YuS: A FHE-friendly Stream Cipher Based on New Quadratic Permutations

Yongqiang Li, Fangzhen Wang, Xingwei Ren, Fen Liu, Xichao Hu, Lin Jiao, and Ya Han

***Abstract*—Permutations with low multiplication depth over prime fields are highly valuable in the design of symmetric ciphers that are compatible with fully homomorphic encryption (FHE). Quadratic permutations, which have the lowest depth, have been widely used in prior designs. In this paper, we propose a construction method that can give new quadratic permutations over $\mathbb{F}_p^m$, and cryptographic properties such as differential uniformity and Walsh spectrum of these permutations are also characterized. We give sufficient conditions for permutations over $\mathbb{F}_p^n$ to attain a differential uniformity of p^{n-1} for $n \geq 3$. Furthermore, it is proven that for these permutations, the maximal 2-norm of Walsh coefficients remains bounded by p^{n-1}, provided either the last $n-1$ entries of the input mask or the last $n-1$ entries of the output mask form a nonzero vector. As an application, we design a new FHE-friendly stream cipher named YuS based on a new quadratic permutation over $\mathbb{F}_p^3$ and a fixed linear mapping. According to our implementation, YuS achieves faster evaluation times and higher throughput compared to Masta, PASTA, PASTA$_{v2}$ and HERA in almost all instances for both BGV and BFV schemes at 80-bit and 128-bit security levels.**

***Index Terms*—Quadratic permutation, Stream cipher, Fully homomorphic encryption, BGV/BFV**

I. INTRODUCTION

FULLY Homomorphic Encryption (FHE) is a cryptographic primitive that enables direct computation over encrypted data, producing encrypted ciphertexts equivalent to those obtained from computations on plaintext. Several prominent FHE schemes have been proposed through breakthrough research in recent years. For instance, the FHEW scheme [1] and the TFHE scheme [2] are designed for Boolean circuits, while the BGV scheme [3] and the BFV scheme [4], [5] are tailored for computations over large finite fields. Additionally, the CKKS scheme [6] supports approximate arithmetic over real and complex numbers.

Manuscript received February 25, 2025; revised July 6, 2025; Accepted September 20, 2025. This work is supported by National Key R&D Program of China under Grant 2022YFF0604702, the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDB0690200, the National Natural Science Foundation under Grant 12371525, and also supported by the Guangdong-Foshan Joint Fund Project under Grant 2024A1515110038. (Corresponding author: Fen Liu, Lin Jiao, Ya Han.)

Yongqiang Li, Fangzhen Wang, Xingwei Ren, Ya Han are with State Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, and are also with School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China. Email: yongq.lee@gmail.com; {wangfangzhen, renxingwei, hanya}@iie.ac.cn

Fen Liu is with School of Computer Science and Artificial Intelligence, Foshan University, Guangdong, China. Email: liufenxd@gmail.com

Xichao Hu, Lin Jiao are with State Key Laboratory of Cryptology, Beijing, China. Email: xchao_h@163.com; jiaolin_jl@126.com

However, a major challenge with FHE schemes is the inherently large ciphertext size, which often leads to substantial communication overhead, particularly in scenarios involving bulk data transfer. To overcome this challenge, the Hybrid Homomorphic Encryption (HHE) framework was proposed in [7], also referred to as a transciphering framework [8]. In this framework, the client first encrypts private data using a symmetric cipher, resulting in ciphertexts whose size matches that of the plaintexts, thereby significantly reducing communication overhead between the client and the server. The server then homomorphically evaluates a symmetric decryption circuit to transform the symmetric ciphertexts into homomorphic ciphertexts, utilizing the homomorphically encrypted key received from the client.

In the context of transciphering framework, there are primarily two lines of research for developing or adapting symmetric ciphers. The first focuses on optimizing the FHE implementation of widely used symmetric ciphers, particularly AES, as demonstrated in works such as [9]–[11]. The second direction seeks to design new FHE-friendly symmetric ciphers specifically tailored for the transciphering framework. Over the past decade, numerous FHE-friendly symmetric ciphers have been proposed. Some ciphers are designed based on operations over $\mathbb{F}_2$, such as LowMC [12], Kreyvium [13], FLIP [14], Rasta [15], Dasta [16], Fasta [17], etc. Others are based on operations over large fields $\mathbb{F}_q$ (q being either a prime p or a power of 2), since many protocols naturally support arithmetic in a larger field $\mathbb{F}_q$ and converting operations over $\mathbb{F}_q$ into bit operations is expensive. Examples include MiMC [18], GMiMC [19], Jarvis [20], CHAGHRI [21], Masta [22], PASTA [23], YuX [24], and PASTA$_{v2}$ [25]. There are also ciphers combining the CKKS and BFV homomorphic encryption schemes to encrypt real numbers, such as HERA [8] and the noisy cipher Rubato [26]. Notably, stream ciphers have attracted particular interest within the transciphering framework due to their capability for offline keystream pregeneration, which can dramatically enhance online decryption efficiency [8]. This offline computation feature enables the flexible distribution of workloads between the client and the server, thereby reducing latency.

Symmetric ciphers typically consist of both linear and nonlinear layers, and in traditional encryption schemes, it is generally unnecessary to distinguish their design principles in terms of implementation cost. However, when integrated into transciphering frameworks, the substantial gap in cost between homomorphic addition and multiplication shifts the focus of designing FHE-friendly ciphers toward minimizing multiplicative depth and complexity. For the linear layer, constructions

such as FLIP [14], Rasta [15], and PASTA [23] employ random matrices in each round to maintain resistance against statistical attacks. However, experiments in [8] (Table 2 of HERA [8]) demonstrate that encoding such randomized linear layers for homomorphic batching can substantially increase both runtime and noise budget consumption compared to using fixed layers. Consequently, several recent designs, such as Masta [22] and PASTA$_{\text{v2}}$ [25], aim to minimize the randomness in the linear layer, while others like the state-of-the-art (SOTA) FHE-friendly cipher HERA [8] directly adopt a fully fixed linear matrix to avoid the overhead of freshly generated matrices.

Meanwhile, the nonlinear layer contributes significantly to the multiplicative depth in such ciphers, and an increase in multiplication depth leads to rapid growth of ciphertext noise. Permutations over finite fields with low multiplication depth are of critical importance in the design of FHE-friendly ciphers. Therefore, quadratic permutations over $\mathbb{F}_p^n$ are particularly significant due to their lowest multiplication depth. One of the most widely used quadratic S-boxes over $\mathbb{F}_p^n$ in previous designs is the Feistel S-box, which is a one-round Feistel construction defined as $S_{Feistel}(x_0, \ldots, x_{n-1}) = (x_0, x_0^2 + x_1, \ldots, x_{n-2}^2 + x_{n-1})$. It is used in the design of PASTA [23], PASTA$_{\text{v2}}$ [25] and Rubato [26].

While algebraic attacks represent the primary threat to FHE-friendly ciphers, statistical attacks also require consideration, particularly in designs employing fixed linear layers across all rounds. Unlike ciphers with randomized linear layers, those with fixed structures may exhibit increased vulnerability to differences and linear trails propagation across multiple rounds. The evaluation of statistical attack resistance typically involves analyzing both the cryptographic properties of the S-boxes and the branch number of the linear layer, ensuring sufficient active S-boxes to resist such attacks. For example, HERA uses this approach by combining the linearity and differential uniformity of its S-box ($S_{cube}(x) = x^3$) with a delicately designed linear layer with high branch number to achieve resistance against both linear and differential attacks. However, it should be noted that HERA's S-box implementation in odd prime fields requires a multiplication depth of 2.

A cipher incorporating fixed linear mappings and quadratic permutations may have better performance compared to previous designs. The design of such ciphers necessitates the construction of quadratic permutations over $\mathbb{F}_p^n$ with well-characterized cryptographic properties, including differential uniformity and Walsh spectrum. There are some works investigating the construction of quadratic permutations over $\mathbb{F}_p^n$. A special type of quadratic permutation, constructed via the shift-invariant technique, is explored in [27]. Specifically, these permutations are of the form $\mathcal{S}_f(x_0, \ldots, x_{n-1}) = (f(\bar{x}), f(\bar{x} \lll 1), \ldots, f(\bar{x} \lll n-1))$, where $f : \mathbb{F}_p^n \mapsto \mathbb{F}_p$ is a quadratic function (also referred to as a local map), and $x \lll i$ denotes a left rotation by i words. This construction is generalized by introducing two local maps in [28]. It is proven that $\mathcal{S}_{f_0,f_1}$ is invertible over $\mathbb{F}_p^n$ if and only if it is a Type-II Feistel for $n \geq 4$. However, the cryptographic properties of these new quadratic permutations are not characterized.

Our Contributions. In this paper, we investigate the problem of constructing new quadratic permutations over $\mathbb{F}_p^n$ and characterize their differential uniformity and Walsh spectrum. First, we prove that quadratic permutations over $\mathbb{F}_p^2$ always have a differential uniformity of p^2. Then we give a method to construct quadratic permutations over $\mathbb{F}_p^n$. We give the conditions under which the permutations over $\mathbb{F}_p^n$ achieve a differential uniformity of p^{n-1} for $n \geq 3$. It is also proven that the highest 2-norm of the Walsh coefficients of these permutations is less than or equal to p^{n-1} when either the last $n-1$ entries of the input mask or the last $n-1$ entries of the output mask form a nonzero vector.

As an application of our method, we present a concrete example of a quadratic permutation over $\mathbb{F}_p^3$ with a differential uniformity of p^2. Utilizing this permutation as the S-box and selecting a fixed matrix as the linear mapping, we design YuS, a stream cipher optimized for efficient decryption circuit evaluation in BGV/BFV schemes. The cipher takes a key from $\mathbb{F}_p^{36}$ and outputs a keystream over $\mathbb{F}_p^{24}$. YuS is based on the SPN structure, with each round consisting of 12 S-boxes over $\mathbb{F}_p^3$ followed by a binary matrix of order 36 and a round-key addition operation. In the last round, a linear mapping and a truncation function are applied before generating the keystream.

We provide a security analysis of YuS against linear attack and algebraic attacks. Based on the Walsh spectrum, the differential uniformity of the S-box and the branch number of the layer mapping, the 2-round instance of YuS demonstrates resistance to both linear and differential attacks for 80-bit and 128-bit security levels. The 4-round and 5-round instances of YuS can resist trivial linearization attacks, achieving 80-bit and 128-bit security, respectively. Moreover, we prove that the Gröbner basis attack on YuS is no better than the trivial linearization attack. Considering the XL attack under a strong assumption, only 2 rounds and 3 rounds of YuS are sufficient to achieve 80-bit and 128-bit security, respectively. Similarly, other algebraic attacks, including the GCD attack and a recent attack exploiting multiple collisions, are also ineffective against the current security parameters.

Finally, we present homomorphic implementations of YuS leveraging both BGV and BFV schemes, realized within the open-source homomorphic encryption libraries HElib and SEAL. Our implementation adopts a row-wise packing strategy, enabling parallel processing of multiple blocks through multiple ciphertexts. The evaluations demonstrate that YuS outperforms existing prime field stream ciphers in terms of evaluation times and throughput in almost all cases across both BGV and BFV schemes, see Table I for details. For example, the total evaluation time and throughput of YuS in BGV implementation at 80-bit security over $\mathbb{F}_{65537}$ is 1.752× faster and 1.314× higher than HERA.

Organization. In Sect. II, we give some preliminaries of FHE and cryptographic properties of S-boxes. In Sect. III, we propose a method for constructing new quadratic permutations over $\mathbb{F}_p^n$ and characterize the differential uniformity and Walsh transform. A concrete example over $\mathbb{F}_p^3$ is also given. In Sect.IV and Sect. V, we give the description of YuS and its design rational respectively. In Sect.VI, we give the security analysis of YuS against linear attack and algebraic attacks. In Sect.VII, we present the implementation and comparison. A

TABLE I
THE ADVANTAGES OF YUS

Cipher	BGV_{80,p_1}	BGV_{128,p_1}	BGV_{80,p_2}	BGV_{128,p_2}	BFV_{80,p_1}	BFV_{128,p_1}	BFV_{80,p_2}	BFV_{128,p_2}
Total Evaluation Time								
Masta	$2.454 \sim 2.987$	4.676	$4.588 \sim 9.109$	11.623	$1.759 \sim 5.807$	$5.978 \sim 6.938$	$1.503 \sim 23.559$	$4.854 \sim 28.175$
HERA	1.752	1.028	2.513	1.076	$0.926 \sim 1.105$	$0.943 \sim 1.094$	$0.935 \sim 5.333$	$0.966 \sim 5.606$
PASTA	–	5.217	–	5.322	–	$8.459 \sim 9.817$	–	$7.275 \sim 42.229$
PASTA_{v2}	–	2.973	–	3.257	–	$3.708 \sim 4.303$	–	$3.351 \sim 19.453$
Throughput								
Masta	$1.840 \sim 2.240$	3.507	$3.416 \sim 3.441$	4.358	$2.638 \sim 4.355$	$4.483 \sim 5.203$	$2.255 \sim 8.835$	$3.641 \sim 10.566$
HERA	1.314	1.542	1.885	1.614	$1.389 \sim 1.657$	$1.414 \sim 1.641$	$1.402 \sim 3.999$	$1.449 \sim 4.205$
PASTA	–	3.913	–	3.991	–	$6.344 \sim 7.363$	–	$5.457 \sim 15.837$
PASTA_{v2}	–	2.230	–	2.443	–	$2.781 \sim 3.227$	–	$2.514 \sim 7.295$

BGV_{k,p_i} means the evaluation of ciphers in BGV under p_i at k-bit security, where $k = 80, 128$, and $p_1 = 65537, p_2 = 4298506241$
The ratios of the total evaluation time and throughput are calculated by $\frac{t_{ciphers}}{t_{\text{YuS}}}$ and $\frac{TP_{\text{YuS}}}{TP_{ciphers}}$ respectively

short conclusion is given in Sect.VIII.

II. PRELIMINARY

A. Fully Homomorphic Encryption

Fully homomorphic encryption (FHE) schemes allow arbitrary functions over encrypted data without the decryption key. Nowadays, FHE schemes are based on the hardness of the LWE problem and are optimized by using Ring-LWE. In this paper, we consider the BGV and BFV schemes, the corresponding libraries are HElib [29] and SEAL [30].

BFV. The Brakerski-Fan-Vercauteren (BFV) scheme, (also known as the Fan-Vercauteren (FV) scheme) is considered as one of the second generation of FHE schemes that is constructed based on the Ring-Learning with Errors (RLWE) problem [31]. Let $\mathcal{R}_t = \mathbb{Z}_t/(X^n+1)$ and $\mathcal{R}_q = \mathbb{Z}_q/(X^n+1)$, we define plaintext and ciphertext spaces over distinct polynomial quotient rings denoted by $\mathcal{P} = \mathcal{R}_t$ and $\mathcal{C} = \mathcal{R}_q \times \mathcal{R}_q$, where $t(q) \in \mathbb{Z}$ is plaintext(ciphertext) coefficient and $n \in \mathbb{Z}$ is the ring dimension. Let $\mathcal{U}(\mathcal{R}_q)$ be the uniform distribution over $\mathcal{R}_q$, $\mathcal{HWT}(w)$ be the uniform random distribution used to sample polynomials of degree n with integer coefficients in $\{-1, 0, 1\}$ and Hamming weight w, and $\mathcal{X}$ be the discrete Gaussian distribution with proper parameters. Left arrow "$\leftarrow$" is denoted as sampling a random element from the corresponding distribution. In a nutshell, the BFV scheme includes three steps as follows:

1) **Key Generation**: Let $\boldsymbol{sk} \leftarrow \mathcal{HWT}(w)$ be the secret key, $a \leftarrow \mathcal{U}(\mathcal{R}_q)$ be a random polynomial in $\mathcal{R}_q$ and $e \leftarrow \mathcal{X}$ be a random error polynomial. The public key $\boldsymbol{pk} = (\boldsymbol{pk}_1, \boldsymbol{pk}_2) = ([-1(a \cdot \boldsymbol{sk} + e)]_q, a) \in \mathcal{R}_q \times \mathcal{R}_q$. The notation $[\cdot]_q$ means that polynomial arithmetic should be done modulo q.
2) **Encryption**: For a plaintext message $\boldsymbol{m} \in \mathcal{P}$, we generate three small random polynomials u, e_1, and e_2, where u is drawn from a set of polynomials of degree n with coefficients in $\{-1, 0, 1\}$, and e_1, e_2 are sampled from $\mathcal{X}$. The ciphertext $\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2) = ([\boldsymbol{pk}_1 \cdot u + e_1 + \Delta \boldsymbol{m}]_q, [\boldsymbol{pk}_2 \cdot u + e_2]_q) \in \mathcal{C}$, where $\Delta = \lfloor \frac{q}{t} \rfloor$ is used to scale the message.
3) **Decryption**: $\boldsymbol{m} = [\lfloor \frac{t[\boldsymbol{c}_1 + \boldsymbol{c}_2 \cdot \boldsymbol{sk}]_q}{q} \rceil]_t$.

