A Brief Overview of the Skein Hash Function Family

Peter Boström

Royal Institute of Technology (RIOT)

February 22, 2012

Skein – Overview

Skein is built from these three new components:

- Threefish
- Unique Block Iteration (UBI)
- Optional Argument System

Modular Design

“Dividing up our design makes Skein easier to understand, analyze, and prove properties about.”

Threefish

Tweakable block cipher, defined with a 256-, 512- and 1024-bit block size.

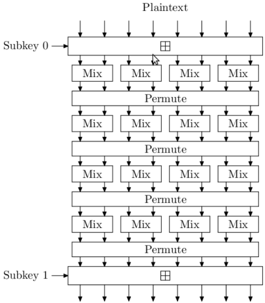
- Design Philosophy
- Substitution-Permutation Network
- MIX function
- Tweakable Block Cipher

Threefish – Design Philosophies

Skein and Threefish was designed with a few principles in mind:

- Simplicity
- Security per clock cycle
- Maximum diffusion
- Flexibility

Threefish – Substitution-Permutation Network

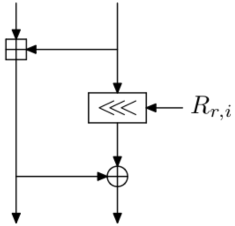


A subkey is inject every 4 rounds, instead of each round. 72/80 rounds in total.

Threefish was designed to maximize diffusion, each input bit affects every output bit after 10 rounds.

But unlike a normal SP network Threefish doesn't use S-boxes..

Threefish – MIX function



Instead of S-boxes, Threefish has a really simple MIX function between two words consisting of a single ADD, rotation and XOR.

Threefish's non-linearity comes from mixing addition modulo 2^{64} and XOR.

Threefish – Tweakable Block Cipher

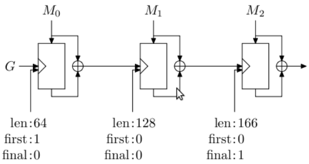
As a tweakable block cipher, Threefish takes a third input, a *tweak* value, aside from plaintext/ciphertext and key.

Threefish's key schedule is generated from both the *key* and *tweak*, which together determine the permutation computed by the cipher.

Tweak

"The purpose of the tweak is to make each block operation in Skein unique."

Unique Block Iteration



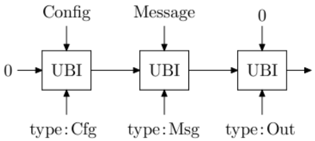
Unique Block Iteration (UBI) is a chaining mode which combines an input value (G) with an arbitrary-length input string to produce a fixed-size output.

The tweak value, encoding message type, length and first/final among others assures that each block is processed with a unique variant of the underlying cipher.

Tweak

"[A] message piece that produces one result in one location will produce a different result in a different location."

Skein Hashing

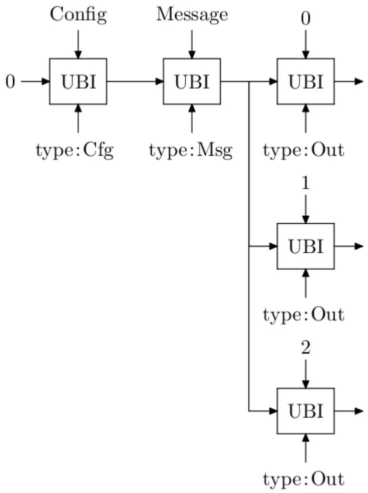


Skein is built on multiple invocations of UBI with different tweak values.

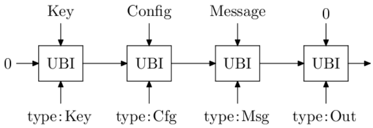
Regular Skein hashing is built from three UBI invocations:

- Configuration [output length, tree-hashing parameters] (IV)
- Message
- Output transformation (Finalization)

Skein Hashing with Larger Output Size



Skein's Flexibility



Skein's flexibility comes from its Optional-Argument System which defines support for using Skein as a/for:

- Tree hashing
- Message Authentication Codes (MAC)
- Digital signatures
- Key-Derivation Function
- Pseudo-random number generator (PRNG)
- Stream cipher

The End

Thanks for listening!