Optimal Threshold Implementations of all bijective 4-bit S-boxes and new Infinite Families

Enrico Piccione[1], Yevhen Perehuda[1], Gregor Leander[1], Patrick Felke[2]

Ruhr University Bochum[1] name.surname@rub.de
University of Applied Sciences Emden/Leer[2] name.surname@hs-emden-leer.de

Abstract. Threshold implementations (TIs) are now an established class of side-channel countermeasures. Their underlying idea is simple and compelling: under mild, practical assumptions, they provide protection against glitches and allow composability. Yet our understanding of one-stage TIs with a minimal number of shares remains limited, both for cryptographically relevant families—particularly power functions—and for concrete small functions.
In particular, until now, except for Feistel-constructions, no optimal TI had been found for families of mathematically well-structured and cryptographically non-trivial functions, especially power functions. In small dimensions, many functions - particularly cryptographically interesting ones — were not known to admit an optimal TI. This was especially true for the 4-bit S-boxes frequently used in practical lightweight ciphers. Optimal refers to share-optimal, direct one-stage, uniform, first-order TI.
Here, we introduce new families of TIs, including the first infinite families of cryptographically nontrivial power permutations. Moreover, every four-bit permutation of algebraic degree t admits a direct, uniform TI with t+1 shares and no fresh randomness.
Those results prove several previous conjectures and claims wrong. Since such permutations are widely used in symmetric ciphers, our constructions offer new implementation options.

1 Introduction

Threshold implementation (TI) is a masking technique designed to protect cryptographic implementations against side-channel attacks, even in the presence of glitches [26, 27]. An s-share TI replaces a function F by a shared function $\mathcal{F}$ satisfying three properties. *Correctness* ensures that recombining the output shares gives F evaluated on the recombined input shares. *Non-completeness* ensures that every output component is independent of at least one input share. *Uniformity* ensures that uniformly distributed input sharings are mapped to uniformly distributed output sharings. For permutations with the same number of input and output shares, uniformity is equivalent to $\mathcal{F}$ being a permutation. If F has algebraic degree t, correctness and non-completeness imply the lower bound $s \geq t + 1$ [26, 30]. We call a one-stage TI with $t + 1$ shares *share-optimal*, or simply *optimal*. Throughout this paper, we target first-order security without fresh randomness.

Constructing a correct and non-complete sharing is straightforward; uniformity is the principal obstacle. Existing approaches start from a direct sharing and search for correction terms, add fresh randomness to achieve uniformity, or decompose the target function into several stages [1,5,12]. The search for correction terms scales poorly and, in general, does not succeed. A major breakthrough by Piccione et al. showed that every bijective S-box of algebraic degree at most t admits a TI with $t+2$ shares [32]; this was subsequently extended to functions between finite Abelian groups [31]. Thus, only one share separates the general upper and lower bounds.

This one-share gap is genuine in the worst case. The three-bit inverse permutation, equivalently the cube function on $\mathbb{F}_{2^3}$ and denoted $\mathcal{Q}_3^3$ in the classification of small S-boxes, provably admits no one-stage three-share TI [5]. Consequently, the universal $t+2$-share construction is optimal as a universal result. At the same time, among bijective S-boxes this example was essentially the only proven obstruction, and there was no structural explanation of why it failed.

The available positive evidence nevertheless suggested a rather pessimistic picture. Before the present work, optimal TIs were known for only nine of the 301 nonlinear affine-equivalence classes of four-bit permutations: five of the six quadratic classes and four of the 295 cubic classes [5, 32]. In particular, the remaining quadratic class $\mathcal{Q}_{300}^4$ was claimed not to admit a one-stage three-share TI. Moreover, almost all previously known permutations with optimal TIs had poor cryptographic parameters. Based on this evidence, Piccione et al. conjectured that no nonlinear power permutation, and more generally no APN permutation, admits an optimal TI [32]. These conjectures captured a widely held intuition: optimal TIs might be exceptional and largely incompatible with strong cryptographic properties.

The theoretical picture was similarly sparse. The main general constructions were based on Feistel or Feistel-like permutations, for which the structure of the shared map makes uniformity accessible, and on methods that lift an S-box with a known TI to a slightly larger dimension [8, 23, 36]. Beyond these constructions, few cryptographically meaningful infinite families were known. Power permutations form a particularly conspicuous omission: they are among the most natural and best-understood objects in the theory of vectorial Boolean functions, yet no nonlinear power permutation was known to meet the optimal $t+1$-share bound.

The existence problem also has a direct implementation cost. If no optimal one-stage TI is known for an S-box, one can use the universal construction with an additional share, or decompose the S-box into lower-degree functions and evaluate them in several stages. The former increases the shared state and the associated logic, while the latter introduces registers and latency and may require remasking between stages. Techniques such as changing of the guards and NullFresh mitigate parts of this cost, but they do not remove the basic area–latency–randomness tradeoff [12, 28]. NullFresh bypasses the one-stage TI existence problem by using two shares over several register-separated stages. One

downside of NullFresh is that the number of stages grows with the size of the S-box and this could be inefficient for some cases.

Determining whether an S-box admits an optimal TI is therefore not merely a classification problem; it directly affects the design space of protected implementations. That said, such implementations, even if they exist, should not necessarily be expected to be preferable to existing ones.

Our contributions

We show that the preceding pessimistic picture is substantially too strong. Our results combine new structural constructions, new computational methods, and concrete hardware implementations. The theoretical constructions and computational results can be read independently. Readers primarily interested in concrete S-boxes and cipher implementations may proceed directly to Sections 5 and 6.

Structural constructions. We first introduce a secondary construction for triangular permutations in Section 3. It strictly generalizes the familiar Feistel setting and reduces the construction of an optimal TI for a larger permutation to suitable sharings of its constituent maps. This framework yields new families based on matrix algebras; in particular, nilpotent and some commutative matrix algebras lead to tractable sufficient conditions for optimal TIs.

We then give a second construction based on projective interpolation over a base field $\mathbb{F}_q$ in Section 4. The construction produces infinite families of highly nonlinear permutations with optimal TIs, and the existence of an optimal TI extends to every permutation in each resulting extended-affine equivalence class. Most notably, it yields the first optimal TIs for nonlinear power permutations, including cryptographically strong quadratic examples. This disproves the power-permutation conjecture of Piccione et al. while leaving the corresponding APN conjecture open. These examples show that share optimality is compatible with strong cryptographic properties.

The complete four-bit case. For concrete S-boxes, we develop a new and efficient search method for completing a correct, non-complete sharing to a uniform one. For cubic four-bit permutations, the vanishing fourth derivative allows us to parameterize the sharing by three correction functions and to search directly for a bijective shared map. Applying this method to all 295 cubic affine-equivalence classes produces a one-stage four-share TI for every class. We also find a one-stage three-share TI for the quadratic class $\mathcal{Q}^4_{300}$, refuting its previously claimed nonexistence. Together with the known results for the other quadratic classes, this proves that every four-bit permutation of algebraic degree t admits a one-stage TI with the optimal number $t+1$ of shares. The details appear in Section 5.

Protected cipher implementations. Finally, as a proof of concept, we use the four-bit result to construct a four-share implementations of PRINCE, MIDORI-64, PRESENT-80, and SKINNY-64-64 [2, 4, 6, 7] in Section 6 where the S-box is

implemented in one-stage. Our aim is not to provide fully optimized implementations that outperform all previous designs, but rather to demonstrate the practical applicability of our results and obtain initial implementation data points. The resulting implementations require no fresh random bits per clock cycle and have latencies of 13, 17, 32, and 33 cycles, respectively. These figures compare favorably with the fresh-randomness-free multi-stage implementations considered in our evaluation, but come at the expected cost of four-share combinational logic and additional randomness for the initial sharing. The results should therefore be viewed as promising initial data points for the corresponding area–latency tradeoff. In particular, a $(t+2)$-share construction may achieve comparable latency with a smaller area, and determining the best implementation strategy requires a more comprehensive optimization effort.

Taken together, our results demonstrate that optimal TIs are ubiquitous in four bits and occur in broad higher-dimensional families with meaningful cryptographic properties. This shifts the central question from whether cryptographically interesting optimal TIs exist at all to identifying the genuine structural obstructions to their existence.

2 Preliminaries

2.1 Vectorial Boolean Functions

Let $b \geq 2$ be an integer. If $d = \sum_{j\geq 0} d_j b^j$, with $0 \leq d_j < b$, we write $\mathrm{w}_b(d) = \sum_{j\geq 0} d_j$ for its base-b digit sum. In particular, $\mathrm{w}_2(d)$ is the binary weight of d, and $\mathrm{w}_b(0) = 0$.

Let q be a power of 2. Every function $F\colon \mathbb{F}_q^n \to \mathbb{F}_q^m$ has a unique representation

$$F(x_1, \ldots, x_n) = \sum_{\underline{u}\in\{0,\ldots,q-1\}^n} c_{\underline{u}} x_1^{u_1} \cdots x_n^{u_n}, \quad c_{\underline{u}} \in \mathbb{F}_q^m.$$

If $q = 2$, it is called the *Algebraic Normal Form (ANF)* of F, and if $n = m = 1$, it is called the *univariate representation* of F. The *algebraic degree* of F is the maximum of $\sum_{i=1}^n \mathrm{w}_2(u_i)$ over all $\underline{u} \in \{0, \ldots, q-1\}^n$ such that $c_{\underline{u}} \neq 0$. The *derivative* of F in the direction a is denoted by $D_aF(x) = F(x+a) + F(x)$ and the *k-th order derivative* of F in $\underline{a} = (a_1, \ldots, a_k)$ is denoted by $D_{\underline{a}}^{(k)}F = D_{a_1}D_{a_2}\cdots D_{a_k}F$. An alternative definition of the algebraic degree is the smallest nonnegative integer t such that for all $\underline{a} = (a_1, \ldots, a_{t+1})$ we have that $D_{\underline{a}}^{(t+1)}F = 0$.

For $b \in \mathbb{F}_q^m$ we define

$$D_aF^{-1}(b) := \{x \in \mathbb{F}_q^n : D_aF(x) = b\}, N_F(a,b) := |D_aF^{-1}(b)|.$$

The *differential uniformity* of F is

$$\mathcal{U}_F := \max\{N_F(a,b) : a \in \mathbb{F}_q^n \setminus \{0\}, b \in \mathbb{F}_q^m\}.$$

We call F *differentially k-uniform* if $\mathcal{U}_F = k$. In fields of characteristic 2 it is $D_aF(x+a) = D_aF(x)$, so $\mathcal{U}_F \geq 2$. A function attaining this minimum is called *almost perfect nonlinear* (APN). In this work we use the following notions of equivalence, which all preserve the differential uniformity.
Two functions F and G are *affine equivalent* if

$$G = B \circ F \circ A,$$

where A and B are affine permutations of the underlying vector spaces $\mathbb{F}_q^n$ and $\mathbb{F}_q^m$, respectively. They are *extended-affine equivalent*, or *EA-equivalent*, if there exists an $\mathbb{F}_2$-linear map L such that G and $F + L$ are affine equivalent.

2.2 Threshold Implementation

Threshold implementations evaluate a function on additive shares while preserving the sharing of its output. We recall the three defining properties: correctness, non-completeness, and uniformity [26, 27]. Throughout this paper, we consider first-order TIs without fresh randomness. Let X and Y be finite-dimensional vector spaces over $\mathbb{F}_2$. A *Boolean s-sharing* of $x \in X$ is a vector $\underline{x} = (x_1, \ldots, x_s) \in X^s$ such that $\sum_{i=1}^{s} x_i = x$. The set of such vectors is denoted by $\mathrm{Sh}_s(x)$. Let $F\colon X \to Y$ and let $\mathcal{F}\colon X^s \to Y^s$.

The function $\mathcal{F}$ is *correct* with respect to F if for all $x \in X$ and for all $\underline{x} \in \mathrm{Sh}_s(x)$ we have that $\mathcal{F}(\underline{x}) \in \mathrm{Sh}_s(F(x))$. An equivalent definition is that

$$F\left(\sum_{i=1}^{s} x_i\right) = \sum_{j=1}^{s} \mathcal{F}_j(\underline{x})$$

for all $\underline{x} = (x_1, \ldots, x_s) \in X^s$. In this case, we say that $\mathcal{F}$ is an *s-sharing* of F.
In general, non-completeness requires each output share to be independent of at least one input share. In this work we use the stronger convention that the function $\mathcal{F}$ is called *non-complete* if for all $i \in \{1, \ldots, s\}$, the function $\mathcal{F}_i$ is independent of the share x_i.

Assuming that $\mathcal{F}$ is an s-sharing of F, the function $\mathcal{F}$ is *uniform* if for all $x \in X$ and all $\underline{y} \in \mathrm{Sh}_s(F(x))$, we have that

$$|\mathrm{Sh}_s(x) \cap \mathcal{F}^{-1}(\underline{y})| = \frac{|X|^{s-1}}{|Y|^{s-1}}.$$

The function $\mathcal{F}$ is a *Threshold Implementation (TI)* of F with s shares if $\mathcal{F}$ is a uniform, non-complete s-sharing of F.

We use the following three facts throughout the paper:

1. If F is bijective and $\mathcal{F}$ is an s-sharing of F, then $\mathcal{F}$ is uniform if and only if $\mathcal{F}$ is bijective [26, 31].
2. The existence of a TI with s shares is preserved under affine equivalence [5].
3. Every permutation of algebraic degree at most t admits a threshold implementation with $t + 2$ shares [32].

We also call such a TI *direct* or *one-stage*. An alternative approach is to decompose the target permutation into simpler permutations,

$$F = G_1 \circ \cdots \circ G_\ell,$$

each admitting a uniform TI with the same number s of shares. The corresponding shared functions can then be evaluated successively: correctness is preserved under composition, and uniformity ensures that each stage receives uniformly distributed shares of its input. Non-completeness is required at each stage; the composed shared function need not itself be non-complete. In hardware, successive stages are separated by registers. Such an implementation may use fewer shares than a direct TI, at the cost of additional stages.
If F has algebraic degree $t \geq 1$, correctness and non-completeness imply that any direct TI requires at least $t+1$ shares [26,30]. We call a direct TI with $t+1$ shares *share-optimal*, or simply *optimal*.

2.3 Hardware implementation

Threshold implementations (TIs) protect computations by operating on shares of sensitive values. In the first-order probing model, every individual wire must have a distribution independent of the sensitive input. The glitch-extended model additionally exposes the inputs feeding a probed computation, traced back to registers or circuit inputs. Classical TIs address this through non-completeness: each output share omits at least one input share. Registers prevent glitches from propagating between stages, while uniformity ensures that uniform input sharings yield uniform output sharings suitable for subsequent computations.

Daemen's *changing of the guards* technique [12] establishes uniformity jointly across an S-box layer by using shares from neighbouring S-box computations to remask outputs. The individual S-box sharings need not be uniform, but their coordinated arrangement, together with an initial supply of additional shares, avoids fresh randomness for each S-box.

NullFresh, introduced by Shahmirzadi and Moradi [35], considers a broader class of masked circuits related to domain-oriented masking (DOM) [17]. Registered partial computations are chosen such that existing input shares provide the required masking. Security requires the joint distribution of all registered values exposed by a glitch-extended probe to be independent of the sensitive input. Hence, an output-share function may depend on all input shares if its staged evaluation satisfies this condition. This permits first-order, two-share implementations of nonlinear functions without contradicting the lower bound for classical TIs. Ovchinnikov et al. [28] subsequently generalised this approach to uniform, multistage masked computations for arbitrary functions whose output dimension does not exceed their input dimension.

