无线局域网产品采用的 ECDSA 和 ECDH 密码算法

椭圆曲线和参数

1. 符号约定

对本文中使用到的符号约定如下：

p	192 比特的素数
$\mathbf{F}_p$	p个元素的有限域
a,b	$\mathbf{F}_p$中的元素，确定了椭圆曲线方程 $y^2 = x^3 + ax + b$
$\boldsymbol{E}$	以椭圆曲线方程 $y^2 = x^3 + ax + b$ 定义的$\mathbf{F}_p$上的椭圆曲线
O	椭圆曲线上的一个特殊点，称为无穷远点
$\boldsymbol{E}(\mathbf{F}_p)$	$\boldsymbol{E}$在$\mathbf{F}_p$上的点及无穷远点构成的集合
n	曲线点集$\boldsymbol{E}(\mathbf{F}_p)$的阶$\#\boldsymbol{E}(\mathbf{F}_p)$，要求为奇素数
$x \bmod n$	用n除x所得的余数 r ， $0 \le r \le n-1$
G	$\boldsymbol{E}(\mathbf{F}_p)$的生成元，称为基点，$G=(x_G, y_G)$
$[x, y]$	大于等于 x 且小于等于 y 的整数集合
$\lceil x \rceil$	取大于等于 x 的最小整数
t	$\mathbf{F}_p$中的元素的比特长度，$t = \lceil \log_2 p \rceil$，本文中$t = 192$
l	$\mathbf{F}_p$中的元素的字节长度，$l = \lceil t / 8 \rceil$，本文中$l = 24$
$\log_2 x$	以2为底的 x 的对数
$\|\|$	字节串的并置
d_U	用户U的私钥，$d_U \in [1, n-1]$
P_U	用户U的公钥，$P_U = d_U \cdot G$

2. 数学基础

2.1 有限域$\mathbf{F}_p$

p为素数，$\mathbf{F}_p$的元素表示为整数0, 1, 2, ..., $p-1$。

（1）$\mathbf{F}_p$中加法运算为整数的模 p加法运算：即$a, b \in \mathbf{F}_p$, $a + b = (a + b) \bmod p$。

（2）$\mathbf{F}_p$中乘法运算为整数的模 p乘法运算：即$a, b \in \mathbf{F}_p$, $a \cdot b = (a \cdot b) \bmod p$。

（3）$\mathbf{F}_p$中加法群的单位元为整数0。

（4）$\mathbf{F}_p$中乘法群的单位元为整数1。

（5）$\mathbf{F}_p$中加法群的元素a的逆元素为$p - a$。

（6）$\mathbf{F}_p$中乘法群的元素a的逆元素为b， b满足$a \cdot b = 1 \bmod p$ ，记为 a^{-1}。

2.2 椭圆曲线定义

$\mathbf{F}_p$上椭圆曲线方程为 $y^2 = x^3 + ax + b \quad (4a^3 + 27b^2 \neq 0 \bmod p)$，椭圆曲线点集 $\boldsymbol{E}(\mathbf{F}_p) = \{ (x, y) \mid x, y \in \mathbf{F}_p$，且满足 $y^2 = x^3 + ax + b \} \cup \{O\}$， 其中$O$是椭圆曲线的无穷远点，曲线$\boldsymbol{E}(\mathbf{F}_p)$的阶为$n = \#\boldsymbol{E}(\mathbf{F}_p)$。按 2.3 定义的点加运算，$\boldsymbol{E}(\mathbf{F}_p)$构成一个Abel群。

2.3 点加运算

点$P, Q \in \boldsymbol{E}(\mathbf{F}_p)$，$P = (x_1, y_1)$，$Q = (x_2, y_2)$，加法规则如下：

（1） $P + O = O + P = P$;

（2） $-P = (x_1, -y_1)$， $P + (-P) = O$;

（3） 若$Q \neq -P$, 记$P + Q = (x_3, y_3)$，

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda (x_1 - x_3) - y_1 \end{cases}$$

其中

$$\lambda = \begin{cases} (y_2 - y_1)/(x_2 - x_1) & \text{，当} x_1 \neq x_2 \\ (3x_1^2 + a)/2y_1 & \text{，当} x_1 = x_2 \end{cases}$$

2.4 多倍点运算

多倍点运算：点$P \in \boldsymbol{E}(\mathbf{F}_p)$ ，整数$k > 0$，$k \cdot P = \underbrace{P + P + \cdots + P}_{k \text{ 个}}$。

3. 数据类型转换约定

3.1 整数至字节串的转换

输入： 非负整数x和期望得到的字节串长度k，满足： $2^{8k}>x$。

输出： 长度为k的字节串M。

（1）设$M_1, M_2, \cdots, M_k$表示M中从左至右的每个字节，字节$M_i=(M_{i1}, M_{i2}, \cdots, M_{i8})$代表整数$\sum_{j=1}^{8} 2^{8-j} M_{ij}$，$1 \le i \le k$。

（2）输出字节串M满足： $x = \sum_{i=1}^{k} 2^{8(k-i)} M_i$。

3.2 字节串至整数的转换

输入： 长度为k的字节串M。

输出： 整数x。

（1）设$M_1, M_2, \cdots, M_k$表示M中从左至右的每个字节，字节$M_i=(M_{i1}, M_{i2}, \cdots, M_{i8})$代表整数$\sum_{j=1}^{8} 2^{8-j} M_{ij}$，$1 \le i \le k$。

