要求　第11部分：无线局域网媒体访问控制和物理层规范》中涉及到无线局域网鉴别与保密基础结构(WAPI)的部分进行了适应性修改。本修改单为GB 15629.11—2003的第1号修改单。

本修改单中采用的WAPI机制已经ISO/IEC授权的相关机构审查获得认可，并分配了用于WAPI的以太类型字段(EtherType Field)0x88B4和TCP/UDP端口号3810。

本修改单修改内容说明如下：

——增补了适应和满足其他安全体制在WAPI框架下的应用的内容；

——增补了与WAPI应用和密码算法相关的接口详细定义；

——增补了相关文本的细化内容，便于厂商更易理解和使用；

——第2章中的规范性引用文件做了增补；

——第3章中的术语和定义增加了新的内容；

——第4章中的缩略语增加了新的内容；

——第5、7、8、10、11章中的WAPI相应内容作了增补，相应的图、表作了重新编号；

——增补了附录A、附录C、附录D中的相应内容；

——添加了附录E。

其中，对GB 15629.11—2003修改时涉及到的有关章条的信息如下：

修改项号	GB 15629.11—2003章条号	修改说明
1	第2章	在第2章中删除原规范性引用文件中的第4～9项，并增加新规范性引用文件
2	第3章	在第3章中增加新项目
3	第4章	在第4章中增加新项目
4	5.4.3.1	在5.4.3.1内容之后增加新内容
5	5.4.3.1.1	删除该条
6	5.4.3.3	在5.4.3.3内容之后增加新内容
7	5.4.3.4	对5.4.3.4的内容进行替换
8	5.5	对5.5的内容进行替换

修改项号	GB 15629.11—2003 章条号	修改说明
9	5.7.5	对 5.7.5 的内容进行替换
10	5.7.6	对 5.7.6 的内容进行替换
11	5.8	对 5.8 的图 11 进行替换
12	第 5 章	在 5.8 内容之后增加 5.9
13	6.1	在 6.1.3 内容之后增加 6.1.4
14	7.1.3.1	对 7.1.3.1 中 7.1.3.1.1 之前的内容进行替换
15	7.1.3.1.9	对 7.1.3.1.9 进行替换
16	7.2.3.1	在 7.2.3.1 的表 5 中增加新项目
17	7.2.3.4	在 7.2.3.4 的表 7 中增加新项目
18	7.2.3.6	在 7.2.3.6 的表 9 中增加新项目
19	7.2.3.6	在 7.2.3.6 的表 9 中增加新项目
20	7.2.3.9	在 7.2.3.9 的表 12 中增加新项目
21	7.3.1.4	对 7.3.1.4 的内容进行替换
22	7.3.1.7	在 7.3.1.7 的表 17 中增加新项目
23	7.3.1.9	在 7.3.1.9 的表 18 中增加新项目
24	7.3.2	在 7.3.2 的表 19 中增加新项目
25	7.3.2	在 7.3.2.7 内容之后增加新内容
26	第 8 章	对第 8 章的内容进行替换
27	10.3.4	对 10.3.4 的内容进行替换
28	10.3.5	对 10.3.5 的内容进行替换
29	10.3.6	对 10.3.6 的内容进行替换
30	10.3.7	对 10.3.7 的内容进行替换
31	10.3.10	在 10.3.10 之后增加 10.3.11、10.3.12、10.3.13、10.3.14、10.3.15
32	11.3	对 11.3 的内容进行替换
33	11.4	对 11.4 的内容进行替换
34	A.4.4.1	对 A.4.4.1 的 PC1、PC2 与 PC3 项目进行替换
35	A.4.4.1	删除 A.4.4.1 的 PC3
36	第 C.2 章	在第 C.2 章中的第一张图前增加新文本
37	第 C.2 章	对第 C.2 章中图“3102_D\LmeEnum(31)”进行替换
38	第 C.2 章	对第 C.2 章中图“3104_d\StationData(31)”进行替换
39	第 C.2 章	对第 C.2 章中图“3115_d\Defragment(31)”进行替换
40	第 C.2 章	对第 C.2 章中图“3117_d\Frame_1(31)” 进行替换
41	第 C.2 章	对第 C.2 章中图“3119_d\Frame_3(31)” 进行替换

修改项号	GB 15629.11—2003 章条号	修改说明
42	第 C.2 章	对第 C.2 章中图“3120_d\Frame_4(31)”进行替换
43	第 C.2 章	对第 C.2 章中图“GetElem_1a(1)”进行替换
44	第 C.2 章	对第 C.2 章中图“3121_d\FrameType(31)”进行替换
45	第 C.2 章	对第 C.2 章中图“3201_d\StationConfig(5)”进行替换
46	第 C.2 章	对第 C.2 章中图“3202_d\PrivOperation(5)”进行替换
47	第 C.3 章	在第 C.3 章中的第一张图前增加新文本
48	第 C.3 章	对第 C.3 章中图“Station_1b(3)”进行替换
49	第 C.3 章	对第 C.3 章中图“Sta_signals_2d(3)”进行替换
50	第 C.3 章	对第 C.3 章中图“Sta_signallists_3c(3)”进行替换
51	第 C.3 章	对第 C.3 章中图“Sta_Mpdu_ gen_1a(1)”进行替换
52	第 C.3 章	对第 C.3 章中图“prepare_1b(2)”进行替换
53	第 C.3 章	对第 C.3 章中图“Fragment_2b(2)”进行替换
54	第 C.3 章	对第 C.3 章中图“rx_coord_2b(4)”进行替换
55	第 C.3 章	对第 C.3 章中图“rx_coord_3b(4)”进行替换
56	第 C.3 章	对第 C.3 章中图“rxcoord_3.1a(4)”进行替换
57	第 C.3 章	对第 C.3 章中图“Mac_Mgmt_1a(1)”进行替换
58	第 C.3 章	对第 C.3 章中图“Mib_access_1a(2)”进行替换
59	第 C.3 章	对第 C.3 章中图“Mib_import_export_2b(2)”进行替换
60	第 C.3 章	对第 C.3 章中图“Mlme_request_1b(3)”进行替换
61	第 C.3 章	对第 C.3 章中图“Mlme_request_3a(3)”进行替换
62	第 C.3 章	对第 C.3 章中图“MLME_1a(1)”进行替换
63	第 C.3 章	对第 C.3 章中图“ps_monitor(2)”进行替换
64	第 C.3 章	对第 C.3 章中图“ps_monitor_2a(2)”进行替换
65	第 C.3 章	对第 C.3 章中图“sta_Mm_svc_1b (2)”进行替换
66	第 C.3 章	对第 C.3 章中图“sta_Mm_svc_1.1a(2)”进行替换
67	第 C.3 章	对第 C.3 章中图“mmpdu_svc_1.1b(2)”进行替换
68	第 C.3 章	对第 C.3 章中图“Auth_req_1a(2)”进行替换
69	第 C.3 章	对第 C.3 章中图“Deauth_2a(2)”进行替换
70	第 C.3 章	对第 C.3 章中图“Auth_rsp_1b(2)”进行替换
71	第 C.3 章	在第 C.3 章中图“Auth_rsp_1b(2)”之后，插入图“auth_rsp_2b(2)”
72	第 C.3 章	对第 C.3 章中图“receive_1a(1)”进行替换
73	第 C.3 章	对第 C.3 章中图“nav_clear_1b(2)”进行替换
74	第 C.3 章	对第 C.3 章中图“start_rx_1b(2)”进行替换

修改项号	GB 15629.11—2003 章条号	修改说明
75	第 C.3 章	对第 C.3 章中图“Validate_rx_2c(2)”进行替换
76	第 C.3 章	对第 C.3 章中图“pre_filter_1d(4)”进行替换
77	第 C.3 章	对第 C.3 章中图“report_rx_4a(4)”进行替换
78	第 C.3 章	在第 C.3 章中图“report_rx_4a(4)”之后，插入图“filter_1b(3)”
79	第 C.3 章	对第 C.3 章中图“defragment_3c(3)”进行替换
80	第 C.4 章	在第 C.4 章中的第一张图前增加新文本
81	第 C.4 章	对第 C.4 章中图“AP_1b(3)”进行替换
82	第 C.4 章	对第 C.4 章中图“AP_signals_2d(3)”进行替换
83	第 C.4 章	对第 C.4 章中图“AP_signallists_3b(13)”进行替换
84	第 C.4 章	对第 C.4 章中图“ap_MPDU_ gen_1a(1)”进行替换
85	第 C.4 章	对第 C.4 章中图“prepare_1b(2)”进行替换
86	第 C.4 章	对第 C.4 章中图“fragment_2b(2)”进行替换
87	第 C.4 章	对第 C.4 章中图“rx_coord_2b(4)”进行替换
88	第 C.4 章	对第 C.4 章中图“rx_coord_3b(4)”进行替换
89	第 C.4 章	对第 C.4 章中图“rx_coord_3.1a(4)”进行替换
90	第 C.4 章	对第 C.4 章中图“Mac_Mgmt_1a(1)”进行替换
91	第 C.4 章	对第 C.4 章中图“Mlme_indicate_1a(1)”进行替换
92	第 C.4 章	对第 C.4 章中图“Mib_access_1a(2)”进行替换
93	第 C.4 章	对第 C.4 章中图“Mib_import_export_2b(2)”进行替换
94	第 C.4 章	对第 C.4 章中图“Mlme_request_1b(3)”进行替换
95	第 C.4 章	对第 C.4 章中图“Mlme_request_3a(3)”进行替换
96	第 C.4 章	对第 C.4 章中图“ap_MLME_1a(1)”进行替换
97	第 C.4 章	对第 C.4 章中图“ps_monitor_1a(2)”进行替换
98	第 C.4 章	对第 C.4 章中图“ps_monitor_2a(2)”进行替换
99	第 C.4 章	对第 C.4 章中图“ap_Mm_svc_1b(1)”进行替换
100	第 C.4 章	对第 C.4 章中图“ap_auth_req_1a(1)”进行替换
101	第 C.4 章	在第 C.4 章中图“Auth_req_1a(2)”之后，插入图“auth_rsp_1b(2)”和图“auth_rsp_2b(2)”
102	第 C.4 章	对第 C.4 章中图“mmpdu_svc_1.1b(2)”进行替换
103	第 C.4 章	对第 C.4 章中图“ap_init_1b(3)”进行替换
104	第 C.4 章	对第 C.4 章中图“AP_Start_Bss_3c(8)”进行替换
105	第 C.4 章	对第 C.4 章中图“receive_1a(1)”进行替换
106	第 C.4 章	对第 C.4 章中图“nav_clear_1b(2)”进行替换
107	第 C.4 章	对第 C.4 章中图“start_rx_1b(2)”进行替换

修改项号	GB 15629.11—2003 章条号	修改说明
108	第 C.4 章	对第 C.4 章中图“Validate_rx_2c(2)”进行替换
109	第 C.4 章	对第 C.4 章中图“pre_filter_1d(4)”进行替换
110	第 C.4 章	对第 C.4 章中图“filter_sta_2b(4)”进行替换
111	第 C.4 章	对第 C.4 章中图“report_rx_4a(4)”进行替换
112	第 C.4 章	在第 C.4 章中图“report_rx_4a(4)”之后，插入图“filter_1b(3)”
113	第 C.4 章	对第 C.4 章中图“degragment_3c(3)”进行替换
114	附录 D	对附录 D 内容进行替换
115		在附录 D 的后面增加附录 E

本部分由西安西电捷通无线网络通信有限公司负责起草，参加单位有国家密码管理局商用密码研究中心、国家无线电监测中心、中国电子技术标准化研究所、西安电子科技大学、西安邮电学院、国家信息安全工程技术研究中心、中国信息安全产品测评认证中心、江南计算技术研究所、中国人民解放军信息安全测评认证中心等。

本部分主要起草人：赖晓龙、曹军、铁满霞、张变玲、黄振海、李大为、宋起柱、郭宏、王育民、沈昌祥、陈晓桦、李广森、叶续茂、庞辽军、刘平、涂学峰、李建东、阚润田、窦向阳、姚忠邦、徐冬梅、简练、刘伟、许福英、朱志祥、罗鹏、赵铁山。

1. 在第2章中删除原规范性引用文件中的第4～9项，并增加新规范性引用文件：

GB/T 16262.1—2006 信息技术 抽象语法记法一(ASN.1) 第1部分：基本记法规范(ISO/IEC 8824-1:1995,IDT)

GB/T 16262.2—2006 信息技术 抽象语法记法一(ASN.1) 第2部分：信息客体规范(ISO/IEC 8824-2:1995,IDT)

GB/T 16262.3—2006 信息技术 抽象语法记法一(ASN.1) 第3部分：约束规范(ISO/IEC 8824-3:1995,IDT)

GB/T 16262.4—2006 信息技术 抽象语法记法一(ASN.1) 第4部分：ASN.1规范的参数化(ISO/IEC 8824-4:1995,IDT)

GB/T 16263.1—2006 信息技术 ASN.1编码规则 第1部分：基本编码规则(BER)、正则编码规则(CER)和非典型编码规则(DER)规范(ISO/IEC 8825-1:1995,IDT)

GB/T 16263.2—2006 信息技术 ASN.1编码规则 第2部分：紧缩编码规则(PER)规范(ISO/IEC 8825-2:1996,IDT)

GB/T 17969.1—2000 信息技术 开放系统互连OSI登记机构的操作规程 第1部分：一般规程

《无线局域网产品密码算法应用指南》 国家密码管理局商用密码研究中心 2005年12月

ISO/IEC 8802.11:1999 信息技术 系统间远程通信和信息交换 局域网和城域网 特定要求 第11部分：无线局域网媒体访问控制(MAC)和物理(PHY)层规范

RFC3280 因特网X.509公钥证书基础结构(PKI)证书和证书吊销列表(CRL)框架

2. 按照英文字母顺序在第3章增加以下项，并重新编号：

3.51

鉴别器实体(AE) authenticator entity

为鉴别请求者在接入服务之前提供鉴别操作的实体。该实体驻留在AP或STA中。

3.52

鉴别和密钥管理(AKM)套件 authentication and key management (AKM) suite

鉴别和密钥管理方法的集合。

3.53

鉴别服务实体(ASE) authentication service entity

为鉴别器和鉴别请求者提供身份鉴别服务的实体。该实体驻留在ASU中。

3.54

鉴别服务单元(ASU) authentication service unit

基本功能是实现对用户证书的管理和用户身份的鉴别等，是基于公钥密码技术的WAI鉴别基础结构中重要的组成部分。

3.55

鉴别请求者实体(ASUE) authentication supplicant entity

在接入服务之前请求进行鉴别操作的实体。该实体驻留在STA中。

3.56

大数结尾 Big-endian

计算机存储多八位位组数据的一种方式，最高数据有效八位位组在最低地址。

3.57

基密钥(BK) base key(BK)

用于导出单播会话密钥的密钥。基密钥由证书鉴别过程协商得到或由预共享密钥导出。

3.58

基密钥安全关联(BKSA)　base key security association (BKSA)

证书鉴别过程的结果或预共享密钥导出的结果。

3.59

密钥加密密钥(KEK)　key encryption key(KEK)

用于密钥管理协议中密钥数据字段的加密密钥。

3.60

小数结尾　little-endian

计算机存储多八位位组数据的一种方式,最低数据有效八位位组在最低地址。

3.61

消息鉴别密钥(MAK)　message authentication key(MAK)

提供密钥管理协议数据源鉴别和完整性校验的密钥。

3.62

消息完整性校验码(MIC)　message integrity code(MIC)

用对称密钥通过密码算法对输入数据运算后产生的数值。如果改变了输入数据,没有正确的密钥将不能生成正确的校验码。

3.63

组播会话密钥(MSK)　multicast session key(MSK)

用于保护站点发送的组播 MPDU 的随机值。组播会话密钥由组播主密钥导出,它包括组播加密密钥和组播完整性校验密钥。

3.64

组播会话密钥安全关联(MSKSA)　multicast session key security association (MSKSA)

组播密钥通告过程的结果。

3.65

通告主密钥(NMK)　notification master key(NMK)

用于导出组播/站间加密密钥和组播/站间完整性校验密钥的辅助密钥。在组播密钥通告过程中,通告主密钥为组播主密钥;在站间密钥通告过程中,通告主密钥为站间主密钥。

3.66

预共享密钥(PSK)　preshared key (PSK)

发布给 STA 的静态密钥。该密钥的发布方法超出本规范范围。

3.67

站间会话密钥(STAKey)　STAKey

用来保护一个基本服务集(BSS)中站与站之间的直接通信的对称密钥。站间会话密钥由站间主密钥导出,它包括站间加密密钥和站间完整性校验密钥。

3.68

站间密钥安全关联(STAKeySA)　STAKey security association (STAKeySA)

基本服务集(BSS)中的站与站之间通过 AP 通告的站间密钥过程的结果。一个站间密钥安全关联包含一个站间密钥与有关的其他参数。

3.69

单播会话密钥(USK)　unicast session key (USK)

由基密钥通过伪随机函数导出的随机值。它分为四个部分:单播加密密钥、单播完整性校验密钥、

消息鉴别密钥、密钥加密密钥。

3.70

单播会话密钥安全关联(USKSA)　unicast session key security association (USKSA)

单播密钥协商过程的结果。

3.71

无线局域网鉴别基础结构(WAI)　WLAN authentication infrastructure

本规范定义的用于无线局域网中身份鉴别和密钥管理的安全方案。

3.72

WAPI 密钥管理　WAPI key management

包括单播密钥协商、组播密钥通告和站间密钥通告的密钥管理。

3.73

WAPI 安全网络　WAPI Security Network

启用 WAPI 安全机制的网络。WAPI 网络由信标等帧中的 WAPI 信息元素标识。

3.74

无线局域网保密基础结构(WPI)　WLAN privacy infrastructure

本规范定义的用于无线局域网中数据传输保护的安全方案,包括数据加密、数据鉴别和重放保护等功能。

3. 按照英文字母顺序在第 4 章增加以下项:

ADDID	地址索引
AKM	鉴别和密钥管理
BK	基密钥
BKID	基密钥标识
BKSA	基密钥安全关联
ECDH	椭圆曲线密码体制的 Diffie-Hellman 交换
IV	初始化向量
KEK	密钥加密密钥
MAK	消息鉴别密钥
MCK	组播完整性校验密钥
MDF	管理定义字段
MEK	组播加密密钥
MIC	消息完整性校验码
MSK	组播会话密钥
MSKID	组播会话密钥索引
MSKSA	组播会话密钥安全关联
NMK	通告主密钥
OID	客体标识符
OUI	组织全球惟一性标识
PSK	预共享密钥
STAKeyID	站间密钥索引
STAKeySA	站间密钥安全关联
UCK	单播完整性校验密钥
UEK	单播加密密钥

USK	单播会话密钥
USKID	单播会话密钥索引
USKSA	单播会话密钥安全关联

4. 在 5.4.3.1 内容之后增加以下内容：

在 WAPI ESS 中，开放系统链路验证是必须的，而解除链路验证将导致与被解除链路验证的站间所有的关联终止，同时对应于被解除链路验证的站的受控端口状态设置为 Off 以及 USKSA 被删除。解除链路验证通知经过 MAC 层提供给 WAI。

对应 WAPI，解除链路验证还删除站中任何相关的 USKSA、MSKSA 和 STAKeySA，将相应的 WAI 受控端口状态设置为 Off。如果 BK 缓存没有启用，解除链路验证还将删除 USKSA 对应的 BK-SA。

在 WAPI IBSS 中，开放系统链路验证时可选的。但一个站要能够识别解除链路验证帧，解除链路验证将导致对应于被解除链路验证站的 WAI 受控端口状态被设置为 Off 以及 USKSA 被删除。

5. 删除 5.4.3.1.1。

6. 在 5.4.3.3 内容之后增加以下内容：

本规范支持用于实现站间双向鉴别的无线局域网鉴别基础结构(WAI)，它在链路验证和关联后建立。在 BSS 中，当鉴别成功后，站可以安全地访问 AP；否则，AP 和 STA 都拒绝和对方进行通信。在 IBSS 中，当鉴别成功后，站可以安全地访问对端站；否则，两个站都拒绝和对方通信。详细的定义在第 8 章。

在 BSS 或 IBSS 中，WAI 支持基于证书的鉴别和基于共享密钥的鉴别。证书鉴别利用公钥证书来鉴别站和 AP。

对于 WAPI，WAI ASUE 和 AE 通过 WAI 非受控端口交换协议信息，只有当 WAI 鉴别过程通过 WAI 非受控端口成功完成后，WAI 受控端口才允许数据通过。

7. 5.4.3.4 的内容替换为：

在有线 LAN 中，只有在物理上连接到线缆的站可以侦听 LAN 的通信量。对无线共享媒体，情况不是这样。任何一台符合本规范的 STA 都可以侦听到在覆盖范围内所有具有与该 STA 相同的 PHY 的通信量。因此没有保密机制的无线链路连接到有线 LAN 上，会严重降低有线 LAN 的安全等级。

为使无线 LAN 的功能达到有线 LAN 设计中隐含的等级，本规范提供了加密消息内容的能力。该功能由保密服务提供。保密是一种 SS。

本规范规定了 WLAN 保密基础结构 (WPI)，它用来保护非 WAPI 协议数据，实现数据的加密、完整性、防重放等功能。详细信息见第八章。

本规范使用 WPI 机制实现实际的消息加密。为支持 WPI 提供了 MIB 功能。

注意保密可仅被数据帧调用。所有的站初始启动是“不加密的”以建立链路验证、鉴别和保密服务。

对于所有本规范的 STA，默认的保密状态为“不加密”。如果保密服务没有被调用，所有消息应不加密发送。如果该默认状态未被某一方或另一方接受，则 LLC 实体之间将不能成功地进行数据帧通信。配置成强制加密模式的 STA 接收到未加密的数据帧，或加密的数据帧使用接收站不支持的密钥，这些帧均会被丢弃，而不告知 LLC(或当 AP 接收到“To DS”字段置位的帧时，将其丢弃，而不告知分发服务)。为避免重传过程浪费 WM 带宽，这些帧将在 WM 上被确认[若接收时没有出现帧检验序列(FCS)错误]。

8. 5.5 的内容替换为：

每个 STA 需要为所有通过无线媒体与自己直接通信的 STA 维护两个状态变量：

——链路验证状态：值为未链路验证和已链路验证；

——关联状态：值为未关联和已关联。

这两个变量为每个远端 STA 建立了三种本地状态：

——状态 1：未链路验证，未关联（初始启动状态）；

——状态 2：已链路验证，未关联；

——状态 3：已链路验证，已关联。

图 8 给出了这些站状态变量与服务间的关系。

图 8　状态转换图

源 STA 和目的 STA 的当前状态决定了这两个 STA 之间允许交换的帧类型。状态 1 只允许第 1 类帧；状态 2 允许第 1 类和第 2 类帧；状态 3 允许所有类型的帧（即第 1、2 和 3 类帧）。帧类型定义如下：

5.5.1　第 1 类帧（状态 1、2 与 3 均允许）

a) 控制帧

1) RTS；

2) CTS；

3) ACK；

4) CF-End +ACK；

5) CF-End。

b) 管理帧

1) 探询请求/响应；

2) 信标；

3) 链路验证：链路验证成功后使 STA 能交换第 2 类帧，而链路验证不成功则使 STA 处于状态 1；

4）解除链路验证：处于状态 2 或状态 3 时，解除链路验证通告使 STA 状态改变到状态 1，发送第 2 类帧前，STA 应再次处于已链路验证状态；

5）通告通信量指示消息（ATIM）。

c）数据帧

数据：帧控制比特“To DS”和“From DS”均未置位的数据帧。

5.5.2 第 2 类帧（当且仅当已链路验证的状态 2、3 允许）

管理帧：

a）关联请求/响应

关联成功后允许交换第 3 类帧，而关联失败则使 STA 处于状态 2。

b）重新关联请求/响应

重新关联成功后使 STA 允许交换第 3 类帧，而重新关联失败则使 STA（重新关联消息所发送的目的 STA）处于状态 2。只有在发送方 STA 已关联在相同 ESS 内时，重新关联帧才能发送。

c）解除关联

处于状态 3 时，解除关联通告使 STA 状态变为状态 2。如果该 STA 希望使用 DS，则该 STA 应再次处于状态 3。

如果 STA A 从一个没有与其建立链路验证的 STA B 那里接收到一个地址 1 字段中有单播地址的第 2 类帧，则 STA A 应发送解除链路验证帧到 STA B。

5.5.3 第 3 类帧（当且仅当已关联的状态 3 允许）

a）数据帧

数据子类型：允许传送的数据帧，即为了使用 DSS，帧控制比特“To DS”或“From DS”可被置位。

b）管理帧

管理帧：处于状态 3 时，解除链路验证通告意味着同时解除关联，将 STA 从状态 3 变为状态 1。与其他站点关联前，STA 应再次处于已链路验证状态。

c）控制帧

PS-Poll。

如果 STA A 从已链路验证但还未关联的 STA B 那里接收到一个地址 1 字段中有单播地址的第 3 类帧，则 STA A 将发送解除关联帧到 STA B。

如果 STA A 从没有与其建立链路验证的 STA B 那里接收到一个地址 1 字段中有单播地址的第 3 类帧，则 STA A 将发送解除链路验证帧到 STA B。

注：在该条款中，“接收”指的是满足第 8 章和第 9 章定义的所有过滤准则的帧。

9. 5.7.5 的内容替换为：

STA 要调用 WPI 保密算法（由相关的 MIB 属性控制，见第 11 章），保密服务使 MPDU 加密。

对于采用保密服务的 STA，保密服务对 MPDU 进行加、解密。

10. 5.7.6 的内容替换为：

当一台 STA 同另一台 STA 进行链路验证时，链路验证服务引发一个用于交换的链路验证管理帧。链路验证算法在管理帧体中进行标识。

在 IBSS 环境中，任一 STA 都可能是启动 STA（STA 1）；而在 ESS 环境中，STA 1 为移动 STA，STA 2 为 AP。

本规范定义了一种链路验证服务的子类型：开放系统。所采用的子类型在链路验证管理帧中指明，因此链路验证帧标识了链路验证算法。类型为链路验证的管理帧将使用单播帧传输。类型为解除链路验证的管理帧则为通告帧，因此可以使用组播帧发送。两个站按照下面的描述成功地进行链路验证交

换后，双向的链路验证关系就建立了。在 BSS 中，链路验证在站和 AP 将建立，在 IBSS 中，链路验证在两个站之间建立。

开放系统链路验证是最简单的一种链路验证算法，实际上它是一个空链路验证算法。如果一个站的 dot11Authentication 置为开放系统链路验证，任何请求与它建立开放系统链路验证的的站都可以成功，但一个站可以拒绝和任何特定的站建立开放系统链路验证。开放系统链路验证包含一个两步的交换序列，第一步身份标识和请求链路验证，第二步是链路验证响应及结果，如果结果为成功，则两个站的双向链路验证将建立。

链路验证(序列的第一帧)

——消息类型：管理；

——消息子类型：链路验证；

——信息项：

a) 链路验证算法标识，取值为“开放系统”；

b) STA 身份声明(由头中的 SA 字段确定)；

c) 链路验证过程序列号为 1；

——消息方向：序列的第一帧总是从 STA 1 到 STA 2。

链路验证(序列的最后一帧)

——消息类型：管理；

——消息子类型：链路验证；

——信息项：

a) 链路验证算法标识，取值为“开放系统”；

b) 链路验证过程序列号为 2；

c) 请求链路验证的结果，取值为“successful”(成功)或“unsuccessful”(不成功)，具体定义见 7.3.1.9。

——消息方向：从 STA 2 到 STA 1。

11. 5.8 的图 11 替换为：

图 11 适用于本规范的基本参考模型构成

12. 在 5.8 内容之后增加 5.9。

5.9 安全关联的建立

5.9.1 基础结构

STA 通过被动侦听信标帧或主动探询获得 AP 的安全策略，如图 11a 所示。如果 AP 使用的是 WAI 证书鉴别和密钥管理机制，AP 发送鉴别激活分组启动证书鉴别过程，证书鉴别过程成功结束后，

AP 和 STA 进行单播密钥协商过程和组播密钥通告过程。如果 AP 使用的是预共享密钥机制，AP 和 STA 直接进行单播密钥协商过程和组播密钥通告过程。

STA 和 AP 之间的单播数据利用单播密钥协商过程所协商推导出的单播加密密钥 UEK、单播完整性校验密钥 UCK 进行保护；AP 利用由自己通告的组播主密钥导出的组播加密密钥 MEK、组播完整性校验密钥 MCK 保护发送广播/组播数据，STA 接收时采用由 AP 通告的组播主密钥所导出的组播加密密钥 MEK、组播完整性校验密钥 MCK 进行解密。

图 11a 基础结构模式下安全关联的建立

5.9.2 自组网模式

IBSS 模式下基于预共享密钥的鉴别和密钥管理过程的启动：

在 IBSS 模式下，两个 STA 间应进行两次五步握手（其中前三步握手完成单播密钥协商过程，后两步握手完成组播密钥通告过程），协商出两个独立的单播密钥，并通告各自的组播密钥。实际通信过程中，一对 STA 间的单播数据采用 MAC 地址较大的 STA 作为 AE 启动的单播密钥协商过程所协商推导出的单播加密密钥 UEK、单播完整性校验密钥 UCK 进行加解密，而 MAC 地址较小的 STA 作为 AE 启动的单播密钥协商过程所协商推导出的单播加密密钥 UEK、单播完整性校验密钥 UCK 并不采用；每个 STA 发送的广播/组播数据采用由自己通告的组播主密钥导出的组播加密密钥 MEK、组播完整性校验密钥 MCK 进行加密，接收时采用由发送方 STA 通告的组播主密钥所导出的组播加密密钥 MEK、组播完整性校验密钥 MCK 进行解密。

基于共享密钥的鉴别和密钥管理过程启动原因或条件如下：

a）当 STA 从另一个尚未与之完成单播密钥协商的 STA 处收到信标帧或探询响应帧，启动单播密钥协商过程；

b）当 STA 从另一个尚未与之完成单播密钥协商的 STA 处收到单播密钥协商过程的第一个协议分组时，启动单播密钥协商过程；

c）当 STA 收到 WPI 封装的广播/组播数据，但又没有相应的组播密钥解密时，以 MLME-PROTECTEDFRAMEDROPPED. indication 原语通知本 STA 的站管理实体。若站管理实体收到该原语，且尚未与该原语中描述的 STA 完成单播密钥协商，则启动单播密钥协商过程。

IBSS 模式下基于证书的鉴别和密钥管理过程的启动：

当两个 STA 选择使用基于证书的鉴别方法，它们将各自发起证书鉴别过程、单播密钥协商过程和组播密钥通告过程，建立两个独立的基密钥 BK、两个独立的单播密钥 USK，并通告各自的组播密钥。实际通信过程中，STA 对间的单播数据采用 MAC 地址较大的 STA 作为 AE 启动的单播密钥协商过程

所协商推导出的单播加密密钥 UEK、单播完整性校验密钥 UCK，而 MAC 地址较小的 STA 作为 AE 启动的单播密钥协商过程所协商推导出的单播加密密钥 UEK、单播完整性校验密钥 UCK 并不采用；每个 STA 发送的广播/组播数据采用由自己通告的组播主密钥导出的组播加密密钥 MEK、组播完整性校验密钥 MCK 进行加密，接收时采用由发送方 STA 通告的组播主密钥所导出的组播加密密钥 MEK、组播完整性校验密钥 MCK 进行解密。

基于证书的鉴别和密钥管理过程启动原因或条件如下：

a) 当 STA 从另一个尚未与之完成证书鉴别的 STA 处收到信标帧或探询响应帧，启动证书鉴别过程；

b) 当 STA 从另一个尚未与之完成证书鉴别的 STA 处收到证书鉴别过程的第一个协议分组时，启动证书鉴别过程；

c) 当 STA 收到 WPI 封装的广播/组播数据，但又没有相应的组播密钥解密时，以 MLME-PROTECTEDFRAMEDROPPED. indication 原语通知该 STA 的站管理实体。若站管理实体收到该原语，且尚未与该原语中描述的 STA 完成证书鉴别，则启动证书鉴别过程。

若 STA 不知道对端 STA 的 WAPI 安全参数，则在启动鉴别和密钥管理过程之前，应向对端 STA 发送探询请求帧，通过对端 STA 回应的探询响应帧获得对方的安全机制。

13. 在 6.1.3 内容之后增加 6.1.4。

6.1.4 MAC 数据服务基础结构

MAC 数据平面结构(即一个 MSDU 全部或部分传输的处理过程)如图 11b 所示。

在发送端，一个 MSDU 经过节电模式下的帧交付延迟、序列号分配、分段、完整性保护、加密、组帧等部分或全部处理过程。WAPI 的受控端口可以阻止 MSDU 的传输。

在接收端，收到的数据经过 MPDU 帧头与 FCS 的校验、重复帧丢弃、解密、完整性校验、重放检测、重组等处理过程。若受控端口未打开或 MSDU 为非 WAPI 协议数据，则受控/非受控端丢弃该 MSDU。在 MSDU 重组前，WPI-SMS4 的 MPDU 帧必须严格排序，否则，重组将失败。

图 11b MAC 数据平面结构

14. 7.1.3.1 中 7.1.3.1.1 之前的内容替换为：

帧控制字段包含以下子字段：协议版本、类型、子类型、去往 DS、来自 DS、多分段标记、重传、功率管理、多数据标记、帧保护标识及排序。帧控制字段的格式如图 13 所示。

B0 B1B2	B3B4 B7	B8	B9	B10	B11	B12	B13	B14	B15
协议版本	类型	子类型	到 DS	来自 DS	多分段标识	重传	功率管理	多数据标识	帧保护标识	排序
比特数 2	2	4	1	1	1	1	1	1	1	1

图 13 帧控制字段

15. 7.1.3.1.9 替换为：

7.1.3.1.9 **帧保护标识字段**

帧保护标识字段长度为 1 比特。如果帧体中包含的信息经密码算法处理并封装，此字段值置为 1。此字段的值仅在数据帧中可被置为 1，其他类型的帧中此字段置为 0。

16. 在 7.2.3.1 的表 5 中增加以下项：

表 5 信标帧体

顺 序	信 息	备 注
22	WAPI 信息元素	WAPI 信息元素字段仅在启用 WAPI 机制的 STA 生成的信标帧中包含

17. 在 7.2.3.4 的表 7 中增加以下项：

表 7 关联请求帧体

顺 序	信 息	备 注
9	WAPI 信息元素	WAPI 信息元素字段仅在启用 WAPI 机制的 STA 生成的关联请求帧中包含

18. 在 7.2.3.6 的表 9 中增加以下项：

表 9 重新关联请求帧体

顺 序	信 息	备 注
10	WAPI 信息元素	WAPI 信息元素字段仅在启用 WAPI 机制的 STA 生成的重新关联请求帧中包含

19. 在 7.2.3.6 的表 9 中增加以下项：

表 9 重新关联请求帧体

顺 序	信 息	备 注
10	WAPI 信息元素	WAPI 信息元素字段仅在启用 WAPI 机制的 STA 生成的重新关联请求帧中包含

20. 在 7.2.3.9 的表 12 中增加以下项：

表 12 探询响应帧体

顺 序	信 息	备 注
22	WAPI 信息元素	WAPI 信息元素字段仅在启用 WAPI 机制的 STA 生成的探询响应帧中包含
23～n	请求信息元素	探询请求帧的请求信息元素

21. 7.3.1.4 的内容替换为：

能力信息字段包括用来指示请求或广播能力的多个子字段。该字段长度为 2 个八位位组，包括 ESS、IBSS、CF-Pollable、CF-Poll 请求和保密子字段，其余子字段为保留。能力信息字段格式如图 27 所示。

图 27 能力信息固定字段

仅在管理帧子类型中才解释每个能力信息子字段，以定义传输规则。

在发送的信标或探询响应帧中，AP 设置 ESS 子字段为 1，IBSS 子字段为 0；而 IBSS 的 STA 设置 ESS 子字段为 0，IBSS 子字段为 1。

STA 根据表 15 设置关联和重新关联请求管理帧的 CF-Pollable 和 CF-Poll 请求子字段。

表 15 STA 对 CF-Pollable 子字段和 CF-Poll 请求子字段的用法

CF-Pollable	CF-Poll 请求	含　　义
0	0	STA 为非 CF-Pollable
0	1	STA 为 CF-Pollable，没有请求放置于 CF-Polling 列表中
1	0	STA 为 CF-Pollable，请求放置于 CF-Polling 列表中
1	1	STA 为 CF-Pollable，请求从不被轮询

AP 根据表 16 设置信标、探询响应、关联响应和重新关联响应等管理帧的 CF-Pollable 和 CF-Poll 请求子字段。AP 将关联响应和重新关联响应管理帧的 CF-Pollable 和 CF-Poll 请求子字段的值设置为与 AP 发送的最后的信标或探询响应帧中的值相同。

表 16 AP 对 CF-Pollable 子字段和 CF-Poll 请求子字段的用法

CF-Pollable	CF-Poll 请求	含　　义
0	0	AP 上无点协调器
0	1	AP 上的点协调器仅用于交付(非轮询)
1	0	AP 上的点协调器用于交付和轮询
1	1	保留

如果对 BSS 中交换的所有数据类型的帧启用了加密保护，AP 在发送的信标帧、探询响应帧、关联响应帧和重新关联响应帧中设置保密子字段值为 1。如果数据加密保护未启用，AP 设置管理帧中保密子字段为 0。

在一个 BSS 内非 AP 的 STA 在发送的关联和重新关联帧中设置保密子字段值为 0，AP 忽略关联和重新关联帧中的保密子字段。

如果对 IBSS 中交换的所有数据类型的帧启用了加密保护，STA 在发送的信标帧、探询响应帧、关联响应帧和重新关联响应帧中设置保密子字段值为 1。如果数据加密保护未启用，STA 设置管理帧中保密子字段为 0。

在 STA 发送的信标帧和探询响应帧中包含 WAPI 信息元素时，保密子字段设置为 1。

22. 在 7.3.1.7 的表 17 中增加原因码 10～32，并修改最后一行保留项：

表 17 原因码

原因码	含　　义
10～12	保留
13	无效的信息元素
14	消息鉴别码校验失败
15～19	保留
20	无效的 AKMP
21～23	保留
24	根据安全策略而拒绝的密码套件
25	单播密钥协商超时
26	组播密钥通告超时
27	单播密钥协商中的信息元素与(重新)关联请求/探询响应/信标帧中的信息元素不一致
28	无效的组播密码套件
29	无效的单播密码套件
30	不支持的 WAPI 信息元素版本
31	无效的 WAPI 信息元素能力
32	WAI 证书鉴别失败
33～65 535	保留

23. 在 7.3.1.9 的表 18 中增加状态码 22～50,并修改最后一行保留项:

表 18 状态码

状态码	含　　义
22～39	保留
40	无效的信息元素
41～42	保留
43	无效的 AKMP
44～45	保留
46	根据安全策略而拒绝的密码套件
47	无效的单播密码套件
48	无效的组播密码套件
49	不支持的 WAPI 信息元素版本
50	无效的 WAPI 信息元素能力
51～65 535	保留

24. 在 7.3.2 的表 19 中增加以下项：

表 19　元素 ID

信息元素	元素 ID
WAPI 信息元素	68

25. 在 7.3.2.7 之后增加 7.3.2.8。

7.3.2.8　WAPI 信息元素

WAPI 信息元素包含鉴别和保密套件选项，见图 41a。所有的实现 WAPI 的 STA 应支持该信息元素。WAPI 信息元素的长度应最大为 255 八位位组。

元素标识 ID	长度	版本	鉴别和密钥管理(AKM)套件计数	鉴别和密钥管理(AKM)套件	单播密码套件计数	单播密码套件	组播密码套件	WAPI 能力信息	BKID 计数	BKID 列表
1	1	2	2	4×m	2	4×n	4	2	2	16×s

八位位组数

图 41a　WAPI 信息元素格式

——元素标识 ID 应为 68。

——长度字段标识 WAPI 信息元素中除元素标识 ID 和长度字段以外的字段的八位位组数。

——版本字段标识 WAPI 协议的版本号，本规范中版本号为 1，其他值保留。

——鉴别和密钥管理(AKM)套件计数字段标识 STA 支持的鉴别和密钥管理机制个数。

——鉴别和密钥管理(AKM)套件字段包含 STA 支持的鉴别和密钥管理机制，m 为鉴别和密钥管理套件计数字段的值。

——单播密码套件计数字段标识 STA 支持的单播密码算法个数。

——单播密码套件字段包含 STA 支持的单播密码算法，n 为单播密码套件计数字段的值。

——组播密码套件字段包含 STA 支持的组播密码算法。

——WAPI 能力信息：

WAPI 能力信息字段表示请求或声明的能力信息，该字段长度为 2 个八位位组，各比特定义如图 41b 所示。

B0	B1 … B15
预鉴别	保留

图 41b　WAPI 能力信息

比特 0 为预鉴别标识。AP 如果支持预鉴别，设置此字段值为 1，否则设置为 0。非 AP 的 STA 设置此字段值为 0。

——BKID 计数和列表字段。

BKID 计数和列表字段仅用于发往 AP 的关联或重新关联请求帧中。BKID 计数字段表示 BKID 列表字段中包含的 BKID 的个数。BKID 列表字段包含 0 个或多个 STA 当前缓存的和目的 AP 之间的有效的 BKID，s 为 BKID 计数字段的值。其中 BKID 可能为：

a) STA 缓存的通过和目的 AP 预鉴别得到的 BKID；

b) STA 缓存的通过证书鉴别得到的 BKID；

c) STA 缓存的通过预共享密钥得到的 BKID。

7.3.2.8.1　鉴别和密钥管理套件

套件选择格式如图 41c 所示。

图 41c　套件选择格式

表 19a 提供本规范定义的套件。

表 19a　鉴别和密钥管理套件

OUI 3 八位位组	类型 1 八位位组	含　　义
00-14-72	0	保留
00-14-72	1	WAI 证书鉴别和密钥管理
00-14-72	2	WAI 预共享密钥鉴别和密钥管理
00-14-72	3～255	保留
其他	0～255	保留

