CSE300 - Modern Cryptography	Lecture Date here
Lecture #10	Instructor and Scribe: Grzegorz Malewicz (Greg)

1 Fillings and preservings

Normally we consider functions that are hard to invert for strings of all lengths starting from length n_0. It may be difficult to produce such a function, and perhaps it is easier to design a hard-to-invert function that is hard to invert for strings of length, say $1, 2^2, 3^2, 4^2, 5^2, \ldots$. It turns out that if someone gave us a construction that is hard for lengths that are dense enough then we can build a function that is hard for all lengths, by "filling in the gaps". We formalize these intuitions below. The theorems are needed to prove the amplification theorem.

Definition 1 *A function $f : \Sigma^* \rightarrow \Sigma^*$ is called a (strong) sparse one-way function for $B \subseteq N$ if*

- *f is efficiently computable, and*
- *for any efficient randomized algorithm A, with coins r(n), and any polynomial $q(n)$, $\exists n_0\ \forall n > n_0, n \in B$*

$$Pr_{x\in\{0,1\}^n, c\in\{0,1\}^{r(|f(x)1^n|)}}[f(A(f(x)1^n c)) = f(x)] \leq \frac{1}{q(n)}$$

Observe that for $B = N$ this definition reduces to the original definition of one-way function. We say that a function is *polynomially sparse* if it is hard to invert at least for polynomially dense points, i.e. for $B = \{\beta(n)|n \in N\}$, where $\beta(n)$ is a polynomial. This is a stronger definition that the one encountered in literature, but it is enough for the purpose of proving amplification theorem. Typical definition is $\exists\beta$ a polynomial $\exists n_0$, such that $\forall n > n_0\ |B \cap \{1, 2, \ldots, \beta(n)\}| \geq n$.

Theorem 1 *The following are equivalent:*
(a) there exists a OWF
(b) there exists a polynomially sparse OWF
(c) there exists a polynomially sparse length-preserving OWF
(d) there exists a length-preserving OWF

Proof: $(a) => (b)$ and $(d) => (a)$ are trivial
$(b) => (c)$
We assume that all polynomials that appear in the proof are monotonically increasing. Let f be polynomially sparse OWF. Let $s(n)$ be a polynomial that bounds the size of the output for input of size n. Select a polynomial $s'(n)$ such that for all n, $s'(n) \geq s(n) + 2$ and $s'(n) \geq n$. We define a function $f'(n)$ and show that it is polynomially

sparse length-preserving OWF. We specify the value of f' for inputs of length in $B' = \{s'(\beta(k)) : k \in N\}$ (for other lengths take f' to be identity).

$$f'(x) = f(x|_{\beta(k)})01^{s'(\beta(k))-|f(x|_{\beta(k)})|-1}, \quad where \ |x| = s'(\beta(k))$$

Since $|f(x|_{\beta(k)})| \leq s(\beta(k))$, and $s'(\beta(k))$ was chosen to be at least by two greater than $s(\beta(k))$ the tail of $f'(x)$ is 0 and at least one 1. Observe that for any $x \in \{0,1\}^{s'(\beta(n))}$, $\tilde{x} \in \{0,1\}^{s'(\beta(m))}$ if $f'(x) = f'(\tilde{x})$ then $f(x|_{\beta(n)}) = f(x|_{\beta(m)})$, and $m = n$ (because we use a separating 0 in f').

Assume by the way of contradiction that f' is not a sparse OWF for B'. Hence there exists an efficient randomized algorithm A with coins $r(n)$ and a polynomial $q(n)$ such that $\forall n_0 \ \exists n > n_0, n \in B'$ such that

$$Pr_{x \in \{0,1\}^n, c \in \{0,1\}^{r(|f'(x)1^n|)}}[f'(A(f'(x)1^n \cdot c)) = f'(x)] > \frac{1}{q(n)}$$

We use algorithm A to invert f with high probability for x of lengths in B. Observe that we don't have to invert f for all lengths in B. It is sufficient if we show that we can often invert f for infinitely many lengths from B. Then we are done because in such a case there is no n_0 such that $\forall n > n_0$ and $n \in B$ f is hard to invert, and hence the existence of infinitely many lengths would contradict the assumption that f is a sparse OWF for B.

We build a set $EASY \subseteq B$ of easily invertible lengths inductively. We start with a set $\{n_2\}$ (the only length that may not be easily invertible) and provide a procedure that adds a new element to this set. When we apply this procedure over and over again we get an infinite set, which yields the desired contradiction. Here is how the procedure works.

