



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY

# Computer Security and Cryptography

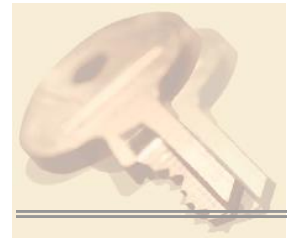
## CS381

来学嘉

计算机科学与工程系 电院3-423室

34205440 1356 4100825 laix@sjtu.edu.cn

2015-05



# Organization



- Week 1 to week 16 (2015-03 to 2014-06)
- 东中院-3-102
- Monday 3-4节; week 9-16
- Wednesday 3-4节; week 1-16
- lecture 10 + exercise 40 + **random tests** 40 + other 10
- Ask questions in class – counted as points
- Turn ON your mobile phone (after lecture)
- Slides and papers:
  - <http://202.120.38.185/CS381>
    - computer-security
  - <http://202.120.38.185/references>
- TA: Geshi Huang [gracehgs@mail.sjtu.edu.cn](mailto:gracehgs@mail.sjtu.edu.cn)
- Send homework to the TA

**Rule: do the homework on your own!**



# Contents



- **Introduction** -- What is security?
- **Cryptography**
  - Classical ciphers
  - Today's ciphers
  - Public-key cryptography
  - **Hash functions and MAC**
  - Authentication protocols
- **Applications**
  - Digital certificates
  - Secure email
  - Internet security, e-banking
- **Computer and network security**
  - Access control
  - Malware
  - Firewall



# Content



- Hash function – usage and basic properties
- Iterated hash function – Relationship between Hash function and its round (compress) function
- Real compressing functions
  - Using block cipher
  - Dedicated hash functions, MD5, SHA1
- Security and attacks
- SHA-3
- MAC



# References

- Bart Preneel, The State of Cryptographic Hash Functions, <http://www.cosic.esat.kuleuven.ac.be/publications/>
- G. Yuval, "How to swindle Rabin," Cryptologia, Vol. 3, 1979, pp. 187-189
- Ralph Merkle. One way Hash functions and DES. In Gilles Brassard, editor, Advances in Cryptology: CRYPTO 89, LNCS 435. Springer-Verlag. 1989: 428–446.
- Ivan Damgard. A design principle for Hash functions. In Gilles Brassard, editor, Advances in Cryptology: CRYPTO 89, LNCS 435. Springer-Verlag. 1989:416~427.
- ISO/IEC 10118, Information technology - Security techniques - Hash-functions,
  - Part 1: General",
  - Part 2: Hash-functions using an n-bit block cipher algorithm,"
  - Part 3: Dedicated hash-functions,"
  - Part 4: Hash-functions using modular arithmetic,"
- M. Naor, M. Yung, "Universal one-way hash functions and their cryptographic applications," Proc. 21st ACM Symposium on the Theory of Computing, 1990, pp. 387-394.
- X. Lai, J.L. Massey, "Hash functions based on block ciphers," Advances in Cryptology, Proceedings Eurocrypt'92, LNCS 658, R.A. Rueppel, Ed., Springer-Verlag, 1993, pp. 55-70
- L.R. Knudsen, X. Lai, B. Preneel, "Attacks on fast double block length hash functions," Journal of Cryptology, Vol. 11, No. 1, Winter 1998, pp. 59-72.



# References



- Joux, “Multicollisions in Iterated Hash Functions. Applications to Cascaded Constructions,” *Crypto 2004 Proceedings*, Springer-Verlag, 2004.
- John Kelsey and Bruce Schneier Second Preimages on n-bit Hash Functions for Much Less than  $2^n$  Work, Eurocrypt 2005,
- Ronald Rivest. The MD4 Message Digest Algorithm. RFC1320, <http://rfc.net/rfc1320.html>. April 1992.
- Ronald Rivest. The MD5 Message Digest Algorithm. RFC1321, <http://rfc.net/rfc1321.html>. April 1992.
- Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. RIPEMD-160: A Strengthened Version of RIPEMD. In Dieter Gollmann, editor, Fast Software Encryption, Cambridge, UK, Proceedings, LNCS-1039. Springer.1996: 71~82.
- NIST. Secure Hash standard. Federal Information Processing Standard. FIPS-180-1. April 1995
- Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. Cryptology ePrint Archive, Report 2004/199, 2004. <http://eprint.iacr.org/2004/199.pdf>
- Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu. Crypt-analysis of the Hash Functions MD4 and RIPEMD, Advances in Cryptology – EUROCRYPT 2005, LNCS-3494. Springer.2005: 1~18..
- NIST Selects Winner of Secure Hash Algorithm (SHA-3) Competition". NIST. 2012-10-02.
- G Bertoni,et al, Sponge functions, ECRYPT hash workshop, 2007



