SHA-3

From:

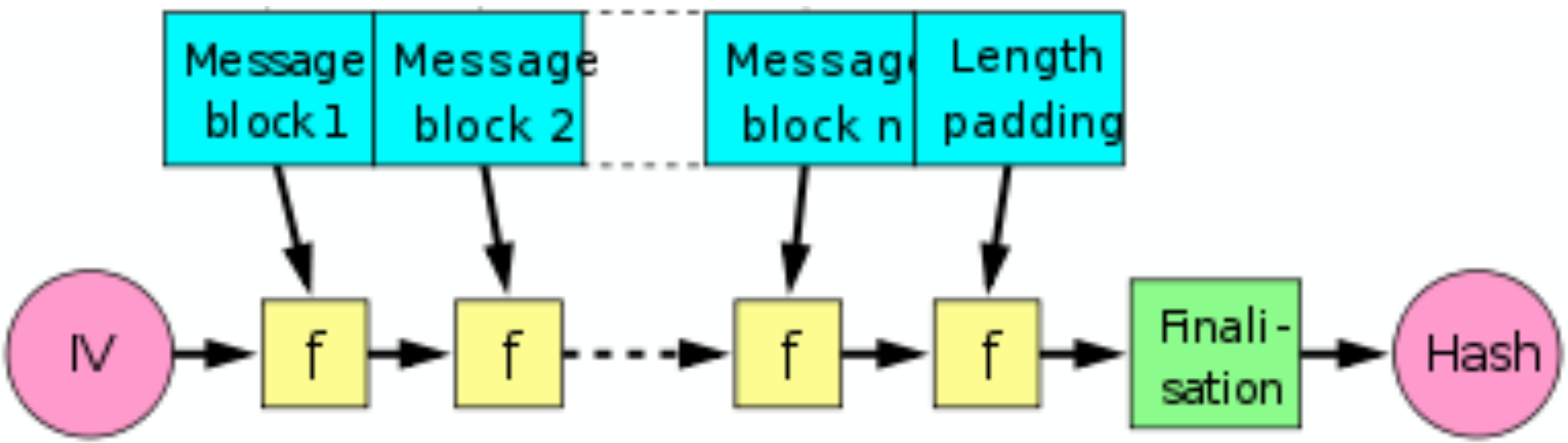
SHA3 where we've been, where we're going

written by John Kelsey (NIST) for the RSA Conference 2013

Origins

- Hash functions appeared as an important idea at the dawn of modern public crypto.
- Many ideas floating around to build hash functions from block ciphers (DES) or mathematical problems.
- Ways to build hash functions from compression functions
  - Merkle-Damgaard
- Ways to build compression functions from block ciphers
  - Davies-Meyer, MMO, etc.

Merkle-Damgaard



- Used in all widespread hash functions before 2004
  - MD4, MD5, RIPE-MD, RIPE-MD160, SHA0, SHA1, SHA2

Image from Wikipedia

The MD4 Family

- Rivest published MD4 in 1990
- 128-bit output
- Built on 32-bit word operations
- Add, Rotate, XOR, bitwise logical operations
- Fast
- First widely used dedicated hash function

