CryptoCurrency & BlockChain

密碼貨幣與區塊鏈 (2)

金融科技導論

陳君明

jmchen@crypto.tw

Agenda

- Abstract Algebra: Groups
- Elliptic Curves
- Keys, Addresses
- ECDSA
- Side-Channel Attacks

Abstract Algebra: Groups

Floor and Ceiling

- **Definition**
  1) The **floor** $\lfloor x \rfloor$ of $x \in \boldsymbol{R}$ is the largest integer $\leq x$
  2) The **ceiling** $\lceil x \rceil$ of $x \in \boldsymbol{R}$ is the smallest integer $\geq x$
- **Example**
  - $\lfloor e \rfloor = 2$, $\lceil e \rceil = 3$, $\lfloor -3.1416 \rfloor = -4$, $\lceil -3.1416 \rceil = -3$
- **Example**
  1) $25 = 3 \times 7 + 4$ [7: divisor, 3: quotient, 4: remainder]
  2) $25 \bmod 7 = 25 - \lfloor 25/7 \rfloor \times 7 = 25 - 3 \times 7 = 25 - 21$

Modular Function

- **Definition**

  $m, n \in \boldsymbol{Z}, \; m > 0$, define

  $$\boldsymbol{n} \;\textbf{mod}\; \boldsymbol{m} = n - \lfloor n / m \rfloor \times m$$

  - i.e., the remainder after dividing n by m, which is ≥ 0 and $< m$

- **Example** 58 in the base 3 representation

$$a_0 = 58 \bmod 3 = 1 \qquad 19 = \lfloor 58 / 3 \rfloor$$

$$a_1 = 19 \bmod 3 = 1 \qquad 6 = \lfloor 19 / 3 \rfloor$$

$$a_2 = 6 \bmod 3 = 0 \qquad 2 = \lfloor 6 / 3 \rfloor$$

$$a_3 = 2 \bmod 3 = 2 \qquad 0 = \lfloor 2 / 3 \rfloor$$

$$(2011)_3 = 2 \times 3^3 + 0 \times 3^2 + 1 \times 3^1 + 1 \times 3^0 = 58$$

Group 群

- **Definition** A **group** $(G, *)$ is a set G with an operation $*$, such that the following conditions are satisfied：

  1) Closure $a * b \in G$ for all $a, b \in G$
  2) Associativity $a*(b*c) = (a*b)*c$ for all $a, b, c \in G$
  3) Identity There is an element $e \in G$ such that $a = a * e = e * a$ for each $a \in G$
  4) Inverse For each $a \in G$, there is an element $b \in G$ such that $a * b = b * a = e$

Group

- **Example** Each of the following sets with the specified operation is a group
  - $\boldsymbol{Z}$, $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{C}$ with + (addition)
  - $\boldsymbol{Q}^*$, $\boldsymbol{R}^*$, $\boldsymbol{C}^*$ with × (multiplication)
  - $5\boldsymbol{Z} = \{ 5a \mid a \in \boldsymbol{Z} \}$ with +
  - $\{1, -1\}$ with ×
  - $\boldsymbol{Z}_6 = \{0, 1, 2, 3, 4, 5\}$ with + modulo 6
  - $\boldsymbol{Z}_7^* = \{1, 2, 3, 4, 5, 6\}$ with × modulo 7
  - $\{(x, y) \in \boldsymbol{R}^2 \mid y^2 = x^3 + ax + b\} \cup \{\infty\}$ with point addition and point doubling laws on elliptic curves

Abelian Group 交換群

- **Definition** A group $(G, *)$ is **commutative** or **abelian** if $a * b = b * a$ for all $a, b \in G$
- **Example**
  - $\boldsymbol{Z}, \boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{C}$ with + are commutative
  - $\boldsymbol{Z}_9{}^* = \{1, 2, 4, 5, 7, 8\}$ with [× modulo 9] is commutative
  - $\boldsymbol{Z}_p{}^* = \{1, 2, \ldots, p-1\}$ with [× modulo p] is commutative for every prime p

×	1	2	4	5	7	8
1	1	2	4	5	7	8
2	2	4	8	1	5	7
4	4	8	7	2	1	5
5	5	1	2	7	8	4
7	7	5	1	8	4	2
8	8	7	5	4	2	1

Cyclic Group 循環群

- **Definition** A group $(G, *)$ is **cyclic** if there exists a **generator** $g \in G$ such that every $a \in G$ is of the form $a = g * \ldots * g$ (n copies) for some $n \in \boldsymbol{Z}$
- **Example**
  - $(\boldsymbol{Z}, +)$ is cyclic with generators 1 and –1
  - $(\boldsymbol{Z}_7^*, \times)$ is cyclic: $\{1 = 3^0 = 3^6, 2 = 3^2, 3 = 3^1, 4 = 3^4, 5 = 3^5, 6 = 3^3\}$
  - $(\boldsymbol{Z}_9^*, \times)$ is cyclic with generators 2 and 5
- **Example**
  - $(\boldsymbol{Q}, +)$ is not cyclic
  - $(\boldsymbol{Z}_8^*, \times)$ is not cyclic (Klein 4)

