Introduction to modern lattice-based cryptography (Part II)

Damien Stehlé

LIP – CNRS/ENSL/INRIA/UCBL/U. Lyon

Singapore, June 2010

Plan

1- Background on Euclidean lattices.

2- The SIS problem, or how to hash.

3- **The LWE problem, or how to encrypt.**

4- Cryptanalysis.

5- Advanced topics: IBE and FHE.

The LWE problem

a- **Non structured LWE.**

b- Structured LWE.

c- Encrypting with LWE.

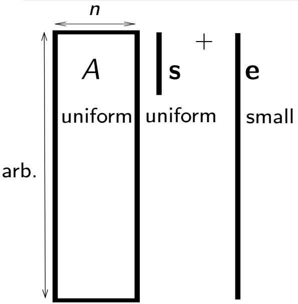
$\mathrm{LWE}_{\alpha,q}$ [Regev'05]

Let $\mathbf{s} \in \mathbb{Z}_q^n$. Let $\Sigma_{\mathbf{s},\alpha}$ be the distribution corresponding to:

$$(\mathbf{a}; \langle \mathbf{a}, \mathbf{s} \rangle + e\ [q])\,, \text{ with } \mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n),\ e \hookleftarrow \nu_{\alpha q} \text{ (small Gaussian).}$$

The Learning With Errors Problem — Comp-LWE_α

Let $\mathbf{s} \in \mathbb{Z}_q^n$. Given arbitrarily many samples from $\Sigma_{\mathbf{s},\alpha}$, find $\mathbf{s}$.



Many interpretations:

- Learning problem, like LPN (over $\mathbb{Z}_2$).
- Approximate linear algebra.
- Closest codeword problem.
- Lattice problem ...

LWE as a one-way function

- OWF: easy to evaluate and hard to invert.
- LWE's OWF: $\mathbf{s} \in \mathbb{Z}_q^n \mapsto A\mathbf{s} + \mathbf{e}\ [q]$.

A one-way function with trapdoor.

- Generate A together with T_A.
- $T_A \cdot (A\mathbf{s} + \mathbf{e}) = T_A \cdot \mathbf{e}\ [q]$.
- Both T_A and $\mathbf{e}$ are small $\Rightarrow$ we know $T_A \cdot \mathbf{e}$ over $\mathbb{Z}$.
  We recover $\mathbf{e}$ and then $\mathbf{s}$ by linear algebra.
- Sufficient condition:

$$\frac{q}{2} > \sqrt{n}\alpha q \cdot \max \|\mathbf{t}_i\| \ \Leftarrow\ n^{1.5}\alpha = \widetilde{o}(1).$$

LWE as a lattice problem

Comp-LWE$_\alpha$

Let $\mathbf{s} \in \mathbb{Z}_q^n$. Given $(A; A\mathbf{s} + \mathbf{e}\ [q])$ with $A \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and $\mathbf{e} \hookleftarrow \nu_{\alpha q}^m$ for arbitrary m, find $\mathbf{s}$.

Let $L_A = \{\mathbf{b} \in \mathbb{Z}^m : \exists \mathbf{x} \in \mathbb{Z}_q^n,\ \mathbf{b} = A\mathbf{x}\ [q]\}$.

- L_A is an m-dimensional lattice and $\widehat{L_A} = \frac{1}{q} A^\perp$.
- BDD$_{\alpha,q}$ (bounded distance decoding):
  Take $A \hookleftarrow U(\mathbb{Z}_q^{m \times n})$, $\mathbf{e} \hookleftarrow \nu_{\alpha q}^m$ and $\mathbf{b} \in L_A$ arbitrary. Given A and $\mathbf{b} + \mathbf{e}$, find $\mathbf{b}$.
- If we can solve LWE, then we can solve BDD.

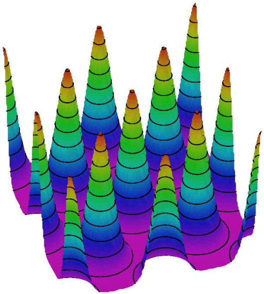
How hard is LWE?

