ICS 33 040 40
M 32

YD

中华人民共和国通信行业标准

YD/T 1701-2007

H.323 网络安全技术要求

Security Requirements of H.323 Network

2007-09-29 发布　　2008-01-01 实施

中华人民共和国信息产业部　发布

目　录

前　言……II
1　范围……1
2　规范性引用文件……1
3　定义……2
4　缩略语……4
5　约定……6
6　H.323 网络安全威胁……6
7　H.323 网络安全体系结构……7
8　终端注册安全技术要求……9
9　呼叫连接安全技术要求……12
10　呼叫控制（H.245）安全技术要求……15
10.1　简介……15
10.2　安全 H.245 通道操作……15
10.3　不安全 H.245 通道操作……16
10.4　安全能力交换……16
10.5　主角色……16
10.6　逻辑通道信令……16
10.7　H.245 消息认证与完整性……16
11　媒体流机密性技术要求……16
12　密钥管理安全技术要求……18
12.1　RAS 通道上的密钥管理……18
12.2　H.225.0 呼叫建立密钥管理……18
13　网守与 AAA 服务器间通信安全技术要求……18
附录 A（资料性附录）H.323 系统介绍……20
附录 B（规范性附录）H.235 ASN.1……23
附录 C（规范性附录）H.323 网络直接和选择路由呼叫安全……29
附录 D（规范性附录）基线安全轮廓……46
附录 E（规范性附录）采用本地 H.235/H.245 密钥管理的声音加密安全轮廓……56
附录 F（资料性附录）H.323 实现细节……79
附录 G（资料性附录）本标准章条编号与 ITU-T H.235 章条编号对照……82
参考文献……84

前　言

本标准的附录 B 等同采用 ITU-T H.235.0（2005）《H.323 安全性：H 系列（H.323 和其他基于 H.245 的）多媒体系统的安全性框架》附件 A；

本标准的附录 C 修改采用 ITU-T H.235.4（2005）《H.323 安全性：直接和选择性选路呼叫安全性》，主要差异为：删除了 ITU-T H.235.4（2005）的第 2 章、第 3 章、第 4 章、第 5 章；

本标准的附录 D 修改采用 ITU-T H.235.1（2005）《H.323 安全性：基线安全概要》，主要差异为：删除了 ITU-T H.235.1（2005）的第 2 章、第 3 章、第 4 章，对 ITU-T H.235.1（2005）的第 5 章做了修改；

本标准的附录 E 修改采用 ITU-T H.235.6（2005）《H.323 安全性：具有本地 H.235/H.245 密钥管理的话音加密概要》，主要差异为：删除了 ITU-T H.235.6（2005）的第 2 章、第 3 章、第 4 章、第 10 章和附录 I，对 ITU-T H.235.6（2005）的第 5 章做了修改；

本标准的附录 F 等同采用 ITU-T H.235.6（2005）附录 I.1。

ITU-T H.235v4（2005）分成 10 个部分，根据我国实际情况的需要，采用了其中的 4 个部分，具体的章节对应见附录 G。

本标准附录 A、附录 F、附录 G 是资料性附录。

本标准附录 B、附录 C、附录 D、附录 E 是规范性附录。

本标准由中国通信标准化协会提出并归口。

本标准起草单位：中兴通讯股份有限公司、信息产业部电信研究院、中国电信集团公司

本标准主要起草人：卢　忱、吴永明、落红卫、王　东、齐　勇、彭宏利、陈剑勇

H.323 网络安全技术要求

1 范围

本标准规定了 H.323 网络安全技术要求，主要包括到 H.323 网络实体间的连接建立，呼叫控制及媒体交换阶段中的所有通信通道安全，包括认证、机密性与完整性三个部分。

本标准适用了 H.323 网络。这里所指的 H.323 网络可包括局域网（Local Area Networks）、企业网（Enterprise Area Networks）、城域网（Metropolitan Area Networks）、内部网络（Intra-Networks）以及互联网络（Inter-Networks，包括 Internet），还包括采用基于分组传输（如 PPP）的 GSTN 或 ISDN 上的拔号连接或点对点连接。这些网络可由单个网段构成，也可以是由多个网段通过其他通信链路互连而成。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准。然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

ITU-T H.323	基于分组的多媒体通信系统
ITU-T H.225.0	基于分组的多媒体通信系统的呼叫信令协议和媒体流的打包
ITU-T H.245	多媒体通信的控制协议
ITU-T H.235v1（1998）	H.323安全性：H系列（H.323和其他基于H.245）多媒体系统（版本1）
ITU-T H.235v2（2000）	H.323安全性：H系列（H.323和其他基于H.245）多媒体系统（版本2）
ITU-T H.235v3（2003）	H.323安全性：H系列（H.323和其他基于H.245）多媒体系统（版本3）
ITU-T H.235.0（2005）	H.323安全性：H系列（H.323和其他基于H.245）多媒体系统的安全框架
ITU-T H.235.3（2005）	混合安全轮廓
ITU-T Q.931	ISDN用户网络接口基本呼叫控制3层规范
ITU-T X.509	公开密钥和属性证书框架
ITU-T X.800	CCITT应用开放系统互连安全体系架构
ITU-T X.803（1994）	信息技术—开放系统互联—上层安全模型
ITU-T X.810	信息技术—开放系统互联—开放系统安全架构：总览
ITU-T X.811	信息技术—开放系统互联—开放系统安全架构：认证架构
ISO/IEC 9797	信息技术—安全技术—使用基于块密码算法密码控制功能的数据完整性机制
ISO/IEC 9798-2	信息技术—安全技术—实体认证——第3部分：对称加密算法机制
ISO/IEC 9798-3	信息技术—安全技术—实体认证——第3部分：数字签名技术机制
ISO/IEC 9798-4	信息技术—安全技术—实体认证——第4部分：使用密码校验函数认证
ISO/IEC 10116	信息处理—安全技术——*n*比特块密码算法运算模式
ISO/IEC 10118-3	信息技术—安全技术—散列函数 第3部分:专用散列函数

IETF RFC2045	多用途因特网邮件扩展（MIME）第1部分：因特网消息体格式
IETF RFC2104	HMAC：用于消息认证的密钥散列
IETF RFC2138（1997）	RADIUS协议
IETF RFC2198	用于冗余音频数据的RTP载荷
IETF RFC2246（1999）	TLS协议版本1.0
IETF RFC2401	因特网协议安全体系结构
IETF RFC2405	IP封装安全载荷（ESP）
IETF RFC2412	OAKLEY密钥确定协议
IETF RFC2833	用于DTFM数字的RTP载荷，电话音调与技术
IETF RFC3280（2002）	Internet X.509公钥证书与撤消列表轮廓（CRL）
IETF RFC3546	传输层安全扩展
IETF RFC3550	RTP：实时应用的传输层协议
IETF RFC3830	MIKEY：多媒体因特网密钥

3 定义

ITU-T H.323、ITU-T H.225.0、ITU-T H.245、ITU-T H.235.0 和 ITU-T X.800、ITU-T X.803、ITU-T X.810、ITU-T X.811 确立的以及下列定义适用于本标准。

GK

指H.323网中网守，提供地址翻译、接入控制、带宽管理功能。

MCU

多点控制单元，是网络中一个功能实体，提供3个或3个以上的终端或网关参加多点会议的功能，与各端点间进行能力协商，音、视频集中处理（切换和混合）和会议控制等功能。

安全策略

提供安全服务的一系列准则。

安全服务

该服务保证了系统和数据传输的足够的安全性。

安全轮廓

在一个特定的方案中，为保证 H.323 多媒体通信安全，并达到一致性与互操作性，所定义的集合（或子集）。

被动威胁

在不改变系统状态的情况下对非授权的暴露信息的威胁。

对等实体认证

确认对等实体是所声称的实体。

访问控制

防止未授权的使用资源，包括用未授权的方式使用资源。

否认

对于全部或者部分参与通信的实体，有一个实体否认。

公钥证书

含有代表拥有者公钥的证书及其他一些可选信息，用来进行验证。公钥证书是由可信任第三方以不可伪造方式签发。

共享秘密

通信双方所共知的秘密。

机密性

信息对未授权的个体、实体或者过程（程序）的不可用或不可泄露。

加密

对数据进行密码转换以产生密文。加密可能是不可逆的，这种情况下没有相应的解密过程。

解密

将密文恢复成明文的过程。

拒绝服务

授权用户对资源的不可访问或是操作延迟。

可用性

一个被授权的实体的可接入性和可用性。

口令

保密的认证信息，通常由一串字符组成。

媒体流

可以是音频、视频及数据或它们的组合。媒体流数据携有用户或应用数据，但不含有控制数据。

密文

加密后产生的数据。该数据是没有语义的。

密钥

一串符号序列用来控制加密与解密操作。

密钥管理

与一安全策略对应的密钥的产生、存储、分发、删除、存档和密钥申请。

明文

可理解的有语义的数据。

认证

参看数据源认证和对等实体认证。

认证信息

用来证实所声称身份合法的信息。

授权

授权，包括基于访问权限的授权访问。

数据完整性

数据没有以非授权方式被改变和毁坏的特性。

数据源认证

确定接收数据的源端是发送端所声称的。

数字签名

附加的数据或者是对一数据单元的密码转换，可实现认证、完整性与不可否认性。

威胁

对安全潜在的侵犯。

通道

信息传输路径。

隐私性

个体对与他们相关信息的控制和保密权利，这些信息可能被别人存储、收集或暴露。

证书

由一个安全权威机构或可信第三方发布的安全相关的数据。与安全信息一起用于提供完整性和数据源认证。

主动威胁

未经授权、故意改变系统状态的威胁，包括篡改、重放、伪造、冒充一个授权实体和拒绝服务。

4 缩略语

下列缩略语适用于本标准。

3DES	Triple DES	三重DES
ACF	Admission Confirm	接入确认
ARJ	Admission Reject	接入拒绝
ARQ	Admission Request	接入请求
AES	Advanced Encryption Algorithm	高级加密算法
ASN.1	Abstract Syntax Notation No.1	抽象语法符号
CA	Certificate Authority	证书权威机构
CBC	Cipher Block Chaining	密码块链
CFB	Cipher Feedback Mode	密码反馈模式
CRL	Certificate Revocation List	证书撤消列表
DES	Data Encryption Standard	数据加密标准
DH	Diffie-Hellman	一种公钥体制，实现密钥交换算法
DNS	Domain Name System	域名系统
DRC	Direct Route Call	直接选路呼叫
DTMF	Dual Tone Multi Frequency	双音多频
ECB	Electronic Code Book Mode	电子代码本模式
EP	End Point	端点
GCF	Gatekeeper Confirm	网守发现确认
GK	Gatekeeper	网守
GKID	Gatekeeper Identifier	网守标识符
GRJ	Gatekeeper Reject	网守发现拒绝
GRQ	Gatekeeper Request	网守发现请求
GW	Gateway	网关

HMAC	Hash Message Authentication Code	散列消息认证码
ID	Identifier	标识符
IPSec	Internet Protocol Security	互联网协议安全
IV	Initial Vector	初始化矢量
LCF	Location Confirm	位置确认
LRJ	Location Reject	地址解析拒绝
LRQ	Location Request	地址解析请求
MAC	Message Authentication Code	消息认证码
MC	MultiPoint Control	多点控制器
MCU	Multipoint Control Unit	多点控制单元
MD5	Message Digest No. 5	消息摘要算法
MPS	Multiple Payload Stream	多重载荷流
NAT	Network Address Translation	网络地址转换
OFB	Output Feedback Mode	输出反馈模式
OID	Object Identifier	对象标识符
OLC	Open Logic Channel	开放逻辑信道
PDU	Protocol Data Unit	协议数据单元
PKCS	Public-Key Crypto System	公钥密码系统
PKI	Public Key Infrastructure	公钥密码体制
PRF	Pseudo Random Function	伪随机函数
RAS	Registration，Admission and Status	注册、接入和状态
ROC	Read Out Counter	读出计数器
RSA	Rivest，Shamir and Adleman（public key algorithm）	著名公钥算法
RTP	Realtime Transport Protocol	实时传输协议
RTCP	Realtime Transport Control Protocol	实时传输控制协议
SDU	Service Data Unit	服务数据单元
SEQ	SEQuence	序列号
SHA1	Secure Hash Algorithm No.1	安全散列算法
SRTP	Secure Real Time Transport Protocol	安全实时传输协议
SSL	Secure Socket Layer	安全套接字
TCP	Transport Control Protocol	传输控制协议
TLS	Transport Level Security	传输层安全
TSAP	Transport Service Access Point	传输服务访问点
UDP	User Datagram Protocol	用户数据报协议
VoIP	Voice over Internet Protocol	网际协议上的语音
XOR	Exclusive OR	异或运算

5 约定

在文本中，对象标识符通过符号引用（例如，“I11”）。本标准定义了各种对象标识符（OID）用来表示信令安全能力、规程或安全算法。这些OID与分层的数值树相关，可以是由外部组织分配的或是由ITU-T维护的OID树的一部分。与ITU-T H.235相关的OID在文本中有下面的表示形式：

"OID" = {itu-t（0）recommendation（0）h（8）235 version（0）**V N**}，其中 **V** 代表单个十进制数，表示ITU-T H.235标准的版本，例如1、2、3、4。*N* 代表单个十进制数，惟一区分出OID实例，由此标识不同的规程、算法或安全能力。

因此，ASN.1编码的OID由数字序列组成。为方便起见，文本中每一个OID都使用便于记忆的速记文本串表示。每个OID串和ASN.1数字序列之间存在映射关系。符合本标准的实现应该仅使用ASN.1编码的数字。

6 H.323网络安全威胁

目前，依托专网及互联网部署了大量具有多媒体能力基于ITU-T H.323协议的终端，开展基于分组网（固定与3G移动）语音（VoIP）、视频等业务及其他一些增值业务，并有可能在未来成为用户接入的主流方式。由于互联网本身的开放性和缺乏有效监控，安全问题日益凸现，其主要安全威胁有以下几个方面。

（1）拒绝服务攻击

基于开放端口的拒绝服务（DoS）攻击。从网络攻击的方法和产生的破坏效果来看，DoS算是一种既简单又有效的攻击方式。攻击者向服务器发送相当多数量的带有虚假地址的服务请求，但因为所包含的回复地址是虚假的，服务器将等不到回传的消息，直至所有资源被耗尽。其中包括：

- 耗费系统资源的DoS攻击；
- 大量服务请求式DoS攻击；
- 利用系统漏洞式的DoS攻击；
- 对VoIP网关等关键设备进行SYN或ICMP数据包的大流量攻击，以致通信中断，无法正常提供业务。

（2）服务窃取

主要是针对非授权接入。其中包括：

- 窃取合法用户身份，假冒合法用户身份；
- 冒充合法网络节点进行相应欺骗。

（3）信令流监听

由于H.323控制信令的开放性。任何人通过网络监听器都可以监听H.323信令流。恶意用户拦截并篡改建立呼叫后传输的数据包，修改数据流中的域，使H.323呼叫不能正常使用，从而引入以下威胁。

- 会话劫持：结合嗅探以及欺骗技术在内的攻击手段。攻击者作为第三方参与到正常的会话当中，然后可以插入恶意数据进行会话监听，甚至可以代替某一方接管会话。
- 中间人攻击：使用ARP欺骗或DNS欺骗，将会话双方的通信流暗中改变，而这种改变对于会话双方来说是完全透明的。
- 电话跟踪：通过对通话信令数据包的嗅探和协议分析，提取目标电话状态信息。在一定时候可以劫持该通话链路或者随时终止该会话。

（4）媒体流的监听

H.323系统中RTP/RTCP是在基于分组的网络上传输等时话音信息的协议。由于协议本身是开放的，恶意用户可以通过网络监听器监听媒体流，从而可以引入如下威胁。

- 信息失窃：在 H.323 中的通信内容被未授权用户窃取。
- 重放攻击：即使是一小段媒体流都可以被重放出来而不需要前后信息关联。可能会有人在数据网络上通过网络监听器监听所有信息并重放。

7 H.323 网络安全体系结构

7.1 H.323 网络安全范围

图1所示为H.323网络安全范围，其中阴影部分是本标准所涉及的规定范围。

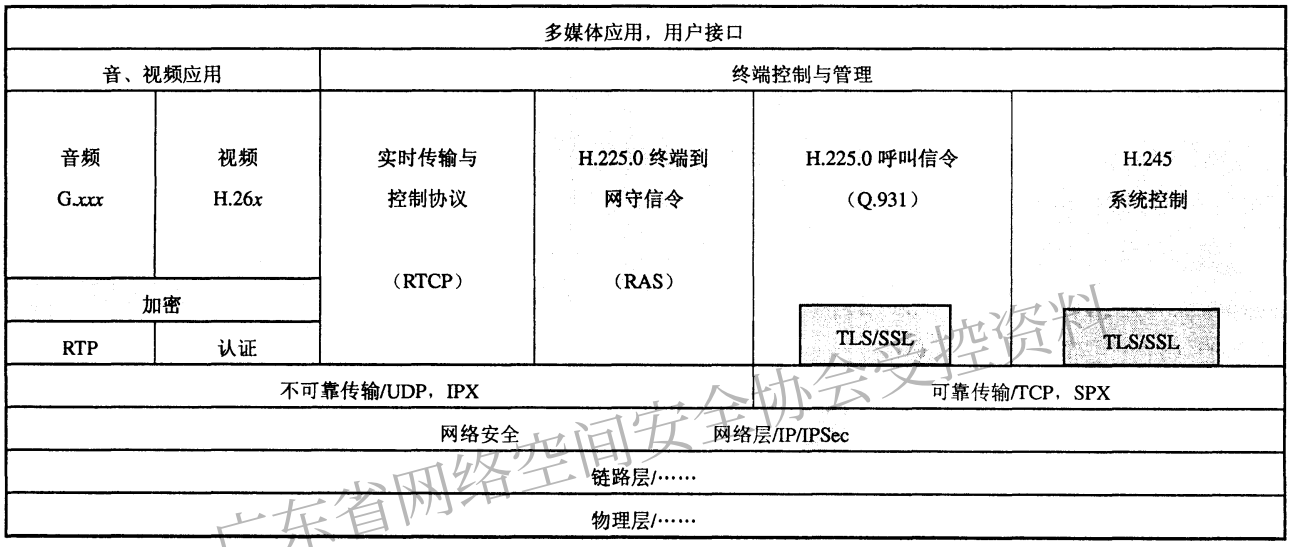


图 1 H.323 网络安全范围示意

H.323 系统中，H.225.0 和 H.245 是 H.323 系统的核心协议。H.225.0 主要包括 RAS 和呼叫信令协议。RAS 主要用于传送终端登记、认证和呼叫状态信息。呼叫信令协议主要是基于 ITU-T Q.931 制定的，用于完成呼叫建立过程。H.245 是通用多媒体通信控制协议，主要实现媒体流通道的打开、维护和关闭。H.245 控制信号在一条专门的可靠通道（如 TCP）上传送，称为 H.245 控制通道。RTP/RTCP 是媒体流实时传输/实时传输控制协议，媒体流安全传输将使用 H.245 通道中给出的算法与密钥进行加密和解密。

H.323 端点之间建立通信关系一般执行三个控制过程：呼叫接纳（RAS）、呼叫控制（H.225.0 呼叫信令协议）与连接控制（H.245）。

要实现安全的 H.323 系统业务，首先要保证终端或 MCU 与网守之间安全传递 RAS 消息，以完成安全注册，确保只有合法用户能够使用 H.323 业务并进行相应的资源使用授权，如国际、长途业务授权等。在保证 RAS 安全基础上，能够建立安全的呼叫连接通道（H.225.0）与呼叫控制（H.245）通道，在此基础上，协商 RTP 媒体流通信的加密算法和密钥，完成媒体流通信机密性。

7.2 H.323 网络安全机制

7.2.1 安全机制

实现 H.323 网络安全，有两种主要安全机制：

- 通过传输层/网络层安全机制，如 TLS 或 IPSec 技术，实现 RAS 信令通道、H.225.0 呼叫信令通道及 H.245 信令通道保护。这种情况下，能够不增加 H.323 协议簇安全保障机制，如认证、完整性等，对

各种信令通道进行操作。

- 通过对 H.323 协议簇中所涉及的信令本身增加安全机制，实现各种通道安全能力协商与安全保护。

实际实现 H.323 网络安全时，宜根据具体网络应用环境，组合使用以上两种安全机制。

7.2.2 安全认证

应用层安全认证方法有基于公钥密码的数字证书和基于对称密码的预共享秘密两种。

（1）公钥证书方法：借助于公正的第三方证书权威机构（CA），为用户生成包含有用户公钥的数字证书，并通过证书服务器供通信双方使用。与数字证书内公钥相对应的私钥，则由用户秘密保存。通过交换数字证书实现的数字签名，既能够完成认证，也能够防止中间人攻击。数字签名本身并不自动证明谁是接收者，要实现证书授权，需要与证书内其他内容有关的一些策略，例如，包括证书颁发者身份及其所规定的用户账号标识等。

（2）共享密钥方法：这种认证方法需要通信双方事先协商，如在客户预订业务时，获得一个共享密钥。

对等实体可以支持双向和单向认证，这种认证可以发生在某些或全部通信通道。

7.2.3 媒体流机密性

实现媒体机密性，需要提供安全控制通道，以建立密钥交换（会话密钥）；同时，需要协商安全算法以便对该密钥实施保护。基于分组传输的媒体流机密性，依赖于 H.245 逻辑通道特征，这些通道可以是单向的（针对逻辑通道而言）。

在逻辑通道上所携带的机密（加密）数据是建立在端到端加密的基础上。

7.2.4 信任单元

信任单元指被终端信任的网络单元，如 MCU、网守等。这些网络单元知道通信终端的安全信息。只有在信任单元之间的连接得到保护，不受中间人攻击的情况下，才能保证终端之间的通信安全。

7.2.5 可信第三方密钥托管

在 H.323 网络中，安全实体在信令元中宜支持信任第三方（TTP）能力。

7.2.6 安全轮廓表

H.323 系统中不同环境与应用下的安全机制及用法等安全要求，借助安全轮廓表形式进行说明，见表 1。

表1总结描述了实现H.323网络安全解决方案时构造的不同密码机制与算法。能够基于口令组合散列算法、口令组合加密算法及数字签名三种方法之一来构造一个实际应用的用户安全轮廓。其中口令散列及口令加密安全机制是基于对称密钥密码实现H.323网络安全。安全轮廓用于管理终端之间（终端—网守、网守—网守、网关—网守）分配对称密钥/口令，包括安全简单电话终端（安全语音简单端点类型）、多点控制单元（MCU）等。

表 1 给出的一般安全轮廓表，能够处理不同的安全需求。其中垂直阴影部分表示基本安全需求，水平阴影部分表示媒体流的加密。“认证/完整性”是指认证、完整性、身份认证+完整性三种情况。

对于一个具体应用的安全轮廓所描述的功能，宜根据实际环境与需要有选择地或全部实现。对于符合 ITU-T H.235 安全终端，应该通过信令消息中安全相关域（fields）内的对象标识符，指定要配置什么样的安全轮廓。

在实际应用中，终端在最初的 RRQ/GRQ 消息内，可以同时提供多个安全轮廓，而让网守选择最匹配的一个，并由 RCF/GCF 消息回答响应。网守之间 LRQ/LCF 事务也可以携带几个安全轮廓。

表1 安全轮廓表

安全服务	调用功能（算法）			
	RAS	H.225.0	H.245[a]	RTP
认证/完整性	口令散列 口令加密 数字签名	口令散列 口令加密 数字签名	口令散列 口令加密 数字签名	
不可抵赖	数字签名	数字签名	数字签名	
机密性				DES算法、3DES算法、AES算法 CBC模式或EOFB模式
访问控制				
密钥管理	用户口令分配证书分配	用户口令分配证书分配	认证Diffie-Hellman密钥交换	集成的密钥管理（分配、更新）密钥保护算法可以有DES、3DES、AES）

[a] 在H.225.0快速连接内，隧道H.245或嵌入H.245

8 终端注册安全技术要求

8.1 概述

终端注册安全主要体现在身份认证与完整性方面，不包括网守与端点之间的消息保密。存在以下三种认证方法：

- 口令+对称加密算法；
- 口令+散列（Hash）算法；
- 证书+数字签名算法。

以上三种认证方法，既能实现单向认证（终端向网守），也能实现双向认证。每一种认证方法既能够使用基于时间戳的二次握手（two pass）协议，也能够使用挑战/响应的三次握手（three pass）协议。对于时间戳机制，终端与网守之间应该有一个可接受的时间基准（导出时间戳用）。可接受时间偏差数由本地具体实现所考虑。挑战/响应协议使用一个随机生成的、不可预测的挑战数作为来自于认证者的挑战。

每一种认证方法，要求终端与网守的标识符都是可知的。时间戳认证机制应该精细调整时间粒度，防止消息重放攻击。例如，如果时戳仅仅按分钟增加，则在端点“A”已经发送一个消息给端点“B”时，在1min以内，另一个端点C可以探测到端点A，从而实施重放攻击。

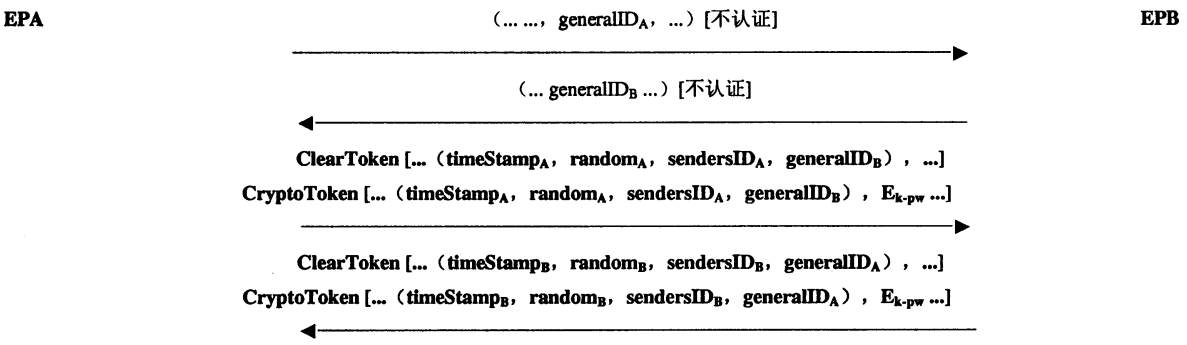
另外，如果消息是多播的，则不对该消息进行安全认证。

8.2 认证协议

8.2.1 基于口令对称加密算法的认证

假定一个终端标识符与相关口令已在开户期间进行了交换。以下几个小节认证规程图中：EPA（实体A）代表终端，EPB（实体B）代表网守。

图2中（a）与（b）参照ISO/IEC 9798-2中的5.2.1节与5.2.2节内容，分别描述了二次握手与三次握手协议涉及的令牌格式及消息交换。



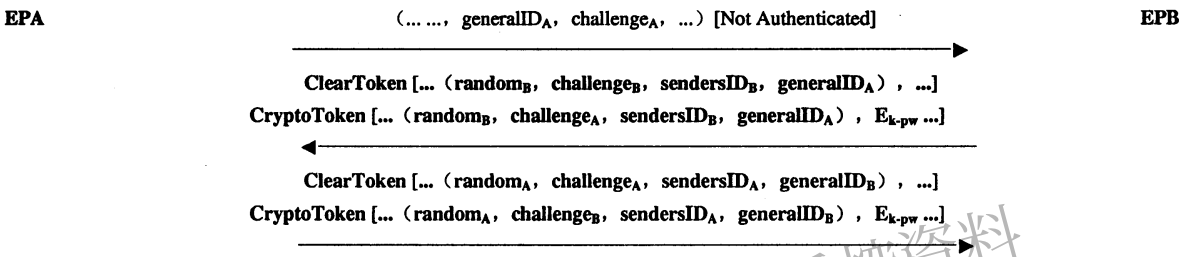
注：（1）EPB返回令牌（token）是可选的，如果不返回，则仅完成单向认证。

（2）E_{k-pw} 指示由口令导出的密钥进行加密时的值。

（3）**random**是单调增加计数器，以保证带有相同时间戳的多个消息惟一。

（4）第三个消息内，EPA提供一个单独的**ClearToken**，其内的OID与**CryptoToken**内的OID相同，对于第四个消息也是一样

（a）二次握手



注：（1）EPA挑战数**challenge**$_A$ 与返回的EPB的加密令牌**CryptoToken**不是必需的，如果不返回，则仅完成网守对终端进行的单向认证。

（2）E_{k-pw} 指示由口令导出的密钥进行加密时的值。

（3）第三个消息内，EPA提供一个新的明文挑战数**challenge**$_A$ 并放入单独**ClearToken**内，其内的OID与**CryptoToken**内的OID相同。EPA在紧接的消息中也返回**challenge**$_B$作为挑战应答；与第二个消息相似，反之亦然。

（4）对于多个正在完成的消息**random**（即一个单调增加计数器）必须使一个挑战惟一

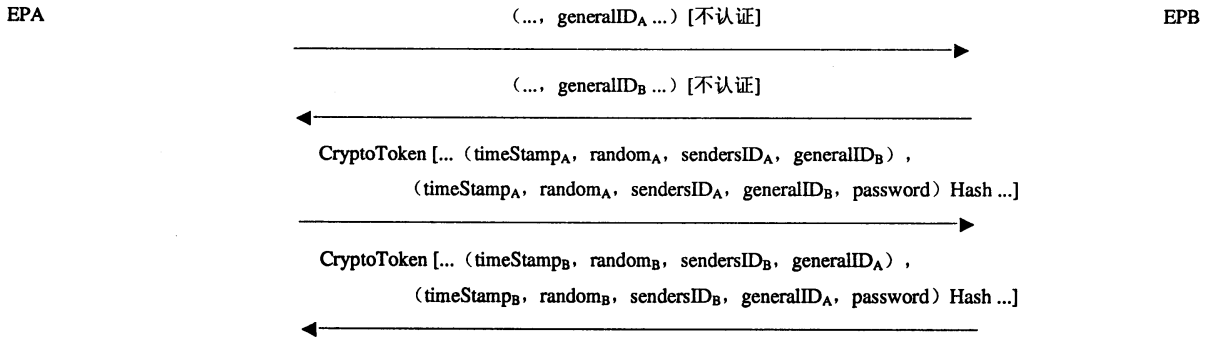
（b）三次握手

图2　基于口令对称加密认证协议

图中具体结构与字段实现可参考ITU-T H.225.0与ITU-T H.235.0。

8.2.2　基于口令散列（Hash）算法的认证

图3中（a）与（b）参照 ISO/IEC 9798-4 中的 5.2.1 与 5.2.2 节内容。基于口令+散列算法描述了二次握手或三次握手认证所需要的令牌（token）格式与消息交换。

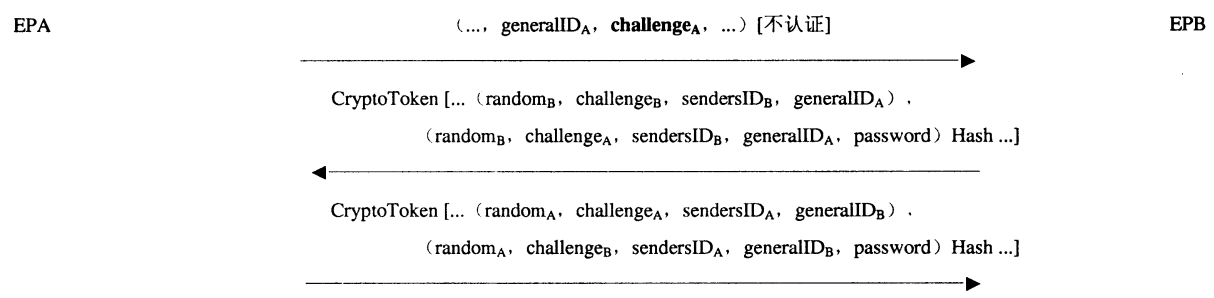


注：（1）EPB返回的标记（token）是可选的；如果不返回，只完成单向认证。

（2）**Hash**指示使用一个散列函数对所包含的值进行运算。

（3）**random**是一个单调增加计数器，使具有相同时间戳的多个消息保持惟一

（a）二次握手



注：（1）EPB返回的标记（token）是可选的；如果不返回，只完成单向认证。

（2）**Hash**指示一个Hashing函数对这个所包含的值上进行运算。

（3）第三个消息中，EPA提供一个新的**Challenge$_A$**，并以明文方式放入到**cryptoHashedToken**内**ClearToken**中，EPA也返回经过散列的**challenge$_B$**作为响应。类似于第二个消息，反之亦然。

