CS 276: Homework 6

Due Date: Saturday November 2nd, 2024 at 8:59pm via Gradescope

1 The OR of Two Hash Proof Systems

We will present a hash proof system for the language of DDH tuples and then build a hash proof system for the OR of two such proof systems.

Definition 1.1 (Hash Proof System) *A hash proof system (HPS) is a tuple of algorithms* $(\mathsf{Gen}, \mathsf{SKHash}, \mathsf{PKHash})$ *with the following syntax:*

- Gen *takes a security parameter* 1^λ *and outputs a public key* pk *and a secret key* sk*.*
- SKHash*: Takes* sk *and an instance* $x \in \mathcal{X}$ *and outputs* $y \in \mathcal{Y}$*.*
- PKHash*: Takes* pk*, an instance* $x \in \mathcal{X}$*, and a witness* w *and outputs* $y \in \mathcal{Y}$*.*

Note that $\mathcal{X}$ *is the input space, and* $\mathcal{Y}$ *is the output space.*

The HPS satisfies the following properties:

- ***Correctness:*** *If* $x \in L$ *and* w *is a valid witness for* x*, then* $\mathsf{SKHash}(\mathsf{sk}, x) = \mathsf{PKHash}(\mathsf{pk}, x, w)$*.*
- ***Smoothness:*** *For any* $x \notin L$*, the following distributions are identical:*

$$\{(\mathsf{pk}, y) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda), y \leftarrow \mathsf{SKHash}(\mathsf{sk}, x)\}$$
$$\{(\mathsf{pk}, y) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda), y \xleftarrow{\$} \mathcal{Y}\}$$

1.1 HPS for DDH tuples

We will present an HPS for the language of DDH tuples.

Let $\mathbb{G}$ be a cyclic group of order p, where p is a large prime. Let g, h be two generators of $\mathbb{G}$. Let the DDH language L be the following:

$$L = \{(g^w, h^w) \in \mathbb{G}^2 : w \in \mathbb{Z}_p\}$$

[1]

Let $\mathcal{X} = \mathbb{G}^2$, let $x = (a, b) \in \mathcal{X}$, and let $\mathcal{Y} = \mathbb{G}$. For any tuple $x = (g^w, h^w) \in L$, let w serve as the witness. Then we can construct a hash proof system for L as follows:

Definition 1.2 (HPS For The DDH Language L)

- $\mathsf{Gen}(1^\lambda)$*: Sample* $\mathsf{sk} = (r, s) \xleftarrow{\$} \mathbb{Z}_p^2$*. Let* $\mathsf{pk} = g^r \cdot h^s$*. Then output* $(\mathsf{pk}, \mathsf{sk})$*.*
- $\mathsf{SKHash}(\mathsf{sk}, x)$*: Output* $y = a^r \cdot b^s$*.*
- $\mathsf{PKHash}(\mathsf{pk}, x, w)$*: Output* $y = \mathsf{pk}^w$*.*

[1] Note that the DDH problem asks an adversary to distinguish (g, h, g^w, h^w) from (g, h, g^w, h^v), for $h \xleftarrow{\$} \mathbb{G}$ and $(w, v) \xleftarrow{\$} \mathbb{Z}_p^2$, so the ability to decide whether a given tuple belongs to L is sufficient to solve DDH.

Question 1: Prove that the HPS constructed above satisfies correctness and smoothness.

Solution The solution is based on [ABP14].

Theorem 1.3 *The HPS for L given in definition 1.2 satisfies correctness.*

Proof. To prove correctness, it suffices to show that for any $\mathsf{sk} = (r, s)$ and any $x = (a, b)$ and w for which $a = g^w$ and $b = h^w$,

$$\mathsf{SKHash}(\mathsf{sk}, x) = \mathsf{PKHash}(\mathsf{pk}, x, w)$$

That is shown as follows:

$$\begin{aligned}\mathsf{SKHash}(\mathsf{sk}, x) &= a^r \cdot b^s \\ &= (g^w)^r \cdot (h^w)^s \\ &= (g^r \cdot h^s)^w \\ &= \mathsf{pk}^w \\ &= \mathsf{PKHash}(\mathsf{pk}, x, w)\end{aligned}$$

Theorem 1.4 *The HPS for L given in definition 1.2 satisfies smoothness.*

Proof. It will help to focus on the discrete log of each group element, because then we can treat these computations as linear functions. Let $\tilde{h}, \tilde{a}, \tilde{b} \in \mathbb{Z}_p$ be defined such that $h = g^{\tilde{h}}$, $a = g^{\tilde{a}}$, and $b = g^{\tilde{b}}$.