（2）输出整数x满足： $x = \sum_{i=1}^{k} 2^{8(k-i)} M_i$。

3.3 域元素至字节串的转换

输入： 有限域$\mathbf{F}_p$中元素c。

输出： 长度为 l 的字节串S。

按照3.1节描述的方法把 c转换成为一个长度为 l 的字节串S。

3.4 字节串至域元素的转换

输入： 长度为 l 的字节串 S。

输出： 有限域$\mathbf{F}_p$中元素c。

按照3.2节描述的方法把 S转换成为一个整数c；如果c不在区间$[0, p-1]$中，则报错。

3.5 点至字节串的转换

无穷远点O用字符串方式表示为单字节$PC = 00$。

椭圆曲线上的非无穷远点$P = (x_P, y_P)$ 指定使用非压缩方式表示。

输入： 椭圆曲线上的非无穷远点$P = (x_P, y_P)$。

输出： 长度为$2l + 1$的字节串PO。

（1）按照3.3节描述的方法分别将x_P, y_P转换成长度为 l 的字节串X_1， Y_1。

（2）单字节PC赋值为04，输出字符串 $PO = PC \| X_1 \| Y_1$。

3.6 字节串至点的转换

输入： 长度为 $2l + 1$ 的字节串PO。

输出： 椭圆曲线上的非无穷远点$P = (x_P, y_P)$。

（1）设$PO = PC \| X_1 \| Y_1$，其中PC为单字节，X_1, Y_1分别是长度为 l 的字节串，若PC不等于04则报错。

（2）按照3.4节描述的方法将字节串X_1, Y_1分别转换成域元素x_P, y_P。

（3）输出点$P = (x_P, y_P)$。

4. 椭圆曲线参数

ECDSA算法和ECDH算法的密钥长度选定为 192 比特，采用域$\mathbf{F}_p$上的椭圆曲线，其参数为 $\{ p, a, b, G, n \}$，以十六进制形式表示如下：

p： BDB6F4FE3E8B1D9E0DA8C0D46F4C318CEFE4AFE3B6B8551F

a： BB8E5E8FBC115E139FE6A814FE48AAA6F0ADA1AA5DF91985

b： 1854BEBDC31B21B7AEFC80AB0ECD10D5B1B3308E6DBF11C1

x_G：4AD5F7048DE709AD51236DE65E4D4B482C836DC6E4106640

y_G：02BB3A02D4AAADACAE24817A4CA3A1B014B5270432DB27D2

n： BDB6F4FE3E8B1D9E0DA8C0D40FC962195DFAE76F56564677

5. ECDSA 算法

ECDSA 算法参引 ISO/IEC 15946：2002（E），具体参引第 2 部分第 6 章。

6. ECDH 算法

ECDH 算法参引 ISO/IEC 15946：2002（E），具体参引第 3 部分第 8 章第 4 节。

7. SHA-256 算法

SHA-256 算法参引 ISO/IEC 10118：2004（E），具体参引第 3 部分第 10 章。

8. 实例

8.1 ECDSA 算法实例

用户A的私钥d_A：

d_A： 3AC0E717EB61602EFCBB1DE81AA144A272B44BA1F16936AC

公钥$P_A=d_A \cdot G=(x_1 , y_1)$：

x_1： 7E1969FD0B001810A4E7F414C23F2BADF6B2DE96AE6B7856

y_1： 29426771EDD3001F4A4253D8EEB9FFC18684C6C0B43ACA08

十六进制字节串消息：

M： 00FFEEDDCCBBAA998877665544332211

（1）计算 M 的杂凑值 $e=h(M)$：

723AE33F076F199ECDFEFBC7169B7BE471ECB43E01ECE80ACA7539B48A4B0A90

其中，h 为杂凑算法 SHA-256。

（2）取随机整数 $k\in[1,n-1]$，假定为：

5ABC270DBCEE31A4B00132331DDD596173EAF656ABCC39CB

（3）将杂凑值e用用户A的私钥d_A签名后得到签名结果：

A9F40F155FCF18E8D35AB47EE65CD2F906465155A71DFA38

7EAFA7E5A2335CD337E37B39601D2D5022E1799799F0E262

8.2 ECDH 算法实例

用户A的临时私钥记为d_A，临时公钥记为$P_A=d_A \cdot G=(x_2 , y_2)$：

d_A： 3AC0E717EB61602EFCBB1DE81AA144A272B44BA1F16936AC

x_2： 7E1969FD0B001810A4E7F414C23F2BADF6B2DE96AE6B7856

y_2： 29426771EDD3001F4A4253D8EEB9FFC18684C6C0B43ACA08

用户B的临时私钥记为d_B，临时公钥记为$P_B=d_B \cdot G=(x_3, y_3)$：

d_B： 25FBB32EFBEC6ECB1314332A026582DB7BE00C051CF2FA80

x_3： 0621D8ADAB0952752EBEAE5007F6AE455C61860D1CEADB25

y_3： 6A58D5D55087325DAC434C0DD28A9F8159070C8AAECD21D8

按照ECDH算法，用户A、B分别计算出共享信息$K_{AB}= d_A \cdot P_B = K_{BA}= d_B \cdot P_A=(x_4, y_4)$：

x_4： 3A74DDFA3080F6B5A1688C6EB7B098240B5AFC672450A425

y_4： 7FF89712A653D6E1B30CD24AC6C72BD3A90F2F9EACE3F3F6