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 to 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, considerable research went into lightweight block ciphers, motivated by the need to provide an acceptable security level for specific applications at much lower area, latency, or power consumption levels than, say, the AES [1]. In lightweight ciphers, all components must be tightly optimised, often leading to original solutions, as embodied by designs like CLEFIA [2], KATAN [3], KLEIN [4], LED [5], PRESENT [6], PRINCE [7], SIMON and SPECK [8], and MIDORI [9], to name just a few.

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 and key together select 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 storage address. 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, the tweak being 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 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 use a counter based mode with memory overhead, for instance AEGIS [20, 21], and Intel’s SGX [22]. 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 $\epsilon$ (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 $2n$ 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 [23, 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 $R_m = \mathbb{F}_2[p] = \mathbb{F}_2[X]/(X^m + 1)$ 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. We did not patent QARMA. To the best of our knowledge, we are not aware of any third party 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. $F$) denotes its inverse. QARMA is a three-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 $\tau$ are permutations, $M$ and $Q$ are MixColumns-like operations, with $Q$ involutory, and $\omega$ is a LFSR.

Write $n = 16m$. 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 $4 \times 4$
matrix, for instance, the internal state admits representations

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

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 $t_k$ denotes a (round) tweak, i.e. a value derived only from the key, tweak, and round constants.

### 2.2 Key Specialisation

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

For encryption, put $w^1 = o(w^0) := (w^0 \gg 1) + (w^0 \gg (16m-1))$ 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 $\alpha$, QARMA possesses a property very similar to PRINCE’s $\alpha$-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 \( R(IS; tk) \) 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.** \((\tau(IS))_i = s_{\tau(i)}\) for \( 0 \leq i \leq 15 \), where \( \tau \) 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. \( IS = M \cdot IS \).
4. **SubCells.** For the chosen S-Box \( \sigma \), the \( S \) layer acts on the state as follows: \( s_i \leftarrow \sigma(s_i) \) for \( 0 \leq i \leq 15 \). 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 \( \omega \) updates the tweak cells with indexes 0, 1, 3, 4, 8, 11, and 13. For \( m = 4 \), \( \omega \) 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 \( \overline{R}(IS; tk) \) is the inverse of the forward round function \( R \). Its short form 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 \( P(IS; tk) \), which is 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 \( R \).
2. The pseudo-reflector \( P(IS; tk) \) i.e.
Fig. 3. The QARMA Encryption Algorithm

Algorithm QARMA, Encryption

1: Write $K = w^0||k^0$
2: $w^1 \leftarrow o(w^0)$, $k^1 \leftarrow k^0$
3: IS $\leftarrow P + w^0$
4: for $i = 0$ to $r - 1$ do
5: \hspace{1em} IS $\leftarrow R(IS, k^0 + T + c_i)$ (short round for $i = 0$)
6: \hspace{1em} $T \leftarrow \omega \circ h(T)$
7: IS $\leftarrow R(IS, w^1 + T)$
8: IS $\leftarrow P(IS, k^1)$
9: IS $\leftarrow R(IS, w^0 + T)$
10: for $i = r - 1$ down to 0 do
11: \hspace{1em} $T \leftarrow \overline{R} \circ \overline{h}(T)$
12: \hspace{1em} IS $\leftarrow \overline{R}(IS, k^0 + T + c_i + \alpha)$ (short round for $i = 0$)
13: C $\leftarrow IS + w^1$

(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 $\overline{R}$.

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

2.7 Putting QARMA Together

The encryption algorithm of QARMA, is given in Figure 3. QARMA, has $2r + 2$ rounds.

The round constants are derived from the expansion of the constant $\pi$. 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 $\alpha$ constant, as shown in Table II – 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 $\pi$ as the $\alpha$ constant, set $c_0 = 0$, and then each $c_i$ is a successive block 128 bits of $\pi$, as shown in Table II.

