CS 4770: Cryptography

CS 6750: Cryptography and Communication Security

Alina Oprea
Associate Professor, CCIS
Northeastern University

February 26 2018

Recap

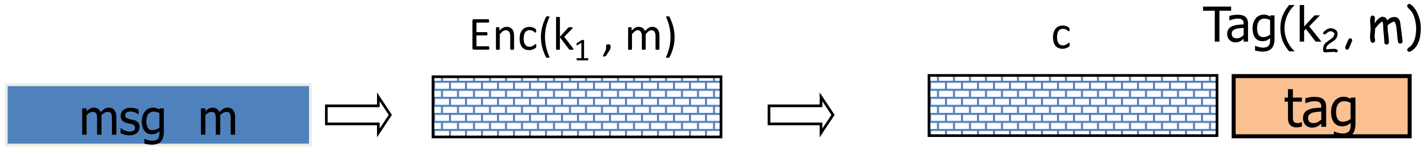
- Integrity vs confidentiality
  - Complementary properties
  - Both are needed in practice
- Message Authentication Codes (MAC)
  - MACs on single block (e.g., 128-bit) can be built from PRFs
  - CBC-MAC for integrity on longer messages
- Authenticated encryption
  - Combine CPA secure encryption and secure MAC into secure authenticated encryption scheme

Combining MAC and ENC (CCA)

Encryption key k_1. MAC key = k_2

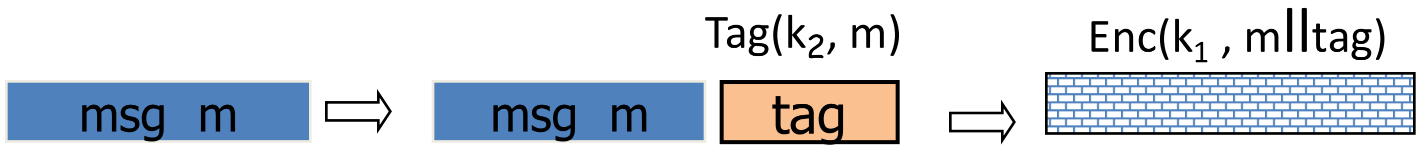
Option 1: (SSH)

Enc-and-MAC



Option 2: (SSL)

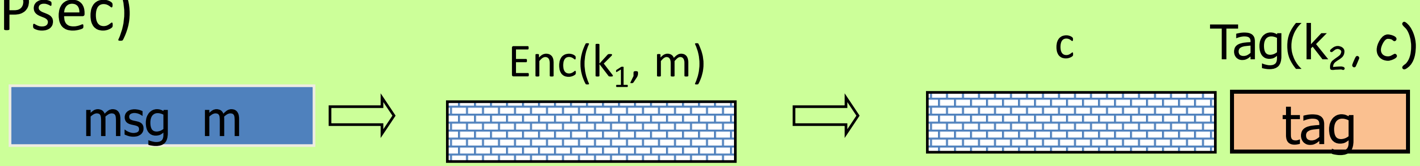
MAC-then-enc



Option 3: (IPsec)

Always correct

Enc-then-MAC



A.E. Theorems

Let (Enc,Dec) be CPA secure encryption and (Tag,Ver) secure MAC. Then:

1. **Encrypt-then-MAC** (IPSec): always provides A.E.

2. **MAC-then-encrypt** (SSL): may be insecure against CCA attacks

However: when (Enc,Dec) is rand-CTR mode or rand-CBC and no padding oracle available, Mac-then-Encrypt provides A.E.

Important: Encryption and MAC keys need to be independent

Outline

- TLS record protocol
  - MAC-then-Encrypt
  - Solution against replay attack
- Collision-resistant hash functions
  - Definitions
  - Examples
- Merkle-Daamgard transform
  - How to construct hash function from compression function
- Birthday paradox

The TLS Record Protocol (TLS 1.2)

$k_{b \rightarrow s}, k_{s \rightarrow b}$ $\qquad$ $k_{b \rightarrow s}, k_{s \rightarrow b}$

Unidirectional keys: $k_{b \rightarrow s}$ and $k_{s \rightarrow b}$

Stateful encryption:

- Each side maintains two 64-bit counters: $ctr_{b \rightarrow s}$, $ctr_{s \rightarrow b}$
- Init. to 0 when session started
- ctr++ for every record
- Purpose: replay defense

TLS record: encryption (CBC AES-128, HMAC-SHA1)

$k_{b\to s} = (k_{mac}, k_{enc})$

Browser side **Enc($k_{b\to s}$, data, $ctr_{b\to s}$) :**

Step 1: tag ⟵ **Tag**(k_{mac} , [++$ctr_{b\to s}$ || header || data])

Step 2: pad [header || data || tag] to AES block size

Step 3: CBC encrypt with k_{enc} and new random IV

Step 4: prepend header

TLS record: decryption (CBC AES-128, HMAC-SHA1)

Server side **Dec**$(\mathbf{k_{b \to s}}$ **, record,** $\mathbf{ctr_{b \to s}})$:

Step 1: CBC decrypt record using k_{enc}

Step 2: check pad format: send bad_record_mac if invalid

Step 3: check tag on [$++ctr_{b \to s}$ ll header ll data]

send bad_record_mac if invalid

Provides authenticated encryption

(provided no other information is leaked during decryption)

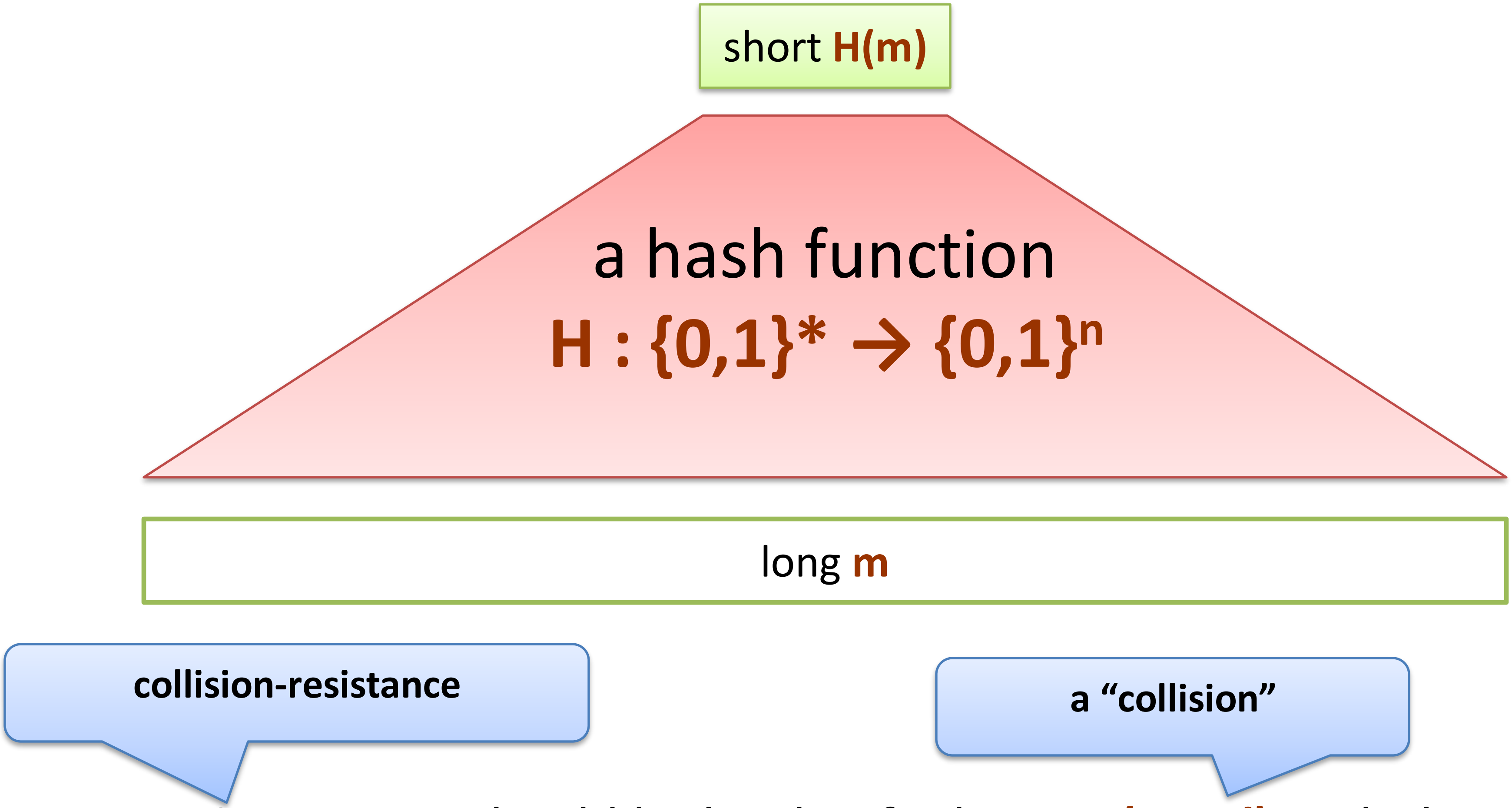
Review secret-key cryptography

- Stream ciphers
  - PRG (passive adversaries)
- Block ciphers
  - PRF, PRP (active adversaries, access to oracles)
  - Modes of operation to encrypt longer messages
- Integrity
  - Message Authentication Codes
- Authenticated encryption
  - Encrypt-then-MAC always secure
  - MAC-then-Encrypt secure only sometimes
- Practical attacks
  - Padding oracle has serious security implications

