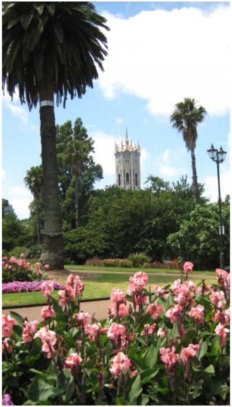
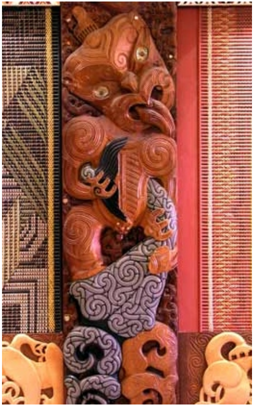
Computational problems in lattices, and public key signatures

Steven Galbraith





University of Auckland, New Zealand

Plan

- Learning with errors
- Computational problems in lattices
- Public key encryption and homomorphic encryption from LWE
- Hazards and challenges
- Public key signatures

Please ask questions at any time.

Learning with Errors (LWE)

Oded Regev (2005)

- Let q be an odd prime and $n, m \in \mathbb{N}$. [Example: $n = 320$, $m = 2000$, $q = 4093$.]
- Let $\underline{s} \in \mathbb{Z}_q^n$ be secret (**column** vector).
- Suppose one is given an $m \times n$ matrix A chosen uniformly at random with entries in $\mathbb{Z}_q$ and a length m vector

  $$\underline{b} \equiv A\underline{s} + \underline{e} \pmod{q}$$

  where the vector $\underline{e}$ has entries chosen independently from a "discrete normal distribution" on $\mathbb{Z}$ with mean 0 and standard deviation $\sigma = \alpha q$ for some $0 < \alpha < 1$ (e.g., $\sigma = 3$).
- The LWE problem is to find the vector $\underline{s}$.
- Decisional-LWE: Distinguish pairs $(A, \underline{b})$ as above from uniformly chosen pairs $(A, \underline{b})$.

Discrete Gaussians

- The Gaussian distribution (= normal distribution) on $\mathbb{R}$ with mean 0 and variance σ^2 has probability density function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)}.$$

- To define the discrete Gaussian on $\mathbb{Z}$ compute

$$M = 1 + 2\sum_{k=1}^{\infty} e^{-k^2/(2\sigma^2)}$$

and define the distribution on $\mathbb{Z}$ by

$$\Pr(x) = \frac{1}{M} e^{-x^2/(2\sigma^2)} \quad \text{for } x \in \mathbb{Z}.$$

- Sampling closely from this distribution in practice is non-trivial!

Remarks on Learning with Errors

- LWE: Given A and $\underline{b} \equiv A\underline{s} + \underline{e} \pmod{q}$ to find $\underline{s}$.
- If $\underline{e} = 0$ then easy.
- The solution $\underline{s}$ is not uniquely determined, but one value $\underline{s}$ is significantly more likely than the others if m is large enough. In other words, for a vector $\underline{s}' \neq \underline{s}$, $\underline{b} - A\underline{s}' \pmod{q}$ is not likely to look like a vector sampled from the discrete Gaussian distribution.
  Hence LWE is well-defined as a maximum likelihood problem.
- There is a reduction from LWE to Decisional-LWE.

Lattices

- Let $\underline{b}_1, \dots, \underline{b}_n$ be linearly independent vectors in $\mathbb{R}^n$.
- The set $L = \{\sum_{i=1}^n x_i \underline{b}_i : x_i \in \mathbb{Z}\}$ is a (full rank) lattice. Call its elements **points** or **vectors**.
- Alternative definition: A discrete subgroup of $\mathbb{R}^n$.
- Everyone working with lattices should declare whether their vectors are **rows** or **columns**. Today I am using **columns**.
- The **basis matrix** is the $n \times n$ matrix B whose columns are the vectors $\underline{b}_1, \dots, \underline{b}_n$.
- A lattice has many different bases.

