Aqsa Khalid
Lecturer
School of Computing

CS3002 Information Security

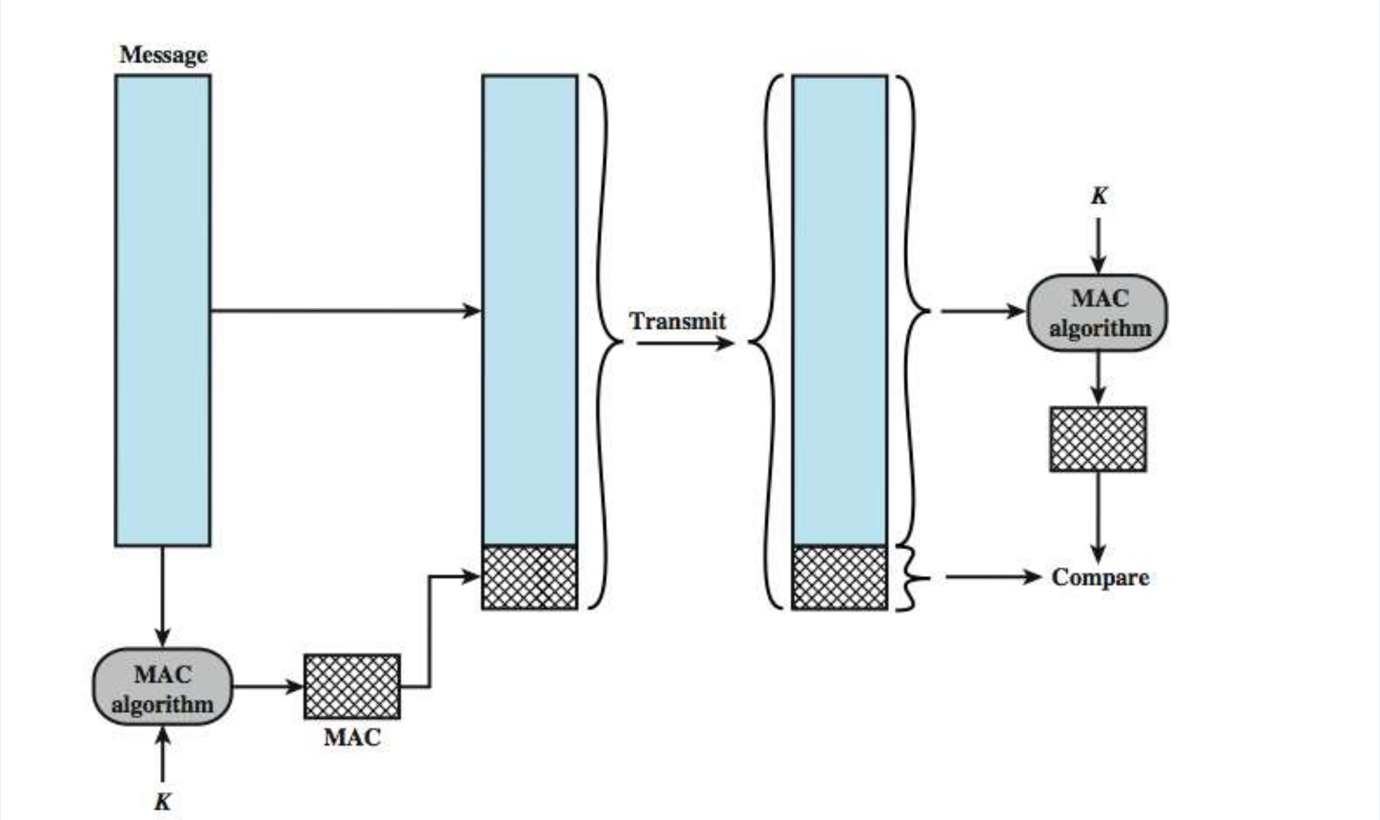
Message Integrity & Authenticity Protection

Reference: Stallings CNS chap 11, 12

Message Authentication

- Protection against active (e.g. alteration, falsification) attacks
- 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 checksum which only protects against accidental data errors

