Chinese Journal of Electronics
vol. 33, no. 3, pp. 635–644, May 2024
https://doi.org/10.23919/cje.2023.00.127

RESEARCH ARTICLE

New Algebraic Attacks on Grendel with the Strategy of Bypassing SPN Steps

Wenxiao QIAO[1,2], Siwei SUN[3,4], and Lei HU[1,2]

1. *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China*
2. *School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China*
3. *School of Cryptology, University of Chinese Academy of Sciences, Beijing 100049, China*
4. *State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China*

Corresponding author: Siwei SUN, Email: siweisun.isaac@gmail.com
Manuscript Received April 13, 2023; Accepted August 7, 2023

Abstract — The rapid development of modern cryptographic applications such as zero-knowledge, secure multi-party computation, fully homomorphic encryption has motivated the design of new so-called arithmetization-oriented symmetric primitives. As designing ciphers in this domain is relatively new and not well-understood, the security of these new ciphers remains to be completely assessed. In this paper, we revisit the security analysis of arithmetization-oriented cipher Grendel. Grendel uses the Legendre symbol as a component, which is tailored specifically for the use in zero-knowledge and efficiently-varifiable proof systems. At FSE 2022, the first preimage attack on some original full GrendelHash instances was proposed. As a countermeasure, the designer adds this attack into the security analysis and updates the formula to derive the secure number of rounds. In our work, we present new algebraic attacks on GrendelHash. For the preimage attack, we can reduce the complexity or attack one more round than previous attacks for some instances. In addition, we present the first collision attack on some round-reduced instances by solving the constrained input/constrained output problem for the underlying permutations.

Keywords — Grendel, Solving univariate equation, Bypassing substitution-permutation networks steps, Preimage attack, Collision attack.

Citation — Wenxiao QIAO, Siwei SUN, and Lei HU, "New Algebraic Attacks on Grendel with the Strategy of Bypassing SPN Steps," *Chinese Journal of Electronics*, vol. 33, no. 3, pp. 635–644, 2024. doi: 10.23919/cje.2023.00.127.

I. Introduction

With the rapid development of new applications such as zero-knowledge (ZK), secure multi-party computation (MPC), fully homomorphic encryption (FHE), some efficient symmetric schemes suitable for these scenarios have been proposed in recent years. Different from traditional symmetric ciphers, the design goal of such ciphers is not to reduce execution time, energy consumption, memory footprint, etc. Instead, they optimize the algebraic complexity, i.e., they attempt to minimize the number of nonlinear operations in their natural algorithm description. Therefore, they are often referred to as arithmetization-oriented (AO) ciphers.

Since the cipher LowMC [1] was proposed at Eurocrypt 2015 by Albrecht *et al.*, which is friendly to MPC and FHE, an increasing number of AO symmetric-key primitives have been raised, including FLIP [2], Kreyvrium [3], Rasta [4], MiMC [5], Feistel-MiMC [5], GMiMC [6], Ciminion [7], Poseidon [8], and Neptune [9]. Most of them use low-degree nonlinear functions such as $x \to x^d$.

However, for some use cases of ZK proof systems, it is feasible to reach efficiency with high-degree nonlinear functions. In such a use case, the complexity of proving/verifying a given statement $y=F(x)$ for a certain function F is considered. Instead of proving/verifying $y=F(x)$ directly, proving/verifying an equivalent relation $G(x,y)=c$ may be more efficient. This approach is used in Vision [10], Rescue-Prime [11] and Grendel [12]. In Vision, a high-degree function $x \to x^{-1}$ is used. In Rescue-Prime, a low-degree function $x \to x^d$ and a high-degree function $x \to x^{\frac{1}{d}}$ are used. In Grendel, the nonlinear layer uses the Legendre symbol as a component, which is a function of high degree $\frac{p-1}{2}$ over an odd prime field $\mathbb{F}_p$.

On the other side, these AO symmetric primitives

Associate Editor: Prof. Debiao HE, Wuhan University.

propose new challenges for cryptanalysts to analyze their security. And as designing symmetric ciphers in this domain is reasonably new and not well-studied, some fatal errors may be neglected at the design phase. For instance, soon after the publication of LowMC, a higher-order differential cryptanalysis [13] and an optimized interpolation cryptanalysis [14] were given, which made LowMC move to LowMC v2. Later the difference enumeration technique [15] was proposed, which made LowMC v2 move to LowMC v3. However, the recent attacks [16]–[19] have demonstrated that some instances in LowMC v3 are still insecure.

In order to better evaluate the security of AO hash functions, Ethereum Foundation put forward a challenge[*1] to analyze the constrained input/constrained output (CICO) problem for some round-reduced versions of permutations underlying several sponge-based AO hash functions, including Feistel-MiMC, Poseidon, Rescue-Prime and Reinforced Concrete [20]. Later, the work done by Bariant *et al.* [21] showed that the security cryptanalysis presented by the challenge's authors or designers of three such primitives (i.e., Feistel-MiMC, Poseidon and Rescue-Prime) was too optimistic.

In this paper, we evaluate the security of the AO cipher Grendel [12]. Grendel defines a family of permutations, which are used to obtain hash functions by applying the sponge construction [22]–[24]. At FSE 2022, Grassi *et al.* [25] proposed the first preimage attack on some original full GrendelHash instances. This attack considered the case where the output of GrendelHash consists of one field element. It used the strategy that if let the input state only have one unknown field element (denoted by x) and all Legendre symbols in the scheme be fixed, then the output of GrendelHash can be written as a univariate polynomial in terms of x, of which the degree may be low. As a countermeasure, the designer adds this attack into the security analysis and updates the formula to derive the secure number of rounds.

Our contributions In our work, we propose three new algebraic attacks on GrendelHash and consider the case where the digest consists of one field element. In the work done by Bariant *et al.* [21] at FSE 2023, the strategy of bypassing substitution-permutation networks (SPN) steps was used to solve the CICO problem for the cryptographic permutations which are intended for use in a sponge construction to build hash functions. We give a generalization of this strategy such that it can be used to deal with more than just CICO problem. Our results are detailed as follows:

1) With bypassing one SPN step, we can find a preimage with a lower complexity than previous attacks for some round-reduced instances.

2) Regarding the special case where the sponge capacity consists of one field element, with bypassing two SPN steps, a) For the preimage attack, the complexity can be reduced further and we can attack one more round than previous attacks for some round-reduced instances; b) By solving the CICO problem for the underlying permutations, we propose the first collision attack on some round-reduced instances.

The results are summarized in Tables 1–3. The complexity of original preimage attack is derived from the formula presented in Scetion V in [25]. The complexity of designer's attack is derived from the formula presented in Table 1 in [12].

Table 1 A summary of the attacks on GrendelHash instances with parameters $\alpha = 2, \lambda = 128, \log_2(p) \approx 256$

Instance (α, m, c, R)	N	Type	Complexity	Ref.
$(2, 4, 1, 31)$	23	Preimage	$2^{138.34}$	Original [25]
			$\mathbf{2^{123.05}}$	Designer's [12]
			$\mathbf{2^{119.64}}$	Section VI.2
	24	Preimage	$2^{143.49}$	Original [25]
			$2^{128.17}$	Designer's [12]
			$\mathbf{2^{124.74}}$	Section VI.2
	24	Collision	$\mathbf{2^{124.74}}$	Section VI.3
$(2, 8, 1, 17)$	13	Preimage	$2^{134.35}$	Original [25]
			$\mathbf{2^{123.40}}$	Designer's [12]
			$\mathbf{2^{122.50}}$	Section V
			$\mathbf{2^{112.32}}$	Section VI.2
	14	Preimage	$2^{143.61}$	Original [25]
			$2^{132.61}$	Designer's [12]
			$\mathbf{2^{121.50}}$	Section VI.2
	14	Collision	$\mathbf{2^{121.50}}$	Section VI.3
$(2, 12, 1, 12)$	9	Preimage	$2^{129.05}$	Original [25]
			$\mathbf{2^{122.34}}$	Designer's [12]
			$\mathbf{2^{117.71}}$	Section V
			$\mathbf{2^{103.54}}$	Section VI.2
	10	Preimage	$2^{142.43}$	Original [25]
			$2^{135.64}$	Designer's [12]
			$\mathbf{2^{116.71}}$	Section VI.2
	10	Collision	$\mathbf{2^{116.71}}$	Section VI.3

Note: The complexity is estimated in field operations. R denotes the secure number of rounds; N denotes the number of attacked rounds; Bold numbers indicate the attacks are valid.

