CS 7810 Graduate Cryptography October 17, 2017

Lecture 10: Weak One-Way Functions and Hardness Amplification

Lecturer: Daniel Wichs *Scribe: Ariel Hamlin*

1 Topic Covered

- Weak OWF
- Hardness Amplification
- Universal OWFs
- Hash Functions

2 Weak One-Way Functions

We have previously explored the notion of one-way functions (OWF) and how they relate to other symmetric key primitives. Here we introduce an even weaker notion of OWFs and how they related to OWFs. A brief note on terminology, we refer to the OWF introduced last week as *strong* OWF as opposed to *weak* OWF.

Definition 1 [Weak One-Way Functions]
We say a function f is a *weak one-way function* if the following holds:

- f is computable in time $\mathsf{poly}(n)$
- For all PPT adversaries $\mathcal{A}$, the exists a polynomial $q(\cdot)$ such that for any $n \in \mathbb{Z}$

$$\Pr[f(x) = f(x') \; : \; x \leftarrow \{0,1\}^n, y = f(x), x' \leftarrow A(y)] \leq 1 - \frac{1}{q(n)}$$

♢

The main difference between strong OWF and weak OWF depends on the advantage of $\mathcal{A}$, in weak OWF, the adversary only needs to fail to invert f with some non-negligible probability as opposed to strong where it must be a negligible advantage.

As an example, consider a function f defined as $f(a,b) = a \cdot b$ where $a, b > 1$ are random n-bits numbers (if $a = 1$ or $b = 1$ this is trivially invertible) . Note that if a or b is even then it is easy to compute a pre-image $(2, a \cdot b/2)$ of $a \cdot b$. Since the probability of this occurring is 3/4, it is clear that f is not a strong OWF. But it may nevertheless meet our definition for weak OWFs. If a and b are large primes then f is non-trivial to factor (this is the hard problem that many encryption schemes that are used in practice employ). Fortunately, others have shown that primes are actually fairly common. If a and b are chosen at random there is a $\frac{1}{\mathsf{poly}(n)}$ chance that they will both be prime. Thus this is a good candidate for a weak OWF.

3 Hardness Amplification

Given a weak OWF $f$it is possible construct a strong OWF? Consider the following construction:

$$f'(x_1, \cdots, x_m) = f(x_1)||f(x_2)|| \cdots ||f(x_m)$$

where f has input size n and f' has an input size of $n' = n \cdot m$. We think of m as a parameter. Intuitively, to invert f', one has to invert m independent instances of f. Since inverting one instance of f is mildly hard, if we set m large enough than inverting all m instances should be truly hard. For example, assume we knew that no polynomial time attacker can invert a single instance of f with probability better than $1/2$. Then it seems "obvious" that when $m = n$ then no polynomial time attacker could invert f' with probability better than $1/2^n$. Unfortunately, such intuition is often deceptive. The reason is that, even though the instances are chosen independently, the adversary sees all of them and does not have to invert them independently of each other. Still, we will be able to show via a non-trivial proof that when m is chosen large enough then hardness does indeed amplify (although we cannot show that is amplifies all the way to $1/2^n$ as our intuition seemed to indicate – and there are actual counterexamples to this!). We note that there are other examples in cryptography (e.g., interactive arguments) where similar attempts at hardness amplification completely fail and breaking m independent instances is not much harder than breaking one way instance. So although our intuition turns out to be correct in the case of one-way functions, things are much more subtle than one may have initially thought!

Theorem 1 *If f is a weak OWF then there exits a polynomial m such that f' is a strong OWF. In particular this holds for $m = 2nq(n)$.*

Proof: Assume otherwise, that f' is not a OWF, thus there exists a PPT $\mathcal{A}'$ and polynomial p' for infinitely many n such that

$$\Pr\left[f(x'_1) = y_1, \cdots, f(x'_m) = y_m \ : \ \begin{array}{c} \forall i \in [m] : x_i \leftarrow \{0,1\}^n, y_i = f(x_i), \\ (x'_1, \cdots, x'_m) \leftarrow \mathcal{A}'(y_1, \cdots, y_m) \end{array} \right] \geq \frac{1}{p'(n')} \geq \frac{1}{p(n)} \tag{1}$$

for some polynomial $p(n) = p'(n') = p'(nm) = p(2n^2q(n))$.

Define the following adversary that places the output of f in the i-th location, and generates values for the rest of the inputs into $\mathcal{A}'$.

