Université Libre de Bruxelles
Faculté des Sciences

2023-2024

INFO-F-405: Introduction to cryptography

# 1. Historical ciphers and general principles

## Historical ciphers

### Exercise 1

A cipher with $\mathcal{M} = \mathcal{C}$ for certain keys $k \in \mathcal{K}$ is called *involutory* if its encryption and decryption procedures become identical, i.e., for all $m \in \mathcal{M}$, $E_k(m) = D_k(m)$. Under which keys $k \in [0, 25]$ the shift encryption scheme (see slides) becomes involutory?

### Exercise 2

The following text was encrypted using the shift encryption scheme. Try all possible secret keys to find the one that decrypts the following message:

```
Fbzr crbcyr jrne Fhcrezna cnwnznf. Fhcrezna jrnef Puhpx Abeevf cnwnznf.
```

How many attempts are needed to be sure to find the correct one?

Assuming the secret key is a uniformly-distributed random variable. What is the probability that the correct key is found after $t$ attempts?

Here is another ciphertext to decrypt:

```
'Yvccf, nficu!' - zj fev fw kyv wzijk kyzexj gvfgcv kip kf gizek nyve kyvp
cvrie r evn gifxirddzex crexlrxv.
```

### Exercise 3

In one of the previous exercices, we performed cryptanalysis by knowing only the ciphertexts. Here we consider known-plaintext attacks where the attacker gets access to some know plaintext/ciphertext pairs and wants to determine the secret key to be able to decrypt past or future messages.

How many pairs of plaintext/ciphertext characters (or bits) must be known in order to determine the secret key without ambiguity in the following ciphers?

1. In the shift encryption scheme?
2. In a general mono-alphabetic substitution scheme?
3. In a general poly-alphabetic substitution scheme (assuming the length $t$ of the key is known)?
4. In the Vigenère cipher ($t$ is known)?
5. In the binary Vigenère cipher ($t$ is known)?

# Perfect secrecy vs computational security

### Exercise 4

A bank defines an encryption algorithm to encrypt their account numbers. An account number is a 12-digit number, i.e., an element of $\mathbb{Z}_{10^{12}}$. Every time an account number is encrypted, a fresh key is chosen randomly and uniformly from the same set $\mathbb{Z}_{10^{12}}$. The encryption goes as follows:

$$E_k(m) = m + k \pmod{10^{12}} \quad D_k(c) = c - k \pmod{10^{12}}$$

Does this cipher satisfy perfect secrecy?

### Exercise 5

What happens if the one-time pad is incorrectly used and that two distinct plaintexts are encrypted with the same key, and why?

1. The key is compromised.
2. The two plaintexts are revealed.
3. The difference between the two plaintexts is revealed.
4. The authenticity of the plaintext is compromised.

### Exercise 6

Suppose that the probability of wining the lottery is $\frac{1}{10^6}$. What is more likely guessing an AES-128 key on the first try or winning the lottery 6 times in a row? What about winning the lottery 7 times in a row?

### Exercise 7

Assume an adversary performs an exhaustive key search on a huge network of $10^9$ computers, each capable of testing $10^9$ keys per second. After about how much time will a 128-bit key typically be found?

1. A few seconds.
2. A few days.
3. A few years.
4. A few centuries.
5. A few times the age of the universe.

## Security definitions

### Exercise 8

Suppose that $(\text{Gen}, E, D)$ is an IND-CPA secure probabilistic encryption scheme, i.e., there is no known way of winning the IND-CPA game other than with a probability negligibly close to $\frac{1}{2}$ or with over-astronomical resources. Let the key and plaintext spaces be $\{0,1\}^n$ with $n \geq 128$. Point out the IND-CPA secure schemes in the following list:

- $E'_k(m) = E_k(m)||\text{first bit}(m)$
- $E'_k(m) = (\text{last bit}(m) \oplus \text{first bit}(m))||E_k(m)$
- $E'_k(m) = E_k(m)||1$
- $E'_k(m) = E_k(m)||E_k(m)$
  (+a side-question: will the first and last $n$ bits of $E'_k(m)$ always be equal?)
- $E'_k(m) = k||E_k(m)$
- $E'_k(m) = E_k(m)||(k \oplus 1^n)$
- $E'_{k1,k2}(m) = E_{k1}(m)||E_{k2}(m)$
- $E'_k(m) = E_{0^n}(m)$

- $E'_k(m) = E_k(m)||(m \oplus k)$

For each scheme that is *not* IND-CPA secure, please specify how an adversary can win the IND-CPA game, i.e., what the adversary chooses as plaintexts $m_0$ and $m_1$, what are the queries to be made before or after this choice, how the adversary distinguishes between the ciphertexts of $m_0$ and $m_1$.

## Exercise 9

Let $(\text{Gen}, E, D)$ be a symmetric-key encryption scheme with diversification that is IND-CPA-secure. The definition of IND-CPA requires that the adversary respects the condition that the diversifier $d$ is a nonce. What would happen if this condition was not respected? Show how he can win this variant of IND-CPA game where $d$ is not unique, with the least number of queries.

