An Introduction to Hash Functions

Christophe Clavier - Florent Bruguier

University of Limoges - University of Montpellier

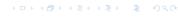


C. Clavier - F. Bruguier (Unilim - UM)

- 1 What is a Hash Function?
 - Definition and properties
 - Examples of hash functions
 - How does it work ?
- Security of Hash Functions
 - Security requirements
 - Security considerations
 - Complexity figures
- Applications
 - Secured password storage
 - Data integrity
 - Entity authentication
 - Message authentication
 - Digital signature

- Generic Attacks
 - Birthday paradox
 - Collision search
- Some Dedicated Attacks
 - A burst of new attacks
 - The SHA-3 competition





Outline What is a Hash Function? Security of Hash Functions Applications Generic Attacks Some Dedicated Attacks

O •0000 00000 00000 0000000 00000

- What is a Hash Function ?
 - Definition and properties
 - Examples of hash functions
 - How does it work ?
- Security of Hash Functions
 - Security requirements
 - Security considerations
 - Complexity figures
- Applications
 - Secured password storage
 - Data integrity
 - Entity authentication
 - Message authentication
 - Digital signature

- Generic Attacks
 - Birthday paradox
 - Collision search
- 5 Some Dedicated Attacks
 - A burst of new attacks
 - The SHA-3 competition



◆ロト ◆団 ト ◆ 恵 ト ◆ 恵 ・ 夕 Q で

C. Clavier - F. Bruguier (Unilim - UM)

An Introduction to Hash Functions

Hanoi, April 2020

3 / 34

Outline What is a Hash Function

Security of Hash Function

tions Generic Att

Some Dedicated Attack

Definition and properties

$$\begin{array}{ccc} \mathcal{H}: \{0,1\}^* & \longrightarrow \{0,1\}^n \\ m & \longmapsto \mathcal{H}(m) \end{array}$$



$$\mathcal{H}: \{0,1\}^* \longrightarrow \{0,1\}^n$$

$$m \longmapsto \mathcal{H}(m)$$

ullet takes a message m of arbitrary length as an input



An Introduction to Hash Functions

Hanoi, April 2020

4 / 34

Outline What is a Hash Function

Security of Hash Function

lications Generic Atta 2000 0000000 Some Dedicated Attacks

Definition and properties

$$\begin{array}{ccc} \mathcal{H}: \{0,1\}^* & \longrightarrow \{0,1\}^n \\ m & \longmapsto \mathcal{H}(m) \end{array}$$

- takes a message m of arbitrary length as an input
- output a fixed length message digest or hash value of 128 to 512 bits





$$\begin{array}{ccc} \mathcal{H}: \{0,1\}^* & \longrightarrow \{0,1\}^n \\ m & \longmapsto \mathcal{H}(m) \end{array}$$

- ullet takes a message m of arbitrary length as an input
- output a fixed length message digest or hash value of 128 to 512 bits



A hash function is \dots



An Introduction to Hash Functions

Hanoi, April 2020

4 / 34

Outline What is a Hash Function in Outline Out

Security of Hash Function

olications Generic Atta

Some Dedicated Attack

Definition and properties

$$\begin{array}{ccc} \mathcal{H}: \{0,1\}^* & \longrightarrow \{0,1\}^n \\ m & \longmapsto \mathcal{H}(m) \end{array}$$

- takes a message m of arbitrary length as an input
- output a fixed length message digest or hash value of 128 to 512 bits



A hash function is ...

• public: this is not a secret



$$\begin{array}{ccc} \mathcal{H}: \{0,1\}^* & \longrightarrow \{0,1\}^n \\ m & \longmapsto \mathcal{H}(m) \end{array}$$

- takes a message m of arbitrary length as an input
- output a fixed length message digest or hash value of 128 to 512 bits



A hash function is ...

- public: this is not a secret
- deterministic: anybody can compute $\mathcal{H}(m)$ unambiguously



An Introduction to Hash Functions

Hanoi, April 2020

4 / 34

Outline What is a Hash Function

Security of Hash Functions

lications Generic Atta

Some Dedicated Attac

Definition and properties

$$\begin{array}{ccc} \mathcal{H}: \{0,1\}^* & \longrightarrow \{0,1\}^n \\ m & \longmapsto \mathcal{H}(m) \end{array}$$

- takes a message *m* of arbitrary length as an input
- output a fixed length message digest or hash value of 128 to 512 bits



A hash function is ...

- public: this is not a secret
- ullet deterministic: anybody can compute $\mathcal{H}(m)$ unambiguously
- keyless: while used as an important cryptographic primitive



ullet Change one bit in $m\Longrightarrow$ about half the bits change in $\mathcal{H}(m)$



◆ロト ◆部 ト ◆ 恵 ト ◆ 恵 ・ 夕 Q ○

An Introduction to Hash Functions

Hanoi, April 2020

5 / 34

Outline What is a Hash Function ?

OOOOO

Definition and properties

Security of Hash Function: 00000 olications Generic Atta

Some Dedicated Attacks

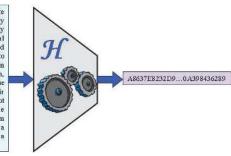
Definition and properties

- ullet Change one bit in $m\Longrightarrow$ about half the bits change in $\mathcal{H}(m)$
 - Expected random behavior



- Change one bit in $m \Longrightarrow$ about half the bits change in $\mathcal{H}(m)$
 - Expected random behavior

Cryptographic hash functions that compute a fixed size message digest from arbitrary size messages are widely used for many purposes in cryptography, including digital signatures. NIST was recently informed that researchers had discovered a way to "break" the current Federal Information Processing Standard SRA-1 algorithm, which has been in effect since 1994. The researchers have not yet published their complete results, so NIST has not confirmed these findings. However, the researchers are a reputable research team with expertise in this area. Previously, a brute force attack would expect to find a collision in 2²⁰ hash operations.





◆□▶ ◆御▶ ◆巻▶ ◆巻▶ ○巻 ○夕○○

C. Clavier - F. Bruguier (Unilim - UM)

An Introduction to Hash Function

Hanoi, April 2020

5 / 34

Security of Hash Function

ations Generic Attac

A8637E8232D9

Some Dedicated Attacks

Definition and properties

What is a Hash Function ?

- Change one bit in $m \Longrightarrow$ about half the bits change in $\mathcal{H}(m)$
 - Expected random behavior

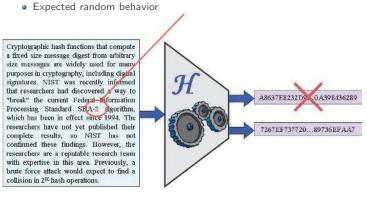
Cryptographic hash functions that compute a fixed size message digest from arbitrary size messages are widely used for many purposes in cryptography, including digital signatures. NIST was recently inferinced that researchers had discovered way to "break" the current Federal information Processing. Standard SEA-2 algorithm, which has been in effect since 1994. The researchers have not yet published their complete results, so NIST has not confirmed these findings. However, the researchers are a reputable research team with expertise in this area. Previously, a brute force attack would expect to find a collision in 280 hash operations.



0A398436289

7267EF737720...89736EFAA7

• Change one bit in $m \Longrightarrow$ about half the bits change in $\mathcal{H}(m)$



The hash value can be regarded as a fingerprint of the message

USTH

C. Clavier - F. Bruguier (Unilim - UM)

An Introduction to Hash Functions

Hanoi, April 2020

5 / 34

Outline What is a Hash Function ?

○ ○○●○

Examples of hash functions

OOOOO

Applications

Generic Attacks

Some Dedicated Attacks

Examples of hash functions

Example (some hash functions)



< ロ > → 데 > → 트 > → 트 > → 의 < (P)

Examples of hash functions

Example (some hash functions)

• MD4, MD5, RIPEMD have 128-bit hash values



◆ロト ◆部 ト ◆ 恵 ト ◆ 恵 ・ 夕 Q ○

An Introduction to Hash Functions

Hanoi, April 2020

6 / 3

Outline What is a Hash Function ?

OOOOO

Examples of hash functions

OOOOO

olications Generi

Some Dedicated Attack

Examples of hash functions

Example (some hash functions)

- MD4, MD5, RIPEMD have 128-bit hash values
- SHA-1 and RIPEMD-160 have 160-bit hash values



Examples of hash functions

Example (some hash functions)

- MD4, MD5, RIPEMD have 128-bit hash values
- SHA-1 and RIPEMD-160 have 160-bit hash values
- SHA-224 has a 224-bit hash value



◆□▶ ◆御▶ ◆巻▶ ◆巻▶ ● めぐぐ

An Introduction to Hash Functions

Hanoi, April 2020

6 / 3

Outline What is a Hash Function is

Cocooo

Examples of hash functions

Security of Hash Function

pplications Gen

Some Dedicated Attacks

Examples of hash functions

Example (some hash functions)

- MD4, MD5, RIPEMD have 128-bit hash values
- SHA-1 and RIPEMD-160 have 160-bit hash values
- SHA-224 has a 224-bit hash value
- SHA-256 has a 256-bit hash value



Examples of hash functions

Example (some hash functions)

- MD4, MD5, RIPEMD have 128-bit hash values
- SHA-1 and RIPEMD-160 have 160-bit hash values
- SHA-224 has a 224-bit hash value
- SHA-256 has a 256-bit hash value
- SHA-384 has a 384-bit hash value



<ロ > ← □ > ← □ > ← □ > ← □ = ・ の Q (~)

An Introduction to Hash Functions

Hanoi, April 2020

6 / 34

Outline What is a Hash Function

OOOOO

Examples of hash functions

Security of Hash Function

Applications

Generic Attacks

Some Dedicated Attack

Examples of hash functions

Example (some hash functions)

- MD4, MD5, RIPEMD have 128-bit hash values
- SHA-1 and RIPEMD-160 have 160-bit hash values
- SHA-224 has a 224-bit hash value
- SHA-256 has a 256-bit hash value
- SHA-384 has a 384-bit hash value
- SHA-512 has a 512-bit hash value





The Merkle-Damgård construction

How to tackle with arbitrarily long inputs?

