IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 0, No. 0, pp. 0–0. DOI:10.46586/tosc.v0.i0.0-0

Algebraic Attack on FHE-Friendly Cipher HERA Using Multiple Collisions

Fukang Liu[1], Abul Kalam[2], Santanu Sarkar[2], Willi Meier[3]

[1] Tokyo Institute of Technology, Tokyo, Japan
liu.f.ad@m.titech.ac.jp
[2] Indian Institute of Technology Madras, Chennai, India
abulkalam.sunny@gmail.com, santanu@iitm.ac.in
[3] University of Applied Sciences and Arts Northwestern Switzerland (FHNW), Windisch, Switzerland
willimeier48@gmail.com

Abstract. Fully homomorphic encryption (FHE) is an advanced cryptography technique to allow computations (i.e., addition and multiplication) over encrypted data. After years of effort, the performance of FHE has been significantly improved and it has moved from theory to practice. The transciphering framework is another important technique in FHE to address the issue of ciphertext expansion and reduce the client-side computational overhead. To apply the transciphering framework to the CKKS FHE scheme, a new transciphering framework called the Real-to-Finite-Field (RtF) framework and a corresponding FHE-friendly symmetric-key primitive called HERA were proposed at ASIACRYPT 2021. Although HERA has a very similar structure to AES, it is considerably different in the following aspects: 1) the power map $x \mapsto x^3$ is used as the S-box; 2) a randomized key schedule is used; 3) it is over a prime field $\mathbb{F}_p$ with $p > 2^{16}$. In this work, we perform the first third-party cryptanalysis of HERA, by showing how to mount new algebraic attacks with multiple collisions in the round keys. Specifically, according to the special way to randomize the round keys in HERA, we find it possible to peel off the last nonlinear layer by using collisions in the last-round key and a simple property of the power map. In this way, we could construct an overdefined system of equations of a much lower degree in the key, and efficiently solve the system via the linearization technique. As a result, for HERA with 192 and 256 bits of security, respectively, we could break some parameters under the same assumption made by designers that the algebra constant ω for Gaussian elimination is $\omega = 2$, i.e., Gaussian elimination on an $n \times n$ matrix takes $\mathcal{O}(n^{\omega})$ field operations. If using more conservative choices like $\omega \in \{2.8, 3\}$, our attacks can also successfully reduce the security margins of some variants of HERA to only 1 round. However, the security of HERA with 80 and 128 bits of security is not affected by our attacks due to the high cost to find multiple collisions. In any case, our attacks reveal a weakness of HERA caused by the randomized key schedule and its small state size.

Keywords: HERA · algebraic attack · multiple collisions · randomized key schedule

1 Introduction

Most traditional symmetric-key primitives are designed over $\mathbb{F}_2$, and the corresponding cryptographic components including nonlinear layers, affine layers, and key schedules are all fixed. However, such a design strategy has evolved due to the new requirements on symmetric-key primitives in applications to secure multi-party computation (MPC) [KHS+22, AGR+16, DKR+22, ARS+15, DGGK21, GØSW23], fully homomorphic encryption (FHE) [NLV11, MJSC16, CCF$^+$18, DEG$^+$18, HKC$^+$20, HL20, CHK$^+$21, CHMS22, AMT22, CIR22, HKL$^+$22, DGH$^+$23], and zero-knowledge proof systems (ZKP) [AGP$^+$19, AAB$^+$20, GKR$^+$21, GKL$^+$22, BBC$^+$23, GHR$^+$23].

In the case of FHE-friendly symmetric-key primitives, it is common to randomize some components, like affine layers [MJSC16, DEG$^+$18, HKC$^+$20, HL20, CHMS22, CIR22, DGH$^+$23] or key schedules [CHK$^+$21, HKL$^+$22]. By such a way to randomize the cipher, each input of the cipher is then evaluated with an ever-changing function under the same key. In this way, conventional cryptanalysis techniques like the differential attack [BS90], linear attack [Mat93], integral attack [KW02], and cube attack [DS09] cannot work, since they all require that a number of different inputs are evaluated with the same function under the same key. Hence, although many such FHE-friendly ciphers have a very small number of rounds and use low-degree nonlinear layers, they are still resistant to these conventional attacks. Due to the low number of rounds and low-degree nonlinear layers in such ciphers, the most effective attack becomes the algebraic attack, and especially the linearization attack, which is commonly used by designers to determine the secure number of rounds [DEG$^+$18, HKC$^+$20, HL20, CHK$^+$21, CIR22, HKL$^+$22, DGH$^+$23].

Linearization attack. The linearization technique is the simplest method to solve an overdefined system of multivariate equations over a field $\mathbb{F}$. Let us consider m equations in terms of n variables, where the degree of each equation is upper bounded by d. In such a system of equations, there are at most $\sum_{i=1}^{d}\binom{n+i-1}{i} = \binom{n+d}{d} - 1$ non-constant terms of degree smaller than d formed by the n variables[1]. For simplicity, we treat $\binom{n+d}{d} - 1$ as $\binom{n+d}{d}$ throughout this paper. In the linearization technique, each such different term is treated as an independent variable. In this way, the system of m nonlinear equations can be converted into m linear equations in $\binom{n+d}{d}$ variables. If $m \geq \binom{n+d}{d}$ and the coefficient matrix of size $m \times \binom{n+d}{d}$ is full-rank, we can expect to find one solution of the n variables via Gaussian elimination. Usually, we can choose $m = \binom{n+d}{d}$, and the time complexity to find the solution is estimated as $\mathcal{O}(\binom{n+d}{d}^{\omega})$ field operations, where $2 \leq \omega \leq 3$.

For simplicity, let us denote FHE-friendly ciphers by $z = F(\mathrm{IV}, k)$ where $k = (k_1, \ldots, k_n) \in \mathbb{F}^n$ is the secret key and $z = (z_1, \ldots, z_s) \in \mathbb{F}^s$ is the output. For ciphers that use randomized components as stated above, F will ever-change for different inputs IV under the same k. To mount the linearization attack, the attacker first upper bounds the degree d of the expression of z in terms of k according to the fixed nonlinear layers. In this way, he can collect s equations in k of degree d from each input IV. By collecting $\binom{n+d}{d}$ equations according to $\binom{n+d}{d} \cdot \frac{1}{s}$ different inputs, he can apply the linearization technique to solve the overdefined system of equations, and compute the secret key k with time complexity $\mathcal{O}(\binom{n+d}{d}^{\omega})$. It is thus clear that d will have a significant impact on the time complexity of this attack.

The literature. There have been a number of FHE-friendly ciphers, like LowMC [ARS$^+$15], Kreyvrium [CCF$^+$18], FLIP [MJSC16], Rasta [DEG$^+$18], Masta [HKC$^+$20], DASTA [HL20], FASTA [CIR22], Pasta [DGH$^+$23], HERA [CHK$^+$21], Rubato [HKL$^+$22], Chaghri [AMT22] and Elisabeth-4 [CHMS22]. Among them, LowMC has been well analyzed in recent years [RST18, BBDV20, BBVY21, LIM21, Din21, LSW$^+$22, LMSI22], mainly due to its application to the post-quantum signature scheme Picnic [CDG$^+$17], and there are some successful full-round attacks [RST18, LIM21, Din21, LSW$^+$22, LMSI22]. Moreover, the original designs of FLIP, Chaghri and Elisabeth-4 have been broken with algebraic techniques [DLR16, LAW$^+$23, GBJR23]. Although there is a full-round attack [GAH$^+$23] on some variants of Rubato over the ring $\mathbb{Z}_q$ when q is a non-prime number, as clarified by the

[1] When the equations are over $\mathbb{F}_2$, there are at most $\sum_{i=1}^{d}\binom{n}{i}$ non-constant terms formed by the n variables.

designers of Rubato (i.e., on page 6 of [HKL+22]), q is implicitly assumed to be a prime number, i.e., Rubato is indeed defined over prime fields. For Rasta-like ciphers over $\mathbb{F}_2$ (e.g., Rasta, DASTA, FASTA), the most effective attack is the linearization attack by peeling off the last nonlinear layer such that the attacker can set up a system of equations of a much lower degree [LSMI21, LSMI22]. Such an attack highly relies on a special property of the nonlinear layer defined by the n-bit χ transform [Dae95]. However, this attack can only reduce the security margins of these ciphers by 1 round, and they are still secure. For Rasta-like ciphers over prime fields (e.g., Masta, Pasta), as far as we know, there is no third-party cryptanalysis, and the most effective attack is still the straighforward linearization attack which cannot reach full rounds. Different from Rasta-like ciphers which are based on the SPN structure and use randomized affine layers, HERA is the first FHE-friendly cipher with the randomized key schedule proposed at ASIACRYPT 2021, and Rubato published at EUROCRYPT 2022 adopts the same way to randomize its round keys, though the two ciphers are quite different. In addition, both HERA and Rubato are designed to be friendly to the CKKS FHE scheme proposed at ASIACRYPT 2017 [CKKS17], which is an FHE scheme for arithmetic of approximate numbers. Although HERA has been proposed for 2 years, we are not aware of any third-party cryptanalysis.

