

Cryptography CS 555

Lecture 11



Message Authentication Code

Review of Cryptographic Hash Functions

- A hash function $h: \{0,1\}^* \rightarrow \{0,1\}^m$
- Constructs a short “fingerprint” or “message digest” of an arbitrarily long message
- Security properties (in increasing order of strength)
 - preimage resistant (one-way)
 - 2-nd preimage resistant (weak collision resistant)
 - collision resistant (strong collision resistant)
 - useful whenever non-repudiation is desired
- Birthday attack takes time $O(2^{m/2})$ to find collisions with probability >0.5

Review of Cryptographic Hash Functions

- Iterative construction of a hash function
 $h:\{0,1\}^* \rightarrow \{0,1\}^m$ from a compression function
 $f:\{0,1\}^{m+t} \rightarrow \{0,1\}^m$
 - preprocessing (padding to have length be multiple of t) and divide into t -bit blocks, x_1, x_2, \dots, x_n
 - iteratively apply the compression function $H_i=f(H_{i-1},x_i)$, where $H_0=IV$, and H_n is the hash value
- Merkle-Damgard Construction
 - collision-free if the compression function is collision-free

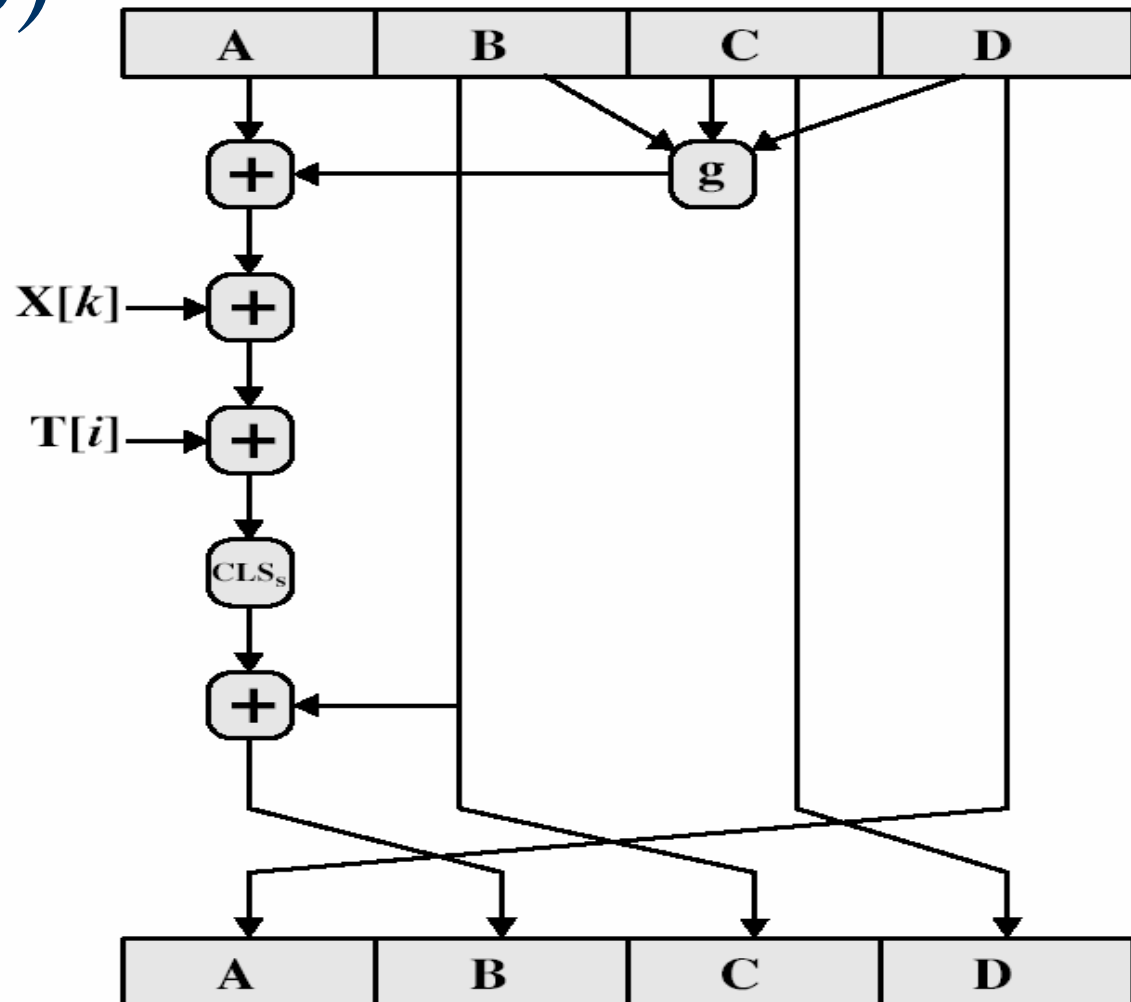
Review of Cryptographic Hash Functions

- MD2, MD4, MD5 (outputs 128 bits)
 - all broken in the sense of not being collision resistant
- SHA1 (output 160 bits)
 - still secure today
- SHA-256, SHA-384, SHA-512
 - to match level of security provided by AES

Review of Cryptographic Hash Functions

- On constructing collision-free compression functions
 - the construction of $f(H_{i-1}, x_i)$ is often similar to $E_{x_i}[H_{i-1}]$
 - where x_i is used to give (or generate) round keys
 - collision attacks mean finding two keys that encrypt a given message into the same ciphertext
 - this is an attack model not considered in block ciphers, and not modeled by the PRP requirement

MD5 Compression Function (Single Step)



