Shared-key encryption

Daniel Bosk

School of Computer Science and Communication,
KTH Royal Institute of Technology, Stockholm

Department of Information and Communication Systems,
Mid Sweden University, Sundsvall

6th April 2020

1 Shared-key cryptography
- Ciphers
- Security

Idea

- Alice and Bob share a (small) common secret.
- Alice takes a message, combines it with the secret, sends it to Bob.
- If Eve captures whatever Alice sent, she shouldn't learn anything about the message.
- Bob combines what he received with the secret and gets the message.

Idea

- Alice and Bob share a (small) common secret.
- Alice takes a message, combines it with the secret, sends it to Bob.
- If Eve captures whatever Alice sent, she shouldn't learn anything about the message.
- Bob combines what he received with the secret and gets the message.

Idea

- Alice and Bob share a (small) common secret.
- Alice takes a message, combines it with the secret, sends it to Bob.
- If Eve captures whatever Alice sent, she shouldn't learn anything about the message.
- Bob combines what he received with the secret and gets the message.

Idea

- Alice and Bob share a (small) common secret.
- Alice takes a message, combines it with the secret, sends it to Bob.
- If Eve captures whatever Alice sent, she shouldn't learn anything about the message.
- Bob combines what he received with the secret and gets the message.

Block-cipher encryption

Input A fixed-sized *key* k, a fixed-sized block of *plaintext* p.

Output A fixed-sized block of *ciphertext* c.

Notation $\mathrm{Enc}_k(p) = c$

Block-cipher decryption

Input A fixed-sized *key* k, a fixed-sized block of *ciphertext* c.

Output A fixed-sized block of *plaintext* p.

Notation $\mathrm{Dec}_k(c) = p$

Block-cipher encryption

Input A fixed-sized *key* k, a fixed-sized block of *plaintext* p.

Output A fixed-sized block of *ciphertext* c.

Notation $\text{Enc}_k(p) = c$

Block-cipher decryption

Input A fixed-sized *key* k, a fixed-sized block of *ciphertext* c.

Output A fixed-sized block of *plaintext* p.

Notation $\text{Dec}_k(c) = p$

Definition (Crypto system[1])

- A *crypto system* is a tuple $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where:
  - $\mathcal{M}$ is a finite set of *plaintexts* or messages,
  - $\mathcal{C}$ is a finite set of *ciphertexts*,
  - $\mathcal{K}$ is the *keyspace*, a finite set of keys.
  - $\mathcal{E}$ and $\mathcal{D}$ are the sets of encryption and decryption rules, respectively.
- For every $k \in \mathcal{K}$ there is a $\mathrm{Enc}_k \in \mathcal{E}$ and $\mathrm{Dec}_k \in \mathcal{D}$ such that
  - $\mathrm{Enc}_k : \mathcal{M} \to \mathcal{C}$ and $\mathrm{Dec}_k : \mathcal{C} \to \mathcal{M}$ are functions and
  - $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$ for all plaintexts $m \in \mathcal{M}$.

[1] **Stinson2006cta**.

Definition (Crypto system[1])

- A *crypto system* is a tuple $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where:
  - $\mathcal{M}$ is a finite set of *plaintexts* or messages,
  - $\mathcal{C}$ is a finite set of *ciphertexts*,
  - $\mathcal{K}$ is the *keyspace*, a finite set of keys.
  - $\mathcal{E}$ and $\mathcal{D}$ are the sets of encryption and decryption rules, respectively.
- For every $k \in \mathcal{K}$ there is a $\mathrm{Enc}_k \in \mathcal{E}$ and $\mathrm{Dec}_k \in \mathcal{D}$ such that
  - $\mathrm{Enc}_k : \mathcal{M} \to \mathcal{C}$ and $\mathrm{Dec}_k : \mathcal{C} \to \mathcal{M}$ are functions and
  - $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$ for all plaintexts $m \in \mathcal{M}$.

[1] **Stinson2006cta**.