The constructions studied here satisfy the classical TI conditions without additional random inputs. We focus particularly on direct, single-stage implementations with few shares, treating circuit area, latency, and randomness as separate efficiency measures.

3 A construction for triangular permutations

In this section, we give a construction for threshold implementations of triangular permutations with the optimal number of shares. The TIs preserve the triangular structure of the original permutation, allowing us to achieve uniformity by proving bijectivity in one group of shares while fixing the other. This construction includes the known construction for Feistel [8], and the more general construction for Feistel-like permutations [23]. Note that it was observed in [23] that the construction based on Shannon expansion presented in [36] is equivalent to a Feistel construction. In the next subsection, we then consider triangular functions whose first coordinate is affine in one group of variables. We reduce uniformity to the invertibility of a block matrix and we derive sufficient conditions in terms of the algebra generated by the coefficient matrices, treating both nilpotent and commutative algebras. The nilpotent case recovers recursive Feistel-like constructions. In the commutative case, this yields a criterion that depends on the vector space generated by the coefficient matrices.

Let A and B be vector spaces over $\mathbb{F}_2$. Let $G\colon A \times B \to A$ and $H\colon B \to B$, and define

$$F(a,b) = (G(a,b), H(b)). \tag{1}$$

We say that F is a *triangular function* if it is of the form (1), and a *triangular permutation* if it is also bijective. The latter holds if and only if H is bijective and, for every $b \in B$, the map $a \mapsto G(a,b)$ is bijective. Feistel-like functions are the special case $G(a,b) = G^{(1)}(a) + G^{(2)}(b)$; here, bijectivity is equivalent to that of $G^{(1)}$ and H.

We study when a triangular permutation F of algebraic degree t admits a threshold implementation with $t+1$ shares that preserves the triangular structure, meaning that we want a permutation of $A^{t+1} \times B^{t+1}$ of the form

$$\mathcal{F}(\underline{a}, \underline{b}) = (\mathcal{G}(\underline{a}, \underline{b}), \mathcal{H}(\underline{b})), \tag{2}$$

where $\mathcal{H}$ is a threshold implementation of H with $t+1$ shares and $\mathcal{G}$ is a non-complete $t+1$ sharing of G and for every fixed $\underline{b}$, the map $\underline{a} \mapsto \mathcal{G}(\underline{a}, \underline{b})$ must be bijective. The case in which H is the identity function provides a construction for the general case. Let C be a finite-dimensional vector space over $\mathbb{F}_2$ and let $\lambda\colon B \to C$ be a surjective linear map such that $G(a,b) = G^{(0)}(a, \lambda(b))$ for some $G^{(0)}\colon A \times C \to A$. Define

$$F^{(0)}(a,c) = (G^{(0)}(a,c), c). \tag{3}$$

Suppose that $F^{(0)}$ admits a triangular threshold implementation

$$\mathcal{F}^{(0)}(\underline{a}, \underline{c}) = (\mathcal{G}^{(0)}(\underline{a}, \underline{c}), \mathcal{I}(\underline{c})), \tag{4}$$

where $\mathcal{I}$ is a threshold implementation of the identity function with $t+1$ shares and $\mathcal{G}^{(0)}$ is a non-complete $t+1$ sharing of $G^{(0)}$ such that for all $\underline{c} \in C^{t+1}$, the function $\underline{a} \in A^{t+1} \mapsto \mathcal{G}^{(0)}(\underline{a}, \underline{c})$ must be bijective. Let $\Lambda(\underline{b}) = (\lambda(b_1), \ldots, \lambda(b_{t+1}))$. Then $\mathcal{G}(\underline{a}, \underline{b}) = \mathcal{G}^{(0)}(\underline{a}, \Lambda(\underline{b}))$.

The following theorem summarises this construction.

Theorem 1. *Let $\lambda\colon B \to C$ be balanced and linear. Suppose that $H\colon B \to B$ is a permutation admitting a threshold implementation with $t+1$ shares, and that $F^{(0)}$ in (3) is a permutation admitting a triangular threshold implementation with $t+1$ shares of the form (4). Let $G(a,b) = G^{(0)}(a,\lambda(b))$. Then the permutation F as in (1) admits a triangular threshold implementation with $t+1$ shares.*

Proof. Use the map in (2), with $\mathcal{G}$ defined above and $\mathcal{H}$ a threshold implementation of H. Correctness follows from the linearity of λ and the correctness of $\mathcal{G}^{(0)}$. Since Λ acts separately on each share, it preserves non-completeness. Finally, $\mathcal{H}$ is bijective and, for every fixed $\underline{b}$, the map $\underline{a} \mapsto \mathcal{G}^{(0)}(\underline{a}, \Lambda(\underline{b}))$ is bijective. Hence $\mathcal{F}$ is bijective and therefore uniform. □

Remark 1. Suppose that $G^{(0)}(a,c) = G^{(1)}(a,c) + G^{(2)}(c)$ and that $\mathcal{G}^{(1)}$ is a correct, non-complete sharing for which $\underline{a} \mapsto \mathcal{G}^{(1)}(\underline{a},\underline{c})$ is bijective for every fixed $\underline{c}$. Adding any correct, non-complete sharing of $G^{(2)}$ preserves this bijectivity: for fixed $\underline{c}$, it only translates the output.

We remark that the $t+2$ TI construction in [31, 32] can be adapted for triangular permutations.

Proposition 1. *If $F^{(0)}$ in (3) is a permutation with algebraic degree t, then $F^{(0)}$ admits a triangular threshold implementation with $t+2$ shares.*

Proof. See Appendix A.1. □

3.1 Triangular permutations from matrix algebras

Here we consider a particular case of functions $F^{(0)}$ as in (3) where for every $c \in C$, the map $a \mapsto G^{(0)}(a,c)$ is affine. We show that this problem can be investigated using matrix algebra theory.

Let C be a vector space over $\mathbb{F}_2$, let $G\colon C \to \mathbb{F}_2^n$ and $L\colon C \to \mathrm{M}_{n\times n}(\mathbb{F}_2)$, and define

$$F(x,c) = (x + L(c)x, c). \tag{5}$$

This map is a permutation if and only if $I_n + L(c)$ is invertible for every $c \in C$. Recall that by Remark 1, if we find a threshold implementation with $t+1$ shares of F that is a triangular permutation, then we can construct a threshold implementation with $t+1$ shares of $F'(x,c) = (x + L(c)x + G(c), c)$ for every $G\colon C \to \mathbb{F}_2^n$ with algebraic degree at most t.

Let t be the algebraic degree of F, let $s = t+1$ and let $H(x,c) = L(c)x$. Let $\mathcal{I}(x_1,\dots,x_{t+1}) = (x_2, x_3, \dots, x_{t+1}, x_1)$ and let $\mathcal{H}$ be a non-complete $t+1$ sharing of H of the form $\mathcal{H}(\underline{x},\underline{c}) = \mathbf{L}(\underline{c})\mathcal{I}(\underline{x})$, where $\mathbf{L}\colon C^s \to \mathrm{M}_{s\times s}(\mathbb{V})$ and $\mathbb{V}$ is the vector space generated by $\mathrm{Im}\,(L)$. To construct $\mathbf{L} = (L_{i,j})_{1\le i,j\le s}$, note that the algebraic degree of L is less than or equal to $t-1 = s-2$ and write

$$L\left(\sum_{j=1}^{s} c_j\right) = \sum_{\substack{S\subseteq\{1,\dots,s\}\\ |S|\le s-2}} T_S, \qquad T_S = \sum_{I\subseteq S} L\left(\sum_{i\in I} c_i\right),$$

with $T_\varnothing = L(0)$. Each T_S takes values in $\mathcal{V}$ and depends only on the shares indexed by S. Thus every term $T_S x_j$ omits at least one share index i and it can be assigned to $\mathcal{H}_i$.

In the following, we write $L_{i,j} = L_{i,j}(\underline{c})$ and $c = \sum_{i=1}^{s} c_i$. Let $\underline{z} = \mathcal{I}(\underline{x})$, and since $\mathcal{H}_i = \sum_{j=1}^{s} L_{i,j} z_j$ we must have, by non-completeness, that $L_{i,j}$ does not depend on c_i and that

$$L_{1,s} = L_{2,1} = \cdots = L_{s,s-1} = 0. \tag{6}$$

Then correctness of $\mathcal{H}$ gives

$$\sum_{i=1}^{s} L_{i,1} = \cdots = \sum_{i=1}^{s} L_{i,s} = L(c). \tag{7}$$

We want to study when

$$\mathcal{F}(\underline{x}, \underline{c}) = (\mathcal{I}(\underline{x}) + \mathbf{L}(\underline{c})\mathcal{I}(\underline{x}), \mathcal{I}(\underline{c})) \tag{8}$$

is a threshold implementation of F. To prove uniformity, we must show that the matrix

$$\begin{bmatrix} I_n + L_{1,1} & L_{1,2} & \cdots & L_{1,s} \\ L_{2,1} & I_n + L_{2,2} & \cdots & L_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ L_{s,1} & L_{s,2} & \cdots & I_n + L_{s,s} \end{bmatrix} = I_{ns} + \mathbf{L}, \tag{9}$$

is invertible for every $\underline{c}$. Our approach to solving this problem is to study the matrix algebra generated by the vector space $\mathbb{V}$.

Remark 2. A matrix algebra over $\mathbb{F}_2$ is an $\mathbb{F}_2$-subspace of $\mathrm{M}_{n\times n}(\mathbb{F}_2)$ closed under multiplication (we do not require it to contain I_n). We call $\mathbb{A}$ *nilpotent* if $\mathbb{A}_k = \{0\}$ for some positive integer k, where $\mathbb{A}_k$ is the span of all products of k elements of $\mathbb{A}$. For finite-dimensional matrix algebras, this is equivalent to every element being nilpotent. The algebra is *commutative* if $MN = NM$ for all $M, N \in \mathbb{A}$. We are interested in matrices M for which $I_n + M$ is invertible, or equivalently, for which 1 is not an eigenvalue. Every nilpotent matrix has this property. For further details, see [15].

Theorem 2. *Let F be as in* (5) *of algebraic degree t, and let $\mathbb{A}$ be the algebra generated by* $\mathrm{Im}\,(L)$. *If $\mathbb{A}$ is nilpotent, then F is a permutation admitting a triangular threshold implementation with $t+1$ shares.*

Proof. Choose k such that $\mathbb{A}_k = \{0\}$. Since $L(c) \in \mathbb{A}$, the matrix $L(c)$ is nilpotent for every c, so $I_n + L(c)$ is invertible and F is a permutation. Every block of $\mathbf{L}^k$ is a sum of products of k elements of $\mathbb{A}$, and hence $\mathbf{L}^k = 0$. Therefore, $I_{ns} + \mathbf{L}$ is invertible. □

Remark 3. By Levitzki's theorem [22] (see also [34, Theorem 1.3.1]), a nilpotent matrix algebra is simultaneously strictly upper triangularisable. Thus there is an invertible matrix P for which $P^{-1}L(c)P$ is strictly upper triangular for every c. Then F in (5) is a recursive Feistel-like permutation and we know that it admits a threshold implementation with $t+1$ shares by [23].

An upper triangular matrix M over $\mathbb{F}_2$ has all its eigenvalues in $\mathbb{F}_2$, meaning that if $I_n + M$ is invertible, all diagonal entries of M must be zero. Thus simultaneous triangularisation over $\mathbb{F}_2$ reduces to the nilpotent case. We next consider commutative algebras for which we can establish a simple criterion by using the fact that the determinant of (9) can be computed block-wise.

Theorem 3. *Let F be as in (5) of algebraic degree $t \geq 2$. Let $\mathbb{A}$ be the algebra generated by $\mathrm{Im}\,(L)$, let $\mathbb{V}$ be the vector space generated by $\mathrm{Im}\,(L)$ and let $\mathbb{V}_{2,t}$ be the vector space generated by*

$$\left\{ \prod_{i=1}^{r} L_i \colon 2 \leq r \leq t,\ L_i \in \mathbb{V} \right\}.$$

If $\mathbb{A}$ is commutative and $I_n + M$ is invertible for all $M \in \mathbb{V}_{2,t}$, then F is a permutation admitting a triangular threshold implementation with $t+1$ shares.

Proof. Since $L(c)^2 \in \mathbb{V}_{2,t}$ for every $c \in C$ and $(I_n + L(c))^2 = I_n + L(c)^2$, we have that $I_n + L(c)$ is invertible, and that F is a permutation. Let $\mathbf{L}$ be such that it satisfies the additional identity

$$\sum_{i=1}^{s} L_{i,i} = L\left(\sum_{i=1}^{s} c_i \right). \tag{10}$$

It is easy to see that it is always possible to construct such an $\mathbf{L}$ for which the sharing $\mathcal{F}$ is correct and non-complete. We claim that $\mathcal{F}$ is bijective. Fix $\underline{c}$ and set $c = \sum_{i=1}^{s} c_i$. It remains to prove that $I_{ns} + \mathbf{L}$ is invertible. Replace the last input coordinate z_s by $\sum_{i=1}^{s} z_i$ and the last output coordinate by the sum of all output coordinates. We obtain the matrix

$$\begin{bmatrix} I_n + \widetilde{L}_{1,1} & \widetilde{L}_{1,2} & \cdots & \widetilde{L}_{1,t} & * \\ \widetilde{L}_{2,1} & I_n + \widetilde{L}_{2,2} & \cdots & \widetilde{L}_{2,t} & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \widetilde{L}_{t,1} & \widetilde{L}_{t,2} & \cdots & I_n + \widetilde{L}_{t,t} & * \\ 0 & 0 & \cdots & 0 & I_n + L(c) \end{bmatrix}, \tag{11}$$

where $\widetilde{L}_{i,j} = L_{i,j} + L_{i,s}$ for $1 \leq i, j \leq t$. Let $\widetilde{\mathbf{L}} = (\widetilde{L}_{i,j})_{1 \leq i,j \leq t}$. The block determinant of $I_{nt} + \widetilde{\mathbf{L}}$ is equal to $I_n + \sum_{k=1}^{t} \Lambda_k$ where Λ_k is the sum of the $k \times k$ principal (block) minors of $\widetilde{\mathbf{L}}$. By (7) and (10), we have that Λ_1 is equal to zero. For $2 \leq k \leq t$, every term in Λ_k is a product of k elements of $\mathbb{V}$. □

Remark 4. Restricting the condition in Theorem 3 to $\mathbb{V}_{2,t}$, rather than imposing it on all of $\mathbb{A}$, is essential for obtaining non-nilpotent examples. Indeed, if $\mathbb{A}$ is not nilpotent, it contains a non-nilpotent matrix M. Since the sequence of powers of M is eventually periodic over the finite field, some positive power $E = M^r$ is a nonzero idempotent. Thus $E^2 = E \neq 0$, and $I_n + E$ is singular because it vanishes on the nonzero subspace $\mathrm{Im}\,(E)$.

Example 1. Suppose that $\mathbb{A}$ is isomorphic to $\mathbb{F}_{2^n} = \mathbb{F}_2(\alpha)$. Let r be an integer with $1 \leq r < n/t$, and suppose that the vector space $\mathbb{V}$ generated by the image of L satisfies $\mathbb{V} \subseteq \langle \alpha, \alpha^2, \ldots, \alpha^r \rangle$. Then $\mathbb{V}_{2,t} \subseteq \langle \alpha^2, \ldots, \alpha^{tr} \rangle$. Since $tr < n$, this space does not contain 1. Under the field identification, $I_n + M$ is therefore invertible for every $M \in \mathbb{V}_{2,t}$, so Theorem 3 applies.

