

# Hash functions

Gianluca Dini  
Dept. of Ingegneria dell'Informazione  
University of Pisa  
Email: [gianluca.dini@unipi.it](mailto:gianluca.dini@unipi.it)  
Version: 2023-03-30

1

## An example

The input size is finite but arbitrary

Nel mezzo del cammin di nostra vita  
mi ritrovai per una selva oscura  
che' la diritta via era smarrita.

Ahi quanto a dir qual era e` cosa dura  
esta selva selvaggia e aspra e forte  
che nel pensier rinova la paura!


↓

MD5

↓

0xd94f329333386d5abef6475313755e94

128 bit      The hash size is fixed, generally smaller than the message size




UNIVERSITÀ DI PISA

apr. '23

Hash functions

2

2



UNIVERSITÀ DI PISA

## Informal properties


- Applicable to messages of any size
- Output of fixed length (digest, hash value, fingerprint)
- No key (!)
- “Easy” to compute
- “Difficult” to invert
- “Unique” (the hash of a message can be used to "uniquely" represent the message) →
  - The output should be highly sensitive to all inputs →
  - if we make minor modifications to the input, the output should look like very different

apr. '23

Hash functions

3

3



UNIVERSITÀ DI PISA

## Informal properties

- The fingerprint must be *highly* sensitive to *all* input bits
  - Input «I am not a crook»
    - Hash (MD5): 6d17fcd4ae0e82fa4409f4ea6f4106a6
  - Input «I am not a cook»
    - Hash (MD5): 9ebe3d42d5c01fc59fe3daacbf42f515
- <https://www.fileformat.info/tool/hash.htm>

apr. '23

Hash functions

4

4

## Example: protecting files

- Software packages

package name  
 $F_1$

package name  
 $F_2$

...

package name  
 $F_n$

read-only  
public space

$H(F_1)$   
 $H(F_2)$   
 $H(F_n)$

- When user downloads package, can verify that contents are valid
  - H collision resistant  $\Rightarrow$   
attacker cannot modify package without detection
- No key needed (public verifiability), but requires read-only space


apr. '23

Hash functions

5

5

## Example: protecting files



The screenshot shows a web page titled 'Prelievo da WinRAR.it'. It contains instructions for downloading a file and verifying its integrity. The file name is 'WinRAR-x64-600b1it.exe' and its size is '3.442 K'. A red box highlights the checksums: MD5: c11ac9a41e5d178e65417faa6dccf75f, SHA-1: c9a2e9ca312573aaaa7b0c16fd49cb3ce40bf54f, and SHA-256: 07a60c7da09679960aa2e9e7335194506cff71caebf0be62b97069d8619221f6.

apr. '23


Hash functions

6

6

Foundations of Cybersecurity

3



UNIVERSITÀ DI PISA

# Properties: collisions


- A hash function  $H: \{0,1\}^* \rightarrow \{0,1\}^n$
- Properties
  - *Compression*:  $H$  maps an input  $x$  of arbitrary finite length into an output  $H(x)$  of fixed length  $n$
  - *Ease to compute*: given  $x$ ,  $H(x)$  must be “easy” to compute
  - *Many-to-one*: a hash function is many-to-one and thus implies collisions (pigeonhole principle)
- (Def) A collision for  $H$  is a pair  $x_0, x_1$  s.t.  $H(x_0) = H(x_1)$  and  $x_0 \neq x_1$

apr. '23

Hash functions

7

7



UNIVERSITÀ DI PISA

# Security properties

- Preimage resistance (one-wayness)
  - For essentially all pre-specified outputs, it is *computationally infeasible* to find any input which hashes to that output
    - i.e., given an output  $y$ , to find  $x$  such that  $y = h(x)$  for which  $x$  is not known
- 2nd-preimage resistance (weak collision resistance)
  - it is computationally infeasible to find any second input which has the same output as any specified input
    - i.e., given  $x$ , to find  $x' \neq x$  such that  $h(x) = h(x')$
- Collision resistance (strong collision resistance)
  - it is computationally infeasible to find any two distinct inputs which hash to the same output,
    - i.e., find  $x, x'$  such that  $h(x) = h(x')$


apr. '23

Hash functions

8

8

# Classification



- One-way hash function (OWHF)
  - Provides preimage resistance, 2-nd preimage resistance
  - OWHF is also called weak one-way hash function
- Collision resistant hash function (CRHF)
  - Provides 2-nd preimage resistance, collision resistance
  - CRHF is also called strong one-way hash function

