Information Security CS3002 (Sections BDS-7A/B, BSE-7A) Lecture 11

Instructor: Dr. Syed Mohammad Irteza

Assistant Professor, Department of Computer Science

25 (26) September, 2024

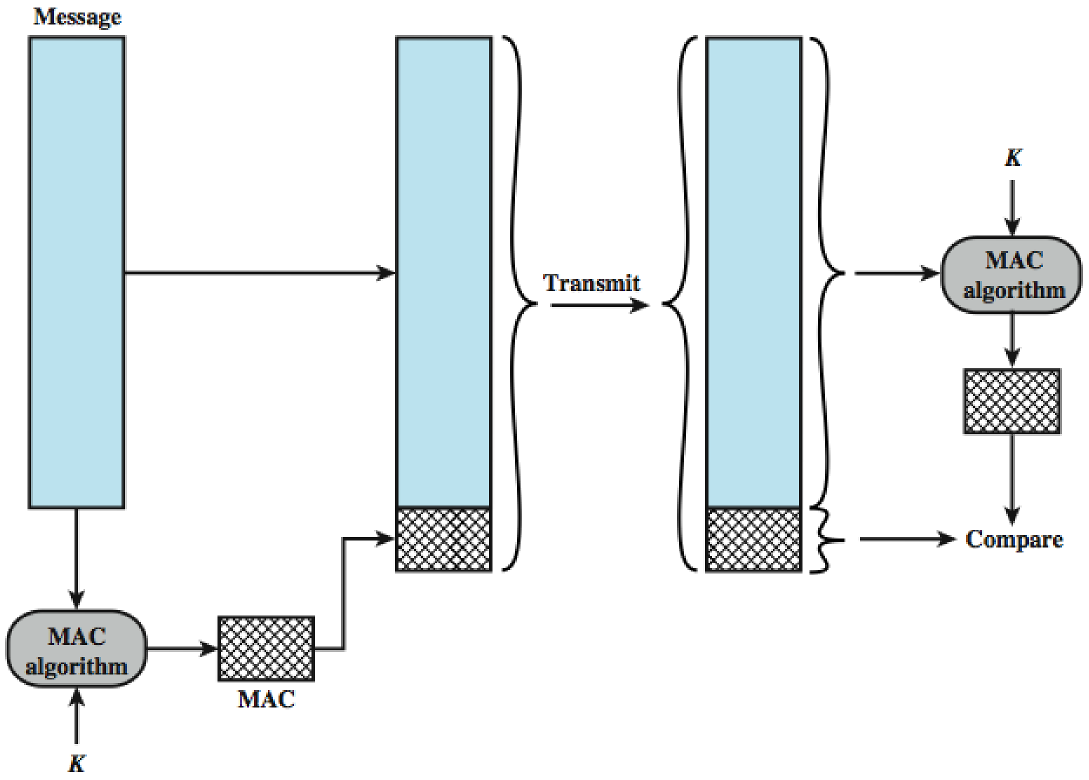
Message Integrity and Authenticity Protection

- Reference – Stallings, “Cryptography and Network Security”
  - Chapter 11 and 12

Message Authentication

- Protection against active attacks (e.g. alteration, falsification)
- Receiver verifies received message is authentic
  - contents unaltered
  - from authentic source
- Message Authentication Code (MAC): a small block of data appended to message, used for authenticity checking at receiver
  - Note the difference from <u>checksum</u> which only protects against accidental data errors

