ECDSA 可公开验证广义签密

韩益亮 杨晓元

(武警部队网络与信息安全重点实验室 西安 710086)

(武警工程学院电子技术系 西安 710086)

摘 要 签密是一种能够同时提供加密和认证功能的密码体制.该文首次将签密的应用范围推广到仅需要机密性或完整性的场合,提出了广义签密的定义.广义签密不仅具有一般签密的属性,针对特定的输入还可提供单独的加密或签名功能.基于椭圆曲线数字签名标准 ECDSA,文中提出一个广义签密方案 SC-ECDSA,无特定验证方时等价于 ECDSA 签名,无特定发送方时等价于 AtE(OTP$_{\$}$,MAC)结构加密,第三方可在 ECDSA 的模式下公开验证.在 Random Oracle 模型上证明了该方案的安全性:CUF-CPA 的机密性、与 ECDSA 同等的不可伪造性和不可否认性.效率分析表明,在适当的安全参数下 SC-ECDSA 的计算量比目前最快的 SCS 签密降低了 78%.

关键词 签密;广义签密;数字签名;认证加密;椭圆曲线数字签名算法

中图法分类号 TP309

New ECDSA-Verifiable Generalized Signcryption

HAN Yi-Liang YANG Xiao-Yuan

(*Key Laboratory on Network and Information Security of Armed Police Force, Xi'an* 710086)

(*Department of Electronic Technology, Engineering College of Armed Police Force, Xi'an* 710086)

Abstract Signcryption is a new cryptographic primitive that simultaneously fulfills both the functions of signature and encryption. The definition of generalized signcryption is proposed in the paper firstly. Generalized signcryption has a special feature that provides confidentiality or authenticity separately under specific inputs. So it is more useful than common ones. Based on ECDSA, a signcryption scheme called SC-ECDSA is designed. It will be equivalent to an AtE(OTP$_{\$}$, MAC) encryption scheme or ECDSA when one of party is absent. A third party can verify the signcryption text publicly in the method of ECDSA. Security properties are proven based on Random Oracle mode: Confidentiality(CUF-CPA), unforgeability(UF-CMA) and non-repudiation. For typical security parameters for high level security applications, compared with the others, SC-ECDSA presents a 78% reduction in computational cost.

Keywords signcryption; generalized signcryption; digital signature; authenticated encryption; ECDSA

1 引 言

保密且完整地传输或存储消息是信息安全的主题之一.为满足这一要求,传统的做法是将加密和认证码组合[1],SSL(Secure Socket Layer)、IPSec(Internet Protocol Security)等安全协议都是如此.在公钥密码系统中也可以将加密和数字签名组合成

收稿日期:2005-01-24;修改稿收到日期:2006-02-12.本课题得到国家自然科学基金(60473029)资助.**韩益亮**,男,1977 年生,硕士,讲师,主要研究方向为密码学、分布式系统安全. E-mail: yilianghan@hotmail.com. **杨晓元**,男,1959 年生,教授,主要研究领域为密码学、隐藏信息检测.

为“签名再加密(Sign-then-Encrypt)”结构,如 PGP(Pretty Good Privacy).但这并不是有效的方法:第一,效率不高,计算量和数据量是两者之和;第二,随意的组合方案无法保证安全性,甚至一些安全协议如 WEP(Wired Equivalent Privacy)也被证明不安全.Zheng 提出了一种称为签密(signcryption)[2]的新体制,能在一个逻辑步骤内同时完成加密和签名双重功能,而计算量和数据量小于两者之和.签密是密码学中一个活跃的领域,近几年的研究主要集中在两个方向:(1)构造具体的签密方案;(2)研究抽象的设计结构.

前一方向侧重于和具体签名方案相结合,以基于 ElGamal 型签名的方案为主.Zheng 的第一个签密方案 SCS[2]基于一个短签名 SDSS,效率很高,但在不可否认性的实现上存在缺陷.Bao 等将 SCS 改进为可公开验证的签密[3].文献[4]提出了一个基于韩国数字签名标准 KCDSA 的签密.文献[5]提出了基于数字签名标准 DSA 的签密 SC-DSA.第一个基于 RSA 的签密 TBOS 在 2003 年被提出[6],TBOS 的最大特点在于签密文和普通 RSA 密文或 RSA 签名大小相同.近两年,随着基于身份的密码体制的发展,许多基于身份的签密方案也相继出现[7,8].在椭圆曲线签密方案中,只有 ECSCS[9]是安全的,目前还没有基于标准椭圆曲线签名的方案.

后一个方向来自于认证加密[10,11],主要考虑可证明安全的明文通用填充结构.Krawczyk 证明了加密与签名组合的安全性[1].An 等详细研究了签密的安全问题并提出了一种结合承诺方案的签密结构[12].Dodis 等提出了适用于一般陷门函数的签密结构[13]和几个明文填充方案[14].Dent 将混合密码的一些结果运用到签密的设计[15,16].这些方法侧重于理论安全性,对效率方面并未关注,与实用方案还有一定距离.

签密在需要提供加密和认证功能的场合十分高效,但如果在某些情况下一种安全属性不再需要时,普通签密方案将不再可行.Zheng 建议将算法切换到另外的签名和加密算法.于是系统至少要实现三个方案才能满足要求,这将增加额外的开销.

本文的工作源于以上结果.(1)将签密的应用范围推广到仅需加密或认证的场合,给出了广义签密的定义.(2)设计安全、实用的椭圆曲线签名方案.第三节提出了基于椭圆曲线数字签名标准 ECDSA 的签密方案 SC-ECDSA.第四节基于 ECDSA 安全性假设,在 Random Oracle(随机预言机)模型[17]下对 SC-ECDSA 的安全性进行了形式化证明.最后将 SC-ECDSA 与已经提出的一些典型方案进行了效率比较.

2 签密与广义签密

2.1 签密方案的定义

签密是一种特殊的公钥密码方案,其实质是一个两方协议.执行协议的双方是签密方(发送方)S 和解签密方(接收方)R.S 对消息空间 M 中的一则消息 m 签密产生签密文 ω,R 解密出原始消息同时加以验证.类似于签名,发生争议时第三方可以进行仲裁.以下给出签密的有关定义①.

定义 1. 签密方案 $\Sigma=(Gen,SC,DSC)$ 由三个算法组成:

密钥生成算法 Gen 为用户 U 产生密钥对,$(SDK_U,VEK_U)\leftarrow Gen(U,T)$,$T$ 为安全参数,SDK 为私钥,VEK 为公钥.