×	1	3	5	7
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

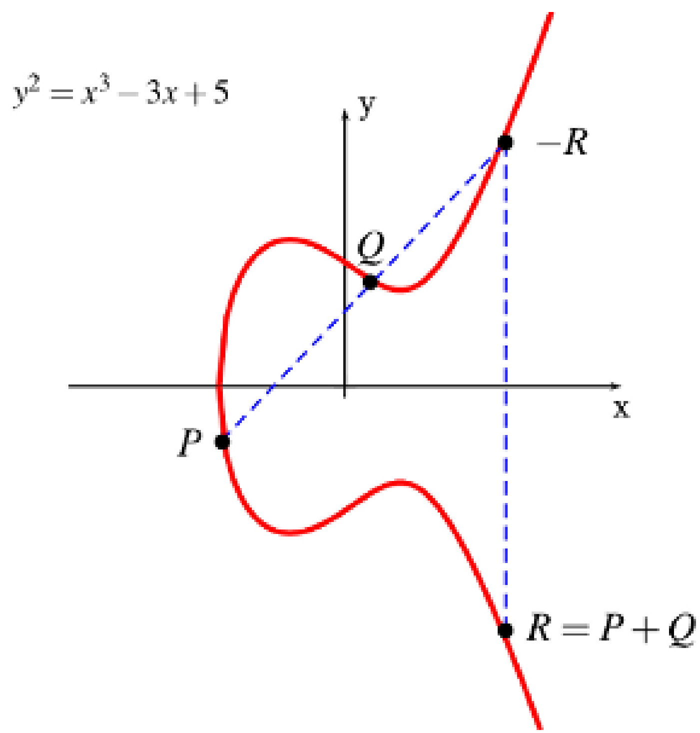
Group Order

- **Definition** The **order** (denoted as $|G|$) of a group $(G, *)$ is the number of the elements in G
- **Example**
  - $|\boldsymbol{Z}_p| = p$, $|\boldsymbol{Z}_p^*| = p - 1$ for any prime p
  - $|\boldsymbol{Z}_9^*| = 6$
  - $|\boldsymbol{Z}_n^*| = |\{a \in \boldsymbol{Z}_n \mid \gcd(a, n) = 1\}| = \phi(n)$
    - Euler ϕ-function [ϕ : phi]

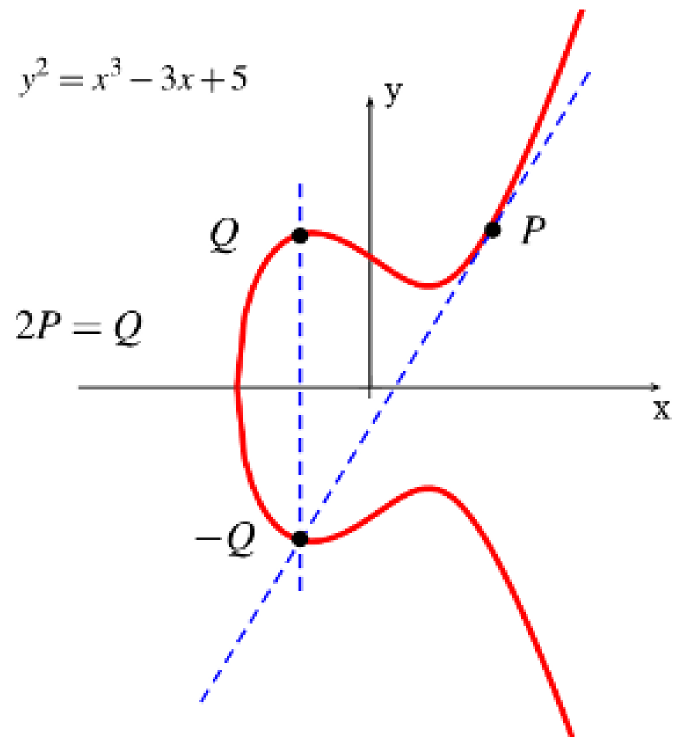
Elliptic Curves

Elliptic Curve 橢圓曲線

- The rich and deep theory of Elliptic Curves has been studied by mathematicians over 150 years
- Elliptic Curve over ***R*** : $y^2 = x^3 + ax + b$



Point Addition



Point Doubling

Image Courtesy: http://www.embedded.com/design/safety-and-security/4396040/An-Introduction-to-Elliptic-Curve-Cryptography

質數體 (Prime Field) 上的曲線

Addition:

$(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$

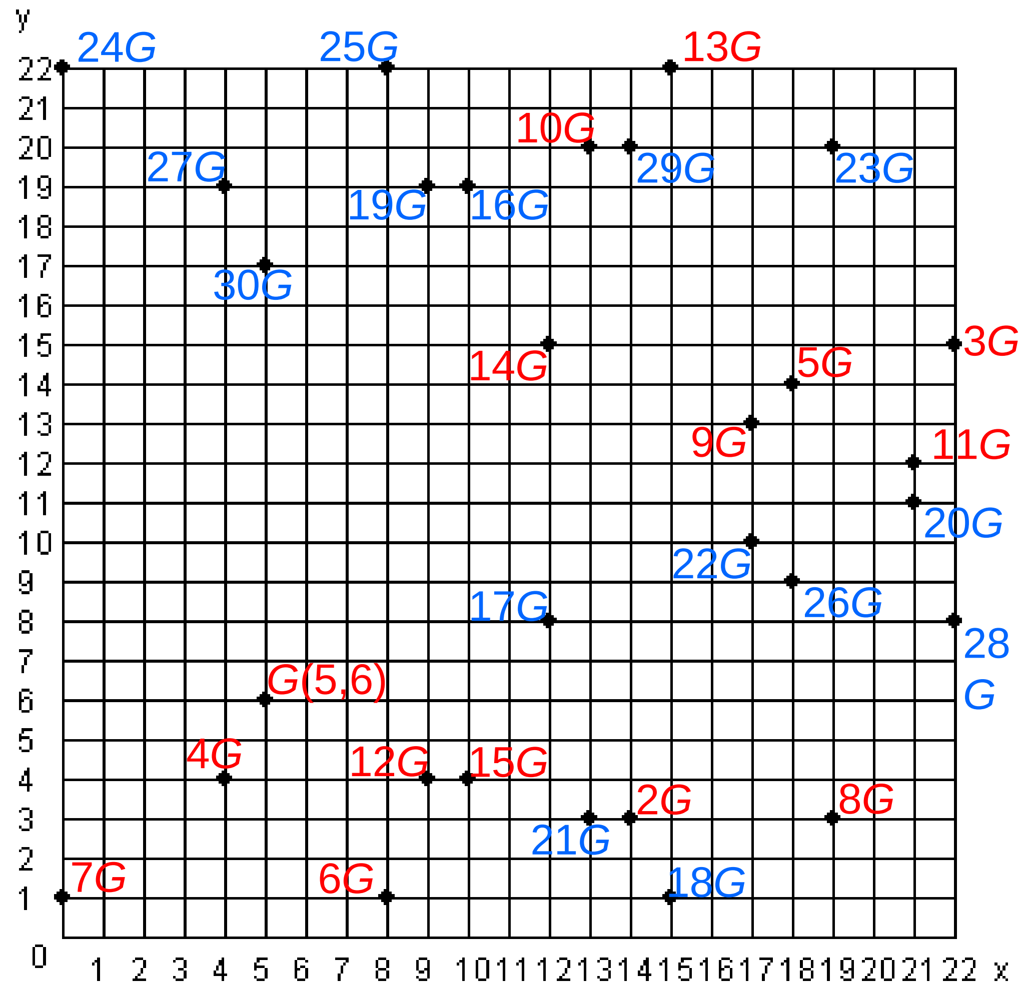
Doubling:

$(x_3, y_3) = [2]\,(x_1, y_1)$

$$s = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} \bmod p & \text{(addition)} \\[2ex] \dfrac{3x_1^2 + a}{2y_1} \bmod p & \text{(doubling)} \end{cases}$$

$$x_3 = s^2 - x_1 - x_2 \mod p$$

$$y_3 = s(x_1 - x_3) - y_1 \mod p$$



$y^2 = x^3 + 5x + 1$ over F_{23}

30 solutions

Example

- Given $E: y^2 = x^3 + 2x + 2 \mod 17$ and point $P = (5, 1)$

Goal: Compute $2P = P + P = (5, 1) + (5, 1) = (x_3, y_3)$

$$s = \frac{3x_1^2 + a}{2y_1} = (2 \cdot 1)^{-1}(3 \cdot 5^2 + 2) = 2^{-1} \cdot 9 \equiv 9 \cdot 9 \equiv 13 \mod 17$$

$$x_3 = s^2 - x_1 - x_2 = 13^2 - 5 - 5 = 159 \equiv 6 \mod 17$$

$$y_3 = s(x_1 - x_3) - y_1 = 13(5 - 6) - 1 = -14 \equiv 3 \mod 17$$

Finally $2P = (5, 1) + (5, 1) = (6, 3)$

Example

- The points on an elliptic curve and the point at infinity $\mathcal{O}$ form cyclic subgroups

$$
\begin{array}{ll}
2P = (5, 1) + (5, 1) = (6, 3) & 11P = (13, 10) \\
3P = 2P + P = (10, 6) & 12P = (0, 11) \\
4P = (3, 1) & 13P = (16, 4) \\
5P = (9, 16) & 14P = (9, 1) \\
6P = (16, 13) & 15P = (3, 16) \\
7P = (0, 6) & 16P = (10, 11) \\
8P = (13, 7) & 17P = (6, 14) \\
9P = (7, 6) & 18P = (5, 16) \\
10P = (7, 11) & 19P = \mathcal{O}
\end{array}
$$

This elliptic curve has order $\#E = |E| = 19$ since it contains 19 points in its cyclic group.

Double and Add

- Example

$$17\,P \;=\; (2P) + P + \ldots. + P$$

[1 doubling & 15 additions]

$$= \;(10001)_2\,P \;=\; 2(2(2(2P))) + P$$

[4 doublings & 1 addition]

Double and Add

Example: $26P = (11010_2)P = (d_4 d_3 d_2 d_1 d_0)_2\ P$.

Step		
#0	$P = \mathbf{1}_2 P$	inital setting
#1*a*	$P + P = 2P = \mathbf{10}_2 P$	DOUBLE (bit d_3)
#1*b*	$2P + P = 3P = 10^2 P + 1_2 P = \mathbf{11}_2 P$	ADD (bit $d_3 = 1$)
#2*a*	$3P + 3P = 6P = 2(11_2 P) = \mathbf{110_2} P$	DOUBLE (bit d_2)
#2*b*		no ADD ($d_2 = 0$)
#3*a*	$6P + 6P = 12P = 2(110_2 P) = \mathbf{1100_2} P$	DOUBLE (bit d_1)
#3*b*	$12P + P = 13P = 1100_2 P + 1_2 P = \mathbf{1101}_2 P$	ADD (bit d_1=1)
#4*a*	$13P + 13P = 26P = 2(1101_2 P) = \mathbf{11010_2} P$	DOUBLE (bit d_0)
#4*b*		no ADD ($d_0 = 0$)

Bitcoin和Ethereum使用的曲線

The elliptic curve domain parameters over $\mathbb{F}_p$ associated with a Koblitz curve `secp256k1` are specified by the sextuple $T = (p, a, b, G, n, h)$ where the finite field $\mathbb{F}_p$ is defined by:

$$p = \text{FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F}$$
$$= 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$$

256-bit prime

The curve E: $\boxed{y^2 = x^3 + ax + b}$ over $\mathbb{F}_p$ is defined by:

$$a = \text{00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000}$$
$$b = \text{00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007}$$

橢圓曲線 secp256k1

https://en.bitcoin.it/wiki/Secp256k1

The base point G in compressed form is:

$$G = \text{02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798}$$

and in uncompressed form is:

$$G = \text{04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8}$$

Finally the order n of G and the cofactor are:

$$n = \text{FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141}$$

256-bit prime

$$h = \text{01}$$

Key Pairs 金鑰對

- The base point G is fixed on the given Elliptic Curve
- $P = [m]\,G$
  - Given m, it is **easy and fast** to find the point P
    - Using "double and add" for scalar multiplication
  - Given P, it is **extremely hard** to find the integer m
    - Elliptic Curve Discrete Logarithm Problem (橢圓曲線離散對數問題)
  - A randomly generated integer m is a **private key**
    - A private key is used to sign Bitcoin transactions with ECDSA
  - The point P is the **public key** corresponding to m
    - A public key is used by other nodes to verify Bitcoin transactions
    - **A Bitcoin address is the hash value of a public key *P***