Quantum worst-case to average-case reduction $(\gamma \approx n/\alpha)$

Any efficient LWE algorithm succeeding with non-negligible probability leads to an efficient **quantum** SIVP algorithm.

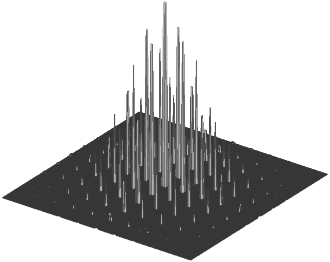
- Efficient quantum computers make LWE more secure!
- [Peikert'09] de-quantumized the reduction, for large q.
- [SSTX'09]: simpler (but weaker) quantum reduction.

How hard is $\mathrm{BDD}_{\alpha,q}$? Rough intuition.



$L \longrightarrow \hat{L}$

Fourier transform



- The Fourier transform of the distribution is implemented with the quantum Fourier transform.
- The input quantum state is built with the LWE oracle.
- The measurement gives a small SIS solution.

Decisional LWE

$$\Sigma_{\mathbf{s},\alpha}: \quad (\mathbf{a}; \langle \mathbf{a}, \mathbf{s} \rangle + e\ [q]) \quad \text{with} \quad \mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n),\ e \hookleftarrow \nu_{\alpha q}.$$

Comp-LWE$_\alpha$

Let $\mathbf{s} \in \mathbb{Z}_q^n$. Given arbitrarily many samples from $\Sigma_{\mathbf{s},\alpha}$, find $\mathbf{s}$.

Dec-LWE$_\alpha$

Let $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$. Distinguish between (arbitrarily many) samples from $\Sigma_{\mathbf{s},\alpha}$ or from $U(\mathbb{Z}_q^2)$.

Dec-LWE and Comp-LWE efficiently reduce to each other.

The LWE problem

a- Non structured LWE.

b- **Structured LWE.**

c- Encrypting with LWE.

Ideal LWE

Let $R_q = \mathbb{Z}_q[x]/(x^n+1)$ with $n = 2^k$ and q prime.

Let $\Psi_{\leq \alpha q}$ be the set of ellipsoidal Gaussians $(\nu_{r_i})_i$ in $\mathbb{R}^n$, where each component has standard deviation $r_i \leq \alpha q$.

For $\psi \in \Psi_{\leq \alpha q}$ and $s \in R_q$, we define:
$\Sigma^{Id}_{s,\psi}$: $(a; as + \mathbf{e}\ [q])$ with $a \hookleftarrow U(R_q)$, $\mathbf{e} \hookleftarrow \psi$.

Comp-Id-LWE$_\alpha$

Let $s \in R_q$ and $\psi \in \Psi_{\leq \alpha q}$. Given arbitrarily many samples from $\Sigma^{Id}_{s,\psi}$, find s.

- One sample from Σ^{Id} encodes n samples from Σ.
- But it costs about the same as 1 sample from Σ:
  We use R_q to multiply vectors, with FFT!
- Same matrix interpretation, but with negacylic blocks.

Ideal LWE is hard

$$\Sigma_{s,\psi}^{Id}: \quad (a; as + \mathbf{e}\ [q]) \quad \text{with} \quad a \hookleftarrow U(R_q),\ \mathbf{e} \hookleftarrow \psi.$$

Comp-Id-LWE$_\alpha$

Let $s \in R_q$ and $\psi \in \Psi_{\leq \alpha q}$. Given arbitrarily many samples from $\Sigma_{s,\psi}^{Id}$, find s.

Any efficient **Id-LWE** algo. succeeding with non-negligible probability leads to an efficient quantum **Id-SIVP** algo.

