



# 中华人民共和国国家标准

GB/T 17902.3—2005/ISO/IEC 14888-3:1998

---

## 信息技术 安全技术 带附录的数字签名 第 3 部分：基于证书的机制

Information technology—Security techniques—Digital signatures with  
appendix—Part 3: Certificate-based mechanisms

(ISO/IEC 14888-3:1998, IDT)

2005-04-19 发布

2005-10-01 实施



中华人民共和国国家质量监督检验检疫总局  
中国国家标准化管理委员会

发布

## 目 次

前言 .....	III
1 范围 .....	1
2 规范性引用文件 .....	1
3 概述 .....	1
4 术语和定义 .....	2
5 符号和记法 .....	2
6 基于离散对数的数字签名机制 .....	2
6.1 密钥生成过程 .....	2
6.2 签名过程 .....	3
6.3 验证过程 .....	4
7 基于因子分解的数字签名机制 .....	6
7.1 密钥生成过程 .....	6
7.2 签名过程 .....	6
7.3 验证过程 .....	7
附录 A(规范性附录) 基于离散对数的带附录的基于证书的数字签名的例子 .....	8
A.1 基于非椭圆曲线的例子 .....	8
A.1.0 符号和记法 .....	8
A.1.1 数字签名算法(DSA) .....	8
A.1.2 Pointcheval/Vaudenay 签名 .....	10
A.2 基于椭圆曲线的例子 .....	12
A.2.1 椭圆曲线 DSA .....	12
附录 B(规范性附录) 基于因子分解的带附录的基于证书的数字签名的例子 .....	14
B.1 基于 GB 15851 的散列的数字签名 .....	14
B.1.1 域参数的生成 .....	14
B.1.2 签名密钥和验证密钥的生成 .....	14
B.1.3 签名过程 .....	14
B.1.4 验证过程 .....	15
B.2 ESIGN .....	15
B.2.1 域参数的生成 .....	15
B.2.2 签名密钥和验证密钥的生成 .....	15
B.2.3 签名过程 .....	15
B.2.4 验证过程 .....	16
附录 C(资料性附录) FIPS PUB 186 素数 $P$ 和 $Q$ 的生成 .....	17
附录 D(资料性附录) 椭圆曲线数学背景 .....	18
D.1 椭圆曲线和点 .....	18
D.1.1 $F_p$ 上的椭圆曲线加法规则 .....	18
D.1.2 $F_{2^m}$ 上的椭圆曲线加法规则 .....	18
附录 E(资料性附录) 带附录的基于证书的数字签名的数值例子 .....	20

E.1 数字签名算法(DSA) .....	20
E.1.1 DSA 参数 .....	20
E.1.2 DSA 签名密钥和验证密钥 .....	20
E.1.3 DSA 每个消息的数据 .....	20
E.1.4 DSA 签名 .....	20
E.1.5 DSA 验证数值 .....	20
E.2 Pointcheval/vaudenay 签名算法 .....	20
E.2.1 Pointcheval/vaudenay 参数 .....	20
E.2.2 Pointcheval/vaudenay 签名密钥和验证密钥 .....	21
E.2.3 Pointcheval/vaudenay 每个消息的数据 .....	21
E.2.4 Pointcheval/vaudenay 签名 .....	21
E.2.5 Pointcheval/vaudenay 验证数值 .....	21
E.3 椭圆曲线 DSA .....	21
E.3.1 例 1:域 $F_{2^m}$ , $m=191$ .....	21
E.3.2 例 2:域 $F_p$ , 192 比特素数 $p$ .....	22
E.4 基于 GB 15851—1995 的带散列的数字签名 .....	23
E.4.1 $v$ 为奇数( $v=3$ )的例子 .....	23
E.4.2 $v$ 为偶数( $v=2$ )的例子 .....	25
E.5 ESIGN 签名算法 .....	27
E.5.1 ESIGN 域参数 .....	27
E.5.2 签名密钥和验证密钥 .....	27
E.5.3 ESIGN 签名过程 .....	27
E.5.4 ESIGN 验证 .....	29
附录 F(资料性附录) 所选签名方案具有的特性 .....	31
附录 G(资料性附录) 专利信息 .....	32
参考文献 .....	33
图 1 带随机性证据的签名过程 .....	4
图 2 带随机化证据的验证过程 .....	5

## 前 言

GB/T 17902《信息技术 安全技术 带附录的数字签名》由以下几个部分组成：

第 1 部分：概述；

第 2 部分：基于身份的机制；

第 3 部分：基于证书的机制。

本部分为 GB/T 17902 的第 3 部分，等同采用国际标准 ISO/IEC 14888-3:1998《信息技术 安全技术 带附录的数字签名 第 3 部分：基于证书的机制》(英文版)。

本部分的附录 A 和附录 B 是规范性附录，附录 C 到附录 G 是资料性附录。

本部分由中华人民共和国信息产业部提出；

本部分由全国信息安全标准化技术委员会归口；

本部分由中国电子技术标准化研究所、信息安全国家重点实验室起草。

本部分主要起草人：叶茅枫、陈星、罗锋盈、胡磊、叶顶峰、张振峰、黄家英。

# 信息技术 安全技术 带附录的数字签名

## 第3部分:基于证书的机制

### 1 范围

GB/T 17902 规定了任意长度消息的带附录的数字签名机制并适用于提供数据原始鉴别、抗抵赖和数据完整性的方案。

GB/T 17902 的本部分规定了带附录的基于证书的数字签名机制。特别是,本部分提供了:

- 1) 基于证书的签名机制的一般描述,其安全性是基于所用交换群上的离散对数问题的困难性(见第6章)。
- 2) 基于证书的签名机制的一般描述,其安全机制是基于因子分解的困难性(见第7章)。
- 3) 使用任意长度消息的基于证书机制的带附录的各种常规数字签名机制(见附录A和附录B)。

### 2 规范性引用文件

下列文件中的条款通过 GB/T 17902 的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 15851—1995 信息技术 安全技术 带消息恢复的数字签名方案(idt ISO/IEC 9796:1991)

GB/T 17902.1—1999 信息技术 安全技术 带附录的数字签名 第1部分:概述

GB/T 17902.2—2005 信息技术 安全技术 带附录的数字签名 第2部分:基于身份的机制(ISO/IEC 14888-2:1999, IDT)

GB/T 18238.3—2002 信息技术 安全技术 散列函数 第3部分:专用散列函数(idt ISO/IEC 10118-3:1998)

ISO/IEC 9796-2:1997 信息技术 安全技术 带消息恢复的数字签名方案 第2部分:使用散列函数的机制

ISO/IEC 10118-4:1998 信息技术 安全技术 散列函数 第4部分:使用模数算法的散列函数

### 3 概述

在 GB/T 17902 的本部分中使用了 GB/T 17902.1—1999 中所给的定义、符号、数字长度和记法。

数字签名的验证需要签名实体的验证密钥。所以,验证方必须把正确的验证密钥与签名实体关联起来。对基于证书的机制来说,这种关联必须通过某种证书的方法来提供。例如,验证密钥是取自一个证书。

GB/T 17902 的本部分的目的是规定 GB/T 17902.1—1999 中描述的一般模型的下列过程和函数:

- a) 生成密钥的过程
  - 1) 生成域参数
  - 2) 生成签名和验证密钥
- b) 形成签名的过程
  - 1) (可选)形成预签名
  - 2) 为签名准备消息

- 3) 计算证据
- 4) 计算签名
- c) 验证过程
  - 1) 为验证准备消息
  - 2) 检索证据
  - 3) 计算验证函数
  - 4) 验证证据

#### 4 术语和定义

GB/T 17902.1—1999 确立的以及下列术语和定义适用于 GB/T 17902 的本部分。

##### 4.1

###### 有限交换群 finite commutative group

一个带二元操作“ $*$ ”的有限集合  $J$ , 满足:

- a) 对所有  $a, b, c \in J, (a * b) * c = a * (b * c)$
- b) 存在  $e \in J$ , 对所有  $a \in J, e * a = a$
- c) 对所有  $a \in J$ , 存在  $b \in J, b * a = e$
- d) 对所有  $a, b \in J, a * b = b * a$

##### 4.2

###### 有限交换群中元素的阶 order of an element in a finite commutative group

如果  $a^0 = e$ , 并且  $a^{n+1} = a * a^n$  (其中  $n \geq 0$ ) 被递归地定义, 则  $a \in J$  的阶是满足  $a^n = e$  的最小正整数  $n$ 。

#### 5 符号和记法

GB/T 17902.1—1999 确立的以及下列符号和记法适用于 GB/T 17902 的本部分:

$E$	一个有限交换群
$\#E$	$E$ 的基数
$a \parallel b$	$b$ 到 $a$ 的串接
$Q$	$\#E$ 的一个因子
$G$	在 $E$ 中阶为 $Q$ 的一个元素
$\text{gcd}(U, N)$	整数 $U$ 和 $N$ 的最大公因子
$T_1$	赋值的第一部分
$T_2$	赋值的第二部分
$Z_N$	整数 $U$ 的集合, 满足 $0 \leq U < N$
$Z_N^*$	整数 $U$ 的集合, 满足 $0 < U < N$ , 且 $\text{gcd}(U, N) = 1$
$\lfloor a \rfloor$	等于或小于 $a$ 的最大整数

#### 6 基于离散对数的数字签名机制

##### 6.1 密钥生成过程

###### 6.1.1 生成域参数

对基于离散对数的数字签名机制, 域参数的集合  $Z$  确定如下参数:

- a) 一个有限交换群  $E$
- b)  $\#E$  的一个或多个因子  $Q$
- c) 在  $E$  中阶为  $Q$  的一个或多个元素  $G$



在群  $E$  中,使用乘法符号。签名机制将使用  $E$  中的一个元素  $G$ 。需要说明的是,特定的签名机制可以对  $E, Q, G$  的选择附加约束。

### 6.1.2 生成签名密钥和验证密钥

签名实体的签名密钥是一个秘密生成的随机或伪随机的整数  $X$ ,使得  $0 < X < Q$  和  $\gcd(X, Q) = 1$ 。其相应的公开验证密钥  $Y$  是  $E$  的元素,并计算如下:

$$Y = G^X$$

注:在选取  $X$  时可以考虑排除少部分整数。

在某些情况下,需要确认域参数和密钥的有效性。但是,这超出了本标准的范围。

### 6.2 签名过程

在本章中描述一类签名机制的签名过程。本签名机制的签名函数是由  $(S, T_1, T_2)$  的一个排列  $(A, B, C)$  作为签名方程的系数来确定的。

$$AK + BX + C \equiv 0 \pmod{Q}$$

这个排列将被指定或在设置签名系统时设定。

签名过程和签名消息的形成是由八个步骤组成(见图1):

- a) 生成随机数
- b) 生成预签名
- c) 准备签名消息
- d) 计算证据(签名的第一部分)
- e) 计算赋值
- f) 计算签名的第二部分
- g) 构造附录
- h) 构造签名消息

在本过程中,签名实体使用它的私有签名密钥  $X$  和域参数  $E, G$  和  $Q$ 。

#### 6.2.1 生成随机数

签名实体生成一个秘密随机数,它是一个整数  $K$ ,其中  $0 < K < Q$ ,并满足  $\gcd(K, Q) = 1$ 。本步的输出是  $K$ ,它为签名实体秘密保存。

注:从可能的  $K$  值中可以考虑排除几个整数。

#### 6.2.2 生成预签名

本步的输入是随机数  $K$ ,在  $E$  中签名实体用  $K$  来计算方程

$$\Pi = G^K$$

本步的输出为预签名  $\Pi$ 。

#### 6.2.3 准备签名消息

该消息被分为输入数据  $M_1$  和  $M_2$  两个部分。这两个部分中的一个部分可能为空,并且这两个部分不必是不同的(细节见 GB/T 17902.1—1999)。