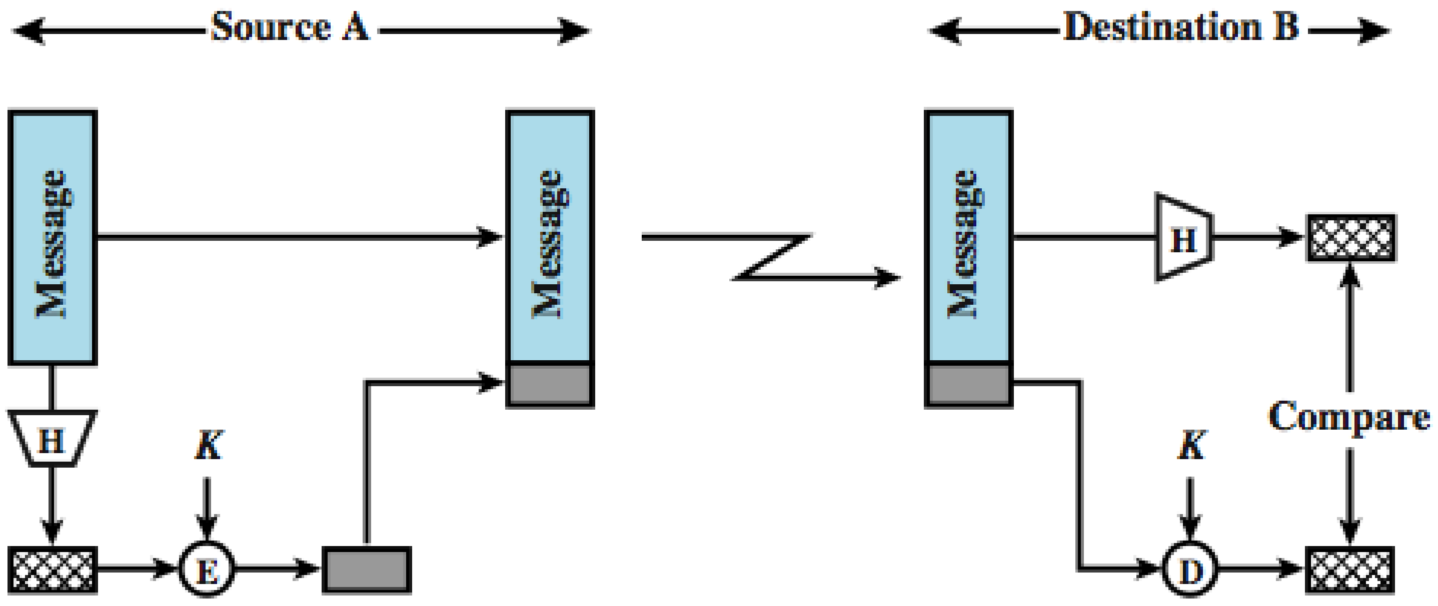
Message Authentication Code



MAC algorithm will always have a secret (K) as input.
Checksums do not require any input other than data.

