Supplementary Material

April 13, 2023

1 Improved MitM Attack on Ascon-XOF

For practical attacks on Ascon, we also apply our tools on Ascon-XOF with 64-bit hash in two tweaked settings given by the designers of Ascon [1]. In the setting of equivalent IV=0 and and one 64-bit word, we reduce time of the 2-round preimage attack from 2^{39} [1] to $2^{33.16}$, and also achieve the first 3-round preimage attack with $2^{54.33}$ in Sect. 1.1. When increasing to 3-word rate as [1], we reduce the 3-round preimage attack from 2^{48} [1] to 2^{33}, and also achieve the first 4-round preimage attack with $2^{53.59}$ in Sect. 1.2. We also give an experiment of 2-round MitM collision attack on Ascon-XOF with 64-bit hash in Sect. 1.3, following the setting given by the designers in [1], i.e., equivalent IV=0 and one 64-bit word rate.

For ease of reading, we recall the round function of Ascon, which consists of three operations: constant addition p_C, non-linear Sbox p_S and linear layer p_L. Denote the internal states of round r as $A^{(r)} \xrightarrow{p_S \circ p_C} S^{(r)} \xrightarrow{p_L} A^{(r+1)}$. The Sbox for Ascon maps $(a_0, a_1, a_2, a_3, a_4) \in \mathbb{F}_2^5$ to $(b_0, b_1, b_2, b_3, b_4) \in \mathbb{F}_2^5$, and the algebraic normal form (ANF) of the Sbox is listed as follows:

$$\begin{aligned}
b_0 &= a_4a_1 + a_3 + a_2a_1 + a_2 + a_1a_0 + a_1 + a_0,\\
b_1 &= a_4 + a_3a_2 + a_3a_1 + a_3 + a_2a_1 + a_2 + a_1 + a_0,\\
b_2 &= a_4a_3 + a_4 + a_2 + a_1 + 1,\\
b_3 &= a_4a_0 + a_4 + a_3a_0 + a_3 + a_2 + a_1 + a_0,\\
b_4 &= a_4a_1 + a_4 + a_3 + a_1a_0 + a_1.
\end{aligned} \tag{1}$$

1.1 Improved MitM preimage attack on Ascon-XOF

In the setting with an all-zero equivalent IV and for a rate of 64, we consider the practical MitM preimage attacks on Ascon-XOF with a 64-bit hash value and a 64-bit security claim against preimage attack. The paddings and round constants are also omitted for simplicity as [1] . Applying our automatic tools, we find an improved 2-round preimage attack and the first 3-round preimage attack on Ascon-XOF in this setting.

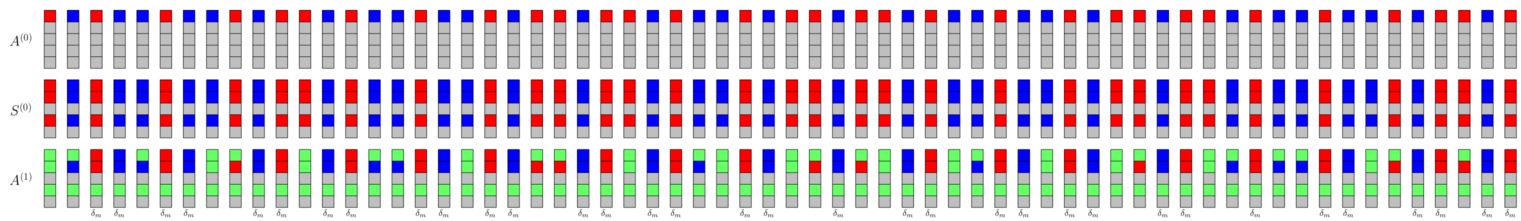

Figure 1: The MitM preimage attack on 2-round `Ascon-XOF`

1.1.1 Improved MitM preimage attack on 2-round `Ascon-XOF`

The 2-round MitM preimage attack on `Ascon-XOF` is given in Figure 1. In the starting state $A^{(0)}$, the 256-bit outer part $\{A^{0)}_{\{*,1\}}, A^{0)}_{\{*,2\}}, A^{0)}_{\{*,3\}}, A^{0)}_{\{*,4\}}\}$ are all zeros, which are marked by ⬛. The 64 bits inner part $A^{(0)}_{\{*,0\}}$ have 32 bits 🟥 and 32 bits 🟦. Then after the first substitution layer p_S, we have

