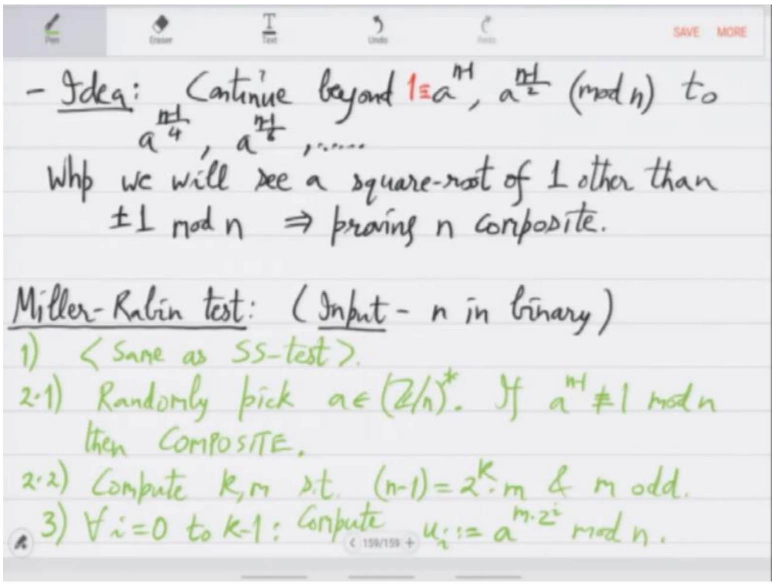
Computational Number Theory and Algebra
Prof. Nitin Saxena
Department of Aerospace Engineering
Indian Institute of Technology - Kanpur

Lecture – 25
Deterministic Primality Test (AKS) and RSA cryptosystem

(Refer Slide Time: 00:14)



- Idea: Continue beyond $1 \equiv a^{n-1}, a^{\frac{n-1}{2}} \pmod n$ to $a^{\frac{n-1}{4}}, a^{\frac{n-1}{8}}, \ldots\ldots$

Whp we will see a square-root of 1 other than ± 1 mod n $\Rightarrow$ proving n composite.

Miller-Rabin test: (Input - n in binary)

1) < Same as SS-test >.

2.1) Randomly pick $a \in (\mathbb{Z}/n)^*$. If $a^{n-1} \not\equiv 1$ mod n then COMPOSITE.

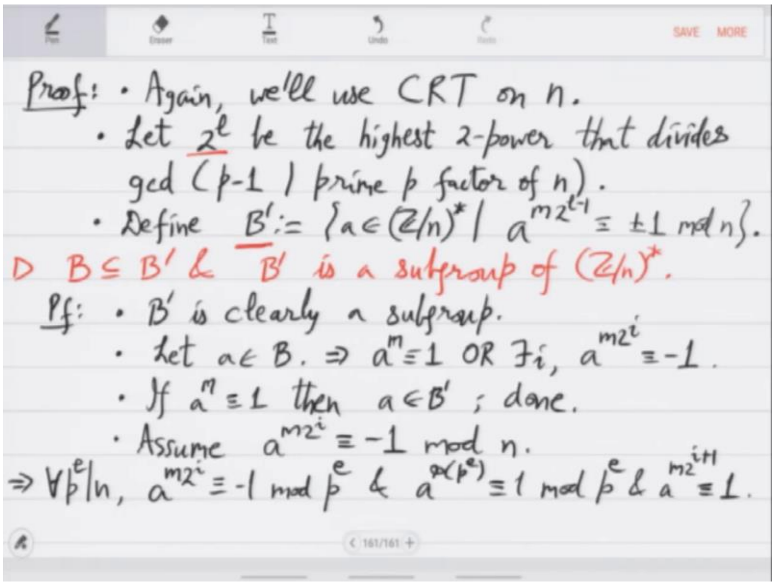
2.2) Compute k, m s.t. $(n-1) = 2^k \cdot m$ & m odd.

3) $\forall i = 0$ to $k-1$: Compute $u_i := a^{m \cdot 2^i}$ mod n.

So last time we saw this algorithm in green which is proposed by Miller and Robin. So input is number n is given in binary first step you just check whether n is a perfect power or not or n is even and then for a random a what you will do is you look at these u i’s look at step 3 so u i is just e raised to n - 1 a raised to n - 1 by 2 dot dot a raised to m mod n and you see whether this sequence has something unusual.

So unusual would mean that 1 has a square root other than + - 1 that is what you check it is a very fast algorithm. Now we were in the mid in the middle of this theorem proof of this theorem which says that if n has 2 or more distinct prime factors then the number of bad a is which will force the algorithm to say prime when in fact n is composite is only 1 4th phi n by 4. So let us continue with the proof.

(Refer Slide Time: 01:35)



Proof: • Again, we'll use CRT on n.

• Let 2^{ℓ} be the highest 2-power that divides $\gcd(p-1 \mid \text{prime } p \text{ factor of } n)$.

• Define $B' := \{a \in (\mathbb{Z}/n)^* \mid a^{m2^{\ell-1}} \equiv \pm 1 \bmod n\}$.

▷ $B \subseteq B'$ & B' is a subgroup of $(\mathbb{Z}/n)^*$.

Pf: • B' is clearly a subgroup.

• Let $a \in B$. $\Rightarrow a^m \equiv 1$ OR $\exists i$, $a^{m2^i} \equiv -1$.

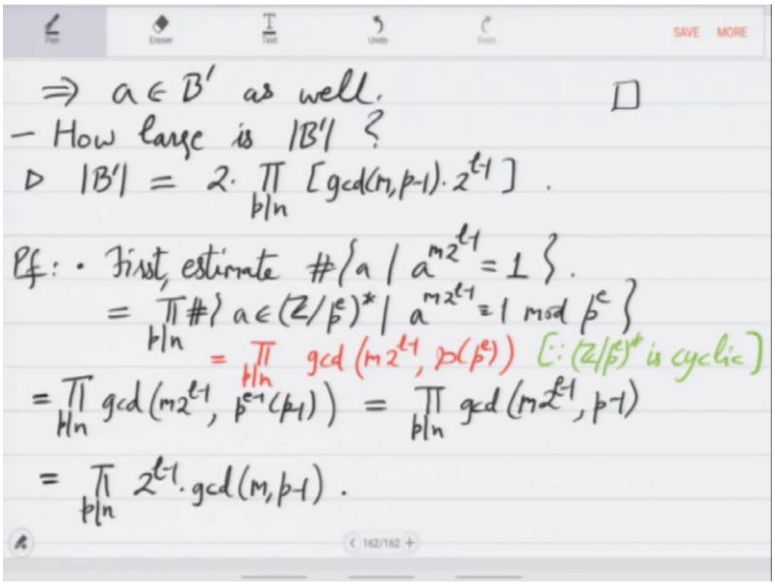
• If $a^m \equiv 1$ then $a \in B'$; done.

• Assume $a^{m2^i} \equiv -1 \bmod n$.

