Recall

- Collision-resistant Hash Function family from domain $D$ to range $R$ is a set of hash functions $\mathcal{H} = \{ h^{(i)} : i \in I \}$, where $I$ is the set of indices and each function $h^{(i)} : D \rightarrow R$
- Any efficient adversary given $h^{(i)}$, where $i \leftarrow I$, can output $x, x' \in D$ such that $h^{(i)}(x) = h^{(i)}(x')$ only with negligible probability
- One bit compressing (i.e., $|D| = 2 |R|$) can be constructed from the hardness of the discrete logarithm assumption as follows. Let the discrete logarithm problem be hard in the group $G$, then for $b \in \{0, 1\}$ and $x \in \mathbb{Z}_{|G|}$, we have:

$$h^{(y)} : \{0, 1\} \times \mathbb{Z}_{|G|} \rightarrow G$$

$$h^{(y)}(b, x) = y^b g^x$$

$$\mathcal{H} = \{ h^{(y)} : y \in G \}$$
We can construct a \( t \)-bit compression function as follows: Let \( b \in \{0, 1\}^t \) and \( y^{(1)}, \ldots, y^{(t)} \in \mathbb{Z}_{|G|} \).

\[
h(y^{(1)}, \ldots, y^{(t)})(b, x) = y^{(1)}^{b_1} \cdots y^{(t)}^{b_t} g^x
\]

Each function is indexed by \((y^{(1)}, \ldots, y^{(t)})\) and each \(y^{(i)} \in \{0, 1\}^n\). So, index size is \(tn\).

- **Prove:** If Discrete Logarithm assumption holds in \(G\) then the construction above is a CRHF, where \(t = \text{poly}(n)\)
- **Prove:** If \(H^{(n)}\) is a CRHF family with functions \(\{0, 1\}^{n+1} \rightarrow \{0, 1\}^n\), for all large enough \(n\), then the construction above is a CRHF family, where \(t = \text{poly}(n)\)
- **Think:** What is the difference between the above two theorems
In particular, with $t = n$ and $G = \{0, 1\}^n$, the previously constructed function is a length halving family of hash functions where all functions are $\{0, 1\}^{2n} \rightarrow \{0, 1\}^n$
Tree-based Hashing

- We are interested in hashing $\{0, 1\}^{tn}$ down to $\{0, 1\}^n$
- One-bit compression at a time needs $(t - 1)n \times n$ size indices. Can we do better?
Let $\mathcal{H}$ be a CRHF family with functions $\{0, 1\}^{2n} \rightarrow \{0, 1\}^n$ and key size $K$.

We will construct CRHF family $\mathcal{H}^{(t)}$ with functions $\{0, 1\}^{tn} \rightarrow \{0, 1\}^n$ and key size $K$, for $t \geq 2$.

Let $x \in \{0, 1\}^{tn}$ be represented as $(x^{(1)}, \ldots, x^{(t)})$, where each $x^{(i)} \in \{0, 1\}^n$. The function is calculated in an iterated fashion as represented below. Each box represents an application of a function $h \in \mathcal{H}$ and the output of the hash function is $y$. Call this new function $\text{itr}_t(h)$ function. So, we have $\mathcal{H}^{(t)} = \{\text{itr}_t(h) : h \in \mathcal{H}\}$. 

\[
\begin{align*}
x^{(1)} & \quad x^{(2)} & \quad x^{(3)} & \quad x^{(4)} & \quad \cdots & \quad x^{(t)} \\
& \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
& \quad & \quad & \quad & \quad & \quad & y
\end{align*}
\]
Proof

- Our adversary $\tilde{A}$ on input a hash function $h$ feeds $\text{itr}_t(h)$ function to $A^*$
- Suppose $A^*$ produces $x = (x^{(1)}, \ldots, x^{(t)})$ and $z = (z^{(1)}, \ldots, z^{(t)})$ such that it is a collision of the function $\text{itr}_t(h)$ function
- Suppose the input to the last $h$-box in the evaluation of $\text{itr}_t(h)(x)$ is $a$ and the input to the last $h$-box in the evaluation of $\text{itr}_t(h)(z)$ is $b$. We know that the output of the last $h$-box is same in these two cases. If $a \neq b$, then we have found a collision.
- If $a = b$, then the output of the second last $h$-box is identical in $\text{itr}_t(h)(x)$ and $\text{itr}_t(h)(z)$ evaluation. Therefore, we can recurse on $(x^{(1)}, \ldots, x^{(t-1)})$ and $(z^{(1)}, \ldots, z^{(1)t - 1})$ that also produce a collision (i.e. the output of the second last $h$-box)