ECC 应用说明

注：

此文<章节 1,3,5,6>相关原理及算法描述，摘录于网络及书籍《椭圆曲线密码学导论》等；如有错误，欢迎指正；

本文为钜泉 ECC 模块使用辅助说明文档，仅供参考。

版本记录	版本修改说明
V0.1	1. 初始版本 @20170207
V0.2	1. 修改页眉文档名，去掉指定芯片 ID HT502X,后续的 HT603X 系列也增加了 ECC 模块；

1 椭圆曲线点运算原理

1.1 椭圆曲线的定义

一条椭圆曲线是在**射影平面上**满足方程

$$Y^2Z+a_1XYZ+a_3YZ^2=X^3+a_2X^2Z+a_4XZ^2+a_6Z^3 \quad [1]$$

的所有点的集合，且曲线上的每个点都是非奇异（或光滑）的。

直角坐标系中，x=X/Z ， y=Y/Z 代入方程[1]得到：

$$y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6 \quad [2]$$

用于加密的常用曲线

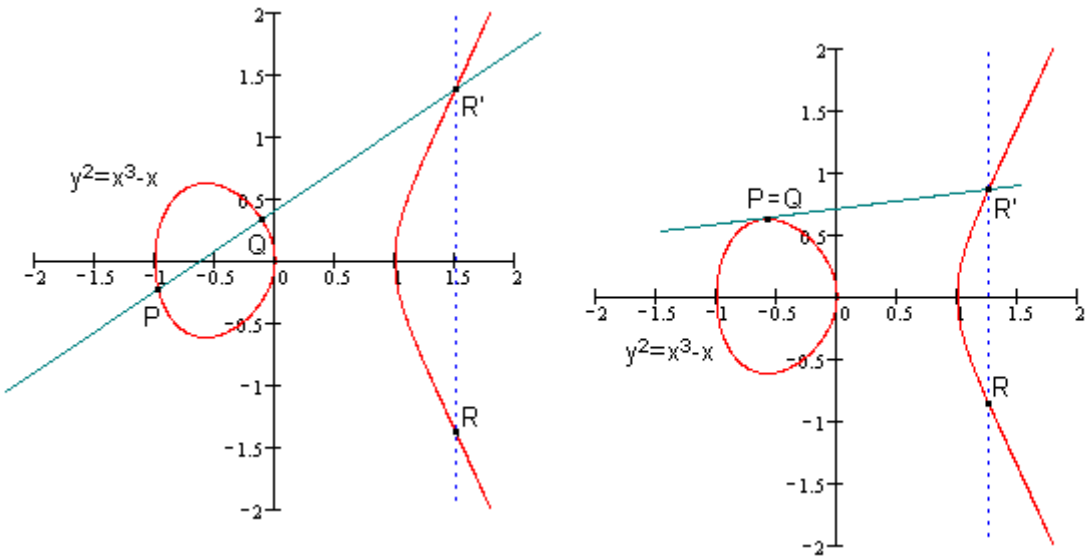
$$y^2 = x^3+ ax + b \quad [3]$$

1.2 椭圆曲线运算

1.2.1 点加（double 加）运算

运算法则：任意取椭圆曲线上两点 P、Q （若 P、Q 两点重合，则做 P 点的切线）做直线交于椭圆曲线的另一点 R’，过 R’做 y 轴的平行线交于 R。

我们规定 P + Q = R （如图）

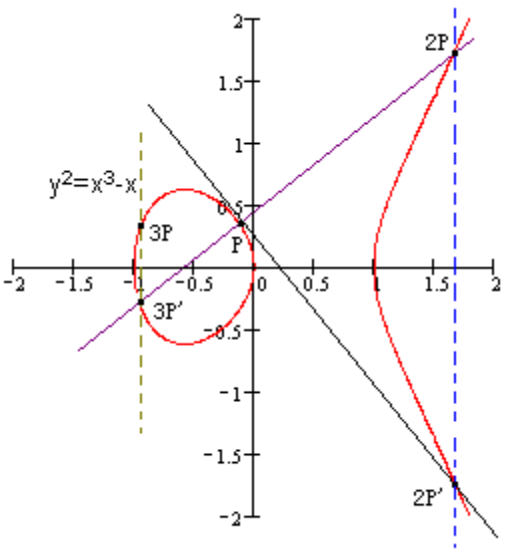


注：这里的 + 不是实数中普通的加法，而是从普通加法中抽象出来的加法，他具备普通加法的一些

性质，但具体的运算法则显然与普通加法不同

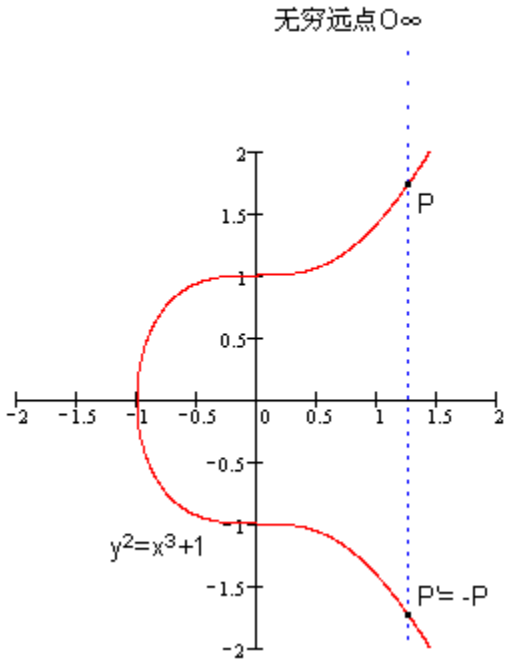
1.2.2 标量乘法运算

运算法则：k 个相同的点 P 相加，我们记作 kP。如下图：P+P+P = 2P+P = 3P



1.2.3 负元定义

椭圆曲线无穷远点 O∞与椭圆曲线上一点 P 的连线交于 P'，过 P'作 y 轴的平行线交于 P，所以有 无穷远点 O∞+ P = P 。这样，无穷远点 O∞的作用与普通加法中零的作用相当（0+2=2），我们把无穷远点 O∞ 称为 零元。同时我们把 P'称为 P 的负元（简称，负 P；记作，-P）。（参见下图）



注：P(x,y)的负元不是简单的 P'(x,-y),只有 $y^2 = x^3 + ax + b$ 下成立。

1.3 密码学中的椭圆曲线

$$y^2 = x^3 + ax + b$$

有限域约束，离散化

