IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 22, 2019, accepted December 22, 2019, date of publication January 6, 2020, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964259

Improved Cryptanalysis of Reduced-Version QARMA-64/128

YA LIU[1,2,3], TIANDE ZANG[1], DAWU GU[3], FENGYU ZHAO[1,2], WEI LI[4,5,6], AND ZHIQIANG LIU[3]

[1]Department of Computer Science and Technology, University of Shanghai for Science and Technology, Shanghai 200093, China
[2]Shanghai Key Laboratory of Modern Optical System, Engineering Research Center of Optical Instrument and System, Ministry of Education, University of Shanghai for Science and Technology, Shanghai 200093, China
[3]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[4]School of Computer Science and Technology, Donghua University, Shanghai 201620, China
[5]Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai 200240, China
[6]Shanghai Key Laboratory of Integrate Administration Technologies for Information Security, Shanghai 200240, China

Corresponding author: Ya Liu (liuya@usst.edu.cn)

This work was supported in part by the National Cryptography Development Fund under Grant MMJJ20180202, and in part by the Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-17-008.

ABSTRACT QARMA is a new tweakable block cipher used for memory encryption, the generation of short tags and the construction of the keyed hash functions in future. It adopts a three-round Even-Mansour scheme and supports 64 and 128 bits of block size, denoted by QARMA-64 and QARMA-128, respectively. Their tweak lengths equal the block sizes and their keys are twice as long as the blocks. In this paper, we improve the security analysis of reduced-version QARMA against impossible differential and meet-in-the-middle attacks. Specifically, first exploit some properties of its linear operations and the redundancy of key schedule. Based on them, we propose impossible differential attacks on 11-round QARMA-64/128, and meet-in-the-middle attacks on 10-round symmetric QARMA-128 and the last 12 rounds of asymmetric QARMA-128. Compared with the previously best known results on QARMA-64, our attack can recover 16 more bits of master key with the almost complexities. Compared with the previously best known results on symmetric QARMA-128, the memory complexity of our attack in Section IV is reduced by a factor of 2^{48}. Moreover, the meet-in-the-middle attack on 12-round QARMA-128 is the best known attack on QARMA-128 in terms of the number of rounds.

INDEX TERMS Tweakable block ciphers, QARMA, meet-in-the-middle attacks, impossible differential cryptanalysis, tweaks.

I. INTRODUCTION

In 2002, Liskov et al. proposed the tweakable block cipher which is widely applied in complex and diversified applications [1]. It has a special public input — the tweak except the normal plaintext, key and ciphertext. Changing the tweak is more easier than changing the keys. Therefore, it is very suitable for various complex applications such as encryption protocols, authentication encryption algorithms and disk encryptions [2], [3], to protect the confidentiality and authenticity of the data.

QARMA is a hardware-oriented lightweight tweakable block cipher proposed by Roberto Avanzi in 2017 [4]. It is inspired by PRINCE, MANTIS and Midori, but adopts different structure and components. QARMA applies a three-round Even-Mansour scheme with a non-involutory and keyed middle permutation. It has two kinds of block sizes, 64 bits and 128 bits, denoted by QARMA-64 and QARMA-128, respectively. Their tweak sizes equal the block sizes and the key sizes are twice as long as the block lengths. The designers hoped that it could be used for memory encryption, the generation of short tags and the construction of keyed hash functions because of its low latency.

In recent years, the meet-in-the-middle and impossible differential attacks are used to analyze the security of several famous block ciphers to obtain some good results. In 1999, Knudsen and Biham independently proposed the impossible differential attack [5], [6]. The basic idea is to construct a truncated differential with the probability of zero as a distinguisher to retrieve the correct subkey. This truncated

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

differential with the probability of zero is called an impossible differential, which is consisted of two truncated differentials with the probability of one resulting in a contradiction in the middle. Then they separately append several rounds E_0 and E_2 at its top and bottom to form an attacking path. According to it, the adversary chooses some plaintext-ciphertext pairs to satisfy the plaintext and ciphertext differences, and then guesses the related round subkeys of E_0 and E_2. If some plaintext-ciphertext pair can be encrypted or decrypted to obtain the input and output differences of impossible differential under some guessed round subkey, this round subkeys may be wrong and removed from the key space. Finally, the remaining subkey may be correct. Given enough plaintext-ciphertexts, the correct subkey should be recovered. Up to now, many new results have been presented to improve its efficiency, such as the early abort technique [7], the state-test technique [8], pre-computation tables [9] and automated algorithms [10]–[14].

