ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Family Name: ....

First Name: ....

Section: ....

Advanced Cryptography

Midterm Exam
Solutions

May 19th, 2006

Duration: 2 hours 30 minutes

This document consists of 12 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 and you have to do it *now*.

1 Attack on a simple Feistel Scheme

Let C be the block cipher that consists of the 2-round Feistel scheme of Figure 1. The plaintext is denoted by x and the output ciphertext by y.

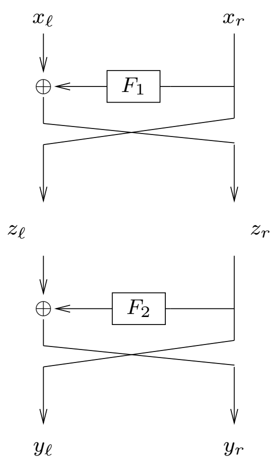
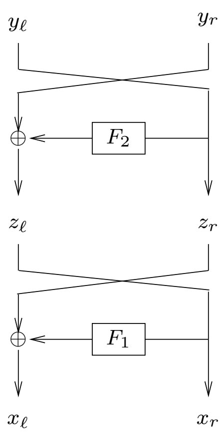


Figure 1: 2-round Feistel scheme.

We use the notation x_ℓ, y_ℓ (resp. x_r, y_r) for the plaintext/ciphertext on the left (resp. right) leave, i.e., $x = x_\ell \| x_r$ and $y = y_\ell \| y_r$ where the operator "$\|$" denotes the concatenation.

1. Draw the inverse scheme for the Feistel scheme of Figure 1.



Make care that F_i^{-1} perhaps do not exist.

We consider that the functions F_i simply performs a xor between the input and a subkey. We denote by k_1 the subkey of the first round, and by K_2 the subkey of the second round. Consequently, we have $F_i(\alpha) = \alpha \oplus k_i$.

2. Express z_ℓ, z_r in term of x_ℓ, x_r, k_1.

$$z_\ell = x_r$$
$$z_r = x_\ell \oplus x_r \oplus k_1$$

3. Express y_ℓ, y_r in term of x_ℓ, x_r, k_1, k_2.

$$y_\ell = x_\ell \oplus x_r \oplus k_1$$
$$y_r = x_\ell \oplus k_1 \oplus k_2$$

4. Compute the differential coefficient $\mathrm{DP}^C(a, b)$ for any fixed (unknown) a, b.

$$\begin{array}{rcl} \mathrm{DP}^C(a,b) & = & \Pr[C(X \oplus a) = C(X) \oplus b] \\ & = & \Pr[(x_\ell \oplus a_\ell \oplus x_r \oplus a_r \oplus k_1) \| (x_\ell \oplus a_\ell \oplus k_1 \oplus k_2) \\ & & \quad = (x_\ell \oplus x_r \oplus k_1 \oplus b_\ell) \| (x_\ell \oplus k_1 \oplus k_2 \oplus b_r)] \\ & = & \Pr[a_\ell \oplus a_r = b_\ell \text{ and } a_\ell = b_r] \\ & = & \begin{cases} 1 & \text{when } a_\ell \oplus a_r = b_\ell \text{ and } a_\ell = b_r \\ 0 & \text{otherwise} \end{cases} \end{array}$$

5. Consider $x_\ell, y_\ell, k_i \in \{0,1\}$. Compute $[C]^1$ the distribution matrix of C at order 1.

First, you compute the ouptuts with respect to the input and the key:

	$C(00)$	$C(01)$	$C(10)$	$C(11)$
$k_1 \| k_2 = 00$	00	10	11	01
$k_1 \| k_2 = 01$	01	11	10	00
$k_1 \| k_2 = 10$	11	01	00	10
$k_1 \| k_2 = 11$	10	00	01	11

You finally obtain the following probabilities:

	$y = 00$	$y = 01$	$y = 10$	$y = 11$
$x = 00$	1/4	1/4	1/4	1/4
$x = 01$	1/4	1/4	1/4	1/4
$x = 10$	1/4	1/4	1/4	1/4
$x = 11$	1/4	1/4	1/4	1/4

6. Is the cipher C a markov cipher? Justify your answer.

Yes, it is a Markov Cipher.
The definition is the following, if $\mathrm{DP}_x^C(a,b) = E_X(\mathrm{DP}(a,b)]$ then C is a Markov cipher. Using the response of point 4, it is straightforward.