NIST Curve Standards in FIPS 186

Table D-1: Bit Lengths of the Underlying Fields of the Recommended Curves

Bit Length of *n*	Prime Field	Binary Field
161 – 223	**len**(p) = 192	m = 163
224 – 255	**len**(p) = 224	m = 233
256 – 383	**len**(p) = 256	m = 283
384 – 511	**len**(p) = 384	m = 409
≥ 512	**len**(p) = 521	m = 571

NIST Curves over Prime Fields

D.1.2 Curves over Prime Fields

For each prime p, a pseudo-random curve

$$E : y^2 \equiv x^3 - 3x + b \pmod{p}$$

of prime order n is listed[4]. (Thus, for these curves, the cofactor is always $h = 1$.) The following parameters are given:

- The prime modulus p
- The order n
- The 160-bit input seed *SEED* to the SHA-1 based algorithm (i.e., the domain parameter seed)
- The output c of the SHA-1 based algorithm

---

[4] The selection $a \equiv -3$ for the coefficient of x was made for reasons of efficiency; see IEEE Std 1363-2000.

- The coefficient b (satisfying $b^2\, c \equiv -27 \pmod{p}$)
- The base point x coordinate G_x
- The base point y coordinate G_y

The integers p and n are given in decimal form; bit strings and field elements are given in hexadecimal.

Curve P-256

D.1.2.3 Curve P-256

$p =$ 115792089210356248762697446949407573530086143415290314195533631308867097853951

$n =$ 115792089210356248762697446949407573529996955224135760342422259061068512044369

$SEED =$ c49d3608 86e70493 6a6678e1 139d26b7 819f7e90

$c =$ 7efba166 2985be94 03cb055c 75d4f7e0 ce8d84a9 c5114abc af317768 0104fa0d

$b =$ 5ac635d8 aa3a93e7 b3ebbd55 769886bc 651d06b0 cc53b0f6 3bce3c3e 27d2604b

$G_x =$ 6b17d1f2 e12c4247 f8bce6e5 63a440f2 77037d81 2deb33a0 f4a13945 d898c296

$G_y =$ 4fe342e2 fe1a7f9b 8ee7eb4a 7c0f9e16 2bce3357 6b315ece cbb64068 37bf51f5

NIST Curves over Prime Fields

P-192: $p = 2^{192} - 2^{64} - 1$, $a = -3$, $h = 1$,

$b =$ 0x 64210519 E59C80E7 0FA7E9AB 72243049 FEB8DEEC C146B9B1

$n =$ 0x FFFFFFFF FFFFFFFF FFFFFFFF 99DEF836 146BC9B1 B4D22831

P-224: $p = 2^{224} - 2^{96} + 1$, $a = -3$, $h = 1$,

$b =$ 0x B4050A85 0C04B3AB F5413256 5044B0B7 D7BFD8BA 270B3943 2355FFB4

$n =$ 0x FFFFFFFF FFFFFFFF FFFFFFFF FFFF16A2 E0B8F03E 13DD2945 5C5C2A3D

P-256: $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$, $a = -3$, $h = 1$,

$b =$ 0x 5AC635D8 AA3A93E7 B3EBBD55 769886BC 651D06B0 CC53B0F6 3BCE3C3E
27D2604B

$n =$ 0x FFFFFFFF 00000000 FFFFFFFF FFFFFFFF BCE6FAAD A7179E84 F3B9CAC2
FC632551

P-384: $p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$, $a = -3$, $h = 1$,

$b =$ 0x B3312FA7 E23EE7E4 988E056B E3F82D19 181D9C6E FE814112 0314088F
5013875A C656398D 8A2ED19D 2A85C8ED D3EC2AEF

$n =$ 0x FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF C7634D81
F4372DDF 581A0DB2 48B0A77A ECEC196A CCC52973

P-521: $p = 2^{521} - 1$, $a = -3$, $h = 1$,

$b =$ 0x 00000051 953EB961 8E1C9A1F 929A21A0 B68540EE A2DA725B 99B315F3
B8B48991 8EF109E1 56193951 EC7E937B 1652C0BD 3BB1BF07 3573DF88
3D2C34F1 EF451FD4 6B503F00

$n =$ 0x 000001FF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFA 51868783 BF2F966B 7FCC0148 F709A5D0 3BB5C9B8
899C47AE BB6FB71E 91386409

Security Level

(NIST) SP 800-57 Part 1

Bits of security	Symmetric key algorithms	Finite Field Cryptography (FFC, e.g., DSA, D-H)	Integer Factorization Cryptography (IFC, e.g., RSA)	Elliptic Curve Cryptography (ECC, e.g., ECDSA)
80	2TDEA*	$L = 1024$ $N = 160$	$k = 1024$	$f = 160\text{-}223$
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224\text{-}255$
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256\text{-}383$
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384\text{-}511$
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

* The assessment of at least 80-bits of security for 2TDEA is based on the assumption that an attacker has no more than 2^{40} matched plaintext and ciphertext blocks ([ANSX9.52], Annex B).