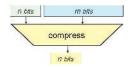


C. Clavier - F. Bruguier (Unilim - UM)

The Merkle-Damgård construction

How to tackle with arbitrarily long inputs?

• A compression function maintains an *n*-bit internal state while processing *m*-bit message blocks

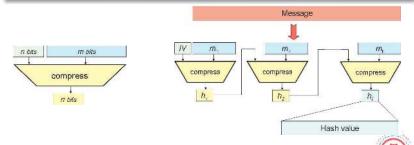




The Merkle-Damgård construction

How to tackle with arbitrarily long inputs?

- A compression function maintains an *n*-bit internal state while processing *m*-bit message blocks
- A chaining construction builds the hash function upon the compression function





C. Clavier - F. Bruguier (Unilim - UM)

An Introduction to Hash Functions

Hanoi, April 2020

7 / 34

sh Function ? Security of Hash Fu

Applications

OOOOOOOOOO

Some Dedicated Attacks

- What is a Hash Function ?
 - Definition and properties
 - Examples of hash functions
 - How does it work ?
- Security of Hash Functions
 - Security requirements
 - Security considerations
 - Complexity figures
- Applications
 - Secured password storage
 - Data integrity
 - Entity authentication
 - Message authentication
 - Digital signature

- Generic Attacks
 - Birthday paradox
 - Collision search
- **6** Some Dedicated Attacks
 - A burst of new attacks
 - The SHA-3 competition





Security requirements

Preimage resistance (one-wayness)

Given $y \in \{0,1\}^n$, it should be impossible to find x s.t. $\mathcal{H}(x) = y$



C. Clavier - F. Bruguier (Unilim - UM)

An Introduction to Hash Functions

Hanoi, April 2020

9 / 3

Outline What is a Hash Function ? Security of Hash Functions

O OOOOO O OOOO

Applications

Generic Attacks

Some Dedicated Attacks

Security requirements

Preimage resistance (one-wayness)

Given $y \in \{0,1\}^n$, it should be impossible to find x s.t. $\mathcal{H}(x) = y$

Idea: the function is not invertible



4□ > 4□ > 4 = > 4 = > = 900

Outline What is a Hash Function ?

Security of Hash Function

Applications

Generic Attacks

Some Dedicated Attacks

Security requirements

Preimage resistance (one-wayness)

Given $y \in \{0,1\}^n$, it should be impossible to find x s.t. $\mathcal{H}(x) = y$

Idea: the function is not invertible

Second preimage resistance

Given x, it should be impossible to find $x' \neq x$ s.t. $\mathcal{H}(x) = \mathcal{H}(x')$



An Introduction to Hash Functions

Hanoi, April 2020

9 / 34

Outline What is a Hash Function

Security of Hash Functions

oplications Gener

S

Some Dedicated Attack

Security requirements

Preimage resistance (one-wayness)

Given $y \in \{0,1\}^n$, it should be impossible to find x s.t. $\mathcal{H}(x) = y$

Idea: the function is not invertible

Second preimage resistance

Given x, it should be impossible to find $x' \neq x$ s.t. $\mathcal{H}(x) = \mathcal{H}(x')$

Idea: the knowledge of a preimage does not help to find a second one



Outline What is a Hash Function
O OOOOO

Security of Hash Functions

pplications

Generic Attacks

Some Dedicated Attacks

Security requirements

Preimage resistance (one-wayness)

Given $y \in \{0,1\}^n$, it should be impossible to find x s.t. $\mathcal{H}(x) = y$

Idea: the function is not invertible

Second preimage resistance

Given x, it should be impossible to find $x' \neq x$ s.t. $\mathcal{H}(x) = \mathcal{H}(x')$

Idea: the knowledge of a preimage does not help to find a second one

Collision resistance

It should be impossible to find x and x' s.t. $\mathcal{H}(x) = \mathcal{H}(x')$



C. Clavier - F. Bruguier (Unilim - UM)

An Internalization to Hook Franctions

Hanoi, April 2020

9 / 34

An Introduction to Hash Function

Outline What is a Hash Function

O OOOO

Security requirements

Security of Hash Functions

Applications Gene

S

Some Dedicated Attacks

Security requirements

Preimage resistance (one-wayness)

Given $y \in \{0,1\}^n$, it should be impossible to find x s.t. $\mathcal{H}(x) = y$

Idea: the function is not invertible

Second preimage resistance

Given x, it should be impossible to find $x' \neq x$ s.t. $\mathcal{H}(x) = \mathcal{H}(x')$

Idea: the knowledge of a preimage does not help to find a second one

Collision resistance

It should be impossible to find x and x' s.t. $\mathcal{H}(x) = \mathcal{H}(x')$

Idea: if someone provides $y = \mathcal{H}(x)$, and next reveals x, he can not cheat



Outline What is a Hash Function
O 00000

Security of Hash Functions

Applications

Generic Attacks

Some Dedicated Attacks

Security requirements

Preimage resistance (one-wayness)

Given $y \in \{0,1\}^n$, it should be impossible to find x s.t. $\mathcal{H}(x) = y$

Idea: the function is not invertible

Second preimage resistance

Given x, it should be impossible to find $x' \neq x$ s.t. $\mathcal{H}(x) = \mathcal{H}(x')$

Idea: the knowledge of a preimage does not help to find a second one

Collision resistance

It should be impossible to find x and x' s.t. $\mathcal{H}(x) = \mathcal{H}(x')$

Idea: if someone provides $y = \mathcal{H}(x)$, and next reveals x, he can not cheat

Question

What does impossible means?

TH

C. Clavier - F. Bruguier (Unilim - UM)

An Introduction to Hash Functions

Hanoi, April 2020

9 / 34

Outline What is a Hash Function ?
OOOOO

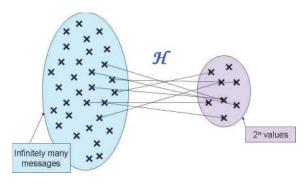
Security considerations

Security of Hash Functions

oplications Generic Atta

Some Dedicated Attacks

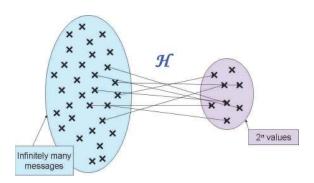
Security considerations



• There are infinitely many messages



Security considerations



- There are infinitely many messages
 - Preimages and second preimages always exist



An Introduction to Hash Functions

Hanoi, April 2020

10 / 34

Outline What is a Hash Function ? Security of Hash Functions

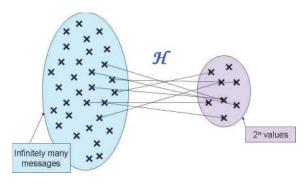
O 00000

Security considerations

Applicatio 000000 Generic Attacks

Some Dedicated Attacks

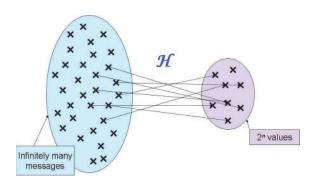
Security considerations



- There are infinitely many messages
 - Preimages and second preimages always exist
 - Collisions are unavoidable



Security considerations



- There are infinitely many messages
 - Preimages and second preimages always existCollisions are unavoidable
- Impossibility (absolute) so reduces to computational unfeasibility (relative)



Outline O	What is a Hash Function ?	Security of Hash Functions	Applications 000000	Generic Attacks	Some Dedicated Attacks
Security considerations					

Generic Attacks

• A generic attack is one whose complexity depends only on the size of the hash result, not on the details of the algorithm



Generic Attacks

- A generic attack is one whose complexity depends only on the size of the hash result, not on the details of the algorithm
- Some generic attacks apply to all hash functions



An Introduction to Hash Functions

Hanoi, April 2020

1 / 34

Outline What is a Hash Function
O OOOOO

Security of Hash Functions

ications Generic Attac

Some Dedicated Attacks

Generic Attacks

- A generic attack is one whose complexity depends only on the size of the hash result, not on the details of the algorithm
- Some generic attacks apply to all hash functions
 - Finding preimages or second preimages is always possible by trivial exhaustive search within 2ⁿ computations



