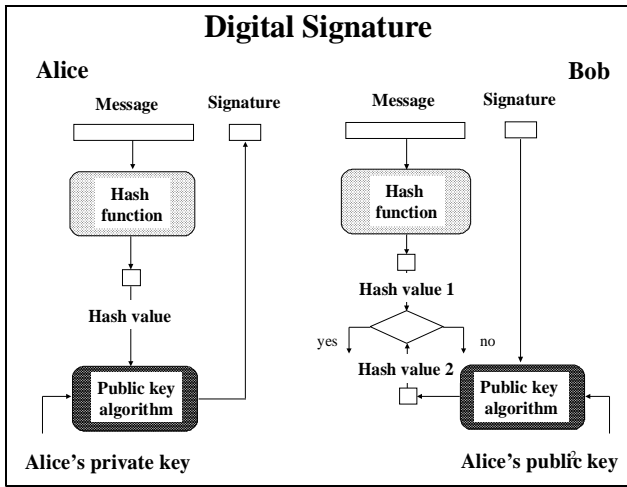


ECE 646 Lecture 12

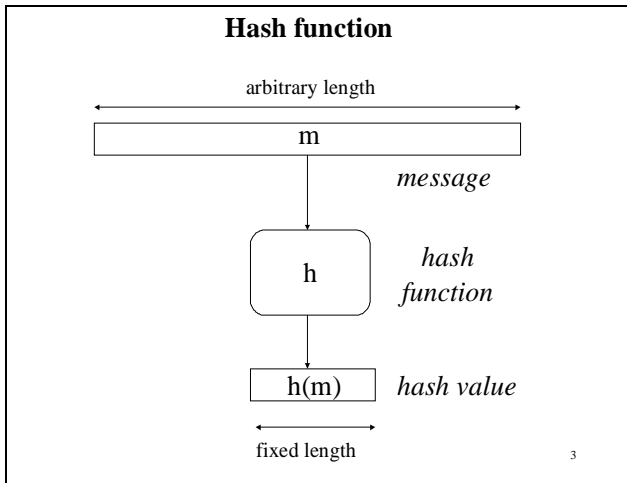
Hash functions & MACs

1

Digital Signature



Hash function



3

Vocabulary

hash function

message digest

hash value

message digest

hash total

fingerprint

imprint

cryptographic checksum

compressed encoding

MDC, Message Digest Code

Hash functions

Basic requirements

1. Public description, NO key
2. Compression
arbitrary length input → fixed length output
3. Ease of computation

Hash functions

Security requirements

It is computationally infeasible

Given

To Find

1. Preimage resistance

y

x , such that
 $h(x) = y$

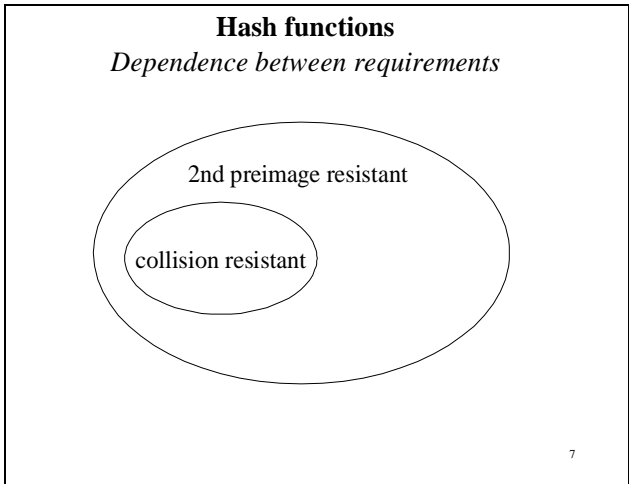
2. 2nd preimage resistance

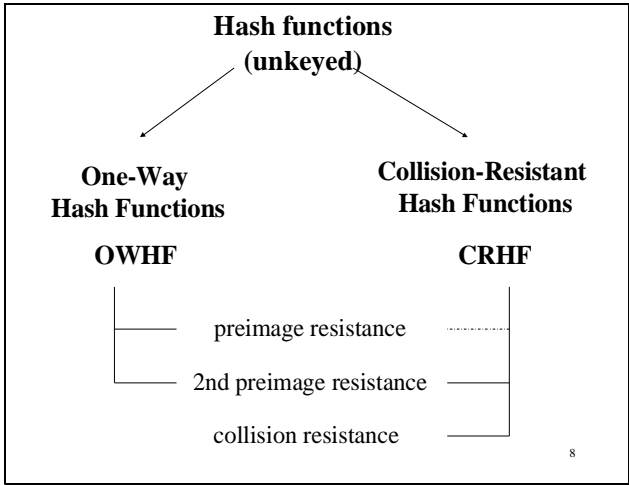
x and $y=h(x)$

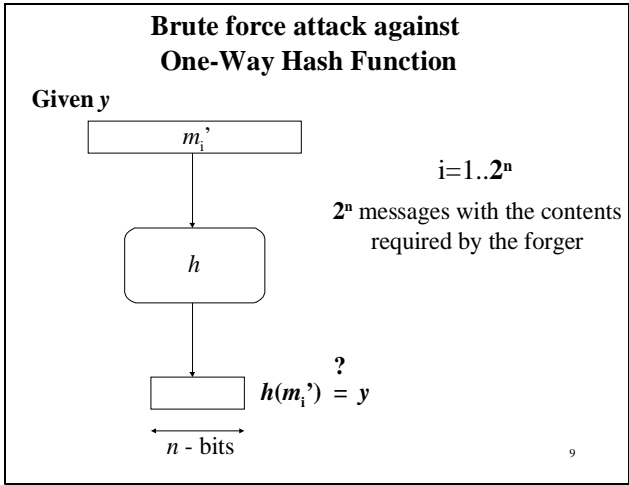
$x' \neq x$, such that
 $h(x') = h(x) = y$

3. Collision resistance

$x' \neq x$, such that
 $h(x') = h(x)$







Creating multiple versions of the required message

I { state confirm } { thereby } { - } that I { borrowed received }

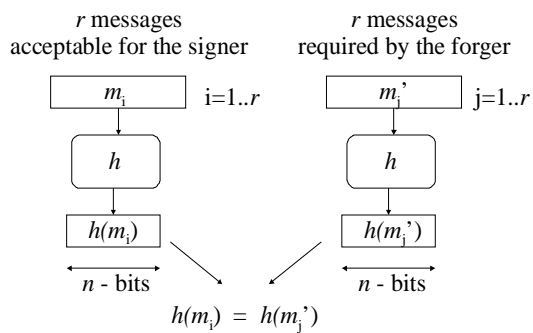
{ \$10,000 } { ten thousand dollars } from { Mr. } { Kris } { Dr. } { Krzysztof }

Gaj on { November 15, } { 11 / 15 / } 2005. This { money } { sum of money }

{ should } { is required to } be { returned } { given back } to { Mr. } { Dr. } Gaj

by the { 22nd } { 23rd } day of { November } { December } 2005.

Brute force attack against Collision Resistant Hash Function *Yuval*



Message required by the forger

I { state confirm } { thereby } { - } that I { borrowed received }

{ \$10,000 } { ten thousand dollars } from { Mr. } { Kris } { Dr. } { Krzysztof }

Gaj on { November 15, } { 11 / 15 / } 2005. This { money } { sum of money }

{ should } { is required to } be { returned } { given back } to { Mr. } { Dr. } Gaj

by the { 22nd } { 23rd } day of { November } { December } 2005.

Message acceptable for the signer

I { state } { thereby } that on { November 15, } 2005
confirm { - } 11 / 15 /

I { borrowed } { Mr. } { Kris } a { book }
received { Dr. } { Krzysztof } manuscript }

on { security in wireless networks. } This { text }
{ fast implementations of cryptography. } item }

{ should } { returned }
is required to be { given back } to { Mr. } Gaj
{ Dr. }

by the { 22nd } { November }
{ 23rd } day of { December } 2005.

