Public Key
Cryptography:
NON-RSA Encryption
RSA

Let $n$ be a security parameter

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

   $$(m^e)^d \equiv m^{ed} \equiv m^{ed} \pmod{R} \equiv m^1 \pmod{R} \equiv m$$
Is RSA Hard to Crack?

Hardness Assumption for RSA: The following problem is hard: Given \((N, e, c)\) where \(N = pq\) and \(c \equiv m^e \pmod{N}\) for some \(m\), Find \(m\).

Objection: Hardness assumption not natural.
Objection: Hardness assumption does not have a long history of being tested.
We Want: An Encryption scheme based on Factoring being hard.

Is there one? Vote: Yes, No, or Unk?
Is RSA Hard to Crack?

Hardness Assumption for RSA: The following problem is hard: Given \((N, e, c)\) where \(N = pq\) and \(c \equiv m^e \pmod{N}\) for some \(m\), Find \(m\).

Objection: Hardness assumption not natural.
Objection: Hardness assumption does not have a long history of being tested.
We Want: An Encryption scheme based on Factoring being hard.

Is there one? **Vote:** Yes, No, or Unk?
Yes. Rabin Encryption.
Rabin Encryption
1. Solve $m^2 \equiv 1 \pmod{7}$
1. Solve \( m^2 \equiv 1 \pmod{7} \), \( m = 1, 6 \)
1. Solve \( m^2 \equiv 1 \pmod{7} \) \( m = 1, 6 \)

2. Solve \( m^2 \equiv 2 \pmod{7} \)
1. Solve $m^2 \equiv 1 \pmod{7}$ $m = 1, 6$
2. Solve $m^2 \equiv 2 \pmod{7}$ $m = 3, 4$
1. Solve $m^2 \equiv 1 \pmod{7}$ $m = 1, 6$
2. Solve $m^2 \equiv 2 \pmod{7}$ $m = 3, 4$
3. Solve $m^2 \equiv 3 \pmod{7}$
Math for Rabin Encryption – Square Roots Mod 7

1. Solve \( m^2 \equiv 1 \pmod{7} \) \( m = 1, 6 \)
2. Solve \( m^2 \equiv 2 \pmod{7} \) \( m = 3, 4 \)
3. Solve \( m^2 \equiv 3 \pmod{7} \) NONE

Since \( a^2 = (-a)^2 \) will always have, for all prime \( p \), \( p-1/2 \) elements of \( \{1, \ldots, p\} \) have sqrts mod \( p \).

\( p-1/2 \) elements of \( \{1, \ldots, p\} \) do not have sqrts mod \( p \).

Note: Computing Square Roots Mod \( n \) will mean determining if they exists and if so return all of them.
1. Solve $m^2 \equiv 1 \pmod{7}$ \hspace{1cm} m = 1, 6
2. Solve $m^2 \equiv 2 \pmod{7}$ \hspace{1cm} m = 3, 4
3. Solve $m^2 \equiv 3 \pmod{7}$ \hspace{1cm} NONE
4. Solve $m^2 \equiv 4 \pmod{7}$
1. Solve $m^2 \equiv 1 \pmod{7}$ $m = 1, 6$
2. Solve $m^2 \equiv 2 \pmod{7}$ $m = 3, 4$
3. Solve $m^2 \equiv 3 \pmod{7}$ NONE
4. Solve $m^2 \equiv 4 \pmod{7}$ $m = 2, 5$

Since $a^2 = (−a)^2$ will always have, for all prime $p$, $p−1$ elements of \{1, ..., $p$\} have sqrts mod $p$.

$p−1$ elements of \{1, ..., $p$\} do not have sqrts mod $p$.

Note: Computing Square Roots Mod n will mean determining if they exist and if so return all of them.
Math for Rabin Encryption – Square Roots Mod 7

1. Solve \( m^2 \equiv 1 \pmod{7} \) \( m = 1, 6 \)
2. Solve \( m^2 \equiv 2 \pmod{7} \) \( m = 3, 4 \)
3. Solve \( m^2 \equiv 3 \pmod{7} \) NONE
4. Solve \( m^2 \equiv 4 \pmod{7} \) \( m = 2, 5 \)
5. Solve \( m^2 \equiv 5 \pmod{7} \)

Since \( a^2 = (-a)^2 \) will always have, for all prime \( p \), \( p - 1 \) elements of \( \{1, \ldots, p\} \) have sqrts mod \( p \). \( p - 1 \) elements of \( \{1, \ldots, p\} \) do not have sqrts mod \( p \).

