Hash Functions

Bimal Roy
Indian Statistical Institute, Kolkata)

February, 2018

Hash Function

- Mapping of arbitrary length message to a fix length bit string called digest.
- $h : \{0,1\}^* \rightarrow \{0,1\}^d$ (d is digest length).
- Digest hides possible structure in message.

Desirable Properties

- Deterministic and fast.
- Infeasible to generate a message from digest (preimage)
- Small change in the message produce an uncorrelated digest (second preimage)
- Infeasible to find two different messages with the same digest (collision).

Hash Function

Collision Resistance

- Hard to find $m, m' \in \{0,1\}^*$ for which h(m)=h(m').
- Usage : Commitment, Signature.

Preimage Resistance

- Given $D \in \{0,1\}^d$, hard to find m such that $h(m) = D$.
- Usage : Commitment.

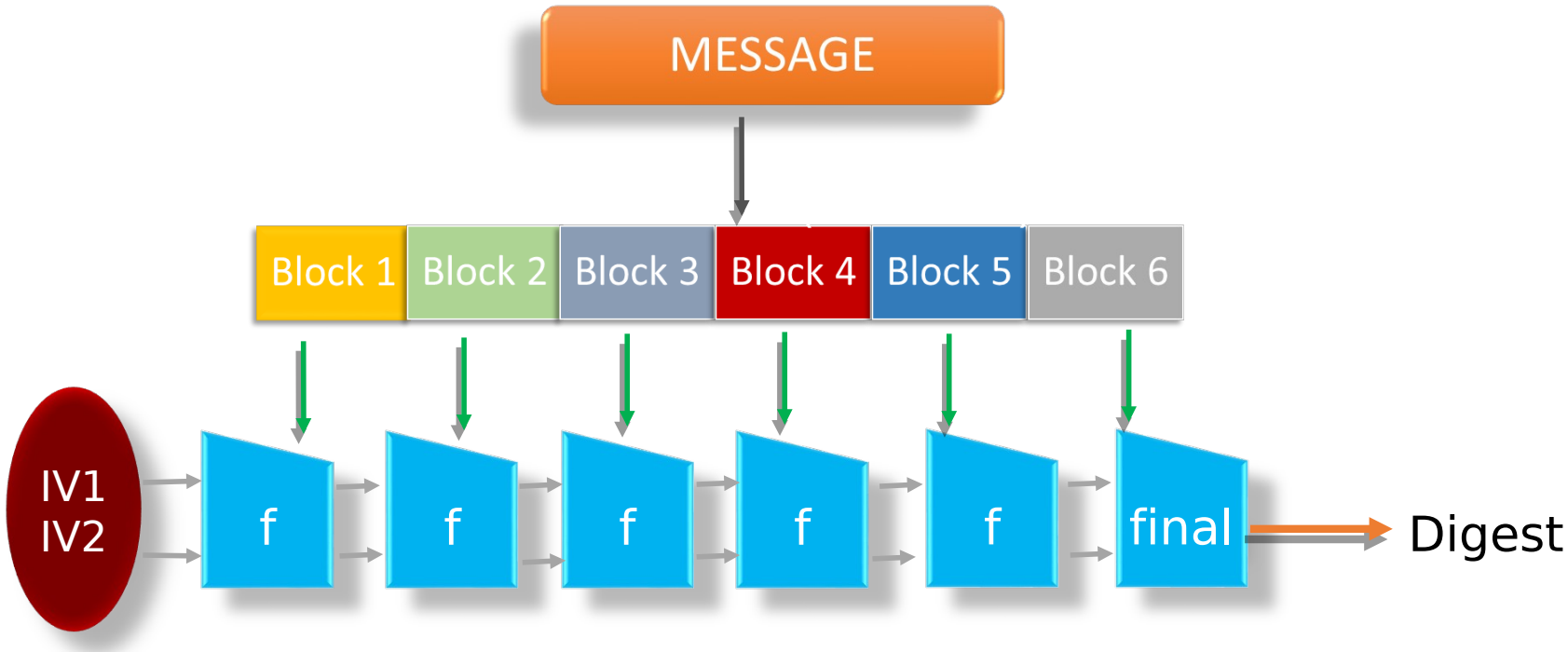
Second Preimage Resistance

- Given $m \in \{0,1\}^*$, hard to find $m' \neq m$ such that $h(m') = h(m)$.
- Usage : Commitment.

Applications

- Message authentication.
- Software integrity.
- One time password.
- Digital signature.
- Time stamping.

