

目 次

前言 III

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 长江电子航道图数据保护总体框架 1

 4.1 总体要求 1

 4.2 各参与方的职责 2

5 各参与方的交互程序 3

 5.1 SA 创建自己的数字证书 3

 5.2 DS 加入数据保护体系的基本程序 3

 5.3 OEM 加入数据保护体系的基本程序 3

 5.4 DC 向 OEM 购买新设备的程序 3

 5.5 SA 与 DC 的交互程序 4

 5.6 DC 购买并使用 DS 提供的电子航道图的程序 4

6 用户许可证 4

 6.1 M_ID 的格式 4

 6.2 M_KEY 的格式 4

 6.3 HW_ID 的格式 4

 6.4 用户许可证的格式 5

 6.5 用户许可证的创建程序 5

 6.6 用户许可证的解密程序 6

7 单元许可证 6

 7.1 一般要求 6

 7.2 电子航道图单元名称与过期日期的格式 7

 7.3 加密后的单元密钥(ECK1,ECK2)格式 8

 7.4 加密校验和的格式 8

 7.5 单元许可证的传输 8

 7.6 元许可证文件的定义 8

 7.7 创建单元许可证的程序 9

7.8	核查单元许可证文件的程序	11
7.9	核实单元许可证 ENC 校验和的程序	11
7.10	解密单元许可证中单元密钥的程序	11
8	数字证书	12
8.1	SA 公开密钥和 DS 公开密钥的格式	12
8.2	自我签名密钥格式	13
8.3	DS 证书的格式规则	14
8.4	数字证书的创建程序	15
9	数字签名	16
9.1	数字签名文件格式	16
9.2	ENC 数字签名文件格式	17
9.3	数字签名的创建程序	18
10	加密	19
10.1	一般要求	19
10.2	加密 ENC 基本单元文件的程序	19
10.3	解密 ENC 基本单元文件的程序	19
10.4	加密 ENC 更新文件的程序	19
10.5	解密 ENC 更新文件的程序	20
10.6	使用基本许可证文件选择单元密钥的程序	20
10.7	使用元许可证文件选择单元密钥	20

7.8 核查单元许可证文件的程序..... 11

7.9 核实单元许可证 ENC 校验和的程序 11

7.10 解密单元许可证中单元密钥的程序 11

8 数字证书..... 12

8.1 SA 公开密钥和 DS 公开密钥的格式 12

8.2 自我签名密钥格式..... 13

8.3 DS 证书的格式规则 14

8.4 数字证书的创建程序..... 15

9 数字签名..... 16

9.1 数字签名文件格式..... 16

9.2 ENC 数字签名文件格式 17

9.3 数字签名的创建程序..... 18

10 加密 19

10.1 一般要求 19

10.2 加密 ENC 基本单元文件的程序..... 19

10.3 解密 ENC 基本单元文件的程序..... 19

10.4 加密 ENC 更新文件的程序 19

10.5 解密 ENC 更新文件的程序..... 20

10.6 使用基本许可证文件选择单元密钥的程序 20

10.7 使用元许可证文件选择单元密钥 20

前 言

JT/T 765《长江电子航道图制作规范》分为五个部分:

- 第1部分:术语;
- 第2部分:数据传输;
- 第3部分:显示准则;
- 第4部分:数据有效性检验;
- 第5部分:数据保护。

本部分为 JT/T 765 的第 5 部分。

本部分按照 GB/T 1.1—2009 给出的规则起草。

本部分代替 JT/T 765.5—2009《长江电子航道图制作规范 第5部分:数据保护》,与 JT/T 765.5—2009 相比主要技术变化如下:

- 删除了原标准中的部分示例(见 2009 年版的 6.4 等),并将部分示例转化为文字叙述(见 6.4 等)。

本部分对应于国际海道组织(IHO)S-63《IHO 数据保护方案》(IHO Data Protection Scheme)。本部分与 IHO S-63 的一致性程度为非等效。

本部分由交通运输信息通信及导航标准化技术委员会提出并归口。

本部分起草单位:长江航务管理局、长江航道局、大连海事大学。

本部分主要起草人:但乃越、杜经农、朱业汉、杨大鸣、章娟、刘青、张娜、俞建林、王大彬、程大炜、赵德鹏、李源惠、宋良福、曹成、顾网林、李海、董华。

本部分所代替标准的历次版本发布情况为:JT/T 765.5—2009。

长江电子航道图制作规范

第 5 部分：数据保护

1 范围

JT/T 765 的本部分规定了长江电子航道图数据保护总体框架、各参与方的交互程序、用户许可证、单元许可证、数字证书、数字签名、加密等要求。

本部分适用于长江电子航道图制作、系统开发、设计和应用,其他内河电子航道图系统也可参照使用。

2 规范性引用文件

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

JT/T 765.1 长江电子航道图制作规范 第 1 部分:术语

3 术语和定义

JT/T 765.1 界定的术语和定义适用于本文件。

4 长江电子航道图数据保护总体框架

4.1 总体要求

4.1.1 长江电子航道图数据保护方案规定了四类参与方,分别为系统管理员、数据服务商、设备制造商、数据用户。

4.1.2 系统管理员为任何加入长江电子航道图数据保护方案的设备制造商分配一对唯一的制造商标识符(M_ID)和制造商密钥(M_KEY)值,并将所有设备制造商的 M_ID 和 M_KEY 记录下来,形成设备制造商信息表。设备制造商为购买其产品的数据用户分配一个唯一的硬件标识符(HW_ID),并用自己的 M_KEY 将其加密后形成用户许可证,随产品一同提供给数据用户。

4.1.3 对于任何加入长江电子航道图数据保护方案的数据服务商,系统管理员向其提供设备制造商信息表,并保证表内容的及时更新。

4.1.4 数据用户需要购买数据服务商的电子航道图时,向其提供自己的用户许可证,数据服务商用单元密钥对电子航道图数据进行加密,并利用用户许可证中的 HW_ID 加密单元密钥,形成单元许可证,将电子航道图和单元许可证发送给数据用户。数据用户收到电子航道图和单元许可证后,利用自己用户许可证中的 HW_ID 对单元许可证解密,得到单元密钥,并用单元密钥对电子航道图进行解密。

4.1.5 为在上述保护过程中实现身份认证,采用数字证书的方式来进行签名认证。系统管理员创建自己的系统管理员数字证书,并将该证书发布给设备制造商、数据服务商和数据用户。数据服务商创建自己的数据服务商自我签名密钥文件(SSK)并提交给系统管理员,系统管理员核实 SSK 文件后,用自己的私有密钥对数据服务商公开密钥进行签名,生成数据服务商数字证书,并发布给数据服务商。数据服务商在向数据用户发送电子航道图前,先用自己的私有密钥对电子航道图签名,并和数据服务商数字证

书结合,形成电子航道图的签名文件,随电子航道图一起提供给数据用户。数据用户收到电子航道图签名文件后,利用系统管理员数字证书中的公开密钥来核实数据服务商证书,再利用数据服务商证书中的公开密钥来核实电子航道图的签名,如果正确,则电子航道图可以使用,否则因不能确认来源而拒绝使用。

4.2 各参与方的职责

4.2.1 系统管理员

系统管理员(SA)应根据需要维护策略的正常运行,制作并发布方案根证书,接受设备制造商(OEM)的申请,并为符合要求的 OEM 创建 M_ID 和 M_KEY 信息。SA 还应为数据服务商(DS)发放数字证书,在电子航道图发布过程中实现身份认证。SA 的主要职责是:

- a) 确认 OEM 发送的 M_ID、M_KEY 申请表格;
- b) 确认 DS 发送的数据服务商申请表格;
- c) 与 OEM 签订协议书;
- d) 与 DS 签订协议书;
- e) 为每个 OEM 创建一一对应的 M_ID 和 M_KEY;
- f) 为每个 DS 创建数据服务商证书;
- g) 创建系统管理员数字证书以提供给电子航道图显示与信息系统(ECDIS)用户;
- h) 负责管理各种文档,并将系统成员所需文档发送给成员。

4.2.2 数据服务商

数据服务商(DS)应负责加密和签署电子航道图(ENC)信息,通过网络或者数据光盘的形式为用户提供更新 ENC 服务。每个长江电子航道图数据提供商应符合国家相关的资质规定,并经 SA 认证通过,具有向用户发布数据的资格。DS 的主要职责是:

- a) 向 SA 申请发布 ENC 数据的资格;
- b) 与 SA 签署协议,由 SA 向其分配 M_ID 和 M_KEY;
- c) 创建自己的 SSK 证书文件;
- d) 鉴定 ENC 请求用户的合法性;
- e) 创建加密 ENC 文件所需的单元密钥;
- f) 创建单元许可证文件和元数据文件;
- g) 压缩、加密以及签署 ENC 并发送给用户;
- h) 产生单元许可证文件和元数据文件,发送给请求用户;
- i) 管理加密 ENC 的密钥以及其他用户资料等文档文件。

4.2.3 设备制造商

设备制造商(OEM)主要负责开发一套电子航道图处理系统(EPS),该系统按照本规范要求具备对 ENC 数据的保护功能。同时,还具备处理保护的 ENC 数据,以便在 ECDIS 平台显示正确的电子航道图数据信息的能力。为了加入长江电子航道图数据保护方案,相应的 OEM 应完成方案的要求,取得资格认证后才能加入本保护方案,并且获取 M_ID 和 M_KEY 来实现他们的系统,提供支持该安全方案的船舶导航产品。OEM 的主要职责是:

- a) 申请制造商资格;
- b) 与 SA 签署协议;
- c) 为每台设备创建 HW_ID;

- d) 将每个 HW_ID 记录在 HW_ID 登记表中,并且进行管理;
- e) 开发符合本规范要求的 EPS 系统;
- f) 为电子航道图显示与信息系统(ECDIS)创建用户许可证文件。

4.2.4 数据用户

数据用户(DC)是电子航道图信息的使用者,从 DS 那里接收加以保护的 ENC 信息。ECDIS 用户负责鉴别 ENC 的数字签名和按照保护方案定义的程序对 ENC 进行解密。DC 的主要职责是:

- a) 从 SA 网站下载系统管理员数字证书,用以鉴别 DS 的合法性;
- b) 向 DS 提供用户许可证文件,以获得新的 ENC 文件;
- c) 管理多个 DS 各自的用户许可证文件。

5 各参与方的交互程序

5.1 SA 创建自己的数字证书

SA 创建自己的私有密钥和公有密钥,使用私有密钥对公有密钥进行签名,将签名算法的返回值与公有密钥结合,创建 SA 数字证书。

5.2 DS 加入数据保护体系的基本程序

DS 加入数据保护体系的基本程序如下:

- a) DS 创建自己的公开密钥、私钥对;
- b) DS 创建 SSK 文件;
- c) DS 填写“长江电子航道图数据保护系统数据服务商证书请求表”;
- d) DS 将申请表、SSK 文件以及数据服务商公开密钥文件发送给 SA;
- e) SA 通过电话、电子邮件等方式对申请表格和 SSK 文件的来源等进行确认;
- f) SA 核实 SSK 文件的格式是否符合本规范要求;
- g) SA 利用 DS 发送的公开密钥文件核实 SSK 文件;
- h) 如果对 SSK 文件的鉴别合法,且该 DS 符合国家相关资质要求,SA 创建 DS 证书,否则,给请求的 DS 发出错误信息。

5.3 OEM 加入数据保护体系的基本程序

OEM 加入数据保护体系的基本程序如下:

- a) OEM 完成 M_ID/M_KEY 申请表的填写(主要包括 OEM 的详细信息以及请求所需文件);
- b) SA 核实 M_ID/M_KEY 表格第一部分(主要核实设备制造商对表格的填写是否完整);
- c) SA 核实 OEM 是否签署“长江电子航道图数据保护协议书”;
- d) SA 核实 OEM 开发的系统是否符合本规范标准;
- e) SA 核实 OEM 没有 M_ID/M_KEY 对(主要确认该申请的 OEM 未曾有过 M_ID/M_KEY,即表明申请者是新的 OEM);
- f) SA 创建 M_ID/M_KEY,并且存储到 M_ID/M_KEY 登记表内;
- g) SA 将 M_ID/M_KEY 发送给 OEM,同时也将这个新的 M_ID/M_KEY 发送给系统内所有的 DS。

5.4 DC 向 OEM 购买新设备的程序

DC 按如下程序向 OEM 购买设备:

- a) DC 向 OEM 递交标识身份的用户许可证文件(主要是申请 ENC 文件);

- b) OEM 生成一份包含 DC 硬件设备标识 HW_ID 和 M_ID 信息的用户许可证文件;
- c) OEM 将研制生产的 EPS 系统和新生成的用户许可证文件以及 ECDIS 系统一起发送给 DC。

5.5 SA 与 DC 的交互程序

DC 按如下程序从 SA 处获得并验证数字证书:

- a) DC 从 SA 获取 SA 数字证书;
- b) 对 SA 数字证书格式进行核对;
- c) 利用 SA 公开密钥鉴别 SA 数字证书(比较数字证书,包括 SA 公开密钥与从 SA 网站获得的公开密钥是否一致)。

5.6 DC 购买并使用 DS 提供的电子航道图的程序

DC 购买并使用 DS 提供的电子航道图的程序如下:

- a) DC 向 DS 递交用户许可证以申请电子航道图文件;
- b) DS 从用户许可证中获得 M_ID,并查对从 SA 那里得到的 M_ID/M_KEY 登记表,找到与该 M_ID对应的 M_KEY;
- c) DS 使用 M_KEY 解密用户许可证,获得 DC 的 HW_ID,用来加密单元密钥;
- d) DS 生成单元许可证文件;
- e) DS 用 zip 算法压缩电子航道图单元数据,并用单元密钥加密压缩后的数据;
- f) DS 利用自己的私有密钥对加密后的电子航道图单元数据进行数字签名;
- g) DS 将加密后的电子航道图单元数据、单元许可证文件和数字签名文件一起发送给 DC;
- h) DC 核对 DS 发送的相关文件(主要包括签名文件是否包含签名和证书对,核实签名文件是否符合本规范要求的格式);
- i) DC 鉴别签名文件中的 DS 证书是否合法。如果合法,从签名文件中提取 DS 公开密钥;
- j) DC 鉴别电子航道图单元数据的签名文件;
- k) DC 通过系统软件提取自己机器上的 HW_ID 来解密单元许可证,得到电子航道图数据文件的单元密钥,利用系统解密数据并使用。

