

SM9 标识密码算法

第 2 部分：数字签名算法

目 次

1 术语和定义.....	2
2 符号.....	2
3 算法参数与辅助函数.....	3
3.1 总则.....	3
3.2 系统参数组.....	3
3.3 系统签名主密钥和用户签名密钥的产生.....	4
3.4 辅助函数.....	4
3.4.1 概述.....	4
3.4.2 密码杂凑函数.....	4
3.4.2.1 密码杂凑函数 $H_v()$	4
3.4.2.2 密码函数 $H_1()$	4
3.4.2.3 密码函数 $H_2()$	4
3.4.3 随机数发生器.....	5
4 数字签名生成算法及流程.....	5
4.1 数字签名生成算法.....	5
4.2 数字签名生成算法流程.....	5
5 数字签名验证算法及流程.....	6
5.1 数字签名验证算法.....	6
5.2 数字签名验证算法流程.....	6

SM9 标识密码算法

第 2 部分：数字签名算法

本部分规定了用椭圆曲线对实现的基于标识的数字签名算法，包括数字签名生成算法和验证算法，并给出了数字签名与验证算法及其相应的流程。

1 术语和定义

1.1

消息 message

任意有限长度的比特串。

1.2

签名消息 signed message

由消息以及该消息的数字签名部分所组成的一组数据元素。

1.3

签名密钥 signature key

在数字签名生成过程中由签名者专用的秘密数据元素，即签名者的私钥。

1.4

签名主密钥 signature master key

处于标识密码密钥分层结构最顶层的密钥，包括签名主私钥和签名主公钥，其中签名主公钥公开，签名主私钥由 KGC 秘密保存。KGC 用签名主私钥和用户的标识生成用户的签名私钥。在标识密码中，签名主私钥一般由 KGC 通过随机数发生器产生，签名主公钥由签名主私钥结合系统参数产生。

1.5

标识 identity

可唯一确定一个实体身份的信息。标识应由实体无法否认的信息组成，如实体的可识别名称、电子邮箱、身份证号、电话号码、街道地址等。

1.6

密钥生成中心 key generation center; KGC

在本部分中，负责选择系统参数、生成签名主密钥并产生用户签名私钥的可信机构。

2 符号

下列符号适用于本部分。

A, B: 使用标识密码系统的两个用户。

cf: 椭圆曲线阶相对于 N 的余因子。

cid : 用一个字节表示的曲线的识别符, 其中 0x10 表示 F_p (素数 $p > 2^{191}$) 上常曲线(即非超奇异曲线), 0x11 表示 F_p 上超奇异曲线, 0x12 表示 F_p 上常曲线及其扭曲曲线。

ds_A : 用户 A 的签名私钥。

e : 从 $G_1 \times G_2$ 到 G_T 的双线性对。

eid : 用一个字节表示的双线性对 e 的识别符, 其中 0x01 表示 Tate 对, 0x02 表示 Weil 对, 0x03 表示 Ate 对, 0x04 表示 R-Ate 对。

G_T : 阶为素数 N 的乘法循环群。

G_1 : 阶为素数 N 的加法循环群。

G_2 : 阶为素数 N 的加法循环群。

g^u : 乘法群 G_T 中元素 g 的 u 次幂, 即 $g^u = g \cdot g \cdot \dots \cdot g$, u 是正整数。
 u 个

$H_v()$: 密码杂凑函数。

$H_1(), H_2()$: 由密码杂凑函数派生的密码函数。

hid : 在本部分中, 用一个字节表示的签名私钥生成函数识别符, 由 KGC 选择并公开。

(h, S) : 发送的签名。

(h', S') : 收到的签名。

ID_A : 用户 A 的标识, 可以唯一确定用户 A 的公钥。

M : 待签名消息。

M' : 待验证消息。

$\text{mod } n$: 模 n 运算。例如, $23 \text{ mod } 7 = 2$ 。

N : 循环群 G_1 、 G_2 和 G_T 的阶, 为大于 2^{191} 的素数。

P_{pub-s} : 签名主公钥。

P_1 : 群 G_1 的生成元。

P_2 : 群 G_2 的生成元。

ks : 签名主私钥。

$\langle P \rangle$: 由元素 P 生成的循环群。

$[u]P$: 加法群 G_1 、 G_2 中元素 P 的 u 倍。

$\lceil x \rceil$: 顶函数, 不小于 x 的最小整数。例如, $\lceil 7 \rceil = 7, \lceil 8.3 \rceil = 9$ 。

