Advance Course on Foundation of Cryptography, Lecture 4
Black-Box Impossibilities
The Basics

Handout Mode

Benny Applebaum & Iftach Haitner, Tel Aviv University

Tel Aviv University.

April 27, 2017
Talk plan

- Motivation and definition
- Random permutations are hard to invert
- Impossibility for Basing OWP on OWF
- Impossibility for basing efficient PRG on OWF
Motivating example: Basing Key-Agreement on OWFs

- Key-Agreement protocols (KA) can be based on the existence of TDP, RSA or discrete log assumptions, and ...
- We don’t know how to base KA on the existence of OWFs/OWPs.
- Can we base KA on OWFs/OWPs?
- Proving unconditional negative result seems beyond reach.
  Assume RSA assumption holds.
    \[\Rightarrow\] key-agreement protocols exist.
    \[\Rightarrow\] OWFs imply the existence of key-agreement protocols in a trivial sense.
**Definition 1 (A fully Black-box construction of $B$ from $A$)**

**Black-box construction**: Oracle-aided PPT $C$ such that $C^f$ implements $B$ for any algorithm $f$ implementing $A$.

**Black-box proof of security**: Oracle-aided PPT $R$ such that $R^{f,\text{Brk}}$ breaks $f$, for any algorithms $f$ implementing $A$, and Brk breaking $C^f$.

- Fully-black-box constructions relativize: hold relative to any oracle.
- Most constructions in cryptography are (fully) black-box, e.g., pseudorandom generator from OWF.
- Some “non black-box” techniques apply in restricted settings.
Proving BB impossibility result

Assume $\exists$ fully-BB construction $(C, R)$ of a KA from OWP.
Section 1

Random Permutations are Hard To invert
Some preliminaries

- \( k! := 1 \cdot 2 \cdot 3 \cdots k \)
- \( \binom{k}{a} := \frac{k!}{a! \cdot (k-a)!} \)
- \( k! = \binom{k}{a} \cdot a! \cdot (k-a)! \)
- \( k! \geq \left( \frac{k}{e} \right)^a \) and \( \binom{k}{a} \leq \left( \frac{ek}{a} \right)^a \)

Let \( N = 2^n \).

- How many bits it takes to describe \( S \subseteq \{0, 1\}^n \) of (known) size \( a \)?
  There are \( \binom{N}{a} \) such sets, so it takes \( \log(\binom{N}{a}) \) bits.

Let \( \Pi_n \) be the set of all permutations over \( \{0, 1\}^n \).

- How many bits it takes to describe \( \pi \in \Pi_n \)?
  \( |\Pi_n| = N! \), so it takes \( \log(N!) \) bits.

What is the number of of \( M \)-size oracle-circuits?

Claim 2 (HW 1)

Number of \( M \)-size oracle-circuits mapping \( n \)-bit strings to \( n \)-bit strings, with oracle access to a function \( n \)-bit strings to \( n \)-bit strings, is at most \( 2^{2M+(M+1)n\log(Mn+n)+1} \).
Notation

For circuit (or deterministic algorithm) $D$ and $f : \{0, 1\}^n \mapsto \{0, 1\}^n$, let

- $\text{Inv}_f^D(y) = \begin{cases} 1, & D(y) \in f^{-1}(y), \\ 0, & \text{ow.} \end{cases}$

- $\text{Inv}^D(f) = E_{x \leftarrow \{0,1\}^n} [\text{Inv}_f^D(f(x))]$. 
Random permutations are hard to invert

**Theorem 3 (Gennaro-Tevisan, ’01 )**

For large enough \( n \in \mathbb{N} \) and \( 2^{n/5} \)-query circuit \( D \),

\[
\Pr_{\pi \leftarrow \Pi_n} \left[ \text{Inv}^D(\pi) = \Pr_{y \leftarrow \{0,1\}^n} [D(y) = \pi^{-1}(y)] > 2^{-n/5} \right] \leq 2^{-2^{3n/5}}
\]

Random permutations are exponentially hard for exponential-query circuits.

