



中华人民共和国国家标准

GB/T 28455—2012

信息安全技术 引入可信第三方的实体 鉴别及接入架构规范

Information security technology—Entity authentication involving a trusted
third party and access architecture specification

2012-06-29 发布

2012-10-01 实施

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

目 次

前言	III
引言	IV
1 范围	1
2 规范性引用文件	1
3 术语和定义	2
4 缩略语	2
5 引入可信第三方的实体鉴别及接入架构	3
5.1 概述	3
5.2 访问控制的范围	4
5.3 系统、角色和端口	4
5.4 端口访问实体(PAE)	8
5.5 IEEE Std 802.3-2005 中端口访问控制的使用	15
6 链路上的 TAEP 封装(TAEPoL)	15
6.1 概述	15
6.2 八位位组的发送和标识	15
6.3 TAEPoL MPDU 在 GB/T 15629.2(IEEE 802.2)逻辑链路控制(LLC)中的格式	16
6.4 TAEPoL MPDU 在 GB/T 15629.3(IEEE 802.3)中的格式	16
6.5 标签 TAEPoL MPDU	17
6.6 TAEPoL PDU 的格式	17
6.7 接收到 TAEPoL PDU 和 TAEPoL 协议格式处理的确认	21
7 对等鉴别访问控制协议	21
7.1 概述	21
7.2 鉴别过程	22
7.3 PCAP 状态机	23
8 端口接入控制管理	47
8.1 一般要求	47
8.2 管理功能	47
8.3 被管对象	48
8.4 数据类型	48
8.5 鉴别访问控制器 PAE 被管对象	49
8.6 请求者 PAE 管理对象	54
8.7 系统管理对象	57
9 端口接入控制 MIB 定义	58
附录 A (规范性附录) PICS 形式表	85
附录 B (资料性附录) 基于 TAEP 封装的鉴别协议	91

附录 C (资料性附录) 适用于无线城域网的 TAAA 机制	116
附录 D (资料性附录) 局域网媒体访问控制技术	136
附录 E (资料性附录) 单向控制功能的考虑	219
参考文献	221



前 言

本标准按照 GB/T 1.1—2009 给出的规则起草。

本标准由全国信息安全标准化技术委员会(SAC/TC 260)提出并归口。

本标准主要起草单位:西安西电捷通无线网络通信股份有限公司、国家密码管理局商用密码检测中心、信息安全国家重点实验室、中国电子技术标准化研究所、国家无线电监测中心检测中心、西安电子科技大学、西安邮电学院、广州杰赛科技股份有限公司、深圳市明华澳汉科技股份有限公司、中国信息安全认证中心、国家信息安全工程技术研究中心、国家计算机网络应急技术处理协调中心、国家信息技术安全研究中心、公安部第一研究所、工业和信息化部通信计量中心、公安部信息安全等级保护评估中心、国防科技大学、北京市政务网络管理中心、重庆邮电大学、宇龙计算机通信科技(深圳)有限公司、中国人民大学、中国人民解放军信息安全测评认证中心、中国电信集团公司、国家信息中心、北京大学深圳研究生院、中国电力科学研究院、北京中电华大电子设计有限责任公司、东南大学、中国移动通信集团设计院有限公司、中国人民解放军信息工程大学、江南计算技术研究所、北京邮电大学、上海龙照电子有限公司、北京五龙电信技术公司、北京网贝合创科技有限公司、深圳市宏电技术股份有限公司、北大方正集团公司、海尔集团公司、北京广信融科技术有限公司、北京六合万通微电子技术有限公司、弘浩明传科技(北京)有限公司、北京城市热点资讯有限公司、北京华安广通科技发展有限公司、迈普通信技术有限公司、长春吉大正元信息技术股份有限公司、清华大学、北京天一集成科技有限公司、桂林电子工业学院、西安立人科技股份有限公司、宽带无线 IP 标准工作组、WAPI 产业联盟等。

本标准主要起草人:黄振海、赖晓龙、李大为、冯登国、宋起柱、铁满霞、曹军、李建东、李宁、舒敏、朱志祥、陈晓桦、郭晓雷、李京春、余亚莉、王育民、张变玲、肖跃雷、高波、高昆仑、潘峰、胡亚楠、蒋庆生、肖雳、朱建平、贾焰、施伟年、李琴、李广森、吴亚非、梁朝晖、梁琼文、罗旭光、龙昭华、沈凌云、张伟、徐平平、马华兴、高峰、仇洪冰、朱跃生、王雅辉、兰天、王志坚、杜志强、张国强、田小平、田辉、张永强、寿国梁、毛立平、曹竹青、郭志刚、高宏、韩康、王钢、白国强、陈志峰、李建良、李大伟、王立仁、高原、岳林、井京涛。

引 言

网络通信经常处于这样的环境,非授权的终端设备可以物理地连接到网络上,授权的终端设备所连接的网络也不一定是它所期望的,因此在终端和网络通信前,需要通过鉴别和授权功能互相鉴别对方身份的合法性,以保证通信的安全。对此通信和信息技术业界一直在寻找经济有效的安全解决方案,安全的网络应受到保护,免遭恶意和无意的攻击,并且应满足业务对信息和服务的保密性、完整性、可用性、抗抵赖、可核查性、真实性和可靠性的要求。

因此本文件的主要目标是提出一套适用于网络访问控制和身份管理的、支撑上层业务的、具有普遍适用性的实体鉴别与安全接入协议和结构。本标准将采用非对称密码技术,并引入在线的可信第三方,构建鉴别协议,并定义网络安全接入架构。

本标准主要内容是:

- 引入可信第三方的实体鉴别及接入架构采用三元结构,将参加鉴别和授权的实体置于对等的角色,利用逻辑的端口控制方法完成双方的鉴别和授权;
- 本标准确定的访问控制方法可应用于无线网络访问控制、有线网络访问控制以及 IP 自适应移动访问控制系统等。

本标准的使用者是通信行业的生产企业、检测机构和科研机构。

本标准的发布机构提请注意,声明符合本标准时,可能涉及到 5.4.5.4 与“一种三元结构的对等访问控制方法”、“一种三元结构的对等访问控制系统”等相关的专利的使用。

本标准的发布机构对于该专利的真实性、有效性和范围无任何立场。

该专利持有人已向本标准的发布机构保证,他愿意同任何申请人在合理且无歧视的条款和条件下,就专利授权许可进行谈判。该专利持有人的声明已在本标准发布机构备案。相关信息可通过以下联系方式获得:

专利权人:西安西电捷通无线网络通信股份有限公司

地址:西安市高新区科技二路 68 号西安软件园秦风阁 A201

联系人:刘长春

邮政编码:710075

电子邮件:ipri@iwncomm.com

电 话:029-87607836

传 真:029-87607829

网 址:<http://www.iwncomm.com>

请注意除了上述专利外,本标准的某些内容仍可能涉及专利。本标准的发布机构不承担识别这些专利的责任。

信息安全技术 引入可信第三方的实体鉴别及接入架构规范

1 范围

本标准规定了引入可信第三方的实体鉴别及接入架构的一般方法。包括：

- a) 引入可信第三方的实体鉴别及接入架构的框架；
- b) 引入可信第三方的实体鉴别及接入架构的基本原理；
- c) 定义引入可信第三方的实体鉴别及接入架构的不同级别以及相应收发数据时端口的行为；
- d) 定义引入可信第三方的实体鉴别及接入架构的参与实体间的消息交互协议；
- e) 定义使用消息交互协议完成引入可信第三方的实体鉴别及接入架构的过程；
- f) 规定协议交互消息中的数据编码；
- g) 建立引入可信第三方的实体鉴别及接入架构管理的需求，识别管理对象，定义管理操作；
- h) 描述远程管理者利用简单网络管理协议(SNMP)所能进行的管理操作；
- i) 描述符合本标准的设备应满足的需求，见附录 A。

本标准适用于无线网络访问控制、有线网络访问控制和 IP 网络访问控制系统等。



2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 15629.2—2008 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 2 部分：逻辑链路控制

GB/T 15629.3—1995 信息处理系统 局域网 第 3 部分：带碰撞检测的载波侦听多址访问(CSMA/CD)的访问方法和物理层规范

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

ISO/IEC 9798-3:1998/Amd. 1:2010 信息技术 安全技术 实体鉴别 第 3 部分：采用数字签名技术的机制 修改单 1 (Information technology—Security techniques—Entity authentication—Part 3: Mechanisms using digital signature techniques—Amendment 1)

IEEE Std 802.3TM-2005 局域网和城域网规范 第 3 部分：带检测冲突的载波检测多址存取(CSMA/CD)方法和物理层规范 [IEEE Standard for Local and Metropolitan Area Networks—Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications]

IEEE Std 802.1DTM-2004 局域网和城域网规范 媒体访问控制桥 [IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) Bridges]

IEEE Std 802.1QTM-2003 局域网和城域网规范 局域网虚拟桥 (IEEE Standards for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks)

IEEE Std 802.1xTM-2004 局域网和城域网规范 基于端口的网络访问控制 (IEEE Standards for Local and Metropolitan Area Networks—Port-Based Network Access Control)

3 术语和定义

下列术语和定义适用于本文件。

3.1

鉴别访问控制器 authentication access controller

位于点到点链路一端的实体,该实体可以鉴别和被鉴别另外一端的实体,它与鉴别服务器有直接的通信。

3.2

请求者 requester

REQ

位于点到点链路一端的实体,该实体可以鉴别和被鉴别另外一端的实体,它必须通过鉴别访问控制器与鉴别服务器通信。

3.3

鉴别服务器 authentication server

AS

提供鉴别服务给请求者和鉴别访问控制器,使请求者和鉴别访问控制器可以相互鉴别。

3.4

桥端口 bridge port

IEEE802.1D 或者 802.1Q 桥的端口。

3.5

边缘端口 edge port

连到一个 LAN 的端口,该 LAN 没有连接其他桥。

3.6

网络接入端口 network access port

系统和链路的连接点,它可以是物理端口,如一个 LAN MAC,或是一个逻辑端口,如 GB 15629.11 中工作站和无线接入点的关联。

3.7

端口访问实体 port access entity

PAE

关联于端口的协议实体,它可以支持自适应、请求者或鉴别访问控制器的功能。

3.8

系统 system

有一个或多个网络接入端口的设备,包括终端工作站、服务器、桥和路由器等。

4 缩略语

AAC	鉴别访问控制器	Authentication Access Controller
AS	鉴别服务器	Authentication Server
ASF	警告标准论坛	Alerting Standards Forum
CBAP	基于证书的鉴别协议	Certificate Based Authentication Protocol
CHAP	挑战握手认证协议	Challenge Handshake Authentication Protocol
CMAP	基于证书的 WMAN 鉴别协议	Certificate-based WMAN Authentication Protocol

DHCP	动态主机配置协议	Dynamic Host Configuration Protocol
FDDI	分布式光纤数据接口	Fiber Distributed Data Interface
IBAP	基于身份签名的鉴别协议	Identity Based Authentication Protocol
IP	互联网协议	Internet Protocol
LAC	链路访问控制	Link Access Control
LAN	IEEE 802 局域网	IEEE 802 Local Area Network
LLC	逻辑链接控制	Logical Link Control
MAC	媒体访问控制	Media Access Control
MIC	消息完整性校验	Message Integrity Check
MPDU	MAC 协议数据单元	MAC Protocol Data Unit
OID	客体标识符	Object Identifier
OUI	组织全球唯一性标识	Organizationally Unique Identifier
PCAP	对等控制访问协议	Peer Control Access Protocol
PDU	协议数据单元	Protocol Data Unit
Port	网络访问端口	network access port
PPP	点到点协议	Point-to-Point Protocol
REQ	请求者	Requester
SNAP	子网络访问协议	Subnetwork Access Protocol
SNMP	简单网络管理协议	Simple Network Management Protocol
TAEP	三元鉴别可扩展协议	Tri-element Authentication Extensible Protocol
TAEPoL	基于链路的三元鉴别可扩展协议	TAEP over Link
TePA	三元对等鉴别	Tri-element Peer Authentication
TePA-AC	基于三元对等鉴别的访问控制技术	TePA-based Access Control
TP	可信第三方	the Trusted third Party
VLAN	虚拟局域网	Virtual LAN

5 引入可信第三方的实体鉴别及接入架构

5.1 概述

引入可信第三方的实体鉴别及接入系统采用三元结构,将参加鉴别和授权的实体置于对等的角色,利用逻辑的端口控制方法完成双方的鉴别和授权。本标准确定的访问控制方法可应用于无线网络访问控制、有线网络访问控制以及 IP 自适应移动访问控制系统等。

本章描述了引入可信第三方的实体鉴别及接入系统的结构框架、控制功能以及采用该机制的设备所进行的各项操作之间的关系。

引入可信第三方的实体鉴别及接入架构对系统功能进行了扩展,它提供了一种访问控制方法,可以用来阻止请求者对鉴别访问控制器系统的资源进行未授权的访问,同时阻止请求者误访问未授权的鉴别访问控制器系统;还可以让请求者用来阻止来自未授权鉴别访问控制器系统的连接。例如,对于 MAC 桥系统,引入可信第三方的实体鉴别及接入架构可以实现限制用户只能访问公共桥端口,或者在一个组织内,限制组织内资源只能被组织内用户访问等应用场景。访问控制是通过对连接在受控端口上的系统进行鉴别来实现的,所谓端口是指参加鉴别和授权的实体附着于它们之间的逻辑点到点连接的接口,例如 GB 15629.11—2003 无线局域网的关联、GB/T 15629.3—1995 LAN 交换机的端口。根据鉴别的结果,请求者系统或鉴别访问控制器系统决定是否给予对方授权,允许对方通过受控端口访问自己的资源。如果对方没有获得授权,根据受控端口的 OperControlledDirections 参数限制在请求者系统和鉴别访问控制器系统间未授权的数据流动。引入可信第三方的实体鉴别及接入架构采用基于三元

对等鉴别的访问控制技术(TePA-AC),可以被一个系统用来鉴别其他任何连接在该系统受控端口上的系统,系统可以是路由器、终端设备、交换机、无线接入点、无线基站、网关、应用程序等。附录 C 和附录 D 描述了三元对等鉴别的访问控制在城域网和局域网设备的参考实现。

本标准中,请求者系统和鉴别访问控制器系统之间的鉴别采用基于密码技术的鉴别协议实现。鉴别协议运行要求双方具有“密钥”的信任基础,即双方共享一个秘密——密钥,作为双方的信任凭证。如果只有请求者系统和鉴别访问控制器系统这两种实体,密钥管理将是对多个请求者系统和多个鉴别访问控制器系统之间的管理,也就是管理多对多的信任关系。多对多的信任关系导致系统实现异常复杂,为了降低系统实现的复杂性,本标准定义了第三种实体——鉴别服务器。鉴别服务器和请求者系统有“密钥”的信任基础,鉴别服务器和鉴别访问控制器系统也有“密钥”的信任基础,而请求者系统和鉴别访问控制器系统之间没有“密钥”的信任基础。这样多对多的信任关系将演变为两个多对一的信任关系,有效地降低了系统实现的复杂性。

本标准中,请求者系统和鉴别访问控制器系统之间的鉴别可以通过鉴别服务器作为中介来实现,鉴别协议在请求者系统、鉴别访问控制器系统和鉴别服务器三个实体上运行,称为三元对等鉴别。

5.2 访问控制的范围

引入可信第三方的实体鉴别及接入架构的操作假设所操作的端口在请求者与鉴别访问控制器之间提供点到点的连接。

本标准提供了一个用于在请求者与鉴别访问控制器之间、鉴别访问控制器和鉴别服务器之间传递消息的协议,并根据协议执行的结果来决定请求者与鉴别访问控制器的端口状态。

5.3 系统、角色和端口

5.3.1 概述

系统的端口提供了一种手段,通过该方式可以访问其他系统提供的服务,也可以通过该方式向其他系统提供服务。引入可信第三方的实体鉴别及接入架构可以控制系统的端口状态,保证只有被授权的系统才能访问该系统提供的服务,或者访问被授权系统提供的服务。

为了描述引入可信第三方的实体鉴别及接入架构的操作,一个系统的端口(更准确地说,是端口的端口控制实体 PAE)可以采用以下两种角色:

- 鉴别访问控制器(AAC):如果系统需要通过端口提供资源给其他系统访问,那么它采用鉴别访问控制器的角色;鉴别访问控制器也可以通过该端口访问其他系统的资源,鉴别访问控制器可以直接与鉴别服务器通信。
- 请求者(REQ):如果系统需要通过端口访问其他系统提供的资源,那么它采用请求者的角色;请求者要通过鉴别访问控制器与鉴别服务器通信。

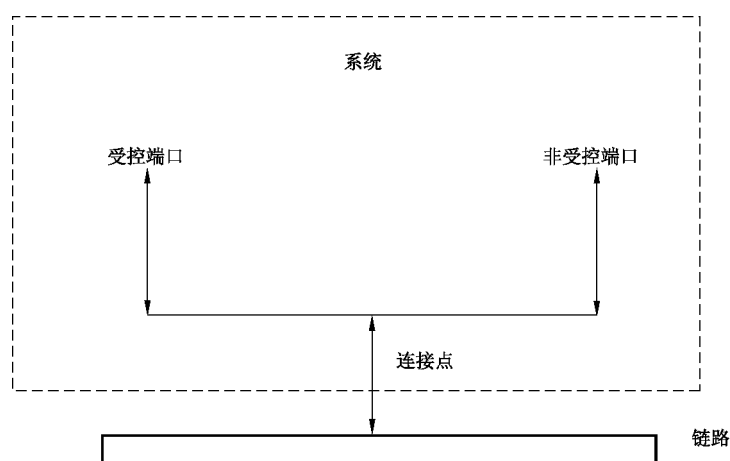
另外还定义了一个系统角色:

- 鉴别服务器(AS):鉴别访问控制器和请求者进行鉴别时,需要通过鉴别服务器完成鉴别协议的交互过程。鉴别服务器作为鉴别访问控制器和请求者共同信任的可信第三方提供初始信任。

以上描述的三种角色在通常情况下的系统中都需要,以完成鉴别协议交换。但在某些特殊情况下,鉴别服务器并不是必须存在的。例如,当两个系统采用共享密钥鉴别,且不需要其他的管理时,鉴别服务器就没有必要存在。一个系统可以采用其中一个角色或多个角色,例如,AAC 可和 AS 在一个系统中实现;PAE 可以在一个协议交换中采用 AAC 角色,而在另一个协议交换中采用 REQ 角色。

5.3.2 受控端口与非受控端口

引入可信第三方的实体鉴别及接入架构在系统和链路的连接点上建立两个不同的访问点,一个是非受控端口,不论授权状态如何,允许链路上的系统之间不受控地交换数据包;一个是受控端口,只有处于授权状态,才允许交换数据包。非受控端口和受控端口是同一连接点的两个部分,从物理连接点得到的数据帧会被它们同时得到。见图 1。



说明: 尽管数据包在受控端口和非受控端口均可得到,但协议实体在某一时刻只能关联到其中一个端口。

图 1 受控端口与非受控端口

系统与链路的连接点可以是物理端口,也可以是逻辑端口,该端口提供到其他系统的一对一连接。例如,在交换的 LAN 中,该连接点可以通过 MAC 地址实现,MAC 地址使鉴别访问控制器和请求者之间一对多的关系成为可能;在 IP 网络中,该连接点可以通过 IP 地址实现;对于应用程序,该连接点可以通过应用程序和会话层/传输层的接口实现;在 GB 15629.11—2003 无线局域网中,无线接入点和工作站之间存在独立的关联实现一对一的连接;在 PPP 连接中,终端和网关之间是点对点的连接。在本标准中,两个系统之间建立单独的关联是引入可信第三方的实体鉴别及接入架构方法实施的前提条件。

受控端口和非受控端口是逻辑概念,它将数据流进行了分类。管理和控制流可以通过非受控端口,信息流则通过受控端口,在不影响系统通信管理的前提下,通过受控端口对信息流进行控制。

图 2 描述了受控端口 AuthControlledPortStatus 和 LAC 控制的结果。任何对链路的访问都要受到端口的链路访问控制能力 LAC(LAC 在 802 网络是 MAC,在 IP 网络中是 IP)当前的管理性和可操作性状态以及 AuthControlledPortStatus 参数的控制。如果 LAC 在物理上或管理上不可操作时,那么无论在受控端口还是非受控端口都不会发生基于 LAC 的任何协议交换。在图 2 的系统 1 中,因为到链路连接点的 LAC 是可操作的,受控端口和非受控端口均可访问链路;而在图 2 的系统 2 中,因为到链路连接点的 LAC 是不可操作的,受控端口和非受控端口都不能访问链路,LAC 的不可操作性也导致系统的受控端口状态为非授权的。

AuthControlledPortStatus 通过一个开关来表示,开或者关表示允许或阻止数据包通过该端口。图 2 中有两个系统,每个系统均有一个受控端口,端口的 OperControlledDirections 参数为 Both。图 2 的系统 2 中受控端口的 AuthControlledPortStatus 为 Unauthorized,因此不允许数据包通过,故开关的状态是断开的;图 2 的系统 1 中受控端口 AuthControlledPortStatus 为 Authorized,因此允许数据包通过的,故开关的状态是闭合的。

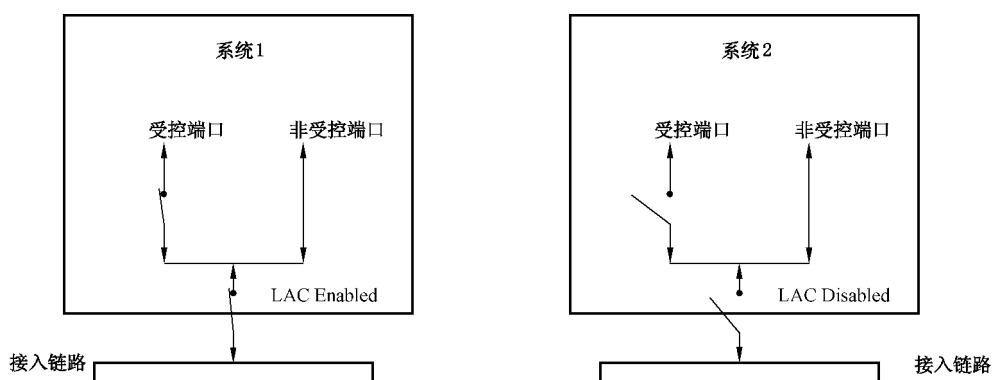


图 2 LAC 对端口的影响

除了 AuthControlledPortStatus 以外,受控端口的 AuthControlledPortControl 参数使管理员可以控制端口的授权状态。该参数有 3 个值:强制授权的 ForceAuthorized、自动的 Auto、强制非授权的 Force Unauthorized,其默认值为 Auto。AuthControlledPortStatus 和 AuthControlledPortControl 之间的关系如下:

- 当 AuthControlledPortControl 的值为 ForceUnauthorized 时,使得鉴别访问控制器或请求者状态机将受控端口 AuthControlledPortStatus 的值设置为 Unauthorized,即受控端口无条件地设置为未授权;
- 当 AuthControlledPortControl 的值为 ForceAuthorized 时,使得鉴别访问控制器或请求者状态机将 AuthControlledPortStatus 的值设置为 Authorized,即受控端口无条件地鉴别访问控制器或请求者为授权;
- 当 AuthControlledPortControl 的值为 Auto 时,鉴别访问控制器和请求者状态机根据请求者、鉴别访问控制器和鉴别服务器之间的鉴别交换的结果确定 AuthControlledPortStatus 的值。

在上述 3 种情况中,AuthControlledPortStatus 的值直接反映鉴别访问控制器和请求者状态机中 portStatus 变量的值。以下 3 个因素影响 portStatus 变量的值:

- 鉴别访问控制器状态机的授权状态(如果端口上没有实施状态机,则认为是“授权的”)。
- 请求者状态机的授权状态(如果端口没有实施状态机,则认为是“授权的”)。
- REQ Access Control With AAC Administrative Control 参数。该参数有两个可能值:Active 和 Inactive,默认是 Inactive,支持 Active 是可选的。只有当 AAC 状态机和 REQ 状态机在一个端口都实现时,该参数才有效。如果该参数值是 Inactive,则 portStatus 参数值仅被 AAC 状态机的授权状态确定。如果该参数的值是 Active,则 portStatus 参数值被 AAC 状态机和 REQ 状态机共同决定,AAC 和 REQ 状态机中任何一个状态是 Unauthorized 时,则 portStatus 参数值为 Unauthorized。

一个系统中每个端口的 AuthControlledPortControl 参数值都受到系统 SystemAuthControl 参数值的控制。SystemAuthControl 参数值可以为 Enabled 和 Disabled,默认值是 Disabled。如果 SystemAuthControl 被置为 Enabled,那么鉴别对于系统即为可用,则每个端口的授权状态均由 AuthControlledPortControl 参数来控制。如果 SystemAuthControl 被置为 Disabled,则表示所有端口的 AuthControlledPortControl 参数均被置为 ForceAuthorized。实际上,将 SystemAuthControl 参数设置为 Disabled 会导致所有端口的鉴别被禁止,使得所有受控端口被授权。

通常希望其他系统的协议交换使用本系统的一个或多个受控端口,然而,一个给定的协议可能需要绕过授权功能而使用非受控端口。图 3 显示了鉴别访问控制器系统和请求者系统中的受控和非受控端口,以及 PAEs 根据授权交换的结果来控制受控端口的授权状态,其中,为了完成协议交换,PAEs 需要利用非受控端口。

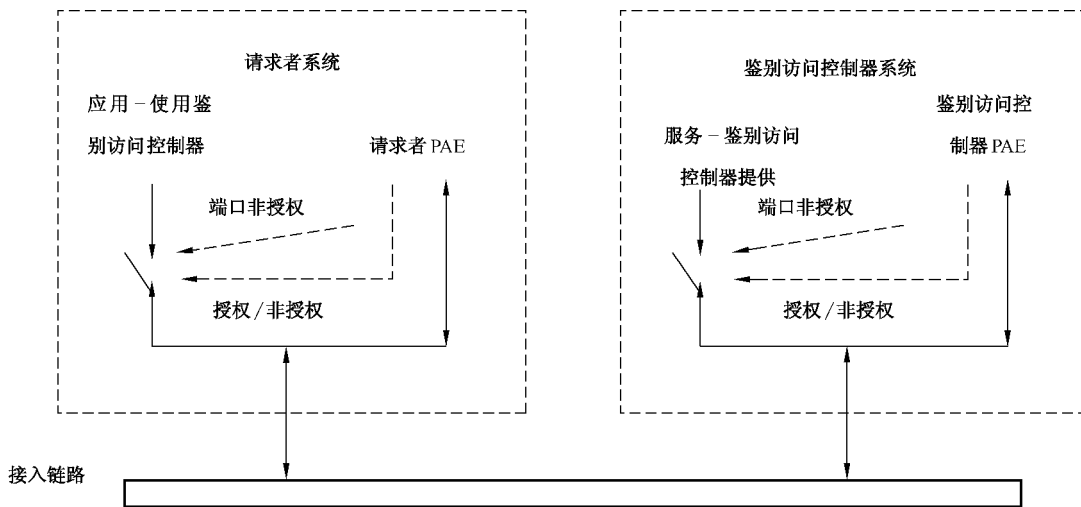


图 3 鉴别访问控制器系统和请求者系统中的受控和非受控端口

图 4 显示了请求者、鉴别访问控制器和鉴别服务器，以及它们之间信息交换的关系。

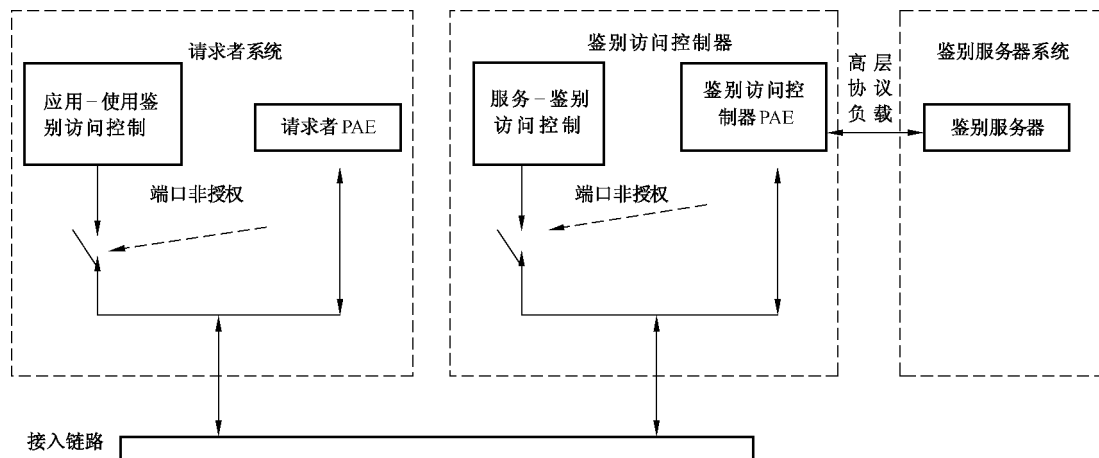


图 4 请求者、鉴别访问控制器和鉴别服务器系统

图 4 中，鉴别访问控制器和请求者的受控端口为未授权状态，因此请求者系统无法访问鉴别访问控制器系统提供的服务。两个 PAE 利用各自的非受控端口，使用鉴别协议互相通信，鉴别访问控制器 PAE 通过高层协议负载的鉴别协议与鉴别服务器进行通信。

鉴别访问控制器和鉴别服务器之间的通信可以利用 LAN，或使用其他通信信道。如果鉴别服务器和鉴别访问控制器没有设置在一台设备上，则其之间应进行鉴别协议交换。

5.3.3 接收和发送控制

每个受控端口的受控方向参数 AdminControlledDirections 和 OperControlledDirections 决定未授权的控制端口是对双向的通信进行控制(向内进入的帧和向外发送的帧)，还是仅对来向通信(仅只对向内进入的帧)进行控制。受控方向参数取值为 Both 或 In。这两个参数的含义如下：

- AdminControlDirection=Both:表示对受控端口的向内进入和向外发送的双向传输均进行控制。如果 AdminControlledDirections=Both,则 OperControlledDirections 参数的值无条件等于 Both。
- AdminControlledDirection=In:表示仅对受控端口向内进入的传输进行控制。如果 Admin-

ControlledDirections=In,则 OperControlledDirections 参数的值在初始化且端口的 LAC 服务可用时为 In。然而,如果出现下列任何一种情况,则 OperControlledDirections 参数值将为 Both:

- 端口是一个桥端口,且桥检测状态机(见 IEEE Std 802.1D-2004 中的第 17 章)检测到另外一个桥连接到该端口;
- 端口是一个桥端口,且该端口的 Edge Port 参数值为 FALSE;
- 端口的 LAC 不可用。

AdminControlledDirection 参数的值只能通过管理操作进行修改。引入可信第三方的实体鉴别及接入架构的实施应支持能够独立地将每个受控端口的 AdminControlledDirection 参数设置为 Both 以及独立地将每个受控端口的 AdminControlledDirection 参数设置为 In。

注:当一个与受控端口相连的设备需要从受控端口获得协议消息,即从受控端口传输出去协议消息,以支持某种形式的启动或初始化时(例如,Wake-on-LAN),In 的设置允许基于端口的访问控制将安全条件放宽,但是它仍然希望在鉴别完成之前阻止与之相连的设备给受控端口发送数据(参见 E.1)。很明显,受控端口采用这种形式放宽的安全限制,减弱了基于端口的访问控制处理某些攻击的有效性。

5.4 端口访问实体(PAE)

5.4.1 概述

端口访问实体(PAE)对第 7 章定义的协议进行操作。对于支持端口访问控制功能的系统,每个端口都存在 PAE,无论该系统扮演请求者角色还是鉴别访问控制器角色。

在鉴别交换中扮演请求者角色的 PAE 被称为请求者 PAE。

在鉴别交换中扮演鉴别访问控制器角色的 PAE 被称为鉴别访问控制器 PAE。

这两种 PAE 角色均根据鉴别过程的结果控制受控端口的授权/未授权状态。

5.4.2 鉴别访问控制器角色

鉴别访问控制器 PAE 负责对连接到其受控端口的请求者 PAE 进行鉴别,并且对受控端口的授权状态进行相应的控制。

在鉴别过程中,鉴别访问控制器 PAE 可能会使用鉴别服务器。鉴别服务器可能与鉴别访问控制器处于同一系统,或者处于能通过远程通讯机制、基于 LAN 或其他机制进行访问的其他系统中。请求者 PAE 和鉴别访问控制器 PAE 的通信,以及鉴别访问控制器 PAE 和鉴别服务器(当鉴别服务器和鉴别访问控制器不在同一个系统中时)之间的通信将通过第 7 章描述的协议和程序来完成。

5.4.3 请求者角色

请求者 PAE 负责将请求者的鉴别凭证发送给鉴别访问控制器 PAE,作为对鉴别访问控制器 PAE 请求的响应,还负责根据与鉴别访问控制器 PAE 进行鉴别交换的结果来控制受控端口的授权状态。请求者 PAE 也可能发起鉴别交换,完成特定的注销交换。

5.4.4 端口访问限制

鉴别一般发生在系统初始化时或者请求者系统连接到鉴别访问控制器系统的端口时。在鉴别成功完成之前,请求者系统只能访问鉴别访问控制器系统来完成鉴别交换,或者访问鉴别访问控制器系统提供的、没有访问控制限制的、处于鉴别访问控制器非受控端口上或者请求者非受控端口上的服务(见 5.3)。一旦鉴别成功完成,两个系统均允许请求者系统访问通过鉴别访问控制器系统受控端口提供的服务。

支持受控端口的 LAC 的操作状态可以被启用或禁用。如果 LAC 的操作状态为禁用,那么无论受控端口的授权状态是什么,LAC 都不能使用。

注 1: IEEE Std 802.1D-2004 中的第 6 章描述了表示端口的 MAC 启用/禁用状态的桥 (Bridge) 端口。GB/T 15629.3—1995 描述了类似的参数,它定义了通过聚合提供的逻辑端口的启用/禁用状态。

在使用 PAE 的系统中,受控端口在鉴别之前一直处于未授权状态,因此是禁用的。一旦鉴别成功,且 PAE 决定授权所鉴别的用户来访问受控端口,则受控端口将进入授权状态。如果受控端口没有被其他原因禁用(例如,由于管理方面的原因而禁用 LAC),受控端口就可以使用(见 5.3)。

注 2: 当鉴别访问控制器系统的受控端口处于未授权状态时,动态主机配置协议(DHCP)和其他初始化传输可能不会通过受控端口发送和接收,这依赖于端口控制的层次和端口的 ControlledDirection 参数。因此,鉴别需要在终端工作站的初始化序列时就进行(例如,在 DHCP 和 IP 初始化之前)。当请求者的受控端口处于未授权状态时,不能发送和接收 DHCP 和其他初始化传输。请求者在授权之前试图利用受控端口,将会导致 DHCP 或初始化包的丢失,还可能会导致无法获取 IP 地址。

除了能控制受控端口的授权状态之外,PAE 的操作还能支持受控端口授权状态的老化,可在任何时间激活请求者或鉴别访问控制器进行重新鉴别。在重新鉴别期间,受控端口保持授权状态,仅在重新鉴别失败时,才转变到未授权状态。

在某些配置中不需要对特定端口进行鉴别(例如,在 Inter-Bridge 链路中,端口连接到服务器上),因此鉴别是针对每个端口进行配置的。这种配置的管理操作将在第 8 章中描述。

5.4.5 TAEP

5.4.5.1 概述

PAE 包含鉴别控制和互相通信两个功能,其中鉴别控制功能实现 PAE 的状态转换,见第 7 章;通信功能传递两个 PAE 之间的鉴别消息。以下定义的 TAEP(Triple-element Authentication Extensible Protocol)协议支持鉴别访问控制器 PAE 和请求者 PAE 之间、鉴别访问控制器 PAE 和鉴别服务器之间的通信功能。

5.4.5.2 TAEP 分组格式

图 5 给出了 TAEP 分组格式。

Code(8位比特)	Identifier(8位比特)	Length(16位比特)
Data		

图 5 TAEP 分组格式

TAEP 分组各个域的定义见表 1。

表 1 TAEP 分组域定义

名称	长度(比特)	描述
Code 编码	8	表示 TAEP 分组的类型 1 Request 2 Response 3 Success 4 Failure
Identifier 标识	8	用于匹配 Request 和 Response 分组
Length 长度	16	表示整个 TAEP 分组的八位位组数,即指包括 Code、Identifier、Length 和 Data 所有字段的长度总和
Data 数据	可变	分组含 0 个或多个八位位组,其格式由 Code 字段的值决定

5.4.5.3 TAEP 分组字段各个域定义

5.4.5.3.1 Request 和 Response

格式见图 6。

Code(8位比特)	Identifier(8位比特)	Length(16位比特)
Application Type (8位比特)	Reserved(24位比特)	
Type(8位比特)	Type-Data	

图 6 Request 和 Response 分组格式

图 6 给出了 Request 和 Response 分组格式。

其中 Application Type 字段表示 TAEP 协议运行的应用环境：

0:表示未特定定义的应用环境。

1~223:表示公有的应用环境,数值含义等待定义。

224~239:表示私有的应用环境,用于 TAEP 在私有应用环境中使用,数值含义由使用者定义。

240~255:保留。

Application Type 字段默认值为 0。

其中 Type 字段长度为 1 个八位位组,它表示 Request 和 Response 分组的类型,定义见表 2。

表 2 Request 和 Response 分组类型定义

Type 值	定义	描述
1	Identity	Identity 类型用于鉴别访问控制器询问请求者的身份。对于 Identity 类型的 Request 分组,包含可显示的消息,采用 UTF-8 编码的 ISO 10646 字符,用于提示请求者;请求者应发送 Identity 类型的 Response 分组来响应 Identity 类型的 Request 分组
2	Notification	Notification 类型用于将一个可显示的消息从鉴别访问控制器传递到请求者。在鉴别过程完成之前,鉴别访问控制器可在任何时候发送一个 Notification 类型的 Request 分组到请求者,除非正在进行的鉴别协议禁止使用 Notification 类型的分组;请求者收到 Notification 类型的 Request 分组后应使用 Notification 类型的 Response 来响应鉴别访问控制器
3	Nak (Response only)	Nak 在 Response 分组中存在,用于表示不支持 Request 分组中提议的鉴别方法。Nak 类型的 Type-Data 字段的内容包含可变长度的 8 位位组,每个 8 位位组表示 1 个通用的鉴别方法。数值 4~249 表示可以替代使用鉴别方法,如果没有可替代的鉴别方法,Type-Data 字段内容为数值 0
248	TAEP-IBAP	TAEP-IBAP 是一个基于身份签名机制的鉴别协议,Type-Data 定义参见附录 B 的 B.1 中描述 TAEP-IBAP 类型的分组中 Type-Data 字段的内容
249	TAEP-CBAP	TAEP-CBAP 是一个基于三实体公钥的鉴别协议,Type-Data 定义参见附录 B.2 中描述 TAEP-CBAP 类型的分组中 Type-Data 字段的内容

表 2 (续)

Type 值	定义	描 述
250	TP Authentication	TP Authentication 类型用于鉴别访问控制器向鉴别服务器询问可用的鉴别方法
254	Expanded Types	Expanded Types 用来支持厂商自定义的鉴别协议以及扩展通用鉴别协议类型的范围到预定数值 255 以外
255	Experimental use	Experimental Use 类型分组的 Type-Data 字段没有固定的格式,它用于测试新的 TAEP 鉴别协议类型

TP Authentication 类型用于鉴别访问控制器向鉴别服务器询问可用的鉴别方法。如果鉴别访问控制器需要从鉴别服务器获得鉴别方法的类型,则鉴别访问控制器向鉴别服务器发送 TP Authentication 类型的 Request 分组来请求可用的鉴别方法,该分组包含请求者和鉴别访问控制器的身份;鉴别服务器将发送 TP Authenticaiton 类型的 Response 分组进行响应,该分组包含鉴别服务器支持的用于该请求者和鉴别访问控制器的鉴别方法类型。TP Authenticaion 类型的分组 Type-Data 字段格式见图 7。

Type (8位比特)	Subtype (24 位比特)
SubData	

图 7 TP Authentication 和 NAK 类型分组中 Type-Data 字段格式

TP Authentication 类型分组中 Type-Data 字段各个域的定义见表 3。

表 3 TP Authentication 类型分组中 Type-Data 字段域定义

名 称	长度(比特)	描 述
Type 类型	8	Type=250
Subtype 子类型	24	00-00-00 Identity FF-FF-FF 通用鉴别方法 其他 Vendor ID,由 IANA 分配的厂商的 SMI 网络管理专用企业代码
Subdata 子数据	可变	当 SubType 取值为“00-00-00”时,SubData 字段包含长度和身份内容,前 2 个八位位组为长度字段,表示后面身份字段的八位位组个数; 当 SubType 取值为“FF-FF-FF”时,表示通用的鉴别方法,SubData 字段为 4 个八位位组,表示通用的鉴别协议类型; 当 SubType 取值为其他值,即厂商的 Vendor ID 值,SubData 字段为 4 个八位位组,表示厂商自定义的鉴别方法

当鉴别服务器传递所支持的多种鉴别方法时,TP Authenticaion 类型的分组 Type-Data 字段格式见图 8。

Type (8位比特)	Subtype (24位比特)
SubData	
Type (8位比特)	Subtype (24位比特)
SubData	
.....	

图 8 TP Authentication 类型的分组中 Type-Data 字段格式

Expanded Types 用来支持厂商自定义的鉴别协议以及扩展通用鉴别协议类型的范围到预定数值 255 以外。

Expanded Types 类型的分组 Type 和 Type-Data 字段格式见图 9。

Type (8位比特)	Vendor-ID (24位比特)
Vendor-Type	

图 9 Expanded Types 类型的分组中 Type-Data 字段格式

Expanded Type 类型分组中 Type-Data 字段各个域的定义见表 4。

表 4 Expanded Type 类型分组中 Type 和 Type-Data 字段域定义

名称	长度(比特)	描述
Type 类型	8	Type=254
Vendor-ID 厂商标识	24	00-00-00 通用 其他 厂商的 Vendor ID, 由 IANA 分配的厂商的 SMI 网络管理专用企业代码
Vendor-Type 厂商类型	32	当 Vendor-ID 为 0 时, Vendor-Type 字段用于扩展通用的鉴别协议类型。Vendor-Type 字段为 4 个八位位组, 数值 0~255 保留, 用于兼容在非扩展时用一个八位位组表示的鉴别协议类型, 数值 3 是个例外。 当 Vendor-ID 为 0 且 Vendor-Type 为 3 时, 其意义为 Expanded-NAK。 当 Vendor-ID 非 0 时, Vendor-Type 字段为 4 个八位位组, 表示厂商自定义的鉴别协议类型。

Expanded Types 类型的分组使用示例见图 10 和图 11。

Type=254	Vendor-ID=0
Vendor-Type=3(NAK)	
Type=254	Vendor-ID=20(MIT)
Vendor-Type=6	
Type=254	Vendor-ID=0
Vendor-Type=249(TAEP-CBAP)	

图 10 支持多种可选鉴别协议的 Expanded Types 类型的分组中 Type-Data 字段格式

Type=254	Vendor-ID=0
Vendor-Type=3(NAK)	
Type=254	Vendor-ID=0
Vendor-Type=0 (没有可选鉴别协议)	

图 11 没有可选鉴别协议的支持多种可选鉴别协议的 Expanded Types 类型的分组中 Type-Data 字段格式

5.4.5.3.2 Failure 和 Success

图 12 给出了 Failure 和 Success 分组的格式,分组中不包含 Data 字段。

Code(8位比特)	Identifier(8位比特)	Length(16位比特)
------------	------------------	---------------

图 12 Failure 和 Success 分组格式

5.4.5.4 TAEP 复用模型

5.4.5.4.1 一般要求

TAEP 复用模型见图 13。

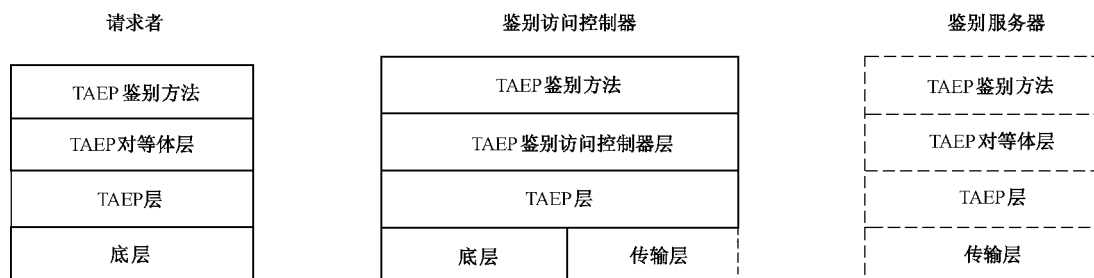


图 13 TAEP 复用模型

图 13 的虚线部分表示可选,该部分在运行中是否起作用由 TAEP 鉴别方法决定。

5.4.5.4.2 底层和传输层

底层和传输层负责在请求者、鉴别访问控制器和鉴别服务器之间传送和接收 TAEP 分组,该传输层是逻辑概念,表示该层和相邻的底层可以不是同一种技术。TAEP 可以运行在多种底层技术上,包括 PPP、IEEE 802.3、IEEE 802.11、IEEE 802.16、IEEE 802.15、IP、TCP 等。

5.4.5.4.3 TAEP 层

TAEP 层通过底层传送和接收 TAEP 分组,实现重复分组检测和重传,在对等体层和鉴别访问控制器层之间传送消息。

5.4.5.4.4 TAEP 对等体层和鉴别访问控制器层

根据 Code 字段的值,TAEP 层解析收到的 TAEP 分组,传送到 TAEP 对等体层或 TAEP 鉴别访问控制器层。

5.4.5.4.5 TAEP 鉴别方法

通过 TAEP 对等体和 TAEP 鉴别访问控制器层传送消息,TAEP 鉴别方法层实现了鉴别算法。由于在 TAEP 层没有实现分片,因此 TAEP 鉴别方法层必须自己实现分片功能。

5.4.5.5 端对端的协议

TAEP 是一个端对端的协议,是指在对等体和鉴别访问控制器之间 TAEP 协议的运行。因此,一条链路的两端都可以同时作为鉴别访问控制器和对等体。在这种情况下,两端都必须实现 TAEP 鉴别访问控制器层和 TAEP 对等体层。另外,TAEP 鉴别方法的实现必须指出鉴别访问控制器和对等体的功能。

存在鉴别服务器的情况下,即 TAEP 鉴别方法需要鉴别服务器的存在,实现鉴别访问控制器功能的一端必须能够和鉴别服务器无限制通信。

5.4.5.6 TAEP 消息交换

TAEP 消息交换的步骤如下:

- a) 鉴别访问控制器发送 Request 分组给请求者要求开始鉴别,Request 有一个类型字段指示请求的种类,类型是 Identity。
- b) 请求者发送 Response 分组给鉴别访问控制器来响应有效的 Request,Response 分组中包含一个类型字段,对应于 Request 分组中的类型字段,消息中包含有对等体的身份。
- c) 鉴别访问控制器发送 Request 分组给鉴别服务器,Request 有一个类型字段指示请求的种类,类型是 TP Authentication。
- d) 鉴别服务器发送 Response 分组给鉴别访问控制器,Response 分组中包含一个类型字段,对应于 Request 分组中的类型字段。
- e) 鉴别访问控制器根据鉴别服务器返回的鉴别方法类型,选择一种鉴别方法开始鉴别过程。发送 Request 分组给请求者,请求者响应 Response 分组给鉴别访问控制器,Request 和 Response 的序列根据需要持续交互。鉴别访问控制器向鉴别服务器发送 Request 分组,而鉴别服务器向鉴别访问控制器响应 Response 分组,此 Request 和 Response 的序列会持续需要的长度。鉴别访问控制器负责重传 Request 分组。
- f) 对话一直持续到鉴别访问控制器不能鉴别请求者,鉴别访问控制器将发送 Failure 分组给请求者;或者鉴别访问控制器判断成功的鉴别已经完成,鉴别访问控制器或停止发送 Request 分组,结束消息交互,或发送 Success 分组给请求者。

以上步骤描述了典型 TAEP 消息交换,但根据鉴别方法的不同,步骤过程会有不同。

5.4.5.7 底层、传输层协议

在鉴别访问控制器和鉴别服务器之间传递 TAEP 消息使用 TAEP-AS-SVC 服务协议,鉴别服务器作为服务端在 UDP/TCP 端口 5111 上接收 TAEP 消息,鉴别访问控制器作为客户端发送 TAEP 消息。

请求者和鉴别访问控制器通过第 6 章定义的 TAEPoL 进行 TAEP 消息传递。

5.4.6 注销机制

有很多操作能导致受控端口变成未授权状态,使受控端口根据自己的 OperControlledDireciton 参数来控制通过端口的访问:

- a) 请求者和鉴别访问控制器之间的鉴别交换会使得对端口的授权失败;
- b) 管理控制能阻止端口变成授权状态;

- c) 端口的 LAC 可以处于任何原因而不可操作(包括硬件失败或管理原因);
- d) 请求者和鉴别访问控制器之间的连接失败会导致鉴别访问控制器的授权状态超时;
- e) 重新鉴别计时器超时,但重新鉴别没有成功;
- f) 请求者 PAE 可能不会应答鉴别访问控制器 PAE 的重新鉴别信息请求;
- g) 请求者 PAE 会发布一个明确的注销请求。

当一个用户从一个终端工作站注销,在某些情况下,可能会发生用户(或不同的用户)不通过新的登陆请求就能访问终端工作站和网络的情况。提供明确的注销机制能保证会话结束,这不仅关于用户对终端工作站的访问,也涉及与终端工作站相连的鉴别访问控制器系统受控端口的授权状态。因此一个明确的注销会使得请求者和鉴别访问控制器 PAE 将受控端口设置为未授权状态。

5.4.7 自适应端口选择

一个 PAE 可静态地采用请求者或鉴别访问控制器的角色,也可根据情况动态选择请求者和鉴别访问控制器角色。若对方为鉴别访问控制器,自适应 PAE 将采用请求者角色;若对方为请求者,自适应 PAE 将采用鉴别访问控制器角色。如果双方均为自适应 PAE,那么根据优先级和物理地址来确定角色。优先级高的 PAE 将成为鉴别访问控制器,另外一个将成为请求者。如果优先级一样,那么物理地址高的 PAE 将成为鉴别访问控制器。如果 PAE 没有物理地址,那么 PAE 就无法使用自适应端口选择的功能,只能静态设定。PAE 通过第 6 章定义的 TAEPoL 帧进行自适应选择。

5.5 IEEE Std 802.3-2005 中端口访问控制的使用

当使用了引入可信第三方的实体鉴别及接入架构,同时也使用了 IEEE Std 802.3-2005 中的第 43 章定义的端口聚合时,PAE 的鉴别和授权机制应在各自端口上进行,当端口处于未授权状态时,不应考虑将端口聚合。任何已成为聚合端口族中的成员端口状态变为未授权时,这些成员端口均应从聚合端口族中去掉。

为了保证此环境中能产生安全聚合,完成单次聚合的端口应被授权,可以给实体本身授权,或者给那些授权后能参加聚合的实体授权。决定是否授权一个端口参加聚合,该过程不属于本标准范畴。

注:链路聚合会将该聚合密钥和其他聚合密钥联合,以便产生 LACP 使用的最终聚合密钥。

6 链路上的 TAEP 封装(TAEPoL)

6.1 概述

本章定义了请求者 PAE 和鉴别访问控制器 PAE 之间负载 TAEP 分组的封装技术。该封装称为链路上的 TAEP,或 TAEPoL。

在不同形式的链路上,封装的方法会有所不同,但其功能是相似的,即要能够传递 TAEP 数据分组和管理请求者 PAE 和鉴别访问控制器 PAE 之间的鉴别会话。

本标准定义了 GB/T 15629.2(IEEE 802)系列网络中的 TAEPoL,对于其他网络形式时的 TAEPoL 协议本标准没有定义。本章中的 TAEPoL 表示在 GB/T 15629.2(IEEE 802)系列网络中的 TAEP 链路封装。

6.2 八位位组的发送和标识

所有的 TAEPoL PDU 均包含整数个八位位组,这些八位位组编号从 1 开始,按照它们在 MAC 中的顺序依次递增。每个八位位组中的位编号从 1 到 8,1 为最低位。

当用连续的八位位组来表示二进制数时,低编号的八位位组为二进制数字中的高位八位位组。

当 TAEPoL PDU 的编码(元素)使用第 6 章中的图表示时,表示规则如下:

- a) 八位位组 1 显示在页的最顶部,高编号的八位位组显示在底部;
- b) 在一行中如果出现多个八位位组,那么低编号八位位组在最左边,高编号八位位组在右边;
- c) 在一个八位位组内,位 8 在左边,位 1 在右边。

6.3 TAEPoL MPDU 在 GB/T 15629.2(IEEE 802.2)逻辑链路控制(LLC)中的格式

TAEPoL MPDU 在 GB/T 15629.2 LLC 格式的定义见图 14。从 GB/T 15629.2 MPDU 的 Length/Type 字段开始。图 14 中除 SNAP-编码以太网类型字段以外的其他字段均在 6.6 中定义。

字段	八位位组
SNAP—编码以太网类型	1~8
协议版本(6.6.2)	9
类型(6.6.3)	10
长度(6.6.4)	11~12
内容(6.6.5)	13~N

图 14 TAEPoL MPDU 在 GB/T 15629.2 逻辑链路控制(LLC)中的格式

SNAP-编码以太网类型字段:长度为(1~8)个八位位组,内容如下:

- a) 八位位组 1~3 载有标准 SNAP 报头,包含 16 进制值 AA-AA-03;
- b) 八位位组 4~6 载有 SNAP 协议标示符(PDI),包含 16 进制值 00-00-00;
- c) 八位位组 7~8 载有以太网类型值,定义见表 5。

表 5 标准以太网类型分配

分 配	值
PAE Ethernet Type	0×891b

6.4 TAEPoL MPDU 在 GB/T 15629.3(IEEE 802.3)中的格式

TAEPoL MPDU 在 GB/T 15629.3 中的格式如图 15 所示。图 15 中除 PAE 以太网类型字段外的其他字段均在 6.6 中定义。

字段	八位位组
PAE 以太网类型	1~2
协议版本(6.6.2)	3
类型(6.6.3)	4
长度(6.6.4)	5~6
内容(6.6.5)	7~N

图 15 TAEPoL MPDU 在 GB/T 15629.3 中的格式

PAE 以太网类型字段长度为 2 个八位位组,表示 PAE 要使用的以太网类型值,定义见表 5。

6.5 标签 TAEPoL MPDU

PAE 发送的 TAEPoL MPDU 不应该附加虚拟 LAN(VLAN)的标签,但是可以附加优先级标签。所有的 PAE 都应能接受优先级标签和非优先级标签的 TAEPoL MPDU。

标签报头的结构见 IEEE Std 802.1Q。

6.6 TAEPoL PDU 的格式

6.6.1 概述

TAEPoL PDU 字段的格式见图 16,各个字段的定义见 6.6.2、6.6.3、6.6.4、6.6.5。

字段	八位位组序号
协议版本	1
类型	2
长度	3~4
内容	5~N

图 16 TAEPoL PDU 的格式

6.6.2 协议版本

该字段长度为 1 个八位位组,用一个无符号数表示。它的值表示 TAEPoL 帧的发送端所支持的 TAEPoL 协议版本。符合本标准的此字段值应为 0000 0001。

6.6.3 类型

该字段长度为 1 个八位位组,用一个无符号数表示,它的值标识所发送的帧的类型,定义如下:

- TAEP-Packet:值 0000 0000 表示帧载有 TAEP 分组;
- TAEPoL-Start:值 0000 0001 表示帧为 TAEPoL-Start 帧;
- TAEPoL-Logoff:值 0000 0010 表示帧为 TAEPoL-Logoff 请求帧;
- TAEPoL-Key:值 0000 0011 表示帧为 TAEPoL-Key 帧;
- TAEPoL-Encapsulated-ASF-Alert:值 0000 0100 表示帧载 TAEPoL-Encapsulated-ASF-Alert。

lert。

除上述 5 个值以外的所有其他值都不能使用,因为它们是为本协议未来扩展保留的。

TAEPoL-Encapsulated-ASF-Alert 类型的帧由 Alerting Standards Forum(ASF)作为允许告警(例如,特定的 SNMP 陷阱,参见 E.4),均通过非受控端口提供。在非受控端口收到的所有具有此类型的 TAEPoL 帧都会被传递给协议实体,该协议实体负责处理 ASF 告警确认,并进一步说明可以载此类型帧的警告消息的语法或语义,或者在接受到此种类型帧时的协议行动。

6.6.4 长度

该字段长度为 2 个八位位组,用一个无符号二进制数表示。该字段的值定义了内容字段的长度,值 0 表示没有内容字段。

6.6.5 内容

在负载有 TAEP-Packet 的 TAEPoL 帧内,该字段包含 5.4.5 定义的 TAEP 分组,即仅仅封装一个 TAEP 分组。

在负载有 TAEPoL-Key 的 TAEPoL 帧内,该字段包含 6.6.6.1 定义的 Key Descriptor,即仅仅封装一个 Key Descriptor。

在负载有 TAEPoL-Start 的 TAEPoL 帧内,该字段包含 6.6.6.2 定义的 Hello,即仅仅封装一个 Hello 或不封装任何内容。

在负载有 TAEPoL-Logoff 的 TAEPoL 帧内,该字段包含 6.6.6.3 定义的 Logoff,即仅仅封装一个 Logoff 或不封装任何内容。

在负载有 TAEPoL-Encapsulate-ASF-Alert 的 Packet Type 帧内,此字段包含 ASF 描述的 ASF 告警(参见 E.4)。

注: TAEPoL 帧可以负载的 TAEP 分组的最大值依赖于发送帧的 MAC 方法所支持的最大 MAC 帧的大小。

6.6.6 内容字段中封装数据定义

6.6.6.1 Key Descriptor

6.6.6.1.1 Key Descriptor 格式

Key Descriptor 的格式见图 17,各个字段的定义见 6.6.6.1.2、6.6.6.1.2、6.6.6.1.2、6.6.6.1.2、6.6.6.1.2、6.6.6.1.2、6.6.6.1.2、6.6.6.1.8。

长度——2个八位位组
标识——2个八位位组
重放计数器——8个八位位组
算法——OID
保留——8个八位位组
MIC——32个八位位组
协议数据—— n 个八位位组

图 17 Key Descriptor 的格式

6.6.6.1.2 长度字段

长度字段长度为 2 个八位位组,是一个整数,表示包含长度字段在内的 Key Descriptor 中所有字段的八位位组数。

6.6.6.1.3 标识字段

标识字段长度为 2 个八位位组,表示密钥的性质,格式见图 18。

B0	B1~B3	B4	B5	B6	B7~B8	B9~B15
Ack	Key type	Request	Encryption	MIC	Operation type	Reserved

图 18 标识字段的格式

标识字段中各位的定义如下：

- Ack 位为 1 个比特，当鉴别访问控制器发送的 TAEPoL 帧要求被响应，则在帧中被置位，否则复位。请求者响应帧中的 Ack 位使用和鉴别访问控制器发送帧的一样。
- Key type 位为 3 个比特，用于表示 TAEPoL 帧交互的密钥的类型。
- Request 位为 1 个比特，当请求者要求鉴别访问控制器开始一个密钥交换协议时，在请求者发送的帧中被置位。若鉴别访问控制器应请求者的要求开始一个密钥交换协议，鉴别访问控制器发送的帧中该位也置位表示是请求者要求开始密钥交换协议的。
- Encryption 位为 1 个比特，如果 Key Data 字段中的 Key 是密文，则该位置位，否则复位。
- MIC 位为 1 个比特，如果 TAEPoL 帧中包含 MIC 值，该位置位，否则复位。
- Operation type 为 2 个比特，表示使用该 TAEPoL 帧的操作类型。例如表示密钥的建立、更新或者删除。
- Reserved 为 7 个比特，保留。

6.6.6.1.4 重放计数器字段

重放计数器为 8 个八位位组长，是一个整数。当基共享密钥建立以后，重放计数器初始化为 0。请求者在响应一个 TAEPoL 帧时，使用收到的帧中的重放计数器值作为重放计数器值。它是一个序列，协议用它来检查重放攻击。请求者在收到有效的 TAEPoL-Key 帧后，递增收到帧中的重放计数器值作为自己的重放计数器值。鉴别访问控制器在收到有效的 TAEPoL-Key 帧后，递增收到帧中的重放计数器值作为自己的重放计数器值。有效的帧是指帧的 MIC 值校验正确。

6.6.6.1.5 算法字段

算法字段长度是可变的，该字段是一个 OID，采用 DER 编码。它表示 MIC 字段使用的算法，该算法的选择需符合国家密码政策的规定。

6.6.6.1.6 保留字段

保留字段为 8 个八位位组。

6.6.6.1.7 MIC 字段

MIC 字段为 32 个八位位组，采用算法字段描述算法计算。它是 TAEPoL-Key 帧的 MIC 值，包含帧的全部内容，在进行 MIC 计算时 MIC 字段屏蔽为 0。

6.6.6.1.8 协议数据字段

协议数据字段是变长字段，包含用于密钥交换协议的附加数据，其封装格式见图 19。

类型	数据
----	----

图 19 协议数据的格式

其中：

- 类型字段为 1 个八位位组：
 - 0x00 保留；
 - 0x01~0xFE，用于分配给不同的协议；
 - 0xFF 私有，用于私有协议的使用。

6.6.6.2 Hello 格式

Hello 的格式见图 20。

地址类型——1个八位位组
地址长度——1个八位位组
地址
鉴别访问控制器的地址
邻居的地址
序列号——6个八位位组
优先级——1个八位位组
保留——1个八位位组
MIC——32个八位位组

图 20 Hello 的格式

其中：

——地址类型字段：表示后面地址的种类，取值定义如下：

- 0 无地址；
- 1 MAC 地址；
- 其他值保留。

——地址长度字段：表示一个地址字段的八位位组数，当地址类型为 0 时，该字段必须为 0。

——地址字段：表示 PAE 的地址。

——鉴别访问控制器的地址字段：表示两个 PAE 中角色为鉴别访问控制器的 PAE 的地址。

——邻居地址字段：表示邻居 PAE 的地址。

——序列号字段：表示单调递增的整数，用于防止重放攻击。

——优先级字段：表示 PAE 成为鉴别访问控制器的优先程度，取值范围是 0~255，值越大，优先级越高。

——MIC 字段：表示完整性校验码。如果双方支持安全 Hello，即设置了预共享密钥且采用 HMAC-SHA256 算法计算，则它是 TAEPoL-Start 帧的 MIC 值，包含帧的全部内容。计算时 MIC 字段屏蔽为 0；如果双方不支持安全 Hello，该字段值为 0。

6.6.6.3 Logoff 格式

Logoff 的格式见图 21。

随机数——32个八位位组
MIC——32个八位位组

图 21 Logoff 的格式

其中：

——随机数字段为 32 个八位位组。

——MIC 字段为 32 个八位位组。它采用 HMAC-SHA256 算法对 TAEPoL-Logoff 帧进行 MIC 计算，包含帧的全部内容，计算时 MIC 字段屏蔽为 0。

6.7 接收到 TAEPoL PDU 和 TAEPoL 协议格式处理的确认

当且仅当如下条件为真时,一个 PAE 才应处理收到 TAEPoL PDU:

- a) 接入链路的封装指示该 PDU 为 TAEPoL PDU。在 IEEE 802 网络中,PAE Ethernet Type 字段包含如 6.3 描述的 PAE Ethernet Type 的值;
- b) 类型字段包含 TAEP-Packet、TAEPoL-Start、TAEPoL-Logoff 或 TAEPoL-Key 其中之一,详见 6.6.3。

注: 6.7 描述的操作模式如下,在受控端口和非受控端口,潜在的 MAC 收到的任何 PDU 均可用。端口访问实体总是连接到非受控端口的(永远不会连到受控端口);因此,无论受控端口的状态如何,受控端口可用的任何 TAEPoL PDU 都应被丢弃。

7 对等鉴别访问控制协议

7.1 概述

鉴别过程利用 TAEP 在请求者和鉴别访问控制器之间进行鉴别。TAEP 是一个支持多种鉴别协议的封装,通过 TAEP,各种鉴别协议可以在请求者和鉴别访问控制器之间使用。TAEPoL 在请求者和鉴别访问控制器之间传递消息,每个 PAE 包含两个逻辑分离的组件,一套 PCAP 状态机和一个高层的实体。对于请求者,高层实体包括 TAEP 功能;对于鉴别访问控制器,高层实体包含 TAEP 功能和其他用于鉴别访问控制器及鉴别服务器通信的必要功能。本章描述了 PCAP 状态机、TAEP 以及 PCAP 与高层实体的接口。

图 22 描述了请求者和鉴别访问控制器中 PCAP 状态机和高层实体的接口。系统的 PortEnable 信号通告给 PCAP,指示一个端口是活跃的。PCAP 在物理层和高层之间传递 TAEP 消息,来自 PCAP 的 taepReq 指示 TAEP 有消息要处理,请求者的消息流使用来自 TAEP 的 taepResp/taepNoResp 参数指示 TAEP 准备好接收另外一个消息。鉴别访问控制器的消息流控制和请求者类似。在高层实体中,TAEP 和 TAEP 关联的鉴别协议驱动鉴别会话。一旦完成会话,高层实体使用 taepSuccess 和 taepFail 通知 PCAP。

由于 TAEP 消息都由 TAEP 协议实体产生,因此将高层实体和 PCAP 分开。TAEP 除了允许事先确定的鉴别方法,还允许在决定鉴别方法前交换更多的信息。

鉴别访问控制器的 TAEP 实体要处理解析 TAEP 消息,根据所选择的具体鉴别协议,不仅要和请求者通信,还要和鉴别服务器通信。若鉴别服务器不支持该鉴别协议,鉴别服务器将返回 NAK 消息给鉴别访问控制器。一般情况下,鉴别访问控制器应首先获得鉴别服务器支持的鉴别协议类型。当鉴别访问控制器和请求者协商鉴别协议时,选择鉴别服务器支持的类型。不同的鉴别协议会完成不同的鉴别功能,可为单向鉴别无密钥、单向鉴别有密钥、双向鉴别无密钥和双向鉴别有密钥。鉴别访问控制器的 TAEP 实体应对 TAEP 消息进行处理,而非简单的传递给鉴别服务器。在请求者、鉴别访问控制器和鉴别服务器之间存在两个 TAEP 会话——请求者与鉴别访问控制器之间、鉴别访问控制器与鉴别服务器之间。请求者和鉴别访问控制器根据 TAEP 关联的鉴别协议运行的结果,使用 taepSuccess 和 taepFail 通知 PCAP,从而 PAE 可以根据结果控制受控端口的状态。

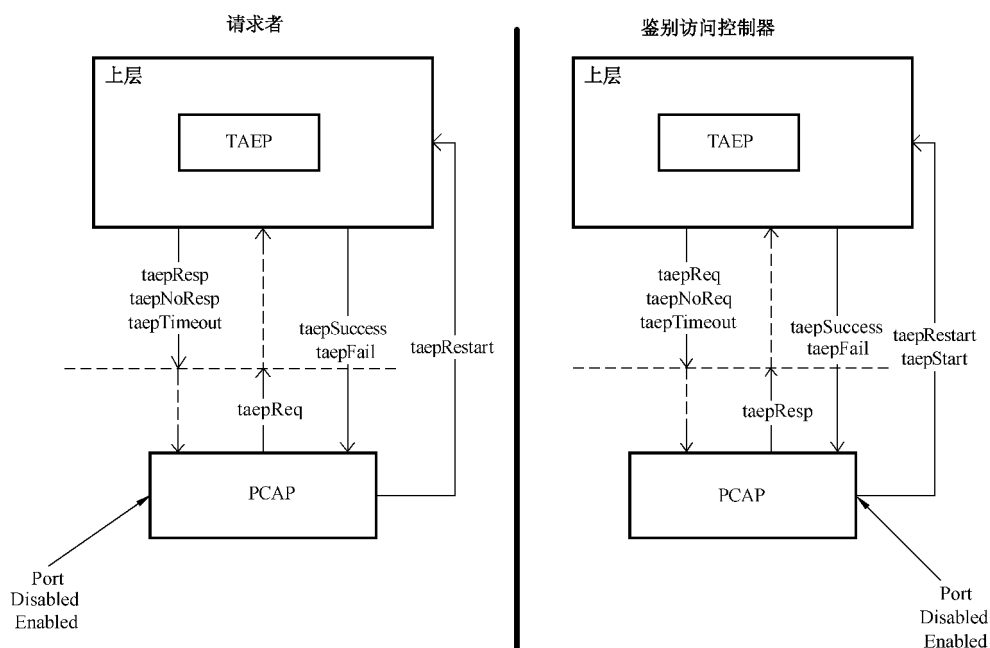


图 22 PCAP 状态机与高层的关系

7.2 鉴别过程

鉴别可以由请求者或鉴别访问控制器发起,如果鉴别功能在端口上激活(例如:端口支持鉴别访问控制器 PAE, AuthControlledPortControl 参数设置为 Auto, SystemAuthControl 参数设置为 Enabled),当鉴别访问控制器 PAE、请求者 PAE 收到端口从 Disabled 变到 Enabled 状态的信息或收到 TAEPoL-Start 帧,其将发起鉴别。在鉴别开始之前,鉴别访问控制器和请求者的受控端口是非授权的。