Keys, Addresses

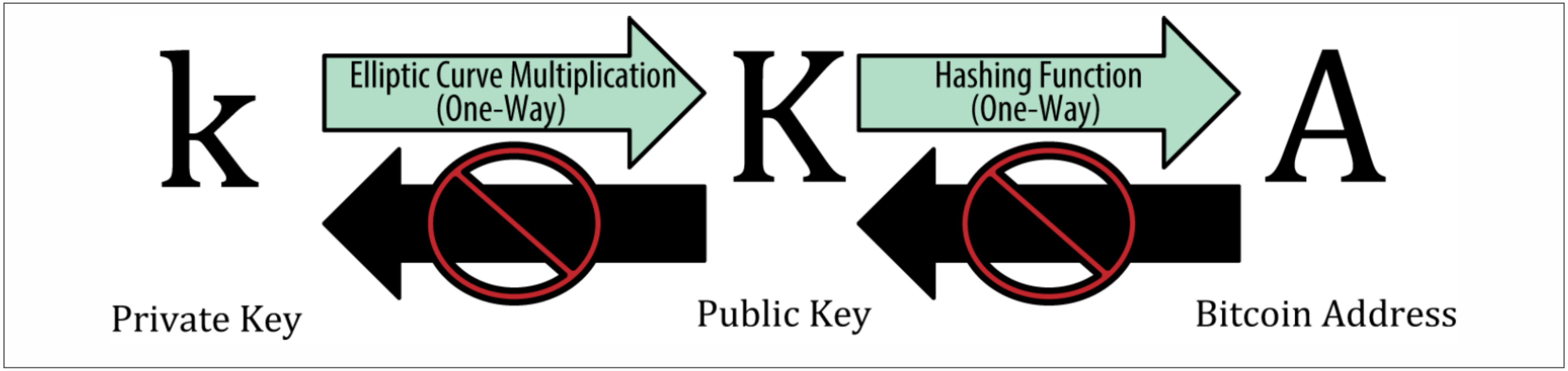


Figure 4-1. Private key, public key, and bitcoin address

The size of bitcoin's private key space, 2^{256} is an unfathomably large number. It is approximately 10^{77} in decimal. The visible universe is estimated to contain 10^{80} atoms.

"Mastering Bitcoin" by Andreas M. Antonopoulos

Bitcoin Address

- Address = RIPEMD160(SHA256(public key representation))
- Example
  - ECDSA private key = 18E14A7B6A307F426A94F8114701E7C8E774E7F9A47E2C2035DB29A206321725
  - Public key *P* = 04 50863AD64A87AE8A2FE83C1AF1A8403CB53F53E486D8511DAD8A04887E5B235
    22CD470243453A299FA9E77237716103ABC11A1DF38855ED6F2EE187E9C582BA6
  - SHA256(*P*) = 600FFE422B4E00731A59557A5CCA46CC183944191006324A447BDB2D98D4B408
  - RIPEMD160(SHA256(*P*)) = 010966776006953D5567439E5E39F86A0D273BEE
  - Address (Base58Check encoded): 16UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM
  - https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses#How_to_create_Bitcoin_Address
- Base58 is a set of lower and capital letters and numbers without (0, O, l, I), i.e., 0 (number zero), O (capital o), l (lower L), I (capital i)

HITCON Enterprise 2014
台灣駭客年會 企業場

私鑰數據庫？

比特币 (Bitcoin)

比特币「私钥数据库」是怎么回事？

1：All bitcoin private keys

2：比特币私钥数据库

2 条评论　分享

查看全部 4 个回答

知乎用户

10 人赞同

转载自贴吧 原地址 那些说比特币算法可以被轻易破解的同学

先说比特币地址和私钥，你必须要明白比特币的加密学原理是基于椭圆曲线加密算法的，具体来说是secp256k1

比特币地址和私钥是由ECDSA椭圆曲线加密算法计算出来的，由ECDSA私钥计算出我们常用的Bitcoin-qt格式比特币地址需要有十个步骤

https://www.zhihu.com/question/23608006/answer/25141783

ECDSA

ECDSA Signing 簽章

Parameter	
CURVE	the elliptic curve field and equation used
G	elliptic curve base point, a generator of the elliptic curve with large prime order *n*
n	integer order of *G*, means that $n * G = O$

Suppose Alice wants to send a signed message to Bob. Initially, they must agree on the curve parameters $(CURVE, G, n)$. In addition to the field and equation of the curve, we need G, a base point of prime order on the curve; n is the multiplicative order of the point G.

Alice creates a key pair, consisting of a private key integer d_A, randomly selected in the interval $[1, n-1]$; and a public key curve point $Q_A = d_A * G$. We use $*$ to denote elliptic curve point multiplication by a scalar.

For Alice to sign a message m, she follows these steps:

1. Calculate $e = \mathrm{HASH}(m)$, where HASH is a cryptographic hash function, such as SHA-1.
2. Let z be the L_n leftmost bits of e, where L_n is the bit length of the group order n.
3. Select a random integer k from $[1, n-1]$.
4. Calculate the curve point $(x_1, y_1) = k * G$.
5. Calculate $r = x_1 \bmod n$. If $r = 0$, go back to step 3.
6. Calculate $\boxed{s = k^{-1}(z + r d_A) \bmod n}$. If $s = 0$, go back to step 3.
7. The signature is the pair (r, s).

k : ephemeral key

http://en.wikipedia.org/wiki/Elliptic_Curve_DSA

ECDSA Verification 驗章

For Bob to authenticate Alice's signature, he must have a copy of her public-key curve point Q_A. Bob can verify Q_A is a valid curve point as follows:

1. Check that Q_A is not equal to the identity element O, and its coordinates are otherwise valid
2. Check that Q_A lies on the curve
3. Check that $n * Q_A = O$

After that, Bob follows these steps:

1. Verify that r and s are integers in $[1, n-1]$. If not, the signature is invalid.
2. Calculate $e = \mathrm{HASH}(m)$, where HASH is the same function used in the signature generation.
3. Let z be the L_n leftmost bits of e.
4. Calculate $w = s^{-1} \bmod n$.
5. Calculate $u_1 = zw \bmod n$ and $u_2 = rw \bmod n$.
6. Calculate the curve point $(x_1, y_1) = u_1 * G + u_2 * Q_A$.
7. The signature is valid if $r \equiv x_1 \pmod{n}$, invalid otherwise.

