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

2016-04

Organization

- Week 1 to week 16 (2016-02-24 to 2016-06-08)
- 东上院502
- 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: '薛伟佳' icelikejia@qq.com，'黄格仕' <huang.ge.shi@foxmail.com>
- Send homework to: laix@sjtu.edu.cn and to TAs

Rule: do not disturb others!

2

Contents

- Introduction -- What is security?
- Cryptography
  - Classical ciphers
  - Today's ciphers
  - Public-key cryptography
  - Hash functions/MAC
  - Authentication protocols
- Applications
  - Digital certificates
  - Secure email
  - Internet security, e-banking

Network security
SSL
IPSEC
Firewall
VPN
Computer security
Access control
Malware
DDos
Intrusion
Examples
Bitcoin
Hardware
Wireless

3

Content

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

4

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 ->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.
  - 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: $y=x^e$ mod N, T: N=p*q

7

☆ Hash function and applications

Definition. A hash function is an easily computable and publicly known function that maps the set of all binary sequences (message) of finite length to the set of binary sequences (hash code) of some fixed length

(The basic concept was likely duo to Yuval 1979)

Applications

- Modification Detection Code MDC
  M → $\boxed{\text{Hash}}$ → H
  M’ → $\boxed{\text{Hash}}$ → H’ ?= H
- Digital signatures
  M → $\boxed{\text{Hash}}$ → H → S(H), S(H) is the signature of message M
  M’ → $\boxed{\text{Hash}}$ → H’ ?= H, if yes, S(H) is also a valid signature of M’
  attack: submit M to sign, but attach M’.

Requirement: one-way and collision-free

8

Random oracle and hash function

- A random oracle is a “thing” that answers a question x:
  - If x is “new”, then the answer y is a uniform random variable;
  - If x has been asked once, then the answer y is the same as before.

Remark: RO is not a function (the answer y for a new question x is not uniquely determined.)

ROM (random oracle model): a security proving framework, in which the adversary attacks a system by querying ROs.

- In ROM, security proof is easier than in standard model.
- In a system that is proved secure in ROM, we replace the RO with a hash function, and hope the system is indeed secure.
- This approach is widely used.
- ROM has a flaw that, one can design a system which is secure in ROM but breakable when RO is replace by any fixed hash function.

Modification Detection

- Modification Detection Code MDC

  M →Hash→ H

  M’→Hash→ H’ $\stackrel{?}{=}$ H
- To provide integrity:
  - Store H=Hash(M) securely. Check H’=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’≠H, replace the website with the backup copy.
- Attack: to find a M’ ≠M, but H’=H.
- Requirement: second preimage resistant (one-way)

10

Digital signatures

- Digital signatures

  To sign a message M, first hash message M: H=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' ≠M, but H'=H; then signature of M is the same as the signature of M'. In the real attack, submit M to sign, but present (M', $S_x(H)$)
- Requirement: collision-free

11

Security of hash function

- One-way (second pre-image attack resistance) (weakly one-way)
  - Given, *M* and *H=Hash(M),* it is infeasible to find *M' ≠M, Hash(M')=Hash(M).*
- Collision free (collision resistance) (strong one-way)
  - It is infeasible to find different *M', M, Hash(M')=Hash(M).*
- Second pre-image and collision always exist! The key point is *infeasible*
- *Note:* one-wayness is much more important than collision free.
- If hash code length is m-bit, then:
  - To find second pre-image needs at most 2^m computations of *Hash*
  - To find collision needs at most $2^{m/2}$ computations of *Hash*

12

Birthday paradox

- 23 people in a room, it is likely that 2 of them have the same birthday

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

p=probability(2 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

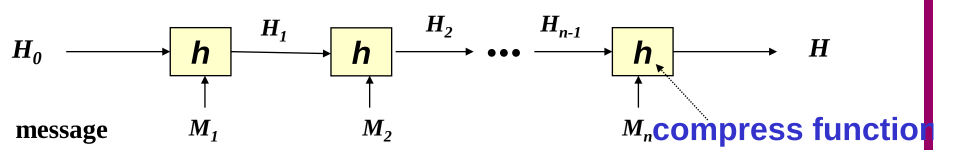
$$P = \frac{(N-1)}{N}\frac{(N-2)}{N}\cdots\frac{(N-(m-1))}{N} \approx e^{-m(m-1)/2N}$$

$e=2.71828$

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

13

Iterated Hash function



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

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

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

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

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

$H=H_n$

14

Attacks

Target attack (2nd pre-image attack):

