CBWIPS

宽 带 无 线 IP标 准 工 作 组 标 准

CBWIPS ××××.××—××××

基于虎符 TePA 的 IP 安全技术规范

TePA-based IP Security Technical Specification

（征求意见稿）

（在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。）

XXXX - XX - XX 发布　　XXXX - XX - XX 实施

工业和信息化部宽带无线 IP 标准工作组　发 布

目　　录

CBWIPS

前　言

本标准按照GB/T1.1—2009给出的规则起草。

CBWIPS

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的连接到网络上，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。

因此本标准的主要目标是提出一套适用于网络访问控制、数据保护及身份管理的，具有普遍适用性的实体鉴别与安全接入协议和结构，并为其通信提供安全保护方法。本规范将采用密码技术，并引入在线的可信第三方，构建鉴别协议，并定义网络安全IP层数据安全保护。

本标准主要内容是：

——引入可信第三方的实体鉴别及IP层数据安全保护，将参加鉴别和授权的实体置于对等的角色。

——引入可信第三方的实体鉴别及IP层数据安全保护，实现IP数据通信过程节点之间身份鉴别。

——实现节点之间IP数据加密和完整性的保护。

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。

CBWIPS

基于虎符 TePA 的 IP 安全技术规范

1 范围

本标准规定了引入可信第三方的实体鉴别及IP层数据安全保护的方法。包括：

a) 引入可信第三方的实体鉴别及IP层数据安全保护的结构框架；
b) 引入可信第三方的实体鉴别及IP层数据安全保护的基本原理；
c) 定义引入可信第三方的实体鉴别及IP层数据安全保护的参与实体间的消息交互协议；
d) 定义使用消息交互协议完成实体鉴别及IP层数据安全保护方法和过程；
e) 规定协议交互消息中的报文格式；
f) 规定IP层数据安全保护方法框架、数据封包及数据转交的过程。

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，仅所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 15843.3-2008/XG1 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 第1号修改单

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 第1号修改单（Information technology -- Security techniques -- Entity authentication--Part 3： Mechanisms using digital signature techniques--Amendment 1）

3 术语和定义

下列术语和定义适用于本标准。

鉴别访问控制器（AAC），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。

请求者（REQ），Requester，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。

鉴别服务器（AS），Authentication Server，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。

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

下列缩略语适用于本标准。

DHCP	动态主机配置协议 （Dynamic Host Configuration Protocol）
IP	互联网协议 （Internet Protocol）
MIC	消息完整性校验 （Message Integrity Check）
ICV	完整性检查值 （Integrity Check Value）
Port	网络访问端口 （Network Access port）
REQ	请求者 （Requester）
TAEP	三元鉴别可扩展协议 （Tri-element Authentication Extensible Protocol）
TePA	三元对等鉴别 (Tri-element Peer Authentication)
TISec	基于三元对等鉴别的IP安全技术（TePA-based IP Security）
VPN	虚拟专用网络 (Virtual Private Network)

TAI	基于三元对等鉴别的鉴别基础结构（TePA-based Authentication Infrastructure）
TUE	隧道统一封装 （Tunnel Universal Encapsulating ）
SR	静默转交 （Silent Relay）
DR	坞接转交 （Dock Relay）
RSAP	注册安全关联参数 （Registry Security Association Parameters）
ASAP	鉴别安全关联参数 （Authentication Security Association Parameters）
TSAP	隧道安全关联参数 （Tunnel Security Association Parameters）

5 引入可信第三方的实体鉴别及 IP 层数据安全保护架构

5.1 概述

引入可信第三方的实体鉴别，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应用于IP层数据的安全保护。

本章描述了引入可信第三方的实体鉴别及IP层数据安全保护的结构框架、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IP层数据安全保护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。

引入可信第三方的实体鉴别及IP层数据安全保护设计目标为节点之间提供身份鉴别的安全框架，从而为IP层数据传输提供安全通路，具体包括节点身份认证、数据完整性、数据秘密性等。TISec技术包括TAI鉴别协议、TRD数据转交技术及TUE协议三部分。TAI鉴别协议提供节点身份鉴别、权限管理、密钥管理、证书管理等功能，并提供不同密码算法的组合使用。TRD主要负责节点数据转交的处理。TUE协议定义了IP层数据的封装方法，用以保护数据秘密性、完整性及抗重放能力。

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IP层数据安全保护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隧道安全关联参数。

5.3 系统

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IP层数据安全保护系统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IP层数据安全保护的操作，系统的端口可以定义为以下两种角色：

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。

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TISec系统中都需要，以完成鉴别协议交互。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和AS在一个系统中实现TISec系统工作概况如图1。

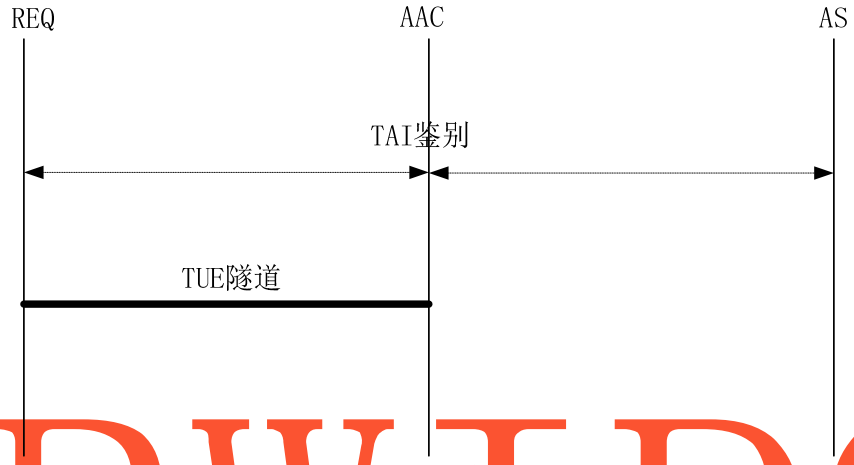


图1 TISec工作流程

TISec系统工作流程：

1、REQ、AAC和AS通过TAI鉴别协议完成身份鉴别；

2、根据步骤1鉴别结果，REQ和AAC之间建立基于TSAP的TUE安全隧道；

3、REQ和AAC之间数据通信通过TUE协议封装完成。

5.4 功能与角色职能

请求者REQ和鉴别访问控制器AAC作为TISec系统实体，在不同应用场景下会有不同的物理设备作为载体。如REQ可以实施在嵌入式终端、手持设备及PC主机等，AAC作为TISec系统的汇聚节点可以集成在网关设备或者作为单独的接入服务器。

REQ和AAC作为TUE隧道数据传输的两端，TUE所封装的IP数据可以是REQ和AAC节点自身产生，同时也可以是REQ和AAC所保护的从属设备，如AAC可以保护其汇聚网络后方的所有网络节点数据通信的安全性，详细过程参见第8章节。

TISec用来保护IP数据的安全性，应用可分为两种场景：

1、 网络对网络的数据安全的保护，TISec节点作为安全网关的方式保护网络两端的数据安全，分组数据源可为多台节点设备或者TISec节点后的整个网络。

2、 端对端（TISec节点之间）的数据安全保护，作为通信节点的设备实施TISec之后，它们之间的IP通信安全由TISec保护。

6 TAI 协议

要实现通信节点之间IP层数据的安全保护，必须实现对节点之间的身份鉴别，本标准中以AS作为可信的第三方。图2为TISec网络基本结构。

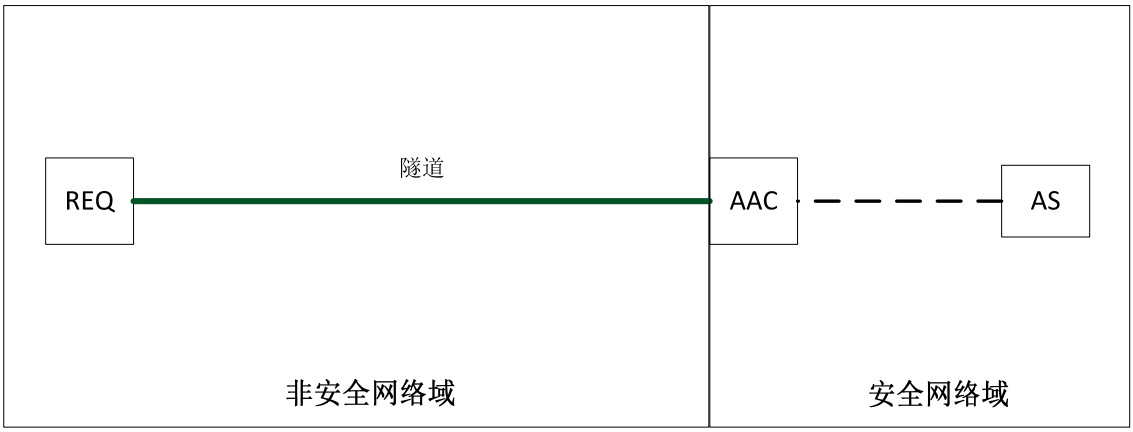


图2 TISec网络基本结构

图 2 中将 TISec 安全框架所处网络区域划分为非安全网络域和安全网络域两部分，非安全网络域泛指公共网络接入域，如互联网环境，而安全网络域专指 AAC 所处的受控制网络区域，如与互联网环境隔绝的机构私有网络域。

网络访问控制管理通过下面所述的控制技术，按照设定的控制机制，实现终端到访问节点的连结被合法的使用，其中的数据传输受到保护。

本网络访问控制机制用于网络层，所定义的安全协议报文传递时，使用数据报文。对于传输层使用TCP/UDP协议时，可使用TCP/UDP端口号4113/3833分别完成安全协议和数据交互，本标准中规定4113端口用于TAI鉴别协议，3833端口用于TUE协议。

6.1 TAI 预授权

TAI协议支持证书和预共享密钥两种鉴别方式，预授权方法如下：

1、证书颁发

AS 自颁发证书 $Cert_{AS}$ 及相应私钥 Key_{AS}，然后 AS 给 REQ 颁发证书 $Cert_{REQ}$ 及相应私钥 Key_{REQ}，给 AAC 颁发证书 $Cert_{AAC}$ 及相应私钥 Key_{AAC}。