签密算法 SC 为概率算法,对于消息 $m\in M$,$\omega\leftarrow SC(m,SDK_S,VEK_R)$,$\omega$ 为签密文.

解签密算法 DSC 为确定算法,对于签密文 ω,$m\cup\{\perp\}\leftarrow DSC(\omega,SDK_R,VEK_S)$,$\perp$ 表示验证失败.

定义 2(正确性). 签密方案 $\Sigma=(Gen,SC,DSC)$ 是正确的①:

$\forall S,R,m\in M$,

$\exists DSC(SC(m,SDK_S,VEK_R),SDK_R,VEK_S)=m$.

Zheng 给出了签密的安全性概念.一个签密方案是安全的,如果满足如下安全属性[2]:

不可伪造性(unforgeability):适应性攻击者(包括接收者)冒充发送方伪造一则签密文在计算上是不可行的.

不可否认性(non-repudiation):发送方想要否认他曾发出的签密文时,第三方进行仲裁在计算上是可行的.

机密性(confidentiality):适应性攻击者(除发送方和接收方外的任何第三方)想要获得关于明文的部分信息在计算上是不可行的.

文献[18]给出了一个签密的形式化证明模型,并证明 SCS 的安全性.

2.2 广义签密

在一些应用系统中,如果既有同时满足机密性和完整性的要求,又有只需一种功能的要求,或者在

① Dodis Y., Signcryption. In: Tilborg V., Henk C. A. ed., Encyclopedia of Cryptography and Security, Berlin: Springer, 2005, http://www.signcryption.net

签密的应用系统中某一种安全属性不再需要时，定义 1 中的签密方案将不再可行. 注意到此时并不仅仅是功能的冗余，而是协议执行者中只有一方是特定的，即只有一方拥有密钥，签密方案将不可行. 文献[2]建议将算法切换到另外的签名或加密算法. 于是系统必须实现三个方案才能满足应用要求，这将增加额外的开销. 既然签密能够将加密和签名的功能综合，那么在必要时也应该能够分开，因此本文将签密的概念进行了推广.

广义签密，即具有更强适应性的签密体制，在要求同时满足机密性和完整性时能够提供加密和签名双重功能，而仅要求机密性/完整性时，无需任何修改和附加计算就可以单独提供加密/签名功能. 也就是说，在特殊情况下签密可以等价于一个签名方案或加密方案. 等价的含义包括效率和安全性两个方面.

于是会存在三种情况：签密、签名、加密. 在公钥密码领域中，执行认证操作需要特定发送方的信息(私钥和公钥)，执行加密操作需要特定接收方的信息(私钥和公钥)，而执行签密需要双方的信息(私钥和公钥)，因此可以用协议执行者的标识来区分. 同时有特定的发送方和接收方时为签密，仅有特定的发送方时为签名，仅有特定的接收方时为加密.

定义 3.　广义签密方案 $\Sigma=(Gen,SC,DSC)$ 由三个算法组成：

Gen 为密钥生成算法，定义同上.

签密算法 SC 为概率算法，对于消息 $m\in M$，$\omega\leftarrow SC(m,SDK_S,VEK_R)$. 当 $R\in\varnothing$ 时，存在 $SC(m,SDK_S,VEK_R)=Sig(m,SDK_S)$，$DSC(\omega,SDK_R,VEK_S)=Ver(\tau,VEK_S)$.

解签密算法 DSC 为确定算法，对于签密文 ω，$m\cup\{\perp\}\leftarrow DSC(\omega,SDK_R,VEK_S)$. 当 $S\in\varnothing$ 时，存在 $SC(m,SDK_S,VEK_R)=Enc(m,VEK_R)$，$DSC(\omega,SDK_R,VEK_S)=Dec(\varepsilon,SDK_R)$.

其中，$ENC=(Gen,Enc,Dec)$ 为加密方案，Gen 同上，$\varepsilon\leftarrow Enc(m,VEK_R)$，$m\leftarrow Dec(\varepsilon,SDK_R)$.

$SIG=(Gen,Sig,Ver)$ 为签名方案，Gen 同上，$\tau\leftarrow Sig(m,SDK_S)$，$\{T,\perp\}\leftarrow Ver(\tau,VEK_S)$，$T$ 表示签名有效，$\perp$ 表示无效.

3　SC-ECDSA：基于椭圆曲线的签密

3.1　ECDSA

ECDSA[19] (Elliptic Curve Digital Signature Algorithm，椭圆曲线数字签名算法)是数字签名标准 DSA 在椭圆曲线上的模拟. Brown 对其安全性作了精确的证明[20]，目前除副本签名(duplicate signature)问题之外[21]还未发现其它缺陷. ECDSA 以其高安全性和高效率的特点，成为最著名的签名算法之一，已被众多的标准化组织作为数字签名标准：ISO 15946-2、ANSI X9.62、IEEE1363-2000、FIPS 186.2、SECG 和 RFC 3278. ECDSA 正在逐步替代 RSA 和 DSA.

3.2　SC-ECDSA

本节将构造一个基于 ECDSA 的广义签密，称之为 SC-ECDSA. 尽管 ECDSA 为 DSA 的模拟，但 SC-ECDSA 的构造技术与文献[5]提出的 SC-DSA 不同. SC-DSA 基于 DSA 的变形 MDSA，将 DSA 的输出 (r,s) 变为 $(h/s,r/s)$. 而 SC-ECDSA 未对 ECDSA 作任何变形.

3.2.1　参数说明

(1) 椭圆曲线域参数

椭圆曲线域参数的选择遵循 SEC1 标准①，描述为一个六元组：$T=(p,a,b,G,n,h)$. 其中 G 为基点，$ord(G)=n$. 无穷远点 O 为群 $\langle G\rangle$ 的零元.

符号说明：

$Q=[x]G$，表示曲线上有理点的标量乘法.

$\|$ 表示消息的级联.

$\in_R$ 表示从集合中随机选择一个元素.

$Bind$ 为关于 Alice 和 Bob 的身份标识.

$\{0,1\}^l$ 表示长度为 l bits 的二进制序列.

$Kenc$，$Kmac$ 和 $Ksig$ 为二进制序列.

(2) Hash 函数

$H:\{0,1\}^*\to Z_p^*$.

$K:Z_p^*\to\{0,1\}^{Z+*}$.

$LH(\cdot):\{0,1\}^*\to\{0,1\}^{l+z}$，输出长消息摘要，可选 SHA-256、SHA-384 或 SHA-512.

$MAC_k:\{0,1\}^l\times\{0,1\}^t\to\{0,1\}^z$，以 k 为密钥的消息认证函数，$|k|=t$.

