Lattices and Homomorphic Encryption, Spring 2013	Instructors: Shai Halevi, Tal Malkin
Trapdoor Sampling	
March 26, 2013	Scribe: Yuan Kang

1 Trapdoor Sampling [MP12]

The GPV Signature scheme assumes that we can generate trapdoor matrices. This process has two steps:

1. Construct a special purpose, "easy lattice", G, [1] that is not random at all, as described in the handout, and

2. Show how to sample a nearly-uniform A, together with a trapdoor that "maps" A to G

The "easy lattice" is $G \in \mathbb{Z}_q^{n \times m'}, m' = \lceil n\log(q) \rceil$, such that:

(a) It is easy to sample $\mathcal{D}_{\mathcal{L}_{\vec{u}}^{\perp}(G),s}$ for any $\vec{u} \in \mathbb{Z}_q^n$ and parameter $s \geq 2\sqrt{n}$. [2]

(b) Given $[\vec{s}G + \vec{e}]$, with small $||\vec{e}||_\infty < \frac{q}{4}$, one can efficiently recover $\vec{s}$.

1.1 Step (2): Mapping A to G

Definition 1. *As in the first property, denote:*

$$m' = \lceil n\log(q) \rceil$$

In addition denote

$$m'' = \lceil n\log(q) + \sqrt{n} \rceil$$

and

$$m = m' + m'' = \lceil 2n\log(q) + \sqrt{n} \rceil$$

Let $A \in \mathbb{Z}^{n \times m}$ denote

$$A = [\underbrace{\overline{A}}_{m''} | \underbrace{A_1}_{m'}]$$

A matrix $R \in \mathbb{Z}_q^{m'' \times m'}$ is a trapdoor of A iff

- *R is "small"*
- $\underbrace{A_1}_{n \times m'} = \underbrace{G}_{n \times m'} - \underbrace{\underbrace{\overline{A}}_{n \times m''} \underbrace{R}_{m'' \times m'}}_{n \times m'}$. *In matrix notation:* $A = [\overline{A}|G] \begin{pmatrix} I & -R \\ 0 & I \end{pmatrix}$

The algorithm to generate (A, R) proceeds as follows:

[1] ie. it is easy to solve LWE or SIS

[2] Recall the definition $\mathcal{L}_{\vec{u}}^{\perp}(A) = \{\vec{x} \in \mathbb{Z}_q^n | A\vec{x} = \vec{u} \mod q\}$

- Choose $R \in \mathbb{Z}_q^{m'' \times m'}$, where each entry in R is chosen at random from the discrete Gaussian, $\mathcal{D}_{\mathbb{Z},\sqrt{n}}$. R is the trapdoor, and note that it is "small," for example with high probability, we have for all $\vec{x}$, that $||\vec{x}R||_\infty \leq ||\vec{x}||_\infty 2n\log(q)$, and the same applies for $||.||_2$ (so $S_1(R) < 2n\log(q)$).
- To choose A, first draw a uniform matrix $\overline{A} \in_R \mathbb{Z}_q^{n \times m''}$, then set

$$\begin{aligned} A &= [\overline{A}|G] \begin{pmatrix} I & -R \\ 0 & I \end{pmatrix} \\ &= [\overline{A}|G - \overline{A}R] \in \mathbb{Z}_q^{n \times (m'+m'')} \end{aligned}$$

Fact 1. *A is nearly uniform. Recall that $f_{\overline{A}} = \overline{A}\vec{x} \mod q$ is a strong seeded extractor, and the columns of R have high min-entropy, so $\overline{A}R$ is nearly uniform, even given $\overline{A}$.*

