Columbia University
E6291: Topics in Cryptography
Professor Luca Trevisan

Handout 5
February 10, 1999

A Weaker Version of the Goldreich-Levin Result

These notes contain a complete proof of a weaker (but simpler) version of the result of Goldreich and Levin.

Goldreich and Levin prove[1] that if f is a strong one-way injective function then it is hard to guess the XOR of a random subset of the bits of x given $f(x)$. By "hard" we mean that the best guessing probability for an efficient algorithm is at most $1/2 + \nu(n)$, where $\nu()$ is negligible and n is the length of x. In this notes we present a much simpler calculation that shows that the guessing probability for an efficient algorithm is at most $3/4 + \nu(n)$.[2] This is a worthwhile calculation to study, because it gives some flavor of the techniques used to prove more general results and is a good way to familiarize with (applications of) results on expectations of random variables.

1 Statement of the Result

We use the notation $GL(y, r) = \sum_i x_i r_i \pmod 2$ where $y = f_n(x)$.

Lemma 1.1 *Let $f_n : \{0,1\}^n \to \{0,1\}^n$ be a family of permutations. Suppose there is an algorithm G such that*

$$\mathbf{Pr}[G(f_n(x), r) = GL(x, r)] \geq \frac{3}{4} + \epsilon(n)$$

(where the probability is taken over uniform x and r and over the internal randomization of G)

then there is an algorithm A such that

$$\mathbf{Pr}[A(f_n(x)) = x] \geq \frac{\epsilon(n)}{4}$$

(where the probability is taken over uniform x and over the internal randomization of A.)

Furthermore A runs by making $O(n(\log n)/\epsilon^2)$ calls to G.

As an immediate consequence we have the following theorem.

Theorem 1.2 *Suppose that f_n is a strong one-way permutation, then a polynomial time algorithm that is trying to compute $GL(x, r)$ given $f(x)$ and r has a guessing probability at most $3/4 + \nu(n)$ where ν is negligible and n is the length of x.*

[1] Their result is even more general than the one that we state.

[2] The observation that this weaker result is much simpler to prove is not new, and has appeared in several places.

2 Description of the Algorithm

In order to prove Lemma 1.1 we have to describe algorithm A and analyze its probability of success.

We first describe the algorithm

$A(y)$
begin
  for $i = 1$ **to** n
    // Tries to guess x_i where $x = f^{-1}(y)$
    for $j = 1$ **to** $k = (\ln 4n)/2\epsilon^2$
      pick at random $r \in \{0,1\}^n$
      let r' be equal to r except $r'_i = 1 - r_i$
      $b = G(y, r)$
      $b' = G(y, r')$
      $g_j = b \oplus b'$
    $x_i =$ value occurring more often in $g_1, \ldots, g_k$
  return x
end

3 Proof of Lemma 1.1

We shall call

$$I_n = \left\{ x \in \{0,1\}^n : \mathbf{Pr}[G(f_n(x), r) = GL(x, r)] \geq \frac{3}{4} + \frac{\epsilon}{2} \right\}$$

where the probability is taken over uniform r, and over the randomness of G.

Lemma 3.1 *I_n contains at least $(\epsilon/2)2^n$ elements.*

Proof. For a fixed x, call $succ(x) = \mathbf{Pr}[G(f_n(x), r) = GL(x, r)]$. This is a non-negative random variable such that $\mathbf{E}[succ(x)] \geq 3/4 + \epsilon$ (by the assumption of Lemma 1.1). Using Lemma 4.1 below, we can say

$$\mathbf{Pr}[succ(x) \geq 3/4 - \epsilon/2] \geq \epsilon/2$$

and so

$$\mathbf{Pr}[x \in I] \geq \epsilon/2$$

■

Lemma 3.2 *When algorithm $A()$ receives in input $f_n(x)$ with $x \in I_n$, then, for every i and every j, there is a probability $\geq 1/2 + \epsilon$ that when the following piece of code is executed, g_j contains the right value of x_i.*

```
pick at random r ∈ {0,1}^n
let r' be equal to r except r'_i = 1 − r_i
b = G(y,r)
b' = G(y,r')
g_j = b ⊕ b'
```

Proof. We note that it is always true that $GL(y,r) \oplus GL(y,r') = x_i$. This is because

$$GL(y,r) \oplus GL(y,r') = \sum_h x_h r_h + \sum_h x_h r'_h = \sum_h (r_h + r'_h) x_h = x_i \pmod 2$$

where the last step is due to the fact that $r_h + r'_h = 0 \pmod 2$ for all $h \neq i$, by definition of r'.