REQ 存有 $Cert_{AS}$，$Cert_{REQ}$，和 Key_{REQ}。

AAC 存有 $Cert_{AS}$，$Cert_{AAC}$，和 Key_{AAC}。

AS 存有 $Cert_{AS}$ 和 Key_{AS}，以及证书管理相关信息。

REQ 的证书中的证书持有者字段要设置为 REQ 的身份。

2、共享密钥发放

AS 将对应于 REQ 的共享密钥 Key_{share} 发放给 REQ 和 AAC。

6.2 TAI 消息格式

TAI交互消息定义规定了请求者REQ和鉴别访问控制器AAC及鉴别服务器AS之间的数据分组格式。TAI的交互消息采用TAEP协议进行封装，TAEP封装中Type字段取值为240用于标识TAI协议。图3为TAI协议家别交互过程分组传递示意图。

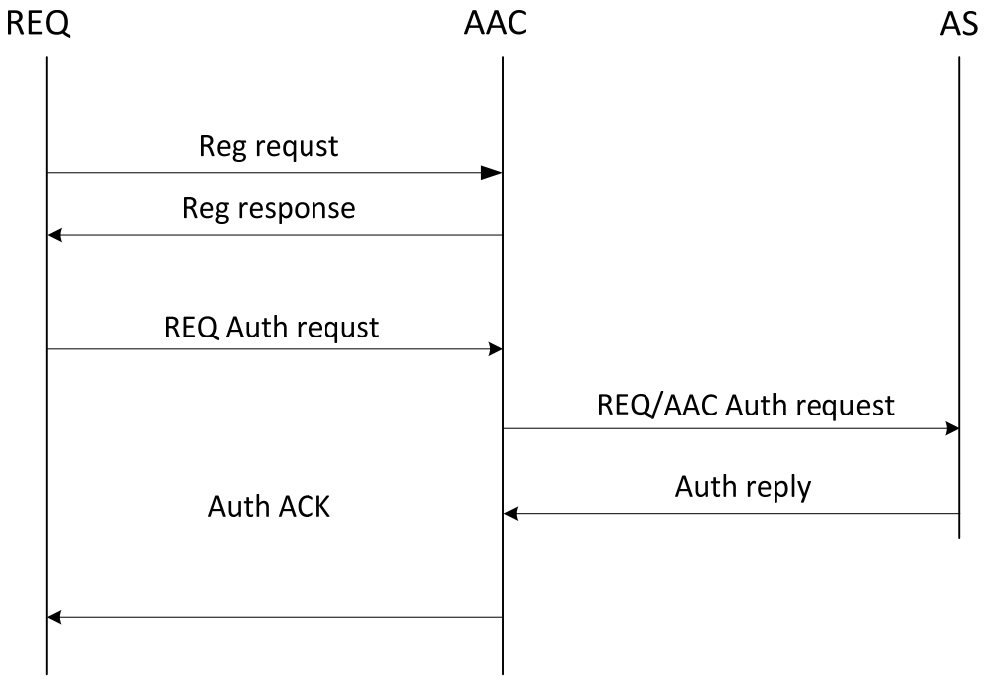


图3 TAI鉴别交互流程

图 4 为 TISec 实体之间进行身份鉴别使用的 TAI 鉴别过程，通过六次分组传递过程完成。

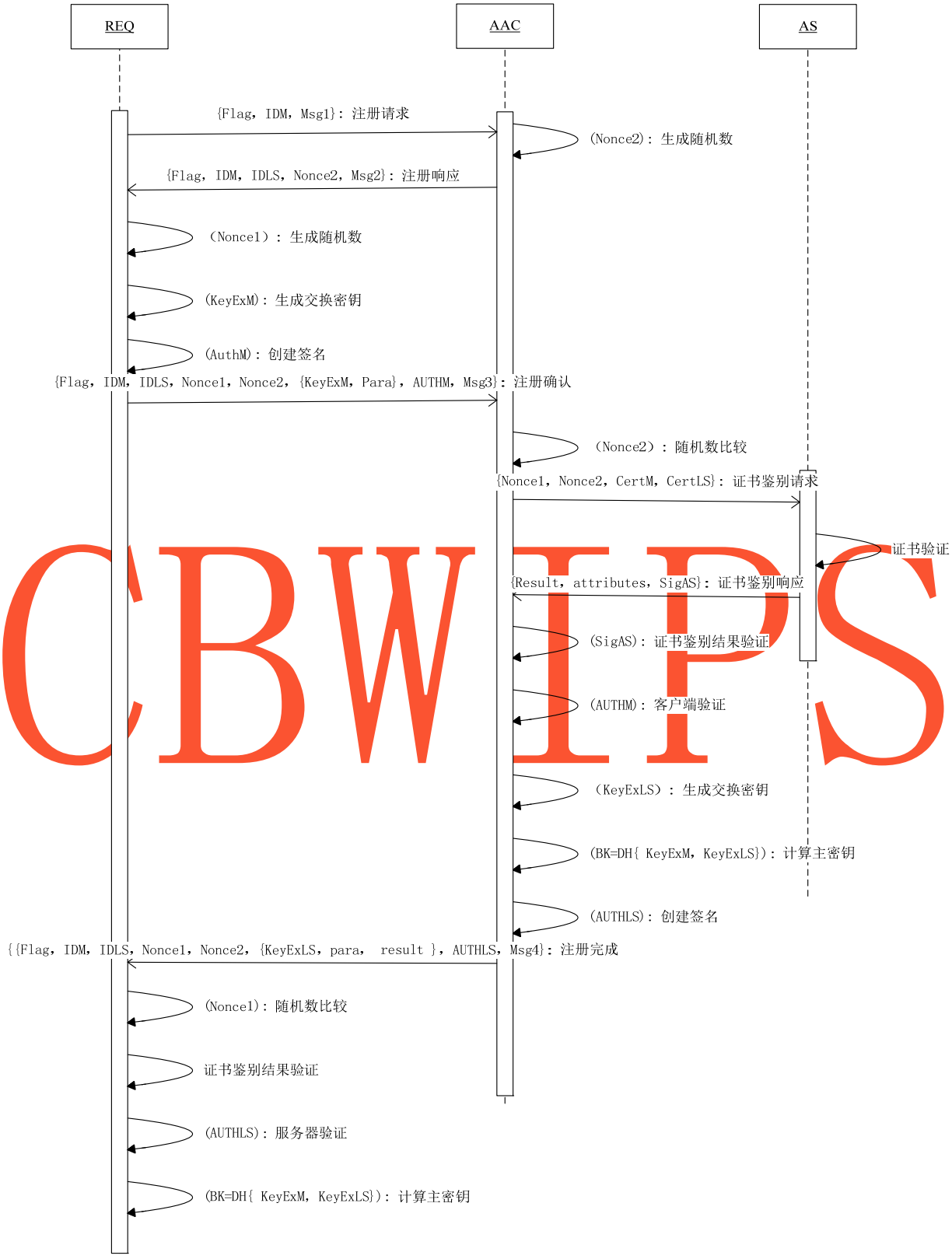


图4 TAI鉴别交互流程

6.2.1 TAI 鉴别交互帧格式

版本（Version）	类型（Type）	子类型（Subtype）	保留（Reserve）	长度(Length)	标识（Flag）	数据(Data)
2Bytes	1Byte	1Byte	2Bytes	2Bytes	1Byte	变长

图 5 TAI 鉴别帧数据格式

其中：

——版本号字段长度为2个八位位组，表示鉴别基础结构的版本号，当前版本号为1，其他保留；

——类型字段长度为1个八位位组，表示协议类型，定义如下：

1 TAI协议分组；

其他值保留。

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

1 注册请求；

2 注册响应；

3 注册确认；

4 注册完成；

5 证书鉴别请求；

6 证书鉴别响应；

7 密钥更新请求；

8 密钥更新相应；

其他值保留。

——保留字段长度为2个八位位组，默认值为0。

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

——数据字段的内容根据类型和子类型的值而定，它除了包含固定的内容，还可以包含可选的属性。定义TAI协议分组的最大长度为65535个八位位组。

6.2.2 共用固定字段格式如下

6.2.2.1标识 FLAG

长度为 1 个八位位组。格式如下：

方向	认证方式	认证结果	扩展标识	保留
1Bit	1Bit	1Bit	1Bit	4Bits

图 6 标识 FLAG 格式

图 6 中各位标识如下：

方向位： 0 标示 REQ 到 AAC，1 标示 AAC 到 REQ；

认证方式位： 0 标示共享密钥 1 标示证书方式；

认证结果位： 0 标示失败 1 标示成功；

扩展标识位： 0 标示无扩展 1 标示有扩展；

其余四位保留。

6.2.2.2 身份类型，该类型图示如图 7：

身份标识	身份长度	身份数据
2Bytes	2Bytes	0-65535Bytes

图 7 身份类型数据格式

——身份标识字段表示身份类型，长度为 2 个八位位组。身份标识定义如下：
1 表示该字段的身份数据由 X.509 v3 证书的持有者名称、颁发者名称、序列号字段组成；
2 表示该字段的身份数据由 GBW 证书的持有者名称、颁发者名称、序列号字段组成；
其他值保留。
——身份长度字段长度为 2 个八位位组，标识身份数据字段的八位位组数。
——份数据字段为从证书中提取出的持有者名称、颁发者名称、序列号字段：
身份长度，2 个字节，是身份字段格式的总长度。

6.2.2.3 用户签名字段格式

签名类型	签名长度	签名值 Result
1Byte	2Bytes	变长

图 8 用户签名数据格式

图 8 中签名类型 1 ECDSA192 签名；
签名长度 2 字节，为签名字段的总长度；
签名值 Result 字段格式如下表 1：

表 1 签名值字段格式

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

6.2.2.4 para 字段格式

参数标识	字段长度	参数内容
1Byte	2Bytes	变长

图 9 para 字段数据格式

图 9 中 para 参数字段由参数标识和参数长度和参数内容组成，参数标识字段长度为 1 个八位位组；参数长度字段为 2 个八位位组，表示参数内容字段的八位位组数。参数字段的值定义如下：

