Massachusetts Institute of Technology
6.857: Network and Computer Security
Professors Ronald L. Rivest and Yael Tauman Kalai

Handout R2
February 14, 2020
Christos Nestor Chachamis

Recitation 2: Encryption and Hashing

Plan for today

1. Encryption
2. Hashing
   (a) Properties
   (b) Theorems
   (c) Commitments
   (d) Birthday Paradox

1 Encryption

Alice wants to send a secret message to Bob, without an adversary understanding what the secret message is. Some ways to do this is by "walking in the woods", steganography or encryption. Encryption usually reveals that some sort of communication exists.

Encryption involves the encoding of the secret message or information into a ciphertext, usually with the use of a secret key. The ciphertext is then transmitted to the other party and it is decrypted. Encryption is successful ("secure") if Eve cannot understand what the secret message is. In order to prove security we need to make assumptions on what the adversary is allowed to do.

One example of encryption is the one-time pad.

2 Hashing

A hash function is a function $h : \{0,1\}^* \to \{0,1\}^d$ that maps a binary string of arbitrary length into a binary string of fixed length. In other words, compressing strings of any length into strings of convenient length.

2.1 Properties

In class we saw some desirable properties for a hash function $h(\cdot)$:

- **One - Wayness**: This means that the hash function is hard to invert. That is, given a value $z = h(x)$ for some x, it should be hard to find an x' such that $h(x') = z = h(x)$. This does not mean that we find the exact x, but rather any x' that is mapped to z.
- **Collision Resistance**: This property guarantees that it is hard to find two strings $x' \neq x$, such that $h(x) = h(x')$. Note that collisions should exist, since the domain of the hash function is much larger than its range. So, it is not impossible to find such a pair, but rather it will take a long time to find one.
- **Target Collision Resistance**: This is similar to collision resistance. TCR states that, given an input x, it is hard to find $x' \neq x$ such that $h(x') = h(x)$.

2.2 Theorems

In class we saw the following theorem:

Theorem 1 *If a hash function is collision resistant, then it is one-way.*

Proof. Suppose on the contrary that h is not one-way. Then, there exists an efficient process A that computes $A(v) = Z$ with $h(z) = v$. Then, we can find a collision with high probability using the following method: try x at random, compute $x' = A(h(x))$ and return (x, x') if $x \neq x'$. If the preimage of $h(x)$ is a singleton then we cannot find $x' \neq x$. However, this happens with small probability.

In general, we can show that

$$CR \Rightarrow TCR \Rightarrow OW$$

2.3 Commitments

In class we saw that one application of hashing is commitments. In particular, a commitment scheme allows a user to commit to a particular value (for example a bid for an auction). We defined the commitment function to be

$$\text{com}(x, r) = h(x||r)$$

We didn't really discuss why we need the randomness there. One reason is that there are not really many values for the auction price, so we need the extra bits of randomness to increase the domain.

Let's prove the following theorem:

Theorem 2 *If h is CR, then commitment is CR.*

Proof. Suppose that commitment is not CR. Then we can find pairs $(x_1, r_1) \neq (x_2, r_2)$ such that

$$\text{com}(x_1, r_1) = \text{com}(x_2, r_2)$$

Define $z_i = x_i||r_i$. Then $z_1 \neq z_2$ and

$$h(z_1) = h(x_1||r_1) = h(x_2||r_2) = h(z_2),$$

so we have found a collision in h, contradiction.

2.4 Birthday Paradox and Attack

Original formulation: In a room of 23 people, there is a 50% chance that two of them have the same birthday.

This gives rise to the birthday attack: simply check random pairs until a collision is found, or you run out of resources. Note: for this to work, we need to utilize a data structure like hash tables for fast lookup.

Theorem 3 *It is possible to find a collision in $O(\sqrt{2^d}) = O(2^{\frac{d}{2}})$ time for a hash function of pre-image power 2^d.*

Proof. The expected number of collisions we find after iterating n times is

$$\mathbb{E}[\#\text{collisions}] = \sum_{i,j}^{n} \Pr[h(x_i) = h(x_j)] = \binom{n}{2} 2^{-d} \approx n^2 2^{-d},$$

so by setting $n^2 2^{-d} = 1$ or $n = 2^{\frac{d}{2}}$ we are expected to find a collision.