Laboratory session 1

Implementation and linear cryptanalysis of a simplified AES-like cipher

Nicola Laurenti, Laura Crosara March 31, 2025

Laboratory session 1— Contents

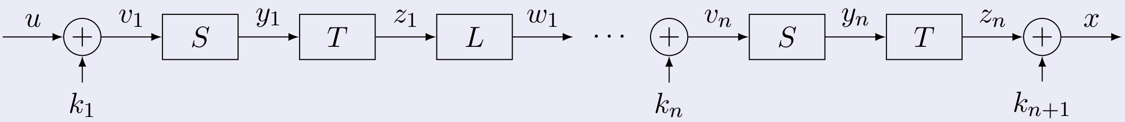
Simplified AES-like cipher

It is a (S, T, L) iterated cipher with $n = 5$ rounds and

$$\mathcal{M} = \mathcal{X} = \mathcal{K} = \mathbb{F}^{\ell} \quad , \quad \mathbb{F} = \mathrm{GF}(p) \quad , \quad \ell_x = \ell_u = \ell_k = \ell = 8 \quad , \quad p = 11$$

$$\mathcal{K}' = \mathbb{F}^{\ell_{k'}} \quad , \quad \ell_{k'} = 4$$

Encryption



Decryption is performed by the inverse blocks in reverse order with inverse key sequence.

Simplified AES-like cipher

subkey generation

$$k \in \mathcal{K} \quad , \quad k_i \in \mathcal{K}' \quad , \quad g : \mathcal{K} \mapsto (\mathcal{K}')^{n+1}$$

Each subkey k_i, $i \in \{1, \ldots, n+1\}$, is defined as

$$\begin{aligned} k_1 &= [k(1), k(3), k(5), k(7)] \quad , \quad & k_2 &= [k(1), k(2), k(3), k(4)] \\ k_3 &= [k(1), k(4), k(5), k(8)] \quad , \quad & k_4 &= [k(1), k(4), k(6), k(7)] \\ k_5 &= [k(1), k(3), k(6), k(8)] \quad , \quad & k_6 &= [k(3), k(4), k(5), k(6)] \end{aligned}$$

where $k(j)$ stands for the j-th symbol of the key $k \in \mathcal{K}$.

subkey sum

$$v_i = w_{i-1} + [k_i, k_i] \mod p$$

where $w_0 = u$

Simplified AES-like cipher

substitution

$$v_i = (v_i(1), \ldots, v_i(\ell)) \quad , \quad y_i = (y_i(1), \ldots, y_i(\ell)) \quad , \quad y_i(j) = f(v_i(j))$$

transposition flip the second half of vector y_i, that is

$$z_i = [y_i(1), \ldots, y_i(4), y_i(8), \ldots, y_i(5)]$$

linear write (by rows) vector z_i to 2×4 matrix Z_i, then

$$W_i = \begin{bmatrix} 2 & 5 \\ 1 & 7 \end{bmatrix} Z_i \mod p$$

Read matrix W_i by rows into vector w_i.

Implement a simple AES-like encryptor

Task 1

Using a programming language of your choice, implement the encryptor for a simplified AES-like cipher with the parameters given in the previous slides and the following substitution function:

$$f \quad : \quad y_i(j) = 2v_i(j) \bmod p \quad , \qquad j \in \{1, \ldots, \ell\}$$

with all operations in the field $\mathbb{F} = \mathrm{GF}(p)$.

Check that your implementation is correct by verifying that the encryption of $u = [1, 0, \ldots, 0]$ with the key $k = [1, 0, \ldots, 0]$ is $x = [4, 0, 0, 9, 7, 0, 0, 3]$.

Implement a simple AES-like decryptor

Task 2

Implement the decryptor for this simplified AES-like cipher. Note that decryption is performed by the inverse blocks in reverse order. Therefore, you have to implement the inverse of each function used to encrypt the message (subkey sum, substitution, transposition and linear), taking into consideration that all the operations must be done in the field $\mathbb{F} = \mathrm{GF}(p)$.

Check that your implementation is correct by verifying that by concatenating encryption and decryption with the same key k you retrieve the original plaintext u. Experiment with different (u, k) pairs.