$|m|=l$，$l+|MAC(\cdot)|=|LH(x_2)|$，此处 $|m|$ 表示序列 m 的长度.

补充定义 $H(0)\to 0$，$K(0)\to 0$，$LH(0)\to 0$，$MAC_0\to 0$.

3.2.2　算法描述

在以下描述中，消息 $m\in\{0,1\}^l$ 将由签密方 Alice

① Standards for Efficient Cryptography Group, SEC1: elliptic curve cryptography Version 1.0. Certicom Research. 2000 http://www.secg.org/

签密并发送给验证方 Bob,Eve 为攻击者.

SC-ECDSA(Gen,SC,DSC)

密钥生成:

Gen(Alice,T)

$d_A \in_R \{1,2,\cdots,n-1\}$;

$Q_A=[d_A]G$; return(d_A,Q_A).

Gen(Bob,T)

$d_B \in_R \{1,2,\cdots,n-1\}$;

$Q_B=[d_B]G$; return(d_B,Q_B);

$(0,O) \leftarrow Gen(U,T), U \in \varnothing$.

签密算法:签密者 Alice 执行签密操作.

$SC(m,d_A,Q_B)$

1. $k \in_R \{1,2,\cdots,n-1\}$;
2. $R \leftarrow [k]G=(x_1,y_1)$; $r \leftarrow x_1 \bmod p$;
3. $[k]Q_B=(x_2,y_2)$;
4. $Kenc \leftarrow LH(x_2)$; $(Kmac,Ksig) \leftarrow K(y_2)$;
5. If $d_A=0$, $s \leftarrow \varphi$;
   Else $s \leftarrow k^{-1}(H(m \parallel Bind \parallel Ksig)+rd_A) \bmod n$;
6. $e \leftarrow MAC_{Kmac}(m)$;
7. $c \leftarrow (m \parallel e) \oplus Kenc$;

Return $\omega=(c,R,s)$.

解验证算法:接收者 Bob 执行

$DSC(\omega,d_B,Q_A)$

1. $r \leftarrow R$;
2. $(x_2,y_2)=[d_B]R$;
3. $Kenc \leftarrow LH(x_2)$; $(Kmac,Ksig) \leftarrow K(y_2)$;
4. $(m \parallel e) \leftarrow c \oplus Kenc$;
5. $e' \leftarrow MAC_{Kmac}(m)$;
   If $e \neq e'$, return $\perp$; else if $s=\varphi$, return m;
6. $u_1 \leftarrow s^{-1}H(m \parallel Bind \parallel Ksig)$; $u_2 \leftarrow s^{-1}r$;
7. $R' \leftarrow [u_1]G+[u_2]Q_A$;
   If $R' \neq R$, return $\perp$; else return m.

3.2.3 公开验证

Alice 想要否认她曾发出的签密文时,由第三方来仲裁. Bob 将 DSC 算法执行至第 4 步,然后公开 $\omega'=(H(m \parallel Bind \parallel Ksig),R,s)$,由第三方执行以下公开验证算法 VP.

$VP(\omega',Q_A)$

1. $u_1 \leftarrow s^{-1}H(m \parallel Bind \parallel Ksig)$; $u_2 \leftarrow s^{-1}r$;
2. $R' \leftarrow [u_1]G+[u_2]Q_A$;

If $R' \neq R$, return $\perp$; else return T.

SC-ECDSA 是目前唯一能够按标准数字签名模式公开验证的签密方案. Bao&Deng 方案[3]是第一个可公开验证的方案,但基于一个非标准签名算法,且公开了消息的 Hash 值 $H(m)$,不具有语义安全性. SC-DSA[5]尽管声称基于标准的 DSA,但验证过程和 DSA 不相同. SC-ECDSA 中(R,s)为 $H(m \parallel Bind \parallel Ksig)$的 ECDSA 签名,公开验证算法和 ECDSA 的验证算法完全相同. SC-ECDSA 用随机密钥 $Ksig$ 填充消息,不会影响机密性.

3.2.4 签名模式和加密模式

在只需要保证消息完整性的环境中执行签名模式. 令接收方 $R=\varphi$, SC-ECDSA 将变为 ECDSA 签名.

$(m,R,s) \leftarrow SC(m,d_A,O)$,

$\{T,\perp\} \leftarrow DSC(\omega,0,Q_A)$.

在只需要保证消息机密性的环境中执行加密模式. 令发送方$S=\varphi$, SC-ECDSA 将变为一个加密算法.

$(c,R) \leftarrow SC(m,0,Q_B)$,

$m \bigcup \{\perp\} \leftarrow DSC(\omega,d_B,O)$.

3.3 SC-ECDSA 的正确性

(1)考虑 $S,R \notin \varnothing$的情况.

令 S 为 Alice, R 为 Bob.

要证明 SC-ECDSA 方案的正确性,即证明 $DSC(SC(m,d_A,Q_B),d_B,Q_A)=m$.

证明.

等式左边为解签密算法:

$DSC(SC(m,d_A,Q_B),d_B,Q_A)=$

$DSC((c,R,s),d_B,Q_A)$.

将解签密算法中的参数记为(x'_2,y'_2),$Kenc'$,$Kmac'$,$Ksig'$.

$[d_B]R \rightarrow [d_Bk]G \rightarrow [k]Q_B=(x'_2,y'_2)$

$\Rightarrow x'_2=x_2$, $y'_2=y_2$

$\Rightarrow Kenc'=Kenc, Kmac'=Kmac, Ksig'=Ksig$

$\Rightarrow e'=e$, $m'=m$.

因此,解签密算法步骤 1~4 正确.

又 $s^{-1}=k(H(m \parallel Bind \parallel Ksig)+rd_A)^{-1}$

令 $h=H(m \parallel Bind \parallel Ksig)$

$\Rightarrow u_1=k(h+rd_A)^{-1}h$, $u_2=k(h+rd_A)^{-1}r$

$\Rightarrow R'=[u_1]G+[u_2]Q_A=[k]G$

$\Rightarrow R=R'$.

因此,解签密算法步骤 5,6 正确.

$DSC(SC(m,d_A,Q_B),d_B,Q_A)=m$. 证毕.

(2)在签名模式下,$R \in \varnothing$,$d_B=0$,$Q_B=O$,算法将简化为 ECDSA 签名.

