Practical Password Recovery on an MD5 Challenge and Response*

Yu Sasaki[1], Go Yamamoto[2], and Kazumaro Aoki[2]

[1] The University of Electro-Communications,
Chofugaoka 1-5-1, Chofu-shi, Tokyo, 182-8585, Japan
[2] NTT, 1-1 Hikarinooka, Yokosuka-shi, Kanagawa-ken, 239-0847 Japan

Abstract. This paper shows an attack against APOP protocol which is a challenge-and-response protocol. We utilize the Wang's attack to make collisions in MD5, and apply it to APOP protocol. We confirmed that the first 3 octets of secret key can be recovered by several hundred queries under the man-in-the-middle environment.

1 Introduction

After the Wang's breakthrough [14] to find collision in hash functions, the security of hash functions has much attention. Since how to derive the Wang's attack seems unclear, many studies are conducting to clarify this topic, for example [2]. On the other hand, people started to consider the influences of tractable collision against real applications of hash function. We know that tractable collision is undesirable property for a hash function, but we are not sure how to influence the property to applications of hash function such as digital signature, message authentication, and challenge-and-response protocols. For example, [5] shows how to alter a PostScript document to be signed, and [1] shows the security of NMAC and HMAC can be proven without collision-resistance property. However, no result is known for challenge-and-response protocols.

This paper shows an attack against APOP protocol which is based on MD5 [11] message authentication. Our attack utilizes the collision tractable property in MD5, and successfully recovers the first three octets of password under man-in-the-middle environment. Moreover, the paper is an answer of the problem written in [6]:

> Because of the message extension attack on the prefix approach, the "suffix" approach, MD5$(m \cdot k)$, would seem to be preferred. But another problem arises: the key may be vulnerable to cryptanalysis, depending on the properties of the compression function.

The following sections organize as follows. Section 2 introduces the notation and the previous results. Section 3 abstracts the properties of collisions to be

* We notified Information-technology Promotion Agency, Japan of the result followed by the Japanese ordinance, December 8, 2006. The notification number is IPA#10155887.

used. Section 4 describes an attack for challenge-and-response protocols whose challenge is generated as MD5($m\|k$), where m is variable length. Section 5 discusses how to apply the attack to APOP. Section 6 concludes this paper.

2 Preliminaries

2.1 Notations and Definitions

Bit Strings. For a bit string $b = b_1 b_2 b_3 \cdots b_l$, let $|b|$ be the length l, and let $[b]^i_j$ be the substring $b_i b_{i+1} \cdots b_{j-1} b_j$.

Description of MD5. MD5 [11] is a hash function with Merkle-Damgård structure, which takes an arbitrary length message M as input, and outputs 128-bit length hash value $H(M)$. First of all, an input message M is padded and divided into 512-bit length block messages $(M_0, M_1, \ldots, M_{n-1})$. These block messages will go through compression function (CF) of MD5 with a 128-bit initial value. The initial chaining variable (H_0) are set as follows: $a_0 =$ `0x67452301`, $b_0 =$ `0xefcdab89`, $c_0 =$ `0x98badcfe`, $d_0 =$ `0x10325476`. The procedure of MD5 algorithm is as follows:

$$H_1 = CF(M_0, H_0),\ H_2 = CF(M_1, H_1),\ \ldots,\ H_n = CF(M_{n-1}, H_{n-1})$$

H_n will be the hash value of M.