A faster trapdoor one-way function

- Evaluation cost: $\widetilde{O}(n^2) \Rightarrow \widetilde{O}(n)$ bit operations.
- For the inversion, use the structured T_A from Id-SIS.
- $T_A \cdot (A\mathbf{s} + \mathbf{e}) = T_A\mathbf{e}$ over the integers.
  Multiply by T_A^{-1} to recover $\mathbf{e}$, and then $\mathbf{s}$.
- Evaluation/inversion cost: $\widetilde{O}(n^2) \Rightarrow \widetilde{O}(n)$ bit operations.

Decisional Ideal LWE

$$\Sigma^{Id}_{s,\psi}: \quad (a; as + \mathbf{e}\ [q]) \quad \text{with} \quad a \hookleftarrow U(R_q),\ \mathbf{e} \hookleftarrow \psi.$$

Comp-Id-LWE$_\alpha$

Let $s \in R_q$ and $\psi \in \Psi_{\leq \alpha q}$. Given arbitrarily many samples from $\Sigma^{Id}_{s,\psi}$, find s.

Dec-Id-LWE$_\alpha$

Let $s \hookleftarrow U(R_q)$ and $\psi \in \Psi_{\leq \alpha q}$, choosing the st. devs. from an exponential variate. Distinguish between (arbitrarily many) samples from $\Sigma^{Id}_{s,\psi}$ or from $U(R_q^2)$.

If $x^n + 1$ has n factors modulo q, then Dec-Id-LWE and Comp-Id-LWE efficiently reduce to each other.

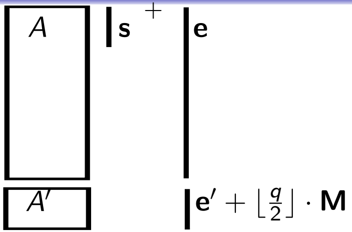
The LWE problem

a- Non structured LWE.

b- Structured LWE.

c- Encrypting with LWE.

Encrypting with LWE



- Public key: $A \in \mathbb{Z}_q^{m \times n}$, $A' \in \mathbb{Z}_q^{n \times n}$; secret key: T_A.
- Encryption: compute $[A\mathbf{s} + \mathbf{e}; A'\mathbf{s} + \mathbf{e}' + \lfloor \frac{q}{2} \rfloor \cdot \mathbf{M}]$.
- Decryption: recover $\mathbf{s}$ from the first part of the ciphertext, using T_A; compute $A'\mathbf{s}$ to obtain $\mathbf{e}' + \lfloor \frac{q}{2} \rfloor \mathbf{M}$; round to the closest multiple of $\lfloor \frac{q}{2} \rfloor$ to recover $\mathbf{M}$.

Any semantic attack leads to an algorithm for Dec-LWE.

Encrypting with Id-LWE

We could do the same ... but there is much better.

$\Sigma^{Id}_{s,\psi}$: $(a; as + e\ [q])$ with $a \hookleftarrow U(R_q)$, $e \hookleftarrow \psi$.

Let $s \hookleftarrow U(R_q)$ and ψ "small". Distinguishing between samples from $\Sigma^{Id}_{s,\psi}$ or from $U(R_q^2)$ is computationally infeasible.

Simplification: We can also take s small.
The transformation $(a_i, b_i) \hookrightarrow (a_i, b_i - a_1^{-1} b_1)$ maps:

$$U(R_q^2) \text{ to } U(R_q^2) \quad \text{and} \quad \Sigma^{Id}_{U(R_q),\psi} \text{ to } \Sigma^{Id}_{\psi,\psi}.$$

Encrypting with Id-LWE

$\Sigma_{s,\psi}^{Id}$: $(a; as + e\ [q])$ with $a \hookleftarrow U(R_q)$, $e \hookleftarrow \psi$.

Let s and ψ "small". Distinguishing between samples from $\Sigma_{s,\psi}^{Id}$ or from $U(R_q^2)$ is computationally infeasible.

- Secret key: s (small); Public key: a_1, $a_2 = a_1 s + e$.
- Encryption: $(c_1, c_2) = (a_1 t + e_1, a_2 t + e_2 + \lfloor \frac{q}{2} \rfloor M)$, with t random and small.
- Decryption: $c_2 - c_1 s$ is "small $+\lfloor \frac{q}{2} \rfloor M$".

