The New SHA3 Hash Functions

John Kelsey, NIST

What is a Hash Function?

H = hash(message)

- Variable-length (long) input → 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 hash(M)=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 = hash(M)*
  - Can never take more than 2^n work
  - 256-bit hash: 256 bits of preimage resistance

Good hash function → 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 *128-bits vs collisions*
  - 2^{n} preimage resistance *256-bits vs preimages*
- 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

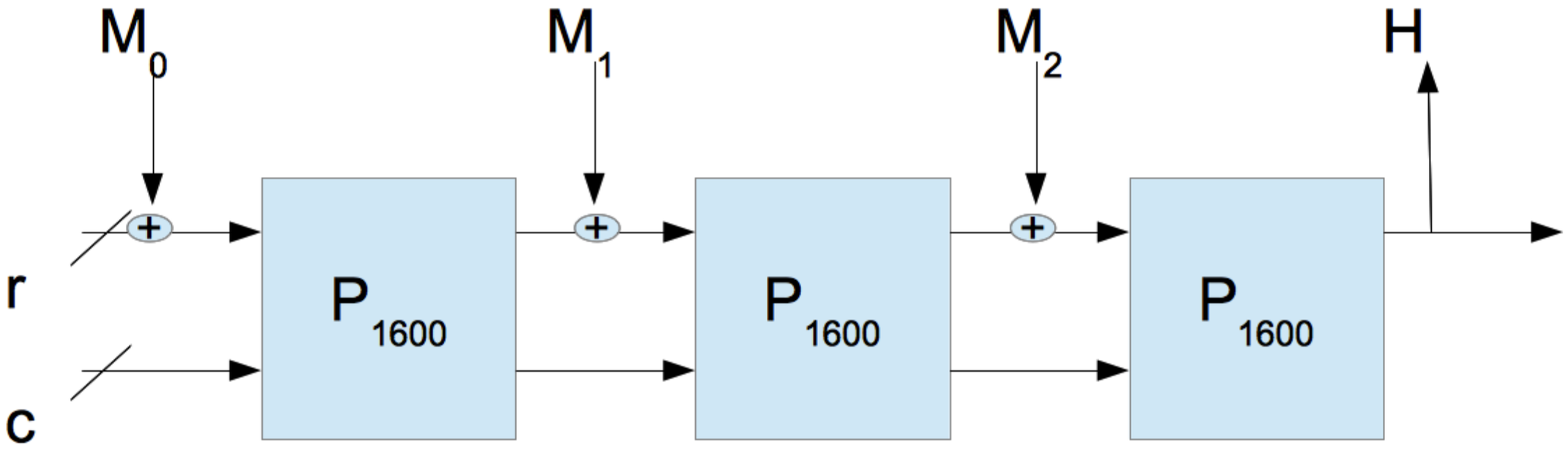
Massive participation from crypto community

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.

