Solution to Homework 2

1 Long Answer: A Simpler Block Cipher Based MAC

Lemma 1. *Let $E : \mathcal{K}_E \times \mathcal{Y} \to \mathcal{Z}$ be a (t, q, ϵ_E)-PRF, and $F : \mathcal{K}_F \times \mathcal{X} \to \mathcal{K}_E$ be a (t, q, ϵ_F)-PRF. Define $G : \mathcal{K}_F \times (\mathcal{X} \times \mathcal{Y}) \to \mathcal{Z}$ by $G_k(x, y) = E_{F_k(x)}(y)$. Then G is a $(t - O(q), q, q\epsilon_E + \epsilon_F)$-PRF.*

Proof. Let A be any adversary that attempts to distinguish G_k from a random function $R : (\mathcal{X} \times \mathcal{Y}) \to \mathcal{Z}$ using at most $t - O(q)$ time and q queries. Assume without loss of generality that A does not make any repeated queries. Let $S : \mathcal{X} \to \mathcal{K}_E$ be a random function, and define $G'_S(x, y) = E_{S(x)}(y)$. Note that $\Pr[A^{G_k} = 1] - \Pr[A^{G'_S} = 1] \le \epsilon_F$. (Why? Defining $B^f = A^{E_{f(\cdot)}(\cdot)}$, we find that $\Pr[A^{G_k} = 1] = \Pr[B^{F_k} = 1]$ and $\Pr[A^{G'_S} = 1] = \Pr[B^S = 1]$, so the difference of these two quantities is $\operatorname{Adv} B$, which is $\le \epsilon_F$ by the PRF-security of F.) All that remains is to show that $\Pr[A^{G'_S} = 1] - \Pr[A^R = 1] \le q\epsilon_E$.

To show this, we'll use a hybrid argument. Define the oracle $\mathcal{O}[f_1, \dots, f_q]$ (parametrized by functions $f_1, \dots, f_q : \mathcal{Y} \to \mathcal{Z}$) as follows:

Initialize:
1. Set $\mathrm{ctr} = 0$.
2. Initialize $\alpha(x)$ to be undefined for all $x \in \mathcal{X}$.

On query (x, y):
1. If $\alpha(x)$ is undefined, set $\mathrm{ctr} = \mathrm{ctr} + 1$ and $\alpha(x) = \mathrm{ctr}$.
2. Return $f_{\alpha(x)}(y)$.

Note: if we randomly draw $k_1, \dots, k_q \xleftarrow{\$} \mathcal{K}_E$, then $\mathcal{O}[E_{k_1}, \dots, E_{k_q}]$ behaves exactly the same as G'_S. Also, if we randomly select q independent random functions $T_1, \dots, T_q : \mathcal{Y} \to \mathcal{Z}$, then $\mathcal{O}[T_1, \dots, T_q]$ behaves exactly the same as R (for adversaries who do not make any repeated queries). Thus all that remains is to show that $\Pr[A^{\mathcal{O}[E_{k_1},\dots,E_{k_q}]} = 1] - \Pr[A^{\mathcal{O}[T_1,\dots,T_q]} = 1] \le q\epsilon_E$.

Fix any i, and consider the oracle $\mathcal{O}^*_i[f] = \mathcal{O}[E_{k_1}, \dots, E_{k_{i-1}}, f, T_{i+1}, \dots, T_q]$. Notice that $\Pr[A^{\mathcal{O}^*_i[E_{k_i}]} = 1] - \Pr[A^{\mathcal{O}^*_i[T_i]} = 1] \le \epsilon_E$. (Why? Defining $B^f = A^{\mathcal{O}^*_i[f]}$, we find that $\Pr[A^{\mathcal{O}^*_i[E_{k_i}]} = 1] - \Pr[A^{\mathcal{O}^*_i[T_i]} = 1] = \operatorname{Adv} B \le \epsilon_E$, by the PRF-security of E.) Finally, by summing over $i = 1, \dots, q$ and applying the triangle inequality, we see that $\Pr[A^{\mathcal{O}[E_{k_1},\dots,E_{k_q}]} = 1] - \Pr[A^{\mathcal{O}[T_1,\dots,T_q]} = 1] = \sum_i (\Pr[A^{\mathcal{O}^*_i[E_{k_i}]} = 1] - \Pr[A^{\mathcal{O}^*_i[T_i]} = 1]) \le n\epsilon_E$. This completes the proof of the lemma. □