7.3.2.8.2　单播密码套件

套件选择格式如图 41d。

图 41d　套件选择格式

表 19b 提供本规范定义的套件。

表 19b　密码套件

OUI 3 八位位组	类型 1 八位位组	含　　义
00-14-72	0	保留
00-14-72	1	WPI-SMS4
00-14-72	2～255	保留
其他	0～255	保留

7.3.2.8.3　组播密码套件

表 19c 提供本规范定义的套件。

表 19c　密码套件

OUI 3 八位位组	类型 1 八位位组	含　　义
00-14-72	0	保留
00-14-72	1	WPI-SMS4
00-14-72	2～255	保留
其他	0～255	保留

26. 第 8 章的内容替换为：

8　安全[1)]

采用无线局域网鉴别与保密基础结构 WAPI 为系统提供安全保护。WAPI 包含以下部分：

1) 本部分涉及的密码算法相关信息与国家密码管理局联系。

——WAI 鉴别及密钥管理；

——WPI 数据传输保护。

8.1 WAI 鉴别及密钥管理

支持 WAI 鉴别及密钥管理的 STA 通过以下四种方式实现。

a) 在 BSS 中基于证书的方式

1) STA 通过 AP 的信标帧或探询响应帧识别 AP 支持 WAI 鉴别及密钥管理套件；

2) STA 和 AP 之间进行链路验证；

3) 在关联过程中，STA 在关联请求中包含 WAPI 信息元素确定选择的密码套件；

4) STA 和 AP 进行证书鉴别过程，协商出 BK；

5) STA 和 AP 进行单播密钥协商过程、组播密钥通告过程；

6) 把协商出来的密钥和密码套件通知 WPI 模块，进行数据传输保护。

b) 在 BSS 中基于共享密钥(在 STA 和 AP 之间)的方式

1) STA 通过 AP 的信标帧或探询响应帧识别 AP 支持 WAI 鉴别及密钥管理套件；

2) STA 和 AP 之间进行链路验证；

3) 在关联过程中，STA 在关联请求中包含 WAPI 信息元素确定选择的密码套件；

4) 预共享密钥导出 BK 后，STA 和 AP 进行单播密钥协商过程、组播密钥通告过程；

5) 把协商出来的密钥和密码套件通知 WPI 模块，进行数据传输保护。

c) 在 IBSS 中基于证书的方式

1) STA 通过对端 STA 的信标帧或探询响应帧识别支持的 WAI 鉴别及密钥管理套件；

2) STA 和对端 STA 进行可选的链路验证；

3) STA 和对端 STA 进行证书鉴别过程，协商出 BK；

4) STA 和对端 STA 进行单播密钥协商过程、组播密钥通告过程；

5) 把协商出来的密钥和密码套件通知 WPI 模块，进行数据传输保护。

d) 在 IBSS 中基于共享密钥的方式

1) STA 通过对端 STA 的信标帧或探询响应帧识别支持的 WAI 鉴别及密钥管理套件；

2) STA 和对端 STA 进行可选的链路验证；

3) STA 和对端 STA 进行单播密钥协商过程、组播密钥通告过程(预共享密钥导出 BK)；

4) 把协商出来的密钥和密码套件通知 WPI 模块，进行数据传输保护。

如果 STA 与 AP/STA 关联时选择采用 WAPI 安全机制，则必须进行相互身份鉴别和密钥协商。若采用基于证书的方式，整个过程包括证书鉴别、单播密钥协商与组播密钥通告；若采用预共享密钥的方式，整个过程则为单播密钥协商与组播密钥通告。

STA 与 AP /STA 之间的鉴别数据分组利用以太类型字段为 0x88B4 的 WAPI 协议传送，AP/STA 与 ASU 之间的鉴别数据报文通过 UDP 套接口传输，ASU 的端口号为 3810。

WAI 鉴别和密钥管理完成的时间必须小于 MIB 值 gb15629dot11wapiConfigSATimeout，它开始于 STA 的站管理实体决定建立 WAPI 安全网络，到 MLME-SETPROTECTION. request 原语被激发结束。若在 MIB 值 gb15629dot11wapiConfigSATimeout 的时间内没有完成安全关联的建立，则两个 STA 将解除链路验证。

8.1.1 鉴别系统结构

8.1.1.1 系统和端口

STA 提供两种访问 LAN 的逻辑通道，定义为两类端口，即受控端口与非受控端口。

一个 STA 提供其他 STA 连接到鉴别服务单元(ASU)的端口(即非受控端口)，确保只有鉴别成功的 STA 才能使用该 STA 提供的数据端口(即受控端口)访问网络或收发数据。在基于端口的接入控制操作中，本规范定义三个实体：

a) 鉴别器实体 AE(Authenticator Entity):为鉴别请求者实体在接入服务之前提供鉴别操作的实体。该实体驻留在 AP 或 STA 中;

b) 鉴别请求者实体 ASUE(Authentication SUpplicant Entity):在接入服务之前请求进行鉴别操作的实体。该实体驻留在 STA 中;

c) 鉴别服务实体 ASE(Authentication Service Entity):为鉴别器实体和鉴别请求者实体提供相互鉴别服务的实体。该实体驻留在 ASU 中。

鉴别器实体和鉴别请求者实体都称为鉴别子系统。

8.1.1.2 受控和非受控接入

下图描述了鉴别子系统中受控端口和非受控端口。

非受控端口允许鉴别数据(WAPI 协议数据)在 WLAN 中传送,该传送过程不受当前鉴别状态的限制。对于受控端口,只有当该端口的鉴别状态为已鉴别时,才允许协议数据(非 WAPI 协议的其他协议数据)通过。受控端口和非受控端口可以是连接到同一物理端口的两个逻辑端口,所有通过物理端口的数据都可以到达受控端口和非受控端口,此时根据鉴别状态决定数据的实际流向(受控端口或非受控端口)。

图 42 鉴别子系统示意图

图 43 给出了与受控端口相关的两种不同的鉴别状态 On 和 Off,分别允许和拒绝受控端口的协议数据单元 MSDU 通过。其中 On 表示端口状态为已鉴别,Off 表示端口状态为未鉴别。图 43 给出了两个系统,在鉴别子系统 1 中,受控端口鉴别状态是未鉴别,此时受控端口拒绝通过任何数据;在鉴别子系统 2 中,受控端口鉴别状态是已鉴别,受控端口允许 MSDU 通过。

图 43 受控端口的鉴别状态

系统的每一个受控端口状态由系统鉴别控制参数 gb15629dot11wapiControlledAuthControl 确定。系统鉴别控制参数的值为“启动鉴别”或“不启动鉴别”。如果系统鉴别控制参数设置为“不启动鉴别”

时，所有的受控端口的鉴别控制状态为“已鉴别”；如果系统鉴别控制参数设置为“启动鉴别”，系统的每一个受控端口的鉴别状态由鉴别控制类型 gb15629dot11wapiControlledPortControl 决定，鉴别控制类型取值如下：

a) 强制非鉴别：鉴别器实体强制某一个受控端口的状态为未鉴别，即无条件指定受控端口状态为“未鉴别”(即不允许通过该受控端口传送数据)；

b) 自动：自动是指根据鉴别器实体和鉴别请求者实体之间通过鉴别服务单元相互鉴别的结果来设定受控端口状态(只有鉴别通过才可通过受控端口传送数据)。

除鉴别数据外，系统中 STA 之间的网络协议数据交换是通过受控端口来实现的。图 44 给出了受控端口和非受控端口的逻辑结构图，系统中受控端口的鉴别状态是由鉴别器实体根据 ASU 对 STA 的鉴别结果来设定的。

图 44　受控端口和非受控端口的用法

图 45 给出了鉴别请求者、鉴别器和鉴别服务实体之间的关系及信息交换过程。在该图中，鉴别器和鉴别请求者的受控端口均处于未鉴别状态，拒绝数据通过受控端口。

图 45　鉴别系统结构

8.1.2　WAPI 安全关联的管理

8.1.2.1　WAPI 安全关联定义

安全关联是一组用来保护信息的策略和密钥，WAPI 中包含四种安全关联：

——BKSA：基密钥安全关联，是证书鉴别过程完成后或通过预共享密钥信息得到的结果；

——USKSA：单播会话密钥安全关联，单播密钥协商的结果；

——MSKSA：组播会话密钥安全关联，组播密钥通告的结果；

——STAKeySA：站间密钥安全关联，站间密钥通告的结果。

BKSA

BKSA 是双向的，当证书鉴别过程成功或设置了预共享密钥，BKSA 在 ASUE 或 AE 中创建。BKSA 用来创建 USKSA，BKSA 在它的生存期内被缓存。它包含以下内容：

——BKID，标识 BKSA；

——AE 的 MAC 地址；

——ASUE 的 MAC 地址；

——BK；

——生存期；

——AKM；

——其他安全参数(可选)。

USKSA

USKSA 是单播密钥协商的结果，它是双向的。USKSA 是基于 BK 协商的，在生存期中被缓存。对于每一对 ASUE 和 AE，最多只有两个 USKSA。在 BSS 中，一般只有一个 USKSA 处于有效状态，但在密钥更新时，会有两个 USKSA 处于有效状态，在接收到使用新 USKSA 加密的单播数据 MPDU 时，旧 USKSA 被置为无效状态。在 IBSS 中，MAC_{AE}(AE 的 MAC 地址)大于 MAC_{ASUE}(ASUE 的 MAC 地址)协商出的 USKSA 用于单播数据 MPDU 的保护，其更新密钥的状态和 BSS 相同；MAC_{AE}小于 MAC_{ASUE}协商出的 USKSA 不用于单播数据 MPDU 加密，仅用于组播密钥通告，当新的 USKSA 处于有效状态时，旧的 USKSA 立刻处于无效状态。

USKSA 包含以下内容：

——USKID；

——USK；

——选择的单播密码套件；

——生存期；

——ASUE 的 MAC 地址；

——AE 的 MAC 地址；

——其他安全参数，比如包括用于预鉴别和 STAKey 的重放计数器。

MSKSA

MSKSA 是组播密钥通告的结果。在 BSS 中，只有一个 MSKSA 处于有效状态，AP 用它加密发送的广播/组播 MPDU，STA 用它解密收到的广播/组播 MPDU。在 IBSS 中，每个 STA 有多个有效的 MSKSA，一个用来加密发送的广播/组播 MPDU，其他分别用于解密各个对端 STA 发送的广播/组播 MPDU。在进行 MSKSA 更新时，对于 AP 或每个对端 STA，STA 中用于解密广播/组播 MPDU 的 MSKSA 可以有两个处于有效状态，在使用新的 MSKSA 解密收到广播/组播 MPDU 后，旧的 MSKSA 才被置为无效状态；而用于发送加密广播/组播 MPDU 的 MSKSA 只有一个处于有效状态。一个 MSKSA 包含以下内容：

——方向(接收或发送)；

——MSKID；

——选择的组播密码套件；

——生存期；

——MSK；

——AE 的 MAC 地址；

——其他安全参数(可选)。

STAKeySA

STAKeySA 是 STAKey 协商的结果，它是从发起 STA 到对端接收 STA 的单向安全关联。一个

STAKeySA 包含以下内容：

——STAKeyID；

——STAKey；

——采用的单播密码套件(采用 AP 通告的组播密码套件)；

——发起 STA 的 MAC 地址；

——对端接收 STA 的 MAC 地址。

——其他安全参数(可选)。

8.1.2.1.1 **ESS 中的安全关联**

若采用 WAPI 安全机制，STA/AP 使用 WAPI 信息元素标识，WAPI 信息元素包含在信标帧或探询响应帧中；如果在信标帧或探询响应帧中没有 WAPI 信息元素字段，那么该 STA/AP 未启用 WAPI 安全机制。

在 ESS 中漫游的 STA 通过下面三种方法建立 BKSA：

——在关联后，执行证书鉴别过程建立 BKSA 或通过预共享密钥直接建立 BKSA；

——STA 通过执行证书鉴别过程，可缓存与 ESS 中某个 AP 建立的 BKSA，当 STA 移动到了该 AP，STA 可以在关联/重新关联请求中的 WAPI 信息元素中包含一个或多个 BKSA 对应的 BKID，若 AP 中有相对应有效的 BKSA，则可以跳过证书鉴别过程而直接进行单播密钥协商过程。如果 AP 中没有对应有效的 BKSA，或 STA 在关联/重新关联请求中的 WAPI 信息元素中没有包含 BKID，则 STA 和 AP 必须通过证书鉴别过程或预共享密钥建立 BKSA；

——若 STA 已经和 ESS 中某个 AP 完成了 BKSA 和 USKSA 的建立，它可以和该 ESS 中的其他 AP 在关联之前进行预鉴别过程，建立 BKSA。如果预鉴别过程成功完成，STA 和 AP 将缓存 BKSA，基于 BKSA，STA 通过单播密钥协商过程建立 USKSA，通过组播密钥通告过程建立 MSKSA。

8.1.2.1.2 **IBSS 中的安全关联**

如果在 IBSS 中采用 WAPI 安全机制，每对 STA 之间都要建立安全关联。任一对 STA 协商使用任何它们共同支持的单播密码套件。每个 STA 在其信标帧或探询响应帧中包含它支持的组播密码套件和单播密码套件列表。两个 STA 必须广播有交集的单播密码套件列表，并支持对方广播的组播密码套件，它们才建立 BKSA。

当 IBSS 中某个 STA 要和另一个 STA 建立安全关联，但它却不知道对方的安全策略，它必须通过探询请求帧获得对方的安全策略。两个 STA 通过单播密钥协商过程选择单播密码算法。单播密钥协商响应中的 WAPI 信息元素包含选择的单播密码套件，而单播密钥协商确认中包含的 WAPI 信息元素是 STA 在信标帧或探询响应帧中包含的 WAPI 信息元素。一对 STA 将使用 MAC 地址大的 STA 作为 AE 实体所协商的单播密码算法。

当两个 STA 选择使用基于证书的鉴别方法，它们将各自发起证书鉴别过程、单播密钥协商过程和组播密钥通告过程，建立两套 BKSA、USKSA、MSKSA。当使用预共享密钥的鉴别方法，两个 STA 的预共享密钥直接作为 BK，从而建立一个 BKSA，它们各自发起单播密钥协商过程和组播密钥通告过程建立两个 USKSA 和 MSKSA。

8.1.2.1.3 **WAPI 安全关联的删除**

当 STA/AP 收到关联、重新关联、解除关联、链路验证、解除链路验证的原语，或当它相信它已经离开了另一个 STA/AP 的无线信号范围，它将删除一些安全关联。

在 BSS 中，非 AP 的 STA 将删除 USKSA 和 MSKSA，AP 将删除 USKSA。

在 IBSS 中，STA 将删除 USKSA 和接收 MSKSA。

若某个安全关联的生存期到期或处于无效状态，该安全关联将被删除。STA 基于自己的管理策略，可以使某些安全关联处于无效状态。

8.1.2.2 **WAPI 安全策略的选择**

WAPI 机制的选择是通过关联过程完成的，STA 在(重新)关联帧中包含 WAPI 信息元素来执行 WAPI 策略的选择。

STA/AP 在通过信标帧或探询响应帧中包含 WAPI 信息元素字段标识它支持的 WAPI 安全策略，要与之关联的其他 STA 从中选择要使用的 WAPI 安全策略，在关联请求帧中用 WAPI 信息元素字段标识选择的 WAPI 安全策略，从而完成安全策略的选择；若与之关联的其他 STA 不支持它所通告的 WAPI 安全策略，则不和它进行关联。

STA 和 AP 通过以下步骤建立 WAPI 安全网络：

a) STA 根据 SSID 选择 AP；

b) STA 和 AP 进行链路验证；

c) STA 和 AP 的关联过程完成 WAPI 安全机制的选择。根据选择，STA 和 AP 执行 5.3 中的过程。

在 IBSS 中，STA 根据对端 STA 的信标帧或探询响应帧中的 WAPI 信息元素，从中选择要使用的 WAPI 安全策略，包括鉴别和密钥管理方法、组播密码套件，但不选择单播密码套件。根据所选择的 WAPI 安全策略中鉴别和密钥管理方法开始鉴别，在单播密钥协商过程中协商单播密码套件。

8.1.3 **证书**

鉴别服务单元 ASU(Authentictaion Service Unit)是基于公钥密码技术的 WAI 鉴别基础结构中重要的组成部分，它的基本功能是实现对用户证书的有效性鉴别。

用户证书为公钥证书，它是 WAI 系统构造中重要的环节。公钥证书是网络用户的数字身份凭证，通过私钥验证可以惟一地确定网络用户的身份。

本规范支持两种格式的证书：X.509 v3 和 GBW 证书。用户和 ASU 证书用于签名验证。

8.1.3.1 **X.509 v3 证书**

本规范采用的 X.509 v3 证书，其中签名算法为 ECDSA-192，杂凑算法为 SHA-256。公钥算法标识、签名算法标识以及椭圆曲线参数均采用 OID 方式表示。公钥算法字段利用 OID 值 1.2.840.10045.2.1 标识椭圆曲线算法，并在密钥用途字段中标明为签名用途；签名算法字段利用 OID 值 1.2.156.11235.1.1.1 标识基于 SHA-256 的 ECDSA-192 算法；椭圆曲线参数字段利用 OID 值 1.2.156.11235.1.1.2.1 标识国家密码管理局批准的专用于无线局域网的曲线参数。X.509 v3 证书格式参照 RFC3280。

X.509 v3 证书中的所有字段均采用 ASN.1/DER 进行编码。X.509 v3 证书以 base64binary 为编码类型、以 PEM 格式进行存储，文件名的后缀为.cer。

8.1.3.2 **GBW 证书**

8.1.3.2.1 **证书定义**

公钥证书的版本号
证书的序列号
证书颁发者名称
证书的有效期
证书持有者名称
证书持有者的公钥信息
扩展
证书颁发者对证书的签名采用的算法
证书颁发者对证书的签名

图 46 公钥证书的格式

其中：

公钥证书的版本号

该字段指定证书的格式，以使具体的协议能提取该公钥证书的有效数据项。

证书的序列号

每个由 ASU 颁发的公钥证书都需要分配一个惟一的序列号，由证书的序列号和证书颁发者的名称可以惟一地确定证书持有者。

证书颁发者名称

该字段指定证书颁发者的身份。

证书的有效期

该字段用于规定公钥证书可以有效使用的时间，采用 UTC 时间格式，表示 1970 年 1 月 1 日 0 时到当前时间的秒数。

证书持有者名称

该字段指定证书持有者的身份。

证书持有者的公钥信息

该字段为证书持有者的公钥信息。

扩展

该字段用于描述证书的增强属性。

签名算法

该字段指定了证书颁发者对证书的签名所采用的签名算法，包括杂凑算法名称和签名算法名称。本规范采用国家密码管理局批准的用于 WLAN 的椭圆曲线密码(ECC)体制实现签名算法。

证书颁发者对证书的签名

该字段由证书颁发者(ASU)对该证书上的所有字段项(除过证书颁发者对证书的签名采用的算法)进行行签名得到。

8.1.3.2.2 **证书格式**

GBW 证书格式如下：

版本号	序列号	颁发者名称	有效期	持有者名称	持有者公钥	扩展	签名算法	颁发者签名
2	4	6～256	8	6～256	可变	可变	可变	可变

八位位组数：

图 47 证书内容定义

其中：

——版本号字段长度为 2 个八位位组，表示一个整数，当前版本号为 1。

——序列号字段长度为 4 个八位位组，表示一个整数。

——颁发者名称字段由长度字段与内容字段组成。其中长度字段为 1 个八位位组，表示内容字段的八位位组数。

——有效期字段长度为 8 个八位位组，由 4 个八位位组的起始时间和 4 个八位位组的截止时间组成。起始时间与截止时间均表示从 1970 年 1 月 1 日 0 时起到当前时间的秒计数值。

——持有者名称字段由长度字段与内容字段组成。其中长度字段为 1 个八位位组，表示内容字段的八位位组数。

——持有者公钥字段由长度字段与内容字段组成。其中长度字段为 2 个八位位组，表示内容字段的八位位组数。内容字段包括公钥算法标识、公钥算法参数和公钥值字段：

a) 公钥算法标识字段长度为 1 个八位位组，值为 1 表示 ECDSA-192，其他保留；

b) 公钥算法参数字段表示签名算法的参数，由参数标识和参数长度和参数内容组成，参数标识字

段长度为 1 个八位位组，参数长度字段为 2 个八位位组，表示参数内容字段的八位位组数：

当公钥算法标识字段值为 1 时，参数字段的值定义如下：

- 参数标识为 1 时，标识参数以 OID 方式表示，参数长度字段表示 OID 标识的八位位组数，参数内容为 OID 编码，本规范采用的 ECC 参数的 OID 为 1.2.156.11235.1.1.2.1，OID 编码采用 ASN.1/DER。
- 参数标识其他值保留。

c) 公钥值字段为公钥信息，在公钥算法标识为 1 时，其长度为 49 个八位位组。

——扩展字段包括扩展属性个数与扩展属性列表两个字段。其中扩展属性个数字段为 1 个八位位组，标识扩展属性的个数；扩展属性列表字段中的每个扩展属性以 TLV 格式定义。

——签名算法字段表示颁发者采用何种算法对证书进行签名，包含长度和内容两个子字段。其长度字段为 2 个八位位组，表示内容字段的八位位组数。内容字段由 1 个八位位组的杂凑算法标识、1 个八位位组的签名算法标识和参数组成。

a) 杂凑算法标识定义如下：

1 表示 SHA-256 杂凑算法；

其他值保留。

b) 签名算法标识定义如下：

1 表示 192 位的椭圆曲线数字签名算法，即 ECDSA-192；

其他值保留。

c) 参数字段表示签名算法的参数，由参数标识和参数长度和参数内容组成，参数标识字段长度为 1 个八位位组，参数长度字段为 2 个八位位组，表示参数内容字段的八位位组数：

当签名算法标识字段值为 1 时，参数字段标识椭圆曲线的参数，定义如下：

- 参数标识为 1 时，标识参数以 OID 方式表示，参数长度字段表示 OID 标识的八位位组数，参数内容为 OID 编码，本规范采用的 ECC 参数的 OID 为 1.2.156.11235.1.1.2.1，OID 编码采用 ASN.1/DER。
- 参数标识其他值保留。

——颁发者签名字段包含长度和签名值，长度子字段为 2 个八位位组，表示签名值字段的八位位组数，签名值是颁发者对前述所有字段(除签名算法字段外)的签名值，该签名值是按照《无线局域网产品密码算法应用指南》中的规则将签名结果转化成的八位位组串。

扩展字段中的每个扩展属性采用如下 TLV 格式定义：

图 48　扩展属性

——类型字段长度为 1 个八位位组，表示扩展属性的类型，定义如下：

1 签署密钥标识；

其他类型值保留。

——长度字段长度为 2 个八位位组，标识属性值字段的八位位组数。

——属性值字段包含类型字段标识的属性内容。

当类型字段为 1 时，属性值字段表示证书颁发者的签署密钥标识，标识颁发者用于签名证书的私钥对应的公钥。签署密钥标识由颁发者证书的持有者名称字段和颁发者证书的序列号字段组成。

证书的颁发和吊销等管理方法以及 ASU 之间的通信超出本规范范围。

8.1.3.2.3 证书的颁发格式

证书颁发文件名的默认后缀为.wcr。证书文件各字段如无特殊说明均按照大数结尾顺序进行编码存储。

证书颁发格式如下：

证书文件标识	版本号	证书文件长度	摘要	属性1	属性2	属性3
八位位组数：16	2	2	4～65539	4～65539	4～65539	4～65539

图 49 证书颁发格式

其中：

证书文件标识字段表示 GBW 证书文件存储标识字段，用来检查对应的文件是否为 GBW 类型的证书颁发文件，规定 GBW 证书标识字段为“WAI 15629.11-2003”的 ASCII 码组成的字符串（十六进制表示为 0x57 0x41 0x49 0x31 0x35 0x36 0x32 0x39 0x2E 0x31 0x31 0x2D 0x32 0x30 0x30 0x33），长度为 16 个八位位组。

版本号字段为一个整数，表示证书文件颁发格式的版本号，当前值为 1，长度为 2 个八位位组。

证书文件长度字段表示证书文件中摘要字段和所有属性字段的八位位组数，长度为 2 个八位位组。

摘要字段为采用某种摘要生成算法对摘要字段后面其他字段生成的摘要。摘要字段格式如下：

摘要算法标识	摘要长度	摘要数据
八位位组数：2	2	0～65535

图 50 摘要字段

其中：

摘要算法标识表示相应的摘要生成算法，长度为 2 个八位位组。摘要算法标识定义如下：

1 表示 SHA-256 杂凑算法，此时摘要长度字段值为 32，摘要数据为 32 个八位位组的摘要内容；

其他值保留。

摘要长度字段表示摘要数据字段的总八位位组数，本字段长度为 2 个八位位组。

摘要数据字段为摘要的数据内容。

属性字段格式如下：

属性标识	属性长度	属性数据
八位位组数：2	2	0～65535

图 51 属性字段

其中：

属性标识字段表示属性类型，长度为 2 个八位位组。属性标识定义如下：

1 表示该字段的属性数据为证书颁发者的证书；

2 表示该字段的属性数据为用户证书；

3 表示该字段的属性数据为用户证书对应的私钥；

其他值保留。

属性长度字段表示该属性数据字段的八位位组数，本字段长度为 2 个八位位组。

属性数据字段为该属性的数据内容。

8.1.4 WAI 协议

8.1.4.1 WAI 协议分组格式

WAI 协议分组中各字段如无特殊说明均按照大数结尾顺序编码发送。

ASUE、AE 和 ASU 之间的 WAI 协议分组数据的格式定义如下：

版本	类型	子类型	保留	长度	分组序号	分片序号	标识	数据
2	1	1	2	2	2	1	1	可变

八位位组数：

图 52　WAI 鉴别系统的 WAI 协议分组数据基本格式

其中：

——版本字段长度为 2 个八位位组，表示鉴别基础结构的版本号。当前版本为 1；

——类型字段长度为 1 个八位位组，表示协议类型，定义如下：

1 WAI 协议分组；

其他值保留。

——子类型字段的长度为 1 个八位位组，当类型字段的值为 1 时，子类型字段值定义如下；当类型字段为其他值时，子类型字段值保留。

1 表示预鉴别开始分组；

2 表示站间密钥请求分组；

3 表示鉴别激活分组；

4 表示接入鉴别请求分组；

5 表示接入鉴别响应分组；

6 表示证书鉴别请求分组；

7 表示证书鉴别响应分组；

8 表示单播密钥协商请求分组；

9 表示单播密钥协商响应分组；

10 表示单播密钥协商确认分组；

11 表示组播密钥/站间密钥通告分组；

12 表示组播密钥/站间密钥响应分组；

其他值保留。

——保留字段长度为 2 个八位位组，默认值为 0。

——长度字段长度为 2 个八位位组，其值表示 WAI 协议分组所有字段的八位位组数。

——分组序号字段长度为 2 个八位位组，其值表示协议分组序号。第一个分组序号为 1，后序分组依次按 1 递增。

——分片序号字段长度为 1 个八位位组，其值表示分片的顺序编号，每一个分组的第一个分片序号为 0，后序分片依次按 1 递增。

——标识字段长度为 1 个八位位组，比特 0 表示后续是否有分片，值为 0 表示没有，值为 1 表示有。比特 1 至比特 7 保留。

——数据字段的内容根据类型和子类型的值而定，它除了包含固定的内容，还可以包含可选的属性。

其中：分组序号字段、分片序号字段和标识字段仅在 ASUE 和 AE 之间的 WAI 协议分组中有效。

定义 WAI 协议分组的最大长度为 65535 个八位位组。

8.1.4.1.1　WAI 协议分组数据字段的固定内容

a) 标识 FLAG

长度为 1 个八位位组。格式如下：

B0	B1	B2	B3	B4	B5	B6	B7
BK 更新标识	预鉴别标识	证书验证请求标识	可选字段标识	USK 更新标识	STAKey 协商标识	STAKey 删除标识	保留

图 53　标识 FLAG

——BK 更新标识比特:1 表示 BK 更新分组;0 表示非 BK 更新分组。

——预鉴别标识比特:1 表示预鉴别分组;0 表示非预鉴别分组。

——证书验证请求标识比特:1 表示要求验证对方的证书;0 表示不需要验证。

——可选字段标识比特:1 表示分组中有可选字段;0 表示分组中没有可选字段。

——USK 更新标识比特:1 表示 USK 更新分组;0 表示非 USK 更新分组。

——STAKey 协商标识比特:1 表示 STAKey 协商分组;0 表示非 STAKey 协商分组。

——STAKey 删除标识比特:1 表示 STAKey 删除分组;0 表示非 STAKey 删除分组。

b) BKID

基密钥标识,长度为 16 个八位位组,其计算方法为,BKID= KD - HMAC-SHA256(BK,MAC_{AE} ‖ MAC_{ASUE})。其中“‖”为链接操作,对于 KD-HMAC-SHA256 算法而言,第一个参数表示密钥,第二个参数表示文本。

c) USKID

单播会话密钥索引,长度为 1 个八位位组。其中比特 0 有意义。

d) MSKID/STAKeyID

组播会话密钥索引/站间密钥索引。长度为 1 个八位位组,其中比特 0 有意义。

e) 结果

长度为 1 个八位位组,内容为 0(表示成功)或其他值(失败的原因)。

f) 一次性随机数

长度为 32 个八位位组。

g) 密钥数据

密钥数据字段由长度字段与内容字段组成。其中长度字段为 1 个八位位组,表示内容字段的八位位组数。

h) 密钥通告标识

长度为 16 个八位位组,表示一个整数。

i) 数据序列号

长度为 16 个八位位组,表示一个整数。

j) 证书

图 54 证书

其中:

——证书标识字段表示证书类型,长度为 2 个八位位组。证书标识定义如下:

1 表示该字段的证书数据为 X.509 v3 证书;

2 表示该字段的证书数据为 GBW 证书;

其他值保留。

——证书长度字段为证书数据字段的八位位组数。

——证书数据字段为该属性字段的内容。

k) 身份

图 55 身份

其中：

——身份标识字段表示身份类型，长度为 2 个八位位组。身份标识定义如下：

1 表示该字段的身份数据由 X.509 v3 证书的持有者名称、颁发者名称、序列号字段组成；

2 表示该字段的身份数据由 GBW 证书的持有者名称、颁发者名称、序列号字段组成；

其他值保留。

——身份长度字段长度为 2 个八位位组，标识身份数据字段的八位位组数。

——身份数据字段为从证书中提取出的持有者名称、颁发者名称、序列号字段：

图 56 身份数据

注：对于 X.509 v3 证书，身份数据采用 x.509 证书相应字段的编码方式，即 ASN.1/DER 编码；对于 GBW 证书，身份数据采用 GBW 证书中的相应字段编码方式。

1) ADDID

地址索引，长度为 12 个八位位组。

图 57 地址索引

8.1.4.1.2 WAI 协议分组数据字段的属性内容

属性采用类型-长度-值(TLV)的格式构成，格式如下：

图 58 属性格式

类型字段表示属性的类型，其长度为 1 个八位位组，类型值定义如下：

1 签名属性

2 证书验证结果

3 身份列表

其他值保留。

长度字段表示值字段的八位位组数，本字段长度为 2 个八位位组。

值字段表示属性的内容。

a) 签名属性

图 59 签名属性

其中身份为 8.1.4.1.1k)中定义。

签名算法包含长度和内容两个子字段。其长度字段为 2 个八位位组，表示内容字段的八位位组数。内容字段由 1 个八位位组的杂凑算法标识、1 个八位位组的签名算法标识和参数字段组成：

——杂凑算法标识定义如下：

1 表示 SHA-256 杂凑算法；

其他值保留。

——签名算法标识定义如下：

1 表示 192 位的椭圆曲线数字签名算法，即 ECDSA-192；

其他值保留。

——参数字段表示签名算法的参数，由参数标识、参数长度和参数内容组成。参数标识字段长度为 1 个八位位组；参数长度字段为 2 个八位位组，表示参数内容字段的八位位组数；当签名算法标识字段值为 1 时，参数字段的值定义如下：

- 参数标识为 1 时，标识参数以 OID 方式表示，参数长度字段表示 OID 标识的八位位组数，参数内容为 OID 编码，本规范采用的 ECC 参数的 OID 为 1.2.156.11235.1.1.2.1，OID 编码采用 ASN.1/DER。
- 参数标识其他值保留。

——签名值字段包含长度和内容，长度子字段为 2 个八位位组，表示内容子字段的八位位组数。内容子字段为签名的值，是按照《无线局域网产品密码算法应用指南》中的规则将签名结果转化成的八位位组串。

b) 证书验证结果

类型 (1)	长度 (2)
一次性随机数 1 (32)	
一次性随机数 2 (32)	
验证结果 (1)	证书（可变）
验证结果 (1)	证书（可变）

注：括号内单位为八位位组数。

图 60 证书验证结果

c) 身份列表

类型 (1)	长度 (2)
保留 (1)	身份个数 (2)
身份 1 (可变)	
身份 2 (可变)	
……	

注：括号内单位为八位位组数。

图 61 身份列表

其中身份 1、身份 2……为 8.1.4.1.1 中 k)定义的身份。

8.1.4.2 证书鉴别过程

证书鉴别过程是基于 STA/AP 的证书进行鉴别及密钥协商，并建立 BKSA。其过程如下图所示。

图 61a 证书鉴别过程

8.1.4.2.1 鉴别激活分组

鉴别激活分组数据字段的格式如下：

标识 FLAG	鉴别标识	本地 ASU 的身份	STA_{AE}的证书	ECDH 参数
1	32	可变	可变	可变

八位位组数：

图 61b 鉴别激活分组数据字段格式

其中：

——标识字段长度为 1 个八位位组，定义如前，比特 0、1 有意义。当 STA 关联或重新关联至 AP 时进行证书鉴别过程，比特 0（BK 更新标识）的值为 0；当证书鉴别过程进行 BK 更新时，比特 0（BK 更新标识）的值为 1。如果不是预鉴别过程，比特 1（预鉴别标识）的值为 0；如果是预鉴别过程，比特 1（预鉴别标识）的值为 1。

——鉴别标识字段长度为 32 个八位位组。若标识字段比特 0（BK 更新标识）的值为 0，则鉴别标识字段的值由 AE 采用随机数生成算法生成；若标识字段比特 0（BK 更新标识）的值为 1，则鉴别标识字段的值为上一次证书鉴别过程所协商的鉴别标识。

——本地 ASU 的身份字段标识 AE 信任的 ASU，采用 8.1.4.1.1 中的定义。

——STA_{AE}（作为 AE 实体的站）的证书字段表示作为 AE 实体的站的证书，采用 8.1.4.1.1 的定义。

——ECDH 参数字段由参数标识和参数长度和参数内容组成，参数标识字段长度为 1 个八位位组参数长度字段为 2 个八位位组，表示参数内容字段的八位位组数。参数字段的值定义如下：

- 参数标识为 1 时，参数内容以 OID 方式表示，参数长度字段表示 OID 标识的八位位组数，参数内容为 OID 编码。本规范采用值为 1.2.156.11235.1.1.2.1 的 OID 表示国家密码管理局批准的 ECC 域参数，OID 编码采用 ASN.1/DER。
- 参数标识其他值保留。

当 STA 关联或重新关联至 AP/STA，ASUE 和 AE 选择采用证书鉴别及密钥管理方法，或 AE 的本地策略要求重新进行证书鉴别过程，或 AE 收到 ASUE 的预鉴别开始分组时，AE 向 ASUE 发送鉴别激活分组激活 ASUE 进行双向证书鉴别。

ASUE 接收到由 AE 发送的鉴别激活分组后，进行如下处理：

a) ASUE 检查鉴别激活分组中标识字段的比特 0（BK 更新标识）的值，当值为 1 时执行 b)操作；当值为 0 时执行 c)操作；

b) ASUE 检查鉴别激活分组中鉴别标识字段与上一次证书鉴别过程中保存的鉴别标识是否一致，若不一致，则丢弃该鉴别激活分组；否则执行 c)操作；

c) ASUE 根据鉴别激活分组中的 AE 信任的 ASU 身份选择由该 ASU 颁发的证书或者根据本地策略选择证书，产生用于 ECDH 交换的临时私钥 x、临时公钥 x·P 和 32 个八位位组的 ASUE 询问 N_{ASUE}（随机数），生成接入鉴别请求分组，发送给 AE。

8.1.4.2.2 接入鉴别请求分组

由 ASUE 发往 AE，接入鉴别请求分组数据字段的格式如下：

标识 FLAG	鉴别标识	ASUE 询问	ASUE 密钥数据	STA_{AE} 的身份	STA_{ASUE} 证书	ECDH 参数	ASUE 信任的 ASU 列表	ASUE 的签名
1	32	32	可变	可变	可变	可变	可变	可变

八位位组数：

图 61c 接入鉴别请求分组数据字段格式

其中：

——标识字段长度为 1 个八位位组，定义如前，比特 0、1、2、3 有意义。本字段除比特 2（证书验证请求

标识)、比特3(可选字段标识)以外的其他比特,应与AE发送的鉴别激活分组中标识字段对应比特相同。比特2(证书验证请求标识)为1表示ASUE要求验证AE证书的有效性,为0表示不需要验证AE证书的有效性。当在比特0(BK更新标识)为0时,比特2必须为1,即不是进行BK更新时,必须验证AE证书的有效性。比特3(可选字段标识)为1表示分组中有可选字段,为0表示没有。

——鉴别标识字段长度为32个八位位组。本字段值应与AE发送的鉴别激活分组中鉴别标识字段值相同。

——ASUE询问字段长度为32个八位位组,由ASUE采用随机数生成算法生成,记作N_{ASUE}。

——ECDH参数字段,和鉴别激活分组中的ECDH参数字段相同;当ASUE发起BK更新时,该字段和初次证书鉴别过程的鉴别激活分组中的ECDH参数字段相同。

——ASUE密钥数据格式如前定义,内容是ASUE生成的用于ECDH交换的临时公钥。

——STA_{AE}的身份字段,采用8.1.4.1.1中的定义。

——STA_{ASUE}(作为ASUE实体的站)的证书字段表示作为ASUE实体的站的证书,采用8.1.4.1.1的定义。

——STA_{ASUE}信任的服务器列表字段,该字段为可选字段,采用身份列表属性表示,其定义如前。内容包含STA_{ASUE}信任的服务器,但不包含STA_{ASUE}的证书颁发者。若ASUE除了信任他的证书颁发者以外,还信任其他的某些实体,可以通过该字段通知鉴别服务器。

——ASUE的签名字段采用签名属性表示,其定义如前,它是对本分组中除本字段之外所有数据字段的签名。

当ASUE收到AE的鉴别激活分组或ASUE需要进行BK更新时,ASUE发送接入鉴别请求分组给AE。AE收到ASUE发来的接入鉴别请求分组后,进行如下处理:

a) 如果AE没有发送鉴别激活分组,则检查鉴别标识字段值和上一次证书鉴别过程中保存的鉴别标识是否相同,若相同,执行b)操作;否则丢弃该分组。如果AE发送了鉴别激活分组,则比较鉴别标识字段值及标识字段的比特0、比特1与AE发送的鉴别激活分组中相应字段的值是否相同,若不同,则丢弃该分组;否则,执行b)操作。

b) 检查STA_{AE}的身份字段是否与自己的身份一致,以及ECDH参数字段是否与自己在鉴别激活分组中的ECDH参数一致,若不一致,则丢弃该分组;否则验证ASUE签名,若验证不通过,则丢弃该分组;若标识字段的比特2为1或AE的本地策略要求使用ASU鉴别STA_{ASUE}的证书,则AE生成证书鉴别请求分组,发往ASU;否则执行c)操作。

c) AE本地鉴别STA_{ASUE}的证书,即根据本地缓存的STA_{ASUE}证书的验证结果及其根据本地策略所定义的时效性确定STA_{ASUE}证书的验证结果。若该证书鉴别结果成功,本地生成用于ECDH交换的临时私钥y、临时公钥y·P和32个八位位组的AE询问N_{AE}(随机数),使用自己的临时私钥y和接入鉴别请求分组中ASUE的临时公钥x·P进行ECDH计算,得到主密钥种子$(x \cdot y \cdot P)_{abscissa}$,对其进行扩展KD-HMAC-SHA256 ($(x \cdot y \cdot P)_{abscissa}$, $N_{AE} \| N_{ASUE} \|$ "base key expansion for key and additional nonce"),生成长度为16个八位位组的基密钥BK和长度为32个八位位组的下一次证书鉴别过程的鉴别标识种子,然后对该鉴别标识种子进行SHA-256运算,得到长度为32个八位位组的下一次证书鉴别过程的鉴别标识。然后设定接入结果为成功,同时标识字段的比特3(可选字段标识)置为0,表示没有可选字段,构造没有可选字段的接入鉴别响应分组发送给ASUE。若ASUE证书鉴别结果不成功,AE设定接入结果为相应内容,同时标识字段的比特3(可选字段标识)置为0,表示没有可选字段。AE的询问N_{AE}和AE的密钥数据(AE的临时公钥)可设置任意值,构造没有可选字段的接入鉴别响应分组发送给ASUE,然后解除与STA_{ASUE}的链路验证。