签名者 Alice 执行签名操作:

$SC(m,d_A,O)$

1. $k \in_R \{1,2,\cdots,n-1\}$;
2. $R=[k]G=(x_1,y_1)$; $r=x_1 \bmod p$;

3. $[k]O=O$; $0\leftarrow LH(0)$; $(0,0)\leftarrow K(0)$;

4. $s\leftarrow[k^{-1}](H(m\|0\|0)+rd_A) \bmod n$;

5. $0=MAC_0(m)$;

6. $m\leftarrow(m\|0)\oplus 0$;

Return $\omega=(m,R,s)$.

任一接收者执行验证操作：

$DSC(\omega,0,Q_A)$

1. $O\leftarrow[0]R$; $0\leftarrow LH(0)$; $(0,0)\leftarrow K(0)$;

2. $m\leftarrow m\oplus 0$;

3. $0\leftarrow MAC_0(m)$;

4. $u_1\leftarrow s^{-1}H(m\|0\|0)$; $u_2\leftarrow s^{-1}r$;

5. $R'=[u_1]G+[u_2]Q_A$;

If $R'\neq R$, return $\perp$; else return T.

显然，将空操作省去，即为ECDSA签名. ECDSA签名的正确性无须证明.

(3) 在加密模式下，$S\in\varnothing$，$d_A=0$，$Q_A=O$，将简化为一个加密算法.

任一加密者 Alice 执行加密操作：

$SC(m,0,Q_B)$

1. $k\in_R\{1,2,\cdots,n-1\}$;

2. $R\leftarrow[k]G=(x_1,y_1)$; $r\leftarrow x_1 \bmod p$;

3. $[k]Q_B=(x_2,y_2)$;

$Kenc\leftarrow LH(x_2)$; $(Kmac,Ksig)\leftarrow K(y_2)$;

4. $e\leftarrow MAC_{Kmac}(m)$;

5. $c\leftarrow(m\|e)\oplus Kenc$;

Return $\omega=(c,R)$.

接收者 Bob 执行解密操作：

$DSC(\omega,d_B,O)$

1. $(x_2,y_2)=[d_B]R$;

$Kenc\leftarrow LH(x_2)$; $(Kmac,Ksig)\leftarrow K(y_2)$;

2. $(m\|e)\leftarrow c\oplus Kenc$;

3. $e'\leftarrow MAC_{Kmac}(m)$;

If $e'\neq e$, return $\perp$; else return m.

若 $DSC(SC(c,R),d_B,O)=m$，则算法正确. 由(1)中对步骤 1～4 的证明可知结论成立.

4　SC-ECDSA 的安全性

签密的安全性概念如 2.1 节所述. 根据攻击者是否包括执行协议的两方，又可定义内部安全(insider security)和外部安全(outsider security)的概念，内部安全更强[12]. 本文将在内部安全的意义上讨论安全性.

签密安全性的形式化证明建立在加密和签名的可证明安全性理论基础之上，文献[6,18]引入了各自的证明技术. 本文将在现有结论的基础上进行归约证明.

定义 4.　椭圆曲线离散对数问题(Elliptic Curve Discrete Logarithm Problem，ECDLP). 有 $x\leftarrow ECDLP(G,Y)$，使得 $Y=[x]G$，其中 G 为椭圆曲线的基点，$x\in[1,2,\cdots,n-1]$，$Y\in\langle G\rangle$.

假设 1(ECDLP 假设). 在参数 T 下，在时间 t'' 内解决 ECDLP 的概率 $\xi=Adv_{\mathrm{ECDLP}}(T,t'')$ 是可忽略的.

假设 2(Random Oracle 假设).　Hash 函数具有 Random Oracle(随机预言机)属性，即 Hash 函数是确定的、有效的函数，输出服从均匀分布.

4.1　不可伪造性

在内部安全的意义下，签密的伪造者为 Bob 和 Eve. Bob 拥有解签密时用到的私钥 d_B，因此 Bob 具有最强的伪造能力. 给定一则从 Alice 发出的对消息 m 的签密，Bob 能够用自己的私钥 d_B 解密 c，于是对签密的伪造转化为对 ECDSA 签名的伪造.

Brown 证明了 ECDSA 的安全性[21]：如果 Hash 函数是理想化的 Random Oracle，则 ECDSA 对主动存在性伪造(UF-CMA，Unforgeability-Chosen Message Attacks)是安全的.

以下将在 ECDSA 的安全性基础上对 SC-ECDSA 的不可伪造性进行证明.

定理 1.　Hash 函数 H 和 MAC 为 Random Oracle，如果存在一个对 SC-ECDSA 的 UF-CMA 攻击者 ASC，能够在安全参数 T 下，通过对签密预言机和随机预言机的 (q_{sc},q_h,q_m) 次询问，在时间 t 内伪造成功，那么也存在一个对 ECDSA 的 UF-CMA 攻击者 AS，能够在安全参数 T 下，通过对签名预言机和随机预言机的 (q_{sc},q_h+q_{sc}) 次询问，在时间 t' 内伪造成功. Adv 为攻击成功的最大概率，则有

$$Adv_{\mathrm{ASC}}^{\mathrm{UF\text{-}CMA}}(T,t,q_{sc},q_h,q_m)\leq Adv_{\mathrm{AS}}^{\mathrm{UF\text{-}CMA}}(T,t',q_{sc},q_h+q_{sc})+2q_h+q_{sc}(q_{sc}-1)/2n.$$

证明.

给出对 SC-ECDSA 的 UF-CMA 攻击实验.

Hash 函数 H 和 MAC 为随机预言机 $Oracle\_H$ 和 $Oracle\_MAC$，各自维护列表 L_H 和 L_{MAC}，保存为每次询问提供的值. 提供一个签密预言机 $Oracle\_SC$，对输入的消息提供签密服务.

Game 0：

$(d_A,Q_A)\leftarrow Gen(A,T)$;

$(d_B,Q_B)\leftarrow Gen(B,T)$;

$Bind\leftarrow A\|B$;

$(m^*, \omega^*) \leftarrow ASC(T, Q_A, Q_B, Oracle\_SC, Oracle\_H, Oracle\_MAC)$;

If $m^* \leftarrow DSC(\omega^*, Q_A, d_B)$,且 ASC 从未向 $Oracle\_SC$ 询问过 m^*,则 ASC 伪造成功.

$Oracle\_SC(m)$

Return $SC(m, d_A, Q_B)$.

$Oracle\_H(m \| Bind \| Ksig)$

If $(m \| Bind \| Ksig, h)$ in L_H, return h;

Else $h \in_R \{0,1\}^{|p|}$, add$(m \| Bind \| Ksig, h)$ to L_H;

Return h.

$Oracle\_MAC(m)$

If (m, e) in L_{MAC}, return e;

Else $e \in_R \{0,1\}^z$, add(m, e) to L_{MAC};

Return e.

