BILL, RECORD LECTURE!!!!
Public Key
Cryptography: RSA
Article Title: Whack a Mole: The new president (of Colombia) calls off talks with a lesser-known leftist insurgent group.

In 2016 FARC, a left-wing insurgent group in Columbia, signed a peace treaty that ended 50 years of conflict. Yeah! The former president of Columbia got the Nobel Peace Prize (the leader of FARC did not – I do not know why). However a more extreme insurgent group, ELN, is still active. Why did FARC negotiate but ELN did not?

Quote:

And the ELN’s strong encryption system has prevented the army from extracting information from seized computers, as it did with FARC.

Caveat: The article did not say what system they used. Oh Well.
**Article Title:** Whack a Mole: The new president (of Colombia) calls off talks with a lesser-known leftist insurgent group.

**Context:** In 2016 FARC, a left-wing insurgent group in Columbia, signed a peace treaty that ended 50 years of conflict **Yeah!**

The former president got the Nobel Peace Prize (the leader of FARC did not – I do not know why). However a more extreme insurgent group, ELN, is still active. Why did FARC negotiate but ELN did not? 

Quote: And the ELN’s strong encryption system has prevented the army from extracting information from seized computers, as it did with FARC. 

Caveat: The article did not say what system they used. **Oh Well.**
Article Title: Whack a Mole: The new president (of Colombia) calls off talks with a lesser-known leftist insurgent group.

Context: In 2016 FARC, a left-wing insurgent group in Columbia, signed a peace treaty that ended 50 years of conflict Yeah! The former president of Columbia got the Nobel Peace Prize (the leader of FARC did not – I do not know why).
Article Title: Whack a Mole: The new president (of Colombia) calls off talks with a lesser-known leftist insurgent group.

Context: In 2016 FARC, a left-wing insurgent group in Columbia, signed a peace treaty that ended 50 years of conflict Yeah!
The former president of Columbia got the Nobel Peace Prize (the leader of FARC did not – I do not know why).
However a more extreme insurgent group, ELN, is still active. Why did FARC negotiate but ELN did not?:

And the ELN's strong encryption system has prevented the army from extracting information from seized computers, as it did with FARC.

Caveat: The article did not say what system they used. Oh Well.
Article Title: Whack a Mole: The new president (of Colombia) calls off talks with a lesser-known leftist insurgent group.

Context: In 2016 FARC, a left-wing insurgent group in Columbia, signed a peace treaty that ended 50 years of conflict Yeah! The former president of Columbia got the Nobel Peace Prize (the leader of FARC did not – I do not know why).

However a more extreme insurgent group, ELN, is still active. Why did FARC negotiate but ELN did not?:

Quote: ... And the ELN’s strong encryption system has prevented the army from extracting information from seized computers, as it did with FARC.
Article Title: Whack a Mole: The new president (of Colombia) calls off talks with a lesser-known leftist insurgent group.

Context: In 2016 FARC, a left-wing insurgent group in Columbia, signed a peace treaty that ended 50 years of conflict Yeah! The former president of Columbia got the Nobel Peace Prize (the leader of FARC did not – I do not know why). However a more extreme insurgent group, ELN, is still active. Why did FARC negotiate but ELN did not?:

Quote: ... And the ELN’s strong encryption system has prevented the army from extracting information from seized computers, as it did with FARC.

Caveat: The article did not say what system they used. Oh Well.
The Academic Code
Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called **public key cryptography**.
The Academic Code

Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called public key cryptography.

When Academics Say:
The Academic Code

Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called public key cryptography.

When Academics Say: ... of great theoretical and practical importance.
The Academic Code

Academics often talk in code that **sounds** like normal speech, so you might not realize it. They talk in public, so this could be called **public key cryptography**.

**When Academics Say:** ... of great theoretical and practical importance.

**They Mean:**
The Academic Code

Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called public key cryptography.

When Academics Say: ... of great theoretical and practical importance.
They Mean: interesting to me.
The Academic Code

Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called public key cryptography.

**When Academics Say:** ... of great theoretical and practical importance.

**They Mean:** interesting to me.

**When Academics Say:**
The Academic Code

Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called public key cryptography.

When Academics Say: ... of great theoretical and practical importance.  
They Mean: interesting to me.

When Academics Say: It has long been known that...
Academics often talk in code that *sounds* like normal speech, so you might not realize it. They talk in public, so this could be called *public key cryptography*.

**When Academics Say:** . . . of great theoretical and practical importance.  
**They Mean:** interesting to me.

**When Academics Say:** It has long been known that . . .  
**They Mean:**
The Academic Code

Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called public key cryptography.

**When Academics Say:** ... of great theoretical and practical importance.

**They Mean:** interesting to me.

**When Academics Say:** It has long been known that...

**They Mean:** I haven’t bothered to look up the original reference.
The Academic Code

Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called public key cryptography.

**When Academics Say:** ... of great theoretical and practical importance.
**They Mean:** interesting to me.

**When Academics Say:** It has long been known that....
**They Mean:** I haven’t bothered to look up the original reference.

**When Academics Say:**
The Academic Code

Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called public key cryptography.

When Academics Say: ... of great theoretical and practical importance.
They Mean: interesting to me.

When Academics Say: It has long been known that ....
They Mean: I haven’t bothered to look up the original reference.

When Academics Say: The proof is left to the reader.
The Academic Code

Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called public key cryptography.

When Academics Say: ... of great theoretical and practical importance.
They Mean: interesting to me.

When Academics Say: It has long been known that....
They Mean: I haven’t bothered to look up the original reference.

When Academics Say: The proof is left to the reader.
They Mean:
The Academic Code

Academics often talk in code that sounds like normal speech, so you might not realize it. They talk in public, so this could be called public key cryptography.

When Academics Say: ... of great theoretical and practical importance.
They Mean: interesting to me.

When Academics Say: It has long been known that ....
They Mean: I haven’t bothered to look up the original reference.

When Academics Say: The proof is left to the reader.
They Mean: Someone smarter than me can surely prove this.
The Academic Code, More Examples

When Academics Say:

When Academics Say: 

The agreement of my theory and the empirical data is Excellent.

They Mean:

The agreement of my theory and the empirical data is Good.

When Academics Say: 

The agreement of my theory and the empirical data is Good.

They Mean:

The agreement of my theory and the empirical data is Non-existent.

When Academics Say: 

It is generally believed that ...

They Mean:

Me and my friends think ...
The Academic Code, More Examples

When Academics Say: The agreement of my theory and the empirical data is Excellent.

They Mean: The agreement of my theory and the empirical data is Good.

When Academics Say: The agreement of my theory and the empirical data is Good.

They Mean: The agreement of my theory and the empirical data is Non-existent.

When Academics Say: It is generally believed that...

They Mean: Me and my friends think...
The Academic Code, More Examples

When Academics Say: The agreement of my theory and the empirical data is Excellent.
They Mean:
The Academic Code, More Examples

When Academics Say: The agreement of my theory and the empirical data is Excellent.

They Mean: The agreement of my theory and the empirical data is Good.
The Academic Code, More Examples

When Academics Say: The agreement of my theory and the empirical data is Excellent.
They Mean: The agreement of my theory and the empirical data is Good.

When Academics Say:
When Academics Say: The agreement of my theory and the empirical data is Excellent.
They Mean: The agreement of my theory and the empirical data is Good.

When Academics Say: The agreement of my theory and the empirical data is Good.
They Mean: The agreement of my theory and the empirical data is Non-existent.
The Academic Code, More Examples

**When Academics Say:** The agreement of my theory and the empirical data is Excellent.
**They Mean:** The agreement of my theory and the empirical data is Good.

**When Academics Say:** The agreement of my theory and the empirical data is Good.
**They Mean:**
The Academic Code, More Examples

**When Academics Say:** The agreement of my theory and the empirical data is Excellent.

**They Mean:** The agreement of my theory and the empirical data is Good.

**When Academics Say:** The agreement of my theory and the empirical data is Good.

**They Mean:** The agreement of my theory and the empirical data is Non-existent.
The Academic Code, More Examples

When Academics Say: The agreement of my theory and the empirical data is Excellent.  
They Mean: The agreement of my theory and the empirical data is Good.

When Academics Say: The agreement of my theory and the empirical data is Good.  
They Mean: The agreement of my theory and the empirical data is Non-existent.

When Academics Say:
The Academic Code, More Examples

When Academics Say: The agreement of my theory and the empirical data is Excellent.
They Mean: The agreement of my theory and the empirical data is Good.

When Academics Say: The agreement of my theory and the empirical data is Good.
They Mean: The agreement of my theory and the empirical data is Non-existent.

When Academics Say: It is generally believed that....
When Academics Say: The agreement of my theory and the empirical data is Excellent.
They Mean: The agreement of my theory and the empirical data is Good.

When Academics Say: The agreement of my theory and the empirical data is Good.
They Mean: The agreement of my theory and the empirical data is Non-existent.

When Academics Say: It is generally believed that. . .
They Mean:
When Academics Say: The agreement of my theory and the empirical data is Excellent.  
They Mean: The agreement of my theory and the empirical data is Good.

When Academics Say: The agreement of my theory and the empirical data is Good.  
They Mean: The agreement of my theory and the empirical data is Non-existent.

When Academics Say: It is generally believed that. . . .  
They Mean: Me and my friends think. . . .
Public Key

Cryptography: RSA
What does RSA Stand For?

RSA stands for Rivest-Shamir-Adelman. They are the ones who came up with this cryptosystem.
What does RSA Stand For?

RSA stands for Rivest-Shamir-Adelman.
What does RSA Stand For?

RSA stands for **Rivest-Shamir-Adelman**.
They are the ones who came up with this cryptosystem.
Recall Fermat’s little Theorem

**Thm** If $p$ is prime and $a \in \mathbb{N}$ then

$$a^p \equiv a \pmod{p}.$$
Slight Variant on Fermat’s Little Theorem

Recall Fermat’s little Theorem

**Thm** If $p$ is prime and $a \in \mathbb{N}$ then

$$a^p \equiv a \pmod{p}.$$ 

We want to divide both sides by $a$ and get $a^{p-1} \equiv 1 \pmod{p}$. 