当 PAE 是请求者角色时,它发送 TAEPoL-Start 帧请求对方 PAE 开始鉴别。TAEPoL-Start 帧不包含 Hello 字段。

当 PAE 是鉴别访问控制器角色时,如果鉴别访问控制器 PAE 收到端口的状态是 Enabled 或收到请求者 PAE 发来的 TAEPoL-Start 帧时,则鉴别访问控制器 PAE 通知上层协议实体 TAEP 开始鉴别。鉴别访问控制器 PAE 把来自 TAEP 协议实体的 TAEP-Request 帧发送出去开始鉴别协议。一般情况下,TAEP 会通过某一个 TAEP-Request 帧开始鉴别协议,然而任何一个 TAEP-Request 帧都可以开始鉴别协议。收到 TAEP-Request 帧的请求者回复一个 TAEP-Response 帧给鉴别访问控制器。

当 PAE 是自适应角色时,如果收到 TAEP-Request 帧,则将角色设置为请求者,开始鉴别过程;如果收到不包含 Hello 字段的 TAEPoL-Start 帧,则将角色设置为鉴别访问控制器,开始鉴别过程;如果收到端口的状态是 Enabled 或收到对方 PAE 发来的包含 Hello 字段的 TAEPoL-Start 帧时,它将发送 TAEPoL-Start 帧来确定自己的角色。TAEPoL-Start 帧中的优先级为 MIB 变量 Priority 的值,如果没有收到对方 PAE 的 TAEPoL-Start 帧,鉴别访问控制器 PAE 的地址为自己的地址,邻居地址为空;如果收到对方 PAE 的 TAEPoL-Start 帧,邻居地址为对方 PAE 的地址。若收到 TAEPoL-Start 帧中的优先级值大于自己的优先级值,鉴别访问控制器 PAE 的地址为对方 PAE 的地址,如果收到 TAEPoL-Start 帧中的优先级值小于自己的优先级值,鉴别访问控制器 PAE 的地址为自己的地址;如果优先级相同,则地址大的 PAE 的地址作为鉴别访问控制器的地址。若自适应 PAE 收到对方 TAEPoL-Start 帧,根据优先级和地址决定自己是请求者时,按照请求者的角色进行状态转换。如为鉴别访问控制器,则按照鉴别访问控制器的角色进行状态转换。

鉴别访问控制器和请求者都可以提供定期的重鉴别功能,鉴别访问控制器 PCAP 可以在任何时候通知上层实体 TAEP 开始重鉴别,请求者 PCAP 也可以在任何时候发送 TAEPoL-Start 帧给鉴别访问控制器 PAE 开始重鉴别。对于请求者提出的重鉴别请求,鉴别访问控制器 PCAP 可以对 TAEPoL-Start 帧进行验证后再决定是否开始重鉴别。当受控端口在重鉴别前处于已授权状态时,在重鉴别进行期间,受控端口的状态保持不变。如果重鉴别失败,则受控端口的状态从已授权的转变为非授权的。

如果请求者希望鉴别访问控制器 PAE 执行 Logoff(例如,将受控端口置为非授权状态),请求者 PAE 发起一个 TAEP-Logoff 消息给鉴别访问控制器 PAE,那么鉴别访问控制器 PAE 立刻将受控端口置为非授权状态。

请求者和鉴别访问控制器 PAE 都可以定期的使授权状态信息过期而触发重鉴别,超时时间间隔是从上一次授权状态确认后经过的 reAuthPeriod 秒。状态变量 reAuthEnabled 控制是否允许定期重鉴别。

管理员可以设置是否允许重鉴别和更改 reAuthPeriod 变量,默认是禁止重鉴别。当允许重鉴别时,reAuthPeriod 的默认值是 3 600 s。

鉴别访问控制器 PAE 负责它和请求者 PAE 之间消息的重传,准确的说,是 TAEP 实体(PAE 的一部分)负责重传,而不是 PCAP。因此,如果鉴别访问控制器 PAE 和请求者 PAE 之间的 TAEP-Packet 丢失了,鉴别访问控制器 PAE 将重传。TAEP-Start 帧是个例外,如果请求者 PAE 认为需要,它可以重传该帧。在 FAIL 和 SUCCESS 状态下的 TAEP 消息也都可以由鉴别访问控制器 PAE 或请求者 PAE 进行重传,因为这些消息并不被鉴别访问控制器 PAE 或请求者 PAE 确认。

鉴别访问控制器的 TAEP 实体负责它和鉴别服务器之间消息的重传。

当支持对等控制的设备与不支持对等控制的传统设备通信时,其兼容性考虑如下:如果 MIB 变量 systemRole 为请求者 REQ,traditionEnabled 为 TRUE,则当 PAE 发送的 TAEPoL-Start 帧没有响应,重传适当次数后,请求者可以认为它已经被对方授权,并强制打开自己受控端口。

注:这种情况会带来安全的隐患,这意味着请求者也不对对方进行鉴别就授权。因此,使用者应该谨慎将 traditionEnabled 置为 TRUE。

7.3 PCAP 状态机

7.3.1 一般要求

本章规定了 PCAP 状态机的下述状态:

- a) 端口时钟状态机;
- b) 鉴别访问控制器 PAE 状态机;
- c) 密钥传送状态机;
- d) 密钥接收状态机;
- e) 重鉴别时钟状态机;
- f) 鉴别访问控制器后台状态机;
- g) 控制方向状态机;
- h) 请求者 PAE 状态机;
- i) 请求者后台状态机;
- j) 自适应端口状态机。

这些状态机是基于每端口定义的,表 6 总结了状态机对于实现鉴别访问控制器或请求者功能的支持要求。表中 M 表示必须支持,O 表示可选支持。

表 6 鉴别访问控制器和请求者对状态机的支持要求

	功 能 支 持	
	鉴别访问控制器	请 求 者
端口时钟状态机	M	M
鉴别访问控制器 PAE 状态机	M	
密钥传送状态机	O	O
密钥接收状态机	O	O
重鉴别计时器状态机	M	M
鉴别访问控制器后台状态机	M	
控制方向状态机	M	
请求者 PAE 状态机		M
请求者后台状态机		M
自适应端口状态机	O	O

7.3.2 状态图中的符号约定

规程的操作可用状态图描述。每一状态图表示一个功能域,由一些连接且互斥的状态组成。每一个状态都由下图所示的方框表示,在任一给定时刻仅有一个状态起作用。这些状态被水平线分为两部分。上部分为状态的名称,下部分包含所产生信号的名称,方括号内的短语描述动作。表 7 描述了状态图中符号意义。

表 7 状态图中符号

符号	表 示
()	在布尔表达式中强制设定运算的顺序或在状态框中作为参数的边界
;	在状态框作为一个动作结束分隔符
=	赋值动作,运算符右侧的表达式结果赋给运算符左侧的变量
!	逻辑非
&&	逻辑与
	逻辑或
!=	不等于,如果运算符左边和右边的值不相同,结果为真;否则为假
==	等于,如果运算符左边和右边的值相同,结果为真;否则为假

状态间所有允许的转换通过由起始状态到终止状态的箭头以图形化表示。转移上的标记是在采用转移必须满足的条件。标记 UCT 指定无条件转移。所有条件的结果是真或是假,如果是真,则条件被满足。

状态的转移和消息的发送接收都是瞬间发生地。当进入一个状态,并且离开状态的条件未立即满足时,该状态将持续有效,以连续的方式发送信息并执行该状态内所包含的动作。

状态图包含对它们描述的规程的权威性说明。当描述性文本和状态图出现明显矛盾时,则状态图应处于优先地位。然而,这并不表示可以忽视在状态图内没有表述而在与之平行文本中的任何清晰描述。

7.3.3 时钟和全局变量

7.3.3.1 时钟

所有状态机中定义的时钟,如果其值不为0,则该值将被端口时钟状态机递减。PCAP 状态机中使用时钟的粒度是 1 s,启动时钟的初始值是整数,表示一个整数秒的时间段。以下是各个时钟参数的描述:

- a) aWhile。鉴别访问控制器后台用来确定在超时之前,等待上层进程的响应的的时间,该时钟的初始值是 serverPeriod。
- b) authWhile。请求者后台用来决定在超时之前,等待来自鉴别访问控制器的请求的时间,该时钟的初始值是 authPeriod。
- c) heldWhile。请求者状态机用来确定在多长时间不向鉴别访问控制器连接,该时钟的初始值为 heldPeriod。
- d) quietWhile。鉴别访问控制器状态机用来确定在多长时间不向请求者连接,该时钟的初始值为 quietPeriod。
- e) reAuthWhen。重鉴别时钟状态机用来确定何时进行重鉴别,该时钟的初始值为 reAuthPeriod。
- f) startWhen。请求者状态机用来确定在何时传送 TAEPoL-Start 帧,该时钟的初始值为 startPeriod。

7.3.3.2 全局变量

全局变量可以被一个以上的状态机访问,它用来执行状态机间通信和初始化功能,如下:

- a) authAbort。该变量被鉴别访问控制器 PAE 状态机设置为 TRUE,通知后台鉴别状态机终止鉴别过程;当鉴别过程被终止后,后台鉴别状态机将该变量设置为 FALSE。
- b) authFail。如果鉴别进程失败,该变量被设置为 TRUE;该变量在初始化鉴别前,被鉴别访问控制器状态机设置为 FALSE。
- c) authPortStatus。该变量表示鉴别访问控制器状态机当前的授权状态,根据状态机的运行被设置为 Unauthorized 或 Authorized。如果没有实现鉴别访问控制器 PAE 状态机,该变量值为 Authorized。
- d) authStart。该变量被鉴别访问控制器 PAE 状态机设置为 TRUE,通知后台鉴别状态机开始鉴别过程;一旦后台鉴别过程开始后,后台鉴别状态机将该变量设置为 FALSE。
- e) authTimeout。如果鉴别进程获得请求者的响应失败,该变量被设置为 TRUE。该变量可能被管理行为设置,也可以在 AUTHENTICATED 状态中由状态机设置。该变量由鉴别访问控制器 PAE 状态机设置为 FALSE。
- f) authSuccess。如果鉴别过程成功,该变量被设置为 TRUE。在开始鉴别之前,该变量被鉴别访问控制器 PAE 状态机设置为 FALSE。
- g) taepFail。如果上层确定鉴别失败,它设置该变量为 TRUE。
- h) taepolTaep。如果收到一个封装 TAEP 分组的 TAEPoL 帧,外部实体设置该变量为 TRUE。
- i) taepSuccess。如果上层确定鉴别成功,它设置该变量为 TRUE。
- j) taepTimeout。如果上层确定请求者对请求没有响应,它设置该变量为 TRUE。
- k) initialize。该变量是被外部控制的。当该变量被置位时,所有的 TAEPoL 状态机被强迫进入

初始状态。PCAP 状态机将一直保持在初始状态,直到该变量被复位。

- l) keyAvailable。当一个新的密钥可以被用于密钥状态机开始新的密钥交换,该变量被外部实体设置为 TRUE。当密钥传输状态机已经传送密钥,该变量被设置为 FALSE。
- m) keyDone。当密钥状态机处于可以检测 portValid 变量的状态时,该变量被设置为 TRUE。
- n) keyRun。当传输密钥状态机应该运行时,该变量被 PCAP 状态机设置为 TRUE。PAE 设置该变量为 FALSE,指示 PAE 状态机已经被复位,密钥状态机应该被中止运行。
- o) keyEnabled。该变量表示受控端口是否受密钥交换的结果管理,TRUE 表示受管理。
- p) keyFail。该变量为 TRUE 表示密钥交换过程失败。
- q) keySuccess。该变量为 TRUE 表示密钥交换过程成功。
- r) portControl。该变量的值根据相应端口的 AuthControlledPortControl 和 SystemAuthControl 参数得出。如果 SystemAuthControl 参数设置为 Enabled,portControl 的值直接反映了 AuthControlledPortControl 参数的值,如果 SystemAuthControl 参数设置为 Disabled,portControl 的值是 ForceAuthorized。它可以有以下取值:
 - 1) ForceUnauthorized。受控端口被保持在 Unauthorized 状态。
 - 2) ForceAuthorized。受控端口被保持在 Authorized 状态。
 - 3) Auto。受控端口的状态根据请求者和鉴别访问控制器之间的鉴别交换结果设置为 Authorized 或 Unauthorized。
- s) portEnabled。该变量是外部控制的,它的值反映了端口的 LAC 的操作状态。如果端口的 LAC 服务处于可操作状态,该变量的值为 TRUE;否则为 FALSE。
- t) portValid。当端口已经被认为符合需要的安全等级,该变量被设置为 TRUE;否则为 FALSE。
- u) reAuthenticate。该变量在 reAuthWhen 计时器超时后被 Reauthentication Timer 状态机设置为 TRUE,该变量也可以被管理行为设置为 TRUE。它在鉴别访问控制器 PAE 状态机运行中被设置为 FALSE。重鉴别也许不会立刻开始,鉴别访问控制器不会中断当前的鉴别,等它结束后,鉴别访问控制器开始一个新的鉴别。
- v) reqAbort。该变量被请求者 PAE 状态机设置为 TRUE,用来通知请求者后台状态机中止鉴别序列。它被请求者后台状态机设置为 FALSE。
- w) reqFail。该变量被请求者 PAE 状态机设置为 FALSE,在不成功鉴别序列完成后,它被请求者后台状态机设置为 TRUE。
- x) reqPortStatus。该变量反映了请求者受控端口的授权状态,它在状态机运行中被设置为 Authorized 或 Unauthorized。如果请求者状态机没有实现,该变量为 Authorized。
- y) reqStart。该变量被请求者 PAE 状态机设置为 TRUE,用来通知请求者后台状态机开始鉴别序列。它被请求者后台状态机设置为 FALSE。
- z) reqSuccess。该变量被请求者 PAE 状态机设置为 FALSE,在成功鉴别序列完成后,它被请求者后台状态机设置为 TRUE。
- aa) reqTimeout。该变量被请求者 PAE 设置为 FALSE,如果鉴别序列超时,它被请求者后台状态机设置为 TRUE。
- ab) systemRole。该变量可以取 selfAdapted、REQ、AAC 三种值。它可以由管理层设定,也可以由自适应端口状态机更改。
- ac) reKeyCount。该变量表示更新密钥的次数,被密钥接收状态机设置。

ad) reKeyMax。为常量,该常量表示允许更新密钥的尝试次数,其默认值是 255。

7.3.4 端口计时器状态机

给定端口的端口计时器状态机负责每一秒消耗该端口计时器变量,对应于外部系统时钟功能。计时器变量被端口的各个状态机设置为初始值。端口计时器状态机按照端口计时器状态机描述的功能实现(参见 IEEE Std 802.1x),见图 23。

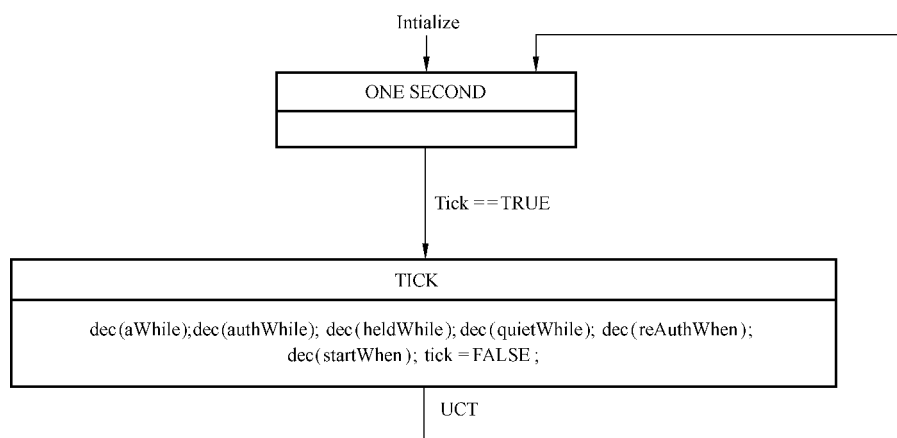


图 23 端口计时器状态机

变量 Tick:为了模拟外部系统时钟功能的每一秒嘀嗒功能,设置该变量。当系统时钟产生一秒嘀嗒时,该变量设置为 TRUE,端口计时器状态机的操作设置该变量为 FALSE。

7.3.5 鉴别访问控制器状态机

7.3.5.1 一般要求

鉴别访问控制器的状态机见图 24。鉴别访问控制器状态机有以下状态:

- a) INITIALIZE;
- b) DISCONNECTED;
- c) RESTART;
- d) CONNECTING;
- e) AUTHENTICATING;
- f) AUTHENTICATED;
- g) ABORTING;
- h) HELD;
- i) FORCE_AUTH;
- j) FORCE_UNAUTH;
- k) AUTHORIZED。

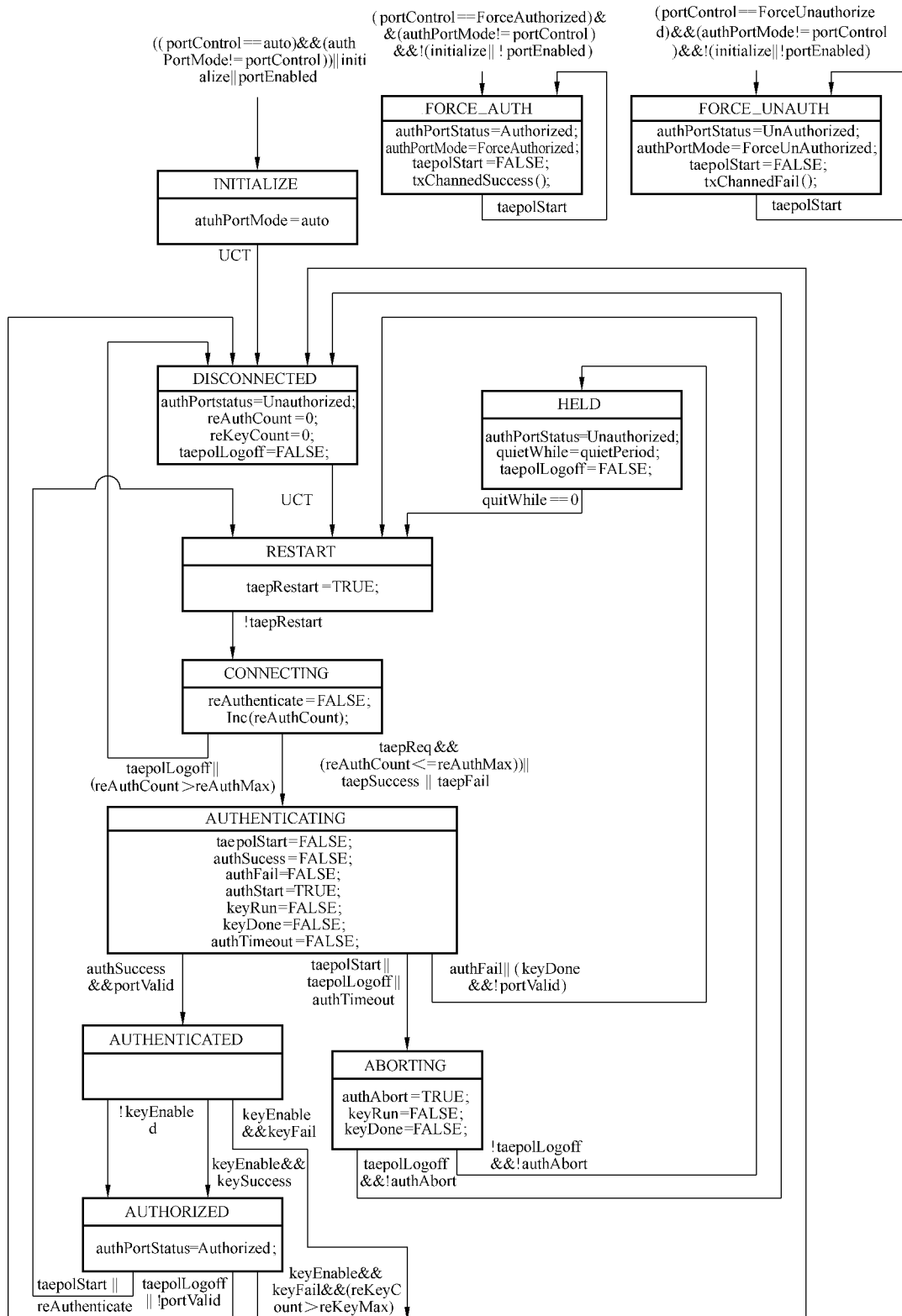


图 24 鉴别访问控制器状态机



7.3.5.2 鉴别访问控制器 PAE 状态机定义中使用的变量、常量和过程

7.3.5.2.1 变量

鉴别访问控制器 PAE 状态机定义中使用了以下几种变量：

- a) taepolLogoff: 如果收到封装 TAEPoL-Logoff 的 TAEPoL 帧, 则该变量被设置为 TRUE, 否则该变量被鉴别访问控制器 PAE 状态机的操作设置为 FALSE;
- b) taepolStart: 如果收到封装 TAEPoL-Start 的 TAEPoL 帧, 则该变量被设置为 TRUE, 否则该变量被鉴别访问控制器 PAE 状态机的操作设置为 FALSE;
- c) taepReq: 当发送一个 TAEP 帧给请求者时, 该变量被上层设置为 TRUE; 当 TAEP 帧已被传送, 该变量被鉴别访问控制器后台状态机设置为 FALSE;
- d) taepRestart: 该变量被鉴别访问控制器状态机设置为 TRUE, 来指示它因为 TAEPoL、超时、初始化事件正在重启状态机;
- e) authPortMode: 该变量和 PortControl 一起用于表示鉴别访问控制器 PAE 状态机在 Auto 和 non-Auto, 操作状态间的切换, 其取值如下:
 - 1) ForceUnauthorized: 受控端口应保持在 Unauthorized 状态;
 - 2) Auto: 受控端口的状态根据请求者和鉴别访问控制器之间的鉴别交换结果设置为 Authorized 或 Unauthorized;
 - 3) ForceAuthorized: 受控端口应保持在 Authorized 状态;
- f) reAuthCount: 该变量用于统计重新进入 CONNECTING 状态的次数, 如果统计值超过 reAuthMax, 则下次鉴别开始之前, 将强制端口进入 Unauthorized 状态。

7.3.5.2.2 常量

鉴别访问控制器 PAE 状态机定义中定义了以下几个常量：

- a) quietPeriod: 该常量用于 quietWhile 计时器, 默认值为 60 s, 可被管理实体设置为 0~65 535 s 中的任何值;
- b) reAuthMax: 该常量表示端口进入 Unauthorized 状态前允许重鉴别的尝试次数, 其默认值是 2。

7.3.5.2.3 过程

鉴别访问控制器 PAE 状态机定义中定义了以下几个过程：

- a) txCannedFail(): 封装 TAEP-Failer 分组的 TAEPoL 帧从鉴别访问控制器发送给请求者。若端口上没有 TAEP 通信, TAEP 分组的 identifier 字段可以设置为任何值; 若端口上有 TAEP 通信, TAEP 分组的 identifier 字段应置为不同于最近一次发送的 TAEP 分组的其他分组。
- b) txCannedSuccess(): 封装 TAEP-Success 分组的 TAEPoL 帧从鉴别访问控制器发送给请求者。若端口上没有 TAEP 通信, TAEP 分组的 identifier 字段可以设置为任何值; 若端口上有 TAEP 通信, TAEP 分组的 identifier 字段应置为不同于最近一次发送的 TAEP 分组。

7.3.5.3 鉴别访问控制器 PAE 状态机维护的计数器

7.3.5.3.1 概述

为诊断目的, 鉴别访问控制器 PAE 状态机可对下述计数器进行维护。这些计数器的值可通过第 8 章描述的管理操作来访问。假设这些计数器的值在超过最大值后会回转为 0。

7.3.5.3.2 authEntersConnecting

统计状态机从任何状态转移到 CONNECTING 状态的转移次数。

7.3.5.3.3 authTaepLogoffsWhileConnecting

统计收到 TAEPoL-Logoff 消息时,状态机从 DISCONNECTED 状态转移到 CONNECTING 状态的转移次数。

7.3.5.3.4 authEntersAuthenticating

统计收到 TAEP-Response/Identity 消息时,状态机从 CONNECTING 状态转移到 AUTHENTICATING 状态的转移次数。

7.3.5.3.5 authAuthSuccessesWhileAuthenticating

统计收到后台鉴别状态机指示请求者鉴别成功的消息时(authSuccess = TRUE),状态机从 AUTHENTICATING 状态转移到 AUTHENTICATED 状态的转移次数。

7.3.5.3.6 authAuthTimeoutsWhileAuthenticating

统计收到后台鉴别状态机指示请求者鉴别超时的消息时(authTimeout = TRUE),状态机从 AUTHENTICATING 状态转移到 ABORTING 状态的转移次数。

7.3.5.3.7 authAuthFailWhileAuthenticating

统计收到后台鉴别状态机指示请求者鉴别失败的消息时(authFail = TRUE),状态机从 AUTHENTICATING 状态转移到 HELD 状态的转移次数。

7.3.5.3.8 authAuthTaepStartsWhileAuthenticating

统计收到来自请求者的 TAEPoL-Start 的消息(是指 TAEPoL-Start 传来的消息,还是 TAEPoL-Start 消息?)时,状态机从 AUTHENTICATING 状态转移到 ABORTING 状态的转移次数。

7.3.5.3.9 authAuthTaepLogoffWhileAuthenticating

统计收到来自请求者的 TAEPoL-Logoff 的消息时,状态机从 AUTHENTICATING 状态转移到 ABORTING 状态的转移次数。

7.3.5.3.10 authAuthReauthsWhileAuthenticated

统计收到重鉴别请求时(reAuthenticate = TRUE),状态机从 AUTHENTICATED 状态转移到 RESTART 状态的转移次数。

7.3.5.3.11 authAuthTaepStartsWhileAuthenticated

统计收到来自请求者的 TAEPoL-Start 的消息时,状态机从 AUTHENTICATED 状态转移到 CONNECTING 状态的转移次数。

7.3.5.3.12 authAuthTaepLogoffWhileAuthenticated

统计收到来自请求者的 TAEPoL-Logoff 的消息时,状态机从 AUTHENTICATED 状态转移到 DISCONNECTED 状态的转移次数。

7.3.5.4 INITIALIZE 状态

如果 portControl 变量设置为 Auto, 而 authPortMode 变量设置为 Auto 以外其他的值, 或者端口的 LAC 不可操作, 或者状态机正在初始化, 则进入该状态, 并将 authPortMode 变量的值设置为 Auto。初始化完成后, 如果 authPortMode 和 portControl 的值相同, 并且端口 LAC 是可操作的, 鉴别访问控制器状态机将无条件转移到 DISCONNECTED 状态; 若在 authPortMode 和 portControl 的值不同的情况下, 将转移到 FORCE_AUTH 或 FORCE_UNAUTH 状态。

7.3.5.5 DISCONNECTED 状态

如果收到来自请求者的显式注销请求, 该状态可以从 CONNECTING、AUTHENTICATED 和 ABORTING 状态转入。在任何时候, 如果端口 LAC 不可操作, PAE 将转入 DISCONNECTED 状态。在该状态下, authPortStatus 变量被设置为 Unauthorized, 因此 AuthControlledPortStatus 的值也被置为 Unauthorized, TaepolLogoff 变量被设置为 False, reAuthCount 复位。

该状态无条件转移到 RESTART 状态。

7.3.5.6 RESTART 状态

当鉴别访问控制器 PAE 需要通知上层它已经重启了, 进入 RESTART 状态, 鉴别访问控制器状态机可以通过以下几个途径进入该状态:

- 从 DISCONNECTED 状态无条件转移到 RESTART 状态;
- 在 AUTHENTICATE 状态收到 taepolStart 或 reAuthenticate 信号, 将从 AUTHENTICATE 状态进入 RESTART 状态;
- 没有收到 taepolLogoff 而离开 ABORTING 状态, 将从 ABORTING 状态进入 RESTART 状态;
- 当 quietWhile 计时器超时, 从 HELD 状态进入 RESTART 状态。

taepRestart 变量被设置为 TRUE 来通知鉴别访问控制器 PAE 已经重启了。当 TAEP 已经确认了重启, 将 taepRestart 置为 FALSE 后, 鉴别访问控制器 PAE 从 RESTART 状态进入 CONNECTING 状态。

7.3.5.7 CONNECTING 状态

当鉴别访问控制器从 RESTART 状态进入该状态后, 鉴别访问控制器端口是可操作的, 上层处于同步状态, 准备好和请求者建立通信。鉴别访问控制器通过以下几个途径退出该状态:

- 如果收到 TAEPoL-Logoff 帧, 状态机转变到 DISCONNECTED 状态, 以强迫端口置为 Unauthorized 状态。
- 当上层准备好发送最初的 TAEP-Request 消息, 或者 TAEP 成功, 或者 TAEP 失败, 从该状态转移到 AUTHENTICATING 状态。
- TAEP 负责最初的和后续的 TAEP-Request 消息, 然而重新进入该状态将更新记录 (reAuthCount), 如果记录 (reAuthCount) 超过了最大值 (reAuthMax), 将转移到 DISCONNECTED 状态, 设置 authPortStatus 为 Unauthorized 并清除 reAuthCount。

7.3.5.8 AUTHENTICATING 状态

在该状态, 请求者正在被鉴别。authSuccess、authTimeout、authFail、keyRun 和 keyDone 变量被设

置为 FALSE, authStart 变量被设置为 TRUE 来通知后台鉴别状态机开始鉴别过程。鉴别过程可以产生以下结果:

- 鉴别过程因为请求和响应序列的超时而停止, 变量 authTimeout 设置为 TRUE, 致使状态机转移到 ABORTING 状态;
- 鉴别过程因为鉴别结果是失败而停止, 变量 authFail 设置为 TRUE, 致使状态机转移到 HELD 状态;
- 鉴别过程因为鉴别结果是成功而停止, 变量 authSuccess 设置为 TRUE, 如果 portValid 是 TRUE, 将致使状态机转移到 AUTHENTICATED 状态。

如果收到 TAEPoL-Start 或 TAEPoL-Logoff 帧, 状态机将转移到 ABORTING 状态以中止鉴别过程。如果收到重鉴别请求, 该请求不会立刻生效, 它要等到鉴别进程完成以后才会生效。

7.3.5.9 AUTHENTICATED 状态

在该状态下, 鉴别访问控制器已经成功地鉴别了请求者, 但还不能改变端口的状态。

如果不进行密钥交换, 状态机转移到 AUTHORIZED 状态。如果进行密钥交换, 密钥交换成功则转移到 AUTHORIZED 状态; 失败则转移到 DISCONNECTED 状态。

7.3.5.10 ABORTING

在该状态, 鉴别过程因为收到 TAEPoL-Start、TAEPoL-Logoff 或者 authTimeout 而提起中止。authAbort 变量被设置为 TRUE, 通知后台鉴别状态机它应该停止鉴别过程。

如果是 TAEPoL-Logoff 导致状态机进入 ABORTING 状态, 状态机从该状态转移到 DISCONNECTED 状态; 否则一旦鉴别过程中止, 状态机转移到 RESTART 状态。

7.3.5.11 HELD 状态

在该状态下, 状态机忽略并丢弃所有的 TAEPoL 帧, 以削弱强力攻击。由于鉴别失败, 该状态从 AUTHENTICATING 状态转移过来。authPortStatus 变量被设置为 Unauthorized, quietWhile 计时器采用 quietPeriod 初始值启动。当 quietWhile 计时器超时时, 状态机转移到 RESTART 状态。

7.3.5.12 FORCE_AUTH

当且仅当以下 4 个条件同时为真, 状态机将从任何其他状态转移到 FORCE_AUTH 状态:

- portControl 变量设置为 ForceAuthorized;
- authPortMode 变量设置为除 ForceAuthorized 以外的其他值;
- 端口的 LAC 可操作;
- 状态机未处于初始化状态。

authPortStatus 被设置为 Authorized 时, 鉴别访问控制器发送 TAEP Success 分组给请求者, 如果收到来自请求者的 TAEPoL-Start 消息, 将重新进入该状态, 发送 TAEP Success 分组。

7.3.5.13 FORCE_UNAUTH

当且仅当以下 4 个条件同时为真, 状态机将从任何其他状态转移到 FORCE_UNAUTH 状态:

- portControl 变量设置为 ForceUnauthorized;
- authPortMode 变量设置为除 ForceUnauthorized 的其他值;
- 端口的 LAC 可操作;
- 状态机未处于初始化状态。

authPortStatus 被设置为 Unauthorized 时, 鉴别访问控制器发送 TAEP Failure 分组给请求者, 如果收到来自请求者的 TAEPoL-Start 消息, 将重新进入该状态, 并发送 TAEP Failure 分组。

7.3.5.14 AUTHORIZED

在 AUTHORIZED 状态下,鉴别访问控制器已对请求者鉴别成功, portValid 变量为 TRUE, authPortStatus 变量设置为 Authorized, reauth 计数器被复位以允许后续的重鉴别请求。

如果收到 TAEPoL-Start 帧或需要进行重鉴别,则状态机将转移到 RESTART 状态以和请求者进行重鉴别。

如果收到 TAEPoL-Logoff 帧,或 portValid 值为 FALSE,则状态机将转移到 DISCONNECTED 状态以迫使 authPortStatus 值置为 Unauthorized。

如果在该状态下进行密钥更新过程时的密钥交换失败次数大于最大允许值,则状态机转移到 DISCONNECTED 状态以迫使 authPortStatus 值置为 Unauthorized。

7.3.6 密钥传输状态机

密钥传输状态机实现图 25 描述的功能。

如果以下所有条件均为真,密钥传输状态机将向对方发送 TAEPoL-Key PDUs:

- 端口没有正在进行初始化;
- portControl 值置为 Auto;
- 允许密钥传输;
- 有新的密钥材料要传送;
- 后台鉴别状态机置位 keyRun 指示密钥状态机可以运行。

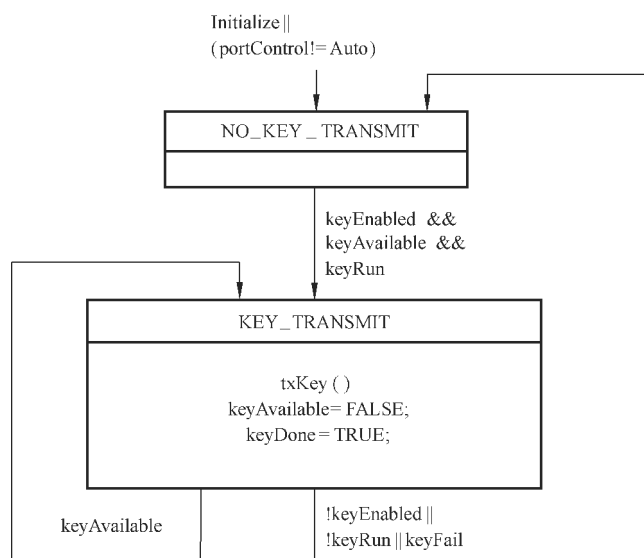


图 25 密钥传输状态机

密钥传输状态机使用以下过程传述密钥:

txKey(), 包含 TAEPoL-Key 的 TAEPoL 帧被发送。

7.3.7 密钥接收状态机

7.3.7.1 一般要求

密钥接收状态机实现图 26 描述的功能。

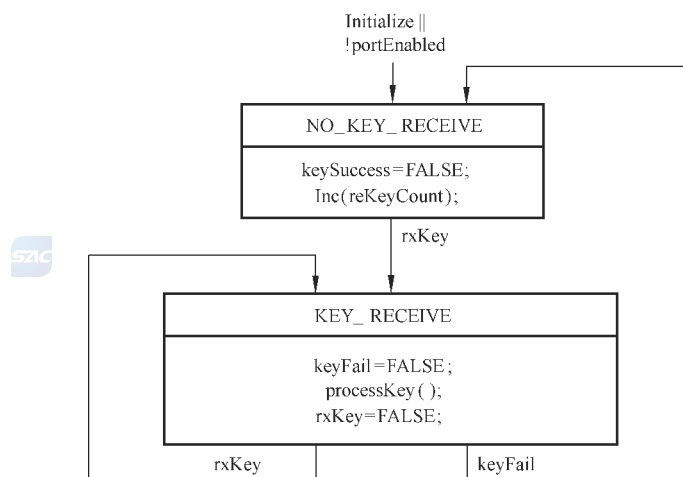


图 26 密钥接收状态机

密钥接收状态机允许 TAEPoL-Key PDU 被请求者或鉴别访问控制器接收,并根据相应的密码机制进行处理。

7.3.7.2 变量

密钥接收状态机的变量为 rxKey。如果请求者或鉴别访问控制器收到一个 TAEPoL-Key 帧,则该变量被设置为 TRUE。当密钥接收状态机对收到的密钥信息完成处理后,该变量被设置为 FALSE。

7.3.7.3 过程

processKey()。该过程处理收到的 TAEPoL-Key 帧中的密钥信息,并根据处理结果设置 keyFail 和 keySuccess 变量。

7.3.8 重鉴别计时器状态机

如果定期的重鉴别被激活(即, reAuthEnabled 值为 TRUE),则端口的重鉴别计时器状态机负责保证重鉴别定期进行。重鉴别计时器状态机实现图 27 描述的功能。

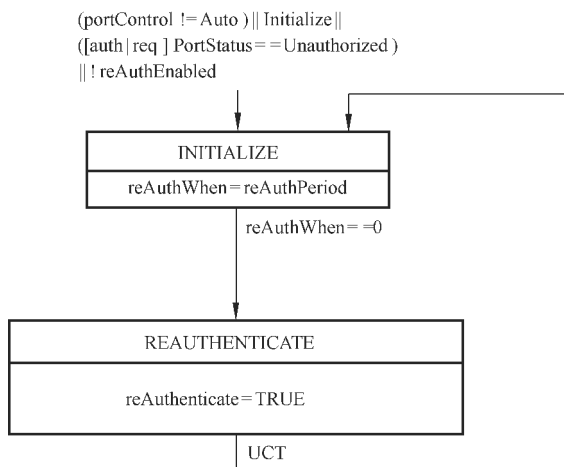


图 27 重鉴别计时器状态机

重鉴别计时器状态机使用以下常量：

——reAuthPeriod：该常量表示定期重鉴别间隔的秒数（非 0），默认值是 3 600 s。

——reAuthEnabled：该常量表示是否允许该端口定期进行重鉴别，TRUE 表示允许，FALSE 表示不允许。

7.3.9 鉴别访问控制器后台状态机

7.3.9.1 概述

鉴别访问控制器后台状态机具有以下状态：

- a) REQUEST；
- b) RESPONSE；
- c) SUCCESS；
- d) FAIL；
- e) TIMEOUT；
- f) IDLE；
- g) INITIALIZE；
- h) IGNORE。

鉴别访问控制器后台状态机实现图 28 描述的功能。

7.3.9.2 变量和过程

7.3.9.2.1 变量

鉴别访问控制器后台状态机使用以下变量：

——taepNoReq：上层不发送 TAEP 帧到请求者时，该变量被设置为 TRUE；

——taepReq：上层发送 TAEP 帧到请求者时，该变量被设置为 TRUE；

——taepResp：鉴别访问控制器状态机设置该变量为 TRUE 以通知上层有新的 TAEP 帧要处理，当上层已经得到该帧后，上层设置该变量为 FALSE。

7.3.9.2.2 常量

常量 serverPeriod 为 aWhile 计时器的初始值，默认值为 30s。

7.3.9.2.3 过程

鉴别访问控制器后台状态机使用以下过程：

- a) txReq(x)。上层需要发送 TAEP 分组时将封装该分组的 TAEPoL 帧发送给请求者。
- b) abortAuth。该过程允许鉴别访问控制器后台状态机在通知鉴别访问控制器状态机中止鉴别会话前，释放所有的系统资源。

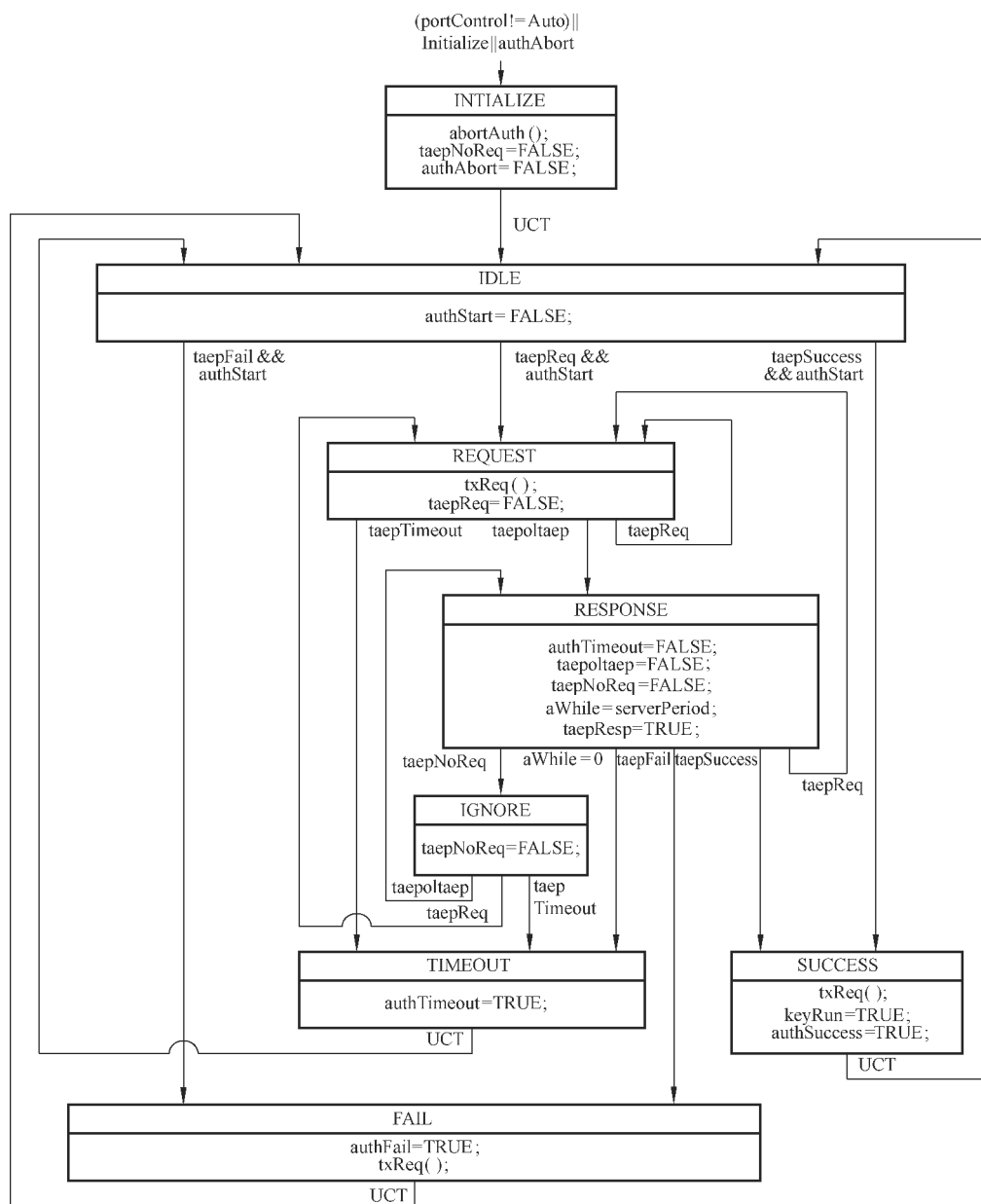


图 28 鉴别访问控制器后台状态机

7.3.9.3 鉴别访问控制器后台维护的计数器



7.3.9.3.1 概述

为便于诊断,鉴别访问控制器后台状态机对以下计数器进行维护。这些计数器的值可通过第 8 章给出的管理操作进行访问。如果这些计数器增长超过最大允许值,它们回转为 0。

7.3.9.3.2 backendAuthSuccesses

该参数表示状态机收到上层成功的指示(例如,taepSuccess=TRUE),从 RESPONSE 状态转移到 SUCCESS 状态的次数。

7.3.9.3.3 backendAuthFails

表示状态机收到上层失败的指示(例如,taepFail=TRUE),从RESPONSE状态转移到FAIL状态的次数。

7.3.9.4 REQUEST

在该状态下,状态机收到来自上层的TAEP-Request分组,并将该分组通过TAEPoL帧发送到请求者。若TAEP-Request分组丢失,TAEP将重传该分组,同时状态机将转回REQUEST状态。

如果从请求者处收到包含TAEP-Response分组的TAEPoL帧,状态机将转移到RESPONSE状态。

7.3.9.5 RESPONSE

在该状态下,状态机收到来自请求者封装在TAEPoL帧中的TAEP-Response分组,并将该分组传递给上层,然后等待上层的指令。aWhile变量用于判定上层响应是否超时,包含以下几种情况:

- 如果发生超时,状态机将转入TIMEOUT状态;
- 如果上层指示鉴别成功(taepSuccess),则状态机转入SUCCESS状态;
- 如果上层指示鉴别失败(taepFail),则状态机转入FAIL状态;
- 如果上层决定忽略收到的TAEP-Response分组,它将置位taepNoReq(将该位置为何值),则状态机转入IGNORE状态;
- 如果上层准备发送TAEP-Request分组,则状态机转入REQUEST状态。

7.3.9.6 SUCCESS

状态机设置全局变量authSuccess为TRUE,以通知鉴别访问控制器状态机鉴别会话已成功完成。如果KeyRun变量被设置为TRUE,状态机转入IDLE状态。

7.3.9.7 FAIL

状态机设置全局变量authFail为TRUE,以通知鉴别访问控制器状态机鉴别会话已失败,状态机将转入IDLE状态。

7.3.9.8 TIMEOUT

如果鉴别超时或上层通过taepTimeout通知超时,则状态机进入TIMEOUT状态。状态机设置全局变量authTimeout为TRUE,以通知鉴别访问控制器状态机鉴别会话已因超时而停止,状态机将转入IDLE状态。

7.3.9.9 IDLE

在该状态下,状态机等待鉴别访问控制器状态机通知是否开始新的鉴别会话。当authStart变成TRUE时,表示上层已经准备开始一个新的会话,状态机转入REQUEST状态。上层可以不进行鉴别会话而直接置位taepSuccess或taepFail,使状态机转入SUCCESS或FAIL状态。

7.3.9.10 INITIALIZE

如果鉴别访问控制器状态机设置全局变量authAbort为TRUE,或者系统进行初始化,则状态机将进入INITIALIZE状态。abortAuth过程用于释放系统资源,一旦变量initialize、taepNoReq和authAbort被设置为FALSE,则状态机转入IDLE状态。

7.3.9.11 IGNORE

如果上层决定忽略收到的 TAEP-Response 消息,则状态机将进入 IGNORE 状态。在这种情况下,上层置位 taepNoReq,指示没有相应于 TAEP-Response 消息的请求要发送。

当收到下一个 TAEP-Response 消息时,该状态将清除 TAEPNoReq 变量,转入 RESPONSE 状态。

7.3.10 受控方向状态机

7.3.10.1 一般要求

端口的受控方向状态机负责确保端口的 operControlledDirections 参数正确反映 adminControlledDirections 参数的当前状态、LAC 的操作状态以及是否有 Birdge 存在。如果给定 Birdge 的端口 OperControlledDirections 被设置为 IN,则允许 Bridge 在该端口上转发从其他端口收到的帧,但不允许转发和处理从该端口收到的帧。为了防止配置错误,通常将两个 Bridge 相连的端口都配置为 IN。如果端口的 operEdge 变量设置为 FALSE(见 IEEE Std 802.1D-2004 中的第 17 章),则 operControlledDirections 被强制为 BOTH。

受控方向状态机实现图 29 描述的功能(参见 IEEE Std 802.1x)。

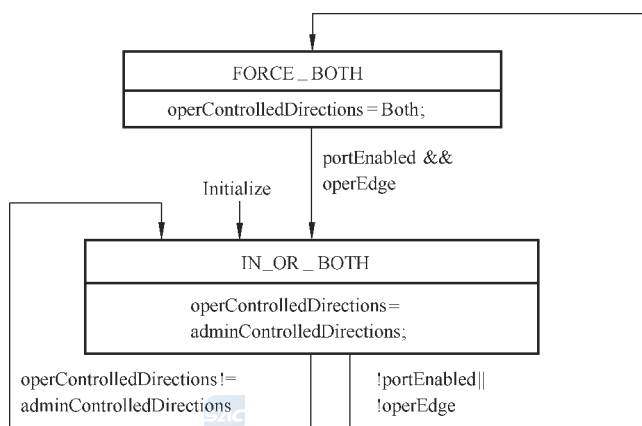


图 29 受控方向状态机

如果该状态机在 Bridge 端口上实现,则该端口支持 operControlledDirections 参数和 adminControlledDirections 参数值 IN 和 BOTH, Bridge 受控方向状态机也应在非受控端口上实现,以符合 IEEE Std 802.1D-2004 中的第 17 章。

状态机初始化时进入 IN_OR_BOTH 状态,同时 operControlledDirections 参数值设置为 adminControlledDirections 参数的值。如果端口的 LAC 为不可操作或端口为 Bridge 端口而非 Edge 端口,则状态机转入 FORCE_BOTH 状态。如果 operControlledDirections 参数和 adminControlledDirections 参数的值不一致,则状态机将重新进入 IN_OR_BOTH 状态。

处于 FORCE_BOTH 状态时,状态机 operControlledDirections 参数值设置为 BOTH。

如果进入 FORCE_BOTH 状态的条件不复存在,状态机将从 FORCE_BOTH 状态转入 IN_OR_BOTH 状态。

7.3.10.2 变量

受控方向状态机使用以下变量:

- adminControlledDirections: 值可为 IN 或 BOTH。该参数被状态机使用,但其值不能被状态机更改,其值只能被管理层改动。

- operControlledDirections:由状态机决定,值可为 IN 或 BOTH。
- operEdge:值由 Bridge 端口维护(见 IEEE Std 802.1D-2004 中的第 17 章),如果端口不是 Bridge 端口,则 operEdge 值为 TRUE。

7.3.11 请求者 PAE 状态机

7.3.11.1 概述

请求者 PAE 状态机有以下状态:

- a) LOGOFF;
- b) DISCONNECTED;
- c) CONNECTING;
- d) AUTHENTICATING;
- e) HELD;
- f) AUTHENTICATED;
- g) RESTART;
- h) REQ_FORCE_AUTH;
- i) REQ_FORCE_UNAUTH;
- j) AUTHORIZED。

请求者 PAE 状态机实现图 30 描述的功能。

7.3.11.2 变量、常量和过程

7.3.11.2.1 变量

请求者状态机使用以下变量:

- a) taepRestart:该变量被请求者状态机设置为 TRUE 以指明其正在重启状态机,并且有一个新的 TAEP-Request 帧需要 TAEP 处理。当上层准备好建立鉴别会话,该变量被设置为 FALSE。
- b) logoffSent:指示 PAE 状态机处于 LOGOFF 状态,TAEPoL-Logoff 消息已经被发送。
- c) reqPortMode:该变量和 PortControl 一起将请求者 PAE 状态机在 Auto 和 non-Auto 操作状态间切换,它可以取以下值:
 - 1) ForceUnauthorized:受控端口应保持 Unauthorized 状态;
 - 2) Auto:受控端口的状态根据请求者和鉴别访问控制器之间的鉴别交换结果设置为 Authorized 或 Unauthorized;
 - 3) ForceAuthorized:受控端口应保持 Authorized 状态。
- d) startCount:该变量用于统计已发送但没有收到响应的 TAEPoL-Start 消息的数量。
- e) userLogoff:该变量由外部控制,它反映了请求者系统中的用户登录/登出状态。如果请求者系统认为用户是登录状态,该变量设置为 FALSE;如果请求者系统认为用户是登出状态,该变量设置为 TRUE。
- f) traditionEnabled:该变量由管理层设置,当该变量为 TRUE 且鉴别访问控制器未响应时,请求者状态机将受控端口置为 Authorized 状态;当该变量为 FALSE 且鉴别访问控制器未响应时,请求者状态机将受控端口置为 Unauthorized 状态。

7.3.11.2.2 常量

请求者状态机使用以下常量:

- heldPeriod: 表示 heldWhile 计时器的初始化值, 其默认值为 60s;
- startPeriod: 表示 startWhen 计时器的初始化值, 其默认值为 30s;
- maxStart: 在请求者认为没有鉴别访问控制器存在前, 连续发送 TAEPoL-Start 消息的最大次数, 默认值为 3。

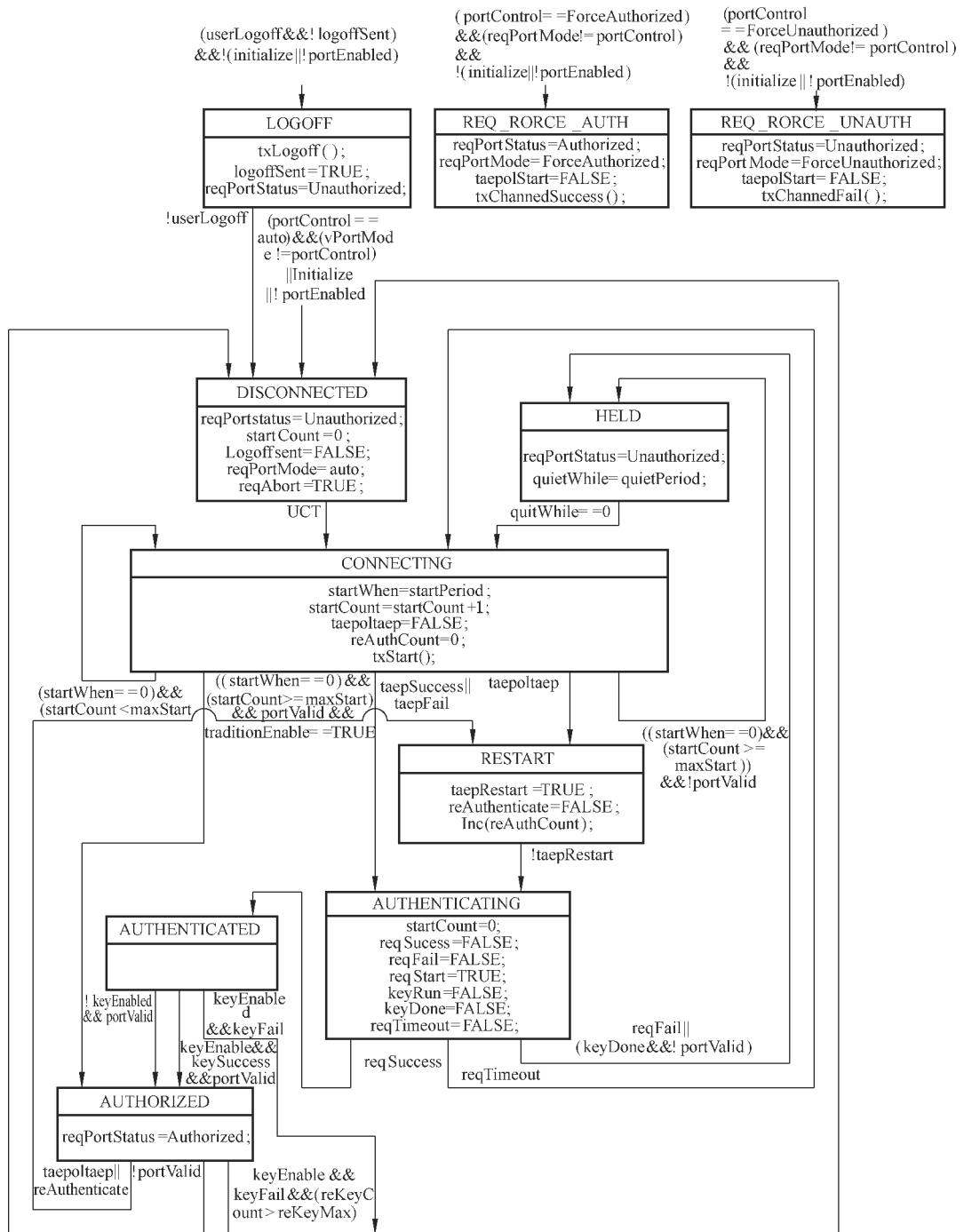


图 30 请求者 PAE 状态机

7.3.11.2.3 过程

请求者状态机使用以下过程：

- a) txStart(): 传送 TAEPoL-Start 帧给鉴别访问控制器；
- b) txLogoff(): 传送 TAEPoL-Logoff 给鉴别访问控制器。

7.3.11.3 LOGOFF

如果系统用户要求登出,则状态机进入该状态,同时向鉴别访问控制器发送 TAEPoL-Logoff 帧。如果 userLogoff 变量值置为 FALSE,则状态机进入 DISCONNECTED 状态。

7.3.11.4 DISCONNECTED

当端口的 LAC 服务不可用,或者系统被初始化,则状态机进入该状态。当系统的用户要求登录时,状态机将从 LOGOFF 状态转入该状态。如果 portValid 变量值为 FALSE,则状态机从 AUTHENTICATED 状态进入该状态。

如果状态机初始化完成且 LAC 服务可用,则其转入 CONNECTING 状态。

7.3.11.5 CONNECTING

在该状态下,状态机各端口可操作,请求者试图连接鉴别访问控制器。请求者发送 TAEPoL-Start 帧,启动 startWhen 计时器用于重传 TAEPoL-Start 帧。如果 startWhen 计时器超时,则 TAEPoL-Start 将被重传最多 maxStart 次。如果在 maxStart 次重传后,仍未收到响应,则状态机认为它已经关联到一个不识别 TAEPoL 帧的系统,若 portValid 为 TRUE,则状态机转入 AUTHORIZED 状态。请求者状态机在收到 TAEP-Request 帧后将转入 RESTART 状态。如果收到上层指示 taepSuccess 或 taepFail,请求者将直接转入 AUTHENTICATING 状态。

7.3.11.6 AUTHENTICATING

在该状态下,状态机已经收到来自鉴别访问控制器的 TAEP-Request 帧。reqStart 变量被置位,通知请求者后台状态机应处理来自鉴别访问控制器的请求。一旦请求者后台状态机完成工作,reqSuccess、reqFail 或 vTimeout 被设置为 TRUE,状态机将退出该状态。如果 reqSuccess 被为 TRUE,转移到 AUTHENTICATED 状态;如果 reqFail 或者 reqTimeout 是 TRUE,状态机分别转移到 HELD 或 CONNECTING 状态。

7.3.11.7 HELD

如果请求者后台状态机设置 reqFail 变量为 TRUE,则表面鉴别失败,状态机从 AUTHENTICATING 状态转入 HELD 状态。如果未连接任何鉴别访问控制器且 portValid 变量为 FALSE,则状态机从 CONNECTING 状态转入该状态。

heldWhile 计时器以 heldPeriod 为初始值启动,如果 heldWhile 超时,则状态机转入 CONNECTING 状态。如果收到鉴别访问控制器发送的请求帧,则状态机转入 RESTART 状态。

7.3.11.8 AUTHENTICATED

当请求者后台状态机把 reqSuccess 变量置位,请求者 PAE 状态机从 AUTHENTICATING 状态转入 AUTHENTICATED 状态。如果不进行密钥交换,则状态机转入 AUTHORIZED 状态;如果进行密钥交换,若密钥交换成功则转入 AUTHORIZED 状态,失败则转入 DISCONNECTED 状态。

7.3.11.9 RESTART

当请求者 PAE 需要通知上层其已经重启,则状态机进入 RESTART 状态。当在 AUTHENTICATED、CONNECTING 或者 HELD 状态收到 TAEP 分组,taepRestart 变量将设置为 TRUE 以通知上层请求者 PAE 已经重启。

当上层确认此次重启,其复位 taepRestart 变量为 FALSE,请求者 PAE 状态机从该状态退出,进入 AUTHENTICATING 状态。

7.3.11.10 REQ_FORCE_AUTH

如果同时符合以下 4 个条件,则任何其他状态都应转入 REQ_FORCE_AUTH 状态:

- portControl 变量被设置为 ForceAuthorized;
- reqPortMode 变量被设置为除 ForceAuthorized 以外的其他值;
- 端口的 LAC 可操作;
- 状态机未处于初始化状态。

reqPortStatus 被设置为 Authorized 时,reqPortMode 也将设置为 ForceAuthorized。

7.3.11.11 REQ_FORCE_UNAUTH

如果同时符合以下 4 个条件,则任何其他状态都应转入 REQ_FORCE_UNAUTH 状态:

- portControl 变量设置为 ForceUnauthorized;
- reqPortMode 变量设置为除 ForceUnauthorized 以外的其他值;
- 端口的 LAC 可操作;
- 状态机未处于初始化状态。

当 reqPortStatus 被设置为 Unauthorized,请求者发送 TAEPoL-Logoff 帧到鉴别访问控制器,reqPortMode 变量被设置为 ForceAuthorized。

这一系列动作的效果是强制端口到 Unauthorized 状态,并通过发送登出请求将该状态反映到鉴别访问控制器。

7.3.11.12 AUTHORIZED

当且仅当 portValid 为 TRUE,并同时满足下面三种情况之一时,才能进入 AUTHORIZED 状态:

- 请求者成功鉴别鉴别访问控制器;
- 请求者认为鉴别访问控制器不识别 TAEPoL 帧;
- 密钥交换成功。

如果请求者和鉴别访问控制器建立连接超时,则状态机从 CONNECTING 转入该状态;如果管理实体设置 keyEnabled 为 FALSE,则状态机从 AUTHENTICATED 状态直接转入该状态;如果管理实体设置 keyEnabled 为 TRUE,且密钥交换成功,则状态机从 AUTHENTICATED 状态转入该状态。

如果收到 TAEPoL/TAEP 帧或需要进行重新鉴别,则状态机转入 RESTART 状态。

如果 portValid 为 FALSE,状态机转入 DISCONNECTED 状态以迫使 reqPortStatus 成为 Unauthorized。

如果状态机在 AUTHORIZED 状态进行密钥更新过程,且密钥交换失败次数大于允许的最大值,则状态机转入 DISCONNECTED 状态以迫使 reqPortStatus 成为 Unauthorized。

7.3.12 请求者后台状态机

7.3.12.1 概述

请求者后台状态机有以下几种状态：

- a) REQUEST;
- b) RESPONSE;
- c) SUCCESS;
- d) FAIL;
- e) TIMEOUT;
- f) IDLE;
- g) INITIALIZE;
- h) RECEIVE。

请求者后台状态机用于实现图 31 描述的功能。

7.3.12.2 变量、常量和过程

7.3.12.2.1 变量

请求者后台状态机使用以下变量：

- taepNoResp:对于刚收到的 TAEP 帧,如果上层没有相应的 TAEP-Response,则该变量被上层置为 TRUE。当请求者状态机确认它已经看到该变量,则设置该变量为 FALSE。
- taepReq:当收到一个可由 TAEP 处理的 TAEP 帧时,请求者后台状态机设置该变量为 TRUE。当上层收到了 TAEP 帧,它设置该帧为 FALSE。
- taepResp:如果需要发送 TAEP 帧,则上层设置该变量为 TRUE。当 TAEP 帧发出后,请求者状态机设置该变量为 FALSE。

7.3.12.2.2 常量

常量 authPeriod 为 authWhile 计时器的初始值,默认值为 30 s。

7.3.12.2.3 过程

请求者后台状态机使用以下过程(见图 31):

- a) abortREQ():该过程允许请求者后台状态机在通知请求者状态机中止鉴别会话前,释放所有的系统资源;
- b) getREQrsp():该过程用于模拟请求者后台状态机从上层获得响应;
- c) txREQrsp():该过程用于将包含 TAEP 分组的 TAEPoL 帧发送给鉴别访问控制器。

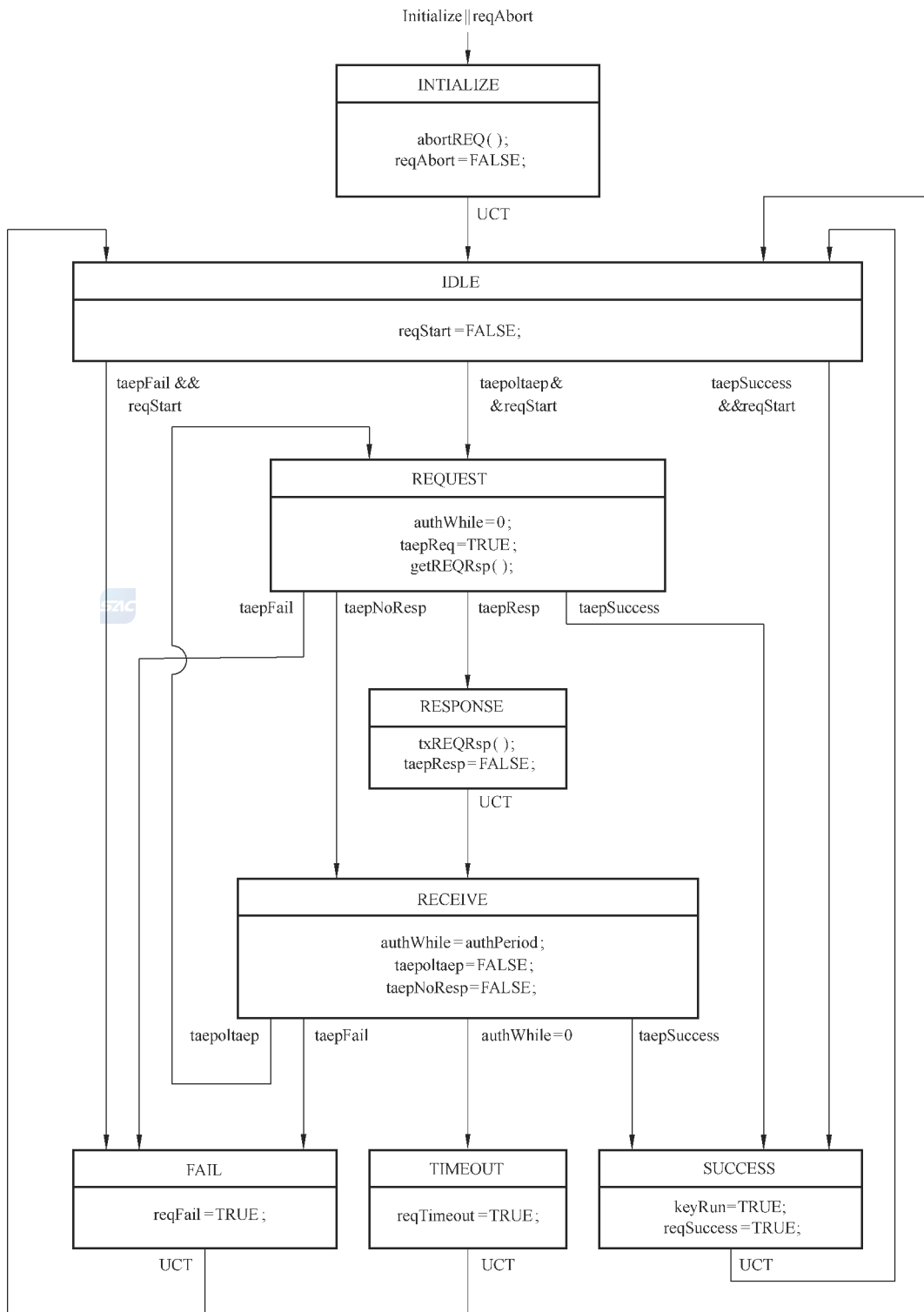


图 31 请求者后台状态机

7.3.12.3 REQUEST

在该状态下,状态机从鉴别访问控制器收到 TAEP-Request 分组,并触发 TAEP 处理。
 当 TAEP 表明其有 Response 分组需要发送时,状态机从 REQUEST 状态转入 RESPONSE 状态。

如果 TAEP 决定忽略鉴别访问控制器的请求,则状态机从 REQUEST 状态转入 RECEIVE 状态。如果 TAEP 决定鉴别失败(鉴别失败如何决定),状态机从 REQUEST 状态转入 FAIL 状态。如果 TAEP 决定鉴别成功,状态机从 REQUEST 状态转入 SUCCESS 状态。

7.3.12.4 RESPONSE

在该状态下,状态机发送 TAEP-Response 给鉴别访问控制器,然后无条件转入 RECEIVE 状态。

7.3.12.5 SUCCESS

在该状态下,状态机设置全局变量 reqSuccess 为 TRUE,通知请求者状态机鉴别会话已成功结束。同时,全局变量 keyRun 也被设置为 TRUE,指示请求者密钥传输状态机可以运行。然后,状态机转入 IDLE 状态。

7.3.12.6 FAIL

在该状态下,状态机设置全局变量 reqFail 为 TRUE,通知请求者状态机鉴别会话已经以失败结束。然后,状态机转入 IDLE 状态。

7.3.12.7 TIMEOUT

在该状态下,状态机设置全局变量 reqTimeout 为 TRUE,通知请求者状态机鉴别会话已因超时而结束。然后,状态机转入 IDLE 状态。

7.3.12.8 IDLE

在该状态下,状态机等待请求者状态机通知开始一个新的鉴别会话。如果 reqStart 为 TRUE,状态机转入 REQUEST 状态;如果 taepSuccess 为 TRUE,状态机转入 SUCCESS 状态;如果 taepFail 为 TRUE,状态机转入 FAIL 状态。

7.3.12.9 INITIALIZE

如果系统进行初始化,或请求者状态机设置全局变量 reqAbort 为 TRUE,则状态机进入该状态。abortREQ() 进程被调用,状态机释放系统资源。

一旦变量初始化和 reqAbort 都是 FALSE,则状态机转入 IDLE 状态。

7.3.12.10 RECEIVE

在该状态下,请求者等待来自鉴别访问控制器的下一个 TAEP-Request。为了防止鉴别访问控制器中止鉴别会话而使请求者挂死,将起用超时保护。

当发送 TAEP-Response 给鉴别访问控制器后,状态机从 RESPONSE 状态转入该状态。

如果 TAEP 决定丢弃前一个 TAEP-Request 帧,则状态机从 REQUEST 状态转入该状态。

7.3.13 自适应端口状态机

7.3.13.1 概述

自适应端口状态机有以下五种状态:

- a) DISCONNECT;
- b) CONNETCTING;
- c) DECIDING;

- d) REQ_STATUS;
- e) AAC_STATUS。

自适应端口状态机实现图 37 描述的功能。

7.3.13.2 过程

自适应端口状态机使用以下过程(见图 32):

- a) txstart():发送 TAEPoL-Start 帧到对方;
- b) decide():根据收到的 TAEPoL-Start 帧设置 systemRole 的值。

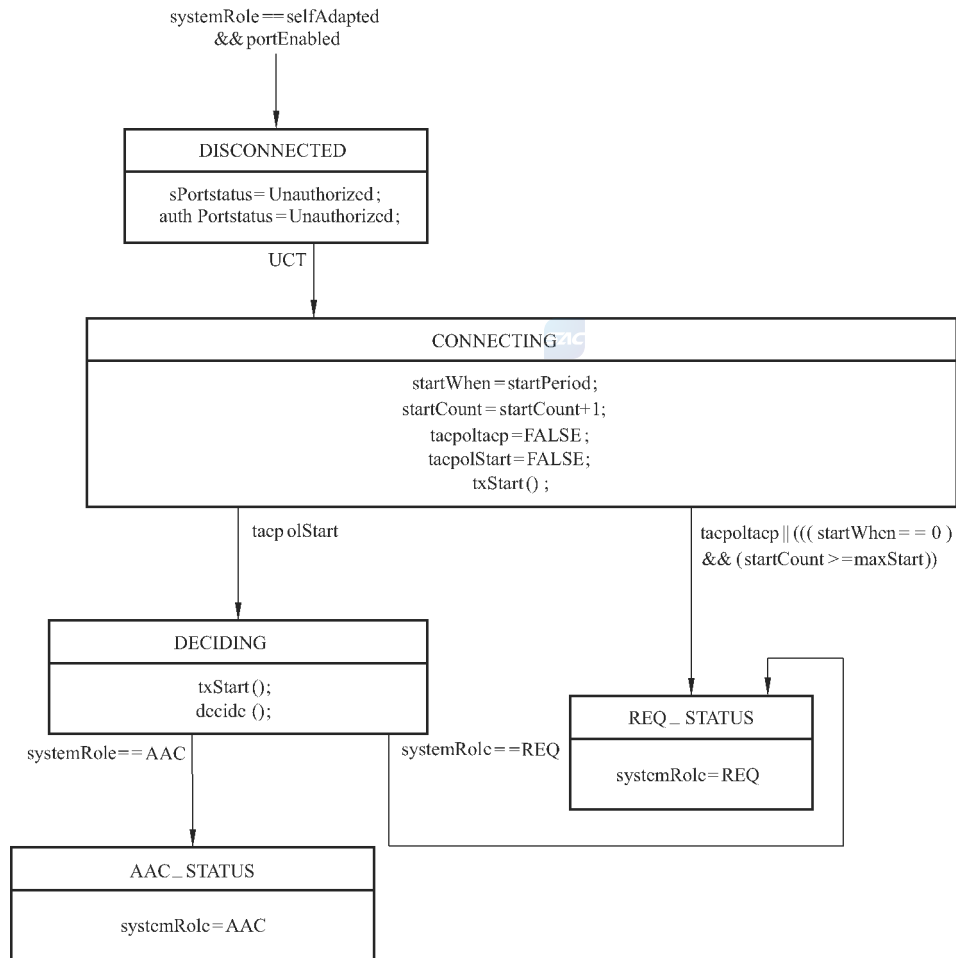


图 32 自适应端口状态机

7.3.13.3 DISCONNECT

当全局变量 systemRole 的值为 selfAdapted,且 LAC 服务可用时,自适应端口状态机进入该状态。当端口状态设置为 Unauthorized 时,状态机无条件转入 CONNECTING 状态。

7.3.13.4 CONNECTING

端口在该状态下可操作,PAE 试图决定自己的角色。PAE 发送 TAEPoL-Start 帧,同时启动 startWhen 计时器。如果 startWhen 计时器超时,TAEPoL-Start 将被重传最多 maxStart 次。如果在 maxStart 次重传后,仍未收到响应,状态机认为它已经关联到一个不识别 TAEPoL 帧的系统,则其转

入 REQ_STATUS 状态。如果收到响应为 TAEP-Request 帧,则请求者状态机也转入 REQ_STATUS 状态。如果收到 TAEPoL-Start 帧,则状态机转入 DECIDING 状态。

7.3.13.5 DECIDING

在该状态下,状态机发送 TAEPoL-Start 帧,然后调用 deciding() 进程决定自己的角色。如果 deciding() 进程设置 systemRole 为 AAC,则状态机转入 AAC_STATUS 状态。如果 deciding() 进程设置 systemRole 为 REQ,则状态机转入 REQ_STATUS 状态。

7.3.13.6 REQ_STATUS

在该状态下,状态机设置 systemRole 为 REQ,PAE 为请求者。

7.3.13.7 AAC_STATUS

在该状态下,设置 systemRole 为 AAC,PAE 为鉴别访问控制器。

8 端口接入控制管理

8.1 一般要求

本章定义了被管对象集合及其功能,允许对端口接入控制进行管理配置和监视,包含以下功能:

- 介绍管理功能,帮助确定支持管理功能对端口接入控制的需求;
- 建立用于模仿端口接入控制及其管理对象操作的状态机之间的通信;
- 规定每个被管对象支持的管理操作。

本章规定的管理功能关系到对 PCAP 协议的控制和监视以及对会话参数的监视,不包括鉴别访问控制器和鉴别服务器之间通信可能使用的任何协议的管理规范。假定被选协议有关的规范将包括适当的管理功能定义。

8.2 管理功能

8.2.1 概述

管理功能描述用户必须实现的功能,并说明其用途。这些功能支持通信资源的计划、组织、监督、控制、保护及安全,可被分类为配置、故障、性能、安全及计费管理等功能域。8.2.2~8.2.6 分别对每个功能域作介绍,并同时介绍通信资源管理通常所需的功能以及端口接入控制管理在对应功能域上所提供的特殊功能。

8.2.2 配置管理

配置管理功能用于对对象提供通信资源的识别、初始化、复位与关闭、操作参数的提供以及资源间关系的建立与发现。端口接入控制管理在该功能域上提供的功能如下:

- 为鉴别访问控制器配置操作参数(见 8.5.2.2 和 8.5.2.3);
- 为请求者配置操作参数(见 8.6.2.2 和 8.6.2.3);
- 为系统配置操作参数(见 8.7.2.3);
- 为端口初始化状态机(见 8.7.2.4)。

8.2.3 故障管理

故障管理功能提供故障的预防、检测、诊断及修复。当被管设备出现异常后,设备能够自动告警。

端口接入控制管理在该功能域上提供的功能如下：

- a) 提取鉴别访问控制器的统计信息(见 8.5.2.2)；
- b) 提取请求者的统计信息(见 8.6.2.2)；
- c) 为鉴别访问控制器配置操作参数(见 8.5.2.2 和 8.5.2.3)；
- d) 为请求者配置操作参数(见 8.6.2.2 和 8.6.2.3)；
- e) 为系统配置操作参数(见 8.7.2.3)；
- f) 通过 SNMP trap 发送告警。

8.2.4 性能管理



性能管理功能用于提供对通信资源的行为和通信动作的效力的评估。端口接入控制管理在该功能域上提供的功能如下：

- a) 提取鉴别访问控制器的统计信息(见 8.5.2.2)；
- b) 提取请求者的统计信息(见 8.6.2.2)；
- c) 为鉴别访问控制器配置操作参数(见 8.5.2.2 和 8.5.2.3)；
- d) 为请求者配置操作参数(见 8.6.2.2 和 8.6.2.3)。

8.2.5 安全管理

安全管理功能用于提供对资源的保护。端口接入控制管理在该功能域上提供的功能如下：

- a) 为鉴别访问控制器配置操作参数(见 8.5.2.2 和 8.5.2.3)；
- b) 为请求者配置操作参数(见 8.6.2.2 和 8.6.2.3)；
- c) 强迫重新鉴别(见 8.5.2.4)。

8.2.6 计费管理

计费管理功能用于提供费用的识别与分发以及费用的设置。端口接入控制管理在该功能域上提供的功能如下：

提取会话计费统计(见 8.5.5)。

8.3 被管对象

对被管对象的操作提供关于该被管对象有关的过程或实体的信息,或者实现该被管对象有关的过程或实体的控制。端口接入控制的管理根据支持端口控制的单个端口有关被管资源来描述,端口的被管资源就是 7.3 中建立的那些过程和实体,特别是：

- a) 支持鉴别访问控制器 PAE 操作的状态机(见 7.3.4、7.3.5、7.3.8 和 7.3.9)。与这些资源有关的被管对象和操作在 8.5 中定义。
- b) 支持 REQ PAE 操作的状态机(见 7.3.4、7.3.8、7.3.11 和 7.3.12)。与这些资源有关的被管对象和操作在 8.6 中定义。

除此之外,对于单个鉴别访问控制器 PAE 或请求者 PAE 的操作而言,某些被管资源并不明确,因此它们将作为系统及其端口的部分管理能力进行描述。与这些资源有关的被管对象和操作在 8.7 中定义,这些资源将根据 8.5、8.6 和 8.7 中定义的被管对象和操作描述。

注：本章规定的值作为管理操作的输入和输出,属于抽象信息元素。格式或编码问题与传输或表示该信息的特定协议有关。

8.4 数据类型

以下规定了管理协议中与编码无关的操作的语义。操作参数的数据类型只能定义成符合规范的数

据类型。使用的数据类型如下：

- a) 布尔类型(Boolean)；
- b) 枚举类型(Enumerated),用于指定值的搜集；
- c) 无符号类型(Unsigned),用于定义为某些量个数的所有参数；
- d) MAC 地址类型(MAC 地址)；
- e) 时间间隔类型(Time Interval),一个表示正整数秒的 Unsigned 类型值,用于所有的 TAEPOL 协议超时参数；
- f) 计数器类型(Counter),用于定义为某些量计数的所有参数(计数器增加并取 264 模值)。

8.5 鉴别访问控制器 PAE 被管对象

8.5.1 一般要求

鉴别访问控制器 PAE 和支持其操作的状态机在 7.3.4、7.3.5、7.3.8 和 7.3.9 中描述。由这些被管资源组成的对象如下：

- a) 鉴别访问控制器配置被管对象(见 8.5.2)；
- b) 鉴别访问控制器统计被管对象(见 8.5.3)；
- c) 鉴别访问控制器诊断被管对象(见 8.5.4)；
- d) 鉴别访问控制器会话统计被管对象(见 8.5.5)。

支持鉴别访问控制器功能的端口应支持鉴别访问控制器配置被管对象定义的管理功能,并支持鉴别访问控制器统计、诊断、会话统计等对被管对象定义的管理功能。提供这种管理功能的手段(例如,支持的管理协议)应在与实现有关的 PICS 中规定。

8.5.2 鉴别访问控制器配置

8.5.2.1 一般要求

鉴别访问控制器配置被管对象模型化鉴别访问控制器资源配置的修改或查询操作。每个支持鉴别访问控制器功能的端口均有一个鉴别访问控制器配置被管对象。对鉴别访问控制器配置被管对象执行的管理操作有：

- a) 读取鉴别访问控制器配置(见 8.5.2.2)；
- b) 设置鉴别访问控制器配置(见 8.5.2.3)；
- c) 重新鉴别(见 8.5.2.4)。

8.5.2.2 读取鉴别访问控制器配置

8.5.2.2.1 目的

请求关于端口有关的鉴别访问控制器的配置信息。

8.5.2.2.2 输入

端口号:端口驻留的系统为该端口分配的识别号。

所分配的端口号不需连续,也就是意味着某些端口号为虚拟项,没有真正的 LAN 端口(例如,未来通过增加 MAC 接口允许系统扩展)。这样的虚拟端口应以与永久禁止端口所关联的 MAC 一致的方式支持管理操作。

注:当 MAC 桥(见 IEEE Std 802.1D-2004)实现端口接入控制时,很容易使端口接入控制管理使用的端口号与由桥分配的端口号相同。但这并不总是可能的,例如,当还实现了 GB/T 15629.3 的链路会聚时,端口接入控制工作在物理端口上,MAC 桥却利用会聚端口。

8.5.2.2.3 输出

读取鉴别访问控制器配置的输出如下：

- a) 端口号。端口驻留的系统为该端口分配的识别号。
- b) 鉴别访问控制器 PAE 状态。鉴别访问控制器 PAE 状态机的当前状态(见 7.3.5),该参数能取以下值：
 - 1) INITIALIZE;
 - 2) DISCONNECTED;
 - 3) CONNECTING;
 - 4) AUTHENTICATING;
 - 5) AUTHENTICATED;
 - 6) ABORTING;
 - 7) HELD;
 - 8) FORCE_AUTH;
 - 9) FORCE_UNAUTH;
 - 10) RESTART;
 - 11) AUTHORIZED。
- c) 鉴别访问控制器后台状态。鉴别访问控制器后台状态机的当前状态(见 7.3.9),该参数能取以下值：
 - 1) REQUEST;
 - 2) RESPONSE;
 - 3) SUCCESS;
 - 4) FAIL;
 - 5) TIMEOUT;
 - 6) IDLE;
 - 7) INITIALIZE;
 - 8) IGNORE。
- d) AdminControlledDirections。端口有关的 AdminControlledDirections 参数的当前值(见 5.3),该参数能取以下值：
 - 1) Both;
 - 2) In。
- e) OperControlledDirections。端口有关的 OperControlledDirections 参数的当前值(见 5.3),该参数能取以下值：
 - 1) Both;
 - 2) In。
- f) AuthControlledPortControl。端口有关的 AuthControlledPortControl 参数的当前值(见 5.3),该参数能取以下值：
 - 1) ForceAuthorized;
 - 2) ForceUnauthorized;
 - 3) Auto。
- g) AuthControlledPortStatus。端口有关的 AuthControlledPortStatus 参数的当前值(见 5.3),该参数能取以下值：
 - 1) Authorized;

- 2) Unauthorized。
- h) quietPeriod。鉴别访问控制器 PAE 状态机目前使用的常量 quietPeriod 的值(见 7.3.5.2.2)。
- i) reAuthPeriod。重鉴别计时器状态机目前使用的常量 reAuthPeriod 的值(见 7.3.8)。
- j) reAuthEnabled。重鉴别计时器状态机使用的使能/禁止控制(见 7.3.8)。
- k) KeyEnabled。若密钥信息使能,为 TRUE;若禁止,为 FALSE(见 7.3.3.2)。

8.5.2.3 设置鉴别访问控制器配置

8.5.2.3.1 目的

配置控制与端口有关的鉴别访问控制器操作的参数。

8.5.2.3.2 输入

任何标为可选的参数可从操作中省略,以允许配置参数子集的选择性修改。但是,实现应支持包含下列所有参数的能力:

- a) 端口号。端口驻留的系统为该端口分配的识别号。
- b) AdminControlledDirections (可选的)。分配给端口有关的 AdminControlledDirections 参数的新值(见 5.3),该参数能取以下值:
 - 1) Both;
 - 2) In。
- c) AuthControlledPortControl (可选的)。分配给端口有关的 AuthControlledPortControl 参数的新值(见 5.3),该参数能取以下值:
 - 1) ForceAuthorized;
 - 2) ForceUnauthorized;
 - 3) Auto。
- d) quietPeriod (可选的)。分配给鉴别访问控制器 PAE 状态机的常量 quietPeriod 的新值(见 7.3.5.2.2)。
- e) taepTimeout (可选的)。新值,单位为秒(s),用于确定等待请求者响应 TAEP Request 的时间长短(见 7.3.3.2)。
- f) reAuthPeriod (可选的)。分配给重鉴别计时器状态机的常量 reAuthPeriod 的新值(见 7.3.8)。
- g) reAuthEnabled (可选的)。分配给重鉴别计时器状态机的常量 reAuthEnabled 的新值(见 7.3.8)。
- h) KeyEnabled (可选的)。分配给 KeyEnabled 参数的新值(见 7.3.3.2)。

8.5.2.3.3 输出

无。

8.5.2.4 重新鉴别

8.5.2.4.1 目的

引起端口的鉴别访问控制器 PAE 状态机重新鉴别请求者。若已经有一个正在进行的鉴别过程,该重新鉴别不会发生直到当前的鉴别完成(成功或不成功)。

8.5.2.4.2 输入

端口号:端口驻留的系统为该端口分配的识别号。

8.5.2.4.3 输出

无。

8.5.2.4.4 效果

该操作引起端口的鉴别访问控制器 PAE 状态机的重新鉴别变量(见 7.3.3.2)被设置为 TRUE。

8.5.3 鉴别访问控制器统计值

8.5.3.1 概述

鉴别访问控制器统计值管理对象模型化修改、查询关联于鉴别访问控制器操作的统计值的操作。对于每个支持鉴别访问控制器功能的端口,都有一个鉴别访问控制器统计值管理对象。能够对该对象进行的管理操作如下:

读取鉴别访问控制器统计值(见 8.5.3.2)。

8.5.3.2 读鉴别访问控制器统计值

8.5.3.2.1 目的

请求关于鉴别访问控制器操作的统计信息。

8.5.3.2.2 输入

端口号。端口驻留的系统为该端口分配的识别号。

8.5.3.2.3 输出

读取鉴别访问控制器统计值的输出如下:

- a) 端口号。端口驻留的系统为该端口分配的识别号。
- b) 收到的 TAEPoL 帧。鉴别访问控制器收到的有效的 TAEPoL 帧,包括所有类型的数量。
- c) 发送的 TAEPoL 帧。鉴别访问控制器发送的 TAEPoL 帧,包括所有类型的数量。
- d) 收到的 TAEPoL-Start 帧。鉴别访问控制器收到的 TAEPoL-Start 帧的数量。
- e) 收到的 TAEPoL-Logoff 帧。鉴别访问控制器收到的 TAEPoL-Logoff 帧的数量。
- f) 收到的 TAEP-Resp/Id 帧。鉴别访问控制器收到的有效的 TAEP-Resp/Id 帧的数量。
- g) 收到的 TAEP-Response 帧。鉴别访问控制器收到的有效的 TAEP-Response 帧(除过 Resp/Id 帧)的数量。
- h) 发送的 TAEP-Initial Request 帧。鉴别访问控制器发送的 TAEP 初始请求帧的数量。
- i) 发送的 TAEP-Request 帧。鉴别访问控制器发送的有效的 TAEP-Request 帧的数量(除过初始的请求帧)。
- j) 收到的无效 TAEPoL 帧。鉴别访问控制器收到的类型字段无法识别的 TAEPoL 帧的数量。
- k) 收到的长度错误 TAEPoL 帧。鉴别访问控制器收到的长度字段无效的 TAEPoL 帧的数量。
- l) 最后的 TAEPoL 帧版本。在最近收到的 TAEPoL 帧中的版本号。
- m) 最后的 TAEPoL 帧源。在最近收到的 TAEPoL 帧中的源 MAC 地址。

8.5.4 鉴别访问控制器诊断

8.5.4.1 概述

鉴别访问控制器诊断管理对象模型化修改、查询关联于鉴别访问控制器操作的诊断信息的操作。

对于每个支持鉴别访问控制器功能的端口,都有一个鉴别访问控制器诊断管理对象。能够对该对象进行读取鉴别访问控制器诊断值的管理操作。

8.5.4.2 读取鉴别访问控制器诊断值

8.5.4.2.1 目的

请求关于鉴别访问控制器操作的诊断信息。

8.5.4.2.2 输入

端口号:端口驻留的系统为该端口分配的识别号。

8.5.4.2.3 输出

读取鉴别访问控制器诊断值的输出如下:

- a) 端口号,端口驻留的系统为该端口分配的识别号;
- b) authTEntersConnecting (定义见 7.3.5.3.2);
- c) authTaepLogoffsWhileConnecting (定义见 7.3.5.3.3);
- d) authEntersAuthenticating (定义见 7.3.5.3.4);
- e) authAuthSuccessWhileAuthenticating (定义见 7.3.5.3.5);
- f) authAuthTimeoutsWhileAuthenticating (定义见 7.3.5.3.6);
- g) authAuthFailWhileAuthenticating (定义见 7.3.5.3.7);
- h) authAuthTaepStartsWhileAuthenticating (定义见 7.3.5.3.8);
- i) authAuthTaepLogoffWhileAuthenticating (定义见 7.3.5.3.9);
- j) authAuthReauthsWhileAuthenticated (定义见 7.3.5.3.10);
- k) authAuthTaepStartsWhileAuthenticated (定义见 7.3.5.3.11);
- l) authAuthTaepLogoffWhileAuthenticated (定义见 7.3.5.3.12);
- m) backendAuthSuccesses (定义见 7.3.9.3.2);
- n) backendAuthFails (定义见 7.3.9.3.3)。

8.5.5 鉴别访问控制器会话统计

8.5.5.1 概述

鉴别访问控制器会话统计管理对象模型化修改、查询关联于一个会话的统计值的操作。对于每个支持鉴别访问控制器功能的端口,都有一个鉴别访问控制器会话统计对象。该管理对象记录了当前会话的统计值(如果存在一个激活的会话)或者前一个会话的统计值(如果当前没有激活的会话),可对该对象进行读取鉴别访问控制器会话统计的管理操作。

在会话期间(例如,在端口处于授权状态期间)对关联于每个端口的会话统计进行维护。当鉴别访问控制器 PAE 状态机的 portStatus 变量从 Unauthorized 转变到 Authorized,统计参数被设置为 0 以进行初始化。当 portStatus 保持在 Authorized,会话统计根据它们各自的参数定义更新。当 portStatus 成为 Unauthorized,会话统计值被冻结,不再更新。

8.5.5.2 读取鉴别访问控制器会话统计

8.5.5.2.1 目的

请求关于鉴别访问控制器操作的诊断信息。

8.5.5.2.2 输入

端口号:端口驻留的系统为该端口分配的识别号。

8.5.5.2.3 输出

读取鉴别访问控制器会话统计的输出如下:

- a) 端口号。端口驻留的系统为该端口分配的识别号。
- b) 收到的会话八位位组。在会话期间从端口上收到的用户数据帧中的八位位组数。
- c) 发送的会话八位位组。在会话期间从端口上发送的用户数据帧中的八位位组数。
- d) 收到的会话帧。在会话期间从端口上收到的用户数据帧数。
- e) 发送的会话帧。在会话期间从端口上发送的用户数据帧数。
- f) 会话标识。会话的标识符,对于该鉴别访问控制器是唯一的。该标识是可打印的字符串,至少为3个字符。
- g) 会话鉴别方法。用于建立该会话的鉴别方法,该参数可以取下面的值:
 - 1) Server In。鉴别方法需要服务器。
 - 2) Server empty。鉴别方法不需要服务器。
- h) 会话时间。会话持续的时间,以秒(s)计。
- i) 会话终止原因。会话终止的原因,该参数可以取下面的值:
 - 1) 请求者 Logoff;
 - 2) 端口失败;
 - 3) 请求者重启动;
 - 4) 重鉴别失败;
 - 5) AuthControlledPortControl 设置为 ForceUnauthorized;
 - 6) 端口重初始化;
 - 7) 端口管理性的禁止;
 - 8) 其他未明原因。
- j) 会话用户名。代表请求者 PAE 身份的用户名。

8.6 请求者 PAE 管理对象

8.6.1 概述

请求者 PAE 和支持其操作的状态机在 7.3.4、7.3.11、7.3.12 中描述,包含这些管理资源的管理对象如下:

- 请求者配置管理对象;
- 请求者统计管理对象。

支持请求者功能的端口要支持由请求者配置管理对象定义的管理功能,也许支持由请求者统计管理对象定义的管理功能。

这些管理功能提供的手段将在 PICS 中规定。

8.6.2 请求者配置

8.6.2.1 一般要求

请求者配置被管对象模型化请求者资源配置的修改或查询操作。每个支持请求者功能的端口均有一个请求者配置被管对象。对请求者配置被管对象执行的管理操作有:

- 读取请求者状态；
- 设置请求者配置。

8.6.2.2 读取请求者状态

8.6.2.2.1 目的

请求有关请求者的配置信息。

8.6.2.2.2 输入

端口号：端口驻留的系统为该端口分配的识别号。

所分配的端口号不需连续，也就是意味着某些端口号为虚拟项，没有真正的 LAN 端口（例如，未来通过增加 MAC 接口允许系统扩展）。这样的虚拟端口应以与永久禁止端口所关联的 MAC 一致的方式支持管理操作。

注：当 MAC 桥（见 IEEE Std 802.1D-2004）实现端口接入控制时，端口接入控制管理使用的端口号与由桥分配的端口号相同。

8.6.2.2.3 输出

读取请求者状态的输出如下：

- a) 端口号。端口驻留的系统为该端口分配的识别号。
- b) 请求者 PAE 状态。请求者 PAE 状态机当前的状态（见 7.3.11），可以有以下参数：
 - 1) DISCONNECTED；
 - 2) LOGOFF；
 - 3) CONNECTING；
 - 4) AUTHENTICATING；
 - 5) AUTHENTICATED；
 - 6) HELD；
 - 7) RESTART；
 - 8) REQ_FORCE_AUTH；
 - 9) REQ_FORCE_UNAUTH；
 - 10) AUTHORIZED。
- c) heldPeriod。当前用在请求者 PAE 状态机的 heldPeriod 常量的值（见 7.3.11.2.2）。
- d) authPeriod。当前用在请求者 PAE 状态机的 authPeriod 常量的值（见 7.3.12.2.2）。
- e) startPeriod。当前用在请求者 PAE 状态机的 startPeriod 常量的值（见 7.3.11.2.2）。
- f) maxStart。当前用在请求者 PAE 状态机的 maxStar 常量的值（见 7.3.11.2.2）。
- g) reqControlledPortStatus。该参数直接反映了由请求者 PAE 状态机维护的 portStatus 变量的值，它可以取下面的值：
 - 1) Authorized；
 - 2) Unauthorized。
- h) 请求者后台状态。请求者后台状态机的当前状态，该参数可以有以下值：
 - 1) INITIALIZE；
 - 2) IDLE；
 - 3) REQUEST；
 - 4) RESPONSE；

- 5) RECEIVE;
- 6) FAIL;
- 7) SUCCESS;
- 8) TIMEOUT。

8.6.2.3 设置请求者配置

8.6.2.3.1 目的

配置控制请求者操作的参数。

8.6.2.3.2 输入

任何被标为 optional 的参数可以在操作中省略,以允许选择性的修改配置参数的子集。具体实现要具有包含以下所有参数的能力:

- a) 端口号:端口驻留的系统为该端口分配的识别号;
- b) heldPeriod(optional):分配给请求者 PAE 状态机 heldPeriod 常量的新值;
- c) authPeriod(optional):分配给请求者 PAE 状态机 authPeriod 常量的新值;
- d) startPeriod(optional):分配给请求者 PAE 状态机 startPeriod 常量的新值;
- e) maxStart(optional):分配给请求者 PAE 状态机 maxStart 常量的新值。

8.6.2.3.3 输出

空。

8.6.3 请求者统计

8.6.3.1 一般要求

请求者统计值管理对象模型化修改、查询关联于请求者操作的统计值的操作。对于每个支持请求者功能的端口,都有一个请求者统计值管理对象。能够对该对象进行读取请求者统计值的管理操作(见 8.6.3.2)。

8.6.3.2 读取请求者统计

8.6.3.2.1 目的

请求关于请求者操作的统计信息。

8.6.3.2.2 输入

端口号。端口驻留的系统为该端口分配的识别号。

8.6.3.2.3 输出

读取请求者统计的输出如下:

- a) 端口号。端口驻留的系统为该端口分配的识别号。
- b) 收到的 TAEPoL 帧。请求者收到的有效的 TAEPoL 帧,包括所有类型的数量。
- c) 发送的 TAEPoL 帧。请求者发送的 TAEPoL 帧,包括所有类型的数量。
- d) 发送的 TAEPoL-Start 帧。请求者发送的 TAEPoL-Start 帧的数量。
- e) 发送的 TAEPoL-Logoff 帧。请求者发送的 TAEPoL-Logoff 帧的数量。

- f) 发送的 TAEP-Resp/Id 帧。请求者发送的有效的 TAEP-Resp/Id 帧的数量。
- g) 发送的 TAEP-Response 帧。请求者发送的有效的 TAEP-Response 帧(除过 Resp/Id 帧)的数量。
- h) 收到的 TAEP-Req/Id 帧。请求者收到的有效的 TAEP-Req/Id 帧的数量。收到的 TAEP-Request 帧。
- i) 收到的 TAEP-Request 帧。请求者收到的 TAEP 请求帧(除过 Req/Id 帧)的数量。
- j) 收到的无效 TAEPoL 帧。请求者收到的类型字段无法识别的 TAEPoL 帧的数量。
- k) 收到的长度错误 TAEPoL 帧。请求者收到的长度字段无效的 TAEPoL 帧的数量。
- l) 最后的 TAEPoL 帧版本。在最近收到的 TAEPoL 帧中的版本号。
- m) 最后的 TAEPoL 帧源。在最近收到的 TAEPoL 帧中的源 MAC 地址。

8.7 系统管理对象

8.7.1 概述

包含管理资源的对象有系统配置管理对象。

支持 PAE 功能的端口,不论采用请求者或者鉴别访问控制器的角色,要支持由系统配置管理对象定义的管理功能。这些管理功能提供的手段将在 PICS 中规定。

8.7.2 系统配置

8.7.2.1 一般要求

系统配置被管对象模型化系统资源配置的修改或查询操作。每个支持端口访问控制功能的系统均有一个系统配置管理对象。对系统配置被管对象执行的管理操作有:

- 读取系统配置;
- 设置系统配置;
- 初始化。

8.7.2.2 读取系统配置

8.7.2.2.1 目的

读取同系统相关的配置信息

8.7.2.2.2 输入

空。

8.7.2.2.3 输出

读取系统配置的输出如下:

- a) SystemAuthControl(Optional)。系统的 SystemAuthControl 参数值,该参数仅存在于支持鉴别访问控制器功能的系统。
- b) 对于系统的每一个端口:
 - 1) 端口号。端口驻留的系统为该端口分配的识别号。
 - 2) 协议版本。端口支持的 TAEPoL 实现的协议版本号。
 - 3) PAE 能力。端口 PAE 的能力,该参数指示了端口 PAE 支持鉴别访问控制器、请求者或自适应功能。

- 4) PAE 优先级。端口 PAE 的优先级。0 保留给请求者,255 保留给鉴别访问控制器,自适应优先级为 1~254。
- 5) 安全 Hello 能力。指示端口 PAE 是否支持安全 Hello。

8.7.2.3 设置系统配置

8.7.2.3.1 目的

设置系统相关的配置信息。

8.7.2.3.2 输入

SystemAuthControl。要配置给系统的 SystemAuthControl 参数的值,参数可以取 Enabled 或 Disabled。

8.7.2.3.3 输出

空。

8.7.2.4 初始化端口

8.7.2.4.1 目的

使端口的 TAEPoL 状态机初始化。

8.7.2.4.2 输入

端口号。端口驻留的系统为该端口分配的识别号。

8.7.2.4.3 输出

空。

8.7.2.4.4 效果

该操作导致端口的初始化全局变量在很短的一段时间内被设置为 TRUE,然后设置为 FALSE。

注：“很短的一段时间”要足够长以使端口的所有状态机能够识别状态的变化来进行全局的转移。这个时间段是实现相关的。

9 端口接入控制 MIB 定义

在下面的 MIB 定义中,若 DESCRIPTION 文本与第 8 章中对应的定义出现偏差时,应优先采用第 8 章中的定义。

```
GBT28455-PAE-MIB DEFINITIONS ::= BEGIN
-----
-- GB/T28455 MIB
-----

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Counter32, Counter64,
    Unsigned32, TimeTicks
FROM SNMPv2-SMI
    MacAddress, TEXTUAL-CONVENTION, TruthValue
```


FROM SNMPv2-TC
 MODULE-COMPLIANCE, OBJECT-GROUP
 FROM SNMPv2-CONF
 SnmpAdminString
 FROM SNMP-FRAMEWORK-MIB
 InterfaceIndex
 FROM IF-MIB
 ;

gbt28455paeMIB 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：86-29-87607832
 传真：86-29-87607829
 E-mail：bwips@chinabwips.org
 Address：P. O. BOX 88, Xi'an High-Tech Industrial Development Zone, China
 Postcode：710075
 Tel：+86 29 87607832
 Fax：+86 29 87607829”
 DESCRIPTION
 “管理本文件的端口接入控制模块。”
 DESCRIPTION
 ::= { iso(1) member-body(2) cn(156) bwips(11235) gbt28455(28455)
 gbt28455-mibs(1) 1 }

paeMIBObjects OBJECT IDENTIFIER ::= { gbt28455paeMIB 1 }

 -- 文本约定

PaeControlledDirections ::= TEXTUAL-CONVENTION

STATUS mandatory

DESCRIPTION

"用于 AAC PAE 的控制模式值。"

SYNTAX INTEGER { both(0), in(1) }

PaeControlledPortStatus ::= TEXTUAL-CONVENTION

STATUS mandatory

DESCRIPTION

"AAC PAE 受控端口的状态值。"

SYNTAX INTEGER {authorized(1),unauthorized(2)}

PaeControlledPortControl ::= TEXTUAL-CONVENTION

STATUS mandatory

DESCRIPTION

"AAC PAE 受控端口的控制值。"

SYNTAX INTEGER {forceUnauthorized(1),auto(2),forceAuthorized(3)}

-- PAE MIB 中的组

gbt28455PaeSystem OBJECT IDENTIFIER ::= {paeMIBObjects 1}

gbt28455PaeAAC OBJECT IDENTIFIER ::= {paeMIBObjects 2}

gbt28455PaeREQ OBJECT IDENTIFIER ::= {paeMIBObjects 3}

-- PAE 系统组

gbt 28455PaeSystemAuthControl OBJECT-TYPE

SYNTAX INTEGER{enabled(1),disabled(2)}

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"用于系统中端口接入控制的管理状态:使能/禁止。"

REFERENCE

"GB/T28455 8.6.1,SystemAuthControl"

::={gbt28455PaeSystem 1}

gbt28455PaeSystemAbnormalAlert OBJECT-TYPE

SYNTAX VALUEABLES{厂商 OUI}

SYNTAX VALUEABLES{设备 SN}

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"用于系统中异常状态时触发告警。"

REFERENCE

"GB/T28455 8.2.3"

::={gbt28455PaeSystem 2}

-- PAE 端口表

gbt28455PaePortTable OBJECT-TYPE

SYNTAX SEQUENCE OF GBT28455PaePortEntry

MAX-ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"PAE 支持的每个端口的系统级信息表。出现在表中的一项对应本系统的一个端口。"

REFERENCE

"GB/T28455 8.7.2"

::={gbt28455PaeSystem 2}

gbt28455PaePortEntry OBJECT-TYPE

SYNTAX GBT28455PaePortEntry

MAX-ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"端口的端口号、协议版本、初始化控制。"

INDEX{gbt28455PaePortNumber}

::={gbt28455PaePortTable 1}

GBT28455PaePortEntry ::=

SEQUENCE {

gbt28455PaePortNumber	InterfaceIndex,
gbt28455PaePortProtocolVersion	Unsigned32,
gbt28455PaePortCapabilities	BITS,
gbt28455PaePortPriority	Integer,
gbt28455PaePortSecureHello	TruthValue,
gbt28455PaePortInitialize	TruthValue,
gbt28455PaePortReauthenticate	TruthValue

}

gbt28455PaePortNumber OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"与该端口有关的端口号。"

REFERENCE

"GB/T28455 8.7.2, 端口号"

::={gbt28455PaePortEntry 1}

gbt28455PaePortProtocolVersion OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS mandatory
DESCRIPTION
"与该端口有关的协议版本。"
REFERENCE
"GB/T28455 8.7.2,协议版本"
 ::= { gbt28455PaePortEntry 2 }

gbt28455PaePortCapabilities OBJECT-TYPE

SYNTAX BITS {
 gbt28455PaePortAACCapable(0),-- 支持 AAC 功能
 gbt28455PaePortREQCapable(1),-- 支持 REQ 功能
 gbt28455PaePortSelfAdaptedCapable(2)-- 支持自适应功能
 }
MAX-ACCESS read-only
STATUS mandatory
DESCRIPTION
"指示该端口支持的且可通过该 MIB 管理的 PAE 功能。"
REFERENCE
"GB/T28455 8.7.2,PAE 能力"
 ::= { gbt28455PaePortEntry 3 }

gbt28455PaePortPriority OBJECT-TYPE

SYNTAX INTEGER {
 REQ(0),-- 请求者的优先级为 0
 AAC(255),-- 鉴别访问控制器的优先级为 255
 selfAdaptedPAE-1(1),
 selfAdaptedPAE-2(2),
 ... ,
 selfAdaptedPAE-254(254) --自适应 PAE 的优先级为 1~254
 }

MAX-ACCESS read-only
STATUS mandatory
DESCRIPTION
"指示该端口支持的 PAE 优先级的能力。"
REFERENCE
"GB/T28455 8.7.2,PAE 优先级"

::= { gbt28455AuthConfigEntry 4 }

gbt28455PaePortSecureHello OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-write 
STATUS mandatory
DESCRIPTION
"指示端口 PAE 是否支持安全 Hello. True 表示支持,Flase 表示不支持。"

REFERENCE

"GB/T28455 8.7.2,安全 Hello "

::={gbt28455PaePortEntry 5}

gbt28455PaePortInitialize OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"用于该端口的初始化控制。设置该属性为 TRUE 时引起端口初始化,一旦初始化完成,该属性值翻转为 FALSE。"

REFERENCE

"GB/T28455 8.7.2.4,初始化端口"

::={gbt28455PaePortEntry 6}

gbt28455PaePortReauthenticate OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"用于该端口的重新鉴别控制。设置该属性为 TRUE 时,引起该端口的 PAE 状态机重新鉴别对方;设置该属性为 FALSE 时不起任何作用。

读取该属性时,总返回 FALSE。"

REFERENCE

"GB/T28455 8.5.2.4,重新鉴别"

::={gbt28455PaePortEntry 7}

-- PAE AAC 组

-- AAC 配置表

gbt28455AuthConfigTable OBJECT-TYPE

SYNTAX SEQUENCE OF GBT28455AuthConfigEntry

MAX-ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"包含与每个端口有关的 AAC PAE 的配置对象的表。出现在表中的项对应可鉴别访问它自己的端口。"

REFERENCE

"GB/T28455 8.5.2,AAC 配置"

::={gbt28455PaeAAC 1}

gbt28455AuthConfigEntry OBJECT-TYPE

SYNTAX GBT28455AuthConfigEntry

MAX-ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"用于 AAC AE 的配置信息。"

INDEX { gbt28455PaePortNumber }

::= { gbt28455AuthConfigTable 1 }

GBT28455AuthConfigEntry ::=

SEQUENCE {

gbt28455AuthPaeState

INTEGER,

gbt28455AuthBackendAuthState

INTEGER,

gbt28455AuthAdminControlledDirections

PaeControlledDirections,

gbt28455AuthOperControlledDirections

PaeControlledDirections,

gbt28455AuthAuthControlledPortStatus

PaeControlledPortStatus,

gbt28455AuthAuthControlledPortControl

PaeControlledPortControl,

gbt28455AuthQuietPeriod

Unsigned32,

gbt28455ServerTiomeout

Unsigned32,

gbt28455AuthReAuthPeriod

Unsigned32,

gbt28455AuthReAuthEnabled

TruthValue,

gbt28455AuthKeyEnabled

TruthValue

}

gbt28455AuthPaeState OBJECT-TYPE

SYNTAX INTEGER {

initialize(1),

disconnected(2),

connecting(3),

authenticating(4),

authenticated(5),

aborting(6),

held(7),

forceAuth(8),

forceUnauth(9),

restart(10),

authorized(11)

}

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"AAC PAE 状态机的当前值。"

REFERENCE

"GB/T28455 8.5.2, AAC PAE 状态"

::={gbt28455AuthConfigEntry 1}

gbt28455AuthBackendAuthState OBJECT-TYPE

SYNTAX INTEGER {

request(1),
response(2),
success(3),
fail(4),
timeout(5),
idle(6),
initialize(7),
ignore(8)

}

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"Backend AAC 状态机的当前状态。"

REFERENCE

"GB/T28455 8.5.2, Backend AAC 状态"

::={gbt28455AuthConfigEntry 2}

gbt28455AuthAdminControlledDirections OBJECT-TYPE

SYNTAX PaeControlledDirections

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"端口的管理控制方向参数的当前值。"

REFERENCE

"GB/T28455 8.5.2, 管理控制模式"

::={gbt28455AuthConfigEntry 3}

gbt28455AuthOperControlledDirections OBJECT-TYPE

SYNTAX PaeControlledDirections

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"端口的操作控制方向参数的当前值。"

REFERENCE

"GB/T28455 8.5.2, 操作控制模式"

::={gbt28455AuthConfigEntry 4}

gbt28455AuthAuthControlledPortStatus OBJECT-TYPE

SYNTAX PaeControlledPortStatus

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"端口的受控端口状态参数的当前值。"

REFERENCE

"GB/T28455 8.5.2,AuthControlledPortStatus"

::={gbt28455AuthConfigEntry 5}

gbt28455AuthAuthControlledPortControl OBJECT-TYPE

SYNTAX PaeControlledPortControl

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"端口的受控端口控制参数的当前值。"

REFERENCE

"GB/T28455 8.5.2,AuthControlledPortControl"

::={gbt28455AuthConfigEntry 6}

gbt28455AuthQuietPeriod OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"AAC PAE 状态机当前使用的 quietPeriod 常量的值,单位为秒。"

REFERENCE

"GB/T28455 8.5.2,quietPeriod"

DEFVAL{60}

::={gbt28455AuthConfigEntry 7}

gbt28455ServerTiomeout OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"后台鉴别状态当前使用的 serverTiomeout 常量的值,单位为秒。"

REFERENCE

"GB/T28455 8.5.2,serverTiomeout"

DEFVAL{30}

::={gbt28455AuthConfigEntry 8}

gbt28455AuthReAuthPeriod OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"Reauthentication Timer 状态机当前使用的 reAuthPeriod 常量的值,单位为秒。"

REFERENCE

"GB/T28455 8.5.2, reAuthPeriod"

DEFVAL{3600}

::={gbt28455AuthConfigEntry 9}

gbt28455AuthReAuthEnabled OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"Reauthentication Timer 状态机(7.3.8)使用的使能/禁止控制。"

REFERENCE

"GB/T28455 8.5.2, reAuthEnabled"

DEFVAL{false}

::={gbt28455AuthConfigEntry 10}

gbt28455AuthKeyEnabled OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"AAC PAE 状态机当前使用的 keyEnabled 常量的值。"

REFERENCE

"GB/T28455 8.5.2, keyEnabled"

::={gbt28455AuthConfigEntry 11}

-- AAC 统计表

gbt28455AuthStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF GB15629dotxxAuthStatsEntry

MAX-ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"包含与每个端口有关的 AAC PAE 的统计对象的表。出现在表中的项对应可鉴别访问它自己的端口。"

REFERENCE

"GB/T28455 8.5.3, AAC 统计"

::={gbt28455PaeAAC 2}

gbt28455AuthStatsEntry OBJECT-TYPE

SYNTAX GBT28455AuthStatsEntry

MAX-ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"用于 AAC PAE 的统计信息。"

INDEX { gbt28455PaePortNumber }

::= { gbt28455AuthStatsTable 1 }

GBT28455AuthStatsEntry ::=

SEQUENCE {

gbt28455AuthTaepolFramesRx	Counter32,
gbt28455AuthTaepolFramesTx	Counter32,
gbt28455AuthTaepolStartFramesRx	Counter32,
gbt28455AuthTaepolLogoffFramesRx	Counter32,
gbt28455AuthTaepolRespIdFramesRx	Counter32,
gbt28455AuthTaepolRespFramesRx	Counter32,
gbt28455AuthTaepolReqIdFramesTx	Counter32,
gbt28455AuthTaepolReqFramesTx	Counter32,
gbt28455AuthInvalidTaepolFramesRx	Counter32,
gbt28455AuthTaepLengthErrorFramesRx	Counter32,
gbt28455AuthLastTaepolFrameVersion	Unsigned32,
gbt28455AuthLastTaepolFrameSource	MacAddress

}

gbt28455AuthTaepolFramesRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 AAC 已经收到的有效的任意类型 TAEPOL 帧的个数。"

REFERENCE

"GB/T28455 8.5.3, 收到的 TAEPOL 帧"

::= { gbt28455AuthStatsEntry 1 }

gbt28455AuthTaepolFramesTx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 AAC 已经发送的任意类型 TAEPOL 帧的个数。"

REFERENCE

"GB/T28455 8.5.3,发送的 TAEPOL 帧"
 ::= {gbt28455AuthStatsEntry 2}

gbt28455AuthTaepolStartFramesRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 AAC 已经收到的 TAEPOL Start 帧的个数。"

REFERENCE

"GB/T28455 8.5.3,收到的 TAEPOL Start 帧"

::= {gbt28455AuthStatsEntry 3}

gbt28455AuthTaepolLogoffFramesRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 AAC 已经收到的 TAEPOL Logoff 帧的个数。"

REFERENCE

"GB/T28455 8.5.3,收到的 TAEPOL Logoff 帧"

::= {gbt28455AuthStatsEntry 4}

gbt28455AuthTaepolRespIdFramesRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 AAC 已经收到的 TAEP Resp/Id 帧的个数。"

REFERENCE

"GB/T28455 8.5.3,收到的 TAEPOL Resp/Id 帧"

::= {gbt28455AuthStatsEntry 5}

gbt28455AuthTaepolRespFramesRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 AAC 已经收到的有效的 TAEP Response 帧(非 Resp/Id 帧)的个数。"

REFERENCE

"GB/T28455 8.5.3,收到的 TAEPOL Response 帧"

::= {gbt28455AuthStatsEntry 6}



gbt28455AuthTaepolReqIdFramesTx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 AAC 已经发送的 TAEP Req/Id 帧的个数。"

REFERENCE

"GB/T28455 8.5.3,发送的 TAEPOL Req/Id 帧"

::={gbt28455AuthStatsEntry 7}

gbt28455AuthTaepolReqFramesTx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 AAC 已经发送的 TAEP Request 帧(非 Req/Id 帧)的个数。"

REFERENCE

"GB/T28455 8.5.3,发送的 TAEPOL Request 帧"

::={gbt28455AuthStatsEntry 8}

gbt28455AuthInvalidTaepolFramesRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 AAC 已经收到的无效的 TAEPOL 帧的个数,其中帧的类型没有被识别出来。"

REFERENCE

"GB/T28455 8.5.3,收到的无效 TAEPOL 帧"

::={gbt28455AuthStatsEntry 9}

gbt28455AuthTaepLengthErrorFramesRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 AAC 已经收到的长度错误的 TAEPOL 帧的个数,其中帧体中 Length 字段无效。"

REFERENCE

"GB/T28455 8.5.3,收到的 TAEP 长度错误帧"

::={gbt28455AuthStatsEntry 10}

gbt28455AuthLastTaepolFrameVersion OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only



STATUS mandatory

DESCRIPTION

"最近收到的 TAEPOL 帧的协议版本号。"

REFERENCE

"GB/T28455 8.5.3,最后一个 TAEPOL 帧的版本"

::={gbt28455AuthStatsEntry 11}

gbt28455AuthLastTaepolFrameSource OBJECT-TYPE

SYNTAX MacAddress

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"最近收到的 TAEPOL 帧的源 MAC 地址。"

REFERENCE

"GB/T28455 8.5.3,最后一个 TAEPOL 帧的源"

::={gbt28455AuthStatsEntry 12}

-- AAC 会话统计表

gbt28455AuthSessionStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF GBT28455AuthSessionStatsEntry

MAX-ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"包含与每个端口有关的 AAC PAE 的会话统计对象的表。出现在表中的项对应可鉴别访问它自己的端口。"

REFERENCE

"GB/T28455 8.5.5"

::={gbt28455PaeAAC 3}



gbt28455AuthSessionStatsEntry OBJECT-TYPE

SYNTAX GBT28455AuthSessionStatsEntry

MAX-ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"用于 AAC PAE 的会话统计信息。表示为每个还在进行的会话收集的当前值或者端口当前没有激活会话时最后有效会话的最终值。"

INDEX{gbt28455PaePortNumber}

::={ gbt28455AuthSessionStatsTable 1}

GBT28455AuthSessionStatsEntry ::=

SEQUENCE {

gbt28455AuthSessionOctetsRx	Counter64,
gbt28455AuthSessionOctetsTx	Counter64,
gbt28455AuthSessionFramesRx	Counter32,
gbt28455AuthSessionFramesTx	Counter32,
gbt28455AuthSessionId	SnmpAdminString,
gbt28455AuthSessionAuthenticMethod	INTEGER,
gbt28455AuthSessionTime	TimeTicks,
gbt28455AuthSessionTerminateCause	INTEGER,
gbt28455AuthSessionUserName	SnmpAdminString
}	

gbt28455AuthSessionOctetsRx OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"在会话期间,该端口上用户数据帧中收到的八位位组数。"

REFERENCE

"GB/T28455 8.5.5,收到的会话八位位组"

::={gbt28455AuthSessionStatsEntry 1}

gbt28455AuthSessionOctetsTx OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"在会话期间,该端口上用户数据帧中发送的八位位组数。"

REFERENCE

"GB/T28455 8.5.5,发送的会话八位位组"

::={gbt28455AuthSessionStatsEntry 2}

gbt28455AuthSessionFramesRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"在会话期间,该端口收到的用户数据帧的个数。"

REFERENCE

"GB/T28455 8.5.5,收到的会话帧"

::={gbt28455AuthSessionStatsEntry 3}

gbt28455AuthSessionFramesTx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "在会话期间,该端口发送的用户数据帧的个数。"
 REFERENCE
 "GB/T28455 8.5.5,发送的会话帧"
 ::= {gbt28455AuthSessionStatsEntry 4}

gbt28455AuthSessionId OBJECT-TYPE

SYNTAX SnmpAdminString
 MAX-ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "会话的惟一标识,可打印的至少 3 个字符的 ASCII 串。"
 REFERENCE
 "GB/T28455 8.5.5,会话标识"
 ::= {gbt28455AuthSessionStatsEntry 5}

gbt28455AuthSessionAuthenticMethod OBJECT-TYPE

SYNTAX INTEGER {remoteAuthServer(1),localAuthServer(2)}
 MAX-ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "用于建立会话的鉴别方法。"
 REFERENCE
 "GB/T28455 8.5.5,会话鉴别方法"
 ::= {gbt28455AuthSessionStatsEntry 6}

gbt28455AuthSessionTime OBJECT-TYPE

SYNTAX TimeTicks
 MAX-ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "会话持续时间,单位为秒。"
 REFERENCE
 "GB/T28455 8.5.5,会话时间"
 ::= {gbt28455AuthSessionStatsEntry 7}

gbt28455AuthSessionTerminateCause OBJECT-TYPE

SYNTAX INTEGER {
 REQLogoff(1),
 portFailure(2),
 REQRestart(3),



```
reauthFailed(4),  
authControlForceUnauth(5),  
portReInit(6),  
portAdminDisabled(7),  
notTerminatedYet(999)  
}
```

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"会话终止的原因。"

REFERENCE

"GB/T28455 8.5.5,会话终止原因"

::={gbt28455AuthSessionStatsEntry 8}

gbt28455AuthSessionUserName OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"表示 REQ PAE 身份的用户名。"

REFERENCE

"GB/T28455 8.5.5,会话用户名"

::={gbt28455AuthSessionStatsEntry 9}

-- PAE REQ 组

-- REQ 配置表

gbt28455REQConfigTable OBJECT-TYPE

SYNTAX SEQUENCE OF GBT28455REQConfigEntry

MAX-ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"包含与每个端口有关的 REQ PAE 的配置对象的表。出现在表中的项对应远端系统发起的鉴别它自己的端口。"

REFERENCE

"GB/T28455 8.6.2"

::={gbt28455PaeREQ 1}

gbt28455REQConfigEntry OBJECT-TYPE

SYNTAX GBT28455REQConfigEntry

MAX-ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 "用于 REQ PAE 的配置信息。"
 INDEX { gbt28455PaePortNumber}
 ::= {gbt28455REQConfigTable 1}

GBT28455REQConfigEntry ::=

SEQUENCE {	
gbt28455REQPaeState	INTEGER,
gbt28455REQHeldPeriod	Unsigned32,
gbt28455REQAuthPeriod	Unsigned32,
gbt28455REQStartPeriod	Unsigned32,
gbt28455REQMaxStart	Unsigned32,
gbt28455REQControlledPortStatus	PaeControlledPortStatus,
gbt28455REQAccessCtrlWithAuth	INTEGER,
gbt28455REQBackendState	INTEGER
}	

gbt28455REQPaeState OBJECT-TYPE

SYNTAX INTEGER {

disconnected(1),
logoff(2),
connecting(3),
authenticating(4),
authenticated(5),
held(6),
restart(7),
sForceAuth(8),
sForceUnauth(9),
Authorized(10)

}

MAX-ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "REQ PAE 状态机(7.3.11)的当前状态。"
 REFERENCE
 "GB/T28455 8.6.2,REQ PAE 状态"
 ::= {gbt28455REQConfigEntry 1}

gbt28455REQHeldPeriod OBJECT-TYPE

SYNTAX Unsigned32
 MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"REQ PAE 状态机(7.3.11.2.2)当前使用的 heldPeriod 常量的值,单位为秒。"

REFERENCE

"GB/T28455 8.6.2,heldPeriod"

DEFVAL{60}

::={gbt28455REQConfigEntry 2}

gbt28455REQAuthPeriod OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"REQ PAE 状态机(7.3.12.2.2)当前使用的 authPeriod 常量的值,单位为秒。"

REFERENCE

"GB/T28455 8.6.2,authPeriod"

DEFVAL{30}

::={gbt28455REQConfigEntry 3}

gbt28455REQStartPeriod OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"REQ PAE 状态机(7.3.11.2.2)当前使用的 startPeriod 常量的值,单位为秒。"

REFERENCE

"GB/T28455 8.6.2,startPeriod"

DEFVAL{30}

::={gbt28455REQConfigEntry 4}

gbt28455REQMaxStart OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-write

STATUS mandatory

DESCRIPTION

"REQ PAE 状态机(7.3.11.2.2)当前使用的 maxStart 常量的值。"

REFERENCE

"GB/T28455 8.6.2,maxStart"

DEFVAL{3}

::={gbt28455REQConfigEntry 5}

gbt28455REQControlledPortStatus OBJECT-TYPE

SYNTAX PaeControlledPortStatus

MAX-ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "REQ PAE 状态机(7.3.11)的当前状态。"
 REFERENCE
 "GB/T28455 8.6.2,REQ PAE 状态"
 ::= {gbt28455REQConfigEntry 6}

gbt28455REQBackendState OBJECT-TYPE

SYNTAX INTEGER {
 initialize(1),
 idle(2),
 request(3),
 response(4),
 receive(5),
 fail(6),
 success(7),
 timeout(8)
 }

MAX-ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "REQ Backend 状态机的当前状态。"
 REFERENCE
 "GB/T28455,Backend REQ 状态"
 ::= {gbt28455REQConfigEntry 7}

 -- REQ 统计表

gbt28455REQStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF GBT28455REQStatsEntry
 MAX-ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION

"包含与每个端口有关的 REQ PAE 的统计对象的表。出现在表中的项对应远端系统发起的鉴别它自己的端口。"

REFERENCE
 "GB/T28455 8.6.3"
 ::= { gbt28455PaeREQ 2}

gbt28455REQStatsEntry OBJECT-TYPE

SYNTAX GBT28455REQStatsEntry



MAX-ACCESS not-accessible
STATUS mandatory
DESCRIPTION
 "用于 REQ PAE 的统计信息。"
INDEX {gbt28455PaePortNumber}
 ::= {gbt28455REQStatsTable 1}

GBT28455REQStatsEntry ::=
SEQUENCE {
 gbt28455REQTaepolFramesRx Counter32,
 gbt28455REQTaepolFramesTx Counter32,
 gbt28455REQTaepolStartFramesTx Counter32,
 gbt28455REQTaepolLogoffFramesTx Counter32,
 gbt28455REQInvalidTaepolFramesRx Counter32,
 gbt28455REQTaepLengthErrorFramesRx Counter32,
 gbt28455REQLastTaepolFrameVersion Unsigned32,
 gbt28455REQLastTaepolFrameSource MacAddress
}

gbt28455REQTaepolFramesRx OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS mandatory
DESCRIPTION
 "该 REQ 收到的任意类型的 TAEPoL 帧的个数。"
REFERENCE
 "GB/T28455 8.6.3,收到的 TAEPoL 帧"
 ::= {gbt28455REQStatsEntry 1}

gbt28455REQTaepolFramesTx OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS mandatory
DESCRIPTION
 "该 REQ 发送的任意类型的 TAEPoL 帧的个数。"
REFERENCE
 "GB/T28455 8.6.3,发送的 TAEPoL 帧"
 ::= {gbt28455REQStatsEntry 2}

gbt28455REQTaepolStartFramesTx OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS mandatory



DESCRIPTION

"该 REQ 发送的 TAEPoL-Start 帧的个数。"

REFERENCE

"GB/T28455 8.6.3, 发送的 TAEPoL-Start 帧"

::={gbt28455REQStatsEntry 3}

gbt28455REQTaepolLogoffFramesTx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 REQ 发送的 TAEPoL-Logoff 帧的个数。"

REFERENCE

"GB/T28455 8.6.3, 发送的 TAEPoL-Logoff 帧"

::={gbt28455REQStatsEntry 4}

gbt28455REQInvalidTaepolFramesRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 REQ 收到的无效的 TAEPoL 帧的个数, 其中帧的类型没被识别出来。"

REFERENCE

"GB/T28455 8.6.3, 收到的无效 TAEPoL 帧"

::={gbt28455REQStatsEntry 5}

gbt28455REQTaepLengthErrorFramesRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"该 REQ 收到的长度错误的 TAEPoL 帧的个数, 其中帧体中 Length 字段无效。"

REFERENCE

"GB/T28455 8.6.3, 收到的 TAEP 长度错误帧"

::={gbt28455REQStatsEntry 6}

gbt28455REQLastTaepolFrameVersion OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS mandatory

DESCRIPTION

"最近收到的 TAEPoL 帧的协议版本号。"

REFERENCE

"GB/T28455 8.6.3,最后一个 TAEPoL 帧的版本"
 ::= { gbt28455REQStatsEntry 7 }

gbt28455REQLastTaepolFrameSource OBJECT-TYPE
SYNTAX MacAddress
MAX-ACCESS read-only
STATUS mandatory
DESCRIPTION
 "最近收到的 TAEPoL 帧的源 MAC 地址。"
REFERENCE
 "GB/T28455 8.6.3,最后一个 TAEPoL 帧的源"
 ::= { gbt28455REQStatsEntry 8 }

-- GB/T28455 MIB — 一致性信息

gbt28455PaeConformance OBJECT IDENTIFIER
 ::= { gbt28455paeMIB 2 }
gbt28455PaeGroups OBJECT IDENTIFIER
 ::= { gbt28455PaeConformance 1 }
gbt28455PaeCompliances OBJECT IDENTIFIER
 ::= { gbt28455PaeConformance 2 }

-- 一致性单元

gbt28455PaeSystem Group OBJECT-GROUP
OBJECTS {
 gbt28455PaeSystemAuthControl,
 gbt28455PaePortProtocolVersion,
 gbt28455PaePortCapabilities,
 gbt28455PaePortPriority,
 gbt28455PaePortSecureHello,
 gbt28455PaePortInitialize,
 gbt28455PaePortReauthenticate
}
STATUS mandatory
DESCRIPTION
 "提供关于 PAE 的系统信息及通过 PAE 控制的对象集合。"
 ::= { gbt28455PaeGroups 1 }

gbt28455PaeAuthStatsGroup OBJECT-GROUP
OBJECTS {
 gbt28455AuthTaepolFramesRx,



```

gbt28455AuthTaepolFramesTx,
gbt28455AuthTaepolStartFramesRx,
gbt28455AuthTaepolLogoffFramesRx,
gbt28455AuthTaepolRespIdFramesRx,
gbt28455AuthTaepolRespFramesRx,
gbt28455AuthTaepolReqIdFramesTx,
gbt28455AuthTaepolReqFramesTx,
gbt28455AuthInvalidTaepolFramesRx,
gbt28455AuthTaepLengthErrorFramesRx,
gbt28455AuthLastTaepolFrameVersion,
gbt28455AuthLastTaepolFrameSource
}

```

STATUS mandatory

DESCRIPTION

"提供 AAC PAE 统计信息的对象集合。"

::={gbt28455PaeGroups 2}

gbt28455PaeAuthSessionStatsGroup OBJECT-GROUP

OBJECTS {

```

gbt28455AuthSessionOctetsRx,
gbt28455AuthSessionOctetsTx,
gbt28455AuthSessionFramesRx,
gbt28455AuthSessionFramesTx,
gbt28455AuthSessionId,
gbt28455AuthSessionAuthenticMethod,
gbt28455AuthSessionTime,
gbt28455AuthSessionTerminateCause,
gbt28455AuthSessionUserName
}

```

STATUS mandatory

DESCRIPTION

"为 AAC PAE 提供关于当前或最终会话的统计对象集合。"

::={gbt28455PaeGroups 3}

gbt28455PaeREQConfigGroup OBJECT-GROUP

OBJECTS {

```

gbt28455REQPaeState,
gbt28455REQHeldPeriod,
gbt28455REQAuthPeriod,
gbt28455REQStartPeriod,
gbt28455REQMaxStart,
gbt28455REQControlledPortStatus,
gbt28455REQAccessCtrlWithAuth,

```



gbt28455REQBackendState

}

STATUS mandatory

DESCRIPTION

"提供 REQ PAE 配置信息的对象集合。"

::={gbt28455PaeGroups 4}

gbt28455PaeAuthConfigGroup OBJECT-GROUP

OBJECTS {

gbt28455AuthPaeState,

gbt28455AuthBackendAuthState,

gbt28455AuthAdminControlledDirections,

gbt28455AuthOperControlledDirections,

gbt28455AuthAuthControlledPortStatus,

gbt28455AuthAuthControlledPortControl,

gbt28455AuthQuietPeriod,

gbt28455AuthServerTimeout,

gbt28455AuthReAuthPeriod,

gbt28455AuthReAuthEnabled,

gbt28455AuthKeyTxEnabled

}

STATUS mandatory

DESCRIPTION

"提供 AAC PAE 配置信息的对象集合。"

::={gbt28455PaeGroups 5}

gbt28455PaeREQStatsGroup OBJECT-GROUP

OBJECTS {

gbt28455REQTaepolFramesRx,

gbt28455REQTaepolFramesTx,

gbt28455REQTaepolStartFramesTx,

gbt28455REQTaepolLogoffFramesTx,

gbt28455REQInvalidTaepolFramesRx,

gbt28455REQTaepLengthErrorFramesRx,

gbt28455REQLastTaepolFrameVersion,

gbt28455REQLastTaepolFrameSource

}

STATUS mandatory

DESCRIPTION

"提供 REQ PAE 统计信息的对象集合。"

::={gbt28455PaeGroups 6}

 -- 一致性声明

gbt28455PaeCompliance MODULE-COMPLIANCE

STATUS mandatory

DESCRIPTION

"设备支持端口接入控制的一致性声明。"

MODULE

MANDATORY-GROUPS {

gbt28455PaeSystem Group

}

GROUP gbt28455PaeAuthConfigGroup

DESCRIPTION

"对于支持 PAE 的 AAC 功能的系统,该组是必备的。"

OBJECT gbt28455AuthAdminControlledDirections

SYNTAX INTEGER {

both(0)

}

MIN-ACCESS read-only

DESCRIPTION

"支持 in(1)是可选的。"

OBJECT gbt28455AuthOperControlledDirections

SYNTAX INTEGER {

both(0)

}

DESCRIPTION

"支持 in(1)是可选的。"

OBJECT gbt28455AuthKeyEnabled

MIN-ACCESS read-only

DESCRIPTION

"不支持 TAEPOL-Key 帧的 AAC PAE 可实现该对象为仅读的,返回值为 FALSE。"

GROUP gbt28455PaeAuthStatsGroup

DESCRIPTION

"对于支持 PAE 的 AAC 功能的系统,该组是必备的。"

GROUP gbt28455PaeAuthSessionStatsGroup 

DESCRIPTION

"对于支持 PAE 的 AAC 功能的系统,该组是可选的。"

GROUP gbt28455PaeREQConfigGroup

DESCRIPTION

"对于支持 PAE 的 REQ 功能的系统,该组是必备的。"

GROUP gbt28455PaeREQStatsGroup

DESCRIPTION

"对于支持 PAE 的 REQ 功能的系统,该组是必备的。"

::={gbt28455PaeCompliances 1}

END



附 录 A
(规范性附录)
PICS 形式表

A.1 引言

声称与本标准一致的协议实现供应商应填写下列协议实现一致性声明(PICS)形式表。

已填写的 PICS 形式表即为正在讨论的用于实现的 PICS。PICS 是对已实现协议的能力和选项的声明。

PICS 可有多种用途,包括:

- 协议实现者将其作为检查清单,通过对其进行监督以降低与本部分不一致的风险;
- 协议实现的供应商和获得者或潜在的获得者将 PICS 作为实现能力的详细指示,说明了它与标准的 PICS 形式表所提供的通常理解基础的相对关系;
- 实现的用户或潜在的用户将其作为初始检查与另一种实现互操作的可能性的基础(注意,尽管从不保证互操作,但不能互操作往往能从不兼容的 PICS 中预测出来);
- 协议测试者将其作为选择合适测试的基础,根据这些测试对实现一致性的声称进行评估。

A.2 缩略语和状态符号

A.2.1 状态符号

M	必备的
O	可选的
O. <n>	可选的,但要求至少支持一组由相同数字<n>标记的选项
X	禁止的
pred:	条件项的符号,包括谓词标识(见 A.3.4)



A.2.2 缩略语

N/A	不适用
PICS	协议实现一致性声明

A.3 填写 PICS 形式表说明

A.3.1 PICS 形式表的通用结构

PICS 形式表的第一部分,即“实现标识和协议概要”,用于填写为充分识别供应商和实现方式所必需的信息。

PICS 形式表的主要部分是一张具有固定格式的调查表,它被分为若干部分,每部分包含若干独立项。对调查表各项的回答放在最右边的一列中,它或者是简单地勾出一个答案以指明一个受限制的选择(通常用“是”或“否”),或者是键入一个值或值的集合或范围(注意,对于某些项目,若一组可能的答案中两种或多种选择都适用,则所有有关的选择都要勾出)。

每个项目通过第一栏的项目引用标识,第二栏包含需回答的问题,第三栏记录项目的状态即该支持

是必备的、可选的还是有条件的,另见 A.3.4。第四栏包含在本附录的正文中规定该项目的的一个或多个引用材料,第五栏提供空格以填写答案。

供应商还可以提供或被要求提供进一步的信息,这些信息分为附加信息和异常信息两类。当提供时,每类进一步的信息要在分别标以 A<i>或 X<i>项目的子部分中提供,为了达到交叉引用的目的,其中<i>是该项目无二义性的标识(例如一个简单的数字),对其格式或表示没有其他的限制。

包括任何附加信息和异常信息的一张已填好的 PICS 形式表,是正在讨论的用于实现的 PICS。

注意,一种实现能以一种以上的方式配置,单个的 PICS 能够描述所有这些配置。然而,如果使信息的表示更容易和更清楚,那么供应商还可以提供一个以上的 PICS,且每个 PICS 覆盖实现配置能力的某个子集。

A.3.2 附加信息

附加信息项目允许供应商提供进一步的信息以试图帮助解释 PICS。不企图或不希望它供给大量的信息,在没有任何这种信息的情况下,也可认为 PICS 是完整的。例子可能是若干方法的一种概括,其中单个实现能被建立起来,以便工作在各种环境和配置下;或者关于实现方面的信息虽然超出本附录范围,但对某些项的回答有影响。

附加信息项的引用可在调查表中的任何答案后填写,且还可以包含在异常信息项目中。

A.3.3 异常信息

供应商希望用与指定的需求相矛盾的方式(在任何条件业已加上之后)回答带有必备状态的项目,这或许是一种偶然发生的情况。在支持栏中,找不到预先写好的答案,相反,供应商应在支持栏中写入遗漏的答复,同时写入异常信息项目的引用 X<i>,并应在异常项目自身中提供一个合适的理由。

按这种方式,要求异常项目的实现不符合本附录的规定。

注意,对于上述情况的一种可能理由是,本标准中已经报告的某种缺陷,期望对其进行纠正以更改实现不能满足的需求。

A.3.4 条件状态

A.3.4.1 条件项目

PICS 形式表包含了若干有条件的项目。这些有条件的项目自身的适用性及其状态是否适用(必备的或可选的)均取决于是否支持某些其他项目。

当一组项目在适用性方面受限于同一条件时,单独的关于条件的初步问题出现在该组的开头处。如果选择的是 N/A 答案,则表示可跳到调查表较后的位置,否则单独的条件项目在状态栏中用条件符号示出。

条件符号形式为“<pred>:<s>”,其中<pred>为 A.3.4.2 描述的谓词,而<s>为状态符号 M 或 O。

如果谓词值为真(见 A.3.4.2),则有条件的项目可适用,其状态通过谓语后的状态符号给出,用通常的方式填写支持项目。若谓词值为假,则有条件的项目不适用,选择 N/A 答案。

