

CS556 Introduction to Cryptography - Prof. Ming-Deh Huang

Scribe: Iftikhar A Burhanuddin

burhanud@usc.edu

Class #9, #10 - September 24, 26, 2002

Announcements:

Notes for classes #7 and #8 are online. HW #2 will be posted later today and will be due next week. Check the webpage. Please read § 4.1 for next Tuesday.

Topics for today:

1. Homomorphisms
2. RSA security
3. RSA problem
4. Group Membership test
5. Breaking RSA

1. Homomorphisms

Definition. Let $(G, *)$ and (H, \odot) be groups. A function $\phi : G \rightarrow H$ is a *Homomorphism* if, for all $a, b \in G$,

$$\phi(a * b) = \phi(a) \odot \phi(b)$$

A homomorphism that is also a bijection is called an *isomorphism*. If ϕ is an isomorphism from $(G, *)$ to $(G, *)$ then the function is called an *automorphism*. So $E(x)$ and $D(x)$, the RSA encryption and decryption functions are automorphisms on $\mathbf{Z}/n\mathbf{Z}^*$. In the previous lectures we referred to this property by saying that these functions were *multiplicative* as E (and D) send the product x and y to product of $E(x)$ and $E(y)$ (and $D(x)$ and $D(y)$ respectively).

2. RSA security

In the previous lecture, we saw that if we can decrypt 1% (or any tiny fraction) of all the RSA ciphertext instances we can decrypt 100% (the whole)

and hence in some sense the system was “homogeneous”. We’ll henceforth refer to the procedure of randomizing ciphertext to fall into our “lucky set” as *Randomized Reduction*. Informally this says that if the system is indeed secure every ciphertext is as hard to decrypt as every other. So there is no subset which is easy to decrypt - no particular weak spot.

But this results in differing viewpoints: From the adversarial/cryptanalytic perspective: to crack 100% it is *enough* to crack 1% and from the cryptographer’s point of view: the adversary should not be able to decrypt even 1%.

3. RSA problem

Given n, e, y to compute $D(y)$, that is, to compute x such that $E(x) = y$. This is the foundational security question of RSA. Other attacks take advantage of implementation loopholes of RSA and/or weaknesses of protocols based on RSA. We’ll look at the broader range of attacks later but for now we’ll focus on this particular issue.

We know that i) **Factoring $n \Rightarrow$ Solving RSA problem** since if we can compute p and q given $n(= pq)$ then we can compute $\phi(n) = (p - 1)(q - 1)$ which in turn implies that we can solve for d ($de \equiv 1 \pmod{\phi(n)}$) and RSA is broken. Hence RSA is often termed as a public key cryptosystem based on integer factoring. But is it necessary to factor to break RSA? The question of whether **Factoring $n \Leftarrow$ Solving RSA problem**, is an open question.

From the above argument we see that ii) **Computing $\phi(n) \Rightarrow$ Solving RSA. Exercise.** Computing $\phi(n) \Rightarrow$ Factoring n

We next claim the following *weakening*: iii) **Finding m such that $a^{\phi(m)} \equiv 1 \pmod{n}$ for all a where $\gcd(a, n) = 1 \Rightarrow$ Solving RSA.** The remainder of this lecture and the next one will be about coming up with a constructive proof (that is the proof will give us an algorithm) for the above claim.

Say $m = \text{lcm}(p - 1, q - 1)$. FLT says $a^{p-1} \equiv 1 \pmod{p}$ for all a , with $\gcd(a, p) = 1$. Also $a^{q-1} \equiv 1 \pmod{q}$ for all a , with $\gcd(a, q) = 1$. Since m is a multiple of $p - 1$ and $q - 1$ we get $a^m \equiv 1 \pmod{p}$ and $a^m \equiv 1 \pmod{q}$ and since $\gcd(p, q) = 1$, we have $a^m \equiv 1 \pmod{n}$. Notice that $m | \phi(n)$ and $m \neq \phi(n)$

Definition. A number a is said to be k -smooth if all prime factors of a are bounded by k . Ex: $p=17$, $p-1$ is 2-smooth

Definition. A number b is said to be square free if $n^2 \nmid b$ for all $n > 1$.
 Ex: 2.23.37

Suppose $p - 1$ and $q - 1$ are both k -smooth. If $p - 1$ is square free then $p - 1 \mid k!$. More generally, $p - 1 \mid (k!)^c$, with $c \leq \log p$. Similarly $q - 1 \mid (k!)^c$. So $\text{lcm}(p - 1, q - 1) \mid (k!)^c$. So take $m = (k!)^c$.