Note: Computing Square Roots Mod \( n \) will mean determining if they exists and if so return all of them.
Math for Rabin Encryption – Square Roots Mod 7

1. Solve $m^2 \equiv 1 \pmod{7}$ $m = 1, 6$
2. Solve $m^2 \equiv 2 \pmod{7}$ $m = 3, 4$
3. Solve $m^2 \equiv 3 \pmod{7}$ NONE
4. Solve $m^2 \equiv 4 \pmod{7}$ $m = 2, 5$
5. Solve $m^2 \equiv 5 \pmod{7}$ NONE
1. Solve $m^2 \equiv 1 \pmod{7}$ $m = 1, 6$
2. Solve $m^2 \equiv 2 \pmod{7}$ $m = 3, 4$
3. Solve $m^2 \equiv 3 \pmod{7}$ NONE
4. Solve $m^2 \equiv 4 \pmod{7}$ $m = 2, 5$
5. Solve $m^2 \equiv 5 \pmod{7}$ NONE
6. Solve $m^2 \equiv 6 \pmod{7}$

Since $a^2 = (-a)^2$ will always have, for all prime $p$, $p - 1$ elements of \{1, ..., $p$\} have sqrts mod $p$. $p - 1$ elements of \{1, ..., $p$\} do not have sqrts mod $p$.

Note: Computing Square Roots Mod n will mean determining if they exist and if so return all of them.
1. Solve $m^2 \equiv 1 \pmod{7}$ $m = 1, 6$
2. Solve $m^2 \equiv 2 \pmod{7}$ $m = 3, 4$
3. Solve $m^2 \equiv 3 \pmod{7}$ NONE
4. Solve $m^2 \equiv 4 \pmod{7}$ $m = 2, 5$
5. Solve $m^2 \equiv 5 \pmod{7}$ NONE
6. Solve $m^2 \equiv 6 \pmod{7}$ NONE

Since $a^2 = (-a)^2$ will always have, for all prime $p$, $p-1$ elements of $\{1, \ldots, p\}$ have sqrts mod $p$. $p-1$ elements of $\{1, \ldots, p\}$ do not have sqrts mod $p$.

Note: Computing Square Roots Mod $n$ will mean determining if they exist and if so return all of them.
Math for Rabin Encryption – Square Roots Mod 7

1. Solve $m^2 \equiv 1 \pmod{7}$ $m = 1, 6$
2. Solve $m^2 \equiv 2 \pmod{7}$ $m = 3, 4$
3. Solve $m^2 \equiv 3 \pmod{7}$ NONE
4. Solve $m^2 \equiv 4 \pmod{7}$ $m = 2, 5$
5. Solve $m^2 \equiv 5 \pmod{7}$ NONE
6. Solve $m^2 \equiv 6 \pmod{7}$ NONE

Since $a^2 = (\mathbf{-}a)^2$ will always have, for all prime $p$, $p - 1$ elements of \{1, \ldots, p\} have sqrts mod $p$. $p - 1$ elements of \{1, \ldots, p\} do not have sqrts mod $p$. Note: Computing Square Roots Mod $n$ will mean determining if they exist and if so return all of them.
1. Solve \( m^2 \equiv 1 \pmod{7} \) \( m = 1, 6 \)
2. Solve \( m^2 \equiv 2 \pmod{7} \) \( m = 3, 4 \)
3. Solve \( m^2 \equiv 3 \pmod{7} \) NONE
4. Solve \( m^2 \equiv 4 \pmod{7} \) \( m = 2, 5 \)
5. Solve \( m^2 \equiv 5 \pmod{7} \) NONE
6. Solve \( m^2 \equiv 6 \pmod{7} \) NONE

Since \( a^2 = (-a)^2 \) will always have, for all prime \( p \), \( \frac{p-1}{2} \) elements of \( \{1, \ldots, p\} \) have sqrts mod \( p \).
\( \frac{p-1}{2} \) elements of \( \{1, \ldots, p\} \) do not have sqrts mod \( p \).

Note: Computing Square Roots Mod \( n \) will mean determining if they exists and if so return all of them.
Theorem: \( c \) has a sqrt mod \( p \) iff \( c^{(p-1)/2} - 1 \equiv 0 \).

\[
c = m^2 \implies c^{(p-1)/2} \equiv (m^2)^{(p-1)/2} \equiv m^{p-1} \equiv 1.
\]

The equation \( x^{(p-1)/2} - 1 \equiv 0 \) has \((p-1)/2\) roots.
There are \((p-1)/2\) numbers that have sqrts. Hence
If \( c \) does not have a sqrt root then \( c^{(p-1)/2} - 1 \not\equiv 0 \).

Theorem: If \( p \equiv 3 \pmod{4} \) then easy to compute sqrt mod \( p \).
Given \( c \) if \( c^{(p-1)/2} \not\equiv 1 \) NO. If \( \equiv 1 \) then:

\[
(c^{(p+1)/4})^2 \equiv c^{(p+1)/2} \equiv c(c^{(p-1)/2}) \equiv c \times 1 \equiv c.
\]

So output \( c^{(p+1)/4} \) and other sqrt is \( p - c^{(p+1)/4} \).

Note: If \( p \equiv 1 \pmod{4} \) also easy to do sqrt.