$\Rightarrow \forall p^e \| n$, $a^{m2^i} \equiv -1 \bmod p^e$ & $a^{\varphi(p^e)} \equiv 1 \bmod p^e$ & $a^{m2^{i+1}} \equiv 1$.

So what we did the way we started is we defined this B prime which is a sub group of z n star and B is a subset of B prime this we had shown so we will now focus on upper bounding the size of B prime.

(Refer Slide Time: 01:56)



$\Rightarrow a \in B'$ as well. $\square$

– How large is $|B'|$?

▷ $|B'| = 2 \cdot \prod_{p|n} [\gcd(m, p-1) \cdot 2^{\ell-1}]$.

Pf: • First estimate $\#\{a \mid a^{m2^{\ell-1}} = 1\}$.

$= \prod_{p|n} \#\{a \in (\mathbb{Z}/p^e)^* \mid a^{m2^{\ell-1}} \equiv 1 \bmod p^e\}$

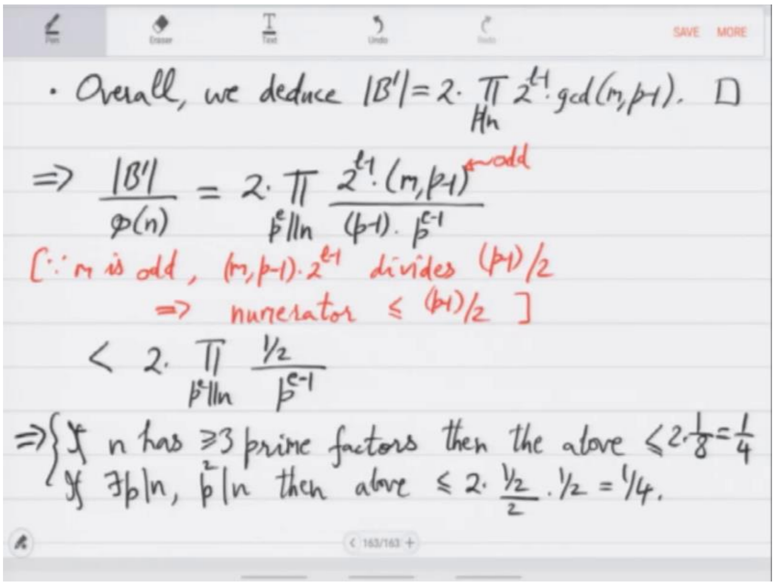
$= \prod_{p|n} \gcd(m2^{\ell-1}, \varphi(p^e))$ [$\because (\mathbb{Z}/p^e)^*$ is cyclic]

$= \prod_{p|n} \gcd(m2^{\ell-1}, p^{e-1}(p-1)) = \prod_{p|n} \gcd(m2^{\ell-1}, p-1)$

$= \prod_{p|n} 2^{\ell-1} \cdot \gcd(m, p-1)$.

So we have shown that B is a subset and we have shown this bound we have shown this bound which is related to basically gcd of m with p - 1 and do this for all the primes that divide n. This is the size of B prime.

(Refer Slide Time: 02:18)



• Overall, we deduce $|B'| = 2\cdot\prod_{p^e\|n} 2^{e-1}\cdot gcd(m,p-1)$. □

$$\Rightarrow \frac{|B'|}{\phi(n)} = 2\cdot\prod_{p^e\|n}\frac{2^{e-1}\cdot\overset{\leftarrow odd}{(m,p-1)}}{(p-1)\cdot p^{e-1}}$$

[$\because$ m is odd, $(m,p-1)\cdot 2^{e-1}$ divides $(p-1)/2$

$\Rightarrow$ numerator $\leq (p-1)/2$]

$$< 2\cdot\prod_{p^e\|n}\frac{1/2}{p^{e-1}}$$

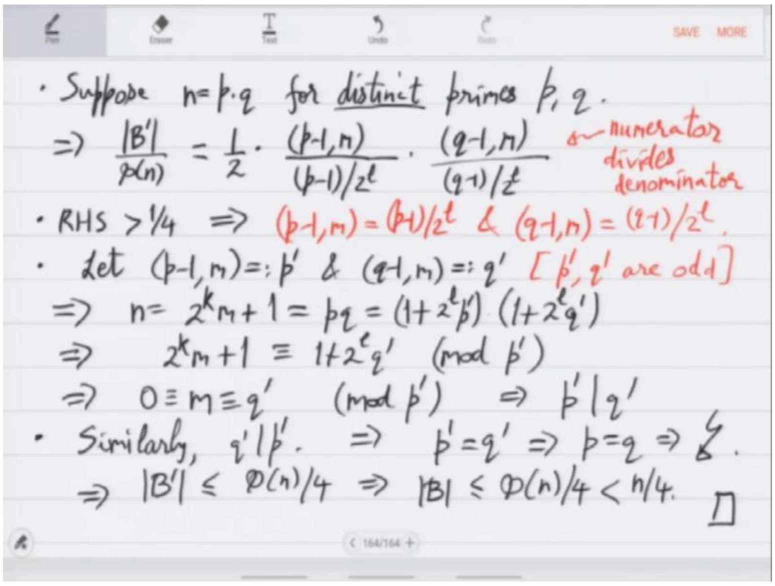
$\Rightarrow$ { If n has ≥ 3 prime factors then the above $\leq 2\cdot\frac{1}{8}=\frac{1}{4}$

If $\exists p|n$, $p^2|n$ then above $\leq 2\cdot\frac{1/2}{2}\cdot 1/2 = 1/4$.

And now look at the ratio of B prime over phi n. So this ratio came out to be smaller than product of half by p raised to e – 1. So, clearly if n has 3 prime factors or more then the above is less than 2 times 1 over 8 which is 1 over 4 because half will be the contribution from every prime p so you will get 1 by 4. So the proof will be over other good cases if there is a prime p dividing n such that p square divides n which means e is at least 2 for this p.

Then the above fraction will be less than equal to twice so this 1 contribution will already be half will be half divided by 2 because p is 2 - 1 will be at least p which is at least 2 and the other will give you at least half so which is again 1 by 4.

(Refer Slide Time: 04:34)



• Suppose $n = p\cdot q$ for distinct primes p, q.

$$\Rightarrow \frac{|B'|}{\phi(n)} = \frac{1}{2}\cdot\frac{(p-1,n)}{(p-1)/2^\ell}\cdot\frac{(q-1,n)}{(q-1)/2^\ell}$$ ← numerator divides denominator

• RHS $> 1/4 \Rightarrow (p-1,m) = (p-1)/2^\ell$ & $(q-1,m) = (q-1)/2^\ell$.

• Let $(p-1,m) =: p'$ & $(q-1,m) =: q'$ [p', q' are odd]

$\Rightarrow n = 2^k m + 1 = pq = (1+2^\ell p')(1+2^\ell q')$

$\Rightarrow 2^k m + 1 \equiv 1 + 2^\ell q' \pmod{p'}$

$\Rightarrow 0 \equiv m \equiv q' \pmod{p'} \Rightarrow p' | q'$

• Similarly, $q' | p'$. $\Rightarrow p' = q' \Rightarrow p = q \Rightarrow$ ↯.

