·安全技术· 文章编号：1000-3428(2016)01-0145-05 文献标识码：A 中图分类号：TP309

基于离散对数的数字签名标准对比研究

冯泽宇[a]，巩博儒[b]，赵运磊[a]

（复旦大学 a. 软件学院；b. 计算机科学技术学院，上海 201203）

摘 要： 在国家密码管理局公开征集下一代商密公钥密码算法标准的背景下，从效率和安全性方面，对基于离散对数问题（DLP）或椭圆曲线 DLP 的 ISO/IEC 14888-3 中 7 种数字签名标准及国密 SM2 标准进行对比分析。结果表明，数字签名算法（DSA）是 Schnorr 和 ElGamal 签名算法的结合，其应用广泛，现已发展为 EC-DSA，且安全性高于 SM2。Pointcheval/Vaudenay 算法是可证明安全的，KCDSA 和 EC-KCDSA 的效率及安全性均较高，EC-RDSA 和 EC-GDSA 的签名生成过程较快。给出针对 EC-RDSA 的攻击方法，证明其在自适应性选择消息攻击下不是强存在性不可伪造的。上述研究结果对我国下一代商密公钥密码算法标准的设计和制定具有参考作用。

关键词： 离散对数问题；椭圆曲线离散对数问题；数字签名标准；随机谕示模型；SM2 算法

中文引用格式： 冯泽宇，巩博儒，赵运磊. 基于离散对数的数字签名标准对比研究[J]. 计算机工程，2016，42(1)：145-149.

英文引用格式： Feng Zeyu，Gong Boru，Zhao Yunlei. Comparative Study of Digital Signature Standards Based on Discrete Logarithm[J]. Computer Engineering，2016，42(1)：145-149.

Comparative Study of Digital Signature Standards Based on Discrete Logarithm

FENG Zeyu[a]，GONG Boru[b]，ZHAO Yunlei[a]

（a. Software School；b. School of Computer Science，Fudan University，Shanghai 201203，China）

【Abstract】 As Chinese state encryption administration is seeking the next generation of Digital Signature Standard（DSS），this paper analyzes and compares seven DSS listed in ISO/IEC 14888-3 and SM2 which are based on Discrete Logarithm Problem（DLP） or Elliptic Curve Discrete Logarithm Problem（ECDLP）. Results show that the widely used Digital Signature Algorithm（DSA） is a combination of Schnorr and ElGamal signature algorithm and it becomes Elliptic Curve Digital Signature Algorithm（EC-DSA）. SM2 may be more vulnerable than EC-DSA. Moreover，the Pointcheval/Vaudenay algorithm is provably secure. The Korean Certificate-based Digital Signature Algorithm（KCDSA） and its elliptic curve version Elliptic Curve Korean Certificate-based Digital Signature Algorithm（EC-KCDSA） performs better both in security and efficiency issues. The signature algorithms of Elliptic Curve Russia Digital Signature Algorithm（EC-RDSA） and Elliptic Curve Germany Digital Signature Algorithm（EC-GDSA） are faster. It is worth noting that an attack against EC-RDSA is proposed，implying that EC-RDSA is not strongly existential unforgeability under the adaptive chosen-message attack. The comparative results is helpful for the research，as well as for the finalization of the next generation of DSS.

【Key words】 Discrete Logarithm Problem（DLP）；Elliptic Curve Discrete Logarithm Problem（ECDLP）；Digital Signature Standard（DSS）；Random Oracle Model（ROM）；SM2 algorithm

DOI： 10.3969/j.issn.1000-3428.2016.01.026

1 概述

数字签名概念由 Diffie 和 Hellman 在 1976 年提出，随后出现了很多数字签名算法（Digital Signature Algorithm，DSA），如 RSA，ElGamal[1]，Schnorr[2]，Fiat-Shamir[3]。一些组织根据这些算法制定了数字签名标准（Digital Signature Standard，DSS），其中最著名的是美国国家标准与技术研究所（NIST）提出的数字签名标准，其所用的数字签名算法主要基于离散对数问题（Discrete Logarithm Problem，DLP）的难解性。DSA 实际上是 ElGamal 和 Schnorr 签名算法的一种变体。

DSA 推出后得到了密码学界的广泛关注，一些专家学者指出 DSA 实际上存在很多缺点，其中很重要的一条是 DSA 的安全性未得到证明。文献[4-5]对 DSA 做了略微改进，提出 DSA 的 2 种变体，并说

