



WK01-03: Crypto Building Blocks

Securing Fixed and Wireless Networks, COMP4337/9337

Never Stand Still

Sanjay Jha, Nadeem Ahmed

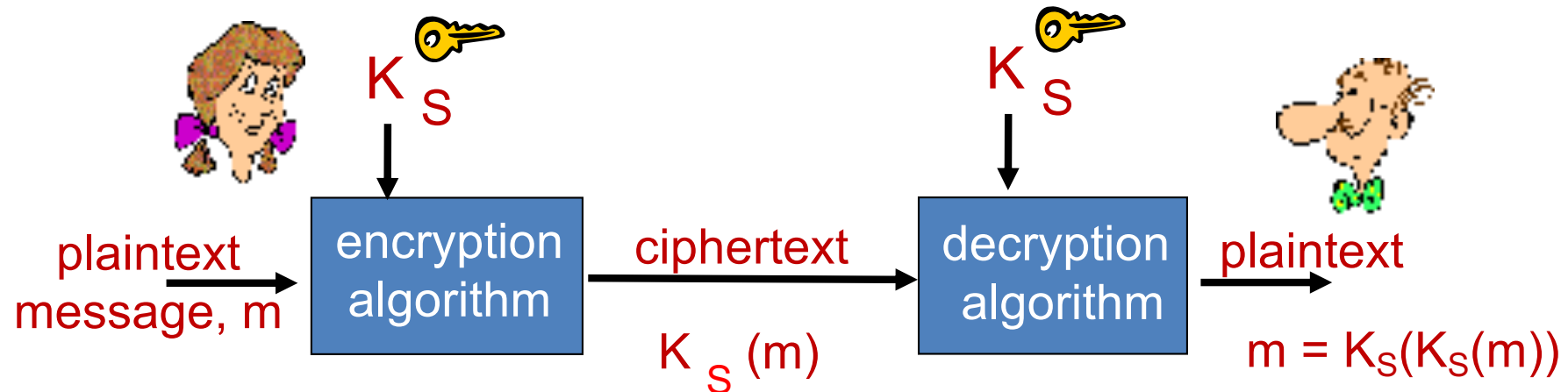
Today's Agenda

- Learn Crypto building blocks for Security Fundamentals
 - Secret Keys – Asymmetric/Symmetric
 - Integrity: Message Digests (Hash)
 - Integrity and Authentication: HMAC
 - Non-Repudiation: Digital Signature

Security Fundamentals (recap)

- *confidentiality*: only sender, intended receiver should “understand” message contents
- *authentication*: sender, receiver want to confirm identity of each other
- *message integrity*: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection
- *non-repudiation*: *no one (including the sender) can deny that message was sent by the sender*
- *access and availability*: services must be accessible and available to users

