CSE 207C: Lattices in Cryptology Winter 2002

Problem Set 2

Instructor: Daniele Micciancio Mar. 4, 2002

Due: Mar 14, 2002

In this problem you will use lattices (together with some algebraic tools described below) to solve two cryptanalysis problems. We first recall the algebraic tools required.

The resultant

The resultant is an algebraic tool that can be used to eliminate variables in multivariate systems of polynomial equations. Here we recall the definition and basic properties.

Let $p(x, y)$ and $q(x, y)$ be two polynomials over a field F, and let (a, b) be a common solution, i.e., $p(a, b) = q(a, b) = 0$. We want to eliminate variable x, and obtain a single equation $z(y) = 0$ such that b is a solution.

Think of p, q as polynomials in a single variable x, with coefficients in $F[y]$, e.g., $p(x, y) = \sum_i p_i(y) \cdot x^i$, and similarly for q. Then polynomials p, q can be represented as vectors with entries in $F[y]$: $[p_0(y), p_1(y), \ldots, p_{\deg_x(p)}(y), 0, \ldots, 0]$, where $\deg(p)$ is the degree of p with respect to x. Similarly for q. We are going to use vectors with $n = \deg_x(p) + \deg_x(q)$ coordinates. Build a $n \times n$ matrix $M(y)$ with entries in $F[y]$ whose rows correspond to polynomials $p(x, y)$, $x \cdot p(x, y)$,..., $x^{\deg_x(q)-1} \cdot p(x, y)$ and $q(x, y)$, $x \cdot q(x, y)$,..., $x^{\deg_x(p)-1} \cdot q(x, y)$. The *resultant* of p and q with respect to x is $z(y) = \det(M(y))$, i.e., the determinant of matrix $M(y)$.

theorem 1 *For any (sufficiently large) field F, and polynomials $p, q \in F[x, y]$, if $p(a, b) = q(a, b) = 0$, then $z(b) = 0$, where $z(y)$ is the resultant of p and q with respect to x.*

The relatively simple proof of the theorem above is left as an (optional) excercise. From the proof it is easy to see that the theorem holds even when F is not a field, but more generally, any integral domain (a ring satisfying special properties). In this problem set we use the resultant to eliminate variables in polynomial equations over ring $F = Z_N$ which is not a field (or even an integral domain) unless N is prime. In the solution to the problem, you can disregard this technical point, and assume that variable elimination works. In practice, if some problem arise because Z_N is not a field, this will likely expose the factorization of N, allowing key recovery attack.

So, just assume that the resultant gives a way to perform variable elimination, i.e., given equations $p(x, y) = 0$ and $q(x, y) = 0$, the resultant gives a single equation $z(y) = 0$ such that all solutions to the original equations are also solutions to the new one. Notice that the degree of z is usually larger than the degree of the original equations. In particular, $\deg_y(z) \leq (\deg_x(p)\deg_y(q) + \deg_x(q)\deg_y(p)$.

The Franklin-Reiter attack

The following is a simple attack to low exponent RSA due to Franklin and Reiter. You might need this technique to complete the solution to problem 1.

Assume that two related messages m and $m' = m + r$ are encrypted with low exponent RSA, and the difference r between the messages is known. An observer sees the ciphertexts $c = m^3 \bmod N$ and $c' = (m')^3 \bmod N$. Then the message m can be recovered from r, c, c', N as follows

$$m = \frac{r(c' + 2c - r^3)}{c' - c + 2r^3} \pmod N$$

as easily verified by substitution and simple algebraic manipulation.

