信息安全研究 第2卷 第11期 2016年11月
Journal of Information Security Research Vol.2 No.11 Nov. 2016

SM2 椭圆曲线公钥密码算法综述

汪朝晖[1] 张振峰[2]
[1](北京华大信安科技有限公司 北京 100015)
[2](中国科学院软件研究所可信计算与信息保障实验室 北京 100190)
(wangzh@istecc.com)

Overview on Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves

Wang Zhaohui[1] and Zhang Zhenfeng[2]
[1](*Beijing Huada Infosec Technology Co., ltd, Beijing* 100015)
[2](*Laboratory of Trusted Computing and Information Assurance, Institute of Software, Chinese Academy of Sciences, Beijing* 100190)

Abstract Public key cryptographic algorithm SM2 based on elliptic curves (SM2 algorithm for abbreviation) was firstly issued in December 2010, had become the Chinese commercial cryptographic standard (GM/T 0003—2012) in 2012, and had become the Chinese national cryptographic standard (GB/T 32918—2016) in 2016. This paper briefly describe the development background of SM2 algorithm, describe SM2 algorithm in details, introduce the researches on its security, and evaluate its implementation efficiencies. All the researches on SM2 algorithm so far indicate that the provable securities of SM2 algorithm reach the supreme levels of public key cryptographic algorithms' securities, and its implementation efficiencies are equivalent to or slightly superior to those similar elliptic curve cryptographic algorithms in some international standards.

Key words public key cryptographic algorithm; elliptic curve cryptography; digital signature; key exchange; encryption; decryption; SM2 algorithm

摘 要 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). 简介 SM2 算法的研制背景,详细描述 SM2 算法,介绍 SM2 算法安全性研究情况,并评估其实现效率. 迄今为止与 SM2 算法相关的研究表明,SM2 算法的可证安全性达到了公钥密码算法的最高安全级别,其实现效率相当于或略优于一些国际标准的同类椭圆曲线密码算法.

关键词 公钥密码算法;椭圆曲线密码算法;数字签名;密钥交换;加密;解密;SM2 算法

中图法分类号 TP309

SM2 椭圆曲线公钥密码算法(elliptic curve cryptography, ECC)是我国公钥密码算法标准. SM2 算法的主要内容包括 3 部分:数字签名算法;密钥交换协议和公钥加密算法(下文分别称为 SM2 签名算法;SM2 密钥交换协议和 SM2 加密算法).

Koblitz[1]和 Miller[2]各自独立地提出将椭圆曲线应用于公钥密码系统. ECC 所基于的椭圆曲线性质如下:1)有限域上椭圆曲线在点加运算下

收稿日期:2016-10-25

构成有限交换群，且其阶与基域规模相近；2）类似于有限域乘法群中的乘幂运算，椭圆曲线多倍点运算构成一个单向函数.

在多倍点运算（也称点乘运算）中，已知多倍点与基点，求解倍数的问题称为椭圆曲线离散对数问题（elliptic curve discrete logarithm problem，ECDLP）. 对于一般 ECDLP，目前只存在指数级计算复杂度的求解方法. 与大数分解问题（integer factorization problem，IFP）及有限域上离散对数问题（discrete logarithm problem，DLP）相比，ECDLP 的求解难度要大得多. 因此，在相同安全程度要求下，ECC 较其他公钥密码算法所需的密钥规模要小得多.

表 1 列出了这几种公钥密码算法在同等安全强度下的私钥位长比较.

表 1　RSA（或 DSA）和 ECC 的私钥位长比较

破解运算量（MIPS 年）	RSA（或 DSA）私钥位长	ECC 私钥位长	RSA（或 DSA）/ECC 私钥位长
10^4	512	106	5∶1
10^8	768	132	6∶1
10^{11}	1024	160	7∶1
10^{20}	2048	210	10∶1
10^{78}	21000	600	35∶1

由于在相同安全强度下 ECC 比 RSA 的私钥位长及系统参数小得多，这意味着应用 ECC 所需的存储空间要小得多，传输所用的带宽要求更低，硬件实现 ECC 所需逻辑电路的逻辑门数要较 RSA 少得多，功耗更低. 这使得 ECC 比 RSA 更适合实现到资源严重受限制的设备中，如低功耗要求的移动通信设备、无线通信设备和智能卡等.

ECC 的优势使其成为了最具发展潜力和应用前景的公钥密码算法，至 2000 年国际上已有多个国家和行业组织将 ECC 采纳为公钥密码算法标准. 在此背景下，我国从 2001 年开始组织研究自主知识产权的 ECC，通过运用国际密码学界公认的公钥密码算法设计及安全性分析理论和方法，在吸收国内外已有 ECC 研究成果的基础上，于 2004 研制完成了 SM2 算法. SM2 算法于 2010 年 12 月首次公开发布，2012 年 3 月成为中国商用密码标准（标准号为 GM/T 0003—2012），2016 年 8 月成为中国国家密码标准（标准号为 GB/T 32918—2016）.

SM2 算法正在我国商用密码行业进行大规模应用和推广，2011 年 3 月中国人民银行发布了《中国金融集成电路（IC）卡规范》（简称 PBOC3.0），PBOC3.0 采用了 SM2 算法以增强金融 IC 卡应用的安全性，以 PBOC3.0 为参考规范的非金融类应用也基本采用 SM2 算法. 国际可信计算组织（TCG）发布的 TPM 2.0 规范[3-4]采纳了 SM2 算法. 2016 年 10 月，ISO/IEC SC27 会议通过了 SM2 算法标准草案，SM2 算法进入 ISO 14888-3 正式文本阶段. 这些 SM2 算法成为国际标准历程中的重要事件，将进一步促进 SM2 算法的应用和推广.

1　SM2 算法描述

SM2 算法包括数字签名算法、密钥交换协议、公钥加密算法和系统参数 4 部分.

1.1　SM2 系统参数

ECC 的系统参数是有限域上的椭圆曲线，包括：有限域 F_q 的规模 q；定义椭圆曲线 $E(F_q)$ 方程的 2 个元素 $a,b\in F_q$；$E(F_q)$ 上的基点 $G=(x_G, y_G)$（$G\neq O$），其中 x_G 和 y_G 是 F_q 中的 2 个元素；G 的阶 n 及其他可选项（如 n 的余因子 h 等）.