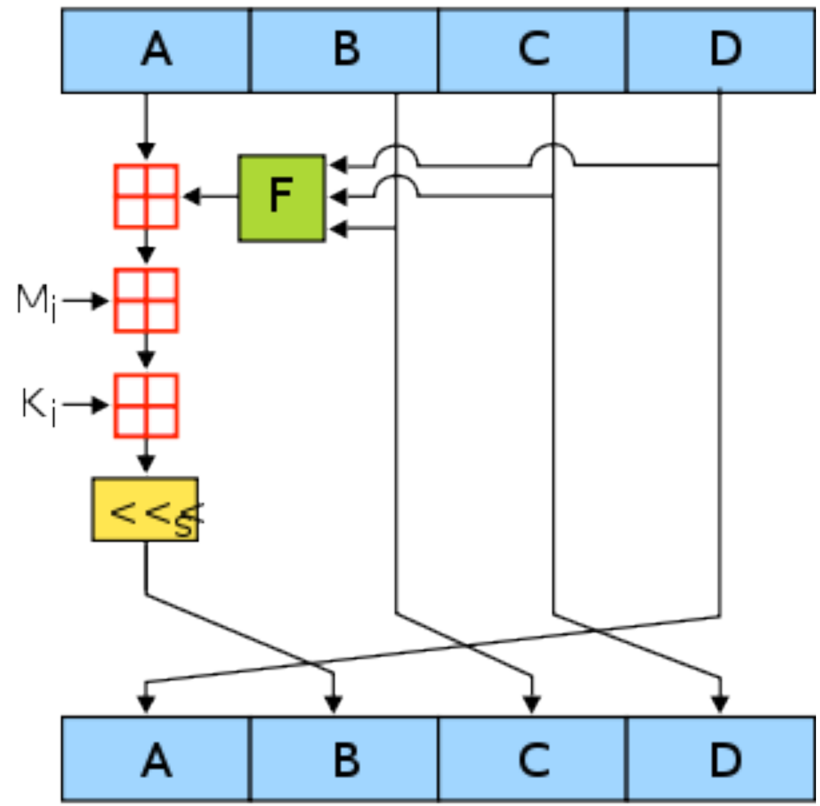


Image from Wikipedia MD4 Article

MD5

- Several researchers came up with attacks on weakened versions of MD4
- Rivest created stronger function in 1992
- Still very fast
- Same output size
- *Some attacks known*
  - *Den Boer/Bosselaers*
  - *Dobbertin*

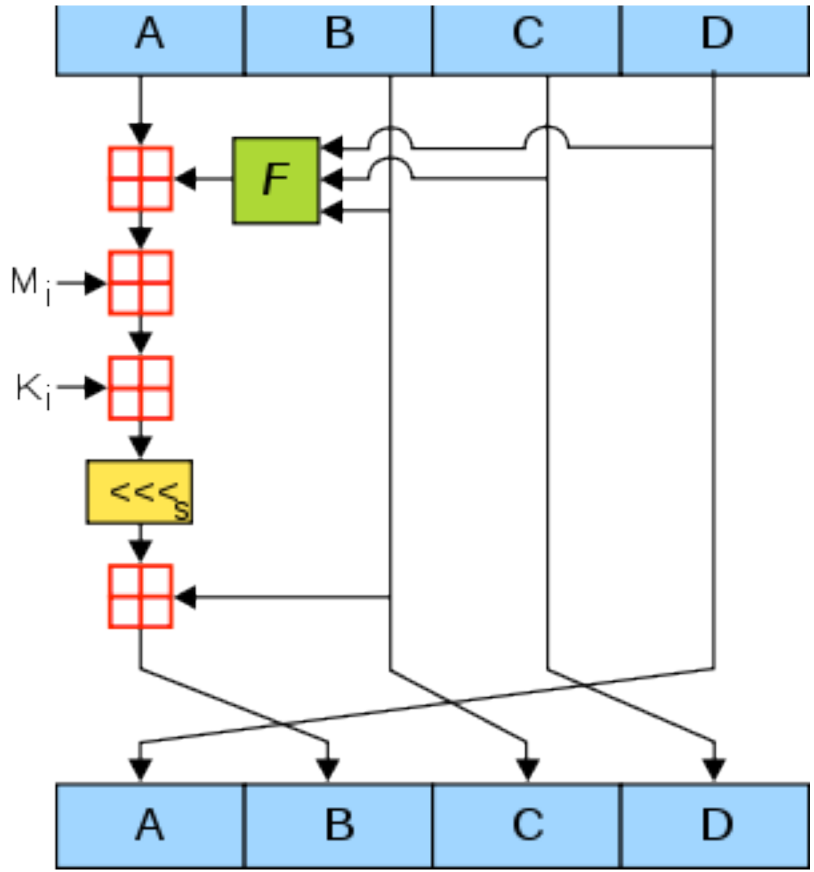


Image from Wikipedia MD5 Article

SHA0 and SHA1

- SHA0 published in 1993
- 160-bit output
  - (80 bit security)
- NSA design
- Revised in 1995 to SHA1
  - Round function (pictured) is same
  - Message schedule more complicated
- *Crypto '98 Chabaud/Joux attack on SHA0*

