Truncated Differential Attacks: 
New Insights and 10-round Attacks on QARMA

Zahra Ahmadian¹, Akram Khalesi¹, Dounia M’foukh², Hossein Moghimi¹ and María Naya-Plasencia²

¹ Department of Electrical Engineering, Shahid Beheshti University, Tehran, Iran, 
z_amadian@sbu.ac.ir,a_khalesi@sbu.ac.ir,h.moghimi@mail.sbu.ac.ir
² Inria, Paris, France, 
dounia.mfoukh@inria.fr,maria.naya_plasencia@inria.fr

Abstract. Truncated differential attacks were introduced by Knudsen in 1994 [1]. They are a well-known family that has arguably received less attention than some other variants of differential attacks. This paper gives some new insight on truncated differential attacks and provides the best-known attacks on both variants of the lightweight cipher QARMA, in the single tweak model, reaching for the first time 10 rounds while contradicting the security claims of this reduced version. These attacks use some new truncated distinguishers as well as some evolved key-recovery techniques.

Keywords: Cryptanalysis · truncated differentials · QARMA · distinguisher · key recovery

1 Introduction

In the realm of modern cryptography, the design and analysis of secure block ciphers play a pivotal role in ensuring the confidentiality of sensitive data. The development of advanced encryption algorithms has been an ongoing endeavor, aiming to thwart increasingly sophisticated attacks while maintaining implementation efficiency. One of the key aspects in this evolution is the study of the variations of differential attacks, which are powerful techniques utilized by cryptanalysts to probe the vulnerabilities of cryptographic primitives. This paper focuses on the truncated differential attack, proposed first in 1994 by Knudsen [1], as a tool for the security evaluation of block ciphers.

Despite some instances of cryptanalysis based on truncated differential attacks as an independent attack [2, 3, 4], this attack has received less attention compared to other variations, such as the impossible differential attack, higher-order differential attacks, as well as boomerang and rectangle attacks. The primary utilization of an automated truncated differential path search has been targeted for discovering paths with minimal activation of S-boxes, to serve as a tool for high-probability concrete differential paths [5]. It also aids in identifying contradictions pertinent to impossible differential attacks [6].

In a recent work [7], a novel MILP (Mixed Integer Linear Programming) tool has been introduced for identifying the optimum truncated differential paths and applied on MIDORI, SKINNY, and CRAFT block ciphers covering a greater number of rounds with higher probabilities compared to their concrete differential counterparts.

This work subsequently garnered some interest in truncated differential attacks. In [8], considering that [7] has utilized certain approximations, an effective algorithm for accurately calculating the truncated differential path probability for a given truncated path is proposed. Another technique aimed at calculating the probability of truncated
differentials, considering the clustering effects (also referred to as the differential effect) has been outlined in [9]. Furthermore, certain studies have employed the methodology presented in [7] to automate the discovery of other distinguishers. These encompass the triangle attack [10] and mixture differential attacks [11], as examples. The significant advantages of truncated differential attacks, compared to concrete differential attacks, are as follows.

- **Simplicity.** The truncated differential attack utilizes a word-oriented variable definition. Moreover, this kind of attack does not inherently depend on the S-box details, hence its MILP model is free from the bottleneck of S-box modeling. Consequently, the truncated differential automatic search tools display enhanced efficiency and running time compared to the automatic tools for finding optimum concrete (bit-oriented) differentials.

- **Efficiency.** There are notable instances where the truncated differential distinguisher outperforms its concrete counterpart. Some examples include KLEIN, MIDORI, SKINNY, and CRAFT, for which truncated differential paths have been proposed for the number of rounds, proving that there are no concrete differential distinguishers [7, 2, 3].

- **Value-insensitivity.** The truncated differential distinguisher is inherently independent of the concrete value of the active words. This makes the key recovery part of the attack more flexible, potentially requiring less key material to be guessed, at the two edges of the distinguisher.

Reflection ciphers are a class of symmetric encryption algorithms that exhibit a unique property: the set of encryption functions is identical to the set of decryption functions. In other words, the encryption and decryption processes are the same, making the cipher "reflect" the input to produce the output. This design strategy aims to reduce the implementation cost of the cipher, by minimizing the overhead of decryption on top of the encryption.

PRINCE block cipher [12], an SPN cipher with FX construction, stands as one of the most renowned examples of a reflection cipher. To be precise, it possesses the $\alpha$-reflection property, meaning that decryption is equivalent to the encryption with the related key of $K_{dec} = K_{enc} \oplus \alpha$, where $\alpha$ is a constant. In [13], a new attack called the reflection attack is proposed as a dedicated tool for cryptanalysis of PRINCE-like ciphers. It exploits the existence of too many fixed points in the intermediate rounds of the cipher and its extension to the full cipher.

Following in the footsteps of PRINCE, MANTIS [14] emerges as the subsequent reflection cipher again in the FX framework. It takes inspiration from PRINCE’s design while evolving into a tweakable block cipher. Notably, MANTIS integrates certain choices from MIDORI’s components [15] to enhance its structure. However, a practical attack on MANTIS, has been presented in [16], attributed to the MANTIS’s extremely lightweight components, including the tweak schedule, and the vulnerability resulting from the interaction between the MIDORI-inspired round function and the PRINCE-inspired inner rounds.

QARMA family of block ciphers is the most recent reflection cipher, boasting additional features such as being tweakable, lightweight, and low-latency [17]. Drawing inspiration from PRINCE, MIDORI, and MANTIS, QARMA exhibits notable differences both in the structure and in the choice of components. Unlike its predecessors, QARMA adopts a three-round Even-Mansour (EM) construction [18] rather than adhering to the FX construction. This departure from the FX construction was motivated by the cryptanalysis presented in [19]. Furthermore, QARMA’s decision to pivot to EM construction is motivated by the improved time, memory, and data complexities, which offer superior bounds compared to the FX construction.

Insights gleaned from the MITM and accelerated exhaustive search attacks on PRINCE [20], that exploited the unkeyed central construction of PRINCE, the designer of QARMA
included a key addition in the middle permutation of QARMA. Moreover, this middle permutation is non-involutory to avoid predictable differences at its two sides. Another innovation within the QARMA design pertains to the introduction of a family of almost MDS matrices defined over a ring with zero divisors. They allow to encode rotations in their operation while maintaining the minimal latency associated with binary matrices. The matrices used in QARMA are with the minimum and close to minimum fixed points for 64 and 128-bit versions, respectively. This property as well as suitable whitening keys around the middle permutation makes it secure against the reflection attacks [13].

Similar to PRINCE and MANTIS, QARMA claims a $k$-bit time-data trade-off security, where $k$ is its key size, meaning that for any attack on QARMA, the product of time and data complexities is less than $2^k$.

It is worth noting that the QARMA cipher has been the subject of various cryptanalysis efforts, most of which in the related tweak model, including MITM attack [21, 22], statistical saturation attack [23] and impossible differential attack [24, 25]. The only single-tweak attack [21] is a 10-round MITM attack, but it fails to meet the time-data tradeoff threshold. A review and discussion on the details of QARMA attacks, is provided in Sec. 5.2 of the paper. The designer of QARMA has proposed some security bounds against the differential attack by counting the minimum number of active S-Boxes using Mouha et al.’s MILP search method [26]. However, the resistance of this cipher against the truncated differential attack has not been evaluated, either by the designer or external cryptanalysts.

Contributions. This paper gives new insights into the theory of the -relatively less discussed- truncated differential attack and adds a new dimension to the cryptanalysis of QARMA by introducing the first valid single tweak truncated differential attack on both versions of 10-round QARMA. The contributions of this paper are as follows:

- Extension of truncated differential attack theory: The paper extends the theory of truncated differential attacks by providing a series of evidences and theorems, and formulating the complexities of the truncated differential attack. It is also proved that SPN ciphers with MDS MixColumn are resistant to this kind of attack.

- Discovering Optimal Truncated Differential Distinguishes for QARMA: The almost-MDS property of the MixColumn matrix within the QARMA cipher renders it susceptible to truncated differential analysis. Focusing this cipher, and by employing the automated MILP-based method proposed in [7], the paper identifies the optimum 6 and 4-round truncated differential distinguishers for QARMA family of ciphers.

- Attack on 10-round QARMA: Based on the identified distinguishers, the paper proposes the first valid attacks on both versions of the 10-round QARMA cipher with respect to the security claim trade-off given by the designer of QARMA; i.e. $DT < 2^k$ with data and time complexities $D$ and $T$ and the key size $k$. The attack exploits some evolved key-recovery methods based on list merging techniques and precomputation.

