5.4 辅助函数

5.4.1 概述

在本部分规定的椭圆曲线数字签名算法中，涉及到两类辅助函数：密码杂凑函数与随机数发生器。

5.4.2 密码杂凑函数

本部分规定使用国家密码管理局批准的密码杂凑算法，如SM3密码杂凑算法。

5.4.3 随机数发生器

本部分规定使用国家密码管理局批准的随机数发生器。

5.5 用户其它信息

作为签名者的用户A具有长度为$entlen_A$比特的可辨别标识ID_A，记$ENTL_A$是由整数$entlen_A$转换而成的两个字节，在本部分规定的椭圆曲线数字签名算法中，签名者和验证者都需要用密码杂凑函数求得用户A的杂凑值Z_A。按本文本第1部分4.2.5和4.2.4给出的细节，将椭圆曲线方程参数*a*、*b*、*G*的坐标x_G、y_G 和P_A的坐标x_A、y_A的数据类型转换为比特串，Z_A=$H_{256}(ENTL_A \parallel ID_A \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$。

6 数字签名的生成算法及流程

6.1 数字签名的生成算法

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

A1：置$\overline{M}$=$Z_A \parallel M$；

A2：计算$e = H_v(\overline{M})$，按本文本第1部分4.2.3和4.2.2给出的细节将*e*的数据类型转换为整数；

A3：用随机数发生器产生随机数$k \in$[1,n-1]；

A4：计算椭圆曲线点(x_1,y_1)=[*k*]*G*，按本文本第1部分4.2.7给出的细节将x_1的数据类型转换为整数；

A5：计算*r*=(*e*+x_1) mod*n*，若*r*=0或*r*+*k*=*n*则返回A3；