记 SM2 算法中使用的密码杂凑算法为 $H_v(\ )$，其输出是位长恰为 v 的杂凑值，SM2 算法目前版本中 v 只取为 256. SM2 算法的系统参数为 256 b 素数域上的椭圆曲线，具体定义请参考文献[5]的第 5 部分.

1.2　SM2 数字签名算法

数字签名算法由一个签名者对数据产生数字签名，并由一个验证者验证签名的可靠性. 每个签名者有一个公钥和一个私钥，其中私钥用于产生签名，验证者用签名者的公钥验证签名.

SM2 数字签名算法中，作为签名者的用户 A 的密钥对包括其私钥 d_A 和公钥 $P_A=[d_A]G=(x_A, y_A)$，用户 A 具有位长为 $entlen_A$ 的可辨别标识 ID_A，记 $ENTL_A$ 是由整数 $entlen_A$ 转换而成的 2 B 数据，签名者和验证者都需要用密码杂凑算法求得用户 A 的杂凑值 $Z_A=H_{256}(ENTL_A\|ID_A\|a\|b\|x_G\|y_G\|x_A\|y_A)$. SM2 数字签名算法规定 H_{256} 为 SM3 密码杂凑算法.

1.2.1　数字签名的生成算法

设待签名的消息为 M，为了获取消息 M 的数

字签名(r,s)，作为签名者的用户A应实现以下运算步骤：

A1. 置$\bar{M}=Z_A\|M$；

A2. 计算$e=H_v(\bar{M})$，将e的数据类型转换为整数；

A3. 用随机数发生器产生随机数$k\in[1,n-1]$；

A4. 计算椭圆曲线点$(x_1,y_1)=[k]G$，将x_1的数据类型转换为整数；

A5. 计算$r=(e+x_1) \bmod n$，若$r=0$或$r+k=n$，则返回A3；

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

A7. 将r,s的数据类型转换为字节串，消息M的签名为(r,s).

数字签名生成流程如图1所示：

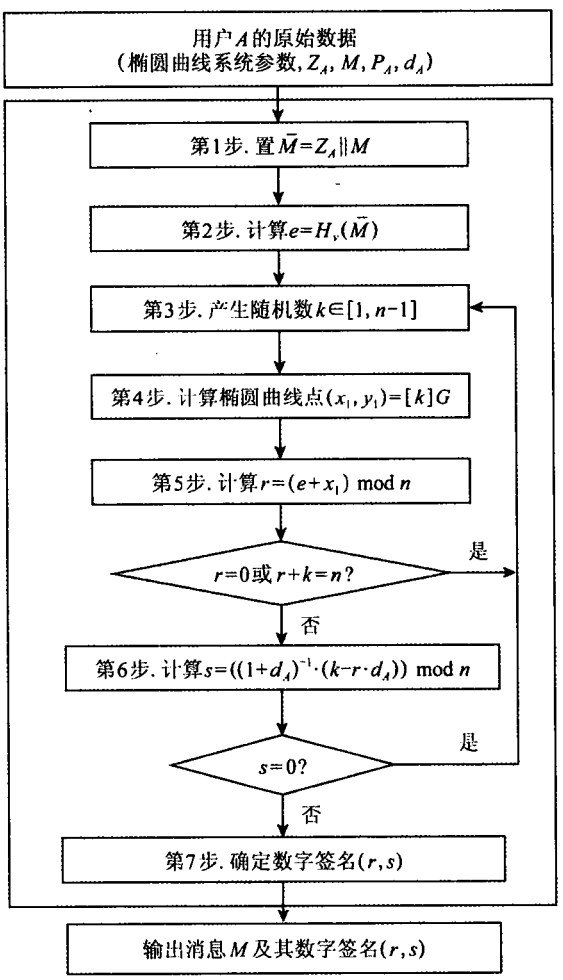

图1 SM2数字签名生成流程

1.2.2 数字签名的验证算法

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

B1. 检验$r'\in[1,n-1]$是否成立，若不成立则验证不通过；

B2. 检验$s'\in[1,n-1]$是否成立，若不成立则验证不通过；

B3. 置$\bar{M}'=Z_A\|M'$；

B4. 计算$e'=H_v(\bar{M}')$，将e'的数据类型转换为整数；

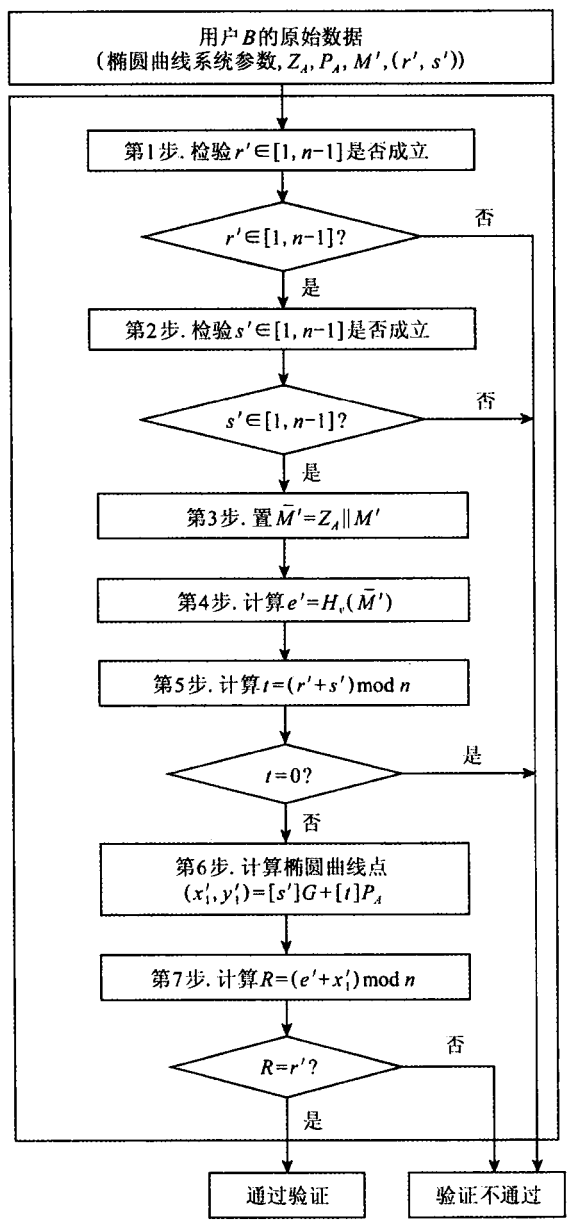

图2 SM2数字签名验证流程

B5. 将 r',s'的数据类型转换为整数,计算 $t=(r'+s') \bmod n$,若 $t=0$,则验证不通过;

B6. 计算椭圆曲线点$(x_1', y_1')=[s']G+[t]P_A$;

B7. 将 x_1' 的数据类型转换为整数,计算 $R=(e'+x_1') \bmod n$,检验 $R=r'$是否成立,若成立则验证通过;否则验证不通过.