$\Rightarrow |B'| \leq \phi(n)/4 \Rightarrow |B| \leq \phi(n)/4 < n/4$. □

So these are good cases so the only case when this is not which in which case this exceeds 1 by 4 is so suppose n is equal to p times q for distinct primes this is the only case that we have to analyze. So in this case the ratio is half times gcd of p - 1 by m let us write it like this and q - 1 mgcd divided by q - 1 by 2 raised to l. Now this RHS exceeds 1 by 4 then what it means is that the product is exceeding 1 by 8.

So see you have to remember that the numerators divide the denominator here. So p - 1, mgcd divides p - 1 divided by 2 raised to l - 1 this is because m is odd. So removing these 2 powers will not change the divisibility and the symmetric thing for q – 1, n m so this means that either this fraction is 1 or it is smaller than half right. So if both of them are less than equal to half it is the same expression as before but here it will it will be better to argue.

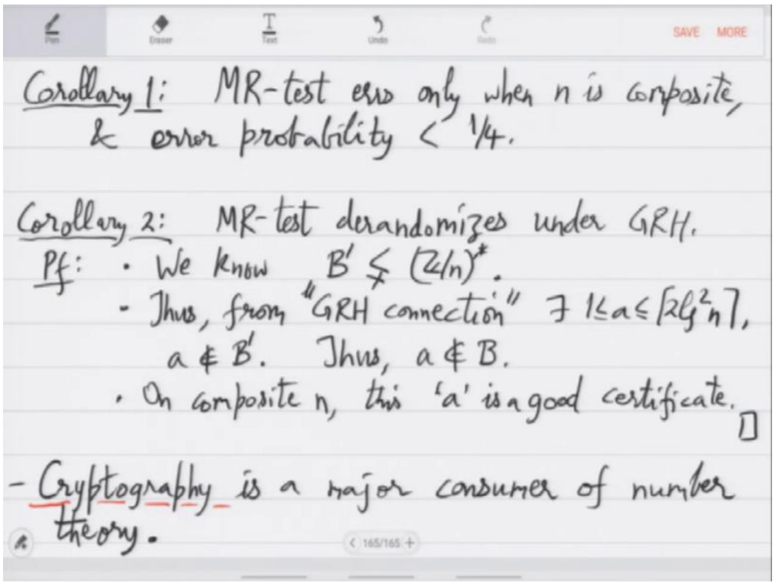
So now look at these 2 fractions again either so each of these fractions is either 1 or it is smaller than half if even 1 of these is smaller less than equal to half then this product will be less than equal to 1 4th. So right hand side greater than 1 by 4 means the only possibility is that both of them are each of them is 1. So p - 1 m is p - 1 2 raised to l and q - 1 mgcd is q - 1 divided by 2 raised to l that is the only possibility only bad case to analyze.

So let us call this p - 1 by 2 raise to l p prime. So let p - 1 be p prime and q - 1 m be q prime both are odd because m is odd. So p - 1 is 2 raise to l times p prime and q - 1 is 2 raised to l times q prime and n is 2 raised to k m + 1 but n is also p times q. So p is 1 + 2 raised to l p prime 1 + 2 raised to l q prime right. So this means this means that 2 raised to k m + 1 is 1 + 2 raised to l q prime mod p prime.

So mod p prime this first term p prime vanishes and p prime is also odd so which means that m is congruent to q prime mod p prime but p prime by definition divides m. So m is actually zero. So which means that p prime divides q prime and analogously q prime will divide p prime. So both of them will mean that p prime and q prime are the same which will mean if p prime and q prime are same then p and q are same which is a contradiction.

So this contradiction means that B prime by phi n is 1 by 4 or less which implies that B is upper bounded by B prime so it is also less than phi n by 4 which is much less than n by 4. So these bad is which will fool the algorithm on a composite n output is prime these are less than 1 4th. So the error probability is less than 1 4th that is what we have shown.

(Refer Slide Time: 11:14)



Corollary 1: MR-test errs only when n is composite, & error probability $< 1/4$.

Corollary 2: MR-test derandomizes under GRH.

Pf: • We know $B' \subsetneq (\mathbb{Z}/n)^*$.

- Thus, from "GRH connection" $\exists\, 1 \le a \le \lceil 2\lg^2 n \rceil$, $a \notin B'$. Thus, $a \notin B$.

• On composite n, this 'a' is a good certificate. □

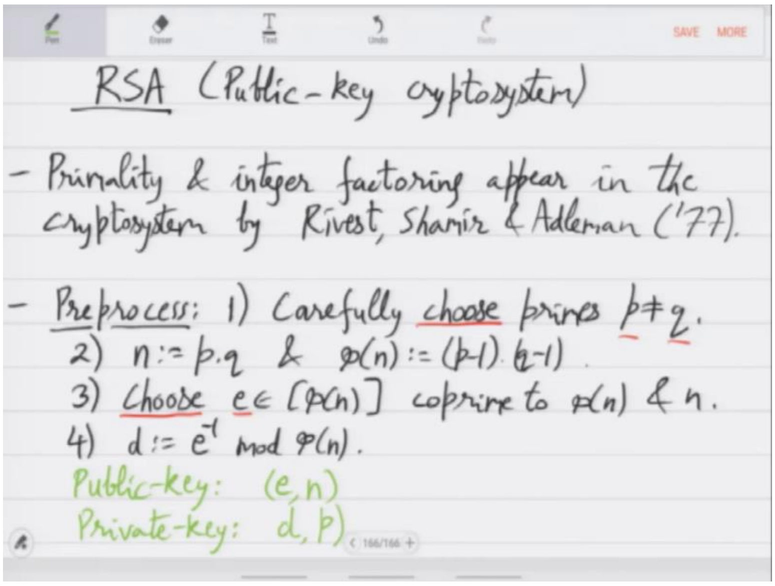
- Cryptography is a major consumer of number theory.

So Miller Robin test is only when n is composite and error probability is less than 1 4th. So it is a very good test its quadratic time and less than 25% error. Second simple observation is that MR test also can be derandomized under GRH. Why is that well this is just this Ankhony Bark theorem under GRH so we now know that B prime is a proper subgroup of z n star. Thus from the GRH connection there exists an a between 1 and 2 log square n such that e is not in B prime.

Thus the same a is not in B because B is a subset of B prime. So on a composite n this a is a good certificate. So this will force miller robin test to output composite. So it works and all this is clearly deterministic polynomial time. So that finishes both solution and Miller Robin test. Now obviously these are fundamental results. These are the best one of the best industry grade algorithms to test whether a number is prime or not where the number can be humongous so the digits can be thousands and or more.

Where is it used so some of you may know that cryptography is a major consumer of number theory. So that we have already seen when we covered the entro cryptosystem but that used lattices. NTRU is a lattice based SVP problem based L cube algorithm based cryptosystem. But there is something even more elementary simpler which we will now exhibit which is called the RSA cryptosystem currently that is used more often over the internet especially when you connect to your bank or when you use ssh https and so on. So example when you use HTTPS or SSH or SFTP or digital signatures etcetera.