2 Theoretical Background

In this paper, we consider a block cipher $E : \mathbb{F}_2^n \times \mathbb{F}_2^k \rightarrow \mathbb{F}_2^n$, with an $n$-bit block and a $k$-bit key. The input and output difference variables of $E$ are denoted by $\Delta X$ and $\Delta Y$.

Definition 1 (Differential Probability). For block cipher $E$, the differential probability of the concrete input difference $\alpha \in \mathbb{F}_2^n$ and output difference $\beta \in \mathbb{F}_2^n$ is defined as:

$$Pr_{x,K}(\alpha \xrightarrow{E} \beta) = Pr_{x,K}(\Delta Y = \beta|\Delta X = \alpha) = Pr_{x,K}[E_K(x) \oplus E_K(x \oplus \alpha) = \beta]$$  (1)
The differential \((\alpha \xrightarrow{E} \beta)\) is called an efficient distinguisher if \(Pr(\alpha \xrightarrow{E} \beta) \gg 2^{-n}\).

**Definition 2** (Truncated Differential Probability). For block cipher \(E\), the truncated differential probability with input truncated difference \(\Delta_{in} \subseteq F_2^n\), and output truncated difference \(\Delta_{out} \subseteq F_2^n\), is defined as:

\[
Pr_{x,K}(\Delta_{in} \xrightarrow{E} \Delta_{out}) = Pr_{x,K}(\Delta Y \subseteq \Delta_{out} | \Delta X \subseteq \Delta_{in}) = Pr_{x,K}[E_K(x) \oplus E_K(x \oplus \alpha) \in \Delta_{out} | \alpha \in \Delta_{in}]
\]  

(2)

**Definition 3** (Efficient Truncated Differential). The truncated differential \((\Delta_{in} \rightarrow \Delta_{out})\) is called efficient if it can distinguish cipher \(E\) from a Pseudo Random Permutation (PRP), which holds if:

\[
Pr(\Delta_{in} \xrightarrow{E} \Delta_{out}) > Pr(\Delta_{in} \xrightarrow{PRP} \Delta_{out}) = \frac{|\Delta_{out}|}{2^n}.
\]  

(3)

**Preposition 1.** For block cipher \(E\) with concrete input-output differential pair \((\alpha, \beta)\), it holds that:

\[
Pr(\alpha \xrightarrow{E} \beta) = Pr(\beta \xrightarrow{E^{-1}} \alpha)
\]  

(4)

**Lemma 1.** For block cipher \(E\) with truncated input-output differential pair \((\Delta_{in}, \Delta_{out})\) it holds that:

\[
Pr(\Delta_{out} \xrightarrow{E^{-1}} \Delta_{in}) = Pr(\Delta_{in} \xrightarrow{E} \Delta_{out}) \frac{|\Delta_{in}|}{|\Delta_{out}|}
\]  

(5)

**Proof.** By using Bayes theorem,

\[
Pr(\Delta_{out} \xrightarrow{E^{-1}} \Delta_{in}) = Pr(\Delta X \subseteq \Delta_{in} | \Delta Y \subseteq \Delta_{out}) = Pr(\Delta Y \subseteq \Delta_{out} | \Delta X \subseteq \Delta_{in}) \frac{Pr(\Delta X \subseteq \Delta_{in})}{Pr(\Delta Y \subseteq \Delta_{out})} = Pr(\Delta_{in} \xrightarrow{E} \Delta_{out}) \frac{|\Delta_{in}|}{|\Delta_{out}|}
\]  

(6)

where the last equality holds by assuming uniform distributions for \(\Delta X\) and \(\Delta Y\).

**Example 1.** Fig. 1 shows a 9-round truncated differential distinguisher for Skinny-64 with the probability of \(2^{-40}\) in the forward direction [7]. The reverse truncated differential in the backward direction is depicted by red arrows, which has the probability of \(2^{-56}\). Note that this trail is consistent with Lemma 1, where \(|\Delta_{in}| = 2^4\) and \(|\Delta_{out}| = 2^{20}\) and \(P(\Delta_{out} \xrightarrow{E^{-1}} \Delta_{in}) = 2^{-40} \frac{2^4}{2^{20}} = 2^{-56}\).

**Preposition 2.** The truncated differential \((\Delta_{in} \xrightarrow{E} \Delta_{out})\) is efficient, iff \((\Delta_{out} \xrightarrow{E^{-1}} \Delta_{in})\) is efficient.
Zahra Ahmadian, Akram Khalesi, Dounia M’foukh, Hossein Moghimi and María Naya-Plasencia

\[ (\Delta_{\text{in}} \xrightarrow{E} \Delta_{\text{out}} ) \text{ is efficient } \iff Pr(\Delta_{\text{in}} \xrightarrow{E} \Delta_{\text{out}}) > \frac{|\Delta_{\text{out}}|}{2^n} \]
\[ \iff (\Delta_{\text{out}} \xrightarrow{E^{-1}} \Delta_{\text{in}} ) \text{ is efficient.} \quad (7) \]

\textbf{Definition 4} (Optimum Truncated Differential). The truncated differential \((\Delta_{\text{in}} \xrightarrow{E} \Delta_{\text{out}})\) is called optimum if it is efficient and has the maximal \(Pr(\Delta_{\text{in}} \xrightarrow{E} \Delta_{\text{out}})\).

Despite the concrete differential attack in which the data required for the distinguisher is only proportional to the inverse of the differential probability \([27]\), this is not the case with the truncated differential distinguisher. This will be discussed more in Sec. 3.

\textbf{Preposition 3.} The differential \((\Delta_{\text{in}} \xrightarrow{E} \Delta_{\text{out}})\) is the optimum truncated differential for \(E\), iff \((\Delta_{\text{out}} \xrightarrow{E^{-1}} \Delta_{\text{in}})\) is the optimum one for \(E^{-1}\).

\textbf{Proof.} This proposition is proved by contradiction. Suppose that \((\Delta_{\text{out}} \xrightarrow{E^{-1}} \Delta_{\text{in}})\) is not optimum. So, there is another efficient differential \((U \xrightarrow{E^{-1}} V)\) such that \(Pr(U \xrightarrow{E^{-1}} V)[U] > Pr(\Delta_{\text{out}} \xrightarrow{E^{-1}} \Delta_{\text{in}})[\Delta_{\text{out}}]\). By Lemma 1, it holds that:
\[
P(U \xrightarrow{E^{-1}} V)[U] > Pr(\Delta_{\text{out}} \xrightarrow{E^{-1}} \Delta_{\text{in}})[\Delta_{\text{out}}] \\
P(V \xrightarrow{E} U)[V] > Pr(\Delta_{\text{in}} \xrightarrow{E} \Delta_{\text{out}})[\Delta_{\text{in}}] \quad (8)
\]

Eq. (8) means that \((\Delta_{\text{in}} \xrightarrow{E} \Delta_{\text{out}})\) is not the optimum truncated differential distinguisher for \(E\), which is a contradiction. The proof of the reverse direction is analogous to this proof.

\section{Truncated Differential Attack}

Let \((\Delta_{\text{in}} \xrightarrow{E} \Delta_{\text{out}})\) be an \(r_{\text{in}}\)-round truncated differential of probability \(2^{-p}\) for block cipher \(E\). As shown in Fig. 2, we extend \(\Delta_{\text{in}}\) in the backward direction for \(r_{\text{in}}\) rounds to get the difference \(D_{\text{in}}\) and \(\Delta_{\text{out}}\) in the forward direction for \(r_{\text{out}}\) rounds to get \(D_{\text{out}}\), both with probability 1. We denote \(|D_x| = 2^{d_x}\) and \(|\Delta_x| = 2^{\delta_x}\), where \(x \in \{\text{in, out}\}\). According to Def. 3, \((\Delta_{\text{in}} \xrightarrow{E} \Delta_{\text{out}})\) is an efficient distinguisher if
\[
p < n - \delta_{\text{out}} \quad (9)
\]

In the following, we formulate the parameters of the chosen plaintext attack constructed over the truncated differential distinguisher \((\Delta_{\text{in}} \xrightarrow{E} \Delta_{\text{out}})\).

\textbf{Data Complexity.} To generate the pairs required for the attack, we construct \(2^n\) structures in plaintext, each of which is constant in non-active bits, and take all \(2^{d_{\text{in}}}\) values in active bits of \(D_{\text{in}}\). So, each structure can generate about \(2^{2d_{\text{in}}-1}\) pairs with differences belonging to \(D_{\text{in}}\).