A6：计算$s = ((1 + d_A)^{-1} \cdot (k - r \cdot d_A))$ mod*n*，若*s*=0则返回A3；

A7：按本文本第1部分4.2.1给出的细节将*r*、*s*的数据类型转换为字节串，消息M 的签名为(*r*,*s*)。

注：数字签名生成过程的示例参见附录A。

6.2 数字签名生成算法流程

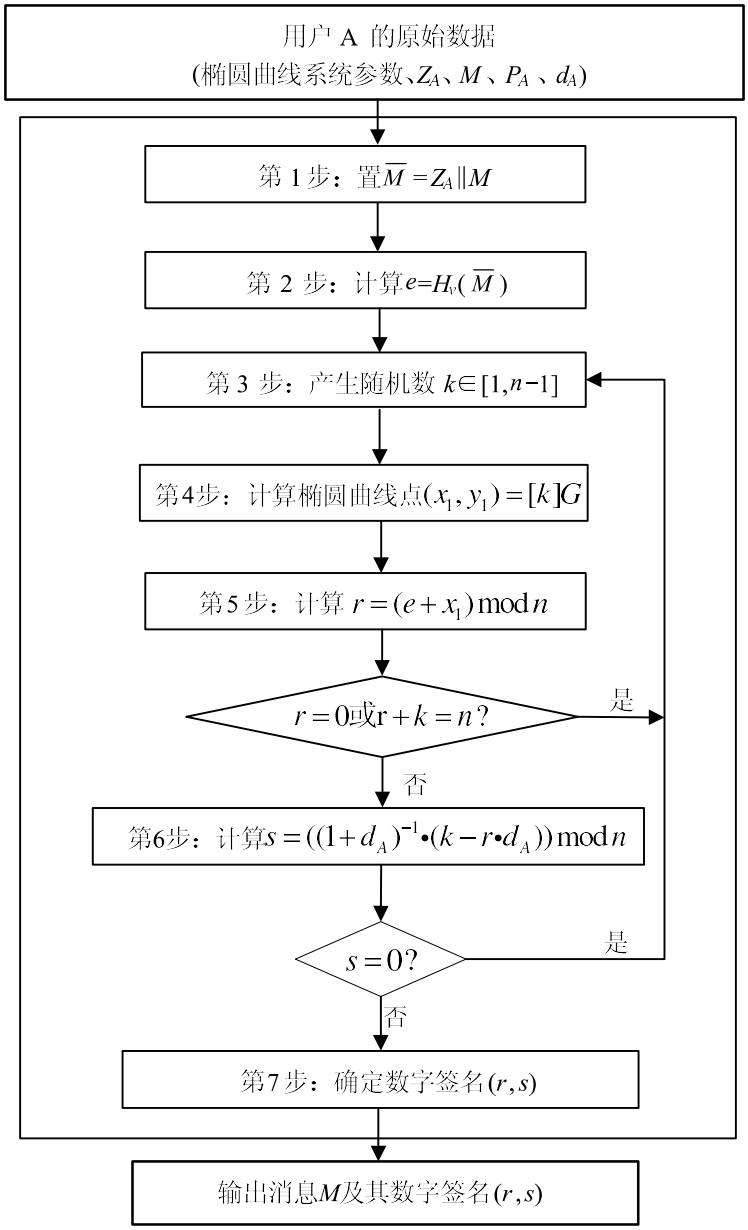

图1　数字签名生成算法流程

7　数字签名的验证算法及流程

7.1　数字签名的验证算法

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

B1：检验$r'\in[1,n\text{-}1]$是否成立，若不成立则验证不通过；

B2：检验$s'\in[1,n\text{-}1]$是否成立，若不成立则验证不通过；

B3：置$\overline{M}'=Z_A\|M'$；

B4：计算$e'=H_v(\overline{M}')$，按本文本第1部分4.2.3和4.2.2给出的细节将e'的数据类型转换为整数；

B5：按本文本第1部分4.2.2给出的细节将r'、s'的数据类型转换为整数，计算$t=(r'+s')\bmod n$，若$t=0$，则验证不通过；

B6：计算椭圆曲线点$(x_1',y_1')=[s']G+[t]P_A$；

B7：按本文本第1部分4.2.7给出的细节将x_1'的数据类型转换为整数，计算$R=(e'+x_1') \bmod n$，检验$R=r'$是否成立，若成立则验证通过；否则验证不通过。

注：如果Z_A不是用户A所对应的杂凑值，验证自然通不过。数字签名验证过程的示例参见附录A。

7.2 数字签名验证算法流程

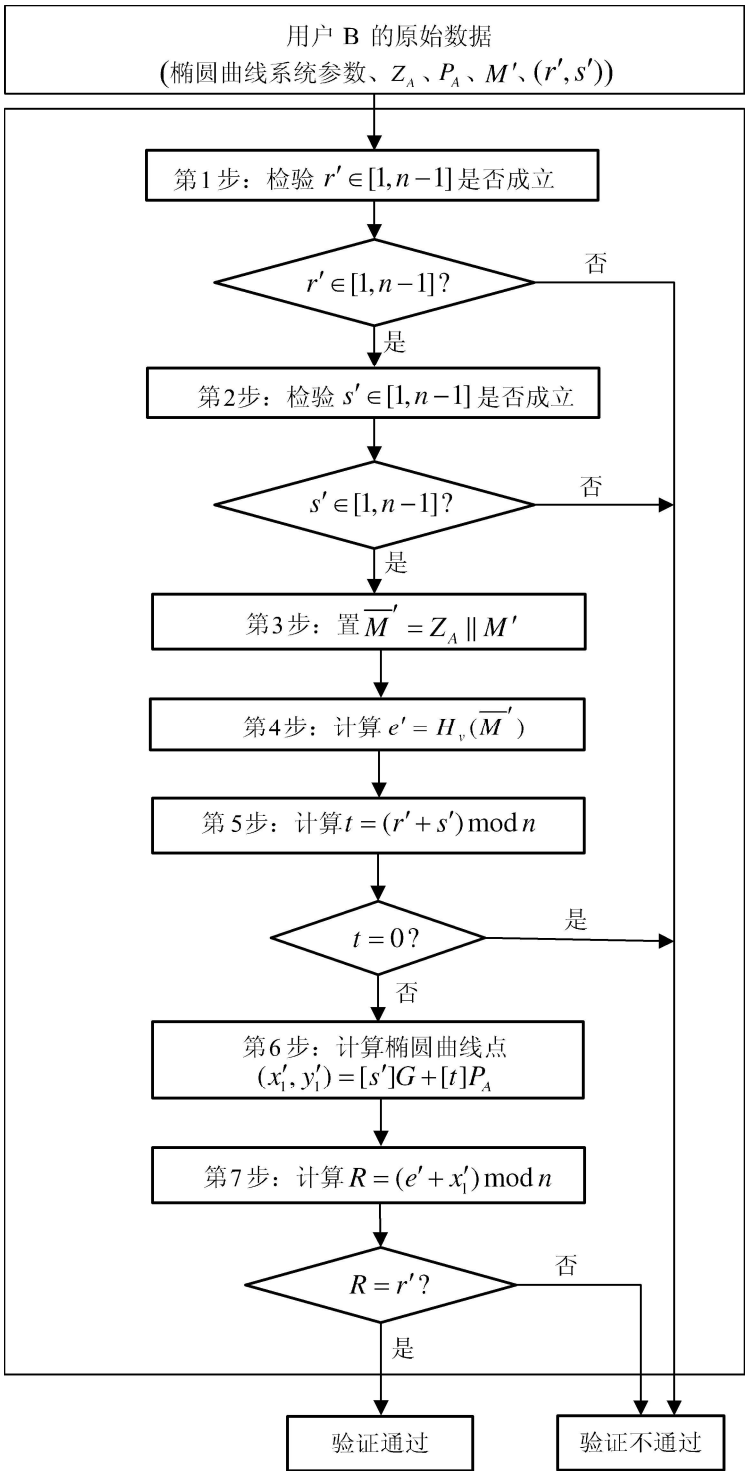

图2 数字签名验证算法流程