Topic 5

Cryptographic Checksums

Understand message authentication codes (MACs) and cryptographic hash functions, and their applications to protect message integrity and authenticity

Overview

- Need for a Cryptographic Checksum
- Definitions
- Constructions
  - MAC (Message Authentication Code)
  - Hash Functions
  - HMAC
- Hash Function Applications
- Conclusion

Next Topic: Digital Signatures

Source: Stalling's book: chapters 11 and 12

Need for a Cryptographic Checksum

Why do I emphasise '*a certain degree*' here?

- Conventional (symmetric) encryption, $A \rightarrow B$: $E_K[M]$
  - Provides confidentiality, as ONLY A and B share K.
  - Provides ***a certain degree*** of origin authentication, as it could only come from A.
  - Does not provide signature, as
    - Receiver B could also generate the encryption.
    - Sender A could deny sending the message – repudiation of origin.
- Public-key (asymmetric) encryption, $A \rightarrow B$: $E_{KU_b}[M]$ (using B's public key)
  - Provide confidentiality.
  - Provides no origin authentication.

Why not?

Need for a Cryptographic Checksum

- Digital signing (cont.), $A \rightarrow B$: $M||E_{KR_a}[h(M)]$ (using A's private key)
  - Provides origin authentication and non-repudiation, as
    - Only A has KR_a, so signed item must have come from A.
    - Any party can use KU_a to verify the item.
    - Provided KU_a is trust-worthy, and the signature is dated.
  - Provide no confidentiality.

Is this good? $A \rightarrow B$: $M||E_{KR_a}[M]$

- Some of the cryptographic operations mentioned above could only provide message authentication and integrity protections provided that the message has some structures or is recognisable.
- We therefore need some redundancy (or check-value which is not forgeable) for the receiver to verify the message – this check-value is a cryptographic checksum.

Need for a Cryptographic Checksum

- Cryptographic checksum can be used to protect
  - Content authentication (= origin + integrity) for any kind of messages including unstructured.
  - Non-repudiation (digital signatures)
  - Anti-replay (i.e. achieve freshness)

- It is also attractive to the applications that require authentication and integrity without confidentiality
  - secure broadcast;
  - source code distribution.

Definitions: Digest Functions

- Given a message *M* of arbitrary length, a Message Digest function, *H,* produces a fixed-sized output, *h* (called a message digest, checksum, hash value, or fingerprint, of *M*), i.e. $h=H(M)$.
  - *h* should be a function of all the bits of M.

This is a message of any length ……..

$\downarrow$

H(x) — Fixed length value

$\downarrow$

01100100…1

This is a many-to-one mapping, so collisions are unavoidable, but we should make finding collisions as difficult as we can.

MD Functions - Requirements

- In addition to compression, such a function should also have the one-way and collision-resistance properties, i.e.
  - **Compression**
    - H can be applied to a block of data of any size, but produces a fixed-length output.
  - **One-way property (preimage resistant)**
    - $H(x)$ is easy to compute for any given x.
    - For any given h, it is hard to compute x such that $H(x)=h$.
  - **Weak collision resistance (2^{nd} preimage resistant)**
    - Given x, it is hard to find $y \neq x$ such that $H(y)=H(x)$.
  - **Strong collision resistance (collision resistance)**
    - It is hard to find two different messages, $x \neq y$, such that $H(y)=H(x)$.
- If H is strong collision resistant, then H is also weak collision resistant.

MD Functions - Requirements

- **Signature forgery if weak collision resistance property is not met.**
  - Assuming that A has sent a signed message M to B, *i.e.* $M||s$ where $s=E_{KRa}[H(M)]$ and KRa is A's private key;
    - An attacker intercepts A's signature and message;
    - The attacker finds another message M' with $H(M)=H(M')$;
    - The attacker now has your signature s on the message M'.
  - Think about the implication of this attack in real-life!

