[CS304] Introduction to Cryptography and Network Security

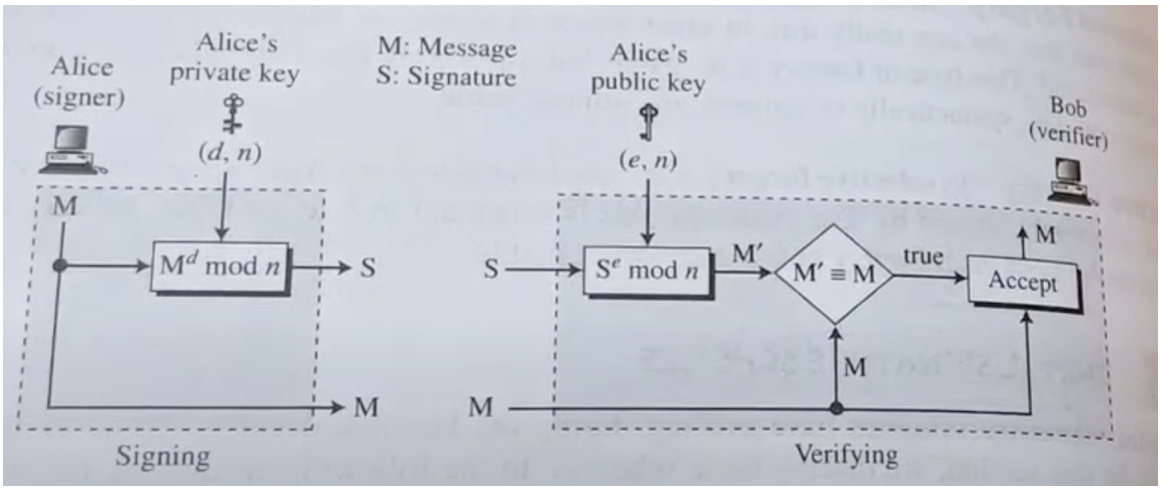
Course Instructor: Dr. Dibyendu Roy — Winter 2022-2023
Scribed by: Akshay (202051018) — Lecture (Week 12)

1 RSA Signature Algorithm (A Digital Signature Algorithm (DSA))

We will use RSA enc/dec with little tweaks.

Alice:
M (message)
n = p.q (two large prime numbers)
$\phi(n) = (p-1).(q-1)$
$e.d \equiv 1 \ mod \ (\phi(n))$
Public key (PK) : (e,n)
Secret key (SK) : (d,p,q)
$S = M^d \ mod \ n$ (signature)

Bob:
$V = S^e \ mod \ n$
If V = M then output 1 else output 0.



For signing we use private key.
For verification we use public key.

1.1 Formal Definition of a DSA

P : plaintext space
S : signature space
K : key space
Sign : signing algorithm
V : verification algorithm

$$sign(p,k) = s$$

$$V(p,s,k) = \begin{cases} 1 & \text{if } s = sign(p,k) \\ 0 & \text{otherwise} \end{cases}$$

2 RSA Enc Analysis for Signing

$$\text{n} = \text{p.q}$$
$$\phi(n) = (p-1).(q-1)$$
$$e.d \equiv 1 \ mod \ \phi(n)$$
$$\text{PK : (e,n)}$$
$$\text{SK : (d,p,q)}$$
$$C = m^e \ mod \ n$$

where, C is ciphertext and m is message/plaintext.

$C_1 = m_1^e \ mod \ n$
$C_2 = m_2^e \ mod \ n$
Encrypted two different messages using same key.

2.1 Computation on Encrypted data

$$C_1 \times C_2 = m_1^e m_2^e \ mod \ n$$
$$= (m_1 m_2)^e \ mod \ n$$

$C_1 \times C_2$ is the ciphertext corresponding to message : m_1 times m_2.
If I have two ciphertexts, I can multiply them and ensure that this new ciphertext is corresponding to the message we get by multiplying the messages corresponding to original two ciphertexts.
We are able to compute encryption of multiplication of two messages *without performing the encryption.*

$C_1 + C_2$: not possible ($\because \not\equiv (m_1 + m_2)^e$)

Constant times ciphertext is possible.
$a^e C_1 \equiv a^e m_1^e \equiv (am_1)^e \ mod \ n$
So, we can get the ciphertext corresponding to message a times m_1 by multiplying C_1 with a^e.