——参数标识为 1 时，标识参数以 OID 方式表示，参数长度字段表示 OID 标识的八位位组数，参数内容为 OID 编码。

——参数标识其他值保留。

6.2.2.5交换密钥数据字段

交换密钥长度	交换密钥内容
1Byte	变长

图 10 交换密钥数据格式

图10中交换密钥数据字段由长度字段与内容字段组成。其中长度字段为1 个八位位组，表示内容字段的八位位组数。

6.2.3 注册请求报文字段格式

用户身份	用户索引	MSG1
4Bytes	4Bytes	变长

图 11 注册请求报文格式

注册请求报文对应的子类型为1，封装在TAEP分组中时，Code为1（Request），Type值为240（TAI）。
图 11 中用户身份内容使用授权 SPI 值，32 位位组；
用户索引使用 32 位用户唯一标识；
MSG1 保留自定义。

6.2.4 注册响应报文字段格式

用户身份	AAC 身份	AAC 一次性随机数	MSG2
4Bytes	4Bytest	32Bytes	变长

图 12 注册相应报文格式

注册响应报文对应的子类型为2，封装在TAEP分组中时，code为2（Response），Type值为240（TAI）。
图 12 中用户身份内容使用授权 SPI 值，32 位位组；
AAC 身份内容使用默认服务器 SPI 值，32 位位组，值为 0x00000000；
AAC 一次性随机数 Nonce 为 32 字节；
MSG2 保留自定义。

6.2.5 注册确认报文字段格式

$\{ID_{REQ}, ID_{AAC}, Nonce1, Nonce2, \{KeyEx_{REQ}, Para\}, AUTH_{REQ}, Msg3\}$

REQ 身份	AAC 身份	REQ 随机数	AAC 随机数	REQ 交换密钥	AAC 签名	MSG3
变长	4Bytes	32Bytes	32Bytes	变长	变长	变长

图 13 注册确认报文格式

注册确认报文对应的子类型为3，封装在TAEP分组中时，Code为1（Request），Type值为240（TAI）。
图 13 中 REQ 身份内容使用自身证书；
AAC 身份内容使用默认 AAC SPI 值，32 位位组，值为 0x00000000；
REQ 交换密钥依照 6.2.2.5 章节固定格式描述部分；
REQ 签名依照 6.2.2.3 章节共用字段格式部分描述；
MSG3 见交 6.1.10 附加消息。

6.2.6 注册完成报文字段格式

{{ID_{REQ}，ID_{AAC}，Nonce1，Nonce2，{$KeyEx_{AAC,}$ para}， result }，$AUTH_{AAC}$，Msg4}

REQ 身份	AAC 身份	REQ 随机数	AAC 随机数	AAC 交换密钥	证书 鉴别结果	AAC 签名	MSG4
变长	变长	32Bytes	32Bytes	变长	变长	变长	变长

图 14 注册完成报文格式

注册请求报文对应的子类型为4，封装在TAEP分组中时，Code为2（Response），Type值为240（TAI）。
REQ 身份内容使用自身证书；
AAC 身份内容使自身证书；
AAC 交换密钥依照固定格式描述部分；
证书鉴别结果见 6.2.8 证书鉴别响应部分字段描述；
AAC 签名依照共用字段格式部分描述；
MSG4 见交 6.2.10 附加消息。

6.2.7 证书鉴别请求报文字段格式

地址索引 ADDID	AAC 挑战	REQ 挑战	REQ 证书	AAC 证书	REQ 信任的 AS 列表
12Bytes	32Bytes	32Bytes	变长	变长	变长

图 15 证书鉴别请求报文格式

在证书鉴别请求中，AAC 向 AS 发送证书鉴别请求报文，该报文对应的子类型为 5，封装在 TAEP 分组中时，Code 为 1（Request），Type 值为 240（TAI）报文数据字段格式及内容如下：

地址索引，12 字节，由 4 字节用户 SPI 索引和 8 字节的保留字段 0 依次组成。即相当于接入用户的网络地址索引编号。

AAC 挑战，32 个字节，由 AAC 采用随机数生成算法生成。REQ 对此随机数将在验证结果中

的一次性随机数中返回。

REQ 挑战，32 个字节，由 REQ 采用随机数生成算法生成。REQ 对此随机数将在验证结果中的一次性随机数中返回。

REQ 证书，变长，由证书类型 组成。

AAC 证书，变长，由证书类型 组成。

REQ 信任的 AS 列表，由身份列表类型组成。

附注：

证书类型，该类型如下：

证书标识	证书长度	证书数据
2Bytes	2Bytes	0-65535Bytes

其中：

证书标识字段表示证书类型，2 个字节，其取值如下：

1，表示 X.509 证书；

2，表示该证书为 GBW 证书；

其它值保留。

证书长度是证书字段格式的总长度。

身份类型，该类型图示如下：

身份标识	身份长度	身份数据
2Bytes	2Bytes	0-65535Bytes

——身份标识字段表示身份类型，长度为 2 个八位位组。身份标识定义如下：

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

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

其他值保留。

——身份长度字段长度为 2 个八位位组，标识身份数据字段的八位位组数。

——身份数据字段为从证书中提取出的持有者名称、颁发者名称、序列号字段：

身份长度，2 个字节，是身份字段格式的总长度。

身份列表类型，该类型图示如下：

其中，下一个属性字段是下一个属性的类型。若后面无其他属性，该字段为 0。

类型=3 （1）	长度 （2）
保留（1）	身份个数（2）

身份 1（变长）
身份 2（变长）
……

长度，2 个字节，是身份列表字段的总长度。
保留字段的值默认是 0。
身份个数，2 个字节，表示该子类型中的身份字段总数；
身份，可变长度，为图 1-5 中的格式。

6.2.8 证书鉴别响应报文字段格式

地址索引 ADDID	证书的验证结果	REQ 信任的服务器签名	AAC 信任的服务器签名
12Bytes	变长	变长	变长

图 16 证书鉴别相应报文格式

在证书鉴别响应中，AS 向 AAC 发送证书鉴别响应报文，报文对应的子类型为 6，封装在 TAEP 分组中时，Code 为 2（Response），Type 值为 240（TAI）。图 16 中：

地址索引，12 字节，由 4 字节用户 SPI 索引和 8 字节的保留字段 0 依次组成。

证书的验证结果，可变长度，证书验证结果类型。

REQ 信任的服务器签名，可变长度，签名属性类型。它对本分组中的证书的验证结果字段进行签名。签名算法由 AS 自己定义，与密码算法套件无关。进行签名运算时，本签名字段使用的签名属性中的签名值为 0。

AAC 信任的服务器签名，可变长度，签名属性类型。它对本分组中除地址索引和本字段的所有其他字段进行签名，该签名的算法由 AS 自己定义，与密码算法套件无关。进行签名运算时，本签名字段使用的签名属性中的签名值为 0。签名的算法为：先对报文的整个内容进行摘要，然后用私钥对摘要加密。

注：

若 REQ 信任的服务器和 AAC 信任的服务器是一个，则证书鉴别响应分组中不包含 REQ 信任的服务器签名字段。

证书结果类型定义如下：

其中：

长度，2 个字节，表示证书结果类型的总长度。

一次性随机数，即 REQ 挑战和 AAC 挑战的原样拷贝。

证书字段，请参考上一小节的证书类型。

结果字段，1 个字节，具体的值及含义：

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， 表示证书未按照规定用途使用；

7， 表示证书吊销状态未知；

8， 表示证书错误原因未知。

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

其他值保留。

签名属性类型定义如下：

类型=1	长度	身份	签名算法	签名值
1Byte	2Bytes	变长	变长	变长

其中：

长度，2 个字节，签名属性类型总的长度。

身份，可变长度，其格式为图 1-5 中定义的格式。

签名算法，包含长度和内容两个子字段。其中长度为 2 个字节，表示内容字段的字节数。内容字段由 1 个字节的杂凑算标识和 1 个字节的签名算法标识组成。

取值及含义分别如下：

杂凑算法标识，1 个字节，取值及含义如下：

表示 SHA-256 杂凑算法。

其它值保留。

签名算法标识，1 个字节，取值及含义如下：

1，表示 192 位的椭圆曲线数字签名算法，即 ECDSA-192；

其它值保留。

签名值，可变长度，包含长度和内容。签名的方法为：先对报文的整个内容进行摘要，然后用私钥对摘要加密。

6.2.9 TAI 密钥交换消息

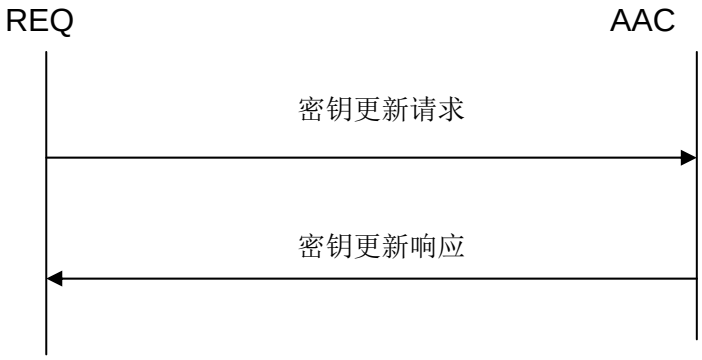


图 17 TAI 密钥交换

a） 终端根据策略（时间或数据包），发起密钥更新请求{Flag，ID_{REQ}，ID_{AAC}，sn，Nonce3，MIC_Key_{Kreq}}，sn 单调递增。

b） 接入服务器收到密钥更新请求后，验证 sn 单调递增和 MIC 正确后，生成 Nonce4，计算 Km’|Kd’|Ki’=prf（Key_{share}，Nonce3，Nonce4，”ID_{REQ}|ID_{AAC}|IP”）。发送密钥更新响应{Flag，ID_{REQ}，ID_{AAC}，sn，Nonce4，MIC_Key_{REQ}}。

c） 终端收到密钥更新响应后，验证 sn 与密钥更新请求中的 sn 相同后，计算 Km’|Kd’|Ki’=prf（Key_{share}，Nonce3，Nonce4，”ID_{REQ}|ID_{AAC}|IP”），验证 MIC 正确后，启用新密钥。