Not quite right: What if $a \equiv 0 \pmod{p}$? Then not true. Hence:

**Thm** If $p$ is prime and $a \in \mathbb{N}$ and $a \not\equiv 0 \pmod{p}$ then

$$a^{p-1} \equiv 1 \pmod{p}.$$ 

We will refer to both as Fermat’s Little Theorem.
Recall Fermat’s little Theorem

**Thm** If $p$ is prime and $a \in \mathbb{N}$ then

$$a^p \equiv a \pmod{p}.$$ 

We want to divide both sides by $a$ and get $a^{p-1} \equiv 1 \pmod{p}$. Not quite right: What if $a \equiv 0 \pmod{p}$? Then not true. Hence:
Slight Variant on Fermat’s Little Theorem

Recall Fermat’s little Theorem

Thm If $p$ is prime and $a \in \mathbb{N}$ then

$$a^p \equiv a \pmod{p}.$$  

We want to divide both sides by $a$ and get $a^{p-1} \equiv 1 \pmod{p}$. Not quite right: What if $a \equiv 0 \pmod{p}$? Then not true. Hence:

Thm If $p$ is prime and $a \in \mathbb{N}$ and $a \not\equiv 0 \pmod{p}$ then

$$a^{p-1} \equiv 1 \pmod{p}.$$
Recall Fermat’s little Theorem

Thm If $p$ is prime and $a \in \mathbb{N}$ then

$$a^p \equiv a \pmod{p}.$$ 

We want to divide both sides by $a$ and get $a^{p-1} \equiv 1 \pmod{p}$. Not quite right: What if $a \equiv 0 \pmod{p}$? Then not true. Hence:

Thm If $p$ is prime and $a \in \mathbb{N}$ and $a \not\equiv 0 \pmod{p}$ then

$$a^{p-1} \equiv 1 \pmod{p}.$$ 

We will refer to both as Fermat’s Little Theorem.
$11^{999,999,999} \pmod{107}$

Repeated squaring would take $\sim \log(999,999,999) \sim 30$ mults.
Repeated squaring would take $\sim \lg(999,999,999) \sim 30$ mults.
By Fermat’s Little Thm $11^{106} \equiv 1 \pmod{107}$.
$11^{999,999,999} \pmod{107}$

Repeated squaring would take $\sim \lg(999,999,999) \sim 30$ mults. By Fermat's Little Thm $11^{106} \equiv 1 \pmod{107}$. Note $999,999,999 \equiv 27 \pmod{106}$
$11^{999,999,999} \pmod{107}$

Repeated squaring would take $\sim \lg(999,999,999) \sim 30$ mults. By Fermat’s Little Thm $11^{106} \equiv 1 \pmod{107}$. Note $999,999,999 \equiv 27 \pmod{106}$

$$999,999,999 = 106k + 27 \text{ (don’t care what } k \text{ is)}$$
$11^{999,999,999} \pmod{107}$

Repeated squaring would take $\sim \lg(999,999,999) \sim 30$ mults. By Fermat’s Little Thm $11^{106} \equiv 1 \pmod{107}$. Note $999,999,999 \equiv 27 \pmod{106}$

$999,999,999 = 106k + 27$ (don’t care what $k$ is)

$11^{999,999,999} = 11^{106k} \times 11^{27} = (11^{106})^k \times 11^{27} \equiv 1^k 11^{27} \equiv 11^{27} \pmod{107}$
$11^{999,999,999} \pmod{107}$

Repeated squaring would take $\sim \lg(999,999,999) \sim 30$ mults.

By Fermat’s Little Thm $11^{106} \equiv 1 \pmod{107}$. Note $999,999,999 \equiv 27 \pmod{106}$

$$999,999,999 = 106k + 27 \text{ (don’t care what } k \text{ is)}$$

$$11^{999,999,999} = 11^{106k} \times 11^{27} = (11^{106})^k \times 11^{27} \equiv 1^k 11^{27} \equiv 11^{27} \pmod{107}$$

$11^{999,999,999} \equiv 11^{999,999,999} \pmod{106} \pmod{107} \equiv 11^{27} \pmod{107}$
\(11^{999,999,999} \pmod{107}\)

Repeated squaring would take \(\sim \lg(999,999,999) \sim 30\) mults. By Fermat’s Little Thm \(11^{106} \equiv 1 \pmod{107}\). Note \(999,999,999 \equiv 27 \pmod{106}\)

\[999,999,999 = 106k + 27 \text{ (don’t care what } k \text{ is)}\]

\[11^{999,999,999} = 11^{106k} \times 11^{27} = (11^{106})^k \times 11^{27} \equiv 1^k 11^{27} \equiv 11^{27} \pmod{107}\]

\[11^{999,999,999} \equiv 11^{999,999,999} \pmod{106} \pmod{107} \equiv 11^{27} \pmod{107}\]

Now do normal repeated squaring, \(2\lg(27) = 10\). Can do better. Recall its really \(\lg(27) + \) the number of 1’s in the binary rep of 27.
Repeated squaring would take $\sim \log(999,999,999) \sim 30$ mults. By Fermat’s Little Thm $11^{106} \equiv 1 \pmod{107}$. Note $999,999,999 \equiv 27 \pmod{106}$

$$999,999,999 = 106k + 27 \text{ (don’t care what } k \text{ is)}$$

$$11^{999,999,999} = 11^{106k} \times 11^{27} = (11^{106})^k \times 11^{27} \equiv 1^k 11^{27} \equiv 11^{27} \pmod{107}$$

$$11^{999,999,999} \equiv 11^{999,999,999} \pmod{106} \pmod{107} \equiv 11^{27} \pmod{107}$$

Now do normal repeated squaring, $2 \log(27) = 10$. Can do better. Recall its really

$\log(27) +$ the number of 1’s in the binary rep of 27.

Can we generalize?
$11^{999,999,999} \pmod{107}$

Repeated squaring would take $\sim \lg(999,999,999) \sim 30$ mults.
By Fermat’s Little Thm $11^{106} \equiv 1 \pmod{107}$. Note $999,999,999 \equiv 27 \pmod{106}$

$$999,999,999 = 106k + 27 \text{ (don’t care what } k \text{ is)}$$

$$11^{999,999,999} = 11^{106k} \times 11^{27} = (11^{106})^k \times 11^{27} \equiv 1^k 11^{27} \equiv 11^{27} \pmod{107}$$

$$11^{999,999,999} \equiv 11^{999,999,999} \pmod{106} \pmod{107} \equiv 11^{27} \pmod{107}$$

Now do normal repeated squaring, $2\lg(27) = 10$. Can do better.
Recall its really $\lg(27)+$ the number of 1’s in the binary rep of 27.
Can we generalize? Yes
Exponentiation with Really Big Exponents

**Generalize** $p$ prime, $a \not\equiv 0 \pmod{p}$, $m \in \mathbb{N}$. 
Exponentiation with Really Big Exponents

**Generalize** $p$ prime, $a \not\equiv 0 \pmod{p}$, $m \in \mathbb{N}$.
We want to compute $a^m \pmod{p}$.

We know that $a^{p-1} \equiv 1 \pmod{p}$.

Divide $m$ by $p-1$:

$m = k(p-1) + r$

Hence:

$a^m \equiv a^{k(p-1)+r} \equiv (a^{p-1})^k \cdot a^r \equiv 1^k \cdot a^r \equiv a^r \pmod{p}$

Since $r \equiv m \pmod{p-1}$,

$a^m \equiv a^m \pmod{p-1} \pmod{p}$

This last equation is the important point.
Exponentiation with Really Big Exponents

**Generalize** $p$ prime, $a \not\equiv 0 \pmod{p}$, $m \in \mathbb{N}$.

We want to compute $a^m \pmod{p}$.

We know that $a^{p-1} \equiv 1 \pmod{p}$.
Exponentiation with Really Big Exponents

**Generalize** $p$ prime, $a \not\equiv 0 \pmod{p}$, $m \in \mathbb{N}$.

We want to compute $a^m \pmod{p}$.

We know that $a^{p-1} \equiv 1 \pmod{p}$. Divide $m$ by $p - 1$:
**Generalize** $p$ prime, $a \not\equiv 0 \pmod{p}$, $m \in \mathbb{N}$.

We want to compute $a^m \pmod{p}$.

We know that $a^{p-1} \equiv 1 \pmod{p}$. Divide $m$ by $p - 1$:

$m = k(p - 1) + r$
Exponentiation with Really Big Exponents

Generalize \( p \) prime, \( a \not\equiv 0 \pmod{p} \), \( m \in \mathbb{N} \).

We want to compute \( a^m \pmod{p} \).

We know that \( a^{p-1} \equiv 1 \pmod{p} \). Divide \( m \) by \( p - 1 \): \( m = k(p - 1) + r \)

Hence:
Generalize $p$ prime, $a \not\equiv 0 \pmod{p}$, $m \in \mathbb{N}$.

We want to compute $a^m \pmod{p}$.

We know that $a^{p-1} \equiv 1 \pmod{p}$. Divide $m$ by $p - 1$:

$m = k(p - 1) + r$

Hence:

$$a^m \equiv a^{k(p-1)+r} \equiv (a^{p-1})^k \times a^r \equiv 1^k a^r \equiv a^r$$
Exponentiation with Really Big Exponents

**Generalize** \( p \) prime, \( a \not\equiv 0 \pmod{p} \), \( m \in \mathbb{N} \).

We want to compute \( a^m \pmod{p} \).

We know that \( a^{p-1} \equiv 1 \pmod{p} \). Divide \( m \) by \( p - 1 \):

\[
m = k(p - 1) + r
\]

Hence:

\[
a^m \equiv a^{k(p-1)+r} \equiv (a^{p-1})^k \times a^r \equiv 1^k a^r \equiv a^r
\]

Since \( r \equiv m \pmod{p - 1} \), \( a^m \equiv a^{m \mod{p-1}} \pmod{p} \)
Generalize $p$ prime, $a \not\equiv 0 \pmod{p}$, $m \in \mathbb{N}$.

We want to compute $a^m \pmod{p}$.

