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

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.bzcbs.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

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、叶顶锋、张振峰、黄家英。

目　次

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 $n\geqslant 0$)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
$a\|\|b$	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
$\gcd(U,N)$	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 $0\leqslant U<N$
Z_N^*	整数 U 的集合,满足 $0<U<N$,且 $\gcd(U,N)=1$
$\lfloor a \rfloor$	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。

$$AK+BX+C\equiv 0(\bmod Q)$$

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

$$\prod = G^K$$

本步的输出为预签名 $\prod$。

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 $\prod$ 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

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

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 $\sum$。

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$((R,S),\text{text})$。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 $M\|((R,S),\text{text})$。

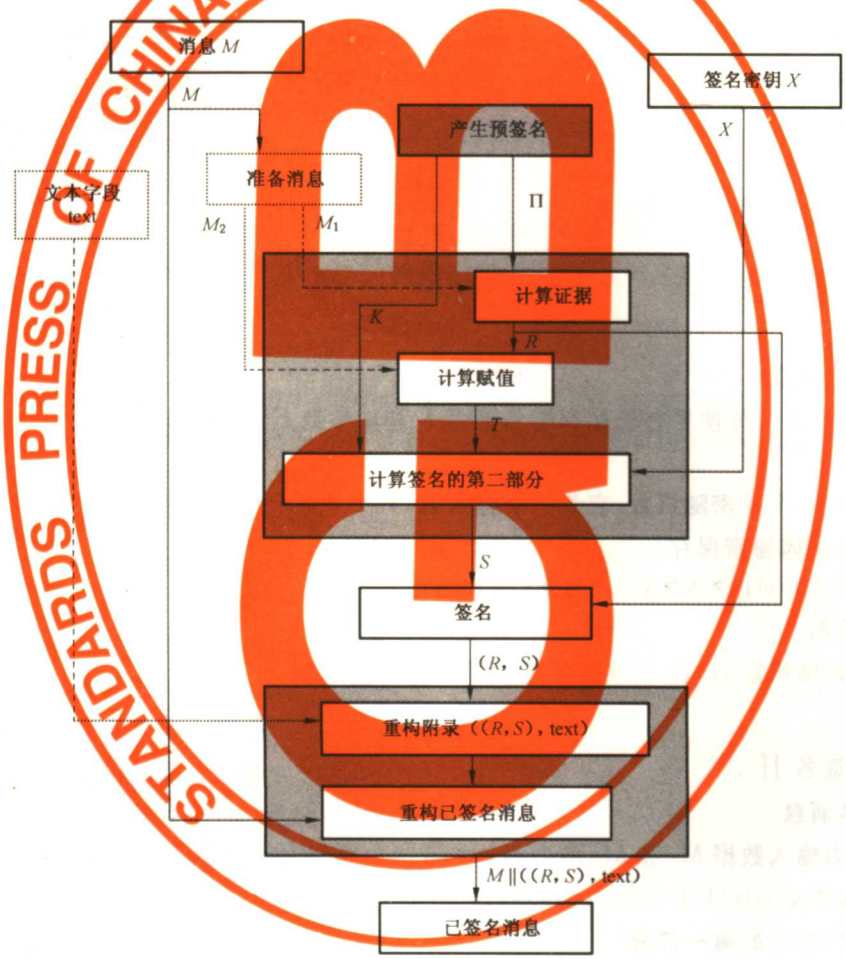

图 1 带随机性证据的签名过程

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。

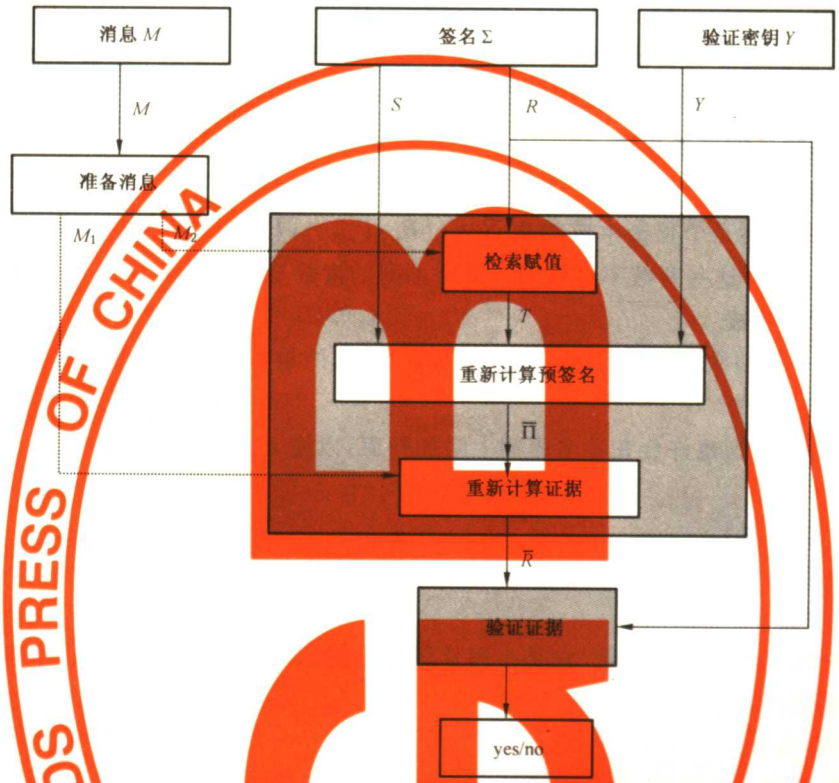

图 2 带随机化证据的验证过程

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 $\overline{\prod}$ 如下：

$$\overline{\prod} = Y^m G^n$$

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

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 $\overline{\prod}$ 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 $X=(\{P_1,P_2,\cdots,P_r\},s)$,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 Π,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 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 Π 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 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 Π 也是有效的输入。输出是一个单部分签名 $\sum = S$。

7.2.5 构造附录

附录是由签名 $\sum$ 和一个可选文本字段 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 $M \| (\sum, \text{text})$。

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 $\overline{H}$。

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 $\overline{H}$ 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 $0 \leqslant U < P$
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 $R = \prod \bmod Q$ 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 $0 \leqslant I < 8$ 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 $F^{(P-1)/Q} \bmod P > 1$ 的整数