- CPA-secure assuming the hardness of Dec-Id-LWE.
- Key-sizes are quasi-optimal.
- Complexity and ciphertext expansion are quasi-optimal.

This is ElGamal!!!

- Secret key: s (small); Public key: a_1, $a_2 = a_1 s + e$.
- Encryption: $(c_1, c_2) = (a_1 t + e_1, a_2 t + e_2 + \lfloor \frac{q}{2} \rfloor M)$, with t random and small.
- Decryption: $c_2 - c_1 s$ is "small $+ \lfloor \frac{q}{2} \rfloor M$".

- Secret key: s; Public key: g_1, $g_2 = g_1^s$.
- Encryption: $(c_1, c_2) = (g_1^t, g_2^t M)$, with t random.
- Decryption: c_2 / c_1^s is M.

Plan

1- Background on Euclidean lattices.

2- The SIS problem, or how to hash.

3- The LWE problem, or how to encrypt.

4- **Cryptanalysis.**

5- Advanced topics: IBE and FHE.

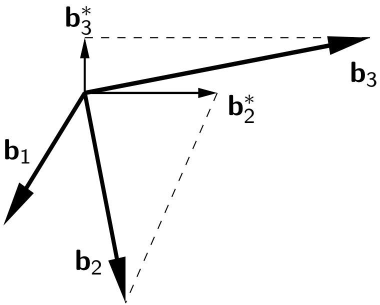
Attacking SIS/Id-SIS/LWE/Id-LWE

- The only known attack consists in finding a small vector/basis of the lattice $A^{\perp} = \{\mathbf{s} \in \mathbb{Z}^{mn} : \mathbf{s}A = \mathbf{0}\ [q]\}$.
- Generalized birthday attack: may be feasible if m is large. Its cost is easily determined [MR'09].
- Lattice reduction: may be applied to a subset of the rows (trade-off between approximation factor and existence of short vectors).

But... although quite old (Lagrange, Gauss, Hermite, Minkowski, etc)... lattice reduction is not so well understood.

Lattice reduction

- Principle: start from an arbitrary basis of the lattice, and progressively improve it.
- Quality of a basis: measured by the Gram-Schmidt Orth.



- $\mathbf{b}_i^* = \text{argmin}\|\mathbf{b}_i + \sum_{j<i} \mathbb{R}\mathbf{b}_j\|$
- Quality measure: $(\|\mathbf{b}_i^*\|)_{i=1..n}$.

Why?

- The slower the $\|\mathbf{b}_i^*\|$'s decrease, the more orthogonal.
- Their product is constant.
- If they decrease slowly, then $\mathbf{b}_1$ must be small.

LLL

Size-reduction: $|\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle| \leq \|\mathbf{b}_j^*\|^2/2$, for all $j < i$.

Ensures that $\max \|\mathbf{b}_i\| \leq \sqrt{n} \cdot \max \|\mathbf{b}_i^*\|$.

Lenstra-Lenstra-Lovász reduction

A basis $(\mathbf{b}_i)_i$ is LLL-reduced if it is size-reduced and $\|\mathbf{b}_{i+1}^*\| \geq \|\mathbf{b}_i^*\|/2$ for all i (Lovász' condition).

LLL algorithm: size-reduce; if any, take an i violating Lovász' condition, swap vectors i and $i+1$, and restart (else, stop).

The LLL algorithm runs in polynomial time, and the first output vector satisfies $\|\mathbf{b}_1\| \leq 2^n \cdot \lambda(L)$.

