COMP3260/6360
Data Security
Lecture 10

Prof Ljiljana Brankovic
School of Electrical Engineering and Computer Science

Lecture Overview

1. Hash Functions
   a) Requirements
   b) Simple Hash Functions
   c) Birthday Attack
   d) Security
   e) Hash Algorithms: MD5 and SHA

2. Digital Signatures
   a) Direct Digital Signatures
   b) Arbitrated Digital Signatures
   c) Attacks and Forgeries
   d) Requirements
   e) Digital Signature Algorithms

Hash Functions and Digital Signatures

Chapter 11 from text: Cryptographic Hash Functions

Chapter 13 from text: Digital Signatures

Note that in-text references and quotes are omitted for clarity of the slides. When you write as essay or a report it is very important that you use both in-text references and quotes where appropriate.

Hash Functions

Requirements:

- can be applied to block of data of any size\produces a fixed-length output
- easy to compute
- hash function is ***one-way function*** and it is computationally infeasible to find x such that H(x) = h, for any given h – ***preimage resistant***
- for any given x it is computationally infeasible to find y such that H(x)=H(y) - ***weak collision resistance, or second preimage resistance***
- it is computationally infeasible to find any pair (x,y) such that H(x)=H(y) - ***strong collision resistance, or collision resistance***

Simple Hash Functions

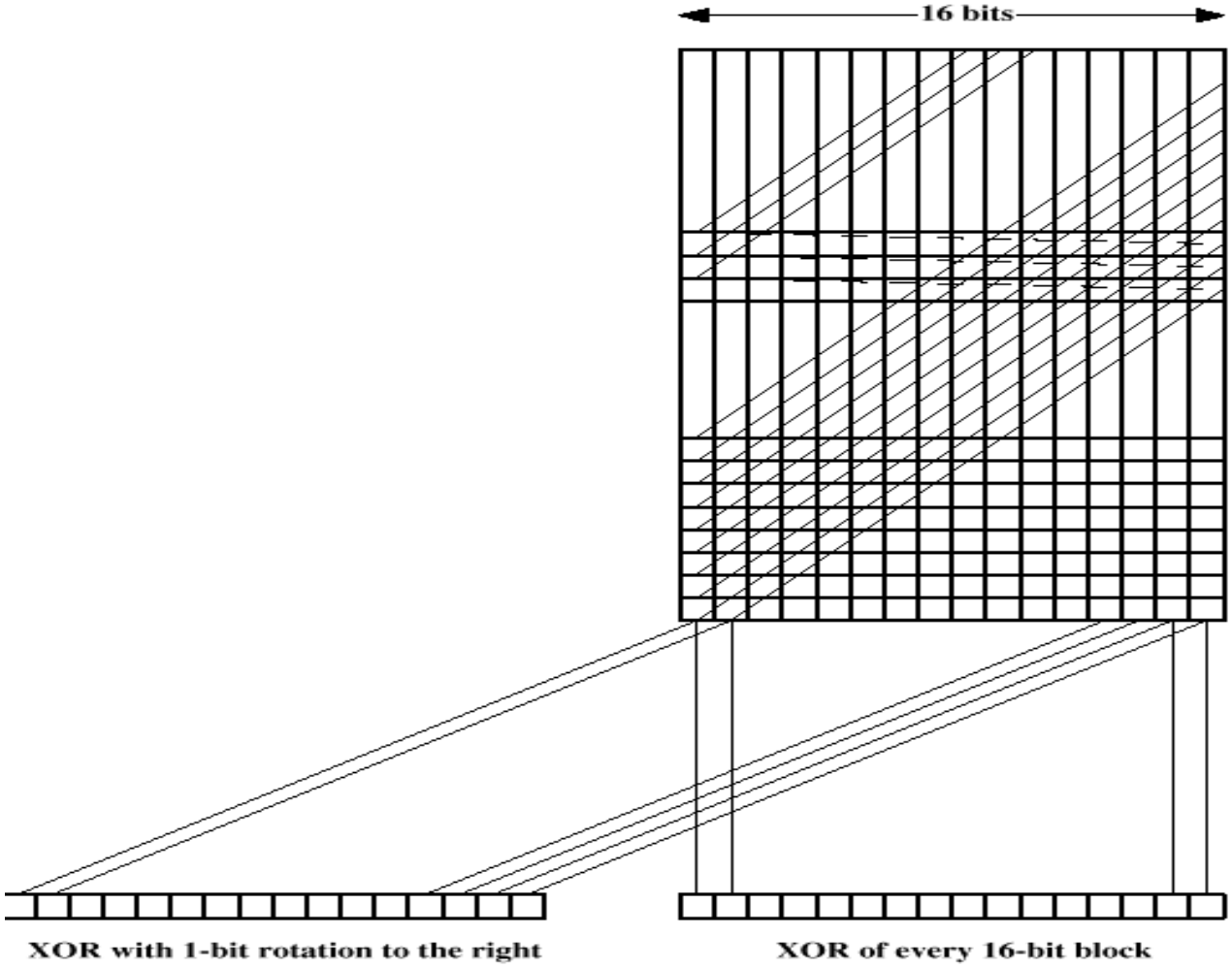


Figure 8.8 Two Simple Hash Functions

Birthday Attack

- A is prepared to sign a message (to create an *m*-bit hash code and encrypt it with A's private key)
- The opponent generates $2^{m\backslash 2}$ variations of the message with the same meaning, and an equal number of false messages
- The two sets are compared to find a pair that produces the same hash code. If no match is found, the additional messages are generated
- The opponent offers the valid variation to A for signature but sends the false variation

Dear Anthony,

{This letter is / I am writing} to introduce {you to / to you} {Mr. / --} Alfred {P. / --}