Next,

$$\begin{aligned}\text{let } \tilde{\mathbf{x}} &= [\tilde{a}, \tilde{b}]^T \\ \mathbf{v} &= [1, \tilde{h}]^T \\ \mathbf{M} &= \begin{bmatrix}\mathbf{v} & \tilde{\mathbf{x}}\end{bmatrix} = \begin{bmatrix}1 & \tilde{a} \\ \tilde{h} & \tilde{b}\end{bmatrix} \\ \mathsf{sk} &= [r, s]^T\end{aligned}$$

$$\begin{aligned}\text{Then } \mathsf{pk} &= g^{r + s \cdot \tilde{h}} = g^{\mathsf{sk}^T \cdot \mathbf{v}} = g^{(\mathsf{sk}^T \cdot \mathbf{M})_1} \\ \mathsf{SKHash}(\mathsf{sk}, x) &= g^{r \cdot \tilde{a} + s \cdot \tilde{b}} = g^{\mathsf{sk}^T \cdot \tilde{\mathbf{x}}} = g^{(\mathsf{sk}^T \cdot \mathbf{M})_2}\end{aligned}$$

To prove smoothness, it suffices to prove that if $x \notin L$, then for a uniformly random sk, $\mathsf{sk}^T \cdot \mathbf{v}$ and $\mathsf{sk}^T \cdot \tilde{\mathbf{x}}$ are uniformly random and independent.

If $x \notin L$, then $\tilde{b} \neq \tilde{h} \cdot \tilde{a}$. Then $\mathbf{v}$ and $\tilde{\mathbf{x}}$ are not parallel, so $\mathbf{M}$ is full-rank. This implies that for a uniformly random sk, the values of $\mathsf{sk}^T \cdot \mathbf{v}$ and $\mathsf{sk}^T \cdot \tilde{\mathbf{x}}$ are uniformly random and independent. As a result, pk and $\mathsf{SKHash}(\mathsf{sk}, x)$ will be uniformly random and independent as well. ■

1.2 HPS for the OR of two languages

Now we will construct a HPS for the OR of two DDH languages, with the help of a bilinear map.

Let $\mathbb{G}_0$ and $\mathbb{G}_1$ be cyclic groups of order p, where p is a large prime. Let (g_0, h_0) be generators of $\mathbb{G}_0$, and let (g_1, h_1) be generators of $\mathbb{G}_1$. Let us define the following languages:

$$\begin{aligned}
L_0 &= \{(g_0^w, h_0^w) \in \mathbb{G}_0^2 : w \in \mathbb{Z}_p\} \\
L_1 &= \{(g_1^w, h_1^w) \in \mathbb{G}_1^2 : w \in \mathbb{Z}_p\} \\
L_\vee &= \{(a_0, b_0, a_1, b_1) \in \mathbb{G}_0^2 \times \mathbb{G}_1^2 : (a_0, b_0) \in L_0 \vee (a_1, b_1) \in L_1\}
\end{aligned}$$

Let $x = (a_0, b_0, a_1, b_1)$, and let the witness for $x \in L_\vee$ be a value $w \in \mathbb{Z}_p$ such that either (1) $a_0 = g_0^w$ and $b_0 = h_0^w$ or (2) $a_1 = g_1^w$ and $b_1 = h_1^w$.

Furthermore, let $e : \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_T$ be an efficiently computable pairing function that satisfies:

$$e(g_0^r, g_1^s) = e(g_0, g_1)^{r \cdot s}$$

for any $r, s \in \mathbb{Z}_p$.

Now, we can construct a HPS for $L_\vee$.

Definition 1.5 (HPS For $L_\vee$)

- $\mathsf{Gen}(1^\lambda)$: *Sample* $\mathsf{sk} = (r, s, t, u) \xleftarrow{\$} \mathbb{Z}_p^4$. *Compute*

  $$\mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{pk}_3) = \left(g_0^r \cdot h_0^t, g_0^s \cdot h_0^u, g_1^r \cdot h_1^s, g_1^t \cdot h_1^u\right)$$

  Finally, output $(\mathsf{pk}, \mathsf{sk})$.

