Big MACs attacks
Unconditionally secure MACs

Message Authentication Codes

- Message authentication codes are keyed hash functions satisfying certain security requirements.
- These requirements are different from those required by unkeyed hash functions.
A common approach to MACs*

*Attempts to turn an unkeyed hash function into a MAC.

Recall the main steps of an iterated hash

Preprocessing
Given input string \( x \), where \(|x| \geq m+t+1\), construct string
\[
y = y_1 \parallel y_2 \parallel \ldots \parallel y_r,
\]
where \(|y_i| = t\) for \(1 \leq i \leq r\) using a public algorithm.

Processing
Let IV be a public initial bitstring of length \( m \). Compute
\[
z_0 \parallel IV
z_1 \parallel \text{compress}(z_0 \parallel y_1)
\ldots
z_r \parallel \text{compress}(z_{r-1} \parallel y_r).
\]

Optional output transformation
Define \( h(x) = g(z_r) \), where \( g: \{0,1\}^m \rightarrow \{0,1\}^l \) is a public function.
Turning our hash into a MAC

- Assume that $h$ does not have a preprocessing step or an output transformation. In this case, the message $x$ is a multiple of $t$, where $\text{compress}: \{0,1\}^m \rightarrow \{0,1\}^m$ is the compression function.

- Let $\text{IV} = K$ be the secret key bitstring of length $m$. Compute
  
  $z_0 \leftarrow \text{IV}$
  $z_i \leftarrow \text{compress}(z_{i-1} \parallel x_i)$
  
  $\ldots$
  $z_r \leftarrow \text{compress}(z_{r-1} \parallel x_r)$

  Return $h_K(x) = z_r$.

- Let $x'$ be any string of length $t$, and consider the message $x \parallel x'$.

Forgery

- The adversary obtains a list of valid pairs $(x_1,y_1), (x_2,y_2), \ldots, (x_q,y_q)$, by querying the oracle. Later, the adversary outputs the pair $(x,y)$, where $x \notin \{x_1, \ldots, x_q\}$.

- If $(x,y)$ is a valid pair, it is said to be a forgery.

- An adversary that produces a forgery with probability $\epsilon$ is said to be an $\epsilon$-forger.
Nested Macs

- Suppose \((X, Y, K, G)\) and \((Y, Z, L, H)\) are hash families.
- The composition is the hash family \((Y, Z, M, G \circ H)\)
in which \(M = K \times L\) and \(G \circ H = \{g \circ h : f \in G, h \in H\}\),
where \(g \circ h_{(K,L)}(x) = h_L(g_K(x))\)
for all \(x \in X\).

Attacks on nested MACs

**Big MAC attack**

The adversary is allowed to choose value for \(x\) and query a big MAC oracle for values of \(h_L(g_K(x))\). The adversary attempts to output a pair \((x', z)\) such that \(z = h_L(g_K(x'))\).

**Little MAC attack**

The adversary is allowed to choose value for \(y\) and query a little MAC oracle for values of \(h_L(y)\). The adversary attempts to output a pair \((y', z)\) such that \(z = h_L(y')\).

**Unknown-key collision attack**

The adversary is allowed to choose values for \(x\) and query a hash oracle for value of \(g_K(x)\). The adversary attempts to output a pair \(x', x''\) such that \(x' \neq x''\) and \(g_K(x') = g_K(x'')\).
Probability of a big MAC attack

Theorem.
Suppose \((Y, Z, M, G \circ H)\) is a nested MAC.
- Suppose there does not exist an \((q_1, q_1)\)-collision attack for a randomly chosen function \(g_k \in G\), when the key \(K\) is secret.
- Further, suppose that there does not exist an \((q, q)\)-forger for a randomly chosen function \(h_L \in H\), where \(L\) is secret.
- Finally, suppose there exists an \((q, q)\)-forger for the nested MAC, for a randomly chosen function \((g \circ h)_{(K,L)} \in G \circ H\).
then \(e \leq e_1 + e_2\).

HMAC
- HMAC is a nested MAC algorithm constructed from an hash function (in e.g., SHA-1) using a secret key, \(K\).
- \(ipad\) and \(opad\) are 512-bit constants defined in hexadecimal to be
  \(ipad = 3636...36\)
  \(opad = 5C5C...5C\)
Unconditionally secure MACs

- Assuming each key is used only once, we construct a MAC for which there provably does not exist either an
  - (0,0)-forger, known as an impersonation; or
  - (1, 1)-forgers, known as a substitution.
- Define the deception probability $P_d$ to be the maximum $e$ such that an $(e, q)$-forgery exists.

