COMP232 Cybersecurity
Public-Key Encryption
Public-key, or asymmetric encryption

- **Public-key encryption** techniques. It is particular and most important kind of

- **Asymmetric encryption** (or asymmetric key encryption):
  - **One key** is used for encryption (usually publicly known, *public key*);
  - **Another key** is used for decryption (usually *private*, or *secret* key)
Public-key encryption

(a) Encryption
Components of public-key encryption

- Plaintext
- Encryption algorithm
- Public and private key
- Ciphertext
- Decryption algorithm
Essential steps in communications using public-key encryption

- Each user generates a pair of keys;
- Each user makes one of the keys publicly accessible (public key). The other key of the pair is kept private;
- If B wishes to send a private message to A, B encrypts the message using A’s public key;
- When A receives the message, A decrypts it using A’s private key. No other recipient can decrypt the message – nobody else knows A’s private key.
Public-key encryption

- **Advantages**
  - All keys (public and private) are generated locally;
  - No need in distribution of the keys;
  - Moreover, each user can change his own pair of public/private key at any time;

- **Disadvantages**
  - It is more computationally expensive.
Applications of Public-Key Cryptosystems

- **Encryption/decryption**: the sender encrypts a message with the recipient’s public key.

- **Digital signature (authentication)**: the sender “signs” the message with its private key; a receiver can verify the identity of the sender using sender’s public key.

- **Key exchange**: both sender and receiver cooperate to exchange a (session) key.
Authentication using public-key systems

(b) Authentication
Requirements for Public-Key Cryptography

- **Diffie and Hellman conditions**
- **“Easy part”**
  - It is computationally easy for a party B to generate a pair (public key, private key).
  - It is computationally easy for a sender A, knowing the public key of B and the message M to generate a ciphertext:
  - It is computationally easy for the receiver B to decrypt the resulting ciphertext using his private key
Requirements for Public-Key Cryptography

• “Difficult part”
  • It is computationally infeasible for anyone, knowing the public key, to determine the private key,

• Additional useful requirement (not always necessary)
  • Either of the two related keys can be used for encryption, with the other used for decryption.
Public-key cryptography and number theory

• Many public-key cryptosystems use non-trivial number theory;
• Security of most known RSA public-key cryptosystem is based on the hardness of factoring big numbers;

• We will overview basic notions of divisors, prime numbers, modular arithmetic
Divisors and prime numbers

• Divisors
• Let \( a \) and \( b \) are integers and \( b \) is not equal to 0;
• then we say \( b \) is a divisor of \( a \) if there is an integer \( m \) such that \( a = mb \);

• Prime numbers
• An integer \( p \) is a prime number if its only divisors are 1, -1, p, -p
gsd and relatively prime numbers

- \( \gcd(a, b) \) is a greatest common divisor of \( a \) and \( b \)
- Examples: \( \gcd(12, 15) = 3; \gcd(49, 14) = 7. \)

- \( a \) and \( b \) are *relatively prime* if \( \gcd(a, b) = 1. \)
- Example: \( \gcd(9, 14) = 1. \)
Modular arithmetic

• If $a$ is an integer and $n$ is a positive integer, we define $a \mod n$ to be the remainder when $a$ is divided by $n$:

  $a =qn+r$,

• Here $q$ is a quotient and $r = a \mod n$

• If $(a \mod n) = (b \mod n)$ then $a$ and $b$ are congruent modulo $n$;

• It is easy to see, that $(a \mod n) = (b \mod n)$ iff $n$ is a divisor of $a-b$. 
Modular arithmetic. Properties

• \[(a \mod n) + (b \mod n)] \mod n = (a+b) \mod n\]

• \[(a \mod n) – (b \mod n)] \mod n = (a-b) \mod n\]

• \[(a \mod n) \times (b \mod n)] \mod n = (a \times b) \mod n\]

• Example: 3 \mod 5 \times 4 \mod 5 = 12 \mod 5 = 2 \mod 5