MATH 314 Spring 2018 - Class Notes

4/30/2015

Scribe: Mark Padilla

Summary: Hash Functions and their Properties

Notes:

- **Hash Functions** $h(x)$
  - A functions that takes in a long message m and produces a much shorter output called a **digest**
  - If two different inputs to a hash functions produce the same **digest**, we call this a **collision**
  - Every has function has collisions (Pigeon Hole Principle)
    - Ideally, we'd like them (collisions) to be hard to find

- **Ideal Properties of a Cryptographic Hash**
  1. Pre-Image Resistant
     - For any possible digest y, it should be hard to find an input x such that $h(x) = y$ (Hard to invert $h(x)$)
  2. Weak-Collision Resistant
     - For any input m, it should be hard to find another input n such that $h(m) = h(n)$
  3. Strong-Collision Resistant
     - It should be hard to find any two inputs m and n such that $h(m) = h(n)$

- **Discrete Log Hash Function**
  - Pick two primes p and q where $q = 2p + 1$
  - Inputs are numbers between 1 and $p^2 - 1$
    - Take the input x and write it in **"base p"**
      - $x = a_0 + a_1 p$ where $0 <= a_0 <= p - 1$ and $0 <= a_1 <= p - 1$
  - Pick two different primitive roots α and β $(mod\ p)$
    - $h(x) = h(a_0 + a_1 p) = \alpha^{a_0} \beta^{a_1}$
  - $h(x)$ takes input of size about p^2 and produces output of size about $q \approx \sqrt{p^2} = p$

- About p^2 numbers are getting mapped to one of $2p+1$ outputs so there are lots of collisions
- Nevertheless, finding a collision is hard in the sense that producing a single collision is as hard as solving the **discrete log problem** $\alpha = \beta^c \ (mod\ q)$

- **Why is this?**
  - Suppose you find a collision to this discrete log hash. You find $h(x) = h(y)$
    - $x = a_0 + a_1 p \ base\ p$
    - $x = b_0 + b_1 p \ base\ p$
    - $h(x) = \alpha^{a_0}\beta^{a_1} \equiv \alpha^{b_0}\beta b_1 \ (mod\ q)$
  - This means
    - $\alpha^{a_0 - b_0}\beta^{a_1 - b_1} \equiv 1 \ (mod\ q)$
  - Plug in $\alpha \equiv \beta^c (mod\ q)$
    - $(\beta^c)^{a_0 - b_0}\beta^{a_1 - b_1} \equiv 1 \ (mod\ q)$
  - This means
    - $\beta^{c(a_0 - b_0) + a_1 - b_1} \equiv \ (mod\ q)$
  - Because β is a **primitive root**, the only way this can happen is if $c(a_0 - b_0) + a_1 - b_1$ is a multiple of $q-1$
  - This means
    - $c(a_0 - b_0) + a_1 - b_1 \equiv 0 \ (mod\ q-1)$
  - Found c such that $\alpha \equiv \beta^c \ (mod\ q)$
  - **This means the discrete log hash has strong collision resistance**

- **Disadvantages of Discrete Log Hash**
  - Can't deal with arbitrarily long inputs
  - Slow compared to other hashes
    - SHA-1 or SHA-2 is used much more instead

- **Bad Example of a Cryptographical Hash Function**
  - Fix a large number n
  - $h(x) = x \ (mod\ n)$
  - This is not pre-image resistant or weak collision resistant. Easy to find lots of collisions

- **Birthday Paradox**

- How many people do you need in a room before it is likely that two people share a birthday?
- How many people do we need before the probability is greater than 50%
- Suprisingly, the answer is only 23 people

- 365 days and k people, the probability that no two people share a birthday is $1(\frac{364}{365})(\frac{363}{365})(\frac{362}{365})...(\frac{366-k}{365}) \approx e^{\frac{-k^2}{365}}$
- In general, if we have N randomly chosen objects chosen with replacement k times, the probability that no two objects are chosen twice is approximately $e^{\frac{-k^2}{N}}$
- **IMPORTANT:** if $k > \sqrt{N}$, this problem is quickly going to 0

- **Message "Signing"**
  - One way to "sign" a message m would be to put a signature after the message which is $h(m)$ for some hash function
  - Now you send the message $(m, h(m))$
  - Bob can check if a message is valid by computing $h(m)$ for the message he receives and seeing if it matches the signature
  - This protects against accidental mistakes, but if Eve is maliciously changing the message, she could change the has too
  - Signature has nothing to do with Alice
  - Goal: Use public key cryptography to create a signature that only Alice could have produces but that anyone can check is from Alice.