（4）对于多个待处理信息**random**（即一个单调递增计数器）用来使挑战保持惟一

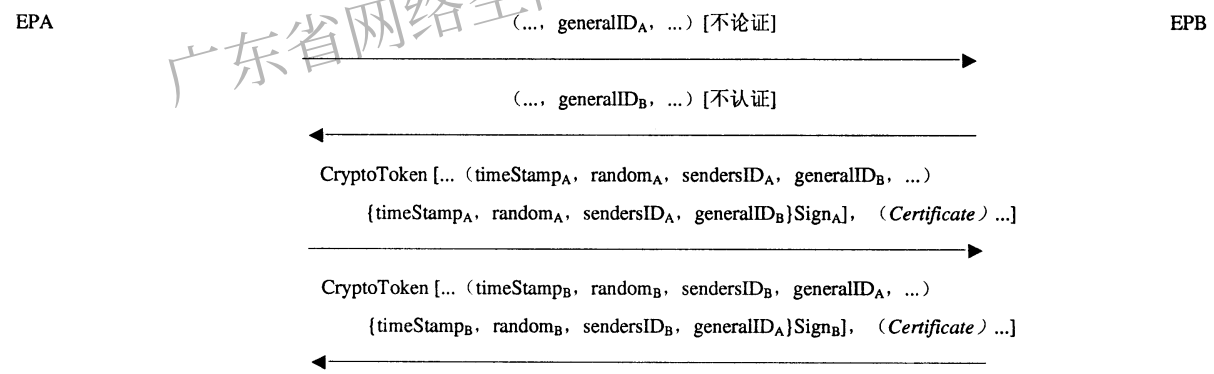
（b）三次握手

图3　散列算法基于口令认证

在图3（a）中的CryptoToken结构，被用于在RAS消息交换中传递所使用的明文认证参数，如时间戳、终端标识符、网守标识符等。通过预共享口令，使用散列函数对这些认证参数进行计算时，完成认证/完整性检查。具体结构与字段实现可参见附录D。

8.2.3 基于证书数字签名的认证

图4中（a）与（b）参照ISO/IEC 9798-3的第5.2.1节内容，描述了基于证书+数字签名认证所需要的令牌（token）格式与消息交换。假定终端与网守标识符及其关联的证书在用户申请业务开通期间被分配/交换。



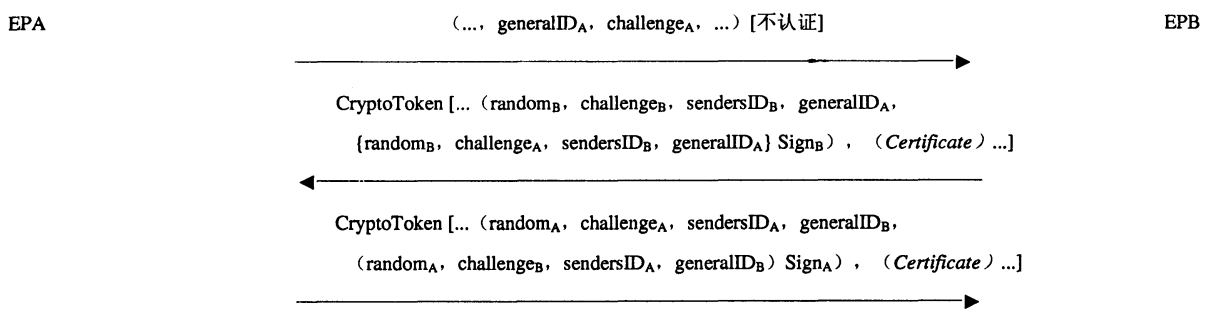
注：（1）EPB返回token是可选的；如果不返回，则仅完成单向认证。

（2）可以将第三方“支付”类型证书包含在EPA内（Certificate）。

（3）**Sign**指示签名（与证书相关）在所包含的值上完成。

（4）**random**是一个单调增加计数器，保证多个相同时间戳的消息惟一

（a）二次握手



注：（1）EPB返回token是可选的；如果不返回，则仅完成单向认证。

（2）可以将第三方"支付"类型证书包含在EPA内（Certificate）。

（3）**Sign**指示签名（与证书相关）在所包含的值上完成。

（4）第三个消息中，EPA提供一个新的**challenge$_A$**，以明文形式放入**GeneralToken**内。EPA也返回签名的**challenge$_B$**作为响应；类似于第二个消息，反之亦然。

（5）对于多个待完成的消息，**random**（单调增加计数器）必须使一个挑战惟一

（b）三次握手

图4 基于证书数字签名认证

如果消息是多播的，则目的实体标识符（在EPA消息源为generalID$_B$，反之亦然）不应该放在ClearToken相应域中。

8.2.4 认证判别依据

基于以上几种认证方法，接收实体或终端可以基于几个标准对被认证实体或终端进行验证：

- Timestamp 是否处于活跃度内，random 是否惟一。
- 挑战/响应是否一致。
- generalID 身份与自身标识符是否相等。
- 核实 Diffie-Hellman 参数，例如测试 1024bit 素数与生成元是否正确。Diffie-Hellman 参数是否安全的测试是一个耗时的处理，只有当本地策略需要时才进行。
- 被认证者的认证值是否与自身验证计算的结果相匹配。

8.3 共享秘密与口令的使用

使用对称密码技术来达到认证、完整性及机密性目的时，术语口令与共享秘密具有相同意义。共享秘密在用户业务开通过程中进行分配或配置，或者通过信令消息交换过程计算得到，例如使用Diffie-Hellman算法导出共享秘密。

口令可看作是用户可以记住的字母和/或数字字符串。口令应该是不可猜测的，并且要定期修改，以提供足够的安全。应该使用强单向散列函数将口令转换为一个固定长度的字符串来作为共享秘密。

9 呼叫连接安全技术要求

9.1 简介

呼叫连接安全涉及两个方面：一是在接收呼叫之前要进行认证，以保证呼叫建立与连接通道安全（如H.225.0）；二是通过对端点的认证来进行呼叫授权。本章主要考虑呼叫建立与呼叫连接通道的安全认证，安全授权方面要求在第13章中规定。

呼叫连接安全存在4种方法：

（1）利用独立的安全协议实现呼叫连接安全。在交换呼叫连接信令消息之前，可以通过在一个安全的众所周知端口上（H.225.0），如1300端口号，使用TLS或IPSec，保证呼叫信令通道安全。第9.2节规定了一种动态协商TLS/IPSec实现呼叫安全相关协议规程。

（2）基于共享秘密的对称密码技术，在不安全通道上实现安全认证和完整性检查，并通过对安全能力与密钥的协商机制进行扩展，可以确定后续通道的安全。附录D使用基于口令的散列（Hash）算法，描述了呼叫连接安全具体协议规程。

（3）利用证书在不安全通道上实现安全认证和完整性检查，并通过对安全能力与密钥的协商机制进行扩展，能够确定后续通道的安全。密钥协商有关内容在第12章介绍。

（4）在一个特定业务认证（如AAA）基础上，实现认证与授权（第13章内容）。

H.323网络安全模式，在开始交换第一个呼叫连接消息时，呼叫连接通道（H.225.0）必须操作在已事先协商好的安全或不安全的模式。

在呼叫双方没有重叠的安全能力情形下，被叫端可以拒绝这个连接。返回的错误不宜携带任何有关安全不匹配信息；呼叫终端将不得不通过其他手段确定该问题。呼叫终端接收到一个没有足够安全能力的消息时，宜结束这个呼叫。

如果主叫与被叫终端具有兼容的安全能力，双方假定 H.245 通道必须在所协商的安全模式下进行操作。如果不能建立已经协商好的安全模式下的 H.245 通道，则认为是产生了一个协议错误，并且终止连接。

9.2 采用 TLS/IPSec 实现呼叫连接安全

采用 TLS/IPSec 实现呼叫连接安全有两种配置方法：静态配置方法和基于 RAS 信令的动态协商方法。

静态配置方法假定建立呼叫前双方已预先知道对方（此处可能是终端也可能是网守）所支持的安全协议（IPSec 或 TLS），并预先配置好双方共同支持的保护 H.225.0 呼叫信令的安全协议。这种方法适用于节点规模较小的 H.323 系统。

对于规模较大的 H.323 系统，特别是跨域多网守直接路由模式的网络环境，用户之间预先配置策略方法是困难的。这种情况下，宜采用动态协商方法。

下面结合图 5 与图 6，详细描述直接路由呼叫模式下，通信双方利用 RAS 信令动态协商选择 H.225.0 呼叫信令安全机制方法与协议流程。

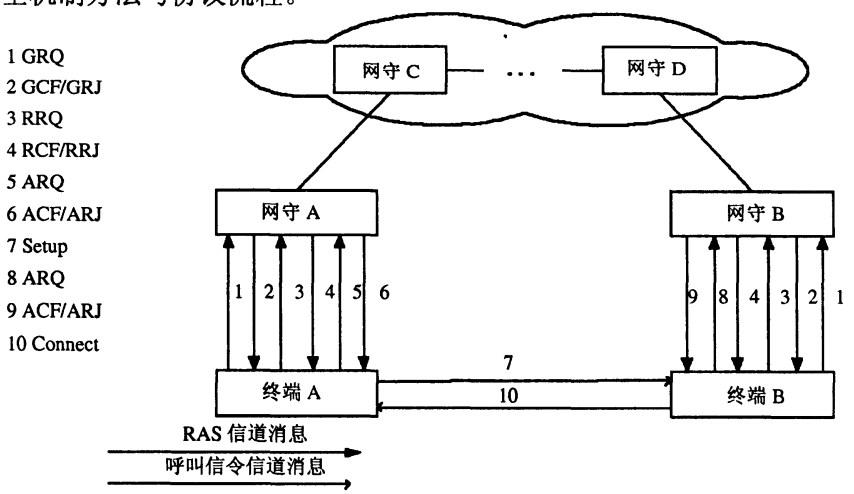


图5 不同网守处注册的两终端间直接呼叫场景

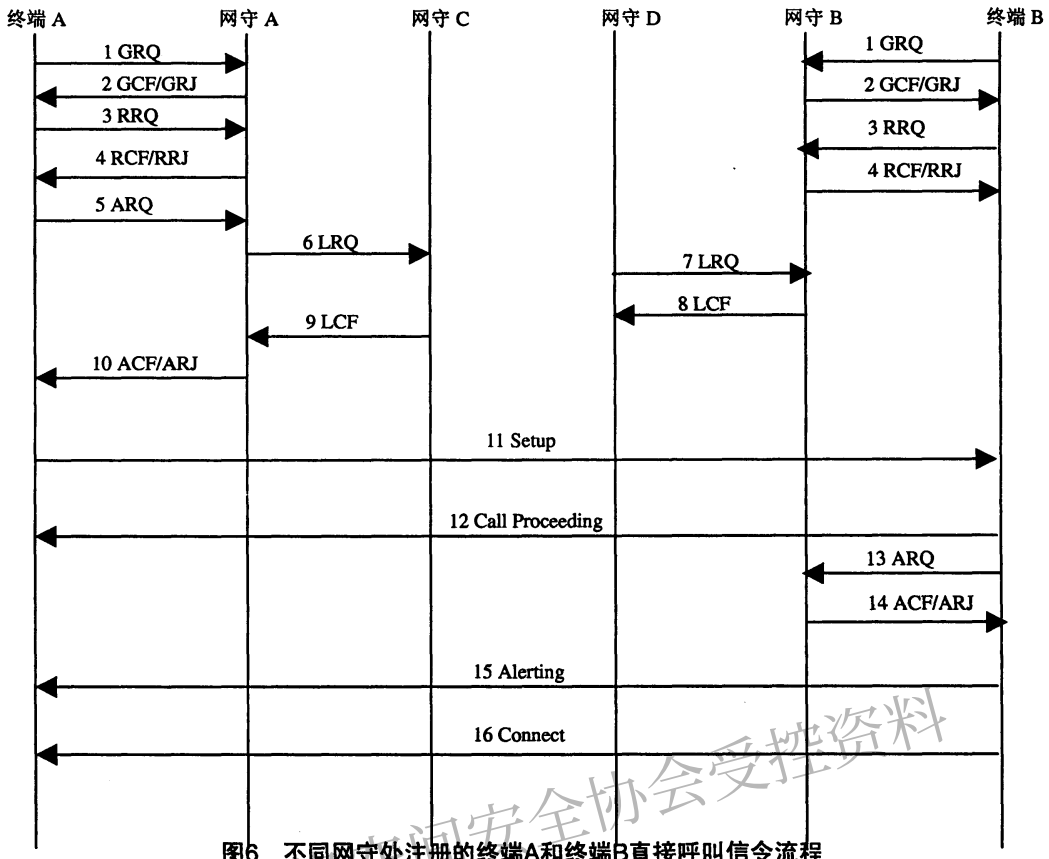


图6 不同网守处注册的终端A和终端B直接呼叫信令流程

（1）终端 A 使用 GRQ 信令 nonStandardData 字段携带自己支持的安全协议及这些安全协议的等级（在此假定 TLS 安全等级为 0.1，IPSec 安全等级为 0.2）等相关参数发送给网守 A，同理，终端 B 将自己支持的安全协议及这些安全协议的等级（在此假定 TLS 安全等级为 0.1）等相关参数发送给网守 B。GRQ 信令的 nonStandardData 字段格式及填写内容如下：

选用 nonStandardData 字段的 GRQ 信令格式如下（其他字段均省略）：

GatekeeperRequest ::= SEQUENCE -- ***（GRQ）***

{

.....

nonStandardData **NonStandardParameter**，

}

其中nonStandardData字段格式如下:

NonStandardParameter ::= SEQUENCE

{

nonStandardIdentifier NonStandardIdentifier，

data OCTET STRING

}

在 nonStandardData 字段的 data 成员中填写终端支持的安全协议（TLS、IPSec 等）及这些安全协议的等级。

（2）网守 A 收到终端 A 支持的安全机制列表（TLS 安全等级为 0.1；IPSec 安全等级为 0.2）后保存在本地，并给终端 A 回应 GCF 信令。网守 B 收到终端 B 支持的安全机制列表（TLS、安全等级为 0.1）后保存在本地，并给终端 B 回应 GCF 信令。

（3）终端 A 向网守 A 发起 RRQ 信令，网守 A 回应 RCF 信令，终端 A 完成在网守 A 的注册过程；同理，终端 B 完成在网守 B 的注册过程。

（4）网守 A 收到终端 A 发起的 ARQ 信令后，向下一个直接相邻的网守 C 发送 LRQ 信令，并在 LRQ 信令中携带本地保存的终端 A 的安全机制列表（TLS 安全等级为 0.1；IPSec 安全等级为 0.2），依此类推，终端 A 的安全机制列表被逐级转发至网守 B。

（5）将网守B收到的终端A的安全机制列表（TLS安全等级为0.1；IPSec安全等级为0.2）与本地保存的终端B的安全机制列表（TLS安全等级为0.1）做比较，选择TLS作为终端A和终端B之间保障Q.931呼叫信令的安全协议，并使用LCF信令携带TLS发送到网守D，依此类推，TLS将被逐级转发回网守A。

（6）网守 A 收到保障 Q.931 呼叫信令的 TLS 安全协议后，使用 ACF 信令的 nonStandardData 字段中携带 TLS 发送给终端 A；

AdmissionConfirm::= SEQUENCE -- *（ACF）*

{

.....

nonStandardData **NonStandardParameter,**

}

其中nonStandardData字段格式如下：

NonStandardParameter ::= SEQUENCE

{

nonStandardIdentifier NonStandardIdentifier,

data OCTET STRING

}

在 nonStandardData 字段的 data 成员中填写终端 A 和终端 B 之间共有的保障 Q.931 呼叫信令的 TLS 安全协议。

（7）终端A使用TLS在终端A和B之间建立安全通道，用于保护H.225.0呼叫信令安全。

10 呼叫控制（H.245）安全技术要求

10.1 简介

呼叫控制通道（H.245）也宜是安全的，包括认证与机密性，以实现后续的媒体流加密。

依赖于所包含的 H.323 终端是否拥有某些协商与/或信令方式，整个 H.245 通道（逻辑通道 0）应该在呼叫信令通道中使用 H.225.0 信令来协商出所需要的算法和密钥，然后以一种安全方式来打开。

H.245 消息在交换期间，对媒体流加密算法与加密密钥进行协商。这种（协商）能力允许不同的媒体通道采用不同的机制加密。例如，集中式多点会议，可以使用不同的密钥对会议中的每个媒体流进行加密。

10.2 安全 H.245 通道操作

假定呼叫连接建立规程已经指示了一个安全模式，H.245 控制通道的协商握手与认证应该发生在任何

其他 H.245 消息交换之前。在保证 H.245 通道安全以后，终端使用 H.245 协议与它们在不安全模式下的操作是一样的。

10.3 不安全 H.245 通道操作

作为选择，H.245 通道可以操作在非安全模式下。此时，两个实体打开一个安全的逻辑通道，用来完成认证和/或共享秘密推导。该共享秘密能够用来保护媒体会话密钥或更新媒体会话密钥。

10.4 安全能力交换

H.323 终端通过使用 ITU-T H.245 第 8.3 节的能力交换规程，对安全机制与加密参数等能力进行交换。每个加密算法与一个特定媒体编码器结合都意味着一个新的安全能力定义。终端在能力交换中可以一起提供加密的和无加密的编码器，这样将允许终端依据开销及可用资源伸缩其安全能力。安全能力交换完成后，终端可以打开一条安全的逻辑通道进行安全媒体通信，就像它们在非安全方式下操作一样。

10.5 主角色

H.245 主从机制被用来建立主角色实体，以便进行双向通道操作及解决其他冲突。主角色也用于安全方法内。尽管某一媒体流安全模式是通过源终端来设置，但主角色终端完成加密密钥生成，而不管其是否为接收者或加密媒体的源。例如，MC 是一个主角色终端，宜为与会的终端产生共享密钥进行多播通道操作。

10.6 逻辑通道信令

终端打开安全媒体逻辑通道与它们打开不安全逻辑通道是一样的。每条通道都可以独立于其他通道进行操作，如生成一安全通道。各个逻辑通道独立操作模式是通过定义在 H.245 的 OpenLogicalChannel 消息的 dataType 域来实现的。在建立逻辑通道的协议交换期间，加密密钥应该从主到从的方向传递（而不管是谁发起了 OpenLogicalChannel）。

10.7 H.245 消息认证与完整性

H.245 消息认证与完整性安全可以通过（隧道）封装到 H.225.0 消息内，通过 H.225.0 消息安全认证与完整性来保证。如果 H.245 消息是在隧道内，在 **h323-UserInformation** 消息中 **h323-uu-pdu** 的域集合应设置如下：

- **h323-message-body** 设置成待传输的 H.225.0 消息类型。
- **h245Tunnelling** 设置为 TRUE。
- **h245Control** 含有 H.245 PDU（协议数据单元）字节串。

如果 H.245 消息不能封装到所有 225.0 消息中，则 H.245 消息被隧道封装在一个特别的 H.225.0 **facility**（能力）消息中来实现安全认证与完整性。

11 媒体流机密性技术要求

媒体流将使用H.245通道中给出的算法与密钥来进行编码。图7与图8演示了通用流程。当新密钥被发送者接收并被用来加密时，SDU头应以某种方式向接收者指出要使用新密钥。在ITU-T H.323内，RTP头（SDU）将改变它的载荷类型以指示交换新密钥或以明文信息传输，具体过程参见附录E。

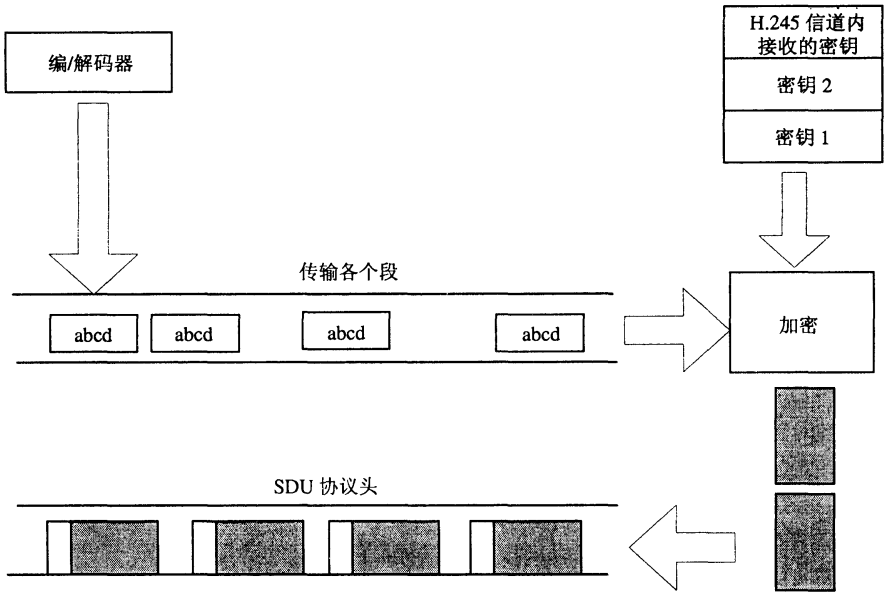


图7　媒体加密

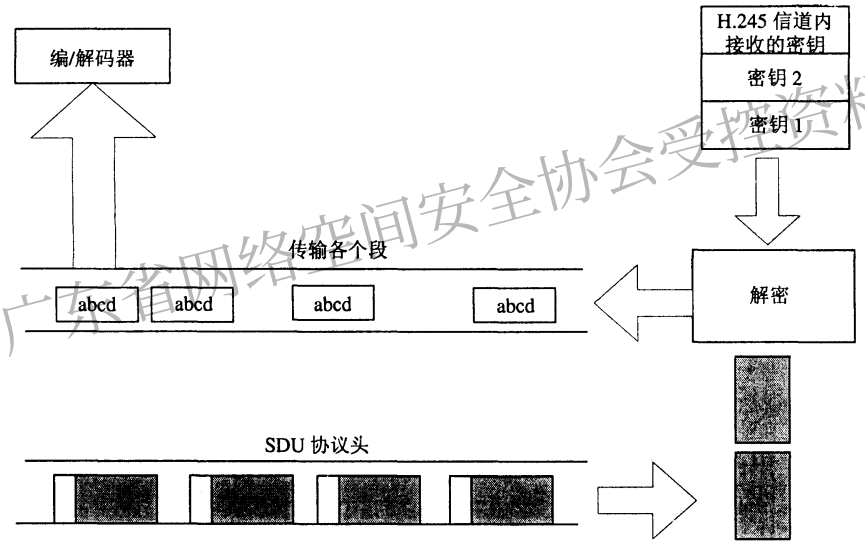


图8　媒体解密

媒体会话密钥保护

媒体会话密钥使用三种机制进行保护。如果 H.245 通道是安全的，则会话密钥不需要施加任何保护。如果一个预共享秘密与算法在 H.245 通道外部建立，例如通过 H.225.0 通道协商出来或通过带外方法实现，那么这个预共享秘密用于对媒体加密密钥进行保护；当 H.245 通道不安全时，可以使用证书（证书也可以用在安全 H.245 通道上），利用证书内的公钥加密媒体会话密钥。

会议中，任意一个与会终端（接收者或发送者）如果怀疑丢失了逻辑通道同步，宜请求一个新密钥。这样做的原因是主角色终端也可以决定异步分配新密钥。接收一个加密更新请求以后，主角色终端将发出密钥更新。如果会议是多点的，MC（也称主角色终端）在把这个新密钥给该发送者之前，应该分配这个新密钥给所有的接收者。接收者应该在尽可能早的时间内利用这个新密钥。

12 密钥管理安全技术要求

12.1 RAS 通道上的密钥管理

在某些情况下，希望在网守控制下发布（RAS）会话密钥。会话密钥可以从网守到一个或多个终端，或从一个终端到另一个终端。需要密钥管理的情形是，可路由网守在 RAS 消息中（如 RCF 或 ACF）发出会话密钥到一个终端，用来加密一个网守—路由信令通道。另一种情形是，网守发出会话密钥，用来加密后续的 RAS 通信（如 RRQ 或 ARQ）。对于多网守路由模式下的密钥交换规程可参考附录 C。为避免频繁的密钥协商，如单位时间内多次进行 Differ-Hellman 算法协商与密钥交换，网守与端点可以共享一个强秘密钥或知道彼此的公钥。

为了安全传输密钥，可以使用 IPSec/SSL 建立一个安全 RAS 或呼叫信令通道，或在不安全的明文通道使用公钥加密和证书实现。

12.2 H.225.0 呼叫建立密钥管理

H.323 系统直接路由模式下跨网守管理范围的呼叫建立时的密钥分配过程，可以让网守根据安全管理策略来选择密钥分配方式。本标准规定可供策略选择的处理流程，不限制如何制订管理策略（可静态配置或根据呼叫量情况动态配置），根据实际情况，可以使用端到端的 Diffie-Hellman（简称 DH）算法协商出一个共享密钥，用作主密钥或作为动态会话密钥。

为了使网守能够自由选择会话密钥分配方式，对应以下三种情况有三种独立过程，具体规程参考附录 C。

- 用于主叫终端支持 DH，主叫终端与被叫网守使用 DH 过程协商会话密钥情况；
- 用于主叫节点不支持 DH 过程，被叫 GK 产生会话密钥情况；
- 用于主叫节点不支持 DH 过程，主/被叫 GK 使用 DH 过程协商会话密钥情况。

主叫终端可以提交一个或几个 Diffie-Hellman 实例。被叫终端应该提供尽可能多的 DH 实例作为他/她的安全策略许可，以增加一个成功寻找公共参数集的可能性。

在从呼叫者 SETUP 消息内所提供的未经排序的 DH 实例中选择一个匹配 DH 实例后，被叫者返回这个双方一致认可的实例。由于安全原因或缺少处理能力而拒绝 DH 实例时，被叫者在响应消息内应该将 DH 参数集置为空。被叫者应该包括它的 DH 令牌（token）在 Setup-to-Connect 响应中。被叫者可以包括它的 DH 令牌（token）在 SETUP 后面的立即响应消息中或包括 DH token 在某些后面的阶段，但至多是在 CONNECT 消息中。

某些原因造成一些路由功能网守可能没有传递所有 Setup-to-Connect 响应给被叫者，结果是一个或多个可能包括 DH 令牌的 H.225.0 响应消息被丢弃或不能到达被叫者，导致呼叫者将不能计算出这个 DH 主密钥与媒体会话密钥。为防止这种情形，被叫者应总是包括相同的 DH 令牌，在每一个 Setup-to-Connect 响应消息中。

13 网守与 AAA 服务器间通信安全技术要求

在一个完整的基于 H.323 的多媒体环境中，AAA 服务器是一个重要的补充功能。例如，用户认证，业务授权，也有账目、计费、账单及其他业务功能。小规模网络环境下，网守可以提供 AAA 功能。而在一个分解体系结构中，网守可能并不总提供这样的服务，或是因为没有对 AAA 数据库的访问权限或是处于不同管理域内。同样，终端或用户通常并不知道他们的 AAA。

网守与 AAA 服务器之间的通信有各种应用方法与协议，而 RADIUS（参见 IETF RFC2138）因其已广泛布署而被认为是最重要的一种。图 9 演示了一种带多媒体终端的方案，一个网守与相连的 AAA。AAA 如何与 GK 之间通信不在本标准范围内。

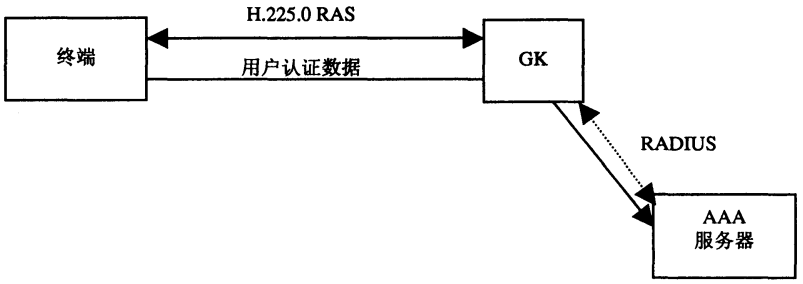


图9　带有AAA的H.323安全

提供 AAA 支持的网守宜至少提供下列两种模式：

（1）缺省模式。该模式中，终端并不知道 AAA，需要与网守（GK）有信任关系。端点以加密形式将用户认证数据发给 GK。GK 对其解密，分离出用户认证信息，并转发给 AAA。例如，基于口令散列的 ClearToken 加密是通过将终端与 GK 之间共享的一个独特的秘密应用到 CryptoToken 上来完成的。加密密钥可以从口令中导出，并以此来完成在 GK 的认证。

（2）RADIUS模式。这里，AAA与端点用户共享一个共同的秘密，对于AAA RADIUS认证，GK并不是可信的。GK仅仅是从AAA接收一个RADIUS挑战（challenge），放入Access-Challenge内并转发给端点，而将用户的响应作为一个RADIUS响应，放入Access-Request转发给AAA。在网守发现期间，端点与GK协商RADIUS挑战/响应能力在AuthenticationMechanism消息的AuthenticationBES中。

当用一个GCF或其他任何RAS消息对端点进行挑战时，并且接收到了带有一个挑战的RADIUS Access-Challenge 消息，GK则将这个16字节挑战放入ClearToken中的challenge域。然后可以向用户呈现该挑战并等待键盘敲入的响应。端点将用一个消息（如RAS）进行答复，其中响应是放在ClearToken中的challenge域。

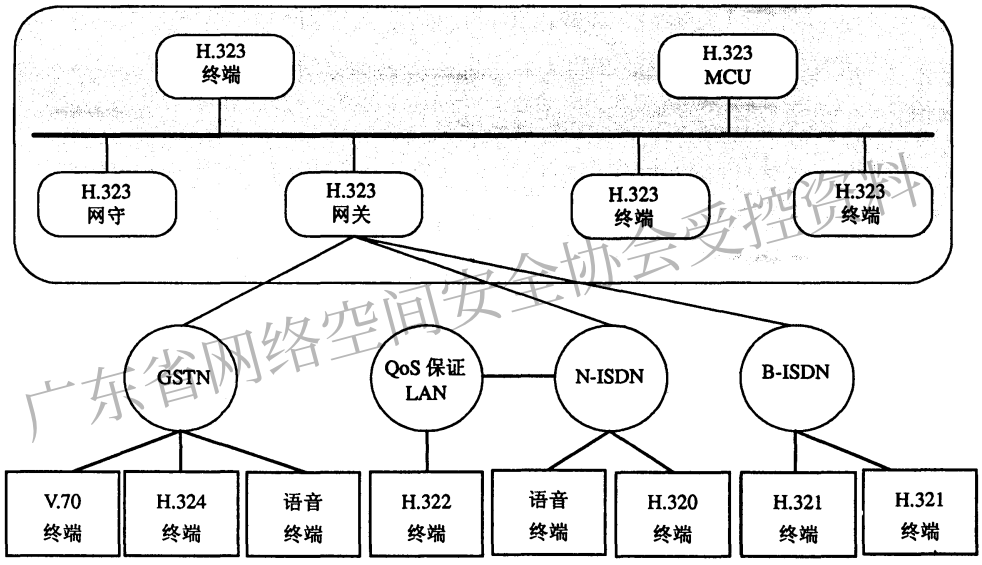
附 录 A
（资料性附录）
H.323系统介绍

ITU-T H.323适用于基于分组网络的语音、视频和数据及其组合的多媒体通信。适用网络可以是不提供QoS的分组网络，其中包括局域网、企业网、城域网以及像Internet在内的互联网。也包括拨号连接或基于GSTN、ISDN并且使用基于包传输（例如PPP）的点到点连接。这些网络构成单一网络或者有着复杂拓扑多个网络段的互联网（其中不同网络段可以通过其他通信连接互联）。

ITU-T H.323重点规定了系统结构和控制过程。其中，H.323必须支持语音，而视频和数据是任选功能。

A.1 系统结构

H.323系统结构如图A.1所示。



图A.1 H.323系统结构

H.323系统的组成部件称为H.323实体，包括终端、网关、网守、多点控制器、多点处理器和多点控制单元。标准中控制消息和过程定义组成部件之间如何通信。其中，终端、网关和MCU统称为端点。端点可以发起呼叫，也可以接受呼叫，媒体信息流在端点生成或终结。网守、多点控制器、多点处理器则不可呼叫，但是网守参与呼叫的控制，具有传输层地址，是可寻址的H.323实体。多点控制器和多点处理器执行多点呼叫信息流的处理和控制，是系统的功能实体，物理上总是位于某个端点之中，因此没有独立的传输层地址，既不可呼叫又不可寻址的H.323实体。

