ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Family Name: ......................

First Name: ........................

Section: ............................

# Cryptography and Security Course

## (Crypto Part)

Midterm Exam

December 10th, 2004

Duration: 1 hour 45 minutes

This document consists of 11 pages.

### Instructions

Electronic devices are not allowed.

Answers must be written on the exercises sheet.

This exam contains 2 *independent* exercises.

Answers can be either in French or English.

Questions of any kind will certainly *not* be answered. Potential errors in these sheets are part of the exam.

You have to put your full name on *each* page.

## A Exhaustive Search on 3DES

We consider 3DES with three independent keys. Let $P, C \in \{0,1\}^{64}$ be a plaintext/ciphertext couple, where $C = \mathsf{3DES}_k(P)$ for some unknown key $k = (k_1, k_2, k_3)$ (see Figure 1). We want to recover $k$ by an exhaustive search.

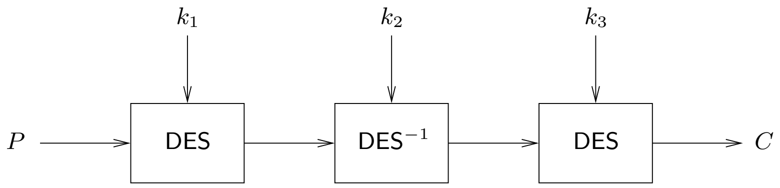


Figure 1: 3DES with three independent keys.

1. What is the total number of DES encryptions/decryptions of Algorithm 1?

---

**Algorithm 1** Exhaustive key search algorithm on 3DES

---

**Require:** A plaintext/ciphertext couple $(P, C)$

```
for each possible key K = (K_1, K_2, K_3) do
    if C = 3DES_K(P) then
        display K = (K_1, K_2, K_3)
    end if
end for
```

2. Let $\mathsf{C}^* : \{0,1\}^{64} \to \{0,1\}^{64}$ denote a uniformly distributed random permutation. What is the probability that $\mathsf{C}^*(P) = C$.

3. Assuming that $\mathsf{3DES}_K$ roughly behaves like $\mathsf{C}^*$ when $K$ is a uniformly distributed random key, estimate the amount of wrong keys (i.e., different from $k$) displayed by Algorithm 1.

4. Suppose you have $t$ distinct plaintext/ciphertext pairs, denoted $(P_i, C_i)$ for $i = 1, \ldots, t$, all encrypted under the same (still unknown) key $k$ (so that $C_i = \mathsf{3DES}_k(P_i)$). Write an algorithm similar to Algorithm 1 that reduces the number of wrong keys that are displayed (but which does at least display $k$). What is the total number of DES encryptions/decryptions of this algorithm?

5. Express the mean number of wrong keys that are displayed by your algorithm in function of $t$ (which is the number of available plaintext/ciphertext couples). Evaluate the necessary number of couples in order to be almost sure that *only* the good key $k = (k_1, k_2, k_3)$ is displayed.

# B Multicollisions on Hash Functions

## Preliminaries

In this problem, we will consider a cryptographic hash functions $h : \mathcal{M} \to \mathcal{H}$, where $\mathcal{M} = \{0,1\}^N$ and $\mathcal{H} = \{0,1\}^n$. We generalize the notion of collision to the one of $r$-collision. A $r$-collision on the cryptographic hash function $h : \mathcal{M} \to \mathcal{H}$ is a set of $r$ distinct messages $m_1, m_2, \ldots, m_r \in \mathcal{M}$ such that $h(m_1) = h(m_2) = \cdots = h(m_r)$. The aim of this problem is to study $r$-collisions first in the realistic case of iterated hash functions (for example hash functions based on the Merkle-Damgård construction), then in a more idealistic model, called the *Random Oracle Model* (where hash functions are replaced by random functions).