7. Does C provide perfect secrecy if it is used only once? Justify your answer.

Yes, it provides perfect secrecy.
From point 5, we note that $[C]^1$ is equal to $[C^*]^1$, which implies that $\mathrm{Dec}^1[C] = 0$ and thus C provides perfect secrecy.

8. Using two queries, define an efficient distinguisher between C and the perfect cipher C^*. Compute its advantage.

We can use the work done at point 4. In fact we are running a differential distinguisher.
Let k the number of bits per block of the cipher C.

(a) pick $x, a \in \{0,1\}^k$

(b) submit x to the encryption oracle, i.e. $y_1 \leftarrow C(x)$

(c) submit $x + a$ to the encryption oracle, i.e. $y_2 \leftarrow C(x + a)$

(d) if $y_1 \oplus y_2 = (a_\ell + a_r) \| a_\ell$

    $\rightarrow$ output 1

(e) else

    $\rightarrow$ output 0

Here, we have an input difference of $a = a_\ell \oplus a_r$,

- if the encryption oracle implements C, we always have an output difference of $b = (a_\ell + a_r) \| a_\ell$ (see point 4), i.e. the probability is 1.
- if the encryption oracle implements C^*, we have this difference with probability 2^{-k}.

Thus, the advantage is $1 - 2^{-k}$ considering a k-bit C.

2 GCM: the Galois Counter authenticated encryption Mode

We consider 128-bit strings as elements of the Galois field $\mathrm{GF}(2^{128})$ so that the addition corresponds to the bitwise XOR operation denoted "$\oplus$" and the multiplication is denoted by "$\cdot$". We assume we have a conventional choice for the representation of the Galois field.

We consider a keyed hash function which, given a bitstring X of bitlength multiple of 128 and a 128-bit key H defines

$$\mathsf{GHASH}_H(X) = X_1 \cdot H^m \oplus \cdots \oplus X_m \cdot H$$

where $X = X_1 || \cdots || X_m$ is the decomposition of X into m blocks of 128 bits.

1. Assuming that H is uniformly distributed, show that for any m, GHASH_H is a $m2^{-128}$-XOR-universal hash function from the set of bitstrings of length up to $128m$ to the set of 128-bit strings.

   Recall: an ε-XOR-universal hash function h_K is a family of functions depending on some parameter K such that for any different x and y and any δ, we have

   $$\Pr_K[h_K(x) \oplus h_K(y) = \delta] \leq \varepsilon$$

   when K is uniformly distributed.

First, note that

$$\Pr[H(x) \oplus H(y) = a] = \Pr[(X_1 \oplus Y_1)H^m \oplus \cdots \oplus (X_m \oplus Y_m)H = a]$$

We see that we have a polynom of degree m. Such a polynom have *at most* m solutions, but there is 2^{128} possibilities for H.
Thus, the above probability is at most $\frac{m}{2^{128}}$ and we conclude that we have a $m2^{-128}$-XOR-universal hash function.

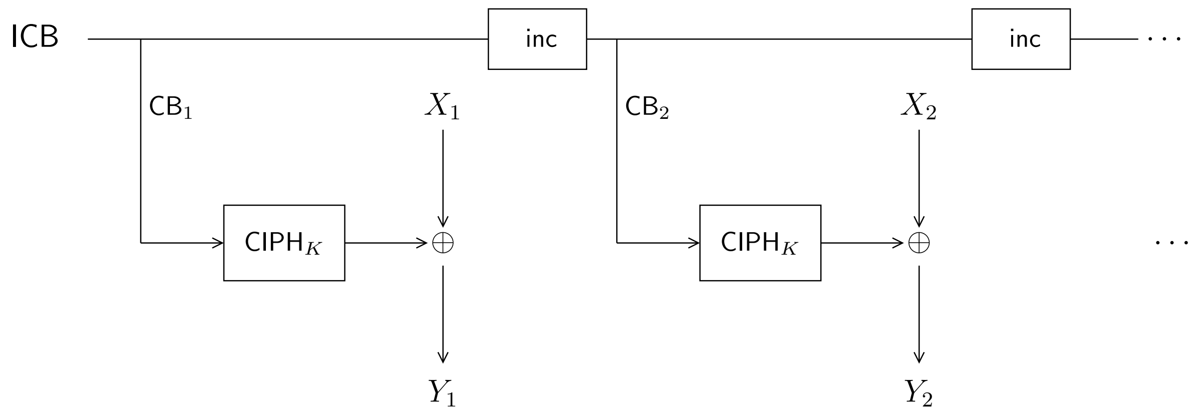


Figure 2: $\mathsf{GCTR}_K(ICB, X)$

Given a 128-bit string X, we define X_H of 96 bits and X_L of 32 bits such that

$$X = X_H \| X_L.$$

The function $\mathsf{inc}(X)$ is defined as follows:

$$\mathsf{inc}(X) = X_H \| X'_L$$

where $X'_L = X_L + 1 \bmod 2^{32}$ and X_L is considered as an integer.

