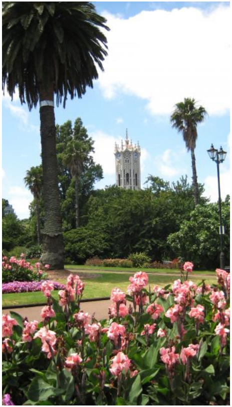
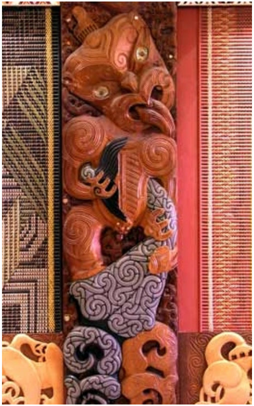
Open problems in lattice-based cryptography

Steven Galbraith

University of Auckland, New Zealand

Plan

Goal: Highlight some hot topics in cryptography, and good targets for mathematical cryptanalysis.

- Approximate GCD
- Homomorphic encryption
- NTRU and Ring-LWE
- Multi-linear maps

Please ask questions at any time.

Lattice-based cryptography

Lattice-based cryptography refers to any system whose security depends on computational assumptions based on lattices (in contrast to factoring-based cryptography, discrete-logarithm based cryptography, etc).

Some achievements:

- Fully homomorphic encryption
- Multilinear maps
- Attribute-based encryption for general circuits
- Cryptography based on worst-case assumptions
- Security against quantum computers (hopefully)

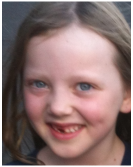
Symmetric encryption from approximate GCD

(van Dijk, Gentry, Halevi and Vaikuntanathan, 2010)

- Let p be large prime, known to Alice and Bob.
- To encrypt $m \in \{0, 1\}$ to Bob, Alice does:
  - Choose $q, e \in \mathbb{Z}$ with $|e| \ll p$ and q large.
  - Compute $c = pq + 2e + m$, and send to Bob.
- To decrypt c Bob does
  - $m = [[c]_p]_2$.
- Here $[c]_p$ denotes the integer in $(-p/2, p/2]$ congruent modulo p to c.

The approximate GCD problem

- Suppose Eve sees many communications between Alice and Bob.



- She sees $c_i = pq_i + (2e_i + m)$ for $1 \leq i \leq k$.
- One of her goals might be to compute p, and hence read all messages.

Homomorphic encryption

- A nice feature of this system is that it is homomorphic.
- Let $c_1 = pq_1 + 2e_1 + m_1$ and $c_2 = pq_2 + 2e_2 + m_2$.
- Then $c_1 + c_2 = p(q_1 + q_2) + 2(e_1 + e_2) + (m_1 + m_2)$ is an encryption of $m_1 + m_2 \pmod{2}$.
- Also, $c_1 c_2 = p(\star) + 2(e_1 e_2 + e_1 m_2 + e_2 m_1) + (m_1 m_2)$ is an encryption of $m_1 m_2 \pmod{2}$.
- Homomorphic encryption is a hot topic in crypto these days – Nigel will probably talk more about this.

Can turn into a public key encryption scheme

- Bob publishes many encryptions of zero $X_i = pq_i + 2e_i$, $1 \leq i \leq k$.
- To encrypt to Bob, Alice chooses $I \subseteq \{1, 2, \ldots, k\}$ and computes
$$c = \sum_{i \in I} X_i \quad + 2e + m$$
and sends c to Bob.
- Full security analysis given by van Dijk, Gentry, Halevi and Vaikuntanathan.
- Variant where $X_0 = pq_0$ is also given in public key, and computations are modulo X_0.
- (ρ, η, γ)-Approximate GCD problem: Given $X_1, \ldots, X_k \in \mathbb{Z} \cap [0, 2^\gamma]$ find an integer $2^{\eta-1} < p < 2^\eta$ such that $[X_i]_p < 2^\rho$ for all $1 \leq i \leq k$.
In what sense is this well-defined?

Euclid algorithm on approx-GCD

- Given $X_1 = pq_1 + e_1, X_2 = pq_2 + e_2$ one can run Euclid's algorithm.
- Since Euclid considers most-significant bits first, the algorithm will begin the same as if one was computing $\gcd(pq_1, pq_2)$.

Euclid algorithm on approx-GCD

- Given $X_1 = pq_1 + e_1, X_2 = pq_2 + e_2$ one can run Euclid's algorithm.
- Since Euclid considers most-significant bits first, the algorithm will begin the same as if one was computing $\gcd(pq_1, pq_2)$.
- Euclid on (a, b) computes a sequence (r_i, s_i, t_i) such that $r_i = as_i + bt_i$ and $|r_i s_i| \approx |b|, |r_i t_i| \approx |a|$.
- Run Euclid on (pq_1, pq_2) we expect to get $r_i = p$ and $q_1 s_i + q_2 t_i = 1$.
- This means $s_i, t_i \approx q_2, q_1$ and so

$$X_1 s_i + X_2 t_i = p(q_1 s_i + q_2 t_i) + (e_1 s_i + e_2 t_i).$$

  As long as $|e_1 s_i - e_2 t_i| \gg p$ then Euclid does not find p. Hence, if $\gamma - \eta + \rho \gg \eta$ then Euclid is not useful.
- Howgrave-Graham has also worked on this problem.