Computational Problems (Informally)

- Shortest vector problem (SVP): Given a basis matrix B for a lattice L find a non-zero vector $\underline{v} \in L$ such that $\|\underline{v}\|$ is minimal.
  The norm here is usually the standard Euclidean norm in $\mathbb{R}^n$, but it can be any norm such as the ℓ_1 norm or ℓ_∞ norm.
- Closest vector problem (CVP): Given a basis matrix B for a full rank lattice $L \subseteq \mathbb{R}^n$ and an element $\underline{t} \in \mathbb{R}^n$ find $\underline{v} \in L$ such that $\|\underline{v} - \underline{t}\|$ is minimal.

Reducing LWE to CVP

- LWE: Given A and $\underline{b} \in \mathbb{Z}^m$ where $\underline{b} \equiv A\underline{s} + \underline{e} \pmod{q}$, find $\underline{s}$.
- Let $L = \{\underline{v} \in \mathbb{Z}^m : \underline{v} \equiv A\underline{s} \pmod{q} \text{ for } \underline{s} \in \mathbb{Z}^n\}$.
- To solve LWE we want to find a lattice point $\underline{y} \equiv A\underline{s} \pmod{q}$ close to $\underline{b}$. Once we have computed $\underline{y} \in L \subset \mathbb{Z}^m$ one can easily compute $\underline{s} \in \mathbb{Z}^n$ with $\underline{y} \equiv A\underline{s} \pmod{q}$.
- Usually, the desired solution $\underline{s}$ corresponds to the closest lattice point in the Euclidean norm.
- Hence, solve LWE by lattice basis reduction on L followed by Babai nearest plane algorithm or enumeration or randomised variant (see Lindner-Peikert 2011, Liu-Nguyen 2013).
- Optimal to choose $m \approx \sqrt{n \log(q)/\log(\delta)}$.
  ($\delta =$ Hermite factor.)
- Hence typically require $m > n > 300$ for security.
  [Vadim commented that n could be smaller if q is very large.]

SIS problem (Ajtai)

- Let $A \in \mathbb{Z}_q^{m \times n}$ and let $\underline{s} \in \mathbb{Z}_q^m$ be a short vector.
  Let $\underline{b} \equiv A^T \underline{s} \pmod{q}$.
  The (inhomogeneous) SIS problem is: Given $(A^T, \underline{b})$ to find $\underline{s}$.
- One can solve SIS by solving CVP: Find any vector $\underline{y} \in \mathbb{Z}^m$ such that $A^T \underline{y} \equiv \underline{b} \pmod{q}$ and then solve the CVP instance $(L, \underline{y})$ where

$$L = \{\underline{v} \in \mathbb{Z}^m : A^T \underline{v} \equiv 0 \pmod{q}\}.$$

- If $\underline{v}$ is close to $\underline{y}$ then $\underline{s} = \underline{y} - \underline{v}$ is a short vector such that $A^T \underline{s} \equiv \underline{b} \pmod{q}$.

LWE = SIS

- An LWE instance $\underline{b} = A\underline{s} + \underline{e} \pmod{q}$, where $\underline{s}$ is chosen from the error distribution, becomes an $(n+m) \times m$ SIS instance

$$(A|I_m)\binom{\underline{s}}{\underline{e}} \equiv \underline{b} \pmod{q}.$$

- Conversely, given SIS instance $\underline{b} \equiv A^T\underline{s} \pmod{q}$ we can compute column-HNF $A^T U = (A'|I_n)$ to have

$$\underline{b} \equiv A^T U(U^{-1}\underline{s}) \equiv (A'|I_n)\binom{\underline{y}}{\underline{z}} \equiv A'\underline{y} + \underline{z} \pmod{q}.$$

- Micciancio-Mol: (m,n)-SIS $\leq (m-n,n)$-LWE, $(m,m-n)$-SIS $\leq (m,n)$-LWE
- Subtlety: with LWE an attacker can discard rows if it makes the problem easier, but for SIS one needs to be more careful.