Note that using Straus's algorithm (also known as Shamir's trick) a sum of two scalar multiplications $u_1 * G + u_2 * Q_A$ can be calculated faster than with two scalar multiplications.[3]

http://en.wikipedia.org/wiki/Elliptic_Curve_DSA

Ephemeral Key & RNG

- The **entropy**, **secrecy**, and **uniqueness** of the DSA/ECDSA **random ephemeral key *k*** is critical
  - Violating any one of the above three requirements can reveal the entire private key to an attacker
  - Using the same value twice (even while keeping *k* secret), using a predictable value, or leaking even a few bits of *k* in each of several signatures, is enough to break DSA/ECDSA
- [December 2010] The ECDSA private key used by **Sony** to sign software for the **PlayStation 3** game console was recovered, because Sony implemented *k* as static instead of random

http://en.wikipedia.org/wiki/Digital_Signature_Algorithm
http://en.wikipedia.org/wiki/Elliptic_Curve_DSA

Ephemeral Key & RNG

- [August 2013] Bugs in some implementations of the Java class *SecureRandom* sometimes generated collisions in *k*, allowing in stealing **bitcoins** from the containing wallet on **Android app**
  - http://www.theregister.co.uk/2013/08/12/android_bug_batters_bitcoin_wallets
- [August 2013] 158 accounts had used the same signature nonces *r* value in more than one signature. The total remaining balance across all 158 accounts is only 0.00031217 BTC. The address, 1HKywxiL4JziqXrzLKhmB6a74ma6kxbSDj, appears to have stolen bitcoins from 10 of these addresses. This account made 11 transactions between March and October 2013. These transactions have netted this account over 59 bitcoins (approximately $12,000 USD).
  - http://eprint.iacr.org/2013/734.pdf
- This issue can be prevented by deriving *k* deterministically from the **private key** and the **message hash**, as described by **RFC 6979**

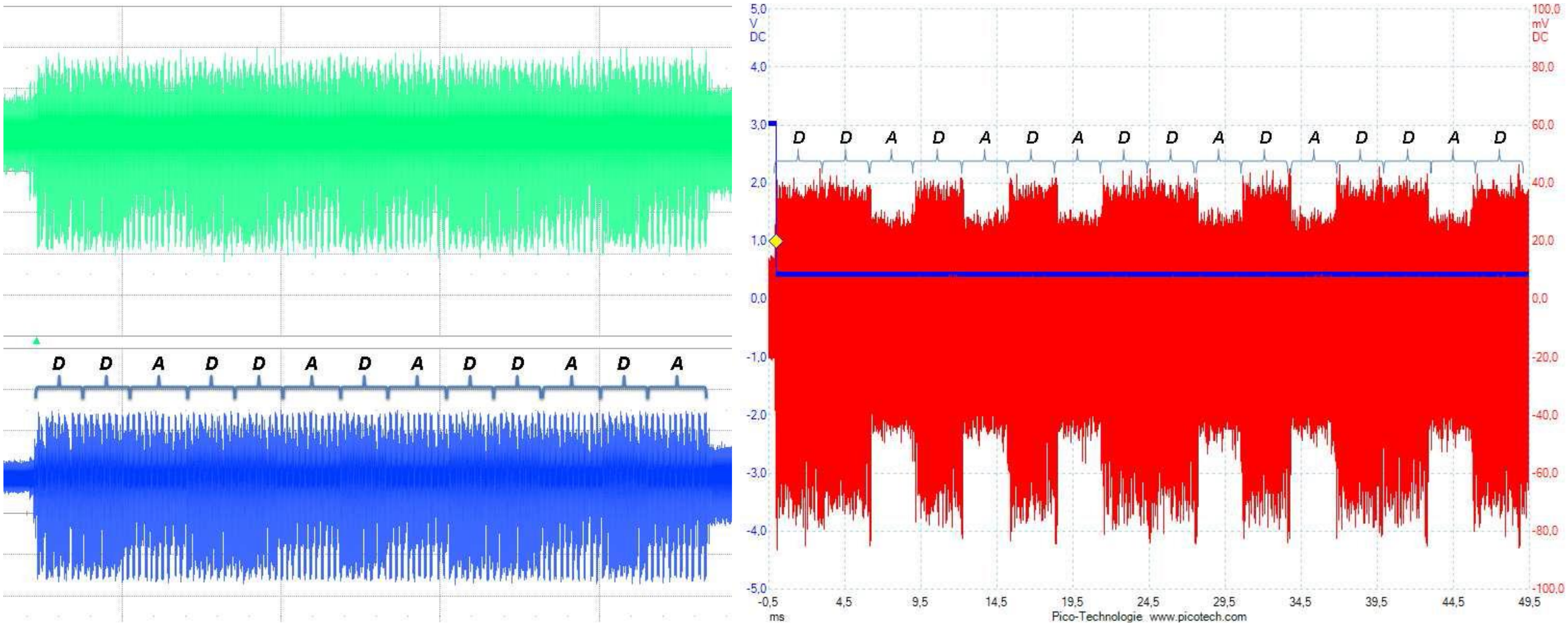
Side-Channel Attacks

ECDSA (Elliptic-Curve Digital Signature Algorithm)

- 必須安全計算點的倍數

Key creation	
Choose secret signing key $1 < s < q - 1$. Compute $\boxed{V = sG \in E(\mathbb{F}_p).}$ Publish the verification key V.	
Signing	
Choose document $d \bmod q$. Choose random element $e \bmod q$. Compute $\boxed{eG \in E(\mathbb{F}_p)}$ and then, $s_1 = x(eG) \bmod q$ and $s_2 \equiv (d + ss_1)e^{-1} \pmod{q}$. Publish the signature (s_1, s_2).	

