ICT Course: Introduction to Cryptography

Nguyen Minh Huong

ICT Department, USTH

March 1, 2023

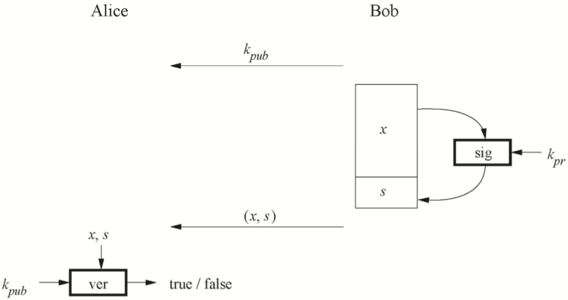
Session 7: Digital signature and Hash function

Digital Signature

Digital signature provides:

- Message authentication: assures that message is authentic to one user (the same functionality of hand-written signature)
- Integrity
- Nonrepudiation

Principles of Digital Signatures



- Key generation?
- Signature Algorithm?
- Verification function?

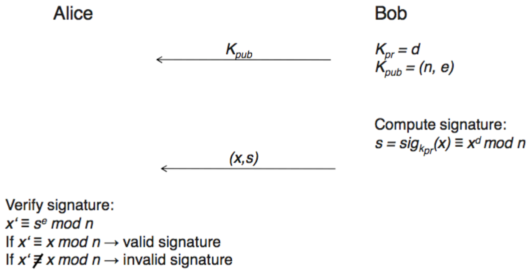
RSA signature scheme

- Generate private keys and public keys: use the RSA key generation
- Generate signature: "encrypt" the message with private key to obtain s, append s to the message
- Verify signature: "decrypt" the signature with public key and compare it with the message

RSA signature scheme - Details

Key generation:

- Private key: $k_{pr} = d$
- Public key: $k_{pub} = (n, e)$



26 Chapter 10 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

RSA Signature Scheme - Security

- n needs to be at least 1024 bits
- Forgery Attack:
  - Trudy can generate a valid signature for a random message
  - Method:
    1. Choose a signature
    2. Compute the respectively message from k_{pub}
  - Trudy can claim to Alice that he's Bob
- Solution against Forgery Attack: Padding Scheme

Elgamal Digital Signature

- Key generation?
- Signature generation and Verification
- Example: Bob wants to send a message to Alice. This time, it should be signed with the Elgamal digital signature scheme. Describe the key generation, signature generation and verification process.

$$p = 29, \alpha = 2, d = 12, k_E = 5$$

Digital Signature Algorithm (DSA)-Overview

- Federal US Government standard for Digital Signature
- Proposed by NIST
- Based on Elgamal signature scheme
- Signature length: 320 bits
- Signature verification is slower compared to RSA

DSA - Key Generation

Key generation:

1. Generate a prime p: $2^{1023} < p < 2^{1024}$
2. Find a prime divisor q of $p - 1$: $2^{159} < q < 2^{160}$
3. Find α: $ord(\alpha) = q$
4. Choose a random integer d: $0 < d < q$
5. Compute $\beta \equiv \alpha^d \ mod \ p$

Key pairs:

- $k_{pub} = (p, q, \alpha, \beta)$
- $k_{pr} = d$

DSA - Signature Generation - Verification

Signature Generation

1. Choose random ephemeral key k_E: $0 < k_E < q$
2. Compute $r \equiv (\alpha^{k_E}\ modp)\ mod\ q$
3. Compute $s \equiv (SHA(x) + d * r) * k_E^{-1}\ mod\ q$

Signature Verification

1. Compute $w \equiv s^{-1}\ mod\ q$
2. Compute $u_1 \equiv w * SHA(x)\ mod\ q$
3. Compute $u_2 \equiv w * r\ mod\ q$
4. Compute $v \equiv (\alpha^{u_1} * \beta^{u_2}\ mod\ p)\ mod\ q$
5. Verification $ver(x, (r, s))$:
   if $v \equiv r\ mod\ q$, then valid signature
   if $v \not\equiv r\ mod\ q$, then invalid signature

