This week: Hash functions

1. Definition
2. Applications and properties
3. Constructions (next class)

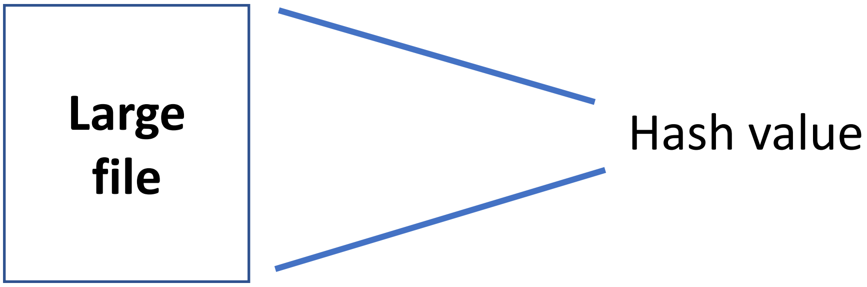
Definition: A **hash function** $H: \{0,1\}^* \rightarrow \{0,1\}^k$ maps strings of arbitrary length to strings of length k.

A hash function is deterministic, efficient, and public (**no secret keys**).

In practice: SHA256 or SHA3 which map strings to $\{0,1\}^{256}$.

Hash functions have many applications in cryptography, and several desired security properties, depending on the application.

Application 1: Authenticating files



Store the large file F on a remote (possibly untrusted) server.

Keep only a (succinct) hash $H(F)$ (of size 256 bits).

When the user wants to use the file, it will fetch it from the server and receive F'.

Check that $H(F') = H(F)$.

To ensure integrity it suffices to use a **collision resistant hash function**

Definition: A hash function $H: \{0,1\}^* \to \{0,1\}^k$ is said to be **collision resistant** if it is hard to find $x \neq x'$ s.t. $H(x) = H(x')$

Note: By the birthday paradox one can find a collision in time roughly $2^{k/2}$. Therefore, to ensure security we need to take $k \approx 256$.

In theory: We consider a **hash family**, where each hash function $H(hk,\cdot)$ is associated with a (public) **hash key** $hk \in \{0,1\}^k$.

The reason we consider of a hash family as opposed to a single hash function is that we model the adversaries as non-uniform Turing machines and such machines can have a collision as non-uniform advice.

Definition: A **hash family** H is said to be **collision resistant** if for every PPT adv A there exists a negligible function μ s.t. for every security parameter $k \in N$,

$$\Pr_{hk \leftarrow \{0,1\}^k}[A(hk) = (x, x') \text{ s.t. } \mathrm{x} \neq x' \wedge \ H(hk, x) = H(hk, x')] \leq \mu(k)$$

Remark: For the above application of file authentication, it suffices for the hash function to be **target collision resistant**!

The target collision resistant property says that an adversary cannot choose x and then given a random hash key hk find x' s.t. $H(hk, x) = H(hk, x')$.

Definition: A hash family H is said to be **target collision resistant** if for every PPT adv (A_1, A_2) there exists a negligible function μ s.t. for every security parameter $k \in N$,

$$\Pr[A_1(1^k) = x \ \wedge A_2(hk, x) = x' \text{ s.t. } H(hk, x) = H(hk, x')] \leq \mu(k)$$

where the prob is over a randomly chosen $hk \leftarrow \{0,1\}^k$.

In practice we use a single hash function (such as SHA256 or SHA3). Thus, we cannot hope to get security against arbitrary non-uniform poly-time Turing machines (which can have a witness hardwired). Moreover, we use a hash function with a fixed security parameter so asymptotic security does not even make sense.

Application 2: Password storage

A server, instead of storing a password pw in the clear,

will store $H(pw)$.

Upon login, the server will check that indeed the received password matches the stored hash value.

The goal is for the hash value to hide the password.

To ensure that the password is not revealed H needs to be a **one-way function.**

Definition: A hash function $H:\{0,1\}^* \to \{0,1\}^k$ is **one-way** if for any efficient adversary A there exists a negligible function μ such that for every $k \in N$,

$$\Pr_{x \leftarrow \{0,1\}^k}[A(H(x) = x' : H(x') = H(x)] \leq \mu(k)$$

Application 3: Hash-then-Sign Paradigm

Goal 1: Extending the message space (and improving efficiency)

Given a signature scheme $(Gen, Sign, Ver)$ for signing messages of length k (i.e., the msg space is $\{0,1\}^k$),
we wish to construct a signature scheme that signs messages of arbitrary length, (i.e., the msg space is $\{0,1\}^*$).

