Lecture 3A: RSA

UC Berkeley EECS 70
Summer 2022
Tarang Srivastava

Announcements!

- Read the Weekly Post
- **HW 3** and **Vitamin 3** have been released, due **Thursday** (grace period Fri)
- HW 3 covers last Wednesday, Thursday and Today's lecture
- Any topic that's out of scope in this lecture will be in Orange.
  - You are not responsible for these topics, they're just here to give context
  - These topics will be covered in CS170 and CS161
- In this lecture, we will use small prime numbers as examples but in implementation we use large prime numbers (256 bits $\approx 10^{77}$ or more).

Alice and Bob

Alice and Bob wish to send messages to each other **privately**.
Eve is able to intercept and read the messages.
How can Alice and Bob **encrypt** their messages, so even if Eve intercepts them she cannot understand them (i.e. **decrypt**).

Using a Codebook

How can Alice and Bob **encrypt** their messages, so even if Eve intercepts them she cannot understand them (i.e. **decrypt**).

Public Key Cryptography

Alice generates a **Public Key** (K), and a corresponding **Private Key** (k).
The public key K is known to everyone (including Eve), the private key k is known only to Alice.

Anyone can encode their message using the public key, and send it to Alice.
Only Alice knows the private key, so only she can decrypt the messages sent to her.

RSA

<u>Setting up a Public Key</u>

Pick two large primes p and q. Let $N = pq$

Choose an e that is coprime to the product $(p-1)(q-1)$

Compute the private key $d = e^{-1} \ (mod\ (p-1)(q-1))$.

Announce to the world the public key: $K = (N, e)$

<u>Encrypting Messages</u>

Let x be your message. $E(.)$ is the encryption function. Send $E(x) = x^e \ (mod\ N)$.

<u>Decrypting Messages</u>

Let y be the encrypted message. D(.) is the decryption function. $D(y) = y^d \ (mod\ N)$.

<u>Why does this work?</u>

Decrypting an encrypted message returns original message. $D(E(x)) = x$

Summary Questions

Pick two large primes p and q. Let $N = pq$
Choose an e that is coprime to the product $(p-1)(q-1)$
Compute private key $k = d = e^{-1} \pmod{(p-1)(q-1)}$.
Announce to the world: $K = (N, e)$
Encryption: $E(x) = x^e \pmod{N}$.
Decryption: $D(y) = y^d \pmod{N}$.

	N	e	p and q	d	Private Key	Public Key	Encryption Function	Decryption Function	x	y
Who knows										
Definition										

RSA Example

Pick two large primes p and q. Let $N = pq$
Choose an e that is coprime to the product $(p-1)(q-1)$
Compute private key $k = d = e^{-1}\ (mod\ (p-1)(q-1))$.
Announce to the world: $K = (N, e)$
Encryption: $E(x) = x^e\ (mod\ N)$.
Decryption: $D(y) = y^d\ (mod\ N)$.

Alice Setting Up Public Key

Bob Encrypting Message

Alice Decrypting Message

Why does encryption/decryption work?

Thm: For every x in {0, 1, ..., N-1}, $\boldsymbol{(x^e)^d \equiv x \ (mod\ N).}$ (i.e. D(E(x)) = x)
Proof:

Why can't Eve reconstruct the Private Key?

Idea: $d = e^{-1} \ (mod\ (p-1)(q-1))$, but Eve knows e so why can't she just find the inverse?

Why can't Eve then figure out p and q?

Eve knows N so why can't she figure out p and q using that?

We showed that every number has a prime factorization.
That is, given a natural number n there exist a unique set of primes such that n is equal to their product.

Finding this unique set of primes is **hard**.

What does it mean for a problem to be hard?
In this class, we will say that if the best solution is as good as guess-and-check it is hard.
To find the prime factorization you would have to try every factor for that number.

Is there a faster algorithm to find the factorization?
Unsolved Problem. It is possible with Quantum Computer.

Why can't Eve just take the log?

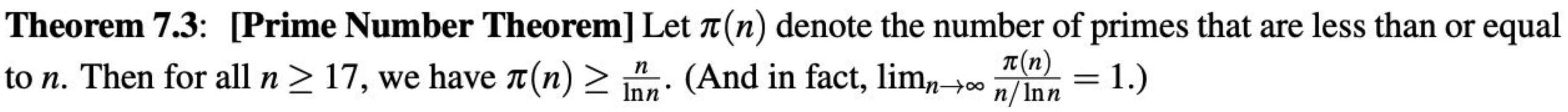
Eve knows the public key (N, e). Eve then encrypts some message $y = x^e \ (mod\ N)$.
Then, the decryption is

$$x = y^d \ (mod\ N)$$

So, why can't Eve just do log_y on both sides to **leak *d*** the private key?