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 **rows**.
- The **basis matrix** is the $n \times n$ matrix B whose rows 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.

Lattice attack on approx GCD

- Recall $X_i = pq_i + e_i$.
- Consider the lattice whose rows are spanned by

$$B = \begin{pmatrix} 2^\rho & -X_2 & -X_3 & \cdots & -X_t \\ 0 & X_1 & 0 & \cdots & 0 \\ 0 & 0 & X_1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & X_1 \end{pmatrix}.$$

- Note that

$$(q_1, q_2, \ldots, q_t)B = (2^\rho q_1, e_1 q_2 - e_2 q_1, \ldots, e_1 q_t - e_t q_1)$$

is of length $\sqrt{t}2^{\rho+\gamma-\eta}$.

Lattice attack on approx GCD

- The Gaussian heuristic suggests the lattice contains a vector of length

$$\sqrt{\frac{t}{2\pi e}}\det(B)^{1/t} \approx \sqrt{\frac{t}{2\pi e}}2^{(\rho+(t-1)\gamma)/t}.$$

- So for large enough t then the target vector is especially short and might be found using lattice reduction.

Research problems

- Also attacks by: Chen-Nguyen and Coron, Naccache and Tibouchi ; Cohn-Heninger.
  These attacks show that the errors (hence, parameter ρ) cannot be too small.
  But mainly the security comes from the size of the q_i rather than the size of the errors.
- The suggested parameters make the scheme astronomically large.
- Find a better attack and kill it off completely!

Adaptive attacks

- It is standard (and realistic) in crypto to consider the setting where an attacker has access to a decryption oracle.
- Recall that decryption of a ciphertext c means computing $m = [[c]_p]_2$.
  Given a decryption oracle one can query it with even integers $c \approx p$ and determine p by binary search.
- The security notion we would like is called "IND-CCA1".
- Open problem: To design an IND-CCA1 variant of this scheme.
- Similar attacks apply to all known homomorphic encryption schemes.
- Loftus, May, Smart and Vercauteren have given an IND-CCA1 variant of the Smart-Vercauteren scheme.
- Micciancio and Peikert (EUROCRYPT 2012) have given IND-CCA1 secure encryption from LWE. But it is not homomorphic.

Multi-linear

- Coron, Lepoint and Tibouchi have given a multi-linear map based on somewhat similar ideas.
- It is too complicated to write down.
- A good idea would be to study this scheme carefully to assess its security.

End of part 1

Any comments or questions?

NTRU/Ring-LWE – History

- NTRU: Hoffstein, Pipher, Silverman (ANTS 1998). Rejuvinated by Stehlé and Steinfeld ; Lopez-Alt, Tromer and Vaikuntanathan
- LWE: Regev (2005)
- Ring-LWE: Lyubashevsky, Peikert and Regev

Cyclotomic rings

- $n = 2^k$, $R = \mathbb{Z}[x]/(x^n + 1)$. Then $x^n + 1$ is irreducible.
- R is a subring of $\mathbb{Q}(\zeta_{2n})$, which is a Galois extension of $\mathbb{Q}$.
- For $q \equiv 1 \pmod{2n}$ prime, let $R_q = R/(q) = \mathbb{Z}[x]/(q, x^n + 1)$
- Note: $x^n + 1$ splits completely modulo q.
- The canonical embedding $\sigma : R \to \mathbb{R}^n$ is formed using the n conjugate pairs of injective homomorphisms $\sigma_i : R \to \mathbb{C}$.

NTRU/Ring-LWE

- The "error distribution" on R is "diagonal in the canonical embedding", meaning that one samples independently n discrete Gaussians on $\mathbb{Z}$ and pulls back under σ to give an "error vector" $\underline{e} \in R$.
- Suppose we sample $\underline{s}, \underline{e}$ from the error distribution on R.
- The NTRU problem is: Given $\underline{a} = \underline{e}\ \underline{s}^{-1}$ in R_q, to compute $(\underline{s}, \underline{e})$.
  (This is not "traditional" NTRU.)
  Stehlé-Steinfeld: $\underline{a}$ is indistinguishable from uniform.
- The Ring LWE problem is: Given $(\underline{a}, \underline{b} = \underline{a}\ \underline{s} + \underline{e} \pmod{q}) \in R_q^2$ to compute $(\underline{s}, \underline{e})$.
- One can write NTRU as $(\underline{a}, 0 = \underline{a}\ \underline{s} - \underline{e} \pmod{q})$.