\textbf{Lemma 2.} The probability that a pair with plaintext difference \(D_{\text{in}}\) come up with a difference \(\Delta_{\text{in}}\) after \(r_{\text{in}}\) rounds is \(P_{\text{filt}} = 2^{-(d_{\text{in}}-\delta_{\text{in}})}\).
Figure 2: Truncated differential attack framework

Proof. We can assume the differential \((\Delta_{in} \xrightarrow{E^{-1}} D_{in})\) for the first \(r_{in}\) rounds of \(E\) in backward direction, with probability 1. Applying Lemma 1 to this differential, yields:

\[
Pr(D_{in} \xrightarrow{E^{-1}} \Delta_{in}) = Pr(\Delta_{in} \xrightarrow{E^{-1}} D_{in}) \big| \Delta_{in} \big| D_{in} = 1 \cdot 2^{\delta_{in}} = 2^{-(d_{in} - \delta_{in})}
\]  

(10)

Therefore, the number of total pairs required for the attack must be equal to \((2^{-p} \times P_{filt})^{-1} = 2^{p + d_{in} - \delta_{in}}\). This gives the number of required structure as \(2^{s + 2d_{in} - 1} = 2^{p + d_{in} - \delta_{in}}\) which yields \(s = p - d_{in} - \delta_{in} + 1\). Finally, the data required for the attack would be as follows.

\[
D = 2^{s + d_{in}} = 2^{p - \delta_{in} + 1}
\]  

(11)

Note that to minimize the data complexity, it is necessary to minimize the value of \(p - \delta_{in}\), which is consistent with the definition of the optimum distinguisher, given in Def. 4.

**Time Complexity.** The probability that a differential pair with difference \(\Delta_{in}\) at round \(r_{in}\) have a difference belonging to \(D_{out}\) at the output is \(P_{sieve} = 2^{-(n - d_{out})}\). So, the total number of sieved pairs supposed to be processed in the key recovery phase of the attack is \(P = 2^{p - n + d_{in} + d_{out} - \delta_{in}}\), and the time complexity of the attack is:

\[
T = (2^{p - \delta_{in} + 1} + 2^{p - \delta_{in} + 1} \frac{C_S}{C_E} + 2^{p - n + d_{in} + d_{out} - \delta_{in}} \frac{C_{KR}}{C_E})C_E
\]  

(12)

where \(C_E, C_S,\) and \(C_{KR}\) are the time complexities of the encryption, the sieving step, and the key recovery step, respectively.

The concrete differential attack is a symmetric attack, which means that if there is an attack in the forward direction, there is also another one with the same main parameters, using the reverse distinguisher in the backward direction [28]. In the following theorem, we show that the same case is valid for the truncated differential attack.

**Theorem 1.** Suppose that there is a chosen plaintext truncated differential attack on block cipher \(E\) based on the distinguisher \((\Delta_{in} \xrightarrow{E} \Delta_{out})\) of probability \(2^{-p}\), with data complexity \(D\) and the total sieved pairs \(P\). we can construct a chosen ciphertext attack, using the reversed truncated differential \((\Delta_{in} \xleftarrow{E^{-1}} \Delta_{out})\), with the same data complexity \(D\) and total sieved pairs \(P\).
The time complexity of the attack mainly depends on

Truncated differential attacks are particularly effective on word-oriented Substitution-

which must be equal to

This distinguisher is efficient if

So, the total number of pairs would be

Our definition gives

MixColumn: multiplies matrix

Finally, the sieving probability is

The time complexity of the attack mainly depends on D' and P', as

Note that the concrete differential attack can be regarded as a special case of truncated
differential attack, in which \( \delta_{in} = \delta_{out} = 0 \).

4 Potential Targets for Truncated Differential Attack

Truncated differential attacks are particularly effective on word-oriented Substitution-

Permutation Network (SPN) ciphers. While their efficiency remains unlinked to the

S-box specification, it becomes significantly reliant on the differential characteristics of the

MixColumn matrix. In the following, we present the general structure of a word-oriented

SPN cipher, a framework that has served as the foundation for various block cipher designs

including AES, MIDORI, SKINNY, QARMA, and CRAFT. Subsequently, we establish a theorem

that identifies a prerequisite condition for the effectiveness of the truncated differential
distinguisher.

Definition 5 (Word-oriented SPN cipher). The block cipher \( E \), featuring an internal

state matrix of \( t \times t \) of \( m \)-bit words, is called a word-oriented SPN cipher, if it undergoes

the following sequence of four operations in each round, in any order of execution:

- Subkey addition: XORs a subkey of size \( t^2 \) \( m \)-bit words to the internal state.
- S-box: applies \( m \)-bit S-boxes to each \( m \)-bit word of the internal state, in parallel.
- Permutation: applies the word-wise permutation \( \pi \) on \( Z_{12} \) within the internal state,

i.e. \( Y[i] = \pi(X[i]) = X[\pi(i)] \) for \( i \in 0, \ldots, t^2 - 1 \), where \( X[i] \) denotes the \( i^{th} \) word

of the internal state. Here, every column of \( X[i] \) gets mapped to exactly to the \( t \)
columns of \( Y[i] \).
- MixColumn: multiplies matrix \( M \) to each column of the internal state, in parallel,

where, \( M \) is a \( t \times t \) matrix over \( \mathbb{F}_2^m \).
While MDS (Maximum Distance Separable) matrices do provide optimal diffusion for the MixColumn operation, this constraint has been intentionally relaxed in several ciphers to gain implementation advantages. In the next theorem, we show that an MDS MixColumn matrix in the word-oriented SPN cipher is a sufficient condition for provable security against truncated differential attack.

**Theorem 2.** There is no efficient truncated differential distinguisher for 3 rounds of a word-oriented SPN cipher with an MDS MixColumn matrix.

**Proof.** Without loss of generality, we consider the order of operations as given in Def. 5. Let $X_i$, $Y_i$, and $Z_i$ be the truncated differences of the input state, input to MixCol, and output of MixColumn for round $i$, respectively. Let’s denote the number of zero columns in $Z_i$ as $c_i$ for $i = 1, 2, 3$, which is also equivalent to the number of zero columns in $Y_i$. Each column in $Z_i = X_{i+1} = \pi^{-1}(Y_{i+1})$ inherits at least $c_{i+1}$ zero words from $Z_{i+1}$. Therefore, the count of zero words in $Z_i$, excluding those within zero columns, is at least $(t - c_i)c_{i+1}$. We use $P_i$ to denote the truncated differential probability for round $i$. Considering the uniform distribution of the MDS matrix output (as stated in Lemma 3), it holds that:

$$P_i \leq 2^{-m(t-c_i)c_{i+1}} \quad i = 1, 2$$

Let $w_3$ denote the number of zero words of $Z_3$, not belonging to a zero column so $P_3 = 2^{-mw_3}$. Therefore, the probability of the truncated differential path would be upper-bounded by $P_1P_2P_3 \leq 2^{-m(tc_2-c_1c_2+c_3-c_2c_3+w_3)}$. The probability of $Z_3$ being the truncated differential pattern of the output of a PRP is $P_{PRP} = 2^{-m(tc_3+w_3)}$. To prove the theorem, it suffices to show that the upper bound of $P_1P_2P_3$ is less than or equal to $P_{PRP}$, which holds if $c_2 = 0$ or $c_1 + c_3 \leq t$. In the former case, we have $P_1P_2P_3 \leq 2^{-mw_3} = P_{PRP}$, which proves the claim. In the latter, consider a nonzero column of $Z_2$ and $Y_2$, corresponding to the MixColumn operation of round 2. This MixColumn matrix has at least $c_1$ and $c_3$ input and output zero words. Since the MixColumn matrix is MDS, it follows that $c_1 + c_3$ must be less than $t$, thereby completing the proof.

Theorem 2 implies that word-oriented SPN ciphers with non-MDS MixColumn matrices would be potentially vulnerable to truncated differential attacks. This assertion is corroborated by observations made during truncated differential cryptanalysis of MIDORI, SKINNY, and CRAFT [7], all of which use non-MDS MixColumn matrices. Within the QARMA family of block ciphers, to avoid the expensive implementation of MDS matrices, an almost-MDS matrix is selected as the MixColumn matrix. This motivated us to evaluate its security against truncated differential attack, which is outlined in the following Sections.

### 5 Introduction of QARMA Block cipher family

QARMA is a family of lightweight tweakable block ciphers proposed in 2017 [17], fitting to applications such as memory encryption, the generation of very short tags, and the construction of keyed hash functions. This family of block ciphers has two versions:
QARMA-64 and QARMA-128. QARMA-64 has a 64-bit block size with a 128-bit encryption key, while QARMA-128 has a 128-bit block size with a 256-bit key. The design of QARMA was influenced by PRINCE [12] and MANTIS [14]. The cipher is intended for fully-unrolled hardware implementations with low latency, such as memory encryption.

