The QARMA Block Cipher Family

Almost MDS Matrices Over Rings With Zero Divisors, Nearly Symmetric Even-Mansour Constructions With Non-Involutory Central Rounds, and Search Heuristics for Low-Latency S-Boxes

Roberto Avanzi

Qualcomm Product Security Initiative, Munich, Germany
ravanzi@qti.qualcomm.com

Abstract. We introduce and analyse a family of Almost MDS matrices defined over a ring with zero divisors that allows us to encode rotations in its operation while maintaining the minimal latency associated to $\{0, 1\}$-matrices. We also describe new S-Box search heuristics aimed at minimising the critical path.
These techniques are used to define some components of QARMA, a new family of lightweight tweakable block ciphers. QARMA is targeted to a very specific set of use cases, such as memory encryption, generation of very short tags by truncation, and the construction of keyed hash functions, in fully unrolled hardware implementations. The structure of the cipher is inspired by PRINCE. However, it differs from reflector constructions in that it is a three-round Even-Mansour scheme with a non-involutory keyed middle permutation designed to thwart various classes of attacks. QARMA aims a providing conservative security margins while still achieving best-in-class latency. QARMA exists in 64- and 128-bit block sizes, with 128- and 256-bit keys, respectively. Implementors are also offered a reduced set of S-Boxes to choose from.

Keywords: Tweakable Block Ciphers, Almost MDS Matrices, Even-Mansour Schemes, S-Box Search Heuristics, Memory Encryption, Short Tags, Pointer Signing

1 Introduction

During the last few years, lightweight block ciphers have been the subject of considerable research, motivated by the need to provide an acceptable security level for *specific* applications at much lower area, latency, or power consumption points than, say, the AES [1]. In lightweight block ciphers, all components must be tightly optimised, often leading to original solutions, as exemplified by CLEFIA [2], KATAN and KTANTAN [3], KLEIN [4], LED [5], PRESENT [6], PRINCE [7], SIMON and SPECK [8], and MIDORI [9], to name just a few of all published proposals.

At the same time, there has been research in *tweakable* block ciphers [10] (TBC) that, besides the key and a plaintext or ciphertext, accept a *third* input called the *tweak*. The tweak, along with the key, selects the permutation computed by the cipher. TBCs are used in the design of encryption modes of operation and hash functions, as well as in disk [11] and memory encryption [12]. In these cases the tweak is a counter or chaining input, or an absolute address on a memory or storage device. A further application is to software security: to enforce code flow integrity very short tags are inserted in pointers in unused bits of the address space; in this case the tweak is the label of the pointer's context.

Hence, *changing the tweak must be a very agile operation* and *a tweakable cipher is secure if it cannot be broken even assuming the adversary has full tweak control.*

The first TBC, the AES submission *Hasty Pudding Cipher*, was quickly followed by schemes to create TBCs from ordinary primitives used as black boxes [13,14,15]. There have been only a very few ad-hoc designs since, such as MERCY [16], the cipher THREEFISH which is at the core of the Skein hash function [17], and the ciphers Deoxys-BC, Joltik-BC, and Kiasu-BC based on the TWEAKEY framework [18]. More recently, we have SKINNY and MANTIS [19]. Designing a TBC is a difficult task as care must be taken in how the user-controlled tweak is included in the design.

Several memory encryption solutions, such as Intel's SGX [20], use a counter based mode with memory overhead. However, if the use case does not allow any form of memory expansion, the most straightforward solution seems to be a TBC in ECB mode. Since all generic construction to tweak a block cipher suffer from increased latency, an ad-hoc approach is necessary.

For applications such as memory encryption and software security, reduction of total latency is the most important performance parameter, whereas area and energy come second - for instance, for memory and disk encryption energy consumption is dominated by that of the memory or mass storage controller and related hardware.

Security Model. *We shall assume the attacker does not have control on the key, but she may have full control on the tweak. A TBC is understood to offer n bits of (time-data tradeoff) security if no better attacks are possible than time $2^{n-d-\epsilon}$ with 2^d chosen or known {plaintext, ciphertext, tweak} triples, for a small ϵ (e.g. 2).*

Contents of the Paper. This paper makes four contributions.

The first is QARMA, a family of hardware-oriented lightweight TBCs which also serves as a frame for the other three contributions. It is targeted at memory encryption and design of short hash functions. QARMA's latency is sufficiently small to allow usage in tweaked ECB modes that eschew the masking value derivation typical of higher latency XEX-like constructions. Absolute minimisation of gate count is not our primary design goal, so we allow some more expensive choices to build better security margins. QARMA reuses some concepts from PRINCE, MIDORI and MANTIS, but there are several important differences both in structure and in choice of components: the common aspects allow us to concentrate on the design and analysis of the original ones. The TWEAKEY framework is taken as an inspiration: The bits of key and tweak are not permuted synchronously, but instead only those of the tweak are shuffled between rounds; additionally, a LFSR is used to update the tweak.

QARMA supports blocks sizes of $n = 64$ and $n = 128$ bits. The tweak is also n bits long and the key is always $2\,n$ bits long. QARMA-128 is suitable for more complex memory or storage encryption techniques as the tweak has sufficient entropy for an address, various tags, and a version counter or nonce. We do not consider the fact that QARMA-128 only supports 256-bit keys a limitation. Note that for storage or memory encryption the use of larger keys is common: For instance, if AES-128 in XTS-mode is used, 256 bits of key material are *already* required to meet NIST FIPS compliance requirements [21, Appendix A.9]. Hence, the same key storage structures can be reused if QARMA-128 replaces or supplements the AES.

The other three contributions are: (i) The first characterisation of Almost MDS circulant matrices over the ring $R_m = \mathbb{F}_2[\rho] = \mathbb{F}_2[X]/(X^m + 1)$ that we are aware of, in order to model circular rotations – these matrices may help hedge against some types of iterative characteristics, (the involutory ones) can be used to construct better reflectors for lightweight reflection ciphers, have optimal critical path and we also determine those with equally efficient inverses; (ii) Adoption of an Even-Mansour scheme with a keyed *pseudo-reflector* to avoid certain types of cryptanalysis, and (iii) Heuristics to efficiently find S-Boxes with short critical path.

Structure of the Paper. As QARMA serves as frame for the other results, we begin with its specification in Section 2, where only some technical details are missing. The design decisions, other contributions, and missing details are presented in Section 3. We discuss security in Section 4. Implementation results are presented in Section 5.

Statements. QARMA is unpatented and, to the best of our knowledge, we are not aware of any intellectual property encumbering it. The algorithm is placed in the public domain. QARMA is pronounced like the sanskrit word *Karma*.

2 Specification of QARMA

2.1 General Definitions and Notation

The overall scheme of the TBC QARMA is depicted in Figure 1. There, and throughout the paper, a bar over a function (e.g. $\overline{F}$) denotes its inverse. QARMA is a 3-round Even-Mansour construction where the permutations are parameterized by a core key, and the key mixings between rounds are derived from a whitening key. The first and third permutations are functionally the inverse of each other and are further parameterized by a tweak, and the central permutation is designed to be easily inverted by means of a simple transformation of the key.

The cipher is depicted in more detail in Figure 2. QARMA is a bricklayer SPN. It is easily seen that components and ideas have been borrowed from other ciphers such as PRINCE, MIDORI and MANTIS, but because of the specific keying and central permutation, this is also a new design with its own security properties.

The keys k^0, k^1, w^0, and w^1 are derived from a master key K via a simple *key specialisation.* The letters P, C and T denote the plaintext, the ciphertext and the tweak; S represents a layer of sixteen m-bit S-Boxes, h and τ are permutations, M and Q are MixColumns-like operations, with Q involutory, and ω is a LFSR.

Fig. 1. The Overall Scheme

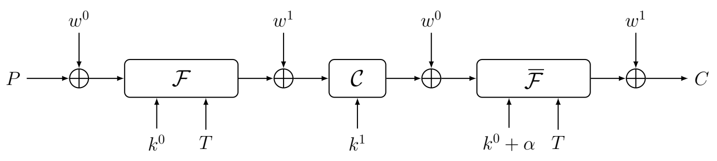
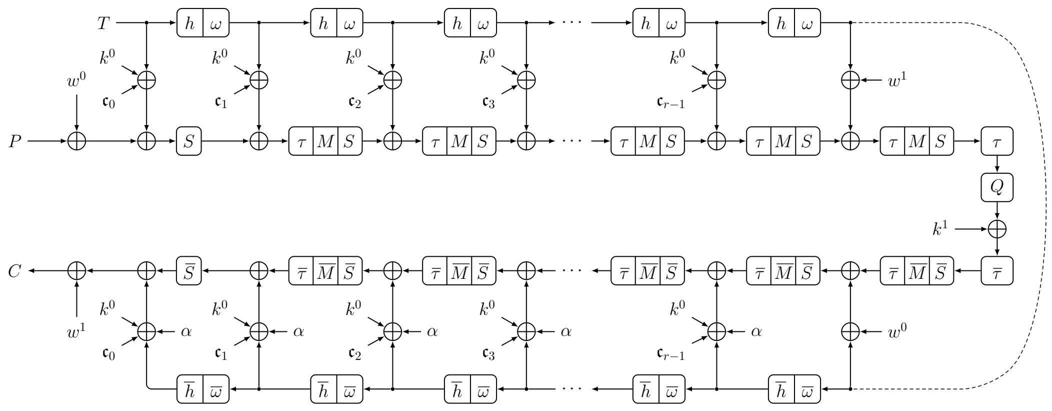


Fig. 2. The Structure of $\texttt{QARMA}_r$



Write $n = 16\,m$. It is $m = 4$ or 8. All n-bit values are represented as arrays of sixteen m-bit *cells*, which are indexed in big endian order, while the bits inside a cell are ordered in little endian order. Any array of sixteen cells is also viewed as a 4×4 matrix, for instance, the internal state admits representations

$$\mathrm{IS} = s_0 \| s_1 \| \cdots \| s_{14} \| s_{15} = \begin{pmatrix} s_0 & s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 & s_7 \\ s_8 & s_9 & s_{10} & s_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{pmatrix} , \tag{1}$$

so that $4{\times}4$ matrices operates column-wise on these values by left multiplication. The plaintext is given as $P = p_0 \| p_1 \| \cdots \| p_{14} \| p_{15}$, the tweak as $T = t_0 \| t_1 \| \cdots \| t_{14} \| t_{15}$.

Throughout the paper, we use the symbol "$+$" to denote addition in all algebraic structures. In particular it denotes the exclusive or in the `QARMA` ciphers, which do not use modular addition. The symbol `tk` denotes a (round) *tweakey*, i.e. a value derived only from the key, tweak, and round constants.

2.2 Key Specialisation

The $2\,n = 32\,m$ bit key K is first partitioned as $w^0 \| k^0$ where w^0 and k^0, the *whitening* and *core* keys, are $16\,m$ bits each.