Interlude: 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 a secret vector.
- Suppose one is given an $n \times m$ matrix $\mathbf{A}$ chosen uniformly at random with entries in $\mathbb{Z}_q$ and a length m vector

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

  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 $\underline{s}$.
- Can be expressed as $\underline{b} \equiv (\underline{s}, \underline{e})\binom{\mathbf{A}}{\mathbf{I}} \pmod{q}$.

Encryption from Ring-LWE

- Public key: $(\underline{a}, \underline{b} = \underline{a}\ \underline{s} + \underline{e} \pmod{q}) \in R_q^2$
- Private key: $(\underline{s}, \underline{e})$
- Encrypt $\underline{m} \in \{0,1\}^n$ encoded in R:
  - Choose small $\underline{r}, \underline{e}_1, \underline{e}_2$
  - Compute $\underline{u} = \underline{a}\ \underline{r} + \underline{e}_1 \pmod{q}$, $\underline{v} = \underline{b}\ \underline{r} + \underline{e}_2 + [q/2]\underline{m}$
  - Send $(\underline{u}, \underline{v})$
- Decrypt $(\underline{u}, \underline{v})$:

$$\underline{v} - \underline{u}\ \underline{s} \equiv \underline{e}\ \underline{r} + \underline{e}_2 - \underline{e}_1\ \underline{s} + [q/2]\underline{m} \pmod{q}$$

so most significant bits yield $\underline{m}$.

Encryption from NTRU

- Public key: $\underline{a} = 2\underline{e}(2\underline{s}+1)^{-1} \pmod{q}) \in R_q$
- Private key: $2\underline{s}+1$
- Encrypt $\underline{m} \in R$
  - Sample short $\underline{e}_1, \underline{e}_2 \in R$
  - $c = \underline{a}\ \underline{e}_1 + 2\underline{e}_2 + \underline{m}$
- Decrypt c:

$$c(2\underline{s}+1) \equiv 2\underline{e}\ \underline{e}_1 + 2\underline{e}_2(2\underline{s}+1) + (2\underline{s}+1)\underline{m} \pmod{q}$$

so least significant bits yield $\underline{m}$.

Other applications of Ring-LWE/NTRU

- Lopez-Alt, Tromer and Vaikuntanathan have given a homomorphic encryption scheme based on NTRU.
- Brakerski, Gentry and Vaikuntanathan have given homomorphic encryption based on LWE/Ring-LWE.
- Vadim will talk about efficient public key signatures based on Ring-LWE and NTRU.

Lattice attack on NTRU (Coppersmith-Shamir)

- NTRU: Given $\underline{a}$ such that there exist $(\underline{s}, \underline{u}, \underline{e})$ with $\underline{a}\ \underline{s} + q\underline{u} = \underline{e}$.
- Let $\mathbf{A}$ be circulant matrix corresponding to $\underline{a}$ and let $\underline{s}$ be a vector corresponding to the ring element. Then $\underline{s}\mathbf{A}$ is a vector corresponding to $\underline{s}\ \underline{a}$.
  Then
  $$(\underline{s}, \underline{u}) \begin{pmatrix} \mathbf{I} & \mathbf{A} \\ 0 & q\mathbf{I} \end{pmatrix} = (\underline{s}, \underline{e})$$
  is a short vector in the row lattice.
- To prevent this attack need to use large dimension.

Lattice attack on Ring-LWE

- Given $(\underline{a}, \underline{b} = \underline{a}\, \underline{s} + \underline{e} + q\underline{u}) \in R_q^2$.
- Just like the previous case

$$(\underline{s}, \underline{u}) \begin{pmatrix} \mathbf{I} & \mathbf{A} \\ 0 & q\mathbf{I} \end{pmatrix} = (\underline{s}, \underline{b} - \underline{e}) \approx (0, \underline{b}).$$

- Hence, we have an instance of the closest vector problem in a lattice.
  Natural to expect since NTRU is like Ring-LWE with $\underline{b} = 0$.

Interlude: Lattice attack on LWE

- LWE: Given A and $\underline{b} \equiv \underline{s}A + \underline{e} \pmod{q} \in \mathbb{Z}^m$, find $\underline{s} \in \mathbb{Z}^n$.
- Let $L = \{\underline{v} \in \mathbb{Z}^m : \underline{v} \equiv \underline{s}A \pmod{q}$ for $\underline{s} \in \mathbb{Z}^n\}$.
  Then L is a lattice of rank m and (usually) volume q^{m-n}.
- To solve LWE we want to find a lattice point $\underline{y} \equiv \underline{s}A \pmod{q}$ 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 $\underline{y} \equiv \underline{s}A \pmod{q}$.
- 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 $m \approx \sqrt{n \log(q)/\log(\delta)}$.
  (δ = Hermite factor.)