Barton, the {new / newly appointed} {chief / senior} jewellery buyer for {our / the}

Northern {European / Europe} {area / division}. He {will take / has taken} over {the / --}

responsibility for {all / the whole of} our interests in {watches and jewellery / jewellery and watches}

in the {area / region}. Please {afford / give} him {every / all the} help he {may need / needs}

to {seek out / find} the most {modern / up to date} lines for the {top / high} end of the

market. He is {empowered / authorized} to receive on our behalf {samples / specimens} of the

{latest / newest} {watch and jewellery / jewellery and watch} products, {up / subject} to a {limit / maximum}

of ten thousand dollars. He will {carry / hold} a signed copy of this {letter / document}

as proof of identity. An order with his signature, which is {appended / attached}

{authorizes / allows} you to charge the cost to this company at the {above / head office}

address. We {fully / --} expect that our {level / volume} of orders will increase in

the {following / next} year and {trust / hope} that the new appointment will {be / prove}

{advantageous / an advantage} to both our companies.

Figure 11.8 A Letter in 2^{3} ^{7} Variations [DAVI89]

Security of Hash Functions

Brute force

- hash function - one way 2^{m-1}
  - weak collision resistance 2^{m-1}
  - strong collision resistance $2^{m\backslash 2}$
- MAC $\min(2^k, 2^m)$

Cryptanalysis

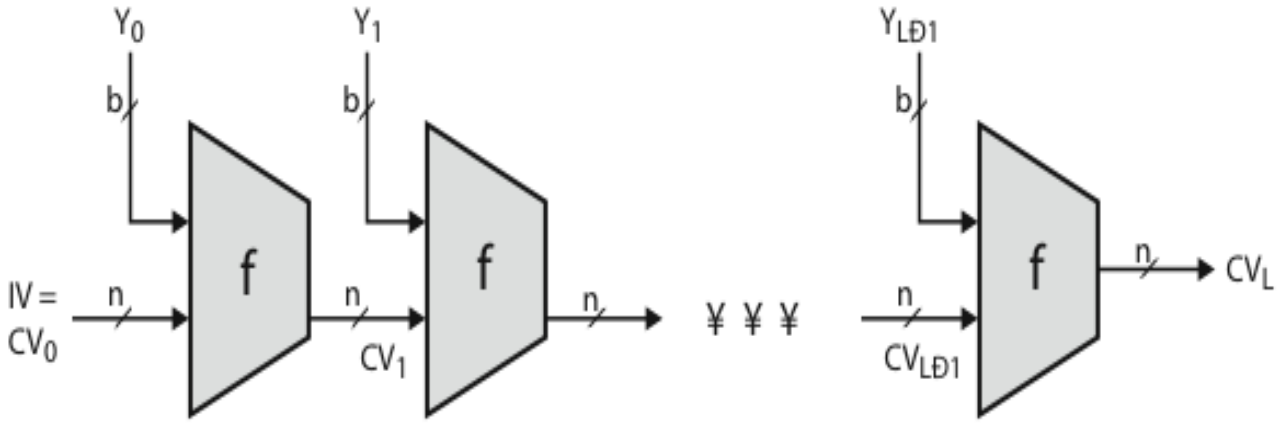
Hash and MAC Algorithms

- Hash Functions
  - condense arbitrary size message to fixed size
  - by processing message in blocks
  - through some compression function
  - either custom or block cipher based
- Message Authentication Code (MAC)
  - fixed sized authenticator for some message
  - to provide authentication for message
  - by using block cipher mode or hash function

Hash Algorithms

- We see similarities in the evolution of hash functions & block ciphers
  - increasing power of brute-force attacks
  - leading to evolution in algorithms
  - from DES to AES in block ciphers
  - from MD4 & MD5 to SHA-1, SHA-2 amd SHA3 in hash algorithms
- Also, hash functions tend to use common iterative structure as do block ciphers.

Hash Algorithm Structure



IV = Initial value
CV_i = chaining variable
Y_i = ith input block
f = compression algorithm

L = number of input blocks
n = length of hash code
b = length of input block

MD5

- Designed by Ronald Rivest (the R in RSA)
- Latest in a series of MD2, MD4
- Produces a 128-bit hash value
- In the past it was the most widely used hash algorithm
  - in recent times have both brute-force & cryptanalytic concerns
- Specified as Internet standard RFC1321

Strength of MD5

- MD5 hash is dependent on all message bits
- Rivest claims security is good as can be
- Known attacks are:
  - Boer & Bosselaers (93) found a pseudo collision
  - Dobbertin (96) created collisions on MD compression function (but initial constants prevent exploit)
  - On 18 March 2006, Klima reported an algorithm that finds a collision within one minute on a single notebook computer.
- Conclusion is that MD5 is vulnerable

SHA Algorithms

- The Secure Hash Algorithm (SHA) was developed and published by NIST in 1993. It was revised in 1995 and the revised version was called SHA-1.

- Since then 3 new versions have been defined; SHA-256, SHA-384 and SHA-512 (SHA-2). SHA-1 is phased out. There is a collision attack on SHA-1 that uses 2^{69} operations and thus it is no longer considered secure.

