Cryptography II

ECS 153 Spring Quarter 2021

Module 14

Public Key Cryptography

- Two keys
  - *Private key* known only to individual
  - *Public key* available to anyone
    - Public key, private key inverses
- Idea
  - Confidentiality: encipher using public key, decipher using private key
  - Integrity/authentication: encipher using private key, decipher using public one

Requirements

1. It must be computationally easy to encipher or decipher a message given the appropriate key
2. It must be computationally infeasible to derive the private key from the public key
3. It must be computationally infeasible to determine the private key from a chosen plaintext attack

El Gamal Cryptosystem

- Based on discrete logarithm problem
  - Given integers *n, g*, and *b* with $0 \le a < n$ and $0 \le b < n$; then find an integer *k* such that $0 \le k < n$ and $a = g^k \bmod n$
  - Choose *n* to be a prime *p*
  - Solutions known for small *p*
  - Solutions computationally infeasible as *p* grows large

Algorithm

- Choose prime p with $p - 1$ having a large factor
- Choose generator g such that $1 < g < p$
- Choose k_{priv} such that $1 < k_{priv} < p - 1$
- Set $y = g^{k_{priv}} \bmod p$
- Then public key $k_{pub} = (p, g, y)$ and private key is k_{priv}

Example

- Alice: p = 262643; g = 9563, k_{priv} = 3632
  - 262643 = 2 x 131321, also prime
- Alice's public key k_{pub} = (262643, 9563, 27459)
  - As $y = g^{k_{priv}} \bmod p = 9563^{3632} \bmod 262643 = 27459$

Enciphering and Deciphering

Encipher message m:

- Choose random integer k relatively prime to $p - 1$
- Compute $c_1 = g^k \bmod p$; $c_2 = my^k \bmod p$
- Ciphertext is $c = (c_1, c_2)$

Decipher ciphertext (c_1, c_2)

- Compute $m = c_2 c_1^{-k_{priv}} \bmod p$
- Message is m

Example Encryption

- Bob wants to send Alice PUPPIESARESMALL
- Message to send: 152015 150804 180017 041812 001111
- First block: choose k = 5
  - $c_{1,1} = 9563^5 \bmod 262643 = 15653$
  - $c_{1,2} = (152015)27459^5 \bmod 262643 = 923$
- Next block: choose k = 3230
  - $c_{2,1} = 9563^{3230} \bmod 262643 = 46495$
  - $c_{2,2} = (150804)27459^{3230} \bmod 262643 = 109351$
- Continuing, enciphered message is (15653,923), (46495,109351), (176489,208811), (88247,144749), (152432,5198)

Example Decryption

Alice receives (15653,923), (46495,109351), (176489,208811), (88247,144749), (152432,5198)

- First block: $(923)15653^{-3632} \bmod 262643 = 152015$
- Second block: $(109351)46495^{-3632} \bmod 262643 = 150804$
- Third block: $(208811)176489^{-3632} \bmod 262643 = 180017$
- Fourth block: $(144749)\ 88247^{-3632} \bmod 262643 = 41812$
- Fifth block: $(5198)\ 152432^{-3632} \bmod 262643 = 1111$

So the message is 152015 150804 180017 041812 001111

- Which translates to “PUP PIE SAR ESM ALL” or PUPPIESARESMALL

Notes

- Same letter enciphered twice produces two different ciphertexts
  - Defeats replay attacks
- If the integer k is used twice, and an attacker has plaintext for one of those messages, deciphering the other is easy
- c_2 linear function of m, so forgery possible
  - m message, (c_1, c_2) ciphertext; then (c_1, nc_2) is ciphertext corresponding to message nm

RSA

- First described publicly in 1978
  - Unknown at the time: Clifford Cocks developed a similar cryptosystem in 1973, but it was classified until recently
- Exponentiation cipher
- Relies on the difficulty of determining the number of numbers relatively prime to a large integer *n*

Background

- Totient function ϕ(n)
  - Number of positive integers less than *n* and relatively prime to *n*
    - *Relatively prime* means with no factors in common with *n*
- Example: ϕ(10) = 4
  - 1, 3, 7, 9 are relatively prime to 10
- Example: ϕ(21) = 12
  - 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20 are relatively prime to 21

Algorithm

- Choose two large prime numbers p, q
  - Let $n = pq$; then $\phi(n) = (p-1)(q-1)$
  - Choose $e < n$ such that e is relatively prime to $\phi(n)$.
  - Compute d such that $ed \bmod \phi(n) = 1$
- Public key: (e, n); private key: d
- Encipher: $c = m^e \bmod n$
- Decipher: $m = c^d \bmod n$