以 ASC 作为子程序,模拟对 ECDSA 的 UF-CMA 攻击者 AS:

Game 1:

$AS(T, Q_A, Oracle\_Sign, Oralce\_H)$

$(d_B, Q_B) \leftarrow Gen(R, T)$;

$Bind \leftarrow A \| B$;

$(m^*, \omega^*) \leftarrow ASC(T, Q_A, Q_B, Sim\_SC, Sim\_H, Sim\_MAC)$;

If $m^* \leftarrow DSC(\omega^*, d_B, Q_A)$,且 ASC 从未向签名预言机 $Oracle\_Sign$ 询问过 m^*

$(c^*, R^*, s^*) \leftarrow \omega^*$; return (m^*, R^*, s^*);

Else return $\perp$.

签名预言机 $Oracle\_Sign$ 对 m 提供签名服务,模拟对 m 的签密,Hash 函数 H 为随机预言机 $Oracle\_H$,并维护一个列表 L_H.

$Sim\_SC(m)$

$(r, s) \leftarrow Oracle\_Sign(m)$;

$R \leftarrow r$;

$(x_2, y_2) = [d_B]R$;

$Kenc \leftarrow LH(x_2)$; $(Kmac, Ksig) \leftarrow K(y_2)$;

$e \leftarrow Sim\_MAC_{Kmac}(m)$;

$c \leftarrow (m \| e) \oplus Kenc$;

$h \leftarrow Oracle\_H(m)$, add(m, h) to L_H;

$h' \leftarrow Oracle\_H(m \| Bind \| Ksig)$;

Add$(m \| Bind \| Ksig, h')$ to L_H;

$k \leftarrow ECDLP(T, R)$;

$s' \leftarrow k^{-1}(h' + k\,s - h) \bmod n$;

$\omega \leftarrow (c, R, s')$;

Return ω.

$Sim\_H(m \| Bind \| Ksig)$

$m \leftarrow m \| Bind \| Ksig$;

$h \leftarrow Oracle\_H(m)$, add(m, h) to L_H;

$h' \leftarrow Oracle\_H(m \| Bind \| Ksig)$;

Add$(m \| Bind \| Ksig, h')$ to L_H;

Return h'.

$Sim\_MAC(m)$

If (m, e) in the list L_{MAC}, return e;

Else $e \in_R \{0,1\}^z$, add(m, e) to L_{MAC}, return e.

Sim_SC 产生的 h 可能和 Sim_H 碰撞. 当 ASC 已经向 Sim_H 询问 q_h 次后,向 Sim_SC 作 q_{sh} 次询问. h 表示发生碰撞,$\neg h$ 表示未发生,$Prob$ 表示概率,产生碰撞的概率为

$$Prob(h) = 2q_h + q_{sc}(q_{sc} - 1)/2n.$$

ASC wins 表示事件 ASC 攻击成功,AS wins 表示事件 AS 攻击成功,$ECDLP$ wins 表示 $ECDLP$ 成功解决. 显然,$ECDLP$ wins 与 AS wins 相关,$ECDLP$ 的难度远大于 AS wins,$ECDLP$ wins$\subseteq$ AS wins.

事件 h 不发生时存在以下关系:

$$\begin{aligned}
&Prob(ASC \text{ wins} | \neg h)\\
&= Prob(ECDLP \text{ wins} \vee (AS \text{ wins} | \neg h))\\
&= Prob(AS \text{ wins} | \neg h).
\end{aligned}$$

于是 AS wins 的概率为

$$\begin{aligned}
&Prob(ASC \text{ wins})\\
&= Prob(ASC \text{ wins} \wedge \neg h) +\\
&\quad Prob(ASC \text{ wins} \wedge h)\\
&\leq Prob(ASC \text{ wins} | \neg h) Prob(\neg h) + Prob(h)\\
&= Prob(AS \text{ wins} | \neg h) Prob(\neg h) + Prob(h)\\
&= Prob(AS \text{ wins} \wedge \neg h) + Prob(h)\\
&= Adv_{AS}^{\text{UF-CMA}}(T, t', q_{sc}, q_h + q_{sc}) +\\
&\quad 2q_h + q_{sc}(q_{sc} - 1)/2n
\end{aligned}$$

又 ASC wins 是在安全参数 T 下,通过对签密预言机和随机预言机的(q_{sc}, q_h, q_m)次询问,在时间 t 内发生的,因此有

$$Adv_{ASC}^{\text{UF-CMA}}(T, t, q_{sc}, q_h, q_m) = Prob(ASC \text{ wins}).$$

证毕.

如果 ECDSA 对 UF-CMA 是安全的,则 SC-ECDSA 对 UF-CMA 也是安全的.

4.2 不可否认性

签密的不可否认性与签名的不可否认性相同. 如果不存在副本签密[21],则不可伪造性蕴含不可否认性. 若可能伪造签密,则 Alice 有机会否认,因为 Bob 没有足够的证据证明签密来自 Alice.

由于椭圆曲线两个对称点的 x 坐标相同:$R = (x, y)$,$-R = (x, -y)$,ECDSA 中曲线上的点集到模 n 整数集的映射 f:$R \rightarrow r$ 不是一一映射,对三元组(m_1, R, s)和$(m_2, -R, s)$能够得到相同的签名文

本(r,s)，从而产生了副本签名. SC-ECDSA 中映射 $f: R\to R$ 是点的集合到点的集合的一一变换，SC-ECDSA 签密是消息空间到签密文本空间的一一映射，因此不存在副本签密.

SC-ECDSA 的不可否认性通过公开验证三元组$(H(m\parallel Bind\parallel Ksig),R,s)$来实现. 因此，SC-ECDSA 满足不可否认性要求.

4.3　机密性

本节将构造一个可证明安全的加密方案，然后证明 SC-ECDSA 和该加密方案具有相同的机密性.

文献[1]研究证明了 AtE(Authentication then Encryption)在 CBC(Cipher Block Chaining)和 OTP(One Time Padding，用伪随机数填充数的流密码异或加密方式)方式下是 CUF-CPA(Ciphertext UnForgeable-Chosen Plaintext Attacks)安全的.

定义 5(CUF-CPA).　加密方案是 CUF-CPA 安全的，如果不存在多项式时间的密文伪造者 F，通过向加密预言机 Oracle_E(使用密钥 k)询问除 m^* 之外的任何消息而成功伪造由 k 加密的密文. 即，F 在时间 t 内询问 q 次 Qbits 的明文，成功伪造合法密文的最大概率 $E(q,Q,t)$是可忽略的. CUF-CPA 蕴含 IND-CCA(Indistinguishability-chosen plaintext attacks)，可以实现安全的信道.

