ICS 35.040
L 80

中华人民共和国国家标准

GB/T 15851.3—2018
代替 GB/T 15851—1995

信息技术　安全技术　带消息恢复的数字签名方案　第3部分：基于离散对数的机制

Information technology—Security techniques—Digital signature schemes giving message recovery—Part 3：Discrete logarithm based mechanisms

（ISO/IEC 9796-3：2006，MOD）

2018-12-28 发布　　2019-07-01 实施

国家市场监督管理总局
中国国家标准化管理委员会　发布

国家图书馆 专用

目　　次

国家图书馆 专用

国家图书馆 专用

前　言

GB/T 15851《信息技术　安全技术　带消息恢复的数字签名方案》目前分为以下部分：

——第2部分：基于大数分解的机制；

——第3部分：基于离散对数的机制。

本部分为GB/T 15851的第3部分。

本部分按照GB/T 1.1—2009给出的规则起草。

本部分代替GB/T 15851—1995《信息技术　安全技术　带消息恢复的数字签名方案》，与GB/T 15851—1995相比，主要技术差异如下：

——补充规定了密钥生成过程；

——除带消息恢复的数字签名框架外，新定义了五种数字签名方案，并规定了每种签名方案的签名和验证方法；

——补充规定了杂凑函数的用法；

——部分签名方案增加使用椭圆曲线或有限域；

——新增对完全和部分消息恢复的说明；

——新增规范性引用文件；

——使用8个新增附录替换原附录A和附录B，用于描述密钥导出函数及数字签名方案的说明实例等。

本部分使用重新起草法修改采用ISO/IEC 9796-3:2006《信息技术　安全技术　带消息恢复的数字签名方案　第3部分：基于离散对数的机制》及其勘误。

本部分与ISO/IEC 9796-3:2006相比存在结构变化，删除了第11章，将第12章改为第11章，第13章改为第12章，增加8.3.6、9.3.6、10.3.6、11.3.6和12.3.6。

本部分与ISO/IEC 9796-3:2006的技术性差异及其原因如下：

——删除了第11章及其相关内容，以与我国技术水平相适应。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本部分由全国信息安全标准化技术委员会(SAC/TC 260)提出并归口。

本部分起草单位：西安西电捷通无线网络通信股份有限公司、无线网络安全技术国家工程实验室、中关村无线网络安全产业联盟、中国电子技术标准化研究院、重庆邮电大学、中国电子科技集团公司第三十研究所、国家密码管理局商用密码检测中心、国家无线电监测中心检测中心、北京大学深圳研究生院、天津市无线电监测站、中国人民解放军信息安全测评认证中心、北京计算机技术及应用研究所、福建省无线电监测站、国家信息技术安全研究中心、北京数字认证股份有限公司、中国电信股份有限公司上海研究院、工业和信息化部宽带无线IP标准工作组等。

本部分主要起草人：王月辉、杜志强、李琴、曹军、黄振海、李大为、宋起柱、许玉娜、张璐璐、龙昭华、李明、铁满霞、张变玲、李楠、朱跃生、李广森、颜湘、张国强、童伟刚、万洪涛、朱正美、陈志宇、葛培勤、侯鹏亮、许福明、高波、郑骊。

本部分所代替标准的历次版本发布情况为：

——GB/T 15851—1995。

国家图书馆专用

引　言

数字签名机制可以提供实体鉴别、数据源鉴别、不可抵赖和数据完整性服务。

数字签名机制满足以下要求：

——如果只有公开验证密钥，没有私有签名密钥，对任何给定的消息生成一个有效的签名在计算上是不可行的；

——签名方已生成的签名既不能用来为一个新消息生成一个有效的签名，也不能用来恢复签名密钥；

——对于签名者，找到带有相同签名的两个不同的消息在计算上是不可行的。

大部分签名机制是基于非对称密码技术，包括下列三个基本操作：

——生成密钥对的过程，每个密钥对包括一个私有签名密钥和相对应的公开验证密钥；

——使用私有签名密钥的过程，称作签名生成过程；

——使用公开验证密钥的过程，称作签名验证过程。

数字签名机制有两类：

——对于给定的私有签名密钥，签名者对相同消息的签名是相同的，这种机制称为非随机的（或确定的）[参见 ISO/IEC 14888-1]；

——对于给定的消息和给定的私有签名密钥，每个签名处理过程生成的签名是不同的，这种机制称为随机的。

本部分规定了随机的数字签名机制。

数字签名方案也可以分成下列两类：

——当整个消息需要存储和/或随着签名一起传输时，该方案称作“带附录的签名方案”[参见 ISO/IEC 14888]；

——从签名中可以恢复整个或者部分消息时，该方案称作“带消息恢复的签名方案”。

如果消息足够短，则整个消息可包括在签名里并通过签名验证过程从签名中恢复；否则消息的一部分可包括在签名里，余下部分存储起来或随着签名一起传输。本部分规定的机制提供完整或者部分恢复，目的在于降低存储和传输管理。

本部分包括五个签名方案。本部分规定的方案使用杂凑函数对整个消息进行运算。ISO/IEC 10118 规定了杂凑函数。在本部分中规定的一些方案使用有限域上的椭圆曲线一个群。ISO/IEC 15946-1:2002 描述了基于有限域上的椭圆曲线实现密码系统的数学背景和基本技术。本部分所定义机制的特点参见附录 G，这些机制与 ISO/IEC 9796-3:2000 以及 ISO/IEC 15946-4:2004 所定义机制的对应关系参见附录 H。

国家图书馆 专用

信息技术　安全技术　带消息恢复的数字签名方案　第3部分：基于离散对数的机制

1　范围

GB/T 15851的本部分规定了五种带消息恢复功能的数字签名方案。这些方案的安全性是基于定义在有限域或有限域上的椭圆曲线的离散对数问题的难度。

本部分也定义了在杂凑权标里的一个可选控制字段，其能够增强签名的安全性。

本部分规定了随机机制。

在本部分中规定的机制能够完全或者部分恢复消息。

注：带附录的基于离散对数的数字签名方案参见ISO/IEC 14888-3。

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。

ISO/IEC 10118(所有部分)　信息技术　安全技术　散列函数(Information technology—Security techniques—Hash-functions)

ISO/IEC 15946-1　信息技术　安全技术　基于椭圆曲线的密码技术　第1部分：总则(Information technology—Security techniques—Cryptographic techniques based on elliptic curves—Part 1：General)

ISO/IEC 15946-5　信息技术　安全技术　基于椭圆曲线的密码技术　第5部分：椭圆曲线生成(Information technology—Security techniques—Cryptographic techniques based on elliptic curves—Part 5：Elliptic curve generation)

3　术语和定义

下列术语和定义适用于本文件。

3.1

数据输入　data input

取决于完整消息或部分消息的八位位组串，其组成了签名生成过程的一部分输入。

3.2

域参数　domain parameter

常见且已知的或者可以被域中所有实体访问的数据项。

[ISO/IEC 14888-1：1998]

注：域参数集合可以包括数据项，诸如杂凑函数标识、杂凑权标的长度、消息中可恢复部分的最大长度、有限域参数、椭圆曲线参数或者其他能够表明域的安全策略的参数。

国家图书馆专用

3.3

椭圆曲线　elliptic curve

椭圆曲线点 $P=(x,y)$ 与所定义的无穷远点 o 构成椭圆曲线点集合，其中 x，y 是给定有限域上的域元素，其满足给定的非奇异、三次方椭圆曲线方程。

[ISO/IEC 15946-1:2002]

注：给定有限域上的椭圆曲线的数学定义，参见附录 A 中 A.4。

3.4

显式给定有限域　explicitly given finite field

$[0,p-1]$ 上的 e 元组集合，其中 p 为素数，$e\geq 1$，并能计算出乘法表。

注 1：给定有限域上的数学定义，参见 A.3。

注 2：有限域的更多细节参见 ISO/IEC 15946-1:2002。

3.5

杂凑码　hash-code

杂凑函数输出的八位位组串。

注：改编自 ISO/IEC 10118-1:2000。

3.6

杂凑函数　hash-function

将八位位组字符串映射为固定长度的八位位组字符串的函数，该函数满足下列两特性：

——对于给定输出，找出映射为该输出的输入，在计算上是不可行的；

——对于给定输入，找出映射为同一输出的第二个输入，在计算上是不行的。

注 1：改编自 ISO/IEC 10118-1:2000。

注 2：计算上的可行性取决于特定安全要求和环境。

注 3：为了达到本部分的目的，ISO/IEC 10118-2 和 ISO/IEC 10118-3 中描述的杂凑函数具有如下限制：在 ISO/IEC 10118 中描述的杂凑函数将比特串映射成比特串，而在本部分里，杂凑函数将八位位组串映射成八位位组串。因此在本部分中使用的杂凑函数输出长度在比特上只能是 8 的倍数，八位位组串和比特串之间的映射受到 OS2BSP 和 BS2OSP 的影响。

3.7

杂凑权标　hash-token

一个杂凑码与一个可选的控制字段拼接而成的消息，该控制字段可用于标识所指杂凑函数和填充方法。

[ISO/IEC 14888-1:1998]

注：除非杂凑函数是由签名机制或由域参数唯一确定的，否则可给出带有杂凑函数标识符的控制字段。

3.8

消息　message

任意有限长度的八位位组串。

3.9

参数生成过程　parameter generation process

给出输出域参数和用户密钥的过程。

3.10

预签名　pre-signature

用于签名生成过程计算的八位位组串，此八位位组串是对随机数发生器生成的随机数进行运算的结果，与消息无关。

国家图书馆专用

3.11

私有签名密钥　private signature key

一种特定于某一实体的秘密数据项，在签名生成过程中只能由该实体使用。

3.12

公开验证密钥　public verification key

一种数据项，在数学上与私有签名密钥相对应，可为所有实体所知，并由验证方在签名验证过程中使用。

3.13

随机的　randomized

随机的程度取决于随机数发生器。

3.14

随机数发生器　randomizer

在预签名生成过程中由签名实体生成的秘密整数，其不被其他实体所知。

注：改编自 ISO/IEC 14888-1:1998。

3.15

签名　signature

签名生成过程生成的一个八位位组串和整数对。用于提供鉴别。

注：改编自 ISO/IEC 14888-1:1998。

3.16

签名生成过程　signature generation process

输入消息、签名密钥和域参数，输出签名的过程。

注：改编自 ISO/IEC 14888-1:1998。

3.17

签名验证过程　signature verification process

其输入为已签署的消息、验证密钥和域参数，其输出为恢复后的消息（如果有效）的过程。

注：改编自 ISO/IEC 14888-1 验证过程的定义。

3.18

已签消息　signed message

一组由签名中不能恢复的消息部分、签名以及可选的文本字段等数据项组成的集合。

[ISO/IEC 14888-1:1998]

3.19

用户密钥　user keys

一组私有签名密钥和公开验证密钥的数据项。

4　缩略语和符号、约定

4.1　缩略语和符号

下列缩略语和符号适用于本文件。

A　　实体，通常是签名方

B　　实体，通常是验证方

国家图书馆 专用

d	数据输入(八位位组串)
d'	可恢复的数据输入(八位位组串)
E	显式给定有限域上的的椭圆曲线
ECKNR	椭圆曲线 KCDSA/Nyberg-Rueppel 消息恢复签名
ECMR	椭圆曲线 Miyaji 消息恢复签名
ECNR	椭圆曲线 Nyberg-Rueppel 消息恢复签名
ECPV	椭圆曲线 Pintsov-Vanstone 消息恢复签名
F	显式给定有限域
G	基础组(有限域元素/椭圆曲线点)产生器
Hash，$Hash_1$，$Hash_2$	杂凑函数
h	(被截取的)杂凑权标(八位位组串)
h'	可恢复的(被截取的)杂凑权标(八位位组串)
K	随机数(整数)
KDF	密钥导出函数(与 MGF 同义)
L_{clr}	不可恢复部分的八位位组数(整数)
L_{dat}	数据输入的八位位组数(整数)
L_F	显式有限域 F 的八位位组数(非负整数)
L_{rec}	可恢复部分的八位位组(最大)数(整数)
L_{red}	(添加的)冗余八位位组数(整数)
$L(x)$	整数 x 或八位位组串 x 的八位位组数(非负整数)
L_{Hash}	杂凑函数 Hash 输出的八位位组数(非负整数)
M	消息(八位位组串)
M_{clr}	消息的不可恢复部分(八位位组串)
M_{rec}	消息的可恢复部分(八位位组串)
M'	已恢复消息(八位位组串)
M'_{clr}	接收到的消息的不可恢复部分(八位位组串)
M'_{rec}	消息的已恢复部分(八位位组串)
MGF	掩码生成函数
NR	Nyberg-Rueppel 消息恢复签名
n	群 G 的阶(素数)
O	椭圆曲线的无穷远点
P	取决于密钥生成机制的元素，采用密钥生成机制 I ，则 $P=G$，采用密钥生成机制 II，$P=Y_A$[见 7.3]
p	素数
Π	预签名(八位位组串)
Π'	已恢复的预签名(八位位组串)
Q	取决于密钥生成机制的元素，采用密钥生成机制 I ，则 $Q=Y_A$，采用密钥生成机制 II，$Q=G$[见 7.3]
q	素数方幂
r	签名的第一部分(八位位组串)

国家图书馆专用

r'	已恢复签名的第一部分(八位位组串)
s	签名的第二部分(整数)
s'	已恢复签名的第二部分(整数)
x_A	实体 A 的私有签名密钥
Y_A	实体 A 的公开验证密钥
$\{0,1\}^*$	有限比特串集合
$\{0,1\}^{8*}$	有限八位位组串集合
$\{0,1\}^l$	l 长度的比特串集合,l 为非负整数
$\{0,1\}^{8l}$	l 长度的八位位组串集合,l 为非负整数
$[a,b]$	整数 x 的集合,$a\leqslant x\leqslant b$,其中 a 和 b 均为整数
$\|x\|$	比特串 x 的长度
$\|X\|$	集合 X 的基数
$[x]^l$	八位位组串 x 的最左侧 l-比特,当 $8l>L(x)$ 时,右侧填充 0
$[x]_l$	八位位组串 x 的最右侧 l-比特,当 $8l>L(x)$ 时,左侧填充 0
$x \bmod n$	模 n 的算术计算,$r\in[0,n-1]$,$(x-r)$ 除以 n,x 是整数
$x \oplus y$	x 和 y 的按位异或运算
$x\|\|y$	比特串 x 和比特串 y 的级联
$X\times Y$	X 和 Y 的笛卡尔乘积

4.2 转换函数和掩码生成函数

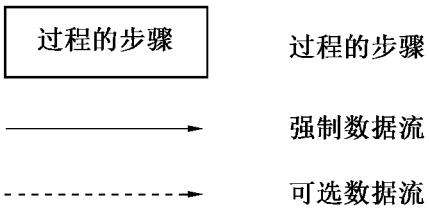
下列转换函数和掩码生成函数适用于本文件。

BS2IP	将比特串转换为整数[见附录 B 中 B.2]
BS2OSP	将比特串转换为八位位组串[见 B.1]
EC2OSP	将椭圆曲线转换为八位位组串[见 B.6]
FE2IP	将有限域元素转换为整数[见 B.4]
FE2OSP	将有限域元素转换为八位位组串[见 B.5]
I2BSP	将整数转换为比特串[见 B.2]
I2OSP	将整数转换为八位位组串[见 B.3]
MGF1	掩码生成函数 1 [见附录 C 中 C.2]
MGF2	掩码生成函数 2 [见 C.3]
OS2BSP	将八位位组串转换为比特串[见 B.1]
OS2ECP	将八位位组串转换为椭圆曲线[见 B.6]
OS2FEP	将八位位组串转换为有限域元素[见 B.5]
OS2IP	将八位位组串转换为整数[见 B.3]

4.3 附图说明

如下说明用于第 7 章中描述带消息恢复的数字签名的生成和验证过程的附图。

国家图书馆 专用



5 签名机制和杂凑函数之间的绑定

本部分规定的签名机制要求进行杂凑函数的选择。ISO/IEC 10118 规定了杂凑函数。在使用时需要将签名机制和杂凑函数进行绑定。如果没有这种绑定,攻击者可以使用较弱的杂凑函数(不是实际使用的),因此攻击者可以仿造签名。本部分凡涉及采用密码技术解决机密性、完整性、真实性、不可否认性需求的应遵循密码相关国家标准和行业标准。