$$\begin{cases} S^{(0)}_{\{z,0\}} = S^{(0)}_{\{z,1\}} = S^{(0)}_{\{z,3\}} = A^{(0)}_{\{z,0\}}, \\ S^{(0)}_{\{z,2\}} = 1, \\ S^{(0)}_{\{z,4\}} = 0, \end{cases} \tag{2}$$

where $0 \leq z \leq 63$. Then after the linear layer p_L, we can get $A^{(1)}$ for matching, where $m = 34$. In the computation from $A^{(0)}$ to $A^{(1)}$, there is no DoF of 🟥 bits and 🟦 bits consumed. Therefore, $\text{DoF}_{\mathcal{R}} = 32$, $\text{DoF}_{\mathcal{B}} = 32$. We use one message block to conduct the MitM attack. The 2-round MitM preimage attack is given in Algorithm 1. A space of 2^{32+32} is traversed to find a 64-bit preimage.

Algorithm 1: Improved Preimage Attack on 2-round `Ascon-XOF`

---

1 Inversely precompute $S^{(1)}_{\{*,0\}}$ with the first 64-bit hashing value
2 Compute forward to determine the 34-bit matching point with fixing both 🟥 and 🟦 in $A^{(0)}$ as 0, *i.e.*, compute 34 $f'''_{\mathcal{M}} = f_{\mathcal{G}}$.
3 Traversing the $2^{\lambda_{\mathcal{R}}} = 2^{32}$ values for 🟥 in $A^{(0)}$ while fixing 🟦 as 0, compute forward to determine the 34-bit matching point, *i.e.*, compute 34 $f'_{\mathcal{M}} = f_{\mathcal{R}} \oplus f_{\mathcal{G}}$. Build the table L_1 and store the 32 bits 🟥 of $A^{(0)}$, which is indexed by the 34-bit matching point.
4 Traversing the $2^{\lambda_{\mathcal{B}}} = 2^{32}$ values for 🟦 in $A^{(0)}$ while fixing 🟥 as 0, compute forward to determine the 34-bit matching point, *i.e.*, compute 34 $f''_{\mathcal{M}} = f_{\mathcal{B}} \oplus f_{\mathcal{G}}$. Build the table L_2 and store the 32 bits 🟦 of $A^{(0)}$, which is indexed by the 34-bit matching point.
5 **for** *values matched between* L_1 *and* L_2 **do**
6   **if** *it leads to the given hash value* **then**
7   Output the preimage
8   **end**
9 **end**

The time complexity of steps in Alg. 1 are analyzed below:

- In Line 3, the time complexity is 2^{32} 2-round `Ascon`.
- In Line 4, the time complexity is 2^{32} 2-round `Ascon`.
- In Line 5, the time is $2^{32+32-34} = 2^{30}$ 2-round `Ascon`.

The total time complexity is $2^{32}+2^{32}+2^{30} \approx 2^{33.16}$ 2-round `Ascon`. The memory is 2^{33} to store L_1 and L_2.

1.1.2 Improved MitM preimage attack on 3-round `Ascon-XOF`

In the same setting, we also find the first 3-round preimage attack on `Ascon-XOF`, as shown in Figure 2. The starting state $A^{(0)}$ have 41 bits ■ and 11 bits ■. In the computation from $A^{(0)}$ to $A^{(2)}$, the consumed DoFs of ■ are 29 and there is no DoF of ■ consumed. Therefore, $\mathrm{DoF}_{\mathcal{R}} = 41 - 29 = 12$, $\mathrm{DoF}_{\mathcal{B}} = 11$. We have 11-bit matching point in $A^{(2)}$ ($m = 11$).