5.1 Specifications of QARMA-64 and QARMA-128

QARMA is an Even-Mansour cipher that uses three stages at the middle, and whitening keys XORed in at the beginning and end of the cipher. The structure of this cipher is shown in Fig. 3 and 4. For QARMA-n, n = 64 or 128, the data is split into 16 m-bit words, m = 4 or 8 respectively, arranged in a 4 × 4 internal state matrix IS, which is shown as:

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

(19)

The key size of QARMA-n is 2n = 32m bits. The secret key K is divided into two halves of length n-bit, \( K = w^0 || k^0 \), and extends to \( k^1 = k^0 \) and \( w^1 = (w^0 \gg 1) + (w^0 \gg (16m - 1)) \). For the sake of simplicity, in the rest of the paper, we refer to \( k^0 \) as \( k \), \( w^0 \) as \( w \), and \( w^1 \) as \( w' \).

The tweak size of QARMA-n is \( n = 16m \) bits denoted as \( T = t_0 || t_1 || \ldots || t_{15} \). The tweak update function consists of a permutation \( h \) as well as an LFSR \( \omega \). The permutation \( h \) is applied as \( h(T) = t_{h(0)} || t_{h(1)} || \ldots || t_{h(15)} \) where \( h = \{ 6, 5, 14, 15, 0, 1, 2, 3, 7, 12, 13, 4, 8, 9, 10, 11 \} \). The LFSR \( \omega \) updates the tweak words \( \{ t_0, t_1, t_4, t_5, t_8, t_{11}, t_{12}, t_{14}, t_{15} \} \) as \( (b_3, b_2, b_1, b_0) \rightarrow (b_0 \oplus b_1, b_3, b_2, b_1) \) for \( m = 4 \) and \( (b_7, b_6, ..., b_0) \rightarrow (b_0 \oplus b_2, b_7, ..., b_1) \) for \( m = 8 \).

The Forward Round Function \( R(IS; tk) \) of QARMA-n is composed of the following layers:

1. AddRoundTweakey. The round tweakey \( tk \) is added to \( IS \).
2. ShuffleCells. For both versions of QARMA, the internal state \( IS \) is shuffled according to the word permutation \( \tau \).

\[
\begin{pmatrix}
  s_0 & s_1 & s_2 & s_3 \\
  s_4 & s_5 & s_6 & s_7 \\
  s_8 & s_9 & s_{10} & s_{11} \\
  s_{12} & s_{13} & s_{14} & s_{15}
\end{pmatrix}
\xrightarrow{\tau}
\begin{pmatrix}
  s_0 & s_{11} & s_6 & s_{13} \\
  s_{10} & s_1 & s_{12} & s_7 \\
  s_5 & s_{14} & s_3 & s_8 \\
  s_{15} & s_4 & s_9 & s_2
\end{pmatrix}
\] (20)

3. MixColumns. The MixColumns matrix \( M_m \), which is multiplied by \( IS \) in QARMA-\( n \) is:

\[
M_4 = \begin{pmatrix}
  0 & \rho & \rho^2 & \rho \\
  \rho & 0 & \rho & \rho^2 \\
  \rho^2 & \rho & 0 & \rho \\
  \rho & \rho^2 & \rho & 0
\end{pmatrix},
M_8 = \begin{pmatrix}
  0 & \rho & \rho^4 & \rho^5 \\
  \rho^5 & 0 & \rho & \rho^4 \\
  \rho^4 & \rho^5 & 0 & \rho \\
  \rho & \rho^4 & \rho^5 & 0
\end{pmatrix}
\] (21)

which is defined over ring \( R_m = F_2[X]/(X^m + 1) \), and the multiplication by the image \( \rho \) of \( X \) in the ring \( R_m \) is just a simple circular left rotation of \( X \). \( M_m \) is a symmetric and involutive matrix, i.e. \( M_m = M_m^{-1} = M_m^T \).

4. SubCells. A \( m \)-bit S-Box is applied to all words of \( IS \) in QARMA-\( n \).

The ShuffleCells and MixColumns layers are omitted for the final round. The Backward Round Function \( \overline{R}(IS; tk) \) is the inverse of the forward round function \( R \). The Pseudo-Reflector function \( \overline{P}(IS; tk) \) is positioned at the center of the cipher, which is

\[
\overline{P} = \overline{R}(\overline{\tau}(k \oplus Q_m(\tau(IS))))
\] (22)

where \( Q_m = M_m \), and the bar over a transformation denotes its inverse. Finally, the \((2r + 2)\)-round QARMA-\( r-n \) is defined as \( \overline{R}^r(\overline{P}^r(\cdot)) \).

5.2 Security Claims and Attacks on QARMA

Time-Data Trade off for QARMA-\( n \). The designer of QARMA has claimed that: Similarly to MANTIS and PRINCE, for QARMA-64 and QARMA-128, with \( r = 7 \) and \( r = 11 \) respectively, we claim that they attain \( n \) bits of tradeoff security. This statement means that any attack on QARMA-\( n \) with data and time complexities \( D \) and \( T \), is a valid attack as far as \( DT < 2^{2n} \).

Number of rounds. For QARMA-64, \( r \) is chosen as 7, i.e. the total round of QARMA-64 would be 16 rounds. However, it is stated that the cipher is believed to be secure against practical attacks already for \( r = 6 \), i.e. 14 rounds, with some use cases even allowing for \( r = 5 \), i.e. 12 rounds. For QARMA-128, \( r \) is chosen as 11, i.e. the total round of QARMA-128 would be 24 rounds. However, it is stated that the cipher is believed to be secure against practical attacks already for \( r = 8 \), i.e. 18 rounds.

Cryptanalysis history. Most of the cryptanalytic work on QARMA-64 and QARMA-128 is in the related tweak model. In [22], the idea of two related tweaks in the MITM attacks on 8 and 9 rounds of QARMA-64, along with a related tweak on 10 rounds of QARMA-128, has been proposed. They are based on a 5-round MITM distinguisher demanding a \( \delta \)-set on tweak variables. For QARMA-64, the \( TD \) is \( 2^{106} \) and \( 2^{105} \), respectively, while QARMA-128 holds \( 2^{244} \). Li et al. in [23] proposed a new cryptanalytic method that can be seen as a related-tweak statistical saturation attack by making a link between related-tweak statistical saturation distinguishers and the tweak difference invariant bias. By applying this approach, a related-tweak statistical saturation attack for 10-rounds of QARMA-64 and an 11-round attack on QARMA-128 were obtained.
In [24], two related-tweak impossible differential attacks on the 11 rounds of both versions of QARMA, without whitening keys, a MITM attack on the 10 rounds of QARMA-128 with whitening keys, and 12 rounds of QARMA-128 with the whitening keys are proposed. In [30] and [25], two related-tweak impossible differential attacks on QARMA-64 and QARMA-128 are proposed, respectively. The former, which is a 10-round key recovery attack with time and data complexity of $2^{125.8}$ and $2^{62}$, violates the TD threshold claimed for QARMA-64. The latter is an 11-round attack on QARMA-128 that omits the outer whitening key with time complexity and data complexity of $2^{145.98}$ and $2^{102.54}$.

The only work in the single-tweak model is [21], which proposes 10-round MITM attacks for both versions of QARMA. However, it does not meet the time-data tradeoff threshold. For QARMA-64 the time complexity and data complexity were reported as $2^{70}$ and $2^{53}$ respectively, but its memory complexity, which is the lower bound of the time complexity, is $2^{110}$. For QARMA-128, the time complexity and data complexity were $2^{141}$ and $2^{105}$ while, its memory complexity remains consistent at $2^{232}$. Hence, both of these attacks do not satisfy the time-data tradeoff threshold and can not be considered as valid attacks of QARMA.

A summary of all the attacks on reduced-round QARMA, along with the new attacks presented in this paper, is given in Tab. 1.

## 6 Truncated Differential attack on QARMA-64 and QARMA-128

In this section, we present 10-round attacks on both versions of QARMA which are the best valid attacks on this cipher in the single-tweak model. We first introduce the optimum 6-round truncated distinguisher for this cipher, then based on the 4 inner rounds of which, we propose the 10-round key recovery attack.

### 6.1 6-round Truncated Distinguisher for QARMA-$n$

In order to find an appropriate truncated differential path, we use the MILP tool proposed in [7] to efficiently automate the search process. In this method, the only thing that needs to be modeled is the Differential Branching Table (DBT) of MixColumn's matrix $M_m$, which is given in Appendix A for $m = 4$ and 8.

