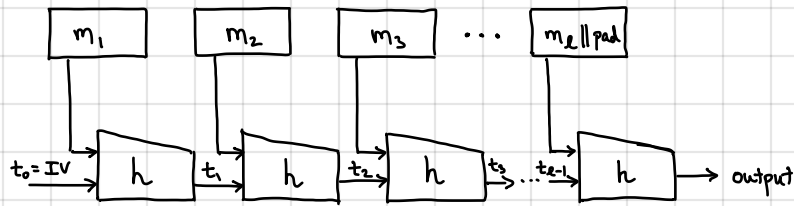


# Constructing CRHFs!

Many cryptographic hash functions (e.g., MD5, SHA-1, SHA-256) follow the Merkle-Damgård paradigm: start from hash function on short messages and use it to build a collision-resistant hash function on a long message:

1. Split message into blocks
2. Iteratively apply compression function (hash function on short inputs) to message blocks



$h$ : compression function

$t_0, \dots, t_l$ : chaining variables

padding introduced so last block is multiple of block size

↳ must also include an encoding of the message

length: typically of the form  $100 \dots 0 \parallel \langle s \rangle$

where  $\langle s \rangle$  is a fixed-length binary representation of message length in blocks

Recall:  $100 \dots 0$  padding was used in the ANSI standard

if not enough space to include the length, then extra block is added (similar to CBC encryption)

Hash functions are deterministic, so IV is a fixed string (defined in the specification) — can be taken to be all-zeroes string, but usually set to a custom value in constructions

for SHA-256:  
 $X = \{0, 1\}^{256} = Y$

Theorem. Suppose  $h: X \times Y \rightarrow X$  be a compression function. Let  $H: Y^{\leq l} \rightarrow X$  be the Merkle-Damgård hash function constructed from  $h$ . Then, if  $h$  is collision-resistant,  $H$  is also collision-resistant.

Proof. Suppose we have a collision-finding algorithm  $A$  for  $H$ . We use  $A$  to build a collision-finding algorithm for  $h$ :

1. Run  $A$  to obtain a collision  $M$  and  $M'$  ( $H(M) = H(M')$  and  $M \neq M'$ ).
2. Let  $M = m_1, m_2, \dots, m_u$  and  $M' = m'_1, m'_2, \dots, m'_v$  be the blocks of  $M$  and  $M'$ , respectively. Let  $t_0, t_1, \dots, t_u$  and  $t'_0, t'_1, \dots, t'_v$  be the corresponding chaining variables.
3. Since  $H(M) = H(M')$ , it must be the case that

$$H(M) = h(t_{u-1}, m_u) = h(t'_{v-1}, m'_v) = H(M')$$

If either  $t_{u-1} \neq t'_{v-1}$  or  $m_u \neq m'_v$ , then we have a collision for  $h$ .

Otherwise,  $m_u = m'_v$  and  $t_{u-1} = t'_{v-1}$ . Since  $m_u$  and  $m'_v$  include an encoding of the length of  $M$  and  $M'$ , it must be the case that  $u = v$ . Now, consider the second-to-last block in the construction (with output  $t_{u-1} = t'_{u-1}$ ):

$$t_{u-1} = h(t_{u-2}, m_{u-1}) = h(t'_{u-2}, m'_{u-1}) = t'_{u-1}$$

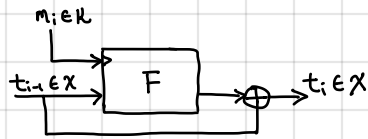
Either we have a collision or  $t_{u-2} = t'_{u-2}$  and  $m_{u-1} = m'_{u-1}$ . Repeat down the chain until we have collision or we have concluded that  $m_i = m'_i$  for all  $i$ , and so  $M = M'$ , which is a contradiction.

Note: Above construction is sequential. Easy to adapt construction (using a tree) to obtain a parallelizable construction.

Sufficient now to construct a compression function.

Typical approach is to use a block cipher.

Davies-Meyer: Let  $F: \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}$  be a block cipher. The Davies-Meyer compression function  $h: \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{X}$  is then



$$h(k, x) := F(k, x) \oplus x$$

Many other variants also possible:  $h(k, x) = F(k, x) \oplus k \oplus x$   
[used in Whirlpool hash family]

Need to be careful with design!

- $h(k, x) = F(k, x)$  is not collision-resistant:  $h(k, x) = h(k', F^{-1}(k', F(k, x)))$
- $h(k, x) = F(k, x) \oplus k$  is not collision-resistant:  $h(k, x) = h(k', F^{-1}(k', F(k, x) \oplus k \oplus k'))$

Theorem. If we model  $F$  as an ideal block cipher (ie, a truly random permutation for every choice of key), then Davies-Meyer is collision-resistant.

Conclusion: Block cipher + Davies-Meyer + Merkle-Damgård  $\Rightarrow$  CRHFs

Examples: SHA-1: SHACAL-1 block cipher with Davies-Meyer + Merkle-Damgård

SHA-256: SHACAL-2 block cipher with Davies-Meyer + Merkle-Damgård

→ birthday attack run-time:  $\sim 2^{80}$   
attack ran in time  $\sim 2^{64}$  (100,000x faster)  
January, 2020: chosen-prefix collision in  $\sim 2^{63.4}$  time!  
← no longer secure [first collision found in 2017!]  
- SHA-1 extensively used (eg, git, svn, software updates, PGP/GPG signatures, certificates) → attacks show need to transition to SHA-2 or SHA-3

Recently: SHA-3 family of hash functions standardized (2015)

↳ Relies on different underlying structure ("sponge" function)

↳ Both SHA-2 and SHA-3 are believed to be secure (most systems use SHA-2 - typically much faster)

Why not use AES?

- Block size too small! AES outputs are 128-bits, not 256 bits (so birthday attack finds collision in  $2^{64}$  time)

- Short keys means small number of message bits processed per iteration.

- Typically, block cipher designed to be fast when using same key to encrypt many messages

↳ In Merkle-Damgård, different keys are used, so alternate design preferred (AES key schedule is expensive)