G	$F^{(P-1)/Q} \bmod P$，$E = Z_P^*$ 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：

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

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 $R=\prod \bmod Q$，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，

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

$$S=(K^{-1}(h(M)+XR)) \bmod Q$$

$h(M)$ 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 串接后得到的，为 $(R,S)\|\text{text}$。

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 $M\|(R,S)\|\text{text}$。

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 $\overline{\prod}$：

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

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 $\overline{\prod}$。注意 M_1 为空。输出为重新计算证据 $\overline{R}$。

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 $\overline{R}$ 与 A.1.1.4.2 中的 R 值进行比较。如果 $\overline{R}=R$，则签名有效。

A.1.2 Pointcheval/Vaudenay 签名

Pointcheval/vaudenay 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

$$R = \prod \bmod Q$$

定义，而赋值函数由公式

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

得到，式中 $H=h(R\|H)$ 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/vaudenay 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：

$$SK - RX - H \equiv 0 (\bmod Q)$$

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

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

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 $R = \prod \bmod Q$，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，

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

$$S=K^{-1}(h(R\|M)+XR) \bmod Q$$

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 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 $\overline{\prod}$：

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

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 $\overline{\prod}$ 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。

因此签名公式变成为:

$$SK - RX + H \equiv 0 (\mathrm{mod}\ Q)$$

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

$$\prod = G^K$$

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 $R = \prod_x \mathrm{mod}\ Q$,其中 $\prod_x$ 是点 $\prod$ 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,

$$R = \prod_x \mathrm{mod}\ Q$$

$$S = (K^{-1}(XR - H)) \mathrm{mod}\ Q$$

从而

$$(R,S) = ((\prod_x) \mathrm{mod}\ Q, (K^{-1}(XR - H)) \mathrm{mod}\ Q)$$

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 串接后得到的,为$(R,S)\|\text{text}$。