- Network security services:
  - confidentiality: messages, traffic amount & origin/destination
  - integrity: verifying that a message has not been altered (data integrity), replayed (originality) or delayed (timeliness)
  - authentication: verifying identity of a participant
  - availability: ensuring some degree of access
  - nonrepudiation & nonforgeability

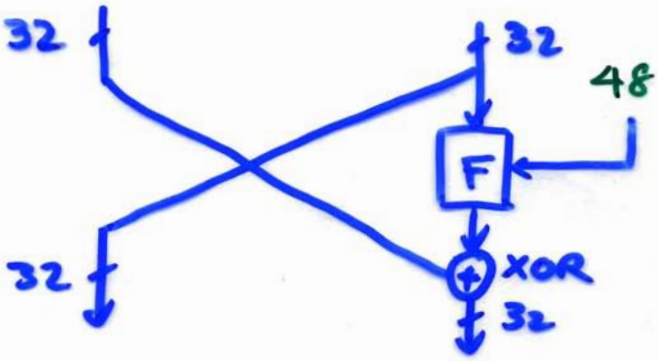
- Approaches:
  - information theoretic
  - cryptographic
  - firewalls

- basic cryptographic tools — provide building blocks for security mechanisms & secure systems
  - symmetric-key ciphers: both participants share a common secret key for encryption & decryption
    - eg. DES, 3DES, AES
  - public key ciphers: each participant has a private key known only to itself, & a public key that is published
    - to send a message to a participant A, encrypt using A's public key; A decrypts using his private key
    - can also be used in reverse for authentication instead of confidentiality: A can encrypt a message with his private key so that it can only be decrypted with his public key
    - eg. RSA, ElGamal
  - cryptographic hash functions: one-way functions that compute a message digest/ cryptographic checksum st. it is computationally difficult for an adversary to find any plaintext message with a given digest value, or to find 2 different messages with the same digest value
    - eg MD5, SHA-1, SHA-2

- **Encryption algorithms / ciphers**
  - turn plaintext into ciphertext in such a way that the plaintext can only be recovered by a participant that has the secret decryption key
  - algorithm itself is assumed known
  - types of attacks:
    - ciphertext only: attacker only has a set of ciphertexts, tries to recover info on message or key
    - known plaintext: attacker has samples of both plaintext & the corresponding ciphertext, tries to recover key
    - chosen plaintext: attacker can choose arbitrary plaintexts & obtain the corresponding ciphertexts, tries to recover key
      eg. in public key cryptography

Data Encryption Standard (DES)

- block cipher: takes a fixed length string of plaintext bits & performs a series of complicated operations on it to produce the ciphertext
- a 64-bit block of plaintext is encrypted using a key consisting of 56 bits + 8 parity bits, producing a 64-bit block of ciphertext
- procedure:
  - plaintext bits are permuted
  - 16 identical processing rounds are applied
  - the inverse of the original permutation is applied



48 bit subkey: different subkey for each round, applied in reverse order for decryption

- structure of each of the 16 processing rounds
- the F function permutes & expands the half block to 48 bits, XORs in the subkey, performs a nonlinear transform (by means of substitution boxes with

lookup tables) to 32 bits, & permutes the result

- complicated procedure provides 'confusion & diffusion', i.e. making the relationship between the key & ciphertext complicated, & diffusing any statistical redundancy in the plaintext in the ciphertext statistics (Shannon 49)

- to encrypt >64 bits, cipher block chaining (CBC) can be used to hide patterns in plaintext blocks
  - each plaintext block is XORed with the previous block's ciphertext (or an initialization vector, for the 1st block)

- attacks
  - brute force: exhaustive search over all 2^{56} possible keys, practical with current processor speeds
  - analytical: lower time complexity but requires a large no. of known / chosen plaintexts

- DES is no longer recommended by NIST for new systems, but is still widespread in legacy systems

- **Triple DES (3DES)**
  - leverages cryptanalysis resistance of DES while tripling the key size
  - a 3DES key has $3 \times 56 = 168$ independent bits used as 3 DES keys
  - encryption procedure:
    - DES encrypt w/ DES key 1
    - DES decrypt w/ DES key 2
    - DES encrypt w/ DES key 3
  - decryption procedure reverses the encryption steps
  - interoperable with legacy DES systems
    - set all 3 keys to the same legacy DES key
  - 3DES solves DES's key length problem but inherits inefficiencies of small 64 bit block size & hardware-centric DES design
- **Advanced Encryption Standard (AES)**
  - new standard issued by NIST in 2001
  - block length 128 bits, key sizes 128, 192 or 256 bits
  - uses a substitution-permutation network consisting of 10, 12 or 14 rounds of
    - substitution boxes: nonlinear transforms

designed using finite field theory to have certain desirable properties

- permutation boxes: permutations of intermediate inputs / outputs
- at each round a subkey derived from the main key is XORed in

- has fast hardware & software implementations

- RSA (Rivest, Shamir & Adleman 77, Cocks 73)
  - security is based on the premise that factoring large numbers & the RSA problem are computationally expensive
    - RSA problem: perform the RSA private key decryption operation using only the public key
  - key generation procedure:
    - choose 2 large distinct primes p & q randomly
    - compute $n = pq$ & $\varphi(n) = (p-1)(q-1)$
    - choose an integer e st. $1 < e < \varphi(n)$, & e & $\varphi(n)$ are coprime (ie. have no common factor >1)
    - compute $d \in \mathbb{Z}^+$ s.t. $de \equiv 1 \mod \varphi(n)$

    → public key: $\langle n, e \rangle$; private key: $\langle n, d \rangle$
    - p, q & other intermediate values must be kept secret / securely deleted
  - encryption: a plaintext message $m < n$ is encrypted to form ciphertext $c = m^e \mod n$
  - decryption: m is recovered from c via $m = c^d \mod n$

    Proof: $c^d = m^{ed}$

    $ed = k(p-1)+1 = h(q-1)+1$ since $ed \equiv 1 \mod \varphi(n)$ for some $k, h \in \mathbb{Z}^+$

    Case 1: m is not a multiple of p

    $\Rightarrow m^{p-1} \equiv 1 \mod p$ by Fermat's little thm

    $m^{ed} = m^{k(p-1)+1} = (m^{p-1})^k m \equiv m \mod p$

Case 2: m is a multiple of p

$$\Rightarrow m^{ed} \equiv 0 \bmod p$$
$$\equiv m \bmod p$$

Similarly, $m^{ed} \equiv m \bmod q$

$\therefore m^{ed} \equiv m \bmod pq$ by the Chinese Remainder Thm since p & q are distinct primes

i.e. $c^d \equiv m \bmod n$ □

- <u>authentication</u>:

$c = m^d \bmod n$

$m = c^e \bmod n$

- as before, $c^e = m^{de}$
$\equiv m \bmod n$

- security relies on sufficiently large n (at least 1024 bits) & d
  - larger than keys for symmetric key ciphers because factoring is faster than exhaustive search over the key space
  - → RSA is several orders of magnitude slower than symmetric-key ciphers
  - → public-key ciphers generally used for authentication & session key establishment while symmetric-key ciphers are used for the majority of encryption

- cryptographic hash functions MD5, SHA-1, SHA-2
  - compute a fixed length cryptographic checksum from an arbitrary length message
  - security relies on the complexity of the calculation procedure (many rounds of bitwise operations, bit shifts, addition of various constants applied to the message)
  - operations are efficiently implemented in software