Second Preimage Attack on SHAMATA-512

Kota Ideguchi* and Dai Watanabe†
Hitachi Ltd.

February 19, 2009

Abstract

We present a second preimage attack on SHAMATA-512, which is a hash function of 512-bit output and one of the first round candidates of the SHA-3 competition. The attack uses differential paths that hold with a probability one and a meet-in-the-middle approach to find second preimages. The time complexity is about $2^{451.7}$ computation of the step function and the memory complexity is about $2^{452.7}$ blocks of 128 bits.

1 Short Description of SHAMATA-512

The hash function SHAMATA[1] is a register based hash function. The internal state (chaining value) is of 2048-bit length and stored in 16 128-bit registers; four B registers and twelve K registers. A message is padded to a multiple of 128 bits and the message blocks are processed by the step function sequentially. Let $pad(x) = M_0||M_1||\cdots||M_{l-1}$ be a l-block padded message. The hash value $y = H(x)$ is computed as follows:

$$\begin{aligned} S_0 &= \text{Initialization}(IV), \\ S_{i+1} &= \text{StepFunction}(S_i, M_i, i), \quad i = 0, 1, \cdots, l-1 \\ y &= \text{Finalization}(S_l, l), \end{aligned}$$

where S_i is the internal state before the i-th step is applied. We call the update process, described by $S_{i+1} = \text{StepFunction}(S_i, M_i, i)$, the i-th step. The StepFunction is called *UpdateRegister* in the specification of the hash function.

*kota.ideguchi.yf@hitachi.com
†dai.watanabe.td@hitachi.com

The StepFunction is defined as follows:

$$
\begin{aligned}
B[0]_{i+1} &= B[2]_i \oplus P(M_i) \oplus (i+1), \\
B[2]_{i+1} &= K[0]_i \oplus K[9]_i \oplus B[0]_i \oplus ARF^2(B[2]_i \oplus P(M_i) \oplus (i+1)), \\
K[10]_{i+1} &= B[0]_i \oplus ARF^2(B[2]_i \oplus P(M_i) \oplus (i+1)), \\
K[2n]_{i+1} &= K[2n+2]_i, \quad n = 0, 2, 4, 8 \\
B[1]_{i+1} &= B[3]_i \oplus Q(m_i) \oplus (i+1), \\
B[3]_{i+1} &= K[1]_i \oplus K[10]_i \oplus B[1]_i \oplus ARF^2(B[3]_i \oplus Q(m_i) \oplus (i+1)), \\
K[11]_{i+1} &= B[1]_i \oplus ARF^2(B[3]_i \oplus Q(m_i) \oplus (i+1)), \\
K[9]_{i+1} &= K[11]_i \oplus Q'(M_i), \\
K[7]_{i+1} &= K[9]_i, \\
K[5]_{i+1} &= K[7]_i \oplus P(M_i), \\
K[3]_{i+1} &= K[5]_i \oplus Q(M_i), \\
K[1]_{i+1} &= K[3]_i \oplus P'(M_i),
\end{aligned}
$$

where $B[n]_i$ and $K[n]_i$ are values of the registers $B[n]$ and $K[n]$, respectively, before the i-th step. The functions P and Q are linear functions which are defined by the multiplication of a MDS matrix. Q' and P' are also linear functions whose outputs are concatenations of the halves of the outputs of P and Q. ARF is the AES round function without `AddRoundKey`.

2 Second Preimage Attack by Meet-in-the-Middle

In the section, we describe a second preimage attack for SHAMATA-512.

2.1 Notation

Let $x^{(0)}$ and $y^{(0)}$ be the target message and its hash value respectively.

$$y^{(0)} = H(x^{(0)}). \tag{1}$$

Let the padded message consist of l 128-bit message blocks, $pad(x^{(0)}) = M_0^{(0)} || M_1^{(0)} || \cdots || M_{l-1}^{(0)}$. The internal state before the i-th step is denoted by $S_i^{(0)}$. The values of registers $B[n]$ and $K[n]$ before the i-th step are denoted by $B[n]_i^{(0)}$ and $K[n]_i^{(0)}$, respectively.