Generic Attacks

- A generic attack is one whose complexity depends only on the size of the hash result, not on the details of the algorithm
- Some generic attacks apply to all hash functions
 - Finding preimages or second preimages is always possible by trivial exhaustive search within 2^n computations
 - Finding collisions is always possible within $2^{n/2}$ computations (birthday paradox)



<ロ > ←回 > ←回 > ← 差 > ← 差 > − 差 − − 9 へ(

An Introduction to Hash Functions

Hanoi, April 2020

11 / 34

Outline What is a Hash Function
O OOOOO

Security considerations

Security of Hash Functions

OOOOOO

Generic Attacks

Some Dedicated Attack

Generic Attacks

- A generic attack is one whose complexity depends only on the size of the hash result, not on the details of the algorithm
- Some generic attacks apply to all hash functions
 - Finding preimages or second preimages is always possible by trivial exhaustive search within 2ⁿ computations
 - Finding collisions is always possible within $2^{n/2}$ computations (birthday paradox)

A secure hash function must not be vulnerable to better attacks



• A machine able to perform 10⁹ computations per second will perform:



Security of Hash Functions

Complexity figures

- A machine able to perform 10^9 computations per second will perform: 2^{46} computations per day



- \bullet A machine able to perform 10^9 computations per second will perform:

 - 2⁴⁶ computations per day
 2⁵⁵ computations per year



Security of Hash Functions

Complexity figures

- A machine able to perform 10^9 computations per second will perform:

 - 2⁴⁶ computations per day
 2⁵⁵ computations per year
 2⁹⁰ computations in 15 billions years



- A machine able to perform 10⁹ computations per second will perform:

 - 2⁴⁶ computations per day
 2⁵⁵ computations per year
 2⁹⁰ computations in 15 billions years

One must choose the hash output size n so that an attacker can not reach $2^{n/2}$ computations



Complexity figures

- A machine able to perform 10^9 computations per second will perform:

 - 2⁴⁶ computations per day
 2⁵⁵ computations per year
 2⁹⁰ computations in 15 billions years

One must choose the hash output size n so that an attacker can not reach $2^{n/2}$ computations

•
$$n = 80$$
 \longrightarrow $2^{n/2} = 2^{40}$ (feasible)



- A machine able to perform 10⁹ computations per second will perform:

 - 2⁴⁶ computations per day
 2⁵⁵ computations per year
 2⁹⁰ computations in 15 billions years

One must choose the hash output size n so that an attacker can not reach $2^{n/2}$ computations

- n = 80
- $\rightarrow 2^{n/2} = 2^{40} \text{ (feasible)}$
- n = 128 (MD5)
- \longrightarrow $2^{n/2} = 2^{64}$ (becomes difficult)



Security of Hash Functions

Complexity figures

- A machine able to perform 10^9 computations per second will perform:

 - 2⁴⁶ computations per day
 2⁵⁵ computations per year
 2⁹⁰ computations in 15 billions years

One must choose the hash output size n so that an attacker can not reach $2^{n/2}$ computations

- \longrightarrow $2^{n/2} = 2^{40}$ (feasible) • *n* = 80
- \longrightarrow 2^{n/2} = 2⁶⁴ (becomes difficult) • n = 128 (MD5)
- \longrightarrow $2^{n/2} = 2^{80}$ (believed secure for the moment) • n = 160 (SHA-1)



- A machine able to perform 10⁹ computations per second will perform:
 - 2⁴⁶ computations per day
 - 2⁵⁵ computations per year
 - 2⁹⁰ computations in 15 billions years

One must choose the hash output size n so that an attacker can not reach $2^{n/2}$ computations

- n = 80 $\longrightarrow 2^{n/2} = 2^{40}$ (feasible)
- $n = 128 \text{ (MD5)} \longrightarrow 2^{n/2} = 2^{64} \text{ (becomes difficult)}$
- n = 160 (SHA-1) $\longrightarrow 2^{n/2} = 2^{80}$ (believed secure for the moment)
- $n = 256 \text{ (SHA-256)} \longrightarrow 2^{n/2} = 2^{128} \text{ (highly secure)}$



◆ロト ◆部 ト ◆ 恵 ト ◆ 恵 ・ 夕 Q (*)

An Introduction to Hash Functions

Hanoi, April 2020

12 / 34

Security of Hash Functions

Application

Generic Attacks

Some Dedicated Attacks

- What is a Hash Function ?
 - Definition and properties
 - Examples of hash functions
 - How does it work ?
- Security of Hash Functions
 - Security requirements
 - Security considerations
 - Complexity figures
- Applications
 - Secured password storage
 - Data integrity
 - Entity authentication
 - Message authentication
 - Digital signature

- Generic Attacks
 - Birthday paradox
 - Collision search
- Some Dedicated Attacks
 - A burst of new attacks
 - The SHA-3 competition



Secured password storage

• To be granted access to her account, Alice must present a password which is to be compared with a previously stored value



Secured password storage

- To be granted access to her account, Alice must present a password which is to be compared with a previously stored value
- Clear text password storage may be jeopardized (reading, modification) by unauthorized file access



Secured password storage

- To be granted access to her account, Alice must present a password which is to be compared with a previously stored value
- Clear text password storage may be jeopardized (reading, modification) by unauthorized file access

Solution

Store a list of {user, $\mathcal{H}(user's password)$ }

When Alice identifies herself by presenting password p, check that $\mathcal{H}(p) = \mathcal{H}(\mathsf{Alice's\ password})$



<ロ > → □ → → □ → → □ → つへで

An Introduction to Hash Functions

Hanoi, April 2020

14 / 34

Outline What is a Hash Function
O 00000

Security of Hash Functions 00000 Application ○●○○○○ Generic Attacks

Some Dedicated Attacks

Secured password storage

- To be granted access to her account, Alice must present a password which is to be compared with a previously stored value
- Clear text password storage may be jeopardized (reading, modification) by unauthorized file access

Solution

Store a list of {user, $\mathcal{H}(user's password)$ }

When Alice identifies herself by presenting password p, check that $\mathcal{H}(p)=\mathcal{H}(\mathsf{Alice's\ password})$

• This solution does not prevent from dictionary attacks



Secured password storage

- To be granted access to her account, Alice must present a password which is to be compared with a previously stored value
- Clear text password storage may be jeopardized (reading, modification) by unauthorized file access

Solution

Store a list of $\{user, \mathcal{H}(user's password)\}\$

When Alice identifies herself by presenting password p, check that $\mathcal{H}(p) = \mathcal{H}(\mathsf{Alice's\ password})$

- This solution does not prevent from dictionary attacks
 - But usage of salt technique may circumvent the problem



An Introduction to Hash Functions

Hanoi, April 2020

14 / 34

Outline What is a Hash Function
O OOOOO

Security of Hash Functions

Applications

OOOOO

Generic Attacks

Some Dedicated Attacks

Secured password storage

- To be granted access to her account, Alice must present a password which is to be compared with a previously stored value
- Clear text password storage may be jeopardized (reading, modification) by unauthorized file access

Solution

Store a list of $\{user, \mathcal{H}(user's password)\}$

When Alice identifies herself by presenting password p, check that $\mathcal{H}(p) = \mathcal{H}(\text{Alice's password})$

- This solution does not prevent from dictionary attacks
 - But usage of salt technique may circumvent the problem
- The hash function needs to be preimage resistant



Data integrity

• Alice downloads a text *m* from an internet server



An Introduction to Hash Functions

Hanoi, April 2020

15 / 34

Outline What is a Hash Function 7

Security of Hash Functions
00000

oplications Gener ○●○○○ ○○○ Some Dedicated Attacks

Data integrity

- ullet Alice downloads a text m from an internet server
- She wants to make sure the text hasn't been changed since it has been sent by the server



Data integrity

- Alice downloads a text *m* from an internet server
- She wants to make sure the text hasn't been changed since it has been sent by the server

Solution

Add a $\mathcal{H}(m)$ of the text m next to it, so that anybody can check whether the hash value matches that of the downloaded text



[□▶◀♬▶◀불▶◀불▶ 불 쒸٩(

An Introduction to Hash Functions

Hanoi, April 2020

15 / 34

Outline What is a Hash Function

O OOOOO

Data integrity

ecurity of Hash Functions

Applications

Generic Attacks

Some Dedicated Attack

Data integrity

- Alice downloads a text m from an internet server
- She wants to make sure the text hasn't been changed since it has been sent by the server

Solution

Add a $\mathcal{H}(m)$ of the text m next to it, so that anybody can check whether the hash value matches that of the downloaded text

• This solution does not prevent from an attacker who controls the web site



Data integrity

- Alice downloads a text *m* from an internet server
- She wants to make sure the text hasn't been changed since it has been sent by the server

Solution

Add a $\mathcal{H}(m)$ of the text m next to it, so that anybody can check whether the hash value matches that of the downloaded text

- This solution does not prevent from an attacker who controls the web site
- The hash function needs to be second preimage resistant



<ロ > ←回 > ←回 > ← 直 > ← 直 > への

An Introduction to Hash Functions

Hanoi, April 2020

15 / 34

Outline What is a Hash Function ? Sec

ish Functions Appli OOC Generic Attacks

Some Dedicated Attacks

Entity authentication

• Juliet wants to identify Romeo on the phone



Entity authentication

- Juliet wants to identify Romeo on the phone
- Password based authentication is not appropriate (eavesdropper → one-time password!)



