



華東師範大學
EAST CHINA NORMAL UNIVERSITY

中国商用密码介绍

张磊

密码与网络安全系
软件工程学院
华东师范大学



目录

- 网络空间安全形势与商用密码
- 商用密码介绍
- SM2 椭圆曲线公钥密码算法
- SM9 标识密码算法

网络空间安全形势与商用密码





- 网络空间安全形势与商用密码
- 棱镜计划是一项由美国国家安全局自2007年小布什时期起开始实施的绝密电子监听计划
- 美国情报机构一直在九家美国互联网公司中进行数据挖掘工作，谷歌、雅虎、微软、苹果、Facebook、美国在线、PalTalk、Skype、YouTube等
- 中央国家安全委员会，俗称“中央国安委”、“国安委”，全称为“中国共产党中央国家安全委员会”，是中国共产党中央委员会下属机构
 - 经中国共产党第十八届中央委员会第三次全体会议决定，于2013年11月12日正式成立





- 网络空间安全形势与商用密码

密码在网络空间中的重要作用

1

密码支撑构建网络空间安全防护综合体。

2

密码助力打造网络空间数据共享价值链。

3

密码推进形成网络空间安全协同生态圈。

4

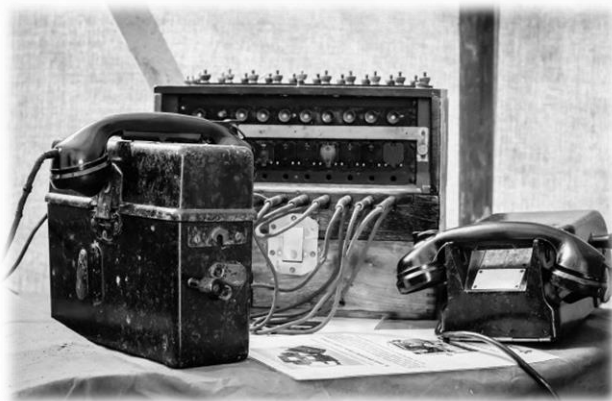
密码促进激发网络空间安全发展创造力。



• 网络空间安全形势与商用密码

国内形势—初识密码

- 1930年1月15日深夜，我党第一份加密电报变为一组组无线电波，划过从香港到上海的漫漫夜空。这标志着我党的机要密码工作正式诞生。
- 密码工作从无到有，在我国革命和建设的各个历史时期，都发挥了不可替代的重要作用。
- 特别是党的十八大以来，在以习近平同志为核心的党中央坚强领导下，在中央密码工作领导小组指挥下，密码事业取得历史性成就、实现历史性变革。





- 网络空间安全形势与商用密码

国内形势-- 《中华人民共和国密码法》颁布实施

2019年10月26日《密码法》颁布，于2020年1月1日施行。要求密码工作坚持总体国家安全观，遵循统，领导、分级负责明确了“党管密码”的根本原则，创新发展、服务大局、依法管理、保障安全的原则





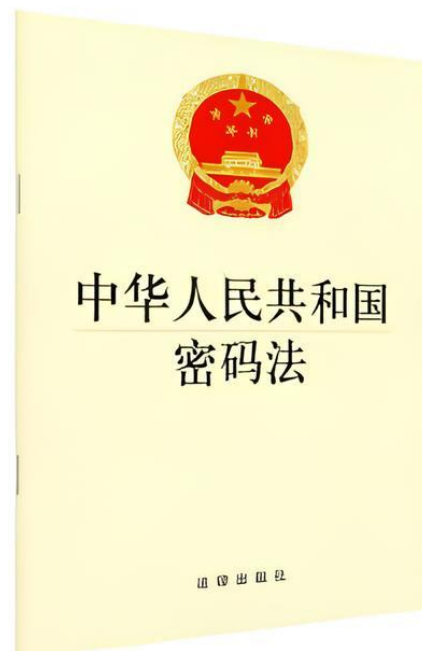
• 网络空间安全形势与商用密码

国内形势--《密码法》实施之前

1999年颁布《商用密码管理条例》是我国密码领域**第一部**行政法规其中明确商用密码管理的总原则有两个，党管密码、依法行政

在《密码法》实施前，商用密码管理依据的法律法规主要包括：

- 一部涉及规范多项密码管理工作的法律--《**电子签名法**》
- 一部行政法规《**商用密码管理条例**》
- 八个专项管理规定及若干规范性文件



《密码法》实施前商用密码法律法规体系



• 网络空间安全形势与商用密码

国内形势--《密码法》立法情况

- 第一章总则。规定了立法目的、密码工作的基本原则、领导和管理体制以及密码发展促进和保障措施；
- 第二章核心密码、普通密码。规定了核心密码、普通密码使用要求、安全管理制度以及国家加强核心密码、普通密码工作的一系列特殊保障制度和措施；
- 第三章**商用密码**。规定了商用密码标准化制度、检测认证制度、市场准入管理制度、使用要求、进出口管理制度、电子政务电子认证服务管理制度以及商用密码事中事后监管制度；





• 密码分类

我国《密码法》将密码分为三类，分别是核心密码、普通密码和商用密码。

核心密码、普通密码

属于国家秘密

- 用于保护国家秘密
- 核心密码保护信息的最高密级为绝密级
- 普通密码保护信息的最高密级为机密级

商用密码

不属于国家秘密

- 公民、法人和其他组织可以依法适用商用密码保护网络与信息安全

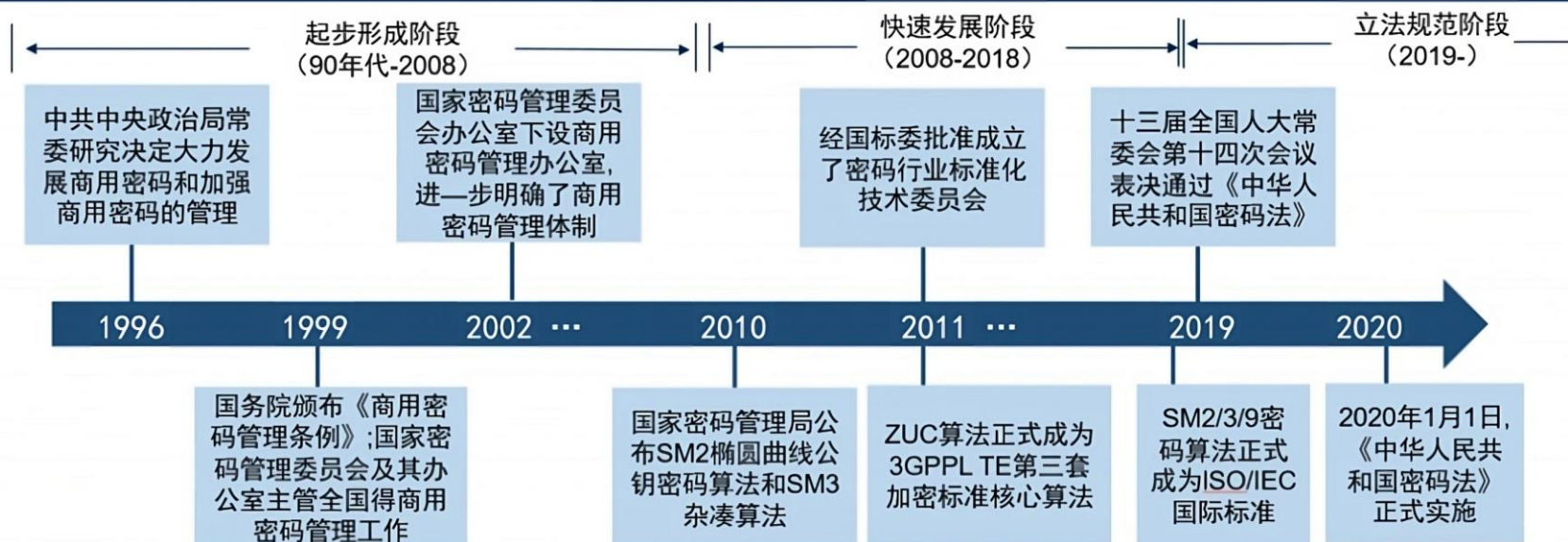
商用密码介绍





• 商用密码介绍—发展历程

我国商用密码行业发展历程





• 商用密码介绍—发展历程

- 2011年9月，祖冲之（ZUC）序列密码算法纳入国际第三代合作伙伴计划组织（3GPP）的4G移动通信密码算法国际标准，用于移动通信系统空中传输信道的信息加密和完整性保护，这是我国密码算法首次成为国际标准。
- 2017年11月，SM2和SM9数字签名算法成为国际标准，进入标准发布阶段。
- 2018年10月，SM3密码杂凑算法正式成为ISO/IEC国际标准。
- 2018年11月，SM2和SM9数字签名算法正式成为ISO/IEC国际标准。
- 2020年4月，ZUC序列密码算法正式成为ISO/IEC国际标准。
- 2021年2月，SM9标识加密算法正式成为ISO/IEC国际标准。
- 2021年6月，SM4分组密码算法正式成为ISO/IEC国际标准。
- 2021年10月，SM9密钥交换协议正式成为ISO/IEC国际标准。



- 商用密码介绍--商密算法概述





- 商用密码介绍--商密算法概述

商用密码算法			简介	发布	对标国际算法
对称密码算法	分组密码	SM1	以芯片、IP 核形式，硬件部署	未公开	AES、3DES
		SM4	用途广泛，可用于大数据量的加密	公开	AES、3DES
		SM7	轻量级分组密码，适合资源受限环境	未公开	AES、3DES
	序列密码	ZUC	流密码，密钥与明文逐比特异或计算	公开	SNOW 3G、RC4
非对称密码算法		SM2	基于椭圆曲线，支持数字签名、密钥交换、公钥加密	公开	RSA、ECC、ECDSA
		SM9	基于双线性对，是标识密码算法，用户公钥与标识相关；私钥由 KGC 基于标识生成；支持数字签名、密钥交换、公钥加密	公开	IBE
密码杂凑算法		SM3	哈希算法，计算摘要	公开	MD5、SHA系列