The meet-in-the-middle attack was proposed by Diffie and Hellman in 1977 [15]. There are two kinds of attacking methods. The original attacking idea is to find the round subkeys K_1 and K_2 satisfying $E_1(P, K_1) = E_2^{-1}(C, K_2)$ for some plaintext-ciphertext (P, C), where $E = E_1 \circ E_2$ and K_1 and K_2 are the related round subkeys of E_1 and E_2. However, this method cannot attack an amount of block ciphers. Therefore, Demirci et al. proposed a new idea in 2008 to overcome this flaw [16]. They split a block cipher into three subciphers $E = E_0 \circ E_1 \circ E_2$. In the middle part E_1, the adversary constructs a distinguisher in a precomputation table, which is some correspondences mapping some active cell of the input to a specific cell of the output. Find the related round subkeys K_0 and K_2 of E_0 and E_2 such that $E_0(P, K_0)$ and $E_2^{-1}(C, K_2)$ can match with the precomputation table for some plaintext-ciphertext (P, C). However, this approach requires high memory to store the precomputation table. After that, researchers proposed several techniques such as multi-set [17], differential enumeration [17], sufficient tabulation [18] and key-dependent sieve [19]. Moreover, Guo et al. presented results on meet-in-the-middle attacks on generic Feistel constructions [20]–[22]. These two attacks have been used to analyze the securities of many famous block ciphers successfully, for example, Camellia [23]–[25], TWINE [26], [27], Piccolo [28], AES [29] and so on.

Up to now, the attacks on QARMA are mainly divided into cryptanalytic results on reduced-version QARMA with symmetric structures and asymmetric structures, respectively. For reduced-version QARMA with the asymmetric structure, there are several cryptanalytic results as follows. In 2016, Zong and Dong proposed an meet-in-the-middle attack on 10 rounds of QARMA-64 with 2^{53} chosen plaintexts, 2^{116} encryptions and 2^{116} blocks [30]. But they could not attack a lot of rounds because of high memory requirements. In 2018, Zong et al. gave an impossible differential attack on 11 rounds of QARMA-64 with 2^{61} chosen plaintexts, $2^{64.4}$ encryptions and 2^{64} blocks [31]. However, they only gave the time complexity to retrieve 48-bit round subkeys of QARMA-64. In 2019, Li et al. gave an statistical saturation cryptanalysis of 11 rounds of QARMA-128 with $2^{126.1}$ known plaintexts, $2^{126.1}$ encryptions and 2^{71} blocks [32]. For reduced-version QARMA with symmetric structures, there is only one cryptanalytic result, i.e., Li and Jin presented an meet-in-the-middle attack on 10 rounds of QARMA-128 with 2^{88} chosen plaintexts, 2^{156} encryptions and 2^{145} blocks in 2018 [33]. Since they selected two cells as the ordered sequence, their attack required more time and memory complexities. Clearly, papers [31], [32] and [33] are the previously best known results on QARMA-64/128. For the previous best result on QARMA-64 [31], they only recover 48 bits of round subkeys in the key recovery phase. Therefore, we try to construct other attacking paths such that more related round subkeys can be retrieved in the key recovery phase in order to improve the cryptanalytic results. For the previous best known result on QARMA-128, paper [32] can be success in a weak attack scenario — the known plaintext attack. But the data complexity is $2^{126.1}$, which is close to the full code book. In [33], the meet-in-the-middle distinguisher contains two cells as an ordered sequence, which make this attack require high time and memory complexities. In order to improve these cryptanalytic results, we need build some good distinguishers with one cell output as an ordered sequence, which will reduce the time and memory complexities.

In this paper, we propose improved impossible differential and meet-in-the-middle attacks on the block cipher QARMA-64/128. First, we construct an 4.5-round impossible differential of QARMA-64/128. Based on it, we add 5 rounds at its top and 1.5 rounds at its bottom to attack 11-round QARMA-64/128. The attack on 11-round QARMA-64 requires $2^{58.38}$ chosen plaintext-tweak combinations, 2^{69} encryptions (the time complexity in the key recovery phase is $2^{64.92}$ encryptions) and $2^{63.38}$ blocks, and the attack on 11-round QARMA-128 requires $2^{111.38}$ chosen plaintext-tweak combinations, 2^{137} encryptions and $2^{120.38}$ blocks, respectively. Second, we present the meet-in-the-middle attacks on 10-round symmetric QARMA-128 and the last 12 rounds of asymmetric QARMA-128, both of which contain the outer whitening key layer. For 10-round symmetric QARMA-128, we build a 4.5-round meet-in-the-middle distinguisher. By appending 3 rounds at its top and 2.5 rounds at its bottom, we mount a meet-in-the-middle attack on 10-round QARMA-128, which requires 2^{88} chosen plaintext-tweak combinations, $2^{164.43}$ encryptions and 2^{97} blocks, respectively. For 12-round asymmetric QARMA-128, we build a 6.5-round meet-in-the-middle distinguisher. By appending 3 rounds at its top and 2.5 rounds at its bottom, we present a meet-in-the-middle attack on 12-round QARMA-128, which requires 2^{88} chosen plaintext-tweak combinations, $2^{155.88}$ encryptions and 2^{154} blocks, respectively. In Table 1, we summarize our results along with previous attacks on QARMA-64/128. Compared with the previously best known results on QARMA-64/128 [31], [33], we can attack the same number of rounds with less complexities or recover more master keys. For 11-round QARMA-64, we can recover 16 more bits of master keys