$\lfloor x \rfloor$: 底函数, 不大于 x 的最大整数。例如, $\lfloor 7 \rfloor = 7, \lfloor 8.3 \rfloor = 8$ 。

$x \parallel y$: x 与 y 的拼接, x 和 y 是比特串或字节串。

$[x, y]$: 不小于 x 且不大于 y 的整数的集合。

β : 扭曲曲线参数。

3 算法参数与辅助函数

3.1 总则

本部分规定了一个用椭圆曲线对实现的基于标识的数字签名算法。该算法的签名者持有一个标识和一个相应的签名私钥, 该签名私钥由密钥生成中心通过签名主私钥和签名者的标识结合产生。签名者用自身签名私钥对数据产生数字签名, 验证者用签名者的标识验证签名的可靠性。

在签名的生成和验证过程之前, 都要用密码杂凑函数对待签消息 M 和待验证消息 M' 进行压缩。

3.2 系统参数组

系统参数组包括曲线识别符 cid ；椭圆曲线基域 F_q 的参数；椭圆曲线方程参数 a 和 b ；扭曲曲线参数 β (若 cid 的低 4 位为 2)；曲线阶的素因子 N 和相对于 N 的余因子 cf ；曲线 $E(F_q)$ 相对于 N 的嵌入次数 k ； $E(F_{q^{d_1}})$ (d_1 整除 k) 的 N 阶循环子群 \mathcal{G}_1 的生成元 P_1 ； $E(F_{q^{d_2}})$ (d_2 整除 k) 的 N 阶循环子群 \mathcal{G}_2 的生成元 P_2 ；双线性对 e 的识别符 eid ；(选项) \mathcal{G}_2 到 \mathcal{G}_1 的同态映射 ψ 。

双线性对 e 的值域为 N 阶乘法循环群 \mathcal{G}_T 。

3.3 系统签名主密钥和用户签名密钥的产生

KGC 产生随机数 $ks \in [1, N-1]$ 作为签名主私钥，计算 \mathcal{G}_2 中的元素 $P_{pub-s} = [ks]P_2$ 作为签名主公钥，则签名主密钥对为 (ks, P_{pub-s}) 。KGC 秘密保存 ks ，公开 P_{pub-s} 。

KGC 选择并公开用一个字节表示的签名私钥生成函数识别符 hid 。

用户 A 的标识为 ID_A ，为产生用户 A 的签名私钥 ds_A ，KGC 首先在有限域 F_N 上计算 $t_1 = H_1(ID_A || hid, N) + ks$ ，若 $t_1 = 0$ 则需重新产生签名主私钥，计算和公开签名主公钥，并更新已有用户的签名私钥；否则计算 $t_2 = ks \cdot t_1^{-1} \bmod N$ ，然后计算 $ds_A = [t_2]P_1$ 。

3.4 辅助函数

3.4.1 概述

在本部分规定的基于标识的数字签名算法中，涉及到两类辅助函数：密码杂凑函数与随机数发生器。

3.4.2 密码杂凑函数

3.4.2.1 密码杂凑函数 $H_v()$

密码杂凑函数 $H_v()$ 的输出是长度恰为 v 比特的杂凑值。本部分规定使用国家密码管理主管部门批准的密码杂凑函数，如 SM3 密码杂凑算法。

3.4.2.2 密码函数 $H_1()$

密码函数 $H_1(Z, n)$ 的输入为比特串 Z 和整数 n ，输出为一个整数 $h_1 \in [1, n-1]$ 。 $H_1(Z, n)$ 需要调用密码杂凑函数 $H_v()$ 。关于密码杂凑函数 $H_v()$ ，应符合本部分 3.4.2.1 的规定。

