CS 6160 Cryptology Lecture 3: One-Way Functions

Maria Francis

September 8, 2020

Introduction to One-Way Functions

- One-way Functions (OWF) are easy to compute, difficult to invert.
- One-way Permutation (OWP) permutes elements as well.
- A Trapdoor Permutation (TDP) is an OWP that has a secret, once you know the secret you can invert the OWP easily.
- Existence of OWFs are not known! Their existence (trivially!) implies $P = NP$.
- In fact, we do not yet know how to prove "If $P \neq NP$ then OWFs exist"!

Motivation

> The existence of one-way functions (owf) is arguably the most important problem in computer theory.
> –*Leonard Levin*

- Fundamental tool in cryptography – used in authentication and crypto applications.
- The idea is to make it cheap for legitimate users and prohibitively expensive for malicious users.
- We will also see that the existence of one-way function implies the existence of Pseudorandom generators, Pseudorandom functions, Bit commitment schemes, Digital signatures, MACs, Symmetric key encryption schemes secure against CCA2.

OWFs

A function $f : \{0,1\}^* \to \{0,1\}^*$ is one way if it satisfies:

1. $\exists$ poly-time algorithm which computes $f(x)$ correctly $\forall\ x$. ($\Rightarrow$ easy to compute)
2. $\forall$ PPT algorithm A,

   $$Pr[f(z) = y : x \xleftarrow{R} \{0,1\}^k; y = f(x); z \leftarrow A(y, 1^k)] \leq \mathrm{negl}(k),$$

   where $\xleftarrow{R}$ means randomly chosen and k is the security parameter.

Note :k has a unary representation – base-1 system where N is rep. by repeating 1 N times.

- x is randomly chosen from $k$bit numbers,
- That is, in polynomial in k time Eve has negl. probability of finding ANY preimage of $f(x)$.

OWPs

A function f is a OWP if it is a OWF and a permutation, i.e. f

1. satisfies all requirements of a OWF and
2. is a permutation, i.e. it is a one-one onto function from a set to itself

A function f is a TDP if it is a OWP and *given certain information* f can be inverted in PPT.

An attempt at formally defining TDP

A function f is a TDP if

1. it satisfies the requirements for OWP and
2. there exists a poly-time algorithm I, some constant c and a string t_k s.t. for all large enough k, $|t_k|$ is at most $O(k^c)$ and for any $x \in \{0,1\}^k$, $I(f(x), t_k) = z$ where $f(x) = f(z)$.

More useful definition of TDP and possible candidates for TDP will be discussed when we study RSA.

Candidate for OWF:Integer Multiplication

- We focus on number theoretic candidates for now.
- For OWPs and TDPs, all examples that we know are number theoretic. For OWFs, there are other candidates.
- Function $f : \mathbb{P}_k * \mathbb{P}_k \rightarrow X$, where $\mathbb{P}_k$ is the set of k-bit primes and X is the set of $2 * k$-bit numbers.
- Not a permutation!
- Consider $f(p, q) = n$ where $n = p * q$.
- Only seeing n, there is no known poly-time algorithm A s.t. $A(n)$ outputs $p^{'}, q^{'}$ such that $p^{'} * q^{'} = n$.
- Note that by unique factorization theorem, either $p^{'} = p, q^{'} = q$ or $p^{'} = q, q^{'} = p$.

Candidate for OWF:Integer Multiplication

Integer multiplication is a OWF:
time tested conjecture
and the most common assumption in crypto!

- Why not test all numbers from 2 to $\sqrt{n}$ - Here n is a $2k$ bit length number, the algo runs in $O(2^k)$.
- How do we make use of one way functions?
  If factoring is hard then X is PROVABLY secure.
- This means to break X we need to be able to factor large integers easily which seems *highly unlikely*.

Candidate for OWP:Modular Exponentiation

- The modular exponentiation function $f_{p,g}(x) = g^x \mathrm{mod} p$, where p is a prime, is a OWP.
- We have, $\mathbb{Z}_n = \{0, \ldots, n-1\}$.
- Ex: Show that

$$\mathbb{Z}_n{}^* = \{x : x \in \mathbb{Z}_n \text{ and } \gcd(x, n) = 1\}$$

  is a multiplicative group.
- $\mathbb{Z}_n{}^*$ is the set of elements in $\mathbb{Z}_n$ that are relatively prime to n.
- $0 \notin \mathbb{Z}_n{}^*$.
- If $n = p$, a prime, then $\mathbb{Z}_p{}^* = \mathbb{Z}_p \setminus \{0\}$.