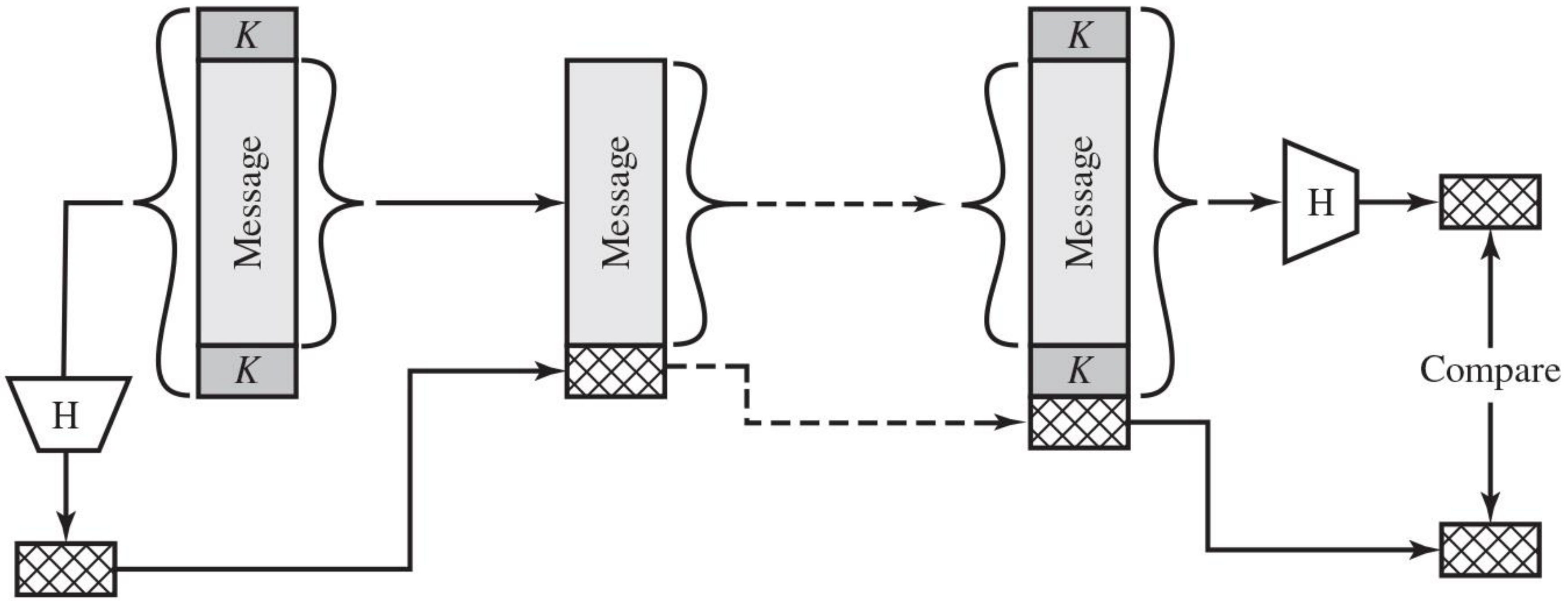
Hash-based MAC

- can also combine encryption with hash functions
  - First compute the hash of data and then encrypt the result with secret key



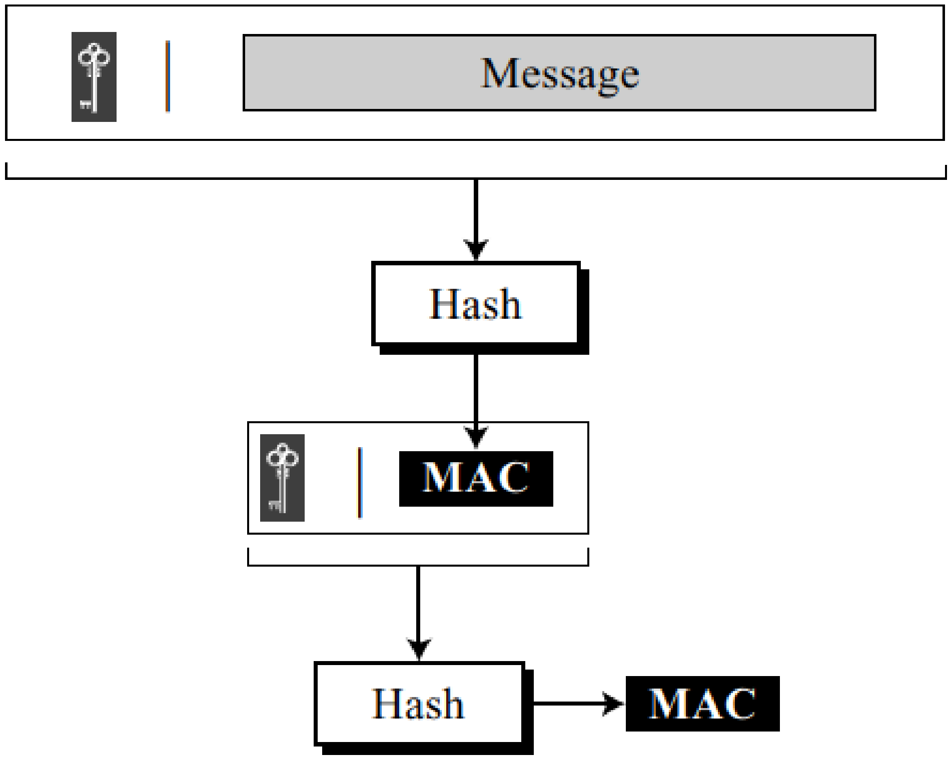
Prefix-Postfix MAC

- can also skip encryption altogether and just use hash functions
  - prepend/append/bracket the data with secret key before hashing



Nested hashing MAC

- Hash-based MACs depend on the security of hash function (discussed later). If the hash function is not strong enough, we can **apply hashing twice**



Firstly, key is prepended to message and hashed to get an intermediate MAC.