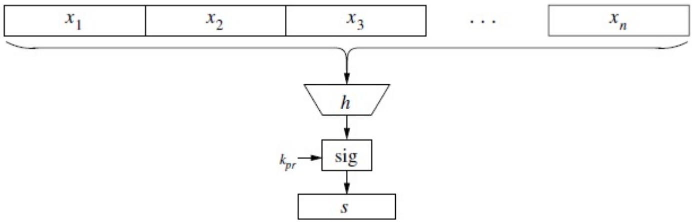
DSA - Example

Bob wants to send a message x to Alice which is to be signed with the DSA algorithm. Suppose the hash value of x is $h(x) = 26, p = 59, q = 29, \alpha = 3, d = 7, k_E = 10$. What is the key pair? Describe the signature generation and verification.

Hash function - Why it is required?

Signing a long message:

- If the message is divided into block size that is less than allowed input size of digital signature algorithm
  - High computational load
  - Message overhead
  - Security limitations
- a short signature for the message: hash function



Basic protocol for Digital Signature with a hash function

Alice		**Bob**
	$\xleftarrow{k_{pub,B}}$	
		$z = h(x)$ $s = \text{sig}_{k_{pr,B}}(z)$
	$\xleftarrow{(x,s)}$	
$z' = h(x)$ $\text{ver}_{k_{pub,B}}(s, z') = \text{true/false}$		

Desirable properties of hash function:

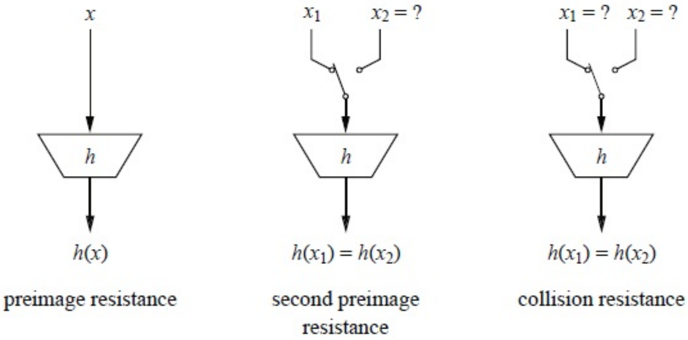
- Fast to compute
- Output is fixed length and independent of input length
- Computed fingerprint (output of the hash function) is highly sensitive to all input bits (a minor modification of input leads to different output)

Security requirements of hash functions

Properties needed for a hash function to be "secure":

1. preimage resistance (one-wayness): h(x) is one-way
2. second preimage resistance (weak collision resistance):
   $\nexists x_2 \neq x_1 : h(x_1) = h(x_2)$
3. collision resistance (strong collision resistance):
   $\nexists h(x_1) = h(x_2) : x_1 \neq x_2$

Security requirements of hash functions



Hash function algortihms

Two general types of hash functions:

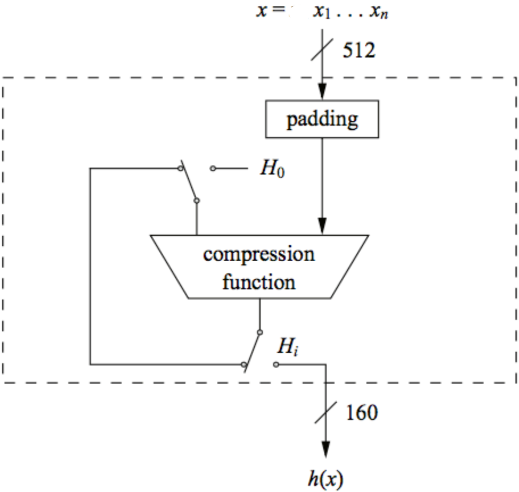
- Dedicated hash functions: algorithms are specifically designed to serve as hash functions, e.g., MD4 family: MD5, SHA-1...
- Block cipher hash functions: using block cipher to construct hash functions

Hash Algorithm SHA-1

- part of MD4 family
- based on Merkel-Damgard construction
- 160-bit output from a message of maximum length 2^{64} bits
- Widely used

