Hash Functions

A hash function is a computationally efficient function that maps arbitrary length binary strings into some fixed length hash values.

For use in cryptography we assume that:

- **No Collision.** It is computationally infeasible to find two strings that have the same hash values.

- Given an hash value, it is computationally infeasible to find a string whose hash value is the given one.

Hash functions are used for signatures (signing the hash of a message as opposed to the message itself) to save time and space.

They are also used for data integrity, where hash values of data are occasionally computed and stored securely, to later verify that data has not been altered.

Key Establishment and Management

**Key Establishment** Making the secret key available to several parties. This consists of key agreement and key transport.

**Key Management** Managing key establishment and maintenance of ongoing keying relationship between parties, including replacement of keys when necessary.

Using symmetric-key techniques, Trent may share a secret key with each of the participants (distributed through a secure channel) and issue a session key, encrypted with the secret key, for each two parties that want to establish secure communication.

Note, however, that Trent has to be involved with each 2-party communication, and may be able to read every message in the system.

Example of Key Management with Public-Key Techniques

Each participant has a public/private key pair. Trent has a private signing function $S_T$ and a public verification function $V_T$. Trent certifies the public key of each participant by signing the participant’s name and public key.
The advantages of this method is that Trent cannot read the message sent in the system, pre-communication about keys is eliminated, and Mallory cannot impersonate others.

However, if ST is compromised, every communication becomes insecure.

**Number Theory**

The set of integers \( \{ \ldots, -2, -1, 0, 1, 2, \ldots \} \) is denoted by \( \mathbb{Z} \).

1. For integers \( a \) and \( b \), we say that \( a \) divides \( b \), and denote it by \( a \mid b \), if \( b = ac \) for some integer \( c \).

2. For integers \( a \) and \( m \geq 1 \), let \( q \in \mathbb{Z} \) (the *quotient*) and \( r \in \mathbb{Z} \), \( 0 \leq r < m \), (the *remainder*) be such that \( a = qm + r \). \( r \) is denoted by \( a \mod m \) and \( q \) is denoted by \( a \div m \).

3. An integer \( c \) is a *common divisor* of \( a \) and \( b \) if it divides both. It is their *greatest common divisor*, \( \gcd(a, b) \) any other common divisor divides it.

4. \( a \) and \( b \) are *relatively prime* if their \( \gcd \) is 1.

**The Fundamental Theorem of Arithmetic**

Every integer \( m \geq 2 \) has a unique factorization as a product of prime number:

\[
n = p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k}
\]

where the \( p_i \)'s are mutually distinct primes and the \( e_i \)'s are positive integers.

**Corollary of the Fundamental Theorem** Let \( p_1, p_2, \ldots \) be an enumeration of all the primes (i.e., \( p_i \) is the \( i \)th largest prime). From the Fundamental theorem of arithmetic, for every integers \( a \) and \( b \), there exist unique \( k \), \( a_i \)'s and \( b_i \)'s such that

\[
a = p_1^{a_1} \cdot p_2^{a_2} \cdots p_k^{a_k} \\
b = p_1^{b_1} \cdot p_2^{b_2} \cdots p_k^{b_k}
\]

where \( p_k \) is the largest prime that divides either \( a \) or \( b \), and each \( a_i \) and \( b_i \) non-negative integers.
Then
\[ \gcd(a, b) = p_1^{\min(a_1, b_1)} \cdot p_2^{\min(a_2, b_2)} \cdots p_k^{\min(a_k, b_k)} \]

Thus, if \( \gcd(a, b) = 1 \) then for every \( i = 1, \ldots, k \), either \( a_i = 0 \) or \( b_i = 0 \).

**Euclidean Algorithm**  Given two integers, \( r_0 \) and \( r_1 \), \( r_0 \geq r_1 > 0 \), compute \( \gcd(r_0, r_1) \).

Compute
\[
\begin{align*}
q_1, r_2 & \text{ s.t. } r_0 = q_1 \cdot r_1 + r_2, \quad 0 \leq r_2 \leq r_1 \\
q_2, r_3 & \text{ s.t. } r_1 = q_2 \cdot r_2 + r_3, \quad 0 \leq r_3 < r_2 \\
& \vdots \\
q_i, r_{i+1} & \text{ s.t. } r_{i-1} = q_i \cdot r_i + r_{i+1}, \quad 0 \leq r_{i+1} < r_i 
\end{align*}
\]

until \( r_{i+1} = 0 \). Then, \( r_i = \gcd(r_0, r_1) \).

For example, if \( r_0 = 15 \) and \( r_1 = 12 \), then we get \( r_2 = 3 \) and \( r_3 = 0 \), thus \( \gcd(15, 12) = 3 \).

**Why Does it Work?**  The correctness of the Euclidean Algorithm is based on the following fact:

For every integers \( a > b > 0 \), \( \gcd(a, b) = \gcd(b, a \mod b) \).