Side-Channel Attack



D (double) or **A** (add) depends on the bits of **Secret Key**

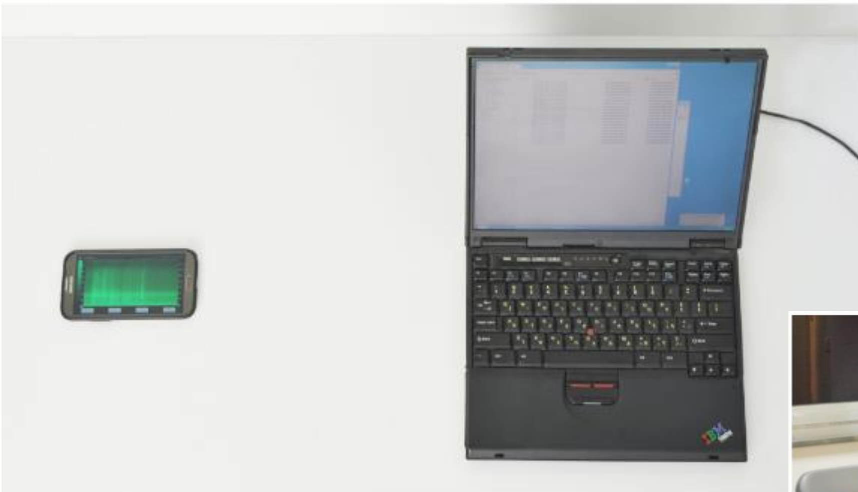
Image Courtesy https://eprint.iacr.org/2015/354.pdf

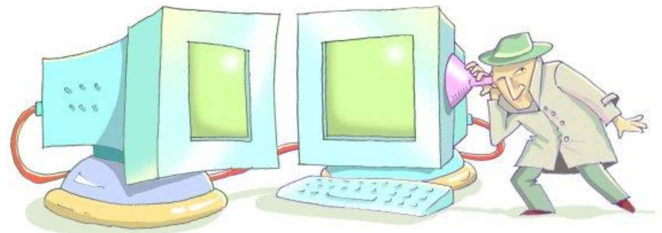
隔空抓鑰——ECDSA Key Extraction from Mobile Devices

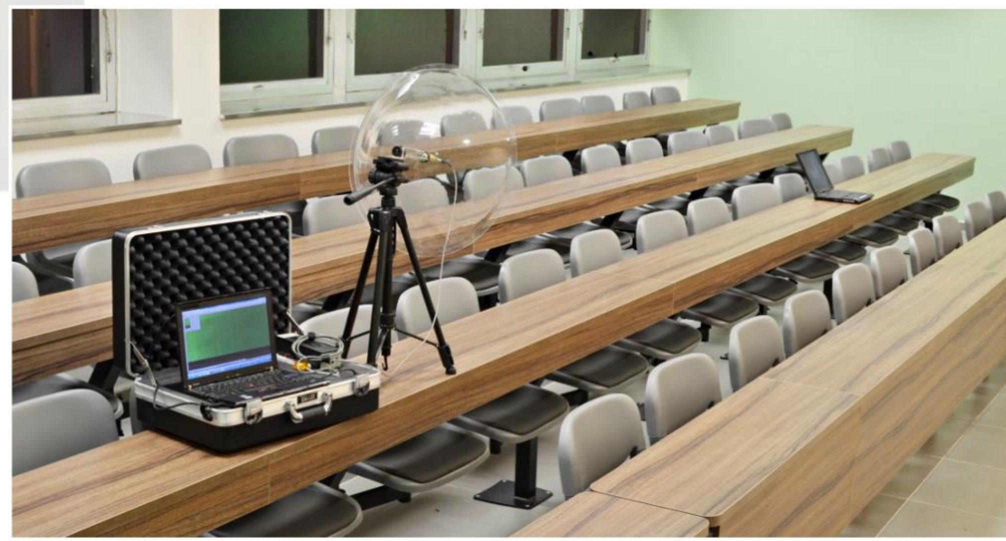
- Fully extract secret signing keys from OpenSSL and CoreBitcoin running on iOS devices