For encryption, we put $w^1 = (w^0 \ggg 1) + (w^0 \gg (16\,m - 1))$ and $k^1 = k^0$.

Since the first r rounds of the cipher (ignoring initial whitening) differ from last r rounds solely by the addition of a non-zero constant α, QARMA possesses a property very similar to PRINCE's α-reflection: The encryption circuit can be used for decryption when $k^0 + \alpha$ is used as the core key, the whitening keys w^0 with w^1 are swapped, and $k^1 = Q \cdot k^0$.

2.3 The Forward Round Function

The *Forward Round Function* $\mathcal{R}(\mathrm{IS}; \mathtt{tk})$ is composed by four operations, performed in the following order:

1. AddRoundTweakey. The round tweakey tk defined in Section 2.7 is XORed to IS.
2. ShuffleCells. $(\tau(\mathrm{IS}))_i = s_{\tau(i)}$ for $0 \le i \le 15$, where τ is the MIDORI cell permutation, i.e. $\tau = [\,0, 11, 6, 13, 10, 1, 12, 7, 5, 14, 3, 8, 15, 4, 9, 2\,]$.
3. MixColumns. Each column of the cipher internal state array is multiplied by the matrix M defined in Section 3.1, i.e. $\mathrm{IS} = M \cdot \mathrm{IS}$.
4. SubCells. For the chosen S-Box σ, the S layer acts on the state as follows: $s_i \hookleftarrow \sigma(s_i)$ for $0 \le i \le 15$. The S-Boxes are defined in §§ 3.3 and 3.4.

A *short* version of the forward round function exists which omits the ShuffleCells and MixColumns operations, similarly to the AES final round.

After AddRoundTweakey the tweak T is updated by the function described next.

2.4 The Tweak Update Function

First, the cells of the tweak are permuted as $h(T) = t_{h(0)} \| \cdots \| t_{h(15)}$, where h is the same permutation $h = [\,6, 5, 14, 15, 0, 1, 2, 3, 7, 12, 13, 4, 8, 9, 10, 11\,]$ used in MANTIS.

Then, a LFSR ω updates the tweak cells with indexes 0, 1, 3, 4, 8, 11, and 13. For $m = 4$, ω is a maximal period LFSR that maps cell (b_3, b_2, b_1, b_0) to $(b_0 + b_1, b_3, b_2, b_1)$. For $m = 8$, it maps cell $(b_7, b_6, \ldots, b_0)$ to $(b_0 + b_2, b_7, b_6, \ldots, b_1)$, and its cycles on the non-zero values have all length 15 or 30.

2.5 The Backward Round Function

The *Backward Round Function* $\overline{\mathcal{R}}(\mathrm{IS}; \mathtt{tk})$ is the inverse of the forward round function $\mathcal{R}$. Similarly to the forward round function, a *short backward round function* is defined which omits ShuffleCells and MixColumns.

The tweak update using the inverse LFSR $\overline{\omega}$ and the inverse permutation $\overline{h}$ must be applied before AddRoundTweakey.

2.6 The Central Construction and the Pseudo-Reflector

Two *central rounds* – a forward and a backward one – that use the whitening key instead of the core key, bracket the cipher's *Pseudo-Reflector* $\mathcal{P}(\mathrm{IS}; \mathtt{tk})$, which is

Fig. 3. The QARMA Encryption Algorithm

```
Algorithm QARMA_r Encryption
 1: Write $K = w^0 \| k^0$
 2: $w^1 \hookleftarrow (w^0 \ggg 1) + (w^0 \gg (16\,m - 1))$, $k^1 \hookleftarrow k^0$
 3: $\mathrm{IS} \hookleftarrow P + w^0$
 4: for $i = 0$ to $r - 1$ do
 5:     $\mathrm{IS} \hookleftarrow \mathcal{R}(\mathrm{IS}, k^0 + T + \mathfrak{c}_i)$ (short round version if $i = 0$)
 6:     $T \hookleftarrow \omega \circ h(T)$
 7: $\mathrm{IS} \hookleftarrow \mathcal{R}(\mathrm{IS}, w^1 + T)$
 8: $\mathrm{IS} \hookleftarrow \mathcal{P}(\mathrm{IS}, k^1)$
 9: $\mathrm{IS} \hookleftarrow \mathcal{R}(\mathrm{IS}, w^0 + T)$
10: for $i = r - 1$ down to $0$ do
11:     $T \hookleftarrow \overline{h} \circ \overline{\omega}(T)$
12:     $\mathrm{IS} \hookleftarrow \overline{\mathcal{R}}(\mathrm{IS}, k^0 + T + \mathfrak{c}_i + \alpha)$ (short round version if $i = 0$)
13: $C \hookleftarrow \mathrm{IS} + w^1$
```

essentially just a key addition and a matrix multiplication of the internal state. In more detail, this central construction is defined as follows:

1. A forward round $\mathcal{R}$.
2. The pseudo-reflector $\mathcal{P}(\mathrm{IS}; \mathsf{tk})$ i.e.
   (a) `ShuffleCells`.
   (b) Multiplication of the state by the involutory matrix Q defined in Section 3.1.
   (c) `AddRoundTweakey`. The round tweakey tk is XORed to the state.
   (d) Inverse `ShuffleCells`.
3. A backward round $\overline{\mathcal{R}}$.

It is clear that if steps (b) and (c) were swapped, then tk would have to be replaced with $\overline{Q} \cdot \mathsf{tk} = Q \cdot \mathsf{tk}$ to obtain the same function. Because of this, if tk is the tweakey used during encryption, $M \cdot \mathsf{tk}$ must be used instead to decrypt.

2.7 Putting QARMA Together

The encryption algorithm of $\mathtt{QARMA}_r$ is given in Figure 3. $\mathtt{QARMA}_r$ has $2\,r + 2$ rounds.

The round constants are derived from the expansion of the constant π. For the 64-bit version of QARMA we replace the first block of sixteen digits of the fractional part with zeros and select the seventh block as the α constant, as shown in Table 1 – as a *hommage* to PRINCE. For the 128-bit cipher, instead, we just take the first block of 128 bits in the fractional part of π as the α constant, set $\mathfrak{c}_0 = 0$, and then each $\mathfrak{c}_i$ is a successive block 128 bits of π, as shown in Table 2.

Note that even though the round constants are symmetric, the constant α is always added to the last r backward rounds. This, together with the pseudo-reflector design, prevents perfect symmetry in the data processing part.

Table 1. The Round Constants for the 64-bit Ciphers

α = C0AC29B7C97C50DD	$\mathfrak{c}_0$ = 0000000000000000	$\mathfrak{c}_1$ = 13198A2E03707344	
$\mathfrak{c}_2$ = A4093822299F31D0	$\mathfrak{c}_3$ = 082EFA98EC4E6C89	$\mathfrak{c}_4$ = 452821E638D01377	
$\mathfrak{c}_5$ = BE5466CF34E90C6C	$\mathfrak{c}_6$ = 3F84D5B5B5470917	$\mathfrak{c}_7$ = 9216D5D98979FB1B	$\cdots$

Table 2. The Round Constants for the 128-bit Ciphers

α = 243F6A8885A308D3 13198A2E03707344	$\mathfrak{c}_0$ = 0000000000000000 0000000000000000	
$\mathfrak{c}_1$ = A4093822299F31D0 082EFA98EC4E6C89	$\mathfrak{c}_2$ = 452821E638D01377 BE5466CF34E90C6C	
$\mathfrak{c}_3$ = C0AC29B7C97C50DD 3F84D5B5B5470917	$\mathfrak{c}_4$ = 9216D5D98979FB1B D1310BA698DFB5AC	
$\mathfrak{c}_5$ = 2FFD72DBD01ADFB7 B8E1AFED6A267E96	$\mathfrak{c}_6$ = BA7C9045F12C7F99 24A19947B3916CF7	
$\mathfrak{c}_7$ = 0801F2E2858EFC16 636920D871574E69	$\mathfrak{c}_8$ = A458FEA3F4933D7E 0D95748F728EB658	
$\mathfrak{c}_9$ = 718BCD5882154AEE 7B54A41DC25A59B5	$\mathfrak{c}_{10}$ = 9C30D5392AF26013 C5D1B023286085F0	$\cdots$

2.8 Parameters

For the 64-bit version of QARMA we choose $r = 7$, i.e. 16 rounds, but we believe the cipher to be safe against practical attacks already for $r = 5$, i.e. 12 rounds. For the 128-bit version we choose $r = 11$, i.e. 24 rounds, and believe the cipher safe against practical attacks already for $r = 8$, i.e. 18 rounds. We argue in Section 4 that these parameters offer sufficient security margins.

3 Main Design Decisions

3.1 The Diffusion Matrices

In this section we describe how we constructed and selected the matrices $M = \overline{M} = Q = \mathrm{circ}(0, \rho, \rho^2, \rho)$ for QARMA-64; and $M = \mathrm{circ}(0, \rho, \rho^2, \rho^5)$ (which admits inverse $\overline{M} = \mathrm{circ}(0, \rho^5, \rho^6, \rho)$) with $Q = \mathrm{circ}(0, \rho, \rho^4, \rho)$ for QARMA-128.

3.1.1 Construction QARMA's diffusion layer is composed of a cell permutation and of a matrix multiplication. Its complexity mostly comes from the matrix.

The usual requirements on a diffusion matrix are:

1. It should guarantee mathematically provable good diffusion.
2. It should be as lightweight as possible.

Regarding the first requirement, diffusion is usually measured by the branch number [1], i.e. the smallest nonzero joint number of active inputs and outputs of the matrix. For an invertible $s \times s$ matrix M, the branch number $\mathcal{B}_M$ cannot be greater than $s + 1$. The MDS (Maximum-Distance Separable) matrices are those that attain this maximum: They have long been the preferred choice of block cipher designers, but they tend to lead to expensive implementations. Hence there has been a recent flurry of research on lighter diffusion layers, culminating with those of PRINCE, PRIDE and MIDORI. In particular in MIDORI a $\{0, 1\}$-matrix M – namely a matrix whose entries are in $\{0, 1\}$ – is used that is *Almost MDS*, i.e. with $\mathcal{B}_M = s$.

The second requirement is often understood as *minimising the weight of the matrix*, but for fully unrolled HW implementation a better statement would be minimising the maximum of the weights of all the rows, taking into account the weights of the cells and the underlying algebraic structure.

$\{0,1\}$-Matrices are clearly optimal from this point of view. They are never MDS, but they are Almost MDS in dimensions 2, 3, and 4 [22], and only in these dimensions. However, they may also contribute to the weaknesses of some ciphers. For instance, in the case of `MIDORI`, they permit the propagation of iterative characteristics that are built from the fixed points of the S-Box.

In order to harden against attacks exploiting these properties it is desirable to have diffusion layers that do not help propagate such high likelihood characteristics. More in detail, assuming we are using only one S-Box, if the diffusion layer carries the sum of the outputs of two active S-Boxes unchanged into two different target cells, these cells will carry over the same characteristic twice into the next S-Box layer: the second copy of that characteristic will come for free.