(Refer Slide Time: 16:45)



RSA (Public-key cryptosystem)

- Primality & integer factoring appear in the cryptosystem by Rivest, Shamir & Adleman ('77).

- Preprocess: 1) Carefully choose primes $p \neq q$.
  2) $n := p \cdot q$ & $\varphi(n) := (p-1) \cdot (q-1)$.
  3) Choose $e \in [\varphi(n)]$ coprime to $\varphi(n)$ & n.
  4) $d := e^{-1} \bmod \varphi(n)$.
  Public-key: (e, n)
  Private-key: $d, p)$

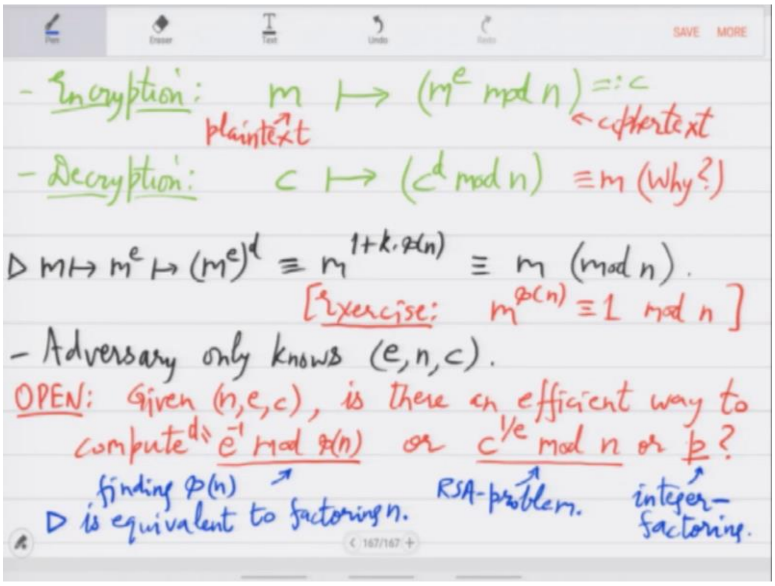
So what is this mechanism why will it need prime numbers so let us do that. This is again a public key cryptosystem it is also called asymmetric cryptosystem because for example when you connect to your bank. Your bank will tell you something will publish the public key but will hide the private key and that is enough for you to encrypt your message and send such that no adversary can break it at the same time bank and decrypt it.

So both Primality and the opposite of this which is or the search version of Primality is integer factoring. So both Primality and integer factoring appear in a cryptosystem designed by Rivest, Shamir and Adelman in 1977. So again we will show the encryption decryption before that the pre-processing is. So preprocessing steps are carefully choose p and q primes these will be very large primes and there will be also some other conditions which you need for the system to be secure.

n will be a number product of p q and phi n is the Euler torsion function p - 1 times q – 1. Also carefully choose an e between 1 to phi n and compute e inverse mod phi n. So since e is co prime to phi n the inverse will exist. So we chose p q and we chose e right p q e are the 3 choices and based on this your bank has public key and private key. So they are as follows. Public key is e and n private key is d on of course p and q.

So bank will not; will never tell anybody what is d and what is p and q the factorization is hidden. Public key is the number and this e for exponentiation, so, based on this now we have encryption decryption.

(Refer Slide Time: 21:30)



- Encryption: $m \mapsto (m^e \bmod n) =: c$
  plaintext ← ciphertext
- Decryption: $c \mapsto (c^d \bmod n) \equiv m$ (Why?)

▷ $m \mapsto m^e \mapsto (m^e)^d \equiv m^{1+k\cdot\phi(n)} \equiv m \pmod n$.

[Exercise: $m^{\phi(n)} \equiv 1 \bmod n$]

- Adversary only knows (e,n,c).

OPEN: Given (n,e,c), is there an efficient way to compute $d := \underline{e^{-1} \bmod \phi(n)}$ or $\underline{c^{1/e} \bmod n}$ or $\underline{p}$?

finding $\phi(n)$ ▷ is equivalent to factoring n. RSA-problem. integer-factoring.

Encryption is suppose m is the message that you want to send to the bank view it as a binary string which you think of as a number representing a number modern and then just exponentiated by e that is the ciphertext and send the ciphertext. So this is called plain text or the message this is the ciphertext. Decryption is on the ciphertext bank will apply c to the d and magically this will be message why?

Why is this original message is the question right so let us now prove that. So, m is going to m to the e which is going to m to the e d. What is e d? Remember d is e inverse mod phi n so this is 1 + k times phi n for some integer k. What is m to the phi n? Mod n this is 1, so this is m mod n. So prove this as an exercise that m to the phi n is 1 mod n. So because of this bank knows m. Now the question to ask is an adversary listening in the middle listening to you and the bank.

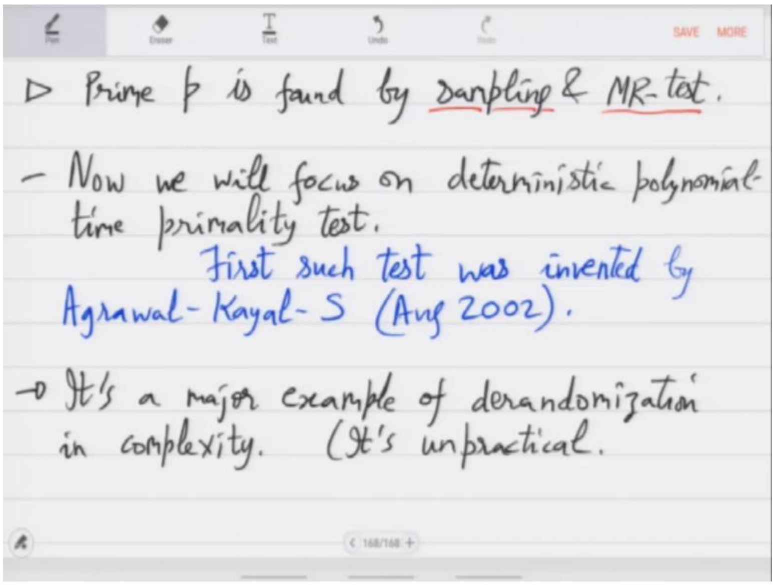
The message you send m the c that you send to the bank. So adversary has e n and c so from that so only knows e n c. Now using this how will adversary find p and q that is the question of factoring in n how will adversary find m? Well that is the question of computing c to the 1 by e right these are the 2 natural attacks or algorithms adversary will try to find both are as conjectured to be hard.

So what is open is? Given n e c is there an efficient way to compute well d what is d? e inverse mod phi n. Is there an efficient way to compute d that would immediately tell the adversary m using the decryption strategy or c raised to 1 by e mod n right or just factor n. So any of these things this, this or p could any of these be computed by the adversary efficiently.

So till now there is no known way it is an open question. It is a very important open question. So we can give names to these this is called e inverse mod phi n not this, this one series to 1 by e mod n this question is called the RSA problem and this is the factoring n problem. And this is the question of finding fn and it is known that this is equivalent to factoring. So basically the question is solve RSA problem 8th root finding or factor n.