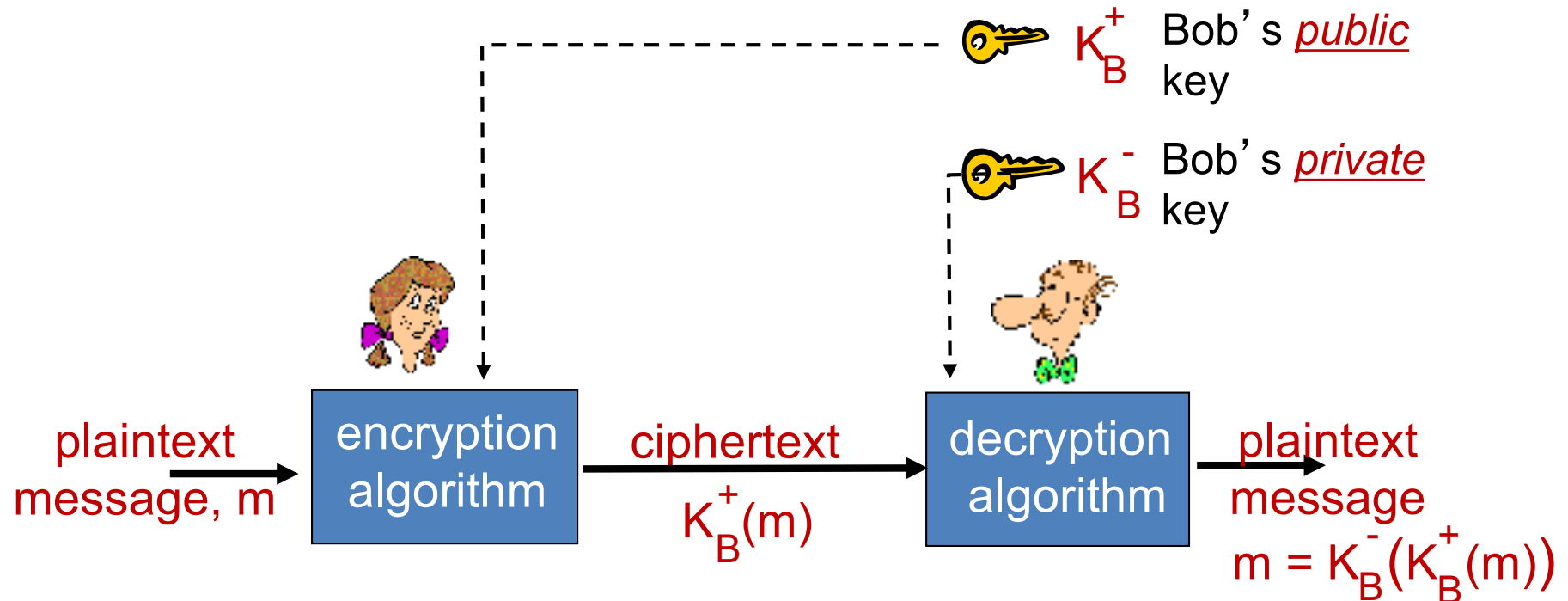
Symmetric Key Cryptography



Symmetric key crypto: Bob and Alice share same (symmetric) key: K_S

Q: How do Bob and Alice agree on the key value?

Public Key Cryptography



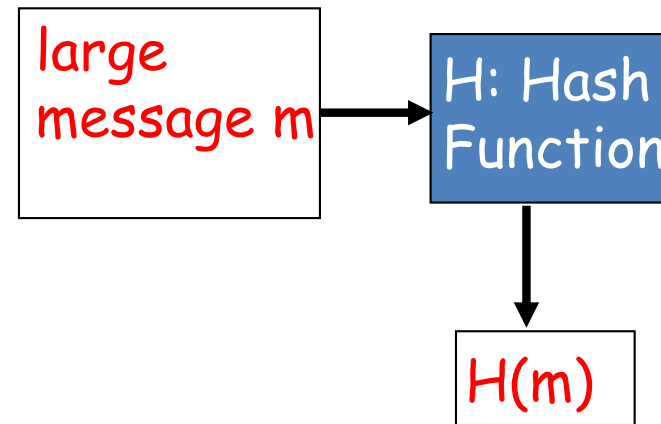
RSA: Rivest, Shamir, Adelson algorithm

Confidentiality vs Integrity

- Confidentiality: message private and secret
- Integrity: protection against message tempering
- Encryption alone may not guarantee integrity
 - Attacker can modify a message under encryption without learning what it is
- Public Key Crypto Standards (PKCS)
 - “RSA encryption is intended primarily to provide confidentiality... It is not intended to provide integrity”
- Both Confidentiality and integrity are needed for security

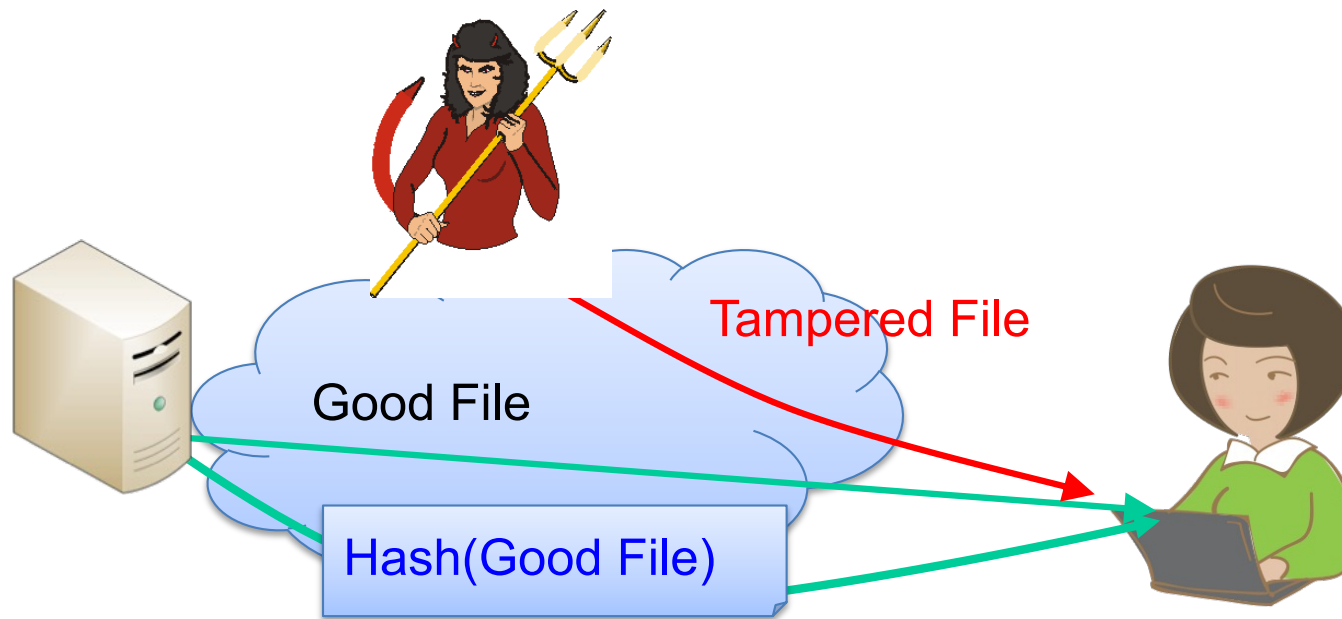
Hash/Message Digests (MD)