BGV. The Brakerski-Gentry-Vaikuntanathan (BGV) scheme is another FHE scheme that belongs to the second generation of FHE schemes. Its security also stems from the hardness of RLWE problem [31]. Let $\mathcal{R}_t = \mathbb{Z}_t/(X^n+1)$ and $\mathcal{R}_{q_\ell} = \mathbb{Z}_{q_\ell}/(X^n+1)$, where $q_\ell \in \mathbb{Z}$ is the ciphertext modulus at level $\ell, 0 \le \ell \le L$. We define plaintext and ciphertext spaces over distinct polynomial quotient rings denoted by $\mathcal{P} = \mathcal{R}_t$ and $\mathcal{C} = \mathcal{R}_{q_\ell} \times \mathcal{R}_{q_\ell}$. The BGV scheme is similar to BFV scheme as follows:

1) **Key Generation**: Let $\boldsymbol{sk} \leftarrow \mathcal{HWT}(w)$ be the secret key, $a \leftarrow \mathcal{U}(\mathcal{R}_{q_\ell})$ be a random polynomial in $\mathcal{R}_{q_\ell}$ and $e \leftarrow \mathcal{X}$ be a random error polynomial. The public key $\boldsymbol{pk} = (\boldsymbol{pk}_1, \boldsymbol{pk}_2) = ([-1(a \cdot \boldsymbol{sk} + te)]_{q_\ell}, a) \in \mathcal{R}_{q_\ell} \times \mathcal{R}_{q_\ell}$.
2) **Encryption**: For a plaintext message $\boldsymbol{m} \in \mathcal{P}$, we generate three small random polynomials u, e_1, and e_2, where u is drawn from a random polynomial of degree n with coefficients in $\{-1, 0, 1\}$, and e_1, e_2 from $\mathcal{X}$. The ciphertext $\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2) = ([\boldsymbol{pk}_1 \cdot u + te_1 + \boldsymbol{m}]_{q_\ell}, [\boldsymbol{pk}_2 \cdot u + te_2]_{q_\ell}) \in \mathcal{C}$.
3) **Decryption**: $\boldsymbol{m} = [[\boldsymbol{c}_1 + \boldsymbol{c}_2 \cdot \boldsymbol{sk}]_{q_\ell}]_t$.

Patching Encode. For a positive integer M which is a power of two, and $N = \frac{M}{2}$, the M-th cyclotomic polynomial is $\Phi_M(X) = X^N + 1$. For a prime number $p \equiv 1 \mod M$, $\mathcal{R}_p = \mathbb{Z}_p[X]/(\Phi_M(X))$ is the plaintext space. It is well known that $\Phi_M(X) = \prod_{i=0}^{N-1}(X - \zeta^{2i+1})$, where $\zeta \in \mathbb{Z}_p$ is a M-th primitive root of unity mod p, i.e., $order_p(\zeta) = M$. Applying the Chinese Remainder Theorem (CRT), we deduce the isomorphism as follows:

$$\mathcal{R}_p \cong^{CRT} \prod_{i=0}^{N-1} \mathbb{Z}_p[X]/(X - \zeta^{2i+1}) = \prod_{i=0}^{N-1} \mathbb{Z}_p[\zeta^{2i+1}].$$

So we can encode N integers ($\in \mathbb{Z}_p$) as a polynomial ($\in \mathcal{R}_p$) and which can be accelerated by fast number-theoretic transform (FNTT). Moreover, the isomorphism allows us to perform component-wise addition and multiplication on vectors in $\mathbb{Z}_p^N$ by performing corresponding operations on elements of $\mathcal{R}_p$, which are also called simple SIMD operations.

B. Cryptography Properties of an S-box

Let $\mathbb{F}_p$ be the finite field with p elements, where p is a prime number. Let $\mathbb{F}_p^n$ be the n dimension linear space over $\mathbb{F}_p$, and $\bar{c}$ denotes a vector $(c_0, \ldots, c_{n-1})$ in $\mathbb{F}_p^n$. For a vectorial

function $F(\bar{x}) = (f_0(\bar{x}), \ldots, f_{n-1}(\bar{x}))$ over $\mathbb{F}_p^n$, its differential uniformity $\Delta(F)$ is defined as

$$\Delta(F) = \max_{\bar{a},\bar{b}\in\mathbb{F}_p^n,\bar{a}\neq(0,\ldots,0)} |\{\bar{x} \in \mathbb{F}_p^n \mid F(\bar{x}) - F(\bar{x}-\bar{a}) = \bar{b}\}|,$$

where $\bar{x} - \bar{a} = (x_0 - a_0, \ldots, x_{n-1} - a_{n-1})$, and $F(x)$ is called differential δ-uniform when $\Delta(F) = \delta$ [32]. For $f(\bar{x}) : \mathbb{F}_p^n \mapsto \mathbb{F}_p$, the Walsh transform of $f(\bar{x})$ is defined as $\lambda_f(\bar{u}) = \sum_{\bar{x}\in\mathbb{F}_p^n} \chi(f(x) - \bar{u}\cdot\bar{x})$, where $\bar{u} \in \mathbb{F}_p^n$, $\chi(x) = e^{\frac{2\pi i}{p}(x)}$, $\bar{u}\cdot\bar{x} = \sum_{i=0}^{n-1} u_i x_i$ is the usual scalar dot production. For a vectorial function $F(\bar{x}) = (f_0(\bar{x}), \ldots, f_{n-1}(\bar{x}))$ over $\mathbb{F}_p^n$, the linearity of $F(\bar{x})$ is the highest 2-norm of the Walsh coefficients of all its components, i.e., $\mathcal{L}(F) = \max_{\bar{0}\neq\bar{v}\in\mathbb{F}_p^n,\bar{u}\in\mathbb{F}_p^n} |\lambda_F(\bar{u},\bar{v})|$, where $\lambda_F(\bar{u},\bar{v}) = \lambda_{\bar{v}\cdot F}(\bar{u}) = \sum_{\bar{x}\in\mathbb{F}_p^n} \chi(v\cdot F(x) - u\cdot x)$.

A function $f(\bar{x}) : \mathbb{F}_p^n \mapsto \mathbb{F}_p$ is called bent if $|\lambda_f(\bar{u})| = \sqrt{p^n}$ for all $\bar{u} \in \mathbb{F}_p^n$ [33]. A function $f : \mathbb{F}_p^n \mapsto \mathbb{F}_p^m$ is called balanced if for every $\bar{b} \in \mathbb{F}_p^m$, it holds $|\{\bar{x} \in \mathbb{F}_p^n : f(\bar{x}) = \bar{b}\}| = p^{n-m}$. When $n = m$, a balanced function $f : \mathbb{F}_p^n \mapsto \mathbb{F}_p^n$ is also called a permutation over $\mathbb{F}_p^n$. A function $f : \mathbb{F}_p^n \mapsto \mathbb{F}_p^m$ is perfect nonlinear if for every nonzero $\bar{d} \in \mathbb{F}_p^n$, the difference $f(\bar{x}) - f(\bar{x}-\bar{d})$ is balanced [34].

Note that a bent function is not balanced and vice versa, since $|\lambda_f(\bar{0})| = 0$ for a balanced function f. The following result relates perfect nonlinear functions and bent functions.

Lemma 1: [34] A perfect nonlinear function from $\mathbb{F}_q^n$ to $\mathbb{F}$ is bent. The converse is true if q is a prime.

Lemma 2: Let $F(\bar{x}) = (f_0(\bar{x}), \ldots, f_{n-1}(\bar{x}))$ be a permutation over $\mathbb{F}_p^n$. Then for every nonzero $(c_0, \ldots, c_{n-1}) \in \mathbb{Z}_p^n$, $c\cdot F = \sum_{i=0}^{n-1} c_i f_i$ is a balanced function from $\mathbb{F}_p^n$ to $\mathbb{F}_p$.

III. Quadratic Permutations over $\mathbb{F}_p^n$

In this section, we characterize quadratic permutations over $\mathbb{F}_p^n$ and give some constructions with nontrivial differential uniformity. Throughout this section, p is a prime number.

Let $f(\bar{x}) = \sum_{0\le i\le j\le n-1} a_{i,j}x_i x_j$ be a homogeneous polynomial from $\mathbb{F}_p^n$ to $\mathbb{F}_p$ with degree 2. Let M be a matrix that relates the coefficients of f as

$$M[i,j] = \begin{cases} a_{i,j}, & 1 \le i \le j \le n \\ 0, & n \ge i > j \ge 1. \end{cases}$$

Then $f(\bar{x}) = (x_0, \ldots, x_{n-1})\cdot M \cdot (x_0, \ldots, x_{n-1})^T$, and M is called the coefficient matrix of f.

Lemma 3: Let $q(\bar{x}) = f(\bar{x}) + l(\bar{x}) + c \in \mathbb{F}_p[x_0, \ldots, x_{n-1}]$ be a balanced quadratic function, where $f(\bar{x})$ is a homogeneous polynomial with degree 2 and $a(\bar{x}) = l(\bar{x}) + c$ is a polynomial with degree 1. Suppose M is the coefficient matrix of f. Then the matrix $M + M^T$ is not of full rank.

Proof: Let M be the coefficient matrix of $f(\bar{x})$. Note that for $\bar{a} \in \mathbb{F}_p^n$, it holds

$$\begin{aligned} & f(\bar{x}) - f(\bar{x}-\bar{a}) \\ = \;& \bar{x}M\bar{x}^T - (\bar{x}-\bar{a})M(\bar{x}-\bar{a})^T \\ = \;& \bar{x}M\bar{x}^T - (\bar{x}M\bar{x}^T - \bar{a}M\bar{x}^T - \bar{x}M\bar{a}^T + \bar{a}M\bar{a}^T) \\ = \;& \bar{a}(M+M^T)\bar{x}^T - \bar{a}M\bar{a}^T, \end{aligned} \tag{1}$$

since $(\bar{x}M\bar{a}^T)^T = \bar{a}M^T\bar{x}^T$.

Assume $M + M^T$ is of full rank. Then for any nonzero $\bar{a} \in \mathbb{F}_p^n$, $\bar{a}(M + M^T)$ is a nonzero vector in $\mathbb{F}_p^n$. Therefore, the kernel of the linear mapping $\bar{a}(M + M^T)x^T$ is a linear space over $\mathbb{F}_p$ of dimension $n-1$. Then according to Equality (1),the difference equation

$$\begin{aligned} b &= q(\bar{x}) - q(\bar{x}-\bar{a}) \\ &= f(\bar{x}) - f(\bar{x}-\bar{a}) + l(\bar{a}) \\ &= \bar{a}(M+M^T)\bar{x}^T + l(\bar{a}). \end{aligned}$$

always has p^{n-1} solutions, for $\bar{0} \neq \bar{a} \in \mathbb{F}_p^n$ and $b \in \mathbb{F}_p$. This means $q(\bar{x})$ is a perfect nonlinear function and hence a bent function from $\mathbb{F}_p^n$ to $\mathbb{F}_p$ according to Lemma 1. This contradicts to that $q(\bar{x})$ is balanced. ■

A. On the Differential Uniformity of Quadratic Permutations over $\mathbb{F}_p^2$

In this subsection, we prove that the differential uniformity of quadratic permutations over $\mathbb{F}_p^2$ is always equal to p^2.

Theorem 1: Let $Q(x_0, x_1)$ be a quadratic permutation over $\mathbb{F}_p^2$, where p is an odd prime number. Then the differential uniformity of $Q(x_0, x_1)$ is p^2.

Proof: Let

$$\begin{aligned} Q(x_0,x_1) &= (f_0(x_0,x_1), f_1(x_0,x_1)) \\ &= (a_0x_0^2 + a_1x_0x_1 + a_2x_1^2 + a_3x_0 + a_4x_1, \\ &\quad b_0x_0^2 + b_1x_0x_1 + b_2x_1^2 + b_3x_0 + b_4x_1), \end{aligned}$$

where $a_i, b_i \in \mathbb{F}_p, 0 \le i \le 4$. We need to prove that there exists $(0,0) \neq (d_0,d_1) \in \mathbb{F}_p^2$, such that $Q(x_0,x_1) - Q(x_0 - d_0, x_1 - d_1)$ is a constant in $\mathbb{F}_p^2$.

Note that

$$\begin{aligned} & f_0(x_0,x_1) - f_0(x_0-d_0, x_1-d_1) \\ = \;& a_0(2d_0x_0 - d_0^2) + a_1(d_0x_1 + d_1x_0 - d_0d_1) \\ & + a_2(2d_1x_1 - d_1^2) + a_3d_0 + a_4d_1 \\ = \;& (2a_0d_0 + a_1d_1)x_0 + (a_1d_0 + 2a_2d_1)x_1 - f_0(-d_0,-d_1). \end{aligned}$$

Similarly, we also have

$$\begin{aligned} & f_1(x_0,x_1) - f_1(x_0-d_0, x_1-d_1) \\ = \;& (2b_0d_0 + b_1d_1)x_0 + (b_1d_0 + 2b_2d_1)x_1 - f_1(-d_0,-d_1). \end{aligned}$$

Therefore,

$$\begin{aligned} & Q(x_0,x_1) - Q(x_0-d_0, x_1-d_1) \\ = \;& (f_0(x_0,x_1) - f_1(x_0-d_0,x_1-d_1), \\ & f_1(x_0,x_1) - f_2(x_0-d_0,x_1-d_1)) \\ = \;& ((2a_0d_0 + a_1d_1)x_0 + (a_1d_0 + 2a_2d_1)x_1 - k_0, \\ & (2b_0d_0 + b_1d_1)x_0 + (b_1d_0 + 2b_2d_1)x_1 - k_1), \end{aligned} \tag{2}$$

where $k_0 = f_0(-d_0,-d_1), k_1 = f_1(-d_0,-d_1)$. Then we claim that the matrix

$$M_{\bar{d}} = \begin{pmatrix} 2a_0d_0 + a_1d_1, & a_1d_0 + 2a_2d_1 \\ 2b_0d_0 + b_1d_1, & b_1d_0 + 2b_2d_1 \end{pmatrix}$$

is not of full rank for all $(0,0) \neq (d_0,d_1) \in \mathbb{F}_p^2$. Otherwise, if there exist $(0,0) \neq (d_0,d_1) \in \mathbb{F}_p^2$ such that $\mathrm{rank}(M_{\bar{d}}) = 2$, then $(x_0,x_1) = M_{\bar{d}}^{-1}\cdot(f_0(-d_0,-d_1), f_1(-d_0,-d_1))^T$ is the solution of $Q(x_0,x_1) - Q(x_0-d_0,x_1-d_1) = (0,0)$ in $\mathbb{F}_p^2$.

