Preimage attack on GAGE variants

Nasour Bagheri[1], Sadegh Sadeghi[2]

[1] Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran, NBagheri@sru.ac.ir

[2] Department of Mathematics, Faculty of Mathematical Sciences and Computer, Kharazmi University, Tehran, Iran, s.sadeghi.khu@gmail.com

Abstract. GAGE is a candidate of the first round of the NIST lightweight cryptography competition. It is a lightweight Sponge based hash function, supports different sets of parameters. However, the hash length is always 256 bits. For example, the designers' claimed security against preimage attack is 2^{256} when the rate is 128 bits and the capacity is 256 bits. However, in this note, we show that the security for this parameter set is 2^{128}.

Keywords: NIST lightweight cryptography competition, GAGE, Preimage attack

1 Introduction

A cryptographical hash function maps any message of arbitrary length to a string of specific length, e.g. n bits, where the output string is known as the message digest or hash value. More formally, we can define a hash function as follows:

$$H : \{0,1\}^* \rightarrow \{0,1\}^n$$

Three main criteria for a secure cryptographical hash function are: preimage resistant, second-preimage resistant and collision resistant. Among them, preimage attack means that given any $h \in \{0,1\}^n$ to find a $M \in \{0,1\}^*$ such that $H(M) = h$. For an ideal hash function, which is modeled as a random oracle also, the expected complexity of finding a preimage for an n-bit hash function is 2^n.

GAGE [2], which is a candidate of the first round of the NIST lightweight cryptography competition, uses Sponge [1] based construction to produce a 256-bit hash value for any given message M. The input message is at the first padded by a string $\{80\|00^*\}$, however, it has no impact on the proposed attack in this note. GAGE supports different parameter sets that provide a different level of security. A variant of this scheme has the rate $r = 128$ bits, the capacity $c = 256$ bits and $b = r + c = 384$ bits. For $n = 256$, the security claim against preimage attack is 2^{256}. Given the message M which is padded as $M_{pad} = M_0\|M_1\|\ldots\|M_{l-2}\|M_{l-1}$ and the permutation $Q : \{0,1\}^b \rightarrow \{0,1\}^b$, a brief representation of this scheme is depicted in Figure 1 and works as follows, where $\perp$ denotes an empty string:

1. $M_{pad} \rightarrow M_0\|M_1\|\ldots\|M_{l-2}\|M_{l-1}$
2. $(S = S_r\|S_c) \leftarrow 0$
3. $H(M) \leftarrow \perp$:
4. **Absorbing Phase:** for $0 \leq i \leq l-1$ do:

(a) $(S = S_r \| S_c) \leftarrow (S_r \oplus M_i) \| S_c$
(b) $(S = S_r \| S_c) \leftarrow Q(S)$

5. **Squeezing Phase:** for $0 \leq i \leq \frac{n}{r} - 1$ do:
   (a) $(S = S_r \| S_c) \leftarrow Q(S)$
   (b) $H(M) \leftarrow H(M) \| S_r$
6. return $H(M)$

Given that for the target parameter set $n = 2 \times r$, to produce the hash value we need to call the permutation function 2 times in the squeezing phase.