# One-way functions



- **Oneway function**  $f: X \rightarrow Y$ , given  $x$ , easy to compute  $f(x)$ ; but for given  $y$  in  $f(X)$ , it is hard to find  $x$ , s.t.,  $f(x)=y$ .
  - $\text{Prob}[f(A(f(x)))=f(x)] < 1/p(n)$  (TM definition, existence unknown)
  - Example: hash function, discrete logarithm;
- **Keyed function**  $f(X,Z)=Y$ , for known key  $z$ , it is easy to compute  $f(.,z)$ 
  - Block cipher
- **Keyed oneway function**:  $f(X,Z)=Y$ , for known key  $z$ , it is easy to compute  $f(.,z)$  but for given  $y$ , it is hard to  $x,z$ , s.t.,  $f(x,z)=y$ .
  - MAC function: keyed hash  $h(z,X)$ , block cipher CBC
- **Trapdoor oneway function**  $f_T(x)$ : easy to compute and hard to invert, but with additional knowledge  $T$ , it is easy to invert.
  - Public-key cipher; RSA:  $y=x^e \bmod N$ ,  $T: N=p*q$



# Hash function and applications



**Definition.** A **hash function** is an efficiently computable and publicly known function that maps the set of all arbitrarily long binary sequences (**message**) to the set of a binary sequence (**hash code/digest**) of some fixed length

## Applications

- **Modification Detection Code MDC**

$M \rightarrow \text{Hash} \rightarrow H$

$M' \rightarrow \boxed{\text{Hash}} \rightarrow H' \neq H$

- **Digital signatures**

$M \rightarrow \text{Hash} \rightarrow H \rightarrow S(H)$ ,  $S(H)$  is the signature of message  $M$

$M' \rightarrow \boxed{\text{Hash}} \rightarrow H' \neq H$ , if yes,  $S(H)$  is also a valid signature of  $M'$

attack: for any  $M$  with signature  $S(H)$ , find another  $M'$  s.t.  
 $\text{Hash}(M) = \text{Hash}(M')$ .

**Requirement:** one-way and collision-free





# Random oracle and hash function



- A **random oracle (RO)** is an “idealized function” that on any input (query) it answers (produces as output) a random string in a consistent manner:
  - If  $x$  is “new”, then the answer  $y$  is a uniform random variable;
  - If  $x$  has been asked before, then the answer  $y$  is remain same.
- RO represents a random function over which an adversary has no control
- ROM (**random oracle model**): a framework for provable security, in which both the protocol designer and the adversary can have access to ROs.
- In ROM, security proof is easier than in standard model (without RO).
- In a system that is proved secure in ROM, we replace the RO with a **hash function**, and hope that security remains.
- **This approach is widely used.**
- Limitations: RO can not be realized by any efficient algorithm (we can only assume that a hash function is a RO)

There exists counterexample crypto-systems that are secure in ROM but breakable when RO is replaced by any hash function.



# Modification Detection



- Modification Detection Code MDC

$$M \rightarrow \text{Hash} \rightarrow H$$

$$M' \rightarrow \text{Hash} \rightarrow H' \stackrel{?}{=} H$$

- To provide integrity:
  - Store  $H = \text{Hash}(M)$  securely. Check  $H' = \text{Hash}(M') \stackrel{?}{=} H$
  - Example. Simple protection of web-site:
    - compute hash code  $H$ , backup the site.
    - Check hash code  $H'$  of website regularly, if  $H' \neq H$ , replace the website with the backup copy.
- Attack: to find a  $M' \neq M$  such that  $\text{Hash}(M') = \text{Hash}(M)$ .
- Requirement: second preimage resistant (one-wayness)



# Digital signatures



- **Digital signatures**

To sign a message  $M$ , first hash message  $M$ :  $H = \text{Hash}(M)$ , then apply the signature function on  $H$ :  $S_x(H)$  is the (user  $x$ 's) signature of message  $M$ .

Reason:

**Performance:** Only need sign a short hash-code instead of a long message.

**Security:** Signature needs redundancy for security. Simple redundancy scheme appears not secure (example: ISO 9796-1). Signature scheme with “provable security” all use hash.