This contradicts the condition that $Q(x_0, x_1)$ is a permutation over $\mathbb{F}_p^2$.

Then the determinant of $M_{\bar{d}} = 0$, which means

$$\begin{aligned} 0 &= (2a_0d_0 + a_1d_1)(b_1d_0 + 2b_2d_1) \\ &\quad -(2b_0d_0 + b_1d_1)(a_1d_0 + 2a_2d_1) \\ &= 2a_0b_1d_0^2 + (4a_0b_2 + a_1b_1)d_0d_1 + 2a_1b_2d_1^2 \\ &\quad -2a_1b_0d_0^2 - (4a_2b_0 + a_1b_1)d_0d_1 - 2a_2b_1d_1^2 \\ &= 2(a_0b_1 - a_1b_0)d_0^2 + 4(a_0b_2 - a_2b_0)d_0d_1 \\ &\quad +2(a_1b_2 - a_2b_1)d_1^2 \end{aligned}$$

for all $(0,0) \neq (d_0, d_1) \in \mathbb{F}_p^2$. This is equivalent to

$$\begin{aligned} a_0b_1 - a_1b_0 &= 0, &(3)\\ a_0b_2 - a_2b_0 &= 0, &(4)\\ a_1b_2 - a_2b_1 &= 0. &(5) \end{aligned}$$

Note that $Q(x_0, x_1)$ is a permutation over $\mathbb{F}_p^2$, then according to Lemma 2, $f_i(x_0, x_1), 0 \leq i \leq 1$ are balanced functions from $\mathbb{F}_p^2$ to $\mathbb{F}_p$. Then according to Lemma 3, we also have the matrices

$$\begin{pmatrix} 2a_0, & a_1 \\ a_1, & 2a_2 \end{pmatrix}, \begin{pmatrix} 2b_0, & b_1 \\ b_1, & 2b_2 \end{pmatrix}$$

are also not of full rank. This means

$$\begin{aligned} 4a_0a_2 - a_1^2 &= 0, &(6)\\ 4b_0b_2 - b_1^2 &= 0. &(7) \end{aligned}$$

In the following, we prove that

$$M = \begin{pmatrix} 2a_0, & a_1, & 2b_0, & b_1 \\ a_1, & 2a_2, & b_1, & 2b_2 \end{pmatrix}$$

is not of full rank. We prove this by the following two cases:

Case 1. $b_2 = 0$. Then by Equality (7) we get $b_1 = 0$. Then the matrix M becomes

$$M = \begin{pmatrix} 2a_0, & a_1, & 2b_0, & 0 \\ a_1, & 2a_2, & 0, & 0 \end{pmatrix}.$$

Let $m[i]$ denotes the i-th column of M. Then the submatrix $(M[0], M[1])$, and $(M[[0], M[2])$ are not of full rank according to Equality (6), (3). According to Equality (4), we have $a_2b_0 = 0$ since $b_2 = 0$. This also means $(M[1], M[2])$ is not of full rank. Then $\mathrm{rank}(M) < 2$ when $b_2 = 0$.

Case 2. $b_2 \neq 0$. Then the submatrices $(M[0], M[1])$, $(M[0], M[2])$, $(M[1], M[3])$ and $(M[2], M[3])$ are not of full rank according to Equality (6), (3), (5), (7) respectively. Next we prove that $(M[0], M[3])$ and $(M[1], M[2])$ are also not of full rank, which is equivalent to proving that

$$4a_0b_2 - a_1b_1 = 0, a_1b_1 - 4a_2b_0 = 0.$$

Multiplying Equality (5) by b_1, and substitute b_1^2 by Equality (7), we get

$$0 = a_1b_1b_2 - a_2b_1^2 = a_1b_1b_2 - 4a_2b_0b_2.$$

Then it holds

$$a_1b_1 - 4a_2b_0 = 0$$

since $b_2 \neq 0$. Note that according to Equality (4), we have $a_0b_2 = a_2b_0$ and hence

$$a_1b_1 - 4a_0b_2 = 0.$$

Therefore, $\mathrm{rank}(M) < 2$ when $b_2 \neq 0$.

Then there exist $(0,0) \neq (\delta_0, \delta_1) \in \mathbb{F}_p^2$, such that

$$(\delta_0, \delta_1) \cdot \begin{pmatrix} 2a_0, & a_1, & 2b_0, & b_1 \\ a_1, & 2a_2, & b_1, & 2b_2 \end{pmatrix} = (0,0,0,0).$$

According to Equality (2), this is equivalent to

$$\begin{aligned} &Q(x_0, x_1) - Q(x_0 - \delta_0, x_1 - \delta_1) \\ =\ &(-f_0(-\delta_0, -\delta_1), -f_1(-\delta_0, -\delta_1)) \end{aligned}$$

for all $(x_0, x_1) \in \mathbb{F}_p^2$. Then we complete the proof. ■

B. Quadratic Permutations over $\mathbb{F}_p^n$ and its Cryptographic Properties

In this subsection, we present a general method for constructing quadratic permutations over $\mathbb{F}_p^n$ for $n \geq 3$. We also establish the necessary conditions under which the resulting constructions achieve a differential uniformity of p^{n-1}. Furthermore, we characterize the Walsh coefficients of these permutations.

Theorem 2: Suppose $n \geq 3$, M_0, M_1 are two matrices over $\mathbb{F}_p$ with order $n-1$, and denote $M_x = x \cdot M_0 + M_1$, where $x \cdot M$ is the matrix $[M_{i,j}x]_{0 \leq i,j \leq n-2}$. Let $F(x_0, \ldots, x_{n-1}) = (x_0, M_{x_0} \cdot (x_1, \ldots, x_{n-1})^T)$. The following statements hold.

1) Let $d(x_0)$ be the polynomial of the determinant of M_{x_0}. If $d(x_0) = 0$ does not have roots in $\mathbb{F}_p$, then $F(x_0, \ldots, x_{n-1})$ is a permutation over $\mathbb{F}_p^n$.
2) If $\mathrm{rank}(M_0) = n-1$, then the differential uniformity of $F(x_0, \ldots, x_{n-1})$ equals p^{n-1}.

Proof:

1) Suppose there exists $(x_0, \ldots, x_{n-1}) \neq (y_0, \ldots, y_{n-1})$ such that

$$F(x_0, \ldots, x_{n-1}) = F(y_0, \ldots, y_{n-1}),$$

which is equivalent to

$$(x_0, M_{x_0}(x_1, \ldots, x_{n-1})^T) = (y_0, M_{y_0}(y_1, \ldots, y_{n-1})^T).$$

Therefore, we have $x_0 = y_0$, and hence $M_{x_0} = M_{y_0}$. Note that M_{x_0} is a matrix of full rank for all $x_0 \in \mathbb{F}_p$, since $d(x_0) \neq 0$ for all $x_0 \in \mathbb{F}_p$. Then we have $(x_1, \ldots, x_{n-1}) = (y_1, \ldots, y_{n-1})$. This means $F(x_0, \ldots, x_{n-1})$ is a permutation over $\mathbb{F}_p^n$.

2) Remember that $M_{x_0} = x_0 \cdot M_0 + M_1$, then $F(x_0, \ldots, x_{n-1})$ can be written as

$$\begin{aligned} &F(x_0, \ldots, x_{n-1}) \\ =\ &(x_0, x_0M_0(x_1, \ldots, x_{n-1})^T + M_1(x_1, \ldots, x_{n-1})^T). \end{aligned}$$

Let $(0, \ldots, 0) \neq (\Delta_0, \ldots, \Delta_{n-1}) \in \mathbb{F}_p^n$. Then it holds

$$\begin{aligned} &F(\bar{x}) - F(\bar{x} - \bar{\Delta}) \\ =&(\Delta_0, x_0M_0(x_1, \ldots, x_{n-1})^T + M_1(\Delta_1, \ldots, \Delta_{n-1})^T \\ &- (x_0 - \Delta_0)M_0(x_1 - \Delta_1, \ldots, x_{n-1} - \Delta_{n-1})^T) \\ =&(\Delta_0, x_0M_0(\Delta_1, \ldots, \Delta_{n-1})^T + \Delta_0M_0(x_1, \ldots, x_{n-1})^T \\ &+ M_{-\Delta_0}(\Delta_1, \ldots, \Delta_{n-1})^T) \\ =&(\Delta_0, \mathcal{M} \cdot (\bar{x})^T + M_{-\Delta_0}(\Delta_1, \ldots, \Delta_{n-1})^T), \end{aligned}$$

where $\mathcal{M}$ is a $(n-1)\times n$ matrix and $M_0(\Delta_1,\ldots,\Delta_{n-1})^T$ is the first column of $\mathcal{M}$ and the last column of $\mathcal{M}$ is $\Delta_0 M_0$, i.e., $\mathcal{M}=[M_0(\Delta_1,\ldots,\Delta_{n-1})^T,\Delta_0 M_0]$. Note that if $\mathrm{rank}(\mathcal{M})=n-k$, then the following difference equation

$$F(\bar{x})-F(\bar{x}-\bar{\Delta})=\bar{c} \tag{8}$$

has p^k or 0 solutions for any $(c_0,\ldots,c_{n-1})\in\mathbb{F}_p^n$. Then we investigate $\mathrm{rank}(\mathcal{M})$ as follows.

First, suppose $\Delta_0\neq 0$. Then

$$\mathrm{rank}(\Delta_0 M_0)=\mathrm{rank}(M_0)=n-1.$$

Therefore, $\mathrm{rank}(\mathcal{M})=n-1$, and hence the Equation (8) has p or 0 solutions all $(c_0,\ldots,c_{n-1})\in\mathbb{F}_p^n$.

Second, suppose $\Delta_0=0$. Then $\mathcal{M}=[M_0\cdot(\Delta_1,\ldots,\Delta_{N-1})^T,0_{(n-1)\times(n-1)}]$. Note that $\mathrm{rank}(M_0)=n-1$, then $M_0\cdot(\Delta_1,\ldots,\Delta_{n-1})^T$ can not be a zero vector. Therefore, $\mathrm{rank}(\mathcal{M})=1$ and hence the Equation (8) has 0 or p^{n-1} solutions for all $(c_0,\ldots,c_{n-1})\in\mathbb{F}_p^n$. Then we complete the proof. ∎

We characterize the Walsh coefficients of the above construction as follows.

Theorem 3: Suppose $n\geq 3$, M_0,M_1 are two matrices over $\mathbb{F}_p$ with order $n-1$, $\mathrm{rank}(M_0)=n-1$, and $M_x=x\cdot M_0+M_1$. Let $F(x_0,\ldots,x_{n-1})=(x_0,M_{x_0}\cdot(x_1,\ldots,x_{n-1})^T)$ be a permutation over $\mathbb{F}_p^n$, $\bar{a}=(a_0,\ldots,a_{n-1}),\bar{b}=(b_0,\ldots,b_{n-1})\in\mathbb{F}_p^n$, and

$$\lambda_F(\bar{a},\bar{b})=\sum_{\bar{x}\in\mathbb{F}_p^n}\chi(\bar{b}\cdot F(\bar{x})-\bar{a}\cdot\bar{x}).$$

Then it holds

$$|\lambda_F(\bar{a},\bar{b})|=\begin{cases}p^n, & a_0=b_0 \text{ and } b_i=a_i=0 \text{ for } 1\leq i\leq n-1,\\ p^{n-1}, & (b_1,\ldots,b_{n-1})\neq(0,\ldots,0), \text{ and } \mathrm{rank}(\mathcal{M}_l)\neq 2,\\ 0 & \text{others,}\end{cases}$$

where $\mathcal{M}_l=\begin{pmatrix}(b_1,\ldots,b_{n-1})M_0\\ (b_1,\ldots,b_{n-1})M_1-(a_1,\ldots,a_{n-1})\end{pmatrix}$.

Proof: For $\bar{a},\bar{0}\neq\bar{b}\in\mathbb{F}_p^n$, we have

$$\begin{aligned}&|\lambda_F(\bar{b},\bar{a})|^2\\ =\;&|\lambda_F(\bar{b},\bar{a})\cdot\overline{\lambda_F(\bar{b},\bar{a})}|\\ =\;&\left|\sum_{\bar{x}\in\mathbb{F}_p^n}\chi(\bar{b}\cdot F(\bar{x})-\bar{a}\cdot\bar{x})\cdot\sum_{\bar{y}\in\mathbb{F}_p^n}\chi(-\bar{b}\cdot F(\bar{y})+\bar{a}\cdot\bar{y})\right|\\ =\;&\left|\sum_{\bar{x},\bar{d}\in\mathbb{F}_p^n}\chi(\bar{b}\cdot(F(\bar{x})-F(\bar{x}-\bar{d}))-\bar{a}\cdot\bar{d})\right|.\end{aligned}$$

According to the proof of differential uniformity of $F(\bar{x})$, it holds

$$F(\bar{x})-F(\bar{x}-\bar{d})=(d_0,\mathcal{M}\cdot(\bar{x})^T+M_{-d_0}(d_1,\ldots,d_{n-1})^T),$$

where

$$\mathcal{M}=[M_0(d_1,\ldots,d_{n-1})^T,d_0M_0]$$

is an $(n-1)\times n$ matrix. Note that

$$(d_0,M_{-d_0}(d_1,\ldots,d_{n-1})^T)=-F(-\bar{d}),$$

then

$$\begin{aligned}&\bar{b}\cdot(F(\bar{x})-F(\bar{x}-\bar{d}))\\ =\;&\bar{b}\cdot(d_0,M_{-d_0}(d_1,\ldots,d_{n-1})^T)+(b_1,\ldots,b_{n-1})\cdot\mathcal{M}\cdot(\bar{x})^T,\end{aligned}$$

and hence

$$\begin{aligned}&|\lambda_F(\bar{b},\bar{a})|^2\\ =\;&\left|\sum_{\bar{x},\bar{d}\in\mathbb{F}_p^n}\chi(\bar{b}\cdot(F(\bar{x})-F(\bar{x}-\bar{d}))-\bar{a}\cdot\bar{d})\right|\\ =\;&\left|\sum_{\bar{d}\in\mathbb{F}_p^n}\left(\chi(f_{\bar{a},\bar{b},\bar{d}})\sum_{\bar{x}\in\mathbb{F}_p^n}\chi(\bar{b}[1:]\cdot\mathcal{M}\cdot\bar{x}^T)\right)\right|\\ =\;&\left|\sum_{\bar{d}\in\mathbb{F}_p^n}\left(\chi(f_{\bar{a},\bar{b},\bar{d}})\sum_{x_0\in\mathbb{F}_p}\chi(\bar{b}[1:]\cdot M_0\cdot(\bar{d}[1:])^Tx_0)\right.\right.\\ &\left.\left.\cdot\sum_{\bar{x}[1:]\in\mathbb{F}_p^{n-1}}\chi(\bar{b}[1:]\cdot(d_0M_0)\cdot(\bar{x}[1:])^T)\right)\right|,\end{aligned}\tag{9}$$

where $f_{\bar{a},\bar{b},\bar{d}}=-\bar{b}\cdot F(-\bar{d})-\bar{a}\cdot\bar{d}$, and $\bar{v}[1:]=(v_1,\ldots,v_{n-1})$ for the vector $\bar{v}=(v_0,v_1,\ldots,v_{n-1})\in\mathbb{F}_p^n$.

