Alin Tomescu, CSE408
Tuesday, March 8th, Lecture #11

Public key cryptography

Diffie-Hellman key exchange

- You have Alice and Bob, as usual. Eve's there too, listening as always.
- Alice and Bob agree on a large number p and a base g between 2 and p. Eve can see both p and g.
- Alice picks a random number a and computes $x = g^a \bmod p$.
  - Alice sends x to Bob. (Note that a remains secret to Alice)
- Bob picks b random and computes $y = g^b \bmod p$.
  - Bob sends y to Alice. (Note that b remains secret to Bob)
- Alice computes $k = y^a \bmod p = g^{ab} \bmod p$
- Bob computes $k' = x^b \bmod p = g^{ab} \bmod p$
- Alice and Bob now both have a secret key $k = k'$, which is around 2048 bits large
- Given $g^a \bmod p$ and $g^b \bmod p$, Eve will not be able to easily compute $g^{ab} \bmod p$.

Alice and Bob can start sending message back and forth by encrypting messages using their secret key k.

The discrete logarithm problem

Motivation: Eve has $x = g^a\ mod\ p$ and $y = g^b\ mod\ p$ but she can't compute $k = g^{ab}\ mod\ p$. Why?

Eve has to solve the **discrete log problem** in order to figure out the value of a or b. Eve can't just take $\log x$ and/or $\log y$ because x and y were obtained by raising g to a certain power modulo p. If modular arithmetic were not used, then a simple $\log$ would have obviously worked.

Discrete log problem: Given g, p and $g^a\ mod\ p$ compute a.

To the best of our knowledge, this is a hard problem. The best algorithms run roughly better than $\sqrt{p}$, since p is like 2^{2000} large, then you can see how the square root of that is still a huge number.

Eve can't do a discrete logarithm, but maybe she can do something else. For instance, Diffie-Hellman is vulnerable to the Omen problem: it is not clear that to compute $g^{ab}\ mod\ p$ you actually need to know a or b.

Man in the middle attack (MITM)

What if Eve tampers with messages?

- She can send $g^e\ mod\ p$ to Bob (by intercepting Alice's $g^a\ mod\ p$)
- She can send $g^e\ mod\ p$ to Alice (by intercepting Bob's $g^b\ mod\ p$)
- She can compute both $k_A = g^{ea}\ mod\ p$ and $k_B = g^{eb}\ mod\ p$, which will be the keys Alice and Bobs will end up computing with the bad information they got from Eve.
  - Note that k_A and k_B will be different, therefore Eve has to extra work.
- Anytime Alice or Bob send a message, Eve has to intercept it, decrypt it using the right key, re-encrypt it under the other key, and finally let the altered ciphertext pass through to its destination.

Alin Tomescu, CSE408
Tuesday, March 8th, Lecture #11

Number theory

Terms to know: prime number, composite number, 1 is neither prime nor composite, greatest common divisor (always greater than 1), a and b are relatively prime or coprime $\Leftrightarrow \gcd(a, b) = 1$.

Notation: $a|b$ means a divides b, the same thing as saying that $b \ mod \ a = 0$

The Extended Euclidian Algorithm

Algorithm: Given a and b, compute $m = \gcd(a, b)$ and x, y such that $ax + by = m$

Essentially, the algorithm starts with two simple equalities:

$$1 \times a + 0 \times b = a$$
$$0 \times a + 1 \times b = b$$

By manipulating these equalities, the algorithm will obtain the coefficients x and y, such that:

$$x \times a + y \times b = \gcd(a, b)$$

How do we get to this final, quite useful, equality? By noting that if we have:

$$x_1 \times a + y_1 \times b = c_1$$
$$x_2 \times a + y_2 \times b = c_2$$

Then, it is also true that:

$$(x_1 \times a + y_1 \times b) - q(x_2 \times a + y_2 \times b) = c_1 - q \times c_2$$

Therefore, starting with a and b, we compute q and r such that $a = q \times b + r$ (so we divide a by b, getting $q = \lfloor a/b \rfloor$ and $r = a \bmod b$).