- Collision attack: find  $M' \neq M$ , but  $H' = H$ ; then signature of  $M$  is the same as the signature of  $M'$ . In the real attack, sign on message  $M$ , but forge signature on  $M'$ , i.e.,  $(M', S_x(H')) = S_x(H)$
- **Requirement: collision-free**



# Security of hash function



- **Second preimage resistance (target collision resistance)**
  - Given,  $M$  and  $H = \text{Hash}(M)$ , it is infeasible to find  $M' \neq M$ ,  $\text{Hash}(M') = \text{Hash}(M)$ .
- **Collision resistance** (collision resistance)
  - It is infeasible to find distinct  $M', M$ ,  $\text{Hash}(M') = \text{Hash}(M)$ .
- Second pre-image and collision always exist! The hope is to make it computationally *infeasible*
- *Note*: collision resistance implies second preimage resistance.
- If hash code length is  $m$ -bit, then:
  - To find second pre-image needs at most  $2^m$  computations of  $\text{Hash}$
  - To find collision needs at most  $2^{m/2}$  computations of  $\text{Hash}$



# Birthday paradox



- 23 people in a room, it is likely that there exists at least one collision (two or more persons are of the same birthday)

**Theorem 1.** Randomly chose  $N^{1/2}$  elements from a set containing  $N$  elements, then

$p$ =probability( 2 or more selections are the same)  $\geq 1/2$

Proof. Randomly chose  $m$  elements from a set containing  $N$  elements, the probability  $m$  elements are **all different** is

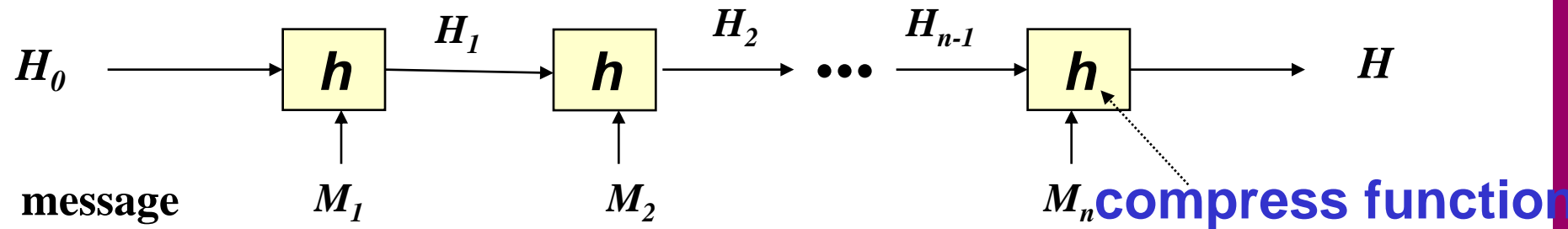
$$P = \frac{(N-1)}{N} \frac{(N-2)}{N} \dots \frac{(N-(m-1))}{N} \approx e^{-m(m-1)/2N}$$

$$e = 2.71828$$

**For  $m = 1.2 * N^{1/2}$ ,  $p = 1 - P \approx 1 - e^{-1.4/2} = 0.5$**



# Iterated Hash function



**compressing (round) function**  $h: \{0,1\}^m \times \{0,1\}^l \rightarrow \{0,1\}^m$

**initial value**  $H_0$  (m-bit)

**message**  $M=(M_1, \dots, M_n)$ ,  $M_i$  are  $l$ -bit blocks

**Hash code**  $H=Hash(H_0, M)$

$H_i = h(H_{i-1}, M_i)$   $i=1, 2, \dots, n$  (chaining value, an  $m$ -bit block)

$H=H_n$



# Attacks



**Target attack** (2nd pre-image attack):

Given  $H_0$  and  $M$ , find  $M' \neq M$ , but  $\text{Hash}(H_0, M) = \text{Hash}(H_0, M')$

**Free-start target attack** (2nd pre-image attack with arbitrary IV):

Given  $(H_0, M)$ , find  $(H_0', M') \neq (H_0, M)$ , s.t.

$$\text{Hash}(H_0, M) = \text{Hash}(H_0', M')$$

**Chosen-message target attack:** For given  $H_0$ , specify a set  $C$ , such that for each  $M$  in  $C$ , there is an  $M' \neq M$ , s.t.

$$\text{Hash}(H_0, M) = \text{Hash}(H_0, M')$$

**Collision attack:** Given  $H_0$ , find  $M$  and  $M' \neq M$ , s.t.

$$\text{Hash}(H_0, M) = \text{Hash}(H_0, M')$$

**Semi free-start collision attack:** Find  $H_0, M, M' \neq M$ , s.t.

$$\text{Hash}(H_0, M) = \text{Hash}(H_0, M')$$

**Free-start collision attack:** Find  $(H_0, M)$  and

$$(H_0', M') \neq (H_0, M), \text{ but } \text{Hash}(H_0, M) = \text{Hash}(H_0', M')$$

- Target attack  $\rightarrow$  collision attack
- Secure Hash against free-start attacks is also secure against 'usual' attacks



# Why so many attacks? – MD5

- Boer & Bosselaers [93]: free-start collision (pseudo collision: same message, different IV )

