Hashes, MACs & Authenticated Encryption

Tom Chothia
ICS Lecture 4

Today's Lecture

Hashes and Message Authentication Codes
- Properties of Hashes and MACs
- CBC-MAC, MAC -> HASH (slow),
- SHA1, SHA2, SHA3
- HASH -> MAC, HMAC

Authenticated Encryption
- CCM

Hashes

- A hash of any message is a short string generated from that message.
- The hash of a message is always the same.
- Any small change makes the hash totally different.
- It is very hard to go from the hash to the message.
- It is very unlikely that any two different messages have the same hash.

Signatures

- Using RSA $E_{pub}(D_{priv}(M)) = M$
- This can be used to sign messages.
- Sign a message with the private key and this can be verified with the public key.
- Any real crypto suite will not use the same key for encryption and signing.
  - as this can be used to trick people into decrypting.

Signatures

Alice has a signing key Ks
and wants to sign message M

Plain Text

Detached Signature: $D_{ks}(\underbrace{\#(M)})$

RSA decrypt with key ks

SHA hash

Signed: $M,D_{ks}(\#(M))$

Uses of Hashing

- Download/Message verification
- Tying parts of a message together (hash the whole message)
- Hash message, then sign the hash.
- Protect Passwords
  - Store the hash, not the password

Attacks on hashes

- Preimage Attack: Find a message for a given hash: very hard.
- Prefix Collision Attack: a collision attack where the attacker can pick a prefix for the message.
- Collision Attack: Find two “random” messages with the same hash.

Birthday Paradox

- How many people do you need to ask before you find 2 that have the same birthday?
- 23 people, gives (23*22)/2 = 253 pairs.
- Prob. that two people have a different birthday is: 364/365
- $(364/365)^{(23*22/2)} = 0.4995$

Message Authentication Codes

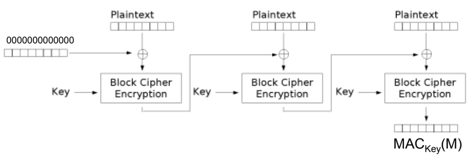
- MACs are hashes with a key.
  - Written $MAC_{Key}(M)$
- You can only make or check the hash, if you know the key.
- Stops guessing attacks.

Message Authentication Codes

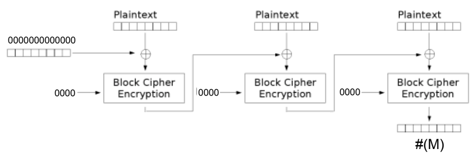
- MACs are sometimes used for authentication:
  - E.g. in Alice and Bank share keyA, Alice sends to the bank:
    “Pay Bob £10”,MAC_{keyA}(“Pay Bob £10”)

Possible attack on MAC: “Length extension attack” add data to a MAC without knowing the key

CBC MAC



An Inefficient Hash Function



The SHA Family of Hash

- The most common (and best) hashes are the SHA hashes.
- 1993, The US National Institute of Standards and Technology (NIST), developed a new hash SHA-0
- 1995, the NSA stepped in and “fixed” it: SHA-1 (160-bit hash).

SHA1

- A birthday attack on SHA-1 should need 2^80 hash tests
- In 2005 a 2^63 attack was found.
- Not really practical, but no-one trusts SHA-1 any more.
- So … SHA-2

SHA2

- SHA2 is an improved version of SHA1 with a longer hash.
- 256 or 512 bits: also called SHA256, SHA512.
- Based on SHA-1 it has some of the same weaknesses. So, even though it seems secure the cryptographers aren’t happy.

The SHA-3 Competition

- Submissions opened on October 31, 2008,
- Round 1
  - 13 submissions rejected without comment.
  - 10 withdrawn by authors.
  - 16 rejected for design or performance.
    - Inc. Sony’s
- Conference in Feb 2009. 14 scheme picked to go through to round 2.
  - Dropped schemes include
    - Ron Rivest’s,
    - Lockheed Martin

The SHA-3 Competition

- Winner announced on October 2, 2012 as
  - Keccak, (Daemen et al. the AES guy)
- This is too soon for it to be in standard APIs
- Expect this to be the standard soon.

Merkle–Damgård (MD) Hashes

- The MD family of hashes is also popular.
- MD4 & MD5 used, but weak.
  - Only useful when we only care about preimage attacks or Integrity.
- MD6: Ron Rivest’s candidate for SHA3.
  - Seems good & fast.

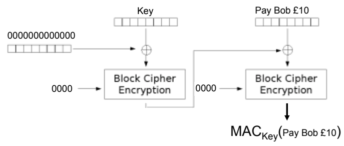
Broken Hash to MAC

- If we had a Hash we could try to make a MAC by:

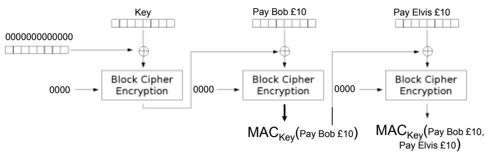
$$MAC_{Key}(M) = H(Key,M)$$

- But this might allow a length extension attack.

Broken Hash to MAC



Broken Hash to MAC



HMAC

- To stop this (and other attacks) we use HMACs:
- Given a hash function H we define:

$$HMAC_K(M) = H\big((K \text{ xor opad}), H((K \text{ xor ipad}), M) \big)$$

opad= 0x5c5c….5c ipad= 0x3636..36

Today's Lecture

Hashes and Message Authentication Codes
- Properties of Hashes and MACs
- CBC-MAC, MAC -> HASH (slow),
- SHA1, SHA2, SHA3
- HASH -> MAC, HMAC

Authenticated Encryption
- CCM

Cipher Texts Can Be Altered

- AES encryption with a particular key maps any 128-bit block to a 128-bit block (or 256)
- AES decrypt also maps any 128-bit block to a 128-bit block.
- Decrypt can be run on any block (not just encryptions).

Block mode

- CBC mode: any change affects all of the rest of the message.
- ECB mode: any change affects only the block.
- CTR mode: any change affects only the bits altered.

Known Plain Text Attacks

- If I know the plaintext I can change CTR encrypted messages.
- I.e. if I know Enc(M1) and I know M1, I can make a ciphertext that decrypts to any message I want, e.g. M2:

Dec(Enc(M1) xor (M1 xor M2)) = M2

Authenticated Encryption Modes

- Authenticated encryption modes stop this.
- With Authenticated Encryption you can only form a valid ciphertext if you know the key.
- Most common way to do this is to add a MAC to the ciphertext.

CCM mode encryption

- First calculate an AES CBC-MAC on the data.
- Then encrypt the message followed by the MAC using the same key and CTR mode.
- Not rocket science, but proven secure
  - Fully defined as RFC 3610

Today's Lecture

Hashes and Message Authentication Codes
- Properties of Hashes and MACs
- CBC-MAC, MAC -> HASH (slow),
- SHA1, SHA2, SHA3
- HASH -> MAC, HMAC

Authenticated Encryption
- CCM