# Cryptlib 密码服务库改进与实现

李 林, 杨先文
(解放军信息工程大学 电子技术学院, 郑州 450004)

**摘 要**: 介绍了一种密码服务库,指出了其缺少椭圆曲线加密算法组件,并对其进行了相应的改进,将椭圆曲线集成加密方案(elliptic curve integrated encryption scheme,ECIES)加入到核心密码算法组件库之中。正确设计了 ECIES 加密体制结构,实现了 ECIES 组件的功能调用,提升了该密码库密码服务能力的完整性。

**关键词**: 密码; 服务; 椭圆曲线; 算法组件; 加密方案; 完整性
**中图分类号**: TP309 **文献标志码**: A **文章编号**: 1001-3695(2012)12-4662-04
doi:10.3969/j.issn.1001-3695.2012.12.067

# Improvement and realization of cryptography service library Cryptlib

LI Lin, YANG Xian-wen
(*Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou* 450004, *China*)

**Abstract**: This paper introduced one cryptography service library, presented one disadvantage existed, the absence of elliptic curve encryption algorithm component, presented the corresponding improvement, embedded elliptic curve integrated encryption scheme into the library of core cryptography algorithm components. It designed the instruction of ECIES, which was proved to be right, implemented the function call of ECIES component. All of this improve the completeness of cryptography service competence provided by the library.

**Key words**: cryptography; service; elliptic curve; algorithm component; encryption scheme; integrity

密码服务作为一种服务理念,已经逐渐被人们所接受。密码服务[1]是一种由系统提供的安全服务实现,为保护系统资源的应用,提供加密、解密、签名认证等密码函数支持。伴随着人们对于密码需求的不断提升,密码服务已经进入了集成密码服务阶段,趋向于标准化、规范化、系统化和平台化。密码服务库是一种精心设计的密码文件库,目的是为了解决密码服务问题。Cryptlib 是新西兰 Gutmann[2]设计的一个密码服务库。该库采用了面向对象设计概念,并引入内核机制。在应用方面,该库可以通过不同的配置满足不同操作系统的密码需求。

密码服务满足的是密码需求,公钥密码系统在现今世界的密码应用中扮演着举足轻重的角色,安全、高效的公钥体制密码算法是对密码研究人员的主要诉求之一。ECC(elliptic curve crytosystem)于 1985 年由 Miller 和 Koblitz 分别提出,两人均将其简记为 ECC[3]。自 1985 年以来,ECC 接受着来自全世界的密码学家、数学家、计算机学家的严格审查。一方面,没有发现明显弱点的事实增强了人们对于 ECC 安全性的信心;另一方面,伴随着系统效率的不断提升,就现在而言,ECC 不仅变为现实,而且成为已知最高效的公钥密码系统。

## 1 Cryptlib 基础知识

Cryptlib 密码库的结构遵循一定的设计准则[2]来实现,具有对象独立、对象智能、平台无关、内外部代码完全隔离和层次化等优点。其结构分为三个层次,提供三层接口,分别针对不同的服务需求。其体系结构如图 1 所示。

各种密码算法构成密码库的核心组件,密码库以一定的形式组合各密码组件,来完成其密码服务能力,如密码库需调用下层的公钥加密算法来实现上层的安全数字信封。由表 1 可知,Cryptlib 密码库核心密码组件的公钥加密算法中仅包含 Elgamal 和 RSA,并不包含椭圆曲线加密体制。而椭圆曲线密码作为最高效的公钥密码系统,有着广泛的应用。椭圆曲线加密体制的缺失使得密码服务库存在着密码服务功能不全,能力不足的缺陷,从而限制了其应用广度。

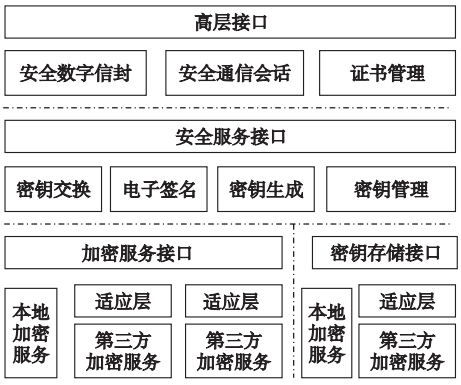


**图1 Cryptlib体系结构**

## 2 椭圆曲线加密体制

### 2.1 加密方案

ECC 的实现存在软件和硬件两种方式[4],考虑到 Cryptlib 密码库为软件实现,本文只讨论软件实现的椭圆曲线加密方案。同时,基于椭圆曲线的加密方案有很多,本文只挑出几种