6 用户许可证

6.1 M_ID 的格式

M_ID 是 4 位长度的 16 进制数字(两字节),以 ASCII 码表示。SA 应向每个加入长江电子航道图数据保护体系的设备制造商分配一对 M_ID/M_KEY。

6.2 M_KEY 的格式

M_KEY 是 10 位长度的 16 进制数字(五字节)。

6.3 HW_ID 的格式

HW_ID 是五字节的十进制数字,由 OEM 分配给购买其系统的 DC。OEM 应为每一个系统指定一个唯一的 HW_ID,建议 HW_ID 以不连续的方式来分配。在用户许可证中 HW_ID 以加密的形式存在。HW_ID 用 M_KEY 作为加密密钥,利用 Blowfish 算法进行加密。加密以后的 HW_ID 是八字节长的十进制数,被转化成 16 位长度的 16 进制数,以 ASCII 码表示。

6.4 用户许可证的格式

用户许可证为 28 字符长度,用 ASCII 文本书写,其格式和字段长度要求见表 1。

表 1

加密后的 HW_ID	校验和	M_ID
16 位 16 进制数	8 位 16 进制数	4 位 16 进制数

表 1 中每个字段中的 16 进制数均以 ASCII 码表示,其中任何字母字符应大写。用户许可证中的校验和是用 CRC32 算法对加密后的 HW_ID 进行散列计算后得到。该散列计算结果被转化为 8 位 16 进制数。

示例:
用户许可证的最终表现形式为:73871727080876A07E450C043031。

6.5 用户许可证的创建程序

用户许可证的创建程序由系统或者应用程序提供商执行,生成的用户许可证发送给 DC,用作 DC 请求单元许可证时的自身标识。用户许可证和系统一起交给终端用户。用户许可证的创建过程如下:

- a) M_KEY 作为密钥,利用 Blowfish 算法对 HW_ID 进行加密;
- b) 将结果转换成 16 位 16 进制字符串,每一个字母都应大写表示;
- c) 利用 CRC32 算法散列上面的 16 位 16 进制字符串;
- d) 将 c) 输出的结果转化为 8 位 16 进制字符串,每一个字母都应大写表示,即 Check Sum(校验和);
- e) 将 d) 输出的结果附加在 b) 的输出结果;
- f) 将 M_ID 转化为 4 位 16 进制字符串,任何字母应以大写表示;
- g) 将 f) 的输出结果附加到 e) 的输出结果的后面,即用户许可证。

示例:
用户许可证的标识符及内容参见表 2,创建程序见表 3。

表 2

标 识 符	内 容	备 注
HW_ID	3132333438	16 进制表示的 HW_ID
M_KEY	3938373635	16 进制表示的 M_KEY
M_ID	3031	16 进制表示的 M_ID

表 3

步 骤	内 容	备 注
步骤 a) 的输入	3132333438 和 3938373635	16 进制表示的 HW_ID 和 M_KEY
步骤 a) 的输出	八字节	不可打印
步骤 c) 的输入	73871727080876A0	这是步骤 a) 输出的 16 进制字符串表现形式。这个字符串从左到右输入给散列算法,例如 73、87、17……
步骤 c) 的输出	7E450C04	用 16 进制表示的 CRC32 散列结果
步骤 e) 的输出	73871727080876A07E450C04	将 CRC32 散列结果附加在加密后的 HW_ID 后
用户许可证	73871727080876A07E450C043031	

6.6 用户许可证的解密程序

用户许可证的解密程序由 DS 的系统运行。用户许可证的结构在 6.4 中定义。用户许可证的解密程序如下：

- a) 从用户许可证中提取 M_ID(4 位 16 进制字符)；
- b) 从用户许可证中提取校验和(Check Sum,为 8 位 16 进制字符)；
- c) 利用 CRC32 算法散列加密的 HW_ID(用户许可证中前 16 位字符)；
- d) 比较 b)和 c)的输出,如果相同,则该用户许可证是合法的。如果不同,则该用户许可证是不合法的,从而不能获得相应的 HW_ID；
- e) 如果用户许可证是合法的,把这个加密的 HW_ID 转换成八字节；
- f) 将 M_KEY 作为解密密钥,利用 Blowfish 算法解密这个加密的 HW_ID,所得结果就是 HW_ID。

示例：

用户许可证的解密程序参见表 4 和表 5。

表 4

标 识 符	内 容	备 注
用户许可证	73871727080876A07E450C043031	
M_KEY	3938373635	16 进制的 M_KEY

表 5

步 骤	内 容	备 注
步骤 a)的输出	3031	提取的 M_ID
步骤 b)的输出	7E450C04	提取出来的校验和
步骤 c)的输入	73871727080876A0	字节从左到右输出给散列函数,例如 73、87、17
步骤 c)的输出	7E450C04	从加密的 HW_ID 中提取出来的校验和
步骤 f)的输出	3132333438	16 进制表示的 HW_ID

7 单元许可证

7.1 一般要求

7.1.1 电子航道图单元数据在发布前应由 DS 使用单元密钥进行加密,该单元密钥是与该电子航道图单元数据唯一对应的。DC 应获得这个密钥才能解密数据。这个单元密钥由 DS 以加密表格的形式——单元许可证提供给 DC。一个单元许可证文件可以包括一个或者多个电子航道图的单元密钥。

7.1.2 单元许可证的发布针对指定的 HW_ID,在安装系统之间是不可转换的。加密方案支持能在所有客户之间相互传送的普通加密 CD 形式。

7.1.3 DS 通过其得到的用户许可证获得用户的 HW_ID,通过用户的 HW_ID 加密电子航道图单元密钥从而形成单元许可证,确保单元许可证不能在 DC 之间进行传送。终端用户利用其 HW_ID 对单元许可证进行解密,以便获得解密电子航道图所需的单元密钥。

7.1.4 发布单元许可证需要两个文件：

- a) 包含解密电子航道图所需本质信息的基本许可证文件,在 7.5 中定义；

b) 包含易于数据管理和解密的补充信息的元许可证文件,在 7.6 中定义。

7.1.5 客户系统应负责管理来自多个 DS 的权证文件。来源于某个 DS 的权证文件不能用来解密来自另一个 DS 的电子航道图。

7.1.6 一个单元许可证包括两个加密的单元密钥,即包含当前版本电子航道图解密密钥的单元密钥 1 (ECK1),以及用作下一版本电子航道图单元的解密密钥 ECK2。单元许可证里面的 ECK2 使得 DS 能周期性地改变用于电子航道图下一版本的单元密钥,而无须对其他 DC 发布新的单元许可证。

7.1.7 除了这些密钥以外,单元许可证还包括过期日期(这个日期之后的更新文件不能使用)和单元标识(便于用户在有好几个单元和权证文件的情况下,系统能够将两者相匹配)。

7.1.8 终端用户将单元许可证里面的过期日期与需要解密的电子航道图单元进行比较,如果需要解密的电子航道图单元或者更新文件的生产日期在许可证的过期日期之后,系统将不会使用单元许可证来解密数据,以防止用户使用过期的电子航道图单元文件或者更新文件。

7.1.9 电子航道图的基本单元或更新文件的发布日期编码包含在电子航道图数据集文件的 DSID-ISDT 子字段里。这些数据文件总是加密的,除非已经解密,否则无法得到发布日期。为了提供一种获得发布日期而不需要解密这些电子航道图文件的简便机制,DSID-ISDT 字段的内容应被复制到数据集所含文件 CATALOG.031 的 CATD-COMT 字段中,以保障系统能快速检查这个电子航道图信息是否在许可证过期日期之后,而无须解密电子航道图。

示例:

CATALOG.031 文件中的 CATD-COMT 编码如下:

VERSION = 1.0,EDTN = 10,UPDN = 0,UADT = 20030224,ISDT = 20030224。

7.1.10 单元许可证以 ASCII 文本的形式书写,任何字母字符都应大写表示,其格式和字段长度见表 6。

表 6

航道图单元名称	过期日期	加密后的 CK1	加密后的 CK2	加密后的校验和
8 个字母或数字	8 个数字	16 位 16 进制数	16 位 16 进制数	16 位 16 进制数

示例:

单元许可证中各字段数据参见表 7。

表 7

航道图单元名称	过期日期	加密后的单元密钥 1 (ECK1)	加密后的单元密钥 2 (ECK2)	加密校验和
NO4D0613	20000830	BEB9BFE3C7C6CE68	B16411FD09F96982	795C77B204F54D48

单元许可证最终的表现形式为:NO4D061320000830BEB9BFE3C7C6CE68B16411FD09F96982795C77B204F54D48。

7.2 电子航道图单元名称与过期日期的格式

电子航道图单元名称是八字节的字母数字串,遵循 JT/T 765.2—2016 附录 B 中所定义的单元文件命名规则。

示例:

如表 7 所示,航道图单元名称表示为:NO4D0613。

过期日期的格式应为:年月日(YYYYMMDD)。

示例:

如表 7 所示,过期日期 2000 年 8 月 30 日表示为:20000830。

7.3 加密后的单元密钥(ECK1,ECK2)格式

7.3.1 单元密钥是五字节随机数字,使用从用户许可证中获得的数据用户的 HW_ID 进行 Blowfish 加密,然后转化成 16 进制。Blowfish 加密算法将需要加密的数据长度填充成八字节的倍数,加密后单元密钥从五字节长度(10 个 16 进制字符)变为八字节长度(16 个 16 进制字符)。

7.3.2 ECK1 包括电子航道图单元的当前版本的单元密钥。ECK2 包括用于下一版本电子航道图的单元密钥。

7.3.3 ECK1 用于可从 DS 获得的电子航道图的当前版本。如果电子航道图不能正确解密,应用 ECK2 解密。如果 ECK2 也不能正确解密,则应从 DS 获得更新的单元许可证。

示例:

如表 7 所示,加密后的 ECK1、ECK2 为:

- a) ECK1: BEB9BFE3C7C6CE68;
- b) ECK2: B16411FD09F96982。

7.4 加密校验和的格式

对单元许可证中加密校验和之前的所有字段用 CRC32 算法进行散列计算,然后用 HW_ID 对散列计算的结果进行加密,就得到了加密校验和。加密校验和是 16 位的 16 进制数字。

示例:

如表 7 所示,加密校验和为:795C77B204F54D48。

7.5 单元许可证的传输

单元许可证应存储在名称为 ENC. PMT 的文件里面。ENC. PMT 文件包括一个或多个作为连续记录存储在该文件中的单元许可证。

示例:

ENC. PMT 文件包括 2 个单元许可证:

- a) NO4D061320000830BEB9BFE3C7C6CE68B16411FD09F96982795C77B204F54D48;
- b) GB30401020000830698FFC77EC13C2FF442934F3563E50A4A858D1F973860701。

7.6 元许可证文件的定义

7.6.1 元许可证文件总体格式

元许可证文件(PERMIT.TXT)是针对终端用户开发者的,用来通知 DC 的征订状态,以及帮助开发者理解哪些基于客户征订结束日期和电子航道图版本日期的航道图单元需要解密。

在元许可证文件中任何字母的字符都应大写。DS 应提供如何从存储介质或网络中获得元许可证文件的方法。

元许可证文件中各部分的内容和格式见 7.6.2 ~ 7.6.6。

示例:

元许可证文件(PERMIT.TXT)的格式为:

```
:DATE 20000131 11:11
:VERSION 1
:ENC
GB30401020010110797C4CB45C5B4B7B0BC8A74680E798F2910351530D1FF963,0,2,
GB40401S200101100D0CC631D5B07B04252DAC4A7EC568BFD2C08583E26FF27E,0,3,
GB50401S200101100C9EAA1802FD92628086CE08A71125104F9FDFBE6C9656B5,0,3,
:ECS
```

7.6.2 日期和时间

字段名称: :DATE。

日期格式: YYYYMMDD。

时间格式: HH:MM ,使用 24h 制。

字段名称、日期和时间用空格字符分隔。

示例:

:DATE 20000131 11:11

7.6.3 版本

字段名称: :VERSION。

版本数字是一个整数,用来定义元许可证文件的格式版本数字。

字段名称与版本数字用空格字符分隔。

示例:

:VERSION 1

7.6.4 元许可证类型

字段名称: :ENC。

字段包含来自 DS 的 ENC 发布证书中的可用元许可证的定义。

字段里面包含一个或多个 7.6.5 中定义的单元许可证记录。

7.6.5 单元许可证记录

单元许可证记录的字段组成见表 8。

表 8

单元许可证记录的字段	备 注
单元许可证	在 7.1.10 中定义
服务级别指示	0 – 定期订购的许可证 1 – 单次购买的许可证
EDTN	电子航道图单元的 DSID-EDTN 参数
Reserved	为将来应用保留的参数字段
Comment	用于注释的文本字段

7.6.6 单元许可证类型 ECS

字段名称: :ECS。

字段包含来自 DS 的电子航道图系统的发布证书中的可用元许可证的定义。

字段里面包含一个或者多个 7.6.5 中定义的单元许可证记录。

7.7 创建单元许可证的程序

创建单元许可证的程序通常由 DS 执行,用于给特定 DC 创建单元许可证。

以下程序产生符合 7.1 中定义的单元许可证:

a) 移除电子 ENC 文件名称中的文件扩展名,剩下 8 个字符是单元许可证中的单元名称;

- b) 将过期日期,格式为 YYYYMMDD,附加在 a)中得到的单元名称的后面;
- c) 将 HW_ID 的第一个字节附加到 HW_ID 的后面形成六字节的 HW_ID(称作 HW_ID6),HW_ID6 就是所创建的用于加密单元密钥的一个 48 位密钥;
- d) 使用 c)中得到的 HW_ID6 作为加密密钥,利用 Blowfish 算法对单元密钥进行加密,以创建 ECK1;
- e) 将 ECK1 转换为 16 位 16 进制字符,任何字母字符都应以大写表示;
- f) 将 e)中的结果附加到 b)的后面;
- g) 将 HW_ID6 作为密钥,利用 Blowfish 算法,对单元密钥 2(CK2)进行加密,从而创建 ECK2;
- h) 将 ECK2 转换为 16 位 16 进制字符,任何字母字符都应以大写表示;
- i) 将 h)的输出结果附加到 f)的后面;
- j) 利用 CRC32 算法对 i)的输出结果进行散列计算,注意这里的散列计算是在转化为 16 进制字符之后,而用户许可证的散列计算是针对原始的二进制数据;
- k) HW_ID6 作为密钥,利用 Blowfish 算法对 j)中的散列结果进行加密;
- l) 将 k)中的输出结果转换为 16 位 16 进制字符串。任何字符字母都应大写,形成 ENC 的校验和;
- m) 将 l)的输出结果附加到 i)的后面,形成单元许可证。