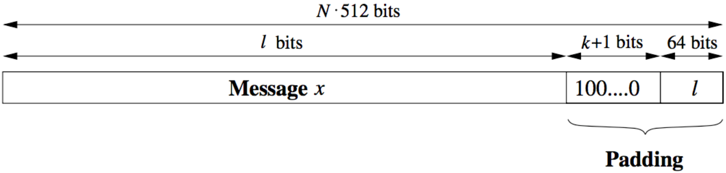
SHA-1 High-level Diagram

Compress function: 80 rounds, 20 rounds/stage



SHA-1 Prepocessing

- Padding:
    - Message length: l bits
    - Padded message length: multiple of 512 bits



- Dividing the padded message: 512 bit blocks x_i, each block $16\ words\ x\ 32\ bits$
- Initial fixed hash value H_0: 160 bits ≡ 5 words $H_0^{(i)}$

SHA-1 Hash computation

- Message schedule: 80 words W_j are derived from 16 words of a block in the padded message

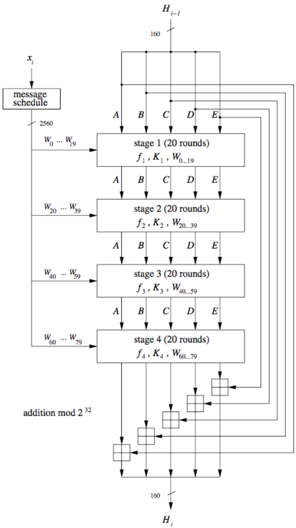
$$W_j = \begin{cases} x_i^{(j)} & \text{if } 0 \leq j \leq 15 \\ (W_{j-16} \oplus W_{j-14} \oplus W_{j-8} \oplus W_{j-3})_{<<<1} & \text{if } 16 \leq j \leq 79 \end{cases}$$

denote: $X_{<<<n}$: circular left shift of the word X by n bit positions.

SHA-1 Hash computation - Compression function

80-round compression function:

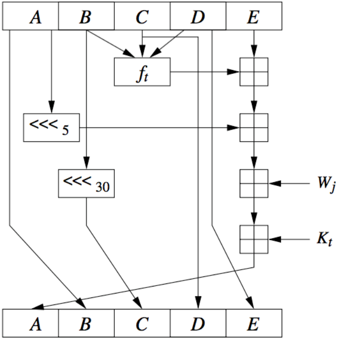
- In each stage t (20 rounds): internal function f_t, constant K_t are specified
- output after 80 rounds is added to the input value H_{i-1} modulo 2^{32} in word-wise fashion.



SHA-1 Hash function - Compression function

The operation within round j in stage t:

$A, B, C, D, E = (E + f_t(B, C, D) + (A)_{<<<5} + W_j + K_t), A, (B)_{<<<30}, C, D$



SHA-1 - Parameters

Initial hash values:

$$A = H^{(0)}{}_0 = 674523010$$

$$B = H^{(1)}{}_0 = EFCDAB890$$

$$C = H^{(2)}{}_0 = 98BADCFE0$$

$$D = H^{(3)}{}_0 = 103254760$$

$$E = H^{(4)}{}_0 = C3D2E1F0.$$

Round functions and round constants for the SHA rounds

Stage t	Round j	Constant K_t	Function f_t
1	$0 \ldots 19$	$K_1 =$ 5A827999	$f_1(B,C,D) = (B \wedge C) \vee (\bar{B} \wedge D)$
2	$20 \ldots 39$	$K_2 =$ 6ED9EBA1	$f_2(B,C,D) = B \oplus C \oplus D$
3	$40 \ldots 59$	$K_3 =$ 8F1BBCDC	$f_3(B,C,D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$
4	$60 \ldots 79$	$K_4 =$ CA62C1D6	$f_4(B,C,D) = B \oplus C \oplus D$

Exercise

Compute the output of the first round of stage 1 of SHA-1 for a 512-bit input block of

1. $x = 0 \ldots 00$
2. $x = 0 \ldots 01$ (i.e., bit 512 is one).

Ignore the initial hash value H_0 for this problem (i.e., $A_0 = B_0 = \ldots = 00000000hex$).