基金项目： 国家自然科学基金资助项目（61472084，61272012）；中科院信工所信息安全国家重点实验室开放课题基金资助项目。

作者简介： 冯泽宇（1991－），男，硕士研究生，主研方向为密码学、信息安全；巩博儒，博士研究生；赵运磊，教授、博士生导师。

收稿日期： 2014-11-10 **修回日期：** 2015-01-11 **E-mail：** zeyufeng13@fudan.edu.cn

明这 2 种变体是可证明安全的,其中一个变体用到随机谕示模型(Random Oracle Model,ROM)。随后文献[6]推出了 DSA 的一个改良版本,即基于认证的数字签名算法(Korean Certificate-based Digital Signature Algorithm,KCDSA),该算法提高了签名效率,并引用文献[5]中的一个变体,所以,该算法也是可证明安全的。

椭圆曲线密码学(Elliptic Curve Cryptography,ECC)由 Neal Koblitz 和 Victor S. Miller 于 1985 年提出,并于 2004 年、2005 年投入应用,其主要依赖于椭圆曲线离散对数问题(Elliptic Curve Discrete Logarithm Problem,ECDLP)的难解性。ECDLP 可看作传统 DLP 的扩展,即将传统 DLP 中的$\mathbb{Z}_p^*$子群用有限域上的椭圆曲线点群代替。由于相同参数条件下,ECDLP 的困难性远高于 DLP,因此基于 ECDLP 的数字签名算法的安全性也更强。在提供相同等级安全保证的前提下,ECC 系统所需参数较少,密钥长度也较短,因此,可节省存储空间、占用带宽较少,这在资源受限的计算环境中有很大的优势。

椭圆曲线数字签名算法(Elliptic Curve Digital Signature Algorithm,EC-DSA)[7]是 DSA 与 ECC 的结合,由于其安全性更好,因此在 1992 年被 Scott Vanstone 提出后,于 1998 年被选作为 ISO 标准,其后又被各大权威组织选定为安全标准。该标准和 DSA 标准、Pointcheval/Vaudenay 标准、KCDSA 标准收录于国际标准草案 ISO/IEC 14888-3 文件[8]中,除此之外该文件还收录了 EC-KCDSA(Elliptic Curve Korean Certificate-based Digital Signature Algorithm),EC-RDSA(Elliptic Curve Russia Digital Signature Algorithm)[9]和 EC-GDSA(Elliptic Curve Germany Digital Signature Algorithm)[10]等基于 ECC 的数字签名标准。近期国家密码管理局计划对数字签名标准进行公开征集。签名算法的效率和安全性无疑是重要的考量指标,本文从 DSA 出发,对各数字签名标准进行研究,并与中国商密 SM2 数字签名标准进行对比[11]。

2 预备知识

2.1 离散对数问题

给定一个素数 p,选取$\mathbb{Z}_p^*$的 q 阶子群的生成元 g,对$\mathbb{Z}_p^*$上整数 y,寻找唯一的整数 x,使得 $y=g^x \bmod p$。一般地,若 p 足够大,则认为该问题在概率多项式时间内是难解的。

2.2 椭圆曲线离散对数问题

给定素数 p 和椭圆曲线 E,在给定 E 上的点 P,Q 的情况下求出小于 p 的正整数 k,使得 $Q=kP$。可以证明,已知 k 和 P 计算 Q 比较容易,而由 Q 和 P 计算 k 则比较困难。

2.3 存在性不可伪造

一个数字签名算法通常由 3 个算法组成:KeyGen,Sign,Verify。自适应性选择消息攻击下的存在性不可伪造的定义需要用到以下博弈:

(1)启动。挑战者运行 KeyGen,并将结果中的公钥 PK 交给敌手,而私钥 SK 自己保留。

(2)签名询问。敌手以自适应的方式发出签名的询问 $m_1,m_2,\cdots,m_q$。对每个询问 m_i,挑战者运行 Sign 算法,生成 m_i 的签名 σ_i,并将 σ_i 发送给敌手。这些询问可以是自适应性的,即每个询问 m_i 都可以根据 $m_1,m_2,\cdots,m_{i-1}$的询问结果来决定。