Compression function of MD5 takes M_{n-1} and H_{n-1} as input, and outputs H_n. First, M_{n-1} is divided into 32-bit messages $(m_0, m_1, \ldots, m_{15})$. Hash value H_{n-1} is divided into 32-bit chaining variables (a_0, b_0, c_0, d_0). The compression function consists of 64 steps. In step i, chaining variables $a_{i-1}, b_{i-1}, c_{i-1}, d_{i-1}$ are updated as follows.

$$a_i = d_{i-1} \qquad b_i = b_{i-1} + (a_{i-1} + f(b_{i-1}, c_{i-1}, d_{i-1}) + m_k + t_i) \lll s_i$$
$$c_i = b_{i-1} \qquad d_i = c_{i-1}$$

f is a Boolean function defined in each round. m_k is one of $(m_0, \ldots, m_{15})$, and index k is defined in each steps. t_i is a constant value defined in each step. $\lll s_i$ denotes left rotation by s_i bits, and s_i is defined in each step. Finally, H_n is computed as $(a_0 + a_{64}, b_0 + b_{64}, c_0 + c_{64}, d_0 + d_{64})$.

2.2 Wang's Results and Its Extension

How to generate MD5 collision in Wang et al.'s method? Wang et al.'s attack is a differential attack. It generates a collision with complexity of 2^{38} MD5 computation[3]. Let m and m' be a pair of messages that yields a collision. Difference Δ is defined to be value of calculation for m' 32-bit wordwise subtracted by value of calculation for m. For example, $\Delta m = m' - m$. The attack procedure is as follows.

[3] In original paper, complexity was estimated as 2^{37} MD5 computation. However, [9] pointed out the mistakes of their estimation, and reevaluated the correct complexity.

1. Find “Message Difference (ΔM)” that yields a collision with high probability.
2. Determine the propagation of ΔM. Propagation of difference is called “Differential Path (DP)”.
3. To realize the DP, generate “Sufficient Conditions (SC)” on the value of chaining variables of calculation for m_i.
4. Determine procedures of “Message Modification (MM)” that satisfy SCs.
5. Search a message that satisfies all SCs by randomly generating messages and applying MM. Let the searched message be M_*.
6. Calculate $M'_* = M_* + \Delta M$. Finally, M_* and M'_* become a collision pair.

This attack generates a collision with two blocks messages. Let $M^{(1)}$ and $M^{(2)}$ is a message for the first and the second block respectively. ΔM of their attack is as follows.

$$\Delta M^{(1)} = (\Delta m_0^{(1)}, \ldots, \Delta m_{15}^{(1)}) = (0,0,0,0,2^{31},0,0,0, \quad 0,0,0,2^{15},0,0,2^{31},0)$$
$$\Delta M^{(2)} = (\Delta m_0^{(2)}, \ldots, \Delta m_{15}^{(2)}) = (0,0,0,0,2^{31},0,0,0, \quad 0,0,0,-2^{15},0,0,2^{31},0)$$

The proposed APOP attack uses the collision attack on MD5 proposed by Wang et al. The property that $m_{15}^{(2)}$ does not have any difference is important for the APOP attack.

The best improved attack known. While no paper has been published on improvement of ΔM and DP, some papers proposed the improvement on SC and MM. Liang and Lai pointed out that Wang et al.’s SCs are insufficient, and showed the truly sufficient conditions [9]. Regarding MM, in November 2005, Sasaki et al. proposed the message modification for starting collision search from the intermediate step [12, 13]. In March 2006, Klima independently proposed message modification named “Tunnel” [7]. The concepts of these two techniques are the same, but Klima applied his technique more widely than Sasaki et al. Klima’s attack is the fastest attack, which finds a collision within a minute by standard notebook PC.

Properties to be used in our attack. APOP attack needs to generate collisions of MD5 many times. In the proposed attack, collision generation technique is based on Wang et al.’s collision attack. We use the same ΔM, DP and SC as Wang et al.’s. Regarding MM, we slightly change the technique because of the limitation of the APOP. Sasaki et al.’s and Klima’s message modification are also available to quickly generate collisions.

2.3 Related Work

In Asiacrypt 2006, a related-key key recovery attack on NMAC/HMAC MD5 was proposed by Contini and Yin [3]. In NMAC/HMAC MD5, message authentication tag is generated by calculating $\mathrm{MD5}_{sk2}(\mathrm{MD5}_{sk1}(m))$, where $\mathrm{MD5}_{ski}(m)$ denotes MD5 calculation for m by using i-th secret key as its initial value. The only difference of NMAC and HMAC is the existence of Key Deriving Function.

