CS 290G (Fall 2014) – Introduction to Cryptography Oct 23rdd, 2014

Lecture 5: RSA OWFs

Instructor: Rachel Lin *Scribe: Tiawna Cayton*

1 Recap

Last class we discussed a collection of one-way functions (OWFs), or a collection of functions in which it is hard to invert a randomly chosen function from the collection. We talked about one collection of OWFs in particular, called the RSA collection. The RSA collection is a set of functions that includes all f of the form

$$f_{N,e}(x) = x^e mod N$$

with $(N, e) \in I$.

$$I = \{(N, e) : N = product\ of\ 2\ primes, e \in \mathbb{Z}^*_{\phi(N)}\}$$

$$\forall (N, e) \in I, \exists d, d = e^{-1}\ in\ \mathbb{Z}_\phi(N)^*$$

In this case, d is a nice feature because it can be used as a trapdoor, creating a trapdoor function.
We also showed that

$$\forall y \in \mathbb{Z}^*_{\phi(N)}\ \exists x\ s.t.\ f_{N,e}(x) = y$$

Given (N,e,d,y),

$$x = (y)^d \qquad\qquad x^e mod N = ... = y$$

Thus, the RSA collection is a collection of "trapdoor permutations" which will be very useful for public key encryption.

2 Computational Security

Today, we are going to come back to symmetric key encryption, or secret key encryption.

Remember our first definition of security was perfect security which can be acquired by the one-time pad (OTP). An OTP XORs a randomly generated key with a message in order to generate the cipher text. This method is very impractical because $|k| = |m|$. This property is inherit, as we showed in Theorem 1: Any perfectly secure encryption must have $|k| \geq |m|$. Theorem 1 relies on the fact that an adversary is computationally unbounded.

It is a natural idea to expand the original key by some method in order to achieve security without storing a long key. Using a pseudo-random generator(PRG):

$$k \leftarrow \{0,1\}^n \cdots \rightarrow k' \leftarrow \{0,1\}^{l(n)}, l(n) = n^3$$

such that k' is "random-looking" for all computationally bounded adversaries.

Because it contradicts Theorem 1, it is a fact that k' cannot be truly random.

Suppose we have a short seed,

$$D = k \leftarrow \{0,1\}^n, k' \leftarrow G(k)$$

Using a PRG function G on k to map n bits to l(n) bits. PRG is deterministic (does not add entropy/randomness). We want to say that D "looks" random. Since there are many statistical properties that a bit-stream should meet in order to look random, we must define this property.

3 Computational Indistinguishability

When can we say that two distributions look the same? $D_1 \sim D_2$

A distribution looks random when it is similar to a uniformly distributed bit string, that is $D_1 \sim U_{l(n)}$.

When a computer is given two distributions, with unlimited time and resources, differences will be found. However, the difficulty of finding differences grows as the parameters grow. As an example, a computer seeing limited samples and attempting to tell two distributions apart is similar to a human looking at different graphs for a short time. We will use an asymptotic approach - sequences of x and y, rather than just x and y. The outputs should look more similar as parameters grow. We will look at sequences of distributions $\{D_{1n}\}_{n\in N} \sim \{D_{2n}\}_{n\in N}$

Definition Ensemble of Distributions
A sequence $\{x_n\}_{n\in\mathbb{N}}$ is called an ensemble if $\forall n \in \mathbb{N}$, x_n is a distribution over $\{0,1\}^{l(n)}$ for some polynomial l.

Definition Computational Indistinguishability
Two ensembles $\{X_n\}, \{Y_n\}$ are computationally indistinguishable if $\forall$ non-uniform PPT distinguisher D $\exists$ negligible $\epsilon(\cdot)$ such that

$$Adv_D = |Pr[t \leftarrow X_n : D(t) = 1] - Pr[t \leftarrow Y_n : D(t) = 1]| \leq \epsilon(n)$$

When two ensembles are indistinguishable, we write $\{X_n\} \approx \{Y_n\}$

Definition Pseudo Random Distribution
We say an ensemble of distributions $\{D_n\}$ over $\{0,1\}^{l(n)}$ is pseudo-random if $\{D_n\}_{n\in\mathbb{N}} \approx \{U_{l(n)}\}_{n\in\mathbb{N}}\}$ In particular, $\{D_n\}$ is pseudo-random if it meets our statistical properties for pseudo-randomness.

1. Closure under efficient operation
If we have two computationally indistinguishable ensembles, processing the samples will result in a computationally indistinguishable ensemble.
e.g.

$$\begin{aligned} \{X_n\} &\approx \{Y_n\} \\ \{t \leftarrow X_n : t+1\}_{n\in\mathbb{N}} \\ \{t \leftarrow Y_n : t+1\}_{n\in\mathbb{N}} \end{aligned}$$

These distributions are computationally indistinguishable, or it can be shown that the original distributions were not computationally distinguishable.