The intuition is that if the S-Box outputs are subject to different linear transformations per each target cell before being added, then the two resulting output characteristics will be different, and at least one will be less likely to propagate through the next S-Box layer. Hence, we look beyond $\{0,1\}$-matrices.

A problem with MDS matrices over binary extension fields is that any multiplication by an element different from 0 and 1 requires a modular reduction step (for a polynomial basis) or expansion step (for a normal basis) which adds latency. So the next logical step is to consider a different underlying algebraic structure, for instance a quotient ring $R_m = \mathbb{F}_2[X]/(X^m+1)$. The multiplication by the image ρ of X in the ring R_m (an element such that $\rho^m = 1$, and thus such that $\{1, \rho, \rho^2, \ldots, \rho^{m-1}\}$ is a basis for R_m as a $\mathbb{F}_2$-algebra) is just a simple circular rotation of the bits (to the left), with only signal propagation latency. Matrices over R_m allow us to easily include rotations in the diffusion layer.

It turns out that, while it seems difficult to construct MDS matrices over generic rings, we can find Almost MDS matrices over the quotient rings $R_m = \mathbb{F}_2[X]/(X^m + 1)$. Since R_m contains zero divisors (for $m \geq 2$), care is to be taken when constructing invertible matrices. We thus restrict ourselves to the following circulants:

$$M = \operatorname{circ}(0, \rho^a, \rho^b, \rho^c) = \begin{pmatrix} 0 & \rho^a & \rho^b & \rho^c \\ \rho^c & 0 & \rho^a & \rho^b \\ \rho^b & \rho^c & 0 & \rho^a \\ \rho^a & \rho^b & \rho^c & 0 \end{pmatrix} \quad . \tag{2}$$

Theorem 1. *Let $R_m = \mathbb{F}_2[\rho]$ be the quotient ring $\mathbb{F}_2[X]/(X^m + 1)$ where ρ is the image of X in R, $m \geq 2$. The Almost MDS matrices M of the form (2) over the ring R_m are precisely the invertible ones, i.e. are those for which $\rho^{4a} + \rho^{4b} + \rho^{4c} \in R_m^*$.*

Proof. Since $\det(M) = \rho^{4a} + \rho^{4b} + \rho^{4c}$, the invertible matrices are those for which this value is invertible, i.e. in R^*. Since Almost MDS matrices are invertible by definition, we need only prove that any M of form (2) has branch number four.

Let us consider a column vector $V = (v_0, v_1, v_2, v_3)^t$. We need to verify when the sum of the weights of $V \neq 0$ and $U := M \cdot V$ is always at least four, so we need to consider the cases where V has weights one, two, and three. Since M is circulant, there are only four distinct cases up to circular permutation of the entries of V:

(i) $V = (v_0, 0, 0, 0)^t$ with $v_0 \neq 0$. Then

$$U = M \cdot V = \begin{pmatrix} 0 & \rho^a & \rho^b & \rho^c \\ \rho^c & 0 & \rho^a & \rho^b \\ \rho^b & \rho^c & 0 & \rho^a \\ \rho^a & \rho^b & \rho^c & 0 \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ v_0\,\rho^c \\ v_0\,\rho^b \\ v_0\,\rho^a \end{pmatrix}$$

and since ρ is not a zero divisor in R, the vector U has weight 3.

(ii) $V = (v_0, v_1, 0, 0)^t$ with $v_0, v_1 \neq 0$. Then

$$U = M \cdot V = \begin{pmatrix} 0 & \rho^a & \rho^b & \rho^c \\ \rho^c & 0 & \rho^a & \rho^b \\ \rho^b & \rho^c & 0 & \rho^a \\ \rho^a & \rho^b & \rho^c & 0 \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} v_1\,\rho^a \\ v_0\,\rho^c \\ v_0\,\rho^b + v_1\,\rho^c \\ v_0\,\rho^a + v_1\,\rho^b \end{pmatrix}$$

and the first two entries of U are nonzero.

(iii) $V = (v_0, 0, v_2, 0)^t$ with $v_0, v_2 \neq 0$. Then

$$U = M \cdot V = \begin{pmatrix} 0 & \rho^a & \rho^b & \rho^c \\ \rho^c & 0 & \rho^a & \rho^b \\ \rho^b & \rho^c & 0 & \rho^a \\ \rho^a & \rho^b & \rho^c & 0 \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ 0 \\ v_2 \\ 0 \end{pmatrix} = \begin{pmatrix} v_2\,\rho^b \\ v_0\,\rho^c + v_2\,\rho^a \\ v_0\,\rho^b \\ v_0\,\rho^a + v_2\,\rho^c \end{pmatrix}$$

and there is nothing to prove also in this case.

(iv) $V = (v_0, v_1, v_2, 0)^t$ with $v_0\, v_1\, v_2 \neq 0$. Then

$$U = M \cdot V = \begin{pmatrix} 0 & \rho^a & \rho^b & \rho^c \\ \rho^c & 0 & \rho^a & \rho^b \\ \rho^b & \rho^c & 0 & \rho^a \\ \rho^a & \rho^b & \rho^c & 0 \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ 0 \end{pmatrix} = \begin{pmatrix} v_1\,\rho^a + v_2\,\rho^b \\ v_0\,\rho^c + v_2\,\rho^a \\ v_0\,\rho^b + v_1\,\rho^c \\ v_0\,\rho^a + v_1\,\rho^b + v_2\,\rho^c \end{pmatrix} = \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

and we need to prove that at least one of the u_i, $0 \leq i < 4$, must be non-zero. Assuming the contrary, we obtain $v_2 = v_1\,\rho^{a-b}$ from $u_0 = 0$ and $v_0 = v_1\,\rho^{-b+c}$ from $u_2 = 0$. Substituting in the relation for $u_3 = 0$ we obtain

$$0 = v_1\,\rho^{a-b+c} + v_1\,\rho^b + v_1\,\rho^{a-b+c} = v_1\,\rho^b$$

whence $v_1 = 0$, a contradiction.

Note that the branch number is tight. For vectors V of weight between one and three and entries v_i which are either 0 or $1+\rho+\rho^2+\ldots+\rho^{m-1}$, it is $\mathsf{wt}(V)+\mathsf{wt}(U)=4$. □

Remark 1. If $m=4$ or 8, then *all* matrices (2) are invertible. In fact, for $m=4$ the determinant $\rho^{4a}+\rho^{4b}+\rho^{4c}$ is always 1 and for $m=8$ it is equal to either 1 or ρ^4.

Now let us consider the matrices of type (2) with equally lightweight inverse.

Theorem 2. *Let $R_m=\mathbb{F}_2[X]/(X^m+1)=\mathbb{F}_2[\rho]$ be defined as in Theorem 1. The Almost MDS matrices $M=\operatorname{circ}(0,\rho^a,\rho^b,\rho^c)$ that admit an inverse of the same form $\overline{M}=\operatorname{circ}(0,\rho^d,\rho^e,\rho^f)$, i.e. with entries of weight at most one, are those that satisfy $a\equiv c+\tau$ where $2\tau\equiv 0 \bmod m$ (for odd m this implies $\tau=0$, for even m it can be $\tau=0$ or $m/2$).*

In this case, the parameters of the matrix $\overline{M}$ are: $d\equiv a-2b$, $e\equiv -b$, and $f\equiv d+\tau \bmod m$.

The involutory matrices M are those for which, additionally, $2b\equiv 0$.

Proof. Since M and $\overline{M}$ are circulant, $M\cdot\overline{M}=I$ is equivalent to

$$\begin{pmatrix}\rho^{a+f}+\rho^{b+e}+\rho^{c+d}\\ \rho^{b+f}+\rho^{c+e}\\ \rho^{a+d}+\rho^{c+f}\\ \rho^{a+e}+\rho^{b+d}\end{pmatrix}=\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \tag{3}$$

Ignoring for the moment the first component in (3), we see that

$$\text{(i)}\ \ b+f\equiv c+e\ \ ,\quad \text{(ii)}\ \ a+d\equiv c+f\ \ ,\quad \text{(iii)}\ \ a+e\equiv b+d \bmod m\ \ .$$

Upon adding relations (i), (ii) and (iii) we get $2a+(d+e+f)=2c+(d+e+f)\equiv 0$, in other words $2(a-c)\equiv 0 \bmod m$, whence $a\equiv c+\tau$ where $2\tau\equiv 0 \bmod m$. Replacing this into (ii) we obtain at once $d\equiv f+\tau$, which is equivalent to $f\equiv d+\tau$. In particular, $a+f\equiv c+d \bmod 0$ holds always, which means that ρ^{a+f} and ρ^{c+d} cancel out in the first relation in (3), and it must be $\rho^{b+e}=1$, i.e. $e\equiv -b \bmod m$. Finally, substituting in relation (iii) we obtain $d\equiv a-2b \bmod m$.

The last statement is proved by substituting $a\equiv d$ into $d\equiv a-2b \bmod m$. □

Remark 2. Hence, *there are two degrees of freedom to define the matrices M, and one extra bit in case m is even. If M has to be involutory, there is only one degree of freedom (and two bits for even m).*

3.1.2 Selection We now describe the selection process for the diffusion matrices.

1. For `QARMA`-64 ($m=4$) we initially restrict our attention to

$$M_{4,1}=Q_{4,1}=\operatorname{circ}(0,\rho,\rho^2,\rho^3)\ \ ,$$

$$M_{4,2} = Q_{4,2} = \mathrm{circ}(0, 1, \rho^2, 1)$$

and

$$M_{4,3} = Q_{4,3} = \mathrm{circ}(0, \rho, \rho^2, \rho) \ .$$

These matrices are all involutory.

2. For `QARMA`-128 ($m = 8$) we consider

$$M_{8,1} = Q_{8,1} = \mathrm{circ}(0, \rho^2, \rho^4, \rho^6) \ ,$$

$$M_{8,2} = Q_{8,2} = \mathrm{circ}(0, \rho^1, \rho^4, \rho^5) \ ,$$

and

$$M_{8,3} = \mathrm{circ}(0, \rho, \rho^2, \rho^5) \quad \text{admitting inverse} \quad \overline{M}_{8,3} = \mathrm{circ}(0, \rho^5, \rho^6, \rho) \ ,$$
$$\text{together with} \quad Q_{8,3} = \mathrm{circ}(0, \rho, \rho^4, \rho) \ .$$

We also define the `MIDORI` circulant $M_0 := \mathrm{circ}(0, 1, 1, 1)$.

We selected among these matrices according to various criteria.

