Lecture 17: Composing Hashes

Outline

- We will consider some techniques of composing hash functions
- Moreover, we aim to understand why they work or do not work

Iterated Hash I

Setting

- Suppose we are given sets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ such that $|\mathcal{A}| \geqslant |\mathcal{B}| \geqslant |\mathcal{C}|$
- Suppose $\mathcal{H}$ is a hash function family from the domain $\mathcal{A}$ to the range $\mathcal{B}$
- Suppose $\mathcal{G}$ is a hash function family from the domain $\mathcal{B}$ to the range $\mathcal{C}$
- We are interested in constructing a new hash function family with domain $\mathcal{A}$ and range $\mathcal{C}$

Iterated Hash II

- Suppose we define the following family of hash functions

$$\mathcal{I} = \{g \circ h \colon h \in \mathcal{H}, g \in \mathcal{G}\},$$

where we define $(g \circ h)(x) := g(h(x))$. These hash functions have domain $\mathcal{A}$ and range $\mathcal{C}$

Question

Does this new family of hash functions $\mathcal{I}$ inherit good properties from the hash function families $\mathcal{H}$ and $\mathcal{G}$?

- Next we shall formalize one such question. Note that there can be multiple such questions. We only illustrate using one question

Formal Question

Suppose the collision probabilities of the hash function families $\mathcal{H}$ and $\mathcal{G}$ are α and β respectively. That is,

$$\text{For distinct } x_1, x_2 \in \mathcal{A}, \text{ we have } \mathbb{P}\left[h(x_1) = h(x_2) \colon h \xleftarrow{\$} \mathcal{H}\right] = \alpha$$

$$\text{For distinct } y_1, y_2 \in \mathcal{B}, \text{ we have } \mathbb{P}\left[g(y_1) = g(y_2) \colon g \xleftarrow{\$} \mathcal{G}\right] = \beta$$

What is the collision probability of the new hash function family $\mathcal{I}$?

Iterated Hash IV

- Let us begin our analysis
- For distinct $x_1, x_2 \in \mathcal{A}$, we are interested in computing the probability

$$\mathbb{P}\left[(g \circ h)(x_1) = (g \circ h)(x_2) \colon (g \circ h) \xleftarrow{\$} \mathcal{I}\right]$$

- Note that we can express this collision probability as follows

$$\mathbb{P}\left[(g \circ h)(x_1) = (g \circ h)(x_2) \colon (g \circ h) \xleftarrow{\$} \mathcal{I}\right]$$
$$= \mathbb{P}\left[(g \circ h)(x_1) = (g \circ h)(x_2) \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right]$$

- Let us represent $y_1 = h(x_1)$ and $y_2 = h(x_2)$

Iterated Hash V

- Now, we can write

$$
\begin{aligned}
&\mathbb{P}\left[(g\circ h)(x_1) = (g\circ h)(x_2) \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right] \\
=&\mathbb{P}\left[(g\circ h)(x_1) = (g\circ h)(x_2), y_1 = y_2 \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right] \\
&+\mathbb{P}\left[(g\circ h)(x_1) = (g\circ h)(x_2), y_1 \neq y_2 \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right]
\end{aligned}
$$

- Note that if $y_1 = y_2$, then we will surely have $(g\circ h)(x_1) = (g\circ h)(x_2)$. So, the probability expression

$$
\mathbb{P}\left[(g\circ h)(x_1) = (g\circ h)(x_2), y_1 = y_2 \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right]
$$

is identical to

$$
\mathbb{P}\left[y_1 = y_2 \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right] = \mathbb{P}\left[y_1 = y_2 \colon h \xleftarrow{\$} \mathcal{H}\right] = \alpha
$$

- Note that if $y_1 \neq y_2$, then we can write

$$\begin{aligned}
&\mathbb{P}\left[(g \circ h)(x_1) = (g \circ h)(x_2), y_1 \neq y_2 \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right] \\
=&\mathbb{P}\left[(g \circ h)(x_1) = (g \circ h)(x_2) | y_1 \neq y_2 \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right] \\
&\times \mathbb{P}\left[y_1 \neq y_2 \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right] \\
=&\mathbb{P}\left[g(y_1) = g(y_2) | y_1 \neq y_2 \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right] \\
&\times \mathbb{P}\left[y_1 \neq y_2 \colon h \xleftarrow{\$} \mathcal{H}\right] \\
=&\beta(1 - \alpha)
\end{aligned}$$

