ICS 33.160.25
M 74
备案号：

SJ

中华人民共和国电子行业标准

SJ/T 11407.1—2009

数字接口内容保护系统技术规范
第1部分：系统结构

Content protection specifications for digital interface—
Part 1: System architecture

2010-01-20 发布　　2010-03-01 实施

中华人民共和国工业和信息化部 发布

目　　次

前　言

SJ/T 11407《数字接口内容保护系统技术规范》拟根据数字接口内容保护需求分为几个部分。

本部分为SJ/T 11407的第1部分。

本部分的附录A为资料性附录。

本部分由全国音频、视频及多媒体设备与系统标准化技术委员会归口。

本部分由四川长虹电器股份有限公司、西安电子科技大学、中国电子技术标准化研究所、北京浦奥得数码技术有限公司、TCL多媒体技术控股有限公司、青岛海信电器股份有限公司、深圳创维-RGB电子有限公司、深圳国微技术有限公司、厦门华侨电子股份有限公司、上海广电（集团）有限公司、数源科技股份有限公司，康佳集团股份有限公司等单位共同起草。

本部分主要起草人：王育民、詹阳、范科峰、葛建华、张恩阳、刘贤洪、杨金峰、张素兵、康红娟、田海博、高明、裴庆祺，李新国、张新法、王艳艳、余有勇、任飞、潘贡、于国福等。

引　言

随着数字化技术、网络技术、计算机技术、多媒体技术、存储技术的发展，对数字内容的复制、修改、传播变得非常容易。数字电视的发展对数字内容保护提出了迫切的要求。本标准的目的主要是保护设备接口之间数字内容的传送。

本标准规定了数字接口安全传输数字内容时所需遵照的规范，为有保护需求的数字内容在数字设备间安全传输提供了有效的技术实现方案。

鉴于本部分只为数字设备间内容安全传输提供技术框架，不包括技术框架中的一些技术细节，比如协议消息长度数值、异常处理等。因此，这些参数数值必须根据具体接口应用确定，在相关接口应用规范中给出。

本标准的应用包括但不限于以下领域：

数字接口；

数字电视；

数字家庭；

移动设备。

本标准的发布机构提请注意如下事实，声明符合本标准时，可以使用涉及以下相关专利。

本标准的发布机构对于专利的范围、有效性和验证资料不提出任何看法。

专利持有人已向本标准的发布机构保证，他愿意同任何申请人在合理和非歧视的条款和条件下，就使用授权许可证进行谈判。在这方面，该专利持有人的声明已在本标准的发布机构备案。有关资料可从以下地址获得：

专利名称1：内容保护系统以及方法（中国专利申请号：200510115315）；

专利名称2：一种椭圆曲线密码系统及实现方法（中国专利申请号：200510115512）；

专利名称3：电子设备接口间基于公钥证书的认证密钥协商和更新方法（中国专利申请号：200510124342）；

专利持有人：北京浦奥得数码技术有限公司；

联系人：康静；

地址：北京市朝阳区酒仙桥路14号兆维华灯大厦A215、213室；

邮编：100016；

电话：010-58671045；

传真：010-58071042。

请注意除上述已经识别出的专利外，本标准的某些内容有可能涉及专利。本标准的发布机构不应承担识别这些专利的责任。

数字接口内容保护系统技术规范
第1部分：系统结构

1 范围

本部分确立了一个用于消费电子数字设备间安全传输数字内容的内容保护系统框架。

本部分适用于消费电子数字设备，其它数字设备也可参照使用。

本部分适用的数字设备接口包括DisplayPort接口、HDMI接口、DTV-CI接口、DVI接口、IEEE1394接口、USB接口以及其它数据接口。

2 规范性引用文件

下列文件中的条款通过本部分的引用而成为本部分的条款。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部分，然而，鼓励根据本部分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部分。

GB/T 5271.8—2001 信息技术 词汇 第8部分：安全

RFC 2104 用于消息认证的带密钥散列函数

3 术语和定义

下列术语和定义适用于本部分。

3.1

数字接口内容保护系统(DICP) content protection system for digital interface

用于实现消费电子数字设备间数字内容安全传输的模型系统。

3.2

DICP设备 DICP device

具有DICP功能的独立物理设备，是构成DICP的基本要素。

3.3

DICP接口 DICP interface

DICP设备中实现数字内容安全传输的与其它DICP设备连接的通信接口。

3.4

识别管理单元 identifying management unit

DICP设备中用于对数字内容保护标志进行识别并对设备自身DICP接口进行管理的功能性实体，可选地具有内容权限识别功能。

3.5

DICP发送端 DICP transmitter

DICP中能够通过一个或多个DICP接口加密并发送数字内容的单元。

3.6

DICP接收端 DICP receiver

DICP中能够通过一个或多个DICP接口接收并解密（加密过的）数字内容的单元。

3.7

DICP转发器 DICP repeater.

DICP中同时具有DICP发送端和DICP接收端功能的设备，它首先通过一个或多个DICP接口接收并解密数字内容（加密过的），然后再通过一个或多个DICP接口重新加密并发送数字内容。

3.8

DICP-X接口 DICP-X interface

DICP设备中将DICP规范与X类型接口相结合后形成的接口。

3.9

源接口 source interface

DICP设备中用于实现数据发送功能的DICP接口。

3.10

目的接口 destination interface

DICP设备中用于实现数据接收功能的DICP接口。

3.11

证书吊销列表（CRL） certificate revocation list（CRL）

一个已标识的列表，它指定了一套证书发布者认为无效的证书。

3.12

公钥基础设施（PKI） public-key infrastructure（PKI）

支持公钥管理体制的基础设施，提供鉴别、加密、完整性和不可否认性服务。

3.13

证书认证中心（CA） certificate athority（CA）

负责创建和分配证书，受用户信任的权威性机构。

3.14

根CA root CA

信任体系中的最高管理机构和信任源点。

3.15

接口认证状态寄存器 interface state register

识别管理单元中保存的设备自身DICP接口与识别管理单元认证结果的状态寄存器。

3.16

S单元 S unit

DICP设备中，数字内容传输的起点和终点，即数字内容从输出设备的S单元向外流向输入设备的S单元。

注：S可以是长期存储体，也可以是实时节目流处理过程中的临时缓存区，在DICP接收端，S还可能是显示屏幕。

4 符号和缩略语

4.1 符号

下列符号适用于本部分。

A‖B	字符串A和B进行顺序连接所构成的新的字符串。
AUA_i	在CRL更新协议里用到的第i个消息认证码。
Cert_Root	DICP根CA所持有的公钥证书。
Cert_X	X所持有的公钥证书。
Cert_X_Adm	Cert_X的签发者所持有的公钥证书。
CRL_X	X内存储的CRL。

$Cert\_CRL_x\_Adm$　　签发CRL_x的CA所持有的公钥证书，可以是DICP根CA所持有的公钥证书，也可以是某一个具有相应授权的二级CA所持有的公钥证书。

D_Counter　　计算加密的数据量，32比特，计数单位比特。

D_W_Counter　　解密错误数目计数器，其计数值的上限为D_W_Max。

D_W_Max　　D_W_Counter的计数值上限，表示允许出现的解密错误的数目。

G　　椭圆曲线基点。

ID_Des　　协议响应方接口公钥证书中的持有者标志。

ID_Source　　协议发起方接口公钥证书中的持有者标志。

ID_X　　模块X的证书中的持有者ID。

K_E　　加密密钥、解密密钥。

K_E[0]　　存储偶加密密钥的专用存储区。

K_E[1]　　存储奇加密密钥的专用存储区。

K_M　　接口存储的主密钥，与共享方的唯一标志共同存储。

K_MAC　　完整性密钥、完整性验证密钥。

K_MAC[0]　　存储偶完整性密钥的专用存储区。

K_MAC[1]　　存储奇完整性密钥的专用存储区。

LSB　　最低权重位。

S_CRL　　DICP安全模块中用于存储CRL的专用存储区。

SK　　与证书里面的公钥相对应的私钥。

KDI_Record　　关键数据信息记录。

x、y　　DH交换中生成的临时随机数。

xyG　　DH交换生成的共享密钥。

4.2 缩略语

下列缩略语适用于本部分。

CRL　证书吊销列表 (Certificate Revocation List)

CRRT　证书吊销记录类型 (Certificate Revocation Record Type)

DH　DH密钥交换算法 (Diffie-Hellman)

ECC　椭圆曲线密码系统 (Elliptic Curves Cryptography)

ISI　接口状态信息 (Interface State Information)

LC　连接数量 (Linked Count)

LDI　连接设备信息 (Linked Device Information)

LD　连接深度 (Linked Depth)

DICP　数字接口内容保护系统 (Content Protection System for Digital Interface)

KDI　关键数据信息 (Key Data Information)

5 数字接口内容保护系统描述

5.1 DICP概述

5.1.1 DICP组成

一个DICP应包含一个DICP发送端、一个或多个DICP接收端、零个或多个DICP转发器。本部分描述的DICP规定了数字内容在DICP发送端和DICP接收端之间传输时传输参与各方所需完成的一组操作。

当传输的数字内容无需保密时，数字内容直接在DICP设备间传输。

当传输的数字内容需要保密时，应首先执行认证过程，认证过程包括DICP设备中识别管理单元与设备自身DICP接口间的认证以及直接相连的两个DICP接口间的认证。认证完成后，在直接相连的两个DICP

接口间加密传送数字内容。同时，DICP必须能随时通过CRL更新来维护系统的完整性，把被吊销的设备和接口排除在系统之外。

注1：在接口和本地识别管理单元之间，也需要执行认证协议，该协议参考本标准中接口之间的认证协议。

注2：如果在接口和本地识别管理单元之间使用本标准规定的安全传输技术，可以去掉接口和本地识别管理单元具有物理安全传输信道的假设，这部分内容不属于本标准的范围。

5.1.2 DICP 拓扑结构

DICP设备间通过DICP接口连接，DICP支持两种连接拓扑结构：树型拓扑结构和总线型拓扑结构。

图1展示了DICP设备的两种连接拓扑结构，其中X，Y，Z表示不同的DICP接口。从X接口向下构成树型拓扑结构，数字内容以单播的方式从DICP发送端经DICP转发器流向DICP接收端；从Z接口向下构成总线型拓扑结构，数字内容以广播的方式直接从DICP发送端流向DICP接收端。

注：例如X接口可能表示DICP-DisplayPort类型接口，Y接口可能表示DICP-HDMI类型接口，Z接口可能表示DICP-DVI类型接口。

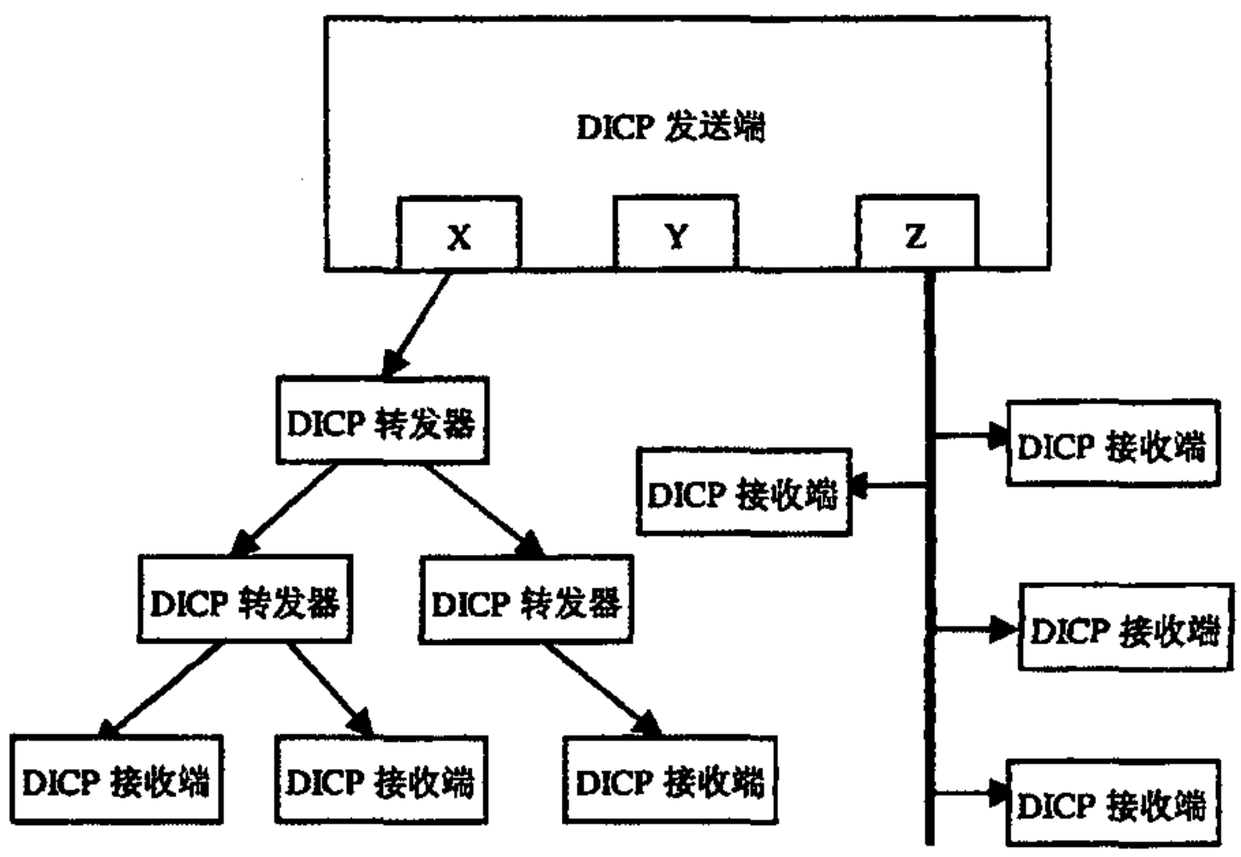


图1 DICP 连接拓扑结构

DICP中，不同的连接拓扑结构导致DICP设备间安全传输的实现方式有很大的差异，主要表现在：

a) 传输方式：树型拓扑结构下，使用单播方式加密传输数字内容；在总线型拓扑结构下，则需要使用广播方式加密传输内容；

b) 连接规模控制机制：树型拓扑结构下，通常使用连接深度和连接数量的方式来控制接收某一内容的设备规模；在总线型拓扑结构下，通常使用加密密钥计数的方式来控制接收某一内容的设备规模；

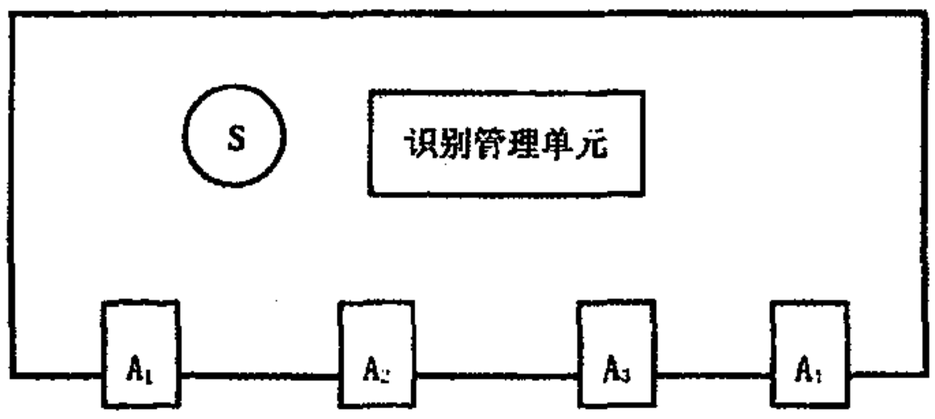
c) 认证协议发起方：树型拓扑结构下，由数字内容发送方根据内容保护要求来触发认证协议；而在总线型拓扑结构下，只有当接收方发现不能解密数字内容时才触发认证协议。

针对上述实现差异，DICP体系根据策略协商来确定传输方式和连接规模控制机制，根据协商的传输方式来决定认证协议发起的流程。

5.2 DICP 设备

一个DICP设备应包含一个识别管理单元、一个S单元和至少一个DICP接口。图2所示的设备是一个典型的DICP设备，其中：识别管理单元和S单元位于设备内部，若干个DICP接口（假设接口A_1－A_3）位于设备的外部边界上，各个接口的类型可以相同，也可以不同。

注：DICP设备可以是各种数字设备，包括机顶盒、数字电视机、数码摄像机、数码照相机、刻录机、介质（如CD、VCD、DVD盘片等）播放器等。



A_1—A_4是设备的接口，应至少有一个是CPS接口

图2　典型的 DICP 设备示意图

DICP设备应具备如下特性：

a) 识别管理单元内包含有 DICP 根证书、二级 CA 证书、设备证书和对应的私钥、CRL 以及接口认证状态寄存器。可以完成的基本操作有：
   1) 提取内容保护标志：识别并提取不同的内容保护标志；
   2) 保护本地信息：保密设备证书私钥信息，保持 DICP 根证书、二级 CA 证书、CRL 以及接口认证状态寄存器的完整性；
   3) 执行认证协议：完成识别管理单元与本设备各个 DICP 接口间的认证；
   4) 产生和验证签名：产生用于认证协议的签名，并能对发送过来的签名进行验证；
   5) 更新本地 CRL：验证 CRL 有效性，更新识别管理单元中的 CRL，并协调更新本设备上各个 DICP 接口中的 CRL；
   6) 控制传输内容：从 S 单元中提取数字内容，并完成内容传输决策；
   7) 控制网络规模：按照本地设定的策略控制下游连接网络的规模。

   注1：识别管理单元还可完成与设备其它内容保护系统的认证。

   注2：识别管理单元还可选地具有处理授权域的能力。

   注3：在处理的节目流中含有不止一种内容保护标志（节目流中含有多个节目）时，按照所有内容保护标志中的最严格的规定执行。

b) DICP 接口内包含有 DICP 根证书、二级 CA 证书、接口证书和对应的私钥以及 CRL。可以完成的基本操作有：
   1) 保护本地信息：保密接口证书的私钥信息，保持 DICP 根证书、二级 CA 证书及 CRL 的完整性；
   2) 执行认证协议：完成与本地识别管理单元间的认证及与其它连接的 DICP 设备上 DICP 接口间的认证；
   3) 安全传输内容：完成数字内容的安全传输；
   4) 验证及更新 CRL：验证 CRL 有效性，及时完成与本地识别管理单元及其它连接的 DICP 设备上 DICP 接口间的 CRL 更新；
   5) 产生和验证签名：产生用于认证、密钥激活等协议的签名，并能对发送过来的签名进行验证。

   注：安全传输内容中包括密钥激活、随机数产生、密钥更新及内容加/解密等操作。

在 DICP 设备内部必须满足如下要求：

a) S 单元能够保证存储或者流经该单元的明文数字内容的完整性和保密性；
b) 在设备内部的传输应该能够防止物理搭线窃听、数据篡改等操作，默认能够保证传输数据的完整性和保密性。

5.3　数据、算法和协议

本条列出了DICP实现中用到的数据、算法和协议。

5.3.1 数据

DICP中用到的要求保密且不能篡改的数据、可以公开但不可篡改的数据分别列于表1和表2。对于某一个具体实现，根据实际需要选择实现表1和表2中全部或者部分数据，对于所实现的部分，需要满足保密性和（或）完整性要求。