$\mathcal{A}_i(y)$:

Set $y_i = y$.

For all $j \in [m] \setminus \{i\}$, choose $x_j \leftarrow \{0,1\}^n$ and set $y_j = f(x_j)$.

Let $(x'_1, \cdots, x'_m) = \mathcal{A}'(y_1, \cdots, y_m)$.

If $f(x'_i) = y_i$, output x'_i else output $\perp$.

Define the following adversary which uses $\mathcal{A}_i$ as a subroutine.

$\mathcal{A}(y)$:

For each $i \in [m]$, run $\mathcal{A}_i(y)$ for $\ell = 2mn \cdot p(n)$ iterations with independent randomness. Output the first value that's not $\perp$.

We now analyze the success probability of $\mathcal{A}$ in inverting f. Define the set

$$G_i = \left\{ x \ : \ \Pr[\mathcal{A}_i(f(x)) \neq \perp] \geq \frac{1}{2mp(n)} \right\}$$

where the probability is over the randomness of $\mathcal{A}_i$. We will now use two claims to complete the proof; the first claim says that for some $i \in [m]$, the set G_i contains a large fraction of all x's. The second claim then says that for this i, the probability of ℓ iterations of $\mathcal{A}_i(f(x))$ all outputting $\perp$ is very small.

Claim 1 *There exists some* $i \in [m]$ *such that* $\Pr_{x \leftarrow \{0,1\}^n}[x \in G_i] \geq \left(1 - \frac{1}{2q(n)}\right)$.

Proof: Assume otherwise that for all i we have $\Pr_{x \leftarrow \{0,1\}^n}[x \in G_i] < \left(1 - \frac{1}{2q(n)}\right)$.

Define S' as the event that $\mathcal{A}'$ successfully inverts f' as defined by the experiment in equation (1).

Then

$$\Pr[S'] = \Pr[S' \wedge (\forall i \in [m] \ : \ x_i \in G_i)] + \Pr[S' \wedge (\exists i \in [m] \ : \ x_i \notin G_i)]$$

We can bound:

$$\begin{aligned} \Pr[S' \wedge (\forall i \in [m] \ : \ x_i \in G_i)] &\leq \prod_{i \in [m]} \Pr[x_i \in G_i] \\ &< \left(1 - \frac{1}{2q(n)}\right)^{m=2nq(n)} \\ &< e^{-n} \end{aligned}$$

We can also bound:

$$\begin{aligned} \Pr[S' \wedge (\exists i \in [m] \ : \ x_i \notin G_i)] &\leq \sum_{i \in [m]} \Pr[S \wedge x_i \notin G_i] \\ &\leq \sum_{i \in [m]} \Pr[S | x_i \notin G_i] \\ &< \frac{m}{2mp(n)} < \frac{1}{2p(n)} \end{aligned}$$

where the second line follows by the definition of the set G_i; the probability that $\mathcal{A}'$ succeeds to invert $f(x_1), \ldots, f(x_m)$ if we condition on $x_i \notin G_i$ is at most $\frac{1}{2mp(n)}$.

Combining these we get

$$\Pr[S'] < e^{-n} + \frac{1}{2p(n)} < \frac{1}{p(n)}.$$

This is a contradiction, thus Claim 1 holds. □

Claim 2 $\Pr[A(f(x)) \neq \perp \;:\; x \leftarrow \{0,1\}^n] > 1 - \frac{1}{q(n)}$

Proof: Let S be the event that $A(f(x)) \neq \perp$ and let i be the value from Claim 1.

$$\begin{aligned}
\Pr[\neg S] &\leq \Pr[\neg S \wedge x \in G_i] + \Pr[\neg S \wedge x \notin G_i] \\
&\leq \Pr[\neg S \wedge x \in G_i] + \Pr[x \notin G_i] \\
&\leq \Pr[\neg S \wedge x \in G_i] + \frac{1}{2q(n)} \\
&\leq \left(1 - \frac{1}{2mp(n)}\right)^{\ell = 2mn \cdot p(n)} + \frac{1}{2q(n)} \\
&\leq e^{-n} + \frac{1}{2q(n)} < \frac{1}{q(n)}.
\end{aligned}$$

The fourth line follows since, when $x \in G_i$ then the probability of each of the ℓ iterations of $\mathcal{A}_i(f(x))$ outputting $\perp$ is at most $\left(1 - \frac{1}{2mp(n)}\right)$ and the iterations are independent. □

By the above claim, $\mathcal{A}$ succeeds to invert $f(x)$ with probability $> 1 - \frac{1}{q(n)}$ for infinitely many values n which contradicts f being weakly one-way. This proves the theorem. □

4 Universal OWF

We now show how to construct a *universal OWF* f_{univ} which is guaranteed to be one-way if *any* one-way function exists.