- Extends to randomized algorithms.
- Constants are somewhat arbitrary and non tight.
- Since \# of \( 2^{n/5} \)-size circuits is bounded by \( 2^{\tilde{O}(2^{n/5})} \), Thm 3 yields that

\[
\Pr_{\pi \leftarrow \Pi_n} \left[ \exists \ 2^{n/5} \text{-size circuit } D \text{ with Inv}^D(\pi) > 2^{-n/5} \right] \leq 2^{-2^{n/2}}
\]

Random permutations are exponentially hard for all exponential-size circuits simultaneously.
Proving GT theorem (Thm 3), lazy evaluation approach

Lazy evaluation approach: for any $2^{n/5}$-size circuit $C$ and $x \in \{0, 1\}^n$

$$\Pr_{\pi \leftarrow \Pi_n} [C(\pi(x)) = x] \leq 2^{-4n/5}$$

It follows that for any $2^{n/5}$-size circuit $C$

$$\Pr_{\pi \leftarrow \Pi_n} \left[ \Pr_{x \leftarrow \{0,1\}^n} [C(\pi(x)) = x] > 2^{-3n/5} \right] < 2^{-n/5}$$

Not strong enough to rule out inversion by all small circuits simultaneously.
Proving GT theorem (Thm 3), warmup

Assume $D$ makes no queries.

For how many $\pi \in \Pi_n$ it holds that $\text{Inv}^D(\pi) = 1$?

Answer: one! (defined by $\pi^{-1}(y) = D(y)$)

Hence,

$$\Pr\left[\pi \leftarrow \Pi_n \left| \text{Inv}^D(\pi) = 1 \right. \right] \leq \frac{1}{N!} \leq \left(\frac{e}{2^n}\right)^{2^n}$$

For how many $\pi \in \Pi_n$ it holds that $\text{Inv}^D(\pi) \geq \varepsilon$?

What if $D$ makes oracle queries?
Proving GT theorem (Thm 3)

Lemma 4 (compression lemma)

For any \( q \)-query circuit \( D \) and \( \varepsilon > 0 \), exist algorithms \( \text{Enc} \) and \( \text{Dec} \) s.t. the following holds: Let \( \pi \in \Pi_n \) be s.t. \( \text{Inv}^D(\pi) > \varepsilon \), then

- \( \text{Dec}(\text{Enc}(\pi)) = \pi \).
- \( |\text{Enc}(\pi)| \leq \log((N - a)!) + 2 \cdot \log \left( \binom{N}{a} \right) \), for some \( a \geq \frac{\varepsilon N}{q+1} \).

\[ \Rightarrow \] # of permutations in \( \Pi_n \) with \( \text{Inv}^D(\pi) > \varepsilon \) is at most \( (N - a)! \cdot \left( \binom{N}{a} \right)^2 \).

- Let \( D \) be a \( q = 2^{n/5} \)-query circuit. By Lemma 4

\[
\Pr_{\pi \leftarrow \Pi_n} \left[ \text{Inv}^D(\pi) > 2^{-n/5} \right] \leq \frac{(N - 2^{3n/5})! \cdot \left( \binom{N}{2^{3n/5}} \right)^2}{N!} = \frac{\left( \binom{N}{2^{3n/5}} \right)}{2^{3n/5}!} \leq 2^{-2^{3n/5}}
\]

proving Thm 3.
Proving Compression Lemma, warm-up

Assume $D$ makes no queries.

- Fix $\pi \in \Pi_n$ with $\text{Inv}^D(\pi) > \varepsilon$, and let $\mathcal{Y}_\pi = \{y \in \{0, 1\}^n : D(y) = \pi^{-1}(y)\}$
  
  Note that $|\mathcal{Y}| \geq \varepsilon N$.

- $\pi$ is determined by $\mathcal{Y}_\pi$ and $\mathcal{V}_\pi = \{(x, \pi(x)) : \pi(x) \notin \mathcal{Y}_\pi\}$.
  
  Proof: $\pi$ is determined by the sets $\mathcal{V}_\pi$ and $\mathcal{V}'_\pi = \{(D(y), y) : y \in \mathcal{Y}_\pi\} = \{(x, \pi(x)) : \pi(x) \in \mathcal{Y}_\pi\}$.

- Given $\mathcal{Y}_\pi$, the set $\mathcal{V}_\pi$ is determined by the set $\mathcal{X}_\pi = \pi^{-1}(\mathcal{Y}_\pi) = \{\pi^{-1}(y) : y \in \mathcal{Y}_\pi\}$, and a permutation $\tau_\mathcal{Y}$ over $\{0, 1\}^n \setminus \mathcal{X}_\pi$.