Variants of LWE

- We may assume $\underline{s}$ is sampled from the error distribution.
- Can consider fixed number m of LWE samples, or an arbitrary number.
- Binary-LWE: $\underline{s} \in \{0, 1\}^n$ and $\underline{e}$ from error distribution.
- Can choose parameters so that the solution is not well-defined.

Binary-LWE

- $\underline{b} \equiv A\underline{s} + \underline{e} \pmod{q}$ where $\underline{s} \in \{0,1\}^n$ and $\underline{e}$ is from a discrete Gaussian.
- There are recent hardness results on binary-LWE by Micciancio-Peikert and Brakerski-Langlois-Peikert--Regev-Stehlé:
  If certain problems in $n/\log(n)$-dimensional lattices are hard, then binary LWE is hard for $\underline{s} \in \{0,1\}^n$.
- Direct reduction of LWE to CVP does not exploit size of $\underline{s}$.
- Instead, reduce LWE to SIS, then reduce SIS to CVP.
- Challenge: Fully understand binary-LWE.

Public Key Cryptography from LWE (Regev encryption)

- Private key: $\underline{s}$ (column vector).
- Public key: $A, \underline{b} = A\underline{s} + \underline{e} \pmod{q}$, q odd prime.
- To **encrypt** $\mathrm{M} \in \{0, 1\}$:
  - Choose $\underline{u} \in \{0, 1\}^m$ (row vector).
  - Set $\mathrm{c}_1 = \underline{u}A \pmod{q}, \mathrm{c}_2 = \underline{u}\ \underline{b} + \mathrm{M}(q-1)/2 \pmod{q}$.
- To **decrypt**: Compute $v = \mathrm{c}_2 - \mathrm{c}_1\ \underline{s} \pmod{q}$ reduced to the interval $\{-(q-1)/2, \dots, -1, 0, 1, \dots, (q-1)/2\}$.
  If $|v| < q/4$ then output 0, else output 1.
- To break the cryptosystem one could try to compute $\underline{s}$ or $\underline{u}$. Note that c_1 can be viewed as multiple modular subset-sum instances on the same secret $\underline{u}$.

Public Key Cryptography from LWE (Regev encryption)

- Regev shows that the IND-CPA security of the encryption scheme follows from the decisional-LWE assumption.
- There are variants of the scheme that can be applied in the setting of ring-LWE (essentially re-animating the corpse of NTRU).
- Various techniques to improve bandwidth so that a ciphertext encrypts more than one bit (e.g., Lindner-Peikert 2011).

Homomorphic encryption from LWE

- Regev encryption is homomorphic for addition: Given two ciphertexts

  $$\mathrm{c}_{i,1} = \underline{u}_i A \pmod q, \qquad \mathrm{c}_{i,2} = \underline{u}_i\, \underline{b} + \mathrm{M}_i(q-1)/2 \pmod q$$

  for $i \in \{1,2\}$ then

  $$\mathrm{c}_{1,1} + \mathrm{c}_{2,1} = (\underline{u}_1 + \underline{u}_2)A \pmod q$$

  and

  $$\mathrm{c}_{1,2} + \mathrm{c}_{2,2} = (\underline{u}_1 + \underline{u}_2)\underline{b} + (\mathrm{M}_1 + \mathrm{M}_2)(q-1)/2 \pmod q$$

  give an encryption of $\mathrm{M}_1 + \mathrm{M}_2 \pmod 2$.
- Brakerski-Vaikuntanathan showed that a natural "tensor product" operation on ciphertexts $(\mathrm{c}_{1,1}, \mathrm{c}_{1,2})$ and $(\mathrm{c}_{2,1}, \mathrm{c}_{2,2})$, followed by a "key switching" operation provides an encryption of $\mathrm{M}_1\mathrm{M}_2 \pmod 2$.