SHA Parameters

	SHA-1	SHA-256	SHA-384	SHA-512
Message digest size	160	256	384	512
Message size	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Block size	512	512	1024	1024
Word size	32	32	64	64
Number of steps	80	64	80	80
Security*	80	128	192	256

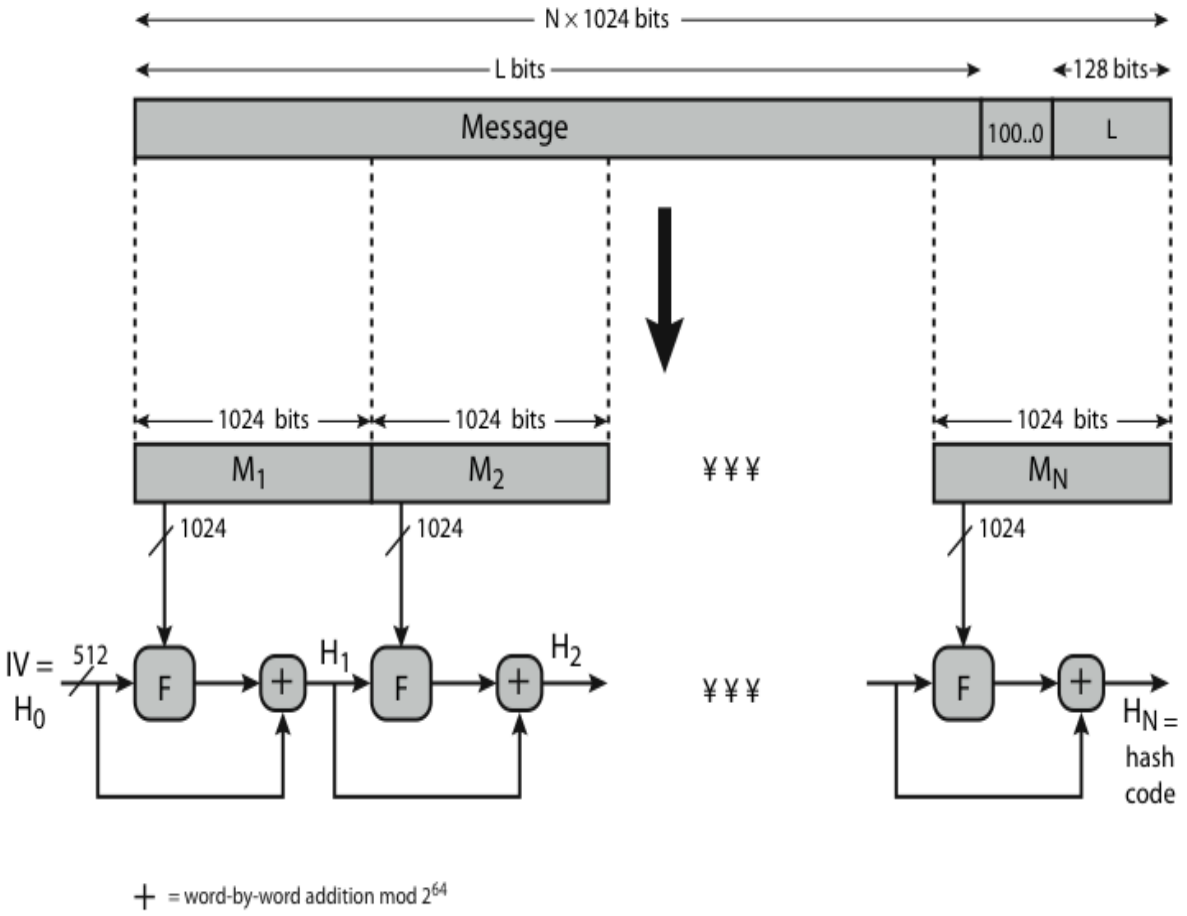
All sizes are measured in bits

*Security refers to the fact that a birthday attack on a message digest of size *n* produces a collision with a workfactor of approximately $2^{n/2}$

SHA-512

- Steps
  1. Append padding bits
  2. Append length
  3. Initialise hash buffer
  4. Process message in 1024-bit blocks
  5. Output.

SHA-512 Overview



SHA-512

Steps explained:

1. Append padding bits – the padding length is 896 mod 1024; the padding is always added, even if the message is already 896 mod 1024; the padding consists of a single 1 followed by the necessary number of 0s.
2. Append length – 128-bit representations of the original message length. Thus the total length (original message + padding + length) is a multiple of 1024.

SHA-512

3. Initialise hash buffer – 512 bit buffer holds intermediate results and the final hash code; the buffer can be seen as eight 64-bit registers a, b, c, d, e, f, g and h. The initial value (IV) of the buffer is as follows:

   a = 6A09E667F3BCC908
   b = BB67AE8584CAA73B
   c = 3C6EF372FE94F82B
   d = A54FF53A5F1D36F1
   e = 510E527FADE682D1
   f = 9B05688C2B3E6C1F
   g = 1F83D9ABFB41BD6B
   h = 5BE0CDI9137E2179