数字签名机制的使用者应权衡实现绑定的不同方式的成本与效益,并做出评估。评估也应包括可能存在的仿造签名的成本评估。

注 1:本部分使用的杂凑函数的一个安全要求是“抗碰撞”。

注 2:有很多方式能够实现上述的绑定。以下选项按照危险程度的递增列出:

a) 特殊的签名机制需要特殊的杂凑函数。验证过程应该使用此杂凑函数。ISO/IEC 14888-3 给出了上述选项的实例:DSA 机制要求使用专用的杂凑函数 3(即 SHA-1,在 ISO/IEC 10118-3 定义)。

b) 允许一套杂凑函数,在证书域参数里明确指定使用的杂凑函数。在证书域里,验证过程也使用此证书里说明的杂凑函数。在证书域之外,如果权威机构(CA)不遵循用户的策略,威胁就会增加。例如,外部 CA 生成一个证书,允许使用其他的杂凑函数,签名伪造的问题可能出现。在这种情况下,被误导的验证方可能与此 CA 产生分歧。

c) 允许一套杂凑函数,使用其他方式指定杂凑函数,例如,在消息或者双向协商里指定。验证方使用此指定的杂凑函数。但是攻击者可能使用另一个杂凑函数仿造签名,所以存在威胁。

注 3:注 2c)中的“其他方式”可以是把杂凑函数标识符的形式包括在八位位组串 d 中。如果 d 中包括杂凑函数标识符,甚至当验证方接受使用足够弱的杂凑函数(其能够导致消息被发现)的时候,攻击者也不能欺骗地把相同八位位组 d_1 的签名重复使用在对不同八位位组 d_2 的签名上。但是在使用弱杂凑函数的情况下,攻击者依旧能够发现一个对“随机的”d_1 的新签名。

注 4:通过要求在 d_1 中出现指定的结构,就能够防止注 3 中提到的产生一个对“随机的”d_1 的签名攻击。例如,对 d_1 做个长度限制,使其远远小于签名机制的能力。对于许多数字签名机制,尽管在消息中没有包括杂凑函数标识符,如果掩码生成函数 MGF 是基于杂凑函数的,则 d_1 的长度限制也能够阻止攻击者重复使用已存在的签名。上述是假设弱杂凑函数是“一般目的”的杂凑函数,而不是专为仿造签名而设计的。

6 带消息恢复的数字签名框架

6.1 过程

6.2～6.4 包括了本部分规定的五种签名方案的总体模型的高级描述。总体模型的详细描述在第 7 章中。

本部分规定的数字签名方案有如下过程:

a) 参数生成过程;

b) 签名生成过程;

c) 签名验证过程。

6.2 参数生成过程

6.2.1 域参数

参数分为两部分:域参数和用户密钥。域参数包括定义有限群的参数,如有限域的一个乘法群或者有限域上的椭圆曲线的一个加法群,以及其他常见且已知的或者可以被域中所有实体访问的公共信息。与密码方案中规定的域参数一样,也应规定下面的参数:

a) 使用的数字签名方案标识符;

b) 冗余类型;

c) (可选的)杂凑函数;

d) 用户密钥生成过程。

有限域上的椭圆曲线的一个加法群的实现以及数学背景见 ISO/IEC 15946-1。构建有限域上的椭圆曲线的方法见 ISO/IEC 15946-5。

6.2.2 用户密钥

每个实体都有自己的公开和私有密钥。实体 A 的用户密钥包括:

a) 私有签名密钥 x_A;

b) 公开验证密钥 Y_A;

c) (可选的)其他信息,由实体 A 规定,用于签名生成/验证过程。

注 1:用户密钥只在规定的域参数集的上下文中有效。

注 2:签名验证方可能要求确保域参数和公开验证密钥是有效的,否则尽管进行了签名验证,也不能保证达到了预期的安全。签名方也可能要求确保域参数和公开验证密钥是有效的,否则攻击者可能生成被验证的签名。

6.3 签名生成过程

签名生成过程需要以下的数据项:

a) 域参数;

b) 签名方 A 的私有签名密钥 x_A;

c) 消息 M。

本部分中规定的所有方案,签名生成过程包括如下处理:

a) 拆分消息;

b) (可选的)计算冗余,或者计算消息摘要;

c) 有限群的计算,或者是有限域的乘法群或者是有限域上的椭圆曲线的加法群;

d) 基点 G 的群阶求模计算;

e) 组成已签消息。

签名生成过程的输出是(r,s)对,其组成 A 对消息 M 的数字签名。

6.4 签名验证过程

签名验证过程需要以下的数据项:

a) 域参数;

b) 签名方 A 的公开验证密钥 Y_A;

c) 不可恢复的消息 M'_{clr}(如果存在)。

国家图书馆 专用

d) 接收到的消息 M 的签名，其使用八位位组串 r' 和整数 s' 表示。

本部分中规定的所有方案，签名验证过程包括如下处理：

a) 签名大小验证；

b) 有限群的计算，或者是有限域的乘法群或者是有限域上的椭圆曲线的加法群；

c) 基点 G 的群阶求模计算；

d) 恢复数据输入或者消息；

e) 签名检查。

如果所有处理成功，验证方接受签名，否则拒绝。

7 带消息恢复的数字签名总体模型

7.1 要求

7.1.1 域参数

本部分规定的数字签名机制的用户应选择以下的数字签名方案的域参数：

a) 显式给定有限域 F，或者显式给定有限域 F 上的椭圆曲线 E；

b) F 或者 E 的基点 G 的阶 n。

为了实现带消息恢复的数字签名机制，在用户间就上述选择达成一致是必要的。

注 1：n 的大小影响机制的安全性，所以它的选择需要满足安全目的。

注 2：方案使用的两个可能的群通常用作乘法表示(有限域的乘法群)和加法表示(椭圆曲线的点构成的群)。在第 7 章，为了方便表达，使用了乘法表示。

注 3：显式给定有限域的定义，参见 A.3。

注 4：显式给定有限域上的椭圆曲线的定义，参见 A.4。

注 5：与椭圆曲线有关的有效实施和密码技术，参见 ISO/IEC 15946-1:2002。

国家图书馆样书

7.1.2 冗余类型

用户选择的冗余类型是：

a) 自然冗余；

b) 填充冗余；或者

c) 两者兼之。

为了实现带消息恢复的数字签名机制，在用户间就上述冗余达成一致是必要的。

如果用户选择填充冗余，填充冗余的八位位组数 L_{red} 应是固定的。带有填充冗余的消息可以被消息或者可恢复消息的杂凑权标所构建。

如果用户选择自然冗余 L_{red} 应设置为 0。带有自然冗余的消息意味着消息包括诸如 ASCII 码的冗余或者消息的冗余在一些应用里可以隐性验证。

自然或者填充冗余可以是达成一致的任何事物，能够被通信双方校验。包括自然和填充冗余的整个冗余应比应用里规定的某个最小值大。总的来说，单独的自然冗余只能用来完整的消息恢复。

7.2 函数和过程总结

本部分规定的签名方案能够恢复消息。更准确地说，做为签名验证过程一部分，签名生成函数输入的一部分数据可以从签名中恢复出来。

签名方案包括以下函数和过程：

a） 用户密钥生成过程；

b） 签名生成过程；

c） 签名验证过程。

7.3 用户密钥生成过程

使用下述两种方式计算公开验证密钥和私有签名密钥(签名实体应保证私有签名密钥是秘密的)：

a） 密钥生成过程 I

如果域参数集合是有效的，可以使用下面的步骤生成私有签名密钥和与之对应的公开验证密钥：

1） 在集合[1，$n-1$]中选择随机或伪随机整数 x_A。x_A 应防止未经授权的泄露，而且应是不可预测的。

2） 计算 $Y_A=G^{x_A}$。

3） 密钥对即为(Y_A，x_A)，其中 Y_A 用于公开验证密钥，x_A 用于私有签名密钥。

算法统一表示为 $P=G$ 和 $Q=Y_A$。

b） 密钥生成过程 II

如果域参数集合是有效的，可以使用下面的步骤生成私有签名密钥和与之对应的公开验证密钥：

1） 在集合[1，$n-1$]中选择随机或伪随机整数 e。计算 x_A，满足 x_A 在[1，$n-1$]区间，$x_Ae=1 \bmod n$。x_A 和 e 应防止未经授权的泄露，而且应是不可预测的。

2） 计算 $Y_A=G^e$。

3） 密钥对即为(Y_A，x_A)，其中 Y_A 用于公开验证密钥，x_A 用于私有签名密钥。

算法统一表示为 $P=Y_A$ 和 $Q=G$。

在使用公开验证密钥之前，验证方需要确保公开验证密钥的有效性和所属。有效性可以通过多种方式获得，见 6.2.2。

注 1：有些方案私有签名密钥 x_A 的范围是[1，$n-2$]。

注 2：密钥生成 I 是比较流行，经常使用的方式。在有些环境下，求模运算代价比较高，密钥生成 II 可能更有优势。

7.4 签名生成过程

7.4.1 过程

图 1 描述了签名生成过程，包括以下处理：

a） 生成随机数和预签名；

b） 拆分消息；

c） 生成数据输入；

d） 计算签名；

e） 组成已签消息。

注：图 1 中显示的域参数只是做为示例使用，每个方案可能要求与方案相关的域参数。

国家图书馆公用

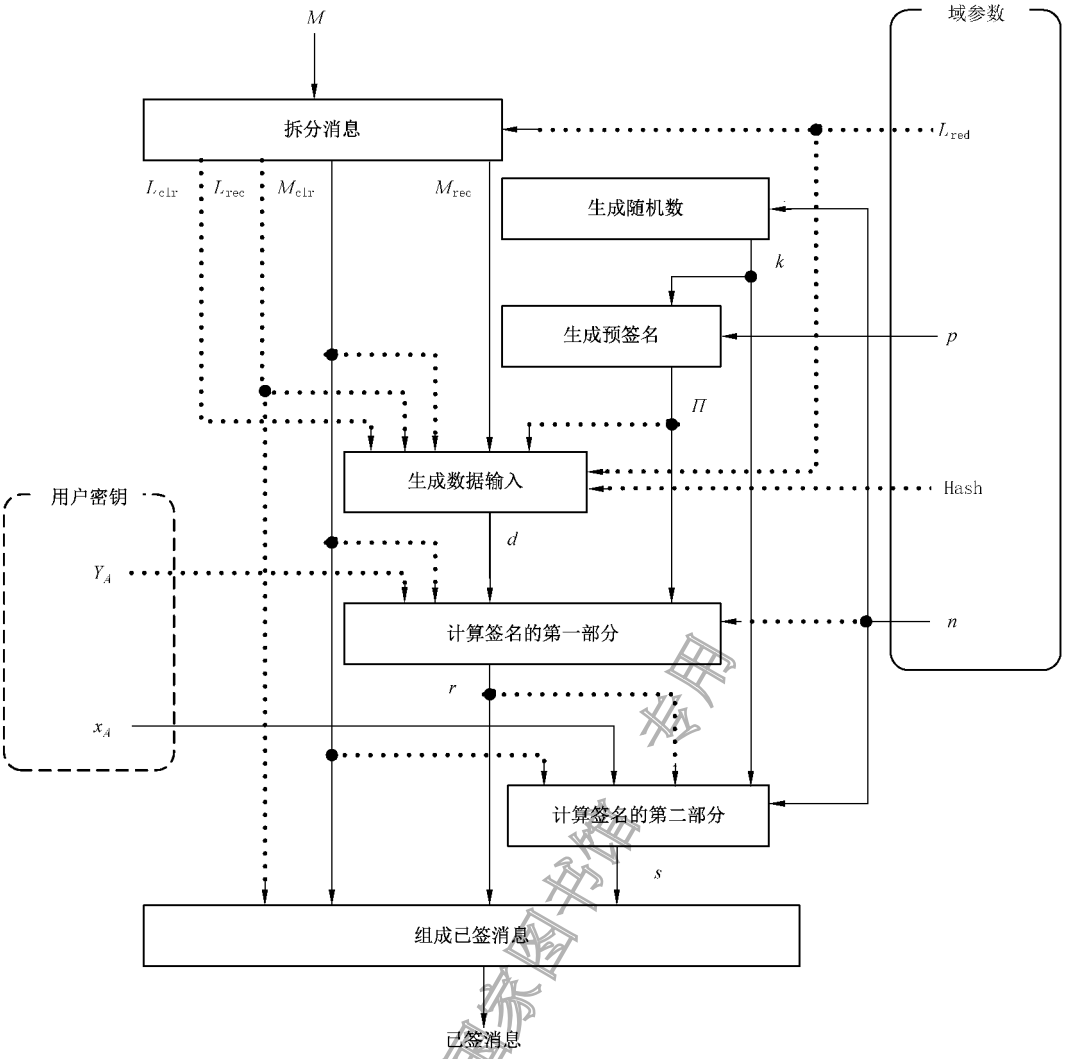


图 1　签名生成过程

7.4.2　生成随机数和预签名

在签名计算之前，签名实体应有一个可用的、新鲜的、秘密的随机数。这个随机数是一个整数 k，($1 \leqslant k \leqslant n-1$)。签名方案的实现者应确保满足以下要求：

a)　同一个随机数用于两个不同消息的签名的可能性是非常小的；

b)　已经使用过的随机数值不会被泄露，一旦使用过，这些随机数会被销毁。

首先，通过随机数发生器生成一个随机数，整数 k。然后对随机数进行运算生成预签名 Π(八位位组串)。在任何随机数签名方案里，签名生成过程中，预签名都是一个中间数据项。预签名是一个公开数据项，随机数的值只是在此次签名生成过程里有效。

注 1：使用后的随机数，如果后续泄露，就会给私钥带来威胁。使用过的随机数不会再被签名方或验证方使用，所以可安全地清除。如果同一个随机数用于两个不同的消息签名，或者签名使用的随机数被泄露，则从签名中恢复私钥是可能的。

注 2：通过随机数发生器生成随机数以及预签名的计算可以离线进行。在上述情况下，随机数可安全存储以便签名生成过程将来使用。

国家图书馆专用

7.4.3 拆分消息

消息 M 被拆分成消息的可恢复部分 M_{rec} 和不可恢复部分 M_{clr}，它们的八位位组数分别使用 L_{rec} 和 L_{clr} 表示。

7.4.4 生成数据输入

数据输入函数的输入是带有填充冗余的可恢复消息 M_{rec}，或者是带有自然冗余的可恢复消息 M_{rec}。输入可选地包括不可恢复部分 M_{clr}，长度 L_{rec} 和 L_{clr}。在使用填充冗余的情况下，数据输入包括杂凑权标。杂凑权标由杂凑码自己或者与链接在杂凑码后面的杂凑函数标识符一起组成，杂凑码通过杂凑(可恢复部分)消息计算得出。杂凑权标是否包括杂凑函数标识符是由域参数控制的。数据输入函数的输出是八位位组串 d。

注 1：数据输入的选择可以由每个应用或者签名方案决定。

注 2：使用填充冗余生成数据输入的方法参见附录 D。

7.4.5 计算签名

这个方案生成的签名有两部分，分别是 r 和 s。第一部分八位位组串 r，是对预签名 Π 和数据输入 d(及其他的可选参数)计算后的结果，d 是依赖消息的一个八位位组串。第二部分整数 $s(0<s<n)$，是对第一部分 r，随机数 k 和私有签名密钥 x_A(及其他的可选参数)计算后的结果。

