ICS 35.030
CCS L 80
备案号：

中 华 人 民 共 和 国 密 码 行 业 标 准

GM/T XXXX—XXXX

基于 SM2 算法的无证书及隐式证书公钥机制

Certificateless and implicit-certificate-based public key mechanisms based on the SM2 algorithms

（送审稿）

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

XXXX-XX-XX 发布　　XXXX-XX-XX 实施

国家密码管理局　发 布

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分:标准化文件的结构和起草规则》的规定起草。

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专利的责任。

本文件由密码行业标准化技术委员会提出并归口。

本文件起草单位：深圳奥联信息安全技术有限公司、兴唐通信科技有限公司、国汽（北京）智能网联汽车研究院有限公司、国家工业信息安全发展研究中心、中汽数据（天津）有限公司、中国电力科学研究院有限公司、北京信安世纪科技股份有限公司、中国石油勘探开发研究院、北京汽车研究总院有限公司、北京数字认证股份有限公司、北京国脉信安科技有限公司、国家电网能源互联网技术研究院、中国测绘科学研究院、中国科学院信息工程研究所、北京邮电大学、中国大唐集团科学技术研究总院有限公司、大唐微电子技术有限公司、大唐移动通信设备有限公司、大唐高鸿数据网络技术股份有限公司、格尔软件股份有限公司、大唐电信科技股份有限公司、中国信息通信研究院、中国移动研究院、中安网脉（北京）技术股份有限公司、广州汽车集团股份有限公司汽车工程研究院、中汽研软件测评（天津）有限公司、联通智网科技股份有限公司、上汽通用五菱汽车股份有限公司、握奇数据股份有限公司、三未信安科技股份有限公司、中国联通中讯邮电咨询设计院有限公司、青岛国创智能家电研究院有限公司、北京江南天安科技有限公司、博雅中科（北京）信息技术有限公司、安徽问天量子科技股份有限公司。

本文件主要起草人：程朝辉、万兆泽、刘建行、陈雪鸿、赵万里、翟峰、汪宗斌、冯梅、王冲华、袁峰、李志虎、马照亭、刘奇旭、谭儒、徐国爱、张永强、郑强、李峰、郑丽娟、王妮娜、张金池、陈中林、周光涛、巩军、邵学彬、邓宇、徐晖、沈天珺、于润东、田野、李增欣、但波、熊开新、张渊、车业蒙、王首媛、王亮、金添、郭忠泉、浦雨三、李雪雁、刘会议。

引　言

Al-Riyami 和 Paterson 在 2003 年提出无证书公钥密码（Certificateless-Public Key Cryptography）。无证书公钥系统不依赖数字证书验证用户的标识和公钥绑定关系的真实性并且密钥生成中心不具有密钥委托功能。扩展后的无证书公钥密码模型和安全定义允许基于标准公钥密码算法构造无证书公钥机制。

隐式证书（Implicit Certificate）不包括证书签发机构的签名且证书处理者需要根据证书中的数据计算得出证书拥有者的公钥。

本文件描述基于 SM2 算法构造的无证书公钥机制和隐式证书公钥机制。

基于SM2算法的无证书及隐式证书公钥机制

1 范围

本文件规定基于SM2算法的无证书及隐式证书公钥机制，包括密钥生成与校验机制、数字签名机制、公钥加密机制。

本文件规定的机制适用于带宽和计算资源受限的应用环境,其中数字签名机制适用于商用密码应用中的数字签名和验证，加密机制适用于商用密码应用中的消息加解密。

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。

GB/T 32918.1 信息安全技术 SM2椭圆曲线公钥密码算法 第1部分:总则

GB/T 32918.2 信息安全技术 SM2椭圆曲线公钥密码算法 第2部分:数字签名算法

GB/T 32918.4 信息安全技术 SM2椭圆曲线公钥密码算法 第4部分:公钥加密算法

GB/T 32918.5 信息安全技术 SM2椭圆曲线公钥密码算法 第5部分:参数定义

GB/T 32905 信息安全技术 SM3密码杂凑算法

GB/T 32915 信息安全技术 二元序列随机性检测方法

GM/T 0090 标识密码应用标识格式规范

3 术语和定义

GB/T 32918.1、GB/T 32918.2、GB/T 32918.4界定的以及下列术语和定义适用本文件。

3.1

密钥生成中心 key generation center；KGC

负责选择椭圆曲线系统参数、生成主密钥并产生用户部分私钥和声明公钥的可信机构。

3.2

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、电话号码等。标识可进一步包括其他辅助信息，如标识用途、标识有效期等。

注：该标识等同于GB/T 32918.2定义的可辨别标识。

3.3

主密钥 master key

处于无证书密码系统密钥分层结构最顶层的密钥，包括系统主私钥和系统主公钥，其中系统主公钥公开，系统主私钥由KGC秘密保存。

3.4

用户的声明公钥 user's claimed public key

用于与椭圆曲线系统参数、系统主公钥、用户的标识一起计算用户实际公钥的公开值。

注：用户的声明公钥在隐式证书中也称为公钥还原数据。

3.5

用户的实际公钥　user's real public key

由椭圆曲线系统、系统主公钥、用户的标识和用户的声明公钥一起计算得出的公钥值。

注：用户实际公钥和用户私钥在指定的椭圆曲线系统参数下是一一对应的。

3.6

证书认证机构　certificate authority；CA

对数字证书进行全生命周期管理的实体，也称为电子认证服务机构。

3.7

隐式证书　implicit certificate

包含用户标识、公钥还原数据和签发者标识等信息但不显式包含证书认证机构（CA）的数字签名的数字证书。

注：用户关联的实际公钥可由椭圆曲线系统参数、CA的公钥、隐式证书中相关的用户信息和公钥还原数据计算得出。

3.8

公钥还原数据　public key reconstruction data

包含在隐式证书中用于计算还原用户实际公钥的数据。

4　符号和缩略语

4.1　符号

下列符号适用于本文件。

A,B：使用无证书公钥密码系统的两个用户。

C ：公钥加密的密文。

d_A'：用户A的部分私钥。

d_A ：用户A的私钥。

$E(F_q)$：F_q上椭圆曲线E的所有有理点(包括无穷远点O)组成的集合。

e：密码杂凑算法作用于消息M以及相关前缀数据的输出值。

F_q：包含q个元素的有限域。

G：椭圆曲线的一个基点，其阶为素数。

H_v( ) ：消息摘要长度为v比特的密码杂凑算法。

H_A：关于用户A的标识、部分椭圆曲线系统参数和系统主公钥的杂凑值。

IC_A：用户A的隐式证书，其至少包括用户A的标识、公钥还原数据和隐式证书签发者标识。

ID_A：用户A的标识。

KDF：密钥派生函数。

ks：KGC秘密，用于派生伪随机数。

ms：系统主私钥。

M：待签名消息或待加密消息。

M'：待验证消息。

mod n：模n运算。例如，23 mod 7=2。

n：基点G的阶(n是$\#E(F_q)$的素因子)。

O：椭圆曲线上的一个特殊点，称为无穷远点或零点，是椭圆曲线加法群的单位元。

P_{pub}：系统主公钥。

param：椭圆曲线系统参数。

P_A：用户A的实际公钥。

W_A：用户A的声明公钥。

q：有限域F_q中元素的数目。

t_A：KGC为用户A生成的部分私钥。

U_A：用户A生成的部分公钥。

a, b：F_q中的元素，它们定义F_q上的一条椭圆曲线E。

$x \| y$：x与y的拼接，其中x、y可以是比特串或字节串。

x_G, y_G：点G的x轴值和y轴值。

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

(r, s)：发送的签名。

(r', s')：收到的签名。

$[k]P$：椭圆曲线上点P的k倍点，即，$[k]P = \underbrace{P+P+...+P}_{K个}$，$k$是正整数。

ENC$(param, M, P_A)$：公钥加密算法，其使用椭圆曲线系统参数*param*、公钥P_A对消息M进行加密。

DEC$(param, C, d_A)$：公钥解密算法，其使用椭圆曲线系统参数*param*、私钥d_A解密密文C。

SIGN$(param, Z_A, M, d_A)$：数字签名生成算法，其使用椭圆曲线系统参数*param*、杂凑值Z_A、私钥d_A对消息M进行签名并输出(r, s)。

