Aufgabenblatt 6
IT-Security
Angewandte Informatik
WS 2018/19

Freiwillige
Zusatzaufgabe

Lernziele – 5 bzw. 7 Punkte

- AES-128-Verfahren
- Realisieren anhand eines Standard-Textes

Beim AES-Verfahren wird nicht mehr mit Langzahlarithmetik, sondern eher mit Bitoperationen gearbeitet.

Aufgaben

Sie sollen das AES-128-Verfahren realisieren.

1. Laden Sie sich die offizielle Definition vom Web herunter und machen Sie einen Entwurf. Entwerfen Sie für jeden Schritt eine eigene Routine, die Sie auch getrennt vom Rest testen können.
2. Die Signaturen sind hier einfach:
   - func Block encrypt(Block plain, Key k) - Ein 128 bit langer Block plain wird mit der 128 bit lange Key k verschlüsselt und als Wert geliefert.
   - func Block decrypt(Block cipher, Key k) - Ein 128 bit langer Block cipher wird mit der 128 bit lange Key k entschlüsselt und als Wert geliefert.
3. Die Realisierung sollte in jedem Fall schrittweise erfolgen; jeder Abschnitt entspricht einer eigenen Routine:
   - keyExpansion(...) und Invert_keyExpansion(...)
   - AddRoundKey(...)
   - SubBytes(...) und Invert_SubBytes(...)
   - ShiftRows(...) und Invert_ShiftRows(...)
   - MixColumns(...) und Invert_MixColumns(...)
4. Jede dieser Routinen wird isoliert getestet, z.B. mit Unittests.
5. Dann erfolgen die Tests mit den Testvektoren aus der Literatur.
6. Der endgültige Test erfolgt mit folgendem Verfahren: ein Block mit Zu-

fallswerten wird 16x verschlüsselt und dann 16x entschlüsselt. Der am Ende vorhandene Block muss dem Original entsprechen.

7. In den Links gibt es zahlreiche Verweise auf Testvektoren. Die von NIST sind in jedem Falle zu benutzen.
8. Wer SubBytes() mit Tabellen realisiert, erhält 5 Punkte, wer noch die Routinen zum Generieren der Tabellen schreibt, weitere 2 Punkte.

Bitte beachten Sie folgendes Prinzip: **Es kommt auf Korrektheit und nicht auf Performanz an**.

Links

- https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf
- https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Standards-and-Guidelines/documents/examples/AES_Core128.pdf
- https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Standards-and-Guidelines/documents/examples/AES_Core256.pdf
- https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Standards-and-Guidelines/documents/aes-development/rijndael-unix-refc.tar
- https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Standards-and-Guidelines/documents/aes-development/rijndael-vals.zip
- http://www.ccp4.ac.uk/dist/checkout/openssl-1.0.1j/test/evptests.txt
- https://www.3amsystems.com/Crypto-Toolbox/Test_vectors
- https://dxr.mozilla.org/mozilla-beta/source/security/nss/cmd/bltest/tests/aes_cts/aes-cts-type-1-vectors.txt
- https://www.ietf.org/rfc/rfc3962.txt
- https://www.hanewin.net/encrypt/aes/aes-test.htm

Abnahme

Zur Abnahme des gehören folgende Dateien:

- Source-Code und
- Projektdateien, z.B. netbeans-project oder make-Dateien etc.

Die Vorführung besteht in folgenden Ablauf: (1) Source-Code-Begutachtung, (2) Übersetzung und (3) Testlauf

Da es viele Beispielrealisierungen im Internet dazu gibt, ist die Verführung groß, diese leicht überarbeitet abzugeben. Machen Sie das nicht, Sie lernen

nichts.

Auch diese Aufgabe kann per Email abgegeben werden, dann mit allen Sourcen, Übersetzungsdateien, z.B. make bzw. eclipse/netbeans-Projekte, einschließlich der Tests. Alles muss ohne weiteres auf dem Rechner des Dozenten laufen können (Java, C++, Linux, CentOS 6.10). Programmieren Sie also portabel. Im Falle von Python, Ruby oder Lua bitte noch die notwendige Laufzeitumgebung mitliefern.