Information Security Summary

Fabian Bösiger

April 1, 2021

Contents

1 Introduction

Key space: $\mathcal{K}$
Plaintext space: $\mathcal{M}$
Ciphertext space: $\mathcal{C}$
Encryption algorithm: $\text{Enc}_k(m) : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$
Decryption algorithm: $\text{Dec}_k(c) : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$
Encryption scheme: $(\text{Enc}_k, \text{Dec}_k)$ Correctness: $\forall k : \text{Dec}_k(\text{Enc}_k(m)) = m$

2 Historical Ciphers

2.1 Caesars's Shift Cipher

$\mathcal{M} = \{A, \ldots, Z\} = \{0, \ldots, 25\}$
$\mathcal{K} = \{0, \ldots, 25\}$
$\text{Enc}_k(m_0, \ldots, m_n) = (m_0 + k \mod 25, \ldots, m_n + k \mod 25)$
$\text{Dec}_k(c_0, \ldots, c_n) = (c_0 - k \mod 25, \ldots, c_n - k \mod 25)$

2.1.1 Vulnerabilities

Brute force attack.

2.2 Substitution Cipher

$\mathcal{M} = \{A, \ldots, Z\} = \{0, \ldots, 25\}$
$\mathcal{K} = \{0, \ldots, 25\}$
$\text{Enc}_k(m_0, \ldots, m_n) = (\pi(m_0), \ldots, \pi(m_n))$
$\text{Dec}_k(c_0, \ldots, c_n) = (\pi^{-1}(c_0), \ldots, \pi^{-1}(c_n))$

2.2.1 Vulnerabilities

Statistical patterns of the language.

2.3 Vigenere Cipher

TODO

2.3.1 Vulnerabilities

3 Information-Theoretic Security

If the key k is chosen randomly and $c := \mathrm{Enc}_k(m)$ is given to the adversary, the adversary should not learn any additional information about the plaintext m.

An encryption scheme is perfectly secret if for some random variables M, C and every m, c: $P(M = m) = P(M = m \mid C = c)$.

Equivalently: M and C are independet.

Equivalently: The distribution of C does not depend on M.

Equivalently: For every m_0, m_1 we have that $\mathrm{Enc}(k, m_0)$ and $\mathrm{Enc}(K, m_1)$ have the same distribution.

In every perfectly secret encryption scheme, we have $|\mathcal{K}| \geq |\mathcal{M}|$.

3.1 One-Time Pad

$\mathcal{M} = \mathcal{K} = \{0, 1\}^t$
$\mathrm{Enc}_k(m) = k \operatorname{xor} m$
$\mathrm{Dec}_k(c) = k \operatorname{xor} c$

3.1.1 Correctness

$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = k \operatorname{xor}(k \operatorname{xor} m)$

3.1.2 Perfect Secrecy

$$
\begin{aligned}
& P(C = c \mid M = m) && (1)\\
&= P(M \operatorname{xor} K = c | M = m) && (2)\\
&= P(m \operatorname{xor} K = c) && (3)\\
&= P(K = m \operatorname{xor} c) && (4)\\
&= 2^{-t} && (5)\\
&= P(C = c \mid M = m_0) = P(C = c \mid M = m_1) && (6)
\end{aligned}
$$

3.1.3 Vulnerabilities

Perfectly secret. But the key is as long as the message and cannot be reused.

4 Computational Security

A system X is (t, ϵ)-secure if every Turing Machine that operates in time t can break X with probability of at most ϵ.

A function $\mu : \mathbb{N} \to \mathbb{R}$ is negligible, if for every natural number c there exists n_0 such that for all $x > n_0$: $|\mu(x)| < \frac{1}{x^c}$

M and C are independet from the point of view of a computationally limited adversary with high probability.

More formally: X is secure if for all probabilistic poly-time turing machines M, $P(M$ breaks the scheme $X)$ is negligible.