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

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
Table 1. The Round Constants for the 64-bit Ciphers
\[\begin{align*}
\alpha &= \text{COAC29B7C97C50DD} & c_0 &= 0000000000000000 \\
c_1 &= \text{A4093822299F31D0} & c_2 &= 082EFA98EC4E6C89 \\
c_3 &= \text{BE5466CF34E90C6C} & c_4 &= 452821E638D01377 \\
c_5 &= \text{BE5466CF34E90C6C} & c_6 &= 3F84D585B5470917 \\
c_7 &= \text{9216D5D98979FB1B} \\
\end{align*}\]

Table 2. The Round Constants for the 128-bit Ciphers
\[\begin{align*}
\alpha &= \text{243F6A885A308D3}13198A2E3707344 & c_0 &= 0000000000000000 \\
c_1 &= \text{A4093822299F31D0}082EFA98EC4E6C89 \\
c_2 &= \text{C0AC29B7C97C50DD}3F84D585B5470917 \\
c_3 &= \text{2FFD72DBD01ADFB7}B8E1AFED6A267E96 \\
c_4 &= \text{BA7C9045F12CF99}24A19947B3916CF7 \\
c_5 &= \text{9216D5D98979FB1B}D1310BA698DFB5AC \\
c_6 &= \text{718BCD5882154AEE}7B54A41DC25A59B5 \\
c_7 &= \text{9C30D5392AF26013}C5D1B023286085F0 \\
\end{align*}\]

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 \( M = Q = \text{circ}(0, \rho, \rho^2, \rho) \) for QARMA-64; and \( M = \text{circ}(0, \rho, \rho^2, \rho^5) \) (which admits inverse \( \bar{M} = \text{circ}(0, \rho^5, \rho^6, \rho) \)) with \( Q = \text{circ}(0, \rho, \rho^4, \rho) \) 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 \( B_M \) 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 \( B_M = s \).

The second requirement is often understood as minimizing the weight of the matrix, but for fully unrolled HW implementation a better statement would be minimizing the maximum of the weights of all the rows, taking into account the weights of the cells and the underlying algebraic structure.
Matrices are clearly optimal from this point of view. They are never MDS, but they are Almost MDS in dimensions 2, 3, and 4 \cite{24}, and only in these dimensions. However, they may also contribute to the weaknesses of some ciphers. For instance, in the case of \texttt{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 \( \mathbb{F}_2[X]/(X^m + 1) \). The multiplication by the image \( \rho \) 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:

\[
M = \text{circ}(0, \rho^a, \rho^b, \rho^c) = \begin{pmatrix}
0 & \rho^a & \rho^b & \rho^c \\
\rho^c & 0 & \rho^a & \rho^b \\
\rho^b & \rho^c & 0 & \rho^a \\
\rho^a & \rho^b & \rho^c & 0 \\
\end{pmatrix}.
\]

\textbf{Theorem 1.} Let \( R_m = \mathbb{F}_2[\rho] \) be the quotient ring \( \mathbb{F}_2[X]/(X^m + 1) \) where \( \rho \) 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 \( \rho^{4a} + \rho^{4b} + \rho^{4c} \in R_m^* \).

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

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

and since $\rho$ 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

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

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

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

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 $v_0 = v_1 \rho^{a-b+c}$ 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 $\text{wt}(V) + \text{wt}(U) = 4$. □

**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 $\rho^4$. 


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 $M = \text{circ}(0, \rho^a, \rho^b, \rho^c)$ that admit an inverse of the same form $\overline{M} = \text{circ}(0, \rho^d, \rho^e, \rho^f)$, i.e. with entries of weight at most one, are those that satisfy $a \equiv c + \tau$ where $2\tau \equiv 0 \mod m$ (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 $f \equiv d + \tau \mod m$.