13

Birthday paradox

How many students must be in a class so that there is a greater than 50% chance that

- 1. one of the students shares the teacher's birthday (up to the day and month)?*
- 2. any two of the students share the same birthday (up to the day and month)?*

14

Birthday paradox

How many students must be in a class so that there is a greater than 50% chance that

- 1. one of the students shares the teacher's birthday (day and month)?*
 $\sim 366/2 = 188$
- 2. any two of the students share the same birthday (day and month)?*

$$\sim \sqrt{366} \approx 19$$

15

**Brute force attack against
Collision Resistant Hash Function**

Probability p that two different messages have the same hash value:

$$p = 1 - \exp\left(-\frac{r^2}{2^n}\right)$$

For $r = 2^{n/2}$ $p = 63\%$

**Brute force attack against
Collision Resistant Hash Function**

Storage requirements

J.J. Quisquater

collision search algorithm

Number of operations: $2\sqrt{\pi/2} 2^{n/2}$

Storage: Negligible

Hash value size

One-Way	Collision-Resistant
----------------	----------------------------

Older algorithms:

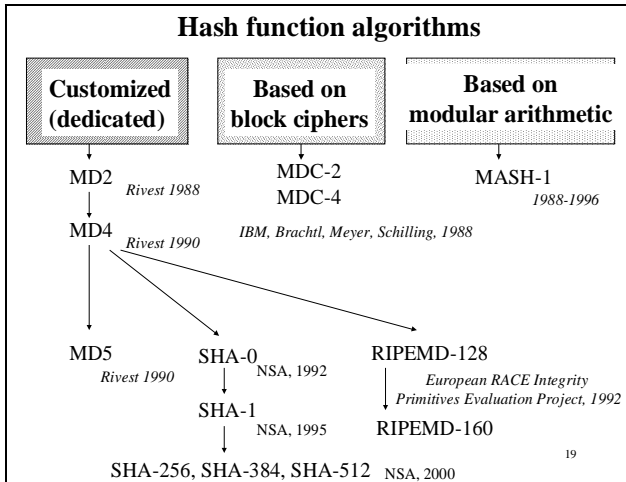
$n \geq 64$	$n \geq 128$
8 bytes	16 bytes

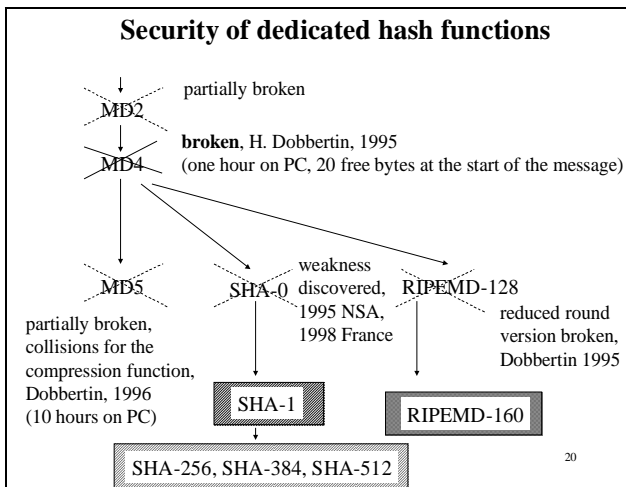
Current algorithms:

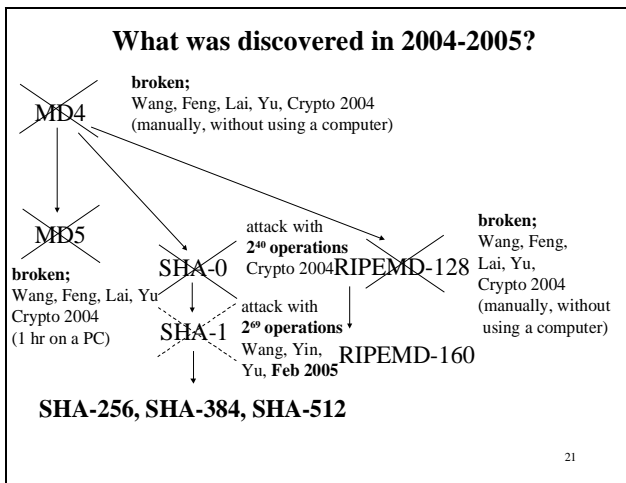
$n \geq 80$	$n \geq 160$
10 bytes	20 bytes