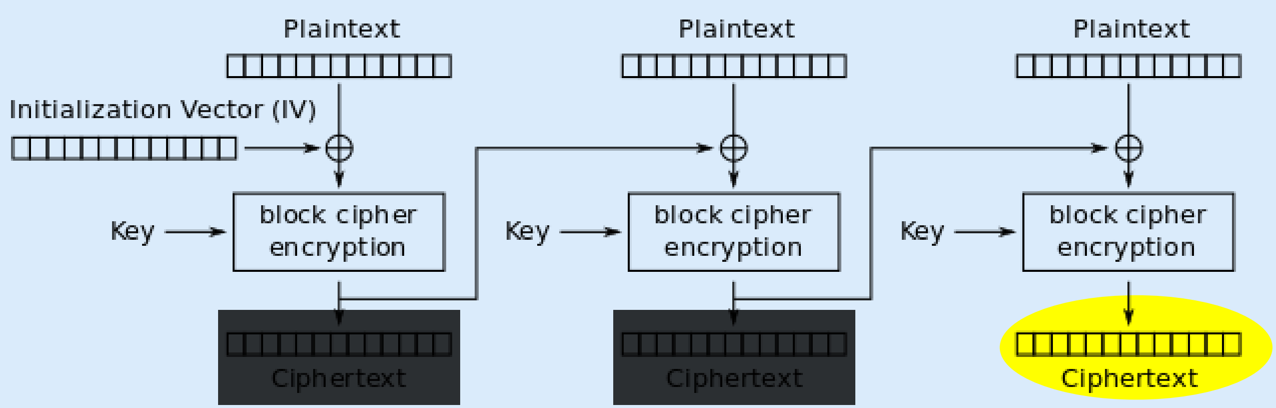
Message Authentication Code



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

CBC MAC

- can use symmetric encryption algorithms in the chaining (CBC) mode
  - Because only sender & receiver have the key needed