They proposed the method to recover sk_1 of NMAC/HMAC MD5. The related part to this paper is that their attack recovers sk_1 without knowing sk_2 by using the known property of MD5: If MD5(a) and MD5(b) ($a \neq b$) are collision, MD5($a||c$) and MD5($b||c$) will also collide for any c. This property is applied to NMAC/HMAC.

$$[\mathrm{MD5}_{sk1}(m) = \mathrm{MD5}_{sk1}(m')] \Rightarrow [\mathrm{N/HMAC}_{sk1,sk2}(m) = \mathrm{N/HMAC}_{sk1,sk2}(m')]$$

Therefore, they can know the occurrence of collision in the first part by looking authentication tags.

Another point is that both of their attack and our attack use the message modification (MM). However, since IV of NMAC/HMAC is secret information, MMs by Contini and Yin are different from MMs by Wang et al.'s. On the other hand, MM used in our attack is based on MM by Wang et al.'s.

Very recently, Leurent independently proposes an attack against APOP protocol [8]. The result includes almost part of this paper, but the motivation is quite different.

3 Properties of Collisions

For a function $h : \{0,1\}^* \to \{0,1\}^k$, if distinct messages $m, m' \in \{0,1\}^*$ satisfy $h(m) = h(m')$ then the pair of the messages (m, m') is called a *collision pair*. In this section we prepare some notions that characterize additional property of collision pairs.

3.1 Extendable Collision Pairs

Many dedicated hash functions like SHA-1 is designed on the Merkle-Damgård construction. In such cases, one can expect that some collisions for the function possesses the property defined below.

Definition 1 (Extendable Collision Pairs). *Let $h : \{0,1\}^* \to \{0,1\}^k$ be a function and let a pair of distinct messages (m, m') is a collision pair, that is, $h(m) = h(m')$ holds. We say (m, m') is an extendable collision pair, if $h(m||x) = h(m'||x)$ holds for any $x \in \{0,1\}^*$.*

This property relates to length extension attack, that is, given $h(m)$ and the length of m but not m itself, one can compute $h(m||x)$ for a given x.

Proposition 1. *Let (m, m') be a collision pair such that $|m| = |m'|$. If for message m message length extension attack is applicable for any $x \in \{0,1\}^*$, then (m, m') is an extendable collision pair.*

Proof. Suppose one can apply message length extension attack on m. It implies that $h(m||x)$ can be computed on input $h(m)$, $|m|$, and x. Since (m, m') is a collision pair, hence $h(m) = h(m')$, which implies $h(m||x) = h(m'||x)$. $\square$

3.2 Tail Collision Pairs

Cryptographic hash functions are usually designed to be collision-resistant. However, some works recently revealed that it is not so hard as considered before to find collisions for SHA-1, MD5, and so on. In this section extending the notion of collision pairs we consider *tail collision pairs*, that is harder to find. In Section 5 we will point out that at least for MD5 Wang's work finds even tail collision pairs, not only collision pairs.

Definition 2 (x-Tail Collision Pairs). *Let $h : \{0,1\}^* \to \{0,1\}^k$ be a function. An x-tail collision pair is a pair of distinct messages (m, m') such that $h(m||x) = h(m'||x)$.*

If x is set to the NULL string, NULL-tail collision pair is nothing but the conventional collision pair.

Definition 3 (Depth and Proper x-Tail Collision Pairs). *Let (m, m') be an x-tail collision pair. We define the depth of the pair by the minimum positive integer d such that (m, m') is a $[x]_d^1$-tail collision pair. If the depth of an x-tail collision pair (m, m') equals to $|x|$, then we call (m, m') a proper x-tail collision pair.*

Definition 4. *An extendable x-tail collision pair (m, m') is an x-tail collision pair such that $(m||x, m'||x)$ is an extendable collision pair.*

An extendable x-tail collision pair is also an extendable $x||y$-tail collision pair for an arbitrary $y \in \{0,1\}^*$.