A.3.4.2 谓词

谓词为下列情况之一:

- a) 在 PICS 形式表中某个项目的项目引用:如果该目标标记为被支持,谓词的值为真,否则为假。
- b) 谓词名称通过使用布尔操作符“OR”组合项目引用所构造的布尔表达式来定义:如果一个或多个项目标记为“被支持”,则谓词值为真,否则为假。

- c) 谓语句名称通过使用布尔操作符“AND”组合项目引用所构造的布尔表达式来定义:如果所有项目标记为“被支持”,则谓语句值为真,否则为假。

在谓语句或用于成组有条件项目预备的问题中引用的每个项目在项目栏中用星号(*)标注。

A.4 GB/T 28455 的 PICS 形式表

A.4.1 实现标识

见表 A.1。

表 A.1 实现标识

供应商	
咨询有关 PICS 的联系方式	
实现名称与版本	
用于完整标识的其他信息,比如,机器的名称与版本,操作系统名称	
<p>注 1: 只有前三项对于所有实现都要求;在满足完整标识的情况下可适当给出其他信息。</p> <p>注 2: 对项目名称和版本应作适应解释,以符合供应商的术语(如类型、系列和模型)。</p>	

A.4.2 协议概要

见表 A.2。

表 A.2 协议概要

协议规范的标识	GB/T 28455—2012
作为本 PICS 一部分的对 PICS 形式表的修订和勘误的标识	Amd. : Corr. : Amd. : Corr. :
是否需要异常项目? (见 A.3.3:答案“是”则意味着实现不符合本附录)	No <input type="checkbox"/> Yes <input type="checkbox"/>
声明的日期	


A.5 主要能力与选项

见表 A.3。

表 A.3 主要能力与选项

项目	特征	状态	引用条号	支持
* auth	通过非受控端口支持端口访问实体 PAE 的操作,称为 AAC PAE	O.1	5.3,7,8.5,8.7	Yes [] No []
* req	通过非受控端口支持端口访问实体 PAE 的操作,称为 REQ PAE	O.1	5.3,7,8.6,8.7	Yes [] No []

表 A.3 (续)

项目	特 征	状态	引用条号	支持
sysm	支持系统配置功能	M	5,7,3,8	Yes[]
authM1	支持配置 AAC 操作的能力	auth:M	5,8,5,2	Yes[]N/A[]
authM2	支持维护和获取 AAC 统计的能力	auth:M	5,8,5,3	Yes[]N/A[]
authM3	支持与 AuthControlledPortControl 的值强制未授权、自动及强制已授权一致的受控端口操作	M	5,3	Yes[]N/A[]
authM4	支持通过管理设置 AuthControlledPortControl 的值为强制未授权、自动或强制已授权的能力	M	5,3, 8,5,2	Yes[]N/A[]
	口操作			
authM6	支持通过管理对 Subjectt 的定期重新鉴别与对 reAuthTimer 和 reAuthEnabled 参数的配置	auth:M	5,4,3,7,3,8, 8,5,2	Yes[]N/A[]
reqM1	支持配置 REQ 操作的能力	req:M	5,8,6,2	Yes[]N/A[]
reqM2	支持维护和获取 REQ 统计的能力	req:M	5,8,6,3	Yes[]N/A[]
authO1	支持维护和获取 AAC 诊断的能力	auth:O	8,5,4	Yes[]No[]
authO2	支持维护和获取 AAC 会话统计的能力	auth:O	8,5,5	Yes[]No[]
* authO3	通过管理,支持与 AdminControlledDirections 和 OperControlledDirections 的值 Both 一致的受控端口操作,并支持设置 AdminControlledDirections 参数为 Both 和 In 值的能力	 O	5,3,2, 8,5,2	Yes[]No[]
* authO4	通过管理,支持向 REQ 发送密钥信息或接受密钥信息的能力及修改 KeyTransmissionEnabled 参数的能力	auth:O	7,3,6,8,5,2	Yes[]No[]
* reqO1	通过管理,支持向 AAC 发送密钥信息的能力及修改 KeyTransmissionEnabled 参数的能力	auth:O	7,3,6,8,6,2	Yes[]No[]
* ether	支持基于 GB/T 15629.3 以太 MAC 的 TAEPOL 封装	O,2	6,4	Yes[]No[]
* trfddi	支持基于令牌环网/FDDI MAC 的 TAEPOL 封装	O,2	6,3	Yes[]No[]
mgt	依靠端口访问控制 SNMP MIB,利用第 8 章定义的功能,支持远程控制	O	9	Yes[]No[]

A.6 TAEPOL 帧格式

TAEPOL 帧格式,见表 A.4。

表 A.4 TAEPoL 帧格式

项目	特 征	状态	引用条号	支持
taepol	用于 AAC 和 REQ 之间的 TAEPoL 封装	M	6	Yes[]
norif	RIF 不会出现在基于令牌环网/FDDI 封装的 TAEPoL 帧中	trfddi:M	6.3	Yes[]N/A []
vtag	TAEPoL 帧不会标有 VLAN	M	6.5	Yes[]
ptag1	支持标有优先级的 TAEPoL 帧的接收	M	6.5	Yes[]
ptag2	支持标有优先级的 TAEPoL 帧的发送	O	6.5	Yes[]No[]
petype	定义的发送 TAEPoL 帧中的 PAE 以太类型字段	ether:M	6.3	Yes[]N/A []
psnap	定义的发送 TAEPoL 帧中的 SNAP 编码的以太类型字段	trfddi:M	6.3	Yes[]N/A []
pver	定义的用于发送 TAEPoL 帧中的协议版本	M	6.6.2	Yes[]
ptype	分组类型的保留值将不用于发送的 TAEPoL 帧中	M	6.6.3	Yes[]
pvalid	帧将根据有效规则被处理和解释	M	6.7	Yes[]
ppvi	接收时检查协议版本标识	X	6.7	No[]

PAE 支持见表 A.5。

表 A.5 PAE 支持

项目	特 征	状态	引用条号	支持
	状态机支持			
mach	根据每个端口支持的 PAE 角色,实现支持每个端口所要求的系列状态机	M	7,表 6	Yes[]
timers	定义支持的端口定时器状态机	M	7.3.4,图 23,	Yes[]
apasm	定义支持的 AAC PAE 状态机	auth:M	7.3.5, 图 24, 7.3.5.2,7.3.3	Yes[] N/A []
akey	定义支持的 AAC 密钥状态机	authO4:M	7.3.6,7.3.7, 图 25,图 26, 7.3.3	Yes[] N/A []
skey	定义支持的 REQ 密钥状态机	reqO1:M	7.3.6,7.3.7, 图 25,图 26, 7.3.3	Yes[] N/A []
rtsm	定义支持的重新鉴别定时器状态机	auth:M	7.3.8,图 27, 7.3.3	Yes[] N/A []
basasm	定义支持的后台鉴别状态机	auth:M	7.3.9,图 28, 7.3.3, 7.3.9.2	Yes[] N/A []

表 A.5 (续)

项目	特 征	状态	引用条号	支持
cdsm	定义支持控制方向的状态机	auth:M	7.3.10,图 29, 7.3.3, 7.3.10.2	Yes[] N/A []
cdbd	定义的任意桥端口支持的桥检测状态机	bridge:M	7.3.10, IEEE Std 802.1D- 2004 中的第 17 章	Yes[] N/A []
spsm	定义支持的 REQ PAE 状态机	req:M	7.3.11,图 30, 7.3.3, 7.3.11.2	Yes[] N/A []
spbe	定义支持的 REQ 后台状态机	req:M	7.3.12,图 31, 7.3.3, 7.3.12.2	Yes[] N/A []

谓语句: bridge = auth AND authO3 AND {端口为桥端口}



附录 B
(资料性附录)
基于 TAEP 封装的鉴别协议

B.1 TAEP-IBAP(TAEP-Identity Based Authentication Protocol)

B.1.1 概述

基于 ID 鉴别协议也就是基于身份密码体制的鉴别协议,若需要进行公钥撤销查询,则鉴别过程还需要鉴别服务器的参与,也就是请求者、鉴别访问控制器与鉴别服务器共同完成鉴别;若不需要进行公钥撤销查询,则请求者和鉴别访问控制器完成鉴别。本附录中的公私钥是指基于 ID 的公私钥。

B.1.2 TAEP-IBAP 协议分组格式

TAEP-IBAP 协议是基于 ID 的鉴别协议,其协议分组封装格式见图 B.1。

Message Type (8 位比特)	Data(可变)
-------------------------	----------

图 B.1 TAEP-IBAP 协议分组格式

其中:

——Message Type 类型:定义见表 B.1。

表 B.1 TAEP-IBAP 类型分组 Message Type 子类型定义

Message Type 值	定义
1	消息 1
2	消息 2
3	消息 3
4	消息 4
5	消息 5
其他	保留

——Data:每种分组的 Data 将在协议分组介绍中进行介绍。

B.1.3 TAEP-IBAP 协议字段的定义

TAEP-IBAP 协议分组中字段的定义如下:

- a) 公钥撤销结果:长度为 1 个八位位组,比特 0 有意义。
- b) 一次性随机数:长度为 32 个八位位组。
- c) 请求者身份标识,见图 B.2。

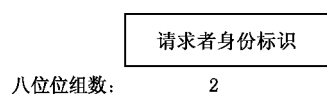


图 B.2 请求者身份标识

- d) 基于 ID 的公钥,见图 B.3。

TTP 证书身份标识	网络标识	请求者身份标识	公钥有效期限
八位位组数: 2	1	2	1

图 B.3 基于 ID 的公钥

其中:

- TTP 证书身份标识字段表示基于 ID 公私钥对的申请主体的 TTP 证书的身份;
- 网络标识表示请求者和鉴别访问控制器所在网络的标识;
- 公钥有效期限表示基于身份密码体制的公钥使用时间期限。

e) 标识 FLAG, 见图 B.4。

对鉴别访问控制器的公钥撤销查询请求	对请求者的公钥撤销查询请求	保留
比特位数: 1	1	6

图 B.4 标识 FLAG

f) 鉴别访问控制器的公钥撤销查询结果, 见图 B.5。

请求者询问	鉴别访问控制器公钥	公钥撤销结果	鉴别服务器签名
八位位组数: 4	可变	1	可变

图 B.5 鉴别访问控制器的公钥撤销查询结果

其中:

- 请求者询问为请求者产生的随机数;
- 鉴别访问控制器公钥的定义如 j) 密钥数据;
- 公钥撤销结果的定义如 a);
- 鉴别服务器签名是鉴别服务器利用基于 ID 的私钥对请求者询问、鉴别访问控制器公钥和鉴别访问控制器公钥撤销结果的签名。

g) 可变长属性字段

可变长属性字段采用类型-长度-值(TLV)格式, 格式见图 B.6:

类型	长度	值
八位位组数: 1	2	可变

图 B.6 属性格式

其中:

——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 消息鉴别码属性;
- 3 密钥数据;

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。

h) 签名属性,见图 B.7。

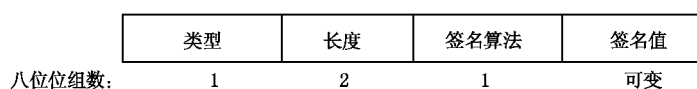


图 B.7 签名属性

其中:

——签名算法字段只包含 1 个八位位组的签名算法标识,定义如下:

1 表示 160 位的基于 ID 的数字签名算法;

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。

i) 消息鉴别码属性,见图 B.8。



图 B.8 消息鉴别码属性

其中:

——消息鉴别码算法字段只包含 1 个八位位组的消息鉴别码算法标识,定义如下:

1 表示采用 HMAC-SHA256 算法;

其他值保留。

——消息鉴别码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。内容子字段为消息鉴别码。

j) 密钥数据,见图 B.9。



图 B.9 属性格式

其中:

——密钥算法字段只包含 1 个八位位组的密钥算法标识,定义如下:

1 表示采用 ECDH 算法;

其他值保留。

——密钥值字段为密钥。

B.1.4 TAEP-IPAP 协议过程

B.1.4.1 一般要求

鉴别过程如图 B.10 所示。

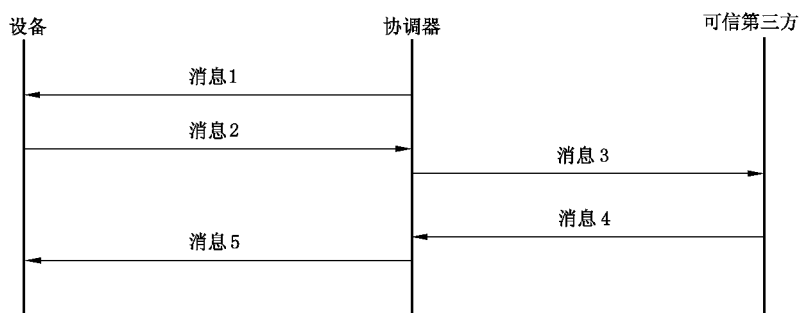


图 B.10 基于 ID 的鉴别过程

B.1.4.2 消息 1

消息 1 格式见图 B.11, Message Type=1。

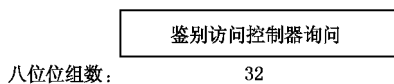


图 B.11 消息 1 格式

其中:

鉴别访问控制器询问字段长度为 32 个八位位组, 记为 N_1 。

请求者接收到由鉴别访问控制器发送的消息 1 后, 进行如下处理:

- a) 利用随机数产生器产生 32 个八位位组的请求者询问 N_2 ;
- b) 利用预安装的 ECC 域参数产生用于 ECDH 交换的临时私钥 x 和临时公钥 $x \cdot P$;
- c) 在预安装的基于 ID 的公共参数下, 请求者使用基于 ID 的私钥对鉴别访问控制器询问 N_1 、请求者询问 N_2 、对应公钥的后两个字段、临时公钥 $x \cdot P$ 和鉴别访问控制器的身份标识进行签名;
- d) 若请求者请求对鉴别访问控制器的公钥撤销查询, 则将标识 FLAG 比特 0 的值为设为 1; 否则, 设为 0; 然后生成消息 2 发送给鉴别访问控制器。

B.1.4.3 消息 2

当请求者收到鉴别访问控制器的消息 1 时, 请求者发送消息 2, Message Type=2 给鉴别访问控制器。

消息 2 格式见图 B.12。

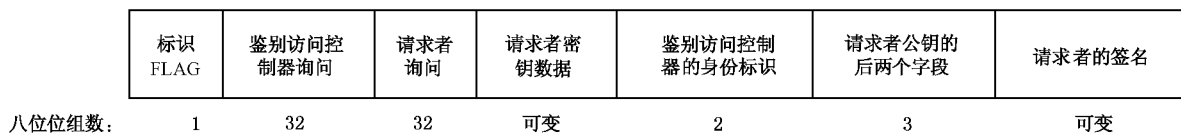


图 B.12 消息 2 格式

其中:

——标识 FLAG 字段长度为 1 个八位位组, 比特 0 和比特 1 有意义。比特 0(请求者对鉴别访问控制器的公钥撤销查询请求标识)为 1 表示请求者要求验证鉴别访问控制器的公钥是否已经被撤销, 为 0 表示不需要验证鉴别访问控制器的公钥是否已经被撤销。比特 1(鉴别访问控制器对请求者的公钥撤销查询请求标识)为 1 表示鉴别访问控制器要求验证请求者的公钥是否已经被撤销, 为 0 表示不需要验证请求者的公钥是否已经被撤销。

- 鉴别访问控制器询问字段长度为 32 个八位位组。本字段值应与鉴别访问控制器发送的消息 1 中鉴别访问控制器询问字段值相同。
- 请求者询问字段长度为 32 个八位位组,由请求者采用随机数生成算法生成,记作 N_2 。
- 请求者密钥数据格式如前定义,内容是请求者生成的用于 ECDH 交换的临时公钥。
- 鉴别访问控制器的身份标识字段的格式定义如前。
- 请求者公钥的后两个字段定义如前。
- 请求者的签名字段采用签名属性表示,其定义如前。它是对本消息中除本字段之外所有数据字段的签名。

鉴别访问控制器收到请求者发来的消息 2 后,进行如下处理:

- a) 检查鉴别访问控制器询问是否与自己在消息 1 中发送的值相同,若不同,则丢弃该消息;否则,执行 b) 操作。
- b) 检查鉴别访问控制器的身份标识是否与自己的身份标识相同,若不同,则丢弃该消息;否则,执行 c) 操作。
- c) 检查请求者公钥的后两个字段中的最后一个字段,若该字段表示的公钥使用时限已经逾期,则丢弃该消息;否则,执行 d) 操作。
- d) 级联请求者公钥的后两个字段与请求者公钥的前两个字段,然后使用该请求者公钥和预安装的基于 ID 的公开参数验证请求者的签名,若签名验证不成功,则丢弃该消息;否则,执行 e) 操作。
- e) 检查标识 FLAG 的比特 0,若比特 0 的值为 1,则执行 f) 操作;否则,执行 g) 操作。
- f) 利用随机数生成算法生成另一个鉴别访问控制器询问。若鉴别访问控制器也要求验证请求者的公钥是否被撤销,则以该鉴别访问控制器询问、消息 2 中的请求者询问、请求者公钥的后两个字段和设置好的标识 FLAG 为内容生成消息 3;否则,以该鉴别访问控制器询问、消息 2 中的请求者询问和设置好的标识 FLAG 为内容生成消息 3。生成的消息 3 在鉴别访问控制器与鉴别服务器之间的会话密钥保护下发送给鉴别服务器。鉴别访问控制器与鉴别服务器之间的会话密钥是利用其基于 ID 的公私钥生成的。
- g) 若鉴别访问控制器要求验证请求者的公钥是否被撤销,则利用随机数生成算法生成另一个鉴别访问控制器进行询问。以该鉴别访问控制器询问、请求者公钥的后两个字段和设置好的标识 FLAG 为内容生成消息 3,然后在鉴别访问控制器与鉴别服务器之间的会话密钥保护下发送给鉴别服务器;否则,执行 h) 操作。
- h) 利用预安装的 ECC 域参数生成用于 ECDH 交换的临时私钥 y 、临时公钥 $y \cdot P$,使用自己的临时私钥 y 和消息 2 中请求者的临时公钥 $x \cdot P$ 进行 ECDH 计算,得到主密钥种子 $(x \cdot y \cdot P)_{\text{abscissa}}$,对其进行扩展 KD-HMAC-SHA256 $((x \cdot y \cdot P)_{\text{abscissa}}, N_1 || N_2 || \text{"base key expansion for key and additional nonce"})$,生成 16 个八位位组的基密钥 BK。其中, N_1 是鉴别访问控制器询问, N_2 是请求者询问。然后鉴别访问控制器使用自己基于 ID 的私钥计算请求者询问 N_2 、对应公钥的后两个字段、临时公钥 $y \cdot P$ 和请求者的身份标识的签名,生成消息 5 发送给请求者。

B. 1. 4. 4 消息 3

若消息 2 中的标识 FLAG 指示要验证鉴别访问控制器的公钥撤销查询或鉴别访问控制器自己要求验证请求者的公钥撤销查询,则鉴别访问控制器向鉴别服务器发送消息 3, Message Type=3。

消息 3 格式见图 B. 13。

标识 FLAG	请求者 询问	鉴别访问控 制器询问	请求者公钥的 后两个字段	鉴别访问控制器公钥 的后两个字段
八位位组数:	1	32	32	3

图 B.13 消息 3 格式

其中:

- 标识 FLAG 字段长度为 1 个八位位组,定义如前。
- 请求者询问字段长度为 32 个八位位组。本字段值应与请求者发送的消息 2 中请求者询问字段值相同。当标识 FLAG 字段的比特 0 的值为 0,比特 1 的值为 1 时,该字段不存在。
- 鉴别访问控制器询问字段长度为 32 个八位位组。由鉴别访问控制器采用随机数生成算法生成,记为 N_3 。
- 请求者公钥的后两个字段的定义如前。当标识 FLAG 字段的比特 0 的值为 1,比特 1 的值为 0 时,该字段不存在。
- 鉴别访问控制器公钥的后两个字段的定义如前。当标识 FLAG 字段的比特 0 的值为 0,比特 1 的值为 1 时,该字段不存在。

鉴别服务器收到消息 3 后,进行如下处理:

- a) 检查标识 FLAG 字段的比特 0 和比特 1 的值,若比特 0 和比特 1 的值都为 1,则执行 b)操作;若比特 0 的值为 1 而比特 1 的值为 0,则执行 c)操作;若比特 0 的值为 0 而比特 1 的值为 1,则执行 d)操作。
- b) 鉴别服务器使用本地公钥的前两个字段合成请求者和鉴别访问控制器的公钥,然后查询公钥撤销表生成对应的公钥撤销结果。使用鉴别服务器基于 ID 的私钥对请求者询问、鉴别访问控制器的公钥和它的公钥撤销结果进行签名,以鉴别访问控制器询问、请求者的公钥撤销结果、鉴别访问控制器的公钥撤销结果和上述签名为内容生成消息 4,然后在鉴别访问控制器与鉴别服务器之间的会话密钥保护下发送给鉴别访问控制器。
- c) 鉴别服务器使用本地公钥的前两个字段合成鉴别访问控制器的公钥,然后查询公钥撤销表生成对应的公钥撤销结果。使用鉴别服务器基于 ID 的私钥对请求者询问、鉴别访问控制器的公钥和它的公钥撤销结果进行签名,以鉴别访问控制器询问、鉴别访问控制器的公钥撤销结果和上述签名为内容生成消息 4,然后在鉴别访问控制器与鉴别服务器之间的会话密钥保护下发送给鉴别访问控制器。
- d) 鉴别服务器使用本地公钥的前两个字段合成请求者的公钥,然后查询公钥撤销表生成对应的公钥撤销结果。以鉴别访问控制器询问和请求者的公钥撤销结果为内容生成消息 4,然后在鉴别访问控制器与鉴别服务器之间的会话密钥保护下发送给鉴别访问控制器。

B.1.4.5 消息 4

鉴别服务器完成消息 3 处理后,向鉴别访问控制器发送消息 4,Message Type=4。

消息 4 格式见图 B.14。

标识 FLAG	鉴别访问控制 器询问	请求者的公 钥撤销结果	鉴别访问控制器的 公钥撤销查询结果
八位位组数:	1	32	1 可变

图 B.14 消息 4 格式

其中:

- 标识 FLAG 字段长度为 1 个八位位组,定义如前。该字段值和消息 3 中的标识 FLAG 值相同。

- 鉴别访问控制器询问字段长度为 32 个八位位组。该字段值和消息 3 中的鉴别访问控制器询问字段的值相同。
- 请求者的公钥撤销结果字段长度为 1 个八位位组, 比特 0 有意义(值为 1 表示该公钥已被撤销, 值为 0 表示该公钥未被撤销)。当标识 FLAG 字段的比特 0 的值为 1, 比特 1 的值为 0 时, 该字段不存在。
- 鉴别访问控制器的公钥撤销查询结果字段定义如前。当标识 FLAG 字段的比特 0 的值为 0, 比特 1 的值为 1 时, 该字段不存在。

鉴别访问控制器收到消息 4 后, 进行如下处理:

- a) 先检查标识 FLAG 值是否与鉴别访问控制器发送的消息 3 中的标识 FLAG 值相同, 若不同, 则丢弃消息; 否则, 则检查标识 FLAG 字段的比特 0 和比特 1 的值, 若比特 0 和比特 1 的值都为 1, 则执行 b) 操作; 若比特 0 的值为 1 而比特 1 的值为 0, 则执行 c) 操作; 若比特 0 的值为 0 而比特 1 的值为 1, 则执行 d) 操作。
- b) 检查鉴别访问控制器询问与自己在消息 3 中的鉴别访问控制器询问是否相同, 若不同, 则丢弃该消息; 否则, 验证请求者的公钥撤销结果。若请求者的公钥已被撤销, 则断开与请求者的 TAEP-IPAP 连接; 否则, 执行 e) 操作后鉴别访问控制器使用自己基于 ID 的私钥计算请求者询问 N_2 、对应公钥的后两个字段、临时公钥 $y \cdot P$ 、请求者的身份标识和鉴别访问控制器的公钥撤销查询结果的签名, 生成消息 5 发送给请求者。
- c) 检查鉴别访问控制器询问与自己在消息 3 中的鉴别访问控制器询问是否相同, 若不同, 则丢弃该消息; 否则, 执行 e) 操作后鉴别访问控制器使用自己基于 ID 的私钥计算请求者询问 N_2 、对应公钥的后两个字段、临时公钥 $y \cdot P$ 、请求者的身份标识和鉴别访问控制器的公钥撤销查询结果的签名, 生成消息 5 发送给请求者。
- d) 检查鉴别访问控制器询问与自己在消息 3 中的鉴别访问控制器询问是否相同, 若不同, 则丢弃该消息; 否则, 验证请求者的公钥撤销结果。若请求者的公钥已被撤销, 则断开与请求者的 TAEP-IPAP 连接; 否则, 执行 e) 操作后鉴别访问控制器使用自己基于 ID 的私钥计算请求者询问 N_2 、对应公钥的后两个字段、临时公钥 $y \cdot P$ 和请求者的身份标识的签名, 生成消息 5 发送给请求者。
- e) 利用预安装的 ECC 域参数生成用于 ECDH 交换的临时私钥 y 、临时公钥 $y \cdot P$, 使用自己的临时私钥 y 和消息 2 中请求者的临时公钥 $x \cdot P$ 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 $((x \cdot y \cdot P)_{abscissa}, N_1 || N_2 || \text{“base key expansion for key and additional nonce”})$, 生成 16 个八位位组的基密钥 BK。其中, N_1 是鉴别访问控制器询问, N_2 是请求者询问。

B. 1. 4. 6 消息 5

鉴别访问控制器处理完成消息 4 或消息 2 后, 发送消息 5, Message Type=5, 格式见图 B. 15。

	标识 FLAG	请求者 询问	鉴别访问控制 器密钥数据	请求者的 身份标识	鉴别访问控制器公 钥的后两个字段	鉴别访问控制器的 公钥撤销查询结果	鉴别访问控 制器的签名
八位位组数:	1	32	可变	2	3	可变	可变

图 B. 15 消息 5 格式

其中:

- 标识 FLAG 字段长度为 1 个八位位组, 定义如前。该字段值比特 0 的值应与请求者发送的消息 2 中标识 FLAG 字段比特 0 的值相同。

- 请求者询问字段长度为 32 个八位位组。本字段值应与请求者发送的消息 2 中请求者询问字段值相同。
- 鉴别访问控制器密钥数据格式如前定义,内容是鉴别访问控制器生成的用于 ECDH 交换的临时公钥。
- 请求者的身份标识字段如前定义。
- 鉴别访问控制器公钥的后两个字段如前定义。
- 鉴别访问控制器的公钥撤销查询结果字段如前定义。当标识 FLAG 字段的比特 0 的值为 0 时,该字段不存在。
- 鉴别访问控制器的签名字段采用签名属性表示,其定义如前。它是本消息中除本字段之外所有数据字段的签名。

鉴别访问控制器收到消息 4,或收到消息 2 后,发送消息 5。

请求者收到消息 5 后,进行如下处理:

- a) 检查请求者询问字段值是否与请求者发送的消息 2 中的请求者询问值相同,若不同,则丢弃该消息;否则,检查请求者的身份标识字段值是否与自己的身份标识值相同。若不同,则丢弃该消息;否则执行 b)操作。
- b) 检查标识 FLAG 字段的比特 0 的值与自己发送的消息 2 中相应字段的值是否相同,若不同,则丢弃该消息;否则执行 c)操作。
- c) 检查鉴别访问控制器公钥的后两个字段的最后一个字段,若公钥的使用时限已逾期,则丢弃该消息;否则,执行 d)操作。
- d) 级联公钥的前两个字段和鉴别访问控制器公钥的后两个字段合成鉴别访问控制器的基于 ID 的公钥,使用该公钥和预安装的基于 ID 的公开参数验证鉴别访问控制器的签名是否正确,若不正确,则丢弃该消息;否则,执行 e)操作。
- e) 若标识 FLAG 的比特 0 的值为 0 时,则执行 f)操作;否则,执行 g)操作。
- f) 请求者使用自己的临时私钥 x 和鉴别访问控制器的临时公钥 $y \cdot P$ 进行 ECDH 计算,得到主密钥种子 $(x \cdot y \cdot P)_{\text{abscissa}}$,对其进行扩展 KD-HMAC-SHA256 $((x \cdot y \cdot P)_{\text{abscissa}}, N_1 || N_2 ||$ “base key expansion for key and additional nonce”),生成 16 个八位位组的基密钥 BK。其中, N_1 是鉴别访问控制器询问, N_2 是请求者询问。
- g) 检查鉴别访问控制器的公钥撤销查询结果字段的第三个子字段,若公钥撤销结果显示鉴别访问控制器的公钥已被撤销,则请求者断开与鉴别访问控制器的 TAEP-IPAP 连接;否则,使用请求者询问、鉴别访问控制器的公钥撤销结果、鉴别访问控制器的公钥和基于 ID 的公开参数验证该字段的第四个子字段是否正确(验证鉴别服务器的签名是否正确)。若签名验证失败,则丢弃该消息;否则,执行 f)操作。

B. 1. 4. 7 公钥算法说明

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

- a) 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。
- b) 临时公钥 $x \cdot P, y \cdot P$ 是椭圆曲线域参数定义的椭圆曲线上的点。
- c) ECDH 协商出来密钥种子 $(x \cdot y \cdot P)_{\text{abscissa}}$ 是 $x \cdot y \cdot P$ 的 x 坐标, $x \cdot y \cdot P$ 不能是无穷远点。

B. 2 基于证书的身份鉴别协议 TAEP-CBAP(TAEP-Certificate Based Authentication Protocol)

B. 2. 1 概述

基于证书鉴别协议也就是基于公钥证书体制的鉴别协议。

TAEP-CBAP 过程见图 B. 16。

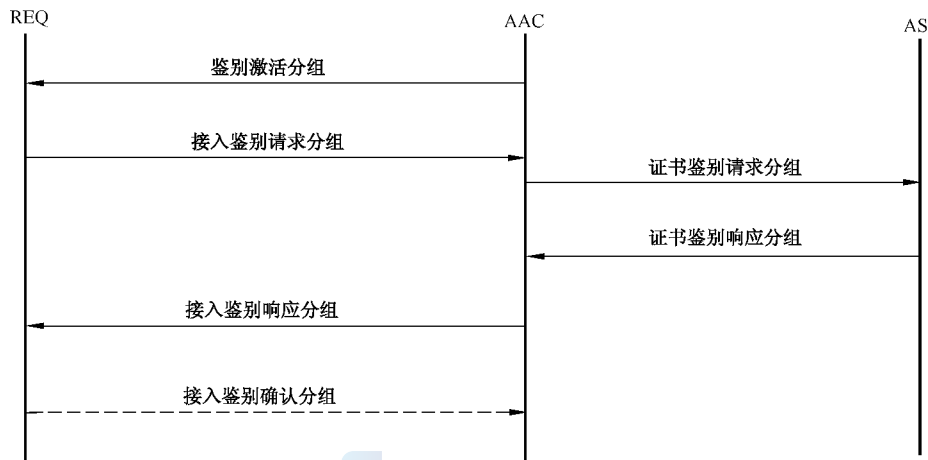


图 B. 16 TAEP-CBAP 过程

TAEP-CBAP 各分组的主要组成见图 B. 17。

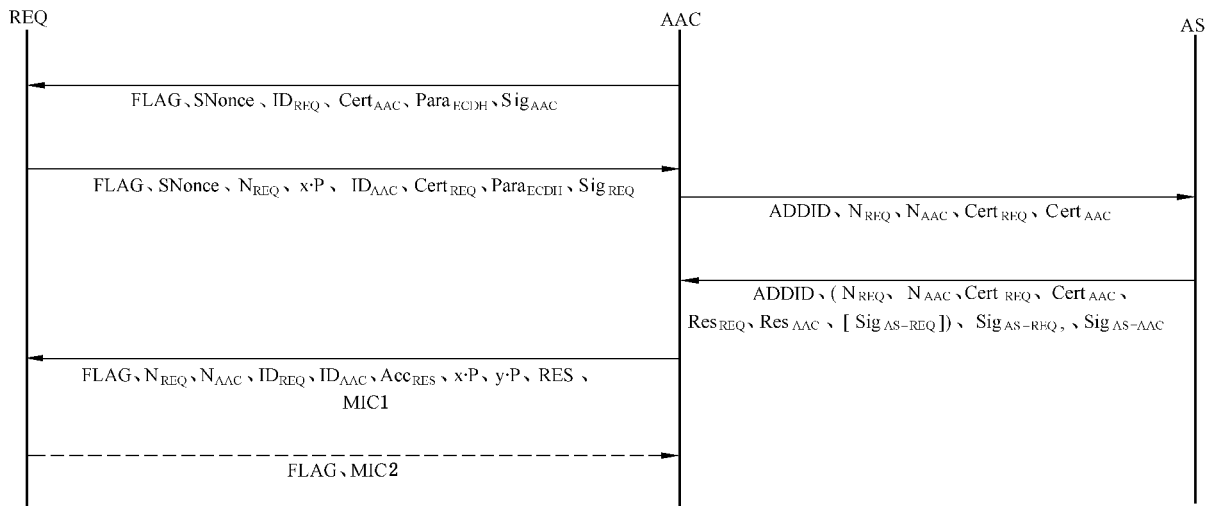


图 B. 17 TAEP-CBAP 各分组的主要组成示意

B. 2. 2 TAEP-CBAP 协议分组格式

TAEP-CBAP 协议是基于证书的鉴别协议,其分组封装格式见图 B. 18。

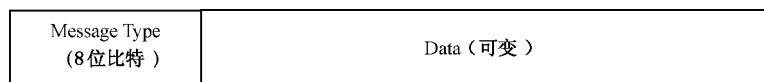


图 B. 18 TAEP-CBAP 协议分组格式

其中:

——Message Type 类型:定义见表 B. 2。

表 B.2 TAEP-CBAP 类型分组 Message Type 子类型定义

Message Type 值	分组定义
1	鉴别激活分组
2	接入鉴别请求分组
3	证书鉴别请求分组
4	证书鉴别响应分组
5	接入鉴别响应分组
6	接入鉴别确认分组
其他	保留

——Data 数据:每个分组的数据 Data 将在协议分组介绍中进行介绍。其数据元素封装格式见图 B.19。

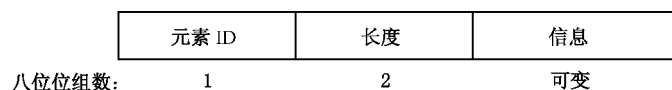


图 B.19 数据信息元素封装格式

其中:

- 元素 ID 字段:表示元素在每一个分组中的元素身份标识。每个分组根据分组内包含的元素集合给每个元素分配一个分组内唯一的元素 ID。元素 ID 字段长为 1 个八位位组。
- 长度字段:表示元素信息字段的八位位组数。长度字段长为 2 个八位位组。
- 信息字段:表示元素的内容,对应于每个元素,其长度也不同,这将在每个分组中介绍元素时进行具体介绍。

B.2.3 固定格式定义

B.2.3.1 标识 FLAG

FLAG 格式见图 B.20。



图 B.20 FLAG 格式

其中:

- 标识 ID 字段表示标识的来源,长度为 2 个八位位组。标识 ID 定义如下:
 - 0 表示该字段的标识来自 REQ;
 - 1 表示该字段的标识来自 AAC;
 - 其他值保留。
- 长度字段为鉴别数据字段的八位位组数。
- 标识 FLAG 字段为该属性字段的内容。

B.2.3.2 鉴别标识 SNonce

SNonce 格式见图 B.21。



图 B.21 SNonce 格式

其中:

——标识 ID 字段表示鉴别标识的来源,长度为 2 个八位位组。标识 ID 定义如下:

0 表示该字段的标识来自 REQ;

1 表示该字段的标识来自 AAC;

其他值保留。

——鉴别长度字段为鉴别数据字段的八位位组数。

——鉴别标识字段为该属性字段的内容。

B.2.3.3 证书

证书格式见图 B.22。



图 B.22 证书格式

其中:

——标识 ID 字段表示证书的所有者身份,长度为 2 个八位位组。标识 ID 定义如下:

0 表示该字段的标识来自 REQ;

1 表示该字段的标识来自 AAC;

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。

B.2.3.4 Para_{ECDH}

Para_{ECDH} 格式见图 B.23。

图 B.23 Para_{ECDH} 格式

其中:

——标识 ID 字段表示标识的来源,长度为 2 个八位位组。标识 ID 定义如下:

0 表示该字段的标识来自 REQ;

1 表示该字段的标识来自 AAC;

其他值保留。

——长度字段为鉴别数据字段的八位位组数。

——参数字段为该属性字段的内容。

B.2.3.5 数字签名 Sig

数字签名 Sig 格式见图 B.24。



图 B.24 数字签名 Sig 格式

其中:

- 标识 ID 字段表示鉴别标识的来源,长度为 2 个八位位组。标识 ID 定义如下:
 - 0 表示该字段的标识来自 REQ;
 - 1 表示该字段的标识来自 AAC;
 - 其他值保留。
- 长度字段为鉴别数据字段的八位位组数。
- 数字签名字段为该属性字段的内容。

B.2.3.6 身份 ID

身份 ID 格式见图 B.25。



图 B.25 身份 ID 格式

其中:

- 标识 ID 字段表示鉴别标识的来源,长度为 2 个八位位组。标识 ID 定义如下:
 - 0 表示该字段的标识来自 REQ;
 - 1 表示该字段的标识来自 AAC;
 - 其他值保留。
- 长度字段为鉴别数据字段的八位位组数。
- 身份 ID 字段为该属性字段的内容。

B.2.3.7 ADDID

身份索引,长度为 12 个八位位组。ADDID 格式见图 B.26。

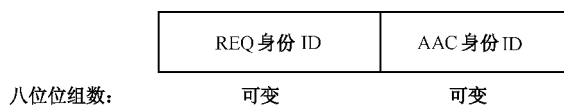


图 B.26 ADDID 格式

B.2.4 鉴别激活分组

B.2.4.1 概述

鉴别方法协商过程成功完成后, AAC 向 REQ 发送鉴别激活分组。鉴别激活分组主要包括:标识 FLAG、一次性随机数 SNonce(标识鉴别的新鲜性,亦称鉴别标识)、本地 ASU 的身份、AAC 的证书 Cert_{AAC}、Para_{ECDH} 及 AAC 的签名 Sig_{AAC}。为了方便 REQ 选择鉴别证书,还可包含 AAC 所信任的 ASU 信息。该分组封装在 TAEP-Request-CBAP/TAEPoL 中, Message Type=1。

鉴别激活分组信息字段被定义为包含 1 个八位位组的元素 ID 字段、1 个八位位组的长度字段以及可变长度的特定元素信息字段的通用格式,如图 B.27 所示。

	元素 ID	长度	信息
八位位组数:	1	1	长度

图 B.27 元素封装格式

鉴别激活分组有效元素的集合见表 B.3。



表 B.3 鉴别激活分组有效元素集合

信息元素	元素 ID
标识 FLAG 元素	0
鉴别标识 SNonce 元素	1
本地 REQ 的身份元素	2
AAC 的证书 Cert _{AAC} 元素	3
Para _{ECDH} 元素	4
AAC 的签名 Sig _{AAC} 元素	5
保留	6
保留	7~15
保留	16~255

B.2.4.2 标识 FLAG 元素

标识 FLAG 信息字段长度为 1 个八位位组。其中,标识 FLAG 字段定义见 B.2.3.1。

B.2.4.3 鉴别标识 SNonce 元素

鉴别标识 SNonce 信息字段定义见 B.2.3.2。

B.2.4.4 本地 REQ 的身份元素

本地 REQ 的身份信息字段长度为可变个八位位组了,定义见 B.2.3.6。

B.2.4.5 AAC 的证书 Cert_{AAC} 元素

AAC 的证书 Cert_{AAC} 信息字段长度为可变个八位位组。表示作为 AAC 实体的证书。定义见 B.2.3.3。

B.2.4.6 Para_{ECDH} 元素

ECDH 参数信息字段长度为可变个八位位组,定义见 B.2.3.4。

B.2.4.7 AAC 的签名 Sig_{AAC} 元素

AAC 的签名 Sig_{AAC} 信息字段长度为可变个八位位组,定义见 B.2.3.5。

AAC 向 REQ 实体发送鉴别激活分组激活 REQ 实体进行双向证书鉴别。

REQ 实体接收到由 AAC 发送的鉴别激活分组后,进行如下处理:

- REQ 实体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 b)操作;
- REQ 实体根据鉴别激活分组中的 AAC 信任的 REQ 身份选择由该 REQ 实体颁发的证书或

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 $x \cdot P$ 和 32 个八位位组的 REQ 询问 NREQ(随机数),生成接入鉴别请求分组,发送给 AAC。

B.2.5 接入鉴别请求分组

B.2.5.1 概述

REQ 收到鉴别激活分组后,验证 AAC 的签名 Sig_{AAC} ,验证通过后构造接入鉴别请求分组发送给 AAC,否则直接丢弃鉴别激活分组。

接入鉴别请求分组主要包括:标识 FLAG、鉴别激活分组中的一次性随机数 SNonce、本地生成的一次性随机数 N_{REQ} (用于生成下一次鉴别标识值的部分资料)、本地生成的临时公钥 $x \cdot P$ 、AAC 的身份信息 ID_{AAC} (可根据鉴别激活分组中的 AAC 公钥证书 $Cert_{AAC}$ 得到)、REQ 选择的鉴别证书 $Cert_{REQ}$ 、 $Para_{ECDH}$ 及签名信息 Sig_{REQ} (利用 REQ 的长期私钥计算得到)。若 REQ 所信任的 AS 不止一个,该分组还可包含 REQ 所信任的 AS 列表,方便 AS 系统的鉴别。

接入鉴别请求分组封装在 TAEP-Request-CBAP/TAEPoL 中,Message Type=2。

接入鉴别请求分组信息字段被定义为包含 1 个八位位组的元素 ID 字段、1 个八位位组的长度字段以及可变长度的特定元素信息字段的通用格式。本部分为每个元素分配唯一的元素 ID,长度字段规定了信息字段中的八位位组数,各元素的封装格式定义见图 B.28。

	元素 ID	长度	信息
八位位组数:	1	1	长度

图 B.28 元素封装格式

接入鉴别请求分组的有效元素集合见表 B.4。

表 B.4 接入鉴别请求分组有效元素集合

信息元素	元素 ID
标识 FLAG 元素	0
鉴别标识 SNonce 元素	1
本地询问 N_{REQ} 元素	2
临时公钥 $x \cdot P$ 元素	3
AAC 的身份 ID_{AAC} 元素	4
鉴别证书 $Cert_{REQ}$ 元素	5
$Para_{ECDH}$ 元素	6
REQ 信任 ASU 列表元素	7
签名 Sig_{REQ} 元素	8
保留	9~15
保留	16~255

B.2.5.2 标识 FLAG 元素

标识 FLAG 字段定义同 B.2.3.1。其中,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 AAC 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 REQ 要求验证 AAC 证书的有效性,为 0 表示不需要验证

AAC 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。

B.2.5.3 鉴别标识 SNonce 元素

鉴别标识 SNonce 字段定义同 B.2.3.2。本字段值应与 AAC 发送的鉴别激活分组中鉴别标识字段值相同。

B.2.5.4 本地询问 N_{REQ} 元素

本地询问 N_{REQ} 字段长度为 32 个八位位组,本地询问 N_{REQ} 元素格式见图 B.29。

	元素 ID	长度	本地 N _{REQ}
八位位组数:	1	1	32

图 B.29 N_{REQ} 元素格式

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

B.2.5.5 临时公钥 x, P 元素

临时公钥 x, P 元素格式见图 B.30。

	元素 ID	长度	临时公钥 x,P
八位位组数:	1	1	可变

图 B.30 临时公钥 x, P 元素格式

临时公钥 x, P 字段长度为可变个八位位组,临时公钥 x, P 的内容是 REQ 生成的用于 ECDH 交换的临时公钥。

B.2.5.6 AAC 的身份 ID_{AAC} 元素

AAC 的身份 ID_{AAC} 字段长度为可变个八位位组,格式见图 B.31。

	元素 ID	长度	身份 ID
八位位组数:	1	1	可变

图 B.31 AAC 的身份格式

AAC 的身份 ID_{AAC} 字段长度为可变个八位位组,身份 ID 定义见 B.2.3.6。

B.2.5.7 鉴别证书 Cert_{REQ} 元素

REQ 的证书 Cert_{REQ} 字段长度为可变个八位位组。表示作为 REQ 实体的证书,格式见图 B.32。

	元素 ID	长度	证书
八位位组数:	1	1	可变

图 B.32 REQ 的证书格式

其中,证书字段定义见 B.2.3.3。REQ 选择的鉴别证书 Cert_{REQ} 字段表示作为 REQ 实体的站的证书。

B.2.5.8 Para_{ECDH} 元素

ECDH 参数字段定义同 B.2.3.4。ECDH 参数字段和鉴别激活分组中的 ECDH 参数字段相同。

B.2.5.9 REQ 信任 ASU 列表元素

REQ 信任 ASU 列表字段长度为可变个八位位组,格式见图 B.33。

	元素 ID	长度	REQ 信任 ASU 列表
八位位组数:	1	1	可变

图 B.33 REQ 信任 ASU 列表格式

REQ 信任的 ASU 列表字段,该字段为可选字段,采用身份列表属性表示。内容包含 REQ 实体信任的服务器,但不包含 REQ 的证书颁发者。若 REQ 实体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。

REQ 的签名 Sig_{REQ} 字段长度为可变个八位位组,格式见图 B.34。

	元素 ID	长度	签名 Sig_{REQ}
八位位组数:	1	1	可变

图 B.34 REQ 的签名 Sig_{REQ} 格式

其中,签名 Sig_{REQ} 字段定义见 B.2.3.5。

它采用签名属性表示,是对本分组中除本字段之外所有数据字段的签名。

当 REQ 实体收到 AAC 的鉴别激活分组或 REQ 实体需要进行鉴别时,REQ 实体发送接入鉴别请求分组给 AAC。AAC 收到 REQ 实体发来的接入鉴别请求分组后,进行如下处理:

- 如果 AAC 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 AAC 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 AAC 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。
- 检查 REQ 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 REQ 实体的签名,若验证不通过,则丢弃该分组;若标识字段的比特 2 为 1 或 AAC 的本地策略要求使用 AS 鉴别 REQ 实体的证书,则 AAC 生成证书鉴别请求分组,发往 AS;否则执行 c) 操作。
- AAC 本地鉴别 REQ 实体的证书,即根据本地缓存的 REQ 实体证书的验证结果及其根据本地策略所定义的时效性确定 REQ 实体证书的验证结果。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 $x \cdot P$ 和 32 个八位位组的 AAC 询问 N_{REQ} (随机数),使用自己的临时私钥 y 和接入鉴别请求分组中 REQ 实体的临时公钥 $x \cdot P$ 进行 ECDH 计算,得到主密钥种子 $(x \cdot y \cdot P)_{\text{abscissa}}$,对其进行扩展 KD-HMAC-SHA256 $((x \cdot y \cdot P)_{\text{abscissa}}, N_{\text{AAC}} || N_{\text{REQ}} || \text{“base key expansion for key and additional nonce”})$,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 REQ 实体。若 REQ 实体证书鉴别结果不成功,AAC 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。AAC 的询问 N_{AAC} 和 AAC 的密钥数据(AAC 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 REQ 实体,然后解除与 REQ 实体的链路验证。

B.2.6 证书鉴别请求分组

B.2.6.1 概述

AAC 收到接入鉴别请求分组后,验证 S_{Nonce} 、 ID_{AAC} 、签名信息 Sig_{REQ} ,验证通过后构造证书鉴别

请求分组并发往 AS, 否则直接丢弃接入鉴别请求分组。

证书鉴别请求分组主要包括: 地址索引 ADDID(标识鉴别所服务的主体)、接入鉴别请求分组中的 N_{REQ} 、本地生成的一次性随机数 N_{AAC} (用于生成下一次鉴别标识值的部分资料)、 $Cert_{REQ}$ 和 $Cert_{AAC}$ 。若接入鉴别请求分组中包含 REQ 信任的 ASU 列表, 则证书鉴别请求分组中也应包含该信息。

证书鉴别请求分组封装在 TAEP-Request-CBAP/TAEP-AS-SVC 中, Message Type=3。

证书鉴别请求分组信息字段被定义为包含 1 个八位位组的元素 ID 字段、1 个八位位组的长度字段以及可变长度的特定元素信息字段的通用格式。本部分为每个元素分配唯一的元素 ID, 长度字段规定了信息字段中的八位位组数, 各元素的封装格式定义见图 B. 35。

	元素 ID	长度	信息
八位位组数:	1	1	长度

图 B. 35 元素封装格式

证书鉴别请求分组有效元素集合如表 B. 5 所示。

表 B. 5 证书鉴别请求分组有效元素集合

信息元素	元素 ID
ADDID 元素	0
AAC 询问 N_{AAC} 元素	1
REQ 询问 N_{REQ} 元素	2
REQ 的证书 $Cert_{REQ}$ 元素	3
AAC 的证书 $Cert_{AAC}$ 元素	4
REQ 信任 ASU 列表元素	5
保留	6
保留	7~15
保留	32~255

B. 2. 6. 2 ADDID 元素

ADDID 字段长度为 12 个八位位组, 格式见图 B. 36。

	元素 ID	长度	ADDID
八位位组数:	1	1	12

图 B. 36 ADDID 元素

其中, ADDID 字段定义见 B. 2. 3. 7。

B. 2. 6. 3 AAC 询问 N_{AAC} 元素

AAC 询问 N_{AAC} 字段长度为 32 个八位位组, 格式见图 B. 37。

	元素 ID	长度	AAC 询问 N_{AAC}
八位位组数:	1	1	32

图 B. 37 AAC 询问 NAAC 元素格式

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

B.2.6.4 REQ 询问 N_{REQ} 元素

REQ 询问 N_{REQ} 字段定义同 B.2.5.4。本字段值应与 REQ 实体发送的接入鉴别请求分组中 REQ 询问字段值相同。

B.2.6.5 REQ 的证书 $Cert_{REQ}$ 元素

REQ 的证书 $Cert_{REQ}$ 字段定义同 B.2.5.7。该字段和接入鉴别请求分组中 REQ 的证书字段相同。

B.2.6.6 AAC 的证书 $Cert_{AAC}$ 元素

AAC 的证书 $Cert_{AAC}$ 字段定义同 B.2.4.5。内容包含 AAC 的证书。

B.2.6.7 REQ 信任 ASU 列表元素

REQ 信任 ASU 列表字段定义同 B.2.5.9, 本字段值应与 REQ 实体发送的接入鉴别请求分组中的 REQ 信任的 ASU 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 AAC 自己需要进行证书验证, AAC 向 AS 发送证书鉴别请求分组。

AAC 接收到 REQ 实体发送的接入鉴别请求分组并向 AS 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 REQ 实体发送的接入鉴别请求进行处理。

B.2.7 证书鉴别响应分组元素

B.2.7.1 概述

AS 收到证书鉴别请求分组后, 验证 $Cert_{REQ}$ 和 $Cert_{AAC}$ 证书的有效性, 构造证书鉴别响应分组返回给 AAC。

证书鉴别响应分组主要包括: 地址索引 ADDID、证书验证结果及 REQ 信任的 AS 对证书验证结果的签名 Sig_{AS-REQ} 、AAC 所信任的 AS 的签名 Sig_{AS-AAC} 。其中证书验证结果包含 N_{REQ} 、 N_{AAC} 、 $Cert_{REQ}$ 、 $Cert_{AAC}$ 、REQ 证书的验证结果 Res_{REQ} 与 AAC 证书的验证结果 Res_{AAC} 。若 AAC 和 REQ 信任的 AS 相同, 则证书鉴别相应分组中的签名只存在一个。

证书鉴别响应分组封装在 TAEP-Response-CBAP/TAEP-AS-SVC 中, Message Type=4。

证书鉴别响应分组信息字段被定义为包含 1 个八位位组的元素 ID 字段、1 个八位位组的长度字段以及可变长度的特定元素信息字段的通用格式。本部分为每个元素分配唯一的元素 ID, 长度字段规定了信息字段中的八位位组数, 各元素的封装格式定义如图 B.38 所示。

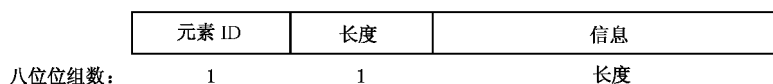


图 B.38 元素封装格式

证书鉴别响应分组有效元素集合见表 B.6。

表 B.6 证书鉴别响应分组有效元素集合

信息元素	元素 ID
ADDID 元素	0
证书的验证结果元素	1
REQ 信任的服务器签名 Sig _{AS-REQ} 元素	2
AAC 信任的服务器签名 Sig _{AS-AAC} 元素	3
保留	4~15
保留	32~255

B.2.7.2 ADDID 元素

ADDID 字段长定义同 B.2.6.2。该字段值和证书鉴别请求分组中的 ADDID 字段的值相同。

B.2.7.3 证书的验证结果元素

证书的验证结果字段长度为可变个八位位组,格式见图 B.39。

元素 ID	长度	证书的验证结果
1	1	可变

八位位组数:

图 B.39 证书的验证结果元素

证书的验证结果字段采用证书验证结果属性表示。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 AAC 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 REQ 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 REQ 证书,第二个证书及结果对应于证书鉴别请求分组中的 AAC 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。

B.2.7.4 REQ 信任的服务器签名 Sig_{AS-REQ} 元素

REQ 信任的服务器签名 Sig_{AS-REQ} 字段长度为可变个八位位组,格式见图 B.40。

元素 ID	长度	REQ 信任的服务器签名 Sig _{AS-REQ}
1	1	可变

八位位组数:

图 B.40 REQ 信任的服务器签名 Sig_{AS-REQ} 字段

其中 REQ 信任的服务器签名 Sig_{AS-REQ} 字段定义见 B. 2. 3. 5。它对本分组中证书的验证结果字段的签名。

B. 2. 7. 5 AAC 信任的服务器签名 Sig_{AS-AAC} 元素

AAC 信任的服务器签名 Sig_{AS-AAC} 字段长度为可变个八位位组, 格式见图 B. 41。

元素 ID	长度	REQ 信任的服务器签名 Sig_{AS-AAC}
八位位组数: 1	1	可变

图 B. 41 AAC 信任的服务器签名 Sig_{AS-AAC} 字段格式

其中 AAC 信任的服务器签名 Sig_{AS-AAC} 字段定义见 B. 2. 3. 5。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 REQ 信任的服务器和 AAC 信任的服务器为同一个, 即 REQ 信任的服务器和 AAC 信任的服务器的身属性相同, 则证书鉴别响应分组中 REQ 信任的服务器签名字段和 AAC 信任的服务器签名字段只存在一个; 若 REQ 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 REQ 信任的服务器签名字段。

AS 收到证书鉴别请求分组后, 向 AAC 发送证书鉴别响应分组。

AAC 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 AAC 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) AAC 查找自身所信任的 AS 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 REQ 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 $y \cdot P$, 使用自己的临时私钥 y 和 REQ 的临时公钥 $x \cdot P$ 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 $((x \cdot y \cdot P)_{abscissa}, N_{AAC} || N_{REQ} || \text{“base key expansion for key and additional nonce”})$, 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 REQ 实体。若接入鉴别请求分组中 REQ 要求验证 AAC 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 REQ 证书鉴别结果不成功, AAC 设定接入结果为不成功, AAC 的询问 N_{AAC} 和 AAC 的密钥数据(AAC 的临时公钥)可设置任意值。构造接入鉴别响应分组发送给 REQ, 然后解除与 REQ 实体的链路验证。若接入鉴别请求分组中 REQ 实体要求验证 AAC 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。

B. 2. 8 接入鉴别响应分组

B. 2. 8. 1 概述

AAC 收到证书鉴别响应分组后, 验证 N_{REQ} 、 N_{AAC} 、 Sig_{AS-AAC} 等, 若无效, 则直接丢弃, 否则构造接入鉴别响应分组返回给 REQ。

接入鉴别响应分组主要包括：标识 FLAG、 N_{REQ} 、 N_{AAC} 、 ID_{REQ} （可根据 $Cert_{REQ}$ 得到）、接入结果、 ID_{AAC} 、REQ 密钥数据 $x \cdot P$ 、本地生成的临时公钥 $y \cdot P$ 、复合的证书验证结果 RES 及 AAC 生成的 MIC1。其中复合的证书验证结果 RES 为证书鉴别响应分组中除 ADDID 之外的其他内容。

接入鉴别响应分组封装在 TAEP-Response-CBAP/TAEPoL 中，Message Type=5。

接入鉴别请求分组信息字段被定义为包含 1 个八位位组的元素 ID 字段、1 个八位位组的长度字段以及可变长度的特定元素信息字段的通用格式。本部分为每个元素分配唯一的元素 ID，长度字段规定了信息字段中的八位位组数，各元素的封装格式定义见图 B.42。

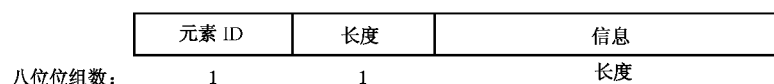


图 B.42 元素封装格式

接入鉴别响应分组有效元素集合见表 B.7。

表 B.7 接入鉴别响应分组有效元素集合

信息元素	元素 ID
标识 FLAG 元素	0
REQ 询问 N_{REQ} 元素	1
AAC 询问 N_{AAC} 元素	2
接入结果 Acc_{RES} 元素	3
REQ 密钥数据 $x \cdot P$ 元素	4
AAC 密钥数据 $y \cdot P$ 元素	5
AAC 的身份 ID_{AAC} 元素	6
REQ 身份 ID_{REQ} 元素	7
证书验证结果 RES 元素	8
MIC1 元素	9
保留	10~15
保留	32~255

B.2.8.2 标识 FLAG 元素

标识 FLAG 字段定义同 B.2.3.1。比特 0、1、3 有意义。本字段比特 0、比特 1 应与 REQ 实体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 REQ 实体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。

B.2.8.3 REQ 询问 N_{REQ} 元素

REQ 询问 N_{REQ} 字段定义同 B.2.5.4。本字段值应与 REQ 实体发送的鉴别接入请求分组中 REQ 实体的询问字段值相同。

B.2.8.4 AAC 询问 N_{AAC} 元素

AAC 询问 N_{AAC} 字段定义同 B.2.6.3。本字段值应与 AAC 发送的证书鉴别请求分组中 AAC 的询

问字段值相同。

B.2.8.5 接入结果 RES 元素

格式见图 B.43。

	元素 ID	长度	RES
八位位组数:	1	1	1

图 B.43 接入结果 RES 元素

接入结果字段的长度为 1 个八位位组。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。

B.2.8.6 REQ 密钥数据 $x \cdot P$ 元素

REQ 密钥数据 $x \cdot P$ 字段长度为可变个八位位组,格式见图 B.44。

	元素 ID	长度	REQ 密钥数据 $x \cdot P$
八位位组数:	1	1	可变

图 B.44 REQ 密钥数据 $x \cdot P$ 字段

REQ 密钥数据是 REQ 实体生成的用于 ECDH 交换的临时公钥,本字段值应与 REQ 实体发送的鉴别接入请求分组中 REQ 实体密钥数据字段值相同。

B.2.8.7 AAC 密钥数据 $y \cdot P$ 元素

AAC 密钥数据 $y \cdot P$ 字段长度为可变个八位位组,格式见图 B.45。

	元素 ID	长度	AAC 密钥数据 $y \cdot P$
八位位组数:	1	1	可变

图 B.45 AAC 密钥数据 $y \cdot P$ 元素

内容是 AAC 生成的用于 ECDH 交换的临时公钥。

B.2.8.8 AAC 的身份 ID_{AAC} 元素

AAC 的身份 ID_{AAC} 字段定义同 B.2.5.6。

B.2.8.9 REQ 身份 ID_{REQ} 元素

REQ 的身份 ID_{REQ} 字段定义同 B.2.4.4。

B.2.8.10 证书验证结果 RES 元素

证书验证结果 RES 字段长度为可变个八位位组,格式见图 B.46。

	元素 ID	长度	证书验证结果 RES
八位位组数:	1	1	可变

图 B.46 证书验证结果 RES 元素

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。

B.2.8.11 MIC1 元素

B.2.8.11.1 MIC1 元素格式见图 B.47。

	元素 ID	长度	MIC1
八位位组数:	1	1	可变

图 B.47 MIC1 元素格式

MIC1 字段的计算过程如下:

- AAC 使用自己的临时私钥 y 和接入鉴别请求分组中 REQ 的临时公钥 $x \cdot P$ 进行 ECDH 计算,得到主密钥种子 $(x \cdot y \cdot P)_{\text{abscissa}}$;
- AAC 对主密钥种子进行扩展 KD-HMAC-SHA256 $((x \cdot y \cdot P)_{\text{abscissa}}, N_{\text{AAC}} || N_{\text{REQ}} || \text{“base key expansion for key and additional nonce”})$, 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;
- 利用 BK 对接入鉴别响应分组中除 MIC1 字段外的所有字段计算的杂凑值作为 MIC1 的值。

B.2.8.11.2 REQ 实体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。REQ 实体收到接入鉴别响应分组后,进行如下处理:

- 根据 ID_{AAC} 和 ID_{REQ} 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;
- 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;
- 比较 REQ 实体的询问与 AAC 在接入鉴别请求分组中发送的 REQ 实体询问是否相同、比较 REQ 密钥数据与 REQ 发送的鉴别接入请求分组中 REQ 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;
- 验证 AAC 的签名是否正确,若不正确,则丢弃该接入鉴别响应分组;如果签名正确,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 AAC 的链路验证;否则执行 e) 操作;
- 若 REQ 实体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 REQ 实体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 AS 签名,若不正确,则丢弃该接入鉴别响应分组;否则检查 AAC 证书的鉴别结果是否为有效,若无效,解除与该 AAC 的链路验证;若有效,则执行 f) 操作;
- REQ 实体使用自己的临时私钥 x 和 AAC 的临时公钥 $y \cdot P$ 进行 ECDH 计算,得到密钥种子 $(x \cdot y \cdot P)_{\text{abscissa}}$, 将其扩展为 KD-HMAC-SHA256 $((x \cdot y \cdot P)_{\text{abscissa}}, N_{\text{AAC}} || N_{\text{REQ}} || \text{“base key expansion for key and additional nonce”})$, 以生成长度为 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。

B.2.8.11.3 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:

- 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;
- 临时公钥 $x \cdot P, y \cdot P$ 是椭圆曲线域参数定义的椭圆曲线上的点;
- ECDH 协商出来密钥种子 $(x \cdot y \cdot P)_{\text{abscissa}}$ 是 $x \cdot y \cdot P$ 的 x 坐标, $x \cdot y \cdot P$ 不能是无穷远点。

B.2.9 接入鉴别确认分组

B.2.9.1 概述

REQ 收到接入鉴别响应分组后,验证 $N_{\text{REQ}}, ID_{\text{REQ}}, Sig_{\text{AS-REQ}}$ 等,若无效,则直接丢弃,否则构造接入鉴别确认分组返回给 AAC。接入鉴别确认分组主要包括:标识 FLAG 及 MIC2。

接入鉴别确认分组封装在 TAEP-Response-CBAP/TAEPoL 中,Message Type=6。

接入鉴别确认分组信息字段被定义为包含 1 个八位位组的元素 ID 字段、1 个八位位组的长度字段以及可变长度的特定元素信息字段的通用格式。本部分为每个元素分配唯一的元素 ID,长度字段规定了信息字段中的八位位组数,各元素的封装格式定义见图 B.48。

	元素 ID	长度	信息
八位位组数:	1	1	长度

图 B.48 元素封装格式

接入鉴别响应分组有效元素集合见表 B.8。

表 B.8 接入鉴别响应分组有效元素集合

信息元素	元素 ID
标识 FLAG 元素	0
MIC2 元素	1
保留	2~15
保留	16~255

B.2.9.2 标识 FLAG 元素

标识 FLAG 字段定义同 B.2.8.2。

B.2.9.3 MIC2 元素

MIC2 元素格式见图 B.49。

	元素 ID	长度	MIC2
八位位组数:	1	1	可变

图 B.49 MIC2 元素格式

MIC2 字段的计算过程为:

- a) REQ 使用自己的临时私钥 x 和接入鉴别响应分组中 AAC 的临时公钥 $y \cdot P$ 进行 ECDH 计算,得到主密钥种子 $(x \cdot y \cdot P)_{\text{abscissa}}$;
- b) REQ 对主密钥种子进行扩展 KD-HMAC-SHA256 $((x \cdot y \cdot P)_{\text{abscissa}}, N_{\text{AAC}} || N_{\text{REQ}} || \text{“base key$

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；

c) 利用 BK 对接入鉴别确认分组中除 MIC2 字段外的所有字段计算的杂凑值作为 MIC2 的值。

AAC 根据证书鉴别响应分组中 REQ 证书的验证结果 Res_{REQ} 及本地策略，决定是否允许客户端 REQ 接入，再向 REQ 发送鉴别结果分组 TAEP-Success 或 TAEP-Failure。

REQ 收到接入鉴别响应分组后，验证 Sig_{AAC} 及其他字段，若无效，则直接丢弃；否则，根据 AAC 证书的验证结果 Res_{AAC} 及本地策略，决定是否接入该网络，若拒绝接入网络，可向 AAC 发送 TAEPoL-Logoff。

基于证书的核心鉴别过程成功完成后，AAC 和 REQ 分别根据对方的临时公钥与本地的临时私钥进行 ECDH 计算，并将结果与双方生成的一次性随机数 N_{AAC} 、 N_{REQ} 通过 HASH 计算得到下一次鉴别标识 S_{Nonce} ，用于鉴别更新时双方的同步锁定。

需要说明几点：

- 基于证书的核心鉴别过程支持双向与单向鉴别，若使用单向鉴别，则 AS 无需对 AAC 的证书 $Cert_{AAC}$ 进行验证，即证书鉴别请求、证书鉴别响应及接入鉴别响应等分组中均不包含 $Cert_{AAC}$ 、 Res_{AAC} 等字段。
- TAEP-CBAP 协议功能高度集中，既可实现双向鉴别，又可实现单向鉴别，还支持鉴别更新以及简化的鉴别更新。所谓简化的鉴别更新就是指不需要 AS 验证证书，AAC 与 REQ 之间的直接签名验证，反映在分组中的内容上，就是接入鉴别响应分组中不包含复合的证书验证结果字段。简化的鉴别过程只能用作鉴别更新过程，不能用作客户端与网络连接时的首次鉴别。

附录 C
(资料性附录)

适用于无线城域网的 TAAA 机制

C.1 引入 TAAA 机制后的安全子层架构

三元接入鉴别与授权 TAAA(Tri-element Access Authentication and Authorization)机制提供了从 BS 到 SS 上密钥数据的安全分发。通过这个 TAAA 机制,实现了 SS 和 BS 之间的密钥数据同步,此外,BS 还利用该协议加强了对网络业务的有条件访问。图 C.1 显示了引入 TAAA 机制后系统的安全组件的协议栈。

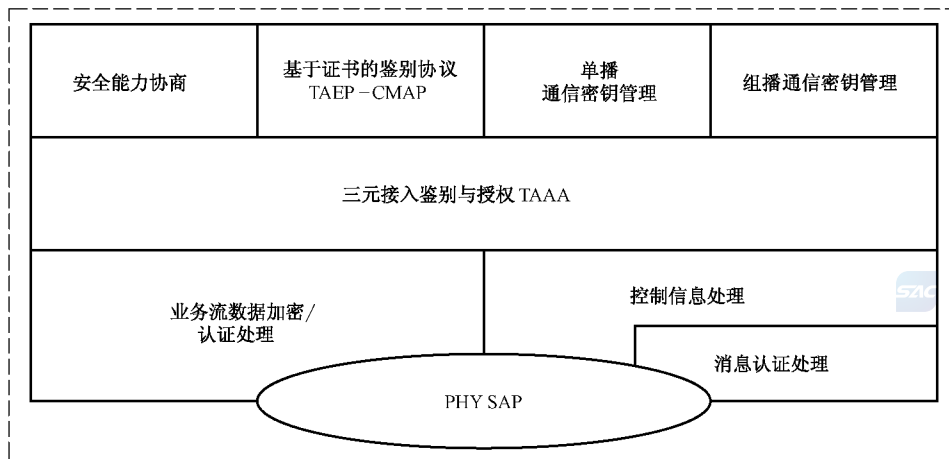


图 C.1 安全子层

C.2 三元接入鉴别与授权 TAAA

C.2.1 概述

SS 利用 TAAA 机制从 BS 获得授权和通信密钥材料、维持周期性的重授权和密钥更新。TAAA 机制使用 X.509 数字证书(IETF RFC3280)、公钥算法(签名算法推荐 ECC)以及对称加密算法(推荐 SMS4)来实现 BS 和 SS 之间的安全通信。

TAAA 机制使用了客户端/服务器模型,在这个模型中,SS 作为 TAAA 机制的“客户端”,向 BS 请求密钥材料,而 BS 作为 TAAA 机制“服务器”,对这些请求进行响应,确保了每个 SS 只收到向他授权的密钥材料。TAAA 机制使用授权密钥管理过程的完成。

TAAA 机制用公钥加密机制在 BS 和 SS 之间建立共享的授权密钥(AK)。这个共享的授权密钥确保后续 TEK 的安全交换。这种密钥分发的两层机制使得 TEK 的更新不再出现公钥操作的计算代价。