We know that $a^{p-1} \equiv 1 \pmod{p}$. Divide $m$ by $p-1$:

$m = k(p - 1) + r$

Hence:

$$a^m \equiv a^{k(p-1)+r} \equiv (a^{p-1})^k \times a^r \equiv 1^k a^r \equiv a^r$$

Since $r \equiv m \pmod{p-1}$, $a^m \equiv a^{m \pmod{p-1}} \pmod{p}$

This last equation is the important point.
Needed Mathematics- The $\phi$ Function

Next few slides are on the $\phi$ function.


Needed Mathematics- The \( \phi \) Function

Next few slides are on the \( \phi \) function.

YES, you have already seen it.
Needed Mathematics- The $\phi$ Function

Next few slides are on the $\phi$ function.

YES, you have already seen it.

Who first said

**Math is best learned twice... at least twice.**
Needed Mathematics- The $\phi$ Function

Next few slides are on the $\phi$ function.

YES, you have already seen it.

Who first said

**Math is best learned twice... at least twice.**

My CMSC 858R class thought *either Gauss or Gasarch.*
Next few slides are on the $\phi$ function.

YES, you have already seen it.

Who first said

Math is best learned twice... at least twice.

My CMSC 858R class thought either Gauss or Gasarch.

Answer: Said by Larry Denenberg, who was a grad student in CS the same time Bill Gasarch was.
Needed Mathematics- The $\phi$ Function

Next few slides are on the $\phi$ function.

YES, you have already seen it.

Who first said

Math is best learned twice... at least twice.
My CMSC 858R class thought either Gauss or Gasarch.

Answer: Said by Larry Denenberg, who was a grad student in CS the same time Bill Gasarch was. Popularized by Bill Gasarch.
Needed Mathematics- The $\phi$ Function

Next few slides are on the $\phi$ function.

YES, you have already seen it.

Who first said
\textbf{Math is best learned twice... at least twice.}
My CMSC 858R class thought either Gauss or Gasarch.

\textbf{Answer:} Said by Larry Denenberg, who was a grad student in CS the same time Bill Gasarch was. Popularized by Bill Gasarch. Probably not said by Gauss.
Needed Mathematics- The $\phi$ Function

Next few slides are on the $\phi$ function.

YES, you have already seen it.

Who first said

Math is best learned twice... at least twice.

My CMSC 858R class thought either Gauss or Gasarch.

Answer: Said by Larry Denenberg, who was a grad student in CS the same time Bill Gasarch was. Popularized by Bill Gasarch. Probably not said by Gauss. Probably not true for Gauss.
Recall If $p$ is prime and $1 \leq a \leq p - 1$ then $a^{p-1} \equiv 1 \pmod{p}$. 
Recall If $p$ is prime and $1 \leq a \leq p - 1$ then $a^{p-1} \equiv 1 \pmod{p}$.

Recall For all $m$, $a^m \equiv a^m \pmod{p-1} \pmod{p}$.

So arithmetic in the exponents is mod $p - 1$. 
Recall If $p$ is prime and $1 \leq a \leq p - 1$ then $a^{p-1} \equiv 1 \pmod{p}$.

Recall For all $m$, $a^m \equiv a^m \pmod{p-1} \pmod{p}$.

So arithmetic in the exponents is mod $p - 1$.

We need to generalize this to when the mod is not a prime.
Recall If $p$ is prime and $1 \leq a \leq p - 1$ then $a^{p-1} \equiv 1 \pmod{p}$.

Recall For all $m$, $a^m \equiv a^{m \pmod{p-1}} \pmod{p}$.

So arithmetic in the exponents is mod $p - 1$.

We need to generalize this to when the mod is not a prime.

Definition $\phi(n)$ is the number of numbers in $\{1, \ldots, n\}$ that are relatively prime to $n$. 
Recall If $p$ is prime and $1 \leq a \leq p - 1$ then $a^{p-1} \equiv 1 \pmod{p}$.

Recall For all $m$, $a^m \equiv a^m \pmod{p-1} \pmod{p}$.

So arithmetic in the exponents is mod $p - 1$.

We need to generalize this to when the mod is not a prime.

**Definition** $\phi(n)$ is the number of numbers in $\{1, \ldots, n\}$ that are relatively prime to $n$.

Recall If $p$ is prime then $\phi(p) = p - 1$. 
Recall If $p$ is prime and $1 \leq a \leq p - 1$ then $a^{p-1} \equiv 1 \pmod{p}$.

Recall For all $m$, $a^m \equiv a^m \pmod{p-1} \pmod{p}$.

So arithmetic in the exponents is mod $p - 1$.

We need to generalize this to when the mod is not a prime.

Definition $\phi(n)$ is the number of numbers in $\{1, \ldots, n\}$ that are relatively prime to $n$.

Recall If $p$ is prime then $\phi(p) = p - 1$.

Recall If $a, b$ rel prime then $\phi(ab) = \phi(a)\phi(b)$.
Theorem for Primes, Theorem for $n$

We restate and generalize.
We restate and generalize.

**Fermat’s Little Theorem** If $p$ is prime and $a \not\equiv 0 \pmod{p}$ then

$$a^m \equiv a^m \pmod{p-1} \pmod{p}.$$
We restate and generalize.

**Fermat’s Little Theorem** If \( p \) is prime and \( a \not\equiv 0 \pmod{p} \) then

\[
a^m \equiv a^m \mod{p-1} \pmod{p}.
\]

Restate:

**Fermat’s Little Theorem** If \( p \) is prime and \( a \) is rel prime to \( p \) then

\[
a^m \equiv a^m \mod{\phi(p)} \pmod{p}.
\]
We restate and generalize.

**Fermat’s Little Theorem** If $p$ is prime and $a \not\equiv 0 \pmod{p}$ then

$$a^m \equiv a^m \mod{p-1} \pmod{p}.$$

Restate:

**Fermat’s Little Theorem** If $p$ is prime and $a$ is rel prime to $p$ then

$$a^m \equiv a^m \mod{\phi(p)} \pmod{p}.$$

Generalize:

**Fermat-Euler Theorem** If $n \in \mathbb{N}$ and $a$ is rel prime to $n$ then

$$a^m \equiv a^m \mod{\phi(n)} \pmod{n}.$$
Examples

$14^{999,999} \pmod{393}$
**Examples**

$$14^{999,999} \pmod{393}$$

$$\phi(393) = \phi(3 \times 131) = \phi(3) \times \phi(131) = 2 \times 130 = 260.$$
Examples

$14^{999,999} \pmod{393}$

$\phi(393) = \phi(3 \times 131) = \phi(3) \times \phi(131) = 2 \times 130 = 260.$

$14^{999,999} = 14^{999,999} \pmod{260} \pmod{393} \equiv 14^{39} \pmod{393}$
Examples

\[ 14^{999,999} \pmod{393} \]

\[ \phi(393) = \phi(3 \times 131) = \phi(3) \times \phi(131) = 2 \times 130 = 260. \]

\[ 14^{999,999} = 14^{999,999} \pmod{260} \quad \pmod{393} \equiv 14^{39} \pmod{393} \]

Now just do repeated squaring.
Bait and Switch

I got you interested in the theorem

$$a^m \equiv a^m \mod \phi(n) \pmod{n}$$

by telling you that it can be used to do things like

$$17^{191,992,194,299,292,777} \pmod{150}.$$  

with \textbf{much less than} $2 \log(191, 992, 194, 299, 292, 777)$ mults.
Bait and Switch

I got you interested in the theorem

$$a^m \equiv a^m \text{mod } \phi(n) \pmod{n}$$

by telling you that it can be used to do things like

$$17^{191,992,194,299,292,777} \pmod{150}.$$ 

with **much less than** $2 \lg(191, 992, 194, 299, 292, 777)$ mults. This is true! There will be some HW using it.
Bait and Switch

I got you interested in the theorem

\[ a^m \equiv a^m \mod \phi(n) \pmod{n} \]

by telling you that it can be used to do things like

\[ 17^{191,992,194,299,292,777} \pmod{150} \]

with much less than \( 2 \lg(191,992,194,299,292,777) \) mults. This is true! There will be some HW using it.

You are thinking
Bait and Switch

I got you interested in the theorem

\[ a^m \equiv a^m \mod \phi(n) \pmod{n} \]

by telling you that it can be used to do things like

\[ 17^{191,992,194,299,292,777} \pmod{150}. \]

with much less than \(2 \lg(191, 992, 194, 299, 292, 777)\) mults. This is true! There will be some HW using it.

You are thinking A&B will need to do \(a^m \pmod{n}\) for large \(m\).
Bait and Switch

I got you interested in the theorem

$$a^m \equiv a^m \mod \phi(n) \pmod{n}$$

by telling you that it can be used to do things like

$$17^{191,992,194,299,292,777} \pmod{150}.$$ 

with much less than $2 \lg(191, 992, 194, 299, 292, 777)$ mults. This is true! There will be some HW using it.

You are thinking A&B will need to do $a^m \pmod{n}$ for large $m$.

No. That is not what we will be doing, though I see why you would think that. Or you see why I think you would think that. Or . . . .
I got you interested in the theorem

\[ a^m \equiv a^m \mod \phi(n) \pmod{n} \]

by telling you that it can be used to do things like

\[ 17^{191,992,194,299,292,777} \pmod{150}. \]

with much less than \( 2 \log(191, 992, 194, 299, 292, 777) \) mults.

This is true! There will be some HW using it.

**You are thinking** A&B will need to do \( a^m \pmod{n} \) for large \( m \).

**No.** That is not what we will be doing, though I see why you would think that. Or you see why I think you would think that. Or . . . .

We will just use the theorem:

\[ a^m \equiv a^m \mod \phi(n) \pmod{n}. \]
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p$, $q$.
   Easy.

