6.892 Computing on Encrypted Data November 18, 2013

Lecture 10

Lecturer: Vinod Vaikuntanathan *Scribe: Conner Fromknecht*

Gaussian Sampling and Identity-Based Encryption

1 Introduction

In this lecture we will build upon the basic Identity-Based Encryption scheme we developed in the previous lecture. Our motivation stems from the fact that the trapdoor from our basic IBE scheme can be easily recovered via relatively simple attacks, which is clearly unacceptable. We will first review the ISIS_β problem, the notion of trapdoors for lattices, Gaussian sampling, and lattice switching before assembling our final construction of the IBE scheme.

2 $\text{ISIS}_\beta(q, \mathbf{A}, \mathbf{y})$:

Given prime $q, \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, and $\mathbf{y} \leftarrow \mathbb{Z}^n$. Find $\mathbf{r} \in \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{y} \mod q\}$ st. $\|\mathbf{r}\|_\infty \leq \beta$

3 Trapdoors for Lattices

Given a $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, we can define one of two notions for a trapdoor.

3.1 Strong Trapdoor

Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{A}$ admits a strong trapdoor $\mathbf{T}_A \in \mathbb{Z}^{m \times m}$ if:

- $\mathbf{A}\mathbf{T}_A = 0 \mod q$ ($\mathbf{T}_A$ is a basis of $\Lambda^\perp(\mathbf{A})$)
- $\mathbf{T}_A$ is full-rank over $\mathbb{Z}$
- $\|\mathbf{T}_A\|_\infty \leq \text{poly}(n)$

3.2 G-Trapdoor

Assuming $q = 2^k$ and $m = nk$, let

$$\mathbf{g}_k^t = [1, 2, \ldots, 2^{k-1}] \in \mathbb{Z}_q^{1 \times k}, \mathbf{S}_k = \begin{bmatrix} 2 & & & \\ -1 & 2 & & \\ & \ddots & \ddots & \\ & & -1 & 2 \end{bmatrix} \in \mathbb{Z}^{k \times k}$$

Clearly $\mathbf{S}_k$ is short, and since $\mathbf{g}_k^t \mathbf{S}_k = 0 \mod q$, $\mathbf{S}_k$ is a short basis for $\Lambda^\perp(\mathbf{g}_k^t)$. From these, we can construct a primitive canonical matrix $\mathbf{G}$ and a strong trapdoor $\mathbf{S}$ for $\mathbf{G}$ st:

$$\mathbf{G} = \begin{bmatrix} \cdots \mathbf{g}_k^t \cdots & & & \\ & \cdots \mathbf{g}_k^t \cdots & & \\ & & \ddots & \\ & & & \cdots \mathbf{g}_k^t \cdots \end{bmatrix} \in \mathbb{Z}_q^{n \times m}, \mathbf{S} = \begin{bmatrix} \mathbf{S}_k & & & \\ & \mathbf{S}_k & & \\ & & \ddots & \\ & & & \mathbf{S}_k \end{bmatrix} \in \mathbb{Z}^{m \times m}$$

$\mathbf{A}$ admits a G-Trapdoor $\mathbf{T}_A \in \mathbb{Z}^{m \times m}$ if:

- $\mathbf{A}\mathbf{T}_A = \mathbf{G} \mod q$
- $\mathbf{T}_A$ is full-rank over $\mathbb{Z}$
- $\|\mathbf{T}_A\|_\infty \leq \text{poly}(n)$

Since we know a trapdoor for $\mathbf{G}$, it is easy to find a short $\mathbf{r}' \in \{0,1\}^m$ st. $\mathbf{G}\mathbf{r}' = \mathbf{y}$. We can then solve $\mathbf{A}\mathbf{r} = \mathbf{y} \mod q$ by letting $\mathbf{r} = \mathbf{T}_A\mathbf{r}'$ and solving $\mathbf{A}\mathbf{r} = \mathbf{A}(\mathbf{T}_A\mathbf{r}') = \mathbf{G}\mathbf{r}' = \mathbf{y} \mod q$ for $\mathbf{r}'$ and simply computing $\mathbf{T}_A\mathbf{r}'$.

