Crypto Engineering
Hash functions & MACs

2020-10-02

Exercise 1: Meet-in-the-middle preimage attack on BRSS/PGV-13 + MD

BRSS/PGV-13 is an alternative to Davies-Meyer, defined as $f(h, m) = \mathcal{E}(m, h) \oplus c$ for a cipher $\mathcal{E}$ and with c a constant. It can be shown in the ideal cipher model that a Merkle-Damgård function with such a compression function is secure up to the birthday bound for both collision *and* preimage attacks (Black & al., 2010).

Q. 1 If $\mathcal{E}$ is ideal, what is the cost, given h and t, of finding m such that $f(h, m) = t$? Conclude about the preimage security of f itself.

A *meet-in-the-middle* preimage attack on a function $H_{x,y} = F_x \circ G_y$ aims at finding x and y s.t. $H_{x,y}(\mathrm{IV}) = t$, where t is a given target. It works by splitting the computation of H into *forward computations* $G_{y_i}(\mathrm{IV})$ and *backward computations* $F_{x_1}^{-1}(t)$ for many candidate values x_i, y_i.

Q. 2 We assume that $F_x, G_y, H_{x,y}$ all behave as random functions and have signature $\{0,1\}^n \to \{0,1\}^n$.

1. What is the probability over y that $G_y(\mathrm{IV}) = \alpha \in \{0,1\}^n$? Does this probability depend on α?
2. What is the probability over y that $G_y(\mathrm{IV}) \in \mathcal{S} \subseteq \{0,1\}^n$, $\#\mathcal{S} = q$?
3. How many candidate values x_i and y_i should (roughly) be selected to minimize the time cost of the attack?
4. What is the total time and memory cost of the attack (assuming that you can use a data structure with constant access time)?

Q. 3 Show how to compute a two-block preimage for $\mathcal{H}$ with the above compression function, using a meet-in-the-middle attack.

Q. 3 Give a rough explanation of how the attack of the previous question is prevented when using a Davies-Meyer compression function.

Exercise 2: `SuffixMAC`

Let $\mathcal{H} = \{0,1\}^* \to \{0,1\}^n$ be a (usual, narrow-pipe) Merkle-Damgård hash function. We define `SuffixMAC` : $\{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^n$ associated with $\mathcal{H}$ as `SuffixMAC`$(k, m) = \mathcal{H}(m||k)$.

Q. 1

1. What is the generic average complexity of finding a collision (m, m') for $\mathcal{H}$?
2. Does this complexity change if one requires m and m' to be of the same length $\ell > n$?

Q. 2 Let (m, m') be a colliding pair for $\mathcal{H}$ where m and m' have the same length.

1. Give an existential forgery attack for `SuffixMAC` with query cost 1.
2. What is the total cost of this attack if one has to compute (m, m')?
3. Is this attack "meaningful" if $\kappa < n/2$? What if $\kappa = n$?

Q. 3 What comments can you make about instantiating `SuffixMAC` in the following ways:

1. $\mathcal{H}$ is taken to be SHA-256, $\kappa = 256$?
2. $\mathcal{H}$ is taken to be SHA-512, $\kappa = 256$?
3. $\mathcal{H}$ is taken to be SHA-512/256, $\kappa = 256$?

Exercise 3: Raw `CBC-MAC`

Let $\texttt{CBC-ENC}(k, \text{IV}, m)$ denote CBC encryption of the message m and initial value IV with a block cipher $\mathcal{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$. We define $\texttt{CBC-MAC}(k, m)$ as the last output block of $\texttt{CBC-ENC}(k, 0^n, m)$.

Q. 1 Does the fact that `CBC-MAC` uses a constant IV 0^n in its call to `CBC-ENC` result in a security problem?

Q. 2 In this question, for the sake of simplicity, we assume that no padding is used by `CBC-ENC`.

Let $m_1 \in \{0,1\}^n$ denote a one-block message.

1. Give an explicit expression for $\tau_1 := \texttt{CBC-MAC}(k, m_1)$
2. Give an explicit expression for $\tau_2 := \texttt{CBC-MAC}(k, m_1 || (m_1 \oplus \tau_1))$
3. Deduce an existential forgery attack on `CBC-MAC`. What is its query and time cost?

Q. 3 We now define $\texttt{CBC-MAC}'$ as $\texttt{CBC-MAC}'(k, m) = \mathcal{E}(k', \texttt{CBC-MAC}(k, m))$, where k' is a key independent from k.

Explain (roughly) why this additional processing prevents the above attack.