The goal of the attack is to find a message x which is not equal to $x^{(0)}$ and gives the same hash value as $x^{(0)}$ does:

$$y^{(0)} = H(x), \quad x \neq x^{(0)}. \tag{2}$$

We denote the padded message by $pad(x) = M_0 || M_1 || \cdots || M_{l-1}$. The internal state before the i-th step is denoted by S_i. The values of registers $B[n]$ and $K[n]$ before the i-th step is denoted by $B[n]_i$ and $K[n]_i$, respectively. The differences between the values related to $x^{(0)}$ and those related to x are defined as follows:

$$\sigma_i = S_i \oplus S_i^{(0)}, \quad b[n]_i = B[n]_i \oplus B[n]_i^{(0)}, \quad k[n]_i = K[n]_i \oplus K[n]_i^{(0)}, \quad m_i = M_i \oplus M_i^{(0)}, \tag{3}$$

2.2 Second Preimage Attack

Our attack aims to find a second preimage x such that the block length of $pad(x)$ is the same as that of $pad(x^{(0)})$, which is l. If the internal state difference after $l-1$-th step, σ_l, is equal to zero for two messages with the same block length l, then the hash value difference becomes also zero. Thus, the attack aims to find x that derives $\sigma_l = 0$.

The attack can be applied when $l \geq 27$. We assume that this condition holds.

Our attack uses a meet-in-the-middle approach. We divide a message into two segments: the first λ message blocks $M_0||\cdots||M_{\lambda-1}$ and the last $(l-\lambda)$ message blocks $M_\lambda||\cdots||M_{l-1}$. The integer λ must satisfy $\lambda \geq 13$ and $(l-\lambda) \geq 14$. Such a λ exists, because $l \geq 27$.

A procedure of the attack is as follows.

Step 1 For candidates of the first segment $M_0||\cdots||M_{\lambda-1}$, build 2^{448} messages of λ blocks that have the property that when each of these messages is used to update the initial internal state difference $\sigma_0 = 0$ to an internal state difference after the $\lambda-1$-th step σ_λ, the internal state difference σ_λ satisfies the following equations:

$$\begin{aligned} &b[0]_\lambda = k[0]_\lambda = k[2]_\lambda = k[4]_\lambda = k[6]_\lambda = k[8]_\lambda = k[10]_\lambda = 0, \\ &b[2]_\lambda = k[7]_\lambda, \quad k[5]_\lambda = 0. \end{aligned} \tag{4}$$

A way to build such messages is described in section 2.3.

As will be explained in section 2.3, we need to compute the step function λ times to build such a message. Thus, $\lambda \times 2^{448}$ step function evaluations are required for the step 1. We store the built messages and the corresponding internal states after the $(\lambda-1)$-th step in pairs on a storage and denote this set by V_1. Thus, the memory to store $(\lambda+7)\times 2^{448}$ 128-bit blocks is required.

Step 2 For candidates of the second (last) segment $M_\lambda||\cdots||M_{k-1}$, build 2^{448} messages of $(l-\lambda)$ blocks that have the property that when each of these messages is used to reverse the final internal state difference $\sigma_l = 0$ to an internal state difference before the λ-th step σ_λ, the internal state difference σ_λ satisfies the equation (4).

A way to build such messages is described in section 2.4.

As will be explained in section 2.4, we need to compute the inverse of the step function $(l-\lambda)$ times to build such a message. $(k-\lambda)\times 2^{448}$ step function evaluations are required for the step 2 because the computational complexity of the inverse of the step function is the same as that of the step function. We store the built messages and the corresponding internal states difference before the λ-th step in pairs on a storage and denote this set by V_2. Thus, the memory to store $(l-\lambda+7)\times 2^{448}$ 128-bit blocks is required.

Step 3 Because the space of the internal state difference (4) is 896-bit volume, there exists with a high probability a pair of messages (x_1, x_2), where x_1 belongs to V_1 and x_2 belongs to V_2, that derive the same internal state difference after the $(\lambda-1)$-th step, σ_λ. Then, the concatenated message $x_1||x_2$ is a (padded) second preimage, because this message updates the initial internal state difference $\sigma_0 = 0$ to the final internal state difference $\sigma_l = 0$.

An overview of the attack is depicted in figure 1.

The above attack needs $(\lambda+(l-\lambda))\times 2^{448} = l \times 2^{448}$ evaluations of the step function and memory to store $(\lambda+7+l-\lambda+7)\times 2^{448} = (l+14)\times 2^{448}$ 128-bit blocks. Actually, because the step1 and step2 are commutative and the step3 can be merged with the step2 or step1, we can easily improve the procedure and reduce the memory to $(\max(\lambda, l-\lambda)+16)\times 2^{448}$ block size. When we choose $\lambda = 13$, which is always possible, the memory becomes about $2^{452.7}$ block size.