The attack procedure is given in Alg. 2, and the time complexities of steps are analyzed below:

- In Line 4, the time is $2^{12+41} = 2^{53}$ 3-round `Ascon`.
- In Line 8, since U stores 41 ■ bits and 11-bit matching point, building L_1 is just to retrieve the values in $U[c_{\mathcal{R}}]$. Assume one table access is about one Sbox application, the time of Line 8 is $2^{12+29+12} \times \frac{1}{192} = 2^{53} \times 2^{-7.58} = 2^{45.42}$ 3-round `Ascon`.
- In Line 11, the time is $2^{12+29+11} = 2^{52}$ 3-round `Ascon`.
- In Line 14, the time is $2^{12+29+12+11-11} = 2^{53}$ 3-round `Ascon`.

The total time complexity is $2^{53} + 2^{45.42} + 2^{52} + 2^{53} \approx 2^{54.33}$ 3-round `Ascon`. The memory is 2^{41} to store U.

1.2 Improved MitM preimage attack on `Ascon-XOF` with increased rate

We also consider the attacks on `Ascon-XOF`, where the rate is increased to the first 3 words as [1]. We target on 3-/4-round `Ascon-XOF` with a 64-bit hash value and a 64-bit security claim against preimage attack. The equivalent IV is set to 0. For simplicity, the attacks don't consider the constant addition and the paddings.

1.2.1 Improved MitM preimage attack on 3-round `Ascon-XOF` with 3-word rate

The 3-round MitM preimage attack on `Ascon-XOF` with 3-block rate is given in Figure 3. In the starting state $A^{(0)}$, we have $A^{(0)}_{\{z,3\}} = A^{(0)}_{\{z,4\}} = 0 (0 \le z \le 63)$, due to the $IV = 0$ which are marked by ■. Furthermore, we choose an initial structure, where $A^{(0)}_{\{z,0\}} = A^{(0)}_{\{z,2\}} + c_z = x_z + c_z$ marked by ■/■ and $A^{(0)}_{\{z,1\}} = 0$ marked by ■. The x_z is binary variable and c_z is constant for $0 \le z \le 63$. Thus, after the first round, we get the following structure as Equ. (3):