• 商用密码介绍--商用密码技术产品进展

01

产品创新日趋活跃

- 在硬件方面，现已有国产化商用密码卡实现主控器使用国产化芯片，且算法及配套外设均实现国产化，密码卡的制造工艺、综合性能、兼容性、安全性等方面取得长足进步。
- 在软件方面，数据加密平台产品可实现密码保密一体化、分布式加密、集中式管控等功能，实现了在现有系统上的安全能力叠加增强。
- 在与新技术新应用结合方面，物联网安全密码应用解决方案实现了物联网环境身份认证、传输加密、信息防篡改的安全需求，轻量化密钥管理身份标识与密钥信息占用极小的物联网终端系统资源，在保障物联网终端安全的基础上不影响其工作效率。

02

技术取得长足进步

- 在国家政策的大力引导下，商用密码的应用正在向工业互联网、车联网、智慧城市等新兴领域融入。
- 不同领域的应用诉求对密码技术和相关适配设备提出差异化要求，特别是新兴领域对密码功能和性能的要求更高，如面向云和大数据场景的同态加密技术、高性能密码技术，面向智慧城市等场景的轻量级加密技术。



03

算法体系不断突破

- 已建成基于SM2算法的国家电子认证信任体系。
- ZUC、SM2、SM3、SM4、SM9等一系列商用密码算法构成了我国完整的密码算法体系，部分密码算法被采纳为国际标准，为促进国际密码学发展、丰富产业选择和保障应用安全提供了中国方案。









04

通过认证的产品日益丰富

- 截止到2021年底，共有2155款产品获得了公开有效的商用密码产品认证，以上产品分别由536家商用密码产业链企业生产或制造。
- 获得商用密码产品认证的硬件产品1677种，占比为78%；软件产品478种，占比为22%。



• 商用密码介绍--商用密码重点领域应用情况

 金融领域： 在网上银行、电子保单和网上证券安全方面，通过基于密码技术的数字证书、数字签名、电子签章、时间戳等，提供身份认证以及业务数据和电子合同的机密性完整性保护。	 通信领域： 三大运营商建设的600多万个基站系统全部支持ZUC算法，高通、华为、展讯、MTK等主流基带芯片厂商的产品都支持ZUC算法，基于这些基带芯片生产的数亿部移动智能手机默认支持ZUC算法。	 工业互联网领域： 工业互联网平台应用密码技术实现不同层级间的安全传输及跨平台的身份互认需求，建设以密码为核心的云安全保障体系。	 交通领域： 《数字中国发展报告（2020年）》统计数据显示，当前我国高速公路电子不停车收费（ETC）车道达到6.6万条，客车ETC使用率超过70%，货车ETC使用率超过56%。截止到2021年9月，全国已实现314个地级以上城市交通一卡通互联互通。
 电力领域： 在电力行业重要信息系统中，提供了体系化、规模化的安全保障，可保障发电环节、输电环节、变电环节、配电环节及用电信息采集环节的安全稳定运行。	 医疗领域： 满足医疗云/医疗信息系统上各项业务应用的安全需求，为构建一体化“互联网+医疗服务”提供密码支撑保障，助力数字化转型，全面提升各类医疗信息系统密码应用水平。	 电子政务领域： 国家电子政务外网基于商用密码SM2算法建立身份认证系统，确保网络行为主体身份的唯一性、真实性和合法性；利用密码技术确保网络数据的安全传输；基于PKI技术的电子认证保证网上传递数据的真实性、完整性、保密性；采用SM9标识密码算法和数字签名、数据加密技术实现强身份认证机制和邮件内容加密。	 教育领域： 主要应用于全国教育管理信息化业务和教育卡等方面。按照教育部要求，在教育和科研计算机网、教育管理、教育资源、电子校务、教育基础数据、教育卡等信息系统，以及面向社会服务的教育政务系统中加强密码应用。

SM2 椭圆曲线公钥密码算法





- SM2 椭圆曲线公钥密码算法
 - SM2 公钥加密算法
 - SM2 数字签名算法
 - SM2 密钥交换协议

SM2椭圆曲线公钥密码算法（简称SM2算法）于2010年12月首次公开发布，2012年成为中国商用密码标准（标准号为GM/T 0003—2012），2016年成为中国国家密码标准（标准号为GB/T 32918—2016）。



- **SM2 椭圆曲线公钥密码算法--公钥加密算法**

仅介绍基于素数域 \mathbb{F}_p 上椭圆曲线构造的SM2公钥加密方案。

设 p 为一大于3的素数、 $a, b \in \mathbb{F}_p$ 且 $4a^3 + 27b^2 \neq 0$ ，则 \mathbb{F}_p 上的同余方程 $y^2 \equiv x^3 + ax + b \pmod{p}$ 的所有解 (x, y) ($x, y \in \mathbb{F}_p$) 连同同一个无穷远点 \mathcal{O} 共同构成 \mathbb{F}_p 上的椭圆曲线，记为 $E(\mathbb{F}_p)$ 。椭圆曲线 $E(\mathbb{F}_p)$ 上点的个数用 $\#E(\mathbb{F}_p)$ 表示，称为椭圆曲线 $E(\mathbb{F}_p)$ 的阶。



- SM2 椭圆曲线公钥密码算法--公钥加密算法

该方案的参数包括构造该方案所需的椭圆曲线参数、运算符号以及辅助函数。

椭圆曲线参数：SM2 椭圆曲线公钥密码算法所需的椭圆曲线参数为 $(p, a, b, G, n, h, SEED)$ 。

- \mathbb{F}_p ：特征为 p 的有限域， p 要求为一个长度不小于192比特的大素数，即 $p > 2^{191}$ 。
- a, b ：椭圆曲线 $E(\mathbb{F}_p)$ 对应的同余方程的系数， a 和 $b \in \mathbb{F}_p$ 。
- $n, G = (x_G, y_G)$ ： G 为 $E(\mathbb{F}_p)$ 的基点且为非无穷远点， x_G 和 $y_G \in \mathbb{F}_p$ 。 n 为 G 的阶，即 n 是使得 $nG = \mathcal{O}$ 成立的最小正整数。 n 要求为一个长度不小于192比特的大素数（即 $n > 2^{191}$ ）且 $n > 4p^{1/2}$ 。由 G 可得到 $E(\mathbb{F}_p)$ 上的循环群 $\langle G \rangle = \{\mathcal{O}, G, 2G, \dots, (n-1)G\}$ 。
- h ：余因子，其为 $E(\mathbb{F}_p)$ 的阶与 n 相除的整数部分。
- $SEED$ ：长度不小于192比特的字符串（ $SEED$ 为可选参数，用于随机产生椭圆曲线，即可从 $SEED$ 派生出椭圆曲线对应的同余方程的系数 a, b ）。



- **SM2 椭圆曲线公钥密码算法--公钥加密算法**

运算符号

- $x \oplus y$: 将 x 和 y 进行逐比特异或运算, 其中 x 和 y 均为比特串或者字节串
- $x \parallel y$: 将 x 和 y 进行拼接, 其中 x 和 y 均为比特串或者字节串
- kP : 椭圆曲线上点 P 的 k 倍点
- $[x, y]$: 大于等于 x 且小于等于 y 的整数集合
- $\lceil x \rceil$: 大于或等于 x 的最小整数
- $\lfloor x \rfloor$: 小于或等于 x 的最大整数



- SM2 椭圆曲线公钥密码算法--公钥加密算法

辅助函数

1. 哈希函数 $H: \{0,1\}^l \rightarrow \{0,1\}^{256}$: SM2公钥加密方案将GB/T 32905-2016中定义的SM3算法作为哈希函数来使用。即 $H: \{0,1\}^l \rightarrow \{0,1\}^{256}$, 输入为长度 $l < 2^{64}$ 比特的任意消息, 输出长度为256比特的哈希值。
2. 密钥派生函数 $KDF: \{0,1\}^L \rightarrow \{0,1\}^{klen}$: 输入为长度 L 不小于192比特的密钥派生材料 Z , 输出为 $klen$ 比特的密钥 K 。若SM2公钥加密方案中使用的哈希函数的输出的哈希值长度为256比特, 则 $L = 512$, $klen < 256 \times (2^{32} - 1)$ 。
3. 随机数生成器: SM2公钥加密方案规定使用国家密码管理局批准的随机数生成器(可使用真随机数生成器也可使用伪随机数生成器)。



- SM2 椭圆曲线公钥密码算法--公钥加密算法

SM2公钥加密方案的明文空间为 $\mathcal{M} = \{0,1\}^{klen}$ ($klen < 256 \times (2^{32} - 1)$)、密文空间为 $\mathcal{C} = \{0,1\}^{klen+768}$ 、密钥空间为 $\mathcal{K} = \left\{ \begin{pmatrix} p, a, b, G, n, h, SEED, \\ H, KDF, RNG, d, Y \end{pmatrix} \middle| Y = dG \bmod p \right\}$ 。
SM2公钥加密方案 $PKE=(Gen, Enc, Dec)$ 的各个算法描述如下。

- $Gen(\lambda)$: 当输入安全参数 λ , 该算法执行如下步骤:

1. 选择椭圆曲线参数 $(p, a, b, G, n, h, SEED)$ 。
2. 选择三类辅助函数: 哈希函数 $H: \{0,1\}^l \rightarrow \{0,1\}^{256}$ 、密钥派生函数 $KDF: \{0,1\}^L \rightarrow \{0,1\}^{klen}$ 以及随机数生成器。
3. 用随机数生成器产生随机数 $d \in [1, n-2]$ 作为私钥。
4. 计算椭圆曲线点 $Y = dG$ 。
5. 输出公钥 $pk = (p, a, b, G, n, h, SEED, H, KDF, RNG, Y)$ 和私钥 d 。



