

# **Testing - Rypto**

## **April 7, 2017**

# Table of Contents

Preface.....	2
Testing arrangements.....	2
Test cases.....	2
Unit tests.....	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
Integration tests.....	3
Performance tests.....	3
How to repeat tests.....	3
Unit tests.....	3
Integration tests.....	3
Performance tests.....	3
Test results.....	3
Unit tests.....	3
Integration tests.....	3
Performance tests.....	3

## Preface

"Tietorakenteet ja algoritmit" – exercise.

Rypto is a software, which can encrypt and decrypt.

## Testing arrangements

Unit testing is done with CUnit framework and gradle.

Higher-level tests are implemented as a shell script.

## Test cases

### Unit tests

#### 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-

## **Integration tests**

## **Performance tests**

## **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.

## **Integration tests**

## **Performance tests**

## **Test results**

## **Unit tests**

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

## **Integration tests**

## **Performance tests**