In the next two sections, we explain how to obtain the elements of V_1 and V_2.

2.3 Building Message Candidates for the First Segment of Message

In the section, we show how to obtain a message of λ blocks $M_0||\cdots||M_{\lambda-1}$ that is used to update the initial internal state difference σ_0 to an internal state difference σ_λ satisfying equation (4).

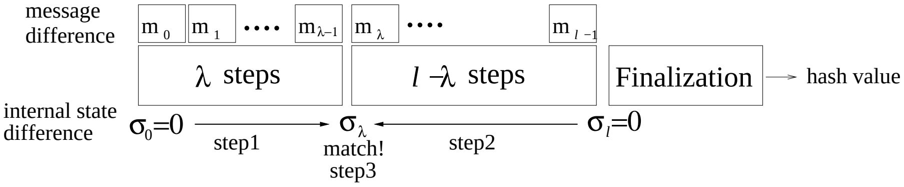

Figure 1: Overview of attack

The step function updates an internal state difference σ_i as follows:

$$b[0]_{i+1} = b[2]_i \oplus P(m_i), \qquad b[2]_{i+1} = k[0]_i \oplus k[9]_i \oplus b[0]_i \oplus \Delta_{X_i^{(0)}}(b[2]_i \oplus P(m_i)), \tag{5}$$

$$k[10]_{i+1} = b[0]_i \oplus \Delta_{X_i^{(0)}}(b[2]_i \oplus P(m_i)), \qquad k[2n]_{i+1} = k[2n+2]_i, \quad n = 0,2,4,8 \tag{6}$$

$$b[1]_{i+1} = b[3]_i \oplus Q(m_i), \qquad b[3]_{i+1} = k[1]_i \oplus k[10]_i \oplus b[1]_i \oplus \Delta_{Y_i^{(0)}}(b[3]_i \oplus Q(m_i)), \tag{7}$$

$$k[11]_{i+1} = b[1]_i \oplus \Delta_{Y_i^{(0)}}(b[3]_i \oplus Q(m_i)), \qquad k[9]_{i+1} = k[11]_i \oplus Q'(m_i), \tag{8}$$

$$k[7]_{i+1} = k[9]_i, \qquad k[5]_{i+1} = k[7]_i \oplus P(m_i), \tag{9}$$

$$k[3]_{i+1} = k[5]_i \oplus Q(m_i), \qquad k[1]_{i+1} = k[3]_i \oplus P'(m_i), \tag{10}$$

where $\Delta_X(r)$, $X_i^{(0)}$, and $Y_i^{(0)}$ are defined as follows,

$$\Delta_X(r) = ARF^2(r) \oplus ARF^2(X \oplus r),$$
$$X_i^{(0)} = B[2]_i^{(0)} \oplus P(M_i^{(0)}) \oplus (i+1), \qquad Y_i^{(0)} = B[3]_i^{(0)} \oplus Q(M_i^{(0)}) \oplus (i+1).$$

Theorem 1. *Consider any internal state difference before the i-th step σ_i. If nine message block differences $m_i, \cdots, m_{i+8}$ are set by the equations $P(m_j) = b[2]_j$ for $j = i, i+1, \cdots, i+8$, then an internal state difference after the $(i+8)$-th step σ_{i+9} satisfies the following equations:*

$$b[0]_{i+9} = k[0]_{i+9} = k[2]_{i+9} = k[4]_{i+9} = k[6]_{i+9} = k[8]_{i+9} = k[10]_{i+9} = 0,$$
$$b[2]_{i+9} = k[7]_{i+9}, \quad k[5]_{i+9} = 0. \tag{11}$$

Proof. We prove the theorem by using the equations (5)-(10) and $P(m_j) = b[2]_j$, $(j = i, \cdots, i+8)$.
First, by using the left equation (5) and $P(m_j) = b[2]_j$, $(j = i, \cdots, i+8)$, the following equations hold:

$$b[0]_j = 0, \quad j = i+1, \cdots, i+9$$

If $P(m_j) = b[2]_j$, the input difference of ARF^2 at the $j-th$ step becomes zero and then the output difference of ARF^2 also becomes zero. Therefore, by using the left equation of (6), the following equations hold:

$$k[10]_j = 0, \quad j = i+2, \cdots, i+9$$

Then, by using the right equation of (6), the following equations hold.