We usually suppose that for distinct messages x and x', a proper x-tail collision pair (m, m') is very unlikely to be a proper x'-tail collision pair at the same time. We call this hypothesis *Tail Collision Soundness.*

4 The Attack

Let $\mathcal{O}$ be an oracle which outputs $h(x||s)$ on input x, where s is the secret stored inside $\mathcal{O}$. Let $\mathcal{T}_n$ be an oracle which on input x with $|x| < n$ outputs (m, m') such that (m, m') is an extendable proper x-tail collision pair of h. Then, our attack extracts the first n bits of s with accesses to $\mathcal{O}$ and $\mathcal{T}_n$.

Theorem 1. *Suppose h is a function with Tail Collision Soundness. Let $\mathcal{O}$ and $\mathcal{T}_i$ are oracles as described above. Let s be the secret string stored in $\mathcal{O}$. Then, there is an algorithm that outputs the first n bits of s, within $2n$ times $\mathcal{O}$ calls and a single $\mathcal{T}_i$ call for each $i = 1, 2, \ldots, n$.*

This attack affects to the real life. In Section 5 it is shown that one can implement $\mathcal{T}_{32}$ for MD5, which impacts on the security of APOP [10].

Proof. Consider the algorithm described below.

```
Input: $n \in \mathbb{N}$,
Output: the first $n$ bits of $s$.

1. Set $G \leftarrow \mathbf{NULL}$.
2. Send $G||0$ to $\mathcal{T}_{|G|+1}$ and receive $(m, m')$.
3. Send $m$ to $\mathcal{O}$ and receive $a$.
4. Send $m'$ to $\mathcal{O}$ and receive $a'$.
5. If $a = a'$, then set $G \leftarrow G||0$. Otherwise set $G \leftarrow G||1$.
6. If $|G|$ reaches $n$, then output $G$ and exit. Otherwise goto Step 2.
```

It suffices to show that Step 5 gives the correct guess for $[s]^1_{|G|+1}$ assuming that G is the correct guess for $[s]^1_{|G|}$ in Step 2.

In Step 5, $a = a'$ holds if $G||0$ is the correct guess for $[s]^1_{|G|+1}$, because $\mathcal{T}_{|G|+1}$ returns an extendable $G||0$-tail collision pair, that is, $(m||[s]^1_{|G|+1}, m'||[s]^1_{|G|+1})$ is an extendable collision pair, hence $h(m||s) = h(m'||s)$.

Suppose $G||0$ is not the correct guess and $a = a'$ holds. In particular (m, m') is an s-tail collision pair. Let d be the depth of s-tail collision pair (m, m'). Then, (m, m') is a proper $[s]^1_d$-tail collision pair. Suppose $d > |G|$. (m, m') is also a proper $G||0$-tail collision pair, while $[s]^1_d \neq G||0$ since $G||0$ is not the correct guess for s. This contradicts our assumption of Tail Collision Soundness. If $d \leq |G|$, then $[s]^1_d = [G||0]^1_d$ since G is the correct guess for $[s]^1_{|G|}$. Hence the depth of proper $G||0$-tail collision pair (m, m') is less than $|(G||0)|$. This contradicts to the definition. □

In real life, when one deals with APOP, mail user agents can take parts as the oracle $\mathcal{O}$ described above, while $\mathcal{O}$ does not necessarily accept an input with an arbitrary length. If the input for $\mathcal{O}$ is restricted to octet strings, the guess at Step 2 should be repeated for 255 times in the worst case.

5 In Real Life

5.1 Brief Description of APOP

APOP is an optional feature of POP defined in [10]. In the conventional POP, users are authenticated by plaintext passwords. APOP authenticates users by an MD5 challenge-response.