An Introduction to Hash Functions

Hanoi, April 2020

16 / 34

Outline What is a Hash Function
O 00000

OOOOO

Applications

Generic Attacks 00000000000 Some Dedicated Attack

Entity authentication

- Juliet wants to identify Romeo on the phone
- $\bullet \ \ \, \mathsf{Password} \ \, \mathsf{based} \ \, \mathsf{authentication} \ \, \mathsf{is} \ \, \mathsf{not} \ \, \mathsf{appropriate} \\ \mathsf{(eavesdropper} \to \mathsf{one-time} \ \, \mathsf{password!})$

Solution

Challenge-response protocol + shared secret s



Entity authentication

- Juliet wants to identify Romeo on the phone
- Password based authentication is not appropriate (eavesdropper → one-time password!)

Solution

Challenge-response protocol + shared secret s

 $oldsymbol{0}$ Juliet: sends a random challenge r



<ロ > ←回 > ←回 > ← 直 > ← 直 > への

im - UM) An Introduction to Hash Functio

Hanoi, April 2020

16 / 34

Outline What is a Hash Function
O 00000

Security of Hash Functions

Applications

Generic Attacks

Some Dedicated Attack

Entity authentication

- Juliet wants to identify Romeo on the phone
- $\bullet \ \ \, \mathsf{Password} \ \, \mathsf{based} \ \, \mathsf{authentication} \ \, \mathsf{is} \ \, \mathsf{not} \ \, \mathsf{appropriate} \\ (\mathsf{eavesdropper} \, \to \, \mathsf{one\text{-}time} \ \, \mathsf{password!})$

Solution

Challenge-response protocol + shared secret s

- lacktriangledown Juliet: sends a random challenge r
- ② Romeo: sends challenge response as $C = \mathcal{H}(s||r)$



Entity authentication

- Juliet wants to identify Romeo on the phone
- Password based authentication is not appropriate $(\mathsf{eavesdropper} \to \mathsf{one}\text{-}\mathsf{time}\;\mathsf{password!})$

Solution

Challenge-response protocol + shared secret s

- Juliet: sends a random challenge r
- 2 Romeo: sends challenge response as $C = \mathcal{H}(s||r)$
- 3 Juliet: computes $\mathcal{H}(s||r)$ and compares it with C



Message authentication

• John and Chris share a secret key K



Message authentication

- ullet John and Chris share a secret key K
- ullet John wants to send a message m to Chris without anybody being able to modify it



◆ロト ◆部 ト ◆ 恵 ト ◆ 恵 ・ か へ ○ ○

An Introduction to Hash Functions

Hanoi, April 2020

17 / 34

Outline What is a Hash Function ? Security of Hash Functions Applications

O OOOO OOOO

Message authentication

Generic Attacks

Some Dedicated Attacks

Message authentication

- ullet John and Chris share a secret key K
- ullet John wants to send a message m to Chris without anybody being able to modify it
- Chris wants to make sure the sender is John and the message wasn't tampered with



Message authentication

- ullet John and Chris share a secret key K
- ullet John wants to send a message m to Chris without anybody being able to modify it
- Chris wants to make sure the sender is John and the message wasn't tampered with







Message authentication

- John and Chris share a secret key K
- John wants to send a message m to Chris without anybody being able to modify it
- Chris wants to make sure the sender is John and the message wasn't tampered with

Solution

• HMAC(m,K) computes a Message Authentication Code (MAC) on m using a keyed hash function



Message authentication

- ullet John and Chris share a secret key K
- ullet John wants to send a message m to Chris without anybody being able to modify it
- Chris wants to make sure the sender is John and the message wasn't tampered with

Solution

- HMAC(m,K) computes a Message Authentication Code (MAC) on m using a keyed hash function
- HMAC $(m, K) = \mathcal{H}(K \oplus \text{opad}||\mathcal{H}(K \oplus \text{ipad}||m))$



An Introduction to Hash Functions

Hanoi, April 2020

7 / 34

Outline What is a Hash Function ? Security of Hash Fu

Applications Generic

Generic Attacks

Some Dedicated Attacks

Digital signature

ullet The previous technique does not provide non-repudiation (John could claim Chris produced the MAC himself, using shared key K)



- The previous technique does not provide non-repudiation (John could claim Chris produced the MAC himself, using shared key K)
- To ensure non-repudiation, use digital signatures



An Introduction to Hash Functions

Hanoi, April 2020

8 / 34

Outline What is a Hash Function 7
O OOOOO

Digital signature

rity of Hash Functions 200 Generic Attacks

Some Dedicated Attacks

Digital signature

- \bullet The previous technique does not provide non-repudiation (John could claim Chris produced the MAC himself, using shared key K)
- To ensure non-repudiation, use digital signatures
 - Only John can sign the message using his private key



Digital signature

- ullet The previous technique does not provide non-repudiation (John could claim Chris produced the MAC himself, using shared key K)
- To ensure non-repudiation, use digital signatures
 - Only John can sign the message using his private key
 - Anybody can verify the signature using John's public key



◆□▶ ◆御▶ ◆巻▶ ◆巻▶ ○巻 ○夕○○

An Introduction to Hash Functions

Hanoi, April 2020

18 / 34

Outline What is a Hash Function in OOOOO

Digital signature

sh Functions Applic

Generic Attacks

Some Dedicated Attacks

Digital signature

- ullet The previous technique does not provide non-repudiation (John could claim Chris produced the MAC himself, using shared key K)
- To ensure non-repudiation, use digital signatures
 - Only John can sign the message using his private key
 - Anybody can verify the signature using John's public key
- But . . .



- ullet The previous technique does not provide non-repudiation (John could claim Chris produced the MAC himself, using shared key K)
- To ensure non-repudiation, use digital signatures
 - Only John can sign the message using his private key
 - Anybody can verify the signature using John's public key
- But . . .
 - Signing several megabytes is too slow



An Introduction to Hash Functions

Hanoi, April 2020

18 / 34

Outline What is a Hash Function
O OOOOO

of Hash Functions Ap

Generic Attacks

Some Dedicated Attack

Digital signature

- \bullet The previous technique does not provide non-repudiation (John could claim Chris produced the MAC himself, using shared key K)
- To ensure non-repudiation, use digital signatures
 - Only John can sign the message using his private key
 - Anybody can verify the signature using John's public key
- But . . .
 - Signing several megabytes is too slow
 - Signature algorithms may be malleable



- The previous technique does not provide non-repudiation (John could claim Chris produced the MAC himself, using shared key K)
- To ensure non-repudiation, use digital signatures
 - Only John can sign the message using his private key
 - Anybody can verify the signature using John's public key
- But ...
 - Signing several megabytes is too slow
 - Signature algorithms may be malleable

Solution

Only sign the hash of the message: $Sign(\mathcal{H}(m))$



An Introduction to Hash Functions

Hanoi, April 2020

18 / 34

Outline What is a Hash Function ?
O 00000

curity of Hash Functions

olications Generic Atta

Some Dedicated Attack

Digital signature

- ullet The previous technique does not provide non-repudiation (John could claim Chris produced the MAC himself, using shared key K)
- To ensure non-repudiation, use digital signatures
 - Only John can sign the message using his private key
 - Anybody can verify the signature using John's public key
- But . .
 - Signing several megabytes is too slow
 - Signature algorithms may be malleable

Solution

Only sign the hash of the message: $Sign(\mathcal{H}(m))$

• If the message is modified, the signature is not valid anymore . . .



- The previous technique does not provide non-repudiation (John could claim Chris produced the MAC himself, using shared key K)
- To ensure non-repudiation, use digital signatures
 - Only John can sign the message using his private key
 - Anybody can verify the signature using John's public key
- But ...
 - Signing several megabytes is too slow
 - Signature algorithms may be malleable

Solution

Only sign the hash of the message: $Sign(\mathcal{H}(m))$

- If the message is modified, the signature is not valid anymore . . .
- ... provided the hash function is collision resistant



An Introduction to Hash Functions

Hanoi, April 2020

18 / 34

Security of Hash Function

Applicatio

Generic Attacks

Some Dedicated Attacks

- What is a Hash Function?
 - Definition and properties
 - Examples of hash functions
 - How does it work ?
- Security of Hash Functions
 - Security requirements
 - Security considerations
 - Complexity figures
- Applications
 - Secured password storage
 - Data integrity
 - Entity authentication
 - Message authentication
 - Digital signature

- Generic Attacks
 - Birthday paradox
 - Collision search
- Some Dedicated Attacks
 - A burst of new attacks
 - The SHA-3 competition





Birthday paradox

The classical problem

Question

How many persons are needed for having more than 50% chance that two of them share the same birthday ?



C. Clavier - F. Bruguier (Unilim - UM)

An Introduction to Hash Functions

Hanoi, April 2020

20 / 34

Outline What is a Hash Function
O OOOOO

Birthday paradox

Functions Applic

Generic Attacks

Some Dedicated Attacks

Birthday paradox

The classical problem

Question

How many persons are needed for having more than 50% chance that two of them share the same birthday ?