注：具体实现可以通过硬件或特殊的软件技术来保证这些数据的保密性或/和完整性。

表1 要求保密且不能篡改的数据

名称	数据长度 bit	说明
私钥 SK	192	每一个具有公钥证书的 DICP 设备及 DICP 接口都有相应的证书私钥，属于长期存储数据
ECC-DH 指数 x 或 y	192	临时密钥数据
DH 共享密钥 xyG	384	临时密钥数据
主密钥 K_M	256	更新频度较低的数据
加密密钥 K_E	128	频繁更新数据
完整性密钥 K_MAC	128	频繁更新数据
CRL 传输认证密钥 K_CRL	256	更新频度较低的数据
广播会话密钥 K_S	128	更新频度较低的数据
广播加密密钥 K_E_G	128	可选项，由接口类型决定实现
广播完整性密钥 K_MAC_G	128	可选项，由接口类型决定实现

表2 要求不可篡改的数据

名称	数据长度 byte	说明
公钥证书 Cert_X	112	Cert_X 可以是 DICP 设备公钥证书，也可以是 DICP 接口公钥证书
二级 CA 证书 Cert_X_Adm	108	签发 Cert_X 的二级 CA 公钥证书
根证书 Cert_Root	108	DICP 根 CA 公钥证书
接口状态寄存器 ISR	16	保存设备识别管理单元与自身 DICP 接口间的认证状态
接口状态信息 ISI	1	接口的相关状态信息
连接数量 LC	1	DICP 发送端用来记录下游设备的连接数量，由具体应用决定
连接深度 LD	1	DICP 发送端用来记录下游设备的连接深度，由具体应用决定
连接设备信息 LDI	858	限制最多存储 127 台设备唯一标识符
广播信道连接数 *n*	1	*n* 为拥有该广播信道加密密钥的下游设备的数量
CRL 存储区 S_CRL	最小 528	用于存储 CRL 和 Cert_CRL_Adm 的专用存储区，其具体容量大小由具体应用决定，设备出厂时其内必须装有 CRL 和 Cert_CRL_Adm

DICP中用到的接口状态信息ISI的定义应符合以下规定：

ISI_0：指示接口与设备识别管理单元的认证状态，1表示未认证，0表示已认证；

ISI_1：指示接口本地加/解密密钥的状态，1表示接口当前不存在数字内容加/解密密钥，0表示接口当前存在数字内容加/解密密钥；

ISI_2：指示接口与其它设备对应接口的认证状态，1表示未认证，0表示已认证。

5.3.2 算法

DICP中用到的不可篡改的算法列于表 3。对于某一个具体实现，根据实际需要选择实现表 3中全部或者部分算法，对于所实现的部分，需要满足保密性和（或）完整性要求。

表3 要求不可篡改的算法

算法	接口	参数说明	算法说明
随机数产生算法	Rand(Len)	Len：需要返回的随机数比特长度	在具体接口规范中规定
签名算法	E_S(SK,M)	SK：私钥 M：待签数据	在具体接口规范中规定
签名验证算法	E_V(PK,M)	PK：公钥 M：待验证数据	默认相应于签名算法的验证算法
标量乘算法	E_M(x,G)	x：乘数 G：基点	默认椭圆曲线上的标量乘算法
流密码算法	S_E(K_E,[IV],Data)	K_E：加密密钥 IV：可选初始向量 Data：待加密数据	在具体接口规范中规定
	S_D(K_E,[IV],Data)	K_E：解密密钥 IV：可选初始向量 Data：待解密数据	在具体接口规范中规定，与上面的加密算法相匹配
分组密码算法	B_E(K_E,[K_MAC],[IV],Data)	K_E：加密密钥 Data：待加密数据 K_MAC：可选完整性密钥 IV：可选初始向量	在具体接口规范中规定
	B_D(K_E,[K_MAC],[IV],Data)	K_E：解密密钥 Data：待解密数据 K_MAC：可选完整性密钥 IV：可选初始向量	在具体接口规范中规定
杂凑算法	Hash(Data)	Data：待杂凑数据	在具体接口规范中规定
HMAC 算法	HMAC(Key,Data)	Key：杂凑密钥 Data：待杂凑数据	在具体接口规范中规定
密钥生成算法	K_G(…)	参数见调用该算法的上下文	在具体接口规范中规定

注1：HMAC杂凑算法应符合RFC 2104的规定，使用的杂凑函数为SHA-256杂凑函数，使用过程中输入密钥的长度大于或等于256、小于512 bit。

注2：在DICP中采用的随机数产生算法，必须能够对各种已知的密码学分析具有足够的抵抗能力。附录A给出了一种伪随机数生成器及伪随机数生成方法。在具体接口的DICP实施规范中对所采用随机数产生算法的进行具体说明。

注3：流密码算法在具体接口规范中定义。

注4：本部分未定义具体密钥生成算法，具体算法在具体接口规范中定义。

5.3.3 协议

5.3.3.1 各种协议及其触发条件

DICP中需要执行的协议包括：协商协议、认证协议、系统完整性更新协议、信息收集协议、密钥激活协议及安全传输协议。

所述协议的触发条件规定如下：

a) 协商协议

当DICP接口间发起认证时，首先执行协商协议。

注：协商协议作为完整认证过程的第一个流程执行。

b) 认证协议

协商完成后触发，在单播传输中由DICP发送端发起认证协议，在广播传输中由DICP接收端发起认证协议。

c) 系统完整性更新协议

具体协议执行触发条件见8.2.1和8.2.2。

d) 信息收集协议

认证完成后或系统完整性更新完成后触发。

e) 密钥激活协议

单播认证完成后或在单播信息收集完成后触发单播信道密钥激活协议；广播方式中在广播会话密钥激活完成后触发广播信道密钥激活协议。

f) 安全传输协议

密钥激活协议完成后，进入安全传输过程。

5.3.3.2 协议消息封装格式

5.3.3.1所列出的协议a)～e)，其协议消息作为协议单元载荷发送。DICP规定协议单元载荷的通用封装格式如下：

DICP标志||协议标志||协议消息号||协议消息

DICP规定安全传输协议中所涉及的消息格式如下：

DICP标志||安全传输标志||安全传输消息

协议单元载荷及安全传输协议消息各段的含义及填充方法应符合以下规定：

a) DICP标志：区别与其它内容保护体系的特定标识，用“UDICP”的ASCII编码填充；

b) 协议标志：DICP中特定协议的标识，1个字节；

c) 协议消息号：封装消息在所属协议中的次序，1个字节；

d) 协议消息：实际发送的协议消息内容；

e) 安全传输标志：安全传输协议标识，1个字节；

f) 安全传输消息：实际发送的与安全传输有关的消息内容。

将本部分应用于具体接口时，具体接口的规范可以根据接口的传输速率、带宽等对协议消息的封装格式重新编码，重新编码后的封装格式应该能够表达编码前的所有信息。

5.4 DICP 内容保护基本流程

本条以设备A和设备B通过DICP接口A_2、B_1传输数字内容为例，说明DICP内容保护的基本流程。

图3展示了DICP内容保护的基本流程。图3中，数字内容从设备A的S单元出发，流经设备A的识别管理单元、设备A的接口A_2、设备B的接口B_1、设备B的识别管理单元，到达设备B的S单元。

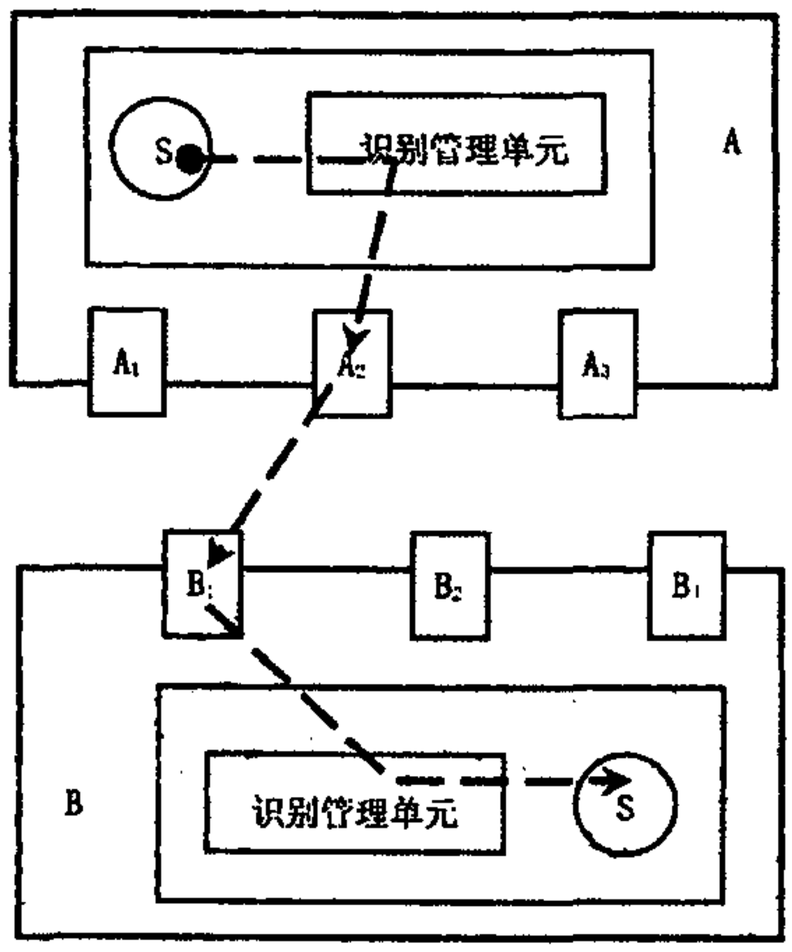


图3　DICP 内容保护基本流程

以下分别描述内容传输中各个功能实体的操作流程（各个操作按编号顺序执行）：

a)　设备A中识别管理单元操作流程：

1)　从S单元提取需要发送给设备B的数字内容；

2)　检测提取的数字内容是否需要保护：如果需要保护，则执行第3)步；否则，把提取的数字内容直接发送到接口A_2，a)模块操作完成；

3)　检测接口A_2是否通过认证：如果已通过认证，则把加密指示和提取的数字内容发送到接口A_2，a)模块操作完成；否则，执行第4)步；

4)　发起与接口A_2的双向认证过程：如果认证成功，则把加密指示和提取的数字内容发送到接口A_2；否则，禁止通过接口A_2向外发送受保护的数字内容。

b)　设备A中接口A_2操作流程：

1)　检测接收到的数据中有无加密指示：如果有加密指示，则执行第2)步；否则，直接把数字内容发送到接口B_1，b)模块操作完成；

2)　查询本地是否具有内容加密密钥：如果有加密密钥，则启用安全传输过程，把数字内容用加密密钥加密后传输到接口B_1，b)模块操作完成；否则，执行第3)步；

3)　发起与接口B_1的认证过程：如果认证成功，则启用安全传输过程，把数字内容用加密密钥加密后传输到接口B_1；否则，禁止向接口B_1发送数字内容，并发送错误指示给识别管理单元。

c)　设备B中接口B_1操作流程：

1)　检测是否启用安全传输接收数字内容：如果启用了安全传输，则执行第2)步；否则，直接把接收到的数字内容发送给识别管理单元，c)模块操作完成；

2)　检测是否完成了对识别管理单元的认证：如果已通过认证，则执行第4)步；否则，执行第3)步；

3)　发起与识别管理单元的认证过程：如果认证成功，则执行第4)步；否则，禁止向识别管理单元发送受保护的数字内容，c)模块操作完成；

4)　查询本地是否具有内容解密密钥：如果有解密密钥，则把数字内容用解密密钥解密后发送给识别管理单元，c)模块操作完成；否则，执行第5)步；

5) 发起与接口A_2的认证过程：如果认证成功，则把数字内容用获得的解密密钥解密后发送给识别管理单元；否则，抛弃接收的内容，并向A_2接口报错，指示认证失败。

d) 设备B中识别管理单元操作流程：

1) 处理接收内容的内容保护标志；

2) 接收数字内容，并发送到S单元中。

注1：内容以明文形式存在设备A的S单元，S单元表示数字内容的起点和终点。无论传输内容是从设备内的存储体内读出，还是通过另一个数据接口输入的实时的节目流，在输出设备A中S单元都表示传输内容的来源和传输的起点。

注2：传输中加解密密钥，在单播中为认证后双方共同产生的密钥，在广播中为发送方本地产生的密钥。

6 认证机制

认证机制包括：协商流程、认证过程、信息收集和密钥激活四个过程。

6.1 协商流程

认证过程以协商流程开始，协商本次传输使用的认证方式、传播方式以及加/解密算法。

当满足5.3.3.1a)所述条件时，执行如下协商流程：

a) 如果协商流程由内容发送方接口（以下简称源接口）触发（见5.4中b）3）步），则源接口向内容接收方接口（以下简称目的接口）发送消息Mes1_Consult，内容如下：

DICP标志||协商标志||第1轮消息标志

b) 如果协商流程由目的接口触发（见5.4中c）5）步），或者目的接口接收到源接口发送的协商请求消息Mes1_Consult，则目的接口发送消息Mes2_Consult，内容如下：

DICP标志||协商标志||第2轮消息标志||接口能力列表

其中，接口能力列表包含以下内容：

认证方式||传播方式||所能支持的加密算法

接口能力列表字段说明：

认证方式：单向认证和/或双向认证；

传播方式：单播和/或广播；

加密算法：流密码算法和/或AES-CCM算法。

c) 源接口接收到目的接口的协商消息Mes2_Consult后，根据Mes2_Consult中的目的接口能力列表、本地能力信息以及已确定的传播方式，根据以下规则生成协商结果：

1) 认证方式：

- 双方均支持双向认证，则执行双向认证；
- 如果源设备不支持双向认证，仅支持单向认证，而目的设备支持单向认证，则执行单向认证；
- 如果目的设备不支持双向认证，仅支持单向认证，而源设备仅支持双向认证，则终止协商过程。

2) 传播方式：

DICP中一台DICP设备在一次传输中只允许使用一种传播方式，传播方式自上而下传递。

传播方式的协商分两种情况：源接口传播方式已经确定和源接口传播方式未确定。

源接口传播方式已经确定时，协商规则如下：

- 源接口使用单播时，如果目的接口支持单播，则执行单播，否则终止协商过程；
- 源接口使用广播时，如果目的接口支持广播，则执行广播，否则终止协商过程。

源接口传播方式未确定时，协商规则如下：

- 一方仅支持单播时，如果另一方支持单播，则执行单播，否则终止协商过程；

- --方仅支持广播时，如果另一方支持广播，则执行广播，否则终止协商过程；
- 如果双方都支持广播和单播，则终止协商过程。

注1：本部分暂不考虑双方都支持广播和单播的情况。

注2：源接口传播方式未确定时协商得到的传播方式同时作为该DICP设备在此次传输中使用的传播方式。

注3：源接口必须能支持该DICP设备在此次传输中确立的传播方式，并将该设备传播方式确立为该源接口此次使用的传播方式，否则终止协商过程。

3) 加密算法：

加密算法以及其优先级在具体的应用标准中定义，如果接口支持多种算法，并且没有规定优先级，则按照以下的原则确定加密算法：

- 如果传播内容为音视频数据，双方优先选用流密码算法，如果有一方不支持流密码算法，考虑使用AES-CCM算法；
- 如果传播内容为除音视频数据以外的数据，双方优先选用AES-CCM算法，如果有一方不支持AES-CCM算法，则终止协商；
- 如果双方没有同时支持的算法，则终止协商。

d) 源接口生成协商结果后，发送消息Mes3_Consult给目的接口，内容如下：

DICP标志||协商标志||第3轮消息标志||协商结果

注：消息Mes3_Consult中的协商结果包括6.1c)中的1)～3)四项协商结果。

e) 双方保存协商结果，结束协商流程。

6.2 认证过程

协商流程结束后即触发认证机制，认证机制是认证过程的核心。本条规定了认证机制所依赖的公钥证书体系及认证机制的实现。

6.2.1 公钥证书体系

6.2.1.1 信任模型

DICP中采用的公钥证书体系信任模型如图4所示。该信任模型具有唯一的根CA（即DICP根CA），使用二级CA的PKI结构，由根CA给每一个二级CA签发证书，再由相应的二级CA签发DICP设备证书或DICP接口证书。

信任模型中二级CA签发的证书可以是识别管理单元所持有的设备证书，也可以是DICP接口所持有的接口证书，统一用Cert_X表示，通常可以由上下文区分，容易引起混淆时本部分会明确指出是设备证书还是接口证书。

DICP设备的识别管理单元中存放有设备证书Cert_X、对应签发设备证书的二级CA证书Cert_X_Adm和DICP根CA证书Cert_Root，这些证书构成如下证书列表：

Cert_X（设备证书）||Cert_X_Adm（二级CA证书）||Cert_Root（DICP根CA证书）

DICP设备的DICP接口中存放有接口证书Cert_X、对应签发接口证书的二级CA证书Cert_X_Adm和DICP根CA证书Cert_Root，这些证书构成如下证书列表：

Cert_X（接口证书）||Cert_X_Adm（二级CA证书）||Cert_Root（DICP根CA证书）

识别管理单元或者DICP接口可以使用自身存储的Cert_Root来验证所接收到的证书列表“Cert_X||Cert_X_Adm”中证书的签名，从而确定其中公钥证书的可信任性，进而可使用Cert_X的公钥来验证识别管理单元或者接口的签名，以完成实体认证。识别管理单元或者DICP接口存储的根证书Cert_Root也可用来验证接收到的CRL中包含的签名。

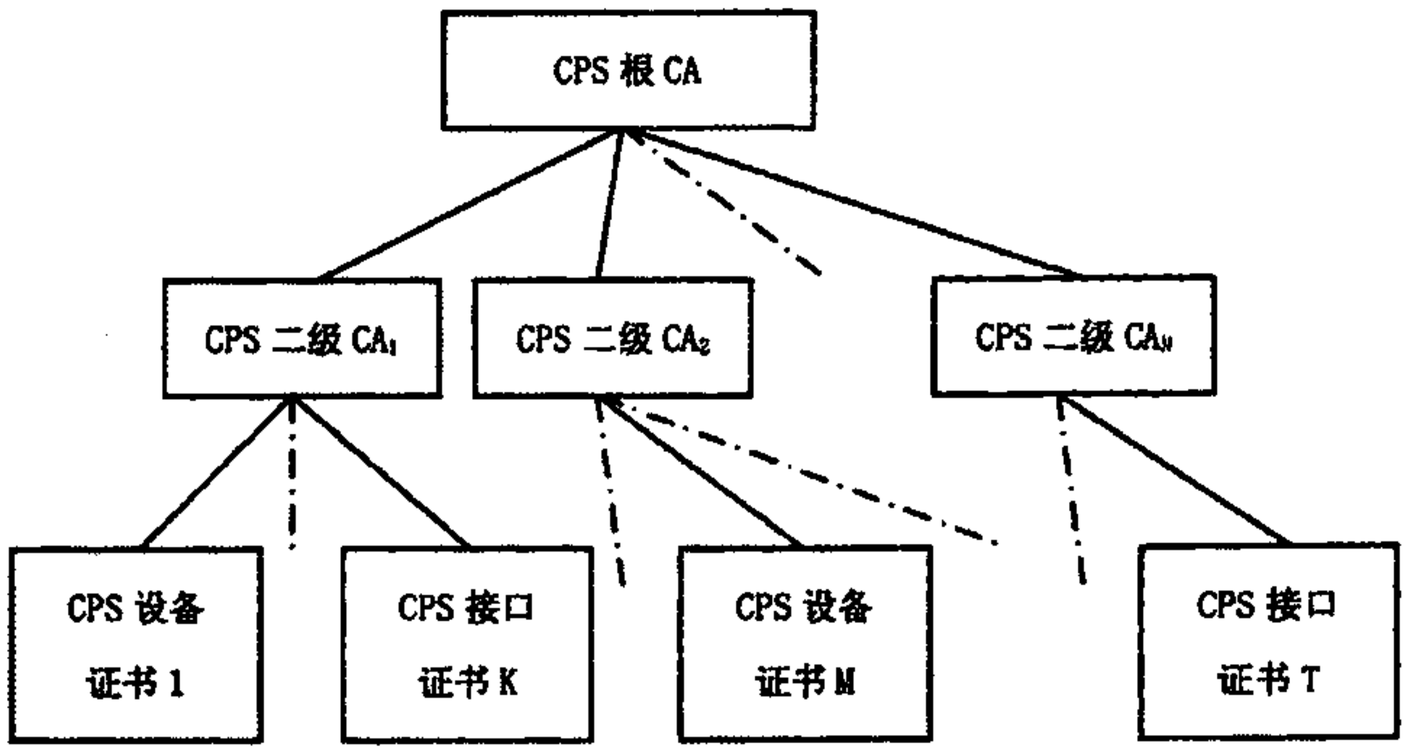


图4 DICP 中使用的 PKI 信任模型

6.2.1.2 证书产生和发放

DICP根CA根据需要设立相应的二级CA，给二级CA发放CA证书，并在其中的“持有者功能标识”域中赋予相应权限，负责特定的业务。

DICP规定：如果二级CA证书中“持有者功能标识”域里LSB0==1，则该二级CA有权签发设备证书；如果二级CA证书中“持有者功能标识”域里LSB1==1，则该二级CA有权签发接口证书。