Note: There are certain algorithms in which all such computations are possible. All those algorithms are known as **Fully Homomorphic Encryptions**.

2.1.1 Comment on RSA Signing Algo

Same concept (of computation) is applicable on RSA signing algorithm.

$$s_1 = m_1^d \ mod \ n$$
$$s_2 = m_2^d \ mod \ n$$

$$s_1 \times s_2 = (m_1 \times m_2)^d \ mod \ n$$

If we multiply two signatures, we will get signature corresponding to $m_1 \times m_2$ (computation on authenricated data).
$\implies$ Forging a signature on message in RSA signature algorithm is very easy. We can produce a signature without using secret key.

2.2 How to Prevent Signature Forging via Computation on Authenticated Data?

Whatever the signature algorithm we are using, sign on the hash of the message.
On RSA:
$S_1 = (h(m_1))^d \ mod \ n$
$S_2 = (h(m_2))^d \ mod \ n$
$S_1 \times S_2 \neq (h(m_1 m_2))^d \ mod \ n$

How the verfication will be done? :

$$(S)^e \ mod \ n$$
$$= h(m) \ mod \ n$$

Verifier have S and m.
He/She will compute h(m) and check if h(m) is equal to $(S)^e$ or not.

3 Some Maths ..

3.1

$$ax \equiv b \ mod \ m \qquad (1)$$
$$\implies ax - my = b \qquad (2)$$

($\because$ ax-b = multiple of m or m divides ax-b)

$ax_0 + my_0 = gcd(a, m)$ (from prior knowledge)
If gcd(a,m) divides b then (2) will have solution (can be obtained by Extended Euclidean Algorithm (EEA))
gcd(a,m) divides b $\implies$ $t.gcd(a, m) = b$

$$ax_o + my_o = gcd(a, m)$$

Multiply both sides with t

$$a(tx_o) + m(ty_o) = t.gcd(a, m)$$
$$aX + mY = b$$

Equation:
$ax \equiv b \ mod \ m$ or $ax - my = b$

Solution: Exists only when gcd(a,m) divides b

$x = x_o + \frac{m}{gcd(a,m)}n$

$y = y_o + \frac{a}{gcd(a,m)}n$

where (x_o, y_o) is initial solution obtained from EEA,
n is integer ($\mathbb{Z}$)

3.2

$$x \equiv a_1 \ mod \ m_1 \qquad (1)$$

$$x \equiv a_2 \ mod \ m_2 \qquad (2)$$

Here, gcd(m_1, m_2) = 1
From eqn (1) :
$x = a_1 + m_1 y$
If this x is also a solution of (2),
$a_1 + m_1 y \equiv a_2 \ mod m_2$

$\implies m_1 y \equiv (a_2 - a_1) \ mod \ m_2$
gcd(m_1, m_2) = 1
On comparing it with result of section 3.1 :
a $\rightarrow m_1$ x $\rightarrow$ y
m $\rightarrow m_2$

$y = y_o + \frac{m_2}{gcd(m_1,m_2)}n$

$\implies y = y_o + m_2 n$

where y_o is initial solution which can be obtained by using EEA on $m_1 y \equiv (a_2 - a_1) \ mod \ m_2$
So,
$x = a_1 + m_1(y_o + m_2 n)$
$x = (a_1 + m_1 y_o) + n.m_1 m_2$
$x = x_o + n.m_1 m_2$
$x \equiv x_o \ mod \ m_1 m_2$

System:
$x \equiv a_1 \ mod \ m_1$
$x \equiv a_2 \ mod \ m_2$
$\gcd(m_1, m_2) = 1$

Solution:
$x = (a_1 + m_1 y_o) + n.m_1 m_2$
$= x_o + n.m_1 m_2$
$x \equiv x_o \ mod(m_1 m_2)$
where, y_o is initial solution which can be obtained by using EEA on $m_1 y = (a_2 - a_1) \ mod \ m_2$

3.3 Chinese Remainder Theorem (CRT)

The system of eqns :

$$x \equiv a_1 \ mod \ m_1$$
$$x \equiv a_2 \ mod \ m_2$$
$$.$$
$$.$$
$$.$$
$$x \equiv a_r \ mod \ m_r$$

when $m_1, m_2, ...m_r$ are pairwise relative prime then the above system has a unique solution under modulo $(m_1 m_2 ....m_r)$

Define δ_j

$$\delta_j = \begin{cases} 1 \ mod \ m_j \\ 0 \ mod \ m_i, \quad i \neq j \end{cases}$$

Then,
$x = \sum_{j=1}^{r} \delta_j . a_j$ satisfies all the eqns.