8.1.4.2.3 证书鉴别请求分组

由AE发往ASU,证书鉴别请求分组数据字段定义如下:

ADDID	AE 询问	ASUE 询问	STA_{ASUE} 的证书	STA_{AE} 的证书	ASUE 信任的 ASU 列表
12	32	32	可变	可变	可变

八位位组数：

图 61d　证书鉴别请求分组数据字段

其中：

——ADDID 字段长度为 12 个八位位组，由 $MAC_{AE} \| MAC_{ASUE}$ 组成。

——AE 询问字段长度为 32 个八位位组。由 AE 采用随机数生成算法生成。

——ASUE 询问字段长度为 32 个八位位组。本字段值应与 ASUE 发送的接入鉴别请求分组中 ASUE 询问字段值相同。

——STA_{ASUE} 的证书字段，定义如前。该字段和接入鉴别请求分组中 STA_{ASUE} 的证书字段相同。

——STA_{AE} 的证书字段，定义如前。内容包含 STA_{AE} 的证书。

——ASUE 信任的服务器列表字段，该字段为可选字段，采用身份列表属性表示，其定义如前。本字段值应与 ASUE 发送的接入鉴别请求分组中的 ASUE 信任的服务器列表字段相同。

若接入鉴别请求分组中的标识 FLAG 指示要进行证书验证或 AE 自己需要进行证书验证，AE 向 ASU 发送证书鉴别请求分组。

AE 接收到 ASUE 发送的接入鉴别请求分组并向 ASU 发送证书鉴别请求分组后，在证书鉴别请求分组超时时间内不对 ASUE 发送的接入鉴别请求进行处理。

ASU 收到证书鉴别请求分组后，进行如下处理：

a) ASU 参照 RFC3280 验证 STA_{AE} 证书和 STA_{ASUE} 证书，若无法验证，则将相应证书的验证结果置为证书的颁发者不明确再执行 b)操作；否则验证 STA_{AE} 证书和 STA_{ASUE} 证书的状态，执行 b)操作。

b) 根据 STA_{AE} 证书和 STA_{ASUE} 证书的验证结果，构造证书鉴别响应分组，并且附加相应的签名，发往 AE。

8.1.4.2.4　证书鉴别响应分组

由 ASU 发往 AE，证书鉴别响应分组数据字段的格式如下：

ADDID	证书的验证结果	ASUE 信任的服务器签名	AE 信任的服务器签名
12	可变	可变	可变

八位位组数：

图 61e　证书鉴别响应分组数据字段格式

其中：

——ADDID 字段长度为 12 个八位位组。该字段值和证书鉴别请求分组中的 ADDID 字段的值相同。

——证书的验证结果字段采用证书验证结果属性表示，其格式定义如前。字段中的第一个一次性随机数值和证书鉴别请求分组中的 AE 询问值相同，第二个一次性随机数值和证书鉴别请求分组中的 ASUE 询问值相同。字段中的第一个证书及结果对应于证书鉴别请求分组中的 STA_{ASUE} 证书，第二个证书及结果对应于证书鉴别请求分组中的 STA_{AE} 证书。证书结果定义如下：

0 表示证书有效；

1 表示证书的颁发者不明确；

2 表示证书基于不可信任的根证书；

3 表示证书未到生效期或已过期；

4 表示签名错误；

5 表示证书已吊销；

6 表示证书未按规定用途使用；

7 表示证书吊销状态未知；

8 表示证书错误原因未知；

其他值保留。

——ASUE 信任的服务器签名字段采用签名属性表示，其定义如前。它对本分组中证书的验证结果字段的签名。

——AE 信任的服务器签名字段采用签名属性表示，其定义如前。它对本分组中除本字段和 ADDID 字段之外所有数据字段的签名。

注：若 ASUE 信任的服务器和 AE 信任的服务器为同一个，即 ASUE 信任的服务器和 AE 信任的服务器的身份属性相同，则证书鉴别响应分组中 ASUE 信任的服务器签名字段和 AE 信任的服务器签名字段只存在一个；若 ASUE 证书的验证结果为证书的颁发者不明确，则证书鉴别响应分组不包含 ASUE 信任的服务器签名字段。

ASU 收到证书鉴别请求分组后，向 AE 发送证书鉴别响应分组。

AE 收到证书鉴别响应分组后，进行如下处理：

a) 根据 ADDID 确定对应的证书鉴别请求分组，检查证书的验证结果字段中的第一个一次性随机数值与自己在证书鉴别请求分组中的 AE 的询问是否相同，若相同，则执行 b)操作；否则，丢弃该证书鉴别响应分组；

b) AE 查找自身所信任的 ASU 的签名，验证其签名，若不正确，则丢弃该证书鉴别响应分组；否则执行 c)操作。

c) 若 ASUE 证书鉴别结果成功，本地生成用于 ECDH 交换的临时私钥 y 和临时公钥 y · P，使用自己的临时私钥 y 和 ASUE 的临时公钥 x · P 进行 ECDH 计算，得到密钥种子$(x \cdot y \cdot P)_{abscissa}$，对其进行扩展 KD-HMAC-SHA256 ($(x \cdot y \cdot P)_{abscissa}$, $N_{AE} \| N_{ASUE} \|$ "base key expansion for key and additional nonce")，生成长度为 16 个八位位组的基密钥 BK 和长度为 32 个八位位组的下一次证书鉴别过程的鉴别标识种子，然后对该鉴别标识种子进行 SHA-256 运算，得到长度为 32 个八位位组的下一次证书鉴别过程的鉴别标识并保存。然后设定接入结果为成功，构造接入鉴别响应分组发送给 ASUE。若接入鉴别请求分组中 ASUE 要求验证 AE 证书，则接入鉴别响应分组中标识字段的比特 3(可选字段标识)置为 1，表示有可选字段；否则置为 0，表示没有可选字段。

若 ASUE 证书鉴别结果不成功，AE 设定接入结果为不成功，AE 的询问 N_{AE} 和 AE 的密钥数据(AE 的临时公钥)可设置任意值。构造接入鉴别响应分组发送给 ASUE，然后解除与 STA_{ASUE} 的链路验证。若接入鉴别请求分组中 ASUE 要求验证 AE 证书，则接入鉴别响应分组中标识字段的比特 3(可选字段标识)置为 1，表示有可选字段；否则置为 0，表示没有可选字段。

8.1.4.2.5 接入鉴别响应分组

由 AE 发往 ASUE，接入鉴别响应分组数据字段的格式如下：

标识 FLAG	ASUE 询问	AE 询问	接入结果	ASUE 密钥数据	AE 密钥数据	STA_{AE} 的身份	STA_{ASUE} 身份	复合的证书验证结果	AE 的签名
1	32	32	1	可变	可变	可变	可变	可变	可变

八位位组数：

图 61f 接入鉴别响应分组数据字段格式

其中：

——标识字段长度为 1 个八位位组，定义如前，比特 0、1、3 有意义。本字段比特 0、比特 1 应与 ASUE 发送的鉴别接入请求分组中标识字段值相同。比特 3(可选字段标识)由 ASUE 根据上下文环境设置。比特 3(可选字段标识)为 1 表示分组中有可选字段，为 0 表示没有。

——ASUE 询问字段长度为 32 个八位位组。本字段值应与 ASUE 发送的鉴别接入请求分组中 ASUE 的询问字段值相同。

——AE 询问字段长度为 32 个八位位组。字段值应与 AE 发送的证书鉴别请求分组中 AE 的询问字段值相同。

——ASUE 密钥数据格式如前定义，内容是 ASUE 生成的用于 ECDH 交换的临时公钥，本字段值应与 ASUE 发送的鉴别接入请求分组中 ASUE 密钥数据字段值相同。

——AE 密钥数据格式如前定义，内容是 AE 生成的用于 ECDH 交换的临时公钥。

——STA_{AE}的身份字段，定义如前。

——STA_{ASUE}的身份字段，其定义如前。

——接入结果字段的长度为 1 个八位位组，其定义如前。具体意义如下：

0 表示接入成功，对应证书验证结果值为 0；

1 表示无法验证证书，对应证书验证结果值为 1；

2 表示证书错误，对应证书验证结果除 0 和 1 之外的其他值；

3 表示本地策略禁止；

其他值保留。

——复合的证书验证结果字段是可选的，若存在，则由证书鉴别响应分组中除 ADDID 外的其他各个字段组成，并且内容和它们相同。

——AE 的签名字段采用签名属性表示，其定义如前。它是对本分组中除本字段之外所有数据字段的签名。

AE 收到证书鉴别响应分组，或收到接入鉴别请求分组后，发送接入鉴别响应分组。

ASUE 收到接入鉴别响应分组后，进行如下处理：

a) 根据 STA_{AE}的身份和 STA_{ASUE}的身份判断是否为对应当前接入鉴别请求分组的接入鉴别响应分组，若不是，则丢弃该接入鉴别响应分组；否则，执行 b)操作。

b) 检查标识字段的比特 0、比特 1 与自己发送的接入鉴别请求分组中相应字段的值是否相同，若不同，则丢弃该分组；否则执行 c)操作。

c) 比较 ASUE 的询问与自己在接入鉴别请求分组中发送的 ASUE 询问是否相同、比较 ASUE 密钥数据与 ASUE 发送的鉴别接入请求分组中 ASUE 密钥数据是否相同，若不同，则丢弃该接入鉴别响应分组；否则，执行 d)操作。

d) 验证 AE 的签名是否正确，若不正确，则丢弃该接入鉴别响应分组；否则若该接入鉴别响应分组中的接入结果为不成功，则解除与该 STA_{AE}的链路验证；否则执行 e)操作。

e) 若 ASUE 在接入鉴别请求分组中不要求进行证书验证，执行 f)操作；否则 ASUE 在复合的证书鉴别结果中查找自身所信任的鉴别服务器的签名，验证 ASU 签名，若不正确，则丢弃该接入鉴别响应分组；否则检查 AE 证书的鉴别结果是否为有效，若无效，解除与该 STA_{AE}的链路验证；若有效，则执行 f)操作。

f) ASUE 使用自己的临时私钥 x 和 AE 的临时公钥 y·P 进行 ECDH 计算，得到密钥种子$(x \cdot y \cdot P)_{abscissa}$，对其进行扩展 KD-HMAC-SHA256 ($(x \cdot y \cdot P)_{abscissa}$, $N_{AE} \| N_{ASUE} \|$ "base key expansion for key and additional nonce")，生成长度为 16 个八位位组的基密钥 BK 和长度为 32 个八位位组的下一次证书鉴别过程的鉴别标识种子，然后对该鉴别标识种子进行 SHA-256 运算，得到长度为 32 个八位位组的下一次证书鉴别过程的鉴别标识。

在证书鉴别过程中，要进行 ECDH 协商出基密钥。对于 ECDH 算法，做以下说明：

1) 临时私钥 x、y 是在[1..n−1]间的整数，n 是椭圆曲线域参数中基点 P 的阶。

2) 临时公钥 x·P、y·P 是椭圆曲线域参数定义的椭圆曲线上的点。

3) ECDH 协商出来密钥种子$(x \cdot y \cdot P)_{abscissa}$是 x·y·P 的 x 坐标，x·y·P 不能是无穷远点。

8.1.4.3 单播密钥协商过程

单播密钥协商过程使用基密钥完成单播会话密钥的协商，建立 USKSA。

图 61g 单播密钥协商过程

8.1.4.3.1 单播密钥协商请求分组

单播密钥协商请求分组数据字段格式如下：

标识 FLAG	BKID	USKID	ADDID	AE 询问
1	16	1	12	32

八位位组数：

图 61h 单播密钥协商请求分组数据字段格式

其中：

——标识 FLAG 字段长度为 1 个八位位组，定义如前，比特 4(USK 更新标识)有意义。当 AE 进行会话密钥更新时，比特 4 (USK 更新标识)的值为 1；否则为 0。

——BKID 字段长度为 16 个八位位组，表示当前作为共享密钥的基密钥，其值计算如前定义。

——USKID 字段长度为 1 个八位位组，其中比特 0 标识当前协商的单播会话密钥，其他位保留。本字段的比特 0 在 BKSA 建立后第一次单播密钥协商时初值为 0，以后再重新进行单播密钥协商时该位在 0 和 1 之间翻转。

——ADDID 字段长度为 12 个八位位组，如前定义。MAC 地址 1 为 AE 的 MAC 地址，其中 MAC 地址 2 为 ASUE 的 MAC 地址。

——AE 询问字段长度为 32 个八位位组，记作 N_1。若标识字段的比特 4 (USK 更新标识)的值为 0，则 AE 的询问为 AE 产生的随机数；若标识字段的比特 4 (USK 更新标识)的值为 1，则 AE 询问为上一次单播密钥协商过程所协商的值。

在 AE 完成证书鉴别过程并建立有效的 BKSA 后，或采用预共享密钥鉴别方式时，或缓存的 BKSA 被使用，或进行单播密钥更新时，AE 向 ASUE 发送单播密钥协商请求分组开始与 ASUE 进行单播密钥协商。

ASUE 接收到与其相关联的 AE 发送的单播密钥协商请求分组后，进行如下处理：

a) 首先检查 BKID 所指 BKSA 是否有效，若无效，则丢弃该分组；否则检查标识字段的比特 4 (USK 更新标识)是否为 0，若是 0，执行 c)操作；若是 1，则检查 USKID 所指的 USKSA 是否有效，若有效，则丢弃该分组；否则，执行 b)操作。

b) 检查 AE 询问与本地保存的值是否相同，若不同，则丢弃该分组；否则，执行 c)操作。

c) ASUE 利用随机数产生器产生 ASUE 询问 N_2，然后计算 KD-HMAC-SHA256 (BK ，ADDID ‖ N_1 ‖ N_2 ‖ “pairwise key expansion for unicast and additional keys and nonce”)，其中 N_1 为 AE 询问，N_2 为 ASUE 询问。生成 96 个八位位组，前 64 个八位位组为单播会话密钥(第一个 16 个八位位组为单播加密密钥 UEK，第二个 16 个八位位组为单播完整性校验密钥 UCK，第三个 16 个八位位组为消息鉴别密钥 MAK，第四个 16 个八位位组为密钥加密密钥 KEK)。最后 32 个八位位组为下一次单播会话密钥协商过程的 AE 询问的种子，然后对该种子使用 SHA-256 函数计算得到长度为 32 个八位位组的下一次单播密钥协商过程的 AE 询问并保存。

d) 用消息鉴别密钥 MAK 通过 HMAC-SHA256 算法本地计算消息鉴别码，构造单播密钥协商响

应分组发往 AE。

e) 若为 BSS 模式，则 ASUE 利用原语 MLME-SETWPIKEYS. request 安装新协商的单播会话密钥；若为 IBSS 模式，只有当 AE 的 MAC 地址大于 ASUE 的 MAC 地址时，ASUE 才利用原语 MLME-SETWPIKEYS. request 安装新协商的单播会话密钥。对于新安装的密钥，ASUE 调用原语 MLME-SETPROTECTION. request 仅启用其接收功能，即允许用其解密 AE 发来的单播数据。

8.1.4.3.2 单播密钥协商响应分组

单播密钥协商响应分组数据字段格式如下：

标识 FLAG	BKID	USKID	ADDID	ASUE 询问	AE 询问	WIE_{ASUE}	消息鉴别码
1	16	1	12	32	32	可变	20

八位位组数：

图 61i 单播密钥协商响应分组数据字段格式

其中：

——标识 FLAG 字段长度为 1 个八位位组，定义如前，比特 4(USK 更新标识)有意义。当 AE 进行会话密钥更新时，比特 4 (USK 更新标识)的值为 1；否则为 0。

——BKID 字段长度为 16 个八位位组，表示当前共享的基密钥，其值计算如前定义。

——USKID 字段长度为 1 个八位位组，其中比特 0 标识当前协商的单播会话密钥，其他位保留。本字段的比特 0 初始值为 0，每次重新进行单播密钥协商时该位在 0 和 1 之间翻转。

——ADDID 字段长度为 12 个八位位组，如前定义。MAC 地址 1 为 AE 的 MAC 地址，其中 MAC 地址 2 为 ASUE 的 MAC 地址。

——AE 询问字段长度为 32 个八位位组。若 ASUE 发起密钥更新，ASUE 设置标识字段的比特 4 (USK 更新标识)的值为 1，AE 询问字段为上一次单播密钥协商过程所协商的值；否则该字段和单播密钥协商请求分组中的 AE 询问字段相同。

——ASUE 询问字段长度为 32 个八位位组。由 ASUE 利用随机数产生器生成。

——WIE_{ASUE}字段为 ASUE 选择的 WAPI 信息元素。在 BSS 模式下，该字段和 ASUE 在关联请求帧中发送的 WAPI 信息元素相同；在 IBSS 模式下，该字段包含 ASUE 选择的单播密码算法、AE 通告的组播密码算法和当前使用的鉴别和密钥管理套件列表。

——消息鉴别码字段长度为 20 个八位位组。其值为 ASUE 利用最新协商的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法对本字段之前的所有协议数据字段内容计算得到，不包含分组头

ASUE 进行密钥更新时，或收到 AE 的单播密钥协商请求分组并构造单播密钥协商响应后，发送单播密钥协商响应分组给 AE。

AE 收到单播密钥协商响应分组后，进行如下处理：

a) 若标识字段的比特 4(USK 更新标识)为 1，执行 b)操作；否则执行 c)操作。

b) 若当前有有效的 USKSA 并且 USKID 所指 USKSA 无效，则执行 c)操作；否则，丢弃该分组。

c) 检查 AE 询问值是否正确，若不正确，则丢弃该分组；否则，执行 d)操作。

d) 计算 KD-HMAC-SHA256(BK, ADDID ‖ N_1 ‖ N_2 ‖ "pairwise key expansion for unicast and additional keys and nonce")，其中 N_1 是 AE 询问，N_2 是 ASUE 询问。生成 96 个八位位组，前 64 个八位位组为单播会话密钥(第一个 16 个八位位组为单播加密密钥 UEK，第二个 16 个八位位组为单播完整性校验密钥 UCK，第三个 16 个八位位组为消息鉴别密钥 MAK，第四个 16 个八位位组为密钥加密密钥 KEK)。后 32 个八位位组为下一次单播会话密钥协商过程的 AE 询问的种子，然后对种子使用 SHA-256 函数计算得到长度为 32 个八位位组的下一次单播会话密钥协商过程的 AE 询问。利用消息鉴别密钥 MAK 通过 HMAC-SHA256 算法本地计算消息鉴别码，与分组中的消息鉴别码字段值比较，若相同，则执行操作 e)；否则，丢弃该分组。

e) 若标识字段的比特4(USK更新标识)为0,在基础模式下,则检查WIE_{ASUE}字段和自己收到的关联请求帧的WAPI信息元素是否相同,若不同,解除与STA_{ASUE}的链路验证;若相同,则执行操作f);在IBSS模式下,检查WIE_{ASUE}字段中选择的单播密钥算法是被支持的,然后执行操作f);否则解除与ASUE的链路验证;若标识字段的比特4(USK更新标识)为1,则执行操作f)。

f) 用消息鉴别密钥MAK通过HMAC-SHA256算法本地计算消息鉴别码,构造单播密钥协商确认分组,发送给ASUE。

g) 若为BSS模式,则AE利用原语MLME-SETWPIKEYS. request安装新协商的单播会话密钥;若为IBSS模式,只有当AE的MAC地址大于ASUE的MAC地址时,AE才利用原语MLME-SETWPIKEYS. request安装新协商的单播会话密钥。对于新安装的密钥,AE调用原语MLME-SETPROTECTION. request启用其收发功能,即可利用其对单播数据进行加解密。若此次单播密钥协商过程为更新过程,则一旦使用新密钥正确解密过数据时,删除旧的单播会话密钥;或者启用新密钥收发数据60秒之后,自动删除旧的单播密钥。

8.1.4.3.3 单播密钥协商确认分组

单播密钥协商确认分组数据字段格式如下:

标识FLAG	BKID	USKID	ADDID	ASUE询问	WIE_{AE}	消息鉴别码
1	16	1	12	32	可变	20

八位位组数:

图61j 单播密钥协商确认分组数据字段格式

其中:

——标识FLAG字段长度为1个八位位组,定义如前,本字段和单播密钥协商响应分组的标识字段相同。

——BKID字段长度为16个八位位组,本字段和单播密钥协商响应分组的BKID字段相同。

——USKID字段长度为1个八位位组,其中比特0标识当前协商的单播密钥,其他位保留。本字段和单播密钥协商响应分组的USKID字段相同。

——ADDID字段长度为12个八位位组,本字段和单播密钥协商响应分组的ADDID字段相同。

——ASUE询问字段长度为32个八位位组。本字段和单播密钥协商响应分组中的ASUE询问字段相同。

——WIE_{AE}字段为AE在信标帧和探询响应帧中发送的WAPI信息元素。

——消息鉴别码字段长度为20个八位位组。其值为AE利用最新协商的消息鉴别密钥MAK通过HMAC-SHA256算法对本字段之前的所有协议数据字段内容计算得到,不包含分组头。

AE收到单播密钥响应分组后,发送单播密钥协商确认分组给ASUE。

ASUE接收到AE的单播密钥协商确认分组后,进行如下处理:

a)检查ASUE询问与自己在单播密钥协商响应分组中发送的值是否相同,若不同,则丢弃该分组;否则,执行b)操作。

b)利用消息鉴别密钥MAK通过HMAC-SHA256算法本地计算消息鉴别码,与分组中的消息鉴别码字段值比较,若相同,则执行操作c);否则,丢弃该分组。

c)若标识字段的比特4(USK更新标识)为0,则检查WIE_{AE}字段和自己收到的信标帧和探询响应帧的WAPI信息元素是否相同,若相同,则执行操作d);否则,解除与STA_{AE}的链路验证。若标识字段的比特4(USK更新标识)为1,则执行操作d)。

d) 调用原语MLME-SETPROTECTION. request启用新安装的单播会话密钥的发送功能,即允许利用该新密钥加密发送单播数据。若此次单播密钥协商过程为更新过程,则还需删除旧的单播会话密钥。

8.1.4.4 组播密钥/站间密钥通告过程

组播密钥/站间密钥通告过程使用单播密钥协商过程协商出来的密钥进行组播密钥/站间密钥通告，并建立 MSKSA 或 STAKeySA。

图 61k 组播密钥/站间密钥通告过程

8.1.4.4.1 组播密钥/站间密钥通告分组

单播密钥协商成功后，或 AE 要更新组播密钥时，或 AE 收到 STAKey 建立请求分组时，AE 向 ASUE 发送组播密钥/站间密钥通告分组通告组播/站间主密钥。

组播密钥/站间密钥通告分组数据字段格式如下：

标识 FLAG	MSKID/ STAKeyID	USKID	ADDID	数据序号	密钥通告标识	密钥数据	消息鉴别码
1	1	1	12	16	16	可变	20

八位位组数：

图 61l 组播密钥/站间密钥通告分组数据字段格式

其中：

——标识 FLAG 字段长度为 1 个八位位组，定义如前，比特 5、6 有意义。当 AE 通告的密钥是 STAKey 时，比特 5(STAKey 协商标识)为 1；否则为 0。比特 6(STAKey 密钥删除标识)为 1 表示删除由 USKID 和 ADDID 所确定的 STAKey，否则为 0。

——MSKID/STAKeyID 字段长度为 1 个八位位组，其中比特 0 标识当前通告的密钥，其他位保留。本字段中比特 0 初始值为 0，每次更新通告密钥时，该位在 0 和 1 之间翻转。

——USKID 字段长度为 1 个八位位组，其中比特 0 标识计算消息鉴别码字段值所用的消息鉴别密钥 MAK。

——ADDID 字段长度为 12 个八位位组。当通告的密钥是 STAKey 主密钥时，该字段的值为发起方的 MAC 地址 ‖ 对端的 MAC 地址；当通告的密钥为组播主密钥时，该字段的值为 MAC_{AE} ‖ MAC_{ASUE}。

——数据序号字段长度为 16 个八位位组，表示一个整数，用于标识已经使用当前通告的密钥加密发送的数据分组序号(WPI 组播数据分组中的 PN)，之后 STA 收到的数据帧序号应大于本字段值，否则丢弃。

——密钥通告标识字段长度为 16 个八位位组，表示一个整数，初始值为 0x5C365C365C365C365C365C365C365C36，在每次密钥更新通告时该字段值加 1。若通告的密钥不变，则本字段值保持不变。AE 端判断密钥通告标识字段值单调递增溢出后，与所有已关联的 STA 解除链路验证。该字段还用作密钥通告数据的 IV。

——密钥数据字段格式如前定义，其内容字段是 AE 利用密钥加密密钥 KEK 采用协商选择的单播密码算法对通告主密钥 NMK 加密(不带 MIC)后的密文，通告主密钥 NMK 为 AE 生成的 16 个八位位组的随机数，加密所用的 IV 为密钥通告标识字段。

——消息鉴别码字段长度为 20 个八位位组。其值为 AE 利用 USKID 字段标识的消息鉴别密钥

MAK 通过 HMAC-SHA256 算法对本字段之前的所有协议数据字段内容计算得到，不包含分组头。

ASUE 接收到 AE 发送的组播密钥/站间密钥通告分组后，进行如下处理：

a) 若标识字段的比特 5(STAKey 协商标识)为 1，则为站间密钥通告；否则为组播密钥通告。若 ASUE 不支持或不允许 STAKey 使用，则当标识字段的比特 5(STAKey 协商标识)为 1 时，丢弃该分组。ASUE 利用 USKID 字段标识的消息鉴别密钥 MAK 计算校验值，与消息鉴别码字段值进行比较，若不同，则丢弃该分组；否则执行 b)操作。

b) 检查密钥通告标识字段值是否单调递增，若为单调递增，则执行 c)操作；否则丢弃该分组。

c) 对密钥数据解密得到 16 个八位位组的通告主密钥 NMK，利用 KD-HMAC-SHA256 算法进行扩展，则生成长度为 32 个八位位组的会话密钥(其中前 16 个八位位组为加密密钥，后 16 个八位位组为完整性校验密钥)。若标识字段的比特 5(STAKey 协商标识)为 1，则上述 NMK 密钥为站间主密钥，会话密钥为站间会话密钥；否则为组播主密钥，会话密钥为组播会话密钥。

d) 保存密钥通告标识字段值，生成组播密钥/站间密钥响应分组，发送给 AE。

e) 安装或删除密钥。若此次为组播密钥通告过程，则利用原语 MLME-SETWPIKEYS. request 安装新的组播会话密钥，并调用原语 MLME-SETPROTECTION. request 启用其接收功能；若此次组播密钥通告过程为 BKSA 建立后的首次通告过程，则将受控端口的状态设置为 On；若此次组播密钥通告过程为更新过程，则一旦使用此新密钥正确解密过数据时，删除旧的组播密钥。若此次为站间密钥通告过程，对于站间密钥的发起方而言，利用原语 MLME-SETWPIKEYS. request 安装新的站间密钥，并调用原语 MLME-SETPROTECTION. request 启用其发送功能，若此次站间密钥通告过程为更新过程，则删除旧密钥；对于站间密钥的对端而言，安装新的站间密钥，并启用其接收功能，若此次站间密钥通告过程为更新过程，则一旦使用新密钥正确解密过数据时，删除旧的站间密钥。若此次为站间密钥删除过程，对于站间密钥的对端而言，删除新安装的站间密钥。

若 AE 通告了新的组播密钥，ASUE 保存组播密钥，在接收组播数据帧时根据 KeyID 字段选择组播解密密钥，当 ASUE 接收到 AE 用最新通告的组播密钥加密的组播数据帧，并且校验和解密均正确时，丢弃旧的组播密钥。

8.1.4.4.2 组播密钥/站间密钥响应分组

组播密钥/站间密钥响应分组数据字段格式如下：

标识 FLAG	MSKID/ STAKeyID	USKID	ADDID	密钥通告标识	消息鉴别码
1	1	1	12	16	20

八位位组数：

图 61m 组播密钥/站间密钥响应分组数据字段格式

其中：

——标识字段长度为 1 个八位位组，定义如前，此字段值应与 AE 发送的组播密钥/站间密钥通告分组中的标识字段值相同。

——MSKID/ STAKeyID 字段长度为 1 个八位位组，值为 AE 发送的组播密钥/站间密钥通告分组中的 MSKID/ STAKeyID 字段值。

——USKID 字段长度为 1 个八位位组，其中比特 0 标识计算消息鉴别码字段值所用的消息鉴别密钥 MAK。此字段值应与 AE 发送的组播密钥/站间密钥通告分组中的 USKID 字段值相同。

——ADDID 字段长度为 12 个八位位组，该字段的值和组播密钥/站间密钥通告分组中 ADDID 字段值相同。

——密钥通告标识字段长度为 16 个八位位组，表示一个整数，取值为 AE 发送的组播密钥/站间密钥通告分组中的密钥通告标识字段值。

——消息鉴别码字段长度为 20 个八位位组。其值为 ASUE 利用 USKID 字段标识的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法对本字段之前的所有协议数据字段内容计算得到，不包含分组头。

ASUE 向 AE 发送组播密钥/站间密钥响应分组，AE 接收到 ASUE 发送的组播密钥响应分组后，进行如下处理：

a) 利用 USKID 字段标识的消息鉴别密钥 MAK 计算校验值，与消息鉴别码字段值进行比较，若不同，则丢弃该分组；否则，执行 b)操作。

b) 比较标识、MSKID/STAKeyID 字段、USKID 字段、ADDID 字段和密钥通告标识字段与发送的组播密钥/站间密钥通告分组中的相应字段值，若均相同，则本次组播密钥/站间密钥通告成功；否则，丢弃该分组。

c) 通告成功后，若此次通告的密钥尚未安装，则利用原语 MLME-SETWPIKEYS. request 安装新密钥。若此次密钥通告过程为组播密钥通告，则 AE 调用原语 MLME-SETPROTECTION. request 启动新密钥的发送功能，即利用此密钥加密组播数据；若此次组播密钥通告过程为 BKSA 建立后的首次通告过程，则将受控端口的状态设置为 On；若此次组播密钥通告过程为更新过程且已通告给所有已关联的 ASUE，则删除旧密钥。若此次密钥通告过程为向站间密钥发起方的通告，则 AE 对于站间密钥的发起方至对端的单播数据直接转发，不需进行解密、加密。

AE 在更新组播密钥过程中，使用旧的组播密钥对组播数据帧进行加密发送，当对所有已关联到该 AP 的 STA 均组播密钥通告后，才启用最新通告的组播密钥用于组播数据帧的加密发送。

8.1.4.5 STAKey 建立过程

当 STA1 直接发送数据给同一个 BSS 下的 STA2(无需 AP 转发)时，STA1 可以要求和 STA2 建立 STAKey 用于加密从 STA1 到 STA2 的单播数据，加解密的算法采用 AP 通告的组播密码套件。

该过程采用组播密钥/站间密钥通告过程完成，其过程如下图所示。STA1 向 AP 发送 STAKey 建立请求，然后 AP 采用利用随机数生成算法生成 STAKey 主密钥，使用组播密钥/站间密钥通告过程把 STAKey 主密钥传递给 STA1 和 STA2。

若在 AP 把 STAKey 通告给 STA2 后，通告给 STA1 失败，AP 则通过组播密钥/站间密钥通告过程通知 STA2 删除相应的 STAKey，把组播密钥/站间密钥通告过程的组播密钥/站间密钥通告分组中标识 FLAG 字段的 STAKey 删除标识位设为 1。

若 STA1 要删除某个 STAKey，则 STA1 在 STAKey 建立请求分组中 FLAG 字段的 STA 密钥删除标识位设为 1。

STAKey 站间密钥建立请求分组数据字段格式如下：

标识 FLAG	STAKeyID	USKID	ADDID	重放计数器	消息鉴别码
1	1	1	12	16	20

八位位组数：

图 61n 站间密钥建立请求分组数据字段格式

其中：

——标识 FLAG 字段长度为 1 个八位位组，定义如前。比特 5、6 有意义，比特 5(STAKey 协商标识)为 1。当建立 STAkey 时，比特 6(STAKey 删除标识)为 0；删除 STAKey 时比特 6 为 1。

——STAKeyID 字段长度为 1 个八位位组，值为发起方选择的密钥索引。

——USKID 字段长度为 1 个八位位组，其中比特 0 标识计算消息鉴别码字段值所用的消息鉴别密钥 MAK。此字段值应与 AP 发送的组播密钥通告分组中的 USKID 字段值相同。

——ADDID 字段长度为 12 个八位位组，该字段的值为发起方的 MAC 地址 ‖ 对端接收方的 MAC 地址。

——重放计数器字段长度为 16 个八位位组，表示一个整数。初始值为 1，在每次密钥更新通告时

该字段值加 1。

——消息鉴别码字段长度为 20 个八位位组。其值为 STA 利用 USKID 字段标识的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法对本字段之前的所有协议数据字段内容计算得到，不包含分组头。

图 61o 站间密钥建立流程图

8.1.4.6 预鉴别

若目的 AP 支持预鉴别，STA 可以通过当前关联的 AP 和目的 AP 进行预鉴别。STA 通过传送一个预鉴别开始分组给目的 AP，请求开始预鉴别。当前关联的 AP 收到预鉴别开始分组，如果重放计数器和消息鉴别码有效，向目的 AP 转发预鉴别开始分组。目的 AP 收到该分组后，根据 ADDID 字段向相应的 STA 发出鉴别激活分组，开始 WAI 的证书鉴别过程。

预鉴别开始分组数据字段格式定义如下：

标识 FLAG	USKID	ADDID	重放计数器	消息鉴别码
1	1	12	16	20

八位位组数：

图 61p 预鉴别开始分组的数据字段格式

其中：

——标识 FLAG 字段长度为 1 个八位位组，定义如前。

——USKID 字段长度为 1 个八位位组，其中比特 0 标识计算消息鉴别码字段值所用的消息鉴别密钥 MAK。此字段值应与 AP 发送的组播密钥通告分组中的 USKID 字段值相同。

——ADDID 字段长度为 12 个八位位组，该字段的值为发起方 STA 的 MAC 地址 ‖ 目的 AP 的 MAC 地址。

——重放计数器字段长度为 16 个八位位组，表示一个整数。初始值为 1，每次发送预鉴别开始分组，该字段值加 1。该标识和 STAKey 建立请求分组中的重放计数器为同一个计数器。

——消息鉴别码字段长度为 20 个八位位组。其值为 STA 利用 USKID 字段标识的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法对本字段之前的所有协议数据字段内容计算得到，不包含分组头。

STA 在与当前关联的 AP 完成单播会话密钥的协商和安装之后才能和其他 AP 进行预鉴别过程。仅当目的 AP 在 WAPI 信息元素中声明的 WAPI 能力信息字段中标识支持预鉴别时，STA 才能与该 AP 进行预鉴别过程。只有采用证书鉴别和密钥管理方法时才可以启动预鉴别过程。

STA 在和当前关联的 AP 完成单播会话密钥的协商并安装密钥之后可启动预鉴别过程。STA 的鉴别单元发送 WAI 的“预鉴别开始”分组激活鉴别过程，该分组的 DA 为目的 AP 的 BSSID，RA 为当前关联的 AP 的 BSSID。目的 AP 应采用 BSSID 作为 AP 鉴别器单元的 MAC 地址。当前关联的 AP 应把 STA 发送的预鉴别分组桥接到 DS，并且应把来自 DS 的预鉴别分组发送到 STA。

通过 DS 接收到“预鉴别开始”分组的 AP 应与对应的 STA 开始鉴别过程。DS 将转发此鉴别消息到与 STA 当前关联的 AP。

如果 WAI 的证书鉴别过程成功，则预鉴别的结果为 BKSA。当 STA 与预鉴别过的 AP 进行关联后，可以利用 BKSA 进行单播密钥协商和组播密钥通告过程。如果 AP 和 STA 之间的预鉴别结果缓存的状态不同步，单播密钥协商过程将会失败，在这种情况下管理信息库 gb15629dot11wapiStatsWAIUnicastHandshakeFailures 的值应加 1。

无论目的 AP 是否在 STA 的信号范围之内，只要目的 AP 支持预鉴别，STA 的 ASUE 可以和目的 AP 进行预鉴别过程。

因为 AP 可能删除缓存的预鉴别结果 BKSA(可能由于资源受限超时等原因)，所以即使预鉴别过程已经成功，STA 和目的 AP 关联后，仍可能需要执行完成的 WAI 证书鉴别和密钥协商过程。

注：对于采用 0x88B4 协议传输的任意 WAI 分组(包括证书鉴别、单播密钥协商、组播密钥通告以及站间密钥建立等过程的所有分组)，若格式错误，则将 MIB 值 gb15629dot11wapiStatsWAIFormatErrors 加 1；若签名验证错误，则将 MIB 值 gb15629dot11wapiStatsWAISignatureErrors 加 1；若消息鉴别码校验错误，则将 MIB 值 gb15629dot11wapiStatsWAIHMACErrors 加 1。

8.1.4.7 缓存的 BKSA 和 WAPI 密钥管理

STA 可保留和 AP 之间的鉴别结果 BKSA，且 BKSA 在缓存时不能被修改。

如果 ESS 内 STA 确定有和将要关联的 AP 有效的 BKSA，STA 将在关联/重新关联请求中的 WAPI 信息元素中包含该 BKSA 的 BKID，当接收到带有一个或多个 BKID 的关联/重新关联请求时，AP 检查其是否缓存有该 BKID 的 BKSA 并且有效，如果确认该 BKSA 有效，AP 启动单播会话密钥协商和组播密钥通告过程；否则在关联后启动完整的鉴别和密钥协商过程。

如果 STA 和 AP 均声称有缓存的 BKSA，但单播会话密钥协商过程失败，AP 和 STA 应删除 BKID 对应的 BKSA。

如果 STA 切换到正在与之进行预鉴别过程但尚未协商出 BKSA 的 AP，应启动完整的 WAI 证书鉴别和密钥协商过程。

8.1.4.8 密钥更新

AE 可以发起基密钥、单播密钥和组播密钥的更新，ASUE 可以发起基密钥和单播密钥的更新。密钥更新时选择的密钥标识必须是当前无效的密钥标识，密钥更新后旧密钥的删除必须在新密钥激活后才能进行。

当基密钥更新后，单播密钥必须更新。从基密钥更新开始到单播密钥更新结束必须在 MIB 值 gb15629dot11wapiConfigSATimeout 的时间内完成，否则两个 STA 将解除链路验证。

单播密钥更新的过程也不进行 WAPI 信息元素的确认或单播密码算法的选择，采用初始单播密码协商过程所确定的单播密码算法。

8.1.4.9 超时处理

如果 STA 和 AP 发出消息后，在超时时间内没有收到响应，则将 MIB 值 gb15629dot11wapiStatsWAITimeoutCounters 加 1。对于证书鉴别过程的分组，则重传 gb15629dot11wapiConfigCertificateUpdateCount 次消息；对于单播密钥协商的分组，则重传 gb15629dot11wapiConfigUnicastUpdateCount 次消息；对于组播密钥/站间密钥通告的分组，则重传 gb15629dot11wapiConfigMulticastUpdateCount 次消息。证书鉴别请求分组的超时缺省为 10 秒，接入鉴别请求分组的超时缺省为 31 秒，其他分组的超时缺省为 1 秒。

在证书鉴别、单播密钥协商和组播密钥通告过程中如果重发规定次数后仍未收到响应，则 STA 和 AP 之间解除链路验证。若为证书鉴别过程，则 MIB 值 gb15629dot11wapiStatsWAICertificateHandshakeFailures 加 1；若为单播密钥协商过程，则 MIB 值 gb15629dot11wapiStatsWAIUnicastHandshakeFailures 加 1；若为组播密钥通告过程中，则 MIB 值

gb15629dot11wapiStatsWAIMulticastHandshakeFailures 加 1。

在站间密钥通告过程中如果重发规定次数后仍未收到响应，则 MIB 值 gb15629dot11wapiStatsWAI MulticastHandshakeFailures 加 1。

8.1.4.10 WPI-SMS4 密钥导出体系

在 WAI 鉴别及密钥管理系统中，使用 KD-HMAC-SHA256 函数从主密钥导出各密钥及询问。

8.1.4.10.1 BK 密钥导出体系

图 61q BK 密钥导出体系结构

8.1.4.10.2 单播密钥导出体系

图 61r 单播密钥导出体系结构

8.1.4.10.3 组播/站间密钥导出体系

图 61s 组播/站间密钥导出体系结构

8.1.4.10.4 预共享密钥导出体系

图 61t 预共享密钥导出体系结构

8.1.4.11 WAI 协议分组的分片与重组

当 STA 或 AP 有一份要发送的 WAI 协议分组时，它要判断向本地哪个接口发送分组，并查询该接口获得其最大传输单位 MTU(Maximum Transmission Unit)。将分组长度与 MTU 进行比较，若分组长度大于 MTU 时，发送端需要对此分组进行分片，每一片均具有自己的 WAI 协议首部。除最后一个分片外，其他分片的长度相等且为一个小于 MTU 的最大值，并建议该值为 8 个八位位组的整数倍。