Example 2. Let us consider the case where $\mathrm{Im}\,(L) = \{0, M\}$. We claim that $I_n + N$ is invertible for every $N \in \mathbb{V}_{2,t}$ if and only if every irreducible factor $f(z)$ of the minimal polynomial $\mu_M(z)$, other than z, either has degree strictly greater than t or has degree t and coefficient of z equal to 1. For $p(z) = 1 + \sum_{j=2}^{t} a_j z^j$, the matrix $p(M)$ is invertible if and only if $\gcd(p, \mu_M) = 1$. Let $f(z) = 1 + b_1 z + \cdots + b_{d-1} z^{d-1} + z^d$ be an irreducible factor of μ_M. If $d < t$, then $p(z) = (1 + b_1 z) f(z)$ has constant term 1, coefficient of z equal to 0, and degree at most $d + 1 \leq t$. Since f divides p, the matrix $p(M)$ is singular and we have a contradiction. If $d = t$, the only nonzero multiple of f of degree at most t is f itself. Hence f divides p if and only if $b_1 = 0$. Finally, if $d > t$, then f cannot divide any nonzero polynomial of degree at most t.

4 Family of highly nonlinear permutations from projective interpolation

In this section, we present a construction of threshold implementations with the optimal number of shares based on interpolation over a base field $\mathbb{F}_q$. We first present a general sufficient criterion, then specialise it by deriving a sufficient condition expressed in terms of the exponents in the univariate representation. Remarkably, we can prove that if a permutation satisfies the construction's hypothesis, then all permutations in its extended affine equivalence class admit a threshold implementation with $t + 1$ shares.

Let F be a permutation of $\mathbb{F}_q^m$ of algebraic degree t such that $q \geq t + 1$. Let $\mu, \lambda_1, \ldots, \lambda_t$ be distinct elements of $\mathbb{F}_q$ and let

$$\alpha_i = \prod_{\substack{1 \leq j \leq t \\ j \neq i}} \frac{\mu + \lambda_j}{\lambda_i + \lambda_j} \tag{12}$$

be the i-th Lagrange interpolation coefficient obtained by evaluating the corresponding Lagrange interpolation polynomial at μ. The sufficient condition that allows us to construct a threshold implementation of F with $t + 1$ shares is the existence of a map $H \colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ such that, for all $w, v \in \mathbb{F}_q^m$, we have that

$$F(w + \mu v) = \sum_{i=1}^{t} \alpha_i F(w + \lambda_i v) + H(v). \tag{13}$$

We show that the $t+1$ terms on the right side of Equation (13) are exactly what we need to construct a $t + 1$ TI of F. Note that the term $H(v)$ enables us to

make use of Lagrange interpolation with t points instead of $t+1$ (which would have resulted in the more restrictive condition "$q \geq t+2$"). This is also known as projective interpolation.

Theorem 4. *Let F be a permutation of $\mathbb{F}_q^m$ of algebraic degree t such that $q \geq t+1$. Let $\mu, \lambda_1, \ldots, \lambda_t$ be distinct elements of $\mathbb{F}_q$. If there is a map $H\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ such that Equation (13) holds for all $w, v \in \mathbb{F}_q^m$, then F admits a threshold implementation with $t+1$ shares.*

Proof. *Let $x_1, \ldots, x_{t+1} \in \mathbb{F}_q^m$ and let*

$$w = \sum_{i=1}^{t} \frac{\lambda_i}{\mu + \lambda_i} x_i + x_{t+1} \qquad v = \sum_{i=1}^{t} \frac{1}{\mu + \lambda_i} x_i. \tag{14}$$

Let $u_0 = u_1 = 0$, $u_i = x_{i-1}$ for $1 < i < t+1$ and $u_{t+1} = 0$. Let

$$\begin{aligned} y_i &= \alpha_i F(w + \lambda_i v) + u_i + u_{i-1} \\ y_{t+1} &= H(v) + u_{t+1} + u_t. \end{aligned}$$

We claim that the function $\mathcal{F}$ that maps $(x_1, \ldots, x_{t+1})$ to $(y_1, \ldots, y_{t+1})$ is a threshold implementation of F.

By (13), we have that $\sum_{i=1}^{t+1} y_i = F(w + \mu v)$. So if we prove that $w + \mu v$ is equal to $\sum_{i=1}^{t+1} x_i$, then $\mathcal{F}$ is correct. The coefficient of x_{t+1} in $w + \mu v$ is 1, and the coefficient of x_i, for $1 \leq i \leq t$, is

$$\frac{\lambda_i}{\mu + \lambda_i} + \frac{\mu}{\mu + \lambda_i} = 1.$$

Let us prove the non-completeness of $\mathcal{F}$. For $1 \leq i \leq t$, neither u_i nor u_{i-1} depends on x_i, and the coefficient of x_i in $w + \lambda_i v$ is

$$\frac{\lambda_i}{\mu + \lambda_i} + \frac{\lambda_i}{\mu + \lambda_i} = 0.$$

The last output share is independent of x_{t+1}. To conclude, let us prove the uniformity of $\mathcal{F}$. We show that any preimage $(x_1, \ldots, x_{t+1})$ is uniquely determined by $(y_1, \ldots, y_{t+1})$. Let $\beta_i = \alpha_i F(w + \lambda_i v)$ and let $\sigma = F^{-1}\left(\sum_{i=1}^{t+1} y_i\right)$. Since $\sigma = w + \mu v$ and $y_1 = \alpha_1 F(w + \lambda_1 v)$, we obtain $w + \lambda_1 v = F^{-1}(\alpha_1^{-1} y_1)$ and since $\mu \neq \lambda_1$, we can write w and v by using $y_1, \ldots, y_{t+1}$. Therefore, we can write each β_i by using $y_1, \ldots, y_{t+1}$. Thus we can write $x_1 = \beta_2 + y_2$ and for $2 \leq j \leq t-1$ we have

$$x_j = \beta_{j+1} + y_{j+1} + x_{j-1}.$$

We are left with determining x_t, since x_{t+1} can then be determined trivially. Since x_t is equal to

$$(\mu + \lambda_t)\left(v + \sum_{i=1}^{t-1} \frac{1}{\mu + \lambda_i} x_i\right),$$

we can write x_t in terms of $y_1, \ldots, y_{t+1}$. Since $\sum_{i=1}^{t+1} x_i = \sigma$, we can express x_{t+1} in terms of $y_1, \ldots, y_{t+1}$. We conclude that $\mathcal{F}$ is bijective and therefore $\mathcal{F}$ is uniform. □

Corollary 1. *Under the hypotheses of Theorem 4, every permutation in the extended affine equivalence class of F admits a threshold implementation with $t+1$ shares.*

Proof. Equation (13) *holds for any linear function L because $L(w+\mu v)+\sum_{i=1}^{t} \alpha_i L(w+\lambda_i v)$ is equal to $L(\mu v) + \sum_{i=1}^{t} \alpha_i L(\lambda_i v)$ since $\sum_{i=1}^{t} \alpha_i = 1$.* □

Corollary 2. *Under the hypotheses of Theorem 4, if F is a triangular permutation, then F admits a threshold implementation with $t+1$ shares that is a triangular permutation.*

Proof. Without loss of generality, it is enough to prove it for $F(x,c) = (G(x,c), c)$. Then G satisfies Equation (13) *for some H. Let*

$$w = \sum_{i=1}^{t} \frac{\lambda_i}{\mu+\lambda_i}(x_i, c_i) + (x_{t+1}, c_{t+1}), \qquad v = \sum_{i=1}^{t} \frac{1}{\mu+\lambda_i}(x_i, c_i),$$

let $u_0 = u_1 = 0$, $u_i = (x_{i-1}, c_{i-1})$ for $1 < i < t+1$ and $u_{t+1} = 0$. Let

$$\begin{aligned} y_i &= \alpha_i G(w + \lambda_i v) + u_i + u_{i-1} \\ y_{t+1} &= H(v) + u_{t+1} + u_t. \end{aligned}$$

Then the map $\mathcal{G}$ that maps $(\underline{x}, \underline{c})$ to $\underline{y}$ is a correct and non-complete $t+1$ sharing of G. By following the same proof as Theorem 4, for all $\underline{c}$ the map $\underline{x} \mapsto \mathcal{G}(\underline{x}, \underline{c})$ is a permutation. □

4.1 A criterion based on the univariate representation

From the univariate representation of F, we can determine if F satisfies Equation (13). Let d be a positive integer and write $k \preceq d$ if each binary digit of k is at most the corresponding digit of d. Then we have that

$$(w+\mu v)^d + \sum_{i=1}^{t} \alpha_i (w+\lambda_i v)^d = \sum_{k \preceq d} w^{d-k} v^k \left(\mu^k + \sum_{i=1}^{t} \alpha_i \lambda_i^k \right).$$

Therefore, this expression is independent of w if and only if $\mu^k + \sum_{i=1}^{t} \alpha_i \lambda_i^k = 0$ for every $k \preceq d$ with $k \neq d$. The coefficient for $k = d$ gives the term depending on v which is included in H. A problem with this characterization is that it could be that different choices of $\mu, \lambda_1, \ldots, \lambda_t$ work for different exponents. In the following, we show some sufficient conditions on the exponents.

Lemma 1. *Let m and t be positive integers and let q be a power of 2 such that $q \geq t+1$. Then there exist distinct $\mu, \lambda_1, \ldots, \lambda_t \in \mathbb{F}_q$ such that for every positive integer $d < q^m$ such that either $\mathrm{w}_q(d) \leq t$ or $\mathrm{w}_2(d) \leq \mathrm{w}_2(t)$, there exists a map $H\colon \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ such that, for all $w, v \in \mathbb{F}_{q^m}$, we have that*

$$(w + \mu v)^d = \sum_{i=1}^{t} \alpha_i (w + \lambda_i v)^d + H(v).$$

Proof. Let us prove the case $\mathrm{w}_q(d) \leq t$. Write $d = \sum_{j=0}^{m-1} \epsilon_j q^j$. For $z \in \mathbb{F}_q$, we have

$$p(z) = (w + zv)^d = \prod_{i=0}^{m-1} (w^{q^i} + z v^{q^i})^{\epsilon_i} = \sum_{j=0}^{t} p_j z^j$$

where p_t depends only on v. Let $R(z) = \prod_{i=1}^{t}(z + \lambda_i)$. Then the polynomial $p(z) + p_t R(z)$ has degree at most $t-1$ and therefore it can be interpolated using t points. So we have that $p(\mu) + p_t R(\mu) = \sum_{i=1}^{t} \alpha_i p(\lambda_i)$.

Let us prove the case $\mathrm{w}_2(d) \leq \mathrm{w}_2(t)$. Let

$$\ell(z) = \prod_{b \in \mathbb{F}_q \setminus \{\mu, \lambda_1, \ldots, \lambda_t\}} (z + b), \tag{15}$$

then $\alpha_i = \frac{\ell(\lambda_i)}{\ell(\mu)}$. So we have that $(w + \mu v)^d + \sum_{i=1}^{t} \alpha_i (w + \lambda_i v)^d$ is equal to

$$\frac{1}{\ell(\mu)} \sum_{z \in \mathbb{F}_q} \ell(z)(w + zv)^d = \frac{1}{\ell(\mu)} \sum_{k \preceq d} w^{d-k} v^k \sum_{z \in \mathbb{F}_q} \ell(z) z^k.$$

We claim that there exist distinct elements $\mu, \lambda_1, \ldots, \lambda_t \in \mathbb{F}_q$ for which ℓ has algebraic degree at most $\mathrm{w}_2(q-1) - \mathrm{w}_2(t)$. This implies that, for $k \neq d$, $\ell(z) z^k$ has algebraic degree at most $(\mathrm{w}_2(q-1) - \mathrm{w}_2(t)) + (\mathrm{w}_2(t) - 1) = \mathrm{w}_2(q-1) - 1$ and therefore $\sum_{z \in \mathbb{F}_q} \ell(z) z^k = 0$. This concludes the proof by showing that all the terms in w of $\frac{1}{\ell(\mu)} \sum_{k \preceq d} w^{d-k} v^k \sum_{z \in \mathbb{F}_q} \ell(z) z^k$ vanish. Let us prove that we can choose ℓ with algebraic degree $\mathrm{w}_2(q-1) - \mathrm{w}_2(t)$. Write $q - 1 - t = \sum_{i=1}^{r} 2^{\epsilon_i}$ with $\epsilon_1 < \cdots < \epsilon_r$. Let $A_1, \ldots, A_r$ be a list of disjoint affine subspaces of $\mathbb{F}_q$ with respective $\mathbb{F}_2$-dimensions $\epsilon_1, \ldots, \epsilon_r$. Let $A = \cup_{i=1}^{r} A_i$ and choose $\mu, \lambda_1, \ldots, \lambda_t$ to be the elements of $\mathbb{F}_q \setminus A$. Such disjoint affine subspaces exist: choose a binary basis of $\mathbb{F}_q$ and partition an initial segment of $q-1-t$ binary vectors into aligned blocks whose sizes are the powers of two in the binary expansion of $q-1-t$. For each $A_i = a_i + U_i$, the subspace polynomial $\prod_{u \in U_i}(z + u)$ is linearized. Hence $P_i(z) = \prod_{a \in A_i}(z + a)$ defines an $\mathbb{F}_2$-affine map. Thus $\ell(z) = \prod_{i=1}^{r} P_i(z)$ has algebraic degree at most $r = \mathrm{w}_2(q-1-t) = \mathrm{w}_2(q-1) - \mathrm{w}_2(t)$. □

Theorem 5. *Let F be a permutation of $\mathbb{F}_q^m$ of algebraic degree t such that $q \geq t+1$. Suppose that the univariate representation of F is of the form*

$$F(x) = \sum_{d \in J_t} c_d x^d$$

where

$$J_t = \{0 \le d < q^m \colon \mathrm{w}_q(d) \le t \text{ or } \mathrm{w}_2(d) \le \mathrm{w}_2(t)\}.$$

Then F admits a threshold implementation with $t+1$ shares.

Corollary 3. *Let q be a power of 2 such that $q \ge 4$. Then every quadratic permutation F of $\mathbb{F}_{q^m}$ that is EA-equivalent to a quadratic function of the form*

$$\sum_{0 \le i < j \le m} c_{i,j} x^{q^i + q^j} \tag{16}$$

admits a threshold implementation with 3 shares. In particular, F cannot be APN, and it can be a member of the following infinite families:

1. *Quadratic power permutations.*
2. *Bracken-Tan-Tan binomial permutations [11].*
3. *Göloğlu-Kölsch trivariate permutations [16].*

Proof. By Theorem 5 and Corollary 1, F admits a threshold implementation with 3 shares. F has differential uniformity at least $q \ge 4$ because

$$F(x+a) + F(x) + F(a) + F(0) = \sum_{0 \le i < j \le m} c_{i,j}(x^{q^i} a^{q^j} + x^{q^j} a^{q^i})$$

is a polynomial with at least q zeroes. See Appendix A.2 for details about the infinite families. □

Corollary 4. *Let m, t be positive integers with $1 < t < m$ and $\gcd(m,t) = 1$. Let q be a power of 2 such that $q \ge t+1$ and $\gcd(t, q-1) = 1$. Then*

$$F(x) = x^{\frac{q^t - 1}{q-1}}$$

is a power permutation of $\mathbb{F}_{q^m}$ that admits a threshold implementation with $t+1$ shares.