VERIFY$(param, Z_A, P_A, M', (r', s'))$：数字签名验证算法，其使用椭圆曲线系统参数*param*、杂凑值Z_A、公钥P_A对消息M'的签名(r', s')进行验证并输出签名正确性。

4.2 缩略语

下列缩略语适用于本文件。

CA:证书认证机构（Certificate Authority）

COER:正则八位字节编码规则（Canonical Octet Encoding Rules）

GHR:全球 Handle 注册机构（Global Handle Registry）

KGC:密钥生成中心（Key Generation Center）

LHS:本地 Handle 服务（Local Handle Service）

5 机制参数和辅助函数

5.1 概述

本文件规定的公钥机制包括密钥生成与校验机制、数字签名机制和公钥加密机制。其中密钥生成与校验机制包括主密钥的生成机制以及用户密钥对的生成与校验机制和流程。主密钥由KGC生成，包括系统主私钥和系统主公钥。用户的私钥和声明公钥由KGC和用户协同生成。用户公开其用户标识和声明公钥，其实际公钥由椭圆曲线系统参数、系统主公钥、用户标识和用户的声明公钥根据本文件规定的方法计算生成。

本文件规定的密钥生成机制是一种可用于生成隐式证书应用中所需密钥数据的机制。密钥数据包括CA的密钥对、用户公钥还原数据和私钥。校验机制可用于校验隐式证书中用户公钥还原数据和私钥的正确性。隐式证书用于分发用户标识、公钥还原数据等。

无证书以及基于隐式证书的数字签名机制和公钥加密机制分别基于标准的基础数字签名算法（*SIGN*和*VERIFY*）和基础公钥加密算法（*ENC*和*DEC*）构造。本文件规定基础数字签名算法和基础公钥加密算法

为GB/T 32918规定的SM2椭圆曲线公钥密码算法。本文件规定的数字签名机制可实现对消息的数字签名和验证。本文件规定的公钥加密机制可实现对消息的加解密。规定机制的数据示例见附录A。

无证书公钥机制中的用户标识可采用GM/T 0090规定的格式。隐式证书的格式和编码方式以及隐式证书内容与本文件中用户标识的对应关系不在本文件的范围内。隐式证书公钥机制和无证书公钥机制的应用参考示例分别见附录B和附录C。

5.2 椭圆曲线系统参数

椭圆曲线系统参数包括有限域F_q的规模q（当$q=2^m$时，还包括元素表示法的标识和约化多项式）；定义椭圆曲线$E(F_q)$的方程的两个元素a、$b\in F_q$、$E(F_q)$上的基点$G=(x_G, y_G)$($G\neq O$)、G的阶n及其他可选项（如n的余因子等）。

椭圆曲线系统参数及其验证应符合GB/T 32918.1-2016第5章及GB/T 32918.5-2017的规定。

5.3 辅助函数

5.3.1 概述

本文件规定的无证书公钥机制中涉及以下辅助函数:密码杂凑算法、密钥派生函数、随机数发生器，使用以下基础算法：数字签名生成算法*SIGN*和数字签名验证算法*VERIFY*、公钥加密算法*ENC*和公钥解密算法*DEC*。

5.3.2 密码杂凑算法

本文件规定使用的密码杂凑算法遵循GB/T 32905 SM3密码杂凑算法。

5.3.3 密钥派生函数

本文件规定使用的密钥派生函数遵循GB/T 32918.4-2016第5章规定的密钥派生函数定义。

5.3.4 随机数发生器

本文件规定使用的随机数发生器产生的随机数应满足GB/T 32915要求。

5.3.5 数字签名生成和验证算法

本文件规定使用的数字签名生成算法*SIGN*和数字签名验证算法*VERIFY*分别遵循GB/T 32918.2-2016第6章数字签名的生成算法及流程和第7章数字签名的验证算法及流程。

5.3.6 公钥加密和解密算法

本文件规定使用的公钥加密算法*ENC*和公钥解密算法*DEC*分别遵循GB/T 32918.4-2016第6章加密算法及流程和第7章解密算法及流程。

5.4 用户标识信息

用户A具有长度为$entlen_A$比特的标识ID_A，记$ENTL_A$是由整数$entlen_A$转换而成的两个字节。在本文件规定的密钥生成机制的KGC、数字签名机制的签名者和验证者以及公钥加密机制中的发送者都需要用密码杂凑算法求得用户A的杂凑值H_A。按GB/T 32918.1-2016中4.2.6和4.2.5给出的方法，将椭圆曲线方程参数a、b、G 的坐标 x_G、y_G和P_{pub}的坐标x_{Pub}、y_{Pub}的数据类型转换为比特串，计算$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$。

6 密钥生成机制及流程

6.1 主密钥生成机制

KGC用随机数发生器产生随机数$ms \in [1, n-1]$作为系统主私钥。计算系统主公钥$P_{pub}=[ms]G$。

注：在隐式证书系统中，ms和P_{pub}分别是CA的签名私钥和签名公钥。

6.2 用户密钥对的生成机制

用户A和KGC一起协同生成用户的密钥对:用户私钥d_A和声明公钥W_A。两者应实现以下运算步骤:

A1：用户A用随机数发生器产生随机数$d_A' \in [1, n-1]$；

A2：用户A计算$U_A=[d_A']G$，并将标识ID_A和U_A提交KGC；

K1：KGC计算$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$；

K2：KGC用随机数发生器产生随机数$w \in [1, n-1]$；

K3：KGC计算$W_A=[w]G+U_A$；

K4：KGC按GB/T 32918.1-2016中4.2.6和4.2.5给出的方法将W_A的坐标x_{WA}、y_{WA}的数据类型转换为比特串，计算$\lambda=H_{256}(x_{WA} \| y_{WA} \| H_A) \bmod n$，按 GB/T 32918.1-2016中4.2.4和4.2.3给出的方法将λ的数据类型转换为整数；

