Day 21 – Discrete Log Problem and Applications
Soloway-Strassen Prime Testing

- Pick at random \( a_1, a_2, \ldots, a_k \in \{1, 2, \ldots, n-1\} \).
- For each \( a = a_i \), check the two equalities
  \[
  \begin{align*}
  f(a, n) &= a^{\frac{n-1}{2}} \pmod{n} \\
  \gcd(a, n) &= 1.
  \end{align*}
  \]
  - If any of the two equalities fail, then \( n \) is not a prime.
  - If both equalities check out for all \( a_i \), we accept \( n \) as a prime.

The bigger \( k \) is, the higher the probability of reaching the correct conclusion.

This week HW: you test primality of some number \( n \) using 5 random integers \( a_1, \ldots, a_5 \).

In Wolfram Alpha:

\[
\begin{align*}
\text{\texttt{\textbackslash{}gcd[\textbackslash{}a,\textbackslash{}n]}} & \quad \text{\texttt{\textbackslash{}jacobisymbol[\textbackslash{}a,\textbackslash{}n]}} \\
\text{\texttt{\textbackslash{}powermod[\textbackslash{}a, \frac{\textbackslash{}n-1}{2}, \textbackslash{}n]}}
\end{align*}
\]
An attack on RSA - quadratic factoring

To factor n, Eve wants to find pairs $x > y$ such that

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

If neither $x + y$, nor $x - y$ are divisible by $n$, then $\gcd(x + y, n)$ and $\gcd(x - y, n)$ give nontrivial factors of $n$. This will definitely happen if $0 \leq y < x \leq (n - 1)/2$, as in this case

$$0 < x - y \leq n - 1 \quad \text{and} \quad 0 < x + y \leq n - 1.$$

The idea above allows one simple attack on RSA:

1. Randomly select distinct integers $x_1, x_2, \ldots, x_k \in [0, (n - 1)/2]$.
2. Compute

$$x_i^2 \pmod{n} \text{ for all } i.$$

3. If we find a pair $i \neq j$ with

$$x_i^2 \equiv x_j^2 \pmod{n},$$

then $\gcd(x_i + x_j, n)$ and $\gcd(x_i - x_j, n)$ are nontrivial factors of $n$.

More advanced attacking techniques come Math187B.

Also check out CSE 107.
Discrete Log Problem

Fix a prime $p$ and a number $a$ not divisible by $p$. For a given integer $\beta$, we want to find $x$ such that

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

Before, $3^x \equiv 1 \pmod{p} \Rightarrow x = \log_3(1) \rightarrow \text{unique soln.}$

Now, under modulo $18$, $3^x \equiv 11 \pmod{18}$ may be hard to solve, may not have soln.

May have more than 1 soln. \rightarrow \text{when will the soln exist and unique?}

**Definition 1.** We say that $a$ is a **primitive root mod $p$** if

$$a^i \not\equiv 1 \pmod{p} \text{ for all } i < p-1.$$ 

The first time $a^n \equiv 1 \pmod{p}$ is when $n = p-1$.

**Theorem 1.** If $g$ is a primitive root mod $p$, then

$$g^1, g^2, \ldots, g^{p-1} \pmod{p}$$

give a permutation of the integers $1, 2, \ldots, p-1$.

$$\{g^1, g^2, \ldots, g^{p-1}\} = \{1, 2, \ldots, p-1\}$$

3
Recall the Möbius function:
\[
\mu(n) = \begin{cases} 
(-1)^k & \text{if } n \text{ is the product of } k \text{ distinct primes} \\
0 & \text{otherwise}
\end{cases}
\]

Definition 2. The \textit{n-th cyclotomic polynomial} is given by
\[
\Phi_n(x) = \prod_{d \mid n} (1 - x^{\frac{n}{d}})
\]

**Theorem 2.** Each prime \( p \) has exactly \( \phi(p-1) \) primitive roots. In fact, \( a \) is a primitive root modulo \( p \) if and only if
\[
\Phi_{p-1}(a) \equiv 0 \pmod{p}
\]

\[\begin{array}{c|cccc}
\text{Ex} : \Phi_8(x) \quad n=8 \\
0 & 1 & 2 & 3 & 4 \\
1 & 0 & 1 & -1 & -1 \\
\end{array}\]

\[\Phi_8(x) = \frac{(1-x^2)(1-x^3)}{(1-x)(1-x^2)(1-x^3)} = \frac{1+x^3}{1+x} \]

\[\Phi_{10}(x) = \frac{(1-x^2)(1-x^5)}{(1-x^2)(1-x^5)} = \frac{1+x^5}{1+x} = x^4-x^3+x^2-x+1
\]

**Cyclotomic Polynomial** \([n]\)

Primitive roots mod \( p \) are the roots of \( \Phi_{p-1}(x) \)

\[\Phi_{10}(2) = 2^4 - 2^3 + 2^2 - 2 + 1 \equiv 11 \pmod{11} \neq 0
\]
\[\Phi_{10}(3) = 3^4 - 3^3 + 3^2 - 3 + 1 \equiv 61 \pmod{11} \neq 0
\]