2. Given $p$, $q$ find $N = pq$ and $R = (p−1)(q−1)$.
   Easy.

3. Given $R$ find an $e$ rel prime to $R$. ($e$ for encrypt).
   Easy.

4. Given $R$, $e$ find $d$ such that $ed \equiv 1 \pmod{R}$.
   Easy.

5. Given $N$, $e$ find $d$ such that $ed \equiv 1 \pmod{R}$.
   Hard.

6. Compute $m^e \pmod{N}$.
   Easy.
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p, q$. 
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p, q$.  **Easy.**
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p, q$. Easy.
2. Given $p, q$ find $N = pq$ and $R = (p - 1)(q - 1)$.
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p$, $q$. Easy.
2. Given $p$, $q$ find $N = pq$ and $R = (p - 1)(q - 1)$. Easy.
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p, q$. Easy.
2. Given $p, q$ find $N = pq$ and $R = (p - 1)(q - 1)$. Easy.
3. Given $R$ find an $e$ rel prime to $R$. ($e$ for encrypt.)
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p$, $q$. Easy.
2. Given $p$, $q$ find $N = pq$ and $R = (p - 1)(q - 1)$. Easy.
3. Given $R$ find an $e$ rel prime to $R$. ($e$ for encrypt.) Easy.
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p, q$. **Easy.**
2. Given $p, q$ find $N = pq$ and $R = (p - 1)(q - 1)$. **Easy.**
3. Given $R$ find an $e$ rel prime to $R$. ($e$ for encrypt.) **Easy.**
4. Given $R, e$ find $d$ such that $ed \equiv 1 \pmod{R}$. **Hard.**
5. Compute $m^e \pmod{N}$. **Easy.**
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p, q$. **Easy.**
2. Given $p, q$ find $N = pq$ and $R = (p - 1)(q - 1)$. **Easy.**
3. Given $R$ find an $e$ rel prime to $R$. ($e$ for encrypt.) **Easy.**
4. Given $R, e$ find $d$ such that $ed \equiv 1 \pmod{R}$. **Easy.**
Easy and Hard

Easy or Hard?

1. Given \( L \), generate two primes of length \( L \): \( p \), \( q \). **Easy.**
2. Given \( p \), \( q \) find \( N = pq \) and \( R = (p - 1)(q - 1) \). **Easy.**
3. Given \( R \) find an \( e \) rel prime to \( R \). (\( e \) for encrypt.) **Easy.**
4. Given \( R \), \( e \) find \( d \) such that \( ed \equiv 1 \pmod{R} \). **Easy.**
5. Given \( N \), \( e \) find \( d \) such that \( ed \equiv 1 \pmod{R} \).
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p, q$. Easy.
2. Given $p, q$ find $N = pq$ and $R = (p - 1)(q - 1)$. Easy.
3. Given $R$ find an $e$ rel prime to $R$. ($e$ for encrypt.) Easy.
4. Given $R, e$ find $d$ such that $ed \equiv 1 \pmod{R}$. Easy.
5. Given $N, e$ find $d$ such that $ed \equiv 1 \pmod{R}$. Hard.
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p, q$. **Easy.**
2. Given $p, q$ find $N = pq$ and $R = (p - 1)(q - 1)$. **Easy.**
3. Given $R$ find an $e$ rel prime to $R$. ($e$ for encrypt.) **Easy.**
4. Given $R, e$ find $d$ such that $ed \equiv 1 \pmod{R}$. **Easy.**
5. Given $N, e$ find $d$ such that $ed \equiv 1 \pmod{R}$. **Hard.**
6. Compute $m^e \pmod{N}$. **Easy.**
Easy and Hard

Easy or Hard?

1. Given $L$, generate two primes of length $L$: $p, q$.  **Easy.**
2. Given $p, q$ find $N = pq$ and $R = (p - 1)(q - 1)$.  **Easy.**
3. Given $R$ find an $e$ rel prime to $R$.  ($e$ for encrypt.)  **Easy.**
4. Given $R, e$ find $d$ such that $ed \equiv 1 \pmod{R}$.  **Easy.**
5. Given $N, e$ find $d$ such that $ed \equiv 1 \pmod{R}$.  **Hard.**
6. Compute $m^e \pmod{N}$.  **Easy.**
RSA

Let $L$ be a security parameter
RSA

Let $L$ be a security parameter

1. Alice picks two primes $p, q$ of length $L$ and computes $N = pq$.
RSA

Let $L$ be a security parameter

1. **Alice** picks two primes $p, q$ of length $L$ and computes $N = pq$.
2. **Alice** computes $R = \phi(N) = \phi(pq) = (p - 1)(q - 1)$.
RSA

Let $L$ be a security parameter

1. **Alice** picks two primes $p, q$ of length $L$ and computes $N = pq$.
2. **Alice** computes $R = \phi(N) = \phi(pq) = (p - 1)(q - 1)$.
3. **Alice** picks an $e \in \{\frac{R}{3}, \ldots, \frac{2R}{3}\}$ that is relatively prime to $R$. 

**PRO** Alice and Bob can execute the protocol easily.

**Biggest PRO** Alice and Bob never had to meet!

**Question** Can Eve find out $m$?
RSA

Let $L$ be a security parameter

1. **Alice** picks two primes $p, q$ of length $L$ and computes $N = pq$.
2. **Alice** computes $R = \phi(N) = \phi(pq) = (p - 1)(q - 1)$.
3. **Alice** picks an $e \in \{ \frac{R}{3}, \ldots, \frac{2R}{3} \}$ that is relatively prime to $R$.
4. **Alice** finds $d$ such that $ed \equiv 1 \pmod{R}$.
Let $L$ be a security parameter

1. **Alice** picks two primes $p$, $q$ of length $L$ and computes $N = pq$.
2. **Alice** computes $R = \phi(N) = \phi(pq) = (p - 1)(q - 1)$.
3. **Alice** picks an $e \in \{\frac{R}{3}, \ldots, \frac{2R}{3}\}$ that is relatively prime to $R$.
4. **Alice** finds $d$ such that $ed \equiv 1 \pmod{R}$.
5. **Alice** broadcasts $(N, e)$. (Bob and Eve both see it.)
RSA

Let $L$ be a security parameter

1. **Alice** picks two primes $p, q$ of length $L$ and computes $N = pq$.
2. **Alice** computes $R = \phi(N) = \phi(pq) = (p - 1)(q - 1)$.
3. **Alice** picks an $e \in \{\frac{R}{3}, \ldots, \frac{2R}{3}\}$ that is relatively prime to $R$.
4. **Alice** finds $d$ such that $ed \equiv 1 \pmod{R}$.
5. **Alice** broadcasts $(N, e)$. (Bob and Eve both see it.)
6. **Bob** To send $m \in \{1, \ldots, N - 1\}$, send $m^e \pmod{N}$. 

PRO

Alice and Bob can execute the protocol easily.

Biggest PRO

Alice and Bob never had to meet!

Question

Can Eve find out $m$?
RSA

Let $L$ be a security parameter

1. Alice picks two primes $p, q$ of length $L$ and computes $N = pq$.
2. Alice computes $R = \phi(N) = \phi(pq) = (p - 1)(q - 1)$.
3. Alice picks an $e \in \{\frac{R}{3}, \ldots, \frac{2R}{3}\}$ that is relatively prime to $R$.
4. Alice finds $d$ such that $ed \equiv 1 \pmod{R}$.
5. Alice broadcasts $(N, e)$. (Bob and Eve both see it.)
6. Bob To send $m \in \{1, \ldots, N - 1\}$, send $m^e \pmod{N}$.
7. If Alice gets $m^e \pmod{N}$ she computes

$$(m^e)^d \equiv m^{ed} \equiv m^{ed} \pmod{R} \equiv m^1 \pmod{R} \equiv m.$$
RSA

Let $L$ be a security parameter

1. Alice picks two primes $p, q$ of length $L$ and computes $N = pq$.
2. Alice computes $R = \phi(N) = \phi(pq) = (p - 1)(q - 1)$.
3. Alice picks an $e \in \left\{ \frac{R}{3}, \ldots, \frac{2R}{3} \right\}$ that is relatively prime to $R$.
4. Alice finds $d$ such that $ed \equiv 1 \pmod{R}$.
5. Alice broadcasts $(N, e)$. (Bob and Eve both see it.)
6. Bob To send $m \in \{1, \ldots, N - 1\}$, send $m^e \pmod{N}$.
7. If Alice gets $m^e \pmod{N}$ she computes

$$
(m^e)^d \equiv m^{ed} \equiv m^{ed \mod R} \equiv m^{1 \mod R} \equiv m.
$$

PRO Alice and Bob can execute the protocol easily.
RSA

Let $L$ be a security parameter

1. **Alice** picks two primes $p, q$ of length $L$ and computes $N = pq$.
2. **Alice** computes $R = \phi(N) = \phi(pq) = (p - 1)(q - 1)$.
3. **Alice** picks an $e \in \{\frac{R}{3}, \ldots, \frac{2R}{3}\}$ that is relatively prime to $R$.
4. **Alice** finds $d$ such that $ed \equiv 1 \pmod{R}$.
5. **Alice** broadcasts $(N, e)$. (Bob and Eve both see it.)
6. **Bob** To send $m \in \{1, \ldots, N - 1\}$, send $m^e \pmod{N}$.
7. If **Alice** gets $m^e \pmod{N}$ she computes

   $$(m^e)^d \equiv m^{ed} \equiv m^{ed \pmod{R}} \equiv m^1 \pmod{R} \equiv m.$$ 

**PRO** Alice and Bob can execute the protocol easily.

**Biggest PRO** Alice and Bob never had to meet!
RSA

Let $L$ be a security parameter

1. **Alice** picks two primes $p, q$ of length $L$ and computes $N = pq$.
2. **Alice** computes $R = \phi(N) = \phi(pq) = (p - 1)(q - 1)$.
3. **Alice** picks an $e \in \{\frac{R}{3}, \ldots, \frac{2R}{3}\}$ that is relatively prime to $R$.
4. **Alice** finds $d$ such that $ed \equiv 1 \pmod{R}$.
5. **Alice** broadcasts $(N, e)$. (Bob and Eve both see it.)
6. **Bob** To send $m \in \{1, \ldots, N - 1\}$, send $m^e \pmod{N}$.
7. If **Alice** gets $m^e \pmod{N}$ she computes

   $$(m^e)^d \equiv m^{ed} \equiv m^{ed \mod R} \equiv m^{1 \mod R} \equiv m.$$ 

**PRO** Alice and Bob can execute the protocol easily.

**Biggest PRO** Alice and Bob never had to meet!

**Question** Can Eve find out $m$?
Convention for RSA

Alice sends \((N, e)\) to get the process started.
Convention for RSA

Alice sends \((N, e)\) to get the process started.

