Introduction to Cryptography: Assignment 6

Group number 57

Elwin Tamminga s1013846

Lucas van der Laan s1047485

1

(a) Given $h(m)$, then we can find a preimage $m_0 \in \{0,1\}^n$ by hashing $h(m)$ which gives us $h(h(m)) = H(m)$. Then we can use the algorithm A on $H(m)$ to find the preimage m_0 with effort N and probability p. So the total effort is $\leq N + 1$.

(b) The chance of finding a preimage by hashing it twice is a bit higher than hashing it once, because if there are two different messages with the same hash, then hashing it twice will also return the same hash. But it is also possible to find two different messages that returns a different hash, but when hashed again returns the same hash.

(c) The security strength with respect to collision resistance can be calculated with $\log_2(2^{(N/2)})$.
So $N_1 = 256$ for h_1 because the security strength is $\log_2(2^{(256/2)}) = 128$
And $N_2 = 192$ for h_2 because the security strength is $\log_2(2^{(192/2)}) = 96$
The total security strength of H can be calculated with $\log_2(2^{(N_1/2)} + 2^{(N_2/2)})$.
This leads to the equation: $\log_2(2^{(256/2)} + 2^{(192/2)}) = 128$

(d) To calculate $H_2(m)$ for a message m, we first need to calculate $h_2(m)$ and use the output of this hash function in h_1. So the security strength of H_2 with respect to collision resistance is $\min(s_1, s_2)$ with s_1 being the security strength of h_1 and s_2 is the security strength of h_2.

We can show this by choosing a $h_2 = \{0,1\}^* \longrightarrow \{0,1\}^1$
This would lead to $h_1 = \{0,1\}^1 \longrightarrow \{0,1\}^{n_1}$

The upper bound security strength with respect to collision resistance of h_2 is $l/2 = \frac{1}{2}$ bits. Because the output of this hash function maps with a one-to-one relation to the output of h_2 in H_2, the upper bound security strength of H_2 is also $\frac{1}{2}$ bits.

2

(a) With a key of length 56 the probability of finding the key in a single guess is $\frac{1}{2^{|k|}} = \frac{1}{2^{56}}$.

(b) $N = 4$ (2 SHA-256 queries, 2 DES queries)
$M = 2$ (2 MAC verification queries)

$s = \log_2(\frac{M+N}{P}) = \log_2(6 * 2^{56}) \approx 58$ bits

But the security strength against exhaustive key search is upper-bounded by the key length. So the security strength is $|k| = 56$ bits.

(c) The probability an attacker guesses a tag correctly is $\frac{1}{2^{48}}$.
$N = 0$ (no offline queries)
$M = 1$ (1 MAC verification query)
$s = \log_2(\frac{M+N}{P}) = \log_2(1 * 2^{48}) = 48$ bits.

(d)
- Do an online query to the MAC generation function with a message m which gives the tag T.
- Do an offline query to SHA-256 with message m and truncate the output to 64 bits, which we will call p.
- Find a collision on the truncated SHA-256 digest such that $p' = p$, which is generated by doing offline queries with m'.
- Now we have a m' with a valid tag T' which is equal to T.

(e) The security strength is based on the birthday attack, which has a security strength formula of $\log_2(2^{n/2})$
We can do this, because we are trying to find a collision by finding two different messages resulting in the same hash.
This would result in a security strength of $\log_2(2^{n/2}) = \log_2(2^{64/2}) = 32$ bits.
We then take the minimum of the answer b, c, and e, which results in a total security strength of 32 bits to a forgery attack.