Identify the cipher vulnerability

Observe that

- the substitution function $f(\cdot)$ is linear in the message block
- the subkey generation function $g(\cdot)$ is linear in the key

and conclude that the cipher is linear

Task 3

Identify the overall linear relationship for this simplified AES-like cipher, that is find the matrices $A \in \mathbb{F}^{\ell_x \times \ell_k}$ and $B \in \mathbb{F}^{\ell_x \times \ell_u}$ such that

$$x = E(k, u) = Ak + Bu \bmod p$$

with all operations in the field $\mathbb{F} = \mathrm{GF}(p)$.
(if you do not know how to identify a linear system in a black box model, see Appendix 1)

Carry out linear cryptanalysis

Task 4

From a known plaintext/ciphertext pair (u, x), implement a linear cryptanalysis KPA against this cipher by computing

$$k = A^{-1}(x - Bu) \bmod p$$

with all operations in the field $\mathbb{F} = \mathrm{GF}(p)$ (if you do not know how to compute A^{-1}, the modular inverse of A, see Appendix 2).

You will find a few plaintext/ciphertext pairs, all encrypted with the same key k in a file labeled `KPApairsXxxxxx_linear.txt` in the folder `KPAdataXxxxxx`, where `Xxxxxx` is your team's name. Find the value of the key k.

"Nearly linear" simplified AES-like cipher

Task 5

implement the encryptor for a simplified AES-like cipher with the parameters given in the previous slides and the substitution function described by the following table:

$v_i(j)$	0	1	2	3	4	5	6	7	8	9	10
$y_i(j)$	0	2	4	8	6	10	1	3	5	7	9

where $j \in \{1, \ldots, \ell\}$.

Check that your implementation is correct by verifying that the encryption of $u = [1, 0, \ldots, 0]$ with the key $k = [1, 0, \ldots, 0]$ is $x = [9, 0, 0, 0, 5, 0, 0, 6]$.

Linear cryptanalysis of a “nearly linear” cipher

Task 6

Find a linear approximation of the cipher in Task 5, that is, find matrices $A \in \mathbb{F}^{\ell_x \times \ell_k}$, $B \in \mathbb{F}^{\ell_x \times \ell_u}$ and $C \in \mathbb{F}^{\ell_x \times \ell_x}$ (it might possibly be $C = I$), such that

$$\mathrm{P}\left[Ak + Bu + Cx \bmod p = 0\right] \gg \frac{1}{p^{\ell_x}}$$

and evaluate the above probability by numerical simulation.
From a few known plaintext/ciphertext pair (u, x), implement a linear cryptanalysis KPA against this cipher by computing

$$k = A^{-1}(Cx - Bu) \bmod p$$

and then explore “close” key values to find the key that encrypts u to x exactly.

You will find a few plaintext/ciphertext pairs, all encrypted with the same key k in a file labeled `KPApairsXxxxxx_nearly_linear.txt` in the folder `KPAdataXxxxxx`, where `Xxxxxx` is your team's name. Guess the value of the key k.

Non linear simplified AES-like cipher

Task 7

implement the encryptor for a simplified AES-like cipher with the following parameters:

$$\mathcal{K} = \mathbb{F}^{\ell_k} \quad , \quad \ell_k = 4$$

subkey generation Each subkey k_i, $i \in \{1, \dots, n+1\}$, is defined as

$$k_1 = [k(1), k(2), k(3), k(4)]; \;\; k_2 = [k(1), k(2), k(4), k(3)]; \;\; k_3 = [k(2), k(3), k(4), k(1)];$$
$$k_4 = [k(1), k(4), k(2), k(3)]; \;\; k_5 = [k(3), k(4), k(1), k(2)]; \;\; k_6 = [k(2), k(4), k(1), k(3)]$$

substitution Considering that inverse of $v_i(j)$ is computed in the field $\mathbb{F} = \mathrm{GF}(p)$:

$$f \;\; : \;\; y_i(j) = 2v_i(j)^{-1} \bmod p \quad , \quad j \in \{1, \dots, \ell\}$$

(the remaining parameters are as described before)

Check that your implementation is correct by verifying that the encryption of $u = [1, 0, \dots, 0]$ with the key $k = [1, 0, 0, 0]$ is $x = [5, 0, 3, 2, 5, 2, 1, 1]$.

Meet in the middle attack

Task 8

Implement a "meet-in-the-middle" attack see Appendix 3 against the concatenation of two instances of the non linear simplified AES-like cipher defined in Task 7, with different keys k', k'', respectively.

You will find a few plaintext/ciphertext pairs, all encrypted with the same concatenated cipher, and the same pair of keys k', k'' in a file labeled `KPApairsXxxxxx_non_linear.txt` in the folder `KPAdataXxxxxx`, where `Xxxxxx` is your team's name. Guess the values of the keys k', k''.