SM2 数字签名验证流程如图 2 所示.

1.3 SM2 密钥交换协议

密钥交换协议是 2 个用户 A 和 B 通过交互的信息传递,用各自的私钥和对方的公钥来商定一个只有他们知道的秘密密钥. 这个共享的秘密密钥通常用在某个对称密码算法中.

SM2 密钥交换协议中,用户 A 的密钥对包括其私钥 d_A 和公钥 $P_A=[d_A]G=(x_A, y_A)$,用户 B 的密钥对包括其私钥 d_B 和公钥 $P_B=[d_B]G=(x_B, y_B)$. 用户 A 具有位长为 $entlen_A$ 的可辨别标识 ID_A,记 $ENTL_A$ 是由整数 $entlen_A$ 转换而成的 2 B 数据;用户 B 具有位长为 $entlen_B$ 的可辨别标识 ID_B,记 $ENTL_B$ 是由整数 $entlen_B$ 转换而成的 2 B 数据. A,B 双方都需要用密码杂凑算法求得用户 A 的杂凑值 $Z_A=H_{256}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_A \| y_A)$和用户 B 的杂凑值 $Z_B=H_{256}(ENTL_B \| ID_B \| a \| b \| x_G \| y_G \| x_B \| y_B)$.

1.3.1 密钥派生函数

密钥派生函数的作用是从一个共享的秘密比特串中派生出密钥数据. 在密钥协商过程中,密钥派生函数作用在密钥交换所获共享的秘密比特串上,从中产生所需的会话密钥或进一步加密所需的密钥数据. 密钥派生函数需要调用密码杂凑算法 $H_v(\ )$.

SM2 密钥交换协议中使用的密钥派生函数 $KDF(Z, klen)$如下.

输入:比特串 Z、整数 $klen$(表示要获得的密钥数据的位长,要求该值小于$(2^{32}-1)v$);

输出:位长为 $klen$ 的密钥数据比特串 K.

1) 初始化一个 32 b 构成的计数器 $ct=$ 0x00000001;

2) 对 i 从 1 到 $\lceil klen/v \rceil$ 执行:

2.1) 计算 $Ha_i=H_v(Z \| ct)$;

2.2) $ct++$;

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

4) 令 $K=Ha_1 \| Ha_2 \| \cdots \| Ha_{\lceil klen/v \rceil-1} \| Ha!_{\lceil klen/v \rceil}$.

1.3.2 密钥交换协议

设用户 A 和 B 协商获得密钥数据的位长为 $klen$,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.

记 $w=\lceil (\lceil \mathrm{lb}(n) \rceil /2) \rceil -1$.

用户 A:

A1. 用随机数发生器产生随机数 $r_A \in [1, n-1]$;

A2. 计算椭圆曲线点 $R_A=[r_A]G=(x_1, y_1)$;

A3. 将 R_A 发送给用户 B;

用户 B:

B1. 用随机数发生器产生随机数 $r_B \in [1, n-1]$;

B2. 计算椭圆曲线点 $R_B=[r_B]G=(x_2, y_2)$;

B3. 从 R_B 中取出域元素 x_2,将 x_2 的数据类型转换为整数,计算 $\bar{x}_2=2^w+(x_2 \& (2^w-1))$;

B4. 计算 $t_B=(d_B+\bar{x}_2 \cdot r_B) \bmod n$;

B5. 验证 R_A 是否满足椭圆曲线方程,若不满足则协商失败;否则从 R_A 中取出域元素 x_1,将 x_1 的数据类型转换为整数,计算 $\bar{x}_1=2^w+(x_1 \& (2^w-1))$;

B6. 计算椭圆曲线点 $V=[h \cdot t_B](P_A+[\bar{x}_1]R_A)=(x_V, y_V)$,若 V 是无穷远点,则 B 协商失败;否则将 x_V,y_V 的数据类型转换为比特串;

B7. 计算 $K_B=KDF(x_V \| y_V \| Z_A \| Z_B, klen)$;

B8. (选项)将 R_A 的坐标 x_1,y_1 和 R_B 的坐标 x_2,y_2 的数据类型转换为比特串,计算 $S_B=Hash(0x02 \| y_V \| Hash(x_V \| Z_A \| Z_B \| x_1 \| y_1 \| x_2 \| y_2))$;

B9. 将 R_B,(选项 S_B)发送给用户 A;

用户 A:

A4. 从 R_A 中取出域元素 x_1,计算 $\bar{x}_1=2^w+(x_1 \& (2^w-1))$;

A5. 计算 $t_A=(d_A+\bar{x}_1 \cdot r_A) \bmod n$;

A6. 验证 R_B 是否满足椭圆曲线方程,若不满足则协商失败;否则从 R_B 中取出域元素 x_2,将 x_2

的数据类型转换为整数，计算 $\bar{x}_2 = 2^w + (x_2 \& (2^w - 1))$；

A7. 计算椭圆曲线点 $U = [h \cdot t_A](P_B + [\bar{x}_2]R_B) = (x_U, y_U)$，若 U 是无穷远点，则 A 协商失败；否则将 x_U, y_U 的数据类型转换为比特串；

A8. 计算 $K_A = KDF(x_U \| y_U \| Z_A \| Z_B, klen)$；

A9.（选项）R_A 的坐标 x_1, y_1 和 R_B 的坐标 x_2, y_2 的数据类型转换为比特串，计算 $S_1 = Hash(0x02 \| y_U \| Hash(x_U \| Z_A \| Z_B \| x_1 \| y_1 \| x_2 \| y_2))$，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；

A10.（选项）计算 $S_A = Hash(0x03 \| y_U \| Hash(x_U \| Z_A \| Z_B \| x_1 \| y_1 \| x_2 \| y_2))$，并将 S_A 发送给用户 B.

