CSC 2419: Lattice-based Cryptography **Fall 2025**

Lecture 3: Trapdoor Functions from LWE

Instructor: Akshayaram Srinivasan *Scribe: Ashwin Karthikeyan*

Date: Sept 22, 2025

3.1 Multi-bit PKE

Last week we constructed a Public Key Encryption scheme for single bit messages. To extend this to multi-bit messages $\mathbf{m} \in \{0,1\}^l$, we can simply decompose the message into bits and encrypt each bit seperately, i.e.

$$\tilde{\mathsf{Enc}}(\mathsf{pk}, \mathbf{m}) = (\mathsf{Enc}(\mathsf{pk}, \mathbf{m}_1), \mathsf{Enc}(\mathsf{pk}, \mathbf{m}_2), \ldots, \mathsf{Enc}(\mathsf{pk}, \mathbf{m}_l))$$

It should be noted that independent randomness is used to encrypt each bit of the message. Correctness of the multi-bit PKE scheme follows from the correctness of the underlying PKE scheme. To argue the IND-CPA security of the new PKE scheme, we show that for $\forall \mathbf{m}^1, \mathbf{m}^2 \in \{0,1\}^l)$, no PPT adversary can distinguish between the corresponding ciphertexts except with negligible probability i.e. we have to show that

$$(\mathsf{pk}, \tilde{\mathsf{Enc}}(\mathsf{pk}, \mathbf{m}^1) \approx_c (\mathsf{pk}, \tilde{\mathsf{Enc}}(\mathsf{pk}, \mathbf{m}^2)$$

We argue that the distributions are indistinguishable through a series of hybrid arguemments and provide a sketch of the indistinguishability between consecutive hybrids.

- *Hybrid 0*: This hybrid corresponds to the distribution $(\mathsf{pk}, \tilde{\mathsf{Enc}}(\mathsf{pk}, \mathbf{m}^1) = (\mathsf{pk}, \left\{\mathsf{Enc}(\mathsf{pk}, \mathbf{m}_i^1)\right\}_{i\in[l]})$
- *Hybrid 1*: Same as previous hybrid, but $\mathbf{m}_1^1$ is replaced with $\mathbf{m}_1^2$ i.e. the new distribution is $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathbf{m}_1^2, \left\{\mathsf{Enc}(\mathsf{pk}, \mathbf{m}_i^1)\right\}_{i\in\{2,l\}})$. The only difference between the distributions *Hybrid 1* and *Hybrid 0* is that in *Hybrid 1*, the adversary receives $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathbf{m}_1^2))$, and in *Hybrid 0*, the adversary receives $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathbf{m}_1^1))$. Since the distributions $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathbf{m}_1^2)) \approx_c (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathbf{m}_1^1))$ from the IND-CPA property of the underlying PKE scheme, we can argue that *Hybrid 1* is indistinguishable from *Hybrid 0*.
- *Hybrid i*: Same as *Hybrid i-1*, but $\mathbf{m}_i^1$ is replaced with $\mathbf{m}_i^2$. The indistinguishability argument for *Hybrid i* and *Hybrid i-1* is identical to the argument presented above.
- *Hybrid l*: Same as the distribution $(\mathsf{pk}, \tilde{\mathsf{Enc}}(\mathsf{pk}, \mathbf{m}^2)$

Since the distributions in hybrids *Hybrid i* and *Hybrid i-1* are indistinguishable except with $\mathsf{negl}(\lambda)$ probability, by a simple union bound, hybrids *Hybrid 1* and *Hybrid l* as indistinguishable except with $l \cdot \mathsf{negl}(\lambda) = \mathsf{negl}'(\lambda)$ probability.

3.2 Trapdoor Functions:

In this section, we present another important cryptographic primitive, the Trapdoor Function.

