Contents

1. Definitions

- Safety: for reasonable inputs, get reasonable outputs.
- Security: for unreasonable inputs, get reasonable outputs.

1.1. Security Theatre

- Threat: Possibility of damage,
- Countermeasure: Limits possibility or consequence of damage,
  - mitigates threats, disables attacks, removes/reduces vulnerabilities.
- Vulnerabilities: Weakness in the system,
  - enables threats.
- Attacks: Exploitation of vulnerabilities to realize a threat.

1.2. CIA

- (C) Confidentiality: Information is disclosed to legitimate users.
- (I) Integrity: Information is created or modified by legitimate users.
- (A) Availability: Information is accessible to legitimate users.

Notice that CIA can be conflicting to each other in some scenarios.

1.3. Risk Analysis & Policy, Mechanisms and Assurance

Risk analysis and security policy

- Goal: Infer what can go wrong with the system.
- Outcome: A set of security goals.
- Principle: You never prevent threats, you lower the risk.

Mechanisms
- Goal: Define a strategy to realize the security goals.
- Outcome: Set of security mechanisms.
- Principle: Deploying security mechanisms has a cost.

Assurance
- Goal: Make sure that the security mechanisms realize the security goals.
- Outcome Methodology.
- Principle: Full assurance cannot be achieved.

1.4. Risk analysis

Given

$$\text{Risk exposure} = \text{probability} \times \text{impact}$$

We can set up a risk table to list out all the possible risk with their risk exposure, and determine which risks to mitigate.

2. Cryptography

Design Principles
- Kerkoff Principle: The security of a crypto system must not rely on keeping the algorithm secret.
- Diffusion: Mixing-up symbols.
- Confusion: Replacing a symbol with another.
- Randomization: Repeated encryptions of the same text are different.

2.1. Symmetric Cryptography

Requirements

We use the same key k for encryption E_k and decryption D:
- $D_k(E_k(m)) = m$ for every key k and E, D.
- E_k and D_k are easy to compute.
- Given $c = E_k(m)$, it is hard to find the plaintext m.

Attacks
- **Exhaustive Search** (brute force)
- **Ciphertext only**: know one or several random ciphertext.
- **Known plaintext**: know one or several pairs of **random** plaintext and their corresponding ciphertext.
- **Chosen plaintext**: know one or several pairs of **chosen plaintext** and their corresponding ciphertext.
- **Chosen cipher text**: know one or several pairs of plaintext and their corresponding **chosen ciphertext**.

2.1.1. Stream Cipher

XOR cipher:

- Message and key are xor-ed together

$$E_k(m) = k \oplus m$$

$$D_k(c) = k \oplus c$$

However, this cipher is vulnerable to known-plaintext attack

$$k = (k \oplus m) \oplus m$$

Mauborgne Cipher

- Use the key k as a seed for random number generator and xor with the message

$$E_k(m) = m \oplus \mathrm{RNG}(k)$$

Vulnerable to key re-use attack:

$$\begin{aligned} C_1 &= k \oplus m_1 \\ C_2 &= k \oplus m_2 \\ C_1 \oplus C_2 &= m_1 \oplus m_2 \end{aligned}$$

2.1.2. Block Cipher

Ideal block cipher

- Combines confusion and diffusion.
- Changing single bit in plaintext block or key results in changes to approximately half the ciphertext bits.

DES (Data Encryption Standard)

DES is broken in 1998 and 2006. And Nesting encryption process is not a valid counter-measure.

$$\mathrm{2DES}_{k_1,k_2}(m) = E_{k_2}\left(E_{k_1}(m)\right)$$

To broke this paradigm we can brute for the result of $E_{k_1}(m)$ and $D_{k_2}(c)$, for every possible key pair (k_1, k_2). Then match the valid key candidate. The effective key space only doubled, from 56 bits become 57 bits.

However, triple DES is widely used

$$\mathrm{3DES}_{k_1,k_2,k_3}(m) = E_{k_3}\left(D_{k_2}\left(E_{k_1}(m)\right)\right),$$

with effective key length 112 bits.