1.1 Our Contributions

We propose a new algebraic attack on HERA based on a different idea to construct low-degree equations by peeling off the last non-linear layer. Specifically, if we simply denote such randomized ciphers by $z = F(\mathrm{IV}, k)$ as stated above, all previous attacks on FLIP, Rasta, Rubato and Elisabeth-4 share the same feature that the attackers try to construct some low-degree equations in k from a single input-output pair (IV, z). After collecting sufficiently many (IV, z), they can set up an overdefined system of equations in k and solve it via the linearization technique. However, in our attacks, each low-degree equation in k is set up from 2 well-chosen input-output pairs (IV, z) and (IV', z'). In this sense, our attack is a chosen-IV attack[2]. Overall, our attack procedure can be abstracted as follows:

- **Offline phase**: Find sufficiently many good input pairs $(\mathrm{IV}, \mathrm{IV}')$ by the offline computation.
- **Online phase**: For each input pair $(\mathrm{IV}, \mathrm{IV}')$, evaluate them via F and obtain the corresponding output pair (z, z'). If (z, z') satisfy certain conditions, we can set up some low-degree equations in k.
- **Solving equations**: After collecting many low-degree equations, we solve them with the linearization technique.

Offline phase. In particular, finding multiple good input pairs at the offline phase is equivalent to the problem to find multiple collisions of a function $G : \mathbb{F}_2^{\ell_1} \mapsto \mathbb{F}_2^{\ell_2}$. Finding multiple collisions is a well-studied problem [vOW99, Din20] and it has some applications in cryptanalysis [vOW99, DEM19]. To find 2^{ℓ_3} collisions of G, a straightforward method is to consider $2^{\frac{\ell_2+\ell_3+1}{2}}$ different inputs of G. In this way, we can expect to obtain about $\frac{1}{2^{\ell_2}} \cdot 2^{\ell_2+\ell_3} = 2^{\ell_3}$ collisions in the ℓ_2-bit outputs. The time and memory complexity are both $\mathcal{O}(2^{\frac{\ell_2+\ell_3}{2}})$. Moreover, to ensure the existence of 2^{ℓ_3} collisions, we further need to constrain $\ell_1 \geq \frac{\ell_2+\ell_3+1}{2} \to 2\ell_1 \geq \ell_2 + \ell_3 + 1$ since in total $2^{\frac{\ell_2+\ell_3+1}{2}}$ different inputs are considered.

We also remark that with the advanced algorithm by van Oorschot and Wiener [vOW99] for this problem under $\ell_1 = \ell_2$, the best-known time-memory tradeoff for the time $\mathcal{T}$ and

[2] When sharing our results with the designers of HERA, they pointed out that our attack is a chosen-IV attack.

memory $\mathcal{M}$ is $\mathcal{T}^2 \cdot \mathcal{M} = \mathcal{O}(2^{2\ell_3} \cdot 2^{\ell_1})$ and $\mathcal{M} = \mathcal{O}(2^{\ell_3})$, where a poly-logarithmic factor in 2^{ℓ_1} is hidden in both $\mathcal{O}$ notation. Since we are not aware of the accurate hidden factors, and as the time complexity of our attacks at the offline phase, which is already optimal under this time-memory tradeoff, is close to that of brute force, we decide to only consider the straightforward method. With it, we can explicitly understand the security margins of HERA with different parameters, which is the main goal of this paper.

Online phase. At the online phase, we will construct low-degree equations in k. This is based on a simple property of the power map $x \mapsto x^\alpha$ over the finite field $\mathbb{F}$ when it is a permutation. Specifically, we have

$$\forall x_1, x_2 \in \mathbb{F}: \ (x_1 x_2)^\alpha = x_1^\alpha x_2^\alpha.$$

As will be shown in Lemma 2, if two outputs (y_1, y_2) where $y_1 = x_1^\alpha$ and $y_2 = x_2^\alpha$ satisfy $y_1 = \beta y_2$ and $\beta \neq 0$, we will have $x_1 = \beta^{\frac{1}{\alpha}} x_2$. If $\beta \in \mathbb{F}$ is a known constant, we then obtain a linear equation in (x_1, x_2). This is our core idea to peel off the last nonlinear layer and to set up low-degree equations from 2 chosen input-output pairs of HERA.

Solving equations. We rely on the linearization technique to solve an overdefined system of equations. As already stated before, this is just to perform Gaussian elimination on a large coefficient matrix. Abusing notation, let us assume that the matrix is of size $n \times n$. Then, Gaussian elimination on this matrix takes $\mathcal{O}(n^\omega)$ field operations. The straightforward way to perform Gaussian elimination will result in $\omega = 3$. With Strassen's divide-and-conquer method [Str69] for Gaussian elimination, we have $\omega = \log_2 7 \approx 2.8$. However, when designers evaluate the ciphers' resistance against the linearization attack in order to choose the secure number of rounds, $\omega = 2$ is commonly used [DEG+18, HKC+20, HL20, CHK+21, CIR22, HKL+22, DGH+23]. In this paper, we will give the corresponding costs under the three different values of ω.

Our results: HERA is defined over a prime filed $\mathbb{F}_p$ where $p > 2^{16}$. In addition, there are 4 different security levels, namely it provides λ bits of security where $\lambda \in \{80, 128, 192, 256\}$. By designers' analysis, HERA is secure under $\omega = 2$ for any $p > 2^{16}$. However, with our new algebraic attack, we can violate this claim. The consequences of our new attacks are briefly summarized as follows, and the detailed costs to break various parameters can be referred to Table 1.

- The security of HERA with $\lambda \in \{80, 128\}$ is not affected by this new attack.
- For $\lambda = 192$, we can break HERA for $p \leq 2^{18}$ under $\omega = 2$.
- For $\lambda = 256$, we can break HERA for $p \leq 2^{28}$ under $\omega = 2$.
- For $\lambda = 192$ and $\lambda = 256$, the security margin of HERA is reduced to 1 round from 2 rounds for $p \leq 2^{21}$ and $p \leq 2^{28}$ under $\omega = 2.8$, respectively.
- For $\lambda = 192$ and $\lambda = 256$, the security margin of HERA is reduced to 1 round from 2 rounds for $p \leq 2^{20}$ and $p \leq 2^{26}$ under $\omega = 3$, respectively.

Organization. The paper is organized as follows. In Sect. 2, we briefly introduce HERA. Then in Sect. 3, we describe the straightforward linearization attack on HERA and how its secure number of rounds was determined by designers. Next in Sect. 4, we explain the strategy to peel off the last nonlinear layer by using a simple property of the power map. In Sect. 5, the details of our attacks on HERA will be provided. At last, this paper is concluded in Sect. 6.

Table 1: Summary of the time complexity of our successful attacks on various parameters of HERA under $\omega \in \{2, 2.8, 3\}$.

λ	Rounds	ω	$\lceil \log_2 p \rceil$											
			17	18	19	20	21	22	23	24	25	26	27	28
192	6 (full)	2	2^{185}	2^{187}	–	–	–	–	–	–	–	–	–	–
	5	2.8	2^{167}	2^{175}	2^{179}	2^{180}	2^{187}	–	–	–	–	–	–	–
	5	3	2^{179}	2^{179}	2^{183}	2^{191}	–	–	–	–	–	–	–	–
256	7 (full)	2	2^{217}	2^{224}	2^{225}	2^{226}	2^{227}	2^{228}	2^{229}	2^{243}	2^{245}	2^{247}	2^{249}	2^{251}
	6	2.8	2^{234}	2^{234}	2^{234}	2^{234}	2^{234}	2^{234}	2^{234}	2^{235}	2^{243}	2^{249}	2^{250}	2^{251}
	6	3	2^{251}	2^{251}	2^{251}	2^{251}	2^{251}	2^{251}	2^{251}	2^{251}	2^{251}	2^{251}	–	–

2 Description of HERA

A graphic illustration of HERA is shown in Figure 1, where M is a fixed invertible matrix over $\mathbb{F}_p^{16\times 16}$, and S is the S-box defined by $S(x) = x^3$ over $\mathbb{F}_p$, where $p > 2^{16}$ is a prime number. Since our algebraic attacks are irrelevant to M and we only exploit its invertibility, we refer the interested readers to [CHK+21] for more details of how M is constructed. Throughout this paper, we denote the set of integers i satisfying $i_0 \leq i < i_1$ and $i_0 \leq i \leq i_1$ by $[i_0, i_1)$ and $[i_0, i_1]$, respectively.