用户 B：

B10.（选项）计算 $S_2 = Hash(0x03 \| y_V \| Hash(x_V \| Z_A \| Z_B \| x_1 \| y_1 \| x_2 \| y_2))$，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.

SM2 密钥交换协议流程如图 3 所示：

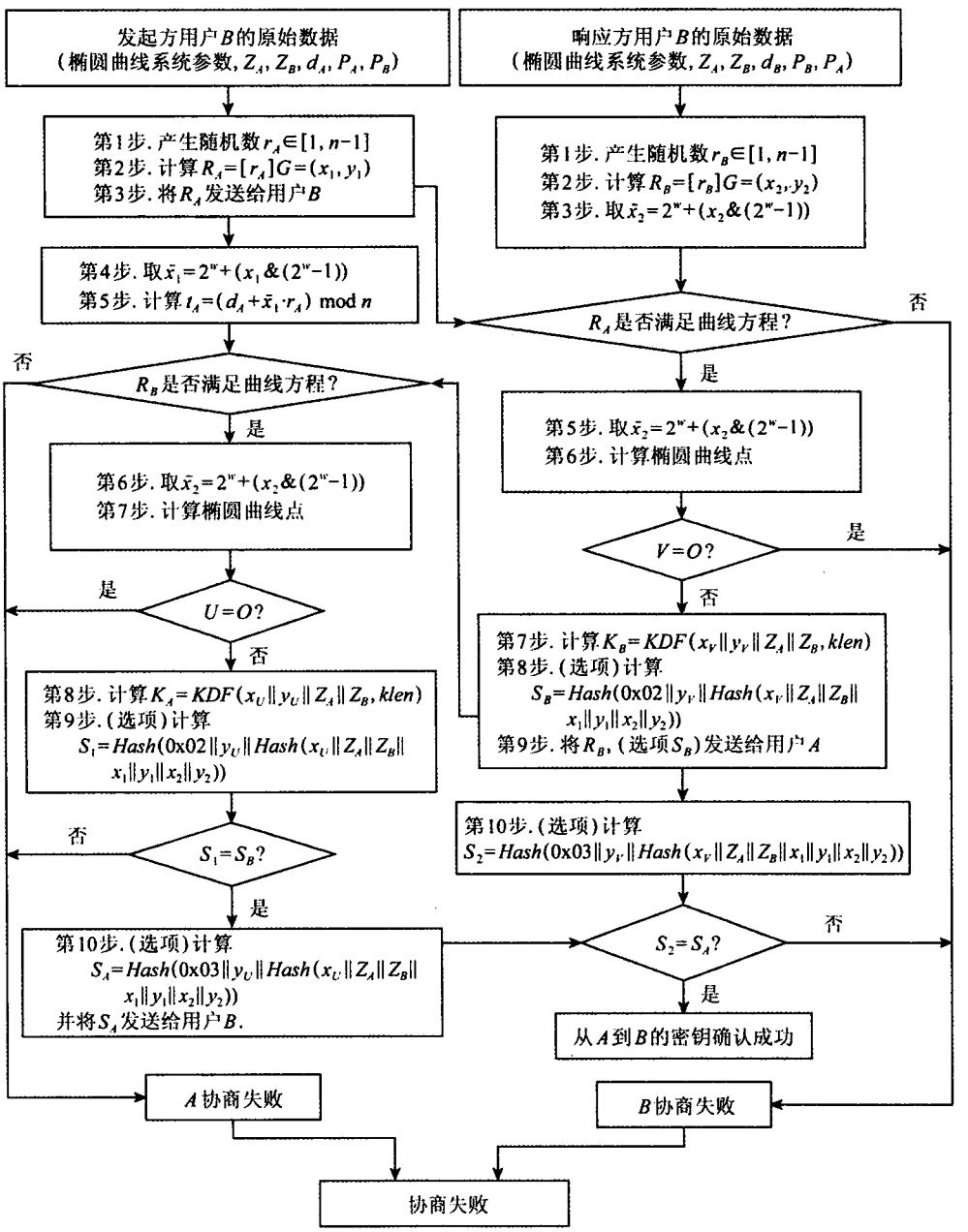

图 3 SM2 密钥交换协议流程

1.4 SM2 公钥加密算法

公钥加密算法规定发送者用接收者的公钥将消息加密成密文，接收者用自己的私钥对收到的密文进行解密还原成原始消息.

SM2 公钥加密算法中，用户 B 的密钥对包括其私钥 d_B 和公钥 $P_B=[d_B]G$.

1.4.1 密钥派生函数

SM2 公钥加密算法也需要使用密钥派生函数.

密钥派生函数 $KDF(Z,klen)$ 如下.

输入：比特串 Z，整数 $klen$（表示要获得的密钥数据的位长，要求该值小于 $(2^{32}-1)v$）；

输出：位长为 $klen$ 的密钥数据比特串 K.

1）初始化一个 32 b 构成的计数器 $ct=$ 0x00000001；

2）对 i 从 1 到 $\lceil klen/v \rceil$ 执行：

2.1）计算 $Ha_i=H_v(Z\|ct)$；

2.2）$ct++$；

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

4）令 $K=Ha_1\|Ha_2\|\cdots\|Ha_{\lceil klen/v \rceil-1}\|Ha!_{\lceil klen/v \rceil}$.

1.4.2 加密算法

设需要加密的消息为比特串 M，$klen$ 为 M 的位长. 为了对明文 M 进行加密，作为加密者的用户 A 应实现以下运算步骤：

A1. 用随机数发生器产生随机数 $k\in[1,n-1]$；

A2. 计算椭圆曲线点 $C_1=[k]G=(x_1,y_1)$，将 C_1 的数据类型转换为比特串；

A3. 计算椭圆曲线点 $S=[h]P_B$，若 S 是无穷远点，则报错并退出；

A4. 计算椭圆曲线点 $[k]P_B=(x_2,y_2)$，将坐标 x_2，y_2 的数据类型转换为比特串；

A5. 计算 $t=KDF(x_2\|y_2,klen)$，若 t 为全 0 比特串，则返回 A1；

A6. 计算 $C_2=M\oplus t$；

A7. 计算 $C_3=Hash(x_2\|M\|y_2)$；

A8. 输出密文 $C=C_1\|C_3\|C_2$.