with the almost complexities than the attack in paper [31]. For 10-round symmetric QARMA-128, the memory complexity of our attack are improved by a factor of 2^{48} compared with the attack in paper [33]. Moreover, we also propose an meet-in-the-middle attack on 12-round asymmetric QARMA-128, which is the best known results on QARMA-128 in terms of the number of rounds.

The rest of this paper is organized as follows. In Section 2, we briefly describe the QARMA block cipher and some related properties. In Sections 3, we present impossible differential attacks on 11-round QARMA-64/128. In Sections 4, we propose the meet-in-the-middle attack on 10-round symmetric QARMA-128 with the outer whitening key layer. In Sections 5, we propose the meet-in-the-middle attack on 12-round asymmetric QARMA-128 with the outer whitening key layer. Finally, we conclude this paper in Section 6.

II. THE DESCRIPTION OF QARMA

A. NOTATIONS

- P, C, tk: the plaintext, the ciphertext and the round tweakey;
- $\Delta P, \Delta C$: the differences of plaintext and ciphertext, respectively;
- X_i, Y_i, Z_i, V_i: the intermediate states after the AddRoundTweakey (ATK), ShuffleCells (SC), MixColumns (MC), SubBytes (SB) operations of the first forward i-th round, and the intermediate states before the ATK, SC, MC, SB operations of the last backward i-th round;
- X_m, Y_m, Z_m, V_m: the intermediate states before the ATK, SC, MC, SC operations in the middle rounds, respectively;
- $X_i[j], Y_i[j], Z_i[j], V_i[j]$: the j-th byte of X_i, Y_i, Z_i, V_i, where $0 \leq j < 16$.
- $\Delta X_i, \Delta Y_i, \Delta Z_i, \Delta V_i$: the differences of X_i, Y_i, Z_i, V_i, respectively;
- $\overline{f}$: the inverse of function f;
- $x \lll k$: left rotation shift of x by k bits;
- $x \ggg k$: right rotation shift of x by k bits;
- $x \gg k$: right shift of x by k bits;
- $X[i](k)$: the k-th bit of cell $X[i]$ and each cell has b bits numbered $0, 1, \ldots, b-1$ from left to right;
- $A \parallel B$: the concatenation of A and B;

B. THE BLOCK CIPHER QARMA

QARMA is a lightweight tweakable block cipher with the SPN (Substitution Permutation Network) round function. It has two versions of block sizes, i.e., 64 bits and 128 bits, denoted by QARMA-64 and QARMA-128, respectively. Their tweaks equal the blocks and the keys are twice as long as the blocks. The internal state of QARMA can be represented as a 4×4 matrix, each cell of which contains b bits with $b = 4$ for QARMA-64 and $b = 8$ for QARMA-128. The cells of marix are numbered in the following order.

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{pmatrix}$$

The encryption algorithm consists of the forward round function $\mathcal{F}$, the backward round function $\overline{\mathcal{F}}$ and the central round function $\mathcal{CR}$, where the backward round function $\overline{\mathcal{F}}$ is the inverse of the forward round function $\mathcal{F}$, and the central function is designed to be easily inverted. The detailed information can be seen in Figure 1.

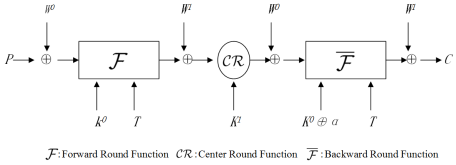


FIGURE 1. The encryption structure scheme of QARMA.