K5：KGC计算$t_A=(w+\lambda*ms) \bmod n$，并将$t_A$和$W_A$安全地返回给用户A；

A3：用户A计算$d_A=(t_A+d_A') \bmod n$；

A4：如果$0<d_A<n-1$，则输出(d_A, W_A)；否则返回A1。

生成加密密钥对时，KGC可使用确定性方法生成t_A和W_A。生成方法见附录D。

KGC将t_A返回用户A时可使用U_A作为公钥使用加密方法ENC加密包括t_A的数据后将密文传递到用户A。用户A使用d_A'解密密文后还原包括t_A的数据。

注：在隐式证书系统中，步骤K3生成的W_A为隐式证书中的公钥还原数据，步骤A3生成用户私钥。

6.3 用户密钥对生成机制流程

用户密钥对生成机制流程见图1。

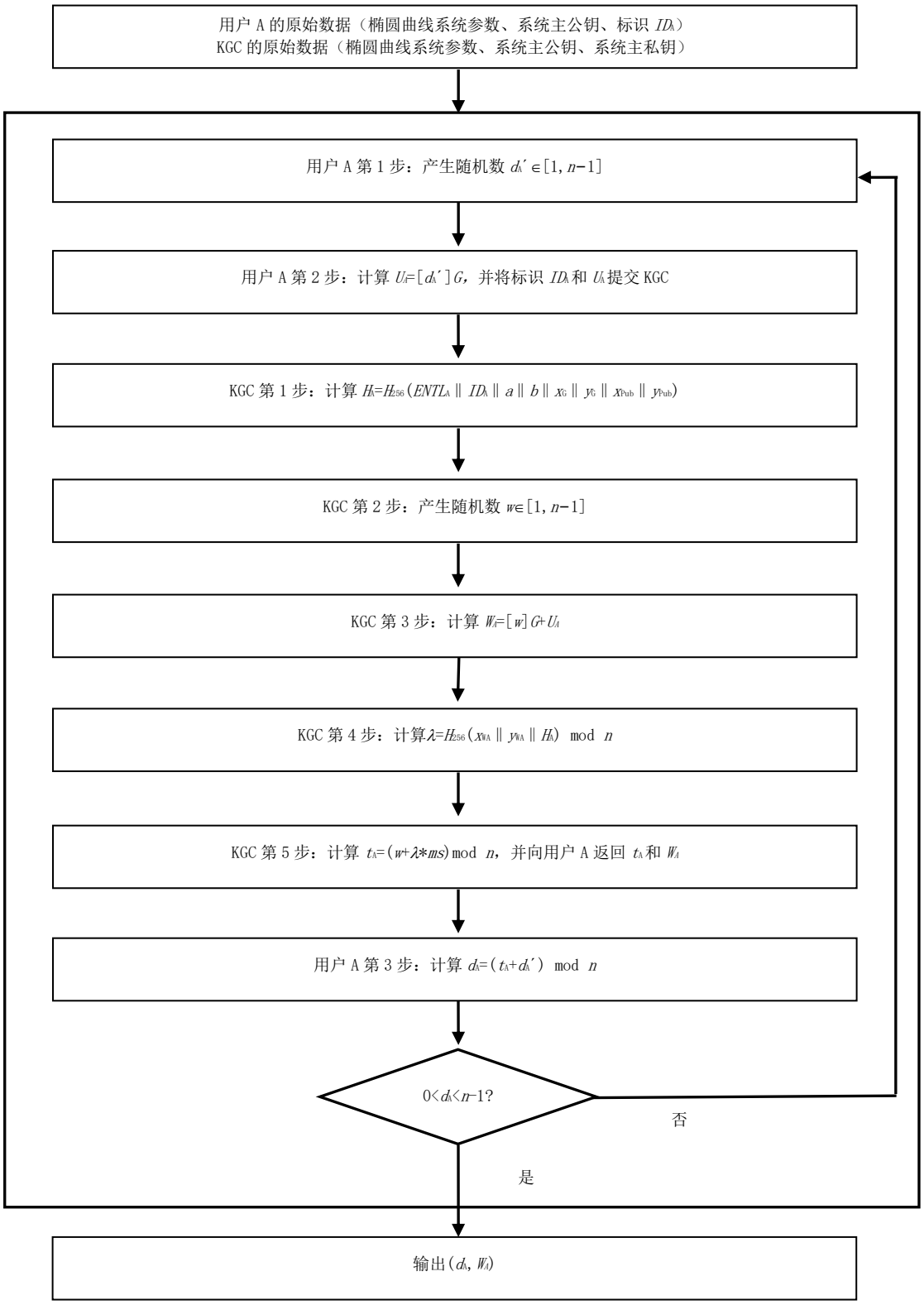

图 1　用户密钥对生成机制流程

6.4　用户密钥对的校验机制

为了验证生成的密钥对(d_A, W_A)的正确性，用户A应实现以下运算步骤:

A1：计算$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$；

A2：按GB/T 32918.1-2016中4.2.6和4.2.5给出的方法将W_A的坐标x_{WA}、y_{WA}的数据类型转换为比特串，计算$\lambda=H_{256}(x_{WA} \| y_{WA} \| H_A) \bmod n$，按 GB/T 32918.1-2016中4.2.4和4.2.3给出的方法将λ的数据类型转换为整数；

A3：计算$P_A=W_A+[\lambda]P_{pub}$；

A4：计算$P_A'=[d_A]G$;

A5：检查$P_A=P_A'$是否成立，若成立则验证通过；否则验证不通过。

注：步骤A3生成的P_A为用户的实际公钥。

6.5 用户密钥对校验机制流程

用户密钥对校验机制流程见图2。

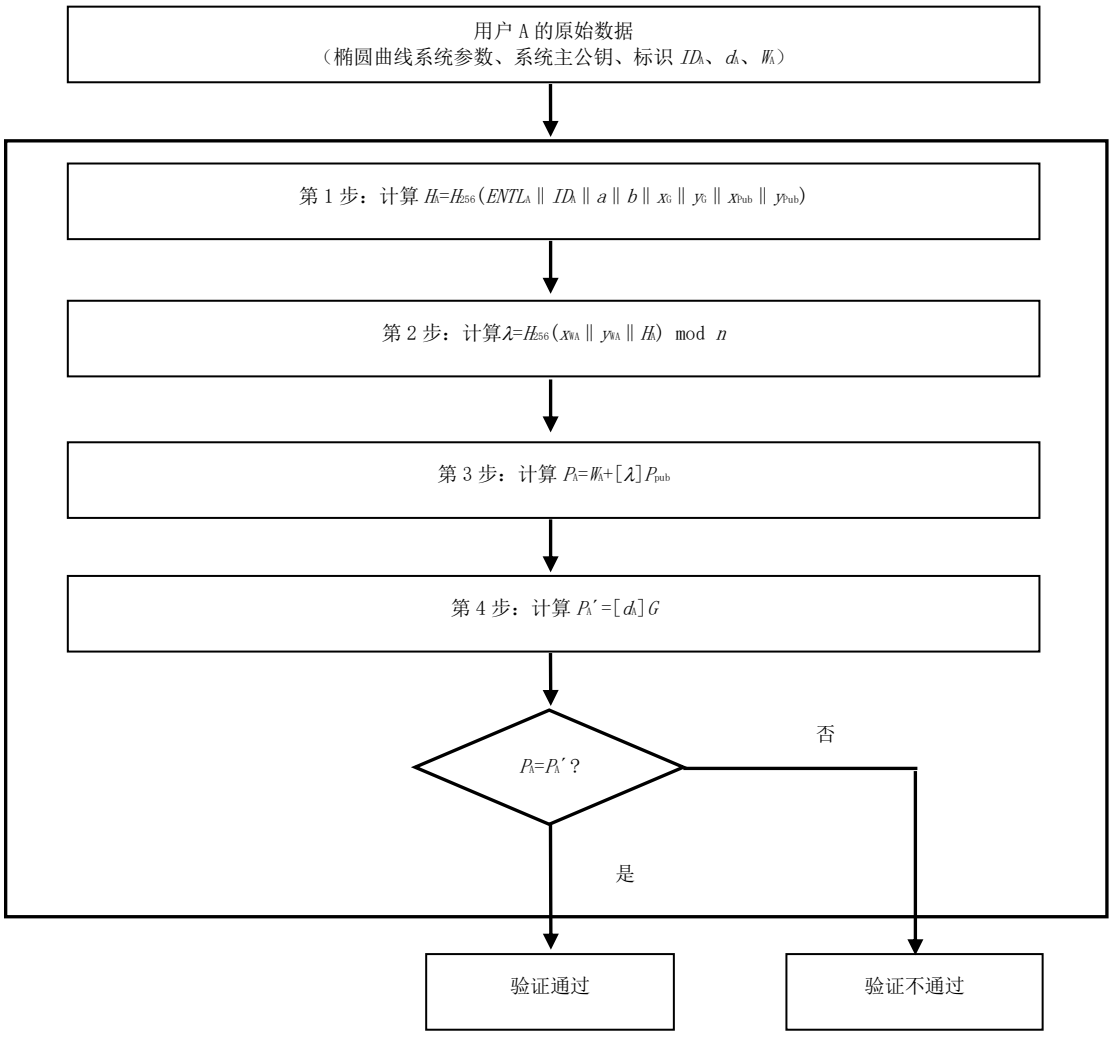

图 2 用户密钥对校验机制流程

7 数字签名机制

7.1 数字签名的生成机制

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：在无证书系统中，按GB/T 32918.1-2016中4.2.6和4.2.5给出的方法将W_A的坐标x_{WA}、y_{WA}的数据类型转换为比特串，执行$SIGN(param, H_A, x_{WA} \| y_{WA} \| M, d_A)$并输出签名$(r, s)$。在隐式证书系统中，执行$SIGN(param, ZE, IC_A \| M, d_A)$并输出签名$(r, s)$，其中$ZE$为空串。

