Hardness: For $m = \text{poly}(n)$ and $q \geq \beta \cdot \text{poly}(n)$

solving $SIS_{n,q,\beta,m}$ is at least as

hard as solving $\text{Gap}SVP_\gamma$ and $SIVP_\gamma$

with $\gamma(n) = \beta \cdot \tilde{O}(\sqrt{n})$

$(\beta \cdot \text{poly}(n))$

DEF (LWE) For $\vec{s} \in \mathbb{Z}_q^n$, the LWE

distribution $A_{\vec{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$

is obtained by sampling $\vec{a} \in \mathbb{Z}_q^n$ RANDOM and $e \leftarrow \chi$ and outputting:

$$\vec{a}, \quad b = \langle \vec{s}, \vec{a} \rangle + e \mod q$$

given m samples $(\vec{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ from $A_{\vec{s}, \chi}$ for RANDOM $\vec{s}$, find $\vec{s}$.

SEARCH-$LWE_{n,q,\chi,m}$

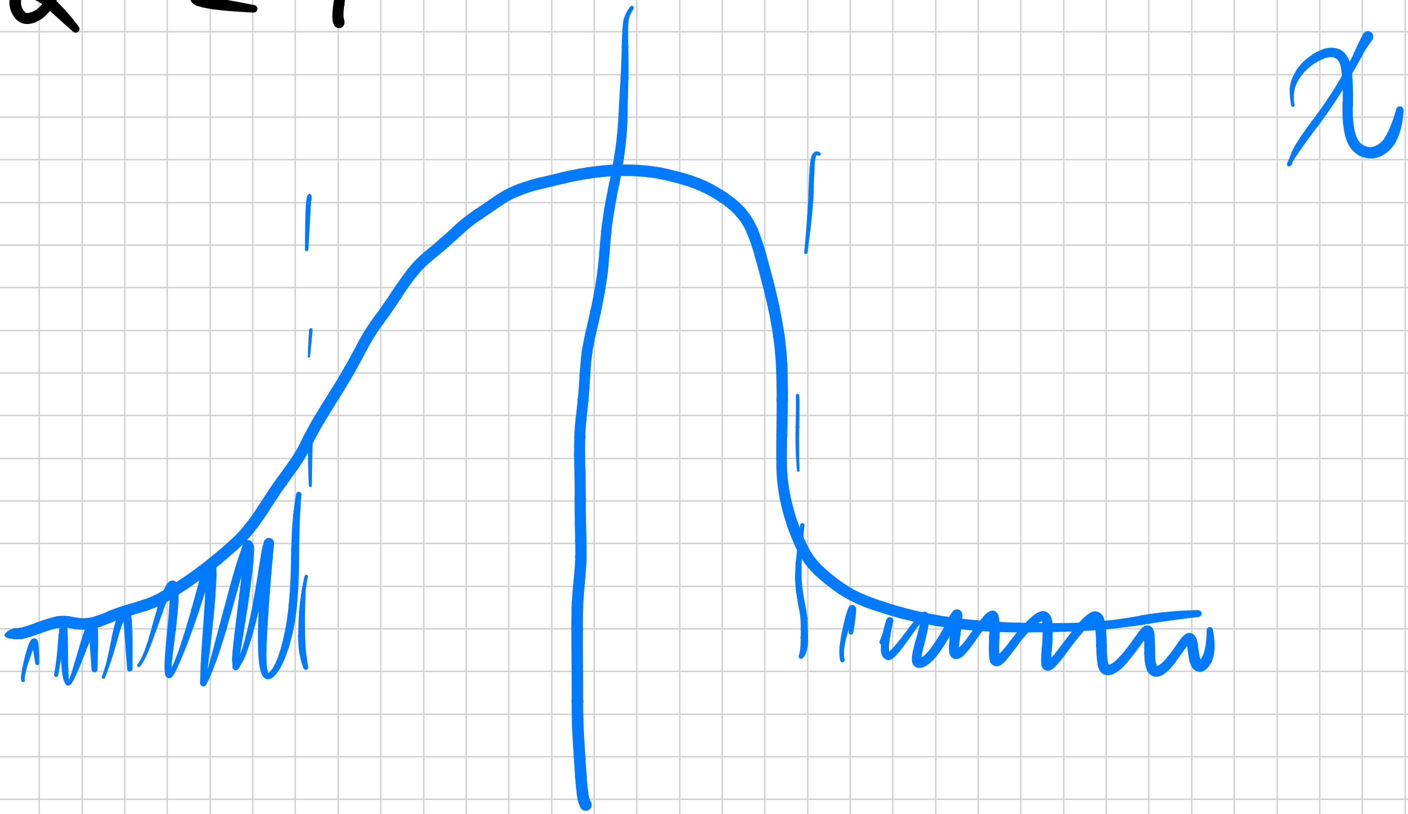
REMARKS:

- Without noise, the problem is easy.
- Error distribution: χ is taken

To be any distribution s.t.

$$\Pr[\,|e| > \alpha \cdot q \;:\; e \leftarrow \chi\,] \leq \mathrm{negl}(\lambda)$$

for $\alpha \ll 1$



$$\mathbb{Z}_q = \left(\lfloor -q/2 \rfloor, \ldots, 0, \ldots, \lfloor q/2 \rfloor \right)$$

- We can combine the m samples in this way: $(A, \vec{b})$ s.t. $A \in \mathbb{Z}_q^{m \times n}$ and $\vec{b} = \vec{s}^{\,t} \cdot A + \vec{e}$ and $\vec{e} \leftarrow \chi^m$

↳ mod q

- Decisional version: Distinguish $(A, \vec{b})$ from uniform $(A, \vec{b})$ over $\mathbb{Z}_q^{m \times (n+1)}$. The two are equivalent.

Hardness: For $m = \text{poly}(n)$ and $q < 2^{\text{poly}(n)}$ then $LWE_{n,q,\chi,m}$ using χ the otherwise, function w.p. $\alpha q \geq 2\sqrt{n}$ $(0<\alpha<1)$ is at least as hard as GapSVP$_\gamma$ and SIVP$_\gamma$ for $\gamma = \tilde{O}(n/\alpha)$.

REGEV PKE

This is based on LWE:

-) $\underline{K}\underline{Gen}(\underline{1}^\lambda)$: $\vec{s} \leftarrow \mathbb{Z}_q^n$ is the sk.

The pk consists of $m \simeq (n+1)\log q$ samples $\underbrace{(\bar{a}_i, b_i = \langle \vec{s}, \bar{a}_i \rangle + e_i)}_{\in \mathbb{Z}_q^{n+1}}$

We can view them as:

$$A = \begin{bmatrix} \bar{A} \\ \vec{b}^t \end{bmatrix} \in \mathbb{Z}_q^{(n+1) \times m}$$

Note: $(-\vec{s}, 1)^t \cdot A$

$$= -\vec{s}^t \cdot \bar{A} + \vec{b}^t = \vec{e}^t \approx \vec{0} \mod q$$

$\rightarrow$ <u>Enc</u> $(\underline{pk}, \underline{\mu \in \{0,1\}})$: Pick $\vec{r} \leftarrow \{0,1\}^m$ and output $\vec{c} = A \cdot \vec{r} + (\vec{0}, \mu \cdot \lfloor q/2 \rfloor) \in \mathbb{Z}_q^{n+1}$

$\rightarrow$ Dec (sk, $\vec{c}$) : Compute

$$(-\vec{s}, 1)^t \cdot \vec{c} = (-\vec{s}, 1)^t \cdot A \cdot \vec{r} + \mu \lfloor q/2 \rfloor$$
$$= \vec{e}^t \cdot \vec{r} + \mu \lfloor q/2 \rfloor$$
$$\approx \mu \lfloor q/2 \rfloor$$

(e.g. output 1 iff the above is larger than q/4.)

Correctness : It works as long as

$\langle \vec{e}, \vec{r} \rangle$ is less than $q/4$. If x is a discrete Gaussian w.p. σ then $\langle \vec{e}, \vec{r} \rangle$ has magnitude $\leq$

$$\sigma\sqrt{m \ln(1/\varepsilon)/\pi} \quad \text{w.p.} \geq 1 - 2\varepsilon.$$

In particular, we can set $\sigma = \Theta(\sqrt{n})$ and $q = \tilde{O}(n)$ which corresponds to $\alpha = \sigma/q = 1/\tilde{O}(\sqrt{n})$ and $\gamma = \tilde{O}(n^{3/2})$.

THM Assuming LWE is hard, then Regev's PKE is CPA-secure.

Sketch of proof. Roughly, it works like this:

- First, switch A the pk to uniform over $\mathbb{Z}_q^{(m+1)\times m}$. No PPT adv. can distinguish by the LWE assumption. The reduction is immediate.
- Second, we can use the LEFTOVER

HASH LEMMA to show that $A \cdot \vec{r}$ is stat. ind. from random so long as $\vec{r}$ has enough min-entropy, i.e. $m \gtrsim (n+1) \cdot \log q$. (In other words, the hash function $A \cdot \vec{r}$ is UNIVERSAL and thus is also a seeded extractor.)