Nonce，一次性随机数，长度为 32 个八位位组。

MIC，和 HMAC-SHA256 相同

Sn，sn 为 16 个八位位组，初始值为 0x5C365C365C365C365C365C365C365C36，在每次密钥更新请求时该字段值加 1。

在密钥更新请求报文对应的子类型为 7，封装在 TAEP 分组中时，code 为 1（Request），Type 值为 240（TAI），密钥更新相应报文对应的子类型为 8，封装在 TAEP 分组中时，code 为 1（Request），Type 值为 240（TAI）。

数据字段定义：

密钥更新请求

Flag	ID_{REQ}	ID_{AAC}	sn	Nonce3	MIC_Key_{Kreq}
1byte	变长	变长	16byte	32byte	32byte

|————————————MIC 计算范围————————————|

密钥更新响应

Flag	ID_{REQ}	ID_{AAC}	sn	Nonce4	$MIC\_Key_{Kreq'}$
1byte	变长	变长	16byte	32byte	32byte

|————————————MIC 计算范围————————————|

6.2.10 端点状态检测消息

REQ和AAC之间完成TAI协议交互之后需要定期检测彼此的存活状态，AAC作为接入汇聚需要维护所有通过鉴别的REQ状态，而所需的交互消息称为端点状态检测消息。REQ和AAC通过状态检测完成各自状态的自制维护，它们之间并不产生注销消息。REQ和AAC之间的请求、应答报文按照6.1.3注册请求和6.1.4注册响应报文格式进行。

端点状态检测消息由REQ主动发起，AAC负责受理并维护已鉴别的REQ端点状态，REQ通过设定的时间周期定期向AAC发送注册请求，AAC收到此消息后更新对应REQ RSAP中的TimeOut时间戳（时间以AAC时间为权威，可通过NTP等辅助方法校正），并为REQ回应注册响应报文。状态检测机制在REQ和AAC两端分别对应不同的处理方法。

1、 REQ端处理机制

REQ鉴别完成后会收到AAC约定的维护参数，包括状态检测消息维护次数和时间间隔。REQ根据参数定义定期发送注册请求，单次注册请求生命周期为10秒，连续两次未收到注册响应则触发异常处理流程。异常流程包括缩短状态检测消息发送周期，连续发出注册请求，如果5次注册请求或者10秒内无注册响应的情况下REQ将进入离线状态。

2、 AAC端处理机制

AAC为通过鉴别的REQ下发状态检测消息维护参数，并根据参数更新RSAP中TimeOut时间戳。AAC根据收到的注册请求判断REQ的存活状态，在总的存活周期内（维护次数和维护周期总时长）保持REQ存活状态，反之将清除相应REQ的RSAP及TSAP状态信息。

6.2.11 TAI 附加消息

TAI附加消息是指在TAI鉴别过程中附加在注册请求、注册响应、注册确认及注册完成四个分组之后的消息字段，分别为MSG1、MSG2、MSG3及MSG4。MSG1和MSG2保留定义，MSG3和MSG4定义用于REQ和AAC扩展信息交互MSG3包括了REQ需要告知AAC的消息内容，MSG4则是AAC需要告知REQ的消息内容，所有附加消息数据字段经过签名保证完整性。

6.2.11.1 附加消息分组格式

Main version （8位比特）	Sub version （8位比特）	Direction （8位比特）	Flag （8位比特）
Length （16位比特）		Data（变长）	

图 17 TAI 附加消息分组格式

TAI 附加消息分组各个域的定义见表 2。

表 2 TAI 附加消息分组域定义

名称	长度（比特）	描述
Main version 主版本	8	表示 TAI 附加消息分组的主版本类型： 1：主版本为 1

		其他留扩展
Sub version 子版本	8	表示 TAI 附加消息分组的子版本类型： 1：子版本为 1 其他留扩展
Direction 传递方向	8	表示 TAI 附加消息分组的数据传递方向： 1：REQ向AAC传递数据； 2：AAC向REQ传递数据； 3：REQ向ACC扩展中心传递数据； 4：AAC扩展中心向REQ传递数据；
Flag 标志	8	表示 TAI 附加消息分组中的唯一性标志： 1：若标志位首位设置，则 TAI 附加消息的算法类型等仅能设置一种；否则，可设置多种算法类型； 2：AAC 向 REQ 传递数据，该标志位首位必须设置。
Length 长度	16	表示 TAI 附加消息分组的八位位组数，即指包括 Main version、Sub version、Direction、Flag、Length 和 Data 所有字段的长度总和
Data 数据	变长	分组含多个八位位组

6.2.11.2 附加消息 Data 字段格式

附加消息数据字段指的是 TAI 附加消息分组格式中的可变的 Data 数据字段。其分组格式如下：

DataType （16位比特）	DataLength （16位比特）
SubType（16位比特）	SubLength（16位比特）
SubData(1)	
SubType（16位比特）	SubLength（16位比特）
SubData(2)	
……	
SubType（16位比特）	SubLength（16位比特）
SubData(n)	

其中 DataType 和 DataLength 字段定义如表 3。

表 3 附加消息数据字段分组定义

名称	长度（比特）	描述
DataType 数据类型	16	表示附加消息数据分组的数据类型定义.其定义如下： 0xF001：附加消息为 TAI 注册请求的附加消息； 0xF002：附加消息为 TAI 注册响应的附加消息； 0xF003：附加消息为 TAI 注册确认的附加消息； 0xF004：附加消息为 TAI 注册完成的附加消息；
DataLength 数据长度	16	表示附加消息数据分组的八位位组数，即指包括 DataType、DataLength 和 DataContent 所有字段的长度总和

其中 SubType、SubData 域的定义和描述见表 4。

表 4 SubType、SubData 定义

SubType定义	说明	SubData描述
0x0000	TUE隧道版本	表示当前TUE版本号。 字段长度为2字节。定义如下： 第1字节：主版本，为0x01； 第2字节：子版本，为0x01；
0x0001	AAC子网信息	表示AAC所支持的子网信息。 每五字节为一个子网信息，字段长度为5的整数倍。其中一条网段定义如下： 第1~4字节：表示子网地址； 第5字节：表示子网掩码；
0x0002	REQ版本	表示当前REQ的版本类型。 字段长度为7字节。定义如下： 第1字节：主版本； 第2字节：子版本； 第3~6字节：Build号； 第7字节：发行类型（ 0：测试版本；1：Beta版本；2：发布版本；其它：保留）；
0x0003	AAC版本	同 REQ版本定义
0x1001	加密算	表示当前REQ（或AAC）支持的加密算法类型。

	法	每3字节为一种加密算法类型，字段长度为3的整数倍。其中一种算法定义如下： 第1字节：算法类型（0：SMS4算法；1：SM1算法；2：AES算法；3：SM6算法；其它：保留）； 第2字节：密钥长度（0：64位；1：128位；2：192位；3：256位；其它：保留）； 第3字节：加密模式（0：CBC；1：OFB；其它：保留）
0x1002	压缩算法	表示当前REQ（或AAC）支持的压缩算法。 字段长度为1字节。定义如下： 0：不支持压缩算法； 1：支持压缩算法； 其它：保留。
0x1003	校验算法	表示当前REQ（或AAC）支持的校验算法。 字段长度为1字节。定义如下： 1：支持HMAC-SHA256校验算法； 2：支持HMAC-SHA1校验算法； 其它：保留。
0x2001	REQ虚拟地址	表示当前AAC为REQ所分配的IP地址。 字段长度为5字节。定义如下： 第1~4字节：IP地址； 第5字节：子网掩码。
0x4001	AAC地址	表示当前AAC地址信息。 字段长度可变。定义如下： 4字节长度：AAC的IPv4地址； 16字节长度：AAC的IPv6地址。
0x4002	REQ资源路径	表示当前REQ需要获取资源的路径，如URI。 字段长度可变。
0x4003	REQ安全自检	表示AAC为REQ下达的安全检测指令。 字段长度为1字节。定义如下： 0：不进行REQ安全自检； 1：进行REQ安全自检。
0x8001	REQ有效	表示当前REQ的有效期限。

	期限	字段长度为8字节。定义如下： 第1~2字节：表示有效期限的开始年份； 第3字节：表示有效期限的开始月； 第4字节：表示有效期限的开始日； 第5~6字节：表示有效期限的终止年份； 第7字节：表示有效期限的终止月； 第8字节：表示有效期限的终止日。
0x8002	REQ DNS	表示当前REQ虚拟地址所依赖的DNS地址。 每4字节为一个DNS信息，可支持多条DNS地址，字段长度为4的整数倍。定义如下： 第1~4字节：REQ DNS的IP地址；
0x8003	TUE维护参数	表示当前TUE隧道的维护参数，通过该参数，可以对TUE隧道进行调整。 字段长度为2字节。定义如下： 第1字节：TUE连续维护次数； 第2字节：TUE维护间隔时间。
0x8004	AAC地址信息	表示AAC对外服务的地址和端口信息。 每8字节为一条链路信息，可支持多条链路，字段长度为8的整数倍。定义如下： 第1~4字节：AAC链路地址； 第5~6字节：AAC链路注册端口； 第7~8字节：AAC链路数据端口；
0x8005	REQ用户名	表示当前REQ关联用户名。 终端用户名为变长数据，最大长度为16字节。
0x8006	AAC准入区名称	表示当前AAC准入区名称，REQ可以同时接入多个AAC，即成为多个准入区。 准入区名称为变长数据，最大长度为16字节。
0x8080	REQ证书更新	表示当前REQ证书更新信息。 字段长度为8字节。定义如下： 第1~4字节：REQ证书更新地址； 第5~6字节：REQ证书更新端口； 第7~8字节：REQ剩余有效天数；

7　安全关联参数 SAP

安全关联参数SAP定义了TAI鉴别和TUE的参数集合，用以标识TAI鉴别和TUE之间对应关系，包括REQ和AAC之间建立SAP的算法信息、SPI、路由信息、节点策略等。

一次TAI会话过程中SAP分为登记和仲裁两个阶段，登记阶段产生注册RSAP和认证ASAP两个参数集合，仲裁阶段产生隧道TSAP参数集合。不同的集合标识REQ和AAC不同的时间域内的交互信息。图18为安全关联参数SAP状态机及转换流程。