BS 在授权密钥管理过程中对 SS 进行鉴权。每个 SS 都有一个 X.509 数字证书。当 SS 向 BS 请求 AK 时,SS 将它的数字证书发给 BS。BS 验证数字证书,然后利用验证后的公钥对 AK 加密后,将其发送给 SS。

TAAA 机制包括安全能力协商、基于证书的鉴别协议 TAEP-CMAP(TAEP Certificate-based

WMAN Authentication Protocol)、单播通信密钥管理、组播通信密钥管理等四个过程。其中安全能力协商过程中的分组在传输时封装在终端能力报告请求和终端能力报告响应消息中,其他三个过程包括 TAEP-CMAP、单播通信密钥管理、组播通信密钥管理的所有分组在传输时封装在管理消息中。

C.2.2 安全能力协商过程

图 C.2 安全能力协商过程完成城域网系统的确认和安全参数的协商。涉及消息流程。

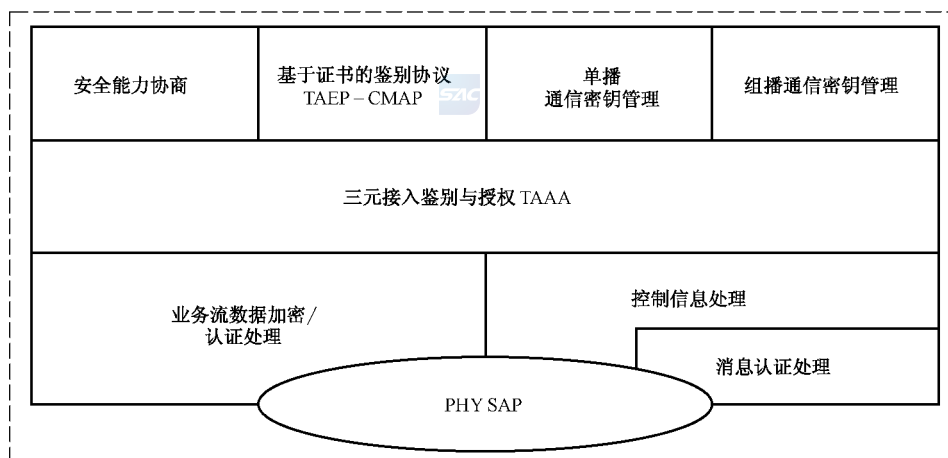


图 C.2 安全能力协商过程

其中,安全能力协商请求分组数据字段格式见表 C.1。

表 C.1 安全能力协商请求分组数据字段格式

FLAG	PFLAG	Negotiation-Parameters
------	-------	------------------------

其中:

- FLAG 字段长度为 1 个八位位组,高 4 位用来表示本安全方案或其他安全方案;
- PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别为协商请求分组,值为 0x11;
- Negotiation-Parameters 字段使用 TLV 属性表示,表示 SS 的协商的安全参数。

属性字段采用表 C.2TLV 格式定义。

表 C.2 扩展属性

类型	长度	数值
----	----	----

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:

- 签名属性;
- 证书验证结果;
- 身份;
- 密钥材料;
- 算法列表;
- SA 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。数值字段表示属性的内容。

在 SS 的能力协商请求消息(SBC-REQ)中嵌入上述安全协商请求分组,BS 在接收到 SS 的能力协商请求消息后,提取安全能力协商请求分组,通过 FLAG 和 PFLAG 字段判断 SS 发送的消息是否为本方案的协商请求分组后,进行如下处理:

- a) 读取 Negotiation-Parameters 字段的值,根据本地策略构造 Neg-Para-Confirm 字段,发送协商响应分组;
- b) 串联 Negotiation-Parameters 字段和 Neg-Para-Confirm 字段的值,保存用于确认协商的分组。其中,协商响应分组数据字段格式见表 C.3。

表 C.3 安全能力协商响应分组数据字段格式

FLAG	PFLAG	Neg-Para-Confirm
------	-------	------------------

其中:

- FLAG 字段长度为 1 个八位位组,高 4 位表示安全方法类别;
- PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别,值为 0x12;
- Neg-Para-Confirm 字段使用 TLV 属性表示,表示 BS 根据本地策略决定的安全参数。

在 BS 的能力协商响应消息(SBC-RSP)中嵌入上述安全协商响应分组,SS 在接收到 BS 的能力协商响应消息后,从中提取安全能力分组字段,从 FLAG 和 PFLAG 字段判断接收的消息是否为本方案的协商分组响应后,进行以下处理:

串联发送的 Negotiation-Parameters 和接收的 Neg-Para-Confirm 字段的值,保存用于确认协商的安全参数。

C.2.3 基于证书的鉴别协议(TAEP-CMAP)

C.2.3.1 TAEP-CMAP 概述

基于证书的鉴别协议(TAEP-CMAP)完成 SS 和 BS 之间的双向身份鉴别或者 BS 对 SS 的单向身份鉴别,身份鉴别成功后,在 SS 和 BS 之间协商授权密钥 AK,同时,BS 为 SS 授权一系列 SA。TAEP-CMAP 协议使用 TAEP 数据封装格式进行封装,完成消息在 SS、BS 和 ASU 之间交互,其过程见图 C.3 所示。

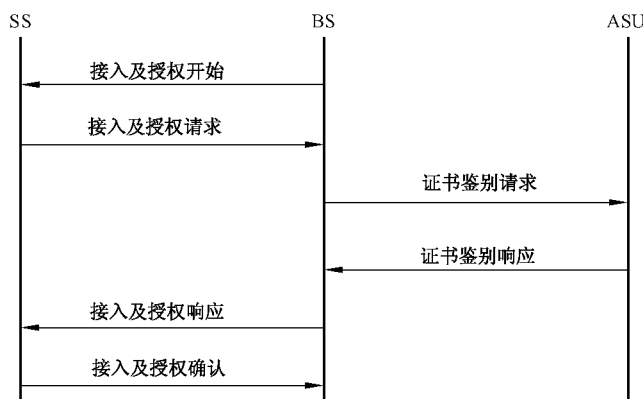


图 C.3 基于证书的鉴别协议(TAEP-CMAP)过程

C.2.3.2 接入及授权开始分组

接入及授权开始分组封装在 TAEP-Request-TAAA 的 Date 字段中,数据字段为空,指示 SS 发送

接入及授权请求。

C.2.3.3 接入及授权请求分组

接入及授权请求分组封装在 TAEP-Response-TAAA 的 Date 字段中,数据字段的格式见表 C.4。

表 C.4 接入及授权请求分组数据字段格式

标识 FLAG	PFLAG	SS 身份 IDSS	SS 第一证书 PrimaryCertSS	SS 第二证书 SecondyCertSS	算法列表 AlgsList	授权密钥 标识符 AKID	SS 挑战 NSS	消息校验码 MIC 或 SS 签名 SigSS
------------	-------	---------------	--------------------------	--------------------------	------------------	---------------------	--------------	----------------------------

其中:

- 标识 FLAG 字段长度为 1 个八位位组,高 4 位标识鉴别方法类别,而低 4 位中的位 0、1、2 有意义。当 SS 初次物理关联至 BS 时,位 0 (AK 更新标识)的值为 0;当进行 AK 更新时,位 0(AK 更新标识)的值为 1。如果 SS 和 BS 进行单证书模式的鉴别过程,位 1(证书模式标识)的值为 0;如果进行双证书模式的鉴别过程,位 1 的值为 1。如果 SS 和 BS 进行双向鉴别过程,位 2 (鉴别类型标识)的值为 0;如果仅进行 BS 对 SS 的单向鉴别过程,位 2(鉴别类型标识)的值为 1。
- PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别,值为 0x21。
- SS 身份字段,其格式定义为 TLV 属性,即类型-长度-值格式。
- SS 第一证书字段,证书格式定义为 X.509 v3。该字段为可选项,当且仅当标识字段 Flag 位 0 的值置为“0”时使用,而置为“1”时省略。也就是说,当且仅当 SS 初次物理关联至 BS 时需要该字段,而当进行 AK 更新时,该字段可以略去。
- SS 第二证书字段,证书格式定义为 X.509 v3。该字段为可选项,当且仅当标识字段 Flag 位 0 的值置为“0”时、并且标识字段 Flag 位 1 的值置为“1”时使用,也就是说,当且仅当 SS 初次物理关联至 BS 并采用双证书的鉴别模式时需要该字段,而当进行 AK 更新时,该字段可以略去。
- 算法列表字段标识 SS 支持的所有分组算法,其格式定义为 TLV 属性。该字段为可选项,当且仅当标识字段 Flag 的位 0 值置为“0”时使用,而置为“1”时省略。也就是说,当且仅当 SS 初次物理关联至 BS 时需要该字段,而当进行 AK 更新时,该字段可以略去。
- 授权密钥标识符字段长度为 1 个八位位组,其中位 0 有意义,标识当前协商或更新的密钥 AK,其值在“0”和“1”之间进行翻转。
- SS 挑战字段长度为 32 个八位位组,可以由伪随机数生成算法产生。
- 消息完整性校验码或 SS 签名字段。消息校验码字段长度为 32 个八位位组,当且仅当标识字段 Flag 位 0 的值置为“1”时使用。也就是说,当且仅当进行 AK 更新时需要该字段,该字段值使用旧 AK 导出的授权完整性密钥计算。而当 SS 初次物理关联至 BS 并使用双证书模式时,也就是说,标识字段 Flag 位 0 的值置为“0”且位 1 值的置为“1”时该字段为 SS 签名,此时使用 SS 第一证书私钥计算签名。

当 SS 物理关联或重新关联至 BS,SS 和 BS 选择采用单向或双向身份鉴别方法,或 SS 的本策略要求进行授权密钥 AK 更新过程时,SS 向 BS 发送接入及授权请求分组。

BS 接收到由 SS 发送的接入及授权请求分组后,进行如下处理:

- a) BS 检查接入鉴别请求分组中标识 FLAG 字段的位 0(AK 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) BS 利用旧的 AK 导出的授权完整性密钥本地计算 MIC 并与接收到的 MIC 进行比较，如果它们的值相等，构造接入及授权响应分组发送给 SS；若不相等，则丢弃该接入及授权请求分组；
- c) 若采用单证书模式，构造证书鉴别请求分组发送给 ASU；若采用双证书模式，则验证 SS 签名，若不正确，则丢弃该接入授权请求分组，否则构造证书鉴别请求分组发送给 ASU。

C.2.3.4 证书鉴别请求分组

由 BS 发往 ASU，证书鉴别请求分组封装在 TAEP-Request-TAAA 的 Date 域中，数据字段的格式见表 C.5。

表 C.5 证书鉴别请求分组数据字段格式

标识 FLAG	PFLAG	SS 第一证书 PrimaryCertSS	SS 第二证书 SecndyCertSS	BS 证书 CertBS	AK 密钥材 料密文 AKM	SS 挑战 NSS	BS 挑战 NBS	BS 签名 SigBS
------------	-------	--------------------------	-------------------------	-----------------	-------------------	--------------	--------------	----------------

其中：

- 标识 FLAG 字段长度为 1 个八位位组，高 4 位标识方法类别，而低 4 位中的位 0、1、2 有意义。当 SS 关联或重新关联至 BS 时进行证书鉴别过程，位 0 (AK 更新标识)的值为 0；当进行 AK 更新时，位 0(AK 更新标识)的值为 1。如果 SS 和 BS 进行单证书模式的鉴别过程，位 1(证书模式标识)的值为 0；如果进行双证书模式的鉴别过程，位 1 的值为 1。如果 SS 和 BS 进行双向鉴别过程，位 2(鉴别类型标识)的值为 0；如果仅进行 BS 对 SS 的单向鉴别过程，位 2(鉴别类型标识)的值为 1。该字段的取值应与接入及授权请求中的标识 FLAG 字段取值相同。
- PFLAG 字段长度为 1 个八位位组，表示该消息所属的协议类别，值为 0x22。
- SS 第一证书字段，证书格式定义为 X.509 v3。
- SS 第二证书字段，证书格式定义为 X.509 v3。该字段为可选项，当且仅当标识字段 Flag 位 1 的值置为“1”时使用，也就是说，当且仅当采用双证书的鉴别模式时需要该字段，采用单证书模式时该字段可以略去。
- BS 证书字段，证书格式定义为 X.509 v3。该字段为可选项，当且仅当标识字段 Flag 位 2 的值置为“0”时使用，而置为“1”时省略。也就是说，当且仅当 SS 与 BS 进行双向身份鉴别时需要该字段，而仅进行 BS 对 SS 的单向鉴别过程时不需要该字段。
- AK 密钥材料密文字段，即由 SS 公钥加密密钥材料，其格式定义为 TLV 属性。该字段为可选项，当且仅当标识字段 Flag 位 2 值的置为“1”时使用，而置为“0”时省略。也就是说，当且仅当只进行 BS 对 SS 的单向身份鉴别过程时需要该字段，而 SS 与 BS 进行双向身份鉴别时不需要该字段。
- SS 挑战字段长度为 32 个八位位组，该字段的取值应与接入及授权请求中的 SS 挑战字段取值相同。
- BS 挑战字段长度为 32 个八位位组，由 BS 通过伪随机数生成算法产生。
- BS 签名字段采用签名 TLV 属性表示，它是对本分组中除本字段之外所有数据字段的签名。该字段为可选项，当且仅当标识字段 Flag 位 2 的值置为“0”时使用，而置为“1”时省略。也就是说，当且仅当 SS 与 BS 进行双向身份鉴别时需要该字段，而仅进行 BS 对 SS 的单向身份鉴别过程时不需要该字段。

BS 在收到 SS 的接入及授权请求分组,并且该分组中的标识 FLAG 字段的位 0 值为“0”时,BS 发送证书鉴别请求分组给 ASU。

ASU 收到 BS 发来的证书鉴别请求分组后,进行如下处理:

- a) ASU 验证 SS 第一证书、SS 第二证书和 BS 证书(SS 第二证书仅在双证书的鉴别模式时存在,BS 证书仅在双向身份鉴别过程中存在),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 SS 第一证书、SS 第二证书和 BS 证书的状态,并执行 b)操作。
- b) 根据 SS 第一证书、SS 第二证书和 BS 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 BS。

C.2.3.5 证书鉴别响应分组

由 ASU 发往 BS,证书鉴别响应分组封装在 TAEP-Response-TAAA 的 Date 字段中,数据字段的格式见表 C.6。

表 C.6 证书鉴别响应分组数据字段格式

标识 FLAG	PFLAG	SS 第一证书 PrimaryCertSS	SS 第一证书验证结果 ResultPrimarySS	SS 第二证书 SecondyCertSS	SS 第二证书验证结果 ResultSecondySS	BS 证书 CertBS	BS 证书验证结果 ResultSS
AK 密钥材料密文 AKM		SS 挑战 NSS	BS 挑战 NBS	ASU ₁ 签名 SigASU ₁		ASU ₂ 签名 SigASU ₂	

其中:

- 标识 FLAG 字段长度为 1 个八位位组,高 4 位标识方法类别,而低 4 位中的位 0、1、2 有意义。当 SS 关联或重新关联至 BS 时进行证书鉴别过程,位 0(AK 更新标识)的值为 0;当证书鉴别过程进行 AK 更新时,位 0(AK 更新标识)的值为 1。如果 SS 和 BS 进行单证书模式的鉴别过程,位 1(证书模式标识)的值为 0;如果进行双证书模式的鉴别过程,位 1 的值为 1。如果 SS 和 BS 进行双向鉴别过程,位 2(鉴别类型标识)的值为 0;如果仅进行 BS 对 SS 的单向鉴别过程,位 2(鉴别类型标识)的值为 1。该字段的取值应与证书鉴别请求中的标识 FLAG 字段取值相同。
- PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别,值为 0x23。
- SS 第一证书字段,证书格式定义为 X.509 v3。
- SS 第一证书验证结果字段采用 TLV 属性表示。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。

- SS 第二证书字段,证书格式定义为 X.509 v3。该字段为可选项,当且仅当标识字段 Flag 位 1 的值置为“1”时使用,而置为“0”时省略。也就是说,当且仅当采用双证书鉴别模式时需要该字段,而采用单证书鉴别模式时不需要该字段。
 - SS 第二证书验证结果字段采用 TLV 属性表示。该字段为可选项,当且仅当标识字段 Flag 位 1 的值置为“1”时使用,而置为“0”时省略。也就是说,当且仅当采用双证书鉴别模式时需要该字段,而采用单证书鉴别模式时不需要该字段。证书结果如上定义。
 - BS 证书字段,证书格式定义为 X.509 v3。该字段为可选项,当且仅当标识字段 Flag 位 2 的值置为“0”时使用,而置为“1”时省略。也就是说,当且仅当 SS 与 BS 进行双向身份鉴别时需要该字段,而仅进行 BS 对 SS 的单向鉴别过程时不需要该字段。
 - BS 证书验证结果字段采用 TLV 属性表示。该字段为可选项,当且仅当标识字段 Flag 位 2 的值置为“0”时使用,而置为“1”时省略。也就是说,当且仅当 SS 与 BS 进行双向身份鉴别时需要该字段,而仅进行 BS 对 SS 的单向鉴别过程时不需要该字段。证书结果如上定义。
 - AK 密钥材料密文字段,其格式定义为 TLV 属性。该字段为可选项,当且仅当标识字段 Flag 位 1 值的置为“1”时使用,而置为“0”时省略。也就是说,当且仅当只进行 BS 对 SS 的单向鉴别过程时需要该字段,而 SS 与 BS 进行双向身份鉴别时不需要该字段。该字段取值应与证书鉴别请求中的 AK 密钥材料密文字段值相等。
 - SS 挑战字段长度为 32 个八位位组,该字段的取值应与证书鉴别请求中的 SS 挑战字段取值相同。
 - BS 挑战字段长度为 32 个八位位组,该字段的取值应与证书鉴别请求中的 BS 挑战字段取值相同。
 - ASU1 签名字段采用签名 TLV 属性表示,它是对本分组中本字段之前所有数据字段的签名。ASU1 表示 BS 所信任的 ASU。
 - ASU2 签名字段采用签名 TLV 属性表示,它是对本分组中 ASU1 签名字段之前所有数据字段的签名。ASU2 表示 SS 所信任的 ASU。该字段为可选项,当且仅当 SS 信任的 ASU 和 BS 信任的 ASU 不同时才会用到该字段,如果他们信任的 ASU 相同,那么该字段将被略去。
- ASU 收到证书鉴别请求分组后,向 BS 发送证书鉴别响应分组。

BS 收到证书鉴别响应分组后,进行如下处理:

- a) 检查证书鉴别响应分组中的 BS 挑战字段的值是否与自己在证书鉴别请求分组中的 BS 挑战字段值相同,若相同,则执行 b) 操作;否则,丢弃该证书鉴别响应分组;
- b) BS 验证 ASU1 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 SS 证书鉴别结果成功,构造接入及授权响应分组发送给 SS。若 SS 证书鉴别结果不成功,BS 设定接入结果为不成功,并构造接入及授权响应分组发送给 SS,然后解除与 SS 的链路验证。

C.2.3.6 接入及授权响应分组

由 BS 发往 SS,接入及授权响应分组封装在 TAEP-Request-TAAA 的 Date 字段中,数据字段的格式如表 C.7。

表 C.7 接入及授权响应分组数据字段格式

标识 FLAG	PFLAG	SS 身份 IDSS	BS 身份 IDBS	SS 第一证书 PrimaryCertSS	SS 第一证书验证结果 ResultPrimarySS
SS 第二证书 SecondyCertSS		SS 第二证书验证结果 ResultSecondySS	BS 证书 CertBS	BS 证书验证结果 ResultBS	ASU1 签名 SigASU ₁
ASU2 签名 SigASU ₂		算法列表 AlgsList	算法标识 Alg	SA 列表 SasList	授权密钥标识符 AKID
AK 密钥材料密文 AKM		AK 有效期 LifeAK	SS 挑战 NSS	BS 挑战 NBS	消息校验码 MIC 或 BS 签名 SigBS

其中：

- 标识 FLAG 字段长度为 1 个八位位组，高 4 位标识方法类别，而低 4 位中的位 0、1、2 有意义。当 SS 物理关联或重新关联至 BS 时，位 0 (AK 更新标识) 的值为 0；当进行 AK 更新时，位 0 (AK 更新标识) 的值为 1。如果 SS 和 BS 进行单证书模式的鉴别过程，位 1 (证书模式标识) 的值为 0；如果进行双证书模式的鉴别过程，位 1 的值为 1。如果 SS 和 BS 进行双向鉴别过程，位 2 (鉴别类型标识) 的值为 0；如果仅进行 BS 对 SS 的单向鉴别过程，位 2 (鉴别类型标识) 的值为 1。该字段的取值应与接入及授权请求中的标识 FLAG 字段取值相同。
- PFLAG 字段长度为 1 个八位位组，表示该消息所属的协议类别，值为 0x24。
- SS 身份字段，其格式定义为 TLV 属性。
- BS 身份字段，其格式定义为 TLV 属性。
- SS 第一证书字段，证书格式定义为 X.509 v3。该字段为可选项，当且仅当标识字段 Flag 位 0 的值置为“0”时使用，而置为“1”时省略。也就是说，当且仅当 SS 物理关联或重新关联至 BS 时需要该字段，而当进行 AK 更新时不需要该字段。
- SS 第一证书验证结果字段采用 TLV 属性表示。该字段为可选项，当且仅当标识字段 Flag 位 0 的值置为“0”时使用，而置为“1”时省略。也就是说，当且仅当 SS 物理关联或重新关联至 BS 时需要该字段，而当进行 AK 更新时不需要该字段。证书结果如上定义。
- SS 第二证书字段，证书格式定义为 X.509 v3。该字段为可选项，当且仅当标识字段 Flag 位 1 的值置为“1”时使用，而置为“0”时省略。也就是说，当且仅当采用双证书鉴别模式时需要该字段，而采用单证书鉴别模式时不需要该字段。
- SS 第二证书验证结果字段采用 TLV 属性表示。该字段为可选项，当且仅当标识字段 Flag 位 1 的值置为“1”时使用，而置为“0”时省略。也就是说，当且仅当采用双证书鉴别模式时需要该字段，而采用单证书鉴别模式时不需要该字段。证书结果如上定义。
- BS 证书字段，证书格式定义为 X.509 v3。该字段为可选项，当标识字段 Flag 位 2 的值置为“0”且位 1 为“0”时使用，其余情况不需要该字段。也就是说，仅当 SS 物理关联或重新关联至 BS，且他们之间需要进行双向身份鉴别时需要该字段，除此之外都不需要该字段。
- BS 证书验证结果字段采用 TLV 属性表示。该字段为可选项，当标识字段 Flag 位 2 的值置为“0”且位 1 为“0”时使用，其余情况不需要该字段。也就是说，仅当 SS 物理关联或重新关联至 BS，且他们之间需要进行双向身份鉴别时需要该字段，除此之外都不需要该字段。证书结果如上定义。
- ASU1 签名字段采用签名 TLV 属性表示，该字段应与证书鉴别响应分组中的 ASU1 签名字段值相同。ASU1 表示 BS 所信任的 ASU。该字段为可选项，当标识字段 Flag 位 0 的值置为

“0”时使用,其余情况不需要该字段。也就是说,仅当 SS 物理关联或重新关联至 BS 时需要该字段,除此之外都不需要该字段。

- ASU2 签名字段采用签名 TLV 属性表示,该字段应与证书鉴别响应分组中的 ASU2 签名字段值相同。ASU2 表示 SS 所信任的 ASU。该字段为可选项,当标识字段 Flag 位 0 的值为“0”且 SS 信任的 ASU 和 BS 信任的 ASU 不同时使用,其余情况不需要该字段。也就是说,仅当 SS 物理关联或重新关联至 BS,且他们信任的 ASU 不同时需要该字段,除此之外都不需要该字段。
- 算法列表字段标识 SS 支持的所有分组算法,其格式定义为 TLV 属性,该字段应与接入及授权请求分组中的算法列表字段值相同。该字段为可选项,当且仅当标识字段 Flag 的位 0 位置为“0”时使用,而置为“1”时省略。也就是说,当且仅当 SS 物理关联或重新关联至 BS 时需要该字段,而当进行 AK 更新时不需要该字段。
- 算法标识字段标识 BS 从 SS 支持的算法列表中确定选取的算法,其格式定义为 TLV 属性。该字段为可选项,当且仅当标识字段 Flag 的位 0 位置为“0”时使用,而置为“1”时省略。也就是说,当且仅当 SS 物理关联或重新关联至 BS 时需要该字段,而当进行 AK 更新时不需要该字段。
- SA 列表字段标识 BS 对 SS 授权的 SA,其格式定义为 TLV 属性。该字段为可选项,当且仅当标识字段 Flag 的位 0 位置为“0”时使用,而置为“1”时省略。也就是说,当且仅当 SS 物理关联或重新关联至 BS 时需要该字段,而当进行 AK 更新时不需要该字段。
- 授权密钥标识符字段长度为 1 个八位位组,其中位 0 有意义,标识协商或更新的当前密钥 AK,其值在“0”和“1”之间进行翻转。该字段应与接入及授权请求分组中的授权密钥标识符字段值相同。
- AK 密钥材料密文字段,其格式定义为 TLV 属性。当采用单证书模式时,该字段标识 BS 选取的 AK 密钥材料使用 SS 第一证书公钥加密后的密文,当采用双证书模式时,该字段标识 BS 选取的 AK 密钥材料使用 SS 第二证书公钥加密后的密文。
- AK 有效期字段标识由 AK 密钥材料导出的密钥 AK 的使用有效期。
- SS 挑战字段长度为 32 个八位位组,该字段的取值应与接入及授权请求中的 SS 挑战字段取值相同。
- BS 挑战字段长度为 32 个八位位组,该字段的取值应与证书鉴别请求中的 BS 挑战字段取值相同。
- 消息完整性校验码或 BS 签名字段。如果标识字段 Flag 位 0 值的置为“0”且位 2 值的置为“0”时使用 BS 签名,其余情况使用消息完整性校验码。也就是说,当且仅当 SS 物理关联或重新关联至 BS,且 SS 与 BS 进行双向身份鉴别时使用 BS 签名,除此之外,使用消息完整性校验码。如果使用消息完整性校验码,其值通过授权密钥标识符字段对应的授权密钥导出的授权完整性密钥计算。

BS 收到证书鉴别响应分组,或收到接入及授权请求分组后,发送接入及授权响应分组。

SS 收到接入及授权响应分组后,进行如下处理:

- a) 根据 BS 身份和 SS 身份判断是否为对应当前接入及授权请求分组的接入及授权响应分组,若不是,则丢弃该接入及授权响应分组;否则,执行 b) 操作。
- b) 检查标识 FLAG 字段的位 0、位 1 与自己发送的接入及授权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 SS 挑战与自己在接入及授权请求分组中发送的 SS 挑战是否相同,若不同,则丢弃该接入及授权响应分组;否则,执行 d) 操作。
- d) SS 在证书鉴别结果中查找自身所信任的鉴别服务单元的签名,验证 ASU2 签名,若不正确,则丢弃该接入及授权响应分组。如果 SS 和 BS 信任的鉴别服务单元相同,则只需验证 ASU1 签名。
- e) 如果 Flag 字段标识需要进行 BS 与 SS 的双向鉴别时,则检查 BS 的证书的鉴别结果是否为有

效,若无效,解除与该 BS 的链路验证;若有效,则执行 f)操作。如果 Flag 字段标识只需进行 BS 对 SS 的单向鉴别时,则直接执行 f)操作。

- f) 如果 Flag 字段标识 AK 更新过程或只需进行 BS 对 SS 的单向鉴别时,验证消息完整性校验码 MIC 的正确性,其值通过授权密钥标识符字段对应的授权密钥导出的授权完整性密钥计算;否则验证 BS 的签名有效性。如果验证不通过,则解除与该 BS 的链路验证;否则执行 g)操作。
- g) SS 使用自己的私钥解密 AK 密钥材料密文 AKM 获得 AK,再利用 SS 挑战和 BS 挑战导出密钥加密密钥和授权完整性密钥。

C.2.3.7 接入及授权确认分组

由 SS 发往 BS,接入及授权确认分组封装在 TAEP-Response-TAAA 的 Date 字段中,数据字段的格式见表 C.8。

表 C.8 接入及授权确认分组数据字段格式

FLAG	PFLAG	SS 身份 IDSS	BS 身份 IDBS	授权密钥标识符 AKID	BS 挑战 NBS	算法标识 Alg	消息校验码 MIC2
------	-------	------------	------------	--------------	-----------	----------	------------

其中:

- 标识 FLAG 字段长度为 1 个八位位组,高 4 位标识方法类别,而低 4 位中的位 0、1、2 有意义。当 SS 物理关联或重新关联至 BS 时,位 0(AK 更新标识)的值为 0;当进行 AK 更新时,位 0(AK 更新标识)的值为 1。如果 SS 和 BS 进行单证书模式的鉴别过程,位 1(证书模式标识)的值为 0;如果进行双证书模式的鉴别过程,位 1 的值为 1。如果 SS 和 BS 进行双向鉴别过程,位 2(鉴别类型标识)的值为 0;如果仅进行 BS 对 SS 的单向鉴别过程,位 2(鉴别类型标识)的值为 1。该字段的取值应与接入及授权请求中的标识 FLAG 字段取值相同。
 - PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别,值为 0x25。
 - SS 身份字段,其格式定义为 TLV 属性。
 - BS 身份字段,其格式定义为 TLV 属性。
 - 授权密钥标识符字段长度为 1 个八位位组,其中位 0 有意义,标识协商或更新的当前密钥 AK,其值在“0”和“1”之间进行翻转。该字段应与接入及授权请求分组中的授权密钥标识符字段值相同。
 - BS 挑战字段长度为 32 个八位位组,该字段应与接入及授权响应分组中的 BS 挑战字段值相同。
 - 算法标识字段标识 BS 从 SS 支持的算法列表中确定选取的算法,其格式定义为 TLV 属性。该字段应与接入及授权响应分组中的算法标识字段值相同。
 - 消息校验码,其值通过授权密钥标识符字段对应的授权密钥导出的消息授权完整性密钥计算。SS 收到接入及授权响应分组后,向 BS 发送接入及授权确认分组。
- BS 收到接入及授权确认分组后,进行如下处理:
- a) 根据 BS 身份和 SS 身份判断是否为对应当前接入及授权请求分组的接入及授权响应分组,若不是,则丢弃该接入及授权确认分组;否则,执行 b)操作;
 - b) 检查标识 FLAG 字段的位 0、位 1 与自己发送的接入及授权响应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 BS 挑战与自己在接入及授权响应分组中发送的 BS 挑战是否相同,若不同,则丢弃该接入及授权确认分组;否则,执行 d)操作;
 - d) 比较算法标识字段与自己在接入及授权响应分组中发送的算法标识是否相同,若不同,则丢弃该接入及授权确认分组;否则,执行 e)操作;
 - e) 验证完整性校验码 MIC 的正确性,其值通过授权密钥标识符字段对应的授权密钥导出的授权完整性密钥计算。如果验证不通过,则丢弃该接入及授权确认分组;否则,接入鉴别成功,启用新的授权密钥 AK。

C.2.4 单播通信密钥管理

C.2.4.1 概述

单播通信密钥管理完成单播通信密钥的生成、分配、和更新,涉及图 C.4 消息流程。

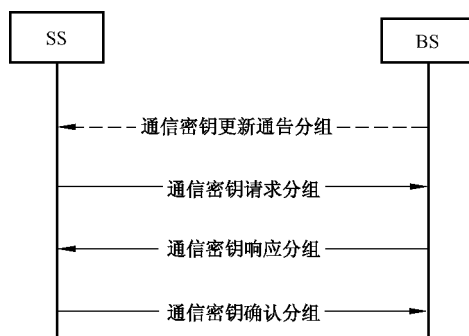


图 C.4 单播通信密钥管理协议

C.2.4.2 通信密钥更新通告分组

通信密钥更新通告分组数据字段格式见表 C.9。

表 C.9 通信密钥更新通告分组数据字段格式

FLAG	PFLAG	SAID	BSID	AKID	TEKIDold	MIC
------	-------	------	------	------	----------	-----

其中:

- FLAG 字段长度为 1 个八位位组,高 4 位表示安全方法类别;
- PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别,值为 0x31;
- SAID 字段长度为 2 个八位位组,表示需要更新的通信密钥所属的安全关联;
- BSID 字段长度为 6 个八位位组,表示 BS 的标志;
- AKID 字段长度为 1 个八位位组,表示所使用的授权密钥的索引值;
- TEKIDold 字段长度为 1 个八位位组,表示需要更新的通信密钥的索引值;
- MIC 字段长度为 32 个八位位组,表示该消息数据的完整性校验值,使用 AKID 对应的授权密钥 AK 导出的授权完整性密钥计算。

在 BS 的状态机指示当前单播通信密钥需要更新时,BS 发送该分组通告 SS 开始更新单播通信密钥。

SS 通过 FLAG 字段和 PFLAG 字段确认接收到 BS 发送的消息为本方案的通信密钥更新通告分组后,进行如下处理:

- a) 用户设备使用 AKID 指向的完整性密钥使用本地的 BSID 重新计算 MIC,如果接收的 MIC 与本地计算的 MIC 不同,则忽略该分组,不进行任何处理;否则执行步骤 b);
- b) 检查 SAID 是否是当前激活的安全连接,如果不是当前激活的安全连接,则忽略通信密钥更新通告分组,不进行任何处理,否则执行 c);
- c) SS 构造通信密钥请求分组,通信密钥请求分组的 SAID 字段拷贝自通信密钥更新通告消息,通信密钥请求分组的 TEKID 字段拷贝自通信密钥更新通告消息。

C.2.4.3 通信密钥请求分组

通信密钥请求分组数据字段格式见表 C.10。

表 C.10 通信密钥请求分组数据字段格式

FLAG	PFLAG	SSID	SAID	AKID	TEKIDold	NSS	MIC
------	-------	------	------	------	----------	-----	-----

其中：

- FLAG 字段长度为 1 个八位位组，高 4 位标识安全方法类别；
- PFLAG 字段长度为 1 个八位位组，表示该消息所属的协议类别，值为 0x32；
- SSID 字段长度为 6 个八位位组，表示用户设备的唯一标志符；
- SAID 字段长度为 2 个八位位组，表示需要更新的通信密钥所属的安全关联；
- AKID 字段长度为 1 个八位位组，表示客户端当前使用的授权密钥索引值；
- TEKIDold 字段长度为 1 个八位位组，表示需要更新的通信密钥的索引值；
- NSS 字段长度为 16 个八位位组，表示用户设备生成的 128 比特随机数；
- MIC 字段长度为 16 个八位位组，使用 AKID 对应的授权密钥 AK 导出的授权完整性密钥计算。

在 SS 的状态机指示某个 SAID 的单播通信密钥需要更新时，SS 构造并发送该分组请求 BS 传输新的单播通信密钥。或者在 SS 接收到 BS 的单播通信密钥更新通告时，SS 构造上述分组。或者在 SS 状态机指示需要为某个 SAID 申请单播通信密钥时，SS 构造上述分组。

BS 通过 FLAG 字段和 PFLAG 字段确认接收到 SS 发送的消息为本方案的通信密钥请求分组后，进行如下处理：

- a) 使用 AKID 指向的完整性密钥重新计算 MIC，如果接收的 MIC 与本地计算的 MIC 不同，则忽略该分组，不进行任何处理；否则执行步骤 b)。
- b) 检查 SSID 是否是 BS 的当前连接用户，如果不是 BS 的当前连接用户，则忽略该分组，不进行任何处理；否则执行步骤 c)。
- c) 检查 SAID 是否是 BS 与 SSID 激活的安全连接，如果是当前激活的安全连接则执行步骤 d)。否则检查该 SAID 是否属于该 SS 的授权安全连接，如果属于该 SSID 的授权安全连接，则执行步骤 d)。否则忽略通信密钥更新通告分组，不进行任何处理。
- d) 检查 AKID 是否指向 BS 与该 SSID 建立的当前授权密钥索引，如果指向有效的授权密钥，则执行步骤 e)，否则忽略该分组，不进行任何处理。
- e) 如果 BS 与 SSID 在 SAID 确定的连接中存在 TEKIDold，则生成新的 TEK 的 Key，并使用 TEKIDold 的翻转值 TEKIDNew 指向。之后执行步骤 f)。
- f) BS 构造通信密钥响应分组，其中的 SSID 字段、SAID 字段、NSS 字段拷贝自通信密钥请求分组，N1BS 字段存放 BS 为该 SSID 的该 SAID 更新 TEK 的会话期间生成的新的 N1BS 随机数，Key 字段使用协商的通信密钥加密算法和通信密钥响应分组中 AKID 字段指向的数据加密密钥加密，加密的内容为 BS 中该 SSID 的 SAID 确定的连接中新生成的单播通信密钥材料 TEKM，TEKM 可以通过随机数产生函数生成，LifeTEKOld 字段填充请求更新的 TEK 的生命周期，LifeTEKNew 字段填充被新生成的 TEK 的生命周期，MIC 字段使用被加密的 TEKM 推导的单播通信完整性密钥计算。

C.2.4.4 通信密钥响应分组



通信密钥响应分组数据字段格式见表 C.11。

表 C.11 通信密钥响应分组数据字段格式

FLAG		PFLAG	SSID	BSID	SAID	AKID	TEKIDold	TEKIDNew
N1SS	N1BS	TEKMC	LifeTEKOld	LifeTEKNew			MIC	

其中：

- FLAG 字段长度为 1 个八位位组，高 4 位标识安全方法类别；
- PFLAG 字段长度为 1 个八位位组，表示该消息所属的协议类别，值为 0x33；
- SSID 字段长度为 6 个八位位组，表示用户设备的唯一标志符；
- BSID 字段长度为 6 个八位位组，表示 BS 的唯一标志符；
- SAID 字段长度为 2 个八位位组，表示需要更新的通信密钥所属的安全关联；
- AKID 字段长度为 1 个八位位组，表示 BS 当前使用的授权密钥索引值；
- TEKIDOld 字段长度为 1 个八位位组，表示需要更新的通信密钥的索引值；
- TEKIDNew 字段长度为 1 个八位位组，表示更新后的通信密钥的索引值；
- N1SS 字段长度为 16 个八位位组，表示用户设备生成的 128 比特随机数；
- N1BS 字段长度为 16 个八位位组，表示 BS 生成的 128 比特随机数；
- TEKMC 字段长度为 16 个八位位组，表示加密新的通信密钥材料后生成的密文分组；
- LifeTEKOld 字段长度为 4 个八位位组，表示与需要更新的通信密钥对应的生命周期；
- LifeTEKNew 字段长度为 4 个八位位组，表示与更新的通信密钥对应的生命周期；
- MIC 字段长度为 32 个八位位组，表示该消息数据的完整性校验值。该字段使用 TEKIDNew 所指向的 TEK 推导的完整性密钥计算。

BS 接收到通信密钥请求分组后，构造上述通信密钥响应分组。

SS 通过 FLAG 字段和 PFLAG 字段确认接收到 BS 发送的消息为本方案的通信密钥响应分组后，进行如下处理：

- a) 使用 AKID 指向的密钥加密密钥解密 TEKMC 字段的密文，获得新的 TEK。执行步骤 b)。
- b) 使用新的 TEKIDNew 指向的 TEK 推导完整性密钥，使用本地的 SSID，本地当前连接的 BSID，通信密钥请求分组中的 SAID，通信密钥响应分组中的 AKID 和 TEKIDOld，TEKIDNew，本地的 N1SS，通信密钥响应分组中的 N1BS、Key 字段和 LifeTEKOld、LifeTEKNew 字段来重新计算 MIC 值，比较本地计算的 MIC 值与接收的 MIC 值是否相同，如果不同，忽略该分组，否则执行步骤 c)。
- c) 构造通信密钥确认分组。其中 BSID 为 SS 当前连接的 BS 标志，SAID 与通信密钥请求分组中的 SAID 相同，TEKIDNew 为指向新安装的 TEK 的索引值，N1BS 拷贝自通信密钥响应分组的 N1BS 字段，MIC 使用 TEKID 指向的 TEK 推导的完整性密钥来计算。
- d) 更新第 a) 步获得的 LifeTEKOld 字段的密钥生命周期参数以及 TEKIDOld 字段到本地 TEK 状态机，根据第 a) 步获得的 TEK 和 LifeTEKNew 字段的密钥生命周期参数以及 TEKIDNew 字段创建新的 TEK 状态机，并作为新的 TEK。此时用户设备可继续使用旧的 TEK 推导的加密密钥解密，在旧的 TEK 生命周期过期前使用新的 TEK 推导的 TEK 解密。用户设备使用新的 TEK 推导的密钥加密。

C.2.4.5 通信密钥确认分组

通信密钥确认分组数据字段格式如下表 C.12。

表 C.12 通信密钥确认分组数据字段格式

FLAG	PFLAG	BSID	SSID	SAID	TEKIDOld	TEKIDNew	N1BS	MIC
------	-------	------	------	------	----------	----------	------	-----

其中：

- FLAG 字段长度为 1 个八位位组，高 4 位标识安全方法类别；
- PFLAG 字段长度为 1 个八位位组，表示该消息所属的协议类别，值为 0x34；
- BSID 字段长度为 6 个八位位组，表示 BS 设备的唯一标志符；
- SSID 字段长度为 6 个八位位组，表示 SS 设备的唯一标志符；
- SAID 字段长度为 2 个八位位组，表示需要更新的通信密钥所属的安全关联；
- TEKIDold 字段长度为 1 个八位位组，表示用户设备旧的 TEK 索引值；
- TEKIDNew 字段长度为 1 个八位位组，表示用户设备新的 TEK 索引值；
- N1BS 字段长度为 16 个八位位组，该字段拷贝自通信密钥响应分组的 N1BS 字段；
- MIC 字段长度为 32 个八位位组，表示该消息数据的完整性校验值，该字段使用 TEKID 所指向的 TEKM 中推导的完整性密钥计算。

在 SS 处理完 BS 的通信密钥响应分组后，SS 构造上述分组。

BS 通过 FLAG 字段和 PFLAG 字段确认接收到 SS 发送的消息为本方案的通信密钥确认分组后，进行如下处理：

- a) 使用 TEKIDNew 指向的 TEKM 推导出完整性密钥，使用本地的 BSID，通信密钥相应分组的 SAID，本地 TEKIDold，TEKIDNew，本地的 N1BS 计算 MIC 值，如果该值与通信密钥确认分组中的 MIC 值不同，忽略该分组，否则执行步骤 b)。
- b) 生成新的 TEK 替换该 SAID 确认的连接中的 TEKIDNew 指向的密钥，旧的 TEK 使用该分组中的 TEKIDold 指向。BS 可以使用旧的 TEK 进行加密，在旧的 TEK 生命周期过期前使用新的 TEKM 推导的 TEK 加密。BS 使用新的 TEKM 推导的密钥解密。

C.2.5 组播通信密钥管理

C.2.5.1 概述

组播通信密钥管理完成 BS 与组内每一个连接 SS 的组播密钥加密密钥 GKEK 和完整性密钥 GKIK 的同步，并完成组播通信加密密钥 GTEK 的同步。涉及图 C.5、图 C.6、图 C.7 消息流程。

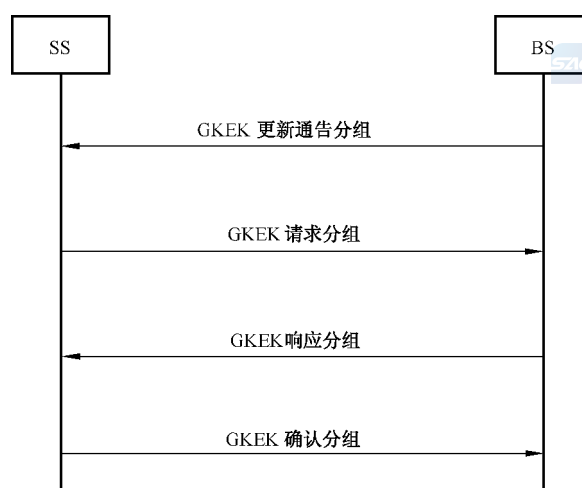


图 C.5 组播密钥加密密钥管理协议

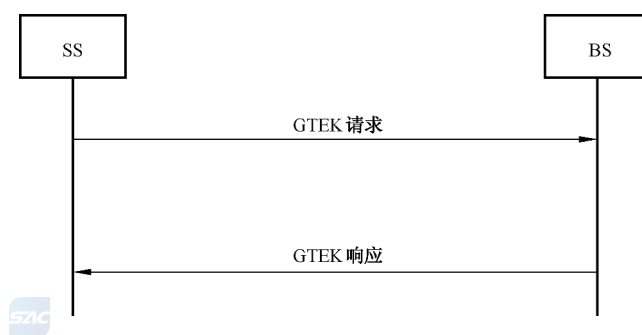


图 C.6 组播通信加密密钥管理协议



图 C.7 组播通信加密密钥广播协议

C.2.5.2 GKEK 更新通告分组

GKEK 更新通告分组数据字段格式见表 C.13。

表 C.13 GKEK 更新通告分组数据字段格式

FLAG	PFLAG	GSAID	BSID	GTEKID	AKID	MIC
------	-------	-------	------	--------	------	-----

其中：

- FLAG 字段长度为 1 个八位位组，高 4 位标识安全方法类别；
- PFLAG 字段长度为 1 个八位位组，表示该消息所属的协议类别，值为 0x41；
- GSAID 字段长度为 2 个八位位组，表示需要更新的 GKEK 所属的安全关联；
- BSID 字段长度为 6 个八位位组，表示 BS 的标志；
- GTEKID 字段长度为 1 个八位位组，表示需要更新的 GKEK 索引值，索引值最低 2 位有效，其他位为 0；
- AKID 字段长度为 1 个八位位组，表明该分组 MIC 字段所使用的 AK 的索引；
- MIC 字段长度为 32 个八位位组，表示该消息数据的完整性校验值，使用 AKID 对应的授权密钥 AK 导出的授权完整性密钥计算。

在 BS 的 GKEK 状态机指示当前 GKEK 需要更新时，BS 发送该分组通告 SS 开始更新 GKEK 密钥。

SS 通过 FLAG 字段和 PFLAG 字段确认接收到 BS 发送的消息为本方案的 GKEK 更新通告分组后，进行如下处理：

- a) 用户设备使用 AKID 指向的完整性密钥使用本地的 BSID 重新计算 MIC，如果接收的 MIC 与本地计算的 MIC 不同，则忽略该分组，不进行任何处理；否则执行步骤 b)；
- b) 检查 GSAID 是否是当前激活的安全连接，如果不是当前激活的安全连接，则忽略 GKEK 更新通告分组，不进行任何处理，否则执行 c)；

- c) SS 构造 GKEK 请求分组, GKEK 请求分组的 GSAID 字段拷贝自通信密钥更新通告消息, GKEK 请求分组的 GTEKID 字段拷贝自通信密钥更新通告消息。

C.2.5.3 GKEK 请求分组

GKEK 请求分组数据字段格式如下表 C.14。

表 C.14 GKEK 请求分组数据字段格式

FLAG	PFLAG	SSID	GSAID	AKID	GKEKID	N2SS	MIC
------	-------	------	-------	------	--------	------	-----

其中:

- FLAG 字段长度为 1 个八位位组,高 4 位标识安全方法类别;
- PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别,值为 0x42;
- SSID 字段长度为 6 个八位位组,表示用户设备的唯一标志符;
- GSAID 字段长度为 2 个八位位组,表示需要更新的 GKEK 安全关联;
- AKID 字段长度为 1 个八位位组,表示客户端当前使用的授权密钥索引值;
- GKEKID 字段长度为 1 个八位位组,表示需要更新的 GKEK 索引值,或者表示用户设备为该 GSAID 生成的新的 GKEK 索引值;
- N2SS 字段长度为 16 个八位位组,表示用户设备生成的 128 比特随机数;
- MIC 字段长度为 32 个八位位组,表示该消息数据的完整性校验值,使用 AKID 对应的授权密钥 AK 导出的授权完整性密钥计算。

在 SS 的 GKEK 状态机指示当前 GKEK 需要更新时,SS 构造并发送该分组请求 BS 传输新的 GKEK。或者在 SS 接收到 BS 的 GKEK 更新通告时,SS 构造上述分组。或者在 SS 状态机指示需要为某个 GSAID 申请 GKEK 密钥时,SS 构造上述分组。

BS 通过 FLAG 字段和 PFLAG 字段确认接收到 SS 发送的消息为本方案的 GKEK 请求分组后,进行如下处理:

- a) 使用 AKID 指向的完整性密钥计算 MIC,重新计算接收的数据,如果校验通过,执行步骤 b);
- b) 检查 SSID 是否是 BS 的当前连接用户,如果不是 BS 的当前连接用户,则忽略该分组,不进行任何处理;否则执行步骤 b);
- c) 检查 GSAID 是否是 BS 与 SSID 激活的组播安全连接,如果是当前激活的安全连接则执行步骤 c),否则检查该 GSAID 是否属于该 SS 的授权安全连接,如果属于该 SSID 的授权安全连接,则执行步骤 c),否则忽略通信密钥更新通告分组,不进行任何处理;
- d) 如果 BS 与 SSID 在 GSAID 确定的连接中不存在 GKEKID,则生成新的 GKEK,并使用通信密钥请求分组中用户设备生成的 GKEKID 指向,之后执行步骤 e);
- e) BS 构造 GKEK 响应分组,其中的 SSID 字段、GSAID 字段、N2SS 字段拷贝自 GKEK 请求分组,N2BS 字段存放 BS 为该 SSID 更新 GKEK 期间生成的新的 N2BS 随机数,GKEKID 字段的值是 GKEK 请求分组的翻转值,Key 字段使用协商的组播通信密钥加密算法和 GKEK 响应分组中 AKID 字段指向的数据加密密钥加密,加密的内容为 BS 中该 SSID 的 GSAID 确定的连接中新生成的 GKEKM,密钥参数填充被加密的 GKEKM 的生命周期 KeyLife,MIC 字段使用被加密的 GKEK 推导的完整性密钥计算。

C.2.5.4 GKEK 响应分组

GKEK 响应分组数据字段格式见表 C.15。

表 C.15 GKEK 响应分组数据字段格式

FLAG		PFLAG	SSID	BSID	GSAID	GKEKID	AKID
N2SS	N2BS	Key	KeyLife		MIC		

其中：

- FLAG 字段长度为 1 个八位位组，高 4 位标识安全方法类别；
- PFLAG 字段长度为 1 个八位位组，表示该消息所属的协议类别，值为 0x43；
- SSID 字段长度为 6 个八位位组，表示用户设备的唯一标志符；
- BSID 字段长度为 6 个八位位组，表示 BS 的唯一标志符；
- GSAID 字段长度为 2 个八位位组，表示需要更新的 GKEK 所属的安全关联；
- AKID 字段长度为 1 个八位位组，表示 BS 当前使用的授权密钥索引值；
- GKEKID 字段长度为 1 个八位位组，表示更新后的 GKEK 索引值；
- N2SS 字段长度为 16 个八位位组，表示用户设备生成的 128 比特随机数；
- N2BS 字段长度为 16 个八位位组，表示 BS 生成的 128 比特随机数；
- Key 字段长度为 16 个八位位组，表示加密新的 GKEKM 后生成的密文分组；
- KeyLife 字段长度为 4 个八位位组，表示与密钥对应的生命周期；
- MIC 字段长度为 32 个八位位组，表示该消息数据的完整性校验值，该字段使用 GKEKID 所指向的 GKEKM 推导的完整性密钥计算。

BS 接收到 GKEK 请求分组后，构造上述 GKEK 响应分组。

SS 通过 FLAG 字段和 PFLAG 字段确认接收到 BS 发送的消息为本方案的 GKEK 响应分组后，进行如下处理：

- a) 使用 AKID 指向的密钥加密密钥解密 Key 字段的密文，获得新的 GKEKM。执行步骤 b)。
- b) 使用新的 GKEKID 指向的 GKEKM 推导的完整性密钥，使用本地的 SSID，本地当前连接的 BSID，GKEK 请求分组中的 GSAID，GKEK 响应分组中的 AKID 和 GKEKID，本地的 N2SS，GKEK 响应分组中的 N2BS、Key 字段和 KeyLife 字段来重新计算 MIC 值，比较本地计算的 MIC 值与接收的 MIC 值是否相同，如果不同，忽略该分组，否则执行步骤 c)。
- c) 构造 GKEK 确认分组，其中 BSID 为 SS 当前连接的 BS 标志，GSAID 与 GKEK 请求分组中的 SAID 相同，GKEKID 为指向新安装的 GKEK 的索引值，N2BS 拷贝自 GKEK 响应分组的 N2BS 字段，MIC 使用 GKEKID 指向的 GKEKM 推导的完整性密钥来计算。
- d) 安装第 a) 步获得的 GKEKM 和 KeyLife 字段的密钥生命周期参数以及 GKEKID 字段的 GKEKID 到本地 GKEKM 状态机，作为新的 GKEKM，设置当前的 GKEKM 作为旧的 GKEKM，并使用新的 GKEKM 的翻转值指向。此时用户设备继续使用旧的 GKEKM 推导的密钥材料解密，在解密完整性校验出错后使用新的 GKEKM 推导的密钥材料解密。

C.2.5.5 GKEK 确认分组

GKEK 确认分组数据字段格式如下表 C.16。

表 C.16 GKEK 确认分组数据字段格式

FLAG	PFLAG	BSID	SSID	GSAID	AKID	GKEKID	N2BS	MIC
------	-------	------	------	-------	------	--------	------	-----

其中：

- FLAG 字段长度为 1 个八位位组，高 4 位标识安全方法类别；

- PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别,值为 0x44;
- BSID 字段长度为 6 个八位位组,表示 BS 设备的唯一标志符;
- SSID 字段长度为 6 个八位位组,表示 BS 设备的唯一标志符;
- GSAID 字段长度为 2 个八位位组,表示需要更新的 GKEK 所属的安全关联;
- AKID 字段长度为 1 个八位位组,表示客户端当前使用的授权密钥索引值;
- GKEKID 字段长度为 1 个八位位组,表示 SS 当前使用的 GKEK 索引值;
- N2BS 字段长度为 16 个八位位组,该字段拷贝自 GKEK 响应分组的 N2BS 字段;
- MIC 字段长度为 32 个八位位组,表示该消息数据的完整性校验值,该字段使用 GKEKID 所指向的 GKEKM 推导的完整性密钥计算。

在 SS 处理完 BS 的 GKEK 响应分组后,SS 构造上述分组。

BS 通过 FLAG 字段和 PFLAG 字段确认接收到 SS 发送的消息为本方案的 GKEK 确认分组后,进行如下处理:

- a) 使用接收的确认分组中 GKEKID 指向的 GKEKM 推导的出完整性密钥,使用本地的 BSID, GKEK 响应分组的 GSAID,本地 GKEKID 的翻转值,本地的 N2BS 计算 MIC 值,如果该值与 GKEK 确认分组中的 MIC 值不同,忽略该分组,否则执行步骤 b);
- b) 生成新的 GKEKM 替换该本地 GKEKID 指向的密钥材料,更新 GKEKID 为 GKEKID 的翻转值。

C.2.5.6 GTEK 请求分组

GTEK 请求分组数据字段格式见表 C.17。

表 C.17 GTEK 请求分组数据字段格式

FLAG	PFLAG	SSID	GSAID	GKEKID	GTEKID	N3SS	MIC
------	-------	------	-------	--------	--------	------	-----

其中:

- FLAG 字段长度为 1 个八位位组,高 4 位标识安全方法类别;
- PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别,值为 0x51;
- SSID 字段长度为 6 个八位位组,表示用户设备的唯一标志符;
- GSAID 字段长度为 2 个八位位组,表示需要更新的 GTEK 所属安全关联;
- GKEKID 字段长度为 1 个八位位组,表示客户端当前使用的 GKEK 索引值;
- GTEKID 字段长度为 1 个八位位组,表示需要更新的 GTEK 索引值,或者表示用户设备为该 SAID 生成的新的 GTEK 索引值;
- N3SS 字段长度为 16 个八位位组,表示用户设备生成的 128 比特随机数;
- MIC 字段长度为 32 个八位位组,表示该消息数据的完整性校验值,该字段使用 GKEKID 所指示的 GKEKM 推导的完整性密钥计算。

在 SS 的 GTEK 状态机指示当前 GKEK 需要更新时,SS 构造并发送该分组请求 BS 传输新的 GTEK。或者在 SS 状态机指示需要为某个 GSAID 申请 GTEK 密钥时,SS 构造上述分组。

BS 通过 FLAG 字段和 PFLAG 字段确认接收到 SS 发送的消息为本方案的 GTEK 请求分组后,进行如下处理:

- a) 使用 GKEKID 指向的 GKEKM 推导的完整性密钥 GKI 计算 MIC,重新计算接收的数据,如果校验通过,执行步骤 b);
- b) 检查 SSID 是否是 BS 的当前连接用户,如果不是 BS 的当前连接用户,则忽略该分组,不进行处理;否则执行步骤 c);

- c) 检查 GSAID 是否是 BS 与 SSID 激活的组播安全连接,如果是当前激活的安全连接则执行步骤 d),否则检查该 GSAID 是否属于该 SS 的授权安全连接,如果属于该 SSID 的授权安全连接,则执行步骤 d),否则忽略 GTEK 请求分组,不进行任何处理;
- d) 如果 BS 与 SSID 在 GSAID 确定的连接中不存在 GTEKID,则生成新的 GTEK,并使用 GTEK 请求分组中用户设备生成的 GTEKID 指向,之后执行步骤 e);
- e) BS 构造 GTEK 响应分组,其中的 SSID 字段、GSAID 字段、GKEKID 字段、N3SS 字段拷贝自 GTEK 请求分组,GTEKID 字段的值是 GTEK 请求分组的翻转值,Key 字段使用协商的通信密钥加密算法和 GTEK 请求分组中 GKEKID 字段指向的数据加密密钥 GKEK 加密,加密的内容为 BS 中该 SSID 的 GSAID 确定的连接中新生成的 GTEK,密钥参数填充被加密的 GTEK 的生命周期 KeyLife,MIC 字段使用被加密的 GTEK 推导的完整性密钥计算。

C.2.5.7 GTEK 响应分组

GTEK 响应分组数据字段格式见表 C.18。

表 C.18 GTEK 响应分组数据字段格式

FLAG	PFLAG	SSID	BSID	GSAID	GKEKID	
GTEKID	N3SS		Key	KeyLife		MIC

其中:

- FLAG 字段长度为 1 个八位位组,高 4 位标识安全方法类别;
- PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别,值为 0x52;
- SSID 字段长度为 6 个八位位组,表示用户设备的唯一标志符;
- BSID 字段长度为 6 个八位位组,表示 BS 的唯一标志符;
- GSAID 字段长度为 2 个八位位组,表示需要更新的 GTEK 所属的安全关联;
- GKEKID 字段长度为 1 个八位位组,表示 BS 当前使用的 GKEK 索引值;
- GTEKID 字段长度为 1 个八位位组,表示更新后的 GTEK 索引值,索引值最低 2 位有效,其他位为 0;
- N3SS 字段长度为 16 个八位位组,表示用户设备生成的 128 比特随机数;
- Key 字段长度为 16 个八位位组,表示加密新的 GTEK 后生成的密文分组,使用的加密密钥由 GKEKID 指向;
- KeyLife 字段长度为 4 个八位位组,表示与密钥对应的生命周期;
- MIC 字段长度为 32 个八位位组,表示该消息数据的完整性校验值,该字段使用 GTEKID 所指向的 GTEKM 推导的完整性密钥计算。

BS 接收到 GTEK 请求分组后,构造上述 GTEK 响应分组。

SS 通过 FLAG 字段和 PFLAG 字段确认接收到 BS 发送的消息为本方案的 GTEK 分组后,进行如下处理:

- a) 使用 GKEKID 指向的密钥加密密钥解密 Key 字段的密文,获得新的 GTEK,解密完整性校验出错则忽略该密文分组,否则执行步骤 b)。
- b) 使用新的 GTEKID 指向的 GTEKM 推导的完整性密钥,使用本地的 SSID,本地当前连接的 BSID,GTEK 请求分组中的 GSAID,GTEK 请求分组中的 GKEKID 和 GTEKID,本地的 N3SS,GTEK 响应分组中的 Key 字段和 KeyLife 字段来重新计算 MIC 值,比较本地计算的 MIC 值与接收的 MIC 值是否相同,如果不同,忽略该分组,否则执行步骤 c)。
- c) 安装第 a) 步获得的 GTEK 和 KeyLife 字段的密钥生命周期参数以及 GTEKID 字段的

GTEKID 到本地 GTEK 状态机,作为新的 GTEK。设置当前的 GTEK 作为旧的 GTEK,并使用新的 GTEKID 的翻转值指向。此时用户设备继续使用旧的 GTEK 解密,在解密完整性校验出错后使用新的 GTEK。

C.2.5.8 GTEK 组内广播分组

GTEK 组内广播分组数据字段格式见表 C.19。

表 C.19 GTEK 组内广播分组数据字段格式

FLAG		PFLAG	BSID	GSAID	GKEKID	GTEKID
Counter	N3SS	Key	KeyLife		MIC	

其中:

- FLAG 字段长度为 1 个八位位组,高 4 位标识安全方法类别;
- PFLAG 字段长度为 1 个八位位组,表示该消息所属的协议类别,值为 0x61;
- BSID 字段长度为 6 个八位位组,表示 BS 的唯一标志符;
- GSAID 字段长度为 2 个八位位组,表示需要更新的 GTEK 所属的安全关联;
- GKEKID 字段长度为 1 个八位位组,表示 BS 当前使用的 GKEK 索引值;
- GTEKID 字段长度为 1 个八位位组,表示更新后的 GTEK 索引值,索引值最低 2 位有效,其他位为 0;
- Counter 字段长度为 32 个八位位组,表示 BS 为该 GSAID 发送组播 GTEK 更新消息的计数值,每发送一次更新消息,该计数值增加 1;
- N3SS 字段长度为 16 个八位位组,表示用户设备生成的 128 比特随机数;
- Key 字段长度为 16 个八位位组,表示利用协商的密码算法加密新的 GTEK 后生成的密文字段,使用的加密密钥由 GKEKID 指向;
- KeyLife 字段长度为 4 个八位位组,表示与密钥对应的生命周期;
- MIC 字段长度为 32 个八位位组,表示该消息数据的完整性校验值,该字段使用 GKEKID 所指示的 GKEKM 推导的完整性密钥计算。

BS 的 GTEK 状态机触发 GTEK 更新后,构造上述 GTEK 响应分组。

SS 通过 FLAG 字段和 PFLAG 字段确认接收到 BS 发送的消息为本方案的 GTEK 组内 GTEK 广播分组后,进行如下处理:

- a) 检查 GSAID 是否是当前激活的 GSAID 中的一个,如果是则执行步骤 b),否则忽略该分组。
- b) 在该 GSAID 确定的连接中,使用 GKEKID 指向的密钥加密密钥解密 Key 字段的密文,获得新的 GTEK。解密完整性校验出错则忽略该密文分组,否则执行步骤 c)。
- c) 使用 GKEKID 指向的 GKEKM 推导的完整性密钥,本地当前连接的 BSID, GTEK 请求分组中的其他字段来重新计算 MIC 值,比较本地计算的 MIC 值与接收的 MIC 值是否相同,如果不同,忽略该分组,否则执行步骤 d)。
- d) 比较本地的 Counter 字段与接收的 Counter 字段的值,如果本地的 Counter 字段小,则执行步骤 e),否则忽略该分组。
- e) 安装第 b)步获得的 GTEK 和 KeyLife 字段的密钥生命周期参数以及 GTEKID 字段的 GTEKID 到本地 GTEK 状态机,作为新的 GTEK。设置当前的 GTEK 为旧的 GTEK,并使用新的 GTEKID 的翻转值指向。此时用户设备继续使用旧的 GTEK 解密,在解密完整性校验出错后使用新的 GTEK。

附 录 D

(资料性附录)

局域网媒体访问控制技术

D.1 范围

本附录规定了基于 TePA 的有线局域网媒体访问控制安全 TLSec(TePA-based wired LAN MAC Security)技术的整体方案,主要涉及基于 TePA 的局域网鉴别协议 TLA(TePA-based LAN Authentication Protocol)以及基于 TLA 协议的局域网保密通信协议 TLP(TLA-based LAN Privacy Protocol)。TLA 协议为有线局域网节点合法访问提供保障,TLP 协议为有线局域网节点之间保密数据通信提供保障。

本附录适用于有线局域网,为有线局域网客户端和网络控制(交换)设备提供媒体访问控制层安全接入控制和保密通信功能。本附录适用于具有安全增强功能的有线局域网媒体访问控制产品的开发、生产。

D.2 概述

本附录旨在解决有线局域网 MAC 层安全问题,TLSec 是一种基于三元对等鉴别的局域网媒体访问控制安全技术,该技术具备如下特点:

- a) 可实现用户与网络接入控制设备之间的直接身份鉴别;
- b) 可实现用户与网络接入控制设备之间的预共享密钥鉴别;
- c) 可实现用于链路层数据保护的密钥的协商及动态更新管理;
- d) 可保障数据的机密性、完整性、源鉴别及防重放;
- e) 支持企业网、电信网等多种网络架构;
- f) 具有高度可扩展性,支持多种鉴别方法;
- g) 采用对等思想,具有普适性;
- h) 最大化利用了现有协议,如 X.509 等;
- i) 协议的子模块相互独立、灵活,便于取舍。

D.3 安全策略整体方案

D.3.1 系统实体

本附录定义了一种三实体三元结构,见图 D.1。基于对等控制思想,可完成用户与网络之间直接的对等双向鉴别,为有线客户端和网络控制(交换)设备提供安全的接入控制与高强度的保密通信功能。根据用户或设备身份 ID(Identity),利用鉴别服务器 AS(Authentication Server)对网络客户端(或端口)和局域网控制设备(或端口)进行鉴别,该鉴别被称为“端口级别的鉴别”。通过这种鉴别可提供自动的用户与网络身份识别,可实现集中鉴别、密钥管理和 LAN 连接配置等功能。

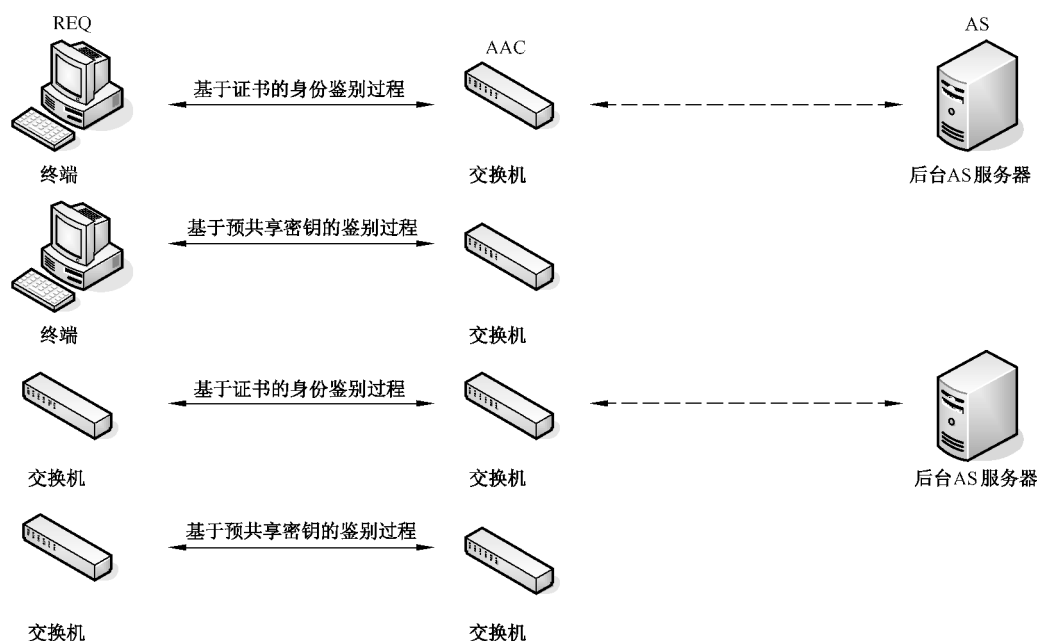


图 D.1 三元架构示意图

本附录中的系统包括三个实体部分：

- 请求者 REQ (Requester)：是位于局域网链路一端的实体，由连接到该链路另一端的鉴别服务器对其进行鉴别。REQ 通常驻留在用户终端或网络接入、控制和交换设备上，用户通过启动客户端软件发起鉴别过程。
- 鉴别访问控制器 AAC (Authentication Access Controller)：通过链路另一端的鉴别服务器对连接到链路对端的 REQ 进行鉴别，也可通过该鉴别服务器完成 REQ 对其的鉴别。AAC 通常驻留在网络接入、控制或交换等设备上，如访问控制器、交换机等，为 REQ 提供服务端口，该端口可以是物理端口也可以是逻辑端口。
- 鉴别服务器 AS (Authentication Server)：为 REQ 和 AAC 提供鉴别服务，是二者所信任的第三方实体。

其中鉴别访问控制器实体和请求者实体都称为鉴别子系统。

D.3.2 系统端口

请求者 REQ 和鉴别访问控制器 AAC 驻留的设备均提供两种访问 LAN 的逻辑通道，定义为两类端口，即受控端口与非受控端口。它们是系统和链路的连接点上建立的两个不同的访问点。鉴别子系统受控端口和非受控端口见图 D.2。

非受控端口，不论授权状态如何，允许链路上的系统之间不受控地交换数据包；受控端口，只有处于授权状态，才允许交换数据包。非受控端口和受控端口是同一连接点的两个部分，从物理连接点得到的数据帧会被它们同时得到。