Newly proposed algorithms:

$n = 128, 192, 256$	$n = 256, 384, 512$
16, 24, 32 bytes	32, 48, 64 bytes







2⁶⁹ operations
Schneier, 2005

In hardware:

Machine similar to the one used to break DES:

Cost = \$50,000-\$70,000 **Time:** 3 years 3 months

or

Cost = \$25M-\$38M **Time:** 56 hours

In software:

Computer network similar to *distributed.net*
used to break DES (~331,252 computers) :

Cost = ~ \$0 **Time:** 38 years

Recommendations of NIST (1)

NIST Brief Comments on Recent Cryptanalytic Attacks on SHA-1
Feb 2005

The new attack is applicable primarily
to the use of hash functions in digital signatures.

In many cases applications of digital signatures
introduce additional context information,
which may make attacks impracticable.

Other applications of hash functions,
such as Message Authentication Codes (MACs),
are not threatened by the new attacks.

Recommendations of NIST (2)

NIST was already earlier planning to withdraw SHA-1
in favor of SHA-224, SHA-256, SHA-384 & SHA-512
by end of 2010

New implementations should use new hash functions.

NIST encourages government agencies to develop plans
for gradually moving towards new hash functions,
taking into account the sensitivity of the systems
when setting the timetables.

First NIST CRYPTOGRAPHIC HASH WORKSHOP

Oct. 31 – Nov. 1 2005 NIST
(Green Auditorium)
Gaithersburg, Maryland

Topics:

- security status of functions currently approved by NIST
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
- short term actions to mitigate the potential problems with the existing applications of SHA-1
- conditions that would warrant an early transition away from any of the approved hash functions
- potential replacement options for any of the approved hash functions
- features of hash functions required for different applications

25

2nd NIST Cryptographic Hash Workshop

August 24-25, 2006, UCSB Santa Barbara

- New Structures of hash functions
- Hash functions in practice
- Attacks against hash functions
- New hash function designs
- Result: Development of one or more new hash functions through a public competition like AES

<http://csrc.nist.gov/groups/ST/hash/index.html>

26

NISH Hash Algorithm Competition

- 2007: Define Requirements for new hash algorithm
- 2008: Submission deadline for new hash functions
- 2009: First hash function candidate conference
 - Presentation of new hash functions
- 2010: Second hash function candidate conference
 - Analysis results of new hash functions
 - Announce finalists
- 2012: Final hash function conference
 - Pick winner, Publish draft standard

27

Hash functions
Applications (1)

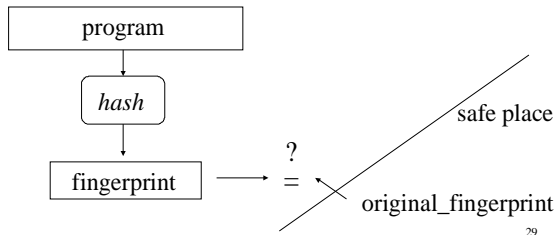
1. Digital Signatures

Advantages

1. Shorter signature
2. Much faster computations
3. Larger resistance to manipulation (one block instead of several blocks of signature)
4. Resistance to the multiplicative attacks
5. Avoids problems with different sizes of the sender and the receiver moduli

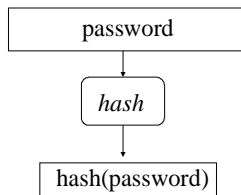
Hash functions
Applications (2)

2. Fingerprint of a program or a document
(e.g., to detect a modification by a virus or an intruder)



Hash functions
Applications (3)