$$y^2 = x^3 + ax + b \pmod{p}$$，p 为大质数，x,y 属于 0 到 p-1 间的整数

并将这条椭圆曲线记为 **Ep(a,b)**，包含满足方程的所有点(x,y)，再加上 无穷远点 O∞

有限域 F_p 上的椭圆曲线运算如下：

1）无穷远点 O∞是零元，有 O∞+ O∞= O∞，O∞+P=P

2）P(x,y)的负元是 (x,-y)，有 P+(-P)= O∞

3）$P(x_1,y_1)$,$Q(x_2,y_2)$的和 $R(x_3,y_3)$ 有如下关系：

$x_3 \equiv k^2 - x_1 - x_2 \pmod p$

$y_3 \equiv k(x_1 - x_3) - y_1 \pmod p$

其中若 P=Q 则 $k=(3x^2+a)/2y_1$ 若 P≠Q，则 $k=(y_2-y_1)/(x_2-x_1)$

1.4 椭圆曲线上简单的加密/解密

公开密钥算法：选定一条椭圆曲线 Ep(a,b)，并取椭圆曲线上一点，作为基点 G

举例 1(ECDH)Diffie-Hellman：

用户 A: 私钥 k_A → 公钥 $K_A = k_A * G$

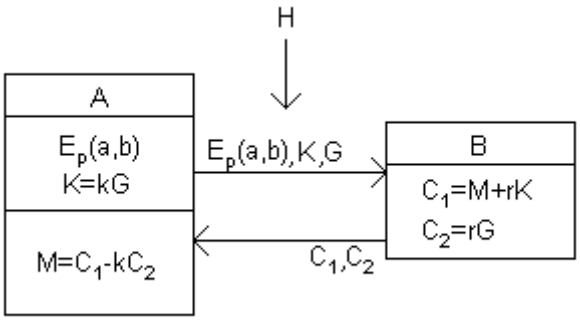
用户 B: 私钥 k_B → 公钥 $K_B = k_B * G$

1）用户 A 传递 Ep(a,b)，G，K_A 给用户 B
2）用户 B 将明文 M 使用 K_A*k_B 进行加密运算得到密文 C,并将 C 和 K_B 传递给用户 A
3）用户 A 使用 K_B*k_A 对 C 解密，得到明文 M

举例 2：

1）用户 A 选定一条椭圆曲线 Ep(a,b)，并取椭圆曲线上一点，作为基点 G。
2）用户 A 选择一个私有密钥 k，并生成公开密钥 K=kG。
3）用户 A 将 Ep(a,b)和点 K，G 传给用户 B。
4）用户 B 接到信息后 ，将待传输的明文编码到 Ep(a,b)上一点 M（**编码方法很多，这里不作讨论**），并产生一个随机整数 r（r<n）。
5）用户 B 计算点 **$C_1=M+rK$；$C_2=rG$。**
6）用户 B 将 C_1、C_2 传给用户 A。
7）用户 A 接到信息后，计算 C_1-kC_2，结果就是点 M。因为
$C_1-kC_2=M+rK-k(rG)=M+rK-r(kG)=M$
再对点 M 进行解码就可以得到明文。

在这个加密通信中，如果有一个偷窥者 H ，他只能看到 Ep(a,b)、K、G、C_1、C_2 而通过 K、G 求 k 或通过 C_2、G 求 r 都是相对困难的。因此，H 无法得到 A、B 间传送的明文信息。