Then the process is repeated once more to get final MAC.

HMAC: Nested hashing MAC

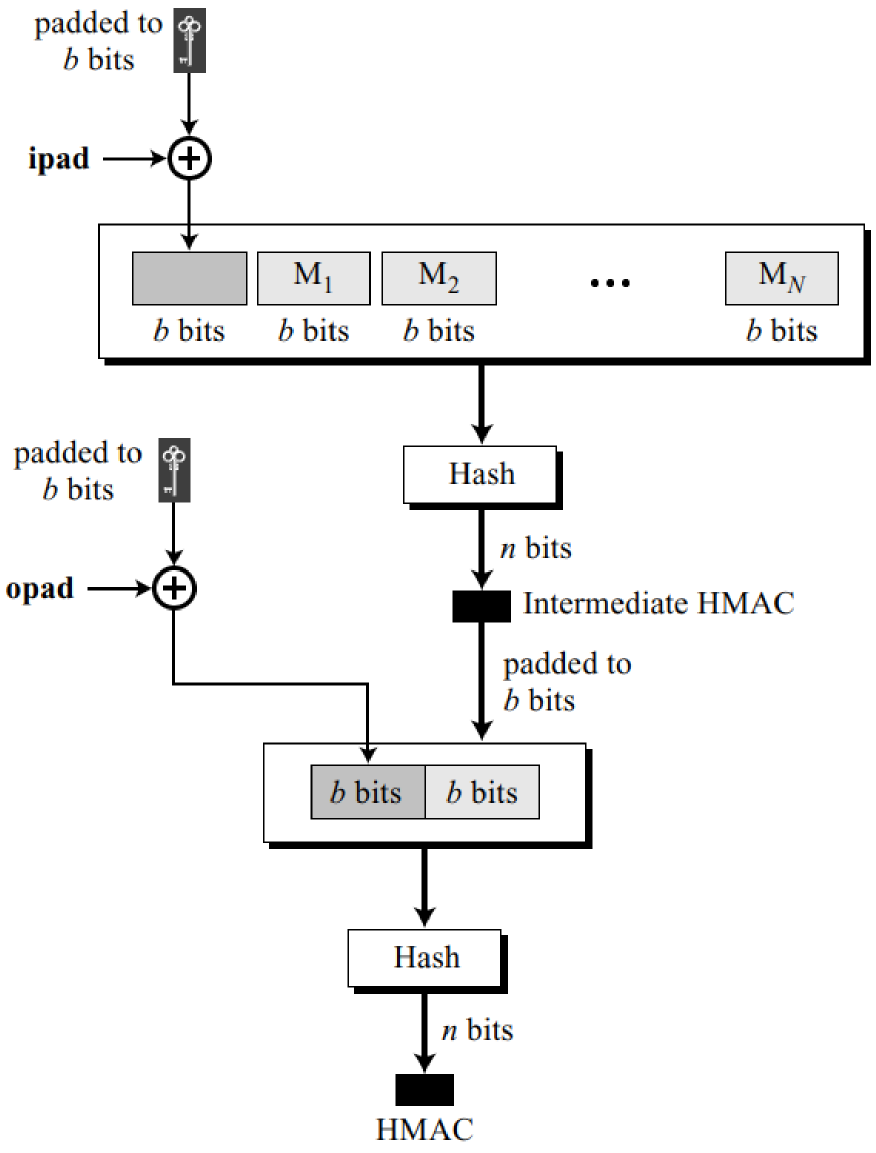
- NIST has standardized a variant of nested hashing based MAC algorithm (here).
- It needs the Message and a Secret Key as inputs

1. The message is divided into N blocks, each of b bits*.
2. The secret key is padded (extended) with 0's to create a b-bit key.
3. Padded key is XORed with a constant called **ipad** (inner pad) to create a b-bit block. The value of ipad is bit sequence 00110110 (0x36) repeated $b/8$ times.
4. The resulting block is prepended to the N-block message. The result is $N + 1$ blocks.
5. The result is then hashed to create an n-bit digest. We call the digest the intermediate HMAC.