HKZ

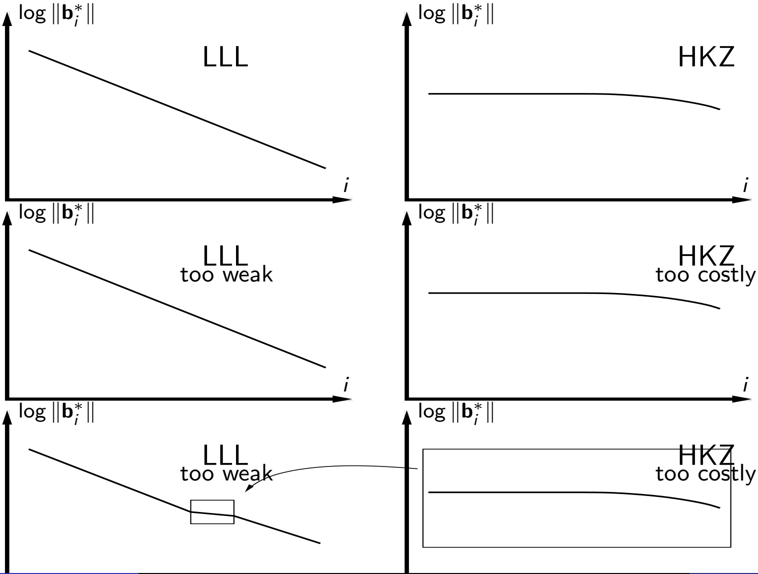
Hermite-Korkine-Zolotarev reduction

A basis $(\mathbf{b}_i)_i$ is HKZ-reduced if it is size-reduced, if $\|\mathbf{b}_1\| = \lambda(L)$ and if after projection orthogonally to $\mathbf{b}_1$, the basis $(\mathbf{b}_i)_{i>1}$ is HKZ-reduced.

HKZ-reduction is polynomial-time equivalent to solving SVP.
Best algorithms:

- Kannan: deterministic, polynomial space, time $n^{O(n)}$.
- Ajtai et al: probabilistic, time and space $2^{O(n)}$.
- Micciancio-Voulgaris: deterministic, time and space $2^{O(n)}$.

BKZ: a trade-off between LLL and HKZ



Schnorr's hierarchy

Lattice reduction rule of the thumb

For block-size k, reduction algorithms can achieve $\|\mathbf{b}_1\| \approx n^{O(n/k)} \cdot \lambda_1$ in time $\mathcal{P}oly(n) \cdot 2^{O(k)}$.

For SIS, this gives the hardness condition $m^{O(m/k)} \gg \beta$.

- Seems satisfied by BKZ for small block-sizes.
- But the cost unexpectedly blows up with block-size ≈ 30.

Warnings

- The runtime of BKZ is not $\mathcal{P}oly(n) \cdot 2^{O(k)}$.
- BKZ is the only available variant of Schnorr's hierarchy.

Solving SVP in practice

Practical boundaries for solving SVP are still being improved.

- The Kannan-Fincke-Pohst enumeration is currently the most practical algorithm.
- Tree pruning, parallelisation, hardware implementation, ...

- In 2005, dimension 50?
- In 2007, dimension 70.
- In 2009, dimension 80.
- Now (Gama et al.'10), dimensions 110-120!

Plan

1- Background on Euclidean lattices.

2- The SIS problem, or how to hash.

3- The LWE problem, or how to encrypt.

4- Cryptanalysis.

5- **Advanced topics: IBE and FHE.**

Advanced topics

a- **Identity-based encryption.**

b- Fully homomorphic encryption.

(H)IBE

- Identity-based encryption: encryption infrastructure in which **a user's public key is uniquely determined by its identity**; the user's private key is computed by a trusted authority, using a master key.
  $\Rightarrow$ No need for a public key distribution infrastructure.
- Question first raised by Shamir in 1984.
- First realization by Boneh and Franklin in 2001, using bilinear pairings on elliptic curves.
- Hierarchical IBE: same as IBE, but each entity in level k of a hierarchy can generate the private keys of all entities of lower levels in the hierarchy.

