SHA3 Competition Status Update

	Narrow-Pipe	**MD**	**Wide-Pipe**	**MD**	**Sponge**	**Sponge-Like**
Bitsliced	*Hamsi*		*JH*		*Keccak*	*Luffa*
AES	*Shavite3*		*Echo*	*Grostl*		*Fugue*
ARX	*Skein*	BLAKE	*BMW*		*Cube*	
Logical/ ARX			*SIMD*	*Shabal*		

John Kelsey, NIST

Overview

- Recent history and timetable
- SHA3 conference discussions
- Weighing the candidates

	Narrow-Pipe	**MD**	**Wide-Pipe**	**MD**	**Sponge**	**Sponge-Like**
Bitsliced	*Hamsi*		*JH*		*Keccak*	*Luffa*
AES	*Shavite3*		*Echo*	*Grostl*		*Fugue*
ARX	*Skein*	BLAKE	*BMW*		*Cube*	
Logical/ ARX			*SIMD*	*Shabal*		

History and Timeline

- SHA3 competition announced Nov 2007
- 63 submissions received Oct 2008
- 51 accepted for first round Dec 2008
- 1st SHA3 Conference Feb 2009
- 14 semifinalists announced July 2009
- 2nd SHA3 Conference Aug 2010
- 4-6 finalists announced by end of year 2010
- 3rd SHA3 Conference Spring 2012
- Winner announced sometime in 2012

SHA3 Conference 2010

- Two weeks ago we had SHA3 Conference in Santa Barbara
- Lots of interesting presentations/papers
- No earth-shaking results
- A lot of interesting discussions

Selecting Finalists

- This is what we've all been thinking about
- Weighing many criteria
  - Cryptanalysis
  - Design diversity
  - Performance
- Rest of this talk is about what we're thinking
- Looking for feedback on our ideas
  - Please tell me where I'm wrong!

Selection: What Do We Need?

	Narrow-Pipe	MD	Wide-Pipe	MD	Sponge	Sponge-Like
Bitsliced	*Hamsi*		*JH*		*Keccak*	*Luffa*
AES	*Shavite3*		*Echo*	*Grostl*		*Fugue*
ARX	*Skein*	BLAKE	*BMW*		*Cube*	
Logical/ ARX			*SIMD*	*Shabal*		

How Will SHA3 Be Used?

- SHA2 (-224, -256, -384, -512) is already being deployed
  - This is the only thing we've had to offer anyone since the SHA1 result was announced.
- SHA3 will deploy into a world where it competes with SHA2
  - If SHA3 is much slower/bigger/etc. than SHA2, will anyone ever use it?

SHA512/256

- We will soon have a standard way to use SHA512 and truncate to 256 bits
  - Much better performance on 64 bit machines.
  - Suggests that competition on 64 bit machines will be SHA512, for all security levels.
- By the time SHA3 sees widespread use, all desktop and laptop machines will probably be 64 bit.
  - Can we assume most machines will have AES instruction or vector instructions?

What Else Are We Doing with Hash Standards?

- We have standard for randomized hash.
- We will probably work out a standard for tree-hashing using any approved hash after the competition is over.
- We use hash functions in KDFs, PRFs, PRNGs, and many other places.
- Sponge hashes have an interesting effect here: Claim security in KDF/PRF/PRNG sorts of modes without novel constructions.

Dual Signatures

- Idea floated at SHA3 workshop in Santa Barbara this year: Future standards should require two hashes where possible
  - DSA / ECDSA: Two separate signatures
  - RSA: One signature with composite hash
- Justification: In many applications, this doesn't cost much. But it makes many attacks impossible or much harder.
  - Is there a < 2^{64} attack now on

    hash(X) = md4(X) || md5(X) ?

Extras

- Some SHA3 candidates offer extra functionality
- Keccak:
  - Built in PRF and PRNG
  - Duplex encryption mode
- HAIFA designs:
  - Built in salt for PRF or randomized hashing
- Skein:
  - Wide block cipher

Should any of this matter in SHA3 selection?

Selection: Design Diversity

	Narrow-Pipe MD		Wide-Pipe MD		Sponge	Sponge-Like
Bitsliced	*Hamsi*		*JH*		*Keccak*	*Luffa*
AES	*Shavite3*		*Echo*	*Grostl*		*Fugue*
ARX	*Skein*	BLAKE	*BMW*		*Cubehash*	
Logical/ARX			*SIMD*	*Shabal*		

We don't want all the finalists to look alike.

- More to the point: We don't want all the finalists to fall to the same attack.
- Question: Is there a strategy to choose finalists so that not too many are likely to fall to a single new attack or insight?
- Best way we know is to consider *design diversity* in choosing finalists.
- AKA avoiding a monoculture

What Makes a Monoculture?