7.2 数字签名的验证机制

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：计算$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$；

B2：按GB/T 32918.1-2016中4.2.6和4.2.5给出的方法将W_A的坐标x_{WA}、y_{WA}的数据类型转换为比特串，计算$\lambda=H_{256}(x_{WA} \| y_{WA} \| H_A) \bmod n$，按GB/T 32918.1-2016中4.2.4和4.2.3给出的方法将λ的数据类型转换为整数；

B3：计算$P_A=W_A+[\lambda]P_{pub}$；

B4：在无证书系统中，执行$VERIFY(param, H_A, x_{WA} \| y_{WA} \| M', P_A, (r', s'))$并输出结果。在隐式证书系统中，执行$VERIFY(param, ZE, IC_A \| M', P_A, (r', s'))$并输出结果，其中$ZE$为空串。

注：SM2数字签名验证机制涉及对公钥P_A的点乘计算。若机制实现允许，步骤B3中P_A的计算可推迟到数字签名验证阶段并采用快速方法计算多个点乘之和。

8 公钥加密机制

8.1 加密机制

设需要发送的消息为比特串M，为了对明文M加密并将密文发送给解密者用户A，作为加密者的用户B应实现以下运算步骤：

B1：计算$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$；

B2：按GB/T 32918.1-2016中4.2.6和4.2.5给出的方法将W_A的坐标x_{WA}、y_{WA}的数据类型转换为比特串，计算$\lambda=H_{256}(x_{WA} \| y_{WA} \| H_A) \bmod n$，按GB/T 32918.1-2016中4.2.4和4.2.3给出的方法将λ的数据类型转换为整数；

B3：计算$P_A=W_A+[\lambda]P_{pub}$；

B4：执行$ENC(param, M, P_A)$并输出结果。

注：SM2加密机制涉及对公钥P_A的点乘计算。若机制实现允许，步骤B3中P_A的计算可推迟到加密阶段并采用快速方法计算多个点乘之和。

8.2 解密机制

为了对密文C进行解密，作为解密者的用户A应实现以下运算步骤：

A1：执行$DEC(param, C, d_A)$并输出结果。

附 录 A
（资料性）
机制数据示例

A.1 概述

本附录选用GB/T 32905给出的密码杂凑算法。
本附录选用GB/T 32918.5给出的SM2算法参数。
本附录中，所有用16进制表示的数，左边为高位，右边为低位。
本附录中，字符串采用GB/T 1988编码，消息和明文采用16进制表示。

A.2 无证书系统密钥生成机制数据示例

椭圆曲线系统参数(*param*)：
q: FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFF
a: FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFC
b: 28E9FA9E 9D9F5E34 4D5A9E4B CF6509A7 F39789F5 15AB8F92 DDBCBD41 4D940E93
n: FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF 7203DF6B 21C6052B 53BBF409 39D54123
G=(x_G, y_G)：
x_G: 32C4AE2C 1F198119 5F990446 6A39C994 8FE30BBF F2660BE1 715A4589 334C74C7
y_G: BC3736A2 F4F6779C 59BDCEE3 6B692153 D0A9877C C62A4740 02DF32E5 2139F0A0
主密钥：
ms: 6BDD93B2 10F79415 FE0F6388 C1C932C2 08319FF7 D7E99C97 2B3535C9 F19A9FF9
P_{pub}=[ms]G=(x_{Pub}, y_{Pub})：
x_{Pub}: F294B710 601DE1C5 D34420C4 902D81C3 A3064490 3E5799BF E4013E56 C55C864C
y_{Pub}: 5C003EB1 B50B9BBB 2E880778 2AA3C38E A800E23B 30B777FA AD8F0F71 46D66AC1
用户密钥对生成过程数据：
d_A': 04914C20 251A59A2 C3111029 44C60043 0A02285A 04331442 28142A18 48004C00
U_A=[d_A']G=(x_U, y_U)：
x_U: DE0ED7FD 0B7D7144 76318954 E5FBBFBF D73A67A8 D65090FA 6955725C 10C14015
y_U: 3340A21A 74CBD4B0 D902AA5E 780F45F9 D79E8C94 C035CFF6 6357F928 8CD02C27
ID_A: Alice
H_A=H_{256}($ENTL_A$ ‖ ID_A ‖ a ‖ b ‖ x_G ‖ y_G ‖ x_{Pub} ‖ y_{Pub})：
CC2DC292 53D9C773 85781813 985D421A 6C6F9CCB CE61FAD3 7EC79D8B 6EFC8F71
w: 6CB28D99 385C175C 94F94E93 4817663F C176D925 DD72B727 260DBAAE 1FB2F96F
W_A=[w]G+U_A=(x_{WA}, y_{WA})：
x_{WA}: DB65BF80 F08E3FEA 9758A949 0F2C257A 2D8ADEAA 59DA786C BFAFEF22 1E78ADB4
y_{WA}: 0728185A 257F64B7 9DFA929C 16C987ED 956FB32D 00B6CAF7 678E56E6 6E01530F
λ=H_{256}(x_{WA} ‖ y_{WA} ‖ H_A) mod n:
F84BBC96 6F668583 D734971C 2F4DE1CC 8A620CD6 E4D09488 E1FBE205 2EFDC9E0
t_A: BBB6EBEB 993D1EE3 E5F4C265 107D3AF7 C094169A 0E7DDD0A 7400638B E271614F

$d_A=(t_A+d_A')\bmod n$:

C048380B BE577886 A905D28E 55433B3A CA963EF4 12B0F14C 9C148DA4 2A71AD4F

用户公钥：

$P_A=W_A+[\lambda]P_{pub}=(x_P, y_P)$：

x_P: 12B72E93 6C902048 F99D77F4 C556D112 F72FDF6A 7BA24DE1 06BC5B73 00C4B5C4

y_P: 8C5AF329 D45B9E62 70D07645 F10B283C 35D8BF9A 35F58AB4 1E8CD4A1 DD70D18B

A.3 无证书系统数字签名机制数据示例

M: 6D657373 61676520 64696765 7374

ID_A: Alice

$W_A=(x_{WA}, y_{WA})$：

x_{WA}: DB65BF80 F08E3FEA 9758A949 0F2C257A 2D8ADEAA 59DA786C BFAFEF22 1E78ADB4

y_{WA}: 0728185A 257F64B7 9DFA929C 16C987ED 956FB32D 00B6CAF7 678E56E6 6E01530F

无证书签名过程示例：

$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$：

CC2DC292 53D9C773 85781813 985D421A 6C6F9CCB CE61FAD3 7EC79D8B 6EFC8F71

$SIGN(param, H_A, x_{WA} \| y_{WA} \| M, 0, d_A)$即 SM2 数字签名生成过程示例：

d_A: C048380B BE577886 A905D28E 55433B3A CA963EF4 12B0F14C 9C148DA4 2A71AD4F

$e=H_{256}(H_A \| x_{WA} \| y_{WA} \| M)$：

F303E917 EAAD2604 7E24E9D7 FB9ADABA A72DC0D2 8A6D217B 06596DBE 344B39F1

k: 34914C20 251A59A2 C3111029 44C60043 0A02285A 04331442 28142A18 48004C14

$[k]G=(x_1, y_1)$：

x_1: 80C8022B 011044A1 392310CD 3B7E722E 444B68D8 B20E948A 8517E5BD 8BB92531

y_1: 1214F7E5 C4D4121D E8845B5D 81F138A2 5707C760 0FC63A62 D2F007C0 56FC0804

(r, s)：

73CBEB43 EBBD6AA5 B747FAA5 37194CE9 79754A40 1AB5B0DA 37B55F72 862F1DFF

F358CDF5 527F157D 2B7DCB74 A82DBA83 60529016 4EBAA046 A584FAB0 D4CEAFB0

无证书验签过程示例：

$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$：

CC2DC292 53D9C773 85781813 985D421A 6C6F9CCB CE61FAD3 7EC79D8B 6EFC8F71

$\lambda=H_{256}(x_{WA} \| y_{WA} \| H_A) \bmod n$:

F84BBC96 6F668583 D734971C 2F4DE1CC 8A620CD6 E4D09488 E1FBE205 2EFDC9E0

$P_A=W_A+[\lambda]P_{pub}=(x_P, y_P)$：

x_P: 12B72E93 6C902048 F99D77F4 C556D112 F72FDF6A 7BA24DE1 06BC5B73 00C4B5C4

y_P: 8C5AF329 D45B9E62 70D07645 F10B283C 35D8BF9A 35F58AB4 1E8CD4A1 DD70D18B

$VERIFY(param, H_A, x_{WA} \| y_{WA} \| M, P_A, (r', s'))$即 SM2 数字签名验证过程示例:

$e'=H_{256}(H_A \| x_{WA} \| y_{WA} \| M)$：

F303E917 EAAD2604 7E24E9D7 FB9ADABA A72DC0D2 8A6D217B 06596DBE 344B39F1

$t=(r'+s')\bmod n$:

6724B93A 3E3C8022 E2C5C619 DF47076D 67C3FAEB 47AA4BF5 897E661A 21288C8C

$[s']G+[t]P_A=(x_1', y_1')$：

x_1': 80C8022B 011044A1 392310CD 3B7E722E 444B68D8 B20E948A 8517E5BD 8BB92531

y_1': 1214F7E5 C4D4121D E8845B5D 81F138A2 5707C760 0FC63A62 D2F007C0 56FC0804

$R=(e'+x_1') \bmod n$

73CBEB43 EBBD6AA5 B747FAA5 37194CE9 79754A40 1AB5B0DA 37B55F72 862F1DFF

A.4 无证书系统公钥加密机制数据示例

M: 01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101
01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101

ID_A: Alice

无证书加密过程示例：

$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$:

CC2DC292 53D9C773 85781813 985D421A 6C6F9CCB CE61FAD3 7EC79D8B 6EFC8F71

$W_A=(x_{WA}, y_{WA})$:

x_{WA}: DB65BF80 F08E3FEA 9758A949 0F2C257A 2D8ADEAA 59DA786C BFAFEF22 1E78ADB4

y_{WA}: 0728185A 257F64B7 9DFA929C 16C987ED 956FB32D 00B6CAF7 678E56E6 6E01530F

$\lambda=H_{256}(x_{WA} \| y_{WA} \| H_A) \bmod n$:

F84BBC96 6F668583 D734971C 2F4DE1CC 8A620CD6 E4D09488 E1FBE205 2EFDC9E0

$P_A=W_A+[\lambda]P_{pub}=(x_P, y_P)$:

x_P: 12B72E93 6C902048 F99D77F4 C556D112 F72FDF6A 7BA24DE1 06BC5B73 00C4B5C4

y_P: 8C5AF329 D45B9E62 70D07645 F10B283C 35D8BF9A 35F58AB4 1E8CD4A1 DD70D18B

$ENC(param, M, P_A)$即 SM2 加密过程示例：

k: 4C62EEFD 6ECFC2B9 5B92FD6C 3D957514 8AFA1742 5546D490 18E5388D 49DD7B4F

$C_1=[k]G=(x_1, y_1)$:

x_1: 11C88AE0 4CEC1BA5 54D03D5B 5970333A 83585826 C2A985DE 5520D9E9 34389EFB

y_1: 84B52D34 4FB21AA8 EA38A494 0C833269 2B8D4DA2 39354921 2EAFDC0F 11CA5C9C

$[k]P_A=(x_2, y_2)$:

x_2: 2629AB80 DA9141F5 7C3B2BAC B37D7EFF 609AE3D7 03DCD97C B8BFA531 8FFCFA9A

y_2: 39A685A9 3BF11374 C5823959 38A5B40A 50381F94 E7A23BF0 921B4EE2 4C5E5A8F

$C=(C_1, C_3, C_2)$:

11C88AE0 4CEC1BA5 54D03D5B 5970333A 83585826 C2A985DE 5520D9E9 34389EFB
84B52D34 4FB21AA8 EA38A494 0C833269 2B8D4DA2 39354921 2EAFDC0F 11CA5C9C
0705DE4A D1F60968 DE489936 BFD3BE7E B4855084 BF08E739 780701FA B236AC2D
4765BAD3 7BAEDEF1 6992224F 40B3A6C2 F216B481 9ABFC69F C6BFDD65 ED4D0E6F
BC722FA3 2672DA83 0E10BAE8 4E06DA0E C4AC1852 66E49D6B CBC62615 779C8ABE

无证书解密过程示例：

$DEC(param, C, d_A)$即 SM2 解密过程。

A.5 隐式证书系统密钥生成机制及隐式证书数据示例

椭圆曲线系统参数($param$)：

q: FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFF

a: FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFC

b: 28E9FA9E 9D9F5E34 4D5A9E4B CF6509A7 F39789F5 15AB8F92 DDBCBD41 4D940E93

n: FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF 7203DF6B 21C6052B 53BBF409 39D54123

$G=(x_G, y_G)$:

x_G: 32C4AE2C 1F198119 5F990446 6A39C994 8FE30BBF F2660BE1 715A4589 334C74C7

y_G: BC3736A2 F4F6779C 59BDCEE3 6B692153 D0A9877C C62A4740 02DF32E5 2139F0A0

主密钥：

ms: 6BDD93B2 10F79415 FE0F6388 C1C932C2 08319FF7 D7E99C97 2B3535C9 F19A9FF9

$P_{pub}=[ms]G=(x_{Pub}, y_{Pub})$:

x_{Pub}: F294B710 601DE1C5 D34420C4 902D81C3 A3064490 3E5799BF E4013E56 C55C864C

y_{Pub}: 5C003EB1 B50B9BBB 2E880778 2AA3C38E A800E23B 30B777FA AD8F0F71 46D66AC1

用户密钥对生成过程数据：

d_A': 04914C20 251A59A2 C3111029 44C60043 0A02285A 04331442 28142A18 48004C00

$U_A=[d_A']G=(x_U, y_U)$:

x_U: DE0ED7FD 0B7D7144 76318954 E5FBBFBF D73A67A8 D65090FA 6955725C 10C14015

y_U: 3340A21A 74CBD4B0 D902AA5E 780F45F9 D79E8C94 C035CFF6 6357F928 8CD02C27

按照附录B.3.2中隐式证书数据和标识数据对应关系获得ID_A:：

8000000A 08080808 08080808 08090909 00020023 73008400 C8

$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$:

2138FC2C BF4B591D 156757D9 98A245BA EAD4FFA8 5586342E FFDA6D5F 33BE9384

w: 6CB28D99 385C175C 94F94E93 4817663F C176D925 DD72B727 260DBAAE 1FB2F96F

$W_A=[w]G+U_A=(x_{WA}, y_{WA})$:

x_{WA}: DB65BF80 F08E3FEA 9758A949 0F2C257A 2D8ADEAA 59DA786C BFAFEF22 1E78ADB4

y_{WA}: 0728185A 257F64B7 9DFA929C 16C987ED 956FB32D 00B6CAF7 678E56E6 6E01530F

$\lambda=H_{256}(x_{WA} \| y_{WA} \| H_A) \mod n$:

D343E4E6 679DBDBF BFED3F7E 9EF2491D EDF1FC14 D6372A5A 2530C731 846D8F21

t_A: 3F4F6FDD DACBC8A8 09CA5BFF C11E19E4 B7A8E467 3DC1690F 442EB9E8 B2ED3900

$d_A=(t_A+d_A') \mod n$:

43E0BBFD FFE6224A CCDB6C29 05E41A27 C1AB0CC1 41F47D51 6C42E400 FAED8500

用户公钥：

$P_A=W_A+[\lambda]P_{pub}=(x_P, y_P)$:

x_P: 5E6203B3 E5B8B4C6 FDA394C9 ED0940C4 143D7AEC C933C129 A3B5CB1A 6A950D1B

y_P: A0CFE288 A04D33DD 6BA589DA 3AB397F7 5A6ED2CD FD9108AB 29B29CD4 0A624BD5

按照附录 B 中隐式证书的格式编码后形成 IC_A：

00030183 08212223 24252627 28008000 000a0808 08080808 08080809 09090002

00237300 8400C881 2183DB65 BF80F08E 3FEA9758 A9490F2C 257A2D8A DEAA59DA

786CBFAF EF221E78 ADB4

A.6 隐式证书系统数字签名机制数据示例

M: 6D657373 61676520 64696765 7374

IC_A:

00030183 08212223 24252627 28008000 000a0808 08080808 08080809 09090002

00237300 8400C881 2183DB65 BF80F08E 3FEA9758 A9490F2C 257A2D8A DEAA59DA

786CBFAF EF221E78 ADB4

按照附录 B.3.2 中隐式证书数据和标识数据对应关系获得 ID_A:

8000000a 08080808 08080808 08090909 00020023 73008400 C8

基于隐式证书 IC_A 解压缩获得的 $W_A=(x_{WA}, y_{WA})$:

x_{WA}: DB65BF80 F08E3FEA 9758A949 0F2C257A 2D8ADEAA 59DA786C BFAFEF22 1E78ADB4

y_{WA}: 0728185A 257F64B7 9DFA929C 16C987ED 956FB32D 00B6CAF7 678E56E6 6E01530F

基于隐式证书的签名过程示例:

$SIGN(param, ZE, IC_A \| M, 0, d_A)$即 SM2 数字签名生成过程示例:

d_A: 43E0BBFD FFE6224A CCDB6C29 05E41A27 C1AB0CC1 41F47D51 6C42E400 FAED8500

$e=H_{256}(IC_A \| M)$:

CF308D39 12BFD344 44F7EDB5 379160E7 4D4CEF12 159C974 9C16A391 CE3946A8B

k: 34914C20 251A59A2 C3111029 44C60043 0A02285A 04331442 28142A18 48004C14

$[k]G=(x_1, y_1)$:

x_1: 80C8022B 011044A1 392310CD 3B7E722E 444B68D8 B20E948A 8517E5BD 8BB92531

y_1: 1214F7E5 C4D4121D E8845B5D 81F138A2 5707C760 0FC63A62 D2F007C0 56FC0804

(r, s):

4FF88F65 13D017E 57E1AFE8 2730FD31 61F94787F A5E526A8 F2C62AD1 35784E99

BF6CCA55 B7AB45D 033577A2 AF860DB5 F0F0D6C9F 12957F54 5B2BC02A EDCF903A

基于隐式证书的验签过程示例:

$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$:

2138FC2C BF4B591D 156757D9 98A245BA EAD4FFA8 5586342E FFDA6D5F 33BE9384

$\lambda=H_{256}(x_{WA} \| y_{WA} \| H_A) \mod n$:

D343E4E6 679DBDBF BFED3F7E 9EF2491D EDF1FC14 D6372A5A 2530C731 846D8F21

$P_A=W_A+[\lambda]P_{pub}=(x_P, y_P)$:

x_P: 5E6203B3 E5B8B4C6 FDA394C9 ED0940C4 143D7AEC C933C129 A3B5CB1A 6A950D1B

y_P: A0CFE288 A04D33DD 6BA589DA 3AB397F7 5A6ED2CD FD9108AB 29B29CD4 0A624BD5

$VERIFY(param, ZE, IC_A \| M, P_A, (r', s'))$即 SM2 数字签名验证过程示例:

$e'=H_{256}(IC_A \| M)$:

CF308D39 12BFD344 44F7EDB5 379160E7 4D4CEF12 159C974 9C16A391 CE3946A8B

$t=(r'+s') \mod n$:

0F6559BB CB7B5DB5 B17278AD 6B70AE75 BC9E05B3 96B4A0D1 FA35F6F2 E9729DB0

$[s']G+[t]P_A=(x_1', y_1')$:

x_1': 80C8022B 011044A1 392310CD 3B7E722E 444B68D8 B20E948A 8517E5BD 8BB92531

y_1': 1214F7E5 C4D4121D E8845B5D 81F138A2 5707C760 0FC63A62 D2F007C0 56FC0804

$R=(e'+x_1') \mod n$

4FF88F65 13D017E 57E1AFE8 2730FD31 61F94787F A5E526A8 F2C62AD1 35784E99

A.7 隐式证书系统公钥加密机制数据示例

M: 01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101

01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101

IC_A:

00030183 08212223 24252627 28008000 000a0808 08080808 08080809 09090002

00237300 8400C881 2183DB65 BF80F08E 3FEA9758 A9490F2C 257A2D8A DEAA59DA

786CBFAF EF221E78 ADB4

按照附录 B.3.2 中隐式证书数据和标识数据对应关系获得 ID_A:

8000000A 08080808 08080808 08090909 00020023 73008400 C8

基于隐式证书 IC_A 解压缩获得的 $W_A=(x_{WA}, y_{WA})$:

x_{WA}: DB65BF80 F08E3FEA 9758A949 0F2C257A 2D8ADEAA 59DA786C BFAFEF22 1E78ADB4

y_{WA}: 0728185A 257F64B7 9DFA929C 16C987ED 956FB32D 00B6CAF7 678E56E6 6E01530F

基于隐式证书的加密过程示例：

$H_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_{Pub} \| y_{Pub})$:

2138FC2C BF4B591D 156757D9 98A245BA EAD4FFA8 5586342E FFDA6D5F 33BE9384

$\lambda=H_{256}(x_{WA} \| y_{WA} \| H_A) \mod n$:

D343E4E6 679DBDBF BFED3F7E 9EF2491D EDF1FC14 D6372A5A 2530C731 846D8F21

$P_A=W_A+[\lambda]P_{pub}=(x_P, y_P)$:

x_P: 5E6203B3 E5B8B4C6 FDA394C9 ED0940C4 143D7AEC C933C129 A3B5CB1A 6A950D1B

y_P: A0CFE288 A04D33DD 6BA589DA 3AB397F7 5A6ED2CD FD9108AB 29B29CD4 0A624BD5

$ENC(param, M, P_A)$即 SM2 加密过程示例：

k: 4C62EEFD 6ECFC2B9 5B92FD6C 3D957514 8AFA1742 5546D490 18E5388D 49DD7B4F

$C_1=[k]G=(x_1, y_1)$:

x_1: 11C88AE0 4CEC1BA5 54D03D5B 5970333A 83585826 C2A985DE 5520D9E9 34389EFB

y_1: 84B52D34 4FB21AA8 EA38A494 0C833269 2B8D4DA2 39354921 2EAFDC0F 11CA5C9C

$[k]P_A=(x_2, y_2)$:

x_2: 91527074 7E888E0D DA97ED6A A91CF8CA FB9357AC A160C6FC A2768E48 267EEDE7

y_2: 75697A80 A3FC9DCB 711E37B4 AD624745 5CD31CC2 AA3C83BB B54A503D 0138DB05

$C=(C_1, C_3, C_2)$:

11C88AE0 4CEC1BA5 54D03D5B 5970333A 83585826 C2A985DE 5520D9E9 34389EFB
84B52D34 4FB21AA8 EA38A494 0C833269 2B8D4DA2 39354921 2EAFDC0F 11CA5C9C
553B13CD 4FC7153B DC6B3B84 C0E7495A AF085070 B34C2F7B DE80DBF6 0BA636E9
1424DDB9 650AC2FB 0BAE46A2 6878F169 26F6DAC6 F8071EFB 12573166 47FDB0CE
40CC55AC BA4CF7C1 809768B3 7FF07E0B 854B9065 5903E21D BE9FC757 C17692EB

