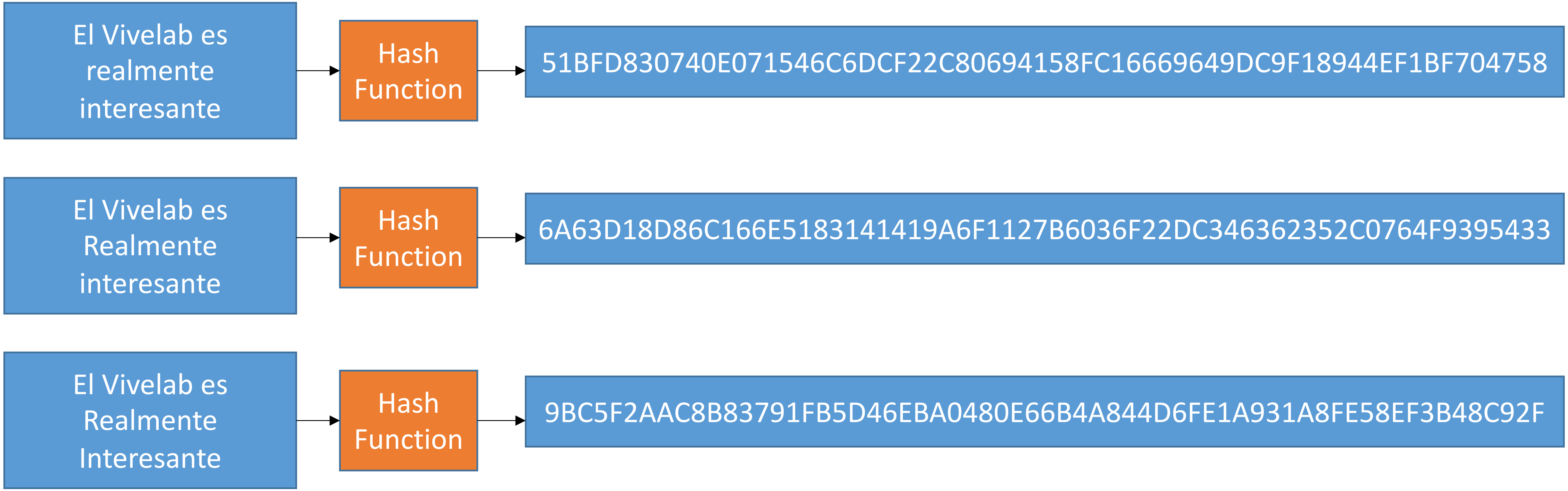
# Introducción a la Criptografía y a la Seguridad de la Información