Take the least n_0 such that n_0 is greater than $s'(e)$ for any $e \in EASY$. By assumption $\exists n > n_0$ $n \in B'$ such that our algorithm A is able to often invert $f'(x)1^n$ for all x of length n. Since $n \in B'$ we have that $n = s'(\beta(k))$, and from the choice of n_0 we know that k is greater than any element in $EASY$ (if $\exists x \in EASY x > \beta(k)$ then because of monotonicity of s' $n > n_0 > s'(x) \geq s'(\beta(k)) = n$, a contradiction. Since $\beta(k) \in B$ it is sufficient to show that there is an algorithm A' that often inverts x of length $\beta(k)$. This will show that we can expand the set $EASY$ by a new element from B, and complete the proof.

Algorithm A' works as follows. Intuitively, on observing a string $y = f(x)1^{\beta(k)}$, where x was drawn from $\{0,1\}^{\beta(k)}$, A' cannot automatically apply A to the string. First we don't know the value of $\beta(k)$ but we have to guarantee that we give A a string of form $f'(x)1^{\beta(k)}$, because we know that A often invert such a string. Since $f(x|_k) = f(x)$, we have to append some $1s$ to string y and insert a 0 at the right place. Fortunately there are only polynomial in k possibilities. So chances are high that if we insert 0 at a random place we get it right :).

Formally A' does the following. It takes $y = f(x)1^{\beta(k)}$, for some $x \in \{0,1\}^{\beta(k)}$. Then selects uniformly at random $i \in \{0, \ldots, 2s'(|y|)\}$, and $j \in \{0, \ldots, s'(|y|)\}$ Then it takes string $f(x)1^{k+i}$ and inserts 0 at the j-th location in string (if j is greater than the length of the string it places 0 at the end if it). Let us denote the resulting string $\tilde{y}$. Then it runs A on $\tilde{y}$, $\tilde{x} = A(\tilde{y})$. Finally A' select uniformly at random $c \in \{0, \ldots, s'(|y|)\}$ and returns $\tilde{x}|_c$ (if c is greater than $|\tilde{x}|$ it returns $\tilde{x}$).

First observe that the algorithm is efficient. Its running time is bounded by the running time of A for input of length at most ..., which is a polynomial on $\beta(k)$. Now we analyze the probability that A' inverts.

Observe that for some i and j in the range they were chosen from $\tilde{y} = f(x)01^{\beta(k)}$, and $|\tilde{y}| = 2s'(\beta(k))$. We call such i and j *good*. There are at most $p_1(\beta(k)) = 2s'(2s'(\beta(k))) + 2s'(\beta(k))$ pairs, which is a polynomial of $\beta(k)$. Hence the probability that good i and j are chosen is at most $\frac{1}{p_1(\beta(k))}$. If a good pair is chosen then whenever A inverts we have $A(f(x)01^{s'(\beta(k))-|f(x)|-1}1^{s'(\beta(k))}coins) = \tilde{x}$, and $f(\tilde{x}) = f(x)01^{s'(\beta(k))-|f(x)|-1}$, but this is equal to $f'(x1^{s'(\beta(k))-\beta(k)}$, hence $f(\tilde{x}|_{beta(k)}) = f(x)$. So $\tilde{x}|_{beta(k)}$ is an inverse of $f(x)$. The probability that A' returns $\tilde{x}|_{beta(k)}$ provided that it inverted for good pair is at least $\frac{1}{s'(|y|)+1} \geq \frac{1}{p_1(\beta(k))}$.

$$\Pr_{x\in\{0,1\}^{\beta(k)},c}[f(A'(f(x)\cdot 1^{beta(k)}\cdot c)) = f(x)] \geq$$

$$\geq \Pr[\, i,j \; is \; good] \cdot \Pr_{x\in\{0,1\}^{\beta(k)},c}[A(f(x)01^{s'(\beta(k))-|f(x)|-1}1^{s'(\beta(k))}c) \; inverts] \cdot$$

$$\Pr[A' \; returns \; correct \mid i,j \; good \; and \; A \; inverts] \geq$$

$$\geq \frac{1}{p_1(\beta(k))^2} \cdot \Pr_{x\in\{0,1\}^{\beta(k)},c}[A(f(x)01^{s'(\beta(k))-|f(x)|-1}1^{s'(\beta(k))}c) \; inverts] \geq$$