定义 6(加密方案 OTP(F)).　以一个函数 $f\in F$ 作为密钥，$r\in_R\{0,1\}^l$，计算 $c=f(r)\oplus x$，密文为 (r,c). 其中，$F=\{f\mid f:\{0,1\}^l\to\{0,1\}^{l'}\}$，$x\in M$.

如果 f 是从 F 中随机选择的，且每次选择的 r 不重复，将其记作 $OTP_\$$.

定义 7.　AtE($OTP_\$$, MAC)组合为：(i)计算 $t=MAC_k(x)$；(ii)将 t 填充至 x 后；(iii)输出 $c=f(r)\oplus(x\parallel t)$. 其中 $MAC_k:\{0,1\}^*\times\{0,1\}^t\to\{0,1\}^n$，$|k|=t$.

引理 1.　如果消息认证函数 MAC 对 IND-CMA(Indistinguishability — Chosen Message Attacks)是安全的，则 AtE($OTP_\$$, MAC)是 CUF-CPA 安全的.

证明见文献[1].

构造一个加密方案 ENC：输入为明文 m，接收者的公钥为 $Q_B=[d_B]G$，G 为基点，$ord(G)=n$，LH 为输出 $l+|n|$ bits 的 Hash 函数，H 为输出 l bits 的 Hash 函数.

加密：

Alice 作如下计算

1. $k\in_R\{1,2,\cdots,n-1\}$；

2. $(x_1,y_1)=R\leftarrow[k]G$；

3. $(x_2,y_2)=[k]Q_B$；

4. $Kenc\leftarrow LH(x_2)$，$(Kmac,Ksig)\leftarrow K(y_2)$；

5. $e\leftarrow H(m\parallel Kmac)$；

6. $c\leftarrow(m\parallel e)\oplus Kenc$；

Return (c,R).

解密：

Bob 计算如下

1. $[d_B]R=(x_2,y_2)$；

2. $Kenc\leftarrow LH(x_2)$，$(Kmac,Ksig)\leftarrow K(y_2)$；

3. $(m\parallel e)\leftarrow c\oplus Kenc$；

4. $e'\leftarrow H(m\parallel Kmac)$

If $e\neq e'$, return $\perp$; else, return m.

引理 2.　ENC 加密方案是 CUF-CPA 安全的.

证明.

定义两个函数：$x(R)=R_x$表示取椭圆曲线上点 R 的 x 坐标，$E(x)=R$ 表示将整数 x 嵌入到椭圆曲线上，得到一个点 R.

在 ENC 方案中，令 $r=x(R)=x_1$，$R=[k]G$，由于 k 随机且不重复，故 r 随机且不重复. 令$f(\cdot)=LH(x([d_B]E(\cdot)))$，此处$[d_B]E(\cdot)$ 表示点的标量乘. 由于 s_B 是保密且随机选择的，故 $f(\cdot)$ 是一个保密且随机选择的函数.

$$\begin{aligned}f(r)&=LH(x([d_B]E(r)))\\&=LH(x([d_B]E(x_1)))\\&=LH(x([d_B]R))\\&=LH(x_2)\\&=Kenc.\end{aligned}$$

$Kenc$ 正是 ENC 中的加密密钥.

$Kmac$ 为双方均可计算的鉴别密钥.

故 ENC 是一个 AtE($OTP_\$$, MAC)的组合.

H 为 Random Oracle，是 IND-CMA 安全的.

由引理 1，ENC 是 CUF-CPA 安全的.　证毕.

定理 2.　ENC 和 SC-ECDSA 具有相同的机密性.

证明.

对 SC-ECDSA 的攻击者能够获得的公开数据有(T,Q_A,Q_B,R,c,s). 此外还能够计算$[H(m\parallel Bind\parallel Ksig)]G=[r]Q_A-[s]R$，其中 $r=x_1=x(R)$，令 $h=H(m\parallel Bind\parallel Ksig)$.

对 ENC 的一个适应性攻击者，能够获得的公开数据有(T,Q_B,R,c). 还可以将$[h]G$ 再公开，在 ECDLP 假设和 Random Oracle 假设之下，$[h]G$ 可以隐藏关于 m 的所有部分信息，因此公开$[h]G$ 不会降低对 ENC 的攻击难度.

假定 $AENC$ 是一个对 ENC 的攻击算法，输入为 $(T, Q_B, R, c, [h]G)$. 输出为 m 的部分信息 $\tilde{m}$. ASC 是一个对 SC-ECDSA 的攻击算法，输入为 (T, Q_A, Q_B, R, c, s). 输出为 m 的部分信息 $\tilde{m}$. 构造确定性多项式时间算法实现 ASC 和 $AENC$ 之间的相互归约.

$AENC$ 到 ASC，构造如下算法：

$ASC(T, Q_A, Q_B, R, c, s)$

1. 计算 $[h]G=[r]Q_A-[s]R$；
2. $\tilde{m} \leftarrow AENC(T, Q_B, R, c, [h]G)$；
3. Return $\tilde{m}$.

如果 $AENC$ 能够获得关于消息 m 的任何部分信息，则 ASC 也可以.

ASC 到 $AENC$，构造如下算法：

$AENC(T, Q_B, R, c, [h]G)$

1. $s \in_R \{1, 2, \cdots, n\}$；
2. 计算 $Q_A=[r^{-1}s]R-[r^{-1}h]G$；
3. $\tilde{m} \leftarrow ASC(T, Q_A, Q_B, R, c, s)$；
4. Return $\tilde{m}$.

如果 ASC 能够获得关于消息 m 的任何部分信息，则 $AENC$ 也可以.

SC-ECDSA 和 ENC 具有同样的机密性.

由引理 2 可知，SC-ECDSA 可以实现安全的信道. 证毕

5 SC-ECDSA 的效率

本节将对 SC-ECDSA 和其它典型签密方案的效率进行比较. 典型方案共有 6 个：基于离散对数的 SCS[2]、Bao&Deng[3]、KCDSA[4] 和 SC-DSA[5]；基于 RSA 的 TBOS[6]；基于椭圆曲线的 ECSCS[9].

5.1 计算复杂性

在公钥密码体制中，有限域上的模乘、模幂(指数)、模逆和椭圆曲线上点的标量乘计算复杂度较高. 相比之下，有限域上数的加法、Hash、对称加/解密所消耗的时间可以忽略. 在此，只以各方案中复杂度较高的运算的数量来衡量计算复杂性. 表 1 给出了 SC-ECDSA 和其它几种体制的比较.