The forward round function $\mathcal{F}$ consists of four operations, which perform in the following order:

- AddRoundTweakey(ATK): The internal state is XORed to the round tweakey tk, where tk is the XOR of the round key, tweak and round constant.
- ShuffleCells(SC): The internal state is updated by a cell permutation P_T used in MIDORI, i.e., $P_T : (s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}) \mapsto (s_0, s_{11}, s_6, s_{13}, s_{10}, s_1, s_{12}, s_7, s_5, s_{14}, s_3, s_8, s_{15}, s_4, s_9, s_2)$.
- MixColumns(MC): The internal state is multiplied by a matrix M, where M is listed as in the following.

$$M = circ(0, \rho^a, \rho^f, \rho^c) = \begin{pmatrix} 0 & \rho^a & \rho^f & \rho^c \\ \rho^c & 0 & \rho^a & \rho^f \\ \rho^f & \rho^c & 0 & \rho^a \\ \rho^a & \rho^f & \rho^c & 0 \end{pmatrix}$$

  Among it, ρ^i is the left rotation shift of the element by i bits. For QARMA-64, $a = c = 1, f = 2$, and $M^{-1} = circ(0, \rho^1, \rho^2, \rho^1)$. For QARMA-128, $a = 1, b = 4$, $f = 5$, and $M^{-1} = circ(0, \rho^1, \rho^4, \rho^5)$.
- SubBytes(SB): Apply the S-Boxes in parallel for each cell of the internal state.

A short version of the round function used in the first and last rounds is to omit the ShuffleCells and MixColumns operations.

The central round function $\mathcal{CR}$ consists of two rounds, performed in the following order:

- A forward round function $\mathcal{F}$.
- The pseudo-reflector $\mathcal{CR}$ consists of ShuffleCells, MixColumns, AddRoundTweakey and InverseShuffleCells.
- A backward round function $\overline{\mathcal{F}}$.

QARMA$_r$ denotes $(2r+2)$-round QARMA, which is composed by r-round forward round functions, r-round backward

TABLE 1. Summary of attacks on QARMA in the single key scenario.

Cipher	Outer whitening	Symmetry	Rounds	Data	Time(EN)	Memory	Attack type	Source
QARMA-64	No	No	10	$2^{53}CPT$	2^{116}	2^{116}	MITM	[30]
	No	No	11	$2^{61}CPT$	$2^{64.4}$	2^{64}	IMD	[31]†
	No	No	11	$2^{58.38}CPT$	$2^{64.92}$	$2^{63.38}$	IMD	$SectionIII$‡
QARMA-128	Yes	Yes	10	$2^{88}CPT$	2^{156}	2^{145}	MITM	[33]
	Yes	No	11	$2^{126.1}KPT$	$2^{126.1}$	2^{71}	TDIB	[32]
	Yes	Yes	10	$2^{88}CPT$	$2^{164.48}$	2^{97}	MITM	$SectionIV$
	Yes	No	12	$2^{88}CPT$	$2^{156.06}$	2^{154}	MITM	$SectionV$
	No	No	11	$2^{111.38}CPT$	2^{137}	$2^{120.38}$	IMD	$SectionIII$

CPT/KPT: Chosen/Known Plaintext-Tweak Combination; ACC: Adaptive Chosen Ciphertexts; EN: Encryptions;
MITM: Meet-in-the-Middle Attacks; IMD: Impossible Differential Attacks; TDIB: Tweak Difference Invariant Bias;
†: Recover 48 bits of round subkeys in this attack; ‡: Recover 64 bits of master subkeys in this attack;

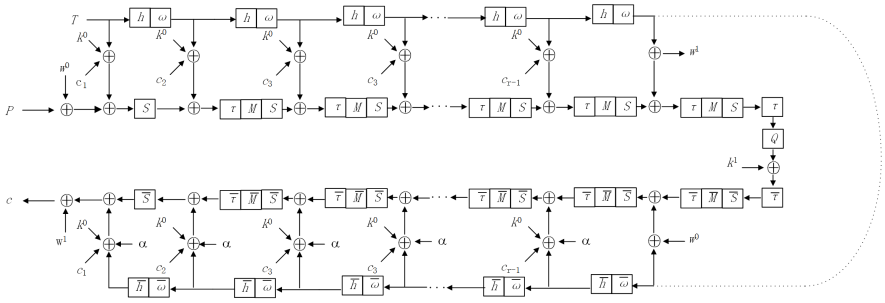


FIGURE 2. The structure of QARMA.

round functions and 2-round central functions. The detailed information can be seen in Figure 2 [4].

The Tweakey Schedule: The $32b$-bit master key is represented as $k^0 \parallel w^0$. Generate a subkey k^1 and w^1 by $k^1 = k^0$ and $w^1 = (w^0 \ggg 1) \oplus (w^0 \gg (16b-1))$. The tweak T is updated by two permutations h and ω. The permutation h is $h(T) = T[h(0)] \parallel \ldots \parallel T[h(15)]$, where h is defined by the permutation $h = [6, 5, 14, 15, 0, 1, 2, 3, 7, 12, 13, 4, 8, 9, 10, 11]$ used in MANTIS. A LFSR ω with the maximal period updates the tweak cells with indexes 0, 1, 3, 4, 8, 11 and 13. For QARMA-64, $\omega(b_3, b_2, b_1, b_0) = (b_0 + b_1, b_3, b_2, b_1)$. For QARMA-128, $\omega(b_7, b_6, \ldots, b_0) = (b_0 + b_2, b_7, b_6, \ldots, b_1)$.