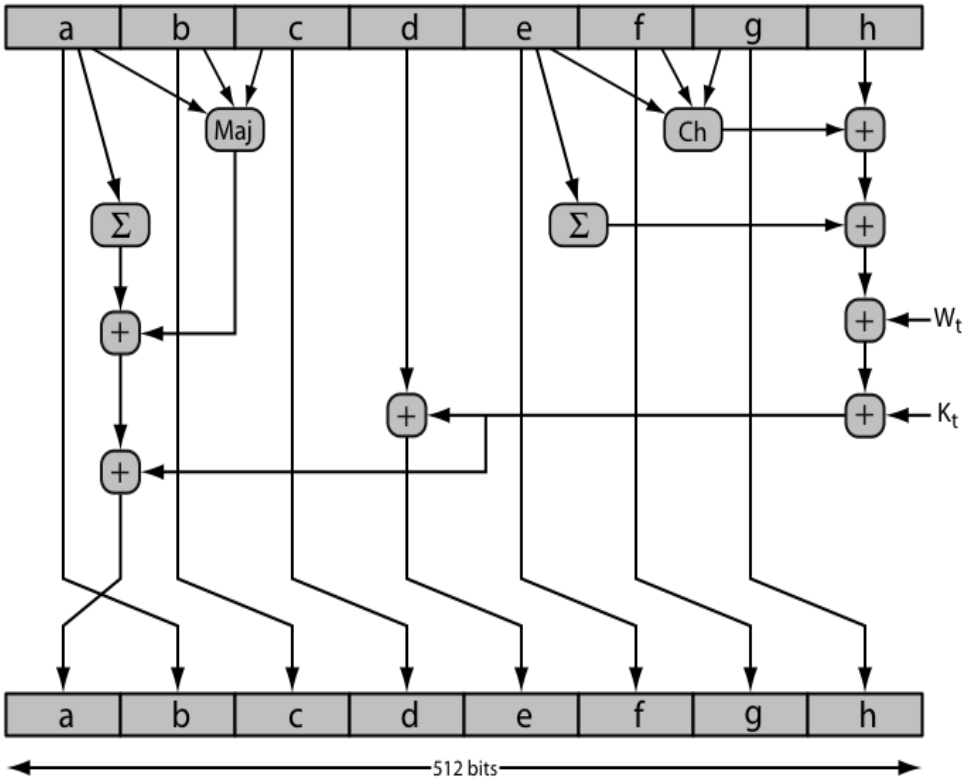
These values are stored in the big-endian format – the most significant byte in the leftmost byte position.

SHA-512

4. Process message in 512-bit blocks – the input to each round t is the 512-bit buffer, the 64-bit W_t derived from the current 1024-bit block of M, and the additive constant K_t.

function	formula	description
Ch(e,f,g)	(e AND f) ⊕ (! e AND g)	If e then f else g
Maj(a,b,c)	(a AND b) ⊕ (a AND c) ⊕ (b AND c)	True if majority of the arguments are true
$\sum_0^{512} a$	ROTR^{28}(a) ⊕ROTR^{34}(a) ⊕ROTR^{39}(a)	ROTR^n(x) is the cicular left shift of x by n positions
$\sum_1^{512} e$	ROTR^{14}(e) ⊕ROTR^{18}(e) ⊕ROTR^{41}(e)	ROTR^n(x) is the cicular left shift of x by n positions

SHA-512



Elementary SHA-512 Operation (Single Round)

SHA-512

5. Output – after all 1024-bit blocks have been processed, the content of the 512-bit buffer is the hash code (or the message digest).

SHA-512

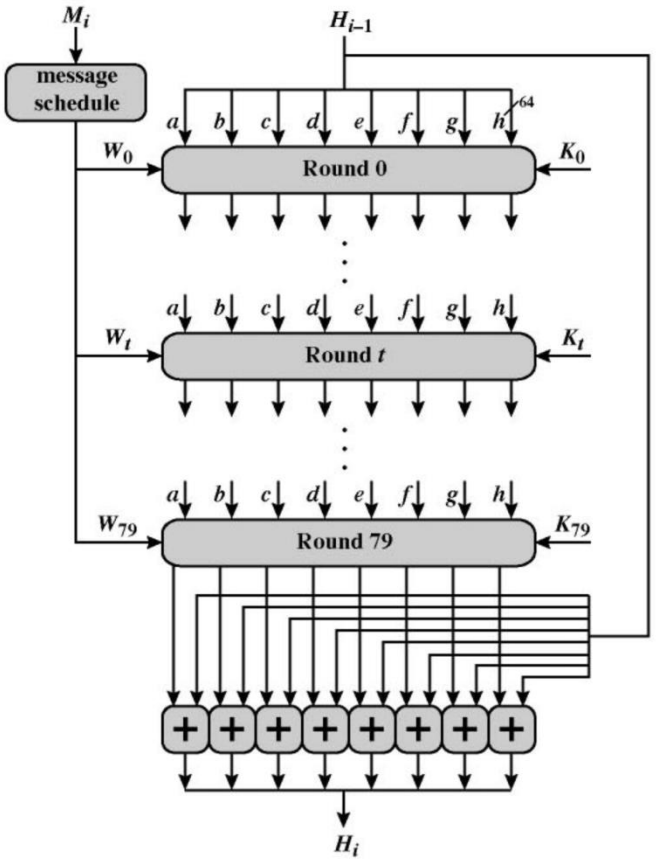
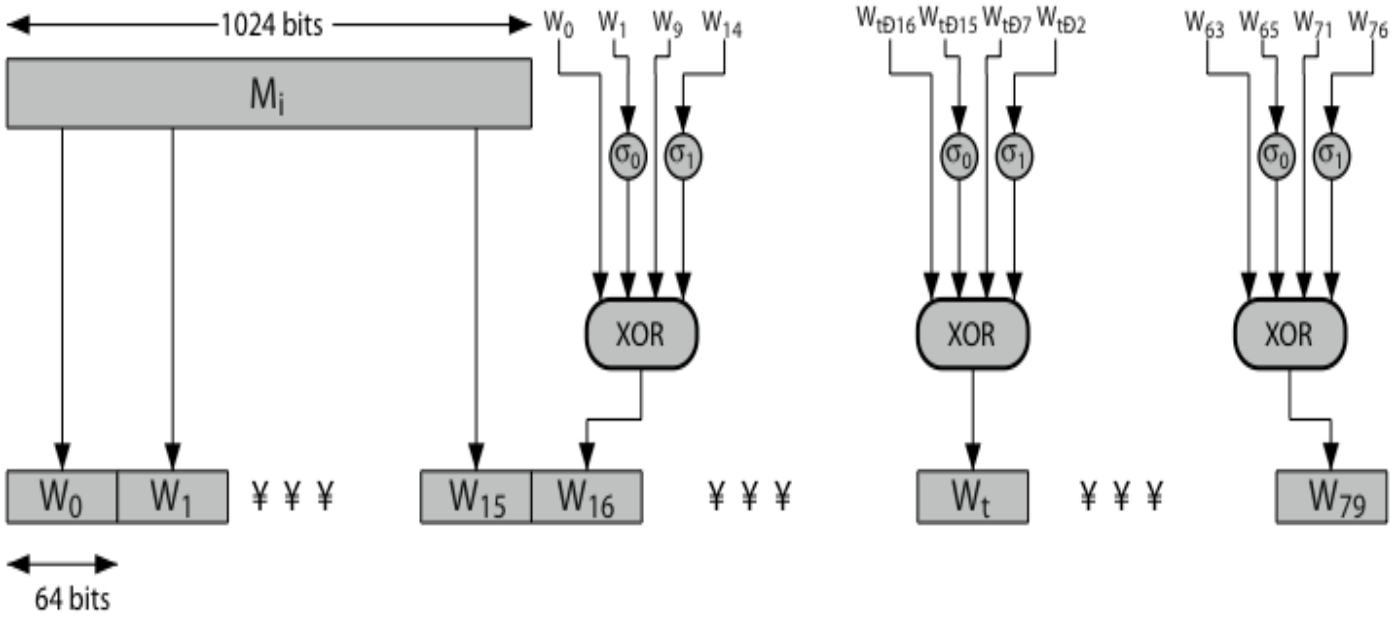