  Hence, $\pi$ can be described using $\log((N - |\mathcal{Y}|)! + 2 \cdot \log \left(\frac{N}{|\mathcal{Y}|}\right)$ bits.

Define $\text{Enc}(\pi)$ to output the description of $\mathcal{Y}_\pi, \mathcal{X}_\pi$ and $\tau_\pi$.

$\text{Dec}(\mathcal{Y}, \mathcal{X}, \tau)$ is defined accordingly.
Proving Compression Lemma

Let $D$ be a $q$-query circuit, and let $\pi$ be s.t. $\text{Inv}^D_\pi(\pi) > \varepsilon$.

**Construction 5 (Useful set $\mathcal{Y}_\pi \subseteq \{0, 1\}^n$)**

1. Set $\mathcal{Y}_\pi = \emptyset$ and $\mathcal{I}_\pi = \{y \in \{0, 1\}^n : D^\pi(y) = \pi^{-1}(y)\}$.
2. While $\mathcal{I} \neq \emptyset$, let $y$ be the smallest lexicographic element in $\mathcal{I}_\pi$.
   - (a) Add $y$ to $\mathcal{Y}_\pi$.
   - (b) Remove $y$ and all answers to $\pi$-queries $D^\pi(y)$ makes, from $\mathcal{I}_\pi$.

**Algorithm 6 (Enc($\pi$))**

Output (description of) $\mathcal{Y}_\pi$ and $\mathcal{V}_\pi = \{(x, \pi(x)) : \pi(x) \notin \mathcal{Y}_\pi\}$.

(Under proper encoding) $|\text{Enc}(\pi)| \leq \log((N - a)!) + 2 \cdot \log \left(\binom{N}{a}\right)$ for $a = |\mathcal{Y}_\pi| \geq \frac{\varepsilon N}{q+1}$.

**Algorithm 7 (Dec($\mathcal{Y}, \mathcal{V}$))**

For all $y \in \mathcal{Y}$ in lexicographic order:

1. Emulate $D^\pi(y)$, answering $\pi$-query using $\mathcal{V}$.
2. If $D$ queries $x$ that is undefined in $\mathcal{V}$, add $(x, y)$ to $\mathcal{V}$.
   - Otherwise, add $(D^\pi(y), y)$ to $\mathcal{V}$.

Output $\mathcal{V}$.
Remarks

- Result can be proven using alternative compression arguments.
- Extends to random functions and random trapdoor permutation families (TDPs).
  
  HW2: state and prove for random function from $n$ bit to $n$ bits

- The short description argument is an very useful paradigm (another example soon).
Section 2

BB Impossibility for Basing OWP on OWF
OWF vs OWP

- Most of what we can based on OWPs can also be based on OWFs
  But constructions are much less efficient and way more complicated
- Can we base OWF on OWP?
- Actually the “correct” question, which we do not know how to answer, is can we base (almost) inject OWF on OWF
- We will show that we we cannot base OWP on OWF in a (fully) black box way
The settings

- Let $g$ be an $q(n)$-query oracle-aided function such that $g^f : \{0, 1\}^{m(n)} \mapsto \{0, 1\}^{m(n)}$ is a permutation for every $f : \{0, 1\}^n \mapsto \{0, 1\}^n$.

- For a start, assume that $m$ is increasing and efficiently invertible, and that on input of length $m(n)$, $g$ only call $f$ on inputs of length $n$.

- $R$ be the reduction: $R$ is an efficient algorithm such that $R^{f, \text{Inv}}$ breaks the one-wayness of $f : \{0, 1\}^n \mapsto \{0, 1\}^n$ for any $\text{Inv}$ that breaks that of $g^f$.