3. Storing passwords

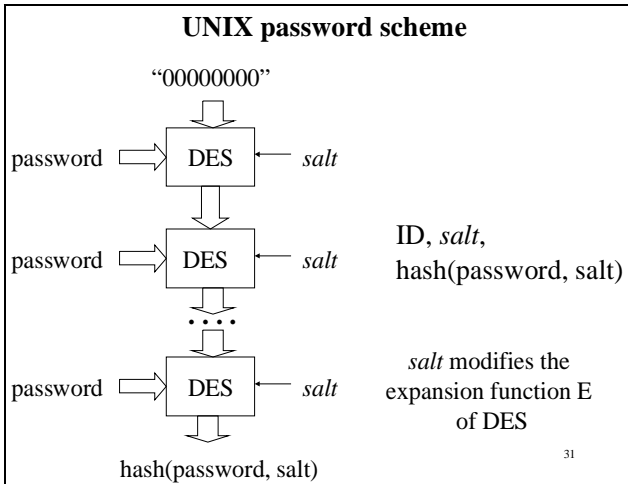


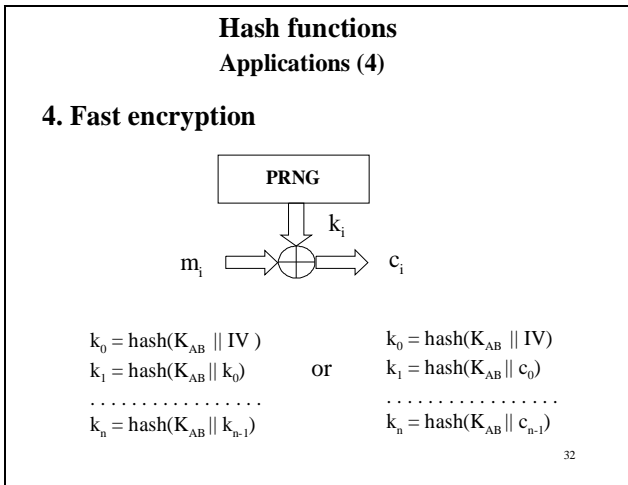
Instead of:

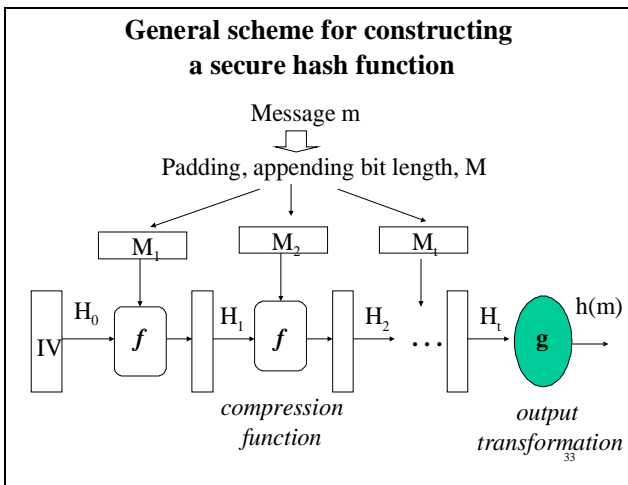
ID, password

System stores:

ID, hash(password)

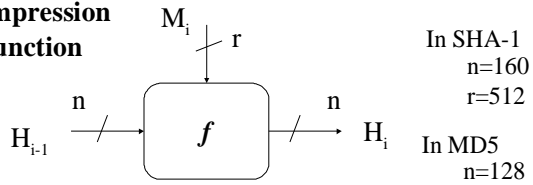






General scheme for constructing a secure hash function

Compression function



In SHA-1
 $n=160$
 $r=512$

In MD5
 $n=128$
 $r=512$

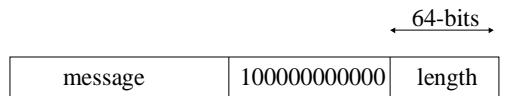
Entire hash

$$H_0 = IV$$

$$H_i = f(H_{i-1}, M_i)$$

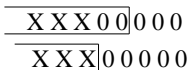
$$h(m) = g(H_t)$$

Hash padding

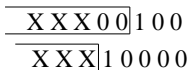


length
of the entire
message in bits

All zero padding:



Correct padding:



Parameters of dedicated hash functions

name	# bits of hash value	# bits of message block	no. of rounds (steps)	speed relative to MD4
MD4	128	512	3 x 16	1.00
MD5	128	512	4 x 16	0.68
SHA-1	160	512	4 x 20	0.28
RIPEMD-128	128	512	4 x 16	0.39
RIPEMD-160	160	512	5 x 16	0.24

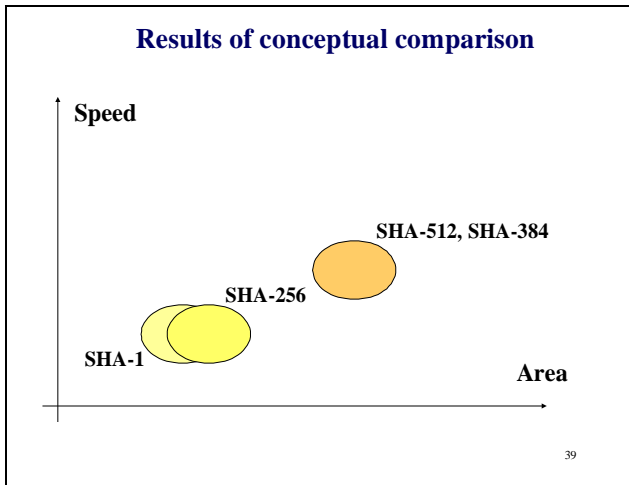
Parameters of new hash functions
Features affecting **security and functionality**

	SHA-1	SHA-256	SHA-384	SHA-512
Size of hash value	160	256	384	512
Complexity of the best attack	2^{80} now 2^{63}	2^{128}	2^{192}	2^{256}
Equivalently secure secret-key cipher	Skipjack	AES-128	AES-192	AES-256
Message size	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$ <small>37</small>

Parameters of new hash functions
Features affecting implementation **speed**

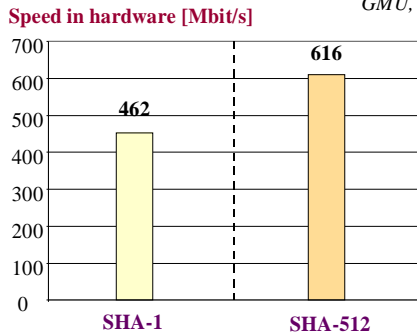
	SHA-1	SHA-256	SHA-384	SHA-512
Message block size	512	512	1024	1024
Number of digest rounds	80	64	80	80

38



Results of the prototype FPGA implementation

GMU, 2002



Complexity of the best attack the same as

SHA-1

2^{80}

Skipjack

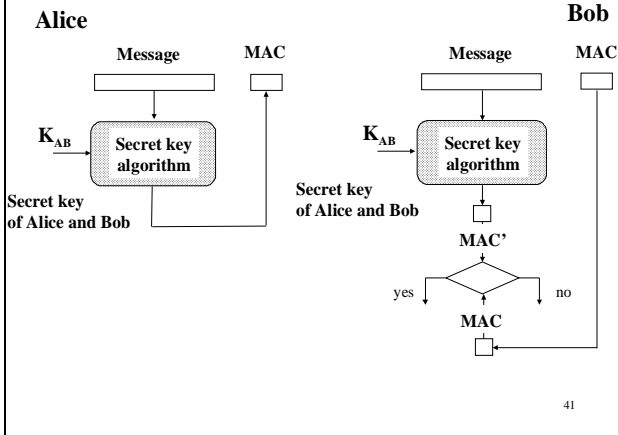
SHA-512

2^{256}

AES-256

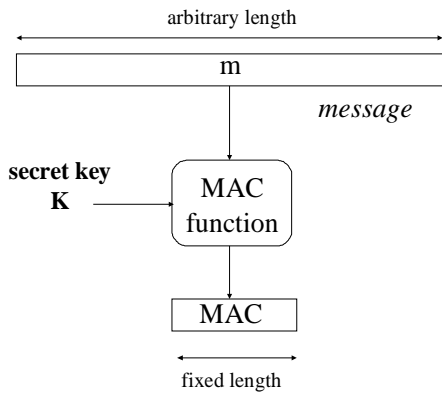
40

Authentication



41

MAC - Message Autentication Codes (keyed hash functions)