2 ECC 加速模块

2.1 概述

ECC256 模块为椭圆曲线(EC，Elliptic Curve)运算加速模块，显著提高基于椭圆曲线的加密协议的实现效率。本模块支持的加密协议或算法包括 ECDSA(EC Digital Signature Algorithm) ，ECDH（EC Deffie-Hellman）及相关变种协议。主要特征如下

- 支持基于简化 Weierstrass 方程（$y^2=x^3+ax+b \bmod p$）的所有素域 GF(p)上椭圆曲线，此椭圆曲线在 **NIST, SEC2, Brainpool** 等协议中定义为常用椭圆曲线
- 支持 ECDSA 签名/认证运算
- 支持公钥是否在椭圆曲线上验证运算
- 支持私钥数据源可选：寄存器或 flash 自加载
- 支持椭圆曲线点运算
  点乘运算（ECSM，EC Scalar Multiplication）
  点加运算（ECA，EC Addition）
  倍加运算（ECD，EC Doubling）
- 支持大数模运算
  模加运算（MA，Modular Addition）
  模减运算（MS，Modular Subtraction）
  模乘运算（MM，Modular Multiplication）
  模除运算（MD，Modular Division）
  模逆运算（MI，Modular Inversion）
- 支持 ECC256，且向下兼容 ECC244 和 ECC192
- 可阻挡 STA 攻击(both Simple and Statistical Timing side channel Attacks)和 SPA 攻击(Simple Power Analysis attacks)

2.2 ECC 各运算寄存器配置

加速模块输入/输出			
	运算模式	输入参数寄存器	输出结果寄存器
ECDSA 运算	**签名运算**	PREG: 大素数 p AREG: 椭圆曲线参数 a PXREG: 椭圆曲线基点 x 坐标 PYREG: 椭圆曲线基点 y 坐标 SXREG: 基点的阶参数 KEYREG: 私钥 k SYREG: 随机大整数	RXREG: 签名结果 r 参数 RYREG: 签名结果 s 参数 ECCSTA: ECC 状态寄存器 [3:3]ECDSA_S = 0(若为 1，重新获取随机大整数，重新运算) [0:0]ECCFLG = 1