Example: Confidentiality

- Take p = 181, q = 1451, so n = 262631 and $\phi(n)$ = 261000
- Alice chooses e = 154993, making d = 95857
- Bob wants to send Alice secret message PUPPIESARESMALL (152015 150804 180017 041812 001111); encipher using public key
  - 152015^{154993} mod 262631 = 220160
  - 150804^{154993} mod 262631 = 135824
  - 180017^{154993} mod 262631 = 252355
  - 041812^{154993} mod 262631 = 245799
  - 001111_{154993} mod 262631 = 070707
- Bob sends 220160 135824 252355 245799 070707
- Alice uses her private key to decipher it

Example: Authentication/Integrity

- Alice wants to send Bob the message PUPPIESARESMALL in such a way that Bob knows it comes from her and nothing was changed during the transmission
  - Same public, private keys as before
- Encipher using private key:
  - $152015^{95857} \bmod 262631 = 072798$
  - $150804^{95857} \bmod 262631 = 259757$
  - $180017^{95857} \bmod 262631 = 256449$
  - $041812^{95857} \bmod 262631 = 089234$
  - $001111^{95857} \bmod 262631 = 037974$
- Alice sends 072798 259757 256449 089234 037974
- Bob receives, uses Alice's public key to decipher it

Example: Both (Sending)

- Same n as for Alice; Bob chooses e = 45593, making d = 235457
- Alice wants to send PUPPIESARESMALL (152015 150804 180017 041812 001111) confidentially and authenticated
- Encipher:
  - $(152015^{95857} \bmod 262631)^{45593} \bmod 262631 = 249123$
  - $(150804^{95857} \bmod 262631)^{45593} \bmod 262631 = 166008$
  - $(180017^{95857} \bmod 262631)^{45593} \bmod 262631 = 146608$
  - $(041812^{95857} \bmod 262631)^{45593} \bmod 262631 = 092311$
  - $(001111^{95857} \bmod 262631)^{45593} \bmod 262631 = 096768$
- So Alice sends 249123 166008 146608 092311 096768

Example: Both (Receiving)

- Bob receives 249123 166008 146608 092311 096768
- Decipher:
  - $(249123^{235457} \bmod 262631)^{154993} \bmod 262631 = 152012$
  - $(166008^{235457} \bmod 262631)^{154993} \bmod 262631 = 150804$
  - $(146608^{235457} \bmod 262631)^{154993} \bmod 262631 = 180017$
  - $(092311^{235457} \bmod 262631)^{154993} \bmod 262631 = 041812$
  - $(096768^{235457} \bmod 262631)^{154993} \bmod 262631 = 001111$
- So Alice sent him 152015 150804 180017 041812 001111
  - Which translates to PUP PIE SAR ESM ALL or PUPPIESARESMALL

Security Services

- Confidentiality
  - Only the owner of the private key knows it, so text enciphered with public key cannot be read by anyone except the owner of the private key
- Authentication
  - Only the owner of the private key knows it, so text enciphered with private key must have been generated by the owner

