RSA AND PROBABILISTIC FACTORING

BURTON ROSENBERG
UNIVERSITY OF MIAMI

Contents

1. Introduction

There are notes for the Introduction to Cryptography course. This is the mathematical background to the RSA cryptosystem including an RP algorithm for primality testing, and the reduction of breaking RSA to integer factorization.

2. Extended Euclidean Algorithm

For positive integers a, b, let (a, b) be the greatest common divisor (GCD) of a and b. That is, (a, b) is a number such that it divides both a and b, and for d dividing both a and b, d also divides (a, b):

$$(a, b)|a, b \text{ and } \forall\, d \text{ s.t. } d|a, b \text{ then } d|(a, b).$$

Theorem 2.1. The GCD of two integers is an integer linear combination of those integers,

$$\exists\, s, t \in Z \text{ s.t. } sa + tb = (a, b).$$

We can prove this along the way of showing how to calculate the GCD.

Date: March 28, 2016.

Definition 2.1 (Euclidean Algorithm)**.** Given integers a, b such that $a > b > 0$, define a sequence of remainders:

$$\begin{aligned} r_0 &= a, \\ r_1 &= b, \\ r_2 &= r_0 \,\%\, r_1, \\ &\cdots \\ r_i &= r_{i-2} \,\%\, r_{i-1}, \\ &\cdots \\ d &= r_{K-1} \,\%\, r_K, \\ 0 &= r_K \,\%\, d \end{aligned}$$

where $x \,\%\, y$ is the remainder when y divides x.

Theorem 2.2. Given the Euclidean Algorithm, as defined above, then d is the GCD of a and b.

My favorite proof of this uses the definition of an ideal.

Definition 2.2 (Ideal)**.** Given integers a, b, the ideal generated by a and b is the set of all integer linear combinations of a and b:

$$\langle a, b \rangle = \{\, s\, a + t\, b \,|\, s, t \in Z \,\}$$

Proof. I leave it to the reader to prove that, for $a > b$,

$$\langle a, b \rangle = \langle b, a \,\%\, b \rangle.$$

It follows immediately that,

$$\langle a, b \rangle = \langle d, 0 \rangle$$

where d is as given by the Euclidean Algorithm, and the ideal is in fact all multiples of d.

Since $a, b \in \langle a, b \rangle$, then $d | a, b$ therefor $d | (a, b)$. Since $(a, b) | a, b$, then (a, b) divides all integer linear combinations of a and b, hence $\forall\, x \in \langle a, b \rangle$ then $(a, b) | x$. Since $d \in \langle a, b \rangle$ then $(a, b) | d$. So $d = (a, b)$. □

The Extended Euclidean Algorithm back-substitutes in the Euclidean Algorithm to calculate the s and t promised by the first theorem.

Algorithm 2.1 (Extended Euclidean Algorithm)**.** Given the r_i and K as in the above Euclidean Algorithm, we have the basis:

$$d = r_{K-1} - q_K r_K.$$

Given the induction hypothesis:

$$d = s_{i-1}\, r_{i-1} - t_{i-1} r_i$$

the induction step subsitutes $r_i = r_{i-2} - q_{i-1} r_{i-1}$,

$$\begin{aligned} d &= s_{i-1} r_{i-1} + t_{i-1}(r_{i-2} - q_{i-1} r_{i-1}) \\ &= t_{i-1} r_{i-2} + (s_{i-1} - t_{i-1} q_{i-1}) r_{i-1} \\ &= s_{i-2} r_{i-2} - t_{i-2} r_{i-1}. \end{aligned}$$

and can therefore by induction derive,

$$d = s_1\, r_1 - t_1\, r_0 = t\, b + s\, a.$$

3. Square roots of unity in Z_n and Miller-Rabin

Let n be an integer, and Z_n be the ring of integers modulo n. We consider all x such that $x^2 = 1 \pmod{n}$.

Definition 3.1 (Nontrivial roots of unity)**.** In Z_n^*, the solutions to $x^2 = 1 \pmod{n}$ are called the roots of unity. If the roots are ± 1, then they are the trivial roots of unity. Else they are the non-trivial square roots of unity.