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 $K \times X \rightarrow Y$

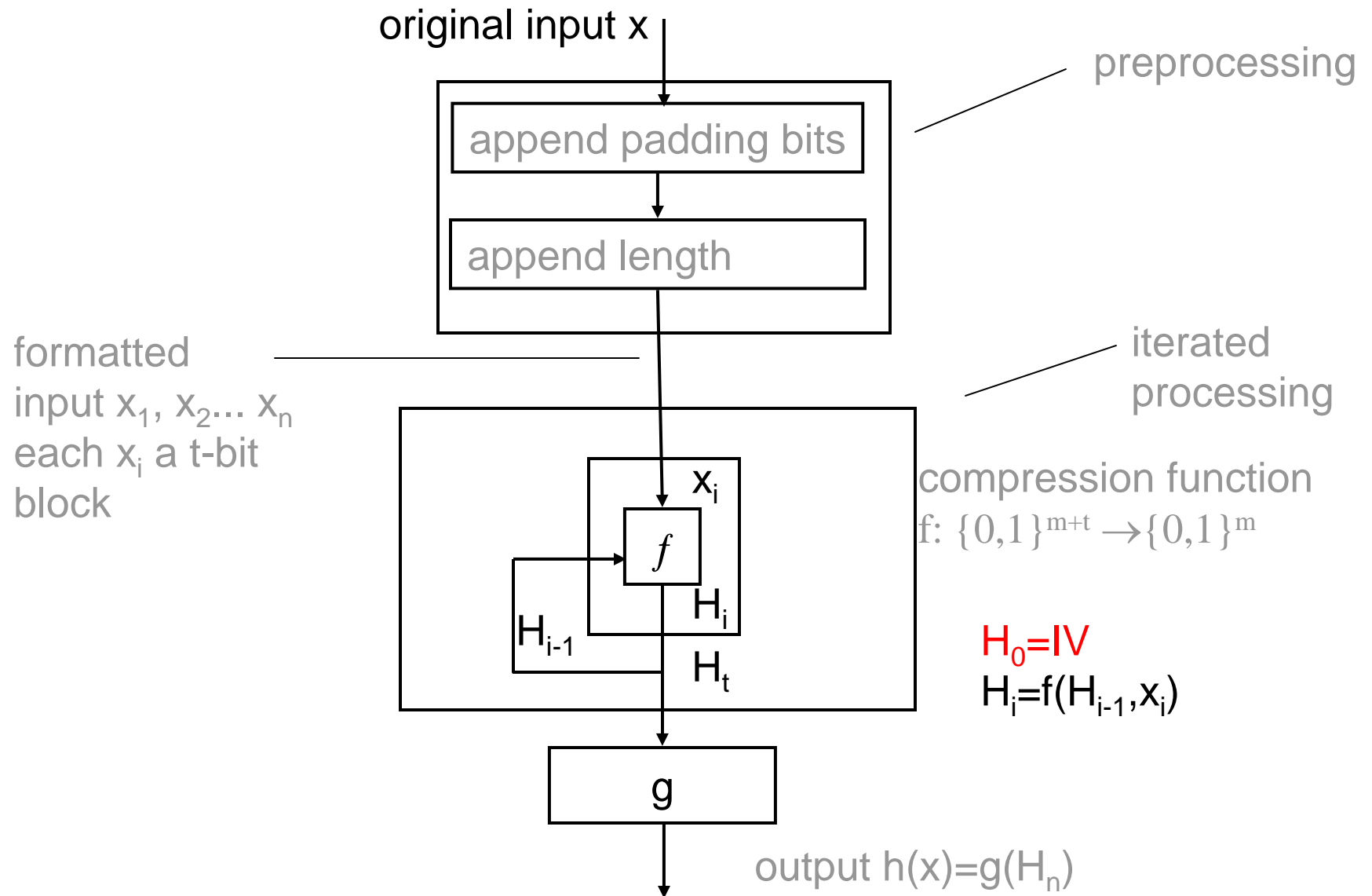
Message Authentication Code

- 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

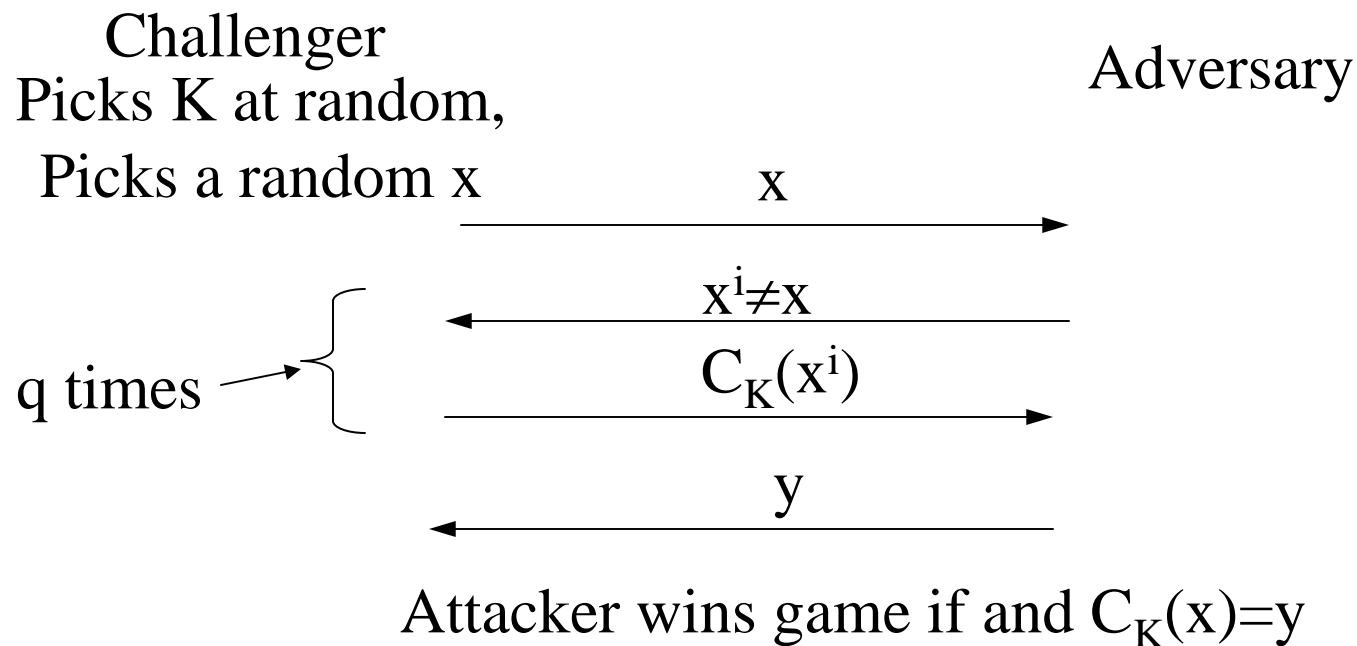
- Given a cryptographic (iterative) hash function h ,
- Define $C_K(M)$ to be $h(M)$ with K as IV
- Is this secure?
- Given a message x and its MAC $C_K(x)$, the adversary can construct x' and $C_K(x')$
 - let $\text{pad}(x)$ be the padding added to x
 - let $x' = x \parallel \text{pad}(x) \parallel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

