Aufgabenblatt 5
IT-Security
Angewandte Informatik
WS 2018/19

Lernziele – 7 Punkte

- SHA-256-Verfahren
- Realisieren anhand eines Standard-Textes

Bei den Hashverfahren wird nicht mehr mit Langzahlarithmetik, sondern eher mit Bitoperationen gearbeitet.

Aufgaben

Sie sollen das einfachste Hashverfahren aus der Gruppe der SHA2-Routinen 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. Jetzt zu den Signaturen. Da der Hashwert schrittweise durch Bearbeitung von Blöcken generiert wird, ist ein Interface erforderlich, der stückweise sich durch die Daten arbeitet:
   - func SHA256 hashInit() - Ein SHA256-Objekt mit den nötigen Tabellen und Puffern wird erzeugt
   - proc hashUpdate(SHA256 context, Block b) – es wird ein vollständiger Block b von Nutzdaten mit der Größe 512 bit (64 byte) dem Kontext hinzugefügt; der Bitstrom wird also um 512 bit verlängert.
   - func hash256 hashFinal(SHA256 context, Block b) – der Bitstrom wird mit dem letzten Block, dessen Länge zwischen 0 und 512 bit liegen kann, beendet und der Hashwert geliefert. Allokatierte Datenstrukturen werden freigegeben.
3. SHA256 ist eine interne Datenstruktur mit Tabellen und Puffern. Hash256 ist ein Array, dessen Bits aneinander gereiht den 256 bit breiten Hashwert ergeben, z.B. 16 32-bit-unsigned int.
4. Beachten Sie die Regeln des Paddings; hier müssen eventuell zwei Blöcke behandelt werden.

5. In den Links gibt es zahlreiche Verweise auf Testvektoren. Die von NIST sind in jedem Falle zu benutzen.

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

Links

- https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
- https://tools.ietf.org/pdf/rfc6234.pdf
- https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Standards-and-Guidelines/documents/examples/SHA256.pdf
- https://en.wikipedia.org/wiki/SHA-2
- https://de.wikipedia.org/wiki/Secure_Hash_Algorithm

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.