7.4.6 组成已签消息

知晓消息的可恢复部分的长度对于成功地开启和验证已签消息是非常必要的。通过包括在已签消息和/或从数据输入 d 中获取的手段，域参数应给出这个信息。

已签消息包括以下部分：

a) 消息的不可恢复部分 M_{clr}；

b) 签名的第一部分 r；

c) 签名的第二部分 s；

d) (可选的)消息的可恢复部分的长度 L_{rec}。

7.5 签名验证过程

7.5.1 过程

图 2 描述了签名验证过程，包括以下处理：

a) 开启已签消息；

b) 签名大小的验证；

c) 恢复预签名或数据输入；

d) 恢复数据输入或消息；

e) 重新计算杂凑权标(可选)；

f) 校验签名。

国家图书馆 样书

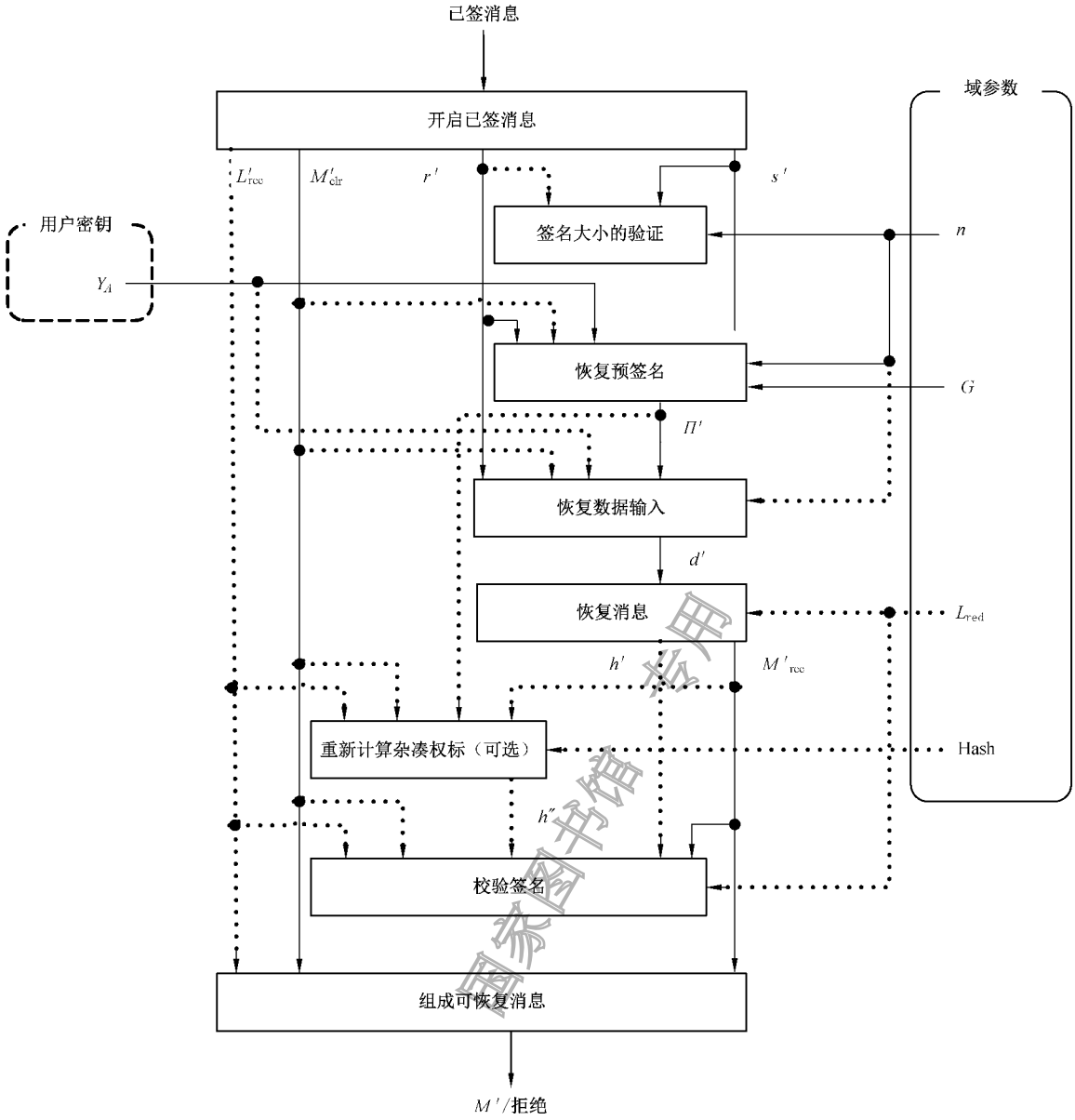


图 2　签名验证过程

签名的校验包括如下：

a）　比较已恢复的杂凑权标和重新计算(被截取)的杂凑权标；或者

b）　校验冗余。

7.5.2　开启已签消息

在进行此步骤前，验证方应确保下列信息可用：

a）　包括在已签消息里的不同的签名/消息部分的长度；

b）　参数 L_{red} 的长度。

验证方提取下列已签消息的不同部分：

a）　消息的不可恢复部分；

b）　签名的第一部分 r'；

c）　签名的第一部分 s'；

d) （可选的）可恢复消息部分的长度 L'_{rec}。

7.5.3 签名大小的验证

验证方需要校验签名部分的大小。

7.5.4 预签名的恢复

在进行此步骤前，验证方应确保下列信息可用：

a) 指明正在使用的此签名方案的公开信息；

b) 签名实体的公开验证密钥 Y_A；

这一步的计算是针对正在使用的签名方案而言的。预签名是由公开验证密钥 Y_A 决定的。给出签名 (r', s')，预签名 Π' 是可恢复的。

7.5.5 输入数据或者消息的恢复

给出签名的第一部分 r' 和已恢复的预签名 Π'，数据输入 d' 是可恢复的。已恢复的数据输入 d' 是一个八位位组串。

7.5.6 杂凑权标的重新计算（可选）

首先，7.4 签名实体使用的杂凑函数可能由域参数和/或者通过从已恢复的杂凑权标中获取的杂凑函数标识符来标识。通过杂凑消息重新计算杂凑码。

把重新计算的杂凑码可选地链接杂凑函数标识符，就能够获得重新计算的杂凑权标。

7.5.7 校验签名

签名的校验包括如下：

a) 比较重新计算（被截取）的杂凑权标 h'' 和已恢复（被截取）的杂凑权标 h'；或者

b) 校验已恢复消息的填充，和/或者自然冗余。

国家图书馆 使用

8 NR（Nyberg-Rueppel 消息恢复签名）

8.1 域参数和用户密钥

域参数使用显式给定有限域 F 的乘法群。

数据输入 d 的八位位组数 L_{dat} 是固定值，其值小于或等于 $L(n)-1$。

按照下列方法生成 NR 的密钥：

a) A 的私有签名密钥 x_A，其是在区间 $[1, n-1]$ 的随机整数。

b) 按照 7.3 计算 A 的公开验证密钥 Y_A。

注：显式给定有限域的定义参见 A.3。

8.2 签名生成过程

8.2.1 输入和输出

签名生成过程的输入包括：

a) 域参数；

b) 私有签名密钥 x_A；和

c) 签名消息 M。

签名生成过程的输出是包括 A 对消息 M 数字签名的一对值$(r,s)\in\{0,1\}^{8L(n)}\times[1,n-1]$。

8.2.2 生成随机数和预签名(有限域计算)

预签名 $\Pi\in\{0,1\}^{8*}$ 按照如下或者等同如下的步骤进行计算:

a) 在区间$[1,n-1]$里选择随机整数 k;

b) 计算有限域元素 $R=P^k$;

c) 将 R 转换为八位位组串 $\Pi=\mathrm{FE2OSP}_F(R)$。

8.2.3 生成数据输入

从消息 M 中生成数据输入 $d\in\{0,1\}^{8L_{\mathrm{dat}}}$ 见 7.4.2 和 7.4.3。

8.2.4 计算签名(模 n 的算术运算)

签名$(r,s)\in\{0,1\}^{8L(n)}\times[1,n-1]$按照如下或者等同如下的步骤进行计算:

a) 将 d 转换为整数$\delta=\mathrm{OS2IP}(d)$,注意 $\delta\in[0,n-1]$;

b) 计算 $\pi=\mathrm{OS2IP}(\Pi)\bmod n$;

c) 计算 $\tilde{r}=(\delta+\pi)\bmod n$;

d) 计算 $s=(k-x_A\tilde{r})\bmod n$;

e) 转换 $r=\mathrm{I2OSP}[\tilde{r},L(n)]$;

f) 清除 k。

如果签名生成过程生成了 $\tilde{r}=0$ 或者 $s=0$,则应采用新的随机值 k 重新进行签名生成过程。

8.2.5 组成已签消息

一对值$(r,s)\in\{0,1\}^{8L(n)}\times[1,n-1]$ 包括 A 对消息 M 的签名。

8.3 签名验证过程

8.3.1 输入和输出

签名验证过程包括三个步骤:消息摘要的计算、有限域计算和签名验证。

签名验证过程的输入包括:

a) 域参数;

b) A 的公开验证密钥 Y_A;

c) 已接收到的对消息 M 的签名,表示为八位位组串 r'和整数 s';和

d) 不可恢复消息 M'_{clr}(如果存在);

签名验证过程的输入或者是可恢复消息 d'或者是“拒绝”。

8.3.2 签名大小的验证

验证是否符合 $r'\in\{0,1\}^{8L(n)}$,$0<\mathrm{OS2IP}(r')<n$ 和 $0<s'<n$;如果不符合,则拒绝签名。

8.3.3 恢复预签名(有限域计算)

从接收到的签名(r',s')里恢复预签名,按照如下或者等同如下的步骤进行:

a) 转换$\tilde{r}'=\mathrm{OS2IP}(r')$;

b) 计算 $R'=P^{s'}Q^{\tilde{r}'}$;

国家图书馆专用

c) 将 R' 转换为八位位组串 $\Pi'=\mathrm{FE2OSP}_F(R')$。

8.3.4 恢复数据输入或者消息

从接收到的签名的第一部分 r' 和已恢复的预签名 Π' 里恢复数据输入，按照如下或者等同如下的步骤进行：

a) 计算 $\pi'=\mathrm{OS2IP}(\Pi')\bmod n$；

b) 计算 $\delta'=(\tilde{r}'-\pi')\bmod n$；

c) 将 δ' 转换为八位位组串 $d'=\mathrm{I2OSP}(\delta', L_{\mathrm{dat}})$。

8.3.5 校验签名

校验冗余。如果正确，输出 d'，否则拒绝。

8.3.6 算例

NR 算例参见附录 F 中 F.1。

9 ECNR(椭圆曲线 Nyberg-Rueppel 消息恢复签名)

9.1 域参数和用户密钥

域参数使用显式给定有限域上的椭圆曲线 E 上阶为 n 的一个加法群。

数据输入 d 的八位位组数 L_{dat} 是固定值，其值小于或等于 $L(n)-1$。

按照下列方法生成 ECNR 的密钥：

a) A 的私有签名密钥 x_A，其是在区间$[1, n-1]$的随机整数；

b) 按照 7.3 计算 A 的公开验证密钥 Y_A。

注 1：显式给定有限域的定义参见 A.3。

注 2：显式给定有限域上的椭圆曲线的定义，参见 A.4。

9.2 签名生成过程

9.2.1 输入和输出

签名生成过程的输入包括：

a) 域参数；

b) 私有签名密钥 x_A；和

c) 签名消息 M。

签名生成过程的输出是包括 A 对消息 M 数字签名的一对值$(r, s)\in\{0, 1\}^{8L(n)}\times[1, n-1]$。

9.2.2 生成随机数和预签名(椭圆曲线计算)

预签名 $\Pi\in\{0,1\}^{8(L_F+1)}$ 按照如下或者等同如下的步骤进行计算：

a) 在区间$[1, n-1]$里选择随机整数 k；

b) 计算有椭圆曲线点 $R=kP$；

c) 将 R 转换为八位位组串 $\Pi=\mathrm{EC2OSP}_E(R, \mathrm{compressed})$。

注：采用未压缩格式的转换函数 EC2OSP 的定义参见 B.6。

9.2.3 生成数据输入

从消息 M 中生成数据输入 $d\in\{0,1\}^{8L_{\mathrm{dat}}}$，见 7.4.2 和 7.4.3。

国家图书馆专用

9.2.4 计算签名(模 n 的算术运算)

签名$(r,s)\in\{0,1\}^{8L(n)}\times[1,n-1]$按照如下或者等同如下的步骤进行计算:

a) 将 d 转换为整数 $\delta=\mathrm{OS2IP}(d)$,注意 $\delta\in[0,n-1]$;

b) 计算 $\pi=\mathrm{OS2IP}(\Pi) \bmod n$;

c) 计算 $\tilde{r}=(\delta+\pi) \bmod n$;

d) 计算 $s=(k-x_A\tilde{r}) \bmod n$;

e) 转换 $r=\mathrm{I2OSP}[\tilde{r},L(n)]$;

f) 清除 k。

如果签名生成过程生成了 $\tilde{r}=0$ 或者 $s=0$,则应采用新的随机值 k 重新进行签名生成过程。

9.2.5 组成已签消息

一对值$(r,s)\in\{0,1\}^{8L(n)}\times[1,n-1]$ 包括 A 对消息 M 的签名。

9.3 签名验证过程

9.3.1 输入和输出

签名验证过程包括三个步骤:消息摘要的计算、椭圆曲线的计算和签名验证。

签名验证过程的输入包括:

a) 域参数;

b) A 的公开验证密钥 Y_A;

c) 已接收到的对消息 M 的签名,表示为八位位组串 r'和整数 s';和

d) 不可恢复消息 M'_{clr}(如果存在)。

签名验证过程的输入或者是可恢复消息 d'或者是“拒绝”。

9.3.2 签名大小的验证

验证是否符合 $\mathrm{OS2IP}(r')\neq 0 \bmod n$,$r'\in\{0,1\}^{8L(n)}$和 $0<s'<n$;如果不符合,则拒绝签名。

9.3.3 恢复预签名(椭圆曲线计算)

从接收到的签名(r',s')里恢复预签名,按照如下或者等同如下的步骤进行:

a) 转换$\tilde{r}'=\mathrm{OS2IP}(r')$;

b) 计算 $R'=s'P+\tilde{r}'Q$;

c) 将 R'转换为八位位组串 $\Pi'=\mathrm{EC2OSP}_E(R',\ \mathrm{compressed})$。

9.3.4 恢复数据输入或者消息

从接收到的签名的第一部分 r'和已恢复的预签名 Π'里恢复数据输入,按照如下或者等同如下的步骤进行:

a) 计算 $\pi'=\mathrm{OS2IP}(\Pi') \bmod n$;

b) 计算 $\delta'=(r'-\pi') \bmod n$;

c) 将 δ'转换为八位位组串 $d'=\mathrm{I2OSP}(\delta',L_{\mathrm{dat}})$。

9.3.5 校验签名

校验冗余。如果正确,输出 d',否则拒绝。

国家图书馆 公用

9.3.6 算例

ECNR 算例参见 F.2。

10 ECMR(椭圆曲线 Miyaji 消息恢复签名)

10.1 域参数和用户密钥

域参数使用有限群为椭圆曲线上的一个加法群。按照下列方法生成 ECMR 的密钥:

a) A 的私有签名密钥 x_A,其是在区间[1,$n-1$]的随机整数;

b) 按照 7.3 计算 A 的公开验证密钥 Y_A。

A 也选择函数:

Mask:$\{0,1\}^{8*}\rightarrow\{0,1\}^{8L(n)}$,例如 Mask($x$)=[Hash($x$)]$_{8L(n)}$, MGF1[$x$,$L(n)$] 或者 MGF2[$x$,$L(n)$],其中:Hash:$\{0,1\}^{8*}\rightarrow\{0,1\}^{8L_{\text{Hash}}}$。

注 1:显式给定有限域的定义参见 A.3。

注 2:显式给定有限域上的椭圆曲线的定义,参见 A.4。

10.2 签名生成过程

10.2.1 输入和输出

签名生成过程的输入包括:

a) 域参数;

b) 私有签名密钥 x_A;和

c) 带有填充/自然冗余的数据 d,即八位位组串$\{0,1\}^{8L(n)}$。

从消息中传输数据 d,见 7.4.2 和 7.4.3。签名生成过程的输出是包括 A 对消息 M 数字签名的一对值(r,s)$\in\{0,1\}^{8L(n)}\times[1,n-1]$。

10.2.2 生成随机数和预签名(椭圆曲线计算)

预签名 $\Pi\in\{0,1\}^{8(2L_F+1)}$,按照如下或者等同如下的步骤进行计算:

a) 在区间[1,$n-1$]里选择随机整数 k。

b) 计算有椭圆曲线点 $R=kP$。

c) 计算 Π=Mask[EC2OSP$_E$(R, uncompressed)]。

注:采用未压缩格式的转换函数 EC2OSP 的定义参见 B.6。

10.2.3 计算签名(模 n 的算术运算)

签名(r,s)$\in\{0,1\}^{8L(n)}\times[1,n-1]$按照如下或者等同如下的步骤进行计算:

a) 转换 $r=d\oplus\Pi$;

b) 计算 s=[OS2IP(r)k−OS2IP(r)−1] / (x_A+ 1) mod n;

c) 清除 k。

如果签名生成过程生成了 $s=0$ 或者 OS2IP(r) mod $n=0$,则应采用新的随机值 k 重新进行签名生成过程。

10.2.4 组成已签消息

一对值(r,s)$\in\{0,1\}^{8L(n)}\times[1,n-1]$ 包括 A 对数据 d 的签名。

国家图书馆 专用

10.3 签名验证过程

10.3.1 输入和输出

签名验证过程包括三个步骤：消息摘要的计算、椭圆曲线的计算和签名验证。

签名验证过程的输入包括：

a) 域参数；

b) A 的公开验证密钥 Y_A；

c) 已接收到的对数据 d 的签名，表示为八位位组串 r' 和整数 s'；和

d) 函数 Mask；

e) 不可恢复消息 M'_{clr}（如果存在）。

签名验证过程的输入或者是可恢复消息 d' 或者是“拒绝”。

10.3.2 签名大小的验证

验证是否符合 $\mathrm{OS2IP}(r') \neq 0 \bmod n$，$r' \in \{0,1\}^{8L(n)}$ 和 $0 < s' < n$；如果不符合，则拒绝签名。