Hash functions

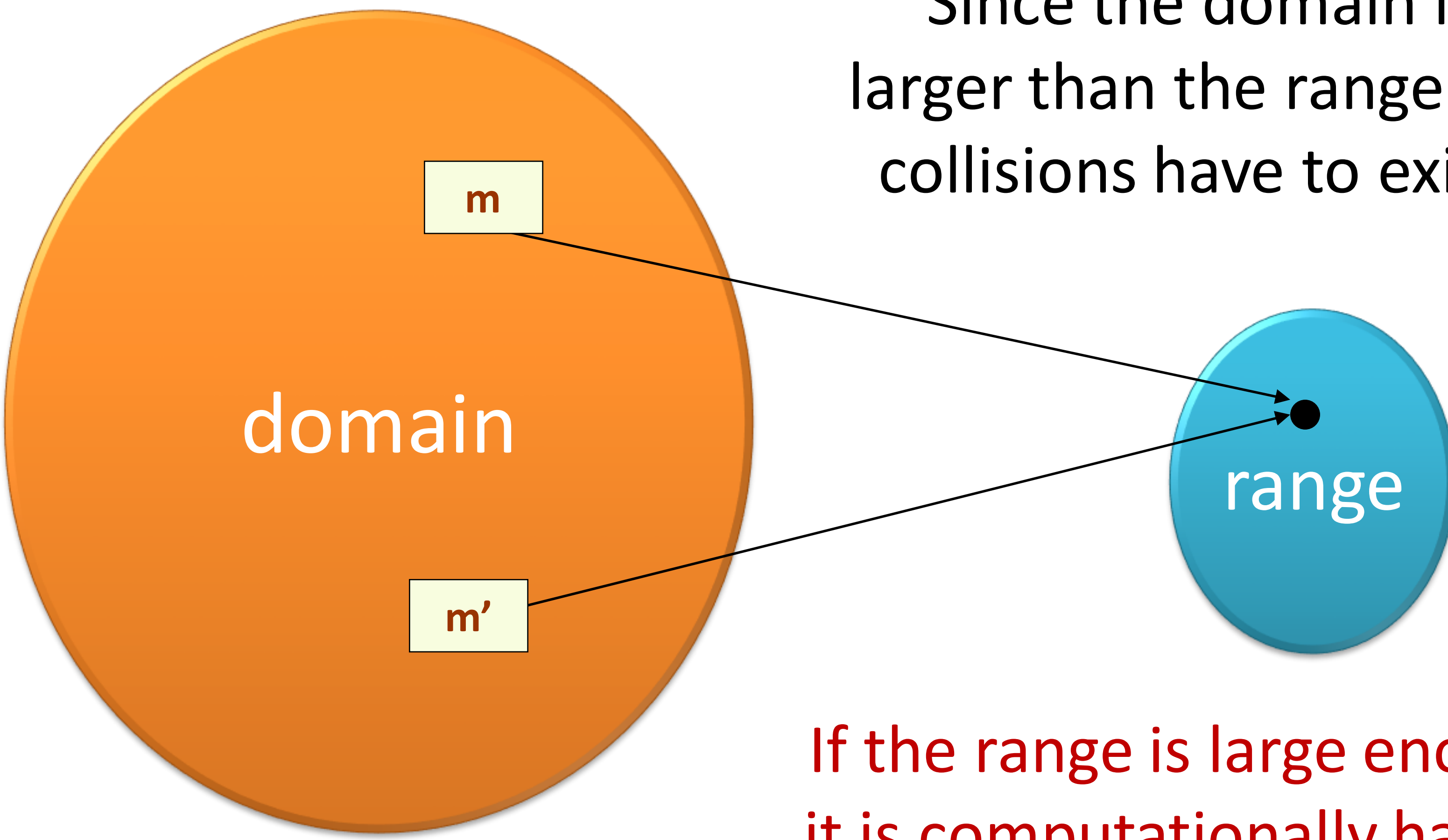
- Cryptographic primitive that does not rely on secret keys
- Many applications
  - Construction of HMAC
  - Password hashing
  - Integrity schemes (Merkle trees)
  - File similarity

Collision-resistant hash functions



Requirement: it should be hard to find a pair **(m,m')** such that **H(m) =H(m')**

Collisions always exist



Since the domain is larger than the range the collisions have to exist.

If the range is large enough, it is computationally hard to find collisions.

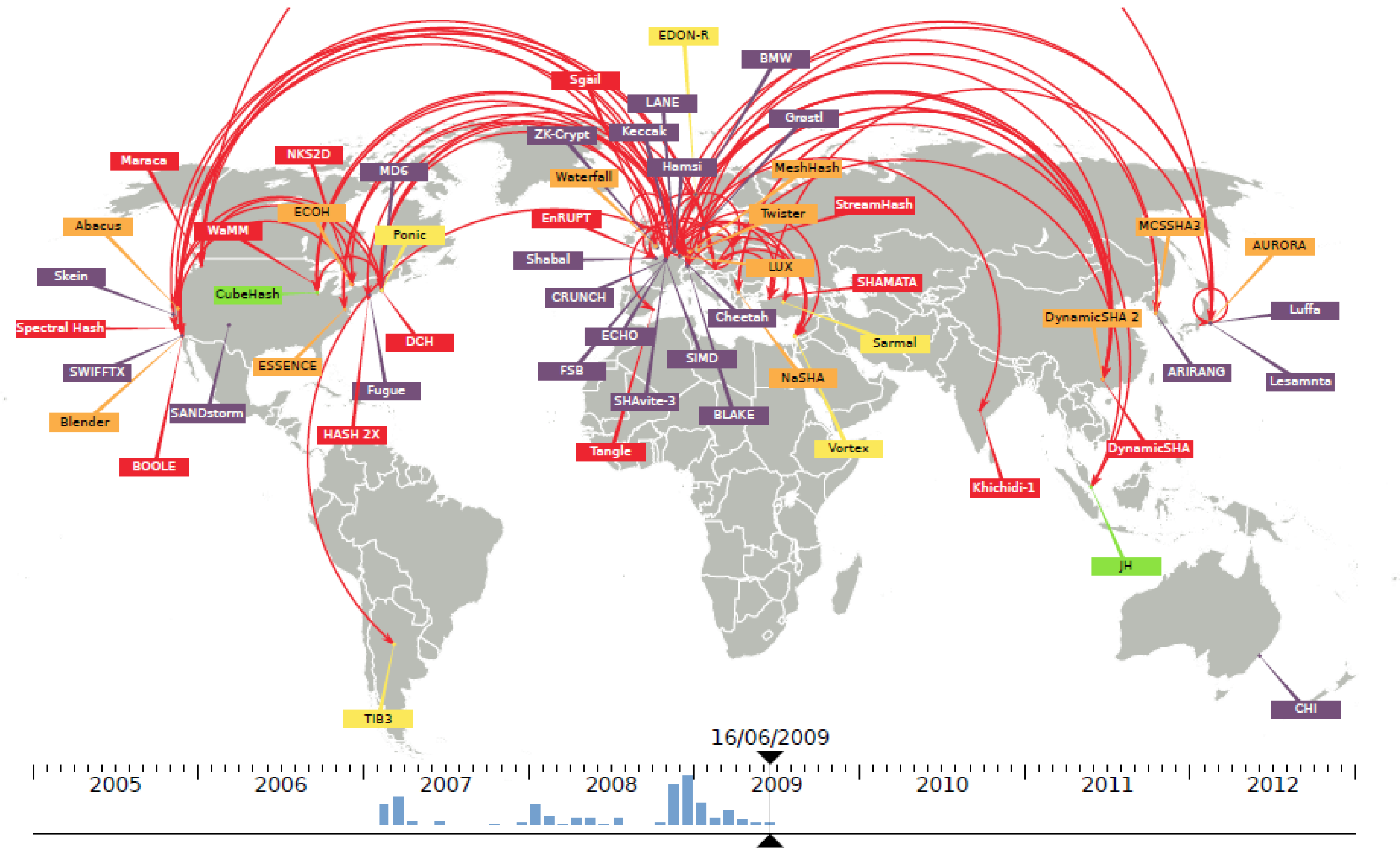
History of hash functions

H is a **collision-resistant hash function** if it is "*practically impossible to find collisions in **H***".