- $\mathsf{SKHash}(\mathsf{sk}, x)$: *Given* $x = (a_0, b_0, a_1, b_1)$, *compute and output*

  $$y = e(a_0, a_1)^r \cdot e(a_0, b_1)^s \cdot e(b_0, a_1)^t \cdot e(b_0, b_1)^u$$

- $\mathsf{PKHash}(\mathsf{pk}, x, w)$: *If* $a_0 = g_0^w$ *and* $b_0 = h_0^w$ *(*$(a_0, b_0) \in L_0$*), then compute and output*

  $$y = e(\mathsf{pk}_0, a_1)^w \cdot e(\mathsf{pk}_1, b_1)^w$$

  If $a_1 = g_1^w$ *and* $b_1 = h_1^w$ *(*$(a_1, b_1) \in L_1$*), then compute and output*

  $$y = e(a_0, \mathsf{pk}_2)^w \cdot e(b_0, \mathsf{pk}_3)^w$$

Question 2: Prove that the HPS for $L_\vee$ satisfies correctness and smoothness.

Solution

Claim 1.6 (Correctness) *The HPS for $L_\vee$ given in def. 1.5 satisfies correctness.*

Proof. If $(a_0, b_0) \in L_0$, then $\mathsf{SKHash}(\mathsf{sk}, x) = \mathsf{PKHash}(\mathsf{pk}, x, w)$. In this case, $a_0 = g_0^w$ and $b_0 = h_0^w$.

$$\begin{aligned}\mathsf{SKHash}(\mathsf{sk}, x) &= e(a_0, a_1)^r \cdot e(a_0, b_1)^s \cdot e(b_0, a_1)^t \cdot e(b_0, b_1)^u \\ &= e(g_0, a_1)^{w\cdot r} \cdot e(g_0, b_1)^{w\cdot s} \cdot e(h_0, a_1)^{w\cdot t} \cdot e(h_0, b_1)^{w\cdot u}\end{aligned}$$

$$\begin{aligned}\mathsf{PKHash}(\mathsf{pk}, x, w) &= e(\mathsf{pk}_0, a_1)^w \cdot e(\mathsf{pk}_1, b_1)^w \\ &= e(g_0^r \cdot h_0^t, a_1)^w \cdot e(g_0^s \cdot h_0^u, b_1)^w \\ &= e(g_0, a_1)^{w\cdot r} \cdot e(g_0, b_1)^{w\cdot s} \cdot e(h_0, a_1)^{w\cdot t} \cdot e(h_0, b_1)^{w\cdot u}\end{aligned}$$

$$\mathsf{SKHash}(\mathsf{sk}, x) = \mathsf{PKHash}(\mathsf{pk}, x, w)$$

Next, if $(a_1, b_1) \in L_1$, then $\mathsf{SKHash}(\mathsf{sk}, x) = \mathsf{PKHash}(\mathsf{pk}, x, w)$. In this case, $a_1 = g_1^w$ and $b_1 = h_1^w$.

$$\begin{aligned}\mathsf{SKHash}(\mathsf{sk}, x) &= e(a_0, a_1)^r \cdot e(a_0, b_1)^s \cdot e(b_0, a_1)^t \cdot e(b_0, b_1)^u \\ &= e(a_0, g_1)^{w\cdot r} \cdot e(a_0, h_1)^{w\cdot s} \cdot e(b_0, g_1)^{w\cdot t} \cdot e(b_0, h_1)^{w\cdot u}\end{aligned}$$

$$\begin{aligned}\mathsf{PKHash}(\mathsf{pk}, x, w) &= e(a_0, \mathsf{pk}_2)^w \cdot e(b_0, \mathsf{pk}_3)^w \\ &= e(a_0, g_1^r \cdot h_1^s)^w \cdot e(b_0, g_1^t \cdot h_1^u)^w \\ &= e(a_0, g_1)^{w\cdot r} \cdot e(a_0, h_1)^{w\cdot s} \cdot e(b_0, g_1)^{w\cdot t} \cdot e(b_0, h_1)^{w\cdot u}\end{aligned}$$

$$\mathsf{SKHash}(\mathsf{sk}, x) = \mathsf{PKHash}(\mathsf{pk}, x, w)$$

Claim 1.7 (Smoothness) *The HPS for $L_\vee$ given in def. 1.5 satisfies smoothness.*

Proof.