(3)输出。最终敌手输出一对(m,σ)。若根据 Verify 算法 σ 是 m 的有效签名,并且 m 不包含在询问阶段产生的 m_i 中,则敌手赢。

本文将敌手 A 攻击签名算法的优势定义为 A 赢得以上博弈的概率,一个签名算法在自适应性选择消息攻击下是(t,q,ε)-存在性不可伪造的[12],必须满足:不存在敌手 A,询问至多 q 次,在时间为 t 的情况下,在以上的博弈中具有的优势至少为 ε。

2.4 强存在性不可伪造

本文将用到自适应性选择消息攻击下的强存在性不可伪造的安全特性,定义这种安全特性需要用到以下博弈:

(1)启动和签名询问。与存在性不可伪造博弈中相同。

(2)输出。敌手输出一对(m,σ)。若根据 Verify 算法 σ 是 m 的有效签名,并且若(m,σ)不包含在询问阶段产生的对(m_i,σ_i)中,则敌手赢。

同样地,将敌手 A 攻击签名算法的优势定义为 A 赢得以上博弈的概率,一个签名算法在自适应性选择消息攻击下是(t,q,ε)-强存在性不可伪造的,必须满足:不存在敌手 A 询问至多 q 次,在时间为 t 的情况下,在以上博弈中具有的优势至少为 ε。

3 数字签名算法

3.1 算法实现过程

数字签名算法包括 3 个过程,即密钥生成过程、签名过程和验证过程。

3.1.1 密钥生成过程

密钥生成过程具体如下:

(1)选取一个素数 q,使得 $2^{\beta-1}<q<2^{\beta}$;

(2)选取一个素数 p,使得 $2^{\alpha-1}<p<2^{\alpha}$,其中,$q$ 可以整除 $p-1$;

(3)选取$\mathbb{Z}_p^*$的 q 阶子群的生成元 g,具体过程如下:

1)选取元素 $g'\in\mathbb{Z}_p^*$并计算 $g=g'^{(p-1)/q} \bmod p$;

2)若 $g=1$ 则返回步骤 1);

(4)秘密生成一个随机整数 x,使得 $0<x<q$;

(5)计算 $y=g^x \bmod p$;

(6)生成公钥(p,q,g,y),私钥 x。

3.1.2 签名和验证过程

签名过程具体如下:

(1)随机选取 k,使得 $0<k<q$;

(2)计算 $r=(g^k \bmod p) \bmod q$;

(3)计算 $s=k^{-1}(h(m)+xr) \bmod q$;

(4)最终签名为(r,s)。

验证过程具体如下:

(1)获得可信公钥(p,q,g,y);

(2)验证 $0<r<q$,$0<s<q$,若不满足则拒绝该签名;

(3)计算 $v=(g^{s^{-1}\cdot h(m)} \cdot y^{r\cdot s^{-1}} \bmod p) \bmod q$;

(4)当且仅当 $v=r$ 时,接受此签名。

参数 α,β 取值位数与安全等级的关系见表 1。

表 1　不同安全等级对应的参数取值位数

安全等级	α	β
2^{80}	1 024	160
2^{112}	2 048	224
2^{128}	3 072	256
2^{192}	7 680	384
2^{256}	15 360	512

另外,DSA 中的(α,h)有 4 种选择,分别为(1 024, SHA-1),(2 048,SHA-224),(2 048,SHA-256),(3 072, SHA-256)。

3.2　Schnorr,ElGamal 和 DSA 算法

DSA 实际上是 Schnorr 签名算法和 ElGamal 签名算法混合的产物,这三者之间的比较见表 2,算法生成的签名及下文中其他签名标准生成的签名均为(r,s)。其中,$x\in_r \mathbb{Z}_q^*$ 表示 x 在 $\mathbb{Z}_q^*$ 中随机取一个元素。

表 2　ElGamal,Schnorr 和 DSA 签名算法对比

算法	签名过程	验证过程
ElGamal	私钥 x,在$(0,p-1)$中随机选取 x 随机选取整数 $k\in(0,p-1)$,且: $gcd(k,p-1)=1$ $r=g^k \bmod p$ $s=k^{-1}(m-xr)$	公钥:$y=g^x \bmod p$ $y^r r^s \bmod p \overset{?}{=} g^m \bmod p$
Schnorr	私钥:$x\in_r \mathbb{Z}_q^*$ $k\in_r \mathbb{Z}_q^*$ $r=h(g^k \bmod p \parallel m)$ $s=(k+xr) \bmod q$	公钥:$y=g^x \bmod p$ $h(g^s y^{-r} \bmod p \parallel m) \overset{?}{=} r$
DSA	私钥:$x\in_r \mathbb{Z}_q^*$ $k\in_r \mathbb{Z}_q^*$ $r=(g^k \bmod p) \bmod q$ $s=k^{-1}(h(m)+xr) \bmod q$	公钥:$y=g^x \bmod p$ $u_1=s^{-1}r \bmod q$ $u_2=s^{-1}h(m) \bmod q$ $(y^{u_1}g^{u_2} \bmod p) \bmod q \overset{?}{=} r$