In an APOP session, the challenge is given in the first message from the APOP server, formatted like `<26099.1163472291@hostname.fqdn>` corresponding to the 'msg-id' defined in [4]. The response from the user agent is supposed to be MD5(*challenge*||*password*).

The attack described in Section 4 can be applied to APOP, where the user agent participates as the oracle $\mathcal{O}$, if one can implement the oracle $\mathcal{T}$ whose outputs are accepted by the user agent.

5.2 Behavior of Popular Mail User Agents

We experimented on actual conditions for accepting challenge strings with popular mail user agents. We choose Thunderbird-1.5, Mew-5.1, Al-Mail-1.13a, Becky!-2.27, EdMax-2.85.5F as the targets. They are widely used in Japan and supports APOP.

First of all challenge strings must begin with `'<'` and must end with `'>'`, since challenge strings are extracted in such a way from the first message of the server, which can contain some comments or greeting messages besides with the challenge string.

The detail of the experiments are as described below.

Set octet string $\texttt{s} = (\texttt{'<'}, \texttt{0x02}, \texttt{0x03}, \ldots, \texttt{0xff}, \texttt{0x01}, \texttt{0x02}, \texttt{0x03}, \ldots, \texttt{0xff}, \texttt{0x01}, \texttt{0x02}, \texttt{0x03}, \ldots, \texttt{0xfe}, \texttt{'>'})$.

Experiment 1:

1. Submit s to each mail user agent as the challenge from the server.
2. Check whether the mail user agent returns the correct response.

Experiment 2:

1. Replace all of `0x0a`(LF), `'<'`, and `'>'` in s with `'X'` except for the first character and the last character.
2. Submit s to each mail user agent as the challenge from the server.
3. Check whether the mail user agent returns the correct response.

Experiment 3:

1. Replace all of `0x0a`(LF), `'<'`, `'>'`, `'@'` in s with `'X'` except for the first character and the last character.
2. Submit s to each mail user agent as the challenge from the server.
3. Check whether the mail user agent returns the correct response.

The results of the experiments are shown in Table 1, where OK implies the agent returns the correct response, NG implies response is incorrect or the agent returns an error.

Table 1. Experimental results on mail user agents

agent name	Exp1	Exp2	Exp3
Thunderbird-1.5	NG	OK	NG
Mew-5.1	NG	OK	NG
Al-Mail-1.13a	NG	OK	NG
Becky!-2.27	NG	OK	NG
EdMax-2.85.5F	NG	OK	OK

The result supports the conditions for accepting challenge strings described below:

- Strings must begin with `'<'` and end with `'>'`.
- Strings must not contain `0x0a`(LF).
- Strings must not contain `'<'` nor `'>'` except for the first character and the last character.
- Strings should contain at least one `'@'`.

In addition, strings must not contain `0x00`(NULL), which terminates string.

5.3 Implementation of $\mathcal{T}_{32}$ for MD5

In this section, we explain how to implement $\mathcal{T}_{32}$ for MD5, that is, how to generate extendable tail collision pair of MD5.

When we generate x and x', we use the collision attack proposed by Wang et al. However, different from the collision attack, x must consist of octets that are accepted by mail user agents. At first, to simplify the situation, we ignore this limitation. Therefore, we assume that any bit sequence can be chosen for x. (We remove this assumption later).

Impact of a fixed tail value. If the following two facts are achieved, we can obtain extendable tail collision pair (x, x').

- x and x' can have the appropriate message differences.
- x that satisfies all SCs can be found in practical time.

The value of the tail part for x and x' must be identical, therefore, no message difference is allowed in the tail part. Fortunately, as is described in Section 2.2, message difference for Wang et al.'s attack does not involve m_{15} but involves the MSB of m_{14}. Therefore, we can locate tail part in m_{15} but cannot locate it in the MSB of m_{14}. Consequently, as long as the tail part is less than or equal to 39 bits (= 4 octets and 7 bits) [4], we can make the desired message differences, and find a collision pair. This means, we can implement $\mathcal{T}_i$, $(0 \leq i \leq 39)$, but cannot implement it for $i \geq 40$.