Proof: Assume \( r = a \mod b \). Thus, \( a = qb + r \). Observe first that \( \gcd(b, r)|a \).

Since \( \gcd(b, r)|b \), it follows that \( \gcd(b, r) \leq \gcd(a, b) \). Similarly, \( \gcd(a, b)|r \) and \( \gcd(a, b)|b \), thus \( \gcd(a, b) \leq \gcd(b, r) \). We can therefore conclude that \( \gcd(a, b) = \gcd(b, r) \).

Going back to the Euclidean algorithm, we conclude that for every \( i \) such that \( r_i > 0 \), \( \gcd(r_0, r_1) = \gcd(r_i, r_{i+1}) \), which establishes the correctness of the algorithm.

**Extended Euclidean Algorithm**  Given two integers, \( r_0 \) and \( r_1 \), \( r_0 \geq r_1 > 0 \), compute \( \gcd(r_0, r_1) \) and a integers \( x \) and \( y \) such that \( r_0x + r_1y = \gcd(r_0, r_1) \).

To compute such \( x \) and \( y \):

Run Euclidean Algorithm starting with \( x_0 = 1; x_1 = 0; y_0 = 0; y_1 = 1 \). After computing \( q_i \) (\( i > 0 \)), compute:
\[
\begin{align*}
& x_{i+1} := x_{i-1} - q_i \cdot x_i; \quad y_{i+1} := y_{i-1} - q_i \cdot y_i;
\end{align*}
\]
This will maintain \( r_0 x_i + r_1 y_i = r_i \). Thus, upon termination (with \( r_{i+1} = 0 \)), we will have \( r_0 x_i + r_1 y_i = r_i = \gcd(r_0, r_1) \).

In our previous example (\( \gcd(15, 12) \)) we have:

\[
\begin{align*}
r_0 &= 15 & x_0 &= 1 & y_0 &= 0 \\
r_1 &= 12 & q_1 &= 1 & x_1 &= 0 & y_1 &= 1 \\
r_2 &= 3 & q_2 &= 4 & x_2 &= 1 & y_2 &= -1 \\
r_3 &= 0
\end{align*}
\]

**Why does the Extended Euclidean Algorithm Work?** We have to show that for every \( i \), \( r_0 x_i + r_1 y_i = r_i \). We show this by induction on \( i \). For the base cases, note that

\[
\begin{align*}
r_0 x_0 + r_1 y_0 &= r_0 1 + r_1 0 = r_0 \\
r_1 x_1 + r_1 y_1 &= r_0 0 + r_1 1 = r_1
\end{align*}
\]

For the inductive step assume that we showed that \( r_0 x_k + r_1 y_k = r_k \) for every \( 1 \leq k \leq i \), and that \( r_{i-1} = q_i r_i + r_{i+1} \), \( x_{i+1} = x_{i-1} - q_i x_i \), and \( y_{i+1} = y_{i-1} - q_i y_i \). Then:

\[
\begin{align*}
r_0 x_{i+1} + r_1 y_{i+1} &= r_0 (x_{i-1} - q_i x_i) + r_1 (y_{i-1} - q_i y_i) \\
&= r_0 x_{i-1} + r_1 y_{i-1} - q_i (r_0 x_i + r_1 y_i) \\
&= r_i - q_i r_i \\
&= r_{i+1}
\end{align*}
\]

**\( Z_m \): The integers modulo \( m \)**

**Definition 1** The integers modulo \( m \), denoted by \( Z_m \), is the set \( \{0, \ldots, m - 1\} \). The arithmetic operations \( + \) and \( \cdot \), are performed modulo \( m \).

**Properties of \( Z_m \):** For every \( a, b, c \in Z_m \):

<table>
<thead>
<tr>
<th>Property</th>
<th>Addition</th>
<th>Multiplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>closure</td>
<td>( a + b \in Z_m )</td>
<td>( a \cdot b \in Z_m )</td>
</tr>
<tr>
<td>commutativity</td>
<td>( a + b = b + a )</td>
<td>( a \cdot b = b \cdot a )</td>
</tr>
<tr>
<td>associativity</td>
<td>( (a + b) + c = a + (b + c) )</td>
<td>( (a \cdot b) \cdot c = a \cdot (b \cdot c) )</td>
</tr>
<tr>
<td>identity</td>
<td>( a + 0 = a )</td>
<td>( a \cdot 1 = a )</td>
</tr>
<tr>
<td>inverse</td>
<td>( m - a = a + (m - a) = 0 )</td>
<td></td>
</tr>
<tr>
<td>distributivity</td>
<td>( a \cdot (b + c) = a \cdot b + a \cdot c )</td>
<td>( (a + b) \cdot c = a \cdot b + a \cdot c )</td>
</tr>
</tbody>
</table>
The additive closure, commutativity, associativity, and inverse imply that \( \mathbb{Z}_m \) is a group w.r.t addition. Identity implies that it is an abelian group. All the properties imply that it is a ring. The inverse allows us subtraction.