#### 6.2.4 计算证据(签名的第一部分)

本步的变量为 6.2.2 的预签名  $\Pi$  和 6.2.3 的  $M_1$ 。这些变量值是证据函数的输入值。证据函数的输出值为证据  $R$ 。

#### 6.2.5 计算赋值

赋值函数的输入为签名的第一部分,它取自于 6.2.4 的证据  $R$  和 6.2.3 的  $M_2$ 。赋值函数的输出为赋值  $T = (T_1, T_2)$ ,其中  $T_1$  和  $T_2$  是满足

$$0 < |T_1| < Q, 0 < |T_2| < Q$$

的整数。

### 6.2.6 计算签名的第二部分

本步的输入是取自于 6.2.1 的随机数  $K$ 、签名密钥  $X$ 、取自于 6.2.5 的赋值  $T=(T_1, T_2)$ 、 $(S, T_1, T_2)$  的排列  $(A, B, C)$  和在 6.1.1 中指定的域参数  $Q$ 。签名实体形成签名方程

$$(AK + BX + C) \equiv 0 \pmod{Q}$$

并且为得到签名的第二部分  $S$  求解签名方程,其中  $0 < S < Q$ 。 $(R, S)$  这对数将被称为签名  $\Sigma$ 。

### 6.2.7 构造附录

附录是由签名和一个可选的文本字段  $text$  构成的,如  $((R, S), text)$ 。文本字段可以包含一个证书,该证书是以密码手段将公开验证密钥与签名实体的标识数据捆绑起来。

注:如 GB/T 17902.1—1999 所示,根据应用的不同,构造附录并把它附加到消息上的方法是不同的。一般需要验证方能将正确的签名与消息捆绑起来。对成功的验证来说,在验证过程之前,验证方必须能将正确的验证密钥与签名捆绑起来。

### 6.2.8 构造签名消息

签名消息是将消息  $M$  和附录串接后得到的,为  $M \parallel ((R, S), text)$ 。

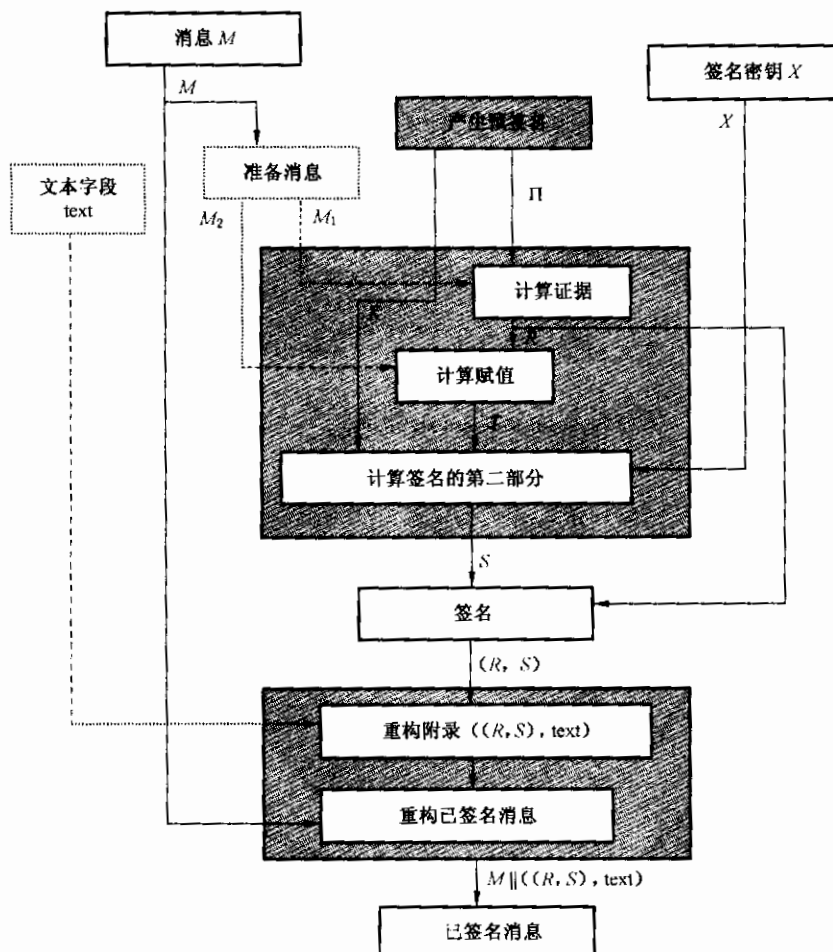


图 1 带随机性证据的签名过程

### 6.3 验证过程

验证过程由四个步骤构成(见图 2):

- a) 为验证准备消息
- b) 检索证据
- c) 计算验证函数
  - 1) 检索赋值



- 2) 重计算预签名
- 3) 重计算证据
- d) 验证证据

在本过程中,验证方使用签名方的验证密钥  $Y$  和域参数:有限群  $E, E$  中的元素  $G$  和它的阶  $Q$ 。

6.3.1 为验证准备消息

验证方从签名的消息中检索  $M$  并将消息分成两个部分  $M_1$  和  $M_2$ 。

6.3.2 检索证据

验证方从附录中检索签名  $(R, S)$ , 并将它分成证据  $R$  和签名的第二部分  $S$ 。

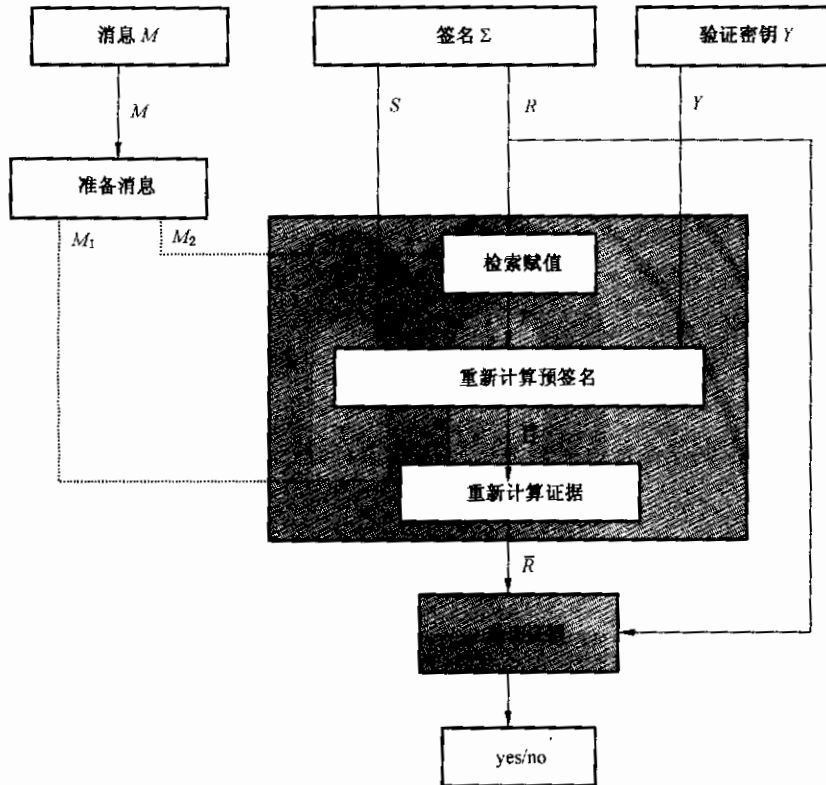


图 2 带随机化证据的验证过程

6.3.3 计算验证函数

6.3.3.1 检索赋值

本步等同于 6.2.5。赋值函数的输入由取自于 6.3.2 的证据  $R$  和取自于 6.3.1 的  $M_2$  组成。赋值  $T=(T_1, T_2)$  被重新计算,它是赋值函数的输出。

6.3.3.2 重新计算预签名

本步的输入是域参数的集  $Z$ 、验证密钥  $Y$ 、取自于 6.3.3.1 的赋值  $T=(T_1, T_2)$  和取自于 6.3.2 的签名的第二部分  $S$ 。签名方根据签名函数规定的次序将系数  $(A, B, C)$  赋予值  $(S, T_1, T_2)$ , 并且计算  $E$  中的元素  $\bar{\Pi}$  如下:

$$\bar{\Pi} = Y^m G^n$$

式中  $m = -A^{-1}B \pmod Q$  而  $n = -A^{-1}C \pmod Q$ 。

6.3.3.3 重新计算证据

本步的计算同 6.2.4。验证方执行证据函数的运算。输入是取自于 6.3.3.2 的  $\bar{\Pi}$  和取自于 6.3.1 的  $M_1$ 。输出是重新计算出的证据  $\bar{R}$ 。

6.3.4 验证证据

如果取自于 6.3.3.3 的重新计算证据  $\bar{R}$  与取自于 6.3.2 的  $R$  一致,签名是有效的。也许还需要附

加检查(见 A.1.2.4.6 的其他检查例子)。

## 7 基于因子分解的数字签名机制

基于因子分解的数字签名机制利用一个确定性证据并生成单签名,即该签名只有一个部分。但它可以为随机的或确定性的(参见 GB/T 17902.1—1999 图 2 和图 4)。在任一种情况下,机制利用一个整数  $N$  来作为验证密钥的一个成分,它的因子分解是签名密钥的一部分。要把  $N$  分解成为其素因子在计算上是不可行的。应对签名密钥的生成过程进行强制约束以使因子分解相当困难。

### 7.1 密钥生成过程

#### 7.1.1 生成域参数

对基于因子分解的数字签名机制来说,域参数的集合  $Z$  可选地包含一个整数  $v$ ,它是验证密钥的系统范围的部分,它符合 7.1.2 中指定的条件。

#### 7.1.2 签名密钥和验证密钥的生成

##### 7.1.2.1 签名密钥的生成

签名实体的签名密钥是个秘密生成的集  $X = (\{P_1, P_2, \dots, P_r\}, s)$ ,它是由一组随机或伪随机选取的素数  $P_i$  和整数  $s$  构成,这些素数不必不同。其中不同的素数至少要有两个。

##### 7.1.2.2 验证密钥的生成

验证密钥  $Y$  是一对整数  $(N, v)$ ,其中  $N$  是所有素数  $P_i$  的乘积, $P_i$  表示为  $\Pi$ ,而  $v$  是一个满足依赖于签名密钥条件的整数。

如果  $v$  指定为域参数,也许会在签名密钥上附加约束,以使  $v$  满足相应的条件。

### 7.2 签名过程

#### 7.2.1 生成预签名(可选)

一个随机化的签名机制使用预签名,它仅仅取决于随机数和签名密钥。预签名的计算分两步。

##### 7.2.1.1 生成随机数

签名实体秘密地生成一个随机数,它是个遵从附加限制的  $\text{mod } N$  的整数  $K$ 。本步的输出是  $K$ ,它为签名实体所秘密保存。

##### 7.2.1.2 计算预签名

预签名是随机数的一个函数,并独立于消息。本步的输入是随机数  $K$  和签名密钥。本步的输出是预签名,用  $\Pi$  表示。

#### 7.2.2 准备签名消息

消息用于构造数据输入  $M_1$  和  $M_2$ 。第二部分  $M_2$  也许为空,并且两个输入不必不同。

#### 7.2.3 计算证据

本步的输入是数据输入  $M_1$ ,输出是散列权标  $H$ ,它由数据输入  $M_1$  确定。注意散列权标被解释为一个  $\text{mod } N$  的整数,满足  $0 < H < N$ 。

#### 7.2.4 计算签名

本步的输入是 7.2.3 中计算出的证据、取自于 7.1.2.1 中的签名密钥和可选数据输入  $M_2$ (见 GB/T 17902.1—1999 图 2)。对一个随机化机制来说,随机数  $K$  和预签名  $\Pi$  也是有效的输入。输出是一个单部分签名  $\Sigma = S$ 。

#### 7.2.5 构造附录

附录是由签名  $\Sigma$  和一个可选文本字段  $\text{text}$  构成的。文本字段可以包括一个证书,它是以密码手段将公开验证密钥与签名实体的标识数据捆绑的。