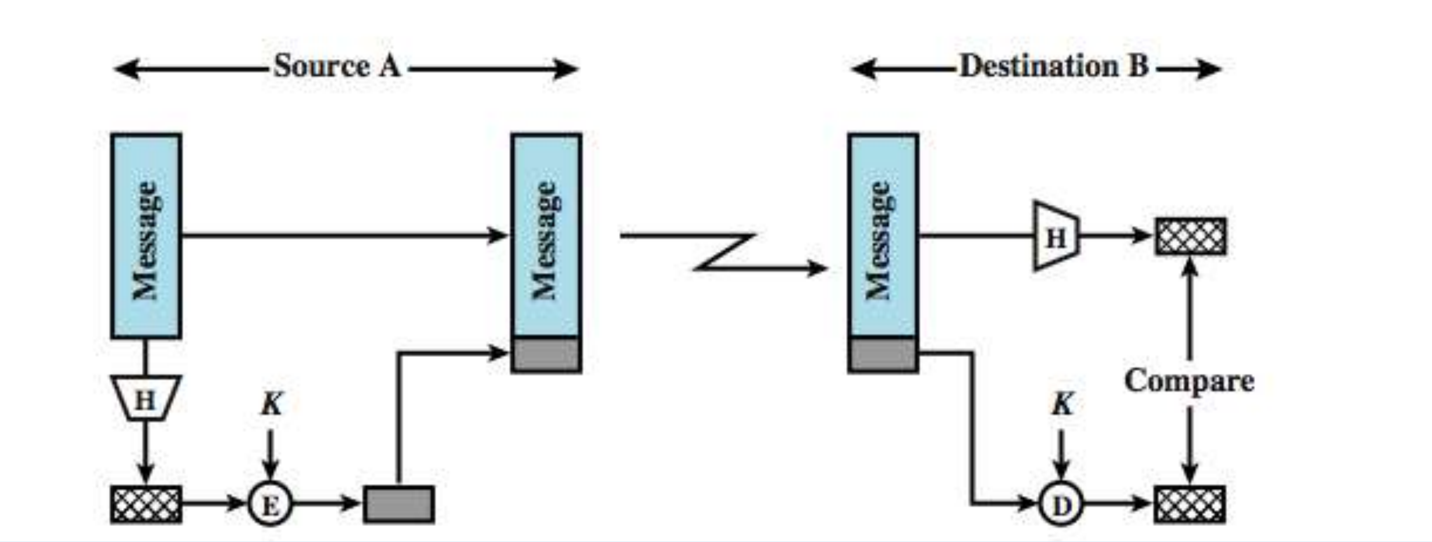


Cipher Block Chaining (CBC) mode encryption

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

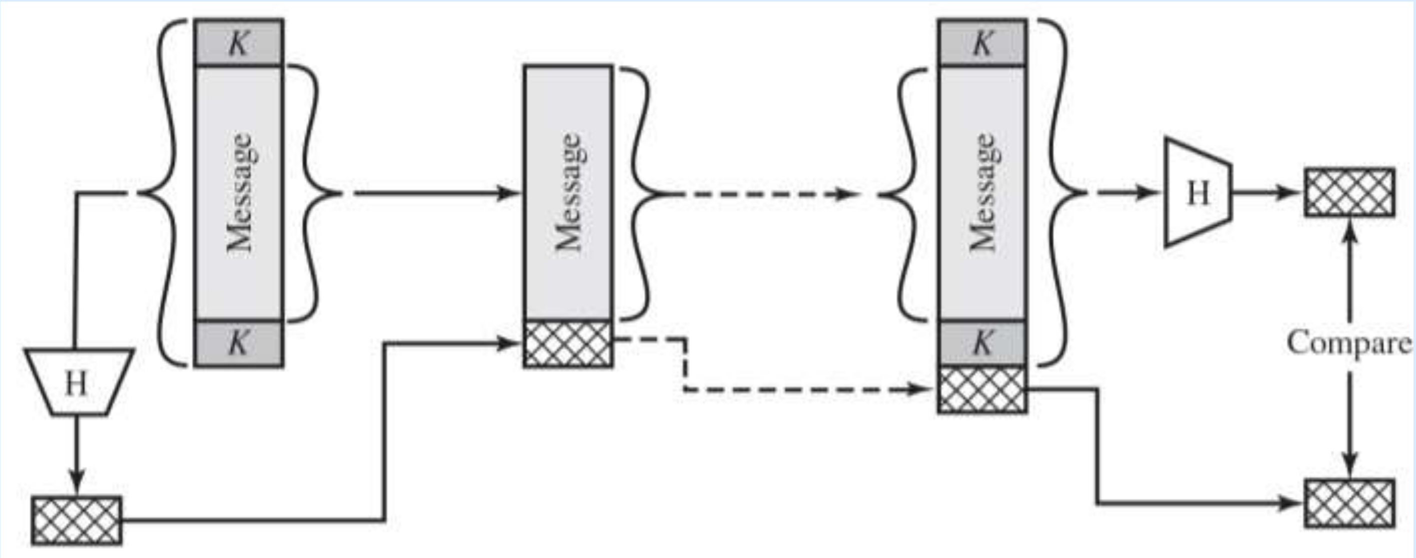
Hash based MAC

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



Prefix-Postfix MAC

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

