CSG399: Gems of Theoretical Computer Science.	Lecture 12. Feb. 17, 2009.
Instructor: Emanuele Viola	Scribe: Dimitrios Kanoulas

Cryptography in constant depth: I

In this class we will make some remarks on Ben-Or and Cleve's proof of Barrington's theorem; we will also examine Applebaum, Ishai, and Kushilevitz's cryptograhy in constant depth.

1 Remarks on Barrington's theorem

Recall Barrington's theorem:

Theorem 1. *Let $f : \{0,1\}^n \to \{0,1\}$ have fan-in 2 circuits of depth d. Then there exists a group program $(g_1^0, \ldots, g_l^0)$, $(g_1^1, \ldots, g_l^1)$, $(k_1, \ldots, k_l)$ and $l = 4^d$, such that $\forall x$, $\Pi g_i^{x_{k_i}} = \alpha^{f(x)}$, where $g_i^0, g_i^1 \in S_5$, the group of permutations of 5 objects, and $\alpha = (1\ 2\ 3\ 4\ 5) \in S_5$.*

Ben-Or and Cleve gave a variant of Barrington's theorem. Under the same hypothesis, they show that there exist $l = O(1)^d$ 3×3 matrices $M_1, \ldots, M_l$, over $GF(2) = \{0,1\}$, where for each i, each entry in M_i is either $0, 1$ or x_j for some j, such that $\forall x$,

$$\Pi_{i=1}^{l} M_i = \begin{pmatrix} 1 & 0 & f(x) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

A typical matrix M_i in the above product is

$$M_i = \begin{pmatrix} 1 & 0 & x_7 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

2 Fast (shallow) cryptography

Cryptography is the study of protocols that achieve various communication goals in a way that is secure against adversaries that are more powerful than the honest parties. These protocols are built out of various primitives, like one-way functions. A one-way function is a function that is easy to compute but "hard to invert," the computational equivalent of physical actions that are hard to reverse, such as breaking a glass. We now define "hard to invert."

Definition 2. *A function $f : \{0,1\}^n \to \{0,1\}^n$ is w-hard to invert (w-HI) if every circuit C of size w satisfies: $Pr_{x \in \{0,1\}^n}[C(f(x)) \in f^{-1}(f(x))] \leq 1/w$.*

Note: To reduce the number of parameters we make the output length and the input length coincide, and similar choices are made later. This can always be achieved up to polynomial losses which is what we are willing to tolerate in this context.

A typical setting of parameters, corresponding to "one-way functions," is a function $f : \{0,1\}^n \to \{0,1\}^n$ that is poly(n)-time computable and also $n^{\omega(1)} - HI$. One-way functions have lots of applications. A one-way function immediately gives rise to a "basic signature scheme" where one can sign a message by writing $f(x)$ into it, and later claim ownership by exhibiting x. More sophisticated and useful signature schemes can be built as well. One-way functions also imply the existence of a type of pseudorandom generators that is different from the one we saw before. Specifically from a one-way function one can construct a pseudorandom generator that looks random even to adversaries that have *more* resources than those used to compute the generator, whereas earlier in this course we saw a different paradigm which gives pseudorandom generators that fool tests using *less* resources than those used in computing the generator.

Multiplication $f(a,b) := a \cdot b$ looks somewhat one-way, because it seems like we need to factor the output to get back (a,b), a task for which the best known algorithm runs in exponential time. However that does not quite work immediately, because $z := a \cdot b$ is even with probability at least $1/2$ and thus we can invert it as $z = f(2, z/2)$. Still, when a, b happen to be prime, the function is believed to be hard to invert. Although primes are only a polynomial fraction of the numbers, one can get a one-way function via the *direct product* construction $f(a_1, b_1, \ldots, a_l, b_l) := (a_1 \cdot b_1, \ldots, a_l \cdot b_l)$, for $l = \text{poly}(n)$.

Thus we see that we have candidate one-way functions that are quite easy to compute. For example, we have seen that multiplication can be computed by circuits of depth $d = O(\log n)$. Today and the next times we will see that if there are one-way functions that can be computed by such circuits of log depth, then in fact there are one-way functions that are computed by circuits of constant depth. These circuits have bounded fan-in, and so our one-way function will have the local property that *each output bit depends on a constant number of input bits.*

Theorem 3. *Suppose $f = \{0,1\}^n \to \{0,1\}^n$ is computable by Branching Programs of width n and length n (i.e. n^2 nodes) and f is $w - HI$, for some w. Then $\exists f' : \{0,1\}^{n'} \to \{0,1\}^{n'}$ such that each output bit of f' depends on 4 input bits, $n' = n^{O(1)}$, and f' is $w' - HI$, for $w' = w - n^{O(1)}$.*

The above theorem shows how we can construct a function f' from f which is very efficient and nearly as secure. Observe in particular that Theorem's 3 hypothesis holds for functions f that have circuits of depth $O(\log n)$.

In our approach we will use a randomized encoding of f.

Definition 4 (Randomized encoding)**.** *A randomized encoding of $f : \{0,1\}^n \to \{0,1\}^n$ with blow-up t is a function $f' : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^t$, such that:*

1. *$\forall x, x', r, r'$: $f(x) \neq f(x') \Rightarrow f'(x,r) \neq f'(x',r')$, and*
2. *there exists a distribution D on circuits C of size t: $\forall x$, the distribution $C(f(x))$ for $C \in D$ is equal to the distribution of $f'(x,r)$ for uniform r.*

We now prove Theorem 3 assuming there exists a randomized encoding f' of f with blow-up $t = n^{O(1)}$ and such that each output bit of f' depends on 4 input bits.

Proof of Theorem 3 under above assumption. f' obviously has the "4-bit" property and also $n' = n^{O(1)} \Leftarrow t = n^{O(1)}$. To complete the proof, we need to prove that the Hard-to-Invert property holds. To prove this assume for contradiction that A' inverts f':

$$\Pr_{x,r}\left[A'(f'(x,r)) \in f'^{-1}(f'(x,r))\right] \geq 1/w',$$

where the size of A' is at most $w' := w - n^{O(1)}$. Define a distribution on A:

$$\text{On input } z\text{: } A(z) := A'(C(z))\mid_n, \text{ where } C \text{ is random from } D.$$

Note $(size\ of\ A) = (size\ A') + (size\ C) \leq w' + n^{O(1)} = w$. Also:

$$\begin{aligned}
\Pr_{x,A}[A(f(x)) \in f^{-1}(f(x))] &= \Pr_{x,C}[A'(C(f(x)))\mid_n \in f^{-1}(f(x))] \\
&= \Pr_{x,r}[A'(f'(x,r))\mid_n \in f^{-1}(f(x))] \\
&\geq 1/w'
\end{aligned}$$

which means that A inverts f with probability at least $1/w$. Fixing the choice of $C \in D$ in the definition of A we contradict the assumption that f is $w - HI$.

In the above derivation, the first equality is by definition. The second is Property 2 of Definition 4. The last inequality follows from our assumption on A' and because we claim that whenever the output $A'(f'(x,r)) = (x',r')$ satisfies $f'(x,r) = f'(x',r')$ then also $f(x) = f(x')$. This is because otherwise, by Property 1 of Definition 4, $f'(x,r) \neq f'(x',r')$. □

In the next classes we will construct randomized encodings.