Further work

- Alex May (2001) used "zero run" and "dimension reducing" tricks to speed up the lattice attack on NTRU.
- Craig Gentry (2001) used a ring homomorphism to reduce to smaller dimensional problem, which is why we now use $x^n + 1$ where $n = 2^k$.
- Gama, Howgrave-Graham and Nguyen (EUROCRYPT 2006) discussed "symplectic lattice reduction" in the context of NTRU.
- Howgrave-Graham (CRYPTO 2007) considered hybrid "meet-in-middle" and lattice reduction approaches.
- Has similar cryptanalytic effort been made on Ring-LWE?

Multilinear maps (Garg, Gentry, Halevi 2013)

- A pairing is a non-degenerate, bilinear map $e : G_1 \times G_2 \rightarrow G_3$.
- Typically constructed out of the Weil or Tate-Lichtenbaum pairing on elliptic curves.
- It would be interesting to have a non-degenerate multilinear map $e : G_1 \times G_2 \times \cdots \times G_k \rightarrow G_{k+1}$.
- We can't really do that yet, but there is something slightly analogous.
- The one-way function $g \rightarrow g^x$ is replaced by "randomised encodings" a of random elements x.
- The "multilinear map" is essentially a homomorphic multiplication of these encodings, followed by an operation that "deterministically extracts some bits" from the product.

Multilinear maps (Garg, Gentry, Halevi 2013)

- Let g be a short vector, defining a principal ideal $I = (g)$ in $R_q = \mathbb{Z}_q[x]/(x^n + 1)$. Also need g invertible and g^{-1} short.
- $z \in R_q$ is random and invertible.
- Public key includes $y = (1 + gr)/z$, $x_i = gb_i/z$, and $p_{\text{zt}} = hz^k/g$, where r, b_i are short and h is medium size.
- To generate "random exponent" one chooses a short vector d in R_q.
- To generate a "randomised (level one) encoding of x" one computes

$$\begin{aligned} u &= dy + \sum_i r_i x_i \\ &= (d + g(r + \sum_i r_i b_i))/z = (d \pmod{(g)} + g(\text{small}))/z. \end{aligned}$$

- Idea: It is hard to determine d given u.

Multilinear maps (Garg, Gentry, Halevi 2013)

- Given randomized (level one) encodings $u_1, \ldots, u_k$ all of the form $(d_i + g \text{ small})/z$ one computes

$$u = u_1 \cdots u_k = (d_1 \cdots d_k + g \text{ smallish})/z^k.$$

- Now, recall $p_{\text{zt}} = hz^k/g$, so

$$up_{\text{zt}} = (d_1 \cdots d_k)(h/g) + h \text{ smallish}.$$

- Since (h/g) is a constant and h smallish is smallishish, the most significant bits of the representation of up_{zt} depend only on $d_1 \cdots d_k$.

Multilinear maps (Garg, Gentry, Halevi 2013)

- Given randomized (level one) encodings $u_1, \ldots, u_k$ all of the form $(d_i + g \text{ small})/z$ one computes

$$u = u_1 \cdots u_k = (d_1 \cdots d_k + g \text{ smallish})/z^k.$$

- Now, recall $p_{\text{zt}} = hz^k/g$, so

$$up_{\text{zt}} = (d_1 \cdots d_k)(h/g) + h \text{ smallish}.$$

- Since (h/g) is a constant and h smallish is smallishish, the most significant bits of the representation of up_{zt} depend only on $d_1 \cdots d_k$.
- Secure? Your guess is as good as mine.

Computational assumption and applications

- The computational assumption needed for crypto applications is: Given a k-multilinear map and $k+1$ randomised encodings $u_1, \ldots, u_{k+1}$ of values $d_1, \ldots, d_{k+1}$ it is hard to compute the value of the k-multilinear map on encodings of $d_1 d_2 \cdots d_{k+1}$.
- Note that can compute the k-multilinear map for values $d_1, \ldots, d_l$ when $l \leq k$.
- Cryptographic applications of multilinear maps:
  - k-party Diffie-Hellman
  - Attribute/Functional encryption
  - Witness encryption
  - Programmable hash functions
  - etc

Differences with pairings

- For pairings, the "encoding" is $d \to g^d$, which is a one-way function (both phrases important here!)
- For GGH the encoding is $d \to dy$, which is not one-way, unless one adds extra randomisation in which case it is not a function.
- Pairings give a group homomorphism from one group to another, typically $E(\mathbb{F}_q) \to \mathbb{F}_{q^k}^*$.
- GGH gives an "algebraic map" (multiplication of ring elements) followed by a non-algebraic map (extraction of most significant bits).

Thank You