*Typically b is large, like 64 bytes (512 bits) or 128 bytes (1024 bits)

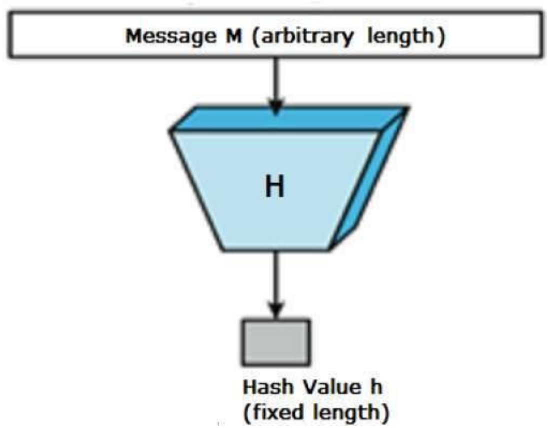
HMAC: Nested hashing MAC

6. The intermediate n-bit HMAC is extended with 0s to make a b-bit block.
7. Steps 2 and 3 are repeated by a different constant **opad** (outer pad), which is sequence 01011100 (0x5C) repeated $b/8$ times.
8. The result is then prepended to the block of step 6.
9. The result of step 8 is hashed with the same hashing algorithm to create the final n-bit HMAC.



Hash Functions

- These are one-way transformations
- Produce ‘message digests’ or ‘fingerprint’ of the input data
- Input to hash functions can be variable size, usually large number of bits.
- Output is small, fixed number of bits
- Collisions: Two different messages having the same hash
  - Since hashing is a many-to-one function, collisions are unavoidable



Cryptographic Hash Functions

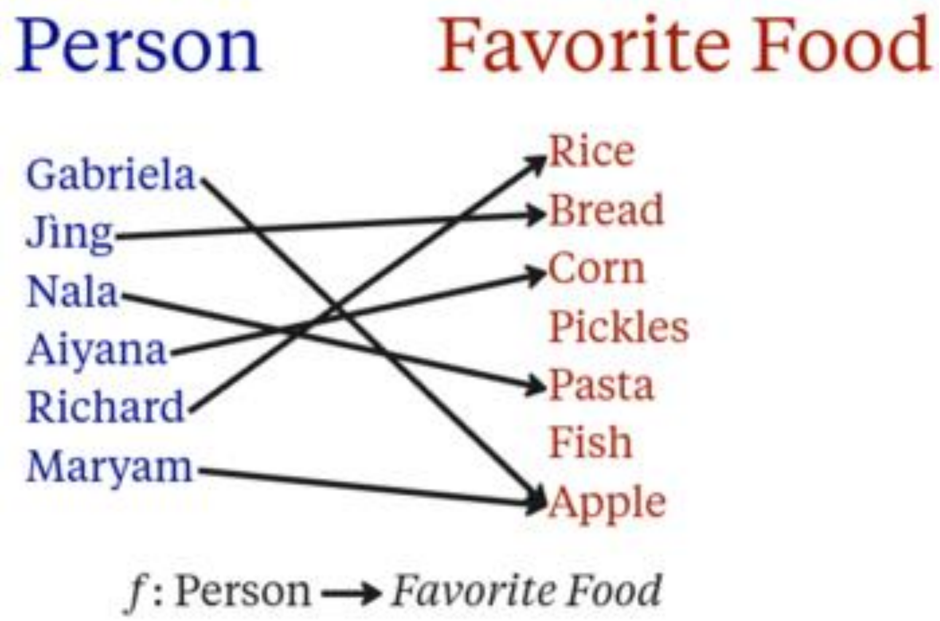
Not all hash functions can be used in security.

