Implementation - Rypto
April 10, 2017

Table of Contents

Preface

"Tietorakenteet ja algoritmit" – excercise.

Rypto is a software, which can encrypt and decrypt.

General structure

Software is comprised of the following main components:

- Library aes – all of the AES-related functionality
- Executable rypto – user interface
- Test cases – CUnit format

Tool chain

To build the project, the following are needed:

- gradle tool
- C compiler with standard libraries

Development environment was MacBook Pro, running macOS Sierra Version 10.12.3. The gradle tool and gcc C compiler were installed from Homebrew.

Library aes

In the library, the helper functions are also visible to facilitate unit testing. All exported symbols begin with AES_ .

The source code of the library is in two files: aes.c and aes.h.

If the library is used in a source file, then file aes.h must be included.

The constants (like AES_S_Box array) were extracted from[FIPS197], unless otherwise mentioned.

The Galois multiplication table AES_g_m was extracted from[WIKI001].

The following implementations were viewed before starting implementation: [GITHUB01], [GITHUB02], [CONTE01].

Types

Two types are defined:

- AES_byte (8-bit unsigned integer)
- AES_word (32-bit unsigned integer)

Functions

AES_KeyExpansion

```
void AES_KeyExpansion(AES_byte *key,
                      AES_word *w)
```

Expands a given key (128 bits, 16 bytes) to an AES Key Schedule (11 x 4 words). Must be done before encryption or decryption.

AES_encrypt

```
void AES_encrypt(AES_byte *plaintext,
                 AES_byte *ciphertext,
                 AES_word *w)
```

Encrypts one block (16 bytes) with an AES Key Schedule.

AES_decrypt

```
void AES_decrypt(AES_byte *plaintext,
                 AES_byte *ciphertext,
                 AES_word *w)
```

Decrypts one block (16 bytes) with an AES Key Schedule.

Performance

Space efficiency: Used space is constant.

Time efficiency: O(N)

Missing features, possible new features

Other key sizes than 128. Other operation modes besides ECB.

References

FIPS197: U.S. Department of Commerce/National Institute of Standards and Technology, Federal Information Processing Standard, FIPS PUB 197 Advanced Encryption Standard (AES), 2001
WIKI001: Wikipedia, Rijndael mix columns, 2017, https://en.wikipedia.org/wiki/Rijndael_mix_columns
GITHUB01: kokke, Small portable AES128 in C , 2017, https://github.com/kokke/tiny-AES128-C
GITHUB02: Huertas, Dani, AES algorithm implementation in C, 2016, https://github.com/dhuertas/AES
CONTE01: Conte, Brad, Implementation of AES in C, 2006, http://bradconte.com/aes_c