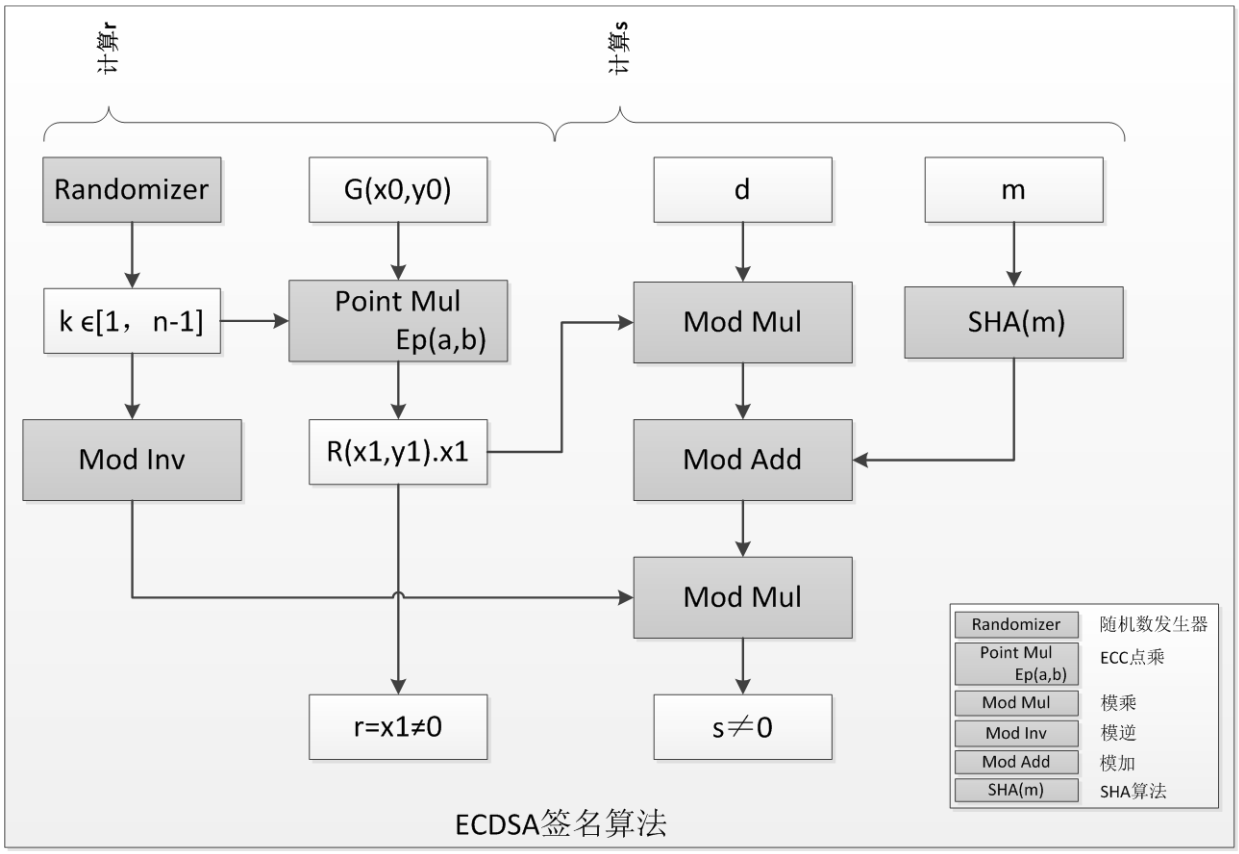
		MREG: HASH 后的消息摘要	
	认证运算	PREG: 大素数 p AREG: 椭圆曲线参数 a PXREG: 椭圆曲线基点 x 坐标 PYREG: 椭圆曲线基点 y 坐标 SXREG: 基点的阶参数 KEYREG：公钥 x 坐标 SYREG: 公钥 y 坐标 MREG: HASH 后的消息摘要 RXREG: 签名结果 r 参数 RYREG: 签名结果 s 参数	ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1 [2:2]ECDSA_V = 1(认证通过) = 0(认证失败)
点在曲线上验证	**公钥验证**	PREG: 大素数 p AREG: 椭圆曲线参数 a SYREG: 椭圆曲线参数 b PXREG: 公钥 x 坐标 PYREG: 公钥 y 坐标	ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1 [4:4]PKV = 1(验证通过) = 0(验证失败)
ECC 点运算	**点加运算**	**P(X1,Y1) + S(X2,Y2) = R(X3,Y3)**	
		PREG: 大素数 p AREG: 椭圆曲线参数 a PXREG: 椭圆曲线第 1 点 x 坐标 PYREG: 椭圆曲线第 1 点 y 坐标 SXREG: 椭圆曲线第 2 点 x 坐标 SYREG: 椭圆曲线第 2 点 y 坐标	RXREG: 点加结果 x 坐标 RYREG: 点加结果 y 坐标（结果点在是椭圆曲线上） ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1
	倍加运算	**2 *P(X1,Y1) = R(X3,Y3)**	
		PREG: 大素数 p AREG: 椭圆曲线参数 a PXREG: 椭圆曲线第 1 点 x 坐标 PYREG: 椭圆曲线第 1 点 y 坐标	RXREG: 点加结果 x 坐标 RYREG: 点倍加结果 y 坐标（结果点在是椭圆曲线上） ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1
	点乘运算	**k*P(X1,Y1) = R(X3,Y3)**	
		PREG: 大素数 p AREG: 椭圆曲线参数 a PXREG: 椭圆曲线点 x 坐标 PYREG: 椭圆曲线点 y 坐标 KEYREG：标量系数 k	RXREG: 点加结果 x 坐标 RYREG: 点乘结果 y 坐标（结果点在是椭圆曲线上） ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1
标量模运算	**模加运算**	**PX + PY (mod n) = RX**	
		PREG: 模运算参数 n PXREG: 第 1 个标量 PYREG: 第 2 个标量	RXREG: 模加结果标量 ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1
	模减运算	**PX - PY (mod n) = RX**	
		PREG: 模运算参数 n PXREG: 第 1 个标量 PYREG: 第 2 个标量	RXREG: 模减结果标量 ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1

	模乘运算	**PX * PY (mod n) = RX**	
		PREG: 模运算参数 n PXREG: 第 1 个标量 PYREG: 第 2 个标量	RXREG: 模乘结果标量 ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1
	模除运算	**PX / PY (mod n) = RX**	
		PREG: 模运算参数 n PXREG: 除数 PYREG: 被除数	RXREG: 模除结果标量 ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1
	模逆运算	**PX^{-1} (mod n) = RX**	
		PREG: 模运算参数 n PXREG: 除数	RXREG: 模逆结果标量 ECCSTA: ECC 状态寄存器 [0:0]ECCFLG = 1