10.3.3 恢复预签名（椭圆曲线计算）

从接收到的签名 (r', s') 里恢复预签名，按照如下或者等同如下的步骤进行：

a) 转换 $R' = [(1 + \mathrm{OS2IP}(r') + s'] / \mathrm{OS2IP}(r')]P + [s' / \mathrm{OS2IP}(r')]Q$；

b) 计算 $\Pi' = \mathrm{Mask}[\mathrm{EC2OSP}_E(R', \text{uncompressed})]$。

10.3.4 恢复数据输入或者消息

计算 $d' = r' \oplus \Pi'$。

10.3.5 校验签名

校验冗余。如果正确，输出 d'，否则拒绝。

10.3.6 算例

ECMR 算例参见 F.3。

11 ECPV（椭圆曲线 Pintsov-Vanstone 消息恢复签名）

11.1 域参数和用户参数

域参数使用显式给定有限域 F 上的椭圆曲线 E 上的阶为 n 的加法群。

填充冗余的长度 L_{red} 和安全参数有关，除了其他冗余规则之外，填充冗余的长度设置为 1～255（含 255）；见 11.2.3。

A 使用一个杂凑函数，一个密钥导出函数和一个对称密码：

a) $\mathrm{Hash}: \{0,1\}^{8*} \rightarrow \{0,1\}^{8[L(n)-1]}$；

b) $\mathrm{KDF}: \{0,1\}^{8*} \rightarrow \{0,1\}^{8L_{\mathrm{key}}}$；和

c) $\mathrm{Sym}: \{0,1\}^{8*} \times \{0,1\}^{8L_{\mathrm{key}}} \rightarrow \{0,1\}^{8*}$。

L_{key} 表示 Sym 使用的密钥的八位位组数。函数 KDF 定义为 $\mathrm{KDF}(x) = \mathrm{MGF2}(x, L_{\mathrm{key}})$，其中 $x \in \{0,1\}^{8*}$。

按照下列方法生成 ECPV 的密钥：

a) A 的私有签名密钥 x_A，其是在区间 $[1, n-1]$ 的随机整数；

国家图书馆 藏用

b） 按照 7.3 计算 A 的公开验证密钥 Y_A。

注 1：显式给定有限域的定义参见 A.3。

注 2：显式给定有限域上的椭圆曲线的定义，参见 A.4。

注 3：L_{key} 和安全参数有关，为了达到安全目的宜选择合适的长度。对称加密可以使用异或（⊕）运算；在这种情形下，L_{key} 可等于数据输入的长度，可恢复消息的最大长度是由域参数决定的；参见 11.2.3。

11.2 签名生成过程

11.2.1 输入和输出

签名生成过程的输入包括：

a） 域参数；

b） 私有签名密钥 x_A；和

c） 签名消息 M。

签名生成过程的输出是包括 A 对消息 M 数字签名的一对值 $(r,s)\in\{0,1\}^{8*}\times[1,n-1]$。

11.2.2 生成随机数和预签名（椭圆曲线计算）

预签名（对称密钥）$\Pi\in\{0,1\}^{8L_{key}}$ 按照如下或者等同如下的步骤进行计算：

a） 在区间 $[1,n-1]$ 里选择随机整数 k；

b） 计算有椭圆曲线点 $R=kP=(x,y)$；

c） 将 x 转换为八位位组串 $S=\mathrm{FE2OSP}_F(x)$；

d） 计算对称密钥 $\Pi=\mathrm{KDF}(S)$。

11.2.3 拆分消息和生成数据输入

A 将消息拆分成可恢复部分 M_{rec}（最左侧八位位组）及 M_{clr}（余下的八位位组）。注意对称密码函数 Sym 的选择可能导致输入的长度限制。M_{rec} 和 M_{clr} 的编码格式需要两个实体协商一致。随机数 nonce 可以用来替代 M_{clr}。

通过 M_{rec} 和填充冗余组成八位位组串 d，按照如下或者等同如下的步骤进行：

a） 将 L_{red} 转换为一个八位位组 $C_{red}=\mathrm{Oct}(L_{red})$；

b） 八位位组 C_{red} 重复次 L_{red} 次形成 $\widetilde{C}_{red}$（这样 $\widetilde{C}_{red}$ 的长度就是 L_{red}）；

c） 计算 $d=\widetilde{C}_{red}||M_{rec}$。

注 1：ECPV 明确规定了生成数据输入的方法。

注 2：原则上这个方法可以使用“single-pass”方式进行处理，因为不可恢复消息部分 M_{clr} 根本不需要处理。

注 3：在 ECPV 里，可恢复消息部分的编码方式和对称密码的填充规则可能数据输入的自然填充，这样就增加了整个冗余的数量。选择 L_{red} 以便整个冗余的八位位组数要多于 $L(n)/2$ 或者 $L_{Hash}/2$。

注 4：ECPV 能够处理任意八位位组数的可恢复消息部分。

注 5：为了满足安全目的，ECPV 建议使用下列规定：

a） 冗余规则可以规定可恢复消息部分是固定长度的，或者开始于关于它的长度的固定长度的表达式；

b） 冗余规则可以规定可恢复消息部分使用 ASN.1 类型的 DER 编码，参见附录 E；

c） 域参数可以规定不可恢复消息部分是固定长度的（可能是空），或者结束于关于它的长度的固定长度的表达式。

11.2.4 计算签名（模 n 的算术运算）

签名 $(r,s)\in\{0,1\}^{8*}\times[1,n-1]$ 按照如下或者等同如下的步骤进行计算：

国家图书馆 专用

a) 计算 $r=\mathrm{Sym}(d,\Pi)$；

b) 计算 $u=\mathrm{Hash}(r||M_{clr})$；

c) 转换 $t=\mathrm{OS2IP}(u)$；其中 $t\in[0,n-1]$；

d) 如果 $t=0$，则应采用新的随机值 k 重新进行签名生成过程；

e) 计算 $s=(k-x_A t) \bmod n$；

f) 如果 $s=0$，则应采用新的随机值 k 重新进行签名生成过程；

g) 清除 k。

输出签名 (r,s) 和局部消息部分 M_{clr}（可以为空）。

11.2.5 组成已签消息

一对值 $(r,s)\in\{0,1\}^{8*}\times[1,n-1]$ 包括 A 对消息 M 的签名。

11.3 签名验证过程

11.3.1 输入和输出

签名验证过程包括三个步骤：消息摘要的计算、椭圆曲线的计算和签名验证。

签名验证过程的输入包括：

a) 域参数；

b) A 的公开验证密钥 Y_A；

c) 已接收到的对数据 M 的签名，表示为八位位组串 r' 和整数 s'；和

d) 不可恢复消息 M'_{clr}（如果存在）。

为了校验 A 的签名，验证方 B 执行 11.3.2～11.3.5。

11.3.2 签名大小的验证

验证是否符合 $0<s'<n$；如果不符合，则拒绝签名。

11.3.3 恢复预签名（椭圆曲线计算）

从签名恢复预签名（对称密钥），按照如下或者等同如下的步骤进行：

a) 计算 $u'=\mathrm{Hash}(r'||M'_{clr})$；

b) 转换 $t'=\mathrm{OS2IP}(u')$；其中 $t'\in[0,n-1]$；

c) 如果 $t'=0$，则拒绝签名；

d) 计算 $R'=s'P+t'Q=(x',y')$ 执行下面的操作：

1) 如果 R' 是无穷远点，则拒绝签名；

2) 否则计算对称密钥 $\Pi'=\mathrm{KDF}[\mathrm{FE2OSP}_F(x')]$。

11.3.4 恢复数据输入

通过计算 $d'=\mathrm{Sym}^{-1}(r',\Pi')$ 恢复数据输入，其中 Sym^{-1} 表示对称密码 Sym 的解密函数。

11.3.5 校验签名

校验 d' 的填充冗余和已恢复的 M'_{rec}，按照如下或者等同如下的步骤进行：

a) 如果 $L(d')<L_{red}$，则拒绝签名。

b) 将 L_{red} 转换为一个八位位组 $C_{red}=\mathrm{Oct}(L_{red})$。

c) 八位位组 C_{red} 重复次 L_{red} 次形成 $\widetilde{C}_{red}$（这样 $\widetilde{C}_{red}$ 的长度就是 L_{red}）。

国家图书馆专用

d) 通过 $\widetilde{C}_{\mathrm{red}}=[d']^{8L_{\mathrm{red}}}$ 检查填充冗余；如果不包括，则拒绝签名。

e) 计算 $M'_{\mathrm{rec}}=[d']_{8[L(d')-L_{\mathrm{red}}]}$。

f) 检查 M'_{rec} 的自然冗余是否符合它的编码格式；如果不符合，则拒绝签名。

g) 检查 M'_{clr}（如果存在）的格式；如果不满足，则拒绝签名。

h) 通过下面的操作恢复 M'：

1) 如果 M'_{clr} 为空或者是随机数 nonce 时，设置 $M'=M'_{\mathrm{rec}}$；

2) 否则，从 M'_{rec} 和 M'_{clr} 恢复 M'。

i) 输出 M'。

11.3.6 算例

ECPV 算例参见 F.4。

12 ECKNR（椭圆曲线 KCDSA/Nyberg-Rueppel 消息恢复签名）

12.1 域参数和用户参数

域参数使用显式给定有限域 F 上的椭圆曲线 E 上的阶为 n 的加法群。

填充冗余的长度 L_{red} 和安全参数有关，除了其他冗余规则之外，填充冗余的长度设置为 1～255（含 255）；见 11.2.3。

A 使用一个掩码生成函数。

MGF：$\{0,1\}^{8*}\rightarrow\{0,1\}^{8L(n)}$。

函数 MGF 的定义为：$\mathrm{MGF}(x)=\mathrm{MGF2}[x,L(n)]$，其中：$x\in\{0,1\}^{8*}$ 及潜在的杂凑函数 Hash。

按照下列方法生成 ECKNR 的密钥：

a) A 的私有签名密钥 x_A，其是在区间 $[1,n-1]$ 的随机整数；

b) 按照 7.3 计算 A 的公开验证密钥 Y_A。

A 的证书导出数据定义为 $z_A=[\mathrm{Cert}_A]^{L_B:\mathrm{Hash}}$，其中：$\mathrm{Cert}_A$ 表示 A 的证书，即把 A 的公开验证密钥 Y_A 转换为比特串。当 $Y_A=(x_0,y_0)$ 时，$\mathrm{Cert}_A=\mathrm{FE2OSP}_F(x_0)\,||\,\mathrm{FE2OSP}_F(y_0)$。$L_B$，Hash 是杂凑函数输入大小的比特长度。例如，在 RIPEMD-160 里 L_B，Hash=512。

注 1：显式给定有限域的定义参见 A.3。

注 2：显式给定有限域上的椭圆曲线的定义，参见 A.4。

12.2 签名生成过程

12.2.1 输入和输出

签名生成过程的输入包括：

a) 域参数；

b) 私有签名密钥 x_A；

c) A 的证书数据 z_A；和

d) 签名消息 M，其被拆分成可恢复部分 M_{rec}（最左侧八位位组）及 M_{clr}（余下的八位位组）。

签名生成过程的输出是包括 A 对消息 M 数字签名的一对值 $(r,s)\in\{0,1\}^{8L(n)}\times[1,n-1]$。

12.2.2 生成随机数和预签名（椭圆曲线计算）

预签名 $\Pi\in\{0,1\}^{8L(n)}$ 按照如下或者等同如下的步骤进行计算：

a) 在区间 $[1,n-1]$ 里选择随机整数 k；

国家图书馆 查阅

b) 计算有椭圆曲线点 $R=kP$；

c) 将 R 转换为八位位组串，计算杂凑值 $\Pi=\mathrm{MGF}[\mathrm{EC2OSP}_E(R,\ \mathrm{compressed})]$。

注：采用压缩格式的转换函数 EC2OSP 的定义参见 B.6。

12.2.3 生成数据输入

从消息中生成带有填充/自然冗余的数据 d，即八位位组串 $\{0,\ 1\}^{8L(n)}$。见 7.4.4。

12.2.4 计算签名(模 n 的算术运算)

签名 $(r,s)\in\{0,\ 1\}^{8L(n)}\times[1,n-1]$ 按照如下或者等同如下的步骤进行计算：

a) 计算 A 的签名的第一部分 $r=d\oplus\Pi\oplus\mathrm{MGF}(z_A||M_{\mathrm{clr}})$；

b) 设置 $t=\mathrm{OS2IP}(r)\bmod n$；

c) 转换 $t=\mathrm{OS2IP}(u)$；其中 $t\in[0,n-1]$；

d) 计算 A 的签名的第二部分 $s=(k-x_At)\bmod n$；

e) 清除 k。

如果签名生成过程生成了 $\mathrm{OS2IP}(r)=0$ 或者 $s=0$，则应采用新的随机值 k 重新进行签名生成过程。

12.2.5 组成已签消息

一对值 $(r,s)\in\{0,\ 1\}^{8L(n)}\times[1,n-1]$ 包括 A 对消息 M 的签名。

12.3 签名验证过程

12.3.1 输入和输出

签名验证过程包括三个步骤：消息摘要的计算、椭圆曲线的计算和签名验证。

签名验证过程的输入包括：

a) 域参数；

b) A 的公开验证密钥 Y_A；

c) A 的证书数据 z_A；

d) 已接收到的对数据 M 的签名，表示为八位位组串 r' 和整数 s'；和

e) 不可恢复消息 M'_{clr}(如果存在)。

签名验证过程的输入或者是可恢复消息 d' 或者是“拒绝”。

12.3.2 签名大小的验证

验证是否符合 $r'\in\{0,\ 1\}^{8L(n)}$，$\mathrm{OS2IP}(r')\neq0$ 和 $0<s'<n$；如果不符合，则拒绝签名。

12.3.3 恢复预签名(椭圆曲线计算)

从已接收的签名 (r',s') 恢复预签名，按照如下或者等同如下的步骤进行：

a) 计算 $t'=\mathrm{OS2IP}(r')\bmod n$；

b) 计算椭圆曲线点 $R'=s'P+t'Q$；

c) 将 R' 转换为八位位组串，计算杂凑值 $\Pi'=\mathrm{MGF}[\mathrm{EC2OSP}_E(R',\mathrm{compressed})]$。

12.3.4 恢复数据输入或者消息

通过已接收的签名的第一部分 r' 和已恢复的预签名 Π' 恢复数据输入，按照如下或者等同如下的步

国家图书馆 专用

骤进行：

a） 计算恢复数据输入 $d'=r'\oplus \Pi'\oplus \mathrm{MGF}(z_A||M'_{\mathrm{clr}})$。

12.3.5 校验签名

校验冗余。如果正确，输出 d'，否则拒绝。

12.3.6 算例

ECKNR 算例参见 F.5。

国家图书馆 专用

附 录 A
（资料性附录）
数 学 转 换

A.1 比特串

一个比特的值或者是“0”或者是“1”。一个比特串是比特 x_0，…，x_{l-1} 的有限序列。比特串的长度是比特串里的比特个数。对于非负整数 n，$\{0,1\}^n$ 表示长度为 n 的比特串集合。$\{0,1\}^* = \bigcup_{n\geq 0}\{0,1\}n$ 表示包括空比特串（长度为 0）在内的比特串的集合。

A.2 八位位组串

一个八位位组是一个长度为 8 的比特串。一个八位位组串是八位位组的有限序列。八位位组串的长度是八位位组串里的八位位组个数。$\{0,1\}^{8*}$ 表示包括空八位位组串（长度为 0）在内的八位位组串的集合。一个八位位组经常使用十六进制格式表示，范围从 00～FF；见 B.3。

A.3 有限域

本条主要描述指定有限域的总体框架。此类有限域称之为显式给定有限域，是由显式数据决定的。对于有限域 F，基数 $q=p^e$，其中 p 为素数，$e\geqslant 1$，F 的显式数据包括 p 和 e 及一个乘法表。此乘法表是 $T=(T_{ij})_{1\leqslant i,j\leqslant e}$ 矩阵，T_{ij} 是[0，$p-1$]区间的 e 元组。

有限域 F 的元素集合是[0，$p-1$]区间所有 e 元组的集合。矩阵 T 的元素也是有限域 F 的元素。

有限域 F 的加法定义为元素依次相加：如果：

$$a=(a_1,\cdots,a_e)\in F \text{ 和 } b=(b_1,\cdots,b_e)\in F,$$

则 $a+b=c$ 计算如下：

$$c=(c_1,\cdots,c_e) \text{ and } c_i=(a_i+b_i)\bmod p\ (1\leqslant i\leqslant e)$$

有限域 F 的点乘定义为元素依次相乘：如果：

$$a=(a_1,\cdots,a_e)\in F \text{ 和 } d\in[0,p-1],$$

则 $d\cdot a=c$ 计算如下：

$$c=(c_1,\cdots,c_e) \text{ 和 } c_i=(d\cdot a_i)\bmod p\ (1\leqslant i\leqslant e)$$

有限域 F 使用乘法表 T 进行乘法计算，如下：

$$a=(a_1,\cdots,a_e)\in F \text{ 和 } b=(b_1,\cdots,b_e)\in F,$$

$$a\cdot b=\Sigma_{1\leqslant i\leqslant e}\Sigma_{1\leqslant j\leqslant e}(a_ib_j \bmod p)T_{ij},$$

其中：乘积 $(a_ib_j \bmod p)T_{ij}$ 使用上述定义的点乘进行计算，之后这些乘积使用上述定义的加法求和。假设乘法表定义了一个满足域常用公理的代数结构；另外存在加法和乘法恒等元，每个元素有加法逆运算和乘法逆运算。