#### 7.2.6 构造签名消息

签名消息是将消息  $M$  和取自于 7.2.5 的附录串接后得到的,为  $M \parallel (\Sigma, \text{text})$ 。

### 7.3 验证过程

#### 7.3.1 准备验证消息

验证方检索已签名的消息并且确定如 7.2.2 中所指定的两个数据输入部分  $M_1$  和  $M_2$ 。

#### 7.3.2 检索证据

验证方按照 7.2.3 中指定的证据函数检索证据  $H$ ，它是数据输入  $M_1$  的一个函数。

#### 7.3.3 计算验证函数

使用从域参数集  $Z$  或验证密钥  $Y$  得到的整数  $v$ ，验证方利用验证函数以获得一个重新计算的证据  $\bar{H}$ 。

#### 7.3.4 验证证据

如果检索出的证据  $H$  与由验证函数重新计算出的证据  $\bar{H}$  一致，则签名是有效的。

## 附录 A

## (规范性附录)

## 基于离散对数的带附录的基于证书的数字签名的例子

这些签名机制的例子是 U. S. NIST 的数字签名算法(DSA), Pointcheval/Vaudenay 和椭圆曲线签名。其方案在下面描述。

用于签名机制的群包括一个乘法群  $Z_P$ ,  $P$  是一个素数(用于: DSA 和 Pointcheval/Vaudenay), 一个由有限域上的椭圆曲线的点形成的加法群(用于: 椭圆曲线 DSA)。

## A.1 基于非椭圆曲线的例子

## A.1.0 符号和记法

$P$	素数
$Z_P$	整数 $U$ 的集合, 其中 $0 \leq U < P$
$Z_P^*$	整数 $U$ 的集合, 其中 $0 < U < P$

## A.1.1 数字签名算法(DSA)

此例取自于美国国家标准技术研究院(NIST)联邦信息处理标准 1994 年 5 月 19 日出版的 186 (FIPS PUB 186)。在第 6 章中定义的一般参数应有如下形式。为了与 GB/T 17902 的本部分的记法一致, 其中的记法与 FIPS PUB 186 稍许有些不同。

DSA 是利用群  $E = Z_P^*$  的签名机制, 其中  $P$  和  $Q$  为素数,  $Q$  整除  $P-1$ 。消息被分为两部分,  $M_1$  为空,  $M_2 = M$ 。证据函数由公式  $R = \prod \text{mod } Q$  来定义, 而赋值函数由公式  $(T_1, T_2) = (-R, -H)$  定义, 其中  $H = h(M)$  是消息  $M$  的散列权标, 根据附录 C 中所给的转换规则, 它被转换为一个整数。散列函数  $h$  是美国国家标准技术研究院(NIST)1995 年 4 月 17 日发布的安全散列标准(SHS)FIPS PUB 180-1 中采用的安全散列算法(SHA)。安全散列算法还在 ISO/IEC DIS 10118-3 中描述。(注意: 对 DSA 来说, 不需要带一个标识散列函数的控制字段, 因此散列权标仅为  $h(M)$ , 见 GB/T 17902.1—1999)。

DSA 签名公式的系数  $(A, B, C)$  设置如下:

$$(A, B, C) = (S, T_1, T_2)$$

因此签名公式变为:

$$(SK - RX - H) \equiv 0 \pmod{Q}。$$

## A.1.1.1 DSA 参数

$L$	$512 + 64I$ , $I$ 为 $0 \leq I < 8$ 的整数
$P$	素数, 其中 $2^{L-1} < P < 2^L$
$Q$	$P-1$ 的素因子, 其中 $2^{159} < Q < 2^{160}$
$F$	满足 $1 < F < P-1$ 和 $F^{(P-1)/Q} \text{ mod } P > 1$ 的整数
$G$	$F^{(P-1)/Q} \text{ mod } P$ , $E = Z_P^*$ 中阶为 $Q$ 的一个元素

整数  $P, Q$  和  $G$  可以是公开的, 并可以为一组用户所共有。

为了与 FIPS 相符, 按 FIPS PUB 186 附录 2 所指定的那样生成参数  $P$  和  $Q$  (细节可见 GB/T 17902 的本部分中的附录 C)。

注 1: 在本信息附录的例子中素数  $P$  的大小是如数字签名算法(DSA)所指定的大小。注意  $P$  的大小限制为最大 1024 比特。到 1994 年 5 月 19 日为止,  $P$  的大小对安全余量来说是足够的。据知, 算法数论的未来发展也许可能使 1024 比特的  $P$  不够大。

注 2: 建议所有用户检查 DSA 公共参数的正确生成。

注 3: 认识到, DSA 具有一个不利的特性就是可能遭到一种攻击, 该攻击找到所用的散列函数的碰撞的复杂度为

$2^{74}$ ，而不是在安全情况下的  $2^{80}$ 。对于那些仍然希望避开这个不利的特性的用户，可以通过使用 A.1.2 的机制来防止这种情况的发生。

#### A.1.1.2 DSA 签名密钥和验证密钥的生成

签名实体的签名密钥是一个秘密生成的随机或伪随机整数  $X$ ，满足  $0 < X < Q$ 。其相应的公开验证密钥  $Y$  为

$$Y = G^X$$

用户的秘密签名密钥  $X$  和公开验证密钥通常在一段时间内是固定的。签名密钥  $X$  必须被秘密保存。

#### A.1.1.3 DSA 签名过程

##### A.1.1.3.1 生成随机数

签名实体计算一个随机的或伪随机的整数  $K$ ，满足  $0 < K < Q$ 。必须为每个签名生成参数  $K$ ，并秘密保存。

##### A.1.1.3.2 生成预签名

本步的输入为随机数  $K$ ，且签名实体计算公式如下：

$$\Pi = G^K \bmod P$$

##### A.1.1.3.3 准备签名的消息

消息被分成为空的  $M_1$  和消息  $M_2 = M$ 。

##### A.1.1.3.4 计算证据

签名实体计算  $R = \Pi \bmod Q$ ，其中证据仅为预签名的一个函数。因此，

$$R = (G^K \bmod P) \bmod Q$$

##### A.1.1.3.5 计算赋值

签名实体计算赋值  $(T_1, T_2) = (-R, -H)$ ，其中  $H = h(M)$  是消息  $M$  的散列权标，且  $M = M_2$ 。

##### A.1.1.3.6 计算签名的第二部分

签名为  $(R, S)$ 。因此，

$$\begin{aligned} R &= (G^K \bmod P) \bmod Q \\ S &= (K^{-1}(h(M) + XR)) \bmod Q \end{aligned}$$

$h(M)$  的值是安全散列算法的 160 位比特串输出。若用于计算  $S$ ，必须将这个比特串转换为一个整数。转换规则在附录 C 中给出。

作为一个可选项，有人也许希望检查是否  $R=0$  或  $S=0$ 。如果  $R=0$  或  $S=0$ ，则应生成  $K$  的一个新值且应重新计算签名。（如果签名生成的恰当的话，不应该出现  $R=0$  或  $S=0$ ）。

##### A.1.1.3.7 构造附录

附录是将  $(R, S)$  和一个可选文本字段  $\text{text}$  串接后得到的，为  $(R, S) \parallel \text{text}$ 。

##### A.1.1.3.8 构造已签名的消息

已签名的消息是将消息  $M$  和附录串接后得到的，为  $M \parallel (R, S) \parallel \text{text}$ 。

#### A.1.1.4 DSA 验证过程

在验证已签名消息的签名前，验证方需要相信  $P, Q$  和  $G$  的值是正确的。

验证方也需要验证过程所需的数据项，如验证密钥（附加的需要的数据项见 GB/T 17902.1—1999 第 9 章）。

##### A.1.1.4.1 为验证准备消息

验证方从已签名的消息中检索  $M = M_2, M_1$  为空。

##### A.1.1.4.2 检索证据

验证方从附录中检索证据  $R$  和签名的第二部分  $S$ 。



**A. 1. 1. 4. 3 检索赋值**

本步等同于 A. 1. 1. 3. 5。赋值函数的输入是由取自于 A. 1. 1. 4. 2 的证据  $R$  和来自于 A. 1. 1. 4. 1 的  $M_2$  构成。赋值  $T=(T_1, T_2)$  被重新计算, 它是由 A. 1. 1. 3. 5 的赋值函数计算出来的。

**A. 1. 1. 4. 4 重新计算预签名**

本步的输入是域参数、验证密钥  $Y$ 、取自于 A. 1. 1. 4. 3 的赋值  $T=(T_1, T_2)$  和来自于 A. 1. 1. 4. 2 的签名的第二部分  $S$ 。验证方将系数  $(A, B, C)$  赋予值  $(S, T_1, T_2)$ , 如通过签名函数所确定的那样, 并且使用  $E$  中的如下公式得到预签名的重新计算值  $\bar{\Pi}$  :

$$\bar{\Pi} = Y^{-A^{-1}B \bmod Q} G^{-A^{-1}C \bmod Q} \bmod P$$

**A. 1. 1. 4. 5 重新计算证据**

本步的计算同 A. 1. 1. 3. 4。验证方运行证据函数。输入为 A. 1. 1. 4. 4 的  $\bar{\Pi}$ 。注意  $M_1$  为空。输出为重新计算证据  $\bar{R}$ 。

**A. 1. 1. 4. 6 验证证据**

假设  $M_2$  为 A. 1. 1. 4. 1 的值, 而  $R$  和  $S$  的值取自于 A. 1. 1. 4. 2。假设  $Y$  为签名实体的公开验证密钥。要验证签名, 验证方首先检查是否  $0 < R < Q$ , 且  $0 < S < Q$ 。如果两个条件有一个不满足, 签名应被拒绝。如果这两个条件均满足, 验证方将 A. 1. 1. 4. 5 的重新计算证据  $\bar{R}$  与 A. 1. 1. 4. 2 中的  $R$  值进行比较。如果  $\bar{R} = R$ , 则签名有效。

**A. 1. 2 Pointcheval/Vaudenay 签名**

Pointcheval/audenay 方法是一种 DSA 算法的变型, 其中  $E = Z_P^*$ ,  $P$  和  $Q$  为素数,  $Q$  整除  $P-1$ 。消息被分为两部分,  $M_1$  为空,  $M_2 = M$ 。证据由公式

$$R = \Pi \bmod Q$$

定义, 而赋值函数由公式

$$(T_1, T_2) = (-R, -H)$$

得到, 式中  $H = h(R \| M)$  是散列权标, 它是由证据  $R$  和消息  $M$  串接后得到的。散列函数  $h$  为安全散列算法 (SHA-1)。注意以上  $T_2$  的重新计算需要将散列代码转换成一个整数。这一步需要某些与此转换一致的方法 (见 ISO/IEC 10118-4:1998 中的例子)。

Pointcheval/audenay 签名方程的系数  $(A, B, C)$  设置如下:

$$(A, B, C) = (S, T_1, T_2)$$

因此签名等式变成:

$$SK - RX - H \equiv 0 \pmod{Q}$$

**A. 1. 2. 1 Pointcheval/Vaudenay 参数**

$P$	素数
$Q$	$P-1$ 的素因子
$F$	满足 $1 < F < P-1$ 和 $F^{(P-1)/Q} \bmod P > 1$ 的整数
$G$	$F^{(P-1)/Q} \bmod P$

注: 应特别关注  $P, Q$  和  $F$  的生成。例如也可以用到 A. 1. 1. 1 中的生成过程。

**A. 1. 2. 2 Pointcheval/Vaudenay 签名密钥和验证密钥的生成**

签名实体的签名密钥是一个秘密生成的随机的或伪随机的整数  $X$ , 满足  $0 < X < Q$ 。相应的公开验证密钥  $Y$  是

$$Y = G^X$$

用户的私有签名密钥  $X$  和公开验证密钥  $Y$  通常在一段时间内是固定不变的。签名密钥  $X$  必须被秘密保存。

**A. 1. 2. 3 Pointcheval/Vaudenay 签名过程**

## A. 1.2.3.1 生成随机数

签名实体计算随机的或伪随机的整数  $K$ , 满足  $0 < K < Q$  且  $\gcd(K, Q) = 1$ 。

## A. 1.2.3.2 生成预签名

本步的输入是随机数  $K$ , 签名实体计算

$$\bar{\Pi} = G^K \bmod P$$

## A. 1.2.3.3 准备签名消息

消息被分为两部分,  $M_1$  为空, 和消息  $M_2$ ,  $M_2 = M$ 。

## A. 1.2.3.4 计算证据

签名实体计算  $R = \bar{\Pi} \bmod Q$ , 其中证据仅为预签名的一个函数。因此,

$$R = (G^K \bmod P) \bmod Q$$

## A. 1.2.3.5 计算赋值

签名实体计算赋值  $(T_1, T_2) = (-R, -H)$ , 其中  $H = h(R \| M)$  是散列权标, 它是由证据和消息  $M$  (即  $M = M_2$ ) 串接后得到的。