Lemma: If $\{X_n\} \approx \{Y_n\}$, then $\forall$ non-uniform PPT M, $\{M(X_n)\} \approx \{M(Y_n)\}$

Proof: By contraposition, Suppose
$\exists D, Adv_D \geq \frac{1}{p(n)}$ for some polynomial p $\Rightarrow \exists D', Adv_{D'} \geq \frac{1}{q(n)}$ for polynomial q

$$\begin{aligned} Adv_{D'} &= Pr[t \leftarrow X_n : D'(t) = 1] - Pr[t \leftarrow Y_n : D'(t) = 1] \\ &= Pr[t \leftarrow X_n : D(M(t)) = 1] - Pr[t \leftarrow Y_n : D(M(t)) = 1] \\ &= Adv_D \leq \frac{1}{p(n)} \end{aligned}$$

showing that $\{M(X_n)\}$ *and* $\{M(Y_n)\}$ are not indistinguishable. Given D we can construct D' to distinguish the original distributions which is a contradiction.

2. Transitivity

$$\{X_n\} \approx \{Y_n\}, \{Y_n\} \approx \{Z_n\} \Rightarrow \{X_n\} \approx \{Z_n\}$$

Lemma: Hybrid Lemma
Let $\{X_{1n}\} \cdots \{X_{mn}\}$ satisfy $\forall i \in [m-1], \{X_{in}\} \approx \{X_{(i+1)n}\}$
Then $\{X_{in}\} \approx \{X_{mn}\}$

We want to show: $\forall D$, $\exists\, \epsilon$ negligible such that

$$|Pr[t \in X_{in} : D(t) = 1] - Pr[t \in X_{mn} : D(t) = 1]| < \epsilon(n)$$

Define $g_{in} = Pr[t \leftarrow X_{in} : D(t)]$

$$|g_{in} - g_{mn}| = |(g_{1n} - g_{mn}| = |(g_{1n} - g_{2n}) + (g_{2n} - g_{3n}) \cdots - g_{mn}| \leq \sum \left|g_{in} - g_{(i+1)n}\right|$$

Since $\{X_{in}\} \approx \{X_{(i+1)n}\}$.

By definition, $\exists \epsilon(\cdot)$ negligible such that $\left|g_{in} - g_{(i+1)n}\right| < \epsilon_i(n)$
$|g_{1n} - g_{mn}| \leq \sum_{i \in [m-1]} \epsilon_i(n)$ which is negligible

Example Suppose we have four sets such that $\{X_1\} \approx \{X_1'\} and \{Y_1\} \approx \{Y_1'\}$.
We want to see if $\{X_1\} \oplus \{Y_1\} \approx \{X_1'\} \oplus \{Y_1'\}$.

$\{D_{1n}\} = \{x \leftarrow X, y \leftarrow Y : x \oplus y\}$
$\{D_{2n}\} = \{x' \leftarrow X', y \leftarrow Y : x' \oplus y'\}$
$\{D_{1n}\} = \{x' \leftarrow X', y' \leftarrow Y' : x' \oplus y'\}$
Then
$\{D_{1n}\} \approx \{D_{2n}\}$ and $\{D_{2n}\} \approx \{D_{3n}\}$
So by the hybrid lemma $\{D_{1n}\} \approx \{D_{3n}\}$

4 Computation Version of OTP

Definition Pseudo-Random Generator G
Function G which maps $\{0,1\}* \rightarrow \{0,1\}*$ is a PRG if:

1. Easy to evaluate in PPT
2. Expansion $\forall x, |G(x)| > |x|$
3. Output is pseudo random
   $\{x \leftarrow \{0,1\}^n : G(x)\} \approx \{U_{|G(x)|}\}$

Assume we have a PRG G that maps x to y of length $l(|x|)$.
$Gen(1^n) : k \leftarrow \{0,1\}^n$, where k is the seed
$Enc(G(k), m)$
$m \in \{0,1\}^{l(n)}$
$k \oplus m = c$

$\forall m_0, m_1, \{k \leftarrow \{0,1\}^n, c_0 \leftarrow Enc(k, m_0) : c_0\} \approx \{k \leftarrow \{0,1\}^n, c_1 \leftarrow Enc(k, m_1) : c_1\}$
This provides strong (computationally indistinguishable) security.

5 Summary

In this lecture, we defined an idea of computational security as computational indistinguishability. That is, using a random key, we expanded the key to be a pseudo random

key of greater length. We used an ensemble of distributions to provide greater parameters and increase the amount of computational power needed to find differences in distributions. We defined pseudo random distributions to have closure and trasitivity.
Ultimately, we showed that a pseudo random generator could give us strong computational security. This gives us a step toward a pseudo random function, which will be used for symmetric (secret) key encryption.