MD Functions - Requirements

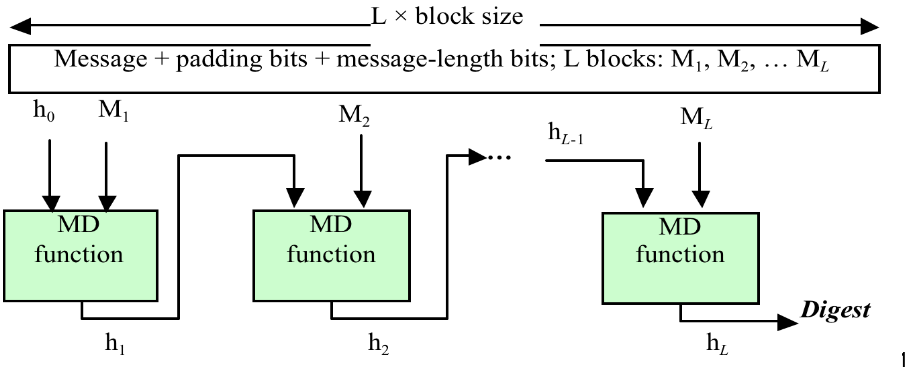
- **Repudiation if strong collision resistance property is not met.**
  - Assuming that A is to send a signed message M to B
    - A chooses two messages M and M' with $H(M)=H(M')$;
    - A signs M by generating signature $s=E_{KRa}[H(M)]$;
    - A sends B $\mathbf{M}||s$;
    - Later A repudiates this signature, saying it was really a signature on the message M'.
  - Think about the implication of this attack, if
    - The communication is for A to make an e-payment; and
    - M is an electronic cheque for £10.
    - M' is an electronic cheque for £1000.

Construction Methods

- Message Authentication Code (MAC) (with a built-in secret key)
  - Block cipher based
- Hash functions (without a built-in secret key)
  - Specifically designed hash functions
  - A hash value generated usually need to be protected by a secret
- HMAC
  - Use a hash function to construct a MAC function by concatenating a secret to the input of the hash function

Construction Methods

- Extension methods
  - Each MD function process a block of M; the output is the input for the next iteration

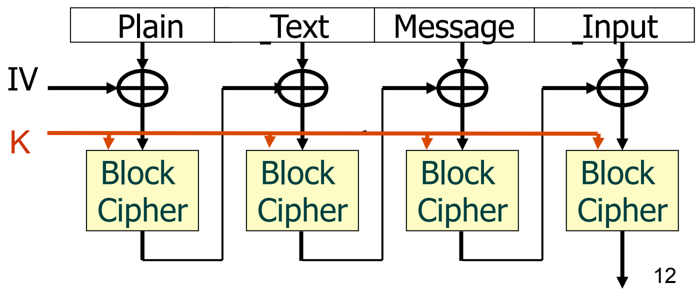


1

h_0 is a constant initial value, and the output of the last block *is the digest* of the entire message.

Message Authentication Code (MAC)

- A public function with a shared secret key that produces a fixed-length output, i.e. **MAC=$f_K(M)$.**
- Block cipher based, e.g. CBC-MAC.
  - Slow (re-keying of block ciphers)
  - Short digest length



12

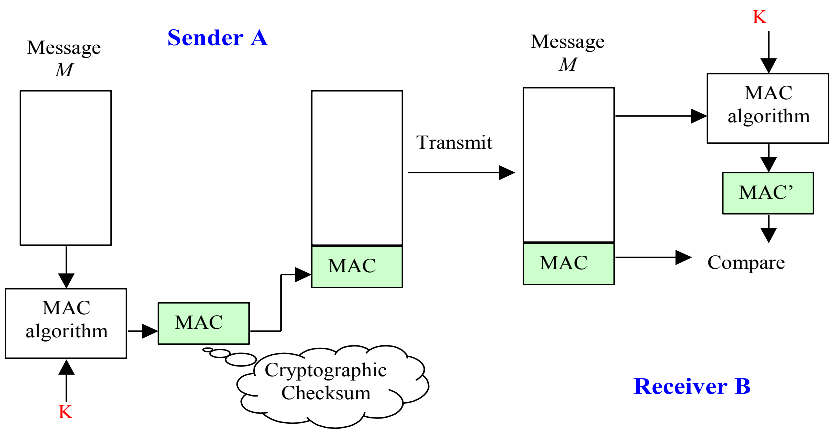
COMP38411: Cryptography and Network Security (Topic 5)

MAC in Operation

- **Sender**
  - uses K and a MACing function, f, to generate a checksum, $MAC=f_K(M)$.
  - then sends $M||MAC$, where || is concatenation of data items.
- **Receiver**
  - computes $MAC'=f_{K'}(M')$, where M' is the message received, and K' is receiver's copy of the key.
  - If $MAC=MAC'$, then the message has not been tampered with.

