The University of North Carolina at Greensboro
CSC 589: Trusted Computing and Security Models
Prof. Stephen R. Tate

Handout 4
February 23, 2008

Change in Assignment 2

The "graduate student only" problem given in Assignment 2 was not a good choice. You can still work on that as a "challenge problem," but the following question is better, and should be substituted for the original problem.

5. In section 3.4.3 it was shown that PRP-CCA security implies PRP-CPA security. In this question, you are to consider the converse: if a family of permutations is PRP-CPA secure, is it necessarily PRP-CCA secure?

   To answer this, consider a family of permutations $F : \{0,1\}^k \times \{0,1\}^k \rightarrow \{0,1\}^k$ (note that $\ell = k$) that is secure — meaning that any adversary that is reasonably bounded on resources has "low" advantage $\mathbf{Adv}_F^{\text{prp-cpa}}(A)$. We don't know, and make no assumptions about, whether F is secure in the PRP-CCA sense.

   First, use F to define a new family of permutations $G : \{0,1\}^k \times \{0,1\}^k \rightarrow \{0,1\}^k$ such that G is secure in the PRP-CPA sense (roughly as secure as F), and yet G is definitely *not* secure in the PRP-CCA sense.

   You should clearly define your function G, and then carefully and completely prove two things: that for any adversary B, $\mathbf{Adv}_G^{\text{prp-cpa}}(B)$ is not much higher than $\mathbf{Adv}_F^{\text{prp-cpa}}(A)$ for some similarly-resourced adversary A, and that there exists an adversary D such that $\mathbf{Adv}_G^{\text{prp-cca}}(D)$ is high (close to 1).

   After the formal parts, think about what this means, and write out in a few English (non-mathematical) sentences what this really means. Given the choice between two families of permutations, one of which was proven secure in the PRP-CPA sense and one of which was proven secure in the PRP-CCA sense, which would you prefer and why?

   Hint: What you want to do is make it so that the value of the key is easily extracted from a particular query to the G^{-1} oracle. For example, you could define $G_K(x)$ from $F_K(x)$ in such a way that $G_K(K) = 0^k$. Notice how this allows a CCA adversary to extract the key from a single call to the G^{-1} oracle, and yet it doesn't necessary help (assuming the rest of it is defined correctly) if all you can query is G. Defining such a G is part of the challenge here (remember that XOR is your friend), and then you need to complete the proofs.