SM2 加密流程如图 4 所示：

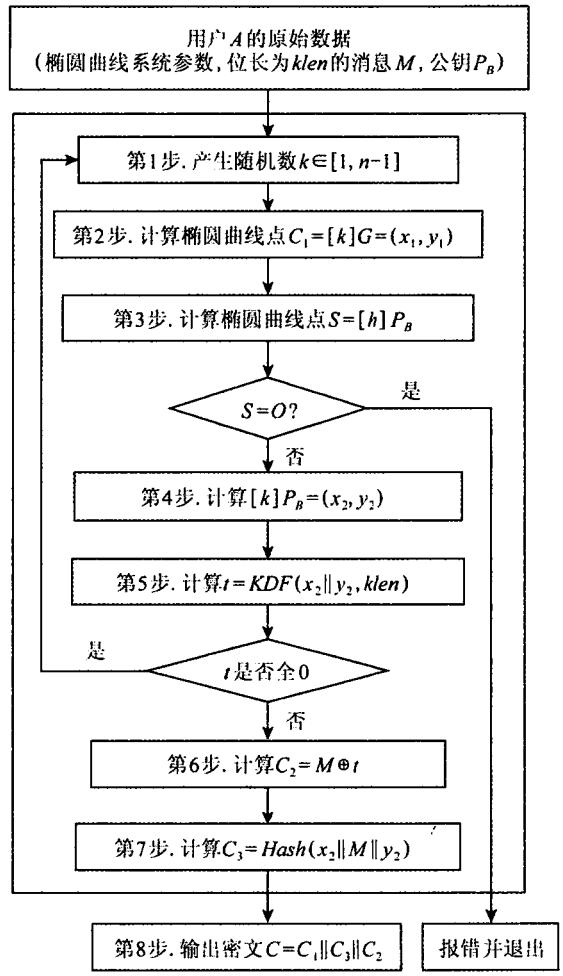

图 4 SM2 加密流程

1.4.3 解密算法

设 $klen$ 为密文中 C_2 的位长.

为了对密文 $C=C_1\|C_3\|C_2$ 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 是否满足椭圆曲线方程，若不满足则报错并退出；

B2. 计算椭圆曲线点 $S=[h]C_1$，若 S 是无穷远点，则报错并退出；

B3. 计算 $[d_B]C_1=(x_2,y_2)$，将坐标 x_2，y_2 的数据类型转换为比特串；

B4. 计算 $t=KDF(x_2\|y_2,klen)$，若 t 为全 0 比特串，则报错并退出；

B5. 从 C 中取出比特串 C_2，计算 $M'=C_2\oplus t$；

B6. 计算 $u=Hash(x_2\|M'\|y_2)$，从 C 中取出比特串 C_3，若 $u\neq C_3$，则报错并退出；

B7. 输出明文 M'.

SM2 解密流程如图 5 所示：

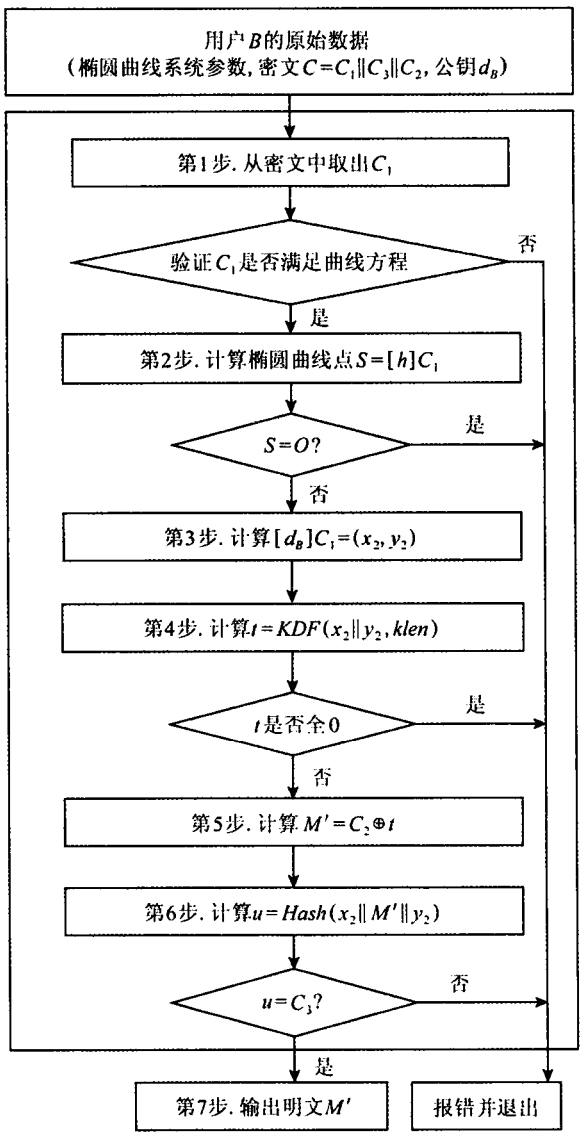

图 5 SM2 解密流程

2 SM2 算法的安全性

2.1 椭圆曲线参数的安全性

已知椭圆曲线 $E(F_q)$，阶为 n 的点 $G\in E(F_q)$ 及 $Q\in\langle G\rangle$，ECDLP 是指确定整数 $k\in[0,n-1]$，使得 $Q=[k]G$ 成立. ECDLP 关系到 ECC 的安全，因此 SM2 算法必须选择安全的椭圆曲线. 若某椭圆曲线存在优于 $n^{1/2}$ 级（n 是基点的阶）计算复杂度的攻击方法，则称此曲线为弱椭圆曲线. 如：F_a 上的超奇异曲线（有限域 F_a 的特征整除 $q+1-\#E(F_q)$）和 F_p 上的异常（anomalous）曲线（$\#E(F_p)=p$）都是弱椭圆曲线.

2.1.1 ECDLP 求解方法

如何判断一条椭圆曲线是安全的还是弱的，需要先了解 ECDLP 的攻击方法. ECDLP 现有的攻击方法有：

1) Pohlig-Hellman 方法[6]. 设 l 是 n 的最大素因子，则算法复杂度为 $O(l^{1/2})$.

2) BSGS 方法. 时间复杂度与空间复杂度均为 $(\pi n/2)^{1/2}$.

3) Pollard 方法[7]. 算法复杂度为 $(\pi n/2)^{1/2}$.