Proof. Suppose that a prime p divides both $d = \frac{q^t-1}{q-1}$ and $q^m - 1$. Since $d(q-1) = q^t - 1$, the multiplicative order of q modulo p divides both m and t, and hence equals 1. Thus $p \mid q-1$. But $d = 1 + q + \cdots + q^{t-1} = t \pmod p$, which contradicts $\gcd(t, q-1) = 1$. Therefore $\gcd(d, q^m - 1) = 1$ and F is a permutation. We can apply Theorem 5 because $\mathrm{w}_q(d) = t$. □

5 Computational results on small permutations

In this section, we investigate share-optimal threshold implementations of small permutations by computational methods. We establish that every four-bit permutation of algebraic degree t admits a direct, uniform TI with $t+1$ shares and no fresh randomness. Specifically, we obtain four-share TIs for all 295 cubic affine-equivalence classes and a three-share TI for the remaining quadratic class,

$\mathcal{Q}^4_{300}$. The latter contradicts the nonexistence claim in Corollary 2 of [5]. It also yields a three-share TI for its five-bit identity extension, $\mathcal{Q}^5_{30}$, invalidating the corresponding nonexistence conclusion for this class in [9]. Together with the previously known constructions, our results complete the four-bit case.

We first revisit the three-bit case, then present our search for four-share TIs of cubic four-bit permutations. By expressing a sharing in terms of derivatives, we reduce uniformity to a collection of smaller bijectivity conditions. We then introduce correction terms to satisfy these conditions while controlling the algebraic degree of the shared functions. Next, we give a three-share TI for $\mathcal{Q}^4_{300}$ and explain its consequences for implementations obtained by composing quadratic permutations. Finally, we extend the investigation to quadratic five-bit permutations and their cubic inverses.

We follow the indexing of [5] for three- and four-bit permutations, and that of [9] for the quadratic five-bit classes $\mathcal{Q}^5_i$. We additionally index the 57 cubic five-bit classes with quadratic inverses by setting $\mathcal{C}^5_{i+58} = \left(\mathcal{Q}^5_i\right)^{-1}$ for $18 \leq i \leq 29$ and $\mathcal{C}^5_{i+57} = \left(\mathcal{Q}^5_i\right)^{-1}$ for $31 \leq i \leq 75$. The remaining quadratic five-bit classes have quadratic inverses.

Following the *Sharing* column in the tables of [5], we report in Table 4 pairs of integers giving the numbers of stages in the provably shortest implementation with three and four shares, respectively. An entry equal to one therefore denotes a direct TI.

5.1 Three-bit permutations

We reconfirm that $\mathcal{Q}^3_3$ does not have a 3-share TI with an alternative proof that involves a SAT solver. We also update the corresponding entry in Table 4 from $2, 2$ to $2, 1$. We believe this must be a typo in the table of [5] since it is stated that they found a 4-share TI of $\mathcal{Q}^3_3$. For the remaining two nonlinear three-bit permutations, the entries are the same as in [5].

5.2 Four-share TIs of cubic four-bit permutations

Previously, direct four-share TIs were known for only four of the 295 cubic affine-equivalence classes of four-bit permutations: $\mathcal{C}^4_1$, $\mathcal{C}^4_3$, $\mathcal{C}^4_{13}$, and $\mathcal{C}^4_{301}$ [5]. We complete this classification by constructing a direct, uniform four-share TI without fresh randomness for every cubic class.

Our search starts from a correct, non-complete sharing obtained using the vanishing fourth derivative. Fixing one input share reduces the uniformity problem to smaller bijectivity conditions, which we satisfy by introducing suitable correction terms. We then coordinate these choices across the values of the fixed share to ensure that the resulting component functions have algebraic degree at most four. The following result is established by applying this construction to all 295 class representatives and verifying each resulting implementation exhaustively.

Theorem 6. *Every cubic permutation of $\mathbb{F}_2^4$ admits a threshold implementation with four shares and with algebraic degree at most four.*

Let $F\colon \mathbb{F}_2^4 \to \mathbb{F}_2^4$ be cubic and $\underline{x} = (x_1, x_2, x_3, x_4)$. Since the algebraic degree is three, the fourth derivative of F is always equal to zero. By moving everything except derivatives of zero to the right-hand side, we have

$$0 = D_{x_1} D_{x_2} D_{x_3} D_{x_4} F(0) = \sum_{I \subseteq \{1,2,3,4\}} F\left(\sum_{i \in I} x_i\right). \tag{17}$$

The sum contains one evaluation for each combination of the four shares. Then $F(x_1+x_2+x_3+x_4)$ is equal to the sum of the remaining fifteen linear evaluations of F. This identity gives a direct way to distribute the terms among four output components. Note that this distribution is not unique. We assign each evaluation to a component whose omitted share does not occur in its argument:

$$\begin{aligned}
\mathcal{F}_1(x_2, x_3, x_4) &= F(x_3 + x_4) + F(x_3) + F(x_2 + x_3 + x_4) \\
&\quad + F(x_2 + x_3) + F(x_4) + F(0), \\
\mathcal{F}_2(x_1, x_3, x_4) &= F(x_1 + x_4) + F(x_1) + F(x_1 + x_3 + x_4) + F(x_1 + x_3), \\
\mathcal{F}_3(x_1, x_2, x_4) &= F(x_2 + x_4) + F(x_2) + F(x_1 + x_2 + x_4) + F(x_1 + x_2), \\
\mathcal{F}_4(x_1, x_2, x_3) &= F(x_1 + x_2 + x_3).
\end{aligned} \tag{18}$$

Thus all fifteen proper-subset evaluations in (17) occur exactly once. Their sum is therefore $F(x_1 + x_2 + x_3 + x_4)$. So, correctness and non-completeness follow. However, that does not guarantee uniformity: for a general F, the map defined by (18) may have collisions.

The useful observation is that the evaluations in each of its first three components occur in pairs whose arguments differ by x_4. Recall that

$$F(z + x_4) + F(z) = D_{x_4} F(z).$$

By rewriting the sharing $\mathcal{F}$ from (18) by grouping into such pairs, substitung z in $D_{x_4}F(z)$. accordingly gives

$$\begin{aligned}
\mathcal{F}_1(x_2, x_3, x_4) &= D_{x_4} F(x_3) + D_{x_4} F(x_2 + x_3) + D_{x_4} F(0), \\
\mathcal{F}_2(x_1, x_3, x_4) &= D_{x_4} F(x_1) + D_{x_4} F(x_1 + x_3), \\
\mathcal{F}_3(x_1, x_2, x_4) &= D_{x_4} F(x_2) + D_{x_4} F(x_1 + x_2), \\
\mathcal{F}_4(x_1, x_2, x_3) &= F(x_1 + x_2 + x_3).
\end{aligned} \tag{19}$$

Now fix x_4. Since F is cubic, $z \mapsto D_{x_4}F(z)$ has degree at most two. Thus the first three components in (19) become functions of degree at most two in the remaining shares. Here, fixing x_4 is a device for the analysis and search: the implementation still accepts all four shares, and the argument must hold for every value of x_4.

Introducing correction terms for uniformity. Define $U, V, W : \mathbb{F}_2^4 \times \mathbb{F}_2^4 \to \mathbb{F}_2^4$ and write $U_{x_4}(z) = U(x_4, z)$, with the analogous convention for V and W. Each correction occurs in two components, therefore preserving correctness after summation: $U_{x_4}(x_1)$ can appear in components 2 and 3, $V_{x_4}(x_2)$ in components 1 and 3, and $W_{x_4}(x_3)$ in components 1 and 2.

$$\begin{aligned}
\mathcal{F}_1(x_2, x_3, x_4) &= D_{x_4}F(x_3) + D_{x_4}F(x_2 + x_3) + D_{x_4}F(0) \\
&\quad + V_{x_4}(x_2) + W_{x_4}(x_3), \\
\mathcal{F}_2(x_1, x_3, x_4) &= D_{x_4}F(x_1) + D_{x_4}F(x_1 + x_3) \\
&\quad + U_{x_4}(x_1) + W_{x_4}(x_3), \\
\mathcal{F}_3(x_1, x_2, x_4) &= D_{x_4}F(x_2) + D_{x_4}F(x_1 + x_2) \\
&\quad + U_{x_4}(x_1) + V_{x_4}(x_2), \\
\mathcal{F}_4(x_1, x_2, x_3) &= F(x_1 + x_2 + x_3).
\end{aligned} \tag{20}$$

Put $\sigma = x_1 + x_2 + x_3$. Then

$$\mathcal{F}_1 + \mathcal{F}_2 + \mathcal{F}_3 = D_{x_4}F(\sigma). \tag{21}$$

holds due to Equation (17) and (18).

The first three components therefore form a correct, non-complete three-sharing of the quadratic function $D_{x_4}F$ for each fixed x_4. Now we also fix σ. Once free shares x_1, x_2 are chosen, the third share must be $x_3 = \sigma + x_1 + x_2$. There are $16^2 = 256$ such input sharings with fixed values of x_4, σ.

On the output side, $\mathcal{F}_4 = F(\sigma)$ is already fixed. So after choosing $\mathcal{F}_1, \mathcal{F}_2$ the third output $\mathcal{F}_3 = D_{x_4}F(\sigma) + \mathcal{F}_1 + \mathcal{F}_2$ is also fixed.

Define an 8-bit sharing map:

$$H_{x_4,\sigma}(x_1, x_2) = \big(\mathcal{F}_1(x_2, \sigma + x_1 + x_2, x_4), \mathcal{F}_2(x_1, \sigma + x_1 + x_2, x_4)\big). \tag{22}$$

The whole sharing is uniform if and only if $H_{x_4,\sigma}$ is a permutation for every pair $(x_4, \sigma) \in (\mathbb{F}_2^4)^2$. Indeed, suppose first that all $H_{x_4,\sigma}$ are bijective and let (y_1, y_2, y_3, y_4) be an output. Since F is a permutation,

$$\sigma = F^{-1}(y_4), \qquad x_4 = F^{-1}(y_1 + y_2 + y_3 + y_4) + \sigma \tag{23}$$

are determined by the output. The inverse of $H_{x_4,\sigma}$ then gives x_1, x_2, after which $x_3 = \sigma + x_1 + x_2$. Thus the complete input sharing is uniquely recovered. On the other hand, a collision of one local map $H_{x_4,\sigma}$ breaks uniformity of the complete shared map. So, finding uniformity reduces to a collection of 256 permutation tests for all (x_4, σ).

For each nonzero x_4, we constrain the three correction terms to be linear mappings (so that the degree of each share is smaller).

$$\begin{aligned}
U_{x_4}(z) &= M_{x_4}^U z, & V_{x_4}(z) &= M_{x_4}^V z, \\
W_{x_4}(z) &= M_{x_4}^W z, & M_{x_4}^U, M_{x_4}^V, M_{x_4}^W &\in \mathbb{F}_2^{4\times 4}.
\end{aligned} \tag{24}$$

This gives 48 unknowns in total. For a proposed matrix triple, we evaluate the sum of collision for $H_{x_4,\sigma}$ for all σ

$$\text{error}_{x_4} = \sum_{\sigma \in \mathbb{F}_2^4} \left(2^8 - |\text{Im}(H_{x_4,\sigma})|\right). \tag{25}$$

The sum of collision is zero exactly when all sixteen maps are permutations. Computing it requires only $16 \cdot 2^8 = 4096$ evaluations.

We can encode these permutation conditions as a SAT instance. We accept a candidate only after exhaustive evaluation confirms its error is zero. We repeat this procedure for all fifteen nonzero values of x_4. We determine the matrices for $x_4 = 0$ afterward by the coupling step described below and check them separately for bijectivity.

Although each correction term is linear in its argument z, matrix entries may depend nonlinearly on the bits of x_4. Multiplying these entries by coordinates of z therefore produces nonlinear terms in the combined inputs (x_4, z). The restriction to linear terms keeps individual search small.

Remark 5. Notably, we aim to bound $\deg(\mathcal{F})$ to reduce latency delay and hardware area. Although each correction term is linear in its second argument, its matrix entries are functions of the four bits of x_4 and may produce terms of degree five in the end. Moreover, without this linearity restriction, the correction terms could have degree up to eight.

We further reduce the degree bound to four by coordinating the matrix choices for different values of x_4. For $x_4 = 0$, the derivative D_0F vanishes identically, and the explicit choice $M_0^U = M_0^V = I_4$, $M_0^W = 0$ gives $H_{0,\sigma}(x_1, x_2) = (x_2, x_1)$, which is a permutation for every σ. Thus, if uniformity were the only remaining requirement, this case could be fixed independently. To achieve the desired degree bound, however, we leave the matrices for $x_4 = 0$ undetermined at this stage and use it to cancel the degree-five terms. The following coupling step selects compatible matrices for the zero and nonzero cases, ensuring $\deg(\mathcal{F}) \leq 4$ while preserving uniformity.

Coupling and control of degree. Suppose that matrix triples have been found for each $x_4 \neq 0$. Those fifteen satisfy their local uniformity conditions independently. Now, to obtain $\deg(\mathcal{F}) \leq 4$, we coordinate them with the matrices at $x_4 = 0$. The purpose of this coupling is to eliminate the degree-four terms (degree-five terms in the corrections) in the matrix entries. We define

$$M_0^i = \sum_{x_4 \in \mathbb{F}_2^4 \setminus \{0\}} M_{x_4}^i, \qquad i \in \{U, V, W\}. \tag{26}$$

For every matrix entry $m(x_4)$, this ensures

$$\sum_{x_4 \in \mathbb{F}_2^4} m(x_4) = 0. \tag{27}$$

The sum is exactly the ANF coefficient of $x_{4,1}x_{4,2}x_{4,3}x_{4,4}$. Only the monomial involving all four variables contributes once. Consequently, zero sum removes the only possible degree-four monomial from every matrix entry. Each entry then has degree at most three in x_4, and multiplication by a coordinate of z gives a correction term of degree at most four. Since the uncorrected terms in (20) have degree at most three, we obtain $\deg(\mathcal{F}) \leq 4$.

The matrices prescribed by (26) satisfy the degree condition, but their uniformity must still be checked. Since $D_0F = 0$, substituting $x_3 = \sigma + x_1 + x_2$ gives

$$H_{0,\sigma}(x_1, x_2) = \begin{pmatrix} M_0^W & M_0^V + M_0^W \\ M_0^U + M_0^W & M_0^W \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} M_0^W \sigma \\ M_0^W \sigma \end{pmatrix}. \tag{28}$$

For each fixed σ, the last vector is a constant translation and does not affect bijectivity. Thus all sixteen maps $H_{0,\sigma}$ are permutations if and only if the 8×8 matrix in (28) is invertible. The uniformity test for the zero row therefore reduces to a single rank computation.

If this matrix is singular, we must modify the nonzero rows while preserving their zero error. For this purpose, we use the cyclic symmetry among the first three shares. Each zero-error matrix permits three cyclic choices, corresponding to cyclic relabellings of the first three output components $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$, and the associated correction functions. These choices preserve the local permutation conditions but can change the matrix sums in (26).

We select a cyclic choice for each of the fifteen nonzero correction terms, recompute the matrices at $x_4 = 0$, and test (28). Every such selection satisfies the degree condition by construction; it is accepted when the zero-row matrix is also invertible. If the available cyclic choices do not yield an invertible matrix, the search can be resumed to obtain alternative zero error triples.

Generalized search algorithm. We provide a generalization of our approach in Algorithm 1 in the appendix. It searches for $(t+1)$-share implementations of permutations $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ of degree $t \in \{1, 2, 3\}$, combining affine sharing for $t = 1$, quadratic unary corrections for $t = 2$, and conditioned matrix corrections with row coupling for $t = 3$. A successful search yields a direct, uniform TI without fresh randomness, with component degree at most t when $t \leq 2$ and at most $\max\{3, n\}$ when $t = 3$. The formulation applies in arbitrary dimension, although suitable corrections are not guaranteed to exist within the chosen search families.