So called **cryptographic hash functions** need to have the following properties

1) Output is a (pseudo-)random combination of 1’s and 0’s distributed variably with a proportion of 50% each
   - A single bit change in input must change the output by roughly 50%
   - Flipping two bits in the input, the bits flipped in the output will be totally unrelated to which bits would flip if you just changed the bits one by one.

Aside: Images and Pre-images

- Consider the following many-to-one function:



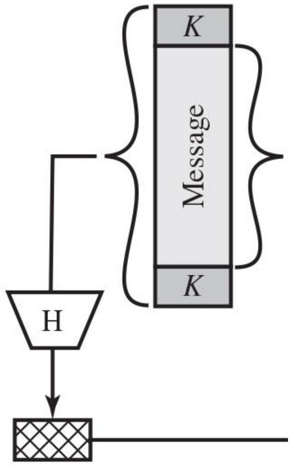
Maryam's *image* is Apple

Preimage of Apple is {Gabriella, Maryam}

Cryptographic Hash Functions

2) Pre-image Resistance (one-way property)

- Given a hash function h and a digest y, it must be extremely difficult for Eve (attacker) to find any message, M', such that $y = h(M')$.
- In other words, it should be virtually impossible to reverse the hashing.
- Otherwise, an attacker could discover a secret value K, that was used to generate a MAC.



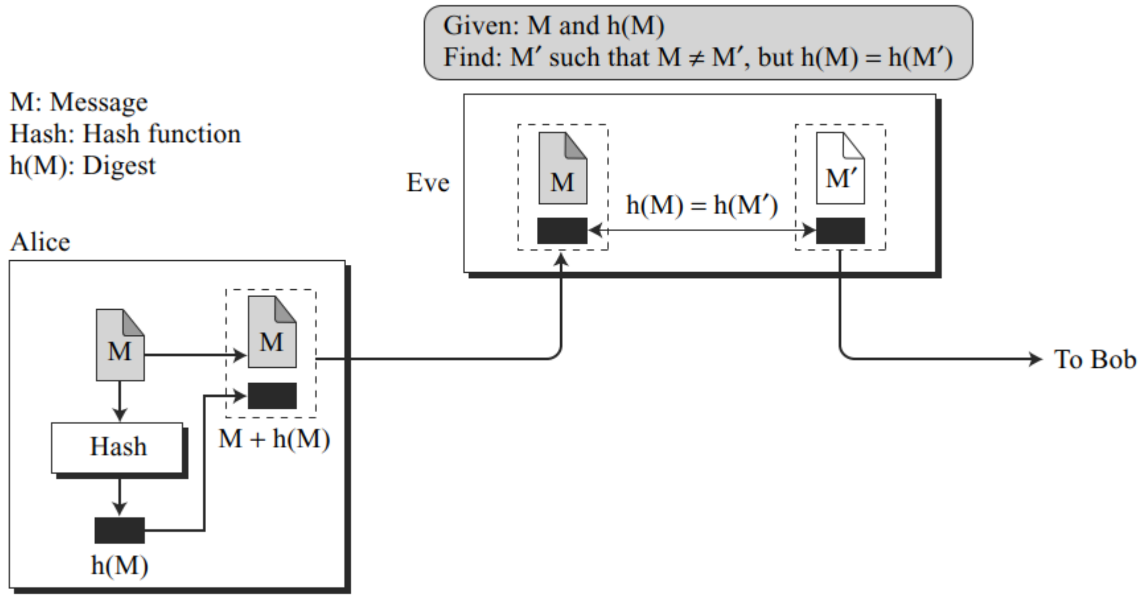
Analogy

In a class of students, find one who has a specific birthday (say 25^{th} Sep)

Cryptographic Hash Functions

3) Second Preimage Resistance

- Given a specific message and its digest, it must be extremely difficult to create another message with the same digest.