Remark: We have $\|k!\| \sim \log n \log \log n$ and $\|(k!)^c\| \sim c \log n \log \log n \sim \log^2 n \log \log n$ (if $k \sim \log n$). So $m = (k!)^c$ can be constructed *quickly* in $O(\log^3 n)$ time using our factor base of primes less than k . Therefore when we generate RSA primes p and q , we must make sure that $p - 1$ and $q - 1$ are not $\log n$ -smooth otherwise the system is vulnerable to the above attack.

Thm 1: Let $n = pq$. Suppose we are given m such that $a^m \equiv 1 \pmod n$ for all $a \in \mathbf{Z}/n\mathbf{Z}^*$ then we can factor n in expected time polynomial in $\log m$

Corollary: Suppose $m = \phi(n)$ is given to us then we can factor n in random polynomial time.

The algorithm we will describe is randomized in nature and *Random* polynomial time is the expected running time.

Thm: Suppose p is prime. Then $\mathbf{Z}/p\mathbf{Z}^*$ is cyclic of order $p - 1$, that is, $\mathbf{Z}/p\mathbf{Z}^* = \langle g \rangle = 1, g, \dots, g^{p-2}$. We'll prove this later.

We know that $a^{p-1} \equiv 1 \pmod p$ for all a with $\text{gcd}(a, p) = 1$. Say $p > 2$ ie p is an odd prime. As $a \in \mathbf{Z}/p\mathbf{Z}^*$, $a = g^i$ for some i , we have $a^{\frac{p-1}{2}} \equiv g^{\frac{p-1}{2}i} \pmod p$. And $g^{\frac{p-1}{2}i}$ equals 1 if i is even (as $p - 1$ divides $\frac{p-1}{2}i$) and equals -1 if i is odd (as $p - 1$ doesn't divide $\frac{p-1}{2}i$) and therefore $a^{\frac{p-1}{2}} \equiv \pm 1 \pmod p$. So half the powers of g (the even powers) give 1 and the other half (the odd powers) give -1 .

Thm: $\left| \{a \in \mathbf{Z}/p\mathbf{Z}^* \mid a^{\frac{p-1}{2}} \equiv 1 \pmod p\} \right| = \frac{|\mathbf{Z}/p\mathbf{Z}^*|}{2} = \frac{p-1}{2}$. We'll formally prove this latter for now the above intuition will do.

4. Group Membership Test

A requirement in our algorithm (which we'll soon describe) is to verify that $a^m \equiv 1 \pmod n$ for all $a \in \mathbf{Z}/n\mathbf{Z}^*$. How do we check that the above congruence holds? For large n there are a lot of bases a , so instead of iteratively picking the bases we'll randomize their selection and claim that we have a *high* level of confidence that the congruence holds for all bases.

Let $H_l = \{a \in \mathbf{Z}/n\mathbf{Z}^* \mid a^l \equiv 1 \pmod{n}\}$. Observe that H_l is a subgroup of $\mathbf{Z}/n\mathbf{Z}^*$. If H_l is the whole group then the congruence is satisfied for all $a \in \mathbf{Z}/n\mathbf{Z}^*$. On the other hand if $H_l \neq \mathbf{Z}/n\mathbf{Z}^*$ then H_l is a proper subgroup and by Lagrange's theorem $\frac{|H_l|}{\phi(n)} \leq \frac{1}{2}$. And so if we draw bases at random from $\mathbf{Z}/n\mathbf{Z}^*$, $a^l \not\equiv 1 \pmod{n}$ with probability $1 - \frac{|H_l|}{\phi(n)} \geq \frac{1}{2}$. If we draw at random atleast 1 out of 2 bases are going to stand witness to the congruence not holding. So after i successful trials the probability of the congruence *not* holding for some base is atleast $1 - (\frac{1}{2})^i$. Hence we can repeat the trials till the error probability becomes really small we have a *high* level of confidence.

So if H_l is the whole group $\mathbf{Z}/n\mathbf{Z}^*$ then the test passes 100% of the time. If $H_l \neq \mathbf{Z}/n\mathbf{Z}^*$ then the test passes $\leq 50\%$ of the time.

Say we are given m such that $x^m \equiv 1 \pmod{n}$ for all x with $\gcd(x, n) = 1$.

Algorithm:

1. Check $H_l = \mathbf{Z}/n\mathbf{Z}^*$ (using the above Group membership test).
2. If it is true $m \leftarrow \frac{m}{2}$, goto [1]. Exit otherwise.

When we exit we have an m such that $x^{2^m} \equiv 1 \pmod{n}$ for all $x \in \mathbf{Z}/n\mathbf{Z}^*$ and $x^m \not\equiv 1 \pmod{n}$ for some $x \in \mathbf{Z}/n\mathbf{Z}^*$.