The searches locate suitable matrices, while the acceptance conditions are exact. Zero defect certifies each nonzero row, the invertibility test certifies the zero row, and the final ANF and exhaustive checks verify the assembled implementation.

Applying this construction independently to the cubic affine-class representatives, and using affine equivalence to transfer the verified implementations, gives the following result.

5.3 A three-share TI for $\mathcal{Q}^4_{300}$.

Applying our correction search Algorithm 1 to $\mathcal{Q}^4_{300}$ yields a direct, uniform three-share TI without fresh randomness. The corrections U, V are quadratic and W is linear, so all three component functions have degree at most two. Appendix A.4 gives an explicit witness. Therefore, we can state the following theorem.

Theorem 7. *Every quadratic permutation of $\mathbb{F}_2^4$ admits a threshold implementation with three shares and with algebraic degree at most two.*

This construction contradicts Corollary 2 of [5]. Following Lemma 3, the authors state that a TI for a three-bit permutation gives a TI for the corresponding four-bit permutation obtained by adding a new variable, and vice versa. However, the reverse implication does not hold. The correction terms in the four-bit sharing may depend on the shares of the new variable. Removing this variable therefore need not give a uniform sharing of the three-bit permutation. Thus, the nonexistence of a three-share TI for $\mathcal{Q}^3_3$ does not imply the nonexistence of a three-share TI for $\mathcal{Q}^4_{300}$. The nonexistence result for $\mathcal{Q}^3_3$ is unaffected.

Our TI of $\mathcal{Q}^4_{300}$ can also be combined with known quadratic decompositions to obtain new three-share implementations of cubic four-bit permutations. By using the decompositions from [5], $\mathcal{C}^4_{130}$ and $\mathcal{C}^4_{131}$ now both admit a two-stage implementation, and $\mathcal{C}^4_{24}$ now admit a three-stage implementation. Table 4 reports these new decompositions, which are provably optimal.

Remark 6. The 6-bit Dillon APN permutation admits a decomposition into four quadratic permutations, as follows from the butterfly structure introduced in [29]. Two factors are affine equivalent to the identity extension of $\mathcal{Q}^3_3$ to six bits, while the other two are Feistel permutations with round functions in $\mathcal{Q}^3_3$. The first function is also affine equivalent to the identity extension of $\mathcal{Q}^4_{300}$ to six bits, meaning that it admits a 3-share TI. The two Feistel functions admit three-share TIs by [8]. In conclusion, we can implement the Dillon APN permutation with 3 shares and 4 stages.

5.4 Five-bit permutations.

We also apply Algorithm 1 to five-bit permutations, including conditioned derivative constructions for the cubic cases. We additionally prompted *ChatGPT (OpenAI, GPT-6)* to search for further share-optimal TIs. The model itself modified the search procedure and carried out the searches, reporting new TIs for the 18 quadratic classes $\mathcal{Q}^5_i$ with $i \in \{29, 35, 36, 38, 51, 54, 55, 60\} \cup \{40, \ldots, 49\}$ and for the cubic class $\mathcal{C}^5_{114}$. We omit the details of these modifications, as they did not transfer readily to the other computational searches reported here.

The verified corpus contains $(t+1)$-share TIs for 99 catalogue entries: 53 quadratic classes, 45 cubic classes, and the affine class. Equivalently, 98 of the 132 nonlinear classes. Previously, only 30 quadratic classes were known to admit a 3-share TI. Since this paper is the first to investigate cubic 5-bit permutation

classes with quadratic inverse, all the results are new. Every implementation is direct, uniform, and uses no fresh randomness. Correctness, non-completeness, and uniformity were checked exhaustively for each implementation. The explicit implementations and verification records are included in the accompanying artifact, and Table 4 lists all covered classes. The remaining 34 classes are unresolved by these searches. Those include the known quadratic APN permutations in dimension 5 that are respectively classes $\mathcal{Q}^5_{74}$ and $\mathcal{Q}^5_{75}$, and the χ function, which is denoted as class $\mathcal{Q}^5_{68}$.

6 Implementation

This section evaluates whether the direct construction from the previous section can be used in complete cipher cores rather than only as an isolated 4-bit permutation. We implement encryption cores for PRINCE, PRESENT-80, MIDORI-64, and SKINNY-64-64 [2, 4, 6, 7]. The selected variants have a 64-bit state and use a cubic 4-bit S-box. Consequently, we set $t = 3$ and use the direct $t + 1 = 4$-share realization of every S-box. The value t denotes the algebraic degree of the unshared S-box; it is not the security order. The implementations considered here target first-order, i.e., one-probe, security in the glitch-extended probing model [14].

These are the first complete round-based encryption cores using the direct, uniform, one-stage $t+1$ mappings for these four ciphers. The performance comparison is shown in Table 1. The distinguishing feature is that a cubic S-box is evaluated directly with four shares in one registered nonlinear interval. It is not decomposed into quadratic functions, divided over several compression stages, or refreshed with online randomness. The complete plaintext and master key are shared too, and the shared key schedule is included in the masking. Thus, after the initial sharing, the four designs consume no fresh random bits during encryption.

Another important feature is the *low-latency* of the direct, one-stage $t + 1$ construction: each S-box evaluation takes a single clock cycle, without intermediate register stages. This gives low encryption cycle counts together with short estimated delays in our implementations.

The purpose of this evaluation is deliberately limited: it serves as a direct application of our results, a proof of concept, and provides initial data points rather than a fully optimized implementation study. We therefore restrict ourselves to simulation and tool-based area and timing estimates. As the main contribution of this paper is theoretical, implementations on physical devices and corresponding real-world measurements are beyond the scope of this work.

6.1 Implementation organization

The hardware implementations were synthesized with Yosys 0.68/0.69 and ABC using the Nangate45 open-cell library at 1.10 V and 25 °C. All four implementations are iterative, encryption-only SystemVerilog cores. Each nonlinear com-

Table 1: Comparison with published protected implementations. For the rows reproduced from Table 2 of Ovchinnikov et al. [28], the area is its third area value, which includes Trivium where fresh randomness is required. PRESENT-80 was absent from that table; its entries are source-reported full-cipher core results or estimates, with the component-sum estimate marked by ^{E}.

Cipher	Masking method and source	Latency (cycles)	Init. (bits)	Fresh (bits/cycle)	Delay (ns)	Area (kGE)
PRINCE	TI, 3 state shares; unmasked key schedule [25]	40	128	0	4.00	9.29
	DOM [10]	24	192	112	8.37	16.63
	NullFresh [35]	24	192	0	5.74	11.46
	TSM [13]	12	192	288	2.06	24.11
	MultiStage [28]	72	192	0	1.45	24.90
	One-stage TI, 4 shares (this work)	**13**	**576**	**0**	**1.35^{L}**	**39.68^{L}**
PRESENT-80	TI, 3 shares (state only) [33]	547	NR	0	–	2.282^{P}
	TI, 3 shares (state and key) [33]	578	NR	0	–	3.322^{P}
	TI, 3 shares, round based (state only) [21]	186	NR	0	–	5.991^{E}
	NullFresh, 2 shares (state only) [35]	565	NR	0	4.59	1.819^{P}
	HPC2, 2 shares (state and key) [19]	2715	Ext.	4	0.99	4.160^{P}
	One-stage TI, 4 shares (this work)	**32**	**432**	**0**	**1.37^{L}**	**37.75^{L}**
MIDORI-64	TI, 3 state shares; unmasked key schedule [25]	32	128	0	4.00	7.30
	NullFresh; unmasked key schedule [35]	32	64	0	4.99	7.56
	HPC3 [18]	85	192	512	1.18	29.05
	GHPC* [19]	51	192	64	0.85	32.70
	HPC2* [19]	153	192	256	1.10	25.08
	HPC2*, low-delay variant [19]	153	192	256	0.95	61.58
	MultiStage [28]	96	192	0	1.11	21.50
	One-stage TI, 4 shares (this work)	**17**	**576**	**0**	**1.29^{L}**	**37.81^{L}**
SKINNY-64-64	TI, nibble serial [4]	704	256	0	0.95	2.00
	HPC3 [18]	99	128	128	1.06	10.34
	GHPC* [19]	99	128	64	0.80	31.01
	GHPC*, low-delay variant [19]	165	128	64	0.45	22.24
	HPC2* [19]	165	128	64	0.53	22.36
	AND–XOR1* [37]	132	128	128	0.45	17.57
	AND–XOR2* [37]	132	128	64	0.50	17.97
	MultiStage [28]	192	128	0	0.52	18.70
	One-stage TI, 4 shares (this work)	**33**	**384**	**0**	**1.12^{L}**	**31.91^{L}**

* Automatically generated row in the source comparison. NR: not reported; Ext.: externally shared inputs. ^{P}Source-reported PRESENT-80 core area without a PRNG. ^{E}Component-sum estimate reported by Kutzner et al. ^{L}Local Nangate45/Yosys–ABC result; the delay is a critical-path timing. Local area includes the practical core but excludes input-mask generation and output recombination. Local initialization counts include independent sharing of the plaintext and complete master key. Fresh denotes fresh random bits consumed per clock cycle. Published latency and implementation conventions are transcribed, not normalized.

ponent is followed by registers, preventing glitch propagation between successive nonlinear layers. Linear operations and key additions are performed share-wise, while public constants are added to share 0. The security argument assumes uniformly shared inputs and preservation of register boundaries and non-completeness after synthesis.

Shared key schedule. The complete master key is supplied and stored as four Boolean shares. The linear key schedules of PRINCE, MIDORI-64, and SKINNY-64-64 are evaluated share-wise. In PRESENT-80, the key is rotated by 61 bits, the most significant nibble is processed by the additional shared S-box. Thus, both the state and key schedule are masked, unlike the entries marked "state only" or "unmasked key schedule" in Table 1.

The cores receive already shared plaintexts and keys. The reported area includes the datapath, registers, and control logic, but excludes random number generation, input sharing, and ciphertext recombination.

Randomness and latency. No fresh randomness is required during encryption. Independently sharing a 64-bit plaintext and a k-bit master key into four shares requires $R_{\text{init}} = 3(64+k)$ bits. This gives 576 bits for PRINCE and MIDORI-64, 432 bits for PRESENT-80, and 384 bits for SKINNY-64-64. These counts assume independent input sharing for each encryption.

Latency is measured from acceptance of `start` to assertion of `done`:

$$L = N_{\text{nl}} + 1,$$

where N_{nl} is the number of registered nonlinear steps. The additional cycle stores the final ciphertext shares in the output registers. The resulting latencies are 13, 32, 17, and 33 cycles for PRINCE, PRESENT-80, MIDORI-64, and SKINNY-64-64, respectively, less than all alternatives [1], as reported in Table 1. The estimated minimum clock period (critical-path timing) are 1.35, 1.37, 1.29, and 1.12 ns, respectively.

6.2 Implemented ciphers

Prince PRINCE is a 64-bit low-latency block cipher with a 128-bit key and a reflection-based round structure [7]. The key is split into 64-bit words K_0 and K_1; K_0, its derived value K_0', and K_1 are used for whitening and round-key addition. In our implementation all operations deriving and selecting these words are carried out on each share separately. The shared state bank executes the forward rounds, the middle transformation, and the inverse rounds over

[1] The TSM [13] implementation of PRINCE requires 12 cycles but consumes 288 fresh random bits per clock cycle. Our implementation requires no fresh randomness during encryption. Moreover, our 13-cycle latency includes twelve registered nonlinear steps and one additional cycle to capture the ciphertext shares in dedicated output registers, providing a registered output interface for integration into a larger synchronous system.

twelve registered nonlinear steps. The inverse S-box does not require a second searched mapping: it is obtained from the forward S-box using the exact affine relation $S^{-1} = A \circ S \circ A$. The linear part of A is applied to each share, while its constant is placed in share 0.

The practical implementation takes 13 cycles and uses 39.68 kGE area. The comparison in Table 1 makes the tradeoff visible. TSM reports 12 cycles, 24.11 kGE, but also 288 fresh bits per cycle [13]. DOM reports 112 fresh bits per cycle and 24 cycles [10]. Among the zero-fresh entries, NullFresh reports 24 cycles and MultiStage reports 72 cycles [28, 35], Moradi and Schneider [25] propose a 3-stage 9.29-kGE, 40-cycle TI, whereas our direct one-stage TI uses 13 cycles and has smaller delay.

Present-80 PRESENT-80 is a lightweight 64-bit substitution–permutation cipher; we use the 80-bit-key variant with 31 rounds [6]. Sixteen shared S-boxes transform the state in parallel. After each substitution, the bit permutation is simply applied to every state share. The round key is taken from the shared 80-bit key register. The final round-key addition is followed by the output-capture period, for 32 cycles in total.

This case study makes the impact of architectural choice easy to see. The two TIs of Poschmann et al. and the NullFresh entry in Table 1 use serialized datapaths and report 547–578 and 565 cycles, respectively [33, 35]. The round-based three-share estimate of Kutzner et al. reports 186 cycles, marked ^{E} rather than as a synthesized full-core area [21]. The two-share HPC2 entry reports 2715 cycles with externally shared inputs and four fresh bits per cycle [19]. Our 32-cycle result follows from executing all 16 state nibbles in parallel and completing one round per registered nonlinear step. It also protects the nonlinear key update, unlike the rows marked "state only." The corresponding cost is 37.75 kGE, far above the serialized source-reported areas, but that is clear architectural tradeoff.

Midori-64 MIDORI-64 is a 64-bit low-energy block cipher with a 128-bit key [2]. The initial and final whitening key is derived from the two 64-bit key halves, and alternating halves are used in the round-key additions. Since this schedule is linear, the same selection and XOR network is duplicated for all four shares. Sixteen shared S-boxes perform SubCell in parallel. ShuffleCell is a nibble permutation, MixColumn is an XOR network, and the public round constants are introduced in share 0. Sixteen nonlinear transformations followed by final capture yield 17 cycles. The local area is 37.81 kGE.

The 32-cycle TI and NullFresh rows in Table 1 use an unmasked key schedule [25,35]. Our entry reports 17 cycles with the complete key shared. Compared with automatically generated private-circuit results, the one reports 85 cycles and 512 fresh bits per cycle for HPC3, 51 cycles and 64 fresh bits per cycle for GHPC, and 153 cycles and 256 fresh bits per cycle for the two HPC2 variants [18, 19]. MultiStage eliminates fresh randomness but reports 96 cycles and 21.50 kGE [28].

Skinny-64-64 SKINNY-64-64 is a tweakable block-cipher family; the implemented SKINNY-64-64 instance has a 64-bit state, a 64-bit tweakey, and 32 rounds [4]. Sixteen shared S-boxes process the whole state each cycle. ShiftRows, MixColumns, and the tweakey permutation are linear and are instantiated independently on every share. Only the upper half of the current shared tweakey is added in a round. After 32 nonlinear steps, the practical wrapper spends one cycle capturing the result, giving 33 cycles and 31.91 kGE.

The original published TI entry in Table 1 is a compact nibble-serial implementation: its 2.00 kGE area is accompanied by 704 cycles [4]. The generated HPC3, GHPC, HPC2, and AND–XOR entries require 99–165 cycles and 64–128 fresh bits per cycle [18,19,37]. MultiStage removes fresh randomness but reports 192 cycles [28]. Our result, again, uses no online randomness and has the smallest listed cycle count in this group.