由表 2 可以看出,ElGamal 算法实际上是在 $\mathbb{Z}_p^*$ 中进行的,而 Schnorr 算法是取 $p-1$ 的素因子 q,是在阶为 q 的子群中进行的运算。这样可以有效缩短签名的长度,使运算变得简单。而 DSA 正是继承了 Schnorr 签名算法的这一优点。签名生成过程中有关 s 的计算,DSA 和 ElGamal 相比只是对 m 进行了哈希,并将减法运算变为加法运算,从而简化验证过程,主要的签名方法还是从 ElGamal 继承而来。

4　Pointcheval/Vaudenay 和 KCDSA 算法

4.1　Pointcheval/Vaudenay 算法

DSA 自推出后,安全性一直未得到证明,Pointcheval 和 Vaudenay 在文献[1]中提出了 DSA 的 2 种变体,并已证明其安全性。

变体 1　先将 DSA 中的 SHA 替换为一个 RO h_1,再将 $r=(g^k \bmod p) \bmod q$ 中的模 q 操作替换为另一个 RO h_2,即:

$$r=h_2(g^k \bmod p)$$

$$s=k^{-1}(h_1(m)+xr) \bmod q$$

相应的验证公式变为:

$$h_2(g^{s^{-1}\cdot h_1(m)} \cdot y^{r\cdot s^{-1}} \bmod p)=r$$

变体 2　将原来 DSA 公式中对 m 的哈希变为对 m 和 r 的哈希,即:

$$r=(g^k \bmod p) \bmod q$$

$$s=k^{-1}(h(m \parallel r)+xr) \bmod q$$

相应的验证公式为:

$$(g^{s^{-1}\cdot h(m \parallel r)} \cdot y^{r\cdot s^{-1}} \bmod p) \bmod q=r$$

4.2　KCDSA 算法

KCDSA 是基于认证的数字签名算法,其在算法效率和安全性方面较 DSA 都有所提升,两者比较见表 3。

表 3　DSA 和 KCDSA 算法比较

算法	签名过程	验证过程
DSA	私钥:$x\in_r \mathbb{Z}_q^*$ $k\in_r \mathbb{Z}_q^*$ $r=(g^k \bmod p) \bmod q$ $s=k^{-1}(h(m)+xr) \bmod q$	公钥:$y=g^x \bmod p$ $u_1=s^{-1}r \bmod q$ $u_2=s^{-1}h(m) \bmod q$ $(y^{u_1}g^{u_2} \bmod p) \bmod q \overset{?}{=} r$
KCDSA	私钥:$x\in_r \mathbb{Z}_q^*$ $k\in_r \mathbb{Z}_q^*$ $r=h(g^k \bmod p)$ $z=h(Cert\_Data)$ $e=r\oplus h(z \parallel m) \bmod q$ $s=x(k-e) \bmod q$	公钥:$y=g^{x^{-1}} \bmod p$ $e=r\oplus h(z \parallel m) \bmod q$ $h(y^s g^e \bmod p) \overset{?}{=} r$

在表 3 中,*Cert_Data* 表示签名者的认证数据,包括签名者的 ID、公钥 y、参数$\{p,q,g\}$。

在效率性方面,KCDSA 只需要在密钥生成阶段求一次逆,而在 DSA 中,在每次签名和验证过程中都要进行模 q 的乘法求逆。虽然在日常电脑上,整个签名、验证过程求逆只占很少的部分,但在计算资源受限的环境中(如智能卡)求逆的代价就会很高。

在安全性方面,将 DSA 中的 $r=(g^k \bmod p) \bmod q$ 替换为 $r=h(g^k \bmod p)$(效仿 Pointcheval/Vaudenay 算法中的变体 1),从而保证 KCDSA 在 RO 下也是可证明安全的。