Then Bob can send Alice messages.
Convention for RSA

Alice sends \((N, e)\) to get the process started.

Then Bob can send Alice messages.

We don’t have Alice sending Bob messages.
Convention for RSA

Alice sends \((N, e)\) to get the process started.

Then Bob can send Alice messages.

We don't have Alice sending Bob messages.

In examples we do in slides and HW we might not have \(e \in \{R/3, \ldots, 2R/3\}\) since we want to have easy computations for educational purposes.
Pick out two students to be Alice and Bob.
Use primes:
\[ p = 31, \text{ Prime.} \]
\[ q = 37, \text{ Prime.} \]
Pick out two students to be Alice and Bob.

Use primes:

\[ p = 31, \text{Prime.} \]
\[ q = 37, \text{Prime.} \]

\[ N = pq = 31 \times 37 = 1147. \]
\[ R = \phi(N) = 30 \times 36 = 1080. \]
Do RSA in Class

Pick out two students to be Alice and Bob. Use primes:
\[ p = 31, \text{ Prime.} \]
\[ q = 37, \text{ Prime.} \]
\[ N = pq = 31 \times 37 = 1147. \]
\[ R = \phi(N) = 30 \times 36 = 1080. \]
Use \( e = 77 \), \( e \) rel prime to \( R \)
Find \( d = 533 \) \( (ed \equiv 1 \pmod{R}) \)
\textbf{Check} \( ed = 77 \times 533 = 41041 \equiv 1 \pmod{1080} \).
Do RSA in Class

Pick out two students to be Alice and Bob.
Use primes:
\( p = 31, \) Prime.
\( q = 37, \) Prime.
\( N = pq = 31 \times 37 = 1147. \)
\( R = \phi(N) = 30 \times 36 = 1080. \)
Use \( e = 77, \) \( e \) rel prime to \( R \)
Find \( d = 533 \) \( (ed \equiv 1 \pmod{R}) \)
Check \( ed = 77 \times 533 = 41041 \equiv 1 \pmod{1080}. \)
Bob pick an \( m \in \{1, \ldots, N - 1\} = \{1, \ldots, 1146\}. \) Do not tell us what it is.
Do RSA in Class

Pick out two students to be Alice and Bob.
Use primes:
\[ p = 31, \text{ Prime.} \]
\[ q = 37, \text{ Prime.} \]
\[ N = pq = 31 \times 37 = 1147. \]
\[ R = \phi(N) = 30 \times 36 = 1080. \]
Use \( e = 77 \), \( e \) rel prime to \( R \)
Find \( d = 533 \) \((ed \equiv 1 \pmod{R})\)
Check \( ed = 77 \times 533 = 41041 \equiv 1 \pmod{1080}. \)
Bob pick an \( m \in \{1, \ldots, N-1\} = \{1, \ldots, 1146\} \). Do not tell us what it is.
Bob compute \( c = m^e \pmod{1147} \) and tell it to us.
Do RSA in Class

Pick out two students to be Alice and Bob.
Use primes:
\( p = 31 \), Prime.
\( q = 37 \), Prime.

\( N = pq = 31 \times 37 = 1147 \).
\( R = \phi(N) = 30 \times 36 = 1080 \).
Use \( e = 77 \), \( e \) rel prime to \( R \)
Find \( d = 533 \) \( (ed \equiv 1 \pmod{R}) \)
Check \( ed = 77 \times 533 = 41041 \equiv 1 \pmod{1080} \).

Bob pick an \( m \in \{1, \ldots, N - 1\} = \{1, \ldots, 1146\} \). Do not tell us what it is.
Bob compute \( c = m^e \pmod{1147} \) and tell it to us.
Alice compute \( c^d \pmod{1147} \), should get back \( m \).
What Do We Really Know about RSA

If Eve can factor then she can crack RSA.

1. Input \((N, e)\) where \(N = pq\) and \(e\) is rel prime to \(R = (p − 1)(q − 1)\). (\(p, q, R\) are NOT part of the input.)
2. Eve factors \(N\) to find \(p, q\). Eve computes \(R = (p − 1)(q − 1)\).
3. Eve finds \(d\) such that \(ed \equiv 1 \pmod{R}\).

If Factoring Easy then RSA is crackable
What Do We Really Know about RSA

If Eve can factor then she can crack RSA.

1. Input \((N, e)\) where \(N = pq\) and \(e\) is rel prime to 
   \(R = (p - 1)(q - 1)\). \((p, q, R\) are NOT part of the input.)
2. Eve factors \(N\) to find \(p, q\). Eve computes \(R = (p - 1)(q - 1)\).
3. Eve finds \(d\) such that \(ed \equiv 1 \pmod{R}\).

If Factoring Easy then RSA is crackable

What about converse?

If RSA is crackable then Factoring is Easy

VOTE True or False or Unknown to Science
What Do We Really Know about RSA

If Eve can factor then she can crack RSA.

1. Input \((N,e)\) where \(N = pq\) and \(e\) is rel prime to 
   \(R = (p - 1)(q - 1)\). \((p, q, R\) are NOT part of the input.)
2. Eve factors \(N\) to find \(p, q\). Eve computes \(R = (p - 1)(q - 1)\).
3. Eve finds \(d\) such that \(ed \equiv 1 \pmod{R}\).

If Factoring Easy then RSA is crackable

What about converse?

If RSA is crackable then Factoring is Easy

VOTE TRUE or FALSE or UNKNOWN TO SCIENCE
UNKNOWN TO SCIENCE.
What Do We Really Know about RSA

If Eve can factor then she can crack RSA.

1. Input \((N, e)\) where \(N = pq\) and \(e\) is rel prime to \(R = (p - 1)(q - 1)\). \((p, q, R\) are NOT part of the input.)
2. Eve factors \(N\) to find \(p, q\). Eve computes \(R = (p - 1)(q - 1)\).
3. Eve finds \(d\) such that \(ed \equiv 1 \pmod{R}\).

If Factoring Easy then RSA is crackable

What about converse?

If RSA is crackable then Factoring is Easy

VOTE TRUE or FALSE or UNKNOWN TO SCIENCE

UNKNOWN TO SCIENCE.

Note In ugrad math classes rare to have a statement that is UNKNOWN TO SCIENCE. Discuss.
**Definition** Let \( f \) be the following function:

**Input** \( N, e, m^e \pmod{N} \) (know \( N = pq \) but don’t know \( p, q \)).

**Outputs** \( m \).

**Hardness assumption (HA)** \( f \) is hard to compute.

One can show, assuming HA that RSA is hard to crack. But this proof will depend on a model of security. See caveats about this on similar DH slides (bribery, timing attacks, Maginot Line).
What Could be True?

The following are all possible:

1) Factoring easy. RSA is crackable.
2) Factoring hard, HA false. RSA crackable, Factoring hard!!
3) Factoring hard, HA true, but RSA is crackable by other means. Timing Attacks. Must rethink our model of security.
4) Factoring hard, HA true, and RSA remains uncracked for years. Increases our confidence but...

Item 4 is current state with some caveats: Do Alice and Bob use it properly? Do they have large enough parameters? What is Eve’s computing power?
What Could be True?

The following are all possible:

1) Factoring easy. RSA is crackable.
What Could be True?

The following are all possible:

1) Factoring easy. RSA is crackable.
2) Factoring hard, HA false. RSA crackable, Factoring hard!!

Item 4 is current state with some caveats: Do Alice and Bob use it properly? Do they have large enough parameters? What is Eve's computing power?
What Could be True?

The following are all possible:

1) Factoring easy. RSA is crackable.
2) Factoring hard, HA false. RSA crackable, Factoring hard!!
3) Factoring hard, HA true, but RSA is crackable by other means. Timing Attacks. Must rethink our model of security.

Item 4 is current state with some caveats: Do Alice and Bob use it properly? Do they have large enough parameters? What is Eve's computing power?
What Could be True?

The following are all possible:

1) Factoring easy. RSA is crackable.
2) Factoring hard, HA false. RSA crackable, Factoring hard!!
3) Factoring hard, HA true, but RSA is crackable by other means. Timing Attacks. Must rethink our model of security.
4) Factoring hard, HA true, and RSA remains uncracked for years. Increases our confidence but . . . .
What Could be True?

The following are all possible:

1) Factoring easy. RSA is crackable.
2) Factoring hard, HA false. RSA crackable, Factoring hard!!
3) Factoring hard, HA true, but RSA is crackable by other means. Timing Attacks. Must rethink our model of security.
4) Factoring hard, HA true, and RSA remains uncracked for years. Increases our confidence but . . . .

Item 4 is current state with some caveats: Do Alice and Bob use it properly? Do they have large enough parameters? What is Eve’s computing power?
RSA has NY, NY
Problem. Will Fix
Plain RSA Bytes!

The RSA given above is referred to as **Plain RSA**.

Insecure!
Plain RSA Bytes!

The RSA given above is referred to as Plain RSA. Insecure!

Scenario
Eve sees Bob send Alice $c_1$ (message is $m_1$).
Plain RSA Bytes!

The RSA given above is referred to as **Plain RSA**.

**Insecure!**

**Scenario**
Eve sees Bob send Alice $c_1$ (message is $m_1$).
Later Eve sees Bob send Alice $c_2$ (message is $m_2$).
The RSA given above is referred to as **Plain RSA**. **Insecure!**

**Scenario**
Eve sees Bob send Alice $c_1$ (message is $m_1$).
Later Eve sees Bob send Alice $c_2$ (message is $m_2$).

What can Eve *easily* deduce?
The RSA given above is referred to as **Plain RSA**.

**Insecure!**

**Scenario**

Eve sees Bob send Alice $c_1$ (message is $m_1$).
Later Eve sees Bob send Alice $c_2$ (message is $m_2$).