apr. '23

Hash functions

9

# Relationship between security properties




- FACT 1 - Collision resistance implies 2nd preimage resistance
- FACT 2 - Collision resistance does not imply preimage resistance
  - However, in practice, CRHF almost always has the additional property of preimage resistance

apr. '23

Hash functions

10

# Attacking Hash Functions



UNIVERSITÀ DI PISA

- An attack is successful if it produces a collision (forgery)
- Types of forgery
  - Selective forgery: the adversary has complete, or partial, control over  $x$
  - Existential forgery: the adversary has no control over  $x$


apr. '23

Hash functions

11

11

# Black box attacks



UNIVERSITÀ DI PISA

- Consider  $H$  as a black box
- Only consider the output bit length  $n$
- Assume  $H$  approximates a random variable
  - Each output is equally likely for a random input (so weak collisions exist for all output values)


apr. '23

Hash functions

12

12

## Specific Black box Attacks



UNIVERSITÀ DI PISA

↓

- Guessing attack
  - find a 2<sup>nd</sup> pre-image
  - Running time:  $O(2^n)$  hash ops
- Birthday attack:
  - find a collision
  - Running time:  $O(2^{n/2})$  hash ops
- These attacks constitute a security upper bound
  - More efficient analytical attacks may exist (e.g., against MD5, SHA-1)


apr. '23

Hash functions

13

13

## Guessing attack



UNIVERSITÀ DI PISA

→

- Objective: to find a 2<sup>nd</sup> pre-image
  - Given  $x_0$ , find  $x_1 \neq x_0$  s.t.  $H(x_0) = H(x_1)$
- The attack

```
int GuessingAttack(x0) {  
    repeat  
        x1 ← random(); // guessing  
    until h(x0) == h(x1)  
    return x1;  
}
```


apr. '23

Hash functions

14

14

# Guessing attack



UNIVERSITÀ DI PISA

- Running time
  - Every step requires
    - 1 random number generation: efficient!
    - 1 hash function computation: efficient!
  - Constant and negligible data/storage complexity
  - Running time in the order of  $2^n$  operations


apr. '23

Hash functions

15

15

# Birthday attack



UNIVERSITÀ DI PISA

- Start with
  - $x_1$  = «Transfer \$10 into Oscar's account»
  - $x_2$  = «Transfer \$10.000 into Oscar's account»
- The attack
  - Do
    - Alter  $x_1$  and  $x_2$  at non-visible locations so that semantics is unchanged (e.g., insert spaces, tabs, return,...)
  - Until  $H(x_1) == H(x_2)$

apr. '23


Hash functions

16

16



# Birthday attack



UNIVERSITÀ DI PISA

- The birthday attack algorithm
  1. Choose  $N = 2^{n/2}$  random input messages  $x_1, x_2, \dots, x_N$  (distinct w.h.p.)
  2. For  $i := 1$  to  $N$  compute  $t_i = H(x_i)$
  3. Look for a collision ( $t_i = t_j$ ),  $i \neq j$ . If not found, go to step 1.
- Attack complexity
  - Running Time:  $2^{n/2}$
  - Space:  $2^{n/2}$
  - Probability of collision is 50%


apr. '23

Hash functions

17

17

# Birthday paradox: intuition



UNIVERSITÀ DI PISA

- Problem #1.
  - In a room of  $t = 23$  people, what is the probability that at least a person is born on 25 December?
    - Answer:  $23/365 = 0.063$
- Problem #2.
  - In a room of  $t = 23$  people, what is the probability that at least 2 people have the same birthdate?
    - Answer: 0.507


apr. '23

Hash functions

18

18

# Birthday attack



UNIVERSITÀ DI PISA


- Apply the birthday paradox to hash function
  - We have:  $2^n$  elements and  $t$  inputs  $(x_1, x_2, \dots, x_t)$
  - $\pi = \text{Pr}[\text{no collision}] = \left(1 - \frac{1}{2^n}\right) \left(1 - \frac{2}{2^n}\right) \dots \left(1 - \frac{t-1}{2^n}\right) = \prod_{i=1}^{t-1} \left(1 - \frac{i}{2^n}\right) \approx \prod_{i=1}^{t-1} e^{-\frac{i}{2^n}} = e^{-\frac{1+2+\dots+t-1}{2^n}} \approx e^{-\frac{t(t-1)}{2^{n+1}}} \cong e^{-\frac{t^2}{2^{n+1}}}$
  - Probability of collision  $\lambda = 1 - \pi$
  - Solve in  $t$ ,  $\Rightarrow t \approx 2^{(n+1)/2} \sqrt{\ln\left(\frac{1}{1-\lambda}\right)}$
  - For  $\lambda = 0.5$ ,  $t \approx 1.2 \times 2^{n/2}$