C. DEFINITIONS AND PROPERTIES

Definition 1 [20]: A l-δ-set is a set of 2^l state values that are all different in l bits (active bits) and all equal in the remaining bits (inactive bits).

Property 1: For a given S-box differential $(\Delta x, \Delta y)$, the average number of solutions to $S(x) \oplus S(x \oplus \Delta x) = \Delta y$ is 1.

Property 2 [33]: Consider a pair (a, d) of 4-byte vectors such that $d = M \cdot a$. If $a = (a[0], a[1], a[2], a[3])$ and $d = (d[0], d[1], d[2], d[3])$, then $\rho^4 d[i] \oplus d[i+2] = \rho^4 \cdot a[i] \oplus \cdot a[i+2]$ for $i = 0, 1$.

Property 3: Consider a pair (a, d) of $4b$-bit vectors such that $d = M \cdot a$. Among it, only one cell of a equals zero for example $a[x] = 0$ with x =0 or 1 or 2 or 3, and the corresponding cell $d[x]$ is non-zero and other cells of d equal zero. Then the probability that happens is 2^{-2b}.

Proof: By the definition of M, we can obtain the following equations:

$$\begin{aligned} d[0] &= \rho^1 \cdot a[1] \oplus \rho^4 \cdot a[2] \oplus \rho^5 \cdot a[3], \\ d[1] &= \rho^5 \cdot a[0] \oplus \rho^1 \cdot a[2] \oplus \rho^4 \cdot a[3], \\ d[2] &= \rho^4 \cdot a[0] \oplus \rho^5 \cdot a[1] \oplus \rho^1 \cdot a[3], \\ d[3] &= \rho^1 \cdot a[0] \oplus \rho^4 \cdot a[1] \oplus \rho^5 \cdot a[2]. \end{aligned}$$

Consider a special scenario satisfying the condition, for example $a[3] = d[0] = d[1] = d[2] = 0$, and $a[0]$, $a[1]$, $a[2]$ and $d[3]$ are non-zero. That is to say $\rho^5 \cdot a[0] \oplus \rho^1 \cdot a[2] = \rho^1 \cdot a[1] \oplus \rho^4 \cdot a[2] = 0$. So $\rho^0\, a[0] \oplus \rho^4\, a[2] = 0$. We can obtain that $\rho^0 \cdot a[0] \oplus \rho^1 \cdot a[1] = 0$. Since $d[2] = \rho^4 \cdot a[0] \oplus \rho^5 \cdot a[1] = \rho^4 \cdot (\rho^0 \cdot a[0] \oplus \rho^1 \cdot a[1])$, we can get $d[2] = 0$. Therefore, this case happens with the probability 2^{-2b}. Similarly, other cases can be proved.

III. IMPOSSIBLE DIFFERENTIAL CRYPTANALYSIS OF 11-ROUND QARMA-64/128

In this section, we first construct a 4.5-round impossible differential distinguisher. Appending 5 rounds at its top and 1.5 rounds at its bottom, we attack 11-round QARMA-64/128. The detailed information can be found in the following.

A. A 4.5-ROUND IMPOSSIBLE DIFFERENTIAL DISTINGUISHER OF QARMA-64/128

In this sub-section, we present a 4.5-round impossible differential distinguisher of QARMA-64/128 in Proposition 1.

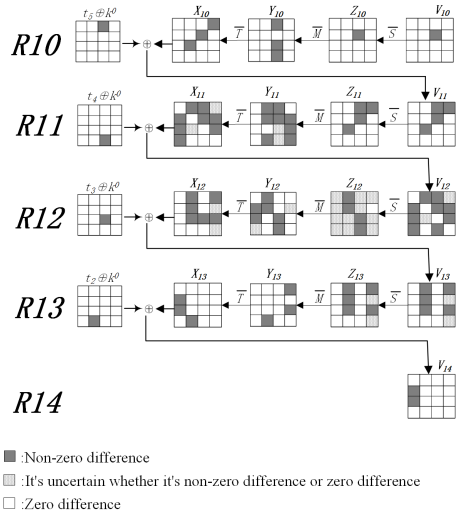


FIGURE 3. A 4.5-round impossible differential distinguisher of QARMA-64/128.