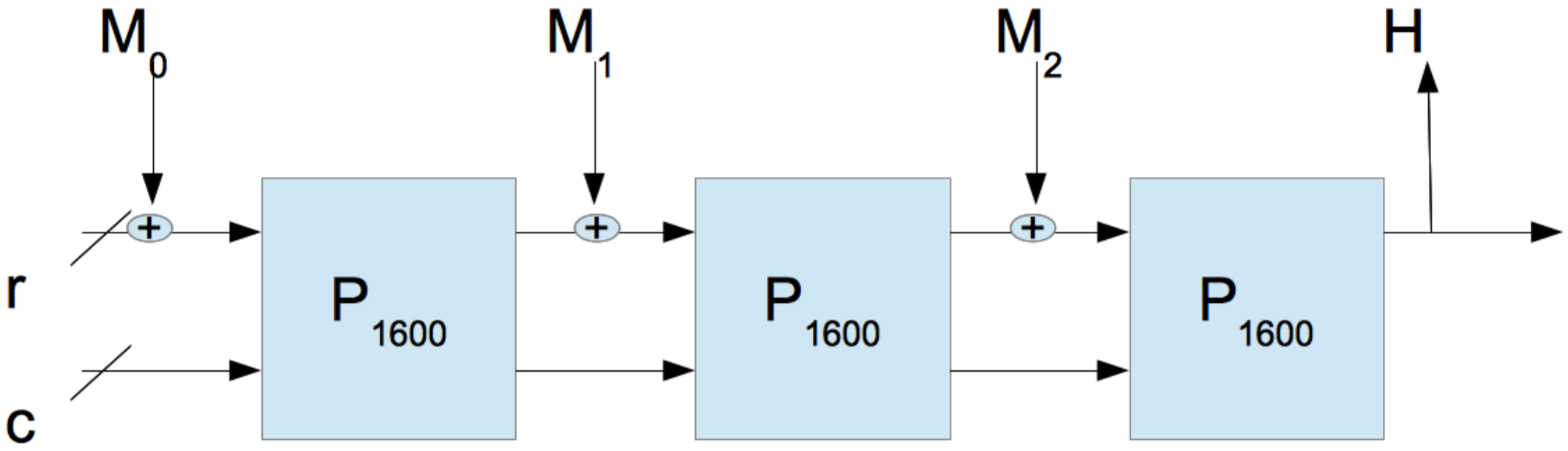
Keccak is a Sponge

- c = capacity = security parameter
- **Security against *preimages and collisions*: $2^{c/2}$**
- Smaller c = lower security = faster performance



Preimages, Collisions, and Keccak

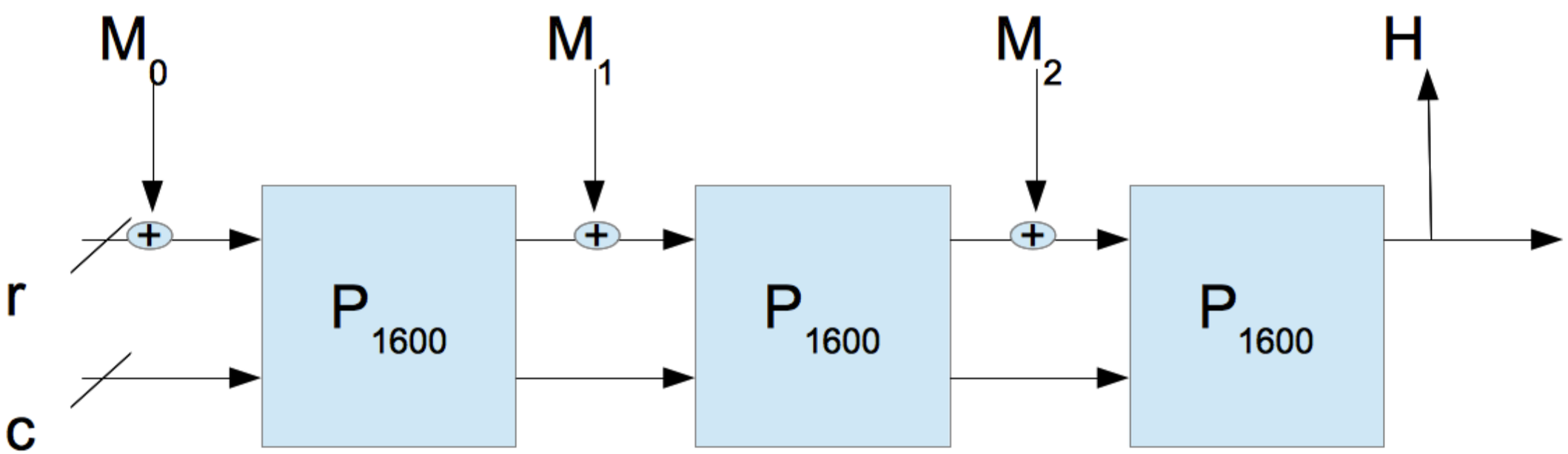
- We required 2^n preimage resistance for submissions
- **Security against *preimages and collisions*: $2^{c/2}$**
- Getting n bit preimage resistance = performance hit



Keccak in Competition

n-bit Keccak versions had 2n bit capacities

Keccak -256			
	Collision Resistance	Preimage Resistance	Capacity
Submitted	128	256	512



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***

Keccak -256			
	Collision Resistance	Preimage Resistance	Capacity
Submitted	128	256	512
Proposed Feb 2013	128	***128***	***256***

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.***

Keccak -256			
	Collision Resistance	Preimage Resistance	Capacity
Submitted	128	256	512
Proposed Feb 2013	128	***128***	***256***

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 → 2n bit capacity → n-bit preimages

- Also standardize variable-length hashes at 128 and 256 bit security levels.

Keccak -256			
	Collision Resistance	Preimage Resistance	Capacity
Submitted	128	256	512
Proposed Feb 2013	128	***128***	***256***
Current Proposal	**128**	***256***	***512***

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?