## A. 1.2.3.6 计算签名

签名为  $(R, S)$ 。因此,

$$\begin{aligned} R &= (G^K \bmod P) \bmod Q \\ S &= K^{-1}(h(R \| M) + XR) \bmod Q \end{aligned}$$

## A. 1.2.3.7 构造附录

附录是由  $(R, S)$  和一个可选文本字段  $\text{text}$  串接后得到的, 为  $(R, S) \| \text{text}$ 。

## A. 1.2.3.8 构造已签名的消息

已签名的消息是由消息  $M$  和附录串接后得到的, 为  $M \| (R, S) \| \text{text}$ 。

## A. 1.2.4 Pointcheval/Vaudenay 验证过程

在验证已签名消息的签名前, 验证方需要确认  $P, Q$  和  $G$  的值和其他所需的数据项是正确的。

## A. 1.2.4.1 为验证准备消息

验证方从已签名的消息中检索  $M_2 = M, M_1$  为空。

## A. 1.2.4.2 检索证据

验证方从附录中检索证据  $R$  和签名的第二部分  $S$ 。

## A. 1.2.4.3 检索赋值

本步等同于 A. 1.2.3.5。赋值函数的输入是由取自于 A. 1.2.4.2 的证据  $R$  和来自于 A. 1.2.4.1 的  $M_2$  构成。赋值  $T = (T_1, T_2)$  被重新计算, 它是 A. 1.2.3.5 的赋值函数的输出。

## A. 1.2.4.4 重新计算预签名

本步的输入是域参数、验证密钥  $Y$ 、取自于 A. 1.2.4.3 的赋值  $T = (T_1, T_2)$  和来自于 A. 1.2.4.2 的签名的第二部分  $S$ 。验证方将系数  $(A, B, C)$  赋予值  $(S, T_1, T_2)$ , 如签名函数所确定的那样, 并且使用  $E$  中的如下公式计算得到预签名的重新计算值  $\bar{\bar{\Pi}}$  :

$$\bar{\bar{\Pi}} = Y^{-A^{-1}B \bmod Q} G^{-A^{-1}C \bmod Q} \bmod P$$

## A. 1.2.4.5 重新计算证据

本步的计算同 A. 1.2.3.4。验证方执行证据函数。输入为 A. 1.2.4.4 的  $\bar{\bar{\Pi}}$  和 A. 1.2.4.1 的  $M_1$ 。输出为重新计算证据  $\bar{R}$ 。

## A. 1.2.4.6 验证证据

假设  $M_2$  为 A. 1.2.4.1 的值, 而  $R$  和  $S$  的值取自于 A. 1.2.4.2。验证方首先检查是否  $0 < R < Q$  同时  $0 < S < Q$ 。如果两个条件有一个不满足, 签名应被拒绝。如果这两个条件均满足, 验证方将 A. 1.2.4.5 的重新计算证据  $\bar{R}$  与 A. 1.2.4.2 中的  $R$  值进行比较。如果  $\bar{R} = R$ , 则签名有效。

**A.2 基于椭圆曲线的例子**

**A.2.1 椭圆曲线 DSA**

下面的机制是 DSA 算法在椭圆曲线上的类比。[参见附录 D, 附加椭圆曲线数学背景信息] 因此它是一个利用椭圆曲线上的点的循环群  $E$  的签名机制。我们采用

$$(A, B, C) = (S, T_1, T_2)$$

其中  $(T_1, T_2) = (-R, -H)$ , 并且  $H$  是消息  $M$  的散列权标。

因此签名公式变成:

$$SK - RX + H \equiv 0 \pmod{Q}$$

**A.2.1.1 椭圆曲线 DSA 参数**

- $F$  一个有限域
- $E$  有限域  $F$  上的椭圆曲线群
- $\#E$   $E$  的基数
- $Q$   $\#E$  的素因子
- $G$  阶为  $Q$  的椭圆曲线上的点

注: 虽然标准文献中椭圆曲线群都写作加法形式, 为了与以上的一般描述保持一致, 我们仍使用乘法记号。

**A.2.1.2 椭圆曲线 DSA 签名密钥和验证密钥的生成**

签名实体的签名密钥是一个秘密生成的随机的或伪随机的整数  $X$ , 满足  $0 < X < Q$  的。其对应的公开验证密钥  $Y$  是

$$Y = G^X$$

用户的私有签名密钥  $X$  和公开验证密钥  $Y$  通常在一段时间内是固定不变的。签名密钥  $X$  必须被秘密保存。

**A.2.1.3 椭圆曲线 DSA 的签名过程**

**A.2.1.3.1 生成随机数**

生成随机的  $0 < K < Q$  的秘密整数  $K$ 。

**A.2.1.3.2 生成预签名**

本步的输入是随机数  $K$ , 签名实体计算

$$\Pi = G^K$$

**A.2.1.3.3 准备签名消息**

消息被分成两部分,  $M_1$  为空, 和消息  $M_2, M_2 = M$ 。

**A.2.1.3.4 计算证据**

签名实体计算  $R = \Pi_X \pmod{Q}$ , 其中  $\Pi_X$  是点  $\Pi$  的  $X$  坐标, 在范围  $[1, Q-1]$  中理解为一个整数(见 GB/T 17902.1—1999 中第 5.2 条)。

**A.2.1.3.5 计算赋值**

签名实体计算赋值  $(T_1, T_2) = (-R, -H)$ , 其中  $H$  是消息  $M$  的散列权标。

**A.2.1.3.6 计算签名的第二部分**

签名为  $(R, S)$ 。因此,

$$\begin{aligned} R &= \Pi_X \pmod{Q} \\ S &= (K^{-1}(XR - H)) \pmod{Q} \end{aligned}$$

从而

$$(R, S) = ((\Pi_X) \pmod{Q}, (K^{-1}(XR - H)) \pmod{Q})$$

**A.2.1.3.7 构造附录**

附录是由  $(R, S)$  和一个可选文本字段  $\text{text}$  串接后得到的, 为  $(R, S) \parallel \text{text}$ 。

#### A.2.1.3.8 构造已签名的消息

已签名的消息是由消息  $M$  和附录串接后得到的,为  $M\|(R,S)\|\text{text}$ 。

#### A.2.1.4 椭圆曲线 DSA 的验证过程

验证实体需要验证过程所必需的数据项。

##### A.2.1.4.1 为验证准备消息

验证方从已签名的消息中检索  $M$ ,并且将消息分成两部分  $M_1$  和  $M_2$ 。 $M_1$  为空, $M_2=M$ 。

##### A.2.1.4.2 检索证据

验证方从附录中检索证据  $R$  和签名的第二部分  $S$ 。

##### A.2.1.4.3 检索赋值

本步等同于 A.2.1.3.5。赋值函数的输入是由取自于 A.2.1.4.2 的证据  $R$  和来自于 A.2.1.4.1 的  $M_2$  构成。赋值  $T=(T_1, T_2)$  被重新计算,它是 A.2.1.3.5 的赋值函数的输出。

##### A.2.1.4.4 重新计算预签名

本步的输入是域参数、验证密钥  $Y$ 、取自于 A.2.1.4.3 的赋值  $T=(T_1, T_2)$  和来自于 A.2.1.4.2 的签名的第二部分  $S$ 。验证方将系数  $(A, B, C)$  赋予值  $(S, T_1, T_2)$ ,如签名函数所确定的那样,并且使用如下公式计算得到预签名的重新计算值  $\bar{\Pi}$  :

$$\bar{\Pi} = G^{-A^{-1}C \bmod Q} Y^{-A^{-1}B \bmod Q}$$

##### A.2.1.4.5 重新计算证据

本步的计算同 A.2.1.3.4。验证方运行证据函数。输入为 A.2.1.4.4 的  $\bar{\Pi}$ 。输出为重新计算证据  $\bar{R}$ 。

##### A.2.1.4.6 验证证据

如果  $M, R$  和  $S$  是从已签名的消息中检索出来的值,并且  $Y$  是签名方的公开验证密钥。要验证签名,验证方首先检查是否  $0 < R < Q$  且  $0 < S < Q$ ;如果两个条件有一个不满足,签名应被拒绝。如果这两个条件均满足,验证方将 A.2.1.4.5 的重新计算证据  $\bar{R}$  与 A.2.1.4.2 中已检索出的  $R$  的内容进行比较。如果  $\bar{R}=R$ ,则签名有效。



## 附录 B

(规范性附录)

## 基于因子分解的带附录的基于证书的数字签名的例子

这些签名机制的例子是基于 GB 15851(确定性的)和 ESIGN(随机化的)带有散列的数字签名。其方案在下面描述。

## B.1 基于 GB 15851 的散列的数字签名

在 GB 15851 中所给的数字签名机制是基于因子分解的确定性签名机制。正因为如此,它不使用随机数或预签名。但有两个秘密素因子  $P_1$  和  $P_2$ , 它们在第 7 章的签名密钥中定义。

## B.1.1 域参数的生成

域参数  $Z$  可选的包括系统范围验证指数  $v$  所作的指定。其他系统参数如散列函数可选的在域参数中指定。

## B.1.2 签名密钥和验证密钥的生成

## B.1.2.1 公开验证指数

如果在域参数集中没有指定,签名实体选择一个正整数  $v$ , 其中  $v < N$ (模数)。

## B.1.2.2 签名密钥的生成

签名实体秘密地生成由两个随机的或伪随机选取的不同素数  $P_i$  组成的集合  $\{P_1, P_2\}$ , 满足如下条件:

——如果  $v$  是奇数, 则  $P_i - 1$  应与  $v$  互素;

——如果  $v$  是偶数, 则  $(P_i - 1)/2$  应与  $v$  互素, 且  $P_1 - P_2$  不应被 8 整除。

在  $P_i$  上的附加约束是可选的, 其约束用于确保  $N = P_1 P_2$  的因子分解在计算上是不可行的。

签名实体计算公开模数  $N = P_1 P_2$  和签名指数  $s$ , 它是一个  $\text{mod } N$  的整数, 满足  $0 < s < N$ , 并且  $sv \equiv 1 \pmod{\text{lcm}(P_1 - 1, P_2 - 1)}$ , 如果  $v$  为奇数;

$sv \equiv 1 \pmod{1/2 \text{ lcm}(P_1 - 1, P_2 - 1)}$ , 如果  $v$  为偶数。

签名密钥  $X$  是集合  $(\{P_1, P_2\}, s)$ 。

## B.1.2.3 验证密钥的生成

验证密钥  $Y$  是  $(N, v)$ 。

## B.1.3 签名过程

签名过程是一个确定性的签名机制。因此, 它不生成预签名。

## B.1.3.1 准备签名消息

数据输入  $M_1 = M$  是消息,  $M_2$  为空。

## B.1.3.2 计算证据

确定性的证据是一个  $\text{mod } N$  的整数  $H$ , 它由消息的散列权标来确定。它是由 ISO/IEC 10118 中定义的填充散列码与一个可选的标识散列函数的控制域串接后得到的。如果散列函数不是域参数唯一指定的, 则控制域是强制的。如果验证密钥是偶数, 其散列权标可以用 *Jacobi* 符号  $1 \pmod{N/2}$  的办法来强制得到。