DICP设备证书和接口证书ID分配表如表4所示。二级CA将其所产生的大量设备证书和接口证书以及相对应的私钥经由安全途径交给硬件生产商，然后由后者将其植入设备中。建议各二级CA在成批签发设备证书或者接口证书时，将所签发证书中的“持有者ID”（54 bit）作为流水号连续发放，以便将来吊销。

表4 DICP 设备证书和接口证书 ID 分配表

ID 值范围	用途
0x000000000~0x07FFFFFF	可用于接口证书和设备证书，仅限于实验测试用
0x080000000~0x0FFFFFFFFFFFF	用于设备证书，产业化应用
0x10000000000000~0x3FFFFFFFFFFFFF	用于接口证书，产业化应用

6.2.1.3 CA 证书格式

DICP体系中CA证书包括DICP根CA证书和二级CA证书，它们具有相同的如下格式：

DICP标识（8 bit）||证书类型（4 bit）||证书版本号（4 bit）||保留（10 bit）||持有者ID（22 bit）||持有者公钥（384 bit）||持有者功能标识（10 bit）||证书签发者标识（22 bit）||签发日期（16 bit）||签名（384 bit）

证书10个字段共计864 bit（108字节），各个字段含义规定如下：

a) DICP标识：标识此证书是哪一个公钥证书体系中的证书；

b) 证书类型：证书的类型，在CA所持证书中此字段值为0，在DICP设备证书中此字段值为1，在DICP接口证书中此字段值为2，其余值保留；

c) 证书版本号：标识证书的版本，现在仅使用版本号0（表示第1版）；

d) 保留：全部置零，留作以后使用；

e) 持有者ID：持有者在DICP中的唯一标识，同时作为本证书的唯一标识符；

f) 持有者公钥：持有该证书的CA的公钥；

g) 持有者功能标识：指明证书中公钥的用途。相应比特位含义如下：

1) LSB0==1：证书中的公钥可以用于验证设备证书；
2) LSB1==1：证书中的公钥可以用于验证接口证书；
3) LSB2==1：证书中的公钥可以用于验证CRL；
4) LSB3==1：证书中的公钥可以用于验证二级CA证书；
5) 其余比特位暂时保留，全部置零。

h) 证书签发者标识：DICP根CA的唯一标识；
i) 签发日期：签发年（7 bit）月（4 bit）日（5 bit）；
j) 签名：证书签发者对证书以上所有信息的有效签名。

注1：DICP体系中的公钥证书标识符为“0x5f”。

注2：在根证书中，证书持有者和证书签发者同为DICP根CA。

注3：签发日期中年份的7 bit数转化为十进制若为y，则实际表示年份为公元(2000+y)年。例如“0000101||0100||11110”表示签发日期为2005年4月30日。

6.2.1.4 设备证书和接口证书格式

DICP设备证书和DICP接口证书具有相同的如下格式：

DICP标识（8 bit）||证书类型（4 bit）||证书版本号（4 bit）||保留（10 bit）||持有者ID（54 bit）||持有者公钥（384 bit）||持有者功能标识（10 bit）||证书签发者标识（22 bit）||签发日期（16 bit）||签名（384 bit）

证书10个字段共计896 bit（112字节），各个字段含义规定如下：

a) DICP 标识：标识此证书是哪一个公钥证书体系中的证书；
b) 证书类型：证书的类型，在 CA 所持证书中此字段值为 0，在 DICP 设备证书中此字段值为 1，在 DICP 接口证书中此字段值为 2，其余值保留；
c) 证书版本号：标识证书的版本，现在仅使用版本号 0（表示第 1 版）；
d) 保留：全部置零，留作以后使用；
e) 持有者 ID：设备证书中为持有该证书的设备的唯一标识符，即设备 ID；接口证书中为持有该证书的接口的唯一标识符，即接口 ID；
f) 持有者公钥：设备证书中为设备证书的公钥；接口证书中为接口证书的公钥；
g) 持有者功能标识：证书持有者功能标识。全部置零，留作以后使用；
h) 证书签发者标识：签发证书的 CA 标识符；
i) 签发日期：即签发年（7 bit）月（4 bit）日（5 bit）；
j) 签名：证书签发者对证书以上所有信息的有效签名。

注：签发日期的补充等同6.2.1.3中的注3。

6.2.2 认证实现

6.2.2.1 认证方式

DICP中规定了可以使用的两种认证方式——双向认证和单向认证。

双向认证中，参与通信的双方应互相鉴别对方身份的合法性，并在鉴别通过后在双方之间建立新的共享密钥。

单向认证中，只需通信一方鉴别通信另一方身份的合法性，不必做相反方向的身份合法性验证，并在鉴别通过后在双方之间建立新的共享密钥。

6.2.2.2 认证实现流程

认证实现所采取的技术包括公钥证书体系、质询-响应机制和椭圆曲线上的DH密钥交换机制等。

6.2.2.2.1 双向认证

假定认证的两个DICP接口模块分别为接口A和接口B，接口A为认证发起方，接口B为认证响应方。图5是DICP双向认证发起方A状态图，图6是DICP双向认证响应方B状态图。

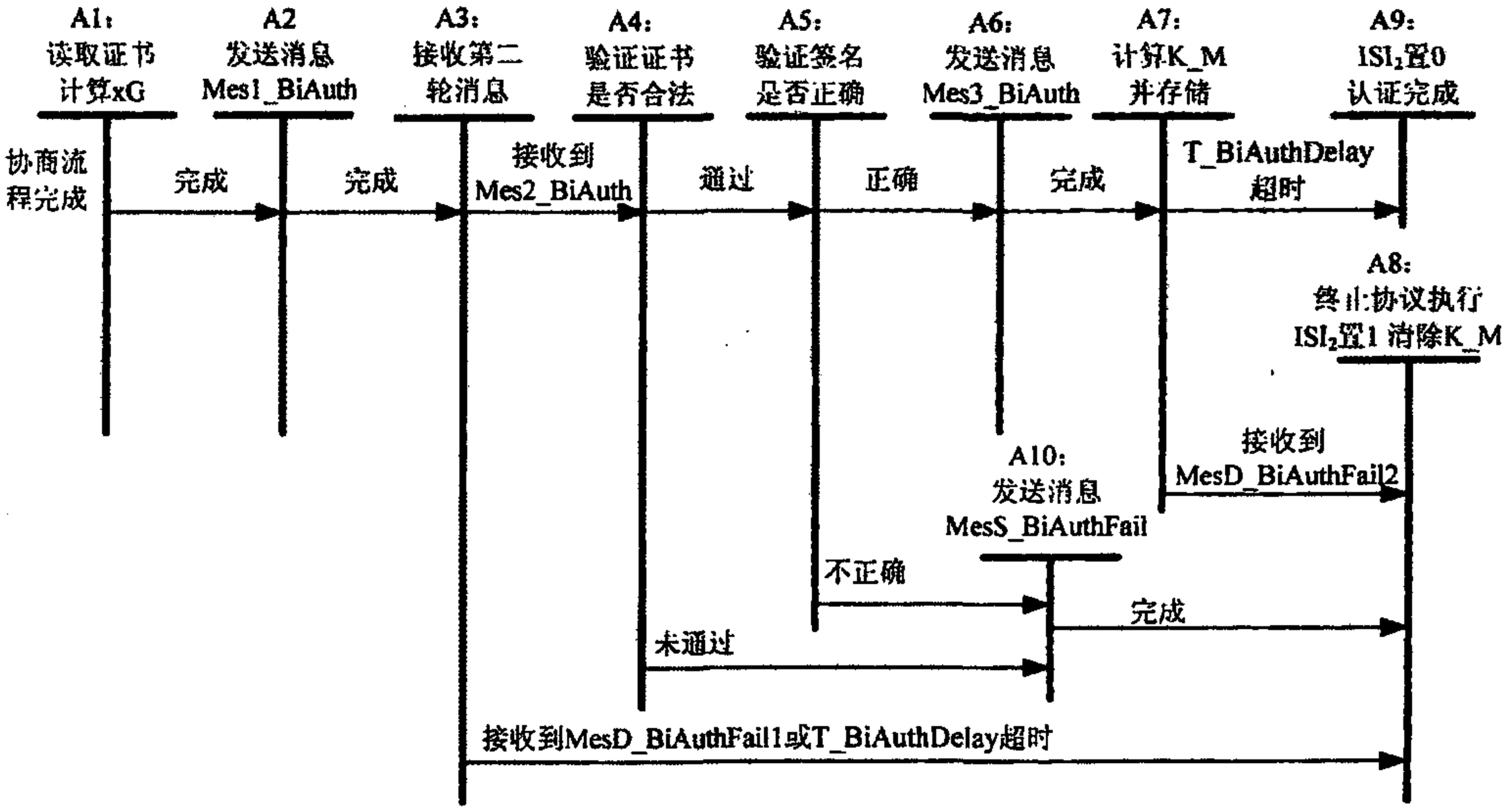


图5　DICP 双向认证发起方状态图

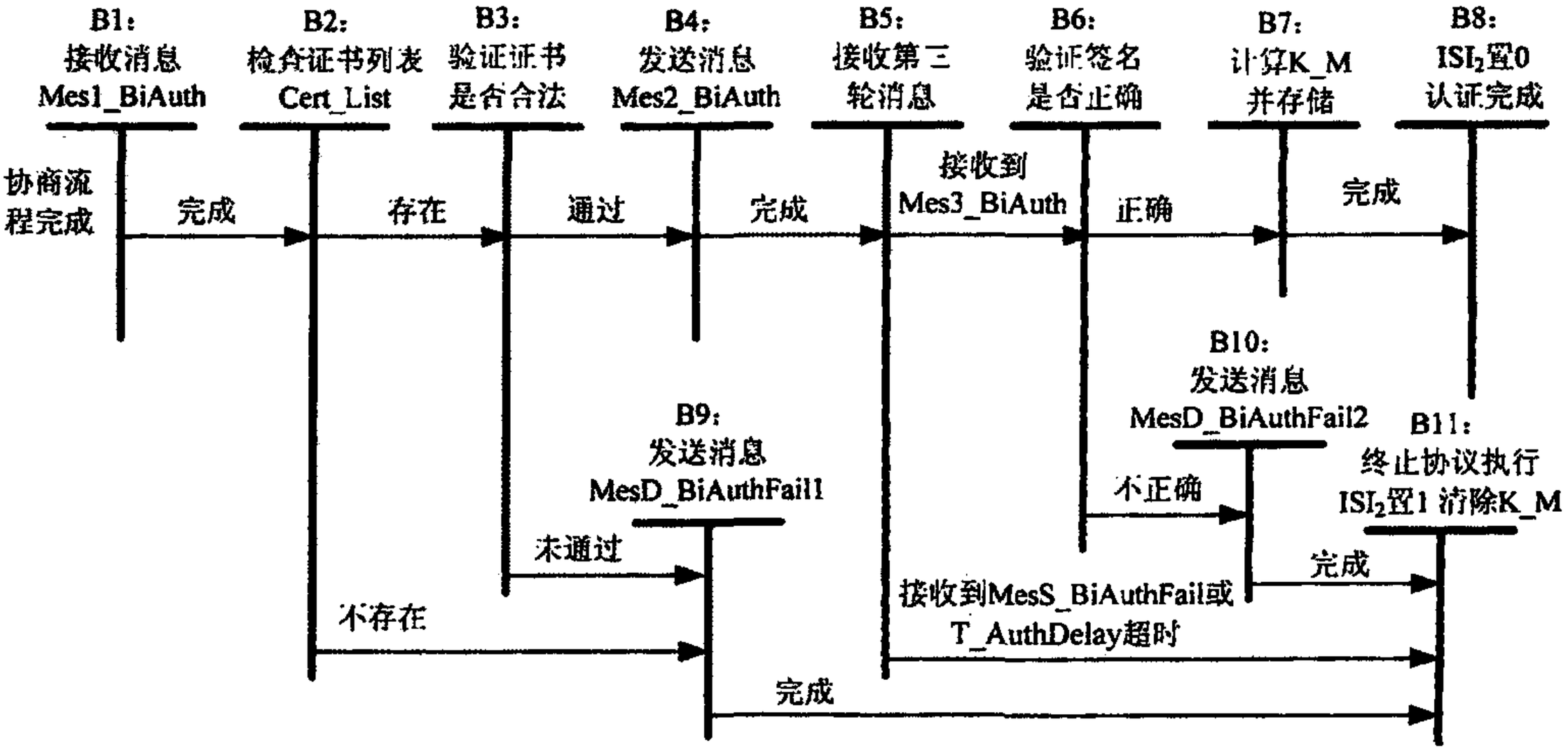


图6　DICP 双向认证响应方状态图

DICP双向认证实现流程如下：

a)　发起方A在双向认证过程中按顺序执行以下流程：

1)　读取存储在本地的接口证书Cert_Source和接口证书的上级CA证书Cert_Source_Adm，级联形成Cert_List，并计算xG。

其中：

- Cert_List：证书列表，组合方式如下：

$$Cert\_Source||Cert\_Source\_Adm$$

- xG：椭圆曲线上定义的标量乘，由E_M(x,G)算法输出。

注：x=Rand（192），是A接口模块调用随机数生成算法在本次DH交换中生成的192 bit随机数。

2) 发送消息Mes1_BiAuth给B。

Mes1_BiAuth消息内容如下：

DICP标志||认证标志||第1轮消息标志||协议消息

协议消息包括以下内容：

Cert_List||xG

3) 消息Mes1_BiAuth发送完成后，设立等待时间T_BiAuthDelay，等待接收第二轮消息。若在T_BiAuthDelay时间内收到B发来的消息Mes2_BiAuth，则执行下一步；若收到MesD_BiAuthFail1或计时超过T_BiAuthDelay，则执行第8)步。

4) 验证Mes2_BiAuth消息中证书是否合法。

证书验证分两个步骤：

- 第一步：检查B证书Cert_Des的ID是否在A存储的CRLA中：若在，则执行第9)步，否则，执行本节a)4)第二步；
- 第二步：使用A存储的根证书公钥验证二级CA证书Cert_Des_Adm，验证通过后使用Cert_Des_Adm公钥验证B接口证书Cert_Des：若两个验证都顺利通过，则执行下一步；否则，执行第10)步。

5) 验证Mes2_BiAuth消息中签名是否正确：若验证正确，则执行下一步；否则，执行第10)步。

验证签名过程中需要进行的计算如下：

- 使用E_M(x, yG)计算xyG；
- 计算K0；
- 使用杂凑算法HMAC()计算HMAC(K0, ID_Source||ID_Des)；
- 使用E_V(PK_Des, 计算得到的HMAC杂凑值||接收的签名值)验证签名。

注1：密钥K_0计算如下：

K_0=K_G($xyG_{left\ 384}$, "00"||"destination hmac key expansion")

其中，$xyG_{left\ 384}$表示取数值xyG的全部384 bit，K_0长度256 bit。

注2：PK_Des是从Cert_Des中获取的B接口公钥。

6) A发送消息Mes3_BiAuth给B。

Mes3_BiAuth消息内容如下：

DICP标志||认证标志||第3轮消息标志||协议消息

协议消息包括内容：

E_S(SK_Source, HMAC(K_1, ID_Des))

协议消息内容说明：

- E_S()：签名算法，参数SK_Source是A接口私钥，参数HMAC(K1, ID_Des)是A接口计算的HMAC杂凑值；
- HMAC()：HMAC杂凑算法，参数K1为密钥，参数ID_Des从Cert_Des中获得。

注：密钥K_1计算如下：

K_1=K_G($xyG_{left\ 384}$"11"||"Source hmac key expansion")

其中，K_1长度256 bit。

7) 消息Mes3_BiAuth发送完成后，A执行如下操作：

- 计算256比特主密钥K_M，如果协商结果为单播，还需将密钥计数器Key_Counter清零，存储<K_M, ID_Des, Key_Counter>；如果为广播，只需储存<K_M, ID_Des>；
- 同时，A设立计时器，若在T_BiAuthDelay时间内接收到消息MesD_BiAuthFail2，则执行第8)步，否则，执行第9)步。

主密钥K_M计算如下：

$$K\_M=K\_G(xyG, ID\_Source||xG||ID\_Des||yG)$$

8) A清除存储的<K_M，ID_Des>或<K_M，ID_Des，Key_Counter>，置认证标志位$ISI_2=1$，终止协议执行，不再执行后续步骤。

9) 置认证标志位$ISI_2=0$，认证协议执行完成，不再执行后续步骤。

10) A向B发送错误报告消息MesS_BiAuthFail，发送完成后执行第8)步。

MesS_BiAuthFail消息形式如下：

DICP标志||认证标志||第3轮消息标志||认证协议失败标志||ID_Source

b) 响应方B在双向认证过程中按顺序执行以下流程：

1) 等待接收A发送过来的消息Mes1_BiAuth，接收到消息Mes1_BiAuth后顺序执行下一步。

2) 检查消息Mes1_BiAuth中是否包含有Cert_List：若有，则执行下一步，否则，执行第9)步。

3) 验证Mes1_BiAuth消息中证书是否合法。

证书验证分两个步骤：

- 第一步：检查A证书Cert_Source的ID是否在B存储的CRLB中：若在，则执行第9)步，否则，执行本条b)3)第二步；
- 第二步：使用B存储的根证书公钥验证二级CA证书Cert_Source_Adm，验证通过后使用Cert_Source_Adm公钥验证A证书Cert_Source：若两个验证都顺利通过，则执行下一步，否则，执行第9)步。

4) 发送消息Mes2_BiAuth给A。

Mes2_BiAuth消息内容如下：

DICP标志||认证标志||第2轮消息标志||协议消息

协议消息包括内容：

Cert_Des||Cert_Des_Adm||yG||E_S(SK_Des, HMAC(K_0, ID_Source||ID_Des))

协议消息内容说明：

- Cert_Des：B接口证书；
- Cert_Des_Adm：签发Cert_Des的CA证书；
- yG：椭圆曲线上定义的标量乘，由E_M(y,G)算法输出；
- E_S()：签名算法，参数SK_Des是B接口私钥，参数HMAC(K0, ID_Source||ID_Des)是B接口计算的HMAC杂凑值；
- HMAC()：HMAC杂凑算法，参数K0是密钥；参数ID_Source||ID_Des是由ID_Source和ID_Des级联得到的字符串。