Sourse: https://www.tau.ac.il/~tromer/mobilesc

Acoustic SCA



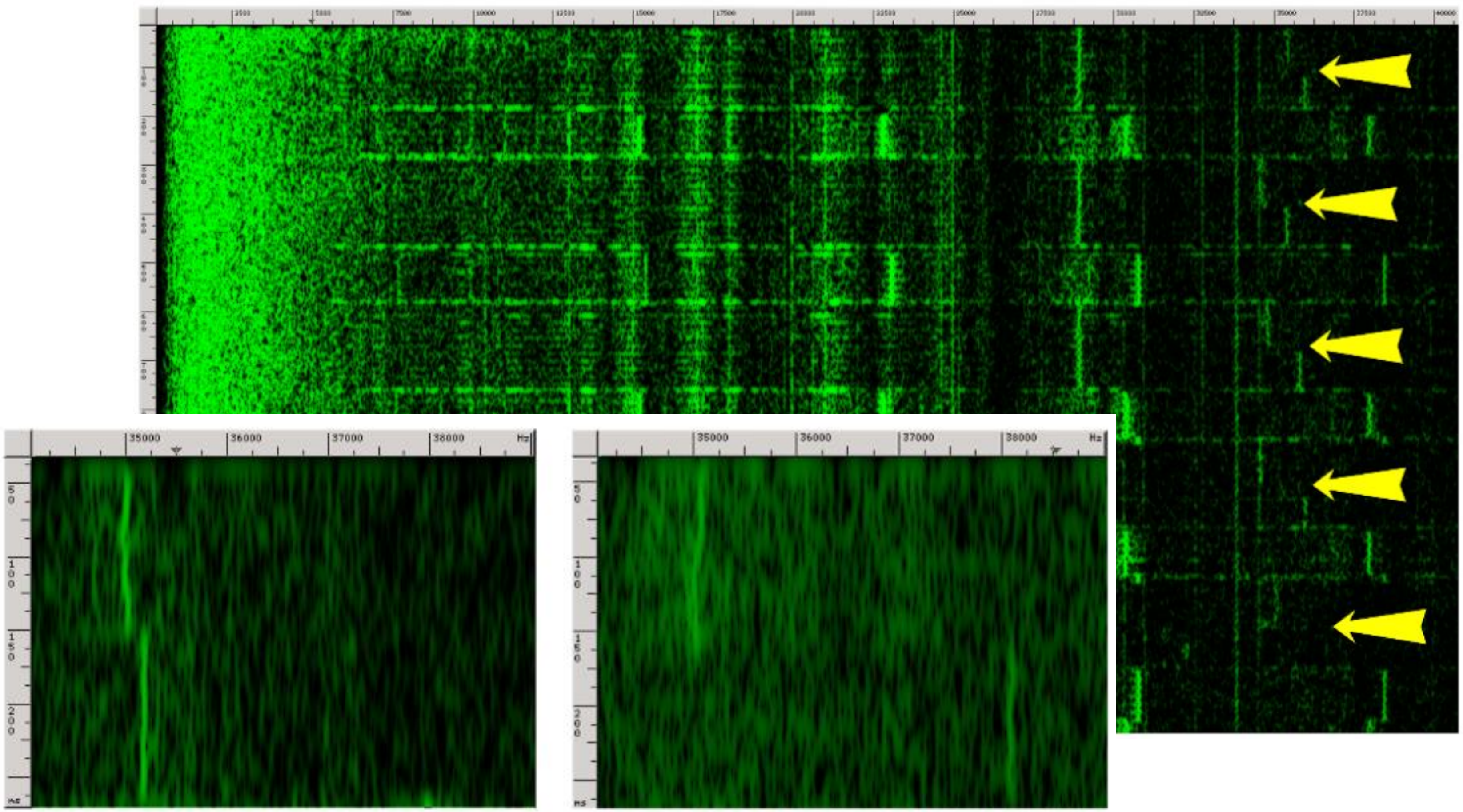




The noise made by a laptop running GnuPG RSA-4096 is collected

Daniel Genkin, Adi Shamir, Eran Tromer, “RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis”, 2013 http://www.cs.tau.ac.il/~tromer/acoustic

Acoustic SCA



(a) attacked bit is zero

(b) attacked bit is one

Image Courtesy https://www.cs.tau.ac.il/~tromer/papers/acoustic-20131218.pdf


Side Channel Attack
旁通道攻擊！

Double and Add Always

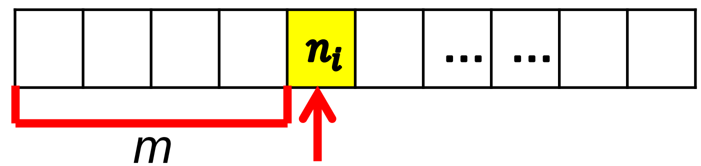
- Double-and-Add 就算遇到 0 也做 Add
  - For $i = k - 1$ down to 0
    - $T \leftarrow 2T$
    - If $n_i = 1$
      - $T \leftarrow T + P$

- If $n_i = 0$
  - Trash $\leftarrow T + P$

  - 可以抵擋 Simple Power Analysis
  - 無法抵擋 Fault Injection (Fault Attack)
    - Trash 出錯不影響結果，可知道該位元為 0

Montgomery Ladder

- 仿照 Double-and-Add
  - 給定
    - $mP = R_1, (m+1)P = R_2$
  - 若 $n_i = 1$
    - $R_1 \leftarrow (2m+1)P = R_1 + R_2$
      $R_2 \leftarrow (2m+2)P = 2R_2$
  - 若 $n_i = 0$
    - $R_2 \leftarrow (2m+1)P = R_1 + R_2$
      $R_1 \leftarrow (2m)P = 2R_1$
  - 最後回傳 R_1



不論 0 或 1
都做一個 Add
和一個 Double

Example: $[26]P$

- $26P = (11010_2)P = (k_4k_3k_2k_1k_0)P$

	$\boldsymbol{R_1}$	$\boldsymbol{R_1}$	$\boldsymbol{R_2}$
Initial	P	$(1_2)P$	$2P$
$k_3 = 1$	$3P = 2P + P$	$(11_2)P$	$4P = 2 \times 2P$
$k_2 = 0$	$6P = 2 \times 3P$	$(110_2)P$	$7P = 3P + 4P$
$k_1 = 1$	$13P = 6P + 7P$	$(1101_2)P$	$14P = 2 \times 7P$
$k_0 = 0$	$26P = 2 \times 13P$	$(11010_2)P$	$27P = 13P + 14P$

- Return $R_1 = 26P$

Scalar Multiplications

Input: $\boldsymbol{P}, k = (1, k_{\ell-2}, \ldots, k_0)_2$
Output: $\boldsymbol{Q} = k\boldsymbol{P}$

$\boldsymbol{R_0} \leftarrow \boldsymbol{P}$
for $j = \ell - 2$ downto 0 do
  $\boldsymbol{R_0} \leftarrow 2\boldsymbol{R_0}$; $\boldsymbol{R_1} \leftarrow \boldsymbol{R_0} + \boldsymbol{P}$
  $b \leftarrow k_j$; $\boldsymbol{R_0} \leftarrow \boldsymbol{R_b}$
endfor
return $\boldsymbol{R_0}$

(a) Double-and-add *always* [5]

Input: $\boldsymbol{P}, k = (1, k_{\ell-2}, \ldots, k_0)_2$
Output: $\boldsymbol{Q} = k\boldsymbol{P}$

$\boldsymbol{R_0} \leftarrow \boldsymbol{P}$; $\boldsymbol{R_1} \leftarrow 2\boldsymbol{P}$
for $j = \ell - 2$ downto 0 do
  $b \leftarrow k_j$
  $\boldsymbol{R_{1-b}} \leftarrow \boldsymbol{R_0} + \boldsymbol{R_1}$; $\boldsymbol{R_b} \leftarrow 2\boldsymbol{R_b}$
endfor
return $\boldsymbol{R_0}$

(b) Montgomery ladder [20, 12]

http://joye.site88.net/papers/Joy03ecc.pdf