Definition 3.1 (Trapdoor Function (TDF)) *For security parameter $\lambda \in \mathbb{N}$ and dimensions $n, m = \mathsf{poly}(\lambda)$, the trapdoor function consists of a PPT tuple of algorithms* $\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Invert}$ *where:*

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{td})$ *On input the security parameter, the key generation algorithm returns a public key* pk *and trapdoor* td.
- $y := \mathsf{Eval}(\mathsf{pk}, x \in \{0,1\}^n)$*: On input the public key* pk *and message x, the evaluation algorithm returns* $y \in \{0,1\}^n$.
- $x := \mathsf{Invert}(\mathsf{td}, y)$*: On input the trapdoor* td *and $y \in \{0,1\}^m$, the invert algorithm returns preimage* $x' \in \{0,1\}^n$.

The Trapdoor function satisfies the following property:

- ***Correctness****: for all security parameters $\lambda \in \mathbb{N}$, $(\mathsf{pk}, \mathsf{td}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, and $x \in \{0,1\}^n$, the following holds true:*
$$\Pr[\mathsf{Invert}(\mathsf{td}, \mathsf{Eval}(\mathsf{pk}, x)) = x] = 1$$
- ***Security****: For $\lambda \in \mathbb{N}$, $(\mathsf{pk}, \mathsf{td}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, randomly samples $x \xleftarrow{\$} \{0,1\}^n$, and for all PPT adversary $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that*
$$\Pr_x[\mathcal{A}(\mathsf{pk}, y) = x | y = \mathsf{Eval}(\mathsf{pk}, x)] \leq 1/2^n + \mathsf{negl}(\lambda)$$
In other words, no PPT adversary can determine the preimage of y except with negl probability more than randomly guessing the preimage.

Remark 3.2 (Instantiation of TDF) *In the seminal work of Goldreich and Levin [gl89], the authors introduce the notion of hardcore-predicate corresponding a one way function f. Let $h(x; r) \to \{0,1\}$ be a hardcore predicate for the one-way function f. The Goldreich-Levin Theorem states that there exists a hardcore predicate $h(x; r)$ such that for every one-way function f, following two distributions are computationally indistinguishable:*
$$\{f(x), r, h(x;r)\}_{x \xleftarrow{\$} \{0,1\}^*} \approx_c \{f(x), r, b\}_{x \xleftarrow{\$} \{0,1\}^*, b \xleftarrow{\$} \{0,1\}}$$
.

Using this notion of hardcore-predicate and assuming the existence of one-way functions, [bhsv89] proposed a construction of Trapdoor Functions.

3.2.1 Public Key Encryption from TDF

We now describe the construction of a public key encryption scheme assuming the existence of a trapdoor function $(\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Invert})$ and a hardcore predicate $h(x; r)$ corresponding to the $\mathsf{Eval}()$ function[1]. In the following discussion, we define the tuple of algorithms for PKE:

- $\mathsf{KeyGen}(1^\lambda)$: Invoke the key generation of the TDF $(\mathsf{pk}_{\mathsf{TDF}}, \mathsf{td}_{\mathsf{TDF}}) \leftarrow \mathsf{TDF.KeyGen}(1^\lambda)$ Set the public key of the encryption scheme as $\mathsf{pk} = \mathsf{pk}_{\mathsf{TDF}}$ and the secret key as $\mathsf{sk} = \mathsf{td}_{\mathsf{TDF}}$
- $\mathsf{Enc}(\mathsf{pk}, \mu)$: Sample $r \xleftarrow{\$} \{0,1\}^n$ and define the ciphertext as $\mathsf{ct} = (\mathsf{Eval}(\mathsf{pk}, r), h(r; r') \oplus \mu, r)$

[1] Trapdoor function is implicitly a one-way function. Therefore, by Goldreich-Levin Theorem, such a hardcore predicate exists

- $\mathsf{Dec}(\mathsf{sk} = \mathsf{td}, \mathsf{ct})$: Parse $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$. Using the trapdoor, compute $r := \mathsf{Invert}(\mathsf{td}, \mathsf{ct}_1)$ and return $h(r; r') \oplus \mathsf{ct}_2$.