注1：y=Rand（192），是B接口模块调用随机数生成算法在本次DH交换中生成的192比特随机数。

注2：ID_Source从Cert_Source中获得，ID_Des从Cert_Des中获得。

注3：密钥K_0计算中，xyG由E_M(y,xG)计算输出，其余过程与本条a)5)注1相同。

注4：签名算法中输入的第二个参数——HMAC杂凑值——按照输入签名体制的数据处理。

5) 消息Mes2_BiAuth发送完成后，B设立等待时间T_BiAuthDelay，等待接收第三轮消息。若在T_BiAuthDelay时间内收到A端发来的消息Mes3_BiAuth，则执行下一步；若收到MesS_BiAuthFail或计时超出T_BiAuthDelay，则执行第11)步。

6) 验证Mes3_BiAuth消息中签名是否正确：若验证正确，则执行下一步；否则，执行第10)步。

验证签名过程中需要进行的计算如下：

- 使用E_M(y,xG)计算xyG；

- 计算K_1；
- 使用杂凑算法HMAC()计算HMAC(K_1, ID_Des)；
- 使用E_V(PK_Source,计算得到的HMAC杂凑值||接收的签名值)验证签名。

注1：密钥K_1计算中，xyG由E_M(y,xG)计算输出，其余过程与本条a)6)注相同。

注2：PK_Source是从Cert_Source中获取的A接口公钥。

7) B计算256 bit主密钥K_M，如果协商结果为单播，还需将密钥计数器Key_Counter清零，存储<K_M，ID_Source，Key_Counter>；如果为广播，只需储存<K_M，ID_Source>。完成后顺序执行下一步。

注：主密钥K_M计算方法与本条a)7)中主密钥计算方法相同。

8) 置认证标志位ISI_2=0，认证协议执行完成，不再执行后续步骤。

9) B向A发送错误报告消息MesD_BiAuthFail1，发送完成后执行第11)步。

MesD_BiAuthFail1消息形式如下：

DICP标志||认证标志||第2轮消息标志||认证协议失败标志||ID_Des

10) B向A发送错误报告消息MesD_BiAuthFail2，发送完成后执行下一步。

MesD_BiAuthFail2消息形式如下：

DICP标志||认证标志||第4轮消息标志||认证协议失败标志||ID_Des

11) B清除存储的<K_M，ID_Source>或<K_M，ID_Source，Key_Counter>，置认证标志位ISI_2=1，终止协议执行。

6.2.2.2.2 单向认证

假定认证的两个DICP接口模块分别为接口C和接口D，接口C为认证发起方，接口D为认证响应方。图7是DICP单向认证发起方C状态图，图8是DICP单向认证响应方D状态图。

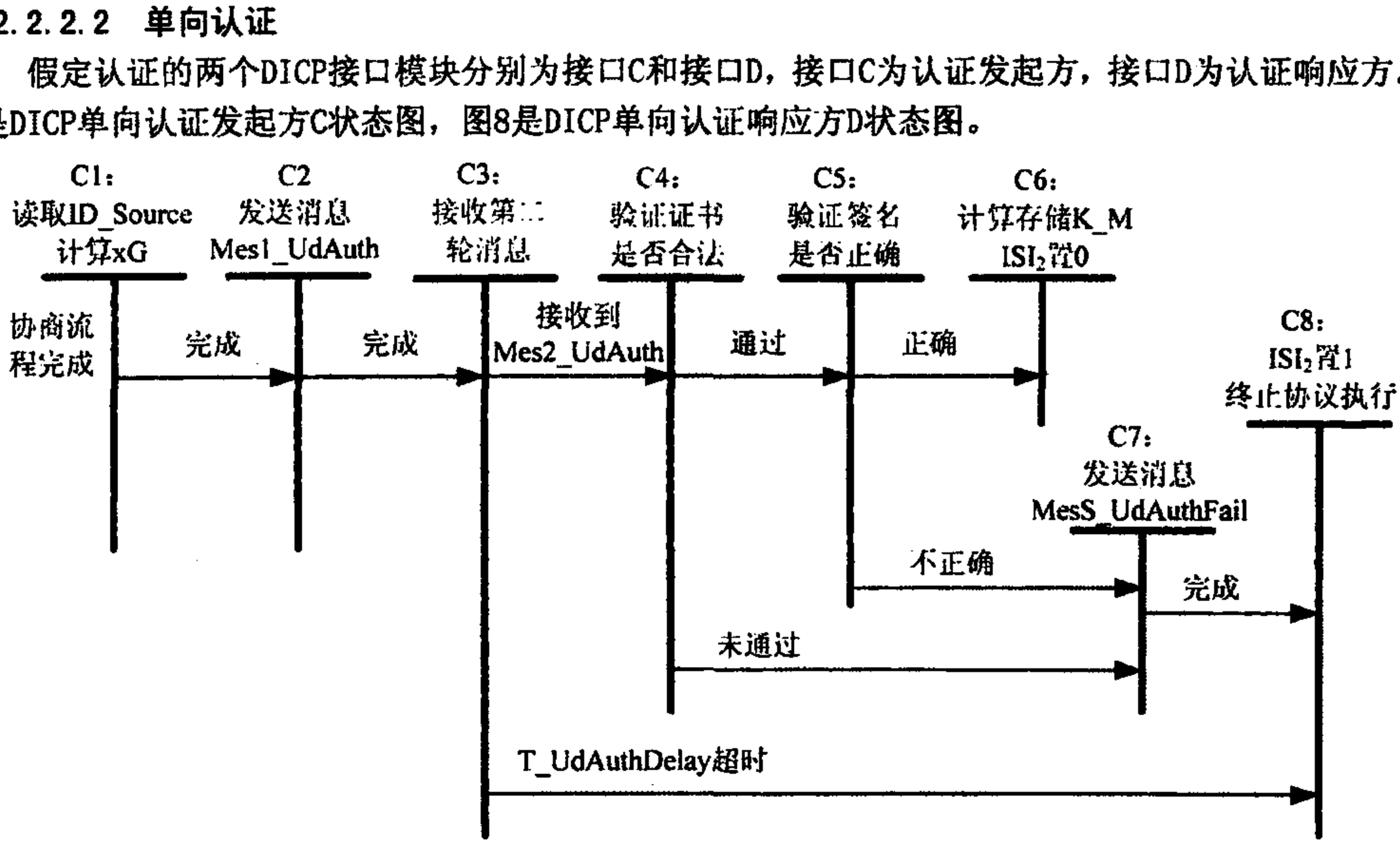


图7 DICP 单向认证发起方状态图

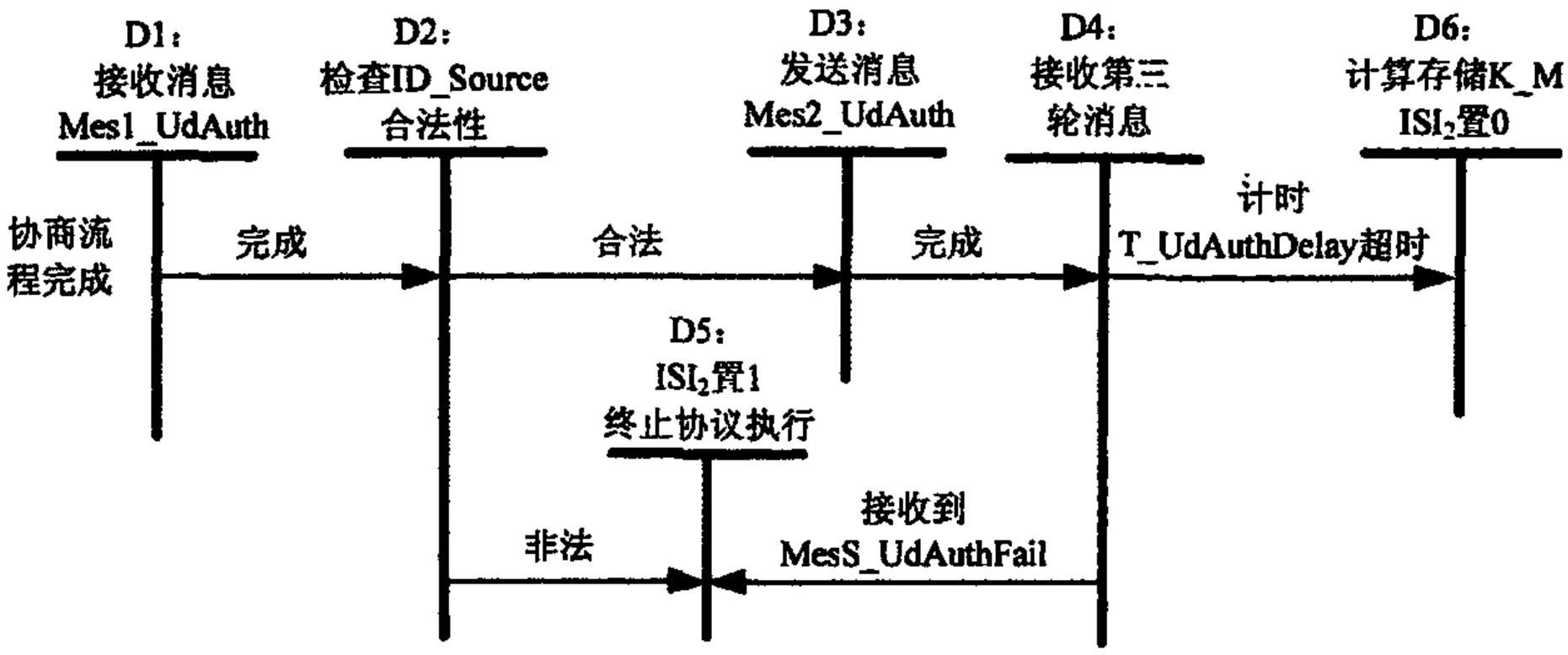


图8 DICP单向认证响应方状态图

DICP单向认证实现流程如下：

a) 发起方C在单向认证过程中按顺序执行以下流程：

1) 读取存储在本地的接口证书Cert_Source中的ID_Source，并计算xG。其中，xG是椭圆曲线上定义的标量乘，由E_M(x,G)算法输出。

注：x=Rand（192），是C接口模块调用随机数生成算法在本次DH交换中生成的192 bit随机数。

2) 发送消息Mes1_UdAuth给D。

Mes1_UdAuth消息内容如下：

DICP标志||认证标志||第1轮消息标志||协议消息

协议消息包括以下内容：

ID_Source||xG

3) 消息Mes1_UdAuth发送完成后，设立等待时间T_UdAuthDelay，等待接收第二轮消息。若在T_UdAuthDelay时间内收到D发来的消息Mes2_UdAuth，则执行下一步；否则，执行第8)步。

4) 验证Mes2_UdAuth消息中证书是否合法。

证书验证分两个步骤：

- 第一步：检查D证书Cert_Des的ID是否在C存储的CRL_C中：若在，则执行第7)步，否则，执行本节a)4)第二步；
- 第二步：使用C存储的根证书公钥验证二级CA证书Cert_Des_Adm，验证通过后使用Cert_Des_Adm公钥验证D接口证书Cert_Des：若两个验证都顺利通过，则执行下一步；否则，执行第7)步。

5) 验证Mes2_UdAuth消息中签名是否正确：若验证正确，则执行下一步；否则，执行第7)步。

验证签名过程中需要进行的计算如下：

- 使用E_M(x,yG)计算xyG；
- 计算K_0；
- 使用杂凑算法HMAC()计算HMAC(K_0, ID_Source||ID_Des)；
- 使用E_V(PK_Des,计算得到的HMAC杂凑值||接收的签名值)验证签名。

注1：密钥K_0计算过程与6.2.2.2.1a)5)注1相同。

注2：PK_Des是从Cert_Des中获取的D接口公钥。

6) 计算256 bit主密钥K_M，如果协商结果为单播，还需将密钥计数器Key_Counter清零，存储<K_M，ID_Des，Key_Counter>；如果为广播，只需储存<K_M，ID_Des>。同时，置认证标志位ISI_2=0，认证协议执行完成，不再执行后续步骤。

主密钥K_M计算如下：

K_M=K_G(xyG, ID_Source||xG||ID_Des||yG)

7) 向D发送错误报告消息MesS_UdAuthFail，发送完成后执行下一步。

MesS_UdAuthFail消息形式如下：

DICP标志||认证标志||第3轮消息标志||认证协议失败标志||ID_Source

8) 置认证标志位ISI_2=1，终止协议执行。

b) 响应方D在单向认证过程中按顺序执行以下流程：

1) 等待接收C发送过来的消息Mes1_UdAuth，接收到消息Mes1_UdAuth后顺序执行下一步。

2) 检查消息Mes1_UdAuth中ID_Source的合法性。检查ID_Source是否在D存储的CRL_0中：若在，则执行第5)步，否则，执行下一步。

3) 发送消息Mes2_UdAuth给C。

Mes2_UdAuth消息内容如下：

DICP标志||认证标志||第2轮消息标志||协议消息

协议消息包括内容：

Cert_Des||Cert_Des_Adm||yG||E_S(SK_Des, HMAC(K_0, ID_Source||ID_Des))

协议消息内容说明：

- Cert_Des：D接口证书；
- Cert_Des_Adm：签发Cert_Des的CA证书；
- yG：椭圆曲线上定义的标量乘，由E_M(y,G)算法输出；
- E_S()：签名算法，参数SK_Des是D接口私钥，参数HMAC(K0, ID_Source||ID_Des)是D接口计算的HMAC杂凑值；
- HMAC()：HMAC杂凑算法，参数K0是密钥；参数ID_Source||ID_Des是由ID_Source和ID_Des级联得到的字符串。

注1：y=Rand（192），是D接口模块调用随机数生成算法在本次DH交换中生成的192 bit随机数。

注2：密钥K_0计算中，xyG由E_M(y,xG)计算输出，其余过程与本条a)5)注1相同。

注3：签名算法中输入的第二个参数——HMAC杂凑值——按照输入签名体制的数据处理。

4) 消息Mes2_BiAuth发送完成后，D设立等待时间T_UdAuthDelay，等待接收第三轮消息。若在T_UdAuthDelay时间内收到C端发来的消息MesS_UdAuthFail，则执行下一步；否则，执行第6)步。

5) 置认证标志位ISI_2=1，终止协议执行，不再执行后续步骤。

6) 计算256 bit主密钥K_M，如果协商结果为单播，还需将密钥计数器Key_Counter清零，存储<K_M，ID_Source，Key_Counter>；如果为广播，只需储存<K_M，ID_Source>。同时置认证标志位ISI_2=0，认证协议执行完成。

注：主密钥K_M计算方法与本条a)6)中主密钥计算方法相同。

6.3 信息收集

信息收集由DICP发送端接口发起，收集该接口下游连接的所有激活的DICP设备的连接拓扑信息及设备信息。连接拓扑信息包括连接数量LC和连接深度LD。单播和广播具有不同的信息收集过程。

6.3.1 单播信息收集

单播信息收集中，由最上层发送端接口发起信息收集，读取与其直接相连的DICP设备识别管理单元中存储的连接信息（指连接数量LC、连接深度LD和连接设备信息LDI），而DICP设备识别管理单元中记录的连接信息则通过汇总本设备所有下游发送端接口（激活状态）存储的连接信息得到。

最上层发送端接口执行完信息收集后，上报本地的识别管理单元，由识别管理单元根据本地策略决定终止此次内容传输或者是继续执行协议。

图9是DICP单播信息收集协议执行过程。

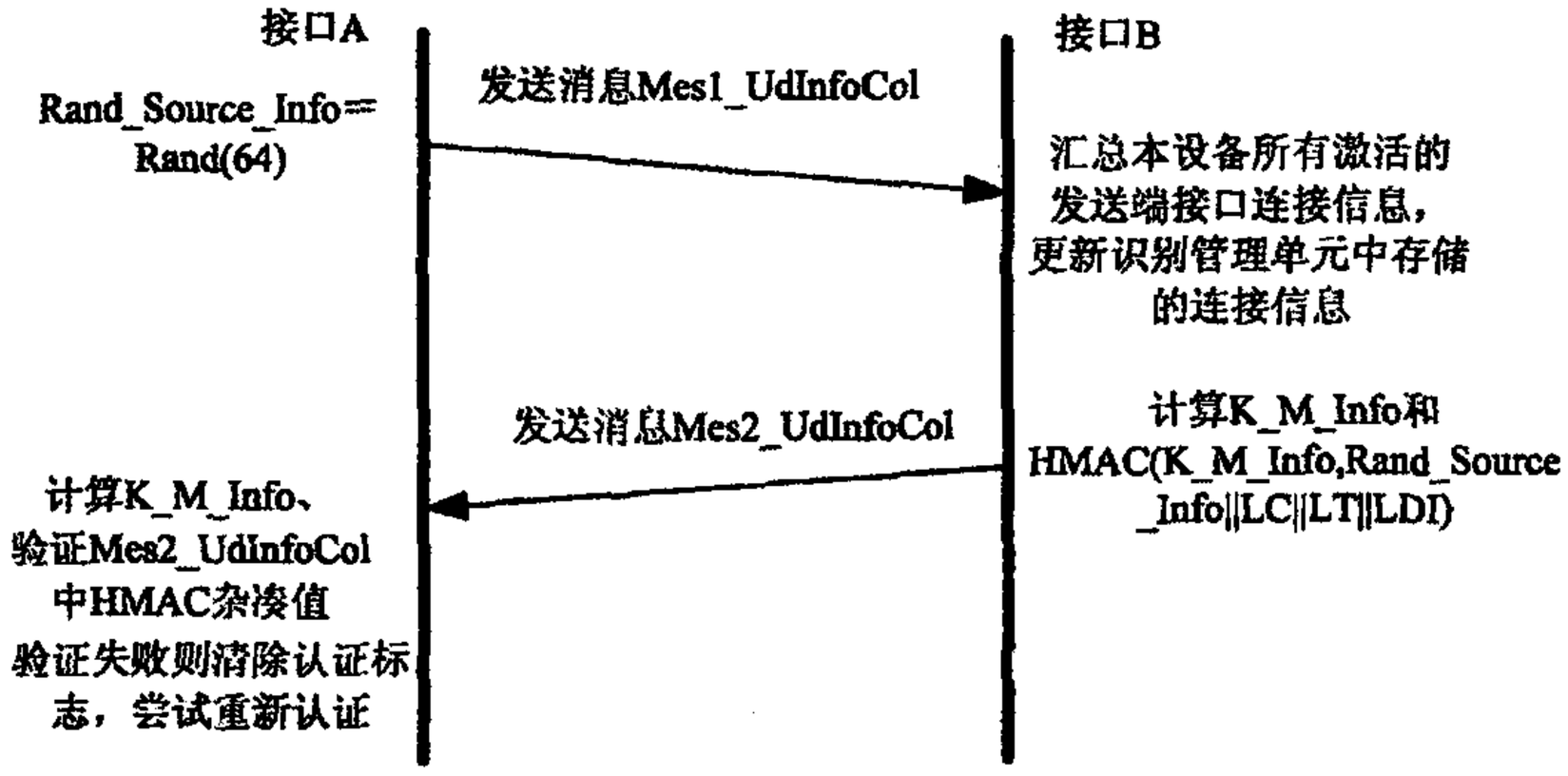


图9 DICP 单播信息收集协议执行过程

假定接口A为信息收集发起方，接口B为信息收集响应方，A、B执行流程如下：

a) 发起方A在单播信息收集过程中按顺序执行以下操作：

1) 发送消息Mes1_UdInfoCol到B。

Mes1_UdInfoCol消息内容如下：

DICP标志||信息收集标志||第1轮消息标志||Rand_Source_Info||ID_Source

其中，Rand_Source_Info由Rand(64)计算得到。

2) 计时等待时间T_UdInfoColDelay，等待B所在设备收集连接信息。如果在T_UdInfoColDelay时间内接收到B发送的消息Mes2_UdInfoCol，则执行下一步；否则，执行第4)步。