4 Basic IBE

In the following section, we will describe the basic IBE scheme discussed in the previous lecture. The scheme is parameterized by (n, q) which are the dimension and modulus respectively (m is assumed to be $\Theta(n \lg q)$) and consists of the tuple of four algorithms $(\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$.

- $\mathsf{Setup}(n, q) : (mpk, msk)$ st. $mpk = \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times 2m}$ and $msk = \mathbf{T}_A \in \mathbb{Z}_q^{2m \times m}$
- $\mathsf{Keygen}_{msk}(id) : sk_{id} =$ short $\mathbf{r} \in \mathbb{Z}^{2m}$ st. $\mathbf{A}\mathbf{r} = \mathbf{y} \mod q$, where $\mathbf{y} = H(id) \in \mathbb{Z}_q^n$
- $\mathsf{Enc}_{mpk}(id, \mu \in \{0,1\}) : c = \mathsf{DualRegev}(\mathbf{A}, \mathbf{y}, \mu) = (\mathbf{A}^t\mathbf{s} + \mathbf{e}, \mathbf{y}^t\mathbf{s} + \mathbf{e}' + \mu \lfloor q/2 \rceil)$
- $\mathsf{Dec}_{sk}(sk_{id}, c) : \mu \approx \mathbf{y}^t\mathbf{s} + \mathbf{e}' + \mu - \mathbf{b}$ where $\mathbf{b} = \mathbf{s}k_{id}^t(\mathbf{A}^t\mathbf{s} + \mathbf{e})$

The problem with this scheme is that it is very easy to recover $\mathbf{T}_A$, if we collect many ($O(n^2)$) equations we can simply solve for $\mathbf{r} = \mathbf{T}_A\mathbf{r}'$ algebraically.

Another attack, proposed by [Nguyen-Regev '09] recovers the secret key $\mathbf{T}_A$ by solving the Hidden Parallelepiped Problem (HPP). Here, the attacker simply plots many $\mathbf{v}_i = \mathbf{msg}_i - \mathbf{sig}_i$. Since the secret is small, the resulting plot produces a distribution that approximates the shape parallelepiped defined by $\mathbf{T}_A$, which can be used in conjunction with gradient descent to learn the secret.

5 Inverse Sampling

Ideally, we would like to have small, random, trapdoor-independent solutions to $\mathbf{A}\mathbf{r} = \mathbf{y}$. To do so, we will start by designing a distribution $\mathcal{D}$ over $\mathbb{Z}^m$ st. $\forall \mathbf{T}_A, (\mathbf{y} \leftarrow \mathbb{Z}_q^n, r \leftarrow \mathsf{SampInv}(\mathbf{A}, \mathbf{T}_A, \mathbf{y}) \overset{s}{\approx} (\mathbf{y} := \mathbf{A}\mathbf{r}, r \leftarrow \mathcal{D})$ where $\overset{s}{\approx}$ is defined as statistically indistinguishable.

Let $\mathcal{D}_{D,\sigma}(\mathbf{x})$ be a *discrete* Gaussian distribution over $\mathbb{Z}^m$. The continuous Gaussian function is described by $\Phi_\sigma(\mathbf{x}) \propto e^{\frac{-\pi\|\mathbf{x}\|^2}{\sigma^2}}$ for $\mathbf{x} \in \mathbb{R}^m$. We then define $\mathcal{D}_{D,\sigma}(\mathbf{x})$ as:

$$\mathcal{D}_{D,\sigma}(\mathbf{x}) = \begin{cases} \Phi_\sigma(\mathbf{x})/\Phi_\sigma(D) & : \forall \mathbf{x} \in D \\ 0 & : \text{otherwise} \end{cases}$$

which is poly(λ)-time sampleable with standard deviation $2^{-\lambda}$ from $\mathcal{D}_{D,\sigma}^m$ as long as $\sigma \geq \sigma_0 = \sigma_0(L)$.