H.323终端是在分组网络上遵从H.323建议标准进行实时通信的端点设备，可以集成在PC中，也可以是独立的IP电话机或可视电话等。

网守为H.323端点提供地址翻译和接入控制服务，还可以提供带宽管理和网关定位等服务。网守是网络的管理点，一个网守管理的所有终端、网关和多点控制单元的集合称为一个管理区。通常，将同属于一个运营机构管辖的H.323端点的集合称为一个管理域。在一个管理域中可含有多个网守。

H.323系统通过网关和其他网络互通。网关的作用主要是完成媒体信息编码转换和信令转换两项功能。

H.323系统的协议栈结构如图A.2所示。

<table>
<tr><td colspan="2">G.7xx</td><td>H.26x</td><td rowspan="3">RTCP</td><td rowspan="3">H.225.0终端
至网守信令
（RAS）</td><td rowspan="3">H.225.0
呼叫
信令</td><td rowspan="3">H.245
媒体信
道控制</td><td rowspan="3">T.120</td></tr>
<tr><td colspan="3">加密</td></tr>
<tr><td colspan="3">RTP</td></tr>
<tr><td colspan="4">不可靠传送协议</td><td colspan="4">可靠传送协议</td></tr>
<tr><td colspan="8">网络层</td></tr>
<tr><td colspan="8">链路层</td></tr>
<tr><td colspan="8">物理层</td></tr>
</table>

图A.2 H.323系统结构

A.2 控制协议

H.225.0和H.245是H.323系统的核心协议。前者主要用于呼叫控制，后者用于媒体信息控制。

在任何呼叫开始之前，首先必须通过H.225.0在端点之间建立呼叫联系。

H.225.0主要包括RAS和呼叫信令协议两部分。

RAS是登记、接纳和状态协议，作用是为网守提供确定端点地址和状态、呼叫接纳等功能。

RAS主要功能有网守搜寻、端点登记、端点定位、呼叫接纳、呼叫退出、带宽管理、状态查询、网关资源指示。

呼叫信令协议主要是基于ITU-T Q.931制定的。不同于Q.931的控制对象包括呼叫和连接，ITU-T H.225.0的控制对象仅限于呼叫，最后建立起端点间的H.245控制通道。而H.245控制各端点之间媒体通信连接的建立和释放。

ITU-T H.245是通用的多媒体通信控制协议，主要针对会议通信设计。H.323采用H.245协议作为控制协议，实现通信通道的建立、维护和释放。H.245的控制信号在一条专门的可靠通道（如TCP）上传送，称为H.245控制通道。

H.245主要功能有：逻辑通道的打开和关闭；收发双方的能力交换；主从确定过程；维护管理功能；会议通信控制。

A.3 控制过程

H.323端点之间建立通信关系一般执行三个控制过程：

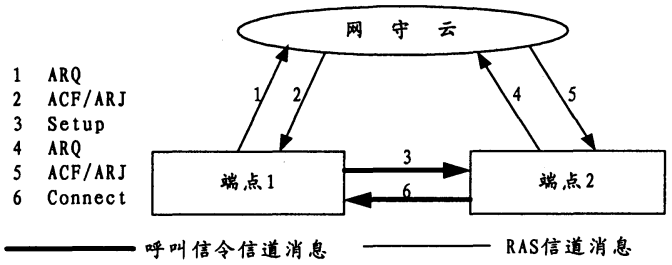
- 呼叫接纳（RAS）；
- 呼叫控制（呼叫信令协议）；
- 连接控制（H.245）。

为了加快呼叫建立速度，ITU-T H.323第三版本又定义了快速连接信令和隧道技术，可以将呼叫控制和连接控制合为一体。

具体信令传输方式见下。

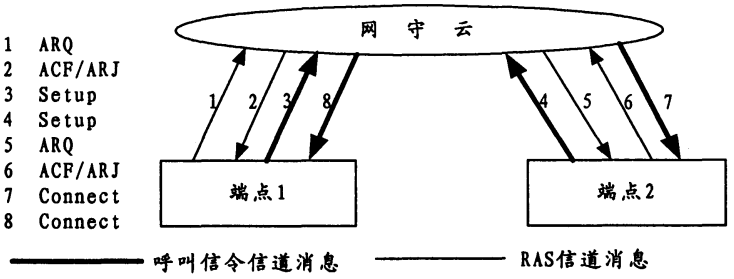
A.3.1 呼叫信令消息的传送

呼叫信令消息的传送方式有直接选路方式和网守选路方式，分别如图A.3和A.4所示。



图A.3 呼叫信令消息的直接选路方式

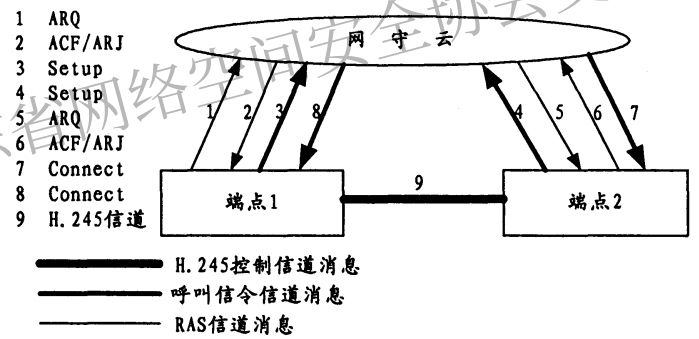
具体信令流程可参考ITU-T H.323。



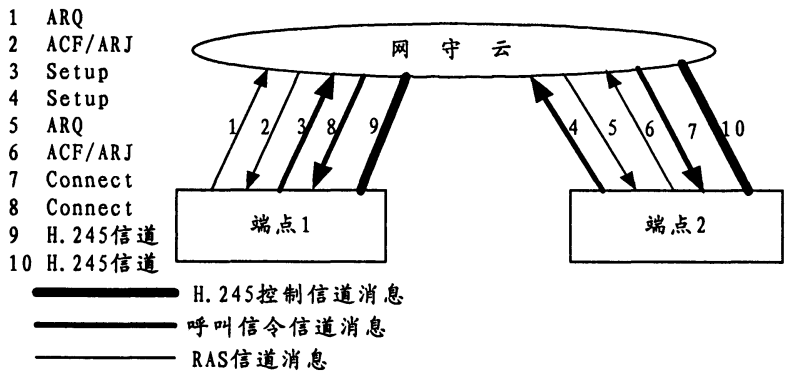
图A.4 呼叫信令消息的网守选路方式

A.3.2 媒体控制消息的传送

媒体控制消息的传送方式有直接选路方式和网守选路方式，分别如图A.5和图A.6所示。



图A.5 媒体控制消息的直接选路方式



图A.6 媒体控制消息的网守选路方式

附 录 B
（规范性附录）
H.235 ASN.1

```
H235-SECURITY-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
-- EXPORTS All

TimeStamp              ::= INTEGER（1..4294967295）-- 自00:00 1/1/1970以来所经过的秒数
ChallengeString        ::= OCTET STRING（SIZE（8..128））
RandomVal              ::= INTEGER -- 32-位整数
Password          ::= BMPString（SIZE（1..128））
Identifier        ::= BMPString（SIZE（1..128））
KeyMaterial            ::= BIT STRING（SIZE（1..2048））

NonStandardParameter ::= SEQUENCE
{
    nonStandardIdentifier    OBJECT IDENTIFIER,
    data                     OCTET STRING
}
DHset ::= SEQUENCE
{
    halfkey        BIT STRING（SIZE（0..2048））, -- = g^x mod n
    modSize        BIT STRING（SIZE（0..2048））, --   n
    generator      BIT STRING（SIZE（0..2048））, -- g
    ...
}

TypedCertificate ::= SEQUENCE
{
    type           OBJECT IDENTIFIER,
    certificate    OCTET STRING,
    ...
}

AuthenticationBES ::= CHOICE
{
```

```
    default    NULL，--加密的明文令牌
    radius          NULL，-- RADIUS-挑战/响应
    ...
}

AuthenticationMechanism      ::= CHOICE
{
    dhExch              NULL，-- Diffie-Hellman认证机制
    pwdSymEnc           NULL，--使用口令的对称加密认证机制
    pwdHash         NULL，--使用口令的散列加密认证机制
    certSign        NULL，--签名证书认证机制
    ipsec               NULL，--使用IPSec建立连接
    tls             NULL，
    nonStandard     NonStandardParameter，--其他认证算法
    ...，
    authenticationBES  AuthenticationBES --用于后端服务器用户认证
}

ClearToken          ::= SEQUENCE   --一个 "token" 可能包含多个值类型
{
    tokenOID        OBJECT IDENTIFIER，
    timeStamp           TimeStamp OPTIONAL，
    password        Password OPTIONAL，
    dhkey               DHset OPTIONAL，
    challenge       ChallengeString OPTIONAL，
    random          RandomVal OPTIONAL，
    certificate     TypedCertificate OPTIONAL，
    generalID       Identifier OPTIONAL，
    nonStandard     NonStandardParameter OPTIONAL，
    ...，
    eckasdhkey          ECKASDH OPTIONAL，--椭圆曲线密钥协商方案-Diffie
                                          -- Hellman模拟（ECKAS-DH）
    sendersID           Identifier OPTIONAL，
    h235Key         H235Key OPTIONAL --V3版本中集中分配的密钥
}

--  对象标识符应放置在 tokenOID内，当ClearToken被直接包含在一个消息中（相对于被加密）
```

```
--   所有其他情形，应用应当使用对象标识符 {0 0}指示 tokenOID值不存在
--
-- 从这开始所有的密码学参数类型
--
--

SIGNED { ToBeSigned } ::= SEQUENCE {
    toBeSigned          ToBeSigned,
    algorithmOID OBJECT IDENTIFIER,
    paramS          Params，--任意"运行时"参数
    signature       BIT STRING --可以是 RSA 或一个ASN.1 编码的ECGDSA签名
}（ CONSTRAINED BY { --验证或签名证书-- } ）


ENCRYPTED { ToBeEncrypted } ::= SEQUENCE {
    algorithmOID        OBJECT IDENTIFIER,
    paramS              Params，--任意"运行时"参数
    encryptedData       OCTET STRING
}（ CONSTRAINED BY { --加密或解密-- ToBeEncrypted } ）

HASHED { ToBeHashed } ::= SEQUENCE {
    algorithmOID        OBJECT IDENTIFIER,
    paramS              Params，--任意"运行时"参数
    hash                BIT STRING
}（ CONSTRAINED BY { --散列-- ToBeHashed } ）

IV8 ::= OCTET STRING（SIZE（8））-- 64-位块密码的初始值
IV16 ::= OCTET STRING（SIZE（16））--128-位块密码初始值

--使用的信令算法必须选择这些参数类型之一
--接收信令端所需要

Params ::= SEQUENCE {
    ranInt              INTEGER OPTIONAL，--某些整数值
    iv8             IV8 OPTIONAL，--8字节初始矢量
    ...,
    iv16            IV16 OPTIONAL，     --  16 字节初始矢量
```

```
iv              OCTET STRING OPTIONAL，--任意长度初始矢量
    clearSalt       OCTET STRING OPTIONAL --用于加密的非加密掺杂密钥
}

EncodedGeneralToken ::= TYPE-IDENTIFIER.&Type（ClearToken --常用令牌（ token）-- ）
PwdCertToken ::= ClearToken（WITH COMPONENTS {...，timeStamp PRESENT，generalID PRESENT}）
EncodedPwdCertToken ::= TYPE-IDENTIFIER.&Type（PwdCertToken）

CryptoToken::= CHOICE
{

    cryptoEncryptedToken SEQUENCE --通用/专用令牌
    {
        tokenOID        OBJECT IDENTIFIER，
        token           ENCRYPTED { EncodedGeneralToken }
    }，
    cryptoSignedToken   SEQUENCE --通用/专用令牌
    {
        tokenOID        OBJECT IDENTIFIER，
        token           SIGNED { EncodedGeneralToken }
    }，
    cryptoHashedToken SEQUENCE --通用/专用令牌
    {
        tokenOID            OBJECT IDENTIFIER，
        hashedVals          ClearToken，
        token HASHED { EncodedGeneralToken }
    }，
    cryptoPwdEncr       ENCRYPTED { EncodedPwdCertToken }，
    ...
}

-- 这些允许在H.245 打开的逻辑通道（OLC）内传递会话密钥
-- 它们被编码为标准的ASN.1以H.245内的OCTET STRING为基础
H235Key::=CHOICE   --用于 H.245 or ClearToken "h235Key" field
{
    secureChannel       KeyMaterial，
    sharedSecret            ENCRYPTED {EncodedKeySyncMaterial}，
```

```
    certProtectedKey        SIGNED { EncodedKeySignedMaterial },
    ...,
    secureSharedSecret      V3KeySyncMaterial -- 用于 H.235 V3端点
}

KeySignedMaterial ::= SEQUENCE {
    generalId       Identifier, -- 从别名
    mrandom         RandomVal, --主随机值
    srandom         RandomVal OPTIONAL, -- 从随机值
    timeStamp           TimeStamp OPTIONAL, --时间戳
    encrptval       ENCRYPTED {EncodedKeySyncMaterial }
}
EncodedKeySignedMaterial ::= TYPE-IDENTIFIER.&Type（KeySignedMaterial）

H235CertificateSignature    ::= SEQUENCE
{
    certificate         TypedCertificate,
    responseRandom          RandomVal,
    requesterRandom         RandomVal OPTIONAL,
    signature           SIGNED { EncodedReturnSig },
    ...
}

ReturnSig ::= SEQUENCE {
    generalId           Identifier, -- 从别名
    responseRandom          RandomVal,
    requestRandom           RandomVal OPTIONAL,
    certificate             TypedCertificate OPTIONAL --请求的证书
}

EncodedReturnSig ::= TYPE-IDENTIFIER.&Type（ReturnSig）
KeySyncMaterial  ::= SEQUENCE
{
    generalID       Identifier,
     eyMaterial         KeyMaterial,
```

```
EncodedKeySyncMaterial    ::=TYPE-IDENTIFIER.&Type（KeySyncMaterial）


V3KeySyncMaterial    ::= SEQUENCE
{
    generalID              Identifier OPTIONAL，-- 对等终端ID
    algorithmOID           OBJECT IDENTIFIER OPTIONAL，-- 加密算法
    paramS                 Params，-- IV
    encryptedSessionKey        OCTET STRING OPTIONAL，-- 加密的会话密钥
    encryptedSaltingKey        OCTET STRING OPTIONAL，-- 加密的媒体掺杂密钥
    clearSaltingKey            OCTET STRING OPTIONAL，--非加密的媒体掺杂密钥
    paramSsalt                 Params OPTIONAL，--用于掺杂密钥的 IV    keyDerivationOID
    OBJECT IDENTIFIER OPTIONAL，--密钥推导方法    ...
}


END    -- H235-安全-消息定义结束
```

附 录 C
（规范性附录）
H.323网络直接和选择路由呼叫安全

C.1 范围

本附录目的是为跨域多网守环境下直接路由模式和选择路由模式的呼叫提供安全规程，该规程结合附录 D 和 ITU-T H.235.3 所规定的安全轮廓一起使用。

本安全轮廓作为一种选项，可以弥补附录 D 或 ITU-T H.235.3 所规定的安全轮廓的不足。通过采用对称密钥管理技术，本安全轮廓也提供了 RAS 信道密钥管理的实现细节。

C.2 简介

H.235.1 基线安全以及 H.235.3 混合安全轮廓，使用网守作为可信任中间实体，按逐跳方式应用共享秘密来保证消息的认证和/或完整性。这种网守路由模型的优点是便于计费。

然而，随着并发呼叫数规模的日益增长，带网守的直接路由模型因其具有更好的性能和可扩展性，得到了越来越广泛的应用。该模型优点是使用网守进行注册，接入、地址解析、带宽控制、呼叫建立则在终端之间以端到端的方式进行。

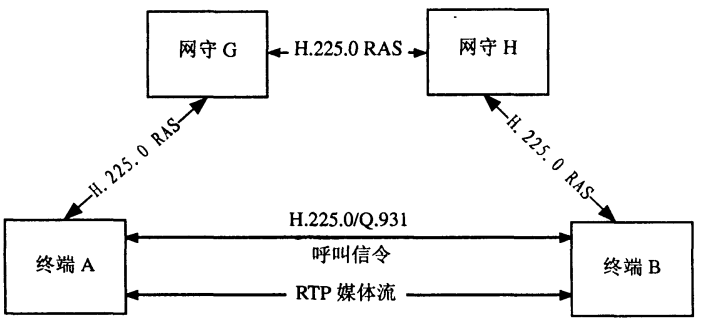
直接路由模型中，不能保证在两个终端之间存在预共享的秘密，因为这要求每个终端要记忆和其他所有终端之间的共享秘密。

本附录描述了支持跨域多网守直接路由模型的呼叫安全增强方法与具体信令规程。

C.3 呼叫场景

本附录描述的呼叫安全场景如图 C.1 示，并假设网守域之间的 IP 网络是不安全的。实现本附录中呼叫安全的前提是终端使用基线安全或混合安全的方式安全注册到网守上，并且部署为直接呼叫路由模式。

因此，使用类似 Kerberos 的方式，发起呼叫终端（DRC1）的网守和接收呼叫终端（DRC2）的网守能够给直接通信的终端提供一个共享秘密。



图C.1 直接呼叫路由场景

基于不同的安全策略，本附录定义了 DRC1、DRC2 和 DRC3 三个过程。

过程 DRC1（参见附录 C.5）可应用于企业环境。此环境中，网守放在不同的地点，但所有网守都遵守企业内统一的安全策略。在这样的环境中，假设由呼叫的源端网守 G 决定该呼叫的安全策略，因此源端网守 G 选择将使用的安全参数。接收端网守 H 接受网守 G 选择的安全参数。

过程 DRC2（参见附录 C.6）和 DRC3（参见附录 C.7）可应用于域间通信环境。此环境中，不同管

理域内的网守可以布署不同的安全策略。

过程 DRC2 适用于主叫终端或网守不支持 Diffie-Hellman 算法的案例。在这样的环境中，假设被叫终端网守 H 决定呼叫的安全策略，因此被叫终端网守 H 选择所使用的安全参数。源端网守 G 接受网守 H 选择的安全参数。

过程 DRC3 适用于主叫终端不支持 Diffie-Hellman 算法，但源端网守和接收端网守都支持 Diffie-Hellman 算法的情形。

开始注册时，以上规程提供信令方法以协商应用 DRC1、DRC2 或 DRC3 中哪一种规程。

C.4 限制

本附录不支持没有网守的直接路由场景。

C.5 DRC1 规程（公司环境）

本节描述的规程可应用于企业环境。该环境中，网守放在不同的地点，但所有网守都遵守企业内统一的安全策略。在这样的环境中，假设由呼叫的源端网守 G 决定该呼叫的安全策略，因此源端网守 G 选择将使用的安全参数。接收端网守 H 接受网守 G 选择的安全参数。

C.5.1 GRQ/RRQ 阶段

在 **GRQ** 和/或 **RRQ** 阶段，支持本安全轮廓的终端应该包含一个 **tokenOID** 为“I10”的 ClearToken，其中，ClearToken 的其他字段不宜使用。支持本安全轮廓的网守如果愿意提供 DRC1 功能，则应该在 **GCF** 或 **RCF** 中包含一个 **tokenOID** 为“I10”的 ClearToken，其中 ClearToken 的其他字段不使用。

C.5.2 ARQ 阶段

在终端 A 开始向终端 B 直接发送呼叫信令消息之前，终端 A 或 B 应该使用 **ARQ** 消息向网守 G 或 H 请求接入许可。终端 A 应该在 **ARQ** 消息中包含一个 **tokenOID** 为“I10”的 ClearToken 字段，其中 ClearToken 的其他字段未使用。

C.5.3 LRQ 阶段

该过程涵盖了单个网守（所有终端共有一个网守）以及多层链接的网守的情形。多网守情形中，网守 G（发起呼叫的域）宜使用（组播的）**LRQ** 定位被叫终端所在的网守 H，参见 ITU-T H.323 第 8.1.6 节“可选的被叫终端信令”。两网守之间的通信应该采用附录 D 规定的安全保护方法。为此，网守之间假定存在共享秘密 K_{GH}。由于网守间的 **LRQ** 消息通常采用组播发送，K_{GH} 通常不能是对共享秘密（pair-wise shared secret），而是假设为一组网守的组内共享秘密。

如果 **LRQ** 方法用来定位远端网守，那么 **LRQ** 应该携带一个包含 **tokenOID** 为“I10”的 ClearToken 字段，ClearToken 的其他字段不宜使用。对于组播的情况，不应该使用 **LRQ** 消息中 ClearToken 字段内的 **generalID**。

C.5.4 LCF 阶段

EK_{BH} 代表终端 B 和网守 H 之间的加密密钥，而 KS_{BH} 为掺杂密钥。如下所述，网守 H 和终端 B 使用伪随机函数（PRF）从共享秘密 K_{BH} 演算这些密钥材料。

网守 H 应该产生一个随机的挑战值 B，并使用附录 C.8 定义的基于 PRF 密钥推导过程，从共享秘密 K_{BH} 推导出密钥 EK_{BH} 以及掺杂密钥 KS_{BH}，其中，挑战值 B 代替 **challenge**，CT_{HG}→**h235Key**→**secureSharedSecret**→**keyDerivationOID** 应该设为“AnnexI-HMAC-SHA1-PRF”，参见附录 C.10。

EK_{GH}代表网守 G 和网守 H 之间的加密密钥，KS_{GH}代表网守 G 和网守 H 之间的掺杂密钥。网守 H 应该产生一个随机的挑战值 G。网守 H 应该使用第 C.8 章定义的基于 PRF 密钥推导过程从共享秘密 K_{GH} 推导出密钥 EK_{GH} 以及掺杂密钥 KS_{GH}，其中，挑战值 G 代替 **challenge**，CT_{HG}→**challenge** 应该包含挑战值 G，终端 B 的标识符应该放在 CT_{HG}→**h235Key**→**secureSharedSecret**→**generalID** 中。

网守 H 应该给网守 G 发送加密的 EK_{BH} 和加密的 KS_{BH}。增强的 OFB（EOFB）加密模式（参见附录 E.5.4）应该和这些秘密及专用于终端的掺杂密钥 KS_{GH} 一起使用。能应用的加密算法如下（参见附录 E.6）：

- 工作在 EOFB 模式的 DES（56 比特）算法，OID 为 Y1（可选）；
- 工作在外层 EOFB 模式的 3DES（168 比特）算法，OID 为 Z1（可选）；
- 工作在 EOFB 模式的 AES（128 比特）算法，OID 为 Z2（缺省且推荐）；
- 工作在 EOFB 模式的 RC2（56 比特）算法，OID 为 X1（可选）。

对于 EOFB 加密模式，网守 H 应该产生一个随机的初始矢量 IV。对于 OID“X1”、OID“Y1”和 OID“Z1”算法，初始矢量为 64 比特，并且应该被携带在 CT_{HG}→**h235Key**→**secureSharedSecret**→**params**→**iv8** 字段内；而对于 OID“Z2”算法，IV 长度为 128 比特，应该在 CT_{HG}→**h235Key**→**secureSharedSecret**→**params**→**iv16** 内携带。

网守 H 应该在 CT_{HG} 中包含加密的 EK_{BH} 和加密的 KS_{BH}，即 $ENC_{EKGH,\ KSGH,\ IV}$（EK_{BH}）和 $ENC_{EKGH,\ KSGH,\ IV}$（KS_{BH}），并设置 **tokenOID** 为“I13”。$ENC_{EKGH,\ KSGH,\ IV}$（EK_{BH}）应该在 CT_{HG}→**h235Key**→**secureSharedSecret**→**encryptedSessionKey** 中传送，$ENC_{EKGH,\ KSGH,\ IV}$（KS_{BH}）应该在 CT_{HG}→**h235Key**→**secureSharedSecret**→**encryptedSaltingKey** 中传送。加密算法应该在 CT_{HG}→**h235Key**→**algorithmOID** 中指示。挑战值 B 应该放在 CT_{HG}→**h235Key**→**secureSharedSecret**→**clearSaltingKey** 中。CT_{HG}→**generalID** 应该设置为网守 G 的标识符，CT_{HG}→**sendersID** 应该设置为网守 H 的标识符。

挑战值 B 应该传送到终端 B，通过 **ClearToken** 内包含 **profileInfo** 字段来完成。具体字段设置：CT_{HG}→**profileInfo**→**elementID**=0 标识这个特殊轮廓元素，CT_{HG}→**profileInfo**→**paramS** 未使用，CT_{HG}→**profileInfo**→**element**→**octets** 应该保存挑战值 B。**LCF** 响应应该包含 CT_{HG}。

C.5.5 ACF 阶段

网守 G，识别出终端 A 和终端 B 支持本过程，应该生成密钥材料和 ClearToken 令牌，具体如下：

除正常 ARQ 操作，网守 G 能够计算出基于呼叫（call-based）的共享密钥 K_{AB}。这个基于呼叫的共享秘密在 ClearToken 中被推送到两个终端（终端 A 与 B）。这些 ClearToken 在 **ACF** 消息中携带并送给主叫方。

被推送的两个 ClearToken，一个 CT_A 用于主叫终端 A，另一个 CT_B 用于被叫终端。每个 **ClearToken** 应该在 **tokenOID** 中包含一个 OID（“I11”或“I12”），指示这个 ClearToken 是针对主叫方（CT_A 为“I11”）还是被叫方（CT_B 为“I12”）。

网守 G 应该解密 CT_{HG}→**h235Key**→**secureSharedSecret**→**encryptedSessionKey** 来获得 EK_{BH}，解密 CT_{HG}→**h235Key**→**secureSharedSecret**→**encryptedSaltingKey** 来获得 KS_{BH}。

本附录中定义的 **ClearToken** 可以结合其他的安全轮廓一起使用，例如附录 D 和 ITU-T H.235.3 所规定的安全轮廓。在这种情况下，本附录中的 **ClearToken** 也应该使用 **ClearToken** 的其他字段。例如，为了和附录 D 所规定的安全轮廓一起使用，**timestamp**、**random**、**generalID**、**sendersID** 和 **dhkey** 字段应该存在并被使用，详细参见附录 D。

网守 G 的网守标识符（GKID）应该放在 CT_A→**sendersID** 和 CT_B→**sendersID** 中，CT_A→**generalID** 中保存终端 A 的标识符，CT_B→**generalID** 中保存终端 B 的标识符。

网守 G 应该使用第 C.8 章定义的基于 PRF 密钥推导过程从共享秘密 K_{GH} 生成掺杂密钥材料 KS_{GH} 和加密密钥材料 EK_{GH}，其中，**challenge** 用 CT_{HG}→**challenge** 替代。

用于加密端到端密钥 K_{AB} 的加密密钥 EK_{AG} 和 EK_{BH} 应该使用第 C.8 章定义的 PRF 密钥推导过程从共享秘密（K_{AG} 或 K_{BH}）推导出来。其中，CT_A→**h235Key**→**secureSharedSecret**→**keyDerivationOID** 和 CT_B→**h235Key**→**secureSharedSecret**→**keyDerivationOID** 应该设置为"AnnexI-HMAC-SHA1-PRF"（见附录 C.14）CT_A→**challenge** 应该设置为挑战值 A。

网守 G 应该把挑战值 B 从 CT_{HG}→**h235Key**→**secureSharedSecret**→**clearSaltingKey** 复制到 CT_B→**challenge**。

CT_B→**profileInfo** 应该保存 CT_{HG} →**profileInfo** 传送来的轮廓元素，这样终端 B 最终获得挑战值 B。

这个会话密钥 K_{AB} 应该使用某个加密算法用 EK_{AG}（目标为终端 A 的 ClearToken）或 EK_{BH}（目标为终端 B 的 ClearToken）加密。

增强 OFB（EOFB）加密模式（参见附录 E.5.4）应该和该秘密、专用于终端的掺杂密钥 KS_{AG} 或 KS_{BH} 一起使用。使用的加密算法如下（参见附录 E.7）：

- 工作在 EOFB 模式的 DES（56 比特）算法，OID 为 Y1（可选）；
- 工作在外层 EOFB 模式的 3DES（168 比特）算法，OID 为 Z1（可选）；
- 工作在 EOFB 模式的 AES（128 比特）算法，OID 为 Z2（缺省且推荐）；
- 工作在 EOFB 模式的 RC2（56 比特）算法，OID 为 X1（可选）。

对于 EOFB 加密模式，网守 G 应该产生一个随机的初始矢量 IV。对于 OID "X1"、OID "Y1" 和 OID "Z1" 算法，初始矢量为 64 比特，并且应该在 CT_A→**h235Key**→ **secureSharedSecret**→**params**→**iv8** 或 CT_B→**h235Key**→**secureSharedSecret**→**params**→**iv8** 内传送；OID "Z2" 算法需要 128 比特的初始矢量，应该在 CT_A→**h235Key**→ **secureSharedSecret**→**params**→**iv16** 或 CT_B→**h235Key**→**secureSharedSecret**→**params**→**iv16** 内传递。

获得的 $ENC_{EKAG,\ KSAG,\ IV}(K_{AB})$ 应该在 CT_A→**h235Key**→**secureSharedSecret**→ **encryptedSessionKey** 中传送，$ENC_{EKBH,\ KSBH,\ IV}(K_{AB})$ 应该在 CT_B→**h235Key**→ **secureSharedSecret**→**encryptedSessionKey** 中传送。加密算法在 CT_A→**h235Key**→**secureSharedSecret**→**algorithmOID** 和 CT_B→**h235Key**→**secureSharedSecret**→**algorithmOID** 中指示。

对于以终端 A 为目的地的 ClearToken，终端 **B** 的标识符（$EPID_B$）应该放在 CT_A→**h235Key**→**secureSharedSecret**→**generalID** 中。同样，对于目的地为终端 B 的 ClearToken，终端 **A** 的标识符（$EPID_A$）应该放在 CT_A→**h235Key**→**secureSharedSecret**→ **generalID** 中。

对于 EOFB 加密模式，**encryptedSaltingKey** 不应该被使用。

网守 G 应该在发送给终端 A 的 ACF 消息中包含两个 ClearToken，即 CT_A 和 CT_B。

C.5.6 SETUP 阶段

终端 A 应该通过检查 ClearToken 内 **tokenOID** 是否为 "I11" 标识 CT_A。

终端 A 应该通过检查 **timestamp** 来验证收到的 CT_A 是否及时（fresh）。然后应该进一步检查 ClearToken 的 **generalID** 和 **sendersID**，以及 **V3KeySyncMaterial** 中的 **generalID**。如果 CT_A 是及时的，终端 A 应该

获取 IV 并按前面网守 G 的描述计算 EK_{AG} 和 KS_{AG}。终端 A 应该解密 CT_A→**secureSharedSecret**→**encryptedSessionKey** 来获得 K_{AB}。

如果收到的 CT_A 验证是及时的，则终端 A 能够发送 SETUP 消息给终端 B。这个 SETUP 消息包含 CT_B。SETUP 消息应该按照附录 D 或 ITU-T H.235.3 规定，使用共享密钥 K_{AB} 自身进行安全保护（认证和完整性）。为此，附录 D 规定的被散列的 ClearToken（不是 CT_B）中，**generalID** 不应该被使用，除非终端 A 已经有可用的 $EPID_B$（如通过配置或以前通信中记住的）。如果终端 A 使用 $EPID_B$ 作为 SETUP 中的 generalID，那么终端 A 应该接受返回的呼叫信令消息中的 sendersID 值并作为真正的 $EPID_B$。

终端 B 应该通过检查 ClearToken 的 **tokenOID** 是否为“I12”判别 CT_B。

终端 B 应该通过检查 **timestamp** 验证收到的 CT_B 是否及时（fresh）。然后应该进一步检查 ClearToken 的 **sendersID** 和 **secureSharedSecret** 内的 **generalID**。如果 CT_B 是及时的，终端 B 应该从 CT_{HG}→**profileInfo**→**element**→**octets** 中获取挑战值 B，获取初始矢量 IV，计算 EK_{BH} 和 KS_{BH}，挑战值 B 替代为 C.8 章中的 **challenge**，如上针对网守的描述。终端 B 应该解密出 CT_B→**secureSharedSecret**→**encryptedSessionKey** 信息来获得 K_{AB}。

对于 CT_B 验证为及时的情形，终端 B 能够答复 CALL-PROCEEDING、ALERTING 或 CONNECT 等消息，呼叫继续。对于 CT_B 验证是不及时或安全验证 SETUP 消息失败的情形，终端 B 应该响应 RELEASE 消息，且 **ReleaseCompleteReason** 设置为安全错误原因。