Part 7a
Cryptographic Hash
Functions

Jorge E. Camargo, PhD

# What is a hash function?



El Vivelab es realmente interesante

Hash Function

51BFD830740E071546C6DCF22C80694158FC16669649DC9F18944EF1BF704758

El Vivelab es Realmente interesante

Hash Function

6A63D18D86C166E5183141419A6F1127B6036F22DC346362352C0764F9395433

El Vivelab es Realmente Interesante

Hash Function

9BC5F2AAC8B83791FB5D46EBA0480E66B4A844D6FE1A931A8FE58EF3B48C92F

# A formal definition of hash function

- Deterministic algorithm
- No encryption
- Not signature

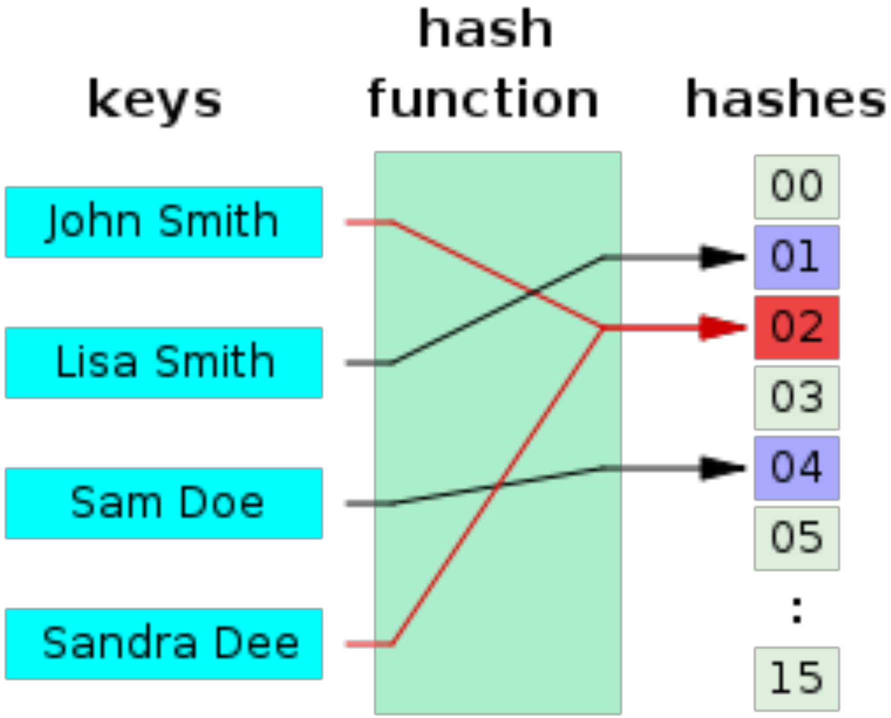
$$H : \left\{ \begin{array}{rcl} \text{any bit string} & \to & \text{m bits} \\ x & \mapsto & y = H(x) \end{array} \right.$$

# Properties of a (cryptographic) hash function

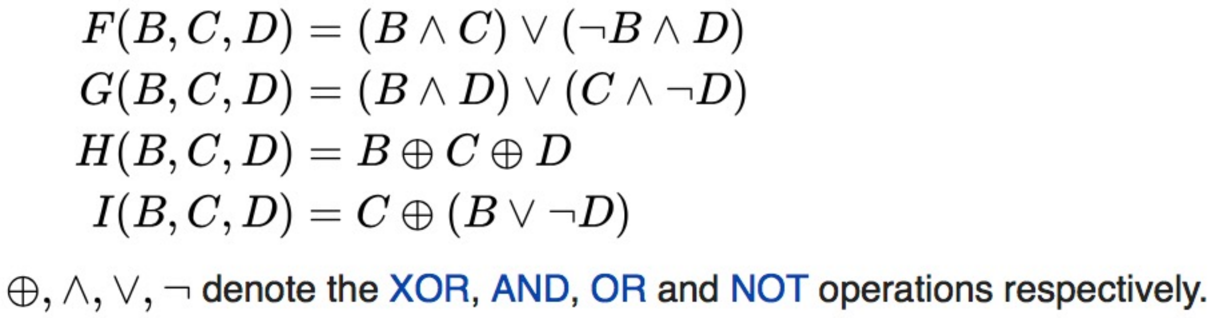
- Deterministic
- Uniformity
- Defined range
- Non-invertibility
- Pre-image resistance: Given a hash value *h* it should be difficult to find any message *m* such that *h = hash(m)*
- Second pre-image resistance: Given an input *m1*, it should be difficult to find a different input *m2* such that *hash(m1) = hash(m2)*
- Collision resistance: It should be difficult to find two different messages *m1* and *m2* such that *hash(m1) = hash(m2)*

# Collisions

- A hash function that maps names to integers form 0 to 15.
- There is a collision between “John Smith” and “Sandra Dee”



# MD5 hash function



$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$
$$G(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$$
$$H(B,C,D) = B \oplus C \oplus D$$
$$I(B,C,D) = C \oplus (B \vee \neg D)$$

$\oplus, \wedge, \vee, \neg$ denote the XOR, AND, OR and NOT operations respectively.