$x = \delta_1 a_1 + \delta_2 a_2 + ... + \delta_r a_r$

If we consider the first equation, take mod with m_1
$(x = \delta_1 a_1 + \delta_2 a_2 + ... + \delta_r a_r) \ mod \ m_1$
Every term will get nullified except $\delta_1 a_1$
$\implies x = \delta_1 a_1 mod \ m_1$
$x = a_1 \ mod \ m_1$ ($\because \delta_1 = 1 \ mod \ m_1$)

$M = m_1 . m_2 .....m_r$

$\gcd(\frac{M}{m_j}, m_j) = 1$ (because of pairwise co-primality)

$\implies$ we can find the inverse of $\frac{M}{m_j}$ under modulo m_j

$\implies \frac{M}{m_j} b_j \equiv 1 mod\ m_j$ (1)

where b_j is the inverse.

$\delta_j = \frac{M}{m_j} b_j$

- If we divide it with m_j we will get remainder = 1 (evident from (1))
- If we divide it with m_i ($i \neq$ j), we will get remainder = 0 because m_i is available in its factorisation.

3.3.1 Uniqueness

Assume $x^{'}$ is another solution of the above system then $x^{'} \equiv x\ mod\ (m_1 m_2 .... m_r)$

$x \equiv a_1\ mod\ m_1$ (1)
$x^{'} \equiv a_1\ mod\ m_1$ (2)
From (1) and (2) :
$x^{'} \equiv x\ mod\ m_1$

Similarly,

$$x \equiv x_2\ mod\ m_2$$

.

.

.

$$x \equiv x_r\ mod\ m_r$$

$\implies x^{'} \equiv x\ mod\ (m_1 m_2 .... m_r)$

because:

- Each m_i is unique and co-prime to each other
- Since $x^{'} - x$ is divisible by each m_i, it will be divisible by their product too.

$\implies x = \sum_{j=1}^{r} \delta_j a_j$ is the unique solution under modulo $(m_1 m_2 .... m_r)$

4 Beginning of Elliptic Curve Cryptography

- We do computations on curve.
- It provides better security.