3) 验证消息Mes2_UdInfoCol中包含的HMAC杂凑值：如果验证通过，则上报识别管理单元，由识别管理单元更新本地的连接信息，A信息收集协议执行完毕，不再执行后续步骤；如果验证失败，则执行第4)步。

验证过程中需要进行如下计算：

- 计算256 bit的密钥K_M_Info；

  K_M_Info计算公式如下：

  K_M_Info=K_G(K_M,"Used for Information Collection Integrity Validation")

- 计算HMAC(K_M_Info,Rand_Source_Info||LC||LT||LDI)杂凑值。

4) 清除本地的认证状态指示，尝试与B重新执行认证协议。

b) 响应方B在单播信息收集过程中按顺序执行以下操作：

1) B等待接收A发送过来的消息Mes1_UdInfoCol，接收到消息Mes1_UdInfoCol后，执行下一步；

2) 汇总本设备所有激活的发送端接口连接信息，汇总完毕后更新本设备识别管理单元中存储的连接信息；

注：识别管理单元应该在汇总完本设备所有激活的发送端接口连接信息后再允许读取连接信息，连接信息包括连接数量LC、连接深度LT和连接设备信息LDI。

3) 发送消息Mes2_UdInfoCol给A。

Mes2_UdInfoCol消息内容如下：

DICP标志||信息收集标志||第2轮消息标志||协议消息

协议消息包含内容如下：

LC||LT||LDI||HMAC(K_M_Info,Rand_Source_Info||LC||LT||LDI)

注：协议消息中K_M_Info和HMAC杂凑值的计算过程同本条b)3)。

6.3.2 广播信息收集

广播信息收集中，只需收集连接在广播总线上的接收端数量LC。每次认证完成后源接口LC增加1，每次某个广播信道的加密密钥失效后，LC减少n(n为拥有该信道加密密钥的下游设备的数量)。

源接口每次更新LC值后，需要上报识别管理单元，由识别管理单元根据本地策略决定终止内容传输还是继续执行协议。

6.4 密钥激活

密钥激活过程使用认证后建立的主密钥生成加密密钥和完整性校验密钥。根据传播信道的不同分为单播信道密钥激活和广播信道密钥激活。

6.4.1 单播信道密钥激活

假定执行密钥激活协议的两个DICP接口模块分别为接口A和接口B，接口A为密钥激活发起方，接口B为密钥激活响应方。图10是DICP单播信道密钥激活发起方A状态图，图11是DICP单播信道密钥激活响应方B状态图。

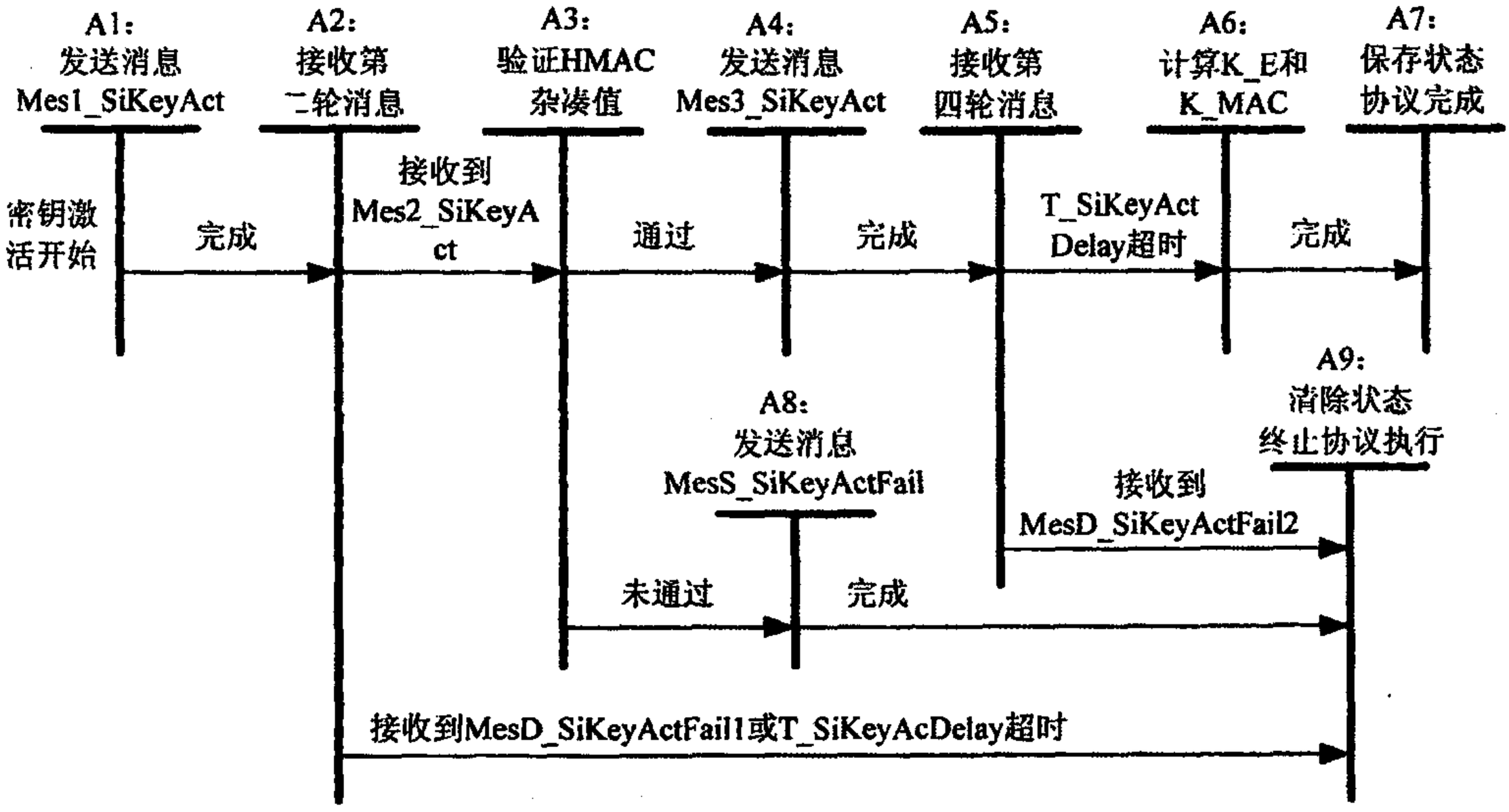


图10 DICP 单播信道密钥激活发起方状态图

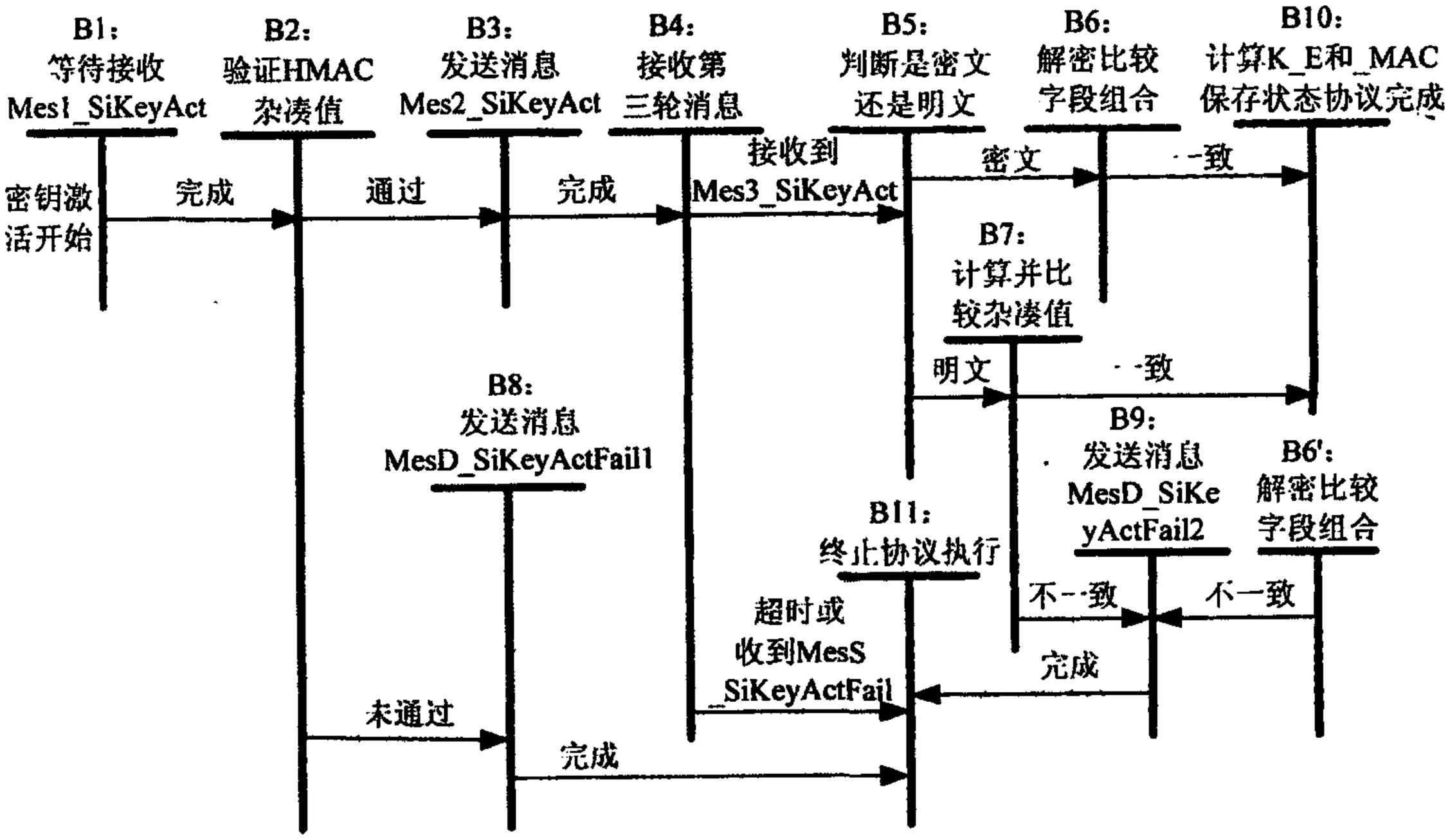


图11　DICP单播信道密钥激活响应方状态图

DICP单播信道密钥激活过程流程如下：

a)　单播信道密钥激活过程中发起方A顺序执行以下流程：

1)　产生随机数Rand_Act _Source=Rand(64)，并向B发送消息Mes1_SiKeyAct。

消息Mes1_SiKeyAct内容如下：

DICP标志||密钥激活标志||第1轮消息标志||协议消息

协议消息包含内容如下：

$$Rand\_Act\_Source, HMAC(K\_M_{lsb0\sim255}, Rand\_Act\_Source||ID\_Source)$$

注：K_M是A和B认证通过后产生的主密钥，双方同时保存。

2)　消息Mes1_SiKeyAct发送完成后，A设立等待时间T_SiKeyActDelay，等待接收第二轮消息。若在T_SiKeyActDelay时间内收到接口B发来的消息Mes2_SiKeyAct，则执行下一步；若接收到MesD_SiKeyActFail1或计时超时，执行第9)步。

3)　验证消息Mes2_SiKeyAct中包含的HMAC杂凑值。根据消息Mes2_SiKeyAct中传递的内容重新计算$HMAC(K_M_{lsb0\sim255}, Rand_Act_Sink||Rand_Act_Source||ID_Source||ID_Des)$，并与接收的到HMAC杂凑值进行比较：若比较结果相同，则执行下一步；否则，执行第8)步。

4)　向B发送消息Mes3_SiKeyAct。

Mes3_SiKeyAct消息内容如下：

DICP标志||密钥激活标志||第3轮消息标志||协议消息

协议消息内容选择如下：

- 如果加密算法的相关参数需要保密，并且接口实现时支持分组加密算法，则使用以下方法构造协议消息：

  $$\text{“Encr”}||E\_S(K\_M_{lsb0\sim127}, Rand\_Act\_Sink||ID\_Des||[Alg\_Para])$$

- 如果加密算法的相关参数不需要保密，或者接口实现时不支持分组加密算法，则使用以下方法构造协议消息：

  $$\text{“HMAC”}||[Alg\_Para], HMAC(K\_M_{lsb0\sim255}, Rand\_Act\_Sink||ID\_Des||[Alg\_Para])$$

以上协议消息内容中，Alg_Para是可选项，传输加密算法各种参数，根据协商的加密算法决定，当且仅当协商的加密算法的参数可以公开时才可以使用上述消息传输。

5) 消息Mes3_SiKeyAct发送完成后，A设立等待时间T_SiKeyActDelay。等待接收第四轮消息。若在T_SiKeyActDelay时间内收到消息MesD_SiKeyActFail2，则执行第9)步；否则，执行下一步。

6) 计算加密密钥K_E(长度128)和完整性密钥K_MAC(长度256)。计算方式如下：

$$K\_E=K\_G(K\_M_{lsb128\sim255}, Rand\_Act\_Sink||Rand\_Act\_Source)$$

$$K\_MAC=K\_G(K\_M_{lsb128\sim255}, Rand\_Act\_Source||Rand\_Act\_Sink)$$

7) 设置本地加密密钥状态ISI_i为0，设置本地加密计数器Cipher_Counter为0，设置D_W_Counter为0，密钥激活协议执行完毕。

8) 向B发送密钥激活失败消息MesS_SiKeyActFail。

消息MesS_SiKeyActFail内容如下：

DICP标志||密钥激活标志||第3轮消息标志||密钥激活失败标志

9) 清除接口A与接口B的状态信息，终止协议执行。

b) 单播信道密钥激活过程中响应方B顺序执行以下流程：

1) 等待接收A发送的消息Mes1_SiKeyAct，接收到消息Mes1_SiKeyAct后顺序执行下一步。

2) 验证消息Mes1_SiKeyAct中包含的HMAC杂凑值，比较接收到的HMAC杂凑值与计算得到的HMAC杂凑值是否相同：若相同，则执行下一步；否则，执行第8)步。

注：HMAC杂凑值计算方法同本条a)1)。

3) 产生新的随机数Rand_Act_Sink=Rand(64)，并向A发送消息Mes2_SiKeyAct。

消息Mes2_SiKeyAct内容如下：

DICP标志||密钥激活标志||第2轮消息标志||协议消息

协议消息包含内容如下：

$Rand\_Act\_Sink, HMAC(K\_M_{lsb0\sim255}, Rand\_Act\_Sink||Rand\_Act\_Source||ID\_Source||ID\_Des)$

4) B设立等待时间T_SiKeyActDelay，等待接收第三轮消息。若在T_SiKeyActDelay时间内收到接口A发来的消息Mes3_SiKeyAct，则执行下一步；若接收到MesS_SiKeyActFail或计时超时，则终止协议执行。

5) 根据消息Mes3_SiKeyAct中包含的协议消息，判断接收的是密文还是明文和杂凑值：如果是密文，则执行下一步；如果是明文和杂凑值则执行第7)步。

6) 解密密文分组，检查Rand_Act_Sink||ID_Des字段与本地合成的该字段的内容是否一致：一致，则执行第10)步；不一致，则执行第9)步。

7) 计算杂凑值，并比较计算值与接收的值是否相同：相同，则执行第10)步；不同，则执行第9)步。

8) 清除接口B与接口A的状态信息，向A发送送密钥激活失败消息MesD_SiKeyActFail1，并终止协议执行。

消息MesD_SiKeyActFail1内容如下：

DICP标志||密钥激活标志||第2轮消息标志||密钥激活失败标志

9) 向A发送密钥激活失败消息MesD_SiKeyActFail2，并终止协议执行。

消息MesD_SiKeyActFail2内容如下：

DICP标志||密钥激活标志||第4轮消息标志||密钥激活失败标志

10) 按照本条a)6)中所示方法计算128 bitK_E和256 bitK_MAC，设置本地加密密钥状态ISI_i为0，设置本地加密计数器Cipher_Counter为0，设置D_W_Counter为0，密钥激活协议执行完毕。

6.4.2 广播信道密钥激活

DICP广播信道密钥激活协议分两部分：广播信道会话密钥激活协议和广播信道加密密钥传输协议。DICP广播信道密钥激活协议由DICP接收端接口发起执行，首先执行广播信道会话密钥激活协议，执行成功后再执行广播信道加密密钥传输协议。

