Hash functions

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
Emai: gianluca.dini@unipi.it
Version: 2024-04-08

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**

2

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-24 Hash functions 3

3

Informal properties

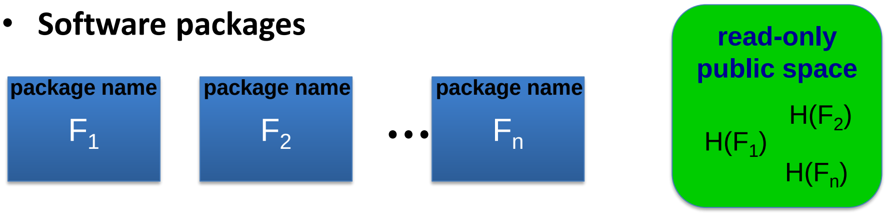
- The fingerprint must be *highly* sensitive to *all* input bits
  - Input «I am not a crook»
    - Hash (MD5): 6d17fcd4ae0e82fa4409f4ea6f4106a6
  - Input «I am not a cook»
    - Hash (MD5): 9ebe3d42d5c01fc59fe3daacbf42f515
- https://www.fileformat.info/tool/hash.htm

Apr-24 Hash functions 4

4

Example: protecting files

- **Software packages**



- 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

5

Example: protecting files

Prelievo da WinRAR.it

- Se il prelievo non è ancora partito, clicca **qui** per scaricare la versione richiesta.
- **Oppure torna alla pagina dei prelievi file**.

Verifica Integrità del file appena prelevato (checksum)

Nome File: WinRAR-x64-600b1it.exe

Dimensione: 3.442 K

MD5: c11ac9a41e5d178e65417faa6dccf75f

SHA-1: c9a2e9ca312573aaaa7b0c16fd49cb3ce40bf54f

SHA-256: 07a60c7da09679960aa2e9e7335194506cff71caebf0be62b97069d8619221f6

6

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-24 Hash functions 7

7

Security properties [1/2]

- 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' ≠ x such that h(x) = h(x')

Apr-24 Hash functions 8

8

Security properties [2/2]

- 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-24 Hash functions 9

9

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-24 Hash functions 10

10

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

Attacking Hash Functions

- 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

Black box attacks →

- 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)

Specific Black box Attacks ↓

- Guessing attack
  - find a 2nd 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)

Guessing attack

- Objective: to find a 2nd 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;
}
```


Guessing attack

- 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

Birthday attack →

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

Birthday attack →

- The birthday attack algorithm
  1. Choose $N = 2^{n/2}$ random input messages $x_1, x_2, \ldots, x_N$ (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%

Birthday paradox: intuition →

- Problem #1.
  - In a room of t = 23 people, what is the probability that at least a person is born on 25 December?
    - Answer: 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

Birthday attack →

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

Birthday attack

↓

UNIVERSITÀ DI PISA

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

Apr-24 Hash functions 21

21

Hash functions

HOW TO BUILD HASH FUNCTIONS

Apr-24 Hash functions 22

22

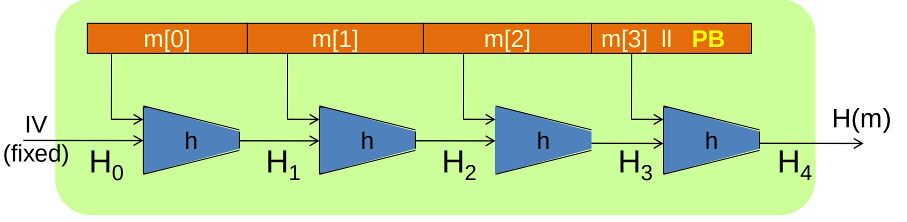
Types of hash functions

- Dedicated hash functions
- Block cipher-based hash functions

How to build a hash function

- 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.

The Merkle-Damgard iterated construction



- Compression function h: T × X ⟶ 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

Merkle-Damgard collision resistance

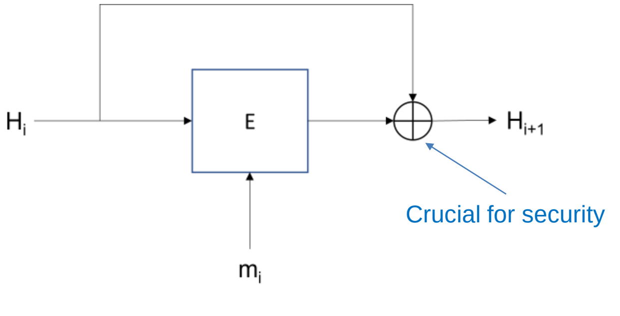
- THEOREM. if compression function h is collision resistant (and message length is part of the input) then so is H.
  - Proof (by contradiction)
    - Collision on H ➔ collision on h. Q.E.D.
- Comment
  - To construct a CRHF, it *suffices* to construct a collision resistant compression function

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

Davies-Meyer

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



Exercise

- Problem
  - If we remove the xor, the compression function is not collision resistant anymore.
  - Proof (by contradiction)
    - Remove the xor ➔ 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’) ➔ H’ = D(m’, E(m, H))

Q.E.D.

Apr-24 Hash functions 29

29

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-24 Hash functions 30

30

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: $O(2^{39})$

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

Other hash functions

- SHA-2 (NIST 2002)
  - 256-bit, 384-bit or 512-bit output lenght
  - 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 lenght

34

Hash functions

USES OF HASH FUNCTIONS

35

Uses of hash functions

- 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

Hash Functions

AUTHENTICATION OF ORIGIN

Integrity vs authentication

- 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

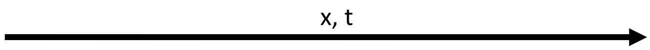
Use of hash functions for authentication

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

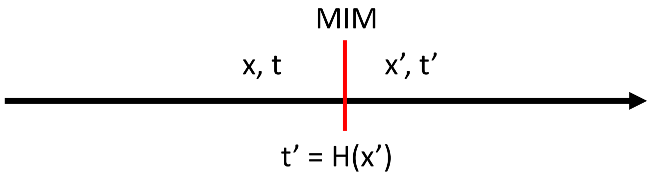
Authentic channel →

Bob

- Alice
  - Let t = H(x)



- MIM attack

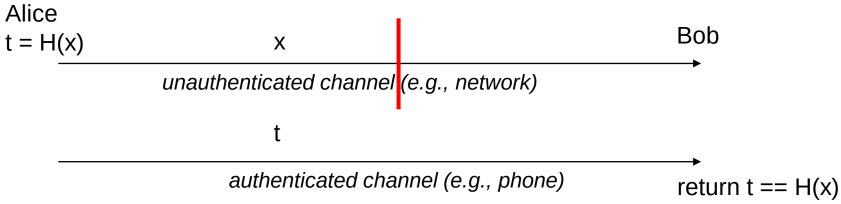


Apr-24 Hash functions 40

40

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-24 Hash functions 41

41

Hash functions with block ciphers

UNIVERSITÀ DI PISA

- $E_k(x||H(x))$ 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