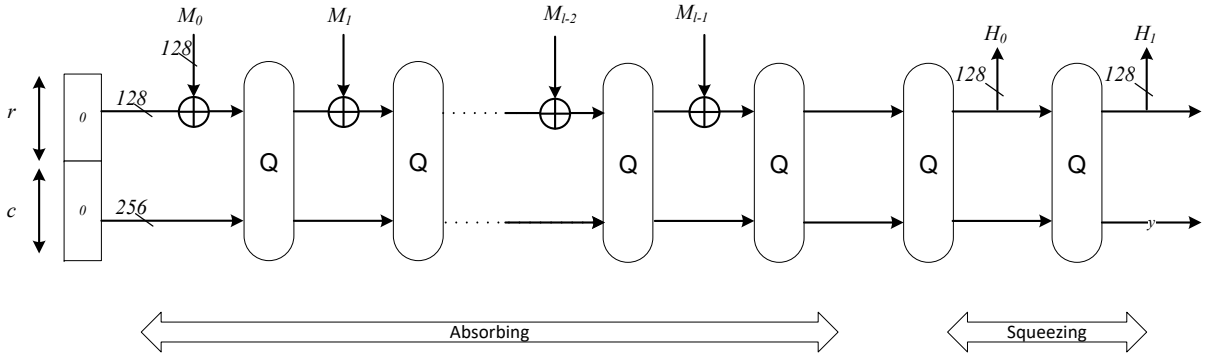

Fig. 1. The hash mode of GAGE when the rate is 128 bits and the capacity is 256 bits

In Table 1, the security claim for different parameter sets is presented.

In the next section we describe a preimage attack against GAGE, when $r = 128$ and $c = 256$, i.e., the parameter set #8 in Table 1.

2 Preimage Attack

Given $h = H_0 \| H_1$, to find a preimage in GAGE, when $r = 128$ and $c = 256$, the adversary does as follow, where Q^{-1} denotes the inverse of the permutation Q and $\{0\}^t$ denotes a t-bit zero string:

1. $S_r \leftarrow H_1$
2. $S_r^{-1} \leftarrow \perp$
3. while $S_r^{-1} \neq H_0$:
   (a) $S_c \xleftarrow{\$} \{0,1\}^{256}$
   (b) $S_r^{-1} \| S_c^{-1} \leftarrow Q^{-1}(S_r \| S_c)$
4. $S_r^{-2} \| S_c^{-2} \leftarrow Q^{-1}(S_r^{-1} \| S_c^{-1})$
5. $S_r^{-3} \| S_c^{-3} \leftarrow Q^{-1}(S_r^{-2} \| S_c^{-2})$
6. $M_3 \xleftarrow{\$} \{0,1\}^{120} \| 80$

Table 1. The claimed preimage security of all instances of GAGE [2], where for all of them $|Hash| = n = 256$ and the maximum message length is expected to be less than 2^{64} and our bounds.

#	b	c	r	Claimed [2, Sec. 1.2]: $min(n, c-1)$	Sec. 2: $min[c, n, max(\frac{c}{2}, (\frac{n}{r}-1)\times r)]$
1	232	224	8	223	224
2	240	224	16	223	224
3	256	224	32	223	224
4	288	224	64	223	192
5	272	256	16	256	240
6	288	256	32	256	224
7	320	256	64	256	192
8	384	256	128	256	128
9	544	512	32	256	256
10	576	512	64	256	256

7. $S_r^{-4} \| S_c^{-4} \leftarrow Q^{-1}((S_r^{-3} \oplus M_3) \| S_c^{-3})$
8. for $0 \leq i \leq 2^{128} - 1$ do:
   (a) $(S) \leftarrow (S_r^{-4} \oplus i) \| S_c^{-4}$
   (b) $T_{rev} \xleftarrow{StoredinTable} (S^i = Q^{-1}(S), i)$
9. for $0 \leq i \leq 2^{128} - 1$ do:
   (a) $(S) \leftarrow (\{0\}^{128} \oplus i) \| \{0\}^{256}$
   (b) $T_{dir} \xleftarrow{StoredinTable} (S^i = Q(S), i)$
10. find a record $(S^i = S_r^i \| S_c^i, i) \in T_{dir}$ and a record $(S^j = S_r^j \| S_c^j, j) \in T_{rev}$ such that $S_c^i = S_c^j$.
11. return $M = i \| (S_r^i \oplus S_r^j) \| j \| M_3$

The attack procedure is also represented in Figure 2. Given that the tables T_{rev} and T_{dir} each has the size 2^{128} and $|S_c| = 256$, we are expecting to find a matching in Step 10. Finding such matching, the rest of the attack will be straight forward. The attack complexity is dominated by Steps 3, 8 and 9, each has the complexity of 2^{128} calls to the underlying permutation Q or its reverse Q^{-1}. On the other hand, given any $M \neq \perp$, calculating the hash value costs at least three calls of Q, for the target parameter set. Hence, the total complexity is of the order 2^{128} calculation of the hash value of a message.