What can Eve **easily** deduce?

Eve can know if $c_1 = c_2$ or not. So what?
Plain RSA Bytes!

The RSA given above is referred to as **Plain RSA**. **Insecure!**

**Scenario**
Eve sees Bob send Alice $c_1$ (message is $m_1$). Later Eve sees Bob send Alice $c_2$ (message is $m_2$).

What can Eve **easily** deduce?

Eve can know if $c_1 = c_2$ or not. So what?
Eve knows if $m_1 = m_2$ or not. Its the NY, NY problem!
Plain RSA Bytes!

The RSA given above is referred to as **Plain RSA**. **Insecure!**

**Scenario**
Eve sees Bob send Alice $c_1$ (message is $m_1$). Later Eve sees Bob send Alice $c_2$ (message is $m_2$).

What can Eve **easily** deduce?

Eve can know if $c_1 = c_2$ or not. So what?
Eve knows if $m_1 = m_2$ or not. Its the NY,NY problem!

That alone makes it insecure.
Plain RSA Bytes!

The RSA given above is referred to as **Plain RSA**. **Insecure!**

**Scenario**
Eve sees Bob send Alice $c_1$ (message is $m_1$).
Later Eve sees Bob send Alice $c_2$ (message is $m_2$).

What can Eve **easily** deduce?

Eve can know if $c_1 = c_2$ or not. So what?
Eve knows if $m_1 = m_2$ or not. Its the NY,NY problem!

That alone makes it insecure.
**Plain RSA is never used and should never be used!**
PKCS-1.5 RSA

How can we fix RSA to make it work? Discuss
PKCS-1.5 RSA

How can we fix RSA to make it work? Discuss Need randomness.
How can we fix RSA to make it work? **Discuss** Need randomness.

We need to change how Bob sends a message;

**BAD** To send \( m \in \{1, \ldots, N - 1\} \), send \( m^e \pmod{N} \).
PKCS-1.5 RSA

How can we fix RSA to make it work? **Discuss** Need randomness.

We need to change how Bob sends a message;

**BAD** To send $m \in \{1, \ldots, N - 1\}$, send $m^e \pmod{N}$.

**FIX** To send $m \in \{1, \ldots, N - 1\}$, pick rand $r$, send $(rm)^e$.

(NOTE- $rm$ means $r$ CONCAT with $m$ here and elsewhere.) Alice and Bob agree on **length** of $r$ ahead of time.
PKCS-1.5 RSA

How can we fix RSA to make it work? **Discuss** Need randomness.

We need to change how Bob sends a message;

**BAD** To send \( m \in \{1, \ldots, N - 1\} \), send \( m^e \pmod{N} \).

**FIX** To send \( m \in \{1, \ldots, N - 1\} \), pick rand \( r \), send \((rm)^e\).

(NOTE- \( rm \) means \( r \) CONCAT with \( m \) here and elsewhere.) Alice and Bob agree on **length** of \( r \) ahead of time.

Alice and Bob pick \( L_1 \) and \( L_2 \) such that \( \lg N = L_1 + L_2 \).

To send \( m \in \{0, 1\}^{L_2} \) pick random \( r \in \{0, 1\}^{L_1} \).

When Alice gets \( rm \) she will know that \( m \) is the last \( L_2 \) bits.
Example

\[ p = 31, \ q = 37, \ N = pq = 31 \times 37 = 1147. \]
Example

\[ p = 31, \quad q = 37, \quad N = pq = 31 \times 37 = 1147. \]
\[ R = \phi(N) = 30 \times 36 = 1080 \]
Example

\[ p = 31, \ q = 37, \ N = pq = 31 \times 37 = 1147. \]
\[ R = \phi(N) = 30 \times 36 = 1080 \]
\[ e = 77 \ (e \text{ rel prime to } R), \ d = 533 \ (ed \equiv 1 \pmod{R}). \]
\[ L_1 = 3. \]
Example

\[ p = 31, \; q = 37, \; N = pq = 31 \times 37 = 1147. \]
\[ R = \phi(N) = 30 \times 36 = 1080 \]
\[ e = 77 \; (e \text{ rel prime to } R), \; d = 533 \; (ed \equiv 1 \; (\text{mod } R)). \]
\[ L_1 = 3. \]
Bob wants to send 1100100 (note- \( L_2 = 7 \) bits).
Example

\[ p = 31, \ q = 37, \ N = pq = 31 \times 37 = 1147. \]
\[ R = \phi(N) = 30 \times 36 = 1080 \]
\[ e = 77 \ (e \text{ rel prime to } R), \ d = 533 \ (ed \equiv 1 \ (\mod R)). \]
\[ L_1 = 3. \]

Bob wants to send 1100100 (note- \( L_2 = 7 \) bits).

1. Bob generates \( L_1 = 3 \) random bits. 100.
Example

\[ p = 31, \; q = 37, \; N = pq = 31 \times 37 = 1147. \]
\[ R = \phi(N) = 30 \times 36 = 1080 \]
\[ e = 77 \; (e \text{ rel prime to } R), \; d = 533 \; (ed \equiv 1 \pmod{R}). \]
\[ L_1 = 3. \]
Bob wants to send 1100100 (note- \( L_2 = 7 \) bits).

1. Bob generates \( L_1 = 3 \) random bits. 100.
2. Bob sends 1001100100 which is 612 in base 10 by sending \( 612^{77} \pmod{1147} \) which is 277.
Example

\[ p = 31, \ q = 37, \ N = pq = 31 \times 37 = 1147. \]
\[ R = \phi(N) = 30 \times 36 = 1080 \]
\[ e = 77 \ (e \text{ rel prime to } R), \ d = 533 \ (ed \equiv 1 \pmod{R}). \]
\[ L_1 = 3. \]
Bob wants to send 1100100 (note- \( L_2 = 7 \) bits).

1. Bob generates \( L_1 = 3 \) random bits. 100.

2. Bob sends 1001100100 which is 612 in base 10 by sending \( 612^{77} \pmod{1147} \) which is 277.

3. Alice decodes by doing \( 277^{533} \pmod{1147} = 612. \)
Example

\[ p = 31, \quad q = 37, \quad N = pq = 31 \times 37 = 1147. \]

\[ R = \phi(N) = 30 \times 36 = 1080 \]

\[ e = 77 \quad (e \text{ rel prime to } R), \quad d = 533 \quad (ed \equiv 1 \pmod{R}). \]

\[ L_1 = 3. \]

Bob wants to send 1100100 (note- \( L_2 = 7 \) bits).

1. Bob generates \( L_1 = 3 \) random bits. 100.

2. Bob sends 1001100100 which is 612 in base 10 by sending \( 612^{77} \pmod{1147} \) which is 277.

3. Alice decodes by doing \( 277^{533} \pmod{1147} = 612. \)

4. Alice puts 612 into binary to get 1001100100. She knows to only read the last 7 bits 1100100.
Example

\[ p = 31, \quad q = 37, \quad N = pq = 31 \times 37 = 1147. \]
\[ R = \phi(N) = 30 \times 36 = 1080 \]
\[ e = 77 \ (e \text{ rel prime to } R), \quad d = 533 \ (ed \equiv 1 \pmod{R}). \]
\[ L_1 = 3. \]
Bob wants to send 1100100 (note- \( L_2 = 7 \) bits).

1. Bob generates \( L_1 = 3 \) random bits. 100.
2. Bob sends 1001100100 which is 612 in base 10 by sending \( 612^{77} \pmod{1147} \) which is 277.
3. Alice decodes by doing \( 277^{533} \pmod{1147} = 612. \)
4. Alice puts 612 into binary to get 1001100100. She knows to only read the last 7 bits 1100100.

**Important** If later Bob wants to send 100 again he will choose a DIFFERENT random 3 bits so Eve won’t know he sent the same message.
RSA has Another Problem
Is PKCS-1.5 RSA Secure?

VOTE

Is PKCS-1.5 RSA Secure? VOTE
Is PKCS-1.5 RSA Secure?

Is PKCS-1.5 RSA Secure? VOTE
- YES (under hardness assumptions and large $n$)
Is PKCS-1.5 RSA Secure?

Is PKCS-1.5 RSA Secure? VOTE
- YES (under hardness assumptions and large $n$)
- NO (there is yet another weird security thing we overlooked)

Scenario
$N$ and $e$ are public. Bob sends $(r m)^e \pmod{N}$.
Eve cannot determine what $m$ is. What can Eve do that is still obnoxious?
Eve can compute $2^e (r m)^e \equiv (2(r m))^e \pmod{N}$. So what?
Eve can later pretend she is Bob and send $(2(r m))^e \pmod{N}$.
Why bad?
Discuss (1) will confuse Alice (2) Sealed Bid Scenario.
Is PKCS-1.5 RSA Secure?

Is PKCS-1.5 RSA Secure? VOTE

▶ YES (under hardness assumptions and large $n$)
▶ NO (there is yet another weird security thing we overlooked)

NO (there is yet another weird security thing we overlooked)
Is PKCS-1.5 RSA Secure?

- YES (under hardness assumptions and large $n$)
- NO (there is yet another weird security thing we overlooked)

NO (there is yet another weird security thing we overlooked)

**Scenario** $N$ and $e$ are public. Bob sends $(rm)^e \pmod{N}$. Eve cannot determine what $m$ is.
Is PKCS-1.5 RSA Secure?

VOTE

▶ YES (under hardness assumptions and large $n$)
▶ NO (there is yet another weird security thing we overlooked)

NO (there is yet another weird security thing we overlooked)

Scenario $N$ and $e$ are public. Bob sends $(rm)^e \pmod{N}$. Eve cannot determine what $m$ is. What can Eve do that is still obnoxious?
Is PKCS-1.5 RSA Secure?