The HERA state is composed of 16 words in $\mathbb{F}_p$. For r rounds HERA with λ bits of security, it takes a nonce $\texttt{nc} \in [0, 2^\lambda)$, a counter $\texttt{cnt} \in [0, 2^{\frac{\lambda}{2}})$, a key $k = (k_1, \ldots, k_{16}) \in \mathbb{F}_p^{16}$ and a predefined constant vector $a = (a_1, \ldots, a_{16}) = (1, \ldots, 16)$ as input. The output of HERA used as the keystream is denoted by $z = (z_1, \ldots, z_{16}) \in \mathbb{F}_p^{16}$. For convenience, we denote the concatenation of $\texttt{nc}$ and $\texttt{cnt}$ by $\texttt{nc}||\texttt{cnt}$, and we further let

$$\mathrm{IV} = \texttt{nc}||\texttt{cnt}.$$

In this way, r rounds of HERA can be simply defined as follows:

$$z = \mathsf{HERA}_r(\mathrm{IV}, k).$$

Similar to Rasta, the nonce $\texttt{nc}$ and the counter $\texttt{cnt}$ will be different for each keystream generated under the same key k. The detailed procedure is described below.

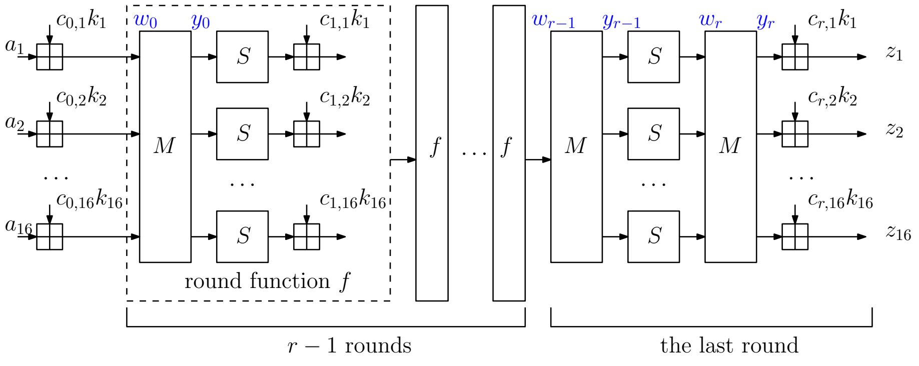

Figure 1: Illustration of HERA

Generating random vectors $(c_0, \ldots, c_r)$. To compute the output $z = (z_1, \ldots, z_{16})$, first, IV is set as the input to an XOF and the outputs of the XOF are $r + 1$ vectors[3]

[3] Although it is not clearly clarified by the designers, each element in the vector should be nonzero. Otherwise, there is an obvious weakness by having some $c_{i,j} = 0$ via the offline computation.

$c_i = (c_{i,1}, \ldots, c_{i,16}) \in \mathbb{F}_p^{16}$ for $0 \le i \le r$. For convenience, this procedure is represented as follows:

$$(c_0, \ldots, c_r) = \mathrm{XOF}(\mathrm{IV}).$$

Computing the output $(z_1, \ldots, z_{16})$. As shown in Figure 1, denote the input and output of the $(i+1)$-th linear layer M by $w_i = (w_{i,1}, \ldots, w_{i,16}) \in \mathbb{F}_p^{16}$ and $y_i = (y_{i,1}, \ldots, y_{i,16}) \in \mathbb{F}_p^{16}$ for $0 \le i \le r$, respectively. In this way, r rounds of HERA can be described by the following formulas:

$$\begin{aligned}
\forall j \in [1,16]: &\quad w_{0,j} = a_j + c_{0,j}k_j,\\
\forall i \in [0,r]: &\quad y_i = M(w_i),\\
\forall i \in [1,r-1], \forall j \in [1,16]: &\quad w_{i,j} = S(y_{i-1,j}) + c_{i,j}k_j,\\
\forall j \in [1,16]: &\quad w_{r,j} = S(y_{r-1,j}),\\
\forall j \in [1,16]: &\quad z_j = y_{r,j} + c_{r,j}k_j.
\end{aligned}$$

Number of rounds and data limit. The designers specified 4 different parameters of the number of rounds r for different security levels λ, which are

$$(r, \lambda) = \{(4,80), (5,128), (6,192), (7,256)\}.$$

Moreover, each key can only be used to encrypt up to $2^{\frac{\lambda}{2}}$ message blocks, which is why we have $\mathrm{cnt} \in [0, 2^{\frac{\lambda}{2}})$.

3 Straightforward Linearization Attack on HERA

The main feature of HERA is that the round keys are randomized by element-wise multiplying a random vector with the master key. Different from Rasta, its linear layers are fixed and they are the same in each round, which is defined by a fixed matrix M. For such a new way to randomize the cipher, many traditional techniques also failed for HERA since each keystream z is computed under a different permutation for the same key. The most effective attack against HERA, similar to the analysis of Rasta, is the linearization attack.

For r rounds of HERA, the keystream z can be trivially expressed as 16 equations over $\mathbb{F}_p$ in 16 unknowns $(k_1, \ldots, k_{16})$. The degree of these equations is 3^r since the degree of the S-box is 3. To crack r rounds of HERA with the linearization attack, the attacker first collects

$$\mathcal{N}_0(r) = \binom{16+3^r}{3^r} \tag{1}$$

equations by making about $\frac{\mathcal{N}_0(r)}{16}$ encryption queries under the same key, where IV will be different for each query. Then, the time complexity to solve the key is estimated as

$$\mathcal{T}_0(r, \omega) = \binom{16+3^r}{3^r}^{\omega}, \tag{2}$$

where ω is the algebra constant for Gaussian elimination and it satisfies $2 \le \omega \le 3$.

To determine the secure number of rounds r for λ bits of security, the designers of HERA choose the minimal r such that

$$\mathcal{T}_0(r, 2) > 2^{\lambda}.$$

According to [CHK+21], the recommended parameters r for different λ are listed in Table 2.

Table 2: The parameters for HERA

λ	80	128	192	256
r	4	5	6	7

Guess-and-determine attack. The designers also took into account the guess-and-determine (GnD) attack. In this attack, the attacker can first guess j unknowns and then solve the equations of degree 3^r in $16-j$ unknowns for r rounds of HERA. The time complexity of this attack is

$$p^j \times \binom{16-j+3^r}{3^r}^{\omega}.$$

It was shown by the designers that if $p > 2^{16}$, this GnD attack can never be faster than the above straightforward linearization attack under $\omega = 2$.

Table 3: The time complexity to break r rounds of HERA with different (r, ω)

λ r	80 4	128 5	192 6	256 7
brute force	p^{16}	p^{16}	p^{16}	p^{16}
$\mathcal{T}_0(r,2)$	2^{119}	2^{167}	2^{217}	2^{267}
$\mathcal{T}_0(r-1,2)$	2^{76}	2^{119}	2^{167}	2^{217}
$\mathcal{T}_0(r,2.8)$	2^{167}	2^{234}	2^{303}	2^{374}
$\mathcal{T}_0(r-1,2.8)$	2^{107}	2^{167}	2^{234}	2^{303}
$\mathcal{T}_0(r-2,2.8)$	2^{59}	2^{107}	2^{167}	2^{234}
$\mathcal{T}_0(r,3)$	2^{179}	2^{251}	2^{325}	2^{401}
$\mathcal{T}_0(r-1,3)$	2^{114}	2^{179}	2^{251}	2^{325}
$\mathcal{T}_0(r-2,3)$	2^{63}	2^{114}	2^{179}	2^{251}

Concrete time complexity. For the parameters in Table 2, we can do some simple calculations for $\mathcal{T}_0(r, \omega)$ with different (r, ω), as shown in Table 3. Therefore, with $\omega = 2$, the security margin for all versions of HERA is only 1 round. With $\omega \in \{2.8, 3\}$, the security margins are 2 rounds. We also calculated the time complexity of the linearization attacks on HERA under $\omega = \{2.8, 3\}$ by considering the GnD strategy, which still suggests that the security margins are 2 rounds, i.e., it cannot help improve the straightforward linearization attack[4].

In this sense, if we can peel off the last nonlinear layer with negligible extra costs, i.e., set up equations of degree 3^{r-1} for r rounds of HERA, attacking r rounds is equivalent to attacking $r-1$ rounds with the linearization technique. In this way, it is possible to break HERA under $\omega = 2$, and reduce the security margins to only 1 round under $\omega = \{2.8, 3\}$.

4 Peeling off the Last Nonlinear Layer

In this section, we start explaining how to set up equations of degree 3^{r-1} for r rounds of HERA. Our attacks on HERA intensively rely on the following two trivial lemmas.

Lemma 1. *Let $P(x)$ be a permutation over a finite field $\mathbb{F}$. For two inputs x_1 and x_2, if $P(x_1) = P(x_2)$, we have $x_1 = x_2$.*

[4] We note that if p is small, e.g., $p \approx 2^{16}$, the best attack on HERA with 256-bit security is the brute force attack, not the linearization attack.

