Admin:

- psct #1 - beh
  - #2 - in (see passwords)
  - #3 - out, due Mon 3/24 (like #04)(groups!)

Project ideas/wktop: 3/21 = next Friday

- evaluate encrypted email options (e.g., ekr-tux)
- overview of open-source crypto libraries
- Note: final projects to be posted online! (unless security vulnerability disclosed.)

Today:

More "crypto math":

- finding large primes
- one-time mac
- divisors, gcd, extended gcd, mult, inverses
- orders of elements
- finding generators
• How to find large (k-bit) random prime p?

  Generate & test: \textbf{do} \ p \leftarrow \text{random k-bit integer} \\textbf{until} \ p \text{ is prime}

• Works because primes are "dense":

  about \ \frac{2^k}{\ln(2^k)} \ k\text{-bit primes (Prime Number Theorem)}

  \Rightarrow \ one \ of \ every \approx 0.69k \ k\text{-bit integers is prime.}

• To test if a large randomly-chosen k-bit integer is prime, it suffices to test

  \ \ \ \ 2^{p-1} \equiv 1 \pmod{p}

  - This works with high probability (w.h.p.) for random \ p;

  doesn't work for \text{adversarially chosen} \ p.

  - See CLRS for Miller-Rabin primality test (randomized)

  - Technically, above gives "base-2 pseudoprime", but this

    is almost always prime

• \exists \text{ deterministic poly-time primality test (Agrawal, Kayal, Saxena 2002):}

  \text{Test} \ (x-a)^p = x^p - a \pmod{p} \quad \text{x variable}

  \text{which is true \text{iff} \ p \text{ is prime}

  \text{Test mod p & mod } x^r-1 \text{ for small } r \& \text{small a's.}

\text{Storage requirements ?! (see handout)}
One-time MAC (soln):

Idea:

\[ T = \text{MAC}_k(M) = ax + b \pmod{p} \]

Need two points to determine line; Eve hears just one: \((M,T)\)

\(p\) large prime (e.g. \(2^{16} + 51\))

Key \(K = (a, b)\) \(0 \leq a < p, 0 \leq b < p\) \((p^2\) keys\)

Security:

If adversary hears \((M,T)\) on the line, and replaces it with \((M', T')\) \([M' \neq M]\), then Bob accepts with probability \(1/p\).

PF: Hearing \((M,T)\) reduces set of possible keys to those satisfying \((\ast)\). Nonetheless, for each possible \(T'\), there is an \((a, b)\) satisfying both \((\ast)\) and

\[ T = aM' + b \pmod{p} \] \((\ast\ast)\)

All such keys are equally likely; Eve has no way to pick correct \(T'\).
Details:

For fixed $M, M' [M \neq M']$, fixed $T \ s.t.$

$$aM + b = T \pmod{p} \quad (\#)$$

For each $T'$, $\exists$ exactly one key $(a, b)$ s.t. $(\#)$ and

$$aM' + b = T' \pmod{p} \quad (\#\#)$$

holds:

$$a = \frac{(T-T')}{(M-M')} \pmod{p}$$
$$b = T - a \cdot M \pmod{p}$$

Thus Eve gains no information on $T' = \text{MAC}_K(M')$ by hearing $(M, T)$. Method is information-theoretically secure.

- True even if Eve can control $M$.
- Note that key $K$ is twice as long as message $M$. 

**Divisors**

- \( d \mid a \equiv "d \text{ divides } a" \) (evenly)
  \[ \equiv (\exists k) a = d \cdot k \]
- \( d \) is a divisor of \( a \) if \( d \neq 0 \) \& \( d \mid a \)
- \( (\forall d) d \mid 0 \)
- \( (\forall a) 1 \mid a \)
- If \( d \) is a divisor of \( a \) \& \( d \) is a divisor of \( b \), then \( d \) is a common divisor of \( a \) \& \( b \).
- The greatest common divisor of \( a \) \& \( b \) is the largest of their common divisors.
  