示例：
创建单元许可证的程序参见表 9 和表 10。

表 9

标 识 符	内 容	备 注
HW_ID	3132333438	五字节的 16 进制字符
CK1	C1CB518E9C	五字节的 16 进制字符
CK2	421571CC66	五字节的 16 进制字符
Cell Name	NO4D0613.000	符合 S-57 标准(版本 3.1)的电子航道图单元文件名
Expiry Date	20000830	过期日期

表 10

步 骤	内 容	备 注
步骤 a)的输出	NO4D0613	电子航道图单元名称
步骤 b)的输出	NO4D061320000830	电子航道图单元名称 + 过期日期
步骤 c)的输出	313233343831	16 进制字符表示的 HW_ID6
步骤 d)或 e)的输出	BEB9BFE3C7C6CE68	16 进制字符表示的 ECK1
步骤 f)的输出	NO4D061320000830BEB9BFE3C7C6CE68	电子航道图单元名称 + 过期日期 + ECK1
步骤 g)或 h)的输出	B16411FD09F96982	16 进制字符表示的 ECK2
步骤 i)的输出	NO4D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982	电子航道图单元名称 + 过期日期 + ECK1 + ECK2
步骤 j)的输入	NO4D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982	步骤 i)输出的 ASCII 值,这些字节从左到右输入给散列函数,例如 NO_4D_06……
步骤 j)的输出	780699093	CRC32 散列计算的结果
步骤 k)的输出	八字节的不可打印值	加密后的 CRC32 计算结果
步骤 l)的输出	795C77B204F54D48	16 进制字符表示的 CRC32 计算结果
单元许可证	NO4D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982795C77B204F54D48	

7.8 核查单元许可证文件的程序

核查单元许可证文件的程序通常在 DC 系统内执行,由以下步骤组成:

- a) 单元许可证在文件 ENC. PMT 和 PERMIT. TXT 里面提供;
- b) DC 应用程序可以将单元许可证文件存储在任何适当位置。应确保应用于同一个 ENC 数据的许可证文件由同一个 DC 提供;
- c) 单元许可证中的前 8 个字符代表相关的 ENC 单元的单元名称(去掉扩展名字);
- d) ENC. PMT 或 PERMIT. TXT 文件可以包含几个许可证文件。

7.9 核实单元许可证 ENC 校验和的程序

核实单元许可证 ENC 校验和的程序通常由 DC 系统执行,由以下步骤组成:

- a) 从单元许可证提取最后的 16 位 16 进制字符(ENC 校验和);
- b) 将上面的 16 位 16 进制字符转化为八字节;
- c) 利用 CRC32 算法散列执行 a)以后所剩下的字符串;
- d) 将 HW_ID 的第一个字节附加到 HW_ID 的后面,形成六字节的 HW_ID(称作 HW_ID6);
- e) 以 HW_ID6 为密钥,利用 Blowfish 算法加密 c)中的散列结果;
- f) 比较 b)和 e)的结果。如果相同,则此单元许可证是合法的。如果不相同,则此单元许可证非法,不能使用。

示例:

核实单元许可证 ENC 校验和的程序参见表 11 和表 12。

表 11

标 识 符	内 容	备 注
HW_ID	3132333438	16 进制字符表示的 HW_ID
单元许可证	N04D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982795C77B204F54D48	

表 12

步 骤	内 容	备 注
步骤 a) 的输出	795C77B204F54D48	16 进制字符
步骤 b) 的输出	八字节的不可打印值	加密后的 CRC32
步骤 c) 的输入	N04D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982	去掉 CRC32 值后的单元许可证
步骤 c) 的输出	780699093	CRC32 散列计算结果
步骤 d) 的输出	313233343831	HW_ID6
步骤 e) 的输出	八字节的不可打印字符	加密后的 CRC32

7.10 解密单元许可证中单元密钥的程序

这个程序通常由 DC 执行,以提取用以解密电子航道图的单元密钥,由下面的几个步骤组成:

- a) 将 HW_ID 的第一个字节附加到 HW_ID 的后面,形成六字节的 HW_ID(称作 HW_ID6);
- b) 从单元许可证中提取出来 ECK1,并且将之从 16 位 16 进制字符转化为八字节;

- c) HW_ID6 作为密钥,利用 Blowfish 算法对 b) 的输出结果 ECK1 进行解密,形成单元密钥 1 (CK1);
- d) 从单元许可证中提取出来 ECK2,并且将之从 16 位 16 进制字符转化为八字节;
- e) HW_ID6 作为密钥,利用 Blowfish 算法对 d) 的输出结果 ECK2 进行解密,形成单元密钥 2 (CK2)。

示例:

解密单元许可证中单元密钥的程序参见表 13 和表 14。

表 13

标 识 符	内 容	备 注
HW_ID	3132333438	16 进制字符表示的 HW_ID
单元许可证	NO4D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982795C77B204F54D48	

表 14

步 骤	内 容	备 注
步骤 a) 的输出	313233343831	HW_ID6
步骤 b) 的输出	八字节不可打印值*	加密后的 ECK1
步骤 c) 的输出	C1CB518E9C	单元密钥 1 的 16 进制字符
步骤 d) 的输出	八字节不可打印值	加密后的 ECK2
步骤 e) 的输出	421571CC66	单元密钥 2 的 16 进制字符
*没有经过加密的单元密钥是五字节长度,但经过加密以后的单元密钥在长度上是八字节。这是因为 Blowfish 在加密时,将单元密钥填充到八字节长度,而当对其解密时对之进行减充。		

8 数字证书

8.1 SA 公开密钥和 DS 公开密钥的格式

SA 公开密钥和 DS 公开密钥都应以适当的文件格式提供。公开密钥(文件)采用 ASCII 文本格式,并按照规定的结构和次序表示。

文件应包括下面 4 个数据字符串:

- a) 第一个数据字符串“p”,共有 32 个字符串块,每块字符串由 4 个字符组成;
- b) 第二个数据字符串“q”,共有 10 个字符串块,每块字符串由 4 个字符组成;
- c) 第三个数据字符串“g”,共有 32 个字符串块,每块字符串由 4 个字符组成;
- d) 第四个数据字符串“y”,共有 32 个字符串块,每块字符串由 4 个字符组成,是 DS 公开密钥。

每一个数据字符串应遵守下述格式:

- a) 前面有一个标题行。标题行在开始处用“//”表示。标题线总是以这种次序出现;
- b) 用 16 进制表示(0~9,A~F),任何字母字符均以大写表示;
- c) 以点号“.”表示结束;
- d) 每 4 个字符之间用空格分开;
- e) 每一个数据字符最后有一个行分割符(为了增加可读性,一些数据被分成两行)。

示例:

```
// BIGp
D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.

// BIGq
8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.

// BIGg
B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710
E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.

// BIGy
444B BA17 1758 0DAF 71AB 52A5 6CCA 8EAB 4C51 E970 0E37 B17B BB46 C0B9 4A36 F73F
0244 7FBD AE5B 7CA9 3870 5AB9 E9EE 471C E7B0 1004 6DF1 3505 42B3 0332 AE67 69C6.
```

8.2 自我签名密钥格式

自我签名密钥由 DS 创建,然后提交给 SA 以获得 DS 证书。自我签名密钥的格式与 DS 证书的格式(见 8.3)相同,本质的区别是:自我签名密钥由 DS 本身创建,而 DS 证书由 SA 创建和发布。自我签名密钥(文件)采用 ASCII 文本格式,并按照规定的结构和次序表示。

自我签名密钥文件应包括下面 6 个数据字符串:

- a) 第一个数据字符串“R”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥的 DS 签名中的“R”元素;
- b) 第二个数据字符串“S”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥的 DS 签名中的“S”元素;
- c) 第三个数据字符串“p”,包括 32 块区域(每区域含 4 个字符);
- d) 第四个数据字符串“q”,包括 10 块区域(每区域含 4 个字符);
- e) 第五个数据字符串“g”,包括 32 块区域(每区域含 4 个字符);
- f) 第六个数据字符串“y”,包括 32 块区域(每区域含 4 个字符),是 DS 公开密钥。

每一个数据字符串应遵守下述格式:

- a) 前面有一个标题行。标题航在开始处用“//”表示,后跟一个空格字符和 ASCII 文本标题字符;
- b) 用 16 进制数字(0~9,A~F)表示,任何字母字符均以大写表示;
- c) 以点号“.”表示结束;
- d) 每 4 个字符之间用空格字符分开;
- e) 每一个数据字符最后有一个行分割符(回车)(为了增加可读性,一些数据被分成两行)。

示例:

```
// Signature part R:
752A 8E5C 3AF5 6CCD 7395 B52E F672 E404 554F AAB6.

// Signature part S:
1756 E5C0 F4B6 BC90 4EC6 5F94 DF93 3ADF 68B8 86C4.

// BIGp
D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.

// BIGq
8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.

// BIGg
B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710
```

```
E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.  
// BIGy  
444B BA17 1758 0DAF 71AB 52A5 6CCA 8EAB 4C51 E970 0E37 B17B BB46 C0B9 4A36 F73F  
0244 7FBD AE5B 7CA9 3870 5AB9 E9EE 471C E7B0 1004 6DF1 3505 42B3 0332 AE67 69C6.
```

8.3 DS 证书的格式规则

DS 使用 512 字节的数字签名算法公开密钥。DS 证书(文件)采用 ASCII 文本格式,并按照规定
的结构和次序表示。

DS 文件应包括下面 6 个数据字符串:

- a) 第一个数据字符串“R”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥文件里的 SA 签
名的“R”元素;
- b) 第二个数据字符串“S”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥文件里的 SA 签
名的“S”元素;
- c) 第三个数据字符串“p”,包括 32 块区域(每区域含 4 个字符),是 DS 公开密钥里面的“p”
元素;
- d) 第四个数据字符串“q”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥里面的“q”
元素;
- e) 第五个数据字符串“g”,包括 32 块区域(每区域含 4 个字符),是 DS 公开密钥里面的“g”
元素;
- f) 第六个数据字符串“y”,包括 32 块区域(每区域含 4 个字符),是 DS 公开密钥里面的“y”
元素。

每一个数据字符串应遵守下述格式:

- a) 前面有一个标题行。标题行在开始处用“//”表示,后跟一个空格字符和 ASCII 文本标题
字符;
- b) 用 16 进制数字(0~9,A~F)表示,任何字母字符均以大写表示;
- c) 以点号“.”表示结束;
- d) 每 4 个字符之间用空格字符分开;
- e) 每一个数据字符最后有一个行分割符(回车)(为了增加可读性,一些数据被分成两行)。

示例:

```
// Signature    part R:  
8FD6 2AC7 27D2 8D0B CD27 BDF2 5CC6 9656 10E3 751F.  
// Signature    part S:  
3DE7 DA37 5A40 80FC 4203 5C6E 37DE A984 2A88 2BDC.  
// BIGp  
D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12  
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.  
// BIGq  
8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.  
// BIGg  
B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710  
E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.  
// BIGy  
444B BA17 1758 0DAF 71AB 52A5 6CCA 8EAB 4C51 E970 0E37 B17B BB46 C0B9 4A36 F73F  
0244 7FBD AE5B 7CA9 3870 5AB9 E9EE 471C E7B0 1004 6DF1 3505 42B3 0332 AE67 69C6.
```

8.4 数字证书的创建程序

8.4.1 生成 SSK 的程序

生成 SSK 的程序通常由数据服务商执行一次,以便创建其自身的 SSK,然后发送到 SA,由 SA 根据该 SSK 创建 DS 证书。程序如下:

- a) 使用算法 SHA-1 散列公开密钥文件,文件里面的所有字节都应进行散列;
- b) 利用 DSA 数字签名算法通过私有密钥,公开密钥的文件的散列和一个随机字符串实现对公开密钥文件进行签名,这将返回 2 个签名元素(“R”和“S”);
- c) 将这个自我签名密钥文件按 8.2 规定的格式书写,然后将公开密钥文件附加到它的后面,形成自我签名密钥文件。

8.4.2 鉴别 SSK 的程序

鉴别 SSK 的程序通常由 SA 在创建和发布 DS 证书之前执行。程序如下:

- a) 提取数字签名元素“R”和“S”,剩余为公开密钥文件内容;
- b) 使用算法 SHA-1 散列公开密钥文件,文件里面的所有字节都应进行散列;
- c) 将 a) 中提取的签名元素,与公开密钥文件及 b) 中获得的公开密钥文件的散列值一起传递给 DSA 进行核实,并返回签名正确与否。

注:如果签名验证正确,SA 将产生 DS 证书。

8.4.3 创建 DS 证书的程序

创建 DS 证书的程序通常由 SA 在鉴别 SSK 以后创建 DS 证书时执行。程序如下:

- a) 从自我签名密钥文件中舍弃签名元素(即前两个元素以及它们的行头),剩余为公开密钥文件;
- b) 使用算法 SHA-1 散列公开密钥文件,文件里面的所有字节都应进行散列;
- c) 利用 DSA 数字签名算法通过 SA 私有密钥、公开密钥的文件的散列和一个随机字符串实现对公开密钥文件进行签名,并返回 2 个签名元素(“R”和“S”);
- d) 书写“R”和“S”内容到证书文件,然后附加 a) 中所得的结果,形成 DS 证书。

8.4.4 鉴别 DS 证书的程序

鉴别 DS 证书的程序通常是 DS 从 SA 获得的 DS 证书在使用之前执行。程序如下:

- a) 从 SA 网站获得 SA 公开密钥;
- b) 提取证书中的头两个数据字符串,剩下的就是公开密钥文件;
- c) 使用算法 SHA-1 散列公开密钥文件,文件里面的所有字节都应进行散列;
- d) 将 b) 中提取的签名元素,与 a) 中 SA 公开密钥及 c) 中公开密钥文件的散列值一起传递给 DSA 进行核实,并返回签名正确与否。

注:如果 DS 证书签名鉴别结果正确,其签名元素“R”和“S”可用于构建 ENC 数字签名。

8.4.5 从签名文件中鉴别 DS 证书的程序

从签名文件中鉴别 DS 证书的程序通常由 DC 执行,在数据服务公开密钥被提取用于鉴别 ENC 签名之前,使用 SA 公开密钥鉴别存储在 ENC 签名中的 DS 证书时使用。鉴定程序如下:

- a) 从签名文件中提取所期望的签名和证书;
- b) 舍弃第一个签名部分(前两个数据字符串和其相应标题,这是 ENC 数据的数据服务签名),留下部分是证书;