- SM2 椭圆曲线公钥密码算法--公钥加密算法

- $Enc(M, pk)$: 当输入明文 $M \in \mathcal{M}$ 、公钥 pk 时, 该算法执行如下步骤:
 1. 用随机数生成器产生随机数 $k \in [1, n - 1]$ 。
 2. 计算椭圆曲线点 $C_1 = kG = (x_1, y_1)$ 。
 3. 计算椭圆曲线点 $S = hY$ 。若 S 是无穷远点, 则报错并终止算法。
 4. 计算椭圆曲线点 $kY = (x_2, y_2)$ 。
 5. 令 $Z = x_2 \parallel y_2$, 计算 $K = KDF(Z)$ 。若 K 为全0比特串, 则返回第一步。
 6. 计算 $C_2 = K \oplus M$, $C_3 = H(x_2 \parallel M \parallel y_2)$ 。
 7. 输出密文 $C = C_1 \parallel C_3 \parallel C_2$ 。



- SM2 椭圆曲线公钥密码算法--公钥加密算法

- $Dec(C, d)$: 当输入密文 $C = C_1 || C_3 || C_2$ 、私钥 d 时, 该算法执行如下步骤:
 1. 验证点 C_1 是否满足椭圆曲线方程。若不满足则报错终止。
 2. 计算椭圆曲线点 $S = hC_1$ 。若 S 是无穷远点, 则报错并终止。
 3. 计算 $dC_1 = (x_2, y_2)$ 。
 4. 令 $Z = x_2 || y_2$, 计算 $K = KDF(Z)$ 。若 K 为全0比特串, 则报错并终止。
 5. 从 C 中取出比特串 C_2 , 计算 $M = C_2 \oplus K$ 。
 6. 计算 $u = H(x_2 || M || y_2)$, 从 C 中取出比特串 C_3 。若 $u \neq C_3$, 则报错并终止。
 7. 输出明文 M 。



• SM2 椭圆曲线公钥密码算法—数字签名算法

该方案的参数包括构造该方案所需的椭圆曲线参数、运算符号、辅助函数和用户其他信息。SM2数字签名方案所涉及的椭圆曲线参数、运算符号和辅助函数可参考 SM2 公钥加密方案所涉及的参数。用户其他信息的格式为 $(ID, entlen, ENTL, Y, Z)$ ，具体定义如下：

- ID ：用户可辨别身份标识。按照标准ISO/IEC 15946-3 3.9， ID 应选择可以无歧义辨别某一实体身份的信息。
- $entlen$ ： ID 的比特长度。
- $ENTL$ ： $entlen$ 转化成比特串后的结果，其长度为两个字节。
- $Y = (x, y)$ ：用户公钥，其为 $E(\mathbb{F}_p)$ 上的一个点。
- $Z = H_{256}(ENTL || ID || a || b || x_G || y_G || x || y)$ ：用户哈希值，其中 a, b, x_G, y_G 是椭圆曲线参数。



• SM2 椭圆曲线公钥密码算法--数字签名算法

SM2数字签名方案的消息空间为 $\mathcal{M} = \{0,1\}^*$ 、签名空间为 $\Sigma = \mathbb{F}_n^* \times \mathbb{F}_n^*$ ，密钥空间为 $\mathcal{K} = \left\{ \left(\begin{array}{c} (p, a, b, G, n, h, SEED, x_G, y_G, H, \\ H_l, RNG, d, Y \end{array} \right) \middle| Y \equiv dG \bmod p \right\}$ 。SM2数字签名方案 $DSS = (DGen, Sign, Verify)$ 的各个算法描述如下。

● $DGen(\lambda)$ ：当输入安全参数 λ 时，该算法执行如下步骤：

1. 选择椭圆曲线参数 $(p, a, b, G, n, h, SEED, x_G, y_G)$ 。
2. 选择两类辅助函数：哈希函数 $H: \{0,1\}^l \rightarrow \{0,1\}^{256}$ 、哈希函数 $H_l = \{0,1\}^* \rightarrow \{0,1\}^l$ 以及随机数生成器 RNG 。
3. 用随机数生成器产生随机数 $d \in [1, n-2]$ 作为私钥。
4. 计算椭圆曲线点 $Y = dG$ 。
5. 输出公私钥对 (pk, sk) ，其中 $pk = (p, a, b, G, n, h, SEED, x_G, y_G, H, H_l, RNG, Y)$ ， $sk = d$ 。



- SM2 椭圆曲线公钥密码算法--数字签名算法

- $Sign(M, sk)$: 当输入消息 $M \in \mathcal{M}$ 、私钥 $sk = d$ 时, 该算法执行如下步骤:

1. 选择用户的其它信息 $(ID, entlen, ENTL, x, y, Z)$ 。
2. 置 $\bar{M} = Z \parallel M$ 。
3. 计算 $e = H_l(\bar{M})$ 。
4. 用随机数生成器产生随机数 $k \in [1, n - 1]$ 。
5. 计算椭圆曲线点 $kG = (x_1, y_1)$ 。
6. 计算 $r = (e + x_1) \bmod n$, 若 $r = 0$ 或 $r + k = n$, 则返回第4步。
7. 计算 $s = ((1 + d)^{-1}(k - r \cdot d)) \bmod n$, 若 $s = 0$, 则返回第4步。
8. 输出签名 $\sigma = (r, s)$ 。



- SM2 椭圆曲线公钥密码算法--数字签名算法

- $Verify(pk, M, \sigma)$: 当输入公钥 $pk = (p, a, b, G, n, h, SEED, x_G, y_G, H, H_l, RNG, Y)$ 、消息 M 、签名 $\sigma = (r, s)$ 时, 该算法执行如下步骤:
 1. 检验 $r \in \mathbb{F}_n^*$ 是否成立, 若不成立则输出 0。
 2. 检验 $s \in \mathbb{F}_n^*$ 是否成立, 若不成立则输出 0。
 3. 置 $\bar{M} = Z \parallel M$ 。
 4. 计算 $e = H_l(\bar{M})$ 。
 5. 计算 $t = (r + s) \bmod n$, 若 $t = 0$, 则输出 0。
 6. 计算椭圆曲线点 $(x'_1, y'_1) = sG + tY$ 。
 7. 计算 $R = (e + x'_1) \bmod n$, 检验 $R = r$ 是否成立, 若成立则输出 1; 否则输出 0。



- SM2 椭圆曲线公钥密码算法--密钥交换算法

SM2密钥交换算法所需的椭圆曲线参数为 $(p, a, b, G, n, h, SEED, w, klen)$

- $p, a, b, G, n, h, SEED$ 可参考SM2公钥加密算法。
- $w = \lceil ([\log_2(n)]/2) \rceil - 1$: 密钥协商计算参数。
- $klen$: 协商获得密钥数据的长度。



- **SM2 椭圆曲线公钥密码算法—密钥交换算法**

该方案的参数包括构造该方案所需的**运算符号**、**辅助函数**和用户**其他信息**。可参考SM2数字签名方案所涉及的参数。



- SM2 椭圆曲线公钥密码算法--密钥交换算法

令协议参与方集合为 $\mathbb{U} = \{u_i, u_j\}$, SM2密钥交换算法 $KA = \{KSetup, KGen, Agree\}$ 各个算法描述如下。

- $KASetup(\lambda)$: 当输入安全参数 λ , 该算法执行如下步骤:

1. 选择椭圆曲线参数 $(p, a, b, G, n, h, SEED, w, klen)$ 。
2. 选择三类辅助函数: 哈希函数 $H: \{0,1\}^l \rightarrow \{0,1\}^{256}$ 、密钥派生函数 $KDF: \{0,1\}^L \rightarrow \{0,1\}^{klen}$ 以及随机数生成器。
3. 输出公共参数 $(p, a, b, G, n, h, SEED, w, klen, H, KDF, RNG)$ 。



- SM2 椭圆曲线公钥密码算法--密钥交换算法

- $KGen(params)$: u_i 和 u_j 分别为自己生成公私钥对。

1. u_i 用随机数生成器产生随机数 $d_i \in [1, p - 1]$ 作为私钥, u_j 用随机数生成器产生随机数 $d_j \in [1, p - 1]$ 作为私钥。
2. u_i 计算椭圆曲线点 $Y_i = d_i G$; u_j 计算椭圆曲线点 $Y_j = d_j G$ 。
3. u_i 输出公私钥对 (pk_i, sk_i) , 其中 $pk_i = Y_i$, $sk_i = d_i$; u_j 输出公私钥对 (pk_j, sk_j) , 其中 $pk_j = Y_j$, $sk_j = d_j$ 。



- SM2 椭圆曲线公钥密码算法--密钥交换算法

- $Agree(params, \mathbb{U})$: 假设参与者集合 \mathbb{U} 对应的公钥集合为 $\mathbb{P} = \{pk_i, pk_j\}$, u_i 为协议发起者。此处密钥协商算法是一个交互式的算法, 具体步骤如下:
 1. u_i 用随机数生成器产生随机数 $r_i \in [1, n-2]$; u_i 计算椭圆曲线点 $R_i = r_i G = (x_1, y_1)$; u_i 将 R_i 发送给参与方 u_j 。
 2. u_j 使用随机数生成器产生随机数 $r_j \in [1, n-2]$;
 - ✧ u_j 计算椭圆曲线点 $R_j = r_j G = (x_2, y_2)$;
 - ✧ u_j 计算 $\overline{x_2} = 2^w + (x_2 \& (2^w - 1))$, $t_j = (sk_j + \overline{x_2} \cdot r_j) \bmod n$ 。
 - ✧ u_j 验证 R_i 是否满足椭圆曲线方程, 若不满足则协商失败; 否则从 R_i 中获得 x_1 , 计算 $\overline{x_1} = 2^w + (x_1 \& (2^w - 1))$ 。
 - ✧ u_j 计算椭圆曲线上点 $V = h \cdot t_j(pk_i + \overline{x_1} R_i) = (x_V, y_V)$, 计算 $K_j = KDF(x_V || y_V || Z_i || Z_j, klen)$, Z_i 为参与方 u_i 的可辨别标识, Z_j 为参与方 u_j 的可辨别标识。