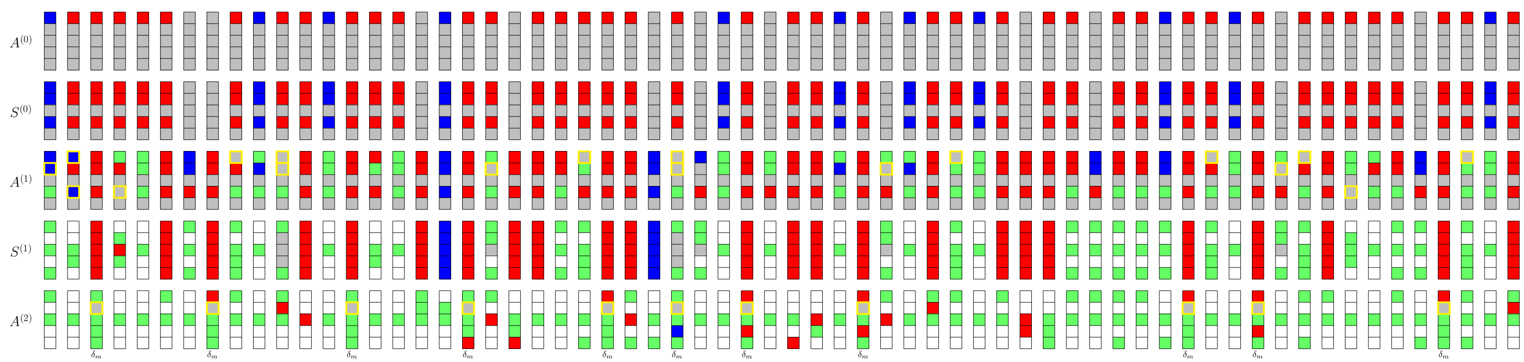

Figure 2: The MitM preimage attack on 3-round `Ascon-XOF`

Algorithm 2: MitM Preimage Attack on 3-round `Ascon-XOF`

Inversely precompute $S^{(2)}_{\{*,0\}}$ with the first 64-bit hashing value

for 2^{12} *values of the* ■ *bits in* $A^{(0)}$ **do**

  Traversing the $2^{\lambda_{\mathcal{R}}} = 2^{41}$ values for ■ in $A^{(0)}$ while fixing ■ as 0, compute forward to determine the 29-bit ■/■ (denoted as $c_{\mathcal{R}} \in \mathbb{F}_2^{29}$), and the 11-bit matching point, *i.e.*, compute 11 $f'_{\mathcal{M}} = f_{\mathcal{R}} \oplus f_{\mathcal{G}}$. Build the table U and store the 41-bit ■ of $A^{(0)}$ as well as the 11-bit matching point in $U[c_{\mathcal{R}}]$.

  for $c_{\mathcal{R}} \in \mathbb{F}_2^{29}$ **do**

    Randomly pick a 41-bit ■ $e \in U[c_{\mathcal{R}}]$, and set ■ in $A^{(0)}$ as 0, compute to the matching point to get 11 $f'''_{\mathcal{M}} = f_{\mathcal{G}} + Const(e)$

    for 2^{12} *values in* $U[c_{\mathcal{R}}]$ **do**

      Restore the values of ■ of $A^{(0)}$ and the corresponding 11-bit matching point (*i.e.*, 11 $f_{\mathcal{R}} \oplus f_{\mathcal{G}} = f'_{\mathcal{M}}$) in a list L_1 indexed by the matching point

    end

    for 2^{11} *values of* ■ **do**

      Set the 41-bit ■ in $A^{(0)}$ as e. Compute to the matching point to get 11 $f''_{\mathcal{M}} = f_{\mathcal{B}} + f_{\mathcal{G}} + Const(e)$. Together with $f'''_{\mathcal{M}}$, compute $f_{\mathcal{B}} = f''_{\mathcal{M}} + f'''_{\mathcal{M}}$ and store ■ in L_2 indexed by the 11-bit matching point

    end

    for *values matched between* L_1 *and* L_2 **do**

      if *it leads to the given hash value* **then**

        Output the preimage

      end

    end

  end

end

$$\begin{cases} A^{(1)}_{\{z,0\}} = c_z + c_{z-19} + c_{z-28}, \\ A^{(1)}_{\{z,1\}} = c_z + c_{z-61} + c_{z-39}, \\ A^{(1)}_{\{z,2\}} = x_z + x_{z-1} + x_{z-6} + 1, \\ A^{(1)}_{\{z,3\}} = c_z + c_{z-10} + c_{z-17}, \\ A^{(1)}_{\{z,4\}} = 0, \end{cases} \tag{3}$$

where $0 \le z \le 63$ and the computation of z is modular 64. Additional, in the second round, we can add the constraint $A^{(1)}_{\{z,1\}} + A^{(1)}_{\{z,3\}} + 1 = 0$ to make $S^{(1)}_{\{z,1\}}$ to be a constant, since $b_1 = (a_3 + a_1 + 1)a_2 + a_4 + a_3a_1 + a_3 + a_1 + a_0$. Therefore, all $A^{(2)}_{\{z,1\}}$ will be constants and get corresponding matching points.

