Collision Resistant Hashing
Definition: A hash function (with output length $\ell$) is a pair of ppt algorithms $(Gen, H)$ satisfying the following:

- $Gen$ takes as input a security parameter $1^n$ and outputs a key $s$. We assume that $1^n$ is implicit in $s$.
- $H$ takes as input a key $s$ and a string $x \in \{0,1\}^*$ and outputs a string $H^s(x) \in \{0,1\}^{\ell(n)}$.

If $H^s$ is defined only for inputs $x \in \{0,1\}^{\ell'(n)}$ and $\ell'(n) > \ell(n)$, then we say that $(Gen, H)$ is a fixed-length hash function for inputs of length $\ell'$. In this case, we also call $H$ a compression function.
The collision-finding experiment

\( Hashcoll_{A,\Pi}(n) \):

1. A key \( s \) is generated by running \( Gen(1^n) \).
2. The adversary \( A \) is given \( s \) and outputs \( x, x' \). (If \( \Pi \) is a fixed-length hash function for inputs of length \( \ell'(n) \), then we require \( x, x' \in \{0,1\}^{\ell'(n)} \).)
3. The output of the experiment is defined to be 1 if and only if \( x \neq x' \) and \( H^s(x) = H^s(x') \). In such a case we say that \( A \) has found a collision.
Security Definition

Definition: A hash function $\Pi = (Gen, H)$ is collision resistant if for all ppt adversaries $A$ there is a negligible function $neg$ such that

$$\Pr[\text{Hashcoll}_{A,\Pi}(n) = 1] \leq neg(n).$$
Weaker Notions of Security

• Second preimage or target collision resistance: Given $s$ and a uniform $x$ it is infeasible for a ppt adversary to find $x' \neq x$ such that $H^s(x') = H^s(x)$.

• Preimage resistance: Given $s$ and uniform $y$ it is infeasible for a ppt adversary to find a value $x$ such that $H^s(x) = y$. 
Domain Extension
The Merkle-Damgard Transform

**FIGURE 5.1**: The Merkle-Damgård transform.
The Merkle-Damgward Transform

Let \((\text{Gen}, h)\) be a fixed-length hash function for inputs of length \(2n\) and with output length \(n\). Construct hash function \((\text{Gen}, H)\) as follows:

- \(\text{Gen}\): remains unchanged
- \(H\): on input a key \(s\) and a string \(x \in \{0,1\}^*\) of length \(L < 2^n\), do the following:

1. Set \(B := \left\lceil \frac{L}{n} \right\rceil\) (i.e., the number of blocks in \(x\)). Pad \(x\) with zeros so its length is a multiple of \(n\). Parse the padded result as the sequence of \(n\)-bit blocks \(x_1, \ldots, x_B\). Set \(x_{B+1} := L\), where \(L\) is encoded as an \(n\)-bit string.
2. Set \(z_0 := 0^n\). (This is also called the IV.)
3. For \(i = 1, \ldots, B + 1\), compute \(z_i := h^s(z_{i-1}||x_i)\).
4. Output \(z_{B+1}\).
Security of Merkle-Damgård

Theorem: If \((\text{Gen}, h)\) is collision resistant, then so is \((\text{Gen}, H)\).
Message Authentication Using Hash Functions
Hash-and-Mac Construction

Let $\Pi = (Mac, Vrfy)$ be a MAC for messages of length $\ell(n)$, and let $\Pi_H = (Gen_H, H)$ be a hash function with output length $\ell(n)$. Construct a MAC $\Pi' = (Gen', Mac', Vrfy')$ for arbitrary-length messages as follows:

- **Gen'**: on input $1^n$, choose uniform $k \in \{0,1\}^n$ and run $Gen_H(1^n)$ to obtain $s$. The key is $k' := \langle k, s \rangle$.
- **Mac'**: on input a key $\langle k, s \rangle$ and a message $m \in \{0,1\}^*$, output $t \leftarrow Mac_k(H^s(m))$.
- **Vrfy'**: on input a key $\langle k, s \rangle$, a message $m \in \{0,1\}^*$, and a MAC tag $t$, output 1 if and only if $Vrfy_k(H^s(m), t) = 1$. 
Security of Hash-and-MAC

Theorem: If $\Pi$ is a secure MAC for messages of length $\ell$ and $\Pi_H$ is collision resistant, then the construction above is a secure MAC for arbitrary-length messages.
Proof Intuition

Let $Q$ be the set of messages $m$ queried by adversary $A$.
Assume $A$ manages to forge a tag for a message $m^* \notin Q$.

There are two cases to consider:

1. $H^S(m^*) = H^S(m)$ for some message $m \in Q$.
   Then $A$ breaks collision resistance of $H^S$.

2. $H^S(m^*) \neq H^S(m)$ for all messages $m \in Q$.
   Then $A$ forges a valid tag with respect to MAC $\Pi$. 

Can we construct a MAC from only CRHF?

Attempt: $Mac_k(m) = H(k||m)$.

Is this secure?

NO. Why not?

Instead, we will try 2 layers of hashing.