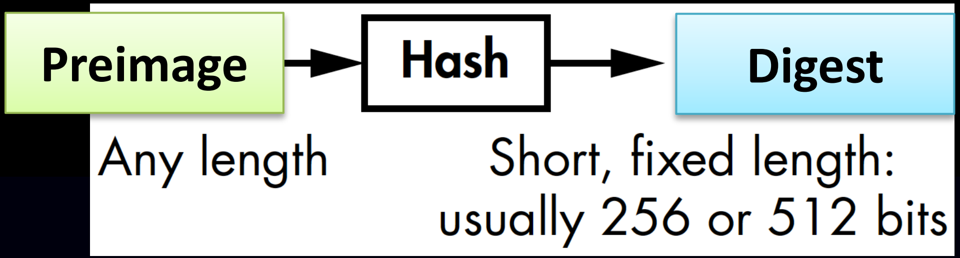
Applied Cryptography
CPEG 472/672
Lecture 6A

Instructor: Nektarios Tsoutsos

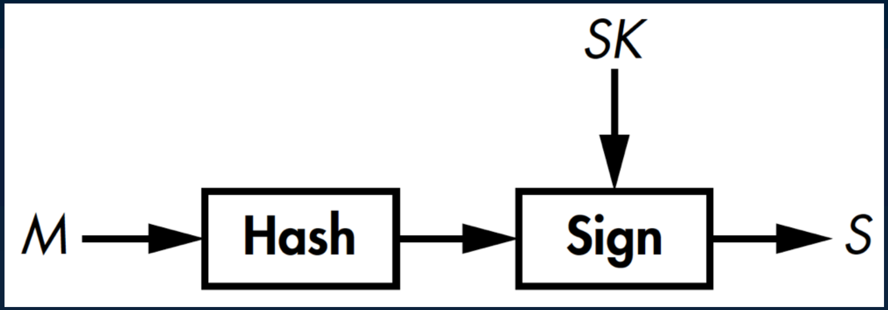
Hash functions



- Cryptographer's Swiss Army Knife
  - Any length input -> short digest
- Examples
  - MD5, SHA-1, SHA-256, SHA-3, BLAKE2
  - Digital signatures, public key encryption
  - integrity verification, key agreement
  - message authentication, password protection
  - Intrusion detection systems
  - Git, Bitcoin

Secure hash functions

- Notion of security in hash functions
  - Goal: ensure data hasn't changed
  - Different pieces of data must have different hashes
  - The hash serves as an identifier
  - Different notion compared to ciphers
- Case study: Digital signatures



Hashes are unpredictable

- Flipping one bit in the input, generates a completely different hash digest
- E.g., SHA256 hashes of "a", "b", "c"
  - ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb
  - 3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d
  - 2e7d2c03a9507ae265ecf5b5356885a53393a2029d241394997265a1a25aefc6

Security guarantees of hashes

- Preimage resistance (one-wayness)
  - Given H so that H=Hash(M), it is impossible to find the original M
  - There are infinite preimages that give the same hash H
    - Even if you have unlimited resources, it is not possible to find the exact message M
- Second-preimage resistance
  - Given M_1, can't find M_2 so that $H(M_2)=H(M_1)$
    - If you could find preimages, you could also find 2nd preimages

Security guarantees of hashes (2)

