Breaking $H^2$-MAC using Birthday Paradox

Fanbao Liu$^{1,2}$, Tao Xie$^1$ and Changxiang Shen$^2$

1 School of Computer, National University of Defense Technology, Changsha, 410073, Hunan, P. R. China
2 School of Computer, Beijing University of Technology, 100124, Beijing, P. R. China

Abstract. We propose an efficient method to break $H^2$-MAC, by using a generalized birthday attack to recover the equivalent key, under the assumption that the underlying hash function is secure (collision resistance).

Keywords: $H^2$-MAC, Equivalent Key Recovery, Birthday Paradox.

1 Introduction

In ISC 2009, Yasuda proposed $H^2$-MAC [5], a variant of HMAC, which aims to remedy the drawback of HMAC and keep its advantages and security at the same time. $H^2$-MAC is defined by removing the outer key of HMAC, which is shown as folllows,

$$H^2\text{-MAC}(K)(M) = H(H(K||pad||M))$$

where $K$ is an $n$-bit key, and $pad \in \{0, 1\}^{m-n}$ is a fixed constant.

$H^2$-MAC is proven to be a secure PRF (pseudorandom function) under the assumption that the underlying compression function is a PRF-AX [5].

In ISA 2011, Wang [3] proposed an equivalent key recovery attack to $H^2$-MAC instantiated with the broken MD5 [2, 4], with complexity about $2^{97}$ on-line MAC queries.

We break $H^2$-MAC by recovering its equivalent key through a generalized birthday attack with two groups. First, we get a lot of MAC values of $H^2$-MAC using different messages in group $G_1$, through on-line queries. Second, we directly compute many values of $H(H(C||pad||m))$, called $H^2$, in group $G_2$ through off-line, where $C$’s and $m$’s can be both randomly generated. If the on-line queries in $G_1$ is $2^{n/2}$ and the off-line computations in $G_2$ is also $2^{n/2}$, then, there is a pair $(m, m')$ that the inner hashing part of $H^2$-MAC and $H^2$ equate with great probability [1]. Therefore, the equivalent key of $H^2$-MAC can be recovered by computing the corresponding value of $H^2$.

$^1$ The secret key of $H^2$-MAC is replaced with a constant, for example, the IV of the underlying hash function.
2 Notations

Let \( H \) be a concrete hash function mapping \( \{0, 1\}^* \rightarrow \{0, 1\}^n \). Let \( IV \) be the initial chaining variable of \( H \). Let \( K \) denote a secret key with \( n \) bits. \( x||y \) denotes the concatenation of two bit strings \( x \) and \( y \). \( |G| \) denotes the number of elements of the set \( G \). \( \text{pad}(M) \) denotes the padding bits of \( M \) in Merkle-Damgård style. \( H^2 \) means that the secret key to \( H^2\text{-MAC} \) is replaced with a constant \( C \) or a known parameter to everybody, hence, \( H^2 \) can be also viewed as the double application of the underlying hash function \( H \).

3 Breaking \( H^2\text{-MAC} \) Using Birthday Paradox

We call \( I_K = H(K||\text{pad}||M) \) the inner hashing of \( H^2\text{-MAC} \), \( Oh = H(I_K) \) the outer hashing of \( H^2\text{-MAC} \), respectively.

We apply the generalized birthday attack with two groups \([1]\) to \( H^2\text{-MAC} \) and then recover its equivalent key \( K_e = H(K||\text{pad}||M_0) \).

We use 1-block messages \( M_i \)'s to generate the corresponding \( H^2\text{-MAC} \) values, and use 1-block messages \( M'_j \)'s to generate the corresponding \( H^2 \) values, where \( 1 \leq i, j \leq 2^{n/2} \). The overall strategy of equivalent key recovery attack to \( H^2\text{-MAC} \) is shown as follows.

1. Generate a group one \( G_1 \) with \( r = |G_1| = 2^{n/2} \) elements, by computing the corresponding values of \( H(H(c||M'_j)) \) for \( r \) different \( c \) and \( M'_j \)'s, which can be randomly generated. Specifically, \( c \) can be a pre-chosen constant.
2. Generate a group two \( G_2 \) with \( s = |G_2| = 2^{n/2} \) elements, by querying the corresponding values to \( H^2\text{-MAC} \) oracle with the secret key \( K \) for \( s \) different \( M_i \)'s, where \( M_i \)'s are randomly generated.
3. There is a pair \( (M_i, M'_j) \) that not only satisfies \( H^2\text{-MAC}_K(M_i) = H^2\text{-MAC}(M'_j) \), but also satisfies \( H(K||\text{pad}||M_i) = H(c||M'_j) \) (an inner collision between \( H^2 \) and \( H^2\text{-MAC} \) happens), with great probability \([1]\).
4. Since \( H(K||\text{pad}||M_i) = H(c||M'_j) \), and we know the value of \( c \) and \( M'_j \), we can compute the value of \( K_e = H(K||\text{pad}||M_i) = H(c||M'_j) \).
5. Let \( \text{pad}_0 \) and \( \text{pad}_1 \) be the padding bits of \( K||\text{pad}||M_i \) and \( K||\text{pad}||M_i||\text{pad}_0||x \), respectively, for arbitrary message \( x \). Hence, we can generate the result of \( H(K||\text{pad}||M_i||\text{pad}_0||x) \) by computing \( y = h(K_e, x||\text{pad}_1) \), then we compute \( H(y) \) further, and we get the very value of \( H^2\text{-MAC}(K||\text{pad}||M_i||\text{pad}_0||x) \), eventually.

**Complexity analysis.** The elements of group \( G_1 \) computed by \( H^2 \) need \( 2^{n/2} \) off-line \( H^2 \) computations. The elements of group \( G_2 \) queried through \( H^2\text{-MAC} \) need \( 2^{n/2} \) on-line \( H^2\text{-MAC} \) queries. The probability of that at least one inner collision happens is 0.632 \([1]\). We can store the values of these elements of both groups in hash tables. The above algorithm will require \( O(2^{n/2}) \) time and space to complete.
4 Conclusion

We can recover the equivalent key in about $2^{n/2}$ on-line queries to $H^2$-MAC and $2^{n/2}$ off-line $H^2$ computations. Our attack shows that the security of $H^2$-MAC is totally dependent on the CR of the underlying hash function, which claims that the security of $H^2$-MAC is totally broken.

Acknowledgement. We thank the anonymous reviewers for their valuable comments. This work was partially supported by the program “Core Electronic Devices, High-end General Purpose Chips and Basic Software Products” in China (No. 2010ZX01037-001-001), and supported by the 973 program of China under contract 2007CB311202, and by National Science Foundation of China through the 61070228 project.

References