CS-7810 Graduate Cryptography February 14, 2025

Problem Set 3

Lecturer: Daniel Wichs *Due: 2/27*

Problem 1 (CPA vs CCA Security) 15 pts

CPA secure symmetric-key encryption guarantees security even if the adversary can see encryptions of chosen plaintexts. Chosen ciphertext security (CCA) guarantees security even if the adversary can see decryptions of arbitrary ciphertexts *except* for the challenge ciphertext that encrypts the unknown message.

Formally, CCA security is defined via the following CCA game $CCASec^b$:

1. The challenger chooses a random secret key $k \leftarrow \{0,1\}^n$.
2. The adversary can make an arbitrary number of encryption queries and decryption queries. In an encryption query the adversary specifies some message m and gets back $c \leftarrow \mathsf{Enc}_k(m)$. In a decryption query the adversary specifies some ciphertext c and gets back $\mathsf{Dec}_k(c)$.
3. The adversary specifies two messages m_0, m_1 and gets back the challenge ciphertext $c^* \leftarrow \mathsf{Enc}_k(m_b)$.
4. The adversary can make an arbitrary number of encryption queries and decryption queries *except* that he cannot ask for a decryption query with c^* itself.

CCA security (sometimes also referred to as CCA-2 security) requires that the adversary cannot distinguish between $b = 0$ and $b = 1$, i.e., $CCASec^0 \approx CCASec^1$. We can also define a relaxed version called CCA-1 security, where we modify the above game so that the adversary cannot ask for any decryption queries in step 4 after seeing the challenge ciphertext.

CCA security is important since the adversary may get honest users to decrypt some ciphertexts that it chooses and to reveal their contents (or at least some partial information, such as whether the ciphertext decrypts to something meaningful or not - e.g., an error message). However, the honest users will not divulge the contents of the challenge ciphertext c^*, which is what the adversary wants to learn. Alternately, CCA-1 security can model a scenario where an adversary can get temporary access to the user's device and can use it to decrypt ciphertexts of its choice, but once the adversary loses this access, any ciphertext c^* sent afterwards should remain secure.

Show the following:

- Show how to combine CPA-secure encryption and MACs to get CCA-2 secure encryption and prove the security of the construction.
- Show that assuming the existence of any of the standard building blocks (e.g., PRGs, PRFs, CPA-secure encryption, MACs), there exists a CPA-secure scheme that is not CCA-1 secure.
- Show that assuming the existence of any of the standard building blocks (e.g., PRGs, PRFs, CPA-secure encryption, MACs), there exists a CCA-1 secure scheme that is not CCA-2 secure.

Hint: for parts 2,3 make your examples as simple as possible. Think of highly contrived schemes that do something very silly just to make the example work. Don't try to construct complicated "realistic" schemes with the above properties.

Problem 2 (Do Hard-Core Bits imply One-Wayness?) 10 pts

Let $f : \{0,1\}^n \to \{0,1\}^n$ be an efficiently computable function (not necessarily a one-way function). Assume that some efficiently computable function $h : \{0,1\}^n \to \{0,1\}$ is a hard-core predicate for f, meaning that for all PPT attackers $\mathcal{A}$ we have

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathcal{A}(1^n, f(x)) = h(x)] \leq \frac{1}{2} + \mathsf{negl}(n).$$

- Show that if f is a permutation, then the above implies that f is necessarily a *one-way* permutation.
- Show that when f is an arbitrary function, then the above does *not* imply that f is necessarily a one-way function. In other words, give an example of a function f that is not one-way but still f has a hard-core predicate h. (Do not rely on any cryptographic assumptions for this part).

Problem 3 (CRHF or Not) 10 pts

Let $\{H_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{n \in \mathbb{N}, s \in \{0,1\}^n}$ be a collision resistant hash function (CRHF) that compresses $2n$ bits to n bits. For each of the following either show that it is also a CRHF or give a counter-example.

- $H'_s(x)$ outputs the first $n-1$ bits of $H_s(x)$.
- $H'_s(x_1, x_2) = H_s(H_s(x_1), H_s(x_2))$ where $x_1, x_2 \in \{0,1\}^{2n}$.
- $H'_s(x) = H_s(G(x))$ where $x \in \{0,1\}^{n+1}$ and $G : \{0,1\}^{n+1} \to \{0,1\}^{2n}$ is a PRG.

Problem 4 (Are CHRHFs also OWFs?) 10 pts

Let $\{H_s : \{0,1\}^{2n} \to \{0,1\}^n\}_{n \in \mathbb{N}, s \in \{0,1\}^n}$ be a collision resistant hash function (CRHF) that compresses $2n$ bits to n bits. Show that $f(s, x) = (s, H_s(x))$ is a OWF.

Optional: Show that this may not hold if $\{H_s : \{0,1\}^{n+1} \to \{0,1\}^n\}_{n \in \mathbb{N}, s \in \{0,1\}^n}$ only compresses $n+1$ bits to n bits.

Problem 5 (Combiners) 10 pts

- Suppose you have two candidate one-way functions f and f'. You are told that at least one of them is secure but you don't know which. Show how to combine them to get a function f^* which is guaranteed to be one-way.

- Same question for two candidate PRFs F, F'. Show how to construct F^* which is guaranteed to be a PRF if at least one of F, F' is.
- Same question for CPA secure encryption schemes $(\mathsf{Enc}, \mathsf{Dec})$ and $(\mathsf{Enc}', \mathsf{Dec}')$.
- Same question for CRHFs H, H'.

Problem 6 (Amplifying a Discrete Log Attack) 10 pts

Fix some cyclic group $\mathbb{G}$ of order q with generator g.

Suppose that there exists a randomized algorithm $\mathcal{A}$ that runs in time T and solves the discrete logarithm problem *on average* with probability .01 meaning that

$$\Pr[\mathcal{A}(g^x) = x \ : \ x \leftarrow \mathbb{Z}_q] \geq .01$$

Note that this probability is over a random x and the randomness of of $\mathcal{A}$.

Show that, if this is the case, then there is also a randomized algorithm $\mathcal{B}$ that runs in time $O(T)$ and solves the discrete logarithm problem *in the worst* case with probability .99, meaning that for *every* $x \in \mathbb{Z}_q$ we have

$$\Pr[\mathcal{B}(g^x) = x] \geq .99.$$

(For this problem, assume all group operations can be performed in $O(1)$ time.)

Note that $\mathcal{B}$ has to improve on $\mathcal{A}$ in two ways: it needs to work for every worst-case x rather than just on average, and its success probability is .99 rather than .01. Hint: Try to solve each of these issues separately.