COMP 8920: Cryptography Winter 2025

Lecture 13 — Feb 25, 2025

Prof. Curtis Bright *Scribe: Rajat Yadav*

Merkle-Damgård Construction

Suppose $c : \mathbb{Z}_2^{m+t} \to \mathbb{Z}_2^m$ is a collision-resistant compression function. For now, take $t \geq 2$. We'll use c to construct a collision resistant hash function $h : \mathcal{X} \to \mathbb{Z}_2^m$ where $\mathcal{X} = \bigcup_{i=(m+t+1)}^{\infty} \mathbb{Z}_2^i$. Suppose $x \in \mathcal{X}$ with $|x|n \geq m + t + 1$ and we express

$$x \text{ as } x_1 \parallel x_2 \parallel ... \parallel x_k$$

where $|x_1| = |x_2| = ... = |x_{k-1}| = t - 1$ and $|x_k| = t - 1 - Q(0 \leq d < t - 1)$. Note that $k = \lceil (\frac{n}{t-1}) \rceil$ and $d = k(t-1) - n$. Now, set $y = x_1, ..., y_{k-1} = x_{k-1},$ and $y_k = x_k \parallel 0^d$ and one additional block $y_{k+1} = \{d \text{ is binary}\}$. Note all $|y_i| = t - 1$ for all $1 \leq i \leq k + 1$.

Let

$$z_1 = 0^{m+1} \parallel y_1 \text{ and } g_1 = c(z_1) \tag{1}$$
$$z_2 = g_1 \parallel 1 \parallel y_2 \text{ and } g_2 = c(z_2) \tag{2}$$
$$\vdots \tag{3}$$
$$z_{k+1} = g_k \parallel 1 \parallel y_{k+1} \tag{4}$$

Finally, $h(x) = g_{k+1}$. Note all y_is were defined so that $x_i \to y_i$ is injective. we now prove that h is collision-resistant assuming c is collision-resistant. We do this by constructing a collision for c, assuming that we can find a collision for h (contrapositive of above).

Proof. Suppose we have a collision (x, x') for h, and denote $y(x) = y_1 \parallel ... \parallel y_{k+1}$, $y_{x'} = y'_1 \parallel ... \parallel y'_{l+1}$ where x is padded with d zeros and x' is padded with d' zeros.

The g values for x and x' will be denoted by g_is and g'_is.

- **Case 1:** Suppose $|x| \equiv |x'| \bmod (t-1)$ (i.e, $d \neq d'$). Thus $y_{k+1} = \{d \text{ in binary}\} \neq y'_{l+1} = \{d' \text{ in binary}\}$. Then $h(x) = c(g_k \parallel 1 \parallel y_{k+1})$ $h(x') = c(g'_l) \parallel 1 \parallel y'_{l+1})$ but $h(x) = h(x')$, so $(g_k \parallel 1 \parallel y_{k+1}, g'_l \parallel 1 \parallel y'_{l+1}$ is a collision for c, since the last $(t-1)$ bits are different $(y_{k+1} \neq y'_{l+1})$.
- **Case 2:** $|x| \equiv |x'| \bmod (t-1)$, i.e, $d = d'$.
  - **Case 2a:** $|x| = |x'|$, so $k = l$. Like in case 1, we have $c(g_k \parallel 1 \parallel y_{k+1}) = c(g'_k \parallel 1 \parallel y'_{k+1})$. If $g_k \neq g'_k$ then we have a collision for c. However, if $g_k = g'_k$ then this is ¬ a collision, since the input strings $g_k \parallel 1 \parallel y_k = g'_k \parallel 1 \parallel y'_{k+1}$. Then $c(g_k \parallel 1 \parallel y_k) = g_k = g'_k = c(g'_{k-1} \parallel 1 \parallel y'_k)$ if $y_k \neq y'_k$ or $g_{k-1} \neq g'_{k-1}$ then this is a collision. Otherwise, $g_{k-1} = g'_{k-1}$,

and we continue this process until eventually $c(0^{m+1} \parallel y_1) = g_1 = g'_1 = c(0^{m+1} \parallel y'_1)$. If $g_1 \neq y'_1$ then this is a collision and we are done. Otherwise, if $y_1 = y'_1$, then we have

$$y_1 = y'_1 \tag{5}$$
$$y_2 = y'_2 \tag{6}$$
$$\vdots \tag{7}$$
$$y_k = y_{k'} \tag{8}$$
$$y_{k+1} = y'_{k+1} \tag{9}$$

which imply $y(x) = y(x')$, but since y is an injection, $x = x'$ which we assumed was $\neg$ the case, so this cannot happen.

– **Case 2b:** $|x| \neq |x'|$ and without loss of generality, suppose we suppose $|x| > |x'|$, so $k > l$. This case proceeds like in 2a, and we eithner find a collision of the form $(g_k \parallel 1 \parallel y_{k+1}, g'_l \parallel 1 \parallel y'_{l+1})$ (or for smaller indices of $g_k, y_k + 1$, etc). or, eventually, if this does not find a collision, we eventually arrive at $c(g_{k-l} \parallel 1 \parallel y_{k-1+l}) = c(0^m \parallel 0 \parallel y'_1)$ but the $(m+1)$th bit of the left string is 1, but the $(m+1)$th bit of the right string is 0, so this must be a collision.

□