当部署了媒体安全时（附录 E），终端 A 和 B 应该使用 Diffie-Hellman 密钥协商过程建立动态的、基于会话的主密钥，并由主密钥保护导出媒体有关的会话密钥。

终端 B 应该设置 **generalID** 为 $EPID_A$，**sendersID** 为 $EPID_B$，以保护目的地为终端 A 的所有 H.225.0 呼叫信令消息（例如，CALL-PROCEEDING、ALERTING 和 CONNECT 消息）。

图 C.2 所示为 DRC1 基本通信流程。

图C.2 基本通信流程（DRC1）

C.6 规程 DRC2（域间环境）

本章描述的规程可应用于域间通信环境。此环境中，位于不同管理域内的网守具有不同的安全策略。规程 DRC2 适用于主叫终端或网守不支持 Diffie-Hellman 算法的情形。

在这样一种环境中，假设接收端网守 H 决定待建立的呼叫安全策略；因此接收端网守 H 选择并选取所应用的安全参数。源端网守 G 接受网守 H 所选取的安全参数。

C.6.1 GRQ/RRQ 阶段

在 GRQ 和/或 RRQ 阶段，支持本安全轮廓的终端应该包含一个单独的 ClearToken，其 tokenOID 为“I20”。其中，ClearToken 其他字段不宜使用。支持本安全轮廓的网守如果愿意提供 DRC2 功能，则应该在 GCF 或 RCF 中包含一个 tokenOID 为“I20”的单独 ClearToken，其中，ClearToken 其他字段不使用。

C.6.2 ARQ 阶段

在终端 A 开始向终端 B 直接发送呼叫信令消息之前，终端 A 或 B 应该使用 ARQ 消息向网守 G 或 H

请求接入许可。终端 A 应该在 ARQ 消息中包含一个 tokenOID 为“I20”的单独 ClearToken；其中，ClearToken 的其他字段不使用。

C.6.3 LRQ 阶段

该过程涵盖了单网守情形以及多层链接网守的情形。多网守情形中，网守 G（发起呼叫的域）宜使用（组播的）LRQ 定位被叫终端所在的网守 H，参见 ITU-T H.323 第 8.1.6 节“可选被叫终端信令”。两网守之间的通信应该采用附录 D 规定的安全保护方法。为此，假定网守之间存在可用的共享秘密 K_{GH}。由于网守间的 LRQ 消息通常采用组播发送，K_{GH} 通常不是对共享秘密（pair-wise shared secret），而是假设为一组网守的组内共享秘密。

注：这种假定限制了一般情形下的可伸缩性，不允许源认证。然而，在一个有限的、小数量的、网守彼此知道的公司网络内，这种限制仍然是可以接受的。使用数字签名保证中间网守通信安全能够克服这种限制，但这不是本标准所规定的范围。

如果 LRQ 方法被用来定位远端网守，那么 LRQ 应该携带一个包含 tokenOID 为“I20”的单独 ClearToken，ClearToken 的其他字段不使用。对于组播情况，LRQ 消息中的 ClearToken 的 generalID 字段不应该被使用。

C.6.4 LCF 阶段

网守 H 发现终端 A 和 B 都支持 DRC2，应该按如下指定过程产生 LCF 消息中的密钥材料和 ClearToken。

让 K_{BH} 表示终端 B 和网守 H 之间的共享秘密，EK_{BH} 代表终端 B 和网守 H 之间的加密密钥，KS_{BH} 为终端 B 和网守 H 之间共享的掺杂密钥。网守 H 产生一个随机挑战值 B。使用 C.12 节定义的基于 PRF 密钥推导过程从共享秘密 K_{BH} 推导出密钥 EK_{BH}，其中，挑战值 B 代替 **challenge**。CT_B→**h235Key**→**secureSharedSecret**→**keyDerivationOID** 应该包含“AnnexI-HMAC-SHA1-PRF”，参见附录 C.10。

网守 H 应该使用 C.8 定义的基于 PRF 密钥推导过程从共享秘密 K_{BH} 推导出掺杂密钥 KS_{BH}，其中，挑战值 B 代替 **challenge**。

EK_{GH} 代表网守 G 和网守 H 之间的加密密钥，KS_{GH} 代表网守 G 和网守 H 之间的掺杂密钥。网守 H 应该产生一个随机挑战值 G。网守 H 应该使用第 C.8 章定义的基于 PRF 密钥推导过程从共享秘密 K_{GH} 推导出密钥 EK_{GH}，其中，挑战值 G 代替 **challenge**，CT_{HG}→**h235Key**→**secureSharedSecret**→**keyDerivationOID** 应该包含“AnnexI-HMAC-SHA1-PRF”，参见附录 C.10。

网守 H 应该使用附录 C.8 定义的基于 PRF 密钥推导过程从共享秘密 K_{GH} 推导出掺杂密钥 KS_{GH}，其中，挑战值 G 代替 **challenge**。

网守 H 在 LCF 消息中创建了两个 ClearToken。CT_{HG} 给网守 G，CT_B 给被叫端 B。CT_{HG}→**tokenOID** 应该包含 OID“I23”，CT_B→**tokenOID** 应该包含 OID“I12”。

挑战值 G 应该放在 CT_{HG}→**challenge** 中，网守 H 的标识符应该放在 CT_{HG}→**sendersID** 中，网守 G 的标识符应该放在 CT_{HG}→**generalID** 中。

挑战值 B 应该放在 CT_B→**challenge** 中，网守 H 标识符应该放在 CT_{HG}→**sendersID** 中，终端 B 的标识符应该放在 CT_{HG}→**generalID** 中。如果 LRQ 消息中的 endpointIdentifier 字段包含了终端 A 的标识符，网守 H 应该把它拷贝到 CT_B→**h235Key**→**secureSharedSecret**→**generalID** 中，也应该把它拷贝到

CT_{HG}→**h235Key**→**secureSharedSecret**→**generalID** 中。

如果网守 H 和终端 B 支持 DRC2，LCF 响应应该包含 CT_{HG} 和 CT_B。

收到网守 H 的 LCF 消息后，网守 G 检查 CT_B 和 CT_{HG}。网守 G 使用挑战值 G 作为 **challenge** 以及 C.12 的 PRF 函数从 K_{GH} 计算出 KS_{GH} 和 EK_{GH}，并解密 CT_{HG}→**h235Key**→**secureSharedSecret**→**encryptedSessionKey** 获得终端 A 和 B 的共享密钥 K_{AB}。

C.6.5 ACF 阶段

网守 H 计算得到终端 A 和 B 之间的基于呼叫（call-based）的共享秘密 K_{AB}。该共享秘密然后通过 ClearToken 被推送到两个终端。这个 ClearToken 被送给发起端网守 G，然后网守 G 在 **ACF** 消息中送给主叫方。

网守 H 应该利用 EK_{GH} 按 $ENC_{EKHG,\ KSHG,\ IV}$（K_{AB}）加密 K_{AB}。把加密的 K_{AB} 放到 CT_{HG}→**h235Key**→**secureSharedSecret**→**encryptedSessionKey**。

增强的 OFB（EOFB）加密模式（参见附录 E.5.4）应该和该秘密及专用于终端的掺杂密钥 KS_{GH} 一起使用，使用的加密算法如下（参见附录 E.7）：

- 工作在 EOFB 模式的 DES（56 比特）算法，OID 为 Y1（可选）；
- 工作在外层 EOFB 模式的 3DES（168 比特）算法，OID 为 Z1（可选）；
- 工作在 EOFB 模式的 AES（128 比特）算法，OID 为 Z2（缺省且推荐）；
- 工作在 EOFB 模式的 RC2（56 比特）算法，OID 为 X（可选）。

对于 EOFB 加密模式，网守 H 应该产生一个随机的初始矢量 IV。对于 OID“X1”，OID“Y1”和 OID“Z1”算法，初始矢量为 64 比特，并且应该在 CT_{HG}→**h235Key**→**secureSharedSecret**→**params**→**iv8** 内传送；OID“Z2”算法需要 128 比特的初始矢量，应该在 CT_{HG}→**h235Key**→**secureSharedSecret**→**params**→**iv16** 内传递。

加密算法在 CT_{HG}→**h235Key**→**secureSharedSecret**→**algorithmOID** 中指示（“X1”，“Y1”，“Z1” or “Z2”）。对于 EOFB 加密算法，**encryptedSaltingKey** 不应该被使用。

类似地，网守 H 应该按 $ENC_{EKBH,\ KSBH,\ IV}$（K_{AB}）加密 K_{AB}。把加密的 K_{AB} 放到 CT_B→**h235Key**→**secureSharedSecret**→**encryptedSessionKey**。

增强的 OFB（EOFB）加密模式（参见附录 E.5.4）应该和该秘密及专用于终端的掺杂密钥 KS_{BH} 一起使用，使用的加密算法如下（参见表 E.6）：

- 工作在 EOFB 模式的 DES（56 比特）算法，OID 为 Y1（可选）；
- 工作在外层 EOFB 模式的 3DES（168 比特）算法，OID 为 Z1（可选）；
- 工作在 EOFB 模式的 AES（128 比特）算法，OID 为 Z2（缺省且推荐）；
- 工作在 EOFB 模式的 RC2（56 比特）算法，OID 为 X1（可选）。

对于 EOFB 加密模式，网守 H 应该产生一个随机的初始矢量 IV。对于 OID“X1”，OID“Y1”和 OID“Z1”算法，初始矢量为 64 比特，并且应该在 CT_B→**h235Key**→**secureSharedSecret**→**params**→**iv8** 内传送；OID“Z2”算法需要 128 比特的初始矢量，应该在 CT_B→**h235Key**→**secureSharedSecret**→**params**→**iv16** 内传递。

加密算法在 CT_B→**h235Key**→**secureSharedSecret**→**algorithmOID** 中指示（“X1”，“Y1”，“Z1”or“Z2”）。对于 EOFB 加密算法，不应该使用 **encryptedSaltingKey**。

终端 A 的 ACF 响应消息应该包含两个 ClearToken，CT_A 用于终端 A，CT_B 用于终端 B。CT_A→**tokenOID** 应该包含一个 OID "I11"。

网守 G 应该生成一随机挑战值 A，使用附录 C.8 定义的基于 PRF 密钥推导过程，从共享秘密 K_{AG} 推导出加密密钥材料 EK_{AG}。同时，挑战值 A 替代为 **challenge**，CT_A→**h235Key**→**secureSharedSecret**→**keyDerivationOID** 应该包含"AnnexI-HMAC-SHA1-PRF"，参见 E.12 节。

网守 G 应该使用一个加密算法 $ENC_{EKAG,\ KSAG,\ IV}$（K_{AB}）加密 K_{AB}。把加密的 K_{AB} 放到 CT_A→**h235Key**→**secureSharedSecret**→**encryptedSessionKey**。

增强的 OFB（EOFB）加密模式（参见附录 E.5.4）应该和该秘密及专用于终端的掺杂密钥 KS_{AG} 一起使用，使用的加密算法如下（参见表 E.6）：

- 工作在 EOFB 模式的 DES（56 比特）算法，OID 为 Y1（可选）；
- 工作在外层 EOFB 模式的 3DES（168 比特）算法，OID 为 Z1（可选）；
- 工作在 EOFB 模式的 AES（128 比特）算法，OID 为 Z2（缺省且推荐）；
- 工作在 EOFB 模式的 RC2（56 比特）算法，OID 为 X1（可选）。

对于 EOFB 加密模式，网守 G 应该产生一个随机的初始矢量 IV。对于 OID "X1"，OID "Y1" 和 OID "Z1" 算法，初始矢量为 64 比特，并且应该在 CT_{HG}→**h235Key**→ **secureSharedSecret**→**params**→**iv8** 内传送；OID "Z2" 算法需要 128 比特的初始矢量，应该在 CT_A→**h235Key**→**secureSharedSecret**→**params**→**iv16** 内传递。加密算法在 CT_A→**h235Key**→**secureSharedSecret**→**algorithmOID** 中指示（"X"，"Y1"，"Z1" or "Z2"）。

网守 G 的标识符（GKID）应该放在 CT_A→**sendersID** 中，CT_A→**generalID** 中保存终端 A 的标识符。终端 B 的标识符应该从 CT_B→**generalID** 中拷贝到 CT_A→**h235Key**→**secureSharedSecret**→**generalID**。

如果网守 G 之前未在 LRQ 的 endpointIdentifier 字段填写终端 A 的标识符，那么网守 G 应该把终端 A 的标识符填写到 CT_B→**h235Key**→**secureSharedSecret**→**generalID** 中。

对于 EOFB 加密算法，不应该使用 **encryptedSaltingKey**。

本附录定义的 **ClearToken** 可以结合其他的安全轮廓一起使用，例如附录 D 和 ITU-T H.235.3 所规定的安全轮廓。在这种情况下，本附录中的 **ClearToken** 也应该使用 **ClearToken** 的其他字段。例如，为了和附录 D 一起使用，**timestamp**，**random**，**generalID**，**sendersID** 和 **dhkey** 字段应该存在并被使用，参见附录 D。

网守 G 的标识符（GKID）应该放在 CT_A→**sendersID** 中，同时 CT_A→**generalID** 应该存放终端 A 的标识符。

终端 A 应该通过检查 CT_A→**tokenOID** "I21" 来识别 CT_A。终端 A 应该通过检查 **timestamp** 验证收到的 CT_A 是否及时（fresh）。然后应该进一步检查 ClearToken 的 **generalID** 和 **sendersID** 以及 **secureSharedSecret** 内的 **generalID**。如果 CT_A 是及时的，终端 A 应该获取初始矢量 IV 并按前述的针对网守 G 的描述计算 EK_{AG} 和 KS_{AG}。其中，CT_A→**challenge** 作为挑战值 A 替代附录 C.8 描述的 **challenge**。终端 A 应该解密 CT_B→ **secureSharedSecret**→ **encryptedSessionKey** 来获得 K_{AB}。

C.6.6 SETUP 阶段

终端 A 应该通过检查 ClearToken 的 **tokenOID** 是否为"I11"判别 CT_A。终端 A 应该通过检查 **timestamp** 验证收到的 CT_A 是否及时（fresh）。然后应该进一步检查 ClearToken 的 **generalID** 和 **sendersID** 以及

secureSharedSecret 中的 **generalID**。如果 CT_A 是及时的，终端 A 应该获取 IV 并按前面网守 G 的描述计算 EK_{AG} 和 KS_{AG}，其中挑战值 A 替代为附录 C.8 描述的 **challenge**。终端 A 应该解密 CT_A →**secureSharedSecret**→ **encryptedSessionKey** 来获得 K_{AG}。

如果收到的 CT_A 验证是及时的，终端 A 能够发送 SETUP 消息给终端 B。这个 SETUP 包含了 CT_B。SETUP 消息应该根据附录 D 的方式用共享密钥 K_{AB} 进行保护（认证和完整性）。为此，附录 D 规定的散列 ClearToken（不是 CT_B）中的 **generalID** 不应该被使用，除非终端 A 已经有 $EPID_B$（如通过静态配置）。如果终端 A 使用 $EPID_B$ 作为 SETUP 中的 generalID，那么终端 A 应该接受返回的呼叫信令消息中的 sendersID 值并作为真正的 $EPID_B$。

终端 B 应该通过检查 ClearToken 的 **tokenOID** 是否为“I12”判别 CT_B。

终端 B 应该通过检查 **timestamp** 验证收到的 CT_B 是否及时（fresh）。然后应该进一步检查验证 ClearToken 的 **sendersID** 和 **secureSharedSecret** 内的 **generalID**。如果 CT_B 被验证为及时的，终端 B 应该获取初始矢量 IV，并按前面网守 H 的描述计算 EK_{BH} 和 KS_{BH}，其中 CT_B→**challenge** 作为挑战值 B 替代为第 C.8 章描述的 **challenge**。终端 B 应该解密 CT_B→ **secureSharedSecret**→ **encryptedSessionKey** 来获得 K_{AB}。

对于 CT_B 验证是及时的情形，终端 B 能够通过回复合适的 CALL-PROCEEDING，ALERTING 或 CONNECT 消息，处理后续呼叫信令。对于 CT_B 验证是不及时情形或安全验证 SETUP 消息失败，终端 B 应该用 RELEASE-COMPLETE 消息进行答复，并且 **ReleaseCompleteReason** 设置为安全错误原因，如由 ITU-T H.235.0 的第 11.1 节所定义。

当部署了媒体安全时（附录 E），终端 A 和 B 应该使用 Diffie-Hellman 密钥协商过程建立动态的主密钥，并由主密钥保护媒体会话密钥。

终端 B 应该设置 **generalID 为** $EPID_A$ 和设置 **sendersID 为** $EPID_B$，以保护目的地为终端 A 的所有 H.225.0 呼叫信令消息。（例如，CALL-PROCEEDING，ALERTING 和 CONNECT 消息）

图 C.3 所示为 DRC2 基本通信流程。

图C.3 基本通信流程（DRC2）

C.7 规程 DRC3（域间环境）

本章描述的过程应用于域间通信环境。此环境中，主叫终端不支持 DH 算法，但主叫域和被叫域的网守支持 DH 过程。在这样的环境中，通过在主叫网守和被叫网守之间交换 DH 参数来计算会话密钥。

C.7.1 GRQ/RRQ 阶段

此场景覆盖多层链接的网守。在 GRQ 和/或 RRQ 阶段，支持本安全轮廓的终端应该包含一个 tokenOID 为“I30”单独的 ClearToken，其中，ClearToken 其他字段未使用。支持本安全轮廓的网守如果愿意提供 DRC3 功能应该在 GCF 或 RCF 中包含一个 tokenOID 为“I30”的单独 ClearToken，其中，ClearToken 其他字段未使用。

C.7.2 ARQ 阶段

在终端 A 开始向终端 B 直接发送呼叫信令消息之前，终端 A 向网守 G 请求接入许可，ARQ 消息中包含一个 tokenOID 为“I30”的单独 ClearToken 字段，ClearToken 其他字段未使用。

C.7.3 LRQ 阶段

网守 G 收到终端 A 的 ARQ 请求后，由于终端 B 不属于网守 G 的管理域，网守 G 向网守 H 发送一个 LRQ 消息，请求终端 B 的地址。

网守 G 产生一个 **LRQ** 消息，该消息包含一个 tokenOID 为“I30”的 ClearToken（在 CryptoHashedToken 内），用来告诉网守 H 需要 DH 密钥协商过程。ClearToken 的 dhkey 字段填写网守 G 产生的 DH 参数（g，p，g^x），其他字段不使用。

网守 G 把 LRQ 消息发送给网守 H。假如是网守云的情况，网守 G 把 LRQ 消息发给最邻近的邻居网守，后者把 LRQ 消息发送给它的最邻近的邻居网守。该转发过程持续直到 LRQ 消息最后到达网守 H。

对于组播的情况，**LRQ** 的 CryptoToken 字段的 generalID 不应该使用。如果网守 G 不能够定位终端 B，那么网守 G 给终端 A 返回 ARJ 消息。两个网守之间的通信应该根据附录 D 中的方法进行保护。

如果网守 G 不支持本轮廓，网守 G 可以自由选择是否回到 DRC2 或发送 ARJ 给终端 A，如果选择了 DRC2，那么所有后续的步骤（包括 LRQ 步骤）和 DRC2 相同。

C.7.4 LCF 阶段

在网守 H 收到网守 G 的 LRQ 消息后，如果发现终端 A 和 B 都支持 DRC3，应该按如下所述产生会话密钥 K_{AB}。

首先，网守 H 产生一个随机挑战值 B，该值应该保存到 CT_B→**challenge** 中，并且 CT_B→**h235Key**→**secureSharedSecret**→**keyDerivationOID** 应该包含“AnnexI-HMAC-SHA1-PRF”，然后使用共享密钥 K_{GH} 和挑战值 B 并采用基于 PRF 的密钥推算过程推算密钥材料 EK_{GH} 和掺杂密钥 KS_{GH} 。

挑战值 B 应该保存在 CT_B→**challenge** 中，网守 H 的标识符保存在 CT_B→**sendersID** 中，终端 B 的标识符保存在 CT_B→**generalID** 中。如果 **LRQ** 的 endpointIdentifier 字段存在终端 A 的标识符，那么网守 H 应拷贝它到字段 CT_B→**h235Key**→**secureSharedSecret**→**generalID** 和 CT_{HG}→**h235Key**→**secureSharedSecret**→**generalID** 中。

网守 H 在 LCF 消息中创建了两个 ClearToken。CT_{HG} 用于网守 G，CT_B 用于被叫终端 B。CT_{HG}→**tokenOID** 应该包含 OID “I33”，CT_B→**tokenOID** 应该包含 OID “I12”。网守 H 创建被叫方的 DH 参数（g，p，g^y）。使用从 LRQ 消息中获得的主叫方的 DH 参数，网守 H 应该计算出会话密钥 $K_{AB} = g^{xy}$ 模 p。

最后，网守 H 应该按 $ENC_{EKBH,\ KSBH,\ IV}$（K_{AB}）加密 K_{AB}。把加密的 K_{AB} 放到 CT_B→**h235Key**→**secureSharedSecret**→**encryptedSessionKey**，并把被叫方的 DH 参数放到 CT_{HG} 的 **dhkey** 字段中。

增强的 OFB（EOFB）加密模式（参见附录 E.5.4）应该和该秘密及专用于终端的掺杂密钥 KS_{GH} 一起使用，使用的加密算法如下（参见表 E.6）：

- 工作在 EOFB 模式的 DES（56 比特）算法，OID 为 Y1（可选）；
- 工作在外层 EOFB 模式的 3DES（168 比特）算法，OID 为 Z1（可选）；
- 工作在 EOFB 模式的 AES（128 比特）算法，OID 为 Z2（缺省且推荐）；
- 工作在 EOFB 模式的 RC2（56 比特）算法，OID 为 X1（可选）。

对于 EOFB 加密模式，网守 H 应该产生一个随机的初始矢量 IV。对于 OID “X1”，OID “Y1” 和 OID “Z1” 算法，初始矢量为 64 比特，并且应该在 CT_{HG}→**h235Key**→ **secureSharedSecret**→**params**→**iv8** 内传送；OID “Z2” 算法需要 128 比特的初始矢量，应该在 CT_{HG}→**h235Key**→**secureSharedSecret**

→params→iv16 内传递。

加密算法在 CT_B**→h235Key→secureSharedSecret→algorithmOID** 中指示（“X1”，“Y1”，“Z1”或“Z2”）。对于 EOFB 加密算法，不应该使用 **encryptedSaltingKey**。

网守 H 给网守 G 发送 LCF 消息。如果网守云存在，那么 LCF 消息通过中继的方式传递。延着这条路径，每个网守从上游的邻居网守接收 LCF 消息，检查 LCF 消息是否包含 CT_{HG}，然后把消息转发给下游的邻居网守。

如果网守 H 不支持 DH 算法或者是安全策略不运行 DRC3，将回到 DRC2 的情况，那么所有后续的步骤（包括 LRQ 步骤）和 DRC2 相同。

C.7.5 ACF 阶段

当收到 LCF 消息后，网守 G 发现（LCF 消息中的）一个 ClearToken 的 **tokenOID** 被设置为“I33”，从该 ClearToken 中获得被叫网守的 DH 参数。网守 G 按下面的方式产生一个 **tokenOID** 设置为“I11”的 ClearToken（用 CT_A 表示）。

首先，网守 G 产生一个随机挑战值 A，该值保存到 CT_A**→challenge** 中，并且 CT_A**→h235Key→secureSharedSecret→keyDerivationOID** 应该包含“AnnexI-HMAC-SHA1-PRF”，然后使用共享密钥 K_{AG} 和挑战值 A 并采用基于 PRF 的密钥推算过程推算密钥材料 EK_{AG} 和掺杂密钥 KS_{AG}。

其次，网守 G 使用主叫方的 DH 参数（LRQ 阶段获得）和被叫方的 DH 参数计算共享密钥 $K_{AG} = g^{xy}$ 模 p。然后，网守 G 从 LCF 消息中拷贝 CT_B 到 ACF 消息中，**tokenOID** 设置为“I12”。

最后，网守 G 应该按 $ENC_{EKAG,\ KSAG,\ IV}(K_{AB})$ 加密 K_{AB}。把加密的 K_{AB} 放到 CT_A**→h235Key→secureSharedSecret→encryptedSessionKey**，并从 LCF 消息中拷贝 CT_B 到 ACF 消息中。

增强的 OFB（EOFB）加密模式（参见附录 E.5.4）应该和该秘密及专用于终端的掺杂密钥 KS_{AG} 一起使用，使用的加密算法如下（参见表 E.6）：

- 工作在 EOFB 模式的 DES（56 比特）算法，OID 为 Y1（可选）；
- 工作在外层 EOFB 模式的 3DES（168 比特）算法，OID 为 Z1（可选）；
- 工作在 EOFB 模式的 AES（128 比特）算法，OID 为 Z2（缺省且推荐）；
- 工作在 EOFB 模式的 RC2（56 比特）算法，OID 为 X1（可选）。

对于 EOFB 加密模式，网守 G 应该产生一个随机的初始矢量 IV。对于 OID“X1”，OID“Y”和 OID“Z1”算法，初始矢量为 64 比特，并且应该在 CT_{HG}**→h235Key→ secureSharedSecret→params→iv8** 内传送；OID“Z2”算法需要 128 比特的初始矢量，应该在 CT_{HG}**→h235Key→secureSharedSecret →params→iv16** 内传递。加密算法在 CT_A**→h235Key→secureSharedSecret→algorithmOID** 中指示(“X1”，“Y1”，“Z1” or “Z2”)。

如果发现 ClearToken（在 LCF 消息中）的 **tokenOID** 为“I23”，可以判断发现了退回到 DRC2 的情况，网守 G 自由选择是否接受网守 H 的安全策略。如果接受，ACF 步骤已及后续的 Setup 步骤将和 DRC2 相同。否则，响应一个指示安全失败的拒绝消息，拒绝原因设为 securityDenial。

网守 G 给终端 A 发送 ACF 消息。

C.7.6 SETUP 阶段

终端 A 应该通过检查 ClearToken 的 **tokenOID** 是否为“I11”判别 CT_A。终端 A 应该通过检查 **timestamp** 验证收到的 CT_A 是否及时（fresh）。然后应该进一步检查 ClearToken 的 **generalID** 和 **sendersID** 以及

secureSharedSecret 中的 **generalID**。如果 CT_A 是及时的，终端 A 应该获取 IV 并按前面网守 G 的描述计算 EK_{AG} 和 KS_{AG}，其中挑战值 A 替代为 C.12 节描述的 **challenge**。终端 A 应该解密 CT_A →**secureSharedSecret**→ **encryptedSessionKey** 来获得 K_{AG}。

如果收到的 CT_A 验证是及时的，终端 A 能够发送 SETUP 消息给终端 B。这个 SETUP 包含了 CT_B。SETUP 消息应该根据附录 D 或 ITU-T H.235.3 所规定的方式用共享密钥 K_{AB} 进行保护（认证和完整性）。为此，附录 D 规定的散列 ClearToken（不是 CT_B）中的 **generalID** 不允许使用，除非终端 A 已经有 $EPID_B$（如通过静态配置）。如果终端 A 使用 $EPID_B$ 作为 SETUP 中的 generalID，那么终端 A 应该接受返回的呼叫信令消息中的 sendersID 值并作为真正的 $EPID_B$。

终端 B 应该通过检查 ClearToken 的 **tokenOID** 是否为“I12”判别 CT_B。

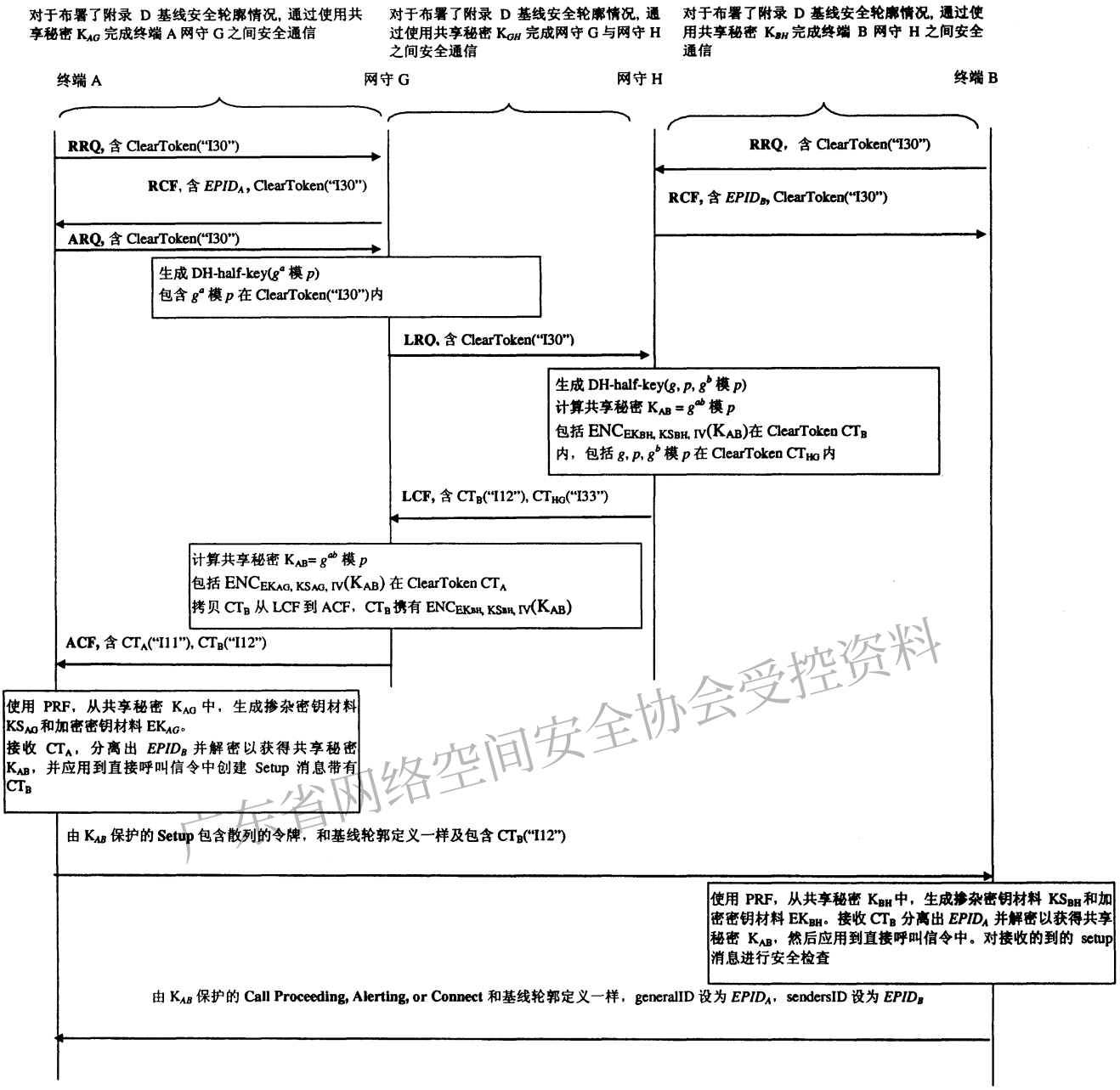
终端 B 应该通过检查 **timestamp** 验证收到的 CT_B 是否及时（fresh）。然后应该进一步检查 ClearToken 的 **sendersID** 和 **secureSharedSecret** 内的 **generalID**。如果 CT_B 是及时的，终端 B 应该获取初始矢量 IV，并按前面网守 H 的描述计算 EK_{BH} 和 KS_{BH}，其中 CT_B→**challenge** 作为挑战值 B 替代为附录 C.8 描述的 **challenge**。终端 B 应该解密 CT_B→ **secureSharedSecret**→ **encryptedSessionKey** 来获得 K_{AB}。

假如 CT_B 验证是及时的，终端 B 能够回复合适的 CALL-PROCEEDING，ALERTING 或 CONNECT 消息，呼叫继续。如果 CT_B 验证是不及时或安全验证 SETUP 消息失败，终端 B 应该响应的 RELEASE-COMPLETE 消息，且 **ReleaseCompleteReason** 设置为安全错误原因。

当部署了媒体安全时（附录 E），终端 A 和 B 应该使用 Diffie-Hellman 密钥协商过程建立动态的主密钥，并由主密钥保护媒体会话密钥。

终端 B 应该设置 **generalID** 为 $EPID_A$ 和设置 **sendersID** 为 $EPID_B$，以保护目的地为终端 A 的所有 H.225.0 呼叫信令消息。（例如，CALL-PROCEEDING，ALERTING 和 CONNECT 消息）