1. It helps to focus on the discrete log of each group element because then we can treat these computations as linear functions. Let $\tilde{h}_0, \tilde{h}_1, \tilde{a}_0, \tilde{a}_1, \tilde{b}_0, \tilde{b}_1 \in \mathbb{Z}_p$ be defined such that

$$\begin{aligned} h_0 = g_0^{\tilde{h}_0}, &\quad h_1 = g_1^{\tilde{h}_1} \\ a_0 = g_0^{\tilde{a}_0}, &\quad a_1 = g_1^{\tilde{a}_1} \\ b_0 = g_0^{\tilde{b}_0}, &\quad b_1 = g_1^{\tilde{b}_1} \end{aligned}$$

Then

$$\begin{aligned}\mathsf{pk}_0 &= g_0^{r+t\cdot\tilde{h}_0}\\ \mathsf{pk}_1 &= g_0^{s+u\cdot\tilde{h}_0}\\ \mathsf{pk}_2 &= g_1^{r+s\cdot\tilde{h}_1}\\ \mathsf{pk}_3 &= g_1^{t+u\cdot\tilde{h}_1}\end{aligned}$$

$$\mathsf{SKHash}(\mathsf{sk}, x) = g_T^{\tilde{a}_0\cdot\tilde{a}_1\cdot r+\tilde{a}_0\cdot\tilde{b}_1\cdot s+\tilde{b}_0\cdot\tilde{a}_1\cdot t+\tilde{b}_0\cdot\tilde{b}_1\cdot u}$$

where $g_T = e(g_0, g_1)$.

2. Let us define some vectors and matrices to represent the discrete log of the group elements above.

$$\begin{aligned}\text{let } \mathsf{sk} &= [r, s, t, u]^T\\ \tilde{\mathsf{pk}} &= [r + t\cdot\tilde{h}_0, s + u\cdot\tilde{h}_0, r + s\cdot\tilde{h}_1, t + u\cdot\tilde{h}_1]^T\\ \tilde{\mathbf{x}} &= [\tilde{a}_0\cdot\tilde{a}_1, \tilde{a}_0\cdot\tilde{b}_1, \tilde{b}_0\cdot\tilde{a}_1, \tilde{b}_0\cdot\tilde{b}_1]^T\\ \mathbf{M} &= \begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 1 & \tilde{h}_1 & 0\\ \tilde{h}_0 & 0 & 0 & 1\\ 0 & \tilde{h}_0 & 0 & \tilde{h}_1\end{bmatrix}\end{aligned}$$

Then

$$\begin{aligned}\mathsf{SKHash}(\mathsf{sk}, x) &= g_T^{\mathsf{sk}^T\cdot\tilde{\mathbf{x}}}\\ \tilde{\mathsf{pk}}^T &= \mathsf{sk}^T\cdot\mathbf{M}\end{aligned}$$

Note that $\tilde{\mathsf{pk}}^T = \mathsf{sk}^T\cdot\mathbf{M}$ uniquely determines the value of pk, and $\mathsf{sk}^T\cdot\tilde{\mathbf{x}}$ uniquely determines the value of $\mathsf{SKHash}(\mathsf{sk}, x)$.

To prove smoothness, we just need to show that when $x \notin L_\vee$, then for a uniformly random sk, the values $\mathsf{sk}^T\cdot\mathbf{M}$ and $\mathsf{sk}^T\cdot\tilde{\mathbf{x}}$ are uniformly random and independent.

3. The following vector $\mathbf{v}$ is perpendicular to the column-span of $\mathbf{M}$.

$$\begin{aligned}\text{Let } \mathbf{v} &= [\tilde{h}_0\cdot\tilde{h}_1, -\tilde{h}_0, -\tilde{h}_1, 1]^T\\ \text{Then } \mathbf{v}^T\cdot\mathbf{M} &= [0, 0, 0, 0]\end{aligned}$$

4. $x \in L_\vee$ if and only if $\mathbf{v}^T\cdot\tilde{\mathbf{x}} = 0$.

$$\begin{aligned}\mathbf{v}^T\cdot\tilde{\mathbf{x}} &= \tilde{a}_0\cdot\tilde{a}_1\cdot\tilde{h}_0\cdot\tilde{h}_1 - \tilde{h}_0\cdot\tilde{a}_0\cdot\tilde{b}_1 - \tilde{h}_1\cdot\tilde{b}_0\cdot\tilde{a}_1 + \tilde{b}_0\cdot\tilde{b}_1\\ &= \tilde{a}_1\cdot\tilde{h}_1\cdot(\tilde{a}_0\cdot\tilde{h}_0 - \tilde{b}_0) + \tilde{b}_1\cdot(\tilde{b}_0 - \tilde{a}_0\cdot\tilde{h}_0)\\ &= (\tilde{b}_0 - \tilde{a}_0\cdot\tilde{h}_0)\cdot(\tilde{b}_1 - \tilde{a}_1\cdot\tilde{h}_1)\end{aligned}$$

