We can use SIS to directly obtain a collision-resistant hash function (CRHF).

Definition. A keyed hash family $H: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is collision-resistant if the following properties hold:

- Compressing: $|\mathcal{Y}| < |\mathcal{X}|$
- Collision-Resistant: For all efficient adversaries $\mathcal{A}$:

$$\Pr[k \xleftarrow{R} \mathcal{K} \,;\, (x, x') \leftarrow \mathcal{A}(1^\lambda, k) : H(k,x) = H(k,x') \text{ and } x \neq x'] = \text{negl}(\lambda).$$

We can directly appeal to SIS to obtain a CRHF: $H: \mathbb{Z}_q^{n \times m} \times \{0,1\}^m \longrightarrow \mathbb{Z}_q^n$ where we set $m > \lceil n \log q \rceil$.

In this case, domain has size $2^m > 2^{n \log q} = q^n$, which is the size of the output space. Collision resistance follows assuming $\mathrm{SIS}_{n,m,q,\beta}$ for any $\beta \geq \sqrt{\lceil n \log q \rceil}$

The SIS hash function supports efficient local updates:

Suppose you have a public hash $h = H(x)$ of a bit-string $x \in \{0,1\}^m$. Later, you want to update $x \mapsto x'$ where x and x' only differ on a few indices (e.g., updating an entry in an address book). For instance, suppose x and x' differ only on the first bit (e.g., $x_1 = 0$ and $x_1' = 1$). Then observe the following

$$h = H(k,x) = A \cdot x$$

$$= \begin{pmatrix} | & | & & | \\ a_1 & a_2 & \cdots & a_m \\ | & | & & | \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \sum_{i \in [m]} x_i a_i = \sum_{i=2}^{m} x_i a_i \qquad \text{since } x_1 = 0$$

$$h' = H(k, x') = A \cdot x'$$

$$= \sum_{i \in [m]} x_i' a_i = x_1' a_1 + \sum_{i=2}^{m} x_i' a_i = a_1 + \sum_{i=2}^{m} x_i' a_i = a_1 + h \qquad \text{since } x_i' = x_i \text{ for all } i \geq 2$$

Thus, we can easily update h to h' by just adding to it the first column of A without (re)computing the full hash function.

The SIS hash function is universal — this will be a very useful property (in conjunction with the leftover hash lemma)

Definition. Let $H: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed hash function. We say H is ε-universal if for all $x_0, x_1 \in \mathcal{X}$ where $x_0 \neq x_1$, $\Pr[k \xleftarrow{R} \mathcal{K} : H(k, x_0) = H(k, x_1)] \leq \varepsilon$. When $\varepsilon = 1/|\mathcal{Y}|$, we say it is universal.

Lemma. The SIS hash function $H: \mathbb{Z}_q^{n \times m} \times \{0,1\}^m \longrightarrow \mathbb{Z}_q^n$ is universal.

Proof. Take any $x_0, x_1 \in \{0,1\}^m$ with $x_0 \neq x_1$. If $H(A, x_0) = H(A, x_1)$, then $A(x_0 - x_1) = 0$. Let $a_1, \dots, a_m \in \mathbb{Z}_q^n$ be columns of A. Then,

$$A(x_0 - x_1) = \sum_{i \in [m]} a_i (x_{0,i} - x_{1,i})$$

Since there exists some $j \in [m]$ where $x_{0,j} \neq x_{1,j}$, the above relation holds only if

$$a_j = \underbrace{\overset{\in \{-1,1\}}{\boxed{(x_{1,j} - x_{0,j})}} \sum_{i \neq j} a_i (x_{0,i} - x_{1,i})}_{\text{independent of } a_j}$$

Thus, $\Pr[A \xleftarrow{R} \mathbb{Z}_q^{n \times m} : A(x_0 - x_1) = 0]$

$$= \Pr[a_1, \dots, a_m \xleftarrow{R} \mathbb{Z}_q^n : a_j = (x_{1,i} - x_{0,i}) \sum_{i \neq j} a_i (x_{0,i} - x_{1,i})]$$

$$= \frac{1}{q^n}$$

Note: When q is prime, this argument also extends to any domain that is subset of $\mathbb{Z}_q^n$. Namely $H: \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m \longrightarrow \mathbb{Z}_q^n$ is universal.