First, suppose $(b_1,\ldots,b_{n-1})=(0,\ldots,0)$. Then it holds

$$\begin{aligned}&|\lambda_F(\bar{b},\bar{a})|^2\\ =\;&p^n\cdot\left|\sum_{\bar{d}\in\mathbb{F}_p^n}\chi(-\bar{b}\cdot F(-\bar{d})-\bar{a}\cdot\bar{d})\right|\\ =\;&p^n\cdot\left|\sum_{\bar{d}\in\mathbb{F}_p^n}\chi(b_0d_0-\bar{a}\cdot\bar{d})\right|\\ =\;&p^n\cdot\left|\sum_{d_0\in\mathbb{F}_p}\chi((b_0-a_0)d_0)\right|\cdot\prod_{i=1}^{n-1}\left|\sum_{d_i\in\mathbb{F}_p}\chi(-a_id_i)\right|.\end{aligned}$$

This means $|\lambda_F(\bar{b},\bar{a})|=p^n$ when $(a_1,\ldots,a_{n-1})=(b_1,\ldots,b_{n-1})=(0,\ldots,0)$ and $a_0=b_0$, since

$$\sum_{x\in\mathbb{F}_p}\chi(ax)=\begin{cases}0, & a\neq 0,\\ p, & a=0.\end{cases}$$

Next, suppose $(b_1,\ldots,b_{n-1})\neq(0,\ldots,0)$. Then we have $d_0=0$, otherwise $|\lambda_F(\bar{b},\bar{a})|=0$ according to Equality (9). This is because $\mathrm{rank}(d_0M_0)=\mathrm{rank}(M_0)=n-1$ for $d_0\neq 0$, which implies $(b_1,\ldots,b_{n-1})\cdot(d_0M)\neq(0,\ldots,0)$, and hence $(b_1,\ldots,b_{n-1})\cdot(d_0M)\cdot(x_1,\ldots,x_{n-1})^T$ is a linear mapping with some coefficient nonzero, and then

$$\sum_{\bar{x}[1:]\in\mathbb{F}_p^{n-1}}\chi(\bar{b}[1:])\cdot(d_0M_0)\cdot(\bar{x}[1:])^T)=0.$$

Thus it holds

$$\begin{aligned}\chi(\bar{b}\cdot F(\bar{d})) &=\chi(\bar{b}\cdot(d_0,(d_0M_0+M_1)\cdot(d_1,\ldots,d_{n-1})^T))\\ &=\chi((b_1,\ldots,b_{n-1})M_1(d_1,\ldots,d_{n-1})^T).\end{aligned}$$

Remember that $\bar{v}[1:]=(v_1,\ldots,v_{n-1})$ for the vector $\bar{v}=(v_0,v_1,\ldots,v_{n-1})\in\mathbb{F}_p^n$. We denote

$$S_{\bar{d}}=\{\bar{d}[1:]\in\mathbb{F}_p^{n-1}\mid(\bar{b}[1:])M_0(\bar{d}[1:])^T=0\}.$$

Then $|S_{\bar{d}}| = p^{n-2}$, since $(b_1,\ldots,b_{n-1}) \neq (0,\ldots,0)$, $\mathrm{rank}(M_0) = n-1$ and hence $(b_1,\ldots,b_{n-1})\cdot M_0 \neq (0,\ldots,0)$. Therefore, according to Equality (9), it holds

$$\begin{aligned}
&|\lambda_S(\bar{b},\bar{a})|^2 \\
=\ & p^{n-1}\left|\sum_{\bar{d}\in\mathbb{F}_p^{n-1}} \chi(-\bar{b}\cdot F(-\bar{d}) - \bar{a}\cdot\bar{d})\right. \\
& \left.\cdot \sum_{x_0\in\mathbb{F}_p} \chi((b_1,\ldots,b_{n-1})\cdot M_0\cdot(d_1,\ldots,d_{n-1})^T x_0)\right| \\
=\ & p^n \left|\sum_{d_0=0,\bar{d}[1:]\in S_{\bar{d}}} \chi((\bar{b}[1:]M_1 - \bar{a}[1:])(\bar{d}[1:])^T)\right| \\
=\ & \begin{cases} p^{2n-2}, & (\bar{b}[1:])M_1 = \bar{a}[1:] \text{ for all } \bar{d}[1:]\in S_{\bar{d}}, \\ 0, & \bar{b}[1:]M_1 - \bar{a}[1:] \text{ is balanced on } S_{\bar{d}}. \end{cases}
\end{aligned}$$

Note that $(b_1,\ldots,b_{n-1})M_1 - (a_1,\ldots,a_{n-1})$ is balanced on the set $S_{\bar{d}}$ if and only if

$$\mathrm{rank}\begin{pmatrix} (b_1,\ldots,b_{n-1})M_0 \\ (b_1,\ldots,b_{n-1})M_1 - (a_1,\ldots,a_{n-1}) \end{pmatrix} = 2.$$

Then we complete the proof. ∎

C. Specification of a Quadratic Permutation over $\mathbb{F}_p^3$

Numerous quadratic permutations can be constructed using the methods described above. In the following, we present a special construction over $\mathbb{F}_p^3$, which we will use to design a new stream cipher that is suitable for FHE implementation. Its multiplication depth is 1, and it requires only 2 multiplications in the implementation.

Theorem 4: Let p be a prime number with $p = 2 \mod 3$. Let $S(\bar{x}) = (x_0, x_0x_2 + x_1, -x_0x_1 + x_0x_2 + x_2) \in \mathbb{F}_p^3[x_0,x_1,x_2]$. Then the following statements hold.

1) $S(\bar{x})$ is a permutation over $\mathbb{F}_p^3$.
2) The differential uniformity of $S(\bar{x})$ is p^2.
3) Let $\bar{a} = (a_0,a_1,a_2), \bar{b} = (b_0,b_1,b_2) \in \mathbb{F}_p^3$. Then

$$|\lambda_S(\bar{a},\bar{b})| = \begin{cases} p^3, & a_0 = b_0 \text{ and } b_i = a_i = 0 \text{ for } 1\le i\le 2, \\ p^2, & (b_1,b_2)\neq(0,0), \text{ and } \mathrm{rank}\begin{pmatrix} -b_2 & b_1+b_2 \\ b_1-a_1 & b_2-a_2 \end{pmatrix} \neq 2, \\ 0 & \text{others.} \end{cases}$$

Proof: Firstly, the matrix $M_{x_0} = \begin{pmatrix} 1 & x_0 \\ -x_0 & x_0+1 \end{pmatrix}$ and its determinant is $d_{x_0} = x_0^2 + x_0 + 1$. Then according to Theorem 2, $S(\bar{x})$ is a permutation over $\mathbb{F}_p^3$ if $d_{x_0} = 0$ does not have roots in $\mathbb{F}_p$. Note that the equation $x^2+x+1=0$ has no roots in $\mathbb{F}_p$ if and only if -3 is a quadratic non-residue modulo p. According to the well-known result of Legendre symbol, it holds $\left(\frac{-3}{p}\right) = 1$ when $p = 1 \mod 3$ and $\left(\frac{-3}{p}\right) = -1$ when $p = 2 \mod 3$ [35].

As for the differential uniformity of $S(\bar{x})$, note that

$$M_{x_0} = \begin{pmatrix} 1 & x_0 \\ -x_0 & x_0+1 \end{pmatrix} = x_0\begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and $\mathrm{rank}\begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix} = 2$, then according to Theorem 2, it holds that the differential uniformity of $S(\bar{x})$ is p^2.

Note that

$$\begin{aligned}
&\begin{pmatrix} (b_1,\ldots,b_{n-1})M_0 \\ (b_1,\ldots,b_{n-1})M_1 - (a_1,\ldots,a_{n-1}) \end{pmatrix} \\
=\ & \begin{pmatrix} (b_1,b_2)\begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix} \\ (b_1,b_2)\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - (a_1,a_2) \end{pmatrix} \\
=\ & \begin{pmatrix} -b_2 & b_1+b_2 \\ b_1-a_1 & b_2-a_2 \end{pmatrix},
\end{aligned}$$

then the Walsh coefficients can be computed directly by Theorem 3, and we complete the proof. ∎

Remark 1: Let p be an odd prime. We show that quadratic permutations over $\mathbb{F}_p^3$ with differential uniformity p^2 can always be constructed by using Theorem 2. For any odd prime p, there exist elements $a,b\in\mathbb{F}_p$ such that the polynomial x^2+ax+b is irreducible over $\mathbb{F}_p$. Consider the matrix $M_{x_0} = \begin{pmatrix} 1 & x_0 \\ -x_0 & ax_0+b \end{pmatrix}$, where $a,b\in\mathbb{F}_p$ are chosen such that x^2+ax+b is irreducible over $\mathbb{F}_p$. Then the determinant $d_{x_0} = x_0^2 + ax_0 + b$ has no roots in $\mathbb{F}_p$. By Theorem 2, the function

$$f(x_0,x_1,x_2) = (x_0, x_1 + x_0x_2, -x_0x_1 + ax_0x_2 + bx_2)$$

is a permutation over $\mathbb{F}_p^3$ with differential uniformity equal to p^2. The Walsh coefficients of the new permutation are also characterized directly by Theorem 3.

IV. Description of YuS

In this section, we provide a detailed description of YuS, a stream cipher designed for efficient evaluation in FHE schemes. YuS is a stream cipher taking key over $\mathbb{F}_p^{36}$ and produce keystream over $\mathbb{F}_p^{36-m}$. The keystream generation starts with a keywhitening, followed by a number of round transformations, as shown in Figure 1, and the last round truncates the first m words. For $\bar{x}\in\mathbb{F}_p^{36}$, the i-th round transformation is

$$\mathrm{RF}_i(\bar{x}) = \mathrm{AK}_{\overline{rk}^i}\circ\mathrm{SL}\circ\mathrm{LP}(\bar{x}),$$

where "LP", "SL" and "AK" are linear layer, S-box layer and adding roundkey operation, respectively. The full r-round YuS is

$$\mathsf{YuS}^r_{\mathrm{Key,nc}}(\bar{x}) = \mathrm{TF}_m\circ\mathrm{LP}\circ\mathrm{RF}_r\circ\cdots\circ\mathrm{RF}_1\circ\mathrm{AK}_{\overline{rk}^0}(\bar{x}),$$

where TF_m is the truncation function, and nc is the nonce.

Linear Layer. For $\bar{v} = (v_0,\ldots,v_{35})\in\mathbb{F}_p^{36}$, the linear layer of YuS is

$$\mathrm{LP}(\bar{v}) = M\cdot(v_0,\ldots,v_{35})^T,$$

where M is a matrix of order 36 and with entries be 0 or 1, see Appendix A for details.

S-box Layer. For $\bar{v} = (v_0,\ldots,v_{35})\in\mathbb{F}_p^{36}$, the S-box layer of YuS is

$$\mathrm{SL}(v_0,\ldots,v_{35}) = (S(v_0,v_1,v_2),\ldots,S(v_{33},v_{34},v_{35})).$$

As shown in Figure 2, the S-box of YuS is a permutation over $\mathbb{F}_p^3$, defined as follows

$$S(x_0,x_1,x_2) = (x_0, x_0x_2 + x_1, -x_0x_1 + x_0x_2 + x_2).$$

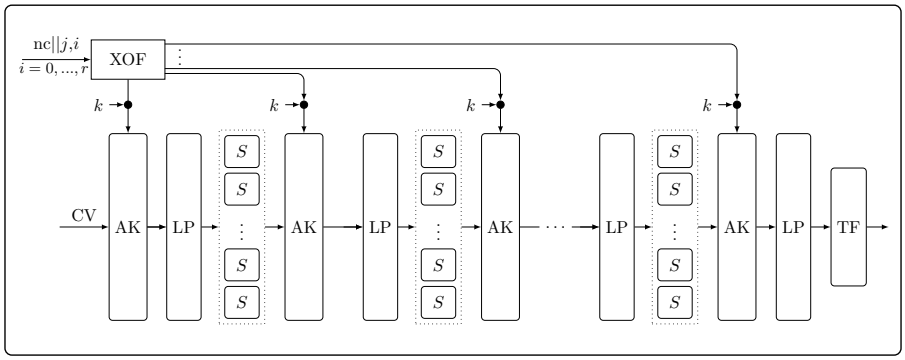

Fig. 1. The r-round YuS construction

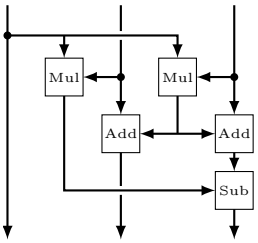

Fig. 2. S-box of YuS

Adding Round Key. For $\bar{v}, \overline{rk} \in \mathbb{F}_p^{36}$, the adding round key operation of YuS is

$$\mathrm{AK}_{\overline{rk}}(\bar{v}) = (v_0 + rk_0, \ldots, v_{35} + rk_{35}).$$

Truncation Function. A truncation function is applied in the last round before the keystream generation. It is a function from $\mathbb{F}_p^{36}$ to $\mathbb{F}_p^{36-m}$, defined by

$$\mathrm{TF}_m(v_0, \ldots, v_{35}) = (v_m, \ldots, v_{35}).$$

Round Key Generation. The round key generation in YuS employs the randomized key schedule method, initially introduced in HERA and also adopted by Rubato. Let $\bar{k} = (k_0, \ldots, k_{35}) \in \mathbb{F}_p^{36}$ be the primitive key. $\overline{rc}^i = (rc_0, \ldots, rc_{35}) \in (\mathbb{F}_p^*)^{36}$ are round constants generated by the nonce (nc) and the counter j with an extendable output function, which means

$$\mathrm{XOF}(\mathrm{nc}||j, i) = \overline{rc}^i, \;\; i = 0, \ldots, r.$$

The i-th round keys $\overline{rk}^i \in \mathbb{F}_p^{36}$ are generated by

$$\overline{rk}^i = (rc_0^i k_0, \ldots, rc_{35}^i k_{35}), \;\; i = 0, \ldots, r.$$

Keystream Generation. The t blocks keystream is generated by the counter mode as

$$\mathsf{YuS}_{\mathrm{KG}} = \mathsf{YuS}^r_{\mathrm{Key},\mathrm{nc}||j}(\mathrm{CV}), \;\; j = 0, \ldots, t-1,$$

where $\mathrm{CV} = (1, 2, \ldots, 36)$ is a constant vector over $\mathbb{F}_p^{36}$.

Recommendation Parameters.

- The base field of YuS is a prime field $\mathbb{F}_p$ with p larger than 16 bits, and $p = 2 \mod 3$.
- The recommended round number of YuS is 5 and 6 for 80-bit and 128-bit security respectively.
- The recommended truncated number m is 12.

V. Design Rationale

In this section, we explain the rationale behind the component and parameter choices in YuS. The primary goal of YuS is to serve as a stream cipher operating on the vector space over prime fields, enabling faster evaluation in BGV/BFV schemes and also achieving higher throughput.

A. Selection of S-box

We want to use an S-box over $\mathbb{F}_p^m$ that satisfies the following properties:

(1). It is a quadratic permutation, which means the multiplicity depth is only 1.
(2). Its differential uniformity should be less than p^m.
(3). Lower the multiplication complexity.

According to Theorem 1, quadratic permutations over $\mathbb{F}_p^2$ always have a differential uniformity of p^2. Then we choose permutations over $\mathbb{F}_p^m$ for $m \geq 3$. The construction of Theorem 2 requires $m-1$ multiplications over $\mathbb{F}_p$. Note that for a state contains n words, the number of multiplications per round transformation is $\frac{n}{m} \cdot (m-1)$. Thus, for a fixed state length, the number of multiplications is minimized when $m = 3$. Consequently, we adopt the construction from Theorem 2 for the case $m = 3$. By choosing suitable matrices M_0 and M_1, we get the construction in Theorem 4 as the S-box of YuS, which is defined as:

$$S(x_0, x_1, x_2) = (x_0, x_0x_2 + x_1, -x_0x_1 + x_0x_2 + x_2).$$

Furthermore, the odd prime p must satisfy $p = 2 \mod 3$ to ensure that the S-box is a permutation over $\mathbb{F}_p^3$.

B. Selection of the State Length

Based on the selection of the S-box, the state length n should be divided by 3. Note that the degree of the r-round words is 2^r, and there are nearly $\sum_{i=0}^{2^r} \binom{n+i-1}{i} = \binom{n+2^r}{2^r}$ different items in the last round by a rough estimation. Thus in order to resist the trivial linearization attack in Section VI-B, the $\log(2, \binom{n+2^r}{2^r})$ should be at least half of the security parameters. We want to reach 80-bit security by 4 rounds. According to Table II, we choose $n = 36$ and then we have 12 S-boxes in one round transformation of YuS.