6.3 Verification

We checked our four-share implementations for functional correctness and first-order glitch-extended probing security.

S-box verification. For the PRINCE, PRESENT-80, MIDORI-64, and SKINNY-64-64 S-boxes, all $2^{16} = 65\,536$ input-share tuples per S-box were checked for correctness and bijectivity, establishing uniformity conditioned on every unshared input. Structural checks established non-completeness at every gate, while exhaustive evaluation validated the mapped-cell expansion used by the verifiers. Yosys proved equivalence of all 64 factored component-coordinate functions to their expanded ANFs. SILVER [20] verified first-order security and glitch-extended probing security and uniformity; INDIANA [3] independently verified first-order glitch-extended probing security. All four S-boxes passed every check.

Round verification. INDIANA verifies seven complete round models, including the linear layer, shared round-key addition, and relevant key updates. The checks cover every original gate through complete output cones; no internal value is replaced with a fresh independent mask. At first order, every probe and its glitch extension lie within a checked cone. All 360 required cone instances pass (Table 2); 264 are byte-distinct targets.

These models use fixed public controls and independent uniform state and key masks. External plaintext/key pins, plaintext loading registers, and the previous ciphertext bank are fixed to zero. PRESENT's pending key-S-box result is derived from the same base-key variables, preserving its actual dependence. These assumptions are part of the verified models.

Complete-core functional verification. Yosys proves equality of all next-state bits and 258 output bits between each RTL core and its hierarchy-preserved mapped netlist. Separate reset and defined-value checks establish functional equivalence for arbitrary synchronous executions after reset. All four cores pass (Table 3). These are functional proofs, separate from probing security.

Table 2: Passed first-order glitch-extended probing checks for the specified round models. Cones retain all original ancestors and register boundaries.

Cipher	Round phase	Verified blocks	Result
PRINCE	Forward	63/63	Pass
	Middle	48/48	Pass
	Inverse	84/84	Pass
PRESENT-80	Regular round with key update	37/37	Pass
MIDORI-64	Upper key half	48/48	Pass
	Lower key half	48/48	Pass
SKINNY-64-64	Regular round	32/32	Pass
Total		360/360	Pass

The original simulation suite passes 12 known-answer tests and 32 random plaintext–key pairs per cipher. An additional controller suite completes 578 encryptions matching the reference ciphers and deliberately aborts 95 operations with reset. All assertions pass, covering loading, requests while busy, reset priority, completion cycles, and retention of individual ciphertext shares.

Table 3: Passed full-core functional checks. Encryption and reset counts refer to the additional controller suite.

Cipher	RTL/netlist	Assertions	Encryptions	Reset-aborted
PRINCE	Pass	Pass	125	13
PRESENT-80	Pass	Pass	162	32
MIDORI-64	Pass	Pass	130	17
SKINNY-64-64	Pass	Pass	161	33
Total			578	95

7 Conclusion

Our results show that optimal threshold implementations are far more common than previously conjectured. Although the three-bit inverse makes the universal $t + 2$-share bound tight, optimal TIs are ubiquitous in four bits and occur in cryptographically meaningful higher-dimensional families. The central question is therefore which structural obstructions prevent their existence. Among them is the question of whether an APN permutation admits an optimal TI. As a first concrete step here, one could try to settle this question for any of the known APN permutations in dimension 5. Another interesting Problem that goes beyond the pure existence is to find lightweight optimal TIs for cases where the known ones are outperformed by $t + 2$ TIs in order to improve the area.

Use of AI: We used different versions of ChatGPT (OpenAI) to assist with editing the manuscript, exploring theoretical ideas by the authors, and conducting computational experiments. The authors take full responsibility for the content and results of this work.

Concurrent and independent work Some of the results presented here were obtained concurrently and independently by Li and Tang [24]. We thank them for their helpful correspondence.

References

1. Andreoli, S., Piccione, E., Budaghyan, L., Stanica, P., Nikova, S.: On decompositions of permutations in quadratic functions. J. Cryptol. **38**(3), 22 (2025). https://doi.org/10.1007/S00145-025-09547-4
2. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A block cipher for low energy. In: Advances in Cryptology – ASIACRYPT 2015, Part II. Lecture Notes in Computer Science, vol. 9453, pp. 411–436. Springer (2015). https://doi.org/10.1007/978-3-662-48800-3_17
3. Beierle, C., Feldtkeller, J., Guinet, A., Güneysu, T., Leander, G., Richter-Brockmann, J., Sasdrich, P.: INDIANA – verifying (random) probing security through indistinguishability analysis. In: Advances in Cryptology – EUROCRYPT 2025, Part VIII. Lecture Notes in Computer Science, vol. 15608, pp. 33–63. Springer (2025). https://doi.org/10.1007/978-3-031-91101-9_2
4. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_5
5. Bilgin, B., Nikova, S., Nikov, V., Rijmen, V., Tokareva, N.N., Vitkup, V.: Threshold implementations of small s-boxes. Cryptogr. Commun. **7**(1), 3–33 (2015). https://doi.org/10.1007/s12095-014-0104-7
6. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: Cryptographic Hardware and Embedded Systems – CHES 2007. Lecture Notes in Computer Science, vol. 4727, pp. 450–466. Springer (2007). https://doi.org/10.1007/978-3-540-74735-2_31
7. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knežević, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçın, T.: PRINCE—a low-latency block cipher for pervasive computing applications—extended abstract. In: Advances in Cryptology – ASIACRYPT 2012. Lecture Notes in Computer Science, vol. 7658, pp. 208–225. Springer (2012). https://doi.org/10.1007/978-3-642-34961-4_14
8. Boss, E., Grosso, V., Güneysu, T., Leander, G., Moradi, A., Schneider, T.: Strong 8-bit sboxes with efficient masking in hardware extended version. J. Cryptogr. Eng. **7**(2), 149–165 (2017). https://doi.org/10.1007/s13389-017-0156-7
9. Bozilov, D., Bilgin, B., Sahin, H.A.: A note on 5-bit quadratic permutations' classification. IACR Trans. Symmetric Cryptol. **2017**(1), 398–404 (2017). https://doi.org/10.13154/tosc.v2017.i1.398-404
10. Bozilov, D., Knezevic, M., Nikov, V.: Optimized threshold implementations: Minimizing the latency of secure cryptographic accelerators. In: Smart Card Research and Advanced Applications – CARDIS 2019. Lecture Notes in Computer Science, vol. 11833, pp. 20–39. Springer (2020). https://doi.org/10.1007/978-3-030-42068-0_2

11. Bracken, C., Tan, C.H., Tan, Y.: Binomial differentially 4 uniform permutations with high nonlinearity. Finite Fields Their Appl. **18**(3), 537–546 (2012). https://doi.org/10.1016/J.FFA.2011.11.006
12. Daemen, J.: Changing of the guards: A simple and efficient method for achieving uniformity in threshold sharing. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10529, pp. 137–153. Springer (2017). https://doi.org/10.1007/978-3-319-66787-4_7
13. Dilip Kumar S. V., Dhooghe, S., Balasch, J., Gierlichs, B., Verbauwhede, I.: Time sharing: A novel approach to low-latency masking. IACR Transactions on Cryptographic Hardware and Embedded Systems **2024**(3), 249–272 (2024). https://doi.org/10.46586/tches.v2024.i3.249-272
14. Faust, S., Grosso, V., del Pozo, S.M., Paglialonga, C., Standaert, F.X.: Composable masking schemes in the presence of physical defaults & the robust probing model. IACR Transactions on Cryptographic Hardware and Embedded Systems **2018**(3), 89–120 (2018). https://doi.org/10.46586/tches.v2018.i3.89-120
15. Gantmacher, F.R.: The theory of matrices. Vols. 1, 2. Translated by K. A. Hirsch, Chelsea Publishing Co., New York (1959)
16. Göloğlu, F., Kölsch, L.: Triprojective almost perfect nonlinear permutations and functions. arXiv preprint arXiv:2605.17545 (2026)
17. Groß, H., Mangard, S., Korak, T.: Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In: Bilgin, B., Nikova, S., Rijmen, V. (eds.) Proceedings of the ACM Workshop on Theory of Implementation Security, TIS@CCS 2016 Vienna, Austria, October, 2016. p. 3. ACM (2016). https://doi.org/10.1145/2996366.2996426
18. Knichel, D., Moradi, A.: Low-latency hardware private circuits. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 1799–1812. ACM (2022). https://doi.org/10.1145/3548606.3559362
19. Knichel, D., Moradi, A., Müller, N., Sasdrich, P.: Automated generation of masked hardware. IACR Transactions on Cryptographic Hardware and Embedded Systems **2022**(1), 589–629 (2022). https://doi.org/10.46586/tches.v2022.i1.589-629
20. Knichel, D., Sasdrich, P., Moradi, A.: SILVER – statistical independence and leakage verification. In: Advances in Cryptology – ASIACRYPT 2020, Part I. Lecture Notes in Computer Science, vol. 12491, pp. 787–816. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_26
21. Kutzner, S., Nguyen, P.H., Poschmann, A., Wang, H.: On 3-share threshold implementations for 4-bit S-boxes. In: Constructive Side-Channel Analysis and Secure Design – COSADE 2013. Lecture Notes in Computer Science, vol. 7864, pp. 99–113. Springer (2013). https://doi.org/10.1007/978-3-642-40026-1_7
22. Levitzki, J.: Über nilpotente unterringe. Mathematische Annalen **105**(1), 620–627 (1931)
23. Li, Z., Tang, D.: Constructions of a family of nonlinear permutations of any possible algebraic degrees with the optimal threshold implementations. IACR Cryptol. ePrint Arch. **2025**, 2016 (2025), https://eprint.iacr.org/2025/2016
24. Li, Z., Tang, D.: New constructions of quadratic and cubic permutations admitting optimal threshold implementations. Cryptology ePrint Archive, Paper 2026/2416 (2026), https://eprint.iacr.org/2026/2416
25. Moradi, A., Schneider, T.: Side-channel analysis protection and low-latency in action: Case study of PRINCE and Midori. In: Advances in Cryptology – ASI-

ACRYPT 2016, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 517–547. Springer (2016). https://doi.org/10.1007/978-3-662-53887-6_19
26. Nikova, S., Rechberger, C., Rijmen, V.: Threshold implementations against side-channel attacks and glitches. In: Ning, P., Qing, S., Li, N. (eds.) Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4307, pp. 529–545. Springer (2006). https://doi.org/10.1007/11935308_38
27. Nikova, S., Rijmen, V., Schläffer, M.: Secure hardware implementation of nonlinear functions in the presence of glitches. J. Cryptol. **24**(2), 292–321 (2011). https://doi.org/10.1007/S00145-010-9085-7
28. Ovchinnikov, A., Shahmirzadi, A.R., Dhooghe, S.: Uniform sharing in multiple stages: NullFresh for arbitrary functions. IACR Transactions on Cryptographic Hardware and Embedded Systems **2026**(2), 893–927 (2026). https://doi.org/10.46586/tches.v2026.i2.893-927
29. Perrin, L., Udovenko, A., Biryukov, A.: Cryptanalysis of a theorem: Decomposing the only known solution to the big APN problem. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 93–122. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_4
30. Petrides, G.: On non-completeness in threshold implementations. In: Bilgin, B., Petkova-Nikova, S., Rijmen, V. (eds.) Proceedings of ACM Workshop on Theory of Implementation Security, TIS@CCS 2019, London, UK, November 11, 2019. pp. 24–28. ACM (2019). https://doi.org/10.1145/3338467.3358951
31. Piccione, E.: Threshold implementations of cryptographic functions between finite abelian groups. IACR Cryptol. ePrint Arch. **2024**, 439 (2024), https://eprint.iacr.org/2024/439
32. Piccione, E., Andreoli, S., Budaghyan, L., Carlet, C., Dhooghe, S., Nikova, S., Petrides, G., Rijmen, V.: An optimal universal construction for the threshold implementation of bijective s-boxes. IEEE Trans. Inf. Theory **69**(10), 6700–6710 (2023). https://doi.org/10.1109/TIT.2023.3287534
33. Poschmann, A., Moradi, A., Khoo, K., Lim, C.W., Wang, H., Ling, S.: Side-channel resistant crypto for less than 2,300 GE. Journal of Cryptology **24**(2), 322–345 (2011). https://doi.org/10.1007/s00145-010-9086-6
34. Radjavi, H., Rosenthal, P.: Simultaneous triangularization. Springer (2012)
35. Shahmirzadi, A.R., Moradi, A.: Re-consolidating first-order masking schemes: Nullifying fresh randomness. IACR Transactions on Cryptographic Hardware and Embedded Systems **2021**(1), 305–342 (2021). https://doi.org/10.46586/tches.v2021.i1.305-342
36. Varici, K., Nikova, S., Nikov, V., Rijmen, V.: Constructions of s-boxes with uniform sharing. Cryptogr. Commun. **11**(3), 385–398 (2019). https://doi.org/10.1007/S12095-018-0345-Y
37. Wu, L., Fan, Y., Preneel, B., Wang, W., Wang, M.: Automated generation of masked nonlinear components: From lookup tables to private circuits. In: Applied Cryptography and Network Security Workshops – ACNS 2024, Part I. Lecture Notes in Computer Science, vol. 14586, pp. 319–339. Springer (2024). https://doi.org/10.1007/978-3-031-61486-6_19

A Appendix

A.1 Triangular TI with $t+2$ shares

We adapt the universal construction of [32] to preserve the triangular structure.

We construct the triangular sharing

$$\mathcal{F}^{(0)}(\underline{a},\underline{c}) = \big(\mathcal{G}^{(0)}(\underline{a},\underline{c}), \mathcal{I}(\underline{c})\big),$$

where $\mathcal{I}(\underline{c}) = (c_2, \ldots, c_{t+2}, c_1)$ and

$$\begin{aligned}
\mathcal{G}_1^{(0)} &= \sum_{j=3}^{t+2} a_j + R_1, \\
\mathcal{G}_i^{(0)} &= a_{i+1} + R_i, \qquad 2 \le i \le t, \\
\mathcal{G}_{t+1}^{(0)} &= a_1 + a_{t+2} + R_{t+1}, \\
\mathcal{G}_{t+2}^{(0)} &= a_1.
\end{aligned}$$

Writing $x_i = (a_i, c_i)$, the R_i terms are

$$\begin{aligned}
R_1 &= G^{(0)}\left(\sum_{j=2}^{t+2} x_j\right) \\
R_i &= D_{x_1} \cdots D_{x_{i-1}} G^{(0)}\left(\sum_{j=i+1}^{t+2} x_j\right), \quad 2 \le i \le t, \\
R_{t+1} &= D_{x_1} \cdots D_{x_t} G^{(0)}(0).
\end{aligned}$$

A.2 Infinite families of quadratic highly nonlinear permutations admitting an optimal TI

The first family is the so-called Gold family of power permutations $F(x) = x^{2^k+1}$ over $\mathbb{F}_{2^n}$ where $n/\gcd(k,n)$ is odd. Let $g = \gcd(k,n)$. Then $2^k + 1 = q^{k/g} + 1$ where $q = 2^g$ and $q \ge 4$ if $g > 1$.

The second family is the so-called Bracken-Tan-Tan family [11] of binomial permutations F over $\mathbb{F}_{2^{3k}}$:

$$F(x) = ux^{2^s+1} + u^{2^k} x^{2^{2k}+2^{k+s}},$$

where s and k are positive integers with $k+s$ divisible by 3, $\gcd(3,k) = \gcd(3,s) = 1$ and u is a primitive element of $\mathbb{F}_{2^{3k}}$. Let $g = \gcd(k,s)$, then F has differential uniformity 2^g and it is a permutation if and only if k/g is odd. Let $q = 2^g$, then we can write $2^s+1 = q^{s/g}+1$ and $2^{2k}+2^{k+s} = q^{2k/g}+q^{(k+s)/g}$. Since $q \ge 3 = t+1$ when $g \ge 2$, we have that all the permutations with differential uniformity at least 4 admit a threshold implementation with 3 shares.