Theorem 1. *Let $E : \{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{128}$ be a (t, q, ϵ)-PRF. Fix $n > 0$. Define $F^{(n)} : \{0,1\}^{128} \times \{0,1\}^{128n} \to \{0,1\}^{128}$ by $F_K^{(n)}(M) = X_n$, where $X_0 = K$, $X_i = E_{X_{i-1}}(M_i)$, and $M = (M_1, \dots, M_n)$. Then F is a $(t - O(q), q, nq\epsilon)$-PRF.*

Proof. This is a straightforward induction on n. For $n = 1$, the result is trivial (since $\epsilon \le q\epsilon$). Also, $F^{(n+1)}$ satisfies the identity

$$F_K^{(n+1)}(M_1, \dots, M_n, M_{n+1}) = E_{F_K^{(n)}(M_1,\dots,M_n)}(M_{n+1}),$$

so the inductive step follows by Lemma 1. □

Some comments:

- You may recognize this scheme as a generalization of the GGM construction. The GGM construction built a tree of branching factor 2. You can consider this construction as an enormous tree of branching factor 2^{128} and depth n, and where the function is only evaluated at some subset of q leaves of the entire tree.

- This scheme is only secure when used with fixed-length messages. For variable-length messages, it is totally insecure.

- Many people struggled with this question. Very few people actually wrote out a full reduction, and if you didn't write out a reduction, you were almost sure to go astray, because there are many seemingly reasonable hybrid strategies you can try that don't actually work (because they cannot be justified by any reduction).

- Three simple attacks were enough to expose gaps in the reasoning in most of the solutions:

  Attack #1: query message $M = (0, 0)$; query message $M' = (0, 1)$
  Attack #2: query message $M = (0, 0)$; query message $M' = (1, 0)$
  Attack #3: query message $M = (0, 0)$; query message $M' = (1, 1)$

- One common mistake was to claim that "since E is a PRF, we can replace E with a random function $R : \mathcal{K}_E \times \mathcal{Y} \to \mathcal{Z}$". However, this typically wasn't justified by any proof, and is just wrong.

  What is sometimes a permissible tweak is to replace E_k with a random function $R : \mathcal{Y} \to \mathcal{Z}$. This is only justified if k is a random key that is not used anywhere else and independent of everything else; and even then, a reduction is still needed to justify such a replacement, because the replacement is not always allowed.

- Some people tried to argue what is the worst case (e.g., where all M_1 values are distinct, or something) and then evaluate its security under that assumption. However, without any proof that this is indeed the worst case, that's just hand-waving.

- In general, few people provided full proofs. It was very common to just make claims about indistinguishability of some pair of oracles, without providing any justification (and typically at least one of these claims was wrong).

- Some answers tried to argue that X_i is indistinguishable from uniform, e.g., by using induction and suggesting that if X_{i-1} is indistinguishable from uniform, then so is $E_{X_{i-1}}(M_i)$. This argument is fine if M is fixed in advance, so it would be a reasonable proof for adversaries who only apply F on one message ($q = 1$). However, for $q > 1$, it's not clear what statements about X_i even mean, because each message M yields a different value X_i; and in any case, the attempted proof strategy fails, because M_i is not fixed.

- I saw some occasional confusion between *events* and *propositions*. A proposition, like $1+1 = 2$, is either true or false. An event, like $X_i = X_j$ (where X_i, X_j are random variables), happens with some probability. It doesn't make sense to say something like "if we assume that $X_i = X_j$ never holds for any message, then $\mathrm{Adv}\, A \leq \epsilon$". If this was an attempt to reason about conditional probabilities, then it should have been explained in those terms.

2 Long Answer: Strong Encryption with a Weak Cipher

If we define $G : \{0,1\}^{128} \to \{0,1\}^{256}$ by $G(K) = (\mathrm{AES}_K(0), \mathrm{AES}_K(1)$, then it is easy to see that G is a $(t, t/2^{127})$-PRG. Also, for any fixed $n > 0$ we can apply the GGM construction to obtain a PRF $F^{(n,\ell)} : \{0,1\}^{128} \times \{0,1\}^n \to \{0,1\}^{128\cdot\ell}$; this will be a $(t, q, (n + \lceil \lg \ell \rceil)\ell qt/2^{128})$-PRF. As one example, by taking $n = 128$, $\ell = 1$, we obtain a 128-bit block cipher that can be used with standard modes (such as CTRC and CBC-MAC in EtA mode).