Authentication matrix

- Consider the hash family $(X, Y, K, H)$ where $X = Y = Z_3$ and $K = Z_3 \times Z_3$.
- For each $K = (a, b) \in K$ and each $x \in X$, define
  $$h_{(a,b)}(x) = ax + b \mod 3$$
  and define
  $$H = \{h_{(a,b)} : (a, b) \in Z_3 \times Z_3\}.$$
Impersonation and payoff

- Let $K_0$ denote the chosen key. For $x \in X$ and $y \in Y$, define
  
  \[ \text{payoff}(x, y) \] 
  
  to be the probability that the pair $(x, y)$ is valid.

- \[ \text{payoff}(x, y) = \Pr[y = h_{K_0}(x)] \] 
  \[ = \frac{|\{K \in \mathcal{K} : h_K(x) = y\}|}{|\mathcal{K}|} \]

  and

  \[ P_{d_0} = \max\{\text{payoff}(x,y) : x \in X \text{ and } y \in Y\}. \]

Substitution and payoff

- Define \[ \text{payoff}(x', y' : x, y) \] to be the probability that $(x', y')$ is a valid pair, given that $(x, y)$ is. Then,

  \[ \text{payoff}(x', y' : x, y) = \Pr[y' = h_{K_0}(x') \mid y = h_{K_0}(x)] \] 
  \[ = \frac{\Pr[y' = h_{K_0}(x') \mid y = h_{K_0}(x)]}{\Pr[y = h_{K_0}(x)]} \] 
  \[ = \frac{|\{K \in \mathcal{K} : h_K(x') = y', h_K(x) = y\}|}{|\{K \in \mathcal{K} : h_K(x) = y\}|} \]

- \[ P_{d_1} = \max\{\text{payoff}(x', y' : x, y) : x, x' \in X; y, y' \in Y; (x, y) \in V; x \neq x'\} \]

  where \[ V = \{ (x, y) : K \in \mathcal{K} : h_K(x) = y \} \geq 1 \].
Strongly universal hash families

Definition
An \((N, M)\) hash family \((X, Y, K, H)\) is strongly universal if and only if for all \(x, x' \in X\) such that \(x \neq x'\), and for every \(y, y' \in Y\):

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

Secure hash families

Lemma.
Suppose that \((X, Y, K, H)\) is a strongly universal \((N, M)\)-hash family. Then

\[
|\{K \in K : h_K(x) = y\}| = \frac{|K|}{M}
\]
for every \(x \in X\) and for every \(y \in Y\).

Theorem.
Any strongly universal \((N, M)\)-hash family \((X, Y, K, H)\) is an authentication code with \(P_{d0} = P_{d1} = 1/M\).
Sum of the payoffs is 1

- Suppose that \((X, Y, K, H)\) is an \((N, M)\)-hash family. Fix \(x \in X\).

Then
\[
\text{payoff}(x, y) = \sum_{y \in Y} \frac{|\{ K \in K : h_K(x) = y \}|}{|K|} = \frac{|K|}{|K|} = 1
\]

- Hence, for every \(x \in X\) there exists an authentication tag \(y\) such that \(\text{payoff}(x, y) \geq 1/M\).

Pd\(_0\) optimal hash families

**Theorem.**

Suppose that \((X, Y, K, H)\) is an \((N, M)\)-hash family. Then \(Pd_0 \geq 1/M\). Further, \(Pd_0 = 1/M\) if and only if
\[
|\{ K \in K : h_K(x) = y \}| = \frac{|K|}{M}
\]
Turning to substitution

- Fix \( x, x' \in X \) and \( y, y' \in Y \), where \( x \neq x' \) and \( (x, y) \in V \).

\[
\text{payoff}(x', y'; x, y) = \frac{|\{K \in \mathcal{K}; h_K(x') = y', h_K(x) = y\}|}{|\{K \in \mathcal{K}; h_K(x) = y\}|}
\]

\[
= \frac{|\{K \in \mathcal{K}; h_K(x) = y\}|}{|\{K \in \mathcal{K}; h_K(x) = y\}|} = 1
\]

- Hence, for every \((x, y) \in V\) and each \( x' \) such that \( x' \neq x \), there exists an authentication tag \( y \) such that \( \text{payoff}(x, y'; x, y) \geq \frac{1}{M} \).

Strongly universal hash families are \( P_{d_1} \) optimal

**Theorem.**

Suppose that \((X, Y, K, H)\) is an \((N, M)\)-hash family. Then \( P_{d_1} \geq \frac{1}{M} \).

**Theorem.**

Suppose that \((X, Y, K, H)\) is an \((N, M)\)-hash family. Then \( P_{d_1} = \frac{1}{M} \) if and only if the hash family is strongly universal.