CS 290G (Fall 2014) – Introduction to Cryptography Oct 21st, 2014

Lecture 5: RSA OWFs

Instructor: Rachel Lin *Scribe: Harichandan Pulagam*

1 Recap

In the last class, we continued the discussion of One-Way Functions (OWFs). We started off by proving that hardness of factoring implies the existence of a Strong OWF, i.e.,

$$\text{Hardness of factoring} \Rightarrow \text{Strong OWF}$$

And then we went on to prove **Hardness Amplication**, i.e.,

$$\text{Weak OWF} \Rightarrow \text{Strong OWF}$$

Corollary Hardness of factoring $\Rightarrow$ Strong OWF

2 Collection of OWFs

Earlier definitions of Strong OWFs and Weak OWFs are elegant and concise. The Strong OWFs have a srong security guarantee, while the Weak OWFs are easy to instantiate from mathematical problems for which hard instances come from special domains and ranges, like factoring. Such a definition is very useful for complexity-theoretic crypto. However, it tends not to be as appropriate for the kinds of hard functions that we use in real-life crypto. In this class, we look at combining these features of Strong OWFs and Weak OWFs, to provide for a more flexible definition, i.e.,

1. Allow arbitrary domains and ranges
2. Consider not a single function, but a collection of functions

Definition 1 *Collection of OWFs is a family $F = \{f_i\colon D_i \to R_i\}$ for $i \in I$ such that the following conditions hold:*

1. *Easy to sample a function from F:*
   $\exists$ PPT Gen: Gen(1_n) $\to$ i $\in$ I
2. *Easy to sample from the domain:*
   $\exists$ PPT Sam: $\forall$ i $\in$ I, Sam(i)$\to$a, where a is a random element in D_i
3. *Easy to evaluate:*
   $\exists$ PPT M, $\forall$i, $\forall$x $\in D_i$ such that M(i,x) = f_i(x)

4. *Hard to invert:* $\forall$ nuPPT A, $\exists$ negligible ε(n) such that:
   P_r[i $\leftarrow$ Gen(1_n), x $\leftarrow$ D_i, y=f_i(x), x' $\leftarrow$ A(i,y): f_i(x')=y] $\leqslant$ ε(n)

Observation Any single OWF f: D$\rightarrow$R is also a collection of OWFs.

$$\text{F} = \{f_o = f\text{: D}\rightarrow\text{R}\}_{I=\{0\}}$$

Examples

1. Hardness of factoring $\Rightarrow$ Collection of OWFs

$$\text{F} = \{f_n(\text{x,y}) = \text{x.y for x,y} \leftarrow \textstyle\prod_n \}_{n\in N}\text{, Domain, } D_n = \prod_n^2$$
(Domain looks complex)

2. Hardness of factoring $\Rightarrow$ Collection of OWFs

$$\text{F} = \{g_n(\vec{x},\vec{y}) = x_1y_1, ..., x_my_m \text{ for } x_iy_i \leftarrow \{0,1\}^n\text{, m=}4n^3\}_{n\in N}$$
(Better domain but g_n is inefficient)

3. **RSA One-Way Permutations (OWPs)**
   - Nicer domain compared to 1
   - More efficient compared to 2
   - Widely used in practice

3 Recap of Basic Number Theory

Modular Arithmetic

Given integers a and N.

Theorem 1 $\forall$ *a,N* $\in \mathbb{Z}$
$\exists$ *unique k, 0*$\leqslant$*y*$<$*N such that: a = kN + y*
y = a mod N, k = $\lfloor \frac{a}{y} \rfloor$

Modular Addition/Multiplication

(a + b) mod N = ((a mod N) + (b mod N)) mod N
(a * b) mod N = ((a mod N) * (b mod N)) mod N

Groups

G is a set of elements with an operation $\oplus$: G $\times$ G $\rightarrow$ G. Then :

1. **Closure**: $\forall$ a,b $\in$ G, a$\oplus$b $\in$ G
2. **Identity**: i $\in$ G, $\forall$a $\in$ G, a$\oplus$i = i$\oplus$a = a
3. **Associativity**: (a$\oplus$b)$\oplus$c = a$\oplus$(b$\oplus$c)
4. **Inverse**: $\forall$ a $\in$ G, $\exists a^{-1}$=b $\in$ G, such that:
   a$\oplus$b = b$\oplus$a = i