$f_{univ}(x)$:

- Parse $x = (<M>, x^*)$ where $<M>$ is a description of a TM M and $|<M>| = log(n)$.
- Run $M(x^*)$ for $O(n^2)$ steps and set z to be the output or $z = 0$ if the computation does not terminate.
- Output $y = (<M>, z)$.

Remark: we will assume that the encoding $< \cdot >$ ignores leading 0's so that the string w and $0^\ell w$ encode the same TM. We need this to ensure that if some TM can be encoded by a string of size a then there exists some encoding of this TM of every size $> a$.

Theorem 2 *If OWFs exists, f_{univ} is a weak OWF. By using hardness amplification we can then convert it to a strong OWF f'_{univ}.*

Proof: Assume g is OWF, we can assume without loss of generality that the run-time of g is $O(n^2)$ (else if the runtime is some n^c, use $g'(x) = g(x')$ where x' is the first $n^{1/c}$ bits of x; we know that g' is one-way if g is as shown on the homework and its run time is $O(n)$). Let $a = |<g>|$ be some constant.

Fix some PPT attacker $\mathcal{A}$. We want to analyze:

$$\Pr\left[f_{univ}(x') = y : \begin{array}{c} x \leftarrow \{0,1\}^n \\ x = (<M>, x^*) \\ y \leftarrow f_{univ}(x) \\ x' \leftarrow \mathcal{A}(y) \end{array} \right]$$

Let S be the event that $\mathcal{A}$ succeeds in the above experiment.

Then

$$\begin{aligned} \Pr[S] &= \Pr[S|M = g]\Pr[M = g] + \Pr[S|M \neq g]\Pr[M \neq g] \\ &= \Pr[S|M = g] + \Pr[M \neq g] \\ &\leq negl(n) + \left(1 - \frac{1}{2^{\log(n)}}\right) \\ &\leq 1 - \frac{1}{2n}. \end{aligned}$$

where the third line holds since $\Pr[S|M = g]$ is the probability that $\mathcal{A}$ inverts g and $\Pr[M = g] > \frac{1}{2^{\log(n)}}$ for all $\log n > a$. $\square$

The idea of a universal one-way function is amazing. It means we don't have to work hard to build secure crypto; we can just be lazy and rely on the above universal one-way function and we can rest assured that our cryptosystems are secure if any secure cryptosystems exist at all! Unfortunately, this approach only works in theory but does not lead to practical cryptosystems that can be used in real life. The problem is that although the universal OWF is *asymptotically secure* the concrete security is terrible! For example, it may be the case that the smallest one-way function g has description size $a = 1000$ bits. In that case, the security of the universal OWF we constructed only starts kicking in when $n > 2^a > 2^{1000}$. Although this is a constant, it is so large that using such large n is completely impossible!

5 Collision Resistant Hash Functions

A hash function H compresses a long input x into a short digest $H(x)$. Although the function is many-to-one, it should be hard to find collisions $x \neq x'$ for which $H(x) = H(x')$. Therefore, if you get a digest $H(x)$ there is only a single corresponding preimage x that anybody can give you that would match this digest (otherwise someone would have found a collision).

DEFINITION 2 A function $H : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^n$ where $\ell(n) > n$ is a *collision resistant hash function (CRHF)* when H is computable in polynomial time and for all PPT $\mathcal{A}$:

$$\Pr[H_s(x) = H_s(x') \wedge x \neq x' \ : \ s \leftarrow \{0,1\}^n, (x, x') \leftarrow \mathcal{A}(1^n, s)] = \mathsf{negl}(n)$$

$\diamondsuit$

Note that the presences of the seed s in the definition. This seed is public and the adversary should not be able to find collisions even if it given the seed. The reason that we need a seed is that, without it, there is always an efficient (non-uniform) adversary $\mathcal{A}$ that simply has a hard coded collision x, x' for which $H(x) = H(x')$ and outputs this collision. The inclusion of the seed s precludes such trivial attacks.