Next,

$$\begin{aligned} x \in L_\vee &\iff (a_0, b_0) \in L_0 \vee (a_1, b_1) \in L_1 \\ &\iff \tilde{b}_0 = \tilde{a}_0 \cdot \tilde{h}_0 \vee \tilde{b}_1 = \tilde{a}_1 \cdot \tilde{h}_1 \\ &\iff (\tilde{b}_0 - \tilde{a}_0 \cdot \tilde{h}_0) \cdot (\tilde{b}_1 - \tilde{a}_1 \cdot \tilde{h}_1) = 0 \\ &\iff \mathbf{v}^T \cdot \tilde{\mathbf{x}} = 0 \end{aligned}$$

5. If $x \notin L_\vee$, then $\tilde{\mathbf{x}}$ is not in the column-span of $\mathbf{M}$ because $\mathbf{v}^T \cdot \tilde{\mathbf{x}} \neq 0$. Then for a uniformly random $\mathsf{sk} \xleftarrow{\$} \mathbb{Z}_p^4$, the values of $\mathsf{sk}^T \cdot \mathbf{M}$ and $\mathsf{sk}^T \cdot \tilde{\mathbf{x}}$ are uniformly random and independent.

■

2 Identity-Based Encryption from LWE

We will construct identity-based encryption (IBE) and prove security from the decisional LWE assumption.

Parameters and Notation: Let n be the security parameter. Let $q \in [\frac{n^4}{2}, n^4]$ be a large prime modulus. Let $m = 20n \log n$, $\alpha = \frac{1}{m^4 \cdot \log^2 m}$, $L = m^{2.5}$, $s = m^{2.5} \cdot \log m$.

Let χ be a Gaussian-weighted probability distribution over $\mathbb{Z}_q$ with mean 0 and standard deviation $\frac{q \cdot \alpha}{\sqrt{2\pi}}$.

Let $H : \{0,1\}^* \to \mathbb{Z}_q^n$ be a random oracle.

Definition 2.1 (Decisional LWE Assumption) *For any $m' \geq m$, the following two distributions are computationally indistinguishable:*

$$\{(\mathbf{A}, \mathbf{u}) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m'}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \xleftarrow{\$} \chi^{m'}, \mathbf{u} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e}\}$$
$$\{(\mathbf{A}, \mathbf{u}) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m'}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^{m'}\}$$

Helper Functions: Our construction will use the following helper functions:

- $\mathsf{TrapdoorSample}(1^n) \to \mathbf{A}, \mathbf{T}$: Samples two matrices $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{T} \leftarrow \mathbb{Z}_q^{m \times m}$ such that $\mathbf{A}$ is statistically close to uniformly random, $\mathsf{ker}(\mathbf{A}) = \mathsf{column\text{-}span}(\mathbf{T})$, and every column of $\mathbf{T}$ is short: $\|\mathbf{T} \cdot \hat{e}_i\| \leq L$ for all $i \in [m]$. In other words, $\mathbf{T}$ is a short basis of $\mathsf{ker}(\mathbf{A})$.

- $\mathsf{PreimageSample}(\mathbf{A}, \mathbf{T}, \mathbf{v})$: Samples $\mathbf{e}$ such that $\mathbf{A} \cdot \mathbf{e} = \mathbf{v} \mod q$ from a distribution proportional to a discrete Gaussian with mean $\mathbf{0}$ and standard deviation s. In other words, $\mathbf{e}$ is a short vector in the preimage of $\mathbf{v}$.

The following lemma will be useful.

Lemma 2.2 *For $\mathbf{v} \in \mathbb{Z}_q^m$ sampled from a discrete Gaussian distribution with mean $\mathbf{0}$ and a sufficiently large standard deviation s, $\Pr[\|\mathbf{v}\| > s\sqrt{m}] \leq \mathsf{negl}(m)$.*

Construction:

- $\mathsf{Setup}(1^n)$: Sample
$$\mathbf{A}, \mathbf{T} \leftarrow \mathsf{TrapdoorSample}(1^n)$$
Finally output $\mathsf{mpk} = \mathbf{A}$ and $\mathsf{msk} = \mathbf{T}$.