In [7], an approximated DBT is employed due to its simplicity. In this approach, all the transition probabilities of DBT table are rounded to the nearest power of $2^{-m}$. For

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Type</th>
<th>Model</th>
<th>Whitening</th>
<th>Symmetry</th>
<th>Rounds</th>
<th>Time</th>
<th>Data</th>
<th>Memory</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MITM</td>
<td>RT</td>
<td>Yes</td>
<td>No</td>
<td>9</td>
<td>2098</td>
<td>216</td>
<td>250</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>RT</td>
<td>Yes</td>
<td>No</td>
<td>10</td>
<td>2125.8</td>
<td>262</td>
<td>257</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>MITM</td>
<td>ST</td>
<td>No</td>
<td>No</td>
<td>10</td>
<td>2116</td>
<td>253</td>
<td>2116</td>
<td>21</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Summary of the external cryptanalysis of QARMA

MITM: Meet In the Middle
SS: Statistical Saturation
TD: Truncated Differential
ID: Impossible Differential
RT/ST: Related Tweak/Single Tweak
TDIB: Tweak Difference Invariant Bias
DBT of $M_4$ and $M_8$, we compute more accurate values for transition probabilities and, due to the relatively wide range of probability values, we use the method given in [5] for MILP modeling of the DBT. We assume independent and uniformly distributed variables for the output differences of the active Sboxes. This assumption has been challenged in [8], where an improved method for computing the precise truncated differential trail probability is proposed. This method counts all the concrete differential trails consistent with the truncated path into account, using an efficient proposed algorithm. This method yet being the most accurate approach, needs the truncated path as an input, so it can not be utilized as a search method for finding the optimum path, but can be used to refine the probability of the truncated path, after finding it by a search tool like [7]. However, based on the values reported in [8], the gap between the approximated method [7] and the precise method [8] is not too much.

To find the optimum truncated differential path, according to Def. 4, we set the objective function to minimize $p - \delta$ in $p$, and to ensure that the returned path is an efficient one, according to Def. 3, we add the constraint $p \leq n - \delta_{out}$ to the model.

6-round distinguishers. We searched for the longest optimum truncated differential distinguisher for the middle part of QARMA-$n$ and found a set of 16 distinguishers covering $R(\mathcal{P}(R^2(\cdot)))$, for both QARMA-64 and QARMA-128. These distinguishers covers 6 full $\mathcal{R}$ (or $\mathcal{R}^*)$, though seven {ShuffleCells + MixColumns} layers are involved in. They are as follows:

$$\tau(M(a \cdot e_i)) \xrightarrow{R^2(\mathcal{P}(R^2(\cdot)))} \tau(M(b \cdot e_i)), \quad a, b \in \mathbb{F}_2^n \setminus \{0\}, \quad i = 0, \ldots, 15 \quad (23)$$

where $a \cdot e_i$ is the $4 \times 4$ matrix whose $i^{th}$ element is $a$, while all other elements are zero. All the 16 paths given in (23) exhibit a reflective pattern and share the following parameters: For QARMA-64, $p = 51.8$, $\delta_{in} = 4$, and $\delta_{out} = 4$, and for QARMA-128, $p = 108.77$, $\delta_{in} = 8$, and $\delta_{out} = 8$. For example, for $i = 4$, we have the following distinguisher for QARMA-64, which is shown in Fig. 5.

$$\begin{pmatrix}
 a \rho & 0 & 0 & 0 \\
 0 & a \rho & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & a \rho^2
\end{pmatrix} \xrightarrow{R^2(\mathcal{P}(R^2(\cdot)))} \begin{pmatrix}
 b \rho & 0 & 0 & 0 \\
 0 & b \rho & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & b \rho^2
\end{pmatrix} \quad (24)$$

Note that even though the output of these distinguishers has $3m$ bits of active words, $\delta_{out} = m$. This is because the three output nibbles are interdependent, as demonstrated in (24), for instance. To investigate the clustering effect for possible improvement in the probability, we fixed the optimum $(\Delta_{in}, \Delta_{out})$ in the MILP model and searched for other possible paths with the same input and output truncated differences. However, we found that there is no other path.

4-round distinguishers. The first and last rounds of transitions in the paths given in (23) are deterministic. So, if we omit these two rounds, we come up with a series of 4-round totally reflective distinguishers with the same probability, which is as follows.

$$a \cdot e_i \xrightarrow{R(\mathcal{P}(\mathcal{R}(\cdot)))} b \cdot e_i \quad (25)$$

where $p = 51.8$ and 108.77 for QARMA-64 and QARMA-128, respectively. An example of this distinguisher has been shown in Fig. 5, being differentiated from the 6-round distinguisher in red. In the next subsection, we use these 4-round paths as the underlying distinguisher for the proposed key recovery attacks.
We use an equivalent representation of QARMA-64 and QARMA-128 with probability \( P = 2^{-51.8} \) and \( 2^{-108.77} \), respectively. The red part is the 4-round path with the same probabilities.

### 6.2 Key Recovery Attack on QARMA-64

We first present the main attack procedure on QARMA-64, by which its key space is reduced by a factor of about \( 2^{10} \). Then, we use it repeatedly to realize a full key recovery attack, satisfying the \( TD \) trade-off threshold.

#### 6.2.1 Reducing the Key Space

We use an equivalent representation of QARMA-64, in which the AddRoundTweakey layer of all rounds, except for the first and last rounds, is replaced by an equivalent key \( u = MC(\tau(k)) \) XORed in after (before) the MixColumn layer in the forward (backward) round functions. We use the 4-round distinguisher given in (25) of probability \( 2^{-51.8} \) for \( i = 4 \), which is shown in Fig. 5, omitting the first and last rounds of the 6-round path. We extend this distinguisher for three rounds in each direction, resulting in a 10-round attack. This attack is shown in Fig. 6. For simplicity in this figure, we have omitted the tweaks, the constants \( c_i \), and also \( \alpha \).

![Figure 5: A 6-round truncated differential distinguisher for QARMA-64 and QARMA-128 with probability \( P = 2^{-51.8} \) and \( 2^{-108.77} \), respectively. The red part is the 4-round path with the same probabilities.](image)

The propagation of active nibbles in the upper and lower parts is exactly the same. This causes all subkeys \( k \) or \( u \) involved in the attack to be the same in the upper and lower parts. The resulting differences in plaintext and ciphertext are active over the same nibbles, and finally \( d_{in} = d_{out} = 36 \).

**Precomputation Phase.** We will use the linear relations in the subkey bits to compute a precomputation table to reduce the time complexity of the attack by reducing the number of subkey bits guessed during the attack steps.

Thanks to the key schedule and the linear description of subkey bits involved in the attack given in Appendix B, we have the following linear relations between the subkey bits of \( w \oplus k \), \( u \) and \( (w' \oplus k) \):

\[
(w' \oplus k)[4]_3 + (w' \oplus k)[11]_2 + (w' \oplus k)[14]_2 = (w \oplus k)[4]_2 + (w \oplus k)[11]_1 + (w \oplus k)[14]_1 + u[5]_1,4.
\]

\[
(w' \oplus k)[4]_3 + (w' \oplus k)[11]_3 + (w' \oplus k)[14]_3 = (w \oplus k)[4]_3 + (w \oplus k)[11]_2 + (w \oplus k)[14]_2 + u[5]_1,2.
\]

\[
(w' \oplus k)[4]_{2,3,4} + (w' \oplus k)[11]_{4} + (w' \oplus k)[14]_{4} = (w \oplus k)[4]_{1,2,3} + (w \oplus k)[11]_{3} + (w \oplus k)[14]_{3} + u[5]_{2,3}.
\]

Thus if we guess the \( 2^{12} \) possible values of \( (w \oplus k)[4, 11, 14] \), the \( 2^4 \) possible values of \( u[8] \), the \( 2^4 \) possible values of \( (w' \oplus k)[4] \), and the \( 2^{24} \) possible values of \( (C, C')[4, 11, 14] \), we would be able to compute \( (w' \oplus k)[11] \) and \( (w' \oplus k)[14] \), according to Step 1 of the attack below. Then, the three linear relations between the subkey bits apply a 3-bit filter on \( (w' \oplus k)[4] \) and only two possible values of \( (w' \oplus k)[4] \) remain for each triplet of
Figure 6: 10-round truncated differential attack on QARMA-64
Therefore, the data required for the attack is we have a size allow to follow the differential path, the following steps are performed:

**Generating Pairs.** We follow the process discussed in Sec. 3 to accurately determine the data required for the attack. Each structure contains $2^{36}$ plaintexts, which are constant in the 7 non-active nibbles $\{0, 1, 2, 3, 6, 9, 12\}$, and take all possible values in the other nibbles. Since none of the differential pairs should share similar values in the active nibbles, the total number of pairs in each structure is $\frac{1}{2}(2^4(2^4 - 1))^9 = 2^{70.16}$. According to the differential branch table of $M_4$, that can be found in Appendix A, the filtering probability is $P_{\text{filter}} = (2^{-7.81})^4 = 2^{-31.25}$, because of the three column transitions in the first MixCol, and the one in the second. It must be held that $2^{2+70.16} = 2^{51.8+31.25}$ which gives $s = 12.9$. Therefore, the data required for the attack is $2^{2+36} = 2^{54.22}$. Finally, the probability of sieving the ciphertext pairs is $P_{\text{sieve}} = (2^{-4})^7 \times (15/16)^9 = 2^{-28.84}$, and the total number of pairs after sieving is $2^{2+70.16-28.84} = 2^{54.22}$.