## B.1.3.3 计算签名

签名为  $S = H^s \pmod{N}$ 。

## B.1.3.4 构造附录

附录是由签名和一个可选的文本字段 *text* 构成的。文本字段可以包含一个证书, 证书以密码的手段将公开验证密钥与签名实体的标识数据捆绑起来。



**B.1.3.5 构造已签名消息**

已签名的消息是由消息  $M$  和附录 A 接后得到的,为  $M\|(S, \text{text})$ 。

**B.1.4 验证过程**

验证实体需要验证过程所需的必要数据项(见 GB/T 17902.1—1999 第 9 章)。

**B.1.4.1 准备验证消息**

验证方从已签名的消息中检索出  $M=M_1$ 。 $M_2$  为空。

**B.1.4.2 检索证据**

证据  $H$  是由 B.1.3.2 的数据输入  $M_1$  重构而成的。

**B.1.4.3 计算验证函数**

使用由域参数  $Z$  或验证密钥  $Y$  得到的整数  $v$  和由验证密钥得到的整数  $N$ ,验证方计算:

$$\bar{H} = S^v \bmod N$$

如果验证指数为偶数, $\bar{H}$  按照其模 8 的余数来修改。

**B.1.4.4 验证证据**

只有当检索出的证据  $H$  与重新计算出的证据  $\bar{H}$  一致时,签名才有效。

**B.2 ESIGN****B.2.1 域参数的生成**

ESIGN 是使用整数  $N=P^2Q$  作为模数的数字签名机制,其中  $P>Q$  是素数,签名指数  $s$  等于验证指数  $v$ ,一个大于或等于 4 的整数。其公共指数可以包含在域参数中,或从附录的可选文本字段的证书中得到。另外,在域参数中指定的(可选的)是个整数  $n$ ,它指明了素数二进制的范围。通常, $n$  为用于表示  $N$  的比特数的 1/3。散列函数的范围限制到  $n-1$  比特位(即: $0<H<2^{n-1}$ )。

**B.2.2 签名密钥和验证密钥的生成****B.2.2.1 签名密钥的生成**

签名实体的签名密钥是一个秘密生成的集  $X=(\{P_1, P_2, P_3\}, s)$ ,它由两个不同的随机或伪随机选取的素数和签名指数  $s$  来确定,其中  $P_1=P_2=P, P_3=Q$ ,且  $P>Q, s\geq 4$ 。因子  $P$  和  $Q$  应被秘密保存。

**B.2.2.2 验证密钥的生成**

验证密钥是一对整数  $Y=(N, v)$ ,其中  $N$  是乘积  $N=P_1P_2P_3=P^2Q, v$  是满足条件  $v=s\geq 4$  的整数。

**B.2.3 签名过程**

ESIGN 的签名过程符合在 GB/T 17902.1—1999 第 8 章中描述的一般模型。它是使用确定性的证据和生成单部分签名的随机化签名机制。

**B.2.3.1 生成预签名**

预签名的计算分两步:

**B.2.3.1.1 生成随机数**

签名实体秘密地生成一个随机数,它是一个随机或伪随机的整数  $K \bmod PQ$ ,满足  $0<K<PQ$ 。本步的输出为  $K$ ,它由签名实体秘密保存。

**B.2.3.1.2 生成预签名**

本步的输入是随机数  $K$  和签名密钥  $X$ 。签名实体计算预签名  $\Pi=(U, V)$ ,其中  $U=K^s \bmod N$  且  $V=(sK^{v-1})^{-1} \bmod P$ 。预签名的第二部分  $V$  应秘密保存。

**B.2.3.2 准备签名消息**

对证据的计算来说,采用整个消息  $M$  来当作输入  $M_1$ 。 $M_1=M$  是消息; $M_2$  为空,见 GB/T 17902.1—1999 中的 8.2。

### B.2.3.3 计算证据

确定性证据是消息的散列函数,表示为  $H$ 。其中  $H$  应小于  $2^{n-1}$ 。

### B.2.3.4 计算签名

本步的输入是  $P$  和  $Q$ ,它们来自于签名密钥  $X$ 、在 B.2.3.1.1 中计算出的随机数  $K$ 、在 B.2.3.1.2 中计算出的预签名  $\Pi = (U, V)$  和在 B.2.3.3 中计算出的证据  $H$ 。签名  $S$  的计算使用公式:

$$S = K + (\lfloor (2^{2n}H - U)/PQ \rfloor V \bmod P) PQ \bmod N$$

本步的输出是签名  $\Sigma = S$ 。

### B.2.3.5 构造附录

附录是由签名和一个可选文本字段  $\text{text}$  构成的。文本字段可以包含一个证书,证书以密码的手段将公开验证密钥与签名实体的标识数据捆绑起来。

### B.2.3.6 构造已签名消息

已签名的消息是由消息  $M$  和附录串接后得到的,为  $M \parallel (S, \text{text})$ 。

## B.2.4 验证过程

验证实体需要验证过程所需的必要数据项。

### B.2.4.1 为验证准备消息

验证方从已签名的消息中检索出  $M = M_1$ 。 $M_2$  为空。

### B.2.4.2 检索证据

证据  $H$  是由数据输入  $M_1$  重新构成的。

### B.2.4.3 计算验证函数

使用由域参数  $Z$  或验证密钥  $Y$  得到的整数  $v$ ,验证方计算  $\bar{H}$ ,它是  $S^v \bmod N$  的高端  $n$  位。

### B.2.4.4 验证证据

只有当重新构造的证据  $H$  与重新计算出的证据  $\bar{H}$  一致时,签名才有效。

## 附录 C

(资料性附录)

FIPS PUB 186 素数  $P$  和  $Q$  的生成

素数生成方案是使用 SHA-1, 由用户和提供“种子(SEED)”构造一个素数  $Q$  来开始的, 其中素数  $Q$  的范围在  $2^{159} < Q < 2^{160}$ 。一旦这步完成, 同一个“种子(SEED)”值被用来构造一个范围在  $2^{L-1} < X < 2^L$  的  $X$ 。然后通过循环  $X$  生成一个同余  $1 \bmod 2Q$  的数来形成  $P$ , 描述如下:

使用如下所示的二进制扩充, 范围在  $0 \leq x < 2^g$  的整数被转换成长度为  $g$  的比特序列:

$$x = x_1 * 2^{g-1} + x_2 * 2^{g-2} + \dots + x_{g-1} * 2 + x_g \rightarrow \{x_1, \dots, x_g\}$$

反之, 长度为  $g$  的比特序列  $\{x_1, \dots, x_g\}$  通过下述规则被转换成一个整数:

$$\{x_1, \dots, x_g\} \rightarrow x_1 * 2^{g-1} + x_2 * 2^{g-2} + \dots + x_{g-1} * 2 + x_g$$

注意: 这个序列的第一位对应于该整数的最高位, 而最后一位对应于该整数的最低位。

设  $L-1 = n * 160 + b$ , 其中  $b$  和  $n$  为整数, 且  $0 \leq b < 160$ 。

第 1 步, 选择一个任意的至少为 160 比特序列, 称之为 SEED。设  $g$  为 SEED 的比特长度。

第 2 步, 计算  $U = \text{SHA}[\text{SEED}] \text{ XOR } \text{SHA}[(\text{SEED}+1) \bmod 2^g]$ 。

第 3 步, 由  $U$  通过设置最高位(第 159 位)和最低位为 1 来形成  $Q$ 。以布尔操作的术语,  $Q = U \text{ OR } 2^{159} \text{ OR } 1$ 。注意到,  $2^{159} < Q < 2^{160}$ 。

第 4 步, 用一个健壮的素性测试算法来测试  $Q$  是否为素数。(健壮的素性测试是指一个非素数的数通过测试被判别为素数的概率至多为  $2^{-80}$ )。

第 5 步, 如果  $Q$  不是个素数, 执行第 1 步。

第 6 步, 置循环初始计数值  $\text{counter} = 0$ , 步长为  $\text{offset} = 2$ 。

第 7 步, 对循环  $k = 0, \dots, n$ , 令  $V_k = \text{SHA}[(\text{SEED} + \text{offset} + k) \bmod 2^g]$ 。

第 8 步, 设  $W$  为整数,  $W = V_0 + V_1 * 2^{160} + \dots + V_{n-1} * 2^{(n-1) * 160} + (V_n \bmod 2^b) * 2^{n * 160}$ , 并设  $X = W + 2^{L-1}$ 。注意  $0 \leq W < 2^{L-1}$ , 从而  $2^{L-1} \leq X < 2^L$ 。

第 9 步, 设  $c = X \bmod 2Q$ , 并且令  $P = X - (c-1)$ 。注意  $P$  同余于  $1 \bmod 2Q$ 。

第 10 步, 如果  $P < 2^{L-1}$ , 则运行第 13 步。

第 11 步, 执行对  $P$  的健壮素性测试。

第 12 步, 如果  $P$  通过第 11 步的测试, 执行第 15 步。

第 13 步, 置  $\text{counter} = \text{counter} + 1$  且  $\text{offset} = \text{offset} + n + 1$ 。

第 14 步, 如果  $\text{counter} \geq 2^{12} = 4096$ , 执行第 1 步, 否则(即  $\text{counter} < 4096$ )执行第 7 步。

第 15 步, 保存 SEED 值和 counter 值以让他人验证  $P$  和  $Q$  的正确生成。

**附录 D**  
(资料性附录)  
**椭圆曲线数学背景**

本附录中的内容直接取自 ANSI X9.62—1998《金融服务业的公钥密码系统:椭圆曲线数字签名算法(ECDSA)》。本附录还提供了附加椭圆曲线数学背景信息,它超出了本部分章条中所提供的内容。更多的椭圆曲线的数学信息请见 Menezes, A《椭圆曲线公钥密码系统》。

注意本附录中的一些记法与本部分中用得稍许有些不同。例如,本附录描述带加法记法的算法,而第 A.2 章中使用乘法记法。如  $gx$  转换为  $xG$ ,  $ab$  转换为  $A+B$ 。

### D.1 椭圆曲线和点

$F_q$  上定义的椭圆曲线  $E$  是一系列点  $P=(x_P, y_P)$  的集合,其中  $x_P$  和  $y_P$  是  $F_q$  的元素,满足某一个方程。并且它还包括一个表示为  $O$  的无穷远点。有时称  $F_q$  为基域。

如果  $q=p$  是个奇素数(所以基域为  $F_p$ ),并且  $p>3$ ,则  $a$  和  $b$  应满足  $4a^3+27b^2 \neq 0 \pmod{p}$ ,并且  $E$  上的每个点  $P=(x_P, y_P)$ (不是点  $O$ )应满足  $F_p$  中的如下方程:

$$y_P^2 + x_P^3 + ax_P + b$$

如果  $q=2^m$  是 2 的幂(所以基域为  $F_{2^m}$ ),则  $b$  在  $F_{2^m}$  中应非零,并且  $E$  上的每个点  $P=(x_P, y_P)$ (不是点  $O$ )应满足  $F_{2^m}$  中的如下方程:

$$y_P^2 + x_P y_P = x_P^3 + ax_P^2 + b$$

椭圆曲线点  $P$ (它不是无穷远点  $O$ )由两个域元素, $P$  的  $x$  坐标和  $P$  的  $y$  坐标表示: $P=(x_P, y_P)$ 。

#### D.1.1 $F_p$ 上的椭圆曲线加法规则

集合  $E(F_p)$  构成一个群,并有如下的加法规则:

- (i)  $O+O=O$
- (ii) 对所有  $(x, y) \in E(F_p)$ ,  $(x, y)+O=O+(x, y)=(x, y)$ 。
- (iii) 对所有  $(x, y) \in E(F_p)$ ,  $(x, y)+(x, -y)=O$ (即一个点  $(x, y)$  的逆元是  $-(x, y)=(x, -y)$ )。

(iv) (规则适用于将两个不同的不互逆元的点相加)。

假设:  $(x_1, y_1) \in E(F_p)$  和  $(x_2, y_2) \in E(F_p)$  是两个满足  $x_1 \neq x_2$  的点。

则:  $(x_1, y_1)+(x_2, y_2)=(x_3, y_3)$

其中  $x_3 = \lambda^2 - x_1 - x_2$ ,  $y_3 = \lambda(x_1 - x_3) - y_1$ , 并且  $\lambda = (y_2 - y_1)/(x_2 - x_1)$ 。

(v) (规则适用于将一个点加倍)。

假设:  $(x_1, y_1) \in E(F_p)$  是满足  $y_1 \neq 0$  的点。

则:  $2(x_1, y_1)=(x_3, y_3)$

其中  $x_3 = \lambda^2 - 2x_1$ ,  $y_3 = \lambda(x_1 - x_3) - y_1$ , 并且  $\lambda = (3x_1^2 + a)/(2y_1)$