- c) 提取剩下的签名部分,即从 b) 中获得的剩下的文件中的前两个数据字符串,形成公开密钥文件;
- d) 使用算法 SHA-1 散列公开密钥文件,文件里面的所有字节都应进行散列;
- e) 将 c) 中提取的签名部分,与 e) 中获得的 SA 公开密钥文件及 d) 中获得的公开密钥文件的散列值一起传递给 DSA 进行核实,并返回签名正确与否。

8.4.6 鉴别 SA 数字证书的程序

8.4.6.1 鉴别 SA 数字证书的程序通常在以下情况下执行:

- a) DS 核实 SA 公开密钥时,该 SA 公开密钥用于鉴别 DS 证书;
- b) DC 核实 SA 公开密钥时,该 SA 公开密钥用于鉴别 ENC 数据的数字签名。

8.4.6.2 鉴别 SA 数字证书的程序如下:

- a) 手动比较独立安装的 SA 数字证书中的 SA 公开密钥和从 SA 网站上获得的公开密钥;
- b) 如果上面的检查失败,DS 将不接受 SA 数字证书;如果 SA 数字证书合法,其所包含的数据服务公开密钥可以用于核实 ENC 签名文件。

8.4.7 更新 SA 数字证书的程序

SA 在下列情况下,将会出版和提供一个新的 SA 数字证书:

- a) 当 SA 数字证书过期时,证书不应包括已经变化的公开密钥;
- b) 当 SA 私有密钥出现安全问题时,SA 数字证书应包括一个新的公开密钥。

SA 将出版新的数字证书,并在 SA 网站上公布公开密钥。所有 DS 和 OEM 会立即得到通知,并收到新的数字证书的复制品。

DS 和 OEM 共同负责将新的 SA 数字证书以及新的 SA 公开密钥通知给 DC。

当一个新的 SA 数字证书或者公开密钥发布时,通常由该保护方案的所有用户执行该程序。按如下程序执行:

- a) 从 SA 网站上获得新的 SA 数字证书以及可印刷的 SA 公开密钥;
- b) 应用程序应下载新的 SA 数字证书,核对公开密钥是否与印刷的公开密钥一致。只有完成这些操作后,应用程序才认为该 SA 公开密钥是正确的;
- c) 使用新发布的证书替代存在的 SA 数字证书。

8.4.8 创建签名参数的程序

创建签名参数(PQG)的程序通常由 SA 和 DS 在创建公开和私有密钥对的时候执行。尽管由 DS 创建的 PQG 参数不必与包含在 SA 公开密钥和 SA 数字证书中的一致,但是使用的密钥长度应一致。

9 数字签名

9.1 数字签名文件格式

数字签名文件的格式与数字 DS 证书文件相同。数字签名(文件)采用 ASCII 文本格式,并按照规定的结构和次序表示。

数字签名文件应包含下面的 6 个数据字符串:

- a) 第一个数据字符串“R”,包括 10 块区域(每区域含 4 个字符),是数字签名中的“R”部分;
- b) 第二个数据字符串“S”,包括 10 块区域(每区域含 4 个字符),是数字签名中的“S”部分;
- c) 第三个数据字符串“P”,包括 32 块区域(每区域含 4 个字符),用来核实数字签名;
- d) 第四个数据字符串“p”,包括 10 块区域(每区域含 4 个字符),用来核实数字签名;

- e) 第五个数据字符串“g”,包括 32 块区域(每区域含 4 个字符),用来核实数字签名;
- f) 第六个数据字符串“y”,包括 32 块区域(每区域含 4 个字符),用来核实数字签名。

每一个数据字符串应遵守下述格式:

- a) 前面有一个标题行。标题行在开始处用“//”表示,后跟一个空格字符和 ASCII 文本标题字符;
- b) 用 16 进制数字(0~9,A~F)表示,任何字母字符均以大写表示;
- c) 以点号“.”表示结束;
- d) 每 4 个字符之间用空格字符分开;
- e) 每一个数据字符最后有一个行分割符(回车)(为了增加可读性,一些数据被分成两行)。

示例:

// Signature part R:

582F 9DE1 FCA6 A6A9 9BF7 CE24 72EF 9DE5 7613 B11C.

46FF 7302 FDF7 CBDF 2193 43B4 337C B6ED E146 E73E.

// BIGp

FCA6 82CE 8E12 CABA 26EF CCF7 110E 526D B078 B05E DECB CD1E B4A2 08F3 AE16

17AE 01F3 5B91 A47E 6DF6 3413 C5E1 2ED0 899B CD13 2ACD 50D9 9151 BDC4 3EE7

3759 2E17.

// BIGq

962E DDCC 369C BA8E BB26 0EE6 B6A1 26D9 346E 38C5.

// BIGg

6784 71B2 7A9C F44E E91A 49C5 147D B1A9 AAF2 44F0 5A43 4D64 8693 1D2D 1427

1B9E 3503 0B71 FD73 DA17 9069 B32E 2935 630E 1C20 6235 4D0D A20A 6C41 6E50

BE79 4CA4.

// BIGy

AA25 DF9E C3CA 96B7 9D01 3ED8 D572 D47C B3F3 80D0 731D EA47 B106 26BA C387 C1FA 3C33

EC55 6845 3744 76BE 5825 6E07 A74D 607F 7A5E 7B7E 3455 71D8 2110 4C8A C4BF.

9.2 ENC 数字签名文件格式

发布 ENC 信息的 DS 总是在 ENC 数字签名文件中创建签名/证书对。DC 需要核实签名文件中的签名/证书对。签名文件应包括签名和证书对。仅有签名的文件不能证明 DS 的身份,因此是非法的。

ENC 数字签名文件采用 ASCII 文本格式,并按照规定的结构和次序表示。

文件以下面的次序包含数据字符串:

- a) 第一个数据字符串“R”,包括 10 块区域(每区域含 4 个字符),是 ENC 签名中的数据服务商签名的“R”部分;
- b) 第二个数据字符串“S”,包括 10 块区域(每区域含 4 个字符),是 ENC 签名中的数据服务商签名的“S”部分;
- c) 第三个数据字符串“R”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥中的系统管理员签名的“R”部分;
- d) 第四个数据字符串“S”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥中的系统管理员签名的“S”部分;
- e) 第五个数据字符串“p”,包括 32 块区域(每区域含 4 个字符);
- f) 第六个数据字符串“q”,包括 32 块区域(每区域含 4 个字符);
- g) 第七个数据字符串“g”,包括 32 块区域(每区域含 4 个字符);
- h) 第八个数据字符串“y”,包括 32 块区域(每区域含 4 个字符),是 DS 的公开密钥。

头两个“R”和“S”数据字符串是 DS ENC 签名,文件剩下的部分与 DS 证书一致。

每个数据字符串应遵守下述格式:

- a) 前面有一个标题行。标题航在开始处用“//”表示,后跟一个空格字符和 ASCII 文本标题字符;
- b) 用 16 进制数字(0~9,A~F)表示,任何字母字符均以大写表示;
- c) 以点号“.”表示结束;
- d) 每 4 个字符之间用空格字符分开;
- e) 每一个数据字符最后有一个行分割符(回车)(为了增加可读性,一些数据被分成两行)。

第二个“R”和“S”用来核实 DS 数字证书(p,q,g,y 字符串)。如果核实成功,DS 公开密钥(y 字符串)可以被提取出来,用于核实加密的 ENC 的数字签名(头两个“R”和“S”)。允许 DC 核实系统管理员数字证书,提取 DS 公开密钥,核实 ENC 数据的数字签名。

示例:

```
// Signature      part R:
63E8 A27F 85FB 7553 C80C E201 64E0 FB6A E8FB 20CE.

// Signature      part S:
8A66 7CCC 24BA F358 CF3F BAA3 BE84 745B 5C3F 8E27.

// Signature      part R:
582F 9DE1 FCA6 A6A9 9BF7 CE24 72EF 9DE5 7613 B11C.

// Signature      part S:
46FF 7302 FDF7 CBDF 2193 43B4 337C B6ED E146 E73E.

// BIGP
FCA6 82CE 8E12 CABA 26EF CCF7 110E 526D B078 B05E DECB CD1E B4A2 08F3 AE16
17AE 01F3 5B91 A47E 6DF6 3413 C5E1 2ED0 899B CD13 2ACD 50D9 9151 BDC4 3EE7
3759 2E17.

// BIGq
962E DDCC 369C BA8E BB26 0EE6 B6A1 26D9 346E 38C5.

// BIGg
6784 71B2 7A9C F44E E91A 49C5 147D B1A9 AAF2 44F0 5A43 4D64 8693 1D2D 1427
1B9E 3503 0B71 FD73 DA17 9069 B32E 2935 630E 1C20 6235 4D0D A20A 6C41 6E50
BE79 4CA4.

// BIGy
AA25 DF9E C3CA 96B7 9D01 3ED8 D572 D47C B3F3 80D0 731D EA47 B106 26BA C387 C1FA
3C33 EC55 6845 3744 76BE 5825 6E07 A74D 607F 7A5E 7B7E 3455 71DB 2110 4C8A C4BF.
```

9.3 数字签名的创建程序

9.3.1 签名电子航道图文件的程序

签名电子航道图文件的程序由 DS 执行,以对其 ENC 数据文件进行数字签名。在签名前,ENC 文件应被压缩和加密。程序如下:

- a) 使用 SHA-1 算法散列所需的 ENC 文件(基本和更新文件)。进行散列以前 ENC 单元应进行压缩和加密。文件里面的所有字节都应被散列;
- b) 将 a) 中得到的散列值、DS 私有密钥和随机字符串传递给 DSA 数字签名算法,将会返回 2 个签名参数(“R”和“S”);
- c) 写在签名文件里面作为头两个数据的字符串应符合 9.2 所定义的格式。文件的剩下部分由 DS 证书组成,DS 证书包含与用于创建签名的私有密钥相关的公开密钥。

9.3.2 鉴定 ENC 数字签名文件的程序

9.3.2.1 鉴定 ENC 数字签名文件的程序,通常由 DS 为核实 ENC 数字签名而执行,前提是 DC 已经鉴别 SA 证书和签名文件中的 DS 证书。程序如下:

- a) 从与 ENC 文件唯一相关的签名文件中提取签名和证书,剩下的就是 DS 公开密钥文件;
- b) 从 a) 中的输出结果中提取签名部分(头两个数据字符串及其的相应部分),剩下的是证书;
- c) 使用 SHA-1 算法散列所需 ENC 文件,文件里面的所有字节都被散列;
- d) 核实签名部分,通过传递签名、DS 公开密钥及 ENC 文件的散列值给 DSA 数字签名算法,将返回正确与否。

9.3.2.2 如果 DC 签名鉴定不正确,数据用户将不能解密 ENC,如果 ENC 鉴定正确,则 ENC 能被解密。

10 加密

10.1 一般要求

10.1.1 每一个 ENC 单元文件使用一个单元密钥加密。同一个单元密钥可以用来加密 ENC 同一版本的所有更新文件。单元密钥以单元许可证的形式传送给 DC。

10.1.2 ENC 数据文件的所有内容应加密,用户许可证和单元许可证也应加密。

10.2 加密 ENC 基本单元文件的程序

10.2.1 加密 ENC 基本单元文件的程序应由 DS 执行,ENC 文件应在加密之前被压缩。程序如下:

- a) 选择加密所需的单元密钥(参照下面的条件);
- b) 使用 a) 中得到的单元密钥,利用 Blowfish 算法加密 ENC 文件,形成加密的 ENC 文件。

10.2.2 选择的加密密钥应遵守下面的条件:

- a) 产生 ENC 单元第一版本的 CK1 和 CK2,CK1 用于加密 ENC 单元;
- b) 对现有的 ENC 单元的新版本,应使用现存的 CK2 密钥加密 ENC 图。将现存的 CK2 变成 CK1,产生 CK2,用于 ENC 单元的下一版本;
- c) 对于 ENC 单元新发布版本应使用当前的 CK1 进行加密;
- d) 更新文件应使用与基本单元相同的单元密钥加密。

10.3 解密 ENC 基本单元文件的程序

解密 ENC 基本单元文件的程序通常由 DC 按如下步骤执行:

- a) 选择适当的 CK1 或 CK2 作为密钥;
- b) 使用 a) 中得到的密钥,利用 Blowfish 算法解密加密的 ENC 基本单元文件,从而产生解密后的 ENC 基本单元文件。

10.4 加密 ENC 更新文件的程序

加密 ENC 更新文件的程序通常由 DS 执行。加密以前应对 ENC 文件进行压缩。程序如下:

- a) 选择用于加密源 ENC 基本单元文件的密钥给更新文件应用;
- b) 将 a) 中得到的密钥作为密钥,利用 Blowfish 算法加密 ENC 更新文件(ENC 更新文件在加密之前应该压缩)。

10.5 解密 ENC 更新文件的程序

解密 ENC 更新文件的程序通常由 DC 按如下步骤执行：

- a) 选择适当的 CK1 或 CK2 作为密钥；
- b) 使用 a) 中得到的密钥,利用 Blowfish 算法解密加密的 ENC 更新文件,从而产生解密后的 ENC 更新文件。

10.6 使用基本许可证文件选择单元密钥的程序

使用基本许可证文件选择单元密钥的程序通常由 DC 执行,用于决定使用单元许可证提供的哪一个单元解密。该程序对 ENC 基本单元和更新单元均适用。使用单元许可证文件来解密时,需要 DC 的应用系统使用试错法来决定使用哪一个单元密钥,与使用元许可证文件解密不同。程序如下：

- a) 将 CK1 作为解密密钥,利用 Blowfish 算法,对 ENC 文件解密；
- b) 解压缩 ENC 文件,如果解压成功,则 ENC 文件被解密完毕,可以使用；
- c) 如果解密不成功,将 CK2 作为解密密钥,利用 Blowfish 算法,对 ENC 文件解密；
- d) 解压缩 ENC 文件,如果解压成功,则 ENC 文件被解密完毕,可以使用；
- e) 如果解密再次不成功,则单元许可证文件中不包括合法的单元密钥,DC 应从 DS 那里获得新的单元许可证。

10.7 使用元许可证文件选择单元密钥

10.7.1 该程序通常由 DC 执行,用于决定使用哪一个单元许可证提供的单元密钥解密。该程序对 ENC 基本单元和更新单元均适用。元许可证文件的使用将会减少 DC 应用程序选择适当的单元密钥的试错次数。程序如下：