**收稿日期**: 2012-05-30; **修回日期**: 2012-07-08

**作者简介**:李林(1986-),男,硕士,主要研究方向为密码系统设计与分析(274001032@qq.com);杨先文(1983-),男,博士,主要研究方向为密码系统设计与分析.

常用且比较完善的加密方案,包括椭圆曲线 ELGmal 加密方案(elliptic curve ELGmal,ECELG)、椭圆曲线集成加密方案(elliptic curve integrated encryption scheme,ECIES)[5]、可证明安全的加密曲线方案(provably secure encryption curve scheme,PSEC)[6]。其中,ECELG 是基于椭圆曲线的 ELGmal 方案,ECIES 是椭圆曲线 ELGmal 加密方案的一种变形,PSEC 是 PSEC-KEM(key encapsulation mechanism)[7] 和 DEM(data encapsulation mechanism)的结合。三种加密方案的主要区别在于:ECELG 需要明文嵌入过程[8],加解密过程相对简单;后两种加密算法都不需要明文嵌入过程,具体过程可能更复杂一些。

表 1　Cryptlib 算法支持

| 算法 | 密钥规模 | 分组规模 |
|---|---|---|
| AES | 128/192/256 | 128 |
| Blowfish | 448 | 64 |
| CAST-128 | 128 | 64 |
| DES | 56 | 64 |
| Triple DES | 112/168 | 64 |
| IDEA | 128 | 64 |
| RC2 | 1024 | 64 |
| RC4 | 2048 | 8 |
| RC5 | 832 | 64 |
| Skipjack | 80 | 64 |
| MD2 | — | 128 |
| MD4 | — | 128 |
| MD5 | — | 128 |
| RIPEMD-160 | — | 160 |
| SHA-1 | — | 160 |
| SHA-2/SHA256 | — | 256 |
| HMAC-MD5 | 128 | 128 |
| HMAC-SHA1 | 160 | 160 |
| HMAC-SHA2 | 256 | 256 |
| HMAC-RIPEMD-160 | 160 | 160 |
| Diffie-Hellman | 4096 | — |
| DSA | 4096 | — |
| ECDSA | 521 | — |
| ECDH | 521 | — |
| Elgamal | 4096 | — |
| RSA | 4096 | — |

本文选取 ECIES 作为椭圆曲线加密体制实现的标准方案。选取 ECIES 的原因主要有两点:a)出于性能考虑,明文嵌入过程是所有需要此过程的椭圆曲线加密体制中最为耗时的部分[9],有时为了提升性能,需要设计专门的密码协处理器,因而综合考虑多种应用环境下的性能问题,不需要明文嵌入过程的加密体制更高效一些;b)应用性,ECIES 被纳入相关国际标准且被广泛应用,目前已知标准化于 ANSI X9.63[5]、ISO/IEC 15946-3[6]、18033-2[7] 标准和 IEEE P1363a[10] 草案标准中。因此,综合性能考虑和应用广度,本文选择 ECIES 作为 Cryptlib 内部实现的椭圆曲线加密方案。

### 2.2　ECIES 加密体制

1)加密体制过程需要用到的元素定义

域参量 $D=(q,FR,S,a,b,P,n,h)$ 的组成:域的秩 $q$;标志 $FR$(field representation)对应 $F_q$ 的名称;一个种子 $S$ 判明椭圆曲线是否为随机生成;$a$、$b$ 为椭圆曲线方程中的两个系数;$p_x$、$p_y$ 为两个域元素,定义一个有限点 $P$;$P$ 的秩 $n$;余因子 $h=\#E(F_q)/n$。

