CRHF from Dlog

Let $G$ be a group generation algorithm that outputs a prime order group.
CRHF from Dlog

Let $\mathcal{G}$ be a group generation algorithm that outputs a prime order group.

**Fixed-Length CRHF**

$$\text{Gen}(1^n): \text{Run } (G, q, g) \leftarrow \mathcal{G}(1^n). \text{ Select } h \leftarrow G.$$  
$$\text{Output } s = (G, q, g, h).$$

$$H_s(x_1, x_2): \text{on input } (x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q, \text{ output } g^{x_1} h^{x_2} \in \mathcal{G}$$
CRHF from Dlog

Let $G$ be a group generation algorithm that outputs a prime order group.

**Fixed-Length CRHF**

Gen($1^n$): Run $(G, q, g) \leftarrow G(1^n)$. Select $h \leftarrow G$.

Output $s = (G, q, g, h)$.

$H^s(x_1, x_2)$: on input $(x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$, output $g^{x_1} h^{x_2} \in G$

**Theorem:** If the discrete logarithm problem is hard relative to $G$, then the construction above is a fixed-length, collision resistant hash function.
Let $\mathcal{G}$ be a group generation algorithm that outputs a prime order group.

### Fixed-Length CRHF

$\text{Gen}(1^n)$: Run $(G, q, g) \leftarrow \mathcal{G}(1^n)$. Select $h \leftarrow G$.

Output $s = (G, q, g, h)$.

$H^s(x_1, x_2)$: on input $(x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$, output $g^{x_1} h^{x_2} \in \mathcal{G}$

**Theorem:** If the discrete logarithm problem is hard relative to $\mathcal{G}$, then the construction above is a fixed-length, collision resistant hash function.

**Proof idea:** Let $\Pi = (\text{Gen}, H)$ as described above. Suppose there exists a p.p.t. adversary $\mathcal{A}$ such that $\text{Hash-Coll}_{\mathcal{A}, \Pi}(n) = \epsilon(n)$. We’ll show $\mathcal{A}_r$ that solves the discrete logarithm problem with the same probability.
CRHF from Dlog

Let $G$ be a group generation algorithm that outputs a prime order group.

**Fixed-Length CRHF**

\[ \text{Gen}(1^n): \text{Run } (G, q, g) \leftarrow G(1^n). \text{ Select } h \leftarrow G. \]
\[ \text{Output } s = (G, q, g, h). \]

\[ H^s(x_1, x_2): \text{on input } (x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q, \text{ output } g^{x_1} h^{x_2} \in G \]

**Theorem:** If the discrete logarithm problem is hard relative to $G$, then the construction above is a fixed-length, collision resistant hash function.

**Proof idea:** Let $\Pi = (\text{Gen}, H)$ as described above. Suppose there exists a p.p.t. adversary $A$ such that $\text{Hash-Coll}_{A, \Pi}(n) = \epsilon(n)$. We’ll show $A_r$ that solves the discrete logarithm problem with the same probability.

$A_r$ receives challenge $(G, q, g, h)$ and has to find $x$ such that $g^x = h$. 

CRHF from Dlog

Let $G$ be a group generation algorithm that outputs a prime order group.

**Fixed-Length CRHF**

Gen($1^n$): Run $(G, q, g) \leftarrow G(1^n)$. Select $h \leftarrow G$.
Output $s = (G, q, g, h)$.

$H^s(x_1, x_2)$: on input $(x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$, output $g^{x_1} h^{x_2} \in G$

**Theorem:** If the discrete logarithm problem is hard relative to $G$, then the construction above is a fixed-length, collision resistant hash function.

Proof idea: Let $\Pi = (Gen, H)$ as described above. Suppose there exists a p.p.t. adversary $A$ such that Hash-Coll$_{A, \Pi}(n) = \epsilon(n)$. We'll show $A_r$ that solves the discrete logarithm problem with the same probability.

$A_r$ receives challenge $(G, q, g, h)$ and has to find $x$ such that $g^x = h$.
$A_r$ sends $s = (G, q, g, h)$ to $A$, who returns $x = (x_1, x_2)$ and $\hat{x} = (\hat{x}_1, \hat{x}_2)$. 
CRHF from Dlog

Let \( G \) be a group generation algorithm that outputs a prime order group.

**Fixed-Length CRHF**

\[
\text{Gen}(1^n): \text{Run } (G, q, g) \leftarrow G(1^n). \text{ Select } h \leftarrow G. \\
\text{Output } s = (G, q, g, h).
\]

\[
H^s(x_1, x_2): \text{on input } (x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q, \text{ output } g^{x_1} h^{x_2} \in G
\]

**Theorem:** If the discrete logarithm problem is hard relative to \( G \), then the construction above is a fixed-length, collision resistant hash function.

Proof idea: Let \( \Pi = (\text{Gen}, H) \) as described above. Suppose there exists a p.p.t. adversary \( A \) such that Hash-Coll\( A, \Pi(n) = \epsilon(n) \). We’ll show \( A_r \) that solves the discrete logarithm problem with the same probability.

\( A_r \) receives challenge \((G, q, g, h)\) and has to find \( x \) such that \( g^x = h \).

\( A_r \) sends \( s = (G, q, g, h) \) to \( A \), who returns \( x = (x_1, x_2) \) and \( \hat{x} = (\hat{x}_1, \hat{x}_2) \).