Correctness of the encryption scheme follows from correctness of TDF. To argue IND-CPA security of the scheme, note that we want to prove the following:

$$(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 1)) \approx_c (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 0))$$

.

Observe that $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 0)) = (\mathsf{Eval}(\mathsf{pk}, r), h(r; r'), r') \approx_c (\mathsf{Eval}(\mathsf{pk}, r), u, r')_{u \xleftarrow{\$} \{0,1\}}$ from Goldreich-Levin Theorem. Using a similar argument, it can be concluded that

$$(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 1)) \approx_c (\mathsf{Eval}(\mathsf{pk}, r), u, r')_{u \xleftarrow{\$} \{0,1\}}$$

3.3 How to construct TDFs for LWE:

Recall the search variant of LWE problem. In that problem, given the samples $(\mathbf{A}, \mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T)$, we wanted to find the secret $\mathbf{s}^T$. If we had some "trapdoot" $\mathbf{T}$ which could be used to recover the secret $\mathbf{s}^T$, we could think of this as a trapdoor function with the following formulation:

- $\mathsf{KeyGen}(1^\lambda)$: Sample random $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and compute the trapdoor $\mathsf{td} = \mathbf{T}$
- $\mathsf{Eval}(\mathsf{pk}, (\mathbf{s}^T, \mathbf{e}^T))$: Return $(\mathbf{A}, \mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T$
- $\mathsf{Invert}(\mathsf{td}, (\mathbf{A}, \mathbf{b}^T))$: Use the trapdoor to recover secret $\mathbf{s}^T$.

We now define the desirable properties of the trapdoor $\mathbf{T}$.

Properties for Trapdoor: For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we defined a trapdoor $\mathbf{T} \in \mathbb{Z}_q^{m\times m}$ such that

1. $\mathbf{AT} = 0^{n\times m} \mod q$
2. If $\mathbf{T} = \begin{bmatrix} \vdots & & \vdots \\ \mathbf{t}_1 & \dots & \mathbf{t}_m \\ \vdots & & \vdots \end{bmatrix}$, then $||t_i||_\infty \leq B$ (low norm).
3. T has full rank over $\mathbb{Z}$.

Given a trapdoor $\mathbf{T}$ with aforementioned properties, we can now describe the $\mathsf{Invert}()$ function: The invert function computes

$$\mathbf{b}^T\mathbf{T} = \mathbf{s}^T(\mathbf{AT}) + \mathbf{e}^T\mathbf{T} = \mathbf{e}^T\mathbf{T} \; modq$$

. Since both $\mathbf{e}$ and $\mathbf{T}$ have low norm[2], we have $\mathbf{b}^T\mathbf{T} = \mathbf{e}^T\mathbf{T}$ over $\mathbb{Z}$ ($\mathbf{e}^T\mathbf{T}$ doesn't wrap around $\mod q$). We can use **Gaussian Elimination** to compute $\mathbf{e}^T$(and consequently, the secret $\mathbf{s}$) from the computation above.

[2]We consider parameters such that q/B is sub-exponential(See Lecture 2).

3.3.1 How to sample $(\mathbf{A}, \mathbf{T})$:

We will first define a Gadget matrix $\mathbf{G}$ and define the Trapdoor for this Gadget matrix. We will then use this Trapdoor to construct a Trapdoor matrix for $\mathbf{A}$. Here, instead of sampling $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, we will sample $\mathbf{A}$ from a distribution that is indistinguishable from uniform distribution.