Proposition 1: The 4.5-round differential $\bigtriangleup \triangleq 000000\Delta V_{10}[6]\ 000000000 \nrightarrow_4 0000\Delta Z_{14}[4]000\Delta Z_{14}[8]0000000$ is impossible,where $\Delta V_{10}[6]$ and $\Delta Z_{14}[4, 8]$ are non-zero and the corresponding tweak $\Delta t_5[2]$ is the only active nibble of Δt_5. The detailed structure can be seen in Figure 3.

Proof: In the forward direction, we encrypt ΔV_{10} through one round to obtain the value of $\Delta V_{11} = 00\alpha_1\alpha_2 00\alpha_3 00\alpha_4 000000$ with $\alpha_i \neq 0 (1 \leq i \leq 4)$. Continue to encrypt ΔV_{11} through one round to obtain the value of $\Delta V_{12} = 0\alpha_5\alpha_6\beta_1\alpha_7 0\alpha_8\alpha_9\alpha_{10}\beta_2 0\alpha_{11}\alpha_{12}0\alpha_{13}0$ with $\alpha_i \neq 0 (5 \leq i \leq 13)$. In the backward direction, decrypt ΔZ_{14} through one round to obtain the value of $\Delta Z_{13} = 0\gamma_1 0\gamma_2 0\gamma_3 0\xi_1 0\gamma_4 0\gamma_5 000\xi_2$ with $\gamma_i \neq 0 (1 \leq i \leq 5)$. Continue to decrypt ΔZ_{13} to obtain the value of ΔZ_{12}. Clearly, $\Delta Z_{12}[7] = 0$, which is contradiction with $\Delta V_{12}[7] \neq 0$. Thus Δ is a 4.5-round impossible differential.

B. THE ATTACKING PROCEDURE OF 11-ROUND QARMA-64/128

Based on the 4.5-round impossible differential distinguisher above, we construct an 11-round impossible differential path of QARMA-64/128 by appending 5 rounds at its top and 1.5 rounds at its bottom. The detailed information can be seen in Figure 4. Our attack contains two phases, i.e., the data collection and key recovery phases. In data collection phase, we select some plaintext-ciphertext pairs satisfying the attacking path. Based on it, guess the round subkeys of the first 5 rounds and the last 1.5 rounds. If some plaintext-ciphertext can be encrypted or decrypted under the guessed subkeys to obtain the impossible differential, the guessed round subkeys will be removed from the key space. Finally, the correct round subkeys will be left. The detailed steps can be given in the following.

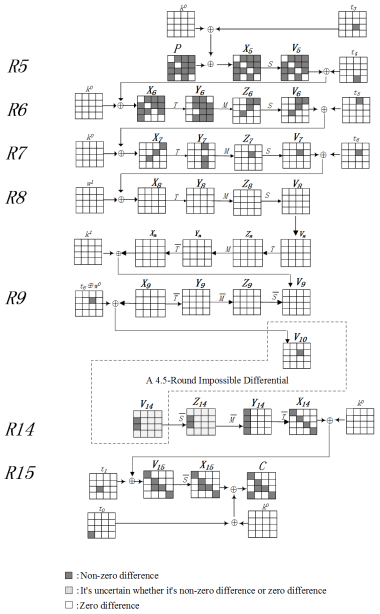


FIGURE 4. Impossible differential attacks on 11-round QARMA-64/128.

1) THE DATA COLLECTION PHASE

Define a structure of plaintexts, each of which takes all possible values at $P[1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 14]$ and $t_3[10]$ and fixed values at other bytes. One structure contains 2^{13b} plaintext-tweak combinations. Choose 2^n structures, so there are 2^{n+13b} plaintexts. Build a hash table to store these plaintexts indexed by the value of $C[1, 2, 3, 4, 6, 7, 8, 11, 13, 14] \parallel C[12] \oplus t_0[12]$. Thus we can collect about $2^{n+15b-1}$ plaintext-ciphertext pairs for QARMA-64/128. Keep the plaintext-ciphertext pairs satisfying $\Delta P[10] = \Delta t_3[10]$. So there are about $2^{n+14b-1}$ plaintext-ciphertext pairs left.

2) THE KEY ROCOVERY PHASE

Before presenting the key recovery phase, we build five precomputation tables to extract the related round subkeys so that the time complexity is reduced.