系统与链路的连接点可以是物理端口，也可以是逻辑端口，该端口提供到其他系统的一对一连接。在交换的 LAN 中，该连接点可以通过 MAC 地址实现，MAC 地址使鉴别访问控制器和请求者之间一对多的关系成为可能。

受控端口和非受控端口是逻辑概念，它将数据流进行了分类。管理和控制流可以通过非受控端口，信息流则通过受控端口，在不影响系统通信管理的前提下，通过受控端口对信息流进行控制。

非受控端口允许管理和控制数据在 LAN 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,才允许用户数据通过。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)。

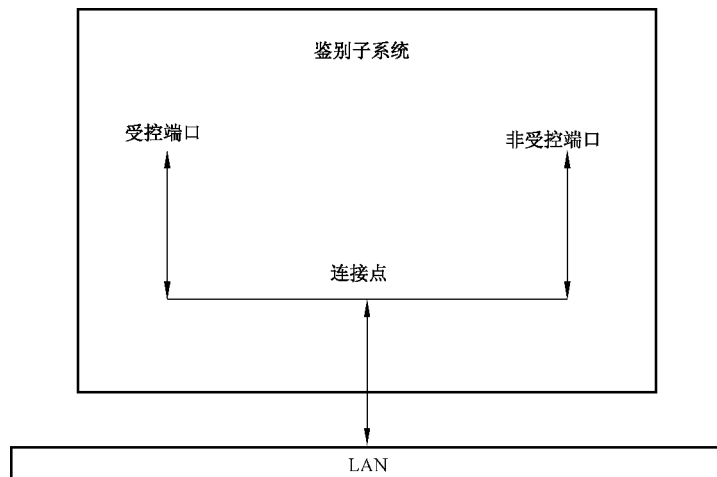


图 D.2 鉴别子系统示意图

受控端口具有两种不同的鉴别状态:On 和 Off,分别允许或拒绝受控端口的协议数据单元 MPDU (MAC Protocol Data Unit)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,见图 D.3。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 MPDU 通过。

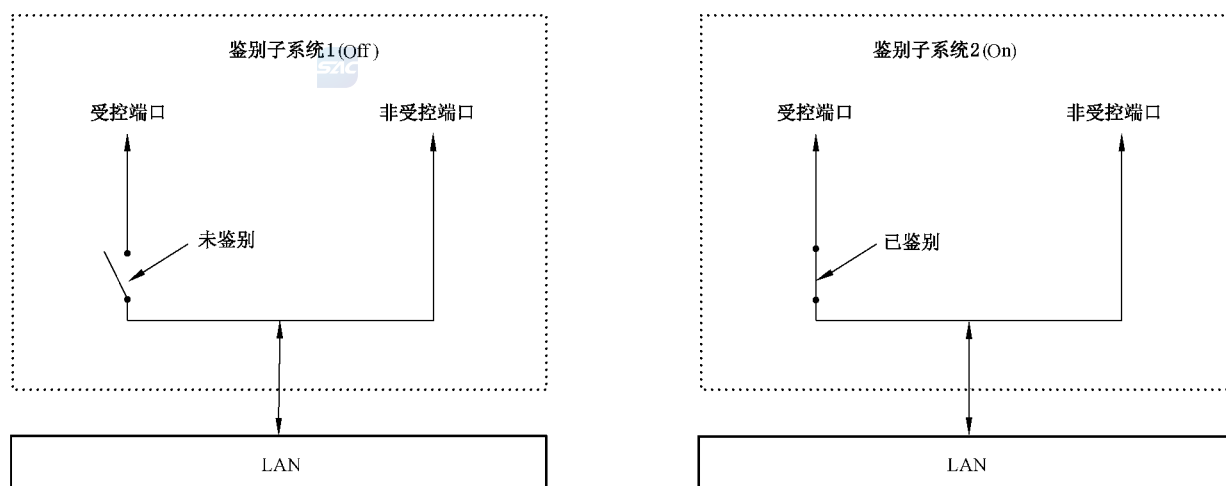


图 D.3 受控端口的鉴别状态

系统的每一个受控端口状态由系统鉴别控制参数 AuthControlledPortStatus 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 AuthControlledPortControl 决定,鉴别控制类型取值如下:

- a) 强制非鉴别:鉴别访问控制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:是指根据鉴别访问控制器和请求者实体之间通过鉴别服务实体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 REQ 与 AAC 之间的网络协议数据交换是通过受控端口来实现的。受控端口和非受控端口的逻辑结构图见图 D.4,系统中受控端口的鉴别状态是由鉴别访问控制器根据 AS 对 REQ 的鉴别结果来设定的。

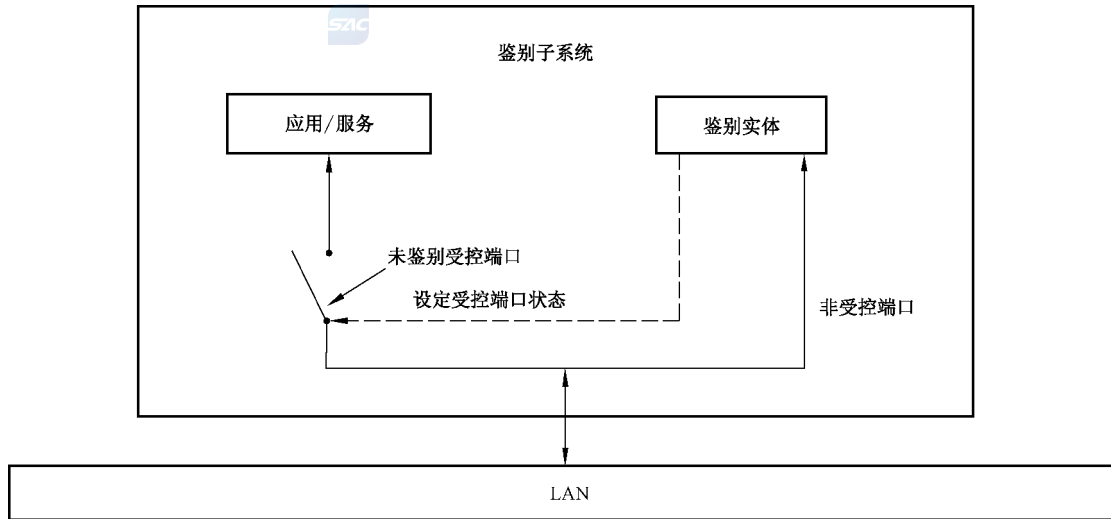


图 D.4 受控端口和非受控端口的用法

请求者、鉴别访问控制器和鉴别服务器之间的关系及信息交换过程见图 D.5。在该图中,鉴别访问控制器和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。AAC 与 AS 之间不需要专用网络。

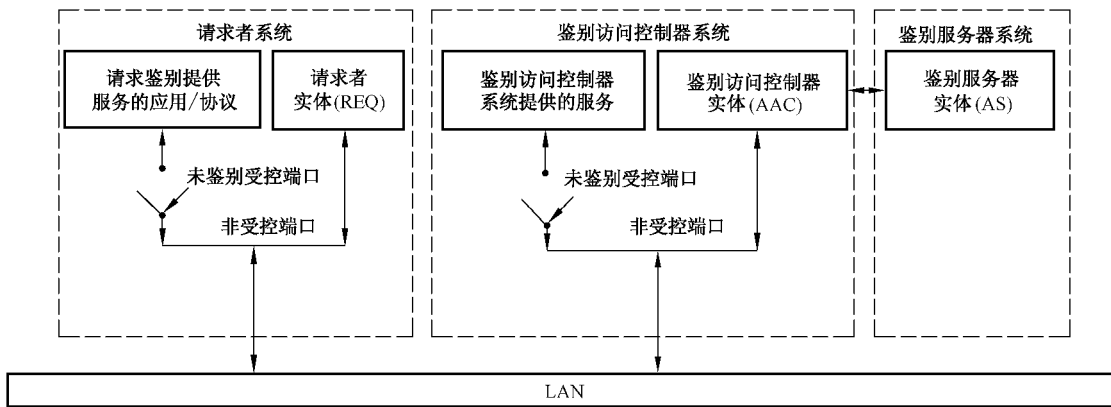


图 D.5 鉴别系统结构

D.3.3 系统安全策略

在 LAN 系统中,REQ 驻留在用户终端或交换设备上,AAC 驻留在交换设备上,AS 可驻留在局域网交换设备上或局域网服务器上,也可以驻留在 Internet 中的服务器上。若存在多个交换设备,则用户终端作为 REQ 所对应的 AAC 驻留的交换设备应为距用户最近的交换设备。在 LAN 系统中,存在多级交换设备时,交换设备之间的安全性保障可采用本协议在网络初始化运行时进行保护,完成合法性确认,但考虑到数据传输的效率,建议:

- a) 在一般网络应用中,交换设备之间,即网络基础设施之间进行身份合法性验证,交换设备和用户终端之间进行身份合法性验证,可以不需要数据加密传输;
- b) 在专网或安全等级要求较高的网络中,除交换设备、交换设备和用户之间完成身份认证之外,还需进行用户数据传输的加密保护。

鉴别系统中,REQ 和 AAC 所驻留设备的每个物理端口内部都包含有受控端口和非受控端口。非受控端口始终处于双向连通状态,主要用来传递 TAEPoL 协议帧,可随时保证接收 REQ 和 AAC 实体发出的 TAEPoL 封装的鉴别分组,受控端口只有在鉴别通过的状态下才打开,用于传递网络资源和服务,即收发非 TAEPoL 的其他协议数据。

TLSec 技术框架示意图见图 D.6。



图 D.6 TLSec 技术框架示意图

TLSec 主要包含两个协议:基于 TePA 的局域网鉴别协议 TLA 和基于 TLA 的局域网保密通信协议 TLP。基于 TePA 的局域网鉴别协议 TLA 主要包含邻居节点发现过程、安全策略协商过程、鉴别及单播密钥协商过程、组播密钥通告过程、交换密钥建立过程以及交换路径探寻过程。其中:

- a) 邻居节点过程,在第 7 章进行介绍,实现了节点获得邻居节点信息;
- b) 安全策略协商,在第 8 章进行介绍,实现对鉴别与密钥套件等安全策略的协商,TLA 支持两种鉴别和密钥管理方法,一种是基于证书的鉴别和密钥管理方法,一种是基于预共享密钥的鉴别和密钥管理方法;
- c) 鉴别和单播密钥协商,在第 9 章进行介绍,根据安全策略协商所选择的套件进行鉴别并建立相邻节点之间的安全路径;
- d) 组播密钥通告,在第 10 章进行介绍,实现了局域网组播密钥的发放;
- e) 站间密钥建立,在第 11 章进行介绍,为需要建立站间密钥的用户终端建立安全路径;
- f) 交换密钥建立,在第 12 章进行介绍,为局域网所有交换设备建立两两之间的安全路径;
- g) 交换路径探寻,在 13.1 进行介绍,建立发送节点到目的节点之间的保密通信连接,获得交换路径信息,用于后续的数据保密通信过程。

基于 TLA 的局域网保密通信协议 TLP 主要包含数据保密通信过程,在第 13 章进行介绍,该保密通信过程使用 TLA 过程建立好的密钥及得到的交换路径信息来保证局域网节点之间的数据通信的机

密性。

TLSec 中的 TLA 将构建一种三段式的安全网络架构(例如 REQ_1 与 AAC_1 之间的安全链路、 AAC_1 与 AAC_2 之间的安全链路以及 AAC_2 与 REQ_2 之间的安全链路)。TLP 就是根据得到的交换路径信息,选择合适的三段式安全路径,采取一种三段式的加密模式实现数据的保密传输。

D.3.4 安全关联的管理

D.3.4.1 安全关联的定义

安全关联是一组用来保护信息的策略和密钥,TLSec 中包含 6 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;
- SWBKSA:交换基密钥安全关联,交换基密钥通告的结果;
- SWkeySA:交换密钥安全关联,交换密钥协商的结果。

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 AAC 或 REQ 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;
- REQ 的 MAC 地址;
- AAC 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 AAC 和 REQ,最多只有两个 USKSA。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。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;
- AAC 的 MAC 地址;
- REQ 的 MAC 地址;
- USK;
- 生存期;
- 选择的单播密码套件;
- 其他安全参数,比如重放计数器等。

MSKSA

MSKSA 是组播密钥通告的结果。MSK 是由 AAC 通告给 REQ 的,在生存期中被缓存。MSK 最多只有两个 MSKSA。一般只有一个 MSKSA 处于有效状态,但在密钥更新时,会有两个 MSKSA 处于有效状态,在接收到使用新 MSKSA 加密的广播/组播 MPDU 后,旧的 MSKSA 才被置为无效状态。一个 MSKSA 包含以下内容:

- MSKID;
- REQ 的 MAC 地址;
- MSK;
- 生存期;
- 选择的组播密码套件;
- 其他安全参数(可选)。

STakeySA

STakeySA 是 STakey 协商的结果,它是双向的。一个 STakeySA 包含以下内容:

- STakeyID;
- 本端 STA 的 MAC 地址;
- 对端 STA 的 MAC 地址。
- STakey;
- 生存期;
- 采用的单播密码套件(采用交换设备通告的组播密码套件);
- 其他安全参数(可选)。

SWBKSA

SWBKSA 是 SWBK 交换基密钥通告的结果或设置了预共享交换基密钥后在交换设备中创建,它是双向的。SWBKSA 用来创建 SWkeySA, SWBKSA 在它的生存期内被缓存。它包含以下内容:

- SWBKID,标识 SWBKSA;
- 本端 SW 的 MAC 地址;
- 对端 SW 的 MAC 地址;
- SWBK;
- 生存期;
- 其他安全参数(可选)。

SWKeySA

SWKeySA 是交换密钥 SWkey 协商的结果,它是双向的。SWkeySA 是基于 SWBK 协商的,在生存期中被缓存。对于每一对 SW_1 和 SW_2 ,最多只有两个 SWkeySA。一般只有一个 SWkeySA 处于有效状态,但在密钥更新时,会有两个 SWkeySA 处于有效状态,在接收到使用新 SWkeySA 加密的单播数据 MPDU 时,旧 SWkeySA 被置为无效状态。当新的 SWkeySA 处于有效状态时,旧的 SWkeySA 立刻处于无效状态。

SWkeySA 包含以下内容:

- SWkeyID;
- 本端 SW 的 MAC 地址;
- 对端 SW 的 MAC 地址;
- SWkey;
- 生存期;
- 支持的多播密码套件;
- 其他安全参数,比如重放计数器。

D.3.4.2 安全关联的删除

当 STA/SW 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,或当它相信它已经断开和另一个 SW/STA 的连接后,它将删除一些安全关联。

STA 将删除 USKSA、MSKSA、STakeySA;SW 将删除 USKSA、SWBKSA、SWkeySA。

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。

D.4 TLSec 协议帧结构

D.4.1 TLA 协议分组中字段固定格式

D.4.1.1 标识 TAEP_FLAG

标识 TAEP_FLAG 固定格式见图 D.7。



图 D.7 标识 TAEP_FLAG 固定格式

具体每个比特代表的意义见图 D.8。

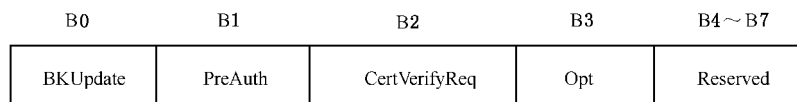


图 D.8 标识 TAEP_FLAG 比特位意义

其中:

- BK 更新标识 BKUpdate:表示是否为 BK 更新过程。如果是 BK 更新过程,此比特位置 1,否则置 0;
- 预鉴别标识 PreAuth:表示是否为预鉴别过程。如果是预鉴别过程,此比特位置 1,否则置 0;
- 证书验证请求标识 CertVerifyReq:表示 REQ 是否请求 AS 对 AAC 进行证书验证;如果需要请求证书验证,此比特位置 1,否则置 0;
- 可选字段标识 Opt:表示此分组是否含有可选字段;如果含有可选字段,此比特位置 1,否则置 0;
- 保留 Reserved:供以后扩展使用。

D.4.1.2 鉴别标识 SNonce

鉴别标识 SNonce 固定格式见图 D.9。

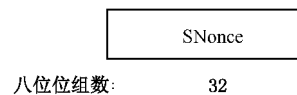


图 D.9 鉴别标识 SNonce 固定格式 1

其中:

- 鉴别标识 SNonce 字段:长度为 32 个八位位组,表示一个整数,用于鉴别更新时双方的同步锁定。

D.4.1.3 随机数 N

随机数 N 固定格式见图 D.10。

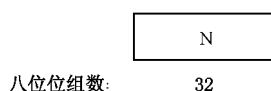


图 D.10 随机数 N 固定格式

其中:

——随机数 N 字段,长为 32 个八位位组,是由随机数生成算法生成的一个整数,也称为本地询问,即本地生成的随机数。

D.4.1.4 身份 ID

身份 ID 固定格式见图 D.11。

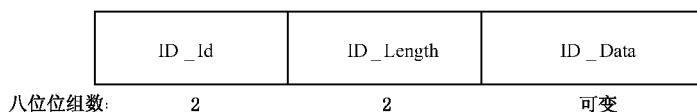


图 D.11 身份 ID 固定格式

其中:

——身份标识 ID_Id 字段:表示身份类型,长度为 2 个八位位组。身份标识:1 表示该字段的身份数据 ID_Data 字段:由 X.509 v3 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 ID_Length 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 ID_Data 字段:为从证书中提取出的持有者名称 ID_DataHolder、颁发者名称 ID_DataLicensor、序列号字段 ID_DataSN。身份数据字段定义见图 D.12。



图 D.12 身份数据 ID_Data 字段定义

其中:

——持有者名称 ID_DataHolder 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;

——颁发者名称 ID_DataLicensor 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;

——序列号 ID_DataSN 字段:长度为 4 个八位位组,表示一个整数。

D.4.1.5 证书 Cert

证书 Cert 固定格式见图 D.13。

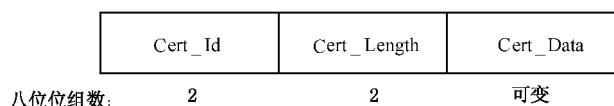


图 D.13 证书 Cert 固定格式

其中:

——证书标识 Cert_Id 字段:表示证书类型,长度为 2 个八位位组。证书标识定义:1 表示该字段的

证书数据为 X.509 v3 证书;其他值保留。

——证书长度 Cert_Length 字段:为证书数据字段的八位位组数。

——证书数据 Cert_Data 字段:为该属性字段的内容。

D.4.1.6 ECDH 参数 Para_{ECDH}

ECDH 参数 Para_{ECDH} 固定格式见图 D.14。

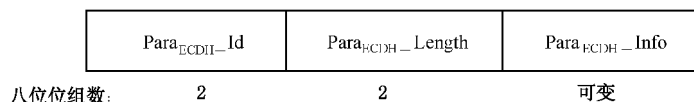


图 D.14 ECDH 参数 Para_{ECDH} 固定格式

其中:

——参数标识 Para_{ECDH}_Id 字段:表示参数类型。

参数标识为 1 时,参数内容以 OID 方式表示,参数长度字段表示参数内容的八位位组数,参数内容为 OID 编码。本标准采用值为 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。

——参数长度 Para_{ECDH}_Length 字段:表示参数内容的八位位组数。

——参数内容 Para_{ECDH}_Info 字段:为 OID 编码。

D.4.1.7 数字签名 Sig

数字签名 Sig 固定格式见图 D.15。

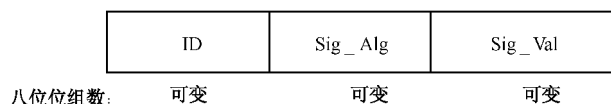


图 D.15 数字签名 Sig 固定格式

其中:

——身份 ID 字段:定义见 D.4.1.4。

——签名算法 Sig_Algorithm 字段:定义见图 D.16。

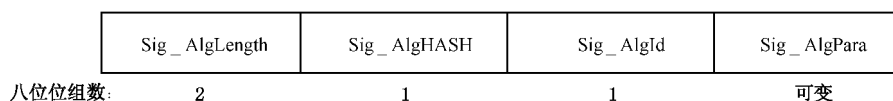


图 D.16 签名算法 Sig_Val 字段定义

其中:

- 长度 Sig_AlgorithmLength 字段:为 2 个八位位组,表示签名算法标识、杂凑算法标识及参数的总长度。
- 签名算法标识 Sig_AlgorithmId 字段,定义如下:
 - ◆ 1 表示 256 位的椭圆曲线数字签名算法,即 ECDSA-256;
 - ◆ 2 表示 384 位的椭圆曲线数字签名算法,即 ECDSA-384;
 - ◆ 3 表示 192 位的椭圆曲线数字签名算法,即 ECDSA-192;
 - ◆ 其他值保留。

- 杂凑算法标识 Sig_Algorithm 字段,定义如下:
 - ◆ 1 表示 SHA-256 杂凑算法;
 - ◆ 其他值保留。
- 参数 Sig_AlgorithmPara 字段:表示签名算法的参数,定义如 D. 4. 1. 6。
- 签名值 Sig_Val 字段:定义见图 D. 17。

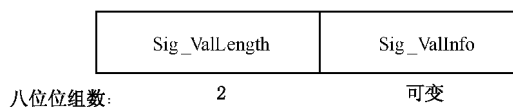


图 D. 17 2 签名值 Sig_Val 字段定义

其中:

- 签名长度 Sig_ValLength 字段:为 2 个八位位组,表示内容字段的八位位组数;
- 签名内容 Sig_ValInfo 字段:为签名的值,将签名结果转化成的八位位组串。

D. 4. 1. 8 MAC 地址

MAC 地址固定格式见图 D. 18。

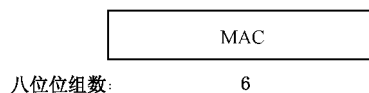


图 D. 18 MAC 地址固定格式

其中:

MAC 地址字段:表示实体的 MAC 地址,长为 6 个八位位组。

D. 4. 1. 9 地址索引 ADDID

地址索引 ADDID 固定格式见图 D. 19。

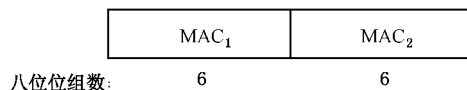


图 D. 19 地址索引 ADDID 固定格式

其中:

MAC 地址字段:表示实体的 MAC 地址,长为 6 个八位位组。

D. 4. 1. 10 消息鉴别码 MIC

消息鉴别码 MIC 固定格式见图 D. 20。

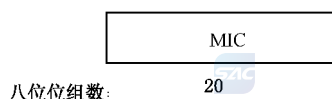


图 D. 20 消息鉴别码 MIC 固定格式

其中:

消息鉴别码 MIC 字段:长度为 20 个八位位组,具体计算方法,在各分组内进行介绍。

D. 4. 1. 11 证书验证结果 RES

证书验证结果 RES 固定格式见图 D. 21。

RES_Length(2)	
N ₁ (32)	
N ₂ (32)	
RES_Result ₁ (1)	Cert ₁ (可变)
RES_Result ₂ (1)	Cert ₂ (可变)
注: 括号内单元为八位位组数。	

图 D. 21 证书验证结果 RES 固定格式

其中:


- 长度 RES_Length 字段: 为后面随机数, 验证结果和证书的总长度。
- 一次性随机数 N 字段: 定义如 D. 4. 1. 3。
- 验证结果 RES_Result 字段: 表示证书的验证结果, 长为 1 个八位位组, 验证结果定义见表 D. 1。

表 D. 1 验证结果字段定义表

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
9~255	保留

——证书字段 Cert: 定义如 D. 4. 1. 5。

注: 如果为单向鉴别过程, 则证书的验证结果中不包含一次性随机数 2, 验证结果 2 以及证书 2。

D. 4. 1. 12 复合的证书验证结果 MRES 

复合证书验证结果 MRES 固定格式见图 D. 22。

RES	Sig
八位位组数: 可变	可变

图 D. 22 复合证书验证结果 MRES 固定格式

其中:

- 证书验证结果 RES 字段: 定义如 D. 4. 1. 11;

——签名 Sig 字段:定义如 D. 4. 1. 7。

D. 4. 1. 13 身份列表 List

身份列表 List 固定格式见图 D. 23。

List_IDNum(2)
ID ₁ (可变)
ID ₂ (可变)
.....
注:括号内单元为八位位组数。

图 D. 23 身份列表 List 固定格式

其中:

——身份个数 List_IDNum 字段:表示身份列表中身份的个数,长为 2 个八位位组;

——身份 ID 字段:定义如 D. 4. 1. 4。

D. 4. 1. 14 TLA 信息元素 TIE

TLA 信息元素 TIE 固定格式见图 D. 24。

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 (USK) 套件计数	单播密码 (USK) 套件	组播密码 (MSK) 套件
八位位组数: 2	4×n	2	4×n	4

注: m 表示鉴别和密钥管理(AKM) 套件计数字段的值, n 表示单播密码套件计数的值。

图 D. 24 信息元素 TIE 固定格式

其中:

——鉴别和密钥管理(AKM)套件计数字段:表示鉴别访问控制器 AAC 或请求者 REQ 支持的或选择的鉴别和密钥管理套件的个数;

——鉴别和密钥管理(AKM)套件字段:表示鉴别访问控制器 AAC 或请求者 REQ 支持的或选择的鉴别和密钥管理套件, m 为鉴别和密钥管理套件计数字段的值;

——单播密码(USK)套件计数字段:表示鉴别访问控制器 AAC 或请求者 REQ 支持的或选择的单播密码套件个数;

——单播密码(USK)套件字段:表示接入设备鉴别访问控制器 AAC 或请求者 REQ 支持的或选择的单播密码套件, n 为单播密码套件计数字段的值;

——组播密码(MSK)套件字段:描述接入设备鉴别访问控制器 AAC 或请求者 REQ 支持的组播密码套件。

套件选择格式见图 D. 25。

OUI	OUI_Type
八位位组数: 3	1

图 D. 25 套件选择格式

鉴别和密钥管理套件类型见表 D. 2。

单播密码套件类别见表 D. 3。

组播密码套件类别见表 D. 4。

表 D. 2 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	TLA 证书鉴别和密钥管理
00-14-72	2	TLA 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

表 D. 3 单播密码套件

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	SMS4-GCM
00-14-72	2~255	保留
其他	0~255	保留

表 D. 4 组播密码套件

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	SMS4-GCM
00-14-72	2~255	保留
其他	0~255	保留

D. 4. 1. 15 单播密码信息元素 UIE

单播密码信息元素 UIE 固定格式见图 D. 26。



图 D. 26 单播密码信息元素 UIE 固定格式

其中:

——单播密码(USK)套件计数字段:表示鉴别访问控制器 AAC 或请求者 REQ 支持的或选择的单

- 播密码套件个数；
- 单播密码(USK)套件字段:表示接入设备鉴别访问控制器 AAC 或请求者 REQ 支持的或选择的单播密码套件, n 为单播密码套件计数字段的值。

D. 4. 1. 16 邻居交换设备信息 NIE

邻居交换设备信息 NIE 固定格式见图 D. 27。

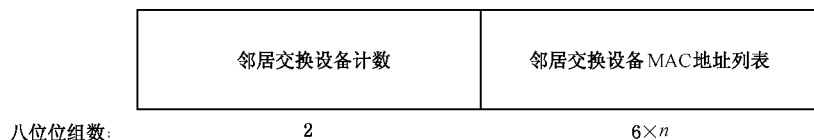


图 D. 27 邻居交换设备信息 NIE 固定格式

其中:

- 邻居交换设备计数字段:表示邻居交换设备的个数;
- 邻居交换设备 MAC 地址列表:表示用户终端拥有的或者选择邻居交换设备的信息列表,这里用 MAC 地址表示交换设备的信息值。

D. 4. 1. 17 邻居信息元素 NeighborItem

邻居信息元素 NeighborItem 固定格式见图 D. 28。

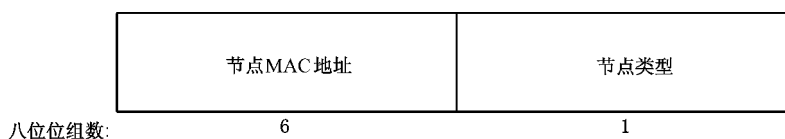


图 D. 28 邻居信息元素 NeighborItem 固定格式

其中:

- 节点 MAC 地址:表示邻居节点的信息,用其 MAC 地址进行标识;
- 节点类型:第一个比特位有效,0 表示为用户终端,1 表示为交换设备;其余比特位保留。

D. 4. 1. 18 公钥信息 PKI

公钥信息 PKI 固定格式见图 D. 29。



图 D. 29 公钥信息 PKI 固定格式

其中:

- 长度 PKI_Length:表示公钥算法标识、公钥算法参数和公钥值的长度。
- 公钥算法标识 PKI_AlglId:长度为 1 个八位位组。

值为 1 表示 ECDSA-256;

值为 2 表示 ECDSA-384;

值为 3 表示 ECDSA-192;

其他保留。

——公钥算法参数 $PKI_AlgPara$, 定义如 D. 4. 1. 6;

——公钥值 PKI_Val 字段; 定义见图 D. 30。

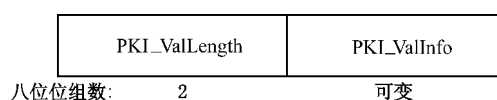


图 D. 30 公钥值 PKI_Val 字段定义

其中:

——公钥长度 $PKI_ValLength$ 字段: 为 2 个八位位组, 表示内容子字段的八位位组数;

——公钥内容 $PKI_ValInfo$ 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。

D. 4. 1. 19 接入结果 Acc_{RES}

接入结果 Acc_{RES} 固定格式见图 D. 31。

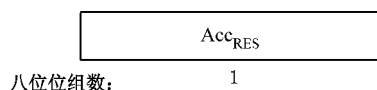


图 D. 31 接入结果 Acc_{RES} 固定格式

其中:

接入结果字段的长度为 1 个八位位组。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。

D. 4. 1. 20 基密钥标识/交换基密钥标识 $BKID/SWBKID$

基密钥标识/交换基密钥标识 $BKID/SWBKID$ 固定格式见图 D. 32。



图 D. 32 基密钥标识/交换基密钥标识 $BKID/SWBKID$ 固定格式

其中:

—— $BKID$: 基密钥标识, 长度为 16 个八位位组, 其计算方法为, $BKID = KD-HMAC-SHA256(BK, MAC_{AC} || MAC_{REQ})$ 。其中“||”为链接操作, 算法 $KD-HMAC-SHA256$ 具体见附录 A. 4。

—— $SWBKID$: 交换基密钥标识, 长度为 16 个八位位组, 其计算方法为, $SWBKID = KD-HMAC-SHA256(SWBK, MAC_{SW1} || MAC_{SW2})$ 。其中“||”为链接操作。

D. 4. 1. 21 单播会话密钥索引/交换密钥索引 USKID/SWKeyID

单播会话密钥索引/交换密钥索引 USKID/SWKeyID 固定格式见图 D. 33。

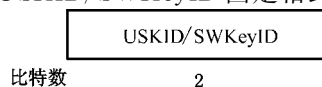


图 D. 33 单播会话密钥索引/交换密钥索引 USKID/SWKeyID 固定格式

其中：

USKID/SWKeyID:单播会话密钥索引/交换密钥索引,长度为 2 个比特位,其中比特 0 有意义。

D. 4. 1. 22 组播会话密钥索引/站间密钥索引 MSKID/STakeyID

组播会话密钥索引/站间密钥索引 MSKID/STakeyID 固定格式见图 D. 34。

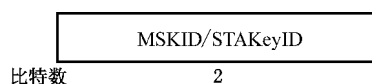


图 D. 34 组播会话密钥索引/站间密钥索引 MSKID/STakeyID 固定格式

其中：

MSKID/STakeyID:组播会话密钥索引/站间密钥索引,长度为 2 个比特位,其中比特 0 有意义。

D. 4. 2 数据元素封装格式

在 TLSec 协议中,定义一种数据元素封装格式见图 D. 35。

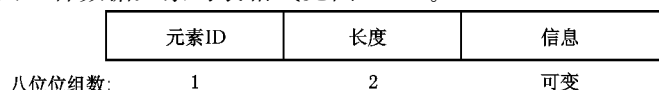


图 D. 35 数据元素封装格式

其中：

- 元素 ID 字段:表示元素在每一个分组中的元素身份标识。每个分组根据分组内包含的元素集合给每个元素分配一个分组内唯一的元素 ID。元素 ID 字段长为 1 个八位位组。
- 长度字段:表示元素信息字段的八位位组数。长度字段长为 2 个八位位组。
- 信息字段:表示元素的内容,对应于每个元素,其长度也不同,这将在每个分组中介绍元素时进行具体介绍。

D. 4. 3 TePA-AC 的 TAEPoL 协议

TLA 协议分组都使用 TePA-AC 标准中定义的 TAEPoL 协议进行封装, TAEPoL 协议在 GB/T 15629.3 网络中其协议帧结构及各组成部分相对位置见图 D. 36。

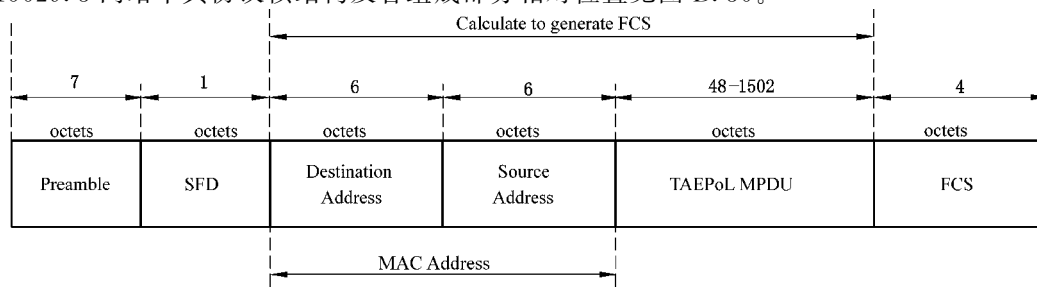


图 D. 36 TAEPoL 协议帧结构示意图

D.4.3.1 TAEPoL 协议各组成部分

D.4.3.1.1 前导码 Preamble

IEEE802.3 帧的前导码有 7 个字节(56 位)交替出现的 0 和 1, 交替序列如下:

预同步码=10

即 16 进制的 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

D.4.3.1.2 帧首定界符 SFD

IEEE802.3 帧的 SFD 字段占 1 个字节, 其比特模式为“10101011”, 它紧跟在前导码后, 用于指示一帧的开始。前导码的作用是使接收端能根据“1”、“0”交变的比特模式迅速实现比特同步, 当检测到连续两位“1”(即读到帧起始定界符字段 SFD 最末两位)时, 便将后续的信息递交给 MAC 子层。

D.4.3.1.3 MAC 地址

每个帧应包含两种地址字段: 依次为目的地址字段和源地址字段。每种地址字段应包含 48 比特, 地址字段见图 D.37。

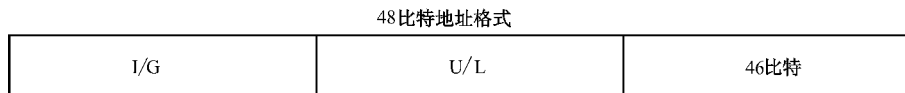


图 D.37 MAC 地址字段格式

D.4.3.1.3.1 目的 MAC 地址

目的地址字段规定该帧应发往的目的地。目的地址字段中的第 1 个比特位用作地址类型指示比特, 以标识目的地址是单地址还是组地址。如果该比特位为“0”, 表示目的地址为单地址; 如果该比特位为“1”, 表示目的地址字段中包含组地址, 它标识连接到局域网上的 0 个、1 个、多个或所有的站。第 2 个比特位用来区分本地管理地址和全球管理地址。对于全球管理地址, 该比特置为“0”。如果地址是要本地分配的, 则给比特置为“1”。注意, 对于广播地址, 该比特也置为“1”。

D.4.3.1.3.2 源 MAC 地址

源地址地段应标识启动发送该帧的站。在源地址字段中, 第 1 比特是被保留并置成“0”。第 2 比特位用法同目的地址字段。

D.4.3.1.4 完整性校验值 FCS

完整性校验值 FCS 字段, 字段长度为 4 个八位位组。该字段存放经 CRC 校验后产生的值。

D.4.3.2 TAEPoL MPDU 结构

TePA-AC 标准中 TAEPoL MPDU 在 GB/T 15629.3 中的格式见图 D.38。

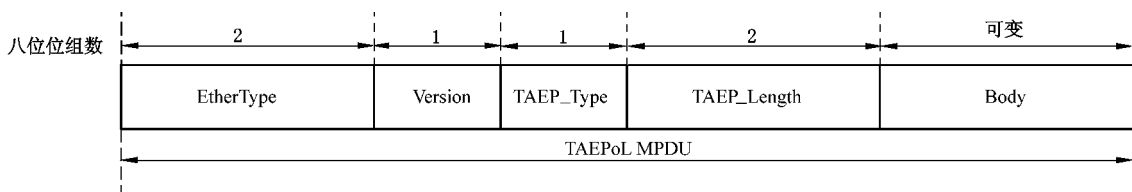


图 D.38 TAEPoL MPDU 结构示意图

其中以太网类型(EtherType)字段长度为 2 个八位位组,定义见表 D. 5。

表 D. 5 标准以太网类型分配

分配	值
EtherType	0x891b

D. 4. 3. 2. 1 TAEPoL PDU 结构

TAEPoL 协议数据单元(TAEPoL PDU)为 TAEPoL MPDU 中除 EtherType 字段以外的部分,其格式见图 D. 39。

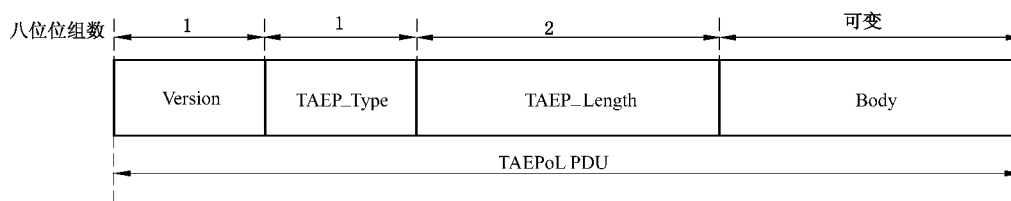


图 D. 39 TAEPoL PDU 的格式

D. 4. 3. 2. 2 协议版本 Version

Version 字段长度为 1 个八位位组,用一个无符号数表示。它的值表示 TAEPoL 帧的发送端所支持的 TAEPoL 协议版本。符合本附录的此字段值应为 0000 0001。

D. 4. 3. 2. 3 类型 TAEP_Type

TAEP_Type 字段长度为 1 个八位位组,用一个无符号数表示。它的值标识所发送的帧的类型,定义如下:

- a) TAEP-Packet: 值 0000 0000 表示帧载有 TAEP 分组;
- b) TAEPoL-Start: 值 0000 0001 表示帧为 TAEPoL-Start 帧;
- c) TAEPoL-Logoff: 值 0000 0010 表示帧为 TAEPoL-Logoff 请求帧;
- d) TAEPoL-Key: 值为 0000 0011 表示帧为 TAEPoL-Key 帧;
- e) TAEPoL-Encapsulated-ASF-Alert: 值为 0000 0100, 用于支持 AlertStandardForum (ASF) 的 Alerting 报文。

除上述 5 个值以外的所有其他值都是 TAEPoL 为未来扩展保留的。

D. 4. 3. 2. 4 长度 TAEP_Length

TAEP_Length 字段长度为 2 个八位位组,用一个无符号二进制数表示。该字段的值定义了内容字段的长度;值 0 表示没有内容字段。

D. 4. 3. 2. 5 内容 Body

如下:

- a) 在负载有 TAEP-Packet 的 TAEPoL 帧内,该字段包含如下定义的 TAEP 分组;仅仅封装一个 TAEP 分组。
- b) 在负载有 TAEPoL-Start 帧内,该字段包含如下所定义的 Hello;仅仅封装一个 Hello 或不封

装任何内容。

- c) 在负载有 TAEPoL-Logoff 帧内,该字段包含如下所定义的 Logoff;仅仅封装一个 Logoff 或不封装任何内容。
- d) 在负载有 TAEPoL-Key 帧内,该字段包含如下所定义的 Key Descriptor;仅仅封装一个 Key Descriptor。
- e) 在负载有 TAEPoL-Encapsulated-ASF-Alert 的帧内,此字段包含 ASF 描述的 ASF 警告。

D.4.3.3 TAEPoL 协议下的数据帧

根据 TAEPoL PDU 中类型字段的取值,构造出不同类型的数据帧。分别为以下:TAEP-
Packet 帧、TAEPoL-Key 帧、TAEPoL-Start 帧、TAEPoL-Logoff 帧等。其中 TLA 协议分组使用
TAEP-
Packet 帧、TAEPoL-Key 帧进行封装。

D.4.3.4 TAEP 分组

TAEP 分组用于在安全策略协商阶段、身份鉴别阶段及交换路径探寻阶段,完成请求者 REQ 与鉴
别访问控制器 AAC 之间的 TAEP 分组传递。TAEP 分组支持鉴别访问控制器和请求者之间、鉴别访
问控制器和鉴别服务器之间的通信功能。TAEP 分组的格式见图 D.40。



图 D.40 TAEP 分组格式示意图

其中:

——Code 字段:长度为 1 个八位位组,表示 TAEP 分组的类型,该字段编码分配方式如下:

- 1 Request 0000 0001;
- 2 Response 0000 0010;
- 3 Success 0000 0011;
- 4 Failure 0000 0100。

——Identifier 字段:长度为 1 个八位位组,用于匹配 Request 和 Response 分组。

——TB_Length 字段:长度为 2 个八位位组,表示整个 TAEP 分组的八位位组数,即指包括 Code、
Identifier、Length 和 Data 所有字段的长度总和。

——Data 字段:长度可变,分组包含 0 个或多个八位位组,其格式由 Code 字段的值决定。

D.4.3.4.1 Request 和 Response

当 Code 字段取值 Request 和 Response 时的 TAEP 分组格式定义见图 D.41。

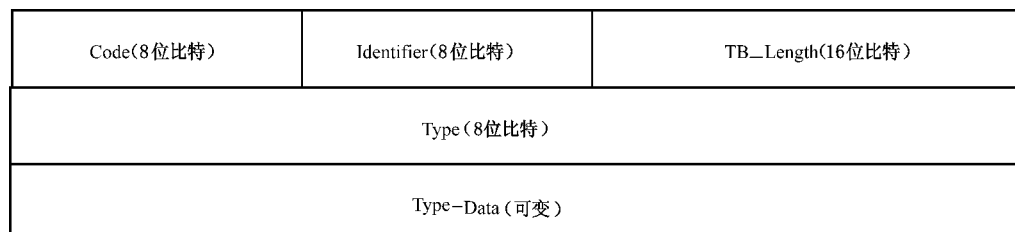


图 D.41 Request 和 Response 类型的 TAEP 分组格式示意图

其中：

Type 字段长度为 1 个八位位组，它表示 Request 和 Response 分组的类型，TePA-AC 中已定义部分类型，根据 TLSec 的应用需求，经申请增加如下的表 D. 6 类型。

表 D. 6 新增 Request 和 Reponse 分组类型定义

Type 值	定 义	描 述
243	Neighbor SW Negotiation	Neighbor SW Negotiation 类型用于邻居用户终端间协商选择一个共同的邻居交换设备，用于后期站间密钥的建立过程。见 D. 9. 8。
244	Neighbor information	Neighbor information 类型用于节点发现邻居节点过程。见 D. 5。
245	TAEP-CAAP	TAEP-CAAP 是一个局域网基于证书的鉴别协议。见 D. 7. 1. 3。
246	Policy Negotiation	Policy Negotiation 类型用于节点之间协商安全策略，当 AAC 需要将自己支持的安全策略信息告知 REQ 时，使用 Policy Negotiation 类型 Request 分组；当 REQ 收到 AAC 发送的 Policy Negotiation 类型 Request 分组后，使用 Policy Negotiation 类型 Response 分组进行响应，告知 AAC，自己所选择用于后续通信的安全策略信息。见 D. 6。
247	SW Routing Seek	SW Routing Seek 类型用于节点探寻交换路径信息。当一个发送源节点需要发送数据包到一个目的节点时，需要根据它们之间的交换路径信息判断此次数据通信的类型，进而选择适用的保密通信策略。节点之间的交换路径信息就需要使用 SW Routing Seek 类型的 Request 分组来发起探寻分组，目的节点收到后，就是用 SW Routing Seek 类型的 Response 分组进行响应发送源节点。

D. 4. 3. 4. 2 Failure 和 Success

当 Code 字段取值 Failure 和 Success 时的 TAEP 分组格式定义见图 D. 42。Failure 和 Success 分组不包含 Data 字段。

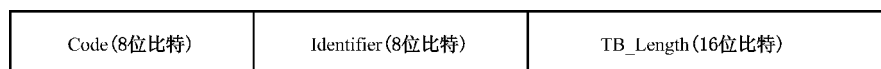


图 D. 42 Failure 和 Success 类型的 TAEP 分组格式示意图

D. 4. 3. 5 TAEPoL-Key 数据帧

D. 4. 3. 5. 1 Key Descriptor 格式



在完成安全策略协商阶段、身份鉴别阶段后，在密钥协商阶段时使用 TAEPoL-Key 帧在请求者 REQ 与鉴别访问控制器 AAC 之间传递消息。

每一个 TAEPoL-Key 帧仅仅封装一个 Key Descriptor，用来描述密钥及其相关信息。Key Descriptor 的格式见图 D. 43。

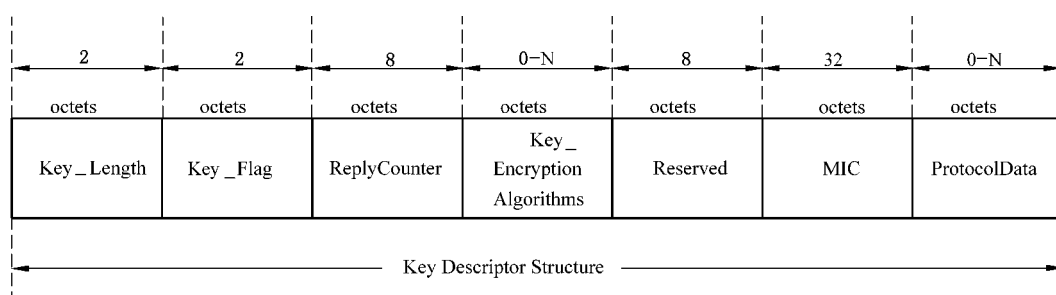


图 D.43 Key Descriptor 的格式示意图

D.4.3.5.2 长度字段 Key_Length

长度字段为 2 个八位位组长,是一个整数,表示包含长度字段在内的 Key Descriptor 中所有字段的八位位组数。

D.4.3.5.3 标识字段 Key_FLAG

标识字段为 2 个八位位组长,表示密钥的性质,其格式见图 D.44。

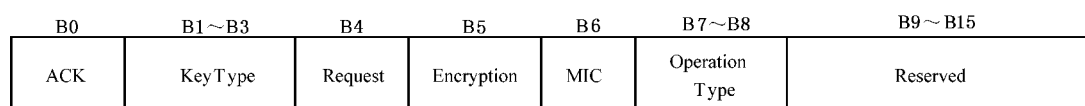


图 D.44 标识 Key_FLAG 字段的格式

其中:

- 应答标识 ACK 为 1 个比特,当鉴别访问控制器发送的 TAEPoL 帧要求被响应,则在帧中被置位,否则复位。请求者响应帧中的 ACK 位使用和鉴别访问控制器发送帧的一样。
- 密钥类型 KeyType 为 3 个比特,用于表示 TAEPoL 帧的密钥交换协议的类型,见表 D.7。

表 D.7 KeyType 类型

KeyType 取值	意义
000	单播密钥
001	组播密钥
010	站间密钥
011	交换基密钥
100	交换密钥
101~111	保留

- 请求标识 Request 为 1 个比特,当请求者要求鉴别访问控制器开始一个密钥交换协议时,在请求者发送的帧中被置位。若鉴别访问控制器应请求者的要求开始一个密钥交换协议,鉴别访问控制器发送的帧中该位也置位,表示是请求者要求开始密钥交换协议的。
- 加密标识 Encryption 为 1 个比特,如果密钥数据 KeyData 字段中的 Key 是密文,则该位置位,否则复位。
- 检验标识 MIC 为 1 个比特,如果 TAEPoL 帧中包含 MIC 值,该位置位,否则复位。
- 操作类型 OperationType 为 2 个比特,用于表示交互过程的类型,见表 D.8。

表 D.8 OperationType 类型

OperationType 取值	意 义
00	建立过程
01	更新过程
10	删除过程
11	保留

D.4.3.5.4 重放计数器 ReplyCounter

重放计数器 ReplyCounter 字段为 8 个八位位组长,是一个整数。当其共享密钥建立以后,重放计数器初始化为 0。请求者在响应一个 TAEPoL-Key 帧时,使用收到的帧中的重放计数器值作为重放计数器值。它是一个序列,协议用它来检查重放攻击。请求者在收到有效的 TAEPoL-Key 帧后,递增收到帧中的重放计数器值作为自己的重放计数器值。鉴别访问控制器在收到有效的 TAEPoL-Key 帧后,递增收到帧中的重放计数器值作为自己的重放计数器值。有效的帧是指帧的 MIC 值校验正确。

D.4.3.5.5 算法字段 Key_ EncryptionAlgorithms

算法 Key Encryption Algorithms 字段长度是可变的,该字段是一个 OID,采用 DER 编码。它表示 MIC 字段使用的算法,该算法的选择需符合国家密码政策的规定。

D.4.3.5.6 保留字段 Reserved

保留字段为八个八位位组。

D.4.3.5.7 MIC 字段

消息完整性校验 MIC 字段为 32 个八位位组,采用 HMAC-SHA256 算法计算。它是 TAEPoL-Key 帧的 MIC 值,包含帧的全部内容,在进行 MIC 计算时 MIC 字段屏蔽为 0。

D.4.3.5.8 协议数据字段 ProtocolData

协议数据字段是变长字段,包含用于密钥交换协议的附加数据,其封装格式见图 D.45。

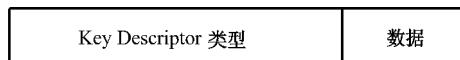


图 D.45 协议数据的格式

其中:

Key Descriptor 类型字段为 1 个八位位组,其中 0x00 保留;0x01~0xFF,用于分配给不同的协议。根据 TLSec 的应用需求,经申请增加如下的表 D.9 类型。

表 D.9 新增 Key Descriptor 类型定义

Type 值	定 义	描 述
0x10	TLA 单播密钥协商协议	参见 D. 7. 1. 4
0x11	TLA 基于预共享密钥的鉴别及单播密钥管理协议	参见 D. 7. 2
0x12	组播密钥通告	参见 D. 8
0x13	TLA 站间密钥建立	参见 D. 9
0x14	TLA 交换基密钥建立	参见 D. 10. 2
0x15	TLA 交换密钥协商	参见 D. 10. 3

D.5 邻居节点发现

D.5.1 邻居节点发现过程概述

新节点接入网络后,需要通过邻居节点发现过程来获取所有的邻居节点信息,同时也将自己的信息告知周围的邻居节点。邻居节点发现过程包含 2 个分组:邻居节点发现请求分组,邻居节点发现响应分组。其中邻居节点发现请求分组使用 Type=244, code=1 即 TAEP-Request-Neighbour Information/TAEPoL 进行封装;邻居节点发现响应分组使用 Type=244, code=2 即 TAEP-Response-Neighbour Information/TAEPoL 进行封装。

每个节点保存的邻居节点列表中,邻居节点信息元素固定格式见 D. 4. 1. 17。

邻居节点发现协议分组数据封装格式见图 D. 46。

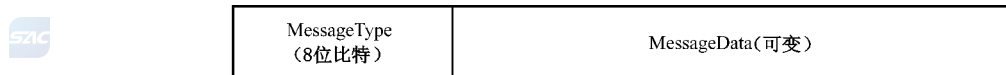


图 D.46 邻居节点发现协议分组数据封装格式

其中:

——MessageType,表示消息分组所属的分组类别,定义如下表 D. 10。

表 D.10 邻居节点发现协议分组 MessageType 子类型定义

MessageType 值	定 义
0x00	保留
0x01	邻居节点发现请求分组
0x02	邻居节点发现响应分组
其余	保留

——MessageData 数据:每种分组的消息数据 MessageData 将在协议分组介绍中进行介绍。其数据元素封装格式见 D. 4. 2。

D.5.2 邻居节点发现请求分组

新节点加入网络后,主动构造“邻居节点发现请求分组”分组,以广播形式进行发送。该分组的目的是 MAC 地址,使用广播 MAC 地址;源 MAC 地址是该新节点的 MAC 地址;该分组的主要字段为新节点的节点类型字段。

对于邻居节点发现请求分组,所有的节点收到后均不进行转发,只需提取发送节点的 MAC 信息及节点类型字段添加到自己的邻居节点列表中;之后,接收节点需构造邻居节点发现响应分组,发送给新节点。

邻居节点发现请求分组中,数据元素都采用 D.4.2 介绍的封装格式进行封装。分组中有效数据信息元素的集合及元素 ID 定义见表 D.11。

表 D.11 邻居节点发现请求分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
新节点的节点类型	0	1	D.4.1.17
保留	1~255	—	—

其中:

新节点 Node 的节点类型:表示新节点 Node 是交换设备还是用户终端。

D.5.3 邻居节点发现响应分组

节点若收到新节点发送的邻居节点发现请求分组,则说明该节点是新节点的邻居节点;该节点作为响应节点,将新节点信息添加到自己的邻居节点列表中后,还需构造邻居节点发现响应分组,告诉新节点自己的信息,以便于新节点收集邻居节点的信息。

邻居节点发现响应分组是单播消息,源 MAC 是响应节点的 MAC 地址,目的节点是新节点的 MAC 地址;该分组的主要字段为响应节点的节点类型字段。

邻居节点发现请求分组中,数据元素都采用 D.4.2 介绍的封装格式进行封装。分组中有效数据信息元素的集合及元素 ID 定义见表 D.12。

表 D.12 邻居节点发现响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
响应节点的节点类型	0	1	D.4.1.17
保留	1~255	—	—

其中:

响应节点的节点类型字段:表示响应节点是交换设备还是用户终端。

D.6 安全策略协商

D.6.1 安全策略协商过程概述

请求者 REQ 和鉴别访问控制器 AAC 通过安全策略协商过程协商所使用的鉴别和密钥管理方法,

包括鉴别及单播密钥管理方法,单播密码套件,组播密码套件等信息。安全策略协商过程包含 2 个分组:安全策略协商请求分组,以及安全策略协商响应分组。安全策略协商请求分组使用 Type=246, code=1 的 TAEP-Request-Policy Negotiation/TAEPoL 分组进行封装;安全策略协商响应分组使用 Type=246, code=2 的 TAEP-Response-Policy Negotiation/TAEPoL 分组进行封装。协商过程见图 D.47。

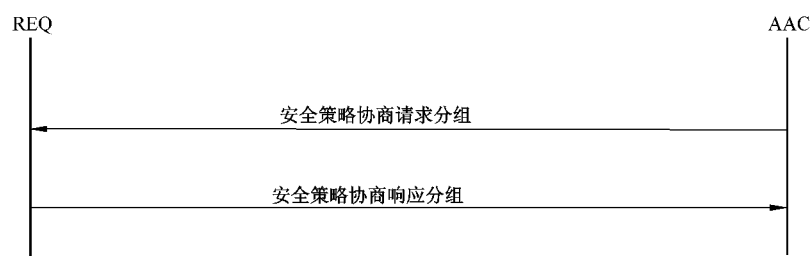


图 D.47 安全策略协商过程

安全策略协商协议分组数据封装格式见图 D.48。

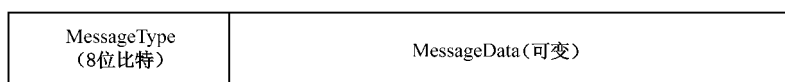


图 D.48 安全策略协商协议分组数据封装格式

其中:

——MessageType,表示消息分组所属的分组类别,定义如下表 D.13。

表 D.13 安全策略协商协议分组 MessageType 子类型定义

MessageType 值	定义
0x00	保留
0x01	安全策略协商请求分组
0x02	安全策略协商响应分组
其余	保留

——MessageData 数据:每种分组的消息数据 MessageData 将在协议分组介绍中进行介绍。其数据元素封装格式见 D.4.2。

D.6.2 安全策略协商请求分组

当用户设备试图接入网络或收到用户设备发出的 TAEPoL-Start 分组时,网络接入设备向客户端发送安全策略协商请求分组,该分组包含 TIE_{AAC} 字段,描述 AAC 支持的所有鉴别和密钥管理套件及密码套件供 REQ 进行选择,并封装在 TAEP-Request/Policy Negotiation 中。

安全策略协商请求分组中,数据元素都采用 D.4.2 介绍的封装格式进行封装。分组中有效数据信息元素的集合及元素 ID 定义见表 D.14。

表 D. 14 安全策略协商请求分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定义
AAC 支持的 TLA 信息元素 TIE _{AAC}	0	可变	D. 4. 1. 14
保留	1~255	—	—

其中：

AAC 支持的 TLA 信息元素 TIE_{AAC}：表示 AAC 所支持的 TLA 信息元素，即 AAC 所支持的鉴别和密钥管理套件以及密码套件等信息。

D. 6. 3 安全策略协商响应分组

当请求者 REQ 收到鉴别访问控制器 AAC 发来的安全策略协商请求分组时，根据安全策略协商请求分组中的 TIE_{AAC} 元素选择一种双方共有的鉴别和密钥管理套件及密码套件，或者 REQ 结合本地策略，选择一种鉴别和密钥管理套件及密码套件，组成安全策略协商响应分组发送给鉴别访问控制器 AAC，该分组封装在 TAEP-Response/Policy Negotiation 中。若 REQ 对安全策略协商请求分组中罗列的鉴别和密钥管理套件或密码套件都不支持，根据本地策略可丢弃该分组。

安全策略协商响应分组中，数据元素都采用 D. 4. 2 介绍的封装格式进行封装。分组中有效数据信息元素的集合及元素 ID 定义见表 D. 15。

表 D. 15 安全策略协商响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定义
REQ 选择的 TLA 信息元素 TIE _{REQ}	0	可变	D. 4. 1. 14
保留	1~255	—	—

其中：

REQ 支持的 TLA 信息元素 TIE_{REQ} 元素：表示 REQ 选择的 TLA 信息元素，即 REQ 选择的鉴别和密钥管理套件以及密码套件等信息；该字段中的鉴别和密钥管理 (AKM) 套件计数以及单播密码套件计数均为 1。

当 AAC 收到 REQ 回应的安全策略协商响应分组时，判断其中的 TIE_{REQ} 字段是否有效，即 AAC 是否支持 REQ 选择的鉴别和密钥管理套件及密码套件，若不支持，则丢弃该分组；否则，根据 REQ 选择的鉴别和密钥管理套件开始相应的核心鉴别过程。

鉴别过程可实现用户与网络之间的双向鉴别，也可仅完成网络对用户的单向鉴别。鉴别过程既可基于公钥证书，也可基于预共享密钥。公钥证书建议采用 X. 509. v3 证书。

D. 7 鉴别和单播密钥管理

D. 7. 1 基于证书的鉴别和单播密钥管理

D. 7. 1. 1 过程概述

基于证书的鉴别和密钥管理过程，主要包括身份鉴别过程，单播密钥协商过程，组播/站间密钥通告过程及交换密钥建立过程，见图 D. 49。

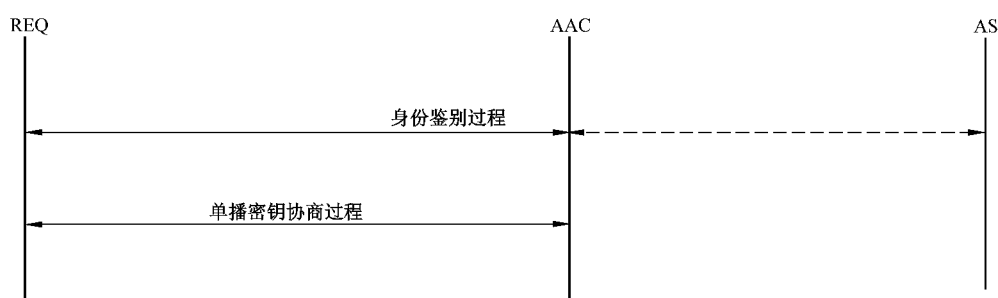


图 D.49 基于证书的鉴别和密钥管理过程

身份鉴别过程是基于三元实体利用公钥证书完成的。根据鉴别的安全等级不同,可分为单向鉴别和双向鉴别。单向鉴别过程仅完成网络对用户的鉴别;双向鉴别完成网络与用户之间的双向鉴别,既可以验证用户的合法性,也可以为用户验证接入网络的合法性。

D.7.1.2 公钥证书

D.7.1.2.1 公钥证书格式

凭借证书和私钥可以唯一地确定网络设备的身份,公钥证书是网络设备在网络环境中的数字身份凭证。通过与密码技术及安全协议相结合,确保公钥证书的唯一性、不可伪造性。

公钥证书格式定义见表 D.16。

表 D.16 公钥证书的格式

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

其中:

- 公钥证书的版本号: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 AS 颁发的公钥证书都需要分配一个唯一的序列号,由证书的序列号和证书颁发者的名称可以唯一确定证书持有者;
- 证书颁发者采用的签名算法:该字段指定了证书颁发者所采用的签名算法,包括签名算法名称、签名长度与签名者所使用的公钥长度;本部分采用国家密码管理委员会办公室批准的用于

- LAN 的椭圆曲线密码(ECC)体制实现签名算法；
- 证书颁发者名称:该字段指定证书颁发者的身份；
 - 证书颁发者的公钥信息:该字段为证书颁发者的公钥信息；
 - 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；
 - 证书类型:该字段表示证书持有者的设备类型,即 REQ、AAC 或 AS；
 - 扩展:该字段保留,用于以后的扩展应用；
 - 证书颁发者对证书的签名:该字段由证书颁发者(AS)对该证书上的所有字段项进行签名得到。

D.7.1.2.2 公钥证书管理

D.7.1.2.2.1 证书颁发

申请证书时,先在 AS 处登记,AS 对证书申请实体的身份进行确认后,按照申请者所需的安全等级为其制作和颁发证书。证书产生步骤如下:

- a) AS 产生实体的非对称密钥对；
- b) 检查公钥信息；
- c) 接受公钥信息；
- d) 添加公钥证书管理所需的数据；
- e) 计算公钥证书的签名；
- f) 审计记录登记,记录 AS 在公钥证书产生过程中的行为。

证书颁发可以采用拖拉模式,由一个证书本(数据库)记录所有用户的证书,用户需要时通过数据库提取出所需证书;另一种采用推进模式,证书生成后送给所有的用户或定期向用户发放。

D.7.1.2.2.2 证书吊销

AS 可以在证书到期之前吊销证书。具体的原因包括:

- a) 实体私钥的损坏或丢失；
- b) 实体请求吊销；
- c) 实体隶属关系的改变；
- d) 实体的终止；
- e) 实体的错误识别；
- f) AS 私钥的损坏；
- g) AS 的终止。

因此,应有一定的程序和快速的通信方法以便能安全而可鉴别地吊销,吊销应提供三种方式:

- 一个或多个实体的一个或多个证书的吊销；
- 由 AS 颁发的基于单个非对称密钥对的一系列公钥证书的吊销；
- 由 AS 颁发的所有公钥证书的吊销。

在已知或怀疑 AS 的私钥泄露时,或在用于签发证书的非对称密钥对被更换时,后两项要求为吊销公钥证书提供了手段。无论公钥证书是过期还是被吊销,旧的公钥证书的拷贝应由可信任的第三方保留一段时间。

当实体或 AS 的私钥因为某种原因而被取消时,颁发该公钥证书的 AS 应立即主动通知系统中的

所有实体,所有有关的公钥证书都被吊销。可采取的形式有:由 AS 鉴别并发送给所有实体的消息、由另一 AS 鉴别的消息、由可信任的第三方保存的一个在线的已吊销公钥证书列表以及公开已吊销或有效的公钥证书列表。

当一个公钥证书因被怀疑或已知某一私钥损坏而被吊销时,该私钥不能继续使用。如果数据已在吊销前签名,公钥证书应只用于验证,而且任何由该公钥证书加密的密钥材料(无论何种类型)都应在操作方便时更换。

吊销列表包括一个带时间戳的顺序表或公钥证书标识符表,以表示由 AS 吊销的公钥证书。在吊销列表中使用两种时间标记:一种是 AS 颁布的吊销日期和时间,另一种是已知或怀疑泄露的日期和时间。

如果知道泄露的日期和时间,就更加容易审计可疑消息。公钥证书在吊销列表上至少应保持到截止期为止。

一旦由于已知或怀疑泄露而执行吊销,如果签名是在怀疑密钥泄露之后进行的或签名日期无法确定,应认为使用有关私钥签发的信息不再有效。不能使用已吊销的公钥加密消息。

吊销列表应该由 AS 注明吊销日期并进行签名,以使实体能确认该表的完整性,并确定颁发日期;吊销列表由 AS 定期发布,即使自上次发布之日起无任何变化。系统的所有实体都可以获得吊销列表,除非由法律、法规所排除在外的。

以下分发机制可用于吊销列表,包括:

- 由可信任的第三方作为消息/报告发送给每个用户;
- 由用户请求可信任的第三方提供指定公钥证书的当前情况;
- 向 AS 索取当前的吊销列表。

AS 应定期生成并公布新的吊销列表。

D. 7. 1. 3 基于证书的鉴别协议 TAEP-CAAP

D. 7. 1. 3. 1 基于证书的鉴别过程概述

基于证书的鉴别过程包含 6 个分组,其中接入鉴别确认分组是可选分组,是否需要接入鉴别确认分组会根据鉴别激活分组的字段情况决定的。基于证书的鉴别过程见图 D. 50。

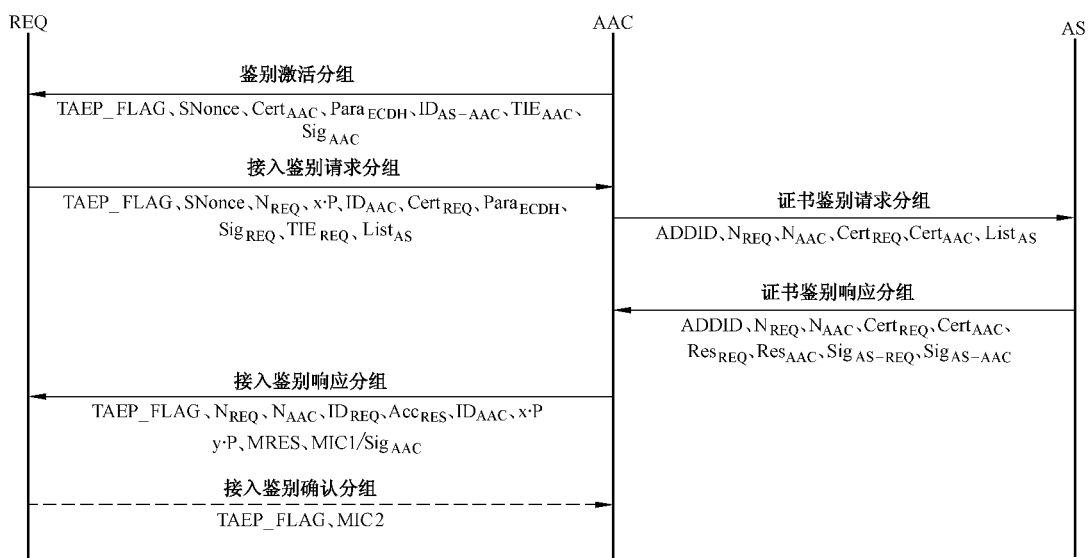


图 D. 50 TAEP-CAAP 过程

基于证书的鉴别协议分组数据封装格式见图 D. 51。

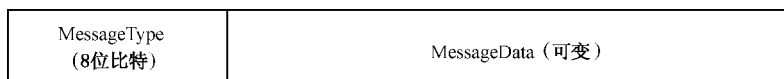


图 D. 51 TAEP-CAAP 类型分组数据封装格式

其中：

——MessageType, 表示消息分组所属的分组类别, 定义如下表 D. 17。

表 D. 17 TAEP-CAAP 类型分组 MessageType 子类型定义

MessageType 值	定义
0x00	保留
0x01	鉴别激活分组
0x02	接入鉴别请求分组
0x03	证书鉴别请求分组
0x04	证书鉴别响应分组
0x05	接入鉴别响应分组
0x06	接入鉴别确认分组
其余	保留

——MessageData 数据: 每种分组的消息数据 MessageData 将在协议分组介绍中进行介绍。其数据元素封装格式见 D. 4. 2。

D. 7. 1. 3. 2 鉴别激活分组

安全策略协商过程成功完成后, 如果 AAC 与 REQ 协商选择采用基于证书的鉴别和密钥管理策略, 则 AAC 向 REQ 发送鉴别激活分组。主要包括: 标识 TAEP_FLAG、一次性随机数 SNonce(标识鉴别的新鲜性, 亦称鉴别标识)、本地信任的 AS 身份 ID_{AS-AAC}、AAC 的证书 Cert_{AAC}、ECDH 参数 Para_{ECDH}、TLA 信息元素 TIE_{AAC} 及 AAC 的签名 Sig_{AAC}。为了方便 REQ 选择鉴别证书, 还可包含 AAC 所信任的 AS 信息。该分组封装在 Code = 1 (Request), Type = 245 (TAEP-CAAP), MessageType = 0x01 的 TAEP-Request-CAAP/TAEPoL 中。