Model for Iterated Hash Functions



Selective Forgery Attack Against MAC

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



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

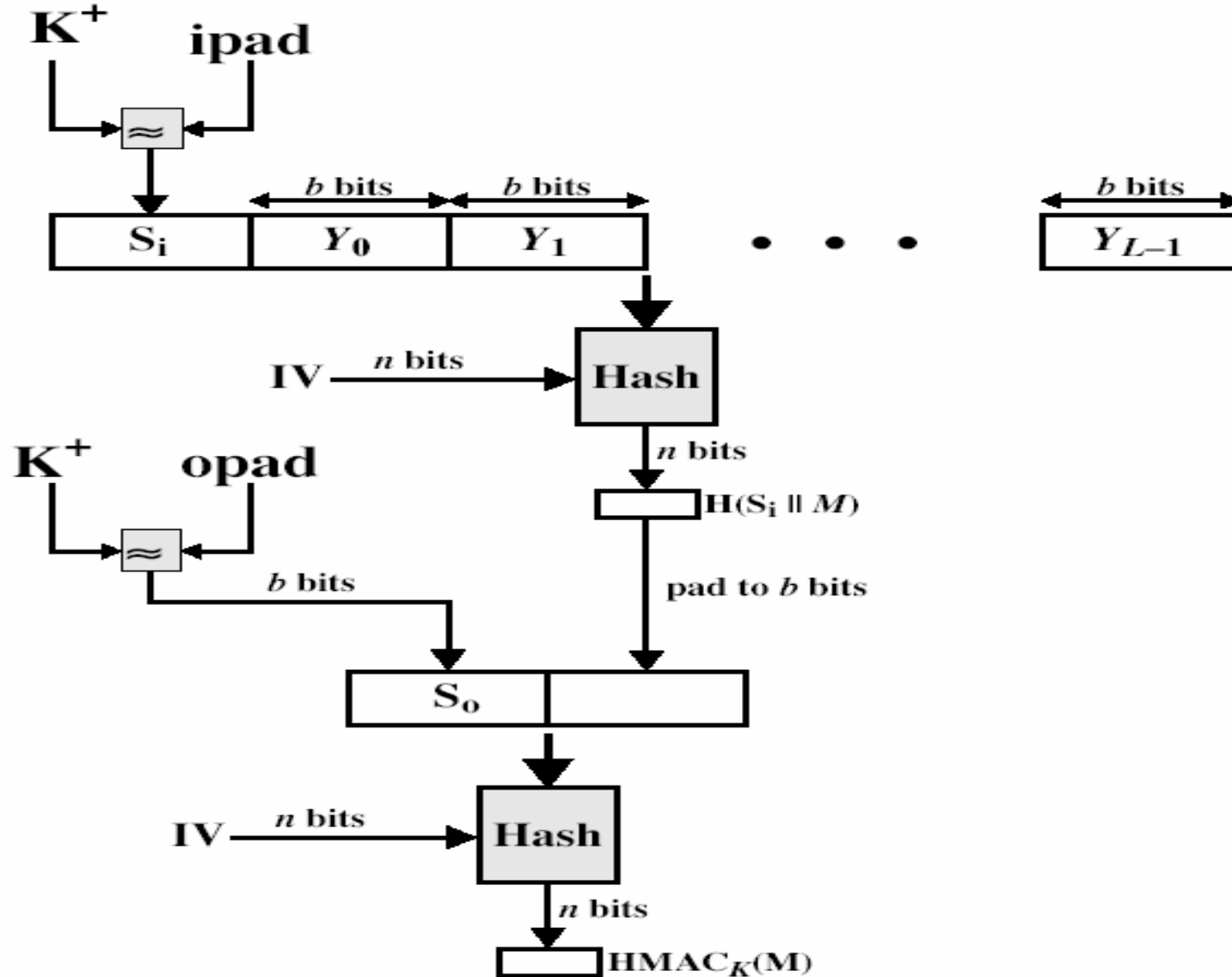
- Use available hash functions without modification.
- Preserve the original performance of the hash function without incurring a significant degradation.
- Use and handle keys in a simple way.
- Allow easy replacement of the underlying hash function in the event that faster or more secure hash functions 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

$$\text{HMAC}_K = \text{Hash}[(K^+ \oplus \text{opad}) \parallel \text{Hash}[(K^+ \oplus \text{ipad}) \parallel M]]$$