群  $E(F_p)$  是可交换的,这意味着对  $E(F_p)$  中的所有点  $P_1$  和  $P_2$ ,  $P_1 + P_2 = P_2 + P_1$ 。如果  $\#E(F_p) = p+1$ , 曲线被称为超奇异的; 否则它是非超奇异的。

#### D.1.2 $F_{2^m}$ 上的椭圆曲线加法规则

集合  $E(F_{2^m})$  构成群,并有如下的加法规则:

- (i)  $O+O=O$
- (ii) 对所有  $(x, y) \in E(F_{2^m})$ ,  $(x, y)+O=O+(x, y)=(x, y)$ 。
- (iii) 对所有  $(x, y) \in E(F_{2^m})$ ,  $(x, y)+(x, x+y)=O$ , (即一个点  $(x, y)$  的逆元是  $-(x, y)=(x, x+y)$ )。

(iv) (规则适用于将两个不同的不互逆元的点相加)。

假设:  $(x_1, y_1) \in E(F_{2^m})$  和  $(x_2, y_2) \in E(F_{2^m})$  是两个满足  $x_1 \neq x_2$  的点。

则:  $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$

其中  $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$ ,  $y_3 = \lambda(x_1 + x_3) + x_3 + y_1$ , 并且  $\lambda = (y_1 + y_2) / (x_1 + x_2)$ 。

(v) (规则适用于将一个点加倍)。

假设:  $(x_1, y_1) \in E(F_{2^m})$  是满足  $x_1 \neq 0$  的点。

则:  $2(x_1, y_1) = (x_3, y_3)$

其中  $x_3 = \lambda^2 + \lambda + a$ ,  $y_3 = x_1^2 + (\lambda + 1)x_3$ , 并且  $\lambda = x_1 + y_1 / x_1$ 。

群  $E(F_{2^m})$  是可交换的, 这意味着对  $E(F_{2^m})$  中的所有点  $P_1$  和  $P_2$ ,  $P_1 + P_2 = P_2 + P_1$ 。



附录 E  
(资料性附录)

带附录的基于证书的数字签名的数值例子

E.1 数字签名算法(DSA)

所有数值生成完整的说明在 FIPS PUB 186 附录 5 中给出。在本例中使用 16 进制表示如下数值。

E.1.1 DSA 参数

$$L=200(512_{10})$$

SEED=d5014e4b	60ef2ba8	b6211b40	62ba3224	e0427dd3	
$F=2$					
P=8df2a494	492276aa	3d25759b	b06869cb	eac0d83a	fb8d0cf7
Cbb8324f	0d7882e5	d0762fc5	b7210eaf	c2e9adac	32ab7aac
49693dfb	f83724c2	ec0736ee	31c80291		
Q=c773218c	737ec8ee	993b4f2d	ed30f48e	dace915f	
G=626d0278	39ea0a13	413163a5	5b4cb500	299d5522	956cefcb
3bff10f3	99ce2c2e	71cb9de5	fa24babf	58e5b795	21925c9c
c42e9f6f	464b088c	c572af53	e6d78802		

E.1.2 DSA 签名密钥和验证密钥

X=2070b322	3dba372f	de1c0ffc	7b2e3b49	8b260614	
Y=19131871	d75b1612	a819f29d	78d1b0d7	346f7aa7	7bb62a85
9bfd6c56	75da9d21	2d3a36ef	1672ef66	0b8c7c25	5cc0ec74
858fba33	f44c0669	9630a76b	030ee333		

E.1.3 DSA 每个消息的数据

K=358dad57	1462710f	50e254cf	1a376b2b	deaadfbf	
$K^{-1}=0d516729$	8202e49b	4116ac10	4fc3f415	ae52f917	
$M="abc" \text{ 的 ASCII 码形式}=616263$					
$h(M)=a9993e36$	4706816a	ba3e2571	7820c26c	9cd0d89d	

E.1.4 DSA 签名

R=8baclab6	6410435c	b7181f95	b16ab97c	92b361c0	
S=41e2345f	1f56df24	58f426d1	55b4ba2d	b6dcd8c8	

E.1.5 DSA 验证数值

$\bar{R}=8baclab6$	6410435c	b7181f95	b16ab97c	92b341c0	
--------------------	----------	----------	----------	----------	--

E.2 Pointcheval/vaudenay 签名算法

下列数值用 16 进制记法表示。

E.2.1 Pointcheval/vaudenay 参数

$$L=200(512_{10})$$

$$F=2$$

P=8df2a494	492276aa	3d25759b	b06869cb	eac0d83a	fb8d0cf7
Cbb8324f	0d7882e5	d0762fc5	b7210eaf	c2e9adac	32ab7aac
49693dfb	f83724c2	ec0736ee	31c80291		

Q=c773218c	737ec8ee	993b4f2d	ed30f48e	dace915f	
G=626d0278	39ea0a13	413163a5	5b4cb500	299d5522	956cefc7
3bff10f3	99ce2c2e	71cb9de5	fa24babf	58e5b795	21925c9c
c42e9f6f	464b088c	c572af53	e6d78802		

**E. 2.2 Pointcheval/vaudenay 签名密钥和验证密钥**

X=2070b322	3dba372f	de1c0ffc	7b2e3b49	8b260614	
Y=19131871	d75b1612	a819f29d	78d1b0d7	346f7aa7	7bb62a85
9bfd6c56	75da9d21	2d3a36ef	1672ef66	0b8c7c25	5cc0ec74
858fba33	f44c0669	9630a76b	030ee333		

**E. 2.3 Pointcheval/vaudenay 每个消息的数据**

K=358dad57	1462710f	50e254cf	1a376b2b	deaadfbf	
$K^{-1}=0d516729$	8202e49b	4116ac10	4fc3f415	ae52f917	

M="abc"的 ASCII 码形式=616263

**E. 2.4 Pointcheval/vaudenay 签名**

R=8bac1ab6	6410435c	b7181f95	b16ab97c	92b361c0	
$R\parallel M=8bac1ab6$	6410435c	b7181f95	b16ab97c	92b361c0	616263
$h(R\parallel M)=2048680b$	36d19516	cf78e869	beae7bc9	ab5dc543	
S=5bfdac3d	665fa38f	6ed315b3	b2f41b86	15187ccd	

**E. 2.5 Pointcheval/vaudenay 验证数值**

$\bar{\Pi}=2fc6cb9a$	c3be0eac	3daf02ee	fb96fca3	846708a2	8dd05730
165fe509	42f7f07e	dfef8e52	fc9369e	3814aa24	607e8047
5d0e61ad	461d6b16	b6cec5ba	ae58946e		
$\bar{R}=8bac1ab6$	6410435c	b7181f95	b16ab97c	92b341c0	

**E. 3 椭圆曲线 DSA**

在下面的例子中,SHA-1 专门用于散列函数,以便散列权标仅仅按照附录 C 对应的数据项转换成 SHA-1 数值。

注:从安全考虑,避免密码性质脆弱的曲线很重要(即:应确保采用的椭圆曲线不受对一些特殊的曲线有效的离散对数算法的攻击)。

**E. 3.1 例 1:域  $F_{2^m}$ ,  $m=191$**

**E. 3.1.1 椭圆曲线 DSA 参数**

域  $F_{2^{191}}$  表示为模不可约多项式  $x^{191} + x^9 + 1$ 。

曲线为  $F_{2^{191}}$  上的  $E: Y^2 + XY = X^3 + aX^2 + b$ , 其中(16 进制)

a=2866537b	67675263	6a68f565	54e12640	276b649e	f7526267
b=2e45ef57	1f00786f	67b0081b	9495a3d9	5462f5de	0aa185ec

基点为  $G=(G_x, G_y)$ , 其中(16 进制)

$G_x=36b3daf8$	a23206f9	c4f299d7	b21a9c36	9137f2c8	4aelaa0d
$G_y=765be734$	33b3f95e	332932e7	0ea245ca	2418ea0e	f98018fb

G 的阶为(10 进制)

Q=1569275433846670190958947355803350458831205595451630533029

**E. 3.1.2 椭圆曲线 DSA 签名密钥和验证密钥**

签名密钥为

X=1275552191113212300012030439187146164646146646463749494799

验证密钥为

$Y = G^x = (Y_x, Y_y)$ , 其中(16进制)

$Y_x = 5de37e75 \quad 6bd55d72 \quad e3769cb30 \quad 96ffeb96 \quad 2614dea4 \quad ce28a2e7$   
 $Y_y = 55c0e0e0 \quad 2f5fb132 \quad caf416ef \quad 85b229bb \quad b8e13520 \quad 03125ba1$

### E. 3. 1. 3 椭圆曲线 DSA 每个消息的数据

$M = \text{"abc"}$  的 ASCII 码形式 = 616263

$h(M) = a9993e36 \quad 4706816a \quad ba3e2571 \quad 7820c26c \quad 9cd0d89d$

按照附录 C 它被转换成一个整数以得到

$H = 968236873715988614170569073515315707566766479517$

随机数的值被解释作一个 mod  $Q$  的整数, 如:

$K = 1542725565216523985789236956265265265235675811949404040041$

### E. 3. 1. 4 椭圆曲线 DSA 签名

$\Pi = G^k = (\Pi_x, \Pi_y)$

$\Pi_x = 438e5a11 \quad fb55e4c6 \quad 5471dcd4 \quad 9e266142 \quad a3bdf2bf \quad 9d5772d5$

$\Pi_y = 2ad603a0 \quad 5bd1d177 \quad 649f9167 \quad e6f475b7 \quad e2ff590c \quad 85af15da$

$R = \Pi_x$ , 按照附录 C 它被转换成一个 mod  $Q$  的整数。

$R = 87194383164871543355722284926904419997237591535066528048 \pmod{Q}$

$S = 308992691965804947361541664549085895292153777025772063598 \pmod{Q}$

### E. 3. 1. 5 椭圆曲线 DSA 验证

重新计算的预签名是按照 A. 2. 1. 4. 4, 由接收的消息和验证密钥得到的。

$\bar{\Pi} = (\bar{\Pi}_x, \bar{\Pi}_y)$ , 其中(16进制)

$\bar{\Pi}_x = 438e5a11 \quad fb55e4c6 \quad 5471dcd4 \quad 9e266142 \quad a3bdf2bf \quad 9d5772d5$

$\bar{\Pi}_y = 2ad603a0 \quad 5bd1d177 \quad 649f9167 \quad e6f475b7 \quad e2ff590c \quad 85af15da$

重新计算的证据  $\bar{R}$  是  $\bar{\Pi}_x$ , 它被转换成一个 mod  $Q$  的整数。

$\bar{R} = 87194383164871543355722284926904419997237591535066528048$

### E. 3. 2 例 2: 域 $F_p$ , 192 比特素数 $p$

#### E. 3. 2. 1 椭圆曲线 DSA 参数

域是  $F_p$ , 其中

$p = 6277101735386680763835789423207666416083908700390324961279$

$F_p$  上的曲线是  $E: Y^2 = X^3 + aX + b$ , 其中(16进制)

$a = \text{ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff}$

$b = 64210519 \quad e59c80e7 \quad 0fa7e9ab \quad 72243049 \quad feb8deec \quad c146b9b1$

基点是  $G = (G_x, G_y)$ , 其中(16进制)

$G_x = 188da80e \quad b03090f6 \quad 7cbf20eb \quad 43a18800 \quad f4ff0afd \quad 82ff1012$

$G_y = 07192b95 \quad ffc8da78 \quad 631011ec \quad 6b24cdd5 \quad 73f977a1 \quad 1e794811$

$G$  的阶为(10进制)

$Q = 6277101735386680763835789423176059013767194773182842284081$

#### E. 3. 2. 2 椭圆曲线 DSA 签名密钥和验证密钥

随机的选择签名密钥, 并且秘密保存它。它的值是一个 mod  $Q$  的整数, 为

$X = 651045770906015076056810763456358567190100156695615665659$

其对应的验证密钥由  $Y = G^x = (Y_x, Y_y)$  来给出, 其中(16进制)

$Y_x = 62b12d60 \quad 690cdcf3 \quad 30babab6 \quad e69763b4 \quad 71f994dd \quad 702d16a5$

$Y_Y = 63bf5ec0 \quad 8069705f \quad fff65e5c \quad a5c0d697 \quad 16dfcd34 \quad 74373902$