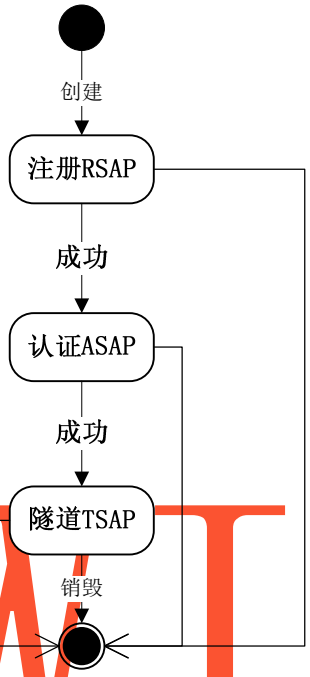


图18 SAP状态转换图

注册RSAP在TAI鉴别请求开始为REQ建立一组参数集合，该集合不包含节点差异化参数（基本信息除外），如算法、策略等信息。

7.1　注册 RSAP 参数信息

注册RSAP用于TAI鉴别发起请求和响应的关联参数，包括如下信息：

SPI （32位比特）	
req_addr （32位比特）	
timeout （32位比特）	
create_time （32位比特）	
tue_dev （16字节）	
peeraddr （32位比特）	
localaddr （32位比特）	
peerport （16位比特）	localport （16位比特）

sockfd （32位比特）			
usr_name （16字节）			
user_type (32位比特)			
encryptflg （8位比特）	compressflg （8位比特）	verifyflg （8位比特）	flag （8位比特）
req_param (变长)			

名称	长度（比特）	描述
SPI REQ 标识	32	REQ 唯一标识
req_addr 虚拟IP	32	REQ 虚拟地址
timeout 隧道超时	32	TUE 隧道超时时间，单位秒，超时后隧道被注销
create_time 注册发起时间	32	AAC收到REQ发起注册请求的系统时间
tue_dev 隧道设备名称	128	AAC给REQ分配的隧道设备的名称
peeraddr 终端IP	32	REQ 发起注册请求的宿主 IP 地址
localaddr AAC IP	32	AAC为REQ建立的隧道的网络设备IP地址
peerport 终端端口	16	REQ 发起注册请求的端口号
localport 服务器端口	16	AAC 为 REQ 建立的隧道的端口号
sockfd 隧道SOCKET	32	AAC 给 REQ 建立的文件描述符
usr_name REQ名称	128	REQ 名称
user_type REQ类型	32	REQ 类型定义如下： 0：PC 设备 1：无人值守设备 2：手持终端设备 其它：保留
encryptflg 加密算法	8	AAC 为 REQ 指定的加密算法类型，取值如下： 0：SMS4 算法 ECB 模式 1：SM1 算法 CBC 模式 2：SMS4 算法 OFB 模式 3：AES 算法 CBC 模式 其它：保留

verifyflg 校验算法	8	AAC 为 REQ 指定的校验算法类型，取值如下： 1：HMAC-SHA256校验算法； 2：HMAC-SHA1校验算法； 其它：保留。
compressflg 压缩算法	8	AAC 为 REQ 指定的压缩算法类型，取值如下： 0：无压缩 1：压缩算法 其它：保留
flag 选项	8	每位定义一个功能选项： 0 位：是否进行动态 IP 分配 其它：保留
req_param 终端参数	变长	REQ 在注册请求时发送给 AAC 的自身参数，数据类型定义见 TISec 附加消息章节

7.2 认证 ASAP 参数信息

SPI （32位比特）	
tue_device （16字节）	
number （16位比特）	tue_type （16 位比特）
sockfd （32位比特）	
dst_addr （32位比特）	
src_addr （32位比特）	
dst_port （16位比特）	src_port （16位比特）
key （128位比特）	

表 5 TAI 注册完成隧道关联参数域定义

名称	长度（比特）	描述
SPI REQ 标识	32	REQ 唯一全局标识
tue_device 隧道设备名称	32	AAC 给 REQ 分配的隧道设备的名称
number 隧道编号	16	隧道编号
tunnel_type 隧道类型	16	隧道类型定义： 0：空类型 1：IP in IP隧道 2：IP in UDP隧道 其它：保留

sockfd 隧道Socket标识	32	AAC 为 REQ 建立的 Socket 文件描述符
dst_addr 隧道终端IP	32	REQ的隧道IP地址，AAC把封装好的隧道数据包发送给此IP地址。
src_addr 隧道服务器 IP	32	AAC端隧道IP地址地址。
dst_port 终端端口	16	REQ的隧道的UDP端口号，AAC把封装好的隧道数据包发送此终端的此UDP端口
src_port AAC 端口	16	AAC 上的隧道的 UDP 端口号
key 密钥	128	数据密钥初始密钥

7.3 隧道 TSAP 参数信息

隧道 TSAP 参数格式如下表：

peeraddr （32位比特）			
localhost （32位比特）			
peerport （16位比特）		localport （16位比特）	
flags （32位比特）			
cttl （8位比特）	encryptflgs （8位比特）	compressflgs （8位比特）	verifyflgs （8位比特）

隧道参数各个域的定义见表 6。

表 6　隧道参数域定义

名称	长度（比特）	描述
peeraddr 终端IP	32	REQ 发起 IP 地址，相对于 AAC 而言，peeraddr 为 TUE 隧道的对端地址
localhost AAC IP	32	AAC IP地址，相对于AAC而言，为TUE隧道的本端地址
peerport 终端端口	16	TUE 隧道使用 UDP 协议通讯，此为 REQ 隧道 UDP 端口号
localport AAC 端口	16	AAC 隧道 UDP 端口号
flags 选项	32	每比特位定义一个功能选项，定义如下： 0x0100：隧道是否被清除，停止工作 0x0200：初始密钥是否设置 0x0400：是否使用动态 IP 0x0800：设置需计算 UDP 校验和 其它位：保留
cttl	8	隧道数据包的生存时间，此值填充到隧道包 IP 报

隧道包生存时间		头的 TTL 字段
encryptflgs 加密算法	8	隧道的加密算法，定义以下值： 0：SMS4 算法 ECB 模式 1：SM1 算法 CBC 模式 2：SMS4 算法 OFB 模式 3：AES 算法 CBC 模式 其它：保留
compressflgs 压缩算法	8	隧道的数据压缩算法，定义以下值： 0：无压缩 1：压缩算法 其它：保留
verifyflgs 校验算法	8	隧道的校验算法，定义以下值： 1：HMAC-SHA256校验算法； 2：HMAC-SHA1校验算法； 其它：保留。

8 数据转交 TRD 和地址管理

数据转交技术TRD是TISec汇聚节点AAC作为数据转发节点采用的方法，通过代理机制AAC作为汇聚节点以代理者的身份接收、转发到不在同一物理链路的REQ数据，从而将REQ和汇聚节点所在网络域的不具备TISec功能的节点组成一张逻辑网络，数据转交技术分为两种，坞接转交（DR）和静默转交（SR）。

8.1 坞接转交 DR

坞接转交是指AAC作为汇聚节点通过主动代理内部网络域设备发往REQ的数据，从而完成数据转交过程中导向和处理的功能，主要用于IPv4网络中的数据转发，其处理过程如下：

1、TAI鉴别完成后REQ和AAC之间建立TUE隧道，此时AAC作为其网络域中的内部设备数据转交节点；

2、此时该汇聚节点AAC将以代理身份充当REQ节点代理者的身份，并接收本广播域内对REQ的数据请求；

3、当REQ节点或者广播域内非REQ节点需要同该IP地址建立通信时，广播域内会产生对IP地址的请求；

4、此时汇聚节点AAC代理并接收数据包，之后通过对应的REQ的TSAP对数据包进行TUE封装处理。

8.2 静默转交 SR

静默转交是指AAC作为汇聚节点不主动发起任何数据请求，仅对所有发往AAC的数据根据相应的TSAP进行封装处理，与坞接转交相比静默转交只做被动处理。

9 TUE 协议

TUE 协议用来保护 REQ 和 AAC 之间的 IP 数据安全，为 IP 数据提供机密性、数据源认证、无连接的完整性及抗重播服务，同时可用来构建虚拟专用网络。TUE 通过 TAI 协议产生的 TSAP 安全关联建立。图 19 为实施 TISec 后网络节点 IP 数据处理工作概要图示。

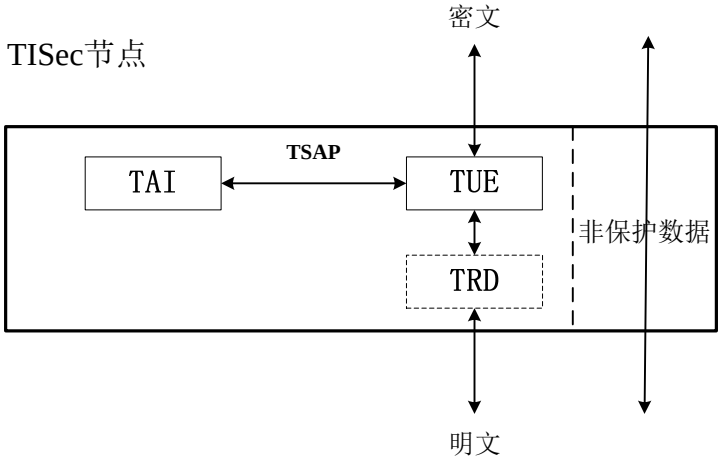


图 19 TISec 组件框架

图 19 中，TISec 节点包含了 TAI、TRD 和 TUE 组件，本标准中专指 REQ 和 AAC。该节点中 TAI、TRD 和 TUE 组件分别负责不同的处理逻辑。TISec 节点作为被保护 IP 数据的起止节点，对通过该设备的 IP 数据进行不同的处理。REQ 和 AAC 在完成 TAI 鉴别后建立 TSAP 安全关联，节点中 TUE 组件对通过的数据进行规则匹配，符合参数标示的数据通过 TUE 进行入栈和出栈处理，规则外数据则按非保护数据处理（本标准对此部分数据处理机制不做规定）。规则内被保护数据通过 TISec 节点时转换为密文，而不被保护数据则按照 TISec 节点默认规则处理，此节点类似策略转发设备，对通过数据进行处理，只有符合的数据包通过 TUE 组件处理。TISec 节点设备物理形态不影响基础 TISec 数据处理流程。

