CubeHash features (2.B.6)

Daniel J. Bernstein *

Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607–7045
cubehash@box.cr.yp.to

This statement lists and describes the advantages and limitations of the CubeHash family of hash functions. This statement is not meant to exclude the possibility of further advantages and limitations being discovered during the SHA–3 candidate evaluation process.

Low area requirements. CubeHashr/b xors b bytes of input into the first b bytes of a 128-byte state. It then modifies the state in place and moves on to the next b bytes of input. This 128-byte state is small enough to fit into software environments having very little memory. The modifications are simple and regular, so CubeHash can also fit into small area on an FPGA or ASIC.

For comparison, although SHA–256 can store its state *between* blocks in just 32 bytes (plus 8 bytes for a message-length counter), SHA–256 needs at least 128 bytes to process a block: 32 bytes for the current state, 32 bytes for the beginning-of-block state, and 64 bytes for the current segment of the message schedule. SHA–512 needs at least 256 bytes.

Parallelizability. Each step of CubeHash consists of 16 independent operations on 32-bit words. The operations are parallelizable and vectorizable, providing tremendous flexibility for the implementor and allowing CubeHash to run at high speeds on a wide variety of computer architectures.

CubeHash does *not* have the sort of global-scale parallelism that would be provided by a message-length tree of block hashes. The advantage of a tree is that it allows very long messages to be split across several processor cores—but the applications that care can achieve the same benefit by building a tree on top of CubeHash, the same way that they have traditionally built a tree on top of SHA–256. (Similar comments apply to incremental hashing etc.) The disadvantage of a tree is that it cannot be implemented in low area. There are "group hashes" that achieve global-scale parallelizability with smaller costs in area, but my impression is that all of those hashes will be broken by quantum computers.

Other message-digest sizes. CubeHash supports h-bit output lengths for every $h \in \{8, 16, 24, \ldots, 512\}$. In particular, CubeHash supports the required output lengths of 224 bits (28 bytes), 256 bits (32 bytes), 384 bits (48 bytes), and 512 bits (64 bytes). Note that some applications do not need collision resistance and are satisfied with output lengths significantly below 256 bits.

* The author was supported by the National Science Foundation under grant ITR–0716498. Date of this document: 2008.10.28.