- For a start, assume for simplicity that on input of length $n$, $R$ only calls $f$ on input of length $n$. 
The attack

Algorithm 8 (The inverter $\text{Inv}^f(z \in \{0, 1\}^{m=m(n)})$

For $i = 1$ to $q = q(n)$

- Find (brute forcly) $x \in \{0, 1\}^m$ and $f' : \{0, 1\}^n \mapsto \{0, 1\}^n$ consistent with $S$ (initially empty) such that $g^{f'}(x) = z$
- Add $(x_1, f(x_1)), \ldots, (x_q, f(x_q))$ to $S$, for $x_1, \ldots, x_q$ being the queries made by $g^{f'}(x)$

Return $x \in \{0, 1\}^m$ such that all queries of $g^f(x)$ are in $S$ and $g^f(x) = z$.

- On $z \in \{0, 1\}^{m(n)}$, $\text{Inv}$ makes at most $q(n)^2$ queries.
- $\text{Inv}^f$ inverts $g^f$ with probability one. Proof
  - Let $x = (g^f)^{-1}(z)$.
  - In each round, $\text{Inv}(z)$ asks at least one additional query of $g^f(x)$ (unless it already found all of them)
  - When the loop ends, $\text{Inv}(z)$ asked all queries made by $g^f(x)$, and hence finds $x$
The impossibility result

- By assumption, for any $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$, $\exists p \in \text{poly}$ such that

$$\Pr_{x \leftarrow \{0,1\}^n, y = f(x)} \left[ R^{\text{Inv}, f}(y) \in f^{-1}(y) \right] \geq 1/p(n)$$

- $R^{\text{Inv}, f}(y \in \{0, 1\}^n)$ makes polynomial number of queries.

- For large enough $n$, let $f_n : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be such that no $2^{n/5}$ algorithm inverts with probability larger than $2^{-n/5}$, and let $f = \{f_n\}$

- Hence, $\Pr_{x \leftarrow \{0,1\}^n, y = f(x)} \left[ R^{\text{Inv}, f}(y) \in f^{-1}(y) \right] < 2^{-n/5}$ for every $n$. A contradiction.

- What goes wrong if we allow $g$ to query $f$ on different input length?
- Essentially nothing, HW 3

- What goes wrong if we allow $R$ to query $f$ on different input length?
Handling reductions of arbitrary query lengths

Let \( F_n \) be all length preserving function over \( \{0, 1\}^n \) and let \( F = \{F_n\}_{n \in \mathbb{N}} \)

For large enough \( n \),

\[
\Pr_{\pi \leftarrow F} \left[ \Pr_{x \leftarrow \{0, 1\}^n, y = f(x)} \left[ R_{\text{Inv}, f}^l (y) \in f^{-1}(y) \right] \geq 2^{-n/5} \right] < 2^{-2^{3n/5}}
\]

Hence, \( \not\exists p \in \text{poly} \) s.t. \( \Pr_{x \leftarrow \{0, 1\}^n, y = f(x)} \left[ R_{\text{Inv}, f}^l (y) \in f^{-1}(y) \right] \geq p(\delta(n)) \), for \( \delta \) being the success probability of \( \text{Inv} \) in inverting \( m(n) \) bit input

**Lemma 9 (Borel-Cantelli)**

Let \( E_1, \ldots, E_2, \ldots \), be a sequence of events such that \( \sum_{n=1}^{\infty} \Pr[E_n] < \infty \), then

\[
\Pr[\limsup_{n \to \infty} E_n = \bigcap_{n=1}^{\infty} \bigcup_{k=n}^{\infty} E_k] = 0.
\]

Namely, with probability one, only finite number of event occur.

Let \( E_n \) be the event over the choice of \( f \leftarrow F \) that

\[
\Pr_{x \leftarrow \{0, 1\}^n, y = f(x)} \left[ R_{\text{Inv}, f}^l (y) \in f^{-1}(y) \right] \geq 2^{-n/5}
\]

(for large enough \( n \)) \( \Pr[E_n] \leq 2^{-2^{3n/5}} < 1/n^2 \)

Hence, \( \sum_{n=1}^{\infty} \Pr[E_n] \leq O(1) + \sum_{n=1}^{\infty} 1/n^2 \in O(1) \)

Hence, with probability one over the choice of \( f \), only for finite finite number of \( n \) it holds that \( \Pr_{x \leftarrow \{0, 1\}^n, y = f(x)} \left[ R_{\text{Inv}, f}^l (y) \in f^{-1}(y) \right] \geq 2^{-n/5} \).
Remarks