Now observe that instead of drawing x from $\{0,1\}^{\beta(k)}$ we can draw it from a possibly larger space $\{0,1\}^{s'(\beta(k))}$ and substitute $f(x)$ with $f'(x|_{\beta(k)})$, and despite that change the probability does not change. This is because the first $\beta(k)$ bits of x determine $f'(x)$, and for fixed first $\beta(k)$ bits and coins, and any selection of the remaining $s'(\beta(k)) - \beta(k)$ extra bits algorithm A behaves the same (it sees the same input so it has to behave the same). So we can use the bound on probability that A inverts.

$$> \frac{1}{p_1(\beta(k))^2} \cdot \frac{1}{q(\beta(k))}$$

The last is an inverse of a polynomial of $\beta(k)$. This shows that the probability that A' inverts $f(x)$ is big, so $\beta(k)$ can be added to set $EASY$.

$(c) => (d)$
Let f be a strong polynomially sparse OWF. We will build a new function f' and show that it is a strong OWF. We assume that all polynomials that appear in the proof are monotonically increasing (if not then we start from a place n_2 to the right of which they are - possible because the coefficient for the highest power is positive).

We define f as follows. For x such that $|x| \in B$, we put $f'(x) = f(x)$. For other x we know that the length of x is not in B, so we take k such that $\beta(k) < |x| < \beta(k+1)$, then take the first $\beta(k)$ bits of x, denoted $x|_{\beta(k)}$, and apply function f to the truncated x, finally we augment the result with extra ones.

- $f'(x) = f(x)$, when $|x| \in B$
- $f'(x) = f(x|_{\beta(k)})1^{|x|-\beta(k)}$, when $|x| \notin B$, $\beta(k) < |x| < \beta(k+1)$

Observe that since f is length-preserving then f' is also length-preserving.

We show that f' is OWF. Assume by the way of contradiction that it is not. Hence there exists an efficient randomized algorithm A with coins $r(n)$ and a polynomial $q(n)$ such that $\forall n_0\ \exists n > n_0$ such that

$$Pr_{x\in\{0,1\}^n, c\in\{0,1\}^{r(|f'(x)1^n|)}}[f'(A(f'(x)1^n \cdot c)) = f'(x)] > \frac{1}{q(n)}$$

We use algorithm A to invert f with high probability for x of lengths in B. Observe that we don't have to invert f for all lengths in B. It is sufficient if we show that we can often invert f for infinitely many lengths from B then we are done because in such a case there is no n_0 such that $\forall n > n_0$ and $n \in B$ f is hard to invert, and hence this the existence of infinitely many lengths would contradict the assumption that f is sparse OWF for B.

We build a set $EASY \subseteq B$ of easily invertible lengths inductively. We start with a set $\{n_2\}$ (the only length that may not be easily invertible) and provide a procedure that adds a new element to this set. When we apply this procedure over and over again we get an infinite set, which yields the desired contradiction. Here is how the procedure works.

Take the least k for which $\beta(k)$ is greater than any of elements in $EASY$ and put $n_0 = \beta(k)$. There is always such k because β tends to infinity. By assumption $\exists n > n_0$ such that our algorithm A is able to often invert $f'(x)$ for all x of length n. Select k' so that $\beta(k') \leq n < \beta(k'+1)$. Let us denote $m = \beta(k')$. Observe that $m \leq n \leq C \cdot m$, where C is a constant dependent only on β. Why can we bound n from above? Because for x big enough we have:

$$\beta(x+1) = \sum_{i=0}^{t} a_i(x+1)^i \leq 2(x+1)^t \leq 2(2x)^t = 2^{t+1}x^t \leq 2^{t+2}\sum_{i=0}^{t} a_i x^i = 2^{t+2}\beta(x)$$

In the same style we can show that for any polynomial $w(x)$ with positive highest coefficient, any $c >$, and big enough x, there exists c' that $w(cx) \leq c'w(x)$ (exercise left for the reader).

¿From the choice of n_0 we know that m is greater than any element in $EASY$. So it is sufficient to show that there is an algorithm A' that often inverts x of length m. This will show that we can expand the set $EASY$ by a new element, and complete the proof.