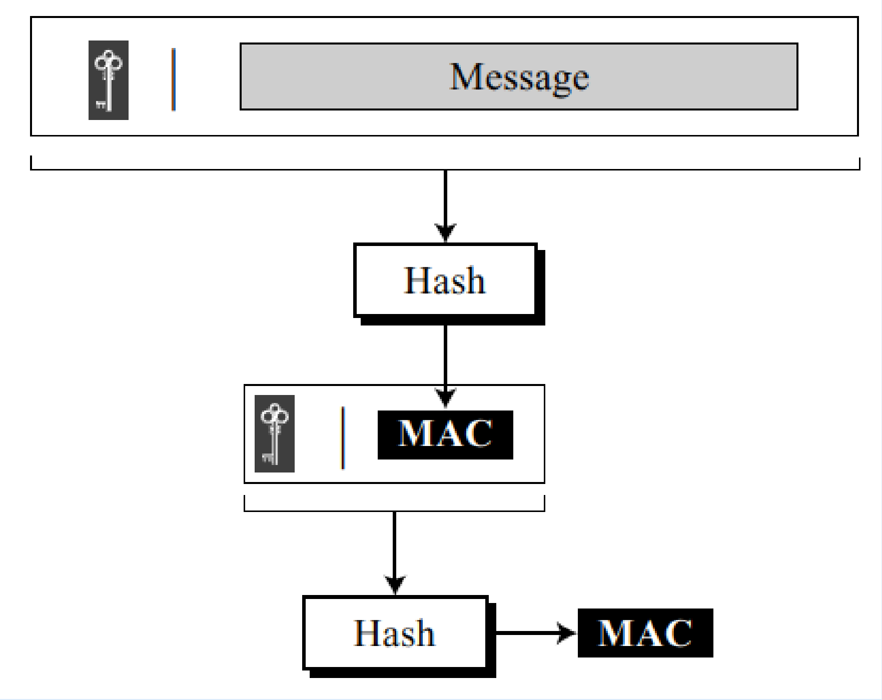


Nested hashing MAC

- For increased security, we can apply prefix-MAC 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

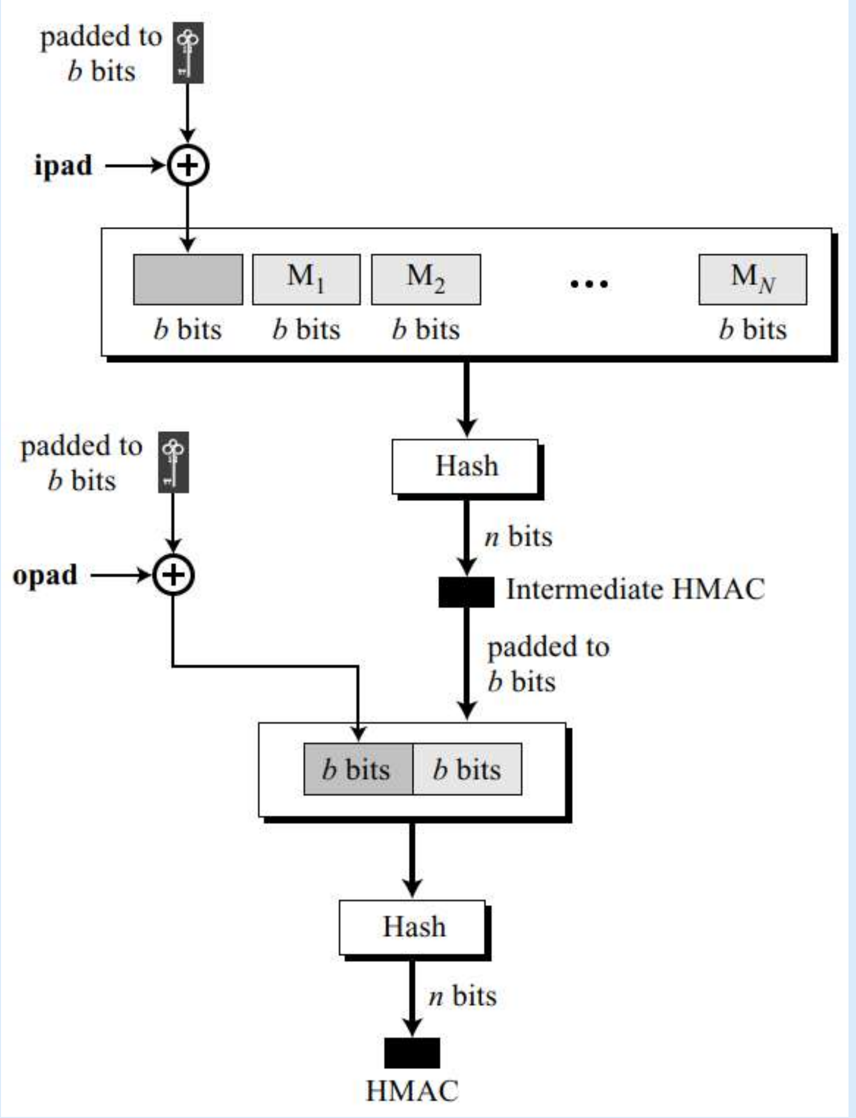
- NIST has standardized a variant of nested hashing based MAC algorithm.
- It needs Message and Secret Key as inputs