鉴别激活分组中, 数据元素 SubData 都采用 D. 4. 2 介绍的封装格式进行封装。分组中有效元素的集合及元素 ID 定义见表 D. 18。

表 D. 18 鉴别激活分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定义
标识 TAEP_FLAG 元素	0	1	D. 4. 1. 1
鉴别标识 SNonce 元素	1	32	D. 4. 1. 2
本地信任的 AS 身份 ID _{AS-AAC} 元素	2	可变	D. 4. 1. 4
AAC 的证书 Cert _{AAC} 元素	3	可变	D. 4. 1. 5
ECDH 参数 Para _{ECDH} 元素	4	可变	D. 4. 1. 6

表 D. 18 (续)

信息元素	元素 ID	元素长度 (八位位组)	定义
TLA 信息元素 TIE _{AAC} 元素	5	可变	D. 4. 1. 14
AAC 的签名 Sig _{AAC} 元素	6	可变	D. 4. 1. 7
保留	7~15		
保留	16~255		

其中：

- 标识 TAEP_FLAG:标识 TAEP_FLAG 的比特 0、2 有意义。当 REQ 实体与 AAC 实体完成安全策略协商后进行证书鉴别过程,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。
- 鉴别标识 SNonce:如果分组中标识 TAEP_FLAG 元素的比特 0 值为 1,即如果是 BK 更新过程,则此分组中鉴别标识 SNonce 为上一次证书鉴别过程中保存的鉴别标识;否则分组中的鉴别标识 SNonce 为鉴别访问控制器 AAC 生成的随机数。
- 本地 AAC 信任的 AS 身份 ID_{AS-AAC}:表示 AAC 信任的 AS 身份 ID。
- AAC 的证书 Cert_{AAC}:表示 AAC 的实体证书。
- ECDH 参数 Para_{ECDH}:表示 ECDH 参数内容。
- TLA 信息元素 TIE_{AAC}: TIE_{AAC} 表示 AAC 所支持的 TLA 信息元素,也就是 AAC 所支持的鉴别与密钥管理套件及密码套件,此字段的值应与安全策略协商过程中 AAC 所发出的安全策略请求分组中的 TIE_{AAC} 字段值相同。
- AAC 的签名 Sig_{AAC}:表示 AAC 利用自己的常用私有密钥对鉴别激活分组除本元素外的其他所有元素进行的签名。该字段为可选字段。

当 REQ 实体与 AAC 实体进行通信时,REQ 和 AAC 选择采用证书鉴别和密钥管理方法,或 AAC 的本地策略要求重新进行证书鉴别过程,或 AAC 收到 REQ 实体的预鉴别开始分组时,AAC 向 REQ 实体发送鉴别激活分组,激活 REQ 实体进行证书鉴别。

REQ 实体接收到由 AAC 发送的鉴别激活分组后,进行如下处理:

- a) 检查鉴别激活分组中 TIE_{AAC} 字段值与之前收到的安全策略协商请求分组中的 TIE_{AAC} 字段值是否相同,如果不同,则丢弃该分组,否则执行 b);
- b) 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 c)操作;当值为 0 时执行 d)操作;
- c) REQ 实体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 d)操作;
- d) 如果收到的鉴别激活分组中包含 SIG_{AAC} 字段,则需要验证 SIG_{AAC} 字段的正确性,如果不正确则丢弃该分组,否则执行 e);如果分组中不包含 SIG_{AAC} 字段,则直接执行 e);
- e) REQ 实体根据鉴别激活分组中的 AAC 信任的鉴别服务器 AS 的身份标识 ID_{AS} 字段选择由该鉴别服务器 AS 实体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 $x \cdot P$ 和 32 个八位位组的 REQ 询问,即随机数 N_{REQ} ,生成接入鉴别请求分组,发送给 AAC。

D. 7. 1. 3. 3 接入鉴别请求分组

接入鉴别请求分组主要包括:标识 TAEP_FLAG、鉴别激活分组中的一次性随机数 SNonce、本地

生成的一次性随机数 N_{REQ} 、本地生成的临时公钥 $x \cdot P$ 、AAC 的身份信息 ID_{AAC} (可根据鉴别激活分组中的 AAC 公钥证书 $Cert_{AAC}$ 得到)、REQ 选择的鉴别证书 $Cert_{REQ}$ 、 $Para_{ECDH}$ 、REQ 信任的 AS 列表 $List_{AS}$ 、TLA 信息元素 TIE_{REQ} 及签名信息 Sig_{REQ} (利用 REQ 的长期私钥计算得到)。若 REQ 所信任的 AS 不止一个,该分组还可包含 REQ 所信任的 AS 列表,方便 AS 系统的鉴别。

接入鉴别请求分组封装在 $Code = 1$ (Request), $Type = 245$ (TAEP-CAAP), $MessageType = 0x02$ 的 TAEP-Request-CAAP/TAEPoL 中。

接入鉴别请求分组中,数据元素 SubData 都采用 D. 4. 2 介绍的封装格式进行封装。分组中有效元素集合见表 D. 19。

表 D. 19 接入鉴别请求分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定义
标识 TAEP_FLAG 元素	0	1	D. 4. 1. 1
鉴别标识 SNonce 元素	1	32	D. 4. 1. 2
本地询问 N_{REQ} 元素	2	32	D. 4. 1. 3
临时公钥 $x \cdot P$ 元素	3	可变	D. 4. 1. 15
AAC 的身份 ID_{AAC} 元素	4	可变	D. 4. 1. 4
鉴别证书 $Cert_{REQ}$ 元素	5	可变	D. 4. 1. 5
$Para_{ECDH}$ 元素	6	可变	D. 4. 1. 6
REQ 信任 AS 列表 $List_{AS}$ 元素	7	可变	D. 4. 1. 13
TLA 信息元素 TIE_{REQ} 元素	8	可变	D. 4. 1. 14
签名 Sig_{REQ} 元素	9	可变	D. 4. 1. 7
保留	10~15		
保留	16~255		

其中:

- 标识 TAEP_FLAG 元素: 比特 0、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 AAC 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 REQ 要求验证 AAC 证书的有效性, 为 0 表示不需要验证 AAC 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 AAC 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 SNonce: 如果存在鉴别激活分组, 则接入鉴别激活分组中的鉴别标识 SNonce 字段值应与接入鉴别激活分组中的鉴别标识 SNonce 字段值相同, 如果没有鉴别激活分组, 也即是由 REQ 发起的 BK 更新的鉴别过程, 那么接入鉴别请求分组中的鉴别标识 SNonce 值应与上一次证书鉴别过程中生成的鉴别标识相同。
- 本地询问 N_{REQ} 元素: 由 REQ 采用随机数生成算法生成。
- 临时公钥 $x \cdot P$ 元素: 内容是 REQ 生成的用于 ECDH 交换的临时公钥。
- AAC 的身份 ID_{AAC} 元素: 根据鉴别激活分组中的 AAC 公钥证书 $Cert_{AAC}$ 得到。
- REQ 的证书 $Cert_{REQ}$ 元素: REQ 的公钥证书。
- ECDH 参数元素: 和鉴别激活分组中的 $Para_{ECDH}$ 字段相同; 当 REQ 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。

- REQ 信任 AS 列表 $List_{AS_REQ}$ 元素:该字段为可选字段,采用身份列表属性表示。身份列表的定义如 D.4.1.13。内容包含 REQ 实体信任的服务器,但不包含 REQ 的证书颁发者。若 REQ 实体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。
- TLA 信息元素 TIE_{REQ} 元素:表示 REQ 所选择的 TLA 信息元素,也就是 REQ 所选择的鉴别与密钥管理套件及密码套件信息。该字段应与其在安全策略协商过程中发送的安全策略协商响应分组中的 TIE_{REQ} 字段值相同。
- REQ 的签名 Sig_{REQ} 元素:采用签名属性表示,是对本分组中除本字段之外所有数据字段的签名。

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

- a) 如果鉴别访问控制器 AAC 发送了鉴别激活分组,则检查收到的分组中的 $SNonce$ 、 $Para_{ECDH}$ 字段值和鉴别激活分组中对应的字段值是否一致,如果有一个不一致,则丢弃该分组,否则执行 b);如果鉴别访问控制器 AAC 没有发送鉴别激活分组,则检查 $SNonce$ 字段值和上一次证书鉴别过程中计算的鉴别标识是否一致,并检查 $Para_{ECDH}$ 字段和上一次鉴别激活分组中的 $Para_{ECDH}$ 是否一致,如果有一个不一致,则丢弃该分组;否则执行 b)。
- b) 检查 ID_{AAC} 与自己的身份是否一致,并检查 TIE_{REQ} 字段的值与安全策略协商过程中收到的安全策略协商响应分组中的 TIE_{REQ} 字段值是否一致,如果有一个不一致,则丢弃该分组;否则执行 c)。
- c) 验证 REQ 实体的签名,若验证不通过,则丢弃该分组;否则执行 d)。
- d) 若标识字段的比特 2 为 1 或 AAC 的本地策略要求使用 AS 鉴别 REQ 实体的证书,则 AAC 生成证书鉴别请求分组,发往 AS;否则执行 e)操作。
- e) AAC 本地鉴别 REQ 实体的证书,即根据本地缓存的 REQ 实体证书的验证结果及其根据本地策略所定义的时效性确定 REQ 实体证书的验证结果。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 $x \cdot P$ 和 32 个八位位组的 AAC 询问 N_{AAC} (随机数),使用自己的临时私钥 y 和接入鉴别请求分组中 REQ 实体的临时公钥 $x \cdot P$ 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, $N_{AAC} || N_{REQ} ||$ “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 REQ 实体。若 REQ 实体证书鉴别结果不成功,AAC 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。AAC 的询问 N_{AAC} 和 AAC 的密钥数据(AAC 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 REQ 实体,然后解除与 REQ 实体的链路验证。

D.7.1.3.4 证书鉴别请求分组

AAC 收到接入鉴别请求分组后,验证 $TAEP_FLAG$ 、 $SNonce$ 、 ID_{AAC} 、 TIE_{REQ} 、ECDH 参数信息、签名信息 Sig_{REQ} ,验证通过后,如果需要请求 AS 进行证书验证,则构造证书鉴别请求分组并发往 AS,否则直接丢弃接入鉴别请求分组。

证书鉴别请求分组主要包括:地址索引 ADDID(标识鉴别所服务的主体)、接入鉴别请求分组中的 N_{REQ} 、本地生成的一次性随机数 N_{AAC} 、 $Cert_{REQ}$ 和 $Cert_{AAC}$ 。若接入鉴别请求分组中包含 REQ 信任的 AS 列表,则证书鉴别请求分组中也应包含该信息。

证书鉴别请求分组封装在 Code=1(Request), Type=245(TAEP-CAAP), Message Type = 0x03 的 TAEP-Request-CAAP/TAEP-AS-SVC 中。

证书鉴别请求分组中,数据元素 SubData 都采用 D.4.2 介绍的封装格式进行封装。证书鉴别请求分组有效元素集合见表 D.20。

表 D.20 证书鉴别请求分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定义
ADDID 元素	0	12	D.4.1.9
AAC 询问 N_{AAC} 元素	1	32	D.4.1.3
REQ 询问 N_{REQ} 元素	2	32	D.4.1.3
REQ 的证书 $Cert_{REQ}$ 元素	3	可变	D.4.1.5
AAC 的证书 $Cert_{AAC}$ 元素	4	可变	D.4.1.5
REQ 信任 AS 列表 $List_{AS-REQ}$ 元素	5	可变	D.4.1.13
保留	6		
保留	7~15		
保留	16~255		

其中:

- ADDID 元素:表示 $MAC_{AAC} || MAC_{REQ}$;
- AAC 询问 N_{AAC} 元素:由 AAC 采用随机数生成算法生成;
- REQ 询问 N_{REQ} 元素:该字段应与收到的接入鉴别请求分组中的 N_{REQ} 字段值相同;
- REQ 的证书 $Cert_{REQ}$ 元素:该字段和接入鉴别请求分组中 REQ 的证书字段相同;
- AAC 的证书 $Cert_{AAC}$ 元素:内容包含 AAC 的证书;
- REQ 信任 AS 列表 $List_{AS-REQ}$ 元素:本字段值应与 REQ 实体发送的接入鉴别请求分组中的 REQ 信任的 AS 服务器列表字段相同。

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

AAC 接收到 REQ 实体发送的接入鉴别请求分组并向 AS 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 REQ 实体发送的接入鉴别请求进行处理。

AS 收到证书鉴别请求分组后,进行如下处理:

- a) 如果此次鉴别过程为单向鉴别,则只需验证请求者 REQ 的证书 $Cert_{REQ}$,如果是双向鉴别则需要同时验证鉴别访问控制器 AAC 的证书 $Cert_{AAC}$ 和请求者 REQ 的证书 $Cert_{REQ}$,参照 RFC3280 进行的证书的验证,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,否则验证证书的状态,然后执行 b);
- b) 根据证书的验证结果,构造证书鉴别响应分组,并且附加相应的签名,发往鉴别访问控制器 AAC。

D.7.1.3.5 证书鉴别响应分组

AS 收到证书鉴别请求分组后,验证 $Cert_{REQ}$ 和 $Cert_{AAC}$ 证书的有效性,构造证书鉴别响应分组返回


给 AAC。

证书鉴别响应分组主要包括：地址索引 ADDID、证书验证结果及 REQ 信任的 AS 对证书验证结果的签名 Sig_{AS-REQ} 、AAC 所信任的 AS 的签名 Sig_{AS-AAC} 。其中证书验证结果包含 N_{REQ} 、 N_{AAC} 、 $Cert_{REQ}$ 、 $Cert_{AAC}$ 、REQ 证书的验证结果 Res_{REQ} 与 AAC 证书的验证结果 Res_{AAC} 。若 AAC 和 REQ 信任的 AS 相同，则证书鉴别相应分组中的签名只存在一个。

证书鉴别响应分组封装在 $Code=2(Response)$ ， $Type=245(TAEP-CAAP)$ ， $MessageType = 0x04$ 的 TAEP-Response-CAAP/TAEP-AS-SVC 中。

证书鉴别响应分组中，数据元素 SubData 都采用 D. 4. 2 介绍的封装格式进行封装。证书鉴别响应分组有效元素集合见表 D. 21。

表 D. 21 证书鉴别响应分组有效元素集合

 信息元素	元素 ID	元素长度 (八位位组)	定义
ADDID 元素	0	12	D. 4. 1. 9
证书的验证结果元素	1	可变	D. 4. 1. 11
REQ 信任的服务器签名 Sig_{AS-REQ} 元素	2	可变	D. 4. 1. 7
AAC 信任的服务器签名 Sig_{AS-AAC} 元素	3	可变	D. 4. 1. 7
保留	4~15		
保留	16~255		

其中：

- ADDID 元素：表示 $MAC_{AAC} || MAC_{REQ}$ ，该字段值和证书鉴别请求分组中的 ADDID 字段的值相同。
- 证书的验证结果元素：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 AAC 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 REQ 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 REQ 证书，第二个证书及结果对应于证书鉴别请求分组中的 AAC 证书。如果此次鉴别为单向鉴别过程，则证书的验证结果字段中不包含一次性随机数 2，验证结果 2，证书 2。也即不包含请求者 REQ 的询问 N_{REQ} 、鉴别访问控制器 AAC 的验证结果，以及鉴别访问控制器 AAC 的证书。
- REQ 信任的服务器签名 Sig_{AS-REQ} 元素：对本分组中证书的验证结果字段的签名，本字段为可选字段，如果 REQ 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此字段。
- AAC 信任的服务器签名 Sig_{AS-AAC} 元素：对本分组中除本字段和 ADDID 字段之外所有数据字段的签名。本字段为可选字段，如果对证书验证结果进行签名的鉴别服务器 AS 和 AAC 信任的鉴别服务器 AS 相同，则不包含此字段。如果 AAC 证书的验证结果为颁发者不明确，则证书鉴别响应分组中也不包含此字段。

AS 收到证书鉴别请求分组后，向 AAC 发送证书鉴别响应分组。AAC 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 AAC 的询问是否相同，若相同，则执行 b) 操作；否则，丢弃该分组。
- b) 如果分组中含有两个签名字段，则检查 Sig_{AS-AAC} 字段是否正确，若不正确，则丢弃该分组，否则

执行 c)；如果分组中只含有一个签名字段，即表明对证书验证结果进行签名的鉴别服务器 AS 也是鉴别访问控制器 AAC 所信任的鉴别服务器 AS，则检查 SIG_{AS-REQ} 字段是否正确，若不正确，则丢弃该分组，否则执行 c)。

- c) 检查证书验证结果中，REQ 的证书验证结果是否合法，如果为合法，执行 d)；如果不合法则执行 e)。
- d) 如果本地生成用于 ECDH 交换的临时私钥 y 和临时公 $y \cdot P$ ，使用自己的临时私钥 y 和 REQ 的临时公钥 $x \cdot P$ 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 ($(x \cdot y \cdot P)_{abscissa}, N_{AAC} || N_{REQ} ||$ “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 REQ 实体。若接入鉴别请求分组中 REQ 要求验证 AAC 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。
- e) AAC 设定接入结果为不成功，AAC 的询问 N_{AAC} 和 AAC 的密钥数据(AAC 的临时公钥)可设置任意值。构造接入鉴别响应分组发送给 REQ，然后解除与 REQ 实体的链路验证。若接入鉴别请求分组中 REQ 实体要求验证 AAC 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。

D.7.1.3.6 接入鉴别响应分组

AAC 收到证书鉴别响应分组后，验证 N_{REQ} 、 N_{AAC} 、 Sig_{AS-AAC} 等，若无效，则直接丢弃，否则构造接入鉴别响应分组返回给 REQ。

接入鉴别响应分组主要包括：标识 TAEP_FLAG、 N_{REQ} 、 N_{AAC} 、 ID_{REQ} (可根据 $Cert_{REQ}$ 得到)、接入结果、 ID_{AAC} 、REQ 密钥数据 $x \cdot P$ 、本地生成的临时公钥 $y \cdot P$ 、复合的证书验证结果 RES 及 AAC 生成的消息鉴别码 MIC1。如果在之前的鉴别激活分组中没有 AAC 的签名 Sig_{AAC} 字段或者此次鉴别过程没有鉴别激活分组，则分组中不需要计算消息鉴别码 MIC1，而是需要计算签名 Sig_{AAC} 。其中复合的证书验证结果 RES 为证书鉴别响应分组中除 ADDID 之外的其他内容。

接入鉴别响应分组封装在 Code=2(Response), Type=245(TAEP-CAAP), MessageType = 0x05 的 TAEP-Response-CAAP/TAEPoL 中。

接入鉴别响应分组中，数据元素 SubData 都采用 D.4.2 介绍的封装格式进行封装。接入鉴别响应分组有效元素集合见表 D.22。

表 D.22 接入鉴别响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定义
标识 TAEP_FLAG 元素	0	1	D.4.1.1
REQ 询问 N_{REQ} 元素	1	32	D.4.1.3
AAC 询问 N_{AAC} 元素	2	32	D.4.1.3
接入结果 Acc_{RES} 元素	3	1	D.4.1.19
REQ 密钥数据 $x \cdot P$ 元素	4	可变	D.4.1.15
AAC 密钥数据 $y \cdot P$ 元素	5	可变	D.4.1.15

表 D.22 (续)

信息元素	元素 ID	元素长度 (八位位组)	定义
AAC 的身份 ID _{AAC} 元素	6	可变	D. 4. 1. 4
REQ 身份 ID _{REQ} 元素	7	可变	D. 4. 1. 4
复合的证书验证结果 MRES 元素	8	可变	D. 4. 1. 14
消息鉴别码 MIC1 元素	9	20	D. 4. 1. 10
AAC 的签名 Sig _{AAC} 字段	10	可变	D. 4. 1. 7
保留	11~15		
保留	16~255		

其中:

- 标识 TAEP_FLAG 元素: 比特 0、1、3 有意义。本字段比特 0、比特 1 应与 REQ 实体发送的接入鉴别请求分组中标识字段值相同。比特 3(可选字段标识)由 REQ 实体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。
- REQ 询问 N_{REQ} 元素: 本字段值应与 REQ 实体发送的接入鉴别请求分组中 REQ 实体的询问字段值相同。
- AAC 询问 N_{AAC} 元素: 本字段值应与 AAC 发送的证书鉴别请求分组中 AAC 的询问字段值相同。
- ACC_{RES} 字段: 表示接入结果。
- REQ 密钥数据 $x \cdot P$ 元素: $x \cdot P$ 是 REQ 实体生成的用于 ECDH 交换的临时公钥,本字段值应与 REQ 实体发送的接入鉴别请求分组中 REQ 实体密钥数据字段值相同。
- AAC 密钥数据 $y \cdot P$ 元素: $y \cdot P$ 是 AAC 生成的用于 ECDH 交换的临时公钥。
- AAC 的身份 ID_{AAC} 元素: 表示 AAC 的身份标识 ID,定义同 D. 4. 1. 4。
- REQ 的身份 ID_{REQ} 元素: 表示 REQ 的身份标识 ID,定义同 D. 4. 1. 4。
- 复合的证书验证结果 MRES 字段: 其定义如 D. 4. 1. 12。此字段仅在双向鉴别过程中需要包含在接入鉴别响应分组中。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。其中签名为请求者 REQ 信任的鉴别服务器 AS 的签名 Sig_{AS-REQ}; 它是对复合的证书验证结果 MRES 中证书的验证结果字段的签名。
- 消息鉴别码 MIC1 字段定义如 D. 4. 1. 10,其计算过程为:
 - AAC 使用自己的临时私钥 y 和接入鉴别请求分组中 REQ 的临时公钥 $x \cdot P$ 进行 ECDH 计算,得到主密钥种子 $(x \cdot y \cdot P)_{\text{abscissa}}$;
 - AAC 对主密钥种子进行扩展 KD-HMAC-SHA256 ($(x \cdot y \cdot P)_{\text{abscissa}}, N_{\text{AAC}} || N_{\text{REQ}} ||$ “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;
 - 利用 BK 对接入鉴别响应分组中除消息鉴别码 MIC1 字段外的所有字段计算的杂凑值作为消息鉴别码 MIC1 的值。
- 签名 Sig_{AAC} 字段: 采用签名属性表示,是对本分组中除本字段之外所有数据字段的签名。

接入鉴别响应分组只需要包含 SIG_{AAC} 字段和消息鉴别码 MIC 字段二者之一即可;如果在此次身份鉴别过程中存在鉴别激活分组,且鉴别激活分组包含 SIG_{AAC} 字段,则此分组中只包含消息鉴别码

MIC 字段;如果此次身份鉴别过程不存在鉴别激活分组或者鉴别激活分组中没有包含 SIG_{AAC} 字段,则此分组中只包含 SIG_{AAC} 字段;

REQ 实体收到接入鉴别响应分组后,进行如下处理:

- a) 根据 ID_{AAC} 和 ID_{REQ} 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) 比较分组中请求者 REQ 密钥数据 $x \cdot P$ 字段值与自己发送的接入鉴别请求分组中的 $x \cdot P$ 字段值是否一致,若不一致,则丢弃该分组,否则执行 d);
- d) 比较 REQ 实体的询问 N_{REQ} 与自己在接入鉴别请求分组中发送的 N_{REQ} 是否相同,若不同,则丢弃该分组;否则,执行 e);
- e) 如果收到的接入鉴别响应分组中含有鉴别访问控制器 AAC 的签名 SIG_{AAC} 字段,执行 f);否则执行 h);
- f) 验证 SIG_{AAC} 的正确性,如果不正确,则丢弃该分组,否则执行 g);
- g) 查看分组中的 Acc_{RES} 字段,如果接入结果为不成功,则得知不能访问该网络,鉴别过程结束;否则执行 h);
- h) 若 REQ 在接入鉴别请求分组中不要求进行证书验证,则执行 i);否则,REQ 在复合的证书鉴别结果中查找自身所信任的 AS 的签名,验证 Sig_{AS-REQ} 的正确性,若不正确,则丢弃该接入鉴别响应分组;否则检查 AAC 证书的鉴别结果是否为有效,若无效,丢弃该分组;若有效,则执行 i);
- i) 请求者 REQ 根据鉴别访问控制器 AAC 的临时公钥 $y \cdot P$ 和自己的临时私钥 x 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 ($(x \cdot y \cdot P)_{abscissa}$, $N_{AAC} || N_{REQ} ||$ “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;
- j) 如果收到的接入鉴别响应分组中含有鉴别访问控制器 AAC 的签名 SIG_{AAC} 字段,则执行 l);否则执行 k);
- k) 验证消息鉴别码 MIC1 字段的正确性,如果不正确,则丢弃分组,否则执行 l);
- l) 如果收到的接入鉴别响应分组中含有消息鉴别码 MIC1 字段,则是否发送接入鉴别确认分组是可选的;如果收到的分组中含有鉴别访问控制器 AAC 的签名 SIG_{AAC} 字段,则需要构造接入鉴别确认分组,发送给鉴别访问控制器 AAC。

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:

- a) 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;
- b) 临时公钥 $x \cdot P, y \cdot P$ 是椭圆曲线域参数定义的椭圆曲线上的点;
- c) ECDH 协商出来密钥种子 $(x \cdot y \cdot P)_{abscissa}$ 是 $x \cdot y \cdot P$ 的 x 坐标, $x \cdot y \cdot P$ 不能是无穷远点。

D.7.1.3.7 接入鉴别确认分组

REQ 收到接入鉴别响应分组后,验证 N_{REQ} 、 ID_{REQ} 、 Sig_{AS-REQ} 等,若无效,则直接丢弃,否则构造接入鉴别确认分组返回给 AAC。如果收到的接入鉴别响应分组中包含 AAC 的签名 Sig_{AAC} ,则接入鉴别确认分组就是必须的。如果收到的接入鉴别响应分组中不包含 AAC 的签名 Sig_{AAC} ,而是包含了消息鉴别码 MIC1 字段,则接入鉴别确认分组就是可选的。接入鉴别确认分组主要包括:标识 TAEP_FLAG 及消息鉴别码 MIC2。

接入鉴别确认分组封装在 Code=2(Response), Type=245(TAEP-CAAP), MessageType = 0x06

的 TAEP-Response-CAAP/TAEPoL 中。

接入鉴别确认分组中,数据元素 SubData 都采用 D. 4. 2 介绍的封装格式进行封装。接入鉴别确认分组有效元素集合见表 D. 23。

表 D. 23 接入鉴别确认分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定义
标识 TAEP_FLAG 元素	0	1	D. 4. 1. 1
消息鉴别码 MIC2 元素	1	20	D. 4. 1. 10
保留	2~15		
保留	16~255		

其中:

——标识 TAEP_FLAG 元素:比特 0 有意义。本字段比特 0 应与 REQ 实体发送的接入鉴别请求分组中标识字段值相同。

——消息鉴别码 MIC2 元素,计算过程为:

- a) REQ 使用自己的临时私钥 x 和接入鉴别响应分组中 AAC 的临时公钥 $y \cdot P$ 进行 ECDH 计算,得到主密钥种子 $(x \cdot y \cdot P)_{\text{abscissa}}$;
- b) REQ 对主密钥种子进行扩展 KD-HMAC-SHA256 $((x \cdot y \cdot P)_{\text{abscissa}}, N_{\text{AAC}} || N_{\text{REQ}} || \text{“base key expansion for key and additional nonce”})$, 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;
- c) 利用 BK 对接入鉴别确认分组中除消息鉴别码 MIC2 字段外的所有字段计算的杂凑值作为消息鉴别码 MIC2 的值。

鉴别访问控制器 AAC 在发送接入鉴别响应分组给请求者之后,如果发送的接入鉴别响应分组中包含的是鉴别访问控制器 AAC 的签名 SIG_{AAC} 字段,则鉴别访问控制器 AAC 需要等待收到接入鉴别响应分组之后才能进行后续的密钥协商过程;如果发送的接入鉴别响应分组中包含的是消息鉴别码 MIC1 字段则可直接开始后续的密钥协商过程。

鉴别访问控制器 AAC 在收到请求者 REQ 发送的接入鉴别确认分组之后,要验证分组中消息鉴别码 MIC2 字段的正确性,如果正确,则意味着请求者 REQ 具有和自己一致的基密钥 BK,就可以开始后续的密钥协商过程,否则就解除与请求者 REQ 之间的链路验证。

AAC 根据证书鉴别响应分组中 REQ 证书的验证结果 Res_{REQ} 及本地策略,决定是否允许客户端 REQ 接入,再向 REQ 发送鉴别结果分组 TAEP-Success 或 TAEP-Failure。

REQ 收到接入鉴别响应分组后,验证 Sig_{AAC} 及其他字段,若无效,则直接丢弃;否则,根据 AAC 证书的验证结果 Res_{AAC} 及本地策略,决定是否接入该网络,若拒绝接入网络,可向 AAC 发送 TAEPoL-Logoff。基于证书的核心鉴别过程成功完成后,AAC 和 REQ 分别根据对方的临时公钥与本地的临时私钥进行 ECDH 计算,并将结果与双方生成的一次性随机数 $N_{\text{AAC}}, N_{\text{REQ}}$ 通过哈希计算得到下一次鉴别标识 SNonce ,用于鉴别更新时双方的同步锁定。

需要说明几点:

- a) 基于证书的核心鉴别过程支持双向与单向鉴别,若使用单向鉴别,则 AS 无需对 AAC 的证书 Cert_{AAC} 进行验证,即证书鉴别请求、证书鉴别响应及接入鉴别响应等分组中均不包含 Cert_{AAC} 、 Res_{AAC} 等字段。

- b) 该协议功能高度集中,即可实现双向鉴别,又可实现单向鉴别,还支持鉴别更新以及简化的鉴别更新。所谓简化的鉴别更新就是指不需要 AS 验证证书,AAC 与 REQ 之间的直接签名验证,反映在分组中的内容上,就是接入鉴别响应分组中不包含复合的证书验证结果字段。简化的鉴别过程只能用作鉴别更新过程,不能用作客户端与网络连接时的首次鉴别。
- c) TAEP-CLAP 提供了 AAC 与 REQ 之间对 ECDH 交换结果的一致性验证机制。文中描述了两种验证机制:
 - 1) 在鉴别激活分组中添加 AAC 的签名,然后在接入鉴别响应分组中不再附加签名,而是利用协商出来的 BK 计算分组的消息鉴别码消息鉴别码 MIC1。在这种机制下,REQ 可以验证 AAC 的签名,也可以通过验证 AAC 利用 BK 计算的消息鉴别码 MIC1 来验证它们拥有一致的 BK。这种机制下接入鉴别确认分组就是可选的。
 - 2) 在鉴别激活分组中不附带 AAC 的签名,而是在接入鉴别响应分组中附带签名,之后 AAC 需要等到接收到 REQ 发送的接入鉴别确认分组后验证 REQ 利用 BK 计算的消息鉴别码 MIC2,进行确认后方可确认与 REQ 拥有一致的 BK,之后就可以开始密钥协商的过程了。

D.7.1.4 单播密钥协商

D.7.1.4.1 单播密钥协商过程概述

基于证书的鉴别和密钥管理方法中,密钥管理包括单播密钥协商与组播密钥通告过程。TLA 的密钥管理协议在单播密钥协商协议的请求分组中添加消息鉴别码 MIC 校验字段,防止 Dos 攻击。其中,该消息鉴别码 MIC 字段利用鉴别过程协商的 BK 计算得到;此外,在密钥协商协议的分组定义中,添加标识业务类型字段,实现不同业务不同密钥。

单播密钥协商使用预共享密钥完成单播会话密钥的协商,建立 USKSA。单播密钥协商过程见图 D.52。

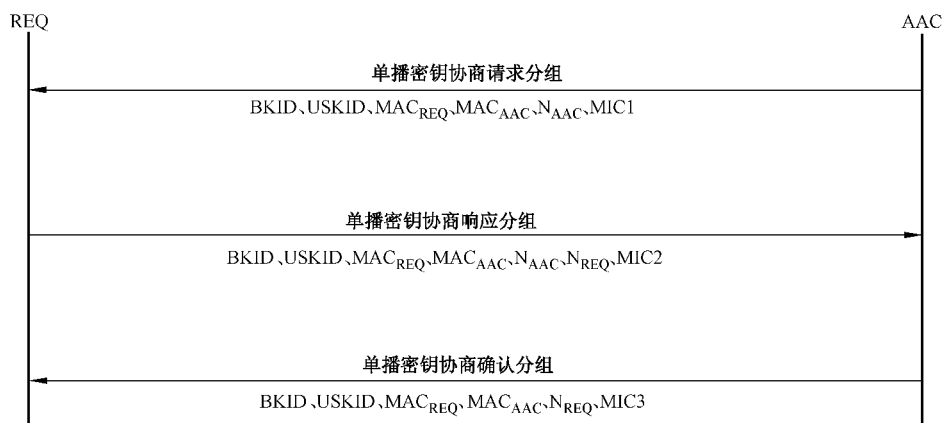


图 D.52 单播密钥协商过程

单播密钥协商协议分组使用 TAEPoL-Key 帧封装,其协议数据字段中 Key Descriptor 类型取值为 0x10,其协议数据字段中的数据字段封装格式见图 D.53。

MessageType (8位比特)	MessageData (可变)
-----------------------	------------------

图 D.53 单播密钥协商协议分组数据封装格式

其中：

——MessageType,表示消息分组所属的分组类别,定义如下表 D. 24。

表 D. 24 单播密钥协商协议分组 MessageType 子类型定义

MessageType 值	定 义
0x00	保留
0x01	单播密钥协商请求分组
0x02	单播密钥协商响应分组
0x03	单播密钥协商确认分组
其余	保留

——MessageData 数据:每种分组的消息数据 MessageData 将在协议分组介绍中进行介绍。其数据元素封装格式见 D. 4. 2。

D. 7. 1. 4. 2 单播密钥协商请求分组

D. 7. 1. 4. 2. 1 帧格式

D. 7. 1. 4. 2. 1. 1 Key_FLAG

Key_FLAG 定义见 D. 4. 3. 5. 3。

Key_FLAG. ACK=1

Key_FLAG. KeyType=000(单播密钥)

Key_FLAG. Request=1

Key_FLAG. Encryption=0

Key_FLAG. MIC=1

Key_FLAG. OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D. 7. 1. 4. 2. 1. 2 协议数据

协议数据包含 BKID、USKID、MAC_{REQ}、MAC_{AAC}、N_{AAC}。分组中的协议数据的集合及元素 ID 定义见表 D. 25。

表 D. 25 单播密钥协商请求分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定义
BKID	0	16	D. 4. 1. 20
USKID	1	1	D. 4. 1. 21
MAC _{REQ}	2	6	D. 4. 1. 8
MAC _{AAC}	3	6	D. 4. 1. 8
N _{AAC}	4	32	D. 4. 1. 3
保留	5~15		
保留	16~255		

其中：

- BKID：表示当前作为共享密钥的基密钥，其定义见 D. 4. 1. 20；
- USKID：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，其定义见 D. 4. 1. 21；
- MAC_{REQ}：请求者 REQ 的 MAC 地址，其定义见 D. 4. 1. 8；
- MAC_{AAC}：鉴别访问控制器 AAC 的 MAC 地址，其定义见 D. 4. 1. 8；
- N_{AAC}：AAC 询问字段定义见 D. 4. 1. 3；若标识 Key_FLAG 字段的 OperationType 字段的值为 00，则 AAC 的询问为 AAC 产生的随机数；若标识 Key_FLAG 字段的 OperationType 字段的值为 01 或者 10，则 AAC 询问为上一次单播密钥协商过程所协商的值。

D. 7. 1. 4. 2. 1. 3 消息鉴别码 MIC

该消息鉴别码 MIC 字段记为 MIC1，是鉴别访问控制器 AAC 利用鉴别过程协商的 BK 计算得到。目的为防止 Dos 攻击，其定义见 D. 4. 1. 10。

D. 7. 1. 4. 2. 2 处理过程



在 AAC 完成基于证书鉴别过程并建立有效的 BKSA 后，或进行单播密钥更新时，AAC 向 REQ 发送单播密钥协商请求分组开始与 REQ 进行单播密钥协商。

REQ 接收到与其相关联的 AAC 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 Key_FLAG 字段的 OperationType 字段的值，若为 00，则执行 c)；若为 01，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 AAC 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) 验证消息鉴别码 MIC 的正确性，如果消息鉴别码 MIC 不正确则丢弃该分组，否则执行 d)。
- d) REQ 利用随机数产生器产生 REQ 询问 N_{REQ}，然后计算 KD-HMAC-SHA256 (BK, ADDID || N_{AAC} || N_{REQ} || “pairwise key expansion for unicast and additional keys and nonce”)，其中 N_{AAC} 为 AAC 询问，N_{REQ} 为 REQ 询问。生成 80 个八位位组，前 48 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 个八位位组为消息鉴别密钥 MAK，第三个 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 AAC 询问的种子，即协商标识种子，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 AAC 询问并保存。
- e) 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 AAC。
- f) 安装新协商的单播会话密钥。对于新安装的密钥，REQ 仅启用其接收功能，即允许用其解密 AAC 发来的单播数据。

D. 7. 1. 4. 3 单播密钥协商响应分组

D. 7. 1. 4. 3. 1 帧格式

D. 7. 1. 4. 3. 1. 1 Key_FLAG

Key_FLAG 定义见 D. 4. 3. 5. 3。

Key_FLAG.ACK=1

Key_FLAG.KeyType=000(单播密钥)

Key_FLAG.Request=1

Key_FLAG.Encryption=0

Key_FLAG.MIC=1

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.7.1.4.3.1.2 协议数据

协议数据包含 BKID、USKID、MAC_{REQ}、MAC_{AAC}、N_{AAC}、N_{REQ}。分组中的协议数据的集合及元素 ID 定义见表 D.26。

表 D.26 单播密钥协商响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定义
BKID	0	16	D.4.1.20
USKID	1	1	D.4.1.21
MAC _{REQ}	2	6	D.4.1.8
MAC _{AAC}	3	6	D.4.1.8
N _{AAC}	4	32	D.4.1.3
N _{REQ}	5	32	D.4.1.3
保留	6~15		
保留	16~255		

其中：

- BKID：表示当前作为共享密钥的基密钥，其定义见 D.4.1.20；
- USKID：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，其定义见 D.4.1.21；
- MAC_{REQ}：请求者 REQ 的 MAC 地址，其定义见 D.4.1.8；
- MAC_{AAC}：鉴别访问控制器 AAC 的 MAC 地址，其定义见 D.4.1.8；
- N_{AAC}：是 AAC 询问字段定义见 D.4.1.3，字段长度为 32 个八位位组；若 REQ 发起密钥更新或者密钥删除，REQ 设置标识 Key_FLAG 字段的 OperationType 字段的值为 01 或者 10，AAC 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 AAC 询问字段相同；
- N_{REQ}：是 REQ 询问字段定义见 D.4.1.3，字段长度为 32 个八位位组，由 REQ 利用随机数产生器生成。

D.7.1.4.3.1.3 消息鉴别码 MIC

记为 MIC2，其值为 REQ 利用最新协商的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法计算得到。其定义见 D.4.1.10。

D.7.1.4.3.2 处理过程

REQ 进行密钥更新时，或收到 AAC 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 AAC。

AAC 收到单播密钥协商响应分组后,进行如下处理:

- a) 若标识 Key_FLAG 字段的 OperationType 字段为 01,执行 b)操作;若标识 Key_FLAG 字段的 OperationType 字段为 00,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 AAC 询问值 N_{AAC} 与之前发送的单播密钥协商请求分组中的 N_{AAC} 值是否一致,若不一致,则丢弃该分组;若一致,执行 d)操作。
- d) 计算 $KD-HMAC-SHA256(BK, ADDID || N_{AAC} || N_{REQ} || \text{“pairwise key expansion for unicast and additional keys and nonce”})$,其中 N_{AAC} 是 AAC 询问, N_{REQ} 是 REQ 询问。生成 80 个八位位组,前 48 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 个八位位组为消息鉴别密钥 MAK,第三个 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 AAC 询问的种子,即协商标识种子,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 AAC 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) 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 REQ。
- f) AAC 安装新协商的单播会话密钥。对于新安装的密钥,AAC 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后,删除旧的单播会话密钥;或者启用新密钥收发数据 60 s 之后,自动删除旧的单播密钥。

D.7.1.4.4 单播密钥协商确认分组

D.7.1.4.4.1 帧格式

D.7.1.4.4.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=0

Key_FLAG.KeyType=000(单播密钥)

Key_FLAG.Request=1

Key_FLAG.Encryption=0

Key_FLAG.MIC=1

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.7.1.4.4.1.2 协议数据

协议数据包含 BKID、USKID、 MAC_{REQ} 、 MAC_{AAC} 、 N_{REQ} 。分组中的协议数据的集合及元素 ID 定义见表 D.27。

表 D.27 鉴别激活分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
BKID	0	16	D.4.1.20
USKID	1	1	D.4.1.21
MAC_{REQ}	2	6	D.4.1.8

表 D. 27 (续)

信息元素	元素 ID	元素长度 (八位位组)	定 义
MAC _{AAC}	3	6	D. 4. 1. 8
N _{REQ}	4	32	D. 4. 1. 3
保留	5~15		
保留	16~255		

其中:

- BKID:表示当前作为共享密钥的基密钥,其定义见 D. 4. 1. 20;
- USKID: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,其定义见 D. 4. 1. 21;
- MAC_{REQ}:请求者 REQ 的 MAC 地址,其定义见 D. 4. 1. 8;
- MAC_{AAC}:鉴别访问控制器 AAC 的 MAC 地址,其定义见 D. 4. 1. 8;
- N_{REQ}:REQ 询问字段 N_{REQ}定义同 D. 4. 1. 3,长度为可变位位组;由 REQ 利用随机数产生器生成,字段值同单播密钥协商响应分组中的 REQ 询问字段 N_{REQ}值。

D. 7. 1. 4. 4. 1. 3 消息鉴别码 MIC

记为 MIC3,其定义见 D. 4. 1. 10,值为 AAC 利用最新协商的消息鉴别密钥 MAK 对单播密钥协商确认分组中除本字段外所有字段及计算得到的下一次单播密钥协商过程中的 AAC 询问通过 HMAC-SHA256 算法计算得到。

D. 7. 1. 4. 4. 2 处理过程

AAC 收到单播密钥响应分组后,发送单播密钥协商确认分组给 REQ。

REQ 接收到 AAC 的单播密钥协商确认分组后,进行如下处理:

- a) 比较 REQ 询问 N_{REQ}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) 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。

D. 7. 2 基于预共享密钥的鉴别及单播密钥管理

D. 7. 2. 1 身份鉴别及单播密钥协商过程概述

基于预共享密钥的鉴别及密钥管理过程将身份鉴别和单播密钥协商综合在一起,在完成单播密钥协商的同时,也完成了双向鉴别过程。基于预共享密钥的鉴别及密钥管理过程主要包含四个分组,见图 D. 54。

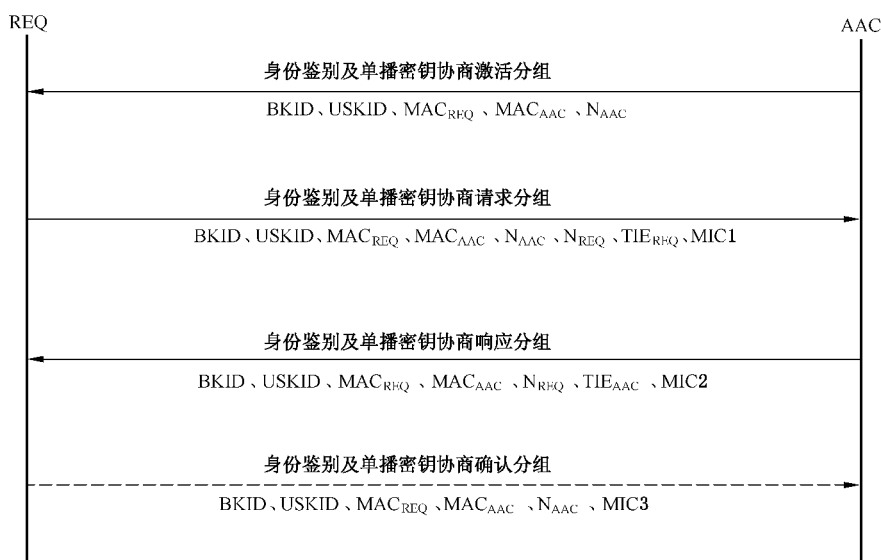


图 D.54 身份鉴别及单播密钥协商过程

身份鉴别及单播密钥协商过程分组使用 TAEPoL-Key 帧封装,其协议数据字段中 Key Descriptor 类型取值为 0x11,其协议数据字段中的数据字段封装格式见图 D.55。

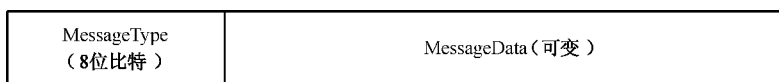


图 D.55 身份鉴别及单播密钥协商协议分组数据封装格式

其中:

——MessageType,表示消息分组所属的分组类别,定义如下表 D.28。

表 D.28 身份鉴别及单播密钥协商协议分组 MessageType 子类型定义

MessageType 值	定义
0x00	保留
0x01	身份鉴别及单播密钥协商激活分组
0x02	身份鉴别及单播密钥协商请求分组
0x03	身份鉴别及单播密钥协商响应分组
0x04	身份鉴别及单播密钥协商确认分组
其余	保留

——MessageData 数据:每种分组的消息数据 MessageData 将在协议分组介绍中进行介绍。其数据元素封装格式见 D.4.2。

D.7.2.2 身份鉴别及单播密钥协商激活分组

D.7.2.2.1 帧格式

D.7.2.2.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=1
 Key_FLAG.KeyType=000(单播密钥)
 Key_FLAG.Request=1
 Key_FLAG.Encryption=0
 Key_FLAG.MIC=0
 Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.7.2.2.1.2 协议数据

协议数据包含 BKID、USKID、MAC_{REQ}、MAC_{AAC}、N_{AAC}。分组中的协议数据的集合及元素 ID 定义见表 D.29。

表 D.29 身份鉴别及单播密钥协商激活分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
BKID	0	16	D.4.1.20
USKID	1	1	D.4.1.21
MAC _{REQ}	2	6	D.4.1.8
MAC _{AAC}	3	6	D.4.1.8
N _{AAC}	4	32	D.4.1.3
保留	5~15		
保留	16~255		

其中：

- BKID:表示当前作为共享密钥的基密钥,其定义见 D.4.1.20;其中 BK 是由预共享密钥导出。
- USKID: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,其定义见 D.4.1.21。
- MAC_{REQ}:请求者 REQ 的 MAC 地址,其定义见 D.4.1.8。
- MAC_{AAC}:鉴别访问控制器 AAC 的 MAC 地址,其定义见 D.4.1.8。
- N_{AAC}:AAC 询问字段定义见 D.4.1.3。若标识 Key_FLAG 字段的 OperationType 字段的值为 00,则 AAC 的询问为 AAC 产生的随机数;若标识 Key_FLAG 字段的 OperationType 字段的值为 01 或者 10,则 AAC 询问为上一次身份鉴别及单播密钥协商过程所协商的值。

D.7.2.2.2 处理过程

REQ 与 AAC 在安全策略协商过程中,如果选择使用基于预共享密钥的鉴别和密钥管理方法,或者 REQ 与 AAC 选择利用基于预共享密钥的鉴别和密钥管理方法来更新它们之间的单播密钥时,则直接利用它们之前预共享的密钥开始鉴别与单播密钥协商过程。AAC 构造身份鉴别及单播密钥协商激活分组,发送给 REQ。

REQ 接收到 AAC 发送的身份鉴别及单播密钥协商激活分组后,进行如下处理:

- a) 首先检查标识 Key_FLAG 中 OperationType 字段值,若为 01 或者 10,则检查 BKID 所指的

BKSA 以及 USKID 所指的 USKSA 是否有效,若其中一个无效,则丢弃该分组,否则执行 b); 如果标识 Key_FLAG 中 OperationType 字段值为 00,则直接执行 c)。

- b) 检查分组中的 N_{AAC} 与上次身份鉴别及单播密钥协商过程中保存的身份鉴别及单播密钥协商标识的值是否一致,若不一致,则丢弃该分组;若一致,则进一步查看标识 Key_FLAG 中 OperationType 字段值,若为 01 则为更新过程,执行 c);若为 10 则为删除过程,则构造身份鉴别及单播密钥协商响应分组发送给 AAC,并删除与 AAC 之间的单播密钥,身份鉴别及单播密钥协商激活分组接收处理完成。
- c) 利用预共享密钥,通过 KD-HMAC-SHA256(PreKey,“Preshared key expansion for unicast and additional keys and nonce”)计算得到 16 个八位位组的基密钥 BK。
- d) REQ 利用随机数产生器产生 REQ 询问 N_{REQ} ,然后计算 KD-HMAC-SHA256 (BK ,ADDID|| N_{AAC} || N_{REQ} ||“pairwise key expansion for unicast and additional keys and nonce”),其中 N_{AAC} 为 AAC 询问, N_{REQ} 为 REQ 询问。生成 80 个八位位组,前 48 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 个八位位组为消息鉴别密钥 MAK,第三个 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 AAC 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 AAC 询问并保存。
- e) 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 AAC。
- f) 安装新协商的单播会话密钥。

注:如果请求者 REQ 需要发起身份鉴别及单播密钥协商的更新过程,则可直接执行上述 c),d)和 e),其中身份鉴别及单播密钥协商标识 NAAC 字段值取决于上一次身份鉴别及单播密钥协商过程中计算的身份鉴别及单播密钥协商标识值,构造身份鉴别及单播密钥协商请求分组发送给鉴别访问控制器 AAC,无需等待鉴别访问控制器 AAC 先发送身份鉴别及单播密钥协商激活分组。

D.7.2.3 身份鉴别及单播密钥协商请求分组

D.7.2.3.1 帧格式

D.7.2.3.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=1

Key_FLAG.KeyType=000(单播密钥)

Key_FLAG.Request=1

Key_FLAG.Encryption=0

Key_FLAG.MIC=1

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.7.2.3.1.2 协议数据

协议数据包含 BKID、USKID、 MAC_{REQ} 、 MAC_{AAC} 、 N_{AAC} 、 N_{REQ} 。分组中的协议数据的集合及元素 ID 定义见表 D.30。

表 D.30 身份鉴别及单播密钥协商请求分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
BKID	0	16	D. 4. 1. 20
USKID	1	1	D. 4. 1. 21
MAC _{REQ}	2	6	D. 4. 1. 8
MAC _{AAC}	3	6	D. 4. 1. 8
N _{AAC}	4	32	D. 4. 1. 3
N _{REQ}	5	32	D. 4. 1. 3
TIE _{REQ}	6	可变	D. 4. 1. 14
保留	7~15		
保留	16~255		

其中：

- BKID：表示当前作为共享密钥的基密钥，其定义见 D. 4. 1. 20；
- USKID：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；其定义见 D. 4. 1. 21；
- MAC_{REQ}：请求者 REQ 的 MAC 地址，其定义见 D. 4. 1. 8；
- MAC_{AAC}：鉴别访问控制器 AAC 的 MAC 地址，其定义见 D. 4. 1. 8；
- N_{AAC}：是 AAC 询问字段定义见 D. 4. 1. 3，字段长度为 32 个八位位组；若 REQ 发起密钥更新或者密钥删除，REQ 设置标识 Key_FLAG 字段的 OperationType 字段的值为 01 或者 10，AAC 询问字段为上一次身份鉴别及单播密钥协商过程所协商的值；否则该字段和身份鉴别及单播密钥协商激活分组中的 AAC 询问字段相同；
- N_{REQ}：是 REQ 询问字段定义见 D. 4. 1. 3，字段长度为 32 个八位位组，由 REQ 利用随机数产生器生成；
- TIE_{REQ}：REQ 所支持的鉴别和密钥管理套件及密钥套件等信息；该字段值与安全策略协商过程中 REQ 发送的安全策略协商响应分组中 TIE_{REQ} 字段值相同，其定义见 D. 4. 1. 14。

D.7.2.3.1.3 消息鉴别码 MIC

记为 MIC1 字段，其值为 REQ 利用最新协商的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法对身份鉴别及单播密钥协商请求分组中除本字段外所有字段进行计算得到的杂凑值。

D.7.2.3.2 处理过程

AAC 收到身份鉴别及单播密钥协商请求分组后，进行如下处理：

- a) 若标识字段的 OperationType 字段值为 01 或者 10，执行 b) 操作；若标识字段的 OperationType 字段值为 00，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) 检查身份鉴别及单播密钥协商标识是否正确，若不正确，则丢弃该分组；否则，执行 d) 操作。
- d) 检查分组中的 TIE_{REQ} 字段值与安全策略协商过程中 REQ 发送的安全策略协商响应分组中 TIE_{REQ} 字段值是否相同，如果不相同，则丢弃该分组；否则，执行 e)。

- e) 利用之前通过预共享密钥计算得到的 BK 计算 KD-HMAC-SHA256(BK, ADDID || N_{AAC} || N_{REQ} || “pairwise key expansion for unicast and additional keys and nonce”), 其中 N_{AAC} 是 AAC 询问, N_{REQ} 是 REQ 询问。生成 80 个八位位组, 前 48 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 个八位位组为消息鉴别密钥 MAK, 第三个 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 AAC 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 AAC 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。
- f) 用消息鉴别密钥 MAK 通过 HMAC-SHA256 算法本地计算消息鉴别码, 构造身份鉴别及单播密钥协商响应分组, 发送给 REQ。
- g) AAC 安装新协商的单播会话密钥。

D.7.2.4 身份鉴别及单播密钥协商响应分组

D.7.2.4.1 帧格式

D.7.2.4.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=0

Key_FLAG.KeyType=000(单播密钥)

Key_FLAG.Request=1

Key_FLAG.Encryption=0

Key_FLAG.MIC=1

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.7.2.4.1.2 协议数据

协议数据包含 BKID、USKID、MAC_{REQ}、MAC_{AAC}、N_{REQ}。分组中的协议数据的集合及元素 ID 定义见表 D.31。

表 D.31 身份鉴别及单播密钥协商响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
BKID	0	16	D.4.1.20
USKID	1	1	D.4.1.21
MAC _{REQ}	2	6	D.4.1.8
MAC _{AAC}	3	6	D.4.1.8
N _{REQ}	4	32	D.4.1.3
TIE _{AAC}	5	可变	D.4.1.14
保留	6~15		
保留	16~255		

其中：

- BKID：表示当前作为共享密钥的基密钥，其定义见 D. 4. 1. 20；
- USKID：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，其定义见 D. 4. 1. 21；
- MAC_{REQ}：请求者 REQ 的 MAC 地址，其定义见 D. 4. 1. 8；
- MAC_{AAC}：鉴别访问控制器 AAC 的 MAC 地址，其定义见 D. 4. 1. 8；
- N_{REQ}：REQ 询问字段 N_{REQ} 定义同 D. 4. 1. 3，长度为可变位位组；由 REQ 利用随机数产生器生成，字段值同身份鉴别及单播密钥协商请求分组中的 REQ 询问字段 N_{REQ} 值；
- TIE_{AAC}：AAC 所支持的鉴别和密钥管理套件及密钥套件等信息；该字段值与安全策略协商过程中 AAC 发送的安全策略协商请求分组中 TIE_{AAC} 字段值相同，其定义见 D. 4. 1. 14。

D. 7. 2. 4. 1. 3 消息鉴别码 MIC

记为 MIC2 字段，表示消息鉴别码，由鉴别访问控制器 AAC 利用生成的消息鉴别密钥 MAK 对身份鉴别及单播密钥协商响应分组中除本字段外所有字段进行计算得到的杂凑值。

D. 7. 2. 4. 2 处理过程

REQ 接收到 AAC 的身份鉴别及单播密钥协商响应分组后，进行如下处理：

- a) 检查 N_{REQ} 字段与之前发送的身份鉴别及单播密钥协商请求分组中的 N_{REQ} 字段是否相同，若不同，则丢弃该分组；否则，执行 b) 操作；
- b) 验证 TIE_{AAC} 字段值与安全策略协商过程中收到的安全策略协商请求分组中的 TIE_{AAC} 字段值是否一致，如果不一致，则丢弃该分组；否则执行 c) 操作；
- c) 利用消息鉴别密钥 MAK 验证身份鉴别及单播密钥协商响应分组中的消息鉴别码 MIC2 字段的正确性，如果不正确则丢弃该分组；否则请求者 REQ 完成对鉴别访问控制器 AAC 的身份鉴别，并协商出和鉴别访问控制器 AAC 之间一致的单播会话密钥，则执行 d) 操作；
- d) 用消息鉴别密钥 MAK 本地计算消息鉴别码消息鉴别码 MIC3，构造身份鉴别及单播密钥协商确认分组，发送给鉴别访问控制器 AAC；是否发送身份鉴别及单播密钥协商确认分组是可选的；
- e) 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。

D. 7. 2. 5 身份鉴别及单播密钥协商确认分组

D. 7. 2. 5. 1 帧格式

D. 7. 2. 5. 1. 1 Key_FLAG

Key_FLAG 定义见 D. 4. 3. 5. 3。

Key_FLAG. ACK=0

Key_FLAG. KeyType=000(单播密钥)

Key_FLAG. Request=1

Key_FLAG. Encryption=0

Key_FLAG. MIC=1

Key_FLAG. OperationType = 00(建立过程)/01(更新过程)/10(删除过程)

D.7.2.5.1.2 协议数据

协议数据包含 BKID、USKID、 MAC_{REQ} 、 MAC_{AAC} 、 N_{AAC} 、 N_{REQ} 。分组中的协议数据的集合及元素 ID 定义见表 D.32。

表 D.32 身份鉴别及单播密钥协商确认分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
BKID	0	16	D.4.1.20
USKID	1	1	D.4.1.21
MAC_{REQ}	2	6	D.4.1.8
MAC_{AAC}	3	6	D.4.1.8
N_{AAC}	4	32	D.4.1.3
保留	5~15		
保留	16~255		

其中：

- BKID：表示当前作为共享密钥的基密钥，其定义见 D.4.1.20；
- USKID：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，其定义见 D.4.1.21；
- MAC_{REQ} ：请求者 REQ 的 MAC 地址，其定义见 D.4.1.8；
- MAC_{AAC} ：鉴别访问控制器 AAC 的 MAC 地址，其定义见 D.4.1.8；
- N_{AAC} ：表示身份鉴别及单播密钥协商标识 N_{AAC} ，该字段值同身份鉴别及单播密钥协商请求分组中的 N_{AAC} 字段的值；其定义见 D.4.1.3。

D.7.2.5.1.3 消息鉴别码 MIC

记为 MIC3 字段，表示消息鉴别码，由请求者 REQ 利用生成的消息鉴别密钥 MAK 对身份鉴别及单播密钥协商确认分组中除本字段外所有字段及下一次身份鉴别及单播密钥协商过程中的身份鉴别及单播密钥协商标识进行计算得到的杂凑值。

D.7.2.5.2 处理过程

AAC 接收到 REQ 的身份鉴别及单播密钥协商确认分组后，进行如下处理：

- a) 检查 N_{AAC} 字段值与收到的身份鉴别及单播密钥协商请求分组中的 N_{AAC} 字段值是否相同，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 验证身份鉴别及单播密钥协商确认分组中的消息鉴别码 MIC3 字段的正确性，如果不正确则丢弃该分组；否则鉴别访问控制器 AAC 可确认与请求者 REQ 之间具有一致的单播会话密钥。

对于新安装的密钥，AAC 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 s 之后，自动删除旧的单播密钥。

D.8 组播密钥通告

D.8.1 组播密钥通告过程概述

组播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通告,并建立 MSKSA。组播密钥通告的核心过程见图 D.56。



图 D.56 组播密钥通告的核心过程

组播密钥通告分组和组播密钥响应分组使用 TAEPoL-Key 帧封装,其协议数据字段中 Key Descriptor 类型取值为 0x12,其协议数据字段中的数据字段封装格式见图 D.57。

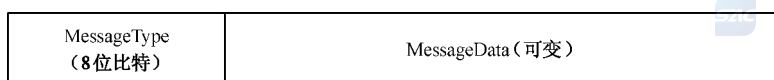


图 D.57 组播密钥通告协议分组数据封装格式

其中:

——MessageType,表示消息分组所属的分组类别,定义如下表 D.33。

表 D.33 组播密钥通告协议分组 MessageType 子类型定义

MessageType 值	定 义
0x00	保留
0x01	组播密钥通告分组
0x02	组播密钥响应分组
其余	保留

——MessageData 数据:每种分组的消息数据 MessageData 将在协议分组介绍中进行介绍。其数据元素封装格式见 D.4.2。

D.8.2 组播密钥通告分组

单播密钥协商成功后,或 AAC 要更新组播密钥时, AAC 向 REQ 发送组播密钥通告分组通告组播主密钥。

D.8.2.1 帧格式

组播密钥通告分组使用 TAEPoL-Key 帧封装,各字段的定义如下。

D. 8. 2. 1. 1 Key_FLAG

Key_FLAG 定义见 D. 4. 3. 5. 3。

Key_FLAG. ACK=1

Key_FLAG. KeyType=001(组播密钥)

Key_FLAG. Request=0

Key_FLAG. Encryption=1

Key_FLAG. MIC=1

Key_FLAG. OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D. 8. 2. 1. 2 协议数据

协议数据包含 USKID、MSKID、MAC_{REQ}、MAC_{AAC}、KN、E(MSK)。分组中的协议数据的集合及元素 ID 定义见表 D. 34。

表 D. 34 组播密钥通告分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
USKID	0	16	D. 4. 1. 21
MSKID	1	1	D. 4. 1. 22
MAC _{REQ}	2	6	D. 4. 1. 8
MAC _{AAC}	3	6	D. 4. 1. 8
KN	4	16	
E(MSK)	5	32	
保留	5~15		
保留	16~255		

其中：

- USKID: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,其定义见 D. 4. 1. 21;
- MSKID: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,其定义见 D. 4. 1. 22;
- MAC_{REQ}:请求者 REQ 的 MAC 地址,其定义见 D. 4. 1. 8;
- MAC_{AAC}:鉴别访问控制器 AAC 的 MAC 地址,其定义见 D. 4. 1. 8;
- KN:表示密钥通告标识,初始化为一个整数,在每次密钥更新通告时该字段值加 1,若通告的密钥不变,则该字段值保持不变;
- E(MSK):其内容字段是 AAC 利用密钥加密密钥 KEK 采用协商选择的单播密码算法对要通告的组播密钥 MSK 加密后的密文(不带 MIC),要通告的组播密钥 MSK 为 AAC 生成的 16 个八位位组的随机数。

D. 8. 2. 1. 3 消息鉴别码 MIC

其值为 AAC 利用 USKID 字段标识的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法计算得到,其定义见 D. 4. 1. 10。

D.8.2.2 处理过程

REQ 接收到 AAC 发送的组播密钥通告分组后,进行如下处理:

- a) 检查密钥通告标识 KN 字段值是否单调递增,若为单调递增,则执行 b) 操作;否则丢弃该分组;
- b) REQ 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 c) 操作;
- c) 对密钥数据解密得到 16 个八位位组的组播密钥 MSK;
- d) 保存密钥通告标识字段值,生成组播密钥响应分组,发送给 AAC;
- e) 安装新的组播会话密钥,并启用其接收功能;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后,删除旧的组播密钥。

若 AAC 通告了新的组播密钥,REQ 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 REQ 接收到 AAC 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。

D.8.3 组播密钥通告响应分组

D.8.3.1 帧格式

组播密钥响应分组使用 TAEPoL-Key 帧封装,各字段的定义如下。

D.8.3.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=0

Key_FLAG.KeyType=001(组播密钥)

Key_FLAG.Request=0

Key_FLAG.Encryption=0

Key_FLAG.MIC=1

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.8.3.1.2 协议数据

协议数据包含 USKID、MSKID、MAC_{REQ}、MAC_{AAC}、KN。分组中的协议数据的集合及元素 ID 定义见表 D.35。

表 D.35 组播密钥通告响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
USKID	0	16	D.4.1.21
MSKID	1	1	D.4.1.22
MAC _{REQ}	2	6	D.4.1.8
MAC _{AAC}	3	6	D.4.1.8
KN	4	16	
保留	5~15		
保留	16~255		

其中：

- USKID: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,其定义见 D. 4. 1. 21;
- MSKID: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,其定义见 D. 4. 1. 22;
- MAC_{REQ}:请求者 REQ 的 MAC 地址,其定义见 D. 4. 1. 8;
- MAC_{AAC}:鉴别访问控制器 AAC 的 MAC 地址,其定义见 D. 4. 1. 8;
- KN:表示密钥通告标识,其值同组播密钥通告响应分组中 KN 字段值。

D. 8. 3. 1. 3 消息鉴别码 MIC

其值为 REQ 利用 USKID 字段标识的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法计算得到,其定义见 D. 4. 1. 10。

D. 8. 3. 2 处理过程

REQ 向 AAC 发送组播密钥响应分组,AAC 接收到 REQ 发送的组播密钥响应分组后,进行如下处理:

- a) 比较标识 Key_FLAG 字段的 OperationType 字段、MSKID 字段、USKID 字段、MAC_{REQ}、MAC_{AAC} 字段和密钥通告标识字段与发送的组播密钥通告分组中的相应字段值,若有一个不同,则丢弃该分组;否则,执行 b) 操作;
- b) 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,若相同,则本次组播密钥通告成功,执行 c) 操作;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,则安装新密钥;AAC 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 REQ,则删除旧密钥。

AAC 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 AAC 的 REQ 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。

D. 9 站间密钥建立

D. 9. 1 站间密钥建立过程概述

交换设备 SW 下的直连用户终端是指直接连接在交换设备 SW 某个端口下的用户终端,包括通过网线直接连接到交换设备 SW 的用户终端及通过集线器(hub)等物理层设备连接到交换设备 SW 的用户终端。通过其他设备连接到交换设备 SW 的用户终端不属于交换设备 SW 的直连用户终端。

当第一用户终端 STA₁ 要与第二用户终端 STA₂ 进行保密通信时,若用户终端 STA₁ 与用户终端 STA₂ 是同一交换设备 SW 下直连用户终端,第一用户终端 STA₁ 可请求通过交换设备 SW 建立与第二用户终端 STA₂ 共享的站间密钥,并利用该站间密钥保障与第二用户终端 STA₂ 之间的数据保密通信。站间密钥的建立过程主要包括站间密钥请求、站间密钥对 STA₂ 的通告、STA₂ 的响应、站间密钥对 STA₁ 的通告、STA₁ 的响应,具体描述见图 D. 58。

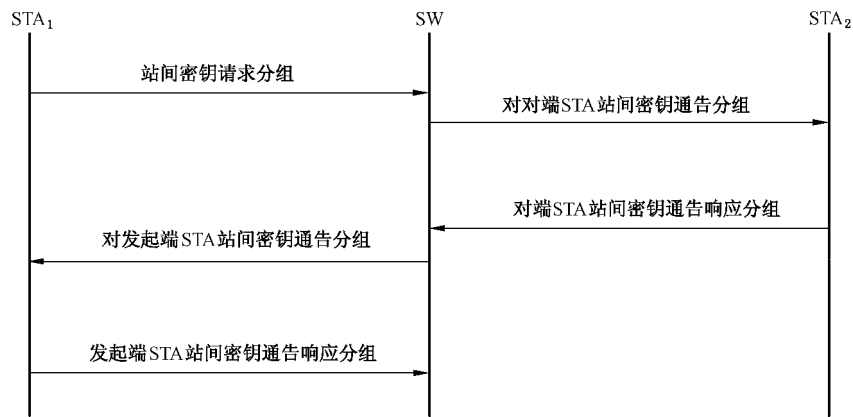


图 D. 58 站间密钥建立过程图示

站间密钥建立协议分组使用 TAEPoL-Key 帧封装,其协议数据字段中 Key Descriptor 类型取值为 0x13,其协议数据字段中的数据字段封装格式见图 D. 59。

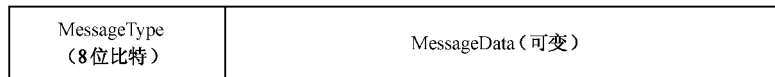


图 D. 59 站间密钥建立协议分组数据封装格式

其中:

——MessageType,表示消息分组所属的分组类别,定义如下表 D. 36。

表 D. 36 站间密钥建立协议分组 MessageType 子类型定义

MessageType 值	定 义
0x00	保留
0x01	站间密钥请求分组
0x02	对对端 STA 站间密钥通告分组
0x03	对端 STA 站间密钥通告响应分组
0x04	对发起端 STA 站间密钥通告分组
0x05	发起端 STA 站间密钥通告响应分组
其余	保留

——MessageData 数据:每种分组的消息数据 MessageData 将在协议分组介绍中进行介绍。其数据元素封装格式见 D. 4. 2。

D. 9. 2 站间密钥请求

当第一用户终端 STA₁ 要与第二用户终端 STA₂ 进行保密通信时,若用户终端 STA₁ 与用户终端 STA₂ 是同一交换设备 SW 下的直连用户终端,第一用户终端 STA₁ 首先检查本地是否保存有与第二用户终端 STA₂ 共享的站间密钥。若有,则使用站间密钥加密数据包;若没有,则第一用户终端 STA₁ 构造站间密钥请求分组,发送给交换设备 SW。

D.9.2.1 帧格式

D.9.2.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=1

Key_FLAG.KeyType=010(站间密钥)