13

MAC in Operation



14

MAC in Operation

- If only A and B know the secret key K, and if MAC=MAC', then the receiver can be assured
  - the message has not been altered - integrity protection;
  - the message is from the alleged sender - origin authentication;
  - the message is of the proper sequence if the message includes a sequence number;
  - the message is fresh - *i.e,* not a replay
    - if the message includes a timestamp; or
    - a random number contributed (fully or partially) by B (the recipient).

(a) What is the implication of the first method,
(b) why the random number should be contributed by B?

Hash Functions – Commonly used hash functions

	SHA-1	SHA-256	SHA-384	SHA-512
Hash value size	160	256	384	512
Block size	512	512	1024	1024
Word size	32	32	64	64
Security	80	128	192	256

* All sizes are measured in bits;   *SHA = Secure Hash Algorithm

* Security refers to the fact that a birthday attack on a message digest of size n produces a collision with a work-factor of approx $2^{n/2}$.

HMAC

- **HMAC** constructs MAC by applying a message and key to a cryptographic hash function, in a nested manner, i.e.

  HMAC(K, M)= H[(K⊕opad) || H[(K⊕ipad) || M)]];

  where

  H = hash function such as SHA-1;

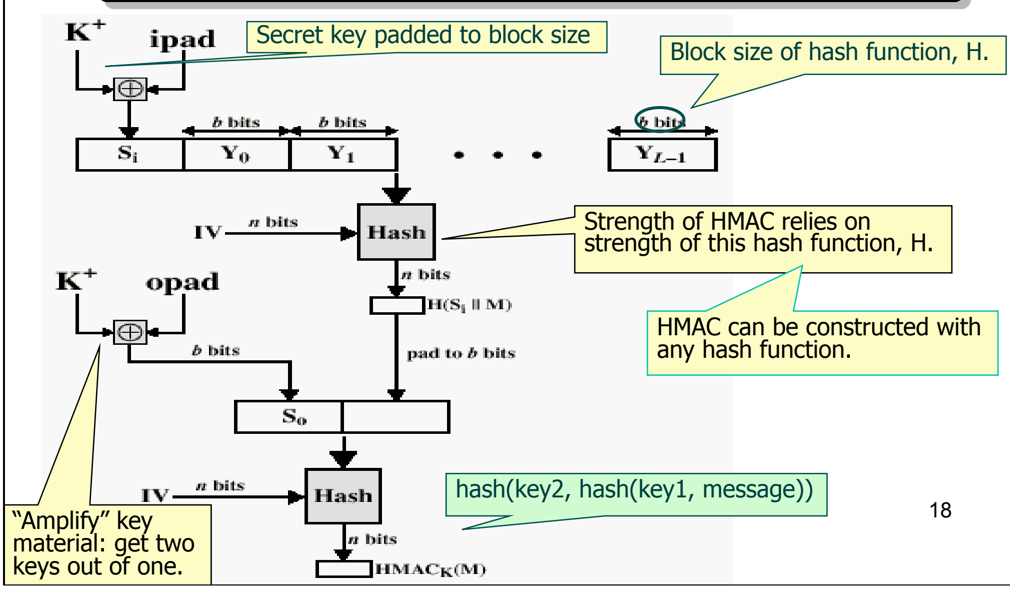
  ipad = a string by repeating the byte 0x36 (00110110) as often as necessary;

  opad = a string by repeating the byte 0x5c (01011100) as often as necessary;

  K = 512-bits (64-bytes) secret key (*if K is shorter, K is padded with zeros on the left so that the result is 64-bytes in length*).

- Used in many security packages, e.g. IP security and SSL/TLS.