**Attack Steps.** For each of the $2^{54.22}$ candidate pairs, in order to verify which keys would allow to follow the differential path, the following steps are performed:

1. We first guess the nibble $(w \oplus k)[4]$ which implies the pair of values in $X_1[4]$. The MixColumn transition to column two in $Y_1$ implies that $\Delta X_1[11] = \Delta X_1[14] = \rho^{-1} \Delta X_1[4]$. On the other hand, we know the differences in nibbles $X_0[11]$ and $X_0[14]$ in the input of the S-box as they are given by the plaintext.

   There is a $2^{-p_0}$ probability of having a possible transition through the DDT and for each transition $2^{p_0}$ values make it possible. Thus, on average we associate one value of the nibbles $(w \oplus k)[11,14]$, per pair and per guess of $(w \oplus k)[4]$. The time complexity of this step is $2^{54.22} \times 2^4 = 2^{58.22}$.

2. We guess the nibble $u[5]$ to be able to use the precomputation table. Since we have $P$ and $P'$, we already have the needed bits of the ciphertexts, i.e. $(C,C')[4,11,14]$. Thus we can read on the precomputation table the 2 possible values of $(w' \oplus k)[4]$ and compute as in step 1, the value of $(w' \oplus k)[11]$ and $(w' \oplus k)[14]$. The time complexity after this step is $2^{58.22} \times 2^4 \times 2 = 2^{61.22}$.

3. Since we guessed $u[5]$, we can compute the pairs of values in $Y_5[5]$, and consequently $X_5[5]$. Due to the linear relation imposed by the MixCol matrix, it holds that $\Delta Z_7[4] = \rho^{-1} \Delta X_5[5]$. Thanks to the reflective structure of the cipher, the same conditions apply to the upper part and we can compute $\Delta Z_2[4] = \rho^{-1} \Delta X_2[5]$.

4. We repeat Steps 1, 2 and 3 with $((w' \oplus k)[5], u[0])$ and $((w' \oplus k)[7], u[15])$. The time complexity is now $3 \times 2^{61.22} = 2^{64.82}$.

5. We now have to match the three different candidates subkey bits we computed. For this, we will merge the list of the $2^9$ differences $(\Delta_1 Z_2[4], \Delta_1 Z_7[4])$ computed in Step 3 using $u[5]$, the list of the $2^9$ differences $(\Delta_2 Z_2[4], \Delta_2 Z_7[4])$ computed in Step 3 using $u[0]$ and the list of the $2^9$ differences $(\Delta_3 Z_2[4], \Delta_3 Z_7[4])$ computed in Step 3 using $u[15]$.

   There is $2^9$ possible values for the differences $(\Delta_i Z_2[4], \Delta_i Z_7[4])$, $i = 1, 2, 3$, therefore for each of the $2^9$ possible values of $((w \oplus k)[4, 11, 14], u[5], (w' \oplus k)[4])$, there is $2^9 = 2$ values of $((w \oplus k)[5, 10, 15], u[0], (w' \oplus k)[5, 10, 15])$ which will match. Similarly,
for each of the two values of \((w \oplus k)[5, 10, 15], u[0], (w' \oplus k)[5, 10, 15]\), there will be two matches in the list of differences \((\Delta_3 Z_2[4], \Delta_3 Z_7[4])\) thus there will be 2 values of \((w \oplus k)[7, 8, 13], u[15], (w' \oplus k)[7, 8, 13]\) for each of the 2 values of \((w \oplus k)[5, 10, 15], u[0], (w' \oplus k)[5, 10, 15]\).

Thus for each of the 2\(^9\) guesses of \((w \oplus k)[4, 11, 14], u[5], (w' \oplus k)[4, 11, 14]\) we match two sets of subkey bits candidates \((w \oplus k)[5, 10, 15], u[0], (w' \oplus k)[5, 10, 15]\) and for each of those two sets we match two sets of subkey bits candidates of \((w \oplus k)[7, 8, 13], u[15], (w' \oplus k)[7, 8, 13]\). So overall, we get \(2^{54.22} \times 2^9 \times 2 = 2^{65.22}\) candidate triplets \((P, P', \text{information bits of key})\).

The information bits of subkeys involved in the attack have linear representations in key bits \(w\) and \(k\), which are shown in Tab. 4 of Appendix B. The resulting linear system of equations has 84 equations (each corresponds to a guessed/implied subkey bit in the attack) in 77 variables \((w\) and \(k\) bits), and its rank is 75. This means that the remaining triplets give \(2^{65.22}\) possible values for 75 key bits, hence reducing the space for about 9.78 bits.

### 6.2.2 Recovering the Whole Key

If we repeat the attack steps once again with a new set of data, the data and time complexity will increase by a factor of 2, each. But, the key space is reduced by a factor of \(2^{9.78}\). So, the remaining candidate keys will become \(2^{65.22 - 9.78} = 2^{55.44}\).

In general, the time complexity of repeating the attack for \(N\) times is \(2^{65.22} \times N\), and the complexity of the exhaustive search of the remaining key bits is \(2^{128 - 9.78N}\), so the time complexity would be \(2^{128 - 9.78N} + 2^{65.22} \times N\), which is minimized at \(N = 7\). All in all, the time, data, and memory complexities of the attack are:

\[
\begin{align*}
\mathcal{D} &= 7 \times 2^{58.9} = 2^{51.48} \\
\mathcal{T} &= 2^{65.22} \times 7 + 2^{128 - 9.78} = 2^{68.03} \\
\mathcal{M} &= 2^{65.22} \\
\mathcal{TDD} &= 2^{119.74} < 2^{128}.
\end{align*}
\] (26) (27)

### 6.3 Key Recovery Attack on QARMA-128

The attack on QARMA-128 shares many similarities with QARMA-64 and uses the same 4-round pattern for truncated distinguisher with probability \(2^{-108.77}\). Therefore, in this section, we will focus on highlighting the distinctions between them to avoid repeating the details already discussed.

### Precomputation Phase.

For QARMA-128, we also have linear relations in the subkey bits which are used in the precomputation phase of the attack. Based on the linear description of subkey bits involved in the attack given in Appendix B, we have the following linear relations between the subkey bits of \((w \oplus k), u\) and \((w' \oplus k)\):

\[
\begin{align*}
(w' \oplus k)[4][2] + (w' \oplus k)[11][3] + (w' \oplus k)[14][7] &= (w \oplus k)[4][1] + (w \oplus k)[11][2] + (w \oplus k)[14][6] + u[5][6] \\
(w' \oplus k)[4][3] + (w' \oplus k)[11][4] + (w' \oplus k)[14][8] &= (w \oplus k)[4][2] + (w \oplus k)[11][3] + (w \oplus k)[14][7] + u[5][6] \\
(w' \oplus k)[4][4] + (w' \oplus k)[11][5] + (w' \oplus k)[14][12][2\ldots 8] &= (w \oplus k)[4][3] + (w \oplus k)[11][4] + (w \oplus k)[14][1\ldots 7] + u[5][7, 8] \\
(w' \oplus k)[4][5] + (w' \oplus k)[11][6] + (w' \oplus k)[14][12] &= (w \oplus k)[4][4] + (w \oplus k)[11][5] + (w \oplus k)[14][1] + u[5][1, 8] \\
(w' \oplus k)[4][6] + (w' \oplus k)[11][7] + (w' \oplus k)[14][13] &= (w \oplus k)[4][5] + (w \oplus k)[11][6] + (w \oplus k)[14][2] + u[5][1, 2] \\
(w' \oplus k)[4][7] + (w' \oplus k)[11][8] + (w' \oplus k)[14][4] &= (w \oplus k)[4][6] + (w \oplus k)[11][7] + (w \oplus k)[14][3] + u[5][2, 3]
\end{align*}
\]
(w′ ⊕ k)[4]s + (w′ ⊕ k)[11]2...s + (w′ ⊕ k)[14]5 = (w ⊕ k)[4]7 + (w ⊕ k)[11]1...7 + (w ⊕ k)[14]4 + u[5][3, 4]

Thus if we guess the 2^{24} possible values of (w ⊕ k)[4, 11, 14], the 2^{8} possible values of u[5], the 2^{8} possible values of (w′ ⊕ k)[4], and the 2^{38} possible values of (C, C′)[4, 11, 14] to be able to compute (w′ ⊕ k)[11] and (w′ ⊕ k)[14], as in Step 1 of the attack. Then, the seven linear relations between the subkey bits apply a 7-bit filter on (w′ ⊕ k)[4] and only two possible values of (w′ ⊕ k)[4] remain for each triplet of ((w ⊕ k)[4, 11, 14], u[5], (C, C′)[4, 11, 14]). We can now compute a precomputation table of the values of (w′ ⊕ k)[4] for each possible ((w ⊕ k)[4, 11, 14], u[5], (C, C′)[4, 11, 14]). Thus we have a size 2^{81} precomputation table of the 2 possible values of (w′ ⊕ k)[4] for each of the 2^{80} possible values of ((w ⊕ k)[4, 11, 14], u[5], (C, C′)[4, 11, 14]).

