Announcements

- No problem set *due* this week
  Start working on set due next week now
  Problem Set will be posted later today, partners later in the week
- Started posting notes from Monday meetings to onCourse
  If they don't appear in a timely manner, please send me an email reminder!
- Hope to have exams finished by end of the week
- Please fill out Midsemester Evaluation
  Chance to give me feedback
  Link in email and onCourse TR Announcements forum
- Should be able to finish tutorials during scheduled class time
  If in Mon tutorial, then can meet during Wednesday class to finish
  If in Wed tutorial, then start on Monday and finish in person on Wednesday

Overview of RSA: Bob picks $k_{pub} = (n, e)$ and $k_{pr} = d$

Encryption: $y = e_{k_{pub}}(x) = x^e \mod n$

Decryption: $x = d_{k_{pr}}(y) = y^d \mod n$

- Bob picks large primes p and q, then $n = pq$ and $\phi(n) = (p-1)(q-1)$
- Find $k_{pr} = d \equiv e^{-1} \mod \phi(n)$ using Extended Euclidean Algorithm

Overview of RSA: Bob picks $k_{pub} = (n, e)$ and $k_{pr} = d$

Encryption: $y = e_{k_{pub}}(x) = x^e \mod n$ Decryption: $x = d_{k_{pr}}(y) = y^d \mod n$

- Bob picks large primes p and q, then $n = pq$ and $\phi(n) = (p-1)(q-1)$
- Find $k_{pr} = d \equiv e^{-1} \mod \phi(n)$ using Extended Euclidean Algorithm

Key Point:

- $k_{pub} = (n, e)$ is known to the world
- If bad actor could factor $n = pq$, then would know $\phi(n)$ and could find $k_{pr} = d$

Overview of RSA: Bob picks $k_{pub} = (n, e)$ and $k_{pr} = d$

Encryption: $y = e_{k_{pub}}(x) = x^e \mod n$ Decryption: $x = d_{k_{pr}}(y) = y^d \mod n$

- Bob picks large primes p and q, then $n = pq$ and $\phi(n) = (p-1)(q-1)$
- Find $k_{pr} = d \equiv e^{-1} \mod \phi(n)$ using Extended Euclidean Algorithm

Key Point:

- $k_{pub} = (n, e)$ is known to the world
- If bad actor could factor $n = pq$, then would know $\phi(n)$ and could find $k_{pr} = d$

The security of RSA depends upon $\phi(n)$ being private to Bob, and thus, the difficulty of factoring $n = pq$

Why RSA is secure with *large* values of n

Question: If Oscar knows n is a 2048-bit number, then suspects p and q are both 1024-bit primes. Why doesn't Oscar build a list of all 1024-bit primes ahead of time and check if they divide n?

Why RSA is secure with *large* values of n

Question: If Oscar knows n is a 2048-bit number, then suspects p and q are both 1024-bit primes. Why doesn't Oscar build a list of all 1024-bit primes ahead of time and check if they divide n?

Prime Number Theorem: If $\pi(n)$ denotes the number of prime numbers less than n, then for large values of n

$$\pi(n) \sim \frac{n}{\ln(n)}$$

Why RSA is secure with *large* values of n

Question: If Oscar knows n is a 2048-bit number, then suspects p and q are both 1024-bit primes. Why doesn't Oscar build a list of all 1024-bit primes ahead of time and check if they divide n?

Prime Number Theorem: If $\pi(n)$ denotes the number of prime numbers less than n, then for large values of n

$$\pi(n) \sim \frac{n}{\ln(n)}$$

$$\begin{aligned}\Rightarrow \#\text{ 1024-bit primes} &= \pi(2^{1024}) - \pi(2^{1023}) \\ &\approx \frac{2^{1024}}{\ln(1024)} - \frac{2^{1023}}{\ln(1023)} \approx 1.26 \times 10^{305}\end{aligned}$$

Why RSA is secure with *large* values of n

Question: If Oscar knows n is a 2048-bit number, then suspects p and q are both 1024-bit primes. Why doesn't Oscar build a list of all 1024-bit primes ahead of time and check if they divide n?

Prime Number Theorem: If $\pi(n)$ denotes the number of prime numbers less than n, then for large values of n

$$\pi(n) \sim \frac{n}{\ln(n)}$$

$$\begin{aligned} \Rightarrow \#\text{ 1024-bit primes} &= \pi(2^{1024}) - \pi(2^{1023}) \\ &\approx \frac{2^{1024}}{\ln(1024)} - \frac{2^{1023}}{\ln(1023)} \approx 1.26 \times 10^{305} \end{aligned}$$

There are only $\approx 10^{80}$ particles in the observable universe!

Opportunities for bad action

- **Oscar:** Oscar could be sitting in the middle and intercept all messages
  Could forge Alice's initial message to Bob with own AES key, and Bob would have no way to know if talking to Oscar or Alice

Opportunities for bad action

- **Oscar:** Oscar could be sitting in the middle and intercept all messages
  Could forge Alice's initial message to Bob with own AES key, and Bob would have no way to know if talking to Oscar or Alice

- **Bob:** Bob could use k_{pub} to fake a message from Alice (e.g. "IOU $1000")

Opportunities for bad action

- **Oscar:** Oscar could be sitting in the middle and intercept all messages
  Could forge Alice's initial message to Bob with own AES key, and Bob would have no way to know if talking to Oscar or Alice

- **Bob:** Bob could use k_{pub} to fake a message from Alice (e.g. "IOU $1000")

- **Alice:** Alice could accuse Bob of faking a message ("Did not promise that!")
  A third party (e.g. a judge) would have no way of determining the truth

Motivation for Digital Signatures

- Provide a way for Alice to sign a message to authenticate that it is from them
- Do in a way that no one can duplicate or forge, even Bob

RSA Digital Signatures

For Alice to sign a message x to Bob, Alice uses *their* RSA credentials $k_{pub} = (n, e)$ and $k_pr = d$

- Alice uses *private key* to compute $s \equiv x^d \mod n$
- Sends (x, s) to Bob
- Bob uses Alice's *public key* to verify:

$$x' \equiv s^e \mod n$$

If $x' = x$, then the signature came from Alice