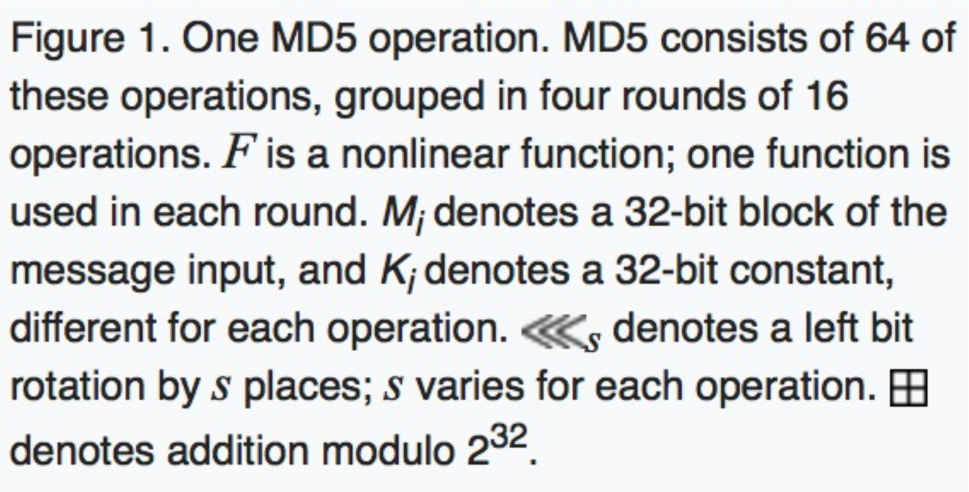
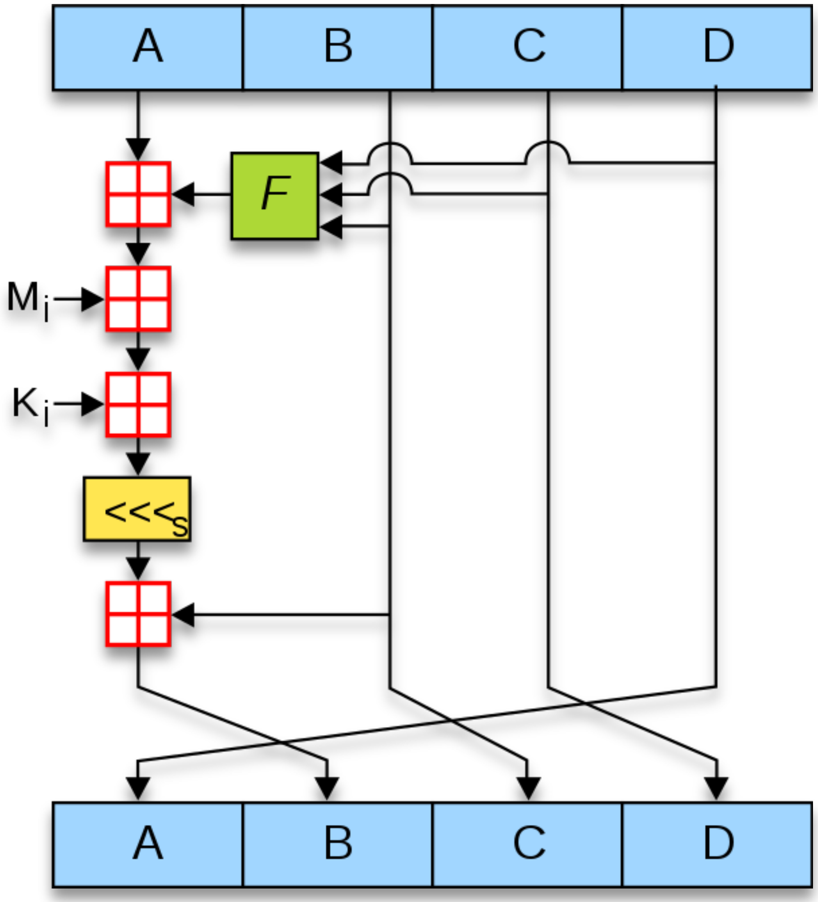
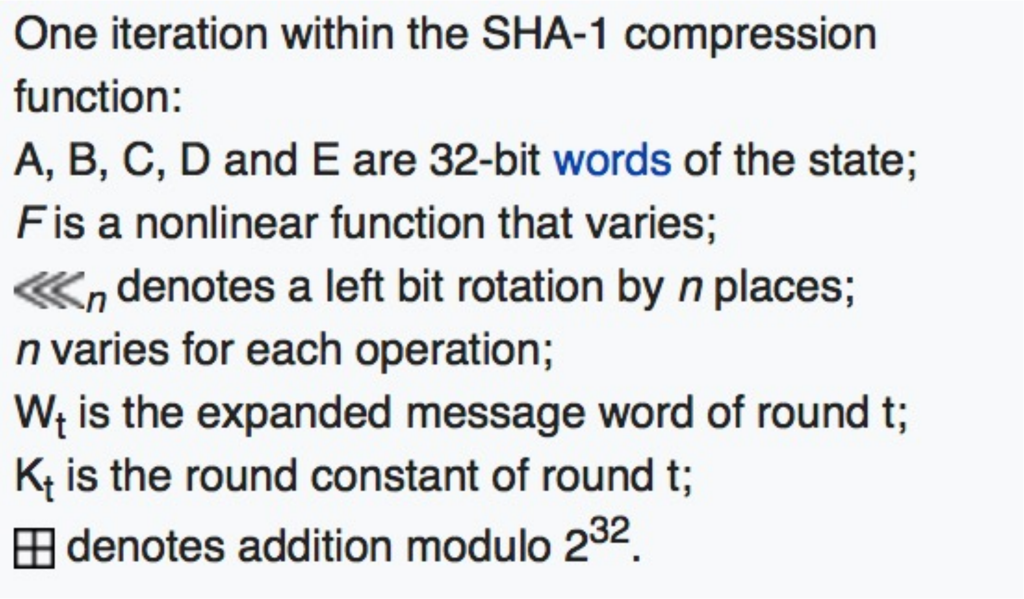


