Lifetimes of cryptographic hash functions

I've written some cautionary articles on using cryptographic hashes to create content-based addresses (compare-by-hash). This page brings together everything I've written and keeps an updated table of the status of popular cryptographic hash functions.

Quick summary of my recommendations on compare-by-hash: If you are using compare-by-hash to generate addresses for data that can be supplied by malicious users, you should have a plan to migrate to a new hash every few years. For example, BitTorrent falls into this category, but rsync doesn't. Keep in mind that new, more secure hashes are likely to have larger outputs (e.g., 256 bits for SHA-2 vs. 160 bits for SHA-1) and be more computationally expensive.

An Analysis of Compare-by-hash appeared in Hot Topics in Operating Systems 2003 The original paper casting doubt on compare-by-hash as the answer to all of life's problems.

The code monkey's guide to cryptographic hash functions appeared in LinuxWorld Practical advice for programmers, plus the chart of popular hash function lifetimes (reproduced below).

Lifetimes of popular cryptographic hashes (the rainbow chart)

Function	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Snefru																												
MD2 (128-bit)[1]																												
MD4																												
MD5															[2]													
RIPEMD															[2]													
HAVAL-128[1]															[2]													
SHA-0																												
SHA-1																												[3]
RIPEMD-160																												
SHA-2 family																		[4]										
SHA-3 (Keccak)																												

Key: Didn't exist/not public | Under peer review | Considered strong | Minor weakness | Weakened | Broken | Collision found

[1] Note that 128-bit hashes are at best 2^64 complexity to break; using a 128-bit hash is irresponsible based on sheer digest length.

[2] What happened in 2004? Xiaoyun Wang and Dengguo Feng and Xuejia Lai and Hongbo Yu happened.

[3] Google spent 6500 CPU years and 110 GPU years to convince everyone we need to stop using SHA-1 for security critical applications. Also because it was cool.

[4] In 2007, the NIST launched the SHA-3 competition because "Although there is no specific reason to believe that a practical attack on any of the SHA-2 family of hash functions is imminent, a successful collision attack on an algorithm in the SHA-2 family could have catastrophic effects for digital signatures." One year later the first strength reduction was published.

The Hash Function Lounge has an excellent list of references for most of the dates. Wikipedia now has references to the rest.

Reactions to stages in the life cycle of cryptographic hash functions

Stage	Expert reaction	Programmer reaction	Non-expert ("slashdotter") reaction
Initial proposal	Skepticism, don't recommend use in practice	Wait to hear from the experts before adding to your crypto library	SHA-what?
Peer review	Moderate effort to find holes and garner an easy publication	Used by a particularly adventurous developers for specific purposes	Name-drop the hash at cocktail parties to impress other geeks
General acceptance	Top-level researchers begin serious work on finding a weakness (and international fame)	Even Microsoft is using the hash function now	Flame anyone who suggests the function may be broken in our lifetime
Minor weakness discovered	Massive downloads of turgid pre-prints from arXiv, calls for new hash functions	Start reviewing other hash functions for replacement	Long semi-mathematical posts comparing the complexity of the attack to the number of protons in the universe
Serious weakness discovered	Tension-filled CRYPTO rump sessions! A full break is considered inevitable	Migrate to new hash functions immediately, where necessary	Point out that no actual collisions have been found
First collision found	Uncork the champagne! Interest in the details of the construction, but no surprise	Gather around a co-worker's computer, comparing the colliding inputs and running the hash function on them	Explain why a simple collision attack is still useless, it's really the second pre-image attack that counts
Meaningful collisions generated on home computer	How adorable! I'm busy trying to break this new hash function, though	Send each other colliding X.509 certificates as pranks	Claim that you always knew it would be broken
Collisions generated by hand	Memorize as fun party trick for next faculty mixer	Boggle	Try to remember how to do long division by hand
Assumed to be weak but no one bothers to break	No one is getting a publication out of breaking this	What's this crypto library function for?	Update Pokemon Wikipedia pages