Let p(365, m) the probability that, given m persons, at least two share the same birthday. Then . . .



<ロ > ← □

Birthday paradox

The classical problem

Question

How many persons are needed for having more than 50% chance that two of them share the same birthday ?

Let p(365, m) the probability that, given m persons, at least two share the same birthday. Then . . .

• Intuitively, p(365, m) increases with m



An Introduction to Hash Functions

Hanoi, April 2020

20 / 34

Outline What is a Hash Function
O OOOOO

Birthday paradox

Security of Hash Functions

Applications

OOOOOOOO

Some Dedicated Attack

Birthday paradox

The classical problem

Question

How many persons are needed for having more than 50% chance that two of them share the same birthday ?

Let p(365, m) the probability that, given m persons, at least two share the same birthday. Then . . .

- ullet Intuitively, p(365, m) increases with m
- Obviously, p(365, 1) = 0



Birthday paradox
The classical problem

Question

How many persons are needed for having more than 50% chance that two of them share the same birthday ?

Let p(365, m) the probability that, given m persons, at least two share the same birthday. Then . . .

- Intuitively, p(365, m) increases with m
- Obviously, p(365, 1) = 0
- Also, p(365, 366) = 1



An Introduction to Hash Functions

Hanoi, April 2020

20 / 34

Outline What is a Hash Function
O OOOOO

Security of Hash Functions

Applications

Generic Attacks

Some Dedicated Attacks

Birthday paradox

The classical problem

Question

How many persons are needed for having more than 50% chance that two of them share the same birthday ?

Let p(365, m) the probability that, given m persons, at least two share the same birthday. Then \dots

- ullet Intuitively, p(365, m) increases with m
- Obviously, p(365, 1) = 0
- Also, p(365, 366) = 1
- For some m^* , $p(365, m^* 1) \leqslant \frac{1}{2}$ and $p(365, m^*) > \frac{1}{2}$



Birthday paradox

The classical problem

Question

How many persons are needed for having more than 50% chance that two of them share the same birthday ?

Let p(365, m) the probability that, given m persons, at least two share the same birthday. Then . . .

- Intuitively, p(365, m) increases with m
- Obviously, p(365, 1) = 0
- Also, p(365, 366) = 1
- For some m^* , $p(365, m^* 1) \leqslant \frac{1}{2}$ and $p(365, m^*) > \frac{1}{2}$



$$p(365, 22) = 0,476$$

 $p(365, 23) = 0,507$ $\longrightarrow m^* = 23$

(ロ) (部) (ま) (ま) ま り()

HIGU

C. Clavier - F. Bruguier (Unilim - UM)

An Introduction to Hash Functions

Hanoi, April 2020

20 / 34

Outline What is a Hash Function
O OOOOO

Birthday paradox

OOOOO

ns Generic At

Some Dedicated Attack

Birthday paradox

Generalization

ullet An urn contains t balls numbered 1 to t



<ロ > ← □

Birthady paradox

Birthday paradox

Generalization

- ullet An urn contains t balls numbered 1 to t
- ullet m balls are drawn at random from the urn



An Introduction to Hash Functions

Hanoi, April 2020

21 / 34

Outline What is a Hash Function
O OOOOO

Birthday paradox

ority of Hash Functions App
OOO OO

OOOOOOOO

Some Dedicated Attack

Birthday paradox

Generalization

- ullet An urn contains t balls numbered 1 to t
- ullet m balls are drawn at random from the urn

What is the probability p(t, m) of at least one coincidence (a ball drawn at least twice) ?



Birthday paradox

Generalization

- ullet An urn contains t balls numbered 1 to t
- ullet m balls are drawn at random from the urn

What is the probability p(t, m) of at least one coincidence (a ball drawn at least twice) ?

If $m = \mathcal{O}(\sqrt{t})$ and $t \to \infty$ then:

$$p(t,m) \longrightarrow 1 - exp\left(-rac{m(m-1)}{2t} + \mathcal{O}\left(rac{1}{\sqrt{t}}
ight)
ight) pprox 1 - exp\left(-rac{m^2}{2t}
ight)$$



An Introduction to Hash Functions

Hanoi, April 2020

21 / 34

Outline What is a Hash Function
O OOOOO

Security of Hash Functions

Applications

Generic Attacks

Some Dedicated Attacks

Birthday paradox

Generalization

Proof

Let q(t, m) = 1 - p(t, m) the probability that each m balls differ:

$$\begin{split} q(t,m) &= \prod_{k=0}^{m-1} \left(1 - \frac{k}{t}\right) \\ ln(q(t,m)) &= \sum_{k=0}^{m-1} ln\left(1 - \frac{k}{t}\right) \\ &= \sum_{k=0}^{m-1} \left[-\frac{k}{t} + o\left(\frac{k}{t}\right) \right] \\ &= -\frac{m(m-1)}{2t} + \mathcal{O}\left(\frac{m}{t}\right) & \text{if } m = \mathcal{O}(\sqrt{t}) \\ p(t,m) &\stackrel{t \to \infty}{\longrightarrow} 1 - \exp\left(-\frac{m(m-1)}{2t} + \mathcal{O}\left(\frac{1}{\sqrt{t}}\right)\right) \approx 1 - \exp\left(-\frac{m^2}{2t}\right) \end{split}$$

USTH

Birthday paradox

Applications

$$p(t,m) \approx 1 - \exp\left(-rac{m^2}{2t}
ight)$$



Birthday paradox Applications

$$p(t,m) \approx 1 - \exp\left(-rac{m^2}{2t}
ight)$$
 $p(t,m) = rac{1}{2} \Longleftrightarrow m = \sqrt{2.ln(2).t} \simeq 1,18\sqrt{t}$



Birthday parado

Birthday paradox

Applications

$$p(t,m) \approx 1 - exp\left(-\frac{m^2}{2t}\right)$$

$$p(t,m) = \frac{1}{2} \Longleftrightarrow m = \sqrt{2.ln(2).t} \simeq 1,18\sqrt{t}$$

Birthday problem

 $t = 365 \Rightarrow m = 22,54$

Among only 23 persons, you'd better bet on a birthday coincidence



An Introduction to Hash Functions

Hanoi, April 2020

23 / 34

Outline What is a Hash Function
O 00000

Security of Hash Functions

Applications

eneric Attacks

Some Dedicated Attacks

Birthday paradox

Applications

$$p(t,m) \approx 1 - exp\left(-rac{m^2}{2t}
ight)$$
 $p(t,m) = rac{1}{2} \Longleftrightarrow m = \sqrt{2.ln(2).t} \simeq 1,18\sqrt{t}$

Birthday problem

 $t = 365 \Rightarrow m = 22,54$

Among only 23 persons, you'd better bet on a birthday coincidence

Hash collisions

For an *n*-bit hash function $(t = 2^n)$, a collision may be expected after having computed about $2^{n/2}$ hash values





Naive method

A first collision search algorithm:

① Choose m_1 at random and store $(m_1, \mathcal{H}(m_1))$



Naive method

A first collision search algorithm:

- Choose m_1 at random and store $(m_1, \mathcal{H}(m_1))$
- ② Choose m_2 at random, check whether $\mathcal{H}(m_2)$ has ever been computed, else store $(m_2, \mathcal{H}(m_2))$



Naive method

A first collision search algorithm:

- **①** Choose m_1 at random and store $(m_1, \mathcal{H}(m_1))$
- ② Choose m_2 at random, check whether $\mathcal{H}(m_2)$ has ever been computed, else store $(m_2, \mathcal{H}(m_2))$
- **3** Choose m_3 at random, check whether $\mathcal{H}(m_3)$ has ever been computed, else store $(m_3, \mathcal{H}(m_3))$



An Introduction to Hash Functions

Hanoi, April 2020

24 / 34

Outline What is a Hash Function
0 00000

Security of Hash Functions OOOOO Generic Attac

Some Dedicated Attacks

Naive method

A first collision search algorithm:

- Choose m_1 at random and store $(m_1, \mathcal{H}(m_1))$
- ② Choose m_2 at random, check whether $\mathcal{H}(m_2)$ has ever been computed, else store $(m_2, \mathcal{H}(m_2))$
- **②** Choose m_3 at random, check whether $\mathcal{H}(m_3)$ has ever been computed, else store $(m_3, \mathcal{H}(m_3))$
- $oldsymbol{0}$... and so on until the current hash is already present in the hash array



Naive method

A first collision search algorithm:

- **①** Choose m_1 at random and store $(m_1, \mathcal{H}(m_1))$
- ② Choose m_2 at random, check whether $\mathcal{H}(m_2)$ has ever been computed, else store $(m_2, \mathcal{H}(m_2))$
- **3** Choose m_3 at random, check whether $\mathcal{H}(m_3)$ has ever been computed, else store $(m_3, \mathcal{H}(m_3))$
- and so on until the current hash is already present in the hash array

This algorithm requires $\mathcal{O}(\sqrt{2^n})$ time and $\mathcal{O}(\sqrt{2^n})$ memory