Remark 1. It is possible to extend the proposed attack against other variants of GAGE also. However, the complexity will be more than 2^{128}, although could be less than the claimed security by the designer, as it has been reported in Table 1. For instance, when $r = 64$ and $c = 320$, i.e., parameters set # 7, it is possible to adapt the present attack and find preimage with the complexity of 2^{192}. In general, the preimage complexity of any variant is upper-bounded by $min[c, n, max(\frac{c}{2}, (\frac{n}{r}-1)\times r)]$.

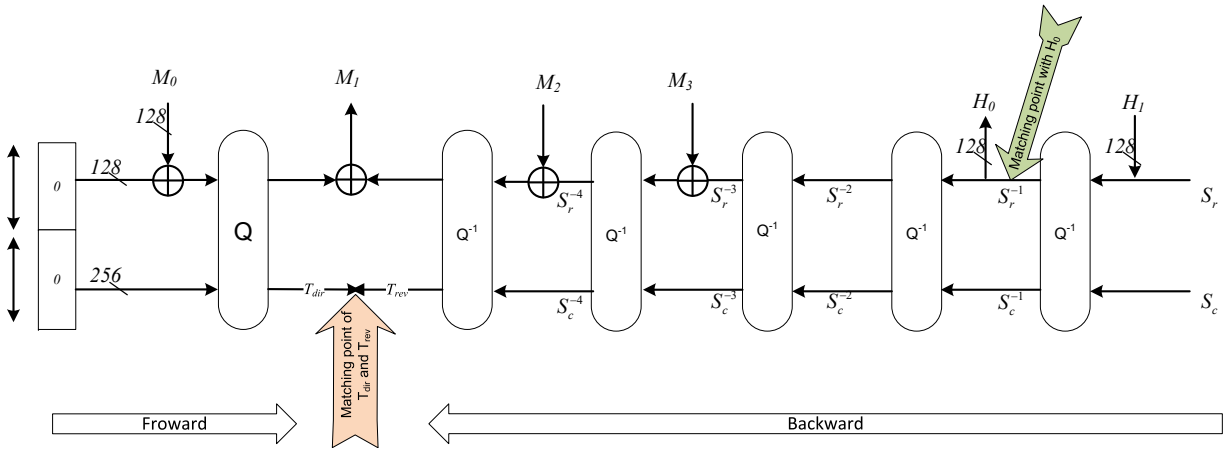

Fig. 2. Illustration of the proposed preimage attack on GAAE when the rate is 128 bits and the capacity is 256 bits

Remark 2. It should be possible to reduce the attack complexity on some variants using the idea of meet in the meddle for the backward part of the attack. However, we leave it as a future work.

3 Conclusion

In this note, we presented a preimage attack against a variant of GAGE, a candidate of the first round of the NIST competition for lightweight cryptography. The proposed attack, which is a structural attack, shows that the exact security of the variant of GAGE for which the rate is 128 bits and the capacity is 256 bits is upper-bounded by 2^{128}, much below the designer claim which is 2^{256}. We also show that the attack complexity of other variants is upper-bounded by $min[c, n, max(\frac{c}{2}, (\frac{n}{r} - 1) \times r)]$, which reduce the security bound of some of the variants of GAGE.

References

1. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Keccak. In T. Johansson and P. Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 313–314. Springer, 2013.
2. D. Gligoroski, H. Mihajloska, and D. Otte. GAGE and InGAGE. NIST, Information Technology Laboratory COMPUTER SECURITY RESOURCE CENTER, 2019. `https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/GAGEandInGAGE-spec.pdf`.