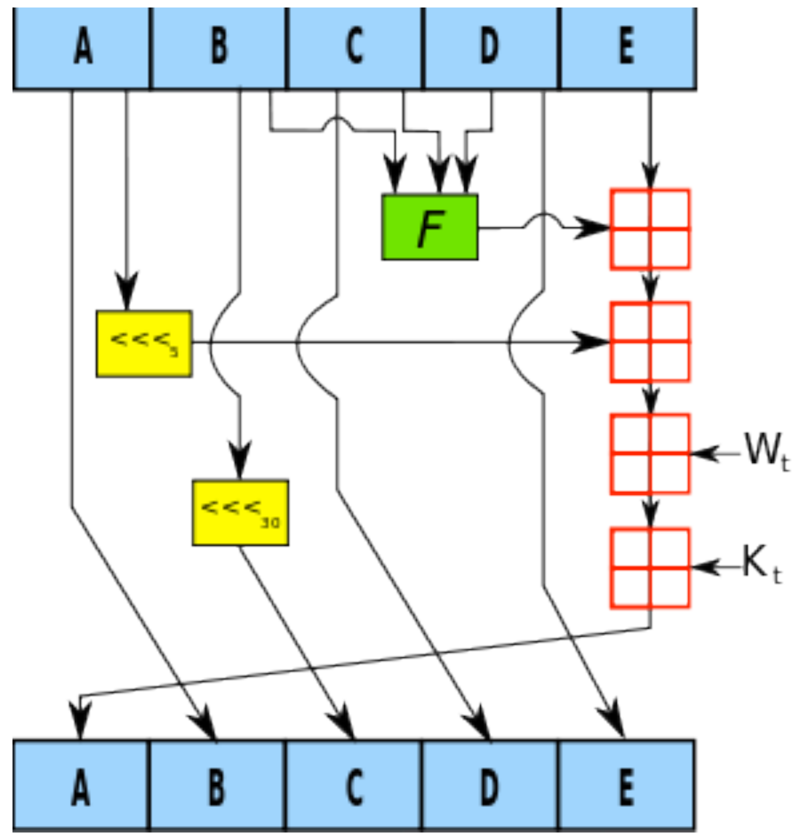
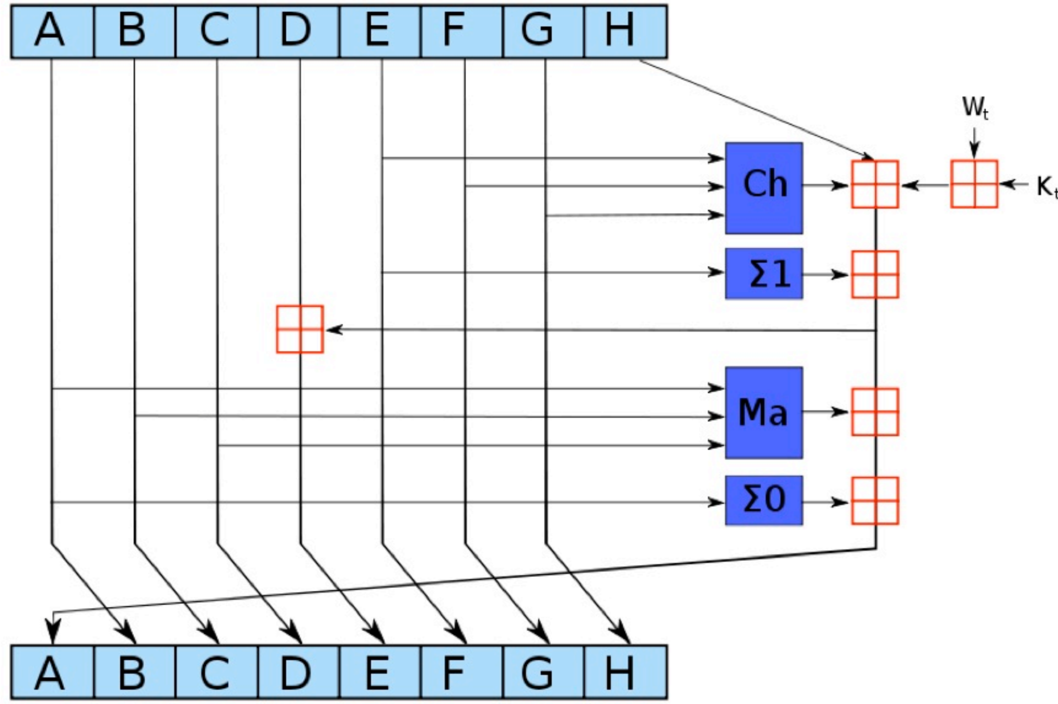


Image from Wikipedia SHA1 Article

SHA2

- Published 2001
- Three output sizes
  - 256, 384, 512
  - 224 added in 2004
- Very different design
- Complicated message schedule

- *Still looks strong*



$\mathrm{Ch}(E,F,G) = (E \wedge F) \oplus (\neg E \wedge G)$

$\mathrm{Ma}(A,B,C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$

$\Sigma_0(A) = (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22)$

$\Sigma_1(E) = (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)$

The bitwise rotation uses different constants for SHA-512. The given numbers are for SHA-256. The red $\boxplus$ is addition modulo 2^{32}.

Image from Wikipedia SHA2 Article

As of 2004, we thought we knew what we were doing.

- MD4 was known to be broken by Dobbertin, but still saw occasional use
- MD5 was known to have theoretical weaknesses from Den Boer/Bosselaers and Dobbertin, but still in wide use.
- SHA0 was known to have weaknesses and wasn't used.
- SHA1 was thought to be very strong.
- SHA2 looked like the future, with security up to 256 bits
- Merkle-Damgaard was normal way to build hashes

Crypto 2004: The Sky Falls

Conference:

- Joux shows a surprising property in Merkle-Damgaard hashes
  - Multicollisions
  - Cascaded hashes don’t help security much
- Biham/Chen attack SHA0 (neutral bits)

Rump Session:

- Joux shows attack on SHA0
- Wang shows attacks on MD4, MD5, RIPEMD, some Haval variants, and SHA0
  - Much better techniques used for these attacks

Aftermath: What We Learned

- *We found out we didn't understand hashes as well as we thought.*
- Wang's techniques quickly extended
  - Better attacks on MD5
  - Claimed attacks on SHA1 (2005)

- Joux's multicollisions extended and applied widely
  - Second preimages and herding
  - Multicollisions even for multiple passes of hash
  - Much more

What to do next?