If \( h = 1 \), return \( x = 0 \)

Otherwise, return \([(x_1 - \hat{x}_1)(\hat{x}_2 - x_2)^{-1} \mod q] \).
CRHF from Dlog

Let $\mathcal{G}$ be a group generation algorithm that outputs a prime order group.

**Fixed-Length CRHF**

$\text{Gen}(1^n)$: Run $(G, q, g) \leftarrow \mathcal{G}(1^n)$. Select $h \leftarrow G$.

Output $s = (G, q, g, h)$.

$H^s(x_1, x_2)$: on input $(x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$, output $g^{x_1} h^{x_2} \in G$

**Theorem:** If the discrete logarithm problem is hard relative to $\mathcal{G}$, then the construction above is a fixed-length, collision resistant hash function.

Proof idea: Let $\Pi = (\text{Gen}, H)$ as described above. Suppose there exists a p.p.t. adversary $\mathcal{A}$ such that Hash-Coll$_{\mathcal{A}, \Pi}(n) = \epsilon(n)$. We’ll show $\mathcal{A}_r$ that solves the discrete logarithm problem with the same probability.

$\mathcal{A}_r$ receives challenge $(G, q, g, h)$ and has to find $x$ such that $g^x = h$.

$\mathcal{A}_r$ sends $s = (G, q, g, h)$ to $\mathcal{A}$, who returns $x = (x_1, x_2)$ and $\hat{x} = (\hat{x}_1, \hat{x}_2)$.

If $h = 1$, return $x = 0$

Otherwise, return $[(x_1 - \hat{x}_1)(\hat{x}_2 - x_2)^{-1} \mod q]$.

Analysis:

$$H^s(x_1, x_2) = H^s(\hat{x}_1, \hat{x}_2)$$

$$\Rightarrow g^{x_1} h^{x_2} = g^{\hat{x}_1} h^{\hat{x}_2}$$
CRHF from Dlog

Let $G$ be a group generation algorithm that outputs a prime order group.

### Fixed-Length CRHF

**Gen($1^n$):** Run $(G, q, g) \leftarrow G(1^n)$. Select $h \leftarrow G$.

Output $s = (G, q, g, h)$.

**$H^s(x_1, x_2)$:** On input $(x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$, output $g^{x_1} h^{x_2} \in G$

**Theorem:** If the discrete logarithm problem is hard relative to $G$, then the construction above is a fixed-length, collision resistant hash function.

**Proof idea:** Let $\Pi = (Gen, H)$ as described above. Suppose there exists a p.p.t. adversary $A$ such that Hash-Coll$_{A, \Pi}(n) = \epsilon(n)$. We'll show $A_r$ that solves the discrete logarithm problem with the same probability.

$A_r$ receives challenge $(G, q, g, h)$ and has to find $x$ such that $g^x = h$.

$A_r$ sends $s = (G, q, g, h)$ to $A$, who returns $x = (x_1, x_2)$ and $\hat{x} = (\hat{x}_1, \hat{x}_2)$.

If $h = 1$, return $x = 0$.

Otherwise, return $[(x_1 - \hat{x}_1)(\hat{x}_2 - x_2)^{-1} \mod q]$.

**Analysis:**

\[
H^s(x_1, x_2) = H^s(\hat{x}_1, \hat{x}_2) \\
\Rightarrow g^{x_1} h^{x_2} = g^{\hat{x}_1} h^{\hat{x}_2} \\
\Rightarrow g^{(x_1 - \hat{x}_1)} = h^{(\hat{x}_2 - x_2)}
\]
CRHF from Dlog

Let $\mathcal{G}$ be a group generation algorithm that outputs a prime order group.

**Fixed-Length CRHF**

$\text{Gen}(1^n)$: Run $(G, q, g) \leftarrow \mathcal{G}(1^n)$. Select $h \leftarrow G$.

Output $s = (G, q, g, h)$.

$H^s(x_1, x_2)$: on input $(x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$, output $g^{x_1} h^{x_2} \in \mathcal{G}$

**Theorem:** If the discrete logarithm problem is hard relative to $\mathcal{G}$, then the construction above is a fixed-length, collision resistant hash function.

**Proof idea:** Let $\Pi = (\text{Gen}, H)$ as described above. Suppose there exists a p.p.t. adversary $\mathcal{A}$ such that $\text{Hash-Coll}_{\mathcal{A}, \Pi}(n) = \epsilon(n)$. We’ll show $\mathcal{A}_r$ that solves the discrete logarithm problem with the same probability.

$\mathcal{A}_r$ receives challenge $(G, q, g, h)$ and has to find $x$ such that $g^x = h$.

$\mathcal{A}_r$ sends $s = (G, q, g, h)$ to $\mathcal{A}$, who returns $x = (x_1, x_2)$ and $\hat{x} = (\hat{x}_1, \hat{x}_2)$.

If $h = 1$, return $x = 0$

Otherwise, return $[(x_1 - \hat{x}_1)(\hat{x}_2 - x_2)^{-1} \mod q]$.

**Analysis:**

$$H^s(x_1, x_2) = H^s(\hat{x}_1, \hat{x}_2)$$

$$\Rightarrow g^{x_1} h^{x_2} = g^{\hat{x}_1} h^{\hat{x}_2}$$

$$\Rightarrow g^{(x_1 - \hat{x}_1)} = h^{(\hat{x}_2 - x_2)}$$

$$\Rightarrow g^{(x_1 - \hat{x}_1)(\hat{x}_2 - x_2)^{-1}} = h^{(\hat{x}_2 - x_2)(\hat{x}_2 - x_2)^{-1}} = h^1 = h$$