Lemma 2. *Let $P(x) = x^\alpha$ be a permutation over a finite field $\mathbb{F}$. For two inputs x_1 and x_2, if $P(x_1) = \beta P(x_2)$ where $\beta \in \mathbb{F}$ and $\beta \neq 0$, we have $x_1 = \beta^{\frac{1}{\alpha}} x_2$.*

Proof. First, we have $(\beta^{\frac{1}{\alpha}} x_2)^\alpha = \beta x_2^\alpha$. Since

$$P(x_1) = \beta P(x_2) = \beta x_2^\alpha = P(\beta^{\frac{1}{\alpha}} x_2)$$

and that $P(x)$ is a permutation, we have

$$x_1 = \beta^{\frac{1}{\alpha}} x_2.$$

□

Based on Lemma 1 and Lemma 2, a linear relation inside an input pair of $P(x)$ can be derived according to the corresponding output pair, even if the outputs are masked with a secret. For example, we consider

$$y_1 = P(x_1) + t_1 \nu, \quad y_2 = P(x_2) + t_2 \nu,$$

where $(t_1, t_2) \in \mathbb{F}^2$ are randomly generated public constants and $\nu \in \mathbb{F}$ is the secret. If we can generate (t_1, t_2) such that $t_1 = t_2$, and observe $y_1 = y_2$ in this case, we directly obtain $x_1 = x_2$, which is a linear relation inside (x_1, x_2). Moreover, if $P(x) = x^\alpha$, we can further relax the conditions on (t_1, t_2, y_1, y_2) in order to derive a linear relation inside (x_1, x_2). Specifically, if $t_1 = \beta t_2$ and $y_1 = \beta y_2$, where $\beta \in \mathbb{F}$ is a known nonzero value, we will have $x_1 = \beta^{\frac{1}{\alpha}} x_2$, which also corresponds to a linear relation.

4.1 Application to HERA

Based on the above simple observations, we describe how to peel off the last nonlinear layer and set up equations of degree 3^{r-1} for r rounds of HERA.

Consider a pair of keystreams (z, z') defined by

$$z \quad = \quad \mathsf{HERA}_r(\mathrm{IV}, k), \quad z' = \mathsf{HERA}_r(\mathrm{IV}', k).$$

Furthermore, let

$$(c_0, \ldots, c_r) = \mathrm{XOF}(\mathrm{IV}), \quad (c'_0, \ldots, c'_r) = \mathrm{XOF}(\mathrm{IV}').$$

For convenience, for the keystream z' generated with the input (r, k, IV'), the input to the $(i+1)$-th M is denoted by $w'_i = (w'_{i,1}, \ldots, w'_{i,16}) \in \mathbb{F}_p^{16}$ where $0 \leq i \leq r$. In addition, we define $z' = (z'_1, \ldots, z'_{16}) \in \mathbb{F}_p^{16}$, and $c'_i = (c'_{i,1}, \ldots, c'_{i,16}) \in \mathbb{F}_p^{16}$ for $0 \leq i \leq r$.

For the matrix M over $\mathbb{F}_p^{16\times 16}$, denote its inverse by M^{-1}. Moreover, we denote the element at the i-th row and the j-th column of M^{-1} by $M^{-1}[i][j]$, where $i, j \in [1, 16]$. For the output pair (w_r, w'_r) of the last nonlinear layer, we then have

$$\begin{aligned} w_{r,i} &= \sum_{j=1}^{16} M^{-1}[i][j](z_j - c_{r,j} k_j) = \sum_{j=1}^{16} M^{-1}[i][j] z_j - \sum_{j=1}^{16} M^{-1}[i][j] c_{r,j} k_j, \\ w'_{r,i} &= \sum_{j=1}^{16} M^{-1}[i][j](z'_j - c'_{r,j} k_j) = \sum_{j=1}^{16} M^{-1}[i][j] z'_j - \sum_{j=1}^{16} M^{-1}[i][j] c'_{r,j} k_j, \end{aligned}$$

where $i \in [1, 16]$.

Linearization via Lemma 1. If

$$\begin{aligned} \exists \hat{\imath} \in [1,16]: &\quad \sum_{j=1}^{16} M^{-1}[\hat{\imath}][j]z_j = \sum_{j=1}^{16} M^{-1}[\hat{\imath}][j]z'_j, \\ \forall j \in [1,16]: &\quad c_{r,j} = c'_{r,j}, \end{aligned}$$

there will be

$$w_{r,\hat{\imath}} = w'_{r,\hat{\imath}}.$$

In other words, when generating such a keystream pair (z, z'), the inputs to the $\hat{\imath}$-th S-box at the last nonlinear layer are identical. Hence, we can set up an equation of degree 3^{r-1} in terms of k to describe this property.

Linearization via Lemma 2. Since the power map $S(x) = x^3$ is used in HERA, and the prime number $p > 2^{16}$ is chosen such that $S(x)$ is a permutation over $\mathbb{F}_p$, we can further use Lemma 2 to relax the conditions to linearize the last nonlinear layer based on a keystream pair. Specifically, if there exists a nonzero $\beta \in \mathbb{F}_p$ such that

$$\begin{aligned} \exists \hat{\imath} \in [1,16]: &\quad \sum_{j=1}^{16} M^{-1}[\hat{\imath}][j]z_j = \beta \cdot (\sum_{j=1}^{16} M^{-1}[\hat{\imath}][j]z'_j), \\ \forall j \in [1,16]: &\quad c_{r,j} = \beta c'_{r,j}, \end{aligned}$$

there will be

$$w_{r,\hat{\imath}} = \beta w'_{r,\hat{\imath}}.$$

In other words, for such a keystream pair (z, z'), the input pair denoted by $(y_{r-1,\hat{\imath}}, y'_{r-1,\hat{\imath}})$ to the $\hat{\imath}$-th S-box at the last nonlinear layer satisfy

$$y_{r-1,\hat{\imath}} = \beta^{\frac{1}{3}} y'_{r-1,\hat{\imath}}.$$

Therefore, we can also set up an equation of degree 3^{r-1} in terms of k to describe the above relation inside $(y_{r-1,\hat{\imath}}, y'_{r-1,\hat{\imath}})$.

Linearization via Lemma 2 and guessing strategies. It is possible to observe that the following conditions are too strict as they correspond to about $16 \log_2 p$ bit conditions:

$$\forall j \in [1,16] : c_{r,j} = \beta c'_{r,j}.$$

To relax these conditions, we can guess n_1 words in k, e.g., we can guess $(k_1, \ldots, k_{n_1})$. For each guess, we can compute

$$\delta = \sum_{j=1}^{n_1} M^{-1}[i][j]c_{r,j}k_j, \quad \delta' = \sum_{j=1}^{n_1} M^{-1}[i][j]c'_{r,j}k_j.$$

Then, we only need to consider the following conditions for each guess in order to linearize the last nonlinear layer:

$$\begin{aligned} \exists \hat{\imath} \in [1,16]: &\quad \sum_{j=1}^{16} M^{-1}[\hat{\imath}][j]z_j - \delta = \beta \cdot (\sum_{j=1}^{16} M^{-1}[\hat{\imath}][j]z'_j - \delta'), \\ \forall j \in [n_1+1,16]: &\quad c_{r,j} = \beta c'_{r,j}, \end{aligned}$$

Specifically, under these conditions, for the corresponding keystream pair (z, z'), we will again have

$$w_{r,\hat{\imath}} = \beta w'_{r,\hat{\imath}}, \quad y_{r-1,\hat{\imath}} = \beta^{\frac{1}{3}} y'_{r-1,\hat{\imath}}.$$

5 Algebraic Attacks on HERA

In our attacks, we assume that the attacker has full control over IV $\in \mathbb{F}_2^{\frac{3\lambda}{2}}$, which corresponds to a chosen-IV attack. According to the above analysis, we always have two types of conditions to satisfy:

Type-1: conditions on the concrete keystream pair (z, z');

Type-2: conditions on the randomly sampled vectors c_r and c'_r.

Observe that Type-2 conditions can be satisfied at the offline phase, as they are irrelevant to k, and only depend on a public extended output function XOF with a controllable input IV. As for Type-1 conditions, we need to check the conditions on the keystream and computing it requires the knowledge of k. Therefore, Type-1 conditions can only be checked at the online phase. In the following, we provide the complexity analysis of the offline and online phases, respectively.