密码函数 $H_1(Z, n)$ ：

输入： 比特串 Z ，整数 n 。

输出： 整数 $h_1 \in [1, n-1]$ 。

步骤 1：初始化一个 32 比特构成的计数器 $ct = 0x00000001$ ；

步骤 2：计算 $hlen = 8 \times \lceil (5 \times (\log_2 n)) / 32 \rceil$ ；

步骤 3：对 i 从 1 到 $\lceil hlen/v \rceil$ 执行：

步骤 3.1：计算 $Ha_i = H_v(0x01 || Z || ct)$ ；

步骤 3.2： $ct++$ ；

步骤 4：若 $hlen/v$ 是整数，令 $Ha!_{\lceil hlen/v \rceil} = Ha_{\lceil hlen/v \rceil}$ ，

否则令 $Ha!_{\lceil hlen/v \rceil}$ 为 $Ha_{\lceil hlen/v \rceil}$ 最左边的 $(hlen - (v \times \lfloor hlen/v \rfloor))$ 比特；

步骤 5：令 $Ha = Ha_1 || Ha_2 || \dots || Ha_{\lceil hlen/v \rceil-1} || Ha!_{\lceil hlen/v \rceil}$ ，按本文第 1 部分 6.2.4 和 6.2.3 给出的细节将 Ha 的数据类型转换为整数；

步骤 6：计算 $h_1 = (Ha \bmod (n-1)) + 1$ 。

3.4.2.3 密码函数 $H_2()$

密码函数 $H_2(Z, n)$ 的输入为比特串 Z 和整数 n ，输出为一个整数 $h_2 \in [1, n-1]$ 。 $H_2(Z, n)$ 需要调用密码杂凑函数 $H_v()$ 。关于密码杂凑函数 $H_v()$ ，应符合本部分 3.4.2.1 的规定。

密码函数 $H_2(Z, n)$ ：

输入： 比特串 Z ，整数 n 。

输出： 整数 $h_2 \in [1, n-1]$ 。

步骤 1: 初始化一个 32 比特构成的计数器 $ct=0x00000001$;

步骤 2: 计算 $hlen=8\times\lceil(5\times(\log_2 n))/32\rceil$;

步骤 3: 对 i 从 1 到 $\lceil hlen/v\rceil$ 执行:

 步骤 3.1: 计算 $Ha_i=H_v(0x02\|Z\|ct)$;

 步骤 3.2: $ct++$;

步骤 4: 若 $hlen/v$ 是整数, 令 $Ha!\lceil hlen/v\rceil=Ha_{\lceil hlen/v\rceil}$,
 否则令 $Ha!\lceil hlen/v\rceil$ 为 $Ha_{\lceil hlen/v\rceil}$ 最左边的 $(hlen-(v\times\lfloor hlen/v\rfloor))$ 比特;

步骤 5: 令 $Ha=Ha_1\|Ha_2\|\dots\|Ha_{\lceil hlen/v\rceil-1}\|Ha!\lceil hlen/v\rceil$, 按本标准第 1 部分 6.2.4 和 6.2.3 给出的细节将 Ha 的数据类型转换为整数;

步骤 6: 计算 $h_2=(Ha \bmod (n-1))+1$ 。

3.4.3 随机数发生器

本部分规定使用国家密码管理主管部门批准的随机数发生器。

4 数字签名生成算法及流程

4.1 数字签名生成算法

设待签名的消息为比特串 M , 为了获取消息 M 的数字签名 (h, S) , 作为签名者的用户 A 应实现以下运算步骤:

- A1: 计算群 G_T 中的元素 $g=e(P_1, P_{pub-s})$;
- A2: 产生随机数 $r\in[1, N-1]$;
- A3: 计算群 G_T 中的元素 $w=g^r$, 将 w 的数据类型转换为比特串;
- A4: 计算整数 $h=H_2(M\|w, N)$;
- A5: 计算整数 $l=(r-h) \bmod N$, 若 $l=0$ 则返回 A2;
- A6: 计算群 G_1 中的元素 $S=[l]ds_A$;
- A7: 消息 M 的签名为 (h, S) 。