3.3.1.1 Defining G and its Trapdoor:

Define

$$\mathbf{G} := \begin{bmatrix} 1 & 2 & 4 & \dots & q/2 & & & & & \\ & & & & & 1 & 2 & 4 & \dots & q/2 \\ & & & & & & & & & \ddots \end{bmatrix}$$

where q is a power of 2 and $\mathbf{G}$ is a $n \times n\log(q)$ matrix. Note that $G = I \otimes \mathbf{g}^T$ where $\mathbf{g}^T = \begin{bmatrix} 1 & 2 & 4 & \dots & q/2 \end{bmatrix}$. Note that the dot product of $\mathbf{g}^T$ and a binary vector is an element in $\mathbb{Z}_q$. Let $\mathbf{G}^{-1}$ denote the bit-decomposition function (not inverse) such that:

$$\mathbf{G}^{-1} : \mathbb{Z}_q \rightarrow \{0,1\}^{n \log q}$$

such that $\mathbf{G}^{-1}\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ gives the bit decomposition of x_1 to x_n stacked on top of each other. Consequently, $\mathbf{G}\mathbf{G}^{-1}(x) = x$

Now, note that

$$\mathbf{g}^T \begin{bmatrix} 2 & & & \\ -1 & 2 & & \\ & -1 & & \\ & & \ddots & \\ & & & 2 \end{bmatrix} = 0^{\log q} (\mod q)$$

So, we can define

$$\mathbf{T}_g := \begin{bmatrix} 2 & & & \\ -1 & 2 & & \\ & -1 & & \\ & & \ddots & \\ & & & 2 \end{bmatrix}$$

to get the Trapdoor matrix $(\mathbf{I}_{n\times n} \otimes \mathbf{T}_g)$ for G. This is because:

$$(\mathbf{I}_{n\times n} \otimes \mathbf{g}^T) \cdot (\mathbf{I}_{n\times n} \otimes \mathbf{T}_g) = (I \cdot I) \otimes (\mathbf{g}^T \mathbf{T}_g) = 0^{n \times n \log q}$$

Additionally, $(\mathbf{I}_{n\times n} \otimes \mathbf{g}^T)$ is low norm ($||I_{n\times n} \otimes \mathbf{g}^T||_\infty = 2$) and full rank over $\mathbb{Z}$, satisfying the desired properties of the trapdoor for the gadget matrix.

3.3.1.2 Defining A and its Trapdoor:

Let $\mathbf{B} \leftarrow \mathbb{Z}_q^{n\times m}$ be sampled uniformly at random and

$$\mathbf{A} = \{\mathbf{B} || \mathbf{B} \cdot \mathbf{R} + \mathbf{G}\}$$

Where || denotes concatenation, and $\mathbf{R}$ is sampled uniformly at random from $\{0,1\}^{m\times n\log(q)}$. So, $\mathbf{A}$ has dimension $n \times (m + n\log(q))$

Note that the marginal distribution of $\mathbf{A}$ is statistically close to uniform(from *Leftover Hash Lemma*). Now,

$$\mathbf{A}\begin{bmatrix}\mathbf{I} & -\mathbf{R} \\ 0 & \mathbf{I}\end{bmatrix} = [\mathbf{B}||\mathbf{G}]$$

So, we have

$$\mathbf{A}\begin{bmatrix}\mathbf{I} & -\mathbf{R} \\ 0 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & 0 \\ \mathbf{G}^{-1}(\mathbf{B}) & \mathbf{T}_g\end{bmatrix} = 0(\mod q)$$

, which gives us the trapdoor

$$\mathbf{T}_A = \begin{bmatrix}\mathbf{I} & -\mathbf{R} \\ 0 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & 0 \\ \mathbf{G}^{-1}(\mathbf{B}) & \mathbf{T}_g\end{bmatrix}$$

$\mathbf{A}\mathbf{T}_A = 0$ as shown, and since the product of two full-rank square matrices is full-rank, $\mathbf{T}_A$ is full-rank as well.

References

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan (2008). "How to Use a Short Basis: Trapdoors for Hard Lattices and New Cryptographic Constructions." Electronic Colloquium on Computational Complexity (ECCC), 14.