- 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

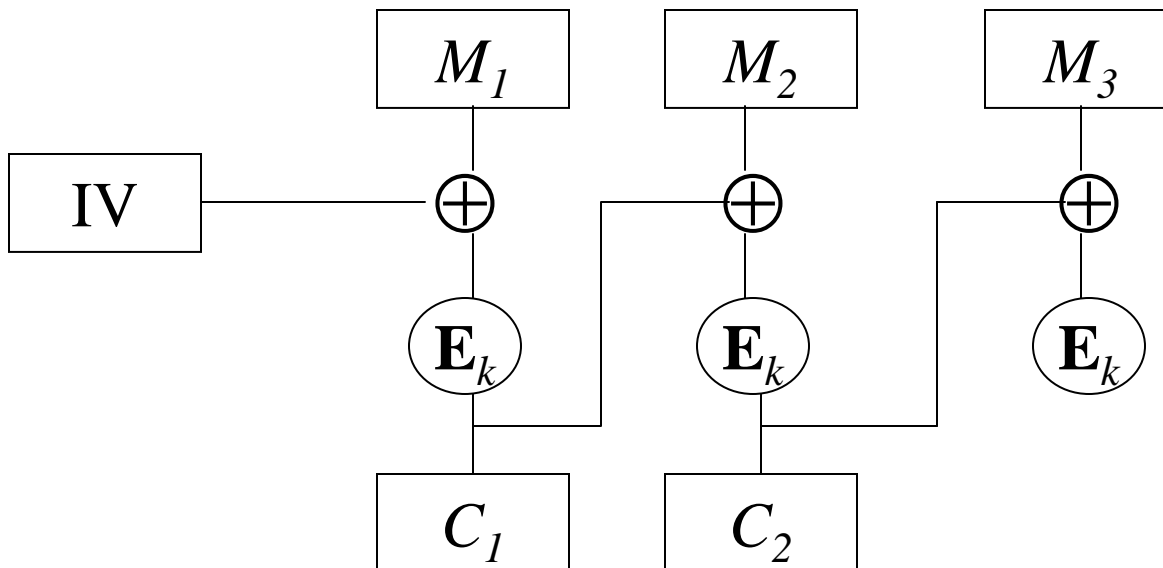
- Security of HMAC relates to that of the underlying hash algorithm
- If used with a secure hash functions (s.t. SHA1) and according to the specification (key size, and use correct output), not known practical attacks against HMAC
- In general, HMAC be attacked as follows:
 - brute force on the key space
 - 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 \mathbf{E} with block size m
- Given message $M = M_1 \parallel M_2 \parallel \dots \parallel M_n$
- 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 $1 \leq i \leq n$
 - $MAC = z_n$
- 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 $M_1, M_2, M_3, M_4,$
 - 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 (x_1, y_1) (x_2, y_2) , $(x_1 || z, y_3)$, then
 - $y_1 = E_K[IV \oplus x_1]$
 - $y_2 = E_K[IV \oplus x_2]$
 - $y_3 = E_K[y_1 \oplus z] = E_K[y_2 \oplus (z \oplus y_1 \oplus y_2)]$
 - let $z' = (z \oplus y_1 \oplus y_2)$, $(x_2 || z', y_3)$ is also a valid pair

Optional Security Enhancement for CBC-MAC

- MAC of M is
 - $z_0 = IV = 0^m$
 - $z_i = \mathbf{E}_{K_1}(z_{i-1} \oplus M_i)$ for $1 \leq i \leq n$
 - $MAC = \mathbf{E}_{K_1} \mathbf{D}_{K_2}[z_n]$
- Reduces threat of exhaustive key search
- Defends against the previous attack

Unconditionally Secure MAC

- MAC's that are unconditionally secure when at most one query can be made
 - secure against adversaries with unlimited computation power
- Intuition: a MAC scheme is unconditionally secure if
 - given one message/MAC pair, for any other message, each value is equally likely to be the MAC value of the message

Example:

- $X=Y=Z_3$
- $K=Z_3 \times Z_3$
- $h_{(a,b)}(x) = ax + b \pmod{3}$
- $H = \{ h_{(a,b)} : (a,b) \in Z_3 \times Z_3 \}$
- Suppose that we know $ax_1 + b \pmod{3} = y_1$
- Then $ax_2 + b \pmod{3}$ can be any value in Z_3

Strongly Universal Hash Families

- Suppose that (X, Y, K, H) is an (N, M) hash family. This hash family is *strongly universal* provided that the following condition is satisfied for every $x \neq x' \in X$ and every $y, y' \in Y$

$$|\{K \in \mathbf{K} : h_K(x) = y, h_K(x') = y'\}| = \frac{|\mathbf{K}|}{M^2}$$

- A strong universal hash family is an unconditionally secure MAC scheme

Data Integrity Combined with Encryption

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

MAC with Encryption

- $C = E_K[x \parallel h_{K'}(x)]$
 - 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
- Alternative 2: $E_K[x], h_{K'}(x)$
 - requires $h_{K'}(x)$ does not leak information about x

Next Lectures..

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