- H_1: Guess the values of $\Delta Z_6[2, 6] \parallel V_5[3, 6, 9, 12]$. Compute the values of $X_5[3, 6, 9, 12] \parallel \Delta X_5[3, 6, 9, 12]$. Store $\Delta Z_6[2, 6] \parallel X_5[3, 6, 9, 12]$ in H_1 indexed by $\Delta P[3, 6, 9, 12]$, where $\Delta P[3, 6, 9, 12] = \Delta X_5[3, 6, 9, 12]$. Thus H_1 has 2^{4b} rows, each of which has 2^{2b} values on average.
- H_2: Guess the values of $\Delta Z_6[3] \parallel V_5[2, 7, 8]$. Compute the values of $X_5[2, 7, 8] \parallel \Delta X_5[2, 7, 8]$. Store $\Delta Z_6[3] \parallel X_5[2, 7, 8]$ in H_2 indexed by $\Delta P[2, 7, 8]$, where $\Delta P[2, 7, 8] = \Delta X_5[2, 7, 8]$. Thus H_2 has 2^{3b} rows, each of which has 2^b values on average.
- H_3: Guess the values of $\Delta Z_6[9] \parallel V_5[1, 4, 11]$. Compute the values of $X_9[1, 4, 11] \parallel \Delta X_9[1, 4, 11]$. Store $\Delta Z_6[9] \parallel X_5[1, 4, 11]$ in H_3 indexed by $\Delta P[1, 4, 11]$, where $\Delta P[1, 4, 11] = \Delta X_5[1, 4, 11]$. Thus, H_3 has 2^{3b} rows, each of which has 2^b values on average.

- H_4: Guess the values of $\Delta Z_{14}[4, 8] \parallel V_{15}[0, 5, 10, 15]$. Compute the values of $X_{15}[0, 5, 10, 15] \parallel \Delta X_{15}[0, 5, 10, 15]$. Store $\Delta Z_{14}[4, 8] \parallel X_{15}[0, 5, 10, 15]$ in H_4 indexed by $\Delta C[0, 5, 10, 15]$, where $\Delta C[0, 5, 10, 15] = \Delta X_{15}[0, 5, 10, 15]$. Thus, H_4 has 2^{4b} rows, each of which has 2^{2b} values on average.
- H_5: Guess the values of $\Delta V_5[14] \parallel V_5[14]$. Compute the values of $X_5[14] \parallel \Delta X_5[14]$. Store $\Delta V_5[14] \parallel V_5[14] \parallel X_5[14]$ in H_5 indexed by $\Delta P[14] \parallel \Delta t_4[14]$, where $\Delta P[14] = \Delta X_5[14]$. Thus H_5 has 2^{2b} rows, each of which has only one values on average.

For each remaining plaintext pairs, do the following steps:

1) Access H_1 to get the value of $\Delta Z_6[2, 6] \parallel X_5[3, 6, 9, 12]$. So the value of $k^0[3, 6, 9, 12]$ can be computed by $k^0[3, 6, 9, 12] = P[3, 6, 9, 12] \oplus X_5[3, 6, 9, 12]$. There are 2^{2b} values of $k^0[3, 6, 9, 12]$ left. Next, encrypt the plaintexts to obtain the value of $Y_6[2, 6, 10, 14] \parallel Z_6[2, 6] \parallel \Delta Z_6[2, 6] \parallel V_6[2, 6] \parallel \Delta V_6[2, 6]$. Keep those subkeys satisfying $\Delta V_6[2] = \Delta t_3[2]$. Similarly, we can decrypt the ciphertexts to obtain the value of $\Delta V_{15}[9]$. Keep those subkeys satisfying $\Delta V_{15}[9] = \Delta t_1[9]$. Thus the remaining subkeys is about one.
2) Access H_2 to get the value of $\Delta Z_6[3] \parallel X_5[2, 7, 8]$. So the value of $k^0[2, 7, 8]$ can be computed by $k^0[2, 7, 8] = P[2, 7, 8] \oplus X_5[2, 7, 8]$. There are 2^b values of $k^0[2, 3, 6, 7, 8, 9, 12]$ left. Next, encrypt the plaintexts to obtain the value of $Y_6[7, 11, 15] \parallel Z_6[3] \parallel \Delta Z_6[3] \parallel V_6[3] \parallel \Delta V_6[3]$.
3) Access H_3 to get the value of $\Delta Z_6[9] \parallel X_5[1, 4, 11]$. So the value of $k^0[1, 4, 11]$ can be computed by $k^0[1, 4, 11] = P[1, 4, 11] \oplus X_5[1, 4, 11]$. There are 2^{2b} values of $k^0[1, 2, 3, 4, 6, 7, 8, 9, 11, 12]$ left. Next, encrypt the plaintexts to obtain the value of $Y_6[1, 5, 13] \parallel Z_6[9] \parallel \Delta Z_6[9] \parallel V_6[9] \parallel \Delta V_6[9]$. Since the value of $k^0[3, 6, 9]$ has known, we can compute the value of $\Delta Z_7[6] \parallel Z_7[6] \parallel \Delta V_7[6]$. Keep those subkeys satisfying $\Delta V_7[6] = \Delta t_6[6]$ and $\Delta Z_7[2, 10, 14] = 0$. Thus about one of 2^b subkeys will left by Property 3.
4) Access H_4 to get the value of $\Delta Z_{14}[4, 8] \parallel X_{15}[0, 5, 10, 15]$. So the value of $k^0[0, 5, 10, 15]$ can be computed by $k^0[0, 5, 10, 15] = C[0, 5, 10, 15] \oplus X_{15}[0, 5, 10, 15]$. There are 2^b values of $k^0[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15]$ left.
5) Access H_5 to get the value of $\Delta X_5[14] \parallel X_5[14]$. So the value of $k^0[14]$ can be computed by $k^0[14] = P[14] \oplus X_5[14]$. There are 2^b values of $k^0[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15]$ left.