- Function $H()$ that takes as input an arbitrary length message and outputs a fixed-length string:
“message signature”
- The values returned by $H()$ are called hash values, hash codes, digests, or simply hashes.
- $H()$ is often called a “hash function”



- Desirable properties:
 - Easy to calculate
 - Irreversibility: Can't determine m from $H(m)$
 - Collision resistance: Computationally difficult to produce m and m' such that $H(m) = H(m')$
 - Seemingly random output

Example on Integrity



- Software distribution protection:
 - For a Good File and Hash(Good File), it is infeasible to find a Tampered File (containing rootkit or Trojan) such that $\text{Hash}(\text{Good File}) = \text{Hash}(\text{Tampered File})$

MD Randomness

Message digest of a hashing should have a random pattern

- Randomness: any bit in digest is “1” half the time
- Change input only one bit, and the hash will change half of the digest bits
- Diffusion: if hash function does not exhibit the avalanche effect to a slight change of input, then it has poor randomization, and thus a cryptanalyst can make predictions about the input, being given only the output

Poor Crypto Hash Example

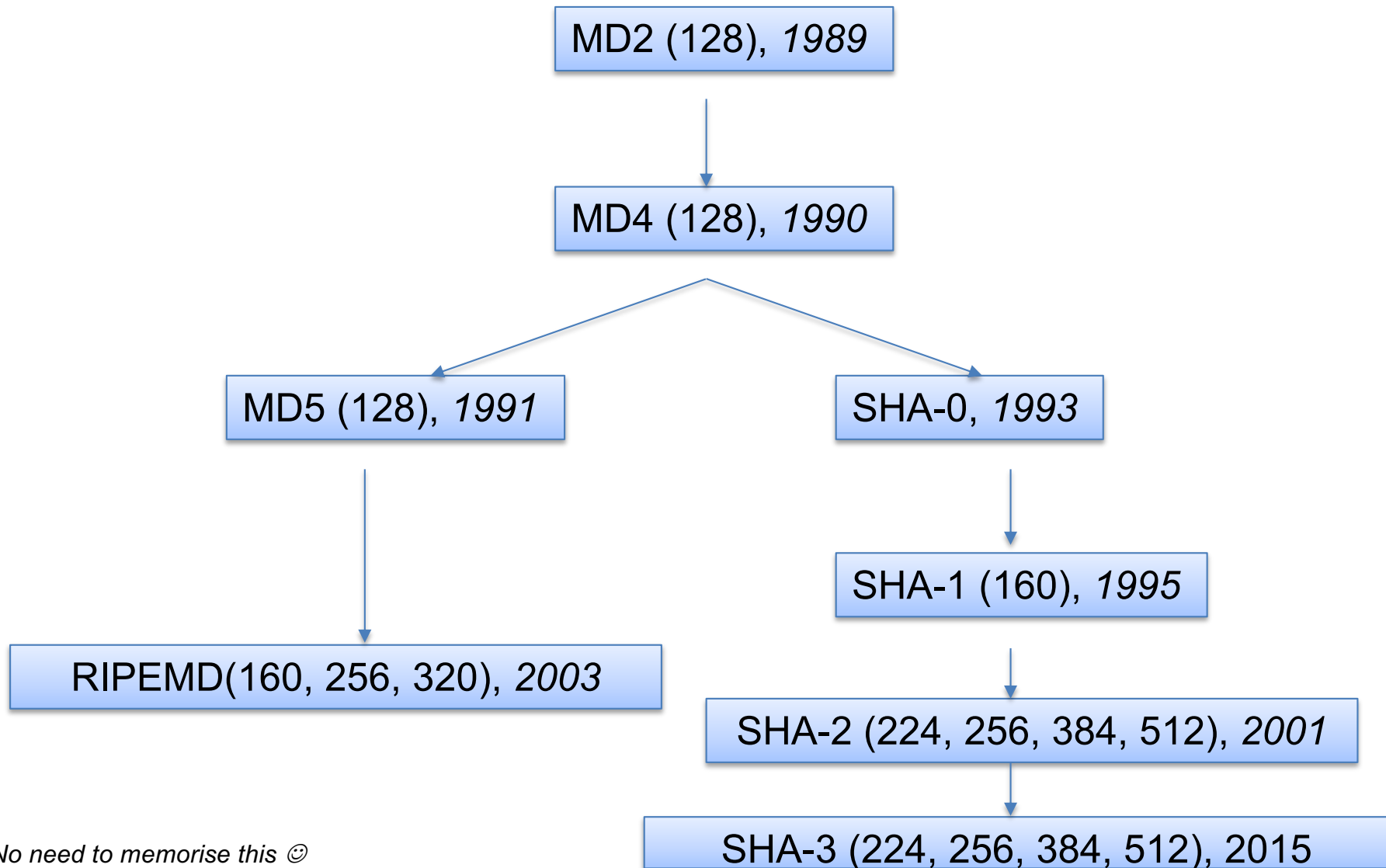
Internet checksum has some properties of hash function:

- produces fixed length digest (16-bit sum) of message
- is many-to-one

But given message with given hash value, it is easy to find another message with same hash value:

<u>message</u>	<u>ASCII format</u>		<u>message</u>	<u>ASCII format</u>
I O U 1	49 4F 55 31		I O U <u>9</u>	49 4F 55 <u>39</u>
0 0 . 9	30 30 2E 39		0 0 . <u>1</u>	30 30 2E <u>31</u>
9 B O B	39 42 D2 42		9 B O B	39 42 D2 42
<hr/>			<hr/>	
B2 C1 D2 AC		different messages but identical checksums!	B2 C1 D2 AC	

Standard Hash Function History



No need to memorise this ☺

Example: SHA-256

- Plaintext

Phishing Explained

Phishing scams are typically fraudulent e-mail messages appearing to come from legitimate sources like your bank, your Internet Service Provider, eBay, or PayPal, for example. These messages usually direct you to a fake web site and ask you for private information (e.g., password, credit card, or other account updates). The perpetrators then use this private information to commit identity theft.

Warning Signs

.....

URLs Don't Match - Place your mouse over the link in the e-mail message. If the URL displayed in the window of your browser is not exactly the same as the text of the link provided in the message, run. It's probably a fake. Sometimes the URLs do match and the URL is still a fake.

- Digest in BASE64

Tx9S2IwqlrGI7hhNJ4s5K7qiYt3PjQSD6vWH4QB17yg

Example: Diffusion in SHA-256

- Plaintext: only change the first letter from P to Q

Phishing Explained

Phishing scams are typically fraudulent e-mail messages appearing to come from legitimate sources like your bank, your Internet Service Provider, eBay, or PayPal, for example. These messages usually direct you to a fake web site

URLs Don't Match - Place your mouse over the link in the e-mail message. If the URL displayed in the window of your browser is not exactly the same as the text of the link provided in the message, run. It's probably a fake. Sometimes the URLs do match and the URL is still a fake.