$\Rightarrow \mu$ is information-theoretically hidden. $\square$

We can also encrypt multiple bits. Trivial ly, we can just encrypt each bit independently: given $\vec{\mu} = (\mu_1, \dots, \mu_\ell)$ we can always output:

$$\mathrm{Enc}(pk, \mu_1), \dots, \mathrm{Enc}(pk, \mu_\ell).$$

(We can also share $\bar{A}$ between multiple users.).

$\Rightarrow$ ctx size $(n+1) \cdot \ell$ over $\mathbb{Z}_q$

We can do better. We take $S \in \mathbb{Z}_q^{n \times \ell}$

to be a longer secret key and pk to be

$$A = \begin{bmatrix} \bar{A} \\ B \approx S^t \cdot \bar{A} \end{bmatrix} \in \mathbb{Z}_q^{(n+l)\times m}$$

The ctx becomes:

$$\vec{c} = A \cdot \vec{r} + (\vec{0}, \vec{\mu} \cdot \lfloor q/2 \rfloor) \in \mathbb{Z}_q^{n+l}$$

Made with Goodnotes

How to build signatures from lattices?

1) Lattice trapdoors; similar to RSA FDH.

2) Fiat-Shamir.

Let's explore a bit of both, starting with 2). Initial idea:

$$A \cdot \vec{x} = \vec{u} \in \mathbb{Z}_q^n$$

Alice		Bob
$sk = \vec{x} \leftarrow \{0,1\}^m$		$pk = (A, \vec{u})$
$\vec{y} \leftarrow \mathbb{Z}_q^m$	$\xrightarrow{\ \vec{z} = A \cdot \vec{y}\ }$	
	$\xleftarrow{\ \beta\ }$	$\beta \leftarrow \mathbb{Z}_q$

$\underline{\vec{r} = \beta \cdot \vec{x} + \vec{y}}$) Check:

$$A \cdot \vec{r} = \beta \cdot A \cdot \vec{x} + A \cdot \vec{y}$$

$= \vec{\alpha} + \beta \cdot \vec{u}$ and $\vec{r}$ is short.

Not hard to see that:

— HVZK: Sim(pk) samples $\beta \leftarrow \mathbb{Z}_q$ $\vec{r} \leftarrow \mathbb{Z}_q^m$ and outputs $\vec{\alpha} = A \cdot \vec{r} - \beta \cdot \vec{u}$ along with β and $\vec{r}$.

Made with Goodnotes

– SS: Given $(\vec{\alpha}, \beta, \vec{\gamma}), (\vec{\alpha}, \beta', \vec{\gamma}')$

we get $A(\vec{\gamma} - \vec{\gamma}') = (\beta - \beta') \cdot \vec{u}$

$$\Rightarrow \vec{x} = (\vec{\gamma} - \vec{\gamma}') \cdot (\beta - \beta')^{-1}$$

Problem: $\vec{x}$ is not short.

<u>Task 2</u>:

<u>Alice</u>		<u>Bob</u>
$\vec{x} = sk$	$\xrightarrow{\vec{\alpha} = A \cdot \vec{y}}$	$pk = (A, \vec{u})$
$\vec{y} \leftarrow \{0,1\}^m$	$\xleftarrow{\beta}$	$\beta \leftarrow \{0,1\}$

$$\vec{\gamma} = \beta \cdot \vec{x} + \vec{y} \longrightarrow A \cdot \vec{\gamma} = \beta \cdot \vec{u} + \vec{\alpha}$$

and $\vec{\gamma}$ "short".

From SS, we can ensure the prover knows $\vec{x} = (\vec{\gamma} - \vec{\gamma}') \cdot (\beta - \beta')^{-1}$

$\in \{-2, -1, 0, 1, 2\}$

But this breaks HVZK! In fact, if $\beta = 1$ then $\beta \vec{x} = \vec{x}$ and $\vec{\gamma} = \vec{x} + \vec{y}$. For instance, if $\vec{\gamma}$

has an entry equal to '2', then the entry of $\vec{x}$ is '1'. Similarly, if $\vec{y}$ has an entry '0', then $\vec{x}$ has an entry '0'.

Formal protocol:

Alice		Bob
	$A \cdot \vec{x} = \vec{u}$	
$sk = \vec{x} \in \{0,1\}^m$		$pk = (A, \vec{u})$
	$\xrightarrow{\alpha = A \cdot \vec{y}}$	
$\vec{y} \leftarrow \{0, \ldots, 5m-1\}^m$	$\xleftarrow{\beta}$	$\beta \leftarrow \{0,1\}$

If $\exists\, i \in \underline{\vec{\gamma} = \beta \cdot \vec{x} + \vec{y}}$,

$\gamma_i \in \{q, 5m\}$ ABORT

and restart.