More number theory facts

- Fermat's Little Theorem: For any prime p and $x \in \mathbb{Z}_p{}^*$,

$$x^{p-1} \equiv 1 \bmod p.$$

- Note that for any $a \in \mathbb{Z}_p{}^*$, the smallest x for which $a^x \equiv 1 \bmod p$ is the order of a in $\mathbb{Z}_p{}^*$
- There may be elements in $\mathbb{Z}_p{}^*$ with order less than $p-1$.
- But we have:For prime p, $\mathbb{Z}_p{}^*$ has at least ONE element g with order $p-1$.
- Such elements are called generators. Why?
- $\{g^1, g^2, \ldots, g^{p-1}\} = \mathbb{Z}_p{}^*$ - Verify!

Modular Exponentiation as a OWP

- Back to $f_{p,g}(x) = g^x \mathrm{mod} p$, where p is a k-bit prime, g a generator of $\mathbb{Z}_p{}^*$ and $x \in (\mathbb{Z}_p \setminus \{0\} = \mathbb{Z}_p{}^*)$.
- g is a generator so we have a permutation from $\mathbb{Z}_p{}^*$ to $\mathbb{Z}_p{}^*$.
- Claim: Computing $y = g^x \mathrm{mod} p$ can be done in poly-time $(O(k^3))$ by repeated squaring!
- The inverse problem is: given $g^x = y \bmod p$, g, p, y find x. Discrete Log Problem, a hard problem
- Thus $f_{p,g}$ is a OWP. No trapdoor known to make inverting easy, so not a TDP

Overview of RSA as TDP

- RSA function $f_{n,e}(x) = x^e \bmod n$, n is the product of two primes p, q, $x \in \mathbb{Z}_n{}^*$, $e \in \mathbb{Z}_{\varphi(n)}{}^*$.
- What is $\varphi(n)$? Euler Totient/phi Function: It is the number of positive integers less than n and relatively prime to n, i.e. $\gcd$ w.r.t. n is 1
- That implies $|\mathbb{Z}_n{}^*| = \varphi(n)$!
- For a prime p, $\varphi(p) = p - 1$ and $n = pq$ implies $\varphi(n) = (p-1)*(q-1) = n - (p+q-1)$ (only for p, q primes!)

Theorem (Euler's Theorem)

For any positive integer n and any $a \in \mathbb{Z}_n{}^*$, $a^{\varphi(n)} \equiv 1 \bmod n$.

A general version of Fermat's!

RSA function as TDP

- For RSA function f, e is always chosen to be in ${\mathbb{Z}_{\varphi(n)}}^*$
  $\Rightarrow \gcd(e, \varphi(n)) = 1$ and e has an inverse mod $\varphi(n)$!
- Ex: Show that there exist an inverse d for $e \mod m$, $ed \equiv 1 \mod m$ iff $\gcd(e, m) = 1$
- Consider $c = f_{n,e}(x) = x^e \mod n$, how to get back x?

$$\begin{aligned} c^d = (x^e)^d \mod n &= x^{ed} \mod n \\ &= x^{1+l(\varphi(n))} \mod n, l \in \mathbb{N} \\ &= x^1 \cdot (x^{\varphi(n)} \mod n)^l \mod n \\ &= x \mod n \quad \text{(by Euler's theorem)}. \end{aligned}$$

RSA function as TDP

- $x \in \mathbb{Z}_n{}^*$ and $\mathbb{Z}_n{}^*$ is a group, so any power of x is in the group, $\Rightarrow f(x) = x^e \mod n$ is also in $\mathbb{Z}_n{}^*$, f is a permutation in $\mathbb{Z}_n{}^*$.
- Public values: n, e and the method to obtain $f_{n,e}(y)$ for some y, c is known.
- Private: x, p, q, d
- Here base is secret and exponent is public, in modular exponentiation exponent is secret, base is public!
- How to invert?
  - If you know how to factor n, then $\varphi(n) = (p-1)(q-1)$ is known and then finding d is easy! No PPT algo for that.
  - Other methods that directly try to compute $\varphi(n)$ or d are just as hard.
- Knowing the factoring is the secret information making f a TDP.