4) 并行 Pollard 方法. 设 r 为并行处理器个数，算法复杂度降至 $(\pi n/2)^{1/2}/r$.

5) MOV 方法[8]. 把超奇异椭圆曲线及具有相似性质的曲线的 ECDLP 降到 F_q 的小扩域上的离散对数问题（亚指数级计算复杂度算法）.

6) 异常曲线离散对数求解方法. 对异常曲线（$\#E(F_p)=p$ 的曲线）的有效攻击方法（多项式级计算复杂度算法）.

7) GHS 方法[9]. 利用 Weil 下降技术求解扩张次数为合数的二元扩域上椭圆曲线离散对数问题，将 ECDLP 转化为超椭圆曲线离散对数问题，而求解高亏格的超椭圆曲线离散对数存在亚指数级计算复杂度算法.

对于一般曲线的离散对数问题，目前的求解方法均为指数级计算复杂度，未发现有效的亚指数级计算复杂度的一般攻击方法；而对于某些特殊曲线的离散对数问题，存在多项式级计算复杂度或者亚指数级计算复杂度算法.

选择曲线时，应避免使用易受上述方法攻击的密码学意义上的弱椭圆曲线.

2.1.2 安全椭圆曲线满足的条件

1) 抗 MOV 攻击条件

Menezes, Okamoto, Vanstone[8] 的约化攻击将有限域 F_q 上的椭圆曲线离散对数问题约化为 F_{q^B}（$B>1$）上的离散对数问题. 该攻击方法只有在 B 较小时是实用的，大多数椭圆曲线不符合这种情况. 抗 MOV 攻击条件确保一条椭圆曲线不易受此约化方法攻击. 多数 F_q 上的椭圆曲线确实满足抗 MOV 攻击条件.

在验证抗 MOV 攻击条件之前，必须选择一个 MOV 阈，它是使得求取 F_{q^B} 上的离散对数问题

至少与求取 F_q 上的椭圆曲线离散对数问题同样难的一个正整数 B. 对于 $q>2^{191}$ 的标准，要求 $B\geqslant 27$. 选择 $B\geqslant 27$ 也限制了对非超奇异椭圆曲线的选取.

2) 抗异常曲线攻击条件

设 $E(F_p)$ 为定义在素域 F_p 上的椭圆曲线，若 $\#E(F_p)=p$，则称椭圆曲线 $E(F_p)$ 为异常曲线. Smart[10]，Satoh 和 Araki[11] 证明可在多项式时间内求解异常曲线的离散对数. 抗异常曲线攻击条件为 $\#E(F_p)\neq p$，满足此条件确保椭圆曲线不受异常曲线攻击. F_p 上的绝大多数椭圆曲线确实满足抗异常曲线攻击条件.

3) 其他条件

为避免 Pohlig-Hellman 方法和 Pollard 方法的攻击，基点的阶 n 必须是一个足够大的素数；为避免 GHS 方法的攻击，F_2m 中的 m 应该选择素数.

SM2 算法采用的椭圆曲线参数经检测完全满足上述安全性条件，其上的 ECDLP 不存在优于 $n^{1/2}$ 级计算复杂度的攻击方法，n 为 256 b 的素数，具有足够的安全位长. 判断椭圆曲线是否安全的具体算法请参考文献[5]的第 1 部分.

2.2 SM2 数字签名算法的安全性

针对数字签名算法的最强攻击行为是自主选择消息攻击(adaptively chosen-message attacks)，攻击者可以访问签名预言机(signing oracle)，除攻击者要伪造签名的消息外，他可以任意选择消息进行签名而获得有效的消息/签名对. 攻击者如果达到以下目标之一，则称数字签名算法被攻破.

1) 完全攻破(total break)：攻击者获得签名私钥，可以对任意消息伪造签名，这是最严重的攻破；

2) 一般性伪造(universal forgery)：攻击者建立一个有效的算法来模仿签名，模仿签名的成功率足够高；

3) 存在性伪造(existential forgery)：也称随机消息签名伪造，攻击者利用已有的消息/签名对，可以生成新的消息/签名对，新的消息与原有消息/签名对具有相关性，攻击者不能自主选择.

上述 3 个攻击目标中，存在性伪造是最低的，对于一个数字签名算法，如果攻击者采用最强的攻击行为，仍然不能达到最低的攻击目标，则该数字签名算法是安全的. Goldwasser 等人[12-13] 提出的自主选择消息攻击下存在性不可伪造(existential unforgeability under adaptively chosen-message attacks，EUF-CMA)已经成为评估数字签名算法安全性的一个标准概念.

Menezes 和 Smart[14] 提出了针对数字签名算法的密钥替换攻击，攻击者拥有公钥 pk 以及该公钥对应的消息/签名对 (m,s)，试图生成另一个公钥 pk'，使得用 pk' 验证 (m,s) 仍然是有效的. 有多项研究[14-17] 已经表明，EUF-CMA 安全的数字签名算法，仍然可能被成功实施密钥替换攻击.

SM2 数字签名算法属于广义 ELGamal 数字签名算法范围，此类数字签名算法的 EUF-CMA 分析和证明已有成熟的模型和方法，包括 GGM(generic group model)[18] 和 ROM(random oracle model)[19].

针对密钥替换攻击，SM2 数字签名算法采取的防御方法是将签名者 ID、公钥和源消息一起 Hash，在 Hash 算法安全的前提下，可以抵抗密钥替换攻击. 文献[20]对 SM2 数字签名算法的 EUF-CMA 和抵抗密钥替换攻击的安全性给出了分析和证明.

2.3 SM2 密钥交换协议的安全性

SM2 密钥交换协议是以 ECDH 密钥交换协议为基础进行设计的. 由于 ECDH 密钥交换协议安全性不高，譬如不能抵抗中间人攻击，SM2 密钥交换协议增加了针对这些安全威胁的防御方法，将信息认证方法加入到双方的交换过程中，具体做法是密钥计算过程中增加了代表双方身份的固定公钥和用户 ID 等参数，由于中间人不掌握固定公钥所对应的私钥且不能冒用用户 ID，因此无法攻击 SM2 密钥交换协议. 关于 SM2 密钥交换协议的安全性分析和证明可以参考文献[21-23].

2.4 SM2 公钥加密算法的安全性

攻击者对公钥加密算法的攻击行为包括：

1) 选择明文攻击(chosen plaintext attack，CPA). 攻击者可以访问加密预言机(encryption oracle)，获得一定的明文/密文对，但他不能访问解密预言机(decryption oracle)，攻击者根据所掌握的信息和资源对他想破解的密文给出一个答案.