9.1 概述

本章定义了请求者REQ和汇聚节点AAC之间IP数据分组的封装协议TUE。

TUE 为 IP 数据提供机密性、数据源验证、抗重放以及数据完整性等安全服务。TUE 包含两个组件，加密器提供机密性，数据完整性和抗重放保护则由身份验证器提供。加密器和身份验证器两者所采用的算法类型则是由 TAI 鉴别协议协商产生的 TSAP 决定。隧道模式下数据报文加密时要将原始 IP 数据包（包括 IP 头及载荷）封装在另外一个 IP 数据报中，传输模式仅对 IP 数据载荷进行处理。为了更好的兼容 IPv4 网络的 NAT 设备，TUE 协议采用 IP-in-UDP 封装。TUE 工作时，将保护的 IP 数据报封装起来，并在原始数据之前添加一个新的 IP 头。新 IP 报头包含两端 TISec 节点的地址（或者为 TISec 节点前 NAT 设备地址）。

9.2 数据处理

TAI 鉴别完成建立 TUE 隧道，此时 TUE 数据处理模块分为出栈和入栈数据，出栈数据处理过程中，数据报进入网络层，网络层检索对应用户隧道 SPI，判断是否为其提供安全服务。隧道 SPI 检索的输出，以 TUE 工作于数据报文加密时为例，可能有下面这两种情况：

（1）不处理，即此时包不属于处理范围，直接交由系统协议栈处理。

（2）提供安全服务，在这种情况下，网络层会为载荷增添新的 IP 报头，并对原始数据进行加密重组处理。

入栈数据处理有别于出栈数据处理过程，当收到 IP 报后，如果报文内没有包含 TUE 数据标识，那么 TUE 数据处理模块就会对策略进行检查，判断该如何对这个包进行处理。并根据特定符字段来检索隧道 SPI。策略的输出可能是下述三种选择：忽略（送往系统协议栈）、提供控制数据服务或提供 TUE 服务。如果策略的输出是忽略，数据报就会直接递交给协议栈由系统自行处理；如果

是提供控制数据服务，则交由相应模块处理，否则进入 TUE 数据入栈处理。

如果 IP 报中包含 TUE 头，就会由 TUE 模块处理此包。TUE 模块会从 IP 数据报中提取出用户 SPI，并检索该用户关联参数，包的处理根据检索的参数进入不同的流程。

出栈数据情景下，TUE 头加在 IP 报前面，此时上层数据连同 IP 报头作为完整的数据字段加密后作为新 IP 的数据段，该字段前加入 IV 向量。IV 向量包括算法运算所需值并包含 TUE 隧道索引 SPI 字段。该值用以标示每包数据出栈的参数协商集合。

数据经由上层递交到网络层进入 TUE 模块，此时需要根据安全参数选择对应的加密容器，对包进行加密，并校验。在进行加密前加密容器根据算法要求会对数据进行填充。随后，TUE 将加密数据和验证数据重组放置到新的 IP 报中。对外出数据报进行处理的最后一步是重新计算位于新 IP 头的校验和，并根据最终 IP 报的大小和网络接口的 MTU 进行对比，如果大于 MTU 值则需要进行分片处理。

9.3 封装格式

按照IP安全的应用部署拓扑，TUE协议分为传输模式和隧道模式两种。传输模式用来保护IP数据包的载荷，不修改原始IP数据报头，这种封装方法应用在节点之间的数据保护，传输模式虽然会受到NAT阻隔，但在IPv6网络中可以很好的维持互联网端对端的应用模型。隧道模式是通过通信节点之间的安全设备对原始IP数据进行重新封装，在原始IP数据之前构造新的IP报头，这种封装方法应用在边界安全网关设备上，用以保护出口网络数据的安全性。

9.3.1 传输模式封装

TUE传输模式具体数据封装格式如图20所示：

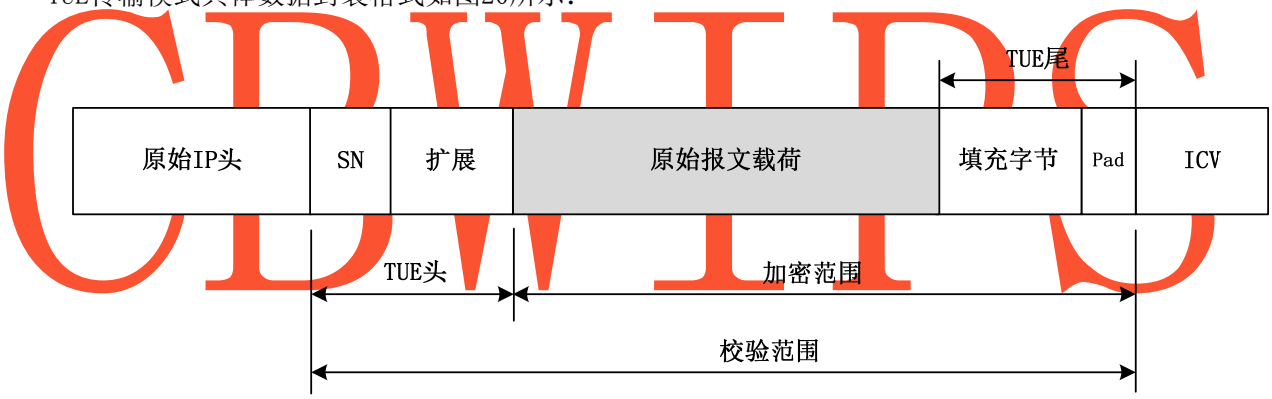


图 20 传输模式封装格式

图 21 中数据格式说明:

1. 原始报文载荷，指原始 IP 报文协议头之上的数据段。
2. 添充字节

SCB2/SM1/SMS4/AES 等对称算法，都要求待加密明文是 16 字节的整数倍，所以在加密之前，根据明文长度需要填充 0－15 个字节。计算填充字节时要将 Pad（填充）字节计算在内。

3. Pad 字节

Pad 字节标示 TUE 数据属性和扩展。

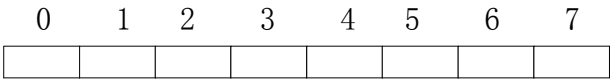


图 21 Pad 字节结构

（1） 0-3 比特为填充字节长度标示，如 0001 表示填充 1 个字节；

（2） 4-7 比特为类型码，组合如下：

0000－表示此数据包为保护数据

0100 表示此数据包为保护数据且为压缩数据

其余值为保留字段

4. ICV

（1）计算出的数据摘要值，在保证数据安全性和系统效率的同时取 hash 计算出的前 8 字节作为数据摘要；

（2）校验范围包括原始IP数据包、SN、IV、填充字节及Pad字节；

（3）hash算法标识通过SPI索引从TSAP中获取。

5. 扩展

扩展包括终端 4 字节 SPI、IV 字段和扩展字节。密文对于隧道数据的加密发送，因为创建了多条加密隧道，各个终端都从相应的隧道发送数据，所以发送加密的数据没有问题。对于隧道数据的接收解密过程，因为用一条隧道接收所有终端的数据，所以对于接收到的每一个数据包，需要标识出是哪一个终端的数据，进而才能正确的解密。在数据包头前附加终端的终端 SPI 作为密钥标示，从而查找相应密钥进行解密。为了实现密钥管理，需要在隧道模块中维护一个终端和它所对应的隧道的动态映射列表，隧道模块每一次启动时加入，每一次删除隧道时相应的删除。

6. SN

SN 序列号是一个独一无二的、单向递增的、并由 TISec 发送节点插在 TUE 数据中的一个整数。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 TUE 帧时，使用收到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。发送方的计数器和接收方的计数器在一个隧道 SAP 建立时被初始化为 0，使用给定隧道 SAP 发送的第一个分组的序列号为 1，如果激活抗重播服务，传送的序列号不允许循环，序列号使 TUE 具有了抵抗重播攻击的能力。

7. 原始IP/协议头

出栈数据情况下，TUE 将原始报文 IP 和协议头加在 TUE 处理过的密文数据前面，此时上层数据载荷字段加密后作为新 IP 的数据段，该字段前加入 IV 向量。IV 向量包括算法运算所需值并包含 TUE 隧道索引 SPI 字段。该值用以标示每包数据出栈的参数协商集合。

数据经由上层递交到网络层进入TUE模块，此时需要根据安全参数选择对应的加密容器，对包进行加密，并校验。在进行加密前加密容器根据算法要求会对数据进行填充。随后，TUE将加密数据和验证数据重组放置到新的IP报中。对外出数据报进行处理的最后一步是重新计算位于新IP头的校验和，并根据最终IP报的大小和网络接口的MTU进行对比，如果大于MTU值则需要进行分段处理