We consider a block cipher with 128-bit blocks which, given a block x and a key K defines a ciphertext block $\mathsf{CIPH}_K(x)$. We define $\mathsf{GCTR}_K(\mathsf{ICB}, X)$ (see Figure 2), the encryption of an arbitrary nonempty bitstring X by key K in CTR mode with initial counter block ICB by

$$\mathsf{GCTR}_K(\mathsf{ICB}, X) = Y_1 || \cdots || Y_{n-1} || Y_n$$

with

$$Y_i = X_i \oplus \mathsf{CIPH}_K\left(\mathsf{CB}_i\right), \qquad X = X_1 || \cdots || X_{n-1} || X_n$$

and

$$\mathsf{CB}_i = \mathsf{inc}(\mathsf{CB}_{i-1}), \qquad \mathsf{CB}_1 = \mathsf{ICB}$$

where X_i are 128-bit blocks and X_n is a nonempty string of length at most 128.
When X is of length 0, $\mathsf{GCTR}_K(\mathsf{ICB}, X)$ is the empty string.

2. Show that the length of the ciphertext and the length of the plaintext are the same.

We note that Y_i is the result of a xor between X_i and a random value. If X_i is non empty, then Y_i in non-empty too and when X_i is empty Y_i is empty. Thus, they have the same length.

3. Let ICB and ICB' be two possible values for the initial counter block.
The ith counter block of ICB (resp. ICB') is denoted CB_i (resp. CB'_i).
Let X and X' be two arbitrary plaintexts of length (in blocks) less than 2^{32}.
The ith block of X (resp. X') is denoted X_i (resp. X'_i).
Let Y and Y' be the ciphertexts, i.e. $Y = \mathsf{GCTR}_K(\mathsf{ICB}, X)$ and $Y' = \mathsf{GCTR}_K(\mathsf{ICB}', X')$.
The ith block of Y (resp. Y') is denoted Y_i (resp. Y'_i).

   What can happen if there exists i, j such that $\mathsf{CB}_i = \mathsf{CB}'_j$?

> We have
> $$X_i = Y_i \oplus \mathsf{CIPH}_K(CB_i)$$
> $$X'_j = Y'_j \oplus \mathsf{CIPH}_K(CB'_j)$$
> Thus,
> $$X_i \oplus X'_i = Y_j \oplus Y'_j$$
> Suppose we know X_i, Y_i since there are encrypted by us. If you find Y'_j you can decrypt, i.e.
> $$X'_i = X_i \oplus Y_j \oplus Y'_j$$
>
> You can also note that $\mathsf{CB}_{i+k} = \mathsf{CB}'_{j+k}$ for any $k = 0, 1, 2, \dots$ since
> $$\mathsf{CB}_{i+1} = \mathsf{inc}(\mathsf{CB}_i)$$
> $$\mathsf{CB}'_{j+1} = \mathsf{inc}(\mathsf{CB}'_j)$$
> Thus, you can decrypt the rest of the conversation.

We now assume that all ICB values are pairwise different and such that the 32 least significant bits consist of a fixed block b. Namely, we have $\mathsf{ICB} = \mathsf{IV}||b$ where IV is a nounce.

4. Show that for all i, j we have $\mathsf{CB}_i \neq \mathsf{CB}'_j$.

> $$\begin{aligned} \Pr[\mathsf{CB}_i = \mathsf{CB}'_j] &= \Pr[\mathsf{IV}\|b_i = \mathsf{IV}'\|b'_i] \\ &= \Pr[\mathsf{IV} = \mathsf{IV}' \text{ and } b_i = b'_i] \\ &= 0 \end{aligned}$$

5. Assuming there exists i, j such that $Y_i = Y'_j$, deduce some mutual information on X_i and X'_j.

Let
$$Y_i = X_i \oplus \mathsf{CIPH}_k(\mathsf{CB}_i)$$
and
$$Y'_j = X'_j \oplus \mathsf{CIPH}_k(\mathsf{CB}'_j)$$
If $Y_i = Y'_j$, then we have
$$X_i \oplus X'_j = \mathsf{CIPH}_k(\mathsf{CB}_i) \oplus \mathsf{CIPH}_k(\mathsf{CB}'_j)$$

6. We assume a model where an adversary can submit chosen plaintexts and receive a fresh ICB together with the corresponding ciphertext in return. Deduce a distinguisher which can submit a total number within the order of magnitude of 2^{64} blocks of plaintext and have an advantage within the order of magnitude of $\frac{1}{2}$.