AES (Advanced Encryption Standard)

It has different encryption modes:

- ECB: electronic code book. Each plaintext block is encrypted independently with the key
  - Fast, easy to perform parallelization.
  - But same block is encrypted to same ciphertext (violates diffusion)

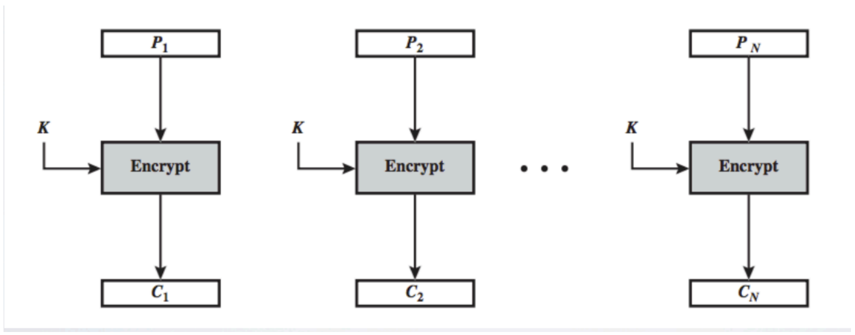

Figure 1: AES ECB mode

- CBC: cipher block chaining
  - Repeating plaintext blocks are not exposed in the ciphertext.
  - No parallelism.

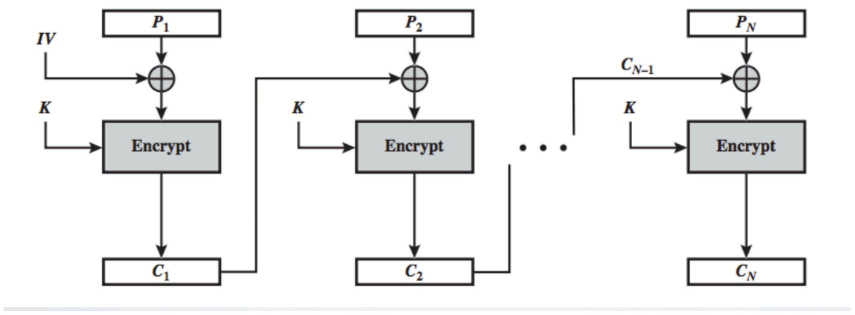

Figure 2: AES ECB mode

- CFB: cipher feedback
- CTR: counter
  - High entropy and parallelism.
  - Vulnerable to key-reused attack

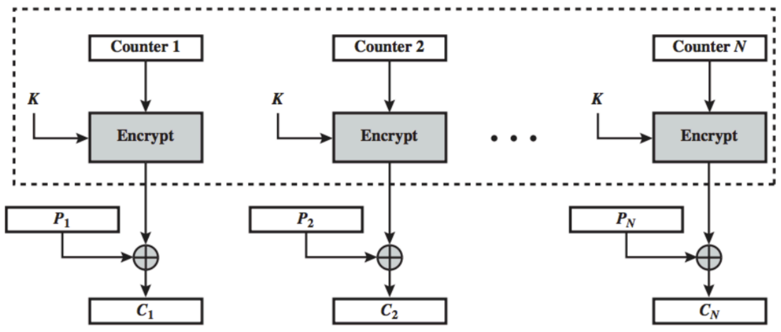

Figure 3: AES ECB mode

Name	Type	Key size (bits)	Speed (cycle/byte)
RC4	stream	40-2048	8
ChaCha20	stream	128/256	4
DES	block	block: 64, key: 56	50
Rijndael	block	block: 128, key: 128192256	18-20

The trade-off between stream cipher and block cipher:
- stream cipher is fast but has low diffusion, whereas
- block cipher is slow but has high diffusion.