This is called the discrete-log problem and it is **hard**. There is no known efficient solution for this problem.
Examples:

How easy is it to find large primes?



Theorem 7.3: [Prime Number Theorem] Let $\pi(n)$ denote the number of primes that are less than or equal to n. Then for all $n \geq 17$, we have $\pi(n) \geq \frac{n}{\ln n}$. (And in fact, $\lim_{n\to\infty} \frac{\pi(n)}{n/\ln n} = 1$.)

If we want a 512-bit prime number
Theorem says there is roughly 1 prime number every 355 numbers.
For 1024-bit numbers there's a prime every 710.
Just try random numbers and you will eventually find a prime number

You can efficiently check if a number is prime using the Miller-Rabin test.

Can Even find a match using the encryption function?

If Bob encrypts some message $y = x^e$ $(mod$ N). Then, could Eve just plug in x' into the encryption function to find a match?

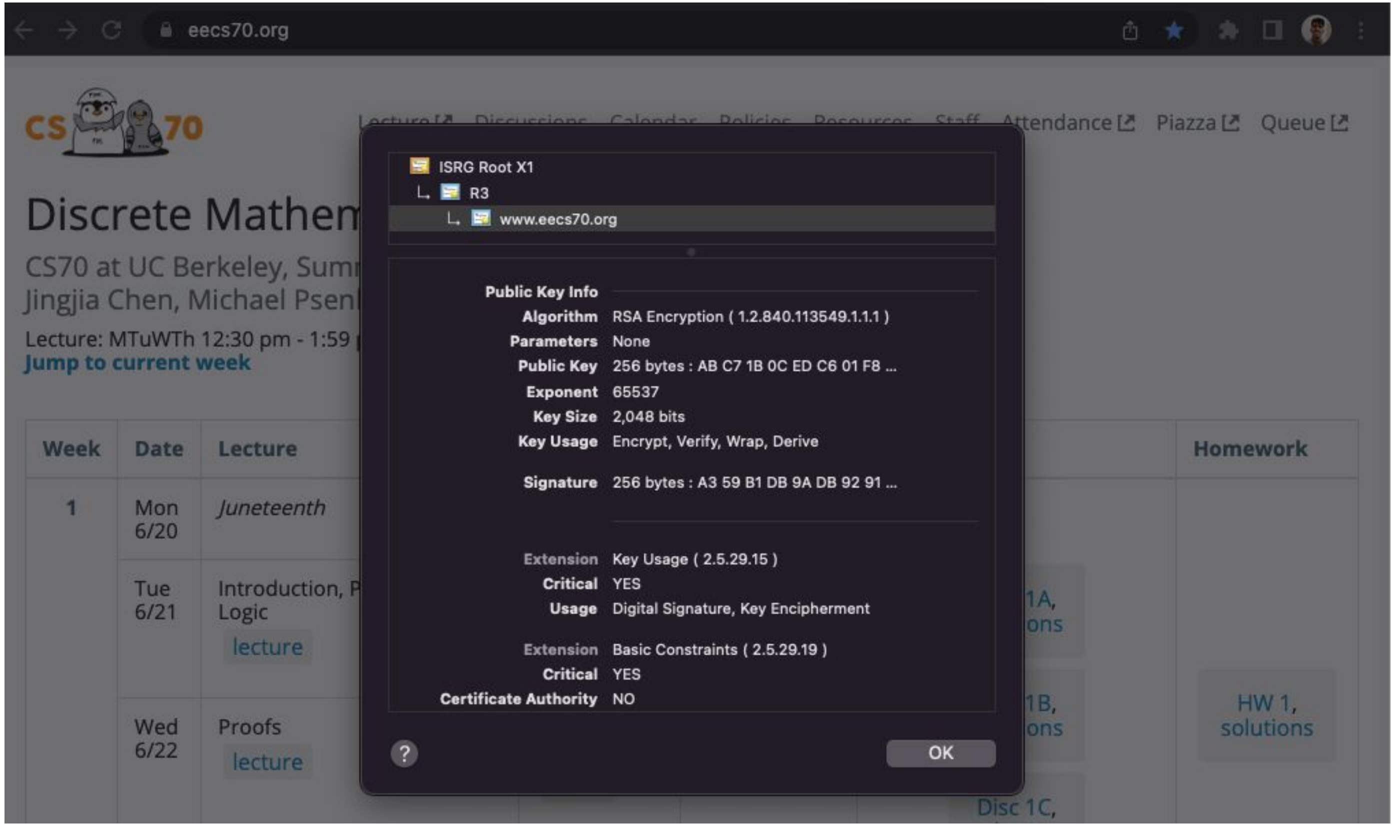
No! For 256-bit prime numbers that is 2^{256} it would take you 37 times the age of the universe to arrive at a guess for a $x' = x$.