We can pick plaintexts X_i and submit them to an oracle which returns the corresponding Y_i together with the IV_i.
Using the previous result, we see that if $\mathsf{CB}_i = \mathsf{CB}'_j$, we have $X_i \oplus X'_j = Y_i \oplus Y'_j$.
We submit queries to the oracle until we have $\mathsf{ICB}_i = \mathsf{ICB}_j$. Then if $X_i \oplus X'_j = Y_i \oplus Y'_j$, we ouptut 1, else we output 0.

The probability of success of such a distinguisher is approximatively $1 - e^{-\frac{2^{64}}{\sqrt{2^{128}}}}$ (using the birthday paradox).

We define two algorithms

authenticated encryption: $\mathsf{ENC}_K(P, A, \mathsf{IV}) = (C, T)$
given a plaintext P, and additional authenticated data A, and an initialization vector IV (to be used as a nounce), computes a ciphertext C and a t-bit tag T

authenticated decryption: $\mathsf{DEC}_K(\mathsf{IV}, A, C, T) = P$ (or fail)
given the initial vector, the additional authenticated data A, the ciphertext C, and the tag T, authenticates A and C and recovers the plaintext P or tell that A and C are not authenticated.

For simplicity, we assume that the length of C is multiple of 8 and that IV is of 96 bits.

Given a bitstring x of length at most 128, we define $\mathsf{pad}(x)$ the string x concatenated with enough zero bits to reach a full block length.

The authenticated encryption is defined by first letting H be the encrypted block by CIPH_K of the all-zero block, letting $J_0 = \mathsf{IV}||b$, letting $C = \mathsf{GCTR}_K(\mathsf{inc}(J_0), P)$, letting $S = \mathsf{GHASH}_H(\mathsf{pad}(A)||\mathsf{pad}(C)||\ell_A||\ell_C)$ where ℓ_A and ℓ_C are the bitlength of A and C respectively, and letting T be the t most significant bits of $\mathsf{GCTR}_K(J_0, S)$.

7. Define the authenticated decryption algorithm.

Encryption, input P, A, IV :

(a) $H \leftarrow \mathsf{CIPH}_K(000\cdots 0)$

(b) $J_0 \leftarrow \mathsf{IV}\|b$

(c) $C \leftarrow \mathsf{GCTR}_K(\mathsf{inc}(J_0), P)$ which is equal to $P \oplus \mathsf{inc}(J_0)$

(d) $S \leftarrow \mathsf{GHASH}_H(\mathsf{pad}(A)\|\mathsf{pad}(C)\|\ell_a\|\ell_c)$

(e) $T \leftarrow MSB_t[\mathsf{GCTR}_K(J_0, S)]$

(f) output (C, T)

Decryption, input A, IV, C, T :

(a) $H \leftarrow \mathsf{CIPH}_K(000\cdots 0)$

(b) $J_0 \leftarrow \mathsf{IV}\|b$

(c) $S \leftarrow \mathsf{GHASH}_H(\mathsf{pad}(A)\|\mathsf{pad}(C)\|\ell_a\|\ell_c)$

(d) if $T = MSB_t[\mathsf{GCTR}_K(J_0, S)]$

$\rightarrow$ output $P \leftarrow \mathsf{GCTR}_K(\mathsf{inc}(J_0), C)$
which is equal to $C \oplus \mathsf{inc}(J_0) = (P \oplus \mathsf{inc}(J_0)) \oplus \mathsf{inc}(J_0)$

(e) else

$\rightarrow$ output *fail*

8. We define $\mathsf{GMAC}_K(A, \mathsf{IV}) = T$ for T such that there exists C such that $\mathsf{ENC}_K(\emptyset, A, \mathsf{IV}) = (C, T)$ where $\emptyset$ denotes a string of length zero.

   What kind of cryptographic primitive do we obtain?

   A message authentication code

9. Let H be as defined in the authenticated encryption. Let IV^i, $i = 1, \ldots, n$ be n arbitrary pairwise different initial vectors. They define J_0^i, $i = 1, \ldots, n$.