2)加密体制过程相关缩写动作定义

a)KDF(key derive function)表示一个由 hash 函数构成的密钥生成函数(若需要 $l$ bit 的密钥,KDF($s$)为哈希值 $H(s,i)$ 的拼接,其中 $i$ 表示哈希算法的应用次数,直到生成 $l$ bit 的密钥)。

b)ENC 表示某一对称加密算法的加密过程。

c)DEC 表示相应的解密过程。

d)MAC 表示某一消息认证码函数。

3)加密过程

输入:域参量 $D=(q,FR,S,a,b,P,n,h)$,公钥 $Q$,明文 $m$。

输出:密文 $(R,C,t)$。

a)随机选取[1,$n-1$]之间的一个整数 $k$;

b)计算 $R=kP$,$Z=hkQ:(z_x,z_y)$,若 $Z=\infty$,则返回到 a);

c)$(k_1,k_2)\leftarrow \mathrm{KDF}(z_x,R)$;

d)分别计算 $C=\mathrm{ENC}(k_1,m)$,$t=\mathrm{MAC}(k_2,C)$;

e)返回 $(R,C,t)$。

4)解密过程

输入:域参量 $D=(q,FR,S,a,b,P,n,h)$,私钥 $d$,密文 $(R,C,t)$。

输出:明文 $m$ 或密文退回。

a)计算 $Z=hdR$,如果 $Z=\infty$ 则返回密文无效;

b)$(k_1,k_2)\leftarrow \mathrm{KDF}(z_x,R)$;

c)计算 $t'=\mathrm{MAC}(k_2,C)$,若 $t'\neq t$,则返回密文无效;

d)$m=\mathrm{DEC}(k_1,C)$。

## 3　密码服务库改进

本文的目标是要将椭圆曲线加密体制加入到密码库中,即将 ECIES 加密体制融入密码库。为了在实现功能的同时,可以融入密码库,设计如下:

### 3.1　结构设计

1)定义算法标志符 CRYPT_ALGO_ECIES

Cryptlib 密码库对密码算法的调用是通过算法标志符来进行的,本文定义算法标志符 CRYPT_ALGO_ECIES,用于对 ECIES 算法的调用。

2)定义密钥数据结构 CRYPT_PKCINFO_ECIES

Cryptlib 中定义了密钥数据结构,用来为加密上下文装载密钥,定义 ECIES 密码体制密钥结构 CRYPT_PKCINFO_ECIES,用于为 ECIES 装载密钥。密钥结构 CRYPT_PKCINFO_ECIES 表示参量集合 $(q,FR,S,a,b,P,n,h,Q,d)$,各参量分别表示如下:域的秩 $q$;标志 $FR$ 为对应 $F_q$ 的名称;种子 $S$ 判明椭圆曲线是否为随机生成;$a$、$b$ 为椭圆曲线方程中的两个系数;$p_x$、$p_y$ 表示域中两个元素,用于定义一个曲线原点 $P$;$P$ 的秩 $n$;余因子 $h=\#E(F_q)/n$;$Q_x$、$Q_y$ 表示域中两个元素,用于定义公钥 $Q$,即曲线上一点;私钥 $d$。

密钥数据结构的域参量可分为椭圆曲线结构(由前八个参量组成)和公私钥对(后两个参量)两个部分。本文预先定义了五个椭圆曲线,其 $FR$ 分别为 CRYPT_ECCFR_P192、CRYPT_ECCFR_P224、CRYPT_ECCFR_P256、CRYPT_ECCFR_P384 和 CRYPT_ECCFR_P521。当使用这些预定义的曲线时,只需设置相应的公钥 $Q$(加密时)或私钥 $d$(解密时),即可完成对密钥结构的设置。

3)ECIES 密码体制的实现

本文使用 C 语言编写实现了 ECIES 密码体制,其具体实现分为三个层次,包括数学基础、密码组件、加密方案。具体编程实现中,本文定义了 ECIES 中所需用到的有限域,即所需的数学基础,包括有限域 $F_p$、$F_2^m$ 的实现,以及建立在它们之上的椭圆曲线,还包含了数据类型及转换;密码组件包括椭圆曲线

域参量、椭圆曲线密钥对的实现，两者的实现即组成了密钥数据结构 CRYPT_PKCINFO_ECIES 的实现。椭圆曲线域参量分为两类：a）$F_p$ 上椭圆曲线域参量 $T=(p,a,b,P,n,h)$，各参量意义为，椭圆曲线方程式 $y^2=x^3+ax+b(\mathrm{mod}\ p)$，$P$ 为椭圆曲线 $E(F_p)$ 上一个基点，$P$ 的秩为素数 $n$，整数 $h$ 是一个余因子 $h=\#E(F_p)/n$；b）$F_2^m$ 上椭圆曲线域参量 $T=(m,f(x),a,b,P,n,h)$，各参量意义为：$m$ 定义有限域 $F_2^m$，一个秩为 $m$ 的不可约多项式 $f(x)$，椭圆曲线方程式为 $y^2+xy=x^3+ax^2+b$，$P$、$n$、$h$ 同上。椭圆曲线密钥对 $(d,Q)$，秘密密钥 $d$（区间 $[1,n-1]$ 内的一个整数），公开密钥 $Q(Q=dP)$。

