The New SHA3
Hash Functions

John Kelsey, NIST
What is a Hash Function?

\[ H = hash(message) \]

- Variable-length (long) input \( \rightarrow \) fixed-length (short) output
  - Typically 256 or 512 bit output.
- Use a hash function as a "message fingerprint."
  - Digital signatures are always of the hash of a message, rather than actual message.
What Can Go Wrong?

- Collision: Find two messages with same hash
  - Find $M, M^*$ so that $\text{hash}(M) = \text{hash}(M^*)$
  - Can never take more than $2^{n/2}$ work
  - 256-bit hash: 128 bits of collision resistance

**Good hash function → very hard to find collisions!**
What Else Can Go Wrong?

• Preimage: Given a hash, find a message

  • Given $H$, find $M$ so that $H = \text{hash}(M)$

  • Can never take more than $2^n$ work

  • 256-bit hash: 256 bits of preimage resistance

Good hash function $\Rightarrow$ very hard to find preimages
Why SHA3?

• Around 2004, lots of new attacks on hash functions came out
  • New attacks, new insights into structures
  • Raised questions about security of existing hash functions (SHA1, SHA2)
  • Lots of encouragement from community to have a competition for new hash standard
  • Modeled off the very successful AES competition
Requirements for SHA3

• Outputs of 224, 256, 384, 512 bits

• n-bit hash:
  For a 256-bit hash output:
  • $2^{n/2}$ collision resistance
  • $2^n$ preimage resistance

• Tunable parameter: allow a tradeoff between security and performance
The SHA3 Competition

• Five public workshops before and during competition.
• Started in 2008 with 64 submissions
• Narrowed down to 14 in 2009
• Further narrowed to 5 in 2010
• Winner announced 2012 – Keccak

Massive participation from crypto community
Keccak

- Innovative design
- Based on sponge construction
  - Includes tunable parameter--capacity
  - Allows a very clean tradeoff between performance and security.
- Also includes variable-length hash functions, and many other extras.
Keccak is a Sponge

- $c = \text{capacity} = \text{security parameter}$

- **Security against preimages and collisions:** $2^{c/2}$

- Smaller $c = \text{lower security} = \text{faster performance}$
Preimages, Collisions, and Keccak

- We required $2^n$ preimage resistance for submissions

- **Security against preimages and collisions:** $2^{c/2}$

- Getting $n$ bit preimage resistance = performance hit
Keccak in Competition

*n-bit Keccak versions had 2n bit capacities*

<table>
<thead>
<tr>
<th>Keccak -256</th>
<th>Collision Resistance</th>
<th>Preimage Resistance</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submitted</td>
<td>128</td>
<td>256</td>
<td>512</td>
</tr>
</tbody>
</table>
Plans to Standardize SHA3

• Extensive discussions with Keccak team on how to move forward with SHA3

• Presented plans at public crypto conferences: RSA Conference, DIMACS, IETF, CHES

• Posted discussions on our public hash-forum mailing list
What to Standardize

- Fixed-length hash functions intended as drop-in replacements for SHA2
  - SHA3-224, SHA3-256, SHA3-384, SHA3-512
  - Intended as “drop-in replacements” for SHA2
- Variable-length hash functions supporting 128 and 256 bit security level
  - SHAKE128 and SHAKE256
- New kind of cryptographic object
Changes to Keccak

- Change padding, simplify spec by having fewer different capacities

- *Reduce capacities: better performance, lower theoretical security*

<table>
<thead>
<tr>
<th>Keccak</th>
<th>-256</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Collision Resistance</td>
<td>Preimage Resistance</td>
</tr>
<tr>
<td>Submitted</td>
<td>128</td>
</tr>
<tr>
<td>Proposed Feb 2013</td>
<td>128</td>
</tr>
</tbody>
</table>
Why reduce capacity?

• Large capacities needed to get n bit preimage resistance
  • Imposes a significant performance penalty

• Practical security of hashes is collision security

• Practical increase in performance for theoretical loss of security.

<table>
<thead>
<tr>
<th>Keccak  -256</th>
<th>Collision Resistance</th>
<th>Preimage Resistance</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submitted</td>
<td>128</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td><strong>Proposed Feb 2013</strong></td>
<td><strong>128</strong></td>
<td><strong>128</strong></td>
<td><strong>256</strong></td>
</tr>
</tbody>
</table>
We got a lot of feedback about reducing capacity

• New SHA3 would not have met competition requirements
• Violated intuitions of hash function security
• Might break drop-in replacement property
• *Loses big advantage of public competition*
  • *Open process, lots of community participation*
  • *Making substantial changes undermines these benefits*
Current Plans for SHA3

- Standardize Keccak fixed hash functions with original capacities
  - \( n \)-bit hash \( \rightarrow \) \( 2n \) bit capacity \( \rightarrow \) \( n \)-bit preimages
  - Also standardize variable-length hashes at 128 and 256 bit security levels.

<table>
<thead>
<tr>
<th>Keccak -256</th>
<th>Collision Resistance</th>
<th>Preimage Resistance</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submitted</td>
<td>128</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td><strong>Proposed Feb 2013</strong></td>
<td><strong>128</strong></td>
<td><strong>128</strong></td>
<td><strong>256</strong></td>
</tr>
<tr>
<td><strong>Current Proposal</strong></td>
<td><strong>128</strong></td>
<td><strong>256</strong></td>
<td><strong>512</strong></td>
</tr>
</tbody>
</table>
Questions / Lessons

• How do we know when we’re getting meaningful feedback?

• *Beware the silent (disgruntled?) majority*

• How do we move from a competition to writing a standard?