Merkle Damgard Hash



Merkle Damgard Hash

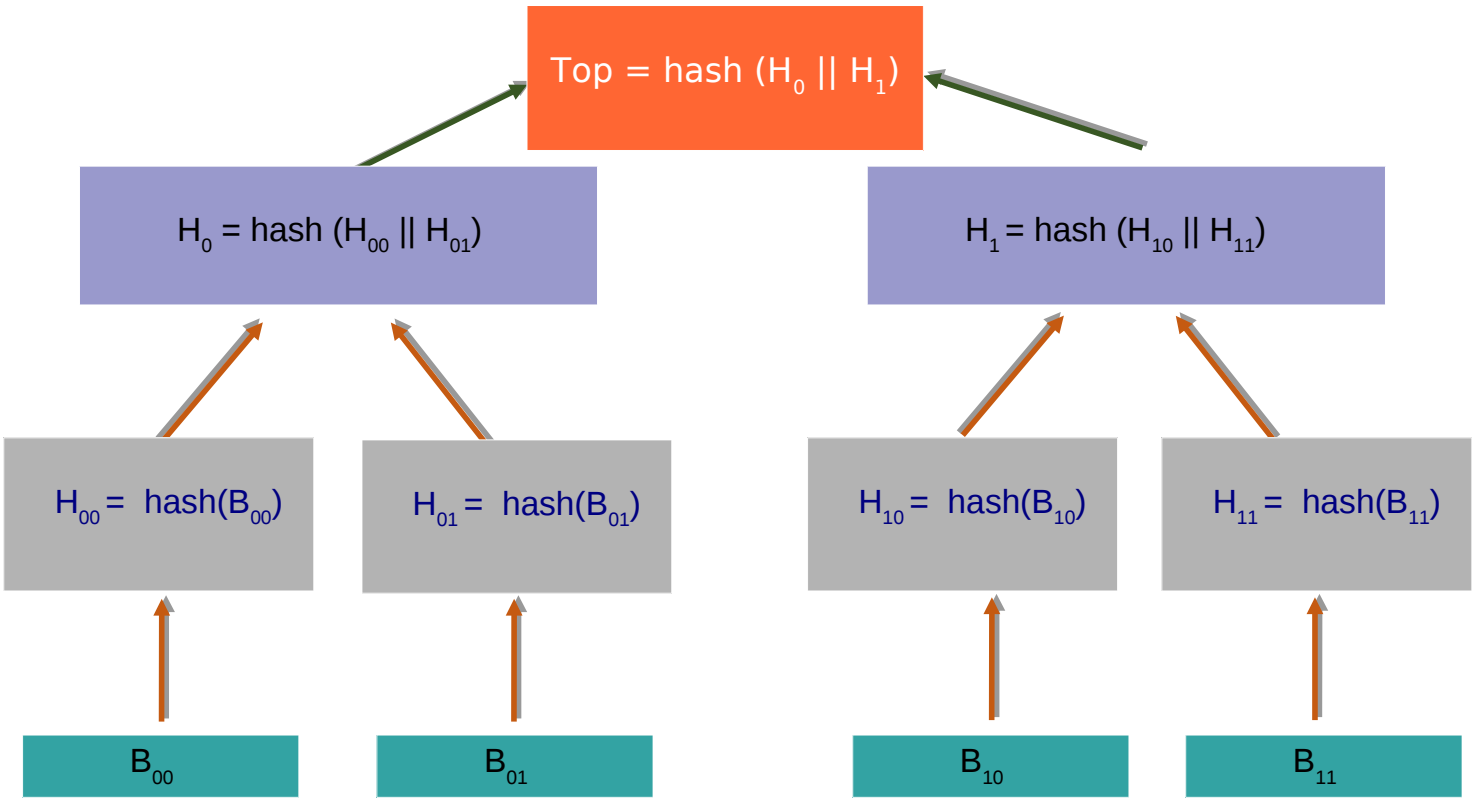
Goal

- Construction of a hash function $h : \{0,1\}^* \rightarrow \{0,1\}^d$ from a compression function $f : \{0,1\}^{d+t+1} \rightarrow \{0,1\}^d$.

Examples

- MD5 : Specified as Internet standard RFC1321, popular hash.
- SHA-1 designed by NSA.

Merkle Tree


Top = hash (H_0 || H_1)
H_0 = hash (H_{00} || H_{01})
H_1 = hash (H_{10} || H_{11})
H_{00} = hash(B_{00})
H_{01} = hash(B_{01})
H_{10} = hash(B_{10})
H_{11} = hash(B_{11})
B_{00}
B_{01}
B_{10}
B_{11}

Merkle Tree

- Leaf nodes are labelled with hash of a data block $H_{ij} = h(B_{ij})$.
- Non leaf nodes are labelled with cryptographic hash of labels of its child nodes.
- Used in
  - Used for set-membership proof at log time.
  - Hash-based signatures.
  - Blockchain Protocol.
  - Distributed hash table.
  - ZCASH - cryptocurrency protocol with anonymous payments..

