CSE 291: Lattices in Cryptography and Cryptanalysis	Nov. 16, 1999

Lecture 8

Lecturer: Daniele Micciancio   *Scribe: Andreas Yannakopoulos*

1 Simple Cryptanalysis

In the first lecture, we mentioned the wide application of lattices in cryptology and especially in cryptanalysis and saw some classic problems in this field. But how exactly can lattices be used to "break" cryptanalysis problems?

Let's consider the following problem: a message **m** is sent to three different recipients using low-exponent RSA with public keys the relatively prime numbers A, B and C respectively. If we assume that we encrypt the message m simply applying the RSA encryption function, then the ciphertexts are $m^3 \bmod A$, $m^3 \bmod B$, $m^3 \bmod C$ for each recipient respectively. An adversary that sees all the ciphertexts can recover the entire message by using the Chinese remainder theorem to compute c=$m^3 \bmod ABC$, notice that $m^3 < ABC$ and recover the message m by computing the cubic root of c over the integers.

But usually the message contains the identity **id** of the recipient and is represented as (m; id) $= m * 2^{|id|} + id$ or, for the purpose of our analysis, as (m; id) = m+id. Then using a low-exponent RSA encryption function, we have the ciphertexts $(m+a)^3 \bmod A$, $(m+b)^3 \bmod B$, $(m+c)^3 \bmod C$ respectively, where a, b, c are the identities of the recipients, and the cryptanalysis above does not hold any longer. How can we "break" the problem using lattices?

2 Cryptanalysis and lattices

Each ciphertext gives us a corresponding modular polynomial equation:

$$\begin{aligned} p_a(m) &= m^3 + a_2m^2 + a_1m + a_0 = 0 \bmod A \\ p_b(m) &= m^3 + b_2m^2 + b_1m + b_0 = 0 \bmod B \\ p_c(m) &= m^3 + c_2m^2 + c_1m + c_0 = 0 \bmod C \end{aligned}$$

For example, $(m+a)^3 = c_a \bmod A$ can be written as $m^3+(3a)m^2+(3a^2)m+(1-c_a) = 0 \bmod A$. Using the Chinese Remainder Theorem [1] , we can combine the coefficients

[1]**Chinese remainder theorem:** Let $n = n_1n_2...n_k$, where the n_i are pairwise relatively prime. Consider the correspondence $a \leftrightarrow (a_1, a_2, ..., a_k)$, where $a \in \mathbb{Z}_n, a_i \in \mathbb{Z}_{n_i}$ and $a_i = a \bmod n_i$ for i=1,2,...,k. Then the above mapping is a one-to-one correspondence (bijection) between $\mathbb{Z}_n$ and the Cartesian product $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times ... \times \mathbb{Z}_{n_k}$.

of the three polynomials to obtain a single polynomial

$$p(m) = 0 \bmod ABC.$$

The polynomial p is a monic (i.e. coefficient of the highest term=1) polynomial of degree 3. Notice that the polynomial equation $p(x) = 0 \bmod (ABC)$ has a solution m which is much smaller than the modulus $M = ABC$. In particular, since $m < A, B, C$ to guarantee unique decryption, we have $m^3 < M$ and therefore $m < M^{1/3}$.

Motivated by such problems as the above, we'll present a general solution which illustrates how lattices can be used to find **small solutions to such polynomial modular equations.**

3 Finding small solutions to polynomial equations

We want to solve the following problem.

Problem 1 *Given a (monic) polynomial equation* $p(x) = c_0 + c_1 x + ... + x^d = 0 \bmod M$*, find a small solution* m *such that* $p(m) \equiv 0 \bmod M$ *and* $|m| < \sqrt[d]{M}$*.*

We first show how to find solutions when they are smaller than some bound X to be determined. Then we will show that $X \approx \sqrt[d]{M}$.

The main points of the algorithm are the following:

- Build a lattice whose vectors correspond to valid equations (equation lattice)
- The small vectors of the lattice correspond to equations with small coefficients. No modular reduction is done in such equations.
- Find an approximately shortest vector in the lattice
- Solve the corresponding equation over $\mathbb{Z}$

We note that if the coefficients of the polynomial p satisfy $\sum_{i=0}^{d} |c_i X^i| < M$, then any small solution $|m| \leq X$ to modular equation $p(m) = 0 \bmod M$ is also a solution to $p(m) = 0$ over the integers because

$$|p(m)| \leq \sum_{i=0}^{d} |c_i X^i| < M.$$

This suggests a way to represent the polynomial p as a vector:

$$\vec{p} = [c_0, c_1 X, c_2 X^2, ..., c_{d-1} X^{d-1}, X^d].$$

Then, the condition $\sum_{i=0}^{d} |c_i X^i| < M$ on the coefficients of p can be easily expressed as a bound on the ℓ_1 norm of the corresponding vector $\|\vec{p}\|_1 < M$.

Unfortunately, the vector corresponding to the given polynomial $\vec{p}$ does not usually satisfies the bound $\|\vec{p}\|_1 < M$. We now show how to build a lattice whose vectors correspond to valid equations (i.e., equations satisfied my the small solution m). We will actually obtain equations with a bigger modulus M^{h-1} where h is an integer parameter to be specified. Then we will look for short vectors in this lattice, i.e., vectors with ℓ_1 norm less than M^{h-1}.