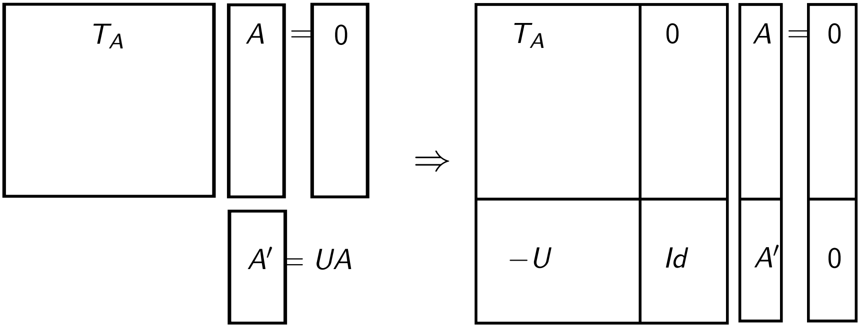
HIBE using LWE

- Encode an identity *id* as a string of bits of length $\leq k$.
- An identity *id* is higher in the hierarchy than id' if *id* is a prefix of id': $id' = (id \| \cdot)$.
- The master has identity $\{\}$.

- Sample A uniform in $\mathbb{Z}_q^{m \times n}$ together with a trapdoor T_A. These are the master's keys.
- Sample $(A_1^0, A_1^1), \ldots, (A_k^0, A_k^1)$ iid uniformly in $\mathbb{Z}_q^{m \times n}$.
- User $id = i_1 \ldots i_\ell$ has public key A_{id}, the vertical concatenation of $A, A_1^{i_1}, \ldots, A_\ell^{i_\ell}$.
- sk_{id} is a short basis of $A_{id}^\perp$.
- Encryption: same as with LWE.

Private key extraction

- Suppose $id' = (id\|\cdot)$. How does user id extract a private key for id' from his/her own private key?
- How to obtain a $T_{A_{id}}$ from a $T_{A_{id'}}$?
- Writing the new rows as combinations of the previous ones suffices to obtain a basis of $A_{id'}^{\perp}$ with small GSO.



Private key randomization

- But now $id' = (id\|\cdot)$ now knows the private key of id!
- id should randomize $T_{A_{id'}}$ before giving it to id'.
- Use the previous basis of $A_{id'}^{\perp}$ with small GSO to sample from $D_{A_{id'}^{\perp},\sigma}$ for a small σ.
- With sufficiently many samples, we obtain a full rank set of short vectors in $A_{id'}^{\perp}$.
- Convert it into a short basis.
- The output distribution is independent of the initial basis.

Cash et al, Eurocrypt'10

Assuming LWE is hard, this scheme is secure against selective-identity chosen plaintext attacks, in the standard model.

More on IBE

Similar techniques lead to signatures that are secure in the standard model (without the random oracle).

Very hot topic:

- Cash-Hofheinz-Kiltz-Peikert at Eurocrypt'10.
- Agrawal-Boneh-Boyen at Eurocrypt'10.
- Boyen at PKC'10.
- Agrawal-Boneh-Boyen at Crypto'10.

Main open problems:

- Improving the efficiency (e.g., using Id-LWE?).
- The SVP approximation factor increases quickly with the number of levels in the hierarchy: $\gamma = n^{O(k)}$.
  Can we avoid this?

Recent developments

a- Identity-based encryption.

b- **Fully homomorphic encryption.**

Homomorphic encryption

- Given $C_1 = \mathcal{E}(M_1)$ and $C_2 = \mathcal{E}(M_2)$, can we compute $\mathcal{E}(f(M_1, M_2))$ for some/any f, without decrypting?
- E.g., for textbook RSA: $M_1^e \cdot M_2^e = (M_1 \cdot M_2)^e \ [N]$.
- An encryption scheme is fully homomorphic if any function (given as a circuit) of any number of M_i's can be evaluated in the ciphertext domain:

$$\forall k, \forall f, \exists g : \mathcal{D}[g(\mathcal{E}(M_1), \ldots, \mathcal{E}(M_k))] = f(M_1, \ldots, M_k).$$

- The bit-size of the output of g must be independent of the circuit size of f.