Definition (Shift Cipher)

- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{29}$
- For each $k \in \mathcal{K}$ we define

$$\mathrm{Enc}_k(m) = (m + k) \bmod 29, m \in \mathcal{M}, \text{ och}$$
$$\mathrm{Dec}_k(c) = (c - k) \bmod 29, c \in \mathcal{C}.$$

Example

- $\mathrm{Enc}_3(7) = 7 + 3 \bmod 29 = 10$ h→J
- $\mathrm{Enc}_3(4) = 4 + 3 \bmod 29 = 7$ e→G
- $\mathrm{Enc}_3(9) = 9 + 3 \bmod 29 = 12$ j→L

Definition (Shift Cipher)

- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{29}$
- For each $k \in \mathcal{K}$ we define

$$\mathrm{Enc}_k(m) = (m + k) \bmod 29, m \in \mathcal{M}, \text{ och}$$
$$\mathrm{Dec}_k(c) = (c - k) \bmod 29, c \in \mathcal{C}.$$

Example

- $\mathrm{Enc}_3(7) = 7 + 3 \bmod 29 = 10$ h→J
- $\mathrm{Enc}_3(4) = 4 + 3 \bmod 29 = 7$ e→G
- $\mathrm{Enc}_3(9) = 9 + 3 \bmod 29 = 12$ j→L

Note

- The shift cipher is a classical cipher — also know as the Caesar Cipher.
- It's easily broken *by hand*!
- It's used here for illustrative purposes.

Definition (Perfect secrecy[2])

- Cryptosystem $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$.
- Stochastic variables M, C.
- *Perfect secrecy* if and only if

$$\Pr(M = m \mid C = c) = \Pr(M = m)$$

for all $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

Note

Equivalent to $H(M \mid C) = H(M)$, i.e. ciphertext does not reveal anything about plaintext.

[2] **ShannonSecrecy**.

Definition (Perfect secrecy[2])

- Cryptosystem $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$.
- Stochastic variables M, C.
- *Perfect secrecy* if and only if

$$\Pr(M = m \mid C = c) = \Pr(M = m)$$

for all $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

Note

Equivalent to $H(M \mid C) = H(M)$, i.e. ciphertext does not reveal anything about plaintext.

[2] **ShannonSecrecy**.

Theorem (Shannon's theorem)

- *Assume cryptosystem* $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *such that* $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{M}|$.
- *This provides perfect secrecy if and only if*
  1. *every key* $k \in \mathcal{K}$ *is used with equal probability* $1/|\mathcal{K}|$,
  2. *for every plaintext* $m \in \mathcal{M}$ *and* $c \in \mathcal{C}$ *there is a unique key such that* $\mathrm{Enc}_k(m) = c$.

Theorem (Shannon's theorem)

- *Assume cryptosystem* $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ *such that* $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{M}|$.
- *This provides perfect secrecy if and only if*
  1. *every key* $k \in \mathcal{K}$ *is used with equal probability* $1/|\mathcal{K}|$,
  2. *for every plaintext* $m \in \mathcal{M}$ *and* $c \in \mathcal{C}$ *there is a unique key such that* $\mathrm{Enc}_k(m) = c$.

Example (One-time Pad)

- Let n be a positive integer.
- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$.
- For each key $k = (k_1, \ldots, k_n) \in \mathcal{K}$, plaintexts $m = (m_1, \ldots, m_n) \in \mathcal{M}$ and ciphertexts $c = (c_1, \ldots, c_n) \in \mathcal{C}$ we define

$$\mathrm{Enc}_k(m) = (m_1 + k_1, \ldots, m_n + k_n)$$

- We also define Dec = Enc.
- $k \in \mathcal{K}$ must be chosen uniformly randomly for each encryption.