- Then, we get the next equation as $(x_1 \times a + y_1 \times b) - q(\ x_2 \times a + y_2 \times b) = r$
- We always repeat the process with the last two equations until we finally get a remainder $r = 0$
  - The remainder before it will be the gcd, and on the same row we'll have the x_i and y_i values such that $x \times a + y \times b = \gcd(a, b)$

Example: $a = 96$ and $b = 38$

x	y	$c = ax + by$	c_{old}/c_{new} (written as $c_{new} \times \boldsymbol{q} + r = c_{old}$)
1	0	$1 \times 96 + 0 \times 38 = 96$	
0	1	$0 \times 96 + 1 \times 38 = 38$	$\mathbf{2} \times 38 + 20 = 96$
The goal now is to get r to equal 0 and get the corresponding x and y values. We do this by subtracting/adding multiples of the rows in this table. Since $\mathbf{2} \times 38 + 20 = 96$, we subtract $2r_2$ from r_1. We repeat.			
1	-2	$1 \times 96 - 2 \times 38 = 20$	$\mathbf{1} \times 20 + 18 = 38$
-1	3	$-1 \times 96 + 3 \times 38 = 18$	$\mathbf{1} \times 18 + 2 = 20$
2	-5	$2 \times 96 - 5 \times 38 = 2$	$\mathbf{9} \times 2 + 0 = 18$
x	y	$r = 0$	**Done.**

Now, the algorithm has finished with the following output:

$$\gcd(a, b) = \gcd(96{,}38) = 2$$

Also, the algorithm gave you $x = 2$ and $b = -5$ such that $ax + by = \gcd(a, b)$, specifically now you have:

$$2 \times 96 - 5 \times 38 = 2$$

Description (pseudo-code):

An easy to follow implementation:

```
eea(a, b) : gcd(a, b), x, y, such that a*x + b*y = gcd(a, b)

let x[], y[], and d[] be the columns of our Extended Euclidian Algorithm table

eea(a, b, 0) {
   if (a < b) swap(a, b);

   // The invariant is that x[i]*a + y[i]*b = d[i]
   x[0] = 1; y[0] = 0; d[0] = a;
   x[0] = 0; y[0] = 1; d[1] = b;
}

eea(a, b, i) {
   d[i] = d[i-2] mod d[i-1]

   q = d[i-2] / d[i-1];
   x[i] = x[i-2] - q * x[i-1];
   y[i] = y[i-2] - q * y[i-1];
}

eea(a, b) {
   for(int i = 0; d[i] != 0; i++)
   {
       eea(a, b, i);
   }

   return d[i-1], x[i-1], y[i-1];
}
```

Proof of correctness:

Suppose $c_0 = a, c_1 = b$ and $a > b$ so Let $c_1, c_2, \ldots$ be as computed in the algorithm.

$$c_2 = c_0 \; mod \; c_1 = c_0 - q_1 c_1$$
$$c_3 = c_1 \; mod \; c_2 = c_1 - q_2 c_2$$
$$c_{n-1} = c_{n-3} \; mod \; c_{n-2} = c_{n-3} - q_{n-2} c_{n-2}$$
$$0 = c_n = c_{n-2} \; mod \; c_{n-1} = c_{n-2} - q_{n-1} c_{n-1}$$
$$\Rightarrow c_{n-1} | c_{n-2} \Rightarrow c_{n-1} | c_{n-3} \Rightarrow \cdots \Rightarrow c_{n-1} | \; a \text{ and } c_{n-1} | \; b$$

Observation, $\gcd(a, b)$ divides all $c_i \Rightarrow \gcd(a, b) \leq c_{n-1}$

Also,

$$\forall i, \gcd(a, b) \mid c_i$$
$$c_{n-1} | \; a \text{ and } c_{n-1} | \; b$$

Therefore, since $\gcd(a, b) \leq c_{n-1}$

$$\mathbf{gcd}(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{c_{n-1}}$$

Alin Tomescu, CSE408
Tuesday, March 8th, Lecture #11

Modular arithmetic

There are two notions of mod:

- mod as in the remainder (division)
  - For instance, $n\ mod\ q$ is the remainder r, such that $n = pq + r$, for some p.
    - Example: $10\ mod\ 3 = 1$, $7\ mod\ 4 = 3$, $100\ mod\ 12 = 4$
- mod as an equivalence relation