Upshot: Sqrt mod a prime is easy!
What about sqrt mod a composite. Try these:

1. Solve $m^2 \equiv 9 \pmod{1147}$

2. Solve $m^2 \equiv 101 \pmod{1147}$

Answers: 3, 34, 1113, 1144.

Solve $m^2 \equiv 9 \pmod{1147}$: 3, 1147 - 3 = 1144 easy. If had 34 then 1147 - 34 = 1144 easy. But how to get 34?

Vote: Is finding sqrts mod $N$ hard? Yes, No, Unk?

Unk: Many computational questions in Number Theory are Unk.
What about sqrt mod a composite. Try these:

1. Solve $m^2 \equiv 9 \pmod{1147}$
2. Solve $m^2 \equiv 101 \pmod{1147}$

1. Solve $m^2 \equiv 9 \pmod{1147}$:
   - Answers: 3, 1144. If had 34 then 1147 − 34 = 1144 easy. But how to get 34?

Vote: Is finding sqrts mod N hard? Yes, No, Unk?

Unk: Many computational questions in Number Theory are Unk.
What about sqrt mod a composite. Try these:

1. Solve \( m^2 \equiv 9 \pmod{1147} \)

2. Solve \( m^2 \equiv 101 \pmod{1147} \)

1. Solve \( m^2 \equiv 9 \pmod{1147} \): Answers: 3, 34, 1113, 1144.
What about sqrt mod a composite. Try these:

1. Solve $m^2 \equiv 9 \pmod{1147}$
2. Solve $m^2 \equiv 101 \pmod{1147}$

1. Solve $m^2 \equiv 9 \pmod{1147}$: Answers: 3, 34, 1113, 1144.
2. Solve $m^2 \equiv 101 \pmod{1147}$:
What about sqrt mod a composite. Try these:

1. Solve $m^2 \equiv 9 \pmod{1147}$
2. Solve $m^2 \equiv 101 \pmod{1147}$

1. Solve $m^2 \equiv 9 \pmod{1147}$: Answers: 3, 34, 1113, 1144.
2. Solve $m^2 \equiv 101 \pmod{1147}$: Answers: Hmm.
What about sqrt mod a composite. Try these:

1. Solve \( m^2 \equiv 9 \pmod{1147} \)
2. Solve \( m^2 \equiv 101 \pmod{1147} \)

1. Solve \( m^2 \equiv 9 \pmod{1147} \): Answers: 3, 34, 1113, 1144.
2. Solve \( m^2 \equiv 101 \pmod{1147} \): Answers: Hmmm.

Solve \( m^2 \equiv 9 \pmod{1147} \): 3, \( 1147 - 3 = 1144 \) easy. If had 34 then \( 1147 - 34 = 1144 \) easy. But how to get 34?

**Vote:** Is finding sqrts mod \( N \) hard? Yes, No, Unk?
What about sqrt mod a composite. Try these:

1. Solve $m^2 \equiv 9 \pmod{1147}$
2. Solve $m^2 \equiv 101 \pmod{1147}$

1. Solve $m^2 \equiv 9 \pmod{1147}$: Answers: 3, 34, 1113, 1144.
2. Solve $m^2 \equiv 101 \pmod{1147}$: Answers: Hmmm.

Solve $m^2 \equiv 9 \pmod{1147}$: 3, $1147 - 3 = 1144$ easy. If had 34 then $1147 - 34 = 1144$ easy. But how to get 34?

Vote: Is finding sqrts mod $N$ hard? Yes, No, Unk?
Unk: Many computational questions in Number Theory are Unk.
\[ m^2 \equiv 101 \pmod{1147} \]

\[ 1147 = 31 \times 37 \]

\[ m^2 \equiv 101 \pmod{31} \]
\[ m^2 \equiv 8 \pmod{31} \]
\[ m \equiv \pm 15 \pmod{31} \]

\[ m^2 \equiv 101 \pmod{37} \]
\[ m^2 \equiv 27 \pmod{37} \]
\[ m \equiv \pm 8 \pmod{37} \]

One approach: Want number \( m \in \{1, \ldots, 1146\} \) such that
\[ m \equiv 15 \pmod{31} \]
\[ m \equiv 8 \pmod{37} \]

Use CRT to get:

\[ m = 15918 \equiv 1007 \pmod{1147} \]
By using $\pm 15 \pmod{31}$ and $\pm 8 \pmod{37}$ can find 4 sqrts.

**Upshot:** sqrts mod $N$ easy if know the factors of $n$.

**Upshot:** Always get 0 or 2 or 4 sqrts if mod $N = pq$.

What about finding sqrts mod $N$ where factors of $N$ are not known?
By using $\pm 15 \pmod{31}$ and $\pm 8 \pmod{37}$ can find 4 sqrts.

**Upshot:** sqrts mod $N$ easy if know the factors of $n$.
**Upshot:** Always get 0 or 2 or 4 sqrts if mod $N = pq$.

What about finding sqrts mod $N$ where factors of $N$ are not known?

