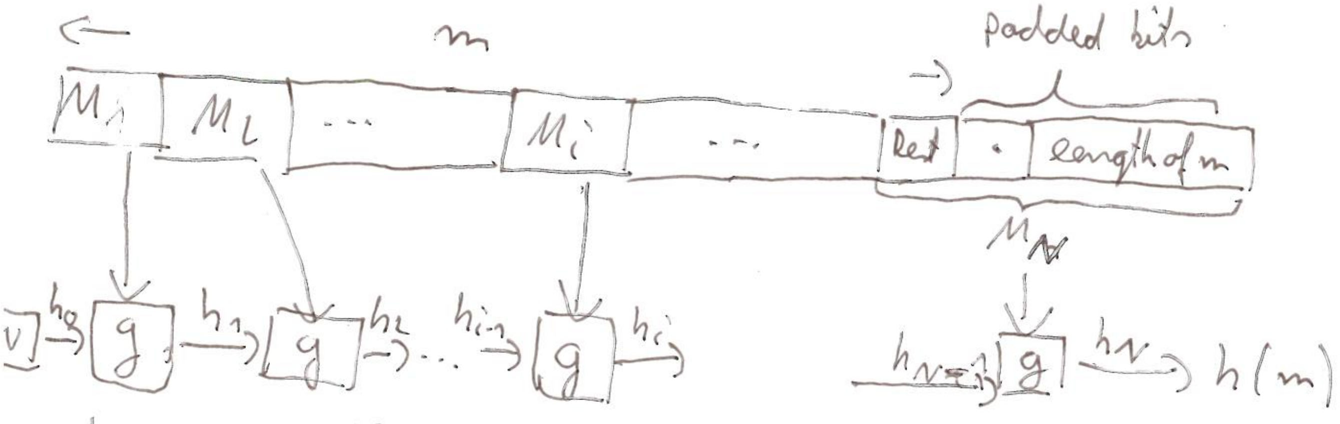
11.2 Construction of Hash Functions

Construction principles of most hash functions:



$h_0 = IV$ (Initial Value)

$h_i = g(h_{i-1}, M_i)$ $\quad i = 1, \dots, N$

$h_N = h(m)$ hash value of m

Some ~~of~~ hash functions of this type are

- MD5 Rivest, 1992, 128 bit hash length
- SHA-1 Successor of SHA (Secure hash standard)
  NIST, 1993, 160 bit length
- SHA-256, SHA-384, SHA-512
  NIST, 2001 256, 384, 512 bit of hash length
- FIPS 180-2 (Federal Information Processing Standards)
  Standard from Aug. 2002, contains the SHA-family, particularly SHA-3

Description of SHA-1

M_i: has length 512 bits

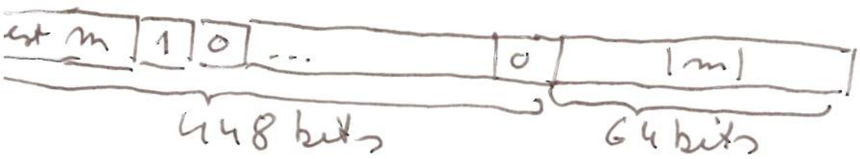
a) Operation on words of 32 bits

- $A \wedge B$, $A \vee B$, $A \oplus B$: bitwise and, or, xor
- $\neg A$: bitwise complement
- $A + B$: addition modulo 2^{32}
- $ROTL^s(A)$: cyclic shift to the left by $0 \leq s \leq 31$ pos.
- $A \| B$: concatenation of A and B

b) Padding of message m to a length s.t $512 \mid l_1$

Note: $|m| \leq 2^{64} - 1$ is assumed ($|m|$: length of m)

SHA-1-PAD (m):



i) append a single 1 to m

ii) Concatenate 0's s.t length is congruent 448 mod 512

iii) Concatenate length of m with 64 bits, i.e., leading zeros are included

c) Function and constants in SHA-1

$$f_i(B,C,D) = \begin{cases} (B \wedge C) \vee (\neg B \wedge D) & 0 \leq i \leq 19 \\ B \oplus C \oplus D & 20 \leq i \leq 39,\ 60 \leq i \leq 79 \\ (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) & 40 \leq i \leq 59 \end{cases}$$

$$k_i = \begin{cases} 5A827999 & 0 \leq i \leq 19 \\ 6ED9EBA1 & 20 \leq i \leq 39 \\ 8F1BBCDC & 40 \leq i \leq 59 \\ CA62C1D6 & 60 \leq i \leq 79 \end{cases}$$

d) Algorithm SHA-1 (see lecture notes)

Severe problems with hash functions have been demonstrated

Recommendation of the NIST from 2005:

- Don't use MD4 or MD5 anymore
- Find alternatives for SHA-1 until 2010, don't use it afterwards

Shamir has suggested to develop a complete redesign of hash functions likewise AES

