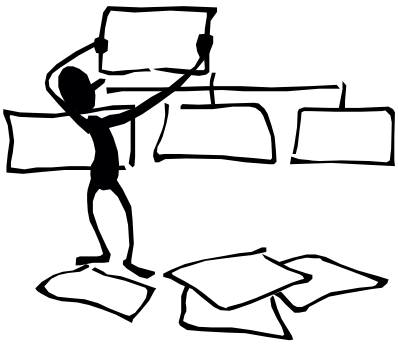
Message Authentication Codes

Murat Kantarcioglu

Based on Prof. Ninghui Li's Slides

Lecture Outline

- Message Authentication Code



Limitation of Using Hash Functions for Authentication

- Require an authentic channel to transmit the hash of a message
  - anyone can compute the hash value of a message, as the hash function is public
  - not always possible
- How to address this?
  - use more than one hash functions
  - use a key to select which one to use

Hash Family

- A hash family is a four-tuple (X, Y, K, H), where
  - X is a set of possible messages
  - Y is a finite set of possible message digests
  - K is the keyspace
  - For each $K \in K$, there is a hash function $h_K \in H$. Each h_K: $X \rightarrow Y$
- Alternatively, one can think of H as a function $K \times X \rightarrow Y$

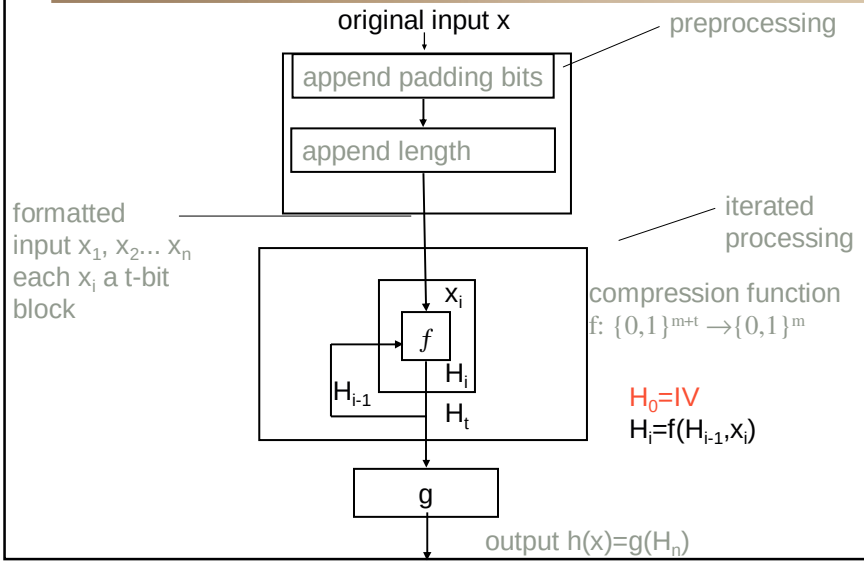
Message Authentication Code

- A MAC scheme is a hash family, used for message authentication
- MAC = $C_K(M)$
- The sender and the receiver share K
- The sender sends (M, $C_k(M)$)
- The receiver receives (X,Y) and verifies that $C_K(X)=Y$, if so, then accepts the message as from the sender
- To be secure, an adversary shouldn't be able to come up with (X,Y) such that $C_K(X)=Y$.

Constructing MAC from Hash Functions

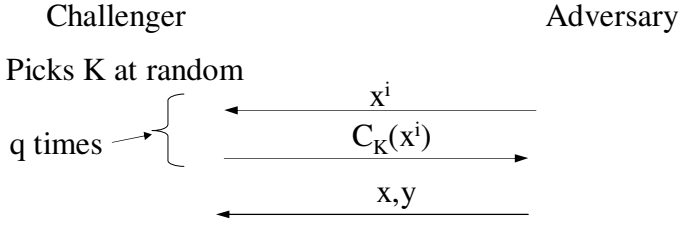
- Given a cryptographic (iterative) hash function h,
- Define $C_K(M)$ to be h(M) with K as IV
- Is this secure?
- Given a message x and its MAC $C_K(x)$, the adversary can construct x' and $C_K(x')$
  - let pad(x) be the padding added to x
  - let x'=x || pad(x) || w, y'=x' || pad(x')
  - then $C_K(x')$ can be computed from $C_K(x)$