1. In the basic IBE construction, the Keygen algorithm uses a deterministic inversion algorithm to generate secret keys given by the following:

Algorithm 1: $\mathsf{dBitDecomp}(y, q)$ - Deterministic Bit Decomposition

for $i = 1$ to $\lceil \lg q \rceil$ **do**
  $r_i = y - y \mod 2$
  $y = \frac{y - y \mod 2}{2}$
end for
return $(r_1, ..., r_{\lceil \lg q \rceil})$

To eliminate the linear relationship between $\mathbf{r}$ and $\mathbf{r}'$, we need to randomize the bit decomposition during Keygen:

Algorithm 2: $\mathsf{pBitDecomp}(y, q)$ - Probabilistic Bit Decomposition

for $i = 1$ to $\lceil \lg q \rceil$ **do**
  $r'_i = \mathcal{D}_{2\mathbb{Z}+(y \mod 2),\sigma'}$
  $y = \frac{y - r'_i}{2}$
end for
return $(r'_1, ..., r'_{\lceil \lg q \rceil})$

We claim that $(\mathbf{r}'_1, \ldots, \mathbf{r}'_{\lceil \lg q \rceil}) \overset{s}{\approx} \mathcal{D}_{\Lambda^{\perp}_{\mathbf{y}}(A),\sigma'}$.

2. We can also use an alternative continuous Gaussian definition: $\Phi_\sigma(\mathbf{x}) \propto e^{-\pi \mathbf{x}^t \Sigma \mathbf{x}}$.
In the basic IBE scheme, $\mathbf{r} = \mathbf{T} \cdot \mathbf{r}', \mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sqrt{\mathbf{T}\mathbf{T}^t \cdot \sigma'^2}}$. Therefore, information about the trapdoor is inherently leaked by the distribution of $\mathbf{r}$. We can counteract this by using the following algorithm which makes $\mathbf{r}$ distributed according to a discrete Gaussian with a target standard deviation σ.

Algorithm 3: $\mathsf{SampInv}(\mathbf{A}, \mathbf{T}_A, \mathbf{y})$

$\mathbf{p} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \sqrt{\sigma^2 - \mathbf{T}\mathbf{T}^t \cdot \sigma'^2}}$
$\mathbf{y}' = \mathbf{y} - \mathbf{A}\mathbf{p}$
$\mathbf{r}' = \mathsf{pBitDecomp}(\mathbf{y}', q)$
return $\mathbf{r} = \mathbf{T}_A \mathbf{r}' + \mathbf{p}$

We can see that $\mathbf{r}$ is now appropriately distributed, namely $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$ and is completely independent of $\mathbf{T}$. Additionally, correctness is displayed by the following:

$$\begin{aligned} \mathbf{A}\mathbf{r} &= \mathbf{A}\mathbf{T}\mathbf{r}' + \mathbf{A}\mathbf{p} \\ &= \mathbf{G}\mathbf{r}' + \mathbf{A}\mathbf{p} \\ &= \mathbf{y}' + \mathbf{A}\mathbf{p} \\ &= \mathbf{y} \end{aligned}$$

6 Lattice Switching

6.1 1-to-1

Let $\mathbf{T} \in \mathbb{Z}_q^{m \times m}$ satisfy $\mathbf{A}_1\mathbf{T} = \mathbf{A}_2$
$(\mathbf{A}_1, \mathbf{A}_1^t\mathbf{s} + \mathbf{e}_1) \rightarrow (\mathbf{A}_2, \mathbf{A}_2^t\mathbf{s} + \mathbf{e}_2) = (\mathbf{A}_1\mathbf{T}, \mathbf{T}^t(\mathbf{A}_1^t\mathbf{s} + \mathbf{e}_1))$