Figure 12.2 SHA-512 Processing of a Single 1024-Bit Block

SHA-512



Creation of 80-word Input Secquence for SHA-512
Processing of Single Block

Chapter 13 – Digital Signatures & Authentication Protocols

To guard against the baneful influence exerted by strangers is therefore an elementary dictate of savage prudence. Hence before strangers are allowed to enter a district, or at least before they are permitted to mingle freely with the inhabitants, certain ceremonies are often performed by the natives of the country for the purpose of disarming the strangers of their magical powers, or of disinfecting, so to speak, the tainted atmosphere by which they are supposed to be surrounded.

—*The Golden Bough*, Sir James George Frazer

Digital Signatures

- have looked at message authentication
  - but does not address issues of lack of trust
- digital signatures provide the ability to:
  - verify author, date & time of signature
  - authenticate message contents
  - be verified by third parties to resolve disputes
- hence include authentication function with additional capabilities

Direct Digital Signatures

- The direct digital signature involves only the two parties that are communicating. It can be produced either by encrypting the entire message with the sender's private key, or by encrypting a hash code of the message with the sender's private key.

- The security of the scheme depends on the security of the sender's private key. The sender can deny sending the messages by claiming that his/her private key was stolen and that somebody else forged the signature.

- To diffuse this threat, it is possible to require every signed message to include a timestamp, and also to require prompt reporting of compromised keys.

Arbitrated Digital Signatures

- Digital signatures can also be implemented with conventional cryptosystem and an arbitrator Y.
- It is assumed that every participant has a secret key that he/she shares with the arbitrator.
- We will denote by K_A the secret key shared by user A and the arbitrator. We denote the message recipient as B.

Arbitrated Digital Signatures

1. User A encrypts their message to B using K_A and send it to the arbitrator Y.

2. Y decrypts the message with K_A.

3. The arbitrator takes the decrypted message, appends to it the statement that they received the message from A, encrypts it with key K_B that they share with user B, and send it to B.