Example (One-time Pad)

- Let n be a positive integer.
- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$.
- For each key $k = (k_1, \ldots, k_n) \in \mathcal{K}$, plaintexts $m = (m_1, \ldots, m_n) \in \mathcal{M}$ and ciphertexts $c = (c_1, \ldots, c_n) \in \mathcal{C}$ we define

$$\mathrm{Enc}_k(m) = (m_1 + k_1, \ldots, m_n + k_n)$$

- We also define Dec = Enc.
- $k \in \mathcal{K}$ must be chosen uniformly randomly for each encryption.

Example (One-time Pad)

- Let n be a positive integer.
- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$.
- For each key $k = (k_1, \ldots, k_n) \in \mathcal{K}$, plaintexts $m = (m_1, \ldots, m_n) \in \mathcal{M}$ and ciphertexts $c = (c_1, \ldots, c_n) \in \mathcal{C}$ we define

$$\mathrm{Enc}_k(m) = (m_1 + k_1, \ldots, m_n + k_n)$$

- We also define $\mathrm{Dec} = \mathrm{Enc}$.
- $k \in \mathcal{K}$ must be chosen uniformly randomly for each encryption.

Example (One-time Pad)

- Let n be a positive integer.
- Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$.
- For each key $k = (k_1, \ldots, k_n) \in \mathcal{K}$, plaintexts $m = (m_1, \ldots, m_n) \in \mathcal{M}$ and ciphertexts $c = (c_1, \ldots, c_n) \in \mathcal{C}$ we define

$$\mathrm{Enc}_k(m) = (m_1 + k_1, \ldots, m_n + k_n)$$

- We also define Dec = Enc.
- $k \in \mathcal{K}$ must be chosen uniformly randomly for each encryption.

Definition (Pseudo-random permutation, PRP[3])

- Let $F\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^n$.
- F is a PRP if
  1. for any $k \in \{0,1\}^s$, F is a bijection;
  2. for any $k \in \{0,1\}^s$, we can 'efficiently' evaluate $F_k(x)$;
  3. for all 'efficient' distinguishers D,

$$\left|\Pr[D^{F_k}(1^n) = 1] - \Pr[D^{f_n}(1^n) = 1]\right| < \epsilon(s)$$

     if we choose $k \in \{0,1\}^s$ and the random permutation f_n uniformly at random.

[3] **KatzLindell-v1**.

Definition (Pseudo-random permutation, PRP[3])

- Let $F\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^n$.
- F is a PRP if
  1. for any $k \in \{0,1\}^s$, F is a bijection;
  2. for any $k \in \{0,1\}^s$, we can 'efficiently' evaluate $F_k(x)$;
  3. for all 'efficient' distinguishers D,

$$\left|\Pr[D^{F_k}(1^n) = 1] - \Pr[D^{f_n}(1^n) = 1]\right| < \epsilon(s)$$

     if we choose $k \in \{0,1\}^s$ and the random permutation f_n uniformly at random.

[3] **KatzLindell-v1**.

Definition (Pseudo-random permutation, PRP[3])

- Let $F\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^n$.
- F is a PRP if
  1. for any $k \in \{0,1\}^s$, F is a bijection;
  2. for any $k \in \{0,1\}^s$, we can 'efficiently' evaluate $F_k(x)$;
  3. for all 'efficient' distinguishers D,

$$\left|\Pr[D^{F_k}(1^n) = 1] - \Pr[D^{f_n}(1^n) = 1]\right| < \epsilon(s)$$

     if we choose $k \in \{0,1\}^s$ and the random permutation f_n uniformly at random.

[3] **KatzLindell-v1**.

Definition (Pseudo-random permutation, PRP[3])

- Let $F\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^n$.
- F is a PRP if
  1. for any $k \in \{0,1\}^s$, F is a bijection;
  2. for any $k \in \{0,1\}^s$, we can 'efficiently' evaluate $F_k(x)$;
  3. for all 'efficient' distinguishers D,

$$\left|\Pr[D^{F_k}(1^n) = 1] - \Pr[D^{f_n}(1^n) = 1]\right| < \epsilon(s)$$

     if we choose $k \in \{0,1\}^s$ and the random permutation f_n uniformly at random.

[3] **KatzLindell-v1**.