$$k[8]_j = 0, \quad j = i+3, \cdots, i+9$$
$$k[6]_j = 0, \quad j = i+4, \cdots, i+9$$
$$k[4]_j = 0, \quad j = i+5, \cdots, i+9$$
$$k[2]_j = 0, \quad j = i+6, \cdots, i+9$$
$$k[0]_j = 0. \quad j = i+7, \cdots, i+9$$

$b_{[}2]_{j+1}$ is equal to $k[9]_j$ for $j = i+7, i+8$ by using the right equation of (5) because $b[0]_j = k[0]_j = 0$ for $j = i+7, i+8$ and the output differences of ARF^2 at the $i+7$-th step and the

$i+8$-th step are zero. Then, by using the left equation of (9), the following equations hold.

$$k[7]_j = b[2]_j, \quad j = i+8, i+9$$

Finally, we obtain the following equation by using the right equation of (9).

$$k[5]_{i+9} = k[7]_{i+8} \oplus P(m_{i+8}) = k[7]_{i+8} \oplus b[2]_{i+8} = 0.$$

□

Then, we can obtain the desired messages by the following procedure.

step 1-1 Starting with the initial internal state difference $\sigma_0 = 0$, we update the internal state using $(\lambda - 9)$ message block differences $m_0, m_1, \cdots, m_{\lambda-8}$ and obtain $\sigma_{\lambda-9}$. In this step, we can choose freely the message block differences $m_0, m_1, \cdots, m_{\lambda-8}$.

step 1-2 We determine message block differences $m_{\lambda-9}, \cdots, m_{\lambda-1}$ by using theorem 1 with $i = (\lambda - 9)$ and obtain the internal state difference σ_λ satisfying the equation (4). Then, a message candidate is obtained by xoring $M_0^{(0)}|| \cdots ||M_{\lambda-1}^{(0)}$ with $m_0|| \cdots ||m_{\lambda-1}$.

The number of message difference $m_0|| \cdots ||m_{\lambda-9}$ that we can choose at step 1-1, is greater than 2^{448} because $(\lambda - 9) \geq 4$. Therefore, we can build the 2^{448} message candidates.

In this procedure, λ step function evaluations is required to build a message candidate and the corresponding internal state difference. Thus, building V_1, which consists of 2^{448} pairs of a message candidate and the corresponding internal state difference, requires $\lambda \times 2^{448}$ evaluations of the step function.

2.4 Building Message Candidates for the Second Segment of Message

In the section, we show how to obtain a message of $l - \lambda$ blocks $M_\lambda|| \cdots ||M_{l-1}$ that is used to reversely update the final internal state difference $\sigma_l = 0$ (after the $(l-1)$-th step) to an internal state difference σ_λ (before the λ-th state) satisfying the equation (4).

Solving the equations (5)-(10) for $b[n]_i$'s and $k[n]_i$'s, we obtain the inverse function of the step function,

$$\begin{aligned}
&b[0]_i = k[10]_{i+1} \oplus \Delta_{X_i^{(0)}}(b[0]_{i+1}), \qquad b[2]_i = b[0]_{i+1} \oplus P(m_i),\\
&k[0]_i = b[2]_{i+1} \oplus k[10]_{i+1} \oplus k[7]_{i+1}, \qquad k[n]_i = k[n-2]_{i+1}, \quad n = 2,4,6,8,10\\
&b[1]_i = k[11]_{i+1} \oplus \Delta_{Y_i^{(0)}}(b[1]_{i+1}), \qquad b[3]_i = b[1]_{i+1} \oplus Q(m_i),\\
&k[1]_i = b[3]_{i+1} \oplus k[11]_{i+1} \oplus k[8]_{i+1}, \qquad k[3]_i = k[1]_{i+1} \oplus P'(m_i),\\
&k[5]_i = k[3]_{i+1} \oplus Q(m_i), \qquad k[7]_i = k[5]_{i+1} \oplus P(m_i),\\
&k[9]_i = k[7]_{i+1}, \qquad k[11]_i = k[9]_{i+1} \oplus Q'(m_i). \qquad (12)
\end{aligned}$$

Theorem 2. *Consider any internal state difference after the $(i+8)$-th step σ_{i+9}. We denote this internal state difference by*

$$b[n]_{i+9} = r_n, \ (n = 0,1,2,3) \qquad k[n]_{i+9} = s_n, \ (n = 0, \cdots, 11) \qquad (13)$$