It is not known whether the 2 are related yes so this is the state of the art. This is the best or most used crypto system on the internet. And in already in the pre-processing step you need good primes otherwise this thing will not work. So finding primes reduces to testing them. So that is where Primality testing enters.

(Refer Slide Time: 29:42)



▷ Prime p is found by sampling & MR-test.

– Now we will focus on deterministic polynomial time primality test.

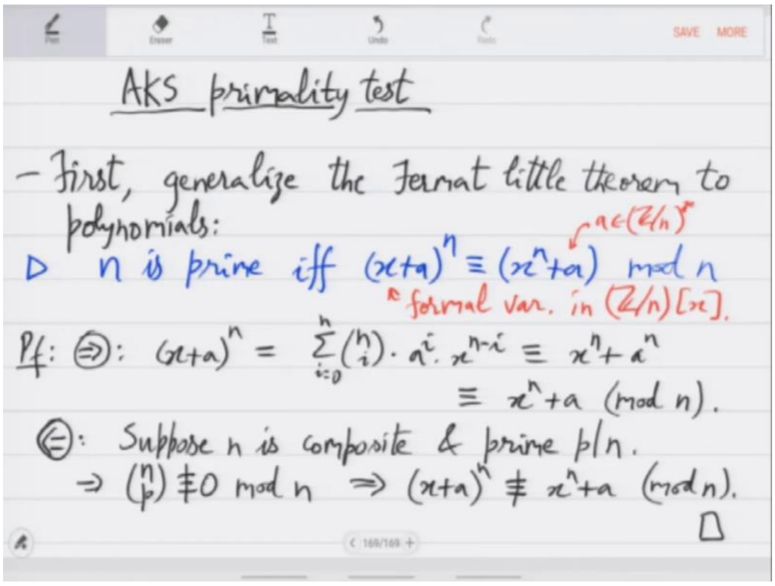
First such test was invented by Agrawal-Kayal-S (Aug 2002).

→ It's a major example of derandomization in complexity. (It's unpractical.

So prime speak p; this prime p is found by random sampling and Primality testing. So you basically if you want l digit prime p you sample randomly an l digit number and then do Miller Rabin test with high probability all this will work and you will get a prime number. If it fails you repeat so on. This can be shown to work well. So we will do one more thing in Primality testing and that is d randomization.

So now we will focus on deterministic polynomial time Primality test. So this was an open question till 2002. In 2002 we solved it, so the first test was invented by Agarwal-Kayal and myself here in august 2002. So I will quickly go through this test. It will be much slower than the Miller Robin or Solvay-Stassen test but it will really be deterministic and polynomial time and it will not use GRH unconditional result. So it is a major example of derandomization but it is not practical.

(Refer Slide Time: 33:05)



AKS primality test

- First, generalize the Fermat little theorem to polynomials:

▷ n is prime iff $(x+a)^n \equiv (x^n+a) \bmod n$ $a \in (\mathbb{Z}/n)^*$

formal var. in $(\mathbb{Z}/n)[x]$.

Pf: (⇒): $(x+a)^n = \sum_{i=0}^{n} \binom{n}{i} \cdot a^i \cdot x^{n-i} \equiv x^n + a^n$

$\equiv x^n + a \pmod n$.

(⇐): Suppose n is composite & prime $p|n$.

$\Rightarrow \binom{n}{p} \not\equiv 0 \bmod n \Rightarrow (x+a)^n \not\equiv x^n + a \pmod n$. □

So let us do the e case Primality test in the remaining time. So first we generalize the Fermat identity to polynomials. So Fermat entities remember was for a prime n a raise to n is what does it mean for polynomials. So we first make it slightly more general so we say that n is prime if and only if x + a to the n is x to the n + a mod n and what is x? x is not a number it is a formal variable in the polynomial ring mod n.

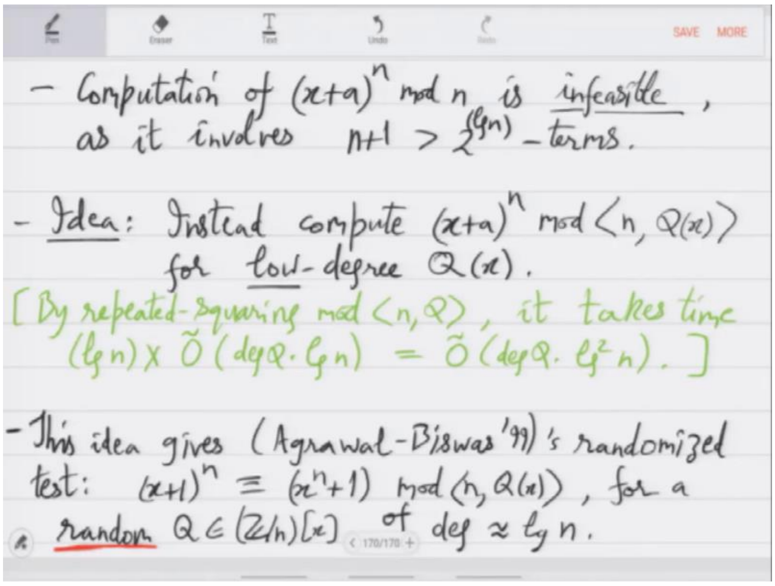
So if x + a to the n is x to the n + a then n is prime and if n is prime then this is true a is just a number co prime to n you can even take it one take it to b 1 is just a number incident star x is a formal variable. This is very easy to prove once stated like this. So in the forward direction what you do you write the binomial expansion x + a to the n is n choose i or a to the i x to the n - i and then you observe that mod n since n is prime.

Mod n all these n choose i's are zero except 2 the first and the last. So i equal to 0 which is x to the n + e to the n which is again x to the n + a because of Fermat’s little theorem on numbers. So this is why x + a to the n is x to the n + a. In the converse if this identity holds for some a and for contradiction suppose n is composite and say prime p divides it then you can show that n choose p is not zero mod n.

This you can show simply by the definition of a binomial and choose p essentially p is dividing in the denominator n times n - 1 times n - p + 1 right. So n would not be able to divide this number and because of that x + a to the n binomial expansion and choose p will survive. So that finishes the contra positive of the converse hence the converse is done. So if n is composite this identity is violated if n is prime it holds.

So this is a very good criterion unlike for mass little theorem that a raise to n equal to e we knew even if it is true for all a it does not mean n is prime because of Carmichael numbers but here indeed it is a criteria; criterion for Primality. Now can it be used in practice the problem is x + a to the n has n + 1 terms n + 1 is huge when n is huge.

(Refer Slide Time: 39:05)



- Computation of $(x+a)^n$ mod n is infeasible, as it involves $n+1 > 2^{(\lg n)}$ – terms.

- Idea: Instead compute $(x+a)^n$ mod $\langle n, Q(x)\rangle$ for low-degree $Q(x)$.