The first criterion is the number of fixed points of the matrix Q, in order to improve the cryptographic properties of the central construction (cf. Section 4.2). From this points of view, optimal involutory matrices over R_4 are $Q_{4,2}$ and $Q_{4,3}$, that have the optimal number of 2^{32} fixed points (cf. Lemma 1 in [23]), and $Q_{8,3}$ that attains the minimum of 2^{64} fixed points. Note that $M_{4,1}$ has 2^{48} fixed points, $M_{8,1}$ has 2^{96} fixed points, and $Q_{8,2}$ has 2^{72} fixed points, which is close to optimal (a proof is found in Appendix B).

The second criterion is the number of active S-Boxes in linear and related-key differential trails. In [24] it is shown how to use mixed integer linear programming (MILP) to count the number of active S-Boxes in these trails. Since we are working on a ring with zero divisors and not over a field, Almost MDS matrices with arbitrary elements do not necessarily have the same diffusion patterns. Hence, we have first determined the diffusion patterns for each matrix, which is a simple and fast exhaustive enumeration over all possible inputs. Our “shortlisted” matrices fall into two different classes:

1. *Class I* includes M_0, $M_{4,1}$ and $M_{8,1}$; and
2. *Class II* includes $M_{4,2}$, $M_{4,3}$, $M_{8,2}$, $M_{8,3}$ and $Q_{8,3}$.

Other matrices belong to these categories but they have excluded because of the limited amounts of differences in relative rotations or because, if involutory, have more fixed points than the selected ones.

Class I matrices have 51 possible column-to-column state transitions, of which the 67 transitions of Class II matrices are a superset. These transitions are depicted in Figures 4 and 5, for Class I and II matrices, respectively. We modelled these two

Table 3. Lower bounds on the number of active S-Boxes

Class I Diffusion Matrices

$r =$	3	4	5	6	7	8	9	10	11
Linear	32	50	64	≥ 70	≥ 76	≥ 82	≥ 100	≥ 114	≥ 124
Rel. Tweak	16	24	34	44	52	60	≥ 64	≥ 70	≥ 78

Class II Diffusion Matrices

$r =$	3	4	5	6	7	8	9	10	11
Linear	32	50	60	≥ 68	≥ 74	≥ 82	≥ 98	≥ 112	≥ 120
Rel. Tweak	14	24	30	42	48	58	≥ 62	≥ 68	≥ 76

Comparison: `MANTIS` (Class I Diffusion Matrix with no ShuffleCells in Reflector)

$r =$	3	4	5	6	7	8
Linear	32	46	60	≥ 68	≥ 76	≥ 82
Rel. Tweak	12	20	34	44	50	56

different behaviours in MILP[1] to compute bounds for the numbers of active S-Boxes in both the linear (and non-related tweak differential) and related-tweak settings. We display these results in Table 3, which have been computed using the `CBC` solver. The results for the linear setting have been computed on the whole cipher for $r \leq 5$ and just half of the cipher (only forward rounds) for $r \geq 6$, in which case the objective value has been doubled. In the related-tweak setting, for $r \geq 9$ we either model only half of the cipher or use a lower bound ("dual bound" for stuck tasks) for the whole cipher, whichever yields a better count.

Since we are reusing the `MIDORI` state shuffle and our matrices have identical or similar state transition patterns to M_0, we did not search for a different `MANTIS` tweak permutation, so it is possible that, for `QARMA`, a different tweak permutation could give improved results. The fact that our bounds are very close to those computer for `MANTIS` (often slightly better because of the improved diffusion in the center, cf. § 4.2, and sometimes slightly lower for Class II matrices) is a confirmation of our approach. Since the bounds on the active S-Boxes between the two matrix classes are very similar, we do not feel that in this situation this criterion must be decisive.

More freedom in the transition space may lead to lower active S-Box counts, as Table 3 shows, hence we we did not consider matrices with even more transitions.

The third criterion is the overall minimisation of the maximum probability of non-trivial differentials on two active cells inside a column through the following operations: an S-Box layer, the diffusion matrix and a second S-Box layer. This is achieved for $M_{4,3}$ and $M_{8,3}$ and for all chosen S-Boxes (see §§ 3.3 and 3.4). We do not report here full tabulates of the resulting data, but to make one important example, for $m = 4$ and with the S-Box σ_0, the average occurrence of a differential

[1] We are grateful to Christof Beierle for sharing his MILP modelling of the Class I transitions. The model of the Class II transitions has been derived from it.

Fig. 4. The Column-wise Active State Transitions for Class I Matrices

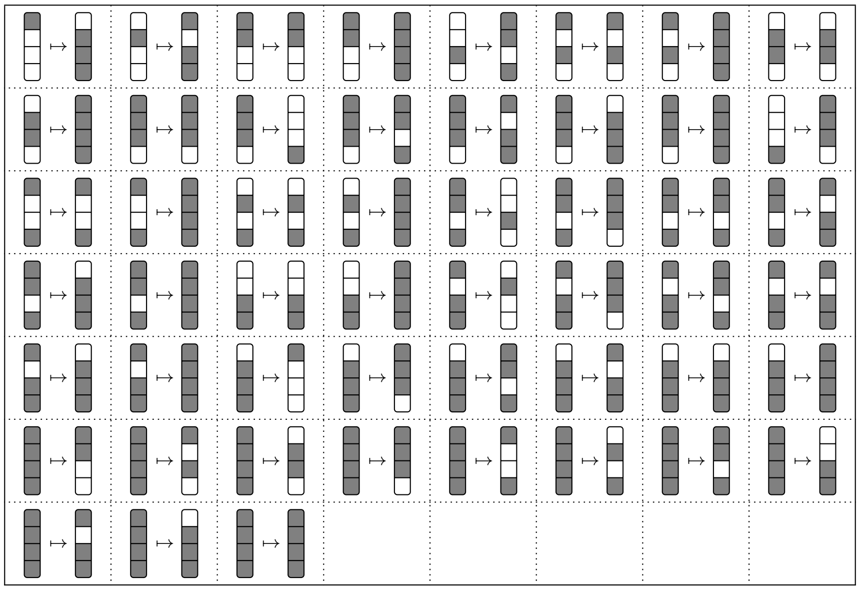
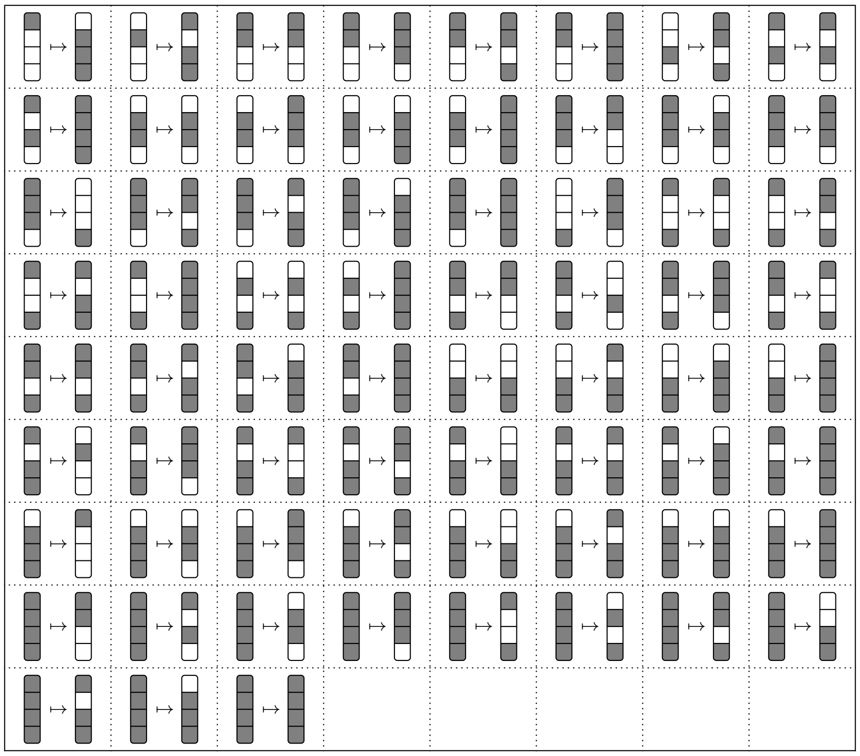


Fig. 5. The Column-wise Active State Transitions for Class II Matrices



goes from 31.6/256 with the MIDORI circulant, to 16.9/256 with $M_{4,1}$, and finally to 8.5/256 with $M_{4,3}$. This proves that the desired goal of increased independence of the outputs, and corresponding disruption of differential characteristics, is achieved.

In light of these observations, *we choose the matrix $M_{4,3} = Q_{4,3}$ for QARMA-64,*[2] *and the matrices $M_{8,3}$ and $Q_{8,3}$ for QARMA-128.*

Remark 3. An analogue of Theorem 1 holds over fields $\mathbb{F}_{2^m}$ as well. A matrix $M = \text{circ}(0, A, B, C) =$ with $ABC \neq 0$ is invertible if and only if its determinant $A^4 + B^4 + C^4 \neq 0$, in which case it is Almost MDS. However, the multiplications by A, B, or C are circuits of non-negligible depth, unless they are equal to 1.

Remark 4. This is not the only work on diffusion matrices on rings other than a finite field. Dehnavi et al. [25,26,27] study matrices over rings $\mathbb{F}_2^m$ where the ring operations are the bitwise XOR and AND. They build MDS matrices whose entries are sums of rotations or shifts. Their research does not consider Almost MDS matrices, and even though their operations are efficient, they are not concerned with minimising the depth. In fact, their diffusion layers have a much higher latency than ours.

3.2 On the Whitened Pseudo-Reflector Construction

Reflection attacks are a specific class of cryptanalysis originally developed against 2K-DES [28] but that can be particularly efficient against PRINCE-like ciphers [23]: If the reflector has too many fixed points, there will be a self-differential with a zero difference after the round key addition at its sides, that may propagate outwards with a palindromic structure if a matching self-differential with a difference of α exists. With respect to MANTIS, by design we took an important step towards resistance against such attacks. The the matrix we chose for QARMA-64 has 2^{32} fixed points out of 2^{64} values, which is the minimum for a linear function (cf. Lemma 1 in [23]), whereas the MIDORI and MANTIS circulant M_0 has 2^{48} (the matrix for QARMA-128 has 2^{64} out of 2^{128}, whereas M_0 has 2^{96} over R_8). But, we decided to take also an additional countermeasure.

Suitable whitening around the reflector can make these attacks less likely, provided that the two whitenings always have a non constant difference as a function of the whitening key. Luckily for us, the PRINCE whitening key expansion was designed this way: for any value of z, equation $w^0 + w^1 = z$ has exactly one solution. Thus, an attacker that does not know w^0 cannot control the difference after the whitening. Since the same tweak is added on both sides, it does not affect the difference.

The central addition of a core key derived round key serves to continue the regular key mixing in the cipher and also to make the pseudo-reflector in general not involutory, with an unpredictable difference at its sides. Since there are values

[2] Note that there is a further, small improvement to 8.3/256 with the matrix $\text{circ}(0, 1, \rho, 1)$ which is Class II, but the difference is so small that we prefer the involutory $M_{4,3}$ instead.

of the core key for which the central rounds are involutory, We intentionally do not add also the tweak in the middle to prevent attacks that could exploit this.