The 'holy grail' of cryptography

- The question was first asked by Rivest, Adleman and Dertouzous in 1978.
- Solved by Craig Gentry in 2009, using ideal lattices.

IBM announcement (25/06/09): `An IBM Researcher has solved a thorny mathematical problem that has confounded scientists since the invention of public-key encryption several decades ago. The breakthrough, called "privacy homomorphism," or "fully homomorphic encryption," makes possible the deep and unlimited analysis of encrypted information [...] without sacrificing confidentiality.`

Many applications:

- Use untrusted parties to run programs (cloud computing).
- Search over private data (PIR), etc.

A somewhat homomorphic scheme

- Sample a good basis B_J^{sk} of an ideal lattice J:
  e.g., each basis vector has norm $\leq \mathcal{P}oly(\lambda) \cdot \lambda_1(J)$.
- Let B_J^{pk} be a bad basis of B_J^{sk} (e.g., its HNF).
- To encrypt $\pi \in \{0,1\}$, take a small random $\rho \in \mathbb{Z}[x]/(x^n+1)$ and output
$$\psi = \pi + 2\rho \bmod B_J^{pk}.$$
- Plaintext space: $\{0,1\}$, ciphertext space: R/J.
- Use Babai's rounding-off to decrypt:
$$\psi - B_J^{sk} \lfloor (B_J^{sk})^{-1} \psi \rceil \;\Rightarrow\; \pi + 2\,\rho.$$

Correctness and security

- Babai's rounding-off is correct as long as the distance to J is $\leq \frac{\lambda_1(J)}{\mathcal{P}oly(n)} =: r_{Dec}$.
- Correctness: it suffices that

$$r_{Enc} := \max_{\pi,\rho} \|\pi + 2\rho\| \leq 1 + 2 \max_{\rho} \|\rho\| \leq r_{Dec}.$$

- Security: Finding a closest vector for a target within r_{Enc} of J must be hard (BDD).
- With lattice reduction, this can be done in time $\approx 2^k$ if $r_{Enc} \leq 2^{n/k} \cdot r_{Dec}$.

More on security

If J and B_J^{sk} are well chosen, if $\pi \in \{0,1\}$ and if ρ is sampled from some discrete Gaussian, then this scheme can be made CPA secure under the assumption that Id-SVP$_\gamma$ is hard to solve for quantum polynomial-time algorithms, for some small γ.

The proof includes a dimension-preserving worst-case to average-case reduction. The distribution for J is the uniform distribution over the set of ideals with norm in $[a, 2a]$.

Why is it (somewhat) homomorphic?

- To encrypt $\pi \in \{0,1\}$, take a small random $\rho \in R$ and output $\psi = \pi + 2\rho \bmod B_J^{pk}$.
- $\psi_i = \pi_i + 2\rho_i \bmod B_J^{pk}$ for $i \in \{1,2\}$ implies, mod J:

$$\begin{aligned} \psi_1 + \psi_2 &= (\pi_1 + \pi_2) + 2(\rho_1 + \rho_2), \\ \psi_1 \times \psi_2 &= (\pi_1 \times \pi_2) + 2(\rho_1 \times \pi_2 + \rho_2 \times \pi_1 + 2\rho_1 \times \rho_2). \end{aligned}$$

- Add/Mult modulo B_J^{pk} on ciphertexts homomorphically performs Add/Mult modulo 2 on plaintexts.
- If we want to apply a mod-2 circuit to plaintexts, we replace it by an integer circuit, that we apply to ciphertexts modulo B_J.

Why is it only "somewhat" homomorphic?

The more operations are applied the further away from J.

- $\mathrm{dist}(\mathbf{C}_1 + \mathbf{C}_2, J) \leq \mathrm{dist}(\mathbf{C}_1, J) + \mathrm{dist}(\mathbf{C}_2, J)$.
- $\mathrm{dist}(\mathbf{C}_1 \times \mathbf{C}_2, J) \leq K \cdot \mathrm{dist}(\mathbf{C}_1, J) \cdot \mathrm{dist}(\mathbf{C}_2, J)$, for some K.