另外,$z=h(Cert_Data)$的使用有效地抵御了参数

生成阶段可能的伪造,又因为每个签名者的 z 都是唯一的,使得搜索到一个哈希碰撞的难度大大增加。

5 基于 ECC 的数字签名标准

在 DSA 的基础上,推出椭圆曲线版本 EC-DSA,其后各个国家都推出了各自的基于 ECC 的数字签名标准,如 EC-DSA、俄罗斯的 EC-RDSA、中国的 SM2、韩国的 EC-KCDSA、德国的 EC-GDSA,这些签名标准的比较见表 4。

表 4 基于 ECC 的数字签名标准

算法	签名过程	验证过程
EC-DSA	私钥:$x\in_r\mathbb{Z}_q^*$ $k\in_r\mathbb{Z}_q^*$ $r=\pi(kG)\bmod q$ $s=k^{-1}(h(m)+xr)\bmod q$	公钥:$Y=xG$ $u_1=s^{-1}r\bmod q$ $u_2=s^{-1}h(m)\bmod q$ $\pi(u_1Y+u_2G)\bmod q\stackrel{?}{=}r$
EC-RDSA	私钥:$x\in_r\mathbb{Z}_q^*$ $k\in_r\mathbb{Z}_q^*$ $r=\pi(kG)\bmod q$ $s=rx+k\cdot h(m)\bmod q$	公钥:$Y=xG$ $u_1=h(m)^{-1}r\bmod q$ $u_2=s\cdot h(m)^{-1}\bmod q$ $\pi(u_2G-u_1Y)\bmod q\stackrel{?}{=}r$
SM2	私钥:$x\in_r\mathbb{Z}_q^*$ $k\in_r\mathbb{Z}_q^*$ $z=h(Cert\_Data)$ $r=h(z\parallel m)+\pi(kG)\bmod q$ $s=(1+x)^{-1}(k-rx)\bmod q$	公钥:$Y=xG$ $t=(r+s)\bmod q$ $e=sG+tY$ $h(z\parallel m)+\pi(e)\bmod q\stackrel{?}{=}r$
EC-KCDSA	私钥:$x\in_r\mathbb{Z}_q^*$ $k\in_r\mathbb{Z}_q^*$ $r=h(kG)$ $z=h(Cert\_Data)$ $e=r\oplus h(z\parallel m)\bmod q$ $s=x(k-e)\bmod q$	公钥:$Y=x^{-1}G$ $e=r\oplus h(z\parallel m)\bmod q$ $h(sY+eG)\stackrel{?}{=}r$
EC-GDSA	私钥:$x\in_r\mathbb{Z}_q^*$ $k\in_r\mathbb{Z}_q^*$ $r=\pi(kG)\bmod q$ $s=x(kr-h(m))\bmod q$	公钥:$Y=x^{-1}G$ $u_1=r^{-1}s\bmod q$ $u_2=r^{-1}h(m)\bmod q$ $\pi(u_1Y+u_2G)\bmod q\stackrel{?}{=}r$

在表 4 中,$\pi(\ )$ 函数的作用是将一个椭圆曲线点转换为一个整数。对于定义在 $GF(p)$ 中的椭圆曲线,$\pi((x_1,y_1))=x_1$,而在 $GF(2^m)$ 中,x_1 可以表示成二进制串($s_{m-1}s_{m-2}\cdots s_0$),此时 $\pi((x_1,y_1))=\sum_{i=0}^{m-1}s_i2^i$。

EC-DSA,EC-KCDSA 即 DSA 和 KCDSA 从基于 DLP 转移到基于 ECDLP,其本质并没有发生变化,因此下面主要分析其他 3 种签名算法。

5.1 EC-RDSA 和 EC-GDSA 算法

EC-RDSA 代表俄罗斯的基于椭圆曲线的数字签名算法,是俄罗斯政府于 2001 年发布的数字签名标准 GOST 34.10-2001,其前身是 GOST 34.10-94,即俄罗斯的 DSA。EC-GDSA 代表德国的基于椭圆曲线的数字签名算法,于 1990 年被提出,比 DSA 和 EC-DSA 的提出要早一些。两者的共同特性是在签名生成过程中避免了乘法求逆,提高了签名生成阶段的效率。