当这些分片到达目的接收端时，需进行重新组装。WAI 协议分片首部中包含的信息足够使目的接收端能够正确组装这些分组分片。

WAI 协议层利用协议分组头中的三个字段进行分片与重新组装过程。对于发送端发送的每份 WAI 协议分组来说，其 WAI 分组序号字段包含一个惟一值(第一个 WAI 协议分组序号为 1，后续的 WAI 协议分组序号依次按 1 递增)，该值在协议分组分片时被复制到每个分片中。若为重传分组，则 WAI 分组序号保持不变，分片序号字段指的是该分片的顺序编号(第一个分片序号为 0，后续的分片序号依次按 1 递增)。分片标识字段用其中一个比特表示"更多的分片"，除了最后一个分片外，其他每个组成协议分组的分片均将该比特置 1。另外，当协议分组被分片后，每个分片的长度字段值表示该分片的实际长度，即 WAI 协议首部与分片数据的总长度(以八位位组为单位)。

分片与重新组装应由 WAI 协议层完成，WAI 协议执行过程中 MAC 层应工作在严格排序的模式下。不论 STA 还是 AP，发送端在协议层发送时按照 WAI 协议分组序号由小到大的顺序进行，当出现分片时，则同一 WAI 协议分组按照分片序号由小到大的顺序发送。而接收时，接收端协议层每接收完一个 WAI 协议分组的所有分片后，即可重新组装，恢复出发送端原始的协议分组，进行处理；若接收到的 WAI 协议分片出现乱序(或者 WAI 协议分组出现乱序)时，则应丢弃掉已收到的当前(或者前一个) WAI 协议分组的所有分片。

由于 WAI 协议(证书鉴别与密钥协商)设计时已经定义了每个协议分组的超时重传机制，因此不再需要定义每个分片的超时重传，即在一个 WAI 协议分组传输过程中，即使只丢失其中一个分片，也需要重传整个协议分组。

8.1.4.12 端口控制与数据传输

当启用 WAPI 时，ASUE 与 AE 的受控端口初始状态为 Off，只有单播密钥协商与组播密钥通告成功后，两者的受控端口状态被设置为 On，允许非 WAPI 的协议数据通过。

在进行 BKSA、USKSA 和 MSKSA 更新时，ASUE 与 AE 的受控端口处于 On 状态。

WAI 协议分组不论受控端口的状态如何，均以明文方式通过非受控端口传输。

8.2 WPI 保密基础结构

WPI 保密基础结构对 MAC 子层的 MPDU 进行加、解密处理，但对于 WAI 协议分组不进行加解密处理。WPI-SMS4 密码套件中采用的分组密码算法为 SMS4，下面详细说明 WPI-SMS4 密码套件的工作模式与封装结构。

8.2.1 WPI-SMS4 工作模式

在 WPI-SMS4 中，完整性校验算法工作在 CBC-MAC 模式，数据保密采用的对称加密算法工作在

OFB 模式。两种模式图示如下：

图 61u 工作模式

8.2.2 密钥

在证书鉴别过程中，ASUE 和 AE 首先通过 ECDH 交换并利用 KD-HMAC-SHA256 算法协商一个 16 个八位位组的基密钥 BK；在预共享密钥模式中，ASUE 和 AE 将所共享的密钥作为种子导出基密钥 BK。然后，在单播密钥协商过程中，ASUE 和 AE 分别交换一个随机数，利用 KD-HMAC-SHA256 算法和基密钥 BK 导出 96 个八位位组，前 64 个八位位组为单播会话密钥（第一个 16 个八位位组为单播加密密钥 UEK，第二个 16 个八位位组为单播完整性校验密钥 UCK，第三个 16 个八位位组为消息鉴别密钥 MAK，第四个 16 个八位位组为密钥加密密钥 KEK），后 32 个八位位组为下一次单播会话密钥协商过程的询问的种子，然后对种子使用杂凑函数 SHA-256 计算得到长度为 32 个八位位组的下一次单播密钥协商过程的询问并保存。最后，在组播密钥/站间密钥通告过程中，ASUE 和 AE 或者发起端和对端分别对 16 个八位位组的通告主密钥 NMK 通过 KD-HMAC-SHA256 算法进行扩展，生成长度为 32 个八位位组的组播/站间会话密钥（前 16 个八位位组为组播/站间加密密钥，后 16 个八位位组为组播/站间完整性校验密钥）。

注：在 BSS 和 IBSS 模式中，可以使用基于证书的密钥管理或基于预共享密钥的密钥管理。在 IBSS 中，发起端 STA 和对端 STA 分别作为 AE 和 ASUE，每一个 STA 都需要作为发起端完成单播密钥协商过程和组播密钥通告过程，选择 MAC 地址大的 STA 发起的协商过程的单播数据密钥作为数据传输使用的密钥。

MAC 地址编码为 6 个八位位组，当进行 MAC 地址比较时，把 MAC 地址作为一个无符号的二进制数并且是按照大数结尾排列的。

在用户输入共享主密钥时，需支持十六进制和 ASCII 码字符两种输入方式。

8.2.3 封装与解封装

WPI-SMS4 的 MPDU 封装结构如下：

图 61v　WPI-SMS4 的 MPDU 封装结构

其中：

——MAC 头字段，当地址 4 存在时，长度为 30 个八位位组；当地址 4 不存在时，长度为 24 个八位位组。当 MAC 头包含服务质量控制子字段时，长度再增加两个八位位组。

——KeyIdx 字段长度为 1 个八位位组，表示 USKID 或 MSKID 或 STAKeyID 值。

——保留字段长度为 1 个八位位组，默认值为 0。

——PN 字段长度为 16 个八位位组，表示一个整数，标识数据分组序号，该数据分组序号作为 OFB、CBC-MAC 模式下数据加密和校验时所需的 IV。数据分组序号 PN 字段按照小数结尾编码发送。

——PDU(数据)字段为 MPDU 数据，最大长度为 2278＝2312－18(WPI 头)－16(MIC)。

——MIC 字段长度为 16 个八位位组。

——FCS 字段长度为 4 个八位位组，为 MAC 帧格式的帧校验序列。

MIC 字段是利用完整性校验密钥采用 CBC-MAC 工作方式对完整性校验数据计算得到，下图为 MIC 计算时完整性校验数据的组成结构。

图 61w　完整性校验数据

其中，完整性校验数据包含两部分内容，叙述如下：

第一部分：

——帧控制字段(比特 4，5，6，11，12，13 置为 0，比特 14 置为 1)，2 个八位位组。

——地址 1，6 个八位位组。

——地址 2，6 个八位位组。

——地址 3，6 个八位位组。

——序列控制字段(比特 4～15 置为 0)，2 个八位位组。

——地址 4，6 个八位位组。若 MAC 帧头中不存在地址 4 时，则该字段的 6 个八位位组的值均置为 0。

——服务质量控制字段，2 个八位位组。若 MAC 帧头包含服务质量控制字段时，则该字段存在。

——KeyIdx 字段。1 个八位位组。

——保留字段。1 个八位位组。

——PDU 数据的长度 L。2 个八位位组。该字段按照大数结尾编码计算。

第二部分：

——PDU 数据。大于 0 个八位位组。

在 WPI-SMS4 中使用 CBC-MAC 模式计算完整性校验码 MIC 时，应保证完整性检验数据的长度为 16 个八位位组的整数倍。若完整性校验数据第一部分的长度不足 16 个八位位组的整数倍，应将第

一部分扩展为16个八位位组的最小整数倍，扩展采用第一部分后面补零的方法；若完整性校验数据第二部分的长度不足16个八位位组的整数倍，应将第二部分扩展为16个八位位组的最小整数倍，扩展采用第二部分后面补零的方法。接收方验证校验时采用相同的处理。

数据发送时，WPI-SMS4的MPDU封装过程如下：

a）利用完整性校验密钥与数据分组序号PN，通过工作在CBC-MAC模式的校验算法对完整性校验数据进行计算，得到完整性校验码MIC；

b）利用加密密钥与数据分组序号PN，通过工作在OFB模式的加密算法对MPDU数据 ‖ MIC进行加密，得到MPDU数据 ‖ MIC的密文；

c）封装后再组帧发送。

数据接收时，WPI-SMS4的MPDU解封装过程如下：

a）判断数据分组序号PN是否有效，若无效，则丢弃该数据，且将MIB值gb15629dot11wapiStatsWPIReplayCounters加1；

b）利用解密密钥与数据分组序号PN，通过工作在OFB模式的解密算法对分组中的MPDU数据 ‖ MIC密文进行解密，恢复出MPDU数据 ‖ MIC明文；若此时没有有效的解密密钥，则丢弃该数据，且将MIB值gb15629dot11wapiStatsWPIDecryptableErrors加1；

c）利用完整性校验密钥与数据分组序号PN，通过工作在CBC-MAC模式的校验算法对完整性校验数据进行本地计算，若计算得到的值与分组中的完整性校验码MIC不同，则丢弃该数据，且将MIB值gb15629dot11wapiStatsWPIMICErrors加1；

d）解封装后将MPDU明文进行重组处理。

8.2.4 数据分组序号PN的使用规则

a）在单播会话中

每次单播密钥更新后，ASUE初始化PN值为0x5C365C365C365C 365C365C365C365C36，AE初始化PN值为0x5C365C365C365C365C365C365C365C37，ASUE和AE每次发送单播数据帧时，先对PN值加2后再使用。

当ASUE接收单播数据帧时，判断帧中对应于单播密钥标识USKID的PN值是否严格单调递增且为奇数，若不是，则丢弃该分组。

当AE接收单播数据帧时，判断帧中对应于单播密钥标识USKID的PN值是否严格单调递增且为偶数，若不是，则丢弃该分组。

AE可根据时间或发送的数据包个数等策略更新单播密钥。此外，PN值的溢出问题需通过AE更新单播密钥来解决。

b）在组播会话中

每次组播密钥更新后，AE初始化PN值为0x5C365C365C365C365C365C365C365C36，AE每次发送组播数据帧时，先对PN值加1后再使用。

当ASUE接收到组播数据时，判断对应于组播密钥标识MSKID的PN值是否严格单调递增，若不是，则丢弃该分组。

AE可根据时间或发送的数据包个数等策略更新单播密钥。此外，PN值的溢出问题需通过AE更新单播密钥来解决。

c）在站间会话中

每次站间密钥建立后，站间密钥的发起方初始化PN值为0x5C365C365C365C365C365C365C 365C36，发起者每次发送至对端的单播数据帧时，先对PN值加1后再使用。

当对端接收到利用站间密钥加密的单播数据时，判断对应于该站间密钥标识STAKeyID的PN值是否严格单调递增，若不是，则丢弃该分组。

站间密钥的发起方可根据时间或发送的数据包个数等策略更新站间密钥。此外，PN值的溢出问

题需通过站间密钥的发起方更新站间密钥来解决。

27. 10.3.4 内容替换为：

该机制支持与对等 MAC 实体建立链路验证关系的过程。

10.3.4.1 MLME-AUTHENTICATE. request

10.3.4.1.1 功能

该原语请求与一个规定的对等 MAC 实体建立链路验证关系。

10.3.4.1.2 服务原语的语义

原语参数如下：

```
MLME- AUTHENTICATE. request    (
                                PeerSTAAddress,
                                AuthenticateFailureTimeout
                                )
```

名　　称	类　　型	有效范围	描　　述
PeerSTAAddress (对等的 STA 的地址)	MAC 地址	任一有效的单个 MAC 地址	规定与之建立链路验证过程的对等 MAC 实体的地址
AuthenticateFailureTimeout (链路验证失败超时)	整数型	≥1	规定终止链路验证规程的时间界限(单位为 TU)

10.3.4.1.3 产生条件

该原语由 SME 产生，STA 用于同一个特定的对等 MAC 实体建立链路验证关系，以便允许在两个 STA 之间交换第 2 类帧。在链路验证规程中，SME 可能产生额外的原语 MLME-AUTHENTICATE. request。

10.3.4.1.4 收后效果

该原语启动链路验证过程。随后 MLME 发布一个反映结果的原语 MLME- AUTHENTICATE. confirm。

10.3.4.2 MLME- AUTHENTICATE. confirm

10.3.4.2.1 功能

该原语报告与指定的对等 MAC 实体尝试建立链路验证关系的过程结果。

10.3.4.2.2 服务原语的语义

原语参数如下：

```
MLME- LINKVERIFY. confirm    (
                              PeerSTAAddress,
                              ResultCode
                              )
```

名　　称	类　　型	有效范围	描　　述
PeerSTAAddress (对等的 STA 的地址)	MAC 地址	任一有效的单个 MAC 地址	规定试图与之建立链路验证过程的对等 MAC 实体的地址。该值必须与相应的原语 MLME-AUTHENTICATE. request 中的参数 PeerSTAAddress 相匹配

名　　称	类　　型	有效范围	描　　述
ResultCode (返回值)	枚举型	SUCCESS INVALID_PARAMETERS TIMEOUT TOO_MANY_SIMULTANEOUS_REQUESTS REFUSED	指示原语 MLME- AUTHENTICATE. request 的结果

10.3.4.2.3 **产生条件**

该原语由 MLME 产生，是用于同规定的对等 MAC 实体建立链路验证的原语 MLME- AUTHENTICATE. request 的结果。

10.3.4.2.4 **收后效果**

通知 SME 链路验证规程的结果。

10.3.4.3 **MLME-AUTHENTICATE. indication**

10.3.4.3.1 **功能**

该原语报告与规定的对等 MAC 实体建立链路验证关系的情况。

10.3.4.3.2 **服务原语的语义**

原语参数如下：

```
MLME- AUTHENTICATE. indication    (
                                   PeerSTAAddress
                                   )
```

名　　称	类　　型	有效范围	描　　述
PeerSTAAddress (对等的 STA 的地址)	MAC 地址	任一有效的单个 MAC 地址	规定与之建立链路验证关系的对等 MAC 实体的地址

10.3.4.3.3 **产生条件**

该原语由 MLME 产生，作为与规定的对等 MAC 实体建立链路验证关系的结果。而该链路验证关系的建立源于由指定的对等的 MAC 实体启动的链路验证过程。

10.3.4.3.4 **收后效果**

通知 SME 链路验证关系的建立。

28. 10.3.5 内容替换为：

该机制支持使与一个对等的 MAC 实体之间的链路验证关系变为无效。

10.3.5.1 **MLME-DEAUTHENTICATE. request**

10.3.5.1.1 **功能**

该原语请求与一个对等的 MAC 实体之间的链路验证关系变为无效。

10.3.5.1.2 **服务原语的语义**

原语参数如下：

```
MLME- DEAUTHENTICATE. request    (
                                  PeerSTAAddress,
                                  ReasonCode
                                  )
```

名　　称	类　　型	有效范围	描　　述
PeerSTAAddress (对等的 STA 的地址)	任一有效的单个 MAC 地址	任一有效的单个 MAC 地址	规定与之解除链路验证关系的对等 MAC 实体的地址
ReasonCode (原因代码)	如帧格式所定义	如帧格式所定义	规定启动解除链路验证过程的原因

10.3.5.1.3　**产生条件**

该原语由 SME 产生，STA 用于使与一个规定的对等 MAC 实体之间的已建立的链路验证关系变为无效，以阻制在两个 STA 之间交换第 2 类帧。在解除链路验证过程中，SME 可能产生额外的原语 MLME-DEAUTHENTICATE. request。

10.3.5.1.4　**收后效果**

该原语启动一个解除链路验证过程。随后 MLME 发布一个反应结果的原语 MLME-DELINKVERIFY. confirm。

10.3.5.2　**MLME- DEAUTHENTICATE. confirm**

10.3.5.2.1　**功能**

该原语报告试图与指定的对等 MAC 实体解除链路验证关系的结果。

10.3.5.2.2　**服务原语的语义**

原语参数如下：

```
MLME-DEAUTHENTICATE. confirm    (
                                 PeerSTAAddress,
                                 ResultCode
                                 )
```

名　　称	类　　型	有效范围	描　　述
PeerSTAAddress (对等的 STA 的地址)	MAC 地址	任一有效的单个 MAC 地址	规定试图与之解除链路验证关系的对等 MAC 实体的地址
ResultCode (返回代码)	枚举	SUCCESS INVALID_PARAMETERS TOO_MANY_ SIMULTANEOUS_ REQUESTS	指示原语 MLME-DEAUTHENTICATE. request 的结果

10.3.5.2.3　**产生条件**

该原语由 MLME 产生，作为与规定的对等 MAC 实体解除链路验证关系的原语 MLME-DE-AUTHENTICATE. request 的结果。

10.3.5.2.4　**收后效果**

通知 SME 解除链路验证过程的结果。

10.3.5.3　**MLME- DEAUTHENTICATE. indication**

10.3.5.3.1　**功能**

该原语报告与一个规定的对等 MAC 实体之间的链路验证关系无效。

10.3.5.3.2　**服务原语的语义**

原语参数如下：

```
MLME-DE AUTHENTICATE. indication    (
                                     PeerSTAAddress,
                                     ReasonCode
                                     )
```

名　　称	类　　型	有效范围	描　　述
PeerSTAAddress (对等的 STA 的地址)	MAC 地址	任一有效的单个 MAC 地址	规定与之解除链路验证关系的对等 MAC 实体的地址
ReasonCode (原因代码)	如帧格式中所定义	如帧格式中所定义	规定启动解除链路验证过程的原因

10.3.5.3.3　**产生条件**

该原语由 MLME 产生，作为与规定的对等 MAC 实体解除链路验证关系的结果。

10.3.5.3.4　**收后效果**

通知 SME 指定的链路验证关系的解除情况。

29. 10.3.6 的内容替换为：

下列原语描述 STA 如何与接入点(AP)进行关联。

10.3.6.1　**MLME-ASSOCIATE. request**

10.3.6.1.1　**功能**

该原语请求与充当 AP 的指定的对等 MAC 实体进行关联。

10.3.6.1.2　**服务原语的语义**

原语参数如下：

```
MLME-ASSOCIATE. request    (
                            PeerSTAAddress,
                            AssociateFailureTimeout,
                            CapabilityInformation,
                            ListenInterval
                            WAPI
                            )
```

名　　称	类　　型	有效范围	描　　述
PeerSTAAddress (对等的 STA 的地址)	MAC 地址	任一有效的单个 MAC 地址	规定与之执行关联过程的对等的 MAC 实体地址
AssociateFailureTimeout (关联失败超时)	整数型	≥1	规定终止关联过程的时间界限(单位为 TU)
CapabilityInformation (能力信息)	如帧格式中所定义	如帧格式中所定义	规定 MAC 实体所用的操作能力定义
ListenInterval (侦听间隔)	整数型	≥0	规定在 STA 唤醒并侦听下一个信标之前可以经过的信标间隔数
WAPI	WAPI 信息元素	如帧格式中所定义	描述 BSS 所支持的密码套件与 AKM 套件

10.3.6.1.3　**产生条件**

当 STA 期望与 AP 进行关联时，该原语由 SME 产生。

10.3.6.1.4　**收后效果**

该原语启动关联过程。随后 MLME 发布一个反映结果的原语 MLME-ASSOCIATE. confirm。

10.3.6.2 **MLME-ASSOCIATE. confirm**

10.3.6.2.1 **功能**

该原语报告试图与充当 AP 的指定的对等 MAC 实体进行关联的结果。

10.3.6.2.2 **服务原语的语义**

原语参数如下：

```
MLME-ASSOCIATE. confirm    (
                            ResultCode
                            )
```

名　　称	类　　型	有效范围	描　　述
ResultCode (返回值)	枚举型	SUCCESS INVALID_PARAMETERS TIMEOUT REFUSED	指示原语 MLME-ASSOCIATE. request 的结果

10.3.6.2.3 **产生条件**

该原语由 MLME 产生，作为与充当 AP 的指定的对等 MAC 实体进行关联的原语 MLME- ASSOCIATE. request 的结果。

10.3.6.2.4 **收后效果**

通知 SME 关联过程的结果。

10.3.6.3 **MLME-ASSOCIATE. indication**

10.3.6.3.1 **功能**

该原语报告与规定的对等 MAC 实体建立关联。

10.3.6.3.2 **服务原语的语义**

```
MLME-ASSOCIATE. indication    (
                               PeerSTAAddress
                               WAPI
                               )
```

名　　称	类　　型	有效范围	描　　述
PeerSTAAddress (对等的 STA 的地址)	MAC 地址	任一有效的单个 MAC 地址	规定与之建立关联的对等 MAC 实体的地址
WAPI	WAPI 信息元素	如帧格式中定义	描述 BSS 所支持的密码套件与 AKM 套件。在 WAPI 信息元素中仅有一个单播密码套件

10.3.6.3.3 **产生条件**

该原语由 MLME 产生，作为与规定的对等 MAC 实体建立关联的结果。而此关联源于由规定的对等 MAC 实体启动的关联过程。

10.3.6.3.4 **收后效果**

通知 SME 关联关系的建立。

30. 10.3.7 的内容替换为：

下列原语描述 STA 如何与另外一个 AP 进行关联。

10.3.7.1 **MLME-REASSOCIATE. request**

10.3.7.1.1 **功能**

该原语请求关联关系改变至一个规定的新的充当 AP 的对等 MAC 实体。

10.3.7.1.2 **服务原语的语义**

原语参数如下：

```
MLME-REASSOCIATE.request    (
                             NewAPAddress,
                             ReassociateFailureTimeout,
                             CapabilityInformation,
                             ListenInterval
                             WAPI
                             )
```

名　　称	类　　型	有效范围	描　　述
NewAPAddress (新的 AP 地址)	MAC 地址	任一有效的单个 MAC 地址	规定与之执行重新关联过程的对等 MAC 实体的地址
ReassociateFailureTimeout (重新关联失败超时)	整数型	≥1	规定终止重新关联过程的时间界限(以 TU 为单位)
CapabilityInformation (能力信息)	如帧格式中定义	如帧格式中定义	规定 MAC 实体所用的操作能力定义
ListenInterval (侦听间隔)	整数型	≥0	规定在 STA 唤醒并侦听下一个信标帧前可以经过的信标间隔数
WAPI	WAPI 信息元素	如帧格式中定义	描述 BSS 所支持的密码套件与 AKM 套件

10.3.7.1.3 **产生条件**

该原语由 SME 产生，STA 用于将关联关系改变至一个规定的新的充当 AP 的对等 MAC 实体。

10.3.7.1.4 **收后效果**

该原语启动一个重新关联过程，随后 MLME 发布反映结果的原语 MLME-REASSOCIATE.confirm。

10.3.7.2 **MLME-REASSOCIATE.confirm**

10.3.7.2.1 **功能**

该原语报告试图与充当 AP 的规定对等 MAC 实体取得重新关联的结果。

10.3.7.2.2 **服务原语的语义**

原语参数如下：

```
MLME-REASSOCIATE.confirm    (
                             ResultCode
                             )
```

名　　称	类　　型	有效范围	描　　述
ResultCode (返回值)	枚举型	SUCCESS INVALID_PARAMETERS TIMEOUT REFUSED	指示原语 MLME-REASSOCIATE.request 的结果

10.3.7.2.3 **产生条件**

该原语由 MLME 产生，作为试图与一个规定的充当 AP 的对等 MAC 实体进行重新关联的原语

MLME-REASSOCIATE. request 的结果。

10.3.7.2.4 **收后效果**

通知 SME 重新关联过程的结果。

10.3.7.3 **MLME-REASSOCIATE. indication**

10.3.7.3.1 **功能**

该原语报告与一个规定的对等 MAC 实体建立重新关联。

10.3.7.3.2 **服务原语的语义**

```
MLME-REASSOCIATE. indication    (
                                 PeerSTAAddress
                                 WAPI
                                 )
```

名　　称	类　　型	有效范围	描　　述
PeerSTAAddress (对等的 STA 的地址)	MAC 地址	任一有效的单个 MAC 地址	规定与之建立重新关联的对等 MAC 实体的地址
WAPI	WAPI 信息元素	如帧格式中定义	描述 BSS 所支持的密码套件与 AKM 套件。在 WAPI 信息元素中仅有一个单播密码套件

10.3.7.3.3 **产生条件**

该原语由 MLME 产生，作为与规定的对等 MAC 实体建立重新关联的结果。而此重新关联源于规定的对等 MAC 实体启动的重新关联过程。

10.3.7.3.4 **收后效果**

通知 SME 重新关联已经建立。

31. 在 10.3.10 之后增加 10.3.11、10.3.12、10.3.13、10.3.14、10.3.15。

10.3.11 **设置 WPI 密钥**

下列原语设置 STA 启用 WPI 时与已鉴别的对等 STA 之间采用的密钥。

10.3.11.1 **MLME-SETWPIKEYS. request**

10.3.11.1.1 **功能**

该原语请求 STA 的 MAC 实体设置启用 WPI 时与已鉴别的对等 STA 的 MAC 实体之间采用的密钥。

10.3.11.1.2 **服务原语的语义**

原语参数如下：

```
MLME-SETWPIKEYS. request    (
                             Keylist
                             )
```

名　　称	类　　型	描　　述
Key (WPI 密钥值)	八位位组串	数据加密密钥与数据完整性校验密钥
Length (WPI 密钥长度)	整数型	密钥的长度，以八位位组为单位
KeyIdx (密钥索引值)	整数型	密钥的编号，取值为 0 或 1

名　　称	类　　型	描　　述
KeyType (密钥类型)	整数型	组播密钥、单播密钥、站间密钥
PeerSTAAddress (对等 STA 的地址)	MAC 地址	任一有效的单个 MAC 地址，当且仅当 KeyType 值为单播密钥或站间密钥时或者为 IBSS 下的组播密钥时有效
AE/ASUE 或者 Initiator/Peer (发起端/对端)	布尔类型	表示该密钥是由 AE 还是由 ASUE 设置，或者由发起端还是由对端设置。当取值为真时，表示密钥由 AE 或发起端设置的；为假时，表示密钥由 ASUE 或对端设置的
GSN (组播分组序号)	整数型	目前加密发送的组播分组序号，当 KeyType 为组播密钥类型时有效
Cipher Suite Selector (密码套件选择)	4 个八位位组	如 WAPI 信息元素格式中所定义，为本次关联请求的密码套件

10.3.11.1.3　产生条件

该原语当密钥协商完成时或共享密钥设置时由 SME 产生。

10.3.11.1.4　收后效果

该原语使 MAC 可利用密钥对数据进行保密处理。随后 MLME 发布一个反映结果的原语 MLME- SETWPIKEYS. confirm。

10.3.11.2　MLME-SETWPIKEYS. confirm

10.3.11.2.1　功能

该原语证实相关的 MLME- SETWPIKEYS. request 原语操作已经完成。

10.3.11.2.2　服务原语的语义

该原语无参数。

10.3.11.2.3　产生条件

该原语由 MLME 产生，用于响应接收到的 MLME- SETWPIKEYS. request 原语。当请求的操作完成时，该原语便被发布。

10.3.11.2.4　收后效果

通知 SME MLME- SETWPIKEYS. request 原语的请求操作完成。

10.3.12　删除 WPI 密钥

下列原语删除 STA 的 WPI 所使用的密钥。

10.3.12.1　MLME-DELETEWPIKEYS. reques

10.3.12.1.1　功能

该原语请求 STA 的 MAC 实体删除启用 WPI 时与已鉴别的对等 STA 的 MAC 实体之间采用的密钥。

10.3.12.1.2　服务原语的语义

原语参数如下：

```
MLME-DELETEWPIKEYS. request    (
                                Keylist
                               )
```

名　　称	类　　型	描　　述
Keylist (密钥列表)	DeleteKeyDescriptors 集合	MAC 使用的密钥列表

每个 DeleteKeyDescriptors 包含的元素如下表所示。

名　　称	类　　型	描　　述
PeerSTAAddress (对等 STA 的地址)	MAC 地址	任一有效的单个 MAC 地址，当且仅当 KeyType 值为单播密钥或站间密钥时或者为 IBSS 下的组播密钥时有效
KeyIdx (密钥索引值)	整数型	密钥的编号，取值为 0 或 1
KeyType (密钥类型)	整数型	组播密钥、单播密钥、站间密钥

10.3.12.1.3　产生条件

该原语当 STA 欲删除密钥时由 SME 产生。

10.3.12.1.4　收后效果

该原语使 MAC 禁止利用密钥对数据进行保密处理。随后 MLME 发布一个反映结果的原语 MLME-DELETEWPIKEYS. confirm。

10.3.12.2　MLME-DELETEWPIKEYS. confirm

10.3.12.2.1　功能

该原语证实相关的 MLME-DELETEWPIKEYS. request 原语操作已经完成。

10.3.12.2.2　服务原语的语义

该原语无参数。

10.3.12.2.3　产生条件

该原语由 MLME 产生，用于响应接收到的 MLME- DELETEWPIKEYS. request 原语。当请求的操作完成时，该原语便被发布。

10.3.12.2.4　收后效果

通知 SME MLME- DELETEWPIKEYS. request 原语的请求操作完成。

10.3.13　STAKey 的建立

10.3.13.1　MLME-STAKEYESTABLISHED. indication

10.3.13.1.1　功能

该原语通知 SME，需要一个 STAKey。

10.3.13.1.2　服务原语的语义

该原语有两个参数，即两个 STA 的 MAC 地址。

原语参数如下：

```
MLME-STAKEYESTABLISHED. indication    (
                                       Address1,
                                       Address2
                                       )
```

名　　称	类　　型	有效范围	描　　述
Address1 (地址 1)	MAC 地址	任一有效的单个 MAC 地址	主发方 STA 的 MAC 地址
Address2 (地址 2)	MAC 地址	任一有效的单个 MAC 地址	对端 STA 的 MAC 地址

10.3.13.1.3　产生条件

当需要一个 STAKey 时，该原语由 MAC 产生。

10.3.13.1.4 **收后效果**

SME 被告知需要一个 STAKey，并且涉及到两个 STA 的 MAC 地址。接着，SME 与两个 STA 交互 STAKey 握手消息。

10.3.14 **设置保护**

10.3.14.1 **MLME- SETPROTECTION. request**

10.3.14.1.1 **功能**

该原语指示发往指定 MAC 地址或接收来自指定 MAC 地址的帧是否需要保护。

10.3.14.1.2 **服务原语的语义**

原语参数如下：

```
MLME-SETPROTECTION. request    (
                                Protectlist
                               )
```

名　　称	类　　型	描　　述
Protectlist (保护列表)	保护元素集合	目前如何使用列表中的每个密钥

每个 Protectlist 包含的元素如下表所示。

名　　称	类　　型	描　　述
Address (地址)	MAC 地址	任一有效的单个 MAC 地址，当且仅当 KeyType 值为单播密钥或站间密钥时或者为 IBSS 下的组播密钥时有效
ProtectType (保护类型)	枚举型(None、Rx、Tx、Rx_Tx)	用于该 MAC 的保护
KeyType (密钥类型)	整数型	组播密钥、单播密钥、站间密钥

10.3.14.1.3 **产生条件**

当发往指定 MAC 地址或接收来自指定 MAC 地址的帧需要保护时。该原语由 SME 产生。

10.3.14.1.4 **收后效果**

收到该原语后，MA 按照 Protectlist 参数规定的 ProtectType 来设置保护并保护数据帧，定义如下：

——None：规定发往指定 MAC 地址或接收来自指定 MAC 地址的帧均不需要保护；

——Rx：规定接收来自指定 MAC 地址的帧应当保护；

——Tx：规定发往指定 MAC 地址帧应当保护；

——Rx_Tx：规定发往指定 MAC 地址或接收来自指定 MAC 地址的帧均应保护。

一旦规定保护发往指定 MAC 地址或接收来自指定 MAC 地址的帧，均应由 MLME-SETPROTECTION. request 原语复位。MLME-SETPROTECTION. request 原语通过定义 None 删除状态。

10.3.14.2 **MLME- SETPROTECTION. confirm**

10.3.14.2.1 **功能**

该原语规定帧的保护请求已经完成。

10.3.14.2.2 **服务原语的语义**

该原语无参数。

10.3.14.2.3 **产生条件**

该原语当保护请求完成时由 MAC 产生。

10.3.14.2.4 **收后效果**

SME 被通知保护请求已经完成。

10.3.15 **保护帧的丢弃**

10.3.15.1 **MLME- PROTECTEDFRAMEDROPPED. indication**

10.3.15.1.1 **功能**

该原语通知 SME，由于没有可用的临时密钥帧而被丢弃。

10.3.15.1.2 **服务原语的语义**

该原语有两个参数，即两个 STA 的 MAC 地址。

参数定义如下：

```
MLME- PROTECTEDFRAMEDROPPED. indication    (
                                            Address1,
                                            Address2
                                            )
```

名　　称	类　　型	有效范围	描　　述
Address1 (地址 1)	MAC 地址	任一有效的单个 MAC 地址	SA 的 MAC 地址
Address2 (地址 2)	MAC 地址	任一有效的单个 MAC 地址	RA 的 MAC 地址

10.3.15.1.3 **产生条件**

当由于没有可用的临时密钥帧而被丢弃时，该原语由 MAC 产生。

10.3.15.1.4 **收后效果**

SME 被通知帧丢弃。IBSS 中的 SME 可利用该原语激活与对端 STA 的安全关联。

32. 11.3 替换为：

11.3 链路验证与解除链路验证

本条描述了用于本规范链路验证与解除链路验证的程序。这些描述中所使用的状态在 5.5 中有描述。

11.3.1 链路验证——源 STA

当接收到一个 MLME-AUTHENTICATITE. request 原语时，源 STA 应当与指定的 STA 使用以下程序进行链路验证：

a) 在 ESS 中，或在 IBSS 中，STA 应当执行链路验证机制。

b) 如果链路验证成功，指定的 STA 的状态变量应当被设定为状态 2。

c) STA 应当产生一个 MLME-AUTHENTICATITE. confirm 原语以通知 SME 链路验证的结果。

在调用 MLME-AUTHENTICATITE. request 原语之前，STA 的 SME 应当使用 MLME-DELETEWPIKEYS. request 原语删除任何 USKSA 以及与指定的 STA 通信的临时密钥。

11.3.2 链路验证——目的 STA

当接收到一个链路验证帧，并且链路验证处理序列号为 1 时，目的 STA 应当与指定的 STA 使用以下程序进行链路验证：

a) STA 应当执行链路验证机制。

b) STA 应当产生一个 MLME-AUTHENTICATITE. indication 原语以通知 SME 链路验证的结果。

当接收到一个 MLME-AUTHENTICATITE. indication 原语时，STA 的 SME 应当使用 MLME-

DELETEWPIKEYS. request 原语删除任何 USKSA 以及与指定的 STA 通信的临时密钥。

如果处于 IBSS 中的 STA 的 SME 决定发起一个 WAPI 关联，并且 SME 不知道对端的安全策略，它可以通过调用 MLME-SCAN. request 产生一个单播探寻请求帧给对端，用以发现对端的安全策略。

11.3.3 解除链路验证——源 STA

当接收到一个 MLME-DEAUTHENTICATITE. request 原语时，源 STA 应当与指定的 STA 使用以下程序解除链路验证：

a) 如果指定 STA 的状态变量为状态 2 或状态 3，STA 应当发送一个解除链路验证帧给指定的 STA。

b) 指定的 STA 的状态变量应当被设定为状态 1。

c) STA 应当产生一个 MLME-DEAUTHENTICATITE. confirm 原语以通知 SME 解除链路验证的完成。

在调用 MLME-DEAUTHENTICATITE. request 原语之前，STA 的 SME 应当使用 MLME-DELETEWPIKEYS. request 原语并调用 MLME-SETPROTECTION. request(None)删除任何 USKSA 以及与指定的 STA 通信的临时密钥。

11.3.4 解除链路验证——目的 STA

当接收到一个解除链路验证帧时，目的 STA 应当与指定的 STA 使用以下程序解除链路验证：

a) 指定的 STA 的状态变量应当被设定为状态 1。

b) STA 应当产生一个 MLME-DEAUTHENTICATITE. indication 原语以通知 SME 解除链路验证。

当接收到一个 MLME-DEAUTHENTICATITE. indication 原语时，STA 的 SME 应当使用 MLME-DELETEWPIKEYS. request 原语并调用 MLME-SETPROTECTION. request(None)删除任何 USKSA 以及与指定的 STA 通信的临时密钥。

33. 11.4 替换为：

11.4 关联、重关联与解除关联

这个子条款描述了一个 STA 与一个 AP 如何完成关联与重关联，并且如何解除关联。这些描述中所使用的状态在 5.5 中有描述。

11.4.1 STA 关联程序

当接收到一个 MLME-ASSOCIATE. request 原语时，STA 应当与 AP 使用以下程序进行关联：

a) STA 应当向已经与它完成链路验证的 AP 发送一个关联请求帧。如果 MLME-ASSOCIATE. request 原语包含一个符合下列条件的 WAPI 信息元素：只有一个单播密码套件和一个已鉴别密钥套件，这个 WAPI 信息元素应当被包含在关联请求帧中。

b) 如果接收到的关联响应帧的状态值是“成功”，则 STA 与 AP 已取得关联。状态变量应当被设为状态 3，并且 MLME 应当产生一个 MLME-ASSOCIATE. confirm 原语以指示操作成功完成。

c) 如果接收到的关联响应帧的状态值是“成功”以外的其他值，或者 AssociateFailureTimeout 过期，则 STA 与 AP 未取得关联。MLME 应当产生一个 MLME-ASSOCIATE. confirm 原语以指示操作失败。

d) SME 应当通过呼叫 ProtectType 被设为“Rx_Tx,”的 MLME. SETPROTECTION. request 原语建立一个 WAPI 关联；或者当它不希望安全通讯时，则什么也不做。

在调用 MLME-ASSOCIATE. request 原语之前，STA 的 SME 应当使用 MLME-DELETEWPIKEYS. request 原语删除任何 USKSA 以及与指定的 STA 通信的临时密钥。

11.4.2 AP 关联程序

当从 STA 接收到一个关联请求帧时，AP 应当与 STA 使用以下程序进行关联：

a) 如果 STA 没有完成链路验证，则 AP 应当向 STA 发送一个解除链路验证帧来终止关联程序。

b) 在一个 WAPI 关联中，AP 应当检查接收到的 WAPI 信息元素值是否与 AP 的安全策略相匹配。如果不匹配，则关联不应被接收。

c) AP 应当终止一个关联响应。如果状态值是“成功”，分配给 STA 的关联标识符应当包含在响应中。

d) 当状态值为“成功”的关联响应被 STA 确认时，STA 被考虑与 AP 关联。STA 的状态变量应当被设为状态 3。

e) MLME 应当产生一个 MLME-ASSOCIATE. indication 原语以通知 SME 进行关联。

f) SME 应当通过呼叫 ProtectType 被设为“Rx_Tx,”的 MLME. SETPROTECTION. request 原语建立一个 WAPI 关联；或者当它不希望安全通讯时，则什么也不做。

g) SME 将通知 DS 新的关联。

当接收到一个 MLME-ASSOCIATE. indication 原语时，STA 的 SME 应当使用 MLME-DELETEWPIKEYS. request 原语删除任何 USKSA 以及与指定的 STA 通信的临时密钥。

11.4.3 STA 重关联程序

当接收到一个 MLME-REASSOCIATE. request 原语时，STA 应当与 AP 使用以下程序进行重关联：

a) 如果状态变量是状态 1，STA 应当通过产生一个 MLME-REASSOCIATE. confirm 原语来通知 SME 重关联失败。

b) STA 应当向新的 AP 发送一个重关联请求帧。如果 MLME-REASSOCIATE. request 原语包含一个符合下列条件的 WAPI 信息元素：只有一个单播密码套件和一个已鉴别密钥套件，这个 WAPI 信息元素应当被包含在重关联请求帧中。

c) 如果接收到的重关联响应帧的状态值是“成功”，则 STA 与 AP 已取得关联。状态变量应当被设为状态 3，并且 MLME 应当产生一个 MLME-REASSOCIATE. confirm 原语以指示操作成功完成。

d) 如果接收到的重关联响应帧的状态值是“成功”以外的其他值，或者 AssociateFailureTimeout 过期，则 STA 与 AP 未取得关联。MLME 应当产生一个 MLME-REASSOCIATE. confirm 原语以指示操作失败。

e) SME 应当通过呼叫 ProtectType 被设为“Rx_Tx,”的 MLME. SETPROTECTION. request 原语建立一个 WAPI 关联；或者当它不希望安全通讯时，则什么也不做。

在调用 MLME-REASSOCIATE. request 原语之前，STA 的 SME 应当使用 MLME-DELETEW-PIKEYS. request 原语删除任何 USKSA 以及与指定的 STA 通信的临时密钥。

11.4.4 AP 重关联程序

任何时候，当 AP 从 STA 接收到一个重关联请求帧时，AP 使用以下程序支持重关联：

a) 如果 STA 没有完成链路验证，则 AP 应当向 STA 发送一个解除链路验证帧来终止重关联程序。

b) 在一个 WAPI 关联中，AP 应当检查接收到的 WAPI 信息元素值是否与 AP 的安全策略相匹配。如果不匹配，则关联不应被接收。

c) AP 应当终止一个关联响应。如果状态值是“成功”，分配给 STA 的关联标识符应当包含在响应中。

d) 当状态值为“成功”的重关联响应被 STA 确认时，STA 被考虑与 AP 关联。STA 的状态变量应当被设为状态 3。

e) MLME 应当产生一个 MLME-REASSOCIATE. indication 原语以通知 SME 进行关联。

f) SME 应当通过呼叫 ProtectType 被设为“Rx_Tx,”的 MLME. SETPROTECTION. request 原语建立一个 WAPI 关联；或者当它不希望安全通讯时，则什么也不做。

g) SME 将通知 DS 新的关联。

当接收到一个 MLME-REASSOCIATE. indication 原语时，STA 的 SME 应当使用 MLME-DELETEWPIKEYS. request 原语删除任何 USKSA 以及与指定的 STA 通信的临时密钥。