More Security Services

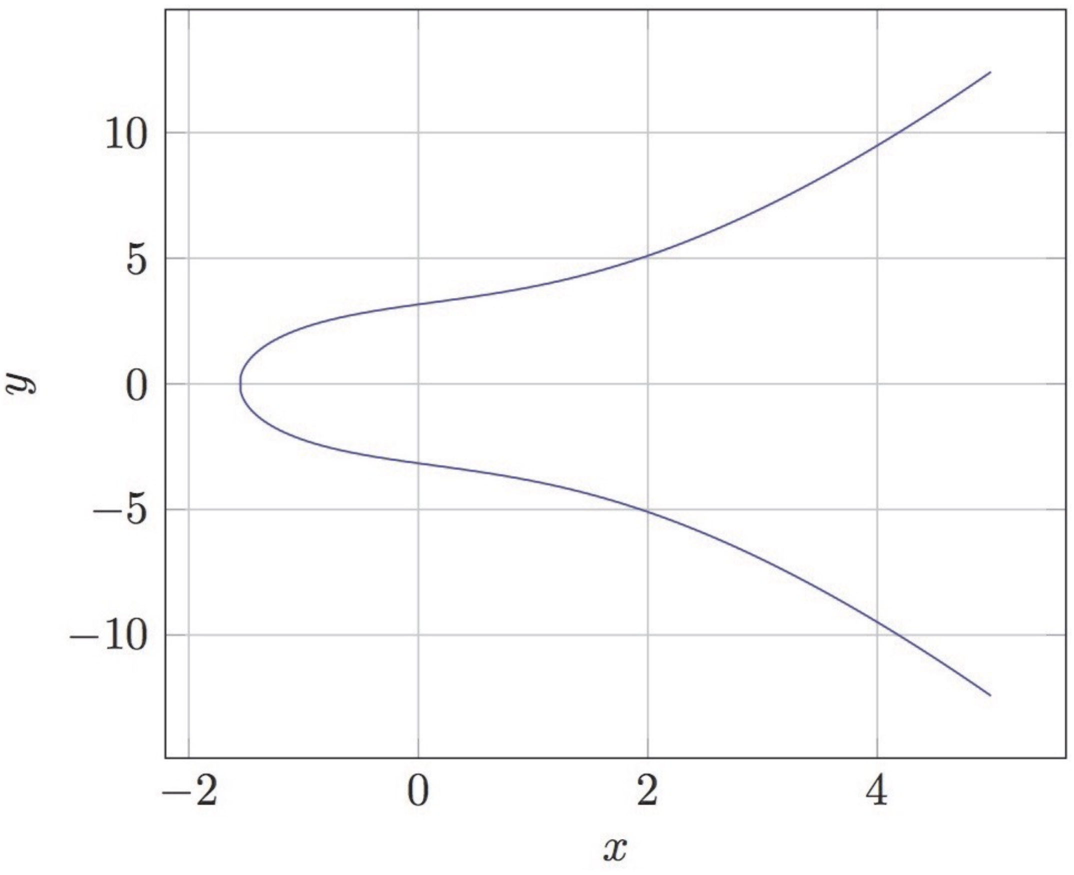
- Integrity
  - Enciphered letters cannot be changed undetectably without knowing private key
- Non-Repudiation
  - Message enciphered with private key came from someone who knew it

Warnings

- Encipher message in blocks considerably larger than the examples here
  - If only characters per block, RSA can be broken using statistical attacks (just like symmetric cryptosystems)
- Attacker cannot alter letters, but can rearrange them and alter message meaning
  - Example: reverse enciphered message of text ON to get NO

Elliptic Curve Ciphers

- Miller and Koblitz proposed this
- *Elliptic curve* is a curve of the form $y^2 = x^3 + ax + b$
  - Curve $y^2 = x^3 + 4x + 10$ plotted at right
- Can be applied to any cryptosystem depending on discrete log problem
- Advantage: keys shorter than other forms of public key cryptosystems, so computation time shorter



Basics

- Take 2 points on the elliptic curve P_1, P_2
  - If $P_1 \neq P_2$, draw line through them
  - If $P_1 = P_2$, draw a tangent to curve there
- If line intersects curve at $P_3 = (x_3, y_3)$
  - Take the sum of P_1, P_2 to be P4 = $(x_3, -y_3)$
- Otherwise, line is vertical, so take $P_1 = (x, y)$; treat ∞ as another point of intersection; third point of intersection is $P_2 = (x, -y)$
  - Given above definition of addition, $P_1 + \infty = (x, y) = P_1$
  - So ∞ is additive identity

The Math

- $P_1 = (x_1, y_1)$; $P_2 = (x_2, y_2)$
- Then if $P_1 \neq P_2$, $m = (y_2 - y_1) / (x_2 - x_1)$
- Otherwise, $m = (3x_1^2 + a) / y_1$
- Next, $P_3 = P_1 + P_2 = (m^2 - x_1 - x_2, m(x_1 - x_3) - y_1) = (x_3, y_3)$
- And $P_4 = (x_{4,}, y_4)$, where $x_4 = x_3$, $y_4 = -y_3$
  - P_4 defined to be sum of P_1, P_2

Basis for the Cryptosystem

- Curve: $y^2 = x^3 + ax + b \bmod p$, where $4a^3 + 27b^2 \neq 0$ and p prime
- Pick a point P and add it to itself n times; call this Q, so $Q = nP$
  - If n is large, generally very hard to compute n from P and Q
- So, elliptic curve cryptosystem has 4 parameters (a, b, p, P)
- Private key k_{priv} chosen randomly such that $k_{priv} < p$
  - In practice, choose k_{priv} to be less than number of integer points on curve
- Public key $k_{pub} = k_{priv} P$
- In what follows, $(x, y) \bmod p = (x \bmod \text{p}, y \bmod p)$