### E.3.2.3 椭圆曲线 DSA 每个消息的数据

$M = \text{"abc"}$  的 ASCII 码形式 = 616263

$h(M) = a9993e36 \quad 4706816a \quad ba3e2571 \quad 7820c26c \quad 9cd0d89d$

按照附录 C 它被转换成一个整数,以得到

$H = 968236873715988614170569073515315707566766479517$

随机数的值被解释作一个 mod  $Q$  的整数,它由  $K$  给出:

$K = 6140507067065001063065065565667405560006161556565665656654$

### E.3.2.4 椭圆曲线 DSA 签名

$\Pi = G^K = (\Pi_x, \Pi_y)$ , 其中(16 进制)

$\Pi_x = 88505238 \quad 0ff147b7 \quad 34c330c4 \quad 3d39b2c4 \quad a89f29b0 \quad f749fead$

$\Pi_y = 9cf9falc \quad befefb91 \quad 7747a3bb \quad 29c072b9 \quad 289c2547 \quad 884fd835$

$R = \Pi_x$ , 按照附录 C 它被转换成一个 mod  $Q$  的整数。

$R = 3342403536405981729393488334694600415596881826869351677613 \pmod{Q}$

按照 A.2.1 中给出的签名函数计算得到的签名值为:

$S = 5735822328888155254683894997897571951568553642892029982342 \pmod{Q}$

### E.3.2.5 椭圆曲线 DSA 验证

散列代码是由接收的消息计算出的:

$\bar{M} = \text{"abc"}$  的 ASCII 码形式 = 616263

$h(\bar{M}) = a9993e36 \quad 4706816a \quad ba3e2571 \quad 7820c26c \quad 9cd0d89d$

重新计算的散列权标是  $h(\bar{M})$ , 按照附录 C 它被转换成一个 mod  $Q$  的整数。

$\bar{H} = 968236873715988614170569073515315707566766479517$

重新计算的预签名是按照 A.2.1.4.4 的  $\bar{\Pi} = (\bar{\Pi}_x, \bar{\Pi}_y)$ , 由接收的消息和验证密钥得到的, 其中(16 进制)

$\bar{\Pi}_x = 88505238 \quad 0ff147b7 \quad 34c330c4 \quad 3d39b2c4 \quad a89f29b0 \quad f749fead$

$\bar{\Pi}_y = 9cf9falc \quad befefb91 \quad 7747a3bb \quad 29c072b9 \quad 289c2547 \quad 884fd835$

重新计算的证据  $\bar{R}$  是  $\bar{\Pi}_x$ , 它被转换成一个 mod  $Q$  的整数。

$\bar{R} = 3342403536405981729393488334694600415596881826869351677613$

由于重新计算的证据等于检索出的证据, 签名被验证。

## E.4 基于 GB 15851—1995 的带散列的数字签名

在本例中, 所有数值均以 16 进制来表示。本例来自于美国 ANSI 标准 X9.31。X9.31 中的例子, 使用一个非标准化的整数 mod  $N$  表示。为了与本标准一致, 需要修改这些整数数值。

### E.4.1 $v$ 为奇数( $v=3$ )的例子

#### E.4.1.1 签名密钥和验证密钥的生成

##### E.4.1.1.1 公开验证指数

$v=3$

##### E.4.1.1.2 私有签名密钥

###### E.4.1.1.2.1 私有素因子

$P_1 = d8cd81f0 \quad 35ec57ef \quad e8229551 \quad 49d3bff7 \quad 0c53520d$   
 $769d6576 \quad 646c7a79 \quad 2e16ebd8 \quad 9fe6fc5b \quad 6060bd97$   
 $8ed64a90 \quad 59c5b039 \quad 98a0e94c \quad 86d78b85 \quad ba37b5af$   
 $d987505f$



$P_2 =$	cc109249	5d867e64	065dee3e	7955f2eb	c7d47a2d
	7c995338	8f97dddc	3e1ca19c	35ca659e	dc3d6c08
	f64068ea	fedbd911	27f9cb7e	dc174871	1b624e30
	b857caad				

**E. 4. 1. 1. 2. 2 私有签名指数**

$s =$	1ccda20b	cffb8d51	7ee96668	66621b11	822c7950
	D55f4bb5	bee37989	a7d17312	e326718b	e0d62ccb
	11415f78	b36be2e6	0d599d4e	41346c82	d845498a
	81b2f663	2fd7d1cc	efcabf74	17350238	109ec289
	d5382762	b77a1c99	96dd1d2b	71a52faf	52aba9de
	d19f3f5d	5d71d054	73ec9c79	92d84128	0bac72b8
	7bf51eb1	cch65c87			

**E. 4. 1. 1. 3 公开验证模数**

$N =$	acd1cc46	dfe54fe8	f9786672	664ca269	0d0ad7e5
	003bc642	7954d939	eee8b271	52e6a947	45050cc2
	67883cd4	34875164	5019afd5	873a8b11	119fb93f
	0a31c654	c3ecff07	3233530c	79be90e0	26e2421d
	d378b88b	40136c48	7d33075a	1612ab90	c5b75d33
	2659a5d0	b5c19576	102d3424	31ac3bbb	a8f98449
	bd58bc0b	5e254633			

**E. 4. 1. 2 签名的生成**

$M =$ “abc”的 ASCII 码形式 = 616263

$h(M) =$	a9993e36	4706816a	ba3e2571	7850c26c	9cd0d89d
----------	----------	----------	----------	----------	----------

字符串  $x'33cc'$  后缀于散列权标, 它表示该散列函数是 SHA-1。由半个字节  $x'b'$  重复构成的填充部分(最右边尾部半个字节总是 16 进制值  $x'a'$ , 作为它与散列函数  $h(M)$  的分隔字段)是散列的前缀。它是以 16 进制头的值  $x'6'$  为领头的。

$H =$	6bbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
	bbbbbbbb	bbbaa999	3e364706	816aba3e	25717850
	c26c9cd0	d89d33cc			

$S = H^e \text{ mod } N$  现在计算如下:

$S =$	500abdc2	48d78e2d	b9182a98	7b296e93	53083435
	070fbe16	b1629a30	7cab53d3	c9b70611	bffa479e
	cb744397	b01c6f1c	b4775051	1510005e	e9f83709
	15788172	98db07fb	b746c6d7	774bb069	64244463
	3abc79c2	0cb81f8b	df9ff07e	eba2efc3	11a80438
	622492c8	89fc0b17	4681e5ce	427149c9	8fe34580
	5112f4d2	d8b53761			

**E. 4. 1. 3 签名的验证**

计算值  $\bar{H} = S^d \text{ mod } (N)$ 。



$\bar{H}$ = 6bbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbaa999	3e364706	816aba3e	25717850
c26c9cd0	d89d33cc			

由于  $\bar{H} = H \pmod{N}$ , 签名被验证。

E. 4.2  $v$  为偶数 ( $v = 2$ ) 的例子

用美国 ANSI 标准 X9.31 填充以确保任何有效证据以 16 进制值  $x'c'$  或  $x'6'$  结尾。

E. 4.2.1 签名密钥和验证密钥的生成

E. 4.2.1.1 公开验证指数

$v = 2$

E. 4.2.1.2 私有签名密钥

E. 4.2.1.2.1 私有素因子

$P_1 =$ dbb3cb4c	375c0ecd	2fd300db	4f085472	93ca004c
Edd2019c	e79ca08a	15eefb25	dd3baf98	183b0c2f
01d7f8b4	931856f6	dd3eba17	7d763c03	e1dceabc
d803be33				
$P_2 =$ eaaa4a53	47999fe7	6fb78760	64bbec66	cb409a77
39ef5a59	06613dc3	7225d41d	2beb1f9f	5ec77a85
38767a87	bb7015d6	07ff26de	61282753	9306balc
fff093a7				

E. 4.2.1.2.2 私有签名指数

$s =$ 199a6985	e9b2bff5	a2841ccc	580fc73a	28a14266
0987eb12	3dbcaeb2	b8ee546d	2356a3a5	7d9c28ed
71e455c4	466cbe30	7787dc5a	9959b747	5a189a8f
038a4741	e4b10153	be08c26e	4401f7ab	6e7e9609
2caf07c0	870b13b6	4f669667	3029ec2c	77aabc39
7fa528a2	45d7073c	e69cc9bd	cd7bef91	599dca48
4000c0bd	8ab0814e			

E. 4.2.1.3 公开验证模数

$N =$ ccd34c2f	4d95ffad	1420e666	c07e39d1	450a1330
4c3f5891	ede57595	c772a369	1ab51d2b	ece1476b
8f22ae22	3365f183	bc3ee2d4	cacdba3a	d0c4d478
1c523a10	efe6203d	6f3bc226	bf9a4597	27b8f122
c482d8c8	6019f9a8	69329187	096430a6	c67cb103
742bcbe6	6906ad23	836ebabb	511d5d80	ab8cb599
74e9aac6	2d785c45			

E. 4.2.2 签名的生成

$M =$  “abc” 的 ASCII 码形式 = 616263

$h(M) =$  a9993e36      4706816a      ba3e2571      7850c26c      9cd0d89d

字符串  $x'33cc'$  后缀于散列权标, 它表示该散列函数是 SHA-1。由半字节  $x'b'$  重复构成的填充部分

(最右边尾部半个字节总是 16 进制值  $x'a'$ ，作为它的散列函数  $h(M)$  的分隔字段)是散列的前缀。它是  
以 16 进制头的值  $x'6'$  为领头的，用于给出一个中间值  $H'$ 。

$H' = 6$	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbaa999	3e364706	816aba3e	25717850	
c26c9cd0	d89d33cc				

由于对于  $n$  来说  $H'$  的 Jacobi 符号  $(H'/N)$  是  $-1$ ， $H'$  在签名前被 2 除以使  $H = H'/2$  的 Jacobi 符号  
为  $+1$ 。

$H = 35$	ddddddd	ddddddd	ddddddd	ddddddd	ddddddd
ddddddd	ddddddd	ddddddd	ddddddd	ddddddd	ddddddd
ddddddd	ddddddd	ddddddd	ddddddd	ddddddd	ddddddd
ddddddd	ddddddd	ddddddd	ddddddd	ddddddd	ddddddd
ddddddd	ddddddd	ddddddd	ddddddd	ddddddd	ddddddd
ddddddd	ddd54cc	9f1b2383	40b55d1f	12b8bc28	
61364e68	6c4e99e6				

现在  $S = H^2 \bmod N$  计算如下：

$S = 232f0e08$	eb9a2395	7646697f	c7884796	d39a04fd
0eff5b72	b60813d4	e6919178	91c96603	876d0879
3aad86da	f2e6187f	f62c226e	81bd6b99	3b27091e
0864895a	f10f222a	eb022961	b444d312	ea3db789
1d4550b2	80cf2469	3d4465b9	57e53cbd	b0f8c29d
2b5ee154	5d6c91a4	5eaaacec	0096d8a5	e4cfe06a
2cd320bd	f853d817			

#### E. 4. 2. 3 签名的验证

计算中间值  $\bar{H}' = S^v \bmod (N)$ 。

$\bar{H}' = 96f56e51$	6fb821cf	36430888	e2a05bf3	562c3552
6e617ab4	100797b7	e994c58b	3cd73f4e	0f03698d
b144d044	558813a5	de6104f6	ecefdc5c	f2e6f69a
3e745c33	1208425f	915de448	e1bc67b9	49db1344
e6a4faea	823c1bca	8b54b3a9	2b8652c8	e89ed325
964dede8	8b295856	e4539738	10680061	98d3f971
13b35c5d	c129c25f			

由于  $\bar{H}'$  以 16 进制值  $x'f'$  结尾，它不是一个以  $x'c'$  或  $x'6'$  结尾的有效证据值，所以计算  
 $\bar{H} = (N - \bar{H}')$

$\bar{H} = 35$	ddddddd	ddddddd	ddddddd	ddddddd	ddddddd
ddddddd	ddddddd	ddddddd	ddddddd	ddddddd	ddddddd
ddddddd	ddddddd	ddddddd	ddddddd	ddddddd	ddddddd
ddddddd	ddddddd	ddddddd	ddddddd	ddddddd	ddddddd
ddddddd	ddddddd	ddddddd	ddddddd	ddddddd	ddddddd
ddddddd	ddd54cc	9f1b2383	40b55d1f	12b8bc28	
61364e68	6c4e99e6				