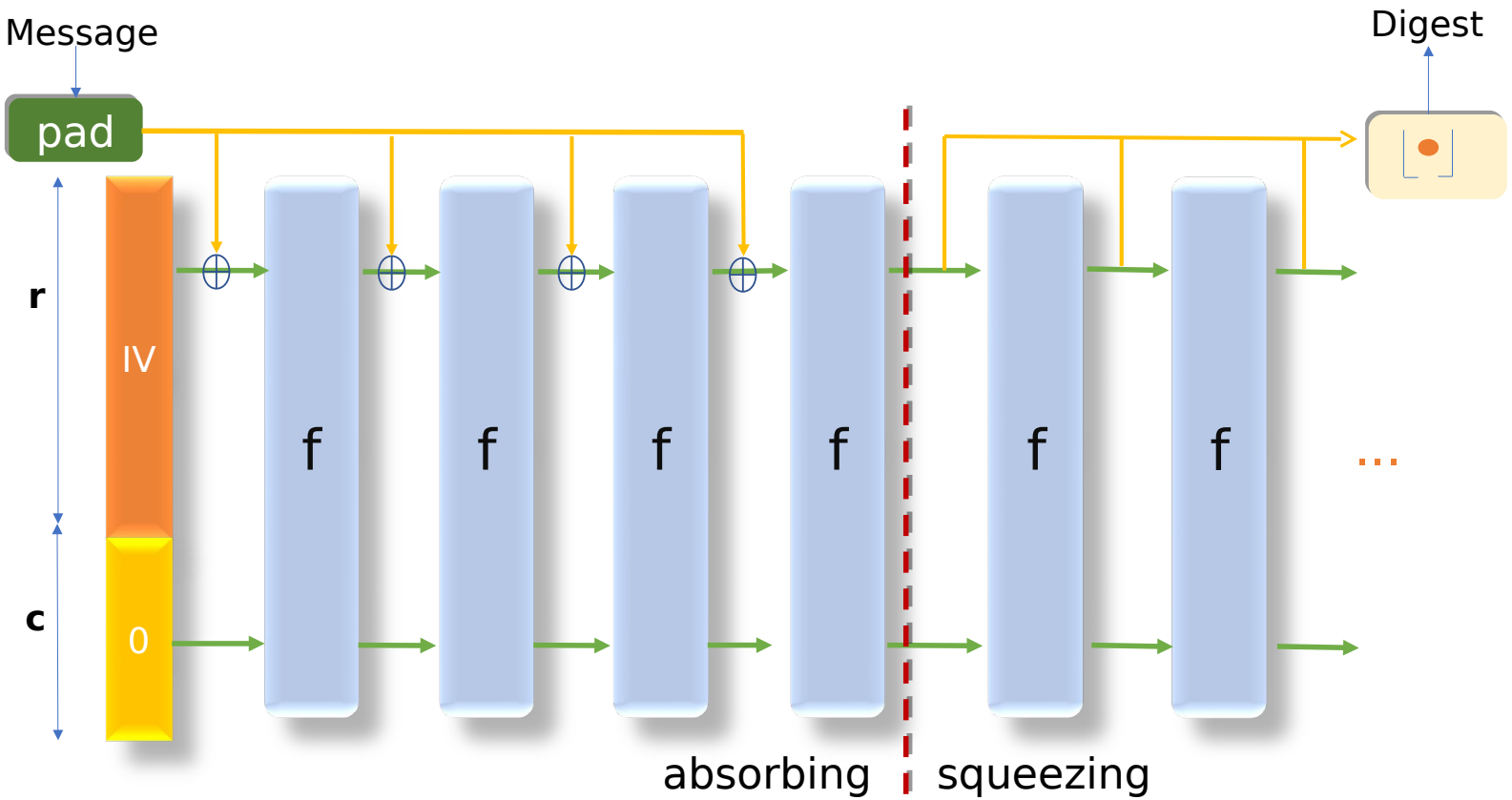
Before SHA-3

- MD4 was to be broken by Dobbertin, but still used occasionally.
- MD5 have theoretical weaknesses but still widely used.
- SHA-1 was thought to be very strong.
- SHA-2 looked like the future, with security up to 256 bits.
- MD was normal way to build hashes.

SHA-3

- 02/11/2007: Call for Proposals.
- 31/10/2008: Submission deadline.
- 10/12/2008: First-round candidates announced.
- 24/07/2009: Second-round candidates announced.
- 09/12/2010: SHA-3 finalists announced.
- 02/10/2012: Keccak announced as the SHA3 winner.

Sponge



Sponge

- Introduced in SHA-3 winner Keccak Specification.
- Calls a b-bit permutation f.
- Memory friendly mode: Implementation stores only the b-bit state.
- r is called rate and $c = b - r$ is called capacity
- Efficiency depends on r. Security depends on c.

Usability of Sponge

- Duplex mode sponge: Used to design MAC or AE.
- Getting popular for lightweight cryptographic design.
- Variants of duplex mode: SpongeWrap, Monkey duplex etc.
- Several designs: KETJE, KEYAK, ASCON, PRIMATEs etc.

Thank you