## Exercise 10

Suppose that $(\text{Gen}, E, D)$ is an IND-CPA secure symmetric-key encryption scheme with diversification, i.e., there is no known way of winning the IND-CPA game other than with a probability negligibly close to $\frac{1}{2}$ or with over-astronomical resources. Let the key space be $\{0,1\}^n$ with $n \geq 128$, the plaintext space to be arbitrary and the diversifier space be the set of non-negative integers $\mathbb{N}$. Point out the IND-CPA secure schemes in the following list:

- $E'_k(d, m) = E_k(0, m)$
- $E'_k(d, m) = E_k(d + 1, m)$
- $E'_k(d, m) = E_k(d, m)\|0^d$
- $E'_k(d, m) = E_k(\lfloor d/2 \rfloor, m)$
- $E'_k(d, m) = E_{0^d\|1^{n-d}}(d, m)$
- $E'_k(d, m) = E_k(\text{random} \in \{0, 1, \ldots, 36\}, m)$
- $E'_k(d, m) = E_k(\text{random} \in \{0, 1, \ldots, 2^{256}\}, m)$

For each scheme that is *not* IND-CPA secure, please specify how an adversary can win the IND-CPA game, i.e., what the adversary chooses as diversifiers, as plaintexts $m_0$ and $m_1$, what are the queries to be made before or after this choice, how the adversary distinguishes between the ciphertexts of $m_0$ and $m_1$.

## Exercise 11

A bank issues a smartcard to each of its $N$ customers. Each smartcard $i$ contains its own secret key $K_i$ of $k$ bits. Because of the way the protocol was designed, each smartcard encrypts the string `"hello"` with its secret key and sends it over the network.

An attacker collects all these $N$ ciphertexts and starts a *multi-target exhaustive key search*. In more details, the attacker encrypts the string `"hello"` with each possible key in the key space until he gets a ciphertext that matches one of the collected ones. When it does, it probably means that he hits the secret key of one of the smartcards.

1. What is the security strength of the scheme under this attack as a function of $k$ and $N$?
2. If a bank issues a billion ($N \approx 2^{30}$) smartcards, each with its own $k = 128$-bit key, is the scheme still secure in practice?
3. What could the bank do to have a security strength of 128 bits?

## Exercise 12

An automated teller machine (ATM) distributes cash to the customers of a bank. It is connected to the bank's server, which is in charge of authorizing and tracking transactions. The bank and the ATM share a $k$-bit secret key $K$. When authorizing, the bank sends the ATM a triplet $(u, m, \mathsf{tag})$ allowing user $u$ to receive an amount $m$ in cash, and $\mathsf{tag}$ is a $n$-bit message authentication code (MAC) computed under key $K$ over the first two components of the triplet. (This is of course a very simplified view for the sake of this exercise.)

1. What is the bank trying to protect against, assuming an adversary who has full access to the network connecting the bank and the ATM?
2. Let us assume for now that the bank and the ATM use a secure MAC function. What is the minimum key length $k$ that is required to have a security strength of 128 bits, and why?
3. Let us further assume that the ATM processes not more than one triplet per second and that the secret key $K$ is securely renewed every 24 hours. What is the minimum length $n$ of the MAC such that the probability of fraudulent transaction is less than $10^{-12}$ per day, and why?

# Others

## Exercise 13

In order to save space or bandwidth we often use compression algorithms. Suppose you want to use data compression with encryption. How should these two operations be combined? Why?

## Exercise 14

Imagine that you have to use a relatively unreliable network i.e., packets often arrive with errors (bit flips) to their destination. Suppose you want to use an error correction code with encryption. How should these two operations be combined? Why?

## Exercise 15

It is possible to use a symmetric crypto system to encrypt a short message e.g., 40-bit and obtain a 40-bit ciphertext. Could we create a secure public key crypto system with short ciphertexts?

## Exercise 16

A company that installs and maintains an open-source operating system is creating an automatic update mechanism for its customers. Each computer connected to the internet can fetch an *update pack* containing the latest changes to be made to the operating system. The company would like to guarantee the integrity of these update packs so as to avoid the installation of malicious software on their customers' computers.

Please propose a solution based on schemes such as encryption, either public-key or secret-key and authentication, either MAC or signature. You also need to describe how the keys are distributed. Would a secret-key or a public-key approach be more appropriate?

## Exercise 17

Suppose Alice is broadcasting packets to 6 recipients $B_1, , B_6$. Privacy is not important but integrity is. In other words, each of $B_1, , B_6$ should be assured that the packets he is receiving were sent by Alice.

Alice decides to use a MAC. Suppose Alice and $B_1, , B_6$ all share a secret key k. Alice computes a tag for every packet she sends using key $k$. Each user $B_i$ verifies the tag when receiving the packet and drops the packet if the tag is invalid. Alice notices

that this scheme is insecure because user $B_1$ can use the key $k$ to send packets with a valid tag to users $B_2, , B_6$ and they will all be fooled into thinking that these packets are from Alice.

Instead, Alice sets up a set of 4 secret keys $S = \{k_1, , k_4\}$. She gives each user $B_i$ some subset $S_i \subseteq S$ of the keys. When Alice transmits a packet she appends 4 tags to it by computing the tag with each of her 4 keys. When user Bi receives a packet he accepts it as valid only if all tags corresponding to his keys in $S_i$ are valid. For example, if user $B_1$ is given keys $\{k_1, k_2\}$ he will accept an incoming packet only if the first and second tags are valid. Note that $B_1$ cannot validate the 3rd and 4th tags because he does not have $k_3$ or $k_4$.

How should Alice assign keys to the 6 users so that no single user can forge packets on behalf of Alice and fool some other user? How many keys are needed for $t$ users if each of them receive only two keys?