描述过程中假定执行广播密钥激活协议的两个DICP接口模块分别为接口A和接口B，接口A为密钥激活发起方，接口B为密钥激活响应方。

6.4.2.1 广播信道会话密钥激活协议

图12是DICP广播信道会话密钥激活发起方A状态图，图13是DICP广播信道会话密钥激活响应方B状态图。

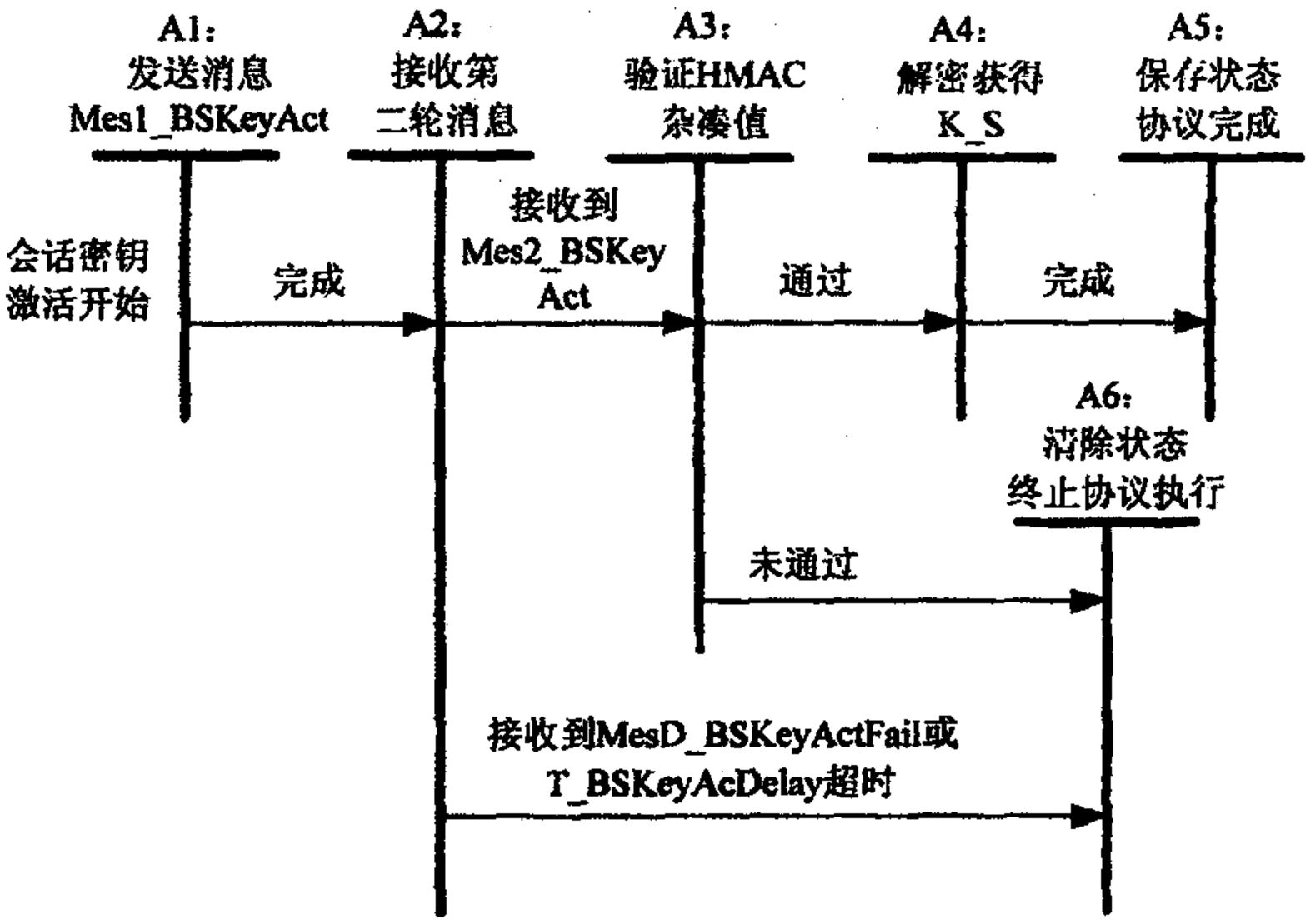


图12 DICP广播信道会话密钥激活发起方状态图

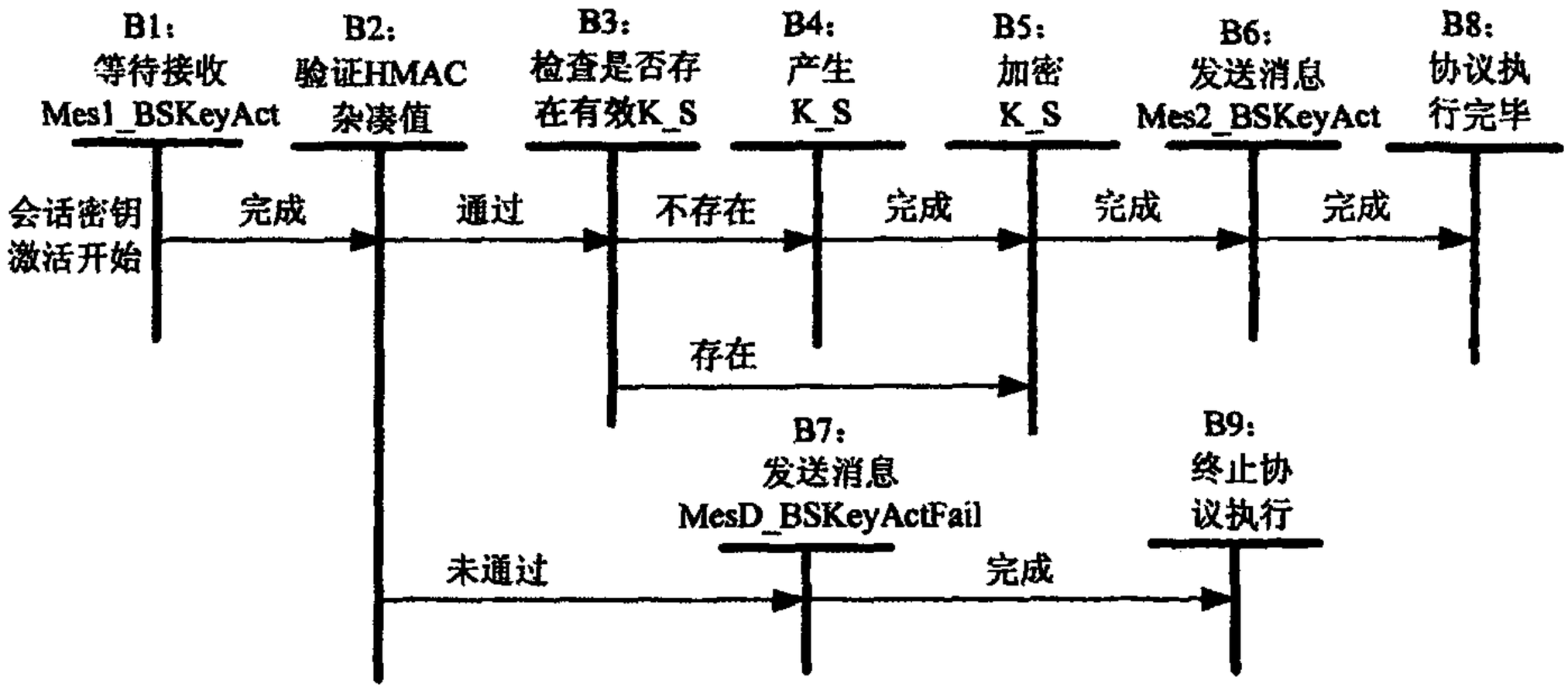


图13 DICP广播信道会话密钥激活响应方状态图

DICP广播信道会话密钥激活过程流程如下：

a) 广播信道会话密钥激活过程中发起方A顺序执行以下流程：

1) 产生随机数Rand_Act_Source=Rand(64)，并向B发送消息Mes1_BSKeyAct。

消息Mes1_BSKeyAct内容如下：

DICP标志||会话密钥激活标志||第1轮消息标志||协议消息

协议消息包含内容如下：

Rand_Act_Source, HMAC($K\_M_{lsb0\sim255}$, Rand_Act_Source||ID_Source)

注：K_M是A和B认证通过后产生的主密钥，双方同时保存。

2) 消息Mes1_BSKeyAct发送完成后，A设立等待时间T_BSKeyActDelay，等待接收第二轮消息。若在T_BSKeyActDelay时间内收到B发来的消息Mes2_BSKeyAct，则执行下一步；若接收到MesD_BSKeyActFail或计时超时，则执行第6)步。

3) 验证消息Mes2_BSKeyAct中包含的HMAC杂凑值。根据消息Mes2_BSKeyAct中传递的内容重新计算HMAC($K\_M_{lsb0\sim255}$, K_S'||ID_Source||ID_Des)，并与接收的到HMAC杂凑值进行比较：若结果相同，则执行下一步；否则，执行第6)步。

4) 解密得到会话密钥K_S(长度128)，解密完成后顺序执行下一步。

K_S计算方式如下：

$$K\_S=B\_D(K\_M_{lsb0\sim255}, K\_S')$$

5) 设置本地会话加密计数器Session_Cipher_Counter为0，会话密钥激活协议执行完毕。

6) 清除A相应状态信息，终止协议执行。

b) 广播信道会话密钥激活过程中响应方B顺序执行以下流程：

1) 等待接收A发送的消息Mes1_BSKeyAct，接收到消息Mes1_BSKeyAct后顺序执行下一步。

2) 验证消息Mes1_BSKeyAct中包含的HMAC杂凑值，比较接收到的HMAC杂凑值与计算得到的HMAC杂凑值是否相同：若相同，则执行下一步；否则，执行第7)步。

注：HMAC杂凑值计算方法同本条a)1)。

3) 检查是否已经存在正在使用的有效的会话密钥K_S，如果存在，则执行第5）步；否则，执行下一步；

4) 生成会话密钥K_S，生成后顺序执行下一步。

$$K\_S=K\_G(K\_M_{lsb128\sim255}, Rand\_Act\_Session)$$

其中，Rand_Act_Session是用于产生会话密钥的随机数，由Rand(128)输出。

5) 加密会话密钥生成K_S'，加密完成后顺序执行下一步。

$$K\_S'=B\_E(K\_M_{lsb128\sim255}, K\_S)$$

6) 向A发送消息Mes2_BSKeyAct，发送完成后会话密钥激活协议执行完毕。

消息Mes2_BSKeyAct内容如下：

DICP标志‖会话密钥激活标志||第2轮消息标志||协议消息

协议消息包含内容如下：

K_S', HMAC($K\_M_{lsb0\sim255}$, K_S'||ID_Source||ID_Des)

7) 向A发送会话密钥激活失败消息MesD_BSKeyActFail，发送完成后终止会话密钥激活协议。

消息MesD_BSKeyActFail内容如下：

DICP标志‖密钥激活标志||第2轮消息标志||会话密钥激活失败标志

6.4.2.2 广播信道加密密钥传输协议

广播信道会话密钥激活协议执行成功后或每隔一定的周期(具体周期在具体的应用接口规范中定义)，执行广播信道加密密钥传输协议。

广播信道加密密钥传输采用广播的方式，由B（发送端接口）直接向A（接收端接口）发送广播信息，A、B两端执行流程如下：

a) 广播信道加密密钥传输过程中B顺序执行以下流程：

1) 产生随机数

Rand_Act_Source1=Rand(128)

Rand_Act_Source2=Rand(128)

2) 计算加密密钥K_E_G(长度128)和完整性密钥K_MAC_G(长度256)。计算方式如下：

$$K\_E=K\_G(K\_S_{lsb128\sim255}, Rand\_Act\_Source1||ID\_channel)$$

$$K\_MAC=K\_G(K\_S_{lsb128\sim255}, Rand\_Act\_Source2||ID\_channel)$$

3) 设置本地加密密钥状态ISI_1为0，设置本地加密计数器Cipher_Counter为0，设置D_W_Counter为0。

4) 向广播信道广播消息消息Mes_BCKey。

Mes_BCKey消息内容如下：

DICP标志||密钥激活标志||广播消息标志||协议消息

协议消息包含内容如下：

$$B\_E(K\_S_{lsb0\sim127}, K\_E||K\_MAC||[Alg\_Para]||Channel\_Cipher\_Counter||Key\_Counter||ID\_Source||ID\_Channel)$$

协议消息内容说明：

- K_E||K_MAC：用于ID_Channel的加密密钥和完整性密钥；
- Alg_Para：可选项，根据协商的加密算法决定，在X-CCM情况下，该字断必选，包含X-CCM算法所需要的初始向量信息；
- Channel_Cipher_Counter：B当前加密计数器的值；
- Key_Counter：当前密钥计数器的值。

注：如果接口不包括任何分组加密算法，则由接口规定具体算法来传输加密算法所需的初始向量信息，该信息应该保密。

b) 广播信道加密密钥传输过程中A顺序执行以下流程：

1) 等待接收B发送的消息Mes_BCKey，接收到消息后顺序执行下一步；

2) 解密密文分组，检查ID_Source||ID_Channel字段与本地合成的该字段的内容是否一致：一致，则执行下一步；不一致，则执行第4)步；

3) 设置D_W_Counter为0，保存ID_Channel的加密密钥K_E和完整性密钥K_MAC，设置本地加密密钥状态ISI1为0，密钥激活协议执行完毕；

4) 重新向B发送会话密钥激活请求消息。

6.4.3 奇偶密钥

密钥激活协议中，会话密钥K_S、加密密钥K_E和完整性密钥K_MAC分别存储在K_S[0]、K_E[0]和K_MAC[0]中或者K_S[1]、K_E[1]和K_MAC[1]中，把存储在K_S[0]、K_E[0]和K_MAC[0]区的密钥称为偶密钥，存储在K_S[1]、K_E[1]和K_MAC[1]区的密钥称为奇密钥，具体存储位置由当前$[Session_Key_Counter]_{lsb0}$、$[Key_Counter]_{lsb0}$或者$[Channel_Key_Counter]_{lsb0}$的值来决定。

对于组密钥：

a) 当$[Session_Key_Counter]_{lsb0}$值为0时，偶密钥为当前工作密钥，奇密钥为备份密钥，供下一次密钥更新时使用；

b) 当$[Group_Key_Counter]_{lsb0}$值为1时，奇密钥为当前工作密钥，偶密钥为备份密钥，供下一次密钥更新时使用。

对于加密密钥、完整性密钥：

a) 当$[Key_Counter]_{lsb0}$或者$[Channel_Key_Counter]_{lsb0}$的值为0时，偶密钥为当前工作密钥，奇密钥为备份密钥，供下一次密钥更新时使用；

b) 当$[Key_Counter]_{lsb0}$或者$[Channel_Key_Counter]_{lsb0}$的值为1时，奇密钥为当前工作密钥，偶密钥为备份密钥，供下一次密钥更新时使用。

7 安全传输

传输双方接口在检测到本地ISI_0、ISI_1和ISI_2同时为0时，根据协商的加密算法执行安全传输。

安全传输过程中使用加密密钥以及完整性密钥，完成有保护需求的数字内容的保密传输，以及和该数字内容相关的保护标志及控制信息的完整传输；同时定期更新加密密钥以及完整性密钥。

以下分别以AES-CCM算法和流密码算法为例说明DICP安全传输过程。

7.1 加/解密算法

7.1.1 AES-CCM 算法

7.1.1.1 加密

发送端接口使用AES-CCM算法的加密流程：

a) 检测ISI，只有当ISI_0、ISI_1和ISI_2同时为0时，才调用以下算法完成数据的加密处理：

$$B\_E(K\_E_i,\ IV, data)$$

加密算法参数说明：

- K_E_i：加密密钥，其中i=[Key_Counter]$_{lsb0}$；
- IV：初始化向量；
- data:待加密数据。

b) 每次完成加密后，发送端接口设置Cipher_Counter或者Channel_Cipher_Counter如下：

$$Cipher\_Counter = Cipher\_Counter + Length(data)$$

$$Channel\_Cipher\_Counter = Channel\_Cipher\_Counter + Length(data)$$

c) 设置完成后，如果Cipher_Counter或者Channel_Cipher_Counter发生了溢出，则计算Key_Counter++。

d) 若Cipher_Counter或者Channel_Cipher_Counter大于2^{20}，则计算备份密钥如下：

$$K\_E_j = B\_E(K\_M_{lsb128\sim255}, K\_E||Key\_Counter)$$

其中：j=[K_Counter]$_{lsb0}$⊕1，j是1 bit变量，⊕表示比特异或运算。

e) 若Key_Counter的值大于ReAuthenticationMaxTime，则对于单播加密，设置接口的认证状态为没有认证，重新执行认证过程；对于广播加密，重新生成广播密钥，执行密钥激活。

示例：

当 ReAuthenticationMaxTime 设置为 2^{16}，吞吐率为 1Gb/s 时：

单播情况下，如果 Cipher_Counter 的计数范围为 $0\sim2^{32}-1$，则大约 3 天需重新认证一次；

广播情况下，如果 Channel_Cipher_Counter 的计数范围为 $0\sim2^{32}-1$，则大约 3 天需重新生成广播密钥，重新执行密钥激活。

注：广播加密情况，不涉及重新认证和主密钥的重新生成，所以安全性相对较低，如果具体接口的应用场景需要长期广播，需要考虑增加重新认证的计数器，并且需要考虑多台设备同时重新认证造成的负担。

7.1.1.2 解密

接收端接口使用AES-CCM算法的解密流程：

a) 检测ISI，只有当ISI_0、ISI_1和ISI_2同时为0时，才调用以下算法完成数据的解密处理：

$$B\_D(K\_E_i, IV, data)$$

解密算法参数说明：

- K_E_i：解密密钥，其中i=[Key_Counter]$_{lsb0}$；
- Alg_Para：必选IV：初始化向量；
- data:待解密数据。

b) 每次完成解密后，接收端接口按照源接口的方式(7.1.1.1b))设置密文计数器Cipher_Counter或者Channel_Cipher_Counter。

c) 设置完成后，如果Cipher_Counter或者Channel_Cipher_Counter发生了溢出，则计算Key_Counter++。

d) 若Cipher_Counter或者Channel_Cipher_Counter大于2^{20}，则按照发送端接口的方式(7.1.1.1d))计算备份密钥，新的备份密钥K_E和K_MAC放入以$[K_Counter]_{lsb0} \oplus 1$为索引的存储区。

e) 若Key_Counter的值大于ReAuthenticationMaxTime，则按照发送端接口的方式(见7.1.1.1e))进行处理。

f) 如果解密失败，则使用计数器D_W_Counter记录解密失败的次数，如果解密失败的次数大于D_W_Max，则认为当前安全传输出现了攻击行为，或者双方的密钥不匹配，此时需要重新执行密钥激活过程。

7.1.2 流密码算法

DICP安全传输过程规定：

a) 若使用的流密码算法能够同时保证数字内容的保密性和关键数据的完整性，则使用流密码算法的基本流程与使用AES-CCM算法的流程相同；

b) 若使用的流密码算法只能保证数字内容的保密性时，建议使用7.1.2.1和7.1.2.2所述的流密码进行加/解密。

注：关键数据的定义在具体接口中实现，如果传输的内容始终不包含任何关键数据，则可以不考虑本条提供的完整性算法。

7.1.2.1 加密

发送端接口使用流密码算法的加密流程：

a) 检测ISI，只有当ISI_0、ISI_1和ISI_2同时为0时，才使用关键数据识别算法生成该帧数据的关键数据位置信息。

注：关键数据识别算法可以在实现时根据不同的接口传输的不同格式的内容实现，也可以由识别管理单元统一处理，把关键数据封装在某一个特定位置。

b) 使用流密码加密算法加密一帧数据：

$$S\_E(K\_E, Alg\_Para(IV), Data)$$

c) 根据流密码算法对关键数据位置信息进行修正，修正后提取各关键位置的密文，构成KDI_Record，如果KDI_Record非空，设置1比特完整性标志S_A=1，否则设置S_A=0。

d) 计算杂凑值(单播和广播相同)：

$$[HMAC(K\_MAC, S\_A||KDI\_Record)]_{lsb0\text{-}127}$$

e) 封装得到的128 bit杂凑值，并与S_A串联在数据帧密文之后，发送给接收端接口。

f) 发送完成后，按照7.1.1.1中AES-CCM算法加密步骤b)~e)的处理方式对Cipher_Counter、Channel_Cipher_Counter和Key_Counter进行处理。

7.1.2.2 解密

接收端接口使用流密码算法的解密流程：

a) 检测ISI，只有当ISI_0、ISI_1和ISI_2同时为0时，才调用以下算法完成数据的解密处理：

$$S\_D(K\_E, Alg\_Para(IV), Data)$$

b) 检测S_A的值，若：

1) S_A==1，则使用关键数据识别算法获得关键数据位置信息，并根据解密算法修正关键数据位置信息，修正后提取关键位置的密文，构成KDI_Record；然后按照7.1.2.1中发送端接口加密流程d)的方式计算128 bit杂凑值。

2) S_A==0，则直接按照7.1.2.1中发送端接口加密流程d)的方式计算128 bit杂凑值。

c) 比较计算得到的杂凑值与接收的杂凑值是否相同：相同，则认为解密成功，否则认为解密失败。

d) 解密成功后，按照7.1.1.2中AES-CCM算法解密流程b)~e)的处理方式对Cipher_Counter、Channel_Cipher_Counter和Key_Counter进行处理。

e) 解密失败后，按照7.1.1.2中AES-CCM算法解密流程f)的处理方式对D_W_Counter进行处理。

注：使用流密码算法一般是为了获得高的加密处理速度，适用于具有严格同步信号的数据帧传输。上述操作中关键数据位置识别、杂凑值计算和加密可以在硬件上实现并行操作。解密操作、关键数据位置识别以及杂凑值计算则可以通过缓存一定长度的密文方式实现并行处理。

7.2 密文封装

密文封装格式如下：

DICP标志||加密标志||[完整性标志]||密文

其中，完整性标志是可选项，在使用7.1.2流密码算法并提供数据完整性时必选。

具体接口应用时，可以根据接口的传输带宽对上述封装重新编码，编码后的封装方式要能够表现编码前所有的信息。

8 系统完整性

一个完整的DICP应保证系统中所有的DICP设备、DICP接口都是合法的。DICP授权组织将非法的DICP设备持有证书的标识ID放入CRL中，同时将非法的DICP接口持有证书的标识ID放入CRL中，并定期组织签发CRL。

