

The Rainstorm Hash Function: Formal Specification (Draft)

Cris (DOSAYGO Corp.), 2023

December 15, 2024

Overview

Rainstorm is a family of keyed hash functions designed by Cris (DOSAYGO Corp.) in 2023. It produces fixed-size outputs of 64, 128, 256, or 512 bits from arbitrary-length byte inputs. Rainstorm uses a 1024-bit internal state and processes input data in 512-bit (64-byte) blocks. Its constants and rotation values were chosen based on extensive testing to ensure desirable mixing properties.

While Rainstorm was created with cryptographic aims, it is currently undergoing evaluation, and no formal proofs of security exist yet. The community is encouraged to analyze and test Rainstorm, applying techniques ranging from heuristic analysis to rigorous cryptography methods.

Domain and Range

Input: A message M is a sequence of bytes ($M[0], \dots, M[L-1]$) with arbitrary length $L \geq 0$.

Seed: A 64-bit unsigned integer s (default $s = 0$). Different seeds produce distinct hash variants.

Output: A digest of length $N \in \{64, 128, 256, 512\}$ bits.

Parameters and Constants

Rainstorm uses fixed 64-bit constants and rotation amounts that were selected through computational searches and empirical testing to encourage strong avalanche properties.

Constants:

$$P = 2^{64} - 1 - 58, \quad Q = 13166748625691186689, \quad R = 1573836600196043749$$

$$S = 1478582680485693857, \quad T = 1584163446043636637, \quad U = 1358537349836140151$$

$$V = 2849285319520710901, \quad W = 2366157163652459183$$

$$K = [P, Q, R, S, T, U, V, W]$$

Rotation constants:

$$Z = [17, 19, 23, 29, 31, 37, 41, 53]$$

Counter values:

$$\text{CTR}_{\text{LEFT}} = 0x\text{fcdab8967452301}_{16}, \quad \text{CTR}_{\text{RIGHT}} = 0x\text{1032547698badcfe}_{16}$$

State Initialization

For a given message length L , seed s , and output length N , the internal state h is a vector of 16 64-bit words.

Define:

$$\begin{aligned} c_0 = 1, \quad c_1 = 2, \quad c_2 = 2, \quad c_3 = 3, \quad c_4 = 5, \quad c_5 = 7, \quad c_6 = 11, \quad c_7 = 13, \\ c_8 = 17, \quad c_9 = 19, \quad c_{10} = 23, \quad c_{11} = 29, \quad c_{12} = 31, \quad c_{13} = 37, \quad c_{14} = 41, \quad c_{15} = 43. \end{aligned}$$

Then:

$$h[i] = s + L + c_i \quad \text{for } i = 0, \dots, 15.$$

Block Processing and Endianness

Rainstorm processes the input in 512-bit (64-byte) chunks. Each chunk is interpreted as eight 64-bit words in little-endian format:

$$data[i] = \text{LE_decode_64}(B[8i : 8i + 7]), \quad i = 0, \dots, 7.$$

The output is produced in little-endian format for each 64-bit word of h .

Rounds and Weak Function

For each 64-byte chunk, Rainstorm performs 4 rounds of an internal function $\text{weakfunc}(h, data, left)$. The boolean $left$ is toggled each round.

Define $\text{ROTR}(x, n)$ as the 64-bit rotate-right operation on x by n bits.

The Weak Function

$$\text{weakfunc}(h, data, left)$$

If $left = true$:

$$ctr = \text{CTR}_{\text{LEFT}}$$

For $i = 0, \dots, 7$:

$$h[i] \leftarrow h[i] \oplus data[i], \quad h[i] \leftarrow h[i] - K[i], \quad h[i] \leftarrow \text{ROTR}(h[i], Z[i])$$

$$h[i+8] \leftarrow h[i+8] \oplus h[i], \quad ctr \leftarrow ctr + h[i], \quad h[(i+1) \bmod 8] \leftarrow h[(i+1) \bmod 8] - ctr$$

If $left = false$:

$$ctr = \text{CTR}_{\text{RIGHT}}$$

For $i = 8, \dots, 15$:

$$h[i] \leftarrow h[i] \oplus data[i-8], \quad h[i] \leftarrow h[i] - K[i-8], \quad h[i] \leftarrow \text{ROTR}(h[i], Z[i-8])$$

$$h[i-8] \leftarrow h[i-8] \oplus h[i], \quad ctr \leftarrow ctr + h[i], \quad h[((i-7) \bmod 8) + 8] \leftarrow h[((i-7) \bmod 8) + 8] - ctr$$

Padding

After processing all full 64-byte chunks, if any bytes remain, a final padded block is formed.

Let r be the remaining length (0 to 63). Initialize a 64-byte array *temp* with $(0x80 + r) \bmod 256$ in every byte, then overwrite the first r bytes of *temp* with the remaining data.

No length encoding beyond this padding is performed.

Process this final block with 4 rounds of weakfunc, then:

$$\text{for } i = 0 \text{ to } 7 : \quad h[i] \leftarrow h[i] - h[i + 8].$$

If $N > 64$ bits, perform $\max(N/64, 2)$ additional calls of $\text{weakfunc}(h, \text{temp}, \text{true})$ to further mix the state.

Output

The final hash is derived from $h[0 \dots 7]$ depending on the desired output length N :

- $N = 64$ bits: output $h[0]$
- $N = 128$ bits: output $(h[0], h[1])$
- $N = 256$ bits: output $(h[0], h[1], h[2], h[3])$
- $N = 512$ bits: output $(h[0], h[1], h[2], h[3], h[4], h[5], h[6], h[7])$

Empirical Results and Analysis Invitation

Rainstorm's design choices were guided by empirical testing, including runs through [SMHasher3](#) (by Frank J. T. Wojcik). Tests indicate strong avalanche characteristics and minimal bias, suggesting that small changes in input bits produce large, unpredictable changes in output bits. Preliminary performance benchmarks show around 247 cycles per hash for small keys and bulk speeds of about 1.92 GiB/s at 3.5 GHz.

These observations are encouraging but are not conclusive cryptographic proofs. Rainstorm has not been formally analyzed by the broader cryptographic community. Interested researchers and practitioners, whether new to cryptanalysis or established in the field, are invited to scrutinize Rainstorm's construction, test it against known cryptographic attacks, and share their findings.

Test Vectors

Below are sample test vectors (with $s = 0$ and $N = 256$):

Message: ""

Digest: e3ea5f8885f7bb16468d08c578f0e7cc15febd31c27e323a79ef87c35756ce1e

Message: "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"

Digest: 9e07ce365903116b62ac3ac0a033167853853074313f443d5b372f0225eede50

Message: "The quick brown fox jumps over the lazy dog"

Digest: f88600f4b65211a95c6817d0840e0fc2d422883ddf310f29fa8d4cbfda962626

Message: "The quick brown fox jumps over the lazy cog"