- All widely used hash functions were called into question
  - MD5 and SHA1 were very widespread
  - SHA2 and RIPE-MD160, neither one attacked, were not widely used.
- At same time, NIST was pushing to move from 80- to 112-bit security level
  - Required switching from SHA1 to SHA2
- Questions about the existing crop of hash functions
  - SHA1 was attacked, why not SHA2?

Pressure for a Competition

- We started hearing from people who wanted a hash competition
- AES competition had happened a few years earlier, and had been a big success
- This would give us:
  - Lots of public research on hash functions
  - A new hash standard from the public crypto community
  - Everything done out in the open

2007: Call for proposals

- We spent a lot of time getting call for proposals nailed down:
  - Algorithm spec
  - Security arguments or proofs
  - Preliminary analysis
  - Tunable security parameter(s)

Security Requirements

- Drop-in replacement
  - Must provide 224, 256, 384, and 512 bit output sizes
  - Must play well with HMAC, KDFs, and other existing hash uses
- N bit output:
  - N/2 bit collision resistance
  - *N bit preimage resistance*
  - N-K bit second preimage resistance
    - K = lg(target message length)
- Eliminate length-extension property!
- Tunable parameter to trade off between security and performance.

Initial submissions

- We started with 64 submissions (10/08)
- 51 were complete and fit our guidelines
- We published those 51 on December 2008

- Huge diversity of designs
- 51 hash functions were too many to analyze well
- There was a *lot* of cryptanalysis early on, many hash functions were broken

Narrowing the field down to 14

BLAKE BMW Cubehash Echo Fugue ***Grostl*** Hamsi
JH ***Keccak*** Luffa SHABAL SHAVite SIMD ***Skein***

- Many of the first 51 submissions were broken or seriously dented in the first year of the competition.
- Others had unappealing performance properties or other problems.
- AES competition had 15 submissions; we took a year to get down to 14.
- Published our selections in July 2009

Choosing 5 finalists

BLAKE Grostl JH Keccak Skein

- Published selection in Dec 2010
- Much harder decisions
  - Cryptanalytic results were harder to interpret
  - Often distinguishers of no apparent relevance
- All five finalists made tweaks for third round
  - BLAKE and JH increased number of rounds
  - Grostl changed internals of Q permutation
  - Keccak changed padding rules
  - Skein changed key schedule constant

Choosing a Winner: Performance

- All five finalists have acceptable performance
- ARX designs (BLAKE and Skein) are excellent on high-end software implementations
- JH and Grostl fairly slow in software
  - Slower than SHA2
- Keccak is very hardware friendly
  - High throughput per area

Keccak performs well everywhere, and very well in hardware.

Complementing SHA2

- SHA3 will be deployed into a world full of SHA2 implementations
- SHA2 still looks strong
- We expect the standards to coexist.
- SHA3 should *complement* SHA2.
  - Good in different environments
  - Susceptible to different analytical insights

Keccak is fundamentally different from SHA2. Its performance properties and implementation tradeoffs have little in common with SHA2.

Wrapup on Selecting a Winner

- Keccak won because of:
  - High security margin
  - Fairly high quality, in-depth analysis
  - Elegant, clean design
  - Excellent hardware performance
  - Good overall performance
  - Flexability: rate is readily adjustable
  - Design diversity from SHA2

Hash Competition Timetable

Date	Event	Candidates Left
11/2/2007	Call for Proposals published, competition began	
10/31/2008	SHA3 submission deadline	***64***
12/10/2008	*First-round candidates announced*	*51*
2/25/2009	First SHA3 workshop in Leuven, Belgium	51
7/24/2009	*Second-round candidates announced*	*14*
8/23/2010	Second SHA3 workshop in Santa Barbara, CA	14
12/9/2010	*SHA3 finalists announced*	*5*
3/22/2012	Third SHA3 workshop in Washington, DC	5
10/2/2012	*Keccak announced as the SHA3 winner*	***1***

Security and Output Size

- Traditionally, hash functions' security level is linked to their output size
  - SHA256: 128 bit security against collisions, 256 against preimage
  - Best possible security for hash with 256-bit output.
- Keccak has variable output length, which breaks this link
  - Need a notion of security level separate from output size
- Keccak is a sponge
  - Security level is determined by *capacity*
  - Tunable parameter for performance/security tradeoff