Since we set $A^{(0)}_{\{z,2\}} = A^{(0)}_{\{z,0\}} + c_z = x_z + c_z$, the starting state $A^{(0)}$ only contains 32 free ■ bits and 32 free ■ bits. Without consuming DoF in the following computation, we have $\mathrm{DoF}_{\mathcal{B}} = 32$, $\mathrm{DoF}_{\mathcal{R}} = 32$. Totally, we get a 64-bit matching point.The attack procedure is given in Alg. 3, and the time complexities of steps are analyzed below:

- In Line 2, the time of solving the linear system is $64^3 = 2^{18}$ bit operations.
- In Line 3, the time complexity of computing $f'''_{\mathcal{M}}$ is 1 3-round `Ascon`.
- In Line 5, the time complexity of computing $f'_{\mathcal{M}}$ is 2^{32} 3-round `Ascon`.
- In Line 8, the time complexity of computing $f''_{\mathcal{M}}$ is 2^{32} 3-round `Ascon`.
- In Line 11, the time is $2^{32+32-64} = 1$ 3-round `Ascon`.

The total time complexity is $2^{18} + 2^{32} + 2^{32} + 2 \approx 2^{33}$ 3-round `Ascon`. The memory is 2^{33} to store L_1 and L_2.

1.2.2 MitM preimage attack on 4-round `Ascon-XOF` with 3-word rate

We also find an MitM preimage attack on 4-round `Ascon-XOF` with 3-word rate as shown in Figure 4. The starting state $A^{(0)}$ contains 12 free ■ bits and 49 free ■ bits due to the initial structure. In the computation from $S^{(0)}$ to $A^{(3)}$, the accumulated consumed DoF of ■ is 37 and the accumulated consumed DoF of ■ is 0. Therefore, $\mathrm{DoF}_{\mathcal{B}} = 12$, $\mathrm{DoF}_{\mathcal{R}} = 12$. There is a 12-bit matching in $A^{(3)}$, i.e., $m = 12$. The details of the attack are given in Alg. 4.

The time complexities of steps in Alg. 4 are analyzed below:

- In Line 2, the time of solving the linear system is $64^3 = 2^{18}$ bit operations.
- In Line 5, the time is $2^{3+49} = 2^{52}$ 4-round `Ascon`.

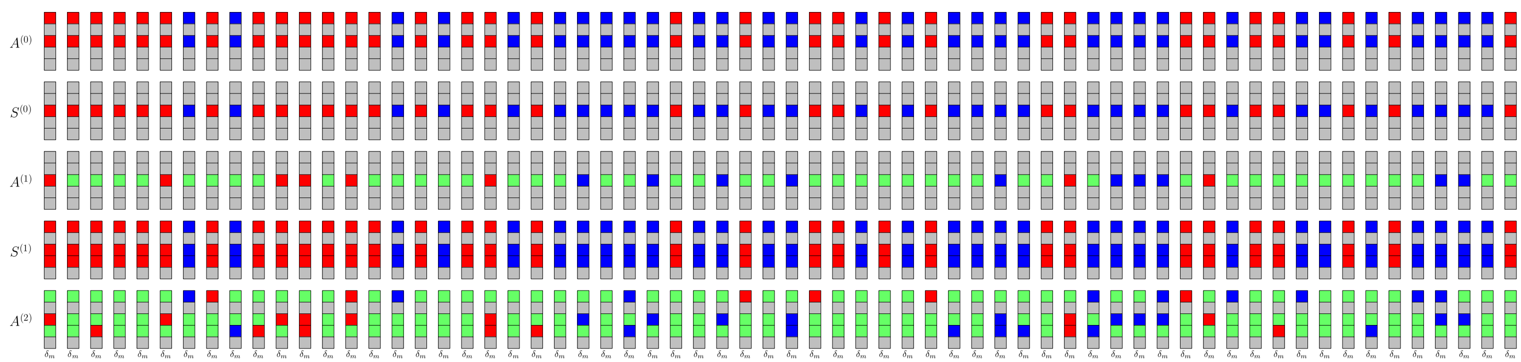

Figure 3: The MitM preimage attack on 3-round `Ascon-XOF` with 3-word rate

Algorithm 3: Improved Preimage Attack on 3-round `Ascon-XOF` for a rate of 192

```
Inversely precompute S^{(2)}_{{*,0}} with the first 64-bit hashing value
```

Inversely precompute $S^{(2)}_{\{*,0\}}$ with the first 64-bit hashing value

Compute the constants satisfying $A^{(1)}_{\{z,1\}} + A^{(1)}_{\{z,3\}} + 1 = 0,\ 0 \leq z \leq 63$ according Equ. (3).