The third family is the Göloğlu-Kölsch family discovered in [16] of trivariate permutations F over $\mathbb{F}_{2^k}^3$:

$$F(x,y,z) = \begin{pmatrix} x^{\sigma+1} + ay^\sigma z + bx^\sigma y + cx^\sigma z, \\ ay^{\sigma+1} + z^\sigma x + bz^\sigma y + cx^\sigma y, \\ z^{\sigma+1} + x^\sigma y \end{pmatrix}$$

where $a, b, c \in \mathbb{F}_{2^k}$, $\sigma = 2^s$, $1 \leq s < k$, $g = \gcd(k, s)$ and $a \neq 0$. If the equation

$$ax^{\sigma^2+\sigma+1} + bx^{\sigma+1} + cx + 1 = 0$$

has no solution $x \in \mathbb{F}_{2^k}$, then the differential uniformity of F is 2^g and if additionally k/g is odd, then F is a permutation. Similarly as before, we have that all the permutations with differential uniformity at least 4 admit a threshold implementation with 3 shares.

A.3 Classification for 3-bit, 4-bit, and 5-bit Permutations.

Table 4: Shortest known threshold implementations of small permutations.

The columns s_3 and s_4 give the number of stages in the shortest known implementation with three and four shares, respectively. A dash indicates that such a decomposition is impossible; a question mark indicates that it is unknown. Blue entries denote new results.

Class	s_3	s_4	Class	s_3	s_4	Class	s_3	s_4	Class	s_3	s_4	Class	s_3	s_4	Class	s_3	s_4
Three-bit permutations																	
$\mathcal{Q}_1^3$	1	1	$\mathcal{Q}_2^3$	1	1	$\mathcal{Q}_3^3$	2	1									
Four-bit permutations																	
$\mathcal{C}_1^4$	–	1	$\mathcal{C}_2^4$	3	**1**	$\mathcal{C}_3^4$	–	1	$\mathcal{Q}_4^4$	1	1	$\mathcal{C}_5^4$	–	**1**	$\mathcal{C}_6^4$	3	**1**
$\mathcal{C}_7^4$	–	**1**	$\mathcal{C}_8^4$	3	**1**	$\mathcal{C}_9^4$	–	**1**	$\mathcal{C}_{10}^4$	3	**1**	$\mathcal{C}_{11}^4$	–	**1**	$\mathcal{Q}_{12}^4$	1	1
$\mathcal{C}_{13}^4$	–	1	$\mathcal{C}_{14}^4$	3	**1**	$\mathcal{C}_{15}^4$	–	**1**	$\mathcal{C}_{16}^4$	3	**1**	$\mathcal{C}_{17}^4$	–	**1**	$\mathcal{C}_{18}^4$	3	**1**
$\mathcal{C}_{19}^4$	–	**1**	$\mathcal{C}_{20}^4$	3	**1**	$\mathcal{C}_{21}^4$	–	**1**	$\mathcal{C}_{22}^4$	3	**1**	$\mathcal{C}_{23}^4$	–	**1**	$\mathcal{C}_{24}^4$	**3**	**1**
$\mathcal{C}_{25}^4$	–	**1**	$\mathcal{C}_{26}^4$	3	**1**	$\mathcal{C}_{27}^4$	–	**1**	$\mathcal{C}_{28}^4$	3	**1**	$\mathcal{C}_{29}^4$	–	**1**	$\mathcal{C}_{30}^4$	–	**1**
$\mathcal{C}_{31}^4$	3	**1**	$\mathcal{C}_{32}^4$	–	**1**	$\mathcal{C}_{33}^4$	3	**1**	$\mathcal{C}_{34}^4$	3	**1**	$\mathcal{C}_{35}^4$	–	**1**	$\mathcal{C}_{36}^4$	3	**1**
$\mathcal{C}_{37}^4$	–	**1**	$\mathcal{C}_{38}^4$	3	**1**	$\mathcal{C}_{39}^4$	–	**1**	$\mathcal{C}_{40}^4$	–	**1**	$\mathcal{C}_{41}^4$	3	**1**	$\mathcal{C}_{42}^4$	–	**1**
$\mathcal{C}_{43}^4$	3	**1**	$\mathcal{C}_{44}^4$	3	**1**	$\mathcal{C}_{45}^4$	–	**1**	$\mathcal{C}_{46}^4$	3	**1**	$\mathcal{C}_{47}^4$	–	**1**	$\mathcal{C}_{48}^4$	–	**1**
$\mathcal{C}_{49}^4$	3	**1**	$\mathcal{C}_{50}^4$	3	**1**	$\mathcal{C}_{51}^4$	–	**1**	$\mathcal{C}_{52}^4$	3	**1**	$\mathcal{C}_{53}^4$	3	**1**	$\mathcal{C}_{54}^4$	–	**1**
$\mathcal{C}_{55}^4$	3	**1**	$\mathcal{C}_{56}^4$	–	**1**	$\mathcal{C}_{57}^4$	3	**1**	$\mathcal{C}_{58}^4$	–	**1**	$\mathcal{C}_{59}^4$	–	**1**	$\mathcal{C}_{60}^4$	3	**1**
$\mathcal{C}_{61}^4$	3	**1**	$\mathcal{C}_{62}^4$	–	**1**	$\mathcal{C}_{63}^4$	3	**1**	$\mathcal{C}_{64}^4$	3	**1**	$\mathcal{C}_{65}^4$	–	**1**	$\mathcal{C}_{66}^4$	3	**1**
$\mathcal{C}_{67}^4$	3	**1**	$\mathcal{C}_{68}^4$	–	**1**	$\mathcal{C}_{69}^4$	–	**1**	$\mathcal{C}_{70}^4$	3	**1**	$\mathcal{C}_{71}^4$	3	**1**	$\mathcal{C}_{72}^4$	–	**1**
$\mathcal{C}_{73}^4$	3	**1**	$\mathcal{C}_{74}^4$	3	**1**	$\mathcal{C}_{75}^4$	–	**1**	$\mathcal{C}_{76}^4$	3	**1**	$\mathcal{C}_{77}^4$	–	**1**	$\mathcal{C}_{78}^4$	3	**1**
$\mathcal{C}_{79}^4$	–	**1**	$\mathcal{C}_{80}^4$	–	**1**	$\mathcal{C}_{81}^4$	3	**1**	$\mathcal{C}_{82}^4$	–	**1**	$\mathcal{C}_{83}^4$	3	**1**	$\mathcal{C}_{84}^4$	–	**1**
$\mathcal{C}_{85}^4$	3	**1**	$\mathcal{C}_{86}^4$	3	**1**	$\mathcal{C}_{87}^4$	–	**1**	$\mathcal{C}_{88}^4$	–	**1**	$\mathcal{C}_{89}^4$	3	**1**	$\mathcal{C}_{90}^4$	–	**1**
$\mathcal{C}_{91}^4$	–	**1**	$\mathcal{C}_{92}^4$	3	**1**	$\mathcal{C}_{93}^4$	–	**1**	$\mathcal{C}_{94}^4$	–	**1**	$\mathcal{C}_{95}^4$	3	**1**	$\mathcal{C}_{96}^4$	3	**1**

Continued on the next page

Table 4 (continued)

Class s_3 s_4	**Class** s_3 s_4	**Class** s_3 s_4	**Class** s_3 s_4	**Class** s_3 s_4	**Class** s_3 s_4
$\mathcal{C}^4_{97}$ – **1**	$\mathcal{C}^4_{98}$ – **1**	$\mathcal{C}^4_{99}$ 3 **1**	$\mathcal{C}^4_{100}$ – **1**	$\mathcal{C}^4_{101}$ 3 **1**	$\mathcal{C}^4_{102}$ – **1**
$\mathcal{C}^4_{103}$ 3 **1**	$\mathcal{C}^4_{104}$ 3 **1**	$\mathcal{C}^4_{105}$ – **1**	$\mathcal{C}^4_{106}$ – **1**	$\mathcal{C}^4_{107}$ 3 **1**	$\mathcal{C}^4_{108}$ – **1**
$\mathcal{C}^4_{109}$ – **1**	$\mathcal{C}^4_{110}$ 3 **1**	$\mathcal{C}^4_{111}$ 3 **1**	$\mathcal{C}^4_{112}$ – **1**	$\mathcal{C}^4_{113}$ – **1**	$\mathcal{C}^4_{114}$ 3 **1**
$\mathcal{C}^4_{115}$ 3 **1**	$\mathcal{C}^4_{116}$ – **1**	$\mathcal{C}^4_{117}$ – **1**	$\mathcal{C}^4_{118}$ 3 **1**	$\mathcal{C}^4_{119}$ – **1**	$\mathcal{C}^4_{120}$ 3 **1**
$\mathcal{C}^4_{121}$ 3 **1**	$\mathcal{C}^4_{122}$ – **1**	$\mathcal{C}^4_{123}$ 3 **1**	$\mathcal{C}^4_{124}$ 3 **1**	$\mathcal{C}^4_{125}$ – **1**	$\mathcal{C}^4_{126}$ 3 **1**
$\mathcal{C}^4_{127}$ 3 **1**	$\mathcal{C}^4_{128}$ – **1**	$\mathcal{C}^4_{129}$ – **1**	$\mathcal{C}^4_{130}$ **2** **1**	$\mathcal{C}^4_{131}$ **2** **1**	$\mathcal{C}^4_{132}$ – **1**
$\mathcal{C}^4_{133}$ – **1**	$\mathcal{C}^4_{134}$ 3 **1**	$\mathcal{C}^4_{135}$ – **1**	$\mathcal{C}^4_{136}$ 3 **1**	$\mathcal{C}^4_{137}$ – **1**	$\mathcal{C}^4_{138}$ 3 **1**
$\mathcal{C}^4_{139}$ – **1**	$\mathcal{C}^4_{140}$ 3 **1**	$\mathcal{C}^4_{141}$ – **1**	$\mathcal{C}^4_{142}$ 3 **1**	$\mathcal{C}^4_{143}$ – **1**	$\mathcal{C}^4_{144}$ – **1**
$\mathcal{C}^4_{145}$ 3 **1**	$\mathcal{C}^4_{146}$ – **1**	$\mathcal{C}^4_{147}$ 3 **1**	$\mathcal{C}^4_{148}$ 3 **1**	$\mathcal{C}^4_{149}$ – **1**	$\mathcal{C}^4_{150}$ 2 **1**
$\mathcal{C}^4_{151}$ 2 **1**	$\mathcal{C}^4_{152}$ – **1**	$\mathcal{C}^4_{153}$ – **1**	$\mathcal{C}^4_{154}$ 3 **1**	$\mathcal{C}^4_{155}$ 3 **1**	$\mathcal{C}^4_{156}$ – **1**
$\mathcal{C}^4_{157}$ – **1**	$\mathcal{C}^4_{158}$ 2 **1**	$\mathcal{C}^4_{159}$ 2 **1**	$\mathcal{C}^4_{160}$ – **1**	$\mathcal{C}^4_{161}$ 3 **1**	$\mathcal{C}^4_{162}$ 3 **1**
$\mathcal{C}^4_{163}$ – **1**	$\mathcal{C}^4_{164}$ 3 **1**	$\mathcal{C}^4_{165}$ 3 **1**	$\mathcal{C}^4_{166}$ – **1**	$\mathcal{C}^4_{167}$ – **1**	$\mathcal{C}^4_{168}$ 2 **1**
$\mathcal{C}^4_{169}$ – **1**	$\mathcal{C}^4_{170}$ – **1**	$\mathcal{C}^4_{171}$ 2 **1**	$\mathcal{C}^4_{172}$ 2 **1**	$\mathcal{C}^4_{173}$ – **1**	$\mathcal{C}^4_{174}$ 3 **1**
$\mathcal{C}^4_{175}$ – **1**	$\mathcal{C}^4_{176}$ 3 **1**	$\mathcal{C}^4_{177}$ – **1**	$\mathcal{C}^4_{178}$ 3 **1**	$\mathcal{C}^4_{179}$ – **1**	$\mathcal{C}^4_{180}$ 3 **1**
$\mathcal{C}^4_{181}$ – **1**	$\mathcal{C}^4_{182}$ – **1**	$\mathcal{C}^4_{183}$ 3 **1**	$\mathcal{C}^4_{184}$ 3 **1**	$\mathcal{C}^4_{185}$ – **1**	$\mathcal{C}^4_{186}$ – **1**
$\mathcal{C}^4_{187}$ 3 **1**	$\mathcal{C}^4_{188}$ – **1**	$\mathcal{C}^4_{189}$ 3 **1**	$\mathcal{C}^4_{190}$ – **1**	$\mathcal{C}^4_{191}$ – **1**	$\mathcal{C}^4_{192}$ 3 **1**
$\mathcal{C}^4_{193}$ – **1**	$\mathcal{C}^4_{194}$ 3 **1**	$\mathcal{C}^4_{195}$ – **1**	$\mathcal{C}^4_{196}$ 3 **1**	$\mathcal{C}^4_{197}$ – **1**	$\mathcal{C}^4_{198}$ 3 **1**
$\mathcal{C}^4_{199}$ – **1**	$\mathcal{C}^4_{200}$ – **1**	$\mathcal{C}^4_{201}$ 3 **1**	$\mathcal{C}^4_{202}$ 3 **1**	$\mathcal{C}^4_{203}$ – **1**	$\mathcal{C}^4_{204}$ – **1**
$\mathcal{C}^4_{205}$ 3 **1**	$\mathcal{C}^4_{206}$ – **1**	$\mathcal{C}^4_{207}$ – **1**	$\mathcal{C}^4_{208}$ 3 **1**	$\mathcal{C}^4_{209}$ – **1**	$\mathcal{C}^4_{210}$ – **1**
$\mathcal{C}^4_{211}$ 3 **1**	$\mathcal{C}^4_{212}$ 3 **1**	$\mathcal{C}^4_{213}$ – **1**	$\mathcal{C}^4_{214}$ 2 **1**	$\mathcal{C}^4_{215}$ 2 **1**	$\mathcal{C}^4_{216}$ – **1**
$\mathcal{C}^4_{217}$ 3 **1**	$\mathcal{C}^4_{218}$ – **1**	$\mathcal{C}^4_{219}$ 3 **1**	$\mathcal{C}^4_{220}$ – **1**	$\mathcal{C}^4_{221}$ 3 **1**	$\mathcal{C}^4_{222}$ – **1**
$\mathcal{C}^4_{223}$ 2 **1**	$\mathcal{C}^4_{224}$ – **1**	$\mathcal{C}^4_{225}$ 3 **1**	$\mathcal{C}^4_{226}$ – **1**	$\mathcal{C}^4_{227}$ – **1**	$\mathcal{C}^4_{228}$ 3 **1**
$\mathcal{C}^4_{229}$ – **1**	$\mathcal{C}^4_{230}$ – **1**	$\mathcal{C}^4_{231}$ 3 **1**	$\mathcal{C}^4_{232}$ – **1**	$\mathcal{C}^4_{233}$ 2 **1**	$\mathcal{C}^4_{234}$ 2 **1**
$\mathcal{C}^4_{235}$ – **1**	$\mathcal{C}^4_{236}$ 2 **1**	$\mathcal{C}^4_{237}$ – **1**	$\mathcal{C}^4_{238}$ 2 **1**	$\mathcal{C}^4_{239}$ – **1**	$\mathcal{C}^4_{240}$ 3 **1**
$\mathcal{C}^4_{241}$ – **1**	$\mathcal{C}^4_{242}$ – **1**	$\mathcal{C}^4_{243}$ 2 **1**	$\mathcal{C}^4_{244}$ 2 **1**	$\mathcal{C}^4_{245}$ – **1**	$\mathcal{C}^4_{246}$ – **1**
$\mathcal{C}^4_{247}$ 3 **1**	$\mathcal{C}^4_{248}$ – **1**	$\mathcal{C}^4_{249}$ 3 **1**	$\mathcal{C}^4_{250}$ – **1**	$\mathcal{C}^4_{251}$ – **1**	$\mathcal{C}^4_{252}$ 2 **1**
$\mathcal{C}^4_{253}$ – **1**	$\mathcal{C}^4_{254}$ 3 **1**	$\mathcal{C}^4_{255}$ – **1**	$\mathcal{C}^4_{256}$ – **1**	$\mathcal{C}^4_{257}$ – **1**	$\mathcal{C}^4_{258}$ 2 **1**
$\mathcal{C}^4_{259}$ 2 **1**	$\mathcal{C}^4_{260}$ 2 **1**	$\mathcal{C}^4_{261}$ – **1**	$\mathcal{C}^4_{262}$ 2 **1**	$\mathcal{C}^4_{263}$ – **1**	$\mathcal{C}^4_{264}$ 2 **1**
$\mathcal{C}^4_{265}$ – **1**	$\mathcal{C}^4_{266}$ 2 **1**	$\mathcal{C}^4_{267}$ – **1**	$\mathcal{C}^4_{268}$ 3 **1**	$\mathcal{C}^4_{269}$ – **1**	$\mathcal{C}^4_{270}$ 3 **1**
$\mathcal{C}^4_{271}$ – **1**	$\mathcal{C}^4_{272}$ – **1**	$\mathcal{C}^4_{273}$ 3 **1**	$\mathcal{C}^4_{274}$ – **1**	$\mathcal{C}^4_{275}$ 3 **1**	$\mathcal{C}^4_{276}$ – **1**
$\mathcal{C}^4_{277}$ 3 **1**	$\mathcal{C}^4_{278}$ 3 **1**	$\mathcal{C}^4_{279}$ – **1**	$\mathcal{C}^4_{280}$ 3 **1**	$\mathcal{C}^4_{281}$ 3 **1**	$\mathcal{C}^4_{282}$ – **1**
$\mathcal{C}^4_{283}$ – **1**	$\mathcal{C}^4_{284}$ – **1**	$\mathcal{C}^4_{285}$ – **1**	$\mathcal{C}^4_{286}$ 2 **1**	$\mathcal{C}^4_{287}$ 3 **1**	$\mathcal{C}^4_{288}$ 2 **1**
$\mathcal{C}^4_{289}$ – **1**	$\mathcal{C}^4_{290}$ – **1**	$\mathcal{C}^4_{291}$ – **1**	$\mathcal{C}^4_{292}$ 2 **1**	$\mathcal{Q}^4_{293}$ 1 1	$\mathcal{Q}^4_{294}$ 1 1
$\mathcal{C}^4_{295}$ – **1**	$\mathcal{C}^4_{296}$ 2 **1**	$\mathcal{C}^4_{297}$ 2 **1**	$\mathcal{C}^4_{298}$ – **1**	$\mathcal{Q}^4_{299}$ 1 1	$\mathcal{Q}^4_{300}$ **1** 1
$\mathcal{C}^4_{301}$ – 1					
Five-bit quadratic permutations					
$\mathcal{Q}^5_{1}$ 1 1	$\mathcal{Q}^5_{2}$ 1 1	$\mathcal{Q}^5_{3}$ 1 1	$\mathcal{Q}^5_{4}$ 1 1	$\mathcal{Q}^5_{5}$ 1 1	$\mathcal{Q}^5_{6}$ 1 1
$\mathcal{Q}^5_{7}$ 1 1	$\mathcal{Q}^5_{8}$ 1 1	$\mathcal{Q}^5_{9}$ 1 1	$\mathcal{Q}^5_{10}$ 1 1	$\mathcal{Q}^5_{11}$ 1 1	$\mathcal{Q}^5_{12}$ 1 1
$\mathcal{Q}^5_{13}$ 1 1	$\mathcal{Q}^5_{14}$ 1 1	$\mathcal{Q}^5_{15}$ 1 1	$\mathcal{Q}^5_{16}$ 1 1	$\mathcal{Q}^5_{17}$ 1 1	$\mathcal{Q}^5_{18}$ 1 1
$\mathcal{Q}^5_{19}$ 1 1	$\mathcal{Q}^5_{20}$ 1 1	$\mathcal{Q}^5_{21}$ 1 1	$\mathcal{Q}^5_{22}$ 1 1	$\mathcal{Q}^5_{23}$ 1 1	$\mathcal{Q}^5_{24}$ 1 1
$\mathcal{Q}^5_{25}$ 1 1	$\mathcal{Q}^5_{26}$ 1 1	$\mathcal{Q}^5_{27}$ 1 1	$\mathcal{Q}^5_{28}$ **1** 1	$\mathcal{Q}^5_{29}$ **1** 1	$\mathcal{Q}^5_{30}$ **1** 1
$\mathcal{Q}^5_{31}$ 1 1	$\mathcal{Q}^5_{32}$ **1** 1	$\mathcal{Q}^5_{33}$ 1 1	$\mathcal{Q}^5_{34}$ 1 1	$\mathcal{Q}^5_{35}$ **1** 1	$\mathcal{Q}^5_{36}$ **1** 1
$\mathcal{Q}^5_{37}$ **1** 1	$\mathcal{Q}^5_{38}$ **1** 1	$\mathcal{Q}^5_{39}$ **1** 1	$\mathcal{Q}^5_{40}$ **1** 1	$\mathcal{Q}^5_{41}$ **1** 1	$\mathcal{Q}^5_{42}$ **1** 1
$\mathcal{Q}^5_{43}$ **1** 1	$\mathcal{Q}^5_{44}$ **1** 1	$\mathcal{Q}^5_{45}$ **1** 1	$\mathcal{Q}^5_{46}$ **1** 1	$\mathcal{Q}^5_{47}$ **1** 1	$\mathcal{Q}^5_{48}$ **1** 1
$\mathcal{Q}^5_{49}$ **1** 1	$\mathcal{Q}^5_{50}$? 1	$\mathcal{Q}^5_{51}$ **1** 1	$\mathcal{Q}^5_{52}$? 1	$\mathcal{Q}^5_{53}$? 1	$\mathcal{Q}^5_{54}$ **1** 1