TABLE II
ESTIMATION OF THE NUMBER OF 4-ROUND MONOMIALS

n	30	33	36
$\log(2, \binom{n+16}{16})$	39.85	41.61	43.24

C. Selection of the Linear Mapping

Based on the above selection of the number of S-boxes in one round transformation, we aim to use a linear mapping over $(\mathbb{F}_p^3)^{12}$ with the following properties:

(1). Maximize the differential branch number and linear branch number.
(2). Avoid using the constant multiplication in $\mathbb{F}_p$.

To achieve this, we generate circulant matrices of order 12, where each entry is a 3×3 binary matrix, and test their differential and linear branch numbers. Based on our search, we did not find MDS matrices over $(\mathbb{F}_p^3)^{12}$. The maximum differential branch number and the maximum linear branch number we obtained are 10 and 6, respectively. It should be noted that the definition of the linear branch number for the matrix differs from the conventional one. Further details are provided in Appendix C.

D. Selection of the Truncation Number

At the final round, we applied a truncation operation, following the style of PASTA and Rubato. The function TF_{12}, defined by $\mathrm{TF}_{12}(x_0, x_1, \ldots, x_{35}) = (x_{12}, \ldots, x_{35})$, maps $\mathbb{F}_p^{36}$ to $\mathbb{F}_p^{24}$ and is balanced. This is because for any output $\bar{v} \in \mathbb{F}_p^{24}$ of TF_{12}, there exist precisely p^{12} input vectors $\bar{x} \in \mathbb{F}_p^{36}$ satisfying $\mathrm{TF}_{12}(\bar{x}) = \bar{v}$.

The purpose of truncation is to obscure the lower-degree equations in the last round. According to the S-box, for the r rounds, the words after the S-box layer consist of 12 words with degree 2^{r-1} and 24 words with degree 2^r. When the last round state is directly output as the keystream, multiplying the inverse of the linear mapping to the keystream yields 12 lower-degree equations with a single query. Even if the truncated number is less than 12, one can still introduce new variables in the last round state and derive the lower-degree equations. Therefore, 12 words are truncated before outputting the keystream.

VI. SECURITY ANALYSIS OF YUS

In this section, we present the security analysis of YuS. Before our evaluations, we outline the security claim of YuS. First, the amount of data encrypted under the same key is limited to $2^{\frac{\lambda}{2}}$ for λ-bit security. Second, the security analysis focuses on the "secret-key model". Other models are beyond the scope of YuS's security guarantees. The number of rounds to prevent all attacks considered in this section and the analysis results are summarized in Table III, where we assume that $p > 2^{16}$.

TABLE III
RECOMMENDED NUMBER OF ROUNDS TO RESIST ALL ATTACKS AND THEIR COMPLEXITY RESULTS, WHICH ARE DENOTED BY $(\text{ROUNDS}, \log_2(\text{TIME}))$. THE LINEAR ALGEBRA CONSTANT ω IS ASSUMED TO BE 2.

Types	$\lambda = 80$	$\lambda = 128$
Linear/Differential Attack	$(2, /)$	$(2, /)$
Trivial Linearization Attack	$(4, 86.5)$	$(5, 128.9)$
Gröbner Basis Attack	$(4, \geq 86.5)$	$(5, \geq 128.9)$
XL Attack	$(2, 106.5)$	$(3, 211.3)$
GCD Attack	$(1, 479.0)$	$(1, 479.0)$

A. Linear Attack

The maximum differential probability and linear probability (squared correlation), which is defined as

$$DP^F = \frac{\Delta(F)}{q}, LP^F = \left(\frac{\mathcal{L}(F)}{q}\right)^2$$

respectively [36], [37], is widely used to measure the resistance of symmetric ciphers to differential attack and linear attack.

Note that the Walsh transform of a nontrivial linear active S-box is p^2, then linear probability of the linear trail of YuS equals $(\frac{p^2}{p^3})^{2 \cdot nl}$, where nl means the number of nontrivial linear active S-boxes in the linear trail. As for the matrix used in YuS, the linear branch number is 6, see Appendix C for details. This means every two rounds of YuS contains at least 6 nontrivial linear active S-boxes. Then for r-round YuS, the linear trail probability is at least

$$p^{-2 \cdot \lfloor \frac{r}{2} \rfloor \cdot 6} = p^{-12 \cdot \lfloor \frac{r}{2} \rfloor}.$$

Then 2 rounds achieve the 80-bit and 128-bit security of YuS since p is at least 16-bit.

Similarly, it can also be inferred that two rounds of YuS can resist differential attacks for both 80-bit and 128-bit security levels. This is because the differential uniformity of the S-box is p^2 according to Theorem 4, which means one active S-box provides a differential probability of p^{-1}. Furthermore, the differential branch number of YuS is 10 (Appendix C), then the probability of 2-round differential trail is at least p^{-10}.

B. Trivial Linearization Attack

Since YuS employs a randomized key schedule scheme and counter-mode keystream generation, we do not need to consider other statistical attacks such as high-order differential attacks [38], integral attacks [39], and cube attacks [40]. However, based on the design principle of FHE-friendly ciphers, the S-box of YuS is a quadratic permutation with a multiplication depth of 1 over $\mathbb{F}_p^3$, making it vulnerable to algebraic attacks. Therefore, the subsequent cryptanalysis will focus on the algebraic analysis of YuS.

The trivial linearization attack is the simplest method for solving a system of multivariate equations by replacing all monomials with independent variables. For r rounds of YuS, the keystream z can be trivially represented by 24 equations of degree 2^r over $\mathbb{F}_p$, involving 36 key variables $(k_0, k_1, \ldots, k_{35})$. All the monomials of degree at most 2^r are expected to appear after r rounds of YuS (as explained in

Appendix B). Hence, the number of monomials appearing in these equations is upper bounded by

$$N(r) = \sum_{i=0}^{2^r} \binom{36+i-1}{i} = \binom{36+2^r}{2^r}, \tag{10}$$

we need to collect the same number of equations using $Q(r) = \frac{N(r)}{24}$ encryption queries under the same key. After replacing all monomials with new variables, the key can be recovered by solving this system of equations through Gaussian elimination. Then the time complexity is estimated as

$$T(r,\omega) = (N(r))^{\omega} = \binom{36+2^r}{2^r}^{\omega}, \tag{11}$$

where ω is the linear algebra constant, satisfying $2 \leq \omega \leq 3$. We assume $\omega = 2$ here and determine the value of r such that $T(r,2) > 2^{\lambda}$ to resist trivial linearization attacks. As a result, 4 rounds and 5 rounds can achieve 80-bit and 128-bit security of YuS, respectively.

Guess-and-determine. Guess-and-Determine (GnD) strategy is a very common method used by attackers to optimize the complexity of the key-recovery attacks. Here, we consider an application of the GnD strategy before trivial linearization. Assuming that enough equations are available, if we select to guess k variables, the time complexity to solve the remaining variables is estimated as

$$T'(r,\omega) = p^k \times \binom{36-k+2^r}{2^r}^{\omega}.$$

However, we find that even with $p = 2^{16}+1$, this strategy will not work. For example, the best complexity for YuS with 80-bit security is $T'(4,2) = 2^{101.41}$, implying that guessing variables is relatively expensive under the current prime field size.

Meet-in-the-Middle. Similarly, Meet-in-the-Middle (MITM) strategy is also an effective technique in cryptanalysis. Let us examine the inverse S-box of YuS, since

$$M^{-1} = \begin{pmatrix} 1 & x \\ -x & x+1 \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1+x}{1+x+x^2} & \frac{-x}{1+x+x^2} \\ \frac{x}{1+x+x^2} & \frac{1}{1+x+x^2} \end{pmatrix},$$

then the algebraic expression of the inverse S-box can be written as

$$S^{-1}(y_0,y_1,y_2) = \left(y_0, \frac{(1+y_0)y_1 - y_0y_2}{1+y_0+y_0^2}, \frac{y_0y_1+y_2}{1+y_0+y_0^2}\right).$$

Given its high algebraic degree, performing an MITM attack where the state passes through the S-box layer in reverse is not advisable. Moreover, the truncation operation in the final round can also effectively prevent this attack. Specifically, assuming the number of truncated words for r-round YuS is m, we can introduce new variables to replace them and perform a MITM attack on the output state of the final S-box layer. This will generate 12 low-degree equations based on the first output word of the S-box. Therefore, as long as $m \geq 12$, it becomes impossible to use multiple encryption queries to collect enough low-degree equations to achieve better attack results.

C. Gröbner Basis Attack

Given a multivariate polynomial system over the field $\mathbb{F}$ as

$$\begin{aligned} E_1(x_0,x_1,\ldots,x_{n-1}) &= E_2(x_0,x_1,\ldots,x_{n-1}) = \cdots \\ &= E_m(x_1,x_2,\ldots,x_{n-1}) = 0, \end{aligned}$$

where $deg(E_i) = d_i$ $(1 \leq i \leq m)$. The Gröbner basis attack is currently one of the most effective methods to solve this system and can be divided into the following steps:

1) Compute a Gröbner basis in the *graded reverse lexicographic* order with the F5 algorithm [41].
2) Convert it into a *lexicographic* order Gröbner basis using the FGLM algorithm [42].
3) Find and solve a univariate polynomial in the Gröbner basis, substitute the solution and repeat this step.

When applied to the r-round YuS, multiple encryption queries under the same key can yield more equations than key variables, thereby constructing an overdetermined system of equations $E_i(k_0,k_1,\ldots,k_{35})$, where $d_i = 2^r$. Hence, we analyze the security of YuS against Gröbner basis attack under the semi-regular assumption [43], thus the time complexity of computing the Gröbner basis dominates and is estimated as

$$O\left(\binom{36+d_{reg}}{d_{reg}}^{\omega}\right),$$

where d_{reg} is the degree of regularity, which can be estimated by the degree of the first non-positive coefficient in the Hilbert series

$$\begin{aligned} H(x) &= \frac{1}{(1-x)^{36}} \times \prod_{i=1}^{m}(1-x^{2^r}) \\ &= \left(\sum_{i=0}^{2^r-1} x^i\right)^{36} \times (1-x^{2^r})^{m-36}. \end{aligned}$$

Note that as m increases and the estimated d_{reg} decreases, then the value of d_{reg} cannot be less than 2^r, since the non-positive coefficient terms only arise from $(1-x^{2^r})^{m-36}$. Assuming $d_{reg} = 2^r$, the corresponding time complexity of this attack is estimated as

$$O\left(\binom{36+2^r}{2^r}^{\omega}\right).$$

Compared to Equation (11), we find that the Gröbner basis attack is no better than the trivial linearization attack. Similarly, it is obviously ineffective if combined with a GnD strategy.

Other modeling methods. Besides algebraic modeling directly from the keystream output, other methods involve introducing intermediate variables to derive low-degree equations. For example, we can linearize all S-boxes of r-round YuS by introducing intermediate variables for the last two output words. This requires $24r$ new variables, and the same number of quadratic equations can be derived based on the algebraic expression of the S-box. However, due to the truncation operation, only 24 linear equations can be extracted from the keystream, resulting in an underdetermined system that cannot be solved. We note that obtaining enough equations through multiple encryption queries is impractical, as each encryption permutation is random and the intermediate variables must be

reintroduced. Likewise, other modeling methods that introduce intermediate variables will also fail.

D. XL Attack

The XL algorithm [44] is another effective technique for expanding and solving multivariate polynomial systems. In brief, the algorithm works as follows: given a target degree D, it multiplies all possible monomials of degree $\leq D - d_i$ by the original system to expand the number of equations. Once the number of equations exceeds the number of monomials, the extended system can be solved by trivial linearization. The correctness of the XL algorithm relies on a strong assumption that all the resulting equations are linearly independent.

Applied to r rounds of YuS, we introduce 12 new variables to represent the truncated words under each encryption query. Based on the MITM idea, there are 12 equations of degree 2^{r-1} and 24 equations of degree 2^r can be derived from the output state of the last S-box layer. Then the XL algorithm under $q \geq 2$ queries needs to determine the minimum degree D such that

$$\begin{aligned}&\sum_{i=0}^{D-2^{r-1}} \binom{36+12q+i-1}{i} \cdot 12q \\ &+ \sum_{i=0}^{D-2^{r}} \binom{36+12q+i-1}{i} \cdot 24q \\ &\geq \sum_{i=1}^{D} \binom{36+12q+i-1}{i} = T_D,\end{aligned}$$

while the time complexity of applying trivial linearization is approximately $O(T_D^\omega)$. Even with the assumption of $\omega = 2$, only 2 rounds and 3 rounds of YuS are sufficient to achieve 80-bit and 128-bit security, respectively. Apart from this, there seems to be no better modeling methods for XL attacks.

E. Other Algebraic Attacks

In the security analysis of both HERA [8] and Rubato [26], the so-called GCD attack was considered, which is a technique for solving univariate systems by computing the greatest common divisor of all univariate polynomials. If we guess all but one of the key variables, this attack can be extended to a multivariate polynomial system modeled by the cipher. For r-round YuS, the time complexity of this attack is estimated as $O(p^{35}r^2 2^r)$. Therefore, even when applied to a single-round instance with $p > 2^{16}$, the required complexity is at least 2^{497}. Furthermore, Liu et al. [45] recently proposed an attack against HERA using multiple collisions, which can effectively peel off the last round of nonlinear layers. However, their attack relies on the complete keystream output, whereas YuS, which incorporates the truncation operation, is resistant to this attack.

VII. Implementation and Comparison

We implemented YuS in the regular transciphering framework using Brakerski-Gentry-Vaikuntanathan (BGV) scheme in HElib version 2.3.0 [29] and Brakerski-Fan-Vercauteren (BFV) scheme in SEAL version 4.1.2 [30].

All of our experiments in Table IV, VII and VIII were conducted on a laptop with a 12th Gen Intel(R) Core(TM) i5-12500H @ 2.50 GHz processor, 13 GiB of memory, and 18 GiB of swap. The operating system is WSL2 Ubuntu 22.04 LTS. We developed the source code in C++17 and compiled it with GNU C++ 11.4.0, without using the Intel® HEXL acceleration library. The `CMAKE_BUILD_TYPE` is set to `Release`. Moreover, each experiment has access to only one thread. Specially, the source code of YuS is stored on github platform. [1]

A. Benchmark

Packing Method for FHE Evaluation. Generally, there are two different packing strategies for FHE evaluation: row-wise packing and column-wise packing.

In column-wise packing, we can input N messages into a vector $\mathcal{V}$,

$$\begin{aligned}\mathcal{V} = \; &(\underbrace{m_{0,0}, m_{0,1}, \cdots, m_{0,B-1}}_{block_0}, \underbrace{m_{1,0}, m_{1,1}, \cdots, m_{1,B-1}}_{block_1}, \\ &\ldots, \underbrace{m_{s',0}, m_{s',1}, \cdots, m_{s',B-1}}_{block_{s'}}),\end{aligned}$$

where B denotes the word length of a block keystream and N denotes the degree of cyclotomic polynomial in the HE schemes. The encoding and encryption are performed as $Ptxt = \texttt{encode}(\mathcal{V})$ and $Ctxt = \texttt{encrypt}(Ptxt)$. This packing strategy enables the concurrent processing of s' blocks using a single ciphertext, where $s' = \lfloor \frac{N}{B} \rfloor$.

In row-wise packing, $B \times N$ integer messages from $\mathbb{Z}_p$ are arranged into a matrix $\mathcal{M}$ as

$$\mathcal{M} = \begin{pmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,N-1} \\ m_{1,0} & m_{1,1} & \cdots & m_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{B-1,0} & m_{B-1,1} & \cdots & m_{B-1,N-1} \end{pmatrix}.$$

Then the encoding and encryption are performed as $Ptxt_i = \texttt{encode}(m_{i,0}, m_{i,1}, ..., m_{i,N-1})$ and $Ctxt_i = \texttt{encrypt}(Ptxt_i)$, $i = 0, ..., B-1$. This packing strategy allows us to process N blocks using B ciphertexts.