If the nine message block differences $m_i, \cdots, m_{i+8}$ is set by the following equations (14)-(22), *then the internal state difference before i-th step, σ_i, satisfies equation* (11)*.*

$$\begin{aligned}
Q(m_{i+8}) =&\tilde{s}_{10} \oplus s_6 \oplus s_3 \oplus \Delta_{X_i^{(0)}}(\tilde{r}_0), && (14)\\
Q(m_{i+7}) =&\tilde{s}_8 \oplus s_4 \oplus s_1 \oplus P'(m_{i+8}), && (15)\\
Q(m_{i+6}) =&s_8 \oplus \tilde{s}_6 \oplus s_2 \oplus r_3 \oplus s_{11} \oplus P'(m_{i+7}), && (16)\\
Q(m_{i+5}) =&s_6 \oplus \tilde{s}_4 \oplus s_0 \oplus r_1 \oplus s_9 \oplus P'(m_{i+6}) \oplus Q(m_{i+8}) \oplus Q'(m_{i+8}), && (17)\\
Q(m_{i+4}) =&r_2 \oplus s_{10} \oplus s_4 \oplus \tilde{s}_2 \oplus \tilde{s}_{11} \oplus P'(m_{i+5}) \oplus Q(m_{i+7}) \oplus Q'(m_{i+7}), && (18)\\
Q(m_{i+3}) =&r_0 \oplus s_8 \oplus s_2 \oplus \tilde{s}_0 \oplus \tilde{s}_9 \oplus P(m_{i+8}) \oplus P'(m_{i+4}) \oplus Q(m_{i+6}) \oplus Q'(m_{i+8})\\
&\oplus Q'(m_{i+6}), && (19)\\
Q(m_{i+2}) =&\tilde{r}_2 \oplus \tilde{s}_{10} \oplus s_6 \oplus s_0 \oplus s_7 \oplus \Delta_{Y_{i+6}^{(0)}}(\tilde{s}_9 \oplus Q'(m_{i+8})) \oplus P(m_{i+7}) \oplus P'(m_{i+3})\\
&\oplus Q(m_{i+5}) \oplus Q'(m_{i+5}) \oplus Q'(m_{i+7}), && (20)\\
Q(m_{i+1}) =&r_2 \oplus \tilde{r}_0 \oplus s_{10} \oplus s_7 \oplus s_5 \oplus s_1 \oplus \Delta_{Y_{i+5}^{(0)}}(s_7 \oplus \Delta_{Y_{i+6}^{(0)}}(\tilde{s}_9 \oplus Q'(m_{i+8})) \oplus Q'(m_{i+7}))\\
&\oplus P(m_{i+8}) \oplus P(m_{i+6}) \oplus P'(m_{i+8}) \oplus P'(m_{i+2}) \oplus Q(m_{i+4})\\
&\oplus Q(m_{i+7}) \oplus Q'(m_{i+4}) \oplus Q'(m_{i+6}), && (21)\\
Q(m_i) =&r_0 \oplus r_3 \oplus s_{11} \oplus s_5 \oplus s_3\\
&\oplus \Delta_{Y_{i+4}^{(0)}}(s_5 \oplus \Delta_{Y_{i+5}^{(0)}}(s_7 \oplus \Delta_{Y_{i+6}^{(0)}}(\tilde{s}_9 \oplus Q'(m_{i+8})) \oplus Q'(m_{i+7})) \oplus P(m_{i+8}) \oplus Q'(m_{i+6}))\\
&\oplus P(m_{i+5}) \oplus P(m_{i+7}) \oplus P'(m_{i+1}) \oplus P'(m_{i+7})\\
&\oplus Q(m_{i+8}) \oplus Q(m_{i+3}) \oplus Q(m_{i+6}) \oplus Q'(m_{i+3}) \oplus Q'(m_{i+5}), && (22)
\end{aligned}$$

where $\tilde{s}_n$'s and $\tilde{r}_0$ are defined as follows.