Idea: Instead of signing $m \in \{0,1\}^*$ sign $H(m) \in \{0,1\}^k$

Formally, the new signature is $(Gen, Sign', Ver')$:

$Sign'(sk, m) = Sign\big(sk, H(m)\big)$

$Ver'(pk, m, \sigma) = 1$ iff $Ver(pk, H(m), \sigma) = 1$

Goal: Ensure that if $(Gen, Sign, Ver)$ is secure against adaptive chosen msg attacks then so is $(Gen, Sign', Ver')$.

The property we need from the hash function is **collision resistance**.

Goal 2: Enhancing security.

Suppose we are given a signature scheme $(Gen, Sign, Ver)$ with msg space M that is secure against **random** messages assuming the adversary has seen signatures for **random** messages. Namely, the security guarantee is that for every PPT adv A and every polynomial $t = t(k)$ there exists a negligible function μ s.t. for every security parameter $n \in N$:

$$\Pr[A(pk. m_1, \sigma_1, \dots, m_t, \sigma_t, m^*) = \sigma^* : \ Ver(pk, m^*, \sigma^*) = 1] \leq \mu(k)$$

where the prob. is over $pk \leftarrow Gen(1^k)$ and $m_1, \dots, m_t, m^* \leftarrow \{0,1\}^k$

Goal: Construct a new signature scheme that is existentially unforgeable against adaptive chosen message attacks.

This can be done exactly as before: Namely, the new signature scheme is $(Gen, Sign', Ver')$:

$Sign'(sk, m) = Sign(sk, H(m))$

$Ver'(pk, m, \sigma) = 1$ iff $Ver(pk, H(m), \sigma) = 1$

Claim: $(Gen, Sign', Ver')$ is existentially unforgeable against adaptive chosen message attack in the **Random Oracle Model** (ROM)

Random Oracle Model: Assumes that the hash function is a truly random function, namely, for every $x \in \{0,1\}^*$, $H(x)$ is randomly chosen in $\{0,1\}^k$.

H is huge! It cannot even be written down!

The assumption is that the parties (including the adversary) have black-box access to H.

Application 4: The Fiat-Shamir paradigm.

This paradigm was originally suggested for converting ID schemes into signature schemes but is also used for eliminating interaction from general public-coin interactive proofs (more about this when we will talk about zero knowledge)

As we mentioned last class, the security of the Fiat-Shamir paradigm relies on the **ROM**.

Intuitively, security follows from the fact that interacting with the random oracle is (almost) no different than interacting with the Verifier.

Application 5: Commitment Scheme

A commitment scheme is a digital analogue of a locked box.

It is a randomized function $Com: M \times \{0,1\}^k \to C$

where M is the message space and C is the set of possible commitments.

It should satisfy the following two security requirements:

Statistical Binding: There do not exist distinct msgs $m_1, m_2 \in M$ and $r_1, r_2 \in \{0,1\}^k$ s.t.

$$Com(m_1, r_1) = Com(m_2, r_2)$$

Computational Hiding: For every $m_1, m_2 \in M$,

$$Com(m_1, r_1) \approx Com(m_2, r_2)$$

for random $r_1, r_2 \leftarrow \{0,1\}^k$

One can switch the requirements to require computational hiding and statistical binding:

Computational Binding: It is **computationally hard** to find distinct $m_1, m_2 \in M$ and $r_1, r_2 \in \{0,1\}^k$ s.t.

$$Com(m_1, r_1) = Com(m_2, r_2)$$

Statistical Hiding: For every $m_1, m_2 \in M$,

$$Com(m_1, r_1) \equiv Com(m_2, r_2)$$

for random $r_1, r_2 \leftarrow \{0,1\}^k$, where $\equiv$ denotes statistical closeness

Definition: A family of distributions $\{D_k\}$ and $\{D'_k\}$ are **statistically close** if there exists a negligible function μ s.t. for any (all powerful) A and for every k $\in N$,

$$\Pr[A(x) = 1] - \Pr[A(x') = 1]| \leq \mu(k)$$

where $x \leftarrow D_k$ and $x' \leftarrow D'_k$

Construction: $\boldsymbol{Com(m, r) = H(m||r)}$.

In the ROM this commitment scheme is statistically hiding, assuming $M = \{0,1\}^k$, and is computationally binding.

To get computational binding collision resistance suffices.