4. B decrypts it with K_B and they can now read both the message and the statement from the arbitrator.

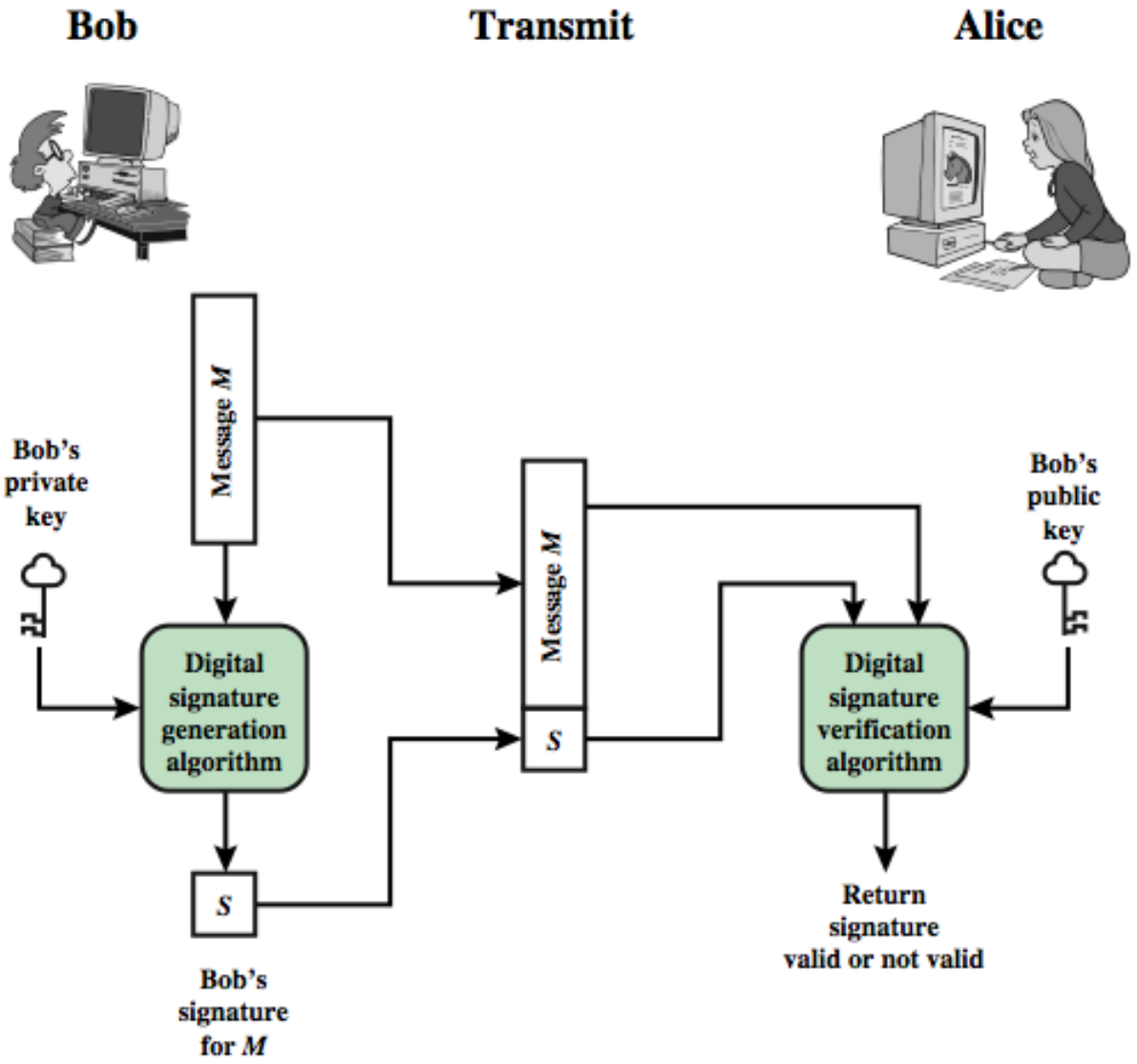
Arbitrated Digital Signature

- The problem with this scheme is that it is not straightforward for B to convince a third party that the message came from A – they have to go through the arbitrator.

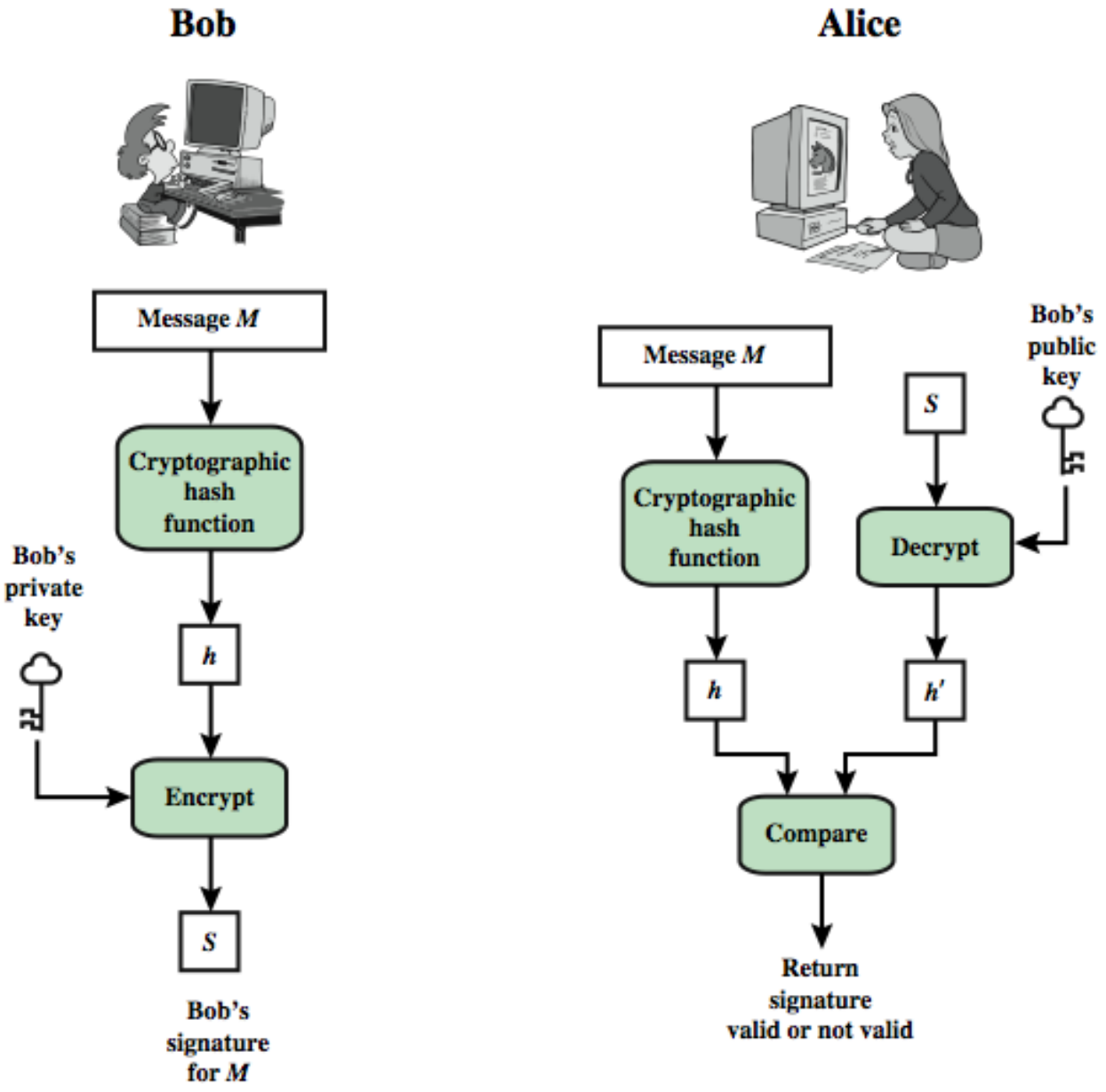
Arbitrated Digital Signatures

- involves use of arbiter Y
  - validates any signed message
  - then dated and sent to recipient
- requires suitable level of trust in arbiter
- can be implemented with either private or public-key algorithms
- arbiter may or may not see message