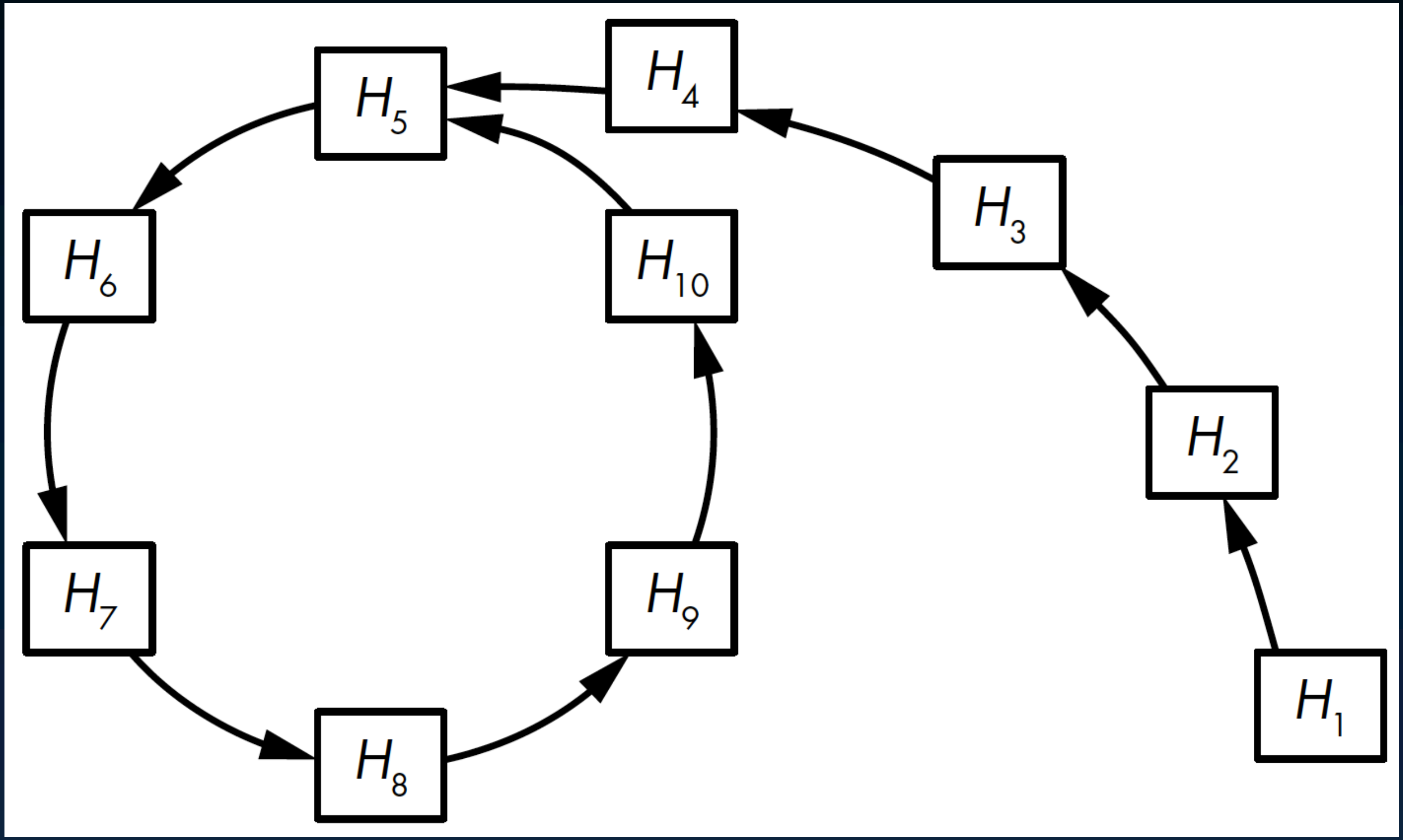
- Collision Resistance
  - Collisions are inevitable due to the pigeonhole principle
    - If you have m holes and n pigeons with n>m, at least one hole must have more than one pigeon
  - We want to hash functions where finding collisions is not possible
    - Collisions resistance is related to 2nd preimage resistance
    - Finding 2nd preimages, allows finding collisions
    - Collision resistant hash functions are also 2nd preimage resistant

Finding collisions

- Uses the Birthday Attack method
  - If hash can have up to N different digests as outputs, we will find a collision after hashing $\sqrt{N}$ messages with about 40% probability
- Memory efficient collision search
  - Rho method
  - Pick a random hash value $H_1 = H_1'$
  - $H_{i+1} = Hash(H_i)$, $H_{i+1}' = Hash(Hash(H_i'))$
  - Iterate until $H_{i+1} == H_{i+1}'$

Rho method to find collisions

- Hash(H4) = H5 = Hash(H10)