4.2 数字签名生成算法流程

数字签名生成算法流程如图1。

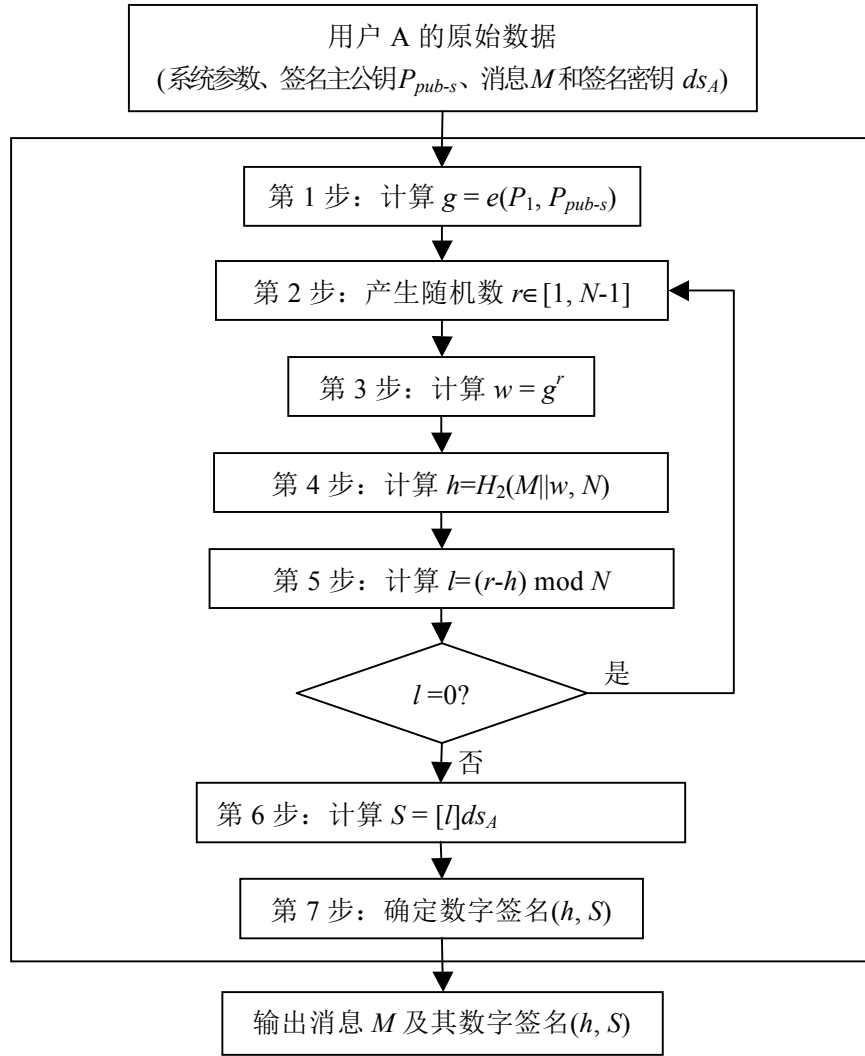


图1 数字签名生成算法流程

5 数字签名验证算法及流程

5.1 数字签名验证算法

为了检验收到的消息 M' 及其数字签名 (h', S') ，作为验证者的用户 B 应实现以下运算步骤：

- B1: 检验 $h' \in [1, N-1]$ 是否成立，若不成立则验证不通过；
- B2: 将 S' 的数据类型转换为椭圆曲线上的点，检验 $S' \in \mathcal{G}_1$ 是否成立，若不成立则验证不通过；
- B3: 计算群 \mathcal{G}_T 中的元素 $g = e(P_1, P_{pub-s})$ ；
- B4: 计算群 \mathcal{G}_T 中的元素 $t = g^{h'}$ ；
- B5: 计算整数 $h_1 = H_1(ID_A || h', N)$ ；
- B6: 计算群 \mathcal{G}_2 中的元素 $P = [h_1]P_2 + P_{pub-s}$ ；
- B7: 计算群 \mathcal{G}_T 中的元素 $u = e(S', P)$ ；
- B8: 计算群 \mathcal{G}_T 中的元素 $w' = u \cdot t$ ，将 w' 的数据类型转换为比特串；
- B9: 计算整数 $h_2 = H_2(M' || w', N)$ ，检验 $h_2 = h'$ 是否成立，若成立则验证通过；否则验证不通过。

