We will now show how to construct digital signatures from SIS in the random oracle model.

We first introduce the inhomogeneous SIS (ISIS) problem.

Inhomogeneous SIS: The inhomogeneous SIS problem is defined with respect to lattice parameters n, m, q and a norm bound β. The $\text{ISIS}_{n,m,q,\beta}$ problem says that for $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $u \xleftarrow{R} \mathbb{Z}_q^n$, no efficient adversary can find a non-zero vector $x \in \mathbb{Z}^m$ where

$$Ax = u \in \mathbb{Z}_q^n \quad \text{and} \quad \|x\| \le \beta$$

Corresponds to finding a short vector in the lattice coset $\mathcal{L}_u^\perp(A) := c + \mathcal{L}^\perp(A)$ where $c \in \mathbb{Z}^m$ is any solution where $Ac = u$ and $\mathcal{L}^\perp(A) = \{x \in \mathbb{Z}^m : Ax = 0 \pmod q\}$

For many choices of parameters, hardness of SIS $\Rightarrow$ hardness of inhomogeneous SIS (HW exercise)

For convenience, from this point forward, we will use the ℓ_∞-norm for vectors. Recall that $\|v\|_\infty \le \|v\|_2 \le \sqrt{n}\,\|v\|_\infty$

↳ if vector is short in ℓ_∞ norm, it is also short in ℓ_2-norm

The SIS and ISIS problems can be leveraged to construct lattice trapdoors. We define the syntax here:

- $\text{TrapGen}(n, m, q, \beta) \to (A, td_A)$: On input the lattice parameters n, m, q, the trapdoor-generation algorithm outputs a matrix $A \in \mathbb{Z}_q^{n \times m}$ and a trapdoor td_A
- $f_A(x) \to y$: On input $x \in \mathbb{Z}_q^m$, computes $y = Ax \in \mathbb{Z}_q^n$
- $f_A^{-1}(td_A, y) \to x$: On input the trapdoor td_A and an element $y \in \mathbb{Z}_q^n$, the inversion algorithm outputs a value $\|x\| \le \beta$

Moreover, for a suitable choice of n, m, q, β, these algorithms satisfy the following properties:

- For all $y \in \mathbb{Z}_q^n$, $f_A^{-1}(td_A, y)$ outputs $x \in \mathbb{Z}_q^n$ such that $\|x\| \le \beta$ and $Ax = y$
- The matrix A output by TrapGen is statistically close to uniform over $\mathbb{Z}_q^{n \times m}$

Lattice trapdoors have received significant amount of study and we will not have time to study it extensively. Here, we will describe the high-level idea behind a very useful and versatile trapdoor known as a "gadget" trapdoor

First, we define the "gadget" matrix (there are actually many possible gadget matrices — here, we use a common one sometimes called the "powers-of-two" matrix):

$$G = \begin{pmatrix} 1 \; 2 \; 4 \; 8 \cdots 2^{\lfloor \log q \rfloor} & & & \\ & 1 \; 2 \; 4 \cdots 2^{\lfloor \log q \rfloor} & & \\ & & \ddots & \\ & & & 1 \; 2 \; 4 \cdots 2^{\lfloor \log q \rfloor} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 2 & 4 & \cdots & 2^{\lfloor \log q \rfloor} \end{pmatrix}}_{g^T} \otimes I_n = g^T \otimes I_n$$

Each row of G consists of the powers of two (up to $2^{\lfloor \log q \rfloor}$). Thus, $G \in \mathbb{Z}_q^{n \times n \lfloor \log q \rfloor}$. Oftentimes, we will just write $G \in \mathbb{Z}_q^{n \times m}$ where $m > n\lfloor \log q \rfloor$. Note that we can always pad G with all-zero columns to obtain the desired dimension.