6.2 2-to-1

$\begin{pmatrix} \mathbf{A}_1, \mathbf{A}_1^t\mathbf{s} + \mathbf{e}_1 \\ \mathbf{A}_2, \mathbf{A}_2^t\mathbf{s} + \mathbf{e}_2 \end{pmatrix} \rightarrow (\mathbf{A}_3, \mathbf{A}_3^t\mathbf{s} + \mathbf{e}_3)$ where $\mathbf{A}_3 = \mathbf{A}_1\mathbf{T}_1 + \mathbf{A}_2\mathbf{T}_2$

6.3 l-to-1

$[\mathbf{A}_1|\mathbf{A}_2|\ldots|\mathbf{A}_l]\begin{pmatrix}\mathbf{r}_1\\ \mathbf{r}_2\\ \vdots\\ \mathbf{r}_l\end{pmatrix} = \mathbf{A}_1\mathbf{r}_1 + \mathbf{A}_2\mathbf{r}_2 + \ldots + \mathbf{A}_l\mathbf{r}_l = \mathbf{y}$ where $\mathbf{r}_2 \ldots \mathbf{r}_l \leftarrow \mathcal{D}_{\mathbb{Z}^m,\sigma}$ are short.

Solve $\mathbf{A}_1\mathbf{r}_1 = \mathbf{y} - \sum_{i>1}\mathbf{A}_i\mathbf{r}_i$

7 Final IBE Scheme

This section details our final IBE implementation, which introduces changes to all but the decryption algorithm from our basic IBE scheme.

- $\mathsf{Setup}(n, q)$: $mpk = \begin{pmatrix}\mathbf{A}_{1,0} & \mathbf{A}_{2,0} & \ldots & \mathbf{A}_{l,0}\\ \mathbf{A}_{1,1} & \mathbf{A}_{2,1} & \ldots & \mathbf{A}_{l,1}\end{pmatrix}, msk = \mathbf{T}_{\mathbf{A}_{1,0}}, \mathbf{T}_{\mathbf{A}_{1,1}}$ using the Guassian sampling techniques described above.
- $\mathsf{Keygen}_{msk}(id \in \{0,1\}^l) : sk_{id} = \mathsf{SampInv}(\mathbf{A}_{id}, \mathbf{T}_{\mathbf{A}_{1,id_1}}, \mathbf{y})$, where $\mathbf{A}_{id} = [\mathbf{A}_{1,id_1}|\mathbf{A}_{2,id_2}|\ldots|\mathbf{A}_{l,id_l}]$
- $\mathsf{Enc}_{mpk}(id, \mu) : c = \mathbf{dualRegev}(\mathbf{A}_{id}, \mathbf{y}, \mu)$

7.1 IBE Security

Let us assume that in the Setup phase we generate trapdoors for all $\mathbf{A}_{i,id_i}$ except for a specific sequence id^*, say 0001. We can then depict our mpk as $\begin{pmatrix}\mathbf{A}^*_{1,0} & \mathbf{A}^*_{2,0} & \mathbf{A}^*_{3,0} & \mathbf{A}_{4,0}\\ \mathbf{A}_{1,1} & \mathbf{A}_{2,1} & \mathbf{A}_{3,1} & \mathbf{A}^*_{4,1}\end{pmatrix}$ where $\mathbf{A}^*$ denotes public keys without trapdoors.

Consider an adversary in the following game:

Oracle	*Adversary*
$mpk \rightarrow$	
	$\leftarrow id$
$sk_{id} \rightarrow$	
	$\leftarrow id^*, \mu_1, \mu_2$
$\mathsf{Enc}_{mpk}(id^*, \mu_b) \rightarrow$	
	$\downarrow b'$

As is standard, the adversary wins if $b' = b$. However, we can see that no matter how many sk_{id}'s the adversary requests, he will never be able to find a trapdoor to invert the encryption under id^*. Since the trapdoor for id^* simply doesn't exist, the best the adversary can do is guess as to which message was returned.