Basically, each time there is a constant prob. of not ABORTING. For security, we also must repeat the protocol in // for

$K = 128$ times.

Let's also say something about trapdoors. The idea here is to generate a lattice A along with "trapdoor" R. The trapdoor allows to solve LWE/SIS w.r.t. A.

Then, we can:

$\text{Sign}(sk, \mu)$: Output short $\vec{x}$ s.t. $A \cdot \vec{x} = RO(\mu)$ $= \vec{u}$ using trapdoor R.

$\text{Vrfy}(\mu, \sigma)$: Check $A \cdot \vec{x} = RO(\mu)$ and $\vec{x}$ "short".

$\sigma = \vec{x}$

The proof of UFCMA in the ROM is very similar to the proof we did for RSA.

Here is something about generating trapdoors.

The first observation is the fact that for some structures, LWE/SIS are easy. Let:

$$\vec{g} = (1 \;\; 2 \;\; 4 \;\; \cdots \;\; 2^{\ell-1}) \in \mathbb{Z}_q^{\ell}$$

$\ell = \lceil \log_2 q \rceil$.

- We can find short $\vec{x} \in \mathbb{Z}^m$ s.t.

$$\langle \vec{g}, \vec{x} \rangle = \vec{g}^t \cdot \vec{x} = u \mod q.$$

Basically, take $\vec{x}$ to be the base 2 representation of the element u. Thus solves SIS.

- Assume $q = 2^{\ell}$; given $\vec{b}^t \approx s \cdot \vec{g}^t \bmod q$. ($s \in \mathbb{Z}_q$)

Then we can recover $msb(s)$ to be

$$b_\ell \approx s \cdot 2^{\ell-1} = msb(s) \cdot q/2 .$$

Similarly, we can recover $b_{\ell-1} \approx s \cdot 2^{\ell-2}$ and so on ...

We can also extend it to multi-column sound:

$$G = I_n \otimes \vec{g}^t = \mathrm{diag}(\vec{g}^t, \ldots, \vec{g}^t)$$

proceed as before for each coordinate of

$$\vec{s} \in \mathbb{Z}_q^n \quad \text{or}$$

$$G = \begin{array}{c} \vec{u} \\ \begin{pmatrix} 1 \cdot \vec{g}^t & 0 \cdot \vec{g}^t & \cdots & 0 \cdot \vec{g}^t \\ 0 \cdot \vec{g}^t & 1 \cdot \vec{g}^t & & \emptyset \\ & \emptyset & \ddots & \\ & & & 1 \cdot \vec{g}^t \end{pmatrix} \end{array}$$

$$= \begin{pmatrix} 1 & 2 & \cdots & 2^{\ell-1} & 0 & \cdots & & & 0 \\ 0 & \cdots & & & 1 & \cdots & 2^{\ell-1} & & 0 \\ & & & & & & & \ddots & \end{pmatrix}$$

Also easy to see that for any invertible H, LWE/SIS are still easy w.r.t. $H \cdot G \in \mathbb{Z}_q^{n \times n\ell}$

Because : $(H \cdot G) \cdot \vec{x} = \vec{u}$

$$\Leftrightarrow G \cdot \vec{x} = H^{-1} \cdot \vec{u}$$

DEF A trapdoor for $A \in \mathbb{Z}_q^{n \times m}$ is any short matrix $R \in \mathbb{Z}^{m \times n\ell}$ s.t.

$$A \cdot R = H G \mod q$$

for some invertible $H \in \mathbb{Z}_q^{n \times n}$.

Fact: We can generate A along with R and H. Then:

- **SIS**: given $\vec{u} \in \mathbb{Z}_q^n$ then we'll output $\vec{x} = R \cdot \vec{w}$ and

$$\vec{w} = G^{-1}(H^{-1} \cdot \vec{u})$$

G^{-1}: is the function that maps $\mathbb{Z}_q^n$ into its binary decomposition. Namely, $G \cdot G^{-1}(\vec{u}) = \bar{u} \mod q$.

- **LWE**: given $\vec{b}^t \approx \vec{s}^t \cdot A$ then we transform it to $\vec{b}^t \cdot R \approx \vec{s}^t \cdot A \cdot R$

Made with Goodnotes

$$= \vec{s}^{\,t} \cdot H \cdot G \mod q$$