3 ECDSA 算法

3.1 ECDSA 签名算法

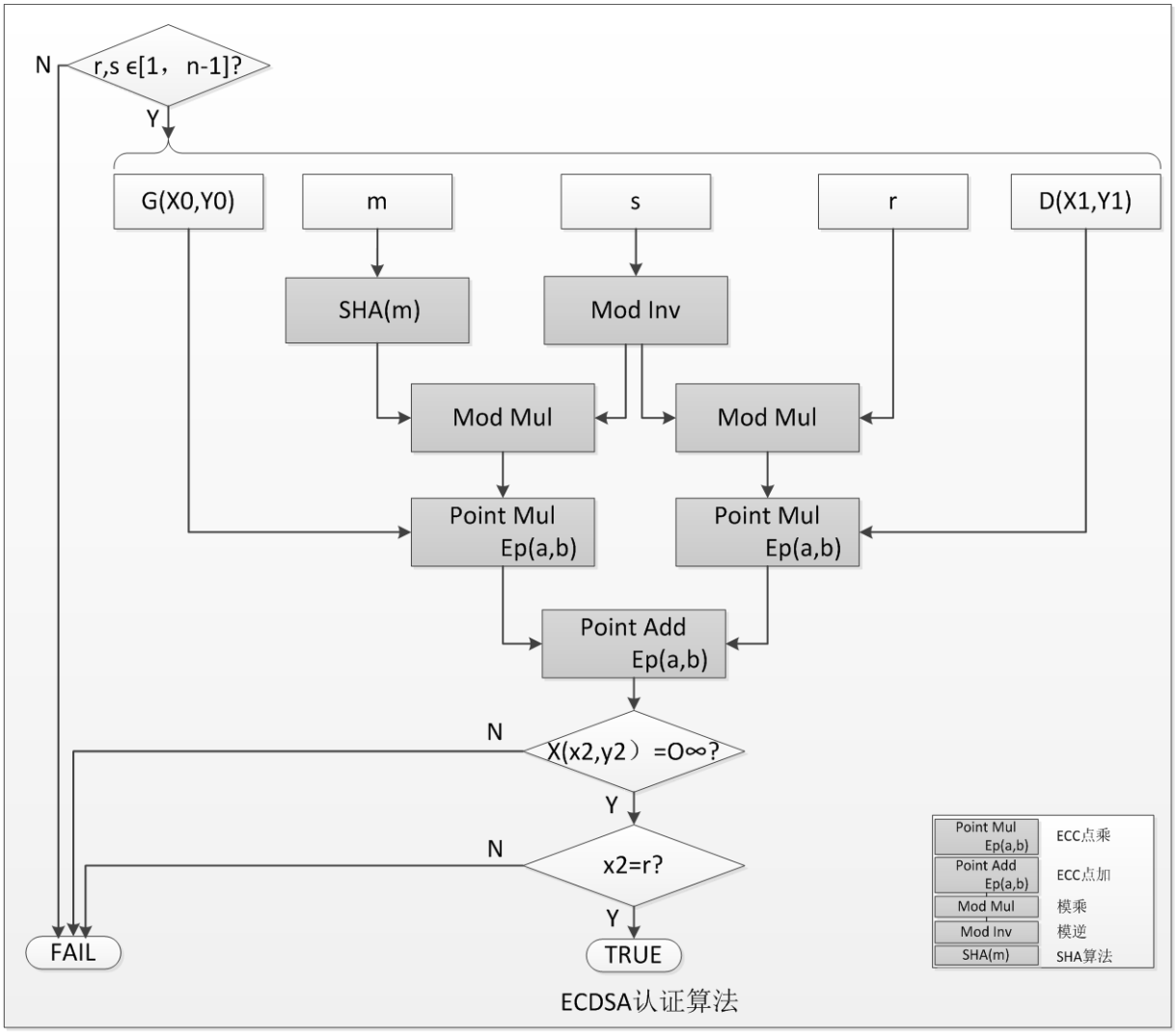
输入: 素域 Fp 上椭圆曲线 Ep(a,b)，素域的阶 p，基点 G(x,y)，G 的阶 n，私钥 d，消息 m
输出: 签名(r, s)
1）在实数域[1, n-1]内选择一个随机大数 k。
2）计算 kG = (x1, y1)并将 x1 转换为整数 X
3）计算 r = X mod n. 若 r = 0,则跳至步骤 1
4）计算 e = SHA(m)
5）计算 k 的逆元 k1，计算 s = k1*(e + d*r) mod n. 若 r = 0,则跳至步骤 1.
6）返回(r, s)



注：图中灰色底纹模块为 IP 可加速的功能模块，其余软件实现

3.2 ECDSA 认证算法

输入: 素域 Fp 上椭圆曲线 Ep(a,b)，素域的阶 p，基点 G(x,y)，G 的阶 n，公钥 D，消息 m，签名(r, s)
输出: 判断签名(r, s)是否合法
1）检验签名是否是区间[1, n-1]内的整数.若任何一个检验失败,则返回("拒绝该签名")
2）计算 e = SHA(m)
3）计算 s 的逆元 s1,并计算 w = s1 mod n.
4）计算 u1 = (e*w) mod n 和 u2 = (r*w) mod n.
5）计算 X = u1*G +u2*D.
6）若 X = 无穷大，则返回("拒绝该签名")
7）将 X 的 x1 坐标转换为整数 t，计算 w = t mod n.
8）若 w= r，则返回("接受该签名")；否则,返回("拒绝该签名").