An Introduction to Hash Functions

Hanoi, April 2020

24 / 34

Outline What is a Hash Function 7

Security of Hash Function

ations Generic Attack

Some Dedicated Attacks

Floyd's collision finding algorithm

• Starting from an arbitrary x_0 , consider the sequence of iterated hashes

$$x_{i+1} = \mathcal{H}(x_i)$$



Floyd's collision finding algorithm

• Starting from an arbitrary x_0 , consider the sequence of iterated hashes

$$x_{i+1} = \mathcal{H}(x_i)$$

• After about $\sqrt{2^n}$ steps, two sequence elements x_{α} and x_{β} will be the same:

$$\begin{cases} x_{\alpha} = x_{\beta} \\ x_{\alpha-1} \neq x_{\beta-1} \end{cases} \tag{1}$$



(ロト 4례 > 4분 > 4분 > 분 - 위익(

An Introduction to Hash Functions

Hanoi, April 2020

25 / 34

Outline What is a Hash Function
O OOOOO

Security of Hash Functio

ications Generic A

Some Dedicated Attacks

Floyd's collision finding algorithm

• Starting from an arbitrary x_0 , consider the sequence of iterated hashes

$$x_{i+1} = \mathcal{H}(x_i)$$

• After about $\sqrt{2^n}$ steps, two sequence elements x_{α} and x_{β} will be the same:

$$\begin{cases} x_{\alpha} = x_{\beta} \\ x_{\alpha-1} \neq x_{\beta-1} \end{cases} \tag{1}$$

ullet This gives a collision on the hash function since $\mathcal{H}(x_{\alpha-1})=\mathcal{H}(x_{\beta-1})$



Floyd's collision finding algorithm

• Starting from an arbitrary x_0 , consider the sequence of iterated hashes

$$x_{i+1} = \mathcal{H}(x_i)$$

• After about $\sqrt{2^n}$ steps, two sequence elements x_α and x_β will be the same:

$$\begin{cases} x_{\alpha} = x_{\beta} \\ x_{\alpha-1} \neq x_{\beta-1} \end{cases} \tag{1}$$

- This gives a collision on the hash function since $\mathcal{H}(x_{\alpha-1}) = \mathcal{H}(x_{\beta-1})$
- From then on, the sequence will endlessly cycle



An Introduction to Hash Functions

Hanoi, April 2020

25 / 34

Outline What is a Hash Function
O OOOOO

Security of Hash Functio

s Generic Attac

Some Dedicated Attack

Floyd's collision finding algorithm

• Starting from an arbitrary x_0 , consider the sequence of iterated hashes

$$x_{i+1} = \mathcal{H}(x_i)$$

• After about $\sqrt{2^n}$ steps, two sequence elements x_{α} and x_{β} will be the same:

$$\begin{cases} x_{\alpha} = x_{\beta} \\ x_{\alpha-1} \neq x_{\beta-1} \end{cases} \tag{1}$$

- This gives a collision on the hash function since $\mathcal{H}(x_{\alpha-1})=\mathcal{H}(x_{\beta-1})$
- From then on, the sequence will endlessly cycle
 - ullet α is called the *tail length*



Floyd's collision finding algorithm

• Starting from an arbitrary x_0 , consider the sequence of iterated hashes

$$x_{i+1} = \mathcal{H}(x_i)$$

• After about $\sqrt{2^n}$ steps, two sequence elements x_{α} and x_{β} will be the same:

$$\begin{cases} x_{\alpha} = x_{\beta} \\ x_{\alpha-1} \neq x_{\beta-1} \end{cases} \tag{1}$$

- This gives a collision on the hash function since $\mathcal{H}(x_{\alpha-1}) = \mathcal{H}(x_{\beta-1})$
- From then on, the sequence will endlessly cycle
 - ullet α is called the *tail length*
 - $\delta = \beta \alpha$ is called the *cycle length*



Floyd's collision finding algorithm

 \bullet Starting from an arbitrary x_0 , consider the sequence of iterated hashes

$$x_{i+1} = \mathcal{H}(x_i)$$

• After about $\sqrt{2^n}$ steps, two sequence elements x_{α} and x_{β} will be the same:

$$\begin{cases} x_{\alpha} = x_{\beta} \\ x_{\alpha-1} \neq x_{\beta-1} \end{cases} \tag{1}$$

- ullet This gives a collision on the hash function since $\mathcal{H}(x_{\alpha-1})=\mathcal{H}(x_{\beta-1})$
- From then on, the sequence will endlessly cycle
 - ullet α is called the *tail length*
 - $\delta = \beta \alpha$ is called the *cycle length*

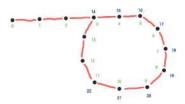
Problem .

How to find two such equal sequence elements ?





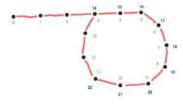
Floyd's collision finding algorithm (example)



 \bullet A collision occurred at $\alpha=$ 3 and $\beta=$ 14



Floyd's collision finding algorithm (example)



- \bullet A collision occurred at $\alpha=$ 3 and $\beta=$ 14
- \bullet The cycle length is $\delta=11$



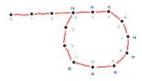
Security of Hash Functions

Applications

Generic Attacks ○○○○○○○○ Some Dedicated Attacks

Floyd's collision finding algorithm (description)

• $\alpha=$ 3 (tail) and $\delta=$ 11 (cycle)





An Introduction to Hash Functions

Hanoi, April 2020

27 / 34

Outline What is a Hash Function in OOOOO

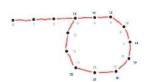
Security of Hash Functions

Generic Attacks

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

- ullet $\alpha=3$ (tail) and $\delta=11$ (cycle)
- $\bullet \ \, x_i = x_{i+k.\delta} \ \, \text{for all} \, \, i \geqslant \alpha \, \, \text{and} \, \, k \in \mathbb{N}$





Security of Hash Functions

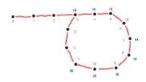
Applications

Generic Attacks

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

- ullet $\alpha=3$ (tail) and $\delta=11$ (cycle)
- ullet $x_i=x_{i+k.\delta}$ for all $i\geqslant lpha$ and $k\in \mathbb{N}$
- $x_i = x_{2i}$ whenever $i \geqslant \alpha$ and $i = k.\delta$





An Introduction to Hash Functions

Hanoi, April 2020

27 / 34

Outline What is a Hash Function
O OOOOO

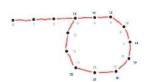
Security of Hash Function

ns Generic Attack

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

- ullet $\alpha=3$ (tail) and $\delta=11$ (cycle)
- $x_i = x_{i+k.\delta}$ for all $i \geqslant \alpha$ and $k \in \mathbb{N}$
- $x_i = x_{2i}$ whenever $i \geqslant \alpha$ and $i = k.\delta$



Floyd's algorithm (step 1)



Security of Hash Functions

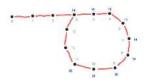
Applications

Generic Attacks

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

- ullet $\alpha=3$ (tail) and $\delta=11$ (cycle)
- ullet $x_i=x_{i+k.\delta}$ for all $i\geqslant lpha$ and $k\in \mathbb{N}$
- $x_i = x_{2i}$ whenever $i \geqslant \alpha$ and $i = k.\delta$



Floyd's algorithm (step 1)

• Start with $(a_0, b_0) \leftarrow (x_0, x_0)$



An Introduction to Hash Function

Hanoi, April 2020

27 / 34

Outline What is a Hash Function O

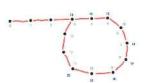
Security of Hash Function

ns Generic Attack

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

- $\alpha = 3$ (tail) and $\delta = 11$ (cycle)
- $x_i = x_{i+k.\delta}$ for all $i \geqslant \alpha$ and $k \in \mathbb{N}$
- $x_i = x_{2i}$ whenever $i \geqslant \alpha$ and $i = k.\delta$



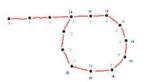
Floyd's algorithm (step 1)

- Start with $(a_0, b_0) \leftarrow (x_0, x_0)$
- ② Iteratively compute $(a_i, b_i) \leftarrow (\mathcal{H}(a_{i-1}), \mathcal{H}(\mathcal{H}(b_{i-1}))) = (x_i, x_{2i})$



Floyd's collision finding algorithm (description)

- $\alpha = 3$ (tail) and $\delta = 11$ (cycle)
- $x_i = x_{i+k.\delta}$ for all $i \geqslant \alpha$ and $k \in \mathbb{N}$
- $x_i = x_{2i}$ whenever $i \geqslant \alpha$ and $i = k.\delta$



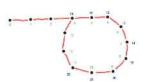
Floyd's algorithm (step 1)

- ② Iteratively compute $(a_i, b_i) \leftarrow (\mathcal{H}(a_{i-1}), \mathcal{H}(\mathcal{H}(b_{i-1}))) = (x_i, x_{2i})$
- **3** Stop whenever $a_{i_0} = b_{i_0}$ (note that $i_0 = k \cdot \delta = \left\lceil \frac{\alpha}{\delta} \right\rceil \cdot \delta$)



Floyd's collision finding algorithm (description)

- $\alpha = 3$ (tail) and $\delta = 11$ (cycle)
- $x_i = x_{i+k.\delta}$ for all $i \geqslant \alpha$ and $k \in \mathbb{N}$
- $x_i = x_{2i}$ whenever $i \geqslant \alpha$ and $i = k.\delta$



Floyd's algorithm (step 1)

- ② Iteratively compute $(a_i, b_i) \leftarrow (\mathcal{H}(a_{i-1}), \mathcal{H}(\mathcal{H}(b_{i-1}))) = (x_i, x_{2i})$
- **3** Stop whenever $a_{i_0} = b_{i_0}$ (note that $i_0 = k.\delta = \left\lceil \frac{\alpha}{\delta} \right\rceil .\delta$)

Cycle length found

A multiple $i_0 = k.\delta$ of the cycle length is obtained



Security of Hash Functions

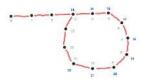
Applications

Generic Attacks

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

ullet $\alpha=3$ (tail) and $\delta=11$ (cycle)





An Introduction to Hash Functions

Hanoi, April 2020

28 / 34

Outline What is a Hash Function?