Without 2nd preimage resistance, Eve can silently replace a message with a forged one.

Analogy

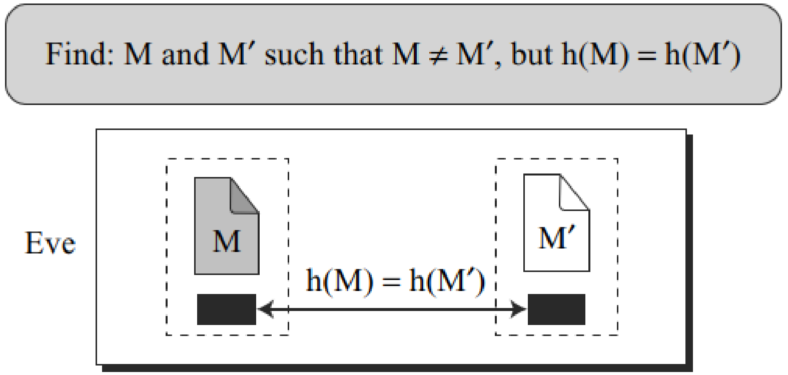
In a class, there is a student Ali, with birthday 25th Sep. Find another one from the class, with the same birthday.

Cryptographic Hash Functions

4) Collision Resistance

- Eve cannot find two messages (from scratch) that hash to the same digest.

M: Message
Hash: Hash function
h(M): Digest



Analogy: In a class, find ANY two students with the same birthday.

For a class of only 23 students, probability of finding two students with same birthday is 50%. When class size increases to 70, you are almost sure (99.9%) to find two such students!

[This is called birthday paradox]

- This type of attack is much easier to launch than the previous kind (forgery). Why?

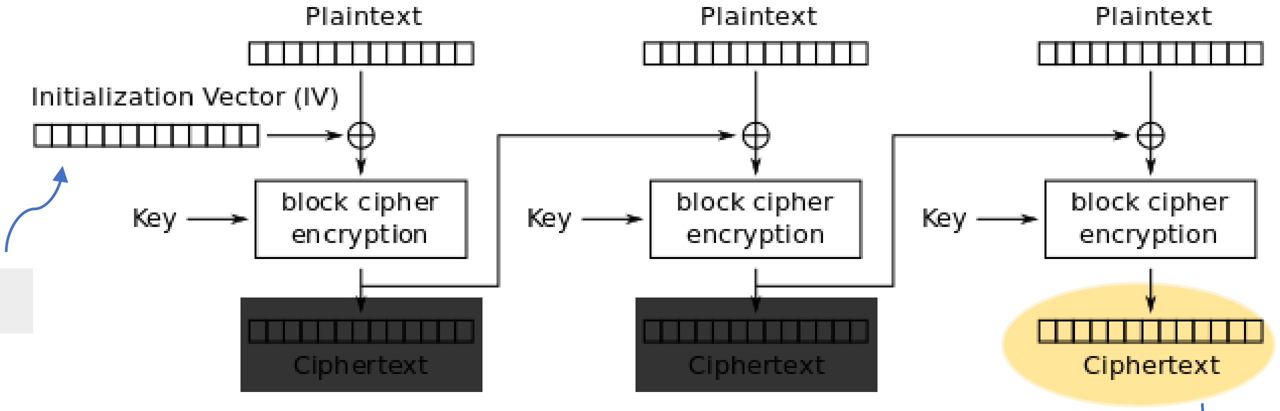
Acknowledgments

- Dr Ammar Haider, Assistant Professor, FAST-NU

Appendix

CBC MAC

- can use any (strong) symmetric encryption algorithm in the cipher block chaining mode
  - Because only sender & receiver have the key needed



set to zeros

Cipher Block Chaining (CBC) mode encryption

Final block output is dependent on all input blocks (i.e. the whole message), so it can act as a MAC