Testing - Rypto
April 5, 2017

Table of Contents

Preface.....2
Testing arrangements.....2
Test cases.....2
    AES_makeword.....2
    AES_RotWord, AES_SubWord.....2
    AES_KeyExpansion.....2
    AES_AddRoundKey, AES_SubBytes, AES_ShiftRows, AES_MixColumns.....2
    AES_Inv*.....2
    AES_encrypt, AES_decrypt.....2
How to repeat tests.....3
    Unit tests.....3
Test results.....3

Preface

"Tietorakenteet ja algoritmit" – excercise.

Rypto is a software, which can encrypt and decrypt.

Testing arrangements

Unit testing is done with CUnit framework and gradle.

High-level tests are implemented with a shell script.

Test cases

AES_makeword

The makeword test case is: 0x01, 0x02, 0x03, 0x04 → 0x01020304.

AES_RotWord, AES_SubWord

The RotWord and SubWord test cases were extracted from the standard, pp. 27, first line of the table.

AES_KeyExpansion

The three Key Schedule test cases were obtained from Sam Trenholme's web site http://www.samiam.org/key-schedule.html .

Selected test cases were the following:

- A key with all bits zero.
- A key with all bits one.
- A key with all bytes different.

AES_AddRoundKey, AES_SubBytes, AES_ShiftRows, AES_MixColumns

Test cases were taken from the standard, pp. 33, first possible cases.

AES_Inv*

Test cases were generated from standard version test cases by inverting input and output.

AES_encrypt, AES_decrypt

Test cases were taken from the standard, pp. 35-

How to repeat tests

Unit tests

The static libcunit.a must be linked to directory libs/ in the project root for tests to run.

Say

```
cradle build
```

from the command line.

Test results

Unit tests – passed on both development machine (Mac OS X) and melkki (Ubuntu Linux).