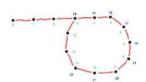
Security of Hash Functions

s Generic Attacks

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

- ullet $\alpha=3$ (tail) and $\delta=11$ (cycle)
- $x_i = x_{i+k.\delta}$ for all $i \geqslant \alpha$ and $k \in \mathbb{N}$





Security of Hash Functions

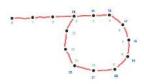
Applications

Generic Attacks

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

- ullet $\alpha=3$ (tail) and $\delta=11$ (cycle)
- $x_i = x_{i+k.\delta}$ for all $i \geqslant \alpha$ and $k \in \mathbb{N}$
- ullet x_{i_0} is known, where $i_0=k.\delta$





(ㅁ) (國) (불) (불) (원)

An Introduction to Hash Functions

Hanoi, April 2020

28 / 34

Outline What is a Hash Function in Outline Out

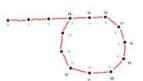
Security of Hash Functions

ns Generic Attacks

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

- ullet $\alpha=3$ (tail) and $\delta=11$ (cycle)
- ullet $x_i=x_{i+k.\delta}$ for all $i\geqslant lpha$ and $k\in \mathbb{N}$
- ullet x_{i_0} is known, where $i_0=k.\delta$



Floyd's algorithm (step 2)



Security of Hash Functions

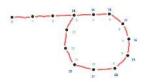
Applications

Generic Attacks

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

- $\alpha=$ 3 (tail) and $\delta=$ 11 (cycle)
- ullet $x_i=x_{i+k.\delta}$ for all $i\geqslant lpha$ and $k\in\mathbb{N}$
- ullet x_{i_0} is known, where $i_0=k.\delta$



Floyd's algorithm (step 2)

3 Start with $(c_0, d_0) \leftarrow (x_0, x_{i_0}) = (x_0, x_{k.\delta})$



An Introduction to Hash Functions

Hanoi, April 2020

28 / 34

Outline What is a Hash Function
OOOOO

Collision search

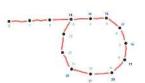
Security of Hash Functions

Generic Attacks

Some Dedicated Attacks

Floyd's collision finding algorithm (description)

- $\bullet \ \alpha = {\rm 3}$ (tail) and $\delta = {\rm 11}$ (cycle)
- ullet $x_i=x_{i+k.\delta}$ for all $i\geqslant lpha$ and $k\in \mathbb{N}$
- x_{i_0} is known, where $i_0 = k.\delta$



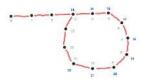
Floyd's algorithm (step 2)

- **3** Start with $(c_0, d_0) \leftarrow (x_0, x_{i_0}) = (x_0, x_{k.\delta})$
- **③** Iteratively compute (c_i, d_i) ← $(\mathcal{H}(c_{i-1}), \mathcal{H}(d_{i-1})) = (x_i, x_{i+k.\delta})$



Floyd's collision finding algorithm (description)

- $\alpha = 3$ (tail) and $\delta = 11$ (cycle)
- $x_i = x_{i+k,\delta}$ for all $i \geqslant \alpha$ and $k \in \mathbb{N}$
- ullet x_{i_0} is known, where $i_0=k.\delta$



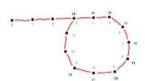
Floyd's algorithm (step 2)

- **3** Start with $(c_0, d_0) \leftarrow (x_0, x_{i_0}) = (x_0, x_{k.\delta})$
- **⑤** Iteratively compute (c_i, d_i) ← $(\mathcal{H}(c_{i-1}), \mathcal{H}(d_{i-1})) = (x_i, x_{i+k.\delta})$
- **5** Stop whenever $c_{i_1} = d_{i_1}$ (note that $i_1 = \alpha$)



Floyd's collision finding algorithm (description)

- $\alpha = 3$ (tail) and $\delta = 11$ (cycle)
- $x_i = x_{i+k,\delta}$ for all $i \geqslant \alpha$ and $k \in \mathbb{N}$
- x_{i_0} is known, where $i_0 = k.\delta$



Floyd's algorithm (step 2)

- **9** Start with $(c_0, d_0) \leftarrow (x_0, x_{i_0}) = (x_0, x_{k.\delta})$
- **⑤** Iteratively compute (c_i, d_i) ← $(\mathcal{H}(c_{i-1}), \mathcal{H}(d_{i-1})) = (x_i, x_{i+k.\delta})$
- **⑤** Stop whenever $c_{i_1} = d_{i_1}$ (note that $i_1 = \alpha$)

Collision found!

The collision is given by $\mathcal{H}(x_{\alpha-1}) = \mathcal{H}(x_{\alpha-1+k\delta})$ with $x_{\alpha-1} \neq x_{\alpha-1+k\delta}$



Floyd's collision finding algorithm (complexity)

 \bullet This algorithm finds a collision in 3 $\left\lceil\frac{\alpha}{\delta}\right\rceil$ $.\delta+2\alpha$ hash evaluations



(D > 4 B > 4 분 > 4 분 > - 분 - 쒸익()

An Introduction to Hash Functions

Hanoi, April 2020

29 / 34

Outline What is a Hash Function? Security of Hash Functions Applications **Generic Attacks** Some I

O ○○○○○
Collision search

Floyd's collision finding algorithm (complexity)

- \bullet This algorithm finds a collision in 3 $\left\lceil\frac{\alpha}{\delta}\right\rceil.\delta + 2\alpha$ hash evaluations
- ullet As both lpha and δ are $\mathcal{O}(\sqrt{2^n})$, so is Floyd's algorithm time complexity



Floyd's collision finding algorithm (complexity)

- This algorithm finds a collision in $3\left\lceil\frac{\alpha}{\delta}\right\rceil.\delta+2\alpha$ hash evaluations
- As both α and δ are $\mathcal{O}(\sqrt{2^n})$, so is Floyd's algorithm time complexity
- Memory requirement is negligible



Some Dedicated Attacks

- - Definition and properties
 - Examples of hash functions
 - How does it work ?
- Security of Hash Functions
 - Security requirements
 - Security considerations
 - Complexity figures
- - Secured password storage
 - Data integrity
 - Entity authentication
 - Message authentication
 - Digital signature

- - Birthday paradox
 - Collision search
- Some Dedicated Attacks
 - A burst of new attacks
 - The SHA-3 competition





Outline What is a Hash Function? Security of Hash Functions Applications Generic Attacks

O 0000 0000 00000 000000000 00000

What's happened ?

• Weaknesses were known on MD4, MD5, RIPEMD since the nineties...



4□ > 4□ > 4□ > 4□ > 4□ > 4□

An Introduction to Hash Functions

Hanoi, April 2020

31 / 34

Outline What is a Hash Function ?
O OOOOO

Security of Hash Function:

Generic Attack

Some Dedicated Attacks

What's happened ?

- Weaknesses were known on MD4, MD5, RIPEMD since the nineties...
 - \bullet but they did not lead to practical attacks



What's happened?

- Weaknesses were known on MD4, MD5, RIPEMD since the nineties. . .
 - ...but they did not lead to practical attacks
- In summer 2004, a team of chinese researchers announced new collision attacks on MD4, MD5, RIPEMD and HAVAL-128



An Introduction to Hash Functions

Hanoi, April 2020

31 / 34

Outline What is a Hash Function 7

Security of Hash Functions

plications Gene

eneric Attacks

Some Dedicated Attack

What's happened?

- Weaknesses were known on MD4, MD5, RIPEMD since the nineties...
 - ...but they did not lead to practical attacks
- In summer 2004, a team of chinese researchers announced new collision attacks on MD4, MD5, RIPEMD and HAVAL-128
- \bullet The crypto community was stunned \dots



What's happened?

- Weaknesses were known on MD4, MD5, RIPEMD since the nineties. . .
 - ...but they did not lead to practical attacks
- In summer 2004, a team of chinese researchers announced new collision attacks on MD4, MD5, RIPEMD and HAVAL-128
- The crypto community was stunned ...
- Extending some previously used collision search techniques, they were able to compute new collisions in a matter of minutes, even seconds



An Introduction to Hash Functions

Hanoi, April 2020

31 / 34

Outline What is a Hash Function ?
OOOOO

A burst of new attacks

Security of Hash Function

ions Generic Atta

Some Dedicated Attack

Results so far . . .