1. The message is divided into N blocks, each of b bits*.
2. The secret key is left-padded 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 512 or 1024 bits

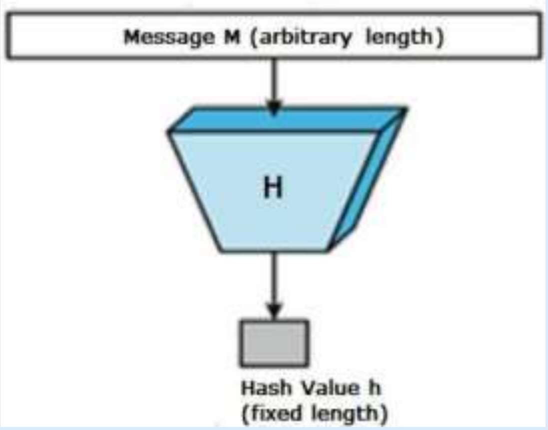
HMAC: Nested hashing MAC

6. The intermediate n-bit HMAC is left padded 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

- Also called 'message digests' or one-way transformations
- Produce '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

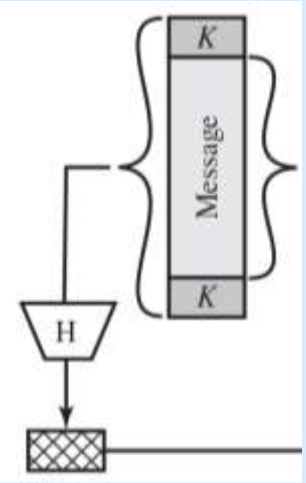
Not all hash functions can be used in security. So called **cryptographic hash functions** need to have the following properties

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.

Cryptographic Hash Functions

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

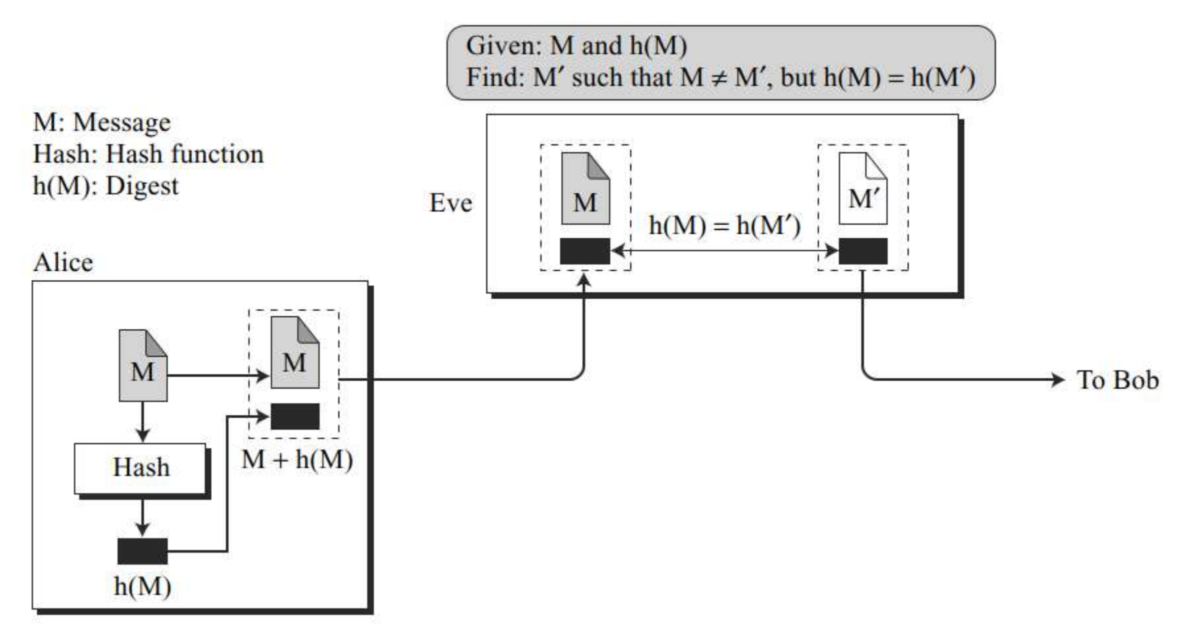
- Given a hash function h and a hashcode y, it must be extremely difficult for Eve (attacker) to find any message, M', such that $y = h(M')$.
- In other words, it should be virtually impossible to do reverse hashing.
- Otherwise, an attacker could discover a secret value K, that was used to generate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