图 C.4 所示为 DRC3 基本通信流程。



图C.4 基本通信流程（DRC3）

C.8 基于 PRF 的密钥推算过程

本章描述了如何从共享密钥和其他参数推算密钥材料的过程。

本章中的过程允许从共享密钥中推算加密密钥和掺杂密钥。本过程对于共享密钥（K_{AG}，K_{BH} 和 K_{GH}）有相同的处理过程。

为了获得最终的密钥材料（如EK_{AG}），PRF函数和表C.1中的参数一起使用，其中*inkey*参数设置为共享密钥（如K_{AG}），label设置为相应的常量（0x2AD01C64 ‖ 挑战值A，其中‖表示串接）。*outkey_len* 设置为最终需要的密钥材料的长度，该长度和加密算法相关。

注：对于EK_{AG}、KS_{AG}、EK_{BH}和KS_{BH}，32位的整数常量（既0x2AD01C64等）取自常数e（即2.71828…）的十进制数字。对于EK_{GH} 和KS_{GH}，32位的整数常量（即0x2AD01C64等）取自常数π（即3.14159…）的十进制数字。对于EK_{AG}、KS_{AG}、

EK_{BH}和KS_{BH}，32位的整数常量由9位十进制数字块组成，分别是第一、第二、第四、第七块。对于EK_{GH}，32位的整数常量由π的前10个数字组成，而对于KS_{GH}，32位的整数常量则由紧接的8个数字组成。

表C.1 从共享秘密计算加密和掺杂密钥

目标密钥	PRF inkey	常量 ‖ 挑战值
EK_{AG}	K_{AG}	0x2AD01C64 ‖挑战值 A
KS_{AG}	K_{AG}	0x150533E1 ‖挑战值 A
EK_{BH}	K_{BH}	0x1B5C7973 ‖挑战值 B
KS_{BH}	K_{BH}	0x39A2C14B ‖挑战值 B
EK_{GH}	K_{GH}	0x54655307 ‖挑战值 G
KS_{GH}	K_{GH}	0x35855C60 ‖挑战值 G

C.9 伪随机函数

本章定义一个伪随机函数，其目的是从一个静态密钥材料和随机数中推算动态密钥。

该密钥推算方法有下列输入参数。

- *inkey*：输入的密钥。
- *inkey_len*：输入密钥的长度。
- *label*：一个特殊的标记，依赖于输出密钥类型和随机的挑战值。
- *outkey_len*：输出密钥的比特位长度。

伪随机函数有下面的输出：

- *outkey*：期望长度的输出密钥。

该 PRF 应该使用 IETF RFC3830 第 4.1.2 节定义的伪随机函数，如下所述。

假设 HMAC（参见[IETF RFC2104]）是基于 SHA-1（参见[ISO 10118-3]）的认证函数，定义

$P(s, label, m) =$ HMAC$(s, A_1 \| label) \|$
HMAC$(s, A_2 \| label) \| ...$
HMAC$(s, A_m \| label)$

其中：$A_0 =$ label，$A_i =$ HMAC(s, A_{i-1})。

下面的过程描述了伪随机函数计算 outkey 的方法，用 PRF（inkey，label）表示。

- 令 $n =$ inkey_len $/ 512$，取整为最近的整数；
- 把 inkey 分割为 n 块，inkey $= s_1 \| ... \| s_n$，其中，除了s_n外的所有的s_i均为512比特；
- 令 $m =$ outkey_len $/ 160$，取整为最近的整数。

然后，输出的密钥 outkey，取至下面结果的 outkey_len 个最高的比特位：

PRF（inkey，label）$= P(s_1, label, m)$ *XOR* $P(s_2, label, m)$ *XOR* ... *XOR* $P(s_n, label, m)$

C.10 对象标识符列表

表C.2 使用到的对象标识符

对象标识符引用	对象标识符值	描　述
"I10"	{itu-t（0）recommendation（0）h（8）235 version（0）3 48}	DRC1过程使用，在GRQ/RRQ、GCF/RCF 和ARQ中指示终端/网守支持DRC1
"I11"	{itu-t（0）recommendation（0）h（8）235 version（0）3 49}	DRC1、DRC2和DRC3过程使用，用于 ClearToken的tokenOID ，指示 CTA 保存一个给主叫方的端到端的密钥
"I12"	{itu-t（0）recommendation（0）h（8）235 version（0）3 50}	DRC1、DRC2和DRC3过程使用，用于 ClearToken的tokenOID ，指示 CTB 保存一个给被叫方的端到端的密钥
"I13"	{itu-t（0）recommendation（0）h（8）235 version（0）3 52}	DRC1过程使用，用于网守之间的ClearToken的tokenOID中，指示CTHG 保存一个给发起方网守的加密密钥
"I20"	{itu-t（0）recommendation（0）h（8）235 version（0）4 53}	DRC2过程使用，在GRQ/RRQ、GCF/RCF 和ARQ中指示终端/网守支持DRC2
"I23"	{itu-t（0）recommendation（0）h（8）235 version（0）4 56}	DRC2过程使用，用于网守之间的CTHG的tokenOID中，指示CTHG 保存一个给发起方网守的加密密钥
"I30"	{itu-t（0）recommendation（0）h（8）235 version（0）4 34}	在 GRQ/RRQ，GCF/RCF，ARQ 中 ClearToken 中使用，指示支持 DRC3，在 LRQ 消息中的 ClearToken 中使用指示携带主叫方的 DH 参数
"I33"	{itu-t（0）recommendation（0）h（8）235 version（0）4 37}	在 LCF 消息中的 ClearToken 中使用指示携带被叫方的 DH 参数
"Annex I-HMAC-SHA1-PRF"	{itu-t（0）recommendation（0）h（8）235 version（0）3 51}	DRC1、DRC2和DRC3过程使用，用于V3KeySyncMaterial 中的keyDerivationOID字段中，指示使用HMAC-SHA1 伪随机函数

附 录 D
（规范性附录）
基线安全轮廓

D.1 范围

本附录规定了使用基于口令的HMAC-SHA1-96散列算法对H.225.0 RAS和呼叫信令消息、隧道H.245消息进行认证和完整性保护、或只认证的方法。

本附录适用于终端—网守、网守—网守、网关—网守之间的通信。

D.2 约定

本附录定义了基线安全轮廓。基线安全框架利用基于口令的安全技术的简单手段提供基本安全能力。基线安全轮廓可以结合语音加密安全轮廓一起使用，达到媒体保密目的。

本附录利用H.323信令的H.235相关字段来提供认证/完整性验证等安全服务。不同的对象标识符（参见D.12节）用于确定使用哪种安全服务和哪个协议版本。过程1说明了如何使用基于口令的散列算法实现安全功能。本附录中，对象标识符用一个符号来引用（例如“A”），参见D.12节。

所有逐跳的安全信息放在**CryptoHashedToken**字段内，该信息在每一跳中被重新计算。

一般情况下，口令、会话密钥和共享秘密3个术语都用于对称加密技术中，有相同的作用。口令和会话密钥的不同之处在于应用背景，口令应用于认证和授权，而会话密钥则用于加密。共享秘密是一个中性词汇，并不针对具体应用场景。

口令（也可看成共享秘密）用于RAS和呼叫信令消息（包括隧道H.245消息）认证和完整性验证，口令可以在初始阶段由用户输入。口令通常有一段较长时间的生存期，口令是由管理机构（如运营商）预先分配的，可在用户订阅过程中产生。为了方便协议处理，需要使用某些算法把初始口令转换为20字节固定长度的形式。本附录应该使用SHA1算法进行上述转化，即共享秘密 = SHA1（口令）。

会话密钥用于媒体流加密，由主从决定的主角色产生，每个RTP会话有单独的会话密钥（OLC过程建立一个RTP会话），密钥生存期为呼叫持续时间。会话密钥由另一密钥加密，后者由通信双方通过Diffie-Hellman共享秘密交换协议协商出来。这里，保护会话密钥的密钥称为主密钥。

ClearToken包含一个类型为32位的整数的**random**字段。该字段的用法：初始值可任意设置，以后针对每个输出的消息，加1递增。这个字段作为HASH函数输入参数的一部分，提供了附加的随机性，可弥补多个具有相同**timestamp**字段的连续消息的特殊情况。输入参数的不同取值导致HASH计算结果的不同，这是防止回放攻击的依据。接收端可以设置一个时间窗，用来跟踪收到的**timestamp**和**random**，如果在窗口范围内收到两次相同的**timestamp**和**random**，则意味着有回放攻击发生。

本轮廓要求设置**ClearToken**的**GeneralID**字段为接收者标识符。这意味着，发给网守的RAS消息，**GeneralID**为GKID；发给终端的RAS消息，**GeneralID**为EPID。类似地，发给网守的呼叫信令消息，**GeneralID**为GKID；发给终端的呼叫信令消息，**GeneralID**为EPID。具体参见附录D.11。

ClearToken的**sendersID**应该设置为发送者的标识符。这意味着，发给网守的RAS消息，**sendersID**为EPID；发给终端的RAS消息，**sendersID**为GKID。类似的，发给网守的呼叫信令消息，**sendersID**为EPID；发给终端的呼叫信令消息，**sendersID**为GKID。具体参见D.11节。

一个块是指块加/解密算法支持的数据包的基本运算单位。对于DES和3DES算法，块大小是64比特，对于AES来说，块大小是128比特。

本附录可提供整个消息的完整性保护。对于RAS消息，涵盖整个RAS消息；对于呼叫信令消息，涵盖包括Q.931头在内的整个呼叫消息。

为了避免引用RC2商标，本附录称RC2算法为RC2-兼容算法。

D.3 基线安全轮廓

D.3.1 概述

本轮廓提供下列特征：针对RAS、H.225.0呼叫信令和H.245消息，提供了基于HOP-BY-HOP的消息认证和完整性验证。

注：不提供端到端的安全特性。

可解决下面几种安全攻击。

- DoS 攻击：迅速检查 HASH 摘要值可阻止这样的攻击。
- 中间人攻击：应用层的 HOP-BY-HOP 消息认证和完整性检查，可阻止中间人强行插入通信环节，例如敌意的路由器。
- 回放攻击：使用时戳和序列号阻止这类攻击。
- 欺骗：使用身份认证来阻止这类攻击。
- 盗用连接：针对所有消息进行认证/完整性验证可阻止这类攻击。

基线安全的其他亮点有：

- 采用来源于 IMTC/ETSI/IETF 组织的可靠的、成熟的、广泛部署的算法。
- 能够根据商业模型的安全需求，分阶段实施。
- 可以应用于许多场合，例如工作组内部、可以扩充的开放环境、多点会议应用。
- 只认证安全框架可以应用于需要 NAT/防火墙穿越的场合。

表D.1总结了本附录涵盖的处理各种安全需求的所有规程。垂直阴影线表示基线安全框架内容，只认证安全框架用斜阴影线表示。

表D.1 基线安全轮廓

安全服务	呼叫功能			
	RAS	H.225.0	H.245[注e]	RTP
认证	口令 HMAC-SHA1-96	口令 HMAC-SHA1-96	口令 HMAC-SHA1-96	
只认证	口令 HMAC-SHA1-96	口令 HMAC-SHA1-96	口令 HMAC-SHA1-96	
不可否认				
完整性	口令 HMAC-SHA1-96	口令 HMAC-SHA1-96	口令 HMAC-SHA1-96	
保密				
接入控制				
密钥管理	基于订阅的口令分配			

注：隧道H.245或H.225.0 Fast Connect

访问控制方法不明确描述，可以利用收到的H.235信令字段（ClearToken、CryptoToken）中的信息来实现访问控制。

本附录不说明基于订阅的密钥分配过程。密钥分配方法不属于本附录的范围。

通信实体可以检查消息中的安全OID（tokenOID、algorithmOID，参见D.12节）来决定到底是否启用了基线安全轮廓。

D.3.2 基线安全轮廓应用能力

基线安全轮廓可以应用于口令分配给安全的H.323实体（终端）和网络元素（GK、Proxy）的情况。所谓安全H.323实体，是指设备本身是可信赖的。它使用基于口令的HMAC-SHA1-96散列算法进行RAS、呼叫信令、H.245信令的认证和完整性检查或只认证，算法细节见过程1说明。呼叫建立信令集成了Diffie-Hellman密钥交换过程。基线安全要求H.245信令嵌入到H.225呼叫信令，即不存在单独的H.245信令通道。

D.3.3 H.323 协议需求

1）GK应该支持路由模式；

2）终端应该支持Fast Connect过程或隧道H.245之一。对于多点会议应用，为了提供丰富的会议控制能力，隧道H.245是不可缺少的。

D.3.4 协议过程概述

本节描述了下面两个过程。

过程1是一种简单的基于对称密钥的消息认证机制，该机制需要两个实体之间共享口令。该过程提供了RAS、呼叫信令、H.245信令的认证和完整性验证。

过程1A是一种简单的基于对称密钥的只认证机制，该机制需要两个实体之间共享口令。该过程不提供消息完整性验证。过程1A可以用于NAT/防火墙穿越的场合。

根据应用安全策略，通信双方可以只进行单向的认证/完整性验证（指终端向网守认证），也可以是相互进行认证/完整性验证。网守决定是否应用反方向的认证/完整性检查。

网守检测到对端来的RAS或呼叫信令消息认证/完整性验证失败时，应该响应一个拒绝消息，其拒绝原因为**securityDenial**或ITU-T H.235.0中定义的其他安全错误码。依赖于网守识别安全攻击的能力，一个网守如果收到一个包含有未定义的（非标的）OID（**tokenOID**，**algorithmOID**）的安全xRQ消息，可以选择响应拒绝原因为**securityDenial**的无安全xRJ消息，或是简单地抛弃消息，并建议记录此安全事件。依赖于安全策略，如果终端要求对收到RAS消息进行认证/完整性验证，则应该抛弃收到的不安全的RAS消息（防止某些设备发送恶意的干扰消息）。类似地，网守收到一个包含有未定义的（非标的）OID（**tokenOID**，**algorithmOID**）的安全的SETUP消息，可以选择响应一个RELEASE COMPLETE消息，拒绝原因设为**securityDenied**，也可以简单地抛弃这个消息。

根据消息中安全OID的值（参见D.12节）和相关字段的设置，间接指示了是否使用了过程1安全机制。

本框架不使用ICV字段，代之用散列值进行完整性验证，散列值放在**CryptoToken**的散列字段中。

D.4 安全基线过程 1

当使用过程1时，应该遵守下面过程。

1）HMAC-SHA1-96算法产生一个12字节（96位）散列值作为认证值，如果密钥是从口令转换而来，那么应该使用D.5节中的描述把口令转换为20字节密钥。

注：为了安全要求，要求口令有很好的随机性。

2）RAS和呼叫信令消息的**CryptoH323Token**应该包含下列字段：

— **nestedCryptoToken** 包含 **CryptoToken**， **CryptoToken** 包含 **cryptoHashedToken**，**cryptoHashedToken**包含下面的字段：

- **tokenOID**设置为“A”，指示认证/完整性验证包含整个RAS消息或H.225.0呼叫信令消息。
- **hashedVals**包含**ClearToken**字段，**ClearToken**含有下列字段：

— **tokenOID**设置为“T”，指示这个**ClearToken**用于本附录定义的认证/完整性验证过程。其他的**tokenOID**值则用非本附录定义的过程。

— **timeStamp**包含时戳。

— **random**包含一个单调增加的随机数（每个消息加1），该数字用于区分两个具有相同时戳的消息。

— **generalID**包含消息接收者标识符。

— **sendersID**包含消息发送者标识符。

— **dhkey**，包含本附录中规定的Diffie-Hellman密钥交换的参数，Diffie-Hellman密钥交换发生在Setup-Connect阶段。

- **halfkey**包含一个随机的公钥。
- **modsize**包含Diffie-Hellman算法规定的素数。
- **generator**包含Diffie-Hellman算法规定的基数。

注：当基线安全框架不和声音加密安全框架一起使用时，**dhkey**应该不存在，或者是设置为{ ‘0’， ‘0’， ‘0’。

注：呼叫发起方可以在Setup消息中提供多个dhkey，给应答方提供多组（**modsize**，**generator**）参数的选择机会，应答方应该选择其中之一在CONNECT消息中返回。

— **token**含有**HASHED**字段，**HASHED**包含下面字段：

- **algorithmOID**设置为“U”，指示使用HMAC-SHA1-96
- **params**设置为NULL
- **hash**则包含有用HMAC-SHA1-96计算得到的12字节的认证值。散列计算应该针对整个消息。

认证值应该在消息经过的每个H.323节点中被验证，然后重新计算后，发给消息路径中的下一节点（EP1-GK1、GK1-GK2、GK2-EP2、EP1-GK2）。

注：认证值是基于消息为单位计算的。

注：应该使用SHA1标准（ISO/IEC 10118-3）规定的填充方法。

注：为了防止回放攻击的可能性，强烈建议在random序号轮回前，改变口令。

注：通过检查**tokenOID**（是否为“A”），接收端可以探测发送方是否在使用过程1。

D.4.1 计算散列值

发送端和接收端分别计算整个ASN.1编码后的消息的散列值。对于只认证安全框架，只计算ASN.1编码的**ClearToken**（OID为“T”的**ClearToken**）字段的散列值。

D.4.2 HMAC-SHA1-96

HMAC-SHA1-96截取160位HMAC-SHA1算法计算得到的散列值的96位。96位取自网络字节序的最左边96位。IETF RFC2104描述了HMAC-SHA1的细节，具体使用中，HMAC-SHA1的密钥K=SHA1（用户口令），文本（text）为ASN.1消息。

D.4.3 认证/完整性验证步骤

消息发送端计算散列值步骤如下：

1） 设置消息中存放散列值的位置为一96位长的缺省值，缺省值可以任意选择，但建议该缺省值不易出现在消息的其他字段中。

2） 对整个消息进行ASN.1编码。对于RAS，包括整个消息；对于呼叫信令消息，包括Q.931部分在内的整个H.225.0呼叫信令消息（不含TPKT部分）。

3） 在编码后的消息比特流中寻找和缺省值的比特流匹配的位置，用96个0比特替换之。

注：如果消息流中出现多处匹配的位置，那么建议重新选择一个缺省值，重新开始本过程。

4） 对步骤3）的消息计算散列值。

5） 用得到的散列值替换步骤3）中的96个0比特。

消息接收者处理过程如下：

1） 对消息进行ASN.1解码。

2） 抽取出收到的散列值，保存在一个局部变量RV中。

3） 在未解码的ASN.1消息中查找和定位出现RV值的比特流模式的位置。

4） 用96个0比特替换步骤3）中散列值的位置。

5） 计算步骤4）处理后的消息的散列值。

6） 比较步骤5）的散列值是否和RV相同，如果相同，则认证成功；否则，用RV值恢复被96个替换的内容，在消息的其他位置中继续查找和定位出现RV值的比特流模式的位置，重新进行步骤3~7。如果处理完整个消息后，认证都不成功，则过程结束，认证失败。

D.5 只认证（过程1A）

终端可以选择只实现认证过程。其计算步骤和过程1大同小异。有区别的地方如下：

1）过程1中的OID“A”用“B”代替，表示只进行认证。

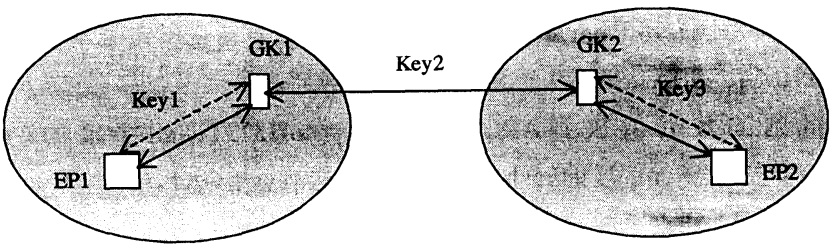
2） 计算散列值时，只针对RAS/H.225.0呼叫信令的一部分（**CryptoToken**的**ClearToken**部分，且**ClearToken**的OID为“T”）。

D.6 基线安全轮廓使用方法举例说明

图D.1~D.3描述了在不同网守部署模式下的共享密钥使用情况。不管何种呼叫路由模型，终端和网守之间都需要一个共享秘密，用来保证RAS消息的安全性。当RAS通道和H.225.0呼叫通道连接相同的两个节点时，同一个口令可以用来保护RAS和H.225.0呼叫信令。

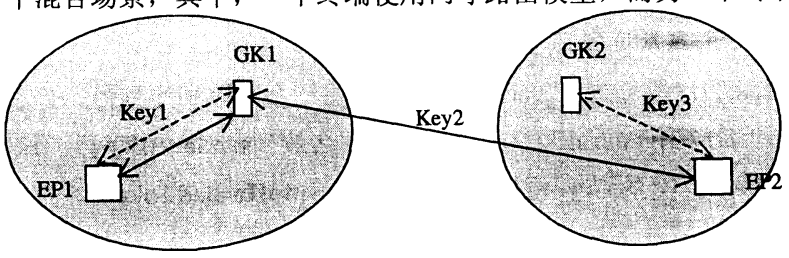
图D.1显示了最具有扩展性的场景。网守使用呼叫路由模型，两个网守之间共享口令。

注：该场景不具备端到端的安全性，安全性依赖于中途的可信任的网守。



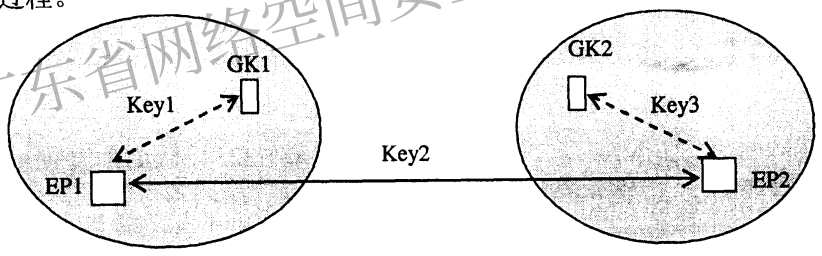
图D.1 两个GK都采用呼叫路由的情况

图D.2显示了一个混合场景，其中，一个终端使用网守路由模型，而另一个终端使用直接路由模型。



图D.2 GK1采用呼叫路由，GK2采用直接路由模型

图D.3显示了两个终端在不同区域中，两个网守都使用直接路由的情况。原则上，建议使用基于证书的安全框架定义的过程。



图D.3 两个GK都采用直接路由模型

考虑图D.1的情况，其中，三个密钥分别在EP1-GK1、GK1-GK2和GK2-EP2之间共享。三个密钥根据D.5节描述的算法各自从共享的口令转化而来。为了达成最大安全性，建议三个密码是无关联的。

下面说明RAS、H.225.0呼叫信令和H.245消息的认证和完整性验证。例子参数针对呼叫路由模型。

D.6.1 RAS消息认证和完整性验证

考虑EP1发送一个ARQ消息给GK1的情况。EP1产生一个时戳和序列号，分别添加到**timeStamp**和**random**字段。把GK1别名放到**generalID**字段，把终端别名放到**sendersID**字段。这些字段存在于ARQ消息的**ryptoH323Token**字段的**CryptoToken**字段的**cryptoHashedToken**字段的**hashedVals**字段的**ClearToken**字段中。

cryptoHashedToken的**tokenOID**设置为“A”，表示对整个ARQ消息计算散列值。如果设置为“B”，则只针对**ClearToken**部分计算散列值。**cryptoHashedToken**的**token**字段的**algorithmOID**设置为“U”，

指示散列算法为HMAC-SHA1-96，**params**设置为NULL（即所有可选字段都不存在）。EP1然后使用20字节密钥KEY1以及HMAC-SHA1-96算法计算整个消息的散列值。

EP1把计算得到的散列值放到ARQ消息的**cryptoH323Token**字段的**CryptoToken**字段的**cryptoHashedToken**字段的**token**字段的**hash**字段。然后发送消息给GK1。

GK1收到ARQ消息后，根据下面的原则验证散列值:

- **timestamp** 是否及时，**random** 是否惟一。
- 检查 **generalID** 是否等于自己的 ID。
- 检查 **sendersID** 是否为 EP1 的 ID。
- 比较重新计算的散列值是否和 ARQ 消息中的散列值一样。

D.6.2 H.225.0 呼叫信令消息认证和完整性验证

考虑EP1发送一个SETUP消息给EP2的情况。EP1产生一个时戳和序列号，分别添加到**timeStamp**和**random**字段。把GK1别名放到**generalID**字段，把终端别名放到**sendersID**字段。这些字段存在于SETUP消息的**cryptoH323Token**字段的**CryptoToken**字段的**cryptoHashedToken**字段的**hashedVals**字段的**ClearToken**字段中。

EP1还生产一个Diffie-Hellman算法的公钥部分，把公钥放在**ClearToken**的**dhkey**字段的**halfkey**字段中，把Diffie-Hellman算法的模数和基数放到**ClearToken**的**dhkey**字段的**modsize**和**generator**当中。

注：建议Diffie-Hellman参数使用一个独立**ClearToken**字段，不和认证信息共用一个**ClearToken**字段。

cryptoHashedToken的**tokenOID**设置为“A”，表示对整个SETUP消息计算散列值。如果设置为“B”，则只针对**ClearToken**部分计算散列值。**cryptoHashedToken**的**token**字段的**algorithmOID**设置为“U”，指示散列算法为HMAC-SHA1-96，**params**设置为NULL（即所有可选字段都不存在）。EP1然后使用20字节的密钥KEY1以及HMAC-SHA1-96算法计算整个消息的散列值。

EP1把计算得到的散列值放到SETUP消息的**cryptoH323Token**字段的**CryptoToken**字段的**cryptoHashedToken**字段的**token**字段的**hash**字段，然后发送消息给GK1。

GK1收到SETUP消息后，根据下面的原则验证散列值:

- **timestamp** 是否及时，**random** 是否惟一。
- 检查 **generalID** 是否等于自己的 ID。
- 检查 **sendersID** 是否为 EP1 的 ID。
- 比较重新计算的散列值是否和 ARQ 消息中的散列值一样。

如果验证成功，在GK1转发SETUP消息给GK2前，GK1用适当的值替换掉SETUP消息中的**ClearToken**字段的**timeStamp**、**random**、**sendersID**和**generalID**。**timestamp**字段包含GK1当前的时间戳，**random**字段包含GK1-GK2之间的序列号，**generalID**为GK2的别名，**sendersID**为GK1的别名。GK1保持收到的Diffie-Hellman不变。

GK1然后用Key2重新计算散列值，把散列值放到SETUP消息的相应字段中。

当GK2收到SETUP消息，GK2进行消息的认证/完整性验证，然后类似于GK1的方式，修改必要的字段，重新计算散列值后发送SETUP消息给EP2。

注：Diffie-Hellman参数只在SETUP和CONNECT消息中存在。

D.6.3 H.245 消息的认证和完整性验证

考虑EP1发送一个TerminalCapabilitySet消息给EP2。EP1检查是否消息发给GK1（即是否为路由模式），如果是，那么把H.245消息嵌入到H.225.0呼叫信令消息中。H.225.0呼叫信令消息中有关认证的字段的设置方法，见前节说明。因为使用隧道H.245，呼叫信令消息需要正确设置下面字段：

- **h245Tunnelling** 字段应该设置为 TRUE。
- **h245Control** 字段包含 H.245 消息 ASN.1 编码后的字节流。

EP1在呼叫信令消息中加入一个**CryptoToken**字段，设置**tokenOID**为“A”或“B”，正确设置**timeStamp**，**random**、**sendersID**、**generalID**等字段。设置hashedVals字段中的**ClearToken**的**tokenOID**为“T”，设置**algorithmOID**为“U”，计算散列值并放入消息中相应字段。

如果当前没有正在发送的呼叫信令消息，则产生一个H.225.0 Facility消息，用来承载H.245控制消息。呼叫信令消息的**h323-uu-pdu**保护下面字段：

- **h323-message-body 设置为 facility，包含下面字段：**
- **reason 设置为 undefinedReason。**
- **h245Tunnelling** 字段应该设置为 TRUE。
- **h245Control** 字段包含 H.245 消息 ASN.1 编码后的字节流。
- 按前节说明正确设置认证信息字段。

按上所述，EP1生成一个包含了**CryptoToken**字段的H.225.0 Facility消息，然后把消息发给GK1。

GK1收到EP1的消息，对消息进行验证，如果成功，取出H.225.0呼叫信令消息中的H.245消息内容，重新嵌入到发送给GK2的H.225.0消息中。同样地，如果当前无正在发送的H.225.0呼叫信令消息，可以产生一个H.225.0 Facility消息，用来承载H.245消息。GK1按照前面所述的方法，正确设置呼叫信令消息的各字段后，发送给GK2。GK2重复类似的过程。

D.6.4 直接路由模式

H.323设备可以按GK路由的模式通信，也可以部署为直接路由模式。直接路由模式需要附加的安全措施（如访问令牌），这些措施在GK路由的环境下是不需要的。附录C描述如何保护直接路由的呼叫模型。

D.7 BES 支持

安全的H.323设备可以按照ITU-T H.235.0第I.1.6节中描述的规程使用BES服务。

D.8 H.235 版本 1 和版本 2 的兼容性

因为很少有部署H.235版本 1 的应用环境，本附录要求支持H.235版本2以上，不考虑版本 1 的兼容性。

D.9 组播行为

如果GRQ和LRQ采用组播传送，那么不执行本附录规定的安全过程。如果是采用单播方式发送，则应该使用本附录规定的过程进行安全保护。

D.10 安全的消息列表

D.10.1 RAS 消息

H.225.0 RAS消息	H.235信令字段	认证和完整性验证
所有消息	cryptoTokens	过程1或1A

D.10.2 H.225.0 呼叫信令消息

H.225.0呼叫信令消息	H.235信令字段	认证和完整性验证
Alerting-UUIE， CallProceeding-UUIE， Connect-UUIE， Setup-UUIE， Facility-UUIE， Progress-UUIE， Information-UUIE， ReleaseComplete-UUIE， Status-UUIE， StatusInquiry-UUIE， SetupAcknowledge-UUIE， Notify-UUIE	cryptoTokens	过程1或1A

D.10.3 H.245 消息

H.245消息内嵌到安全的呼叫信令消息中。

D.11 GeneralID 和 SendersID 的使用方法

表D.2 GeneralID和SendersID的使用方法

消　息	sendersID	generalID
Unicast **GRQ**	**EPID**，如果存在，否则**NULL**	**GKID**
Multicast **GRQ**	**EPID**，如果存在，否则**NULL**	
GCF，**GRJ**	**GKID**	**EPID**，如果存在，否则NULL
Initial **RRQ**	**EPID**，如果存在，否则**NULL**	**GKID**
RCF	**GKID**	**EPID**
RRJ	**GKID**	
URQ，UCF，URJ，BRQ，BCF，BRJ，DRQ，DCF，DRJ，NSM，RIP，SCI，SCR，XRS（EP-to-GK）	EPID	GKID
URQ，UCF，URJ，BRQ，BCF，BRJ，DRQ，DCF，DRJ，NSM，RIP，SCI，SCR，XRS（GK-to-EP）	GKID	EPID
ARQ，IRQ，RAI	EPID	GKID
ACF，ARJ，BCF，LCF，LRJ，IRR，IRQ，RAC，LCF，LRJ，IACK，INAK	GKID	EPID
Unicast LRQ（EP-to-GK）	EPID	GKID
Unicast LRQ（GK-to-GK）	GKID	GKID
Multicast LRQ	EPID	
注：GKID代表网守标识符，EPID代表终端标识符。空格表示标识符不存在		

D.12 对象标志符（OID）列表

表D.3 对象标识符

Object identifier reference	Object identifier值	说 明
"A"	{itu-t（0）recommendation（0）h（8）235 version（0）2 1} {itu-t（0）recommendation（0）h（8）235 version（0）1 1}	过程 1 使用，CryptoToken-tokenOID，指示针对整个消息计算散列值
"E"	{itu-t（0）recommendation（0）h（8）235 version（0）3 9} {itu-t（0）recommendation（0）h（8）235 version（0）2 9}	端到端的ClearToken，传递 sendersID用于接收端验证
"T"	{itu-t（0）recommendation（0）h（8）235 version（0）2 5} {itu-t（0）recommendation（0）h（8）235 version（0）1 5}	过程 1 和 1 A使用，指示 ClearToken 用于基线安全认证过程
"U"	{itu-t（0）recommendation（0）h（8）235 version（0）2 6} {itu-t（0）recommendation（0）h（8）235 version（0）1 6}	过程 1 和 1 A使用，指示Algorithm OID为HMAC-SHA1-96