5.2 SM2 算法

SM2 算法由我国国家密码管理局于 2010 年发布,是 SM2 椭圆曲线公钥密码算法的一部分。发布该算法的主要目的是改善 RSA 算法不安全的现状。随着密码技术和计算技术的发展,目前常用的 1 024 bit RSA 算法面临严重的安全威胁,因此,国家密码管理部门经过研究,决定采用 SM2 椭圆曲线算法替换 RSA 算法。

与 KCDSA 一样,SM2 对待签名消息做了预处理,添加了 z 值(包含签名者自身的信息)以提高安全性和不可抵赖性。SM2 中规定使用的哈希算法为 SM3,SM3 输出长度为 256 bit,其安全性基本等同于 SHA-256,但可能高于 SHA-1,SHA-224。

在安全性方面,文献[13]指出相对于 EC-DSA,在部分已知随机数攻击下,SM2 更容易被攻破,且计算 SM2 中 $s=(1+x)^{-1}(k-rx)\bmod q$ 的 $(1+x)^{-1}$ 时,x 的信息在旁道攻击下更容易泄露,并且 SM2 在错误注入攻击下亦是不安全的,因此,可以认为 SM2 的安全性低于 EC-DSA。

6 密钥、签名长度及计算过程对比

从表 5 可以看出,各个签名算法的密钥长度、生成的签名长度以及整个算法过程包含的模指数运算、求逆运算的次数,其中,exp 表示模指数(模 p);EXP 表示标量乘法;inv 表示求逆(模 q)。

表 5 密钥、签名长度及计算过程比较

算法	密钥长度		签名长度	计算过程		
	签名	验证		密钥生成	签名	验证
DSA	β	α	2β	1 exp	1 exp,1 inv	2 exp,1 inv
Pointcheval/Vaudenay	β	α	2β	1 exp	1 exp,1 inv	2 exp,1 inv
KCDSA	β	α	2β	1 exp,1 inv	1 exp	2 exp
EC-DSA	β	2β	2β	1 EXP	1 EXP,1 inv	2 EXP,1 inv
EC-KCDSA	β	2β	2β	1 EXP,1 inv	1EXP	2 EXP
EC-GDSA	β	2β	2β	1 EXP,1 inv	1EXP	2 EXP,1 inv
EC-RDSA	β	2β	2β	1EXP	1EXP	2 EXP,1 inv
SM2	256	512	512	1EXP	2 EXP,1 inv	2 EXP

7　针对 EC-RDSA 的攻击和改进

EC-RDSA 的签名公式为 $s = rx + k \cdot h(m) \bmod q$,假如对同一个消息签名 2 次,得到 2 个签名为:(r_1, s_1),(r_2, s_2),其中,$s_1 = r_1 x + k_1 \cdot h(m) \bmod q$;$s_2 = r_2 x + k_2 \cdot h(m) \bmod q$,将等式两端分别相加可得 $s_1 + s_2 = (r_1 + r_2)x + (k_1 + k_2) \cdot h(m) \bmod q$。这样就可得到关于 m 的签名 $(r_1 + r_2, s_1 + s_2)$,从而对消息 m 的签名进行伪造。根据定义可知,EC-RDSA 在选择消息攻击下不是强存在性不可伪造的。所以,在签名过程中直接使用 m 的哈希值是不安全的,文献[14]在签名时对 m 的哈希值进行加密,即 $s = rx + k \cdot E_k(h(m)) \bmod q$,并证明了该方法在选择消息攻击下是强存在性不可伪造的。

8　结束语

本文对 ISO/IEC 14888-3 文件中的 7 种数字签名标准和国密局发布的 SM2 标准从效率和安全性方面进行了对比研究,得到以下结论:(1)DSA 算法是 ElGamal 签名算法和 Schnorr 签名算法的结合,自推出以来便因其使用便利而得到广泛应用,现已发展为 EC-DSA。(2)Pointcheval/Vaudenay 算法在 DSA 的基础上进行改进,并对修改版本的安全性进行证明。(3)KCDSA 引用了 Pointcheval/Vaudenay 算法的一个版本,并对签名过程进行改进,使其相对于 DSA 而言在效率和安全性上都有了提升,同时推出 ECC 版本,即 EC-KCDSA。(4)EC-GDSA 和 EC-RDSA 避免了签名生成过程中的求逆运算,提高了签名过程的效率,并在本文中给出一种针对 EC-RDSA 的攻击方法。(5)SM2 是国密局近年来发布的基于 ECDLP 的数字签名标准,其推出的主要目的是替换现已不再安全的 RSA 数字签名标准。目前的分析结果表明,SM2 的安全性低于 EC-DSA。下一步工作将以本文研究为基础,设计更安全高效的数字签名算法。

参考文献

[1] El Gamal T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms[C]//Proceedings of CRYPTO'84. Berlin, Germany: Springer, 1985: 10-18.