The official line: Paradise gained

We have a very simple cryptosystem with extremely strong (even worst-case) security guarantees depending on long-studied and hard computational problems.

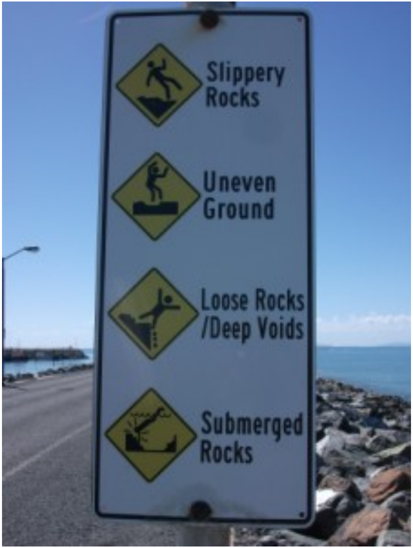
It provides powerful functionality, e.g., homomorphic encryption.

The basic operations are simply vector operations, so everything is easy to implement.

Don't believe the hype!

- These computational problems aren't as well-studied, and sometimes not as hard, as they seem.
- Parameter selection can be non-trivial.
- Worst-case security is not a feature, it's a bug.
- Serious issues about security of these schemes in practical systems.
- The cryptosystems may be hard to implement.

Hazards


Slippery
Rocks
Uneven
Ground
Loose Rocks
/Deep Voids
Submerged
Rocks

Goldilocks problems

- LWE is an example of a “Goldilocks problem”.
  [This was pictured nicely in Vadim's talk with the “tent” graph.]
- If the standard deviation σ is too small compared with q then the CVP instance is not as hard as we'd like.
- If the standard deviation σ is too large compared with q then the problem is not well-defined and it is not necessarily hard to find a vector $\underline{s}$ such that $\underline{b} - A\underline{s} \pmod{q}$ has smallish norm.

Worst-case security is not a feature, its a bug

- All computational problems have easy instances.
- For example:
  - Factoring smooth numbers is easy.
  - CVP is easy if the closest lattice point is inside the parallelepiped centered on the target vector.
- It can be non-trivial to distinguish an easy instance from a hard one.
- Hence, basing security on worst-case instances is a necessity that is a long-standing issue in crypto .
- Compare with RSA: We already choose RSA moduli to be products of two random primes of similar bitlength, since that is heuristically the worst-case instance.
- Our job would be easier if we had computational problems with no easy instances.
- But I agree that it is nice that lattice-based crypto can handle this issue rigorously.

Security under adaptive attacks

- Recall the Regev decryption algorithm: Compute $v = c_2 - c_1 \underline{s} \pmod q$ reduced to the interval $\{-(q-1)/2, \ldots, -1, 0, 1, \ldots, (q-1)/2\}$.
  If $|v| < q/4$ then output 0, else output 1.
- Given a decryption oracle one can call it on $(c_1, c_2) = ((1, 0, \ldots, 0), r)$ and hence learn most significant bit of $(r + \underline{s}_1) \pmod q$.
  It is easy to see that one can **determine the private key** after polynomially many such queries.
- Such attacks can be completely realistic (recall Bleichenbacher's success on attacking standardised variants of RSA).
- There are similar trivial attacks on Gentry/Smart-Vercauteren (Loftus-May-Smart-Vercauteren) and approximate GCD.

Security under adaptive attacks

- Similarly, every fully homomorphic encryption scheme requires certain encryptions of secret values (for example, for the "key switching" technique mentioned earlier).
- Hence, given a decryption oracle, one can determine the private key for every fully homomorphic encryption scheme.
- A good challenge is to obtain IND-CCA1 homomorphic encryption schemes.
  Loftus-May-Smart-Vercauteren have done this for the Smart-Vercauteren scheme.
- Note that Micciancio and Peikert (EUROCRYPT 2012) have given IND-CCA1 secure encryption from LWE. But it is not homomorphic.