**Free-start collision attack:** Find  $(H_0, M)$  and  $(H_0', M') \neq (H_0, M)$ , but  $\text{Hash}(H_0, M) = \text{Hash}(H_0', M')$

- Dobbertin [96]: semi free-start collisions ( different message, chosen IV)

**Semi free-start collision attack:** Find  $H_0, M, M' \neq M$ , but  $\text{Hash}(H_0, M) = \text{Hash}(H_0, M')$

- Wang et.al [2004]:

**Collision attack:** Given  $H_0$ , find  $M$  and  $M' \neq M$ , but  $\text{Hash}(H_0, M) = \text{Hash}(H_0, M')$

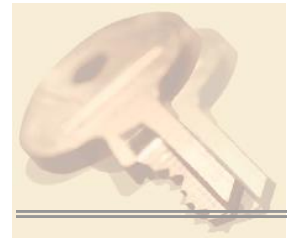




# Complexity of attacks on Hash



- Brute-force target attacks require about  $2^m$  computations of  $h$
- Brute-force collision attacks require about  $2^{m/2}$  computations of  $h$
- Complexity :  $C_{FS\text{-}target} \leq C_{target} \leq 2^m$
- $C_{FS\text{-}collision} \leq C_{semi\text{ }FS\text{-}collision} \leq C_{collision} \leq 2^{m/2}$
- An attack on  $h$  implies an attack on *Hash* of same type
  - The converse is not true, *Hash* ('chain') can be weaker than  $h$  ('link')



# Attacks on Hash

## Trivial free-start attack

$$\text{Hash}(H_0, M_1, M_2) = \text{Hash}(H_1, M_2)$$

## Trivial semi free-start attack [Miyaguchi et al 90]

if  $h$  has a fixed-point  $h(H, M) = H$ , then

$$H = \text{Hash}(H, M) = \text{Hash}(H, M, M) = \text{Hash}(H, M, M, M) = \dots$$

## Long-message target attack [Winternitz 84]:

If the given message has  $n$  blocks, then

$$C_{\text{target}}(\text{Hash}) \leq 2 \times 2^{m/n} \quad \text{for } n \leq 2^{m/2}$$

$$C_{\text{target}}(\text{Hash}) \leq 2 \times 2^{m/2} \quad \text{for } n > 2^{m/2},$$

# Long message attack



Long-message target attack [Winternitz 84]:

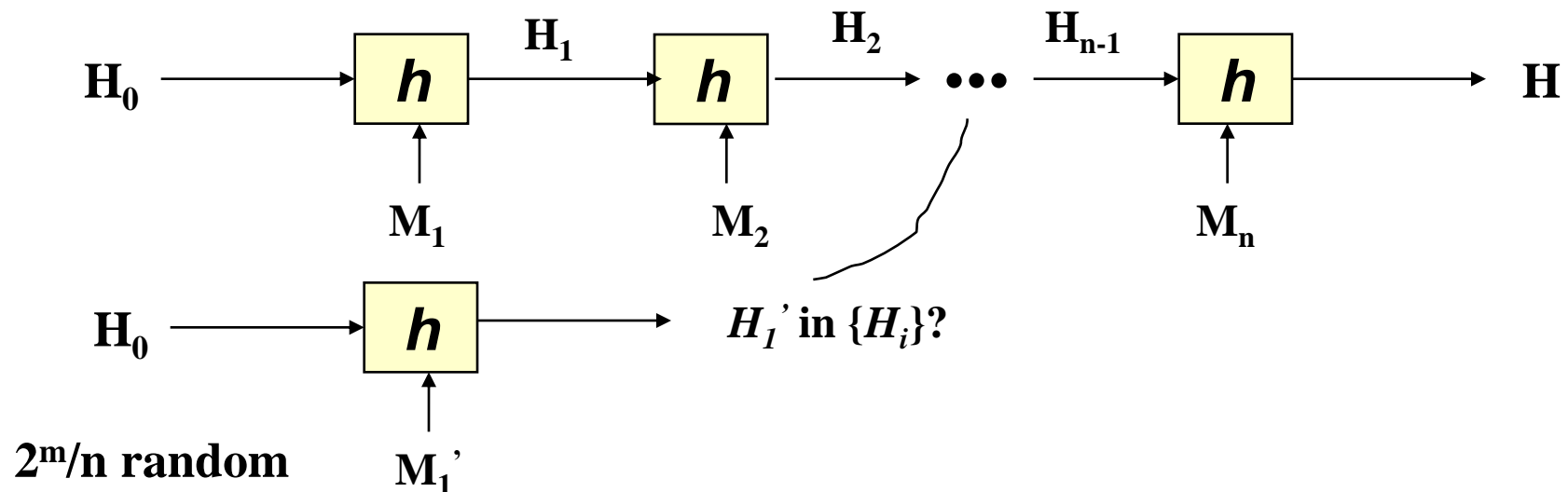
$$C_{\text{target}}(\text{Hash}) \leq 2 \times 2^{m/n} \quad \text{for } n \leq 2^{m/2}$$

$$C_{\text{target}}(\text{Hash}) \leq 2 \times 2^{m/2} \quad \text{for } n > 2^{m/2},$$