apr. '23

Hash functions

19

# Birthday attack



UNIVERSITÀ DI PISA

- In practice,
  - The # of messages we need to hash to find a collision is in the order of the square root of the # of possible output values, i.e.,  $\sqrt{2^n} = 2^{n/2}$
- For example
  - $n = 80$  bit,  $\lambda = 0.5 \Rightarrow t \approx 2^{40.2}$  (doable with current laptops)
  - The probability of collision  $\lambda$  does not influence the attack complexity very much
- Rule of thumb:  $\text{sizeof}(\text{digest}) = 2 \times \text{sizeof}(\text{key})$ 
  - $K$ : block cipher key

apr. '23

Hash functions

20

Hash functions

HOW TO BUILD HASH FUNCTIONS

apr. '23


Hash functions

21

21

Types of hash functions

- Dedicated hash functions
- Block cipher-based hash functions



UNIVERSITÀ DI PISA


apr. '23

Hash functions

22

22

# How to build a hash function



UNIVERSITÀ DI PISA

- Approach
  - Given a CRHF for short messages, construct a CRHF for long messages
- Solution:
  - The Merkle-Damgard iterated construction
  - Most of hash functions follow the Merkle-Damgard construction including SHA.


apr. '23

Hash functions

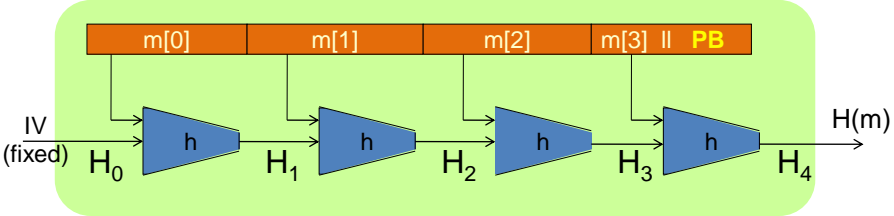
23

23

# The Merkle-Damgard iterated construction



UNIVERSITÀ DI PISA



- **Compression function  $h$ :**  $T \times X \rightarrow T$ 
  - $H_i$  - chaining variables
- **Padding block PB:** 1000... || msg len
  - msg len (on 64 bits) complicates adversary's task
  - If no space for PB add another block

apr. '23


Hash functions

24

24

## Merkle-Damgard collision resistance

- THEOREM. if compression function  $h$  is collision resistant (and message length is part of the input) then so is  $H$ .
  - Proof (by contradiction)
    - Collision on  $H \rightarrow$  collision on  $h$ . Q.E.D.
- Comment
  - To construct a CRHF, it *suffices* to construct a collision resistant compression function



UNIVERSITÀ DI PISA

apr. '23


Hash functions

25

25

## Hash functions from block ciphers

- Use block cipher chaining techniques
  - Matyas-Meyer-Oseas
  - Davies-Meyer
  - Miyaguchi-Preneel
  - Use block ciphers with 192/256 bit blocks
    - E.g. AES
- Cons
  - (digest size = block size) may be not enough for collision resistance
  - Possible solutions
    - Use block cipher with larger blocks (AES-192, AES-256)
    - Hirose scheme: use several instances of the block cipher



UNIVERSITÀ DI PISA

apr. '23

Hash functions

26

26


# Davies-Meyer

- Finding a collision  $h(H, m) = h(H', m')$  requires  $2^{m/2}$  evaluations of  $(E, D) \Rightarrow$  best possible!

apr. '23

Hash functions

27



UNIVERSITÀ DI PISA

27

# Exercise


- Problem
  - If we remove the xor, the compression function is not collision resistant anymore.
  - Proof (by contradiction)
    - Remove the xor  $\Rightarrow h(H, m) = E(m, H)$
    - To construct a collision  $(H, m)$  and  $(H', m')$  is easy
      - Choose a random triple  $(H, m, m')$
      - Determine  $H'$  such that  $E(m, H) = E(m', H') \Rightarrow H' = D(m', E(m, H))$

Q.E.D.