[By repeated-squaring mod $\langle n, Q\rangle$, it takes time $(\lg n) \times \tilde{O}(\deg Q \cdot \lg n) = \tilde{O}(\deg Q \cdot \lg^2 n)$.]

- This idea gives (Agrawal-Biswas '99)'s randomized test: $(x+1)^n \equiv (x^n+1)$ mod $\langle n, Q(x)\rangle$, for a random $Q \in (\mathbb{Z}/n)[x]$ of deg $\approx \lg n$.

So it is not really practical at this point. So computation of x + a to the n mod n is infeasible as it involves n + 1 terms, n + 1 is more than 2 raised to log n. Exponentially many terms are there so you cannot really compute all of them. So the first breakthrough idea is compute an approximation of this instead of the whole thing. So instead compute x + a to the n mod n, Q x for low degree Q.

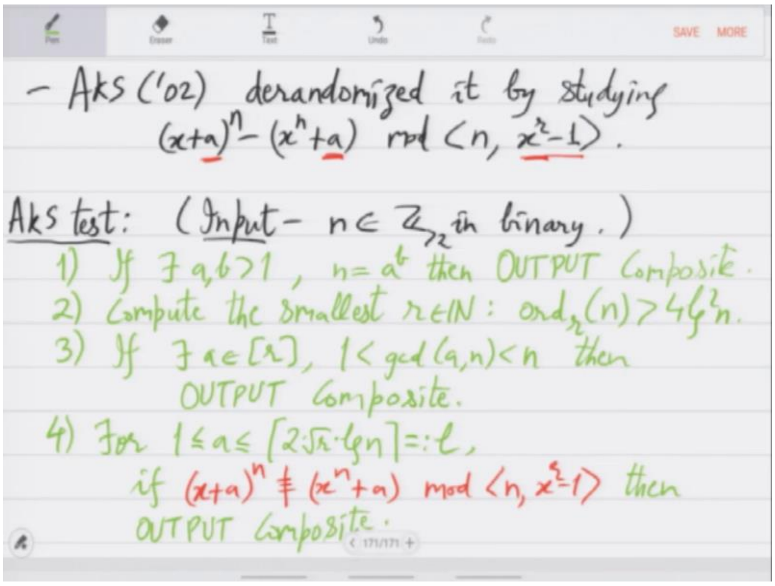
So Q is not very high degree and how do you compute x + a to the n it still has many terms well you can now use repeated squaring. Since you do not know the; do not want the whole expansion of x + a to the n but you only want the remainder you can compute this by repeated squaring. So by repeated squaring mod n, Q it takes time. So you will be doing squaring log n times and each time you are just multiplying 2 things mod n,Q x what is the bit size?

Bit size is degree of Q times log n so everything can be done in o tilde of that. So this is the time complexity. So in this much time you can compute x + a to the n mod n, Q. If degree of

Q is small this is polynomial time small means polynomial in log n. So this actually works it gives a different and a new randomized Primality test. So this idea gives the Agarwal Biswas randomized Primality test.

So what it does is that just check x + 1 to the n is it the same as x + x to the n + 1 mod n, Q x for a random Q x of degree around log n. So that is the slow degree Q you can pick it randomly it is a randomized test. If n passes it you declare n is prime. If it does not pass then obviously n is composite. So that is the new randomized Primality test by Agarwal Biswas. This is what we derandomized.

(Refer Slide Time: 44:00)



- AKS ('02) derandomized it by studying $(x+a)^n - (x^n+a) \bmod \langle n, x^r-1\rangle$.

AKS test: (Input- $n \in \mathbb{Z}_{\geq 2}$ in binary.)

1) If $\exists a,b>1$, $n=a^b$ then OUTPUT Composite.
2) Compute the smallest $r \in \mathbb{N}$: $\text{ord}_r(n) > 4\lg^2 n$.
3) If $\exists a \in [r]$, $1 < \gcd(a,n) < n$ then OUTPUT Composite.
4) For $1 \leq a \leq \lfloor 2\sqrt{r}\cdot\lg n\rfloor =: \ell$, if $(x+a)^n \not\equiv (x^n+a) \bmod \langle n, x^r-1\rangle$ then OUTPUT Composite.

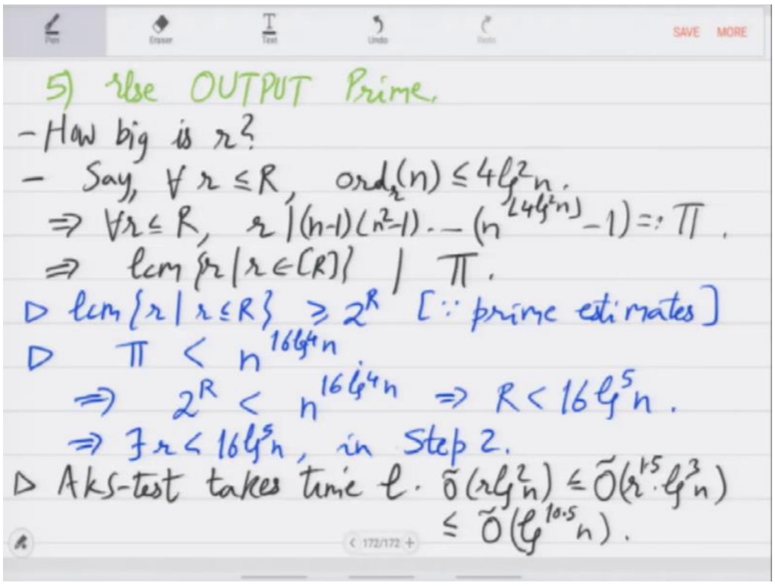
So we de randomized it by studying x + a to the n - x + n x to the n + a modulo n, a very specific Q which is cycloatomic polynomial we use this. So again 2 changes one is we are using a second is we are using specific queue not a random queue. So let us first look at the test because it is so simple and then look at the detailed proof. Input as always is a number n in binary. So step 1 as always is if there exist a b n is e to the b and is a perfect power then output composite.

Otherwise so the motivation for this second step will be clear later it basically fixes the r what are to use compute the smallest r such that the multiplicative order of n mod r is large. It is around quadratic and then check for numbers 1 to r whether they factor n so that the gcd of a n factors n then obviously you are done. Now comes the bulk of the proof bulk of the of of this algorithm it will check x + a to the n identity for all the a's .

Again something unmotivated at this point 2 square root of log n these many is let us call this l. So if the identity holds if this identity holds well or the let us say the opposite if somewhere it fails then that is a certificate that is a proof that n is composite if this fails then output composite. And if you test for all these a's it never failed then you declare prime that is the last step.

So this is the algorithm its basically step 4 the identity being checked for many years but this number of phase depends on r how big is r let us first do that. Let us first estimate that.

(Refer Slide Time: 49:26)



5) Else OUTPUT Prime.

- How big is r?

- Say, $\forall r \leq R$, $ord_r(n) \leq 4\lg^2 n$.

$\Rightarrow \forall r \leq R$, $r \mid (n-1)(n^2-1) \ldots (n^{\lfloor 4\lg^2 n \rfloor}-1) =: \Pi$.