For  $2^{m/n}$  random  $M_1'$ , compute  $H_1' = h(H_0, M_1')$

$\Pr[\text{some } H_1' = \text{some } H_i] \sim 0.63$ , for such  $H_1'$  and  $H_i$

$$\text{Hash}(H_0, \mathbf{M}_1', \mathbf{M}_{i+1}, \dots, \mathbf{M}_n) = \text{Hash}(H_0, \mathbf{M}_1, \dots, \mathbf{M}_i, \mathbf{M}_{i+1}, \dots, \mathbf{M}_n)$$





# MD-strengthening



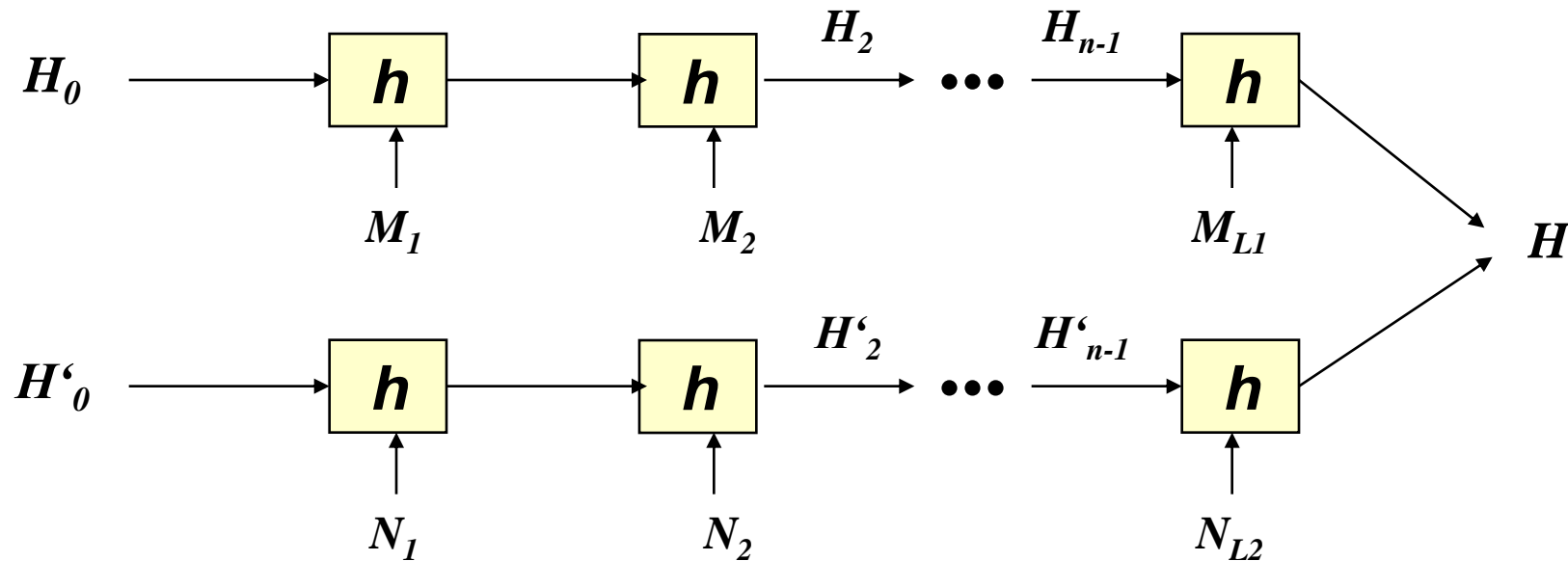
- Taking advantage that  $M'$  can have different length from  $M$ , one can break *Hash* without breaking  $h$ .
- Merkle-Damgaard strengthening:  
Let the last block  $M_n$  be the length of the actual message in bits.
- Th.2 Against free-start collision attack,  $Hash_{MD}$  is as secure as  $h$  [Merkle C89, Damgaard C89, Naor-Yung 89]



# Free-start Collision attack:



- Free-start collision attack on  $Hash_{MD}$  implies free-start collision on  $h$ . (*inverse is obvious*)
- *Proof: exists  $i, j: H_i \neq H'_i, H_{i+1} = H'_{j+1}$*



