Implementation of Naive DES, BitSlice DES, and AES on a PC

Rone Kwei Lim

Outline

- Implementation Description
- Testing and Validation
- Performance Tests
- References

Implementation Description

- Follows the description given in the class lecture notes
- Does the computation in a block on a bit level
- The permutation are performed bit by bit
- The S-boxes are implemented with table lookups
- Encrypt one block at a time (64 bits)

- Follows the description given in class lecture notes and the paper by Eli Biham
- Encryption is performed on 32 blocks in parallel
- Bit *i* from each block is placed in the *ith* word,
- Permutation is performed by exchanging words
- Substitution is not done with table lookups because each word contains bits from different blocks
- Substitution is done with Boolean functions

Implementation Description

Bitslice DES

- A paper by Matthew Kwan gives the S-boxes in terms of Boolean functions(XOR,OR,AND,NOT)
- Each S-box takes an average of 56 Boolean operations to compute
- Since encryption is performed on multiple blocks in parallel, CBC mode is not possible on a sequence of blocks
- But CBC mode can be done on multiple sequences of blocks (encrypt first block from multiple sequences together, then XOR and encrypt second block from each sequence)

- Follows the description given in class lecture notes
- Uses T-boxes, which combines SubByte step with the MixColumn step
- The T-boxes are generated before encryption in $GF(2^8)$
- The implementation uses AES-256 (256 bit key length, 128 bit block length)
- The key expansion takes the 256 bit key and generates 15 128-bit round keys
- This implementation is oriented toward performance, but it happens to be relatively resistant to side-channel timing attacks since the implementation uses table lookups, XORs, and does not have conditional statements that introduce timing differences

Testing and Validation

- After implementation, some of the test values on the NIST website for DES and AES were used to check for correctness of the implementation
- The test values consist of known plaintexts, keys, and ciphertexts
- About ~15 test values were used to check for correctness
- For AES-256, the round key expansion was also checked for 2 test keys

Performance Tests

- Testing Environment:
  - Language: C/C++ with Visual C++ compiler
  - CPU: Intel Core 2
  - Operating System: Windows 2000
- The encryptions are done with no feedback

- Key expansion time for 1,200,000 repetitions: 1.688s
- Encryption time for 3,000,000 repetitions: 5.172s
- Encryption speed: 37.1 Mbps
- The encryption is done after the round key expansion

- Conversion to bitslice time for 320,000 repetitions: 1.657s
- Conversion from bitslice time for 320,000 repetitions: 1.156s
- Key expansion time for 1,600,000 repetitions: 1.687s
- Encryption time (no conversion) for 350,000 repetitions: 2.844s
- Encryption time (with conversion at start and end) for 350,000 repetitions: 5.938s
- Encryption speed (no conversion): 252 Mbps
- Encryption speed (with conversion): 120.7 Mbps

- Conversion is done on 32 blocks
- Each encryption is done on 32 blocks in parallel
- Doing two conversions for each encryption reduces the speed by about 50%

- Round key expansion time for 4,000,000 repetitions: 1.297s
- Encryption time for 16,000,000 repetitions: 5.218s
- Encryption speed: 392.5 Mbps

Summary

- The AES implementation is about 55% faster than the bitslice DES implementation and about 950% faster than the naive DES implementation during encryption
- The DES implementation uses 56-bit key, but the AES implementation uses 256-bit key
- AES is faster and more secure than DES on PC

References I

C. Paar and C. K. Koc.
Cryptographic Engineering.
MEAD Course, 2008.

E. Biham.
A Fast New DES Implementation in Software.
Proceedings of the 4th International Workshop on Fast Software Encryption, 1997.

M. Kwan.
Reducing the Gate Count of Bitslice DES.
Cryptology ePrint Archive, 2000.

J. Daemen and V. Rijmen.
"AES Proposal: Rijndael"
First AES Conference, 1998.