11.4.5 STA 解除关联程序

当接收到一个 MLME-DISASSOCIATE. request 原语时，一个关联的 STA 应当使用以下程序与一个 AP 解除关联：

a) STA 应当向与它关联的 AP 发送一个解除关联帧。

b) 当且仅当 AP 的状态变量不为 1 时，它应当被设为状态 2。

c) MLME 应当产生一个 MLME-DISASSOCIATE. confirm 原语以指示操作成功完成。

在调用 MLME-DISASSOCIATE. request 原语之前，STA 的 SME 应当使用 MLME-DELETEWPIKEYS. request 原语并调用 MLME-SETPROTECTION. request(None)删除任何 USKSA 以及与指定的 STA 通信的临时密钥。

11.4.6 AP 解除关联程序

当从一个关联 STA 接收到一个解除关联帧时，AP 应当使用以下程序与 STA 解除关联：

a) STA 的状态变量应当被设为状态 2。

b) MLME 应当产生一个 MLME-DISASSOCIATE. indication 原语以通知 SME 解除关联。

c) SME 将更新 DS。

当接收到一个 MLME-DISASSOCIATE. indication 原语时，STA 的 SME 应当使用 MLME-DELETEWPIKEYS. request 原语并调用 MLME-SETPROTECTION. request(None)删除任何 USKSA 以及与指定的 STA 通信的临时密钥。

34. A.4.4.1 的 PC1、PC2 与 PC3 项目替换为：

项目	协议能力	引用条号	状态	支持
PC1	是否支持下列 MAC 协议能力？ 链路验证服务	5.4.3.1,5.4.3.2 5.7.6,5.7.7 附录 C	M	是□ 否□
PC1.1	链路验证状态	5.1.1	M	是□ 否□
PC1.2	开放系统链路验证	5.1.1	M	是□ 否□
PC2	安全	8	M	是□ 否□
PC2.1	WAPI 信息元素(IE)	7.3.2.8	M	是□ 否□
PC2.1.1	单播密码套件列表	7.3.2.8.2	M	是□ 否□
PC2.1.2	组播密码套件列表	7.3.2.8.3	M	是□ 否□
PC2.1.2.1	WPI 保密基础结构	8.2	M	是□ 否□
PC2.1.3	鉴别密钥管理套件列表	7.3.2.8.1	M	是□ 否□
PC2.1.3.1	WAI 证书鉴别和密钥管理	8.1	M	是□ 否□
PC2.1.3.2	WAI 预共享密钥鉴别和密钥管理	8.1	M	是□ 否□
PC2.1.3.3	WAI 鉴别和密钥管理	8.1	M	是□ 否□
PC2.1.3.3.1	密钥导出	8.1.4.10	M	是□ 否□
PC2.1.3.3.1.1	基密钥导出	8.1.4.10.1	M	是□ 否□
PC2.1.3.3.1.2	单播密钥导出	8.1.4.10.2	M	是□ 否□
PC2.1.3.3.1.3	组播密钥导出	8.1.4.10.3	M	是□ 否□
PC2.1.3.3.1.4	预共享密钥导出	8.1.4.10.4	M	是□ 否□
PC2.1.3.3.2	单播密钥协商	8.1.4.3	M	是□ 否□
PC2.1.3.3.3	组播密钥通告	8.4.3.2	M	是□ 否□
PC2.1.4	WAPI 能力	7.3.2.8	M	是□ 否□
PC2.1.5	WAPI 预鉴别	8.1.4.6	O	是□ 否□
PC2.1.6	WAPI 安全关联管理	8.1.2	M	是□ 否□
PC2.1.7	BKSA 缓存	8.1.4.7	CF1:M	是□ 否□
PC2.1.8	WAPI 扩展服务集(ESS)	8.1.4.6	CF1:M	是□ 否□
PC2.1.8.1	WAPI 站间密钥	8.1.4.5	O	是□ 否□
PC2.1.9	WAPI 独立基本服务集(IBSS)	8.1.2.1.2	CF2:O	是□ 否□
PC2.1.10	密钥更新	8.1.4.8	O	是□ 否□
PC2.1.11	X.509 v3 证书	8.1.3.1	M	是□ 否□
PC2.1.12	GBW 证书	8.1.3.2	O	是□ 否□

35. 删除 A.4.4.1 的 PC3。

36. 在第 C.2 章中的第一张图前增加以下文本：

“本条描述了不包括安全的 MAC 操作。”

37. 将第 C.2 章中图“3102_D/LmeEnum(31)”替换为：

Package macs or ts

3102_dLmeEnum(31)

```
/*********************************
*Mac和 Mime服务原语中使用的枚举类型
*********************************/
newtype AuthenticateType   /* Mime原语中的KAuthenticateType >参数 */
 inherits Octetstring operators all   ;
 adding literals open _system ;
 axioms Authenticate _system ==mkOS (0, 2) ;
endnewtype AuthenticateType  ;
newtype AuthenticateTypeSet powerset  (AuthenticateType );
endnewtype AuthenticateTypeSet  ;
newtype BssType /* <BssType>参数和BSS描述元素*/
 literals infrastructure,independent,any_bss;endnewtype BssType;
newtype BssTypeSet powerset(BssType); endnewtype BssTypeSet;
newtype CfPriority / *各种请求的<Priority >参数*/
 literals contention,contentionFree; endnewtype CfPriority,
newtype MibStatus /* Mime/PPime Get/Set confirm的<status >参数*/
 literals success,invalid,write_only,read_only;
endnewtype Mibstatus;
newtype MimeStatus / *Mime操作确认的<status >参数*/
 literals success,invalid,timeout,refused,
      tomany_req,already_bss; endnewtype MimeStatus;
newtype PwrSave / *MimePowerMgt的<power save mode >参数*/
 literals sta_active,power_save; endnewtype PwrSave;
newtype Routing / *MAC数据服务的<routing info >参数*/
 literals null_rt;endnewtype routing;
newtype RxStatus / *MaUnitdata指示的Kreception status >参数*/
 literals rx_success,rx_failure, endnewtype RxStatus;
newtype ScanType / *MimeScan request的< scan type >参数*/
 literals active_scan,passive_scan; endnewtype ScanType;
newtype ServiceClass / *MaUnitdata的<service class >参数*/
 literals reorderable,strictlyOrdered; endnewtype ServiceClass,
newtype TxStatus / *MaUnitdata Status的<transmission status >参数*/
 literals successful,retryLimit,txLifetime,noBss,
      excessiveDataLength,nonNullSourceRouting,
      unsupportePriority,unavailablePriority,
      unsupportedServiceClass,unavailableServiceClass,
      unavailableKeyMapping endnewtype TxStatus;
```

38. 将第 C.2 章中图“3104_d\StationData(31)”替换为：

Packge macsorts　　3104_d\StaticData(31)

```
/*************************************
*命名静态数据数值 （形式为sXYZ的名称）
*************************************/
synonym sMaxMsduLng integer=2278; /*MSDU中最大的八位位组数 */
synonym sMacHdrLng integer=24; /*数据头中的八位位组数 */
synonym sWdsAddLng integer=6; /*为WDS(addr4)而添加的八位位组数 */
synonym sCrcLng integer=4; /*用于Crc32（FCS）八位组 */
synonym sMaxMpduLng integer= /*MPDU中的最大的八位位组数 */
 (sMaxMsduLng+sMacHdrLng+sWdsAddLng+sCrcLng);
synonym FrameindexRange=integer /*用于MPDU字节的索引范围 */
 constants0:sMaxMpduLng endsyntype FrameindexRange,
synonym sTsOctet integer=24; /*时戳字段的第一个八位位组数 */
synonym sMinFragLng integer=256; /*用于aMpduMaxLength的最小值 */
synonym sMaxFragNum integer= /*最大分段数 */
 (sMaxMsduLng/(sMinFragLng-sMacHdrLng-sCrcLng));
synonym sAckCtsLng integer=112; /*在ACK和CTS帧中的比特数 */
```

```
/*******************************
*站配置标记 （静态，对MIB的增补）
*******************************/
synonym sVersion integer=0; /*支持的协议版本*/
synonym sCanBeAp Boolea=false;  /*若STA能作为AP操作，则等于真*/
synonym sCanBePc Boolea=false;  /*若AP能作为点协调器，则等真*/
synonym sCfPollable Boolean=true; /*若为对CF_poll响应，则为真*/
```

39. 将第 C.2 章中图“3115_d\Defragment(31)”替换为：

Package macsorts　　3115_d\Defragment(31)

```
/***************************
        重组支持类型
/************************
/* PartialSdu结构用于其至少一个分段已被收到的
不完整的(INCOMPLETE)MSDUs/MMPDUs(统称SDUs)。未分段
的SDU立即被向上报告，且在该结构实例中不存储 。*/
newtype PartialSdu struct
  inUse Boolean,/*若该实例保持任何分段，则为真*/
  rta MacAddr,/*发送站(Addr2)*/
  rsn SeqNum,/*SDU序列号*/
  rCur FragNum,/*最近的Mpdu的分段号*/
  reol Time,/* (now+dUsec(aMaxReceiveLifetime)@第一个Mpdu*/
  rsdu Frame,/*Mpdus串联处的缓冲区 */
 default(false,nullAddr,0,0,0,null.);
endnewtype PartialSdu;
/*在该站重组队列中的项目数 */
/*值是依赖于实现的 (min=3,见9.5)*/
synonym defragSize integer=6;
syntype defragindex=integer constantsl:defragSize
endsyntype defragindex;
/*用于重组Msdus和Mmpdus的PartialSdu队列*/
/* 使用ArSearch算子index:=ArSearch(array,addr,seq,frag)可查找,这里
返回索引来标识符合 ((inUse=true)和(entry!rta=addr)和(entry!rsn=seq)
和(entry!rCur=(frag-1))的第一个元素,若无匹配项目，则等于1 */
/*index:=ArFree(array)返回一个自由项目的索引，若无项目空闲则返回 -1。
可释放用规定算法选定的项目，以避免返回-1 */
/* array:=ArAge(array,age)*/
/* 在(entry!eol<age)处释放，也用于清空队列 */
newtype DefragArray Array(defragindex,PartialSdu);
 adding operators
  ArSearch:DefragArray,MacAddr,SeqNum,FragNum->integer;
  ArFree:DefragArray->integer;
  ArAge:DefragArray,Time->DefragArray;
 operator ArSearch,
  fpar ar DefragArray, adr MacAddr,seq SeqNum,frg FragNum;
  returns integer,referenced;
 operator ArFree fpar ar DefragArray; returns integer; referenced;
 operator ArAge,fpar ar DefragArray,age Time;
  returns DefragArray,referenced;
endnewtype DefragArray,
newtype DefragKeysArray Array(defragindex,PartialSduKeys);
endnewtype DefragKeysArray,
```

40. 将第 C.2 章中图“3117_d\Frame_1(31)”替换为：

Package macsorts　　3117_d\Frame_1(31)

```
/*****************************************************
 *     FRAME类（MAC帧中的字段的基本定义 ）
 *****************************************************/
/*Frame是带算子的Octetstring的子类型,算子可用于创建 MAC头，提取每
个头字段和一些管理帧字段 ，和修改这些字段中的大多数 。
/*有创建和提取管理帧的元素的算子 ,但没有用于帧体 ，WPI头，MIC和
FCS字段的算子，它们被作为 Octetstring直接处理 。*/
newtype Frame inherits Octetstring operators all,
adding operators
 mkFrame :TypeSubtype,MacAddr,MacAddr,Octetstring->Frame;
 mkCtl :TypeSubtype,Octetstring,MacAddr->Frame;
 protocolVer :Frame-> integer,/*协议版本 (2个比特 )*/
 basetype : Frame->BasicType; /*类型字段 (2个比特)*/
 ftype :Frame-> TypeSubtyoe, /*类型和子类型 (6个比特)*/
 setFtype :Frame,TypeSubtype->Frame;
 toDs :Frame->Bit; /*  To DS比特(1 比特) */
 setToDs :Frame,Bit-> Frame;
 frDs :Frame->Bit, /*  Frame DS比特(1 比特)*/
 setFrDs :Frame,Bit->Frame;
 moreFrag ;Frame->Bit; /*More Fragments比特(1 比特)*/
 setMoreFrag :Frame,Bit->Frame;
 retryBit :Frame-> Bit, /* Retry比特(1 比特) */
 setRetryBit :Frame,Bit->Frame;
 pwnMgt :Frame->Bit, /* Power Management比特(1 比特)*/
 setPwrMgt :Frame,Bit-> Frame;
 moreData :Frame->Bit; /* More Data比特 */
 setMoreData :Frame,Bit,-.Frame;
 protectedBit : Frame -> Bit; /*PROTECTED 比特 (1  比特)*/
 setProtectedBit : Frame , Bit -> Frame;
 orderBit :Frame->Bit, /* {strictly}Order{ed}(1 比特) */
 setOrderBit :Frame,Bit->Frame;
 durId :Frame->integer, /*Duration/ID字段(2)*/
 setDurId : Frame,integer->Frame;
 addr1 :Frame->MacAddr, /*Address 1 [DA/RA]字段(6)*/
 setAddr1 :Frame,MacAddr->Frame;
 addr2 :Frame->MacAddr, /*Address 2 [SA/TA]字段 (6) */
 setAddr2 :Frame,MacAddr->Frame;
 addr3 :Frame->MacAddr, /*Address 3  [Bss/DA/SA]字段 */
 setAddr3 :Frame,MacAddr->Frame;
 addr4 :frame-> MacAddr, /*Address 4 [WDS-SA]字段 (6) */
 insAddr4 :Frame,MacAddr->Frame;
 sep :Frame->SepNum; /*序列号(12 比特)*/
 setSeq :Frame,SeqNum->Frame;
 frag :Frame->FragNum; /*分段号 (4 比特 )*/
 setFrag :Frame,FragNum->Frame;
 ts :Frame->Time; /*时戳字段 (8)*/
 setTs :Frame,Time->Frame;
 mkElem :elementID,Octetstring->Frame; /*获得元素  */
 GetElem :Frame,ElementID->Frame;  /*若可得，则得到元素  */
 status :Frame->StatusCode; /*状态代码字段 (2)*/
 setStatus :Frame,StatusCode->Frame;
 LinkVerifyStat :Frame->StatusCode; /*链路验证帧中的状态码  */
 reason :Frame->ReasonCode; /* 原因代码字段 (2) */

/*Frame算子转下页 … */
```

获取管理帧体的成员，如果目标成员不存在，则返回0长度的字节串。

41. 将第 C.2 章中图“3119_d\Frame_3(31)”替换为：

Package macsorts　　3119_d\Frame_3(31)

```
/* ...Frame类公理接上页 */
 forall b in Bit(
  toDs(f)==if(f(1)and 0x01) then 1 else 0 fi;
   setToDs(f,b)==
     Modify !(f,1,(f(1)and 0xFE) or S8(0,0,0,0,0,0,0,b));
   frDs(f)==if(f(1) and 0x02) then 1 else 0 fi;
   setFrDs(f,b)==
     Modify !(f,1,(f(1) and 0xFD) or S8(0,0,0,0,0,0,b,0));
   moreFrag(f)==if(f(1) and 0x04) then 1 else 0 fi;
   setMoreFrag(f,b)==
     Modify (f,1,(f(1) and 0xFB) or S8(0,0,0,0,0,b,0,0));
   retryBit(f)==if(f(1) and 0x08) then 1 else 0 fi;
   setRetryBit(f,b)==
     Modify !(f,1,(f(1) and 0xF7 or S8(0,0,0,0,b,0,0,0));
   pwrMgt(f)==if(f(1) and 0x10) then 1 else 0 fi;
   setPwrMgt(f,b)==
     Modify !(f,1,(f(1) and 0xFB) or S8(0,0,0,b,0,0,0,0));
   moreData(f)==if(f(1) and 0x20) then 1 else 0 fi;
   setMoreData(f,b)==
     Modify !(f,1,(f(1) and 0xFB or S8(0,0,b,0,0,0,0,0));
   protectedBit(f)==if(f(1)and 0x40) then 1 else 0 fi;
   setProtectedBit(f,b)==
     Modify !(f,1,(f(1) and 0xFB) or S8(0,b,0,0,0,0,0,0));
   orderBit(f)==if(f(1) and 0x80) then 1 else 0 fi;
   setOrderBit(f,b)==
     Modify (f,1,(f(1) and 0xFB) or S8(b,0,0,0,0,0,0,0));
   for all c in Capability(
   capA(f,c)==if (B_S(SubStr(f,24,2)) and c) then 1 else 0 fi;
    setCapA(f,c,b)==SubStr(f,0,24)//(B_S(SubStr(f,24,2) and
      (not c)) or (if b then c else O2 fi))//
      SubStr (f,26,Length(f)-26);
   capB(f,c)==if (B_S(SubStr(f,34,2)) and c) then 1 else 0 fi;
     setCapB (f,c,b)==SubStr(f,0,34)//(B_S(SubStr(f,34,2) and
        (not c)) or (if b then c else O2 fi))//
        SubStr (f,36,Length(f)-36);));
      for all sq in SeqNum (
    seq(f)==(octetVal(f(22) and 0xF0)/16)+(octetVal(f(23)*16));
     setSeq (f,sq)==SubStr(f,0,22)//MkString((f(22) and 0x0F)
        or mkOctet ((sq mod 16)*16))//mkOS(sq/16,1)//
        SubStr (f,24,Length(f)-24););
       for all fr in FragNum (
        frag (f)==octetVal(f(22) and 0x0f);
        setFrag (f,fr)==
          SubStr (f,0,22)//MkString((f(22) and 0xF0) or
          mkOctet (fr))//SubStr(f,23,Length(f)-23););
        for all tm in Time (
         ts (f)==tUsec(Usec(octetVal(f(24))+
           ( 256*(octetVal(f(25))+
           ( 256*(octetVal(f(26))+
           ( 256*(octetVal(f(27))+
           ( 256*(octetVal(f(28))+
           ( 256*(octetVal(f(29))+
           ( 256*(octetVal(f(30))+
           ( 256*octetVal(f(31)))))))))))))))))) ) ) ;
/*Frame类公理转下页 … */
```

42. 将第 C.2 章中图“3120_d\Frame_4(31)”替换为：

Package macsorts 3120_d\Frame_4(31)

```
/* …Frame类公理接上页 */
setTs(f,tm)==SubStr(f,0,24)//mkOS(fix(tm),1)//
 mkOS((fix(tm)/256),1)//mkOS((fix(tm)/65536),1)//
 mkOS((fix(tm)/16777216),1)//
 mkOS((fix(tmO/4294967296),1)//
 mkOS(((fix(tm)/4294967296256),1)//
 mkOS(((fix(tm)/4294967296665536),1)//
 mkOS(((fix(tm)/4294967296/16777216),1)//
     SubStr (f,32,Length(f)-32););
    for all stat in StatusCode(
     status (f)==SubStr(f,26,2);
     setStatus (f,stat)==
      SubStr (f,0,26)//stat/SubStr(f,28,Length(f)-28);
     suthStat (f)==SubStr(f,28,2););
   for all rea in ReasonCode( reason(f)==SubStr(f,24,2); );
   for all alg in AuthType(
     AuthType(f)==SubStr(f,24,2); );
   for all u in TU(
     beaxonInt(f)==octetVal(f(32))+(octetVal(f(33))*256);
   listenInt(f)==octetVal(f(26))+(octetVal(f(27))*256););
     for all sta in AssocId(
      AId(f)==octetVal(f(28))+(octetVal(f(29))*256);
      setAId(f,sta)==SubStr(f,0,28)//mkOS(sta mod 256,1)//
        mkOS (sta/256, 1)//SubStr(f,30,Length(f)-30);;)))
endnewtype Frame;
```

```
/*****************************************************************
*      ReasonCode(原因代码)类
*****************************************************************/

newtype ReasonCode inherits Octetstring operators all;
 adding literals unspec_reason, Auth_not_valid,
  deAuth_lv_ss,
  inactivity, ap_overload, class2_err, class3_err,
  disas_lv_ss, asoc_not_Auth;
 axioms
  unspec_reason==mkOS(1,2); Auth_not_valid==mkOS(2,2);
  deAuth_lv_ss=mkOS(3,2); inactivity==mkOS(4,2);
  ap_overload==mkOS(5,2); class2_err==mkOS(6,2);
  class3_err==mkOS(7,2); disas_lv_ss==mkOS(8,2);
  asoc_not_Auth==mkOS(9,2);
endnewtype ReasonCode;
```

```
/*****************************************************************
*      StatusCode(状态代码)类
*****************************************************************/

newtype StatusCode inherits Octetstring operators all;
 adding literals successful, unspec_fail, unsup_cap,
  reason_no_asoc, fail_other, unsupt_alg,Auth_seq_fail,
  chlng_fail, Auth_timeout, ap_full, unsup_rate;
 axioms
  successful==mkOS(0,2); unspec_failure==mkOS(1,2);
  unsup_cap==mkOS(10,2);reasoc_no_asoc==mkOS(11,2);
  fail_other==mkOS(12,2); unsupt_alg==mkOS(13,2);
  Auth_seq_fail==mkOS(14,2); chlng_fail==mkOS(15,2);
  Auth_timeout==mkOS(16,2); ap_full==mkOS(17,2);
  unsup_rate==mkOS(18,2);
endnewtype StatusCode;
```

43. 将第 C.2 章中图“GetElem_1a(1)”替换为：

44. 将第 C.2 章中图“3121_d\FrameType(31)”替换为：

Package macsorts 3121_d\FrameType(31)

```
/*****************************************************
 * Frame Type (帧类型)类
*****************************************/
/* TypeSubtype定义定义完整的6比特帧类型标识符。                    */
/* 这些值对于Frame类的ftype算子是有用的。                        */
newtype TypeSubtype inherits Octetstring operators all;
 adding literals asocreq, asoc_rsp, reasoc_req, reasoc_rsp,
  probe_req, probe_rsp, beacon,atim, disasoc, Auth, deauth,
  ps_poll,rts,cts,ack,cfend,cfend_ack,data,data_ack,
  data_poll,data_poll_ack,null_frame,cfack,cfpoll,cfpoll_ack;
axioms
  asoc_req==MkString(S8(0,0,0,0,0,0,0,0));
  asoc_rsp==MkString(S8(0,0,0,0,1,0,0,0));
  reasoc_req==MkString(S8(0,0,0,0,0,1,0,0));
  reasoc_rsp==MkString(S8(0,0,0,0,1,1,0,0));
  probe_req==MkString(S8(0,0,0,0,0,0,1,0));
  probe_rsp==MkString(S8(0,0,0,0,1,0,1,0));
  beacon==MkString(S8(0,0,0,0,0,0,0,1));
  atim==MkString(S8(0,0,0,0,1,0,0,1));
  disasoc==MkString(S8(0,0,0,0,0,1,0,1));
  Auth==MkString(S8(0,0,0,0,1,1,0,1));
  deauth==MkString(S8(0,0,0,0,0,0,1,1));
  ps_poll==MkString(S8(0,0,1,0,0,1,0,1));
  rts==MkString(S8(0,0,1,0,1,1,0,0));
  cts==MkString(S8(0,0,1,0,0,0,1,1));
  ack==MkString(S8(0,0,1,0,1,0,1,1));
  cfend==MkString(S8(0,0,1,0,0,1,1,1));
  cfend_ack==MkString(S8(0,0,1,0,1,1,1,1));
  data==MkString(S8(0,0,0,1,0,0,0,0));
  data_ack==MkString(S8(0,0,0,1,1,0,0,0));
  data_poll==MkString(S8(0,0,0,1,0,1,0,0));
  data_poll_ack==MkString(S8(0,0,0,1,1,1,0,0));
  null_frame==MkString(S8(0,0,0,1,0,0,1,0));
  cfack==MkString(S8(0,0,0,1,1,0,1,0));
  cfpoll==MkString(S8(0,0,0,1,0,1,1,0));
  cfpoll_ack==MkString(S8(0,0,0,1,1,1,1,0));
endnewtype TypeSubtype
/*BasicTypes定义2比特帧类型组 */
newtype BasicType inherits Bitstring operators all;
  adding literals control,data management, reserved;
  axioms
   control==S8(0,0,1,0,0,0,0,0);  data==S8(0,0,0,1,0,0,0,0);
   management==S8(0,0,0,0,0,0,0,0);  reserved==S8(0,0,1,1,0,0,0,0);
endnewtype BasicType;
```

45. 将第 C.2 章中图“3201_d\StationConfig(5)”替换为：

```
use macsorts;

Package macmib                                          3201_d\StationConfig(5)

/*该包包含MAC MIB属性的定义和MAC状态机使用的 PHY MIB属性的子集。在
Z.100下需要这些内容以允许分析状态机定义 。在将来的修订本中这些内容可
被附录D中出现的 ASN.1 MIB定义代替，因为有符合Z.105的SDL工具可供使用 */

/*****************************************************
*   StationConfig表
*****************************************************/
remote dot11StationID MacAddress nodelay;
remote dot11MediumOccupancyLimit TU nodelay;
synonym dot11CfPollable Boolean=<<package macsorts>>sCfPollable;
remote dot11CfPeriod Integer nodelay;
remote dot11CfpMaxDuration TU nodelay;
remote dot11AuthenticateResponse Timeout TU nodelay;
remote dot11PowerMangementModeMode PwrSave nodelay;
remote dot11DesiredSsid OctetString nodelay;
remote dot11DesiredBssType BssType nodelay;
remote dot11OperationalRateSet RateString nodelay;
remote dot11BeaconPeriod TU nodelay;
remote dot11DtimPeriod Integer nodelay;
remote dot11AssociationResponseTimeout TU nodelay;
remote dot11ReceiveDTIMs Boolean nodelay;
remote dot11AuthenticateType AuthTypeSet nodelay;

/*****************************************************
*   AuthenticateAlgorithms表
*****************************************************/

synonym dot11AuthenticateAlgorithms AuthenticateTypeSet=
  incl(open_system);
   /*注:该集合的成员是所有的
    dot11AuthenticateAlgorithmsEntry实例
    的dot11AuthenticateAlgorithm值。
    */
```

46. 将第 C.2 章中图“3202_d\PrivOperation(5)”替换为：

```
use macsorts:

Package macmib                                          3202_d\PrivOperation(5)

/*****************************************************
 *    Operation表
 *****************************************************/
synonym dot11MacAddress MacAddr=
  <<type MacAddr>>S6(0x00,0x11,0x22,0x33,0x44,0x55);
  /*每个站都有(唯一的)全局管理地址        */
  /*它的值可能在 MlmeReset时被本地管理地址重写     */
  /*但在MAC操作期间总是静态的            */
remote dot11RtsThreshold Integer nodelay;
remote dot11ShortRetryLimit Integer nodelay;
remote dot11LongRetryLimit Integer nodelay;
remote dot11Fragmentation Threshold Integer nodelay;
remote dot11MaxTransmitMsduLifetime TU nodelay;
remote dot11MaxReceiveLifetime TU nodelay;
synonym dot11ManufacturerId Charstring='name of manufacturer';
synonym dot11ProductId Charstring='identifier unique to manufacturer';

/*********************************************
 *    GroupAddresses表
 *********************************************/
remote dot11GroupAddresses MacAddrSet nodelay;
```

47. 在第 C.3 章中的第一张图前增加以下文本：

“本条描述了不包括安全的 MAC 操作。”

48. 将第 C.3 章中图“Station_1b(3)”替换为：

49. 将第 C.3 章中图“Sta_signals_2d(3)”替换为：

```
use macsorts;
use macmib;
```

System Station　　Sta_signals_2d(3)

```
signal
 AtimW,
 Backoff(Integer, Integer),
 BkDone(Integer),
 Busy,
 Cancel,
 CfPolled,
 ChangeNav(Time, Duration,NavSrc),
 Doze,
 Idle,
 MaUnitdata indication(MacAddr,MacAddr,
  Routing Octetstring RxStatus,
  CfPriority, ServiceClass),
 MaUnitdatarequest(MacAddr,MacAddr,
  Routing Octetstring CfPriority, ServiceClass),
 MaUnitdataStatusindication(MacAddr,
  MacAddr,TxStatus,CfPriority, ServiceClass),
 MlmeAssociate confirm( MlmeStatus),
 MlmeAssociate indication(MacAddr),
 MlmeAssociate request( MacAddr,Kusec,
  Capability Integer),
    MlmeAuth  . confirm
 ( MacAddr, AuthType,MlmeStatus),
    MlmeAuth  . indication(MacAddr,    AuthType ),
    MlmeAuth  . request( MacAddr,    AuthType  , Kusec),
     MlmeDeauth  .confirm( MacAddr, MlmeStatus),
     MlmeDeauth  . indication(MacAddr, ReasonCode),
     MlmeDeauth  .request( MacAddr, ReasonCode),
 MlmeDisassociateconfirm( MlmeSataus),
 MlmeDisassociateindication(MacAddr, ReasonCode),
 MlmeDisassociaterequest( MacAddr,ReasonCode),
 MlmeGet confirm( MibStatus, MibAtrib,MibValue),
 MlmeGet request(MibAtrib),
 MlmeJoin confirm( MlmeStatus),
 MlmeJoin request(BssDscr, Integer,Usec, Ratestring),
 MlmePowermgtconfirm( MlmeStatus),
 MlmePowermgtrequest( PwrSave, Boolean, Boolean),
 MlmeReassociateconfirm( MlmeStatus),
 MlmeReassociateindication(MacAddr),
 MlmeReassociaterequest(MacAddr, Kusec,
  Capability, Integer),
 MineReset confirm( MlmeStatus),
 MlmeReset request( MacAddr, Boolean),
 MlmeScan confirm( BssDscrSet, MlmeStatus),
 MlmeScan request( BssTypeSet, MacAddr, Octetstring,
  ScanType,Usec, Intstring Kusec, Kusec),
 MlmeSet confirm( MibStatus,MibAtrib),
 MlmeSet request( MibAtrib,MibValue),
 MlmeStart confirm( MlmeStatus),
 MlmeStart request( Octetstring BssType Kusec,
  Integer, CfParms, PhyParms, IbssParms, Usec,
  Capability,Ratestring,Ratestring);
```

```
signal
 MmCancel,
 MmConfirm( Frame, TxStatus),
 MmIndicate(Frame, Time, Time, StateErr),
 MmRequest(Frame, Imed, Rate),
 MsduConfirm(Frame, CfPriority, TxStatus),
 MsduIndicate(Frame, CfPriority),
 MsduRequest(Frame, CfPriority),
 NeedAck(MacAddr,Time, Duration,Rate),
 PduConfirm(FragSdu, TxResult),
 PduRequest(FragSdu),
 PhyCca Indication(Ccastatus),
 PhyCcarst confirm,
 PhyCcarst request,
 PhyData confirm,
 PhyData indication( Octet),
 PhyRxEnd indication(PhyRxStat),
 PhyRxStart indication(Integer,Rate),
 PhyTxEnd confirm,
 PhyTxEnd request,
 PhyTxStart confirm,
 PhyTxStart request(Integer, Rate),
  PlmeCharacteristicsconfirm( PhyChrstcs),
  PlmeCharacteristicsrequest,
 PlmeGet confirm( MibStatus,
  MibAtrib,MibValue),
 PlmeGet request(MibAtrib),
 PlmeReset confirm( Boolean),
 PlmeReset request,
 PlmeSet confirm( MibStatus,MibAtrib),
 PlmeSet request( MibAtrib,MibValue),
 PsmDone,
 PsChange(MacAddr, PsMode),
 PsIndicate( MacAddr,PsMode),
 PsInquiry( MacAddr),
 PsResponse(MacAddr,PsMode),
 ResetMAC,
 RxCfAck( MacAddr),
 RxIndicate(Frame, Time, Time, Rate),
 Slot,
 SsInquiry( MacAddr),
 SsResponse(MacAddr,
  StationState, StationState),
 SwChnl(Integer, Boolean),
 SwDone,
 TBTT,
 TxConfirm,
 TxRequest(Frame,Rate),
 Wake;
```

50. 将第 C.3 章中图“Sta_signallists_3c(3)”替换为：

51. 将第 C.3 章中图“Sta_Mpdu_ gen_1a(1)”替换为：

52. 将第 C.3 章中图“prepare_1b(2)”替换为：

53. 将第 C.3 章中图“Fragment_2b(2)”替换为：

54. 将第 C.3 章中图“rx_coord_2b(4)”替换为：

55. 将第 C.3 章中图“rx_coord_3b(4)”替换为：

56. 将第 C.3 章中图“rxcoord_3.1a(4)”替换为：

57. 将第 C.3 章中图“Mac_Mgmt_1a(1)”替换为：

58. 将第 C.3 章中图“Mib_access_1a(2)”替换为：

59. 将第 C.3 章中图“Mib_import_export_2b(2)”替换为：

Process MIB

Mib_import_export_2b(2)

```
/* 从其他进程出口的(只读)MIB计数器
值的输出 */
imported
dot11AckFailureCount,
dot11FailedCount,
dot11FcsErrorCount,
dot11FrameDuplicateCount,
dot11MulticateReceivedFrameCount,
dot11MulticastTransmittedFrameCount,
dot11MultipleRetryCount,
dot11ReceivedFragmentCount,
dot11RetryCount,
dot11RtsFailureCount,
dot11RtsSuccessCount,
dot11TransmittedFragmentCount,
```

```
/*从该过程输出的MIB属性的声明*/

    /* 读写属性 */
dcl exported
  dot11AuthenticationAlgorithms AuthTypeSet:=
    incl(open_system),
  dot11FragmentationThteshold Integer:=2346,
  dot11GroupAddresses MacAddrSet:=empty,
  dot11LongRetryLimit Integer:=4,
  dot11MaxReceiveLifetime Kusec:=512,
  dot11MaxTransmitMsduLifetime Kusec:=512,
  dot11MediumOccupancyLimit Kusec:=100,
  mReceiveDTIMs Boolean:=true,
  dot11CfpPeriod Integer:=1,
  dot11CfpMaxDuration Kusec:=200,
  dot11AuthenticationResponseTimeout Kusec:=512,
  dot11RtsThreshold Integer:=3000,
  dot11ShortRetryLimit Integer:=7,
  dot11CurrentChannelNumber Integer:=0,
  dot11CurrentSet Ingeter:=0,
  dot11CurrentPattern Ingeter:=0,
  dot11CurrentIndex Ingeter:=0;
```

```
/* 以下MAC MIB中的只读属性被定义为同义词（命
   名的常量)而不是远程变量，因为它们描述静态
   站的性质，这些站至少在MAC操作的任何单一实
   例期间静态的。

 dot11AuthenticationAlgorithms AuthTypeSet
 dot11CfPollable Boolean,
 dot11MacAddress MacAddr,
 dot11ManufacturerID Octetstring,
 dot11ProductID Octetstring,
 aStationID MacAddr,

   另外，由MAC访问的PHY MIB中所有的只读属性定
   义为同义词
*/
```

```
/* 注：
   列出的用于MAC MIB属性的值为用于那些属性的规定的
   默认值。列出的用于PHY MIB属性的值或者是用于FH
   PHY的默认值，或者是规定范围内的任意值 。在MAC的
   SDL描述中，用于PHY属性的特定值没有标准化意义
*/
```

60. 将第 C.3 章中图“Mlme_request_1b(3)”替换为：

61. 将第 C.3 章中图“Mlme_request_3a(3)”替换为：

62. 将第 C.3 章中图“MLME_1a(1)”替换为：

63. 将第 C.3 章中图“ps_monitor(2)”替换为：

64. 将第 C.3 章中图“ps_monitor_2a(2)”替换为：

65. 将第 C.3 章中图“sta_Mm_svc_1b (2)”替换为：

66. 将第 C.3 章中图“sta_Mm_svc_1.1a(2)”替换为：

67. 将第 C.3 章中图“mmpdu_svc_1.1b(2)”替换为：

68. 将第 C.3 章中图“Auth_req_1a(2)”替换为：

69. 将第 C.3 章中图“Deauth_2a(2)”替换为：

70. 将第 C.3 章中图“Auth_rsp_1b(2)”替换为：

71. 在第 C.3 章中图“Auth_rsp_1b(2)”之后，插入图“auth_rsp_2b(2)”：

72. 将第 C.3 章中图“receive_1a(1)”替换为：

73. 将第 C.3 章中图“nav_clear_1b(2)”替换为：

74. 将第 C.3 章中图“start_rx_1b(2)”替换为：

75. 将第 C.3 章中图“Validate_rx_2c(2)”替换为：

76. 将第 C.3 章中图“pre_filter_1d(4)”替换为：

77. 将第 C.3 章中图“report_rx_4a(4)”替换为：

78. 在第 C.3 章中图“report_rx_4a(4)”之后，插入图“filter_1b(3)”：

79. 将第 C.3 章中图“defragment_3c(3)”替换为：

80. 在 C.4 中的第一张图前增加以下文本：

“本条描述了不包括安全的 MAC 操作。”

81. 将第 C.4 章中图“AP_1b(3)”替换为：

82. 将第 C.4 章中图“AP_signals_2d(3)”替换为：

```
use macsorts,
 use macmib;
```

System Access _Point　　AP_signals _2d(3)

```
newtype DsStatus literals
 assoc, disassoc , reassoc , unknown
endnewtype DsStatus ;
```

```
signal
 AsChange(Frame,DsStatus),
 Backoff(Integer,Integer),
 BkDone(Integer),
 Busy,
 Cancel,
 ChangeNav(Time,Duration,NavSrc),
 Dsinquiry(MacAddr,MacAddr),
 DsNotify(MacAddr,DsStatus),
 DsResponse(MacAddr,MacAddr,DsStatus),
 FromDsm(MacAddr,Octetstring),
 Idle,
 MaUnitdata,indication(MacAddr,MacAddr,
  Routing,OctetstringRxStatus,
  CfPriority,ServiceClass),
 MaUnitdata,request(MacAddr,MacAddr,
  Routing,OctetstringCfPriority,ServiceClass),
 MaUnitdataStatusindication(MacAddr,
  MacAddr,TxStatus,CfPriority,ServiceClass),
 MlmeAssociate.confirm(MlmeStatus),
 MlmeAssociate.indication(MacAddr),
 MlmeAssociate.request(MacAddr,Kusec,Capability,Integer),
 MlmeAuthenticate,confirm
  (McaAddr,AuthenticateType,MlmeStatus),
 MlmeAuthenticate.indication(MacAddr,AuthenticateType),
 MlmeAuthenticate,requesst(MacAddr,AuthenticateType,Kusec),
 MlmeDeAuthenticate.confirm(MacAddr,MlmeStatus),
 MlmeDeAuthenticate.indication(MacAddr,ReasonCode),
 MlmeDeAuthenticate.request(MacAddr,ReasonCode),
 MlmeDisassociate.confirm(MlmeStatus),
 MlmeDisassciate.indication(MacAddr,ReasonCode),
 MlmeDisassociate.request(MacAddr,ReasonCode),
 MlmeGet.confirm(MibStatus,MibAtrlb,MlbValue),
 MlmeGet.request(MibAtrlb),
 MlmeJoin.confirm(MlmeStatus),
 MlmeJoin.request(BssDscr,Integer,Usec,Ratestring),
 MlmePowermgt.confirm(MlmeStatus),
 MlmePowermgt.request(PwSave,Boolean,Boolean),
 MlmeReassociate.confirm(MlmeStatus),
 MlmeReassociate.indication(MacAddr),
 MlmeReassociate.request(MacAddr,Kusec,Capability,Integer),
 MlmeReset.confirm(MlmeStatus),
 MlmeTeset.request,
 MlmeScanconfirm(BssDscrSet,mlmeStatus),
 MlmeScanrequest(BssTypeSet,MacAddr,Octetstring,
  ScanType,Usec,Intstring,Kusec,Kusec),
 MlmeSet confirm(MibStatus,Mibtrib),
 MlmeSet request(MibAtrib,MibValue),
 MlmeStart.confirm(MLmeStatus),
 MlmeStart.request( Octetstring,BssType,Kusec,
  Integer,CfParms,PhyParms,IbssParms,Usec,
  Capability,Ratestring,Ratestring);
```

```
Signal
 MmCancel,
 MmConfirm(Frame,TxStatus),
 MmIndicate(Frame,Time,Time,StateErr),
 MmRequest (Frame,Imed,Rate),
 MsduConfirm (Frame,CfPriority,
  TxStatus),
 MsduIndicate (Frame, ,CfPriority,
  TxStatus),
 MsduRequest (Frame,CfPriority),
 NeedAck(MacAddr,Time,Duration,Rate),
 PduConfirm(FragSdu,TxResult),
 PduRequest(FragSdu),
 PhyCca.indication(Ccastatus),
 PhyCcarst.confirm,
 PhyCcarst.request,
 PhyData.confirm,
 PhyData.indiction(Octet),
 PhyDatarequest(Octet),
 PhyRxEnd.indication(PhyRxStat),
 PhyRxStart.indication(Integer,Rate),
 PhyTxEnd.confirm,
 PhyTxEnd.request,
 PhyTxStart.confirm,
 PhyTxStart.requesst(Integer,Rate),
 PlmeCharacteristicsconfirm(PhyChrstcs),
 PlmeCharacteristicsrequest,
 PlmeGet.confirm(MibStatus,
  MibAtrib,MibValue),
 PlmeGet.requesssst(MibAtrib),
 PlmeResetconfrm(Boolean),
 PlmeResetrequest,
 PlmeSet.confirm(MibStatus,MibAtrib),
 PlmeSet.request(MibAtrib,MibValue),
 Psmdone,
 PsPolled(MacAddr,AsocId),
 PsChange (MacAddr,PsMode),
 Psindicate(MacAddr,PsMode),
 Psinquiry (MacAddr),
 PsResponse (MacAddr,PsMode),
 ResetMAC,
 RxCfAcd(MacAddr),
 RxIndicate(Frame,Time,Time,Rate),
 Slot,
 SsInquiry (MacAddr),
 SsResponse (MacAddr,
  StationState,StationState),
 SwChnl( Integer,Boolean),
 SwDone,
 ToDsm(MacAddr,MacAddr,Octetstring),
 TxConfirm,
 TxRequest(Frame,Rate);
```

