International Journal of Security and Its Applications
Vol. 6 No. 2, April, 2012

Equivalent Key Recovery Attack to H^2-MAC

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

In this paper, 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). We can successfully recover the equivalent key of H^2-MAC in about $2^{n/2}$ on-line MAC queries and $2^{n/2}$ off-line hash computations with great probability. This attack shows that the security of H^2-MAC is totally dependent on the collision resistance of the underlying hash function, instead of the PRF-AX of the underlying compression function in the origin security proof of H^2-MAC.

***KeyWords**: H^2-MAC, Equivalent Key Recovery, Collision Resistance, Birthday Paradox.*

1 Introduction

HMAC [3, 2], a derivative of NMAC, is a practically and commonly used, widely standardized MAC construction. HMAC has two advantages. First, HMAC can directly make use of current hash functions, the most widely used ones are based on Merkle-Damgård construction [5, 8], as black boxes. Second, it is provable secure under the assumption that the compression function of the underlying hash function is a pseudo random function (PRF) [2].

For an iterated hash function H with Merkle-Damgård construction, HMAC is defined with secret prefix approach [10] as follow,

$$\mathrm{HMAC}_{(k_{\mathrm{in}},\, k_{\mathrm{out}})}(M) = H(k_{\mathrm{out}}||H(k_{\mathrm{in}}||M)) \tag{1}$$

where M is an input message with arbitrary length, k_{in} and k_{out} are secret b-bit keys derived from a base key K.

However, HMAC has a drawback of managing its secret keys. It has to call the secret keys twice to complete the MAC computation. In ISC 2009, Yasuda proposed H^2-MAC [16], 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 follow,

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

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 [16].

In 2011, Wang [12] proposed an equivalent key recovery attack to H^2-MAC instantiated with the broken MD5 [9, 13, 15], combining the technologies used in [4] and [14], with complexity about 2^{97} on-line MAC queries.

Our contributions. We propose the first equivalent key recovery attack that breaks the security of H^2-MAC instantiated with secure hash functions, without related key setting. The attack is based on the strong assumption that the underlying hash function is collision resistance (CR), which is dropped in the origin security proof of H^2-MAC [16]. We recall that CR is a stronger definition than the PRF-AX assumption. Moreover, our attack is suitable to all of the H^2-MACs instantiated with secure Merkle-Damgård hash functions, since our attack is a general one based on the birthday paradox. This attack is also applicable to NMAC [7] and HMAC with two random keys, in a related key setting.

We notice that H^2-MAC applies the Merkle-Damgård hash function directly in the outer hashing without any secret key, which can't hide the existence of collisions of the inner hashing. 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 $H(H(C||pad||m))$[1], called H^2, in group G_2 through off-line, where Cs and ms 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 [6]. Therefore, the equivalent key of H^2-MAC can be recovered by computing the corresponding value of H^2.

Organization of this paper. We introduce some preliminaries and background, such as birthday paradox, in section two. In section three, we break the security of H^2-MAC by using a generalized birthday attack with two groups, based on the assumption that the underlying hash function is collision resistance. We conclude the paper in the last section.

2 Preliminaries

In this section, we first present some notations, and then recall the birthday paradox in brief.

2.1 Notations

Let h be a compression function mapping $\{0,1\}^n \times \{0,1\}^b \to \{0,1\}^n$, and let H be a concrete hash function mapping $\{0,1\}^* \to \{0,1\}^n$. 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. $\oplus$ means the bit wise exclusive OR. $pad(M)$ denotes the padding bits of M in Merkle-Damgård style. H^2 means that the secret key to H^2-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.

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

2.2 Birthday Paradox

A generalized variant. Given two groups G_1 with r elements, G_2 with s elements drawn uniformly and independently at random from $\{0,1\}^n$, find $x_1 \in G_1$ and $x_2 \in G_2$, such that $x_1 = x_2$.