apr. '23

Hash functions


28



UNIVERSITÀ DI PISA

28

# The MD4 family



Algorithm		Output [bit]	Input [bit]	No. of rounds	Collisions found
MD5		128	512	64	yes
SHA-1		160	512	80	yes
SHA-2	SHA-224	224	512	64	no
	SHA-256	256	512	64	no
	SHA-384	384	1024	80	no
	SHA-512	512	1024	80	no


apr. '23

Hash functions

29

29

# MD5



- Developed in 1991
- 128-bit outuput lenght
- Collisions found in 2004, should be non longer used
  - Collision attack:  $O(2^{24.1})$
  - Chosen-prefix collision attack:  $O(2^{39})$

apr. '23


Hash functions

31

31

# SHA-1

- Designed by NSA and standardised by NIST in 1995
- 160 bits output length
- Collision on SHA-1 in 2017, now deprecated
  - CWI – Google team
  - Forged PDF documents
  - Running time
    - Over 9+ quintillion SHA1 computations that took 6,500 years of CPU computation and 100 years of GPU computations however  $10^5$  times faster than black box attack
    - <https://www.cwi.nl/news/2017/cwi-and-google-announce-first-collision-for-industry-security-standard-sha-1>



UNIVERSITÀ DI PISA

apr. '23


Hash functions

32

32

# Other hash functions

- SHA-2 (NIST 2002)
  - 256-bit, 384-bit or 512-bit output length
  - No known significant weaknesses but its structure is similar to SHA-1 and MD5
- SHA-3/Keccak
  - Result from public competition from 2008-2012
  - Very different design than SHA family
    - Requirement from NIST to defend from possible weakness in SHA family
  - Support 224, 256, 384, and 512-bit output length



UNIVERSITÀ DI PISA

apr. '23

Hash functions

33

33



Hash functions

USES OF HASH FUNCTIONS


apr. '23

Hash functions

34

34

Uses of hash functions



UNIVERSITÀ DI PISA

- Digital signatures
  - Requires strong collision resistance
- Password storage
  - Requires weak collision resistance
- Authentication of origin
  - Requires weak collision resistance
- Identification (one-time password)
  - Requires weak collision resistance and one-wayness

apr. '23

Hash functions

35

35

Hash Functions

AUTHENTICATION OF ORIGIN


apr. '23

Hash functions

36

36

Integrity vs authentication

  
UNIVERSITÀ DI PISA

- Message integrity
  - The property whereby data has not been altered in an unauthorized manner since the time it was created, transmitted, or stored by an authorized source
- Message origin authentication
  - A type of authentication whereby a party is corroborated as the (original) source of specified data created at some time in the past
- Data origin authentication => data integrity


apr. '23

Hash functions

37

37

# Use of hash functions for authentication



UNIVERSITÀ DI PISA

- The purpose of a hash functions, *in conjunction with other mechanisms* (authentic channel, encryption, digital signature), is to provide message integrity and authentication


apr. '23

Hash functions

38

38

# Authentic channel



UNIVERSITÀ DI PISA

→

- Alice
  - Let  $t = H(x)$
- MIM attack

$x, t$

$x, t$

MIM

$x', t'$

$t' = H(x')$

Bob


apr. '23

Hash functions

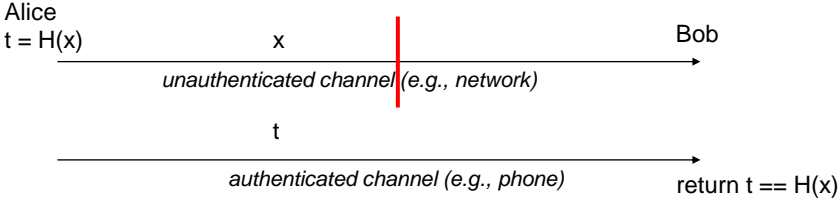
39

39

# Authentic channel




- Alice
  - Computes  $t = H(x)$
  - Sends  $x$  to Bob through the network
  - Reads  $t$  to Bob over the phone
    - An additional channel considered authenticated by assumption



apr. '23 Hash functions 40

40

# Hash functions with block ciphers



- $E_k(x || H(x))$  recommended
  - Confidentiality and integrity
  - As secure as E
  - H has weaker properties than digital signatures
- $x, E_k(H(x))$  not recommended
  - Prove that sender has seen  $H(x)$
  - H must be collision resistant
  - Key  $k$  must be used only for this integrity function
- $E_k(x), H(x)$  not recommended
  - $H(x)$  can be used to check guesses on  $x$
  - H must be collision resistant

apr. '23 Hash functions 41

41