5.1 Offline Phase

We consider the linearization of the last nonlinear layer via Lemma 2 and guessing strategies, as it also covers the case when only Lemma 2 is used by setting $n_1 = 0$. Specifically, we consider the cost to generate 2^b pairs (c_r, c'_r) satisfying the following conditions where $\beta \in \mathbb{F}_p$ can be any nonzero value:

$$\forall j \in [n_1 + 1, 16] : \qquad c_{r,j} = \beta c'_{r,j}.$$

For such a pair, we have

$$(c_{r,n_1+1}, c_{r,n_1+2}, \ldots, c_{r,16}) = (\beta c'_{r,n_1+1}, \beta c'_{r,n_1+2}, \ldots, \beta c'_{r,16}).$$

Hence, we can consider the following pair

$$(1, \frac{c_{r,n_1+2}}{c_{r,n_1+1}}, \ldots, \frac{c_{r,16}}{c_{r,n_1+1}}), \quad (1, \frac{c'_{r,n_1+2}}{c'_{r,n_1+1}}, \ldots, \frac{c'_{r,16}}{c'_{r,n_1+1}}).$$

If there is

$$\forall j \in [n_1 + 2, 16] : \ \frac{c_{r,j}}{c_{r,n_1+1}} = \frac{c'_{r,j}}{c'_{r,n_1+1}}, \tag{3}$$

there must exist $\beta = \frac{c_{r,n_1+1}}{c'_{r,n_1+1}}$ such that

$$\forall j \in [n_1 + 1, 16] : c_{r,j} = \beta c'_{r,j}.$$

In other words, we can equivalently consider the problem to generate 2^b collisions in $(15 - n_1) \times \lceil \log_2 p \rceil$ output bits of the used XOF seeded with IV of $\frac{3\lambda}{2}$ bits.

Costs of generating multiple collisions. Let

$$\ell = (15 - n_1) \times \lceil \log_2 p \rceil. \tag{4}$$

To generate 2^b collisions in ℓ bits of the output of a hash function, whose input length is $\frac{3\lambda}{2}$ bits, the following condition should hold:

$$3\lambda \geq b + \ell + 1. \tag{5}$$

As previously mentioned, the time complexity to find these 2^b collisions is $2^{\frac{b+\ell+1}{2}}$. Here, we only consider the straightforward method to find multiple collisions for its simplicity, which requires memory complexity of $2^{\frac{b+\ell+1}{2}}$. Therefore, the time complexity denoted by $\mathcal{T}_{\text{offline}}$ and memory complexity denoted by $\mathcal{M}_{\text{offline}}$ of the offline phase are as follows:

$$\mathcal{T}_{\text{offline}} = 2^{\frac{b+\ell+1}{2}}, \quad \mathcal{M}_{\text{offline}} = 2^{\frac{b+\ell+1}{2}}.$$

Specifically, we can prepare $2^{\frac{b+\ell+1}{2}}$ different randomly chosen values of IV. For each of them, compute the corresponding $r+1$ vectors $(c_0, \ldots, c_r)$, and store the corresponding vector c_r^f as well as its associated IV, where c_r^f is defined by

$$c_r^f = (\frac{c_{r,n_1+2}}{c_{r,n_1+1}}, \ldots, \frac{c_{r,16}}{c_{r,n_1+1}}) \in \mathbb{F}_p^{15-n_1}.$$

Then, it is expected to obtain $\frac{1}{2^\ell} \times 2^{\ell+b} = 2^b$ collisions in c_r^f, since there are in total $2^{\ell+b}$ pairs formed by $2^{\frac{b+\ell+1}{2}}$ different IV.

5.2 Online Phase

After the offline phase, we have collected 2^b pairs (IV, IV') such that the corresponding pair (c_r, c_r') satisfies Equation 3. We should emphasize that for each different pair (c_r, c_r'), the corresponding constant scalar $\beta = \frac{c_{r,n_1+1}}{c'_{r,n_1+1}}$ may be different, but this does not affect the attack.

Generating keystreams. For each pair (IV, IV'), the following procedure is performed.

Step 1: Compute

$$(c_0, \ldots, c_r) = \text{XOF}(\text{IV}), \quad (c_0', \ldots, c_r') = \text{XOF}(\text{IV}')$$

and

$$\beta = \frac{c_{r,n_1+1}}{c'_{r,n_1+1}}.$$

Step 2: Compute the keystream pair (z, z'), and get the corresponding 16 pairs (h_i, h_i') for $i \in [1, 16]$, where

$$h_i = \sum_{j=1}^{16} M^{-1}[i][j]z_j, \quad h_i' = \sum_{j=1}^{16} M^{-1}[i][j]z_j'.$$

Store $(c_r, c_r', \beta, h_1, h_1', \ldots, h_{16}, h_{16}')$ in a table denoted by `TAB`.

Constructing equations. After constructing `TAB` which contains 2^b entries, we then guess $(k_1, \ldots, k_{n_1})$. For each guess, the following procedure is performed:

Step 1: For each element $(c_r, c_r', \beta, h_1, h_1', \ldots, h_{16}, h_{16}')$ in `TAB`, compute

$$\delta = \sum_{j=1}^{n_1} M^{-1}[i][j]c_{r,j}k_j, \quad \delta' = \sum_{j=1}^{n_1} M^{-1}[i][j]c'_{r,j}k_j.$$

If

$$\exists i \in [1, 16]: \ h_i - \delta = \beta(h_i' - \delta'), \tag{6}$$

move to Step 2. Otherwise, pick another element in `TAB` and repeat.

Step 2: Set up an equation in terms of $16 - n_1$ unknowns $(k_{n_1+1}, \ldots, k_{16})$ of degree 3^{r-1} since the input pair $(y_{r-1,i}, y'_{r-1,i})$ for the i-th S-box at the last nonlinear layer satisfies:

$$y_{r-1,i} = \beta^{\frac{1}{3}} y'_{r-1,i}.$$

Since Equation 6 holds with probability of about $\frac{16}{p}$, for each guess of $(k_1, \ldots, k_{n_1})$, we can expect to set up about $\frac{16}{p} \times 2^b$ equations in $16 - n_1$ unknowns of degree 3^{r-1}.

Solving equations with the linearization technique. To solve equations in $16 - n_1$ unknowns of degree 3^{r-1} with the linearization technique, we need

$$\binom{16 - n_1 + 3^{r-1}}{3^{r-1}}$$

equations. Therefore, the following condition should hold:

$$\frac{16}{p} \times 2^b \geq \binom{16 - n_1 + 3^{r-1}}{3^{r-1}}. \tag{7}$$

The time complexity to solve such a system of equations is estimated as

$$\binom{16 - n_1 + 3^{r-1}}{3^{r-1}}^{\omega}.$$

For each solution of $(k_{n_1+1}, \ldots, k_{16})$ obtained under the guess of $(k_1, \ldots, k_{n_1})$, we need to further check the correctness of $(k_1, \ldots, k_{16})$ by using 2 keystreams and this cost is negligible. Therefore, the time complexity of the online phase[5] is

$$\mathcal{T}_{\text{online}} = p^{n_1} \times 2^{b+1} + p^{n_1} \times \binom{16 - n_1 + 3^{r-1}}{3^{r-1}}^{\omega}.$$

For the memory complexity of the online phase, it is simply the cost to store the $\binom{16-n_1+3^{r-1}}{3^{r-1}}$ equations in $16 - n_1$ variables of degree 3^{r-1}, and the cost to store the 2^b collisions. Hence, its memory complexity is

$$\mathcal{M}_{\text{online}} = 2^b + \binom{16 - n_1 + 3^{r-1}}{3^{r-1}}^2.$$

5.3 Impact on the Security of HERA

According to the above analysis, to crack r rounds of HERA with the security level $\lambda \in \{80, 128, 192, 256\}$, the total time complexity and memory complexity can be formulated as follows:

$$\begin{aligned}
\mathcal{T}_{\text{offline}} + \mathcal{T}_{\text{online}} &= 2^{\frac{b+\ell+1}{2}} + p^{n_1} \times 2^{b+1} + p^{n_1} \times \binom{16 - n_1 + 3^{r-1}}{3^{r-1}}^{\omega}, \\
\mathcal{M}_{\text{offline}} + \mathcal{M}_{\text{online}} &= 2^{\frac{b+\ell+1}{2}} + 2^b + \binom{16 - n_1 + 3^{r-1}}{3^{r-1}}^2,
\end{aligned}$$

where $\ell = (15 - n_1) \times \lceil \log_2 p \rceil$. It is clear that the memory complexity is always smaller than the time complexity due to $\omega \geq 2$.

[5] We do not consider the equivalent cost to perform r rounds of HERA, as this is what the designers used to estimate the time complexity of the linearization attack. If taking this into account, our time complexity only becomes lower.

In addition, the following conditions should hold:

$$\begin{cases} 3\lambda \geq b + \ell + 1 \\ \dfrac{16}{p} \times 2^b \geq \dbinom{16 - n_1 + 3^{r-1}}{3^{r-1}} \\ b + 1 \leq \dfrac{\lambda}{2} \end{cases} \tag{8}$$

The last condition is caused by the restriction that the same key can only be used to generate at most $2^{\frac{\lambda}{2}}$ keystreams.