有限域 F 的加法恒等元（表示为 0_F）是全零的 e 元组，有限域 F 的乘法恒等元（表示为 1_F）是一个非零的 e 元组，其精确格式取决于 T。

注 1：域 F 是基数为 p 的素数域 F' 的 e 维向量空间，其中点乘如上述定义。素数 p 称之为 F 的特征。对于 $1\leqslant i\leqslant e$，θ_i 表示 F' 的 e 元组，其第 i 个元素是 1，其他元素是 0。θ_1，…，θ_e 形成了 F' 上的向量空间域 F 的有序基。注意，如果 $1\leqslant i\leqslant e$，则。$\theta_i\cdot\theta_j=T_{ij}$。

注 2：对于 $e>1$，定义了两种有限域计算常用的基：

——如果 $\theta\in F$，$\theta_i=\theta^{p^{e-i}}$，其中 $1\leqslant i\leqslant e$，则 θ_1，…，θ_e 称之为 F' 上的域 F 的多项式基。注意在这种情况下，$1_F=\theta_e$；

——如果 $\theta\in F$，$\theta_1=\theta^{p^{i-1}}$，其中 $1\leqslant i\leqslant e$，则 θ_1，…，θ_e 称之为 F' 上的域 F 的正规基。注意在这种情况下，$1_F=c\Sigma_{1\leqslant i\leqslant e}\theta_i$，其中 $c\in[0,p-1]$；如果 $p=2$，则 c 只能是 1；此外，可以总是选择 $c=1$ 的正规基。

A.4 椭圆曲线

显式给定有限域 F 上的椭圆曲线 E 是点 $P=(x,y)$ 和无穷远点 o 的集合，其中 x 和 y 是 F 的满足某个方程的元素。在本部分椭圆曲线 E 由两个域元素 $a,b\in F$ 指定，a,b 称为 E 的系数。

p 是 F 的特征。

如果 $p>3$，则 a,b 满足 $4a^3+27b^2\neq 0_F$，E 上的每个点（无穷远点 o 除外）$P=(x,y)$ 满足以下方程：

$$y^2=x^3+ax+b$$

如果 $p=2$，则 b 满足 $b\neq 0_F$，E 上的每个点（无穷远点 o 除外）$P=(x,y)$ 满足以下方程：

$$y^2+xy=x^3+ax^2+b$$

如果 $p=3$，则 a,b 满足 $a\neq 0_F$ 和 $b\neq 0_F$，E 上的每个点（无穷远点 o 除外）$P=(x,y)$ 满足以下方程：

$$y^2=x^3+ax^2+b$$

椭圆曲线上的点组成了有限阿贝尔群，其中无穷远点 o 是恒等元。椭圆曲线的群组运算有很多方法，这些算法的实现不在本部分规定范围内。

注：椭圆曲线群组运算的有效实现参见 ISO/IEC 15946-1。

国家图书馆专用

附 录 B
（规范性附录）
转 换 函 数

B.1 八位位组串/比特串转换：OS2BSP 和 BS2OSP

八位位组串和比特串之间的转换函数 OS2BSP 和 BS2OSP 定义如下：

——函数 OS2BSP(x)，输入是八位位组串 x，输出是比特串 x；函数 BS2OSP(y)，输入是比特串 y，其长度是 8 的倍数，输出是一个八位位组串 x，所以有 y=OS2BSP(x)。

B.2 比特串/整数转换：BS2IP 和 I2BSP

比特串和整数之间的转换函数 BS2IP 和 I2BSP 定义如下：

——函数 BS2IP(x) 把比特串 x 映射成一个整数 x'。映射方法如下：如果 $x=\langle x_{l-1},\cdots,x_0\rangle$ 其中 $x_0,\cdots,x_{l-1}$ 是比特，则 x' 定义为 $x'=\Sigma_{0\leqslant i<l}x_i\times 2^i$。

——函数 I2BSP(m，l)，输入是两个非负整数 m 和 l，输出是一个长度为 l 的比特串 x，所以如果存在 x，则有 BS2IP(x)=m。否则函数失败。

非负整数 n 的比特长度是使用二进制表达的比特个数，例如$\lceil \log_2(n+1)\rceil$。为了表达简练，Oct(m)定义 Oct(m)=I2BSP(m，8)。

注：当且仅当 m 的比特长度大于 l 时 I2BSP(m，l)失败。

B.3 八位位组串/整数转换：OS2IP 和 I2OSP

八位位组串和整数之间的转换函数 OS2IP 和 I2OSP 定义如下：

——函数 OS2IP(x)，输入是八位位组串 x，输出是整数 BS2IP[OS2BSP(x)]；

——函数 I2OSP(m，l)，输入是两个非负整数 m 和 l，输出是一个长度为 l 的八位位组串 x，所以如果存在 x，则有 OS2IP(x)=m。否则函数失败。

非负整数 n 的比特长度是使用基于 256 表达的数字个数，例如$\lceil \log_{256}(n+1)\rceil$；长度值使用 $L(n)$ 表示。

注 1：当且仅当 m 的比特长度大于 l 时 I2OSP(m，l)失败。

注 2：八位位组 x 通常使用长度为 2 的十六进制格式，表示为 OS2IP(x)；当 OS2IP(x) < 16 时，代表比特串 0000 的“0”作为前缀。

B.4 有限域元素/整数转换：$FE2IP_F$

将有限域 F 的元素转换为整数值的函数 $FE2IP_F$ 定义如下：

——函数 $FE2IP_F$ 把元素 $a\in F$ 映射成整数值 a'。映射方法如下：如果 F 的基 $q=p^e$，其中 p 为素数，$e\geqslant 1$，则 F 的元素 a 是 e 元组 $(a_1,\cdots,a_e)$，其中对于 $1\leqslant i\leqslant e$，$a_i\in[0\,..\,p)$，a' 定义为 $a'=\Sigma_{1\leqslant i\leqslant e}a_i p^{i-1}$。

国家图书馆 藏书

B.5 八位位组串/有限域元素转换：$OS2FEP_F$ 和 $FE2OSP_F$

八位位组串和显式给定有限域 F 的元素之间的转换函数 $OS2FEP_F$ 和 $FE2OSP_F$ 定义如下：

——函数 $FE2OSP_F(a)$，输入是有限域 F 的元素，输出是八位位组串 $I2OSP(a',l)$，其中 $a'=FE2IP_F(a)$，l 是 $|F|-1$ 的八位位组个数，例如 $l=\lceil \log_{256}|F| \rceil$。如上述 $FE2OSP_F(a)$ 的输出总是长度为 $\lceil \log_{256}|F| \rceil$ 的八位位组串。

——函数 $OS2FEP_F(x)$，输入八位位组串 x，输出是(唯一的)域元素 $a \in {}_F$。所以如果存在 a，则有 $FE2OSP_F(a)=x$。否则函数失败。

注：当且仅当 x 的长度不等于 $\lceil \log_{256}|F| \rceil$，或者 $OS2IP(x) \geqslant |F|$，$OS2FEP_F(x)$ 失败；长度值表示为 L_F。

B.6 椭圆曲线/八位位组串转换：$EC2OSP_E$ 和 $OS2ECP_E$

B.6.1 压缩椭圆曲线点

显式给定有限域 F(基为 p)上的椭圆曲线 E。

一个点 $P \neq O$(不为无穷远点)可以使用压缩、非压缩或者混合形式表示。

如果 $P=(x,y)$，则(x,y)就是 P 的非压缩形式。

如上描述的椭圆曲线上的点 $P=(x,y)$。点 P 的压缩形式是$(x,\tilde{y})$，其中 $\tilde{y} \in \{0,1\}$按照如下决定：

——如果 $p \neq 2$ 和 $y=0_F$，则 $\tilde{y}=0$；

——如果 $p \neq 2$ 和 $y \neq 0_F$，则 $\tilde{y}=[(y'/p^f) \bmod p] \bmod 2$，其中 $y'=FE2IP_F(y)$，f 是最大的非负整数如 $p^f | y'$；

——如果 $p=2$ 和 $x=0_F$，则 $\tilde{y}=0$；

——如果 $p=2$ 和 $x \neq 0_F$，则 $\tilde{y}=\lceil z'/2^f \rceil \bmod 2$，其中 $z=y/x$，$z'=FE2IP_F(z)$，f 是最大的非负整数如 $2^f / FE2IP_F(1_F)$。

$P=(x,y)$的混合形式是$(x,\tilde{y},y)$，其中 $\tilde{y}$ 如上描述。

B.6.2 点解压缩算法

点的解压缩方法很多，例如从$(x,\tilde{y})$计算 y。主要方法描述如下：

——假设 $p \neq 2$，(x,y) 的压缩形式是$(x,\tilde{y})$。点(x,y)满足方程 $y^2=f(x)$，$f(x)$是参数为 x 的多项式。如果 $f(x)=0_F$，则对于 y 只有一种可能的选择，即 $y=0_F$。否则如果 $f(x) \neq 0$，则对于 y 有两种可能的选择(区别仅仅是符号不同)，由 $\tilde{y}$ 决定正确的选择。有一些比较有名的在有限域里计算平方根的算法，所以 y 的两个选择很容易计算。

——假设 $p=2$，(x,y) 的压缩形式是$(x,\tilde{y})$。点(x,y)满足方程 $y^2+xy=x^3+ax^2+b$。如果 $x=0_F$，则得出 $y^2=b$，y 能够唯一确定并计算。否则如果 $x \neq 0_F$，则设置 $z=y/x$，则有 $z^2+z=g(x)$，其中 $g(x)=(x+a+bx^{-2})$。y 通过值 z 和 x 能够唯一确定并计算。计算 z：对于固定的 x 值，如果 z 是方程 $z^2+z=g(x)$的一个解，则另一个解是 $z+1_F$。很容易地计算 z 的两个候选值，由 $\tilde{y}$ 决定 z 的正确选择。

国家图书馆专用

B.6.3 逆运算

显式给定有限域 F 上的椭圆曲线 E。

椭圆曲线 E 上的点和八位位组串之间的转换函数 $EC2OSP_E$ 和 $OS2ECP_E$ 定义如下：

a) 函数 $EC2OSP_E(P, fmt)$，输入是椭圆曲线 E 上的点 P 和格式符 fmt(压缩、非压缩或混合)。输出是八位位组串 EP，计算如下：

1) 如果 $P=O$，则 $EP=Oct(0)$。

2) 如果 $P=(x,y)\neq O$，压缩格式$(x,\tilde{y})$，则 $EP=H||X||Y$，其中：

i) H 是格式为 $Oct[4U+C\cdot(2+\tilde{y})]$的八位位组，其中：

I) 如果 fmt 是非压缩或混合格式，则 $U=1$，否则 $U=0$；

II) 如果 fmt 是压缩或混合格式，则 $C=1$，否则 $C=0$。

ii) X 是八位位组串 $FE2OSP_F(x)$。

iii) 如果 fmt 是非压缩或混合格式，则 Y 是八位位组串 $FE2OSP_F(y)$，否则 Y 是空八位位组串。

b) 函数 $OS2ECP_E(EP)$，输入是八位位组串 EP。如果在椭圆曲线 E 上存在点 P 及格式符 fmt 如 $EC2OSP_E(P, fmt)=EP$，则函数输出 P(非压缩)，否则函数失败。注意点 P 如果存在，则是唯一定义的，所以函数 $OS2ECP_E(EP)$ 也是确定的。

注：如果格式符 fmt 是非压缩的，则 $\tilde{y}$ 值不需要计算。

国家图书馆 专用

附 录 C
（规范性附录）
掩码生成函数（密钥导出函数）

本附录描述了本部分使用的“掩码生成函数”及实现规范。

掩码生成函数 MGF＊（x，l），输入是一个八位位组串 x 和一个整数 l，输出是一个长度是 l 的八位位组串。八位位组串 x 是任意长度，实现者可以定义 x 的最大长度和 l 的最大值。

注：在一些文档和标准里，术语“密钥导出函数”替代“掩码生成函数”。

C.1 允许的掩码生成函数

本部分允许使用的掩码生成函数是 MGF1，见 C.2，以及 MGF2，见 C.3。

C.2 MGF1

MGF1 是掩码生成函数家族，通过如下的系统参数配置：

——Hash：杂凑函数。

输入是一个八位位组串 x 和一个非负整数 l 的 MGF1(x，l) 定义如下：

$$\{\mathrm{Hash}[x||\mathrm{I2OSP}(0,4)]||\mathrm{Hash}[x||\mathrm{I2OSP}(1,4)]||\cdots||\mathrm{Hash}[x||\mathrm{I2OSP}(k-1,4)]\}^{8l}$$

其中：$k=\lceil l/L_{\mathrm{Hash}}\rceil$。

C.3 MGF2

MGF2 是掩码生成函数家族，通过如下的系统参数配置：

——Hash：杂凑函数。

输入是一个八位位组串 x 和一个非负整数 l 的 MGF2(x，l) 定义如下：

$$\{\mathrm{Hash}[x||\mathrm{I2OSP}(1,4)]||\mathrm{Hash}[x||\mathrm{I2OSP}(2,4)]||\cdots||\mathrm{Hash}[x||\mathrm{I2OSP}(k,4)]\}^{8l}$$

其中：$k=\lceil l/L_{\mathrm{Hash}}\rceil$。

注：除了计数器从 1～k，而不是从 0～$k-1$ 之外，MGF2 和 MGF1 是相同的。

国家图书馆专用

附 录 D
（资料性附录）
数据输入生成方式实例

本附录描述了 7.4.3 和 7.5.4 里使用的带有填充冗余的数据输入生成方式及冗余的校验。这个方式可以和本部分中描述的以下机制一起使用：NR、ECNR、ECMR 和 ECKNR。

D.1 拆分消息和生成数据输入

A 选择杂凑函数 Hash：$\{0,1\}^{8*}\rightarrow\{0,\}^{8L_{red}}$。$A$ 也指定杂凑函数标识符选项的使用。当杂凑函数标识符合理时，A 设置 $L_{HashID}=1$，否则 $L_{HashID}=0$。当 $L_{HashID}=1$ 时，A 为杂凑函数标识符设置一个八位位组串（即 trailer），否则设置空八位位组串。以上信息应做为域参数提供。每个机制规定数据输入的长度；L_{dat} 表示数据输入的八位位组个数。

从消息 M 中生成数据输入 d，按照如下或者等同如下的步骤进行：

a） 计算可恢复部分的最大长度 $L_{max}=L_{dat}-L_{red}-L_{HashID}$。

b） 按照如下方法将消息 M 拆分成可恢复部分 M_{rec}（最左侧八位位组）及 M_{clr}（余下的八位位组）：

1） 如果 $L(M)\leqslant L_{max}$，则设置 $M_{rec}=M$ 和 $M_{clr}=O$（空八位位组串）；

2） 如果 $L(M)>L_{max}$，则将消息 M 拆分成 M_{rec} 和 M_{clr}，例如 $M=M_{rec}||M_{clr}$，满足 $L_{max}>L(M_{rec})$。

c） 将长度转换成八位位组串 $C_{rec}=I2OSP(L_{rec},8)$ 和 $C_{clr}=I2OSP(L_{clr},8)$。

d） 计算杂凑权标 $h\in\{0,1\}^{8L_{red}+8L_{HashID}}$，$h=Hash(C_{rec}||C_{clr}||M_{rec}||M_{clr}||\Pi)||$ trailer。

e） 计算填充串 $pad=I2OSP(0,L_{max}-L_{rec})$。

f） 生成数据输入 $d\in\{0,1\}^{8L_{dat}}$，$d=pad||h||M_{rec}$。

A 在已签消息里应包括长度 L_{rec}，签名 (r,s) 和不可恢复消息部分 M_{clr}。

D.2 校验冗余

B 接收到已签消息，其包括签名的第一部分 r' 和第二部分 s'，可恢复部分长度 L'_{rec} 和不可恢复消息部分 M'_{clr}。从接收到的签名 (r',s') 可以恢复预签名 $\Pi'\in\{0,1\}^{8*}$ 和数据输入 $d'\in\{0,1\}^{8L'_{dat}}$。