	Narrow-Pipe	**MD**	**Wide-Pipe**	**MD**	**Sponge**	**Sponge-Like**
Bitsliced	*Hamsi*		*JH*		*Keccak*	*Luffa*
AES	*Shavite3*		*Echo*	*Grostl*		*Fugue*
ARX	*Skein*	BLAKE	*BMW*		*Cube*	
Logical/ ARX			*SIMD*	*Shabal*		

- Source of nonlinearity (AES/bitslice/ARX)
- Shared design elements
- What else?
- Similarity of domain extenders (all sponges, all HAIFA, etc.)
- Lineage

Shared Design Elements, Nonlinearity, Lineage

Bitsliced	*Hamsi*	*JH*	*Keccak*	*Luffa*
AES	*Shavite3*	*Echo*	*Grostl*	*Fugue*
ARX	*Skein*	BLAKE	*BMW*	*Cubehash*
Logical/ARX	*SIMD*	*Shabal*		

- JH has much in common with AES-based designs
- Keccak is an outlier in Bitsliced category
- SIMD is much closer to ARX than Shabal
- BLAKE is based on something by Bernstein
- All the AES stuff is based on something by Daemen

Nonlinearity: What Can We Evaluate?

- Results published on hashes with each source of nonlinearity.
  - This suggests the community isn't entirely at a loss about how to attack these kinds of designs.
- All four strategies have a lot of existing analysis in block ciphers, hashes, stream ciphers.
  - ARX and ARX/Logical: MDx and SHAx designs, RC5/6, TEA, etc.
  - Bitslice: All the SP network cryptanalysis, Serpent
  - AES: All the AES and AES variant cryptanalysis

Fixed vs Keyed Permutations

Fixed Perm	Hamsi	JH	Keccak	Luffa	Grostl	Fugue	Cubehash
Keyed Perm	Shavite3	Skein	BLAKE	BMW	SIMD	Shabal	ECHO*

- Message modification allows very powerful attacks on hash functions
- Some designs eliminated this by getting rid of message schedule; others kept it.
- This seems like significant difference in designs, directly related to attacks.

* ECHO uses keyed permutation for salt and counter, not message.

Domain Extenders

Narrow-Pipe	**MD**	**Wide-Pipe**	**MD**	**Sponge**	**Sponge -Like**
Hamsi			JH	Keccak	Luffa
	Shavite3	*Echo*	Grostl[3]		Fugue[2]
Skein[1]	*BLAKE*		BMW	Cubehash	
		SIMD	Shabal[1]		
	HAIFA	*DESIGNS*			

1. Skein and Shabal introduce new “chaining modes” based on tweaks to block cipher
2. Fugue is quite different than the other designs
3. Grostl double-width is required by comp. fn.

Evaluating Hashes with New Domain Extenders

- Fairly easy to understand modes that expect pseudorandom behavior from compression functions
  - MD variants, including HAIFA and Skein
  - Hermetic Sponge
- Less clear what to require from modes that don't expect that
  - Cubehash, Luffa not hermetic sponges
  - Shabal doesn't require randomness from compress
  - Hamsi, Fugue not designed for pseudorandom behavior from one compress.

All at Once vs a Little at a Time

- Crypto community has much experience with “all-at-once” hash functions:
  - Expect pseudorandom behavior from compression function....
  - ...or at least something close (Cubehash, Shabal)
- Much less experience with “little bit at a time” modes:
  - This is reflected in sparser cryptanalysis, and in uncertainty about what qualifies as a meaningful attack.
  - Fugue, Hamsi, Luffa (sort-of)

Wrapping Up Design Diversity

- We want to minimize the chances that a single attack will wipe out all our finalists!
- Source of nonlinearity and shared design elements seem really important here.
- No message schedule = no message-modification attacks. This seems like another kind of diversity of design.
- Different domain extenders change what the attacks look like somewhat. Not clear how important this is.

Selection: Cryptanalysis

	Narrow-Pipe	**MD**	**Wide-Pipe**	**MD**	**Sponge**	**Sponge-Like**
Bitsliced	*Hamsi*		*JH*		*Keccak*	*Luffa*
AES	*Shavite3*		*Echo*	*Grostl*		*Fugue*
ARX	*Skein*	BLAKE	*BMW*		*Cube*	
Logical/ ARX			*SIMD*	*Shabal*		

Cryptanalysis and Design Results

- Broadly four kinds of information here:
  - What cryptanalysis has been published?
  - How much analysis has been done?
  - What proofs or other information about domain extenders exists?
  - How well do we understand how to attack/analyze design?

Published Cryptanalysis

- No designs have been broken.
- A few designs have had attacks that “dent” them or raise some questions.
  - It is often quite hard to know how much weight to give partial attacks.
- Big new idea in last couple years has been rebound attacks--including on Grostl, Echo, and JH.
- Many other clever new attacks

How Much Cryptanalysis?

- One interesting problem is that some designs have gotten little cryptanalysis, while others have gotten much cryptanalysis.
  - For example, Cube, Grostl, Blake, Skein, and BMW have all seen a significant number of published analyses.
  - Others, such as Fugue and Shavite3, have seen much less published analysis
- More analysis implies more confidence in our understanding of security.
- ...but may track with designs that are easier to attack, or simpler to understand.

What's Known about Domain Extenders?