[2] Schnorr C P. Efficient Identification and Signatures for Smart Cards[C]//Proceedings of CRYPTO'89. Berlin, Germany: Springer, 1990: 239-252.

[3] Schnorr C P. Efficient Signature Generation by Smart Cards[J]. Journal of Cryptology, 1991, 4(3): 161-174.

[4] Pointcheval D, Stern J. Security Arguments for Digital Signatures and Blind Signatures[J]. Journal of Cryptology, 2000, 13(3): 361-396.

[5] Vaudenay S, Pointcheval D. On Provable Security for Digital Signature Algorithms, LIENS-96-17[R]. LIENS, 1996.

[6] Lim C, Lee P. The Korean Certificate-based Digital Signature[J]. Computers & Electrical Engineering, 1999, 25(4): 249-265.

[7] Johnson D, Menezes A, Vanstone S. The Elliptic Curve Digital Signature Algorithm (ECDSA)[J]. International Journal of Information Security, 2001, 1(1): 36-63.

[8] American National Standards Institute. ISO/IEC 14888-3-2013 Information Technology-Security Techniques-Digital Signatures with Appendix, Part 3: Discrete Logarithm Based Mechanisms[S]. 2013.

[9] Michels M, Naccache D, Petersen H. GOST 34.10-A Brief Overview of Russia's DSA[J]. Computers & Security, 1996, 15(8): 725-732.

[10] Hess E, Schafheutle M, Serf P, et al. The Digital Signature Scheme ECGDSA[EB/OL]. (2006-12-24). https://www.teletrust.de/fileadmin/files/oid/ecgdsa_final.pdf.

[11] 国家密码管理局. SM2 椭圆曲线公钥密码算法[EB/OL]. (2010-12-17). http://www.oscca.gov.cn/News/201012/News_1197.htm.

[12] Boneh D, Shen E, Waters B. Strongly Unforgeable Signatures Based on Computational Diffie-Hellman[C]//Proceedings of PKC'06. Berlin, Germany: Springer, 2006: 229-240.

[13] Liu Mingjie, Chen Jiazhe, Li Hexin. Partially Known Nonces and Fault Injection Attacks on SM2 Signature Algorithm[C]//Proceedings of Information Security and Cryptography Conference. Berlin, Germany: Springer, 2014: 343-358.

[14] Varnovskii N P. Provable Security of Digital Signatures in the Tamper-proof Device Model[J]. Discrete Mathematics and Applications, 2008, 18(4): 427-437.

编辑　陆燕菲

(上接第 144 页)

[10] Dowland P, Furnell S, Papadaki M. Keystroke Analysis as a Method of Advanced User Authentication and Response[C]//Proceedings of the 17th International Conference on Information and Communication Technology. Berlin, Germany: Springer, 2002: 215-226.

[11] Hosseinzadeh D, Krishnan S. Gaussian Mixture Modeling of Keystroke Patterns for Biometric Applications[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2008, 38(6): 816-826.

[12] Bhatt S, Santhanam T. Keystroke Dynamics for Biometric Authentication——A Survey[C]//Proceedings of International Conference on Pattern Recognition, Informatics and Mobile Engineering. Berlin, Germany: Springer, 2013: 21-22.

[13] Peacock A, Ke Xian, Wilkerson M. Typing Patterns: A Key to User Identification[J]. IEEE Security and Privacy, 2004, 2(5): 40-47.

[14] Samura T, Nishimura H. Influence of Keyboard Difference on Personal Identification by Keystroke Dynamics in Japanese Free Text Typing[C]//Proceedings of the 5th International Conference on Emerging Trends in Engineering and Technology. Washington D. C., USA: IEEE Press, 2012.

[15] Samura T, Nishimura H. Keystroke Dynamics for Individual Identification in Japanese Free Text Typing[J]. SICE Journal of Control, Measurement and System Integration, 2011, 4(2): 172-176.

编辑　顾逸斐