Finally, in light of the cryptanalysis in [29], even though it does not (yet?) lead to practical attacks, we wish to attempt here a departure from pure FX constructions, especially since the used permutations are not ideal: The design of QARMA takes precisely this road. An analysis of the security implications of our central construction can be found in § 4.2.

3.3 Choice of the 4 Bit S-Boxes

Despite the existence of excellent classifications of 4-bit S-Boxes [30,31], choosing the right S-Box for a given application is still a non-trivial task. The cipher's designer must balance on one side desirable cryptographic properties and on the other one performance and cost aspects, such as, in our case, critical path length, then area.

In the case of PRINCE the S-Box was chosen to be 4-bit to reduce total complexity and critical path, but once this decision had been made, the choice seems to have been driven essentially by cryptographic properties. At the opposite side of the spectrum, the MIDORI designers did not hesitate to choose an S-Box with several fixed points, and suboptimal algebraic degree. Indeed, because of the presence of fixed points and minimal round key mixing a vast class of weak keys was found [32].

QARMA can use three S-Boxes: a very small involutory S-Box with complexity similar to that of MIDORI, but with fewer fixed points; an S-Box which is affine equivalent to the S-Box S_6 defined in the PRINCE paper; and a "homogeneous" involutory S-Box whose complexity and critical path fall in between.

A key instrument in choosing these S-Boxes is an automated approach for determining S-Boxes with bounded path delay. The designers of MIDORI observe that *the path delay is highly related to the dependency of the computation.* They introduce a metric called *depth* to estimate the path delay.

Definition 1 (depth). *The depth of a circuit is defined as the sum of sequential path delays of basic operations such as* AND*,* OR*,* NAND*,* NOR*,* XOR*,* XNOR *and* NOT *(the binary operators can be extended to higher arity for this purpose).*

To find the MIDORI S-Boxes, all involutory S-Boxes have been examined in order of increasing depth. This method can be slow and memory intensive, for instance, if all S-Boxes have to be generated first, synthetised and then sorted. We replace these expensive search methodologies by slightly less precise heuristics.

3.3.1 Identifying S-Boxes with Almost Optimal Path Delay Let us consider the MIDORI S-Box Sb_0 as a example. The SOP of the least significant bit of the output corresponds to the boolean function $wy\overline{z} + w\overline{y}z + x\overline{z} + x\overline{y}$ on the variables $\{w, x, y, z\}$ ($\overline{x} := \mathsf{NOT}(x)$). Applying de Morgan's theorem, it is straightforward to see that this can be evaluated as the NAND4 of the four expressions

$\mathsf{NAND3}(w, y, \overline{z})$, $\mathsf{NAND3}(w, \overline{y}, z)$, $\mathsf{NAND2}(x, \overline{z})$, and $\mathsf{NAND2}(x, \overline{y})$. A NAND4 gate can be implemented (with some complication because of wiring) with a depth of 2, and thus by using a layer of NOT's of depth 0.5, a layer with two NAND3's and two NAND2's with a depth of 1.5, and the NAND4 gate, this output bit can be implemented with a depth of 4. However, the SOP of the *negation* of the same output bit is $\overline{x}\overline{y}\overline{z} + \overline{w}\overline{x} + yz$ which can be implemented as the NAND3 of $\mathsf{NAND3}(\overline{x}, \overline{y}, \overline{z})$, $\mathsf{NAND2}(\overline{w}, \overline{x})$, and $\mathsf{NAND2}(y, z)$ – with a total depth of 3.5. Negating it and applying de Morgan's theorem again, the original function can be represented as the NOR3 of $\mathsf{NOR3}(x, y, z)$, $\mathsf{NOR2}(w, x)$, $\mathsf{NOR2}(\overline{y}, \overline{z})$, which, upon closer inspection, has a depth of 3. Similar considerations can be made for the other output bits.

In general, considering the SOP and the NOT-SOP of each output bit we can always match the claimed minimal depth of all the S-Boxes from the literature that we have sampled, or exceed this depth by at most 0.5.

Therefore we sieve the S-Boxes by computing the SOP and NOT-SOP of each output bit by the Quine-McCluskey algorithm, and keep, for instance, first those S-Boxes whose output bits can be all expressed as sums of at most three products, each one having at most weight 3 (or as not-products at most three sums of at most three monomials) *and* satisfying additional cryptographic constraints – and then gradually increasing the complexity of the allowed expressions if no matching S-Boxes are found. By allowing generalised NAND and NOR gates, we get tighter upper bounds for the depth. Gradually, we restrict the cryptographic properties until we get a manageable set of S-Boxes that can be further analysed.

3.3.2 Desirable Cryptographic Properties For various ciphers, there are several interesting cryptographic properties that have been required of S-Boxes, depending on the requirements of the cipher and the structure of other cipher components.

S1 *The maximal probability of a differential is* $1/4$.

S2 *The maximal absolute bias of a linear approximation is* $1/4$.

S3 *Each of the* 15 *non-zero component functions has algebraic degree* 3.

Sometimes the first two properties are strengthened, as in the `PRINCE` S-Boxes, to having exactly 15 differentials with probability $1/4$ and exactly 30 linear approximations with absolute bias $1/4$.

Finally, another important property is the *full diffusion* property:

S4 *Each input bit of the S-Box shall influence each output bit non-linearly.*

This property is easily verified from the SOP of each output bit: each input bit should be present in each output bit in at least one product of weight at least two.

Property **S4** is very important in the design of the second `MIDORI` S-Box, to ensure that each input bit will influence non-linearly all 128 bits of the state after 3-rounds - but it is easy to verify that with our design of the diffusion layer and of the 8-bit S-Box, Property **S4** can be relaxed while still achieving the same result:

S4' *There is at most one input bit that only influences three output bits non-linearly.*

3.3.3 A Lightweight S-Box With Fewer Fixed Points The MIDORI S-Box has 4 fixed points and 3 non-zero component functions of algebraic degree just 2. The bit-flipping pattern is given in Table 4 which implies that in half of the cases a single bit input differential will produce a single bit output differential. The S-Box satisfies Properties **S1**, **S2**, and **S4'**. We aim at improving some of these parameters.

We use the candidate sieving technique described at the end of § 3.3.1 to search the involutory S-Boxes with no more than 4 fixed points. In order to search them all, we modify Prissette's algorithm [33] to enumerate all fixed-point free involutions over a specific set to generate all involutions with a specific subset of fixed points.

We found an alternative with the same depth but only two fixed points, and only one non-zero component function of algebraic degree 2 instead of three. The search lasted only a few seconds on a single core of a 2,4 GHz Intel Core i7 laptop.

Interestingly, we found no S-Box where the values of the two fixed points differ on more than one bit, which implies that in this situation it is important *that the diffusion layer does not always map the bits to the same places and the round constants also influence other bits, to prevent characteristics that map a small subset of possible states onto itself*. This observation is one of the motivations in the design of the linear layer that we described in Section 3.1.

The S-Box we propose for the lightest versions of QARMA, is

$$\sigma_0 := [\,0, 14, 2, 10, 9, 15, 8, 11, 6, 4, 3, 7, 13, 12, 1, 5\,]$$

with fixed points 0 and 2. Its single bit input differential weight distribution is equivalent to that of the MIDORI S-Box, the critical path is the same (depth of 3.5) and it has an area of 14 GE.

We suggest this S-Box also as a possible replacement for the MIDORI S-Box, since the two fixed points do not differ in the bit influenced by the round constants.

3.3.4 A Homogeneous S-Box We found also the following involutory S-Box with no fixed points:

$$\sigma_1 := [\,10, 13, 14, 6, 15, 7, 3, 5, 9, 8, 0, 12, 11, 1, 2, 4\,] \ .$$

The output weight distribution for single bit input differentials is given in Table 5. For each b, each input bits has the same number of one bit to b bit differentials. This may make individual bits less distinguishable from each other.

σ_1 satisfies Properties **S1**, **S2**, **S3** and **S4**, its depth is 4 and the area is 16 GE.

3.3.5 The Lightweight S-Box from the PRINCE family It may be requested that the S-Box be non-involutory. For this purpose, we have filtered all the S-Boxes

Table 4. The Bit-Flipping Pattern of the `MIDORI` S-Box and (up to bit ordering) of σ_0

	Times b bits are flipped			
Flipping bit	one	two	three	four
0	5	2	1	0
1	4	4	0	0
2	5	2	1	0
3	2	4	2	0

Table 5. The Bit-Flipping Pattern of the σ_1 S-Box

	Times b bits are flipped			
Flipping bit	one	two	three	four
0	3	4	1	0
1	3	4	1	0
2	3	4	1	0
3	3	4	1	0

Table 6. The Bit-Flipping Patterns of the σ_2 S-Box and its Inverse

σ_2	Times b bits are flipped			
Flipping bit	one	two	three	four
0	2	1	4	1
1	2	4	2	0
2	2	3	2	1
3	6	2	0	0

$\overline{\sigma}_2$	Times b bits are flipped			
Flipping bit	one	two	three	four
0	5	1	1	1
1	2	3	2	1
2	2	5	0	1
3	3	2	3	0

allowed for `PRINCE`, which fall into eight affine equivalence classes. These S-Boxes all satisfy the strengthened version of Properties **S1** and **S2**, and Properties **S3** and **S4**.

In fact, we selected an S-Box which is affine equivalent to the S-Box S_6 as defined in the `PRINCE` paper, because it and its inverse can be implemented with depths of 4.5 and 4 respectively, which seem to be optimal among all `PRINCE` S-Boxes according to the bounds provided by our search tools. These depths are smaller than those required by the default `PRINCE` S-Box, but the area is slightly larger (approximately 20 GE). This S-Box and its inverse are:

$$\begin{aligned}\sigma_2 &= [\,11, 6, 8, 15, 12, 0, 9, 14, 3, 7, 4, 5, 13, 2, 1, 10\,]\\ \overline{\sigma}_2 &= [\,5, 14, 13, 8, 10, 11, 1, 9, 2, 6, 15, 0, 4, 12, 7, 3\,]\ .\end{aligned}$$

All the non trivial component functions have algebraic degree 3. The S-Box has no fixed points and the output weight distribution for single bit input differentials for both σ_2 and $\overline{\sigma}_2$ is given in Table 6.

3.4 The 8-bit S-Box

As in `MIDORI`-128 we construct an 8-bit S-Box Σ by placing two instances of a single 4-bit S-Box in parallel. However, we wire the input and output bits in a single and simpler way, as shown in Figure 6, which is asymmetric. The S-Box σ is one of the S-Boxes σ_i described in § 3.3. If we write a 8-bit cell of the state as $(x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$, σ is applied to (x_7, x_6, x_5, x_4) producing the output bits (x'_7, x'_5, x'_3, x'_1), and to (x_3, x_2, x_1, x_0) producing the output bits (x'_6, x'_4, x'_2, x'_0), and the output of the combined 8-bit S-Box is $(x'_7, x'_6, x'_5, x'_4, x'_3, x'_2, x'_1, x'_0)$. Since the construction is not symmetric, the opposite wiring must be implemented for $\overline{\Sigma}$.

