Hashing methods categories

Slow	Fast	Message Authentication
Argon2 (i, d, id)	SHA-3	HMAC-SHA256
Bcrypt	BLAKE2	HMAC-SHA512
Scrypt	SHA-2	CMAC
PBKDF2	MD5	GMAC
N/A	SHA-1	HMAC-SHA1
N/A	CRC32	HMAC-MD5

Hashing methods use cases

Hashing Method	Use Case
Argon2 (i, d, id)	Password hashing
SHA-3	General-purpose hashing, modern alternative to SHA-2
BLAKE2	High-performance hashing, checksums, message digests
HMAC-SHA256	Message authentication, API signatures, JWT tokens
HMAC-SHA512	Message authentication, high-security applications
SHA-2	General-purpose hashing, digital signatures, certificates, blockchain
Bcrypt	Password hashing - legacy systems
Scrypt	Password hashing - memory-hard function
CMAC	Message authentication with block ciphers (AES-CMAC)
GMAC	Message authentication in GCM mode (AES-GCM)
PBKDF2	Password hashing - legacy compatibility, key derivation
HMAC-SHA1	Message authentication
HMAC-MD5	Message authentication
MD5	Checksums
SHA-1	Digital signatures
CRC32	Error detection

Security status classification

Security Status	Hashing Methods
Recommended	Argon2 (i, d, id), SHA-3, BLAKE2, HMAC-SHA256, HMAC-SHA512
Acceptable	SHA-2, Bcrypt, Scrypt, CMAC, GMAC
Legacy compliant	PBKDF2
Deprecated	HMAC-SHA1, HMAC-MD5
Obsolete	MD5, SHA-1, CRC32

Security status rationale

Recommended: Current best practices with rigorous peer review and strong protection against known attacks. Argon2 resists GPU/ASIC attacks. SHA-3 is the latest NIST standard. BLAKE2 offers high performance with strong security. HMAC-SHA256/512 provide robust message authentication.

Acceptable: Cryptographically sound algorithms widely deployed in production. SHA-2 is secure and ubiquitous in TLS/SSL and blockchain. Bcrypt and Scrypt are proven password hashing functions. CMAC and GMAC serve specific cryptographic purposes with AES.

Legacy Compatible: PBKDF2 for backward compatibility with older systems. Requires high iteration counts for security and lacks memory-hardness. Use only when legacy integration is required.

Deprecated: Being phased out due to weaknesses in underlying hash functions (MD5, SHA-1). HMAC provides limited protection, but base algorithm vulnerabilities make these unsuitable for new implementations. Many compliance frameworks forbid their use.

Obsolete: Cryptographically broken or fundamentally insecure. MD5 and SHA-1 have demonstrated collision vulnerabilities. CRC32 is a non-cryptographic checksum for error detection only. Violates compliance standards and must be replaced immediately.