Organization of the paper In Section II, we give some notations and related definitions. Besides, we give a brief description of Grendel and the algorithm for solving univariate system. In Section III, we revisit the original preimage attack on GrendelHash. In Section IV, we give a generalization of the strategy of bypassing SPN steps. In Section V, we introduce our first preimage attack. In Section VI, we introduce our second preimage attack and the first collision attack. The experimental results of our attacks are given in Section VII. Finally, the paper is concluded in Section VIII.

[*1]It was published on November 1st 2021 at https://www.zkhashbounties.info/.

Table 2 A summary of the attacks on GrendelHash instances with parameters $\alpha = 3, \lambda = 128, \log_2(p) \approx 256$

Instance (α, m, c, R)	N	Type	Complexity	Ref.
$(3, 8, 1, 16)$	12	Preimage	$2^{133.70}$	Original [25]
			$\mathbf{2^{121.19}}$	Designer's [12]
			$\mathbf{2^{120.70}}$	Section V
			$\mathbf{2^{109.92}}$	Section VI.2
	13	Preimage	$2^{143.56}$	Original [25]
			2^{131}	Designer's [12]
			$\mathbf{2^{119.70}}$	Section VI.2
	13	Collision	$\mathbf{2^{119.70}}$	Section VI.3
$(3, 12, 1, 12)$	9	Preimage	$2^{135.94}$	Original [25]
			$\mathbf{2^{127.60}}$	Designer's [12]
			$\mathbf{2^{123.29}}$	Section V
			$\mathbf{2^{108.43}}$	Section VI.2
	10	Preimage	$2^{149.90}$	Original [25]
			$2^{141.49}$	Designer's [12]
			$\mathbf{2^{122.29}}$	Section VI.2
	10	Collision	$\mathbf{2^{122.29}}$	Section VI.3
$(3, 12, 2, 12)$	9	Preimage	$2^{135.94}$	Original [25]
			$\mathbf{2^{126.60}}$	Designer's [12]
			$\mathbf{2^{123.29}}$	Section V

Note: The complexity is estimated in field operations. R denotes the secure number of rounds; N denotes the number of attacked rounds; Bold numbers indicate the attacks are valid.

Table 3 A summary of the attacks on GrendelHash instances with parameters $\alpha = 5, \lambda = 128, \log_2(p) \approx 256$

Instance (α, m, c, R)	N	Type	Complexity	Ref.
$(5, 8, 1, 15)$	11	Preimage	$2^{133.24}$	Original [25]
			$\mathbf{2^{119.46}}$	Designer's [12]
			$\mathbf{2^{119.06}}$	Section V
			$\mathbf{2^{107.53}}$	Section VI.2
	12	Preimage	$2^{143.87}$	Original [25]
			$2^{130.03}$	Designer's [12]
			$\mathbf{2^{118.06}}$	Section VI.2
	12	Collision	$\mathbf{2^{118.06}}$	Section VI.3
$(5, 12, 1, 11)$	8	Preimage	$2^{129.18}$	Original [25]
			$\mathbf{2^{119.58}}$	Designer's [12]
			$\mathbf{2^{115.37}}$	Section V
			$\mathbf{2^{99.74}}$	Section VI.2
	9	Preimage	$2^{143.91}$	Original [25]
			$2^{134.24}$	Designer's [12]
			$\mathbf{2^{114.37}}$	Section VI.2
	9	Collision	$\mathbf{2^{114.37}}$	Section VI.3
$(5, 12, 2, 11)$	8	Preimage	$2^{129.18}$	Original [25]
			$\mathbf{2^{118.58}}$	Designer's [12]
			$\mathbf{2^{115.37}}$	Section V

Note: The complexity is estimated in field operations. R denotes the secure number of rounds; N denotes the number of attacked rounds; Bold numbers indicate the attacks are valid.

II. Preliminaries

In this section, we will first give some notations and related definitions. Then, we will give a brief description of Grendel [12]. Finally, we will describe the algorithm [21] for solving univariate system.

1. Notations and related definitions

In the following content of this paper, p denotes an odd prime. $\mathbb{F}_p$ denotes the finite field with p elements. For $\boldsymbol{x} = (x_0, \ldots, x_{a-1}) \in \mathbb{F}_p^a$ and $\boldsymbol{y} = (y_0, \ldots, y_{b-1}) \in \mathbb{F}_p^b$, the concatenation of $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted by $\boldsymbol{x} \parallel \boldsymbol{y}$, i.e., $\boldsymbol{x} \parallel \boldsymbol{y} = (x_0, \ldots, x_{a-1}, y_0, \ldots, y_{b-1}) \in \mathbb{F}_p^{a+b}$.

Definition 1 The hash function H is λ-bit secure against preimage attack if no algorithm has an expected complexity less than 2^λ for finding u given h such that $H(u) = h$.

Definition 2 The hash function H is λ-bit secure against collision attack if no algorithm has an expected complexity less than 2^λ for finding u_1, u_2 such that $H(u_1) = H(u_2)$ and $u_1 \neq u_2$.

Definition 3 Let $F : \mathbb{F}_p^n \to \mathbb{F}_p^n$ be a permutation. $\mathcal{Z}_t$ is the set of all elements of $\mathbb{F}_p^n$ such that their last t coordinates are equal to 0. The CICO problem is finding $\boldsymbol{x} \in \mathbb{F}_p^n$ such that $\boldsymbol{x} \in \mathcal{Z}_t$ and $F(\boldsymbol{x}) \in \mathcal{Z}_t$.

2. Description of Grendel

Grendel [12] defines a family of permutations with SHARK-like constructions [26]. Based on the Grendel permutation, the Grendel hash function is obtained by applying the sponge construction [22]–[24] and using the primitive as an inner permutation. The different permutations in the family are determined by the triple (p, m, c, λ) representing respectively the size of the field over which the primitive's operations are defined, the number of field elements in the state, the number of field elements in the sponge capacity and the target security level. The secure number of rounds R can be obtained from these parameters as follows:

$$R = \max\{R_{ld}, R_{sub}, R_{root}, R_{Grö}, R'_{Grö}\} \tag{1}$$

where

$$R_{ld} = \min\left\{R \geq 1 | \left(\frac{2\alpha}{p}\right)^{-\frac{R}{2}} \geq 2^{1.25\lambda}, \left(\frac{4\alpha - 2}{p}\right)^{-\frac{R}{2}} \geq 2^{1.25\lambda}\right\}$$

$$R_{sub} = \min\{R \geq 1 | p^{\lceil \frac{m+1}{2} \rceil} \geq 2^{1.25\lambda}, p^m \geq 2^{1.25\lambda}\}$$

$$R_{\text{root}} = \min\{R \geq 1 | p \geq 2^{1.25\lambda}, 2^{Rm-c} \cdot \alpha^R \cdot R^2 \geq 2^{1.25\lambda}\}$$

$$R_{Grö} = \min\left\{R \geq 1 | \begin{pmatrix} 2Rm - 2c + \frac{1+(Rm-c)(\alpha+3)}{8} \\ \frac{1+(Rm-c)(\alpha+3)}{8} \end{pmatrix}^2 \geq 2^{1.25\lambda}\right\}$$

$$R'_{Grö} = \min\left\{R \geq 1 | 2^{Rm-c} \cdot \begin{pmatrix} Rm - c + \frac{1+(Rm-c)(\alpha-1)}{9} \\ \frac{1+(Rm-c)(\alpha-1)}{9} \end{pmatrix}^2 \geq 2^{1.25\lambda}\right\}$$

In other words, R is set to the smallest integer such that the complexities of all attacks presented in Table 1 in [12] are at least $2^{1.25\lambda}$.