83. 将第 C.4 章中图“AP_signallists_3b(13)”替换为：

84. 将第 C.4 章中图“ap_MPDU_ gen_1a(1)”替换为：

85. 将第 C.4 章中图“prepare_1b(2)”替换为：

86. 将第 C.4 章中图“fragment_2b(2)”替换为：

87. 将第 C.4 章中图“rx_coord_2b(4)”替换为：

88. 将第 C.4 章中图“rx_coord_3b(4)”替换为：

89. 将第 C.4 章中图“rx_coord_3.1a(4)”替换为：

90. 将第 C.4 章中图“Mac_Mgmt_1a(1)”替换为：

91. 将第 C.4 章中图“Mlme_indicate_1a(1)”替换为：

92. 将第 C.4 章中图“Mib_access_1a(2)”替换为：

93. 将第 C.4 章中图“Mib_import_export_2b(2)”替换为：

Process MIB

Mib_import_export_2b(2)

```
/*导入从其他过程输出
的MIB（只读）计数器值*/
imported
 dot11AckFailureCount,
 dot11FailedCount,
 dot11FcsErrorCount,
 dot11FrameDuplicateCount,
 dot11MulticateReceivedFrameCount,
 dot11MulticastTransmittedFrameCount,
 dot11MultipleRetryCount,
 dot11ReceivedFragmentCount,
 dot11RetryCount,
 dot11RtsFailureCount,
 dot11RtsSuccessCount,
 dot11TransmittedFragmentCount,
```

```
/*从本过程输出的MIB属性的声明*/


/*读写属性    */
dcl exported
  dot11AuthenticateAlgorithms LinkVerfyTypeSet:=
    incl(open_system),
  dot11FragmentationThteshold Integer:=2346,
  dot11GroupAddresses MacAddrSet: =empty,
  dot11LongRetryLimit Integer: =4,
  dot11MaxReceiveLifetime Kusec: =512,
  dot11MaxTransmitMsduLifetime Kusec:=512,
  dot11MediumOccupancyLimit Kusec: =100,
  mReceiveDTIMs Boolean:=true,
  dot11CfpPeriod Integer: =1,
  dot11CfpMaxDuration Kusec; =200,
  dot11AuthenticateResponseTimeout Kusec:=512,
  dot11RtsThreshold Integer: =3000,
  dot11ShortRetryLimit Integer: =7,
  dot11CurrentChannelNumber Integer:=0,
  dot11CurrentSet Ingeter: =0,
  dot11CurrentPattern Ingeter:=0,
  dot11CurrentIndex Ingeter: =0;
```

```
/*以下的MAC MIB只读属性定义为同义词 （命名常量），
而不是远程变量 ，因为它们描述的是静止站的属性 ，
至少在MAC操作的任意单一实例期间是固定的 。

 dot11AuthenticateAlgorithms AuthenticateTypeSet
 dot11CfPollable Boolean,
 dot11MacAddress MacAddr,
 dot11ManufacturerID Octetstring,
 dot11ProductID Octetstring,
 aStationID MacAddr,

另外，MAC访问的PHY MIB中所有的只读属性定义
为同义词
*/
```

```
/*注意

用于MAC MIB属性的列表值是那些属性的规定默认值 。
用于PHY MIB属性的列表值或者是 FH PHY的默认值，
或者是规定范围内的任意值 。在MAC的SDL描述中，
用于PHY属性的特定值没有标准的有效性
*/
```

94. 将第 C.4 章中图“Mlme_request_1b(3)”替换为：

95. 将第 C.4 章中图“Mlme_request_3a(3)”替换为：

96. 将第 C.4 章中图“ap_MLME_1a(1)”替换为：

97. 将第 C.4 章中图“ps_monitor_1a(2)”替换为：

98. 将第 C.4 章中图“ps_monitor_2a(2)”替换为：

99. 将第 C.4 章中图“ap_Mm_svc_1b(1)”替换为：

100. 将第 C.4 章中图“ap_auth_req_1a(1)”替换为：

101. 在第 C.4 章中图“Auth_req_1a(2)”之后，插入图“auth_rsp_1b(2)”和图“auth_rsp_2b(2)”：

Service AuthRspService
auth_rsp_2b(2)
*
Deauth
(arRdu)
arSta3:=
addr2
(arRdu)
Sst(arSta3,
de_auth)
更新站的状态，
撤销链路验证
存在的关联
MlmeAuthenticate
_indication
(arSta3,
reason
(arRdu))
arSta3=
mBssid
若撤销链路验证来
自当前AP，则请除
mBssid 和mAssoc
而结束关联
(false)
(true)
mAssoc:=false,
mBsid:=
nullAddr
Xport
-

102. 将第 C.4 章中图“mmpdu_svc_1.1b(2)”替换为：

103. 将第 C.4 章中图“ap_init_1b(3)”替换为：

104. 将第 C.4 章中图“AP_Start_Bss_3c(8)”替换为：

105. 将第 C.4 章中图“receive_1a(1)”替换为：

106. 将第 C.4 章中图“nav_clear_1b(2)”替换为：

107. 将第 C.4 章中图“start_rx_1b(2)”替换为：

108. 将第 C.4 章中图“Validate_rx_2c(2)”替换为：

109. 将第 C.4 章中图“pre_filter_1d(4)”替换为：

110. 将第 C.4 章中图“filter_sta_2b(4)”替换为：

111. 将第 C.4 章中图“report_rx_4a(4)”替换为：

112. 在第 C.4 章中图“report_rx_4a(4)”之后，插入图“filter_1b(3)”：

113. 将第 C.4 章中图“degragment_3c(3)”替换为：

114. 附录D的内容替换为：

```
-- ********************************
-- *  IEEE802dot11-MIB 定义开始
-- ********************************
IEEE802dot11-MIB DEFINITION  ::=BEGIN
    IMPORTS
        MODULE-IDENTITY,OBJECT-TYPE,
        NOTIFICATION-TYPE,integer32,counter32      FROM SNMPv2-SMI
        DisplayString,MacAddress,RowStatus,
        TruthValue                                 FROM SNMPv2-TC
        MODULE-COMPLIANCE,OBJECT-GROUP,
        NOTIFICATION-GROUP                         FROM SNMPv2-CONF
        ifIndex                                    FROM RFC1213-MIB;

-- ***********************************************
-- *  模块标识
-- ***********************************************
    ieee802dot11 MODULE-IDENTITY
    LAST-UPDATED "9807080000Z"
    ORGANIZATION "IEEE 802.11"
    CONTACT-INFO
        " WG E-mail: stds-802-11@ieee.org
        Chair: Stuart J. Kerry
        Postal: Philips Semiconductors, Inc.
                1109 McKay Drive
                M/S 48 SJ
                San Jose, CA 95130-1706 USA
        Tel: +1 408 474 7356
        Fax: +1 408 474 7247
        E-mail: stuart.kerry@philips.com
        Editor: Terry L. Cole
        Postal: AMD, M/S PCS4
                5900 E. Ben White Blvd.
                Austin, TX 78741 USA
        Tel: +1 512 602 2454
        Fax: +1 512 602 5051
        E-mail: terry.cole@amd.com "

    DESCRIPTION
        "本规范实体的 MIB 模块。iso(1).member-body(2).us(840).ieee802dot11(10036)"。
          本规范对 IEEE802dot11-MIB 进行如下修改：
            -- 将 dot11WEPDefaultKeysTable 表及其所有子项、dot11WEPKeyMappingsTable 表
```

及其所有子项、dot11PrivacyTable 表及其所有子项、dot11CountersTable 表的 dot11WEPUndecryptableCount 子项、dot11StationConfigTable 表的 dot11PrivacyOptionImplemented 子项的 STATUS 修改为 deprecated。

—— 将 dot11AuthenticationAlgorithmsTable 表的 dot11AuthenticationAlgorithms 项修改为只取 1 值，即仅支持'开放式系统'链路验证。"

::={1 2 840 10036 }

```
-- ************************************
-- *  主体部分
-- ************************************
-- 站管理(SMT)属性
  -- 定义为"SMT 对象类在 STA 上为管理 STA 内的进程提供必要的支持，
  -- 以使 STA 可以作为本规范网络的一部分而协调工作。"
  dot11smt OBJECT IDENTIFIER                          ::={ieee802dot11 1}
  -- dot11smt GROUPS
  -- dot11StationConfigTable                          ::= { dot11smt 1 }
  -- dot11AuthenticationAlgorithmsTable               ::= { dot11smt 2 }
  -- dot11WEPDefaultKeysTable                         ::= { dot11smt 3 }
  -- dot11WEPKeyMappingsTable                         ::= { dot11smt 4 }
  -- dot11PrivacyTable                                ::= { dot11smt 5 }
  -- dot11SMTnotification                             ::= { dot11smt 6 }

-- MAC 属性
  -- 定义为"MAC 对象类为访入控制、帧检验序列的生成和验证和向高层正确地传送有效数据提
     供必要的支持。"
  dot11mac OBJECT IDENTIFIER                          ::={ieee802dot11 2}
  -- MAC GROUPS
  -- 参考 IEEE Std 802.1F-1993
  -- dot11OperationTable                              ::={dot11mac 1}
  -- dot11CountersTable                               ::={dot11mac 2}
  -- dot11GroupAddressesTable                         ::={dot11mac 3}

-- Resource Type ID
  dot11resOBJECT IDENTIFIER                           ::={ieee802dot11 3}
  dot11resAttribute OBJECT IDENTIFIER                 ::={dot11res 1}

-- PHY 属性
  -- 定义为"PHY 对象类向要求的 PHY 操作信息提供必要的支持，这与高层通信的信息可能会
     在 PHY 之间或 STA 之间有所变化。"
  dot11phy OBJECT IDENTIFIER                          ::={ieee802dot11 4}
  -- phy GROUPS
  -- dot11PhyOperationTable                           ::={dot11phy 1}
  -- dot11PhyAntennaTable                             ::={dot11phy 2}
```

```
--  dot11PhyTxPowerTable                    ::={dot11phy 3}
--  dot11PhyFHSSTable                       ::={dot11phy 4}
--  dot11PhyDSSSTable                       ::={dot11phy 5}
--  dot11PhyIRTable                         ::={dot11phy 6}
--  dot11RegDomainsSupportedTable           ::={dot11phy 7}
--  dot11AntennasListTable                  ::={dot11phy 8}
--  dot11SupportedDataRatesTxTable          ::={dot11phy 9}
--  dot11SupportedDataRatesRxTable          ::={dot11phy10}

WEPKeytype ::= OCTET STRING (SIZE (5))

--  ****************************************************************************
--  *  MIB 属性 OBJECT-TYPE 定义如下
--  ****************************************************************************
--  ****************************************************************************
--  *  SMT 站配置表开始
--  ****************************************************************************
dot11StationConfigTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11StationConfigEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
                "站配置属性,以表格形式允许一个代理处有多个实例。"
    ::={ dot11smt 1 }

dot11StationConfigEntry OBJECT-TYPE
        SYNTAX Dot11StationConfigEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "dot11StationConfigTable 的一个入口。有可能一个代理处有多个接口,每个接口有惟一
            的 MAC 地址。本规范接口和 Internet 标准 MIB 文中的接口的关系是一一对应的,同样
            地,ifIndex 对象实例值能直接用于标识此处定义的对象的相应实例。
            ifIndex—每一个本规范接口由一个 ifEntry 来表示。本 MIB 模块中的接口表由 ifIndex 来
            索引。"
        INDEX{ifIndex}
    ::={ dot11StationConfigTable 1 }

Dot11StationConfigEntry ::=
        SEQUENCE {
            dot11StationID                      MacAddress,
            dot11MediumOccupancyLimit           INTEGER,
            dot11CFPollable                     TruthValue,
```

```
        dot11CFPPeriod                          INTEGER,
        dot11CFPMaxDuration                     INTEGER,
        dot11AuthenticationResponseTimeout      INTEGER,
        dot11PrivacyOptionImplemented           TruthValue,
        dot11PowerManagementMode                INTEGER,
        dot11DesiredSSID                        OCTET STRING,
        dot11DesiredBSSType                     INTEGER,
        dot11OperationalRateSet                 OCTET STRING,
        dot11BeaconPeriod                       INTEGER,
        dot11DTIMPeriod                         INTEGER,
        dot11AssociationResponseTimeOut         INTEGER,
        dot11DisassociateReason                 INTEGER,
        dot11DisassociateStation                MacAddress,
        dot11DeauthenticationReason             INTEGER,
        dot11DeauthenticationStation            MacAddress,
        dot11AuthenticationFailStatus           INTEGER,
        dot11AuthenticationFailStation          MacAddress}

dot11StationID OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-write
    STATUS deprecated
    DESCRIPTION
        "dot11StationID 的目的是允许管理者按自己的意图标识 STA。当保持真实的 MAC 地址
        相互独立时,该属性提供了可能性。它的语法是 MacAddress。默认值是 STA 已指定的、
        惟一的 MAC 地址。"
  ::={ dot11StationConfigEntry 1 }

dot11MediumOccupancyLimit OBJECT-TYPE
    SYNTAX INTEGER (0..1000)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "该属性应指示以 TU 为单位的最大的时间值,点协调器可以在该时间值内控制无线媒体的
        使用而不放弃控制以有足够长的时间允许至少一个 DCF 实例访问媒体。该属性的默认值
        应为 100,最大值应为 1000。"
  ::={dot11StationConfigEntry 2}

dot11CFPollable OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

"当该属性值为真时，它应指示 STA 能够在 SIFS 时间内以数据帧响应 CF-Poll。如果 STA 不能在 SIFS 时间内以数据帧响应 CF_Poll，该属性值应为假。"

::={dot11StationConfigEntry 3}

dot11CFPPeriod OBJECT-TYPE
SYNTAX INTEGER(0..225)
MAX-ACCESS read-wirte
STATUS current.
DESCRIPTION
"该属性应描述在 CFP 开始时刻之间的 DTIM 间隔的数量，该值由原语 MLME-START. request 修改。"

::={ dot11StationConfigEntry 4 }

dot11CFPMaxDuration OBJECT-TYPE
SYNTAX INTEGER(0..65535)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"该属性应以 TU 为单位描述 CFP 的最大持续时间，CFP 可由 PCF 生成。属性值由原语 MLME-START. request 修改。"

::={ dot11StationConfigEntry 5 }

dot11AuthenticationResponseTimeOut OBJECT-TYPE
SYNTAX INTEGER (1..4294967295)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"该属性应规定在链路验证序列中响应方 STA 等待下一帧到来之前的 TU 数。"

::={ dot11StationConfigEntry 6 }

dot11PrivacyOptionImplemented OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS deprecated
DESCRIPTION
"该属性取值为真，指示实现了 WEP 机制。该属性默认值为假。"

::= { dot11StationConfigEntry 7 }

dot11PowerManagementMode OBJECT-TYPE
SYNTAX INTEGER { active(1), powersave(2) }
MAX-ACCESS read-write
STATUS current

DESCRIPTION

"该属性应规定 STA 的功率管理模式。当设置为 active 时,表明 STA 不在节能模式。下;当设为 powersave 时,指示 STA 处在节能模式下。功率管理模式根据 7.1.3.1.7 在所有帧中发送。"

::={ dot11StationConfigEntry 8 }

dot11DesiredSSID OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..32))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"该属性反映了用在最近的的原语 MLME_Scan. request 的 DesiredSSID 参数中使用的服务集 ID。该值可由外部管理实体修改并被本地 SME 使用来对扫描过程作出判定。"

::={dot11StationConfigEntry 9 }

dot11desiredBSSType OBJECT_TYPE

SYNTAX INTEGER { infrastructure(1), independent(2), any(3)}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"该属性规定当扫描用于 BSS 同步时,STA 使用的 BSS 类型。其值用于过滤探询响应帧和信标。当设为 infrastructure 时,该 STAinfrastructure 应仅同能力信息字段的 ESS 子字段置为 1 的 BSS 同步。当设为 independent 时,该 STA 应仅同同能力信息字段的 IBSS 子字段置为 1 的 BSS 同步。当设置为 any 时,该 STA 与以上两种任一类型的 BSS 同步。"

::={ dot11StationConfigEntry 10 }

dot11OperationalRateSet OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(1..126))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"该属性应规定 STA 可用于发送数据的数据速率集合。每个八位位组包含一个代表速率的值。每个速率的取值范围为 2 到 127,相对应的数据速率 1Mbit/s 变化到 63.5Mbit/s 以 500kb/s 递增,并支持以该速率(如支持的速率表中指示的)接收数据。该值在发送的信标、探询请求、探询响应、关联请求、关联响应、以及重新关联请求和重新关联响应帧中报告,并用于确定 STA 期望同步的 BSS 是否合适。它还在 BSS 启动时使用,在 10.3 中规定。"

::={ dot11StationConfigEntry 11 }

dot11BeaconPeriod OBJECT-TYPE

SYNTAX INTEGER (1..65535)

MAX-ACCESS read-write

STATUS current
DESCRIPTION
"该属性应规定 STA 将用于给发送信标排序所用的 TU 数量。该值在信标和探询响应帧中发送。"
::={ dot11StationConfigEntry 12 }

dot11DTIMPeriod OBJECT-TYPE
SYNTAX INTEGER(1..255)
MAX-ACCESS read-write
STATUS current
DESCRIPTION11
"该属性应规定在包含 DTIM 计数字段为 0 的 TIM 元素的信标帧的发送之间流逝信标时间间隔的数量。该值在信标帧的 DTIM 周期字段中发送。"
::={ dot11StationConfigEntry 13 }

dot11AssociationResponseTimeOut OBJECT-TYPE
SYNTAX INTEGER(1..4294967295)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"该属性规定请求方 STA 为已发送的关联请求 MMPDU 响应所要等待的 TU 数量。"
::={dot11StationConfigEntry 14 }

dot11DisassociateReason OBJECT-TYPE
SYNTAX INTEGER(0..65535)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"该属性保持最近发送的解除关联帧中的原因代码。如果没有解除关联帧曾被发送,该属性值应为 0。"
REFEREMCE "本规范 7.3.1.7"
::={ dot11StationConfigEntry 15}

dot11DisassociateStation OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"该属性保持来自最近发送的解除关联帧中的地址 1 字段中的 MAC 地址。若没有解除关联帧曾被发送,该属性值应为 0。"
::={ dot11StationConfigEntry 16 }

dot11DeauthenticationReason OBJECT-TYPE

```
        SYNTAX INTEGER (0..65535)
        MAXACCESS read-only
        STATUS current
        DESCRIPTION
                "该属性保持最近发送的解除链路验证帧中的原因代码。如果没有解除链路验证帧曾被
                发送,该属性的值应为 0。"
        REFERENCE   "本规范 7.3.1.7"
    ::={ dot11StationConfigEntry 17}

dot11DeauthenticationStation OBJECT-TYPE
        SYNTAX MacAddress
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "该属性保持来自最近发送的解除链路验证帧中的地址 1 字段的 MAC 地址。若没有解
                除链路验证帧曾被发送,该属性值应为 0。"
    ::={ dot11StationConfigEntry 18 }

dot11AuthenticationFailStatus OBJECT-TYPE
        SYNTAX INTEGER(0..65535)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
                "该属性保持最近的失败的链路验证帧中的状态代码。若没有失败的链路验证帧曾发
                送,该属性值应为 0。"
        REFERENCE "本规范 7.3.1.9"
    ::={dot11StationConfigEntry 19 }

dot11AuthenticationFailStation OBJECT-TYPE
        SYNTAX MacAddress
        MAC-ACCESS read-only
        STATUS current
        DESCRIPTION
                "该属性保持来自最近发送的失败的链路验证帧中的地址 1 字段中的 MAC 地址。若没有
                失败的链路验证帧曾被发送,该属性值应为 0。"
    ::={dot11StationConfigEntry 20}
-- ******************************************************************************
-- *    dot11StationConfig 表结束
-- ******************************************************************************

-- ******************************************************************************
-- * AuthenticationAlgorithms 链路验证算法表开始
-- ******************************************************************************
```

```
dot11AuthenticationAlgorithmsTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot11AuthenticationAlgorithmsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "该(概念上的)属性表为站支持的链路验证算法的集合。下面为默认的值与对应的算法。
        取值为 1,表示开放式系统链路验证;取值为 2,表示共享密钥链路验证。符合本规范的设
        备只支持开放式系统链路验证。"
    REFERENCE "本规范 7.3.1.1"
  ::= { dot11smt 2 }

dot11AuthenticationAlgorithmsEntry OBJECT-TYPE
    SYNTAX Dot11AuthenticationAlgorithmsEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "链路验证算法表项。
          ifIndex—每个接口由一个 ifEntry 描述,该 MIB 模块中的接口表由 ifIndex 来索引。"
    INDEX { ifIndex,dot11AuthenticationAlgorithmsIndex }
  ::= { dot11AuthenticationAlgorithmsTable 1 }

Dot11AuthenticationAlgorithmsEntry ::=
    SEQUENCE {
    dot11AuthenticationAlgorithmsIndex      Integer32,
    dot11AuthenticationAlgorithm              INTEGER,
    dot11AuthenticationAlgorithmsEnable      TruthValue }

dot11AuthenticationAlgorithmsIndex OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "用来标识链路验证算法表中纵列上对象实例的辅助变量。"
  ::= { dot11AuthenticationAlgorithmsEntry 1 }

dot11AuthenticationAlgorithm OBJECT-TYPE
    SYNTAX INTEGER { openSystem(1),sharedKey(2) }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "该属性表为站支持的链路验证算法的集合。下面为默认的值与对应的算法。取值为 1,表
        示开放式系统链路验证;取值为 2,表示共享密钥链路验证。符合本规范的设备只支持开放
        式系统链路验证。"
```

```
    ::= { dot11AuthenticationAlgorithmsEntry 2 }

dot11AuthenticationAlgorithmsEnable OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
                "该属性为真,使能站接收的具有奇数序列号的链路验证帧中描述的相应表项的链路验证
                算法。该属性的缺省值:对于开放式系统链路验证表项,值为 1;对于其他表项,值为 2。
                符合本规范的设备只支持开放式系统链路验证。"
    ::= { dot11AuthenticationAlgorithmsEntry 3 }
-- ***********************************************************************
-- * AuthenticationAlgorithms 链路验证算法表结束
-- ***********************************************************************

-- ****************************************************************************
-- *   WEPDefaultKeys WEP 缺省密钥表开始
-- ****************************************************************************
dot11WEPDefaultKeysTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11WEPDefaultKeysEntry
        MAX-ACCESS not-accessible
        STATUS deprecated
        DESCRIPTION
            "WEP 缺省密钥的概念表。该表包含与四种可能的 keyID 值对应的四个 WEP 缺省密钥
            的值,WEP 缺省密钥在逻辑上为"只写"。若试图读表项的值,会返回不成功状态和空值
            或 0 值,每个 WEP 缺省密钥的缺省值为空。"
        REFERENCE " ISO/IEC 8802-11:1999,8.3.2"
   ::={ dot11smt 3 }

dot11WEPDefaultKeysEntry OBJECT-TYPE
        SYNTAX Dot11WEPDefaultKeysEntry
        MAX-ACCESS not-accessible
        STATUS deprecated
        DESCRIPTION
            "WEP 缺省密钥表中的一个表项(概念上的行)。
            IfIndex —每个标准接口由一个 ifEntry 代表。在 MIB 模块中的接口表由 ifIndex 索引。"
                    INDEX{ifIndex,dot11WEPDefaultKeyIndex}
    ::={ dot11WEPDefaultKeysTable 1 }

Dot11WEPDefaultKeysEntry ::=SEQUENCE{
            dot11WEPDefaultKeyIndex INTEGER
            dot11WEPDefaultKeyValue WEPKeytype}
```

```
dot11WEPDefaultKeyIndex OBJECT-TYPE
        SYNTAX INTEGER(1..4)
        MAX-ACCESS not-accessible
        STATUS deprecated
        DESCRIPTION
            "辅助变量,用于识别 WEP 缺省密钥表中的纵列对象实例。该变量的值等于 WEPDe-
            faultKeyID 加 1。"
   ::={dot11WEPDefaultKeysEntry 1  }

dot11WEPDefaultKeyValue OBJECT-TYPE
            SYNTAX WEPKeytype
            MAX-ACCESS read-write
            STATUS deprecated
            DESCRIPTION
            "WEP 缺省密钥值。"
   ::= { dot11WEPDefaultKeysEntry 2 }
-- ******************************************************************************
-- * WEP 缺省密钥表结束
-- ******************************************************************************

-- ******************************************************************************
-- * WEPKeyMappings 密钥映射表开始
-- ******************************************************************************
dot11WEPKeyMappingsTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11WEPKeyMappingsEntry
        MAX-ACCESS not-accessible
        STATUS deprecated
        DESCRIPTION
            "WEP 密钥映射的概念表。MIB 支持每对 RA/TA 共享一个独立 WEP 密钥的能力。
            密钥映射表对每个 MAC 地址包括 0 个或一个表项,对每个表项包含 2 个字段:WEPOn
            和相应的 WEP 密钥。WEP 密钥映射在逻辑上为'只写'。若试图在该表中读取表项,
            会返回不成功状态和空值或 0 值。所有 WEPOn 字段的缺省值为假。"
        REFERENCE "ISO/IEC 8802-11:1999,8.3.2"
     ::={ dot11smt 4 }

dot11WEPKeyMappingsEntry OBJECT-TYPE
        SYNTAX   Dot11WEPKeyMappingsEntry
        MAX-ACCESS   not-accessible
        STATUS   deprecated
        DESCRIPTION
            "WEP 密钥映射表的一个表项。
            ifIndex—每个标准接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
        INDEX {ifIndex,dot11WEPKeyMappingIndex}
```

```
    ::={ dot11WEPKeyMappingsTable   1 }

Dot11WEpKeyMappingsEntry::=SEQUENCE{
            dot11WEPKeyMappingIndex          Integer32,
            dot11WEPKeyMappingAddress        MacAddress,
            dot11WEPKeyMappingWEPOn          TruthValue,
            dot11WEPKeyMappingValue          WEPKeytype,
            dot11WEPKeyMappingStatus         RowStatus}

dot11WEPKeyMappingIndex OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS not-accessible
        STATUS deprecated
        DESCRIPTION
            "辅助变量,用于识别 WEP 密钥映射表中的纵列对象实例。"
    ::={ dot11WEPKeyMappingsEntry   1 }

dot11WEPKeyMappingAddress OBJECT-TYPE
        SYNTAX MacAddress
        MAX-ACCESS read-create
        STATUS deprecated
        DESCRIPTION
            "使用本密钥映射表项的值的 STA 的 MAC 地址。"
    ::={  dot11WEPKeyMappingEntry   2 }

dot11WEPKeyMappingWEPOn   OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-creat
        SATUS deprecated
        DESCRIPTION
            "关于在同 dot11WEPKeyMappingAddress STA 进行通信时是否使用 WEP 的布
            尔值。"
    ::={  dot11WEPKeyMappingEntry   3 }

dot11WEPKeyMappingValue   OBJECT-TYPE
        SYNTAX WEPKeytype
       MAX-ACCESS read-create
       STATUS deprecated
       DESCRIPTION
           "一个 WEP 密钥值。"
    ::={  dot11WEPKeyMappingEntry   4 }

dot11WEPKeyMappingStatus   OBJECT-TYPE
```

```
        SYNTAX RowStatus
        MAX-ACCESS read-create
        STATUS  deprecated
        DESCRIPTION
            "状态栏,用于创建、修改和删除 WEP 密钥映射表中的纵列对象实例。"
        DEFVAL{active}
    ::={  dot11WEPKeyMappingsEntry  5 }

-- ***********************************************************************
-- *        WEPKeyMappings 密钥映射表结束
-- ***********************************************************************

-- ***********************************************************************
-- *        dot11PrivacyTable 表开始
-- ***********************************************************************
dot11PrivacyTable  OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PrivacyEntry
        MAX-ACCESS not-accessible
        STATUS deprecated
        DESCRIPTION
            "包含涉及到 ISO/IEC 8802-11 保密性的属性的组。以表格形式产生,允许在一个代理上
            有多种实现形式。"
    ::={  dot11smt  5 }

dot11PrivacyEntry  OBJECT-TYPE
        SYNTAX Dot11PrivacyEntry
        MAX-ACCESS not-accessible
        STATUS deprecated
        DESCRIPYION
            "dot11PrivacyTable 中的一个表项。
            ifIndex—每个标准接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
        INDEX{ifIndex}
   ::={  dot11PrivacyTable  1 }

Dot11PrivacyEntry  ::= SEQUENCE {
            dot11PrivacyInvoked                TruthValue,
            dot11WEPDefaultKeyID               INTEGER,
            dot11WEPKeyMappingLength           INTEGER,
            dot11ExcludeUnencrypted            TruthValue,
            dot11WEPICVErrorCount              Counter32,
            dot11WEPExcludedCount              Counter32}

dot11PrivacyInvoked  OBJECT-TYPE
```

```
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS deprecated
        DESCRIPTION
            "当该属性为真时,表示 WEP 机制用于发送数据类型帧。该属性的缺省值为假。"
    ::={  dot11PrivacyEntry  1 }

dot11WEPDefaultKeyID  OBJECT-TYPE
        SYNTAX  INTEGER (0..3)
        MAX-ACCESS read-write
        STATUS deprecated
        DESCRIPTION
            "当属性值设置为 0、1、2 或 3 时,该属性指示使用 WEP 缺省密钥队列中的第 1、2、3 或
            4 个元素。该属性的缺省值为 0。"
        REFERENCE "ISO/IEC 8802-11:1999,8.3.2"
    ::={  dot11PrivacyEntry  2 }

dot11WEPKeyMappingLength  OBJECT-TYPE
        SYNTAX INTEGER (10..4294967295)
        MAX-ACCESS read-write
        STATUS deprecated
        DESCRIPTION
            "dot11WEPKeyMapping 可保持的单元的最大数量。"
    ::={  dot11PrivacyEntry 3 }

dot11ExcludeUnencrypted  OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS deprecated
        DESCRIPTION
            "当该属性为真时,STA 不在 MAC 服务接口上指示接收到的帧控制字段的 WEP 子字段
            值为 0 的 MSDU。当该属性为假时,STA 可以接收帧控制字段的 WEP 子字段为 0 的
            MSDU。该属性的缺省值为假。"
    ::={  dot11PrivacyEntry  4 }

dot11WEPICVErrorCount  OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS deprecated
        DESCRIPTION
            "当接收到一帧,而该帧的帧控制字段的 WEP 子字段设为 1 并且该帧中的 ICV 值与根据
            该帧内容计算得到的 ICV 值不匹配,计数器增加计数。"
    ::={  dot11PrivacyEntry  5 }
```

```
dot11WEPExcludedCount   OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS deprecated
        DESCRIPTION
                "当接收到一帧，而该帧的帧控制字段的 WEP 子字段值设为 0 并且
                dot11ExcludeUnencrypted 的值导致该帧将要被丢弃时，计数器增加计数。"
    ::={   dot11PrivacyEntry   6 }
-- ******************************************************************************
-- *        dot11Privacy 表结束
-- ******************************************************************************

-- ******************************************************************************
-- *    SMT 通告对象开始
-- ******************************************************************************
  dot11SMTnotification OBJECT INDENTIFIER ::={ dot11smt 6  }

  dot11Disassociate NOTIFICATION-TYPE
        OBJECTS { ifIndex, dot11DisassciateReason,dot11DisassociateStation }
        STATUS   current
        DESCRIPTION
            "当 STA 发送解除关联帧时，解除关联通告应被发送。通告的值应包括解除关联帧的目
            的 MAC  的 MAC 地址及解除关联的原因。
            ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
    ::={   dot11SMTnotification 0 1 }

dot11Deauthentication NOTIFICATION-TYPE
        OBJECTS {ifIndex,dot11DeauthenticationReason, dot11DeauthenticationFailStation }
        STATUS current
        DESCRIPTION
            "当 STA 发送解除链路验证帧时，解除链路验证通告应被发送。通告的值应包括解除链
            路验证帧的目的 MAC 的 MAC 地址及解除链路验证的原因。
            ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
    ::={   dot11SMTnotification 0 2 }

dot11AuthenticationFail NOTIFICATION-TYPE
        OBJECTS{ ifIndex,dot11AuthenticationFailStatus, dot11AuthenticationFailStation }
        STATUS current
        DESCRIPTION
            "当 STA 发送状态码不是'成功'的链路验证帧时，链路验证失败通告应被发送。通告的
            值包括链路验证帧的目的 MAC 的 MAC 地址和链路验证失败的原因。
            ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
```

```
    ::={   dot11SMTnotification 0 3 }

-- ****************************************************************************
-- *        SMT 通告对象结束
-- ****************************************************************************

-- ****************************************************************************
-- *        MAC 属性模板
-- ****************************************************************************
-- ****************************************************************************
-- *   dot11OperationTable 表开始
-- ****************************************************************************
dot11OprationTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11OperationEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "包括关于 MAC 操作的 MAC 属性组。它以表格的形式实现,以允许在一个代理上有多
            个实例。"
    ::={   dot11mac 1 }

dot11OperationEntry OBJECT-TYPE
        SYNTAX Dot11OperationEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "dot11OperationEntry 表中的一个项目。
            ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
    ::={   dot11OperationTable 1 }

 Dot11OperationEntry ::= SEQUENCE {
            dot11MACAddress                    MacAddress,
            dot11RTSThreshold                  INTEGER,
            dot11ShortRetryLimit               INTEGER,
            dot11LongRetryLimit                INTEGER,
            dot11FragmentationThresdhold       INTEGER,
            dot11MaxTransmitMSDULifetime       INTEGER,
            dot11MaxReceiveLifeTime            INTEGER,
            dot11ManufacturerID                DisplayString,
            dot11ProductID                     DisplayString}

dot11MACAddress OBJECT-TYPE
        SYNTAX MacAddress
```

```
        MAC-ACCESS read-only
        STATUS current
        DESCRIPTION
            "分配给 STA 的惟一的 MAC 地址。"
    ::={  dot11OperationEntry 1 }

dot11RTSThreshold OBJECT-TYPE
        SYNTAX INTEGER (0..2347)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "该属性应指示 MPDU 中的八位位组数,小于该值(MPDU)将不执行 RTS/CTS 握手。
            当 MPDU 是数据帧或管理帧,MPDU 在地址 1 字段是一个单个地址,并且 MPDU 的长
            度大于该阈值,RTS/CTS 握手应在帧交换序列的开始时刻执行(额外的细节,参考 9.7
            中的表 21)。将此属性设置为比 MSDU 最大长度值还大,将关闭该 STA 发送数据或管
            理类型的帧时的 RTS/CTS 握手。此属性值设为 0,将打开 STA 发送所有数据帧或管理
            帧时的 RTS/CTS 握手。该属性的默认值为 2347。"
    ::={  dot11OperationEntry 2 }

dot11ShortRetryLimit OBJECT-TYPE
        SYNTAX INTEGER (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "该属性应指示尝试发送每一帧的最大次数,帧的长度小于或等于 dot11RTSThreshold。
            该尝试次数应在指示失败状态之前完成。该属性的默认值为 7。"
    ::={  dot11OperationEntry 3 }

dot11LongRetryLimit OBJECT-TYPE
        SYNTAX INTEGER (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "该属性应指示尝试发送一帧的最大次数,帧的长度大于 dot11RTSThreshold。该尝试次
            数应在指示失败状态之前完成。该属性的默认值为 4。"
    ::={  dot11OperationEntry 4 }

dot11FragmentationThreshold OBJECT-TYPE
        SYNTAX INTEGER (256..2346)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "该属性应规定可以交付到 PHY 的 MPDU 的当前的最大长度值的八位位组数。当 MS-
```

DU 加上 MAC 头和尾之后的长度超过该属性值时，该 MPDU 应被分段。当结果帧在地址 1 字段是一个单个地址，且该帧的长度大于此阈值时，MSDU 或 MMPDU 应被分段。该属性默认值应取 2346 或附带 PHY 的 aMPDUMaxLength 两者中的较小值，并且不应超过附带 PHY 的 aMPDUMaxLength 或 2346 两者中的较小值。该属性的值应不小于 256。"

::={ dot11OperationEntry 5 }

dot11MaxTransmitMSDULifetime OBJECT-TYPE
SYNTAX INTEGER (1..4294967295)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"MaxTransmitMSDULifetime 是在初始的 MSDU 发送之后流逝的时间(单位为 TU)。在此时间内，进一步地尝试发送该 MSDU 应被终止。该属性默认值为 512。"
::={ dot11OperationEntry 6 }

dot11MaxReceiveLifetime OBJECT-TYPE
SYNTAX INTEGER (1..4294967295)
MAX-ACCESS read-write
STATUS current
DESCRITION
"MaxReceiveLifetime 是在最初接收到分段的 MMPDU 或 MSDU 之后流逝的时间，单位为 TU。在此时间内，进一步地尝试重组 MMPDU 或 MSDU 应被终止。该属性的默认值将是 512。"
::={ dot11OperationEntry 7 }

dot11ManufacturerID OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..128))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"ManufacturerID 至少应包括生产商的名称。还可能包括识别生产商的附加信息。该属性默认值为空。"
::={ dot11OperationEntry 8 }

dot11ProductID OBJECT-TYPE
SYNTAX DisplayString (SIZE(0..128))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"ProductID 至少应包括一个对生产商来说是惟一的标识符。还可能包括识别生产商的附加信息。该属性默认值为空。"
::={ dot11OperationEntry 9 }

```
-- ****************************************************************************
-- *        dot11OperationEntry 表结束
-- ****************************************************************************

-- ****************************************************************************
-- *        dot11Counters 表开始
-- ****************************************************************************
dot11CountersTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11CounterEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "包括 MAC 计数器的属性的组。以表格形式实现以允许在一个代理上有多个实例。"
    ::={   dot11mac 2 }

dot11CountersEntry OBJECT-TYPE
        SYNTAX Dot11counterEntry
        MAX-ACCESS not-accessible
        DESCRIPTION
            "dot11CountersEntry 表的一个项目。
            ifIndex—每个本规范接口由一个 ifEntry 代表。MIB 模块中的接口表由 ifIndex 索引。"
        INDEX{ifIndex}
    ::={   dot11CountersTable 1 }

 Dot11CountersEntry ::= SEQUENCE {
            dot11TransmittedFragmentCount                Counter32,
            dot11MulticastTransmittedFrameCount          Counter32,
            dot11FailedCount                             Counter32,
            dot11RetryCount                              Counter32,
            dot11MultipleRetryCount                      Counter32,
            dot11FrameDuplicateCount                     Counter32,
            dot11RTSSuccessCount                         Counter32,
            dot11RTSFailureCount                         Counter32,
            dot11ACKFailureCount                         Counter32,
            dot11ReceivedFramentCount                    Counter32,
            dot11MulticastReceivedFrameCount             Counter32,
            dot11FCSErrorCount                           Counter32,
            dot11TransmittedFrameCount                   Counter32,
            dot11WEPUndecryptableCount                   Counter32 }

dot11TransmittedFragmentCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
```

```
        STATUS current
        DESCRIPTION
            "对一个已得到确认的 MPDU,该 MPDU 在地址 1 字段是一个单个地址,或在数据或管
            理类型的地址 1 字段是一个组播地址,本计数器应递增。"
    ::={  dot11CountersEntry 1 }

dot11MulticastTransmittedFrameCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "只有当成功发送的 MSDU 的目的 MAC 地址中的组播比特被置位时,该计数器应递增。
            当在 ESS 中作为 STA 运行,其中这些帧都定向到 AP,这意味着已接收到一个对所有关
            联的 MPDU 的确认。"
    ::={  dot11CountersEntry 2 }

dot11FailedCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当一个 MSDU 由于尝试发送的次数超过 dot11ShortRetryLimit 或
            dot11LongRetryLimit 而未被成功发送时,该计数器应递增。"
    ::={  dot11CountersEntry 3 }

dot11RetryCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当一个 MSDU 在一次或多次重传之后被成功发送时,该计数器应递增。"
    ::={  dot11CountersEntry 4 }

dot11MultipleRetryCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当一个 MSDU 超过一次重传之后被成功发送时,该计数器应递增。"
    ::={  dot11CountersEntry 5 }

dot11FrameDuplicateCount OBJECT-TYPE
        SYNTAX Counter32
```

```
        MAX-ACCESS read-only
        STATUS current
        DESCRIPYION
            "当接收到的帧的序列控制字段指示该帧是复制的帧时,该计数器应递增。"
    ::={  dot11CounterEntry 6 }

dot11RTSSuccessCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当接收到一个响应 RTS 的 CTS 时,该计数器应递增。"
    ::={  dot11CountersEntry 7 }

dot11RTSFailureCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当未收到一个响应 RTS 的 CTS 时,该计数器应递增。"
    ::={  dot11CountersEntry 8 }

dot11ACKFailureCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当预期的 ACK 未接收到时,该计数器应递增。"
    ::={  dot11CountersEntry 9 }

dot11ReceivedFragmentCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "对于每个成功接收的数据或管理类型的 MPDU,该计数器应递增。"
    ::={  dot11CountersEntry 10 }
dot11MulticastReceivedFrameCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当接收到的 MSDU 的目的 MAC 地址的组播比特位置位时,该计数器应递增。"
```