- **1991**: MD5
- **1995**: SHA1
- **2001**: SHA2 -- SHA-256 and SHA-512
- **2004**: Team of Chinese researchers found collisions in MD5
- **2007**: NIST competition for new SHA3 standard
- **2012**: Winner of SHA3 is Keccak

SHA-3 Competition

NIST SHA-3: the battlefield



[courtesy of Christophe De Cannière]

Hash functions – the security definition

security parameter
n

outputs **(m,m')**

H is a **collision-resistant hash function** if

$\forall$

polynomial-time
adversary **A**

Pr[A outputs m, m' such that H(m)=H(m')]
is negligible

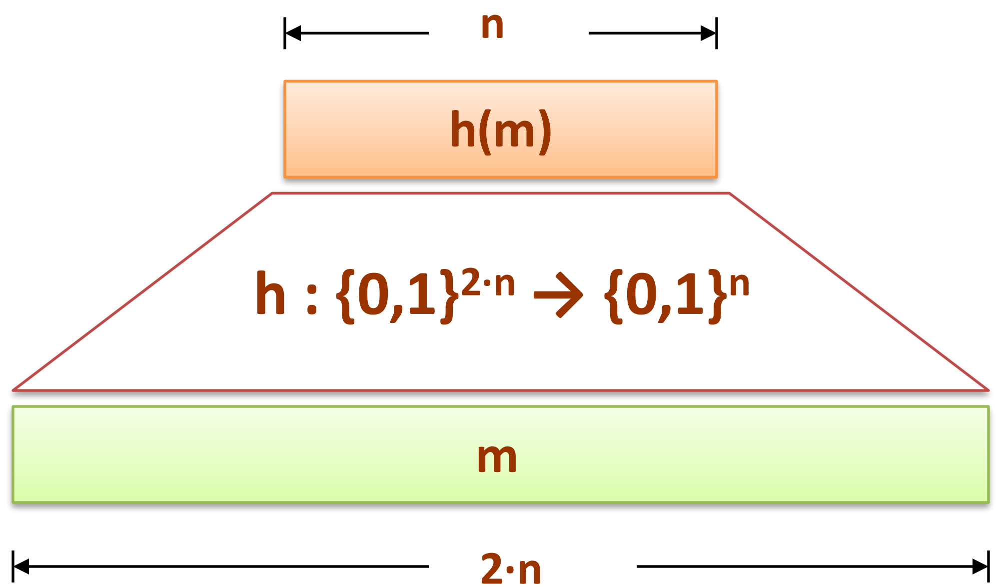
Examples

Are these hash functions collision resistant?

- $H:\{0,1\}^{2n} \rightarrow \{0,1\}^n$
  - $H(x||y) = x \text{ XOR } y$
- $H:\{0,1\}^{2n} \rightarrow \{0,1\}^n$
  - Let p be an n-bit prime
  - $H(x||y) = x + y \bmod p$
- $H: N \rightarrow \{0,1\}^n$
  - Let p be an n-bit prime
  - $H(x) = ax + b \bmod p$, p prime

A common method for constructing hash functions

1. Construct a "***fixed-input-length***" collision-resistant hash function

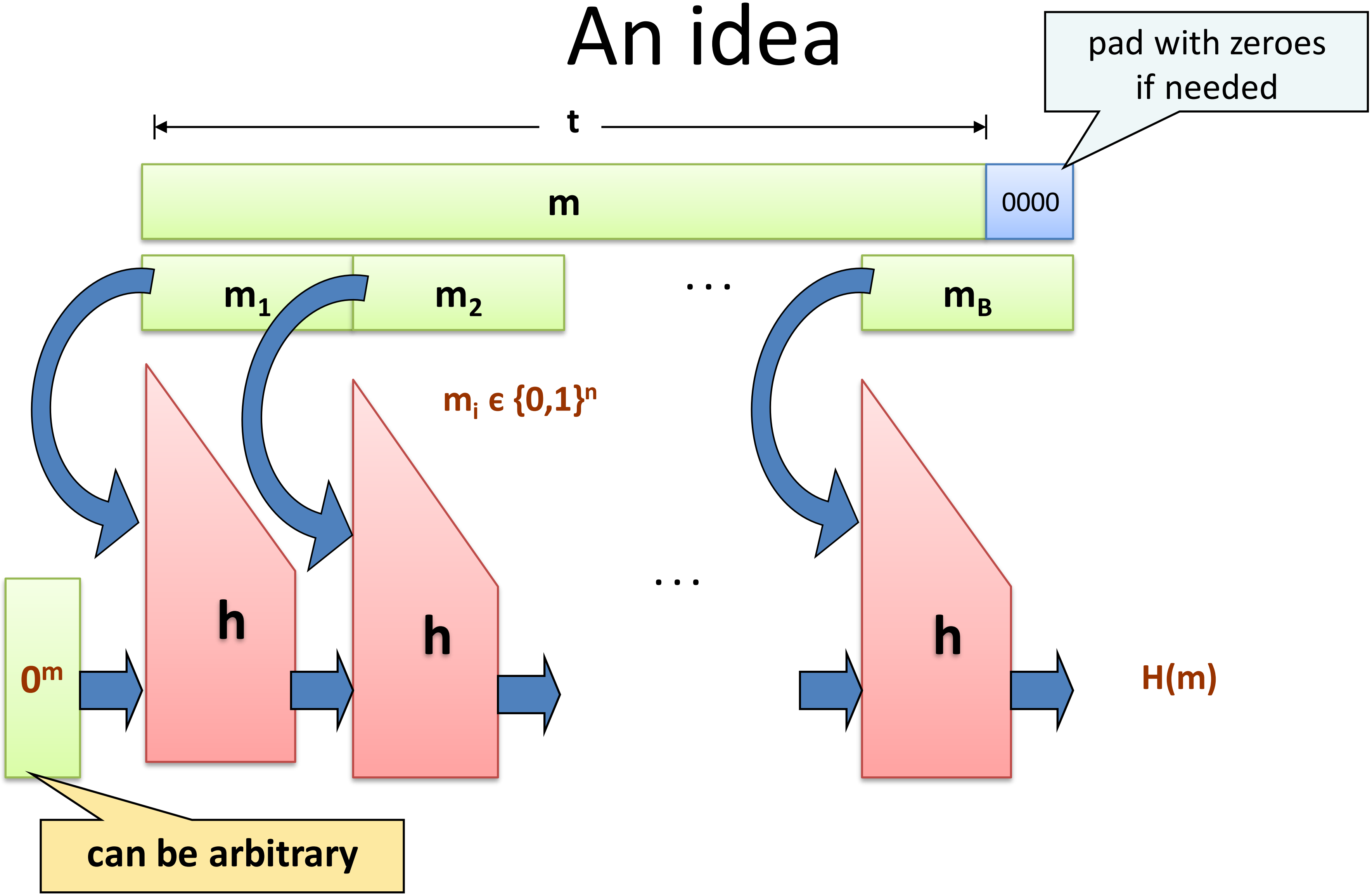


Call it: a collision-resistant **compression function**.