Considering throughput as the most important indicator of evaluating ciphers, the experiments in paper [8] show that row-wise packing outperforms column-wise packing. Therefore, we exclusively adopt row-wise packing for our evaluation.

XOF. We employ SHAKE128 [46][2] as our sole XOF implementation for all evaluated ciphers, ensuring consistent generation of both random vectors and random invertible matrices.

Parameters. The parameter selection involves three critical factors: security, performance, and functionality. Among these, the security level is predominantly determined by two parameters: the degree of cyclotomic polynomial (N) and the modulus of ciphertext coefficients (q). In HElib, the security level is

[1] https://github.com/wfzcipher/YuS_2025_7z.

[2] The shake128 source code and the client-side cipher implementations (except for YuS) are available in the open-source hybrid-HE-framework https://github.com/isec-tugraz/hybrid-HE-framework.

TABLE IV
CLIENT-SIDE PERFORMANCE COMPARISON

Cipher	Parameters	p	Block	Off (Cycles)	On (Cycles)	Total (Cycles)	CPB(Cycles/Byte)
80-bit							
Masta	(4,32,32)	65537	16384	1700141673	5793993	1705935666	1531
Masta	(5,16,16)	65537	16384	651643131	2693527	654336658	1174
HERA	(4,16,16)	65537	16384	245384264	2390934	247775198	444
YuS	(5,36,24)	65537	16384	364430598	3012380	367442978	439
128-bit							
Masta	(6,32,32)	65537	16384	2227989261	5578310	2233567571	2004
HERA	(5,16,16)	65537	16384	276798534	2426025	279224559	501
PASTA	(4,64,32)	65537	16384	6048401761	5022871	6053424632	5433
PASTA_{v2}	(4,64,32)	65537	16384	3532939027	4649743	3537588770	3175
YuS	(6,36,24)	65537	16384	397769671	2725932	400495603	479

Parameters (r, ks, ps): r, ks, ps denote the round number, key size, and plain/cipher size.
Block: the number of blocks in plaintext stream.
Total: the total time.
CPB: CPB = $\frac{8\times\text{Total}}{ps\times\text{Block}\times\lceil\log p\rceil}$

calculated as $\frac{3.8\times N}{\log q-\log 3.2}-20$. For a given N, the parameter q has an upper bound to achieve the desired security level (80 or 128 bits in this paper). Conversely, the size of q is inversely proportional to security, which means that q also has a lower bound. What's more, a small q would result in insufficient noise budget, thereby compromising the scheme's functionality. Considering the design principles are different in HElib and SEAL, we developed a corresponding strategy to select the proper parameters and evaluate ciphers. For BGV in HElib, we can find the lower bound to ensure a tight noise budget when the security level is much greater than 80 or 128 bits. For BFV in SEAL, we use the upper bound to ensure enough noise budget when the security level is equal to 80 or 128 bits. Especially, the SEAL parameters we select from the Security Guidelines for Implementing Homomorphic Encryption [47].

Linear layer implementation of YuS. Generally, the number and depth of multiplication are the main determinants of run time in FHE scheme. However, due to the high number of additions in YuS, the cost of the linear layer (36×36 matrix) is not negligible. Considering the linear matrix of YuS in $\{0,1\}$, we use the "method of the four Russians" [48] in our implementation, which reduces the number of additions from 876 to 412.

B. Comparsion

We evaluate the performance of YuS against Masta, HERA, PASTA and PASTA_{v2} from both client-side and server-side perspectives. To ensure a fair comparison, all ciphers are configured with $p = 65537$ (17-bit).

Client-side. The client-side experimental results are summarized in Table IV. The client-side computation comprises two components: client-off time (for randomization and keystream generation) and client-on time (for plaintext stream encryption using the generated keystream). All measured in the number of clock cycles in CPU. We use Cycles Per Byte (CPB) as the indicator to evaluate the performance of cipher in client-side. It's worth noting that our evaluation does not take into account the time of fully homomorphic encryption for symmetric keys since the encrypted symmetric keys will be used over multiple sessions once it is computed.

The experimental results demonstrate that YuS outperforms other ciphers in terms of CPB at both 80-bit and 128-bit security levels. Notably, YuS exhibits approximately $5.6\times$ faster processing speed compared to PASTA_{v2}, the state-of-the-art cipher for HHE.

Homomorphic operations. In TableV, we define the depth cost of operations as follows: multiplication of two homomorphic ciphertexts incurs a depth cost of 1 unit; multiplication of a constant by a homomorphic ciphertext incurs a depth cost of 1/2 unit; and multiplication of a plaintext vector by a homomorphic ciphertext incurs a depth cost of 2/3 unit. In TableVI, we summarize the available homomorphic operations and multiplication-depth for ciphers we evaluated.

TABLE V
NOISE LEVEL FOR HOMOMORPHIC OPERATIONS.

Operation	Description	Noise level
MultiplyConstant	Ciphertext ct and constant m	1/2
MultiplyVector	Ciphertext ct and vector m	2/3
Multiply	Ciphertext ct_1 and ct_2	1

TABLE VI
HOMOMORPHIC OPERATIONS AND MULTIPLICATION-DEPTH COMPARISON

Cipher	Parameters	Mul	Const-mul	Add	Const-add	Depth
80-bit						
Masta	(4,32,32)	128	5120	5248	160	7.333
Masta	(5,16,16)	80	1536	1616	96	9.000
HERA	(4,16,16)	128	80	1024	16	8.666
YuS	(5,36,24)	120	216	2832	36	5.666
128-bit						
Masta	(6,32,32)	192	7168	7360	224	10.666
HERA	(5,16,16)	160	96	1232	16	10.666
PASTA	(4,64,32)	330	10240	10170	320	8.333
PASTA_{v2}	(4,64,32)	330	10240	10170	320	7.666
YuS	(6,36,24)	144	252	3316	36	6.666

Server-side. For the same reason as on client-side, the initialization time for the FHE schemes is excluded from the server-side evaluation. The server-off time consists of the time for randomization and the time for homomorphic evaluation of keystream. The server-on time refers to the time used to generate encrypted plaintext stream using encrypted keystream and ciphertext stream. The total time is the sum of server-off time and server-on time. All measured in high precision clock

TABLE VII
SERVER-SIDE PERFORMANCE COMPARISON IN HELIB-2.3.0

Cipher	Parameters	p	N	Block	$\lceil \log q \rceil$	λ'	Off (s)	On (s)	Total (s)	Throughput (KiB/s)
80-bit										
Masta	(4,32,32)	65537	16384	16384	507	103.429	18.251	0.087	18.338	59.331
Masta	(5,16,16)	65537	32768	32768	623	180.465	22.209	0.118	22.327	48.730
HERA	(4,16,16)	65537	32768	32768	580	195.408	12.978	0.116	13.094	83.094
YuS	(5,36,24)	65537	16384	16384	375	147.027	7.425	0.049	7.474	109.179
128-bit										
Masta	(6,32,32)	65537	32768	32768	758	144.791	89.644	0.292	89.936	24.195
HERA	(5,16,16)	65537	32768	32768	744	147.886	19.633	0.141	19.774	55.022
PASTA	(4,64,32)	65537	32768	32768	623	180.465	100.192	0.151	100.343	21.686
PASTA_{v2}	(4,64,32)	65537	32768	32768	570	199.480	57.020	0.153	57.173	38.060
YuS	(6,36,24)	65537	32768	32768	422	276.945	19.118	0.114	19.233	84.856

Parameters (r, ks, ps): r, ks, ps denote the round number, key size, and plain/cipher size.
N: degree of the cyclotomic polynomial of BGV in HElib.
Block: the number of blocks in plaintext stream.
$\lceil \log q \rceil$: the ciphertext modulus bits of BGV in HElib.
λ': the security level of BGV in HElib.
Throughput: Throughput $= \frac{ps \times \text{Block} \times \lceil \log p \rceil}{8192 \times \text{Total}}$

TABLE VIII
SERVER-SIDE PERFORMANCE COMPARISON IN SEAL-4.1.2 (BFV SECURITY LEVEL $\lambda = 128$ BIT)

Cipher	Parameters	p	N	Block	$\lceil \log q \rceil$	Off (s)	On (s)	Total (s)	Throughput (KiB/s)	Noise Budget(bit)
80-bit										
Masta	(4,32,32)	65537	16384	16384	424	35.854	0.021	35.875	30.327	111
Masta	(5,16,16)	65537	16384	16384	424	12.948	0.011	12.959	41.980	60
HERA	(4,16,16)	65537	16384	16384	424	6.813	0.011	6.825	79.712	67
YuS†	(5,36,24)	65537	16384	16384	301	6.169	0.009	6.178	132.077	52
YuS	(5,36,24)	65537	16384	16384	424	7.358	0.011	7.369	110.741	164
128-bit										
Masta	(6,32,32)	65537	16384	16384	424	52.775	0.021	52.796	20.608	4
HERA	(5,16,16)	65537	16384	16384	424	8.312	0.013	8.325	65.349	4
PASTA	(4,64,32)	65537	16384	16384	424	74.687	0.022	74.709	14.563	74
PASTA_{v2}	(4,64,32)	65537	16384	16384	424	32.726	0.021	32.747	33.224	98
YuS†	(6,36,24)	65537	16384	16384	301	7.600	0.010	7.610	107.229	20
YuS	(6,36,24)	65537	16384	16384	424	8.822	0.010	8.832	92.392	133

Parameters (r, ks, ps): r, ks, ps denote the round number, key size, and plain/cipher size.
N: degree of the cyclotomic polynomial of BFV in SEAL.
Block: the number of blocks in plaintext stream.
$\lceil \log q \rceil$: the ciphertext modulus bits of BFV in SEAL.
Throughput: Throughput $= \frac{ps \times \text{Block} \times \lceil \log p \rceil}{8192 \times \text{Total}}$
Noise Budget: the final noise budget for encrypted plaintext stream after BFV evaluation in SEAL.
YuS†: the BFV security level is $\lambda = 192$ for this set of parameters, with $\lceil \log q \rceil = 301$.

of C++ (`std::chrono::high_resolution_clock`). We use throughput as the indicator to evaluate the performance of cipher in server-side.

In Table VII, we summarise the results of implementation in HElib. The experimental data demonstrates that YuS achieves optimal performance among all evaluated ciphers in terms of throughput at both 80-bit and 128-bit security levels, while also achieving minimal execution time at the 80-bit security level.

Table VIII presents our implementation results using SEAL. The BFV security level is 128-bit in all ciphers. Obviously, YuS achieves superior performance with the highest throughput and noise budget at both 80-bit and 128-bit security levels. While YuS has a little longer execution time (approximately 0.5 seconds) than HERA, it provides significantly enhanced functionality with nearly $2\times$ and $33\times$ larger noise budgets at 80-bit and 128-bit security levels, respectively, demonstrating YuS's superior functional capabilities compared to HERA.

Additionally, the server-side performance comparison with $p = 4298506241$ (33-bit) is presented in Appendix D.

VIII. CONCLUSION

In this paper, we investigate the problem of constructing nontrivial quadratic permutations over $\mathbb{F}_p^n$ for odd primes p. We prove that quadratic permutations over $\mathbb{F}_p^2$ have a differential uniformity of p^2. Additionally, we propose a method for constructing quadratic permutations over $\mathbb{F}_p^n$ with differential uniformity p^{n-1} for $n \geq 3$. The Walsh coefficients of these permutations are also characterized. Based on the newly constructed quadratic permutations over $\mathbb{F}_p^3$, we propose YuS, an FHE-friendly stream cipher. Our detailed implementations demonstrate that YuS is the fastest prime field-based FHE-friendly stream cipher while also achieving the highest throughput. Furthermore, the permutations constructed in this paper are valuable and can be utilized in designing other arithmetic primitives. For future research, the problem of constructing quadratic permutations over $\mathbb{F}_p^n$ with nontrivial Walsh coefficients requires further study. It is also of particular

interest to investigate implementation optimizations on embedded and GPU platforms to broaden the practical applicability of FHE-friendly ciphers, and to evaluate the resistance of such ciphers against side-channel attacks to strengthen real-world security.

APPENDIX A
THE MATRIX OF LINEAR LAYER

The matrix used in the linear layer of YuS is

$$M = \begin{bmatrix}
1101111110010011101111010110001110111\\
1111101010101101011011111111010011110\\
0100110111101010111111010111111111101\\
1111101111110010011110111101100011110\\
1101111101010101101011011111111010011\\
1010100110111101010111111010111111111\\
1101111101111111001001111011110110001\\
0111101111101010101101011011111111010\\
1111010100110111101010111111010111111\\
0011101111101111110010011110111101110\\
0100111101111101010101101011011111111\\
1111111010100110111101010111111010011\\
1100011101111101111110010011110111110\\
1110100111101111101010101101011011111\\
0111111111010100110111101010111111101\\
1101100011101111101111110010011110111\\
1111110100111011111010101011010110101\\
1010111111111010100110111101010111111\\
0111101100011101111101111110010011111\\
1011111110100111101111101010101101011\\
1111010111111111010100110111101010111\\
1110111101100011101111101111110010011\\
1011011111111010011110111101010101110\\
0111111010111111111010100110111101011\\
0011110111101100011101111101111110011\\
1101011011111111010011110111101010101\\
1010111111010111111111010100110111110\\
0010011110111101100011101111101111111\\
0101101011011111111010011110111101011\\
1101010111111010111111111010100110111\\
1110010011110111101100011101111101111\\
1010101101011011111111010011110111110\\
0111101010111111010111111111010100111\\
1111110010011110111101100011101111110\\
1101010101101011011111110100111101111\\
0110111101010111111010111111111010101
\end{bmatrix}.$$

APPENDIX B
NUMBER OF MONOMIALS IN YuS

Consider the round transformation

$$\mathrm{RF}(\bar{x}) = \mathrm{AK}_{\overline{rk}} \circ \mathrm{SL} \circ \mathrm{LP}(\bar{x})$$

of YuS with input $\bar{x} = (x_0, \ldots, x_{n-1})$ and round key $\overline{rk} = (rk_0, \ldots, rk_{n-1})$, where the linear layer LP can be represented by an $n \times n$ matrix M. Let $M_{i,j}$ $(i, j \in [0, n-1])$ be the entry in the $(i+1)$-th row and the $(j+1)$-th column of the matrix M, $\mathrm{RF}(\bar{x})[i]$ be the i-th component of the output of $\mathrm{RF}(\bar{x})$.