```
    ::={   dot11CountersEntry 11 }

dot11FCSErrorCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当接收到的 MPDU 中检测到一个 FCS 差错时,该计数器应递增。"
    ::={   dot11CountersEntry 12 }

dot11TransmittedFrameCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "对于每个成功发送的 MSDU,该计数器应递增。"
    ::={   dot11CountersEntry 13 }

dot11WEPUndecryptableCount OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS deprecated
        DESCRIPTION
            "当收到的帧的帧控制字段的 WEP 子字段为 1 而映射到发送方 MAC 地址的密钥 WE-
            POn 值却指示该帧不应被加密,或者收到的帧由于接受方 STA 未实现保密选项而丢弃
            时,该计数器递增。"
    ::= { dot11CountersEntry 14 }
-- ***********************************************************************
-- *        dot11CountersEntry 表结束
-- ***********************************************************************

-- ***********************************************************************
-- *        GroupAddresses 表开始
-- ***********************************************************************
dot11GroupAddressesTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11GroupAddressesEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "这是一个概念性列表,它包含一组 MAC 地址。这些 MAC 地址标识了 STA 将要接收
            帧的组播地址。该属性默认值为空。"
    ::={   dot11mac 3 }
```

```
dot11GroupAddressesEntry OBJECT-TYPE
        SYNTAX Dot11GroupAddressesEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "GroupAddresses 中的一个项目(概念行)。
            ifIndex—每个本规范接口用一个 ifEntry 代表。MIB 模块中的接口表由 ifIndex 索引。"
        INDEX{ifIndex,dot11GroupAddressesIndex}
    ::={  dot11GroupAddressesTable 1 }

 Dot11GroupAddressesEntry ::= SEQUENCE{
            dot11GroupAddressesIndex                Integer32,
            dot11Address                            MacAddress,
            dot11GroupAddressesStatus               RowStatus}

dot11GroupAddressesIndex OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "辅助变量,用于标识组地址列表中的纵列对象实例。"
    ::={  dot11GroupAddressesEntry 1 }

dot11Address OBJECT-TYPE
        SYNTAX MacAddress
        MAX-ACCESS read-create
        STATUS current
        DESCRIPTION
            "用于标识组播地址的 MAC 地址,STA 将从该地址接收帧。"
    ::={ dot11GroupAddressEntry 2 }

dot11GroupAddressesStatus OBJECT-TYPE
        SYNTAX RowStatus
        MAX-ACCESS read-create
        STATUS current
        DESCRIPTION
            "用于在组地址列表中创建、修改和删除纵列对象实例的状态栏。"
        DEFVAL{active}
    ::={ dot11GroupAddressEntry 3 }
-- ***************************************************************************
-- *      GroupAddresses 表结束
--  ****************************************************************************
```

```
-- ****************************************************************************
-- *      资源类型属性模板
-- ****************************************************************************
dot11ResourceTypeIDName OBJECT-TYPE
        SYNTAX DisplayString (SIZE(4))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "包含被管理对象的 Rescource Type ID 的名称。该属性只读并且总包含 RTID 值。该属
            性值不应用作为任一个其他被管理对象类别的命名属性。"
        REFERENCE "IEEE Std 802.1F-1993,A.7"
        DEFVAL {"RTID"}
    ::={  dot11restAttribute 1 }

-- ****************************************************************************
-- *      dot11ResourceInfo 表开始
-- ****************************************************************************

dot11ResorceInfoTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11ResourceInfoEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "以来自被管理对象的可读数据表来提供指示标识信息源的信息的方法。"
        REFERENCE "IEEE Std 802.1F-1993,A.7"
    ::={  dot11restAttribute 2 }

dot11ResourceInfoEntry OBJECT-TYPE
        SYNTAX Dot11RescourceInfoEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "dot11ResourceInfo 表的项目。
            ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
    INDEX {ifIndex}
    ::={  dot11ResourceInfoTable 1 }

 Dot11ResourceInfoEntry ::= SEQUENCE {
            dot11manufacturerOUI                    OCTET STRING,
            dot11manufacturerName                   DisplayString,
            dot11manufacturerProductName            DisplayString,
            dot11manufacturerProductVersion         DisplayString}
```

```
dot11manufacturerOUI OBJECT-TYPE
        SYNTAX OCTET STRING(SIZE(3))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "提取组织惟一标识符的值。"
    ::={  dot11ResourceInfoEntry 1 }

dot11manufacturerName OBJECT-TYPE
        SYNTAX DisplayString (SIZE(0..128))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "用于标识资源生产商的可打印的串。最大串长度为 128 八位位组。"
    ::={  dot11ResourceInfoEntry 2 }

dot11manufacturerProductName OBJECT-TYPE
        SYNTAX DisplayString (SIZE(0..128))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "用于标识资源生产商产品名称的可打印的串。最大的串长度为 128 八位位组。"
    ::={  dot11ResourceInfoEntry 3 }

dot11manufacturerProductVersion OBJECT-TYPE
        SYNTAX DisplayingString(SIZE(0...128))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "用于标识资源生产商产品版本的可打印的串,最大串长度为 128 八位位组。"
    ::={ dot11ResourceInfoEntry 4  }
-- ***************************************************************************
-- *       dot11ResourceInfo 表结束
-- ***************************************************************************

-- ***************************************************************************
-- *       PHY 属性模板
-- ***************************************************************************
-- ***************************************************************************
-- *       dot11PhyOperation 表开始
-- ***************************************************************************
dot11PhyOperationTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyOperationEntry
```

```
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "与操作有关的 PHY 级属性。由 ifIndex 索引的表来实现以允许在一个代理上有多个
            实例。"
    ::={  dot11phy 1 }

dot11PhyOperationEntry OBJECT-TYPE
        SYNTAX Dot11OperationEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "dot11PhyOperation 表的项目。
            ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
        INDEX {ifIndex}
    ::={  dot11PhyOperationTable 1 }

Dot11PhyOperationEntry ::= SEQUENCE {
            dot11PHYType                    INTEGER,
            dot11CurrentRegDomain           Integer32,
            dot11TempType                   INTEGER}

dot11PHYType OBJECT-TYPE
        SYNTAX INTEGER{fhss(1), dsss(2), irbaseband(3)}
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "该值是 8 比特整数值,用于标识连接的 PLCP 和 PMD 支持的 PHY 类型。
            目前定义的值和它们对应的 PHY 类型是:
            FHSS   2.4GHz=01,DSSS   2.4GHz=02,IR   Baseband=03"
        ::={  dot11PhyOperationEntry 1 }

dot11CurrentRegDomain OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "这个 PMD 实例支持的当前管理域。
            该对象与 dot11RegDomainsSupported 列表中的 RegDomains 之一相对应。"
    ::={  dot11PhyOperationEntry 2 }

dot11TempType OBJECT-TYPE
        SYNTAX INTEGER {tempType1(1), tempType2(2) }
```

```
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "根据预期的环境条件有不同的操作温度要求。此属性描述了当前 PHY 的操作温度范
            围。当前定义的值和它们相应的温度范围是：
            Type 1 = X'01' —— 0° C 到 40°C 的商用范围；
            Type 2 = X'02' ——－30° C 到 70° C 的工业用范围"。
    ::={  dot11PhyOperationEntry 3 }
-- ****************************************************************************
-- *        dot11PhyOperation 表结束
-- ****************************************************************************

-- ****************************************************************************
-- *         dot11PhyAntenna 表开始
-- ****************************************************************************
dot11PhyAntennaTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyAntennaEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "PhyAntenna 的属性组。由 ifIndex 索引的表来实现，以允许在一个代理上有多个
            实例。"
    ::={  dot11phy 2 }

dot11PhyAntennaEntry OBJECT-TYPE
        SYNTAX Dot11PhyAntennaEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "dot11PhyAntenna 表的项目。
            ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
            INDEX {ifIndex}
    ::={  dot11PhyAntennaTable 1 }

 Dot11PhyAntennaEntry ::= SEQUENCE {
            dot11CurrentTxAntenna                    Integer32,
            dot11DiversitySupport                    INTEGER,
            dot11CurrentRxAntenna                    Integer32 }

dot11CurrentTxAntenna OBJECT-TYPE
        SYNTAX Integer32 (1..255)
        MAX-ACCESS read-write
        STATUS current
```

```
        DESCRIPTION
            "当前被用于发射的天线。该值是出现在 dot11SupportedTxAntenna 值中的一个。该值
            可以由管理代理使用,以控制用哪个天线进行发射。"
    ::={  dot11PhyAntennaEntry 1 }

dot11DiversitySupport OBJECT-TYPE
        SYNTAX INTEGER {fixedlist(1), notsupported(2),dynamic(3)}
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "该实现支持分集,编码为:
            X'01'——分集可用。并可在 dot11DiversitySelectionRx 中定义的固定的天线列表中执
                    行。
            X'02'——不支持分集。
            X'03'——支持分集,并可控制分集。在这种情况下
            dot11DiversitySelectionRx 属性可由 LME 动态修改。"
    ::={  dot11PhyAntennaEntry 2 }

dot11CurrentRxAntenna OBJECT-TYPE
        SYNTAX Integer32 (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "如果 dot11DiversitySupport 指示不支持分集,当前用于接收的天线。所选的天线应为
            在 dot11AntennasListTable 中标记为接收中的一个。"
    ::={  dot11PhyAntennaEntry 3 }
-- ***********************************************************************
-- *        dot11PhyAntenna 表结束
-- ***********************************************************************

-- ***********************************************************************
-- *        dot11PhyTxPower 表开始
-- ***********************************************************************

dot11PhyPowerTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyPowerEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "dot11PhyTxPowerTable 属性组。由 STA ID 索引的表来实现以允许在一个代理上有
            多个实例。"
    ::={  dot11phy 3 }
```

```
dot11PhyTxPowerEntry OBJECT-TYPE
        SYNTAX Dot11PhyTxPowerEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "dot11PhyTxPower 表的项目。
            ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
        INDEX {ifIndex}
    ::={  dot11PhyPowerTable 1 }

Dot11PhyPowerEntry ::= SEQUENCE {
            dot11NumberSupportedPowerLevels      INTEGER,
            dot11TxPowerLevel1                   INTEGER,
            dot11TxPowerLevel2                   INTEGER,
            dot11TxPowerLevel3                   INTEGER,
            dot11TxPowerLevel4                   INTEGER,
            dot11TxPowerLevel5                   INTEGER,
            dot11TxPowerLevel6                   INTEGER,
            dot11TxPowerLevel7                   INTEGER,
            dot11TxPowerLevel8                   INTEGER,
            dot11CurrentTxPowerLevel             INTEGER}

dot11NumberSupportedPowerLevels OBJECT-TYPE
        SYNTAX INTEGER (1..8)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "PMD 支持的功率电平数。该属性能取 1 到 8 中的一个值。"
    ::={  dot11PhyTxPowerEntry 1 }

dot11TxPowerLevel1 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "LEVEL1 的发射输出功率,单位为 mW,这也是默认功率电平。"
    ::={  dot11PhyTxPowerEntry 2 }

dot11TxPowerLevel2 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
```

```
        DESCRIPTION
            "LEVEL2 的发射输出功率,单位为 mW。"
    ::={  dot11PhyTxPowerEntry 3 }

dot11TxPowerLevel3 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "LEVEL3 的发射输出功率,单位为 mW。"
    ::={  dot11PhyTxPowerEntry 4 }

dot11TxPowerLevel4 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "LEVEL4 的发射输出功率,单位为 mW。"
    ::={  dot11PhyTxPowerEntry 5 }


dot11TxPowerLevel5 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "LEVEL5 的发射输出功率,单位为 mW。"
    ::={  dot11PhyTxPowerEntry 6 }

dot11TxPowerLevel6 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "LEVEL6 的发射输出功率,单位为 mW。"
    ::={  dot11PhyTxPowerEntry 7 }

dot11TxPowerLevel7 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "LEVEL7 的发射输出功率,单位为 mW。"
```

```
    ::={  dot11PhyTxPowerEntry 8 }

dot11TxPowerLevel8 OBJECT-TYPE
        SYNTAX INTEGER (0..10000)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "LEVEL8 的发射输出功率,单位为 mW。"
    ::={  dot11PhyTxPowerEntry 9 }

dot11CurrentTxPowerLevel OBJECT-TYPE
        SYNTAX INTEGER (1..8)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当前用于发送数据的 TxPowerLevel N。一些 PHY 也用该值来确定 CCA 所需要的接
            收机灵敏度。"
    ::={  dot11PhyTxPowerEntry 10 }
-- ****************************************************************************
-- *      dot11PhyTxPower 表结束
-- ****************************************************************************

-- ****************************************************************************
-- *      dot11PhyFHSS 表开始
-- ****************************************************************************

dot11PhyFHSSTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyFHSSEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "dot11PhyFHSSTable 的属性组。由 STA ID 索引的表来实现,并允许在一个
            代理上有多个实例。"
    ::={  dot11phy 4 }

dot11PhyFHSSEntry OBJECT-TYPE
        SYNTAX Dot11PhyFHSSEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "dot11PhyFHSS 表的项目。
            ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
        INDEX {ifIndex}
```

```
    ::={   dot11PhyFHSSTable 1 }

 Dot11PhyFHSSEntry ::= SEQUENCE {
            dot11HopTime                        INTEGER,
            dot11CurrentChannelNumber           INTEGER,
            dot11MaxDwellTime                   INTEGER,
            dot11CurrentDwellTime               INTEGER,
            dot11CurrentSet                     INTEGER,
            dot11CurrentPattern                 INTEGER,
            dot11CurrentIndex                   INTEGER }

dot11HopTime OBJECET-TYPE
        SYNTAX INTEGER (224)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "PMD 由信道 2 改变到信道 80 的时间,单位为 μs。"
     ::={   dot11PhyFHSSEntry 1 }

dot11CurrentChannelNumber OBJECT-TYPE
        SYNTAX INTEGER (0..99)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "由射频合成器的频率输出的当前信道号。"
    ::={   dot11PhyFHSSEntry 2 }

dot11MaxDwellTime OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "允许发射机在单个信道上工作的最大时间,单位为 TU。"
    ::={   dot11PhyFHSSEntry 3 }

dot11CurrentDwellTime OBJECT-TYPE
        SYNTAX INTEGER (1..65535)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "当前由 MAC 设置的发射机在单个信道上操作的时间,单位为 TU。默认值是 19TU。"
    ::={   dot11PhyFHSSEntry 4 }
```

```
dot11CurrentSet OBJECT-TYPE
        SYNTAX INTEGER (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "当前的 PHY  LME 模式组,用于确定跳频序列。"
    ::={  dot11PhyFHSSEntry 5 }

dot11CurrentPattern OBJECT-TYPE
        SYNTAX INTEGER (0..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "当前 PHY LME 用来确定跳频序列的模式。"
    ::={  dot11PhyFHSSEntry 6 }

dot11CurrentIndex OBJECT-TYPE
        SYNTAX INTEGER (1..255)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "PHY LME 用来确定 CurrentChannelNumber 的当前索引值。"
    ::={  dot11PhyFHSSEntry 7 }
-- ****************************************************************************
-- *        dot11PhyFHSS 表结束
-- ****************************************************************************

-- ****************************************************************************
-- *        dot11PhyDSSSEntry 表开始
-- ****************************************************************************
dot11PhyDSSSEntry OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11PhyDSSSEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "dot11PhyDSSSEntry 的属性的项目。由 ifIndex 索引的表来实现,并允许一个代理有多
            个实例。"
    ::={  dot11phy 5 }

dot11PhyDSSSEntry OBJECT-TYPE
        SYNTAX Dot11PhyDSSSEntry
        MAX-ACCESS not-accessible
        STATUS current
```

```
        DESCRIPTION
            "dot11PhyDSSSEntry 表的项目。
            ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
        INDEX {ifIndex}
    ::={   dot11PhyDSSSTable 1 }

 Dot11PhyDSSSEntry ::= SEQUENCE {
            dot11CurrentChannel                INTEGER,
            dot11CCAModeSupported              INTEGER,
            dot11CurrentCCAMode                INTEGER,
            dot11EDThreshold                   Integer32 }

dot11CurrentChannel OBJECT-TYPE
        SYNTAX INTEGER (1..14)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "DSSS PHY 的当前操作的频率信道。有效的信道号在 15.4.6.2 中定义。"
    ::={   dot11PhyDSSSEntry 1 }

dot11CCAModeSupported OBJECT-TYPE
        SYNTAX INTEGER (1..7)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "dot11CCAModeSupported 是一个位有效的值,代表了 PHYCCA 支持的所有模式。有
            效值为:
                仅能量检测 (ED_ONLY)=01,
                仅载波侦听 (CS_ONLY)=02,
                载波侦听和能量检测(ED_and_CS)=04,
                或这些值的任意的逻辑和。"
     ::={   dot11PhyDSSSEntry 2 }

dot11CurrentCCAMode OBJECT-TYPE
        SYNTAX INTEGER{edonly(1), csonly(2), edonly(4)}
        MAX-ACCESS read-write
        STATUS current
       DESCRIPTION
            "在当前操作中的 CCA 方法。有效值为:
                    仅能量检测 (edonly)=01,
                    仅载波侦听 (csonly)=02,
                    载波侦听和能量检测 (edandcs)=04"
    ::={ dot11PhyDSSSEbtry 3 }
```

```
dot11EDThreshold OBJECT-TYPE
        SYNTAX integer32
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
           "用于 DSSS PHY 的当前能量检测阈值。"
    ::={ dot11phyDSSSEntry 4 }
-- ***********************************************************************
-- *      dot11phyDSSSEntry 表结束
-- ***********************************************************************

-- ***********************************************************************
-- *      dot11phyIR 表开始
-- ***********************************************************************
dot11PhyIRTable OBJECT-TYPE
      SYNTAX SEQUENCE OF Dot11PhyIREntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
            "dot11PhyIRTable 的属性组,由 ifIndex 索引的表来实现,并允许一个代理上
            有多个实例。"
 ::={ dot11phy 6 }

dot11PhyIREntry OBJECT-TYPE
      SYNTAX Dot11PhyIREntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
    "dot11PhyIR 表中的项目。
     ifIndex—每个本规范接口由一个 ifEntry 来表示。MIB 模块中的接口表由 ifIndex 索引。"
      INDEX {ifIndex}
   ::={ dot11PhyIRTable 1 }

dot11PhyIREntry    ::=SEQUENCE {
           dot11CCAWatchdogTimeMax       Integer32,
           dot11CCAWatchdogCountMax      Integer32,
           dot11CCAWatchdogTimeMin       Integer32,
           dot11CCAWatchdogCountMin      Integer32}

dot11CCAWatchdogTimeMax OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-write
```

```
        STATUS current
        DESCRIPTION
            "该参数同 CCAWatchdogCountMax 一起确定何时在信道中检测到的能量可被忽略。"
    ::={ dot11PhyIREntry 1 }

dot11CCAWatchdogCountMax OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "该参数同 CCAWatchDogTimeMax 一起决定何时在信道中检测到的能量可被忽略。"
   ::={  dot11PhyEntry 2 }

dot11CCAWatchdogTimeMin OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "CCAWatchdogTimeMax 可设置的最小值。"
 ::={  dot11PhyIREntry 3 }

dot11CCAWatchdogCountMin OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "CCAWatchdogCount 可设置的最小值。"
   ::={  dot11PhyIREntry 4 }

-- ******************************************************************************
-- *     dot11PhyIR 表结束
-- ******************************************************************************

-- ******************************************************************************
-- *     dot11RegDomainsSupported 表开始
-- ******************************************************************************
dot11RegDomainsSupportedTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11RegDomainsSupportEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
          "根据管理域不同有不同的操作要求。该属性列表描述了在实现中，
          PLCP 和 PMD 支持的管理域。当前定义的值及它们相应的管理域是：
```

```
        FCC (USA) = X'10', DOC (Canada) = X'20', ETSI (most of Europe)
        = X'30', Spain = X'31', France = X'32', MKK (Japan) = X'40' "
    ::={  dot11phy 7 }

dot11RegDomainsSupportEntry OBJECT-TYPE
      SYNTAX Dot11RegDomainsSupportEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
          "dot11RegDomainsSupport 表中的项目。
          ifIndex—每个本规范接口由一个 ifEntry 来表示。MIB 模块中的接口表由 ifIndex 索引。"
      INDEX {ifIndex,dot11RegDomainsSupportIndex}
    ::={ dot11RegDomainsSupportedTable 1 }

 Dot11RegDomainsSupportEntry ::= SEQUENCE {
          dot11RegDomainsSupportIndex        Integer32,
          dot11RegDomainsSupportValue        INTEGER }

dot11RegDomainsSupportIndex OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
          "辅助变量,用于标识 RegDomainsSupport 表中的纵列对象实例。"
    ::={  dot11RegDomainsSupportEntry 1 }

dot11RegDomainsSupportValue OBJECT-TYPE

      SYNTAX INTEGER {fcc(16),doc(32),etsi(48),spain(49),france(50),mkk(64) }
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
          "根据管理区域不同有不同的操作要求。此属性列表描述了在实现中,PLCP 和 PMD 支
          持的管理域。当前定义的值及它们相应的管理域是:
          FCC(USA)=X'10', DOC(Canada)=X'20', ETSI(most of Europe)=X'30',
          Spain=X'31', Francd=X'32', MKK(Japan)=X'40'
    ::={ dot11RegDomainsSupportEntry 2 }

-- ****************************************************************************
-- * dot11RegDomainsSupported 表结束
-- ****************************************************************************

-- ****************************************************************************
```

```
-- * dot11AntennasListTable 表开始
-- ******************************************************************************
dot11AntennasListTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11AntennasListEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
                "这个表指出天线列表。一个天线可被标记为能够发射、接收并且/或者能参与接收
                分集。该表中的每一个项目代表一个带有自身属性的单个天线。
                该表中可包含天线的最大数是 255。"
    ::={ dot11phy 8 }

dot11AntennasListEntry OBJECT-TYPE
        SYNTAX Dot11AntennasListEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
                "在 dot11AntennasListTable 中的一个项目,代表了单个天线的属性。
                ifIndex—每个本规范接口由一个 ifEntry 代表。MIB 模块中的接口表由 ifIndex
                索引。"
        INDEX { ifIndex,dot11AntennaListIndex}
   ::={dot11AntennasListTable 1 }

Dot11AntennasListEntry ::=SEQUENCE{
                dot11AntennaListIndex           Integer32,
                dot11SupportedTxAntenna         TruthValue,
                dot11SupportedRxAntenna         TruthValue,
                dot11DiversitySelectionRx       TruthValue}

dot11AntennaListListIndex OBJECT-TYPE
      SYNTAX Integer32 (1..225)
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
          "用于标识 dot11AntennasList 表中纵列对象实例的天线的惟一索引。"
  ::={  dot11AntennasListEntry 1 }

dot11SupportedTxAntenna OBJECT-TYPE
      SYNTAX TruthValue
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
          "当为真时,该对象指示由 dot11AntennaIndex 表示的天线可用作发射天线。"
```

```
    ::={   dot11AntennasListEntry 2 }

dot11SupportedRxAntenna OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "当为真时,该对象指示由 dot11AntennaIndex 表示的天线可用作接收天线。"
    ::={   dot11AntennasListEntry 3 }

dot11DiversitySelectionRx OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "当为真时,该对象指示由 dot11AntennaIndex 表示的天线可用作接收分集。
            如果天线如 dot11SupportedRxAntenna 指示可用作接收天线,该对象只可取真。"
    ::={   dot11AntennasListEntry 4 }
-- ******************************************************************************
-- *      dot11AntennasList 表结束
-- ******************************************************************************

-- ******************************************************************************
-- *      SupportedDataRatesTx 表开始
-- ******************************************************************************
dot11SupportedDataRatesTxTable OBJECT-TYPE
        SYNTAX SEQUENCE OF Dot11SupportedDataRatesTxEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "PLCP 和 PMD 支持的发送比特速率,由 X'02-X'7f 的计数表示。
              相对应的数据速率从 1Mb/s 到 63.5Mb/s 以 500kb/s 递增。这些数据速率受每个单个
              PHY 的限制。"
    ::={   dot11phy 9 }

dot11SupportedDataRatesTxEntry OBJECT-TYPE
          SYNTAX Dot11SupportedDataRatesTxEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
              "在 dot11SupportedDataRatesTxTable 中的项目(概念行)。
              ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
          INDEX { ifIndex,dot11SupportedDataRatesTxIndex}
```

```
    ::={dot11SupportedDataRatesTxTable 1 }

Dot11SupportedDataRatesTxEntry ::= SEQUENCE {
                dot11SupportedDataRatesTxIndex          Integer32,
                dot11SupportedDataRatesTxValue          Integer32 }

dot11SupportedDataRatesTxIndex OBJECT-TYPE
      SYNTAX Integer32 (1..8)
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
          "标识访问数据速率的索引对象。范围是从 1 到 8。"
   ::={   dot11SupportedDataRatesTxEntry 1 }

dot11SupportedDataRatesTxValue OBJECT-TYPE
        SYNTAX Integer32 (2..127)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "PLCP 和 PMD 支持的发送比特速率,由 X'02—X'7f 的一个计数表示,相对应的数据速
            率从 1Mb/s 到 63.5Mb/s 以 500kb/s 递增。这些数据速率受每个单个的 PHY 限制。"
      ::={   dot11SupportedDataRatesTxEntry 2 }
-- ******************************************************************************
-- *      dot11SupportedDataRatesTx 表结束
-- ******************************************************************************

-- ******************************************************************************
-- *      SupportedDataRatesRx 表开始
-- ******************************************************************************
dot11SupportedDataRatesRxTable OBJECT-TYPE
      SYNTAX SEQUENCE OF Dot11SupportedDataRatesRxEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
            "PLCP 和 PMD 支持的接收比特速率。由 X'002—X'7f 的一个计数表示。相对应的数
            据速率从 1Mb/s 到 63.5Mb/s 以 500kb/s 递增"
   ::={   dot11phy 10 }

dot11SupportedDataRatesRxEntry OBJECT-TYPE
      SYNTAX Dot11SupportedDataRatesRxEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
```

```
            "dot11SupportedDataRatesRx 表中的项目(概念行)。
             ifIndex—每个本规范接口由一个 ifEntry 表示。MIB 模块中的接口表由 ifIndex 索引。"
      INDEX{ifIndex, dot11SupportedDataRatesRxIndex}
  ::={   dot11SupportedDataRatesRxTable 1 }

 Dot11SupportedDataRatesRxEntry ::= SEQUENCE {
            dot11SupportedDataRatesRxIndex          Integer32,
            dot11SupportedDataRatesRxValue          Integer32 }

dot11SupportedDataRatesRxIndex OBJECT-TYPE
      SYNTAX Integer32 (1..8)
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
            "标识访问数据速率的索引对象。范围为 1 到 8 。"
  ::={   dot11SupportedDataRatesRxEntry 1 }

dot11SupportedDataRatesRxValue OBJECT-TYPE
      SYNTAX Integer32 (2..127)
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "PLCP 和 PMD 支持的接收比特速率。由 X'02—X'7f 的一个计数表示,相对应的数据
            速率从 1Mb/s 到 63.5Mb/s 以 500kb/s 递增。"
  ::={   dot11SupportedDataRatesRxEntry 2 }

-- ******************************************************************************
-- *     dot11SupportedDataRatesRx 表结束
-- ******************************************************************************

-- ******************************************************************************
-- *     一致性信息
-- ******************************************************************************
dot11Conformance OBJECT IDENTIFIER ::= { ieee802dot11 5 }
dot11Groups OBJECT IDENTIFIER ::= {dot11Conformance 1 }
dot11Compliances OBJECT IDENTIFIER ::= { dot11Conformance 2 }

-- ******************************************************************************
-- *     一致性声明
-- ******************************************************************************
dot11Compliance MODULE-COMPLIANCE
      STATUS current
      DESCRIPTION
```

```
            "实现本规范 MIB 的 SNMPv2 实体的一致性声明。"
 MODULE
  MANDATORY-GROUPS {
      dot11SMTbase2,
      dot11MACbase,
      dot11CountersGroup,
      dot11SmtAuthenticationAlgorithms,
      dot11ResourceTypeID,
      dot11PhyOperationComplianceGroup }

 GROUP dot11PhyDSSSComplianceGroup
        DESCRIPTION
              "当对象 dot11PHYType 有值 dsss 时,要求实现该组。该组与
              dot11PhyIRComplianceGroup 和 dot11PhyFHSSComplianceGroup 组互不相容。"

 GROUP dot11PhyIRComplianceGroup
        DESCRIPTION
              "当对象 dot11PHYType 有值 irbaseband 时,要求实现该组。该组与
              dot11PhyDSSSComplianceGroup 和 dot11PhyFHSSComplianceGroup 组互不相容。"

 GROUP dot11PhyFHSSComplianceGroup
        DESCRIPTION
              "当对象 dot11PHYType 有值 fhss 时,要求实现该组。该组与
              dot11PhyDSSSComplianceGroup 和 dot11PhyIRComplianceGroup 组互不相容。"

-- OPTIONAL-GROUPS {
--    dot11SMTprivacy,
--    dot11MACStatistics,
--    dot11PhyAntannaComplianceGroup,
--    dot11PhyTxPowerComplianceGroup,
--    dot11PhyRegDomainsSupportGroup,
--    dot11PhyAntennasListGroup,
--    dot11PhyRatesGroup }
::= { dot11Compliances 1 }

-- ******************************************************************************
-- *  组 - 一致性单元
-- ******************************************************************************
dot11SMTbase OBJECT-GROUP
       OBJECTS {
              dot11StationID,
              dot11MediumOccupacyLimit,
              dot11CFPollable,
```

```
            dot11CFPPeriod,
            dot11CFPMaxDuration,
            dot11AuthenticationResponseTimeOut,
            dot11PowerManagementMode,
            dot11DesiredSSID,
            dot11DesiredBSSType,
            dot11OperationalRatesSet,
            dot11BeaconPeriod,
            dot11DTIMPeriod,
            dot11AssociationResponseTimeOut}
        STATUS deprecated
        DESCRIPTION
            "SMT 对象类在 STA 上提供必要的支持以管理 STA 中的进程,使得 STA 可作为
            本规范网络的一部分协调工作。"
    ::={ dot11Group 1 }

dot11MACbase OBJECT-GROUP
        OBJECTS { dot11MACAddress,
                  dot11Address,
                  dot11GroupAddressesStatus,
                  dot11RTSThreshold,
                  dot11ShortRetryLimit,
                  dot11LongRetryLimit,
                  dot11FreagmentationThreshold,
                  dot11MaxTransmitMSDULifetime,
                  dot11MaxReceiveLifetime,
                  dot11ManufacturerID,
                  dot11ProductID }
        STATUS current
        DESCRIPTION
            "MAC 对象类为访问控制、帧检验序列的生成和验证、以及向高层正确交付有效数据提
            供必要的支持。"
    ::={ dot11Group 3 }

dot11MACStatistics OBJECT-GROUP
        OBJECTS { dot11RetryCount,
                  dot11MutipleRetryCount,
                  dot11RTSSuccessCount,
                  dot11RTSFailCount,
                  dot11ACKFailureCount,
                  dot11FrameDuplicateCount }
        STATUS current
        DESCRIPTION
```

```
            "MACStatistics 包提供 MAC 操作上的扩展统计信息,这个包是完全可选的。"
    ::={ dot11Group 4 }

dot11ResourceTypeID OBJECT-GROUP
        OBJECTS { dot11resourceTypeIDName,
                  dot11manufacturerOUI,
                  dot11manufacturerName,
                  dot11manufacturerProductName,
                  dot11manufacturerProductVersion }
        STATUS current
        DESCRIPTION
            "该属性用于标识 STA,STA 的生产商以及不同的产品名称和版本。"
    ::={ dot11Group 5 }

dot11SmtAuthenticationAlgorithms OBJECT-GROUP
        OBJECTS { dot11AuthenticationAlgorithm, dot11AuthenticationAlgorithmsEnable }
        STATUS current
        DESCRIPTION
            "链路验证算法表。"
        ::= {dot11Groups 6 }

dot11PhyOperationComplianceGroup OBJECT-GROUP
        OBJECTS { dot11PHYType, dot11CurrentRegDomain, dot11TempType }
        STATUS current
        DESCRIPTION
            "PHY 层操作属性。"
          ::={ dot11Groups 7 }

dot11PhyAntennaComplianceGroup OBJECT-GROUP
        OBJECTS { dot11CurrentTxAntenna,dot11DiversitySupport,dot11CurrentRxAntenna }
        STATUS current
        DESCRIPTION
            "本规范的数据速率属性。"
    ::={ dot11Groups 8 }

dot11PhyTxPowerComplianceGroup OBJECT-GROUP
        OBJECTS { dot11NumberSupportedPowerLevels,dot11TxPowerLevel1,
                  dot11TxPowerLevel2, dot11TxPowerLevel3, dot11TxPowerLevel4,
                  dot11TxPowerLevel5, dot11TxPowerLevel6, dot11TxPowerLevel7,
                  dot11TxPowerLevel8, dot11CurrentTxPowerLevel }
        STATUS current
        DESCRIPTION
            "发射功率的控制和管理属性。"
```

```
    ::={ dot11Groups 9 }

dot11PhyFHSSComplianceGroup OBJECT-GROUP
        OBJECTS { dot11HopTime, dot11CurrentChannelnumber, dot11MaxDwellTime,
                  dot11CurrentDwellTime, dot11CurrentSet, dot11CurrentPattern,
                  dot11CurrentIndex }
        STATUS current
        DESCRIPTION
            "本规范的频率跳变配置属性。"
    ::={ dot11Groups 10 }

dot11PhyDSSSComplianceGroup OBJECT-GROUP
        OBJECTS { dot11CurrentChannel, dot11CCAModeSupported,
                  dot11CurrentCCAMode, dot11Edthreshold }
        STATUS current
        DESCRIPTION
            "本规范的 DSSS 配置属性。"
    ::={ dot11Groups 11 }

dot11PhyIRComplianceGroup OBJECT-GROUP
        OBJECTS { dot11CCAWatchdogTimerMax, dot11CCAWatchdogCountMax,
                  dot11CCAWatchdogTimerMin, dot11CCAWatchdogCountMin }
        STATUS current
        DESCRIPTION
            "本规范的基带 IR 的配置属性。"
    ::={ dot11Groups 12 }

dot11PhyRegDomainsSupportGroup OBJECT-GROUP
        OBJECTS { dot11RegDomainsSupportValue }
        STATUS current
        DESCRIPTION
            "规定支持的管理域属性。"
    ::={ dot11Groups 13 }

dot11PhyAntennasListGroup OBJECT-GROUP
        OBJECTS { dot11SupportedTxAntenna,
                  dot11SupportedRxAntenna,
                  dot11DiversitySelectionRx }
        STATUS current
        DESCRIPTION
            "规定支持的管理域属性。"
    ::={ dot11Groups 14 }
dot11PhyRatesGroup OBJECT-GROUP
```

```
        OBJECTS { dot11SupportedDataRatesTxValue,
                  dot11SupportedDataRatesRxValue }
        STATUS current
        DESCRIPTION
            "本规范数据速率属性。"
    ::={ dot11Group 15 }

dot11CountersGroup OBJECT-GROUP
        OBJECTS { dot11TransmittedFragmentCount,
                  dot11MuticastTransmittedFrameCount,
                  dot11FailedCount,
                  dot11ReceivedFragmentCount,
                  dot11MulticastReceivedFrameCount,
                  dot11FCSErrorCount,
                  dot11TransmittedFrameCount }
        STATUS current
        DESCRIPTION
            "dot11MACStatistics 组中未描述在 dot11CountersGroup 中的属性。
            这些对象是必备的。"
    ::={ dot11Groups 16 }

dot11NotofocationGroup NOTIFICATION-GROUP
        NOTIFICATIONS { dot11Disassociate,
                        dot11Deauthentication,
                        dot11AuthenticationFail }
        STATUS current
        DESCRIPTION
            "本规范通告。"
    ::={ dot11Groups 17 }

dot11SMTbase2  OBJECT-GROUP
        OBJECTS { dot11MediumOccupancyLimit,
                  dot11CFPollable,
                  dot11CFPPeriod,
                  dot11CFPMaxDuration,
                  dot11AuthenticationResponseTimeOut,
                  dot11PowerManagementMode,
                  dot11DesiredSSID,
                  dot11DesiredBSSType,
                  dot11OperationalRatesSet,
                  dot11BeaconPeriod,
                  dot11DTIMPeriod,
                  dot11AssociationResponseTimeOut,
```

```
                dot11DisassociateReason,
                dot11DisassociationStation,
                dot11DeauthenticationReason,
                dot11DeauthenticationStation,
                dot11AuthenticationFailStatus,
                dot11AuthenticationFailStation }
        STATUS current
        DESCRIPTION
            "SMTbase2 对象类在 STA 上提供必要的支持以管理 STA 中的进程,使得 STA 可
            作为本规范网络的一部分协调工作。"
    ::= { dot11Groups 18 }

-- ******************************************************************************
-- *    IEEE802dot11-MIB 结束
-- ******************************************************************************
END

-- ******************************************
-- *   GB 15629dot11-WAPI-MIB  定义开始
-- ******************************************
GB15629dot11-WAPI-MIB DEFINITIONS ::= BEGIN
  IMPORTS
    MODULE-IDENTITY,OBJECT-TYPE,Counter32,
    Unsigned32                                  FROM SNMPv2-SMI
    MacAddress,TruthValue,DisplayString         FROM SNMPv2-TC
    MODULE-COMPLIANCE,OBJECT-GROUP              FROM SNMPv2-CONF
    ifIndex                                     FROM RFC1213-MIB;

-- ***********************************************
-- *   模块标识
-- ***********************************************
gb15629dot11wapiMIB MODULE-IDENTITY
  ORGANIZATION
    "中国宽带无线 IP 标准工作组(ChinaBWIPS)(China Broadband Wireless IP Standard Group )"
  CONTACT-INFO
    "工作组信息如下:
    通信地址:中国西安高新技术产业开发区西高新邮局 88#信箱
    邮政编码:710075
    电话:86-29-87607826
    传真:86-29-87607829
    E-mail: bwips@chinabwips.org
    Address:P.O.BOX 88,West High-tech Development Center, Xi'an, China.
    Postcode:710075
```

```
    Tel: +86 29 87607826
    Fax: +86 29 87607829"
  DESCRIPTION
        "管理本规范的 WAPI 模块。"
  ::= { iso(1)  member-body(2)  cn(156)  bwips(11235)  GB15629(15629) GB15629-11(11)
GB15629-11-mibs(1) 1 }

-- ***********************************
-- *  主体部分
-- ***********************************
wapiMIBObjects OBJECT IDENTIFIER              ::= { gb15629dot11wapiMIB  1 }
wapiMIBConformance OBJECT IDENTIFIER          ::= { gb15629dot11wapiMIB 2 }

-- ****************************************
-- *    wapiMIBObjects 中的表
-- ****************************************
-- gb15629dot11wapiConfig                          ::= { wapiMIBObjects 1  }
-- gb15629dot11wapiConfigUnicastCiphers            ::= { wapiMIBObjects 2  }
-- gb15629dot11wapiConfigAuthenticationSuites      ::= { wapiMIBObjects 3  }
-- gb15629dot11wapiStats                           ::= { wapiMIBObjects 4  }

-- ******************************************************************************
-- *    gb15629dot11wapiConfig 表开始
-- ******************************************************************************
    gb15629dot11wapiConfig OBJECT-TYPE
          SYNTAX SEQUENCE OF Gb15629dot11wapiConfigEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
            "安全配置属性,以表格形式允许一个代理处有多个实例。"
      ::={ wapiMIBObjects 1 }

    gb15629dot11wapiConfigEntry OBJECT-TYPE
          SYNTAX Gb15629dot11wapiConfigEntry
          MAX-ACCESS not-accessible
          STATUS current
          DESCRIPTION
            "gb15629dot11wapiConfigTable 的一个表项。有可能一个代理处有多个接口,每个接口
            有惟一的 MAC 地址。本规范接口和 Internet 标准 MIB 上下文接口的关系是一一对应
            的,因此 ifIndex 对象实例值能直接用于标识此处定义的相应对象实例。
           ifIndex—每一个本规范接口由一个 ifEntry 来表示。该 MIB 模块中的接口表由 ifIndex
            来索引。"
          INDEX {ifIndex}
```