__Definition__. Let X be a random variable taking on values in a finite set S. We define the guessing probability of X to be

$$\max_{s\in S} \Pr[X=s]$$

We define the min-entropy of X to be

$$H_\infty(X) = -\log \max_{s\in S} \Pr[X=s]$$

__Intuitively__: if a random variable has k bits of min-entropy, then its most likely outcome occurs with probability at most 2^{-k} (i.e., there exists at least 2^k possible values for X)

__Definition__. Let D_0, D_1 be distributions with a common support S. Then, the statistical distance between D_1 and D_2 is defined to be

$$\Delta(D_0, D_1) = \frac{1}{2}\sum_{s\in S} \left|\Pr[t \leftarrow D_0 : t = s] - \Pr[t \leftarrow D_1 : t = s]\right|$$

If D_0 and D_1 are ε-close, then __no__ adversary can distinguish with advantage better than ε

- ↳ When ε is negligible, we say the two distributions are __statistically indistinguishable__ — denoted $D_0 \overset{s}{\approx} D_1$
- ↳ Contrast with computational indistinguishability which says no __efficient__ adversary can distinguish — denoted $D_0 \overset{c}{\approx} D_1$

__Theorem (Leftover Hash Lemma)__. Let $H : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be an ε-universal hash function. Suppose $x \in \mathcal{X}$ is a random variable with k bits of min-entropy. Then, define the following two distributions:

$$D_0 : k \xleftarrow{R} \mathcal{K},\ y \leftarrow H(k,x);\ \text{output } (k,y)$$
$$D_1 : k \xleftarrow{R} \mathcal{K},\ y \xleftarrow{R} \mathcal{Y};\ \text{output } (k,y)$$

The statistical distance between D_0 and D_1 is at most

$$\Delta(D_0, D_1) \le \frac{1}{2}\sqrt{\frac{|\mathcal{Y}|}{2^k} + (|\mathcal{Y}|\varepsilon - 1)}$$

When H is universal, this becomes

$$\Delta(D_0, D_1) \le \frac{1}{2}\sqrt{|\mathcal{Y}|/2^k} \qquad (\text{since } \varepsilon = 1/|\mathcal{Y}|)$$

__Typical setting__: H is universal and $|\mathcal{Y}| = 2^{k-2\lambda}$. By LHL, $(k, H(k,x)) \overset{s}{\approx} (k, y)$ where $y \xleftarrow{R} \mathcal{Y}$.

This is an example of a "randomness extractor."

We have a source (x) with min-entropy, but not necessarily uniform.

We want to extract from it a __uniform__ random value

} LHL shows that universal hash functions can "smooth" out a non-uniform distribution

↓

Incurs loss of 2λ bits of entropy

__Common application__: extracting uniformly random cryptographic keys from non-uniform source

↳ consider $H : \mathbb{Z}_2^{n\times m} \times \{0,1\}^m \longrightarrow \mathbb{Z}_2^n$

$$H(A,x) := Ax$$

could be binary representation of a group element

suitable for use as a symmetric key

Not typically used in practice because we need distribution with at least $n+2\lambda$ bits of min-entropy ($\gtrsim 384$ bits if $n=\lambda=128$)

__Practical heuristic__: use random oracle

does not have to be uniform — just needs min-entropy

typically, we just take n to be the security parameter and set $m = \Theta(n \log q)$

__In lattices__: If $A \xleftarrow{R} \mathbb{Z}_q^{n\times m}$ and $v \xleftarrow{R} \{0,1\}^m$, then $Av \in \mathbb{Z}_q^n$ is uniform when $m > n\log q + 2\lambda$

By a hybrid argument, if we sample $R \xleftarrow{R} \{0,1\}^{m\times m}$, then AR is __statistically__ close to uniform over $\mathbb{Z}_q^{n\times m}$

We will see this used in many constructions

Commitments from SIS (recall: commitment is a "sealed envelope")

- Setup $(1^\lambda) \to \text{crs}$: Samples a common reference string
- Commit $(\text{crs}, \mu; r) \to \sigma$: Commits to a message μ with randomness r

} Here, opening can simply be the pair (m, r)
Verifier checks that $\sigma = \text{Commit}(\text{crs}, m; r)$

Useful building block for zero-knowledge proofs and other cryptographic protocols