1. How many messages do we need to find a 2-collision with good chances by using the birthday paradox?

## Multicollisions in Iterated Hash Functions

We consider a hash function $h : \mathcal{M} \to \mathcal{H}$ based on the Merkle-Damgård scheme (see Figure 2). We denote by $f : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^n$ the compression function. Recall that in this construction the padding is mandatory and only depends on the length of the message. We

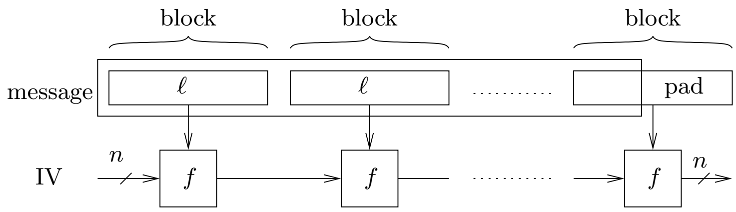


Figure 2: The Merkle-Damgård scheme

will suppose that $\ell \gg n$ (e.g. $\ell = 512$ and $n = 128$), i.e., the size of the message blocks is larger than the size of the hash.

2. Let $x$ be an arbitrary value in $\{0,1\}^n$. Using the birthday paradox, evaluate the number of necessary blocks in order to find two distinct blocks $B$ and $B'$ in $\{0,1\}^\ell$ such that $f(x, B) = f(x, B')$, and give the probability of success.

Let $h_0 : \{0,1\}^{c\times\ell} \to \mathcal{H}$ be a hash function similar to $h$, but without padding, for which the messages we consider have a fixed length $c \times \ell$.

3. Using the preceeding question, show how to find a 4-collision on $h_0$ with $c = 2$. Estimate the success probability.
   **Hint:** Use two (well chosen) 2-collision search on the compression function.

4. Explain how the 4-collision found on $h_0$ in the preceeding question leads to a 4-collision on $h$.

5. Explain how the preceeding idea can be generalized in order to find a $2^t$-collision on $h$ with only $t$ (well chosen) 2-collision searches on the compression function $f$.

6. Deduce from the preceeding questions the complexity (i.e., the total number of calls to $f$) of finding a $2^t$-collision on $h$ together with the probability of success.

## Multicollisions in the Random Oracle Model

In the *Random Oracle Model*, a hash function $H : \mathcal{M} \to \mathcal{H}$ is considered as a random function, uniformly distributed over all possible functions from $\mathcal{M}$ onto $\mathcal{H}$.

7. Let $m_1$ and $m_2$ be two *distinct* fixed elements of $\mathcal{M}$ and let $h_1$ and $h_2$ be two fixed elements of $\mathcal{H}$. Show that the events $H(m_1) = h_1$ and $H(m_2) = h_2$ are independent.

Consider a set of $q$ distinct messages $m_1, m_2, \ldots, m_q$ of $\mathcal{M}$. Thanks to the preceeding questions, we can consider $H(m_1), H(m_2), \ldots, H(m_q)$ as a set of $q$ independent random variables (that we will denote $H_1, H_2, \ldots, H_q$) uniformly distributed in $\mathcal{H}$. We assume the following lemma.

**Lemma 1.** *Let $\mathcal{H} = \{0,1\}^n$. Let $\{H_1, \ldots, H_q\}$ be a set of $q$ independent uniformly distributed random variables of $\mathcal{H}$, where $q < 2^{n-8}$. Let us call $r$-coincidence an element of $\mathcal{H}$ which occurs exactly $r$ times in the sequence $H_1, \ldots, H_q$. Let $\lambda$ be such that $q = (\lambda r!)^{1/r} 2^{n(r-1)/r}$. If $\lambda \leq 1$, then the probability that there is* no *$s$-coincidence for any $s \geq r$ is close to $e^{-\lambda}$.*

8. Using Lemma 1, compute the probability that there is no $s$-coincidence for any $s \geq 2$ in the sequence $H_1, \ldots, H_q$ and use it to prove the birthday paradox (when $n$ is large enough).

9. Compute the number $q$ of distinct messages that are necessary to obtain an $r$-collision with probability $1 - e^{-1/2}$.

10. Show that $q$ is lower-bounded by $2^{96}$ when $r = 4$ and $n = 128$. For a similar probability of success, show that the complexity of finding a 4-collision when $h$ is an iterated hash function is much smaller.

11. Compare the results of questions 6 and 9. Conclude.