Normally I would say

The problem of finding sqrt mod $N$ where the factors of $N$ are not known is believed to be hard.
By using $\pm 15 \pmod{31}$ and $\pm 8 \pmod{37}$ can find 4 sqrts.

**Upshot:** sqrts mod $N$ easy if know the factors of $n$.

**Upshot:** Always get 0 or 2 or 4 sqrts if mod $N = pq$.

What about finding sqrts mod $N$ where factors of $N$ are not known?

Normally I would say

The problem of finding sqrt mod $N$ where the factors of $N$ are not known is believed to be hard.

This time I can say something stronger.
Math for Rabin Encryption – Square Roots Mod $n$

How hard is sqrts mod $N$ when factors of $N$ not known?

Theorem: If finding sqrts mod $N$ is easy then factoring is easy.

1. Given $N = pq$ ($p, q$ unknown) want to factor it.
2. Pick a random $c$ and find its sqrts.
3. If it doesn't have $\geq 4$ sqrts then goto step 2.
4. The four sqrts are of the form $\pm x$ and $\pm y$. Now use $x, y$. We know that $x^2 \equiv y^2 \pmod{N}$.

$x^2 - y^2 \equiv 0 \pmod{N}$

$(x - y)(x + y) \equiv 0 \pmod{N}$

GCD($x - y, N$) or GCD($x + y, N$) likely factor.

Discuss: Why did I use $x, y$ instead of $x, -x$?
Math for Rabin Encryption – Square Roots Mod $n$

How hard is sqrts mod $N$ when factors of $N$ not known?

**Theorem:** If finding sqrts mod $N$ is easy then factoring is easy.

1. Given $N = pq$ ($p, q$ unknown) want to factor it.
2. Pick a random $c$ and find its sqrts.
3. If it doesn't have $\geq 4$ sqrts then goto step 2.
4. The four sqrts are of the form $\pm x$ and $\pm y$. Now use $x, y$. We know that

\[
x^2 \equiv y^2 \pmod{N}.
\]

\[
x^2 - y^2 \equiv 0 \pmod{N}
\]

\[
(x - y)(x + y) \equiv 0 \pmod{N}
\]

$GCD(x - y, N)$ or $GCD(x + y, N)$ likely factor.

**Discuss:** Why did I use $x, y$ instead of $x, -x$?
1. Finding primes is easy.
2. Squaring is easy.
3. If $N$ is factored then $\sqrt{N}$ mod $N$ is easy.
4. If $N$ is not factored then $\sqrt{N}$ mod $N$ is thought to be hard (equiv fo factoring).
Rabin’s Encryption Scheme

\( n \) is a security parameter

1. Alice gen \( p, q \) primes of length \( n \). Let \( N = pq \). Send \( N \).
2. Encode: To send \( m \), Bob sends \( c = m^2 \pmod N \).
3. Decode: Alice can find \( m \) such that \( m^2 \equiv c \pmod N \).
Rabin’s Encryption Scheme

$n$ is a security parameter

1. Alice generate $p, q$ primes of length $n$. Let $N = pq$. Send $N$.

2. Encode: To send $m$, Bob sends $c = m^2 \pmod{N}$.

3. Decode: Alice can find $m$ such that $m^2 \equiv c \pmod{N}$. OH! There will be two or four of them! What to do? Later.

PRO: Easy for Alice and Bob

BIG PRO: Factoring Hard is hardness assumption.

CON: Alice has to figure out which of the square roots is the correct message.

Caveat: If $m$ is English text then Alice can tell which one it is.

Caveat: If not. Hmmm.
Rabin’s Encryption Scheme

$n$ is a security parameter

1. Alice gen $p, q$ primes of length $n$. Let $N = pq$. Send $N$.
2. Encode: To send $m$, Bob sends $c \equiv m^2 \pmod{N}$.
3. Decode: Alice can find $m$ such that $m^2 \equiv c \pmod{N}$. OH! There will be two or four of them! What to do? Later.

**PRO:** Easy for Alice and Bob

**BIG PRO:** Factoring Hard is hardness assumption.

**CON:** Alice has to figure out which of the sqrts is correct message.

**Caveat:** If $m$ is English text then Alice can tell which one it is.

**Caveat:** If not. Hmmm.
How to Modify Rabin’s Encryption?

Let's look at mod $21 = 3 \times 7$.

$1^2, 8^2, 13^2, 20^2 \equiv 1$
$2^2, 5^2, 16^2, 19^2 \equiv 4$
$3^2, 18^2 \equiv 9$
$4^2, 10^2, 11^2, 17^2 \equiv 16$
$6^2, 15^2 \equiv 15$
$7^2, 14^2 \equiv 7$
$9^2, 12^2 \equiv 18$

**Question:** What do the red numbers have in common? **Discuss**
How to Modify Rabin’s Encryption?

Lets looks at mod $21 = 3 \times 7$.