The probability $\Pr(|G_1 \cap G_2| = i)$ that there are i distinct elements in the intersection of the two groups is denoted by $P(2^n, r, s, i)$. $P(2^n, r, s, i)$ converges towards a Poisson distribution $\wp_\lambda(i)$ with parameter λ, where $r \times s/2^n \to \lambda$, $r, s, 2^n \to +\infty$ [6].

A solution x_1, x_2 exists with great probability once $r \times s \gg 2^n$ holds, and if the list sizes are favourably chosen, the complexity of the optimal algorithm is $O(2^{n/2})$ [6, 11].

The birthday problem has numerous applications throughout cryptography and cryptanalysis, and the direct application is collision searching.

3 Breaking H^2-MAC Using Birthday Paradox

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

If we know the value of the inner intermediate chaining variable of H^2-MAC, $I_K = H(K||pad||M)$, we can construct any selective forgery attack to H^2-MAC. However, it seems that we can't get the value of inner hashing $H(K||pad||M)$ for the application of outer hashing Oh.

Attack principle. To find a way out, we notice that if we consider the inner intermediate chaining variable (the equivalent key) of H^2-MAC as a n-bit input x, then we can view H^2-MAC as a simple hash of $H(x)$. Intuitively, find a collision pair that satisfies $H(x) = H(x')$ is easier than recover the secret key of a MAC, even if H is collision resistance. However, we can't use the birthday attack with one group to recover the equivalent key, because we couldn't know the value of x and x', even a collision pair (x, x') is found.

Fortunately, we can use the generalized birthday attack with two groups. If we can get one collision pair (M, M') that not only satisfies $H^2\text{-MAC}_{(K)}(M) = H^2_{(c)}(M')$, but also satisfies $H(K||pad||M) = H(c||M')$, where c is a b-bit constant or a known parameter set by us[2]. Then, we get the very equivalent key K_e of H^2-MAC through the equation of $K_e = H(K||pad||M) = H(c||M')$. Finally, we already know the value of c and M', hence, K_e can be easily computed.

So the equivalent key recovery attack to H^2-MAC is transformed to the problem of finding a collision pair in two groups, where the elements of one group must be computed by on-line query to H^2-MAC, and the elements of the other group can be computed directly through H^2 off-line. Thus problem is the generalized birthday attack with two groups (sometimes, it is also named as meet-in-the-middle attack).

Generalized Birthday Attack to Recover the Equivalent Key of H^2-MAC

Here, we apply the generalized birthday attack with two groups [6] to H^2-MAC and then recover its equivalent key $K_e = H(K||pad||M_0)$.

We use 1-block messages M_is to generate the corresponding H^2-MAC values, and use 1-block messages M'_js to generate the corresponding H^2 values, where $1 \le i, j \le 2^{n/2}$. The overall strategy of equivalent key recovery attack to H^2-MAC is shown as follows.

[2]Here, an inner collision happens between H^2-MAC and H^2.

1. Generate a group one G_1 with $r = 2^{n/2}$ elements, by computing the corresponding values of $H(H(c||M'_j))$ for r different cs and M'_js, which can be randomly generated. Specifically, c can be a pre-chosen constant.
2. Generate a group two G_2 with $s = 2^{n/2}$ elements, by querying the corresponding values to H^2-MAC oracle with the secret key K for s different M_is, where M_is are randomly generated.
3. There is a pair (M_i, M'_j) that not only satisfies $H^2\text{-MAC}_K(M_i) = H_c^2(M'_j)$, but also satisfies $H(K||pad||M_i) = H(c||M'_j)$ (an inner collision between H^2 and H^2-MAC happens), with great probability [6].
4. Since $H(K||pad||M_i) = H(c||M'_j)$, and we know the value of c and M'_j, we can compute the value of $K_e = H(K||pad||M_i) = H(c||M'_j)$.
5. Let pad_0 and pad_1 be the padding bits of $K||pad||M_i$ and $K||pad||M_i||pad_0||x$, respectively, for arbitrary message x. Hence, we can directly generate the result of $H(K||pad||M_i||pad_0||x)$ by computing $y = h(K_e, x||pad_1)$, then we compute $H(y)$ further, finally we get the very value of $H^2\text{-MAC}(K||pad||M_i||pad_0||x)$.