B 校验签名和恢复消息，按照如下或者等同如下的步骤进行：

a） 计算 $L_{max}=L_{dat}-L_{red}-L_{HashID}$；

b） 检查是否 $L'_{rec}\in[0,L_{max}]$，如果不是，则拒绝签名；

c） 恢复填充八位位组，杂凑权标和可恢复部分：$pad'=[d']^{L_{max}-L'_{rec}}$，

$$h'=[[d']_{8L_{red}+8L_{HashID}+8L'_{rec}}]^{8L_{red}+8L_{HashID}}$$ 和 $M'_{rec}=[d']_{8L'_{rec}}$；

d） 校验填充：如果 OS2IP(pad′)=0 不成立，则拒绝签名；

e） 计算长度 $L'_{clr}=L(M'_{clr})$；

f） 将长度转换为八位位组串 $C'_{rec}=I2OSP(L'_{rec},8)$ 和 $C'_{clr}=I2OSP(L'_{clr},8)$；

g） 重新计算杂凑权标 $h''=Hash(C'_{rec}||C'_{clr}||M'_{rec}||M'_{clr}||\Pi')||$ trailer；

h） 校验填充冗余：如果 $h'=h''$ 不成立，则拒绝签名；

i） 输出 $M'_{rec}||M'_{clr}$。

国家图书馆试用

附 录 E
（资料性附录）
ASN.1 模块

E.1 形式定义

本附录定义了 ASN.1 模块，其包括本部分规定的带消息恢复的数字签名的抽象语法。

```
MessageRecoverySignatureMechanisms {
iso(1) standard(0) signature-schemes(9796) part(3) asn1-module(1)
message-recovery-signature-mechanisms(0)
}
DEFINITIONS EXPLICIT TAGS ::=BEGIN
IMPORTS
HashFunctions
FROM DedicatedHashFunctions {
iso(1) standard(0) encryption-algorithms(10118) part(3) asn1-module(1)
dedicated-hash-functions(0) } ;
OID ::=OBJECT IDENTIFIER -- alias
SignatureWithMessageRecovery ::=SEQUENCE {
algorithm ALGORITHM.&id({MessageRecovery}),
parameters ALGORITHM.&Type({MessageRecovery}{@algorithm}) OPTIONAL
}
signatureMechanism OID ::={
iso(1) standard(0) hash-functions(9796) part3(3) mechanism(0)
}
MessageRecovery ALGORITHM ::={
dswmr-nr |
dswmr-ecmr |
dswmr-ecknr |
dswmr-ecpv |
dswmr-ecnr,
... -- Expect additional algorithms --
}
dswmr-nr ALGORITHM ::={
OID nr PARMS HashFunctions
}
dswmr-ecmr ALGORITHM ::={
OID ecmr PARMS HashFunctions
}
dswmr-ecknr ALGORITHM ::={
OID ecknr PARMS HashFunctions
```

国家图书馆专用

```
}
dswmr-ecpv ALGORITHM ::={
OID ecpv PARMS HashFunctions
}
dswmr-ecnr ALGORITHM ::={
OID ecnr PARMS HashFunctions
}
-- Cryptographic algorithm identification -
ALGORITHM ::=CLASS {
&id OBJECT IDENTIFIER UNIQUE,
&Type OPTIONAL
}
WITH SYNTAX { OID &id [PARMS &Type] }
-- Message recovery signature mechanisms --
nr OID ::={ signatureMechanism nr(0) }
ecmr OID ::={ signatureMechanism ecmr(1) }
ecknr OID ::={ signatureMechanism ecknr(3) }
ecpv OID ::={ signatureMechanism ecpv(4) }
ecnr OID ::={ signatureMechanism ecnr(5) }
END -- MessageRecoverySignatureMechanisms -
```

E.2 后续对象标识符的使用

任何一个签名机制都使用杂凑函数。因此后续对象标识符可以参考杂凑函数(例如,ISO/IEC 10118-3 规定的专用的杂凑函数)。

国家图书馆 专用

附 录 F
（资料性附录）
算 例

F.1 NR 算例

注 1：在 F.1 里，参考数据的 ASCII 编码；其等同于 ISO 646 编码。

注 2：F.1.2、F.1.3 和 F.1.4 使用 F.1.1 描述的域参数，用户密钥，随机数和消息。

F.1.1 部分恢复算例

P
ffffffff ffffffff c90fdaa2 2168c234
c4c6628b 80dc1cd1 29024e08 8a67cc74 020bbea6 3b139b22 514a0879
8e3404dd ef9519b3 cd3a431b 302b0a6d f25f1437 4fe1356d 6d51c245
e485b576 625e7ec6 f44c42e9 a637ed6b 0bff5cb6 f406b7ed ee386bfb
5a899fa5 ae9f2411 7c4b1fe6 49286651 ece65381 ffffffff ffffffff

Q
7fffffff ffffffff e487ed51 10b4611a
62633145 c06e0e68 94812704 4533e63a 0105df53 1d89cd91 28a5043c
c71a026e f7ca8cd9 e69d218d 98158536 f92f8a1b a7f09ab6 b6a8e122
f242dabb 312f3f63 7a262174 d31bf6b5 85ffae5b 7a035bf6 f71c35fd
ad44cfd2 d74f9208 be258ff3 24943328 f67329c0 ffffffff ffffffff

Q 的长度 1 023 比特

G 2

签名密钥 x_A
fcf63b30 a349edc2 b135b0d4 fbcf2900
8c9de512 d033dbd5 32e513c3 b2501ef7 c3bae4a5 f1368abc 4f5643e1
0f737660 c9aa959f 8362bc82 7771f89e 88a1bcbc 3276d52b 3e1ab0fc
f398c937 9370241e 66b87ef9 78555971 3282a0ac 7ca11239 976f6605
29b4bc4c 7d0c9412 9ac52410 3a0eed44 f1aaa99f b1791059 0378b037

校验密钥 Y_A
a544638a d770ce35 c5286db8 3c124a77
f382bc7c ed585501 371928f8 1bc5e61f da841361 08beab18 e84f46d6
5cd0a9f2 4a00998d 37312a2e f28f7370 b95ce7ff 2cee0be9 1457beb0
9fe790f1 e31de199 1ca3b8db 7de3f13c 8add8e02 5eaa7a41 3ee276da
364bf447 52022ca5 48133f7c 57e94a0c 20cbff8e 98660f98 e034fe4c

随机数 k
1698cc3 2a59174b 93511339 528fb5d8
ba386493 85630f0a 9624f5ab 71a5ccf9 29c63f3e 0e36a339 207685a4
12cec6a4 3f0ae734 bfd30703 83109786 101b036d e83b4954 048217c2
6d76a398 f7afd556 9e1cf908 091be435 de10c379 35aa8896 ee34df2a
1b29866f 29256ea5 8e2c2558 0cd65489 99579211 c5aad05f ddbda767

预签名 Π
0ebc1b0b baf3c121 ff29d858 7c35e42b
5614ff11 aa40ceed 454c57b6 dd3a7ee0 7732420e 7c8c7b18 2c7aaccc
52c798c0 2ec6e2bc bb67256e 032c0e13 2eaa8ca8 1dab8404 73e81f61
912827b6 23d65fac 29f5414a 2ce7ce88 07fe6891 c58aaf05 e8546e83
196b0f62 6873befe 51c0b7e3 b8ac49b2 5f416791 e0dacc23 f41f25d5

签名消息
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789

国家图书馆专用

消息	41424344	45464748	494a4b4c	4d4e4f50	51525354	55565758	
	595a6162	63646566	6768696a	6b6c6d6e	6f707172	73747576	7778797a
	30313233	34353637	38394142	43444546	4748494a	4b4c4d4e	4f505152
	53545556	5758595a	61626364	65666768	696a6b6c	6d6e6f70	71727374
	75767778	797a3031	32333435	36373839	41424344	45464748	494a4b4c
	4d4e4f50	51525354	55565758	595a6162	63646566	6768696a	6b6c6d6e
	6f707172	73747576	7778797a	30313233	34353637	38394142	43444546
	4748494a	4b4c4d4e	4f505152	53545556	5758595a	61626364	65666768
	696a6b6c	6d6e6f70	71727374	75767778	797a3031	32333435	36373839

F.1.2 ISO/IEC 10118-3 规定的专用的杂凑函数 3(SHA1)算例

杂凑权标长度							21 个八位位组
可恢复长度 L_{rec}						00000000	0000006a
不可恢复长度 L_{clr}						00000000	0000008e
杂凑码			005e4e9b	e8c9a202	80ffab58	d9927041	80dcc44d
杂凑函数标识符							33
数据输入 d				5e4e	9be8c9a2	0280ffab	58d99270
	4180dcc4	4d334142	43444546	4748494a	4b4c4d4e	4f505152	53545556
	5758595a	61626364	65666768	696a6b6c	6d6e6f70	71727374	75767778
	797a3031	32333435	36373839	41424344	45464748	494a4b4c	4d4e4f50
	51525354	55565758	595a6162	63646566	6768696a	6b6c6d6e	6f707172
签名第一部分 r				0ebc795a	56dc8ac4	01aad803	d50f769b
	9795dbd5	f774102f	88909cfd	2482c82a	c27e8f5c	cbdccc6a	7fcf0222
	aa1ff21a	90294621	20cd8cd6	6c96797f	9c18fc18	8f1df778	e95e96da
	0aa257e7	560993e1	602c7983	6e2a11cc	4d44afda	0ed4fa52	35a2bdd3
	6abd62b6	bdca1656	ab1b1946	1c10af18	c6a9d0fc	4c473992	638f9747
签名第二部分 s				1ecf7056	cac6b0d4	a951f8b6	9e9c191f
	930a101e	f3f891ff	d1636615	b2444590	c1a0e3ee	af8f701d	4a796761
	d64fcda2	7622fe9f	f0645eba	617e9747	2bafc0bf	f487efd0	2d2ca4c1
	7705a1e6	0c68c6a9	fadd5ca5	43988d5f	a338f5e1	5bb59edf	41ce6ecc
	2c8832f2	a0565e81	f1696845	2f99ae59	ad24c5d8	bb70a148	9f65a37d

F.1.3 ISO/IEC 10118-3 规定的专用的杂凑函数 1(RIPEMD-160)算例

杂凑权标长度							21 个八位位组
可恢复长度 L_{rec}						00000000	0000006a
不可恢复长度 L_{clr}						00000000	0000008e
杂凑码			525d1604	e8a2a6f6	054ba7a9	ffc4a18e	bab0fe2b
杂凑函数标识符							31
数据输入 d				525d16	04e8a2a6	f6054ba7	a9ffc4a1
	8ebab0fe	2b314142	43444546	4748494a	4b4c4d4e	4f505152	53545556
	5758595a	61626364	65666768	696a6b6c	6d6e6f70	71727374	75767778
	797a3031	32333435	36373839	41424344	45464748	494a4b4c	4d4e4f50
	51525354	55565758	595a6162	63646566	6768696a	6b6c6d6e	6f707172
签名第一部分 r				0f0e7821	bfdc63c8	f52f2400	2635a8cc
	e4cfb00f	d572102f	88909cfd	2482c82a	c27e8f5c	cbdccc6a	7fcf0222
	aa1ff21a	90294621	20cd8cd6	6c96797f	9c18fc18	8f1df778	e95e96da
	0aa257e7	560993e1	602c7983	6e2a11cc	4d44afda	0ed4fa52	35a2bdd3

国家图书馆

	6abd62b6	bdca1656	ab1b1946	1c10af18	c6a9d0fc	4c473992	638f9747
签名第二部分 s				3e1bf266	a2fe5226	79192ef9	14e4f648
	3a89a3c4	87243e86	beecfae9	dabbec98	eaff37d0	b3eaab2c	2308becc
	b3681577	9de664bb	4547c06c	8e456be2	24488268	649c30e2	ffb25460
	86745066	20e5c853	d6194981	607a2386	be38f463	dd820d10	32771638
	7c874364	1ab116eb	00421592	e70b7281	2746acfc	19b601fc	6de5a89d

F.1.4 ISO/IEC 10118-3 规定的专用的杂凑函数 2(RIPEMD-128)算例

杂凑权标长度							17 个八位位组
可恢复长度 L_{rec}							00000000 0000006e
不可恢复长度 L_{clr}							00000000 0000008a
杂凑码				ab8fd266	ddddbddc	48d117ea	f0968b0c
杂凑函数标识符							32
数据输入 d				ab8fd2	66ddddbd	dc48d117	eaf0968b
	0c324142	43444546	4748494a	4b4c4d4e	4f505152	53545556	5758595a
	61626364	65666768	696a6b6c	6d6e6f70	71727374	75767778	797a3031
	32333435	36373839	41424344	45464748	494a4b4c	4d4e4f50	51525354
	55565758	595a6162	63646566	6768696a	6b6c6d6e	6f707172	73747576
签名第一部分 r				0f67aade	21d19edf	db72a970	67267ab6
	62474053	ed851433	8c94a101	2886cc2e	c6829360	cfe0d06e	83d30626
	b429fc24	942d4a25	24d190da	709a7d83	a01d001c	9321fb7c	ed624f92
	c35b5beb	5a0d97e5	6b37848e	722e15d0	5148b3de	12d8fe56	39a6c1d7
	6ec166ba	c1ce2060	b525id4a	2014b31c	caadd500	504b3d96	67939b4b
签名第二部分 s				64dc5bce	568cb0be	22ea47f7	d848a5ef
	fc34fdea	0f11ed67	ee24753f	655e72fa	c0d12fed	da5f0c13	9c9d1544
	8cce2297	6a2b0fb0	00055fd8	4e0d38b9	86fde806	fc74e1d4	ddd8144d
	dd5530a1	66fd03aa	11003478	06e5678f	7dd9927a	5834c0d2	cdffb15c
	14dec608	bb6eac7c	15a3c6c7	05de2a82	4b5a3e9f	f4b26171	9b8daf16

F.1.5 完全恢复(RIPEMD-128)算例

P			1	5654b2a4	8af38b0b	45b10960	41a7f552
	4a97a065	fff0bb31	94cae13e	38c2969e	527dc350	e5b32309	fc3342fb
	741c4294	54020173	aaf8a23d	2ca4a294	27bd8c1c	6384db95	5c944e40
	c321a896	b4d50969	e869d23f	49bf2489	c918c3b3	636e4907	3162512d
	5ce35acc	858f70b6	daaf970e	0086bad1	2062a127	a2afeee0	5dfd9e3d
Q		1	9cafd651	31c5c9a7	d546e3f9	42577f24	220f1b07
Q 的长度							161 比特
G			1	3e2cfad2	bd5e8128	c6f968df	664ab926
	9d3b1ae3	a558100b	22682671	46421b71	43eb37ef	659e992a	0746c1df
	dc7b899e	735063d4	e4a9dcad	46f85bb5	4ffe1774	62e18fe4	43efb87f
	cda522bf	f3097853	d5a1f723	bcde771f	903b7c0a	89974ab2	efc94b69
	4590b2b1	02ed7160	f207d18b	0c748186	34118dbe	c1ff775a	b3a16be4
签名密钥 x_A			478bbe64	7cd50ef3	67ebe30f	dc10c9e0	1ce37fb5
校验密钥 Y_A				6ce2e099	1ca9f778	571ab62d	d535bbd7
	260a481f	19619944	0739667f	5978cc7c	7eb25030	d3abe64a	b599c1af
	6414bed8	3505e2c0	8a42acf7	f2fdab50	6963399c	f5b7303d	8953b565
	edb0efee	6d8ae8af	eeba1890	63c72571	b586092b	1fc0f9d9	d1f82cd0
	6bf307bc	a385dc4c	1a8cfc87	9bc622d1	135277ac	7264ebef	b1fd4127

国家图书馆 备用

随机数 k	5a474224 778948f7 c2aa8890 61fbb3a9 750ec2cb
Π	00 9981665a fddb49bd 99b94480 d0f314e0
	9db3539e 2874acf5 59ed6376 15886470 f9a85e0f 98631dc1 516a6c4b
	68c7c35d 28efcb29 c1ecd84f ed57a9ad a22d8f3f 57247312 e12c21b9
	21d792be c964aed2 48b440b7 a8043ef7 fec79008 186749fd c11f4f6f
	fe1b734a 4a504fb3 eca68d2f 8f105a52 fe97effc 0e67fad1 14de9d02
签名消息	Plaintext
M	70 6c61696e 74657874
杂凑权标截取长度	10 字节
可恢复长度 L_{rec}	00000000 00000009
不可恢复长度 L_{clr}	00000000 00000000
截取的杂凑权标	c42a ebdb3c89 50e9beff
数据输入 d	c42aeb db3c8950 e9beff70 6c61696e 74657874
签名第一部分 r	9af324f6 16d169d0 f1fe0dc1 7d96d4b7 e5e9f641
签名第二部分 s	dc2c286e 4d5e6c1e 14551148 12eecfc5 a8f123eb

F.2 ECNR 算例

注 1：杂凑函数 Hash(T)＝RIPEMD-160($T||C$)，其中 C＝00000001(十六进制)。

注 2：$H||M_{rec}$ 的最右侧 L_{dat} 八位位组作为数据输入，其中 H 是 Hash($C_{rec}||C_{clr}||M_{rec}||M_{clr}||\Pi$)截取的杂凑权标。截取的杂凑权标是杂凑权标的最左侧 L_{red} 八位位组。另外 C_{rec}＝I2OSP(L_{rec}，4)，C_{clr}＝I2OSP(L_{clr}，4)。