Digital Signature Model



Digital Signature Model



Attacks and Forgeries

- key-only attack – BB only knows Alice's public key
- known message attack – BB has a set of messages and their signatures
- generic chosen message attack – BB chooses a set of messages independent of A's public key
- directed chosen message attack – BB chooses a set of messages based on A's public key
- adaptive chosen message attack – BB can use A as an oracle

Attacks and Forgeries

- break success levels
  - total break – BB determines Alice's private key
  - universal forgery – BB can forge A's signature on any message
  - selective forgery – BB can forge A's signature on some message of his choice
  - existential forgery – BB can forge A's signature on at least message which he can't choose

Digital Signature Requirements

- must depend on the message signed
- must use information unique to sender
  - to prevent both forgery and denial
- must be relatively easy to produce
- must be relatively easy to recognize & verify
- be computationally infeasible to forge
  - with new message for existing digital signature
  - with fraudulent digital signature for given message
- be practical to save digital signature in storage

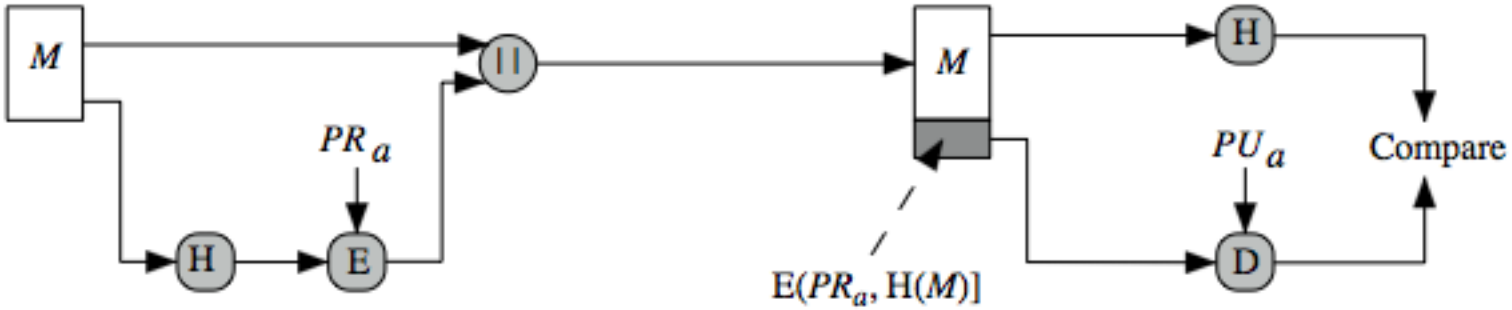
Direct Digital Signatures

- involve only sender & receiver
- assumed receiver has sender's public-key
- digital signature made by sender signing entire message or hash with private-key
- can encrypt using receivers public-key
- important that sign first then encrypt message & signature
- security depends on sender's private-key

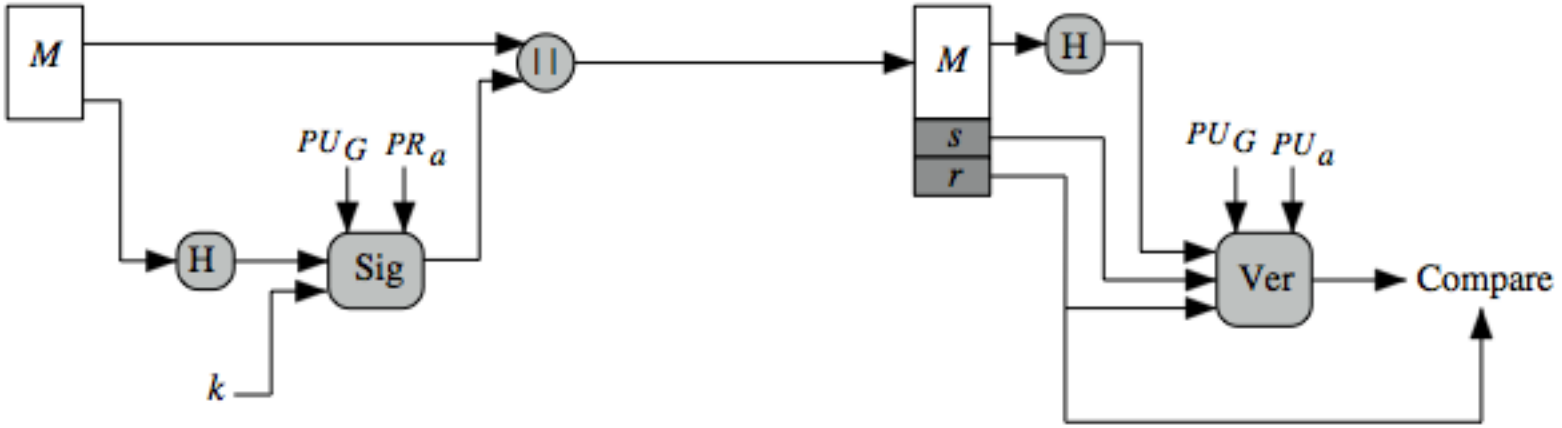
Digital Signature Standard (DSS)