   Assuming that CIPH_K behaves like a perfect random function (PRF) when K is random, show that for any pairwise different $h, j_1, \ldots, j_n$ we have $\Pr[H = h, J_0^i = j_i; i = 1, \ldots, n] = 2^{-128(n+1)}$.

$$
\begin{aligned}
\Pr[H = h, \forall i = 1..n : J_0^i = j_i] &\overset{\text{indep}}{=} \Pr[H = h] \cdot \prod_{i=i}^{n} \Pr[J_0 = j_i] \\
&= \Pr[\mathsf{CIPH}_K(000\cdots 0) = h] \cdot \prod_{i=i}^{n} 2^{-128} \\
&= \Pr[\mathsf{CIPH}_K(000\cdots 0) = h] \cdot 2^{-128n} \\
&\overset{\mathsf{CIPH}_K \approx \text{PRF}}{=} 2^{-128(n+1)}
\end{aligned}
$$

10. Write how T is obtained depending on t, H, J_0, and A by using only the pad and GHASH functions.

   $\mathsf{ENC}_K(\emptyset, A, \mathsf{IV})$ implies that

$$
\begin{aligned}
T &= MSB_t\left(\mathsf{GCTR}\left(J_0, S\right)\right) \\
&= MSB_t\left(\mathsf{GCTR}_K\left(J_0, \mathsf{GHASH}_H\left(\mathsf{pad}(A) \| \emptyset \| \ell_A \| 0\right)\right)\right) \\
&= MSB_t\left(\mathsf{CIPH}_K\left(J_0\right) \oplus \mathsf{GHASH}_H\left(\mathsf{pad}(A) \| \ell_A \| 0\right)\right)
\end{aligned}
$$

11. We assume a model where the adversary can choose values for A and get a fresh IV and a 128-bit $T = \mathsf{GMAC}_K(A, \mathsf{IV})$ in return (i.e. $t = 128$). The goal of the adversary is to output an A that was not submitted together with any IV and the right value for $\mathsf{GMAC}_K(A, \mathsf{IV})$.

    Assuming that CIPH_K behaves like a perfect random function when K is random, show that the success probability of the adversary limited to n queries is upper bounded by $(m + 1)2^{-128}$.

    Hint: consider the case where the adversary outputs a fresh IV and the case where she reuses a received one. In the former case, show that the probability of success is bounded by 2^{-128}. In the latter case, show that it is bounded by $m2^{-128}$ where m is the maximum length in blocks of a value A.

Here, you can make queries to an oracle with input A and you receive responses of the form IV, T. Your objective is to output a valid triplet (A, IV, T) which was not generated by the oracle.
We describe the attack as following

(a) for $i = 1$ to n loop
   i. select A_i
   ii. submit A_i to the oracle, i.e. you obtain $\mathsf{IV_i}, T_i$

You have know a list of n elements of the form $(A_i, \mathsf{IV}_i, T_i)$ and you are searching to build a $n + 1$ element.
Here we distiguish two cases:

Reuse an IV: If you reuse an IV, others IV's are not useful for you. In short, you are looking for a $\hat{A}$ such that

$$\mathsf{GHASH}_H(\mathsf{pad}(A)\|\ell_A\|0) = \mathsf{GHASH}_H(\mathsf{pad}(\hat{A})\|\ell_{\hat{A}}\|0)$$

and thus your are trying to find a collision on GHASH. From point 1, we deduce that this occurs with probability at most $m2^{-128}$.

Fresh IV: In this case you can use no previous query. Here, you are looking for a pair $\hat{\mathsf{IV}}, \hat{A}$ such that

$$\mathsf{CIPH}_K\left(\mathsf{IV}\|b\right) \oplus \mathsf{GHASH}_H\left(\mathsf{pad}(A)\|\ell_A\|0\right) = \mathsf{CIPH}_K\left(\hat{\mathsf{IV}}\|b\right) \oplus \mathsf{GHASH}_H\left(\mathsf{pad}(\hat{A})\|\ell_{\hat{A}}\|0\right)$$

and this occurs with probability 2^{-128}.

We finally have

$$\begin{aligned}
\Pr[\text{success}] &= \Pr[\text{success}|\text{fresh IV}] \cdot \Pr[\text{fresh IV}] + \Pr[\text{success}|\text{reuse IV}] \cdot \Pr[\text{ruse IV}] \\
&\leq \Pr[\text{success}|\text{fresh IV}] + \Pr[\text{success}|\text{reuse IV}] \\
&\leq (m + 1)2^{-128}
\end{aligned}$$

Note: this exercise is inspired by publication NIST SP 800-38D, April 2006.