Key_FLAG.Request=0


Key_FLAG.Encryption=0

Key_FLAG.MIC=0

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.9.2.1.2 协议数据

协议数据包含 USKID、STAKeyID、MAC_{STA1}、MAC_{STA2}、UIE_{STA1}。分组中的协议数据的集合及元素 ID 定义见表 D.37。

表 D.37 站间密钥请求分组有效元素集合 

信息元素	元素 ID	元素长度 (八位位组)	定 义
USKID	0	16	D.4.1.21
STAKeyID	1	1	D.4.1.22
MAC _{STA1}	2	6	D.4.1.8
MAC _{STA2}	3	6	D.4.1.8
UIE _{STA1}	4	可变	D.4.1.15
KN1	5	16	
保留	5~15		
保留	16~255		

其中：

- USKID: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,其定义见 D.4.1.21;
- STAKeyID: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,其定义见 D.4.1.22;
- MAC_{STA1}:该字段的值为 MAC_{STA1},其定义见 D.4.1.8;
- MAC_{STA2}:该字段的值为 MAC_{STA2},其定义见 D.4.1.8;
- UIE_{STA1}:表示第一用户终端 STA₁ 的单播密码信息元素,描述第一用户终端 STA₁ 支持的所有单播密码套件供第二用户终端 STA₂ 进行选择,其定义见 D.4.1.15;
- KN1:表示第一用户终端 STA₁ 的密钥通告标识密钥通告标识,长度为 16 个八位位组,表示一个整数,初始值为 0x5C365C365C365C365C365C365C36,在每次密钥请求时,该字段值加 1;交换设备 SW 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 实体解除关联。

D.9.2.1.3 消息鉴别码 MIC

记为 MIC1 字段,其值为 STA₁ 利用 USKID 字段标识的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法计算得到,其定义见 D.4.1.10。

D.9.2.2 处理过程

交换设备 SW 收到第一用户终端 STA₁ 发来的站间密钥请求分组后,进行如下处理:

- 检查 KN1 字段是否单调递增,若不是,则丢弃该分组;若是,则执行 b);
- 利用与第一用户终端 STA₁ 共享的单播密钥中的消息鉴别密钥 MAK 验证站间密钥请求分组中消息鉴别码 MIC1 字段的正确性,若正确,则执行 c);若不正确,则丢弃该分组;
- 生成一个随机数作为第一用户终端 STA₁ 和第二用户终端 STA₂ 之间的站间密钥 STAkey₁₋₂,构造站间密钥通告分组,将其发送给第二用户终端 STA₂,该站间密钥请求分组不再转发。

D.9.3 对对端 STA 的站间密钥通告

D.9.3.1 帧格式

D.9.3.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=1

Key_FLAG.KeyType=010(站间密钥)

Key_FLAG.Request=0

Key_FLAG.Encryption=1

Key_FLAG.MIC=1

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.9.3.1.2 协议数据

协议数据包含 USKID、MSKID、MAC_{STA1}、MAC_{STA2}、UIE_{STA1}、KN2、E₂(STAkey₁₋₂)。分组中的协议数据的集合及元素 ID 定义见表 D.38。

表 D.38 对对端 STA 的站间密钥通告分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
USKID	0	16	D.4.1.21
STAKeyID	1	1	D.4.1.22
MAC _{STA1}	2	6	D.4.1.8
MAC _{STA2}	3	6	D.4.1.8
UIE _{STA1}	4	可变	D.4.1.15
KN2	5	16	
E ₂ (STAkey ₁₋₂)	6	16	
保留	7~15		
保留	16~255		

其中:

- USKID: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,其定义见 D. 4. 1. 21;
- STAKeyID: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,其定义见 D. 4. 1. 22;
- MAC_{STA1}:该字段的值为 MAC_{STA1},其定义见 D. 4. 1. 8;
- MAC_{STA2}:该字段的值为 MAC_{STA2},其定义见 D. 4. 1. 8;
- UIE_{STA1}:表示第一用户终端 STA₁ 的单播密码信息元素,描述第一用户终端 STA₁ 支持的所有单播密码套件供第二用户终端 STA₂ 进行选择,其定义见其定义见 D. 4. 1. 15;该字段值同收到的站间密钥请求分组中的 UIE_{STA1} 字段值;
- KN2:表示第二用户终端 STA₂ 的密钥通告标识 KN2,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;交换设备 SW 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 实体解除关联;
- E₂(STakey₁₋₂):其内容字段是交换设备 SW 利用与 STA₂ 之间的密钥加密密钥 KEK 采用协商选择的单播密码算法对用户终端 STA₁ 与 STA₂ 之间的站间密钥 STakey₁₋₂ 加密后的密文(不带 MIC),站间密钥 STakey₁₋₂ 为 SW 生成的 32 个八位位组的随机数。

D. 9. 3. 1. 3 消息鉴别码 MIC

记为 MIC2,其值为交换设备 SW 利用与 STA₂ 之间的 USKID 字段标识的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法计算得到,其定义见 D. 4. 1. 10。

D. 9. 3. 2 处理过程

当第二用户终端 STA₂ 收到交换设备 SW 发送的站间密钥通告分组后,进行如下处理:

- a) 检查 KN2 字段是否单调递增,若不是,则丢弃该分组;若是,则执行步骤 b);
- b) 利用与交换设备 SW 共享的单播密钥中的消息鉴别密钥 MAK 验证消息鉴别码 MIC2 字段是否正确,若不正确,则丢弃该分组;若正确,则执行步骤 c);
- c) 利用与交换设备 SW 共享的单播密钥中的密钥加密密钥 KEK 解密 E₂(STakey₁₋₂) 字段即可得到与第一用户终端 STA₁ 之间的站间密钥 STakey₁₋₂;
- d) 根据收到的站间密钥通告分组中的 UIE_{STA1} 字段,选择一种第二用户终端 STA₂ 也支持的单播密码套件;
- e) 保存此次的密钥通告标识 KN2 字段的值,并构造站间密钥通告响应分组,发送给交换设备 SW。

D. 9. 4 对端 STA 的站间密钥通告响应

D. 9. 4. 1 帧格式

D. 9. 4. 1. 1 Key_FLAG

Key_FLAG 定义见 D. 4. 3. 5. 3。

Key_FLAG. ACK=0

Key_FLAG. KeyType= 010(站间密钥)

Key_FLAG. Request=0

Key_FLAG. Encryption=0

Key_FLAG, MIC=1

Key_FLAG, OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.9.4.1.2 协议数据

协议数据包含 USKID、MSKID、MAC_{STA1}、MAC_{STA2}、UIE_{STA2}、KN2。分组中的协议数据的集合及元素 ID 定义见表 D.39。

表 D.39 对端 STA 的站间密钥通告响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
USKID	0	16	D.4.1.21
STakeyID	1	1	D.4.1.22
MAC _{STA1}	2	6	D.4.1.8
MAC _{STA2}	3	6	D.4.1.8
UIE _{STA2}	4	可变	6.1.15
KN2	5	16	
保留	6~15		
保留	16~255		

其中：

- USKID: 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, 此字段值应与接收到的站间密钥通告分组中的 USKID 字段值相同, 其定义见 D.4.1.21;
- STakeyID: 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, 其定义见 D.4.1.22;
- MAC_{STA1}: 该字段的值为 MAC_{STA1}, 其定义见 D.4.1.8;
- MAC_{STA2}: 该字段的值为 MAC_{STA2}, 其定义见 D.4.1.8;
- UIE_{STA2}: 表示第二用户终端 STA₂ 的选择的单播密码信息元素, 即第二用户终端 STA₂ 的选择的单播密码套件信息; 该字段中单播密码套件计数为 1;
- KN2: 表示第二用户终端 STA₂ 的密钥通告标识 KN2, 长度为 16 个八位位组, 表示一个整数, 此字段值应与接收到的站间密钥通告分组中的 KN2 字段值相同;

D.9.4.1.3 消息鉴别码 MIC

记为 MIC3, 其值为 STA₂ 利用 USKID 字段标识的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法计算得到, 其定义见 D.4.1.10。

D.9.4.2 处理过程

交换设备 SW 收到第二用户终端 STA₂ 发送的站间密钥通告响应分组后进行如下处理：

- a) 比较 Key_FLAG 字段的 OperationType 字段、STakeyID 字段、USKID 字段、ADDID 字段和密钥通告标识字段 KN2 与之前发送给第二用户终端 STA₂ 的站间密钥通告分组中对应字段值是否一致, 若有一个不一致, 则丢弃该分组; 若都一致, 则执行步骤 b);
- b) 利用与第二用户终端 STA₂ 共享的单播密钥中的消息鉴别密钥 MAK 验证消息鉴别码 MIC3 字段的正确性, 若不正确, 则丢弃该分组; 若正确, 则保存此次的密钥通告标识 KN2 字段的值,

完成将第一用户终端 STA₁ 与第二用户终端 STA₂ 之间的站间密钥 STakey₁₋₂ 对第二用户终端 STA₂ 通告的过程,并执行步骤 c);

- c) 根据之前通告给第二用户终端 STA₂ 的站间密钥 STakey₁₋₂,构造站间密钥通告分组,将其发送给第一用户终端 STA₁。

D.9.5 对发起端 STA 的站间密钥通告

D.9.5.1 帧格式

D.9.5.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=1

Key_FLAG.KeyType=010(站间密钥)

Key_FLAG.Request=0

Key_FLAG.Encryption=1

Key_FLAG.MIC=1

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.9.5.1.2 协议数据

协议数据包含 USKID、MSKID、MAC_{STA1}、MAC_{STA2}、UIE_{STA2}、KN1、E₁(STakey₁₋₂)。分组中的协议数据的集合及元素 ID 定义见表 D.40。

表 D.40 对发起端 STA 的站间密钥通告分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
USKID	0	16	D.4.1.21
STakeyID	1	1	D.4.1.22
MAC _{STA1}	2	6	D.4.1.8
MAC _{STA2}	3	6	D.4.1.8
UIE _{STA2}	4	可变	6.1.15
KN1	5	16	
E ₁ (STakey ₁₋₂)	6	16	
保留	7~15		
保留	16~255		

其中:

- USKID: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,其定义见D.4.1.21;
- STakeyID: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,其定义见 D.4.1.22;
- MAC_{STA1}:该字段的值为 MAC_{STA1},其定义见 D.4.1.8;
- MAC_{STA2}:该字段的值为 MAC_{STA2},其定义见 D.4.1.8;
- UIE_{STA2}:表示第二用户终端 STA₂ 的选择的单播密码信息元素,即第二用户终端 STA₂ 的选

择的单播密码套件信息；该字段中单播密码套件计数为 1。该字段值与收到的 STA₂ 发送的站间密钥通告响应分组中的字段值 UIE_{STA2} 相同；

- KN1：表示第一用户终端 STA₁ 的密钥通告标识 KN1，长度为 16 个八位位组，表示一个整数，该字段值同收到的站间密钥请求分组中 KN1 字段的值；
- E₁ (STakey₁₋₂)：其内容字段是交换设备 SW 利用与 STA₁ 之间的密钥加密密钥 KEK 采用协商选择的单播密码算法对用户终端 STA₁ 与 STA₂ 之间的站间密钥 STakey₁₋₂ 加密后的密文（不带 MIC），站间密钥 STakey₁₋₂ 为 SW 生成的 32 个八位位组的随机数，同通告给 STA₂ 的站间密钥值。

D.9.5.1.3 消息鉴别码 MIC

记为 MIC₄，其值为交换设备 SW 利用与 STA₁ 之间的 USKID 字段标识的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法计算得到，其定义见 D.4.1.10。

D.9.5.2 处理过程

当第一用户终端 STA₁ 收到交换设备 SW 发送的站间密钥通告分组后，进行如下处理：

- a) 比较 Key_FLAG 字段的 OperationType 字段、STakeyID 字段、USKID 字段、ADDID 字段和 KN1 字段值与之前发送的站间密钥请求分组中的对应字段值是否一致，若不一致，则丢弃该分组；若一致，则执行步骤 b)；
- b) 利用与交换设备 SW 共享的单播密钥中的消息鉴别密钥 MAK 验证消息鉴别码 MIC₄ 字段是否正确，若不正确，则丢弃该分组；若正确，则执行步骤 c)；
- c) 利用与交换设备 SW 共享的单播密钥中的密钥加密密钥 KEK 解密 E₁ (STakey₁₋₂) 字段即可得到与第二用户终端 STA₂ 之间的站间密钥 STakey₁₋₂；
- d) 保存此次的密钥通告标识 KN1 字段的值以及 UIE_{STA2} 字段值，构造站间密钥通告响应分组，发送给交换设备 SW。

D.9.6 发起端 STA 的站间密钥通告响应

D.9.6.1 帧格式

D.9.6.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=0

Key_FLAG.KeyType=010(站间密钥)

Key_FLAG.Request=0

Key_FLAG.Encryption=0

Key_FLAG.MIC=1

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.9.6.1.2 协议数据

协议数据包含 USKID、MSKID、MAC_{STA1}、MAC_{STA2}、UIE_{STA2}、KN1。分组中的协议数据的集合及元素 ID 定义见表 D.41。

表 D.41 对发起端 STA 的站间密钥通告分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
USKID	0	16	D.4.1.21
STAKeyID	1	1	D.4.1.22
MAC _{STA1}	2	6	D.4.1.8
MAC _{STA2}	3	6	D.4.1.8
KN1	4	16	
保留	5~15		
保留	16~255		

其中：

- USKID: 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。此字段值应与接收到的站间密钥通告分组中的 USKID 字段值相同,其定义见 D.4.1.21;
- STAKeyID: 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,MSKID/STAKeyID 的定义见 D.4.1.22;
- MAC_{STA1}: 该字段的值为 MAC_{STA1},其定义见 D.4.1.8;
- MAC_{STA2}: 该字段的值为 MAC_{STA2},其定义见 D.4.1.8;
- KN1: 表示第一用户终端 STA₁ 的密钥通告标识 KN1,长度为 16 个八位位组,表示一个整数,该字段值同收到的站间密钥通告分组中 KN1 字段的值;

D.9.6.1.3 消息鉴别码 MIC

记为 MIC5,其值为 STA₁ 利用 USKID 字段标识的消息鉴别密钥 MAK 通过 HMAC-SHA256 算法计算得到,其定义见 D.4.1.10。

D.9.6.2 处理过程

交换设备 SW 收到第二用户终端 STA₂ 发送的站间密钥通告响应分组后进行如下处理:

- a) 比较 Key_FLAG 字段的 OperationType 字段、STAKeyID 字段、USKID 字段、ADDID 字段和密钥通告标识字段 KN1 与之前发送给第一用户终端 STA₁ 的站间密钥通告分组中对应字段值是否一致,若有一个不一致,则丢弃该分组;若都一致,则执行步骤 b);
- b) 利用与第一用户终端 STA₁ 共享的单播密钥中的消息鉴别密钥 MAK 验证消息鉴别码 MIC5 字段的正确性,若不正确,则丢弃该分组;若正确,则保存此次的密钥通告标识 KN1 字段的值,完成将第一用户终端 STA₁ 与第二用户终端 STA₂ 之间的站间密钥 STAKey₁₋₂ 对第一用户终端 STA₁ 通告的过程。

D.9.7 站间密钥建立补充说明

对上述用户终端 STA 的密钥通告标识 KN 字段的维护和使用补充解释如下:每个用户终端 STA 将维护一个密钥通告标识 KN,其取值为一个整数,初始值为一个定值,在每次发起站间密钥请求分组时会主动对该值加 1 后使用,并在每次收到正确的站间密钥通告分组后会根据其中的密钥通告标识 KN 字段的值对该值进行更新;交换设备 SW 为其下所有直连用户终端分别维护一个密钥通告标识 KN,当其需要为某用户终端主动通告站间密钥时会对该用户终端的密钥通告标识 KN 的值加 1 后使

用,并在每次收到正确的站间密钥通告响应分组后会根据其中的密钥通告标识 KN 字段的值对该值进行更新。在上述实施例中,交换设备 SW 对第二用户终端 STA₂ 的站间密钥通告过程即为主动通告过程,对第一用户终端 STA₁ 的站间密钥通告过程则为被动通告过程。在上述实施例中,第一用户终端 STA₁ 维护一个密钥通告标识 KN1,第二用户终端 STA₂ 维护一个密钥通告标识 KN2,交换设备 SW 分别为第一用户终端 STA₁ 和第二用户终端 STA₂ 维护密钥通告标识 KN1 和密钥通告标识 KN2;第一用户终端 STA₁ 对自己维护的密钥通告标识 KN1 加 1 后用于发起站间密钥请求分组,交换设备 SW 对维护的第二用户终端 STA₂ 的密钥通告标识 KN2 加 1 后用于主动向第二用户终端 STA₂ 发起站间密钥通告分组,第二用户终端 STA₂ 收到正确的站间密钥通告分组后会根据其中的密钥通告标识 KN2 字段的值对自己维护的密钥通告标识 KN2 的值进行更新,交换设备 SW 收到正确的第二用户终端 STA₂ 发送的站间密钥通告响应分组后会根据其中的密钥通告标识 KN2 字段的值对自己维护的密钥通告标识 KN2 的值进行更新,交换设备 SW 使用站间密钥请求分组中的密钥通告标识 KN1 被动地向第一用户终端 STA₁ 发起站间密钥通告分组,第一用户终端 STA₁ 收到正确的站间密钥通告分组后会根据其中的密钥通告标识 KN1 字段的值对自己维护的密钥通告标识 KN1 的值进行更新,当交换设备 SW 收到正确的第一用户终端 STA₁ 发送的站间密钥通告响应分组后会根据其中的密钥通告标识 KN1 字段的值对自己维护的密钥通告标识 KN1 的值进行更新。

交换设备 SW 根据第一用户终端 STA₁ 的请求为第一用户终端 STA₁ 和第二用户终端 STA₂ 建立站间密钥过程中,需要先完成对第二用户终端 STA₂ 的通告,再完成对第一用户终端 STA₁ 的通告。只有对第二用户终端 STA₂ 和第一用户终端 STA₁ 的通告都成功才完成整个站间密钥建立过程。

若用户终端 STA₁/STA₂ 需要更新或者撤销与用户终端 STA₂/STA₁ 之间的站间密钥,也需要构造站间密钥请求分组,发送给交换设备 SW,请求更新或撤销用户终端 STA₁ 和用户终端 STA₂ 之间的站间密钥 STAkey₁₋₂。

实际实现时,若对用户终端 STA₂ 和 STA₁ 的通告不成功,可通过重新通告机制重新发起通告。若对第二用户终端 STA₂ 的通告在达到设定的最大重新通告次数仍没有取得成功,则认为无法为用户终端 STA₁ 和 STA₂ 建立站间密钥;若对第二用户终端 STA₂ 的通告取得成功,但对第一用户终端 STA₁ 的通告在达到设定的最大重新通告次数仍没有取得成功,则认为无法为用户终端 STA₁ 和 STA₂ 建立站间密钥,此时需要通知第二用户终端 STA₂ 撤销刚建立的与第一用户终端 STA₁ 之间的站间密钥,即交换设备 SW 构造站间密钥通告分组给第二用户终端 STA₂,并且分组中需要设置撤销标识。

在用户终端 STA₁ 和 STA₂ 之间需要保密通信时,用户终端 STA₁ 和 STA₂ 均可发起站间密钥请求。根据本地策略,若站间密钥是双向的,可选择由 MAC 地址大的用户终端发起建立的站间密钥作为它们之间数据保密传输使用的密钥;若站间密钥是单向的,则用户终端 STA₁/STA₂ 发送数据包到用户终端 STA₂/STA₁ 时,使用用户终端 STA₁/STA₂ 发起的站间密钥建立过程建立的站间密钥加密数据包,用户终端 STA₁/STA₂ 接收来自用户终端 STA₂/STA₁ 的数据包时,使用用户终端 STA₂/STA₁ 发起的站间密钥建立过程建立的站间密钥解密数据包。

D.9.8 邻居用户终端站间密钥建立

D.9.8.1 过程概述

本附录中 D.9.1 所述站间密钥建立对于存在多个邻居交换设备的第一用户终端 STA₁ 和第二用户终端 STA₂,为第一用户终端 STA₁ 和第二用户终端 STA₂ 分发密钥的交换设备 SW 的选择无法通过交换路径探寻过程得到,需要先发起邻居交换设备选择过程。邻居交换设备选择过程如图 D.60。

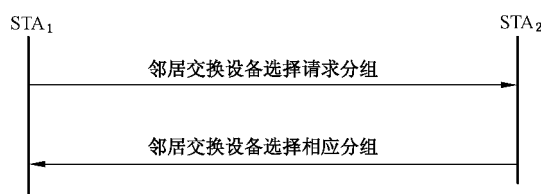


图 D.60 邻居交换设备选择过程图示

邻居交换设备选择过程主要是第一用户终端 STA₁ 和第二用户终端 STA₂ 协商选择一个均为双方邻居的交换设备,之后 STA₁ 就可以请求该邻居交换设备他们通过本附录中 D.9.1 所述站间密钥建立过程建立站间密钥。邻居交换设备选择过程主要包括两个分组:邻居交换设备选择请求分组和邻居交换设备选择响应分组。STA₁ 通过邻居交换设备选择请求分组将 STA₁ 的邻居交换设备列表信息告诉 STA₂;STA₂ 通过邻居交换设备选择响应分组将自己选择的一个同为 STA₁ 和 STA₂ 的邻居的交换设备的信息反馈给 STA₁。

邻居交换设备选择过程分组封装格式见图 D.61。

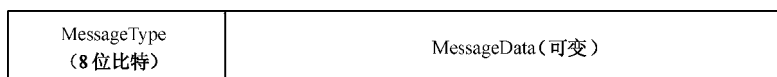


图 D.61 站间密钥建立协议分组数据封装格式

其中:

——MessageType,表示消息分组所属的分组类别,定义如表 D.42。

表 D.42 邻居交换设备选择过程分组 MessageType 子类型定义

MessageType 值	定 义
0x00	保留
0x01	邻居交换设备选择请求分组
0x02	邻居交换设备选择响应分组
其余	保留

——MessageData 数据:每种分组的消息数据 MessageData 将在协议分组介绍中进行介绍。其数据元素封装格式见 D.4.2。

D.9.8.2 邻居交换设备选择请求

当用户终端 STA₁ 试图与自己的邻居用户终端 STA₂ 发起保密通信时,STA₁ 首先发送邻居交换设备选择请求分组给 STA₂,将自己的邻居交换设备列表信息告知 STA₂,供其选择一个交换设备建立 STA₁ 与 STA₂ 之间的站间密钥。该邻居交换设备选择请求分组包括 NIE_{STA1} 字段,描述 STA₁ 的邻居交换设备列表信息供 STA₂ 进行选择,该分组封装在 TAEP-Request/ Neighbor SW Negotiation 中。

邻居交换设备选择请求分组中,数据元素都采用 D.4.3.4.1 介绍的封装格式进行封装。分组中有效数据信息元素的集合及元素 ID 定义见表 D.43。

表 D.43 邻居交换设备选择请求分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
STA ₁ 的邻居交换设备信息 NIE _{STA1}	0	可变	D.4.1.16
保留	1~255	—	—

其中:

STA₁ 的邻居交换设备信息 NIE_{STA1}:表示 STA₁ 的所有邻居交换设备信息。

D.9.8.3 邻居交换设备选择响应

当用户终端 STA₂ 收到邻居用户终端 STA₁ 发来的邻居交换设备选择请求分组时,根据邻居交换设备选择请求分组中的 NIE_{STA1} 元素以及自己的邻居节点信息,选择一个双发共同的邻居交换设备,构造邻居交换设备选择响应分组发送给邻居用户终端 STA₁。该邻居交换设备选择响应分组封装在 TAEP-Response/ Neighbor SW Negotiation 中。

邻居交换设备选择响应分组中,数据元素都采用 D.4.3.4.1 介绍的封装格式进行封装。分组中有效数据信息元素的集合及元素 ID 定义见表 D.44。

表 D.44 邻居交换设备选择响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
STA ₂ 的邻居交换设备信息 NIE _{STA2}	0	可变	D.4.1.16
保留	1~255	—	—

其中:

STA₂ 的邻居交换设备信息 NIE_{STA2} 元素:表示 STA₂ 选择的与 STA₁ 共同的邻居交换设备信息;该字段中邻居交换设备计数字段为 1。

当 STA₁ 收到 STA₂ 回应的邻居交换设备选择响应分组是,判断 NIE_{STA2} 字段是否有效,即 STA₁ 是否有这样的一个邻居交换设备,若无效,则丢弃该分组;否则,根据 STA₂ 选择的邻居交换设备的信息,向该邻居交换设备发起本附录中 D.9.1 所述的站间密钥建立过程,即可建立 STA₁ 与 STA₂ 之间共享的站间密钥。

考虑到两个邻居用户终端的邻居交换设备集合都是一致的,可由发起端 STA 直接选举一个邻居交换设备负责建立两个邻居用户终端的站间密钥,该直接选举可由发起端 STA 选择 MAC 地址最小的邻居交换设备,直接向该交换设备发送站间密钥请求分组。因此,邻居交换设备过程的实现是可选的。

D.10 交换密钥建立

D.10.1 交换密钥建立过程概述

交换密钥 SWKey 包含交换单播加密密钥 SW-UEK(Switch Unicast Encryption Key),交换消息鉴别密钥 SW-MAK(Switch Message Authentication Key),交换密钥加密密钥 SW-KEK(Switch Key Encryption Key)。交换密钥的四个组成部分与单播密钥的四个组成部分一一对应。其中交换单播加密密钥 SW-UEK 用于保护交换设备之间用户数据的机密性和完整性,交换消息鉴别密钥 SW-MAK 用于

保护交换设备之间协议数据的完整性,交换密钥加密密钥 SW-KEK 用于保护设备之间协议数据中密钥数据的机密性。

若交换设备 SW₁ 和 SW₂ 相邻,则它们之间的交换密钥就是它们之间的单播密钥;若交换设备 SW₁ 和 SW₂ 不相邻,则它们之间交换密钥的建立分为两个过程:交换基密钥通告过程和交换密钥协商过程。对于不相邻的交换设备 SW₁ 和 SW₂,若它们之间已存在通过预分发的交换基密钥 SWBK(Switch Basic Key),则只需执行交换密钥协商过程即可完成交换密钥的建立;若它们之间没有预先设置的交换基密钥,则首先需要执行交换基密钥通告过程,再执行交换密钥协商过程,才能完成交换密钥的建立。

由于相邻的交换设备之间的单播密钥就是它们之间的交换密钥,当相邻的交换设备通过预分发或其他安全机制建立了共享的单播密钥时,也就建立了它们之间的交换密钥。初始的网络中可以只有一个或者两个交换设备,之后逐步扩展。因此,当交换设备 SW₁ 通过当前网络中的交换设备 SW_M 接入当前网络时,交换设备 SW₁ 和 SW_M 已建立了单播密钥,即建立了它们之间的交换密钥,同时当前网络中其他所有不相邻的交换设备两两之间也已建立有交换密钥。此时,通过交换设备 SW_M 就可以建立交换设备 SW₁ 和当前网络中其他任意的交换设备(如 SW₂)之间的交换密钥。

D. 10.2 交换基密钥建立

D. 10.2.1 交换基密钥建立过程概述

交换基密钥通告过程为网络中不相邻的交换设备 SW₁ 和交换设备 SW₂ 之间建立交换基密钥,将此密钥应用于交换密钥协商过程,以建立起交换设备 SW₁ 与交换设备 SW₂ 之间共享的交换密钥。

参见图 D. 62,交换基密钥通告过程是由交换设备 SW_M 生成一个随机数,作为交换设备 SW₂ 和 SW₁ 的交换基密钥,并先后将此交换基密钥通告给交换设备 SW₂ 和 SW₁。该过程共包含四个步骤:交换设备 SW_M 对 SW₂ 的交换基密钥通告、交换设备 SW₂ 的交换基密钥通告响应、交换设备 SW_M 对 SW₁ 的交换基密钥通告以及交换设备 SW₁ 的交换基密钥通告响应。其中交换设备 SW_M 对 SW₁ 的交换基密钥通告及交换设备 SW₁ 的交换基密钥通告响应与交换设备 SW_M 对 SW₂ 的交换基密钥通告及交换设备 SW₂ 的交换基密钥通告响应类似,只是通告中的交换基密钥所使用的交换密钥加密密钥 SW-KEK 不同。

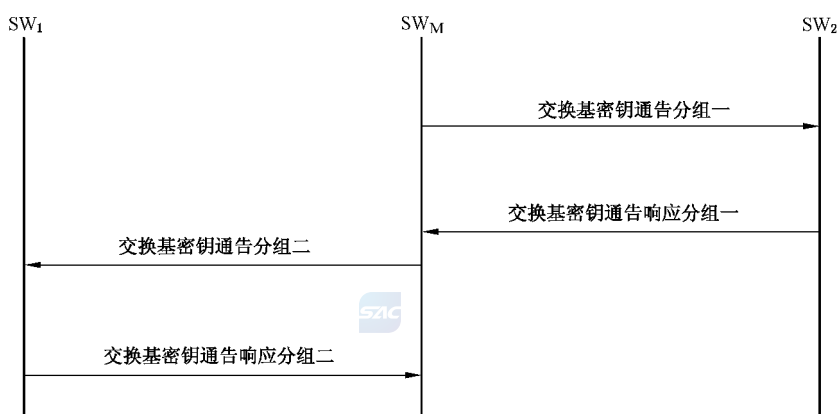


图 D. 62 交换基密钥建立过程

交换基密钥通告过程分组使用 TAEPoL-Key 帧封装,其协议数据字段中 Key Descriptor 类型取值为 0x14,其协议数据字段中的数据字段封装格式见图 D. 63。

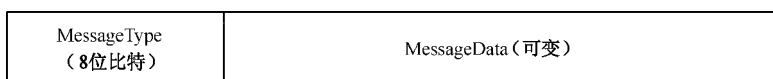


图 D.63 交换基密钥通告过程分组数据封装格式

其中:

——Message Type, 表示消息分组所属的分组类别, 定义如表 D.45。

表 D.45 交换基密钥通告过程分组 Message Type 子类型定义

Message Type 值	定 义
0x00	保留
0x01	交换基密钥通告分组一
0x02	交换基密钥通告响应分组一
0x03	交换基密钥通告分组二
0x04	交换基密钥通告响应分组二
其余	保留

——MessageData 数据: 每种分组的消息数据 MessageData 将在协议分组介绍中进行介绍。其数据元素封装格式 D.4.2。

D.10.2.2 交换基密钥通告分组一

D.10.2.2.1 帧格式

D.10.2.2.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=1

Key_FLAG.KeyType=011(交换基密钥)

Key_FLAG.Request=0

Key_FLAG.Encryption=1

Key_FLAG.MIC=1

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.10.2.2.1.2 协议数据

协议数据包含 SWKeyID、SWBKID、MAC_{SW1}、MAC_{SW2}、KN2、E₂(STakey₁₋₂)。分组中的协议数据的集合及元素 ID 定义见表 D.46。

表 D.46 交换基密钥通告分组一有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
SWKeyID	0	16	D.4.1.21
SWBKID	1	1	D.4.1.22

表 D. 46 (续)

信息元素	元素 ID	元素长度 (八位位组)	定 义
MAC _{SW1}	2	6	D. 4. 1. 8
MAC _{SW2}	3	6	D. 4. 1. 8
KN2	4	16	
E ₂ (STakey ₁₋₂)	5	16	
保留	6~15		
保留	16~255		

其中:

- SWKeyID: 其中比特 0 标识计算消息鉴别码字段值所用的交换消息鉴别密钥 SW-MAK_{2-M}, 其定义见 D. 4. 1. 21;
- SWBKID, 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, 其定义见 D. 4. 1. 20;
- MAC_{SW1}: 该字段的值为 MAC_{SW1}, 其定义见 D. 4. 1. 8;
- MAC_{SW2}: 该字段的值为 MAC_{SW2}, 其定义见 D. 4. 1. 8;
- KN2: 表示交换设备 SW₂ 的密钥通告标识 KN2, 长度为 16 个八位位组, 表示一个整数, 初始值为 0x5C365C365C365C365C365C365C36, 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;
- E₂ (SWBK₁₋₂): 其内容字段是交换设备 SW_M 利用与 SW₂ 之间的交换密钥加密密钥 SW-KEK_{2-M} 采用协商的单播播密码算法对交换基密钥 SWBK₁₋₂ 加密后的密文 (不带 MIC), 交换基密钥 SWBK₁₋₂ 为交换设备 SW_M 生成的 16 个八位位组的随机数。

D. 10. 2. 2. 1. 3 消息鉴别码 MIC

记为 MIC1, 其值为交换设备 SW_M 利用与交换设备 SW₂ 之间的 SWKeyID 字段标识的交换消息鉴别密钥 SW-MAK_{2-M} 通过 HMAC-SHA256 算法计算得到, 其定义见 D. 4. 1. 10。

D. 10. 2. 2. 2 处理过程

当交换设备 SW₂ 收到交换设备 SW_M 发送的交换基密钥通告分组后, 进行如下处理:

- a) 检查 KN2 字段是否单调递增, 若不是, 则丢弃该分组; 否则, 执行步骤 b);
- b) 利用与交换设备 SW_M 之间的交换消息鉴别密钥 SW-MAK_{2-M} 验证消息鉴别码 MIC1 字段是否正确, 若不正确, 则丢弃该分组; 若正确, 执行步骤 c);
- c) 利用与交换设备 SW_M 之间的交换密钥加密密钥 SW-KEK_{2-M} 解密 E₂ 字段即可得到与交换设备 SW₁ 之间的交换基密钥 SWBK₁₋₂;
- d) 保存此次的密钥通告标识 KN2 字段的值, 并构造交换基密钥通告响应分组, 发送给交换设备 SW_M。

D. 10. 2. 3 交换基密钥通告响应分组一

D. 10. 2. 3. 1 帧格式

D. 10. 2. 3. 1. 1 Key_FLAG

Key_FLAG 定义见 D. 4. 3. 5. 3。

Key_FLAG.ACK=0
 Key_FLAG.KeyType=011(交换基密钥)
 Key_FLAG.Request=0
 Key_FLAG.Encryption=0
 Key_FLAG.MIC=1
 Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D. 10.2.3.1.2 协议数据

协议数据包含 SWKeyID、SWBKID、MAC_{SW1}、MAC_{SW2}、KN2。分组中的协议数据的集合及元素 ID 定义见表 D. 47。

表 D. 47 对端 STA 的站间密钥通告响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
SWKeyID	0	16	D. 4. 1. 21
SWBKID	1	1	D. 4. 1. 22
MAC _{SW1}	2	6	D. 4. 1. 8
MAC _{SW2}	3	6	D. 4. 1. 8
KN2	4	16	
保留	5~15		
保留	16~255		

其中：

- SWKeyID: 其中比特 0 标识计算消息鉴别码字段值所用的交换消息鉴别密钥 SW-MAK_{2-M}；此字段值应与接收到的交换基密钥通告分组中的 SWKeyID 字段值相同，其定义见 D. 4. 1. 21；
- SWBKID: 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，SWBKID 的定义见 D. 4. 1. 20；
- MAC_{SW1}: 该字段的值为 MAC_{SW1}，其定义见 D. 4. 1. 8；
- MAC_{SW2}: 该字段的值为 MAC_{SW2}，其定义见 D. 4. 1. 8；
- KN2: 表示交换设备 SW₂ 的密钥通告标识 KN2，长度为 16 个八位位组，表示一个整数，该字段值同接收到的交换基密钥通告分组中的 KN2 字段。

D. 10.2.3.1.3 消息鉴别码 MIC

记为 MIC2，其值为交换设备 SW₂ 利用 SWKeyID 字段标识的交换消息鉴别密钥 SW-MAK_{2-M} 通过 HMAC-SHA256 算法计算得到，其定义见 D. 4. 1. 10。

D. 10.2.3.2 处理过程

交换设备 SW_M 收到交换设备 SW₂ 发送的交换基密钥通告响应分组后，进行如下处理：

- a) 比较 Key_FLAG 字段的 OperationType 字段、SWKeyID 字段、SWBKID 字段、MAC_{SW1} 字段、MAC_{SW2} 字段和 KN2 字段与之前发送给交换设备 SW₂ 的交换基密钥通告分组中的对应字段值是否一致，若有一个不一致，则丢弃该分组；若全部一致，则执行步骤 b)；

- b) 利用与交换设备 SW₂ 之间的交换消息鉴别密钥 SW-MAK_{2-M} 验证消息鉴别码 MIC2 字段的正确性,若不正确,则丢弃该分组;若正确,则保存此次的密钥通告标识 KN2 字段的值,完成将交换设备 SW₁ 与交换设备 SW₂ 之间的交换基密钥 SWBK₁₋₂ 对交换设备 SW₂ 通告的过程,执行步骤 c);
- c) 交换设备 SW_M 根据之前通告给交换设备 SW₂ 的交换基密钥 SWBK₁₋₂,构造交换基密钥通告分组,发送给交换设备 SW₁。

D. 10. 2. 4 交换基密钥通告分组二

D. 10. 2. 4. 1 帧格式

对 SW₁ 的交换密钥通告分组使用 TAEPoL-Key 帧封装,各字段的定义如下。

D. 10. 2. 4. 1. 1 Key_FLAG

Key_FLAG 定义见 D. 4. 3. 5. 3。

Key_FLAG. ACK=1

Key_FLAG. KeyType=011(交换基密钥)

Key_FLAG. Request=0

Key_FLAG. Encryption=1

Key_FLAG. MIC=1

Key_FLAG. OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D. 10. 2. 4. 1. 2 协议数据

协议数据包含 SWKeyID、SWBKID、MAC_{SW1}、MAC_{SW2}、KN1、E₁(STakey₁₋₂)。分组中的协议数据的集合及元素 ID 定义见表 D. 48。

表 D. 48 交换基密钥通告分组一有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
SWKeyID	0	16	D. 4. 1. 21
SWBKID	1	1	D. 4. 1. 22
MAC _{SW1}	2	6	D. 4. 1. 8
MAC _{SW2}	3	6	D. 4. 1. 8
KN1	4	16	
E ₁ (STakey ₁₋₂)	5	16	
保留	6~15		
保留	16~255		

其中:

- SWKeyID:其中比特 0 标识计算消息鉴别码字段值所用的交换消息鉴别密钥 SW-MAK_{1-M},其定义见 D. 4. 1. 21;
- SWBKID,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,其定义见 D. 4. 1. 20;

- MAC_{SW_1} : 该字段的值为 MAC_{SW_1} , 其定义见 D. 4. 1. 8;
- MAC_{SW_2} : 该字段的值为 MAC_{SW_2} , 其定义见 D. 4. 1. 8;
- KN1: 表示交换设备 SW_1 的密钥通告标识 KN1, 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;
- $E_1(SWBK_{1-2})$: 其内容字段是交换设备 SW_M 利用与 SW_1 之间的交换密钥加密密钥 $SW-KEK_{1-M}$ 采用协商的单播播密码算法对交换基密钥 $SWBK_{1-2}$ 加密后的密文 (不带 MIC), 交换基密钥 $SWBK_{1-2}$ 为交换设备 SW_M 生成的 16 个八位位组的随机数。

D. 10. 2. 4. 1. 3 消息鉴别码 MIC

记为 MIC3, 其值为交换设备 SW_M 利用与交换设备 SW_1 之间的 SWKeyID 字段标识的交换消息鉴别密钥 $SW-MAK_{1-M}$ 通过 HMAC-SHA256 算法计算得到, 其定义见 D. 4. 1. 10。

D. 10. 2. 4. 2 处理过程

当交换设备 SW_1 收到交换设备 SW_M 发送的交换基密钥通告分组后, 进行如下处理:

- a) 检查 KN1 字段是否单调递增, 若不是, 则丢弃该分组; 否则, 执行步骤 b);
- b) 利用与交换设备 SW_M 之间的交换消息鉴别密钥 $SW-MAK_{1-M}$ 验证消息鉴别码 MIC3 字段是否正确, 若不正确, 则丢弃该分组; 若正确, 执行步骤 c);
- c) 利用与交换设备 SW_M 之间的交换密钥加密密钥 $SW-KEK_{1-M}$ 解密 E_2 字段即可得到与交换设备 SW_1 之间的交换基密钥 $SWBK_{1-2}$;
- d) 保存此次的密钥通告标识 KN1 字段的值, 并构造交换基密钥通告响应分组, 发送给交换设备 SW_M 。

D. 10. 2. 5 交换基密钥通告响应分组二

D. 10. 2. 5. 1 帧格式

D. 10. 2. 5. 1. 1 Key_FLAG

Key_FLAG 定义见 D. 4. 3. 5. 3。

Key_FLAG. ACK=0

Key_FLAG. KeyType=011(交换基密钥)

Key_FLAG. Request=0

Key_FLAG. Encryption=0

Key_FLAG. MIC=1

Key_FLAG. OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D. 10. 2. 5. 1. 2 协议数据

协议数据包含 SWKeyID、SWBKID、 MAC_{SW_1} 、 MAC_{SW_2} 、KN1。分组中的协议数据的集合及元素 ID 定义见表 D. 49。

表 D.49 对端 STA 的站间密钥通告响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
SWKeyID	0	16	D.4.1.21
SWBKID	1	1	D.4.1.22
MAC _{SW1}	2	6	D.4.1.8
MAC _{SW2}	3	6	D.4.1.8
KN1	4	16	
保留	5~15		
保留	16~255		

其中：

- SWKeyID: 其中比特 0 标识计算消息鉴别码字段值所用的交换消息鉴别密钥 SW-MAK_{1-M}; 此字段值应与接收到的交换基密钥通告分组中的 SWKeyID 字段值相同, 其定义见 D.4.1.21;
- SWBKID: 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, SWBKID 的定义见 D.4.1.20;
- MAC_{SW1}: 该字段的值为 MAC_{SW1}, 其定义见 D.4.1.8;
- MAC_{SW2}: 该字段的值为 MAC_{SW2}, 其定义见 D.4.1.8;
- KN1: 表示交换设备 SW₁ 的密钥通告标识 KN1, 长度为 16 个八位位组, 表示一个整数, 该字段值同接收到的交换基密钥通告分组中的 KN1 字段。

D.10.2.5.1.3 消息鉴别码 MIC

记为 MIC₄, 其值为 SW₁ 利用 SWKeyID 字段标识的交换消息鉴别密钥 SW-MAK_{1-M} 通过 HMAC-SHA256 算法计算得到, 其定义见 D.4.1.10。

D.10.2.5.2 处理过程

交换设备 SW_M 收到交换设备 SW₁ 发送的交换基密钥通告响应分组后, 进行如下处理:

- a) 比较 Key_FLAG 字段的 OperationType 字段、SWKeyID 字段、SWBKID 字段、MAC_{SW1} 字段、MAC_{SW2} 字段和 KN1 字段与之前发送给交换设备 SW₁ 的交换基密钥通告分组中的对应字段值是否一致, 如果有一个不一致, 则丢弃该分组; 如果都一直, 则执行步骤 b);
- b) 利用与交换设备 SW₁ 之间的交换消息鉴别密钥 SW-MAK_{1-M} 验证消息鉴别码 MIC₄ 字段的正确性, 若正确, 则保存此次的密钥通告标识 KN1 字段的值, 完成将交换设备 SW₁ 与交换设备 SW₂ 之间的交换基密钥 SWBK₁₋₂ 对交换设备 SW₁ 通告的过程, 即完成为交换设备 SW₁ 和交换设备 SW₂ 交换基密钥的建立过程; 若不正确, 则丢弃该分组。

在具体实现时, 对交换设备 SW₂ 和 SW₁ 的通告不成功时, 可通过重新通告机制重新发起通告。交换设备 SW₁ 通过交换设备 SW_M 接入网络, 若交换设备 SW_M 对交换设备 SW₂ 的通告在达到设定的最大重新通告次数仍没有取得成功, 则认为无法为交换设备 SW₁ 和 SW₂ 建立一致的交换基密钥, 协议终止; 若对交换设备 SW₂ 的通告取得成功, 但对交换设备 SW₁ 的通告在达到设定的最大重新通告次数仍没有取得成功, 则认为无法为交换设备 SW₁ 和 SW₂ 建立一致的交换基密钥, 此时需要通知交换设备 SW₂ 撤销刚建立的与交换设备 SW₁ 之间的交换基密钥, 即交换设备 SW_M 构造交换基密钥通告分组给交换设备 SW₂, 通知交换设备 SW₂ 将已建立的与交换设备 SW₁ 之间的交换基密钥删除。

若交换设备 SW-M 需要更新或者撤销交换设备 SW₁ 与 SW₂ 之间的交换基密钥,也可以构造交换基密钥通告分组发送给交换设备 SW₂/SW₁,要求交换设备 SW₂/SW₁ 更新或者删除与交换设备 SW₁/SW₂ 之间的交换基密钥。交换基密钥的更新或撤销过程和交换基密钥的建立过程相同,在具体实现时,可通过在上述交换基密钥通告过程中的每个分组中增加一个标识字段进行区分,用于标识通过交换设备 SW 完成交换设备 SW₁ 和 SW₂ 之间交换基密钥的建立、撤销或者更新过程。

D. 10.3 交换密钥协商过程

D. 10.3.1 交换密钥协商过程概述

交换密钥协商过程是交换设备 SW₁ 和 SW₂ 利用它们之间的交换基密钥 SWBK₁₋₂ 来协商共享的交换密钥(交换单播加密密钥 SW-UEK₁₋₂,交换消息鉴别密钥 SW-MAK₁₋₂,交换密钥加密密钥 SW-KEK₁₋₂)。参见图 D. 64,交换密钥协商过程包含四个分组:交换密钥协商激活分组、交换密钥协商请求分组、交换密钥协商响应分组及交换密钥协商确认分组,其中交换密钥协商确认分组是可选的,即在具体实现过程中,交换设备 SW₂ 向交换设备 SW₁ 可发送交换密钥协商确认分组,也可不发送交换密钥协商确认分组。



图 D. 64 交换密钥协商过程

交换密钥协商过程分组使用 TAEPoL-Key 帧封装,其协议数据字段中 Key Descriptor 类型取值为 0x15,其协议数据字段中的数据字段封装格式见图 D. 65。

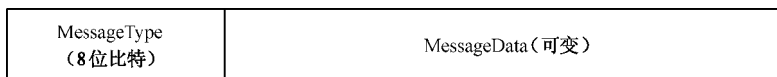


图 D. 65 交换密钥协商过程分组数据封装格式

其中:

——Message Type,表示消息分组所属的分组类别,定义如表 D. 50。

表 D.50 交换密钥协商过程分组 MessageType 子类型定义

MessageType 值	定 义
0x00	保留
0x01	交换密钥激活分组
0x02	交换密钥请求分组
0x03	交换密钥响应分组
0x04	交换密钥确认分组
其余	保留

——MessageData 数据:每种分组的消息数据 MessageData 将在协议分组介绍中进行介绍。其数据元素封装格式 D.4.2。

D.10.3.2 交换密钥协商激活

D.10.3.2.1 帧格式

D.10.3.2.1.1 Key_FLAG

Key_FLAG 定义见 D.4.3.5.3。

Key_FLAG.ACK=1

Key_FLAG.KeyType=100(交换密钥)

Key_FLAG.Request=1

Key_FLAG.Encryption=0

Key_FLAG.MIC=0

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.10.3.2.1.2 协议数据

协议数据包含 SWBKID、SWKeyID、MAC_{SW1}、MAC_{SW2}、N_{SW1}。分组中的协议数据的集合及元素 ID 定义见表 D.51。

表 D.51 交换密钥协商激活分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
SWBKID	0	16	D.4.1.20
SWKeyID	1	1	D.4.1.21
MAC _{SW1}	2	6	D.4.1.8
MAC _{SW2}	3	6	D.4.1.8
N _{SW1}	4	32	D.4.1.3
保留	5~15		
保留	16~255		

其中：

- SWBKID:标识当前的交换基密钥,其定义见 D. 4. 1. 20;
- SWKeyID:其中比特 0 标识当前协商的交换密钥,其他位保留;本字段的比特 0 如果是利用交换基密钥第一次进行交换密钥协商,其值为 0;之后重新进行交换密钥更新时该位在 0 和 1 之间翻转,SWKeyID 的定义见 D. 4. 1. 21;
- MAC_{SW1}:该字段的值为 MAC_{SW1},其定义见 D. 4. 1. 8;
- MAC_{SW2}:该字段的值为 MAC_{SW2},其定义见 D. 4. 1. 8;
- N_{SW1}:表示身份鉴别及单播密钥协商标识 N_{SW1},其定义见 D. 4. 1. 3;若标识 Key_FLAG 中 OperationType 字段值为 00,则该字段值为交换设备 SW₁ 产生的随机数;若标识 Key_FLAG 中 OperationType 字段值为 01 或者 10,则该字段值为上一次交换密钥协商过程计算的交换密钥协商标识的值。

D. 10. 3. 2. 2 处理过程

当交换设备 SW₂ 收到交换设备 SW₁ 发送的交换密钥协商激活分组后,进行如下处理:

- a) 首先检查标识 Key_FLAG 中 OperationType 字段值,若为 01 或者 10,则检查 SWBKID 所指的 SWBKSA 以及 SWKeyID 所指的 SWKeySA 是否有效,若其中一个无效,则丢弃该分组,否则执行 b);如果标识 Key_FLAG 中 OperationType 字段值为 00,则直接执行 c);
- b) 检查分组中的交换密钥协商标识 N_{SW1} 字段值与上一次交换密钥协商过程中计算的交换密钥协商标识值是否一致,若不一致,则丢弃该分组,若一致,则进一步查看标识 Key_FLAG 中 OperationType 字段值,若为 01 则为更新过程,执行步骤 c);若为 10 则为删除过程,则构造交换密钥协商响应分组发往 SW₁,并删除与 SW₁ 之间的交换密钥,交换密钥协商激活分组接收处理完成;
- c) 生成询问 N_{SW2},利用与交换设备 SW₁ 之间的交换基密钥 SWBK₁₋₂、交换密钥协商标识 N_{SW1} 及交换设备 SW₂ 生成的询问 N_{SW2} 计算得到与交换设备 SW₁ 之间的交换密钥(包括交换单播加密密钥 SW-UEK₁₋₂,交换消息鉴别密钥 SW-MAK₁₋₂,交换密钥加密密钥 SW-KEK₁₋₂)以及下一次交换密钥协商过程中的交换密钥协商标识种子,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交换密钥协商过程的协商标识并保存;
- d) 利用计算得到的交换密钥中的交换消息鉴别密钥 SW-MAK₁₋₂本地计算消息鉴别码消息鉴别码 MIC5,构造交换密钥协商请求分组发送给交换设备 SW₁。

D. 10. 3. 3 交换密钥协商请求

D. 10. 3. 3. 1 帧格式

D. 10. 3. 3. 1. 1 Key_FLAG

Key_FLAG 定义见 D. 4. 3. 5. 3。

Key_FLAG. ACK=1

Key_FLAG. KeyType=100(交换密钥)

Key_FLAG. Request=1

Key_FLAG. Encryption=0

Key_FLAG. MIC=1

Key_FLAG. OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D. 10. 3. 3. 1. 2 协议数据

协议数据包含 SWBKID、SWKeyID、MAC_{SW1}、MAC_{SW2}、N_{SW1}、UIE_{SW2}。分组中的协议数据的集合



及元素 ID 定义见表 D. 52。

表 D. 52 交换密钥协商请求分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
SWBKID	0	16	D. 4. 1. 20
SWKeyID	1	1	D. 4. 1. 21
MAC _{SW1}	2	6	D. 4. 1. 8
MAC _{SW2}	3	6	D. 4. 1. 8
N _{SW1}	4	32	D. 4. 1. 3
N _{SW2}	5	32	D. 4. 1. 3
UIE _{SW2}	6	可变	D. 4. 1. 15
保留	7~15		
保留	16~255		

其中：

- SWBKID: 标识当前的交换基密钥, 其定义见 D. 4. 1. 20; 该字段值同收到的交换密钥协商激活分组中的 SWBKID 字段值;
- SWKeyID: 其中比特 0 标识当前协商的交换密钥, 其他位保留; 本字段的比特 0 如果是利用交换基密钥第一次进行交换密钥协商, 其值为 0; 之后重新进行交换密钥更新时该位在 0 和 1 之间翻转, SWKeyID 的定义见 D. 4. 1. 21; 该字段值同收到的交换密钥协商激活分组中的 SWKeyID 字段值;
- MAC_{SW1}: 该字段的值为 MAC_{SW1}, 其定义见 D. 4. 1. 8;
- MAC_{SW2}: 该字段的值为 MAC_{SW2}, 其定义见 D. 4. 1. 8;
- N_{SW1}: 表示身份鉴别及单播密钥协商标识 N_{SW1}, 其定义见 D. 4. 1. 3; 若标识 Key_FLAG 中 OperationType 字段值为 00, 则该字段值为交换设备 SW₁ 产生的随机数; 若标识 Key_FLAG 中 OperationType 字段值为 01 或者 10, 则该字段值为上一次交换密钥协商过程计算的交换密钥协商标识的值;
- N_{SW2}: 表示交换设备 SW₂ 的询问 N_{SW2} 字段, 由交换设备 SW₂ 生成的随机数, 其定义见 D. 4. 1. 3;
- UIE_{SW2}: 表示交换设备 SW₂ 的单播密码信息元素, 描述交换设备 SW₂ 支持的所有单播密码套件供 SW₁ 进行选择, 其定义见 D. 4. 1. 15。

D. 10. 3. 3. 1. 3 消息鉴别码 MIC

记为 MIC5 字段, 其值为交换设备 SW₂ 利用最新协商的交换消息鉴别密钥 SW-MAK_{1,2} 通过 HMAC-SHA256 算法计算得到。

D. 10. 3. 3. 2 处理过程

当交换设备 SW₁ 收到交换设备 SW₂ 发送的交换密钥协商请求分组后, 进行如下处理:

- a) 比较 Key_FLAG 字段的 OperationType 字段、SWBKID 字段、SWKeyID 字段、MAC_{SW1} 字段、MAC_{SW2} 字段和交换密钥协商标识 N_{SW1} 与之前发送的交换密钥协商激活分组中的对应字段值是否一致, 若有一个不一致, 则丢弃该分组; 若都一致, 则执行步骤 b);

- b) 利用与交换设备 SW_2 之间的交换基密钥 $SWBK_{1-2}$ 、交换密钥协商标识 N_{SW1} 及交换设备 SW_2 的询问 N_{SW2} 计算得到与交换设备 SW_2 之间的交换密钥,包括交换单播加密密钥 $SW-UEK_{1-2}$, 交换消息鉴别密钥 $SW-MAK_{1-2}$, 交换密钥加密密钥 $SW-KEK_{1-2}$ 和下一次交换密钥协商过程中的交换密钥协商标识种子,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交换密钥协商过程的协商标识并保存;
- c) 利用计算得到的交换密钥中的交换消息鉴别密钥 $SW-MAK_{1-2}$ 验证交换密钥协商请求分组中的消息鉴别码 MIC5 字段的正确性,若不正确,则丢弃该分组;若正确,执行步骤 d);
- d) 利用计算得到的交换密钥中的交换消息鉴别密钥 $SW-MAK_{1-2}$ 本地计算消息鉴别码消息鉴别码 MIC6,构造交换密钥协商响应分组,发送给交换设备 SW_2 。

D. 10. 3. 4 交换密钥协商响应

D. 10. 3. 4. 1 帧格式

D. 10. 3. 4. 1. 1 Key_FLAG

Key_FLAG 定义见 D. 4. 3. 5. 3。

Key_FLAG. ACK=0

Key_FLAG. KeyType=100(交换密钥)

Key_FLAG. Request=1

Key_FLAG. Encryption=0

Key_FLAG. MIC=1

Key_FLAG. OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D. 10. 3. 4. 1. 2 协议数据

协议数据包含 $SWBKID$ 、 $SWKeyID$ 、 MAC_{SW1} 、 MAC_{SW2} 、 N_{SW1} 、 UIE_{SW2} 。分组中的协议数据的集合及元素 ID 定义见表 D. 53。

表 D. 53 交换密钥协商响应分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
SWBKID	0	16	D. 4. 1. 20
SWKeyID	1	1	D. 4. 1. 21
MAC_{SW1}	2	6	D. 4. 1. 8
MAC_{SW2}	3	6	D. 4. 1. 8
N_{SW2}	4	32	D. 4. 1. 3
UIE_{SW1}	5	可变	D. 4. 1. 15
保留	6~15		
保留	16~255		

其中:

——SWBKID:标识当前的交换基密钥,其定义见 D. 4. 1. 20;该字段值同收到的交换密钥协商请求分组中的 SWBKID 字段值;

——SWKeyID:其中比特 0 标识当前协商的交换密钥,其他位保留;本字段的比特 0 如果是利用交

换基密钥第一次进行交换密钥协商,其值为0;之后重新进行交换密钥更新时该位在0和1之间翻转,SWKeyID的定义见D.4.1.21;该字段值同收到的交换密钥协商请求分组中的SWKeyID字段值;

- MAC_{SW1}:该字段的值为MAC_{SW1},其定义见D.4.1.8;
- MAC_{SW2}:该字段的值为MAC_{SW2},其定义见D.4.1.8;
- N_{SW2}:表示交换设备SW₂的询问N_{SW2}字段,由交换设备SW₂生成的随机数,该字段值同交换密钥协商请求分组中的N_{SW2}字段值,其定义见D.4.1.3;
- UIE_{SW1}:表示交换设备N_{SW1}所选择的单播密码套件,其定义见D.4.1.15。

D.10.3.4.1.3 消息鉴别码 MIC

记为MIC6字段,表示消息鉴别码,由交换设备SW₁利用生成的交换消息鉴别密钥SW-MAK₁₋₂对交换密钥协商响应分组中本字段外的其他字段或对交换密钥协商响应分组中本字段外的其他字段及已计算出的下一次交换密钥协商过程中的交换密钥协商标识N_{SW1}通过杂凑函数计算得到的杂凑值。

D.10.3.4.2 处理过程

交换设备SW₂收到交换设备SW₁发送的交换密钥协商响应分组后,进行如下处理:

- a) 比较Key_FLAG字段的OperationType字段、SWBKID字段、SWKeyID字段、MAC_{SW1}字段、MAC_{SW2}字段和交换设备SW₂的询问N_{SW2}字段与之前发送的交换密钥协商请求分组中的对应字段值是否一致,若有一个不一致,则丢弃该分组;若都一致,则执行步骤b);
- b) 利用计算得到的交换密钥中的交换消息鉴别密钥SW-MAK₁₋₂验证交换密钥协商响应分组中的消息鉴别码MIC6字段的正确性,若不正确,则丢弃该分组;否则,交换设备SW₂确认交换设备SW₁已获得与其一致的交换密钥;
- c) 交换设备SW₂根据本地策略,选择是否发送交换密钥协商确认分组,如果选择发送交换密钥协商确认分组,则利用计算得到的交换密钥中的交换消息鉴别密钥SW-MAK₁₋₂本地计算消息鉴别码MIC7,构造交换密钥协商确认分组,发送给交换设备SW₁。

D.10.3.5 交换密钥协商确认

D.10.3.5.1 帧格式

D.10.3.5.1.1 Key_FLAG

Key_FLAG定义见D.4.3.5.3。

Key_FLAG.ACK=0

Key_FLAG.KeyType=100(交换密钥)

Key_FLAG.Request=1

Key_FLAG.Encryption=0

Key_FLAG.MIC=1

Key_FLAG.OperationType=00(建立过程)/01(更新过程)/10(删除过程)

D.10.3.5.1.2 协议数据

协议数据包含SWBKID、SWKeyID、MAC_{SW1}、MAC_{SW2}、N_{SW1}、UIE_{SW2}。分组中的协议数据的集合及元素ID定义见表D.54。

表 D.54 交换密钥协商确认分组有效元素集合

信息元素	元素 ID	元素长度 (八位位组)	定 义
SWBKID	0	16	D. 4. 1. 20
SWKeyID	1	1	D. 4. 1. 21
MAC _{SW1}	2	6	D. 4. 1. 8
MAC _{SW2}	3	6	D. 4. 1. 8
N _{SW1}	4	32	D. 4. 1. 3
保留	5~15		
保留	16~255		

其中：

- SWBKID：标识当前的交换基密钥，其定义见 D. 4. 1. 20；该字段值同收到的交换密钥协商响应分组中的 SWBKID 字段值；
- SWKeyID：其中比特 0 标识当前协商的交换密钥，其他位保留；本字段的比特 0 如果是利用交换基密钥第一次进行交换密钥协商，其值为 0；之后重新进行交换密钥更新时该位在 0 和 1 之间翻转，SWKeyID 的定义见 D. 4. 1. 21；该字段值同收到的交换密钥协商响应分组中的 SWKeyID 字段值；
- MAC_{SW1}：该字段的值为 MAC_{SW1}，其定义见 D. 4. 1. 8；
- MAC_{SW2}：该字段的值为 MAC_{SW2}，其定义见 D. 4. 1. 8；
- N_{SW1}：表示交换密钥协商标识 N_{SW1}，该字段值同交换密钥协商请求分组中的 N_{SW1} 字段的值，其定义见 D. 4. 1. 3。

D. 10. 3. 5. 1. 3 消息鉴别码 MIC

记为 MIC7 字段，表示消息鉴别码，由交换设备 SW₂ 利用生成的交换消息鉴别密钥 SW-MAK₁₋₂ 对交换密钥协商响应分组中除本字段外所有字段及下一次交换密钥协商过程中的交换密钥协商标识进行计算得到的杂凑值。

D. 10. 3. 5. 2 处理过程

交换设备 SW₁ 收到交换设备 SW₂ 发送的交换密钥协商确认分组后，进行如下处理：

- a) 比较 Key_FLAG 字段的 OperationType 字段、SWBKID 字段、SWKeyID 字段、MAC_{SW1} 字段、MAC_{SW2} 字段及交换密钥协商标识 N_{SW1} 与之前发送的交换密钥协商响应分组中的对应字段值是否一致，若有一个不一致，则丢弃该分组；若都一致，则执行步骤 b)；
- b) 利用计算得到的交换密钥中交换消息鉴别密钥 SW-MAK₁₋₂ 验证交换密钥协商确认分组中的消息鉴别码 MIC7 字段的正确性，若不正确，则丢弃该分组；若正确，则交换设备 SW₁ 确认交换设备 SW₂ 已获得与其一致的交换密钥。

若交换设备 SW_M 需要更新或者撤销交换设备 SW₁ 与 SW₂ 之间的交换基密钥，则构造交换基密钥通告分组发送给交换设备 SW₂/SW₁，要求交换设备 SW₂/SW₁ 更新或者删除与交换设备 SW₁/SW₂ 之间的交换基密钥。交换基密钥的更新或撤销过程和交换基密钥的建立过程相同，在具体实现时，可通过在上述交换基密钥通告过程中的每个分组中增加一个标识字段进行区分，用于标识通过交换设备 SW 完成交换设备 SW₁ 和 SW₂ 之间交换基密钥的建立、撤销或者更新过程。

若交换设备 SW_1/SW_2 需要更新或者撤销与交换设备 SW_2/SW_1 之间的交换密钥时,交换设备 SW_1/SW_2 构造交换密钥协商激活分组发送给交换设备 SW_2/SW_1 ,要求交换设备 SW_2/SW_1 更新或者删除与交换设备 SW_1/SW_2 之间交换密钥;交换密钥的更新、撤销过程和交换密钥的协商过程相同,具体实现时,通过在上述的交换密钥协商过程中的每个分组中增加一个标识字段进行区分,用于标识交换设备 SW_1 与 SW_2 之间交换密钥的协商、撤销或者更新。



附录 E

(资料性附录)

单向控制功能的考虑

E.1 概述

本标准中允许设置端口的 AdminControlledDirections 参数值为 In,是为了支持一些 PC 环境的特性:

- a) 远程唤醒;
- b) 对等唤醒;
- c) 远程控制;
- d) 告警。

E.2 远程唤醒

远程唤醒允许连接在 LAN 内的管理控制台对连接在 LAN 上的 PC 执行管理功能,不论 PC 是否开机,管理控制台均向 PC 发送一个特殊的封包“Magic Packet”,PC 的 LAN 适配卡识别出“Magic Packet”后,激活 PC。在引入可信第三方的实体鉴别及接入架构系统中,如果配置为“Full Control”,则 PC 关机将导致鉴别失败,“Magic Packet”将被 PC 连接的桥端口阻止,从而关闭远程唤醒功能。

使用单向控制允许“Magic Packet”被桥端口转发,恢复远程唤醒功能。然而,因为单向控制将允许任何帧通过端口,该特性将减弱引入可信第三方的实体鉴别及接入架构系统提供的保护。

E.3 对等唤醒

在使用 Windows 对等网络的 PC 环境中,任何 Windows 客户端可以和 LAN 内其他的 PC 共享资源。和远程唤醒一样,如果提供共享资源的 PC 正处于电源管理状态(例如,待机状态),则其共享资源对于其他 PC 将不可用。Windows 提供了 PC 的 MAC 驱动的一些帧模式,当这些模式被 MAC 收到后,和远程唤醒类似,会触发相关的恢复功能。然而,如果“Full Control”被配置,则桥连接 PC 的受控端口可能已经转换为 Disabled 状态,这些帧将被阻止。

同样,单向控制使唤醒功能可以使用,但降低了安全性。

E.4 远程控制

告警标准论坛(ASF)正在提出关于支持鉴别的客户远程控制功能(开机/关机、重启等)的标准项目。该功能使用和远程控制“Magic Packet”相似的方法,和前面的例子一样,该功能依赖于把数据包从桥端口传送到客户的能力。同样,单向控制提供了实现该功能的一个方法。

E.5 告警

E.5.1 一般要求

ASF 正在开发相关告警信息附录,该附录允许 PC 或工作站把存在于物理及操作环境的问题通知

给管理者(例如,温度过高、风扇停转、电源问题等)。这些告警通常采用 SNMP Trap 数据包形式,如果连接工作站的交换端口支持引入可信第三方的实体鉴别及接入方法,则当受控端口处于非授权状态,这些告警将被阻止转发。

为允许这些告警能够被转发到管理者,应利用非受控端口来接收和处理这些告警。关联到非受控端口的告警处理句柄识别包含有效告警信息的输入帧,有选择的转发它们到管理者。

E.5.2 告警封装

类型为 TAEPoL-Encapsulated-ASF-Alert 的数据包提供了使告警信息能够被封装到 TAEPoL 帧中,从而被关联到非受控端口的协议实体识别的方法。

E.5.3 请求者产生的告警

产生 ASF 告警的工作站有两个可能的处理策略:

- 在所有情况下,使用两种形式传送告警:一种封装在 TAEPoL 帧中,一种采用告警的原始形式(例如,SNMP Trap)。
- 根据它是否成功地完成了鉴别,请求者决定是发送告警的原始形式还是发送原始形式和 TAEPoL 封装形式。如果请求者和鉴别访问控制器鉴别成功,可以认为鉴别访问控制器的受控端口处于授权状态,因此就没有必要发送 TAEPoL 封装的告警;如果请求者和鉴别访问控制器鉴别失败,两种形式的告警都要发送。

E.5.4 鉴别访问控制器系统处理的告警

当从非受控端口收到一个 TAEPoL-Encapsulated-ASF-Alert 类型的 TAEPoL 帧,鉴别访问控制器 PAE 把帧发送到负责处理 ASF 告警的协议实体进行进一步处理。该协议实体应能够清楚地确定可接收的 ASF 告警信息的格式,不符合规则的数据帧将被丢弃。对于符合规则的 TAEPoL-Encapsulated-ASF-Alert 帧,ASF 协议实体可以采用下述任一策略进行处理:

- 根据受控端口的状态决定 TAEPoL-Encapsulated-ASF-Alert 帧是否应当被拆封并转发。如果受控端口处于授权状态,由于请求者将发送告警的非封装形式,ASF 协议实体丢弃任何的 TAEPoL-Encapsulated-ASF-Alert 帧。
- 忽略受控端口的状态,被拆封并转发告警。

应注意,当受控端口处于授权状态时,TAEPoL-Encapsulated-ASF-Alert 帧会同时被受控端口和非受控端口收到。

注:使用该机制处理告警时,希望告警的最大速率小于 10 告警/s。

E.5.5 ASF 告警接收器的隐含意义

结合请求者和鉴别访问控制器处理 TAEPoL-Encapsulated-ASF-Alert 帧的策略,这意味着最终 ASF 告警的接收器可能收到同一个告警信息的两份拷贝。

参 考 文 献

- [1] RFC1414 Identification MIB.
 - [2] ISO/IEC 14882:2003 Programming languages—C++.
 - [3] Stubblefield, A. ,Ioannidis, J. ,Rubin, A. ,“Using the Fluhrer, Mantin and Shamir Attack to Break WEP”,2002 NDSS Conference.
 - [4] RFC 3748 Extensible Authentication Protocol(EAP).
 - [5] GB/T 16263.1 信息技术 ASN.1 编码规则:基本编码规则(BER)、正则编码规则(CER)和特异编码规则(DER)的规范(GB/T 16263.1—1996, idt ISO/IEC 8825-1:1995).
-



中 华 人 民 共 和 国
国 家 标 准
信息安全技术 引入可信第三方的实体
鉴别及接入架构规范

GB/T 28455—2012

*

中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100013)
北京市西城区三里河北街16号(100045)

网址: www.gb168.cn

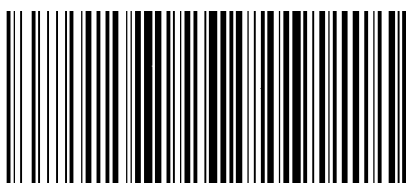
服务热线: 010-68522006

2012年11月第一版

*

书号: 155066·1-45597

版权专有 侵权必究



GB/T 28455-2012