In Practice Sending Same Message Twice

Notice that since all the numbers are fixed, if you send the same message twice it will be encrypted the same way.

In practice, usually append a counter to the message so each message is unique.

You use some derivation of RSA every day

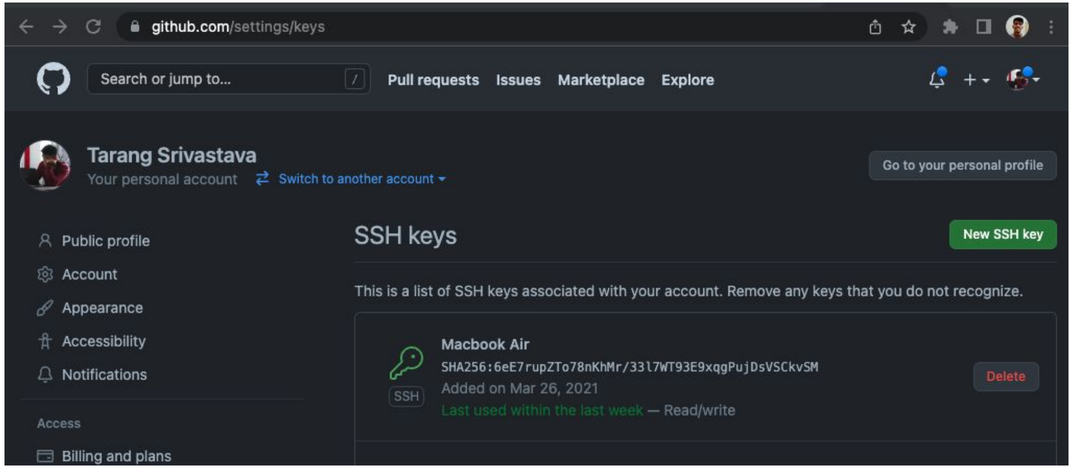


eecs70.org
CS70
Lecture Discussions Calendar Policies Resources Staff Attendance Piazza Queue
Discrete Mathem
CS70 at UC Berkeley, Sum
Jingjia Chen, Michael Psen
Lecture: MTuWTh 12:30 pm - 1:59
Jump to current week
Week Date Lecture Homework
1 Mon 6/20 Juneteenth
Tue 6/21 Introduction, P Logic lecture
Wed 6/22 Proofs lecture
1A, ons
1B, ons
Disc 1C,
HW 1, solutions

ISRG Root X1
└ R3
└ www.eecs70.org

Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Parameters	None
Public Key	256 bytes : AB C7 1B 0C ED C6 01 F8 ...
Exponent	65537
Key Size	2,048 bits
Key Usage	Encrypt, Verify, Wrap, Derive
Signature	256 bytes : A3 59 B1 DB 9A DB 92 91 ...
Extension	Key Usage (2.5.29.15)
Critical	YES
Usage	Digital Signature, Key Encipherment
Extension	Basic Constraints (2.5.29.19)
Critical	YES
Certificate Authority	NO

You use some derivation of RSA every day

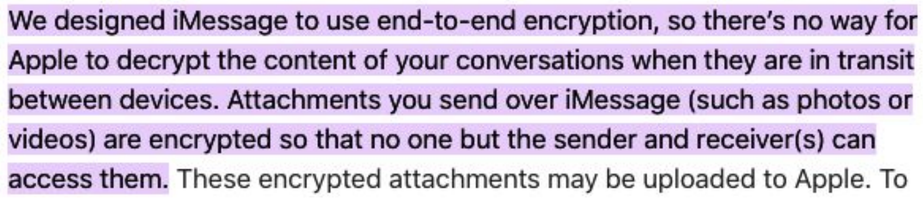


Tarang Srivastava
Your personal account ⇄ Switch to another account

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Macbook Air
SHA256:6eE7rupZTo78nKhMr/33l7WT93E9xqgPujDsVSCkvSM
Added on Mar 26, 2021
Last used within the last week — Read/write



We designed iMessage to use end-to-end encryption, so there's no way for Apple to decrypt the content of your conversations when they are in transit between devices. Attachments you send over iMessage (such as photos or videos) are encrypted so that no one but the sender and receiver(s) can access them. These encrypted attachments may be uploaded to Apple. To

A little story to end...

In 1977, Rivest, Shamir and Adleman publish the RSA algorithm you learned today.

Later that year, the British Intelligence Agency (GCHQ) declassify that they had developed the exact algorithm secretly in 1973.

Why do all this?

- Your company will ask you to make sure their data is secure
- You will want to make sure that your data is secure
- Most importantly, you have a **moral** responsibility to do so