2. Use it to construct a hash function.

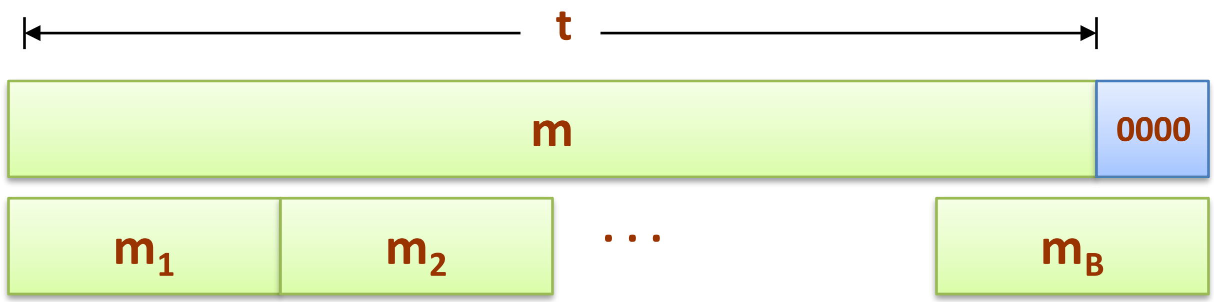
Used in SHA-1, SHA-2, but not in SHA-3!

An idea



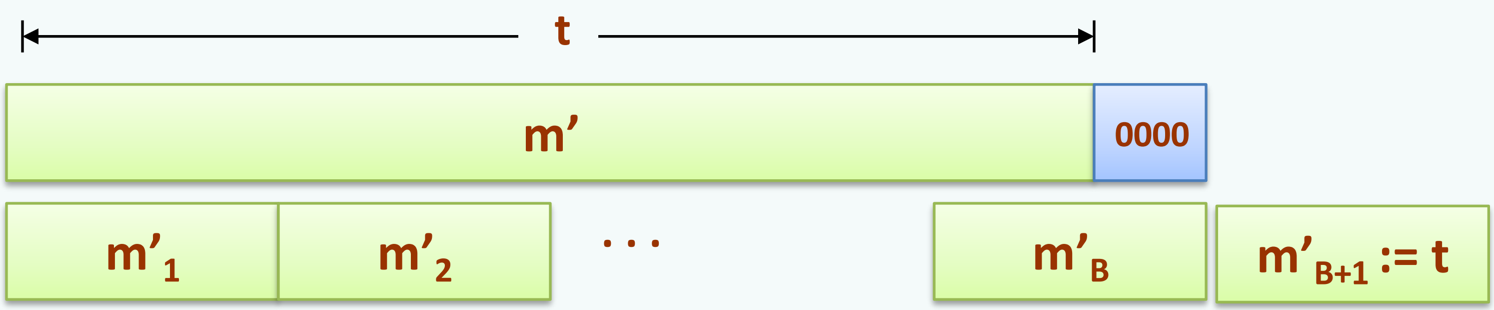
This doesn't work...

Why is it wrong?



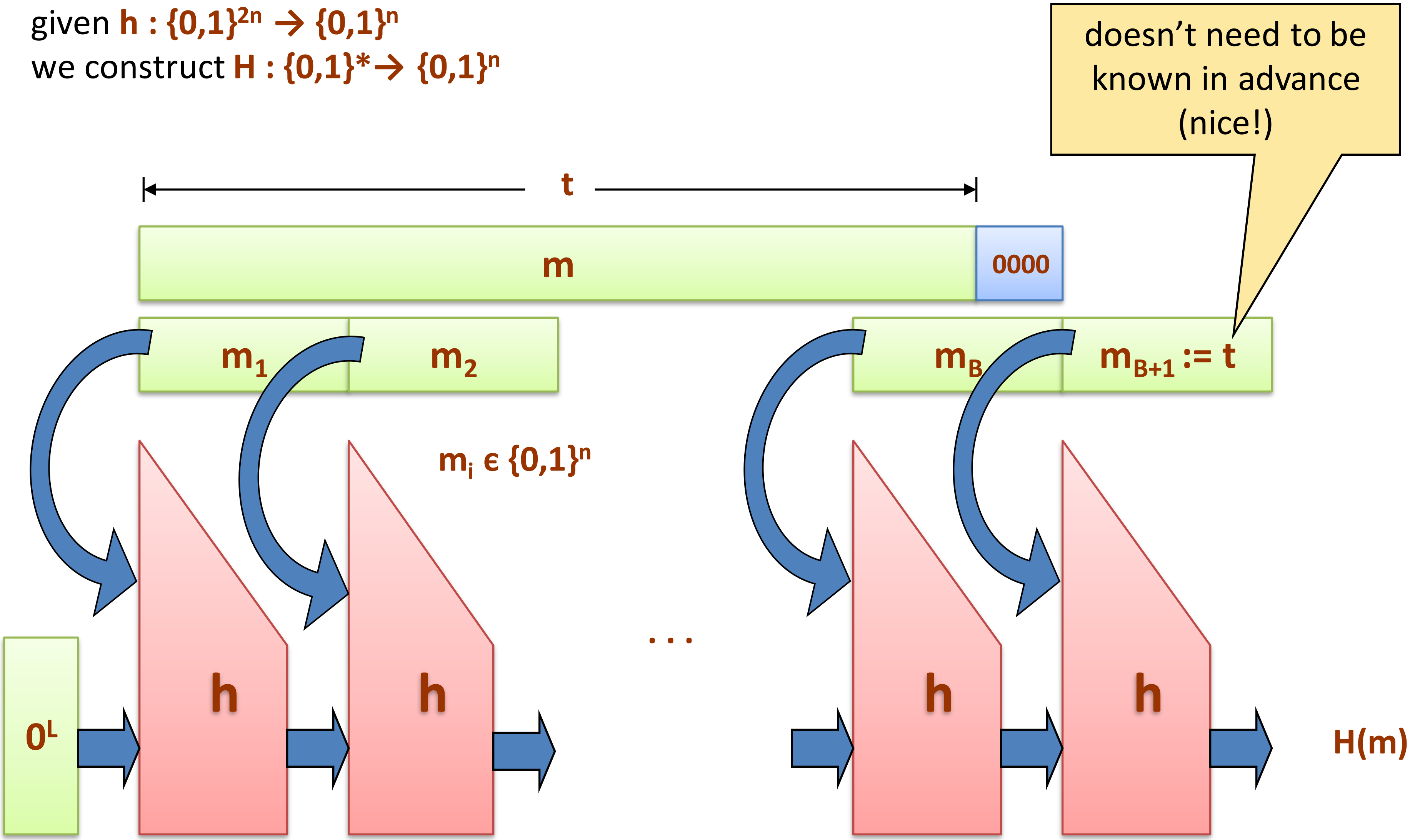
If we set **m' = m || 0000** then **H(m') = H(m).**

Solution: add a block encoding "**t**".



Merkle-Damgård transform

given **h : $\{0,1\}^{2n} \to \{0,1\}^n$**
we construct **H : $\{0,1\}^* \to \{0,1\}^n$**



This construction is secure

We would like to prove the following:

Theorem

If

$$h : \{0,1\}^{2n} \rightarrow \{0,1\}^{n}$$

is a collision-resistant **compression** function

then

$$H : \{0,1\}^{*} \rightarrow \{0,1\}^{L}$$

is a collision-resistant **hash** function.

Proof idea: convert collision on **H** into collision on h.