Our attempt starts from noting that m is a solution to $p(x) = 0 \bmod M$. We get that m is also a solution to $p(x)^i = 0 \bmod M^i$, for example $p(m)^2 = 0 \bmod M^2$ and $p(m)^3 = 0 \bmod M^3$. Multiplying by x^j, we get equations $x^j p(x)^i = 0 \bmod M^i$ that are also satisfied by m, for example $m \cdot p(m) = 0 \bmod M$ and $m^2 p(m)^3 = 0 \bmod M^3$. In order to make the modulus in all these equations the same we multiply them by appropriate powers of M to obtain polynomial equations

$$q_{ij}(x) = M^{h-i-1} x^j p(x)^i = 0 \bmod M^{h-1}$$

where $i = 0, ..., h-1$, $j = 0, ..., d-1$. All these equations have a common solution m which is much smaller than the modulus M^{h-1}. Moreover, any integer linear combination of these polynomials gives also an equation modulo M^{h-1} with solution m. So, we consider the lattice generated by the vectors $\vec{q}_{ij}$ and look for a lattice vector $\vec{q}$ such that $\|\vec{q}\|_1 < M^{h-1}$. If we can find such a vector, then we can recover m by solving the corresponding equation $q(x) = 0 \bmod M^{h-1}$ over the integers.

From the form of the polynomial p(x), we can see that: $degree(q_{ij}) = di + j \in \{0, ..., dh-1\}$ and all polynomials q_{ij} have a different degree. If we arrange the polynomials q_{ij} in a matrix, we get a lower triangular matrix B:

$$B = [\vec{q}_{ij}] = \begin{bmatrix} M^{h-1}X^0 & 0 & \cdots & 0 \\ ? & M^{h-2}X^{d+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ ? & \cdots & ? & M^0 X^{dh-1} \end{bmatrix}$$

where each row correspond to a polynomial. (Notice: in this lecture we are representing lattice vectors as row vectors.)

The basis B is in Hermite normal form [2] and the elements on the diagonal are given by

$$b_{di+j, di+j} = M^{h-i-1} X^{di+j}.$$

Therefore the determinant of the lattice is

$$det(B) = \prod_{i,j} M^{h-i-1} X^{di+j}$$

[2] **Hermite normal form:** since we are using rows to represent vectors, this is slightly different from the definition given in class, because it correspond to an upper triangular matrix when vectors are represented by columns. It is easy to see that the two definitions are equivalent.

$$
\begin{aligned}
&= M^{\sum_{i,j}(h-i-1)} \prod_{i,j} X^{di+j} \\
&= M^{d\sum_{i=0}^{h-1}(h-i-1)} X^{\frac{dh(dh-1)}{2}} \\
&= M^{\frac{dh(h-1)}{2}} X^{\frac{dh(dh-1)}{2}}.
\end{aligned}
$$

Therefore, we can use Minkowski's theorem [3] to get a bound on the (ℓ_2) length of the shortest non-zero lattice vector:

$$\lambda(B) < \sqrt{dh} \cdot \mathrm{unit}(B) = \sqrt{dh} M^{\frac{h-1}{2}} X^{\frac{dh-1}{2}}.$$

(Notice: the dimension of the lattice is dh.)

Using the LLL algorithm we can find a lattice vector $\vec{q}$ such that

$$\|\vec{q}\|_2 < 2^{dh} \cdot \lambda(B) < 2^{dh} \cdot \sqrt{dh} \cdot M^{\frac{h-1}{2}} X^{\frac{dh-1}{2}}.$$

and using the inequality $\|\vec{q}\|_1 \le \sqrt{dh}\|\vec{q}\|_2$ we get

$$\|\vec{q}\|_1 < 2^{dh} \cdot dh \cdot M^{\frac{h-1}{2}} X^{\frac{dh-1}{2}}.$$

Therefore, if we set

$$X = \left(\frac{M^{\frac{h-1}{2}}}{dh \cdot 2^{dh}} \right)^{\frac{2}{dh-1}}.$$

we get $\|\vec{q}\|_1 < M^{h-1}$ and we can recover m by solving the equation $q(x) = 0 \bmod M^{h-1}$ over the integers.

We now want to estimate X and show that $X \approx M^{1/d}$. Let $h = \log M$, so that the construction above has complexity polynomial in the input size $d \cdot \log M$. We have

$$
\begin{aligned}
X &> \left(\frac{M^{\frac{h-1}{2}}}{dh \cdot 2^{dh}} \right)^{\frac{2}{dh}} \\
&= \frac{M^{1/d}}{2^{2+1/d} \cdot (dh)^{2/(dh)}} \\
&> \frac{M^{1/d}}{2^3}.
\end{aligned}
$$

So, X is less than $M^{1/d}$, but only by a constant factor. It is easy to see that the problem of finding a solution $|m| < M^{1/d}$ to a polynomial equation of degree d can be easily reduced to solving a constant number of problems in which the solution $|m'| < X = M^{1/d}/c$. Namely, one can guess the higher order bits of m and express m as $m = w + m'$ where w is known and $|m'| < X$. If w is guessed correctly, then m' is a small solution to the equation $p'(x) = p(w + x) = 0 \bmod M$ and we can find it in polynomial time because $|m'| < X$. Since there are only a small constant number of possible values for w, we can try them all until we find a solution.

[3] **Minkowski Theorem:** For every basis $B \in \mathbb{R}^{m \times n}$ there exists a lattice vector $\vec{v} \in \mathcal{L}(B) \setminus \{\vec{0}\}$ such that $\|\vec{v}\|_2 < \sqrt{n} \cdot unit(B)$, where $unit(B) = \sqrt[n]{\det(B)}$