ECS 227: Modern Cryptography
Lecture 5

Analysis of the PRG(counter) Construction

Phillip Rogaway
25 January 1996

Lecture began by reviewing the notions of finite PRFs and finite PRPs. We then carefully went through two simple theorems, trying, very slowly, to understand the exact meaning of each.

5.5 Security of PRF(counter)

The PRP(counter) construction we gave last lecture works equally well if you start with a PRF $F : \{0,1\}^\kappa \times \{0,1\}^\ell \to \{0,1\}^L$. That is, the generator g defined from F in the PRF(counter) construction is the one which, on input of a string s, returns the $L2^\ell$-bit string $g(s) = F_s(\langle 0\rangle_\ell) \;\|\; F_s(\langle 1\rangle_\ell) \;\|\; \cdots \;\|\; F_s(\langle 2^\ell - 1\rangle_\ell)$.

Proposition 5.5.1 *Let $F : \{0,1\}^\kappa \times \{0,1\}^\ell \leftarrow \{0,1\}^L$ be a Finite PRP, and let g be the* PRP(counter) *generator defined from it. Suppose there is an adversary D which (t, μ, ϵ)-breaks g. Then there is an adversary $\bar{D}$ which $(\bar{t}, \bar{q}, \bar{\epsilon})$-distinguishes F from a random function, where*

$$\begin{aligned} \bar{t} &= t + c(\mu + \ell + L) \\ \bar{q} &= \left\lceil \frac{\mu}{L} \right\rceil \\ \bar{\epsilon} &= \epsilon \end{aligned}$$

Here c is an absolute constant which depends only on details of the model of computation, and $\bar{D}$ is a fixed algorithm (for ℓ and L), with black-box access to D.

The bounds in the theorem above are the sort of bounds we strive for: we insignificantly increase the time and the query-complexity of the adversary in order to achieve *same* advantage in breaking the low-level primitive (F) as we assumed for breaking the high-level primitive (g). We will not always be so fortunate! Nor will the proofs always be so straightforward.

Proof. Recall that $\bar{D}$ is given an oracle which is computing a function $f(\cdot)$ from ℓ-bits to L-bits. Adversary $\bar{D}$ is trying to make a good guess as to whether this function $f(\cdot)$ was chosen as a truly random function, $\rho(\cdot)$, or as a pseudorandom function, $F_a(\cdot)$, for a random key a. To accomplish this mission, $\bar{D}$ will behave as follows:

1. $\bar{D}$ runs D. Recall that D has access to an oracle on which is written a sequence of $L2^\ell$ bits, $b = b_0 b_1 b_2 \ldots b_{L2^\ell - 1}$.

2. Every time D goes to access a bit b_i of its presented string b, $\bar{D}$ calculates a bit to return to D as an answer. The bit which $\bar{D}$ computes is bit number i mod L of $f(\lfloor i/\ell \rfloor)$. That is, $\bar{D}$ makes an oracle call, if necessary, to compute $f(\lfloor i/\ell \rfloor)$; and then $\bar{D}$ returns the appropriate bit of this L-bit word. Bits are numbered left-to-right by $0, 1, \ldots, L-1$, and, for simplicity, we have now dropped the $\langle \cdot \rangle_\ell$-notation.

3. When D finally halts, outputting a bit β, $\bar{D}$ also halts, outputting the exact same bit, β.

This complete our description of the algorithm $\bar{D}$. Now we must argue that $\bar{D}$ achieves the stated bounds on $\bar{t}, \bar{q}, \bar{\epsilon}$. The first two are easy to verify (we assume a RAM model of computation sufficiently powerful to achieve the stated $\bar{t}$). It remains to show that $\bar{\epsilon} = \epsilon$.

By definition

$$\mathrm{Adv}_{\bar{D}}^{F,\rho} = \left|\Pr[\bar{D}^{F_a} = 1] - \Pr[\bar{D}^{\rho} = 1]\right|,$$

where we have made the obvious notational simplifications. But note the following:

- $\Pr[\bar{D}^{F_a} = 1] = \Pr[D^{g(r)} = 1]$. This is true because $\bar{D}$, if it is using an $F_a(\cdot)$ oracle, is computing the input to D using the exact same method that g specifies, when using a random function.

- $\Pr[\bar{D}^{\rho} = 1] = \Pr[D^{\$} = 1]$. This is true because if ρ is a random function then $\rho(0) \,.\, \rho(1) \,.\, \cdots \,.\, \rho(2^\ell - 1)$ is a random string.

Make sure you understand the above two statements. We conclude that

$$\begin{aligned} \mathrm{Adv}_{\bar{D}}^{F,\rho} &= \left|\Pr[D^{g(r)} = 1] - \Pr[D^{\$} = 1]\right| \\ &= \mathrm{Adv}_{D}^{g,\$}, \end{aligned}$$

as required. $\diamondsuit$

5.6 Relating PRFs and PRGs

We have now shown how to get a good PRG given a good Finite PRF. But all we have are Finite PRP. The next proposition says that, even without modification, our Finite PRP *is* a reasonable Finite PRF.