Iterated Hash VII

- Adding these two expressions, we get

$$\mathbb{P}\left[(g \circ h)(x_1) = (g \circ h)(x_2) \colon (g \circ h) \xleftarrow{\$} \mathcal{I}\right]$$
$$= \alpha + \beta(1 - \alpha) = \alpha + \beta - \alpha\beta$$

- Note that if we have $\alpha = 1/|\mathcal{B}|$ and $\beta = 1/|\mathcal{C}|$, the collision probability of the new hash function family is more than both α and β
- This is not a good universal hash function family (according to the way we have defined our universal hash function family)

Concatenation I

- Suppose there are two hash function families $\mathcal{H}$ and $\mathcal{G}$ with domain $\mathcal{D}$ for both the families, and range $\mathcal{R}$ and $\mathcal{R}'$, respectively
- Suppose the collision probability of the hash function families $\mathcal{H}$ and $\mathcal{G}$ are α and β, respectively. That is, for any distinct $x_1, x_2 \in \mathcal{D}$ we have

$$\mathbb{P}\left[h(x_1) = h(x_2) \colon h \xleftarrow{\$} \mathcal{H}\right] = \alpha$$

$$\mathbb{P}\left[g(x_1) = g(x_2) \colon g \xleftarrow{\$} \mathcal{G}\right] = \beta$$

- Now, consider the new hash function family from the domain $\mathcal{D}$ to the range $\mathcal{R} \times \mathcal{R}'$.

$$\mathcal{I} = \left\{(h \| g) \colon h \in \mathcal{H}, g \in \mathcal{G}\right\},$$

where $(h\|g)(x) = h(x)\|g(x)$ (the concatenation of $h(x)$ and $g(x)$ is represented by $h(x)\|g(x)$)

Concatenation II

- Is this a good family of hash functions? In particular, will this hash function family have low collision probability if $\mathcal{H}$ and $\mathcal{G}$, each, have low collision probabilities?
- Let us analyze the collision probability of this new hash function family. For distinct $x_1, x_2 \in \mathcal{D}$, we are interested in the probability

$$\mathbb{P}\left[(h\|g)(x_1) = (h\|g)(x_2) \colon (h\|g) \xleftarrow{\$} \mathcal{I}\right]$$

- This can equivalently be written as

$$\mathbb{P}\left[h(x_1) = h(x_2), g(x_1) = g(x_2) \colon h \xleftarrow{\$} \mathcal{H}, g \xleftarrow{\$} \mathcal{G}\right]$$

- We want this collision probability expression to be $\alpha\beta$. But the events $h(x_1) = h(x_2)$ and $g(x_1) = g(x_2)$ can be related! We will explain this further in the next few slides.

Concatenation III

Lesson Learned

Blindly iterating or concatenating hash functions families might yield worse hash function families. We need to be smart in combining hash functions!

First Example I

Problem

Suppose the domain is $\mathcal{D} = \mathbb{F}^n$ and the range is $\mathcal{R} = \mathbb{F}$, for a field $(\mathbb{F}, +, \times)$. We want to design 2-wise independent hash function families from $\mathcal{D}$ to $\mathcal{R}$.

First Example II

- **First Proposed Solution.** In the class, the following solution was first proposed

$$\mathcal{H} = \left\{ h_{a_1,\dots,a_n} \colon a_1, \dots, a_n \in \mathbb{F} \right\},$$

where the function
$h_{a_1,\dots,a_n}(x_1, \dots, x_n) := a_1x_1 + a_1x_2 + \cdots + a_nx_n$

- Note that for $x = (\overbrace{0, 0, \dots, 0}^{n\text{-times}})$ the probability

$$\mathbb{P}\left[h(x) = 0 \colon h \xleftarrow{\$} \mathcal{H} \right] = 1$$

So, this hash function family is not even 1-wise independent, let alone 2-wise independent

First Example III

- **How to fix this?** The first observation is the following. For a non-zero $x \in \mathbb{F}^n$ and any $y \in \mathbb{F}$, we have

$$\mathbb{P}\left[h(x) = y \colon h \xleftarrow{\$} \mathcal{H}\right] = 1/|\mathbb{F}|$$

  So, the "flaw" in our hash function family exists only when $x = 0^n$; otherwise not.