What you need to turn in

Each team must turn in, through the Moodle assignment submission procedure:

1. the code for your implementation (either as a single file, many separate files, or a compressed folder)
2. a short report (1-3 pages) in a graphics format (PDF, DJVU or PostScript are ok; Word, TEX or LATEX source are not), including:
   - 2.1 a brief description of your implementations for Tasks 1-8, explaining your choices;
   - 2.2 the results of your cryptanalysis effort:
     - 2.2.1 the matrices A and B that you used in Task 3;
     - 2.2.2 your guess $\hat{k}$ for the key we used to encrypt the KPA pairs in Task 4
     - 2.2.3 the matrices A, B and C that you used in Task 5, and an estimate value for the corresponding probability $\mathrm{P}\left[Ak + Bu + Cx = 0\right]$;
     - 2.2.4 your guess $\hat{k}$ for the key we used to encrypt the KPA pairs in Task 6
     - 2.2.5 your guesses $\hat{k}', \hat{k}''$ for the keys we used to encrypt the KPA pairs in Task 8

Appendix 1: identifying a linear system

A general linear system, $y = Au$, with input u and output y can always be identified in a black box approach, by feeding it as inputs the vectors of the standard orthonormal basis

$$e_1 = [100\ldots0] \quad , \quad e_2 = [010\ldots0] \quad , \quad \cdots \quad , \quad e_\ell = [000\ldots01]$$

and observing the corresponding outputs.

In fact, by choosing a sequence of inputs $u_1, \ldots, u_\ell$ such that $u_j = e_j$, and observing the corresponding outputs y_j we obtain that $y_j = Ae_j$ is the j-th column of matrix A.

In our case there are two inputs, the plaintext and the key. By encrypting $(e_1, \ldots, e_\ell)$ and the all-zero vector 0 you can obtain each column a_j of the matrix A and each column b_j of matrix B, as

$$\begin{aligned} k = e_j, u = 0 \quad &\Rightarrow \quad x = E(e_j, 0) = Ae_j + B0 = a_j \quad , \quad j = 1, \ldots, \ell_k \\ k = 0, u = e_j \quad &\Rightarrow \quad x = E(0, e_j) = A0 + Be_j = b_j \quad , \quad j = 1, \ldots, \ell_u \end{aligned}$$

Appendix 2: computing the inverse of a binary matrix

The inverse of a square matrix A in the field $\mathbb{F} = \mathrm{GF}(p)$ is the matrix A^{-1} given by

$$A^{-1} = \tilde{A} \cdot \det(A)^{-1} \bmod p$$
$$\tilde{A} = A^{*} \cdot \det(A)$$

where A^{*} and $\det(A)$ are the inverse and the determinant of A in the real field $\mathbb{R}$ and $\det(A)^{-1}$ is the multiplicative inverse of the determinant in the field $\mathbb{F} = \mathrm{GF}(p)$. So, $\tilde{A}$ is an integer matrix.

Example

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad , \quad \det(A) = -2 \quad , \quad \det(A)^{-1} = 5 \quad , \quad A^{-1} = \begin{bmatrix} 9 & 1 \\ 7 & 5 \end{bmatrix}$$

Appendix 3: “meet in the middle” attack

This is a KPA against a concatenated cipher (see slides), where $x = E''_{k''}(E'_{k'}(u))$ It consists in trying N' distinct guesses for $k' \in \mathcal{K}'$, and N'' distinct guesses for $k'' \in \mathcal{K}''$, with a complexity significantly lower than the product $N'N''$. Given a known plaintext/ciphertext pair (u, x)

1. Generate $N' \leq |\mathcal{K}'|$ random guesses of k', $\hat{k}'_1, \ldots \hat{k}'_{N'}$
2. For each guess $\hat{k}'_i$ compute the corresponding cipher guess $\hat{x}'_i = E'_{\hat{k}'_i}(u)$
3. Sort the table with key and cipher guesses, according to $\hat{x}'_i$
4. Generate $N'' \leq |\mathcal{K}''|$ random guesses of k'', $\hat{k}''_1, \ldots \hat{k}''_{N''}$
5. For each guess $\hat{k}''_i$ compute the corresponding plaintext guess $\hat{u}''_i = D''_{\hat{k}''_i}(x)$
6. Sort the table with key and cipher guesses, according to $\hat{u}''_i$
7. Search for a match between the two sorted tables, that is a pair of guesses $(\hat{k}'_i, \hat{k}''_j)$ such that $x'_i = u'_j$. Then, $\hat{k}' = \hat{k}'_i$ and $\hat{k}'' = \hat{k}''_j$ will be your final guess

If you get several matches you can increase the attack success probability with more KPA pairs