Hard to implement

- Many lattice cryptosystems require samples from discrete Gaussians.
- Computing from such distributions, even just on $\mathbb{Z}$, is non-trivial.
- Three basic approaches: rejection sampling, precomputing cumulative probability table (inversion method), or Knuth-Yao method.
- Each has drawbacks: some require enormous precomputed tables, some require floating-point arithmetic, some require many more random bits as input than one would expect.
- Two challenges are to improve sampling algorithms, and to remove/relax the requirements for Gaussians in the protocols.

Comparison with pairing based cryptography

[Here I recall the previous provocative statements and discuss them in the context of pairings.]

- These computational problems aren't as well-studied, and sometimes not as hard, as they seem.
- Parameter selection can be non-trivial.
- Worst-case security is not a feature, its a bug.
- Serious issues about security of these schemes in practice.
- The cryptosystems are hard to implement.

Public key signatures

There are two general approaches to obtain public key signatures:

- Hash and sign.
  - Requires a trapdoor one-way function $f : D \to R$.
    One hashes message to $H(m) \in R$ and the signature is $f^{-1}(H(m)) \in D$.
  - The public key is a description of f and the private key is the trapdoor.
  - Proposed for lattices by GGH, NTRU, GPV, etc.
- Zero-knowledge proofs.
  - Requires a one-way function $f : D \to R$.
  - The public key is $f(d)$ for some $d \in D$.
    The signature is a proof of knowledge of d, using the message m and a hash function as a source of randomness in the protocol (Fiat-Shamir heuristic).
  - Proposed for lattices by various authors, but really got properly started with Lyubashevsky at Asiacrypt 2009.

Public key signatures

- Lyubashevsky has a sequence of papers with co-authors giving good lattice-based public key signature schemes.
- Public key is an LWE instance $(A, \underline{b} = A\underline{s} + \underline{e} \pmod{q})$ with $\underline{s}$ short, where A is $m \times n$ and $m \gg n$.
- Take a three-move proof of knowledge of $(\underline{s}, \underline{e})$ and apply the Fiat-Shamir transform.
- Basic idea: Choose short vectors $\underline{y}_1, \underline{y}_2$, compute $\underline{b}' = A\underline{y}_1 + \underline{y}_2 \pmod{q}$, receive challenge c, compute $\underline{z}_1 = \underline{y}_1 + \underline{s}c$, $\underline{z}_2 = \underline{y}_2 + \underline{e}c$.
  Verifier checks that $\underline{z}_1$ and $\underline{z}_2$ are short, computes

$$A\underline{z}_1 + \underline{z}_2 - \underline{b}c = A\underline{y}_1 + \underline{y}_2 + (A\underline{s} + \underline{e})c - (A\underline{s} + \underline{e})c = \underline{b}'.$$

Schnorr signatures/identification protocol

- Signer/prover has public key $h = g^a$, where g has order r.
- The prover chooses a random integer $0 \leq k < r$, computes $s_0 = g^k$ and sends s_0 to the verifier.
- The verifier sends a "challenge" $1 \leq s_1 < r$ to the prover.
- The prover returns $s_2 = k + as_1 \pmod r$.
- The verifier then checks that $g^{s_2} = s_0 h^{s_1}$.
- It is easy to see that anyone can produce triples (s_0, s_1, s_2) that satisfy the verification equation, without knowing the private key.
  Hence the protocol is "honest verifier zero knowledge".

Lyubashevsky's proof technique

- Use rejection sampling so that the output distribution of signatures is **independent** of the private key.
- Essentially, for the equation $\underline{z}_1 = \underline{y}_1 + \underline{s}c$ we choose the vector $\underline{y}_1$ so that its entries are chosen from a much larger set than the possible values of $\underline{s}c$.
- Unfortunately, this has major implications for signature size. One also needs to repeat the signing algorithm several times.
- Two main choices for the entries of $\underline{y}_1$: Discrete Gaussian or uniform.
- Since $\underline{s}c$ tends to behave like a Gaussian, one would think that Gaussians are better for $\underline{y}_1$.

