Crypto Engineering (GBX9SY03)
TD Hash functions

2017-10-18

Exercise 1: Multicollisions for Merkle-Damgård hash functions

In 2004, Joux showed a simple attack illustrating the fact that Merkle-Damgård hash functions were not "ideal". This attack consists in computing a collision on many (more than two) messages, i.e. finding m_0, m_1, …, m_q that all have the same hash, more efficiently than what is possible for a random oracle.
In this exercice, we can safely ignore padding issues.

Q. 1: We assume that the expected number of collisions in the elements of two lists L_0 and L_1 of random n-bit elements is $\approx \#L_0 \times \#L_1/2^n$. Let $\mathscr{H} : \{0,1\}^* \to \{0,1\}^n$ be a random oracle; what is the expected complexity of finding an r-collision for $\mathscr{H}$?
Hint: try to find the optimal balance in the list sizes for the case $r = 3$, and generalize the formula.

Q. 2: Recall the structure of a Merkle-Damgård hash function $\mathscr{H} : \{0,1\}^* \to \{0,1\}^n$ based on a compression function $\mathsf{f} : \{0,1\}^n \times \{0,1\}^b \to \{0,1\}^n$. Let m, m' be two messages such that $|m| = |m'| = b$. Explicit the relation between $\mathscr{H}(m)$, $\mathscr{H}(m||m')$ and f.

Q. 3 Take $\mathscr{H}$ as in Q. 2. Let m_0 and m'_0 be two one-block messages colliding through $\mathscr{H}$ (i.e. $h_0 := \mathscr{H}(m_0) = \mathsf{f}(\mathrm{IV}, m_0) = \mathscr{H}(m'_0) = \mathsf{f}(\mathrm{IV}, m'_0)$). Assuming f is ideal, how efficiently can you compute a collision (m_1, m'_1) for $\mathsf{f}(h_0, \cdot)$? Once you know such a collision, how many messages colliding with $\mathscr{H}(m_0||m_1)$ can you easily (i.e. in constant time) create? Conclude about the cost of computing a 2^r-collision for $\mathscr{H}$ and why Merkle-Damgård hash functions are not ideal.

Q. 4 The *concatenation combiner* is a simple construction taking two hash functions $\mathscr{H}_1$ and $\mathscr{H}_2$ and defined as $\mathrm{CAT}_{\mathscr{H}_1,\mathscr{H}_2}(m) := \mathscr{H}_1(m)||\mathscr{H}_2(m)$. Assuming $\mathscr{H}_1$ and $\mathscr{H}_2$ have an output size of n bits and follow the Merkle-Damgård construction, how efficiently can you compute a collision for $\mathrm{CAT}_{\mathscr{H}_1,\mathscr{H}_2}$? Is it possible to significantly improve the collision-resistance of SHA-1 by using $\mathrm{CAT}_{\text{SHA-1,MD5}}$?

Exercise 2: Davies-Meyer fixed-points

In this exercise, we will see one reason why *Merkle-Damgård strengthening* (adding the length of a message in its padding) is necessary in some practical constructions.

Q. 1 Recall the "Davies-Meyer" construction of a compression function f from a block cipher $\mathscr{E}$.

Q. 2 Considering the feed-forward structure of Davies-Meyer, under what conditions would you obtain a fixed-point for such a compression function?

Q. 3 Show how to compute the (unique) fixed-point of $\mathsf{f}(\cdot, m)$ for a fixed m. Given h, is it easy to find m such that it is a fixed-point, if $\mathscr{E}$ is an ideal block cipher?

Q. 4 A *semi-freestart collision attack* for a Merkle-Damgård hash function $\mathscr{H}$ is a triple (h, m, m') s.t. $\mathscr{H}_h(m) = \mathscr{H}_h(m')$, where $\mathscr{H}_h$ denotes the function $\mathscr{H}$ with its original IV replaced by h. Show how to use a fixed-point to efficiently mount such an attack for Davies-Meyer + Merkle-Damgård, when strengthening is not used.

Note: Fixed-points of the compression function can be useful to create the *expandable messages* used in second preimage attacks on Merkle-Damgård.

Exercise 3: Meet-in-the-middle preimage attack on BRSS/PGV-13 + MD

BRSS/PGV-13 is an alternative to Davies-Meyer, defined as $\mathsf{f}(h, m) = \mathscr{E}(m, h) \oplus c$ for a cipher $\mathscr{E}$ 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 $\mathscr{E}$ is ideal, what is the complexity, given h and t, of finding m such that $\mathsf{f}(h, m) = t$? Conclude about the preimage security of f itself.

Q. 2 Show how to compute a two-block preimage for $\mathscr{H}$ with the above compression function, using a meet-in-the-middle attack, and roughly evaluate its complexity (both time and memory).

Q. 3 Give a rough explanation of how the attack of Q. 2 is prevented when using a Davies-Meyer compression function.

Exercise 4: Hash-based message-authentication codes

Q. 1 Recall the definition of a message-authentication code (MAC), existential forgery, and universal forgery.

Q. 2 Let us first assume that $\mathscr{H}$ is a random oracle. Explain (roughly) why the "prefix-MAC" construction $\mathrm{PM}_{\mathscr{H}}(k, m) := \mathscr{H}(k||m)$ is secure? Is there a difference with the "suffix-MAC" variant $\mathrm{SM}_{\mathscr{H}}(k, m) := \mathscr{H}(m||k)$?

Q. 3 Now assume that $\mathscr{H}$ is a Merkle-Damgård hash function. Suppose I know m and its tag $t := \mathrm{PM}_{\mathscr{H}}(k, m)$, and that the size of k is known. How easily can I compute another message and its corresponding tag under $\mathrm{PM}_{\mathscr{H}}(k, \cdot)$? Is this MAC secure against existential forgery?

Q. 4 Still assuming that $\mathscr{H}$ is a Merkle-Damgård function, show how collisions on $\mathscr{H}$ lead to an existential forgery attack of $\mathrm{SM}_{\mathscr{H}}$. What is the expected complexity of this attack for an otherwise secure $\mathscr{H}$? Is this better than what you would expect for a "good" MAC?

Q. 5 Is it reasonable to instantiate prefix/suffix-MAC with SHA-3? With SHA-256? With SHA-512/256?

Note: It can be proven (Yasuda, 2007) that, using appropriate padding rules, the "Sandwich-MAC" construction $\mathrm{SANDWICH}_{\mathscr{H}}(k, m) := \mathscr{H}(k||p||m||p'||k)$ (where p and p' denote padding) is secure, without requiring $\mathscr{H}$ to be a random oracle (in particular, it can be built with a Merkle-Damgård construction).