Grendel permutation uses the Legendre symbol as a component. Therefore we first introduce the definition of the Legendre symbol.

Definition 4 The Legendre symbol is a function $(\frac{\cdot}{p}) : \mathbb{F}_p \to \mathbb{F}_p$ defined as

$$\left(\frac{x}{p}\right) = \begin{cases} 0, & \text{if} \quad x = 0 \\ 1, & \text{if} \quad x \in \mathrm{QR}_p \\ -1, & \text{if} \quad x \notin \mathrm{QR}_p \cup \{0\} \end{cases}$$

where $\mathrm{QR}_p = \{b^2 | b \in \mathbb{F}_p \backslash \{0\}\}$.

Grendel permutation $P : \mathbb{F}_p^m \to \mathbb{F}_p^m$ $(m \geq 2)$ iterates a round function R times. At the i-th $(0 \leq i \leq R-1)$ round the round function consists of the following operations, as depicted in Figure 1.

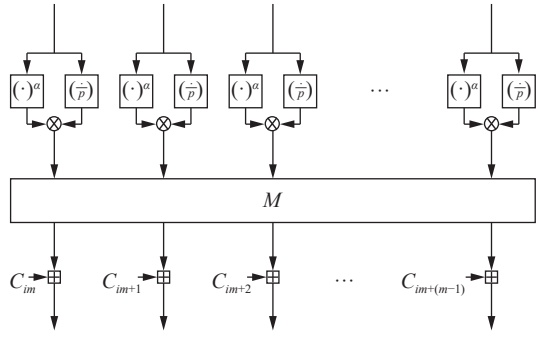


Figure 1 The Grendel round function.

1) SboxLayer (SB): A S-box $S : \mathbb{F}_p \to \mathbb{F}_p$ is applied to the m elements of the state in parallel. The S-box is defined as $S(x) = x^\alpha \times (\frac{x}{p})$, where if $p \equiv 3 \bmod 4$ then $\alpha = 2$ and if $p \equiv 1 \bmod 4$ then $\alpha \geq 3$ is the smallest integer such that $\gcd(\alpha, p-1) = 1$.

2) LinearLayer (L): The state vector is multiplied with an MDS matrix $\boldsymbol{M}$ of size $m \times m$.

3) ConstantAddition (AC): The round constant $C_{im+j} \in \mathbb{F}_p$ is added to the j-th element of the state, where $j \in \{0, 1, \ldots, m-1\}$.

By using the Grendel permutation in a sponge construction, the Grendel hash function is obtained. Specifically, r and c denote the number of field elements in the sponge rate and the sponge capacity respectively. And the number of field elements in the digest is denoted by l. Then for the input consisting of elements of $\mathbb{F}_p$, the procedure for computing its digest consists of the following operations, as depicted in Figure 2.

1) Padding: If the input length is variable, then the input is padded as follows: first append a single 1 to the input, and then append zeros until the total length is a multiple of r. If the input length is fixed, then the input does not need to be processed in this phase. Then divide the result into blocks $\boldsymbol{m}_0, \boldsymbol{m}_1, \ldots, \boldsymbol{m}_{k-1}$, where each block $\boldsymbol{m}_i$ $(0 \leq i \leq k-1)$ consists of r elements of $\mathbb{F}_p$.

2) Initializing: The state is initialized to $\boldsymbol{IV} = (0, 0, \ldots, 0) \in \mathbb{F}_p^m$.

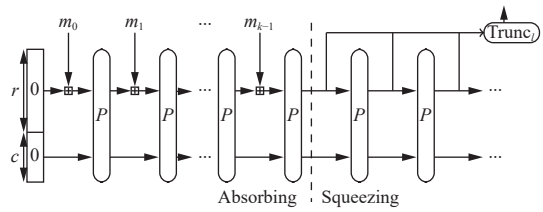


Figure 2 The absorbing and squeezing phases.

3) Absorbing: The blocks $\boldsymbol{m}_0, \boldsymbol{m}_1, \ldots, \boldsymbol{m}_{k-1}$ are added into the top r elements of the state, interleaved with applications of the permutation P. When all blocks are processed, switch to the squeezing phase.

4) Squeezing: First extract $\min(l, r)$ elements from the top r elements of the state. If $l > r$, then apply P to the state and extract $\min(l-r, r)$ elements from the top r elements of the state. Repeat this process until the number of extracted elements reach l.

In addition, for the target security level λ-bit, the number of field elements in the sponge capacity c and the number of field elements in the digest l should satisfy $\frac{c \log_2(p)}{2} \geq \lambda$ and $\frac{l \log_2(p)}{2} \geq \lambda$.

3. Solving univariate system

In this part, we will introduce the algorithm [21] for solving univariate system. Let $g(x) \in \mathbb{F}_p[x]$ and d denote the degree of $g(x)$. Our goal is to find the roots of $g(x)$ in $\mathbb{F}_p$. The algorithm consists of the following steps.

1) Use repeated squaring algorithm [27] in $\mathbb{F}_p[x]/\langle g(x) \rangle$ to compute $q(x) = x^p \bmod g(x)$.

2) Compute $r(x) = \gcd(q(x) - x, g(x))$.

3) Factor $r(x)$.

Complexity evaluation The algorithm's complexity given in [21] is $\mathcal{O}(d \log_2(d)(\log_2(d) + \log_2(p)) \log_2(\log_2(d)))$ field operations.

Specifically, multiplying two polynomials of degree n by using an FFT algorithm needs $\mathcal{O}(n \log_2(n) \log_2(\log_2(n)))$ field operations. $\mathcal{O}(d \log_2(p) \log_2(d) \log_2(\log_2(d)))$ field operations are needed in step 1); $\mathcal{O}(d(\log_2(d))^2 \log_2(\log_2(d)))$ field operations are needed in step 2). In general, the degree of $r(x)$ is only one or two because $g(x)$ has few roots in $\mathbb{F}_p$. Thus, the complexity of the third step is negligible.

We note that this algorithm was also used in the original preimage attack [25] and had a different complexity evaluation. However, the complexity given in [21] is sharper, which will be used in our work.

III. The Original Preimage Attack

In this section, we will briefly revisit the original preimage attack [25] on N-round GrendelHash instances, of which the number of field elements in the digest l is 1. And the number of field elements in the sponge rate $r \geq 2$.

Given a digest $h \in \mathbb{F}_p$, the overall procedure for finding a preimage can be separated into the following steps, as depicted in Figure 3.