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

\[
\begin{pmatrix}
\rho^{a+f} + \rho^{b+e} + \rho^{c+d} \\
\rho^{b+f} + \rho^{c+e} \\
\rho^{a+d} + \rho^{c+f} \\
\rho^{a+e} + \rho^{b+d}
\end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}
\] (3)

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

(i) $b + f \equiv c + e$ , (ii) $a + d \equiv c + f$ , (iii) $a + e \equiv b + d \mod m$ .

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

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

**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

\[
M_{4,1} = Q_{4,1} = \text{circ}(0, \rho, \rho^2, \rho^3) ,
\]

\[
M_{4,2} = Q_{4,2} = \text{circ}(0, 1, \rho^2, 1)
\]

and

\[
M_{4,3} = Q_{4,3} = \text{circ}(0, \rho, \rho^2, \rho) .
\]

These matrices are all involutory.
2. For QARMA-128 ($m = 8$) we consider

\[
M_{8,1} = Q_{8,1} = \text{circ}(0, \rho^2, \rho^4, \rho^6),
\]

\[
M_{8,2} = Q_{8,2} = \text{circ}(0, \rho^1, \rho^4, \rho^5),
\]

and

\[
M_{8,3} = \text{circ}(0, \rho, \rho^2, \rho^5)
\]

admitting inverse \( \overline{M}_{8,3} = \text{circ}(0, \rho^5, \rho^6, \rho) \),

\[
\text{together with } Q_{8,3} = \text{circ}(0, \rho, \rho^4, \rho).
\]

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

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 \( 2^{22} \) fixed points (cf. Lemma 1 in [25]), and \( Q_{8,3} \) that attains the minimum of \( 2^{64} \) fixed points. Note that \( M_{4,1} \) has \( 2^{18} \) 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 [26] 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 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

\(^1\) 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.
Table 3. Lower bounds on the number of active S-Boxes

Class I Diffusion Matrices

<table>
<thead>
<tr>
<th>r</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>32</td>
<td>50</td>
<td>64</td>
<td>≥70</td>
<td>≥76</td>
<td>≥82</td>
<td>≥100</td>
<td>≥114</td>
<td>≥124</td>
</tr>
<tr>
<td>Rel. Tweak</td>
<td>16</td>
<td>24</td>
<td>34</td>
<td>44</td>
<td>52</td>
<td>60</td>
<td>≥64</td>
<td>≥70</td>
<td>≥78</td>
</tr>
</tbody>
</table>

Class II Diffusion Matrices

<table>
<thead>
<tr>
<th>r</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>32</td>
<td>50</td>
<td>60</td>
<td>≥68</td>
<td>≥74</td>
<td>≥82</td>
<td>≥98</td>
<td>≥112</td>
<td>≥120</td>
</tr>
<tr>
<td>Rel. Tweak</td>
<td>14</td>
<td>24</td>
<td>30</td>
<td>42</td>
<td>48</td>
<td>58</td>
<td>≥62</td>
<td>≥68</td>
<td>≥76</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>r</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>32</td>
<td>46</td>
<td>60</td>
<td>≥68</td>
<td>≥76</td>
<td>≥82</td>
</tr>
<tr>
<td>Rel. Tweak</td>
<td>12</td>
<td>20</td>
<td>34</td>
<td>44</td>
<td>50</td>
<td>56</td>
</tr>
</tbody>
</table>

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 just use a dual bound for the whole cipher.

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 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 \( \sigma_0 \), the average occurrence of a differential 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.