- a) 从 CATALOG.031 文件中 CATD-COMT 字段获得 ENC 文件的版本；
- b) 从元许可证文件中的单元许可证记录中获得 EDTN 数字,EDTN 表明应用哪一版本的 CK1；
- c) 如果从 a) 和 b) 中得到的版本数字相同,CK1 应该用于解密 ENC。CK1 作为解密密钥,利用 Blowfish 算法,解密 ENC 基本单元文件；
- d) 如果从 a) 中获得的版本数字比 b) 中的低,则元许可证文件包含的单元密钥用于 ENC 的更新版本,DC 应从 DS 获得更新版本的电子航道图；
- e) 如果从 a) 中获得的版本数字比 b) 中的高,DC 可以尝试使用 CK2 解密 ENC。DC 也应从 DS 那里获得更新的单元许可证。DC 使用 CK2 作为解密密钥,利用 Blowfish 算法,解密 ENC 基本单元解压所得的 ENC 文件。如果解压成功,则 ENC 文件被解密完毕,可以使用。如果解密不成功,意味着单元许可证文件中不包括合法的单元密钥,DC 应从 DS 那里获得新的单元许可证。

10.7.2 ENC OEM 负责确保应用程序适当管理单元许可证。多个 DS 可以提供相关的单元许可证给一个 ENC 服务。因此应用程序应能够管理来自多个 DS 的单元许可证,也可以允许同一个 ENC 单元来源于多个 DS,来源于同一个 DS 的单元许可证不可能用来解密来源于另一个 DS 的 ENC。

前 言

JT/T 765《长江电子航道图制作规范》分为五个部分:

- 第1部分:术语;
- 第2部分:数据传输;
- 第3部分:显示准则;
- 第4部分:数据有效性检验;
- 第5部分:数据保护。

本部分为 JT/T 765 的第5部分。

本部分按照 GB/T 1.1—2009 给出的规则起草。

本部分代替 JT/T 765.5—2009《长江电子航道图制作规范 第5部分:数据保护》,与 JT/T 765.5—2009 相比主要技术变化如下:

- 删除了原标准中的部分示例(见2009年版的6.4等),并将部分示例转化为文字叙述(见6.4等)。

本部分对应于国际海道组织(IHO)S-63《IHO数据保护方案》(IHO Data Protection Scheme)。本部分与 IHO S-63 的一致性程度为非等效。

本部分由交通运输信息通信及导航标准化技术委员会提出并归口。

本部分起草单位:长江航务管理局、长江航道局、大连海事大学。

本部分主要起草人:但乃越、杜经农、朱业汉、杨大鸣、章娟、刘青、张娜、俞建林、王大彬、程大炜、赵德鹏、李源惠、宋良福、曹成、顾网林、李海、董华。

本部分所代替标准的历次版本发布情况为:JT/T 765.5—2009。

长江电子航道图制作规范

第5部分：数据保护

1 范围

JT/T 765 的本部分规定了长江电子航道图数据保护总体框架、各参与方的交互程序、用户许可证、单元许可证、数字证书、数字签名、加密等要求。

本部分适用于长江电子航道图制作、系统开发、设计和应用,其他内河电子航道图系统也可参照使用。

2 规范性引用文件

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

JT/T 765.1 长江电子航道图制作规范 第1部分:术语

3 术语和定义

JT/T 765.1 界定的术语和定义适用于本文件。

4 长江电子航道图数据保护总体框架

4.1 总体要求

4.1.1 长江电子航道图数据保护方案规定了四类参与方,分别为系统管理员、数据服务商、设备制造商、数据用户。

4.1.2 系统管理员为任何加入长江电子航道图数据保护方案的设备制造商分配一对唯一的制造商标识符(M_ID)和制造商密钥(M_KEY)值,并将所有设备制造商的 M_ID 和 M_KEY 记录下来,形成设备制造商信息表。设备制造商为购买其产品的数据用户分配一个唯一的硬件标识符(HW_ID),并用自己的 M_KEY 将其加密后形成用户许可证,随产品一同提供给数据用户。

4.1.3 对于任何加入长江电子航道图数据保护方案的数据服务商,系统管理员向其提供设备制造商信息表,并保证表内容的及时更新。

4.1.4 数据用户需要购买数据服务商的电子航道图时,向其提供自己的用户许可证,数据服务商用单元密钥对电子航道图数据进行加密,并利用用户许可证中的 HW_ID 加密单元密钥,形成单元许可证,将电子航道图和单元许可证发送给数据用户。数据用户收到电子航道图和单元许可证后,利用自己用户许可证中的 HW_ID 对单元许可证解密,得到单元密钥,并用单元密钥对电子航道图进行解密。

4.1.5 为在上述保护过程中实现身份认证,采用数字证书的方式来进行签名认证。系统管理员创建自己的系统管理员数字证书,并将该证书发布给设备制造商、数据服务商和数据用户。数据服务商创建自己的数据服务商自我签名密钥文件(SSK)并提交给系统管理员,系统管理员核实 SSK 文件后,用自己的私有密钥对数据服务商公开密钥进行签名,生成数据服务商数字证书,并发布给数据服务商。数据服务商在向数据用户发送电子航道图前,先用自己的私有密钥对电子航道图签名,并和数据服务商数字证

书结合,形成电子航道图的签名文件,随电子航道图一起提供给数据用户。数据用户收到电子航道图签名文件后,利用系统管理员数字证书中的公开密钥来核实数据服务商证书,再利用数据服务商证书中的公开密钥来核实电子航道图的签名,如果正确,则电子航道图可以使用,否则因不能确认来源而拒绝使用。

4.2 各参与方的职责

4.2.1 系统管理员

系统管理员(SA)应根据需要维护策略的正常运行,制作并发布方案根证书,接受设备制造商(OEM)的申请,并为符合要求的 OEM 创建 M_ID 和 M_KEY 信息。SA 还应为数据服务商(DS)发放数字证书,在电子航道图发布过程中实现身份认证。SA 的主要职责是:

- a) 确认 OEM 发送的 M_ID、M_KEY 申请表格;
- b) 确认 DS 发送的数据服务商申请表格;
- c) 与 OEM 签订协议书;
- d) 与 DS 签订协议书;
- e) 为每个 OEM 创建——对应的 M_ID 和 M_KEY;
- f) 为每个 DS 创建数据服务商证书;
- g) 创建系统管理员数字证书以提供给电子航道图显示与信息系统(ECDIS)用户;
- h) 负责管理各种文档,并将系统成员所需文档发送给成员。

4.2.2 数据服务商

数据服务商(DS)应负责加密和签署电子航道图(ENC)信息,通过网络或者数据光盘的形式为用户提供更新 ENC 服务。每个长江电子航道图数据提供商应符合国家相关的资质规定,并经 SA 认证通过,具有向用户发布数据的资格。DS 的主要职责是:

- a) 向 SA 申请发布 ENC 数据的资格;
- b) 与 SA 签署协议,由 SA 向其分配 M_ID 和 M_KEY;
- c) 创建自己的 SSK 证书文件;
- d) 鉴定 ENC 请求用户的合法性;
- e) 创建加密 ENC 文件所需的单元密钥;
- f) 创建单元许可证文件和元数据文件;
- g) 压缩、加密以及签署 ENC 并发送给用户;
- h) 产生单元许可证文件和元数据文件,发送给请求用户;
- i) 管理加密 ENC 的密钥以及其他用户资料等文档文件。

4.2.3 设备制造商

设备制造商(OEM)主要负责开发一套电子航道图处理系统(EPS),该系统按照本规范要求具备对 ENC 数据的保护功能。同时,还具备处理保护的 ENC 数据,以便在 ECDIS 平台显示正确的电子航道图数据信息的能力。为了加入长江电子航道图数据保护方案,相应的 OEM 应完成方案的要求,取得资格认证后才能加入本保护方案,并且获取 M_ID 和 M_KEY 来实现他们的系统,提供支持该安全方案的船舶导航产品。OEM 的主要职责是:

- a) 申请制造商资格;
- b) 与 SA 签署协议;
- c) 为每台设备创建 HW_ID;

- d) 将每个 HW_ID 记录在 HW_ID 登记表中,并且进行管理;
- e) 开发符合本规范要求的 EPS 系统;
- f) 为电子航道图显示与信息系统(ECDIS)创建用户许可证文件。

4.2.4 数据用户

数据用户(DC)是电子航道图信息的使用者,从 DS 那里接收加以保护的 ENC 信息。ECDIS 用户负责鉴别 ENC 的数字签名和按照保护方案定义的程序对 ENC 进行解密。DC 的主要职责是:

- a) 从 SA 网站下载系统管理员数字证书,用以鉴别 DS 的合法性;
- b) 向 DS 提供用户许可证文件,以获得新的 ENC 文件;
- c) 管理多个 DS 各自的用户许可证文件。

5 各参与方的交互程序

5.1 SA 创建自己的数字证书

SA 创建自己的私有密钥和公有密钥,使用私有密钥对公有密钥进行签名,将签名算法的返回值与公有密钥结合,创建 SA 数字证书。

5.2 DS 加入数据保护体系的基本程序

DS 加入数据保护体系的基本程序如下:

- a) DS 创建自己的公开密钥、私钥对;
- b) DS 创建 SSK 文件;
- c) DS 填写“长江电子航道图数据保护系统数据服务商证书请求表”;
- d) DS 将申请表、SSK 文件以及数据服务商公开密钥文件发送给 SA;
- e) SA 通过电话、电子邮件等方式对申请表格和 SSK 文件的来源等进行确认;
- f) SA 核实 SSK 文件的格式是否符合本规范要求;
- g) SA 利用 DS 发送的公开密钥文件核实 SSK 文件;
- h) 如果对 SSK 文件的鉴别合法,且该 DS 符合国家相关资质要求,SA 创建 DS 证书,否则,给请求的 DS 发出错误信息。

5.3 OEM 加入数据保护体系的基本程序

OEM 加入数据保护体系的基本程序如下:

- a) OEM 完成 M_ID/M_KEY 申请表的填写(主要包括 OEM 的详细信息以及请求所需文件);
- b) SA 核实 M_ID/M_KEY 表格第一部分(主要核实设备制造商对表格的填写是否完整);
- c) SA 核实 OEM 是否签署“长江电子航道图数据保护协议书”;
- d) SA 核实 OEM 开发的系统是否符合本规范标准;
- e) SA 核实 OEM 没有 M_ID/M_KEY 对(主要确认该申请的 OEM 未曾有过 M_ID/M_KEY,即表明申请者是新的 OEM);
- f) SA 创建 M_ID/M_KEY,并且存储到 M_ID/M_KEY 登记表内;
- g) SA 将 M_ID/M_KEY 发送给 OEM,同时也将这个新的 M_ID/M_KEY 发送给系统内所有的 DS。

5.4 DC 向 OEM 购买新设备的程序

DC 按如下程序向 OEM 购买设备:

- a) DC 向 OEM 递交标识身份的用户许可证文件(主要是申请 ENC 文件);

- b) OEM 生成一份包含 DC 硬件设备标识 HW_ID 和 M_ID 信息的用户许可证文件;
- c) OEM 将研制生产的 EPS 系统和新生成的用户许可证文件以及 ECDIS 系统一起发送给 DC。

5.5 SA 与 DC 的交互程序

DC 按如下程序从 SA 处获得并验证数字证书:

- a) DC 从 SA 获取 SA 数字证书;
- b) 对 SA 数字证书格式进行核对;
- c) 利用 SA 公开密钥鉴别 SA 数字证书(比较数字证书,包括 SA 公开密钥与从 SA 网站获得的公开密钥是否一致)。

5.6 DC 购买并使用 DS 提供的电子航道图的程序

DC 购买并使用 DS 提供的电子航道图的程序如下:

- a) DC 向 DS 递交用户许可证以申请电子航道图文件;
- b) DS 从用户许可证中获得 M_ID,并查对从 SA 那里得到的 M_ID/M_KEY 登记表,找到与该 M_ID 对应的 M_KEY;
- c) DS 使用 M_KEY 解密用户许可证,获得 DC 的 HW_ID,用来加密单元密钥;
- d) DS 生成单元许可证文件;
- e) DS 用 zip 算法压缩电子航道图单元数据,并用单元密钥加密压缩后的数据;
- f) DS 利用自己的私有密钥对加密后的电子航道图单元数据进行数字签名;
- g) DS 将加密后的电子航道图单元数据、单元许可证文件和数字签名文件一起发送给 DC;
- h) DC 核对 DS 发送的相关文件(主要包括签名文件是否包含签名和证书对,核实签名文件是否符合本规范要求的格式);
- i) DC 鉴别签名文件中的 DS 证书是否合法。如果合法,从签名文件中提取 DS 公开密钥;
- j) DC 鉴别电子航道图单元数据的签名文件;
- k) DC 通过系统软件提取自己机器上的 HW_ID 来解密单元许可证,得到电子航道图数据文件的单元密钥,利用系统解密数据并使用。

6 用户许可证

6.1 M_ID 的格式

M_ID 是 4 位长度的 16 进制数字(两字节),以 ASCII 码表示。SA 应向每个加入长江电子航道图数据保护体系的设备制造商分配一对 M_ID/M_KEY。

6.2 M_KEY 的格式

M_KEY 是 10 位长度的 16 进制数字(五字节)。

6.3 HW_ID 的格式

HW_ID 是五字节的十进制数字,由 OEM 分配给购买其系统的 DC。OEM 应为每一个系统指定一个唯一的 HW_ID,建议 HW_ID 以不连续的方式来分配。在用户许可证中 HW_ID 以加密的形式存在。HW_ID 用 M_KEY 作为加密密钥,利用 Blowfish 算法进行加密。加密以后的 HW_ID 是八字节长的十进制数,被转化成 16 位长度的 16 进制数,以 ASCII 码表示。

6.4 用户许可证的格式

用户许可证为 28 字符长度,用 ASCII 文本书写,其格式和字段长度要求见表 1。

表 1

加密后的 HW_ID	校验和	M_ID
16 位 16 进制数	8 位 16 进制数	4 位 16 进制数

表 1 中每个字段中的 16 进制数均以 ASCII 码表示,其中任何字母字符应大写。用户许可证中的校验和是用 CRC32 算法对加密后的 HW_ID 进行散列计算后得到。该散列计算结果被转化为 8 位 16 进制数。

示例:
用户许可证的最终表现形式为:73871727080876A07E450C043031。

6.5 用户许可证的创建程序

用户许可证的创建程序由系统或者应用程序提供商执行,生成的用户许可证发送给 DC,用作 DC 请求单元许可证时的自身标识。用户许可证和系统一起交给终端用户。用户许可证的创建过程如下:

- a) M_KEY 作为密钥,利用 Blowfish 算法对 HW_ID 进行加密;
- b) 将结果转换成 16 位 16 进制字符串,每一个字母都应大写表示;
- c) 利用 CRC32 算法散列上面的 16 位 16 进制字符串;
- d) 将 c) 输出的结果转化为 8 位 16 进制字符串,每一个字母都应大写表示,即 Check Sum(校验和);
- e) 将 d) 输出的结果附加在 b) 的输出结果;
- f) 将 M_ID 转化为 4 位 16 进制字符串,任何字母应以大写表示;
- g) 将 f) 的输出结果附加到 e) 的输出结果的后面,即用户许可证。

示例:
用户许可证的标识符及内容参见表 2,创建程序见表 3。

表 2

标 识 符	内 容	备 注
HW_ID	3132333438	16 进制表示的 HW_ID
M_KEY	3938373635	16 进制表示的 M_KEY
M_ID	3031	16 进制表示的 M_ID

表 3

步 骤	内 容	备 注
步骤 a) 的输入	3132333438 和 3938373635	16 进制表示的 HW_ID 和 M_KEY
步骤 a) 的输出	八字节	不可打印
步骤 c) 的输入	73871727080876A0	这是步骤 a) 输出的 16 进制字符串表现形式。这个字符串从左到右输入给散列算法,例如 73、87、17……
步骤 c) 的输出	7E450C04	用 16 进制表示的 CRC32 散列结果
步骤 e) 的输出	73871727080876A07E450C04	将 CRC32 散列结果附加在加密后的 HW_ID 后
用户许可证	73871727080876A07E450C043031	

6.6 用户许可证的解密程序

用户许可证的解密程序由 DS 的系统运行。用户许可证的结构在 6.4 中定义。用户许可证的解密程序如下：

- a) 从用户许可证中提取 M_ID(4 位 16 进制字符)；
- b) 从用户许可证中提取校验和(Check Sum,为 8 位 16 进制字符)；
- c) 利用 CRC32 算法散列加密的 HW_ID(用户许可证中前 16 位字符)；
- d) 比较 b)和 c)的输出,如果相同,则该用户许可证是合法的。如果不同,则该用户许可证是不合法的,从而不能获得相应的 HW_ID；
- e) 如果用户许可证是合法的,把这个加密的 HW_ID 转换成八字节；
- f) 将 M_KEY 作为解密密钥,利用 Blowfish 算法解密这个加密的 HW_ID,所得结果就是 HW_ID。

示例：

用户许可证的解密程序参见表 4 和表 5。

表 4

标 识 符	内 容	备 注
用户许可证	73871727080876A07E450C043031	
M_KEY	3938373635	16 进制的 M_KEY

表 5

步 骤	内 容	备 注
步骤 a)的输出	3031	提取的 M_ID
步骤 b)的输出	7E450C04	提取出来的校验和
步骤 c)的输入	73871727080876A0	字节从左到右输出给散列函数,例如 73、87、17
步骤 c)的输出	7E450C04	从加密的 HW_ID 中提取出来的校验和
步骤 f)的输出	3132333438	16 进制表示的 HW_ID

7 单元许可证

7.1 一般要求

7.1.1 电子航道图单元数据在发布前应由 DS 使用单元密钥进行加密,该单元密钥是与该电子航道图单元数据唯一对应的。DC 应获得这个密钥才能解密数据。这个单元密钥由 DS 以加密表格的形式——单元许可证提供给 DC。一个单元许可证文件可以包括一个或者多个电子航道图的单元密钥。

7.1.2 单元许可证的发布针对指定的 HW_ID,在安装系统之间是不可转换的。加密方案支持能在所有客户之间相互传送的普通加密 CD 形式。

7.1.3 DS 通过其得到的用户许可证获得用户的 HW_ID,通过用户的 HW_ID 加密电子航道图单元密钥从而形成单元许可证,确保单元许可证不能在 DC 之间进行传送。终端用户利用其 HW_ID 对单元许可证进行解密,以便获得解密电子航道图所需的单元密钥。

7.1.4 发布单元许可证需要两个文件：

- a) 包含解密电子航道图所需本质信息的基本许可证文件,在 7.5 中定义；

b) 包含易于数据管理和解密的补充信息的元许可证文件,在 7.6 中定义。

7.1.5 客户系统应负责管理来自多个 DS 的权证文件。来源于某个 DS 的权证文件不能用来解密来自另一个 DS 的电子航道图。

7.1.6 一个单元许可证包括两个加密的单元密钥,即包含当前版本电子航道图解密密钥的单元密钥 1 (ECK1),以及用作下一版本电子航道图单元的解密密钥 ECK2。单元许可证里面的 ECK2 使得 DS 能周期性地改变用于电子航道图下一版本的单元密钥,而无须对其他 DC 发布新的单元许可证。

7.1.7 除了这些密钥以外,单元许可证还包括过期日期(这个日期之后的更新文件不能使用)和单元标识(便于用户在有好几个单元和权证文件的情况下,系统能够将两者相匹配)。

7.1.8 终端用户将单元许可证里面的过期日期与需要解密的电子航道图单元进行比较,如果需要解密的电子航道图单元或者更新文件的生产日期在许可证的过期日期之后,系统将不会使用单元许可证来解密数据,以防止用户使用过期的电子航道图单元文件或者更新文件。

7.1.9 电子航道图的基本单元或更新文件的发布日期编码包含在电子航道图数据集文件的 DSID-IS-DT 子字段里。这些数据文件总是加密的,除非已经解密,否则无法得到发布日期。为了提供一种获得发布日期而不需要解密这些电子航道图文件的简便机制,DSID-ISDT 字段的内容应被复制到数据集所含文件 CATALOG.031 的 CATD-COMT 字段中,以保障系统能快速检查这个电子航道图信息是否在许可证过期日期之后,而无须解密电子航道图。

示例:

CATALOG.031 文件中的 CATD-COMT 编码如下:

VERSION = 1.0,EDTN = 10,UPDN = 0,UADT = 20030224,ISDT = 20030224。

7.1.10 单元许可证以 ASCII 文本的形式书写,任何字母字符都应大写表示,其格式和字段长度见表 6。

表 6

航道图单元名称	过期日期	加密后的 CK1	加密后的 CK2	加密后的校验和
8 个字母或数字	8 个数字	16 位 16 进制数	16 位 16 进制数	16 位 16 进制数

示例:

单元许可证中各字段数据参见表 7。

表 7

航道图单元名称	过期日期	加密后的单元密钥 1 (ECK1)	加密后的单元密钥 2 (ECK2)	加密校验和
NO4D0613	20000830	BEB9BFE3C7C6CE68	B16411FD09F96982	795C77B204F54D48

单元许可证最终的表现形式为:NO4D061320000830BEB9BFE3C7C6CE68B16411FD09F96982795C77B204F54D48。

7.2 电子航道图单元名称与过期日期的格式

电子航道图单元名称是八字节的字母数字串,遵循 JT/T 765.2—2016 附录 B 中所定义的单元文件命名规则。

示例:

如表 7 所示,航道图单元名称表示为:NO4D0613。

过期日期的格式应为:年月日(YYYYMMDD)。

示例:

如表 7 所示,过期日期 2000 年 8 月 30 日表示为:20000830。

7.3 加密后的单元密钥(ECK1,ECK2)格式

7.3.1 单元密钥是五字节随机数字,使用从用户许可证中获得的数据用户的 HW_ID 进行 Blowfish 加密,然后转化成 16 进制。Blowfish 加密算法将需要加密的数据长度填充成八字节的倍数,加密后单元密钥从五字节长度(10 个 16 进制字符)变为八字节长度(16 个 16 进制字符)。

7.3.2 ECK1 包括电子航道图单元的当前版本的单元密钥。ECK2 包括用于下一版本电子航道图的单元密钥。

7.3.3 ECK1 用于可从 DS 获得的电子航道图的当前版本。如果电子航道图不能正确解密,应用 ECK2 解密。如果 ECK2 也不能正确解密,则应从 DS 获得更新的单元许可证。

示例:

如表 7 所示,加密后的 ECK1、ECK2 为:

- a) ECK1: BEB9BFE3C7C6CE68;
- b) ECK2: B16411FD09F96982。

7.4 加密校验和的格式

对单元许可证中加密校验和之前的所有字段用 CRC32 算法进行散列计算,然后用 HW_ID 对散列计算的结果进行加密,就得到了加密校验和。加密校验和是 16 位的 16 进制数字。

示例:

如表 7 所示,加密校验和为:795C77B204F54D48。

7.5 单元许可证的传输

单元许可证应存储在名称为 ENC. PMT 的文件里面。ENC. PMT 文件包括一个或多个作为连续记录存储在该文件中的单元许可证。

示例:

ENC. PMT 文件包括 2 个单元许可证:

- a) NO4D061320000830BEB9BFE3C7C6CE68B16411FD09F96982795C77B204F54D48;
- b) GB30401020000830698FFC77EC13C2FF442934F3563E50A4A858D1F973860701。

7.6 元许可证文件的定义

7.6.1 元许可证文件总体格式

元许可证文件(PERMIT. TXT)是针对终端用户开发者的,用来通知 DC 的征订状态,以及帮助开发者理解哪些基于客户征订结束日期和电子航道图版本日期的航道图单元需要解密。

在元许可证文件中任何字母的字符都应大写。DS 应提供如何从存储介质或网络中获得元许可证文件的方法。

元许可证文件中各部分的内容和格式见 7.6.2 ~7.6.6。

示例:

元许可证文件(PERMIT. TXT)的格式为:

```
:DATE 20000131 11:11
:VERSION 1
:ENC
GB30401020010110797C4CB45C5B4B7B0BC8A74680E798F2910351530D1FF963,0,2,
GB40401S200101100D0CC631D5B07B04252DAC4A7EC568BFD2C08583E26FF27E,0,3,
GB50401S200101100C9EAA1802FD92628086CE08A71125104F9FDFBE6C9656B5,0,3,
:ECS
```

7.6.2 日期和时间

字段名称: :DATE。
日期格式: YYYMMDD。
时间格式: HH:MM,使用 24h 制。
字段名称、日期和时间用空格字符分隔。
示例:
:DATE 20000131 11:11

7.6.3 版本

字段名称: :VERSION。
版本数字是一个整数,用来定义元许可证文件的格式版本数字。
字段名称与版本数字用空格字符分隔。
示例:
:VERSION 1

7.6.4 元许可证类型

字段名称: :ENC。
字段包含来自 DS 的 ENC 发布证书中的可用元许可证的定义。
字段里面包含一个或多个 7.6.5 中定义的单元许可证记录。

7.6.5 单元许可证记录

单元许可证记录的字段组成见表 8。

表 8

单元许可证记录的字段	备 注
单元许可证	在 7.1.10 中定义
服务级别指示	0 – 定期订购的许可证 1 – 单次购买的许可证
EDTN	电子航道图单元的 DSID-EDTN 参数
Reserved	为将来应用保留的参数字段
Comment	用于注释的文本字段

7.6.6 单元许可证类型 ECS

字段名称: :ECS。
字段包含来自 DS 的电子航道图系统的发布证书中的可用元许可证的定义。
字段里面包含一个或者多个 7.6.5 中定义的单元许可证记录。

7.7 创建单元许可证的程序

创建单元许可证的程序通常由 DS 执行,用于给特定 DC 创建单元许可证。
以下程序产生符合 7.1 中定义的单元许可证:
a) 移除电子 ENC 文件名称中的文件扩展名,剩下 8 个字符是单元许可证中的单元名称;

- b) 将过期日期,格式为 YYYYMMDD,附加在 a) 中得到的单元名称的后面;
- c) 将 HW_ID 的第一个字节附加到 HW_ID 的后面形成六字节的 HW_ID(称作 HW_ID6),HW_ID6 就是所创建的用于加密单元密钥的一个 48 位密钥;
- d) 使用 c)中得到的 HW_ID6 作为加密密钥,利用 Blowfish 算法对单元密钥进行加密,以创建 ECK1;
- e) 将 ECK1 转换为 16 位 16 进制字符,任何字母字符都应以大写表示;
- f) 将 e)中的结果附加到 b)的后面;
- g) 将 HW_ID6 作为密钥,利用 Blowfish 算法,对单元密钥 2(CK2)进行加密,从而创建 ECK2;
- h) 将 ECK2 转换为 16 位 16 进制字符,任何字母字符都应以大写表示;
- i) 将 h)的输出结果附加到 f)的后面;
- j) 利用 CRC32 算法对 i)的输出结果进行散列计算,注意这里的散列计算是在转化为 16 进制字符之后,而用户许可证的散列计算是针对原始的二进制数据;
- k) HW_ID6 作为密钥,利用 Blowfish 算法对 j)中的散列结果进行加密;
- l) 将 k)中的输出结果转换为 16 位 16 进制字符串.任何字符字母都应大写,形成 ENC 的校验和;
- m) 将 l)的输出结果附加到 i)的后面,形成单元许可证。

示例:
创建单元许可证的程序参见表 9 和表 10。

表 9

标识符	内 容	备 注
HW_ID	3132333438	五字节的 16 进制字符
CK1	C1CB518E9C	五字节的 16 进制字符
CK2	421571CC66	五字节的 16 进制字符
Cell Name	NO4D0613.000	符合 S-57 标准(版本 3.1)的电子航道图单元文件名
Expiry Date	20000830	过期日期

表 10

步 骤	内 容	备 注
步骤 a)的输出	NO4D0613	电子航道图单元名称
步骤 b)的输出	NO4D061320000830	电子航道图单元名称 + 过期日期
步骤 c)的输出	313233343831	16 进制字符表示的 HW_ID6
步骤 d)或 e)的输出	BEB9BFE3C7C6CE68	16 进制字符表示的 ECK1
步骤 f)的输出	NO4D061320000830BEB9BFE3C7C6CE68	电子航道图单元名称 + 过期日期 + ECK1
步骤 g)或 h)的输出	B16411FD09F96982	16 进制字符表示的 ECK2
步骤 i)的输出	NO4D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982	电子航道图单元名称 + 过期日期 + ECK1 + ECK2
步骤 j)的输入	NO4D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982	步骤 i)输出的 ASCII 值,这些字节从左到右输入给散列函数,例如 NO、4D、06……
步骤 j)的输出	780699093	CRC32 散列计算的结果
步骤 k)的输出	八字节的不可打印值	加密后的 CRC32 计算结果
步骤 l)的输出	795C77B204F54D48	16 进制字符表示的 CRC32 计算结果
单元许可证	NO4D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982795C77B204F54D48	

7.8 核查单元许可证文件的程序

核查单元许可证文件的程序通常在 DC 系统内执行,由以下步骤组成:

- a) 单元许可证在文件 ENC. PMT 和 PERMIT. TXT 里面提供;
- b) DC 应用程序可以将单元许可证文件存储在任何适当位置。应确保应用于同一个 ENC 数据的许可证文件由同一个 DC 提供;
- c) 单元许可证中的前 8 个字符代表相关的 ENC 单元的单元名称(去掉扩展名字);
- d) ENC. PMT 或 PERMIT. TXT 文件可以包含几个许可证文件。

7.9 核实单元许可证 ENC 校验和的程序

核实单元许可证 ENC 校验和的程序通常由 DC 系统执行,由以下步骤组成:

- a) 从单元许可证提取最后的 16 位 16 进制字符(ENC 校验和);
- b) 将上面的 16 位 16 进制字符转化为八字节;
- c) 利用 CRC32 算法散列执行 a) 以后所剩下的字符串;
- d) 将 HW_ID 的第一个字节附加到 HW_ID 的后面,形成六字节的 HW_ID(称作 HW_ID6);
- e) 以 HW_ID6 为密钥,利用 Blowfish 算法加密 c) 中的散列结果;
- f) 比较 b) 和 e) 的结果。如果相同,则此单元许可证是合法的。如果不相同,则此单元许可证非法,不能使用。

示例:

核实单元许可证 ENC 校验和的程序参见表 11 和表 12。

表 11

标识符	内 容	备 注
HW_ID	3132333438	16 进制字符表示的 HW_ID
单元许可证	NO4D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982795C77B204F54D48	

表 12

步 骤	内 容	备 注
步骤 a) 的输出	795C77B204F54D48	16 进制字符
步骤 b) 的输出	八字节的不可打印值	加密后的 CRC32
步骤 c) 的输入	NO4D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982	去掉 CRC32 值后的单元许可证
步骤 c) 的输出	780699093	CRC32 散列计算结果
步骤 d) 的输出	313233343831	HW_ID6
步骤 e) 的输出	八字节的不可打印字符	加密后的 CRC32

7.10 解密单元许可证中单元密钥的程序

这个程序通常由 DC 执行,以提取用以解密电子航道图的单元密钥,由下面的几个步骤组成:

- a) 将 HW_ID 的第一个字节附加到 HW_ID 的后面,形成六字节的 HW_ID(称作 HW_ID6);
- b) 从单元许可证中提取出来 ECK1,并且将之从 16 位 16 进制字符转化为八字节;

- c) HW_ID6 作为密钥,利用 Blowfish 算法对 b) 的输出结果 ECK1 进行解密,形成单元密钥 1 (CK1);
- d) 从单元许可证中提取出来 ECK2,并且将之从 16 位 16 进制字符转化为八字节;
- e) HW_ID6 作为密钥,利用 Blowfish 算法对 d) 的输出结果 ECK2 进行解密,形成单元密钥 2 (CK2)。

示例:

解密单元许可证中单元密钥的程序参见表 13 和表 14。

表 13

标识符	内 容	备 注
HW_ID	3132333438	16 进制字符表示的 HW_ID
单元许可证	NO4D061320000830BEB9BFE3C7C6CE68 B16411FD09F96982795C77B204F54D48	

表 14

步 骤	内 容	备 注
步骤 a) 的输出	313233343831	HW_ID6
步骤 b) 的输出	八字节不可打印值*	加密后的 ECK1
步骤 c) 的输出	C1CB518E9C	单元密钥 1 的 16 进制字符
步骤 d) 的输出	八字节不可打印值	加密后的 ECK2
步骤 e) 的输出	421571CC66	单元密钥 2 的 16 进制字符
*没有经过加密的单元密钥是五字节长度,但经过加密以后的单元密钥在长度上是八字节。这是因为 Blowfish 在加密时,将单元密钥填充到八字节长度,而当对其解密时对之进行减充。		

8 数字证书

8.1 SA 公开密钥和 DS 公开密钥的格式

SA 公开密钥和 DS 公开密钥都应以适当的文件格式提供。公开密钥(文件)采用 ASCII 文本格式,并按照规定的结构和次序表示。

文件应包括下面 4 个数据字符串:

- a) 第一个数据字符串“p”,共有 32 个字符串块,每块字符串由 4 个字符组成;
- b) 第二个数据字符串“q”,共有 10 个字符串块,每块字符串由 4 个字符组成;
- c) 第三个数据字符串“g”,共有 32 个字符串块,每块字符串由 4 个字符组成;
- d) 第四个数据字符串“y”,共有 32 个字符串块,每块字符串由 4 个字符组成,是 DS 公开密钥。

每一个数据字符串应遵守下述格式:

- a) 前面有一个标题行。标题行在开始处用“//”表示。标题线总是以这种次序出现;
- b) 用 16 进制表示(0~9,A~F),任何字母字符均以大写表示;
- c) 以点号“.”表示结束;
- d) 每 4 个字符之间用空格分开;
- e) 每一个数据字符最后有一个行分割符(为了增加可读性,一些数据被分成两行)。

示例:

// BIGp

D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.

// BIGq

8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.

// BIGg

B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710
E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.

// BIGy

444B BA17 1758 0DAF 71AB 52A5 6CCA 8EAB 4C51 E970 0E37 B17B BB46 C0B9 4A36 F73F
0244 7FBD AE5B 7CA9 3870 5AB9 E9EE 471C E7B0 1004 6DF1 3505 42B3 0332 AE67 69C6.

8.2 自我签名密钥格式

自我签名密钥由 DS 创建,然后提交给 SA 以获得 DS 证书。自我签名密钥的格式与 DS 证书的格式(见 8.3)相同,本质的区别是:自我签名密钥由 DS 本身创建,而 DS 证书由 SA 创建和发布。自我签名密钥(文件)采用 ASCII 文本格式,并按照规定的结构和次序表示。

自我签名密钥文件应包括下面 6 个数据字符串:

- 第一个数据字符串“R”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥的 DS 签名中的“R”元素;
- 第二个数据字符串“S”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥的 DS 签名中的“S”元素;
- 第三个数据字符串“p”,包括 32 块区域(每区域含 4 个字符);
- 第四个数据字符串“q”,包括 10 块区域(每区域含 4 个字符);
- 第五个数据字符串“g”,包括 32 块区域(每区域含 4 个字符);
- 第六个数据字符串“y”,包括 32 块区域(每区域含 4 个字符),是 DS 公开密钥。

每一个数据字符串应遵守下述格式:

- 前面有一个标题行。标题行在开始处用“//”表示,后跟一个空格字符和 ASCII 文本标题字符;
- 用 16 进制数字(0~9, A~F)表示,任何字母字符均以大写表示;
- 以点号“.”表示结束;
- 每 4 个字符之间用空格字符分开;
- 每一个数据字符串最后有一个行分割符(回车)(为了增加可读性,一些数据被分成两行)。

示例:

// Signature part R:

752A 8E5C 3AF5 6CCD 7395 B52E F672 E404 554F AAB6.

// Signature part S:

1756 E5C0 F4B6 BC90 4EC6 5F94 DF93 3ADF 68B8 86C4.

// BIGp

D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12
DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.

// BIGq

8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.

// BIGg

B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710

E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.

// BIGy

444B BA17 1758 0DAF 71AB 52A5 6CCA 8EAB 4C51 E970 0E37 B17B BB46 C0B9 4A36 F73F

0244 7FBD AE5B 7CA9 3870 5AB9 E9EE 471C E7B0 1004 6DF1 3505 42B3 0332 AE67 69C6.

8.3 DS 证书的格式规则

DS 使用 512 字节的数字签名算法公开密钥。DS 证书(文件)采用 ASCII 文本格式,并按照规定结构和次序表示。

DS 文件应包括下面 6 个数据字符串:

- 第一个数据字符串“R”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥文件里的 SA 签名的“R”元素;
- 第二个数据字符串“S”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥文件里的 SA 签名的“S”元素;
- 第三个数据字符串“p”,包括 32 块区域(每区域含 4 个字符),是 DS 公开密钥里面的“p”元素;
- 第四个数据字符串“q”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥里面的“q”元素;
- 第五个数据字符串“g”,包括 32 块区域(每区域含 4 个字符),是 DS 公开密钥里面的“g”元素;
- 第六个数据字符串“y”,包括 32 块区域(每区域含 4 个字符),是 DS 公开密钥里面的“y”元素。

每一个数据字符串应遵守下述格式:

- 前面有一个标题行。标题行在开始处用“//”表示,后跟一个空格字符和 ASCII 文本标题字符;
- 用 16 进制数字(0~9,A~F)表示,任何字母字符均以大写表示;
- 以点号“.”表示结束;
- 每 4 个字符之间用空格字符分开;
- 每一个数据字符最后有一个行分割符(回车)(为了增加可读性,一些数据被分成两行)。

示例:

// Signature part R:

8FD6 2AC7 27D2 8D0B CD27 BDF2 5CC6 9656 10E3 751F.

// Signature part S:

3DE7 DA37 5A40 80FC 4203 5C6E 37DE A984 2A88 2BDC.

// BIGp

D0A0 2D76 D210 58DA 4D91 BBC7 30AC 9186 5CB4 036C CDA4 6B49 4650 16BB 6931 2F12

DF14 A0CC F38E B77C AD84 E6A1 2F2A A0D0 441A 734B 1D2B E944 5D10 BA87 609B 75E3.

// BIGq

8E00 82E3 C046 DFE6 C422 F44C C111 DBF6 ADEE 9467.

// BIGg

B08D 786D 0ED3 4E39 7C6B 3ACF 8843 C3BF BAB1 A44D 0846 BB2A C3EE D432 B270 E710

E083 B239 AF0E A5B8 693B F2FC A03B 6A73 E289 84FF 8623 1394 996F 6263 0845 AA94.

// BIGy

444B BA17 1758 0DAF 71AB 52A5 6CCA 8EAB 4C51 E970 0E37 B17B BB46 C0B9 4A36 F73F

0244 7FBD AE5B 7CA9 3870 5AB9 E9EE 471C E7B0 1004 6DF1 3505 42B3 0332 AE67 69C6.

8.4 数字证书的创建程序

8.4.1 生成 SSK 的程序

生成 SSK 的程序通常由数据服务商执行一次,以便创建其自身的 SSK,然后发送到 SA,由 SA 根据该 SSK 创建 DS 证书。程序如下:

- 使用算法 SHA-1 散列公开密钥文件,文件里面的所有字节都应进行散列;
- 利用 DSA 数字签名算法通过私有密钥,公开密钥的文件的散列和一个随机字符串实现对公开密钥文件进行签名,这将返回 2 个签名元素(“R”和“S”);
- 将这个自我签名密钥文件按 8.2 规定的格式书写,然后将公开密钥文件附加到它的后面,形成自我签名密钥文件。

8.4.2 鉴别 SSK 的程序

鉴别 SSK 的程序通常由 SA 在创建和发布 DS 证书之前执行。程序如下:

- 提取数字签名元素“R”和“S”,剩余为公开密钥文件内容;
- 使用算法 SHA-1 散列公开密钥文件,文件里面的所有字节都应进行散列;
- 将 a) 中提取的签名元素,与公开密钥文件及 b) 中获得的公开密钥文件的散列值一起传递给 DSA 进行核实,并返回签名正确与否。

注:如果签名验证正确,SA 将产生 DS 证书。

8.4.3 创建 DS 证书的程序

创建 DS 证书的程序通常由 SA 在鉴别 SSK 以后创建 DS 证书时执行。程序如下:

- 从自我签名密钥文件中舍弃签名元素(即前两个元素以及它们的行头),剩余为公开密钥文件;
- 使用算法 SHA-1 散列公开密钥文件,文件里面的所有字节都应进行散列;
- 利用 DSA 数字签名算法通过 SA 私有密钥、公开密钥的文件的散列和一个随机字符串实现对公开密钥文件进行签名,并返回 2 个签名元素(“R”和“S”);
- 书写“R”和“S”内容到证书文件,然后附加 a) 中所得的结果,形成 DS 证书。

8.4.4 鉴别 DS 证书的程序

鉴别 DS 证书的程序通常是 DS 从 SA 获得的 DS 证书在使用之前执行。程序如下:

- 从 SA 网站获得 SA 公开密钥;
- 提取证书中的头两个数据字符串,剩下的就是公开密钥文件;
- 使用算法 SHA-1 散列公开密钥文件,文件里面的所有字节都应进行散列;
- 将 b) 中提取的签名元素,与 a) 中 SA 公开密钥及 c) 中公开密钥文件的散列值一起传递给 DSA 进行核实,并返回签名正确与否。

注:如果 DS 证书签名鉴别结果正确,其签名元素“R”和“S”可用于构建 ENC 数字签名。

8.4.5 从签名文件中鉴别 DS 证书的程序

从签名文件中鉴别 DS 证书的程序通常由 DC 执行,在数据服务公开密钥被提取用于鉴别 ENC 签名之前,使用 SA 公开密钥鉴别存储在 ENC 签名中的 DS 证书时使用。鉴定程序如下:

- 从签名文件中提取所期望的签名和证书;
- 舍弃第一个签名部分(前两个数据字符串和其相应标题,这是 ENC 数据的数据服务签名),留下部分是证书;

- c) 提取剩下的签名部分,即从 b) 中获得的剩下的文件中的前两个数据字符串,形成公开密钥文件;
- d) 使用算法 SHA-1 散列公开密钥文件,文件里面的所有字节都应进行散列;
- e) 将 c) 中提取的签名部分,与 c) 中获得的 SA 公开密钥文件及 d) 中获得的公开密钥文件的散列值一起传递给 DSA 进行核实,并返回签名正确与否。

8.4.6 鉴别 SA 数字证书的程序

8.4.6.1 鉴别 SA 数字证书的程序通常在以下情况下执行:

- a) DS 核实 SA 公开密钥时,该 SA 公开密钥用于鉴别 DS 证书;
- b) DC 核实 SA 公开密钥时,该 SA 公开密钥用于鉴别 ENC 数据的数字签名。

8.4.6.2 鉴别 SA 数字证书的程序如下:

- a) 手动比较独立安装的 SA 数字证书中的 SA 公开密钥和从 SA 网站上获得的公开密钥;
- b) 如果上面的检查失败,DS 将不接受 SA 数字证书;如果 SA 数字证书合法,其所包含的数据服务公开密钥可以用于核实 ENC 签名文件。

8.4.7 更新 SA 数字证书的程序

SA 在下列情况下,将会出版和提供一个新的 SA 数字证书:

- a) 当 SA 数字证书过期时,证书不应包括已经变化的公开密钥;
- b) 当 SA 私有密钥出现安全问题时,SA 数字证书应包括一个新的公开密钥。

SA 将出版新的数字证书,并在 SA 网站上公布公开密钥。所有 DS 和 OEM 会立即得到通知,并收到新的数字证书的复制品。

DS 和 OEM 共同负责将新的 SA 数字证书以及新的 SA 公开密钥通知给 DC。

当一个新的 SA 数字证书或者公开密钥发布时,通常由该保护方案的所有用户执行该程序。按如下程序执行:

- a) 从 SA 网站上获得新的 SA 数字证书以及可印刷的 SA 公开密钥;
- b) 应用程序应下载新的 SA 数字证书,核对公开密钥是否与印刷的公开密钥一致。只有完成这些操作后,应用程序才认为该 SA 公开密钥是正确的;
- c) 使用新发布的证书替代存在的 SA 数字证书。

8.4.8 创建签名参数的程序

创建签名参数(PQG)的程序通常由 SA 和 DS 在创建公开和私有密钥对的时候执行。尽管由 DS 创建的 PQG 参数不必与包含在 SA 公开密钥和 SA 数字证书中的一致,但是使用的密钥长度应一致。

9 数字签名

9.1 数字签名文件格式

数字签名文件的格式与数字 DS 证书文件相同。数字签名(文件)采用 ASCII 文本格式,并按照规定的结构和次序表示。

数字签名文件应包含下面的 6 个数据字符串:

- a) 第一个数据字符串“R”,包括 10 块区域(每区域含 4 个字符),是数字签名中的“R”部分;
- b) 第二个数据字符串“S”,包括 10 块区域(每区域含 4 个字符),是数字签名中的“S”部分;
- c) 第三个数据字符串“P”,包括 32 块区域(每区域含 4 个字符),用来核实数字签名;
- d) 第四个数据字符串“p”,包括 10 块区域(每区域含 4 个字符),用来核实数字签名;

- e) 第五个数据字符串“g”,包括 32 块区域(每区域含 4 个字符),用来核实数字签名;
- f) 第六个数据字符串“y”,包括 32 块区域(每区域含 4 个字符),用来核实数字签名。

每一个数据字符串应遵守下述格式:

- a) 前面有一个标题行。标题行在开始处用“//”表示,后跟一个空格字符和 ASCII 文本标题字符;
- b) 用 16 进制数字(0~9,A~F)表示,任何字母字符均以大写表示;
- c) 以点号“.”表示结束;
- d) 每 4 个字符之间用空格字符分开;
- e) 每一个数据字符最后有一个行分割符(回车)(为了增加可读性,一些数据被分成两行)。

示例:

// Signature part R:

582F 9DE1 FCA6 A6A9 9BF7 CE24 72EF 9DE5 7613 B11C.

46FF 7302 FDF7 CBDF 2193 43B4 337C B6ED E146 E73E.

// BIGp

FCA6 82CE 8E12 CABA 26EF CCF7 110E 526D B078 B05E DECB CD1E B4A2 08F3 AE16

17AE 01F3 5B91 A47E 6DF6 3413 C5E1 2ED0 899B CD13 2ACD 50D9 9151 BDC4 3EE7

3759 2E17.

// BIGq

962E DDCC 369C BA8E BB26 0EE6 B6A1 26D9 346E 38C5.

// BIGg

6784 71B2 7A9C F44E E91A 49C5 147D B1A9 AAF2 44F0 5A43 4D64 8693 1D2D 1427

1B9E 3503 0B71 FD73 DA17 9069 B32E 2935 630E 1C20 6235 4D0D A20A 6C41 6E50

BE79 4CA4.

// BIGy

AA25 DF9E C3CA 96B7 9D01 3ED8 D572 D47C B3F3 80D0 731D EA47 B106 26BA C387 C1FA 3C33

EC55 6845 3744 76BE 5825 6E07 A74D 607F 7A5E 7B7E 3455 71D8 2110 4C8A C4BF.

9.2 ENC 数字签名文件格式

发布 ENC 信息的 DS 总是在 ENC 数字签名文件中创建签名/证书对。DC 需要核实签名文件中的签名/证书对。签名文件应包括签名和证书对。仅有签名的文件不能证明 DS 的身份,因此是非法的。

ENC 数字签名文件采用 ASCII 文本格式,并按照规定的结构和次序表示。

文件以下面的次序包含数据字符串:

- a) 第一个数据字符串“R”,包括 10 块区域(每区域含 4 个字符),是 ENC 签名中的数据服务商签名的“R”部分;
- b) 第二个数据字符串“S”,包括 10 块区域(每区域含 4 个字符),是 ENC 签名中的数据服务商签名的“S”部分;
- c) 第三个数据字符串“R”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥中的系统管理员签名的“R”部分;
- d) 第四个数据字符串“S”,包括 10 块区域(每区域含 4 个字符),是 DS 公开密钥中的系统管理员签名的“S”部分;
- e) 第五个数据字符串“p”,包括 32 块区域(每区域含 4 个字符);
- f) 第六个数据字符串“q”,包括 32 块区域(每区域含 4 个字符);
- g) 第七个数据字符串“g”,包括 32 块区域(每区域含 4 个字符);
- h) 第八个数据字符串“y”,包括 32 块区域(每区域含 4 个字符),是 DS 的公开密钥。

头两个“R”和“S”数据字符串是 DS ENC 签名,文件剩下的部分与 DS 证书一致。

每个数据字符串应遵守下述格式:

- 前面有一个标题行。标题行在开始处用“//”表示,后跟一个空格字符和 ASCII 文本标题字符;
- 用 16 进制数字(0~9,A~F)表示,任何字母字符均以大写表示;
- 以点号“.”表示结束;
- 每 4 个字符之间用空格字符分开;
- 每一个数据字符最后有一个行分割符(回车)(为了增加可读性,一些数据被分成两行)。

第二个“R”和“S”用来核实 DS 数字证书(p,q,g,y 字符串)。如果核实成功,DS 公开密钥(y 字符串)可以被提取出来,用于核实加密的 ENC 的数字签名(头两个“R”和“S”)。允许 DC 核实系统管理员数字证书,提取 DS 公开密钥,核实 ENC 数据的数字签名。

示例:

```
// Signature      part R:
63E8 A27F 85FB 7553 C80C E201 64E0 FB6A E8FB 20CE.

// Signature      part S:
8A66 7CCC 24BA F358 CF3F BAA3 BE84 745B 5C3F 8E27.

// Signature      part R:
582F 9DE1 FCA6 A6A9 9BF7 CE24 72EF 9DE5 7613 B11C.

// Signature      part S:
46FF 7302 FDF7 CBDF 2193 43B4 337C B6ED E146 E73E.

// BIGP
FCA6 82CE 8E12 CABA 26EF CCF7 110E 526D B078 B05E DECB CD1E B4A2 08F3 AE16
17AE 01F3 5B91 A47E 6DF6 3413 C5E1 2ED0 899B CD13 2ACD 50D9 9151 BDC4 3EE7
3759 2E17.

// BIGq
962E DDCC 369C BA8E BB26 0EE6 B6A1 26D9 346E 38C5.

// BIGg
6784 71B2 7A9C F44E E91A 49C5 147D B1A9 AAF2 44F0 5A43 4D64 8693 1D2D 1427
1B9E 3503 0B71 FD73 DA17 9069 B32E 2935 630E 1C20 6235 4D0D A20A 6C41 6E50
BE79 4CA4.

// BIGy
AA25 DF9E C3CA 96B7 9D01 3ED8 D572 D47C B3F3 80D0 731D EA47 B106 26BA C387 C1FA
3C33 EC55 6845 3744 76BE 5825 6E07 A74D 607F 7A5E 7B7E 3455 71DB 2110 4C8A C4BF.
```

9.3 数字签名的创建程序

9.3.1 签名电子航道图文件的程序

签名电子航道图文件的程序由 DS 执行,以对其 ENC 数据文件进行数字签名。在签名前,ENC 文件应被压缩和加密。程序如下:

- 使用 SHA-1 算法散列所需的 ENC 文件(基本和更新文件)。进行散列以前 ENC 单元应进行压缩和加密。文件里面的所有字节都应被散列;
- 将 a) 中得到的散列值、DS 私有密钥和随机字符串传递给 DSA 数字签名算法,将会返回 2 个签名参数(“R”和“S”);
- 写在签名文件里面作为头两个数据的字符串应符合 9.2 所定义的格式。文件的剩下部分由 DS 证书组成,DS 证书包含与用于创建签名的私有密钥相关的公开密钥。

9.3.2 鉴定 ENC 数字签名文件的程序

9.3.2.1 鉴定 ENC 数字签名文件的程序,通常由 DS 为核实 ENC 数字签名而执行,前提是 DC 已经鉴别 SA 证书和签名文件中的 DS 证书。程序如下:

- a) 从与 ENC 文件唯一相关的签名文件中提取签名和证书,剩下的就是 DS 公开密钥文件;
- b) 从 a) 中的输出结果中提取签名部分(头两个数据字符串及其的相应部分),剩下的是证书;
- c) 使用 SHA-1 算法散列所需 ENC 文件,文件里面的所有字节都被散列;
- d) 核实签名部分,通过传递签名、DS 公开密钥及 ENC 文件的散列值给 DSA 数字签名算法,将返回正确与否。

9.3.2.2 如果 DC 签名鉴定不正确,数据用户将不能解密 ENC,如果 ENC 鉴定正确,则 ENC 能被解密。

10 加密

10.1 一般要求

10.1.1 每一个 ENC 单元文件使用一个单元密钥加密。同一个单元密钥可以用来加密 ENC 同一版本的所有更新文件。单元密钥以单元许可证的形式传送给 DC。

10.1.2 ENC 数据文件的所有内容应加密,用户许可证和单元许可证也应加密。

10.2 加密 ENC 基本单元文件的程序

10.2.1 加密 ENC 基本单元文件的程序应由 DS 执行,ENC 文件应在加密之前被压缩。程序如下:

- a) 选择加密所需的单元密钥(参照下面的条件);
- b) 使用 a) 中得到的单元密钥,利用 Blowfish 算法加密 ENC 文件,形成加密的 ENC 文件。

10.2.2 选择的加密密钥应遵守下面的条件:

- a) 产生 ENC 单元第一版本的 CK1 和 CK2,CK1 用于加密 ENC 单元;
- b) 对现有的 ENC 单元的新版本,应使用现存的 CK2 密钥加密 ENC 图。将现存的 CK2 变成 CK1,产生 CK2,用于 ENC 单元的下一版本;
- c) 对于 ENC 单元新发布版本应使用当前的 CK1 进行加密;
- d) 更新文件应使用与基本单元相同的单元密钥加密。

10.3 解密 ENC 基本单元文件的程序

解密 ENC 基本单元文件的程序通常由 DC 按如下步骤执行:

- a) 选择适当的 CK1 或 CK2 作为密钥;
- b) 使用 a) 中得到的密钥,利用 Blowfish 算法解密加密的 ENC 基本单元文件,从而产生解密后的 ENC 基本单元文件。

10.4 加密 ENC 更新文件的程序

加密 ENC 更新文件的程序通常由 DS 执行。加密以前应对 ENC 文件进行压缩。程序如下:

- a) 选择用于加密源 ENC 基本单元文件的密钥给更新文件应用;
- b) 将 a) 中得到的密钥作为密钥,利用 Blowfish 算法加密 ENC 更新文件(ENC 更新文件在加密之前应该压缩)。

10.5 解密 ENC 更新文件的程序

解密 ENC 更新文件的程序通常由 DC 按如下步骤执行：

- a) 选择适当的 CK1 或 CK2 作为密钥；
- b) 使用 a) 中得到的密钥,利用 Blowfish 算法解密加密的 ENC 更新文件,从而产生解密后的 ENC 更新文件。

10.6 使用基本许可证文件选择单元密钥的程序

使用基本许可证文件选择单元密钥的程序通常由 DC 执行,用于决定使用单元许可证提供的哪一个单元解密。该程序对 ENC 基本单元和更新单元均适用。使用单元许可证文件来解密时,需要 DC 的应用系统使用试错法来决定使用哪一个单元密钥,与使用元许可证文件解密不同。程序如下：

- a) 将 CK1 作为解密密钥,利用 Blowfish 算法,对 ENC 文件解密；
- b) 解压缩 ENC 文件,如果解压成功,则 ENC 文件被解密完毕,可以使用；
- c) 如果解密不成功,将 CK2 作为解密密钥,利用 Blowfish 算法,对 ENC 文件解密；
- d) 解压缩 ENC 文件,如果解压成功,则 ENC 文件被解密完毕,可以使用；
- e) 如果解密再次不成功,则单元许可证文件中不包括合法的单元密钥,DC 应从 DS 那里获得新的单元许可证。

10.7 使用元许可证文件选择单元密钥

10.7.1 该程序通常由 DC 执行,用于决定使用哪一个单元许可证提供的单元密钥解密。该程序对 ENC 基本单元和更新单元均适用。元许可证文件的使用将会减少 DC 应用程序选择适当的单元密钥的试错次数。程序如下：

- a) 从 CATALOG.031 文件中 CATD-COMT 字段获得 ENC 文件的版本；
- b) 从元许可证文件中的单元许可证记录中获得 EDTN 数字,EDTN 表明应用哪一版本的 CK1；
- c) 如果从 a) 和 b) 中得到的版本数字相同,CK1 应该用于解密 ENC。CK1 作为解密密钥,利用 Blowfish 算法,解密 ENC 基本单元文件；
- d) 如果从 a) 中获得的版本数字比 b) 中的低,则元许可证文件包含的单元密钥用于 ENC 的更新版本,DC 应从 DS 获得更新版本的电子航道图；
- e) 如果从 a) 中获得的版本数字比 b) 中的高,DC 可以尝试使用 CK2 解密 ENC。DC 也应从 DS 那里获得更新的单元许可证。DC 使用 CK2 作为解密密钥,利用 Blowfish 算法,解密 ENC 基本单元解压所得的 ENC 文件。如果解压成功,则 ENC 文件被解密完毕,可以使用。如果解密不成功,意味着单元许可证文件中不包括合法的单元密钥,DC 应从 DS 那里获得新的单元许可证。

10.7.2 ENC OEM 负责确保应用程序适当管理单元许可证。多个 DS 可以提供相关的单元许可证给一个 ENC 服务。因此应用程序应能够管理来自多个 DS 的单元许可证,也可以允许同一个 ENC 单元来源于多个 DS,来源于同一个 DS 的单元许可证不可能用来解密来源于另一个 DS 的 ENC。