The rationale behind this is that in combination with the chosen matrices M, if the output of Σ is cyclically rotated by any amount of bits and then fed into

Fig. 6. The Construction of the 8-bit S-Box Σ of QARMA-128

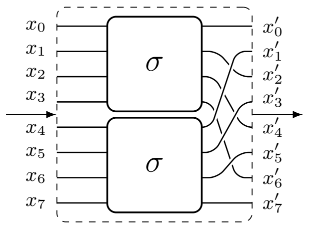
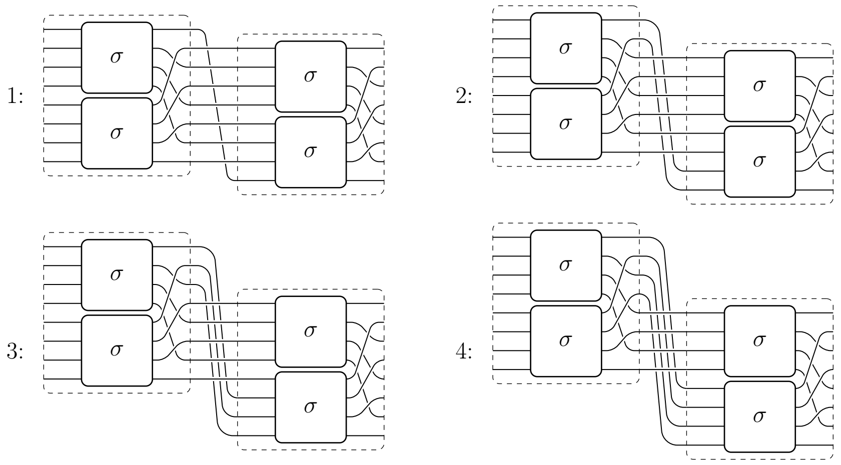


Fig. 7. Alignment of Output and Input Bits of Consecutive Instances of the 8-bit Composite S-Box



another instance of Σ, exactly 2 bits of the output of one of the two 4-bit S-Boxes in the first Σ are wired into each of the two 4-bit S-Boxes of the second instance of Σ. This is clear because all the outputs of one instance of σ will be wired into the even numbered bits of the output of Σ, and the outputs of the other instance of σ into the bits in the odd numbered positions of the output of Σ, so any group of four cyclically adjacent output bits of Σ will be equally partitioned among the two sources. This is also graphically depicted in Figure 7.

If the 4-bit S-Box σ satisfies Property **S4**, a 3-round full diffusion property as in MIDORI-128 (Theorem 1 in [9]) still holds, namely *any input bit nonlinearly affects all 128 bits of the state after 3-rounds* The 3-round full diffusion property holds even under weaker Property **S4'**. The proof is entirely similar, one only needs to add and track a cell state where only three bits are non-linearly influenced.

3.5 On the Round Constants

We did not consider sparse constants: since they are hardwired, and XOR and XNOR have the same costs, XORing together the key, generating the round tweakey has the same cost regardless of the Hamming weight of the constants.

3.6 The ω Function

The indexes of the tweak register modified by ω have been chosen along the length 14 cycle of the tweak cell permutation - and one along the length 2 cycle $(1, 5)$ - in order to minimise the number of cell values with the same difference to the corresponding cells of the round constants (and so make partial slide attacks less likely) and maximise the spread of different values if one starts with a non-zero tweak with all equal cells.

4 Security Analysis

4.1 Generic Attacks on Block Ciphers

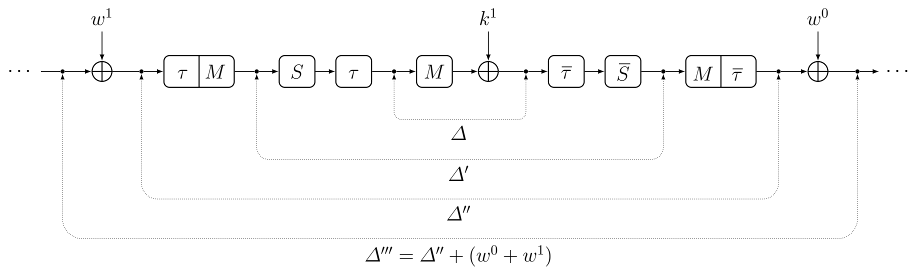
One of the main goals in our design was to be able to carry over the security analysis of MIDORI as verbatim as possible while at the same time improving diffusion characteristics. For this reason, the forward rounds have a very similar structure to the MIDORI rounds, the minor differences allowing us to achieve the desired diffusion goals through the pseudo-reflector (cf. § 4.2). For instance, as already mentioned, MIDORI-128's 3-round full diffusion property can be proved also for QARMA-128.

Looking at Table 3, we see that no related tweak linear or differential distinguisher based on a characteristic is possible for QARMA-64 already when $r = 5$, and note that not only the bounds are not tight (even in the case where the whole cipher is modelled, and not only the forward rounds, the bounds would be only tight if the S-Boxes could be chosen freely for each cell and round), but they express a security level before considering the presence of whitening. $\texttt{QARMA}_7$-64 has two more rounds, which should provide a sufficient security margin.

QARMA-128's state can be viewed as thirty-two 4-bit cells, but we did not find this partition useful for a security analysis. The main reason is that the 8-bit S-Box construction Σ (cf. Figure 6), has branch number 2 viewed as a map from two 4-bit cells (x_7, x_6, x_5, x_4) and (x_3, x_2, x_1, x_0) to two 4-bit cells $(x'_{i+7}, x'_{i+6}, x'_{i+5}, x'_{i+4})$ and $(x'_{i+3}, x'_{i+2}, x'_{i+1}, x'_{i+0})$ for any value of i with the indexes taken mod8. also when combined with the circular rotations. Hence, if the input is active, only one of the two 4-bit outputs is guaranteed to be active. Heuristically, after the first round we can estimate that if one of the two 4-bit outputs in a 8-bit cell is active, then the other output is active with a likelihood of 75%. This which tells us that the actual active S-Box counts are most likely much larger than the counts given in Table 3, and for this reason we did not require $r = 12$ to assume security against linear and differential attacks for QARMA-128. However, a bound along these lines can probably only be proved by converting the cell-wise MILP model into a bit-wise one, which would be computationally infeasible, except for a very small number of rounds.

Against boomerang attacks, impossible differential attacks, and meet-in-the-middle attacks, the arguments for MIDORI apply essentially unchanged.

Fig. 8. Reflection Self-Differentials



Resistance agains invariant subspace attacks similar to the attack against MIDORI-64 is ensured by the fact that the round constants affect with likelihood 1/2 each output bit of an S-Box and by the rotations embedded in the diffusion matrix.

We believe that the use of essentially random constants and the heterogeneity of round types break any self similarities that could be exploited in slide attacks.

4.2 Security Implications of the Central Construction

It is tempting to dismiss the possibility of reflection attacks or attacks on Even-Mansour schemes with involutions simply on the basis that the central rounds do not constitute an involution. However, the central rounds still shows high likelihood self-differentials that depend on k^1, warranting a closer analysis.

Note that, besides the whitening and the central key addition, a further crucial difference with respect to MANTIS is the use of two additional ShuffleCells layers. Without ShuffleCells the central group of state transformations formed by M, S, M, addition of k^1, S and M layers would only act independently on four partitions of the state, each consisting of four cells. Indeed, with ShuffleCells, better diffusion is achieved: any single bit of the state on one side of the central construction affects non-linearly at least four bits in each of twelve cells on the opposite side, as opposed to just four cells without ShuffleCells – and it affects non-linearly all bits of the state after just one more round, as opposed to all bits in just twelve of sixteen cells.

A tangible effect is that the active S-Box counts in Table 3 for both Class I and II matrices are often larger by a count of 2 or 4 than if we evaluate them without modelling the two ShuffleCells operations. (Indeed, if we use MILP to count the number of linearly or non-related-tweak differentially active S-Boxes through the two rounds before and the two rounds after the pseudo-reflector we verify that we have at least sixteen active S-Boxes with the ShuffleCells and only eight without. This result holds for both Class I and Class II matrices. This means that the two sub-ciphers of QARMA are better "coupled" by the central rounds with ShuffleCells.)

However, the most important security improvement is regarding (analogues of) reflection attacks. We use Figure 8 as a reference in the following.

The pseudo-reflector *is* an involution if and only if the key k^1 is a fixed point of M (so there are 2^{32} such values for QARMA-64. resp. 2^{64} for QARMA-128), and if $\Delta = 0$ then also $\Delta'' = 0$, but at this point the formation of reflection characteristics is thwarted by the use of the whitening keys in the central rounds, unless $w^0 + w^1 = \alpha$ or 0, which happens for exactly two values of w^0. We conclude that in this setting for only one key K in 2^{95} (for QARMA-64, resp. 2^{191} for QARMA-128) a reflection attack could be attempted: In the case of iterative characteristics, if the palindromic structure begins with a $\Delta''' = \alpha$, the successive differences are 0, then α again, and so on; if the structure begins with a $\Delta''' = 0$, the successive differences are then α, 0, and so on. At the last round the attacker will distinguish the two cases if she succeeds in getting a final difference of 0 or α. Then, the considerations in [23] would apply, for which, in the case of QARMA$_7$-64, we believe similar attack complexities as in PRINCE to hold (and correspondigly higher complexities would hold for QARMA$_{11}$-128).

If k^1 is not a fixed point of M, i.e. $k^1 \in \ker(M + I)$, then the pseudo-reflector cannot have fixed points, since every such fixed point z would satisfy both $z = M \cdot z + k_1$ and $z = M \cdot z + M \cdot k_1$, which is a contradiction. So attacks based on $\Delta = 0$ cannot be mounted. But this cannot exclude other types of characteristics.

In fact, for any fixed k^1, the self-differential Δ can take only 2^{32} (resp. 2^{64}) equiprobable values, which depend only on the class $k^1 + \mathrm{Im}(M + I)$ of k^1 in $R^{16}/\,\mathrm{Im}(M+I)$. In fact, if $x+Mx+k^1 = \Delta$, then $(x+\ell)+M(x+\ell)+(k^1+(M+I)\ell) = \Delta$, and there is a 1-1 correspondence $x \leftrightarrow x + \ell$ between the sets of values for which this difference holds for the two keys.