Fact 2. *If we can solve LWE for G, then R lets us also solve for A.* [3] *Given input $\vec{b} = \vec{s}A + \vec{e}$, where we denote $\vec{e} = [\underbrace{\vec{e}_1}_{m''} | \underbrace{\vec{e}_2}_{m'}]$, we have*

$$\begin{aligned} \vec{b}\begin{pmatrix} I & R \\ 0 & I \end{pmatrix} &= (\vec{s}A + [\vec{e}_1|\vec{e}_2])\begin{pmatrix} I & R \\ 0 & I \end{pmatrix} \\ &= \vec{s}[\overline{A}|G]\begin{pmatrix} I & -R \\ 0 & I \end{pmatrix}\begin{pmatrix} I & R \\ 0 & I \end{pmatrix} + [\vec{e}_1|\vec{e}_2]\begin{pmatrix} I & R \\ 0 & I \end{pmatrix} \\ &= \vec{s}[\overline{A}|G] + [\vec{e}_1|\vec{e}_1R + \vec{e}_2] \end{aligned}$$

In particular, considering only the last m' entries, we have

$$\vec{b}\begin{pmatrix} R \\ I \end{pmatrix} = \vec{s}G + \underbrace{(\vec{e}_1R + \vec{e}_2)}_{\vec{e'}}$$

As long as $||\vec{e'}||_\infty \leq ||\vec{e}_1||_\infty 2n\log(q) + ||\vec{e}_2||_\infty < \frac{q}{4}$, we can recover $\vec{s}$ from $\vec{s}G + \vec{e'}$. The first inequality follows from the choice of a "small" R, and the second inequality is true as long as $||\vec{e}_1||_\infty, ||\vec{e}_2||_\infty \ll \frac{q}{n\log(q)}$.

Fact 3. *If we can sample from $\mathcal{D}_{\mathcal{L}^\perp_{\vec{u}}(G),s}$, then using R, we can sample $\mathcal{D}_{\mathcal{L}^\perp_{\vec{u}}(A),s'}$, where s' is not much bigger than s.*

- *<u>First attempt</u>: Draw $\vec{z} \leftarrow \mathcal{D}_{\mathcal{L}^\perp_{\vec{u}}(G),s}$, output $\vec{x} = \begin{pmatrix} R \\ I \end{pmatrix}\vec{z}$. This "almost works"; we have $A\vec{x} = A\begin{pmatrix} R \\ I \end{pmatrix}\vec{z} = G\vec{z} = \vec{u}$, and $||\vec{x}||_\infty \leq ||R\vec{z}||_\infty + ||\vec{z}||_\infty \leq (2n\log(q) + 1)||\vec{z}||_\infty$, as needed for SIS. But if $\vec{z}$ is a spherical Gaussian, then $\vec{x}$ is an ellipsoid Gaussian. Even worse, the covariance of $\vec{x}$ has the shape $s^2\begin{pmatrix} R \\ I \end{pmatrix}[R^T|I]$, so after enough samples, we can get the shape of R and recover R itself.*
- *<u>Better attempt</u>: Use "perturbation" [Pei10]. Roughly, choose $\vec{p}$ from an ellipsoid that cancels out that of $\vec{x}$, and output $\vec{p} + \vec{x}$:*

[3] Given input $A, \vec{b} = \vec{s}A + \vec{e}$, for "secret" $\vec{s}$, and "small" $\vec{e}$, find $\vec{s}$.

– *Define the covariance matrix* $\Sigma = \underbrace{s^2 I}_{\text{what we aim for}} - \underbrace{\begin{pmatrix} R \\ I \end{pmatrix} [R^T | I]}_{\text{the "shape" of } \vec{x}}$. *Note that* s *must be large enough so that* Σ *is positive (else it cannot be a covariance matrix). Specifically, we need to have* $s > 1 + S_1(R)$.