F.2.1 素数域上的椭圆曲线

P	ffd5d55f a9934410 d3eb8bc0 4648779f 13174945
椭圆曲线方程	$y^2 \equiv x^3 + ax + b \pmod{p}$
A	710062dc b53dc6e4 2f8227a4 fbac2240 bd3504d4
B	4163e75b b92147d5 4e09b0f1 3822b076 a0944359
G 的 x 坐标	3c1e27d7 1f992260 cf3c31c9 0d80b635 e9fd0e68
G 的 y 坐标	c436efc0 041bbf09 47a304a0 05f8d43a 36763031
n (G 的阶)	2aa3a38f f1988b58 235241ee 59a73f46 46443245
n 的长度(比特)	158 比特
$L_{(n)}$	20
L_{dat}	19
L_{red}	9
L_{rec}	10
L_{clr}	13
签名密钥 x_A	24a3a993 ab59b12c e7379a12 3487647e 5ec9e0ce
Y_A 的 x 坐标	e564ac ae27d227 1c4af829 cface6de cc8cdce6
Y_A 的 y 坐标	7bd48ce1 08ffd3cf a38177f6 83b5bcf4 fd97a4a9
k	08a8bea9 f2b40ce7 40067226 1d5c05e5 fd8ab326
$kG=(x,y)$	
x	177b7c44 ac2f7f79 96aefd27 c68d59e0 f8e01599
y	399ea116 298975bb 449d126f 6c97bddf c4e8782e
Π	02 177b7c44 ac2f7f79 96aefd27 c68d59e0 f8e01599
签名消息	This is a test message!
M	546869 73206973 20612074 65737420 6d657373 61676521
M_{rec}	5468 69732069 73206120

国家图书馆 样张

M_{clr}	74 65737420 6D657373 61676521
杂凑输入	0000000a 0000000d 54686973 20697320 61207465
	7374206d 65737361 67652102 177b7c44 ac2f7f79
	96aefd27 c68d59e0 f8e01599
截取的杂凑权标 H	64 1e6fe77e b1b9cca9
数据输入 $d = H \| M_{rec}$	641e6f e77eb1b9 cca95468 69732069 73206120
签名第一部分 r	1833eff5 4087a911 bb7d3a63 fc2982ff 20ce1b7d
签名第二部分 s	155d498e 35855ab5 04b9adda 0315ca77 4b171e61

F.2.2 扩展域 GF(2^m)上的椭圆曲线

Galois field GF(2^{185}) 的多项式是 $x^{185}+x^{69}+1$。

椭圆曲线方程	$y^2 + xy = x^3 + ax^2 + b$
A	07 2546b543 5234a422 e0789675 f432c894 35de5242
B	00 c9517d06 d5240d3c ff38c74b 20b6cd4d 6f9dd4d9
G 的 x 坐标	07 af699895 46103d79 329fcc3d 74880f33 bbe803cb
G 的 y 坐标	01 ec23211b 5966adea 1d3f87f7 ea5848ae f0b7ca9f
N	04 00000000 00000000 0001e60f c8821cc7 4daeafc1
n 的长度	163 比特
$L(n)$	21
L_{dat}	20
L_{red}	10
L_{rec}	10
L_{clr}	13
x_A	03 d648bcb2 e4d5d151 656c8477 4ed016ba 292a5a38
Y_A 的 x 坐标	07 01b9786f d72171da a883f34c 44deeace a10b8d02
Y_A 的 y 坐标	00 0149bdc1 54c6ab7f 4e5b4a4a 57d528d7 65d7f8ea
k	02 887ac572 8a839081 8b535fcb f04e827b 0f8b543c
$kG = (x, y)$	
x	00 eeddbbcf 22652313 c3484118 5d3ebb53 8c453aee
y	03 7df0f68a c78cd813 0a6ffeda 5ba85ff1 14e93ec7
II	0200 eeddbbcf 22652313 c3484118 5d3ebb53 8c453aee
签名消息	This is a test message!
M	546869 73206973 20612074 65737420 6D657373 61676521
M_{rec}	5468 69732069 73206120
M_{clr}	74 65737420 6D657373 61676521
杂凑输入	00 00000a00 00000d54 68697320
	69732061 20746573 74206d65 73736167 65210200
	eeddbbCf 22652313 c3484118 5d3ebb53 8c453aee
截取的杂凑权标 H	d8f3 55fde3c6 1cb29bc0
数据输入 $d = H \| M_{rec}$	d8f355fd e3c61cb2 9bc05468 69732069 73206120
签名第一部分 r	01 c7d111cd 062b3fc6 5e158d9c 85a37816 280dbb8e
签名第二部分 s	01 0fe6b789 d7ef86bc 5ed726ca 0fc7c96c f6b4faa3

F.2.3 扩展域 GF(p^m)上的椭圆曲线

p	fffffffb
m	5
不可约多项式	$X^5 - 2$
椭圆曲线方程	$y^2 = x^3 + ax + b$

国家图书馆专用

a	00000000 00000000 00000000 00000000 00000000 00000000 00000001
b	00000000 00000000 00000000 00000000 00000000 00000001 00000106
G 的 x 坐标	fcdee3ee eb6a9d0c 821c8b46 d27937bc 0fbac840
G 的 y 坐标	3c329e0d 7a5fb6e4 048a69c1 12f8cb35 dffb7ccc
n	fffffffe 000000f9 fffe3308 f697c1d6 d7de35cf
n 的长度	160 比特
$L(n)$	20
x_A	d648bcb2 e4d5d151 656c8477 4ed016ba 292a5a38
Y_A 的 x 坐标	be00180e c77feb6e a550dbf6 a6d5ccce 8b1f7cf6
Y_A 的 y 坐标	13ad8b66 c59205f7 71112f36 effa0650 72487bef
k	887ac572 8a839081 8b535fcb f04e827b 0f8b543c
$kG=(x,y)$	
x	c9c83609 b667081f 09d4f822 325daa91 01e06c84
y	4e95c220 783a466f 2d2f12aa 6ee07c60 928d2594
Π	02 c9c835f9 f2c2cfd6 951b2642 2531d251 8d5547d7
签名消息	This is a test message!
M	546869 73206973 20612074 65737420 6d657373 61676521
M_{rec}	5468 69732069 73206120
M_{clr}	74 65737420 6d657373 61676521
杂凑输入	0000000a 0000000d 54686973
	20697320 61207465 7374206d 65737361 67652102
	c9c835f9 f2c2cfd6 951b2642 2531d251 8d5547d7
截取的杂凑权标 H	547a d9d64b5e 9b62e920
数据输入 $d=H\|\|M_{rec}$	7ad9d6 4b5e9b62 e9205468 69732069 73206120
签名第一部分 r	ca431002 3e216945 7e3f1498 a1756f0d 50b93d59
签名第二部分 s	951cd069 e020eb4d 3da1c3dc e316819c 260c8d36

F.3 ECMR 算例

F.3.1 素数域上的椭圆曲线

注 1：截取的杂凑权标 h 是专用的杂凑函数 3(即 SHA-1)的输出 $\Pi||M$ 的前 L 八位位组。

注 2：掩码函数是 SHA-1。

p	ffffffff ffffffff ffffffff ffffffff ffff7c67
椭圆曲线方程	$y^2\equiv x^3+ax+b \pmod p$
A	ffffffff ffffffff ffffffff ffffffff ffff7c64
B	26c1d102 82415e10 a4995e19 80b59224 d7120957
椭圆曲线上点的个数	ffffffff ffffffff ffffc748 a4eea1b0 dc8744b9
G 的 x 坐标	1
G 的 y 坐标	22e0d7c6 1eb0627b 334456c7 a50b77fd a9007da6
N	ffffffff ffffffff ffffc748 a4eea1b0 dc8744b9
n 的长度	160 比特
签名密钥 x_A	ddd259e3 d30a77ab c31cdf29 9a0e6cff 7d78f869
Y_A 的 x 坐标	6de7e135 f5b2ad0c e33492fa 61ed55b0 a00be7ba
Y_A 的 y 坐标	79473d9c ea21791a 391d536c 99ebfb13 4b94c3cc
随机数 k	4b13079f a8f2992e 5bcdb38d 6895a31b 91d822c2
kG 的 x 坐标	b4f8c602 dec23b19 358271b8 fe868ad4 a7f7fa9f

国家图书馆专用

kG 的 y 坐标	691b933a c8e59096 63364135 f5015637 f337f424
Π= Mask[EC2OSP$_h$(kG)]	e5a0d1f5 85a1cf2c 431a8c61 b78f316c 80d8928f
签名消息	TestVector
M(=M_{rec}) (M_{clr} 为空)	5465 73745665 63746f72
截取的杂凑权标长度 L(=L_1)	10 个八位位组
可恢复长度 L_{rec}	10 个八位位组
不可恢复长度 L_{clr}	0 个八位位组
截取的杂凑权标 h	3b16 b61a504b 21855dfc
数据输入 $d = h \| M$	3b16b61a 504b2185 5dfc5465 73745665 63746f72
签名第一部分 r	deb667ef d5eaeea9 1ee6d804 c4fb6709 e3acfdfd
签名第二部分 s	1f5d610b b13e61c9 03a24f8f 1af14c0a 122cc560

F.3.2 扩展域 GF(2^m)上的椭圆曲线

Galois field GF(2^{163}) 的多项式是 $x^{163}+x^7+x^6+x^3+1$。

注 1：这是标准多项式基本实现。

注 2：掩码函数是基于 SHA-1 的 MGF1。

椭圆曲线方程	$y^2 + xy = x^3 + ax^2 + b$
a	1
b	2 0a601907 b8c953ca 1481eb10 512f7874 4a3205fd
椭圆曲线上点的个数	8 00000000 00000000 000525fc efce1825 48469866
G 的 x 坐标	3 f0eba162 86a2d57e a0991168 d4994637 e8343e36
G 的 y 坐标	0 d51fbc6c 71a0094f a2cdd545 b11c5c0c 797324f1
n	4 00000000 00000000 000292fe 77e70c12 a4234c33
n 的长度	163 比特
签名密钥 x_A	2 ddd259e3 d30a77ab c31cdf29 9a0e6cff 7d78f869
Y_A 的 x 坐标	6 a15faa2f 38cabcbc 48113b58 6c5148a7 f80c424c
Y_A 的 y 坐标	3 302077a6 3ea741d4 ecf200cf 68cd272f b21eefdc
随机数 k	3 97e49b66 4b13079f a8f2992e 5bcdb38d 6895a31b
kG 的 x 坐标	7 3b811311 c037c110 38350437 95543abd 067af556
kG 的 y 坐标	6 0f7b188b 0ad4345b 910c0a1f 7b301c31 f9f8d9e1
Π= Mask[EC2OSP$_h$(kG)]	e7 acd53a64 16db34c1 788b2011 edaa0db7 9bbd9a21
签名消息	TestVector
M(=M_{rec}) (M_{clr} 为空)	5465 73745665 63746f72
截取的杂凑权标长度 L(=L_1)	11 个八位位组
可恢复长度 L_{rec}	10 个八位位组
不可恢复长度 L_{clr}	0 个八位位组
截取的杂凑权标 h	c76dd5 cc49fa1a bc0aabb4
数据输入 $d = h \| M$	c7 6dd5cc49 fa1abc0a abb45465 73745665 63746f72
签名第一部分 r	20 c100f62d ecc188cb d33f7474 9ede5bd2 f8c9f553
签名第二部分 s	0 2dd0bfcb f8745141 33cdf701 fe774ae3 ff2d7d16

F.3.3 扩展域 GF(p^m)上的椭圆曲线

注 1：GF(p^m)里的元素 τ 定义为 $t_4x^4+t_3x^3+t_2x^2+t_1x+t_0$，表达为 $t_4\ t_3\ t_2\ t_1\ t_0$。

注 2：掩码函数是 SHA-1。

p	ffffff47
m	5
不可约多项式	$X^5 - 2$

国家图书馆 发布

椭圆曲线方程	$y^2 \equiv x^3 + ax + b \pmod{p}$
a	00000000 00000000 00000000 00000000 ffffff44
b	39cd7fda f41a7fb5 488651a5 e362f27f b449e900
椭圆曲线上点的个数	fffffc63 000538e9 fc3bbe32 da01dc69 c2516d77
G 的 x 坐标	00000000 00000000 00000000 00000000 00000002
G 的 y 坐标	8a45f6c7 82f3c45e e2716ce9 26573f3f c5105399
n	fffffc63 000538e9 fc3bbe32 da01dc69 c2516d77
n 的长度	160 比特
x_A	7b5f8464 0e65495c 87e807aa 22b446fb 34e77471
Y_A 的 x 坐标	a39e766 99f8ec98 26c87346 6dd50ba2 94116c31
Y_A 的 y 坐标	9b2ef2cf 22061787 54b154b9 acc2a731 359b675b
随机数 k	8819bc2c 9ef7fdfc 2c348697 cadbc2be 77349b87
kG 的 x 坐标	c4e47f64 de0d2859 33af7a91 c5252d1f 671b20a2
kG 的 y 坐标	ca71997e 285b76f9 292af138 c4267642 eb1458b9
Π= Mask[EC2OSP$_E$(kG)]	d1fdf8a3 d5bcb759 7cc5b859 1c2d2269 2e0a7cd2
签名消息	TestVector
M(=M_{rec}) (M_{clr} 为空)	5465 73745665 63746f72
截取的杂凑权标长度 L(=L_1)	10 个八位位组
可恢复长度 L_{rec}	10 个八位位组
不可恢复长度 L_{clr}	0 个八位位组
截取的杂凑权标 h	55d7 dd9166fd 83eca94f
数据输入 $d = h \| M$	55d7dd91 66fd83ec a94f5465 73745665 63746f72
签名第一部分 r	842a2532 b34134b5 d58aec3c 6f59740c 4d7e13a0
签名第二部分 s	8dd81109 707daa15 df465d58 008073fe 573f4ca2

国家图书馆 专用

F.4 ECPV 算例

注：在 F.5.1 和 F.5.2 里，

——Hash 使用专用的杂凑函数 3(SHA1)，

——KDF(x)定义为 MGF2(x，18)，SHA1 是杂凑函数，

——对称密码 Sym 使用异或(⊕)运算。

F.4.1 素数域 GF(p)上的椭圆曲线

p	ffffffff ffffffff ffffffff fffffffe ffffac73
椭圆曲线方程	$y^2 \equiv x^3 + ax + b \pmod{p}$
a	00000000 00000000 00000000 00000000 00000000
b	00000000 00000000 00000000 00000000 00000007
椭圆曲线上点的个数	01 00000000 00000000 0001b8fa 16dfab9a ca16b6b3
G 的 x 坐标	3b4c382c e37aa192 a4019e76 3036f4f5 dd4d7ebb
G 的 y 坐标	938cf935 318fdced 6bc28286 531733c3 f03c4fee
n	01 00000000 00000000 0001b8fa 16dfab9a ca16b6b3

n 的长度（比特个数）	161 比特
$L(n)$	21
签名密钥 x_A	00 e6a080e0 b2a7a850 ba71d26c 9606669a 4b4a6c18
Y_A 的 x 坐标	8f5788a5 c97ac053 984045f4 c9ff325d d60065ae
Y_A 的 y 坐标	a5329d2a 721b5787 9c215323 37211f64 23e577da
M	Test User 1
	13 0b546573 74205573 65722031
M_{rec}	（M_{rec} 是可打印值 “Test User 1”的 DER 编码）
L_{rec}	13
	fa 2b0cbe77
M_{clr}	（M_{clr} 是随机 nonce，没有冗余且是明确的）
L_{red}	5
	05 05050505
$\widetilde{C}_{red}$	（填充冗余 40 比特，全部冗余超过 80 比特）
$d = \widetilde{C}_{red} \parallel M_{rec}$	0505 05050513 0b546573 74205573 65722031
随机数 k	00 d8a0abc5 b7a4029a c232cbcd a16819e1 b715f9f4
$x_0 = R$ 的 x 坐标	1af46ec4 e95daede 056bfa3b 370075f6 3cb2c34f
R 的 y 坐标	c324f1ba 5324383c 371278d5 f3fe4fe2 9373702c
$\Pi = \mathrm{KDF}(x_0)$	20b9 f76f3b33 6a805e02 92ed0b71 c9aaa767
$r = \mathrm{Sym}(d, \Pi) = d \oplus \Pi$	25bc f26a3e20 61d43b71 e6cd5e02 acd88756
$u = \mathrm{Hash}(r \parallel M_{clr})$	1b5361a3 9bc7ca45 6bbb6f2b f80e4e31 a01b4a1b
$s = (k - x_A t) \bmod n$ [其中 $t = \mathrm{OS2IP}(u)$]	735 171 855 371 469 914 412 919 293 672 210 175 623 097 579 946（数字）