To achieve the optimal time complexity of the key-recovery attack, (b, n_1) should be carefully chosen according to different $(r, \lambda, \log_2 p)$. For this purpose, we can iterate 16 choices[6] of n_1, namely $n_1 \in [0, 15]$, and compute the corresponding minimal b such that

$$\frac{16}{p} \times 2^b \geq \binom{16 - n_1 + 3^{r-1}}{3^{r-1}}.$$

Then, we check the conditions $b + 1 \leq \frac{\lambda}{2}$ an $3\lambda \geq b + \ell + 1$. If they hold, compute the corresponding $\mathcal{T}_{\text{offline}} + \mathcal{T}_{\text{online}}$. Finally, choose valid (b, n_1) that can minimize $\mathcal{T}_{\text{offline}} + \mathcal{T}_{\text{online}}$. Our calculation results are shown in Table 4. The corresponding (b, n_1) are given in Table 5. Note that the accurate value of p^{n_1} depends on p. However, in our calculations, if p satisfies $\lceil \log_2 p \rceil = \mu$, we simply treat p as 2^μ and hence p^{n_1} will be directly $2^{\mu + n_1}$.

Impacts of our attacks. Although we can peel off the last nonlinear layer via multiple collisions, it should be emphasized that in most cases, finding these collisions at the offline phase is much more costly than that of solving the key at the online phase. For this reason, the feasibility of our attacks is closely related to the length of p. According to Table 4, we can draw the following conclusions for the security of HERA:

- For HERA with $\lambda \in \{80, 128\}$, the security is not affected by our attacks.
- Under $\omega = 2$ which is used by designers to determine the secure number of rounds, we show that HERA is insecure when i) $\lambda = 192$ and $p \leq 2^{18}$; ii) $\lambda = 256$ and $p \leq 2^{28}$.
- Under $\omega = 2.8$, the security margins of HERA with the following parameters are reduced to 1 round: i) $\lambda = 192$ and $p \leq 2^{21}$; ii) $\lambda = 256$ and $p \leq 2^{28}$.
- Under $\omega = 3$, the security margins of HERA with the following parameters are reduced to 1 round: i) $\lambda = 192$ and $p \leq 2^{20}$; ii) $\lambda = 256$ and $p \leq 2^{26}$.

Hence, our attacks significantly improve the linearization attacks on HERA with more than 192 bits of security, if p is not that large. In addition, as shown in Table 5, in some cases, the optimal time complexity is obtained with $n_1 > 0$. This suggests that the GnD strategy can be successfully used to improve the linearization attack, which further contradicts the designers' claim.

5.4 Experimental Verification

We provide a proof-of-concept attack to verify the correctness of our technique. It is worth noting that the number of potential monomials grows rapidly as the number of rounds increases, and the cost to generate multiple collisions increases exponentially as the length of p increases. For the sake of efficiency, we only consider 2 rounds of HERA over a small prime field $\mathbb{F}_{17}$. The primary focus of the linearization attack centers around verifying the

[6] If $n_1 = 16$, it is simply the brute force attack.

Table 4: The time complexity to break r rounds of HERA, where - represents that our new attacks cannot work due to Equation 8. For simplicity, we give the logarithm of the time complexity to base 2 in this table. Moreover, the successful full-round attacks under $\omega = 2$ are colored in red, while the successful reduced-round attacks under $\omega \in \{2.8, 3\}$ are colored in blue.

$(\lceil \log_2 p \rceil, \omega)$ \ (r, λ)	$(3, 80)$	$(4, 80)$	$(4, 128)$	$(5, 128)$	$(5, 192)$	$(6, 192)$	$(6, 256)$	$(7, 256)$
$(17, 2)$	-	-	128	188	147	185	176	217
$(17, 2.8)$	-	-	138	222	167	245	234	283
$(17, 3)$	-	-	140	230	179	260	251	284
$(18, 2)$	-	-	135	194	154	187	177	224
$(18, 2.8)$	-	-	145	228	175	247	234	299
$(18, 3)$	-	-	145	236	179	262	251	299
$(19, 2)$	-	-	140	200	160	200	183	225
$(19, 2.8)$	-	-	151	234	179	256	234	307
$(19, 3)$	-	-	152	242	183	271	251	315
$(20, 2)$	-	-	144	219	163	203	191	226
$(20, 2.8)$	-	-	154	251	180	259	234	308
$(20, 3)$	-	-	159	259	191	274	251	328
$(21, 2)$	-	-	148	226	166	206	193	227
$(21, 2.8)$	-	-	157	258	187	262	234	309
$(21, 3)$	-	-	164	266	192	277	251	329
$(22, 2)$	-	-	189	278	170	209	195	228
$(22, 2.8)$	-	-	212	301	194	265	234	310
$(22, 3)$	-	-	218	307	194	280	251	330
$(23, 2)$	-	-	335	358	176	226	200	229
$(23, 2.8)$	-	-	335	358	198	279	234	311
$(23, 3)$	-	-	335	358	202	293	251	331
$(24, 2)$	-	-	-	-	183	230	207	243
$(24, 2.8)$	-	-	-	-	200	283	235	320
$(24, 3)$	-	-	-	-	209	297	251	340
$(25, 2)$	-	-	-	-	189	234	214	245
$(25, 2.8)$	-	-	-	-	203	287	243	322
$(25, 3)$	-	-	-	-	213	301	251	342
$(26, 2)$	-	-	-	-	196	238	221	247
$(26, 2.8)$	-	-	-	-	210	291	249	324
$(26, 3)$	-	-	-	-	215	305	251	344
$(27, 2)$	-	-	-	-	202	242	224	249
$(27, 2.8)$	-	-	-	-	217	295	250	326
$(27, 3)$	-	-	-	-	217	309	259	346
$(28, 2)$	-	-	-	-	209	265	227	251
$(28, 2.8)$	-	-	-	-	224	315	251	328
$(28, 3)$	-	-	-	-	224	327	266	348
$(29, 2)$	-	-	-	-	214	270	230	270
$(29, 2.8)$	-	-	-	-	231	320	258	343
$(29, 3)$	-	-	-	-	231	332	267	361

Table 5: The corresponding (b, n_1) for results in Table 4.

(r,λ) $(\lceil \log_2 p \rceil, \omega)$	$(4,128)$	$(5,128)$	$(5,192)$	$(6,192)$	$(6,256)$	$(7,256)$
$(17,2)$	$(51,3)$	$(60,6)$	$(72,2)$	$(93,2)$	$(97,1)$	$(126,0)$
$(17,2.8)$	$(53,2)$	$(60,6)$	$(77,0)$	$(93,2)$	$(101,0)$	$(27,15)$
$(17,3)$	$(53,2)$	$(60,6)$	$(77,0)$	$(93,2)$	$(101,0)$	$(27,15)$
$(18,2)$	$(52,3)$	$(61,6)$	$(73,2)$	$(94,2)$	$(98,1)$	$(121,1)$
$(18,2.8)$	$(54,2)$	$(61,6)$	$(78,0)$	$(94,2)$	$(102,0)$	$(28,15)$
$(18,3)$	$(54,2)$	$(61,6)$	$(78,0)$	$(94,2)$	$(102,0)$	$(28,15)$
$(19,2)$	$(51,4)$	$(62,6)$	$(71,3)$	$(91,3)$	$(99,1)$	$(122,1)$
$(19,2.8)$	$(53,3)$	$(62,6)$	$(76,1)$	$(91,3)$	$(103,0)$	$(122,1)$
$(19,3)$	$(55,2)$	$(62,6)$	$(79,0)$	$(91,3)$	$(103,0)$	$(29,15)$
$(20,2)$	$(52,4)$	$(60,7)$	$(72,3)$	$(92,3)$	$(100,1)$	$(123,1)$
$(20,2.8)$	$(54,3)$	$(60,7)$	$(77,1)$	$(92,3)$	$(104,0)$	$(123,1)$
$(20,3)$	$(56,2)$	$(60,7)$	$(80,0)$	$(92,3)$	$(104,0)$	$(123,1)$
$(21,2)$	$(53,4)$	$(61,7)$	$(73,3)$	$(93,3)$	$(97,2)$	$(124,1)$
$(21,2.8)$	$(55,3)$	$(61,7)$	$(78,1)$	$(93,3)$	$(105,0)$	$(124,1)$
$(21,3)$	$(55,3)$	$(61,7)$	$(78,1)$	$(93,3)$	$(105,0)$	$(124,1)$
$(22,2)$	$(51,6)$	$(51,10)$	$(74,3)$	$(94,3)$	$(98,2)$	$(125,1)$
$(22,2.8)$	$(51,6)$	$(51,10)$	$(79,1)$	$(94,3)$	$(106,0)$	$(125,1)$
$(22,3)$	$(51,6)$	$(51,10)$	$(79,1)$	$(94,3)$	$(106,0)$	$(125,1)$
$(23,2)$	$(35,13)$	$(35,14)$	$(75,3)$	$(90,4)$	$(99,2)$	$(126,1)$
$(23,2.8)$	$(35,13)$	$(35,14)$	$(78,2)$	$(90,4)$	$(107,0)$	$(126,1)$
$(23,3)$	$(35,13)$	$(35,14)$	$(80,1)$	$(90,4)$	$(107,0)$	$(126,1)$
$(24,2)$	-	-	$(76,3)$	$(91,4)$	$(100,2)$	$(122,2)$
$(24,2.8)$	-	-	$(79,2)$	$(91,4)$	$(108,0)$	$(122,2)$
$(24,3)$	-	-	$(81,1)$	$(91,4)$	$(108,0)$	$(122,2)$
$(25,2)$	-	-	$(77,3)$	$(92,4)$	$(101,2)$	$(123,2)$
$(25,2.8)$	-	-	$(80,2)$	$(92,4)$	$(109,0)$	$(123,2)$
$(25,3)$	-	-	$(80,2)$	$(92,4)$	$(109,0)$	$(123,2)$
$(26,2)$	-	-	$(78,3)$	$(93,4)$	$(102,2)$	$(124,2)$
$(26,2.8)$	-	-	$(81,2)$	$(93,4)$	$(106,1)$	$(124,2)$
$(26,3)$	-	-	$(81,2)$	$(93,4)$	$(110,0)$	$(124,2)$
$(27,2)$	-	-	$(79,3)$	$(94,4)$	$(99,3)$	$(125,2)$
$(27,2.8)$	-	-	$(82,2)$	$(94,4)$	$(107,1)$	$(125,2)$
$(27,3)$	-	-	$(82,2)$	$(94,4)$	$(111,0)$	$(125,2)$
$(28,2)$	-	-	$(80,3)$	$(91,5)$	$(100,3)$	$(126,2)$
$(28,2.8)$	-	-	$(83,2)$	$(91,5)$	$(108,1)$	$(126,2)$
$(28,3)$	-	-	$(83,2)$	$(91,5)$	$(108,1)$	$(126,2)$
$(29,2)$	-	-	$(78,4)$	$(92,5)$	$(101,3)$	$(121,3)$
$(29,2.8)$	-	-	$(84,2)$	$(92,5)$	$(109,1)$	$(121,3)$
$(29,3)$	-	-	$(84,2)$	$(92,5)$	$(109,1)$	$(121,3)$