– *Sample from the ellipsoid discrete Gaussian* $\vec{p} \leftarrow \underbrace{\mathcal{D}_{\mathbb{Z}^m, s\sqrt{n}\sqrt{\Sigma}}}_{\text{"perturbation"}}$

– *Calculate the syndrome* $\vec{v} = \vec{u} - A\vec{p} \mod q$

– *Sample* $\vec{z} \leftarrow \mathcal{D}_{\mathcal{L}^{\perp}_{\vec{v}}(G), 2\sqrt{n}}$, *then set* $\vec{x} = \begin{pmatrix} R \\ I \end{pmatrix} \vec{z}$

– *Output* $\vec{y} = \vec{x} + \vec{p}$

Clearly we have $A\vec{y} = A\vec{x} + A\vec{p} = \vec{v} + A\vec{p} = \vec{u}$. *Moreover,* $\vec{p}$ *has covariance* $4n\Sigma$, *and* $\vec{x}$ *has covariance* $4n \begin{pmatrix} R \\ I \end{pmatrix} [R^T | I]$, *so if they were independent, we would expect their covariance matrices to add, and we get* $4n(\begin{pmatrix} R \\ I \end{pmatrix} [R^T | I] + \Sigma) = 4ns^2 I$.

They are not quite independent, since the mean of $\vec{z}$ *depends on* $\vec{p}$, *but only via* $A\vec{p}$ *in* $\vec{v}$, *which does not give much information about* $\vec{p}$. *We can think of first choosing* $\vec{v}$ *at random, then drawing* $\vec{p}$ *from the discrete Gaussian. Once* $\vec{v}$ *is fixed,* $\vec{p}$ *and* $\vec{x}$ *are independent and their covariances add; since we choose* $\vec{z}$ *from a Gaussian wider than* $\eta_\epsilon(\mathcal{L}^{\perp}(A))$, *for a negligible* ϵ, *the covariance behaves as we expect.*

2 Trapdoor Delegation

Given a trapdoor, R, for $A \in \mathbb{Z}_q^{n \times m}$, generate a trapdoor, R', for an extension of A, $A' = [A|A_1]$, where $A_1 \in \mathbb{Z}_q^{n \times m'}$ is an arbitrary matrix (eg. it can be random), and $m' \geq \lceil n\log(q) \rceil$.

TDelegate(A, R, A_1):

- Calculate $\Delta = G - A_1$. Denote the columns of Δ by $\Delta = (\vec{\delta}_1 | \vec{\delta}_2 | \ldots | \vec{\delta}_{m'})$.
- For $i \in \{1, 2, \ldots, m'\}$, use R to sample from $\mathcal{D}_{\mathcal{L}^{\perp}_{\vec{\delta}_i}(A), s}$, where $s = \lceil 2 + S_1(R) \rceil \approx 2n\log(q) > \eta_\epsilon(\mathcal{L}^{\perp}(A))$ for some negligible ϵ. Denote $\vec{r'}_i \leftarrow \mathcal{D}_{\mathcal{L}^{\perp}_{\vec{\delta}_i}(A), s}$.
- Output the new trapdoor, $R' = (\vec{r'}_1 | \vec{r'}_2 | \ldots | \vec{r'}_{m'}) \in \mathbb{Z}_q^{n \times m'}$.

By construction $A\vec{r'}_i = \vec{\delta}_i \mod q$, so $AR' = \Delta$, and therefore we have

$$A' = (A|A_1) = (A|G - \Delta) = (A|G - AR')$$

So R' is indeed a trapdoor for A'. Also, R' is "small"; roughly, the size of each column of R' is approximately $\sqrt{m}s$, so $S_1(R') \approx \sqrt{m}S_1(R) \approx (n\log(q))^{\frac{3}{2}}$.

Note that if (A, A_1) are random, the distribution of (A', R') is the same as the output of TGen, except for larger parameters, $\tilde{m} = m + m'$, and $S_1(R') \approx (n\log(q))^{\frac{3}{2}}$.

References

[1] Daniele Micciancio and Chris Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In David Pointcheval and Thomas Johansson (editors) *Advances in Cryptology*, EUROCRYPT 2012, pages 700-718, Heidelberg, Germany, 2012. Springer.

[2] Chris Peikert. An Efficient and Parallel Gaussian Sampler for Lattices. In Tal Rabin (editor) *Advances in Cryptology*, CRYPTO 2010, pages 80-97, Heidelberg, Germany, 2010. Springer.