Compute forward to determine the 64-bit matching point with fixing both ${\color{red}\blacksquare}$ and ${\color{blue}\blacksquare}$ in $A^{(0)}_{\{z,2\}}$, *i.e.* as 0, compute 64 $f'''_{\mathcal{M}} = f_{\mathcal{G}}$.

for 2^{32} *values of* ${\color{red}\blacksquare}$ **do**

  With fixing ${\color{red}\blacksquare}$ in $A^{(0)}_{\{z,2\}}$ as 0, compute forward to determine the 64-bit matching point, *i.e.*, compute 64 $f'_{\mathcal{M}} = f_{\mathcal{R}} \oplus f_{\mathcal{G}}$. Build the table L_1 and store the 32-bit ${\color{red}\blacksquare}$ of $A^{(0)}$ indexed by the 64-bit matching point $f'_{\mathcal{M}}$.

end

for 2^{32} *values of* ${\color{blue}\blacksquare}$ **do**

  Set the 32-bit ${\color{red}\blacksquare}$ in $A^{(0)}_{\{z,2\}}$ as 0. Compute to the matching point to get 64 $f''_{\mathcal{M}} = f_{\mathcal{B}} + f_{\mathcal{G}}$ and store ${\color{blue}\blacksquare}$ in L_2 indexed by the 64-bit matching point

end

for *values matched between* L_1 *and* L_2 *(*$f_{\mathcal{M}} = f'_{\mathcal{M}} \oplus f''_{\mathcal{M}} \oplus f'''_{\mathcal{M}}$*)* **do**

