Recovering Secret Keys of HMAC with Preimage-Finding Oracle: How Easy Is It?

Sergey Panasenko,
panasenko@guardant.ru

Marina Skorobogatova,
sma@aktiv-company.ru

JSC «Active soft», Moscow, Russian Federation

Preimage Oracle

Let’s assume that an adversary can obtain preimages for any hash value (calculated by a hash function H) using the preimage Oracle P :

$$\{\rho\} = \mathrm{P}_s(h),$$

where:

- *h* — a hash value for finding preimages;
- *s* — a length of preimages to be found;
- $\{\rho\} = \{p_1, ..., p_z\}$ — a collection of *z* preimages of required length found by the Oracle.

If the hash function H outputs *n*-bit values and its output values are distributed uniformly, then we can assume that on average:

$$z \approx 2^{s-n}$$

Note: when $s \leq n$, *z* can be interpreted as a probability to find a single preimage.

Hash-Based Message Authentication Codes

Hash-based message authentication codes (HMAC) *:

$$\mathrm{HMAC}(K,\ m) = \mathrm{hash}((k \oplus opad) \,\|\, \mathrm{hash}((k \oplus ipad) \,\|\, m))$$

Where:

- *m* — an input message of arbitrary length;
- *K* — a secret key for HMAC calculation, and *k* — its aligned/padded version to the block size *b* of the underlying hash function;
- *ipad* & *opad* — constants of *b*-bit length.

Input messages and corresponding HMAC values are usually known to an adversary.
If the adversary has the preimage Oracle, she can easy obtain intermediate values:

$$k \oplus opad$$

Question: Can she determine the correct value of secret key *k* (e.g. to forge any message)?

* M. Bellare, R. Canetti, H. Krawczyk. Keying Hash Functions for Message Authentication. CRYPTO 1996.

Search Technique

Conditions: the adversary has the preimage Oracle P and a known pair of message *m* (of length *len*) and its HMAC value *h* calculated with the key (aligned/padded) *k*.

Step 1: Using the Oracle the adversary obtains a collection of preimages:

$$\{c_1, \ldots, c_z\} = \mathrm{P}_s\ (h),$$

where $s = b + n$, i.e. the size of collection is $z \approx 2^{s-n} = 2^b$, and:

$$c_i = c_i^l \,||\, c_i^r, \quad i = 1,\ldots,z$$
$$c_i^l = k_i \oplus opad$$
$$c_i^r = \mathrm{hash}((k_i \oplus ipad) \,||\, m_i)$$

Step 2: To find the required key the adversary needs to check every candidate $c_i,\ i = 1,\ldots,z$ by inverting the right part of it to obtain another collection of possible candidates:

$$\{d_{i_1}, \ldots, d_{i_y}\} = \mathrm{P}_{b+len}\ (c_i^r)$$

$$d_{i_j} = d_{i_j}^l \,||\, d_{i_j}^r, \quad j = 1,\ldots,y, \quad y \approx 2^{b+len-n}$$
$$d_{i_j}^l = k_{i_j} \oplus ipad$$

Step 3: Performing the exhaustive search among values $c_i,\ i = 1,\ldots,z$ and $d_{i_j},\ j = 1,\ldots,y$ for every c_i . The criteria for determining that the valid key has been found:

$$k_{i_j} = k_i = k \text{ and } d_{i_j}^r = m$$

It’s Not So Easy

Average work (not considering storage requirements and possible false positives):

- $1 + z/2 \approx 2^{b-1}$ Oracle P calls;
- $y * z/2 \approx 2^{2b+len-n-1}$ comparisons.

Hash function	Block size (equal to the length of *k*) ***b***	Output size ***n***	Length of preimages ***s***	Oracle calls (lower bound at *len* = 1)	Comparisons (lower bound at *len* = 1)
Streebog-256	512	256	768	2^{511}	2^{768}
Streebog-512	512	512	1024	2^{511}	2^{512}

Conclusion:

HMAC over Streebog-256/512 is resistant against the key recovery attack described above (when an adversary has the preimage Oracle).

Thank you!

Questions?

Sergey Panasenko
panasenko@guardant.ru

Marina Skorobogatova
sma@aktiv-company.ru