5.2 数字签名验证算法流程

数字签名验证算法流程如图2。

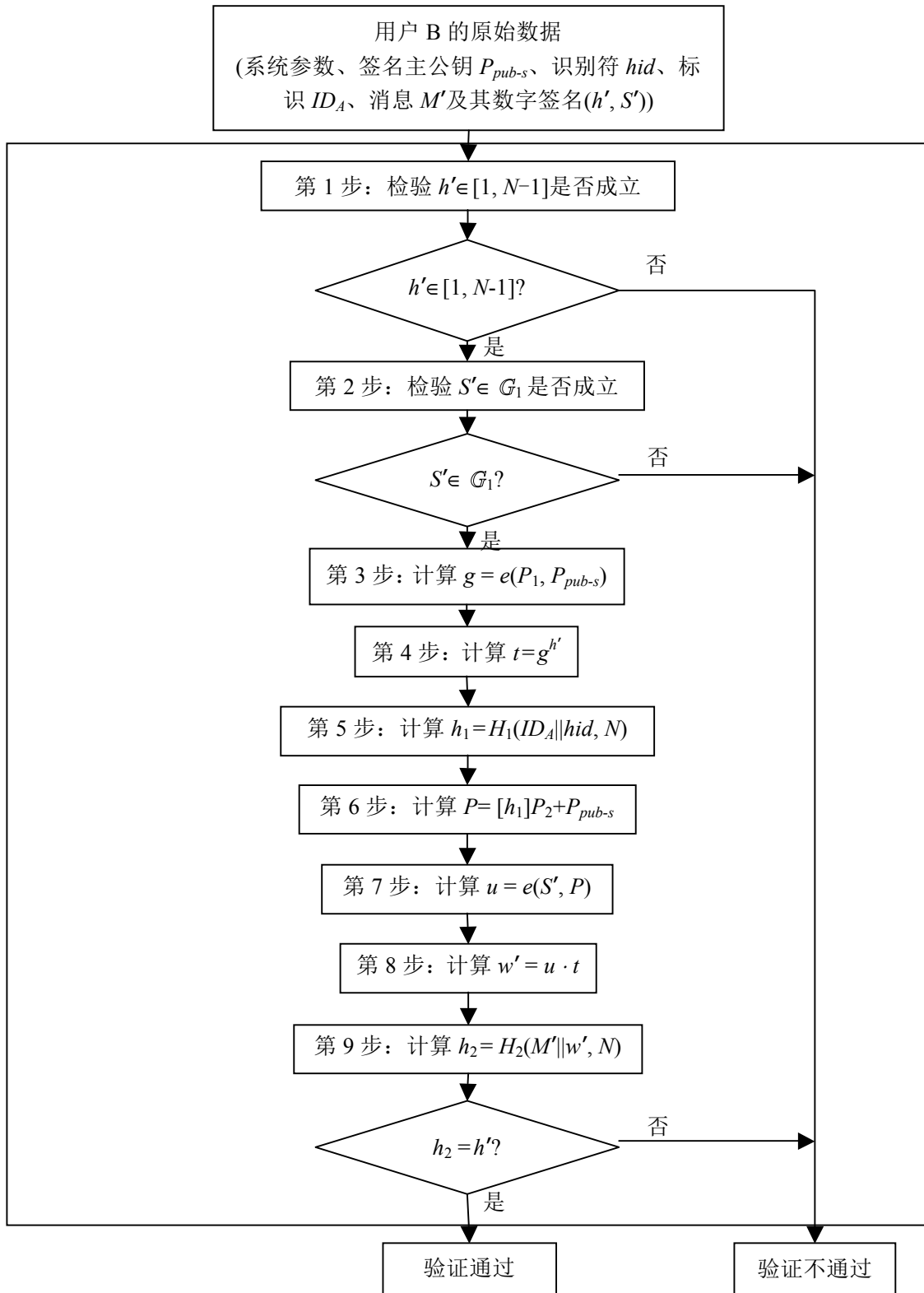


图2 数字签名验证算法流程