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.

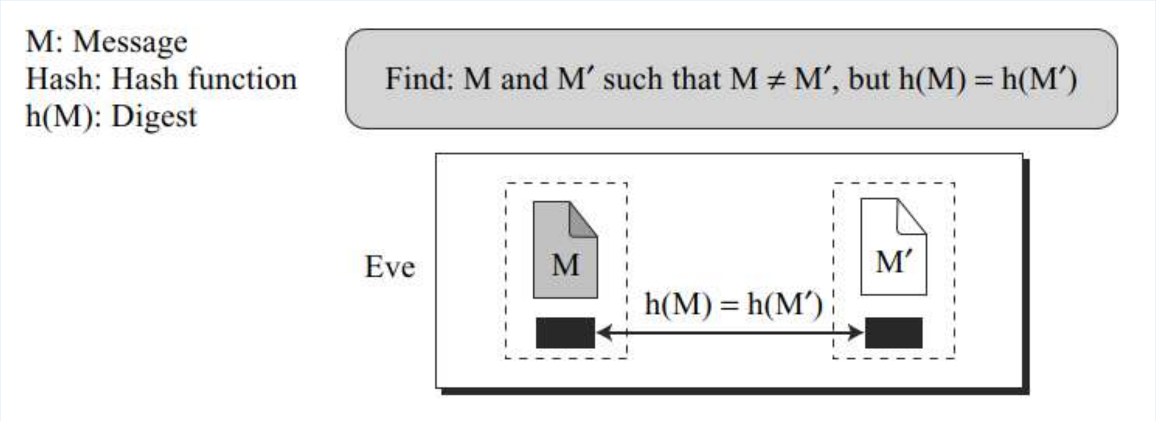


Without 2nd preimage resistance, Eve can forge a message.

Cryptographic Hash Functions

4) Collision Resistance

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



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

Cryptographic Hash Functions

- Collision resistance becomes relevant when one party generates a message for another party to sign. e.g.
    1. Alice asks Bob to prepare a cheque in his name, and she would sign it
    2. Bob finds two messages (cheques) with the same hash — one of which requires Alice to pay a small amount and one that requires a large payment.
    3. Alice signs the first message, and Bob is then able to claim that the second message is authentic.

Hash Functions Security

- Attacks against hash functions
  - Cryptanalysis: exploit logical weakness in algorithm to reverse the hashing
  - Brute force: trial many inputs. Strength is proportional to size of hashcode. For a hash function with n bit output, brute force strength is proportional to:

Preimage & 2nd preimage resistance	2^n
Collision resistance	$2^{n/2}$ (or $\sqrt{2^{\mathrm{n}}}$)

Commonly used hash functions

- MD5: older, 128-bit hash.
  - Proposed in 1992
  - Now considered insecure because it was proved to be not collision-resistant due to
- SHA-1: very widely used, 160-bit hash.
  - Now also considered vulnerable due mathematical weaknesses
- SHA-2: more recent, bigger size, more secure
  - multiple variations: SHA-256, SHA-384, SHA-512
- SHA-3: most recent

Collision generator: https://github.com/corkami/collisions

Digital Signatures

- Combines a hash with a PKC algorithm
- To sign
  - hash the data
  - encrypt the hash with the sender's private key
  - send data signer’s name and signature
- To verify
  - hash the data
  - find the sender’s public key
  - decrypt the signature with the sender's public key
  - the result of which should match the hash

Digital Signatures

Bob
signs a message

Alice
verifies the signature