ECDSA 签名算法

Sammy

2018-04-10

1 公开参数

ECDSA 基于素数域的椭圆曲线，相关参数如表 1所示

表 1: ECDSA 涉及的公开参数

符号	含义
O	无穷远点
q	一个大素数
b	椭圆曲线方程 $y^2 \equiv x^3 - 3x + b$ 中的常数
G	基点 (x_g, y_g)，选定的曲线上的一个点
n	G 的阶，满足 $nG = O$
H	哈希算法

2 具体算法

组成算法的 3 个关键流程分别如算法 1、2、3所示

Algorithm 1: ECDSA 密钥生成

输出: 私钥 d，公钥 Q

$d \in_R [1, n-1]$ /* 随机生成一个整数 d */

$Q \leftarrow d \cdot G$

Algorithm 2: ECDSA 签名

输入: 私钥 d，消息 m

输出: 签名 (r, s)

$r \leftarrow 0$

while $r = 0$ **do**

  $k \in_R [1, n-1]$

  $P = (x, y) = k \cdot G$

  $r \equiv x \pmod n$

$e \leftarrow H(m)$

$s \leftarrow 0$

while $s = 0$ **do**

  $s \leftarrow k^{-1}(e + d \cdot r) \pmod n$

Algorithm 3: ECDSA 验签

输入: 公钥 Q，消息 m，签名 (r, s)

输出: Y 表示签名合法，N 表示签名非法

if $r \notin [1, n-1]$ 或 $s \notin [1, n-1]$ **then**

  return N

$e \leftarrow H(m)$

$w \leftarrow s^{-1} \pmod n$

$A \leftarrow (x_1, y_1) = ewG + rwQ$

if $A = O$ **then**

  return N;

$v \leftarrow x_1 \pmod n$

if $v = r$ **then return** Y

else return N

3 正确性证明

$$
\begin{aligned}
& s \equiv k^{-1}(e + d \cdot r) \pmod{n} \\
\Rightarrow & k \equiv s^{-1}(e + d \cdot r) \pmod{n} \\
\Rightarrow & k \equiv (s^{-1} \cdot e + s^{-1} \cdot d \cdot r) \pmod{n} \\
\Rightarrow & k \equiv (e \cdot w + r \cdot w \cdot d) \pmod{n} \\
\Rightarrow & kG \equiv (e \cdot w + r \cdot w \cdot d) \cdot G \pmod{n} \\
\Rightarrow & kG \equiv e \cdot w \cdot G + r \cdot w \cdot d \cdot G \pmod{n} \\
\Rightarrow & kG \equiv e \cdot w \cdot G + r \cdot w \cdot Q \pmod{n} \\
\Rightarrow & P = A \\
\Rightarrow & r \equiv x \equiv x_1 \equiv v \pmod{n}
\end{aligned}
$$

4 Go 语言的 ECDSA 库

Go 语言 crypto/elliptic 包提供了 FIPS 186-3 标准规定的 4 种曲线 P224、P256、P384 和 P512，我们根据自己的安全强度需求直接利用这些曲线就行。其中，紧接 P 的数字表示以比特为单位衡量的安全强度，越大越强。

相应的演示程序如`ecdsa_test.go`的`TestECDSA`函数。