2.2. Asymmetric Cryptography

Function Requirements

- $D_{Ks}\left(D_{Kp}(m)\right) = D_{Kp}(D_{Ks}(m)) = m$ for every key pair (Kp, Ks).
  - Easy to generate the key pair.
  - Encryption and decryption are easy to compute.
- Hard to matching key Ks given Kp

Name	Speed (cycle/byte)	Key size (bits)	Effective key length (bits)
RSA	10^6	1024	80
RSA	10^6	2048	112
RSA	10^6	3072	128
RSA	10^6	4096	140
RSA	10^6	15360	224-256
ECC	10^6	256	128
ECC	10^6	448	224-256

Summary between symmetric and asymmetric cryptography:
- Symmetric is fast but has key agreement.
  - That is, parties is able to generate a secrete key even if an eavesdropper is listening to the communication channel.
  - Often used to encrypt message
- Asymmetric is slow but does not have key agreement.
  - Used for encrypt shared key or hash.

2.3. Hash Functions

$$H(m) = x$$

An ideal hash function satisfies:

- PR (Preimage Resistance): given x it is hard to find m.
- PR2 (Second Preimage Resistance): given H, m, x it is hard to find m' such that $H(m) = H(m') = x$.
- CR (Collision Resistance): given H, it is hard to find m, m' such that $H(m) = H(m')$.

2.3.1. Security Issue

Due to birthday paradox:

"There are 50% chance that 2 people have the same birthday in a room of 23 people"

Therefore if given hash function of n-bits output, a collision can be found in around $2^{\frac{n}{2}}$ evaluations. Hence SHA-256 has 128 bits security.

SHA-2

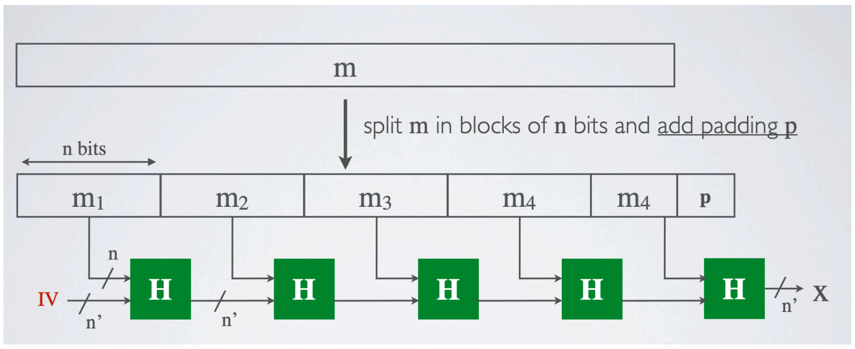

Figure 4: MD5, SHA-1, SHA-2

SHA-3

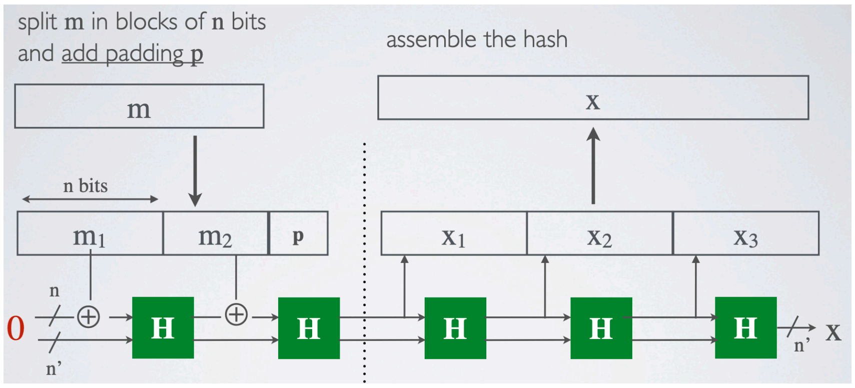

Figure 5: SHA-3

2.3.2. Hash as MAC

MAC stands for message authentication code, commonly used for key exchange, certificate… Given message m and key k, people often sends a whole message

$$m \parallel \mathrm{MAC}_k(m)$$

together. One variant is HMAC, which use a hash function on the message and the key.