Here is another example that builds on the above primitives, worked out in more detail. Let ℓ denote the maximum length of any message x, in blocks. Encrypt x to $c = E^*_{K_e}(x) = (\mathrm{ctr}{+}{+}, F^{(128,\ell)}_{K_e}(\mathrm{ctr}))$; this is an $(t, q, (128 + \lg \ell)\ell qt/2^{128})$-IND-CPA encryption scheme. Also the ciphertext will have length at most $\ell + 1$ blocks. Now define $(K_e, K_a) = G(K)$ and $E_K(x) = (c, F^{(\ell+1,1)})_{K_a}(c)$, where $c = E^*_{K_e}(x)$. Since $F^{(\ell+1,1)})_{K_a}$ is a $(t, q, (\ell + 1)qt/2^{128})$-MAC, E is $(t, q_s, q_v, (\ell{+}1)(q_s{+}q_v)t/2^{128})$-INT-CTXT and thus also IND-CCA2. This scheme requires $129(\ell{+}1)$ AES operations to encrypt an ℓ-block message.

It is possible to build faster schemes. For instance, we can replace the MAC above with $F^{(128,1)}_{K_e}(H_{K_h}(c))$, where H is a fast 2-universal hash function, and the result will be significantly faster for large messages. Or, one can use $E_K(x) = E^*_{K_e}(x || H_{K_a}(x))$, although this takes some proof.

A few comments:

- There was a lot of opportunity for creativity, and several folks came up with clever and interesting schemes. Many of you correctly noticed that the key challenge here is to build a scheme that only uses any given AES key at most twice. (Indeed, if you use the same AES key more than twice, then all bets are off, and there's probably no hope for your scheme to be secure.)

- However, while some people came up with a secure scheme, almost everyone struggled to *prove* their scheme secure. Handwaving and erroneous arguments were widespread.

- If you use the Encrypt-then-Authenticate (EtA) paradigm, you have to make sure that the encryption key is *independent* of the authentication key. For instance, if you use the same key for encryption as for authentication, then there are no guarantees of security. Many people missed this. You would have discovered the need for independent keys if you had tried to write down the reduction. However, because few people bothered to write down reductions, they failed to detect this error.

- Some people tried using a standard MAC, such as CBC-MAC, inside their scheme. However, they often re-used intermediate values from the CBC-MAC elsewhere in their scheme, or re-used the same key elsewhere. When you do that, you completely void the security warranty of CBC-MAC: you are not using it as a black box (as a subroutine), but you are "peeking under the covers." For instance, if you re-use intermediate values, then those intermediate values might get exposed, and the standard proof of security for CBC-MAC is no longer relevant (for all you know, maybe knowing those intermediate values might help the attacker—and in fact, it does, in some cases).

  You would have discovered the problems with re-using intermediate values and keys if you had tried to write down the reduction to breaking CBC-MAC carefully. However, because few people bothered to write down reductions, few noticed this error.

3 Short Answer: A CBC-MAC Variant

(a) If $F : \mathcal{K} \times \mathcal{M} \to \mathcal{S}$ is a (t, q, ϵ)-PRF, then $H : \mathcal{K} \times (\mathcal{M} \times \mathcal{S}) \to \mathcal{S}$ defined by $H_k(x, y) = F_k(x) \oplus y$ is $\boxed{\epsilon + \frac{1}{|\mathcal{S}|}}$-weakly 2-universal. For a random function R, $\Pr[R(x) + y = R(x') + y'] \leq \frac{1}{|\mathcal{S}|}$ for all x, y, x', y' with $(x, y) \neq (x', y')$.

(b) If H is ϵ-weakly 2-universal and F is a (t, q, ϵ')-PRF, then F' defined by $F_{k,k'}(x) = F_k(H_{k'}(x))$ is a $\boxed{(t - O(q), q, \epsilon' + \binom{q}{2}\epsilon)}$-PRF. The probability that some pair of distinct queries x, x' lead to a collision at the input to $F_k(\cdot)$ is at most $\binom{q}{2}\epsilon$; conditioned on this not happening, the distinguishing advantage is no worse than ϵ'.