Similarly, we can compute the same precomputation table for the values of (w′ ⊕ k)[5] and (w′ ⊕ k)[7].

**Generating Pairs.** Each structure contains 2^{9m} = 2^{72} plaintexts, which are constant in 7 non-active words, taking all possible values in the other words. The total number of pairs in each structure is \( \frac{1}{2} (2^{m} (2^{m} - 1))^{9} = 2^{142.94} \). The filtering probability is \( P_{filt} = (2^{-15.99})^{4} = 2^{-63.96} \). It must be held that \( 2^{s+142.94} = 2^{108.77+63.96} \) which gives \( s = 29.78 \). Therefore, the data required for the attack is \( 2^{s+72} = 2^{101.78} \). Finally, the probability of sieving the ciphertext pairs is \( P_{sieve} = (2^{-3})^{7} \times \frac{255}{256} = 2^{-56.05} \), and the total number of pairs after sieving is \( 2^{s+127.95-56.05} = 2^{101.84} \).

**Attack Steps.** Since the attack steps is very similar to QARMA-64’s, which are performed for each of the \( 2^{101.68} \) pairs of data. In the following, we just report the time complexity of each step.

1. The time complexity of this step is \( 2^{101.68} \times 2^{8} = 2^{109.68} \).
2. The time complexity after this step is \( 2^{109.68} \times 2^{8} \times 2 = 2^{118.68} \).
3. In this step, it should hold that \( \Delta Z_{7}[4] = \rho^{-5} \Delta X_{8}[5] \) and \( \Delta Z_{2}[4] = \rho^{-5} \Delta X_{2}[5] \).
4. After repeating Steps 1.2 and 3 with ((w′ ⊕ k)[5], u[0]) and ((w′ ⊕ k)[7], u[15]), the time complexity is now \( 3 \times 2^{118.68} = 2^{120.26} \).
5. The three different candidate subkey bits should be matched by merging the list of the \( 2^{17} \) differences (\( \Delta Z_{2}[4], \Delta_{i} Z_{7}[4] \)), \( 1 \leq i \leq 3 \), computed in Step 3 using u[5], u[0], and u[15].

Thus for each of the \( 2^{17} \) guesses of ((w ⊕ k)[4, 11, 14], u[5], (w′ ⊕ k)[4, 11, 14]) we match two sets of subkey bits candidates ((w ⊕ k)[5, 10, 15], u[0], (w′ ⊕ k)[5, 10, 15]) and for each of those two sets we match two sets of subkey bits candidates of ((w ⊕ k)[7, 8, 13], u[15], (w′ ⊕ k)[7, 8, 13]). So overall, we get \( 2^{101.68} \times 2^{17} \times 2 \times 2 = 2^{120.68} \) candidate triplets (\( P, P', \) information bits of key).

The linear representations of the information bits of subkeys involved in the attack in key bits w and k are shown in Tab. 5 of Appendix B. The resulting linear system of equations has 168 equations in 149 variables, with rank 147. This means that the remaining triplets give \( 2^{120.68} \) possible values for 147 key bits, hence reducing the space for about 26.32 bits.

### 6.3.1 Recovering the Whole Key

The time complexity of repeating the attack \( N \) times is \( 2^{101.78} \times N \), and the complexity of the exhaustive search of the remaining key bits is \( 2^{256-26.32N} \), so the time complexity...
would be $2^{256 - 26.32N} + 2^{120.68} \times N$, which is minimized at $N = 6$. All in all, the time, data, and memory complexities of the attack are:

\[
\begin{align*}
D &= 6 \times 2^{101.78} = 2^{104.36} \\
T &= 2^{256 - 26.32 \times 6} + 2^{120.68} \times 6 = 2^{123.36} \\
M &= 2^{120.68} \\
TD &= 2^{227.62} < 2^{256}.
\end{align*}
\] (28)

7 Conclusion

We have generalized and provided some new insight on truncated differential attacks, and we hope these results will be useful for future research. We have analyzed the \textsc{Qarma} [17] block cipher proposed in 2017, with respect to truncated differential attacks and have been able to propose the best known attacks on this cipher, that reach up to 10 rounds and break the security claims of these reduced versions proposed by the designers (unlike the previous known attacks on 10-round \textsc{Qarma} that had a complexity higher than the security claims). For reaching this attacks we provide some new truncated distinguishers and an improved and dedicated key-recovery part, based on list merging techniques and precomputation, that allows to greatly reduce the time complexity.

Acknowledgment

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement no. 714294 - acronym QUASYModo).

References


### A Differential Branch Tables for QARMA-64/128

Tables 2 and 3 show the Differential Branch Table (DBT) for QARMA-64/128 MixColumn Matrix $M$. whose entry $(a, b)$ reflects the base-2 logarithm of $P(a \xrightarrow{M} b)$, the transition probability of matrix $M$ (and $Q$), for the input and output truncated differential vectors $\mathbf{a} = [a_3, a_2, a_1, a_0]^T$ and $\mathbf{b} = [b_3, b_2, b_1, b_0]^T$, respectively. The impossible transitions are shown by a “-”.

$$DBT(a, b) = \log_2(Pr_x\{Hw(M \cdot x) = b | Hw(x) = a\})$$

where $Hw(\cdot)$ is the truncated Hamming weight operator.
<table>
<thead>
<tr>
<th>0x0</th>
<th>0x1</th>
<th>0x2</th>
<th>0x3</th>
<th>0x4</th>
<th>0x5</th>
<th>0x6</th>
<th>0x7</th>
<th>0x8</th>
<th>0x9</th>
<th>0xa</th>
<th>0xb</th>
<th>0xc</th>
<th>0xd</th>
<th>0xe</th>
<th>0xf</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>0x1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>0x2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0x3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0x4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0x5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0x6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0x7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0x8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0x9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0xa</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0xb</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0xc</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0xd</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0xe</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0xf</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 3: DBT for QARMA-128 MixColumn Matrix $M_8$