Elliptic Curve El Gamal Cryptosystem

- Choose a point P on the curve, and a private key *kpriv*
- Compute $Q = k_{priv}P$
- Public key is (P, Q, a, p)

Encipher: express message as point m on curve; choose random number k

- $c_1 = kP$; $c_2 = m + kQ$
- Ciphertext is (c_1, c_2)

Decipher:

- $m = c_2 - k_{priv}c_1$
- Message is m

Example: Encryption

- Alice, Bob agree to use the curve $y^2 = x^3 + 4x + 14 \bmod 2503$ and the point $P = (1002, 493)$
- Bob chooses private key $k_{priv,Bob} = 1847$
  - Public key $k_{pub\ ,Bob} = k_{priv,Bob}P = 1847(1002, 493) \bmod 2503 = (460, 2083)$
- Alice wants to send Bob message $m = (18, 1394)$
  - She chooses random $k = 717$
  - $c_1 = kP = 717(1002, 493) \bmod 2503 = (2134, 419)$
  - $c_2 = m + k\, k_{pub\ ,Bob} = (18, 1394) + 717(460, 2083) \bmod 2503 = (221, 1253)$

so she sends Bob c_1 and c_2

Example: Decryption

- From last slide, Alice, Bob agree to use the curve $y^2 = x^3 + 4x + 14$ mod 2503 and the point $P = (1002, 493)$
  - Bob's private key $k_{priv,Bob} = 1847$
  - Bob's public key $k_{pub,Bob}$ (460, 2083)
- To decrypt $c_1 = (2134, 419)$, $c_2 = (221, 1253)$, Bob computes:
  - $k_{priv,Bob}c_1 = 1847(2134, 419) \bmod 2503 = (652, 1943)$
  - $m = c_2 - c_1 = (221, 1253) - (652, 1943) \bmod 2503 = (18, 1394)$

obtaining the message Alice sent

Selection of Elliptic Curves

- For elliptic curves for cryptography, selection of parameters critical
  - Example: b = 0, p mod 4 = 3 makes the underlying discrete log problem significantly easier to solve
  - Example: so does a = 0, p mod 3 = 2
- Several such curves are recommended:
  - U.S. NIST: P-192, P-224, P-256, P-384, P-521 using a prime modulus and a binary field of degree 163, 233, 283 409, 571
  - Certicom: same, but degree 239 binary field instead of degree 233 binary field
  - Others: Curve1174, Curve25519

Cryptographic Checksums

- Mathematical function to generate a set of *k* bits from a set of *n* bits (where $k \leq n$).
  - *k* is smaller then *n* except in unusual circumstances
- Example: ASCII parity bit
  - ASCII has 7 bits; 8th bit is "parity"
  - Even parity: even number of 1 bits
  - Odd parity: odd number of 1 bits

Example Use

- Bob receives “10111101” as bits.
  - Sender is using even parity; 6 1 bits, so character was received correctly
    - Note: could be garbled, but 2 bits would need to have been changed to preserve parity
  - Sender is using odd parity; even number of 1 bits, so character was not received correctly

Definition

- Cryptographic checksum $h: A \rightarrow B$:
  1. For any $x \in A$, $h(x)$ is easy to compute
  2. For any $y \in B$, it is computationally infeasible to find $x \in A$ such that $h(x) = y$
  3. It is computationally infeasible to find two inputs $x, x' \in A$ such that $x \neq x'$ and $h(x) = h(x')$
     – Alternate form (stronger): Given any $x \in A$, it is computationally infeasible to find a different $x' \in A$ such that $h(x) = h(x')$.

Collisions

- If $x \neq x'$ and $h(x) = h(x')$, x and x' are a *collision*
  - Pigeonhole principle: if there are n containers for n+1 objects, then at least one container will have at least 2 objects in it.
  - Application: if there are 32 files and 8 possible cryptographic checksum values, at least one value corresponds to at least 4 files

Keys

- Keyed cryptographic checksum: requires cryptographic key
  - AES in chaining mode: encipher message, use last *n* bits. Requires a key to encipher, so it is a keyed cryptographic checksum.
- Keyless cryptographic checksum: requires no cryptographic key
  - SHA-512, SHA-3 are examples; older ones include MD4, MD5, RIPEM, SHA-0, and SHA-1 (methods for constructing collisions are known for these)