Modulus equivalence relation

Congruency mod n

Definition: We say that two numbers a and b are **congruent $\boldsymbol{mod\ n}$**, and we write $a \equiv b\ (mod\ n)$ if, and only if, n divides $a - b$.

$$a \equiv b \pmod{n} \Leftrightarrow n \mid a - b$$

Intuition: Two numbers a and b are congruent mod n if they both have the same remainder after dividing them by n.

$$a \equiv b\ (mod\ n) \Leftrightarrow a \bmod n = b \bmod n$$

Example: $71 \equiv 63 \equiv 7 \equiv -1\ (mod\ 8)$

Theorem: If $a \equiv b\ (mod\ n)$ and $c \equiv d\ (mod\ n)$ then $a + c \equiv b + d\ (mod\ n)$ and $ac \equiv bd\ (mod\ n)$.

Proof (part I):

We have to prove that: $n|(a + c) - (b + d) \Leftrightarrow a + c \equiv b + d\ (mod\ n)$

$a \equiv b\ (mod\ n)$ and $c \equiv d\ (mod\ n) \Rightarrow n|a - b$ and $n|c - d$

Adding the last two division properties, we get $n|a - b + c - d \Leftrightarrow n|(a + c) - (b + d)$

Proof (part II):

Let's prove that $ac \equiv bc\ (mod\ n)$. Easy. $n|a - b \Rightarrow n|c(a - b) \Leftrightarrow n|ac - bc$
Let's prove that $bc \equiv bd\ (mod\ n)$. Easy. $n|c - d \Rightarrow n|b(c - d) \Leftrightarrow n|bc - bd$
Therefore, $ac \equiv bc \equiv bd\ (mod\ n)$

Theorems:

$$a^b \equiv (a\ mod\ n)^b\ (mod\ n)$$
$$a^{b+c} \equiv a^{c+b}\ (mod\ n)$$
$$\left(a^b\right)^c \equiv (a^c)^b\ (mod\ n)$$

Theorem: $(a\ mod\ n + b\ mod\ n) \equiv a + b\ (mod\ n)$.

Note: First two mods are "remainder mods", third mod are "equivalence mods".

First, let's prove that:

$$a \equiv (a\ mod\ n)\ (mod\ n)$$

We have to show that:

$$n|(a - a\ mod\ n)$$

Let $r = a\ mod\ n$, where $a = qn + r \Rightarrow r = a - qn$

$n|(a - a\ mod\ n) \Leftrightarrow n|\big(a - (a - qn)\big) \Leftrightarrow n|qn$, which is true

Alin Tomescu, CSE408
Tuesday, March 8th, Lecture #11
So now, let's prove that:

$$(a \, mod \, n + b \, mod \, n) \equiv a + b \, (mod \, n) \Leftrightarrow$$
$$a - q_1 n + b - q_2 n \equiv a + b \, (mod \, n) \Leftrightarrow$$
$$n | a - q_1 n + b - q_2 n - (a + b) \Leftrightarrow$$
$$n | \big(n(q_1 - q_2)\big), \text{ which is true.}$$

Example: $116 \times 47 \, mod \, 11 = -5 \times 3 \, mod \, 11 = -15 \, mod \, 11 = 7 \, mod \, 11$

Tip: To go from positive to negative numbers in an equivalence class, the negative equivalent q of a positive number p. modulo n is $q = -\big(n - (p \, mod \, n)\big)$.

Example: Take $p = 116, n = 11$, then $116 \, mod \, 11 = 6$. So $q = -(11 - 6) = -5$.

Modular exponentiation

How do we go about computing these big powers in the Diffie-Hellman key exchange protocol?

$$116^{97} \, mod \, 11 = 6^{47} \, mod \, 11 = reduce \, the \, 47 \, somehow?$$

Theorem: If $\gcd(a, b) = 1$, then there exists y such that $ay \equiv 1 \, mod \, b \Leftrightarrow y = a^{-1} \, mod \, b$.

Proof: By the extended Euclidian algorithm, since $\gcd(a, b) = 1$, there exists x, y such that $ay + bx = 1 \Rightarrow b | ay - 1 \Rightarrow ay \equiv 1 \bmod b$.