附 录 E
（规范性附录）
采用本地H.235/H.245密钥管理的声音加密安全轮廓

E.1 范围

本附录详细说明了采用本地H.235/H.245密钥管理的声音加密安全轮廓。本附录详细说明了声音加密过程和本地H.245密钥管理过程。

E.2 约定

当媒体加密使用了负载填充时，某些加密算法自身有关于填充方法的规定。不管采用什么样的填充方法，都不会影响互操作性，但可能有不同的安全效果。这是一个实现问题，不在本附录中说明。

E.3 系统概述

E.3.1 声音加密安全轮廓

声音加密轮廓要求和其他安全轮廓一起使用（如基线安全轮廓），单独使用不安全。

H.323实体可以实现本附录来获得声音的保密性。根据不同商业模型和出口需求，本附录提供了4种加密算法供选择：AES，RC2，DES或3DES。除了CBC块加密模式，H.323实体也可以实现EOFB流加密模式。当应用场合不需要声音加密时，Diffie-Hellman密钥管理过程不需要。

对于实现了H.235版本4或更高版本的H.323实体，应该优先提供128位的AES加密算法以获得更好的安全性能和更高的计算性能。这些实体也可以提供168位的3DES算法，便于和H.235版本4之前的版本进行互通。因为56位的DES算法和56位的RC算法已经不再认为是安全的了，除非要求和早期的H.235版本互通，否则不宜使用这类算法。

H.323实体实现了H.235版本4或更高版本，应该优先接受128位的AES算法。如果未提供AES能力，H.323宜接收168位的3DES算法。H.323实体不宜接受56位的DES算法和56位的RC算法，除非安全策略运行接受这类算法且无更高的支持的加密算法。

表E.1 语音加密轮廓

<table>
<tr><td rowspan="2">安全服务</td><td colspan="4">呼叫功能</td></tr>
<tr><td>RAS</td><td>H.225.0</td><td>H.245</td><td>RTP</td></tr>
<tr><td>认证和完整性</td><td></td><td></td><td></td><td></td></tr>
<tr><td>不可否认</td><td></td><td></td><td></td><td></td></tr>
<tr><td>保密</td><td></td><td></td><td></td><td>56位DES | 56位 RC2 | 168位3DES | 128位AES</td></tr>
<tr><td></td><td></td><td></td><td></td><td>CBC模式或EOFB模式</td></tr>
<tr><td>访问控制</td><td></td><td></td><td></td><td></td></tr>
<tr><td>密码管理</td><td></td><td>经认证的Diffie-Hellman 密钥交换</td><td>集成的H.235会话密钥管理（经认证的Diffie-Hellman 密钥交换，密钥更新）</td><td></td></tr>
</table>

Diffie-Hellman交换过程在通信双方之间产生一个共享密钥，这个共享密钥用于加密（一组）会话密钥。会话密钥用于媒体的加密。

声音加密安全轮廓是对基线安全轮廓的增强选项。可以通过安全能力协商来确定双方是否使用声音加密安全轮廓。如果应用环境不需要使用声音加密功能，那么媒体加密和媒体密钥管理不需要实现。

可以选择的加密算法有：AES，RC2，DES，3DES。

注：3DES向下兼容DES算法。

无论采用哪种加密算法，下面的要求应该遵从：

— 如果需要，初始矢量（IV）应该按E.6.3.1节生成。

— 如果需要，按E.6.3.2节规定进行媒体的填充。

声音载荷应该使用加密算法｛“X”，“Y”，“Z3”或“Z”｝按照E.6节描述的过程以及E.6.3.2节描述的填充方式进行加密。声音载荷可以使用操作于流加密模式（EOFB）的协商好的加密算法｛“X”，“Y”，“Z3”或“Z”｝加密。

E.4 H.245 信令过程

一般而言，控制媒体通道加密参数的方法和控制其他编码参数是一样的。每个终端都指示其（加密）能力，数据源端提供一种使用模式，接收方接受或拒绝这种模式。所有和传输无关的参数，例如算法，都在通用逻辑通道元素中提供，而和传输相关的参数，如密钥 / 加密算法同步，则在传输相关的结构中传递。

E.4.1 安全 H.245 通道

假设H.245信道连接建立过程指示操作于一种安全模式，如TLS，那么在传递H.245消息前，首先需要进行（H.245信道的安全机制）握手和认证。如果协商完成，应该用适合H系列终端的某种机制进行所有证书交换。在实现H.245信道的安全后，终端使用H.245协议方式和普通H.245信道完全相同。

E.4.2 不安全的 H.245 信道

另外，H.245信道也可以操作在不安全的模式。两个实体打开一个安全的逻辑通道用来完成认证和密码推导，例如TLS（IETF RFC2246，IETF RFC3546）或IPSec（IETF RFC2401）可用于打开一个类型为h235Control的逻辑通道，该通道可用于推导对媒体密钥加密的主密钥或传送EncryptionSync消息。

E.4.3 安全能力交换

终端ITU-T H.245第5.2节定义的过程以及配套的ITU-T H系列标准，使用H.245消息交换能力集。现在，能力集可包含带安全和加密参数的能力定义。例如，一个终端提供了收/发H.261视频的能力，它也可以增加一个收/发加密的H.261视频的能力。每一种加密算法结合一种媒体编解码能力产生一种新的编解码能力。在能力交换时，终端可同时提供无加密的编解码能力和对应的带加密的编解码能力。这样允许终端根据当前的负荷和有效资源决定是否加密媒体。

在能力协商完成后，终端可以按不安全模式一样的方式打开一个安全媒体通道。

E.4.4 主角色

为了打开双向逻辑通道和其他冲突调解，H.245主从决定过程用来建立主实体（角色）。主角色也应用于安全方法。虽然媒体流安全模式是由源端设置的，但是主（角色）终端产生加密密钥。无论主角色终端是媒体流接收者还是发送者，媒体流加密密钥都由主角色终端产生。为了支持组播通道的加密操作，MC（也是主角色）应该产生共享密钥。

E.4.5 逻辑通道信令

终端打开安全的媒体通道的方式和打开不安全的媒体通道是相同的。每个媒体通道的安全模式（包括无加密）独立于其他媒体通道。安全模式定义在OpenLogicalChannel消息的dataType字段。主角色终端在OpenLogicalChannel消息（发送通道）或OpenLogicalChannelAck消息（接收通道）中把初始密钥传递给从角色。

OpenLogicalChannelAck应该视为同意加密模式。如果接收端不认可OpenLogicalChannel中的解密模式，则应该返回一个错误原因为dataTypeNotSupported或dataTypeNotAvailable（短暂条件）的OpenLogicalChannelReject消息。

在打开逻辑通道的协议交换过程中，加密密钥由主角色终端传递给从角色终端。对于从角色终端打开的媒体通道，主角色终端在OpenLogicalChannelAck消息中返回初始密钥以及加密同步点（encryptionSync字段）。对于主角色终端打开的通道，初始密钥和加密同步点（encryptionSync字段）在OpenLogicalChannel消息中传递给对方。

E.4.6 快速连接安全

终端可以使用快速连接（Fast connect）过程（参见ITU-T H.323 第8.1.7和第8.1.7.1节）安全地交换密钥材料（主密钥和会话密钥）。E.4.6.1节描述了不使用多个加密算法的简单Fast Connect过程。E.4.6.1.1节描述了使用多个加密算法的情况。

E.4.6.1 单向通道快速连接安全

本过程描述了如何建立一条从主叫到被叫端方向的安全的单向媒体通道。

主叫过程：主叫方提供DH令牌（token）和fastStart结构。DH令牌包含在CryptoToken的ClearToken字段中，或者是一个单独的ClearToken中，见E.4.8节说明。在SETUP-CONNECT交互过程中，完成Diffie-Hellman（DH）共享密钥交换过程。CryptoToken字段的ClearToken子字段应该包含一个dhkey，用于传递本附录中说明的参数。halfkey字段包含各方随机公钥，modsize包含DH素数和原根。使用到的DH参数见表E.4。更多细节，请参考IETF RFC2412的附录E2。

注：因为H.225.0消息是经过认证的，所以DH交换过程也是被认证的。

在呼叫的两个方向上，分别有一个H.225.0呼叫信令消息包含DH公钥和其他DH参数。如果终端标识符在GK上成功注册，主叫端和被叫端还应在上述呼叫信令消息中包含一个新的端到端的ClearToken，该ClearToken的sendersID字段设置为终端标识符，tokenOID字段设置为“E”，任何中途的H.323不应该修改这个ClearToken。

FastStart结构包含建议的（offered）打开逻辑通道命令参数。建议同时提供带H.235安全能力的和不带H.235安全能力的逻辑通道。在H.245能力交换过程中，终端为每个支持的媒体能力提供H235SecurityCapability选项。每个媒体能力都和一个H.235安全能力关联一起（产生一个新的媒体能力）。根据表E.6，建议这些能力指示支持128位AES-CBC（OID=“Z3”），56位的RC2-CBC（OID=“X”），56位的DES-CBC（OID=“Y”），还可以指示支持168位的3DES（OID="Z"），或168位的3DES-EOFB（OID=“Z1”），RC2-EOFB（OID=“X1”），DES-EOFB（OID=“Y1”），AES-EOFB（OID=“Z2”），参见表E.6。

打开逻辑通道（OLC）命令同时携带forwardLogicalChannelParameters和reverseLogicalChannelParameters，其中dataType选择h235Media，并且h235Media的

encryptionAuthenticationAndIntegrity字段选择encryptionCapability，其中encryptionCapability只含一个MediaEncryptionAlgorithm。

针对主从关系的需求，被叫方事先被指定为主角色，参见E.4.4节。主叫方应该设置mediaWaitForConnect为true，表示需要获得会话密钥后，方可发送媒体。在希望使用“早期媒体”（“early media”）的某些场合，被叫方在发送响应消息和加密密钥材料的同时也发送媒体数据。建议主叫方在密钥未知的情况下不进行媒体解码。

注：在该情况下，如果被叫发送加密的媒体给主叫方，主叫方将无法解密，除非在Connect消息中获得共享密钥。

被叫过程：在FastStart过程中，被叫提供它的DH令牌（token，参见E.4.8节）和接受的fastStart结构。假如正在使用Diffie-Hellman过程，建议被叫方尽快地在响应消息中返回它的DH令牌，例如在PROCEEDING或ALERTING消息中。这允许主叫方尽早地计算出主密钥，并准备接收媒体密钥以及加密媒体流。

注：如果双方没有共同支持的加密算法，媒体流可以不加密或连接终止，该行为取决于应用安全策略。

每个实体应该从DH共享密码中取适量的最低比特位，组成主密钥。也就是说，OID“X”、OID“X1”、OID“Y1”和OID“Y2”加密算法取56个比特，OID“Z”、OID“Z1”和OID“Z2”取168个比特，OID“Z3”和OID“Z2”取128个比特，具体参见表E.6。

被叫方发送OpenLogicalChannel（Ack）消息（嵌在fastStart字段中），其中encryptionSync字段包含会话密钥。encryptionSync包含主叫方到被叫方通道的加密密钥。使用KeySyncMaterial或V3KeySyncMaterial之一，密钥传递应按E.7.3节的说明继续进行。会话密钥应该按下面描述的方法用主密钥加密。

注：会话密钥的产生办法不做说明，产生方法是一个实现问题，注意避免产生安全弱的密钥。

使用E.5.3节描述的过程，加密的会话密钥在encryptionSync字段传递。会话密钥存放在KeySyncMaterial的keyMaterial字段。如果会话密钥长度不等于一个加密块的长度，应该在加密前填充为一个加密块的长度，填充方法由加密算法决定。（填充后的）KeySyncMaterial应该用下面的密钥加密：

1）OID“X”、OID“X1”、OID“”Y1”和OID“Y2”算法取DH共享秘密的最低56个比特；

2）OID“Z”、OID“Z1”、OID“Z2”取DH共享秘密的最低168个比特。

另外，如果双方都支持H.235 V3版本的安全过程，优先考虑使用E.5.3.1节描述的改进的密钥传递（参见E.5.2节）。

当用快速启动打开一个双工安全的媒体通道时（由两个单工通道合成），被叫端应该朝主叫的方向打开另一个（第二个）逻辑通道。这个逻辑通道应该在另一个fastStart元素中指示。使用DH共享密钥为主密钥，被叫方在encryptionSync中为第二个通道分配一个不同的会话密钥。

E.4.6.1.1 快速连接中使用多个加密算法

加密算法在OLC消息的dataType.h235Media.encryptionAuthentication-AndIntegrity.encryptionCapability中描述。本过程允许OLC中包含多个按优先排序的加密算法。接收方从提议的加密算法中选择其中之一，然后返回只包含被选加密能力的OLC。

为了提供最大效率，对象ID“NULL-ENCR”（见表E.2）表示“无”加密算法，“无”加密算法表示无加密操作发生。使用这个特殊方法，每个方向的每个媒体类型对应一个OLC。

表E.2 “无”加密算法的对象标识符

对象标志符引用	对象标识符	描 述
“NULL-ENCR”	{itu-t（0）recommendation（0）h（8）235 version（0）3 26}	指示空（NULL）加密算法

主叫方过程（参见ITU-T H.323 第8.1.7.1节）：如果dataType选择h235Media，那么包含的encryptionAuthenticationAndIntegrity的encryptionCapability字段可包含多个加密算法（包括“空”算法）。这个结构提供了在多种加密算法中选择其一用于媒体加密的能力。

被叫方过程（参见ITU-T H.323 第8.1.7.1节）：如果为一个通道提供了多种算法，被叫方必须选择其中之一，并删除OpenLogicalChannel的其他加密算法。

E.4.7 加密的 H.245 DTMF

终端可以选择发送加密的DTMF[IETF RFC2833]信号来达到保密性要求。使用会话密钥，终端可以加密UserInputIndication中的DTMF信号，如下。

1）加密的基本串：encryptedAlphanumeric。

2）加密的iA5串：signal内的encryptedSignalType。

3）加密的通用串：extendedAlphanumeric内的encryptedAlphanumeric。

注：iA5串内的其他RTP参数和时戳、逻辑通道号不加密，拨号音（tone）周期更新不加密，因为这些字段不包含敏感信息。

协商能力secureDTMF和加密的iA5字符串有关联。

建议使用E.5.5节描述的密钥管理过程产生会话密钥。这个会话密钥应该用于加密H.245 DTMF信号。

注：这并不暗示该会话密钥用于RTP载荷的加密。

然而，当设置了rtpPayloadIndication标志，同时启用由UserInputIndication和RTP传递DTMF信号时，强烈建议RTP载荷使用声音安全轮廓进行加密。

表E.3提供了可用加密算法，建议工作在EOFB操作模式（OFB是特例，参见E.5.4节）。为了避免潜在填充，不建议使用需要填充的CBC、CFB或其他块加密模式。

表E.3 H.245 DTMF信令的加密模式

对象标志符引用	对象标识符	描 述
“DES-EOFB-DTMF”	{itu-t（0）recommendation（0）h（8）235 version（0）3 12}	DES-56 EOFB模式
“3DES-EOFB-DTMF”	{itu-t（0）recommendation（0）h（8）235 version（0）3 13}	3DES-168 EOFB模式
“AES-EOFB-DTMF”	{itu-t（0）recommendation（0）h（8）235 version（0）3 14}	AES-128 EOFB模式

E.4.7.1 加密的 Basic string

如果UserInputCapability字段选择encryptedBasicString，那么UserInputIndication消息选择encryptedAlphanumeric字段，其中algorithmOID应该指示所使用的加密算法，paramS包含初始矢量。加密的文字和数字放在encrypted字段。

E.4.7.2 加密的 iA5 string

如果UserInputCapability选择encryptedIA5String，那么SUserInputIndication消息选择signal字段，其中encryptedSignalType字段包含加密后的SignalType。未加密的signalType字段应该包含一个无用的“!”字符。

algorithmOID指示加密算法，paramS包含初始矢量。

E.4.7.3 加密的 General string

如果UserInputCapability选择encryptedGeneralString，那么UserInputIndication消息选择extendedAlphanumeric字段，其中包含的encryptedAlphanumeric字段中的algorithmOID指示使用的加密算法，paramS保护加密初始矢量，encrypted存放加密后的内容。未加密的alphanumeric应该包含一个空串。

E.4.8 Diffie-Hellman 运算

本附录支持Diffie-Hellman共享密钥交换协议。取决于使用环境，协议的DH密钥可以作为一个主密钥或作为媒体密钥。

Diffie-Hellman可用参数g，p表达，其中p是一个大素数，g是一个模p乘法群的本原根。g^x模p定义为主叫方公钥，g^y模p定义为被叫方公钥。IETF RFC2412提供了Diffie-Hellman算法的更多背景信息，以及如何选取p，g参数。

通常，针对一种应用，p和g等于已明确定义的值，但应用也可以选择私有值。被叫方应该意识到，私有的值可能产生弱的安全性，例如主叫方提供的p可能不是素数，g只能生成一个小的乘法群等。由于大范围参数测试不实际，是否接受或拒绝这样的参数由被叫方安全策略决定。

对于已定义（标准化的）的DH参数，使用一个OID来代表这样一组参数可以生成更紧凑的编码消息。一个ClearToken可以用来引用一个已定义的参数实例，其中tokenID设置为DH OID。发送方可以在ClearToken的dhkey字段中同时给出引用的DH参数，但这不是必须的。

当DH参数实例通过上述的OID表达时，CryptoToken（用于附录C）中的ClearToken的DH参数应该为空（或不存在），所有的DH参数实例应该在分开的（一到多个）ClearToken中传递，其中tokenOID保存DH OID，dhkey可选，ClearToken其他字段不使用。

注：不排除在CryptoToken字段中直接包含DH参数值的用法可能性。

如果使用了一个私有DH参数实例，那么DH-OID应该设置为“DHdummy”，私有DH参数值必须明确给出（dhkey字段）。

主叫方可以提供多个ClearToken，每个表示一个不同的DH参数实例。鼓励主叫方提供尽量多的DH参数实例，增加被叫方找到一组共同支持的DH参数的成功概率。

被叫方应该从接受主叫方提供的DH参数实例中选取一个实例。如果被叫方能够接受其中的一个实例，那么被叫方在响应消息返回这组DH参数值，否则被叫方响应一个自己支持的DH参数实例。加密算法的强度和DH参数实例是相关的，参见表E.4。

如果由于安全原因或者是处理能力不足的原因，被叫方拒绝主叫方的提议，那么被叫方应该保留响应消息中的dhkey为空（不存在）。

被叫方应该在CONNECT响应消息中包含DH参数。被叫方可以在中间消息（PROCEEDING，ALERTING）中包含DH参数，但是CONNECT消息仍然需要包含DH参数。

注：被叫方在什么时候包含DH参数，需考虑几个方面：响应事件，被叫方的处理负荷，先期媒体能力（英：early media）等。这些方面是和实现相关的。

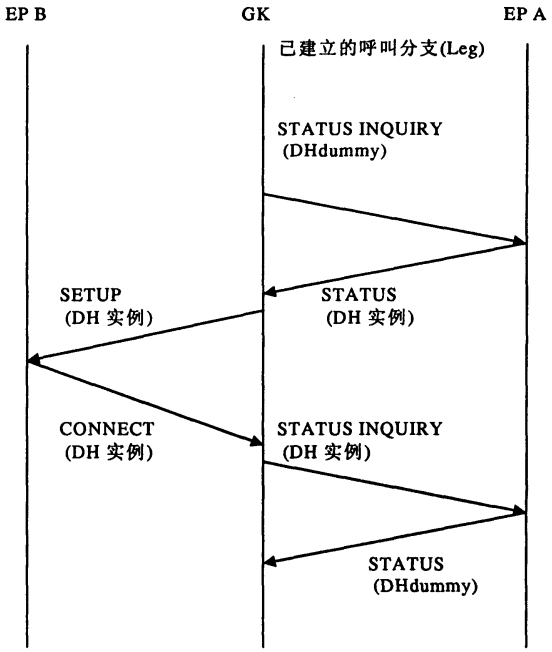
然而由于某些原因，某些路由GK不能给主叫方转送所有的SETUP到CONNECT之间的响应。这样，有可能带有DH参数的一个或多个响应消息不能到达主叫方。为了防止这种情况发生，建议被叫方在SETUP-CONNECT之间的每个响应都带相同的DH参数。

如果ClearToken中的DH OID和dhkey同时存在，但是DH OID引用的标准化参数和dhkey中给出的具体值不一样，那么dhkey的值优先被使用。建议被叫方在响应消息中修正错误的DH OID。

E.4.8.1 呼叫中请求重新 Diffie-Hellman 协商

在呼叫中，H.323网守可以使用本节过程请求重新进行DH参数的协商。为了在一个已经连接到网守的终端和准备连接到网守的终端之间协商DH密钥，这样的重新协商过程可能是必要的（参见图E.1）。几

个补充业务需要DH参数重新协议过程。本节定义的过程只能在H.323终端处于“发送侧暂停”状态时才能执行，具体参见ITU-T H.323 第8.4.6节。

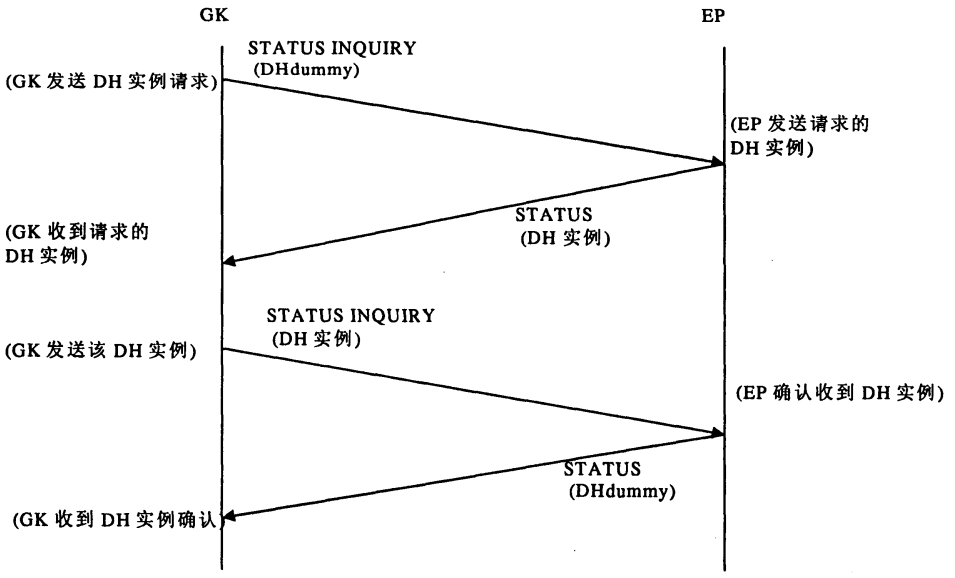


图E.1 补充业务的DH参数重协商用法

为了在呼叫中间请求DH参数，H.323实体应该发送一个包含ClearToken字段的STATUS INQUIRY消息，其中tokenOID设置为DH OID“DHdummy”，其他字段省略。

如果一个H.323实体收到一个包含ClearToken字段的STATUS INQUIRY消息，其中tokenOID设置为DH OID“DHdummy”，终端应该用包含一组DH参数实例的STATUS消息进行响应，见图E.2。DH实例构造应该遵守E.4.8节描述的针对SETUP消息的规则。

注：为了响应STATUS INQUIRY消息，不支持本过程的H.323实体会响应一个无DH参数实例的STATUS消息。



图E.2 呼叫中请求DH参数

为了在呼叫中传递一个被采纳的DH实例，H323实体发送一个携带被采纳的DH实例，见图E.2。被采纳的DH实例构造方法应该遵守E.4.8节描述的针对SETUP响应消息的规则。

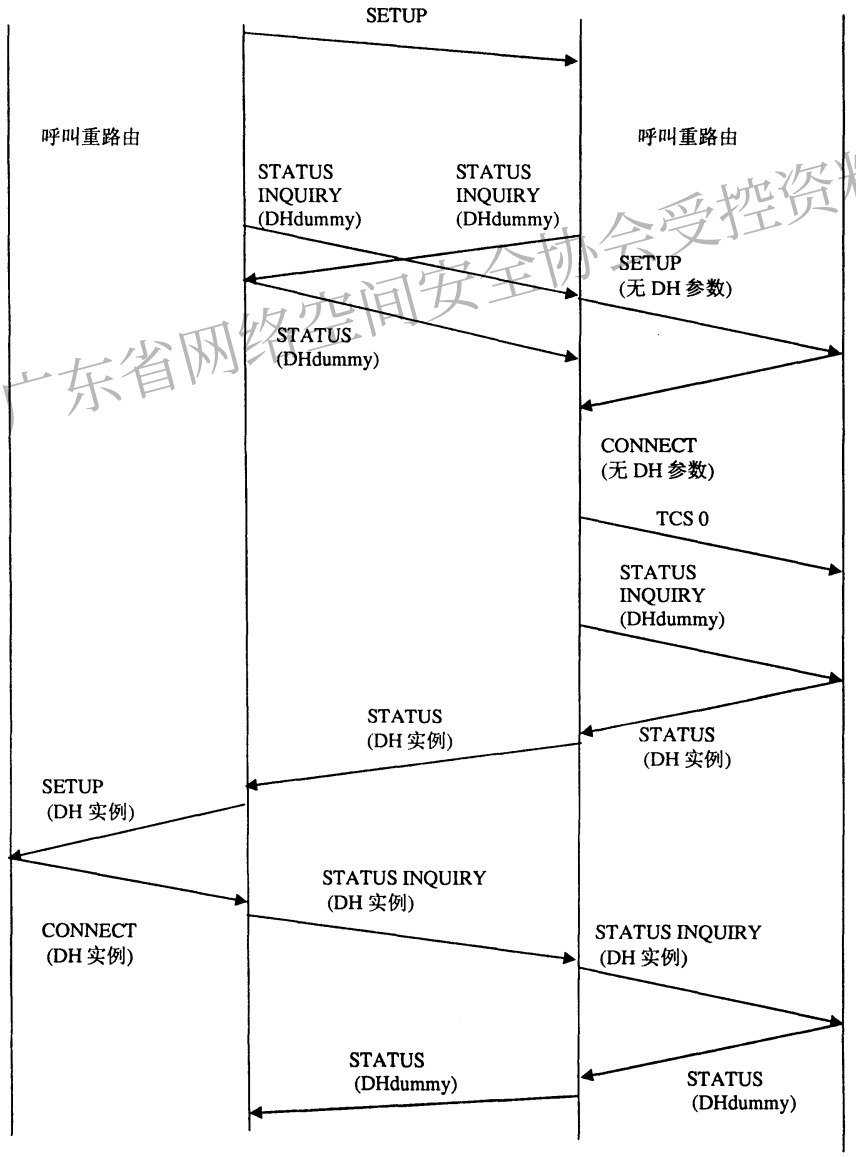
注：为了响应STATUS INQUIRY消息，不支持本过程的H.323实体会响应一个无DH参数实例的STATUS消息。

终端收到STATUS INQUIRY消息后，应该根据收到的DH参数实例和发出的DH参数实例重新计算DH共享秘密。

除了下面将列举的例外情况外，如果GK收到含有DH参数实例的STATUS INQUIRY消息，它应该把这个消息转发给呼叫的另一个分支。

如果一个GK收到一个STATUS INQUIRY的响应消息STATUS，它把这个STATUS消息转发给收到STATUS INQUIRY消息的呼叫分支。

当一个GK在发出一个携带DH-OID“DHdummy”的STATUS INQUIRY消息之后，处于等待接收响应消息的状态，如果这时收到一个携带DH-OID“DHdummy”的且呼叫参考值（CRV）标志等于1的STATUS INQUIRY消息，GK应该响应一个携带DH-OID“DHdummy”的STATUS消息（参见图E.3）。



图E.3　两个GK同时重路由的情况

如果在第二个呼叫分支还未建立时，GK收到带DH-OID“DHdummy”的STATUS INQUIRY消息，GK应该等待第二个呼叫分支建立后，给第二个呼叫分支发送一个空的能力集，然后再转发收到的STATUS INQUIRY消息（见图E.3）。

在它已经发出一个含DH参数实例的STATUS消息后和收到一个含DH参数实例的STATUS INQUIRY之前，GK不应该启动本节定义的过程。

E.5 信令和过程

H.323实体应服从H.323第8节概括的过程（呼叫信令过程）。H.323应该有能力在H.225.0呼叫信令消息中协商H.245安全需求。

假如H.225.0信道自身需要安全保护，应该服从H.323第8章所规定的过程。不同之处是，只有建立了安全信道后，通信才能进行。安全信道建立过程参见第7.2节。

H.225.0交换的目的是提供建立安全的H.245通道的机制。作为可选项，在H.225.0消息交换时，还可进行身份认证。身份认证可使用数字证书、密码、加密和/或散列。

一个H.323终端收到一个携带h245SecurityCapability的SETUP消息，应该在CONNECT消息中响应可采纳的h245SecurityMode。如果双方没有共同的（h245安全）能力，被叫方可以发送一个Release Complete消息来拒绝连接，拒绝原因为SecurityDenied。错误原因“不会传递关于安全能力不匹配的信息”，所以主叫方需要通过其他手段了解失败原因。如果主叫端收到一个不可采纳的安全模式的CONNECT消息，它可以发送拒绝原因为SecurityDenied的Release Complete消息。如果主叫方收到一个无安全能力的CONNECT消息，它可以发送拒绝原因为undefinedReason的Release Complete消息。

如果主叫终端收到一个可采纳的h245Security模式，它应该打开一个操作在该安全模式下的H.245通道。若建立该安全模式下的H.245信道失败，应该视为协议错误，连接终止。

E.5.1 ITU-T H.235 版本 1 兼容性

一个有安全能力的终端不应该向无安全能力的终端返回任何安全相关的字段、指示和状态。如果被叫方收到包含H245Security能力和/或认证令牌的SETUP消息，它可以返回ReleaseComplete消息来拒绝连接。但它应该使用UndefinedReason错误原因。在发送一个包含了H245Security能力和/或认证令牌的SETUP消息时，如果主叫端收到一个不包含H245SecurityMode和/或认证令牌的CONNECT消息，它可以发送拒绝原因为undefinedReason的Release Complete消息来中止连接。

E.5.2 ITU-T H.235 版本 3 特性

针对媒体通道，本附录兼容 ITU-T H.235 版本 3 或更高版本的改进的安全过程，这些过程在ITU-TH.235 版本 1 和版本 2 中不支持。改进的过程如下：

- 改进的密钥传递（V3KeySyncMaterial，见 E.5.3.1 节）。
- 改进的密钥更新，参见 E.5.6.2 节。

在呼叫建立阶段，增加了 ITU-T H.235 版本指示，用于彼此了解对方是否支持 ITU-T H.235 版本 3 或更高版本。

实现了 ITU-T H.235 版本 3 或更高版本的设备，宜使用本节描述的过程来决定 ITU-T H.235 版本 3 能力（改进的密钥传递和更新）。取决于信令过程的结果，为了保持和 ITU-T H.235 版本 1 和版本 2 的兼容性，终端可以使用 E.5.3 节的过程。

为了指示是否支持 ITU-T H.235 版本 3，在呼叫信令阶段，终端应该包含一个指示 ITU-T H.235 版本

3 能力的 ClearToken 字段。缺少这样的 ClearToken 可理解为只支持 ITU-T H.235 版本 1 和版本 2。在该情况下，终端应该使用 E.5.3 节的过程。否则，终端可以使用 E.5.3.1 节描述的改进的过程或 E.5.3 节的过程。

ClearToken 的 tokenOID 设置为“V3”，其值定义如下。其他字段不用，除非用于传递 DH 参数。

“V3”	{itu-t（0）recommendation（0）h（8）235 version（0）3 24}	版本3能力指示，用于呼叫信令的ClearToken中

E.5.3 密钥传递

主角色方应该产生密钥并分配给从角色方。有两个过程用来传递密钥：

- 主要针对ITU-T H.235版本1和版本2的终端，在本节描述。
- 针对ITU-T H.235版本3或更高版本终端，在E.5.3.1节描述。

ITU-T H.235版本1和版本2的终端使用下面的过程传递密钥:

KeySyncMaterial的generalID保存了主角色方的终端标识符，keyMaterial携带会话密钥。为了提供最起码的密钥来源的认证，建议设置generalID的值（参见E.5.6节）。接收方应该验证generalID的正确性。

注：本附录假设每个终端都和网守注册并获得终端标识符，该标志符可以用generalID来传递。本附录不支持没有网守的情况。

应该使用主密钥加密KeySyncMaterial。加密前，需要把KeySyncMaterial填充为加密块大小的整数倍，其中，最后一个字节设置为填充的字节数（包括最后字节）。填充字节的值由加密算法决定。加密结果保存到H235Key的sharedSecret字段。