  if *it leads to the given hash value* **then**

    Output the preimage

  end

end

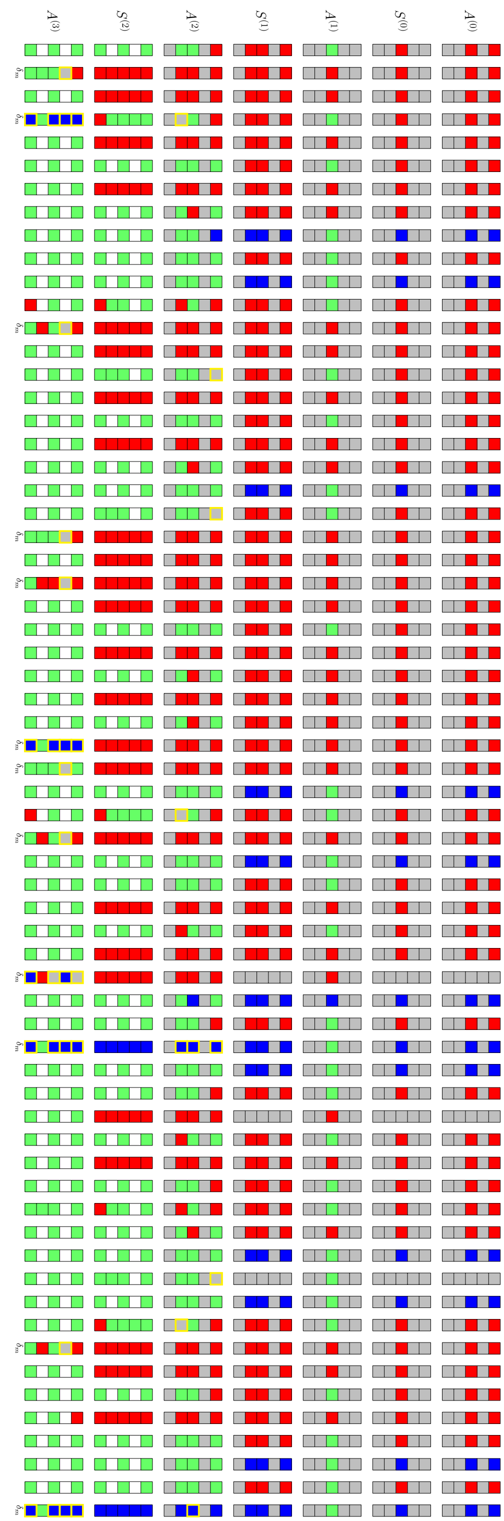

Figure 4: The MitM preimage attack on 4-round `Ascon-XOF` with 3-word rate

Algorithm 4: MitM Preimage Attack on 4-round `Ascon-XOF` for a rate of 192

Inversely precompute $S^{(3)}_{\{*,0\}}$ with the 64-bit hashing value

Compute the constants satisfying $A^{(1)}_{\{z,1\}} + A^{(1)}_{\{z,3\}} + 1 = 0,\ 0 \le z \le 63$ according Equ. (3).

for 2^3 *values of the* ■ *bits in* $A^{(0)}_{\{*,2\}}$

do

  Traversing the $2^{\lambda_{\mathcal{R}}} = 2^{49}$ values for ■ in $A^{(0)}$ while fixing ■ as 0, compute forward to determine the 37-bit ■/■ (denoted as $c_{\mathcal{R}} \in \mathbb{F}_2^{37}$), and the 12-bit matching point, *i.e.*, compute 12 $f'_{\mathcal{M}} = f_{\mathcal{R}} \oplus f_{\mathcal{G}}$. Build the table U and store the 49-bit ■ of $A^{(0)}$ as well as the 12-bit matching point in $U[c_{\mathcal{R}}]$.

  for $c_{\mathcal{R}} \in \mathbb{F}_2^{37}$ **do**

    Randomly pick a 49-bit ■ $e \in U[c_{\mathcal{R}}]$, and set ■ in $A^{(0)}$ as 0, compute to the matching point to get 12 $f'''_{\mathcal{M}} = f_{\mathcal{G}} + Const(e)$

    for 2^{12} *values in* $U[c_{\mathcal{R}}]$ **do**

      Restore the values of ■ of $A^{(0)}$ and the corresponding 12-bit matching point (*i.e.*, 12 $f_{\mathcal{R}} \oplus f_{\mathcal{G}} = f'_{\mathcal{M}}$) in a list L_1 indexed by the matching point

    end

    for 2^{12} *values of* ■ **do**

      Set the 49-bit ■ in $A^{(0)}$ as e. Compute to the matching point to get 12 $f''_{\mathcal{M}} = f_{\mathcal{B}} + f_{\mathcal{G}} + Const(e)$. Together with $f'''_{\mathcal{M}}$, compute $f_{\mathcal{B}} = f''_{\mathcal{M}} + f'''_{\mathcal{M}}$ and store ■ in L_2 indexed by the 12-bit matching point

    end

    for *values matched between* L_1 *and* L_2 **do**

      if *it leads to the given hash value* **then**

        Output the preimage

      end

    end

  end

end

- In Line 9, since U stores 49 ■ bits and 12-bit matching point, building L_1 is just to retrieve the values in $U[c_{\mathcal{R}}]$. Assume one table access is about one Sbox application, the time of Line 9 is $2^{3+37+12} \times \frac{1}{256} = 2^{52} \times 2^{-8} = 2^{44}$ 4-round `Ascon`.

- In Line 12, the time is $2^{3+37+12} = 2^{52}$ 4-round `Ascon`.

- In Line 15, the time is $2^{3+37+12+12-12} = 2^{52}$ 4-round `Ascon`.