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

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

Given (H_0, M), find $(H_0', M') \neq (H_0, M)$, but

$Hash(H_0,M) = Hash(H_0', M')$

Chosen-message target attack: For given H_0, specify a set C,

such that for each M in C, there is an $M' \neq M$, but

$Hash(H_0,M) = Hash(H_0, M')$

Collision attack: Given H_0, find M and $M' \neq M$, but

$Hash(H_0,M) = Hash(H_0, M')$

Semi free-start collision attack: Find H_0, M, $M' \neq M$, but

$Hash(H_0,M) = Hash(H_0, M')$

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

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

- Target attack → 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 $Hash(H_0,M) = Hash(H_0', M')$

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

  Semi free-start collision attack: Find H_0, M, $M' \neq M$, but

  $Hash(H_0,M) = Hash(H_0, M')$

- Wang et.al [2004]:

  Collision attack: Given H_0, find M and $M' \neq M$, but

  $Hash(H_0,M) = Hash(H_0, M')$

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 : $C_{FS\text{-}target} \le C_{target} \le 2^m$
- $C_{FS\text{-}collision} \le C_{semi\ FS\text{-}collision} \le C_{collision} \le 2^{m/2}$
- 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')

17

Attacks on Hash

Trivial free-start attack

$$Hash(H_0,M_1,M_2)=Hash(H_1,M_2)$$

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

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

$$H=Hash(H,M)=Hash(H,M,M)=Hash(H,M,M,M)=\ldots$$

Long-message target attack [Winternitz 84]:

If the given message has n blocks,then

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

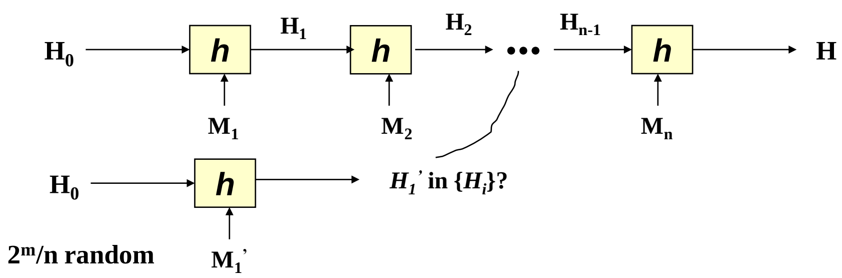
18

Long message attack

Long-message target attack [Winternitz 84]:

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

For 2m/n random M_1', compute $H_1=h(H_0, M_1')$
p(some H_1' = some H_i) ~ 0.63, for such H_1' and H_i
$Hash(H_0,M_1',M_{i+1},\dots M_n)= Hash(H_0,M_1,\dots,M_i,M_{i+1},\dots M_n)$



19

MD-strengthening

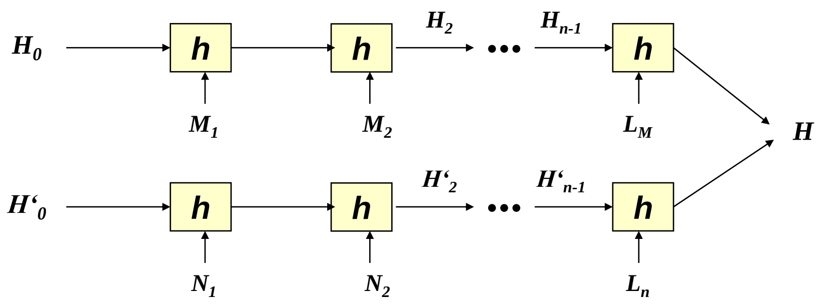
- Taking advantage that *M'* can have different length from *M*, one can break *Hash* without breaking *h*.
- Merkle-Damgaard strengthening:

  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]

20

Free-start Collision attack:

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



- Collision attack on $Hash_{MD}$ implies free-start collision on *h. (inverse is unknown, e.g. MD5, SHA)*

21

Target attack when *h* is not one-way

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

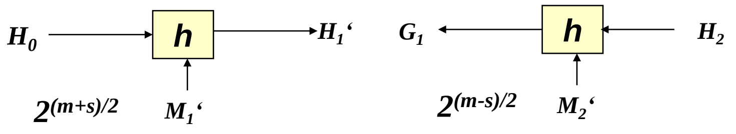
- *Th.3* $C_{target}(Hash_{MD}) \leq 2^{m/2}\, C_{FS\text{-}target}(h)^{1/2}$
  - If obtaining **random inverse** of *h* needs 2^s computations, then target attack on $Hash_{MD}(.,.)$ needs at most $2^{(m+s)/2}$ [Lai-Massey 92]