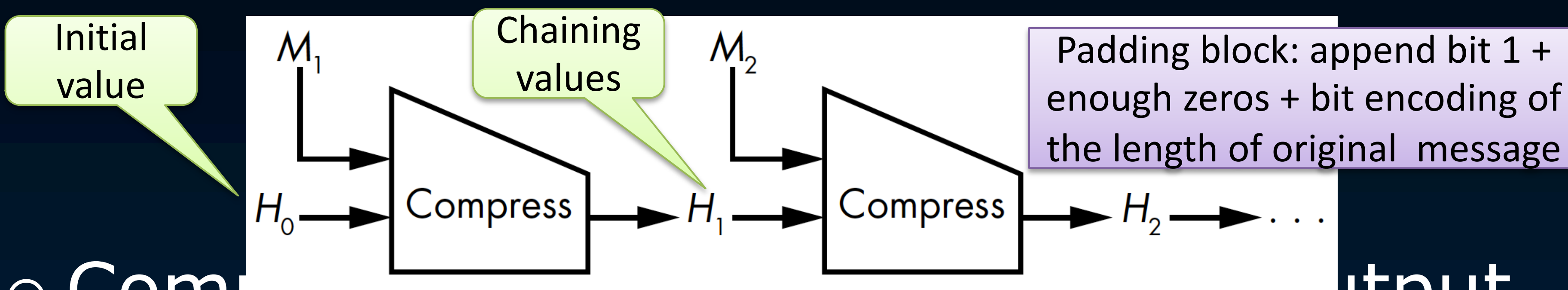


Building hash functions

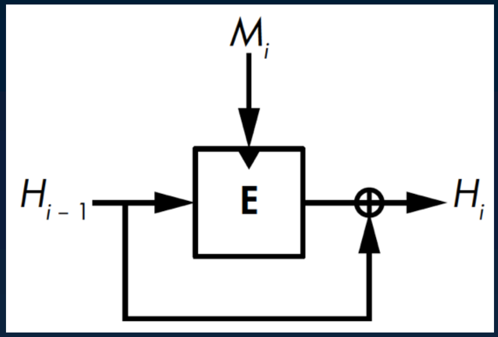
- Split message into blocks and process each block iteratively
  - Iterative hashing
- Two approaches
  - Using compression functions
    - Examples: MD5, SHA-1, SHA-2
  - Using sponge functions
    - Examples: SHA-3

Compression-based hashing

- Uses the Merkle Damgard (M-D) construction and a compression function



- Compress: Reduce 2 inputs in 1 output
  - Can use the Davies-Meyer construction
  - Secure block cipher E
    - M is the key input E
    - H is the ptxt input of E
  - M-D: Turn E into a secure hash

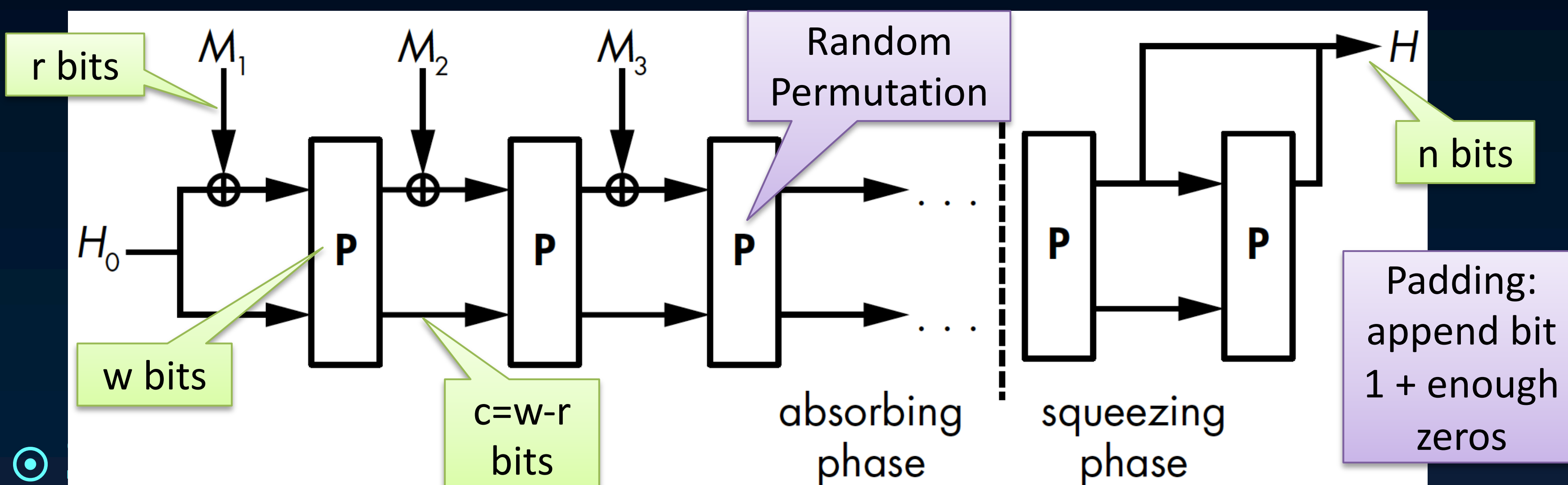


M-D security

- If Compress is preimage and collision resistant, then a hash constructed using M-D is also preimage & collision resistant
  - Any successful preimage attack on M-D is a successful attack on Compress
  - Breaking collision resistance of M-D means breaking collision resistance of Compress
- A collision in Compress does not necessarily give a collision on M-D hash

Sponge-based hashing

- Use a single permutation P instead of compression functions and block ciphers
  - Simpler than M-D functions (mostly XORs)



- Security level (bits) = MIN(c/2, n/2)

Reading for next lecture

- Aumasson: Chapter 6
  - From "*The SHA Family of Hash Functions*" to the end of the chapter
  - We will have a short quiz