SJSU SAN JOSÉ STATE UNIVERSITY

Lesson 5 – Public Key Crypto 1

Yan Chen

CS166 Fall 2024

- Encryption modes: ways to encrypt multiple blocks
  - These modes are for all block ciphers in general

	ECB	CBC	CTR
Full Name	Electronic Codebook	Cipher Block Chaining	Counter
Each Block	Independently	Dependently	Independently
Analogy	Codebook without additive	Codebook with additive	Stream cipher
Encryption	$C_n = E(P_n, K)$	$C_0 = E(IV \oplus P_0, K)$ $C_n = E(C_{n-1} \oplus P_n, K)$	$C_n = P_n \oplus E(IV + n, K)$
Decryption	$P_n = D(C_n, K)$	$P_0 = IV \oplus D(C_0, K)$ $P_n = C_{n-1} \oplus D(C_n, K)$	$P_n = C_n \oplus E(IV + n, K)$
Notes	Same P → Same C	Auto-recover from errors when decrypting	$E(IV + n, K)$ is used as a keystream

- Encryption only provides confidentiality (hide info)
- Use Message Authentication Code (MAC) for integrity
  - ➢ The calculation is the same as CBC encryption
  - ➢ But only save the last cipher block, and call it “MAC”
  - ➢ Alice send Bob the IV, the P’s and the MAC
  - ➢ Bob do the same computation to verify
  - ➢ Any change in P will result in a wrong MAC (error propagates), so using MAC can detect unauthorized writing of information
- Confidentiality and integrity with same work as one encryption is a research topic

- Symmetric key: use same key to encrypt and decrypt
  - Encryption algorithm needs to be invertible
  - Usually, $\oplus$ does the magic: $C = P \oplus K$ then $P = C \oplus K$
- Symmetric key crypto mainly used for confidentiality
  - Transmitting data over insecure channel
  - Secure storage on insecure media
- MAC used for integrity
- Other usages to be covered
  - Anything you can do with a hash function (~Lesson 8)
  - Authentication protocols (~Lesson 19 or 20)

- Distributing the keys for symmetric ciphers is a problem
  - ➢ Alice and Bob needs to exchange the key secretly first...
- Public key crypto eliminate the key distributing problem
  - ➢ Suppose Alice wants to send a secret message to Bob
  - ➢ Alice encrypt the message using Bob’s public key

    - everyone can do this since everyone knows Bob’s public key
  - ➢ Bob decrypt the message using his private key

    - only Bob can do this since only he knows his private key
  - ➢ That is, everyone can send Bob a secret message that only Bob can read, without exchanging a key!

- Public key crypto for encryption (this Lesson)
  - $\{M\}_{Alice}$ = ciphertext after encrypting M by Alice's public key
  - Everyone can do the encryption (i.e., compute $\{M\}_{Alice}$)
  - Only Alice can decrypt $\{M\}_{Alice}$ to get M using her private key
- Public key crypto for key exchange (Lesson 6)
- Public key crypto for signature (Lesson 7)
  - $[M]_{Bob}$ = "signature" after encrypting M by Bob's private key
  - Only Bob can compute $[M]_{Bob}$
  - Everyone can decrypt $[M]_{Bob}$ to get M with Bob's public key
  - If $\{[M]_{Bob}\}_{Bob}$ = M, Bob's signature is verified

- The magic is based on "trap door, one way function"
  - Easy to compute in one direction

    $Y = f(x)$ is easy (using x to get Y is easy)
  - Hard to compute in other direction

    $x = f^{-1}(Y)$ is hard (using Y to get x is hard)
  - Like a trap door: easy to get in, hard to get out!
  - Example: given p & q, calculate N = p * q is easy,

    but given N, hard to find p and q such that N = p * q!
- In general, public-key ciphers are more mathematical than symmetric key ciphers (also harder to design!)

- Prime: an integer (> 1) only is divided by 1 and itself.
  - Examples: 2, 3, 5, 7, 11, 13, 17, 19, 23, ...
  - A non-prime number is called composite
- Suppose x and y are integers, if z = x / y is an integer, then y (and z) is called a divisor (factor) of x.
  - Example: divisors for 24 are 1, 2, 3, 4, 6, 8, 12, and 24
- Greatest Common Divisor (GCD) of x and y: the largest integer d such as d is a divisor of both x and y
  - Example: gcd(24, 32) = gcd(32, 24) = 8
  - If y is a divisor of x, $\gcd(x, y) = y$ (e.g., gcd(12, 3) = 3)

- Two integers x and y are relatively prime if $\gcd(x, y) = 1$.
  - That is, x and y doesn't have other common divisors
  - x and y does NOT have to be primes
  - Example: 9 and 16 are relatively prime since gcd(9, 16) = 1, but both 9 and 16 are not primes
- Totient function $\varphi(n)$: the number of numbers less than n that are relatively prime to n
  - Example: $\varphi(9) = 6$ since 9 is relatively prime to 1,2,4,5,7,8
  - $\varphi(p) = p - 1$ if p is prime
  - $\varphi(pq) = \varphi(p) * \varphi(q) = (p - 1)(q - 1)$ if p and q prime