linear independence of the constructed equations. Therefore, the primary purpose of these experiments is to confirm the linear independence of the equations constructed with our method.

In the experiments, we guess 10 coordinates out of 16 key coordinates. Based on our method, we need to collect $\binom{6+3}{3} - 1 = 83$ equations since there are at most 83 non-constant terms of degree upper bounded by 3 in 6 variables. In this case, we need about $83 \times \frac{17}{16} \approx 89$ collisions, i.e., $b \approx 7$. We perform 100 random tests with a number of 2^{14} random values IV in each test, since we need to generate about $2^b = 2^7$ collisions in 5 nonzero words in $\mathbb{F}_{17}$ and we have $2^{14+13} \times \frac{1}{16^5} = 2^7$.

The aim of our experiments is to compute the number of linearly independent equations for Gaussian elimination, i.e. the rank of the coefficient matrix. For the number of 2^{14} choices of IV, we are getting collisions in the range at around 98 to 155. In a test round when the number of collisions is less than 2^7, we take all collisions. When the number of collisions is $\geq 2^7$, we take exactly 2^7 collisions. In these 100 tests, the coefficient matrix is full-rank for 98 times. In the 68-th test, the number of collisions is 98, and the rank of the coefficient matrix is 81. In the 62-th test, the number of collisions is 108, and the rank of the coefficient matrix is 82.

5.5 Application to Rubato

The FHE-friendly cipher Rubato proposed at EUROCRYPT 2022 shares a similar structure with HERA. However, its design strategy for FHE-friendly ciphers further evolves, since it introduces Gaussian noise in the keystream. Specifically, the keystream of Rubato denoted by z can be simply described as follows:

$$z = \mathrm{Truncate}_t\Big(\mathsf{Rubato}_r(\mathrm{IV}, k)\Big) + (e_1, \ldots, e_t),$$

where the r-round permutation $\mathsf{Rubato}_r(\mathrm{IV}, k)$ is very similar to $\mathsf{HERA}_r(\mathrm{IV}, k)$, which uses the same way to randomize the round keys. In addition, it has a small number of rounds, i.e., $r \in \{2, 3, 5\}$. As can be observed from the above formula, different from HERA, only t output words of the permutation are truncated for generating the keystream, which somehow prevents the application of our strategy to peel off the last nonlinear layer. What is worse, to generate the final keystream, these t truncated output words will be added to t elements (i.e., noise) independently sampled from a Gaussian distribution. The attacker cannot know this Gaussian noise, and hence he has to guess it if he wants to know the t truncated output words of the permutation $\mathsf{Rubato}(\mathrm{IV}, k)$, which further increases the difficulty to apply our techniques to Rubato. Moreover, Rubato is defined over a prime field $\mathbb{F}_p$ with $\lceil \log_2 p \rceil \in \{25, 26\}$ and only parameters providing 80 and 128 bits of security are defined. In some versions, its state size is even larger than 16, e.g., 36 and 64, while the state size of HERA is always 16. Such features of Rubato also make finding collisions in round keys much more costly.

6 Conclusion and Open Problems

As another novel strategy to securely randomize the FHE-friendly ciphers, randomizing the key schedule has not yet been well-understood, though it has been used in two highly efficient CKKS-friendly ciphers HERA and Rubato for the RtF framework. By this research, we reveal a weakness of HERA caused by its special randomized key schedule. The main technique is to peel off the last nonlinear layer via multiple collisions, which then allows us to construct low-degree equations in the secret key, and results in improved algebraic attacks. As a consequence, we contradicted some claims made by the designers, and successfully reduced the security margins of some variants of HERA to only 1 round.

Although Rubato is similar to HERA, due to its usage of Gaussian noise, truncation mode and a larger state size, a direct application of the above attack strategy will fail for Rubato. However, we believe it is an interesting problem to devise a smarter attack on Rubato by taking our observations on HERA's randomized key schedule into account.

Acknowledgment. We thank the designers of HERA and Morten Øygarden for checking the results in this paper, and giving some useful feedback. In addition, we also thank the reviewers of ToSC 2024 Issue (1) for improving the quality of this paper. Fukang Liu is supported by Grant-in-Aid for Research Activity Start-up (Grant No. 22K21282).

References

[AAB+20] Abdelrahaman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of Symmetric-Key Primitives for Advanced Cryptographic Protocols. *IACR Trans. Symmetric Cryptol.*, 2020(3):1–45, 2020.

[AGP+19] Martin R. Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel Structures for MPC, and More. In *ESORICS (2)*, volume 11736 of *Lecture Notes in Computer Science*, pages 151–171. Springer, 2019.

[AGR+16] Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In *ASIACRYPT (1)*, volume 10031 of *Lecture Notes in Computer Science*, pages 191–219, 2016.

[AMT22] Tomer Ashur, Mohammad Mahzoun, and Dilara Toprakhisar. Chaghri - A FHE-friendly Block Cipher. In *CCS*, pages 139–150. ACM, 2022.

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454. Springer, 2015.

[BBC+23] Clémence Bouvier, Pierre Briaud, Pyrros Chaidos, Léo Perrin, Robin Salen, Vesselin Velichkov, and Danny Willems. New Design Techniques for Efficient Arithmetization-Oriented Hash Functions: ttAnemoi Permutations and ttJive Compression Mode. In *CRYPTO (3)*, volume 14083 of *Lecture Notes in Computer Science*, pages 507–539. Springer, 2023.

[BBDV20] Subhadeep Banik, Khashayar Barooti, F. Betül Durak, and Serge Vaudenay. Cryptanalysis of LowMC instances using single plaintext/ciphertext pair. *IACR Trans. Symmetric Cryptol.*, 2020(4):130–146, 2020.

[BBVY21] Subhadeep Banik, Khashayar Barooti, Serge Vaudenay, and Hailun Yan. New Attacks on LowMC Instances with a Single Plaintext/Ciphertext Pair. In *ASIACRYPT (1)*, volume 13090 of *Lecture Notes in Computer Science*, pages 303–331. Springer, 2021.

[BS90] Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.

[CCF+18] Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrède Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream Ciphers: A Practical Solution for Efficient Homomorphic-Ciphertext Compression. *J. Cryptol.*, 31(3):885–916, 2018.

[CDG$^+$17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives. In *CCS*, pages 1825–1842. ACM, 2017.

[CHK$^+$21] Jihoon Cho, Jincheol Ha, Seongkwang Kim, ByeongHak Lee, Joohee Lee, Jooyoung Lee, Dukjae Moon, and Hyojin Yoon. Transciphering Framework for Approximate Homomorphic Encryption. In *ASIACRYPT (3)*, volume 13092 of *Lecture Notes in Computer Science*, pages 640–669. Springer, 2021.

[CHMS22] Orel Cosseron, Clément Hoffmann, Pierrick Méaux, and François-Xavier Standaert. Towards Case-Optimized Hybrid Homomorphic Encryption - Featuring the Elisabeth Stream Cipher. In *ASIACRYPT (3)*, volume 13793 of *Lecture Notes in Computer Science*, pages 32–67. Springer, 2022.

