Hamsi Hamsi is moderately parallelizable, in that they argue that the S-boxes can be evaluated in parallel. They also implement diffusion in their compression function. Regarding security, they make the standard claims for complexity of breaking a good hash function, but they provide no evidence to back up any security claims. A comment on the SHA-3 wiki claims that Hamsi's compression function does not exhibit pseudorandomness with up to 5 rounds.

SWIFFTX SWIFFTX is based on a similar function SWIFFT, but wraps that in a non-linear transformation. The paper for SWIFFT claims provable one-wayness, as well as provable collision resistance and second preimage resistance [Citation 4, sec. 5]. The SWIFFTX paper claims that the lack of pseudorandomness in SWIFFT is addressed by their additional transformation. For performance, SWIFFTX is designed to be highly parallelizable through the use of an FFT.

Sgàil The paper for Sgàil claims that it should provide good performance through parallelization. They also use diffusion, and claim that their algorithm exhibits pseudorandomness. They give an upper bound on susceptibility to linear and differential cryptanalysis which makes such attacks worse than brute force [36]. They list first preimage resistance, second preimage resistance, and collision resistance in their design objectives, but do not give any proofs of these specific criteria. The author discovered a flaw in his own submission, which completely broke its collision resistance and second preimage resistance. He released a corrected version of the standard, but is unsure where it stands with regard to the competition.

Grøstl This algorithm shows evidence for one-wayness, second preimage resistance, and collision resistance. They offer a proof of resistance to differential cryptanalysis, showing that this is far more complex that finding a collision through a standard birthday attack [15]. They also claim a resistance to linear cryptanalysis [16]. Since Grøstl is similar to AES, they also make claims that it is resistant to an attack based on "Integrals" [16]. They note that with regard to algebraic cryptanalysis, solving the system of equations required to break the hash has unknown complexity, but they believe it is worse than brute force, and would likely also break AES [17]. They note that their algorithm is highly parallelizable, to the point of implementing it to take one clock cycle per compression function round.

We chose Grøstl as the best of the four because it seemed to provide the most rigorous security analysis, and because it seemed to lend itself to a very efficient hardware implementation. They seemed to have a reasonable argument for common vulnerabilities, and the fact that some unknown factors were tied to the viability of a widely used and long-standing encryption algorithm should alleviate concerns about those factors to some extent.