$$\begin{gathered}
\tilde{s}_{10} = s_{10} \oplus \Delta_{X_{i+8}^{(0)}}(r_0), \quad \tilde{s}_8 = s_8 \oplus \Delta_{X_{i+7}^{(0)}}(\tilde{s}_{10}), \quad \tilde{s}_6 = s_6 \oplus \Delta_{X_{i+6}^{(0)}}(\tilde{s}_8),\\
\tilde{s}_4 = s_4 \oplus \Delta_{X_{i+5}^{(0)}}(\tilde{s}_6), \quad \tilde{s}_2 = s_2 \oplus \Delta_{X_{i+4}^{(0)}}(\tilde{s}_4), \quad \tilde{s}_0 = s_0 \oplus \Delta_{X_{i+3}^{(0)}}(\tilde{s}_2),\\
\tilde{r}_2 = r_2 \oplus s_{10} \oplus s_7 \oplus \Delta_{X_{i+2}^{(0)}}(\tilde{s}_0), \quad \tilde{r}_0 = r_0 \oplus s_8 \oplus s_5 \oplus \Delta_{X_{i+1}^{(0)}}(\tilde{r}_2),\\
\tilde{s}_{11} = s_{11} \oplus \Delta_{Y_{i+8}^{(0)}}(r_1), \quad \tilde{s}_9 = s_9 \oplus \Delta_{Y_{i+7}^{(0)}}(\tilde{s}_{11}).
\end{gathered}$$

Before proving the theorem, we show that the equations (14)-(22) are easily solved for $m_i, \cdots, m_{i+8}$. First, m_{i+8} is determined by the equation (14). Then, m_{i+7} is determined by the equation (15) because m_{i+8} is already determined. Like this, m_{i+6}, m_{i+5}, m_{i+4}, m_{i+3}, m_{i+2}, m_{i+1} and m_i are determined by the equations (16), (17), (18), (19), (20), (21) and (22), respectively and sequentially.
Now, we prove the theorem 2.

Proof. First, using the equation of the inverse update (12) nine times, we can express the internal state difference before the $\lambda + 1$-th step, σ_i, by σ_{i+9} and $m_i, \cdots, m_8$. Especially, $b[0]_i$,

$k[10]_i$, $k[8]_i$, $k[6]_i$, $k[4]_i$, $k[2]_i$, $k[0]_i$, $b[2]_i$, $k[7]_i$ and $k[5]_i$ is expressed as the follows:

$$
\begin{aligned}
b[0]_i =& \tilde{s}_{10} \oplus s_6 \oplus s_3 \oplus \Delta_{X_i^{(0)}}(\tilde{r}_0) \oplus Q(m_{i+8}), && (23)\\
k[10]_i =& \tilde{s}_8 \oplus s_4 \oplus s_1 \oplus P'(m_{i+8}) \oplus Q(m_{i+7}), && (24)\\
k[8]_i =& s_8 \oplus \tilde{s}_6 \oplus s_2 \oplus r_3 \oplus s_{11} \oplus P'(m_{i+7}) \oplus Q(m_{i+6}), && (25)\\
k[6]_i =& s_6 \oplus \tilde{s}_4 \oplus s_0 \oplus r_1 \oplus s_9 \oplus P'(m_{i+6}) \oplus Q(m_{i+8}) \oplus Q'(m_{i+8}) \oplus Q(m_{i+5}), && (26)\\
k[4]_i =& r_2 \oplus s_{10} \oplus s_4 \oplus \tilde{s}_2 \oplus \tilde{s}_{11} \oplus P'(m_{i+5}) \oplus Q(m_{i+7}) \oplus Q'(m_{i+7}) \oplus Q(m_{i+4}), && (27)\\
k[2]_i =& r_0 \oplus s_8 \oplus s_2 \oplus \tilde{s}_0 \oplus \tilde{s}_9 \oplus P(m_{i+8}) \oplus P'(m_{i+4}) \oplus Q(m_{i+6}) \oplus Q'(m_{i+8}) \oplus Q'(m_{i+6}) \oplus Q(m_{i+3}), && (28)\\
k[0]_i =& \tilde{r}_2 \oplus \tilde{s}_{10} \oplus s_6 \oplus s_0 \oplus s_7 \oplus \Delta_{Y_{i+6}^{(0)}}(\tilde{s}_9 \oplus Q'(m_{i+8})) \oplus P(m_{i+7}) \oplus P'(m_{i+3}) \\
& \oplus Q(m_{i+5}) \oplus Q'(m_{i+5}) \oplus Q'(m_{i+7}) \oplus Q(m_{i+2}), && (29)\\
b[2]_i =& \tilde{r}_0 \oplus P(m_i), && (30)\\
k[7]_i =& r_2 \oplus s_{10} \oplus s_7 \oplus s_5 \oplus s_1 \oplus \Delta_{Y_{i+5}^{(0)}}(s_7 \oplus \Delta_{Y_{i+6}^{(0)}}(\tilde{s}_9 \oplus Q'(m_{i+8})) \oplus Q'(m_{i+7})) \oplus P(m_{i+8}) \\
& \oplus P(m_i) \oplus P(m_{i+6}) \oplus P'(m_{i+8}) \oplus P'(m_{i+2}) \oplus Q(m_{i+4}) \\
& \oplus Q(m_{i+7}) \oplus Q'(m_{i+4}) \oplus Q'(m_{i+6}) \oplus Q(m_{i+1}), && (31)\\
k[5]_i =& r_0 \oplus r_3 \oplus s_{11} \oplus s_5 \oplus s_3 \\
& \oplus \Delta_{Y_{i+4}^{(0)}}(s_5 \oplus \Delta_{Y_{i+5}^{(0)}}(s_7 \oplus \Delta_{Y_{i+6}^{(0)}}(\tilde{s}_9 \oplus Q'(m_{i+8})) \oplus Q'(m_{i+7})) \oplus P(m_{i+8}) \oplus Q'(m_{i+6})) \\
& \oplus P(m_{i+5}) \oplus P(m_{i+7}) \oplus P'(m_{i+1}) \oplus P'(m_{i+7}) \\
& \oplus Q(m_{i+8}) \oplus Q(m_{i+3}) \oplus Q(m_{i+6}) \oplus Q'(m_{i+3}) \oplus Q'(m_{i+5}) \oplus Q(m_i). && (32)
\end{aligned}
$$