Message Modification with fixed m_{15}. To find x that satisfies all SCs, message modification is available. Since we don't generate a collision pair but a tail collision pair, exactly the same message modification as Wang et al.'s attack cannot be applied. However, since tail part of x is at most 39 bits, other 473 bits still have freedom. This enables us to find x that satisfies all SCs in practical time ($\approx 2^{30}$ MD5 computations).

Remember the single-message modification (SMM) proposed by Wang et al. In this method, all sufficient conditions for step i, $(1 \leq i \leq 16)$ are satisfied by

[4] If the tail part is more than 39 bits, it will locate in m_{14}. Since MD5 uses big endian, it will locate in the MSB of m_{14}, and we cannot make the message differences.

modifying m_{i-1}. In the APOP attack, since m_0 to m_{14} can freely be chosen, all SCs up to step 15 are satisfied by the SMM. Similarly, MMM that does not involve m_{15} can be applied to the APOP attack.

After the SMM and MMM that does not involve m_{15} are applied, the number of remaining SCs becomes 34, which obtains the desired x and x' with complexity of 2^{34} MD5 computations. By changing procedures for message modification that involves m_{15} so that m_{15} would not be involved, 4 more SCs can be satisfied. Finally, x and x' are generated with complexity of 2^{30} MD5 computations.

Other message modification named "Tunnel" by Klima [7] or "Message modification for starting the collision search from the intermediate step" by Sasaki et al. [13] are also available. By combining SMM, MMM and these modification, x and x'can be generated in about five seconds on a PC.

5.4 How to avoid the limitation of APOP challenge string?

In the beginning of this section, we assumed that any bit string can be chosen as x. However, some strings cannot be accepted by the actual APOP. Forbidden strings are described in Section 5.2.

Start from `'<'` and include `'@'`. To avoid these limitations, the property that Wang et al.'s attack works for any initial value can be used. That is, we put an additional common message block before the tail collision. Let the 512-bit message that starts from `'<'` and include an `'@'` be *Pre*. The generated tail collision pair becomes $(Pre||x)$ and $(Pre||x')$.

End with `'>'`. APOP string must end with `'>'`. This means we need an additional fixed character in the tail part of x and x'. Because of this limitation, the generated tail collision pair becomes $(Pre||x||$`'>'`$)$ and $(Pre||x'||$`'>'`$)$. As explained before, fixed length can be 39 bits in maximum. Therefore, even with the limitation of `'>'`, we still have freedom for 31 bits, that is, 3 octets in the tail part. This is the reason why the number of recoverable octets is 3. Note that several Japanese mail user agents seem to parse the challenge message as Japanese string. That is, `'>'` may be considered as the part of the next previous character.

Do not include `0x00`(NULL), `0x0a`(LF), `'<'` and `'>'`. APOP string must not include `0x00`(NULL), `0x0a`(LF), `'<'` and `'>'`. For these strings, we have not tried intelligent countermeasures. If these strings are included in the message after message modification, we choose other random message.

6 Conclusion

This paper shows an attack against a challenge-and-response protocol whose challenge is based on MD5$(m||k)$, where m is random string whose length is variable and k is a secret key, utilizing the collision tractable property in MD5. The attack can be applicable to APOP and recovers the first three octets of password using several hundred queries under the man-in-the-middle environment. The result implies that the collision property in hash function does not only influence the security against digital signature, but also challenge-and-response protocols. We should be aware of the risk using collision tractable hash function such as MD5 even if the usage of the hash function does not seem to depend on the collision resistance property.

This paper defines the new properties regarding hash functions, extendable collision pairs and tail collision pairs. How more vulnerable are other dedicated hash functions than those constructed on the Merkle-Damgård way on these properties is an independent interests.

7 Work in Progress

The proposed attack can recover 3 octets of APOP password. So the question is whether we can recover more octets.