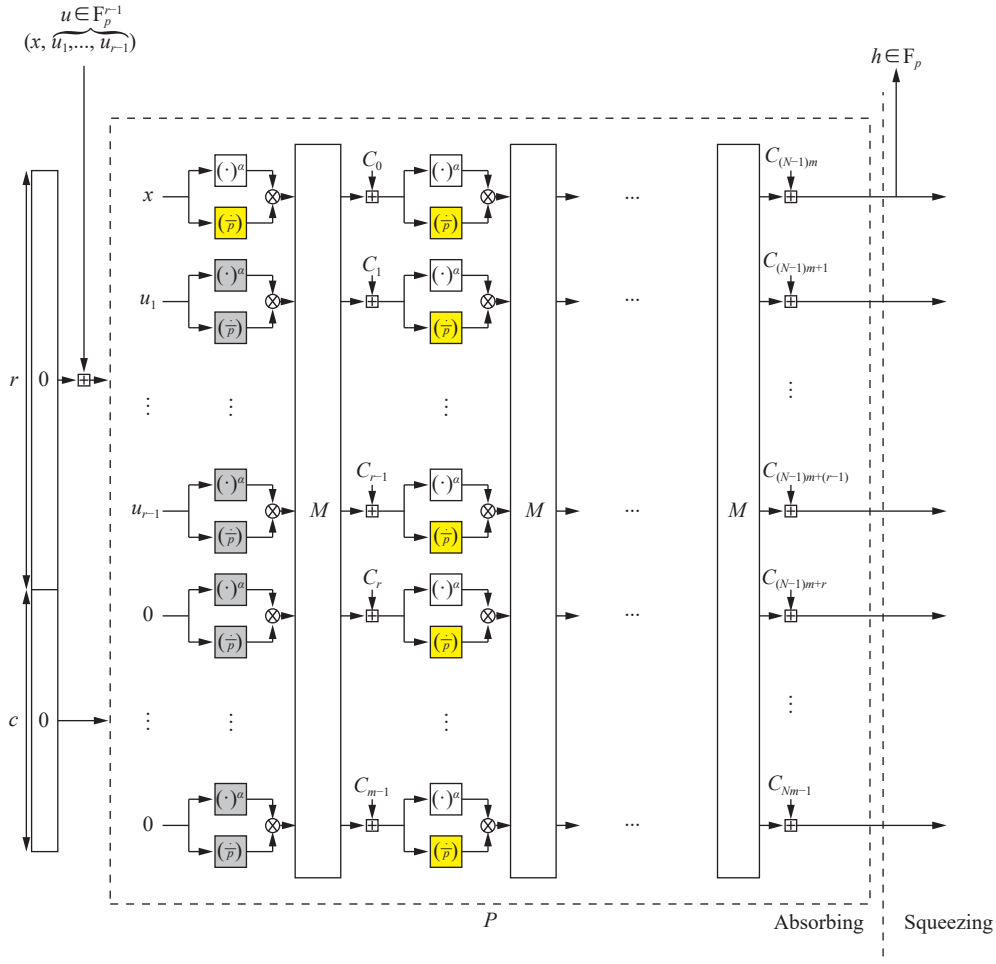


Figure 3 Introduce the steps of the original preimage attack, where the unknown Legendre symbols are colored in yellow and the known intermediate values are colored in gray.

1) Choose $\boldsymbol{u} = (u_1, \ldots, u_{r-1}) \in \mathbb{F}_p^{r-1}$ randomly. Let the input of GrendelHash be $x \parallel \boldsymbol{u} \in \mathbb{F}_p^r$, where x is an unknown variable.

2) Guess the values (only 1 or -1) of the $L = 1 + m(N-1)$ unknown Legendre symbols (i.e., the yellow states in the Figure 3) in the N rounds. Then write the first element in the output state of N-round Grendel permutation P as a polynomial in terms of x, which is denoted by $p(x)$ and has a degree of α^N.

3) Solve the equation $p(x) - h = 0$ in $\mathbb{F}_p$. For each solution x^*, if all computed Legendre symbols equal the guessed ones, then the solution is valid and return $x^* \parallel \boldsymbol{u}$. If all solutions we find are not valid, then go to step 2) and guess the Legendre symbols with values we have not tried before.

4) If all possible guesses are traversed and no valid solution is found, then go to step 1) and try again with a different $\boldsymbol{u}$.

Success probability For a random $\boldsymbol{u} \in \mathbb{F}_p^{r-1}$, it can be expected that there is a value $x_0 \in \mathbb{F}_p$ such that the digest of $x_0 \parallel \boldsymbol{u}$ equals h. The probability that a Legendre symbol is equal to 1 is $\frac{p-1}{2p}$, the probability that a Legendre symbol is equal to -1 is $\frac{p-1}{2p}$, and the probability that a Legendre symbol is equal to 0 is $\frac{1}{p}$. As for x_0, the probability that the L Legendre symbols are different from 0 is $(1 - \frac{1}{p})^L$, which is extremely close to 1 for the parameters considered in the attack. Therefore, the x_0 can be found successfully with a high enough probability. In order to increase the success probability of the attack, try twice with different $\boldsymbol{u}$.

As a countermeasure, the designer added this attack into the security analysis. The total complexity is estimated as $2^{Nm-c} \cdot \alpha^N \cdot N^2$ by the designer in [12].

IV. A Generalization of the Strategy of Bypassing SPN Steps

In this section, we will give a generalization of the strategy of bypassing SPN steps, which was originally proposed by Bariant *et al.* [21] to solve the CICO problem (in the case where $\mathcal{Z}_t = \mathcal{Z}_1$).

Let $F = F_1 \circ F_0$ be a permutation of $\mathbb{F}_p^n$ with a SPN construction. Let $\mathcal{B} \subseteq \mathbb{F}_p^n$ and $\mathcal{H}_j^b = \{(y_1, y_2, \ldots, y_n) \in \mathbb{F}_p^n | y_j = b\}$, where $1 \le j \le n$ and $b \in \mathbb{F}_p$. Our problem is finding $\boldsymbol{x} \in \mathbb{F}_p^n$ such that $\boldsymbol{x} \in \mathcal{B}$ and $F(\boldsymbol{x}) \in \mathcal{H}_j^b$.

If we find an affine space $\mathcal{W} = \{s\boldsymbol{\beta} + \boldsymbol{\gamma} | s \in \mathbb{F}_p\}$ in which $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_{n-1})$, $\boldsymbol{\gamma} = (\gamma_0, \ldots, \gamma_{n-1}) \in \mathbb{F}_p^n$ such

that $F_0^{-1}(\mathcal{W}) \subseteq \mathcal{B}$, then we will handle the permutation F_1 rather than the full permutation F. It could be easier to solve the problem. The detailed procedure can be separated into the following steps, as depicted in Figure 4.

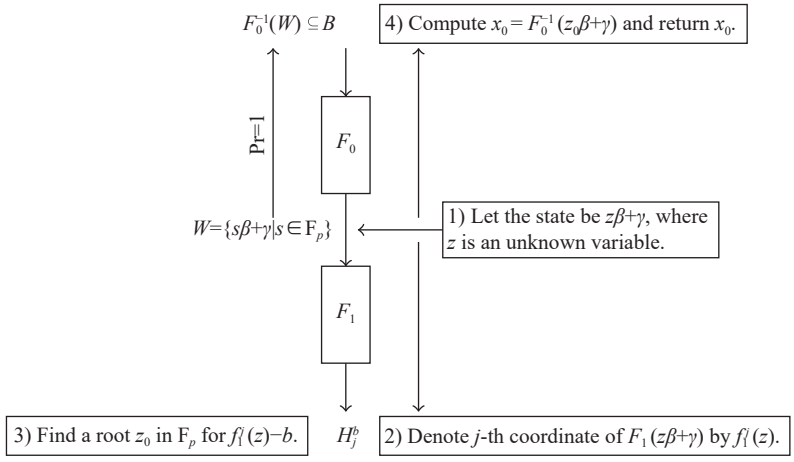


Figure 4 Bypass SPN steps.