$1^2, 8^2, 13^2, 20^2 \equiv 1$

$2^2, 5^2, 16^2, 19^2 \equiv 4$

$3^2, 18^2 \equiv 9$

$4^2, 10^2, 11^2, 17^2 \equiv 16$

$6^2, 15^2 \equiv 15$

$7^2, 14^2 \equiv 7$

$9^2, 12^2 \equiv 18$

**Question:** What do the red numbers have in common? **Discuss**

They all have square roots! They are all also on the RHS.
How to Modify Rabin’s Encryption?

Let's look at mod 21 = 3 * 7.

$1^2, 8^2, 13^2, 20^2 \equiv 1$

$2^2, 5^2, 16^2, 19^2 \equiv 4$

$3^2, 18^2 \equiv 9$

$4^2, 10^2, 11^2, 17^2 \equiv 16$

$6^2, 15^2 \equiv 15$

$7^2, 14^2 \equiv 7$

$9^2, 12^2 \equiv 18$

**Question:** What do the red numbers have in common? **Discuss**

They all have square roots! They are all also on the RHS.

What is it about 21 that makes this work?
A Theorem from Number Theory

Definition: A Blum Int is product of two primes \( \equiv 3 \pmod{4} \).
Example: \( 21 = 3 \times 7 \).

Notation: \( SQ_N \) is the set of squares mod \( N \). (Often called \( QR_N \).)
Example: If \( N = 21 \) then \( SQ_N = \{1, 4, 7, 9, 15, 16, 18\} \).

Theorem: Assume \( N \) is a Blum Integer. Let \( m \in SQ_N \). Then of the two or four sqrts of \( m \), only one is itself in \( SQ_N \).
Proof: Omitted. Note: (1) not that hard, and (2) in Katz book.

We use Theorem to modify Rabin Encryption.
(This modification by Blum and Williams.) \( n \) is a security parameter.

1. Alice gen \( p, q \) primes of length \( n \) such that \( p, q \equiv 3 \pmod{4} \). Let \( N = pq \). Send \( N \).

2. Encode: To send \( m \), Bob sends \( c = m^2 \pmod{N} \). Only send \( m \)'s in \( SQ_N \).

3. Decode: Alice can find 2 or 4 \( m \) such that \( m^2 \equiv c \pmod{N} \). Take the \( m \in SQ_N \).

**PRO:** Easy for Alice and Bob

**Biggest PRO:** Factoring Hard is hardness assumption.

**CON:** Messages have to be in \( SQ_N \).
Can Rabin’s Encryption Scheme Can Be Cracked?

$n$ is a security parameter

1. Alice gen $p, q$ primes of length $n$. Let $N = pq$. Send $N$.
2. Encode: To send $m$, Bob sends $c = m^2 \pmod{N}$.
3. Decode: Alice can find $m$ such that $m^2 \equiv c \pmod{N}$. Picks a poss out somehow.

Vote: Crackable, Uncrackable, Unk
Can Rabin’s Encryption Scheme Can Be Cracked?

$n$ is a security parameter

1. Alice gen $p, q$ primes of length $n$. Let $N = pq$. Send $N$.
2. Encode: To send $m$, Bob sends $c = m^2 \pmod N$.
3. Decode: Alice can find $m$ such that $m^2 \equiv c \pmod N$. Picks a poss out somehow.

Vote: Crackable, Uncrackable, Unk

Crackable:

Attack!: Eve picks an $m$ and tricks Alice into sending message $m$ via $m^2 \equiv c$. Eve is hoping that Bob will find another sqrt of $m^2$.

Say Alice gets $m'$. Then

$m^2 - (m')^2 \equiv 0 \pmod N$.

$(m - m')(m + m') \equiv 0 \pmod N$.

$m - m'$ or $m + m'$ may share factors with $N$ so do $gcd(m - m', N)$ and $gcd(m + m', N)$. Can factor $N$ and hence – game over!
What else to known

1. Alice may need to guess which of the 2 or 4 possible messages is the one to use, which is why its not used. Blum and Williams showed how to make the message unique, but by the time they did RSA was pervasive.

2. RSA and Rabin have similar issues which require padding-randomness

3. RSA has also had attacks as we’ve seen.

4. Rabin can be cracked with Chosen Plaintext Attack.

5. There is a variant of Rabin that thwarts the CPA but not provably equiv to factoring.

Alternate History: Had timing been different Rabin would have been the one everyone uses.
Goldwasser-Micali Encryption
Math Needed For Goldwasser-Micali Encryption

Definition

1. \( SQ_N \) is a number in \( \mathbb{Z}_N \) that does have a \( \sqrt{\cdot} \) mod \( N \)
2. \( NSQ_N \) is a number in \( \mathbb{Z}_N \) that does not have a \( \sqrt{\cdot} \) mod \( N \)
   (often called \( QNR_N \)).