Nov, 2007 NIST put out a call for developing a new hash fct.
Oct, 2012 end of competition, similar to AES

Winner "Keccak" published as NIST FIPS 202, contains
"SHA-3 standard"

Keccak developed by Daemen et al
Finalists were

BLAKE	(Aumasson et al)
Grøstl	(Knudsen et al)
JH	(Hongjun Wu)
Keccak	(Daemen et al)
Skein	(Schneier et al)

- Extension of construction principle
  - Division in „rate" & „capacity" part of hash function
  - Distinction between
    - absorbing phase (message blocks are used)
    - squeezing phase (generate output)

11 Digital Signatures

Method of signing a message in electronic form

Requirements (same as on conventional signatures)

- forgery - proof
- verifiable (proof of ownership)
- firmly connected to document

Problem for certain applications: repeated use of copies

Ex: Signed digital message for money transfer
Countermeasure against repeated use: time stamps

Attacks on signature schemes:

- Key only attack (Oscar knows the public key only)
- Known message attack (Oscar has signatures for a set of messages)
- Chosen message attack (Oscar obtains signatures for a set of chosen messages.)

Attacks may result in

- Total break (O can sign any message)
- Selective forgery (O can sign a particular class of messages
- Existential " (O can sign at least one message)

Known from Cryptography I: RSA signature scheme

Alice signs with public key (e, n), private key d

$s = [h(m)]^d \mod n$

Verification $h(m) = s^e \mod n$

Presented: Cryptography I: El Gamal signature scheme

11.1 ElGamal signature scheme

Parameters: p: prime, a PE mod p, h: hash function

Select random x, $y = a^x \bmod p$

Public key: (p, a, y) Private key: x

Signature generation

Select random k s.t. $k^{-1} \bmod p-1$ exists

$$\left.\begin{aligned} r &= a^k \bmod p \\ s &= k^{-1}(h(m) - x \cdot r) \bmod p-1 \end{aligned}\right\} (*)$$

Signature for m: (r, s)

Remark: k^{-1}, r, $x \cdot r$: can be computed in advance

Verification: Verify $1 \leq r \leq p-1$

$$v_1 = y^r r^s \bmod p$$

$$v_2 = a^{h(m)} \bmod p$$

if $v_1 = v_2$ we accept signature

Verification works:

Hence $v_1 \equiv y^r r^s \equiv a^{x \cdot r} a^{k \cdot s} \equiv a^{x \cdot r + k \cdot s} \overset{(**)}{\equiv}$

$$a^{l(p-1)+h(m)} \equiv \underbrace{(a^{p-1})^l}_{\equiv 1,\ \text{Fermat}} a^{h(m)} \equiv a^{h(m)} \equiv v_2 \pmod p$$

$(**)$: from $(*)$: $k \cdot s \equiv h(m) - x \cdot r \iff h(m) \equiv k \cdot s + x \cdot r \pmod{p-1}$

$\iff x \cdot r + k \cdot s = l(p-1) + h(m)$ for some $l \in \mathbb{Z}$

Security:

a) Don't use the same k twice! Otherwise

$$s_1 = k^{-1}(h(m_1) - x \cdot r) \bmod p-1 \quad (2)$$

$$s_2 = k^{-1}(h(m_2) - x \cdot r) \bmod p-1 \quad (3)$$

$$\Rightarrow (s_1 - s_2)\, k \equiv h(m_1) - h(m_2) \pmod{p-1}$$

$$\Rightarrow k \equiv (s_1 - s_2)^{-1}(h(m_1) - h(m_2)) \pmod{p-1}$$

provided $(s_1 - s_2)^{-1} \bmod p-1$ exists, but it exists with high prob.
Once k is known, x can be determined from (2) or (3), if r is invertible which is the case with high probability.

b) Oscar can forge a signature on a hashed message as follows:
Select any pair (u, v) s.t. $\gcd(v, p-1) = 1$

Compute
$$r \equiv a^u y^v \equiv a^{u + x \cdot v} \pmod p$$

$$s = -r \cdot v^{-1} \bmod p-1$$

Then (r, s) is a valid signature for $h(m) = s \cdot u \bmod p-1$

Proof:
$$v_1 = y^r \cdot r^s \equiv a^{x \cdot r}\, a^{(u + x \cdot v)(-r \cdot v^{-1})}$$

$$\equiv a^{x \cdot r - u \cdot r \cdot v^{-1} - x \cdot r \cdot \underbrace{v \cdot v^{-1}}_{\equiv 1 \pmod{p-1}}}$$

$$\equiv a^{-u \cdot r \cdot v^{-1}} \pmod p$$

$$v_2 = a^{h(m)} \equiv a^{su} \equiv a^{-r \cdot v^{-1} \cdot u} \equiv v_1 \pmod p$$

$$\Rightarrow v_1 = v_2$$