- Euclidean algorithm is used to compute gcd(x, y)
  - Suppose x > y

```
gcd(x, y)
        if y = 0, return x
        else return gcd(y, x mod y)
```

- Extended Euclidean algorithm: based on Bezout's Theorem: $\gcd(x, y) = ax + by$
  - a and b are called Bezout's coefficients (also integers)
  - Not only computes $\gcd(x, y)$, but also computes a and b!
  - If x and y are relatively prime, then $1 = ax + by$, which is used to compute the multiplicative inverses (next section)

- Extended Euclidean algorithm (continued)
  - Suppose $\gcd(x, y) = g$, that is, $g = ax + by$, then given x, y, the following algorithm calculates g, a and b

```
g_0 = x; g_1 = y;
a_0 = 1; a_1 = 0;
b_0 = 0; b_1 = 1;
while g_i > 0
        g_i = g_{i-2} mod g_{i-1};
        q_i = floor(g_{i-2} / g_{i-1});
        a_i = a_{i-2} – q_i * a_{i-1};
        b_i = b_{i-2} – q_i * b_{i-1};
end loop when g_i = 0
g = g_{i-1};
a = a_{i-2};
b = b_{i-2};
```

- Cite the source if you used online code for homework!

- x mod n (modulo): remainder of x divided by n
  - Result "circled" from 0 to n –1 ("Clock" arithmetic)
  - Examples: 14 mod 12 = 2; 20 mod 10 = 0; 33 mod 6 = 3, etc.
- Some useful formulas for modulo
  - $[(\mathrm{a \bmod n}) + (\mathrm{b \bmod n})] \bmod \mathrm{n} = (\mathrm{a} + \mathrm{b}) \bmod \mathrm{n}$
  - $[(\mathrm{a \bmod n})(\mathrm{b \bmod n})] \bmod \mathrm{n} = \mathrm{ab} \bmod \mathrm{n}$
- Congruence: $\mathrm{a} \equiv \mathrm{b} \pmod{\mathrm{n}}$ means $\mathrm{a} \bmod \mathrm{n} = \mathrm{b} \bmod \mathrm{n}$
  - n is the divisor of a – b : $\mathrm{a} - \mathrm{b} = \mathrm{kn}$ for an integer k
  - Example 1 (modular addition): $(3 + 5) \equiv 2 \pmod{6}$
  - Example 2 (modular multiplication): $(3 * 5) \equiv 3 \pmod{6}$

- Some useful properties and formulas of congruence
  - Reflexivity: $a \equiv a \pmod{n}$
  - Symmetry: $a \equiv b \pmod{n} \rightarrow b \equiv a \pmod{n}$
  - Transitivity: $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n} \rightarrow a \equiv c \pmod{n}$
  - $a + k \equiv b + k \pmod{n}$ iff $a \equiv b \pmod{n}$ for any integer k
  - If $a \equiv b \pmod{n}$ then $ka \equiv kb \pmod{n}$ for any integer k
  - If $ka \equiv kb \pmod{n}$ and k is coprime with n, then $a \equiv b \pmod{n}$
  - If $ka \equiv kb \pmod{kn}$, then $a \equiv b \pmod{n}$
  - If $a \equiv b \pmod{n}$ then $a^k \equiv b^k \pmod{n}$ for any integer $k \geq 0$
  - For more, check here

- $-x \bmod N$: additive inverse of x mod N
  - $-x \bmod N = y$ if $x + y \equiv 0 \pmod N$
  - i.e., y is the number that must be added to x to get 0 mod N
  - Example: -2 mod 6 = 4, since $2 + 4 \equiv 0 \pmod 6$
  - -3 mod 6 = ? -2 mod 7 = ? -33 mod 10 = ?
- $x^{-1} \bmod N$: multiplicative inverse of x mod N
  - x^{-1} mod N = y if $xy \equiv 1 \pmod N$
  - i.e., the number that must be multiplied by x to get 1 mod N
  - Example: 5^{-1} mod 6 = 5, since $5 * 5 \equiv 1 \pmod 6$
  - 6^{-1} mod 5 = ? 11^{-1} mod 7 = ? 2^{-1} mod 6 = ?