(1) 与基于离散对数的方案的比较

SCS 是所有四个基于离散对数的方案中最快的一种(SCS, B&D, KCDSA 和 SC-DSA). 根据文献[22]的估计，同等安全条件下椭圆曲线上点的标量乘大约是模幂运算量的 1/8. 根据此结果，SC-ECDSA 密钥生成的运算量约为 SCS 的 1/8；SC-ECDSA 签密操作的运算量约为 SCS 的 1/4，解签密的运算量约为 SCS 的 1/5. 总体来说，SC-ECDSA 的运算量大约比 SCS 节省了 78%.

表 1 计算量比较

Schemes	KG	S	U	AC	VP
SCS	2E	1E+1I	2E	/	/
ECSCS	2kP	1kp+1I	2kP	/	/
B&D	2E	2E+1I	3E	0	2E
KCDSA	2E	2E	3E	save r, sor 3E	2E
SC-DSA	2E	2E+2I	3E+1I	save r, s or 2E+1I	2E+1I
TBOS	2E+2I	2E	2E	0	E
SC-ECDSA	2kP	2kP+1I	3kP+1I	0	2kP+1I

注：(i)“KG”指密钥生成算法；“S”指签密算法；“U”指解签密算法；“VP”指公开验证算法；“AC”指为公开验证而进行的附加计算.

(ii)“E”模幂(指数)运算；“I”指求逆运算；“kP”指椭圆曲线上点的标量乘法. “/”表示功能不具备.

(2) 与基于 RSA 的 TBOS 方案的比较

根据文献[22]的结果，SC-ECDSA 的密钥生成算法和签密算法的计算量约为 TBOS 的 1/8，解签密的运算量约为 TBOS 的 1/5，公开验证的计算量约为 TBOS 的 1/4. 总体来说，SC-ECDSA 的运算量大约比 TBOS 节省了 82%.

(3) 与 ECSCS 的比较

ECSCS 是唯一的基于椭圆曲线的方案，SC-ECDSA 的计算量略高于 ECSCS. 密钥生成算法相同，SC-ECDSA 的签密算法是 ECSCS 的 2 倍，解签密算法是 ECSCS 的 1.5 倍.

综上比较，SC-ECDSA 是以上所有可公开验证的方案中最快的.

5.2 通信(存储)效率

对消息加密或签名后一般会产生数据膨胀. 数据膨胀越小，效率越高.

定义 8(数据率). 签密系统 S 的数据率为 $DR(S)=|m|/|C_\Sigma|$. 明文为 m，$|m|$ 表示明文 m 的长度，C_Σ 表示所有密文(包括为验证和解密而传输的所有附加信息).

表 2 给出了几种方案的数据量. 数据量与各种密码体制的安全参数有关.

能够达到当前基本安全性能的推荐参数为：基于离散，有 $|\alpha|=1024$bits，$|q|=160$bits；RSA，$|N|=1024$bits；ECDLP①，$|p|=131$bits，$|n|=160$bits. 分组密码的分组长度为 64bits(如 IDEA). Hash 函数的长度为 128bits(如 MD5). 长输出 Hash 函数可选 384bits(如 SHA-384). 选择以上参数时各方案的消息数据率见表 3 中 DR1 列.

① http://www.certicom.com/download/aid-111/cert_ecc_challenge.pdf

表 2　数据量比较

Schemes	m	C_Σ
SCS	$\|D(\cdot)\|$	$\|D(\cdot)\|+\|KH(\cdot)\|+\|q\|$
ECSCS	$\|D(\cdot)\|$	$\|D(\cdot)\|+\|h\|+\|n\|$
B&D	$\|D(\cdot)\|$	$\|D(\cdot)\|+\|h(\cdot)\|+\|q\|$
KCDSA	$\|D(\cdot)\|$	$\|D(\cdot)\|+\|h(\cdot)\|+\|q\|$
SC-DSA	$\|D(\cdot)\|$	$\|D(\cdot)\|+2\|q\|$
TBOS	$\|N\|-\|h(\cdot)\|-\|G(\cdot)\|$	$\|N\|$
SC-ECDSA	l	$\|n\|+\|LH(\cdot)\|+2\|p\|$

注：(i) 基于离散对数的方案(SCS、B&D、KCDSA、SC-DSA)：$|\alpha|$ 指有限域的规模，$|q|$ 为生成元阶的规模.

(ii) 基于 RSA 的方案(TBOS)：$|N|$ 表示模数长度，$|G(\cdot)|$ 表示 TBOS 中的 Hash 函数长度.

(iii) 基于 ECDLP 的方案(ECSCS、SC-ECDSA)：$|p|$ 表示有限域 Fp 的规模，$|n|$ 表示基点阶的规模.

(iv) $|D(\cdot)|$ 表示分组密码的分组长度. $|h|$ 表示 Hash 函数的输出长度. $|LH(\cdot)|$ 表示长 Hash 函数的输出长度. $|KH(\cdot)|$ 为 SCS 中 Hash 函数的输出长度，同 $|h|$.

能够达到长期安全性能的推荐参数为：离散对数，有 $|\alpha|=2048$bits，$|q|=192$bits；RSA，$|N|=2048$bits. ECDLP，$|p|=191$bits，$|n|=192$bits. 分组密码的分组长度为 128bits(如 AES)；Hash 函数长度为 160bits(如 SHA-1). 长输出 Hash 函数的输出长度为 512bits(如 SHA-512). 选择以上安全参数时各方案的数据率见表 3 中 DR2 列.

表 3　数据率比较

Schemes	DR1(%)	DR2(%)
SCS	18	26
ECSCS	18	26
B&D	18	26
KCDSA	18	26
SC-DSA	17	25
TBOS	50	67
SC-ECDSA	32	35

可见，除基于 RSA 的 TBOS 方案之外，所有 ElGamal 型方案中 SC-ECDSA 具有最高的数据率.

6　结束语

广义签密的目的是以少量密码组件适应更广泛的应用需求，因此要求在不增加任何额外开销的情况下能够提供签密、单独的加密或签名功能. 广义签密方案主要考虑的问题在于两个方面：(1)有效的区分方法；(2)运算结构的特殊性质对特定输入的适应性. 本文的方法是将一方标识置空，算法在接收到空参数后能够屏蔽或跳过某些操作. 比如 SC-ECDSA 中采用的异或加密结构，密钥为 0 时能够屏蔽加密功能，输出明文本身. SC-ECDSA 的加密结构也可以换为对称加密算法. 但其它加密结构输入空参数时有可能出现弱密钥，安全性需要仔细考虑.