Example 1: ($\mathbb{Z}$,+) is a group because:

1. **Closure:** Addition of two integers will always be an integer.
2. **Identity**: Identity = 0. Adding 0 to any integer a will give the same integer a.
3. **Associativity**: (a$\oplus$b)$\oplus$c = a$\oplus$(b$\oplus$c)
4. **Inverse**: $\forall a \in \mathbb{Z}$, $\exists$ -$a \in \mathbb{Z}$, such that:
   a+(-a) = i = 0

Example 2: ($\mathbb{Z}$,$\times$) is not a group because:
The inverse condition will be violated for integers.
Inverse: $\forall a \in \mathbb{Z}$, $\nexists \frac{1}{a} \in \mathbb{Z}$.

Additive Modular Groups

$\mathbb{Z}_N$ = 0,1,...,N-1
Operation: + mod N
Identity: i=0
Inverse: $\forall a \in \mathbb{Z}_N$, $\exists b \in \mathbb{Z}_N$, such that:
$(a + b)$ mod N = 0, where $b = a^{-1} = N - a$

Multiplicative Modular Groups

(Z_N^*, * mod N), where Z_N^* = {a | 0$\leqslant$a<N, gcd(a, N) = 1}

Theorem 2 *$\forall$a,N, a has an inverse under ($*$ mod N) $\Leftrightarrow$ gcd(a,N) = 1*

Proof. ($\Rightarrow$ direction)
$\exists$b such that b.a mod N = 1
$\exists$k such that b.a = kN + 1

To prove: gcd(a,N) = 1

Proof by Contraposition

Let gcd(a,N) = c $\neq$ 1
$\Rightarrow$ a=k_1c, N=k_2c
$\Rightarrow$ b.a + kN = Kc (some multiple of c) $\Rightarrow$ b.a + kN $\neq$ 1, which violates our initial assumption. Thus, gcd(a,N) = 1

($\Leftarrow$ direction)
if gcd(a,N)=1 $\Rightarrow$ $\exists$b,k such that b.a + kN = 1 (multiples of gcd(a,N))

Theorem 3 ***<u>Euclid Algorithm</u>***

$\forall$a,b Euclid(a,b) $\rightarrow$ x,y such that:
x.a + y.B = gcd(a,b)

Proof. Given: a>b>0.

Euclid(a,b):
if (base case)
a mod b = 0 *//xa+yb = b, i.e., gcd(a,b)=b*
return (x = 0, y = 1)
else if (a mod b $\neq$ 0) *//gcd(a,b) = gcd(a mod b, b)*
Euclid(b,a mod b) $\rightarrow x, y$
*// x.b + y.(a mod b) = gcd(a,b) = x.b + y.(a − $\lfloor \frac{a}{b} \rfloor$) * b)*
return (y, x−$\lfloor \frac{a}{b} \rfloor$y)

Efficiency Euclid is a polynomial time algorithm meaning, it runs in time:

$$\text{poly}(\log a, \log b) = \text{poly}(\log a)$$

Theorem 4 *($\mathbb{Z}_N^*$, * mod N) is a group*

Proof.
a. Closure: a,b $\in \mathbb{Z}_N^* \rightarrow$ ab mod N
b. Identity, i=1
c. Inverse a^{-1}=b such that ba + kN = 1

Examples of Multiplicative Modular Groups
$\mathbb{Z}_7^*$ = {1,2,3,4,5,6}
$\mathbb{Z}_{15}^*$ = {x: x<15 and x not a multiple of 3 and 5}

$\mathbb{Z}_N^*$, N = p.q where p and q are primes.
Size of $\mathbb{Z}_N^* = \phi(N) = |\mathbb{Z}_N^*|$ (**Euler's Totient Function**)
$\phi(p)$ = p-1 for p=prime
$\phi(N)$ for N = p.q = (p-1)(q-1)

Exponentiation in multiplicative group: * mod N
For e $\geqslant$ 0, a $\in Z_N^*$ a^e mod N = (a*a*..*a) mod N

Theorem 5 *Euler's Theorem*
$\forall a \in \mathbb{Z}_N^*$, $a^{\phi(N)}$ mod N = 1.

