CubeHash expected strength (2.B.4)

Daniel J. Bernstein *

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

This is a statement of the expected strength (i.e., cryptanalytic work factor) of CubeHash. See the CubeHash specification for recommended parameters r, b.

224-bit collisions. CubeHash–224 is expected to provide collision resistance of approximately 112 bits.

256-bit collisions. CubeHash–256 is expected to provide collision resistance of approximately 128 bits.

384-bit collisions. CubeHash–384 is expected to provide collision resistance of approximately 192 bits.

512-bit collisions. CubeHash–512 is expected to provide collision resistance of approximately 256 bits.

224-bit preimage resistance. CubeHash–224 is expected to provide preimage resistance of approximately 224 bits, but quantum computers are expected to reduce preimage resistance to approximately 112 bits.

256-bit preimage resistance. CubeHash–256 is expected to provide preimage resistance of approximately 256 bits, but quantum computers are expected to reduce preimage resistance to approximately 128 bits.

384-bit preimage resistance. CubeHash–384 is expected to provide preimage resistance of approximately 384 bits, but quantum computers are expected to reduce preimage resistance to approximately 192 bits.

512-bit preimage resistance. CubeHash–512 is expected to provide preimage resistance of approximately 512 bits, but quantum computers are expected to reduce preimage resistance to approximately 256 bits.

224-bit second-preimage resistance. CubeHash–224 is expected to provide second-preimage resistance of at least $224 - k$ bits for messages shorter than 2^k bits, but quantum computers are expected to reduce preimage resistance to approximately 112 bits.

256-bit second-preimage resistance. CubeHash–256 is expected to provide second-preimage resistance of at least $256 - k$ bits for messages shorter than

* The author was supported by the National Science Foundation under grant ITR–0716498. Date of this document: 2009.09.14.

2^k bits, but quantum computers are expected to reduce preimage resistance to approximately 128 bits.

384-bit second-preimage resistance. CubeHash–384 is expected to provide second-preimage resistance of at least $384 - k$ bits for messages shorter than 2^k bits, but quantum computers are expected to reduce preimage resistance to approximately 192 bits.

512-bit second-preimage resistance. CubeHash–512 is expected to provide second-preimage resistance of at least $512 - k$ bits for messages shorter than 2^k bits, but quantum computers are expected to reduce preimage resistance to approximately 256 bits.

224-bit length-extension resistance. CubeHash–224 is expected to resist all feasible length-extension attacks.

256-bit length-extension resistance. CubeHash–256 is expected to resist all feasible length-extension attacks.

384-bit length-extension resistance. CubeHash–384 is expected to resist all feasible length-extension attacks.

512-bit length-extension resistance. CubeHash–512 is expected to resist all feasible length-extension attacks.

224-bit PRF. HMAC using CubeHash–224 is expected to resist all distinguishing attacks that require much fewer than 2^{112} queries and significantly less computation than a preimage attack. This submission does not include any ad-hoc PRF modes.

256-bit PRF. HMAC using CubeHash–256 is expected to resist all distinguishing attacks that require much fewer than 2^{128} queries and significantly less computation than a preimage attack. This submission does not include any ad-hoc PRF modes.

384-bit PRF. HMAC using CubeHash–384 is expected to resist all distinguishing attacks that require much fewer than 2^{192} queries and significantly less computation than a preimage attack. This submission does not include any ad-hoc PRF modes.

512-bit PRF. HMAC using CubeHash–512 is expected to resist all distinguishing attacks that require much fewer than 2^{256} queries and significantly less computation than a preimage attack. This submission does not include any ad-hoc PRF modes.

224-bit MAC. HMAC using CubeHash–224 is expected to resist all forgery attacks that require much fewer than 2^{112} queries and significantly less computation than a preimage attack. This submission does not include any ad-hoc MAC modes.

256-bit MAC. HMAC using CubeHash–256 is expected to resist all forgery attacks that require much fewer than 2^{128} queries and significantly less computation than a preimage attack. This submission does not include any ad-hoc MAC modes.

384-bit MAC. HMAC using CubeHash–384 is expected to resist all forgery attacks that require much fewer than 2^{192} queries and significantly less computation than a preimage attack. This submission does not include any ad-hoc MAC modes.

512-bit MAC. HMAC using CubeHash–512 is expected to resist all forgery attacks that require much fewer than 2^{256} queries and significantly less computation than a preimage attack. This submission does not include any ad-hoc MAC modes.

224-bit randomized hashing. CubeHash–224 is not expected to degrade the generic security of any of the NIST-specified randomized-hashing modes. This submission does not include any ad-hoc randomized hashing modes.

256-bit randomized hashing. CubeHash–256 is not expected to degrade the generic security of any of the NIST-specified randomized-hashing modes. This submission does not include any ad-hoc randomized hashing modes.

384-bit randomized hashing. CubeHash–384 is not expected to degrade the generic security of any of the NIST-specified randomized-hashing modes. This submission does not include any ad-hoc randomized hashing modes.

512-bit randomized hashing. CubeHash–512 is not expected to degrade the generic security of any of the NIST-specified randomized-hashing modes. This submission does not include any ad-hoc randomized hashing modes.

Output-bit selection. Selection of m output bits (e.g., truncation to the first m bits) is expected to have the usual effects on security.

Supporting rationale. See the accompanying security analysis.