Discuss: Let \( N = 35 \). Find all elements of \( SQ_N \) and \( NSQ_N \).
Math Needed For Goldwasser-Micali Encryption

1. Given \( n \), can gen random primes of length \( n \) easily.
2. Given \( p, q \) let \( N = pq \). Can gen a random \( z \in NSQ_N \) easily.
3. \( SQ_N \times SQ_N = SQ_N \).
4. \( NSQ_N \times SQ_N = NSQ_N \).
5. Given \( p, q, c \) can determine if \( c \) is in \( SQ_{pq} \) easily.
6. Given \( N, c \) determining if \( c \in SQ_N \) seems hard.

Discuss: Lets do some examples mod 35! (thats not a factorial, I’m excited about doing examples!)
Goldwasser-Micali Encryption

\( n \) is a security parameter. Will only send ONE bit. Bummer!

1. Alice gen \( p, q \) primes of length \( n \), and \( z \in NSQ_N \). Computes \( N = pq \). Send \((N, z)\).

2. Encode: To send \( m \in \{0, 1\} \), Bob picks random \( x \in \mathbb{Z}_N \), sends \( c = z^m x^2 \pmod{N} \). Note that:
   2.1 If \( m = 0 \) then \( z^m x^2 = x^2 \in SQ_N \).
   2.2 If \( m = 1 \) then \( z^m x^2 = zx^2 \in NSQ_N \).

3. Decode: Alice determines if \( c \in SQ \) or not. If YES then \( m = 0 \). If NO then \( m = 1 \).

BIG PRO: Hardness assumption natural – next slide.

BIG CON: Messages have to be 1-bit long.

TIME: For one bit you need \( 4 \log N \) steps.
**Goldwasser-Micali Encryption Hardness Assumption**

**SQ problem:** Given \((c, N)\) determine if \(c \in SQ_N\).

**Hardness Assumption:** The SQ problem is computationally hard.

**Note:** SQ problem has been studied by Number Theorists for a long time way before there was crypto. Hence it is a natural problem.

**PRO:** SQ is legit, well studied (unlike RSA assumption)

**CON:** SQ studied by Number Theorists, not computationally.

Back to Goldwasser-Micali:

**BIGGEST CON:** They take life one bit at a time. Really?
Blum-Goldwasser Encryption
Math You Need For Blum-Goldwasser Encryption

(You have seen this before but want to get us all on the same page.)

**Definition**

1. $SQ_N$ is a number in $\mathbb{Z}_N$ that does **have** a sqrt mod $N$
2. $NSQ_N$ is a number in $\mathbb{Z}_N$ that does **not** have a sqrt mod $N$
Math You Need For Blum-Goldwasser Encryption

(You have seen most of this before but want to get us all on the same page.)

1. Given $n$, can gen random primes of length $n$ easily.
2. Given $p, q$ let $N = pq$. Can gen a random $z \in NSQ_N$ easily.
3. $SQ_N \times SQ_N = SQ_N$.
4. $NSQ_N \times SQ_N = NSQ_N$.
5. Given $p, q, c$ can determine if $c$ is in $SQ_{pq}$ easily.
6. Given $N, c$ determining if $c \in SQ_N$ seems hard. More on that later.
7. $LSB(x)$ is the least sig bit of $x$. 
Blum-Goldwasser Enc. $n$ Sec Param, $L$ length of msg

1. Alice: $p, q$ primes len $n$, $p, q \equiv 3 \pmod{4}$. $N = pq$. Send $N$.

2. Encode: Bob sends $m \in \{0, 1\}^L$: picks random $r \in \mathbb{Z}_N$
   
   \begin{align*}
   x_1 &= r^2 \mod N \quad b_1 = \text{LSB}(x_1). \\
   x_2 &= x_1^2 \mod N \quad b_2 = \text{LSB}(x_2). \\
   &\vdots \\
   x_L &= x_{L-1}^2 \mod N \quad b_L = \text{LSB}(x_L). 
   \end{align*}

   Send $c = ((m_1 \oplus b_1, \ldots, m_L \oplus b_L), x_L)$.

3. Decode: Alice: From $x_L$ Alice can compute $x_{L-1}, \ldots, x_1$ by sqrt (can do since Alice has $p, q$). Then can compute $b_1, \ldots, b_L$ and hence $m_1, \ldots, m_L$.

**BIG PRO:** Hardness assumption – next slide.

**TIME:** For $L$ bits need $(L + 3) \log N$ steps. Better than Goldwasser-Micali.
The sequence $b_0, b_1, \ldots, b_L$ is the output of a known pseudorandom generator called BBS (Blum-Blum-Shub).

**BBS problem:** Given $x_L$ compute $b_L, \ldots, b_1$.

**Hardness Assumption:** BBS is computationally hard.

**PRO:** Natural in that BBS predates the cipher.

**CON:** BBS has not been around that long.
Correction to Diffie-Helman
Recall the Diffie-Helman Key Exchange

1. Alice finds a \((p, g)\), \(p\) of length \(n\), \(g\) gen for \(\mathbb{Z}_p\). Arith mod \(p\).
2. Alice sends \((p, g)\) to Bob in the clear (Eve can see it).
3. Alice picks random \(a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}\), sends \(g^a\).
4. Bob picks random \(b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}\), sends \(g^b\).
5. Alice: \((g^b)^a = g^{ab}\). Bob: \((g^a)^b = g^{ab}\). \(g^{ab}\) is shared secret.