5. Breaking RSA

Thm 2: Let $n = pq$, $p \neq q$ primes. Suppose $x^{2^m} \equiv 1 \pmod{n}$ for all $x \in \mathbf{Z}/n\mathbf{Z}^*$ but $x^m \not\equiv 1 \pmod{n}$ for some $x \in \mathbf{Z}/n\mathbf{Z}^*$. Then for randomly chosen $a \in \mathbf{Z}/n\mathbf{Z}^*$, $\gcd(a^m - 1, n) = p$ or q with probability $\frac{1}{2}$.

$x^{2^m} \equiv 1 \pmod{n}$ for all $x \in \mathbf{Z}/n\mathbf{Z}^* \Rightarrow p-1 \mid 2^m$ and $q-1 \mid 2^m \Rightarrow \frac{p-1}{2} \mid m$ and $\frac{q-1}{2} \mid m$.

$x^m \not\equiv 1 \pmod{n}$ for some $x \in \mathbf{Z}/n\mathbf{Z}^* \Rightarrow$ either $x^m \not\equiv 1 \pmod{p}$ or $x^m \not\equiv 1 \pmod{q}$ for some x with $\gcd(x, n) = 1 \Rightarrow p-1 \nmid m$ or $q-1 \nmid m$.

Therefore $\frac{p-1}{2} \mid m$ and $\frac{q-1}{2} \mid m$ and either $p-1 \nmid m$ or $q-1 \nmid m$.

So we end up with 3 cases:

1. $p-1 \nmid m$ but $q-1 \mid m$
2. $p-1 \mid m$ but $q-1 \nmid m$

3. $p - 1 \nmid m$ but $q - 1 \nmid m$

case 1: $p - 1 \nmid m$ but $q - 1 \mid m$
 H_m is not the whole group.

$$H_m \rightarrow \{x \in \mathbf{Z}/p\mathbf{Z}^* \mid x^{\frac{p-1}{2}} \equiv 1 \pmod{p}\} \times \mathbf{Z}/q\mathbf{Z}^*$$

When we draw a random a from $\mathbf{Z}/n\mathbf{Z}^*$ we have $a^m \equiv 1 \pmod{q}$ (since $q - 1 \mid m$) and $a^m \equiv 1 \pmod{p}$ with probability $\frac{1}{2}$. Therefore $\gcd(a^m - 1, n) = q$ with probability $\frac{1}{2}$ and $\gcd(a^m - 1, n) = n$ with probability $\frac{1}{2}$.

case 2: $p - 1 \mid m$ but $q - 1 \nmid m$

Using analysis which is similar to case 1, we have $\gcd(a^m - 1, n) = p$ with probability $\frac{1}{2}$ and $\gcd(a^m - 1, n) = n$ with probability $\frac{1}{2}$.

case 3: $p - 1 \nmid m$ and $q - 1 \nmid m$

$$H_m \rightarrow \{x \in \mathbf{Z}/p\mathbf{Z}^* \mid x^{\frac{p-1}{2}} \equiv 1 \pmod{p}\} \times \{x \in \mathbf{Z}/q\mathbf{Z}^* \mid x^{\frac{q-1}{2}} \equiv 1 \pmod{q}\}$$

Picking a random $a \in \mathbf{Z}/n\mathbf{Z}^*$ we have:

1. Prob $[(a^m \equiv 1 \pmod{p}, a^m \equiv -1 \pmod{q}) \text{ or } (a^m \equiv -1 \pmod{p}, a^m \equiv 1 \pmod{q})] = \frac{1}{2}$ and
2. Prob $[(a^m \equiv 1 \pmod{p}, a^m \equiv 1 \pmod{q}) \text{ or } (a^m \equiv -1 \pmod{p}, a^m \equiv -1 \pmod{q})] = \frac{1}{2}$.

So $\text{Prob}[\gcd(a^m - 1, n) = p \text{ or } q] = \frac{1}{2}$ and we don't get anything useful as $\text{Prob}[\gcd(a^m - 1, n) = 1 \text{ or } n] = \frac{1}{2}$.

Hence in each of the three scenarios with probability $\frac{1}{2}$ we obtain either p or q . Therefore in about two random trials RSA is broken and this completes the proof of Theorem 2. \diamond

Remark: Group membership test with Thm 2 \Rightarrow Thm 1.

The *intuition* behind why the algorithm works is that there is an asymmetry between the groups $\mathbf{Z}/p\mathbf{Z}^*$ and $\mathbf{Z}/q\mathbf{Z}^*$ and we leverage this by searching for a number which exhibits different parity mod p and mod q .

Later we'll use techniques similar to the ones above to search for RSA primes. We might say that the same tricks are used both to break and make a system!

But mind you Thm 1 starts with a *Suppose* and this is a very big assumption $p - 1, q - 1$ being smooth are special cases. When RSA primes are generated care is taken that p and q be big and also that $p - 1$ and $q - 1$ are not be $(\log n)$ smooth.