2) 选择密文攻击(chosen ciphertext attack，CCA1). 攻击者可以访问加密预言机和解密预言机，但在获得一定的明文/密文对后，不能再访问解

密预言机了，攻击者根据所掌握的信息和资源对他想破解的密文给出一个答案.

3) 自主选择密文攻击(adaptively chosen ciphertext attack, CCA2). 攻击者任何时候都可以访问加密预言机和解密预言机，唯一限制是不能直接将其想破解的密文输入解密预言机进行解密，攻击者根据所掌握的信息和资源对他想破解的密文给出一个答案.

显然，上述3种攻击行为中CCA2是最强的.

公钥加密算法的安全性体现在密文所具备的一些安全属性，这些属性包括：

1) 单向性(one-wayness, OW)，攻击者在不拥有私钥的前提下，不能计算出任何密文所对应的明文；

2) 不可区分性(indistinguishability, IND)[24]，攻击者选择2个不同的明文 m_1 和 m_2 输入加密预言机，加密预言机随机选择其中一个明文加密并返回密文 c，攻击者无法以明显区别于1/2的概率正确判断 c 为 m_1 或 m_2 的密文；

3) 不可延展性(non-malleability, NM)[25]，攻击者无法通过密文 c(对应明文为 m)构造出另一合法密文 c'(对应明文为 m')，使得 m 和 m' 之间存在某种有意义的关系(即有利于破解的关系).

公钥加密算法的安全性定义为：在攻击者的某种攻击行为下密文具备的某种安全属性. 如：IND-CCA2是指自主选择密文攻击下密文具备不可区分性，NM-CCA2是指自主选择密文攻击下密文具备不可延展性. 事实上IND-CCA2和NM-CCA2是等价的，为国际密码学界公认的公钥密码算法的最高级别的安全性.

SM2公钥加密算法是基于广义ELGamal加密算法进行设计的，但广义ELGamal加密算法的安全性级别不高，达不到IND-CCA2的安全性. 对公钥加密算法进行安全性增强，以使其达到IND-CCA2的方法比较多，可以概括为以下几类：

1) OAEP方法[26]. OAEP实际上是一种增强的对明文信息的Padding规则，比一般RSA算法的Padding规则更安全，比较适用于增强RSA算法的安全性.

2) 签名加密方法[27]. 对每次加密所涉及的秘密信息或者是密文本身签名，使得攻击者无法通过一密文构造相关联的另一合法密文，而解密预言机一旦遇到非法密文则拒绝输出任何信息，从而抵抗CCA2攻击.

3) 混合加密方法. 将公钥密码和对称密码结合起来，譬如采用密钥封装机制(key encapsulate mechanism, KEM)将对称密钥用公钥密码密文的形式封装起来，真正对明文加密的是使用该对称密钥的对称加密算法，利用公钥加密和对称加密共同生成密文的认证信息，解密预言机若验证认证信息有误则拒绝输出任何信息，从而抵抗CCA2攻击，比较有代表性的混合加密方法是DHAES[28].

4) 使用Hash函数. 通过安全Hash函数产生和验证MAC，对公钥密码算法中涉及的秘密信息和明文信息进行验证，一旦解密预言机发现验证MAC有误，则拒绝输出任何信息，从而抵抗CCA2攻击.

SM2公钥加密算法选择了方法4)以增强安全性，其密文包括 $C_1, C_2, C_3, C_3 = Hash(x_2 \| M \| y_2)$ 就是MAC，计算 C_3 所使用的密钥 (x_2, y_2) 是根据一次性的秘密随机数据进行DH密钥协商生成的，由于在解密时需要验证 C_3 的正确性，使得CCA2攻击者只能通过访问加密预言机获得有效的密文，除此之外不能获得或伪造出任何有效的密文，从而保障了SM2公钥加密算法抵抗CCA2的安全性.

SM2公钥加密算法的安全性为IND-CCA2，这可以用文献[29]中所使用的标准模型(standard model)和方法进行证明，也可以用ROM对SM2公钥加密算法的安全性进行证明.

3 SM2算法的实现效率

ECC运算过程中最耗时的运算是椭圆曲线点乘，无论是软件、FPGA还是集成电路实现，点乘运算占据整个算法运算时间的比例一般超过80%. 此外，算法中如果存在有限域上元素的求逆运算，也将占用不可忽略的时间比例. 故一般通过计算点乘运算和求逆运算的次数，就能大致评估ECC的实现效率.

基于上述判断，可以将SM2算法与国际上同类的ECC算法进行实现效率对比分析：1)SM2数字签名算法的实现效率与ECDSA相当，但SM2数字签名算法中求逆运算 $(1+d_A)^{-1} \bmod n$ 是可

以预计算的(生成签名私钥 d_A 时,应检查$(1+d_A)$ mod n 不为 0),ECDSA 中的求逆运算则不能,所以通过预计算可以使 SM2 数字签名算法实现效率略高于 ECDSA;2)SM2 密钥交换协议的实现效率与 ECMQV 相当;3)SM2 公钥加密算法的实现效率与 ECIES 相当.

上文提到 ECC 较 RSA 在实现效率方面有明显优势,尤其适合实现到智能 IC 卡芯片等资源受限的环境中.作者在一款智能 IC 卡芯片中,验证了这一判断:采用相同的硬件协处理器来实现 SM2 数字签名算法和 RSA 签名算法,该协处理器的核心运算部件是 64 b×64 b 的乘法器,运算数据暂存于双端口 SDRAM 中,运行于 30 MHz 时钟频率下的实测性能为:SM2 签名 118 次/s;RSA-1024 签名、非 CRT 模式 32 次/s,CRT 模式 76 次/s;RSA-2048 签名、非 CRT 模式 4 次/s,CRT 模式 13 次/s.需要指出的是 SM2 算法的安全强度是高于 RSA-2048 的,大致与 RSA-3072 相当,如果在 IC 卡中运行 RSA-3072,将是相当耗时的运算,而 SM2 算法所需的运算负载显然要小很多.

通过在硬件协处理器中采用并行及脉动流水线结构等技术,SM2 算法在 ASIC 可以实现很高的运算性能,适用于大型签名验证服务器.以我国现有 IC 设计和制造技术,已经有 SM2 数字签名性能超过 1 万次/s 的芯片研制成功并得到应用.