- $\mathsf{Gen}(\mathsf{msk}, ID)$: Compute $\mathbf{v} = H(ID)$. Then sample a short vector
$$\mathbf{e} \leftarrow \mathsf{PreimageSample}(\mathbf{A}, \mathbf{T}, \mathbf{v})$$
Note that $\mathbf{A} \cdot \mathbf{e} = \mathbf{v} \mod q$. Finally, output $\mathsf{sk}_{ID} = \mathbf{e}$.

- $\mathsf{Enc}(\mathsf{mpk}, ID, b)$: Let $b \in \{0,1\}$. Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x} \leftarrow \chi^m$ and $x \leftarrow \chi$. Then compute $\mathbf{v} = H(ID)$, and

$$\mathbf{p} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{x}$$
$$c = \mathbf{v}^T \cdot \mathbf{s} + x + b \cdot \lfloor q/2 \rfloor$$

  Output $\mathsf{ct} = (\mathbf{p}, c)$.

- $\mathsf{Dec}(\mathsf{sk}_{ID}, \mathsf{ct})$: Parse $\mathsf{sk}_{ID} = \mathbf{e}$ and $\mathsf{ct} = (\mathbf{p}, c)$. Compute

$$\mu = c - \mathbf{e}^T \cdot \mathbf{p}$$

  If $|\mu - q/2| \le q/4$, then output $b' = 1$. Otherwise, output $b' = 0$.

Question: Prove that the IBE construction given above is correct (except with negligible probability) and secure assuming decisional LWE (def. 2.1).

Solution This problem is based on the IBE construction from [GPV07].

Theorem 2.3 *The IBE scheme is correct except with negligible probability.*

Proof. For any $b \in \{0,1\}$, let us compute $\mathsf{Dec}(\mathsf{sk}_{ID}, \mathsf{Enc}(\mathsf{mpk}, ID, b))$.

$$\begin{aligned}
\mu &= c - \mathbf{e}^T \cdot \mathbf{p} \\
&= \mathbf{v}^T \cdot \mathbf{s} + x + b \cdot \lfloor q/2 \rfloor - \mathbf{e}^T \cdot \mathbf{A}^T \cdot \mathbf{s} - \mathbf{e}^T \cdot \mathbf{x} \\
&= \mathbf{v}^T \cdot \mathbf{s} + x + b \cdot \lfloor q/2 \rfloor - \mathbf{v}^T \cdot \mathbf{s} - \mathbf{e}^T \cdot \mathbf{x} \\
\mu - b \cdot \lfloor q/2 \rfloor &= x - \mathbf{e}^T \cdot \mathbf{x}
\end{aligned}$$

With overwhelming probability, $\mathbf{e}^T \cdot \mathbf{x} \le q/10$ and $x \le q/10$ (lemma 2.4), in which case:

$$\left|\mu - b \cdot \lfloor q/2 \rfloor\right| \le \frac{q}{10} + \frac{q}{10} = \frac{q}{5}$$

Then when $b = 1$,

$$|\mu - q/2| = |\mu - b \cdot q/2| \le q/4$$

When $b = 0$, $\mu \le q/5$, so

$$|\mu - q/2| = q/2 - \mu \ge q/2 - q/5 = .3q > q/4$$

So $\mathsf{Dec}(\mathsf{sk}_{ID}, \mathsf{Enc}(\mathsf{mpk}, ID, b))$ will output b.

Lemma 2.4 *For sufficiently large n, with overwhelming probability, $\mathbf{e}^T \cdot \mathbf{x} \le q/10$ and $x \le q/10$.*

Proof. First, $\mathbf{x} \leftarrow \chi^m$, where χ^m is a discrete Gaussian with standard deviation

$$\begin{aligned} s' &= \frac{\sqrt{m} \cdot q \cdot \alpha}{\sqrt{2\pi}} \\ &= \frac{\sqrt{m} \cdot q}{\sqrt{2\pi} \cdot m^4 \cdot \log^2 m} = \frac{q}{\sqrt{2\pi} \cdot m^{3.5} \cdot \log^2 m} \end{aligned}$$

By lemma 2.2, with overwhelming probability,

$$\|\mathbf{x}\| \leq s'\sqrt{m} = \frac{q}{\sqrt{2\pi} \cdot m^3 \cdot \log^2 m}$$

Next, $\mathbf{e}$ is sampled from a discrete Gaussian with standard deviation $s = m^{2.5} \cdot \log m$. Then by lemma 2.2, with overwhelming probability,

$$\|\mathbf{e}\| \leq s\sqrt{m} = m^3 \cdot \log m$$

Then

$$\begin{aligned} \mathbf{e}^T \cdot \mathbf{x} &\leq \|\mathbf{e}\| \cdot \|\mathbf{x}\| \\ &\leq m^3 \cdot \log m \cdot \frac{q}{\sqrt{2\pi} \cdot m^3 \cdot \log^2 m} \\ &= \frac{q}{\sqrt{2\pi} \cdot \log m} \end{aligned}$$

For sufficiently large n and m, $\frac{q}{\sqrt{2\pi}\cdot\log m} < \frac{q}{10}$.

