Public Key Cryptography
Primes - There is an infinite number of them
Co-primes - \( \Phi(n) = n \left(1 - 1/P_1\right) \left(1 - 1/P_2\right) \cdots \left(1 - 1/P_k\right) \)

Candidate one way functions
- Addition - easy to invert \( ?(\text{subtraction}) \)
- Multiplication - cycles were short - easy to invert (division)

<table>
<thead>
<tr>
<th>exponentiation</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Always 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Always 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Three generates a maximally long cycle, is called a generator.
Thm: For any \( a \neq 0 \), \( a^{p-1} \equiv 1 \mod p \)

We want to use a large prime & a generator that has a long cycle.

Core idea of Diffie-Hellman
- Exponentiation is easy
- Discrete log is hard

\[ p = 7, \ g = 3 \]
\[ 3^x = 6 \mod 7; \ x \ ? = 3 \]
Worst case to solve for \( x \) is \( p-1 \) steps

\[ 3^5 \mod 7 \] - naively is \( p-1 \) steps. Actually \( \log_2 p \) steps

Repeated Squaring

\[ \text{e.g. } \ 3^4 - 4 \text{ operations} \]
\[ (3^2)^2 - 2 \text{ operations} \]
\[ 3^{64} \mod 127 - 6 \text{ operations} \]
Exponentiation by Repeated Squaring

\[ \text{Exp}(g, x, p) \]
if \( x = 0 \), return 1
else if \( x \) is even, return \((\text{Exp}(g, x/2, p))^2 \mod p\)
else if \( x \) is odd, return\((x \cdot \text{Exp}(g, (x - 1)/2, p)) \mod p\)

\( O(\log_2 p) \) -running time.

Diffie-Hellman Secret Sharing

<table>
<thead>
<tr>
<th>A Picks</th>
<th>B picks</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g )</td>
<td>( g )</td>
</tr>
<tr>
<td>( \mod p )</td>
<td>( \mod p )</td>
</tr>
<tr>
<td>computes</td>
<td>computes</td>
</tr>
<tr>
<td>( (g^b)^a \mod p )</td>
<td>( (g^b)^a \mod p )</td>
</tr>
<tr>
<td>( g^{ab} \mod p )</td>
<td>( g^{ab} \mod p )</td>
</tr>
</tbody>
</table>

Thm: If \( p \) is a prime then there exists \( \Phi(p-1) \) generators

Diffie-Hellman Encryption

Bob's public key is \(< g^b \mod p, g, p >\)
private key is \(< b >\)
Alice wants to send a message \( m \) to Bob.
She picks an \( a \) & sends \( g^a \mod p, m \cdot g^{ab} \mod p \)
Bob decrypts \( [m \cdot g^{ab}] / [(g^a)^b] = m \)
Nobody else can decrypt.

Diffie-Hellman Signature

Bob: pub key \(< g^b \mod p, g, p >\)
private key \(< b >\)
Bob wants to sign \( m \)
Bob generates \( S_m \) randomly
Sends: \( m, g^{S_m} \mod p; y = S_m + m \cdot b \mod (p-1) \)
(message with combined signature)
It is hard to extract \( b \) since \( S_m \) cannot be computed.
\( S_m + m \cdot b \) is essentially a random number.

Verification:
\[ g^y \neq (g^{S_m}) \cdot (g^b)^m \]

Can the same \( S_m \) be used for two messages?
\( m_1 \) & \( m_2 \), \( S_m \)
y1 = \( S_m + m_1 \cdot b \)
y2 = \( S_m + m_2 \cdot b \)
\( (y_1 - y_2) / (m_1 - m_2) = b \)
The answer is no because if the same \( S_m \) were used for two messages t
then B's private key would be compromised.

**RSA**

Relies on factoring.
\[ p \times q = n \]
\[ \varphi(n) = n(1 - 1/p)(1 - 1/q) = (p-1)(q-1) \]
Bob will compute \( d \) & \( e \).
\[ d \times e = 1 \mod \varphi(n) \]
public key \(<e,n>\)
private key \(<d>\)

**Encryption**
\[ m \mod n \]
Bob decrypt \( (m^e)^d = m^{ed} = m^{k \mod \varphi(n)+1} = (m^{\varphi(n)})^k \]
\( m' = m \mod n \)

**Thm:** if \( a \) & \( n \) are co-prime, then \( a^{\varphi(n)} \equiv 1 \mod n \)

**Signature**
\[ m^d \mod n \]
Hard to forge since no one other than Bob knows \( d \).
Verify: \( (m^d)^e \equiv m \)