HMAC

- Make keyed cryptographic checksums from keyless cryptographic checksums
- h keyless cryptographic checksum function that takes data in blocks of b bytes and outputs blocks of l bytes. k' is cryptographic key of length b bytes
  - If short, pad with 0 bytes; if long, hash to length b
- *ipad* is 00110110 repeated b times
- *opad* is 01011100 repeated b times
- HMAC-$h(k, m) = h(k' \oplus opad \,||\, h(k' \oplus ipad \,||\, m))$
  - $\oplus$ exclusive or, $||$ concatenation

Strength of HMAC-*h*

- Depends on the strength of the hash function *h*
- Attacks on HMAC-MD4, HMAC-MD5, HMAC-SHA-0, and HMAC-SHA-1 recover partial or full keys
    - Note all of MD4, MD5, SHA-0, and SHA-1 have been broken

Digital Signature

- Construct that authenticates origin, contents of message in a manner provable to a disinterested third party (a “judge”)
- Sender cannot deny having sent message (service is “nonrepudiation”)
  - Limited to *technical* proofs
    - Inability to deny one’s cryptographic key was used to sign
  - One could claim the cryptographic key was stolen or compromised
    - Legal proofs, *etc.,* probably required; not dealt with here

Common Error

- Symmetric: Alice, Bob share key *k*
  - Alice sends *m* || { *m* } *k* to Bob
  - { *m* } *k* means *m* enciphered with key *k*, || means concatenation

  Claim: This is a digital signature

WRONG

This is not a digital signature

- Why? Third party cannot determine whether Alice or Bob generated message

Classical Digital Signatures

- Require trusted third party
  - Alice, Bob each share keys with trusted party Cathy
- To resolve dispute, judge gets $\{ m \} k_{Alice}$, $\{ m \} k_{Bob}$, and has Cathy decipher them; if messages matched, contract was signed

$$\text{Alice} \xrightarrow{\{ m \}k_{Alice}} \text{Bob}$$

$$\text{Cathy} \xleftarrow{\{ m \}k_{Alice}} \text{Bob}$$

$$\text{Cathy} \xrightarrow{\{ m \}k_{Bob}} \text{Bob}$$

Public Key Digital Signatures

- Basically, Alice enciphers the message, or its cryptographic hash, with her private key
- In case of dispute or question of origin or whether changes have been made, a judge can use Alice's public key to verify the message came from Alice and has not been changed since being signed

RSA Digital Signatures

- Alice's keys are (e_{Alice},n_{Alice}) (public key), d_{Alice} (private key)
  - In what follows, we use e_{Alice} to represent the public key
- Alice sends Bob

$$m \;||\; \{\, m \,\}\, e_{Alice}$$

- In case of dispute, judge computes

$$\{\, \{\, m \,\}\, e_{Alice} \,\}\, d_{Alice}$$

- and if it is m, Alice signed message
  - She's the only one who knows d_{Alice}!

RSA Digital Signatures

- Use private key to encipher message
  - Protocol for use is *critical*
- Key points:
  - Never sign random documents, and when signing, always sign hash and never document
  - Don’t just encipher message and then sign, or vice versa
    - Changing public key and private key can cause problems
    - Messages can be forwarded, so third party cannot tell if original sender sent it to her

Attack #1

- Example: Alice, Bob communicating
  - n_A = 262631, e_A = 154993, d_A = 95857
  - n_B = 288329, e_B = 22579, d_B = 138091
- Alice asks Bob to sign 225536 so she can verify she has the right public key:
  - $c = m^{d_B} \bmod n_B = 225536^{138091} \bmod 288329 = 271316$
- Now she asks Bob to sign the statement AYE (002404):
  - $c = m^{d_B} \bmod n_B = 002404^{138091} \bmod 288329 = 182665$

Attack #1

- Alice computes:
  - new message NAY (130024) by (002404)(225536) mod 288329 = 130024
  - corresponding signature (271316)(182665) mod 288329 = 218646
- Alice now claims Bob signed NAY (130024), and as proof supplies signature 218646
- Judge computes $c^{e_B} \bmod n_B = 218646^{22579} \bmod 288329 = 130024$
  - Signature validated; Bob is toast

Preventing Attack #1

- Do not sign random messages
  - This would prevent Alice from getting the first message
- When signing, always sign the cryptographic hash of a message, not the message itself