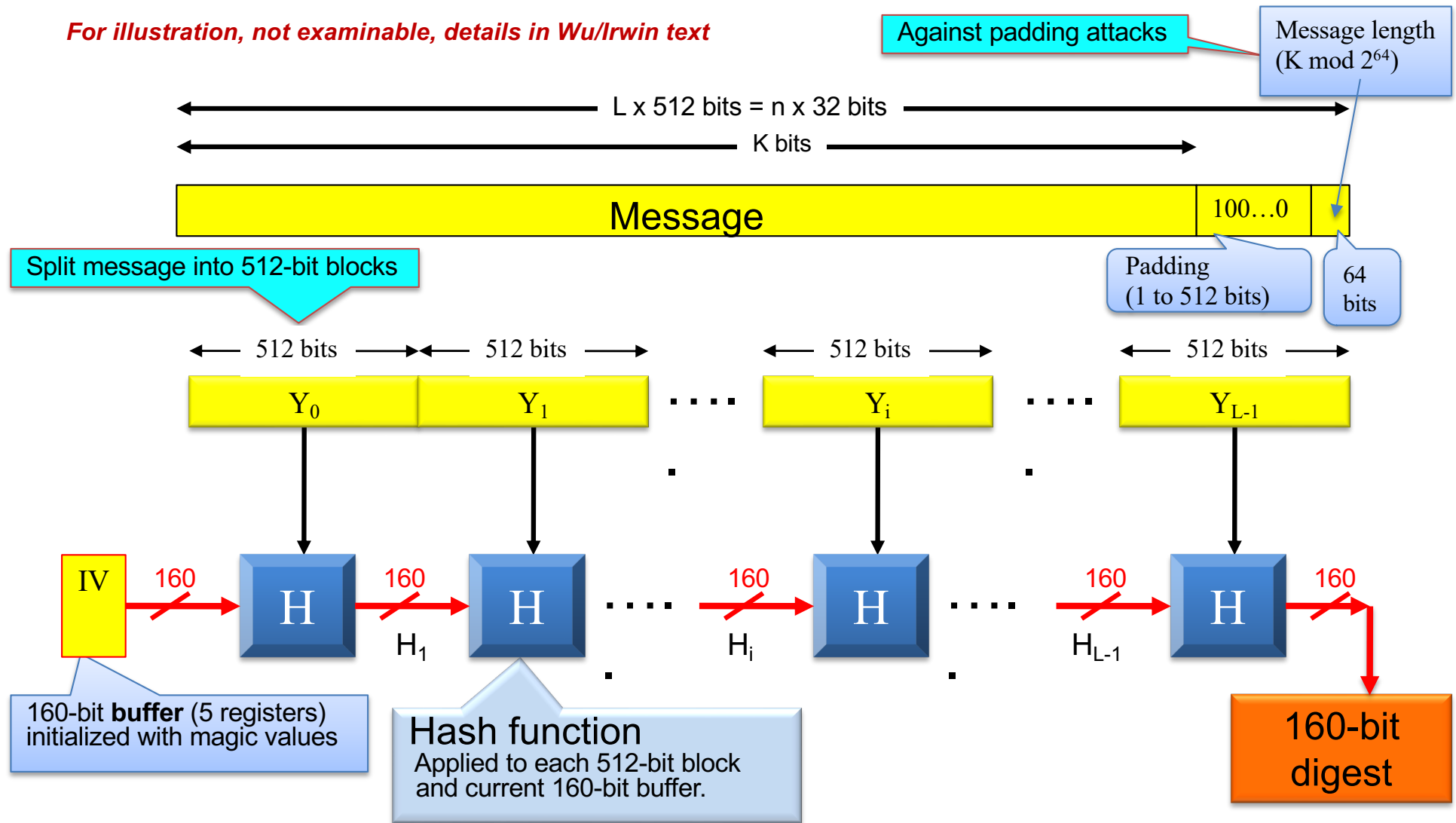
- Digest in BASE64 changes from

`Tx9S2IwqlrGI7hhNJ4s5K7qiYt3PjQSD6vWH4QB17yg =`

to

`AAHew/sW6W4X39EQq7ctFEb3PyHxka+T3D1UQJhDkw =`

Basic Structure of SHA-1



SHA properties

Algorithm Name	Max. Message Size (bits)	Block Size (bits)	Word size (bits)	Digest size (bits)	
SHA-1	2^{64}	512	32	160	
SHA-256	2^{64}	512	32	256	
SHA-384	2^{128}	1024	64	384	
SHA-512	2^{128}	1024	64	512	

No need to memorise this ☺, appreciate and consult table as needed. We will see some related practical issues in WEP protocol later.

Note: 160 bit digest is considered unsafe with fast computers.

If interested in crypto maths, read more on speed of execution of various schemes, which ones can be attacked by Quantum computers.

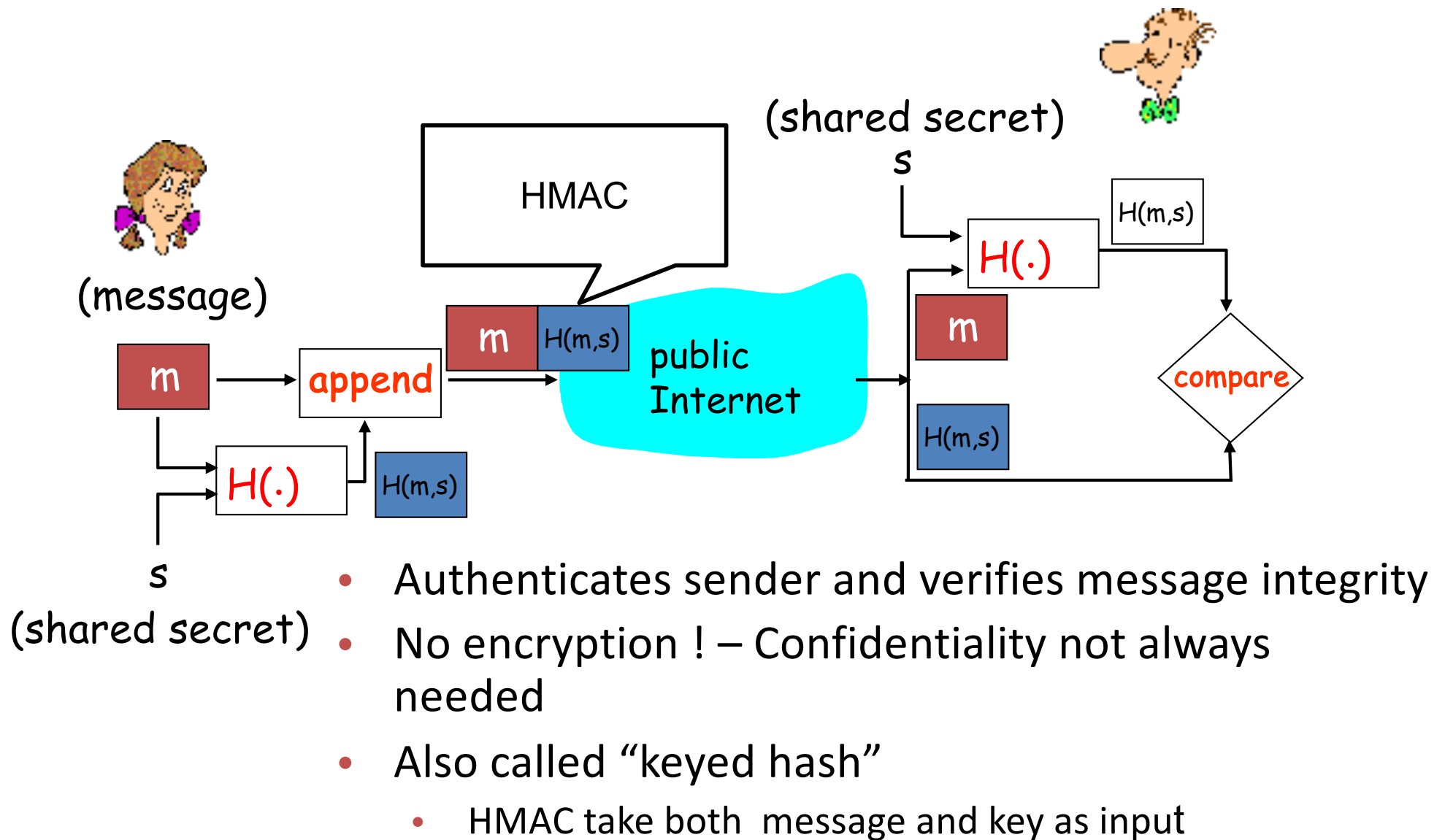
Attacks on Hash (Self study/no-exam)