From the equations (14) and (23), the equation $b[0]_i = 0$ follows. Similarly, $k[10]_i = 0$ is derived from the equations (15) and (24), $k[8]_i = 0$ from the equations (16) and (25), $k[6]_i = 0$ from the equations (17) and (26), $k[4]_i = 0$ from the equations (18) and (27), $k[2]_i = 0$ from the equations (19) and (28), $k[0]_i = 0$ from the equations (20) and (29), $b[2]_i = k[7]_i$ from the equations (21), (30) and (31), and $k[5]_i = 0$ from the equations (22) and (32). □

Then, we can obtain the desired messages by the following procedure.

step 2-1 Starting with the final internal state difference $\sigma_l = 0$, we update reversely the internal state using $(l - \lambda - 9)$ message block differences $m_{l-1}, m_{l-2}, \cdots, m_{\lambda+9}$ and obtain $\sigma_{\lambda+9}$. In this step, we can choose freely the message block differences $m_{\lambda+9}, m_{\lambda+10}, \cdots, m_{l-1}$, except for the padding segment of m_{l-1}.

step 2-2 We determine message block differences $m_{\lambda+8}, \cdots m_\lambda$ by using theorem 2 with $i = \lambda$ and obtain the internal state difference σ_λ satisfying equation (4). Then, a message candidate is obtained by xoring $M_\lambda^{(0)}|| \cdots ||M_{l-1}^{(0)}$ with $m_\lambda|| \cdots ||m_{l-1}$.

The number of message difference $m_{\lambda+9}|| \cdots ||m_{l-1}$ that we can choose at step 2-1, is greater than 2^{448} because $(l - \lambda - 9) \geq 5$. Therefore, we can build 2^{448} message candidates.

In this procedure, $(l - \lambda)$ inverse step function evaluations is required to build a message candidate and the corresponding internal state difference. An evaluation of the inverse step function takes the same time as the step function does. Thus, building V_2, which consists of 2^{448} pairs of a message difference and the corresponding internal state difference, requires $(l - \lambda) \times 2^{448}$ evaluations of the step function.

3 Conclusion

In this note, We presented a second preimage attack on SHAMATA-512. The attack uses differential paths that hold with a probability one and a meet-in-the-middle approach to find second preimages. The time complexity is about $2^{451.7}$ computation of the step function and the memory complexity is about $2^{452.7}$ blocks of 128 bits.

Acknowledgements

The authors would like to be thank Yasuko Fukuzawa, Toru Owada, Hisayoshi Sato, Hirotaka Yoshida and Yasuo Hatano for useful comments.

References

[1] Adem Atalay, Orhun Kara, Ferhat Karakoc and Cevat Manap, “SHAMATA HASH FUNCTION ALGORITHM SPECIFICATIONS,” 2008.