• SM2 椭圆曲线公钥密码算法--密钥交换算法

3. (标准中还设计了下述用于密钥确认的可选执行步骤: u_j 计算 $S_j = H(0X02 || y_v || H(x_v || Z_i || Z_j || x_1 || y_1 || x_2 || y_2 ||))$ 。 u_j 将 R_j (选项 S_j) 发送给用户 u_i 。)
4. u_i 计算 $\overline{x_1} = 2^w + (x_1 \& (2^w - 1))$, $t_i = (sk_i + \overline{x_1} \cdot r_i) \bmod n$ 。 u_i 验证 R_j 是否满足椭圆曲线方程, 若不满足则协商失败; 否则从 R_j 中获得 x_2 , 计算 $\overline{x_2} = 2^w + (x_2 \& (2^w - 1))$ 。 u_i 计算椭圆曲线上点 $U = h \cdot t_i(pk_j + \overline{x_2}R_j) = (x_U, y_U)$, 计算 $K_i = KDF(x_U || y_U || Z_i || Z_j, klen)$ 。
5. 标准中还设计了下述用于密钥确认的可选执行步骤:
 - ✧ u_i 计算 $S_1 = H(0X02 || y_U || H(x_U || Z_i || Z_j || x_1 || y_1 || x_2 || y_2 ||))$, 并检验 $S_1 = S_j$ 是否成立。若等式不成立则表示密钥确认失败, 协议终止; 否则 u_i 计算 $S_i = H(0X03 || y_U || H(x_U || Z_i || Z_j || x_1 || y_1 || x_2 || y_2 ||))$, 并将 S_i 发送给参与方 u_j 。
 - ✧ u_j 计算 $S_2 = H(0X03 || y_v || H(x_v || Z_i || Z_j || x_1 || y_1 || x_2 || y_2 ||))$, 并检验 $S_2 = S_i$ 是否成立, 若等式不成立则从 u_i 到 u_j 的密钥确认失败。)

SM9 标识密码算法





- **SM9 标识密码算法--标识密码体系**

在传统**公钥基础设施PKI**体系下，一个实体具有一对密钥：

- 一个是公开的公钥；
- 一个是自己保管的私钥。

在这样的系统中，信息发送方需要正确获取接收方的公钥，否则就可能出现恶意攻击方可通过提供虚假接收方公钥，即中间人攻击的方式获取信息。（数字签名情况类似）

- 为了证明接收方公钥的可信，传统 KPI 体系就需要一个受信任的第三方来绑定实体与其拥有的密码对。
- 这个受信第三方即证书颁发机构（CA: Certificate Authority），在对实体进行身份核实后，就为其生成一个公私钥对（也可由实体自己生成），并为该实体颁发一个数字证书，证书中包含实体的标识、公钥以及CA的数字签名。
- 信息发送方在发送信息前需要获取接收方的证书。在验证证书的有效性后，发送方使用证书中的公钥加密。这就是基于数字证书的 PKI 系统。



- **SM9 标识密码算法--标识密码体系**

基于数字证书的 PKI 系统有几个问题：

- 发送方在发送信息前必须先获取接收方的证书
- 发送方在收到一个证书后，都需要验证证书的有效性
- 认证中心需要花费大量成本管理和维护证书，非常复杂，难以部署



- **SM9 标识密码算法--标识密码体系**

为了解决传统 PKI 的一些问题，密码学家 Shamir 于1984年提出基于标识的密码体系，即 IBC（Identity-Based Cryptograph）。其最主要观点是：

- 不需要使用证书传递公钥，而是使用用户标识如姓名、IP地址、电子邮箱地址、手机号码等代表用户的标识信息作为公钥
 - 私钥则由密钥中心（**Key Generate Center**, 简称**KGC**）根据系统主密钥和用户标识计算得出。
-
- 自 Shamir 1984年提出 IBC 思想的十几年，一直停留在概念阶段。
 - 直到2000年，D.Boneh 和 M.Franklin, 以及 R.Sakai、K.Ohgishi 和 M.Kasahara 两个团队均提出基于椭圆曲线数学难题的配对构造算法，解决了安全性和实现效率最优化的问题，标识密码终于引发了标识密码的新发展。
 - 2008年，我国将 IBC 正式纳入中国国家算法标准，并获得国家商用密码管理局颁发的算法型号 **SM9**。

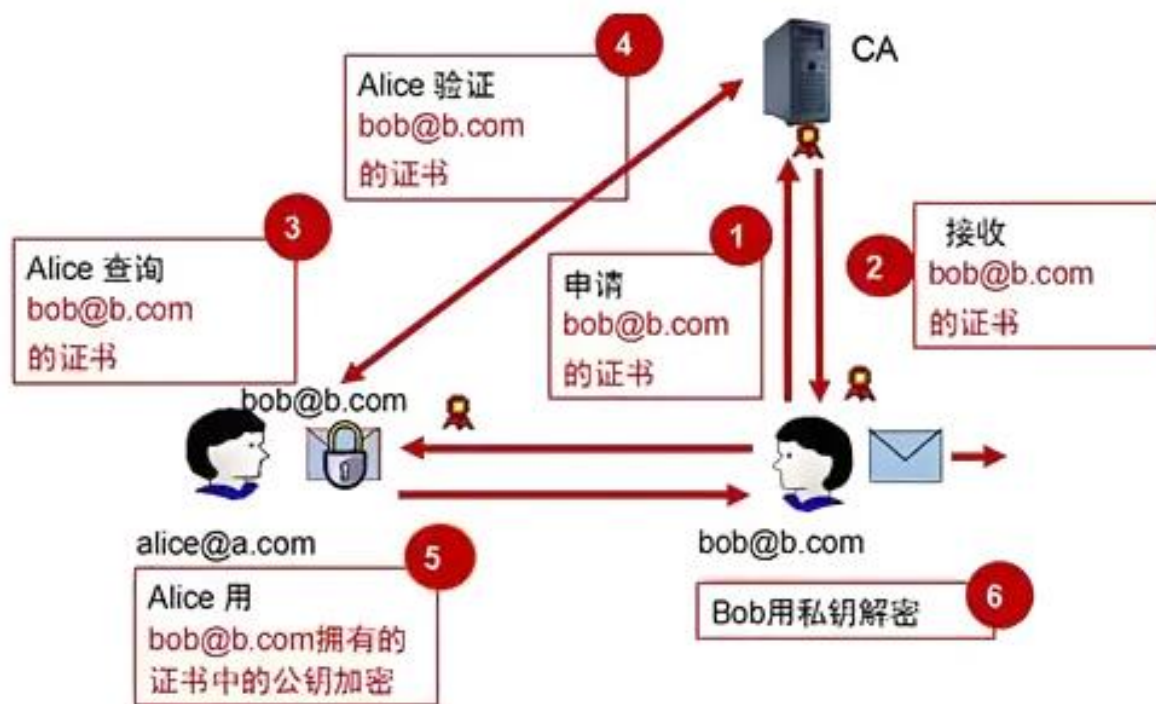


- SM9 标识密码算法--标识密码体系

PKI体系	IBC体系
公钥是随机数	公钥可以是邮箱地址
通过证书将用户的公钥与身份关联起来	公钥即用户的身份标识
信息发送方必须获得接收方的公钥证书	信息发送方只需要获知接收方的身份标识 (如姓名、IP地址、电子邮箱地址、手机号码等)
证书颁发和管理系统复杂难以部署	无需颁发和管理证书
难于实现基于属性、策略的加密	可增加时间或固定IP等方式解密信息的安全策略控制
存放的收信方证书随发送邮件数量的增大而增多，在线通讯交互越繁忙，管理负担和管理成本会同比例放大	发送邮件数量级越大，管理负担和管理成本的增加并不明显
实现成本高、效率低下	实现成本低、效率较高
系统运行维护成本高	运营管理方便，成本低



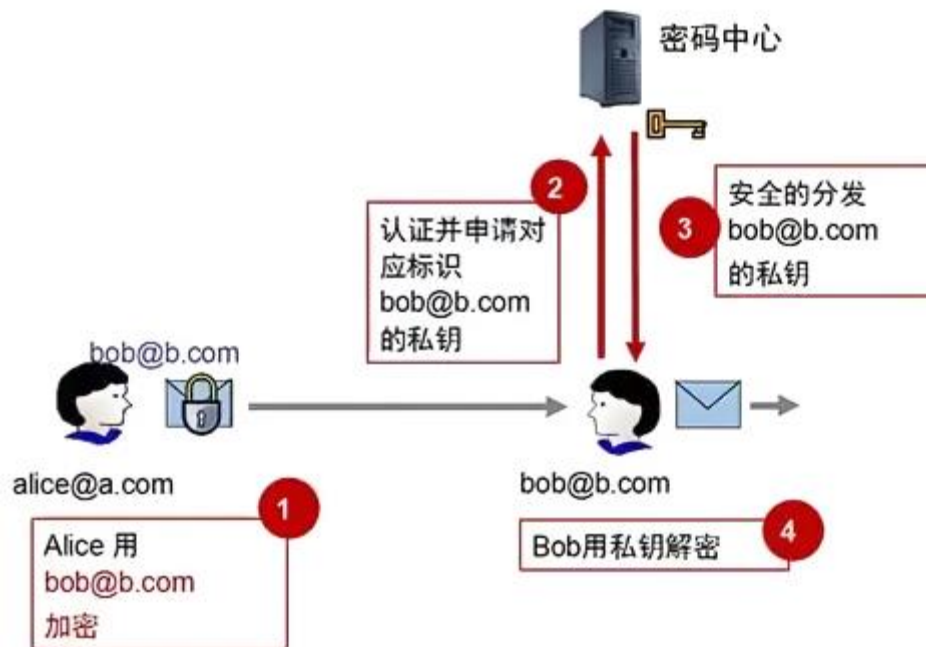
- SM9 标识密码算法--标识密码体系



PKI流程



- SM9 标识密码算法--标识密码体系



IBC流程



- SM9 标识密码算法

- SM9 密钥封装机制和公钥加密算法
- SM9 数字签名算法
- SM9 密钥交换协议



• SM9 标识密码算法—预备知识

双线性映射:

令 \mathbb{G}_1 和 \mathbb{G}_2 分别是阶为素数 p 的加法循环群、 G_1 和 G_2 分别是群 \mathbb{G}_1 和 \mathbb{G}_2 的生成元、 \mathbb{G}_T 是阶为素数 p 的乘法循环群。若 $\bar{e}: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ 为一个双线性映射，则它满足如下性质：

- 双线性性：对于任意的 $u, v \in \mathbb{Z}_p^*$ ， $\bar{e}(G_1^u, G_2^v) = \bar{e}(G_1, G_2)^{uv}$ 。
- 非退化性： $g_T = \bar{e}(G_1, G_2)$ 是 \mathbb{G}_T 的一个生成元。
- 可计算性：对于任意的 $P \in \mathbb{G}_1$ ， $Q \in \mathbb{G}_2$ ，存在高效的算法能计算 $\bar{e}(P, Q)$ 。



• SM9 标识密码算法—预备知识

通常来说，双线性映射 $\bar{e}: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ 可分为如下三类。

- 第一类双线性映射：若 $\mathbb{G}_1 = \mathbb{G}_2$ ，则称该类双线性映射为第一类双线性映射。
- 第二类双线性映射：若 $\mathbb{G}_1 \neq \mathbb{G}_2$ 且存在一个有效可计算的同构映射 $\psi: \mathbb{G}_2 \rightarrow \mathbb{G}_1$ 使得 $\psi(G_2) = G_1$ ，则称该类双线性映射为第二类双线性映射。
- 第三类双线性映射：若 $\mathbb{G}_1 \neq \mathbb{G}_2$ 且不存在上述有效可计算的同构映射 ψ 使得 $\psi(G_2) = G_1$ ，则称该类双线性映射为第三类双线性映射。



- **SM9 标识密码算法—预备知识**

- 密钥生成中心(key generation center; KGC)

负责选择系统参数、生成加密主密钥并产生用户加密私钥的可信机构

- 加密主密钥(encryption master key)

处于标识密码密钥分层结构最顶层的密钥，包括加密主私钥和加密主公钥，其中加密主公钥公开，加密主私钥由 KGC 秘密保存。KGC 用加密主私钥和用户的标识生成用户的加密私钥。在标识密码中，加密主私钥一般由 KGC 通过随机数发生器产生，加密主公钥由加密主私钥结合系统参数产生



- SM9 标识密码算法—密钥封装

系统参数

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

双线性对 e 的值域为 N 阶乘法循环群 \mathcal{G}_T 。



- SM9 标识密码算法—密钥封装

系统加密主密钥和用户加密密钥的产生

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

KGC选择并公开用一个字节表示的签名私钥生成函数识别符 hid 。

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



- SM9 标识密码算法—密钥封装

辅助函数

密码杂凑函数 $H_v()$ 的输出是长度恰为 v 比特的杂凑值。

密码函数 $H_1(Z, n)$ 的输入为比特串 Z 和整数 n ，输出为一个整数 $h_1 \in [1, n-1]$ 。

密码函数 $H_2(Z, n)$ 的输入为比特串 Z 和整数 n ，输出为一个整数 $h_2 \in [1, n-1]$ 。



- SM9 标识密码算法—密钥封装

分组密码算法

分组密码算法包括加密算法 $Enc(K_1, m)$ 和解密算法 $Dec(K_1, c)$ 。 $Enc(K_1, m)$ 表示用密钥 K_1 对明文 m 进行加密，其输出为密文比特串 c ； $Dec(K_1, c)$ 表示用密钥 K_1 对密文 c 进行解密，其输出为明文比特串 m 或“错误”。密钥 K_1 的比特长度记为 K_1_len 。

本部分规定使用国家密码管理主管部门批准的分组密码算法，如SM4分组密码算法。



- SM9 标识密码算法—密钥封装

消息认证码函数

消息认证码函数 $MAC(K_2, Z)$ 的作用是防止消息数据 Z 被非法篡改，它在密钥 K_2 的控制下，产生消息数据比特串 Z 的认证码，密钥 K_2 的比特长度记为 K_2_len 。在本部分的基于标识的加密算法中，消息认证码函数使用密钥派生函数生成的密钥对密文比特串求取消息认证码，从而使解密者可以鉴别消息的来源和检验数据的完整性。

消息认证码函数需要调用密码杂凑函数。

设密码杂凑函数为 $H_v()$ ，其输出是长度恰为 v 比特的杂凑值。

消息认证码函数 $MAC(K_2, Z)$:

输入: 比特串 K_2 (比特长度为 K_2_len 的密钥)，比特串 Z (待求取消息认证码的消息)。

输出: 长度为 v 的消息认证码数据比特串 K 。

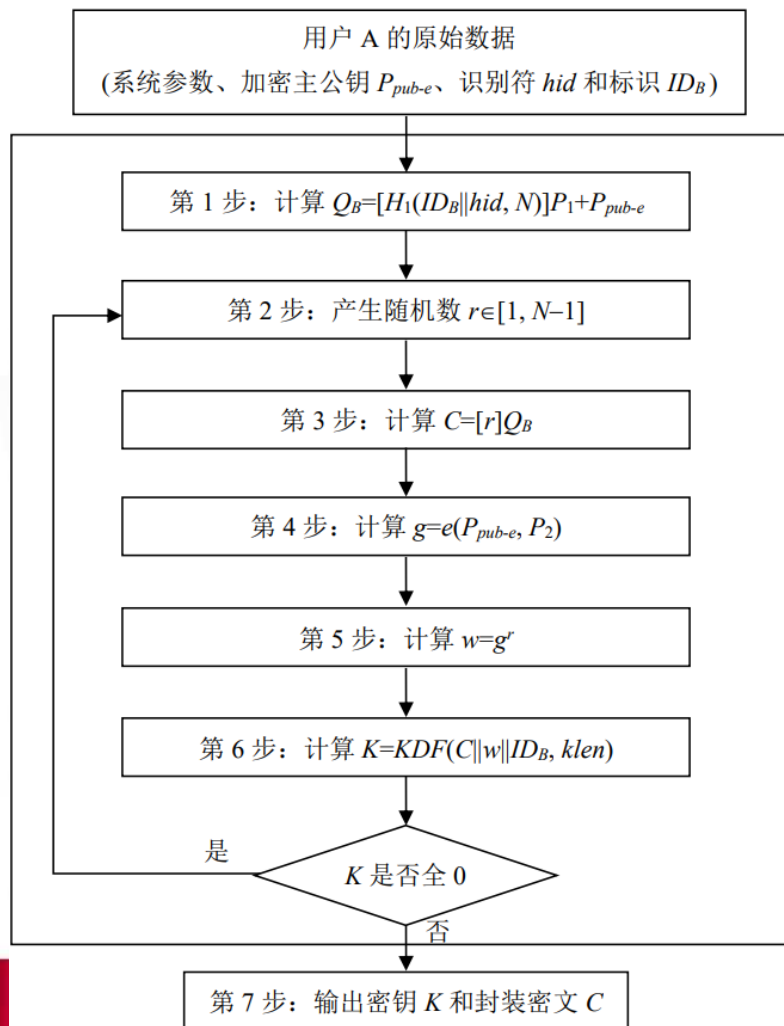
步骤1: $K = H_v(Z || K_2)$ 。



• SM9 标识密码算法—密钥封装(封装算法)

为了封装比特长度为 $klen$ 的密钥给用户B, 作为封装者的用户A需要执行以下运算步骤:

- A1: 计算群 G_1 中的元素 $Q_B=[H_1(ID_B||hid, N)]P_1+P_{pub-e}$;
- A2: 产生随机数 $r \in [1, N-1]$;
- A3: 计算群 G_1 中的元素 $C=[r]Q_B$, 将 C 的数据类型转换为比特串;
- A4: 计算群 G_T 中的元素 $g=e(P_{pub-e}, P_2)$;
- A5: 计算群 G_T 中的元素 $w=g^r$, 将 w 的数据类型转换为比特串;
- A6: 计算 $K=KDF(C||w||ID_B, klen)$, 若 K 为全 0 比特串, 则返回 A2。
- A7: 输出 (K, C) , 其中 K 是被封装的密钥, C 是封装密文。





• SM9 标识密码算法—密钥封装(解封装)