- $x^{-1} \bmod N$ exists only when x and n are relatively prime
  - If x and n have common divisors, then there's no y such that $xy \equiv 1 \pmod{N}$
- $x^{-1} \bmod N$ can be calculated using Extended Euclidean
  - Recall: $\gcd(x, n) = 1 = ax + bN$ if x and N are relatively prime
  - That is, $ax + bN \equiv 1 \pmod{N}$ (1)
  - Since $bN \bmod N = 0$, $ax + bN \equiv ax \pmod{N}$ (2)
  - Given (1) & (2), we get $ax \equiv 1 \pmod{N}$
  - That is, $x^{-1} \bmod N = a$, and a can be computed by Extended Euclidean algorithm if given x and N

- Fermat's little theorem: if p is a prime number, then $a^p \equiv a \pmod{p}$ for any integer a
  - ➢ If a is not a divisor of a prime p, then $a^{p-1} \equiv 1 \pmod{p}$
- Euler's theorem: if a and n are relatively prime, then $a^{\varphi(n)} \equiv 1 \pmod{n}$ where $\varphi(n)$ is the totient function of n
  - ➢ Recall: $\varphi(n)$: the number of relatively primes (< n) to n
- Chinese remainder theorem: if n_i are pairwise coprime, and $0 \leq a_i < n_i$ then $x \equiv a_i \pmod{n_i}$ for $0 \leq i \leq m$ has a unique solution for $x \bmod N$ where N is the product of n_i

- RSA: public key crypto based on modulo & primes
  - Invented by Clifford Cocks (GCHQ) and **R**ivest, **S**hamir, and **A**dleman (MIT)
  - RSA is the gold standard in public key crypto
- To generate keys:
  - Let p and q be two large prime numbers, and N = pq
  - Choose e relatively prime to $(p-1)(q-1) = \varphi(N)$
  - Find $d = e^{-1} \bmod \varphi(N)$ (i.e., $ed \equiv 1 \bmod \varphi(N)$)
  - Public key is (N, e)
  - Private key is d (p and q are also secrets!)

- Message (plaintext) M is treated as a number
  - To encrypt M, compute $C \equiv M^e \bmod N$
  - To decrypt ciphertext C, compute $M \equiv C^d \bmod N$
- Let's (informally) prove it works, that is, $M \equiv M^{ed} \bmod N$
  - Recall: $d = e^{-1} \bmod \varphi(N)$ so $ed \equiv 1 \bmod \varphi(N)$
  - That is, ed – 1 = $k\varphi(N)$ for integer k (Congruence definition)
  - Then $M^{ed} \equiv M^{(ed\text{-}1)+1} \equiv M{*}M^{(ed\text{-}1)} \equiv M{*}M^{k\varphi(N)} \pmod N$
  - Since p and q are both primes, M should be relatively prime to p * q = N, then by Euler's theorem $M^{\varphi(N)} \equiv 1 \pmod N$
  - Then $M^{ed} \equiv M{*}M^{k\varphi(N)} \equiv M{*}(M^{\varphi(N)})^k \equiv M{*}1^k \equiv M \pmod N$, QED

- The security of RSA is based on it's hard to factorize N
  - That is, find p & q that's used to get N
  - "Brute-force": try all p's from 2 to sqrt(N)
  - Once Trudy knows p & q, easy to get private key d using extended Euclidean algorithm (programming assignment 1!)
  - Factorizing N gets harder as N get bigger [O(sqrt(N))]
  - So, in real life, N is big (2048 bits at least)
- Choice of e also matters...just a little
  - Size of e doesn't matter as much as the size of N...
  - But choosing 3 as e may result in cube root attack

- 2 possible cube root attack if e = 3
  - Possibility 1: when $M^e = M^3 < N$, then $M^3 \bmod N = M^3 = C$
    That is, attacker can compute cube root of C to get M
  - Possibility 2: send same message M to 3 users using e = 3
    so $C_1 \equiv M^3 \bmod N_1$, $C_2 \equiv M^3 \bmod N_2$, and $C_3 \equiv M^3 \bmod N_3$
    Can get $C \equiv M^3 \bmod N_1N_2N_3$ by Chinese remainder theorem
    Rest is the same as possibility 1
- Padding random bits on M can prevent the attack
  - For possibility 1, make $M^3 > N$
  - For possibility 2, make M different

- Diffie-Hellman Key Exchange
- ECC

- Public key crypto
  - Public key vs. private key, usages
  - One-way trapdoor function
- Math preliminaries
  - Divisors, GCD, relatively prime, totient function
  - Euclidean algorithm, extended Euclidean algorithm
  - Congruence, modular reverses (additive & multiplicative)
  - Fermat's little theorem, Euler's theorem, Chinese remainder theorem
- RSA
  - Private key: d (p & q also private)
  - Public key: N, e
  - Cube root attack

- Calculate the following
  - $\varphi(12)$
  - $\varphi(13)$
  - $\varphi(15)$
  - $\varphi(53)$
- Calculate the following
  - -3 mod 8
  - -31 mod 5
  - -47 mod 3
  - 7^{-1} mod 6
  - 5^{-1} mod 8
  - 3^{-1} mod 6

References

- Stamp, Mark, “Information Security, Principles and Practice, 2nd ed.,” Wiley, New Jersey, USA, 2011