Lecture 11: Universal OWFs

Lecturer: Jack Doerner *Scribe: Amir Moeini*

1 Topics Covered

- Construction of the universal OWF f_{univ}
- Proof that f_{univ} is $2n$-weak if any strong OWF exists

2 Motivation

Cryptography rests on unproven assumptions: we do not *know* one-way functions exist. This causes unrest. The goal today is to show that if *any* strong OWF exists (even non-constructively), then a specific, explicit function f_{univ} is a weak OWF. From any weak OWF, we can build a strong OWF, so this gives us a concrete foothold in cryptography from the bare existence assumption.

3 Definitions

Definition 1 (Strong OWF)**.** *A function $f : \{0,1\}^* \to \{0,1\}^*$ is a strong OWF if:*

1. *f is PPT*
2. *$\forall$ NUPPT $\mathcal{A}$, $\exists$ negligible ε s.t. $\forall n \in \mathbb{N}$,*

$$\Pr\left[f(x') = f(x) \ : \ x \leftarrow \{0,1\}^n, \ x' \leftarrow A(1^n, f(x))\right] \leq \varepsilon(n).$$

Definition 2 (μ-weak OWF)**.** *For a fixed polynomial μ, f is a μ-weak OWF if:*

1. *f is PPT.*
2. *$\forall$ NUPPT $\mathcal{A}$, $\exists\, n_0$ s.t. $\forall n \geq n_0$,*

$$\Pr\left[f(x') = f(x) \ : \ x \leftarrow \{0,1\}^n, \ x' \leftarrow A(1^n, f(x))\right] \leq 1 - \tfrac{1}{\mu(n)}.$$

4 Universal OWF

Theorem 1. *If a strong OWF exists, then f_{univ}[1] is a $2n$-weak OWF.*

[1] Defined below in Construction 1.

Intuition on how to prove this. If the (unknown) strong OWF had a very short description (say length $< \log n$), then a uniformly random description among all $\log n$-bit programs hits it with probability $1/n$ (inverse-polynomial, not negligible). This hints that guessing a short program and running it for a bounded time on the rest of the input might produce a weakly one-way mapping. Two issues remain: (i) we must run the guessed program within a known polynomial time bound, and (ii) formalize the function then show that the probability of inverting it makes it a weak OWF.

First, let's bound the strong OWF runtime. A strong OWF is PPT. To ensure that it runs in some *specific* polynomial amount of time, we can pad the input with enough (but polynomial) unused bits .

Lemma 1 (An $O(m^2)$ OWF)**.** *If a strong OWF f exists, then there exists a strong OWF g with $|g| \approx |f|$ (up to a constant in any reasonable encoding) such that g runs in time $O(m^2)$ on m-bit inputs.*

Proof Sketch. Assume f runs in time n^c for some $c > 2$ on n-bit inputs. Let $m = n^c$. Define

$$g(a\|b) \;=\; a \;\|\; f(b), \quad \text{with} \quad |b| = n,\; |a| = n^c - n,\; |a\|b| = m.$$

- *Runtime:* Copying a is $O(n^c - n)$, evaluating $f(b)$ is $O(n^c)$, and overhead costs (e.g. computing $n = \sqrt[c]{m}$) are $O(m^2)$. Hence g is $O(m^2)$.
- *One-wayness:* An inverter for g yields one for f by embedding the challenge $y = f(b)$ as the rightmost n bits, and sampling a uniform $(n^c - n)$-bit value a.
- *Size:* g's description is just f's plus, a constant-size wrapper. $\square$

We will use the following fact to ensure that our $O(n^2)$ OWF runs to completion within strict n^3 step budget, once n is large.

Fact 1 ($O(n^2)$ eventually below n^3)**.** *If $t(n) \in O(n^2)$, then $\exists n_t$ s.t. $\forall n \geq n_t$, $t(n) \leq n^3$.*

Additionally, we will need to ensure that as we sample machines of increasing description-length, no members are ever eliminated from the set. This is guaranteed if we use a monotone encoding.

Fact 2 (Monotone Machine Encodings)**.** *There exists a monotone encoding of Turing machines. That is, there exists some encoding of turing machines into bit-strings such that:*

$$\forall M \in \{0,1\}^n \;\; \exists M' \in \{0,1\}^{n+1} \;\; s.t. \;\; \forall x \in \{0,1\}^* \;\; M(x) = M'(x)$$

Construction 1 (The Universal OWF $f_{\mathsf{univ}} : \{0,1\}^* \to \{0,1\}^*$)**.**

1. *On input x, parse $x = M\|x'$ where M is the first $\log |x|$ bits interpreted as a TM description, and x' is the remaining suffix.*
2. *Run M on input x' for $|x|^3$ steps.*
3. *If M halts with output y on its tape, output*

$$f_{\mathsf{univ}}(x) \;=\; M \;\|\; y.$$

4. Otherwise, output a fixed failure tag $\perp$*.*

Now we're ready to prove that Construction 1 is a $2n$-weak OWF.

Proof of Theorem 1. Assume there exists a strong OWF g' (unknown and possibly non-explicit). By lemma 1, there is a strong OWF g with $|g| \approx |g'|$ running in time $O(m^2)$. By Fact 1, there is a constant n_g such that g halts within m^3 steps on all $m \geq n_g$. Let M_g be a shortest monotone encoding for g. For each $n \geq |M_g|$, let $M_g^{(n)}$ be the length-n extension of M_g that encodes the same machine g per Fact 2. Note that such an extension always exists when $n \geq |M_g|$.