- Collision attack on  $Hash_{MD}$  implies free-start collision on  $h$ . (*inverse is unknown*)



# Target attack when $h$ is not one-way

(meet-in-the-middle target attack by working backwards)

- *Th.3*  $C_{\text{target}}(\text{Hash}_{\text{MD}}) \leq 2^{m/2} C_{\text{FS-target}}(h)^{1/2}$ 
  - If obtaining **random inverse** of  $h$  needs  $2^s$  computations, then target attack on  $\text{Hash}_{\text{MD}}(..)$  needs at most  $2^{(m+s)/2}$  [Lai-Massey 92]

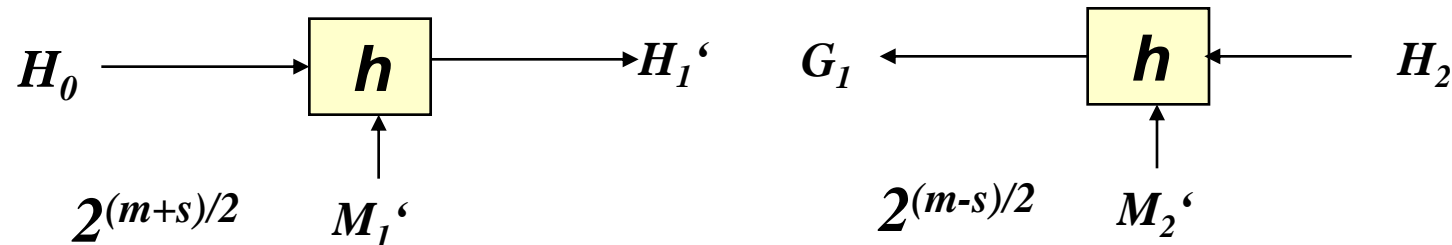
*Attack: given  $\text{Hash}_{\text{MD}}(H_0, M_1, M_2, M_3, ..)$ , i.e, given  $H_2$ , compute*

*forwards  $2^{(m+s)/2}$  values of  $H_1'$ , backwards  $2^{(m-s)/2}$  values of  $G_1$*

$$\text{Pr}[\text{some } H_1' = \text{some } G_1] = 1 - [(1 - 2^{-m})^{(m-s)/2}]^{2^{(m+s)/2}} = 1 - (1 - 2^{-m})^m = 0.63$$

*then, for such  $M_1', M_2'$ ,*

$$\text{Hash}_{\text{MD}}(H_0, M_1', M_2', M_3, ..) = \text{Hash}_{\text{MD}}(H_0, M_1, M_2, M_3, ..)$$





# Meet-in-the-middle



- Randomly choose  $A=\{x_1, \dots, x_X\}$ ,  $B=\{y_1, \dots, y_Y\}$  from a set  $S$  with  $N$  elements.
- Probability that **some**  $x_i = \text{some } y_j$  is  $1 - (1 - Y/N)^X$

$$Pr(x_i \neq y_j)$$

$$= Pr(x_1 \notin \{y_j\})p(x_2 \notin \{y_j\}) \dots p(x_X \notin \{y_j\}) = ((N-Y)/N)^X = (1-Y/N)^X$$

Theorem.  $A, B \subset S$ . if  $|A| |B| \cong |S|$ , then

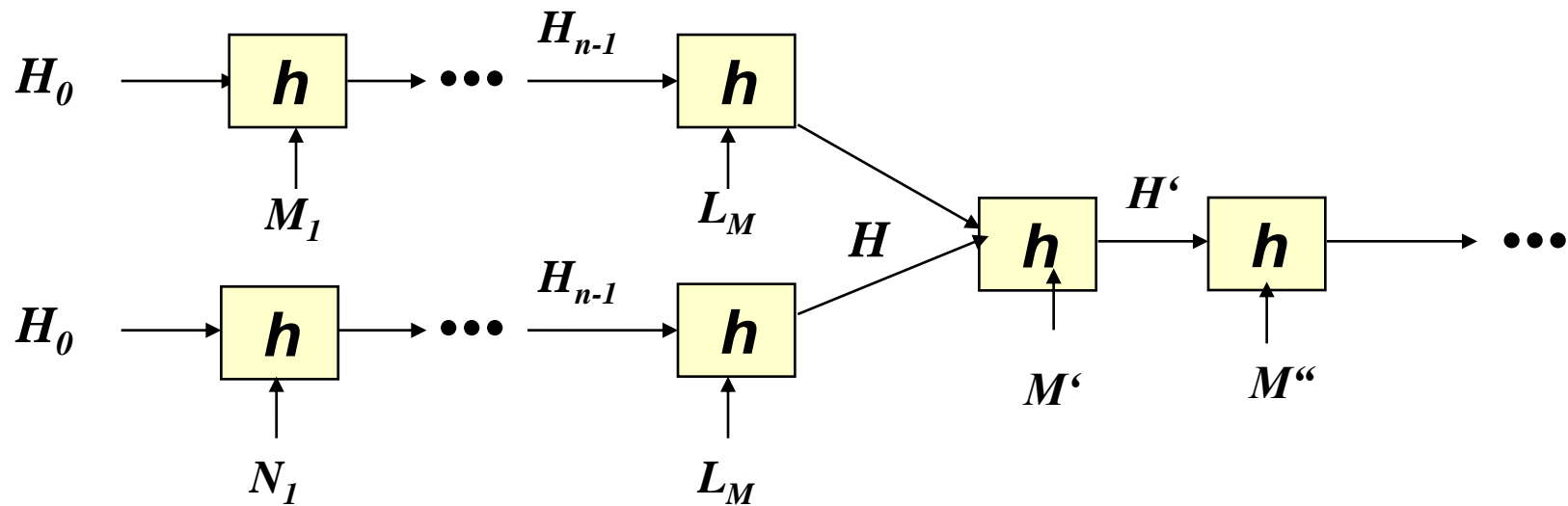
$$P(A \cap B \neq \emptyset) \cong 1 - e^{-1} = 0.63 \quad e = 2.71828$$