Attack #2: Bob's Revenge

- Bob, Alice agree to sign contract LUR (112017)
  - But Bob really wants her to sign contract EWM (042212), but knows she won't
- Alice enciphers, then signs:
  - $(m^{e_B} \bmod n_A)^{d_A} \bmod n_A = (112017^{22579} \bmod 288329)^{95857} \bmod 262631 = 42390$
- Bob now changes his public key
  - Computes r such that $042212^r \bmod 288329 = 112017$; one such $r = 9175$
  - Computes $re_B \bmod \phi(n_B) = (9175)(22579) \bmod 287184 = 102661$
  - Replace public key with (102661,288329), private key with 161245
- Bob claims contract was EWM
- Judge computes:
  - $(42390^{154993} \bmod 262631)^{161245} \bmod 288329 = 042212$, which is EWM
  - Verified; now Alice is toast

Preventing Attack #2

- Obvious thought: instead of encrypting message and then signing it, sign the message and then encrypt it
  - May not work due to surreptitious forwarding attack
  - Idea: Alice sends Cathy an encrypted signed message; Cathy deciphers it, re-enciphers it with Bob's public key, and then sends message and signature to Bob – now Bob thinks the message came from Alice (right) and was intended for him (wrong)
- Several ways to solve this:
  - Put sender and recipient in the message; changing recipient invalidates signature
  - Sign message, encrypt it, then sign the result

El Gamal Digital Signature

- Relies on discrete log problem
  - Choose p prime, g, $d < p$; compute $y = g^d \bmod p$
- Public key: (y, g, p); private key: d
- To sign contract m:
  - Choose k relatively prime to p–1, and not yet used
  - Compute $a = g^k \bmod p$
  - Find b such that $m = (da + kb) \bmod p–1$
  - Signature is (a, b)
- To validate, check that
  - $y^a a^b \bmod p = g^m \bmod p$

Example

- Alice chooses p = 262643, g = 9563, d = 3632, giving y = 274598
- Alice wants to send Bob signed contract PUP (152015)
  - Chooses k = 601 (relatively prime to 262642)
  - This gives $a = g^k \bmod p = 9563^{601} \bmod 29 = 202897$
  - Then solving $152015 = (3632 \times 202897 + 601b) \bmod 262642$ gives b = 225835
  - Alice sends Bob message m = 152015 and signature (a,b) = (202897, 225835)
- Bob verifies signature: $g^m \bmod p = 9563^{152015} \bmod 262643 = 157499$ and $y^a a^b \bmod p = 27459^{202897} 202897^{225835} \bmod 262643 = 157499$
  - They match, so Alice signed

Attack

- Eve learns k, corresponding message m, and signature (a, b)
  - Extended Euclidean Algorithm gives d, the private key
- Example from above: Eve learned Alice signed last message with $k = 5$

$$m = (da + kb) \bmod p{-}1 \Rightarrow 152015 = (202897d + 601 \times 225835) \bmod 262642$$

giving Alice's private key $d = 3632$

El Gamal Digital Signature Using Elliptic Curve Cryptography

- As before, curve is $y^2 = x^3 + ax + b \bmod p$ with n integer points on it
  - Choose a point P on the curve
  - Choose private key *kpriv;* compute $Q = k_{priv}P$, and the corresponding public key is (P, Q, a, b)
- To digitally sign, choose random integer k with $1 \leq k < n$
  - Compute $R = kP$ and $s = k^{-1}(m - k_{priv}x) \bmod n$, where x is first component of R
  - Digital signature is (m, R, s)
- To validate, recipient computes:
  - $V_1 = xQ + sR$
  - $V_2 = mP$
  - If $V_1 = V_2$, signature valid

Example

- Alice, Bob use elliptic curve $y^2 = x^3 + 4x + 14 \bmod 2503$, point $P = (1002, 493)$
  - Curve has $n = 2477$ integer points on it
  - Bob chooses $k_{priv,Bob} = 1874$, so $Q = 1847(1002, 493) \bmod 2503 = (460, 2083)$
- Bob digitally signs message $m = 379$
  - Chooses $k = 877$
  - Computes $R = kP = 877(1002,493) = (1014, 788)$
  - Computes $s = k^{-1}(m - k_{priv,Bob}x) \bmod n = 877^{-1}(379 - 1847 \times 1014) \bmod 2477 = 2367$
  - Sends Alice (379, (1014, 788), 2367)

Example

- To validate signature, Alice computes:
  - $V_1 = xQ + sR = 1014(460,2083) + 2367(1014, 788) = (535, 1015)$
  - $V_2 = mP = 379(1002,493) = (535, 1015)$
- As $V_1 = V_2$, the signature is validated