F.4.2 扩展域 GF(2^m)上的椭圆曲线

m	163
扩展多项式	$X^{163} + X^7 + X^6 + X^3 + 1$
基本表达	A standard polynomial basis implementation
椭圆曲线方程	$y^2 + xy = x^3 + ax^2 + b$
a	00 00000000 00000000 00000000 00000000 00000001
b	00 00000000 00000000 00000000 00000000 00000001
椭圆曲线上点的个数	08 00000000 00000000 00040211 45C1981b 33f14bde
G 的 x 坐标	02 fe13c053 7bbc11ac aa07d793 de4e6d5e 5c94eee8
G 的 y 坐标	02 89070fb0 5d38ff58 321f2e80 0536d538 ccdaa3d9
n	04 00000000 00000000 00020108 a2e0cc0d 99f8a5ef
n 的长度（比特个数）	163 比特
$L(n)$	21
x_A	03 a41434aa 99c2ef40 c8495b2e d9739cb2 155a1e0d
Y_A 的 x 坐标	03 7d529fa3 7e42195f 10111127 ffb2bb38 644806bc
Y_A 的 y 坐标	04 47026eee 8b34157f 3eb51be5 185d2be0 249ed776
M	Test User 1
	13 0b546573 74205573 65722031
M_{rec}	（M_{rec} 是可打印值 “Test User 1”的 DER 编码）
L_{rec}	13
	fa 2b0cbe77
M_{clr}	（M_{clr} 是随机 nonce，没有冗余且是明确的）
L_{red}	5

国家图书馆 提供

$\tilde{C}_{red}$	05 05050505 (填充冗余 40 比特，全部冗余超过 80 比特)
$d = \tilde{C}_{red} \| M_{rec}$	0505 05050513 0b546573 74205573 65722031
随机数 k	a40b301c c315c257 d51d4422 34f5aff8 189d2b6c
$x_0 = R$ 的 x 坐标	04 994d2c41 aa30e529 52b0a94e c6511328 c502da9b
R 的 y 坐标	03 1fc936d7 3163b858 bbc5326d 77c19839 46405264
$\Pi = \mathrm{KDF}(x_0)$	d5bd 8bd7309d 020d5946 1367e723 a3b63d79
$r = \mathrm{Sym}(d, \Pi) = d \oplus \Pi$	d0b8 8ed2358e 09593c35 6747b250 c6c41d48
$u = \mathrm{Hash}(r \| M_{clr})$	db9b5ebe 6b7b9690 54f9aecb 52b54a19 df4495ae
$s = (k - x_A t) \bmod n$ [其中 $t = \mathrm{OS2IP}(u)$]	2 748 261 816 569 194 283 991 299 038 913 308 940 368 752 275 658(数字)

F.5 ECKNR 算例

注 1：这是标准多项式基本实现。

注 2：MGF2 使用 RIPEMD-160 作为杂凑函数。

注 3：生成数据输入的方法参见附录 D，使用 RIPEMD-160 作为杂凑函数。

F.5.1 素数域上的椭圆曲线

p	ffffffff ffffffff 0a13a2a0 a085053b 49a92b05
椭圆曲线方程	$y^2 \equiv x^3 + ax + b \pmod{p}$
a	ffffffff ffffffff 0a13a2a0 a085053b 49a92b02
b	809dc828 d7ec47f1 d1b20800 62a9d350 c3e7b230
G 的 x 坐标	ae316775 14f76709 513c8442 4165d440 0bb7d699
G 的 y 坐标	71fa37b7 27cbd843 c800d474 1448267a 8fdd047e
n	ffffffff ffffffff 0a15341c 63139e6c 9e868967
n 的长度(比特个数)	160
$L(n)$	20
L_{dat}	20
L_{red}	10
L_{rec}	10
L_{clr}	13
签名密钥 x_A	d648bcb2 e4d5d151 656c8477 4ed016ba 292a5a38
Y_A 的 x 坐标	e3daf394 34a4b15c 633a76de d8ada3de 0a70bbe1
Y_A 的 y 坐标	2e32d49c 1b14fdc3 efa071d7 e864cf13 4b8d55af
Cert_A	e3daf394 34a4b15c 633a76de d8ada3de 0a70bbe1 2e32d49c 1b14fdc3 efa071d7 e864cf13 4b8d55af e3daf394 34a4b15c
z_A	633a76de d8ada3de 0a70bbe1 2e32d49c 1b14fdc3 efa071d7 e864cf13 4b8d55af 00000000 00000000 00000000 00000000 00000000 00000000
随机数 k	887ac572 8a839081 8b535fcb f04e827b 0f8b543c
kG 的 x 坐标	0dc9bb8b 272b418d 2a2516da b5642d2d 41321865
kG 的 y 坐标	8540597c 06c9bd33 7f151eb4 982f7c76 23d75fc5
Π	7ecdd6b7 aed42dd7 76e33cb4 43687001 39e2572f
签名消息	This is a test message!
M	546869 73206973 20612074 65737420 6d657373 61676521
M_{rec}	5468 69732069 73206120
M_{clr}	74 65737420 6D657373 61676521

国家图书馆 版权所有

	000000
数据（杂凑输入）	00000000 0a000000 00000000 0d546869 73206973 20612074 65737420
	6d657373 61676521 7ecdd6b7 aed42dd7 76e33cb4 43687001 39e2572f
截取的杂凑权标 h	5b72 1142ea6e 7011bbe6
数据输入 $d = h \| M_{rec}$	5b721142 ea6e7011 bbe65468 69732069 73206120
MGF($z_A \| M_{clr}$)	b0324c93 8ca97523 b925df2c b2be0374 6e834238
签名第一部分 r	958d8b66 c81328e5 7420b7f0 98a5531c 24417437
签名第二部分 s	a1f82e55 b662e65b 579fe513 1c76d17b 1a71470a

F.5.2 扩展域 GF(2^m)上的椭圆曲线

m	163
不可约多项式	$X^{163} + X^8 + X^2 + X + 1$
椭圆曲线方程	$y^2 + xy = x^3 + ax^2 + b$
a	07 2546b543 5234a422 e0789675 f432c894 35de5242
b	00 c9517d06 d5240d3c ff38c74b 20b6cd4d 6f9dd4d9
G 的 x 坐标	07 af699895 46103d79 329fcc3d 74880f33 bbe803cb
G 的 y 坐标	01 ec23211b 5966adea 1d3f87f7 ea5848ae f0b7ca9f
n	04 00000000 00000000 0001e60f c8821cc7 4daeafc1
n 的长度（比特个数）	163
$L(n)$	21
L_{dat}	21
L_{red}	10
L_{rec}	11
L_{clr}	12
x_A	03 d648bcb2 e4d5d151 656c8477 4ed016ba 292a5a38
Y_A 的 x 坐标	05 85bc0f8e b4b5adf5 79dcf2c3 2c8144dc 45a38a97
Y_A 的 y 坐标	04 82de581e 443872d5 6b40700d f41033f8 4ce2d205
	0585 bc0f8eb4 b5adf579 dcf2c32c
$Cert_A$	8144dc45 a38a9704 82de581e 443872d5 6b40700d f41033f8 4ce2d205
	0585bc0f 8eb4b5ad
z_A	f579dcf2 c32c8144 dc45a38a 970482de 581e4438 72d56b40 700df410
	33f84ce2 d2050000 00000000 00000000 00000000 00000000 00000000
随机数 k	02 887ac572 8a839081 8b535fcb f04e827b 0f8b543c
kG 的 x 坐标	06 a2338166 2db382e2 66d7b836 e3f469f5 5247ebe8
kG 的 y 坐标	05 cac246d2 ff718d5f 7c5668d0 2794a7ab 7dce7210
Π	b8 4379c2b5 f49ec40c 72cb2d2d 1c79e1b5 06bf029c
签名消息	This is a test message!
M	546869 73206973 20612074 65737420 6d657373 61676521
M_{rec}	546869 73206973 20612074
M_{clr}	65737420 6d657373 61676521
	00000000
数据（杂凑输入）	0000000b 00000000 0000000c 54686973 20697320 61207465 7374206d
	65737361 676521b8 4379c2b5 f49ec40c 72cb2d2d 1c79e1b5 06bf029c
截取的杂凑权标 h	C333 e25a3d52 354300e9
数据输入 $d = h \| M_{rec}$	c3 33e25a3d 52354300 e9546869 73206973 20612074
MGF($z_A \| M_{clr}$)	1a fa16273d fd0f4d72 3b0dc96e cfaf0cc9 f13c2f28
签名第一部分 r	61 8a8dbfb5 5ba4ca7e a0928c2a a0f6840f d7e20dc0
签名第二部分 s	02 11b0ded9 8d24aa2b 8e66d489 930229d8 072dc2e8

国家图书馆 专用

F.5.3 扩展域 GF(p^m)上的椭圆曲线

p	fffffffb
m	5
不可约多项式	$X^5 - 2$
椭圆曲线方程	$y^2 \equiv x^3 + ax + b \pmod{p}$
a	00000000 00000000 00000000 00000000 00000001
b	00000000 00000000 00000000 00000001 00000106
G 的 x 坐标	fcdee3ee eb6a9d0c 821c8b46 d27937bc 0fbac840
G 的 y 坐标	3c329e0d 7a5fb6e4 048a69c1 12f8cb35 dffb7ccc
n	ffffffe7 000000f9 fffe3308 f697c1d6 d7de35cf
n 的长度（比特个数）	160
$L(n)$	20
L_{dat}	20
L_{red}	10
L_{rec}	10
L_{clr}	13
x_A	d648bcb2 e4d5d151 656c8477 4ed016ba 292a5a38
Y_A 的 x 坐标	1d8ea2e9 91232cfa ef256c2d 800710f3 1c2b57a6
Y_A 的 y 坐标	65655866 4add4c90 b6abf35f d0c9e635 acf20dc9
$Cert_A$	1d8ea2e7 41fe72cf bfa93df4
	f348d571 41be5813 6565585e 5ef264cf bd1641fc 92f72c58 6e29c1bd
	ここでまちがっています
	1d8ea2e7 41fe72cf
z_A	bfa93df4 f348d571 41be5813 6565585e 5ef264cf bd1641fc 92f72c58
	6e29c1bd 00000000 00000000 00000000 00000000 00000000 00000000
随机数 k	887ac572 8a839081 8b535fcb f04e827b 0f8b543c
kG 的 x 坐标	fec4d254 e5ac0b25 30d7482d 7f746bc3 fa3a775a
kG 的 y 坐标	3f6c5e14 ee723450 26a2750f 2e55d9e0 a4ec3e8e
Π	a17fccf4 ce99bab0 a1f76735 410f66bf 6accb85f
签名消息	This is a test message!
M	546869 73206973 20612074 65737420 6d657373 61676521
M_{rec}	5468 69732069 73206120
M_{clr}	74 65737420 6d657373 61676521
	000000
数据（杂凑输入）	00000000 0a000000 00000000 0d546869 73206973 20612074 65737420
	6d657373 61676521 a17fccf4 ce99bab0 a1f76735 410f66bf 6accb85f
截取的杂凑权标 h	cf53 683c2f5c 861badbf
数据输入 $d = h \| M_{rec}$	cf53683c 2f5c861b adbf5468 69732069 73206120
$MGF(z_A \| M_{clr})$	2eb2a254 1c2b20fc 9d58acb5 59ffd1fc 87207871
签名第一部分 r	409e069c fdee1c57 91109fe8 7183972a 9ecca10e
签名第二部分 s	70599c96 8034ad47 b88793c2 68f8347b d24128d5

国家图书馆 专用

附 录 G
（资料性附录）
机制特点总结

本附录总结了五个带消息恢复的数字签名方案 NR、ECNR、ECMR、ECPV 和 ECKNR 的特点。表 G.1 列出了域参数和用户密钥。表 G.2 和表 G.3 列出了生成签名和校验签名所需的操作。

ECPV 包括处理数据输入的操作。其他方案不包括处理数据输入的操作。

杂凑函数或者对称密码比逆运算代价更大。但是，对于整个计算，杂凑计算占的比重微乎其微。

表 G.1 五个机制的总结（域参数和用户密钥）

NR	ECNR	ECMR	ECPV	ECKNR
F，G，n x_A，Y_A P，Q L_{dat}	E，F，G，n x_A，Y_A P，Q L_{dat}	E，F，G，n x_A，Y_A P，Q Hash（或 MGF）	E，F，G，n x_A，Y_A P，Q L_{red}，（L_{rec} 或 L_{clr}） Sym，L_{key} Hash KDF	E，F，G，n x_A，Y_A P，Q MGF

表 G.2 五个机制的总结（签名生成过程）

运算	机制				
	NR	ECNR	ECMR	ECPV	ECKNR
模 n 加法运算	2	2	3	1	1
模 n 乘法运算	1	1	2	1	1
模 n 逆运算	0	0	1	0	0
椭圆曲线的点乘或有限域的幂运算	1	1	1	1	0
按位异或运算	0	0	1	0	2
杂凑	1	1	1（或 0）	1	2
MGF（或 KDF）	0	0	0（或 1）	1	0
对称密钥	0	0	0	1	0
I2OSP	1	1	0	1	0
OS2IP（模 n）	1	1	1	1	1
$EC2OSP_E$ 或 $FE2OSP_F$	1	1	1	1	1

国家图书馆专用

表 G.3　五个机制的总结(签名验证过程)

运算	机制				
	NR	ECNR	ECMR	ECPV	ECKNR
模 n 加法运算	1	1	2	0	0
模 n 乘法运算	0	0	2	0	0
模 n 逆运算	0	0	1	0	0
椭圆曲线的加法或有限域的点乘运算	1	1	1	1	1
椭圆曲线的点乘或有限域的幂运算	2	2	2	2	2
按位异或运算	0	0	1	0	2
杂凑	1	1	1(或 0)	1	2
MGF(或 KDF)	0	0	0(或 1)	1	0
对称密钥	0	0	0	1	0
I2OSP	1	1	0	1	0
OS2IP(模 n)	2	2	1	1	1
$EC2OSP_E$ 或 $FE2OSP_F$	1	1	1	1	1

国家图书馆 专用

附 录 H
（资料性附录）
方案的对应性

本附录描述了本部分定义的机制与 ISO/IEC 9796-3:2000 以及 ISO/IEC 15946-4:2004 所定义机制的对应关系。

本部分	ISO/IEC 9796-3:2000	ISO/IEC 15946-4:2004
NR(第 8 章)	NR(第 9 章)	—
ECNR(第 9 章)	—	ECNR(第 7 章)
ECMR(第 10 章)	—	ECMR(第 8 章)
ECPV(第 11 章)	—	ECPV(第 10 章)
ECKNR(第 12 章)	—	ECKNR(第 11 章)

国家图书馆 专用

参 考 文 献

[1] ISO/IEC 9796-3:2000 Information technology—Security techniques—Digital signature schemes giving message recovery—Part 3: Discrete logarithm based mechanisms

[2] ISO/IEC 14888-1 Information technology—Security techniques—Digital signatures with appendix—Part 1: General

[3] ISO/IEC 15946-4:2004 Information technology—Security techniques—Cryptographic techniques based on elliptic curves—Part 4: Digital signatures giving message recovery

[4] M. ABE and T. OKAMOTO, "A signature scheme with message recovery as secure as discrete logarithm," Advances in Cryptology—Asiacrypt'99, Lecture Notes in Computer Science 1716, pp. 378-389, Springer—Verlag, 1999.

[5] ANSI X9.62—1999 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1999.

[6] ANSI X9.63—1999 Public Key Cryptography For The Financial Services Industry: Elliptic Curve KeyAgreement and Transport Protocols, 1999.

[7] IEEE Std 1363—2000 IEEE Standard Specifications for Public-Key Cryptography.

[8] C. H. LIM and P. J. LEE, "A study on the proposed Korean digital signature algorithm," Advances in Cryptology—Asiacrypt'98, Lecture Notes in Computer Science 1514, pp 175-186, Springer—Verlag, 1998.

[9] A. J. MENEZES, P. C. VAN OORSCHOT and S. A. VANSTONE, "Handbook of applied cryptography," CRC Press, 1997.

[10] A. MIYAJI, "Another Countermeasure to Forgeries over Message Recovery Signature," IEICE Trans.,Fundamentals, vol. E80-A, No.11, pp 2192-2200, 1997.

[11] K. NYBERG and R. A. RUEPPEL, "Message recovery for signature schemes based on the discrete.

logarithm problem," Designs, Codes and Cryptography, 7, pp 61-81, 1996.

[12] L. PINTSOV and S. VANSTONE, "Postal Revenue Collection in the Digital Age," Proceedings of the Fourth International Financial Cryptography Conference, 2000.

[13] D. H. YUM, S. G. SIM and P. J. LEE, "New Signature Schemes Giving Message Recovery Based on EC-KCDSA," Proceedings of the 12th Conference on Information Security and Cryptology (CISC), 2002.

国家图书馆专用

国家图书馆 专用

中 华 人 民 共 和 国
国 家 标 准
信息技术 安全技术 带消息恢复的
数字签名方案
第 3 部分:基于离散对数的机制
GB/T 15851.3—2018
*
中 国 标 准 出 版 社 出 版 发 行
北京市朝阳区和平里西街甲 2 号(100029)
北京市西城区三里河北街 16 号(100045)
网址:www.spc.org.cn
服务热线:400-168-0010
2019 年 1 月第一版
*
书号:155066·1-62111

版权专有 侵权必究

GB/T 15851.3-2018