Algorithm A' works as follows. Intuitively, on observing $f(x)1^m$ A' cannot automatically apply A to the string. Perhaps $n > \beta(k') = m$ and A may not be able to often invert if we give it the input $f(x)1^m$. Hence we have to try all possible lengths that $|x|$ could take. Fortunately there are only polynomial number of choices. Formally A' does the following. It takes $f(x)1^m$, for $x \in \{0,1\}^m$, and runs A on the following inputs: $f(x)1^0 1^m \cdot coins_m, f(x)1^1 1^{m+1} \cdot coins_{m+1}, \ldots f(x)1^{C\cdot m-m}1^{C\cdot m} \cdot coins_{Cm}$, where $coins_j$ is a string of random coins of length $r(|f(x)1^j|)$ (we have to augment the string with as many coins as A requires for input of given length). Let $x'_i = A(f(x)1^{m+2i}), i = 0, \ldots, m(C-1)+1$, be the results of A on input i. Then A' checks if $f'(x'_i) = f(x)1^i = f'(x \cdot 1^i)$ for any i. If it is true for some i then the algorithm A' returns $x'_i|_{\beta(k'')}$, where $\beta(k'') \leq |x'_i| < \beta(k''+1)$. If for all i $f'(x'_i) \neq f(x)1^i$ algorithm A' returns 1^m.

First observe that the algorithm is efficient. Its running time is bounded by a polynomial on m, i.e. by $C \cdot m$ times the running time of A for input of length $s(Cm)+Cm$, where $s(x)$ is the polynomial that bounds the size of the output of f depending on the size of input. Next it requires at most a polynomial in m number of coins, i.e. at most $r'(m) = Cm * (s(m)+m)$. Now we analyze the probability that A' inverts.

Since $m \leq n \leq Cm$ one of the runs of A that A' does will be with $f(x)$ augmented with the string $1^{n-m}1^n$. Suppose that A inverts $f(x)1^{n-m}1^n$. This means that $A(f(x)1^{n-m}1^n) = x'$, such that $f'(x') = f(x)1^{n-m} = f(x)1^{n-\beta(k)}$. But then algorithm A' returns $x'|_{\beta(k'')}$. Now by the definition of f', $f'(x') = f(x'|_{\beta(k'')})1^{|x'|-\beta(k'')}$. Since f is length-preserving the length of $f(x)1^{n-\beta(k)}$ is n, and the length of $f(x'|_{\beta(k'')})1^{|x'|-\beta(k'')}$ is $|x'|$. Moreover $k = k''$ (if $k < k''$ then, because $\beta(k) \leq n < \beta(k+1)$ and $\beta(k'') \leq |x'| < \beta(k''+1)$, $n < |x'|$, a contradiction), so $f(x'|_{\beta(k'')}) = f(x)$, and so $x'|_{\beta(k)}$ is an inverse of $f(x)$. Therefore we can write:

$$\Pr_{x \in \{0,1\}^m, c \in \{0,1\}^{r'(m)}} [f(A'(f(x) \cdot 1^m \cdot c)) = f(x)] \geq$$

$$\geq \Pr_{x \in \{0,1\}^m, c \in \{0,1\}^{r(|f(x)1^n|)}} [f'(A(f(x) \cdot 1^{n-m} \cdot 1^n \cdot c)) = f(x) \cdot 1^{n-m}] =$$

Now observe that instead of drawing x from $\{0,1\}^m$ we can draw it from a possibly larger space $\{0,1\}^n$ and substitute $f(x)$ with $f'(x)$, and despite that the probability does not change. This is because the first m bits of x determine $f'(x) = f(x|_m) \cdot 1^{n-m}$, and for fixed first m bits and coins, and any selection of $n-m$ extra bits algorithm A behaves the same (it sees the same input so it has to behave the same).

$$= \Pr_{x \in \{0,1\}^n, c \in \{0,1\}^{r(|f'(x)1^n|)}} [f'(A(f'(x) \cdot 1^n \cdot c)) = f'(x)] >$$

and from the assumption that A is an inverter we have

$$> \frac{1}{q(n)} \geq \frac{1}{q(Cm)} \geq \frac{1}{C'q(m)}$$

Constant C' is independent on m (depends on C and q only). This shows that A' can invert with high probability. So it is possible to often invert $f(x)$ for $|x| = m$, and m is not yet in set $EASY$. Hence the set can be expanded.

Theorem 2 *The following are equivalent:*
(a) there exists a weak OWF
(b) there exists a weak polynomially sparse OWF
(c) there exists a weak polynomially sparse length-preserving OWF
(d) there exists a weak length-preserving OWF

Proof: The proof is similar to the proof of Theorem 1. We leave it for the reader