  [But \( \gcd(0, 0) = 0 \) by definition.]
- **Examples:**
  - \( \gcd(24, 30) = 6 \)
  - \( \gcd(5, 0) = 5 \)
  - \( \gcd(33, 12) = 3 \)
- **Def:** \( a \) \& \( b \) are relatively prime
  - if \( \gcd(a, b) = 1 \)
• Euclid's algorithm for computing \( \text{gcd}(a, b) \) \([a, b \geq 0]\):

\[
\text{gcd}(a, b) = \begin{cases} 
  a & \text{if } b = 0 \\
  \text{gcd}(b, a \mod b) & \text{else}
\end{cases}
\]

• Example:

\[
\begin{align*}
\text{gcd}(7, 5) &= \text{gcd}(5, 2) \\
&= \text{gcd}(2, 1) \\
&= \text{gcd}(1, 0) \\
&= 1
\end{align*}
\]

• Running time is \( \approx \lg(a) \cdot \lg(b) \) bit operations

(Polynomial running time, like multiplying.)
Theorem \((\forall a,b)(\exists x,y) \ ax + by = \gcd(a,b)\)

Proof "by example" \(a=7, b=5\)

\[
\begin{align*}
7 &= 7 \cdot 1 + 5 \cdot 0 \quad \{ \text{initial values} \} \\
5 &= 7 \cdot 0 + 5 \cdot 1 \\
2 &= 7 \cdot 1 + 5 \cdot (-1) \quad \{ \text{subtract 2 eqns} \} \\
1 &= 7 \cdot (-2) + 5 \cdot 3 \\
&= a \times + b \times y 
\end{align*}
\]

This is the "extended version of Euclid's algorithm".

Computing modular multiplicative inverses with Euclid's extended alg:

Suppose \(a \in \mathbb{Z}_p^*\) (so \(1 \leq a < p \) & \(\gcd(a,p)=1, p \text{ prime}(?))\)

How to compute \(a^{-1} \pmod{p}\)?

If \(p \text{ prime} \): \(a^{-1} = a^{p-2} \pmod{p}\)

Otherwise:

Find \(x, y \) s.t. \(ax + py = 1\)

so \(ax = 1 \pmod{p}\)

and \(x = a^{-1} \pmod{p}\)

Example: \(5^{-1} = 3 \pmod{7}\)
Order of elements (in $\mathbb{Z}_p^*$ or $\mathbb{Z}_n^*$):

Define: $\text{ord}_n(a) = \text{"order of } a, \text{ modulo } n\text{"}$

= least $t > 0$ s.t. $a^t = 1 \pmod{n}$

Recall Fermat's Little Theorem:

If $p$ prime, then $(\forall a \in \mathbb{Z}_p^*) \quad a^{p-1} = 1 \pmod{p}$

For general $n$, we have Euler's Theorem:

$(\forall n) \quad (\forall a \in \mathbb{Z}_n^*) \quad a^{\varphi(n)} = 1 \pmod{n}$

where $\mathbb{Z}_n^* = \{ a : \gcd(a, n) = 1 \}$

= multiplicative group modulo $n$

$\varphi(n) = |\mathbb{Z}_n^*|$

Example: $\mathbb{Z}_{10}^* = \{1, 3, 7, 9\}$

$\varphi(10) = 4$

$3^4 = 1 \pmod{10}$

Thus $\varphi(n)$ is well-defined for all $n$, &

$\text{ord}_n(a)$ is also well-defined.

Can we say more?
Example: \( \mod p = 7 \)

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

\[ \text{order}(1) = 1 \]
\[ \text{order}(2) = 3 \]
\[ \text{order}(3) = 6 \]
\[ \text{order}(4) = 3 \]
\[ \text{order}(5) = 6 \]
\[ \text{order}(6) = 2 \]

Def: \( \langle a \rangle = \{ a^i : i \geq 0 \} \) = subgroup generated by \( a \)

Examples: \( \langle a \rangle = \{ 2, 4, 1 \} \) (in \( \mathbb{Z}_7^* \))

Theorem: \( \text{order}(a) = |\langle a \rangle| \)

Theorem: If \( p \) prime: \( \text{order}_p(a) | (p-1) \).