8. IPv6封装

考虑到IPv6网络最终实施的纯粹性，TUE协议仅规范原始IPv6数据的封装格式，如22所示。

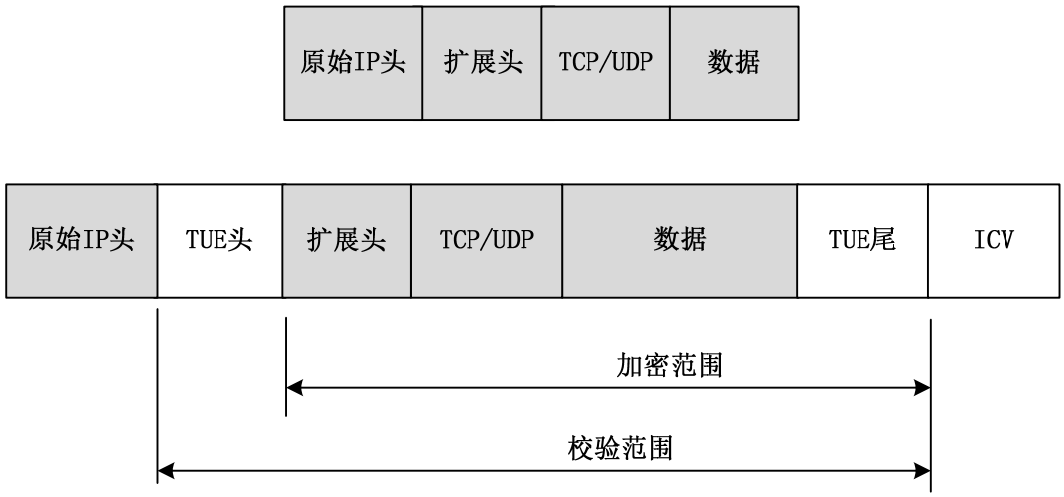


图22 传输模式IPv6封装

9.3.2 隧道模式封装

TUE 传输模式具体数据封装格式如图 23 所示：

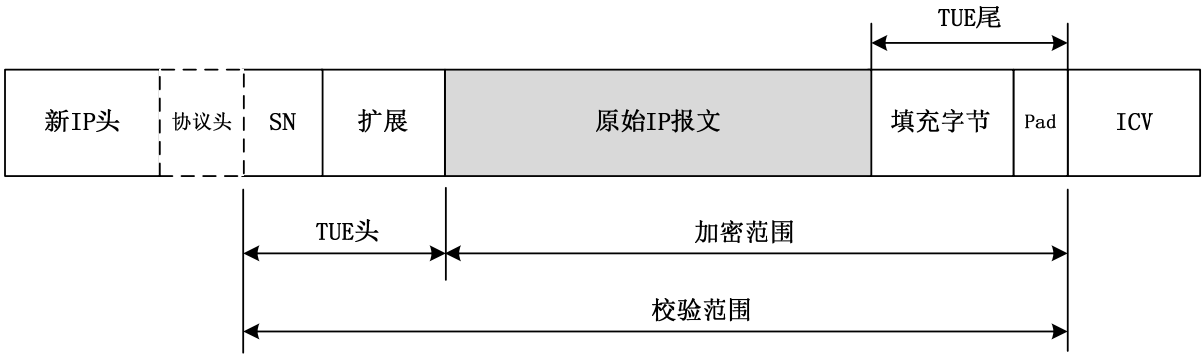


图23 隧道模式封装格式

图 23 中数据格式说明：

1. 原始 IP 报文，指待封装的 IP 报文，包含 IP 报头。
2. 添充字节

SCB2/SM1/SMS4/AES 等对称算法，都要求待加密明文是 16 字节的整数倍，所以在加密之前，根据明文长度需要填充 0－15 个字节。计算填充字节时要将 Pad（填充）字节计算在内。

3. Pad 字节

Pad 字节标示 TUE 数据属性和扩展。

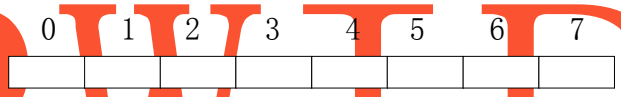


图 24 Pad 字节结构

（1） 0-3 比特为填充字节长度标示，如 0001 表示填充 1 个字节；

（2） 4-7 比特为类型码，组合如下：

0000－表示此数据包为保护数据

0100 表示此数据包为保护数据且为压缩数据

其余值为保留字段

4. ICV

（1）计算出的数据摘要值，在保证数据安全性和系统效率的同时取 hash 计算出的前 8 字节作为数据摘要；

（2）校验范围包括原始IP数据包、SN、IV、填充字节及Pad字节；

（3）hash算法标识通过SPI索引从隧道SAP中获取。

5. 扩展

扩展包括终端 4 字节 SPI、IV 字段和扩展字节。密文对于隧道数据的加密发送，因为创建了多条加密隧道，各个终端都从相应的隧道发送数据，所以发送加密的数据没有问题。对于隧道数据的接收解密过程，因为用一条隧道接收所有终端的数据，所以对于接收到的每一个数据包，需要标识出是哪一个终端的数据，进而才能正确的解密。在数据包头前附加终端的终端 SPI 作为密钥标示，从而查找相应密钥进行解密。为了实现密钥管理，需要在隧道模块中维护一个终端和它所对应的隧道的动态映射列表，隧道模块每一次启动时加入，每一次删除隧道时相应的删除。

6. SN

SN 序列号是一个独一无二的、单向递增的、并由 TISec 发送节点插在 TUE 数据中的一个整数。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 TUE 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。发送方的计数器和接收方的计数器在一个隧道 SAP 建立时被初始化为 0，使用给定隧道 SAP 发送的第一个分组的序列号为 1，如果激活抗重播服务，传送的序列号不允许循环，序列号使 TUE 具有

了抵抗重播攻击的能力。

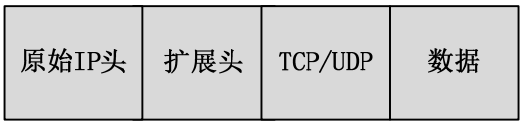
7. 原始 IP/协议头

出栈数据情景下，TUE 头加在 IP 报前面，此时上层数据连同 IP 报头作为完整的数据字段加密后作为新 IP 的数据段，该字段前加入 IV 向量。IV 向量包括算法运算所需值并包含 TUE 隧道索引 SPI 字段。该值用以标示每包数据出栈的参数协商集合。

数据经由上层递交到网络层进入TUE模块，此时需要根据安全参数选择对应的加密容器，对包进行加密，并校验。在进行加密前加密容器根据算法要求会对数据进行填充。随后，TUE将加密数据和验证数据重组放置到新的IP报中。对外出数据报进行处理的最后一步是重新计算位于新IP头的校验和，并根据最终IP报的大小和网络接口的MTU进行对比，如果大于MTU值则需要进行分段处理。

8. IPv6 封装

IPv6封装有两种封装格式，一种原始报文为IPv4，另一种为IPv6。这两种封装格式对应于TUE两个不同的处理逻辑。原始IPv4报文和IPv6报文封装格式分别对应图22和图23所示。



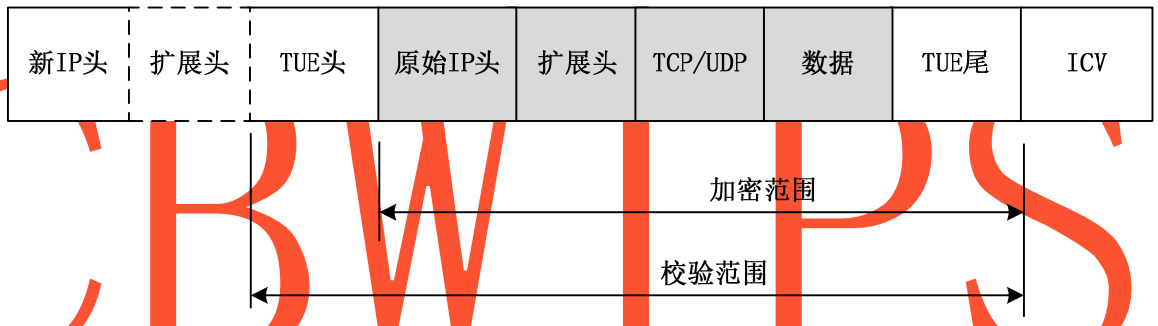


图25 隧道模式IPv6封装

附 录 A
（规范性附录）
引入在线可信第三方机制

A.1 概述

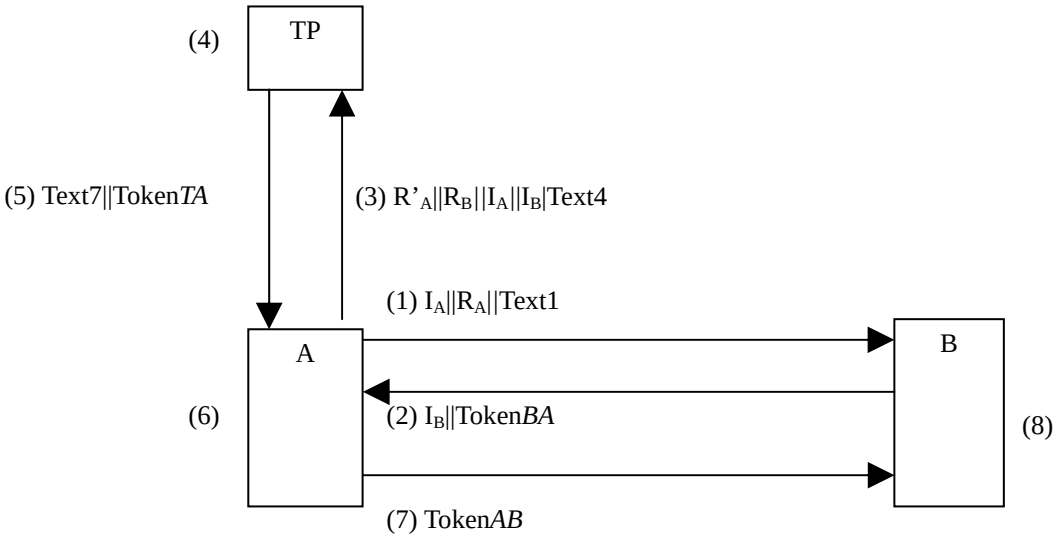
本条款中的鉴别机制要求两个实体A和B通过拥有或可证实A和B有效公钥的在线可信第三方（具有可区别的识别符），来验证对方的公钥。实体A和B拥有TP的有效公钥。

本条款描述了两个五次传递鉴别机制，在实体A和B之间实现了相互鉴别。在此鉴别机制中，有三个元素（A、B和TP），A和B相对TP来说是对等鉴别实体。这两个机制被统称为三元对等鉴别机制TePA（Tri-element Peer Authentication），它们使用ISO/IEC 14888或ISO/IEC 9796定义的签名机制。

本条款中的权标及文本字段说明同条款5.0。

A.2 五次传递鉴别TePA-A（由实体A发起）

在该鉴别机制中，唯一性/时效性通过产生和校验随机数来控制（见GB/T 15843.1的附录B）。该鉴别机制在图6中说明。



图A.1　五次传递鉴别 TePA-A（由实体 A 发起）

权标可以是下面的两种形式：

选项1：

$TokenAB = B||ResA||Text9||sS_T(R_B||ResA||Text5)||sS_A(A||R_A||R_B||B||sS_T(R_B||ResA||Text5)||Text8)$

$TokenBA = R_B||A||Text3||sS_B(B||R_B||A||R_A||Text2)$

$TokenTA = ResB||ResA||sS_T(R'_A||ResB||Text6)||sS_T(R_B||ResA||Text5)$

选项2：

$TokenAB = B||R'_A||Text9||TokenTA||sS_A(A||R_A||R_B||B||sS_T(R'_A||R_B||ResA||ResB||Text5)||Text8)$