Due to the specific S-box layer SL of YuS, $\mathrm{RF}(\bar{x})[i]$ can be categorized into three cases as follows:

$$\mathrm{RF}(\bar{x})[i] = \begin{cases}
\sum_{j=0}^{n-1} e_j^{(i)} x_j + rk_i, \ \ i \in S_0 = \{i \mid i \bmod 3 = 0\},\\
\sum_{j=0}^{n-1} a_j^{(i)} x_j^2 + \sum_{j=0}^{n-1} \sum_{k=j+1}^{n-1} b_{j,k}^{(i)} x_j x_k + \sum_{j=0}^{n-1} e_j^{(i)} x_j + \\
rk_i, \ \ i \in S_1 = \{i \mid i \bmod 3 = 1\},\\
\sum_{j=0}^{n-1} c_j^{(i)} x_j^2 + \sum_{j=0}^{n-1} \sum_{k=j+1}^{n-1} d_{j,k}^{(i)} x_j x_k + \sum_{j=0}^{n-1} e_j^{(i)} x_j + \\
rk_i, \ \ i \in S_2 = \{i \mid i \bmod 3 = 2\},
\end{cases}$$

where

$$a_j^{(i)} = M_{i-1,j} M_{i+1,j}, \ \ b_{j,k}^{(i)} = M_{i-1,j} M_{i+1,k} + M_{i-1,k} M_{i+1,j},$$
$$e_j^{(i)} = M_{i,j}, \ \ c_j^{(i)} = M_{i-2,j} M_{i,j} - M_{i-2,j} M_{i-1,j}, \ \ d_{j,k}^{(i)} =$$
$$M_{i-2,j} M_{i,k} + M_{i-2,k} M_{i,j} - M_{i-2,j} M_{i-1,k} - M_{i-2,k} M_{i-1,j}.$$

For a monomial x_j^2, it does not appear in the output of $\mathrm{RF}(\bar{x})$ if and only if $a_j^{(i)} = 0$ for all $i \in S_1$ and $c_j^{(i)} = 0$ for all $i \in S_2$, which is equivalent to the system of equations

$$\begin{cases}
M_{0,j} M_{2,j} = 0,\\
M_{3,j} M_{5,j} = 0,\\
\vdots\\
M_{n-3,j} M_{n-1,j} = 0,\\
M_{0,j} M_{1,j} = 0,\\
M_{3,j} M_{4,j} = 0,\\
\vdots\\
M_{n-3,j} M_{n-2,j} = 0.
\end{cases} \tag{12}$$

Similarly, a monomial $x_j x_k$ with $j < k$ does not appear in the output if and only if $b_{j,k}^{(i)} = 0$ for all $i \in S_1$ and $d_{j,k}^{(i)} = 0$ for all $i \in S_2$, which is equivalent to the system of equations

$$\begin{cases}
M_{0,j} M_{2,k} + M_{0,k} M_{2,j} = 0,\\
M_{3,j} M_{5,k} + M_{3,k} M_{5,j} = 0,\\
\vdots\\
M_{n-3,j} M_{n-1,k} + M_{n-3,k} M_{n-1,j} = 0,\\
M_{0,j} M_{1,k} + M_{0,k} M_{1,j} = 0,\\
M_{3,j} M_{4,k} + M_{3,k} M_{4,j} = 0,\\
\vdots\\
M_{n-3,j} M_{n-2,k} + M_{n-3,k} M_{n-2,j} = 0.
\end{cases} \tag{13}$$

However, our inspection reveals that both Equation (12) and (13) admit no solutions under the matrix M used in YuS, implying all quadratic terms appear after a single round. For all linear terms, they are also propagated to the next round since the matrix M is full-rank. Hence, all the monomials of degree at most 2 appear in the output, and it is reasonable to believe that all monomials of degree at most 2^r will appear after r rounds of YuS.

Note that an additional linear layer LP is applied before the truncated output of the complete r-round instance, which

ensures a sufficient number of monomials. Let us consider the output of $\mathrm{TF} \circ \mathrm{LP} \circ \mathrm{RF}(\bar{x})$, where the truncation function TF is defined by $\mathrm{TF}(v_0, \ldots, v_{n-1}) = (v_{n/3}, \ldots, v_{n-1})$. A monomial x_j^2 does not appear in the output if and only if the following equations

$$\begin{cases} M_{t,1}M_{0,j}M_{2,j} = 0, \\ M_{t,4}M_{3,j}M_{5,j} = 0, \\ \vdots \\ M_{t,n-2}M_{n-3,j}M_{n-1,j} = 0, \\ M_{t,2}M_{0,j}M_{1,j} = 0, \\ M_{t,5}M_{3,j}M_{4,j} = 0, \\ \vdots \\ M_{t,n-1}M_{n-3,j}M_{n-2,j} = 0, \end{cases} \tag{14}$$

are satisfied for all $t \in S_3$, where $S_3 = [n/3, n-1]$. Similarly, a monomial $x_j x_k$ with $j < k$ does not appear in the output if and only if the following equations

$$\begin{cases} M_{t,1}\left(M_{0,j}M_{2,k} + M_{0,k}M_{2,j}\right) = 0, \\ M_{t,4}\left(M_{3,j}M_{5,k} + M_{3,k}M_{5,j}\right) = 0, \\ \vdots \\ M_{t,n-2}\left(M_{n-3,j}M_{n-1,k} + M_{n-3,k}M_{n-1,j}\right) = 0, \\ M_{t,2}\left(M_{0,j}M_{1,k} + M_{0,k}M_{1,j}\right) = 0, \\ M_{t,5}\left(M_{3,j}M_{4,k} + M_{3,k}M_{4,j}\right) = 0, \\ \vdots \\ M_{t,n-1}\left(M_{n-3,j}M_{n-2,k} + M_{n-3,k}M_{n-2,j}\right) = 0, \end{cases} \tag{15}$$

are satisfied for all $t \in S_3$. Then for a monomial x_j, it does not appear in the output if and only if the following equations

$$\begin{cases} M_{t,0}M_{0,j} = 0, \\ M_{t,1}M_{1,j} = 0, \\ \vdots \\ M_{t,n-1}M_{n-1,j} = 0, \end{cases} \tag{16}$$

are satisfied for all $t \in S_3$. By inspecting the matrix M used in YuS, we confirm that Equation (14)-(16) admit no solutions, thus all possible terms appear in the output of $\mathrm{TF}\circ\mathrm{LP}\circ\mathrm{RF}(\bar{x})$. Indeed, without the diffusion of the last linear layer, some monomials would be lost after the fixed truncation operation, which can be verified in our experiments.

To further evaluate the monomial count in YuS, we conducted an experimental analysis using Magma. Concretely, the results for 1-round and 2-round YuS indicate that the minimum number of monomials in output words is 696 and 91386, respectively, which closely match the theoretical results calculated by Equation (10). Due to memory limitations, there are no results for 3-round YuS, but we believe that this property also holds for more than two rounds.

APPENDIX C
BRANCH NUMBER OF MATRIX OVER $\mathbb{F}_p$ AND ITS COMPUTATION

Let $S(\bar{x})$ be the S-box of YuS, $\bar{a}, \bar{b}$ be the input and output masks of $S(\bar{x})$ respectively. Then according to Theorem 4, it holds

$$|\lambda_S(\bar{a},\bar{b})| = \begin{cases} p^3, & a_0 = b_0 \text{ and } b_i = a_i = 0 \text{ for } 1 \le i \le 2, \\ p^2, & (b_1,b_2) \neq (0,0) \text{ and } \mathrm{rank}\begin{pmatrix} -b_2 & b_1+b_2 \\ b_1-a_1 & b_2-a_2 \end{pmatrix} = 2, \\ 0, & \text{others}, \end{cases}$$

where $\lambda_S(\bar{a},\bar{b}) = \sum_{\bar{x}\in\mathbb{F}_p^3} \chi(\bar{b}\cdot S(\bar{x}) - \bar{a}\cdot\bar{x})$, and $\bar{a} = (a_0,a_1,a_2), \bar{b} = (b_0,b_1,b_2) \in \mathbb{F}_p^3$.

Note that if $|\lambda_S(\bar{a},\bar{b})| \le p^2$ then $(b_1,b_2) \neq (0,0)$. Furthermore, if $(a_1,a_2) \neq (0,0)$, then $|\lambda_S(\bar{a},\bar{b})| \le p^2$. Therefore, we call an S-box nontrivial linear active if $(a_1,a_2) \neq (0,0)$ or $(b_1,b_2) \neq (0,0)$. Suppose $(c_0,c_1,c_2) \mapsto (d_0,d_1,d_2)$ is the input and output mask of an S-box in one linear trail with nonzero probability of YuS. Then $|\lambda_S(\bar{c},\bar{d})| = p^3$ or $|\lambda_S(\bar{c},\bar{d})| = p^2$. If the corresponding S-box is nontrivial linear active, i.e., $(c_1,c_2) \neq (0,0)$ or $(d_1,d_2) \neq (0,0)$, then it holds $|\lambda_S(\bar{c},\bar{d})| = p^2$ since they are in a linear trail with nonzero probability. Then the linear branch number of the linear layer of YuS is defined as follows.

Definition 1: Let $\bar{v} = (v_0, \ldots, v_{35}) \in \mathbb{F}_p^{36}$ be a vector. Its linear weight in YuS is defined as $WL_{\mathsf{YuS}}(\bar{v}) = |\{i : 0 \le i \le 11, (v_{3i+1}, v_{3i+2}) \neq (0,0)\}|$. The linear branch number of a matrix in YuS is defined as

$$LBN_{\mathsf{YuS}}(M) = \min_{\bar{0}\neq\bar{v}\in\mathbb{F}_p^{36}} \{WL_{\mathsf{YuS}}(\bar{v}) + WL_{\mathsf{YuS}}(M\cdot\bar{v}^T)\}.$$

First, we have the following result concerning the upper bound of the linear branch number of a matrix in YuS.

Lemma 4: Let $M \in \mathbb{F}_p^{36\times 36}$ be a matrix of order 36. Then $LBN_{\mathsf{YuS}}(M) \le 7$.

Proof: Let $\bar{v} \in \mathbb{F}_p^{36}$ with $WL_{\mathsf{YuS}}(\bar{v}) = 7$. Without loss of generality, suppose $(v_{3i+1}, v_{3i+2}) \neq (0,0)$ for $0 \le i \le 6$. Then $(v_{3i+1}, v_{3i+2}) = (0,0)$ for $7 \le i \le 11$. Let

$$S_0 = \{0 \le j \le 35 : j \notin \{3i+1, 3i+2 : 7 \le i \le 11\}\},$$

and $S_1 = \{0 \le j \le 35 : j \in \{3i+1, 3i+2 : 0 \le i \le 11\}\}$. Then $|S_0| = 36 - 10 = 26 > |S_1| = 24$. For $\bar{u} \in \mathbb{F}_p^{36}$, denotes $\bar{u}_{S_j}$ be the vector $(u_i, i \in S_j)$, $j = 0, 1$, and M_{S_1,S_0} the submatrix of M with the rows index in S_1 and columns index in S_0. Note that $\mathrm{rank}(M_{S_1,S_0}) \le 24 < 26 = |S_0|$, then there exists $\bar{v} \in \mathbb{F}_p^{36}$, with $\bar{v}_{S_0} \neq \bar{0}$ and

$$(M\cdot\bar{v}^T)_{S_1} = M_{S_1,S_0}\cdot\bar{v}_{S_0} = \bar{0}.$$

This means there exist $\bar{v} \in \{M\cdot\bar{v}^T : v \in \mathbb{F}_p^{36} \text{ and } WL_{\mathsf{YuS}}(\bar{v}) = 7\}$, such that $(v_{3i+1}, v_{3i+2}) = (0,0)$ for all $0 \le i \le 11$, i.e., $LBN_{\mathsf{YuS}}(\bar{v}) = 0$. Therefore, it holds

$$\begin{aligned} LBN_{\mathsf{YuS}}(M) &= \min_{\bar{0}\neq\bar{v}\in\mathbb{F}_p^{36}} \{WL_{\mathsf{YuS}}(\bar{v}) + WL_{\mathsf{YuS}}(M\cdot\bar{v}^T)\} \\ &\le 7 + 0 = 7. \end{aligned}$$

Then we complete the proof. ∎

Given a matrix M, we can determine whether $LBN_{\mathsf{YuS}}(M) \ge d$ by Algorithm 1. As for the matrix used in YuS, it can check that $LBN_{\mathsf{YuS}}(M^T) \ge 6$ is true while $LBN_{\mathsf{YuS}}(M^T) \ge 7$ is false. Therefore, $LBN_{\mathsf{YuS}}(M^T) = 6$.

The differential branch number of the linear layer of YuS is similar to the previous. For a vector $\bar{v} =$

TABLE IX
SERVER-SIDE PERFORMANCE COMPARISON IN HELIB-2.3.0

Cipher	Parameters	p	N	Block	$\lceil \log q \rceil$	λ'	Off (s)	On (s)	Total (s)	Throughput (KiB/s)
80-bit										
Masta	(4,32,32)	4298506241	32768	32768	771	141.904	76.489	0.402	76.891	54.935
Masta	(5,16,16)	4298506241	32768	32768	943	112.343	38.492	0.239	38.731	54.529
HERA	(4,16,16)	4298506241	32768	32768	688	161.631	21.020	0.189	21.209	99.580
YuS	(5,36,24)	4298506241	16384	16384	446	120.415	8.349	0.092	8.441	187.660
128-bit										
Masta	(6,32,32)	4298506241	65536	65536	1116	203.543	330.347	1.151	331.497	25.484
HERA	(5,16,16)	4298506241	32768	32768	800	136.165	30.479	0.214	30.693	68.809
PASTA	(4,64,32)	4298506241	32768	32768	784	139.256	151.594	0.190	151.785	27.829
PASTA_{v2}	(4,64,32)	4298506241	32768	32768	784	139.256	92.677	0.216	92.894	45.471
YuS	(6,36,24)	4298506241	32768	32768	584	193.863	28.297	0.225	28.522	111.072

Parameters (r, ks, ps): r, ks, ps denote the round number, key size, and plain/cipher size.
N: degree of the cyclotomic polynomial of BGV in HElib.
Block: the number of blocks in plaintext stream.
$\lceil \log q \rceil$: the ciphertext modulus bits of BGV in HElib.
λ': the security level of BGV in HElib.
Throughput: Throughput $= \frac{\text{ps} \times \text{Block} \times \lceil \log p \rceil}{8192 \times \text{Total}}$

TABLE X
SERVER-SIDE PERFORMANCE COMPARISON IN SEAL-4.1.2 (BFV SECURITY LEVEL $\lambda = 128$ BIT)

Cipher	Parameters	p	N	Block	$\lceil \log q \rceil$	Off (s)	On (s)	Total (s)	Throughput (KiB/s)	Noise Budget(bit)
80-bit										
Masta	(4,32,32)	4298506241	32768	32768	881	267.890	0.126	268.016	15.760	398
Masta	(5,16,16)	4298506241	32768	32768	881	97.490	0.063	97.553	21.650	313
HERA	(4,16,16)	4298506241	32768	32768	881	60.608	0.064	60.671	34.810	352
YuS†	(5,36,24)	4298506241	16384	16384	424	11.358	0.019	11.376	139.236	52
YuS	(5,36,24)	4298506241	32768	32768	881	64.832	0.068	64.900	48.814	500
128-bit										
Masta	(6,32,32)	4298506241	32768	32768	881	375.671	0.132	375.803	11.240	224
HERA	(5,16,16)	4298506241	32768	32768	881	74.713	0.064	74.777	28.244	254
PASTA	(4,64,32)	4298506241	32768	32768	881	563.116	0.132	563.248	7.499	343
PASTA_{v2}	(4,64,32)	4298506241	32768	32768	881	259.340	0.128	259.467	16.280	369
YuS†	(6,36,24)	4298506241	16384	16384	424	13.319	0.018	13.338	118.761	4
YuS	(6,36,24)	4298506241	32768	32768	881	77.346	0.073	77.420	40.920	451

Parameters (r, ks, ps): r, ks, ps denote the round number, key size, and plain/cipher size.
N: degree of the cyclotomic polynomial of BFV in SEAL.
Block: the number of blocks in the plaintext stream.
$\lceil \log q \rceil$: the ciphertext modulus bits of BFV in SEAL.
Throughput: Throughput $= \frac{ps \times \text{Block} \times \lceil \log p \rceil}{8192 \times \text{Total}}$
Noise Budget: the final noise budget for the encrypted plaintext stream after BFV evaluation in SEAL.
YuS†: the BFV security level is $\lambda = 192$ bit for this set of parameters, with $\lceil \log q \rceil = 424$.

$(v_0, \ldots, v_{35})$, its weight is $W(\bar{v}) = |\{i : 0 \leq i \leq 11, (v_{3i}, v_{3i+1}, v_{3i+2}) \neq (0,0,0)\}|$. Then differential branch number of M is $\min_{\bar{0} \neq \bar{v} \in \mathbb{F}_p^{36}} \{W(\bar{v}) + W(M \cdot \bar{v}^T)\}$. Similarly, it can check that the differential branch number of the matrix in YuS is 10.