1. a, b $\in \mathbb{R}$

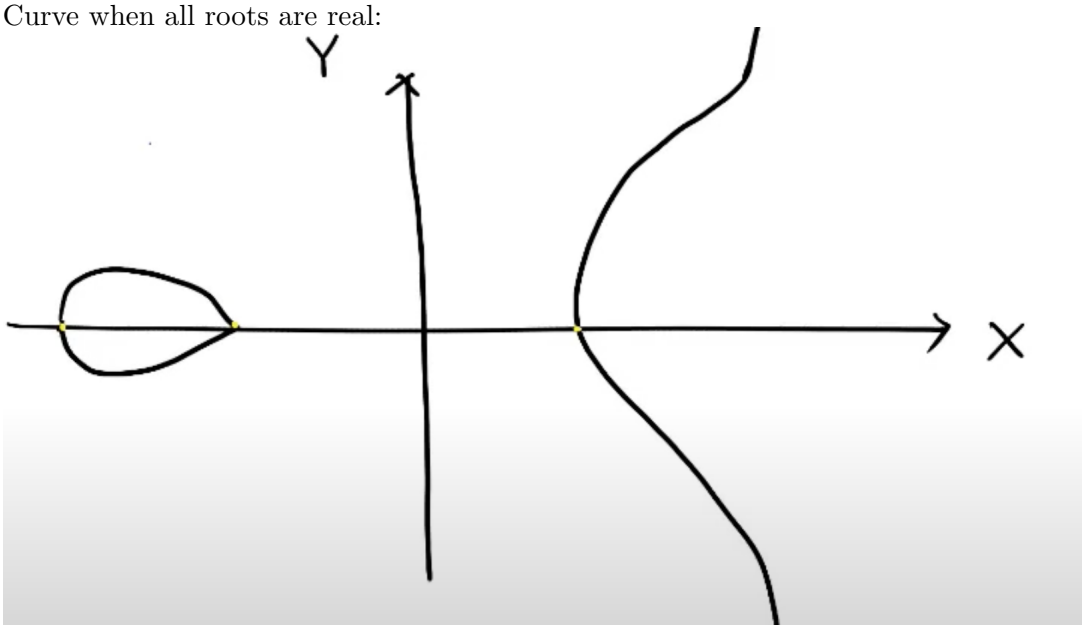
2. $4a^3 + 27b^2 \neq 0$

3. Eqn of the curve
   $y^2 = x^3 + ax + b$
   where (x,y) $\in \mathbb{R}^2$

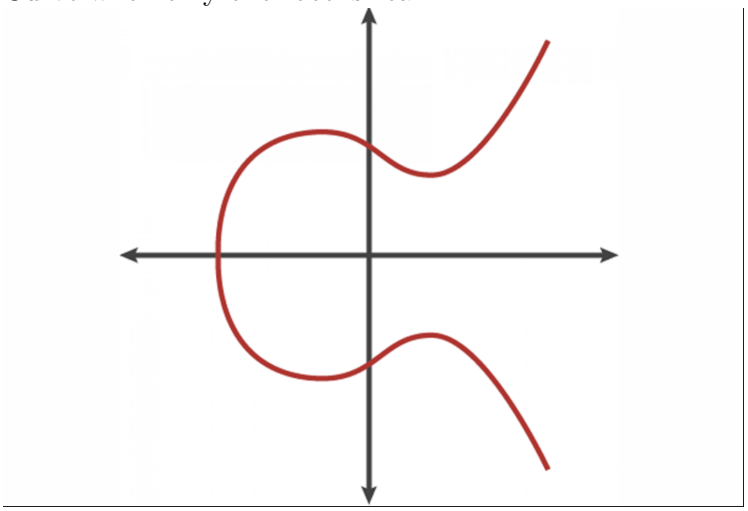
For roots, put y=0 $\implies$ $y^2 = 0$
$\implies$ $x^3 + ax + b = 0$ $\qquad (\star)$
Possibilities:
a) All three roots are real
b) One real root, two comples roots

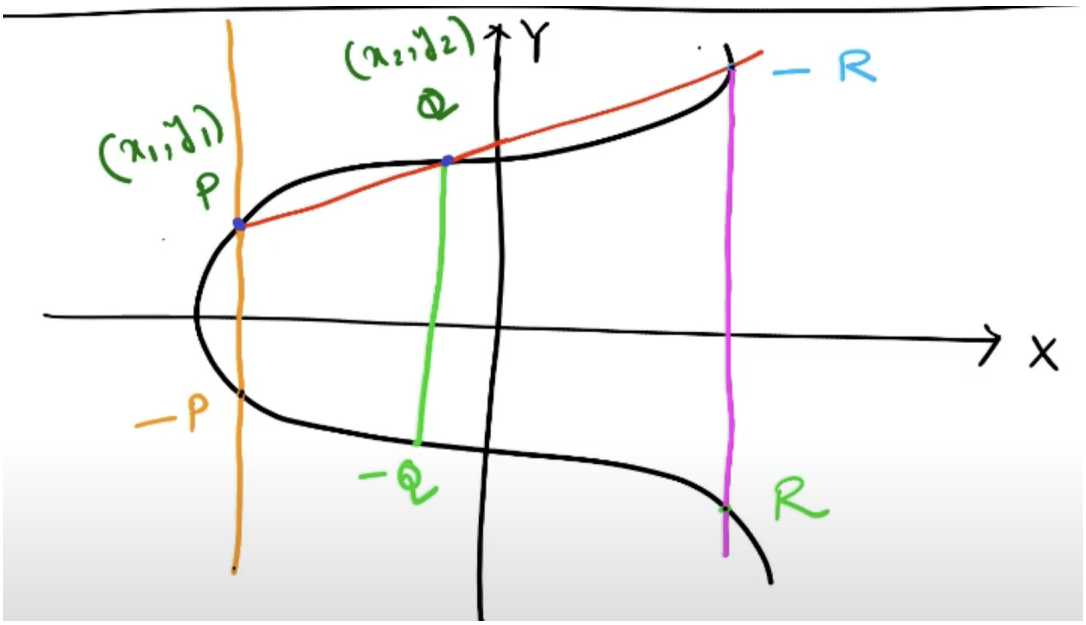
Also, Eqn $(\star)$ have all three roots distinct when $4a^3 + 27b^2 \neq 0$

Curve when all roots are real:



Curve when only one root is real:

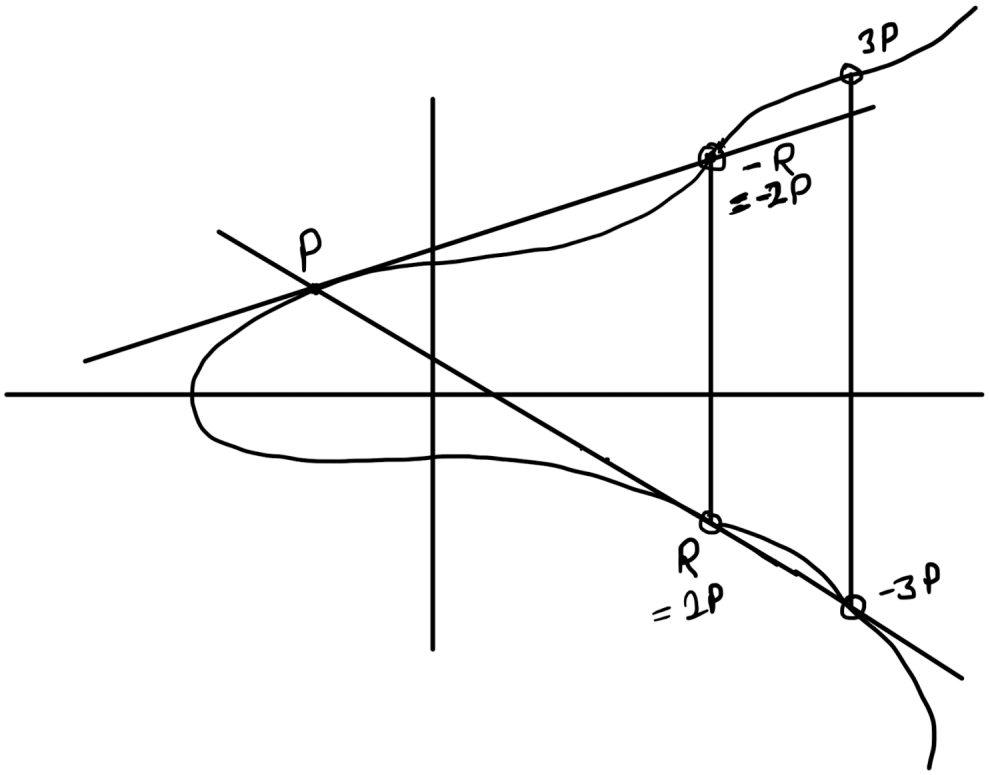




For any two points P and Q on the curve:
$P \boxplus Q = R$ (binary operation)
where, R is the mirror image through X-axis of the point at which line joining P and Q is intersecting the curve again.

1. Θ : point at infinity (treated as identity element)
2. $P \boxplus (-P) = \Theta$ (inverse existing)
3. $P \boxplus \Theta = P$ (identity element property)
4. $(P \boxplus Q) \boxplus R = P \boxplus (Q \boxplus R)$ (associativity)

From the above properties we can conclude that this box plus operator and set of points on elliptic curve are forming a Group.



$P \boxplus P = R \implies 2P = R$
(line P and P is tangent to curve at P)
$3P = 2P \boxplus P$
$nP = (n-1)P \boxplus P$

5 Mathematical Formulae for EC (Elliptic Curve) Cryptography

$y^3 = x^3 + ax + b$
$4a^3 + 27b^2 \neq 0$

5.1 Case 1

P and Q are completely different points.
P: (x_1, y_1) Q: (x_2, y_2)
Let's say eqn of line joining P and Q is :
$y = mx + c$

$m = \frac{y_2 - y_1}{x_2 - x_1}$

$y_1 = mx_1 + c$
$c = y_1 - mx_1$
Similarly, $c = y_2 - mx_2$
Solving line curve:
$(mx + c)^2 = x^3 + ax + b$
On re-arranging:

$x^3 - m^2x^2 + (a - 2mc)x = 0$
Sum of roots, $x_1 + x_2 + x_3 = m^2$
$\implies x_3 = m^2 - x_1 - x_2$

$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{y_3 - y_1}{x_3 - x_1}$

$\implies y_3 = y_1 + m(x_3 - x_1)$
R: $(x_3, -y_3)$

Problem:
P: (x_1, y_1) Q: (x_2, y_2)
P and Q are completely differnt.
R $(x_3, -y_3) = ?$ (unknown)