---

\(^2\) Note that there is a further, small improvement to 8.3/256 with the matrix circ(0, 1, \( \rho \), 1) which is Class II, but the difference is so small that we prefer the involutory \( M_{4,3} \) instead.
Fig. 4. The Column-wise Active State Transitions for Class I Matrices

Fig. 5. The Column-wise Active State Transitions for Class II Matrices
Remark 3. An analogue of Theorem 1 holds over fields $\mathbb{F}_{2^m}$ as well. A matrix $M = \text{circ}(0, A, B, C)$ 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. [27, 28, 29] 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 Central Construction

Reflection attacks are a specific class of cryptanalysis originally developed against 2K-DES [30] but that can be particularly efficient against PRINCE-like ciphers [25]: 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 $\alpha$ 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 [25]), 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. In fact, the map $w^0 \mapsto o(w^0) = w^1$ is a (linear) orthomorphism, i.e. both $o$ and $x \mapsto x + o(x)$ are permutations. 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 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 [31], 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 \[32, 33\], 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 \[34\].

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. The path delay is estimated using a metric called the depth of a circuit, defined as the sum of sequential path delays of basic operations such as AND, OR, NAND, NOR, XOR, XNOR and NOT (we also consider ternary operators).

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 $S_{b_0}$ as an 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 $\{w, x, y, z\}$ ($\overline{z} := \text{NOT}(x)$). Applying de Morgan’s theorem, it is straightforward to see that this can be evaluated as the NAND4 of the four expressions NAND3($w, y, \overline{z}$), NAND3($w, \overline{y}, z$), NAND2($x, \overline{z}$), and 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 $\overline{wy\overline{z} + w\overline{y}z + x\overline{z} + x\overline{y}}$ which can be implemented as the NAND3 of NAND3($\overline{x}, \overline{y}, z$), NAND2($\overline{w}, \overline{x}$), and 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 NOR3($x, y, z$), NOR2($w, x$), 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 three 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
Table 4. The Bit-Flipping Pattern of the \textsc{MIDORI} S-Box and (up to bit ordering) of $\sigma_0$.

<table>
<thead>
<tr>
<th>Flipping bit</th>
<th>Times $b$ bits are flipped</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>one</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5. The Bit-Flipping Pattern of the $\sigma_1$ S-Box.

<table>
<thead>
<tr>
<th>Flipping bit</th>
<th>Times $b$ bits are flipped</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>one</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 6. The Bit-Flipping Patterns of the $\sigma_2$ S-Box and its Inverse.

<table>
<thead>
<tr>
<th>Flipping bit</th>
<th>Times $b$ bits are flipped</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>one</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flipping bit</th>
<th>Times $b$ bits are flipped</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>one</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

all, we modify Prissette’s algorithm \cite{35} 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 \textsc{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 \textsc{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 \textsc{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 3. For each $b$, each input bits has the same number of one to $b$ bit differentials. This may make individual bits less distinguishable from each other.
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 allowed for PRINCE, which fall into eight affine equivalence classes. These S-Boxes all satisfy the strengthened version of Properties $S_1$ and $S_2$, and Properties $S_3$ and $S_4$.

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 20GE). This S-Box and its inverse are:

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

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 $\sigma_2$ and $\bar{\sigma}_2$ is given in Table 3.

3.4 The 8-bit S-Box

As in MIDORI-128 we construct an 8-bit S-Box $\Sigma$ 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 $\sigma$ is one of the S-Boxes $\sigma_i$ described in §3.3. If we write a 8-bit cell of the state as $(x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0)$, $\sigma$ is applied to $(x_7, x_6, x_5, x_4)$ producing the output bits $(x'_7, x'_6, x'_3, x'_1)$, and to $(x_3, x_2, x_1, x_0)$ producing the output bits $(x'_0, x'_4, x'_2, x'_0)$, and the output of the combined 8-bit S-Box is $(x'_7, x'_6, x'_5, x'_4, x'_2, x'_1, x'_0, x'_0)$. Since the construction is not symmetric, the opposite wiring must be implemented for $\bar{\Sigma}$.