注：图中灰色底纹模块为 IP 可加速的功能模块，其余软件实现

4 算法实现

4.1 ECDSA 签名库函数

函数声明：

Bool ECDSA_Signature(ECC_ECDSA_InitTypedef* ECC_ECDSA_InitStruct,
ECC_ECDSA_Signature_InputTypedef* ECDSA_S_InputStruct,
ECC_PointOperate_OutputTypedef* ECDSA_S_OutputStruct)；

入口参数:

ECC_ECDSA_InitStruct ECC 曲线参数初始化变量结构体指针，主要包含 5 个参数:

1) pECCpAddr uint32_t *型变量，其指向对象存储素域的阶 p
2) pECCaAddr uint32_t *型变量，其指向对象存储素域上的曲线参数 A
3) pECCnAddr uint32_t *型变量，其指向对象存储基点的阶 n
4) pECCGxAddr uint32_t *型变量，其指向对象存储基点 x 坐标
5) pECCGyAddr uint32_t *型变量，其指向对象存储基点 y 坐标

ECDSA_S_InputStruct ECDSA-S 运算输入变量结构体指针，主要包含 3 个参数:

6) pdAddr uint32_t *型变量，其指向对象存储私钥 d
7) pkAddr uint32_t *型变量，其指向对象存储随机大整数
8) pMAddr uint32_t *型变量，其指向对象存储 HASH 后的消息摘要 m

ECDSA_S_OutputStruct ECC 运算结果数据结构体指针，主要包含 10 个参数:

1) pRxAddr uint32_t *型变量，ECC 点运算结果 x 坐标/ECDSA 签名结果 r
2) pRyAddr uint32_t *型变量，ECC 点运算结果 y 坐标/ECDSA 签名结果 s

返回参数:

Bool = TRUE ，签名完成
= FALSE ，签名 fail，满足重试条件

4.2 ECDSA 认证库函数

函数声明：

完全硬件实现（ECDSA-V 运算）

Bool ECDSA_Verification(ECC_ECDSA_InitTypedef* ECC_ECDSA_InitStruct,
ECC_ECDSA_Verification_InputTypedef* ECDSA_V_InputStruct)

半软件实现（ECC 点运算）

Bool ECDSA_Verification_HalfSoft(ECC_ECDSA_InitTypedef* ECC_ECDSA_InitStruct,
ECC_ECDSA_Verification_InputTypedef* ECDSA_V_InputStruct)

入口参数:

ECC_ECDSA_InitStruct ECC 曲线参数初始化变量结构体指针，主要包含 5 个参数:

1) pECCpAddr uint32_t *型变量，其指向对象存储素域的阶 p

	2) pECCaAddr	uint32_t *型变量，其指向对象存储素域上的曲线参数 A
	3) pECCnAddr	uint32_t *型变量，其指向对象存储基点的阶 n
	4) pECCGxAddr	uint32_t *型变量，其指向对象存储基点 x 坐标
	5) pECCGyAddr	uint32_t *型变量，其指向对象存储基点 y 坐标
ECC_ECDSA_V_InputStruct	ECDSA-V 运算输入变量结构体指针，主要包含 10 个参数:	
	6) pECCDxAddr	uint32_t *型变量，其指向对象存储公钥 x 坐标
	7) pECCDyAddr	uint32_t *型变量，其指向对象存储公钥 y 坐标
	8) pMAddr	uint32_t *型变量，其指向对象存储 HASH 后的消息摘要 m
	9) prAddr;	uint32_t *型变量，其指向对象存储签名结果 r 参数
	10)psAddr;	uint32_t *型变量，其指向对象存储签名结果 s 参数

返回参数:

Bool	= TRUE ， 认证通过
	= FALSE ， 认证 fail

5 TEST CASE

ECC 点乘运算 R(x, y) = k * G(x, y)
载入参数值：(ECC256) G(x,y)= 6B17D1F2E12C4247F8BCE6E563A440F277037D812DEB33A0F4A13945D898C296 4FE342E2FE1A7F9B8EE7EB4A7C0F9E162BCE33576B315ECECBB6406837BF51F5 K = 101D35748FA6C773DEF6BBBC4E31D720D7CB07C369060856A4EE9F7A9905C315 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF A = -3 mod P FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFC 运算结果： R(x,y) = CA9FDDF2F526D6A33D1CE42A5B1DDB74F1809CADAB9AC14C84B1AF00A3B7CE8E C5F7CD905762254917B1C6D5E6535613C2BB7079F54E7F2D9B4A8CFB91A2C035
载入参数值：(ECC224) G(x,y)= 00000000B70E0CBD6BB4BF7F321390B94A03C1D356C21122343280D6115C1D21 00000000BD376388B5F723FB4C22DFE6CD4375A05A07476444D5819985007E34 K = 000000002455354676AC8FDB21F9A5709085FFBC761E762351E2648E0D6D7F76 P = 00000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000000000000000000000001 A = -3 mod P 00000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFE 运算结果： R(x,y) = 554304D8 B3E4322E 71BA314D 2BCCFE75 E6DEA336 7F774B9D D6B060F4 E7CAFA99 E371A251 B9570D9E 1F60D8DF 6B543F65 ADBBC60E 407C1FBC