- US Govt approved signature scheme
- designed by NIST & NSA in early 90's
- published as FIPS-186 in 1991
- revised in 1993, 1996
- uses the SHA hash algorithm
- DSS is the standard, DSA is the algorithm
- FIPS 186-2 (2000), FIPS 186-3 (2009) and FIPS 186-4 (2013) includes alternative RSA & elliptic curve signature variants
- DSA is digital signature only unlike RSA
- DSA is a public-key technique

DSS vs RSA Signatures



(a) RSA Approach



(b) DSS Approach

Digital Signature Algorithm (DSA)

- creates a 320 bit signature
- with 512-1024 bit security
- smaller and faster than RSA
- a digital signature scheme only
- security depends on difficulty of computing discrete logarithms
- variant of ElGamal & Schnorr schemes

DSA Key Generation

- shared global public key values (p,q,g):
  - choose 160-bit prime number q
  - choose a large prime p with $2^{L-1} < p < 2^{L}$
    - where L= 512 to 1024 bits and is a multiple of 64
    - such that q is a 160 bit prime divisor of $(p-1)$
  - choose $g = h^{(p-1)/q}$
    - where $1 < h < p-1$ and $h^{(p-1)/q} \bmod p > 1$
- users choose private & compute public key:
  - choose random private key: $0 < x < q$
  - compute public key: $y = g^{x} \bmod p$

DSA Signature Creation

- to **sign** a message M the sender:
  - generates a random signature key k, $k<q$
  - k must be random, be destroyed after use, and never be reused
- then computes signature pair:
  - $r = (g^k \bmod p) \bmod q$
  - $s = [k^{-1}(H(M) + xr)] \bmod q$
- sends signature (r,s) with message M

DSA Signature Verification

- having received M & signature (r,s)
- to verify a signature, recipient computes:

$w = s^{-1} \bmod q$

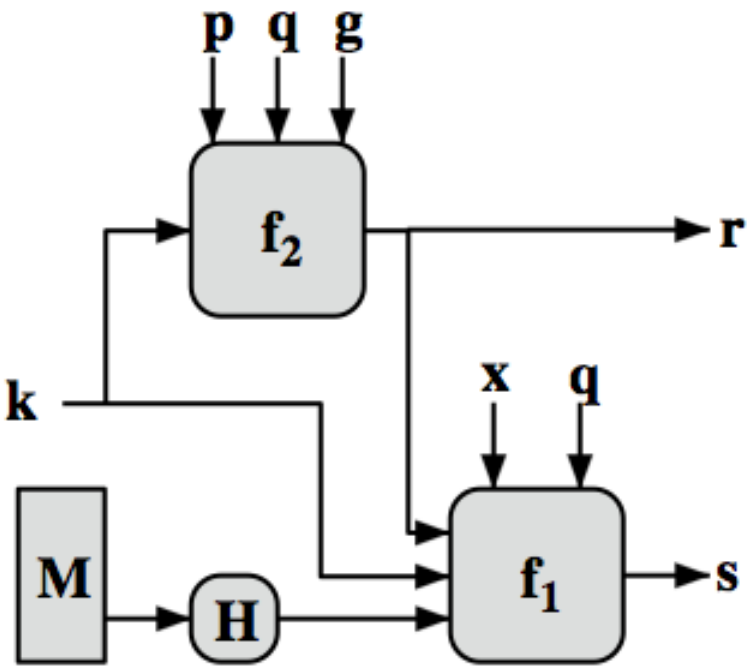
$u1 = [H(M)w] \bmod q$

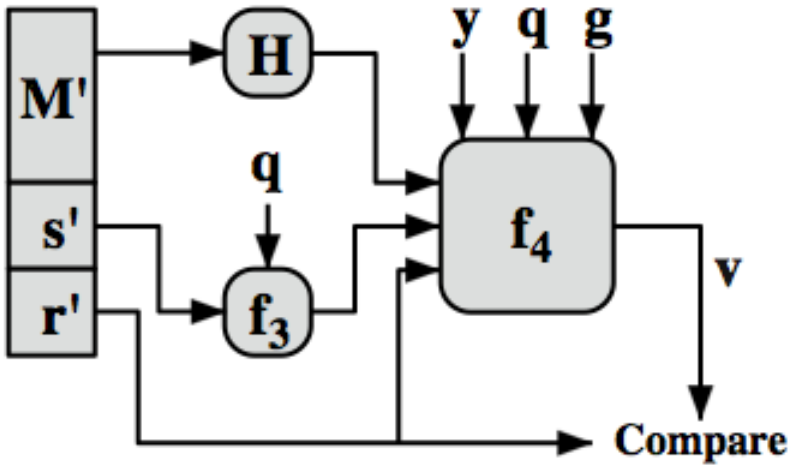
$u2 = (rw) \bmod q$

$v = [(g^{u1}\ y^{u2}) \bmod p] \bmod q$

- if v=r then signature is verified

DSS Overview





$s = f_1(H(M), k, x, r, q) = (k^{-1} (H(M) + xr)) \bmod q$

$r = f_2(k, p, q, g) = (g^k \bmod p) \bmod q$

(a) Signing

$w = f_3(s', q) = (s')^{-1} \bmod q$

$v = f_4(y, q, g, H(M'), w, r')$

$= ((g^{(H(M')w) \bmod q} y^{r'w \bmod q}) \bmod p) \bmod q$

(b) Verifying

Next Week

1. Privacy

Chapter 24, Legal and Ethical aspects,
Section 24.3 Privacy

References

1. W. Stallings. "*Cryptography and Network Security*", Global Edition, Pearson Education, 2017.

2. W. Stallings, "Cryptography and Network Security" Official Slides.

3. L. Brown "Cryptography and Network Security" Lecture Slides accompanying the textbook.