Why inner collision. In the above attack, an inner collision must be found first. The problem is why an inner collision must happen. If we remove the outer hashing of H^2-MAC, we can directly observer that a collision pair (M_i, M'_j) will be found with great probability, after querying the oracle of $H(K||pad||M_i)$ and $H(c||M'_j)$ with enough times. We recall that the application of outer hashing of H^2-MAC can't hide the existence of such inner collision.

How to judge the inner collision. After a collision pair (M_i, M'_j) that satisfies $H^2\text{-MAC}_K(M_i) = H_c^2(M'_j)$ is found, we first generate the padding bits pad_0 for M_i and M'_j, where $pad_0 = pad(c||M'_j)$. Further, we randomly generate a message x, and append x to $M_i||pad_0$ and $M'_j||pad_0$, respectively. We query the corresponding MAC value on-line to the H^2-MAC oracle for $M_i||pad_0||x$, and we compute the corresponding value for $M_i||pad_0||x$ off-line using H^2. After that, we further check whether the equation $H^2\text{-MAC}_K(M_i||pad_0||x) = H_c^2(M'_j||pad_0||x)$ still holds. If so, (M_i, M'_j) is also an inner collision pair between H^2-MAC and H^2, the attack succeeds. Otherwise, (M_i, M'_j) is an outer collision pair, which will be simply discarded.

Success probability. We calculate the success probability of the above attack. We notice that $r = s = 2^{n/2}$ (hence $\lambda = r{\cdot}s/2^n = 1$), the probability sp of that at least one inner collision happens is computed as

$$sp = 1 - P(2^n, r, s, 0) = 1 - \wp_\lambda(0) + \varepsilon = 1 - e^{-1} + \varepsilon \geq 0.632 \tag{3}$$

where $\varepsilon \leq 10^{-5}$ [6].

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 computed by H^2-MAC need $2^{n/2}$ on-line H^2-MAC queries. We can store the element values of both groups in hash tables. The above algorithm requires $O(2^{n/2})$ time and space to complete.

After an inner collision pair (M_i, M'_j) is found, we can apply $H_c^2(M'_j)$ to compute the equivalent key of the H^2-MAC. Eventually, we can use the recovered equivalent key k_e to launch any selective forgery attack to H^2-MAC without further on-line query, based on M_0. This claims that the security of H^2-MAC is totally broken, moreover, we point out that

the security of H^2-MAC is solely dependent on the collision resistance of the underlying hash function, not the secrecy and strength of the used key.

However, it is interesting to notice that H^2-MAC is provable secure under the assumption of that the underlying compression function h is a PRF-AX [16], which means that collision resistance of the underlying hash function can be dropped. Thus proof and assumption obvious violate our result.

4 Conclusion

We recover the equivalent key of H^2-MAC through applying a generalized birthday attack with two groups, based on the assumption of that the underlying hash function is CR. We can recover thus 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

[1] Bellare, M., Canetti, R., Krawczyk, H.: Pseudorandom functions revisited: the cascade construction and its concrete security. Foundations of Computer Science, Annual IEEE Symposium on 0, 514 (1996)

[2] Bellare, M.: New Proofs for NMAC and HMAC: Security Without Collision-Resistance. In: Dwork, C. (ed.) Advances in Cryptology - CRYPTO 2006, Lecture Notes in Computer Science, vol. 4117, pp. 602–619. Springer Berlin / Heidelberg (2006)