$TokenBA = R_B||A||Text3||sS_B(B||R_B||A||R_A||Text2)$

$TokenTA = ResB||ResA||sS_T(R'_A||R_B||ResB||ResA||Text5)$

I$_A$、I$_B$、ResA、ResB、Status和Failure字段的值如下：

$I_A = A\ or\ CertA$

$I_B = B\ or\ CertB$

$ResA = (CertA||Status)、(A||P_A)\ or\ （I_A||Failure）$

$ResB = (CertB||Status)、(B||P_B)\ or\ （I_B||Failure）$

在这里，如果TP知道实体X（ X = {A, B}）的身份和公钥的映射，则I$_X$ = X；否则I$_X$ = CertX。如果X和CertX这两种表示方法都允许使用，那么TP应该可以通过其他机制区分这两种身份。ResX(X = {A, B})的值根据表1确定：

表 F.1 —Res*X*的值

域	选项 1	选项 2
I$_X$	X	CertX
ResX	(X||P$_X$) or （X||Failure）	(CertX||Status) or (CertX||Failure)

Status = True or False。如果证书是被撤销的，该字段的值是False；否则该字段的值是True。

Failure：当公钥或实体*X*的证书不能被TP获得，ResX将被设置为Failure。

该机制的执行过程如下：

(1) A 发送身份 I$_A$ 和随机数 R$_A$，可选文本 Text1 到 B。

(2) B 发送权标身份 I$_B$ 和 TokenBA 到 A。

(3) A 发送随机数 R'$_A$ 和 R$_B$，身份 I$_A$ 和 I$_B$ 以及可选项文本 Text4 到 TP。

(4) 收到来自 A 的信息后，如果 I$_A$ 是 A，则 TP 搜索 A 有效公钥；如果 I$_A$ 是 Cert，TP 检查 CertA

的有效性。如果 I_B 是 B，则 TP 搜索 B 有效公钥；如果 I_B 是 CertB，TP 检查 CertB 的有效性。然而，TP 校验证书的有效性应该被保护不受到拒绝服务攻击，提供该保护的机制描述超出了本规范的范围。

(5) TP 发送可选项文本 Text7 和权标 TokenTA 到 A。TokenTA 中的 ResA 和 ResB 是 A 和 B 的证书及其校验，或者是 A 和 B 的区别性识别符及其公钥。

(6) 收到来自 TP 的信息，A 完成下列步骤：

（i）通过以下方式验证 TokenTA：检验包含在权标 TokenTA 中 TP 的签名；校验步骤(3)中发送给 TP 的随机数 R'_A 与包含在签名数据 TokenTA 中的随机数 R'_A 相一致。

（ii）获得 B 的公钥，验证包含在权标 TokenBA 中 B 的签名。然后检查包含在 TokenBA 中的签名数据中的标识符字段(A)是否与 A 的区别标识符相一致，校验在步骤(1)中发送给 B 的随机数 R_A 与包含在 TokenBA 中的随机数 R_A 相一致。

(7) A 发送 TokenAB 到 B。

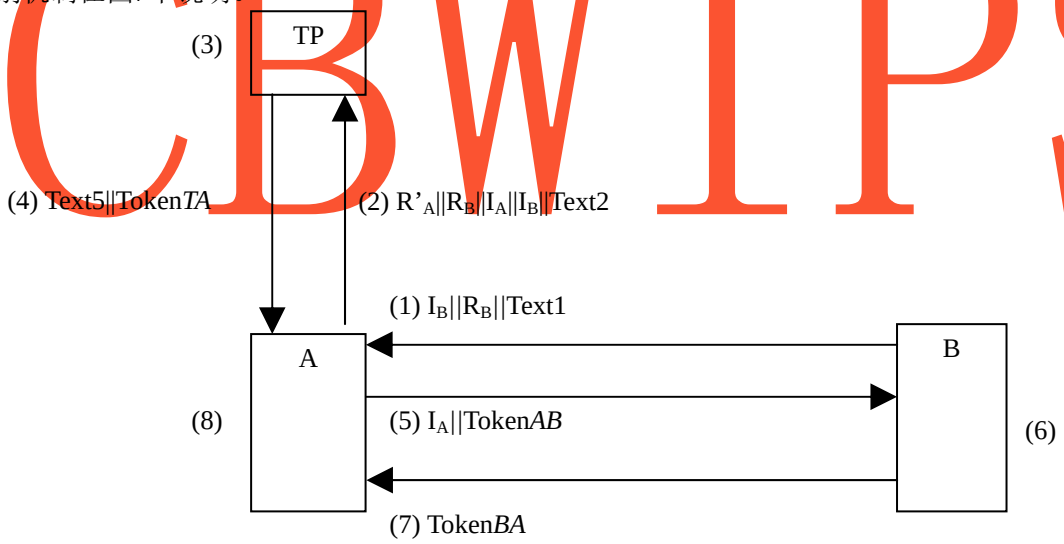
(8) 收到来自A的信息后，B执行下列步骤：

（i）通过以下方式验证 TokenTA：检验包含在权标 TokenTA 中 TP 的签名；校验步骤(2)中发送给 TP 的随机数 R_B 与包含在签名数据 TokenTA 中的随机数 R_B 相一致。

（ii）获得 A 的公钥，验证包含在权标 TokenAB 中 A 的签名。然后检查包含在 TokenAB 中的签名数据中的标识符字段(B)是否与 B 的区别标识符相一致，校验在步骤(2)中发送给 A 的随机数 R_B 与包含在 TokenAB 中的随机数 R_B 相一致。

A.3 五次传递鉴别TePA-B（由实体B发起）

在该鉴别机制中，唯一性/时效性通过产生和校验随机数来控制（见GB/T 15843.1的附录B）。该鉴别机制在图7中说明。



图A.2 五次传递鉴别 TePA-B（由实体 *B* 发起）

权标可以是下面的两种形式：

选项1：

$$TokenAB = R_A||B||ResA||Text7||sS_T(R_B||ResA||Text3)||sS_A(A||R_A||B||R_B||sS_T(R_B||ResA||Text3)||Text6)$$
$$TokenBA = A||Text9||sS_B(B||R_B||A||R_A||Text8)$$
$$TokenTA = ResB||ResA||sS_T(R'_A||ResB||Text4)||sS_T(R_B||ResA||Text3)$$

选项2：

$$TokenAB = R_A||B||R'_A||Text7||TokenTA||sS_A(A||R_A||B||R_B||sS_T(R'_A||R_B||ResA||ResB||Text3)||Text6)$$

$TokenBA = A||Text9||sS_B(B||R_B||A||R_A||\ Text8)$
$TokenTA = ResB||ResA||sS_T(R'_A||R_B||\ ResB||ResA||Text3)$

I_A、I_B、ResA、ResB、Status和Failure字段的值如下：
$I_A = A\ or\ CertA$
$I_B = B\ or\ CertB$
$ResA = (CertA||Status)$、$(A||P_A)\ or$ （$I_A||Failure$）
$ResB = (CertB||Status)$、$(B||P_B)\ or$ （$I_B||Failure$）

在这里，如果TP知道实体Y(Y = {A, B})的身份和公钥的映射，则I_Y = Y；否则I_Y = CertY。如果Y和CertY这两种表示方法都允许使用，那么TP应该可以通过其他机制区分这两种身份。ResY(Y = {A, B})的值根据表2确定：

表 F.2 —ResY 的值

域	选项 1	选项 2
I_Y	Y	CertY
ResY	(Y||P_Y) or (Y||Failure)	(CertY||Status) or (CertY||Failure)

Status = True or False。如果证书是被撤销的，该字段的值是False；否则该字段的值是True。

Failure：当公钥或实体Y的证书不能被TP获得，ResY将被设置为Failure。

该机制的执行过程如下：

(1) B 发送身份 I_B 和随机数 R_B，可选文本 Text1 到 A。

(2) A 发送随机数 R'_A 和 R_B，身份 I_A 和 I_B 以及可选项文本 Text2 到 TP。

(3) 收到来自 A 的信息后，如果 I_A 是 A，则 TP 搜索 A 有效公钥；如果 I_A 是 CertA，TP 检查 CertA 的有效性。如果 I_B 是 B，则 TP 搜索 B 有效公钥；如果 I_B 是 CertB，TP 检查 CertB 的有效性。然而，TP 校验证书的有效性应被保护不受到拒绝服务攻击，提供该保护的机制描述超出了本规范的范围。

(4) TP 发送可选项文本 Text5 和权标 TokenTA 到 A。TokenTA 中的 ResA 和 ResB 是 A 和 B 的证书及其校验，或者是 A 和 B 的区别性识别符及其公钥。

(5) 收到来自 TP 的信息后，A 发送权标身份 I_A 和 TokenAB 到 B。

(6) 收到来自A的信息后，B执行下列步骤：

（i）检验包含在权标 TokenAB 中 TP 的签名；校验步骤(1)中发送给 TP 的随机数 R_B 与包含在签名数据 TokenAB 中的随机数 R_B 相一致。

（ii）获得 A 的公钥，验证包含在权标 TokenAB 中 A 的签名。然后检查包含在 TokenAB 的签名数据中的标识符字段(B)是否与 B 的区别标识符相一致，校验在步骤(1)中发送给 A 的随机数 R_B 与包含在 TokenAB 中的随机数 R_B 相一致。

(7) B 发送 TokenBA 到 A。

(8) 收到来自 B 的信息，A 完成下列步骤：

（i）通过以下方式验证 TokenTA：检验包含在权标 TokenTA 中 TP 的签名；校验步骤(2)中发送给 TP 的随机数 R'_A 与包含在签名数据 TokenTA 中的随机数 R'_A 相一致。

（ii）获得 B 的公钥，验证包含在权标 TokenBA 中 B 的签名。然后检查包含在 TokenBA 的签名数据中的标识符字段(A)是否与 A 的区别标识符一致，校验在步骤(5)步中发送给 B 的随机数 RA 与包含在 TokenBA 中的随机数 RA 相一致。

参考文献

[1] GB/T 15843.3-2008/XG1　信息技术　安全技术　实体鉴别　第 3 部分：用非对称签名技术的机制　第 1 号修改单

CBWIPS