The biggest reason that we cannot recover the 4th octet is that message difference (ΔM) exists in the MSB of m_{14}. Therefore, to extend the recoverable octets, we need to change the ΔM. ΔM that is suitable for the APOP attack will have following two properties.

1. Execution time for generating tail collision pair is practical.
2. The length of tail part can be long, therefore, message differences exist in only early part of the message.

Wang et al.'s ΔM can find collision within few seconds, but can recover only 3 octets. Therefore, there may exist other ΔM that is somehow worse than Wang et al.'s in terms of finding collision, but can recover more octets. However, so far, only one research has existed to analyze the construction method of ΔM and differential path of MD5 [2], and no research has been succeeded in constructing collision attack by using different ΔM from Wang et al.'s. To extend the APOP attack, further study for the construction of ΔM and differential path for MD5 collision attack will be needed.

References

1. Mihir Bellare. New proofs for NMAC and HMAC : Security without collision-resistance. In Cynthia Dwork, editor, *Advances in Cryptology — CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 602–619. Springer-Verlag, Berlin, Heidelberg, New York, 2006.
2. John Black, Martin Cochran, and Trevor Highland. A study of the MD5 attacks: Insights and improvements. In Matthew Robshaw, editor, *Fast Software Encryption — 13th International Workshop, FSE 2006*, volume 4047 of *Lecture Notes in Computer Science*, pages 262–277, Berlin, Heidelberg, New York, 2006. Springer-Verlag.

3. Scott Contini and Yiqun Lisa Yin. Forgery and partial key-recovery attacks on HMAC and NMAC using hash collisions. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology — ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 37–53. Springer-Verlag, Berlin, Heidelberg, New York, 2006.
4. David H. Crocker. *Request for Comments 822: Standard for ARPA Internet Text Messages*. The Internet Engineering Task Force, 1982. (`http://www.ietf.org/rfc/rfc822.txt`).
5. Magnus Daum and Stefan Lucks. Hash collisions (the poisoned message attack). (`http://th.informatik.uni-mannheim.de/people/lucks/HashCollisions/`), 2005.
6. Burton S. Kaliski Jr. and Matthew J. B. Robshaw. Message authentication with MD5. *CryptoBytes*, 1(1):5–8, 1995.
7. Vlastimil Klima. Tunnels in hash functions: MD5 collisions within a minute. (IACR Cryptology ePrint Archive: Report 2006/105 `http://eprint.iacr.org/2006/105`), 2006.
8. Gaetan Leurent. Message freedom in MD4 and MD5 collisions: Application to APOP. FSE2007 submission version, 2007.
9. Jie Liang and Xuejia Lai. Improved collision attack on hash function MD5. (IACR Cryptology ePrint Archive: Report 2005/425 `http://eprint.iacr.org/2005/425`), 2005.
10. John G. Myers and Marshall T. Rose. *Request for Comments 1939 (STD 53): Post Office Protocol - Version 3*. The Internet Engineering Task Force, 1996. (`http://www.ietf.org/rfc/rfc1939.txt`).
11. Ronald L. Rivest. *Request for Comments 1321: The MD5 Message Digest Algorithm*. The Internet Engineering Task Force, 1992. (`http://www.ietf.org/rfc/rfc1321.txt`).
12. Yu Sasaki, Yusuke Naito, Noboru Kunihiro, and Kazuo Ohta. Improved collision attack on MD5. (IACR Cryptology ePrint Archive: Report 2005/400 `http://eprint.iacr.org/2005/400`), 2005.
13. Yu Sasaki, Yusuke Naito, Noboru Kunihiro, and Kazuo Ohta. Improved collision attacks on MD4 and MD5. *IEICE Transactions Fundamentals of Electronics, Communications and Computer Sciences (Japan)*, E90-A(1):36–47, 2007. (The initial result was announced as [12].).
14. Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *Advances in Cryptology — EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer-Verlag, Berlin, Heidelberg, New York, 2005.