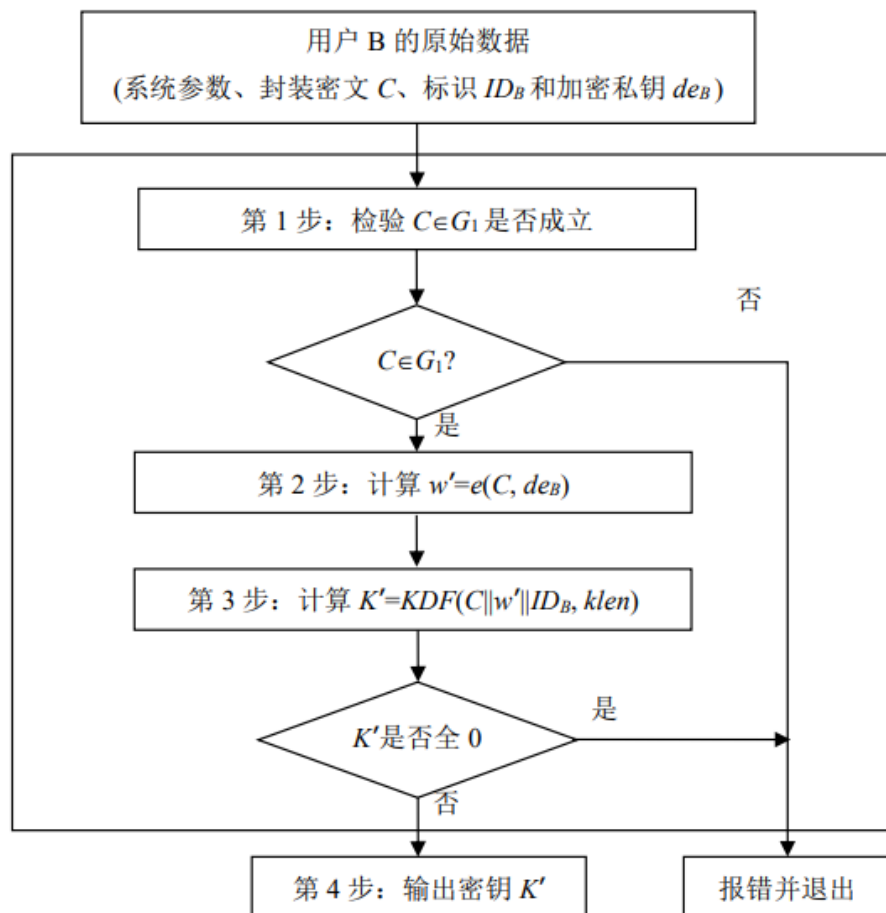
用户B收到封装密文 C 后, 为了对比特长度为 $klen$ 的密钥解封装, 需要执行以下运算步骤:

B1: 验证 $C \in G_1$ 是否成立, 若不成立则报错并退出;

B2: 计算群 G_T 中的元素 $w' = e(C, de_B)$, 将 w' 的数据类型转换为比特串;

B3: 将 C 的数据类型转换为比特串, 计算封装的密钥 $K' = KDF(C || w' || ID_B, klen)$, 若 K' 为全 0 比特串, 则报错并退出;

B4: 输出密钥 K' 。





• SM9 标识密码算法—公钥加密(加密算法)

设需要发送的消息为比特串 M , m_{len} 为 M 的比特长度, K_1_{len} 为分组密码算法中密钥 K_1 的比特长度, K_2_{len} 为函数 $MAC(K_2, Z)$ 中密钥 K_2 的比特长度。

为了加密明文 M 给用户 B , 作为加密者的用户 A 应实现以下运算步骤:

A1: 计算群 G_1 中的元素 $Q_B = [H_1(ID_B || hid, N)]P_1 + P_{pub-e}$;

A2: 产生随机数 $r \in [1, N-1]$;

A3: 计算群 G_1 中的元素 $C_1 = [r]Q_B$, 将 C_1 的数据类型转换为比特串;

A4: 计算群 G_T 中的元素 $g = e(P_{pub-e}, P_2)$;

A5: 计算群 G_T 中的元素 $w = g^r$, 按将 w 的数据类型转换为比特串;

A6: 按加密明文的方法分类进行计算:

a) 如果加密明文的方法是基于密钥派生函数的序列密码算法, 则

1) 计算整数 $klen = mlen + K_2_{len}$, 然后计算 $K = KDF(C_1 || w || ID_B, klen)$ 。令 K_1 为 K 最左边的 $mlen$ 比特, K_2 为剩下的 K_2_{len} 比特, 若 K_1 为全 0 比特串, 则返回 A2;

2) 计算 $C_2 = M \oplus K_1$ 。

b) 如果加密明文的方法是结合密钥派生函数的分组密码算法, 则

1) 计算整数 $klen = K_1_{len} + K_2_{len}$, 然后计算 $K = KDF(C_1 || w || ID_B, klen)$ 。令 K_1 为 K 最左边的 K_1_{len} 比特, K_2 为剩下的 K_2_{len} 比特, 若 K_1 为全 0 比特串, 则返回 A2;

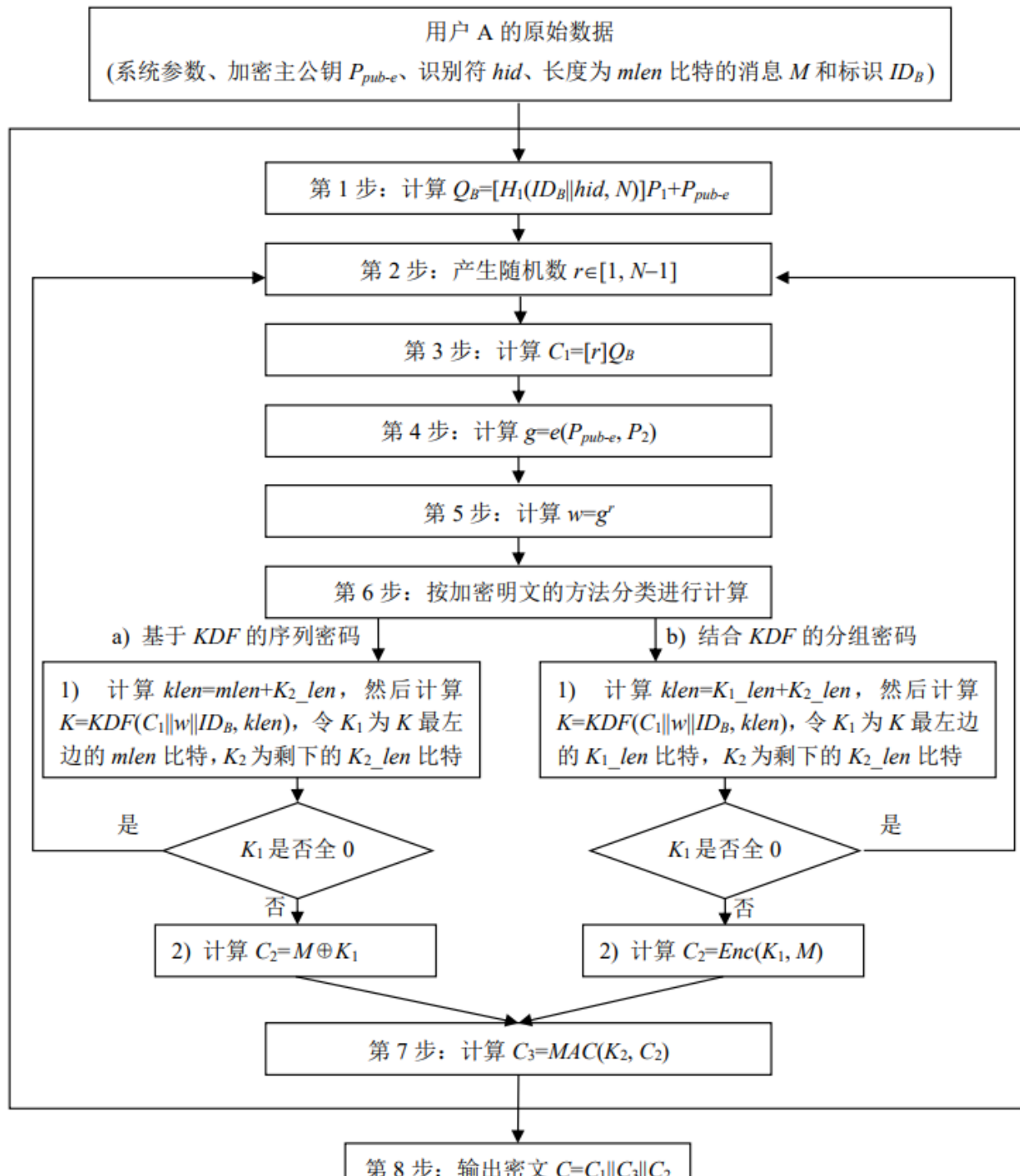
2) 计算 $C_2 = Enc(K_1, M)$ 。

A7: 计算 $C_3 = MAC(K_2, C_2)$;

A8: 输出密文 $C = C_1 || C_3 || C_2$ 。



- SM9 标识密码算法—
公钥加密(加密算法)





• SM9 标识密码算法—公钥加密(解密算法)