Claim 1 (Randomly hitting the strong OWF machine)**.** *When* $|x| = n$*, Construction 1 interprets the first* $\lfloor \log n \rfloor$ *bits of* x *as a Turing machine* M*.* $\forall\, n \geq 2^{|M_g|}$ *,*

$$\Pr\big[\underbrace{M = M_g^{(\log n)} \ : \ M \leftarrow \{0,1\}^{\log n}}_{\textsf{Exp Picks g}}\big] \ = \ \frac{1}{2^{\log n}} \ = \ \frac{1}{n}.$$

Claim 2 (Some negligible term clean up)**.** $\forall$ negligible $\varepsilon\ \exists\, n_\varepsilon$ s.t. $\forall\, n \geq n_\varepsilon$*,*

$$\left(1 - \frac{1}{n}\right) + \varepsilon(n - \log n) \ \leq \ 1 - \frac{1}{2n}.$$

The proofs of the above two claims are intuitive. Next, for any NUPPT $\mathcal{A}$, the law of total probability yields

$$\begin{aligned}
&\Pr\Big[\underbrace{f_{\textsf{univ}}(x) = f_{\textsf{univ}}(x') \ : \ x \leftarrow \{0,1\}^n, \ x' \leftarrow \mathcal{A}(1^n, f_{\textsf{univ}}(x))}_{\mathcal{A}\ \textsf{inverts}}\Big] \\
&= \Pr\big[\mathcal{A}\ \textsf{inverts} \mid \textsf{Exp Picks g}\big] \cdot \Pr\big[\textsf{Exp Picks g}\big] \\
&\quad + \Pr\big[\mathcal{A}\ \textsf{inverts} \mid \neg\,\textsf{Exp Picks g}\big] \cdot \Pr\big[\neg\,\textsf{Exp Picks g}\big]
\end{aligned}$$

To find an upper bound, let's just assume that $\mathcal{A}$ inverts any any machine $M \neq M_g^{(\log n)}$ with probability 1. Combining the last equasion with Claim 1, $\forall\, n \geq 2^{|M_g|}$ we have

$$\Pr[\mathcal{A}\ \textsf{inverts}] \leq \Pr[\mathcal{A}\ \textsf{inverts} \mid \textsf{Exp Picks g}] \cdot \frac{1}{n} \ + \ \left(1 - \frac{1}{n}\right).$$

To get an upper bound for $\Pr[\mathcal{A}\ \textsf{inverts} \mid \textsf{Exp Picks g}]$ we should let $n \geq n_g$ to provide enough compute for $f_{\textsf{univ}}$ to halt. It's easy to see that if Exp Picks g and M_g runs to completion, then $f_{\textsf{univ}}$ is a strong OWF.[2] Therefore, there exists some negligible function ε' such that $\forall\, n \geq \max\{2^{|M_g|}, n_g\}$,

$$\begin{aligned}
&\Pr[\mathcal{A}\ \textsf{inverts} \mid \textsf{Exp Picks g}] \\
&\quad = \Pr\left[g(x) = M_g'(x') \wedge M_g^{\log n} = M_g' : x \leftarrow \{0,1\}^{n-\log n}, M_g' \| x' \leftarrow \mathcal{A}\left(1^n, M_g^{\log n} \| g(x)\right)\right] \\
&\quad \leq \Pr\left[g(x) = g(x') : x \leftarrow \{0,1\}^{n-\log n}, M_g' \| x' \leftarrow \mathcal{A}\left(1^n, M_g^{\log n} \| g(x)\right)\right] \\
&\quad \leq \varepsilon'(n - \log n). && \text{By Definition 1}
\end{aligned}$$

[2] There's a reduction that converts inverting $f_{\textsf{univ}}$ to inverting g by simply concatenating M_g to the beginning of the challenge.

Therefore, there exists another negligible function ε such that

$$\begin{aligned}\Pr[\mathcal{A}\ \mathsf{inverts}] &\le \varepsilon'(n-\log n)\cdot\frac{1}{n} \;+\; \left(1-\frac{1}{n}\right)\\ &\le\; \varepsilon(n-\log n) \;+\; \left(1-\frac{1}{n}\right).\end{aligned}$$

Thus by Claim 2, $\exists\, n_\varepsilon$ s.t. $\forall\, n \ge n_0 = \max\{2^{|M_g|}, n_g, n_\varepsilon\}$,

$$\Pr[\mathcal{A}\ \mathsf{inverts}] \;\le\; 1-\frac{1}{2n}.$$

Note that $\mu(n) = 2n$ didn't depend on our choice of $\mathcal{A}$, while n_0 did (through n_ε). Thus this satisfies Definition 2 and f_{univ} is $2n$-weak. □

Why this is not practical. Security holds only when $\log n \ge |M_g|$. If the shortest description of a strong OWF has, say, $|M_g| = 1000$ bits, then the minimum input length where the universal OWF actually becomes hard to invert is $n \ge 2^{1000}$. This is fine asymptotically, but useless in practice. Whether one can design a more efficient universal OWF (i.e. one that is plausibly one-way for parameters that can be used in practice) is open.