基于隐式证书的解密过程示例：

$DEC(param, C, d_A)$即 SM2 解密过程。

附 录 B
（资料性）
机制在隐式证书应用中的应用示例

B.1 概述

本附录描述使用本文件中规范的密钥机制生成密钥对和隐式证书以及使用数字签名机制保护消息的一个示例。本附录中的内容仅为参考，具体的应用规范由相关标准另行规定。

B.2 一种隐式证书格式

一种可能的隐式证书格式如[7]中的定义：

```
CertificateBase ::= SEQUENCE  {
    version                 Uint8(3),
    type                    CertificateType,
    issuer                  IssuerIdentifier,
    toBeSigned              ToBeSignedCertificate,
    signature               Signature OPTIONAL
}
```

该数据结构包含：

——version：是证书格式的版本号，此字段设置为“3”；

——type：指示证书是显式的或隐式的还是其他的。对于隐式证书，使用本文件规定的机制时此字段设置为 1；

——issuer：证书的签发方；

——toBeSigned: 证书的内容；

——signature：包含在显式证书中。它由 Issuer 字段中标识的签名者在 toBeSigned 的散列上计算的签名。

```
ToBeSignedCertificate ::= SEQUENCE  {
    id                      CertificateId,
    cracaId                 HashedId3,
    crlSeries               CrlSeries,
    validityPeriod          ValidityPeriod,
    region                  GeographicRegion OPTIONAL,
    assuranceLevel          SubjectAssurance OPTIONAL,
    appPermissions          SequenceOfAidSsp OPTIONAL,
    certIssuePermissions    SequenceOfAidGroupPermissions OPTIONAL,
    certRequestPermissions  SequenceOfAidGroupPermissions OPTIONAL,
    canRequestRollover      NULL OPTIONAL,
    encryptionKey           PublicEncryptionKey OPTIONAL,
    verifyKeyIndicator      VerificationKeyIndicator,
    ...
```

```
}
(WITH COMPONENTS { ..., appPermissions PRESENT} |
 WITH COMPONENTS { ..., certIssuePermissions PRESENT} |
 WITH COMPONENTS { ..., certRequestPermissions PRESENT})
```

ToBeSignedCertificate 结构中的字段具有以下含义：

——id 包含在必要时识别证书持有者的信息；

——cracaId 用于识别负责发布证书撤销列表的证书撤销授权 CA，它的证书可能出现在证书撤销列表上；

——crlSeries 表示与证书可能出现的特定证书撤销授权 CA 相关的证书撤销列表系列；

——validityPeriod 包含证书的有效期；

——region 表示证书的有效区域；

——assuranceLevel 表示证书持有者的保证级别；

——appPermissions 表示证书持有者使用此证书签署应用程序数据的权限；

——certIssuePermissions 表示证书持有者使用此证书签署其他证书的权限；certRequestPermissions 表示证书持有者使用此证书签署证书请求的权限；

——canRequestRollover 表示该证书可用于签署请求具有相同权限的另一个证书的请求消息；

——encryptionKey 包含用于加密的公钥，证书持有者持有相应的私钥；

——verifyKeyIndicator 包含可用于恢复公钥的公钥还原数据。

具体的数据类型定义见[7]。证书数据采用正则八位字节编码规则(COER)编码。

B.3 隐式证书及密钥生成过程

B.3.1 CA 的初始化

CA 按照 6.1 规定的主密钥生成机制执行如下过程，其和标准 CA 的系统初始化过程一致。

a) CA 生成系统私钥 ms，系统公钥 $P_{pub}=[ms]G$；

b) CA 将$[ms]G$作为其签名公钥向其根 CA 申请一张 CA 证书。

步骤 b)中证书申请过程为标准证书申请过程，生成的证书为根 CA 为该 CA 签发的 CA 证书。若无根 CA，则该 CA 可生成自签名的 CA 证书。

B.3.2 隐式证书数据与规范中用户标识信息 ID_A 的对应关系

本文件中的 ID_A 至少包括 ToBeSignedCertificate 中从 id 域开始到 encryptionKey 域之前的数据域。若 encryptionKey 域存在且是普通公钥，则 ID_A 包括该数据域，否则 ID_A 不包括该数据域。ID_A 所包括的数据域的编码格式与 ToBeSignedCertificate 的编码格式一致。ID_A 所包括的数据域的顺序与 ToBeSignedCertificate 中对应域的顺序一致。

B.3.3 隐式证书的申请和生成

实体按照 6.2 规定的用户密钥对的生成机制，执行如下过程申请隐式证书：

a) 实体执行 6.2 中的步骤 A1 和 A2，产生随机数 $d_A' \in [1, n-1]$，计算 $U_A=[d_A']G$；

b) 实体生成隐式证书申请：包括公钥数据 U_A 等信息，提交隐式证书申请到 CA；

c) CA 验证隐式证书申请的合法性后，生成隐式证书中除密钥 verifyKeyIndicator 的其他部分，即从 id 开始到 encryptionKey 的部分。如果 encryptionKey 域存在且是普通公钥，则该密钥按照普通密钥生成方法生成。若 encryptionKey 域存在且是隐式证书类型的加密公钥，则密钥

需要按照本文件 6.2 规定的生成机制生成。CA 按照 B.3.2 的方法构造 ID_A，执行 6.2 中的步骤 K1 到 K5，获得 t_A 和 W_A；

d) CA 将 W_A 作为隐式证书中数据域 verifyKeyIndicator 的数据进行编码，生成完整的隐式证书；

e) 如果实体在证书申请过程中生成了加密公私密钥对(Q, c)，并在证书申请中提供加密公钥 Q，则使用公钥 Q 加密 t_A 和隐式证书形成密文 C 后返回给实体。如果实体未提供加密公钥 Q，则使用 U_A 作为加密公钥执行加密运算，将加密结果返回实体；

f) 实体使用解密私钥 c（如果 $Q=U_A$，则 $c=d_A'$）解密 e)中返回的密文 C，获得 t_A 和隐式证书；

g) 实体执行 6.2 中的步骤 A3 和 A4 计算完整签名私钥 d_A 并从隐式证书的数据域 verifyKeyIndicator 获得声明公钥 W_A；

h) 实体按照 6.4 规定的步骤验证用户密钥对的合法性，进而验证隐式证书的合法性。

B.4 使用隐式证书和私钥签名消息数据的过程

在需要对消息数据进行签名时，实体遵循 7.1 规定在隐式证书系统中的数字签名过程执行数据签名计算。

B.5 使用隐式证书验证消息签名的过程

实体验签消息签名的基本过程如下：

a) 按照 B.3.2 的方法根据隐式证书内容形成 ID_A，从 verifyKeyIndicator 提取公钥还原数据作为 W_A，从 CA 的根证书中提取公钥$[ms]G$ 作为 P_{pub}；

b) 根据 7.2 规定在隐式证书应用中的验证签名过程验证数字签名的有效性。

附 录 C
（资料性）
机制在工业互联网标识解析系统中的应用示例

C.1 概述

本附录描述无证书公钥机制的一种应用示例。示例描述在工业互联网标识解析系统中使用本文件中规范的密钥机制生成密钥对以及使用数字签名机制进行实体认证、保护标识解析数据安全的一种可能的方式。本附录中的内容仅为参考，具体的应用规范由相关标准另行规定。

C.2 工业互联网标识解析系统及相关密钥

工业互联网Handle标识解析系统[4]管理Handle命名空间中的Handle，可用于工业互联网的标识解析。Handle系统提供两类服务：Handle解析服务和Handle管理服务。Handle解析过程包括两个主要过程：1）查询全局Handle注册机构（GHR），解析Handle中的命名权威机构的服务信息。2）根据过程1的结果查询本地Handle服务（LHS），解析Handle值。Handle管理服务用于创建、更新、删除Handle的值记录。Handle系统至少包括3种组件：

a) 全局 Handle 注册机构（GHR）：以命名权威机构的 Handle 方式管理和解析本地 Handle 服务的服务信息;
b) 本地 Handle 服务（LHS）：以 Handle 方式管理和解析其负责命名空间下的 Handle;
c) 客户端：查询或管理 Handle 以及 Handle 的值记录。