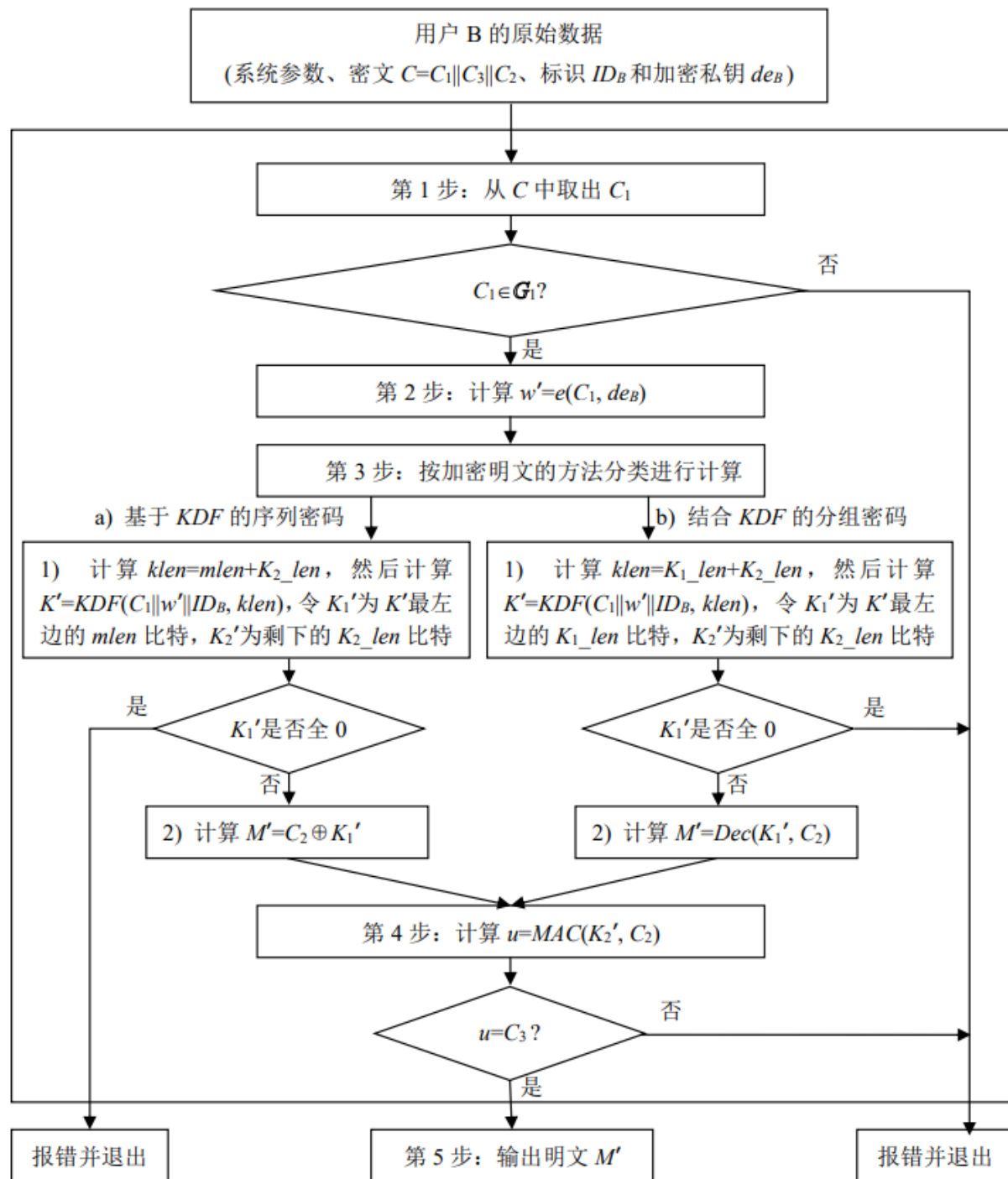
设 m_{len} 为密文 $C=C_1\|C_3\|C_2$ 中 C_2 的比特长度, K_1_{len} 为分组密码算法中密钥 K_1 的比特长度, K_2_{len} 为函数 $MAC(K_2, Z)$ 中密钥 K_2 的比特长度。

为了对 C 进行解密, 作为解密者的用户 B 应实现以下运算步骤:

- B1: 从 C 中取出比特串 C_1 , 将 C_1 的数据类型转换为椭圆曲线上的点, 验证 $C_1 \in \mathbf{G}_1$ 是否成立, 若不成立则报错并退出;
- B2: 计算群 \mathbf{G}_T 中的元素 $w'=e(C_1, de_B)$, 将 w' 的数据类型转换为比特串;
- B3: 按加密明文的方法分类进行计算:
 - a) 如果加密明文的方法是基于密钥派生函数的序列密码算法, 则
 - 1) 计算整数 $k_{len}=m_{len}+K_2_{len}$, 然后计算 $K'=KDF(C_1\|w'\|ID_B, k_{len})$ 。令 K_1' 为 K' 最左边的 m_{len} 比特, K_2' 为剩下的 K_2_{len} 比特, 若 K_1' 为全 0 比特串, 则报错并退出;
 - 2) 计算 $M'=C_2 \oplus K_1'$ 。
 - b) 如果加密明文的方法是结合密钥派生函数的分组密码算法, 则
 - 1) 计算整数 $k_{len}=K_1_{len}+K_2_{len}$, 然后计算 $K'=KDF(C_1\|w'\|ID_B, k_{len})$ 。令 K_1' 为 K' 最左边的 K_1_{len} 比特, K_2' 为剩下的 K_2_{len} 比特, 若 K_1' 为全 0 比特串, 则报错并退出;
 - 2) 计算 $M'=Dec(K_1', C_2)$ 。
- B4: 计算 $u=MAC(K_2', C_2)$, 从 C 中取出比特串 C_3 , 若 $u \neq C_3$, 则报错并退出;
- B5: 输出明文 M' 。



• SM9 标识密码算法— 公钥加密(解密算法)



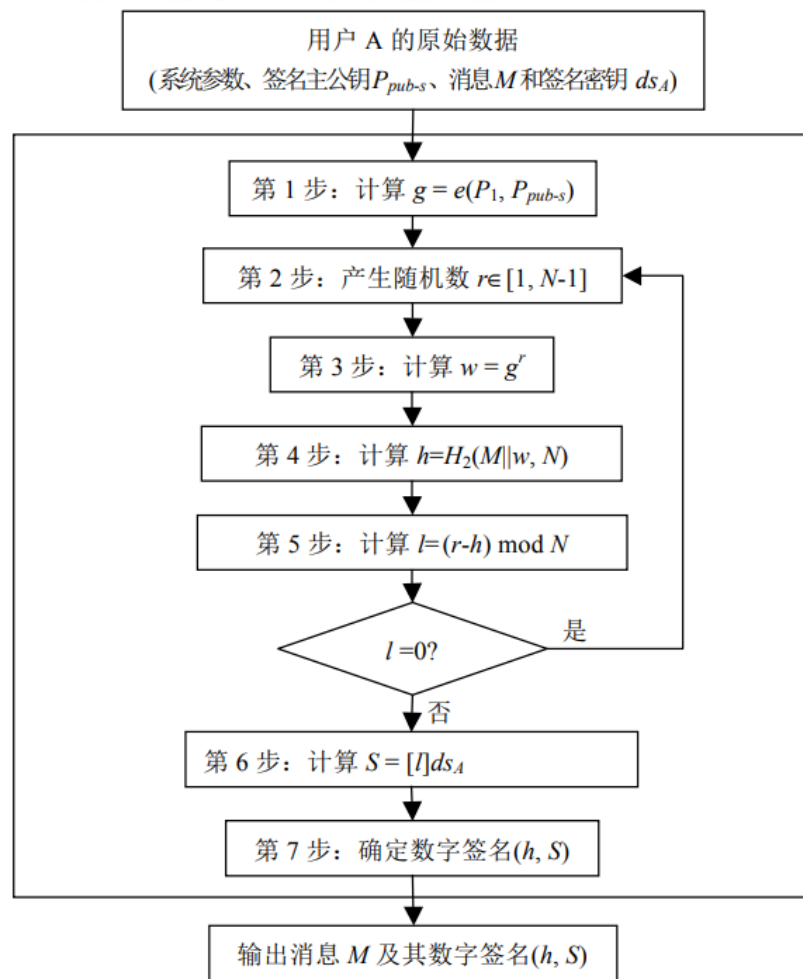


• SM9 标识密码算法—数字签名(签名生成)

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

运算步骤:

- A1: 计算群 \mathcal{G}_T 中的元素 $g = e(P_1, P_{pub-s})$;
- A2: 产生随机数 $r \in [1, N-1]$;
- A3: 计算群 \mathcal{G}_T 中的元素 $w = g^r$, 将 w 的数据类型转换为比特串
- A4: 计算整数 $h = H_2(M || w, N)$;
- A5: 计算整数 $l = (r - h) \bmod N$, 若 $l = 0$ 则返回 A2;
- A6: 计算群 \mathcal{G}_1 中的元素 $S = [l]ds_A$;
- A7: 消息 M 的签名为 (h, S) 。



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

B1: 检验 $h' \in [1, N-1]$ 是否成立，若不成立则验证不通过；

B2: 将 S' 的数据类型转换为椭圆曲线上的点，检验 $S' \in \mathcal{G}_1$ 是否成立，若不成立则验证不通过；

B3: 计算群 \mathcal{G}_T 中的元素 $g = e(P_1, P_{pub-s})$ ；

B4: 计算群 \mathcal{G}_T 中的元素 $t = g^{h'}$ ；

B5: 计算整数 $h_1 = H_1(ID_A || hid, N)$ ；

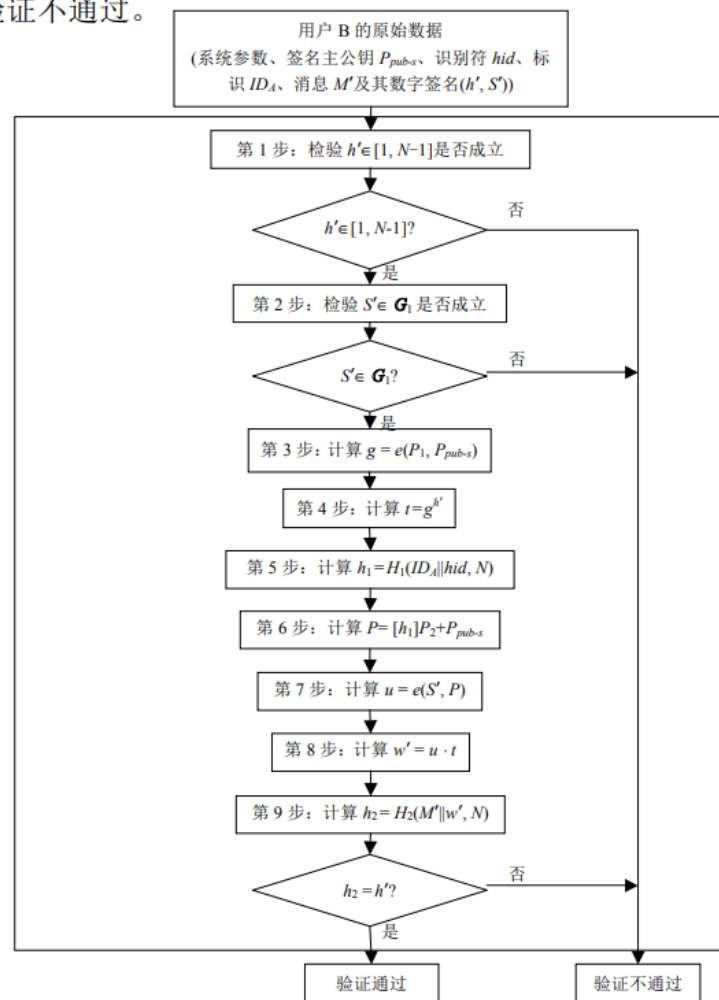
B6: 计算群 \mathcal{G}_2 中的元素 $P = [h_1]P_2 + P_{pub-s}$ ；