- Extend to many constructions of prefect object from a imperfect object
- Other example: Perfect KA from OWF, perfect encryption schemes from a non perfect one.
- Seem to be “less significant” impossibility results
- In some cases, and other the right assumption, can be bypassed using non-black-box technique
BB Impossibility for OWF based PRG

**Definition 10 (pseudorandom generators (PRGs))**

Poly-time $G: \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a pseudorandom generator, if

- $G$ is length extending (i.e., $\ell(n) > n$ for any $n$)
- $G(U_n)$ is pseudorandom (i.e., $\{G(U_n)\}_{n \in \mathbb{N}} \approx_c \{U_{\ell(n)}\}_{n \in \mathbb{N}}$)

We focus on BB constructions of efficient length-doubling PRGs.

**Theorem 11**

*In any fully-BB construction of length-doubling PRG over $n$-bits string from OWP over $\{0, 1\}^n$, the construction makes $\Omega(n/\log n)$ oracle calls.*

- Matches known upper bounds.
- Without the restriction on the OWP input length, yields an optimal $n^{\Omega(1)}$ bound.
Proving Thm 11, cont.

- Let \((C, R)\) be a fully-BB reduction of a \(q(n) \in o(n/\log n)\)-query, length-doubling PRG over \(\{0, 1\}^n\), to OWP over \(\{0, 1\}^n\).

- Assume w.l.o.g. that \(C\) makes distinct queries.

- Assume for simplicity that on inputs of length \(n\), \(R\) only makes length \(n\) queries.

- For \(t = t(n) = \lfloor n/2q(n) \rfloor\), consider the following generator \(G: \{0, 1\}^{3n} \mapsto \{0, 1\}^{2n}\):

\[
\text{Algorithm 12 } (G(x))
\]

1. Emulate \(C^\pi(x_1, \ldots, n)\), while answering the \(i\)'th query \(z\) of \(C\) to \(\pi\), with \(x_{n+i \cdot t+1, \ldots, n+(i+1) \cdot t} \circ Z_{t+1, \ldots, n}\).

2. Output the same output that \(C\) does.

- Let \(\Pi_{n,t}\) be the set of all permutations over \(\{0, 1\}^n\) that are identity over the last \(n - t\) bits (i.e., \(\pi(x)_{n-t+1, \ldots, n} = x_{n-t+1, \ldots, n}\)).

- It holds that \(G(U_{3n/2}) \equiv (C^\pi(U_n))_{\pi \leftarrow \Pi_{n,t}}\).
Proving Thm 11, cont.

- ∃ algorithm \( D \) that distinguishes \( G(U_{3n/2}) \) from \( U_{2n} \) with advantage 
  \[ 1 - 2^{-n/4} > \frac{1}{2} \]. (?)

\[ \implies \text{wlg. } \Pr_{\pi \leftarrow \Pi_{t,n}} [D(C^{\pi}(U_n)) = 1] - \Pr[D(U_{2n}) = 1] > \frac{1}{2} \]

\[ \implies \Pr_{\pi \leftarrow \Pi_{t,n}} [\Pr[D(C^{\pi}(U_n)) = 1] - \Pr[D(U_{2n}) = 1] > \frac{1}{4}] \geq \frac{1}{4} \]

\[ \implies \Pr_{\pi \leftarrow \Pi_{t,n}} [R^{\pi,D} \text{ inverts } \pi \text{ with non-negligible prob.}] \geq \frac{1}{4} \]

- Let \( n' = t(n) \in \omega(\log n) \).

- By the above, exists \( 2^{o(n')} \)-query circuit \( R' \), such that

\[ \Pr_{\pi \leftarrow \Pi_{n'}} [R'^{\pi} \text{ inverts } \pi \text{ with non-negligible prob.}] \geq \frac{1}{4}, \]

in contradiction to Thm 3.
Remarks

- We showed a lower bound on the efficiency of fully-BB constructions of length-doubling PRG from OWPs.
- Actually we ruled out a less restricted type of BB-construction, called weak black box:
  - If \( f \) is a secure implementation of \( A \), then \( C^f \) is a secure implementation of \( B \).
- Results extend to OWFs and TDPs.
- Using similar means, one can prove lower bound on fully-BB constructions of encryption schemes, signature schemes and universal-one-way-hash-functions (UOWHFs), from OWFs/OWPs/TDPs.