本文的 SC-ECDSA 是一个高效的签密方案，具有以下几个优势：(1)基于标准的数字签名 ECDSA；(2)很高的计算和通信(存储)效率；(3)与 ECDSA 相同的不可伪造性、CUF-CPA 的机密性；(4)具备广义签密的特征，应用范围更广.

致　谢　本文的想法源于胡予濮教授的提示和签密发明人 Zheng Yuliang 教授的肯定，在此表示感谢，并且感谢审稿人的修改意见！

参　考　文　献

1 Krawczyk H.. The order of encryption and authentication for protecting communications(or: How secure is SSL?). In: Kilian J. ed.. Proceedings of Advances in Cryptoloty-CRYPTO2001. Lecture Notes in Computer Science 2139. Berlin: Springer-Verlag, 2001, 310～331

2 Zheng Y.. Digital signcryption or how to achieve cost(signature &encryption)<<cost(signature)+cost(encryption). In: Kaliski B. S. ed.. Proceedings of Advances in Cryptoloty-CRYPTO'97. Lecture Notes in Computer Science 1294. Berlin: Springer-Verlag, 1997, 165～179

3 Bao F., Deng R. H.. A signcryption scheme with signature directly verifiable by public key. In: Imai H., Zheng Y. ed.. Proceedings of the Public Key Cryptography'98, Lecture Notes in Computer Science 1431, Berlin: Springer-Verlag, 1998, 55～59

4 Yum D. H., Lee P. J.. New Signcryption Schemes based on KCDSA. In: Proceedings of the 4th International Conference on Information Security and Cryptology, Seoul, Korea, 2002, 305～317

5 Shin J. B., Lee K., Shim K.. New DSA-verifiable signcryption schemes. In: Proceedings of the 5th International Conference on Information Security and Cryptology, Seoul, Korea, 2003, 35～47

6 Malone-Lee J., Mao W.. Two birds one stone: Signcryption using RSA. In: Joye M. ed.. Proceedings of the Topics in Cryptology-Cryptographers' Track, RSA Conference 2003. Lecture Notes in Computer Science 2612. Berlin: Springer-Verlag, 2003, 210～224

7 Boyen X.. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In: Proceedings of Advances in Cryptology-Crypto'03. Lecture Notes in Computer Science 2729. Berlin: Springer-Verlag, 2003, 382～398

8 Libert B., Quisquater J.. Efficient signcryption with key privacy form gap Diffie-Hellman group. In: Bao Feng ed.. Proceedings of the Public Key Cryptography-PKC'04. Lecture Notes in Computer Science 2947. Berlin: Springer-Verlag, 2004, 187～200

9 Zheng Y., Imai H.. How to construct efficient signcryption schemes on elliptic curves. Information Processing Letters, 1998, 68(5): 227～233

10 Bellare M., Namprempre C., Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto T. ed.. Proceedings of Advances in Cryptology-ASIACRYPT2000. Lecture Notes in Computer Science 1976. Berlin: Springer-Verlag, 2000, 531～545

11 Rogaway P.. Authenticated-encryption with associated-data. In: Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS2002), Washingtion DC, USA, 2002, 98～107

12 An J. H., Dodis Y., Rabin T.. On the security of joint signature and encryption. In: Knudsen L. ed.. Proceedings of Advances in Cryptology-EUROCRYPT2002. Lecture Notes in Computer Science 2332. Berlin: Springer-Verlag, 2002, 83～107

13 Dodis Y., Rreedman M., Jarecki S., Walfish S.. Optimal signcryption from any trapdoor permutation. Cryptology ePrint Archive, Report 2004/020, 2004

14 Dodis Y., Rreedman M., Jarecki S., Jarecki S., Walfish S.. Versatile padding schemes for joint signature and encryption. In: Pfitzmann B. ed.. Proceedings of the 11th ACM Conference on Computer and Communication Security (CCS2004), Washington DC, USA, 2004, 196～205

15 Dent Alexander W.. Hybrid signcryption schemes with outsider security. In: Proceedings of the 8th Information Security Conference (ISC 2005), Singapore, 2005, 203～217

16 Dent Alexander W.. Hybrid signcryption schemes with insider security. In: Proceedings of the Information Security and Privacy-ACISP 2005, Brisbane, Australia, 2005, 253～266

17 Bellare M., Rogaway P.. Random oracle are practical: A paradigm for designing efficient protocols. In: Proceeding of the 1st ACM Conference on Computer and Communication Security (CCS1993), Fairfax, Virginia, USA, 1993, 62～73

18 Baek J., Steinfeld R., Zheng Y.. Formal proofs for the security of signcryption. In: Naccache D., Paillier P. eds. Proceedings of the Public Key Cryptography'02. Lecture Notes in Computer Science 2274. Berlin: Springer-Verlag, 2002, 80～98

19 Johnson D., Menezes A.. The elliptic curve digital signature algorithm (ECDSA). Department of C&O, University of Waterloo, Technical Report CORR 99-34, 1999

20 Brown D.. Generic groups, collision resistance, and ECDSA. Design, Codes Cryptography, 2005, 35(1): 119～152

21 Stern J., Pointcheval D., Malone-Lee J., Smart Nigel P.. Flaws in applying proof methodologies to signature schemes. In: Yung Moti ed.. Advances in Cryptology-Crypto'02. Lecture Notes in Computer Science 2442. Berlin: Springer-Verlag, 2002, 93～110

22 Koblitz N., Menezes A., Vanstone S.. The state of elliptic curve cryptography. Designs, Codes and Cryptography, 2000, 30(19): 173～193

HAN Yi-Liang, born in 1977, M.S., lecturer. His research interests include elliptic curve cryptography, computer networks and information security.

YANG Xiao-Yuan, born in 1959, professor. His research interests include cryptography and information hiding.

Background

This work is supported in part by the National Natural Science Foundation of China with the title "Design and Analysis on Novel Public Key Cryptography" (grant No. 60473029).

Signcryption gives an efficient solution to transmit messages confidentially and authentically, which is being considered as an IEEE standard (P1363-3). Plenty of schemes and message padding methods were designed since Zheng proposed SCS in 1997. Unfortunately, all of the known schemes failed to take the applications only requesting for confidentiality or authenticity into account. There is no scheme that can be verified in the method of standard elliptic curve based signature also. This paper is motivated by the above results. Generalized signcryption is defined, which can provide separate signature and encryption besides common functions. SC-ECDSA is the first ECDSA-verifiable scheme.