HMAC Structure

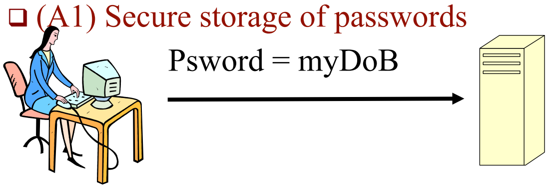


Hash Function Applications

- Secure storage of passwords
- Digital signatures
- Pseudo-random number generations
- Bit commitment (or coin flipping) problem
- Digital payment systems
- Digital right management systems
- etc

Topic 5 - A Quick Question

- For each of the following applications, you identify what property(ies) does the hash function need to have for it to be carried out securely?
- (A1) Secure storage of passwords



Password file:

User_A ***

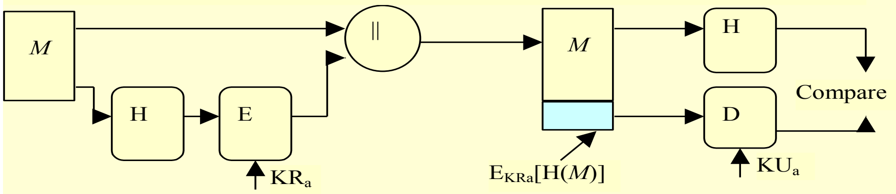
User_B ***

What should we put in there?
What if backup tape is stolen?
What property do we need?

- (A2) Protection against viruses
  - Software manufacturer wants to ensure that the executable file is received by users without modification.
  - They send out the file to users and publishes its hash value on an authentic website.

Topic 5 - A Quick Question (continue)

Integrity, authentication, and *non-repudiation* are provided.
*This is the essence of the digital signature technique.



- Digital signatures
  - One party can sign a message M, many parties can verify.
  - Contract signing, code signing, etc.…
  - Raw signature scheme only signs a few hundred (e.g. 160) bits.
  - What properties do we need?

Exercise 5 (a)

- **Generation of a keyed-hash message authentication code (HMAC) using CrypTool**
  - Create a file containing some plaintext
  - Choose a hash function
  - Select a HMAC variant
  - Enter a key (or keys, depending on the HMAC variant)
  - Generation of the HMAC (automatic) – you can access this facility via Menu: "Indiv. Procedures" \"Hash" \"Generation of HMACs".

Exercise 5 (b)

- **Investigating the sensitivity of hash functions to plaintext modifications using CrypTool**
  1. Select a hash function
  2. Modify characters in plaintext
- **For example:**
  - By adding a space after the word "CrypTool" in the example text, 50.6 % of the bits in the resulting hash value will change.
  - A good hash function should react highly sensitively to even the smallest change in the plaintext –"Avalanche effect" (small change, big impact).
- The facility is available via Menu: "Indiv. Procedures" \"Hash" \"Hash Demonstration".

Exercise 5 (c)

- In this exercise, you are asked to address the Coin Flipping Over the Telephone problem.

  (i) Assuming there is only one car, and Alice and Bob have to decide who can have this car (only one of them can have it, i.e. they cannot share it). Alice and Bob cannot see each other, and they do not trust each other. So they have decided to make a decision by flipping a coin over the telephone. Design a protocol to support this using a hash function.

  (ii) Identify any factors that you should consider to ensure the security of this protocol.

Exercise 5 (c) – flipping a coin (hint)

- Assuming: Alice and Bob agree that if the outcome is 1 then Bob takes the car, if it is 0 then the car goes to Alice.

- **Solution 1:**
  - Alice generates a random bit *b*: 0=heads, 1=tails.
  - Alice asks Bob: heads or tails?
  - Bob sends Alice his choice_B: 'heads' (or 'tails').
  - Alice compares *b* with choice_B: if b=choice_B, then outcome=1; if not, outcome=0.
  - Alice sends the comparison outcome to Bob.

What is wrong with this solution?

Conclusion

- Message encryption can not always provide assurance that the message has not been tampered with during its transit.
- Signing the complete message is often very expensive.
- A hash function is used to produce a *fingerprint* (also called *message digest*) of a file, message, etc, useful for message integrity and authentication as well as signatures (non-repudiation of origin) provided that the hash values are protected using proper cryptographic keys.
- A hash function is used in many applications, e.g. to prove possession of a secret without revealing the secret such as **UNIX password hash**.