- **CIA**, a modern definition. Confidentiality: prevent unauthorized reading of information. Integrity: detect unauthorized writing of information. Availability: data is available in a timely manner when needed.
- **Network Security**. Various protocols play a critical role, and cryptography matters a lot in protocol (especially network protocols) design and analysis.
- **Kerckhoof's Principle**. The system is completely known to the attacker; only the key is secret; the crypto algorithms are not secret.
- **Confusion and Diffusion**. Confusion: obscuring the relationship between plaintext and ciphertext. Diffusion: spreading the plaintext statistics through the ciphertext. A little note: hash function can be viewed as *one way cryptography*.
- **Stream Cipher**. Both A5/1 and RC4 are examples of this symmetric cryptosystem. It generalized the idea of a one-time pad, except that we trade provably security with a relatively small (and manageable) key. The key is stretched into a long stream of bits, which is then used just like a one-time pad.
- **Block Cipher**. It's really just an "electronic" version of a codebook, and employs both confusion and diffusion.

Algorithm 1 RC4 Keystream Byte

$i = (i + 1) \mod 256$
$j = (j + S[i] \mod 256)$
swap $(S[i], S[j])$
$t = (S[i] + S[j] \mod 256)$
$Keystream\ byte = S[t]$

- **Feistel Cipher**. It's a general cipher design principle. $L_i = R_{i-1}$ and $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$.
- **DES**. The security of this cryptosystem has much to do with *S-box*. Steps: an initial permutation before round 1; halves are swapped after last round; a final permutation applied to R_{16}, L_{16}.

Algorithm 2 TEA Encryption

$(K[0], K[1], K[2], K[3]) = 128\ bit\ key$
$(L, R) = plaintext$ $(64-bit$ block)
$delta = 0x9e3779b9$
$sum = 0$
for $i = 1$ to 32 **do**
  $sum = sum + delta$
  $L = L + (((R \ll 4) \oplus K[0]) \oplus (R + sum) \oplus ((R \gg 5) \oplus K[1]))$
  $R = L + (((L \ll 4) \oplus K[2]) \oplus (L + sum) \oplus ((L \gg 5) \oplus K[3]))$
  next i
end for
$ciphertext = (L, R)$

- **Block Cipher Modes**. ECB: encrypt each block independently. CBC: chain the blocks together. For this mode, a random initialization vector is required. CTR: block cipher acts like stream one.
- **Data Integrity**. The encryption process does provide confidentiality, but no guarantee of integrity.

Algorithm 3 Key generation for RSA public key encryption

Ensure: Each entity creates an RSA public key and a corresponding private key. Each entity A should do the following:

1. Generate two large random and distinct primes p and q, each roughly the same size.
2. Compute $n = pq$ and $\phi = (p-1)(q-1)$.
3. Select a random integer e, $1 \le e \le \phi$, such that $\gcd(e, \phi) = 1$.
4. Use the extended Euclidean algorithm to compute the unique integer d, such that $ed \equiv 1 \mod \phi$.
5. A's public key is (n, e), private key is d.

- **RSA Validity Proof**.
  - Since $ed \equiv 1 \mod \phi$, there exists an integer k such that $ed = 1 + k\phi$.
  - Now if $\gcd(m, p) = 1$, then by Fermat's theorem, $m^{p-1} \equiv 1 \mod p$.
  - Raising both sides of this congruence to the power $k(q-1)$ and then multiplying both sides by m yields $m^{1+k(p-1)(q-1)} \equiv m \mod p$.
  - On the other hand if $\gcd(m, p) = p$, then this last congruence is valid since each side is congruence to 0 mod p.
  - Hence, in all cases, $m^{ed} \equiv m \mod p$. By the same argument, $m^{ed} \equiv m \mod q$.
  - Finally, since p and q are distinct primes, it follows that $m^{ed} \equiv m \mod n$. And hence, $c^d \equiv (m^e)^d \equiv m \mod n$.
- **Cube Root attack** on RSA. A simple but practical way to prevent is to pad message with random bits.
- **Cryptographic Hash Function**. This function must provide the following:
  - Compression. For any size input x, the output length, i.e. $h(x)$ is small. Usually a fixed length is predefined.
  - Efficiency. It must be easy to compute $h(x)$ for any input x.
  - One way. Given any value y, it's computationally infeasible to find a value x such that $h(y) = x$.
  - Weak Collision Resistance. Given x and $h(x)$, it's infeasible to find any y, with $y \ne x$, such that $h(y) = h(x)$.
  - Strong Collision resistance. It's (and should be so) infeasible to find any $x \ne y$ such that $h(x) = h(y)$.
- **Birthday Problem**. Strong one. How large much the N be before the probability that someone shares the same birthday with me? Weak one. How many people must be in a room before the probability of at least two share the same birthday is larger than 0.5?

- Yet another “final exam”.