Hash Function Specification

Abstract

This document contains intellectual property and is intended for participents of the course T-110.5211 Cryptosystems **only**. Distribution is **forbidden**.

1 Specification

Implement a hash function following these steps:

1. Initialize a 128-byte state.
2. Pad the message.
3. The block size is one byte. For each block, XOR the block into state word x_{00000}, then perform 8 rounds of state transformation.
4. Perform finalization on the state.
5. Take the first 64 bytes as the hash.

Initialization. The state consists of 32 4-byte words (little-endian, indexed in binary)

$$x_{00000}, x_{00001}, x_{00010}, x_{00011}, x_{00100}, \ldots, x_{11111}.$$

Set the first three words $x_{00000}, x_{00001}, x_{00010}$ to 64, 1, and 8 respectively; all the rest are 0. Perform 80 rounds of state transformation.

Padding. Append 128 as a byte to the message.

Finalization. XOR 1 as a byte into state word x_{11111}. Perform 80 rounds of state transformation.

Rounds. A state transformation round consists of:

1. Add x_{0jklm} into x_{1jklm} modulo 2^{32} for each (j, k, l, m).
2. Rotate x_{0jklm} up 7 bits for each (j, k, l, m).
3. Swap x_{00klm} and x_{01klm} for each (k, l, m).
4. XOR x_{1jklm} into x_{0jklm} for each (j, k, l, m).
5. Swap x_{1jk0m} and x_{1jk1m} for each (j, k, m).
6. Add x_{0jklm} into x_{1jklm} modulo 2^{32} for each (j, k, l, m).
7. Rotate x_{0jklm} up 11 bits for each (j, k, l, m).
8. Swap x_{0j0lm} and x_{0j1lm} for each (j, l, m).
9. XOR x_{1jklm} into x_{0jklm} for each (j, k, l, m).
10. Swap x_{1jkl0} and x_{1jkl1} for each (j, k, l).

Sanity check: Per round, that's 32 XORs and 32 additions modulo 2^{32}.