APPENDIX D
PERFORMANCE OF YuS WITH $p = 4298506241$(33-BIT)

In Table IX and X, all of our experiments were conducted on a Linux server with Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz, 252 GiB of memory, and 2 GiB of swap. We developed our source code in C++17 and compiled it with GNU C++ 11.4.0 (without the Intel® HEXL acceleration library).

ACKNOWLEDGMENT

The authors would like to thank the reviewers for the detailed and constructive comments, which have substantially improved the presentation of the paper.

REFERENCES

[1] L. Ducas and D. Micciancio, "FHEW: Bootstrapping homomorphic encryption in less than a second," in *EUROCRYPT 2015, Part I*, ser. LNCS, E. Oswald and M. Fischlin, Eds., vol. 9056. Springer, Heidelberg, Apr. 2015, pp. 617–640. I

[2] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: Fast fully homomorphic encryption over the torus," *Journal of Cryptology*, vol. 33, no. 1, pp. 34–91, Jan. 2020. I

[3] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *ITCS 2012*, S. Goldwasser, Ed. ACM, Jan. 2012, pp. 309–325. I

[4] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *CRYPTO 2012*, ser. LNCS, R. Safavi-Naini and R. Canetti, Eds., vol. 7417. Springer, Heidelberg, Aug. 2012, pp. 868–886. I

[5] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," Cryptology ePrint Archive, Report 2012/144, 2012, https://eprint.iacr.org/2012/144. I

[6] J. H. Cheon, A. Kim, M. Kim, and Y. S. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *ASIACRYPT 2017, Part I*, ser. LNCS, T. Takagi and T. Peyrin, Eds., vol. 10624. Springer, Heidelberg, Dec. 2017, pp. 409–437. I

[7] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, ser. CCSW '11. New York,

Algorithm 1 Computation of $LBN_{\mathsf{YuS}}(M)$

Input: A matrix $M \in \mathbb{F}_p^{36\times 36}$
Output: Whether $LBN_{\mathsf{YuS}}(M) \geq d$

$S_k \leftarrow \{(k_1,k_2) : 0 \leq k_1,k_2 \leq 11, k_1+k_2 \leq d-1\}$
for $(k_1,k_2) \in S_k$ **do**
  $n_1 := 12+2k_1, n_2 := 24-2k_2$
  if $n_1 > n_2$ **then**
    return False
  end if
  $S_{k_1} \leftarrow Subsets(\{i : 0 \leq i \leq 11\}, k_1)$
  $S_{k_2} \leftarrow Subsets(\{i : 0 \leq i \leq 11\}, k_2)$
  for $I \in S_{k_1}$ **do**
    $Ind_I \leftarrow [3i : 0 \leq i \leq 11]||[3i+1, 3i+2 : i \in I]$
    for $J \in S_{k_2}$ **do**
      $Ind_J \leftarrow [3i+1, 3i+2 : 0 \leq i \leq 11$ and $i \notin J]$
      if $\mathrm{rank}(M^{'} \leftarrow M_{Ind_J, Ind_I}) \neq n_1$ **then**
        return False
      end if
    end for
  end for
end for
return True

NY, USA: Association for Computing Machinery, 2011, pp. 113–124. [Online]. Available: https://doi.org/10.1145/2046660.2046682 I

[8] J. Cho, J. Ha, S. Kim, B. Lee, J. Lee, J. Lee, D. Moon, and H. Yoon, "Transciphering framework for approximate homomorphic encryption," in *ASIACRYPT 2021, Part III*, ser. LNCS, M. Tibouchi and H. Wang, Eds., vol. 13092. Springer, Heidelberg, Dec. 2021, pp. 640–669. I, VI-E, VII-A

[9] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," in *CRYPTO 2012*, ser. LNCS, R. Safavi-Naini and R. Canetti, Eds., vol. 7417. Springer, Heidelberg, Aug. 2012, pp. 850–867. I

[10] B. Wei, R. Wang, Z. Li, Q. Liu, and X. Lu, "Fregata: Faster homomorphic evaluation of aes via tfhe," in *Information Security*, E. Athanasopoulos and B. Mennink, Eds. Cham: Springer Nature Switzerland, 2023, pp. 392–412. I

[11] R. Wang, Y. Wen, Z. Li, X. Lu, B. Wei, K. Liu, and K. Wang, "Circuit bootstrapping: Faster and smaller," in *Advances in Cryptology – EUROCRYPT 2024*, M. Joye and G. Leander, Eds. Cham: Springer Nature Switzerland, 2024, pp. 342–372. I

[12] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner, "Ciphers for MPC and FHE," in *EUROCRYPT 2015, Part I*, ser. LNCS, E. Oswald and M. Fischlin, Eds., vol. 9056. Springer, Heidelberg, Apr. 2015, pp. 430–454. I

[13] A. Canteaut, S. Carpov, C. Fontaine, T. Lepoint, M. Naya-Plasencia, P. Paillier, and R. Sirdey, "Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression," in *FSE 2016*, ser. LNCS, T. Peyrin, Ed., vol. 9783. Springer, Heidelberg, Mar. 2016, pp. 313–333. I

[14] P. Méaux, A. Journault, F.-X. Standaert, and C. Carlet, "Towards stream ciphers for efficient FHE with low-noise ciphertexts," in *EUROCRYPT 2016, Part I*, ser. LNCS, M. Fischlin and J.-S. Coron, Eds., vol. 9665. Springer, Heidelberg, May 2016, pp. 311–343. I

[15] C. Dobraunig, M. Eichlseder, L. Grassi, V. Lallemand, G. Leander, E. List, F. Mendel, and C. Rechberger, "Rasta: A cipher with low ANDdepth and few ANDs per bit," in *CRYPTO 2018, Part I*, ser. LNCS, H. Shacham and A. Boldyreva, Eds., vol. 10991. Springer, Heidelberg, Aug. 2018, pp. 662–692. I

[16] P. Hebborn and G. Leander, "Dasta – alternative linear layer for Rasta," *IACR Trans. Symm. Cryptol.*, vol. 2020, no. 3, pp. 46–86, 2020. I

[17] C. Cid, J. P. Indrøy, and H. Raddum, "Fasta–a stream cipher for fast fhe evaluation," in *Cryptographers' Track at the RSA Conference*. Springer, 2022, pp. 451–483. I

[18] M. R. Albrecht, L. Grassi, C. Rechberger, A. Roy, and T. Tiessen, "MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity," in *ASIACRYPT 2016, Part I*, ser. LNCS, J. H. Cheon and T. Takagi, Eds., vol. 10031. Springer, Heidelberg, Dec. 2016, pp. 191–219. I

[19] M. R. Albrecht, L. Grassi, L. Perrin, S. Ramacher, C. Rechberger, D. Rotaru, A. Roy, and M. Schofnegger, "Feistel structures for MPC, and more," in *ESORICS 2019, Part II*, ser. LNCS, K. Sako, S. Schneider, and P. Y. A. Ryan, Eds., vol. 11736. Springer, Heidelberg, Sep. 2019, pp. 151–171. I

[20] M. R. Albrecht, C. Cid, L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, and M. Schofnegger, "Algebraic cryptanalysis of STARK-friendly designs: Application to MARVELlous and MiMC," in *ASIACRYPT 2019, Part III*, ser. LNCS, S. D. Galbraith and S. Moriai, Eds., vol. 11923. Springer, Heidelberg, Dec. 2019, pp. 371–397. I

[21] T. Ashur, M. Mahzoun, and D. Toprakhisar, "Chaghri-a fhe-friendly block cipher," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 139–150. I

[22] J. Ha, S. Kim, W. Choi, J. Lee, D. Moon, H. Yoon, and J. Cho, "Masta: An he-friendly cipher using modular arithmetic," *IEEE Access*, vol. 8, pp. 194 741–194 751, 2020. I

[23] C. Dobraunig, L. Grassi, L. Helminger, C. Rechberger, M. Schofnegger, and R. Walch, "Pasta: A case for hybrid homomorphic encryption," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2023, no. 3, pp. 30–73, 2023. I

[24] F. Liu, Y. Li, H. Chen, L. Jiao, M. Luo, and M. Wang, "Yux: Finite field multiplication based block ciphers for efficient fhe evaluation," *IEEE Transactions on Information Theory*, vol. 70, no. 5, pp. 3729–3749, 2024. I

[25] F. Schmid, C. Rechberger, L. Grassi, F. Liu, R. Walch, and Q. Wang, "Minimize the randomness in rasta-like designs: How far can we go?: Application to pasta," in *Selected Areas in Cryptography*. Springer, Aug. 2024. I

[26] J. Ha, S. Kim, B. Lee, J. Lee, and M. Son, "Rubato: Noisy ciphers for approximate homomorphic encryption," in *EUROCRYPT 2022, Part I*, ser. LNCS, O. Dunkelman and S. Dziembowski, Eds., vol. 13275. Springer, Heidelberg, May / Jun. 2022, pp. 581–610. I, VI-E

[27] L. Grassi, S. Onofri, M. Pedicini, and L. Sozzi, "Invertible quadratic non-linear layers for MPC-/FHE-/ZK-friendly schemes over $\mathbb{F}_p^n$," *IACR Transactions on Symmetric Cryptology*, vol. 2022, no. 2, p. 20–72, Sep. 2022. [Online]. Available: https://tches.iacr.org/index.php/ToSC/article/view/9849 I

[28] G. Giordani, L. Grassi, S. Onofri, and M. Pedicini, "Invertible quadratic non-linear functions over $\mathbb{F}_p^n$ via multiple local maps," in *Progress in Cryptology - AFRICACRYPT 2023*, N. El Mrabet, L. De Feo, and S. Duquesne, Eds. Cham: Springer Nature Switzerland, 2023, pp. 151–176. I

[29] S. Halevi and V. Shoup, "Design and implementation of HElib: a homomorphic encryption library," Cryptology ePrint Archive, Paper 2020/1481, 2020. [Online]. Available: https://eprint.iacr.org/2020/1481 II-A, VII

[30] "Microsoft SEAL (release 4.1.2)," https://github.com/Microsoft/SEAL, Jul. 2024, microsoft Research, Redmond, WA. II-A, VII

[31] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *J. ACM*, vol. 60, no. 6, pp. 43:1–43:35, 2013. II-A, II-A

[32] K. Nyberg, "Differentially uniform mappings for cryptography," in *Advances in Cryptology — EUROCRYPT '93*, T. Helleseth, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 55–64. II-B

[33] P. Kumar, R. Scholtz, and L. Welch, "Generalized bent functions and their properties," *Journal of Combinatorial Theory, Series A*, vol. 40, no. 1, pp. 90–107, 1985. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0097316585900494 II-B

[34] K. Nyberg, "Constructions of bent functions and difference sets," in *Advances in Cryptology — EUROCRYPT '90*, I. B. Damgård, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 151–160. II-B, 1

[35] T. Nagell, "Euler's criterion and legendre's symbol," *Introduction to Number Theory*, 1951. III-C

[36] M. Matsui, "New structure of block ciphers with provable security against differential and linear cryptanalysis," in *Fast Software Encryption*, D. Gollmann, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 205–218. VI-A

[37] T. Baignères, J. Stern, and S. Vaudenay, "Linear cryptanalysis of non binary ciphers," in *Selected Areas in Cryptography*, C. Adams, A. Miri, and M. Wiener, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 184–211. VI-A

[38] L. R. Knudsen, "Truncated and higher order differentials," in *Fast Software Encryption: Second International Workshop Leuven, Belgium,*

December 14–16, 1994 Proceedings 2. Springer, 1995, pp. 196–211. VI-B

[39] L. Knudsen and D. Wagner, "Integral cryptanalysis," in *Fast Software Encryption: 9th International Workshop, FSE 2002 Leuven, Belgium, February 4–6, 2002 Revised Papers 9*. Springer, 2002, pp. 112–127. VI-B

[40] I. Dinur and A. Shamir, "Cube attacks on tweakable black box polynomials," in *Advances in Cryptology-EUROCRYPT 2009: 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings 28*. Springer, 2009, pp. 278–299. VI-B

[41] J. C. Faugere, "A new efficient algorithm for computing gröbner bases without reduction to zero (f 5)," in *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, 2002, pp. 75–83. 1

[42] L. Bettale, J.-C. Faugere, and L. Perret, "Hybrid approach for solving multivariate systems over finite fields," *Journal of Mathematical Cryptology*, vol. 3, no. 3, pp. 177–197, 2009. 2

[43] R. Fröberg, "An inequality for hilbert series of graded algebras," *Mathematica Scandinavica*, vol. 56, no. 2, pp. 117–144, 1985. VI-C

[44] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, "Efficient algorithms for solving overdefined systems of multivariate polynomial equations," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2000, pp. 392–407. VI-D

[45] F. Liu, A. Kalam, S. Sarkar, and W. Meier, "Algebraic attack on fhe-friendly cipher hera using multiple collisions," *IACR Transactions on Symmetric Cryptology*, vol. 2024, no. 1, pp. 214–233, 2024. VI-E

[46] National Institute of Standards and Technology, "Sha-3 standard: Permutation-based hash and extendable-output functions," National Institute of Standards and Technology, Standard FIPS 202, 2015, available at: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf. VII-A

[47] J.-P. Bossuat, R. Cammarota, I. Chillotti, B. R. Curtis, W. Dai, H. Gong, E. Hales, D. Kim, B. Kumara, C. Lee, X. Lu, C. Maple, A. Pedrouzo-Ulloa, R. Player, Y. Polyakov, L. A. R. Lopez, Y. Song, and D. Yhee, "Security guidelines for implementing homomorphic encryption," Cryptology ePrint Archive, Paper 2024/463, 2024. [Online]. Available: https://eprint.iacr.org/2024/463 VII-A

[48] M. R. Albrecht, G. V. Bard, and W. Hart, "Algorithm 898: Efficient multiplication of dense matrices over GF(2)," *ACM Trans. Math. Softw.*, vol. 37, no. 1, pp. 9:1–9:14, 2010. VII-A

Yongqiang Li received the B.S. and M.S. degrees in mathematics from Beijing Normal University, Beijing, China, in 2005 and 2008, respectively, and the Ph.D. degree in information security from the Institute of Software, Chinese Academy of Sciences, China, in January 2012. He is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include symmetric cryptography and related areas.

Fangzhen Wang received the B.S. degree in mathematics from China University of Petroleum, Qingdao, China, in 2023. He is currently pursuing the M.S. degree at the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include symmetric cryptography and fully homomorphic encryption.

Xingwei Ren received the B.S. degree in mathematics from Hunan University of Science and Technology, Xiangtan, China, in 2022. He is currently pursuing the Ph.D. degree at the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include algebraic analysis and design of the arithmetization-oriented symmetric primitives.

Fen Liu received her B.S. and M.S. degrees from Xidian University in 2014 and 2017, respectively, and earned her Ph.D. degree from the Institute of Information Engineering, Chinese Academy of Sciences in 2024. She is currently a Lecturer at Foshan University, specializing in the design and analysis of block ciphers. Her research focuses on cryptography-based data security and privacy protection.

Xichao Hu received the B.S. degree from University of Electronic Science and Technology of China, Chengdu, China, in 2016, and the Ph.D. degree from the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, in 2021. He is currently a post-doctoral with the State Key Laboratory of Cryptology, Beijing. His research interests include symmetric cryptography and related areas.

Lin Jiao received the B.S. degree from Jilin University, Jilin, China, in 2010, and the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 2016. She is currently an Associate Professor with the State Key Laboratory of Cryptology, Beijing. Her research interests include symmetric cryptography and related areas.

Ya Han received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in 2012, and the Ph.D. degree from the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, in 2018. He is currently an Engineer of the Institute of Information Engineering, Chinese Academy of Sciences. He is engaged in applications related to cryptography and its integration with machine learning.