1 Collection of OWF

A Collection of OWF is a family of functions

\[ F = \{ f_i : \mathcal{X} \rightarrow \mathcal{Y} \}_{i \in I} \]

if

- \( \exists \text{PPTGen}(1^n) \rightarrow i \in I \)
- \( \exists \text{PPTSample} \rightarrow \mathcal{X}, \text{given} i \)
- \( \exists \text{PPTCompute} \rightarrow f_i(x), \text{given} i, x \)

\[ \forall \text{nuPPT A } \exists \varepsilon \text{ st } \Pr[i \leftarrow \text{Gen}(1^n), x \leftarrow \mathcal{X} : A(1^n, i, f_i(x)) \in f_i^{-1}(f_i(x))] \leq \varepsilon(n) \]

2

\( F_{\text{multi}} \)

\( f_{\text{multi}} \) family:

- let \( l = N, \mathcal{D} = \{ p, q \} \text{st } \lvert p \rvert = \lvert q \rvert = i, \text{pandqprime} \)
- \( f_i.p.q = pq \)
- \( \text{Gen}(1^n) \rightarrow n \)
3 Fact

Fact: \( \exists \) a collection of OWF iff \( \exists \) OWF

(IF):
\[
\text{Gen}(1^n) = n \\
\mathcal{D}_i = \{0, 1\}^i \\
f_i(x) = f(x)
\]

(only if):
\[
f(r_1, r_2) = f_i(x), \text{ where,} \\
\text{Gen}_{r_1}(1^n) \rightarrow i \\
\text{Sample}_{r_2}(i) \rightarrow x
\]

4 EXP

\[
\text{Gen}(1^n) \rightarrow p, g \text{ where} \\
p \text{ is a random } n\text{-bit prime} \\
g \text{ is a generator for } \mathbb{Z}_p^* \\
I = \{p, g : p \text{ is prime, } g \text{ is a generator for } \mathbb{Z}_p^*\} \\
f_{p,g}(x) = g^x \mod p \\
\mathcal{D}_{p,g} = \mathbb{Z}_p^*
\]

Discrete log assumption:
\[
\{f_{p,g}\}_{(p,q) \in I} \text{ is a collection of OWF}
\]

\(g^x\) can be calculated quickly by repeated squaring

5 RSA collection

\[
I = \{N = (pq), p \text{ and } q \text{ are prime, } |p| = |q|\} \\
\text{Gen}(1^n) \rightarrow (N, e), \text{ where} \\
N = pq, p \text{ and } q \text{ are random } n\text{-bit primes} \\
e \text{ is a random element in } \mathbb{Z}_{\phi(N)}^* \\
f_{N,e}(x) = x^e \mod N
\]

RSA assumption: this is a collection of OWFs
6 Hard-core bits

A predicate $b : \{0, 1\}^* \rightarrow 0, 1^*$ is a hard-core bit for $f()$ if

$b$ is PPT computable, and

$$\forall \nu \text{PPT } A \exists \varepsilon \text{ st } \forall n \in N, \Pr[A(1^n, f(x)) = b(x)] < \frac{1}{2} + \varepsilon(n)$$

In other words, it is easy to calculate $b(x)$ from $x$, but hard to calculate $b(x)$ from $f(x)$.

7 Examples of hard-core bits

Let

$$half_m(x) = 0 \text{ if } 0 \leq x \leq \frac{m}{2}, 1 \text{ otherwise}$$

For RSA,

$half_m(x)$ is hard-core for $f_{N,e}$

For EXP,

$half_{p-1}(x)$ is hard-core for $f_{p,q}$

8 To prove something is hard-core

Prove that, given some $A$ that guesses $b(x)$

with probability greater than

$$\frac{1}{2} + \frac{1}{poly}, \text{ given } f(x),$$

It is possible to write a $B$ that recovers $x$ given $f(x)$