- But in practice, if the HMAC is badly designed, for instance, using SHA2 and let $\mathrm{MAC}_k(m) = H(k \parallel m)$, then mallory can perform hash length extension attack on the message sent.
- Good HMAC example

$$\mathrm{HMAC}_k(m) = H(k \parallel m \parallel k)$$
$$\mathrm{HMAC}_k(m) = H((k \oplus \mathrm{opad}) \parallel H((k \oplus \mathrm{ipad}) \parallel m)$$

3. Protocols

3.1. Symmetric Protocols

Symmetric protocols utilize the advantage of symmetric cryptography to provide communication with confidentiality and integrity. The protocol defines a procedure of key exchange which ensures two parties are able to defend themselves from the attack of a malicious party during communication. Once the shared key k is set up, they can communicate by:

- Encrypt and MAC (E&M)

$$\mathrm{AE}_k(m) = E_{k(m)} \parallel H_k(m)$$

- MAC then Encrypt (MtE)

$$\mathrm{AE}_k(m) = E_k(m \parallel H_k(m))$$

- Encrypt then MAC (EtM)

$$\mathrm{AE}_k(m) = E_k(m) \parallel H_k(E_k(m))$$

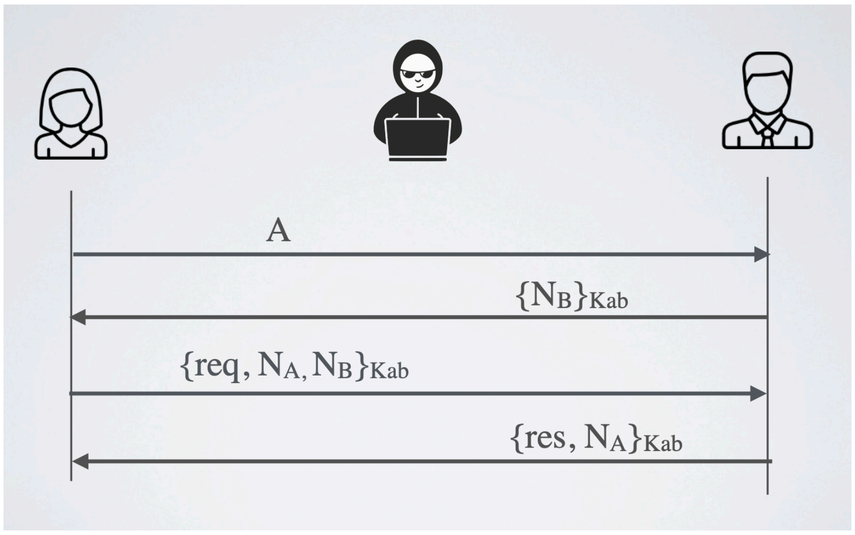

Figure 6: Communication in symmetric protocol

However, the primary issue with symmetric protocols is how to make an agreement on the shared key used K_{ab} used between to parties, say, Alice and Bob.

The naive way is for every connection in a network, a shared key is exchange physically using a secure channel. Therefore, $\frac{1}{2}n(n-1)$ keys are required for a network with n nodes.

A better solution is purposed as a KDC (key distributed center) manages all the keys used, with premisses:
- the key exchange channel between KDC and each party is secure,
- KDC is trusted.

Before, there were a vulnerable version of key exchange protocol

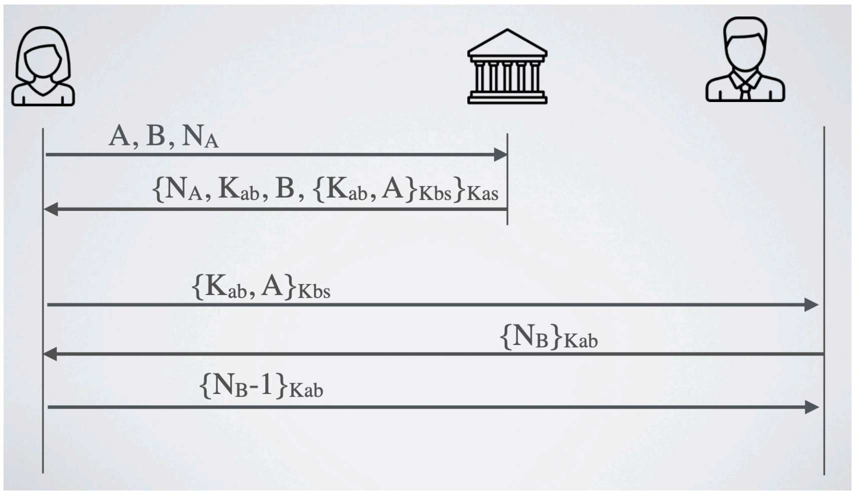

Figure 7: Vulnerable KDC

If the key K_{ab} is compromised, then Mallory is able to perform replay attack.