按照访问权限的不同，Handle系统中Handle值记录可分为两大类：公开访问Handle值记录与受控访问Handle值记录。公开访问Handle值记录没有数据机密性要求，但可具有数据完整性和数据源真实性的安全要求。例如命名权威机构Handle值记录包括服务站点列表信息。该信息数据的完整性和数据源真实性对Handle系统的安全至关重要。对于查询或者管理需要额外授权的受控访问Handle值记录时，需要认证客户端的身份并根据访问控制权限要求进行授权。这类受控访问Handle值记录可能具有敏感性，操作过程可能还需要保护数据的机密性和完整性。

Handle系统至少有以下三类密钥：

a) GHR 的公私密钥对：用于为命名权威机构 Handle 解析提供数据完整性和数据源真实性;
b) LHS 的公私密钥对：用于为本地 Handle 解析提供数据完整性和数据源真实性;
c) Handle 的管理密钥：用于访问受控访问 Handle 时的身份认证，安全会话建立过程的身份认证、会话密钥传递等。

GHR的公私密钥对由GHR的管理机构生成，通过Handle“0.NA/0.NA”的<HS_SITE>类型Handle值方式对外发布。LHS的公私密钥对由LHS的管理机构生成。LHS的公钥通过Handle“0.NA/LHS N.A.”(如“0.NA/10”)的<HS_SITE>类型Handle值方式对外发布，其格式与GHR的<HS_SITE>中公钥相同。Handle客户端查询“0.NA/10”的<HS_SITE>Handle值时可要求GHR对查询响应进行数字签名。客户端收到查询结果后，使用GHR的公钥验证GHR的签名；若签名有效，则接受LHS的公钥。

在访问受控的Handle时需要进行身份认证。Handle系统使用<HS_ADMIN>类型Handle值定义某个Handle的管理权限以及进行身份认证使用的密钥Handle引用。<HS_ADMIN>类型Handle值中的数据域<AdminRef>指向进行客户端认证的Handle管理密钥的Handle和索引。

C.3 全局 Handle 注册机构使用无证书数字签名机制的方法

GHR的公私密钥对可采用本文件第6节规定的密钥生成机制及流程生成。生成过程采用GHR的Handle“0.NA/0.NA”作为ID_A。按照[5]的规定，生成的声明公钥W_A可通过“0.NA/0.NA”Handle的<HS_SITE>记录中<PublicKeyRecoard>数据域发布。[4]中规定<PublicKeyRecord>域的格式为：<PublicKeyRecord>=4字节整数|UTF8编码的密钥类型|2字节保留|公钥数据字节数组。采用本文件生成的密钥时，UTF8编码的密钥类型值为UTF-8编码的算法标识“ECS-SM2”，公钥数据字节数组为W_A按GB/T 35276规定的椭圆曲线点编码方式编码的结果值。

GHR需要对Handle系统中的数据执行数字签名操作时按照本文件第7节中规定的机制生成数字签名。验签方通过查询“0.NA/0.NA”的<HS_SITE>记录中的<PublicKeyRecord>数据域获得W_A,使用“0.NA/0.NA”作为ID_A,按照本文件第7节规定的机制验证GHR生成的数字签名的有效性。

C.4 本地 Handle 服务使用无证书数字签名机制的方法

LHS的公私密钥对可采用本文件第6节规定的密钥生成机制及流程生成。采用LHS的Handle如“0.NA/10”作为ID_A。按照[5]的规定，生成的声明公钥W_A可通过“0.NA/10”Handle的<HS_SITE>记录中的<PublicKeyRecoard>数据域发布，发布方式同C.3。

LHS需要对Handle系统中的数据执行数字签名操作时按照本文件第7节规定的机制生成数字签名。验签方通过查询LHS的Handle如“0.NA/10”的<HS_SITE>记录中的<PublicKeyRecord>数据域获得W_A,使用“0.NA/10”作为ID_A,按照本文件第7节规定的机制验证LHS生成的数字签名的有效性。

C.5 客户端使用无证书数字签名机制的方法

按照[6]的规定，Handle客户端在查询受控访问Handle值记录或者管理Handle及Handle值记录时需要进行身份认证。Handle客户端可采用本文件第6节规定的密钥生成机制及流程生成其密钥对。Handle客户端使用的标识可根据Handle系统的需要进行单独规定，例如：在GHR系统中管理某LHS的Handle时，客户为该LHS的管理者，此时客户端可采用LHS的Handle如“0.NA/10”作为ID_A；在某LHS中，LHS的客户可采用如邮箱地址、Handle管理员账号等其他类型标识作为作为ID_A。

Handle系统使用<HS_ADMIN>类型Handle值定义一个Handle的管理权限以及进行身份认证时使用的密钥Handle引用。<HS_ADMIN>类型Handle值中的<AdminRef>数据域指向进行客户端认证的Handle管理密钥的Handle和索引。采用本文件的机制进行身份认证时，<AdminRef>指向的Handle值记录为标识集合类型记录<HS_ IDSET>。<HS_IDSET>类型Handle值包括数据域：<idset>、<blkset>等。<idset>数据域指定了允许身份认证的标识集合。<idset>域格式如下：<idset>=4字节整数|UTF-8编码的标识列表。4字节整数为标识列表中标识的个数。标识列表包括一个或多个UTF-8编码的标识。标识支持通配符号*或？，其中*可匹配一个或多个字符，？匹配一个字符。例如*.admin@network可匹配alice.admin@network或者bob.admin@network等字符串；admin?@nework则可匹配admin2@network或adminX@network等字符串。<blkset>指定了禁止身份认证的标识集合。<blkset>数据域的格式与<idset>相同。

客户端进行身份认证的具体过程遵循[6]中的规定。身份认证时，LHS首先检查客户端标识是否在<idset>中，如果不在，则拒绝其身份认证；进一步检查客户端标识是否在<blkset>中，如果在，则拒绝其身份认证。身份认证过程涉及数字签名与验证的计算遵循本文件第7节的规定。采用本文件的生成的数字签名的编码可采用如下ASN.1的编码格式:

```
ECS_SIGN ::= SEQUENCE {
    r INTEGER,
```

```
    s INTEGER,
    W OCTET STRING OPTIONAL
}
```

其中r与s是第7节规定的数字签名机制的签名输出结果。W是客户端声明公钥W_A按GB/T 35276规定的椭圆曲线点编码方式编码的结果值。

附 录 D
（资料性）
密钥生成中心用户密钥的确定性生成方法

D.1 概述

密码生成中心需根据用户提供的标识及部分公钥生成确定的部分私钥以及声明公钥时，可采用本附录的方法。

D.2 密钥生成中心生成确定性用户密钥的方法

在 6.1 规定的主密钥生成机制中，密钥生成中心还生成随机选取的长度至少为 256 的比特串 ks。

在 6.2 规定的用户密钥对的生成机制中，步骤 K2 中的随机数 w 可采用如下方式生成，按 GB/T 32918.1-2016 中 4.2.6 和 4.2.5 给出的方法将 U_A 的坐标 x_U、y_U 的数据类型转换为比特串，计算产生 $w=KDF(H_A \| x_U \| y_U \| ks, 8\times\lceil(5\times(\log_2 n))/32\rceil) \bmod n$，其中 KDF 的输出需按 GB/T 32918.1-2016 中 4.2.4 和 4.2.3 给出的方法转换为整数再进行模运算。若 $w=0$，则运算终止。

密钥生成中心生成部分私钥以及声明公钥的其他步骤遵照 6.2 的规定。

参 考 文 献

[1] S. Al-Riyami and K. G. Paterson. Certificateless Public Key Cryptography. In Proc. of Asiacrypt 2003, LNCS 2894, pp. 452－473, 2003

[2] GB/T 1988—1998 信息技术 信息交换用七位编码字符集

[3] GB/T 35276—2017 信息安全技术 SM2密码算法使用规范

[4] RFC 3650 Handle System Overview

[5] RFC 3651 Handle System Namespace and Service Definition

[6] RFC 3652 Handle System Protocol (ver 2.1) Specification

[7] YD/T 3957—2021 基于LTE的车联网无线通信技术 安全证书管理系统技术要求