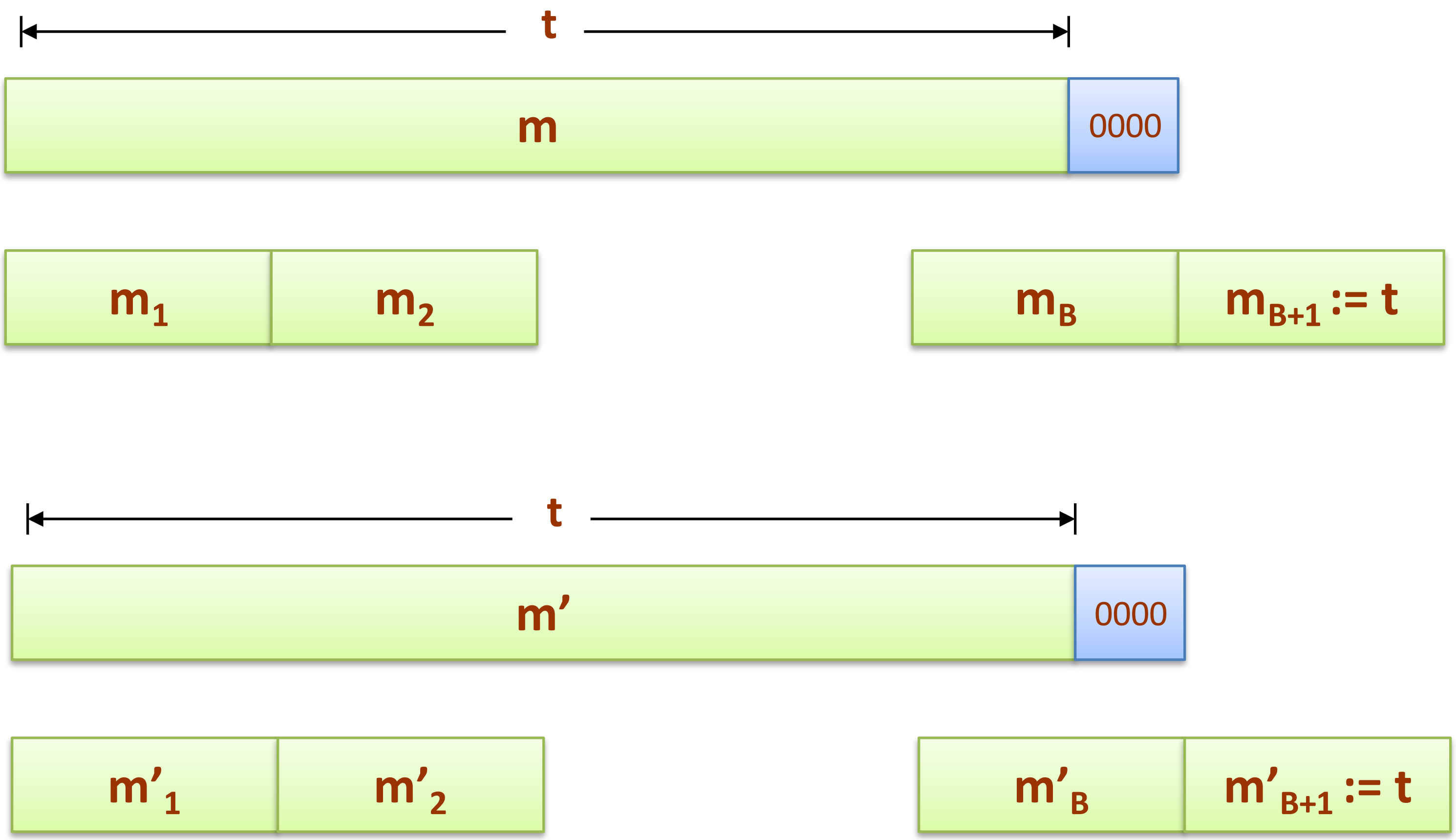
How to compute a collision **(x,x')** in **h** from a collision **(m,m')** in **H**?

We consider two options:

1. **$|m| = |m'|$**

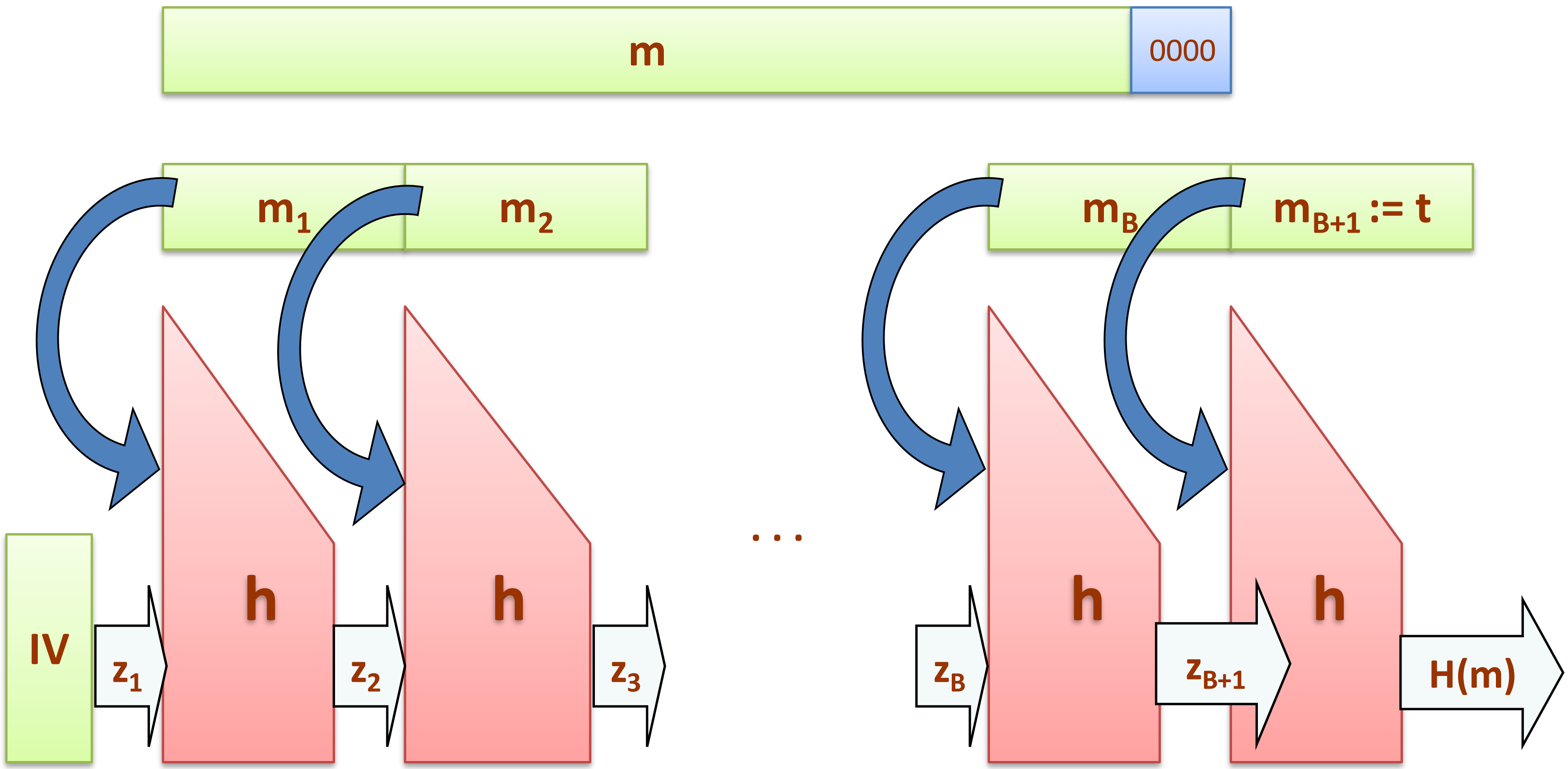
2. **$|m| \neq |m'|$**

Option 1: |m| = |m'|



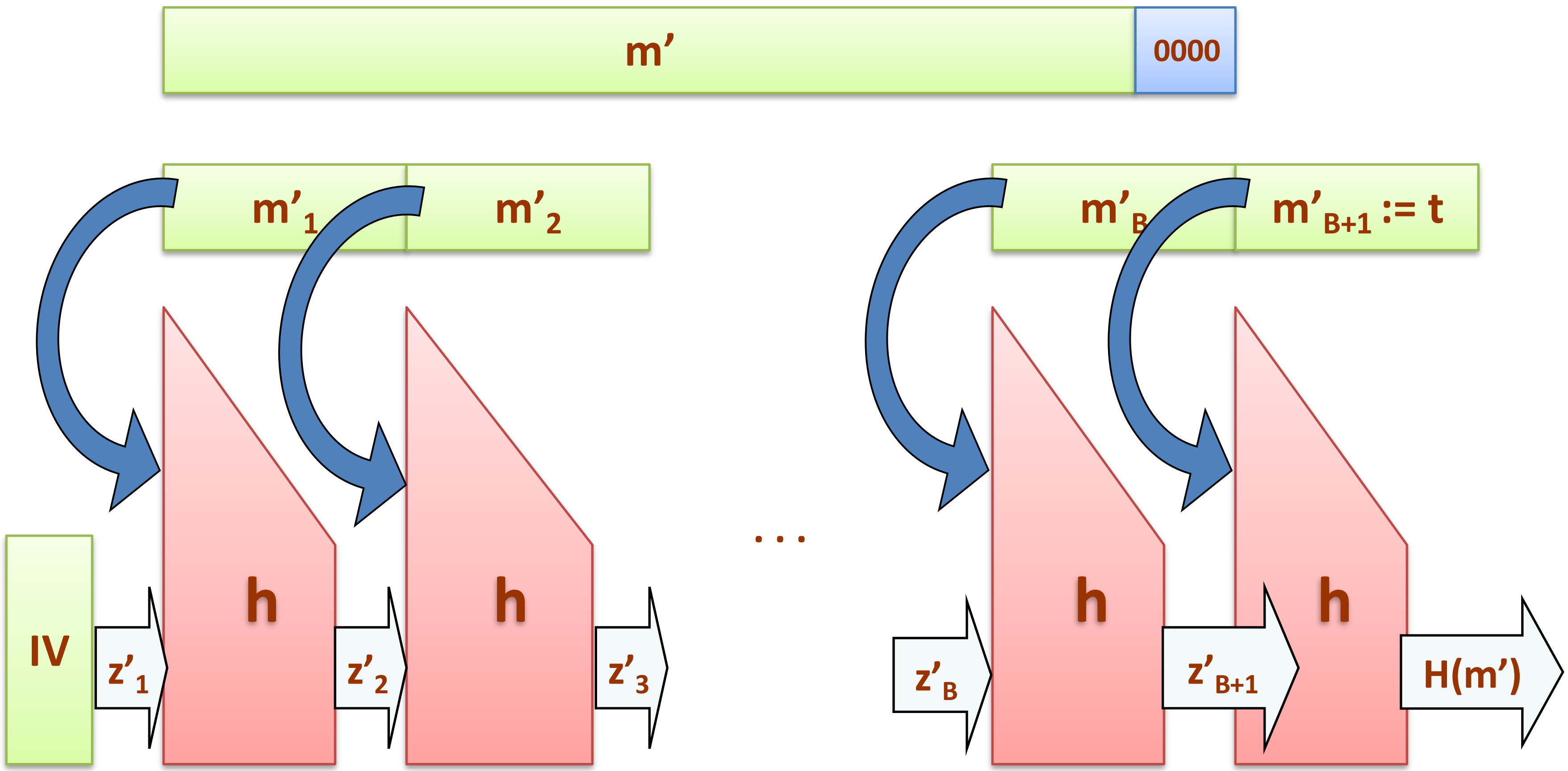
|m| = |m'|

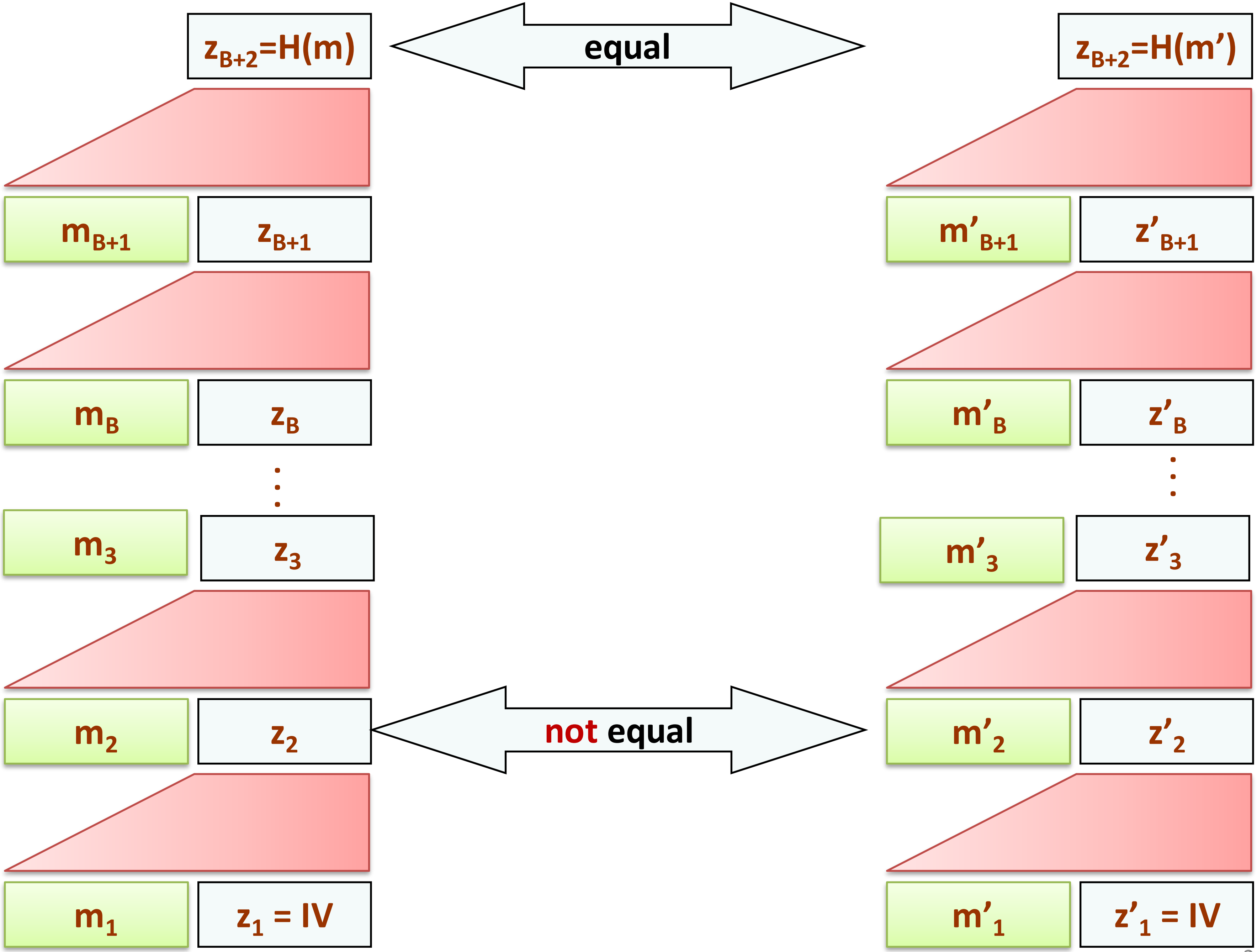
Some notation:



|m| = |m'|

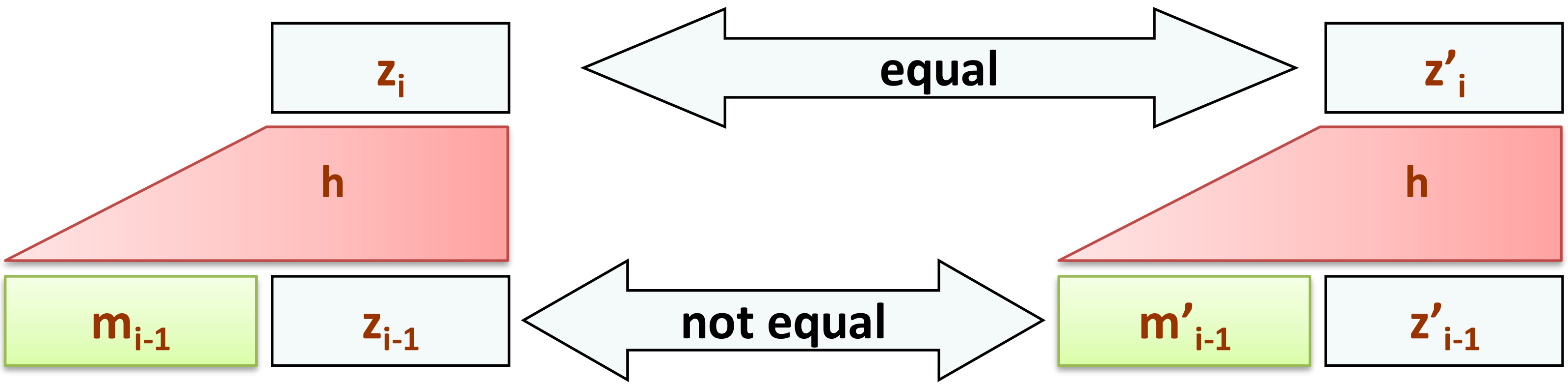
For **m'**:




z_{B+2}=H(m)
equal
z_{B+2}=H(m')
m_{B+1}
z_{B+1}
m'_{B+1}
z'_{B+1}
m_B
z_B
m'_B
z'_B
m_3
z_3
m'_3
z'_3
m_2
z_2
not equal
m'_2
z'_2
m_1
z_1 = IV
m'_1
z'_1 = IV