The total time complexity is $2^{18} + 2^{52} + 2^{44} + 2^{52} + 2^{52} \approx 2^{53.59}$ 3-round `Ascon`. The memory is 2^{49} to store U.

1.3 An experiment on 2-round collision attack on `Ascon-XOF`

To verify the correctness, we give a collision attack on 2-round `Ascon-XOF` with 64-bit hash, following the setting given by the designers, i.e., equivalent IV=0 and one 64-bit word rate. The round constants and paddings are also omitted. The attack is shown in Figure 5. $A^{(0)}$ contains 17 ■ and 17 ■, and there are 16 matching points, where $\text{DoF}_{\mathcal{B}} = 17$, $\text{DoF}_{\mathcal{R}} = 17$ and $m = 16$. In the practical attack, we omit the linear layer p_L in the last round for simplicity. That is, we regard $S^{(1)}_{\{*,0\}}$ as the hash value.

Without loss of generality, we set the 16-bit partial target to all-zero. Then we need 2^{24} different M with the same fixed 16-bit partial target. Since each MitM episode can produce 2^{18} partial target preimages, we need repeat 2^6 MitM episodes. The theoretical time is $2^6 \cdot (2^{17} + 2^{17} + 2^{18}) = 2^{25}$, while the time of exhaustive search is 2^{32}. The memory complexity is 2^{24}.

In each episodes, we traverse the 2^{17} ■ and 2^{17} ■ and set the other 30 bits ■ of $A^{(0)}_{\{*,0\}}$ to be random value. In our practical experiment, when we set the number of MitM episodes to 2^5, we get 2^{26} partial target preimages. Among them we find one collision. The experiment is very close to our expectation (one collision among the 2^{24} partial target preimages). Testing for several times, we get some collision examples, which are listed in Table 1.

References

[1] Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Preliminary analysis of Ascon-Xof and Ascon-Hash (2019), https://ascon.iaik.tugraz.at,

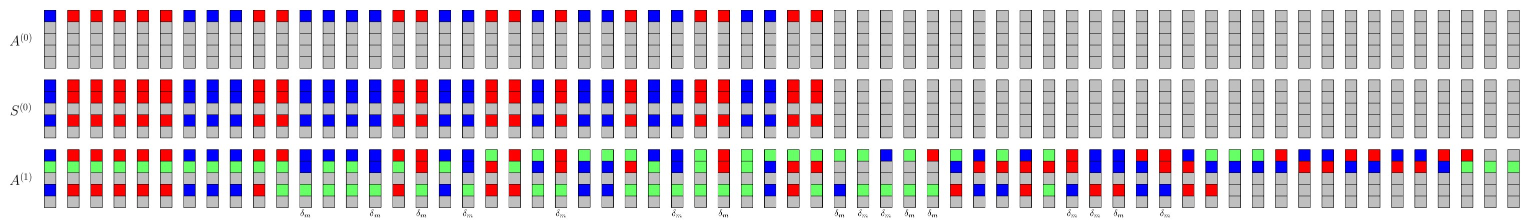

Figure 5: The MitM preimage attack on 4-round `Ascon-XOF` with 3-word rate

Table 1: Collision examples of 2-round `Ascon-XOF` for a rate of 64.

Round	Message	Hash
2	63dc48f8a38448f3 63d2c3eca38448f3	d60459ea403147dc
	7e311ec288cef6e5 583a0ae208cef6e5	67e1582200f036c4
	83cdd944c8cef6e5 4a9bf94588cef6e5	7fec154b80a07f40
	1edc991f6598e891 25e74993dee08bd1	b92c4d688121434c
	84b259191785824b e02b4a331785824b	64294d0b81b0161c
	6d422bfd7d7a4e09 4b006fdc372f9836	35e45468002115b8
	ed422bfd7d7a4e09 cb006fdc372f9836	b5c40468002115b8
	5ef61d6b9a91f2e4 49edf09c41819d60	60e858480161322c