- Setup (1^λ) : Let n, q be lattice parameters, and $m = \Theta(n \log q)$
  Sample $A_1, A_2 \xleftarrow{R} \mathbb{Z}_q^{n \times m}$. Output $\text{crs} = (A_1, A_2)$
- Commit $(\text{crs}, \mu; r)$: Output $\sigma = A_1 m + A_2 r$ where $\text{crs} = (A_1, A_2)$
  $= [A_1 \mid A_2] \begin{bmatrix} m \\ r \end{bmatrix} \in \mathbb{Z}_q^n$

$\mu, r \in \{0,1\}^m$

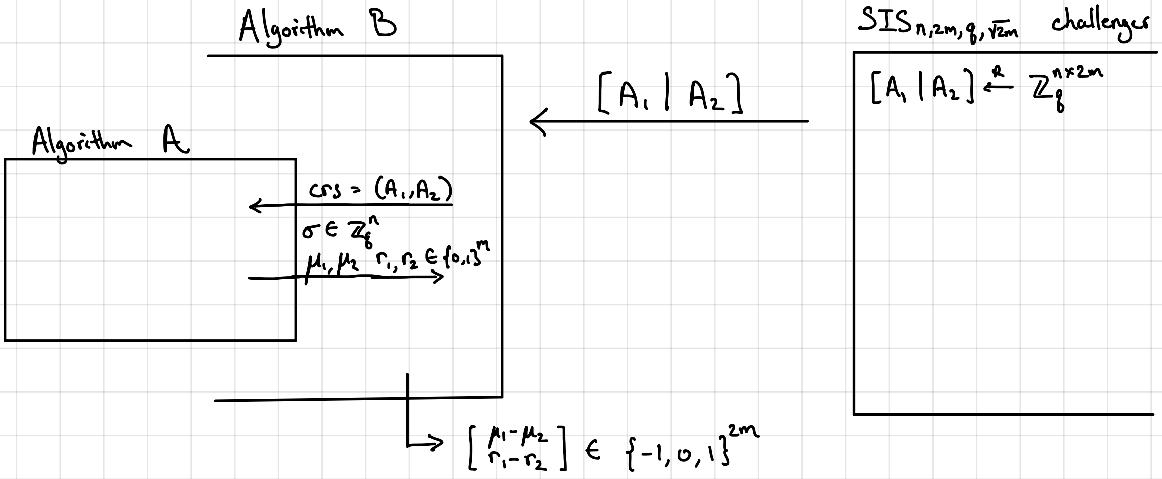
compressing when $m > n \log q$

Theorem (Statistically Hiding). If $m > 3n \log q$, then scheme is statistically hiding.

Proof. By the LHL, for $r \xleftarrow{R} \{0,1\}^m$, $A_2 r \overset{s}{\approx} \text{Uniform}(\mathbb{Z}_q^n)$. Thus, $A_2 r$ acts as a one-time pad for $A_1 m$.

Theorem (Computational Hiding). Under $\text{SIS}_{n, 2m, q, \sqrt{2m}}$, the commitment scheme is computationally binding.

Proof. Suppose $\mathcal{A}$ can break the binding property. We use $\mathcal{A}$ to construct SIS adversary $\mathcal{B}$:



If $\mathcal{A}$ is successful, then $\mu_1 \neq \mu_2$ and $[A_1 \mid A_2] \begin{bmatrix} \mu_1 \\ r_1 \end{bmatrix} = \sigma = [A_1 \mid A_2] \begin{bmatrix} \mu_2 \\ r_2 \end{bmatrix}$, which means $[A_1 \mid A_2] \begin{bmatrix} \mu_1 - \mu_2 \\ r_1 - r_2 \end{bmatrix} = 0$.
Since $\mu_1 \neq \mu_2$ this is a non-zero SIS solution with norm at most $\sqrt{2m}$.

Compare this with Pedersen commitments from discrete log:

Setup (1^λ) : Take a prime-order group $\mathbb{G} \leftarrow \text{GroupGen}(1^\lambda)$. Let p be the order of $\mathbb{G}$.
Sample $g, h \xleftarrow{R} \mathbb{G}$. Output $\text{crs} = (g, h)$

Commit $(\text{crs}, \mu; r)$: Output $g^\mu h^r$.

$\mu, r \in \mathbb{Z}_p$

discrete log		SIS
$g, h \xleftarrow{R} \mathbb{G}$		$A_1, A_2 \xleftarrow{R} \mathbb{Z}_q^{n \times m}$
g^μ	$\longrightarrow$	$A_1 \mu$
h^r		$A_2 r$

We will see many similar parallels between discrete log based systems and lattice-based systems