Model for Iterated Hash Functions



Existential Forgery Attack against MAC

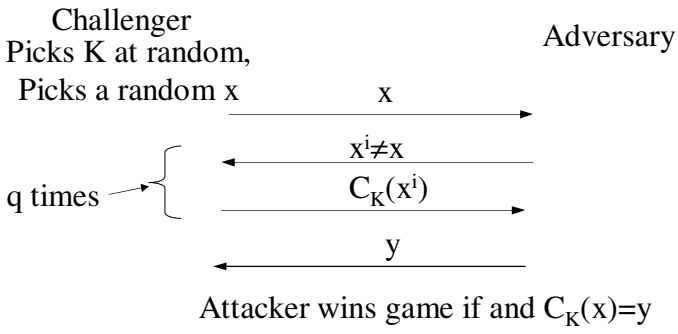
- Let C be a MAC function $C_K(M)$ is the MAC for M under K.



Attacker wins game if $x \notin \{x^1,\ldots,x^q\}$ and $C_K(x)=y$

Selective Forgery Attack Against MAC

- Let C be a MAC function $C_K(M)$ is the MAC for M under K.



Attacker wins game if and $C_K(x)=y$

MAC Security

- The pair (x, z) is called a forgery
- A (ε,q) forger
  - can produce a forgery with probability ε, after making q queries
  - generally talks about existential forgery
- The attacker against the MAC scheme $C_K(M)=h(M)$ with K as IV is a (1,1) forger

Constructing MAC using Hash Functions

- Are the following MAC schemes secure? What kind of forgers exist for them?
  - $C_K(M) = h(K \,||\, M)$, where h is a cryptographic hash function

HMAC Goals

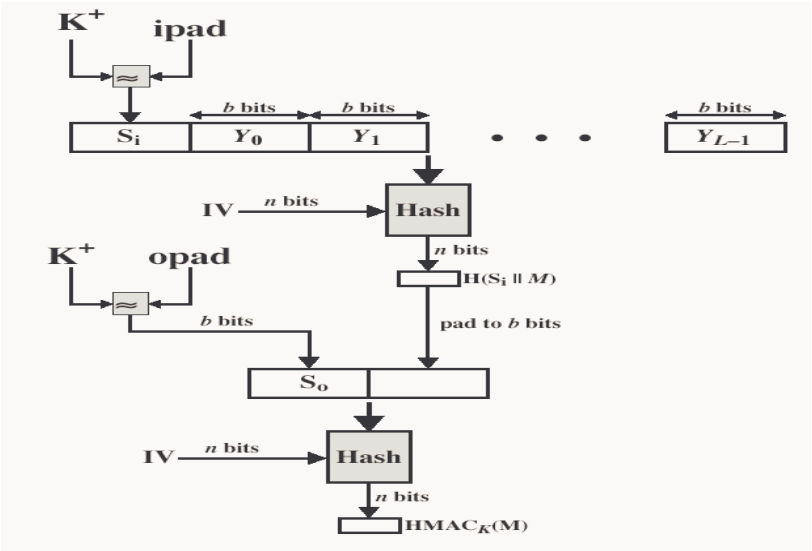
- Use available hash functions without modification.
- Preserve the original performance of the hash function without incurring a significant degradation.
- Use and handle keys in a simple way.
- Allow easy replacement of the underlying hash function in the event that faster or more secure hash functions are later available.
- Have a well-understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions on the underlying hash function.