1) Let the output state of F_0 (i.e., the input state of F_1) be $z\boldsymbol{\beta}+\boldsymbol{\gamma} = (\beta_0 z+\gamma_0, \ldots, \beta_{n-1}z+\gamma_{n-1})$, where z is an unknown variable.

2) Write the j-th element in the output state of F_1 as a polynomial in terms of z, which is denoted by $f_1^j(z)$.

3) Find a root z_0 in $\mathbb{F}_p$ for the polynomial $f_1^j(z)-b$.

4) Compute $\boldsymbol{x}_0 = F_0^{-1}(z_0\boldsymbol{\beta}+\boldsymbol{\gamma})$ and return $\boldsymbol{x}_0$.

V. Our First Preimage Attack

In this section, we will introduce our first preimage attack on N-round GrendelHash instances. The number of field elements in the output $l=1$, the number of field elements in the sponge rate $r \geq 2$, and $\log_2(p) \geq 2\lambda$. In the following, each SPN step consists of one round of Grendel.

1. Bypass one SPN step

Let $\mathcal{B}_1$ be the set of all elements of $\mathbb{F}_p^m$ such that their last c coordinates are equal to 0, i.e., $\mathcal{B}_1 = \{(x_0, x_1, \ldots, x_{m-1}) \in \mathbb{F}_p^m | x_i = 0, m-c \leq i \leq m-1\}$. For a given digest $h \in \mathbb{F}_p$, if we find $\boldsymbol{x} \in \mathbb{F}_p^m$ such that $\boldsymbol{x} \in \mathcal{B}_1$ and $P(\boldsymbol{x}) \in \mathcal{H}_1^h$, where P is the underlying N-round Grendel permutation, then we will find a preimage. Therefore, we can exploit the strategy of bypassing SPN steps presented in Section IV in this attack.

Let N-round Grendel permutation be $P = P_{N-1} \circ P_1$, where P_1 consists of one SPN step of Grendel, i.e., the operations in the 0th round. Let the MDS matrix $\boldsymbol{M}$ be

$$\boldsymbol{M} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,m-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m-1,0} & a_{m-1,1} & \cdots & a_{m-1,m-1} \end{bmatrix}$$

Choose $\boldsymbol{w} = (w_1, \ldots, w_{r-1}) \in \mathbb{F}_p^{r-1}$. Let $\boldsymbol{\beta} = (a_{0,0}, a_{1,0}, \ldots, a_{m-1,0})$ and $\boldsymbol{\gamma} = (\sum_{i=1}^{r-1} a_{0,i}w_i + C_0, \sum_{i=1}^{r-1} a_{1,i}w_i + C_1, \ldots, \sum_{i=1}^{r-1} a_{m-1,i}w_i + C_{m-1})$. Then the affine space $\mathcal{W} = \{s\boldsymbol{\beta}+\boldsymbol{\gamma} | s \in \mathbb{F}_p\}$ satisfies $P_1^{-1}(\mathcal{W}) \subseteq \mathcal{B}_1$, as depicted in Figure 5.

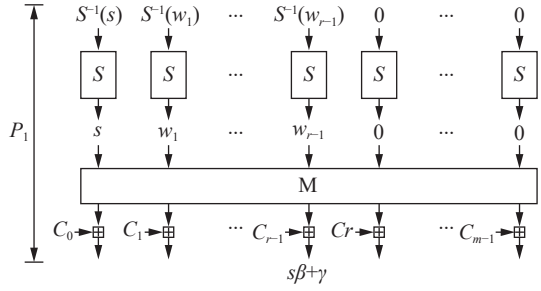


Figure 5 Bypass one SPN step.

Therefore, the remaining task is to find z_0 in $\mathbb{F}_p$ such that $P_{N-1}(z_0\boldsymbol{\beta}+\boldsymbol{\gamma}) \subseteq \mathcal{H}_1^h$.

2. The steps of our first preimage attack

In the following, we will introduce our first preimage attack in detail. Let the input state of P_{N-1} be $z\boldsymbol{\beta}+\boldsymbol{\gamma}$, where z is an unknown variable. Denote the first coordinate in $P_{N-1}(z\boldsymbol{\beta}+\boldsymbol{\gamma})$ by $f_{N-1}^1(z)$. In order to find z_0 in $\mathbb{F}_p$ such that $P_{N-1}(z_0\boldsymbol{\beta}+\boldsymbol{\gamma}) \subseteq \mathcal{H}_1^h$, instead of solving the equation $f_{N-1}^1(z) = h$ in $\mathbb{F}_p$ directly, we exploit the strategy that was used in the original preimage attack [25].

Specifically, the procedure for finding a preimage in this attack can be separated into the following steps (as shown in Algorithm 1).

1) Choose $\boldsymbol{w} = (w_1, \ldots, w_{r-1}) \in \mathbb{F}_p^{r-1}$ randomly. Let $\boldsymbol{\beta} = (a_{0,0}, a_{1,0}, \ldots, a_{m-1,0})$ and $\boldsymbol{\gamma} = (\sum_{i=1}^{r-1} a_{0,i}w_i + C_0, \sum_{i=1}^{r-1} a_{1,i}w_i + C_1, \ldots, \sum_{i=1}^{r-1} a_{m-1,i}w_i + C_{m-1})$.

2) Let the input of P_{N-1} be $z\boldsymbol{\beta}+\boldsymbol{\gamma}$, where z is an unknown variable.

3) Guess the values (only $1/-1$) of the $L_1 = m(N-1)$

unknown Legendre symbols in P_{N-1}. And write the first element in the output state of P_{N-1} as a polynomial in terms of z, which is denoted by $p_1(z)$.

4) Solve the equation $p_1(z) - h = 0$ in $\mathbb{F}_p$. For each solution z^*, if all computed Legendre symbols equal the guessed ones in the $N-1$ rounds, then the solution is valid and return the first r elements in $P_1^{-1}(z^*\boldsymbol{\beta} + \boldsymbol{\gamma})$. If all solutions we find are invalid, then go to step 3) and guess the Legendre symbols with values we have not tried before.

5) If all possible guesses are traversed and no valid solution is found, then go to step 1) and try again with a different $\boldsymbol{w}$.

Algorithm 1 Our first preimage attack on N-Round GrendelHash

Input: The given digest $h \in \mathbb{F}_p$.
Output: A preimage.

1: Choose $\boldsymbol{w} = (w_1, w_2, \ldots, w_{r-1}) \in \mathbb{F}_p^{r-1}$ randomly;
2: Let $\boldsymbol{\beta} = (a_{0,0}, a_{1,0}, \ldots, a_{m-1,0})$ and $\boldsymbol{\gamma} = (\sum_{i=1}^{r-1} a_{0,i}w_i + C_0, \sum_{i=1}^{r-1} a_{1,i}w_i + C_1, \ldots, \sum_{i=1}^{r-1} a_{m-1,i}w_i + C_{m-1})$;
3: Let the input of P_{N-1} be $z\boldsymbol{\beta} + \boldsymbol{\gamma}$, where z is an unknown variable;
4: **for** guessed values (only 1 or -1) of L_1 Legendre symbols **do**
5: Build the polynomial $p_1(z)$ forwards to represent the first element in the output of P_{N-1};
6: Solve the equation $p_1(z) - h = 0$ in $\mathbb{F}_p$;
7: **for** solution z^* we find **do**
8: **if** all computed Legendre symbols equal the guessed ones **then**
9: Compute $\boldsymbol{x} = P_1^{-1}(z^*\boldsymbol{\beta} + \boldsymbol{\gamma})$;
10: **return** $\boldsymbol{x}[0:r]$;
11: **return** No valid solution, try again with a different $\boldsymbol{w}$.