Theorem: \( |\langle a \rangle| \mid |\mathbb{Z}_p^*| \)

or: \( \text{order}_n(a) \mid \varphi(n) \) equivalently.
Generators

Def: If \( \text{ord}_p(g) = p-1 \)
then \( g \) is a generator of \( \mathbb{Z}_p^\times \).
(i.e. \( \langle g \rangle = \mathbb{Z}_p^\times \))

Theorem: If \( p \) is a prime and
\( g \) is a generator mod \( p \), then
\( g^x = y \pmod{p} \)
has a unique solution \( x \) (\( 0 \leq x \leq p-1 \))
for each \( y \in \mathbb{Z}_p^\times \).

Def: \( x \) is the "discrete logarithm"
of \( y \), base \( g \), modulo \( p \).

\[
\begin{align*}
x &= 1 \ 2 \ 3 \ 4 \ 5 \ 6 \\
g^x &= 3 \ 2 \ 6 \ 4 \ 5 \ 1 \\
\text{for } g &= 3, \text{ modulo } 7.
\end{align*}
\]
**Theorem:** $\mathbb{Z}_n^*$ has a generator (i.e. $\mathbb{Z}_n^*$ is cyclic) iff $n$ is
\[ \begin{array}{c}
a, 4, p^m, \text{ or } 2p^m \\ \text{for some prime } p \text{ and } m \geq 1. \end{array} \]

**Theorem:** If $p$ is prime, the number of generators mod $p$ is $\phi(p-1)$

**Example:** $p = 11$

$\mathbb{Z}_{11}^*$ has $\phi(10) = 4$ generators
(they are 2, 6, 7, and 8).

How to find a generator mod a prime $p$?

In general, seems to require knowledge of factorization of $p-1$.

While factoring is hard, we can create primes for which factoring $p-1$ is trivial.
Def: If \( p \) \& \( q \) are both primes \( p \)

\[ p = 2q + 1 \]

then \( p \) is a "safe prime" and

\( q \) is a "Sophie Germain prime".

Examples:

\( p = 23, \ q = 11 \quad p = 11, \ q = 5 \)

\( p = 59, \ q = 29 \quad \ldots \)

Theorem: If \( p \) is a safe prime

then \( p - 1 = 2 \cdot q \)

so \((g \in \mathbb{Z}_p^\times) \) \( \text{ord}_p(g) \in \{1, 2, q, 2q, 3, 2q, 3 \} \).

It is not hard to find safe primes. ("Probability"

that a prime \( p \) is safe is \( \approx 1/\ln(p) \), empirically.)

Can test if \( g \) is a generator mod \( p = 2q + 1 \) easily:

check that \( g^{p-1} = 1 \pmod{p} \) \( \checkmark \) by Fermat

\& \( g^q \neq 1 \pmod{p} \) \( \left[ \text{ord}_p(g) = 2 \right] \)

\& \( g^{2q} \neq 1 \pmod{p} \) \( \left[ \text{ord}_p(g) = q \right] \)

then \( \text{ord}_p(g) = p - 1 \).
We can use "generate & test" again: (for "safe prime" \( p \))

\[
\text{do } g \leftarrow \mathbb{Z}_p^* \\
\text{until } \text{order}_p(g) = p - 1
\]

Generators are quite common:

**Theorem:** If \( p = 2q + 1 \) is a "safe prime"

then \# generators mod \( p \)

\[
= \varphi(p-1) = q - 1 \quad \text{(almost half of them!)}
\]

(In general:

**Theorem:** If \( p \) prime, then

\[
\# \text{ generators mod } p \\
\leq \varphi(p-1) \\
\geq \frac{p-1}{6 \ln \ln (p-1)}
\]

So generate & test works well for finding generators modulo a safe prime \( p \), or modulo any prime \( p \) for which you know \( \varphi(p-1) \).
**Common public-key setups**

**Public system parameters**

- \( P \) large prime (e.g., 1024 bits)
- \( g \) generator mod \( P \)

Alice chooses \( 0 \leq x < P-1 \) as her **secret key**.

Alice publishes \( y = g^x \pmod{P} \) as her **public key**.

- Secrecy of \( x \) protected by difficulty of computing discrete log
  \[
  \log_g P (y) = x
  \]

- Commonly assumed that discrete log problem (DLP) is infeasible for \( P \) large & random, or \( P \) large safe prime.
  (Appears to be roughly as hard as factoring a large integer of the same size as \( P \).
  This is observation, not a theorem.)