(c) Fix $n > 0$. Define the function $F : \{0,1\}^{128n} \times \{0,1\}^{128n} \to \{0,1\}^{128}$ by $F_K(M) = X_n$, where $X_0 = 0$, $X_i = \text{AES}_{K_i}(X_{i-1} \oplus M_i)$, $K = (K_1, \ldots, K_n)$, and $M = (M_1, \ldots, M_n)$. If AES is a (t, q, ϵ)-PRF, then F is a $\boxed{(t - O(q), q, qn\epsilon + n\binom{q}{2}\frac{1}{2^{128}})}$-PRF. F is $qn\epsilon$ indistinguishable from the function $G(M) = Y_n$ given by $Y_0 = 0$, $Y_i = R_i(Y_{i-1} \oplus M_i)$, where $Y_1, \ldots, Y_n$ are n independent random functions. If $M^1, \ldots, M^q$ are the q queries that A makes, set the bad flag to true if there exists i, j, ℓ such that $(M_1^i, \ldots, M_\ell^i) \neq (M_1^j, \ldots, M_\ell^j)$ but $Y_\ell^{i-1} \oplus M_\ell^i = Y_\ell^{j-1} \oplus M_\ell^j$. So long as the bad event does not occur, the output from each $G(M^i)$ is uniform and independent.

(d) One possibility: define $K_i = \text{AES}_K(i)$, where K is the single master key, and then use (c) unchanged. This only increases the insecurity level of the scheme by at most $n\epsilon$, since the function $K \mapsto (K_1, \ldots, K_n)$ is a $(t, n\epsilon)$-PRG.

Comments:

- The assumption that AES is a PRF does not ensure that $\text{AES}_K(K)$ is indistinguishable from a uniform random 128-bit string. It is entirely possible to have a PRF F where $F_K(K) = 0$ for all K, for instance. (It is even possible for F to be a PRP and have this be true.)

- Some people tried using a sequence of keys $K_1, \ldots, K_n$ that were not indistinguishable from uniform: e.g., $K_1 = K$, $K_{i+1} = \text{AES}_{K_i}(0)$. However, there is no reason to think that scheme (c) will remain secure when this choice of $K_1, \ldots, K_n$ is plugged in. (Indeed, the scheme might be insecure; consider, say, $n = 2$ and the message $M = (0, 0)$; then the output takes the form $\text{AES}_{K_2}(K_2)$.)

4 Short Answer: PRGs

For each of the following statements, say whether the claim is true or false. If true, provide a one- or two-line justification of the key idea behind a proof of the statement. If false, provide a one- or two-line sketch of a counterexample.

(a) $\boxed{\text{False}}$. Suppose $G(w, x) = (G'(w), x)$, where $G' : \{0,1\}^n \to \{0,1\}^m$ is a PRG. Then G is a PRG, but H is not a PRG, since outputs of H have the form (u, v, w, v).

(b) $\boxed{\text{False}}$. Consider the same G given above. Then $F_k(x)$ has the form (u, x), which would not be true of a random function.

Comments:

- Remember, a PRG is a fixed *deterministic* function G.
- Some folks talked about "queries" to the PRG. However, the definition of security for a PRG does not allow the adversary the power to make any queries to the PRG; instead, a random seed is chosen by the referee, and the adversary sees the output of the PRG evaluated at that random seed. In particular, the adversary has no control over the input to the PRG, in the game that defines the notion of security for a PRG. One consequence is that if you encounter some other situation where the adversary does have control over the input to the PRG, then there is no a priori reason to think that the game used in the definition of PRG security has any relevance to your situation.
- I was looking for a sketch of a counter-example. Some people tried to argue that there is no reason to think that the definition of a PRG would imply these claims. While that may have been useful intuition, it doesn't prove anything, because it has too much the flavor of "proof by lack of imagination": ok, so *you* can't think of any reason why the implication would be true, but how do you know someone else can't think of a reason why the implication has to be true?

5 Short Answer: Reaction Attacks

(a) $\boxed{\text{False}}$. Let E^* be IND-CPA secure, and define $E_k(x) = (0, 0, E^*_k(x))$ and

$$D_k(a, b, c) = \begin{cases} D^*_k(c) & \text{if } a = 0 \\ 1 & \text{if } a = 1 \text{ and } D^*_k(c) = b \\ 0 & \text{otherwise.} \end{cases}$$

Then E^* is IND-CPA secure, but insecure against reaction attacks.

(b) $\boxed{\text{True}}$. Given a D_k oracle, you can easily simulate a V_k oracle, so any IND-CCA2 attack can easily be converted into a reaction attack.

Comments:

- It is not true that IND-CCA2 security implies INT-CTXT security. It is not true that INT-CTXT security is necessary for security against reaction attacks.