SM2 算法的实现效率可以满足从服务器端到客户端的不同种类的信息安全设备的需求.

4 小　　结

SM2 算法是我国在吸收国际先进成果的基础上研制的具有自主知识产权的 ECC,它在安全性和实现效率方面相当于或略优于国际上同类的 ECC,能取代 RSA 以满足各种应用对公钥密码算法安全性和实现效率的更高要求,具有广阔的推广和应用前景.

参　考　文　献

[1] Koblitz N. Elliptic curve cryptosystems [J]. Mathematics of Computation, 1987, 48(177): 203-209

[2] Miller V S. Uses of elliptic curves in cryptography [G] //LNCS 218: Proc of Advances in Cryptology—CRYPTO'85. Berlin: Springer, 1986: 417-426

[3] Trusted Computing Group. TCG TPM specification 2.0 [EB/OL]. 2013 [2016-10-06]. http://www.trustedcomputinggroup.org/resources/tpm

[4] ISO/IEC 11889: 2015 Information technology-trusted platform module library [S/OL]. 2015 [2016-10-06]. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=66510

[5] 国家密码管理局. SM2 椭圆曲线公钥密码算法[EB/OL]. 2010 [2016-10-06]. http://www.oscca.gov.cn/News/201012/News_1197.htm

[6] Pohlig S, Hellman M. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance [J]. IEEE Trans on Information Theory, 1978, 24(1): 106-110

[7] Pollard J M. Monte Carlo methods for index computation mod p [J]. Mathematics of Computation, 1978, 32(143): 918-924

[8] Menezes B A, Okamoto T, Vanstone S A. Reducing elliptic curves logarithms to logarithms in a finite field [J]. IEEE Trans on Information Theory, 1993, 39(5): 1639-1646

[9] Gaudry P, Hess F, Smart N P. Constructive and destructive facets of Weil descent on elliptic curves [J]. Journal of Cryptology, 2002, 15(1): 19-46

[10] Smart N P. The discrete logarithm problem on elliptic curves of trace one [J]. Journal of Cryptology, 1999, 12(3): 193-196

[11] Satoh T, Araki K. Fermat quotients and the polynomial time discrete logalgorithm for anomalous elliptic curves [J]. Commentarii Mathematici Universitatis Sancti Pauli, 1998, 1(1): 81-92

[12] Goldwasser S, Micali S, Rivest R L. A "paradoxical" solution to the signature problem [C] //Proc of Symp on Foundations of Computer Science. Los Alamitos, CA: IEEE Computer Society, 1984: 441-448

[13] Goldwasser S, Micali S, Rivest R L. A digital signature scheme secure against adaptive chosen-message attacks [J]. Siam Journal on Computing, 1988, 17(2): 281-308

[14] Menezes A, Smart N. Security of signature schemes in a multi-user setting [J]. Designs, Codes and Cryptography, 2004, 33(3): 261-274

[15] Blake-Wilson S, Menezes A. Unknown key-share attacks on the station-to-station (STS) protocol [C] //Proc of Int Workshop on Practice and Theory in Public Key Cryptography. Berlin: Springer, 1999: 154-170

[16] Geiselmann W, Steinwandt R. A key substitution attack on SFLASH [J]. Journal of Discrete Mathematical Sciences & Cryptography, 2005 (2): 137-141

[17] Tan C H. Key substitution attacks on some provably secure signature schemes [J]. IEICE Trans on Fundamentals of Electronics Communications & Computer, 2004, 87(1): 226-227

[18] Nechaev V I. Complexity of a determinate algorithm for the discrete logarithm [J]. Mathematical Notes, 1994, 55 (2): 165-172

[19] Bellare M, Rogaway P. Random oracles are practical: A paradigm for designing efficient protocols [C] //Proc of ACM Conf on Computer & Communication Security. New York: ACM, 1993: 62-73

[20] Zhang Zhenfeng, Yang Kang, Zhang Jiang, et al. Security of the SM2 signature scheme against generalized key substitution attacks [G] //LNCS 9497: Security Standardisation Research. Berlin: Springer, 2015: 140-153

[21] Xu Jing, Feng Dengguo. Comments on the SM2 key exchange protocol [M] //Cryptology and Network Security. Berlin: Springer, 2011: 160-171

[22] Yang A, Nam J, Kim M, et al. Provably-secure (chinese government) SM2 and simplified SM2 key exchange protocols [J/OL]. The Scientific World Journal, 2014: 825984 [2016-10-20]. https://www.hindawi.com/journals/tswj/2014/825984/

[23] Zhao Shijun, Xi Li, Zhang Qianying, et al. Security analysis of SM2 key exchange protocol in TPM2.0 [J]. Security & Communication Networks, 2015, 8(3): 383-395

[24] Goldwasser S, Micali S. Probabilistic encryption [J]. Journal of Computer & System Sciences, 1984, 28(2): 270-299

[25] Dolev D, Dwork C, Naor M, et al. Non-malleable cryptography [C] //Proc of ACM Symp on Theory of Computing. New York: ACM, 1991: 542-552

[26] Bellare B M, Rogaway P. Optimal asymmetric encryption [C] //Proc of Int Cryptology Conf on Advances in Cryptology--Eurocrypt '94. Berlin: Springer, 1994: 92-111

[27] Zheng Y, Seberry J. Practical approaches to attaining security against adaptively chosen ciphertext attacks (extended abstract) [C] //Proc of Int Cryptology Conf on Advances in Cryptology. Berlin: Springer, 1992: 292-304

[28] Abdalla B M, Bellare M, Rogaway P. DHAES: An encryption scheme based on the Diffie-Hellma problem, 1999/007 [R/OL]. Cryptology ePrint Archive. [2016-10-06]. http://eprinl.iacr.org

[29] Cramer R, Shoup V. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack [C] //Proc of Int Cryptology Conf on Advances in Cryptology—Crypto'98. Berlin: Springer, 1998: 13-25

汪朝晖

计算机科学硕士，应用数学博士，主要研究方向为密码算法理论、软硬件实现及应用技术.

wangzh@istecc.com

张振峰

博士，研究员，博士生导师，主要研究方向为密码学与安全协议、网络信任理论与技术.

zfzhang@tca.iscas.ac.cn