Equivalently: No poly-time adversary can distinguish the distributions $\mathrm{Enc}(K, m_0) = \mathrm{Enc}(K, m_1)$ with non-negligible probability.

4.1 Chosen-Plaintext Attack

Learning phase: Adversary can repeatedly send message m that is encrypted using some unknown k and receives $c = \mathrm{Enc}(k, m)$.

Challenge phase: Adversary sends m_0 and m_1, receives $c = \text{Enc}(k, m_b)$ for some unknown b, has to guess b.

CPA-security: Every randomized poly-time adversary guesses b correctly with probability of at most $\frac{1}{2} + \epsilon(n)$ where ϵ is negligible.

CPA-secure encryptions have to be randomized or have a state.

If a CPA-secure encryption exists with $|k| \leq |m|$, then $P \neq NP$.

5 Pseudorandom Functions

Select random permutation $F : \{0,1\}^m \to \{0,1\}^m$, give it to both parties similar to secret key.

Problem: F requires $m * 2^m$ space.

Solution: Pseudorandom functions using a key $F_k : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$.

A keyed permutation F_k is pseudorandom if it cannot be distinguished from a completely random function. More formally assume two scenarios where a distinguisher D tries to distinguish random from pseudorandom function:

Scenario 0: D sends t random messages which are encrypted using the same pseudorandom function F_k with random keys.

Scenario 1: D sends t random messages which are encrypted using a true random function F.

F_k is a pseudorandom function if all probabilistic poly-time distinguishers D cannot distinguish scenarios 0 and 1 with a non-negligible advantage.

If a distinguisher additionaly has access to the inverted function F, we get the definition of a strong pseudorandom function.

6 Block Ciphers

Block ciphers are pseudorandom permutations F_k. THey use a key of K bits to specify a random subset of 2^K mappings. If the section of mappings is random, the resulting cypher will be a good approximation of the ideal block cypher.

6.1 Shannon's Confusion and Diffusion Principle

Diffusion: Ciphertext bits should depend on the plaintext bits in a complex way. If a plaintext bit is changed, ciphertext bits should change with $p = \frac{1}{2}$.

Confusion: Each bit of the ciphertext should depend on the whole key. If one bit of the key is changed, the ciphertext should change entirely.

6.2 Confusion-Diffusion Paradigm

Confusion: Implement large $F_k(m)$ using smaller $f_i(k, m_i)$, called substitution boxes. $F_k(m_1 m_2 \ldots m_n) = f_1(k, m_1) f_2(k, m_2) \cdots f_n(k, m_n)$.

Diffusion: Permute (Mix) the output F_k.

Key idea: Run the confusion and diffusion multiple times.

6.3 Data Encryption Standard (DES)

Input $\rightarrow$ Initial Permutation IP $\rightarrow$ Feistel Network depending on k $\rightarrow$ Final Permutation IP^{-1} $\rightarrow$ Output

TODO

A 3-round feistel network is a pseudorandom permutation.

A 4-round feistel network is a strong pseudorandom permutation.

To fully describe a feistel network we need to describe a key schedule algoritm and the pseudorandom permutation function f.

6.3.1 Vulnerabilities

Key is too short, brute force attack is possible.

Unclear role of the NSA in the design.

6.3.2 Triple Encryption

6.4 Advanced Encryption Standard (AES)

TODO: Topic 2

7 Stream Ciphers

Pseudorandom generators used in practice are called stream ciphers.

7.1 Pseudorandom Generators

7.2 RC4

7.2.1 Vulnerabilities

Some bytes of the output are biased.

The first few bytes sometimes leak information about the key.

7.3 ChaCha

8 Hash Functions and MACs

8.1 Message Authentication Codes (MACs)

Key space: $\mathcal{K}$
Plaintext space: $\mathcal{M}$
Set of tags: $\mathcal{T}$
Tagging algorithm: $\mathrm{Tag} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$
Verification algorithm: $\mathrm{Vrfy} : \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{0, 1\}$
MAC scheme: $(\mathrm{Tag}, \mathrm{Vrfy})$