- Most of submissions have some kind of proof underlying their domain extender
  - Indifferentiability
  - Reduction from finding collisions on hash to finding collisions on compression function
  - Fugue has very different kinds of proofs
- ...but not all do.
  - Not clear how much weight to give to this.
  - Real question is how much these results can guide cryptanalysis of compression function, permutation, etc.

Do We Understand How To Evaluate Components?

- Many partial attacks in MD hashes considered important, yet ignored in other domain extenders.
  - Example: pseudocollisions call MD hashes into question, yet they don't lead directly to an attack.
  - Sponges and wide-pipe MD designs can be based on permutations, making pseudocollisions and free-start preimages trivial to find.
  - Keccak, Cubehash, JH

Nonrandomness

- Symmetries in Cubehash
- Generalized birthday attacks on Grostl's compression function
- Nonrandomness in permutations of Luffa, Shabal, Hamsi, Shavite3

- Do any of these even matter, given the domain extenders?
- Or is this as much warning as cryptanalysts can give us right now?

Completely theoretical stuff

- Theoretical Preimages
  - Cubehash and JH have these
  - Hamsi may also have one, if Shamir's recent result holds up.
- Wide-pipe / narrow-pipe concerns
- Barring some new information, we'll broadly ignore these, as they appear to have no real-world impact ever.

Biggest Question: How to Evaluate Security Margin?

- How much weight should we give to best currently known attack?
- If nobody knows how to analyze something, best known attack isn't so meaningful!
- When is some attack on the compression function relevant, and when is it meaningless or unimportant?
- How useful is it to count papers?
  - Good news: more papers → better understood
  - Bad news: more papers → weaknesses/attacks

Performance

	Narrow-Pipe	**MD**	**Wide-Pipe**	**MD**	**Sponge**	**Sponge-Like**
Bitsliced	*Hamsi*		*JH*		*Keccak*	*Luffa*
AES	*Shavite3*		*Echo*	*Grostl*		*Fugue*
ARX	*Skein*	BLAKE	*BMW*		*Cube*	
Logical/ ARX			*SIMD*	*Shabal*		

Lots of Performance Data on Common Desktop/Laptop Platforms

- SUPERCOP/eBash stuff done by Dan Bernstein has been a big success
- Also several performance comparisons done by outside groups and NIST
- At SHA3 workshop this year, wonderful new results:
  - ASIC results
  - FPGA results
  - Embedded processor results

How Important Are These Numbers?

- Every platform has some users who really want hashes to be fast and small there.
- Easiest to measure desktop performance
- How often is hash function performance critical to application performance?
- In constrained environments, seems like resource usage is more important
  - Not “how fast?” but “how big?” or “can I get it to work at all?”

Measuring Performance

- Quite a bit of variation across platforms and implementations
- ASIC, FPGA, and desktop numbers widely divergent
- Following drawn from some internal representative desktop numbers, Guo et al (SHA3 Conference) and Gaj et al (SHA3 Conference)

Desktop

- BMW
- Shabal
- Skein
- SIMD
- Luffa
- Keccak
- Blake
- JH
- Cube
- Grostl
- Hamsi
- Shavite3*
- Echo*
- Fugue

ASIC throughput

- Luffa
- Keccak
- Cube
- Hamsi
- Blake
- Grostl
- SHAvite3
- JH
- BMW
- Shabal
- Skein
- Echo
- Fugue
- SIMD

FPGA (ratio)

- Keccak
- Cube
- Luffa
- JH
- Grostl
- Shabal
- Blake
- Skein
- SHAvite3
- Fugue
- Hamsi
- BMW
- Echo
- SIMD

Patterns that Jump Out of This Data:

- ARX algorithms often optimized for S/W, not so great on H/W
  - Skein, BMW, SIMD, Shabal
- AES-based algorithms tend to be slow in S/W
  - Not so great in H/W either
  - But AES instruction *really* speeds up SHAvite3 and Echo
- Bitsliced designs do pretty well in H/W and S/W
  - Keccak, Luffa do well, JH does okay
  - Hamsi doesn't seem to do as well

Again, How Much Do We Care?

- How much weight should we give these performance numbers?
- We have less data on H/W—how much weight should we give that?
- Clearly most important requirement is that SHA3 can run almost anywhere (RAM, ROM, gate count)
- Don't want to overemphasize performance
  - Think of MD5—we got used to very fast hashes
- But who will use SHA3 if it's half the speed of SHA2?

Questions and Wrapup

???

	Narrow-Pipe	MD	Wide-Pipe	MD	Sponge	Sponge-Like
Bitsliced	*Hamsi*		*JH*		*Keccak*	*Luffa*
AES	*Shavite3*		*Echo*	*Grostl*		*Fugue*
ARX	*Skein*	BLAKE	*BMW*		*Cube*	
Logical/ ARX			*SIMD*	*Shabal*		

Tell Me What I Got Wrong!!!

- What criteria SHOULD we be including that we're not?
- What criteria should we be IGNORING?
- What really matters w.r.t. performance?
- What kind of design diversity matters?
  - Sources of nonlinearity, domain extenders, ancestry of design elements, etc.
- How can we estimate security margin?
  - Counting published papers to get confidence?

Email me at john.kelsey (at) nist.gov

Or talk to me here today or tomorrow