签名被验证。

## E. 5 ESIGN 签名算法

### E. 5.1 ESIGN 域参数

$$n = 768$$

$$s = v = 1024$$

### E. 5.2 签名密钥和验证密钥

#### E. 5.2.1 签名密钥

签名密钥由两个素数  $P$  和  $Q$  组成, 它们的值(以 16 进制表示)为

$P_1 = P_2 = P =$	fd3764f3	7b98dfe4	8e30b2c4	004e2d03
	0a5e8018	2f94b156	fe6e4b5f	16f902da
	d60e4730	30deab98	75f3d749	de79c361
	8874d195	4102dfe0	47637bab	495c7dc2
	912fdeb9	4b2d5eca	b798e90e	c6e634b7
	b4f1153b	4d9f4bd0	3c45cfc7	2600e549
$P_3 = Q =$	8332d671	713a0dea	71e9453a	b323c499
	2455d957	ef6985a5	3770af04	e1c76529
	a0bc855e	ca025f9c	540cf0b5	3684ea5e
	5777b647	17e78b99	1c2bacb6	9befed40
	f414d805	a1594e56	90ce67f6	42c42714
	7c94balf	2dc9adf8	eacd114b	1723700f

#### E. 5.2.2 验证密钥

验证密钥由验证指数  $v = s = 1024$  和模数组成, 它的值(以 16 进制给出)为

$N =$	805c6554	66eea57c	a1798241	5aalaca7
	Df54ab5c	17953109	9a08cf05	5d6bd99f
	7e5d4ff8	98cb633b	3368dac6	8c3ff751
	1c5ccf45	6adela2	20558dad	17d466df
	f0e7f3b9	3ddd6934	07a18a66	bc74ceb1
	ebac6901	4b6ce22a	78e70676	4ca5de4d
	196c7007	54cb46c7	30f77bc0	bc1955cc
	fb26df7e	4c005dc7	b836acc2	f04e696b
	10578b6d	2cb993f3	4a01fb95	2727517f
	4ab8499a	51829133	16b2fcaa	5c594c3e
	9b8b24ec	313c8863	4b7bbfef	bfdac7eb
	689c79a8	6b5c4401	b7ece53c	ab9f2326
	25c70842	2f5fe450	9631128d	a2775427
	0af91fc9	b09800a0	e4339609	aa9a10b6
	2f6812f1	91a3d598	177001a0	88db58a4
	ad2fef5a	230735e0	0aeb8031	50403d11
	51f15167	65bada30	d57f2b4c	b9438e59
	551828f1	9704aab5	4169f107	e66dae3f

### E. 5.3 ESIGN 签名过程

#### E. 5.3.1 ESIGN 预签名

**E. 5. 3. 1. 1 ESIGN 随机数**

随机数(以 16 进制给出)为:

K=76a4d0dd	5b024775	2d546ca4	27b6e8be
18db2ba7	33842cb7	4399b33f	ca7bfaca
346fcf34	77f20811	5576e1e1	bb3af124
08633c3f	b1928eab	3a1645ae	b58cff4a
9265cc40	8f3f3ad6	8a4ae202	a11511d0
06bb0023	1c86e725	a39af1a6	b1c83f2c
38716dd2	49c82a4f	dc7be305	2c78ffb4
887f7935	ce3932ee	baa8c80b	7491e0b6
38d5f816	3794ec9c	158f088e	1a93bef8
93199aa1	e07bc11f	86fbcf75	76f28b89
261fe806	baff4451	83209223	807f5012
6d4c983c	be96c6de	6acbd5f	9ef1d975

**E. 5. 3. 1. 2 ESIGN 预签名**

预签名为  $\Pi = (U, V)$ , 其中  $U = K^s \bmod N$  和  $V = (sK^{-1})^{-1}$  具有如下值(以 16 进制给出):

U=7350f3fd	13a3e49b	4c7f83ed	334e45e6
28c9aa65	a2a9298c	c6e52b23	fdb1ae1e
2197da72	ae23af02	9241408c	df5287bb
04cf88cc	871721ed	d887a1bc	a8e261f3
69a85e6c	77bf1a97	f411fd5b	5421c276
250c92e4	06954b0b	7fb59209	b8940feb
6a20d4d6	ffee125b	959e8f9f	2486ac2c
9f609561	363b7b7e	3fd93410	94c9d507
3c5075b3	71a41b98	d7e98778	d52922a2
319fed3b	88af194d	841f9837	6f4b905c
e2835b36	1c226bef	b3b2dd84	c8a69b19
6ae5bb51	92b6db42	7e75dd07	a3a2bdf2
8c6aff24	482fdc8b	3592ff0a	e130da0c
513d9d75	31089919	6c94c114	10b90ee8
78fcaa83	02232bbf	17960b74	4e411455
4eb04652	c23b9d13	7f959e06	5499fcf4
7853786b	eaf792d8	b8e76c92	6bc17587
346b2187	d7059cad	9a01df44	475fec58

并且

V=037c592f	20a80f8c	9b296800	12f1d8a8
ede80cde	1a89ae4d	3e73014c	2eca84ae
313c5a34	13388e16	e2ebe89f	42510a45
f68d0417	00ee31f3	f5e3340e	bcd1d226
dbf0b6aa	7d5eabc0	57c90d78	618e2836
28d6eb9e	79d7cef7	82d8cb35	e91f0cb9

**E. 5. 3. 2 ESIGN 消息依赖的数据**

M="abc"的 ASCII 码形式=616263

$h(M) =$  a9993e36      4706816a      ba3e2571      7850c26c      9cd0d89d

字符串  $x'33cc'$  后缀于散列权标, 它表示该散列函数是 SHA-1。由半字节  $x'b'$  重复构成的填充部分 (最右边尾部半个字节总是 16 进制值  $x'a'$ , 作为它与散列函数  $h(M)$  的分隔字段) 是散列的前缀。它是 16 进制头的值  $x'6'$  为领头的, 其中半字节的第三位表明它是一个非零长度的消息。结果生成的散列权标是一个小于  $2^{n-1}$  的整数。其值为 (16 进制):

H=6bbbbbbb	Bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	Bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	Bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	Bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	Bbbbbbbb	bbbaa999	3e364706
816aba3e	25717850	c26c9cd0	d89d33cc

### E. 5. 3. 3 ESIGN 签名

签名  $S = K + (\lfloor (2^{2n}H - U) / PQ \rfloor V \bmod P) PQ \bmod N$  具有如下给出值 (16 进制):

S=53491b7d	54b79d59	a9a16251	e6192609
3336ffb6	368ae323	386360cc	4af5ded3
8a86dd1a	9f3061f7	b66dde43	7d4aa7a1
0b533cb7	89f5c025	d74a4fea	6601fb2e
00241743	fd143a85	6836ba63	d62aa0fe
151636ea	adb8c7c9	cafb5f78	3053227c
0e76bb1b	b889ba2d	73ea27d4	05133979
5502c867	7087de5f	4941f5c8	82a2713f
5f2d0781	ad765763	b930bf8a	0feb7def
1b38696c	0b072aeb	5f9f03d1	44c07c85
5989bc79	9765836d	20299357	b9b636bc
fb778b07	faefbff	57d73a5e	6c35fd4e
a31cc4ae	497ea98e	3e07cc00	0368de91
6559069c	a2362bfc	1b7aff82	32c4fe35
707cc105	e0cf460f	62dc0c99	ecf31551
6bbafacc	b4de790c	f55e384a	1901f624
d351bb3f	d3443467	5f53cf13	6ac986fc
0a71fe11	772ba428	fb09967e	c9b9c8dc

### E. 5. 4 ESIGN 验证

#### E. 5. 4. 1 证据的重新计算

验证方从接收到的消息中计算证据, 方法同 C. 5. 3. 2。

H=6bbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbaa999	3e364706
816aba3e	25717850	c26c9cd0	d89d33cc

#### E. 5. 4. 2 证据的重新构造

重新构造的证据  $\bar{H}$  是值  $S^V \bmod N$  的高端  $n$  位, 它的值给出如下 (16 进制):



$\bar{H}$ =6bbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbbbbbb	bbbbbbbb
bbbbbbbb	bbbbbbbb	bbbaa999	3e364706
816aba3e	25717850	c26c9cd0	d89d33cc

注意,这与前面重新计算的证据  $H$  一致。

**附 录 F**  
**(资料性附录)**  
**所选签名方案具有的特性**

下表提供了各种签名机制的可能具有的优点：

签名机制	具有的特性
数字签名算法	美国 U. S. FIPS; 被广泛接受
Pointcheval/Vaudenay	具有声称可证明安全性的 DSA 变形
椭圆曲线 DSA	ANSI 标准, 期望成为 FIPS, 高效存储, 节省宽带, 高效计算
基于 GB 15851 的带散列的数字签名	确定性签名; 广泛使用, 快速验证
ESIGN	快速

**附 录 G**  
(资料性附录)  
**专利信息**

GB/T 17902 的本部分制定中,等同采用了国际标准 ISO/IEC 14888-3:1998 附录 G 的专利信息。

关于专利的使用遵照国家有关规定。

国际标准的有关专利信息如下:

在该国际标准(ISO/IEC 14888-3:1998)的准备期间,收集了该国际标准的应用所需要的相关专利信息。相关专利如下表所示。但是,关于专利的有效性或范围,ISO/IEC 没有给出权威的或全面的信息。

这些注册专利的持有者已经声明,如果寻求授权许可的使用方同意付费,将在适当的条件下给予许可,使其能够应用本部分。

进一步的信息可从专利持有者那里得到。

专业范围	发明者	专利号	发布日期	联系地址
RSA 系统	Rivest-Shamir-Adleman	US 4,405,829	1983-09-20	RSA Data Security, Inc. Director of Licensing 2955 Campus Drive, Suite 400 San mateo, CA 94403 2507 USA
ESIGN 签名	Okamoto et al	US 4,625,076	1986-11-25	NTT 20-2 Nishi-shinjuku 3-Chome Shinjuku-ku Tokyo 163-1419 Japan
RSA 签名	Kravitz	US 5,231,668	1993-07-27	[不需要许可]

参 考 文 献

- ANSI X9.62—1998, *Public Key Cryptography For the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*
- FIPS PUB 180, 11 May 1993, *Secure Hash Standard*, U. S. National Institute of Standards and Technology, Gaithersburg, Maryland.
- FIPS PUB 180-1, 17 April 1995, *Secure Hash Standard*, Revision 1, U. S. National Institute of Standards and Technology, Gaithersburg, Maryland.
- FIPS PUB 186, 1994, *Digital Signature Standard*, U. S. National Institute of Standards and Technology, Gaithersburg, Maryland.
- FIPS PUB 186 Change Notice, 30 December 1996.
- Koblitz, N., "Elliptic Curve Cryptosystems," *Math. Comp.*, 48(1987), pp. 203-209.
- Miller, V., "Use of Elliptic Curves in Cryptology," *Advances in Cryptology Proceedings of CRYPTO'85*, LNCS 218, Springer-Verlag 1986, pp. 417-425.
- Menezes, A., *«Elliptic Curve Public Key Cryptosystems»*, Kluwer Academic Publishers, 1993.
-

中 华 人 民 共 和 国  
国 家 标 准  
信息技术 安全技术 带附录的数字签名  
第 3 部分：基于证书的机制  
GB/T 17902.3—2005/ISO/IEC 14888-3:1998

\*

中国标准出版社出版发行  
北京复兴门外三里河北街16号  
邮政编码：100045

网址 www.bzcs.com

电话：68523946 68517548

中国标准出版社秦皇岛印刷厂印刷  
各地新华书店经销

\*

开本 880×1230 1/16 印张 2.5 字数 66 千字  
2005年8月第一版 2005年8月第一次印刷

\*

书号：155066·1-23070 定价 18.00 元

如有印装差错 由本社发行中心调换  
版权专有 侵权必究  
举报电话：(010)68533533



GB/T 17902.3-2005