[3] Bellare, M., Canetti, R., Krawczyk, H.: Keying Hash Functions for Message Authentication. In: Koblitz, N. (ed.) Advances in Cryptology CRYPTO' 96, Lecture Notes in Computer Science, vol. 1109, pp. 1–15. Springer Berlin / Heidelberg (1996)

[4] Contini, S., Yin, Y.: Forgery and Partial Key-Recovery Attacks on HMAC and NMAC Using Hash Collisions. In: Lai, X., Chen, K. (eds.) Advances in Cryptology ASIACRYPT 2006, Lecture Notes in Computer Science, vol. 4284, pp. 37–53. Springer Berlin / Heidelberg (2006)

[5] Damgård, I.: A Design Principle for Hash Functions. In: Brassard, G. (ed.) Advances in Cryptology CRYPTO' 89 Proceedings, Lecture Notes in Computer Science, vol. 435, pp. 416–427. Springer Berlin / Heidelberg (1990)

[6] Girault, M., Cohen, R., Campana, M.: A Generalized Birthday Attack. In: Barstow, D., Brauer, W., Brinch Hansen, P., Gries, D., Luckham, D., Moler, C., Pnueli, A., Seegmller, G., Stoer, J., Wirth, N., Gnther, C. (eds.) Advances in Cryptology EUROCRYPT 88, Lecture Notes in Computer Science, vol. 330, pp. 129–156. Springer Berlin / Heidelberg (1988)

[7] Liu, F., Shen, C., Xie, T., Feng, D.: On the security of nmac and its variants. Cryptology ePrint Archive, Report 2011/649 (2011), http://eprint.iacr.org/

[8] Merkle, R.: One Way Hash Functions and DES. In: Brassard, G. (ed.) Advances in Cryptology CRYPTO 89 Proceedings, Lecture Notes in Computer Science, vol. 435, pp. 428–446. Springer Berlin / Heidelberg (1990)

[9] Rivest, R.: The MD5 Message-Digest Algorithm. RFC 1321 (Informational) (Apr 1992), http://www.ietf.org/rfc/rfc1321.txt, updated by RFC 6151

[10] Tsudik, G.: Message authentication with one-way hash functions. SIGCOMM Comput. Commun. Rev. 22, 29–38 (October 1992)

[11] Wagner, D.: A generalized birthday problem. In: Yung, M. (ed.) Advances in Cryptology CRYPTO 2002, Lecture Notes in Computer Science, vol. 2442, pp. 288–304. Springer Berlin / Heidelberg (2002)

[12] Wang, W.: Equivalent Key Recovery Attack on H^2-MAC Instantiated with MD5. In: Kim, T.h., Adeli, H., Robles, R.J., Balitanas, M. (eds.) Information Security and Assurance, Communications in Computer and Information Science, vol. 200, pp. 11–20. Springer Berlin Heidelberg (2011)

[13] Wang, X., Yu, H.: How to Break MD5 and Other Hash Functions. In: Cramer, R. (ed.) Advances in Cryptology EUROCRYPT 2005, Lecture Notes in Computer Science, vol. 3494, pp. 561–561. Springer Berlin / Heidelberg (2005)

[14] Wang, X., Yu, H., Wang, W., Zhang, H., Zhan, T.: Cryptanalysis on HMAC/NMAC-MD5 and MD5-MAC. In: Joux, A. (ed.) Advances in Cryptology - EUROCRYPT 2009, Lecture Notes in Computer Science, vol. 5479, pp. 121–133. Springer Berlin / Heidelberg (2009)

[15] Xie, T., Liu, F., Feng, D.: Could The 1-MSB Input Difference Be The Fastest Collision Attack For MD5?. Eurocrypt 2009, Poster Session, Cryptology ePrint Archive, Report 2008/391 (2008), http://eprint.iacr.org/

[16] Yasuda, K.: HMAC without the "Second" Key. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C. (eds.) Information Security, Lecture Notes in Computer Science, vol. 5735, pp. 443–458. Springer Berlin / Heidelberg (2009)