- So, let us prepend 1 to the input x. This will always ensure that x is non-zero!
- Now, we define the new hash function family

$$\mathcal{H} = \left\{h_{a_0,a_1,\dots,a_n} \colon a_0 a_1, \dots, a_n \in \mathbb{F}\right\},$$

  where the function $h_{a_0,a_1,\dots,a_n}(x_1,\dots,x_n) := a_0 \cdot 1 + a_1 x_1 + a_1 x_2 + \cdots + a_n x_n = a_0 + a_1 x_1 + a_1 x_2 + \cdots + a_n x_n$

First Example IV

- This new hash function family has the property that, for distinct $x, x' \in \mathbb{F}^n$ and $y, y' \in \mathbb{F}$, we have

$$\mathbb{P}\left[h(x) = y, h(x') = y' : h \xleftarrow{\$} \mathcal{H}\right] = \frac{1}{|\mathbb{F}|^2}$$

- So, this is a 2-wise independent hash function family

Problem

Suppose the domain is $\mathcal{D} = \mathbb{F}^n$ and the range is $\mathcal{R} = \mathbb{F}^2$, for a field $(\mathbb{F}, +, \times)$. We want to design 2-wise independent hash function families from $\mathcal{D}$ to $\mathcal{R}$.

Second Example II

- In the previous example, we constructed a 2-wise independent hash function family $\mathcal{H}$ from domain $\mathbb{F}^n$ to the range $\mathbb{F}$
- Using the "concatenation idea" we can now try to define the hash function family from domain $\mathbb{F}^n$ to the range $\mathbb{F}^2$

Second Example III

- **First Idea.** In the class, the proposed idea was to pick to hash functions $h, h' \xleftarrow{\$} \mathcal{H}$ independently at random, and output the hash $\big(h(x), h'(x)\big)$
- Suppose the first hash function is $h = h_{a_0,a_1,\dots,a_n}$ and the second hash function is $h' = h_{b_0,b_1,\dots,b_n}$
- For any $\lambda \in \mathbb{F}$, when $b_0 = \lambda a_0$, $b_1 = \lambda a_1$, $\dots$, $b_n = \lambda a_n$, we have a problem. In this case $h'(x) = \lambda h(x)$ always.
- Think: Why is this an issue? Why is the hash function family not 2-wise independent?

- **Fixing this Issue.** We shall fix this issue iteratively.
- We can prove that if it is not the case that $b_0 = \lambda a_0$, $b_1 = \lambda a_1, \dots, b_n = \lambda a_n$, for some $\lambda \in \mathbb{F}$, then $h'(x)$ is independent and uniformly random over $\mathbb{F}$
- So, the following hash function family is 2-wise independent from the domain $\mathbb{F}^n$ to the range $\mathbb{F}^2$. The hash function family is defined by matrices of rank 2 of the form

$$\begin{pmatrix} a_0 & b_0 \\ a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{pmatrix}$$

- The evaluation of the hash function at $x = (x_1, x_2, \ldots, x_n)$ is provided by the following matrix multiplication

$$(1, x_1, x_2, \ldots, x_n)\begin{pmatrix} a_0 & b_0 \\ a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{pmatrix}$$

Problem

Suppose the domain is $\mathcal{D} = \mathbb{F}^n$ and the range is $\mathcal{R} = \mathbb{F}^{n'}$, for a field $(\mathbb{F}, +, \times)$, where $n' < n$. We want to design 2-wise independent hash function families from $\mathcal{D}$ to $\mathcal{R}$.

Third Example II

- The hash function families are defined by the matrices of column rank n' of the following form

$$\begin{pmatrix} a_{0,1} & a_{0,2} & \cdots & a_{0,n'} \\ a_{1,1} & a_{1,2} & \cdots & a_{1,n'} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n'} \end{pmatrix}$$

- The evaluation of the above hash function at $x = (x_1, \ldots, x_n)$ is defined by the matrix multiplication

$$(1, x_1, x_2, \ldots, x_n) \cdot \begin{pmatrix} a_{0,1} & a_{0,2} & \cdots & a_{0,n'} \\ a_{1,1} & a_{1,2} & \cdots & a_{1,n'} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n'} \end{pmatrix}$$

- This hash function family is 2-wise independent

Lesson Learned

- Concatenation works well, but we have to be careful which functions we choose to concatenate (choosing functions independently might not be a good idea)!