Lyubashevsky public key signatures

- Vadim's Eurocrypt 2012 paper gives full details for SIS and LWE, and detailed security proof using the above ideas. For a security level of around 100-128 bits he gives signatures of around 16500 bits based on Ring-LWE ($n = 512$).
- The schemes can be implemented using uniform distributions instead of discrete Gaussians.
- Güneysu, Lyubashevsky and Pöppelmann (CHES 2012) give a very practical signature scheme implementable on smartcards. For 100-bit (based on non-standard assumptions) security level the signatures are around 9000 bits.
- At CRYPTO 2013 Vadim (with Ducas, Durmus and LePoint), use a "bi-modal trick" and other innovations (and based on non-standard assumptions). Gives signatures of around 5000-5500 bits.
- Getting close to the 2000-3000 bits for RSA signatures at that security level.

New results on public key signatures from LWE (joint with Shi Bai)

- Lyubashevsky proves knowledge of a solution $(\underline{s}, \underline{e})$ to an LWE instance $(A, \underline{b})$. Note that $\underline{s}$ has length n and $\underline{e}$ has length m, where $m \gg n$.
- Our idea is to prove knowledge only of $\underline{s}$.
- Public key: $A, T = AS + E \pmod{q}$ where A and T are $m \times n$ and S and E are $n \times n$.
- We use the fact that if $\underline{c}$ is a length n vector with very short entries $\{-1, 0, 1\}$ and low weight then $E\underline{c}$ is short.
- Let $d \in \mathbb{N}$ and $\underline{v} \in \mathbb{Z}^m$. Define $\lfloor \underline{v} \rceil_d$ to be a length m vector whose i-th entry is $\underline{v}_i / 2^d$.
- Choose d such that $\lfloor E\underline{c} \rceil_d = 0$ with high probability.

New signatures

- Public key: $A, T = AS + E \pmod{q}$.
- Signature (proof of S):
  - Choose $\underline{y}$ length n short entries.
  - $\underline{c} = H(\lfloor A\underline{y} \pmod{q} \rceil_d, \text{message}) =$ length n, entries $\{-1, 0, 1\}$, low weight.
  - Set $\underline{z} = \underline{y} + S\underline{c}$.
  - Do rejection sampling so that distribution of outputs $(\underline{z}, \underline{c})$ is independent of S.
  - Return $(\underline{z}, \underline{c})$.
- Verify: Check that $\underline{z}$ has short enough entries and then check that

$$H(\lfloor A\underline{z} - T\underline{c} \rceil_d, \text{message}) = \underline{c}.$$

- The point:

$$A\underline{z} - T\underline{c} = A(\underline{y} + S\underline{c}) - (AS + E)\underline{c} = A\underline{y} - E\underline{c}.$$

New signatures

- We obtain 13000 bit signatures (at 128-bit security level) based on **standard LWE** (no rings needed!) for parameters for which hardness of LWE is guaranteed by reductions to worst-case instances of standard lattice problems.
- Parameters: $(n, m, q, \sigma) = (584, 1166, \approx 2^{36}, 48)$.
- For these parameters we use uniform distributions during the signing protocol.

New signatures

- The main problem is that we need q to be very large compared with σ.
  Recall: $(n, m, q, \sigma) = (584, 1166, \approx 2^{36}, 48)$.
- Let L be the lattice $L = \{\underline{v} \in \mathbb{Z}^m : \underline{v} \equiv A\underline{s} \pmod{q}\}$.
  The volume of L is q^n.
- By the Gaussian heuristic, the shortest non-zero vector in L has Euclidean norm close to
  $\sqrt{m/(2\pi e)}\det(L)^{1/m} = \sqrt{m/(2\pi e)}q^{n/m} \approx 2235145$.
- However, the error vector has length approximately $\sqrt{m}\sigma \approx 1640$.
- This corresponds to Hermite factor 1.00635^m.

Thank You