Success probability For a random $\boldsymbol{w} \in \mathbb{F}_p^{r-1}$, it can be expected that there is a value z_0 in $\mathbb{F}_p$ such that $P_{N-1}(z_0\boldsymbol{\beta} + \boldsymbol{\gamma}) \subseteq \mathcal{H}_1^h$. As for z_0, the probability that the L_1 Legendre symbols in P_{N-1} are different from 0 is equal to $(1 - \frac{1}{p})^{L_1}$, which is extremely close to 1 for the parameters considered in our attack. Thus, z_0 can be found successfully with a high enough probability. Similarly to that in the original preimage attack, we try twice with different $\boldsymbol{w}$ to increase the success probability of the attack.

Complexity evaluation For each guess, we need to find the roots of the polynomial $p_1(z) - h$ in $\mathbb{F}_p$ and it can be expected that the number of roots in $\mathbb{F}_p$ is 1. Since the degree D_1 of $p_1(x) - h$ is α^{N-1}, then as shown in Section II.3, the complexity of solving the equation $p_1(z) - h = 0$ is equal to

$$C_{sol}^1 = D_1 \log_2(D_1)(\log_2(D_1) + \log_2(p)) \log_2(\log_2(D_1))$$

For the solution, when verifying whether it is valid, we need to compute the Legendre symbols forwards until there is a computed value that do not match the guessed one. The expected number of Legendre symbols that need to be computed is equal to

$$\sum_{i \geq 1} \frac{i}{2^i} = 2$$

In addition, the complexity of computing a Legendre symbol [28] is equal to

$$C_{ver} = \log_2(p)(\log_2(\log_2(p)))^2 \log_2(\log_2(\log_2(p)))$$

Hence, the expected complexity of our first preimage attack on N-round GrendelHash is

$$C_1 = 2^{L_1+1} \times (C_{sol}^1 + 2 \times C_{ver})$$

The results of our first preimage attack on some instances are summarized in Tables 1–3.

Remark In our first preimage attack, the reasons for the reduced complexity compared to the original preimage attack [25] are as follows. First, the number of Legendre symbols needed to be guessed is reduced by 1. Second, the degree of univariate polynomial needed to be solved after guessing the values of the unknown Legendre symbols is α^{N-1}, which is α times lower than that in the original preimage attack. Third, for the algorithm of solving univariate system, we use the complexity evaluation given in [21] at FSE 2023, which is better than that used in [25].

VI. For the Special Case Where $c = 1$

In this section, we will introduce our second preimage attack and the first collision attack on some N-round GrendelHash instances. The number of field elements in the output $l = 1$, the number of field elements in the sponge rate $r \geq 2$, the number of field element in the sponge capacity $c = 1$, and $\log_2(p) \geq 2\lambda$.

1. Bypass two SPN steps

Let $\mathcal{B}_2$ be the set of all elements of $\mathbb{F}_p^m$ such that their last coordinate is equal to 0, i.e., $\mathcal{B}_2 = \{(x_0, x_1, \ldots, x_{m-1}) \in \mathbb{F}_p^m | x_{m-1} = 0\}$. Let the underlying permutation be $P = P_{N-2} \circ P_2$, where P_2 consists of two SPN steps of Grendel, i.e., the operations in the 0th round and the 1st round. Let the inverse of $\boldsymbol{M}$ be

$$\boldsymbol{M}^{-1} = \begin{bmatrix} b_{0,0} & b_{0,1} & \cdots & b_{0,m-1} \\ b_{1,0} & b_{1,1} & \cdots & b_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m-1,0} & b_{m-1,1} & \cdots & b_{m-1,m-1} \end{bmatrix}$$

Let $\boldsymbol{\beta'} = (S(A_0), S(A_1), \ldots, S(A_{m-2}), 0)$ and $\boldsymbol{\gamma'} = (0, 0, \ldots, 0, g)$, where A_i $(0 \leq i \leq m-2)$ and g are constants in $\mathbb{F}_p$ that satisfy the following conditions:

$$\sum_{i=0}^{m-2} A_i \cdot b_{m-1,i} = 0$$

$$g = S\left(\frac{\sum_{j=0}^{m-1} b_{m-1,j} \cdot C_j}{b_{m-1,m-1}}\right) = \left(\frac{\sum_{j=0}^{m-1} b_{m-1,j} \cdot C_j}{b_{m-1,m-1}}\right)^{\alpha+\frac{p-1}{2}}$$

And let $\tilde{\boldsymbol{\beta}}^{\mathrm{T}} = \boldsymbol{M} \cdot \boldsymbol{\beta'}^{T}$ and $\tilde{\boldsymbol{\gamma}}^{\mathrm{T}} = \boldsymbol{M} \cdot \boldsymbol{\gamma'}^{\mathrm{T}} + (C_m, C_{m+1}, \ldots, C_{2m-1})^{\mathrm{T}}$. Then the affine space $\mathcal{W}_1 = \{s\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\gamma}} | s \in \mathbb{F}_p\}$ satisfies that $P_2^{-1}(\mathcal{W}_1) \subseteq \mathcal{B}_2$, as depicted in Figure 6.

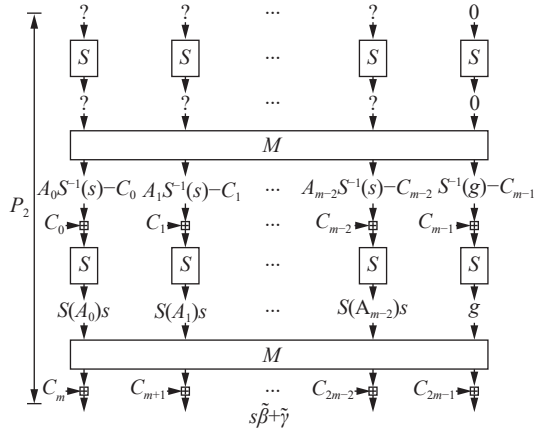


Figure 6 Bypass two SPN steps.

2. Our second preimage attack

In the following, we will describe the steps of our second preimage attack. For a given digest $h \in \mathbb{F}_p$, in order to find z_1 in $\mathbb{F}_p$ such that $P_{N-2}(z_1\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\gamma}}) \subseteq \mathcal{H}_1^h$, we also exploit the strategy that was used in the original preimage attack [25].

Specifically, the procedure for finding a preimage in this attack can be separated into the following steps (as shown in Algorithm 2).

1) Let the input state of P_{N-2} be $z\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\gamma}}$, where z is an unknown variable.

2) Guess the values (only $1/-1$) of the $L_2 = m(N-2)$ unknown Legendre symbols in the P_{N-2}. And write the first element in the output of P_{N-2} as a polynomial in terms of z, which is denoted by $p_2(z)$.

3) Solve the equation $p_2(z) - h = 0$ in $\mathbb{F}_p$. For each solution z^*, if the computed Legendre symbols equal the guessed ones, then the solution is valid and go to step 4). If all solutions are invalid, then go to step 2) and guess the Legendre symbols with values we have not tried before.

4) Compute $\boldsymbol{x} = P_2^{-1}(z^*\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\gamma}})$ and return the first r elements of $\boldsymbol{x}$.

Algorithm 2 Our second preimage attack on N-round GrendelHash

Input: The given digest $h \in \mathbb{F}_p$.

Output: A preimage.