VOTE
▶ YES (under hardness assumptions and large $n$)
▶ NO (there is yet another weird security thing we overlooked)

NO (there is yet another weird security thing we overlooked)

**Scenario** $N$ and $e$ are public. Bob sends $(rm)^e \pmod{N}$. Eve cannot determine what $m$ is.

What can Eve do that is still obnoxious? Eve can compute $2^e(rm)^e \equiv (2(rm))^e \pmod{N}$. So what?
Is PKCS-1.5 RSA Secure? VOTE

- YES (under hardness assumptions and large $n$)
- NO (there is yet another weird security thing we overlooked)

NO (there is yet another weird security thing we overlooked)

**Scenario** $N$ and $e$ are public. Bob sends $(rm)^e \pmod{N}$. Eve cannot determine what $m$ is.

What can Eve do that is still obnoxious?

Eve can compute $2^e(rm)^e \equiv (2(rm))^e \pmod{N}$. So what?

Eve can later pretend she is Bob and send $(2(rm))^e \pmod{N}$. 
Is PKCS-1.5 RSA Secure?  VOTE

- YES (under hardness assumptions and large $n$)
- NO (there is yet another weird security thing we overlooked)

**NO** (there is yet another weird security thing we overlooked)

**Scenario** $N$ and $e$ are public. Bob sends $(rm)^e \pmod{N}$. Eve cannot determine what $m$ is.

What can Eve do that is still obnoxious?

Eve can compute $2^e(rm)^e \equiv (2(rm))^e \pmod{N}$. So what?

Eve can later pretend she is Bob and send $(2(rm))^e \pmod{N}$.

Why bad? **Discuss**
Is PKCS-1.5 RSA Secure?

VOTE
- YES (under hardness assumptions and large $n$)
- NO (there is yet another weird security thing we overlooked)

NO (there is yet another weird security thing we overlooked)

**Scenario** $N$ and $e$ are public. Bob sends $(rm)^e \pmod{N}$. Eve cannot determine what $m$ is.

What can Eve do that is still obnoxious?

Eve can compute $2^e(rm)^e \equiv (2(rm))^e \pmod{N}$. So what?

Eve can later pretend she is Bob and send $(2(rm))^e \pmod{N}$.

Why bad? **Discuss**

(1) will confuse Alice
(2) Sealed Bid Scenario.
Malleability

An encryption system is **malleable** if when Eve sees a message she can figure out a way to send a similar one, where she knows the similarity (she still does not know the message).
An encryption system is **malleable** if when Eve sees a message she can figure out a way to send a similar one, where she knows the similarity (she still does not know the message).

1. The definition above is informal.
An encryption system is **malleable** if when Eve sees a message she can figure out a way to send a similar one, where she knows the similarity (she still does not know the message).

1. The definition above is informal.
2. Can modify RSA so that it’s probably not malleable.
Malleability

An encryption system is **malleable** if when Eve sees a message she can figure out a way to send a similar one, where she knows the similarity (she still does not know the message).

1. The definition above is informal.
2. Can modify RSA so that it’s probably not malleable.
3. That way is called PKCS-2.0-RSA.
Malleability

An encryption system is **malleable** if when Eve sees a message she can figure out a way to send a similar one, where she knows the similarity (she still does not know the message).

1. The definition above is informal.
2. Can modify RSA so that it’s probably not malleable.
3. That way is called PKCS-2.0-RSA.
4. Name BLAH-1.5 is hint that it’s not final version.
Other Public Key Systems
We really want to say
\textbf{Cracking RSA is Exactly as Hard as Factoring}
but we do not know this, and it’s probably false.
Better Hardness Assumptions

We really want to say

**Cracking RSA is Exactly as Hard as Factoring**

but we do not know this, and it’s probably false.

Are there other Public Key Cryptosystems that are equivalent to factoring?
Better Hardness Assumptions

We really want to say

**Cracking RSA is Exactly as Hard as Factoring**

but we do not know this, and it’s probably false.

Are there other Public Key Cryptosystems that are equivalent to factoring?

Yes. On Next Slide.
Rabin’s Encryption System and its Variants

1. Rabin’s encryption is equivalent to factoring $pq$.

2. Rabin’s encryption is hard to use: messages do not decode uniquely.

3. Blum-Williams modified Rabin’s encryption so that messages decode uniquely; but the set of messages you can send is small.

4. Hard to combine Blum-Williams modification with the padding needed to solve the NY,NY problem.

5. Cracking Rabin’s encryption is equivalent to factoring: but this is only if Eve has no other information.

6. If Eve can trick Alice into sending a chosen message, she can crack Rabin. So CPA-insecure.

Why is RSA used and not Rabin? either

1. The problems above make it not practical.

2. The problems above could have been gotten around but RSA just got to the market faster.
Rabin’s Encryption System and its Variants

1. Rabin’s enc equivalent to factoring $pq$. 

2. Rabin’s enc is hard to use: messages do not decode uniquely.

3. Blum-Williams modified Rabin’s Enc so that messages decode uniquely; but the set of messages you can send is small.

4. Hard to combine Blum-Williams modification with the padding needed to solve NY,NY problem.

5. Cracking Rabin Enc EQUIV factoring: but this is only if Eve has no other information.

6. If Eve can trick Alice into sending a chosen message, she can crack Rabin. So CPA-insecure.

Why is RSA used and not Rabin? either

1. The problems above make it not practical.

2. The problems above could have been gotten around but RSA just got to the market faster.
Rabin’s Encryption System and its Variants

1. Rabin’s enc equivalent to factoring $pq$.
2. Rabin’s enc is hard to use: messages do not decode uniquely.
Rabin’s Encryption System and its Variants

1. Rabin’s enc equivalent to factoring $pq$.
2. Rabin’s enc is hard to use: messages do not decode uniquely.
3. Blum-Williams modified Rabin’s Enc so that messages decode uniquely; but the set of messages you can send is small.
1. Rabin’s enc equivalent to factoring $pq$.
2. Rabin’s enc is hard to use: messages do not decode uniquely.
3. Blum-Williams modified Rabin’s Enc so that messages decode uniquely; but the set of messages you can send is small.
4. Hard to combine Blum-Williams modification with the padding needed to solve NY,NY problem.
Rabin’s Encryption System and its Variants

1. Rabin’s enc equivalent to factoring $pq$.
2. Rabin’s enc is hard to use: messages do not decode uniquely.
3. Blum-Williams modified Rabin’s Enc so that messages decode uniquely; but the set of messages you can send is small.
4. Hard to combine Blum-Williams modification with the padding needed to solve NY,NY problem.
5. Cracking Rabin Enc EQUIV factoring: but this is only if Eve has no other information.
Rabin’s Encryption System and its Variants

1. Rabin’s enc equivalent to factoring $pq$.
2. Rabin’s enc is hard to use: messages do not decode uniquely.
3. Blum-Williams modified Rabin’s Enc so that messages decode uniquely; but the set of messages you can send is small.
4. Hard to combine Blum-Williams modification with the padding needed to solve NY, NY problem.
5. Cracking Rabin Enc EQUIV factoring: but this is only if Eve has no other information.
6. If Eve can trick Alice into sending a chosen message, she can crack Rabin. So CPA-insecure.
Rabin’s Encryption System and its Variants

1. Rabin’s enc equivalent to factoring \( pq \).
2. Rabin’s enc is hard to use: messages do not decode uniquely.
3. Blum-Williams modified Rabin’s Enc so that messages decode uniquely; but the set of messages you can send is small.
4. Hard to combine Blum-Williams modification with the padding needed to solve NY,NY problem.
5. Cracking Rabin Enc EQUIV factoring: but this is only if Eve has no other information.
6. If Eve can trick Alice into sending a chosen message, she can crack Rabin. So CPA-insecure.

Why is RSA used and not Rabin? either
Rabin’s Encryption System and its Variants

1. Rabin’s enc equivalent to factoring $pq$.
2. Rabin’s enc is hard to use: messages do not decode uniquely.
3. Blum-Williams modified Rabin’s Enc so that messages decode uniquely; but the set of messages you can send is small.
4. Hard to combine Blum-Williams modification with the padding needed to solve NY,NY problem.
5. Cracking Rabin Enc EQUIV factoring: but this is only if Eve has no other information.
6. If Eve can trick Alice into sending a chosen message, she can crack Rabin. So CPA-insecure.

Why is RSA used and not Rabin? either

1. The problems above make it not practical.
Rabin’s Encryption System and its Variants

1. Rabin’s enc equivalent to factoring $pq$.
2. Rabin’s enc is hard to use: messages do not decode uniquely.
3. Blum-Williams modified Rabin’s Enc so that messages decode uniquely; but the set of messages you can send is small.
4. Hard to combine Blum-Williams modification with the padding needed to solve NY,NY problem.
5. Cracking Rabin Enc EQUIV factoring: but this is only if Eve has no other other information.
6. If Eve can trick Alice into sending a chosen message, she can crack Rabin. So CPA-insecure.