<table>
<thead>
<tr>
<th>in/out</th>
<th>0x0</th>
<th>0x1</th>
<th>0x2</th>
<th>0x3</th>
<th>0x4</th>
<th>0x5</th>
<th>0x6</th>
<th>0x7</th>
<th>0x8</th>
<th>0x9</th>
<th>0xa</th>
<th>0xb</th>
<th>0xc</th>
<th>0xd</th>
<th>0xe</th>
<th>0xf</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0x1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x3</td>
<td>-</td>
<td>-</td>
<td>-14.404</td>
<td>-</td>
<td>-</td>
<td>-8.011</td>
<td>-</td>
<td>-</td>
<td>-8.011</td>
<td>-</td>
<td>-</td>
<td>-8.011</td>
<td>-</td>
<td>-</td>
<td>-0.011</td>
<td></td>
</tr>
<tr>
<td>0x4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-7.994</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-0.005</td>
</tr>
<tr>
<td>0x6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-14.404</td>
<td>-8.011</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-8.011</td>
<td>-</td>
<td>-</td>
<td>-8.011</td>
<td>-0.011</td>
</tr>
<tr>
<td>0x7</td>
<td>-</td>
<td>-16.007</td>
<td>-</td>
<td>-</td>
<td>-16.007</td>
<td>-8.011</td>
<td>-15.99</td>
<td>-</td>
<td>-</td>
<td>-8.06</td>
<td>-</td>
<td>-8</td>
<td>-8.006</td>
<td>-0.022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x8</td>
<td>-</td>
<td>-16.007</td>
<td>-</td>
<td>-</td>
<td>-16.007</td>
<td>-8.011</td>
<td>-</td>
<td>-</td>
<td>-8.06</td>
<td>-</td>
<td>-8</td>
<td>-8.006</td>
<td>-0.022</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x9</td>
<td>-</td>
<td>-16.007</td>
<td>-</td>
<td>-</td>
<td>-16.007</td>
<td>-8.011</td>
<td>-</td>
<td>-</td>
<td>-8.06</td>
<td>-</td>
<td>-8</td>
<td>-8.006</td>
<td>-0.022</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xa</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-7.99</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-0.005</td>
<td></td>
</tr>
<tr>
<td>0xb</td>
<td>-</td>
<td>-15.99</td>
<td>-</td>
<td>-</td>
<td>-8</td>
<td>-16.007</td>
<td>-</td>
<td>-</td>
<td>-8.006</td>
<td>-8</td>
<td>-16.007</td>
<td>-8.011</td>
<td>-8.006</td>
<td>-0.011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xc</td>
<td>-</td>
<td>-15.99</td>
<td>-</td>
<td>-</td>
<td>-8</td>
<td>-16.007</td>
<td>-</td>
<td>-</td>
<td>-8.006</td>
<td>-8</td>
<td>-16.007</td>
<td>-8.011</td>
<td>-8.006</td>
<td>-0.011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xd</td>
<td>-</td>
<td>-16.007</td>
<td>-8.006</td>
<td>-</td>
<td>-</td>
<td>-8</td>
<td>-16.007</td>
<td>-8.011</td>
<td>-8.006</td>
<td>-0.011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xe</td>
<td>-</td>
<td>-16.007</td>
<td>-8.006</td>
<td>-</td>
<td>-</td>
<td>-8</td>
<td>-16.007</td>
<td>-8.011</td>
<td>-8.006</td>
<td>-0.011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xf</td>
<td>-</td>
<td>-16.007</td>
<td>-8.006</td>
<td>-</td>
<td>-</td>
<td>-8</td>
<td>-16.007</td>
<td>-8.011</td>
<td>-8.006</td>
<td>-0.011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The information bits guessed/implied during the attack have linear representations in key bits $w$ and $k$ which are shown in Tab. 4 and Tab. 5. In this paper the bit indices are arranged according to the following patterns. For QARMA-64

$$
\begin{pmatrix}
  x_0 \ldots x_3 & x_4 \ldots x_7 & x_8 \ldots x_{11} & x_{12} \ldots x_{16} \\
  x_{16} \ldots x_{19} & x_{20} \ldots x_{23} & x_{24} \ldots x_{27} & x_{28} \ldots x_{31} \\
  x_{32} \ldots x_{35} & x_{36} \ldots x_{39} & x_{40} \ldots x_{43} & x_{44} \ldots x_{47} \\
  x_{48} \ldots x_{51} & x_{52} \ldots x_{55} & x_{56} \ldots x_{59} & x_{60} \ldots x_{63}
\end{pmatrix}, \tag{30}
$$

and for QARMA-128

$$
\begin{pmatrix}
  x_0 \ldots x_7 & x_8 \ldots x_{15} & x_{16} \ldots x_{23} & x_{24} \ldots x_{31} \\
  x_{32} \ldots x_{39} & x_{40} \ldots x_{47} & x_{48} \ldots x_{55} & x_{56} \ldots x_{63} \\
  x_{64} \ldots x_{71} & x_{72} \ldots x_{79} & x_{80} \ldots x_{87} & x_{88} \ldots x_{95} \\
  x_{96} \ldots x_{103} & x_{104} \ldots x_{111} & x_{112} \ldots x_{119} & x_{120} \ldots x_{127}
\end{pmatrix}, \tag{31}
$$
Table 4: QARMA-64 subkey linear description in \( w \) and \( k \)

<table>
<thead>
<tr>
<th>Guessed/computed subkeys</th>
<th>Linear description</th>
<th>Guessed/computed subkeys</th>
<th>Linear description</th>
<th>Guessed/computed subkeys</th>
<th>Linear description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subkey</td>
<td>bit ( w )</td>
<td>( k )</td>
<td>subkey</td>
<td>bit ( w )</td>
<td>( k )</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>26</td>
<td>26</td>
<td>26</td>
<td>26</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>29</td>
<td>29</td>
</tr>
<tr>
<td>31</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>33</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>39</td>
<td>39</td>
<td>39</td>
<td>39</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>44</td>
<td>44</td>
<td>44</td>
<td>44</td>
<td>43</td>
<td>43</td>
</tr>
<tr>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>51</td>
<td>51</td>
</tr>
<tr>
<td>53</td>
<td>53</td>
<td>53</td>
<td>53</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>53</td>
<td>53</td>
</tr>
<tr>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>57</td>
<td>57</td>
<td>57</td>
<td>57</td>
<td>56</td>
<td>56</td>
</tr>
<tr>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>57</td>
<td>57</td>
</tr>
<tr>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>59</td>
<td>59</td>
</tr>
<tr>
<td>61</td>
<td>61</td>
<td>61</td>
<td>61</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>62</td>
<td>62</td>
<td>62</td>
<td>62</td>
<td>61</td>
<td>61</td>
</tr>
<tr>
<td>63</td>
<td>63</td>
<td>63</td>
<td>63</td>
<td>62</td>
<td>62</td>
</tr>
</tbody>
</table>
Table 5: QARMA-128 subkey linear description in $w$ and $k$

<table>
<thead>
<tr>
<th>subkey</th>
<th>bit</th>
<th>$w$</th>
<th>$k$</th>
<th>$w$</th>
<th>$k$</th>
<th>$w$</th>
<th>$k$</th>
<th>$w$</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>33</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
</tr>
<tr>
<td>44</td>
<td>44</td>
<td>44</td>
<td>44</td>
<td>44</td>
<td>44</td>
<td>44</td>
<td>44</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
<td>51</td>
</tr>
<tr>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>53</td>
<td>53</td>
<td>53</td>
<td>53</td>
<td>53</td>
<td>53</td>
<td>53</td>
<td>53</td>
<td>53</td>
<td>53</td>
</tr>
<tr>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>56</td>
</tr>
<tr>
<td>57</td>
<td>57</td>
<td>57</td>
<td>57</td>
<td>57</td>
<td>57</td>
<td>57</td>
<td>57</td>
<td>57</td>
<td>57</td>
</tr>
<tr>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>61</td>
<td>61</td>
<td>61</td>
<td>61</td>
<td>61</td>
<td>61</td>
<td>61</td>
<td>61</td>
<td>61</td>
<td>61</td>
</tr>
<tr>
<td>63</td>
<td>63</td>
<td>63</td>
<td>63</td>
<td>63</td>
<td>63</td>
<td>63</td>
<td>63</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td>66</td>
<td>66</td>
<td>66</td>
<td>66</td>
<td>66</td>
<td>66</td>
<td>66</td>
<td>66</td>
<td>66</td>
<td>66</td>
</tr>
<tr>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
<td>67</td>
</tr>
<tr>
<td>68</td>
<td>68</td>
<td>68</td>
<td>68</td>
<td>68</td>
<td>68</td>
<td>68</td>
<td>68</td>
<td>68</td>
<td>68</td>
</tr>
<tr>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
<td>69</td>
</tr>
<tr>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>71</td>
<td>71</td>
<td>71</td>
<td>71</td>
<td>71</td>
<td>71</td>
<td>71</td>
<td>71</td>
<td>71</td>
<td>71</td>
</tr>
<tr>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>73</td>
<td>73</td>
<td>73</td>
<td>73</td>
<td>73</td>
<td>73</td>
<td>73</td>
<td>73</td>
<td>73</td>
<td>73</td>
</tr>
<tr>
<td>74</td>
<td>74</td>
<td>74</td>
<td>74</td>
<td>74</td>
<td>74</td>
<td>74</td>
<td>74</td>
<td>74</td>
<td>74</td>
</tr>
<tr>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>76</td>
<td>76</td>
<td>76</td>
<td>76</td>
<td>76</td>
<td>76</td>
<td>76</td>
<td>76</td>
<td>76</td>
<td>76</td>
</tr>
<tr>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
<td>77</td>
</tr>
<tr>
<td>78</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>78</td>
</tr>
<tr>
<td>79</td>
<td>79</td>
<td>79</td>
<td>79</td>
<td>79</td>
<td>79</td>
<td>79</td>
<td>79</td>
<td>79</td>
<td>79</td>
</tr>
<tr>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
<td>81</td>
</tr>
<tr>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
<td>82</td>
</tr>
<tr>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
<td>83</td>
</tr>
</tbody>
</table>

Note: The table continues with more entries not shown here.