The rationale behind this is that in combination with the chosen matrices $M$, if the output of $\Sigma$ is cyclically rotated by any amount of bits and then fed into another instance of $\Sigma$, exactly 2 bits of the output of one of the two 4-bit S-Boxes
in the first $\Sigma$ are wired into each of the two 4-bit S-Boxes of the second instance of $\Sigma$. This is clear because all the outputs of one instance of $\sigma$ will be wired into the even numbered bits of the output of $\Sigma$, and the outputs of the other instance of $\sigma$ into the bits in the odd numbered positions of the output of $\Sigma$, so any group of four cyclically adjacent output bits of $\Sigma$ will be equally partitioned among the two sources. This is also graphically depicted in Figure 7.

If the 4-bit S-Box $\sigma$ satisfies Property S4, a three-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 $\omega$ Function

The indexes of the tweak register modified by $\omega$ 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 both diffusion characteristics and the bounds deriving from the cryptanalysis of Even-Mansour schemes. 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 three-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 (bounds would be only tight if the S-Boxes could be chosen freely for each cell and round), but they also express a security level before taking whitening into account. QARMA-7-64 has four 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 (cf. Figure 5), 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 mod 8. 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 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.

Resistance against invariant subspace attacks similar to the attack against MIDORI-64 is ensured by the fact that the round constants affect all bits.

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.  

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 $w^0 + w^1 = 0$, which happens for exactly two values of $w^0$. We conclude that in this setting for only one key $K$ in $2^{25}$ (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 $\alpha$ again, and so on; if the structure begins with a $\Delta''' = 0$, the differences are $\alpha$, 0, and so on. At the last round the attacker can only try to guess whether the difference has propagated, because of the whitening. Then, the considerations in [25] would apply, for which, in

\[ \Delta'' = \Delta'' + (w^0 + w^1) \]
the case of QARMA-64, we believe similar attack complexities as in PRINCE to hold (and correspondingly 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 $\Delta$ can take only $2^{32}$ (resp. $2^{64}$) equiprobable values, which depend only on the class $k^1 + \text{Im}(M + I)$ of $k^1$ in $R^{16}/\text{Im}(M+I)$. 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 $\Delta$ can then propagate to values $\Delta''$ with some probability, and for each such $\Delta''$ 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 [25]. 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 $\Delta''$ 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 $\Delta$ can be done.

The improvement with respect to the attacks on reflection ciphers is that the likelihood of a useful differential $\Delta'''$ 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

Recall that the whitening key derivation function $o(\cdot)$ is an orthomorphism. Many of the attacks described below apply to single-key Even-Mansour schemes. If orthomorphisms are used to create a key schedule, the complexity of the attacks usually increases and approaches that of schemes with independent keys or using independent permutations (see [36] for the two-round case). We are not claiming that QARMA’s structure offers the security of an EM scheme with independent keys and permutations, but as the attack complexity estimates in the following are made under a single key assumption, the results are likely not tight.

An analogue of the cryptanalysis described in [37] seems unlikely: The single-key three-round with an involution attack is the one that seems closer 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 [37], this is in-degree $t$). This leads to a time/data/memory ($T/D/M$) tradeoff of $TD = 2^{3n/2}$, 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 $T = 2^{7n/4}$, $D = 2^{3n/4}$ 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 two-round attack can be applied, under the assumption that for, a known core key, a certain $\Delta$ will occur with likelihood $2^{n/2}$ (but choosing the plaintext does not seem to give control on this event). In this case the cipher collapses into a single-key two-round EM construction, not a single round EM, because $o(\cdot)$ is an orthomorphism. For each core key, time complexity is slightly smaller than $2^n$, data (known texts) is slightly smaller than $2^{3n/4}$, 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 online memory usage stay the same. We do get an attack with $T$ slightly better than brute force, but $TD \sim 2^{14n/4}$.

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 [38], with the same likelihoods for a known central difference $\Delta$ 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.

Finally, we also observe that a three-round, two-key EM scheme, according to [39] is attacked with a time data tradeoff of $TD = 2^{2n}$ where $M = D$ (for unkeyed permutations). It is an open question whether our scheme, with a second key derived from the first by means of an orthomorphism offers the same security bound.