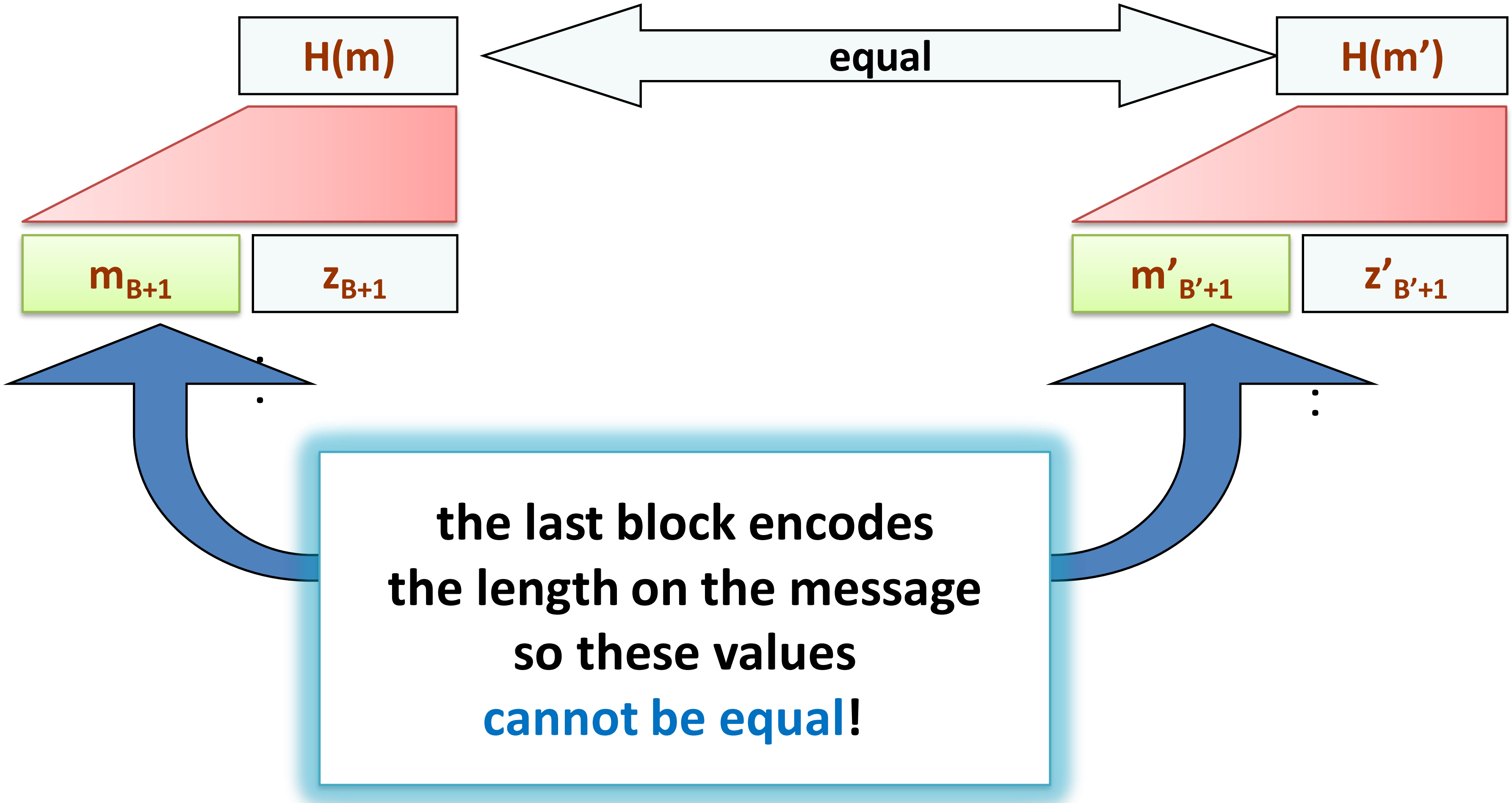
So, we have found a collision!

$$B_i = m_i || z_i$$



Let i be the largest index for which:
$z_i = z_i' \; and \; m_i || z_{i-1} \neq m'_{i-1} \, || z_{i-1}' \Rightarrow$
$h(m_{i-1} || z_{i-1}) = h(m'_{i-1} || z'_{i-1}) \Rightarrow$
There is a collision in h

Option 2: |m| ≠ |m'|

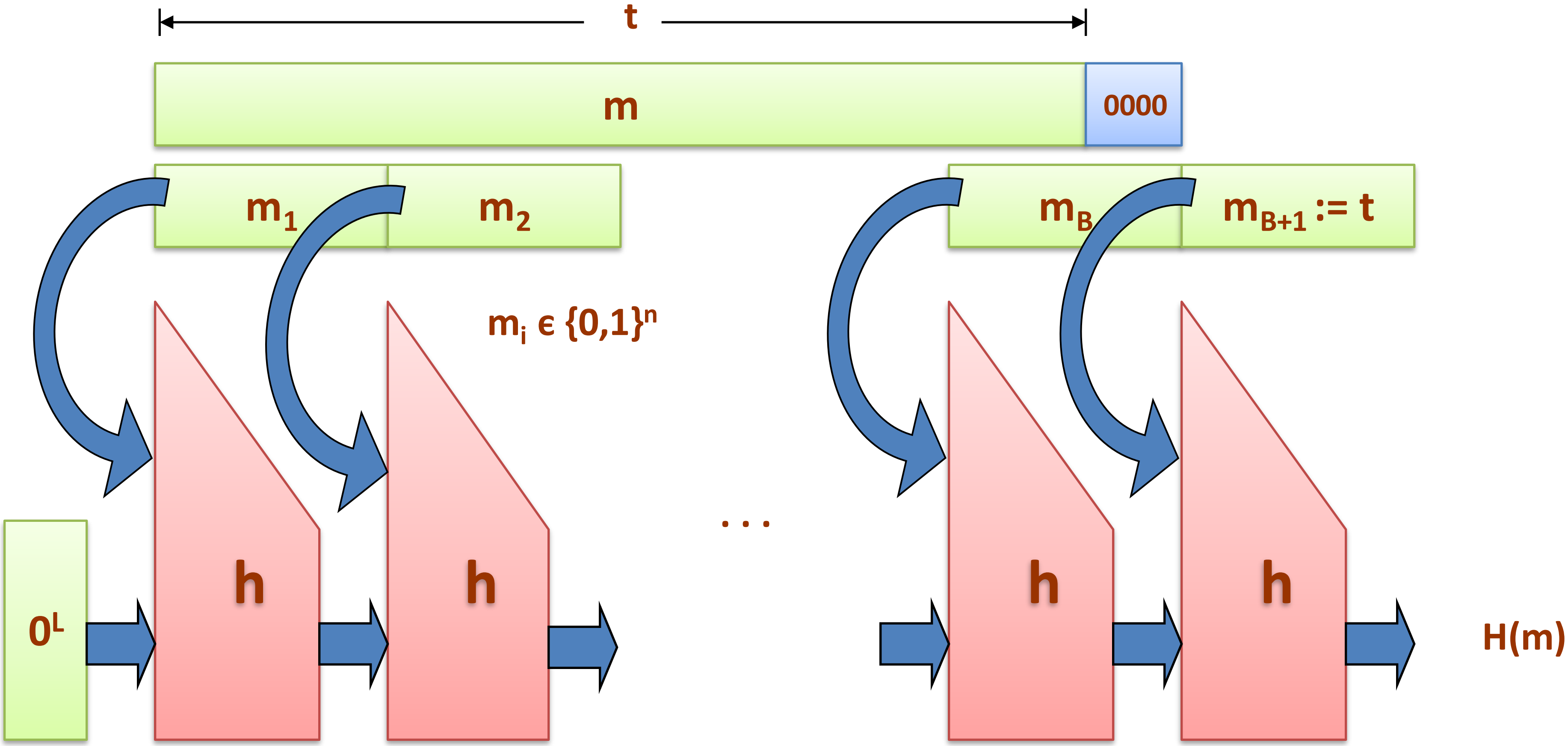


So, again we have found a collision!