### 3.2 功能调用

#### 3.2.1 加密过程

a）创建 ECIES 加密上下文，如图 2 所示。使用函数 cryptCreateContext( ) 创建加密上下文。密码库 Cryptlib 是通过创建加密上下文来完成加密、签名等一系列密码服务的。CRYPT_CONTEXT 是密码库定义的一个加密上下文对象，该对象包含加密、电子签名、哈希或 MAC 信息。

```
int cryptCreateContext
(CRYPT_CONTEXT * cryptContext,
//所创建加密上下文的地址
const CRYPT_USER cryptUser,
//拥有加密上下文的使用者，缺省设置为
CRYPT_UNSED，一般用户
const CRYPT_ALGO_TYPE cryptAlgo
//上下文中使用的算法)；
```

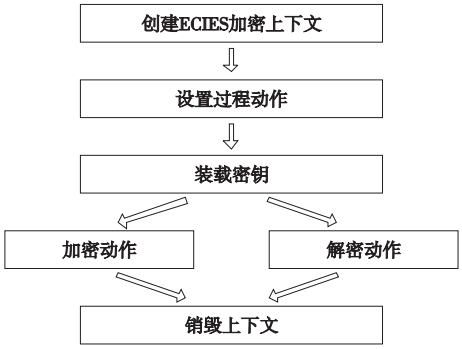


**图2 上下文加解密功能调用过程**

b）设置过程动作结构。使用宏 cryptSetAct( ) 设置 ECIES 过程动作。

```
cryptSetAct ( ECIES_PATH_ACT -> XXX, Algorithm name );
```

c）为生成的上下文装载密钥，如图 3 所示。装载密钥数据结构 CRYPT_PKCINFO_ECIES，密钥装载过程为：

（a）使用函数 cryptSetAttributeString( ) 设置 CRYPT_CTXINFO_LABEL 属性。由于是用来标志密钥的，因此属性必须被设置为唯一值。如果事先没有设置密钥标签而尝试为上下文装载一个密钥，密码库将返回 CRYPT_ERROR_NOTINITED 来指示错误。CryptSetAttributeString( ) 函数如下：

```
int cryptSetAttributeString
(const CRYPT_HANDLE cryptObject,
//添加正文或二进制字符或时间值对象
const CRYPT_ATTRIBUTE_TYPE attributeType,
//被添加的属性
const void * value,
//被添加数据的地址
const int valueLength
//被添加数据的长度)；
```

示例：cryptSetAttributeString( cryptContext, CRYPT_CTXINFO_LABEL, “ECIES key”, 9 )；

（b）在使用 CRYPT_PKCINFO_ECIES 结构之前需要使用宏 cryptInitComponents( ) 初始化，其参量为密钥数据结构指针与密钥类型。其中公钥密码密钥类型分为 CRYPT_KEYTYPE_PRIVATE 和 CRYPT_KEYTYPE_PUBLIC 两种。例如：

```
CRYPT_PKCINFO_ECIES * eciesKey;
cryptInitComponents(eciesKey,CRYPT_KEYTYPE_PUBLIC );
```

（c）初始化之后，使用宏 cryptSetComponent( ) 装载多字节整数字符，其参量包括装载目标位置、多字节整数数据、数据长度（按比特计）。用户可以选择预先设置好的椭圆曲线参量的密钥结构，这时只须设置公钥（加密时）或私钥（解密时）即可完成密钥的设置。当然，用户也可以完全按照自己的数据定义设置自己的密钥数据。

```
cryptSetComponent( eciesKey -> m,
integer data,length of integer data );
```

（d）使用 cryptSetAttributeString( ) 设置上下文 CRYPT_CTXINFO_KEY_COMPONENTS 参量，完成装载。例如：

```
cryptSetAttributeString ( cryptContext, CRYPT_CTXINFO_KEY_COMPONENTS, eciesKey, sizeof( CRYPT_PKCINFO_ECIES ) );
```