These values Δ can then propagate to values Δ'' with some probability, and for each such Δ'' there are two value of w^0 for which $\Delta''' = \Delta'' + (w^0 + w^1) = \alpha$ or 0. This leads to analogues of attacks based on the second type of middle-round characteristic in [23]. Now, we see that for each core key, text and tweak, there are two whitening keys in 2^{64} (resp. 2^{128}) that allow a reflection attack to be bootstrapped. So an attacker would attempt to change texts and tweaks hoping that the correct differential Δ'' is hit that leads to $\Delta''' = \alpha$ or 0, and each time verifying whether the differential holds. Unless k^1 is unknown, no assumption on Δ can be done.

The improvement with respect to the attacks on reflection ciphers is that the likelihood of a useful differential Δ''' decreases from 2^{-32} (resp. 2^{-64}) to 2^{-63} (resp. 2^{-127}) and so we get correspondingly reduced success likelihoods - the final probabilities are not as small as our estimates above regarding the $\Delta = 0$ case (in fact, we obtain 2^{-107} and 2^{-189}, respectively), but most likely with data requirements (upwards of 2^{60} for QARMA-64, for instance) that will not compromise the desired level of tradeoff security. Deriving more precise complexity estimations is definitely an important area of research.

Also, the impossibility to control the self-differentials when k^1 and/or w^0 are unknown should prevent analysis based on impossible reflection characteristics.

4.3 Attacks on Even-Mansour Schemes

The cryptanalysis described in [34] seems unlikely as well: The single-key 3-round with an involution attack is the one that seems more applicable to our design. It can be adapted at once observing that, for each fixed core key, the mapping $x \mapsto \Delta$ assumes only $2^{n/2}$ values which occur $2^{n/2}$ times each (w.r.t. the notation in [34], this is in-degree t). This leads to a time/data/memory ($T/D/M$) tradeoff of $TD = 2^{3\,n/2}$, where the data consists of known texts and memory is online storage. The attack has to be repeated for each candidate core key in order to determine the whitening key as well, so we obtain $T = 2^{7\,n/4}$, $D = 2^{3\,n/4}$ and $M = D$. Since evaluations of the sub-ciphers can be done for pairs of core keys $(k^0, k^0 + \alpha)$, the memory usage can be halved. For QARMA-64 this turns into an attack with 2^{112} time and 2^{47} data.

The single-key 2-round attack can also be applied, under the assumption that for a known core key, a known value Δ will occur with likelihood $2^{-n/2}$ – in other words, for each $2^{n/2}$ known or chosen plaintexts it can be expected that this Δ will occur once (but choosing the plaintext does not seem to give control on this event). Note that in this case the cipher collapses into a single-key 2-round EM construction, not a single round EM, because the values of the central key $w^0 + w^1$ are 1-to-1 with w^0 – an FX-construction like PRINCE or MANTIS would collapse into a single round scheme. For each key, time complexity is slightly smaller than 2^n, data (known texts) is slightly smaller than $2^{3\,n/4}$, and memory is around $2^{n/4}$. The time complexity must be multiplied by 2^n to cover all core keys, and the data by $2^{n/2}$ because of the usable proportion, whereas the online memory usage stay the same. Hence, we do get an attack which is slightly better than brute force, but with $TD \sim 2^{14\,n/4}$.

These estimates are all better than the corresponding complexities for attacks on the simpler FX construction used, say, in PRINCE.

Similarly, for the attacks in [35], with the same likelihoods for a known central difference Δ for a certain core key (resp. class), the equations to solve for the whitening key (and possibly the remaining bits of the core key) would still correspond to the whole cipher minus the central construction, so we do not expect it to be significantly easier than attempting to exploit reflection characteristics.

4.4 Security Claims

Similarly to MANTIS and PRINCE, for QARMA_7-64 and QARMA_{11}-128, we claim that they attain n bits of tradeoff security. In fact, we claim security against practical attacks already for QARMA_5-64, resp. QARMA_8-128: more precisely, no related-tweak attack should be applicable with less than 2^{30} (resp. 2^{60}) chosen or 2^{40} (resp. 2^{80}) known text pairs.

Regarding QARMA-128, if a 192-bit security level is required, we believe that QARMA_9-128 offers it with a substantial security margin. For a 128-bit security level, QARMA_8-128 is a good choice because of the better performance.

Table 7. Area and Critical Path Counts for QARMA-64 and QARMA-128

Cipher	Depth (GE)	Area (GE)
QARMA-64_5-σ_0	100	8971
QARMA-64_6-σ_0	117	10451
QARMA-64_7-σ_0	134	11929
QARMA-64_5-σ_2	107	9484
QARMA-64_6-σ_2	125	11048
QARMA-64_7-σ_2	143	12616
MANTIS$_5$	100	8703
MANTIS$_6$	117	10155
MANTIS$_7$	134	11605
PRINCE	114	7424

Cipher	Depth (GE)	Area (GE)
QARMA-128_8-σ_0	152	26592
QARMA-128_9-σ_0	168	29521
QARMA-128_{10}-σ_0	185	32450
QARMA-128_{11}-σ_0	201	35379
QARMA-128_8-σ_2	164	28127
QARMA-128_9-σ_2	183	31228
QARMA-128_{10}-σ_2	201	34328
QARMA-128_{11}-σ_2	219	37429
AES-128	554	63234
(Encryption only)	294	143888

5 Implementation Considerations

The most important parameter in implementations of QARMA is the latency, which is strictly correlated to the length of the critical path, usually measured in NAND/NOR gate equivalents (GE). We report the latter for QARMA and for some other ciphers, together with the area estimates, also given in GEs, in Table 7.

Striving for a fair comparison with the results in the literature on lightweight block ciphers, we have done pure gate counts for fully unrolled designs, instead using a very specific process. For instance, using the Virtual Silicon 0.18μm VIP Standard Cell Libraries with the Synopsys DesignCompiler is quite common but not ubiquitous, and at the same time in a real world context much finer processes are used and lower latencies achieved. From the values we report it is easy to extrapolate a good approximation of area and latency on a target platform if a similar (i.e. fully unrolled) implementation of PRINCE or MANTIS is available.

We applied several implementation tricks to optimise, fuse, or parallelise various steps. For instance, the critical path for MANTIS and QARMA-64 (with the S-Box σ_0) is the same, since the key addition in the pseudo-reflector can be fused with the previous multiplication of the state by M, which thus maintains a critical path of 2 XOR gates. The same optimisation applies to the backward rounds of both ciphers.

The increased area of QARMA-64 with respect to MANTIS mostly comes from the LFSR on the tweak, the heavier S-Boxes, and the additional middle key mixing. The largest difference between these two ciphers on one side and PRINCE on the other comes from the mixing of key and tweak. This amounts to 256 GE per round.

To model the AES, we use the parameter of a Canright type [36] S-Box implementation with a depth of 23 GE and a total area of 285 GE (this is a unified S-Box for encryption and decryption, which is a fair comparison since all other ciphers accomodate both operations). It is possible to further reduce the latency but at large price in area. For instance, using a hybrid LUT approach [37] the S-Box can be implemented with a critical path of about $\sim$10 GE for an area of $\sim$700 GE: we used these parameters to model the AES and provide different estimates. Even more extreme AES implementations exist: for instance in [7] implementations that require more than 400K gates are reported.

6 Conclusions and Open Questions

We have introduced QARMA, a new lightweight TBC family that comes in 64 and 128 bit block and tweak sizes, aimed at 128 and 256 bit levels of security, respectively.

The design of QARMA uses diffusion layers based on Almost MDS matrices over the ring $R_m = \mathbb{F}_2[\rho] = \mathbb{F}_2[X]/(X^m+1)$, in order to embed different rotation amounts for each input and output cell in its operations. These matrices offer the same latency as $\{0,1\}$-matrices while at the same time offering a better barrier to the simultaneous propagation of characteristics across several active cells in the same round. The confusion layer has been selected using new heuristics for efficiently sieving low-latency S-Boxes. The structure is a single-key (actually, single-key derived *keys*) 3-round EM scheme designed to collapse to a 2-round EM (instead of a single round EM) when one of the permutations can be erased with some probability because of high likelihood self-differentials (including fixed points): This leads to higher attack complexity estimates than for FX constructions.

The cipher's security has been analysed and we believe it offers reasonable security margins with the recommended numbers of rounds.

We believe that QARMA can spur research into the analysis of reflection-like differentials in ciphers that are symmetric around a non-involutory structure. Another interesting open question is how to modify current MILP models for counting active S-Boxes in order to attain better bounds on the number of the active halves in QARMA-128's (and MIDORI-128's) 8-bit S-Boxes – without having to resort to bit-wise models which quickly become impractical.

Acknowledgements. We thank the authors of [19] for the cooperation on MANTIS and useful comments. We also thank Can Acar, Antonio Cardoso Costa, Orr Dunkelman, Meltem Sönmez Turan, Robert Turner, Richard Grisenthwaite, Rene Peralta, and Pouyan Sepehrdad for fruitful discussions. We are grateful to Brian Rosenberg, Renwei Ge and Alex Gantman for their support. We are indebted to the Twitter account @FakeIACR for good laughs during boring programming sessions.

References

1. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography. Springer (2002) http://dx.doi.org/10.1007/978-3-662-04722-4.
2. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-Bit blockcipher CLEFIA. In Biryukov, A., ed.: FSE 2007. Volume 4593 of Lecture Notes in Computer Science., Springer (2007) 181–195

3. Cannière, C.D., Dunkelman, O., Knezevic, M.: KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers. In Clavier, C., Gaj, K., eds.: CHES 2009. Volume 5747 of Lecture Notes in Computer Science., Springer (2009) 272–288
4. Gong, Z., Nikova, S., Law, Y.W.: KLEIN: A New Family of Lightweight Block Ciphers. In Juels, A., Paar, C., eds.: RFIDSec 2011. Volume 7055 of Lecture Notes in Computer Science., Springer (2011) 1–18
5. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.J.B.: The LED Block Cipher. IACR Cryptology ePrint Archive **2012** (2012) 600
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 Paillier, P., Verbauwhede, I., eds.: CHES 2007. Volume 4727 of Lecture Notes in Computer Science., Springer (2007) 450–466
7. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A Low-latency Block Cipher for Pervasive Computing Applications (Full version). IACR Cryptology ePrint Archive **2012** (2012) 529
8. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, `http://eprint.iacr.org/2013/404` (2013)
9. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A Block Cipher for Low Energy. In Iwata, T., Cheon, J.H., eds.: Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II. Volume 9453 of Lecture Notes in Computer Science., Springer (2015) 411–436
10. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable Block Ciphers. In Yung, M., ed.: CRYPTO 2002. Volume 2442 of Lecture Notes in Computer Science., Springer (2002) 31–46
11. Martin, L.: XTS: A mode of AES for encrypting hard disks. IEEE Security & Privacy **8** (2010) 68–69
12. Henson, M., Taylor, S.: Memory encryption: A survey of existing techniques. ACM Comput. Surv. **46** (2013) 53:1–53:26
13. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In Lee, P.J., ed.: Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings. Volume 3329 of Lecture Notes in Computer Science., Springer (2004) 16–31
14. Landecker, W., Shrimpton, T., Terashima, R.S.: Tweakable Blockciphers with Beyond Birthday-Bound Security. In Safavi-Naini, R., Canetti, R., eds.: Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. Volume 7417 of Lecture Notes in Computer Science., Springer (2012) 14–30
15. Shrimpton, T., Terashima, R.S.: A Modular Framework for Building Variable-Input-Length Tweakable Ciphers. [38] 405–423
16. Crowley, P.: Mercy: A fast large block cipher for disk sector encryption. In Schneier, B., ed.: Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings. Volume 1978 of Lecture Notes in Computer Science., Springer (2000) 49–63
17. Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., Walker, J.: The Skein Hash Function Family. `http://www.skein-hash.info` (2010)
18. Jean, J., Nikolic, I., Peyrin, T.: Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In Sarkar, P., Iwata, T., eds.: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II. Volume 8874 of Lecture Notes in Computer Science., Springer (2014) 274–288

