Lecture 11, April 22, 2026

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*

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

El Gamal Cryptosystem

- Based on discrete logarithm problem
  - Given integers *n, g*, and *b* with $0 \leq a < n$ and $0 \leq b < n$; then find an integer *k* such that $0 \leq k < n$ 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

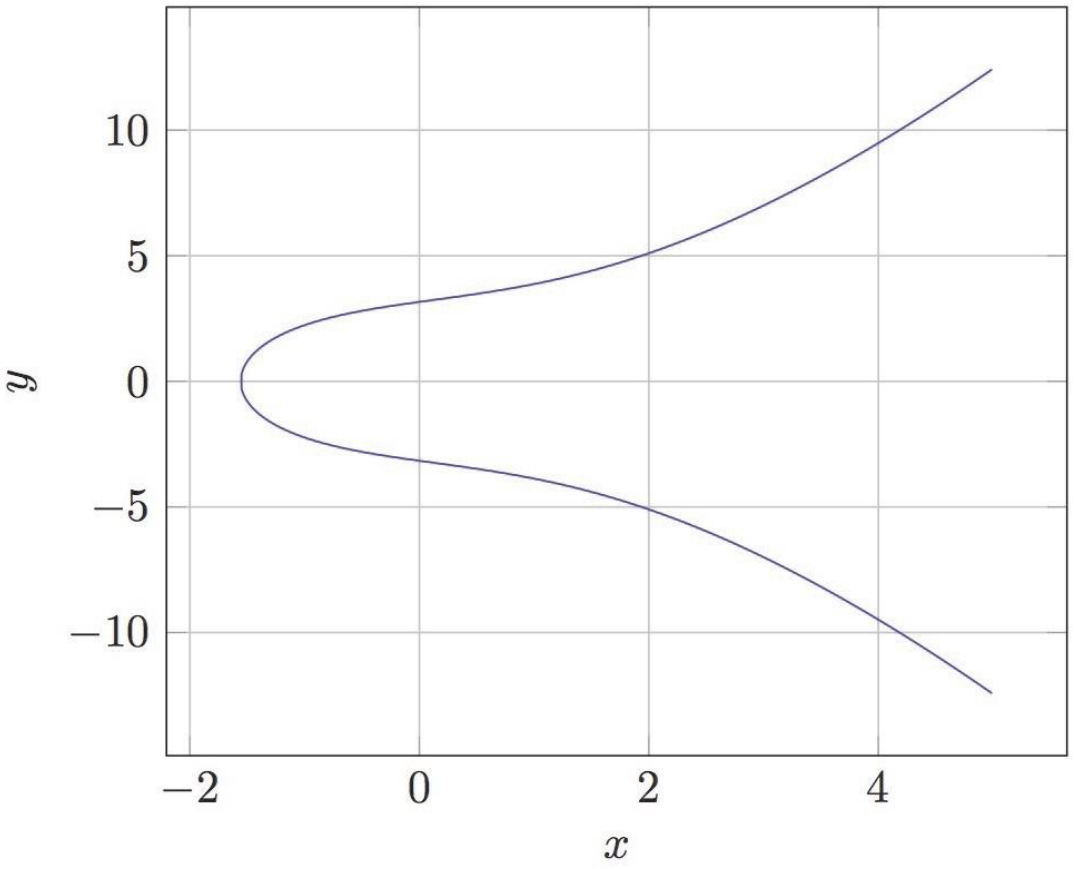
- Which translates to “PUP PIE SAR ESM ALL” or PUPPIESARESMALL

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

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, $(x, y) \bmod p = (x \bmod p, y \bmod p)$

Elliptic Curve El Gamal Cryptosystem

- Choose a point P on the curve, and a private key k_{priv}
- 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 $y^2 = x^3 + 4x + 14 \bmod 2503$ 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)