*This fact has been used in many new attacks on ciphers and hash functions*

# The issue with MD construction



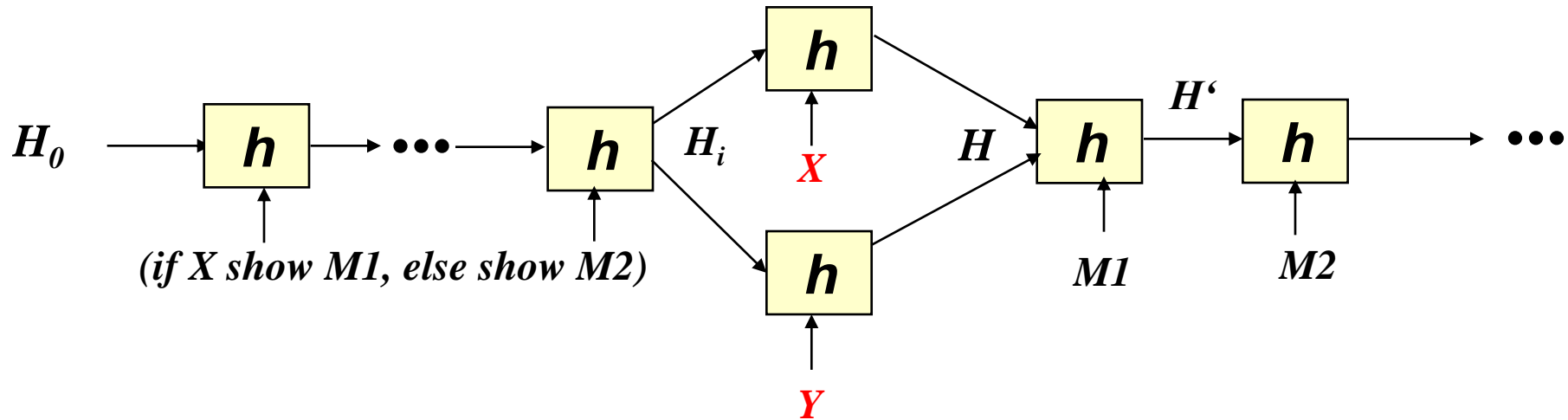
- One collision(2<sup>nd</sup>-preimage) implies arbitrarily many collision(2<sup>nd</sup>-preimage)



- The impact:
  - “random” collision  $\Rightarrow$  “useful/harmful” collision
  - Provable security in Random Oracle model may not hold when replace RO with Hash.



# Document collision with MD5



- Fixed  $H_0$ , select prefix message, from the resulting  $H_i$ , find colliding messages  $X$ ,  $Y$ ; then attach  $(M1, M2)$ .
  - (instruction,  $X$ ,  $M1$ ,  $M2$ )
  - (instruction,  $Y$ ,  $M1$ ,  $M2$ )
  - Have same hash code (signature)
- Stefan Lucks and Magnus Daum (Eurocrypt'05 Rump Session)



# $Hash_{MD}$ - compress

- Free-start collision attacks:  $Hash_{MD}$  is as secure as  $h$
- Collision attack: collision of  $Hash_{MD}$  implies free-start collision of  $h$ . (*inverse is unknown*)
- Free-start target attack on  $h$  implies Target attack on  $Hash_{MD}$
- Target attack:  $Hash_{MD}$  cannot achieve ideal security (  $C < 2^m$  )
- **Goal:** *find secure  $h$  against free-start collision attack* (target attack is harder than collision)
- **open:** how to design a hash function that is secure against target attack and without the undesirable properties?
  - Prefix-free, DME, chop, ROX,..., (next standard?)