**Definition 2** For integers \( a \) and \( b \) and a positive integer \( m \), \( a \equiv b \pmod{m} \) if \( m|(a-b) \).

**Definition 3** Let \( a \in \mathbb{Z}_m \). The multiplicative inverse of \( a \) is \( a^{-1} \in \mathbb{Z}_m \) such that \( aa^{-1} \equiv 1 \pmod{m} \).

Not every integer is invertible (modulo \( m \)). A ring that in which every positive element has a multiplicative inverse is called a field. We will next prove a claim, one of whose implications is that for a prime \( p \), every positive element in \( \mathbb{Z}_p \) is invertible (thus \( \mathbb{Z}_p \) is a field.)

\( a \in \mathbb{Z}_m \) is invertible iff \( \gcd(a,m) = 1 \).

**Proof:** We first show that \( a,2a,\ldots,(m-1)a \) are all distinct modulo \( m \). Assume to the contrary that for some \( i \) and \( j \), \( 1 \leq i < j < m \), for some integers \( \alpha, \beta \), and \( \gamma < m \),

\[
\begin{align*}
  ia &= \alpha m + \gamma \\
  ja &= \beta m + \gamma
\end{align*}
\]

Let \( k = j - i < m \) and \( \delta = \beta - \alpha \). Thus, \( k \cdot a = \delta \cdot m \). Since \( m \) and \( a \) are relatively prime, it follows that they their prime factorizations contains no common primes, and \( m \)'s factors are also \( k \)'s. Consequently, \( k \geq m \), contradicting our choice of \( i \) and \( j \). It now follows that for a unique \( j_a < m \), \((j_a) \cdot a \equiv 1 \pmod{m} \).

**Computing the Multiplicative Inverse**

Use the Extended Euclidean algorithm: To find \( a^{-1} \pmod{m} \), run \( \gcd(a,m) \) to obtain \( x \) and \( y \) s.t. \( ax + ym = 1 \), thus \( ax = 1 \pmod{m} \).

For example, suppose we want to compute \( 3125^{-1} \pmod{9987} \). Then from the
Extended Euclidean Algorithm we obtain:

<table>
<thead>
<tr>
<th>$i$</th>
<th>$q_i$</th>
<th>$x_i$</th>
<th>$y_i$</th>
<th>$r_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>9987</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3125</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>1</td>
<td>$-3$</td>
<td>612</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>$-5$</td>
<td>16</td>
<td>65</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>46</td>
<td>$-147$</td>
<td>27</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>$-97$</td>
<td>310</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>240</td>
<td>$-767$</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>$-577$</td>
<td>1844</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>817</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Therefore, $3125^{-1} \mod 9987 = 1844$ (or, alternatively, $3125 \times 1844 = 5,762,500 = 9987 \times 577 + 1$).

**Chinese Remainder Theorem**

Let $n_1, \ldots, n_k$ be pairwise relatively prime. Let $N = n_1 \cdot \ldots \cdot n_k$. Then the system of congruences:

$$x \equiv a_1 \pmod{n_1}$$
$$x \equiv a_2 \pmod{n_2}$$
$$\ldots$$
$$x \equiv a_k \pmod{n_k}$$

has a unique solution modulo $N$, given by

$$x = \sum_{i=1}^{k} a_i N_i y_i \mod N$$

where each $N_i = N/n_i$ and $y_i = N_i^{-1} \mod n_i$.

**Example:** Suppose $k = 3$, $n_1 = 7$, $n_2 = 11$, $n_3 = 13$, $a_1 = 5$, $a_2 = 3$, and $a_3 = 10$. Thus, $N = 1001$, $N_1 = 143$, $N_2 = 91$, $N_3 = 77$, $y_1 = 5$, $y_2 = 4$, and $y_3 = 12$. It follows that $x = 894 \mod 1001$.

**Euler’s Function**

**Definition 4** For $n \geq 1$, $\phi(n)$ is the number of integers smaller than $n$ that are relatively prime to it.
E.g., $\phi(8) = 4$ (since 1, 3, 5, and 7 are relatively prime to 8.)

For a prime $p$, $\phi(p) = p - 1$: Since 2, \ldots, $p - 1$ are all relatively prime to $p$.

If $p$ and $q$ are prime, and $n = pq$. Then $\phi(n) = \phi(p)\phi(q) = (p - 1)(q - 1)$: The integers that are $< n$ and that are multiples of $p$ or $q$ are:

\[
\begin{align*}
p & \ 2p \ \ldots \ (q - 1)p \\
q & \ 2q \ \ldots \ (p - 1)q
\end{align*}
\]

Thus, $\phi(n)$ is

\[
\left( \frac{pq - 1}{n-1} \right) - \left( \frac{q - 1}{p\text{-multiples}} \right) \left( \frac{p - 1}{q\text{-multiples}} \right) = pq - p - q + 1 = (p-1)(q-1)
\]