Figure 1. One MD5 operation. MD5 consists of 64 of these operations, grouped in four rounds of 16 operations. $F$ is a nonlinear function; one function is used in each round. $M_i$ denotes a 32-bit block of the message input, and $K_i$ denotes a 32-bit constant, different for each operation. $\lll_s$ denotes a left bit rotation by $s$ places; $s$ varies for each operation. $\boxplus$ denotes addition modulo $2^{32}$.



**Fuente**: https://en.wikipedia.org/wiki/MD5

# SHA-1 hash function



One iteration within the SHA-1 compression function:

A, B, C, D and E are 32-bit words of the state;

*F* is a nonlinear function that varies;

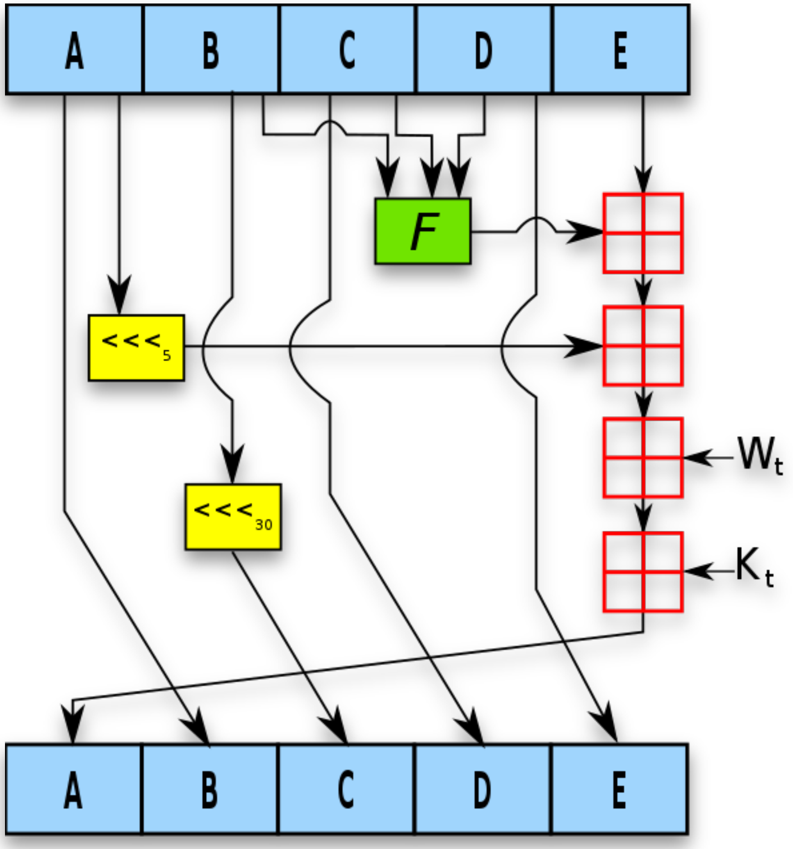
$\lll_n$ denotes a left bit rotation by *n* places;