• Collisions can be found on MD4 with only 3 hash function computations!



Results so far ...

- Collisions can be found on MD4 with only 3 hash function computations!
- Collisions can be found on MD5 using 2-block messages (i.e. 1024 bits)



◆ロト ◆部 ト ◆ 恵 ト ◆ 恵 ・ 夕 へ ○ ○

An Introduction to Hash Functions

Hanoi, April 2020

32 / 34

Outline What is a Hash Function ?

Security of Hash Functions

OOOOOO

Generic Attacks

Some Dedicated Attacks ○○●○○

- Collisions can be found on MD4 with only 3 hash function computations!
- Collisions can be found on MD5 using 2-block messages (i.e. 1024 bits)
 - Complexity 2³⁶ for the first block



Results so far . . .

- Collisions can be found on MD4 with only 3 hash function computations!
- Collisions can be found on MD5 using 2-block messages (i.e. 1024 bits)
 - Complexity 2³⁶ for the first block
 - ullet Complexity as low as 2^{27} for the second block



<ロ > ←回 > ←回 > ← 直 > ← 直 > への

An Introduction to Hash Functions

Hanoi, April 2020

32 / 34

Outline What is a Hash Function
O OOOOO

Security of Hash Function

itions Generic Attac

Some Dedicated Attack

- Collisions can be found on MD4 with only 3 hash function computations!
- Collisions can be found on MD5 using 2-block messages (i.e. 1024 bits)
 - Complexity 2³⁶ for the first block
 - \bullet Complexity as low as 2^{27} for the second block
 - \bullet The second step takes 30 minutes on a regular PC



Results so far . . .

- Collisions can be found on MD4 with only 3 hash function computations!
- Collisions can be found on MD5 using 2-block messages (i.e. 1024 bits)
 - Complexity 2³⁶ for the first block
 - ullet Complexity as low as 2^{27} for the second block
 - \bullet The second step takes 30 minutes on a regular PC
- Collisions are now found on SHA-0 in 2³⁹ computations



4□ > 4回 > 4 回 > 4 回 > 4 回 > 1 回 > 1 回 > 1 回 > 1 回 > 1 回 < 1 回 > 1

An Introduction to Hash Functions

Hanoi, April 2020

32 / 34

Outline What is a Hash Function 7

Security of Hash Functions

oplications Generic Atta

Some Dedicated Attack

- Collisions can be found on MD4 with only 3 hash function computations!
- Collisions can be found on MD5 using 2-block messages (i.e. 1024 bits)
 - Complexity 2³⁶ for the first block
 - \bullet Complexity as low as 2^{27} for the second block
 - The second step takes 30 minutes on a regular PC
- ullet Collisions are now found on SHA-0 in 2^{39} computations
- Collisions were estimated on SHA-1 in about 2⁶⁹ computations



Results so far ...

- Collisions can be found on MD4 with only 3 hash function computations!
- Collisions can be found on MD5 using 2-block messages (i.e. 1024 bits)
 - Complexity 2³⁶ for the first block
 - Complexity as low as 2²⁷ for the second block
 - The second step takes 30 minutes on a regular PC
- Collisions are now found on SHA-0 in 2³⁹ computations
- Collisions were estimated on SHA-1 in about 2⁶⁹ computations
 - Crypto 2005 rump session: (theoretical) collisions in 2⁶³ computations



◆ロト ◆部 ト ◆ 恵 ト ◆ 恵 ・ 夕 へ ○ ○

An Introduction to Hash Functions

Hanoi, April 2020

32 / 34

Outline What is a Hash Function
O OOOOO

Security of Hash Functions

olications Generic Attac

Some Dedicated Attac

- Collisions can be found on MD4 with only 3 hash function computations!
- Collisions can be found on MD5 using 2-block messages (i.e. 1024 bits)
 - Complexity 2³⁶ for the first block
 - Complexity as low as 2^{27} for the second block
 - The second step takes 30 minutes on a regular PC
- Collisions are now found on SHA-0 in 2³⁹ computations
- Collisions were estimated on SHA-1 in about 2⁶⁹ computations
 - \bullet Crypto 2005 rump session: (theoretical) collisions in 2^{63} computations
 - Eurocrypt 2009 rump session: (theoretical) collisions in 2⁵² computations!



Results so far ...

- Collisions can be found on MD4 with only 3 hash function computations!
- Collisions can be found on MD5 using 2-block messages (i.e. 1024 bits)
 - Complexity 2³⁶ for the first block
 - Complexity as low as 2²⁷ for the second block
 - The second step takes 30 minutes on a regular PC
- Collisions are now found on SHA-0 in 2³⁹ computations
- ullet Collisions were estimated on SHA-1 in about 2^{69} computations
 - Crypto 2005 rump session: (theoretical) collisions in 2⁶³ computations
 - Eurocrypt 2009 rump session: (theoretical) collisions in 2⁵² computations!
 - A collision on SHA-1 has been found by Google only on February 2017



4 □ ト 4 圖 ト 4 圖 ト 4 圖 ● 9 0 0 0

An Introduction to Hash Functions

Hanoi, April 2020

32 / 34

Ovalina What is a black Forestina 2

Security of Hash Functions

plications Generic Att

Some Dedicated Attac

- Collisions can be found on MD4 with only 3 hash function computations!
- Collisions can be found on MD5 using 2-block messages (i.e. 1024 bits)
 - Complexity 2³⁶ for the first block
 - Complexity as low as 2^{27} for the second block
 - The second step takes 30 minutes on a regular PC
- Collisions are now found on SHA-0 in 2³⁹ computations
- Collisions were estimated on SHA-1 in about 2⁶⁹ computations
 - Crypto 2005 rump session: (theoretical) collisions in 2⁶³ computations
 - Eurocrypt 2009 rump session: (theoretical) collisions in 2⁵² computations!
 - A collision on SHA-1 has been found by Google only on February 2017
- Nothing announced (yet) on recent SHA-2 hash functions family



An open competition has been launched by the NIST on November 2, 2007 to define a new hash function standard SHA-3 (instead of SHA-1 and SHA-2)



Toward a new standard...

An open competition has been launched by the NIST on November 2, 2007 to define a new hash function standard SHA-3 $\,$ (instead of SHA-1 and SHA-2)

• Fall 2008, 51 submissions were accepted by NIST to Round 1 (of which 5 from France)



An open competition has been launched by the NIST on November 2, 2007 to define a new hash function standard SHA-3 (instead of SHA-1 and SHA-2)

- Fall 2008, 51 submissions were accepted by NIST to Round 1 (of which 5 from France)
- 14 candidates accepted to Round 2 on July 24, 2009:



Toward a new standard...

An open competition has been launched by the NIST on November 2, 2007 to define a new hash function standard SHA-3 $\,$ (instead of SHA-1 and SHA-2)

- Fall 2008, 51 submissions were accepted by NIST to Round 1 (of which 5 from France)
- 14 candidates accepted to Round 2 on July 24, 2009:
 - BLAKE, Blue Midnight Wish, CubeHash, ECHO, Fugue, Grøstl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD, Skein



An open competition has been launched by the NIST on November 2, 2007 to define a new hash function standard SHA-3 (instead of SHA-1 and SHA-2)

- Fall 2008, 51 submissions were accepted by NIST to Round 1 (of which 5 from France)
- 14 candidates accepted to Round 2 on July 24, 2009:
 - BLAKE, Blue Midnight Wish, CubeHash, ECHO, Fugue, Grøstl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD, Skein
- 5 finalist candidates accepted to Round 3 on December 9, 2010:



< ロ > ← □ > ← 重 > ← 重 > ・ 重 ・ 少 Q

An Introduction to Hash Functions

Hanoi, April 2020

33 / 34

Outline What is a Hash Function

O OOOO

The SHA-3 competition

Security of Hash Functions

ations Generic Attack

Some Dedicated Attack

Toward a new standard...

An open competition has been launched by the NIST on November 2, 2007 to define a new hash function standard SHA-3 $\,$ (instead of SHA-1 and SHA-2)

- Fall 2008, 51 submissions were accepted by NIST to Round 1 (of which 5 from France)
- 14 candidates accepted to Round 2 on July 24, 2009:
 - BLAKE, Blue Midnight Wish, CubeHash, ECHO, Fugue, Grøstl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD, Skein
- 5 finalist candidates accepted to Round 3 on December 9, 2010:
 - BLAKE, Grøstl, JH, Keccak, Skein



An open competition has been launched by the NIST on November 2, 2007 to define a new hash function standard SHA-3 (instead of SHA-1 and SHA-2)

- Fall 2008, 51 submissions were accepted by NIST to Round 1 (of which 5 from France)
- 14 candidates accepted to Round 2 on July 24, 2009:
 - BLAKE, Blue Midnight Wish, CubeHash, ECHO, Fugue, Grøstl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD, Skein
- 5 finalist candidates accepted to Round 3 on December 9, 2010:
 - BLAKE, Grøstl, JH, Keccak, Skein
- Proclamation of the winner on October 2, 2012:

(standardized on August 5, 2015)

KECCAK



An Introduction to Hash Functions

Christophe Clavier - Florent Bruguier

University of Limoges - University of Montpellier