E.5.3.1 ITU-T H.235 版本 3 的改进的密钥传递

在ITU-T H.235版本1和版本2中，KeySyncMaterial的ASN.1文法定义以及ENCRYPTED{}操作暴露了大量的文本信息：首先，主角色方的generalID，以及该结构的一些公开的编码比特位。即使generalID被加密，还是可以从消息的其他未加密的部分获知（如senderID）。存在这样一些公开的文本信息大大降低了安全性，导致攻击者可以更加容易地通过“穷举法”方式破解会话密钥，特别是加密算法的加密块很小的时候，如DES-56和RC2算法。

ITU-T H.235版本3能够传递附加的密钥材料：传递一个掺杂（salt）密钥给对方．引入这个掺杂密钥用于EOFB模式，见E.5.4节。

ITU-T H.235版本3扩展了H235Key的secureSharedSecret字段，secureSharedSecret的V3KeySyncMaterial包含下面的信息：如果存在，generalID保存发送者的终端标识符，否则不使用。

algorithmOID指示使用的加密算法和操作模式。

paramsS保存用于会话密钥加密的初始矢量。

注：paramS初始矢量和加密RTP包的初始矢量是两个不同的概念。作为可选项，ClearSalt可包含一个用于会话密钥加密的未加密的掺杂密钥。

encryptedSessionKey保存密钥的密文。

encryptedSaltingKey保存加密的掺杂密钥。该密钥用于EOFB加密算法。

clearSaltingKey可以保存未加密的掺杂密钥。encryptedSaltingKey和clearSaltingKey不应该同时使用。

paramSsalt保存用于加密掺杂密钥的初始矢量。作为可选项，ClearSalt可包含一个未加密的掺杂密钥，用于加密会话密钥掺杂密钥。

注：generalID、algorithmOID和paramS总是按明文方式传递的，相反，encryptedSessionKey和encryptedSaltingKey则保存密钥材料的密文。

主角色根据协商的终端能力，通过V3KeySyncMaterial字段发送密钥给对方。这样，中途的网守不应该改变V3KeySyncMaterial内容。

ITU-T H.235版本3或更高版本建议使用H235Key的secureSharedSecret，但是取决于E.5.2节描述的信令过程的输出结果，可以使用sharedSecret与ITU-T H.235版本1和版本2的终端互通（向后兼容）。

E.5.4 EOFB 模式

OFB模式[ISO/IEC 10116]定义了一种块加密算法运用流加密的操作模式。EOFB模式提供：

- 减少处理时延；
- 发更简单的处理不完整块的方法；
- 好的抗比特错误能力。

EOFB模式对OFB做下列修改：

1）使用一个额外的掺杂密钥；

2）引入一个隐式的包索引。

使用一个额外的掺杂密钥和输出结果XOR操作，可抵抗已知的明文分析破译。这是相比其他操作模式（CBC和OFB）的主要好处。这样，使用EOFB提供了抗高度重复的明文和已知明文的破译能力。

EOFB定义为$C_i = P_i \oplus S_i$以及$S_i = E_{KE}$（KS $\oplus$ S_{i-1}）i=1...n且S_0 = IV其中C_i是第i个密文块，P_i是第i个明文块，S_i是第i个密码块，KE是加密密钥，⊕为比特异或，EOFB参见图F.6说明。

EOFB可以运行在标准的OFB模式，所以EOFB可和OFB后向兼容。在某些希望和OFB模式兼容的场合，掺杂密钥要么设置为全0或encryptedSaltingKey为空。然而，当加密块的尺寸较小时（如DES和RC2），高度推荐使用掺杂密钥。

在最多2^{48}个包处理完后，应该使用一个新的会话密钥和掺杂密钥，否则密钥重复使用可能发生，从而降低安全性。第E.1章定义了DES-56-EOFB，RC2-EOFB，3-DES-EOFB和AES-EOFB。

E.5.5 密钥管理

服从本附录的终端宜使用E.4.6.1规定的Fast Connect过程。如果未使用Fast Connect过程，那么应该使用H.245隧道来保护H.245控制消息的安全。Fast Connect过程允许建立单向的或双向的（两个单向的）逻辑通道。Fast Connect过程负责安全能力的协商，分发主密钥（DH共享秘密）以及分发会话密钥。表E.4提供了各种加密算法的OID以及相关联的Diffie-Hellman组OID。本附录提供了三个D-H组。

1）“DHdummy”：用来表示一个可出口的512位的DH组或是非标准的DH组。

注：未定义标准的DH组，该OID引用任何非标准的DH组。

2）使用一个512比特的DH组来生成RC2（“X”）和DES-56（“Y”）加密算法的主密钥。

3）“DH1024”：这个DH组用于要求高安全性的场合。该OID引用标准化的、固定的DH组。该DH组用于生成3DES（“Z”）加密算法的主密钥。

4）“DH1536”：该DH组是ITU-T H.235版本3的可选项，适用于非常高的安全要求的场合（比1024位更高），该组可用于生成3DES（“Z”，“Z1”）和AES-128（“Z2”，“Z3”）加密算法的主密钥。

推荐使用1024或1536位的DH组，除非处于特殊的安全考虑要求。推荐使用定义好的OID来引用DH组（参见第E.4.8节），除非实现者打算得到没有OID指示的具体的DH参数。在该情况下，实现者宜根据表E.4确定正确的DH组。

终端可以使用非标准的DH组参数。宜使用OID“Dhdummy”来指示这样的非标准DH组。被叫方决定是否接受这样的DH组。

注：协商DH组不意味不再需要协商媒体加密算法。协议媒体加密算法由H.245能力协商过程完成。

注：在连接建立阶段（SETUP到CONNECT）使用的加密算法OID不能用来反推DH参数实例。

表E.4 Diffie-Hellman组

加密算法OID	DH-OID	D-H 组描述
“X”，“X1”(RC2)，“Y”，“Y1”（DES）	“Dhdummy”	模P，任何合适的512比特素数
“Z”，“Z1”（triple-DES），“Z2”，“Z3”（AES）	“DH1024”	模P，1024比特素数 $P = 2^{1024} - 2^{960} - 1 + 2^{64} \times \{[^{894} + 129093\}$ = （179769313486231590770839156793787453197860296048756011706444 423684197180216158519368947833795864925541502180565485980503 646440548199239100050792877003355816639229553136239076508735 759914822574862575007425302077447712589550957937778424442426 617334727629299387668709205606050270810842907692932019128194 467627007）$_{10}$ G = 2
“Z”，“Z1”（triple-DES），“Z2”，“Z3”（AES）	“DH1536”	模P，1536比特素数 $P = 2^{1536} - 2^{1472} - 1 + 2^{64} \times \{[2^{1406} + 741804\}$ = （241031242692103258855207602219756607485695054850245994265411694 195810883168261222889009385826134161467322714147790401219 650364895705058263194273070680500922306273474534107340669624 601458936165977404102716924945320037872943417032584377865919 814376319377685986952408894019557734611984354530154704374720 774996976375008430892633929555996888245787241299381012913029 459299994792636526405928464720973038494721168143446471443848 852094012745984428885933652689632091963391 9）$_{10}$ G = 2

E.5.6 密钥更新和同步

对于64比特的块加密，密钥更新速率保证加密不超过2^{32}个加密块。建议实现者在加密2^{30}个加密块后更新密钥（参见E.6.1节）。对于128比特的加密块，密钥更新速率保证一个密钥加密不超过2^{64}个加密块。建议实现者在加密2^{62}个加密块后更新密钥（参见E.6.1节）。通信双方都可以根据自己的安全策略需要，按某个频度改变会话密钥。例如，主角色可以使用encryptionUpdate或encryptionUpdateCommand分发一个新的会话密钥。另一方面，从角色可以使用encryptionUpdateRequest向主角色请求一个新的会话密钥。

MiscellaneousCommand包含encryptionUpdate和encryptionUpdateCommand，其中encryptionSynch设置为下面的参数。

1）synchFlag：新的动态RTP载荷号指示密钥改变。

2）H235Key：包含新的会话密钥。octet串包含ASN.1编码的H.235 H235Key。

H235Key结构的sharedSecret字段使用下列字段。

1）algorithmOID：56比特的RC2设置为“X”，“X1”，56比特的DES设置为“Y”、“Y1”，168比特的3DES设置为“Z”、“Z1”，128比特的AES设置为"Z3"。

注：会话密钥的加密算法和媒体加密算法相同。

2）paramS：设置为初始矢量。对于64比特的流加密算法，iv8保存一个64比特的初始矢量。对于128比特的流加密算法，iv16保存一个128比特的初始矢量。CBC模式不使用这个字段，应该设置为空（NULL），也就是说，CBC算法的初始适量应该为空。只有EOFB模式使用这个字段。

3）encryptedData：设置为KeySyncMaterial的加密结果。

KeySyncMaterial包含下列字段。

1）generalID：密钥分发者的终端标识符。

注：本附录假设每个终端都和网守注册并获得终端标识符，该标志符可以用generalID来传递。

2）keyMaterial：设置为新的会话密钥。对于DES和RC2算法，这是一个56比特的密钥；对于3DES算法，这是一个168比特的密钥；对于AES算法，这是一个128比特的密钥。主角色应该生成一个至少满足下列安全原则的新密钥。它不是半弱的（semi-weak）DES密钥和充分的随机数源。

MiscellaneousCommand包含encryptionUpdateRequest，encryptionUpdateRequest包含keyProtectionMethod，其中sharedSecret设置为TRUE。

注：由于密钥更新和同步依靠H.245消息，所以需要采用H.245隧道技术。

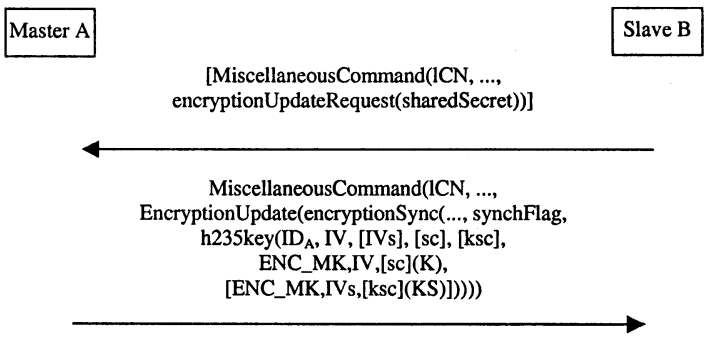
媒体会话密钥并不永久存活。在某个时刻，每个会话密钥会过期。建议生成一个新的密钥用来保护进行的会话。在会议环境中，当一个成员加入或退出安全的会议时，建议产生和分发一个新的组合话密钥，这样可以阻止它们存取过去或将来的数据。

3）基于载荷类型的密钥更新和同步为新的会话密钥定义了新的动态载荷类型，参见E.5.6.1、E.5.6.2和E.5.6.3节。

针对密钥更新，本附录提供了兼容ITU-T H.235版本1和版本2的无确认握手方式和兼容ITU-T H.235版本3或更高版本的更可靠的带确认的握手方式。

E.5.6.1 无确认密钥更新

图E.4显示了无确认密钥分发和更新的握手方法。如果从角色希望更新一个会话密钥，从角色给主角色（Master）发送一个encryptionUpdateRequest。主角色应该通过encryptionUpdate消息给从角色发送一个新会话密钥（主动地或响应从角色的encryptionUpdateRequest）。



图E.4 无确认的密钥分发和更新

其中：lCN是逻辑通道号；synchFlag是新的动态RTP载荷号；IDA是源端的generalID；IV是加密会话密钥的初始矢量；IVs 是加密掺杂矢量的初始矢量；ENC_M，IV，sc（K） 表示用密钥M加密明文K，

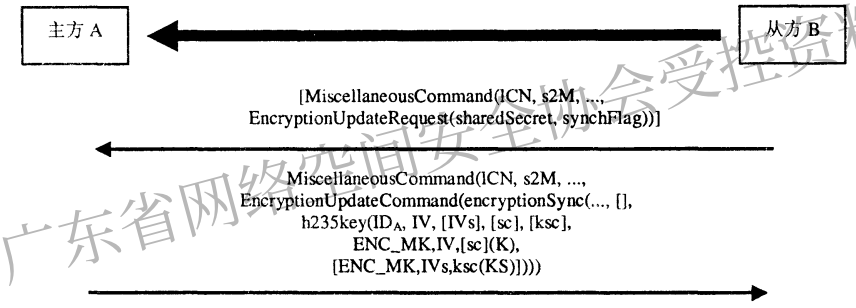
初始矢量是IV以及掺杂密钥sc，sc用于EOFB模式；KS是媒体加密的掺杂矢量（只用于EOFB模式）；K是会话密钥的明文；sc是使用EOFB模式加密会话密钥时的未加密的掺杂矢量；ksc是使用EOFB模式加密掺杂密钥时的未加密的掺杂矢量；s2M/m2S是方向标志（ITU-T H.235版本3或更高版本）（s2m =从到主，m2s =主到从）；[]表示可选项。

下一节描述的密钥更新方法可以使用EOFB加密模式来保护传输的密钥材料。同保护媒体载荷一样的方式，为了使用EOFB模式保护密钥材料，使用了一个额外的掺杂密钥（sc或ksc）。

E.5.6.2 改进的密钥更新

ITU-T H.235版本3或更高版本的终端应该执行有确认机制的密钥更新过程。这将在ITU-T H.235先前版本的基础上，提供可靠的密钥更新方法。应该按照E.5.2节的说明，使用ITU-T H.235版本3特征指示来协商是否具有本过程的能力。

图E.5显示了针对从角色拥有的媒体通道的密钥更新过程。在该例子中，从角色发起密钥更新并向主角色请求一个新会话密钥。从角色应该发送一个MiscellaneousCommand给主角色，其中logicalChannelNumber包含逻辑通道号，sharedSecret应设置为TRUE，direction设置为slaveToMaster以及EncryptionUpdateRequest的synchFlag设置为新的动态RTP载荷号，sharedSecret应设置为TRUE。如果是主角色启动密钥更新，则不用发送EncryptionUpdateRequest消息。



图E.5 请求更新从角色的媒体通道的密钥

主角色响应从角色请求或是代表自己，应该发送一个EncryptionUpdateCommand消息，其中MiscellaneousCommand的logicalChannelNumber设置为逻辑通道号，direction设置为slaveToMaster，encryptionSync内的synchFlag设置为新的动态RTP载荷号。

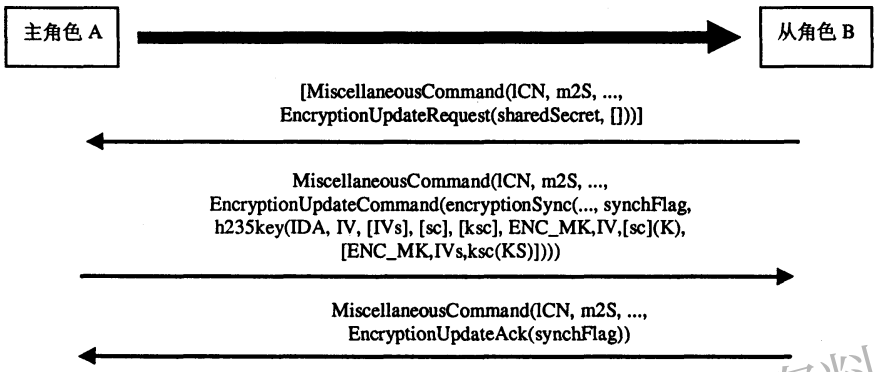
h235key应携带新的会话密钥。h235key的generalID字段应该包含主角色的终端标识符，paramS字段中应该包含使用的初始矢量。加密的会话密钥应该在encryptedSessionKey内传递，其中加密函数应该使用主密钥和paramS中的初始矢量加密会话密钥。对于EOFB模式，未加密的掺杂密钥在paramS内的ClearSalt传递。encryptedSaltingKey应该传递加密的掺杂密钥，其中加密函数应该使用主密钥和paramSsaltIV加密掺杂密钥。clearSaltingKey可以包含一个未加密的掺杂密钥，在这种情况下，encryptedSaltingKey应该为空，反之亦然。传输未加密的掺杂密钥只在不会遇到安全问题的时候使用，在其他情况下，建议加密掺杂密钥。

对于更新主角色的发送方向的加密密钥，主角色发送了EncryptionUpdateCommand后，应继续使用原来的会话密钥加密媒体，直到收到EncryptionUpdateAck消息。主角色收到EncryptionUpdateAck后可以使用新的会话密钥加密媒体，另一方面，从角色在收到EncryptionUpdateCommand消息后，就可以开始使用新的密钥。

注：主角色可以选择任何动态载荷类型值，因为动态载荷类型值的作用范围只局限在这个媒体通道端口（即允许多个不同的媒体通道的动态载荷类型值相同的情况发生）。

注：对于更新从角色的发送方向的加密密钥，从角色可以不响应EncryptionUpdateAck消息，因为主角色可以根据RTP流中动态载荷类型值的改变来判断是否新的密钥生效。

图E.6显示了主角色媒体通道的密钥更新过程。在该例子中，从角色发起密钥更新请求，从角色给主角色发送一个MiscellaneousCommand消息，其中logicalChannelNumber包含逻辑通道号，type选择EncryptionUpdateRequest，direction设置为masterToSlave，EncryptionUpdateRequest中的sharedSecret应设置为TRUE。如果是主角色启动密钥更新，则不用发送该请求消息。



图E.6 请求更新主角色的媒体通道的密钥

主角色响应从角色请求或是代表自己，应该发送一个EncryptionUpdateCommand消息，其中MiscellaneousCommand的logicalChannelNumber设置为逻辑通道号，direction设置为masterToSlave，encryptionSync内的synchFlag设置为新的动态RTP载荷号。h235key应该携带新的会话密钥。应该在h235key内的generalID包含主角色的终端标识符，在paramS中包含使用的初始矢量。加密的会话密钥应该在encryptedSessionKey内传递，其中加密函数应该使用主密钥和paramS中的初始矢量加密会话密钥。对于EOFB模式，未加密的掺杂密钥在paramS内的ClearSalt传递。encryptedSaltingKey应该传递加密的掺杂密钥，其中加密函数应该使用主密钥和paramSsaltIV加密掺杂密钥。clearSaltingKey可以包含一个未加密的掺杂密钥，在这种情况下，encryptedSaltingKey应该为空，反之亦然。传输未加密的掺杂密钥只在不会遇到安全问题的时候使用，在其他情况下，建议加密掺杂密钥。

从角色应该响应MiscellaneousCommand消息来告知收到新的会话密钥，其中logicalChannelNumber包含逻辑通道号，encryptionUpdateAck字段的synchFlag包含新的动态载荷类型。

E.5.6.3 基于载荷类型的密钥更新和同步

初始加密密钥和动态载荷类型值由主角色提供（在OpenLogicalChannel或OpenLogicalChannelAck消息中EncryptionSync字段）。媒体流的接收方一旦检测到RTP头的载荷类型为指定的动态载荷类型值，即开始使用初始密钥进行解密。

如果协商的媒体通道只传送一种媒体类型，那么synchFlag中的动态载荷类型值替换RTP头中的载荷类型值。否则，如果媒体通道传送多种媒体类型，那么RTP包需要按照IETF RFC2198的描述方法格式化，其中动态载荷类型值作为封装载荷类型，真实的载荷类型在补充的包头中指示。

主角色可以随时提供新的会话密钥。新的密钥和媒体流的同步通过设置RTP载荷类型为新的动态载荷类型值来指示。

注：动态载荷类型的具体取值无关紧要，只需每次密钥更新时改变数值。

E.5.7 非终端设备的交互

E.5.7.1 网关

H.323网关应该视为一个信任单元，包括协议网关和安全网关。终端和网关之间的媒体隐私性是可以保证，但是并不保证网关的另一侧的安全性（即认为是不安全的）。

E.5.7.2 新密钥

ITU-T H.323第8.5节的内容说明了MC如何从会议中驱逐一个终端的过程。主角色可以重新分配加密密钥，从而阻止被驱逐的终端继续接收媒体。

E.5.7.3 信任单元

通常，MCU、网关和网守被认为是信令通道的信任单元。如果H.225.0呼叫通道是受安全保护的并且经过网守路由，那么网守必须是信任单元。如果这些H.323组件必须操作媒体流，那么它们也必须是媒体隐私的信任单元。

防火墙代理服务器（不必是H.323实体）也可能是信任单元，因为它们终止连接，且可以操纵消息和媒体流。

E.5.8 多点过程

E.5.8.1 认证

终端和MC（U）之间的认证方式和点对点通信的认证方式相同。MC（U）应该负责设置认证级别和严格程度的相关策略。如ITU-T H.235.0中所述，MC（U）是一个信任单元；

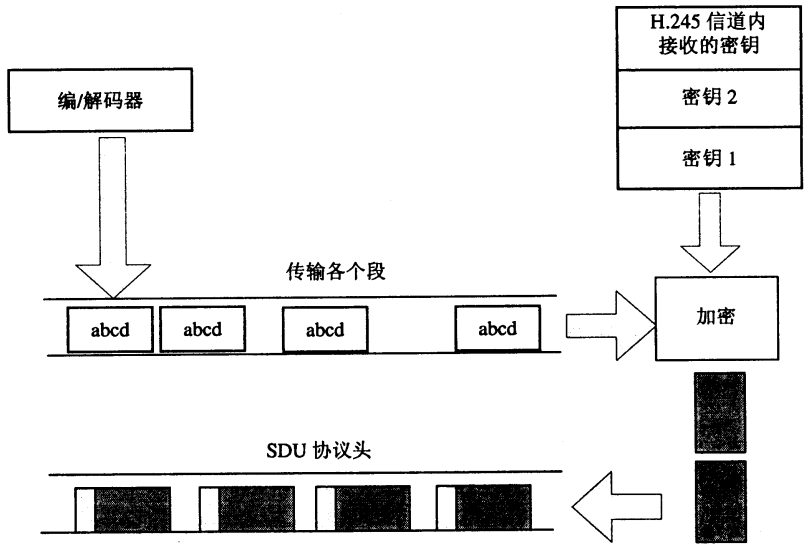
会议中的终端可以受MC（U）采用的认证策略的限制，新的ConferenceRequest/ ConferenceResponse允许终端从MC（U）获得其他与会终端的证书信息。如H.245过程所述，终端可以通过MC（U）请求其他终端的证书，但是无法在H.245信道中进行直接的认证。

E.5.8.2 隐私

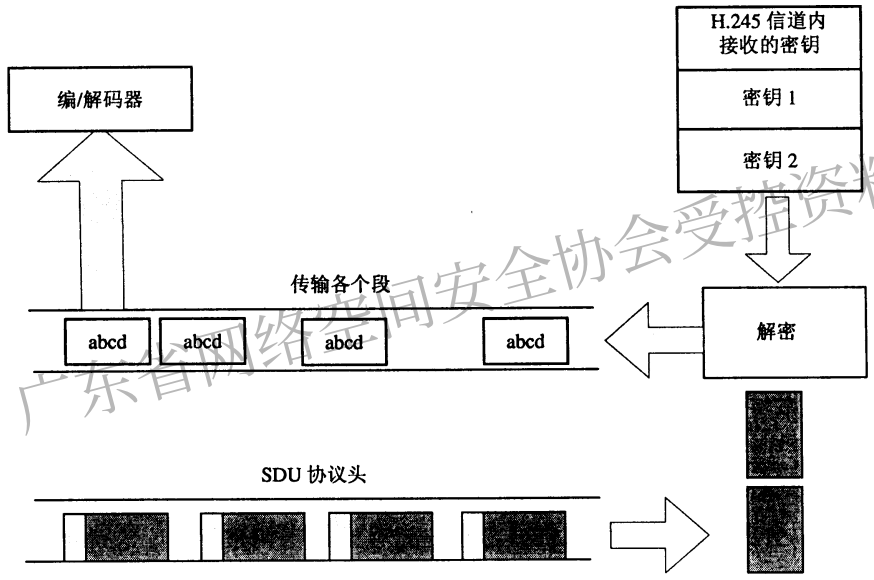
MC（U）赢得主从决定的主地位，因而应给与会终端分配加密密钥。同一个会话内（假设组播会议）的各终端的保密性，可以分配一个共用的密钥或各自分配独立的密钥来达到。这两种模式可以由MC（U）任意选择，不应该受终端控制，除非MC（U）的策略允许。换句换说，一个公共的密钥可用于不同终端的媒体通过中。

E.6 媒体流机密性技术要求

媒体流应使用H.245信道中给出的算法与密钥来进行编码。图E.7与图E.8演示了通用流程。当新密钥被发送者接受并被用来加密时，SDU头应以某种方式向接收者指出要使用新密钥。在ITU-T H.323内，RTP头（SDU）将改变它的载荷类型以指示切换新密钥。



图E.7 媒体加密



图E.8 媒体解密

E.6.1 媒体会话密钥保护

h235Key 包含于 encryptionUpdate 中。h235Key 按 H.235 ASN.1 语法树进行 ASN.1 编码后以 octet string 的方式传入 H.245 消息中。媒体会话密钥可以使用三种机制进行保护。

如果 H.245 信道是安全的，则会话密钥不需要施加任何保护。密钥以明文的形式包含在 H.245 消息中，keyProtectionMethod 使用 secureChannel 选择项。

如果一个预共享秘密与算法在 H.245 信道外部建立，例如，通过 H.225.0 信道协商出来或通过带外方法实现，那么这个预共享秘密用于对媒体加密密钥进行保护，keyProtectionMethod 使用 sharedSecret 选择项。

当 H.245 信道不安全时，可以使用证书（证书也可以用在安全 H.245 信道上），利用证书内的公钥加密媒体会话密钥，keyProtectionMethod 使用 certProtectedKey 选择项。

会议中，任一个与会终端（接收者或发送者）可以请求一个新密钥（encryptionUpdateRequest）。可能的原因之一是终端怀疑丢失了逻辑信道同步。接收一个加密更新请求以后，主角色终端应生成新密钥

并响应请求。主角色终端也可以决定异步分配新密钥，此时应该使用 encryptionUpdate 消息。

当收到一个 encryptionUpdateRequest 消息后，主角色应该发送 encryptionUpdate 消息。如果是多点会议，MC（主角色）在把这个新密钥给发送者之前，应分发这个新密钥给所有的接收者。接收者应在尽可能早的时间内使用这个新密钥。

当收到一个 encryptionUpdateRequest 消息后，主角色应该发送 encryptionUpdate 消息。

一个发送者（假设不是主角色）也可以请求一个新的密钥。如果发送方式是多点会议的成员，应该遵守下面的过程：

1）发送者应该给 MC（主角色）发送 encryptionUpdateRequest。

2）MC 宣生成一个新的密钥并给所有接收终端发送 encryptionUpdate 消息。

3）给所有接收终端分发了密钥后，MC 应该给发送者发送 encryptionUpdate 消息。发送者然后使用这个新的密钥。

E.6.2 媒体反垃圾包

RTP 流的接收者可能希望抵抗针对 RTP/UDP 端口的 DoS 和轰炸攻击。实现了反垃圾包能力的接收者可以快速决定一个 RTP 包是否来自未认证的实体并抛弃它。

反垃圾包能力指示了下面两种反垃圾包机制：

1）用于明文的媒体数据（未加密）的认证。

2）EncryptionCapability 指示支持加密算法，可以联合媒体加密完成认证。

这两个选项都提供了一种通过计算被选取的字段的消息摘要码（MAC）来进行 RTP 包认证的简便方法。该 MAC 可以使用 E.6.2.1 节中定义的对象标识符计算。密码算法是根据：

1）使用一种加密算法（例如，MAC 模式的 DES，参见 ISO/IEC 9797）。DES-MAC 用 OID"N"指示，3DES-MAC 用 OID "O"指示。

2）使用一种单向的密码函数，例如 SHA1，其 OID 为“M”。

MAC 在 antiSpamAlgorithm 的 OID 中指示。算法 OID 也暗示了 MAC 的大小，例如 DES MAC 为 64 比特。为了降低带宽需求，在牺牲某些安全性能的条件下，MAC 可以被截短。例如截短到 32 比特。这种情况需要一个不同的 OID 来表示。反垃圾包方法与载荷加密无关。

反垃圾包使用下面的 RTP 包格式（参见图 E.9），其中 RTP 填充序列解释如下（参见 A.5/H.225.0）。

— RTP 头的 P 比特设置为 1。

— 填充字节按下面的方式添加到载荷的尾部。

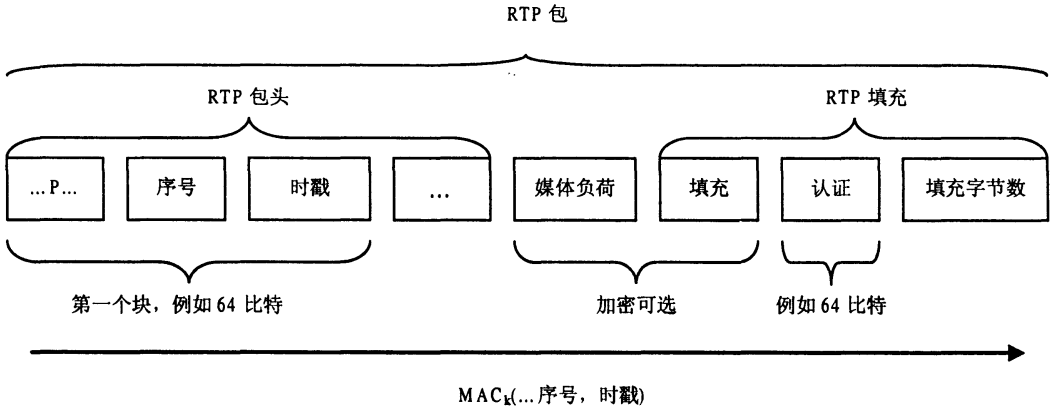


图 E.9 用于反垃圾包的 RTP 包格式

注：如果不使用反垃圾包，那么认证（AUTH）和填充字节数（padlen）字段不使用。

1）只有反垃圾包功能的情况

该情况适用于媒体数据不加密，填充字段为空的时候。RTP 填充内容的最后一个字节包含需要忽视的填充字节数。剩余填充字节携带 MAC 信息。MAC 值应该针对 RTP 头的第一个加密块计算，包括时戳和序列号，MAC 计算过程使用 antiSpamAlgorithm 中指示的加密算法和一个算法必须的对称密钥。共享密钥可以通过静态配置或动态协商。对于大于 64 位的加密块，需要使用更多的 RTP 头的比特位，甚至包括部分的媒体载荷数据。

尽管会话密钥不用于加密媒体，用于 MAC 计算的密钥建议从 H.235 会话密钥分配中获取。密钥管理可以使用支持密钥协商的安全的快速连接过程（参见 ITU-T H.323 附件 J）或手工密钥分配方法。发送方根据上面描述的方法计算 MAC 值，并把结果放到 RTP 填充区的 AUTH 字段。发送方和接收方通过 antiSpamAlgorithm 获知 AUTH 字段的（字节）大小以及 MAC 值的（字节）大小。

接收方的 MAC 验证宜尽早进行，如果可能，最后在 RTP 协议栈中完成，最迟也要在解密和媒体解码前完成。接收方首先按发送方相同的方式重计算 MAC 值，然后对比计算的 MAC 值和 RTP 填充区收到的 MAC 值，如果两个值不匹配，则 RTP 包已经被修改或者是来自未授权的实体。这样，该 RTP 应该抛弃，并记录事件日志。该事件指示可能存在的 DoS 攻击。否则，RTP 包可以进行后续处理，RTP 填充区内容丢弃，有效载荷送给解码器。

注：使用 DES-MAC 只需一个加密操作，相反，SHA1 MAC 是选取一定长度的 RTP 包的部分进行计算，所以加密操作消耗更小的处理资源。

2）发垃圾包联合媒体加密

该案例应用于媒体数据被加密且反垃圾包方法激活的条件下。如果载荷并不结束于加密块的整数倍的界限，需要在载荷之后、MAC 之前添加一些填充字节。媒体载荷按照本节描述加密。

EncryptionCapability 定义了媒体加密算法，而 antiSpamAlgorithm 则定义反垃圾包方法。考虑安全原因，媒体加密和 MAC 计算应使用不同的会话密钥。MAC 密钥 k 由媒体加密密钥 K 经过 SHA1 哈希运算后得到。

k = SHA1（K），按网络字节序截取适量的比特数目构成 MAC 密钥 k。当 antiSpamAlgorithm 指示一种加密算法时，截取的比特应该按正确的方法转换为加密密钥，例如正确设置 DES 的奇偶效验位。