Merkle-Damgård transform

given **h : $\{0,1\}^{2n} \rightarrow \{0,1\}^n$**

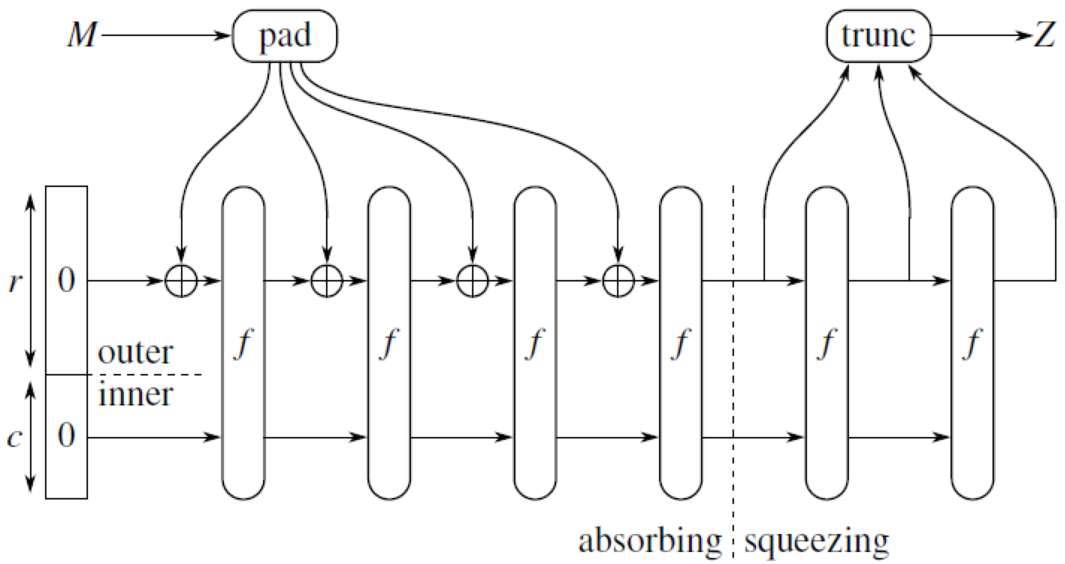
we construct **H : $\{0,1\}^* \rightarrow \{0,1\}^n$**



Need to design compression function h

SHA-3

The sponge construction



- Generalizes hash function: *extendable output function* (XOF)
- Calls a b-bit permutation f, with $b = r + c$
  - r bits of *rate*
  - c bits of *capacity* (security parameter)

Permutation

KECCAK[r, c]

- Sponge function using the permutation KECCAK-f
  - 7 permutations: $b \in \{25, 50, 100, 200, 400, 800, 1600\}$
    ... from toy over lightweight to high-speed ...
- SHA-3 instance: $r = 1088$ and $c = 512$
  - permutation width: 1600
  - security strength 256: post-quantum sufficient
- Lightweight instance: $r = 40$ and $c = 160$
  - permutation width: 200
  - security strength 80: same as (initially expected from) SHA-1

Birthday attacks on hash functions

Birthday paradox

- If we choose q elements $y_1, \ldots y_q$ at random from {1,…,N}, what is the probability that there exists i and j such that $y_i = y_j$?

365 possible days

What is the probability that two people have the same birthday?

Upper bound

- If we choose $y_1, \dots y_q$ uniformly at random from {1,…,N}, the probability of collision is upper bounded by:

$$\text{Coll}(q, N) \leq \frac{q(q-1)}{2N}$$

- Proof: (Union bound)

$$\Pr[\text{Coll}(q, N)] = \Pr\left[\exists\, i, j \; st \; y_i = y_j\right]$$

$$\leq \sum_{i,j} \Pr\left[y_i = y_j\right] = \binom{q}{2} \frac{1}{N} = \frac{q(q-1)}{2N}$$

Lower bound

- If we choose $y_1, \dots y_q$ uniformly at random from {1,…,N} and $q \leq \sqrt{2N}$, the probability of collision is lower bounded by:

$$\frac{q(q-1)}{4N} \leq \mathrm{Coll}(q, N) \leq \frac{q(q-1)}{2N}$$

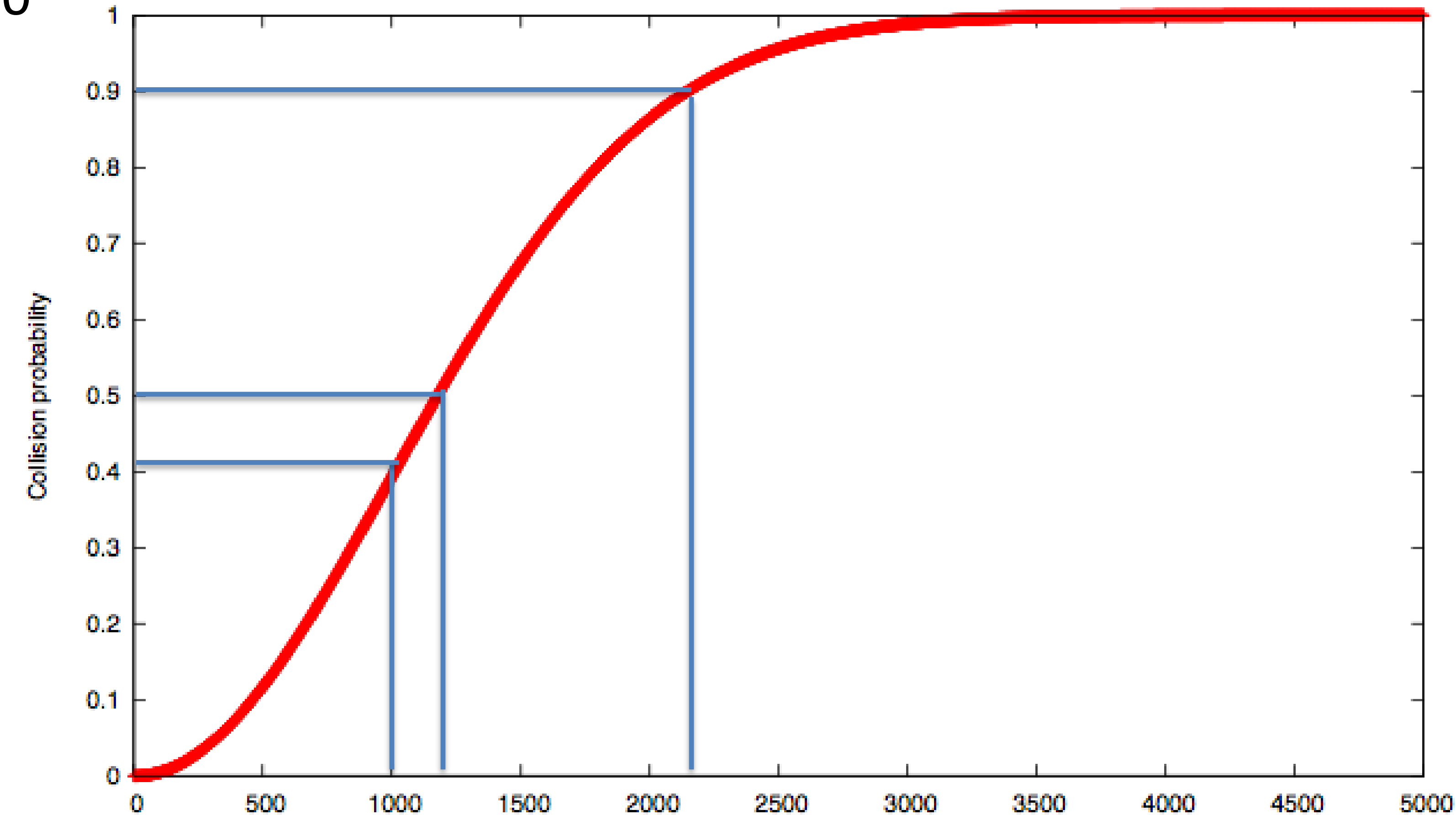
If $q = \Theta\left(\sqrt{N}\right)$, then $\mathrm{Coll}(q, N)$ is approx. ½

Birthday paradox: N = 365, q = 23

Hash functions: $N = 2^{\ell}, q = 2^{\ell/2}$

Collision probability

N=10^6



samples n

Generic attack on collision resistant hash functions

Let H: M → $\{0,1\}^\ell$ be a hash function (|M| >> 2^ℓ)

Generic alg. to find a collision **in time** $\mathbf{O(2^{\ell/2})}$ hashes

Algorithm:

1. Choose $2^{\ell/2}$ random messages in M: $m_1, \ldots, m_{2^\ell}$ (distinct w.h.p)
2. For i = 1, …, $2^{\ell/2}$ compute $t_i = H(m_i)$
3. Look for a collision ($t_i = t_j$)
4. If not found, got back to step 1

Running time: $\mathbf{O(2^{\ell/2})}$ (space $O(\mathbf{2^{\ell/2}})$)

Recap

- Collision-resistant hash functions are useful for many tasks
- Constructing hash functions using Merkle-Daamgard paradigm
  - Traditional designs: MD5, SHA-1, SHA-2
- SHA-3 is the new standard
  - Explicit collision found in MD5
  - Structural differences in SHA-1
- Birthday paradox implies n/2 level of security for n-bit hash function in best case

Acknowledgement

Some of the slides and slide contents are taken from
http://www.crypto.edu.pl/Dziembowski/teaching
and fall under the following:

We have also used slides from Prof. Dan Boneh online cryptography course at Stanford University:

http://crypto.stanford.edu/~dabo/courses/OnlineCrypto/