4.4 Security Claims

Similarly to MANTIS and PRINCE, for QARMA\textsubscript{7}-64 and QARMA\textsubscript{11}-128, we claim that they attain $n$ bits of tradeoff security. In fact, we claim security against practical attacks already for QARMA\textsubscript{5}-64, resp. QARMA\textsubscript{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\textsubscript{9}-128 offers it with a substantial security margin. For a 128-bit security level, QARMA\textsubscript{8}-128 is a good choice because of the better performance.

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 \texttt{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 of using a very specific process. 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 \texttt{PRINCE} or \texttt{MANTIS} is available.

We applied various implementation tricks to fuse or parallelise various steps. For instance, the critical path for \texttt{MANTIS} and \texttt{QARMA}-64 (with the S-Box $\sigma_0$) 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 \texttt{QARMA}-64 with respect to \texttt{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 \texttt{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 \cite{Canright} 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 \cite{HybridLUT} 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 \cite{ExtremeAES} implementations that require more than 400K gates are reported.

6 Conclusions and Open Questions

We have introduced \texttt{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 QARMA Block Cipher Family

QARMA’s diffusion layers are based on a new class of Almost MDS matrices designed to embed input and output dependent rotations while keeping the optimal critical path of \( \{0,1\} \)-matrices. The confusion layer has been selected using new heuristics for sieving low-latency S-Boxes. The structure is three-round EM scheme using two keys related by an orthomorphism, designed to collapse to a two-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


19. omitted: omitted (2016)


23. 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)


A Test Vectors

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

\[ P = \text{fb623599da6e8127} \quad T = \text{477d469dec0b8762} \]
\[ w0 = \text{84be85ce9804e94b} \quad k0 = \text{ec2802d4e0a488e9} \]

Using the S-Box $\sigma_0$. Using the S-Box $\sigma_1$. Using the S-Box $\sigma_2$. 

<table>
<thead>
<tr>
<th>QARMA64_5</th>
<th>QARMA64_5</th>
<th>QARMA64_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = \text{de}5\text{b}6\text{2}9\text{ab}f\text{c}d\text{e}27$</td>
<td>$C = \text{e}b4\text{6}4\text{d}0\text{5}7\text{1}0\text{8}eb0e$</td>
<td>$C = \text{d}9f8\text{4}b6\text{d}6\text{b}0\text{6}c7c1$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QARMA64_6</th>
<th>QARMA64_6</th>
<th>QARMA64_6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = \text{dd}1\text{1}7\text{a}6\text{e}8\text{a}5\text{b}3\text{3}$</td>
<td>$C = \text{18}c7\text{5}4\text{7}b7af08e42$</td>
<td>$C = \text{b}6f\text{b}e2\text{4}7d\text{d}0\text{3}1\text{8}12$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QARMA64_7</th>
<th>QARMA64_7</th>
<th>QARMA64_7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = \text{6f}a\text{a}bc\text{4}6\text{0}bc\text{8}a\text{7}84$</td>
<td>$C = \text{37}6f1\text{7}0\text{6}7\text{8}c8\text{0}c7c$</td>
<td>$C = \text{b}ac6\text{b}e9\text{e}3\text{d}9\text{e}519$</td>
</tr>
</tbody>
</table>

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

\[ P = \text{fb623599da6e8127} \quad T = \text{477d469dec0b8762} \]
\[ w0 = \text{84be85ce9804e94b} \quad k0 = \text{ec2802d4e0a488e9} \]

Using the S-Box $\sigma_0$. Using the S-Box $\sigma_1$. Using the S-Box $\sigma_2$. 