Theorem 3.1. Let n be an odd integer. If Z_n^* there are always the trivial roots of unity. If n is not a prime nor a prime power; else there are also non-trivial roots of unity.

Proof. Since $x^2 = 1$ in Z for $x = \pm 1$, it is true also for Z_n^*.

If n is a prime or a prime power, let $p|n$ be the prime. Then p cannot divide both $(x-1)$ or $(x+1)$. Since

$$x^2 - 1 = (x-1)(x+1),$$

then a square root of unity mod n implies,

$$p|(x-1)(x+1).$$

Hence either $n|(x+1)$ or $n|(x-1)$ and so $x = \pm 1 \pmod{n}$.

Conversely, if n is not a prime or prime power, let $n = a\, b$ with $(a, b) = 1$. Then there exists s and t such that $= s\, a + t\, b = 1$. Note,

$$(s\, a - t\, b)^2 = (s\, a)^2 + (t\, b)^2 = (s\, a + t\, b)^2 = 1 \pmod{n}$$

and therefore $s\, a - t\, b \neq 1 \pmod{n}$ is a nontrivial square root of unity. □

Therefore we can demonstrate that n is composite without having to factor n through the number theoretic properties of n in at least two ways.

Definition 3.2 (Witness to non-primality)**.** Given n. A number w relatively prime to n is a witness of non-primality of n if either,

(1) $w^{n-1} \neq 1 \pmod{n}$, contradicting Little Fermat, or
(2) $w \neq \pm 1 \pmod{n}$ however $w^2 = 1 \pmod{n}$.

In the first case, w is called a weak witness. In the second case, w is called a strong witness.

It turns out that if n is not prime, a random x as a fair chance of being a witness to primality.

Algorithm 3.1 (Miller-Rabin)**.** Chose a $w \in Z_n^*$ at random. Do this by picking a non-zero $w \in Z_n$ at random and checking that $(w, n) = 1$. Else w is a witness by way of having factored n "by mistake".

Check if $w^{n-1} \neq 1 \pmod{n}$. If so, w is a weakness witness. Else write

$$n - 1 = s\, 2^u,$$

with s odd and set $v = w^s \pmod{n}$ and iteratively replace v with $v^2 \pmod{n}$ stopping when $v = 1 \pmod{n}$. If the value before the final squaring is -1, the test is inconclusive. Else w is a strong witness to non-primality.

This shows that problem of primality is co-RP. A problem is co-RP if non-membership (i.e. being a composite) has a witness that can be found in polynomial time with high probability. An algorithm for a problem in co-RP never concludes falsely against the proposition. In this case, if the algorithm concludes that n is not a prime it has a witness as definitive proof. However, with vanishing probability the algorithm can be in error in acclaiming the statement (i.e. that n is a prime), as it only has as proof not having found a witness otherwise.

Note that a strong witness allows n to be factored.

Theorem 3.2. Let ϵ be a nontrivial square root of unit in Z_n^*, with n odd. Then $(\epsilon - 1, n)$ and $(\epsilon + 1, n)$ give nontrivial factors of n.

.

Proof. Since,

$$(\epsilon - 1)(\epsilon + 1) = \epsilon^2 - 1 = 0 \pmod{n}$$

and prime dividing both $\epsilon - 1$ and $\epsilon + 1$ is even. So each prime dividing n is either due to $\epsilon - 1$ or $\epsilon + 1$. Therefore $(n, \epsilon + 1)$ will be a nontrivial factor of n consisting of some of the primes at the full power in n, and $(n, \epsilon - 1)$ will be the remaining of the primes, again at their full power. □

For an odd n, the number of square roots of unity will be 2^r where r is the number of distinct primes tha divide n. This is bets seen by the isomorphism,

$$Z_n \cong \Pi_{p^\alpha || n} Z_{p^\alpha}.$$

But 2, no pun intended, is odd. While Z_{p^α} has exactly two roots of unity, Z_2 has only 1 root of unity, Z_4 has 2, and all Z_{2^j} for $j > 2$ have 4 roots of unity.

