Alin Tomescu, CSE408
Tuesday, March 17th, Lecture #14

Trapdoor one-way functions and PKE-CPA

El Gamal encryption

Alice has a public g and p and she picks a secret a. She publishes $g^a \bmod p$. Since g and p are public, someone else might've maliciously picked g and p. The interesting thing is that even the people who picked g and p cannot figure out a from $g^a \bmod p$.

$$f(a) = g^a \bmod p$$

With the right, g and p, and for a not so large a, $f(a)$ is injective and invertible, but its inverse is really hard to compute so it's a **one way function**.

Definition: f is a (t, ε)-one-way function if $\forall A$ running in time $\leq t$ we have:

$$Adv\ A = \Pr[A(y) = x \text{ where } f(x) = y] \leq \varepsilon$$

Modular exponentiation is a **one-way permutation.**

RSA

Alice picks primes p and q, and computes e, d such that $ed = 1\ mod\ \varphi(n)$, $n = pq$, and sends (n, e) to Bob

Bob sends his message m as $m^e \bmod n$. Alice can compute m as $(m^e \bmod n)^d \bmod n$

So, Bob can compute $f_{e,n}(x) = x^e mod\ n$ and Alice can compute $f_{e,n}^{-1}(x) = x^d\ mod\ n$.

Definition: A (t, q, ε)-trap door one way permutation (**TOWP**) is a one way permutation with a trap-door.

RSA is a **trap door one way permutation**.

RoR security with PKE

Eve will have access to the public key so she can encrypt messages herself. Her goal is to tell whether Alice is sending real or random messages to Bob.

Even in the PK (public-key encryption) world, we still need randomized encryption because Eve can always send Alice a message, get the message encrypted back from her, and then encrypt the message herself comparing if her encryption output is the same as what she received from Alice. If they are the same, then she can tell she's in the real world. Therefore, you need a randomized encryption algorithm that produces different encryptions of the same message m.

Definition: A public key encryption scheme is (t, q, ε) IND-CPA-secure if $\forall A$ running in time $\leq t$ and making $\leq q$ queries has:

$$Adv\ A = \left|\Pr[A^{E_{pk}}(pk) = 1\ ] - \Pr\left[A^{E_{pk}*\$}(pk) = 1\right]\right| \leq \varepsilon$$

Theorem: If (E, D) is $(t, 1, \varepsilon)$-secure then it is $(t, q, (q+1)\varepsilon)$-secure. This depends on the fact that the adversary can perform encryption himself.

Alin Tomescu, CSE408
Tuesday, March 17th, Lecture #14

Proof by contra-positive: Suppose I have an adversary A that can make q queries to an oracle that computes either E_{pk} or $E_{pk} * \$$ and break it (tell which one it is). Then we'll show that we can build another adversary that breaks it in one query.

The new adversary will use a family of oracles O_i defined as follows:

$$O_i(x_j) = \begin{cases} E_{pk}(x_j) \; if \; j < i \\ O(x_j) \; if \; j = i \\ E_{pk}\left(\$(x_j)\right) \; if \; j > i \end{cases}$$

What's the behavior of the attacker when we have O_i and O_{i+1}?

$$O_i(x_j) = \begin{cases} E_{pk}(x_j) \; if \; j < i \\ O(x_j) \; if \; j = i \\ E_{pk}\left(\$(x_j)\right) \; if \; j = i+1 \\ E_{pk}\left(\$(x_j)\right) \; if \; j > i+1 \end{cases}$$

$$O_{i+1}(x_j) = \begin{cases} E_{pk}(x_j) \; if \; j < i \\ E_{pk}(x_j) \; if \; j = i \\ O(x_j) \; if \; j = i+1 \\ E_{pk}\left(\$(x_j)\right) \; if \; j > i+1 \end{cases}$$

$$O_i(x_j) \overset{t,q}{\underset{\varepsilon}{\sim}} O_{i+1}(x_j)$$

So… dead end! We're not doing this by contrapositive.

$E_{pk}(\$) = O_0 \overset{t,q}{\underset{\varepsilon}{\sim}} O_1$ because if you could tell $O_0 = E_{pk}(\$)$ and $O_1 = \begin{cases} O(x_j) \; if \; j = 1 \\ E_{pk}\left(\$(x_j)\right) \; if \; j > 1 \end{cases}$ apart essentially you are telling the difference between what O returns: E_{pk} and $E_{pk}(\$)$ which are (t, q, ε)-indistinguishable.

$$O_0 \overset{t,q}{\underset{\varepsilon}{\sim}} O_1 \overset{t,q}{\underset{\varepsilon}{\sim}} O_2 \sim \ldots \sim O_{q+1} = E_{pk}$$

Therefore, applying transitivity, $E_{pk}(\$) \overset{t,q}{\underset{(q+1)\varepsilon}{\sim}} E_{pk}$

The birthday attack on the discrete log problem

Our goal: Given $y = g^x \bmod p$, g and p, compute x.

Randomized algorithm: Baby-step, giant-step algorithm.

1. Pick random $r_1, r_2, \ldots, r_q$ and $s_1, s_2, \ldots, s_q$ relatively prime to p
2. Compute $g^{r_1}, g^{r_2}, \ldots, g^{r_q}$ mod p and $y^{s_1}, \ldots, y^{s_q}$ mod p

3. Suppose that $g^{r_i} = y^{s_j} \bmod p \Rightarrow g^{r_i/s_i} = y \bmod p$
4. $g^x = g^{x \bmod \varphi(p)} \bmod p$
5. What is $r_i s_j^{-1} \bmod \varphi(p)$?

How big should q be to ensure a collision? $E[\#col] = \frac{q^2}{p}$, so $q = \sqrt{p}$

Quadratic residues attack on El-Gamal

Definition: x is a quadratic residue mod p if $\exists y$ such that $y^2 = x \bmod p$.

Example: $p = 13$

$1^2 = 1, 2^2 = 4, 3^2 = 9$, … and so on, notice that the sequence of squares looks like this:

$$1,4,9,3,12,10,10,12,3,9,4,1$$

Definition: The Legendre symbol $\left(\frac{x}{p}\right) = \begin{cases} 1, if\ x\ is\ a\ QR\ mod\ p \\ -1, otherwise \end{cases}$, where p is a prime

Fact: $\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}} \bmod p$

If x is a square then $x^{\frac{p-1}{2}} = x'^{p-1}$ which is equal to 1. If x is not a square then it $x^{\frac{p-1}{2}}$ could be either 1 or -1. Well, it turns out it has to be -1 with a theorem.

El Gamal encryption:

$$E(m) = (g^r, (g^a)^r m)$$

$$\left(\frac{xy}{p}\right) = \left(\frac{x}{p}\right)\left(\frac{y}{p}\right)$$

By analyzing the Jacobi-symbols $\left(\frac{g^a}{p}\right)$, $\left(\frac{g^r}{p}\right)$ and $\left(\frac{g^{ar}m}{p}\right)$ you can tell whether m is a QR.

Fix: modify m such that you're always encrypting a QR

Another fix: pick g such that g is always a QR