**Question:** How to find all primitive roots if you know one of them?

**Theorem 3.** If \( a \) is a primitive root modulo the prime \( p \), then the set of primitive roots mod \( p \) is
\[
\{a^r : 1 \leq r \leq p-1, \gcd(r, p-1) = 1\}
\]

Raise primitive root \( a \) to all powers \( r \), where \( 1 \leq r \leq p-1 \) and \( r \) is coprime to \( p-1 \).
Can we find all primitive roots mod 11?

\[ g^k \mod 11 \]

<table>
<thead>
<tr>
<th>( k )</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>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g^k )</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>10</td>
<td>9</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Take the powers that are coprime to 10 = 11 - 1

4-primitive roots mod 11: \( \{2, 8, 7, 6\} \)
Solving equations using primitive roots

Under modulo $p = 11$, $2$ is a primitive root and the powers of $2$ are

\[
\begin{array}{cccccccccc}
2^1 & 2^2 & 2^3 & 2^4 & 2^5 & 2^6 & 2^7 & 2^8 & 2^9 & 2^{10} \\
1 & 2 & 4 & 8 & 5 & 10 & 9 & 3 & 6 & 1
\end{array}
\]

Solve the equation $9x \equiv 5 \pmod{11}$.

Suppose $x = 2^k$ then $9x \equiv 5 \pmod{11}$ will become

\[
\begin{align*}
2^k \cdot 2^3 & \equiv 2^4 \pmod{11} \\
2^{k+3} & \equiv 2^4 \pmod{11} \\
2^{k+3} & \equiv 1 \pmod{11}
\end{align*}
\]

By Fermat's Thm, $2^9 \equiv 1 \pmod{11}$, so

\[
2^{k+3} \equiv 1 \pmod{11} \Rightarrow 2^{k+3} \equiv 2^0 \pmod{10} \Rightarrow 2^{k+3} \equiv 0 \pmod{10}
\]

So $x = 2^k \equiv 0 \pmod{11} \Rightarrow 2^k \equiv 0 \pmod{11}$, which has no solution.

Solve the equation $7^x \equiv 5 \pmod{11}$.

\[
\begin{align*}
7^x & \equiv 5 \pmod{11} \iff (7^2)^x \equiv 2^4 \pmod{11} \\
\iff 2^{7x} & \equiv 2^4 \pmod{11} \\
\iff 2^{7x-4} & \equiv 1 \pmod{11}
\end{align*}
\]

By Fermat's Thm, $7^x - 4 \equiv 0 \pmod{10}$, so

\[
7^x \equiv 1 \pmod{10} \Rightarrow x = (7^{-1})(5)(3)(9) = 12 \equiv 2 \pmod{10}
\]

\[x = 2\]
Definition 3. Fix a prime \( p \) and a primitive root \( \alpha \) modulo \( p \). For a given integer \( \beta \), we want to find \( x \) such that
\[
\alpha^x \equiv \beta \pmod{p}.
\]

In this case, we write \( x = \log_{\alpha} \beta \pmod{p} \).

- We take \( \alpha \) to be a primitive root so that the discrete log problem can have a solution for any RHS value \( \beta \).
- Given a prime \( p \), it is fairly easy to find a primitive root.
- For small \( p \), we can compute discrete logs by exhaustive search.
- In general, computing discrete logs is hard (no known polynomial time algorithm)
- In other words, exponentiation \( \text{mod } p \) is believed to be a trapdoor function.
**El Gamal Cryptosystem**

1. Alice and Bob agree on a prime $p$ and a primitive root $r$ modulo $p$.

2. Alice comes up with her secret component $a$ and computes her public key $\alpha = r^a \mod p$.

3. Bob comes up with his secret component $b$ and computes his public key $\beta = r^b \mod p$.

4. Suppose Bob wants to send the message $X$ to Alice. He then picks a session key $k$, which is a random number in the interval $[2, p-2]$.

5. Bob then computes $U = r^k \mod p$ and $V = \alpha^k X = r^{ak}X \mod p$. He then sends $(U, V)$ to Alice.

6. To decrypt Bob’s message, Alice computes

$$U^{-a}V = r^{-ak}\alpha^k X = r^{-ak}X = X.$$
What can the enemy do?

Suppose that Eve intercepts the pair \((U, V)\), but to recover the message, Eve would need to know Alice’s secret component \(a\). Eve then has two choices, either

• solve for \(a\) from \(a = r^b \mod p\), or
• solve for \(k\) from \(U = r^k \mod p\) then compute \(a^{-1}V = a^{-k}a^bX = X\).

One of the major downsides of public key cryptosystems (e.g., RSA, El Gamal) is that they are relatively slow compared to modern ciphers (DES, AES). If you want to encrypt a good bit of data,

1. Use a public key cryptosystem to establish a secret key.
2. Encrypt the actual data using a fast cipher with the established secret key.