Why does Alice choose random \(a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}\).
Why not pick \(a \in \{0, \ldots, p - 1\}\)? Discuss
Recall the Diffie-Helman Key Exchange

1. Alice finds a \((p, g)\), \(p\) of length \(n\), \(g\) gen for \(\mathbb{Z}_p\). Arith mod \(p\).
2. Alice sends \((p, g)\) to Bob in the clear (Eve can see it).
3. Alice picks random \(a \in \{\frac{p}{3}, \ldots, 2\frac{p}{3}\}\), sends \(g^a\).
4. Bob picks random \(b \in \{\frac{p}{3}, \ldots, 2\frac{p}{3}\}\), sends \(g^b\).
5. Alice: \((g^b)^a = g^{ab}\). Bob: \((g^a)^b = g^{ab}\). \(g^{ab}\) is shared secret.

Why does Alice choose random \(a \in \{\frac{p}{3}, \ldots, 2\frac{p}{3}\}\).

Why not pick \(a \in \{0, \ldots, p - 1\}\)? Discuss

If \(g\) is small and \(a\) is small then Eve can determine \(a\) from \(g^a\).
Recall the Diffie-Helman Key Exchange

1. Alice finds a \((p, g)\), \(p\) of length \(n\), \(g\) gen for \(\mathbb{Z}_p\). Arith mod \(p\).
2. Alice sends \((p, g)\) to Bob in the clear (Eve can see it).
3. Alice picks random \(a \in \left\{ \frac{p}{3}, \ldots, \frac{2p}{3} \right\}\), sends \(g^a\).
4. Bob picks random \(b \in \left\{ \frac{p}{3}, \ldots, \frac{2p}{3} \right\}\), sends \(g^b\).
5. Alice: \((g^b)^a = g^{ab}\). Bob: \((g^a)^b = g^{ab}\). \(g^{ab}\) is shared secret.

Why does Alice choose random \(a \in \left\{ \frac{p}{3}, \ldots, \frac{2p}{3} \right\}\).
Why not pick \(a \in \{0, \ldots, p − 1\}\)? Discuss.

If \(g\) is small and \(a\) is small then Eve can determine \(a\) from \(g^a\).
But: Eve can compute \(g^0, g^1, \ldots, g^L\) and if she sees any of those
she knows.
Example

\[ p = 1013 \]
\[ g = 5 \]

Eve computes ahead of time:
\[ 5^0 = 1 \]
\[ 5^1 = 5 \]
\[ 5^2 = 25 \]
\[ 5^3 = 125 \]
\[ 5^4 = 625 \]
\[ 5^5 = 86 \]
\[ 5^6 = 430 \]

If Eve sees Alice send any of 1, 5, 25, 125, 625, 86, 430 then she knows \( a \)

Nothing special about \( a \) being small.
Example

\[ p = 1013 \]
\[ g = 40 \]
\[ a \in \left\{ \frac{p}{3}, \ldots, \frac{2p}{3} \right\} = \{337, \ldots, 674\} \]

**Note:** We assume that Eve KNOWS these endpoints.

Eve computes
\[ 40^0 = 1 \]
\[ 40^1 = 40 \]
\[ 40^2 = 587 \]
\[ 40^3 = 181 \]
\[ 40^4 = 149 \]
\[ 40^5 = 895 \]
\[ 40^6 = 345 \]

If Eve sees Alice send any of 1, 40, 587, 181, 149, 895, 345 then she knows \( a \)

\( g \) was big, \( a \) was big. Didn’t help!
Example

\[ p = 1013 \]
\[ g = 40 \]
\[ a \in \left\{ \frac{p}{3}, \ldots, \frac{2p}{3} \right\} = \{337, \ldots, 674\} \]

Note: We assume that Eve KNOWS these endpoints.

Eve computes
\[ 40^0 = 1 \]
\[ 40^1 = 40 \]
\[ 40^2 = 587 \]
\[ 40^3 = 181 \]
\[ 40^4 = 149 \]
\[ 40^5 = 895 \]
\[ 40^6 = 345 \]

If Eve sees Alice send any of 1, 40, 587, 181, 149, 895, 345 then she knows \( a \)
g was big, \( a \) was big. Didn’t help!

Of course, Eve has to get VERY LUCKY.
1. Alice finds a \((p, g)\), \(p\) of length \(n\), \(g\) gen for \(\mathbb{Z}_p\). Arith mod \(p\).
2. Alice sends \((p, g)\) to Bob in the clear (Eve can see it).
3. Alice picks random \(a \in \{0, \ldots, p - 1\}\), sends \(g^a\).
4. Bob picks random \(b \in \{0, \ldots, p - 1\}\), sends \(g^b\).
5. Alice: \((g^b)^a = g^{ab}\). Bob: \((g^a)^b = g^{ab}\). \(g^{ab}\) is shared secret.