42

MAC functions

Basic requirements

1. Public description, SECRET key parameter
2. Compression
arbitrary length input → fixed length output
3. Ease of computation

MAC functions

Security requirements

Given zero or more pairs

$$m_i, \text{MAC}_k(m_i) \quad i = 1..k$$

it is computationally impossible to find any new pair

$$m', \text{MAC}_k(m')$$

Such that

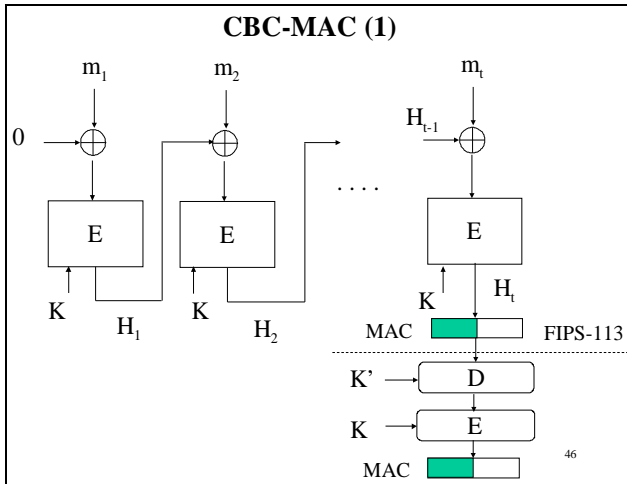
$$m' \neq m_i \quad i = 1..k$$

MAC functions

Security requirements

Resistance against

1. Known-text attack
2. Chosen-text attack
3. Adaptive chosen-text attack

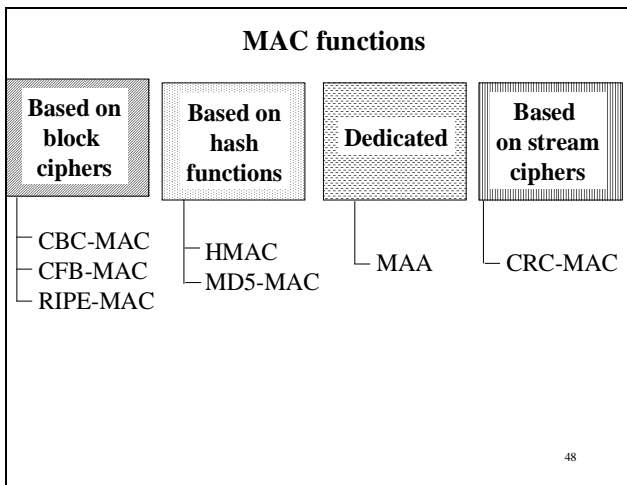


CBC-MAC (1)

$H_0 = IV = 0$
 $H_i = \text{DES}_K(m_i \oplus H_{i-1}) \quad i = 1..t$

 $\text{MAC}(m) = H_t[1..32]$
 or
 $\text{MAC}(m) = E_K(E_{K'}^{-1}(H_t))[1..32]$

47



RIPE-MAC

$$H_0 = IV = 0$$
$$H_i = \text{DES}_K(m_i \oplus H_{i-1}) \oplus m_i \quad i = 1..t$$
$$\text{MAC}(m) = E_K(E_{K^{-1}}(H_t))[0..31]$$
$$K' = K \oplus 0xf0f0\dots f0$$

49

HMAC

Bellare, Canetti, Krawczyk, 1996

Used in SSL and IPSec

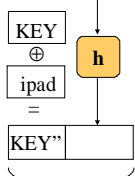
$$\text{HMAC}(m) = h(K \oplus \text{ipad} \parallel h(K \oplus \text{opad} \parallel m))$$

ipad, opad - constant padding strings of the length of the message block size in the hash function h

ipad = repetitions of 0x36 = 00110110
opad = repetitions of 0x5A = 01011010

50

HMAC



- American standard FIPS 198
- Arbitrary hash function and key size

h → HMAC

51