B7: 计算群 \mathcal{G}_T 中的元素 $u = e(S', P)$ ；

B8: 计算群 \mathcal{G}_T 中的元素 $w' = u \cdot t$ ，将 w' 的数据类型转换为比特串；

B9: 计算整数 $h_2 = H_2(M' || w', N)$ ，检验 $h_2 = h'$ 是否成立，若成立则验证通过；否则验证不通过。

• SM9 标识密码算法—数字签名(签名验证)





• SM9 标识密码算法—密钥交换

系统参数：同密钥封装和公钥加密算法

系统加密主密钥和用户加密密钥的产生：

KGC产生随机数 $ke \in [1, N-1]$ 作为加密主私钥，计算 \mathbb{G}_1 中的元素 $P_{pub-e} = [ke]P_1$ 作为加密主公钥，则加密主密钥对为 (ke, P_{pub-e}) 。KGC秘密保存 ke ，公开 P_{pub-e} 。

KGC选择并公开用一个字节表示的加密私钥生成函数识别符 hid 。

用户A和B的标识分别为 ID_A 和 ID_B 。为产生用户A的加密私钥 de_A ，KGC首先在有限域 F_N 上计算 $t_1 = H_1(ID_A || hid, N) + ke$ ，若 $t_1 = 0$ 则需重新产生加密主私钥，计算和公开加密主公钥，并更新已有用户的加密私钥；否则计算 $t_2 = ke \cdot t_1^{-1}$ ，然后计算 $de_A = [t_2]P_2$ 。为产生用户B的加密私钥 de_B ，KGC首先在有限域 F_N 上计算 $t_3 = H_1(ID_B || hid, N) + ke$ ，若 $t_3 = 0$ 则需重新产生加密主私钥，计算和公开加密主公钥，并更新已有用户的加密私钥；否则计算 $t_4 = ke \cdot t_3^{-1}$ ，然后计算 $de_B = [t_4]P_2$ 。



• SM9 标识密码算法—密钥交换

设用户 A 和 B 协商获得密钥数据的长度为 $klen$ 比特，用户 A 为发起方，用户 B 为响应方。
用户 A 和 B 双方为了获得相同的密钥，应实现如下运算步骤：

用户 A:

A1: 计算群 \mathcal{G}_1 中的元素 $Q_B = [H_1(ID_B || hid, N)]P_1 + P_{pub-e}$;

A2: 产生随机数 $r_A \in [1, N-1]$;

A3: 计算群 \mathcal{G}_1 中的元素 $R_A = [r_A]Q_B$;

A4: 将 R_A 发送给用户 B;

用户 B:

B1: 计算群 \mathcal{G}_1 中的元素 $Q_A = [H_1(ID_A || hid, N)]P_1 + P_{pub-e}$;

B2: 产生随机数 $r_B \in [1, N-1]$;

B3: 计算群 \mathcal{G}_1 中的元素 $R_B = [r_B]Q_A$;

B4: 验证 $R_A \in \mathcal{G}_1$ 是否成立，若不成立则协商失败；否则计算群 \mathcal{G}_T 中的元素 $g_1 = e(R_A, de_B)$, $g_2 = e(P_{pub-e}, P_2)^{r_B}$, $g_3 = g_1^{r_B}$ ，将 g_1, g_2, g_3 的数据类型转换为比特串；

B5: 把 R_A 和 R_B 的数据类型转换为比特串，计算 $SK_B = KDF(ID_A || ID_B || R_A || R_B || g_1 || g_2 || g_3, klen)$;

B6: (选项)计算 $S_B = Hash(0x82 || g_1 || Hash(g_2 || g_3 || ID_A || ID_B || R_A || R_B))$;

B7: 将 R_B 、(选项 S_B)发送给用户 A;



- SM9 标识密码算法—密钥交换

用户 A:

A5: 验证 $R_B \in \mathcal{G}_1$ 是否成立, 若不成立则协商失败; 否则计算群 \mathcal{G}_T 中的元素 $g_1' = e(P_{pub-e}, P_2)^{r_A}$,

$g_2' = e(R_B, de_A)$, $g_3' = (g_2')^{r_A}$, 将 g_1', g_2', g_3' 的数据类型转换为比特串;

A6: 把 R_A 和 R_B 的数据类型转换为比特串, (选项)计算 $S_1 = Hash(0x82 \| g_1' \| Hash(g_2' \| g_3' \| ID_A \| ID_B \| R_A \| R_B))$, 并检验 $S_1 = S_B$ 是否成立, 若等式不成立则从 B 到 A 的密钥确认失败;

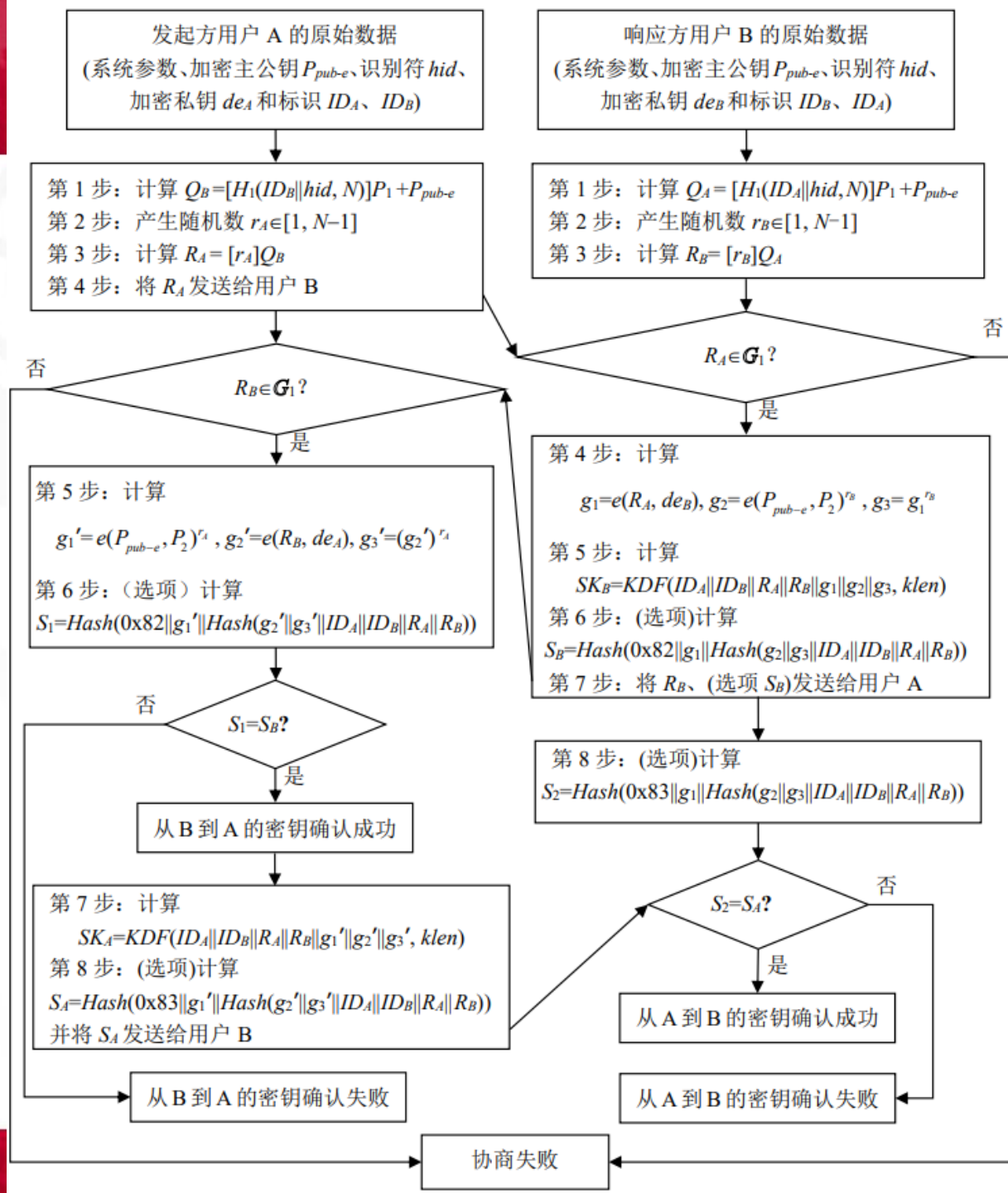
A7: 计算 $SK_A = KDF(ID_A \| ID_B \| R_A \| R_B \| g_1' \| g_2' \| g_3', klen)$;

A8: (选项)计算 $S_A = Hash(0x83 \| g_1' \| Hash(g_2' \| g_3' \| ID_A \| ID_B \| R_A \| R_B))$, 并将 S_A 发送给用户 B。

用户 B:

B8: (选项)计算 $S_2 = Hash(0x83 \| g_1 \| Hash(g_2 \| g_3 \| ID_A \| ID_B \| R_A \| R_B))$, 并检验 $S_2 = S_A$ 是否成立, 若等式不成立则从 A 到 B 的密钥确认失败。

- SM9 标识密码算法——密钥交换





華東師範大學
EAST CHINA NORMAL UNIVERSITY

Thanks !