- MD5 is one of the most widely used cryptographic hash functions
- Attack on MD5 to find collisions efficiently
 - About 15 minutes up to an hour computation time
 - Finding a collision for MD5 is easily feasible
- This attack is also able to break hash functions efficiently, including HAVAL-128, MD4, RIPEMD, SHA-0 and SHA-1
- SHA-1 near-collision attack needs $2^{57.5}$ hashes
 - Marc Stevens: HashClash on 21/2/2019
 - <https://marc-stevens.nl/p/hashclash/>
 - SHA-1 Chosen-prefix collision attack: $2^{77.06}$
- Ref:
 - Xiaoyun Wang, Hongbo Yu: How to Break MD5 and Other Hash Functions. EUROCRYPT 2005: 19-35
 - Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, Xiuyuan Yu: Cryptanalysis of the Hash Functions MD4 and RIPEMD. EUROCRYPT 2005: 1-18
 - Xiaoyun Wang, Yiqun Lisa Yin, Hongbo Yu: Finding Collisions in the Full SHA-1. CRYPTO 2005: 17-36
 - Xiaoyun Wang, Hongbo Yu, Yiqun Lisa Yin: Efficient Collision Search Attacks on SHA-0. CRYPTO 2005: 1-16
 - Hongbo Yu, Gaoli Wang, Guoyan Zhang, Xiaoyun Wang: The Second-Preimage Attack on MD4. CANS 2005: 1-12

Integrity vs Authentication

- Suppose Alice creates msg m , and calculates hash $H(m)$ using a hash function
- Alice send the extended message $(m, H(m))$ to Bob
- Bob upon receipt of this message independently calculates $H(m')$ from the message
- If $H(m) = H(m')$, message integrity verified.
- What if Trudy intercepts the original message, replaces it with a new message n and its hash $H(n)$?
- Authentication : Message Authentication Code (MAC)