在接收方成功验证了 RTP 包后，进行载荷解码，然后丢弃 RTP 填充内容。

E.6.2.1 对象标识符

表E.5 反垃圾包使用的对象标识符

对象标识符引用	对象标识符值	描述
“M”	{itu-t（0）recommendation（0）h（8）235 version（0）2 8}	使用HMAC-SHA1-96算法
“N”	{iso（1）identified-organization（3）oiw（14），secsig（3）algorithm（2）desMAC（10）}	使用DES（56位）MAC（参见ISO/IEC 9797），64位MAC
“O”	{iso（1）identified-organization（3）oiw（14）secsig（3）algorithm（2）desEDE（17）}	使用3DES（168位）MAC（参见ISO/IEC 9797）

E.6.3 RTP/RTCP 议题

加密RTP流的用法将遵守[RTP]文档中推荐的通用方法。媒体加密针对每个包独立处理。

注：如果RTP包的大小超过MTU尺度，部分数据的丢失将导致整个RTP包无法解密。

RTP头不应被加密。对于声音/视频解码器，包括声音/视频载荷头在内的整个声音/视频载荷都应被加密。新的密钥和待解密媒体根据动态载荷类型进行同步（参见E.5.6.3节）。

加密操作只针对RTP包的载荷部分，RTP头不能被加密。包长度必须是8比特的倍数。传输层或网络层如何封装RTP包不属于本附录的范围。所以模式必须允许包的丢失和乱序问题。

由于允许包的丢失，因此加密必须是无状态的。每个包宜独立解密。两种块加密模式应遵守下列规则：

E.6.3.1 初始矢量

大部分的块加密模式包含“链”式操作，某个加密块的运算依赖于上个加密块的输出。因此，在处理RTP包的首个加密块时，需要提供一个初始矢量。初始矢量的长度等于加密块的长度。

当使用CBC模式的块加密算法时，需要一个初始矢量（IV）。初始矢量的长度和加密块的长度相同。例如DES和3DES的初始矢量长度为64比特，AES的初始矢量长度为128比特。

1）CBC模式初始矢量

对于CBC模式，IV构造方式如下：序列号（Seq#）和时戳（Timestamp）拼接一起，构成SSTTTT模式，其中SS为2字节的RTP序号（Seq#），TTTT为4字节的时戳（timestamp）。该模式被重复多次直到产生至少B字节（B为加密块长度）。最后截取B字节构成初始矢量。例如64比特的初始矢量为SSTTTTSS，128比特的初始矢量为SSTTTTSSTTTTSSTT。需要注意的是，个别情况下，用该方法产生的初始矢量可能是弱安全的。

2）EOFB模式初始矢量

OFB模式下，每一个RTP包都需要初始向量。

正如[SRTP]文档中定义，每一个RTP包都有一个隐含的48比特的包索引值i。其中，$i=(2^{16})\times ROC+SEQ$，SEQ是RTP包序号，当SEQ计算到65536轮回后，ROC加1，ROC占用32比特

开始时，ROC应设置为0，当SEQ完成一个循环后，ROC加1并取2^{32}的模。

初始向量=（I‖T [‖I‖T…]），T是32比特的RTP时戳，‖表示两个比特串的拼接。[]表示重复I‖T至达到加密块的长度。

注：I和初始向量并不会通过网络传输对方，只是本地的一个计算。

当丢包和乱序情况发生时，建议接收方按下面的方法计算一个估计值i：

计算$i=(2^{16})\times V+SEQ1$，V从{ROC－1，ROC，ROC+1}模2^{32}中选择，选择i最接近$2^{16}\times ROC+SEQ2$。其中SEQ1是收的包序号，SEQ2是接收方记录的当前序号，把这个i用于解码过程。然后用SEQ1更新SEQ2，用V更新ROC。更多细节参见[SRTP，3.2.1节]说明。

E.6.3.2 填充

ECB或CBC模式要求明文数据是加密块的整数倍，而CFB和OFB模式则能够处理任意的长度数据。有两种方法处理不是加密块倍数的包。

1） 在ECB和CBC模式下，使用密文窃取（ciphertext stealing）处理不完整的加密块，对于CFB和OFB模式，无需特殊考虑。

2）按照[RTP 5.1]描述的方法进行填充。

[RTP 5.1]描述一种把载荷填充为块的倍数的填充方法。填充数据的最后字节指示了总共填充的字节数（包括最后字节）。此时P比特应置1。填充数据的内容由加密算法决定。

H.235设备必须同时支持这两种PADDING模式。

解码方判断PADDING模式的方法：如果RTP头里的P比特被设置，那么采用的是RTP PADDING模式。如果RTP负荷不是块的整数倍，并且RTP头里的P比特没有设置，那么可推断使用了STEALING PADDING模式，否则载荷是加密块的整数倍，不使用填充。

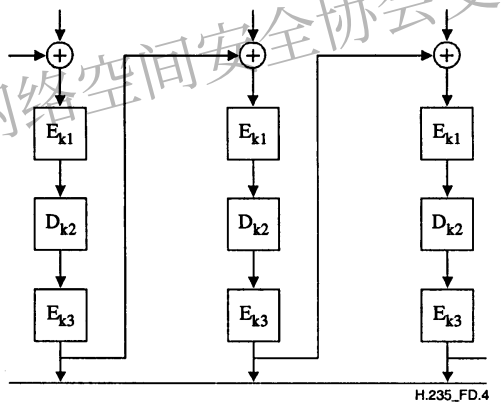
E.6.3.3 安全的载荷流

基于H.323的网络利用H.245信令建立和协商窄带的数据通道（如Modem-over-IP）和用来打包多载荷流的RTP通道。对于单一载荷类型的媒体流或者是某个媒体通道的FEC通道，encryptionSync中的动态载荷类型应替换缺省的载荷类型。对于封装（encapsulating）的媒体流（例如冗余编码或IETF RFC2198编码的FEC流），encryptionSync中的动态载荷类型应替换封装载荷类型。对于多载荷类型的流，应该忽视encryptionSync的syncflag字段中的动态载荷类型值，而应该使用multiplePayloadStreamElement中的载荷类型值。

EncryptiondateCommand消息应用于分发新的密钥材料的改进的密钥更新过程（参见E.5.6.2节）。MultiplePayloadStream只用于多载荷流需要修改密钥的情况，这时应忽视encryptionSync中的动态载荷类型值。

E.6.4 外层 CBC 模式的 3DES

如图 E.10 所示，168 比特的外层 CBC 模式的 3DES 可用于本安全轮廓。在图中，每个 K*i* 指一个 56 比特的密钥。每个加密块与解密块和解密块应使用不同的密钥。未发现 DES 的弱密钥会在 3DES 中导致弱安全问题。然而，遵守本轮廓的实现宜拒绝使用那些引起弱 DES 密钥问题的密钥（参见 IETF RFC2045）。更多的关于 3DES 的信息可以从[Schneier]和 IETF RFC2405 中获取。



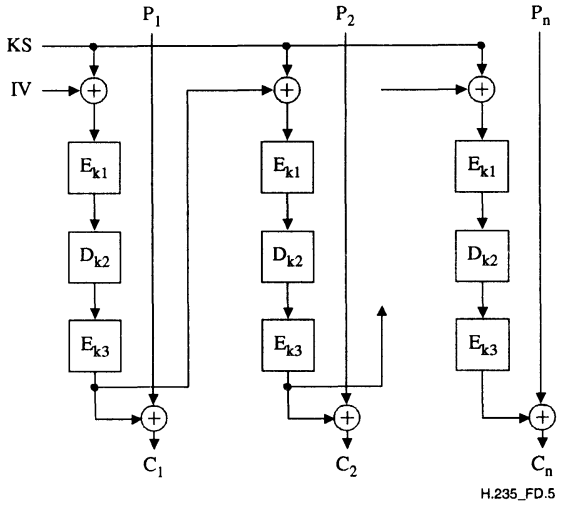
图E.10 外层CBC模式的3DES加密算法

E.6.5 EOFB 模式下的 DES

声音可以使用操作在EOFB模式下的算法加密。EOFB模式允许利用并行计算能力。当工作在EOFB模式时，考虑性能和安全性原因，建议反馈整个加密块（例如，DES算法反馈整个64比特，即n=j=64）。然而，由于EOFB加密并不链接加密块（仅和最邻近的加密块相关），在对付分析明文统计属性的攻击手段方面，EOFB被怀疑是有安全隐患的。为此，建议定期进行密钥更新，最迟要在初始矢量轮回之前更新密钥。计算初始矢量的方法参见E.6.3.1节。

E.6.6 外层 EOFB 模式下的 3DES

如图E.11所示，168比特的外层EOFB模式的3DES可用于本安全轮廓。在图中，每个K*i*指一个56位的密钥。每个加密块和解密块应使用不同的密钥。未发现DES的弱密钥会在3DES中导致弱安全问题。然而，遵守本轮廓的实现宜拒绝使用那些引起弱DES密钥问题的密钥（参见IETF RFC2045）。更多的关于3DES的信息可以从[Schneier]和IETF RFC2405中获取。



图E.11　外层EOFB模式的3DES算法

E.7　对象标识符

表 E.6 列出了使用到的所有对象标识符（Object identifiers）。有些标识符针对 ITU-T H.235 版本 1 和 ITU-T H.235 版本 2。

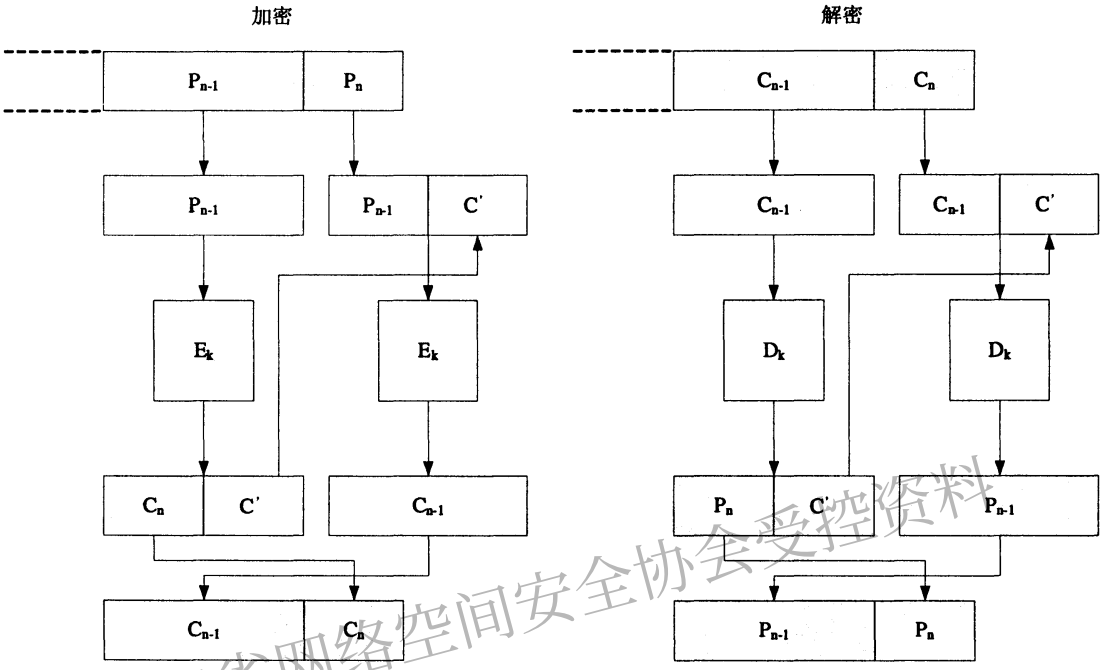
表E.6　Object identifiers

对象标识符引用	对象标识符值	描　述
“Dhdummy”	{itu-t（0）recommendation（0）h（8）235 version（0）2 40} {itu-t（0）recommendation（0）h（8）235 version（0）3 40}	显式提供的非标准的DH组
“DH1024”	{itu-t（0）recommendation（0）h（8）235 version（0）2 43} {itu-t（0）recommendation（0）h（8）235 version（0）3 43}	1024比特DH组
“DH1536”	{itu-t（0）recommendation（0）h（8）235 version（0）3 44}	1536比特DH组
“X”	{iso（1）member-body（2）us（840）rsadsi（113549）encryptionalgorithm（3）2}	使用RC2（56比特）或CBC模式的RC2以及512比特 DH组的声音加密算法
“X1”	{itu-t（0）recommendation（0）h（8）235 version（0）3 27}	使用RC2（56比特）或EOFB模式的RC2以及512比特 DH组的声音加密算法
“Y”	{iso（1）identified-organization（3）oiw（14）secsig（3）algorithm（2）descbc（7）}	使用CBC模式的DES（56比特）以及512比特DH组的声音加密算法
“Y1”	{itu-t（0）recommendation（0）h（8）235 version（0）3 28}	使用EOFB模式的DES（56比特）、反馈长度为64位以及512比特DH组的声音加密算法
“Z1”	{itu-t（0）recommendation（0）h（8）235 version（0）3 29}	使用外层EOFB模式的3DES（168比特）、反馈长度为64位以及1024比特DH组的声音加密算法
“Z2”	{itu-t（0）recommendation（0）h（8）235 version（0）3 30}	使用EOFB模式的AES（128比特）以及1024比特DH组的声音加密算法
“Z3”	{joint-iso-itu-t（2）country（16）us（840）organization（1）gov（101）3 nistAlgorithm（4）aes（1）cbc（2）}	使用CBC模式的AES（128比特）以及1024比特DH组的声音加密算法
“Z”	{iso（1）identified-organization（3）oiw（14）secsig（3）algorithm（2）desEDE（17）}	使用外层CBC模式的3DES（168比特）以及1024比特DH组的声音加密算法

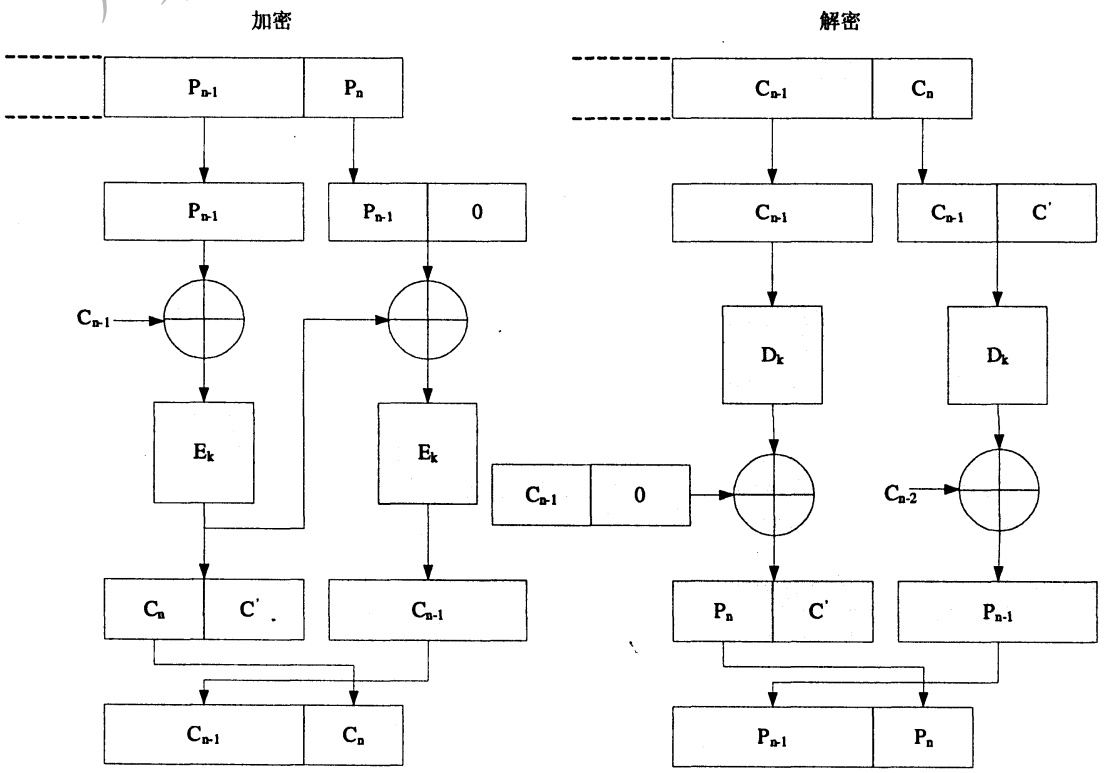
附 录 F
（资料性附录）
H.323实现细节

F.1 密文窃取填充方法（ciphertext stealing）

[Schneier]文献191~196页有关于密文窃取的描述，图F.1~F.7说明了这项技术。



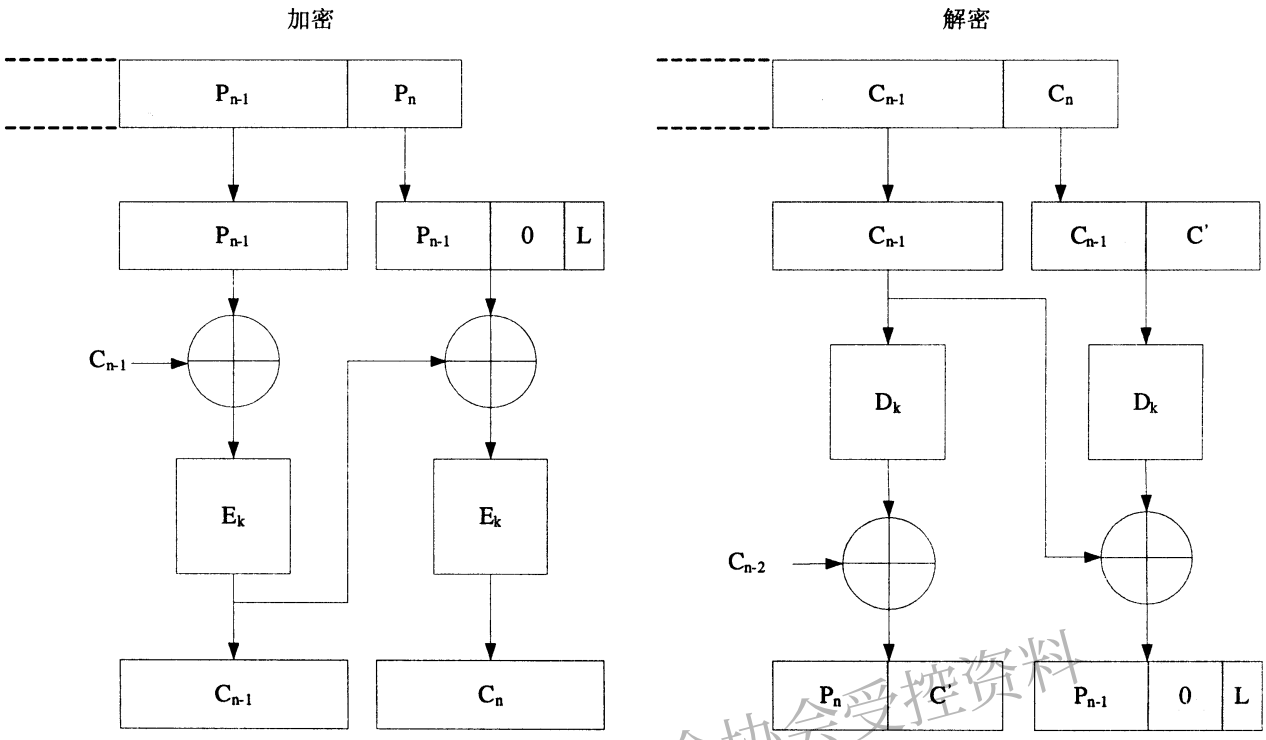
图F.1 ECB模式的密文窃取



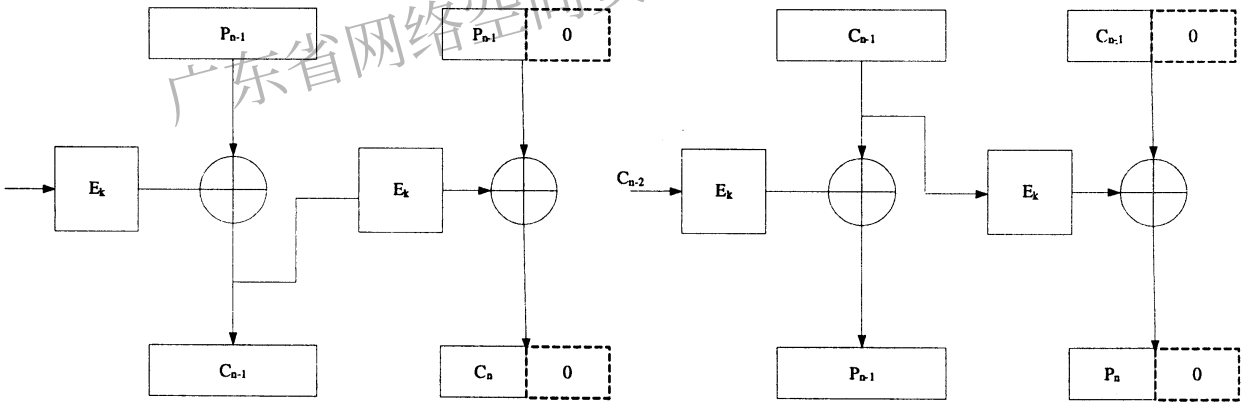
图F.2 CBC模式的密文窃取

注：密文窃取要求负载必须大于一个加密块的长度。

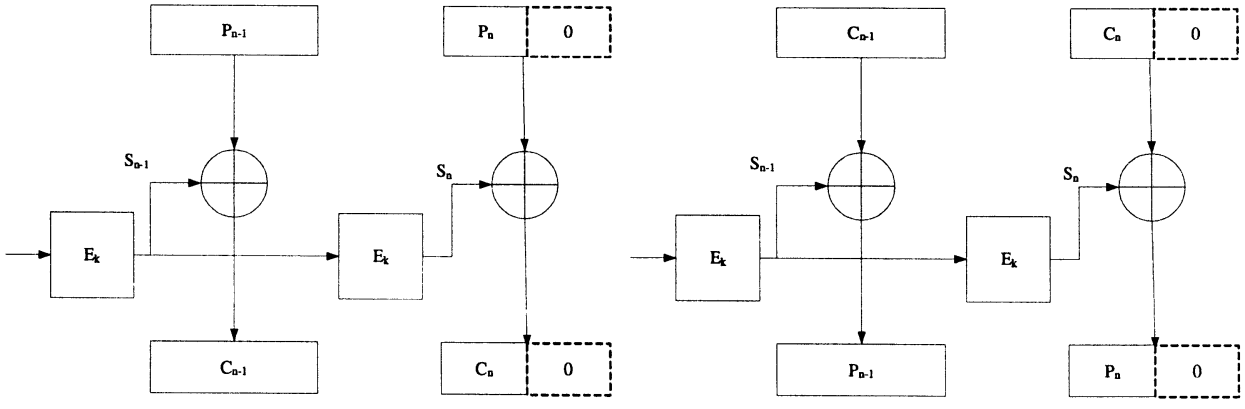
当密文窃取用于CBC模式，如果明文小于等于一个加密块，那么初始矢量充当上一个加密块。



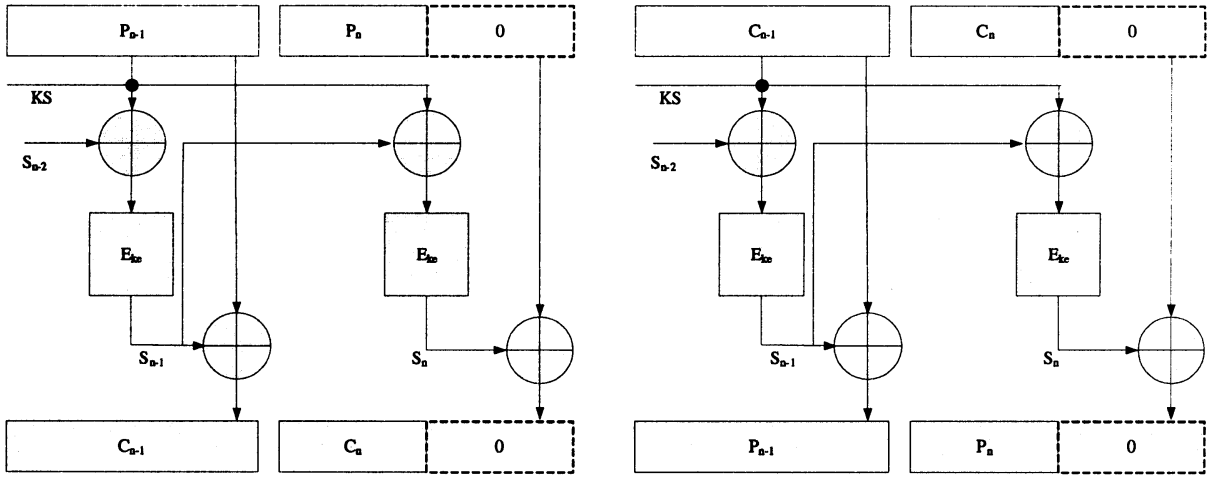
图F.3　无填充的CBC模式



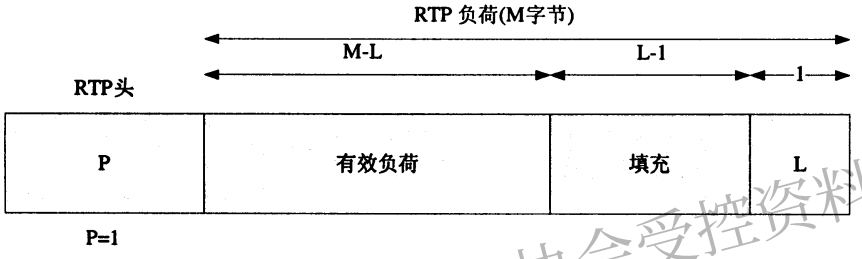
图F.4　无填充的CFB模式



图F.5　无填充的OFB模式



图F.6　无填充的EOFB模式



图F.7　RTP填充模式

附 录 G
（资料性附录）
本标准章条编号与ITU-T H.235章条编号对照

表 G.1 给出了本标准章条编号与 ITU-T H.235 章条编号对照一览表。

表G.1 本标准章条编号与ITU-T H.235章条编号对照

本标准章条编号	ITU-T H.235 系列章条编号
1	无
2	无
3	无
4	无
5	无
6	无
7	无
8	无
9	无
10	无
11	无
12	无
13	无
附录 A	无
附录 B	ITU-T H.235.0 附件 A
附录 C	—
C.1	ITU-T H.235.4 1
C.2、C.3	ITU-T H.235.4 7
C.4	ITU-T H.235.4 8
C.5	ITU-T H.235.4 9
C.6	ITU-T H.235.4 10
C.7	ITU-T H.235.4 11
C.8	ITU-T H.235.4 12
C.9	无
C.10	ITU-T H.235.4 14
附录 D	—
D.1	ITU-T H.235.1 1
D.2	ITU-T H.235.1 5
D.3	ITU-T H.235.1 6
D.4	ITU-T H.235.1 7
D.5	ITU-T H.235.1 8
D.6	ITU-T H.235.1 9
D.7	ITU-T H.235.1 10

表 G.1（续）

本标准章条编号	ITU-T H.235 系列章条编号
D.8	ITU-T H.235.1 11
D.9	ITU-T H.235.1 12
D.10	ITU-T H.235.1 13
D.11	ITU-T H.235.1 14
D.12	ITU-T H.235.1 15
附录 E	—
E.1	ITU-T H.235.6 1
E.2	ITU-T H.235.6 5
E.3	ITU-T H.235.6 6
E.4	—
E.4.1	ITU-T H.235.6 7.1
E.4.2	ITU-T H.235.6 7.2
E.4.3	ITU-T H.235.6 7.3
E.4.4	ITU-T H.235.6 7.4
E.4.5	ITU-T H.235.6 7.5
E.4.6	—
E.4.6.1	ITU-T H.235.6 7.6.1
E.4.6.1.1	ITU-T H.235.6 7.6.1.1
E.4.7	ITU-T H.235.6 7.7
E.4.8	ITU-T H.235.6 7.8
E.5	ITU-T H.235.6 8
E.6	—
E.6.1	ITU-T H.235.6 9.1
E.6.2	ITU-T H.235.6 9.2
E.6.3	—
E.6.3.1	ITU-T H.235.6 9.3.1
E.6.3.2	ITU-T H.235.6 9.3.2
E.6.3.3	ITU-T H.235.6 9.3.4
E.6.4	ITU-T H.235.6 9.4
E.6.5	ITU-T H.235.6 9.5
E.6.6	ITU-T H.235.6 9.6
E.7	ITU-T H.235.6 11
附录 F	—
F.1	ITU-T H.235.6 附录 I.1
附录 G	无

参 考 文 献

[1] ITU-T J.170（2005）,IPCablecom security specification（IP电缆安全规范）

[2] IETF RFC 2104（1997）,HMAC: Keyed-Hashing for Message Authentication（用于消息认证码的密钥散列）

[3] IETF RFC 2268（1998）,A Description of the RC2（r）Encryption Algorithm（RC2加密算法描述）

[4] IETF RFC 2405（1998）,The ESP DES-CBC Cipher Algorithm With Explicit IV（带有显示初始向量IV的ESP DES-CBC密码算法）

[5] IETF RFC 2406（1998）,IP Encapsulating Security Payload（ESP）（IP封装安全载荷（ESP））

[6] IETF RFC 2408（1998）,Internet Security Association and Key Management Protocol（ISAKMP）（因特网安全关联与密钥管理协议（ISAKMP）

[7] IETF RFC 2409（1998）,The Internet Key Exchange（IKE）（因特网密钥交换（IKE））

[8] IETF RFC 2412（1998）,The OAKLEY Key Determination Protocol（OAKLEY密钥确定协议）

[9] IETF RFC 3550（2003）,RTP: A transport Protocol for Real-Time Applications（RTP：实时应用传输层协议）

[10] IETF RFC 3711（2004）,The Secure Real-Time Transport Protocol（安全实时传输协议）

[11] IETF RFC 4120（2005）,The Kerberos Network Authentication Service（V5）（Kewrberos网络认证服务（V5））

[12] [Daemon] DAEMON（J.）,Cipher and Hash function design（密码与散列函数设计）Ph.D. Thesis，Katholieke

[13] [DES FIPS-46-2] US National Institute of Standards，Data Encryption Standard，Federal Information Processing Standard（数据加密标准，联邦信息标准）（FIPS）Publication 46-2，December 1993，

[14] [DES FIPS-74] US National Institute of Standards，Guidelines for Implementing and Using the Data Encryption Standard（实现和使用数据加密标准指导）Federal Information Processing Standard（FIPS）Publication 74，April 1981,http://www.itl.nist.gov/div897/pubs/fip74.htm

[15] [DES FIPS-81] US National Institute of Standards，DES Modes of Operation，（DES运算模型）Federal Information Processing Standard（FIPS）Publication 81,December 1980,http://www.itl.nist.gov/fipspubs/fip81.htm

[16] [FIPS PUB 180-1] NIST，FIPS PUB 180-1: Secure Hash Standard（安全散列标准）April 1995．http://csrc.nist.gov/fips/fip180-1.ps

[17] [FIPSPUB180-2] Federal Information Processing Standard FIPS PUB 180-2,Secure Hash Standard（安全散列标准）,U. S.Department of Commerce，Technology Administration，National Institute of Standards andTechnology,1 August 2002

[18] [LI] ETSI TR 101 772 V1.1.2,Telecommunications and Internet Protocol Harmonization Over Networks（TIPHON）Release 3; Service Independent requirements definition;Lawful interception – top level requirements（电信与因特网协议在网络上融合（TIPHON）第3版；服务无关需求定义；法律解释——顶

级需求）

[19] [OIW] Stable Implementation – Agreements for Open Systems Interconnection Protocols:Part 12 – OS Security（开放系统互联协议协商：第12部——操作系统安全）；Output from the December 1994 Open Systems Environment Implementors' Workshop （OIW）;http://nemo.ncsl.nist.gov/oiw/agreements/stable/OSI/12s_9412.txt

[20] [Schneier] SCHNEIER（B.）,Applied Cryptography: Protocols,Algorithms,and Source Code in C,2nd Edition（应用密码学 :协议，算法与C源代码,第2版）John Wiley & Sons,Inc.，1995

中华人民共和国
通信行业标准
H.323网络安全技术要求
YD/T 1701-2007
*
人民邮电出版社出版发行
北京市崇文区夕照寺街14号A座
邮政编码：100061
北京新瑞铭印刷有限公司印刷

*
开本：880×1230　1/16　　2007年12月第1版
印张：5.75　　2007年12月北京第1次印刷
字数：176千字
ISBN 978-7-115-1574/08-18
定价：50元
本书如有印装质量问题，请与本社联系　电话：(010)67114922