1: Let the input of P_{N-2} be $z\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\gamma}}$, where z is an unknown variable;
2: **for** guessed values (only 1 or -1) of L_2 Legendre symbols **do**
3: Build the polynomial $p_2(z)$ forwards to represent the first element in the output state of P_{N-2};
4: Solve the equation $p_2(z) - h = 0$ in $\mathbb{F}_p$;
5: **for** solution z^* we find **do**
6: **if** all computed Legendre symbols equal the guessed ones **then**
7: Compute $\boldsymbol{x} = P_2^{-1}(z^*\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\gamma}})$;
8: **return** $\boldsymbol{x}[0:r]$;
9: **return** No valid solution.

Success probability It can be expected that there is a value $z_1 \in \mathbb{F}_p$ such that the first element in $P_{N-2}(z_1\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\gamma}})$ equals the given digest h. As for z_1, the probability that the L_2 Legendre symbols are different from 0 is equal to $(1 - \frac{1}{p})^{L_2}$, which is extremely close to 1 for the parameters considered in our attack. Therefore, we can find a preimage for a given digest h with a high enough probability.

Complexity evaluation In this attack, for each guess, the degree D_2 of $p_2(z) - h$ is α^{N-2}. Then the complexity of solving the equation $p_2(z) - h = 0$ is equal to

$$C_{sol}^2 = D_2 \log_2(D_2)(\log_2(D_2) + \log_2(p)) \log_2(\log_2(D_2))$$

Therefore, the expected complexity of our second preimage attack on N-round GrendelHash is

$$C_2 = 2^{L_2} \times (C_{sol}^2 + 2 \times C_{ver}) \tag{2}$$

The results of our second preimage attack on some instances are summarized in Tables 1–3.

3. The first collision attack

In the following, we will present our collision attack by solving the CICO problem (in the case where $\mathcal{Z}_t = \mathcal{Z}_1$) for the underlying permutation P. The procedure for finding a solution to the CICO problem is almost the same as the steps of our second preimage attack. After finding a solution $\boldsymbol{x} = (x_0, x_1, \ldots, x_{r-1}, 0)$ to the CICO problem, where $P(\boldsymbol{x}) = (y_0, y_1 \ldots, y_{r-1}, 0)$, let $\boldsymbol{m} = (x_0, x_1, \ldots, x_{r-1})$ and $\boldsymbol{m'} = (x_0 - y_0, x_1 - y_1, \ldots, x_{r-1} - y_{r-1})$. Then GrendelHash$(\boldsymbol{m})$ = GrendelHash$(\boldsymbol{m} \| \boldsymbol{m'})$.

Specifically, the procedure for the collision attack can be separated into the following steps (as shown in Algorithm 3).

1) Let the input state of P_{N-2} be $z\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\gamma}}$, where z is an unknown variable.

2) Guess the values (only $1/-1$) of the $L_2 = m(N-2)$ unknown Legendre symbols in the P_{N-2}. And write the last element in the output of P_{N-2} as a polynomial in terms of z, which is denoted by $p_3(z)$.

3) Solve the equation $p_3(z)=0$ in $\mathbb{F}_p$. For each solution z^*, if the computed Legendre symbols equal the guessed ones, then the solution is valid and go to step 4). If all solutions are invalid, then go to step 2) and guess the Legendre symbols with values we have not tried before.

4) Compute $\boldsymbol{x}=P_2^{-1}(z^*\tilde{\boldsymbol{\beta}}+\tilde{\boldsymbol{\gamma}})$ and $\boldsymbol{y}=P_{N-2}(z^*\tilde{\boldsymbol{\beta}}+\tilde{\boldsymbol{\gamma}})$. Denote $\boldsymbol{x}=(x_0,x_1,\ldots,x_{r-1},\ 0)$ and $\boldsymbol{y}=(y_0,y_1,\ldots,y_{r-1},0)$. Let $\boldsymbol{m}=(x_0,x_1,\ldots,x_{r-1})$ and $\boldsymbol{m'}=(x_0-y_0,x_1-y_1,\ldots,x_{r-1}-y_{r-1})$. Return $\boldsymbol{m}$ and $\boldsymbol{m}\|\boldsymbol{m'}$.

Algorithm 3 Our collision attack on N-round GrendelHash

Output: A collision.

1: Let the input state of P_{N-2} be $z\tilde{\boldsymbol{\beta}}+\tilde{\boldsymbol{\gamma}}$, where z is an unknown variable;
2: **for** guessed values (only 1 or -1) of L_2 Legendre symbols **do**
3: Build the polynomial $p_3(z)$ forwards to represent the last element in the output state of P_{N-2};
4: Solve the equation $p_3(z)=0$ in $\mathbb{F}_p$;
5: **for** solution z^* we find **do**
6: **if** all computed Legendre symbols equal the guessed ones **then**
7: Compute $\boldsymbol{x}=P_2^{-1}(z^*\tilde{\boldsymbol{\beta}}+\tilde{\boldsymbol{\gamma}})$ and $\boldsymbol{y}=P_{N-2}(z^*\tilde{\boldsymbol{\beta}}+\tilde{\boldsymbol{\gamma}})$;
8: Let $\boldsymbol{m}=\boldsymbol{x}[0:r]$ and $\boldsymbol{m'}=\boldsymbol{x}[0:r]-\boldsymbol{y}[0:r]$;
9: **return** $\boldsymbol{m}$, $\boldsymbol{m}\|\boldsymbol{m'}$.

We note that the success probability and expected complexity of our collision attack are the same as those of our second preimage attack that presented in Section VI.2. The results of our collision attack on some instances are summarized in Tables 1–3.

VII. Experiments

In order to verify the feasibility of our methods, we made experiments for our three new algebraic attacks on the round-reduced GrendelHash instance with parameters $(p,m,r,c,\lambda)=(2^{64}-59,3,2,1,32)$. And the number of attacked round $N=6$. We provide our code at https://github.com/wxqiao/New-algebraic-attacks-on-Grendel. Let the prime field $\mathbb{F}_p=\{0,1,\ldots,p-1\}$.

First, we performed 100 tests for our first preimage attack. By using the Algorithm 1, 83 tests of them can be attacked successfully. For instance, for a given digest $h_1=4509062438608773474\in\mathbb{F}_p$, we found a preimage

$$(1856239268434541265,1100434508024597864 5)\in\mathbb{F}_p^2$$

Second, we performed 1000 tests for our second preimage attack. By using the Algorithm 2, 608 tests of them can be attacked successfully. For instance, for a given digest $h_2=14681296474556465288\in\mathbb{F}_p$, we found a preimage

$$(15277210688656701824,1087535840334498627)\in\mathbb{F}_p^2$$

Third, for our collision attack, by using the method presented in Section VI.3, we found a solution $\boldsymbol{x}=(3978053920827369818,11054388833269671370,0)\in\mathbb{F}_p^3$ to the CICO problem for the underlying Grendel permutation P successfully, where $P(\boldsymbol{x})=(12866554063376284852,7184824262592905636,0)\in\mathbb{F}_p^3$. Let $\boldsymbol{m}=(3978053920827369818,11054388833269671370)\in\mathbb{F}_p^2$ and

$$\begin{aligned}\boldsymbol{m'}=&(3978053920827369818-12866554063376284852,\\&11054388833269671370-7184824262592905636)\\=&(9558243931160636523,3869564570676765734)\in\mathbb{F}_p^2\end{aligned}$$

Then, a collision $\text{GrendelHash}(\boldsymbol{m})=\text{GrendelHash}(\boldsymbol{m}\|\boldsymbol{m'})$ is obtained for this round-reduced GrendelHash instance successfully.

VIII. Conclusion

In this paper, we analyze the security of AO cipher Grendel. We propose three algebraic attacks on some GrendelHash instances. As a result, we can find a preimage with a lower complexity or attack one more round than previous attacks [12], [25] for some GrendelHash instances. In addition, we present the first collision attack on some round-reduced GrendelHash instances by solving the CICO problem for the underlying permutations.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (Grant No. 2022YFB2701900), the National Natural Science Foundation of China (Grant No. 62202444), and the Fundamental Research Funds for the Central Universities.