任何能够接入DICP系统的DICP设备必须支持CRL存储和更新功能。

注1：DICP设备的合法性由DICP设备中识别管理单元所持证书的合法性来标识。

注2：CRL只考虑DICP设备证书和DICP接口证书的吊销问题，DICP根CA证书和二级CA证书的吊销不在本部分范围之内。

注3：DICP中的CRL可以由DICP根CA直接签发，也可以由具有相应授权的二级CA签发。

8.1 DICP 中 CRL 格式及验证

8.1.1 CRL 格式

DICP中规定的CRL格式如下：

“系统标识符（8 bit）||类型（3 bit）||保留（5 bit）||版本号（18 bit）||签发者ID（22 bit）||CRL长度（32 bit）||吊销纪录（L bit）||……||吊销纪录（L bit）||签名（384 bit）”

证书吊销列表（CRL）字段说明：

a) 系统标识符：DICP系统中为“0x5f”，区别与其它系统中的CRL；

b) 类型：吊销列表的类型，现在只定义了类型为0的CRL，其余值保留；

c) 保留：保留字节，未定义，全部置0；

d) 版本号：是一个表示CRL签发时间顺序、逐次递增的整数：签发时间越晚，版本号越大；

e) 签发者ID：此CRL签发者（DICP根CA或者有相应授权的二级CA）的ID；

f) CRL长度：表示整个CRL长度的整数，单位为字节；

g) 吊销记录：格式为“吊销记录类型（CRRT）||吊销内容”；

h) 签名：CRL签发者对以上所有信息的有效签名。

“吊销记录类型”长度为2 bit，吊销记录的长度L（单位：比特）是可变的，具体情况如表5所示。

表5 吊销纪录的格式和含义

CRRT	吊销记录格式	吊销记录长度L bit	实际被吊销的证书	说明
0b00	0b00||被吊销 ID	56	证书 ID 为“被吊销 ID”的单个公钥证书	被吊销 ID 为 54 比特
0b01	0b01||被吊销 ID_1||被吊销 ID_2||0b00	112	证书 ID 对应数值位于闭区间[被吊销 ID_1, 被吊销 ID_2]之间的所有公钥证书	一条纪录可以吊销多个公钥证书，要求“被吊销 ID_1”<“被吊销 ID_2”，即在数值上“被吊销 ID_1”小于“被吊销 ID_2”
0b10	0b10||吊销内容	未定义	未定义，备用	暂时将吊销内容部分全部置零
0b11	0b11||吊销内容	未定义	未定义，备用	暂时将吊销内容部分全部置零

本部分中只定义了证书吊销记录类型（CRRT）为0b00和0b01的两种情形，分别对应吊销单个证书和吊销ID连续的一批证书的情况，类型为0b10和0b1的两种格式未定义，如表5所示。

8.1.2 CRL 验证

DICP中，任何CRL在存储和更新前应首先验证该CRL的有效性。

DICP中用于CRL验证的模块Valid_CRL()描述如下：

a) 调用形式：

Res=Valid_CRL(CRL_x, $Cert\_CRL_x\_Adm$)

b) 输入参数列表：

1) CRL_x：待验证的CRL本身，长度可变，具体长度数值见其内部的“CRL长度”字段；

2) $Cert\_CRL_x\_Adm$：签发CRL_x的CA所持有的公钥证书，可以是DICP根证书，也可以是某一具有相应授权的二级CA所持有的公钥证书，长度固定为108字节。

c) 输出参数列表：

Res：1 bit，Res==1，表示输入的CRL是有效的；Res==0，表示输入的CRL是无效的。

d) 功能实现流程：

1) 检查输入参数CRL_x和$Cert_CRL_x_Adm$是否存在：若二者存在且长度与本节b)中输入参数列表中规定一致，则执行第2)步；否则，令Res=0，执行第8)步；

2) 比较CRL_x中的签发者ID与$Cert_CRL_x_Adm$中的持有者ID是否相同：若二者相同，则执行第3)步；否则，令Res=0，执行第8)步；

3) 比较$Cert_CRL_x_Adm$与DICP根CA证书Cert_Root是否相同：若二者相同，则执行第4)步；否则，执行第5)步；

4) 调用签名验证算法，用Cert_Root中的公钥验证CRL_x中签名的有效性：若签名验证成功，则令Res=1，执行第7)步；否则，令Res=0，执行第8)步；

5) 调用签名验证算法，用DICP根CA证书Cert_Root中的公钥验证$Cert_CRL_x_Adm$中签名的有效性：若签名验证成功，则执行第6)步；否则，令Res=0，执行第8)步；

6) 检查$Cert_CRL_x_Adm$中持有者功能标识的LSB2位是否等于1：若LSB2==1，则执行第7)步，否则，令Res=0，执行第8)步；

7) 调用签名验证算法，用$Cert_CRL_x_Adm$中的公钥验证CRL_x中签名的有效性：若签名验证成功，则令Res=1，执行第8)步；否则，令Res=0，执行第8)步；

8) 模块运行结束，返回Res值。

8.2 完整性信息维护

8.2.1 设备内 CRL 更新

DICP设备内部的识别管理单元和每个DICP接口都应有独立的S_CRL存储区，用以存放CRL；同时DICP设备内部存放的多个CRL应尽量及时保持版本一致，即：当S_CRL存储区足够大且设备内存在不同版本的CRL时，要及时用其中最高版本的CRL来替换原来较低版本的CRL。

DICP设备内CRL的更新可以参照设备间DICP接口CRL的更新过程。

8.2.2 设备间 DICP 接口 CRL 更新

当两个DICP接口在连接过程中检测到CRL版本不一致时，连接双方应启动设备间DICP接口CRL更新协议。

根据接口传输类型的不同规定了两种接口间CRL更新协议：双向通信的DICP接口CRL更新协议和单向通信的DICP接口CRL更新协议。

8.2.2.1 双向通信的 DICP 接口 CRL 更新协议

假定执行双向通信CRL更新协议的两个DICP接口分别为接口A和接口B，接口A为更新发起方，接口B为更新响应方。图14是DICP双向通信CRL更新协议发起方A状态图，图15是DICP双向通信CRL更新协议响应方B状态图。

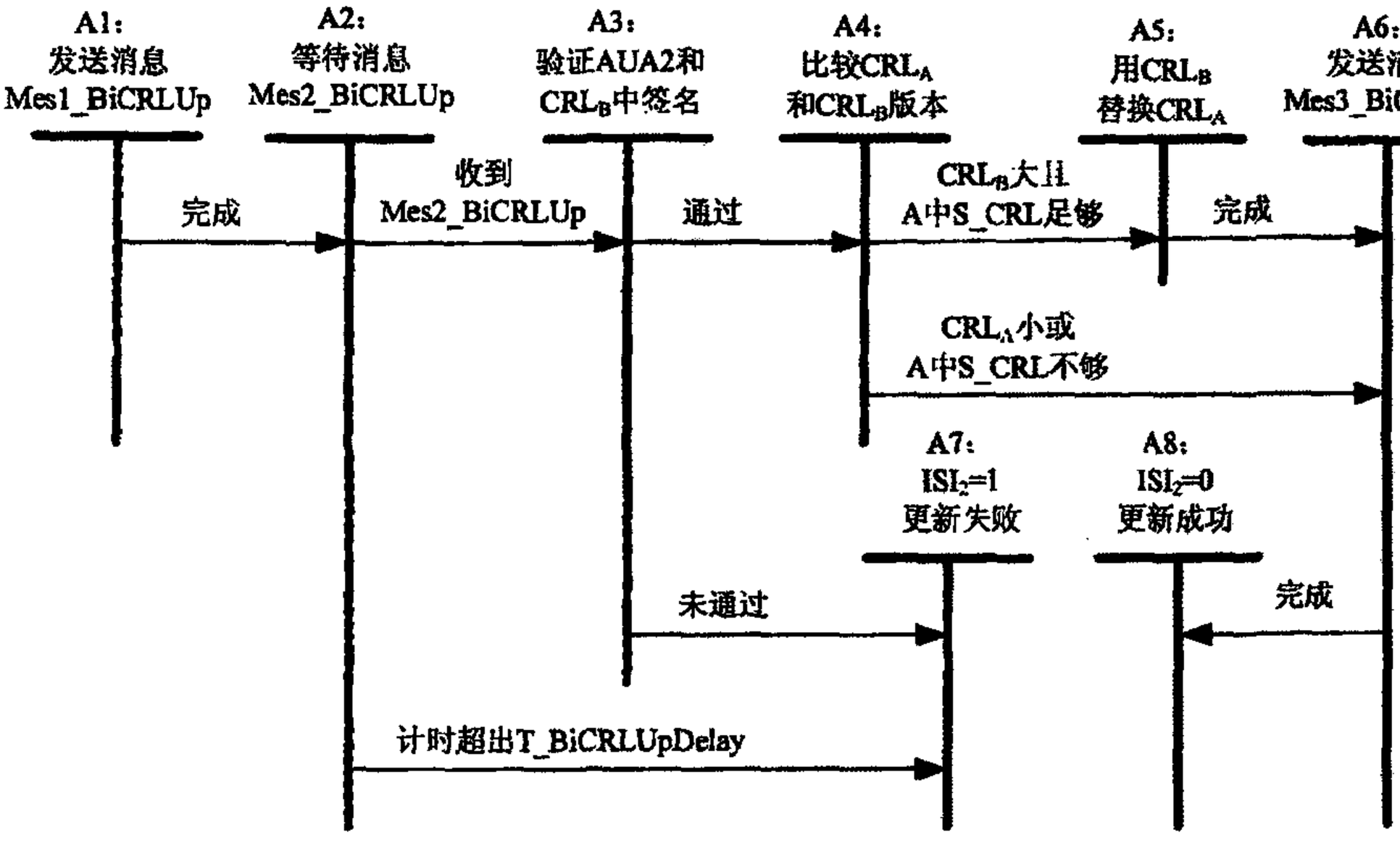


图14 DICP 双向通信 CRL 更新协议发起方状态图

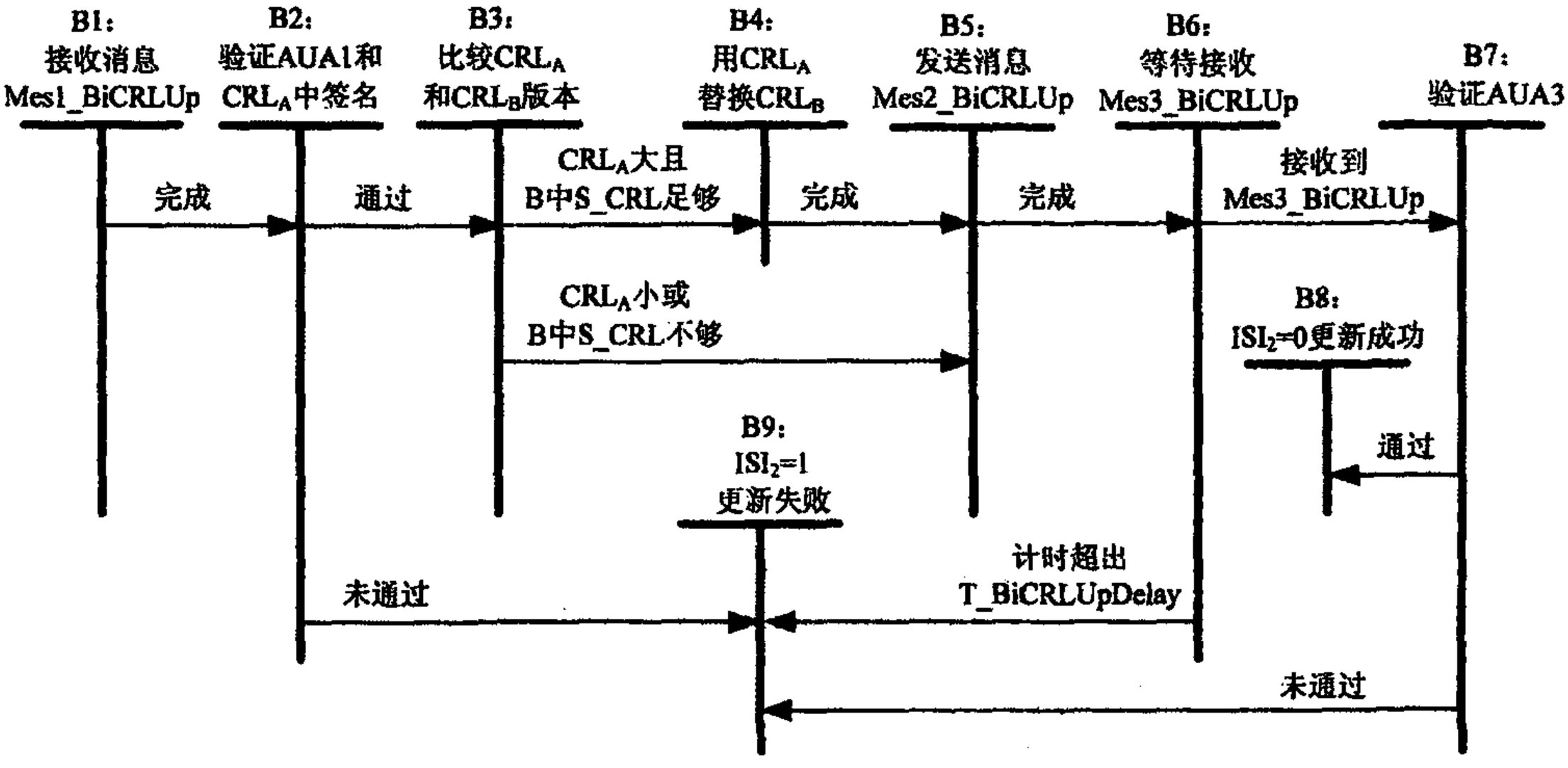


图15 DICP双向通信CRL更新协议响应方状态图

A、B双方利用K_M计算出共同的CRL传输认证密钥K_CRL(256 bit)，如下：

K_CRL=K_G（K_M,“Key for DICP-CRL-Updata-Message”）

注：K_CRL计算中参数“Key for DICP-CRL-Updata-Message”是固定字符串常量。

DICP双向CRL更新实现流程如下：

a) 发起方A在双向通信CRL更新协议中按顺序执行以下流程：

1) 向B发送消息Mes1_BiCRLUp。

Mes1_BiCRLUp消息内容如下：

DICP标志||CRL更新协议||第1轮消息标志||协议消息

协议消息包含内容：

CRL_A, $Cert\_CRL_A\_Adm$, RA_CRL, AUA1

协议消息内容说明：

- CRL_A：A中存储的CRL版本；
- $Cert\_CRL_A\_Adm$：签发CRL_A的CA证书；
- RA_CRL：随机数，由Rand(128)输出；
- AUA1：消息认证码。

AUA1=HMAC(K_CRL,(CRL_A||$Cert\_CRL_A\_Adm$||RA_CRL))

2) 消息Mes1_BiCRLUp发送完成后，A设立等待时间T_BiCRLUpDelay，等待接收B发送的第二轮消息Mes2_BiCRLUp：若在T_BiCRLUpDelay时间内成功接收到消息Mes2_BiCRLUp，则执行下一步；否则，执行第7)步。

3) 验证消息Mes2_BiCRLUp中AUA2和CRL_B中签名：若验证①和②都顺利通过，则执行下一步；否则，执行第7)步。

①、②两个验证过程如下：

- 执行验证①：验证关系式
  AUA2==HMAC(K_CRL,(CRL_A||$Cert\_CRL_A\_Adm$||CRL_B||$Cert\_CRL_B\_Adm$||RA_CRL||RB_CRL))是否成立；
- 执行验证②：调用Valid_CRL()模块，验证新接收到的CRL_B中签名的有效性。

4) 比较CRL_A和CRL_B的版本号：若CRL_B的版本号更高、且A中S_CRL可以放下CRL_B，则执行下一步；否则，执行第6)步。
5) 用CRL_B||Cert_CRLB_Adm替换原来的CRLA||Cert_CRLA_Adm，执行下一步。
6) 向B发送消息Mes3_BiCRLUp，发送完成后A将与B的认证标志位设置为ISI_2=0，更新协议成功执行完毕。

Mes3_BiCRLUp消息内容如下：

DICP标志||CRL更新协议||第3轮消息标志||协议消息

其中协议消息为：“AUA3”，其计算过程如下：

AUA3=HMAC(K_CRL,(CRL_B||Cert_CRL_B_Adm||RB_CRL))

7) A将与B的认证标志位设置为ISI_2=1，CRL更新协议执行失败。

b) 响应方B在双向通信CRL更新协议中按顺序执行以下流程：
1) 等待接收由A发送的第一轮消息Mes1_BiCRLUp，收到消息Mes1_BiCRLUp后执行下一步。
2) 验证消息Mes1_BiCRLUp中AUA1和CRL_A中签名：若验证③和④都顺利通过，则执行下一步；否则，执行第9)步。

③、④两个验证过程如下：

- 执行验证③：验证关系式AUA1==HMAC(K_CRL,(CRL_A||Cert_CRL_A_Adm||RA_CRL))是否成立；
- 执行验证④：调用Valid_CRL()模块，验证新接收到的CRL_A中签名的有效性。

3) 比较CRL_A和CRL_B的版本号：若CRL_A的版本号更高，且B中S_CRL可以放下CRL_A，则执行执行下一步；否则，执行第5)步。
4) 用CRL_A||Cert_CRL_A_Adm替换原来的CRL_B||Cert_CRL_B_Adm，执行下一步。
5) 向A发送消息Mes2_BiCRLUp。

Mes2_BiCRLUp消息内容如下：

DICP标志||CRL更新协议||第2轮消息标志||协议消息

协议消息包含内容：

CRL_B, Cert_CRL_B_Adm, RB_CRL, AUA2

协议消息内容说明：

- CRL_B：B中存储的CRL版本；
- Cert_CRLB_Adm：签发CRLB的CA证书；
- RB_CRL：随机数，由Rand(128)输出；
- AUA2：消息认证码。

AUA2=HMAC(K_CRL(CRL_A||Cert_CRL_A_Adm||CRL_B||Cert_CRL_B_Adm||RA_CRL||RB_CRL))

6) 消息Mes2_BiCRLUp发送完成后，B设立等待时间T_BiCRLUpDelay，等待接收A发送的第三轮消息Mes3_BiCRLUp：若在T_BiCRLUpDelay时间内成功接收到消息Mes3_BiCRLUp，则执行下一步；否则，执行第9)步。
7) 验证消息Mes3_BiCRLUp中AUA3：若验证⑤通过，则执行下一步；否则，执行第9)步。

⑤验证过程如下：

验证关系式AUA3==HMAC(K_CRL,(CRL_B||Cert_CRL_B_Adm||RB_CRL))是否成立。

8) B将与A的认证标志位设置为ISI_2=0，更新协议成功执行完毕。
9) B将与A的认证标志位设置为ISI_2=1，更新协议执行失败。

8.2.2.2 单向通信的 DICP 接口模块 CRL 更新协议

假定执行单向通信CRL更新协议的两个DICP接口分别为接口C和接口D，接口C为更新发起方，接口D为更新响应方。图16是DICP单向通信CRL更新协议发起方C状态图，图17是DICP单向通信CRL更新协议响应方D状态图。