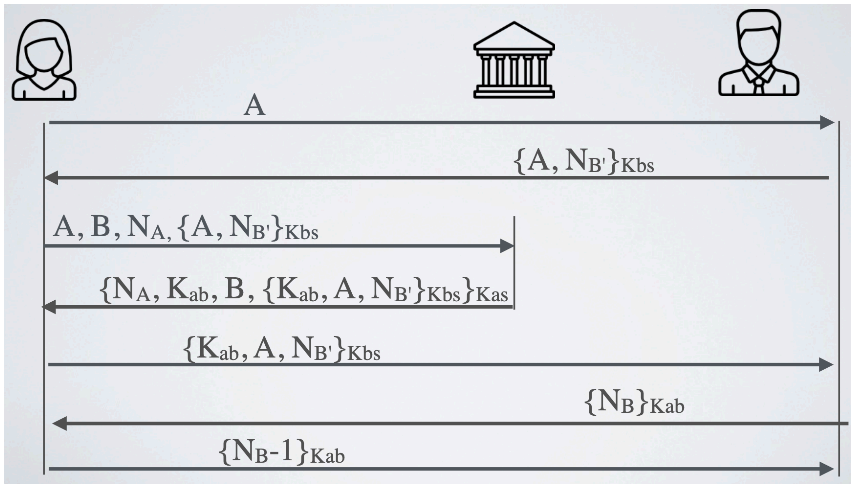

Figure 8: Fixed KDC

But the existence of KDC has drawbacks:
- it is a single point of failure,
- one cannot exchange key with another when zero knowledge.

To solve this, we have DH (Diffie-Hellman) protocol:

- Alice
  - generate g, p as public key, where g is small (2, 5, 7...) and p is at least 2048 bits.
  - choose a, a 2048 bits private key
  - compute $\text{dhA} = g^a \bmod p$
  - send p, dhA, n_0 to Bob
- Then Bob
  - choose b, another 2048 bits private key
  - compute $\text{dhB} = g^b \bmod p$
  - send dhB, n_1 back to Alice
- The session key K is

$$K = g^{ab} \bmod p = (g^a \bmod p)^b \bmod p = \left(g^b \bmod p\right)^a \bmod p.$$

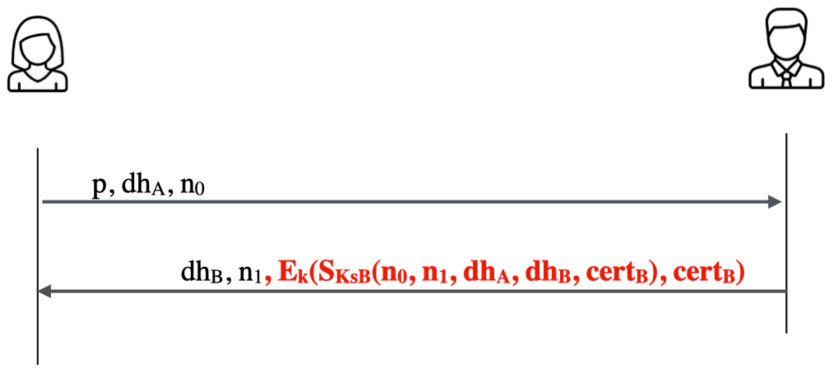

Figure 9: DH key exchange with authentication

3.2. Digital Signature

MAC we have discussed provide a decent method of verification under the scenario of no prior trust is given to the connection. Whereas it is easy to be forged. To solve this issue, digital signature is introduced, where it is

- commonly used in key exchange