Eve comp \(g^0, g^1, \ldots, g^L\). If \(a \in \{0, \ldots, L\}\) Eve knows \(a\).

Not really a problem:

Either

1. If \(L\) is small then Eve would have to get LUCKY to find \(a\).
2. If \(L\) is large then Eve is doing LOTS OF computation.

Upshot: \(a, g\) small did not make attack much easier for Eve.
Is There Harm In Restricting $a, b$?

Have shown that requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ won’t help.
Is There Harm In Restricting $a, b$?

Have shown that requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ won’t help.

Will it hurt?
Vote: restricting $a, b$ will

1. make DH less secure
2. not have any affect.
Is There Harm In Restricting $a, b$?

Have shown that requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ won’t help.

Will it hurt?

Vote: restricting $a, b$ will

1. make DH less secure
2. not have any affect.

(1) Make DH less secure.
Key space is smaller, making it easier for Eve.
Diffie-Helman With Matrices and Lattices
DH and RSA Rely on Number Theory

(We are revisiting the guest lecture on this topic.)

1. DH and RSA rely on problems in Number Theory being hard.
2. If DL is easy then DH is cracked (not conversely).
3. If Factoring is easy then RSA is cracked (not conversely).
4. DL and Factoring are in Quantum-P.
5. If Quantum Computers ever become a reality than DH and RSA are cracked!

How worried should we be? Discuss
Is Quantum Computing Really a Threat?

My opinion

1. Quantum computers seem hard to build. I am mildly skeptical that they will ever pose a threat to DH or RSA.

2. Realize that I do not work in either Quantum Computing or Physics or anything else that would give me special insight.

3. Quantum computing is worth studying for the insight it gives into both quantum and computing. On this topic I know stuff. Or I know people who know stuff.

4. There are classical algorithms for DL and factoring that are forcing crypto people to increase their parameters for RSA and DH.

5. There are already attacks that have forced $e$ to be large, $N$ to be large.

Final Opinion: It is a good idea to find Key-exchange and public-key crypto that does not depend on number theory assumptions.
Post-Quantum Cryptography

This is a great title since

1. It has nothing to do with Quantum, so it's not that hard.
2. It sounds cool and can attract funding.

It just means that we are not using number-theory assumptions.
LWE Key Exchange

LWE means Learning With Errors. We will not need this.

1. We will discuss the protocol and how it works.
2. We will not discuss hardness assumptions.
LWE Key Exchange

1. Alice picks prime $p$, $n$, and $n \times n$ matrix $A$ over $\mathbb{Z}_p$. ($A$ may need to be invertible, not sure.)
2. Alice picks $\vec{y} \in \mathbb{Z}_p^n$ and $\vec{e}' \in \mathbb{Z}_p^n$. $\vec{e}'$ is “small”. Sends $\vec{y}A + \vec{e}'$.
3. Bob picks $\vec{x} \in \mathbb{Z}_p^n$ and $\vec{e} \in \mathbb{Z}_p^n$. $\vec{e}$ is “small”. Sends $A\vec{x} + \vec{e}$.
4. Alice computes $\vec{y}(A\vec{x} + \vec{e}) = \vec{y}A\vec{x} + \vec{y} \cdot \vec{e}$.
5. Bob computes $(\vec{y}A + \vec{e}')\vec{x} = \vec{y}A\vec{x} + \vec{x} \cdot \vec{e}'$.
6. They share $\vec{y}A\vec{x}$
LWE Key Exchange

1. Alice picks prime $p$, $n$, and $n \times n$ matrix $A$ over $\mathbb{Z}_p$. ($A$ may need to be invertible, not sure.)
2. Alice picks $\vec{y} \in \mathbb{Z}_p^n$ and $\vec{e}' \in \mathbb{Z}_p^n$. $\vec{e}'$ is “small”. Sends $\vec{y}A + \vec{e}'$.
3. Bob picks $\vec{x} \in \mathbb{Z}_p^n$ and $\vec{e} \in \mathbb{Z}_p^n$. $\vec{e}$ is “small”. Sends $A\vec{x} + \vec{e}$.
4. Alice computes $\vec{y}(A\vec{x} + \vec{e}) = \vec{y}A\vec{x} + \vec{y} \cdot \vec{e}$.
5. Bob computes $(\vec{y}A + \vec{e}')\vec{x} = \vec{y}A\vec{x} + \vec{x} \cdot \vec{e}'$.
6. They share $\vec{y}A\vec{x}$

Hey! That does not make sense! Neither one has $\vec{y}A\vec{x}$!
Recall:
Alice computes $\bar{y}(Ax + e) = \bar{y}Ax + (\bar{y} + \bar{e})$.
Bob computes $(\bar{y}A + \bar{e}')]x$. Close to $\bar{y}Ax$. 