HMAC

$HMAC_K = Hash[(K^+ \oplus opad) || Hash[(K^+ \oplus ipad)||M)]]$

- K^+ is the key padded out to input block size of the hash function and opad, ipad are specified padding constants
- Key size: L/2 < K < L
- MAC size: at least L/2, where L is the hash output

HMAC Overview



HMAC Security

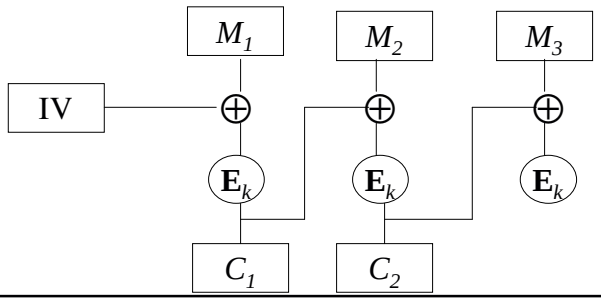
- Security of HMAC relates to that of the underlying hash algorithm
- If used with a secure hash functions (s.t. SHA1) and according to the specification (key size, and use correct output), not known practical attacks against HMAC
- In general, HMAC be attacked as follows:
  - brute force on the key space
  - attacks on the hash function itself
    - birthday attack, although the use of key makes this attack more difficult
    - attacks against the compression function

CBC-MAC

- Given a block cipher **E** with block size m
- Given message $M = M_1 \mid |M_2 || \ldots || M_n$
- MAC of M is $\mathbf{E}_k(M)$
  - $z_0 = IV = 0^m$
  - $z_i = \mathbf{E}_k(z_{i-1} \oplus M_i)$ for $1 \le i \le n$
  - $MAC = z_n$
- Random IV is needed in CBC encryption to prevent codebook attack on first block, not needed here.

Encryption Modes: CBC

- Cipher Block Chaining (CBC): next input depends of previous output
  - Plaintext is *M1*, *M2*, *M3*, *M4*,
  - Ciphertext is: $C_1 = IV \oplus \mathbf{E}_k(M_1)$ $C_2=C_1 \oplus \mathbf{E}_k(M_2)$ $C_3=C_2 \oplus \mathbf{E}_k(M_3)$ $C_4=C_3 \oplus \mathbf{E}_k(M_4)$



Security of CBC-MAC

- Secure for messages of a fixed number of blocks assuming the block cipher is PRP
- Not secure with variable lengths, example attack
  - given three pairs of messages/MACs (x1,y1) (x2,y2), (x1||z,y3), then
    - y1= $\mathbf{E}_K$[IV⊕x1]
    - y2= $\mathbf{E}_K$[IV⊕x2]
    - y3= $\mathbf{E}_K$[y1⊕z]=$\mathbf{E}_K$[y2 ⊕ (z ⊕y1⊕y2)]
    - let z'=(z ⊕y1⊕y2), (x2||z',y3) is also a valid pair

Optional Security Enhancement for CBC-MAC

- MAC of M is
  - z_0=IV=0^m
  - $z_i = \mathbf{E}_{K1}(z_{i-1} \oplus M_i)$ for $1 \le i \le n$
  - MAC=$\mathbf{E}_{K1}\mathbf{D}_{K2}[z_n]$
- Reduces threat of exhaustive key search
- Defends against the previous attack

Data Integrity Combined with Encryption

- Encryption alone does not guarantee data integrity
- Combining encryption with hash
  - $C=E_K[x \,||\, h(x)]$
  - breaking encryption also compromises integrity
  - may be vulnerable to known-plaintext attack

MAC with Encryption

- $C=E_K[x \,||\, h_{K'}(x)]$
  - separate keys used for encryption & for MAC
  - the algorithms E and h should be independent
  - precludes exhaustive key search on MAC key
- Alternative 1: $C=E_K[x], h_{K'}(E_K[x])$
  - allows message authentication without knowing x or K
  - authenticates only the ciphertext
- Alterative 2: $E_K[x],h_{K'}(x)$
  - requires $h_{K'}(x)$ does not leak information about x

CCM Mode

- CCM mode is a NIST standard that provides an authenticated encryption
- MAC is produced as a part of the encryption process
- CCM mode uses CTR mode for encryption and CBC-MAC for authentication
  - Given message $M = M_1 \,||M_2 \,||\, \ldots \,||\, M_n$
  - $C_i = M_i \oplus \mathbf{E}_k(ctr+i \bmod 2^m)$
  - temp=CBC-Mac(M,k) and C'= temp $\oplus \mathbf{E}_k(ctr)$
  - Return $C_1||\ldots.\,||C_n||C'$

Next Lectures..

- Number theory
- Readings:
  - Stingson: 5.1, 5.2, 5.4