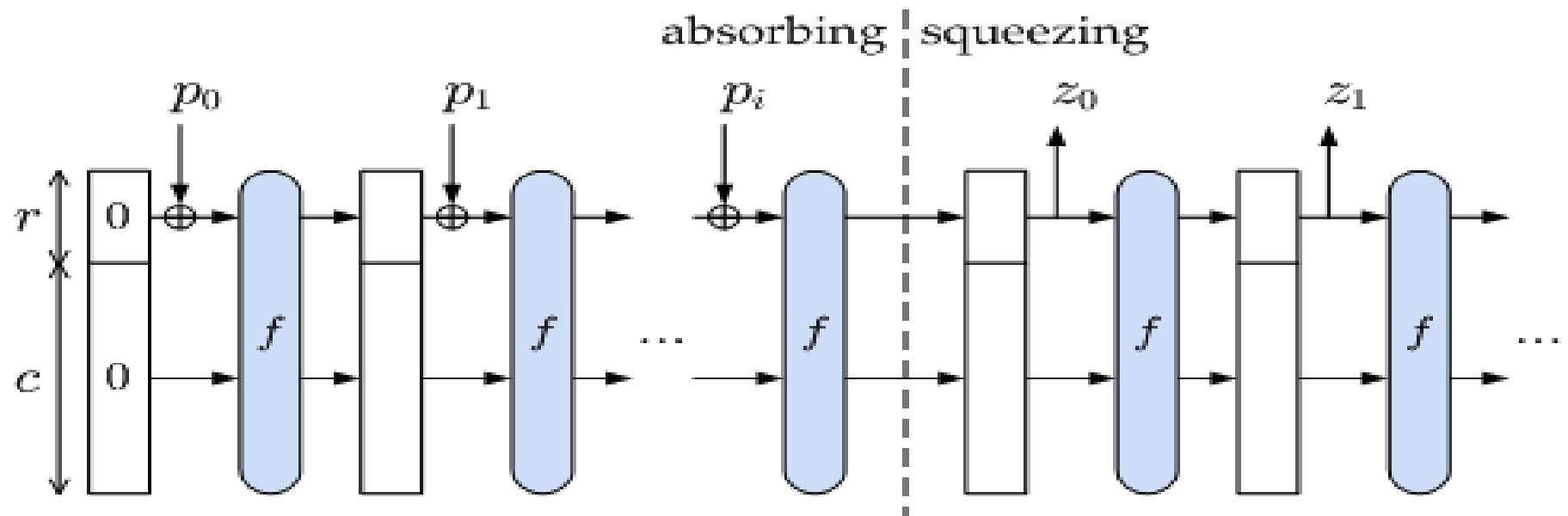
# Compress functions



- Design a cryptographic hash function reduces to finding a oneway **compressing function** from  $\{0,1\}^{m+l}$  to  $\{0,1\}^m$ , where
  - The output (hash-code) size  $m$  is for security (at least 128 bits?)
  - The extra input (message) size  $l$  is for efficiency ( $l=m, 2m, 3m, 4m, \dots$ )
- The current construction – iteration + MD strengthening– has some drawback need to be addressed



# Sponge Construction



The sponge construction (larger)

- $(p_0, \dots, p_i)$  input (message)
- $(z_0, z_1, \dots)$  output (hash code)
- $f$  can be any transformation (permutation)



# Exercise



1. What are the differences between collision attack and target attack?
2. There are  $m$  students in a room. What is the probability that there are exactly 3 of them have the same birthday?

Note: give the solution as a function of  $m$  (approximation is not needed).

1. For double DES  $E_{k2}(E_{k1}M)=C$ , using the birthday argument, by meet-in-the-middle, one can
  - Compute  $E_{k1}(M)=S$  for  $2^{32}$  choices of  $k1$
  - Compute  $D_{k2}(C)=T$  for  $2^{32}$  choices of  $k2$
  - because  $|\{S\}| |\{T\}| \cong 2^{64}$ , we find  $k1, k2$ , s.t  $E_{k2}(E_{k1}M)=C$
  - i.e. the complexity of break double DES is about  $2^{32}$ , not  $2^{56}$ .
- Is this correct, and why?
  - **Deadline: June 2**
  - **Format: Subject: CS381-name-EX.# to [gracehgs@mail.sjtu.edu.cn](mailto:gracehgs@mail.sjtu.edu.cn)**