19. omitted: omitted (2016)
20. Gueron, S.: Intel's SGX Memory Encryption Engine (Slides). `https://drive.google.com/file/d/0Bzm_4XrWnl5zOXdTcUlEMmdZem8/view` (2016)
21. NIST: Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program. Technical report, National Institute of Standards & Technology, Gaithersburg, MD, United States (January, 2016)
22. Choy, J., Khoo, K.: New applications of differential bounds of the SDS structure. In Wu, T., Lei, C., Rijmen, V., Lee, D., eds.: Information Security, 11th International Conference, ISC 2008, Taipei, Taiwan, September 15-18, 2008. Proceedings. Volume 5222 of Lecture Notes in Computer Science., Springer (2008) 367–384 Corrected version at `http://eprint.iacr.org/2008/395`.
23. Soleimany, H., Blondeau, C., Yu, X., Wu, W., Nyberg, K., Zhang, H., Zhang, L., Wang, Y.: Reflection cryptanalysis of prince-like ciphers. J. Cryptology **28** (2015) 718–744
24. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In Wu, C., Yung, M., Lin, D., eds.: Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers. Volume 7537 of Lecture Notes in Computer Science., Springer (2011) 57–76
25. Dehnavi, S.M., Rishakani, A.M., Shamsabad, M.R.M., Maimani, H., Pasha, E.: Construction of new families of mds diffusion layers. Cryptology ePrint Archive, Report 2014/011 (2014) `http://eprint.iacr.org/`.
26. Dehnavi, S.M., Rishakani, A.M., Shamsabad, M.R.M.: Bitwise linear mappings with good cryptographic properties and efficient implementation. Cryptology ePrint Archive, Report 2015/225 (2015) `http://eprint.iacr.org/`.
27. Rishakani, A.M., Dehnavi, S.M., Shamsabad, M.R.M., Maimani, H., Pasha, E.: New concepts in design of lightweight mds diffusion layers. 11th International ISC Conference on Information Security and Cryptology, 2014 (2014) DOI: 10.1109/ISCISC.2014.6994017.
28. Kara, O.: Reflection attacks on product ciphers. Cryptology ePrint Archive, Report 2007/043 (2007) `http://eprint.iacr.org/2007/043`.
29. Dinur, I.: Cryptanalytic time-memory-data tradeoffs for fx-constructions with applications to PRINCE and PRIDE. In Oswald, E., Fischlin, M., eds.: Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. Volume 9056 of Lecture Notes in Computer Science., Springer (2015) 231–253
30. Leander, G., Poschmann, A.: On the Classification of 4 Bit S-Boxes. In Carlet, C., Sunar, B., eds.: Arithmetic of Finite Fields, First International Workshop, WAIFI 2007, Madrid, Spain, June 21-22, 2007, Proceedings. Volume 4547 of Lecture Notes in Computer Science., Springer (2007) 159–176
31. Saarinen, M.O.: Cryptographic Analysis of All 4 $\times$ 4-Bit S-Boxes. In Miri, A., Vaudenay, S., eds.: Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers. Volume 7118 of Lecture Notes in Computer Science., Springer (2011) 118–133
32. Guo, J., Jean, J., Nikolic, I., Qiao, K., Sasaki, Y., Sim, S.M.: Invariant Subspace Attack Against Full Midori64. IACR Cryptology ePrint Archive **2015** (2015) 1189
33. Prissette, C.: An Algorithm to List All the Fixed-Point Free Involutions on a Finite Set. `http://arxiv.org/pdf/1006.3993v1.pdf` (2010)
34. Dinur, I., Dunkelman, O., Keller, N., Shamir, A.: Key Recovery Attacks on 3-round Even-Mansour, 8-step LED-128, and Full AES^2. [38] 337–356
35. Dinur, I., Dunkelman, O., Keller, N., Shamir, A.: Reflections on slide with a twist attacks. Des. Codes Cryptography **77** (2015) 633–651
36. Canright, D.: A very compact s-box for AES. In Rao, J.R., Sunar, B., eds.: Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK,

August 29 - September 1, 2005, Proceedings. Volume 3659 of Lecture Notes in Computer Science., Springer (2005) 441–455

37. Tillich, S., Feldhofer, M., Popp, T., Großschädl, J.: Area, delay, and power characteristics of standard-cell implementations of the AES s-box. Signal Processing Systems **50** (2008) 251–261
38. Sako, K., Sarkar, P., eds.: Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I. In Sako, K., Sarkar, P., eds.: ASIACRYPT 2013. Volume 8269 of Lecture Notes in Computer Science., Springer (2013)

A Test Vectors

A.1 Test Vectors for QARMA-64 with $M = M_{4,1}$

```
P   = fb623599da6e8127    T   = 477d469dec0b8762
w0  = 84be85ce9804e94b    k0  = ec2802d4e0a488e9
```

Using the S-Box σ_0.

```
QARMA64_5
C = de5b629abbfcde27

QARMA64_6
C = dd117a6e8a55b338

QARMA64_7
C = 6faabc460bc8a784
```

Using the S-Box σ_1.

```
QARMA64_5
C = eb464d057108eb0e

QARMA64_6
C = 18c7547b7af08e42

QARMA64_7
C = 376f170678c80c7c
```

Using the S-Box σ_2.

```
QARMA64_5
C = d9f84b6d6b06c7c1

QARMA64_6
C = b6fbe247dd031812

QARMA64_7
C = bac6be9ee3d9e519
```

A.2 Test Vectors for QARMA-64 with $M = M_{4,3}$

```
P   = fb623599da6e8127    T   = 477d469dec0b8762
w0  = 84be85ce9804e94b    k0  = ec2802d4e0a488e9
```

Using the S-Box σ_0.

```
QARMA64_5
C = 3ee99a6c82af0c38

QARMA64_6
C = 9f5c41ec525603c9

QARMA64_7
C = bcaf6c89de930765
```

Using the S-Box σ_1.

```
QARMA64_5
C = 544b0ab95bda7c3a

QARMA64_6
C = a512dd1e4e3ec582

QARMA64_7
C = edf67ff370a483f2
```

Using the S-Box σ_2.

```
QARMA64_5
C = c003b93999b33765

QARMA64_6
C = 270a787275c48d10

QARMA64_7
C = 5c06a7501b63b2fd
```

A.3 Test Vectors for QARMA-128

```
P   = 2fdbb6a2c395e959 fdfa964e98c1a2e7    T   = 242767fd3486a96d fe9c904be82756a2
w0  = 58948b7ef8e5ec7e f9f8f014de0924eb    k0  = 4e7ce2081002eda4 fe20aaa4d868be09
```

Using the S-Box σ_0.

```
QARMA128_8
C = 950e8c0cbfa6a71f dc002c334f66e1f2

QARMA128_9
C = 4f9945f78bc95fdf 2b7ef9eb85faa032

QARMA128_10
C = be8343304ec23a37 7c73e199b0e37cf8
```

Using the S-Box σ_1.

```
QARMA128_8
C = 4d20b727c1e13be2 e2eea20e4c5a40fd

QARMA128_9
C = c68d7fb1e329830c d29b626efa4015f0

QARMA128_10
C = c155ffbbac8505e8 790f72a36088dec5
```

Using the S-Box σ_2.

```
QARMA128_8
C = 9bd498332f494d78 b39db929426ddb44

QARMA128_9
C = 7564cddcf60ecc04 bae52c17e31140ca

QARMA128_10
C = b52d6eac88dc5c76 bea485559b7d20d5
```

B Proof that $M_{8,2}$ has 2^{72} fixed points

Proposition 1. *The matrix* $M = \mathrm{circ}(0, \rho, \rho^4, \rho^5)$ *over* $R_8 = \mathbb{F}_2[\rho] = \mathbb{F}_2[X]/(X^8 + 1)$ *used in* QARMA$_{128}$ *has* 2^{72} *fixed points.*

Proof. Since M acts on the four columns of the state independently, it will suffice to establish that the number of fixed points of M acting on R^4, i.e. on columns, is 2^{18}. A vector $V = (v_0, v_1, v_2, v_3)^t$ is a fixed point of M if and only if

$$(M + I) \cdot V = \begin{pmatrix} v_0 + v_1\,\rho + v_2\,\rho^4 + v_3\,\rho^5 \\ v_0\,\rho^5 + v_1 + v_2\,\rho + v_3\,\rho^4 \\ v_0\,\rho^4 + v_1\,\rho^5 + v_2 + v_3\,\rho \\ v_0\,\rho + v_1\,\rho^4 + v_2\,\rho^5 + v_3 \end{pmatrix} = \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

is zero. Now, $0 = u_0 + v_1\,\rho$, we get $v_0(1 + \rho^6) + v_2(\rho^2 + \rho^4) = 0$ or, in other words

$$(1 + \rho^2)(v_0 + v_2\,\rho^4) = 0$$

which means that v_0 is equal to $v_2\,\rho^4$ up to an element annihilated by $1 + \rho^2$, i.e.

$$v_0 = v_2\,\rho^4 + (1 + \rho^2 + \rho^4 + \rho^6)\,\zeta \ .$$

Similarly,

$$v_1 = v_3\,\rho^4 + (1 + \rho^2 + \rho^4 + \rho^6)\,\zeta' \ .$$

Note that ζ and ζ' are effectively defined modulo $1 + \rho^2$, and they can only define four distinct values of v_0 (resp. v_1) for each fixed value of v_2 (resp. v_3). Substituting these expressions for v_0 and v_1 in $u_0 = 0$ (or $u_2 = 0$) we get

$$(1 + \rho^2 + \rho^4 + \rho^6)(\zeta + \rho\,\zeta') = 0 \ .$$

We see that for each value of ζ there is a unique ζ' (modulo $1 + \rho^2$) such that the last equation is satisfied. Using u_1 (or u_3) in place of u_0 or u_2 leads to the same result. The elements in $\ker(M+I) \subseteq R^4$ are thus uniquely determined by two components and a by a single two bit value, resulting in a cardinality of 2^{18}. □