[CIR22] Carlos Cid, John Petter Indrøy, and Håvard Raddum. FASTA - A Stream Cipher for Fast FHE Evaluation. In *CT-RSA*, volume 13161 of *Lecture Notes in Computer Science*, pages 451–483. Springer, 2022.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *ASIACRYPT (1)*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437. Springer, 2017.

[Dae95] Joan Daemen. Cipher and Hash Function Design Strategies based on Linear and Differential Cryptanalysis. *Ph.D. thesis*, 1995.

[DEG$^+$18] Christoph Dobraunig, Maria Eichlseder, Lorenzo Grassi, Virginie Lallemand, Gregor Leander, Eik List, Florian Mendel, and Christian Rechberger. Rasta: A Cipher with Low ANDdepth and Few ANDs per Bit. In *CRYPTO (1)*, volume 10991 of *Lecture Notes in Computer Science*, pages 662–692. Springer, 2018.

[DEM19] Claire Delaplace, Andre Esser, and Alexander May. Improved Low-Memory Subset Sum and LPN Algorithms via Multiple Collisions. In *IMACC*, volume 11929 of *Lecture Notes in Computer Science*, pages 178–199. Springer, 2019.

[DGGK21] Christoph Dobraunig, Lorenzo Grassi, Anna Guinet, and Daniël Kuijsters. Ciminion: Symmetric Encryption Based on Toffoli-Gates over Large Finite Fields. In *EUROCRYPT (2)*, volume 12697 of *Lecture Notes in Computer Science*, pages 3–34. Springer, 2021.

[DGH$^+$23] Christoph Dobraunig, Lorenzo Grassi, Lukas Helminger, Christian Rechberger, Markus Schofnegger, and Roman Walch. Pasta: A Case for Hybrid Homomorphic Encryption. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(3):30–73, 2023.

[Din20] Itai Dinur. Tight Time-Space Lower Bounds for Finding Multiple Collision Pairs and Their Applications. In *EUROCRYPT (1)*, volume 12105 of *Lecture Notes in Computer Science*, pages 405–434. Springer, 2020.

[Din21] Itai Dinur. Cryptanalytic Applications of the Polynomial Method for Solving Multivariate Equation Systems over GF(2). In *EUROCRYPT (1)*, volume 12696 of *Lecture Notes in Computer Science*, pages 374–403. Springer, 2021.

[DKR$^+$22] Christoph Dobraunig, Daniel Kales, Christian Rechberger, Markus Schofnegger, and Greg Zaverucha. Shorter Signatures Based on Tailor-Made Minimalist Symmetric-Key Crypto. In *CCS*, pages 843–857. ACM, 2022.

[DLR16] Sébastien Duval, Virginie Lallemand, and Yann Rotella. Cryptanalysis of the FLIP Family of Stream Ciphers. In *CRYPTO (1)*, volume 9814 of *Lecture Notes in Computer Science*, pages 457–475. Springer, 2016.

[DS09] Itai Dinur and Adi Shamir. Cube Attacks on Tweakable Black Box Polynomials. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 278–299. Springer, 2009.

[GAH$^+$23] Lorenzo Grassi, Irati Manterola Ayala, Martha Norberg Hovd, Morten Øygarden, Håvard Raddum, and Qingju Wang. Cryptanalysis of Symmetric Primitives over Rings and a Key Recovery Attack on Rubato. In *CRYPTO (3)*, volume 14083 of *Lecture Notes in Computer Science*, pages 305–339. Springer, 2023.

[GBJR23] Henri Gilbert, Rachelle Heim Boissier, Jérémy Jean, and Jean-René Reinhard. Cryptanalysis of Elisabeth-4. Cryptology ePrint Archive, Paper 2023/1436, 2023. https://eprint.iacr.org/2023/1436.

[GHR$^+$23] Lorenzo Grassi, Yonglin Hao, Christian Rechberger, Markus Schofnegger, Roman Walch, and Qingju Wang. Horst Meets Fluid-SPN: Griffin for Zero-Knowledge Applications. In *CRYPTO (3)*, volume 14083 of *Lecture Notes in Computer Science*, pages 573–606. Springer, 2023.

[GKL$^+$22] Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, Markus Schofnegger, and Roman Walch. Reinforced Concrete: A Fast Hash Function for Verifiable Computation. In *CCS*, pages 1323–1335. ACM, 2022.

[GKR$^+$21] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon: A New Hash Function for Zero-Knowledge Proof Systems. In *USENIX Security Symposium*, pages 519–535. USENIX Association, 2021.

[GØSW23] Lorenzo Grassi, Morten Øygarden, Markus Schofnegger, and Roman Walch. From Farfalle to Megafono via Ciminion: The PRF Hydra for MPC applications. In *EUROCRYPT (4)*, volume 14007 of *Lecture Notes in Computer Science*, pages 255–286. Springer, 2023.

[HKC$^+$20] Jincheol Ha, Seongkwang Kim, Wonseok Choi, Jooyoung Lee, Dukjae Moon, Hyojin Yoon, and Jihoon Cho. Masta: An HE-Friendly Cipher Using Modular Arithmetic. *IEEE Access*, 8:194741–194751, 2020.

[HKL$^+$22] Jincheol Ha, Seongkwang Kim, ByeongHak Lee, Jooyoung Lee, and Mincheol Son. Rubato: Noisy Ciphers for Approximate Homomorphic Encryption. In *EUROCRYPT (1)*, volume 13275 of *Lecture Notes in Computer Science*, pages 581–610. Springer, 2022.

[HL20] Phil Hebborn and Gregor Leander. Dasta - Alternative Linear Layer for Rasta. *IACR Trans. Symmetric Cryptol.*, 2020(3):46–86, 2020.

[KHS$^+$22] Seongkwang Kim, Jincheol Ha, Mincheol Son, Byeonghak Lee, Dukjae Moon, Joohee Lee, Sangyub Lee, Jihoon Kwon, Jihoon Cho, Hyojin Yoon, and Jooyoung Lee. AIM: Symmetric Primitive for Shorter Signatures with Stronger Security (Full Version). Cryptology ePrint Archive, Paper 2022/1387, 2022. https://eprint.iacr.org/2022/1387.

[KW02] Lars R. Knudsen and David A. Wagner. Integral Cryptanalysis. In *FSE*, volume 2365 of *Lecture Notes in Computer Science*, pages 112–127. Springer, 2002.

[LAW+23] Fukang Liu, Ravi Anand, Libo Wang, Willi Meier, and Takanori Isobe. Coefficient Grouping: Breaking Chaghri and More. In *EUROCRYPT (4)*, volume 14007 of *Lecture Notes in Computer Science*, pages 287–317. Springer, 2023.

[LIM21] Fukang Liu, Takanori Isobe, and Willi Meier. Cryptanalysis of Full LowMC and LowMC-M with Algebraic Techniques. In *CRYPTO (3)*, volume 12827 of *Lecture Notes in Computer Science*, pages 368–401. Springer, 2021.

[LMSI22] Fukang Liu, Willi Meier, Santanu Sarkar, and Takanori Isobe. New Low-Memory Algebraic Attacks on LowMC in the Picnic Setting. *IACR Trans. Symmetric Cryptol.*, 2022(3):102–122, 2022.

[LSMI21] Fukang Liu, Santanu Sarkar, Willi Meier, and Takanori Isobe. Algebraic Attacks on Rasta and Dasta Using Low-Degree Equations. In *ASIACRYPT (1)*, volume 13090 of *Lecture Notes in Computer Science*, pages 214–240. Springer, 2021.

[LSMI22] Fukang Liu, Santanu Sarkar, Willi Meier, and Takanori Isobe. The Inverse of χ and Its Applications to Rasta-Like Ciphers. *J. Cryptol.*, 35(4):28, 2022.

[LSW+22] Fukang Liu, Santanu Sarkar, Gaoli Wang, Willi Meier, and Takanori Isobe. Algebraic Meet-in-the-Middle Attack on LowMC. In *ASIACRYPT (1)*, volume 13791 of *Lecture Notes in Computer Science*, pages 225–255. Springer, 2022.

[Mat93] Mitsuru Matsui. Linear Cryptanalysis Method for DES Cipher. In *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

[MJSC16] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards Stream Ciphers for Efficient FHE with Low-Noise Ciphertexts. In *EUROCRYPT (1)*, volume 9665 of *Lecture Notes in Computer Science*, pages 311–343. Springer, 2016.

[NLV11] Michael Naehrig, Kristin E. Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *CCSW*, pages 113–124. ACM, 2011.

[RST18] Christian Rechberger, Hadi Soleimany, and Tyge Tiessen. Cryptanalysis of Low-Data Instances of Full LowMCv2. *IACR Trans. Symmetric Cryptol.*, 2018(3):163–181, 2018.

[Str69] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.

[vOW99] Paul C. van Oorschot and Michael J. Wiener. Parallel Collision Search with Cryptanalytic Applications. *J. Cryptol.*, 12(1):1–28, 1999.