<table>
<thead>
<tr>
<th>QARMA64_5</th>
<th>QARMA64_5</th>
<th>QARMA64_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = \text{3e}e9\text{a}6\text{c}8\text{2}af0c38$</td>
<td>$C = \text{5}4\text{b}0\text{a}b9\text{5}ba7c3a$</td>
<td>$C = \text{c}003b93\text{9}9\text{b}3\text{3}765$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QARMA64_6</th>
<th>QARMA64_6</th>
<th>QARMA64_6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = \text{9f}5c4\text{1}ec5\text{2}5\text{6}03c9$</td>
<td>$C = \text{a}5\text{1}dd1\text{e}4\text{3}ec582$</td>
<td>$C = \text{2}7\text{0}a7\text{8}7\text{2}75\text{c}4\text{8}d10$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QARMA64_7</th>
<th>QARMA64_7</th>
<th>QARMA64_7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = \text{bca}f6\text{c}8\text{9}de9\text{3}0\text{76}5$</td>
<td>$C = \text{ed}f6\text{7}f3\text{7}0a4\text{8}3f2$</td>
<td>$C = \text{5}c0\text{6}a7\text{5}01\text{b}6\text{3}2\text{fd}$</td>
</tr>
</tbody>
</table>

A.3 Test Vectors for QARMA-128

\[ P = \text{2f}d\text{b}b\text{6}a2\text{c}39\text{5}e9\text{5}9 \quad T = \text{2}4\text{2}7\text{6}7\text{f}d3\text{4}8\text{6}a9\text{6}d \quad w0 = \text{58}9\text{4}8\text{b}7\text{ef}8\text{e}5\text{e}c7\text{f} \quad k0 = \text{4}e7\text{c}e2\text{0}81\text{0}0\text{2}eda4 \quad fe20\text{a}a4\text{d}8\text{6}8\text{be}09 \]
Using the S-Box $\sigma_0$.

<table>
<thead>
<tr>
<th>Block Cipher</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>QARMA128_8</td>
<td>950e8c0cbfa6a71f dc002c334f66e1f2</td>
</tr>
<tr>
<td>QARMA128_9</td>
<td>4f9945f78bc95fdf 2b7ef9eb85fa032</td>
</tr>
<tr>
<td>QARMA128_10</td>
<td>be8343304ec23a37 7c73e199b0e37cf8</td>
</tr>
</tbody>
</table>

Using the S-Box $\sigma_1$.

<table>
<thead>
<tr>
<th>Block Cipher</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>QARMA128_8</td>
<td>4d20b727c1e13be2 e2eea20e4c5a40fd</td>
</tr>
<tr>
<td>QARMA128_9</td>
<td>c68d7fb1e329830c d29b626efa4015f0</td>
</tr>
<tr>
<td>QARMA128_10</td>
<td>c155fbbac8505e8 790f72a36088dec5</td>
</tr>
</tbody>
</table>

Using the S-Box $\sigma_2$.

<table>
<thead>
<tr>
<th>Block Cipher</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>QARMA128_8</td>
<td>9bd49832f494d78 b39db929426ddb44</td>
</tr>
<tr>
<td>QARMA128_9</td>
<td>7564cddcf60ecc04 bae52c17e31140ca</td>
</tr>
<tr>
<td>QARMA128_10</td>
<td>b52d6eac88dc5c76 bead85559b7d20d5</td>
</tr>
</tbody>
</table>

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

**Proposition 1.** The matrix $M = \text{circ}(0, \rho, \rho^4, \rho^5)$ over $R_8 = F_2[\rho] = F_2[X]/(X^8 + 1)$ used in QARMA 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 $V = (v_0, v_1, v_2, v_3)$ 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.

$$v_0 = v_2 \rho^4 + (1 + \rho^2 + \rho^3 + \rho^5) \zeta.$$ 

Similarly,

$$v_1 = v_2 \rho^4 + (1 + \rho^2 + \rho^4 + \rho^6) \zeta'.$$

Note that $\zeta$ and $\zeta'$ 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 $\zeta$ there is a unique $\zeta'$ (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 $2^{18}$. \qed