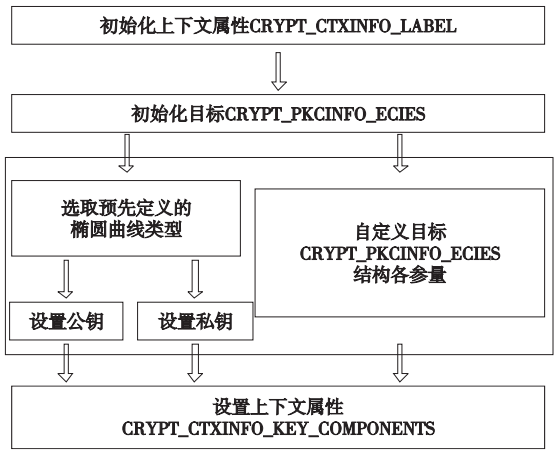


**图3 密钥装载过程**

d）进行加密动作。使用 cryptEncrypt( ) 进行加密。

```
int cryptEncrypt
(const CRYPT_CONTEXT cryptContext,
//用来加密或哈希数据的加密上下文
void * buffer,
//被加密或哈希的数据地址
const int length
//被加密或哈希的数据长度，以字节计)；
```

e）销毁上下文。使用 cryptDestroyContext( ) 销毁上下文。

```
int cryptDestroyContext
(const CRYPT_CONTEXT cryptContext
//被销毁的加密上下文)；
CRYPT_CONTEXT cryptContext;
ECIES_PATH_ACT * eciesAct;
CRYPT_PKCINFO_ECIES * eciesKey;
cryptCreateCotext(&cryptContext,cryptUser,CRYPT_ALGO_ECIES);
//创建 ECIES 加密上下文
cryptSetAct(eciesAct -> KDF,SM3);
//设置过程动作
cryptSetAct(eciesAct -> ENC,DES);
cryptSetAct(eciesAct -> MAC,CBC_MAC);
```

```
cryptSetAttributeString ( cryptContext, CRYPT _ CTXINFO _ LABEL,
"ECIES key",9);
//设置 CRYPT_CTXINFO_LABEL 为唯一值
…
cryptInitComponents( eciesKey, CRYPT_KEYTYPE_PUBLIC );
//初始化动作
eccKey - > curveType = CRYPT_ECCCURVE_P256;
//选取预先定义的椭圆曲线类型
cryptSetComponent( eciesKey - > qx,… );
cryptSetComponent( eciesKey - > qy,… );
//设置公钥从标参量
cryptSetAttributeString( cryptContext,CRYPT_CTXINFO_KEY_COM-
PONENTS,eciesKey,sizeof( CRYPT_PKCINFO_ECIES) );
//设置加密上下文属性,完成装载
…
cryptEncrypt( cryptContext,buffer,length );
//加密动作
cryptDestroyContext( cryptContext );
```

#### 3.2.2　解密过程

a)创建 ECIES 解密上下文。

使用 cryptCreateContext( )定义加密上下文(上下文相较加解密而言,没有什么差别)。

b)设置过程动作结构。

使用宏 cryptSetAct( )设置 ECIES 过程动作。

c)为该上下文装载密钥。

装载密钥数据结构 CRYPT_PKCINFO_ECIES,不同之处在于,解密过程所需设置的是私钥参量。

d)解密动作。

使用 cryptDecrypt( )进行解密。

e)销毁上下文。

使用 cryptDestroyContext( )销毁上下文。

## 4　算法测试

本文提供一组测试向量,需要注意的是下面所列数据均采用十六进制数表示。

### 4.1　密钥数据结构的参量($q$,$FR$,$S$,$a$,$b$,$P$,$n$,$h$,$Q$,$d$)选取

实际测试中,只需给出 $a$、$b$、$P$、$n$、$h$ 即可确定 $q$、$FR$、$S$。

椭圆曲线相关域参量:系数 $a=1$,$b=1$;余因子 $h=2$;$P$ 的秩 $n$ = 00000004 00000000 00000000 00020108 A2E0CC0D 99F8A5EF。

原点 $P$ 本文选取:($P_x$, $P_y$) = (00000002 FE13C053 7BBC11AC AA07D793 DE4E6D5E 5C94EEE8, 00000002 89070FB0 5D38FF58 321F2E80 0536D538 CCDAA3D9)。