Attack: given $Hash_{MD}(H_0,M_1,M_2,M_3,..)$, *i.e, given* H_2, *compute forwards* $2^{(m+s)/2}$ *values of* H_1', *backwards* $2^{(m-s)/2}$ *values of* G_1

$p(\text{some } H'_1 = \text{some } G_1) = 1-[(1-2^{-m})^{(m-s)/2}]^{(m+s)/2} = 1-(1-2^{-m})^m = 0.63$

then, for such M_1', M_2',

$$Hash_{MD}(H_0,M_1',M_2',M_3,..) = Hash_{MD}(H_0,M_1,M_2,M_3,..)$$



22

Meet-in-the-middle

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

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

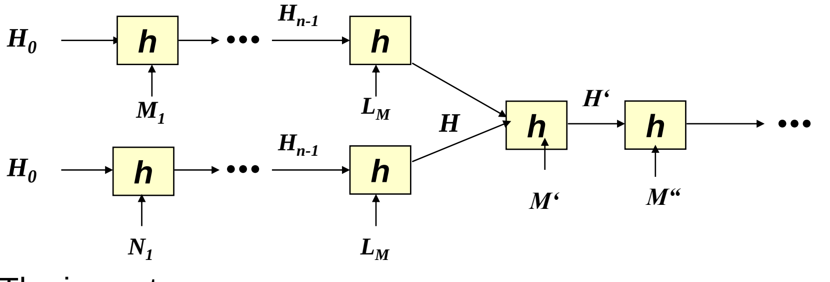
Theorem. A,B$\subset$S. if |A| |B| $\cong$ |S|, *then*

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

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

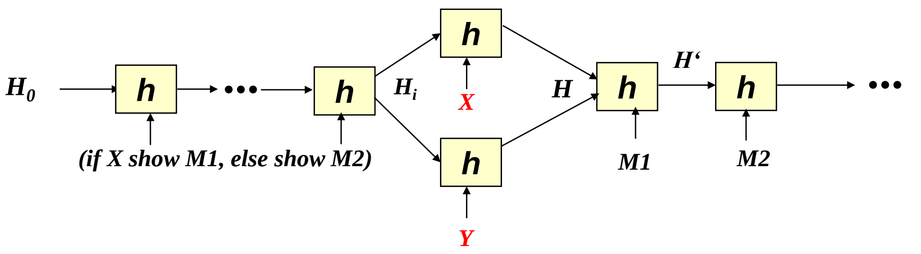
The issue with MD construction

- One collision(2nd-preimage) implies arbitrarily many collision(2nd-preimage)



- The impact:
  - "random" collision $\Rightarrow$ "useful/harmful" collision
  - Provable security in Random Oracle model may have flaw 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)

25

$Hash_{MD}$ - 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?)**

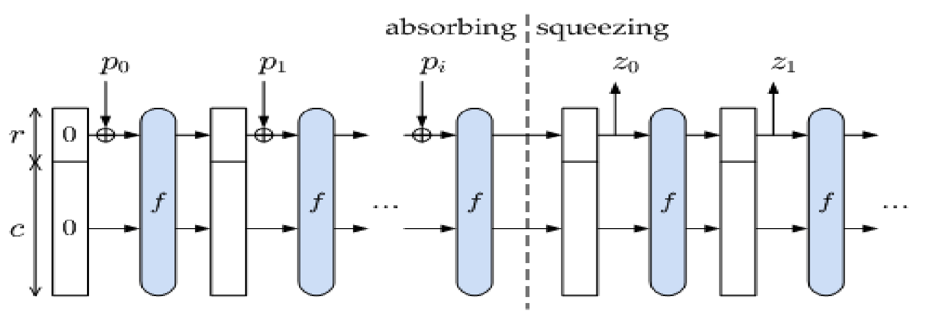
26

Compress functions

- Design a cryptographic hash function implies to find a one-way 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,...)
- The current construction – iteration + MD strengthening– has some drawback need to be addressed

27

Sponge Construction



The sponge construction (larger)

- (p_0,…p_i) input (message)
- (z_0,z_1,..) output (hash code)
- f can be any transformation (permutation)

Exercise 9.

1. What are the differences between collision attack and target attack?
2. 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: before next lecture
  - Format: email Subject: CS381-某某某-EX.#

Constructions of compress functions

Next lecture

- Hash function based on block ciphers
  - Single length, double length
- Dedicated hash functions
  - MD2, MD4, MD5
  - SHA-0,SHA-1,SHA-256,SHA-384,SHA-512
  - RipeMD,RipeMD-128,RipeMD-160
  - HAVAL
  - Tiger, Whirlpool
- Hash functions using modular operations