```
  ::={ gb15629dot11wapiConfig 1 }

gb15629dot11wapiConfigEntry::=
    SEQUENCE {
    gb15629dot11wapiConfigVersion                    Integer32,
    gb15629dot11wapiControlledAuthControl            TruthValue,
    gb15629dot11wapiControlledPortControl            INTEGER,
    gb15629dot11wapiOptionalImplemented              TruthValue,
    gb15629dot11wapiPreauthenticationImplemented     TruthValue,
    gb15629dot11wapiEnabled                          TruthValue,
    gb15629dot11wapiPreauthenticationEnabled         TruthValue,
    gb15629dot11wapiConfigUnicastKeysSupported       Unsigned32,
    gb15629dot11wapiConfigUnicastRekeyMethod         INTEGER,
    gb15629dot11wapiConfigUnicastRekeyTime           Unsigned32,
    gb15629dot11wapiConfigUnicastRekeyPackets        Unsigned32,
    gb15629dot11wapiConfigMulticastCipher            OCTET STRING,
    gb15629dot11wapiConfigMulticastRekeyMethod       INTEGER,
    gb15629dot11wapiConfigMulticastRekeyTime         Unsigned32,
    gb15629dot11wapiConfigMulticastRekeyPackets      Unsigned32,
    gb15629dot11wapiConfigMulticastRekeyStrict       TruthValue,
    gb15629dot11wapiConfigPSKValue                   OCTET STRING,
    gb15629dot11wapiConfigPSKPassPhrase              DisplayString,
    gb15629dot11wapiConfigCertificateUpdateCount     Unsigned32,
    gb15629dot11wapiConfigMulticastUpdateCount       Unsigned32,
    gb15629dot11wapiConfigUnicastUpdateCount         Unsigned32,
    gb15629dot11wapiConfigMulticastCipherSize        Unsigned32,
    gb15629dot11wapiConfigBKLifetime                 Unsigned32,
    gb15629dot11wapiConfigBKReauthThreshold          Unsigned32,
    gb15629dot11wapiConfigSATimeout                  Unsigned32,
    gb15629dot11wapiAuthenticationSuiteSelected      OCTET STRING,
    gb15629dot11wapiUnicastCipherSelected            OCTET STRING,
    gb15629dot11wapiMulticastCipherSelected          OCTET STRING,
    gb15629dot11wapiBKIDUsed                         OCTET STRING,
    gb15629dot11wapiAuthenticationSuiteRequested     OCTET STRING,
    gb15629dot11wapiUnicastCipherRequested           OCTET STRING,
    gb15629dot11wapiMulticastCipherRequested         OCTET STRING  }
gb15629dot11wapiConfigVersion OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "该实体支持的 WAPI 最高版本号。"
    ::= { gb15629dot11wapiConfigEntry 1}
```

```
gb15629dot11wapiControlledAuthControl OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
          "该属性表示实体是否启用鉴别。该值为假,则表示不启用鉴别,受控端口的状态为'已鉴
          别';若为真,则表示启用鉴别,受控端口的状态由端口的控制类型
          gb15629dot11wapiControlledPortControl 来决定。"
    ::= { gb15629dot11wapiConfigEntry 2}

gb15629dot11wapiControlledPortControl OBJECT-TYPE
        SYNTAX INTEGER
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
          "该属性表示实体端口的控制类型,当 gb15629dot11wapiControlledAuthControl 为真时
          有效。值为 0,则表示'自动',受控端口的状态取决于鉴别结果;若为 1,表示'强制未鉴
          别',受控端口的状态无条件为'未鉴别'。"
    ::= { gb15629dot11wapiConfigEntry 3}

gb15629dot11wapiOptionImplemented OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "该变量指示实体是否具有 WAPI 功能。取值为真,表示支持 WAPI 功能;否则,表示
            不支持 WAPI 功能。"
    ::= { gb15629dot11wapiConfigEntry 4  }

gb15629dot11wapiPreauthenticationImplemented OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "该变量指示实体是否支持 WAPI 预鉴别。除非 gb15629dot11wapiOptionImplemented 取值
            为真,否则,该变量不能取值为真。"
  ::= { gb15629dot11wapiConfigEntry 5  }

gb15629dot11wapiEnabled OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
```

```
        DESCRIPTION
            "当为真时,该对象指示在该实体上的 WAPI 已激活。该实体将通过信标帧和探询
            响应帧广播 WAPI 信息元素。"
        ::= { gb15629dot11wapiConfigEntry 6 }

gb15629dot11wapiPreauthenticationEnabled OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "当为真时,该对象指示该实体上的 WAPI 预鉴别已激活。该对象要求将
            dot11WAPI Enabled 也置为真。"
        ::= { gb15629dot11wapiConfigEntry 7 }

gb15629dot11wapiConfigUnicastKeysSupported OBJECT-TYPE
        SYNTAX Unsigned32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
          "该对象指示 WAPI 实体支持的单播密钥个数。"
    ::= { gb15629dot11wapiConfigEntry 8 }

gb15629dot11wapiConfigUnicastRekeyMethod OBJECT-TYPE
        SYNTAX INTEGER { disabled(1), timeBased(2),
                         packetBased(3), timepacket-Based(4) }
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "该对象选择一种机制来重新建立 WAPI 密钥 USK。默认是基于时间的,一天一次。
            USK 的更新由担当 AE 或 ASUE 角色的实体发起。"
        DEFVAL { timeBased }
    ::= { gb15629dot11wapiConfigEntry 9 }

gb15629dot11wapiConfigUnicastRekeyTime OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        UNITS "秒"
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "WAPI USK 需要更新的秒级定时。一旦 USK 通过 MLME-SETWPIKEYS. request
            原语设置后,计时器就开始工作。"
        DEFVAL { 86400 }
    ::= { gb15629dot11wapiConfigEntry 10 }
```

```
gb15629dot11wapiConfigUnicastRekeyPackets OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        UNITS "1000 个分租"
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            " WAPI USK 需要更新的分组计数(以 1000 个分组为单位)。一旦 USK 通过 MLME-
            SETWPIKEYS. request 原语设置后,分组计数器就开始工作,统计采用当前 USK 加
            密的分组个数。"
    ::= { gb15629dot11wapiConfigEntry 11  }

gb15629dot11wapiConfigMulticastCipher OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(4))
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
          "该对象指示实体必须采用的组播密码套件。WAPI 信息元素的组播密码套件应采用该
          变量的值,它包含三个八位位组的 OUI 与一个八位位组的密码套件标识。"
    ::= { gb15629dot11wapiConfigEntry 12  }

gb15629dot11wapiConfigMulticastRekeyMethod OBJECT-TYPE
        SYNTAX INTEGER { disabled(1), timeBased(2),
                         packetBased(3), timepacket-Based(4) }
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "该对象选择一种机制来重新建立 WAPI 密钥 MSK。默认是基于时间的,一天一次。
            MSK 的更新只能由担当 AE 角色的实体发起。"
        DEFVAL { timeBased }
  ::= { gb15629dot11wapiConfigEntry 13  }

gb15629dot11wapiConfigMulticastRekeyTime OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        UNITS "秒"
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "WAPI MSK 需要更新的秒级定时。一旦 MSK 通过 MLME-SETWPIKEYS. request
            原语设置后,计时器就开始工作。"
        DEFVAL { 86400 }
    ::= { gb15629dot11wapiConfigEntry 14  }
```

gb15629dot11wapiConfigMulticastRekeyPackets OBJECT-TYPE
SYNTAX Unsigned32 (1..4294967295)
UNITS "1000 个分租"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"WAPI MSK 需要更新的分组计数(以 1000 个分组为单位)。一旦 MSK 通过 MLME-SETWPIKEYS.request 原语设置后,分组计数器就开始工作,统计采用当前 MSK 加密的分组个数。"
::= { gb15629dot11wapiConfigEntry 15}

gb15629dot11wapiConfigMulticastRekeyStrict OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"不论 STA 什么时候离开拥有密钥 MSK 的 BSS,该对象都会发信号通知密钥 MSK 需要进行更新。"
::= { gb15629dot11wapiConfigEntry 16 }

gb15629dot11wapiConfigPSKValue OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(32))
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"将 PSK 模式选为 WAPI 的 AKM 套件时的 PSK 值。BK 将由该对象推导产生。该对象逻辑上是只写的,若读取该变量将返回不成功的状态或无意义或零。"
::= { gb15629dot11wapiConfigEntry 17 }

gb15629dot11wapiConfigPSKPassPhrase OBJECT-TYPE
SYNTAX DisplayString
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"将 PSK 模式选为 WAPI 的 AKM 套件时的 PSK 值,PSK 通过 gb15629dot11wapiConfigPSKValue 来配置。另一种可替换方法是使用口令一密钥算法来设置 PSK。该变量提供了一种输入口令的方法 。当该对象被写入时,WAPI 实体将使用口令一密钥算法导出一个预共享密钥,同时用导出的密钥填充 gb15629dot11wapiConfigPSKValue。该对象逻辑上是只写的。读取该对象将返回不成功状态或无意义或零。"
::= { gb15629dot11wapiConfigEntry 18 }

gb15629dot11wapiConfigCertificateUpdateCount OBJECT-TYPE

```
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "在每次证书鉴别握手中,WAPI 证书鉴别握手协议的消息重传次数。"
          DEFVAL { 3 }
      ::= { gb15629dot11wapiConfigEntry 19  }

  gb15629dot11wapiConfigMulticastUpdateCount OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "在每次 MSK 握手中,WAPI 组播密钥握手协议的消息重传的次数。"
        DEFVAL { 3 }
      ::= { gb15629dot11wapiConfigEntry 20  }

  gb15629dot11wapiConfigUnicastUpdateCount OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "在每次 3 次握手中,WAPI 单播密钥握手协议的消息重传的次数。"
        DEFVAL { 3 }
      ::= { gb15629dot11wapiConfigEntry 21  }

  gb15629dot11wapiConfigMulticastCipherSize OBJECT-TYPE
        SYNTAX Unsigned32 (0..4294967295)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
          "该对象指示组播密钥长度的位数。对于 SMS4 来说,长度值应该为 256。前面 128 位为
          加密密钥,后面 128 位为完整性校验密钥。"
      ::= { gb15629dot11wapiConfigEntry 22  }

gb15629dot11wapiConfigBKLifetime OBJECT-TYPE
      SYNTAX Unsigned32 (1..4294967295)
      UNITS "秒"
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
          "BK 缓存中,一个 BK 的最大生存期。"
      DEFVAL { 43200 }
```

```
  ::= { gb15629dot11wapiConfigEntry 23 }

gb15629dot11wapiConfigBKReauthThreshold OBJECT-TYPE
      SYNTAX Unsigned32 (1..100)
      UNITS "百分比"
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
          "在重新进行证书鉴别之前,应当渡过 BK 生存期的百分比。"
      DEFVAL { 70 }
  ::= { gb15629dot11wapiConfigEntry 24 }

gb15629dot11wapiConfigSATimeout OBJECT-TYPE
      SYNTAX Unsigned32 (1..4294967295)
      UNITS "秒"
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
          "安全关联建立需要的最长时间。"
      DEFVAL { 60 }
  ::= { gb15629dot11wapiConfigEntry 25 }

gb15629dot11wapiAuthenticationSuiteSelected OBJECT-TYPE
      SYNTAX OCTET STRING (SIZE(4))
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
          "最后一次协商的 AKM 套件。"
  ::= { gb15629dot11wapiConfigEntry 26 }

gb15629dot11wapiUnicastCipherSelected OBJECT-TYPE
      SYNTAX OCTET STRING (SIZE(4))
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
          "最后一次协商的单播密码套件。"
  ::= { gb15629dot11wapiConfigEntry 27 }

gb15629dot11wapiMulticastCipherSelected OBJECT-TYPE
      SYNTAX OCTET STRING (SIZE(4))
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
```

```
            "最后一次协商的组播密码套件。"
    ::= { gb15629dot11wapiConfigEntry 28 }

  gb15629dot11wapiBKIDUsed OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(16))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "最后一次单播密钥握手过程中所使用的 BKID。"
    ::= { gb15629dot11wapiConfigEntry 29 }

  gb15629dot11wapiAuthenticationSuiteRequested OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(4))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "最后一次请求的 AKM 套件。"
    ::= { gb15629dot11wapiConfigEntry 30 }

  gb15629dot11wapiUnicastCipherRequested OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(4))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "最后一次请求的单播密码套件。"
    ::= { gb15629dot11wapiConfigEntry 31 }

  gb15629dot11wapiGroupCipherRequested OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(4))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "最后一次请求的组播密码套件。"
    ::= { gb15629dot11wapiConfigEntry 32 }
-- ******************************************************************
-- *  gb15629dot11wapiConfig 组结束
-- ******************************************************************

-- ******************************************************************
-- * gb15629dot11wapiConfigUnicastCiphers 表开始
-- ******************************************************************
  gb15629dot11wapiConfigUnicastCiphers OBJECT-TYPE
        SYNTAX SEQUENCE OF Gb15629dot11wapiConfigUnicastCiphersEntry
```

```
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "该表格列出了实体支持的单播密码套件。允许通过网络管理激活和禁止每一个单播密
            码套件。WAPI 信息元素中的单播密码套件列表是通过该表格中的信息构成的。"
    ::= { wapiMIBObjects 2 }

gb15629dot11wapiConfigUnicastCiphersEntry OBJECT-TYPE
        SYNTAX Gb15629dot11wapiConfigUnicastCiphersEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "该表格的条目是由接口索引(或所有接口)和单播密码套件来索引的。"
        INDEX { gb15629dot11wapiConfigIndex, gb15629dot11wapiConfigUnicastCipherIndex }
    ::= { gb15629dot11wapiConfigUnicastCiphersTable 1 }

Gb15629dot11wapiConfigUnicastCiphersEntry ::=
    SEQUENCE {
    gb15629dot11wapiConfigUnicastCipherIndex          Unsigned32,
    gb15629dot11wapiConfigUnicastCipher               OCTET STRING,
    gb15629dot11wapiConfigUnicastCipherEnabled        TruthValue,
    gb15629dot11wapiConfigUnicastCipherSize           Unsigned32 }

gb15629dot11wapiConfigUnicastCipherIndex OBJECT-TYPE
    SYNTAX Unsigned32 (1..4294967295)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
      " gb15629dot11wapiConfigUnicastCiphersTable 的其他辅助索引。"
    ::= { gb15629dot11wapiConfigUnicastCiphersEntry 1 }

gb15629dot11wapiConfigUnicastCipher OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE(4))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "所支持的单播密码套件。它包括一个 OUI(前三个八位位组)和一个密码套件标识(最后
      一个八位位组)。"
    ::= { gb15629dot11wapiConfigUnicastCiphersEntry 2 }

gb15629dot11wapiConfigUnicastCipherEnabled OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
```

```
        STATUS current
        DESCRIPTION
            "该对象激活或禁止单播密码。"
    ::= { gb15629dot11wapiConfigUnicastCiphersEntry 3 }

  gb15629dot11wapiConfigUnicastCipherSize OBJECT-TYPE
        SYNTAX Unsigned32 (0..4294967295)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "该对象指示单播密钥长度的位数。对于 SMS4 来说,长度值应该为 256。前面 128 位为
            加密密钥,后面 128 位为完整性校验密钥。"
    ::= { gb15629dot11wapiConfigUnicastCiphersEntry 4 }
--- ***********************************************************************
--- * gb15629dot11wapiConfigUnicastCiphers 表结束
--- ***********************************************************************

--- ***********************************************************************
--- * gb15629dot11wapiConfigAuthenticationSuites  表开始
--- ***********************************************************************
  gb15629dot11wapiConfigAuthenticationSuites OBJECT-TYPE
        SYNTAX SEQUENCE OF Gb15629dot11wapiConfigAuthenticationSuitesEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "该表格列出了实体支持的 AKM 套件。每一种 AKM 套件可以单独激活或禁止。WAPI
            信息元素中的 AKM 套件列表是由该表格的信息构成的。"
    ::= { wapiMIBObjects 3  }

  gb15629dot11wapiConfigAuthenticationSuitesEntry OBJECT-TYPE
        SYNTAX Gb15629dot11wapiConfigAuthenticationSuitesEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "表格 gb15629dot11wapiConfigAuthenticationSuitesTable 中的一个条目。"
        INDEX { gb15629dot11wapiConfigAuthenticationSuiteIndex }
    ::= { gb15629dot11wapiConfigAuthenticationSuitesTable 1 }

  Gb15629dot11wapiConfigAuthenticationSuitesEntry ::=
        SEQUENCE {
        gb15629dot11wapiConfigAuthenticationSuiteIndex        Unsigned32,
        gb15629dot11wapiConfigAuthenticationSuite             OCTET STRING,
        gb15629dot11wapiConfigAuthenticationSuiteEnabled      TruthValue }
```

```
gb15629dot11wapiConfigAuthenticationSuiteIndex OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "该辅助变量被用作表格 gb15629dot11wapiConfigAuthenticationSuitesTable 的一个
            索引。"
    ::= { gb15629dot11wapiConfigAuthenticationSuitesEntry 1 }

gb15629dot11wapiConfigAuthenticationSuite OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(4))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "AKM 套件。它包括一个 OUI(前三个八位位组)和一个密码套件标识(最后一个八位位
            组)。"
    ::= { gb15629dot11wapiConfigAuthenticationSuitesEntry 2 }

gb15629dot11wapiConfigAuthenticationSuiteEnabled OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-write
        STATUS current
        DESCRIPTION
            "该变量指示相应的 AKM 套件是处于激活还是禁止状态。"
    ::= { gb15629dot11wapiConfigAuthenticationSuitesEntry 3 }
-- ******************************************************************
-- * gb15629dot11wapiConfigAuthenticationSuites  表结束
-- ******************************************************************

-- ********************************************************************
-- * gb15629dot11wapiStats  表开始
-- ********************************************************************
gb15629dot11wapiStats OBJECT-TYPE
        SYNTAX SEQUENCE OF Gb15629dot11wapiStatsEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "该表维护 WAPI 的每个 STA 的统计指标。gb15629dot11wapiStatsSTAAddress 设为
            FF-FF-FF-FF-FF-FF 的条目应包含广播/组播业务。"
    ::= { wapiMIBObjects 4  }

gb15629dot11wapiStatsEntry OBJECT-TYPE
```

```
        SYNTAX Gb15629dot11wapiStatsEntry
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "表格 gb15629dot11wapiStatsTable 中的一个条目。"
        INDEX { gb15629dot11wapiConfigIndex,gb15629dot11wapiStatsIndex }
    ::= { gb15629dot11wapiStats 1 }

Gb15629dot11wapiStatsEntry ::=
        SEQUENCE {
        gb15629dot11wapiStatsIndex                          Unsigned32,
        gb15629dot11wapiStatsSTAAddress                     MacAddress,
        gb15629dot11wapiStatsVersion                        Unsigned32,
        gb15629dot11wapiStatsControlledPortStatus           TruthValue,
        gb15629dot11wapiStatsSelectedUnicastCipher          OCTET STRING,
        gb15629dot11wapiStatsWPIReplayCounters              Counter32,
        gb15629dot11wapiStatsWPIDecryptableErrors           Counter32,
        gb15629dot11wapiStatsWPIMICErrors                   Counter32,
        gb15629dot11wapiStatsWAISignatureErrors             Counter32,
        gb15629dot11wapiStatsWAIHMACErrors                  Counter32,
        gb15629dot11wapiStatsWAIAuthenticationResultFailures Counter32,
        gb15629dot11wapiStatsWAIDiscardCounters             Counter32,
        gb15629dot11wapiStatsWAITimeoutCounters             Counter32,
        gb15629dot11wapiStatsWAIFormatErrors                Counter32,
        gb15629dot11wapiStatsWAICertificateHandshakeFailures Counter32,
        gb15629dot11wapiStatsWAIUnicastHandshakeFailures    Counter32,
        gb15629dot11wapiStatsWAIMulticastHandshakeFailures  Counter32}
gb15629dot11wapiStatsIndex OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS not-accessible
        STATUS current
        DESCRIPTION
            "表格 gb15629dot11wapiStatsTable 的一个辅助索引。"
    ::= { gb15629dot11wapiStatsEntry 1 }

gb15629dot11wapiStatsSTAAddress OBJECT-TYPE
        SYNTAX MacAddress
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "STA 的 MAC 地址。"
    ::= { gb15629dot11wapiStatsEntry 2 }
```

```
gb15629dot11wapiStatsVersion OBJECT-TYPE
        SYNTAX Unsigned32 (1..4294967295)
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "STA 所关联的 WAPI 版本。"
  ::= { gb15629dot11wapiStatsEntry 3 }

gb15629dot11wapiStatsControlledPortStatus OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "鉴别子系统实体的受控端口的状态。取值为真,表示'已鉴别';否则,表示'未鉴别'。"
  ::= { gb15629dot11wapiStatsEntry 4 }

gb15629dot11wapiStatsSelectedUnicastCipher OBJECT-TYPE
        SYNTAX OCTET STRING (SIZE(4))
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "关联过程中所使用的单播密码套件。"
  ::= { gb15629dot11wapiStatsEntry 5 }

gb15629dot11wapiStatsWPIReplayCounter OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "根据重放机制丢弃的 WPI MPDU 的数目。"
  ::= { gb15629dot11wapiStatsEntry 6 }

gb15629dot11wapiStatsWPIDecryptableErrors OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "WPI-SMS4 解密时没有有效的密钥而丢弃的 MPDU 数目。"
  ::= { gb15629dot11wapiStatsEntry 7 }

gb15629dot11wapiStatsWPIMICErrors OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
```

```
        STATUS current
        DESCRIPTION
            "WPI-SMS4 解密时 MIC 校验出错而丢弃的 MPDU 数目。"
    ::= { gb15629dot11wapiStatsEntry 8 }

gb15629dot11wapiStatsWAISignatureErrors OBJECT-TYPE
        SYNTAX  Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "接收到的 WAI 分组验证签名错时,该计数器应递增。"
    ::= { gb15629dot11wapiStatsEntry 9 }

gb15629dot11wapiStatsWAIHMACErrors OBJECT-TYPE
        SYNTAX  Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当接收到的 WAI 分组消息鉴别码校验出错时,该计数器应递增。"
    ::= { gb15629dot11wapiStatsEntry 10}

gb15629dot11wapiStatsWAIAuthenticationResultFailures  OBJECT-TYPE
        SYNTAX  Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当 WAI 鉴别结果不成功时,该计数器应递增。"
    ::= { gb15629dot11wapiStatsEntry 11}

gb15629dot11wapiStatsWAIDiscardCounters  OBJECT-TYPE
        SYNTAX  Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当接收到的 WAI 分组被丢弃时,该计数器应递增。"
    ::= { gb15629dot11wapiStatsEntry 12}

gb15629dot11wapiStatsWAITimeoutCounters OBJECT-TYPE
        SYNTAX  Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "当检测到 WAI 分组超时时,该计数器应递增。"
```

```
  ::={ gb15629dot11wapiStatsEntry 13}

gb15629dot11wapiStatsWAIFormatErrors   OBJECT-TYPE
      SYNTAX   Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "当检测到 WAI 分组格式出错时,该计数器应递增。"
  ::={ gb15629dot11wapiStatsEntry 14}

gb15629dot11wapiStatsWAICertificateHandshakeFailures   OBJECT-TYPE
      SYNTAX   Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "当 WAI 证书鉴别过程失败时,该计数器应递增。"
  ::={ gb15629dot11wapiStatsEntry 15}

gb15629dot11wapiStatsWAIUnicastHandshakeFailures   OBJECT-TYPE
      SYNTAX   Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "当 WAI 单播密钥协商过程失败时,该计数器应递增。"
  ::={ gb15629dot11wapiStatsEntry 16}

gb15629dot11wapiStatsWAIMulticastHandshakeFailures   OBJECT-TYPE
      SYNTAX   Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "当 WAI 组播密钥通告过程失败时,该计数器应递增。"
  ::={ gb15629dot11wapiStatsEntry 17}
-- ***********************************************************************
-- * gb15629dot11wapiStats  表结束
-- ***********************************************************************

-- ****************************************
-- *   GB15629.11 WAPI MIB  - 一致性信息
-- ****************************************
gb15629dot11wapiGroups OBJECT IDENTIFIER ::= { wapiMIBConformance 1 }
gb15629wapiCompliances OBJECT IDENTIFIER::= { wapiMIBConformance 2 }
```

```
-- ******************************************************************
-- * 一致性信息 - WAPI
-- ******************************************************************

gb15629dot11wapiCompliance MODULE-COMPLIANCE
        STATUS current
        DESCRIPTION
            "实现本规范 WAPI MIB 的 SNMPv2 实体的一致性声明。"
        MODULE
        MANDATORY-GROUPS {
        gb15629dot11wapiBase }
        -- OPTIONAL-GROUPS {gb15629dot11wapiBKcachingGroup }
    ::= { gb15629WapiCompliances 1  }

-- ******************************************************************
-- * 组- 一致性单元 - WAPI
-- ******************************************************************
gb15629dot11wapiBase OBJECT-GROUP
        OBJECTS {
            gb15629dot11wapiConfigVersion,
            gb15629dot11wapiControlledAuthControl,
            gb15629dot11wapiControlledPortControl,
            gb15629dot11wapiOptionalImplemented,
            gb15629dot11wapiPreauthenticationImplemented,
            gb15629dot11wapiEnabled,
            gb15629dot11wapiPreauthenticationEnabled,
            gb15629dot11wapiConfigUnicastKeysSupported,
            gb15629dot11wapiConfigUnicastRekeyMethod,
            gb15629dot11wapiConfigUnicastRekeyTime,
            gb15629dot11wapiConfigUnicastRekeyPackets,
            gb15629dot11wapiConfigMulticastCipher,
            gb15629dot11wapiConfigMulticastRekeyMethod,
            gb15629dot11wapiConfigMulticastRekeyTime,
            gb15629dot11wapiConfigMulticastRekeyPackets,
            gb15629dot11wapiConfigMulticastRekeyStrict,
            gb15629dot11wapiConfigPSKValue,
            gb15629dot11wapiConfigPSKPassPhrase,
            gb15629dot11wapiConfigCertificateUpdateCount,
            gb15629dot11wapiConfigMulticastUpdateCount,
            gb15629dot11wapiConfigUnicastUpdateCount,
            gb15629dot11wapiConfigMulticastCipherSize,
            gb15629dot11wapiConfigUnicastCipher,
            gb15629dot11wapiConfigUnicastCipherEnabled,
            gb15629dot11wapiConfigUnicastCipherSize,
```

```
            gb15629dot11wapiConfigAuthenticationSuite,
            gb15629dot11wapiConfigAuthenticationSuiteEnabled,
            gb15629dot11wapiConfigSATimeout,
            gb15629dot11wapiAuthenticationSuiteSelected,
            gb15629dot11wapiUnicastCipherSelected,
            gb15629dot11wapiMulticastCipherSelected,
            gb15629dot11wapiBKIDUsed,
            gb15629dot11wapiAuthenticationSuiteRequested,
            gb15629dot11wapiUnicastCipherRequested,
            gb15629dot11wapiMulticastCipherRequested,
            gb15629dot11wapiStatsSTAAddress,
            gb15629dot11wapiStatsVersion,
            gb15629dot11wapiStatsControlledPortStatus,
            gb15629dot11wapiStatsSelectedUnicastCipher,
            gb15629dot11wapiStatsWPIReplayCounters,
            gb15629dot11wapiStatsWPIDecryptableErrors,
            gb15629dot11wapiStatsWPIMICErrors,
            gb15629dot11wapiStatsWAISignatureErrors,
            gb15629dot11wapiStatsWAIHMACErrors,
            gb15629dot11wapiStatsWAIAuthenticationResultFailures,
            gb15629dot11wapiStatsWAIDiscardCounters,
            gb15629dot11wapiStatsWAITimeoutCounters,
            gb15629dot11wapiStatsWAIFormatErrors,
            gb15629dot11wapiStatsWAICertificateHandshakeFailures,
            gb15629dot11wapiStatsWAIUnicastHandshakeFailures,
            gb15629dot11wapiStatsWAIMulticastHandshakeFailures
            }
        STATUS current
        DESCRIPTION
            " gb15629dot11wapiBase 对象类提供了必要的支持管理 STA 的 WAPI 功能。"
    ::= { gb15629dot11WapiGroups 28 }

gb15629dot11wapiBKcachingGroup OBJECT-GROUP
                OBJECTS {gb15629dot11wapiConfigBKLifetime,
                         gb15629dot11wapiConfigBKReauthThreshold}
        STATUS current
        DESCRIPTION
            " gb15629dot11wapiBKcachingGroup 对象类提供了必要的支持管理 STA 的 BK 缓存功能。"
    ::= { gb15629dot11WapiGroups 29 }

-- ******************************************************************************
-- *   GB15629dot11-WAPI-MIB 结束
-- ******************************************************************************
END
```

115、在附录D的后面增加附录E。

附 录 E
（资料性附录）
消息鉴别算法和密钥导出算法的参考实现及测试向量

E.1 消息鉴别算法

E.1.1 参考实现(C语言)

```
#include <string.h>
#define SHA256_BLOCK_SIZE 64
#define SHA256_DIGEST_SIZE 32
typedef unsigned char byte;
typedef struct contxt
{
    byte *buff;
    unsigned length;
} CONTX;

int sha256(CONTX*,int,byte*);

int hmac_sha256(unsigned char *text, int text_len, byte *key, unsigned key_len, byte *digest,
unsigned digest_length)
/*
    a)  unsigned char *text 表示进行 HMAC 运算的文本;
    b)  unsigned text_len 表示进行 HMAC 运算的文本的长度(八位位组数);
    c)  byte *key 表示进行 HMAC 运算的密钥;
    d)  unsigned key_len 表示进行 HMAC 运算的密钥的长度(八位位组数);
    e)  byte *digest 表示进行 HMAC 运算输出的摘要;
    f)  unsigned digest_length 表示进行 HMAC 运算,要求输出的摘要长度(八位位组数),必须小
        于或等于 sha-256 杂凑算法可以输出的摘要长度(八位位组数)。
    g)  返回值,非 0 表示实际输出的摘要长度(八位位组数),0 表示失败。
*/
{
    byte real_key[SHA256_BLOCK_SIZE];
    byte ipad[SHA256_BLOCK_SIZE];
    byte opad[SHA256_BLOCK_SIZE];
    byte temp_digest1[SHA256_DIGEST_SIZE];
    byte temp_digest2[SHA256_DIGEST_SIZE];
    CONTX input_data[2];
    unsigned i;
```

```
    if (digest_length>SHA256_DIGEST_SIZE)
        return 0;

    for(i=0;i< SHA256_BLOCK_SIZE;i++){
        real_key[i]=0;
        ipad[i]=0x36;
        opad[i]=0x5c;
    }

/* if key_len is larger than hash block size, key is hashed first to make its length is equal hash block
size */
    if(key_len> SHA256_BLOCK_SIZE){
        input_data[0].buff=key;
        input_data[0].length=key_len;
        SHA256(input_data,1,real_key);
        key_len= SHA256_BLOCK_SIZE;
    }
    else
        memcpy(real_key,key,key_len);

    for(i=0;i< SHA256_BLOCK_SIZE;i++){
        ipad[i]^=real_key[i];
        opad[i]^=real_key[i];
    }
    /* sha256(Key xor ipad,text)=temp_digest1 */
    input_data[0].buff=ipad;
    input_data[0].length= SHA256_BLOCK_SIZE;
    input_data[1].buff=text;
    input_data[1].length=text_len;
    SHA256(input_data,2,temp_digest1);

    /* sha256(Key xor opad,temp_digest1)=temp_digest2 */
    input_data[0].buff=opad;
    input_data[0].length= SHA256_BLOCK_SIZE;
    input_data[1].buff=temp_digest1;
    input_data[1].length =SHA256_DIGEST_SIZE;
    SHA256(input_data,2,temp_digest2);

    /* output the digest of required length */
    memcpy(digest,temp_digest2,digest_length);
    return digest_length;
}
```

E.1.2 测试向量

测试向量 1	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20
密钥长度	32
数据	abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqabcdbcdecdefdefgefghfghighijhijkijkljklm klmnlmnomnopnopq 0x61 62 63 64 62 63 64 65 63 64 65 66 64 65 66 67 65 66 67 68 66 67 68 69 67 68 69 6a 68 69 6a 6b 69 6a 6b 6c 6a 6b 6c 6d 6b 6c 6d 6e 6c 6d 6e 6f 6d 6e 6f 70 6e 6f 70 71 61 62 63 64 62 63 64 65 63 64 65 66 64 65 66 67 65 66 67 68 66 67 68 69 67 68 69 6a 68 69 6a 6b 69 6a 6b 6c 6a 6b 6c 6d 6b 6c 6d 6e 6c 6d 6e 6f 6d 6e 6f 70 6e 6f 70 71
数据长度	112
摘要	0x47 03 05 fc 7e 40 fe 34 d3 ee b3 e7 73 d9 5a ab 73 ac f0 fd 06 04 47 a5 eb 45 95 bf 33 a9 d1 a3

测试向量 2	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
密钥长度	37
数据	0xcd 重复 50 次
数据长度	50
摘要	0xd4 63 3c 17 f6 fb 8d 74 4c 66 de e0 f8 f0 74 55 6e c4 af 55 ef 07 99 85 41 46 8e b4 9b d2 e9 17

测试向量 3	
密钥	0x0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b
密钥长度	32
数据	Hi There 0x48 69 20 54 68 65 72 65
数据长度	8
摘要	0x19 8a 60 7e b4 4b fb c6 99 03 a0 f1 cf 2b bd c5 ba 0a a3 f3 d9 ae 3c 1c 7a 3b 16 96 a0 b6 8c f7

测试向量 4	
密钥	0x4a 65 66 65
密钥长度	4
数据	what do ya want for nothing? 0x77 68 61 74 20 64 6f 20 79 61 20 77 61 6e 74 20 66 6f 72 20 6e 6f 74 68 69 6e 67 3f
数据长度	28
摘要	0x5b dc c1 46 bf 60 75 4e 6a 04 24 26 08 95 75 c7 5a 00 3f 08 9d 27 39 83 9d ec 58 b9 64 ec 38 43

E.2 密钥导出算法

E.2.1 参考实现

```
#define SHA256_BLOCK_SIZE 64
#define SHA256_DIGEST_SIZE 32

int hmac_sha256(unsigned char, int, byte, unsigned , byte * , unsigned);

void KD_hmac_sha256(byte * text, unsigned text_len, byte * key, unsigned key_len, byte * out-
```

```
put, unsigned length)
/*
    a)  byte *text 表示密钥导出算法的输入文本;
    b)  unsigned text_len 表示输入文本的长度(八位位组数);
    c)  byte *key  表示密钥导出算法的输入密钥;
    d)  unsigned key_len  表示输入密钥的长度(八位位组数);
    e)  byte *output  表示密钥导出算法的输出;
    f)  unsigned length  表示密钥导出算法的输出的长度(八位位组数)。
*/
{
    int i;
    for(i=0;length/SHA256_DIGEST_SIZE;i++,length-=SHA256_DIGEST_SIZE){
                hmac_sha256(text, text_len, key, key_len, &output[i * SHA256_DIGEST_
            SIZE],SHA256_DIGEST_SIZE);
        text=&output[i * SHA256_DIGEST_SIZE];
        text_len=SHA256_DIGEST_SIZE;
    }
    if(length>0)
            hmac_sha256(text, text_len, key, key_len, &output[i * SHA256_DIGEST_SIZE],
            length);
}
```

E.2.2 测试向量

测试向量 1	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20
密钥长度	32
数据	pairwise key expansion for infrastructure unicast 0x70 61 69 72 77 69 73 65 20 6b 65 79 20 65 78 70 61 6e 73 69 6f 6e 20 66 6f 72 20 69 6e 66 72 61 73 74 72 75 63 74 75 72 65 20 75 6e 69 63 61 73 74
数据长度	49
输出	0xe3 a6 45 46 f2 d1 f5 ee b7 d1 ee 06 d2 c9 e5 4a 2c c9 d6 ce c3 b7 6f fd 62 63 f4 26 dc 25 39 af bd 98 80 a5 27 a1 b5 85 59 4b 57 ce 33 21 4f 0c
输出长度	48

测试向量 2	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
密钥长度	37
数据	pairwise key expansion for infrastructure unicast 0x70 61 69 72 77 69 73 65 20 6b 65 79 20 65 78 70 61 6e 73 69 6f 6e 20 66 6f 72 20 69 6e 66 72 61 73 74 72 75 63 74 75 72 65 20 75 6e 69 63 61 73 74
数据长度	49
输出	0x3b 6e ca 4f 08 76 c4 3a b3 1b 26 3f 2c 38 b8 81 21 b5 68 e5 f8 fd 1d 4c fa 4c 7f 8c 60 97 04 3d 7b 40 a8 63 b9 43 b9 f5 bb 37 2f 3a dd a5 da 27
输出长度	48

测试向量 3	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15
密钥长度	16
数据	pairwise key expansion for infrastructure unicast 0x70 61 69 72 77 69 73 65 20 6b 65 79 20 65 78 70 61 6e 73 69 6f 6e 20 66 6f 72 20 69 6e 66 72 61 73 74 72 75 63 74 75 72 65 20 75 6e 69 63 61 73 74
数据长度	49
输出	0xbc 29 f3 e6 09 1f 6a c9 0b a0 20 61 92 12 48 69 5f ee ff 1a 4c ab 53 3b 11 67 d8 54 5f 93 5f 28 11 84 c9 bb 32 f9 87 b9 86 81 0f fb 17 c4 10 f5
输出长度	48

测试向量 4	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20
密钥长度	32
数据	group key expansion for multicast and broadcast 0x67 72 6f 75 70 20 6b 65 79 20 65 78 70 61 6e 73 69 6f 6e 20 66 6f 72 20 6d 75 6c 74 69 63 61 73 74 20 61 6e 64 20 62 72 6f 61 64 63 61 73 74
数据长度	47
输出	0x20 8f 72 54 a4 bf 56 f0 fa 49 5f e1 0c 99 15 05 92 ed 79 df 57 74 a9 6e 13 97 1e c4 a1 5e 16 a7 ed 75 f5 e5 44 bb d3 35 67 eb 88 e7 83 24 a9 d2
输出长度	48

测试向量 5	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
密钥长度	37
数据	group key expansion for multicast and broadcast 0x67 72 6f 75 70 20 6b 65 79 20 65 78 70 61 6e 73 69 6f 6e 20 66 6f 72 20 6d 75 6c 74 69 63 61 73 74 20 61 6e 64 20 62 72 6f 61 64 63 61 73 74
数据长度	47
输出	0x33 32 61 7a 90 8e a5 a0 7f fa 1d 23 79 f3 d8 3e 8b e9 14 1f 15 53 8f d3 ef de 58 01 19 e8 c5 09 5d 25 b2 d3 0a c7 a6 35 ad b4 3c 6c ac f0 aa 2b
输出长度	48

测试向量 6	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15
密钥长度	16
数据	group key expansion for multicast and broadcast 0x67 72 6f 75 70 20 6b 65 79 20 65 78 70 61 6e 73 69 6f 6e 20 66 6f 72 20 6d 75 6c 74 69 63 61 73 74 20 61 6e 64 20 62 72 6f 61 64 63 61 73 74
数据长度	47
输出	0xf2 cb f1 1c 6d 40 b8 09 d0 c0 ed 48 2a 4a 1b 6a 15 1a f1 fb 4c 80 f9 80 5c 93 e5 6e b1 cf 5c b5 ec c1 3e 7a bc af e0 a7 d2 59 5d 51 9b 76 9a 24
输出长度	48

测试向量 7	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20
密钥长度	32
数据	pre-share key expansion for adhoc network 0x70 72 65 2d 73 68 61 72 65 20 6b 65 79 20 65 78 70 61 6e 73 69 6f 6e 20 66 6f 72 20 61 64 68 6f 63 20 6e 65 74 77 6f 72 6b
数据长度	41
输出	0xc0 7a d8 32 25 2a 0c 14 76 18 f4 c0 d0 6b 35 f4 f6 d6 73 5d 1a a3 8e 47 9a 7e e0 ac 1c 0c 38 5b 2d 33 28 74 1e 4d a0 c8 76 fc 6c c9 e3 60 c8 d7
输出长度	48

测试向量 8	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
密钥长度	37
数据	pre-share key expansion for adhoc network 0x70 72 65 2d 73 68 61 72 65 20 6b 65 79 20 65 78 70 61 6e 73 69 6f 6e 20 66 6f 72 20 61 64 68 6f 63 20 6e 65 74 77 6f 72 6b
数据长度	41
输出	0x f0 0b ee f2 f5 5f 85 d8 ee b0 6f 8c c4 1b e6 0e c2 69 f5 82 9a 0b 6e fb 2d 9b 49 5e b1 87 d3 58 59 68 88 c3 d2 6f 94 9f 8d 2e 41 fe bc bb b9 9a
输出长度	48

测试向量 9	
密钥	0x01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15
密钥长度	16
数据	pre-share key expansion for adhoc network 0x70 72 65 2d 73 68 61 72 65 20 6b 65 79 20 65 78 70 61 6e 73 69 6f 6e 20 66 6f 72 20 61 64 68 6f 63 20 6e 65 74 77 6f 72 6b
数据长度	41
输出	0x05 8e b8 7c ff 82 66 47 de 50 7b 14 17 ac 99 6e b5 7f cf 11 fd fc 83 be 59 d5 85 f4 a7 3e 69 7d d4 38 e3 34 fe bb 06 7d 14 6f 01 31 a6 96 4f 26
输出长度	48