Example: The first Carmichael number $561 = 3 \cdot 11 \cdot 17$ has the 8 roots of unity 1, 67, 188, 254, 307, 373, 494, 560. As an example, $(561, 66) = 3 \cdot 11$ and $(561, 68) = 17$.

When n is even, things are diffierent.

4. RSA and the hardness of factoring

The RSA cryptosystem depends on the difficulty of given an n and a e, and n is of the form $n = pq$ where p and q are distinct primes, finding a d such that $x^{ed} = x \pmod{n}$.

Definition 4.1 (RSA public key cryptography)**.** Let p, q be distinct odd primes and $n = pq$. Let e be an element of $Z^*_{\phi(n)}$, where ϕ is the Euler phi function, and d the inverse of e mod $\phi(n)$,

$$e\,d = 1 \pmod{\phi(n)}.$$

Then for any message $m \in Z^*_n$, the encryption of m by public key e is,

$$E_e(m) = m^e \pmod{n}$$

and the decryption of message $c \in Z^*_n$ by secret key d is,

$$D_d(c) = c^d \pmod{n}.$$

Lemma 4.1.

$$D_d(E_d(m)) = m \pmod{n}$$

Proof.

$$D_d(E_e(m)) = m^{de} = m^{k\phi(n)+1} = m^{k\phi(n)}m = 1^k m = m \pmod{n}.$$

□

Knowing the value of $\phi(n)$, the inverse of e mod $\phi(n)$ is efficiently computable using the Extended Euclidean Algorithm. However, knowledge of $\phi(n)$ is equivalent to factoring n. Consider the equation (over Z),

$$x^2 - (n + 1 - \phi(n))x + n = x^2 - (p + q)x + pq = (x - p)(x - q) = 0.$$

It has roots p and q, and the roots can be found in polynomial time using the quadratic formula.

Knowledge of d is not exactly the same as knowledge of $\phi(n)$. However since,

$$ed - 1 = k\phi(n),$$

then,

$$x^{ed-1} = x^{k\phi(n)} = 1^k = 1 \pmod{n}$$

for any $x \in Z_n^*$. This fact can be used with Miller-Rabin to force the appearance of a strong witness of non-primality for n, which consequently can factor n.

Theorem 4.1. Let (e, p, q) be an RSA cryptosystem. With public key $(e, n = pq)$ and private key $d = e^{-1} \pmod{(p-1)(q-1)}$. Let $A(e, n) \to d$ be an Probabilistic Polynomial Time algorithm for extracting the secret key from the public key. Then there exists a Probabilistic Polynomial Time algorithm $B^A(e, n) \to (p, q)$ that factors n.

Proof. Run $A(e, n)$ to get d. Write $ed - 1 = 2^u s$ where s is odd. Choose a non-zero $w \in Z_n$ at random and check if $(n, x) = 1$, else stop with x a factor of n. Set $v = w^s \pmod{n}$ and repeatedly set v to $v^2 \pmod{n}$ until it is 1. It must eventually be one because $ed = k\phi(n)$. If a nontrivial root of unity appears, $\epsilon^2 = 1 \pmod{n}$, with $\epsilon \neq \pm 1$, factor n by $(n, \epsilon + 1)$ or $(n, \epsilon - 1)$. □

Example: Let the primes be 7 and 11 so $n = 77$. Let $e = 7$, and $\phi(77) = 6 \cdot 10 = 60$. The Extended Euclidean algorithm gives,

$$43 \cdot 7 + (-5) \cdot 60 = 1$$

so $7^{-1} = 43 \pmod{60}$ and $d = 43$. Then,

$$7 \cdot 43 - 1 = 300 = 2^2 \cdot 75.$$

Let w be 29 (after a few choices). Then $29^{75} = 43$, and $43^2 = 1$. Then $(44, 77) = 11$ and $(42, 77) = 7$.