$\Rightarrow$ $lcm\{r \mid r \in [R]\} \mid \Pi$.

▷ $lcm\{r \mid r \leq R\} \geq 2^R$ [$\because$ prime estimates]

▷ $\Pi < n^{16\lg^4 n}$.

$\Rightarrow$ $2^R < n^{16\lg^4 n}$ $\Rightarrow$ $R < 16\lg^5 n$.

$\Rightarrow \exists r < 16\lg^5 n$, in Step 2.

▷ Aks-test takes time $\ell \cdot \tilde{O}(r\lg^2 n) \leq \tilde{O}(r^{1.5} \cdot \lg^3 n)$

$\leq \tilde{O}(\lg^{10.5} n)$.

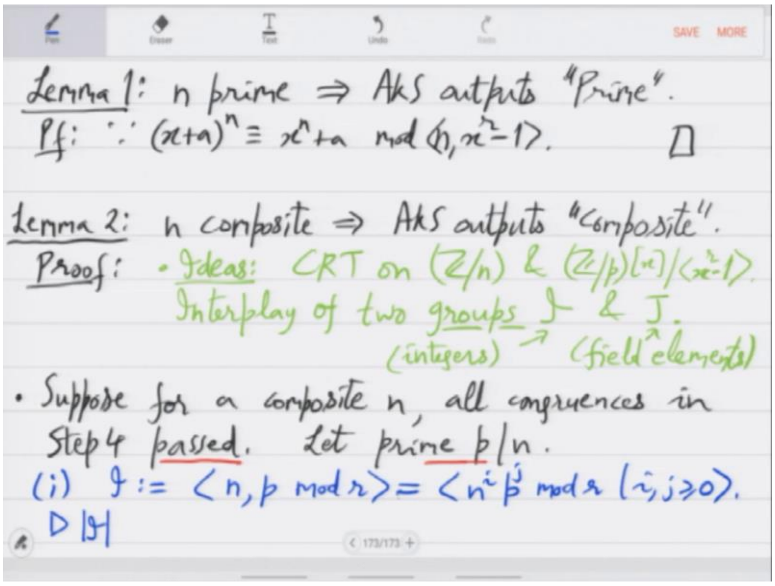
So c for all r less than equal to big R order of n mod r is at most 4 log square and it never exceeded 4 log square n then so what does order of n mod r being small mean it means that if you look at this product n - 1 n square - 1 dot dot n to the 4 log square n - 1 this product is divisible by r. In fact 1 of the factors is divisible by r wherever whatever the order was. So for all the r's up to bigger there is this divisibility condition.

Let us call this product pi so this means that the lcm of all the numbers divides pi how big is it is the lcm of numbers 1 to bigger. So lcm of these numbers this is at least 2 raise to r you can check this by distribution of primes and how big is pi? Well pi is a product of these numbers which are up to n to the 4 log square n. So this is at most n to the 16 log 4n at least smaller than that.

So which will mean that if the lcm divides it then pi has to be bigger than this. So which means r has to be less than 16 log 5 n right so small r cannot exceed this. So this means that there is an r smaller than 16 log n 5 for which step 2 passes. So not everything from 1 to 16 log in to the 5 can fail in step 2 so you get a small r and from that what you get is time complexity. So a case test takes time number of times you will test is l times o tilde r log n square because the degree of this modulus is r.

So r log n square is the time for 1 test so you get l times this l again is around square root r log n right. So this is o tilde r to the 1.5 times log n cube and now you plug in log in to the 5 here so you will get login to the 10.5. So this it is a very slow algorithm but nevertheless it is polynomial time let us now analyze why the answer is always correct there are no random bits involved.

(Refer Slide Time: 54:48)



Lemma 1: n prime ⇒ AKS outputs "Prime".
Pf: ∵ $(x+a)^n \equiv x^n + a \bmod \langle n, x^r - 1 \rangle$. □

Lemma 2: n composite ⇒ AKS outputs "Composite".
Proof: • Ideas: CRT on $(\mathbb{Z}/n)$ & $(\mathbb{Z}/p)[x]/\langle x^r - 1 \rangle$.
Interplay of two groups $\mathcal{G}$ & $\mathcal{J}$.
(integers) (field elements)

• Suppose for a composite n, all congruences in Step 4 passed. Let prime $p|n$.
(i) $\mathcal{G} := \langle n, p \bmod r \rangle = \langle n^i p^j \bmod r \mid i,j \geq 0 \rangle$.
▷ $|\mathcal{G}|$

So first is simple observation that if n is prime then e case outputs prime. Well because if you look at the algorithm only place it can output composite is if n is a perfect power which it is not if n factors in step 3 which it will not or in step 4 this congruence fails which it will not right. So the only possibility is prime. What happens when n is composite that is the main challenge. So this will need some time to prove.

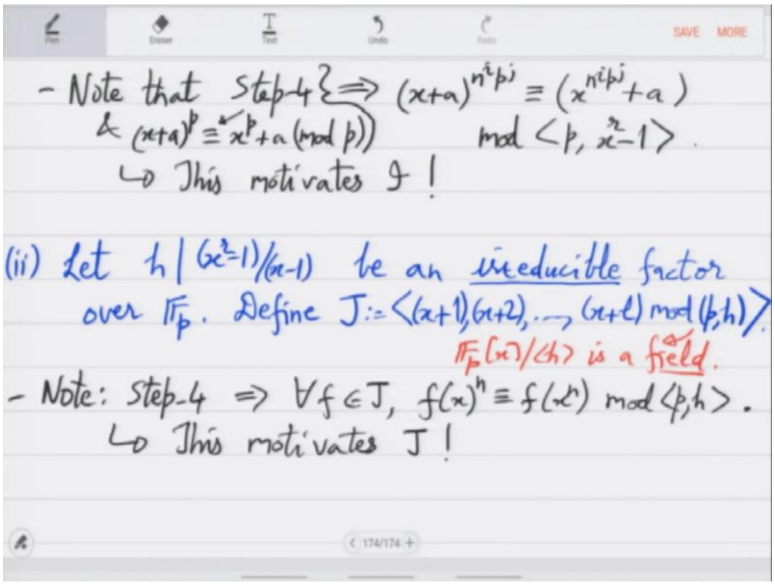
That is the proof here is where all the techniques lie the algorithm is designed so that this proof works. So let me state the idea in advance the ideas involved are obviously you do Chinese remainder theorem on z mod n but also on z mod p so z mod. So remember that you are going modulo both n and x to the r - 1 so you do Chinese remaindering on n but you also do it on x to the r - 1 double k double Chinese remaindering.

And second the very technical thing is we will be working with 2 multiplicative groups one I will call i other i will call j, i will be group of numbers j will be group of field elements. So these are the 2 groups which will help us finish the proof. So let us do that see how this works so for the sake of contradiction it is approved by contradiction suppose for a composite n all congruences in step 4 pass.

All the congruences hold and let prime p be a factor. So now we will see what information can we deduce. So as promised first we will look at multiplicative group i of numbers what are these numbers? So i is defined as the group generated by n and p mod r and note that the size of this group well since it contains n n square. So what does it contain? It contains n to the i p to the j mod r for all i j right that is what these are the elements of the group and to the i p to the j.