私钥 $d$ = 00000003 D74898E5 D6F4A8C4 C4A7F646 71184E77 F37EADE4。

公钥 $Q=dP$,($Q_x$,$Q_y$) = (00000002 A5D2DA08 4F58DF13 C23ED8EE 807BB39A 5E33C5DC, 00000003 FA6C8CB7 A480F857 EB63FDFC 2AFD293F E36E6ED4 )。

### 4.2　输入输出结果

明文 $m$ = 00000000 00000000 01234567 89abcdef 00000000 01234567 00000000 89abcdef;

随机选取 $k$ = 00000002 00000000 00000000 00020108 A2E0CC0D 99F8A5EF;

调用 ECIES,所得加密输出($R$,$C$,$t$):

$R=kP$, ($R_x$, $R_y$) = ( 00000000 8CCFB877 BED5F772 FC1D4250 5F4CE817 65389773, 00000003 CF722FA7 BD4C3AA9 7F182A08 8DFA71A3 5B7E01B5);

$C$ = ENC($k_1$,$m$) = 4e348586 e05d4da8 e9328a45 fb23f76c 8fd829a6 4c7dc023 87692e2c 04c67f1e;

$t$ = MAC($k_2$,$C$) = 56e261d1 c6f9151c;

本文采用 DES 作为 ENC,MAC 使用 DES 的 CBC-MAC,KDF 使用国家商用密码杂凑标准 SM3[11]。应当注意的是,DES 分组规模为 64 bit,国家商用密码杂凑标准 SM3 生成 256 bit 的输出密钥。出于方便,本文明文选取 256 bit,并分成 4 块,将生成的 256 bit 密钥也分为 4 块,分别进行加密,生成密文 $C$;而进行 CBC-MAC 时,只取生成 $k_2$ 的前 64 bit 作为处理过程的加密密钥,初始向量为 0,将处理后的最后一块数据作为输出的 MAC 值,也即 $t$ 。

## 5　结束语

本文介绍了密码服务的概念,指明了密码服务的发展方向,接着引出密码服务库 Cryptlib,指出其存在的一个算法缺陷。文章重点介绍了如何将 ECC 加密体制 ECIES 嵌入到密码库 Cryptlib 之中,同时详细阐述了 ECIES 加密体制的调用是如何进行的,最后进行了算法测试,给出测试向量。在调用 ECIES 时,应用程序可应用预先定义的椭圆曲线域参量,这给应用程序带来了一定的便利性,同时应用程序也可自行定义密钥数据结构各参量,这使得应用椭圆曲线加密变得更加灵活。

椭圆曲线加密体制的成功融入,进一步完善了密码服务库 Cryptlib 的密码服务能力,增强了其功能性,提升了其密码服务的完整性,扩展了其应用广度。

**参考文献:**


[1] 郑斌. 密码片上操作系统设计与实现研究[D]. 郑州:解放军信息工程大学,2011.

[2] GUTMANN P. Cryptlib [EB/OL]. (2010-10) [2012-06]. http://www.cs.auckland.ac.nz/~pgut001/cryptlib/index.html.

[3] SECG. SEC1: elliptic curve cryptography version 1.0 [EB/OL]. (2000-09-20) [2012-06]. http://www.secg.org/collateral/sec1_final.pdf.

[4] 郁滨,蔡振国,陈韬. 参数可选的椭圆曲线加密算法的整体设计[J]. 计算机应用研究,2007,24(6):145-149.

[5] ANSI X9.63, Public key cryptography for the financial services industry, key agreement and key transport using elliptic curve cryptography[S]. USA:ANSI,2001.

[6] ISO/IEC 15946-3, Information technology-security techniques-cryptographic techniques based on elliptic curves:part 3, final draft international standard (FDIS)[S]. [S.l.]:ISO/IEC,2001.

[7] ISO/IEC 18033-2, Information technology-security techniques-encryption algorithms:part 3, asymmetric ciphers[S]. [S.l.]:ISO/IEC,2006.

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

[9] 易小琳,杨峰,鲁鹏程. 嵌入式椭圆曲线加密算法性能的研究与改进[J]. 北京工业大学学报, 2010,36(14):1722-1728.

[10] IEEE 1363a, Standard sepcifications for public-key cryptography: amendment 1,additional techniques [S]. USA:IEEE,2000.

[11] 国家密码管理局. SM3 密码杂凑算法[EB/OL]. (2010-12) [2012-06]. http://www.oscca.gov.cn/UpFile/20101222141857786.pdf.