Let C be a mod 2 circuit with a corresponding integer circuit that evaluates $h(x_1, \ldots, x_t)$ of (total) degree d. Then C is permitted if $tK^d r_{Enc}^d \leq r_{Dec}$. Equivalently:

$$d \leq \frac{\log r_{Dec}}{\log(r_{Enc} \cdot K \cdot t)}.$$

Making the scheme fully homormophic

- If many operations have been applied and the ciphertext ψ corresponding to plaintext π is deemed too noisy, we try to "refresh" it.
- But we cannot decrypt using the secret key sk_1.
- Trick: encode ψ further using a second public key pk_2, and decode homomorphically using $\mathcal{E}_{pk_2}(sk_1)$.

$$\mathcal{D}_{sk_2}\left(Dec(\mathcal{E}_{pk_2}(\psi), \mathcal{E}_{pk_2}(sk_1))\right) = Dec(\psi_1, sk_1) = \pi.$$

- Refreshing as many times as required, we can apply any circuit privately.

The decryption circuit

- Problem: Is the decryption circuit simple enough so that it can be itself be applied without refreshing?
- Decryption: $\psi - B_J^{sk} \lfloor (B_J^{sk})^{-1} \psi \rceil$ provides $\pi + 2\rho$.
- This seems too complicated.
- We need to "squash" the decryption circuit.

Outline of Gentry's solution:

- There exists $\mathbf{v}_J^{sk}$ with: $\forall \psi : B_J^{sk} \lfloor (B_J^{sk})^{-1} \psi \rceil = \lfloor \mathbf{v}_J^{sk} \psi \rceil$.
- Generate random public $\mathbf{v}_i$'s with a secret sparse subset S which sums to $\mathbf{v}_J^{sk}$: $\sum_{i \in S} \mathbf{v}_i = \mathbf{v}_J^{sk}$.
- The $\mathbf{v}_i \cdot \psi$'s can be computed publicly, and then the decryption reduces to summing up the few relevant ones.

More on FHE

Overall, Gentry gets FHE based on two security assumptions: SVP/BDD over ideal lattices and Sparse Subset Sum Problem.

Very hot topic:

- Gentry, STOC'09 and CRYPTO'10.
- van Dijk-Gentry-Halevi, Eurocrypt'10.
- Smart-Vercauteren, PKC'10.
- S.-Steinfeld, IACR eprint: "ciphertext refreshing" costs $\widetilde{O}(k^3)$ bit operations, for security 2^k.

Open problems:

- Improving the efficiency further, in theory and practice.
- Removing the SSSP hardness assumption.

Plan

1- Background on Euclidean lattices.

2- The SIS problem, or how to hash.

3- The LWE problem, or how to encrypt.

4- Cryptanalysis.

5- Advanced topics: IBE and FHE.

Conclusion

- The schemes are becoming more and more efficient, in particular thanks to structured matrices / ideal lattices.
- More and more primitives can be built from lattice problems.
- The best attacks are becoming better understood.

- But still not many schemes are implemented.
- Lattice reduction can probably still be improved.
- Mainly one library used for crytanalysis (Shoup's NTL), and it is known to behave oddly [GN'08].

Open problems

- NTRU remains faster than the provable schemes.
  Can we prove its security?
- Can we improve the efficiency of the lattice-based primitives, e.g., signature in the standard model, (H)IBE, FHE, CCA-secure encryption, etc?
- What is the practicality of all these schemes?
- What are the actual limits of lattice reduction?

More open problems

- Can quantum computers improve lattice algorithms?
- Are ideal lattices weaker than general lattices?
- Are there better algorithms than lattice reduction for SVP_γ with $\gamma = \mathcal{P}oly(n)$?
- Can we use lattice algorithms to factor integers or compute discrete logarithms?
- Which other primitives can be built from lattice problems? Can we do all those using discrete log and pairings?
- Can we adapt (some of) the techniques to linear codes?