*n* varies for each operation;

$W_t$ is the expanded message word of round t;

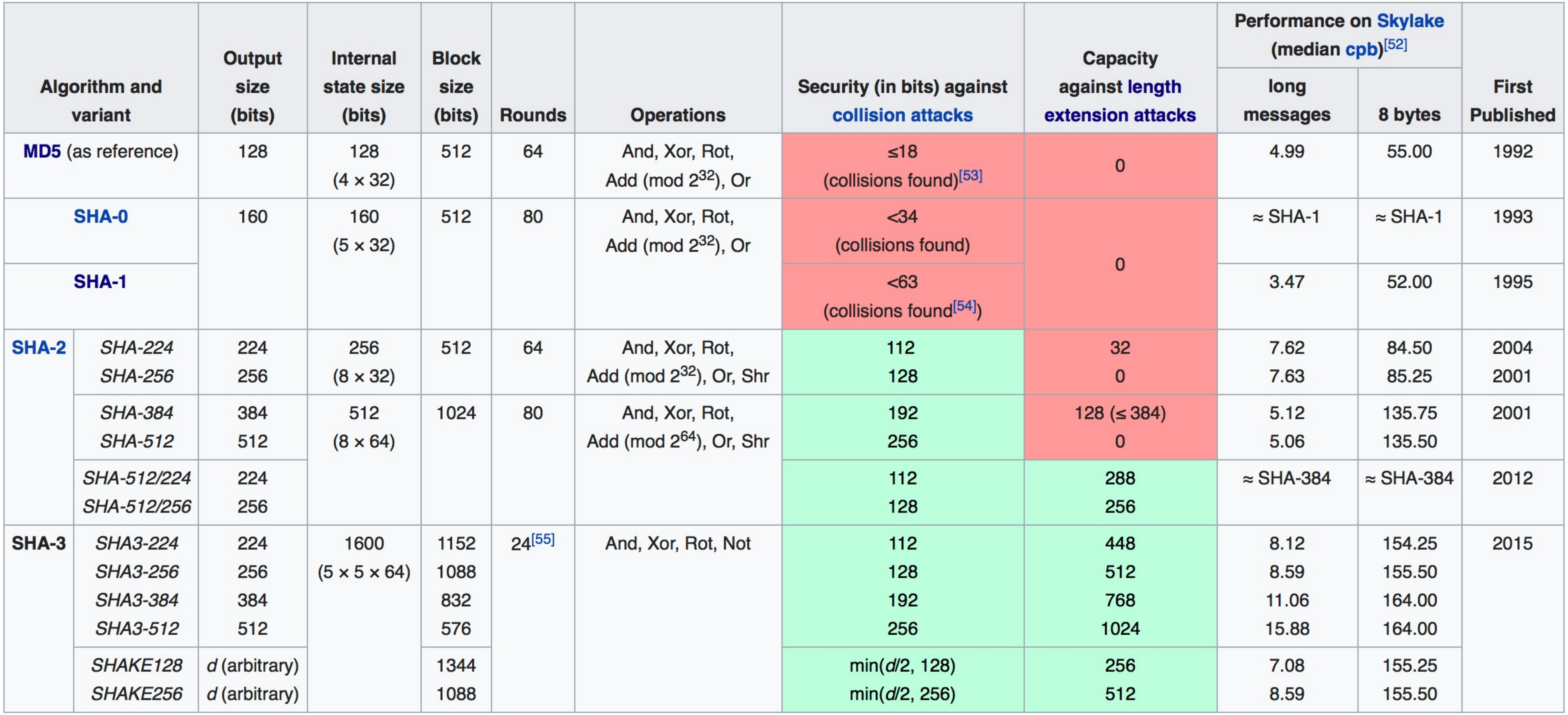
$K_t$ is the round constant of round t;

⊞ denotes addition modulo $2^{32}$.