Continued on the next page

Table 4 (continued)

Class	s_3	s_4	Class	s_3	s_4	Class	s_3	s_4	Class	s_3	s_4	Class	s_3	s_4	Class	s_3	s_4
$\mathcal{Q}^5_{55}$	1	1	$\mathcal{Q}^5_{56}$	?	1	$\mathcal{Q}^5_{57}$	?	1	$\mathcal{Q}^5_{58}$	?	1	$\mathcal{Q}^5_{59}$	?	1	$\mathcal{Q}^5_{60}$	1	1
$\mathcal{Q}^5_{61}$	?	1	$\mathcal{Q}^5_{62}$	?	1	$\mathcal{Q}^5_{63}$	?	1	$\mathcal{Q}^5_{64}$	?	1	$\mathcal{Q}^5_{65}$	?	1	$\mathcal{Q}^5_{66}$	?	1
$\mathcal{Q}^5_{67}$	?	1	$\mathcal{Q}^5_{68}$	?	1	$\mathcal{Q}^5_{69}$	?	1	$\mathcal{Q}^5_{70}$	?	1	$\mathcal{Q}^5_{71}$	?	1	$\mathcal{Q}^5_{72}$	?	1
$\mathcal{Q}^5_{73}$	?	1	$\mathcal{Q}^5_{74}$	?	1	$\mathcal{Q}^5_{75}$	?	1									
Five-bit cubic permutations (with quadratic inverse)																	
$\mathcal{C}^5_{76}$	?	1	$\mathcal{C}^5_{77}$	?	1	$\mathcal{C}^5_{78}$	?	1	$\mathcal{C}^5_{79}$	?	1	$\mathcal{C}^5_{80}$	?	1	$\mathcal{C}^5_{81}$	?	1
$\mathcal{C}^5_{82}$	?	1	$\mathcal{C}^5_{83}$	?	1	$\mathcal{C}^5_{84}$	?	1	$\mathcal{C}^5_{85}$	?	1	$\mathcal{C}^5_{86}$	?	1	$\mathcal{C}^5_{87}$	?	1
$\mathcal{C}^5_{88}$	?	1	$\mathcal{C}^5_{89}$	?	1	$\mathcal{C}^5_{90}$	?	1	$\mathcal{C}^5_{91}$	?	1	$\mathcal{C}^5_{92}$	?	1	$\mathcal{C}^5_{93}$	?	1
$\mathcal{C}^5_{94}$	?	1	$\mathcal{C}^5_{95}$	?	1	$\mathcal{C}^5_{96}$	?	1	$\mathcal{C}^5_{97}$	?	1	$\mathcal{C}^5_{98}$	?	1	$\mathcal{C}^5_{99}$	?	1
$\mathcal{C}^5_{100}$	?	1	$\mathcal{C}^5_{101}$	?	1	$\mathcal{C}^5_{102}$	?	1	$\mathcal{C}^5_{103}$	?	1	$\mathcal{C}^5_{104}$	?	1	$\mathcal{C}^5_{105}$	?	1
$\mathcal{C}^5_{106}$	?	1	$\mathcal{C}^5_{107}$	?	1	$\mathcal{C}^5_{108}$	?	1	$\mathcal{C}^5_{109}$	?	1	$\mathcal{C}^5_{110}$	?	?	$\mathcal{C}^5_{111}$	?	1
$\mathcal{C}^5_{112}$	?	1	$\mathcal{C}^5_{113}$	?	1	$\mathcal{C}^5_{114}$	?	1	$\mathcal{C}^5_{115}$	?	1	$\mathcal{C}^5_{116}$	?	?	$\mathcal{C}^5_{117}$	?	1
$\mathcal{C}^5_{118}$	?	1	$\mathcal{C}^5_{119}$	?	1	$\mathcal{C}^5_{120}$	?	1	$\mathcal{C}^5_{121}$	?	1	$\mathcal{C}^5_{122}$	?	1	$\mathcal{C}^5_{123}$	?	?
$\mathcal{C}^5_{124}$	?	?	$\mathcal{C}^5_{125}$	?	?	$\mathcal{C}^5_{126}$	?	?	$\mathcal{C}^5_{127}$	?	?	$\mathcal{C}^5_{128}$	?	?	$\mathcal{C}^5_{129}$	?	?
$\mathcal{C}^5_{130}$	?	?	$\mathcal{C}^5_{131}$	?	?	$\mathcal{C}^5_{132}$	?	?									

A.4 A three-share TI for $\mathcal{Q}^4_{300}$

For the representative $F = \texttt{0123458967CDEFAB}$, let the values of the correction terms, in hexadecimal input order $0, 1, \ldots, \texttt{F}$, be

$$\begin{aligned} U &= \texttt{00FF44552255AA33}, \\ V &= \texttt{0B29AF059A74D0B6}, \\ W &= \texttt{059CC9508D1441D8}. \end{aligned}$$

Since $F(0) = 0$, the corrected sharing is

$$\begin{aligned} \mathcal{F}_1(x_2, x_3) &= D_{x_2}F(x_3) + V(x_2) + W(x_3), \\ \mathcal{F}_2(x_1, x_3) &= D_{x_3}F(x_1) + U(x_1) + W(x_3), \\ \mathcal{F}_3(x_1, x_2) &= D_{x_1}F(x_2) + U(x_1) + V(x_2). \end{aligned} \tag{29}$$

Every correction occurs twice in the reconstructed output. The identity $D_{x_1}D_{x_2}D_{x_3}F(0) = 0$ therefore gives correctness. Each component omits its corresponding input share and has degree two. Direct enumeration gives 4096 distinct outputs, including 256 distinct output sharings for each fixed secret.

A.5 Algorithm for finding $t + 1$ TI for arbitrary permutations with degree up to 3.

Algorithm 1 Search for a $(t+1)$-share TI in dimension n

Require: A permutation $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ with $t = \deg F \in \{1, 2, 3\}$; finite search limits.
Ensure: A correct, non-complete, uniform $(t+1)$-share implementation $\mathcal{F}$, or NotFound.

Set $s = t + 1$.
Let $x_1, \ldots, x_s \in \mathbb{F}_2^n$ denote the input shares, with $x = x_1 + \cdots + x_s$.
if $t = 1$ **then**
    Write $F(x) = Ax + c$.
    return $\mathcal{F}(x_1, x_2) = (Ax_2 + c, Ax_1)$.
else if $t = 2$ **then**
    Parameterize $U, V, W : \mathbb{F}_2^n \to \mathbb{F}_2^n$ by ANFs of degree at most two.
    Define the corrected sharing:
$$\mathcal{F}_1(x_2, x_3) = D_{x_2}F(x_3) + F(0) + V(x_2) + W(x_3),$$
$$\mathcal{F}_2(x_1, x_3) = D_{x_3}F(x_1) + F(0) + U(x_1) + W(x_3),$$
$$\mathcal{F}_3(x_1, x_2) = D_{x_1}F(x_2) + F(0) + U(x_1) + V(x_2).$$
    Search for coefficients of U, V, W making $\mathcal{F}$ bijective, using local search or SAT.
    if no suitable corrections are found **then**
        return NotFound
    end if
else ▷ $t = 3$, hence $s = 4$
    Condition on the last input share $x_s = h$.
    for all $h \in \mathbb{F}_2^n \setminus \{0\}$ **do**
        Compute D_hF and form $H_{h,\sigma}$.
        Search for $M_h^U, M_h^V, M_h^W \in \mathbb{F}_2^{n\times n}$ such that
$$\sum_{\sigma \in \mathbb{F}_2^n} \left(2^{2n} - |\operatorname{Im} H_{h,\sigma}|\right) = 0.$$
        if no suitable matrix triple is found **then**
            return NotFound
        end if
        Verify the row on all 2^{3n} local inputs.
    end for
    Search for cyclic variants of the nonzero rows satisfying
$$M_0^T = \sum_{h \in \mathbb{F}_2^n \setminus \{0\}} M_h^T, \qquad T \in \{U, V, W\},$$
$$\operatorname{rank} \begin{pmatrix} M_0^W & M_0^V + M_0^W \\ M_0^U + M_0^W & M_0^W \end{pmatrix} = 2n.$$
    if no compatible choices are found **then**
        return NotFound
    end if
    Set $T(h, z) = M_h^T z$ for $T \in \{U, V, W\}$ and $h, z \in \mathbb{F}_2^n$.
    Assemble $\mathcal{F}$ using (20), with $h = x_s$ and all shares in $\mathbb{F}_2^n$.
end if
Compute the exact ANFs of all component functions.
Check that $\mathcal{F}_j$ omits x_j for every $j \in \{1, \ldots, s\}$.
Check component degree at most 2 if $t = 2$, and at most $\max\{3, n\}$ if $t = 3$.
Verify reconstruction and bijectivity on all 2^{sn} shared inputs.
if all verification checks pass **then**
    return $\mathcal{F}$
else
    return NotFound
end if