载入参数值：(ECC192) G(x,y)= 000000000000000188DA80EB03090F67CBF20EB43A18800F4FF0AFD82FF1012 0000000000000000007192B95FFC8DA78631011ED6B24CDD573F977A11E794811 K = 000000000000000A54F21D1FFAD0B3F792AB40CBAF122BF66D077853F483A12 P = 000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFFFFFFFFFF A = -3 mod P 000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFFFFFFFFFC 运算结果： R(x,y) = 29953CFCE855B475372CE54CEC9E377CBC1BB7FDB9B3E571 59C8E9F9D8532D3D202527AF9E36E5F660EA1119965F9E3C
ECC **点加运算** R(x，y) = G(x，y) + S(x，y)
载入参数值： G(x,y)= 6B17D1F2E12C4247F8BCE6E563A440F277037D812DEB33A0F4A13945D898C296 4FE342E2FE1A7F9B8EE7EB4A7C0F9E162BCE33576B315ECECBB6406837BF51F5 S(x，y)= CA9FDDF2F526D6A33D1CE42A5B1DDB74F1809CADAB9AC14C84B1AF00A3B7CE8E C5F7CD905762254917B1C6D5E6535613C2BB7079F54E7F2D9B4A8CFB91A2C035 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF A = -3 mod P FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFC 运算结果： R(x,y) = B68E326959634F75FAFC411C94F1A1BF80D9836B7FD90A4870F60AD90DF6FDCF 2C4A826471F3AF9E44778975E333621030706D1F5DC4EB5AB69F44CE63D93CC7
ECC **倍加运算** R(x，y) = 2 * G(x，y)

载入参数值: G(x,y)= 6B17D1F2E12C4247F8BCE6E563A440F277037D812DEB33A0F4A13945D898C296 4FE342E2FE1A7F9B8EE7EB4A7C0F9E162BCE33576B315ECECBB6406837BF51F5 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF A = -3 mod P FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFC 运算结果: R(x,y) = 7CF27B188D034F7E8A52380304B51AC3C08969E277F21B35A60B48FC47669978 07775510DB8ED040293D9AC69F7430DBBA7DADE63CE982299E04B79D227873D1
大数**模加运算** Rx = Gx + Gy mod p
载入参数值: Gx = C026EB6EE5881FD29C9F4F7FA66E229516A8E7FEFE2C04EDF1E50D3E7151EBB1 Gy = E1979DE0568339B050F8928516420061F309BEA4D9AFE61E615598616C7651B8 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF 运算结果: Rx = A1BE89503C0B5981ED97E204BCB022F709B2A6A2D7DBEB0C533AA59FDDC83D6A ECCSTA:0x1
大数**模减运算** Rx = Gx - Gy mod p
载入参数值: Gx = C026EB6EE5881FD29C9F4F7FA66E229516A8E7FEFE2C04EDF1E50D3E7151EBB1 Gy = E1979DE0568339B050F8928516420061F309BEA4D9AFE61E615598616C7651B8 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF 运算结果: Rx = DE8F4D8D8F04E6234BA6BCFA902C2233239F295B247C1ECF908F74DD04DB99F8
大数**模乘运算** Rx = Gx * Gy mod p

载入参数值: Gx = C026EB6EE5881FD29C9F4F7FA66E229516A8E7FEFE2C04EDF1E50D3E7151EBB1 Gy = E1979DE0568339B050F8928516420061F309BEA4D9AFE61E615598616C7651B8 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF 运算结果: Rx = 3BB1DDF825E1204EB79DB4D83F1C4EC3C38CF4CB13C9B55A5B927DBE58F1FF2F
大数**模除运算** Rx = Gy / Gx mod p
载入参数值: Gx = C026EB6EE5881FD29C9F4F7FA66E229516A8E7FEFE2C04EDF1E50D3E7151EBB1 Gy = E1979DE0568339B050F8928516420061F309BEA4D9AFE61E615598616C7651B8 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF 运算结果: Rx = 7AC0AB1FA466EC85170EA821C6D6B4B7AB7B4003940287DCC2031A9655690E2A
大数**模逆运算** Rx = 1 / Gx mod p
载入参数值: Gx = C026EB6EE5881FD29C9F4F7FA66E229516A8E7FEFE2C04EDF1E50D3E7151EBB1 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF 运算结果: Rx = 912D305B9FA771B7963E8799822F4016909B5D8C22E93E3E70C0F31C7EB372E8
ECC256 **ECDSA 签名运算**