So clearly it has at least as many elements as is the order of n. So the size of this group is at least order of n mod r which by the algorithm is at least 4 log square n. Let me make a remark about why this group i is important what is the significance of this.

(Refer Slide Time: 1:01:04)



- Note that Step-4 & $(x+a)^p \equiv x^p + a \pmod p$ $\Rightarrow$ $(x+a)^{n^i p^j} \equiv (x^{n^i p^j} + a)$ mod $\langle p, x^r - 1 \rangle$.

↳ This motivates $\mathcal{I}$!

(ii) Let $h \mid (x^r-1)/(x-1)$ be an irreducible factor over $\mathbb{F}_p$. Define $J := \langle (x+1), (x+2), \ldots, (x+\ell) \bmod (p,h) \rangle$.

$\mathbb{F}_p[x]/\langle h \rangle$ is a field.

- Note: Step-4 $\Rightarrow$ $\forall f \in J$, $f(x)^n \equiv f(x^n) \bmod \langle p, h \rangle$.

↳ This motivates J!

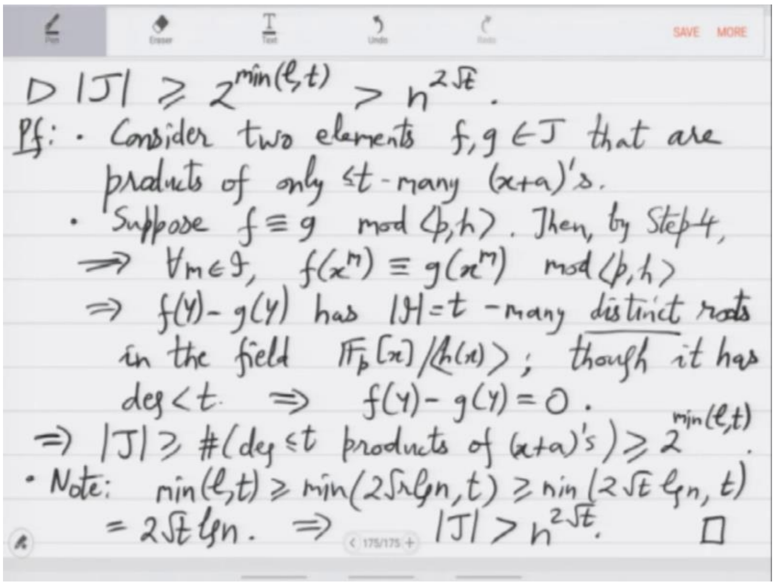
So note that step 4 implies that x + a to the n to the i p raised to j is the same as x to the n to the i p raised to j + a modulo p, x to the r – 1. So step 4 and the property that x + a to the p is x to the p + a mod p both these things right. So step 4 means that x + a to the n is x to the n + a mod n, so p, x to the r - 1. Moreover since we have this identity x + a to the p equal to x to the p + a mod p you can now apply both you can apply n in the exponent you can apply p in the exponent.

And you can actually keep deducing all these congruences for all i j this motivates the definition of i. These are numbers of interest next we do the subgroup or the group j. So let each be an irreducible factor mod p. Now define j to be the group generated by all these x + a's that we tested for. Fine so x + 1 x + 2 dot dot x + l multiply them as much as possible and then take remainder modulo this ideal p, h this is a subgroup of the finite field. So f p x mod h is a field.

So the second subgroup j is sub group of this field except zero again what is the rationale? So note that step 4 implies that for all elements f in j f x to the n is f of x to the n mod p, h why is that well because it is 2 for x + 1 by step 4 congruence what is step 4 congruence it is this x + a to the n x + 1 to the n congruent to x to the n + a x to the n + 1 mod p, h right p divides and h divides x to the r – 1.

So that congruence is true for f equal to x + 1 it is true for x + 2 but then you can also multiply these 2 and then you get this congruence to be true for x + 1 times x + 2 and so on. You can multiply as many times as you like. So this motivates J so both I and J are motivated we have tested for them in the algorithm. And let us talk about now the size of J.

(Refer Slide Time: 1:06:56)



▷ $|J| \geq 2^{\min(\ell,t)} > n^{2\sqrt{t}}$.

Pf: • Consider two elements $f, g \in J$ that are products of only $\leq t$-many $(x+a)$'s.

• Suppose $f \equiv g \bmod \langle p,h \rangle$. Then, by Step4,

$\Rightarrow \forall m \in I,\ f(x^m) \equiv g(x^m) \bmod \langle p,h \rangle$

$\Rightarrow f(Y) - g(Y)$ has $|I| = t$-many distinct roots in the field $\mathbb{F}_p[x]/\langle h(x) \rangle$; though it has deg $< t$. $\Rightarrow f(Y) - g(Y) = 0$.

$\Rightarrow |J| \geq \#(\deg \leq t$ products of $(x+a)$'s$) \geq 2^{\min(\ell,t)}$.

• Note: $\min(\ell,t) \geq \min(2\sqrt{r}\lg n, t) \geq \min(2\sqrt{t}\lg n, t) = 2\sqrt{t}\lg n$. $\Rightarrow |J| > n^{2\sqrt{t}}$. □

We I claim that this is at least 2 raise to l, t exponential in the number of tests you have done its actually at least n raised to 2 square root t why is that well you just look at the number of subsets of 1 to l. So number of subsets is 2 raised to l right so that is what roughly you will get formally what you do is consider 2 elements f, g in j that are products of only t many x + a’s basically pick t or less x + a's and then multiply them.

Look at 2 such products for 2 different ways of picking I claim that they are different suppose not. Suppose f and g are not different in the field then by the step 4 test so f x m is g of x m. So by step 4 you get f x to the m is equal to g x to the m which is f of x to the m and remember that all these x to the m's are different so this means that f of y - g of y formal polynomial has size of i that is t many distinct roots in the finite field f p x mod h x these many distinct roots are there for this polynomial f i - g y. Well but what is the degree of f i - g y it is less than t which could only mean that f and g is the same.

So if you multiply x + a is in 2 different ways t many x + is in 2 different ways you get different products which gives you the a lower bound on the size of J. So number of degree less than equal to t so this is this is at least 2 raised to min of l and t as claimed above. Well this is not finished yet we have to prove the second inequality also so this means that let us consider min of lt.

So min of lt is at least l is 2 square root r log n and t was shown to be well let us keep t for now. Remember that t is less than equal to r because t was the size of the group mod r right. So this is min of 2 square root t log n, t what is the min of these 2 numbers these 2 functions well we have taken t to be more than 4 log square n did not we yes t is at least 4 log square n so in that range in that range min will be the first function.

So you get 2 square root t log n that is the mean which gives you the second inequality. So you get that J is at least n raised to 2 square root t. So we have shown these 2 lower bounds we have shown that j is a large group of finite field elements and we have shown that i is a large group of numbers. So based on this we will deduce the contradiction the next class stop here.