Theorem 2.5 *The IBE scheme is CPA-secure.*

Proof.

The adversary's view: The adversary receives the public key $\mathsf{mpk} = \mathbf{A}$ as well as $\mathbf{v} = H(ID^*)$ for the ID^* under which the challenge ciphertext is computed. Then for a random message $b \xleftarrow{\$} \{0, 1\}$, the adversary receives $\mathsf{Enc}(\mathsf{mpk}, ID^*, b)$, which comprises:

$$\begin{aligned} \mathbf{p} &= \mathbf{A}^T \cdot \mathbf{s} + \mathbf{x} \\ c &= \mathbf{v}^T \cdot \mathbf{s} + x + b \cdot \lfloor q/2 \rfloor \end{aligned}$$

We can express these values as follows.

$$\begin{aligned} \text{Let } \mathbf{A}' &= [A||\mathbf{v}] \\ \mathbf{u}' &= (\mathbf{p}||(\mathbf{v}^T \cdot \mathbf{s} + x)) \quad \text{expressed as a column vector} \\ \mathbf{x}' &= (\mathbf{x}||x) \quad \text{expressed as a column vector} \\ \mathbf{b} &= (0^m||(b \cdot \lfloor q/2 \rfloor)) \quad \text{expressed as a column vector} \end{aligned}$$

Then to phrase things differently, the adversary receives $(\mathbf{A}', \mathbf{u}' + \mathbf{b})$, where $\mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n\times(m+1)}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x}' \xleftarrow{\$} \chi^{m+1}$, and

$$\mathbf{u}' = \mathbf{A}'^T \cdot \mathbf{s} + \mathbf{x}'$$

The decisional LWE assumption (def. 2.1) says that this distribution over $(\mathbf{A}', \mathbf{u}' + \mathbf{b})$ is computationally indistinguishable from

$$\{(\mathbf{A}', \mathbf{u}' + \mathbf{b}) : \mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n\times(m+1)}, \mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^{m+1}\}$$

Finally, the adversary can also query on any[2] ID to learn vectors $(\mathbf{v}_{ID}, \mathbf{e}_{ID})$ for which $\mathbf{v} = H(ID)$ and $\mathbf{v} = \mathbf{A} \cdot \mathbf{e} \mod q$. However, these queries can be simulated by sampling a random $\mathbf{e}$ for each ID, then computing $\mathbf{v} = \mathbf{A} \cdot \mathbf{e} \mod q$, and programming the random oracle so that $H(ID) = \mathbf{v}$.

Reduction: Given an adversary $\mathcal{A}_{IBE}$ that breaks the CPA security of the IBE scheme, we can construct and adversary $\mathcal{A}_{LWE}$ that breaks the decisional LWE assumption.

Construction of $\mathcal{A}_{LWE}$:

1. $\mathcal{A}_{LWE}$ receives $(\mathbf{A}', \mathbf{u}')$, where either
   (a) $\mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n\times(m+1)}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x}' \xleftarrow{\$} \chi^{m+1}$, and $\mathbf{u}' = \mathbf{A}'^T \cdot \mathbf{s} + \mathbf{x}'$
   (b) Or $\mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n\times(m+1)}, \mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^{m+1}$
2. $\mathcal{A}_{LWE}$ sets mpk to be the first m columns of $\mathbf{A}'$ and $\mathbf{v}^*$ to be the final column of $\mathbf{A}'$. $\mathcal{A}_{LWE}$ samples $b \xleftarrow{\$} \{0,1\}$ and sets $\mathbf{b} = (0^m || (b \cdot \lfloor q/2 \rfloor))$. Then $\mathcal{A}_{LWE}$ computes $\mathsf{ct} = \mathbf{u}' + \mathbf{b}$.
3. $\mathcal{A}_{LWE}$ runs $\mathcal{A}_{IBE}$ internally and simulates the CPA security game. $\mathcal{A}_{IBE}$ receives mpk. Then when $\mathcal{A}_{IBE}$ chooses the identity of the encryptor ID^* for the challenge ciphertext, $\mathcal{A}_{IBE}$ receives the challenge ciphertext ct.
4. Whenever $\mathcal{A}_{IBE}$ asks for sk_{ID} or $H(ID)$ for a given ID, $\mathcal{A}_{LWE}$ handles these queries as follows:
   (a) If $\mathcal{A}_{IBE}$ has previously asked for sk_{ID} or $H(ID)$ for this particular ID, then $\mathcal{A}_{LWE}$ looks up the value of sk_{ID} or $H(ID)$ that was computed previously and returns it to $\mathcal{A}_{IBE}$.
   (b) Else if the queried ID is not ID^*, the challenge ID, then $\mathcal{A}_{LWE}$ samples $\mathbf{e} \in \mathbb{Z}_q^m$ from a discrete Gaussian with mean $\mathbf{0}$ and standard deviation s and sets $\mathsf{sk}_{ID} = \mathbf{e}$. Then $\mathcal{A}_{LWE}$ computes $\mathbf{v} = \mathbf{A} \cdot \mathbf{e} \mod q$ and programs $H(ID) = \mathbf{v}$. Finally, $\mathcal{A}_{LWE}$ returns either sk_{ID} or $H(ID)$, depending on which value $\mathcal{A}_{IBE}$ requested.
   (c) Else if the queried ID is ID^*, and $H(ID^*)$ is requested, then $\mathcal{A}_{LWE}$ samples $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_q^n$ and returns $H(ID^*) = \mathbf{v}$.
5. Eventually, $\mathcal{A}_{IBE}$ outputs a guess b' for b. $\mathcal{A}_{LWE}$ checks whether $b' = b$. If so, $\mathcal{A}_{LWE}$ outputs 0. If not, $\mathcal{A}_{LWE}$ outputs 1.

[2] The only exception is that the adversary cannot ask for sk_{ID^*}.

Analysis: First, note that $\mathcal{A}_{LWE}$ correctly simulates the adversary's queries. For each ID that $\mathcal{A}_{IBE}$ queries, $H(ID)$ is a uniformly random vector $\mathbf{v}$. And conditioned on the value of $\mathbf{v}$, sk_{ID} is a vector $\mathbf{e}$ that comes from a Gaussian-weighted distribution with mean $\mathbf{0}$ and standard deviation s such that $\mathbf{v} = \mathbf{A} \cdot \mathbf{e} \mod q$.

Next, if $\mathcal{A}_{LWE}$ was given a sample from the distribution

$$\{(\mathbf{A}', \mathbf{u}') : \mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{x}' \xleftarrow{\$} \chi^{m+1}, \mathbf{u}' = \mathbf{A}'^T \cdot \mathbf{s} + \mathbf{x}'\}$$

then $\mathcal{A}_{LWE}$ has correctly simulated the CPA security game for the IBE scheme, and $\mathcal{A}_{IBE}$ guesses $b' = b$ with non-negligible advantage. In this case, $\mathcal{A}_{LWE}$ will output 0 with probability $\frac{1}{2} + \mathsf{non\text{-}negl}(n)$.

On the other hand, if $\mathcal{A}_{LWE}$ was given a sample from

$$\{(\mathbf{A}', \mathbf{u}') : \mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}, \mathbf{u}' \xleftarrow{\$} \mathbb{Z}_q^{m+1}\}$$

Then $\mathcal{A}_{IBE}$ has no information about b. This is because $\mathcal{A}_{IBE}$ receives $\mathbf{u}' + \mathbf{b}$, in which $\mathbf{b}$ is masked by a uniformly random $\mathbf{u}'$. Then $\mathcal{A}_{IBE}$ guesses $b' = b$ with 0 advantage, and $\mathcal{A}_{LWE}$ will output 0 with probability $\frac{1}{2}$.

In summary, $\mathcal{A}_{LWE}$ will distinguish the two distributions with non-negligible advantage, which breaks the decisional LWE assumption. Since decisional LWE is assumed to be true, then there exists no PPT adversary $\mathcal{A}_{IBE}$ that breaks the CPA security of the IBE scheme. ∎

References

[ABP14] Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. Disjunctions for hash proof systems: New constructions and applications. Cryptology ePrint Archive, Paper 2014/483, 2014.

[GPV07] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. Cryptology ePrint Archive, Paper 2007/432, 2007.