Solution:
$m = \frac{y2 - y_1}{x_2 - x_1}$

$x_3 = m^2 - x_1 - x_2$
$y_3 = y_1 + m(x_3 - x_1)$

5.2 Case 2

P: (x_1, y_1) Q: (x_2, y_2)
$x_1 = x_2$ $y_1 = -y_2$

Solution: $P \boxplus Q = \Theta$

5.3 Case 3

P: (x_1, y_1) Q: (x_2, y_2)
$x_1 = x_2$ $y_1 = y_2$ (tangent case)

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

$\frac{dy}{dx} = \frac{3x^2 + a}{2y}$

$m = \frac{dy}{dx}|_{(x_1, y_1)} = \frac{3x_1^2 + a}{2y_1}$

Problem:
P: (x_1, y_1) Q: (x_2, y_2)
$x_1 = x_2$ $y_1 = y_2$
R $(x_3, -y_3)$ = ? (unknown)

Solution:
$m = \frac{3x_1^2 + a}{2y_1}$

$x_3 = m^2 - x_1 - x_2$
$y_3 = y_1 + m(x_3 - x_1)$

6 Taking EC into Discrete World

We will be considering the curve in $\mathbb{Z}_p \times \mathbb{Z}_p$, where p is a prime number.
$y^2 = x^3 + ax + b$
$(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ and a,b $\in \mathbb{Z}_p$

$4a^3 + 27b^2 \neq 0 \ mod \ p$

Case 1 solution:

$m = (y_2 - y_1) \times (x_2 - x_1)^{-1} \ mod \ p$
$x_3 = m^2 - x_1 - x_2 \in \mathbb{Z}_p$
$y_3 = y_1 + m(x_3 - x_1) \in \mathbb{Z}_p$

Other cases solutions are in similar fashion.

7 Elliptic Curve Diffie Hellman (ECDH) Key Exchange Algorithm

curve E and point P are public.

Alice		Bob
a		b
aP		bP
	$\xrightarrow{\text{aP}}$	
	$\xleftarrow{\text{bP}}$	
a (bP)		b(aP)

Shared secret key: abP
a : Secret key of Alice
aP : Public key of Alice
b : Secret key of Bob
bP : Public key of Bob

Security of ECDH relies on the fact that finding x from xP and P is computationally difficult. This hard problem is known as Discrete log problem on EC.

8 ECDSA (Signature Algo using EC)

(E, P) : Public keys
Secret key: a
Public key: aP

We have a **base point** G on the curve. There exist a large prime number n such that
$n.G = \Theta \implies$ n is order.
$(n-1)G \boxplus G = nG$

Alice:
d_A (a positive integer) : Secret key of Alice.
$Q_A = d_A G$: Public key
m : message

1. e = hash(m)
2. Z : L_n left-most bits of e when L_n is the bit length of n.
3. K : randomly taken from 1 to n-1
4. $(x_1, y_1) = K.G$
5. $r = x_1 \ mod \ n$
   if r=0 then go to step 3.
6. $S = K^{-1}[Z + r.d_A] \ mod \ n$
   if S = 0 then go to step 3.
7. Signature (r,S) on message m.

Bob (Verification):

- Q_A should not be equal to Θ
  (Because if Q_A become identity then we know the secret key d_A = n, but secret key is not supposed to be revealed)
- Q_A should lie on the EC
- $n \times Q_A$ should give Θ.
  $n \times Q_A = n.(d_A.G)$
  $= d_A(n.G)$
  $= \Theta$

1. verify $r, S \in$ [1, n-1]
2. e = hash(m)

3. Z: L_n leftmost bits of e

4. $u_1 = Z.S^{-1}\ mod\ n$
   $u_2 = r.S^{-1}\ mod\ n$

5. $(x_2, y_2) = u_1G + u_2Q_A$
   if $(x_2, y_2) = \Theta$, then signature is invalid.

6. If $r \equiv x_2\ mod\ n$ then signature is valid otherwise invalid.

$C = u_1G + u_2Q_A$
$= u_1G + u_2d_AG$
$= (u_1 + u_2d_A)G$
$= (Z.S^{-1} + r.S^{-1}d_A)G$
$= (Z + r.d_A)S^{-1}G$
$= (Z + r.d_A)(K^{-1}(Z + r.d_A))^{-1}G$
$= (Z + r.d_A)(Z + r.d_A)^{-1}KG$
$= K.G = (x_1, y_1)$