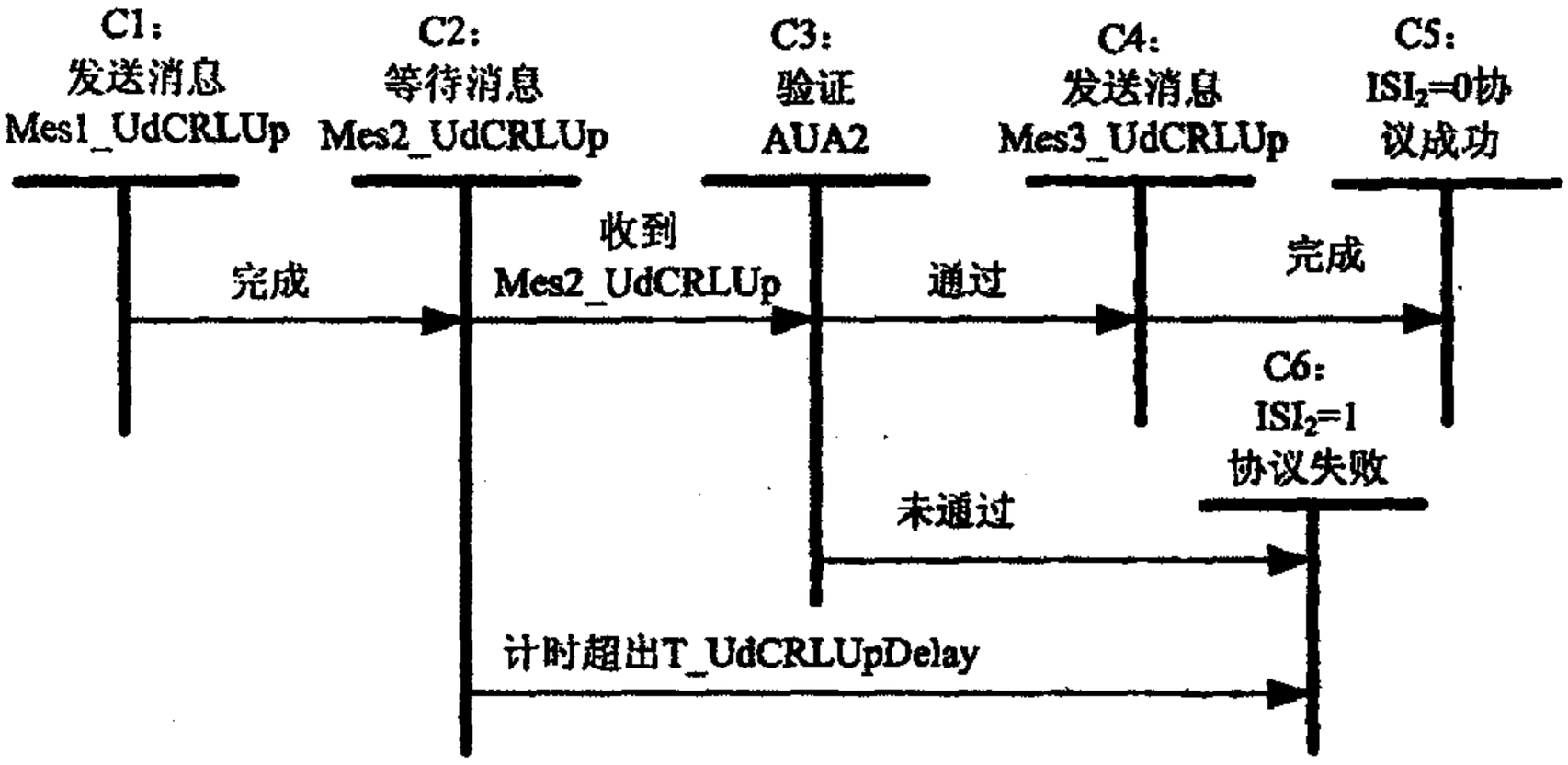


图16 DICP单向通信CRL更新协议发起方状态图

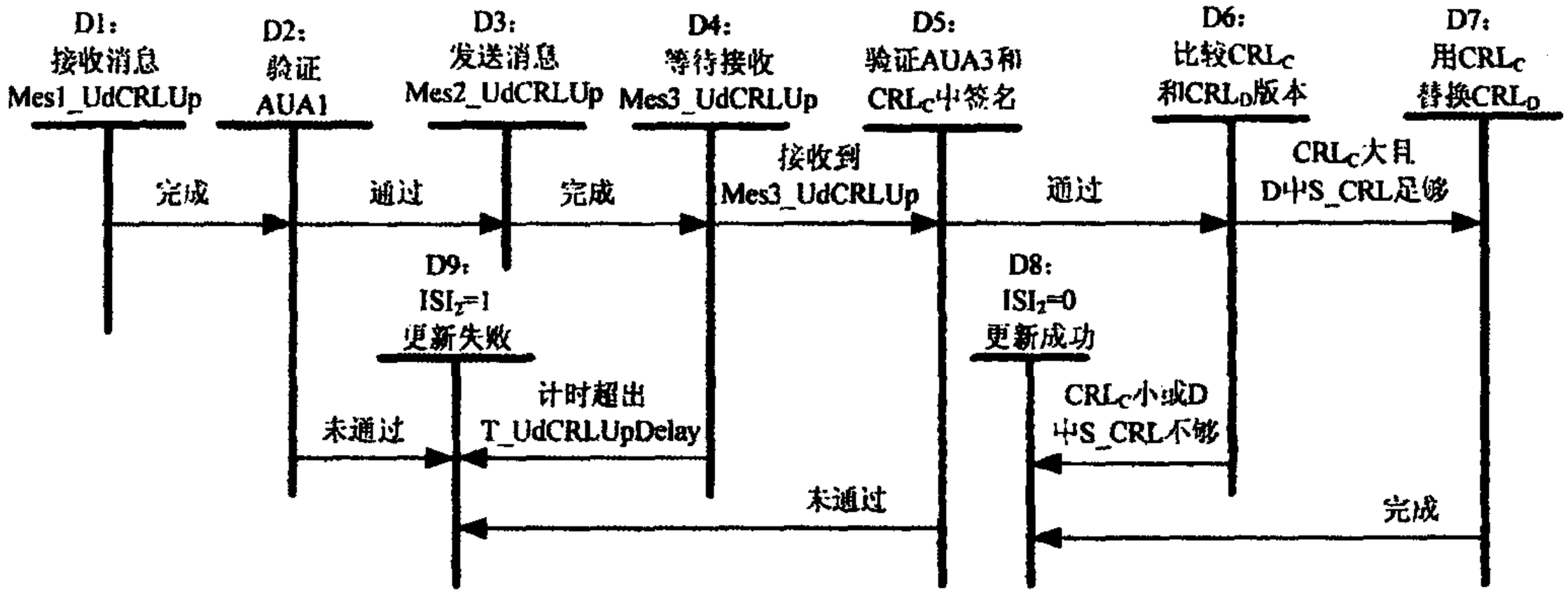


图17 DICP单向通信CRL更新协议响应方状态图

C、D双方利用K_M计算出共同的CRL传输认证密钥K_CRL(256 bit)，如下：

K_CRL=K_G（K_M，“Key for DICP-CRL-Updata-Message”）

注1：通信C和D之间只有C→D的单向数据信道，D→C的反向数据信道不存在。

注2：K_CRL计算中参数“Key for DICP-CRL-Updata-Message”是固定字符串常量。

DICP单向CRL更新实现流程如下：

a) 发起方C在单向CRL更新协议中按顺序执行以下流程：

1) 向D发送消息Mes1_UdCRLUp。

Mes1_UdCRLUp消息内容如下：

DICP标志||CRL更新协议||第1轮消息标志||协议消息

协议消息包含内容：

RC_CRL, AUA1

协议消息内容说明：

- RC_CRL：随机数，由Rand(128)输出；

● AUA1：消息认证码。

AUA1=HMAC(K_CRL,("DICP-CRL-Updata-Begin"||RC_CRL))

注："DICP-CRL-Updata-Begin"是固定字符串常量。

2) 消息Mes1_UdCRLUp发送完成后，C设立等待时间T_UdCRLUpDelay，等待接收D发送的第二轮消息Mes2_UdCRLUp：若在T_UdCRLUpDelay时间内成功接收到消息Mes2_UdCRLUp，则执行下一步；否则，执行第6)步。

3) 验证消息Mes2_UdCRLUp中AUA2：若验证⑥通过，则执行下一步；否则，执行第6)步。

验证⑥过程如下：

验证关系式AUA2==HMAC(K_CRL,(RC_CRL||RD_CRL))是否成立。

4) 向D发送消息Mes3_UdCRLUp。发送完成后执行下一步。

Mes3_UdCRLUp消息内容如下：

DICP标志||CRL更新协议||第3轮消息标志||协议消息

协议消息包含内容：

CRL_c, Cert_CRL_c_Adm, AUA3

协议消息内容说明：

● CRL_c：C中存储的CRL版本；

● Cert_CRLC_Adm：签发CRLC的CA证书；

● AUA3：消息认证码。

AUA3=HMAC(K_CRL,(CRL_c||Cert_CRL_c_Adm||RC_CRL||RD_CRL))

5) C将与D的认证标志位设置为ISI_2=0，更新协议成功执行完毕。

6) C将与D的认证标志位设置为ISI_2=1，更新协议执行失败。

b) 响应方D在单向CRL更新协议中按顺序执行以下流程：

1) 等待接收由C发送的第一轮消息Mes1_UdCRLUp，收到消息Mes1_UdCRLUp后执行下一步。

2) 验证消息Mes1_UdCRLUp中AUA1：若验证⑦通过，则执行下一步；否则，执行第9)步。

验证⑦过程如下：

验证关系式AUA1==HMAC(K_CRL，("DICP-CRL-Updata-Begin"||RC_CRL))是否成立。

3) 向C发送消息Mes2_UdCRLUp，发送完后执行下一步。

Mes2_UdCRLUp消息内容如下：

DICP标志||CRL更新协议||第2轮消息标志||协议消息

协议消息包含内容：

RD_CRL, AUA2

协议消息内容说明：

● RD_CRL：随机数，由Rand(128)输出；

● AUA2：消息认证码。

AUA2=HMAC(K_CRL,(RC_CRL||RD_CRL))

4) 消息Mes2_UdCRLUp发送完成后，D设立等待时间T_UdCRLUpDelay，等待接收C发送的第三轮消息Mes3_UdCRLUp：若在T_UdCRLUpDelay时间内成功接收到消息Mes3_UdCRLUp，则执行下一步；否则，执行第9)步。

5) 验证消息Mes3_UdCRLUp中AUA3和CRL_c中签名：若验证⑧和⑨都顺利通过，则执行下一步；否则，执行第9)步。

验证⑧、⑨过程如下：

● 执行验证⑧：验证关系

AUA3==HMAC(K_CRL,(CRL_c||Cert_CRL_c_Adm||RC_CRL||RD_CRL))是否成立；

● 执行验证⑨：调用Valid_CRL()模块，验证新接收到的CRL_c中签名的有效性。

6) 比较CRL_c和CRL_b的版本号：若CRL_c的版本号更高，且D中S_CRL可以放下CRL_c，则执行下一步；否则，执行第8)步。

7) 用CRL_c，$Cert_CRL_c_Adm$替换原来的CRL_b，$Cert\_CRL_b\_Adm$，执行下一步。

8) D将与C的认证标志位设置为ISI_2=0，更新协议成功执行完毕。

9) D将与C的认证标志位设置为ISI_2=1，更新协议执行失败。

附 录 A
（资料性附录）
一种伪随机数生成器及伪随机数生成方法

随机数在许多网络安全应用中扮演着重要的角色。基于密码学的大量网络安全算法都使用了随机数，例如，1)、在认证方案中的密钥分配，用临时交互号来作为握手信息之一，以阻止重复攻击；2)、会话密钥产生，可由密钥分配中心或者由委托方产生；3)、公钥加密算法中密钥的产生，等等。

在相互认证或者会话密钥生成之类的应用中，对随机数的统计随机性的要求并不很高，但是要求产生的随机数序列是不可预测的。所谓的“真随机数序列”，是各个数之间的统计独立性而使序列不可预测。不过，真正的随机数序列很少用，一般的随机数序列是由算法产生的，只要敌手不能够从先前的随机数推导出后面的随机数就行了，这样的数一般称为伪随机数。

随机数生成器已经嵌入在大多数编译器中了，产生随机数仅仅是函数调用而已。计算机不可能产生完全随机的数字，所谓的随机数发生器都是通过一定的算法对事先选定的若干个随机种子做复杂的运算，用产生的结果来近似模拟完全随机数，这种随机数就是伪随机数。伪随机数是以相同的概率从一组有限的数字中选取的。所选数字并不具有完全的随机性，但是从实用的角度而言，其随机程度已足够了。

随机数的应用对随机数的产生提出了两个不同的要求：随机性和不可预测性。

随机序列应该具有良好的统计特性。其评价标准是：

a) 分布一致性：序列中的随机数的分布应该是一致的，即出现的频率大约相等；

b) 独立性：序列中的任何数都不能由其它数导出。

对随机序列的分布一致性已经有较好的测试方法。但是，尽管有许多测试方法可以用于表明一个序列的独立性不好，还没有某种方法可以表明一个序列的独立性好。通常的策略是多进行一些测试，直到可认为它的独立性是足够强的。密码算法大量使用了这种“似乎随机”的随机数序列，即伪随机数序列。

本附录提供了一种伪随机数生成方法，可以在没有外部即时输入的情况下，产生具有良好密码性质的伪随机数。如图A.1所示。

伪随机数产生过程如下：

a) 在寄存器T、V中分别存入128 bit的随机数种子T_0、V_0以及3处128位AES加密的密钥K1、K2、K3；

b) 利用长度为128 bit的密钥K1，采用AES算法对随机数种子T_i进行加密计算，得到的密文与T_i异或，得到结果X_i；

c) 将上述结果X_i作为T_{i+1}反馈回寄存器T中对随机数种子T_i进行更新；

d) 将上述结果X_i与随机数种子V_i进行异或，得到结果Y_i；

e) 利用长度为128 bit的密钥K2，采用AES算法对Y_i进行加密计算，得到的密文与Y_i进行异或，得到伪随机数R_i；

f) 将输出的伪随机数R_i与寄存器V中的随机数种子V_i进行异或，得到结果Z_i；

g) 利用长度为128 bit的密钥K3，采用AES算法对上述Z_i进行加密计算，得到的密文与Z_i异或，得到结果V_{i+1}；

h) 将上述结果V_{i+1}反馈回寄存器V中对随机数种子V_i进行更新。

由此可见，每次产生了伪随机数之后，都会对寄存器中的随机数种子进行更新，因此，随机数种子在前后两次生成伪随机数的过程中始终是不相同的，这也保证了生成伪随机数的不同。

在采用上述AES算法产生伪随机数的方法中，若用户需要的伪随机数为256 bit，则先产生两个128 bit的伪随机数，然后，将这两个伪随机数串接起来，满足用户需求；若用户需要的伪随机数为192 bit，

则在将两个伪随机数串接起来后，丢弃高位的64 bit，也可以丢弃低位的64 bit，就可以得到192 bit的伪随机数。

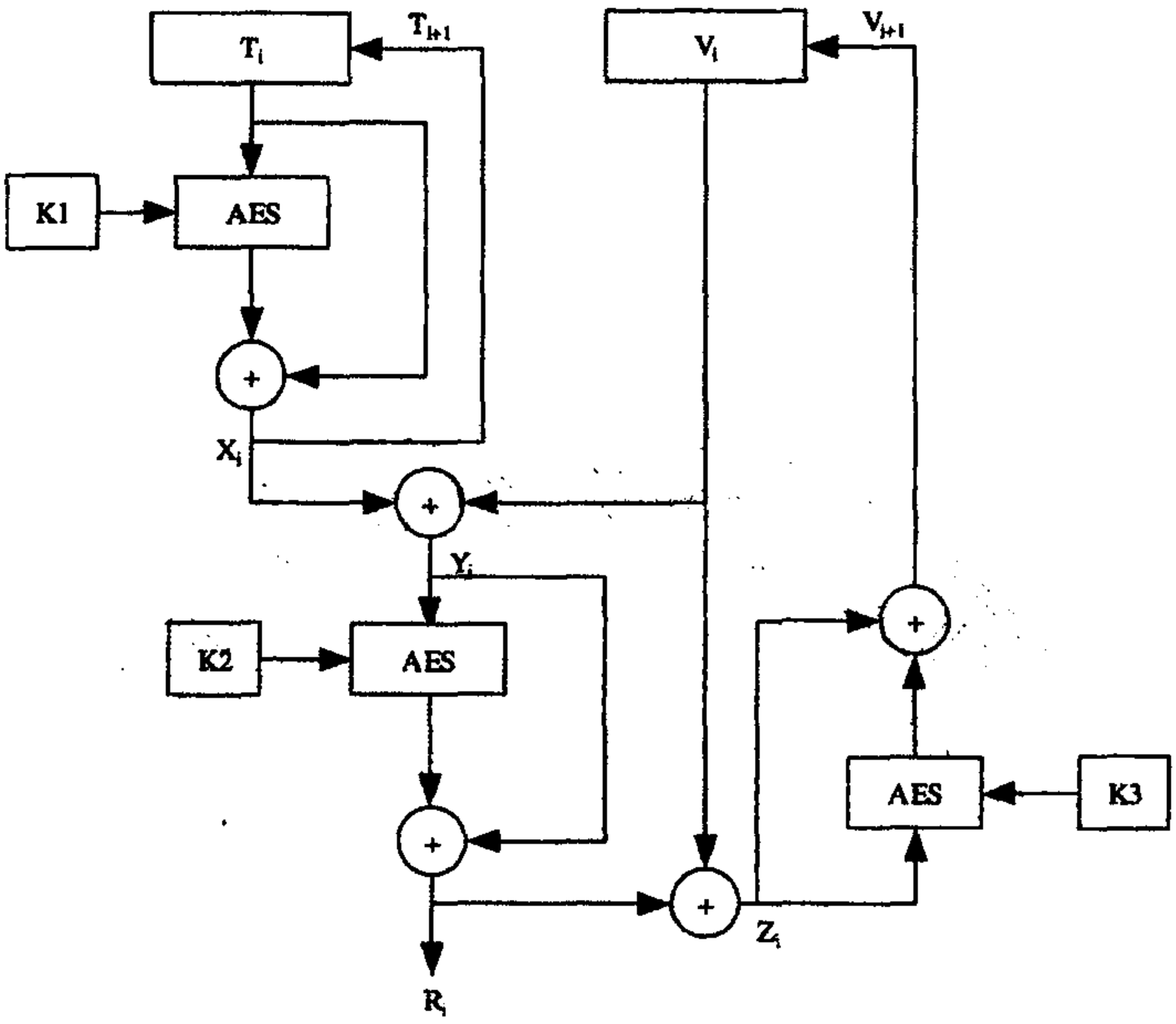


图 A.1 伪随机数产生

参考文献

[1] 数字电视广播条件接收系统规范（讨论稿），国家广播电影电视总局，2001。

[2] IEEE 1394协议及接口设计，张大朴、王晓等，西安电子科技大学出版社，2004.1。

[3] ANSI/SCTE412004: POD Copy Protection System., ENGINEERING COMMITTEE Digital Video Subcommittee, Available at http://www.scte.org.

[4] Digital Transmission Content Protection Specification Revision 1.3, January 7, 2004, Available at http://www.dtcp.com.

[5] DTCP Volume1 Supplement A Informational Version of Revision 1.2a, February 25, 2002, Available at http://www.dtcp.com.。

[6] High-bandwidth Digital Content Protection System, Revision 1.1, June 9, 2003, Available at http:// www.digital-cp.com.