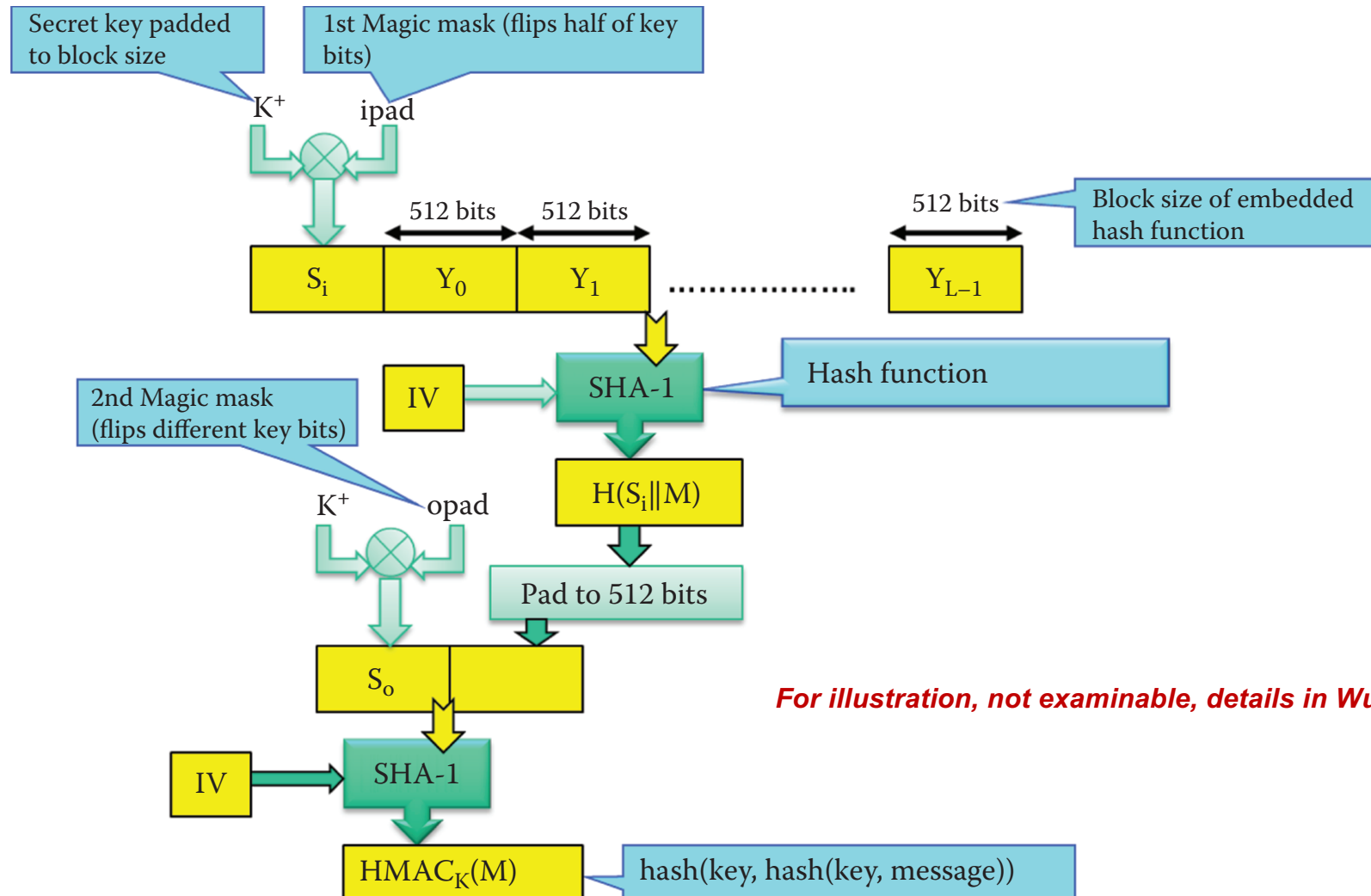
HMAC: Integrity and Authentication



HMAC

- FIPS 198, RFC 2104
 - Keyed-hash message authentication code: a message authentication code that uses a cryptographic key in conjunction with a hash function
 - Runs data and authentication key through hash function twice
 - Hashing is faster than encryption in software
 - Allows the use of any hash function, e.g., SHA-256, or SHA-512
 - No US export restrictions on HMAC and hash
- Used in TLS and IPSec (later weeks)

HMAC



Src: Irwin Wu: figure 20.6

Integrity of files using HMAC

Example:

- Use the following key in ASCII format.

TigersFootballWarEagleBCSChampion

- The corresponding Hex format for the key is

546967657273466F6F7462616C6C5761724561676C6542435
34368616D70696F

Example: HMAC-SHA-256

- Plaintext

Phishing Explained

Phishing scams are typically fraudulent e-mail messages appearing to come from legitimate sources like your bank, your Internet Service

.....

- HMAC Digest in BASE64

HadCmwGX9EGKBon0C+6XEInXCI8 =

- Only change the first char from Tiger to Siger in key, HMAC Digest in BASE64

Ar89wBq+6rxq4Eenho53TMiHvSw =

Hash, MAC and HMAC

- One way Hash Function: Doesn't take any secret key or its operation
 - Easy to compute (both hardware and software solutions possible)
- Message Authentication Code: MAC
 - Authentication code using a secret key
 - Can use block ciphers (next week), last bits of the cipher-text can be used as code. However, encryption software is slow, hence may be avoided.
- HMAC:
 - Specific type of MAC that is based on hash functions along with a secret key
 - Uses a keyed-hash function ($\text{HMAC} = \text{Hash}(\text{Key} + \text{Message})$).

Digital signatures

Cryptographic technique analogous to hand-written signatures:

- Sender (Bob) digitally signs document, establishing he is document owner/creator.
- **Verifiable, non-forgeable:** recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document


Digital signatures

Simple digital signature for message m :

- Bob signs m by encrypting with his private key K_B^- , creating “signed” message, $K_B^-(m)$

Bob's message, m

Dear Alice
Oh, how I have missed
you. I think of you all the
time! ...(blah blah blah)
Bob

 K_B^- Bob's private
key

Public key
encryption
algorithm

$m, K_B^-(m)$

Bob's message,
 m , signed
(encrypted) with
his private key

Digital signatures

- Suppose Alice receives msg m , with signature: $m, K_B^-(m)$
- Alice verifies m signed by Bob by applying Bob's public key K_B^+ to $K_B^-(m)$ then checks $K_B^+(K_B^-(m)) = m$.
- If $K_B^+(K_B^-(m)) = m$, whoever signed m must have used Bob's private key.

Alice thus verifies that:

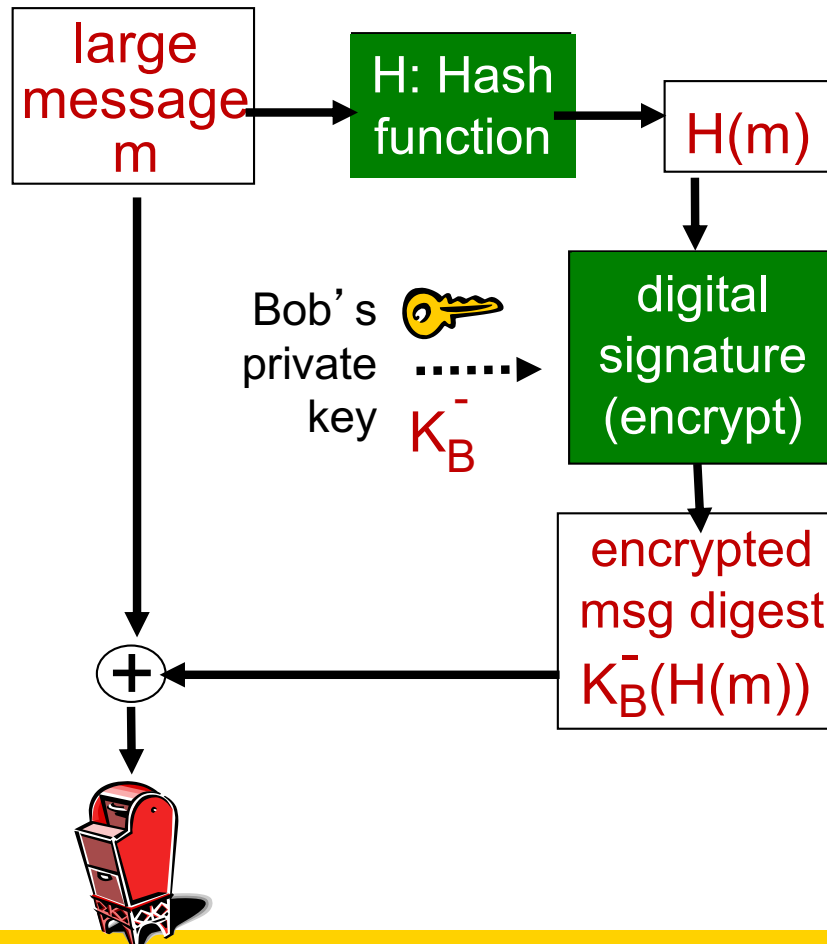
- Bob signed m
- no one else signed m
- Bob signed m and not m'

non-repudiation:

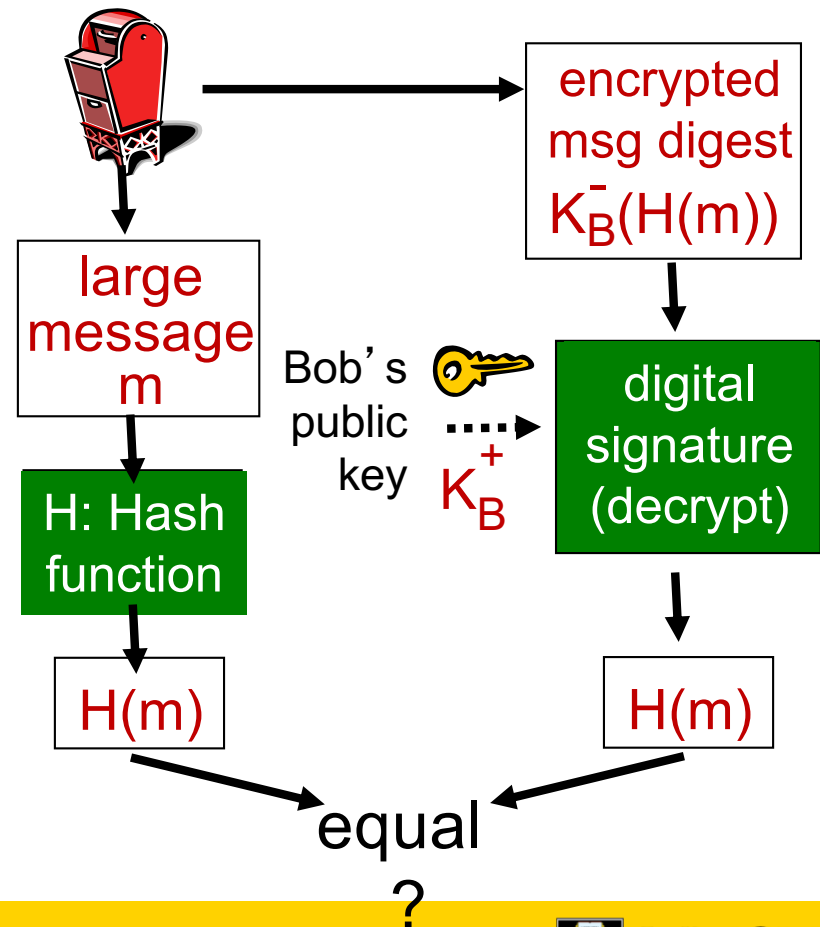
- ✓ Alice can take m , and signature $K_B^-(m)$ to court and prove that Bob signed m

Digital signature = signed message digest

Bob sends digitally signed message:



Alice verifies signature, integrity of digitally signed message:



Summary

- Lot of these crypto mechanism are building block for security protocols: both wired and wireless
- Lab1: Explore some of these techniques hands on.
- Acknowledgement:
 - Adaptation of foils from Kurose/Ross (revision from basic networking subject COMP3331)
 - Some material from Wu & Irwin book has lot more on various protocols and standards in chapter 20.
 - Details of algorithms beyond scope, you can consult many texts on crypto if interested.
 - Stallings: Network Security Essentials, Chapter 2 and 3 has good summary of Symmetric Encryption, Hash, HMAC, MAC etc.
 - Also refer to [Cryptography KA](https://www.cybok.org/) from <https://www.cybok.org/>