So every time both $b = GL(y,r)$ and $b' = GL(y,r')$ we have that $b \oplus b'$ is correct. By our assumption that $x \in I_n$, there are at most a fraction $1/4 - \epsilon/2$ of r for which $b \neq GL(y,r)$ and also a fraction at most $1/4 - \epsilon/2$ of r' for which $b' \neq GL(y,r')$, and so there is at a probability $\leq 1/2 - \epsilon$ that either $b \neq GL(y,r)$ or $b' \neq GL(y,r')$. We conclude that there is at least a probability $1/2 + \epsilon$ that both $b = GL(y,r)$ and $b' = GL(y,r')$. For a stronger reason there is at least a probability $1/2 + \epsilon$ that $b \oplus b'$ is correct. ■

Lemma 3.3 *When algorithm $A()$ receives in input $f_n(x)$ with $x \in I_n$, then, for every i, there is a probability $\geq 1 - 1/2n$ that the decoding of x_i given by the following piece of code is correct.*

```
for j = 1 to k = (ln 4n)/2ε^2
   pick at random r ∈ {0,1}^n
   let r' be equal to r except r'_i = 1 − r_i
   b = G(y,r)
   b' = G(y,r')
   g_j = b ⊕ b'
x_i = value occurring more often in g_1,...,g_k
```

Proof. Consider the random variables $X_1, \ldots, X_k$ where X_j is 1 if g_j is correct and $X_j = 0$ otherwise. By the previous lemma, we have $\mathbf{E}[X_j] \geq 1/2 + \epsilon$ for every j. Call $X = \sum_j X_j$. Using Chernoff, the probability of incorrect decoding is

$$\mathbf{Pr}\left[\sum_j X_j \leq k/2\right] \leq \mathbf{Pr}\left[|X - \mathbf{E}[X]| \geq \epsilon k\right] \leq 2e^{-2\epsilon^2 k} = e^{-\ln(4n)} = 1/2n$$

■

Lemma 3.4 *When algorithm $A()$ receives in input $f_n(x)$ with $x \in I_n$, then there is a probability $\geq 1/2$ that the decoding of x is correct.*

Proof. For each particular index i, the probability of making an error in decoding x_i is at most $1/2n$ by the previous lemma. Then the probability of making an error in at least one of the n coordinates is at most $n \cdot \frac{1}{2n} = \frac{1}{2}$. So the probability of making no error is at least $1 - 1/2 = 1/2$. ∎

Lemma 3.5 *Algorithm $A()$ has a probability $\geq \epsilon/4$ of decoding x correctly.*

Proof. There is a probability at least $\epsilon/2$ that a randomly chosen x is in I_n. Conditioned on this, the probability of correct decoding is at least $1/2$. So the probability of correct decoding is at least $\epsilon/4$. ∎

4 Probability Results

Lemma 4.1 *Let X be a random variable such that $0 \leq X \leq 1$. Then for every $0 < \epsilon < \mathbf{E}[X]$ we have*

$$\mathbf{Pr}[X \geq \mathbf{E}[X] - \epsilon] \geq \epsilon$$

Proof. Consider $\mathbf{Pr}[X < \mathbf{E}[X] - \epsilon]$. Then we have

$$\begin{aligned}
\mathbf{Pr}[X < \mathbf{E}[X] - \epsilon] &= \mathbf{Pr}[1 - X > 1 - \mathbf{E}[X] + \epsilon] \\
&\leq \frac{\mathbf{E}[1 - X]}{1 - \mathbf{E}[X] + \epsilon} \\
&= \frac{1 - \mathbf{E}[X]}{1 - \mathbf{E}[X] + \epsilon} \\
&= 1 - \frac{\epsilon}{1 - \mathbf{E}[X] + \epsilon} \\
&\leq 1 - \epsilon
\end{aligned}$$

And so

$$\mathbf{Pr}[X \geq \mathbf{E}[X] - \epsilon] = 1 - \mathbf{Pr}[X < \mathbf{E}[X] - \epsilon] \geq 1 - (1 - \epsilon) = \epsilon$$

∎

Lemma 4.2 (Chernoff's Bound) *Let $X_1, \ldots, X_k$ be independent 0/1 random variables with the same distribution. Let $X = \sum_i X_i$. Then*

$$\mathbf{Pr}[|X - \mathbf{E}[X]| \geq \epsilon n] \leq e^{-2\epsilon^2 k}$$