载入参数值: G(x,y)= 6B17D1F2E12C4247F8BCE6E563A440F277037D812DEB33A0F4A13945D898C296 4FE342E2FE1A7F9B8EE7EB4A7C0F9E162BCE33576B315ECECBB6406837BF51F5 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF A = -3 mod P FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFC k（私钥）= 101D35748FA6C773DEF6BBBC4E31D720D7CB07C369060856A4EE9F7A9905C315 n（基阶）= FFFFFFFF00000000FFFFFFFFFFFFFFFFBCE6FAADA7179E84F3B9CAC2FC632551 Rand = 732B578FCD30BA636965A177596065A1835C026B3BBCE990F360AD51AB6EC9CA Hash = 76823489310398B830799E95A8A1819B0FC44253C1ED704CB6030C8248BA22FE 运算结果: (r, s) = 124C6384C629842A92ED98844EA505FF0BF33C456AA4ABA89C3D84EC3ABDF10E FDAB2982C6FBE1A124C854F8C94FD52AE4B694D04554B84F74FF7D4FA543B25D
ECC256 ECDSA 认证运算

载入参数值: G(x,y)= 6B17D1F2E12C4247F8BCE6E563A440F277037D812DEB33A0F4A13945D898C296 4FE342E2FE1A7F9B8EE7EB4A7C0F9E162BCE33576B315ECECBB6406837BF51F5 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF A = -3 mod P FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFC D(x,y)（公钥）= CA9FDDF2F526D6A33D1CE42A5B1DDB74F1809CADAB9AC14C84B1AF00A3B7CE8E C5F7CD905762254917B1C6D5E6535613C2BB7079F54E7F2D9B4A8CFB91A2C035 n（基阶）= FFFFFFFF00000000FFFFFFFFFFFFFFFFBCE6FAADA7179E84F3B9CAC2FC632551 Hash = 76823489310398B830799E95A8A1819B0FC44253C1ED704CB6030C8248BA22FE (r, s) = 124C6384C629842A92ED98844EA505FF0BF33C456AA4ABA89C3D84EC3ABDF10E FDAB2982C6FBE1A124C854F8C94FD52AE4B694D04554B84F74FF7D4FA543B25D 运算结果: ECDSA verification: pass
ECC256（串口波特率设置 4800）**点在 ECC 曲线上验证**
载入参数值: D(x,y)= 6B17D1F2E12C4247F8BCE6E563A440F277037D812DEB33A0F4A13945D898C296 4FE342E2FE1A7F9B8EE7EB4A7C0F9E162BCE33576B315ECECBB6406837BF51F5 P = FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF A = -3 mod P FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFC B = 5AC635D8AA3A93E7B3EBBD55769886BC651D06B0CC53B0F63BCE3C3E27D2604B 运算结果: Public Key Validation: pass

6 附表：NIST 推荐的素域 Fp 上的随机椭圆曲线

P-192: $p = 2^{192} - 2^{64} - 1$, $a = -3$, $h = 1$
S = 0x 3045AE6F C8422F64 ED579528 D38120EA E12196D5
r = 0x 3099D2BB BFCB2538 542DCD5F B078B6EF 5F3D6FE2 C745DE65
b = 0x 64210519 E59C80E7 0FA7E9AB 72243049 FEB8DEEC C146B9B1
n = 0x FFFFFFFF FFFFFFFF FFFFFFFF 99DEF836 146BC9B1 B4D22831
x = 0x 188DA80E B03090F6 7CBF20EB 43A18800 F4FF0AFD 82FF1012
y = 0x 07192B95 FFC8DA78 631011ED 6B24CDD5 73F977A1 1E794811
P-224: $p = 2^{224} - 2^{96} + 1$, $a = -3$, $h = 1$
S = 0x BD713447 99D5C7FC DC45B59F A3B9AB8F 6A948BC5
r = 0x 5B056C7E 11DD68F4 0469EE7F 3C7A7D74 F7D12111 6506D031 218291FB
b = 0x B4050A85 0C04B3AB F5413256 5044B0B7 D7BFD8BA 270B3943 2355FFB4
n = 0x FFFFFFFF FFFFFFFF FFFFFFFF FFFF16A2 E0B8F03E 13DD2945 5C5C2A3D
x = 0x B70E0CBD 6BB4BF7F 321390B9 4A03C1D3 56C21122 343280D6 115C1D21
y = 0x BD376388 B5F723FB 4C22DFE6 CD4375A0 5A074764 44D58199 85007E34
P-256: $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$, $a = -3$, $h = 1$
S = 0x C49D3608 86E70493 6A6678E1 139D26B7 819F7E90
r = 0x 7EFBA166 2985BE94 03CB055C 75D4F7E0 CE8D84A9 C5114ABC AF317768 0104FA0D
b = 0x 5AC635D8 AA3A93E7 B3EBBD55 769886BC 651D06B0 CC53B0F6 3BCE3C3E 27D2604B
n = 0x FFFFFFFF 00000000 FFFFFFFF FFFFFFFF BCE6FAAD A7179E84 F3B9CAC2 FC632551
x = 0x 6B17D1F2 E12C4247 F8BCE6E5 63A440F2 77037D81 2DEB33A0 F4A13945 D898C296
y = 0x 4FE342E2 FE1A7F9B 8EE7EB4A 7C0F9E16 2BCE3357 6B315ECE CBB64068 37BF51F5