Proposition 5.6.1 *Let $F\{0,1\}^\kappa \times \{0,1\}^\ell \to \{0,1\}^\ell$ be a Finite PRP. Suppose adversary D (t,q,ϵ)-distinguishes F from a random function. Then D $(t,q,\epsilon - q^2 2^{-\ell-1})$-distinguishes F from a random permutation.*

The proof is simple inclusion/exclusion calculation.

Proof. The assumption in the theorem translates to

$$\begin{aligned}\epsilon &\leq \mathrm{Adv}_D^{F,\rho} \\ &= \left|\Pr\left[D^\rho = 1\right] - \Pr\left[D^{F_a}\right]\right|\end{aligned}$$

Assume wlog that the term inside the absolute value signs is positive. In the experiment on the left hand side (D running with its $\rho(\cdot)$-oracle) let B (bad) be the event that D asks some pair of distinct questions, x and x', for which ρ returns the *same* answer, $\rho(x) = \rho(x')$. Then

$$\begin{aligned}\epsilon &\leq \Pr\left[D^\rho = 1|\overline{B}\right]\Pr\left[\overline{B}\right] + \Pr\left[D^\rho = 1|B\right]\Pr\left[B\right] - \Pr\left[D^{F_a}\right] \\ &\leq \Pr\left[D^\rho = 1|\overline{B}\right] + \Pr\left[B\right] - \Pr\left[D^{F_a}\right] \\ &\leq \left|\Pr\left[D^\rho = 1|\overline{B}\right] - \Pr\left[D^{F_a}\right]\right| + \Pr\left[B\right] \\ &\leq \left|\Pr\left[D^\pi = 1\right] - \Pr\left[D^{F_a}\right]\right| + \Pr\left[B\right] \\ &\leq \mathrm{Adv}_D^{F,\pi} + \Pr\left[B\right].\end{aligned}$$

The forth line follows because, as long as $\overline{B}$ does not happen, adversary D sees, in response to each distinct query, a random string distinct from any string as yet returned; this is the same distribution on strings which D sees in the $\pi(\cdot)$-experiment. We conclude that

$$\mathrm{Adv}_D^{F,\pi} \;\geq\; \mathrm{Adv}_D^{f,\rho} - \Pr[\mathsf{B}]$$

Now we calculate $\Pr[\mathsf{B}]$. Let B_i, for $i \geq 0$, be the event that the first time there is a collision in the returned $\rho(\cdot)$ values occurs in response to the i-th query. Since B is the disjoint union of events B_i, for $1 \leq i \leq q$,

$$\begin{aligned}\Pr\left[\mathsf{B}\right] &= \sum_{i=1}^{q} \Pr\left[\mathsf{B}_i|\overline{\mathsf{B}_{i-1}}\right] \\ &= \sum_{i=1}^{q} \frac{i}{2^\ell} \\ &= \frac{q(q-1)}{2} 2^{-\ell} \\ &\leq q^2 2^{-\ell-1}\end{aligned}$$

This completes the proof of the proposition. $\diamondsuit$
Putting together Propositions 5.5.1 and 5.6.1 immediately gives us Proposition 5.3.1.

5.7 An Attack on PRP(counter)

Concluding this section, we attack the PRP(counter) generator, giving an essentially matching bound. To be concrete and have it for future reference, let us state the following "birthday bound."

Lemma 5.7.1 *Let $P(m,t)$ denote the probablity of at least one collision in the experiment of throwing t balls, independently and at random, into m. Then Then*

$$1 - e^{-\frac{t^2-t}{2m}} \le P(m,t) \le \frac{t^2}{2m}$$

The upper bound is contained in the proof of the our last proposition, where $\Pr[\mathsf{B}] = P(2^\ell, q)$. The lower bound is stated without proof.

From the lower bound above $P(m,t)$ one concludes that the attack we described in section 5.3, when the adversary asks q queries and looks for a repeated block, provides the adversary an advantage of

$$\begin{aligned} P(2^\ell, q) &\ge 1 - e^{-\frac{q^2-q}{2 2^\ell}} \\ &\ge (1-1/e)/2 \cdot (q^2-q)/2^\ell \\ &\ge 0.316 \left(\frac{q}{2^{\ell/2}}\right)^2 \end{aligned}$$

(A technical restriction should accompany the last line.) The second line is by the inequality $1 - e^{-x} \le x$, which you can see from the Taylor expansion of e^{-x}. All together, what we've shown amounts to saying that the PRP(counter) generator is good for generating $O(2^{\ell/2})$ blocks— no more, and no less.

The last 15 minutes or so of the lecture we switched gears, starting to talk about some complexity theory— defining BPP and describing error reduction for this class of languages. This will be picked up in the notes for Lecture 6.