Why is RSA used and not Rabin? either

1. The problems above make it not practical.
2. The problems above could have been gotten around but RSA just got to the market faster.
RSA Summary
Summary of RSA

1. PKCS-2.0-RSA is REALLY used!
2. There are many variants of RSA but all use the ideas above.
3. Factoring easy implies RSA crackable. TRUE.
4. RSA crackable implies Factoring easy: UNKNOWN.
5. RSA crackable implies Factoring easy: Often stated in expositions of crypto. They are wrong!
Summary of RSA

1. PKCS-2.0-RSA is REALLY used!
Summary of RSA

1. PKCS-2.0-RSA is REALLY used!
2. There are many variants of RSA but all use the ideas above.
Summary of RSA

1. PKCS-2.0-RSA is REALLY used!
2. There are many variants of RSA but all use the ideas above.
3. Factoring easy implies RSA crackable. TRUE.
Summary of RSA

1. PKCS-2.0-RSA is REALLY used!
2. There are many variants of RSA but all use the ideas above.
3. Factoring easy implies RSA crackable. TRUE.
4. RSA crackable implies Factoring easy: UNKNOWN.
Summary of RSA

1. PKCS-2.0-RSA is REALLY used!
2. There are many variants of RSA but all use the ideas above.
3. Factoring easy implies RSA crackable. TRUE.
4. RSA crackable implies Factoring easy: UNKNOWN.
5. RSA crackable implies Factoring easy: Often stated in expositions of crypto. They are wrong!
How Important Is Public Key?
Public key is mostly used for giving out keys to be used for classical systems.
This makes the following work:
Public key is mostly used for giving out keys to be used for classical systems. This makes the following work:

1. Amazon – Credit Cards
Public key is mostly used for giving out keys to be used for classical systems. This makes the following work:

1. Amazon – Credit Cards
2. Ebay – Paypal
Public key is mostly used for giving out keys to be used for classical systems.
This makes the following work:

1. Amazon – Credit Cards
2. Ebay – Paypal
3. Facebook privacy –
Public key is mostly used for giving out keys to be used for classical systems. This makes the following work:

1. Amazon – Credit Cards
2. Ebay – Paypal
3. Facebook privacy – just kidding, Facebook has no privacy.
   see: https://www.youtube.com/watch?v=cqggW08BW00
Used Everywhere

Public key is mostly used for giving out keys to be used for classical systems. This makes the following work:

1. Amazon – Credit Cards
2. Ebay – Paypal
3. Facebook privacy – just kidding, Facebook has no privacy. see: https://www.youtube.com/watch?v=cqggW08BW00
4. Every financial institution in the world.
Public key is mostly used for giving out keys to be used for classical systems. This makes the following work:

1. Amazon – Credit Cards
2. Ebay – Paypal
3. Facebook privacy – just kidding, Facebook has no privacy. see: https://www.youtube.com/watch?v=cqggW08BW00
4. Every financial institution in the world.
5. Military – though less is known about this.
Public Key Not Based on Factoring

What if Factoring can be done fast (quantum, fancy number theory, better hardware)?
Public Key Not Based on Factoring

What if Factoring can be done fast (quantum, fancy number theory, better hardware)?

1. Since 1960:

1.1 Math-advances have sped up factoring by 1000 times.
1.2 Hardware-advances have sped up factoring by 1000 times.
1.3 So Factoring has been sped up 1,000,000 times.

2. Factoring is in Quantum P, though making that practical seems a ways off.

3. There are now several Public Key Systems based on other hardness assumptions. See next slide.
Public Key Not Based on Factoring

What if Factoring can be done fast (quantum, fancy number theory, better hardware)?

1. Since 1960:
   1.1 Math-advances have sped up factoring by 1000 times.
What if Factoring can be done fast (quantum, fancy number theory, better hardware)?

1. Since 1960:
   1.1 Math-advances have sped up factoring by 1000 times.
   1.2 Hardware-advances have sped up factoring by 1000 times.
Public Key Not Based on Factoring

What if Factoring can be done fast (quantum, fancy number theory, better hardware)?

1. Since 1960:
   1.1 Math-advances have sped up factoring by 1000 times.
   1.2 Hardware-advances have sped up factoring by 1000 times.
   1.3 So Factoring has been sped up 1,000,000 times.

2. Factoring is in Quantum P, though making that practical seems a ways off.

3. There are now several Public Key Systems based on other hardness assumptions. See next slide.
What if Factoring can be done fast (quantum, fancy number theory, better hardware)?

1. Since 1960:
   1.1 Math-advances have sped up factoring by 1000 times.
   1.2 Hardware-advances have sped up factoring by 1000 times.
   1.3 So Factoring has been sped up 1,000,000 times.

2. Factoring is in Quantum P, though making that practical seems a ways off.
What if Factoring can be done fast (quantum, fancy number theory, better hardware)?

1. Since 1960:
   1.1 Math-advances have sped up factoring by 1000 times.
   1.2 Hardware-advances have sped up factoring by 1000 times.
   1.3 So Factoring has been sped up 1,000,000 times.

2. Factoring is in Quantum P, though making that practical seems a ways off.

3. There are now several Public Key Systems based on other hardness assumptions. See next slide.
Public Key Not Based on Factoring (cont)

Non-factoring based crypto systems:

1. Elliptic Curve Crypto
   Based on elliptic curves (duh).
   Classically this is better than RSA since is secure with smaller parameters. However, a quantum computer can crack it. Has been around since 1985 but hard math made it hard to use.

2. Lattice-based Crypto
   Based on certain lattice problems being hard to solve. Has been around since 1995.

3. Learning-With Errors (LWE)
   Based on the difficulty of learning a function from just a few points. Has been around since 2000. We will cover this later.

4. McEliece Public Key
   Based on error-correcting codes. Hardness assumption is that its hard to error-correct without the parity matrix. Has been around since 1978 but large keys made it a problem. We will cover this later.

None of these are widely used Why?
Non-factoring based crypto systems:

1. **Elliptic Curve Crypto** Based on elliptic curves (duh).
   Classically this is better than RSA since is secure with smaller parameters. However, a quantum computer can crack it. Has been around since 1985 but hard math made it hard to use.

2. **Lattice-based Crypto** Based on certain lattice problems being hard to solve. Has been around since 1995.

3. **Learning-With Errors (LWE)** Based on the difficulty of learning a function from just a few points. Has been around since 2000. We will cover this later.

4. **McEliece Public Key** Based on error-correcting codes. Hardness assumption is that it's hard to error-correct without the parity matrix. Has been around since 1978 but large keys made it a problem. We will cover this later.

None of these are widely used Why?
Non-factoring based crypto systems:

1. **Elliptic Curve Crypto** Based on elliptic curves (duh). Classically this is better than RSA since is secure with smaller parameters. However, a quantum computer can crack it. Has been around since 1985 but hard math made it hard to use.

2. **Lattice-based Crypto** Based on certain lattice problems being hard to solve. Has been around since 1995.
Non-factoring based crypto systems:

1. **Elliptic Curve Crypto** Based on elliptic curves (duh). Classically this is better than RSA since it is secure with smaller parameters. However, a quantum computer can crack it. Has been around since 1985 but hard math made it hard to use.

2. **Lattice-based Crypto** Based on certain lattice problems being hard to solve. Has been around since 1995.

3. **Learning-With Errors (LWE)** Based on the difficulty of learning a function from just a few points. Has been around since 2000. We will cover this later.
Non-factoring based crypto systems:

1. **Elliptic Curve Crypto** Based on elliptic curves (duh). Classically this is better than RSA since is secure with smaller parameters. However, a quantum computer can crack it. Has been around since 1985 but hard math made it hard to use.

2. **Lattice-based Crypto** Based on certain lattice problems being hard to solve. Has been around since 1995.

3. **Learning-With Errors (LWE)** Based on the difficulty of learning a function from just a few points. Has been around since 2000. We will cover this later.

4. **McEliece Public Key** Based on error-correcting codes. Hardness assumption is that its hard to error-correct without the parity matrix. Has been around since 1978 but large keys made it a problem. We will cover this later.
Public Key Not Based on Factoring (cont)

Non-factoring based crypto systems:

1. **Elliptic Curve Crypto** Based on elliptic curves (duh). Classically this is better than RSA since is secure with smaller parameters. However, a quantum computer can crack it. Has been around since 1985 but hard math made it hard to use.

2. **Lattice-based Crypto** Based on certain lattice problems being hard to solve. Has been around since 1995.

3. **Learning-With Errors (LWE)** Based on the difficulty of learning a function from just a few points. Has been around since 2000. We will cover this later.

4. **McEliece Public Key** Based on error-correcting codes. Hardness assumption is that its hard to error-correct without the parity matrix. Has been around since 1978 but large keys made it a problem. We will cover this later.

None of these are widely used
Non-factoring based crypto systems:

1. **Elliptic Curve Crypto** Based on elliptic curves (duh). Classically this is better than RSA since is secure with smaller parameters. However, a quantum computer can crack it. Has been around since 1985 but hard math made it hard to use.

2. **Lattice-based Crypto** Based on certain lattice problems being hard to solve. Has been around since 1995.

3. **Learning-With Errors (LWE)** Based on the difficulty of learning a function from just a few points. Has been around since 2000. We will cover this later.

4. **McEliece Public Key** Based on error-correcting codes. Hardness assumption is that its hard to error-correct without the parity matrix. Has been around since 1978 but large keys made it a problem. We will cover this later.

None of these are widely used Why?
Why Aren’t The NON-RSA Systems Used?

1. Chicken-and-egg problem: since they have not been out there and attacked, and fixed (like RSA) they are not considered secure.

2. Inertia.

3. Changing over would be expensive and a company has to ask itself, is it worth it?

4. There are other security issues that are more pressing. However, they are also not being dealt with.
Why Aren’t The NON-RSA Systems Used?

1. Chicken-and-egg problem: since they have not been out there and attacked, and fixed (like RSA) they are not considered secure.
Why Aren’t The NON-RSA Systems Used?

1. Chicken-and-egg problem: since they have not been out there and attacked, and fixed (like RSA) they are not considered secure.
2. Inertia.

However, they are also not being dealt with.
Why Aren’t The NON-RSA Systems Used?

1. Chicken-and-egg problem: since they have not been out there and attacked, and fixed (like RSA) they are not considered secure.
2. Inertia.
3.
4. Changing over would be expensive and a company has to ask itself, is it worth it?
Why Aren’t The NON-RSA Systems Used?

1. Chicken-and-egg problem: since they have not been out there and attacked, and fixed (like RSA) they are not considered secure.
2. Inertia.
3.
4. Changing over would be expensive and a company has to ask itself, is it worth it?
5.
6. There are other security issues that are more pressing.
Why Aren’t The NON-RSA Systems Used?

1. Chicken-and-egg problem: since they have not been out there and attacked, and fixed (like RSA) they are not considered secure.
2. Inertia.
3. Changing over would be expensive and a company has to ask itself, is it worth it?
4. There are other security issues that are more pressing. However, they are also not being dealt with.
Will These Systems be Used?

NIST (National Institute of Standards and Technology) solicited Quantum-Resistant Crypto Systems.
Will These Systems be Used?

NIST (National Institute of Standards and Technology) solicited Quantum-Resistant Crypto Systems.

Lattice-Based, LWE, and Code based all made it into the 2nd round:
Will These Systems be Used?

NIST (National Institute of Standards and Technology) solicited Quantum-Resistant Crypto Systems.

Lattice-Based, LWE, and Code based all made it into the 2nd round:

BILL, STOP RECORDING LECTURE!!!!!