Corollary $\forall$e, a^e mod N = $a^{emod\phi(N)}$ mod N = $a^{e-k\phi(N)}$ mod N = $a^e.a^{k\phi(N)}$ mod N
$\therefore$ $a^{\phi(N)}$ mod N = 1

Example $2^{6^{1241}}$ mod 21 = $2^{6^{1241 mod \phi(21)}}$ mod 21
ϕ(21) = (3-1)(7-1) = 12
6^{1241} mod 12 = 6.6^{1240} = $6.(36)^{620}$ mod 12 = 0
$\therefore$ $2^{6^{1241}}$ mod 21 = 2^0 mod 21 = 1

4 RSA Collection of OWFs

An RSA Collection of OWFs can be informally defined as: (N,e) : $f_{N,e}$(x) = x^e mod N.

Hardness of RSA Given a random element y $\in \mathbb{Z}_N^*$, it is hard to find x^e for randomly chosen special N and special e.

special N $\Rightarrow$ Product of primes p and q
special e $\Rightarrow$ {e $\in \mathbb{Z}_{\phi(N)}$ and gcd(e,$\phi(N)$) = 1}, where $\phi(N) = |\mathbb{Z}_N^*|$ = (p-1)(q-1)

Assumption for RSA
$\forall$ nuPPT A, $\exists$ negligible ε(n) such that:
P_r [p,q $\leftarrow \prod_n$, N=p.q, e $\leftarrow \mathbb{Z}_{\phi(N)}^*$, y $\leftarrow \mathbb{Z}_{\phi(N)}^*$: $x' \leftarrow$ A(N,e,y): $(x')^e$ mod N = y] $\leqslant \varepsilon(n)$

This assumption motivates the following definition for RSA Collection:

Definition 2 ***RSA Collection***
$\mathbb{F}_{RSA} = \{f_{N,e} = x^e \ mod \ N\}_{N,e \in I}$
where $I = \{(N,e)$*:* $N=p,q \in \prod_n$, $e \in \mathbb{Z}_{\phi(N)}^*\}$
Domain, $D_{N,e} = \{x \mid x \in Z_N^*\}$
Range, $R_{N,e} = Z_N^*$

Theorem 6 *RSA collection is a collection of OWF.*

1. Easy to sample a function:
   Gen(1^n): p,q $\leftarrow \prod_n$, N = p.q, ϕ(N) = (p-1)(q-1)
   e $\leftarrow \mathbb{Z}_{\phi(N)}^*$

2. Easy to sample from domain of $f_{N,e}$:
   Domain = $\mathbb{Z}_N^*$

3. Easy to compute: $f_{N,e} = x^e$ mod N

4. Hard to invert:
   $\forall$A, $\exists$ ε(n) such that:
   P_r[(N,e) $\leftarrow$ Gen(1^n), x $\leftarrow$ $\mathbb{Z}_N^*$, y = $f_{N,e}$(x) = x^e mod N,
   $x' \leftarrow$ A((N,e),y): $(x')^e$ mod N = y] $\leqslant$ ε(n)
   The distribution given by this function and the one by RSA assumption are the same.

Theorem 7 *$\forall f_{N,e} \in \mathbb{F}_{RSA}$, $f_{N,e}$ is a permutation.*

Proof. $\forall y \in \mathbb{Z}_N^*$, $\exists$ x, $f_{N,e}(x) = y$, i.e., x^e mod N = y
Given y, demonstrate the existence of x
e $\in \mathbb{Z}_N^*$
$\Rightarrow$ e is co-prime with ϕ(N)
$\Rightarrow$ $\exists$d = e^{-1} under $*$mod ϕ(N)

Let's consider $x = y^d$ mod N
x^e mod N = $y \Rightarrow (y^d)^e$ mod N = $y^{de \ mod \ \phi(N)}$ mod N $\because$ d = e^{-1}, $y' = y$