References

[1] M. R. Albrecht, C. Rechberger, T. Schneider, *et al.*, "Ciphers for MPC and FHE," in *Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Sofia, Bulgaria, pp.430–454, 2015.

[2] P. Méaux, A. Journault, F. X. Standaert, *et al.*, "Towards stream ciphers for efficient FHE with low-noise ciphertexts," in *Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Vienna, Austria, pp.311–343, 2016.

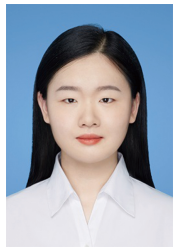
[3] A. Canteaut, S. Carpov, C. Fontaine, *et al.*, "Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression," *Journal of Cryptology*, vol. 31, no. 3, pp. 885–916, 2018.

[4] C. Dobraunig, M. Eichlseder, L. Grassi, *et al.*, "Rasta: A cipher with low ANDdepth and few ANDs per bit," in *Proceedings of the 38th Annual International Cryptology Conference*, Santa Barbara, CA, USA, pp.662–692, 2018.

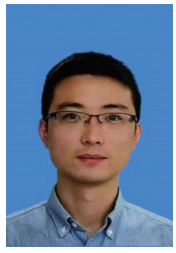
[5] M. Albrecht, L. Grassi, C. Rechberger, *et al.*, "MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity," in *Proceedings of the 22nd International Conference on the Theory and Application of Cryptology and Information Security*, Hanoi, Vietnam, pp.191–219, 2016.

[6] M. R. Albrecht, L. Grassi, L. Perrin, *et al.*, "Feistel structures for MPC, and more, " in *Proceedings of the 24th European Symposium on Research in Computer Security*, Luxem-

bourg, pp.151–171, 2019.
[7] C. Dobraunig, L. Grassi, A. Guinet, *et al.*, "CIMINION: Symmetric encryption based on toffoli-gates over large finite fields," in *Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Zagreb, Croatia, pp. 3–34, 2021.
[8] L. Grassi, D. Khovratovich, C. Rechberger, *et al.*, "Poseidon: A new hash function for zero-knowledge proof systems," in *Proceedings of the 30th USENIX Security Symposium*, online, pp.519–535, 2021.
[9] L. Grassi, S. Onofri, M. Pedicini, *et al.*, "Invertible quadratic non-linear layers for MPC-/FHE-/ZK-friendly schemes over $\mathbb{F}_p^n$: Application to poseidon," *IACR Transactions on Symmetric Cryptology*, vol. 2022, no. 3, pp. 20–72, 2022.
[10] A. Aly, T. Ashur, E. Ben-Sasson, *et al.*, "Design of symmetric-key primitives for advanced cryptographic protocols," *IACR Transactions on Symmetric Cryptology*, vol. 2020, no. 3, pp. 1–45, 2020.
[11] A. Szepieniec, T. Ashur, and S. Dhooghe, "Rescue-prime: A standard specification (SoK)," Available at: https://eprint.iacr.org/2020/1143, 2020.)
[12] A. Szepieniec, "On the use of the Legendre symbol in symmetric cipher design, " Available at: https://eprint.iacr.org/2021/984, 2021.
[13] C. Dobraunig, M. Eichlseder, and F. Mendel, "Higher-order cryptanalysis of LowMC," in *Proceedings of the 18th International Conference*, Seoul, South Korea, pp. 87–101, 2016.
[14] I. Dinur, Y. W. Liu, W. Meier, *et al.*, "Optimized interpolation attacks on LowMC," in *Proceedings of the 21st International Conference on the Theory and Application of Cryptology and Information Security*, Auckland, New Zealand, pp. 535–560, 2015.
[15] C. Rechberger, H. Soleimany, and T. Tiessen, "Cryptanalysis of low-data instances of full LowMCv2," *IACR Transactions on Symmetric Cryptology*, vol. 2018, no. 3, pp. 163–181, 2018.
[16] F. K. Liu, T. Isobe, and W. Meier, "Cryptanalysis of full LowMC and LowMC-M with algebraic techniques," in *Proceedings of the 41st Annual International Cryptology Conference*, Virtual Event, pp.368–401, 2021.
[17] I. Dinur, "Cryptanalytic applications of the polynomial method for solving multivariate equation systems over GF(2)," in *Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Zagreb, Croatia, pp. 374–403, 2021.
[18] F. K. Liu, S. Sarkar, G. L. Wang, *et al.*, "Algebraic meet-in-the-middle attack on LowMC," in *Proceedings of the 28th International Conference on the Theory and Application of Cryptology and Information Security*, Taipei, China, pp. 225–255, 2022.
[19] F. K. Liu, W. Meier, S. Sarkar, *et al.*, "New low-memory algebraic attacks on LowMC in the picnic setting," *IACR Transactions on Symmetric Cryptology*, vol. 2022, no. 3, pp. 102–122, 2022.
[20] L. Grassi, D. Khovratovich, R. Lüftenegger, *et al.*, "Reinforced concrete: A fast hash function for verifiable computation," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, Los Angeles, CA, USA, pp. 1323–1335, 2022.
[21] A. Bariant, C. Bouvier, G. Leurent, *et al.*, "Algebraic attacks against some arithmetization-oriented primitives," *IACR Transactions on Symmetric Cryptology*, vol. 2022, no. 3, pp. 73–101, 2022.
[22] G. Bertoni, J. Daemen, M. Peeters, *et al.*, "Sponge functions," Available at: https://keccak.team/files/SpongeFunctions.pdf, 2007.
[23] G. Bertoni, J. Daemen, M. Peeters, *et al.*, "On the indifferentiability of the sponge construction," in *Proceedings of the 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Istanbul, Turkey, pp.181–197, 2008.
[24] G. Bertoni, J. Daemen, M. Peeters, *et al.*, "Cryptographic sponge functions," Available at: https://keccak.team/files/CSF-0.1.pdf, 2011-01-14.
[25] L. Grassi, D. Khovratovich, S. Rønjom, *et al.*, "The Legendre symbol and the modulo-2 operator in symmetric schemes over $\mathbb{F}_p^n$: Preimage Attack on Full Grendel," *IACR Transactions on Symmetric Cryptology*, vol. 2022, no. 1, pp. 5–37, 2022.
[26] V. Rijmen, J. Daemen, B. Preneel, *et al.*, "The cipher SHARK," in *Proceedings of the 3rd International Workshop on Fast Software Encryption*, Cambridge, UK, pp.99–111, 1996.
[27] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, 3rd ed., Cambridge University Press, New York, NY, USA, pp.75-76, 2013.
[28] R. P. Brent and P. Zimmermann, "An $O(M(n)\log n)$ algorithm for the Jacobi symbol," in *Proceedings of the 9th International Algorithmic Number Theory Symposium*, Nancy, France, pp.83–95, 2010.



Wenxiao QIAO is a Ph.D. candidate of Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include Symmetric cryptanalysis and Design. (Email: qiaowenxiao@iie.ac.cn)



Siwei SUN received the Ph.D. degree from Chinese Academy of Sciences in 2013. He is a Professor in School of Cryptology, University of Chinese Academy of Sciences, Beijing, China. His research interests include Symmetric cryptanalysis and Design.
(Email: siweisun.isaac@gmail.com)

Lei HU received the Ph.D. degree from Chinese Academy of Sciences in 1994. He is a Professor in Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His research interests include basic theory of applications of pseudorandom sequences and arrays, analysis of cryptographic algorithms and theoretical cryptography.
(Email: hulei@iie.ac.cn)