**Fuente**: https://en.wikipedia.org/wiki/SHA-1

# Comparison



| Algorithm and variant | | Output size (bits) | Internal state size (bits) | Block size (bits) | Rounds | Operations | Security (in bits) against collision attacks | Capacity against length extension attacks | Performance on Skylake (median cpb)[52] | | First Published |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | long messages | 8 bytes | |
| **MD5** (as reference) | | 128 | 128<br>(4 × 32) | 512 | 64 | And, Xor, Rot, Add (mod $2^{32}$), Or | ≤18<br>(collisions found)[53] | 0 | 4.99 | 55.00 | 1992 |
| **SHA-0** | | 160 | 160<br>(5 × 32) | 512 | 80 | And, Xor, Rot, Add (mod $2^{32}$), Or | <34<br>(collisions found) | 0 | ≈ SHA-1 | ≈ SHA-1 | 1993 |
| **SHA-1** | | | | | | | <63<br>(collisions found[54]) | | 3.47 | 52.00 | 1995 |
| **SHA-2** | *SHA-224*<br>*SHA-256* | 224<br>256 | 256<br>(8 × 32) | 512 | 64 | And, Xor, Rot, Add (mod $2^{32}$), Or, Shr | 112<br>128 | 32<br>0 | 7.62<br>7.63 | 84.50<br>85.25 | 2004<br>2001 |
| | *SHA-384*<br>*SHA-512* | 384<br>512 | 512<br>(8 × 64) | 1024 | 80 | And, Xor, Rot, Add (mod $2^{64}$), Or, Shr | 192<br>256 | 128 (≤ 384)<br>0 | 5.12<br>5.06 | 135.75<br>135.50 | 2001 |
| | *SHA-512/224*<br>*SHA-512/256* | 224<br>256 | | | | | 112<br>128 | 288<br>256 | ≈ SHA-384 | ≈ SHA-384 | 2012 |
| **SHA-3** | *SHA3-224*<br>*SHA3-256*<br>*SHA3-384*<br>*SHA3-512* | 224<br>256<br>384<br>512 | 1600<br>(5 × 5 × 64) | 1152<br>1088<br>832<br>576 | 24[55] | And, Xor, Rot, Not | 112<br>128<br>192<br>256 | 448<br>512<br>768<br>1024 | 8.12<br>8.59<br>11.06<br>15.88 | 154.25<br>155.50<br>164.00<br>164.00 | 2015 |
| | *SHAKE128*<br>*SHAKE256* | *d* (arbitrary)<br>*d* (arbitrary) | | 1344<br>1088 | | | min(*d*/2, 128)<br>min(*d*/2, 256) | 256<br>512 | 7.08<br>8.59 | 155.25<br>155.50 | |

**Fuente**: https://en.wikipedia.org/wiki/SHA-3

# Hash online

- Test some hash functions in:

http://www.convertstring.com/Hash

# Use of hash functions in Blockchain

- https://colab.research.google.com/drive/171hbYK3ERQzXW1yNLQKMSLoNwL1fzgdb#scrollTo=_wu-qTT0QGz2

# References

- Pinzon, Yoan. “Introducción a la criptografía y a la seguridad de la información”, 2013.
- Menezes A., Handbook of applied Cryptografy, 5th Edition. CRC Press, 2001.
- A Graduate Course in Applied Cryptography by D. Boneh and V. Shoup
- H. Delfs and H. Knebl, Introduction to Cryptography, 3rd ed. 2015.
- J. A. Buchmann, Introduction to Cryptography. 2004.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Ch. 31 in Introduction to Algorithms, 3rd ed. 2009.
- B. Lomas de Zamora: Gradi. Hacking desde cero, Fox Andina, 2011.
- Secure Coding Working Group, Japan Smartphone Security Association (JSSEC), Android Application Secure Design/Secure Coding Guidebook, 2017.