Problem 1

Consider a TCP/IP network where security is implemented at the IP layer using low exponent RSA (say, with exponent 3). IP is an inherently unreliable protocol: when you send an IP packet, there is no guarantee that the packet will reach the destination. We consider a scenario where a sender pass a message m to the TCP protocol. TCP encapsulate the message m using an header h (see below), and pass the concatenated packet $h; m$ to the secure IP protocol. The (secure) IP protocol encrypts the packet and passes $(h; m)^3 \bmod M$ to an underlying (insecure) IP layer. The TCP header h consists of 20 bytes (plus options if any), containing the following information:

- byte 0: 4-bit version and 4-bit header length,
- byte 1: type of service
- bytes 2-3: 16-bits total length (in bytes)
- bytes 4-5: 16-bit identification
- bytes: 6-7: 3-bits flag, 13-bit fragment offset
- byte 8: time to live
- byte 9: protocol
- bytes 10,11: header checksum
- bytes 12,13: source IP address
- bytes 14,15: destination IP address
- bytes 16, ...: options. We assume that no options are given

The rest of the packet contains the actual data.

Assume that because of a network fault (or a malicious attack by an adversary) the packet is not received. Then the reliable TCP protocol will send the same message m again. The header h' used in the second transmission will be identical as in the first transmission, except for the packet identification number and checksum. The packet identification number can be either a counter, or randomly chosen at every transmission. Notice that each time one of the fields is incremented by x, the checksum is decremented by the same amount.

In this problem you should assume that the number of packet sent in between the transmissions is not known, whether the packet identification number is chosen at random or incremented at every transmission, the difference between the counters in the first and second transmission is not known.

Use the resultant, and the lattice based method to solve polynomial equations, to show that given $(h;m)^3 \bmod M$ and $(h';m)^3 \bmod M$, one can efficiently recover the message m, even without knowning the factorization of the modulus M.

[Hint: Try to find the numerical difference $d = (h;m) - (h';m)$ between the two packets before encryption.]

In this problem you are not required to implement the attack. Just give a description and brief analysis of the attack.

Problem 2

In this problem you are asked to analyze the attack to the truncated linear congruential generator described in class. The generator is defined by public parameters a, b, m. Given a seed x_0, one applies the recurrence $x_{i+1} = ax_i + b \bmod m$, and for each i, outputs $c_i = \lfloor x_i/2^k$, where k is the number of truncated lower order bits, e.g., $k = 100$.

Notice that by induction on i, it follows that $x_i = a^i x_0 + b_i \bmod m$ where $b_i = b\sum_{j<i} a^j \bmod m$. Build the matrix corresponding to the equations:

$$A = \begin{bmatrix} a^0 & -1 & 0 & \dots & 0 \\ a^1 & 0 & -1 & \dots & 0 \\ \vdots & & & \ddots & 0 \\ a^l & 0 & \dots & 0 & -1 \end{bmatrix}$$

We want to solve $Ax = b' \pmod m$, where $b' = -[b_0, \dots, b_l]^T$. Remember the method sudied in class. We substitute $x = 2^k c + y$, where $0 \le y_i \le 2^k$, and obtain an equivalent system $Ay = b'' \bmod m$ with small solution y. We then setup a lattice

$$B = \begin{bmatrix} \alpha m I & \alpha A \\ 0 & I \end{bmatrix}$$

where α is a sufficiently large constant. We run the nearest plane algorithm to find a lattice point approximately closest (say, within a factor $2^{n/2}$) to the target $t = [\alpha(b'')^T 0^T]^T$.

We proved that if lattice $L = \{x : Ax = 0 \bmod m\}$ does not contain short vectors then the attack is successful.

In this problem you are asked to analyze the success probability of the attack, as a function of the parameters k (the number of truncated bits), l (the stretching factor of the generator) and $\log m$ (the bitsize of the modulus).

You can either run some experiments, and tell for which values of the parameters you successfully broke the generator, or analyse the success probability of the attack by evaluating the length of the shortest vector in L.

If you decide to perform a theoretical analysis, here is a hint: You want to bound the number of a such that there exists a t for which $|ta^i|$ is small for all i. Prove that all such a's are solutions to a polynomial with small coefficients. This can be done considering all polynomials p with small coefficients, bounding the value of $t \cdot p(a)$ using the bounds on ta^i and the coefficients of p, and finally using the pigeonhole principle. Since there are more polynomials that output values, two polynomials $tp(a)$ and $tp'(a)$ have the same value. It follows that $(p - p')(a) = 0$ and the coefficients of $(p - p')$ are small.