CAS CS 548. Problem Set 1

Due 5 pm Friday, February 17, 2006, in the drop box near the CS office.

Problem 1. (Cramer-Shoup)

(a) The Cramer-Shoup encryption scheme requires "proof" that $\log_g a = \log_{\hat{g}} \hat{a}$. However, it is very straightforward for the decryptor to check that: just check that $a^w = \hat{a}$. This requires simply including w into the secret key. Suppose we modify the Cramer-Shoup scheme by adding w into the secret key and replacing step **D4** on p. 21 with checking if $a^w = \hat{a}$. (Thus, the values d, e, f are no longer used.) Show exactly where (by pointing out what game and what lemma and giving an explanation) would the Cramer-Shoup proof of security break down for such a modified scheme.

(b) Let us now focus on the question of why any consistency check at all is required. Consider now a different modification to the Cramer-Shoup scheme: simply omit the check of consistency altogether, i.e., omit steps **D3** and **D4**. Modify the security proof accordingly, omitting step **D4'** in games $\mathbf{G}_3$, $\mathbf{G}_4$ and omitting game $\mathbf{G}_5$ entirely. Where does the proof break down now?

(c) Suppose G is a DDH group of size q. Show that the usual (two-generator) DDH assumption tightly implies hardness of three-generator DDH. More precisely, suppose that any adversary running in time t can't distinguish (f, g, f^u, g^u) from (f, g, f^u, g^v) (for random $f, g \in G, u, v \in \mathbb{Z}_q$) with advantage greater than ϵ. Show that then any adversary running in time t' can't distinguish (f, g, h, f^u, g^u, h^u) from (f, g, h, f^u, g^v, h^w) (for random $f, g, h \in G, u, v, w \in \mathbb{Z}_q$) with advantage greater than ϵ'. Make this a tight reduction: t' and t, as well as ϵ' and ϵ, should differ only by small additive amounts. Hint: this is buried in the Cramer-Shoup proof.

Problem 2. (CCA2 out of CPA) Consider the following attempt at constructing an IND-CCA2 public-key encryption scheme: start with an IND-CPA public-key encryption scheme $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ with keys $(\mathrm{pk}, \mathrm{sk})$ and a *deterministic* MAC (i.e., for each m, K, there is only one valid tag). To encrypt m, generate a random K, set $c_1 = \mathrm{Enc}_{\mathrm{pk}}(m, K)$ and $c_2 = MAC_K(c_1)$. To decrypt, get m, K using $\mathrm{Dec}_{\mathrm{sk}}$ and reject if the MAC doesn't verify. Demonstrate via a counterexample that the resulting scheme is not necessarily IND-CCA2 secure.