Observation: SIS is easy with respect to G:

$$G \cdot \begin{pmatrix} 2 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = 0 \in \mathbb{Z}_q^n \quad \Rightarrow \text{norm of this vector is } 2$$

Inhomogenous SIS is also easy with respect to G: take any target vector $y \in \mathbb{Z}_q^n$. Let $y_{i,\lfloor \log q \rfloor}, \ldots, y_{i,1}$ be the binary decomposition of y_i (the i^{th} component of y). Then,

$$G \cdot \begin{pmatrix} y_{1,1} \\ y_{1,2} \\ \vdots \\ y_{1,\lfloor \log q \rfloor} \\ y_{2,1} \\ \vdots \\ y_{2,\lfloor \log q \rfloor} \\ \vdots \\ y_{n,1} \\ \vdots \\ y_{n,\lfloor \log q \rfloor} \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{\lfloor \log q \rfloor} 2^j y_{1,j} \\ \vdots \\ \sum_{j=1}^{\lfloor \log q \rfloor} 2^j y_{n,j} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = y$$

↑ Observe that this is a 0/1 vector (binary valued vector), so the ℓ_∞-norm is exactly 1

We will denote this "bit-decomposition" operation by the function $G^{-1} : \mathbb{Z}_q^n \to \{0,1\}^m$

↑ important: G^{-1} is <u>not</u> a matrix (even though G is)!

Then, for all $y \in \mathbb{Z}_q^n$, $G \cdot G^{-1}(y) = y$ and $\|G^{-1}(y)\| = 1$. Thus, both SIS and inhomogeneous SIS are easy with respect to the matrix G.

We now have a matrix with a "public" trapdoor. To construct a <u>secret</u> trapdoor function (useful for cryptographic applications), we will "hide" the gadget matrix in the matrix A, and the trapdoor will be a "short" matrix (i.e., matrix with small entries) that recovers the gadget.

More precisely, a gadget trapdoor for a matrix $A \in \mathbb{Z}_q^{n \times k}$ is a short matrix $R \in \mathbb{Z}_q^{k \times m}$ such that

$$A \cdot R = G \in \mathbb{Z}_q^{n \times m}$$

We say that R is "short" if all values are small. [We will write $\|R\|$ to refer to the largest value in R].

Suppose we know $R \in \mathbb{Z}_q^{m \times m}$ such that $AR = G$. We can then define the inversion algorithm as follows:

- $f_A^{-1}(td_A = R, y \in \mathbb{Z}_q^n)$: Output $x = R \cdot G^{-1}(y)$.

We check two properties.

1. $Ax = AR \cdot G^{-1}(y) = G \cdot G^{-1}(y) = y$ so x is indeed a valid pre-image
2. $\|x\| = \|R \cdot G^{-1}(y)\| \leq m \cdot \|R\| \|G^{-1}(y)\| = m \cdot \|R\|$
   Thus, if $\|R\|$ is small, then $\|x\|$ is also small (think of β as a large polynomial in n).
   (Recall we are using ℓ_∞-norm now)

<u>Important note:</u> When using trapdoor functions in a setting where the adversary can see trapdoor evaluations, we actually need to <u>randomize</u> the computation of f_A^{-1}. Otherwise, we <u>leak</u> the trapdoor. (We will revisit this later.)

<u>Remaining question:</u> How do we generate A together with a trapdoor (and so that A is statistically close to uniform)?

Many techniques to do so; we will look at one approach using the LHL:

Sample $\bar{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$ and $\bar{R} \xleftarrow{R} \{0,1\}^{m \times m}$.

Set $A = [\bar{A} \mid \bar{A}\bar{R} + G] \in \mathbb{Z}_q^{n \times 2m}$

Output $A \in \mathbb{Z}_q^{n \times 2m}$, $td_A = R = \begin{bmatrix} -\bar{R} \\ I \end{bmatrix} \in \mathbb{Z}_q^{2m \times m}$

First, we have by construction that $AR = -\bar{A}\bar{R} + \bar{A}\bar{R} + G = G$, and moreover $\|R\| = 1$. It suffices to argue that A is statistically close to uniform (without the trapdoor R). This boils down to showing that $\bar{A}\bar{R} + G$ is statistically close to uniform given $\bar{A}$. We appeal to the LHL:

1. From the previous lecture, the function $f_A(x) = Ax$ is pairwise independent
2. Thus, by the LHL, if $m \geq 3n \log q$, then Ar is statistically close to uniform in $\mathbb{Z}_q^n$ when $r \xleftarrow{R} \{0,1\}^m$.
3. Claim now follows by a hybrid argument (applied to each column of R)

Thus, given $\bar{A}$, the matrix $\bar{A}\bar{R}$ is still statistically close to uniform. Corresponding, A is statistically close to uniform.

Digital signatures from lattice trapdoors: We can use lattice trapdoors to obtain a digital signature scheme in the random oracle model (this is essentially an analog of RSA signatures):

- $\mathrm{KeyGen}(1^\lambda)$: $(A, td_A) \leftarrow \mathrm{TrapGen}(n, m, q, \beta)$ [lattice parameters n, m, q, β are based on security parameter λ]
  Output $vk = A$ and $sk = td_A$
- $\mathrm{Sign}(sk, m)$: Output $\sigma \leftarrow f_A^{-1}(td_A, H(m))$. Here, $H: \{0,1\}^* \to \mathbb{Z}_q^n$ is modeled as a random oracle.
- $\mathrm{Verify}(vk, m, \sigma)$: Check that $\|\sigma\| \leq \beta$ and that $f_A(\sigma) = H(m)$.

Consider instantiation with gadget trapdoors:
- verification key: $A \in \mathbb{Z}_q^{n \times m}$
  signing key: $R \in \{0,1\}^{m \times m}$ such that $AR = G$
- signature on m: $y \leftarrow H(m) \in \mathbb{Z}_q^n$
  output $\sigma = v = R \cdot G^{-1}(y)$
- verification: check that
  $$A \cdot v = ARG^{-1}(y) = G \cdot G^{-1}(y) = y$$
  and v is short

Rationale for security:
- To forge a signature on m, adversary has to find v such that $Av = H(m)$
- Matrix A is statistically close to uniform and v is uniform, so this corresponds to solving the ISIS problem

Problem: Signing queries leak information about R. Adversary can compute $H(m) = y$ and $G^{-1}(y)$, so signing becomes a linear function!

Early approach of Goldreich-Goldwasser-Halevi was insecure — explicit key-recovery attack by Nguyen, Regev

In the context of the security proof, simulator needs a way to answer signing queries (without a trapdoor for A).

Requirement: Randomize the signing algorithm to hide trapdoor R

Definition. A function $f: \mathcal{X} \to \mathcal{Y}$ is a preimage-sampleable trapdoor function if there exists some efficiently-sampleable distribution D over $\mathcal{X}$ and a trapdoor inversion algorithm SamplePre with the following properties:

$$\left\{ \begin{array}{l} x \leftarrow D \\ y \leftarrow f(x) \end{array} : (x, y) \right\} \overset{s}{\approx} \left\{ \begin{array}{l} y \xleftarrow{R} \mathcal{Y} \\ x \leftarrow \mathrm{SamplePre}(td, x) \end{array} \right\}$$

(trapdoor for preimage sampling)

"forward sampling" — "backward sampling" ← two ways to do the same thing
- One approach in real scheme
- One approach in security proof

Moreover, given f and $y \xleftarrow{R} \mathcal{Y}$, no efficient adversary can find x such that $f(x) = y$.

Definition requires (1) for $x \leftarrow D$, $f(x)$ is uniform over $\mathcal{Y}$
(2) for a random $y \xleftarrow{R} \mathcal{Y}$, inversion algorithm samples a preimage from D conditioned on $f(x) = y$

Observe that a trapdoor permutation is a deterministic preimage sampleable trapdoor function: SamplePre returns the unique trapdoor

If we use a preimage sampleable trapdoor function in digital signature construction, then we can argue security (similar to arguing security of RSA-FDH in random oracle model).