C. COMPLEXITY ANALYSIS

There are 15 cells of round subkeys involved in our attack. For each plaintext-ciphertext pair, about 2^b possible round subkeys are removed by the key recovery. Thus, the probability that a wrong key is not discarded for one plaintext-ciphertext pair is $1 - 2^b/2^{15b}$. For $2^{n+14b-1}$ plaintext-ciphertext pairs, there are approximately $2^{15b} * (1 - 2^{-14b})^{2^{n+14b-1}} \approx 1$ remaining candidates. So $n = 6.38$ for QARMA-64, and $n = 7.38$ for QARMA-128. The data complexity is $2^{58.38}$ chosen plaintexts for QARMA-64 and $2^{111.38}$ chosen plaintexts for QARMA-128. The designers claimed that QARMA can offer n bits of security if no better attacks are possible than time $2^{n-g-\epsilon}$ with 2^g chosen or known {plaintext, ciphertext, tweak} triples for a small ϵ [4]. Therefore, for QARMA-64 the time complexity should be below $2^{128-58.38} = 2^{69.62}$ encryptions, for QARMA-128 the time complexity should be below $2^{256-111.38} = 2^{144.62}$ encryptions. In our attack, the time complexity of online phase is about $2^{n+14b-1} \times (2^{2b} + 2^b + 2^{2b} + 2^b + 2^b)/44 \approx 2^{n+16b-5.46}$ encryptions, i.e., about $2^{64.92}$ for QARMA-64 and $2^{129.92}$ for QARMA-128. At this time, we recover one subkeys $k^0[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15]$. Exhaustively search the remaining 17 cells of subkeys, the time complexity of which is about $2 \times 2^{17b} = 2^{17b+1}$ encryption, i.e., 2^{69} for QARMA-64 and 2^{137} for QARMA-128. Thus the whole time complexity is about 2^{69} encryptions for QARMA-64 and 2^{137} encryptions for QARMA-128. The memory complexity is about $2^{n+14b-1} \times 4 = 2^{n+14b+1}$ blocks, i.e., $2^{63.38}$ for QARMA-64 and $2^{120.38}$ for QARMA-128.

IV. IMPROVED A MEET-IN-THE-MIDDLE ATTACK ON 10-ROUND QARMA-128

In this section, we first construct a new 4.5-round meet-in-the-middle distinguisher in the offline phase. Then we attack 10-round QARMA-128 by appending 3 rounds at its top and 2.5 rounds at its bottom. The detailed information can be found in the following.

A. A 4.5-ROUND MEET-IN-MIDDLE DISTINGUISHER OF QARMA-128

The 4.5-round meet-in-the-middle distinguisher of QARMA-128 is presented as follows.

Proposition 2: Let the first 5 bits of $v_3[1]$ be active. These all 32 values form the δ-set, denoted by $\{v_3^0, v_3^1, \ldots, v_3^{31}\}$. Consider the encryptions of this δ-set through 4.5-round QARMA-128. Then the corresponding 248-bit ordered sequence $\Delta V_8[1] \triangleq (V_8^1[1] \oplus V_8^0[1], V_8^2[1] \oplus V_8^0[1], \ldots, V_8^{31}[1] \oplus V_8^0[1])$ only takes about 2^{96} values out of 2^{248} theoretical values. The detailed information can be found in Figure 5.

Proof: In the following, we will show that the ordered sequence $\Delta V_8[1]$ is determined by 12 bytes, i.e., $Z_5[0, 4, 12] \parallel V_6[1, 6, 9, 10, 11, 15] \parallel V_7[1, 9, 13]$. Specifically, we first select some corresponding tweaks carefully such that $\Delta t_3^i[1] = \Delta V_3^i[1]$. Then we can obtain $\Delta X_4^i = 0$ and $\Delta X_5[5] = \Delta t_4[5]$. Next, guessing the value of $Z_5[0, 4, 12]$, we can obtain the value of $\Delta V_5[0, 4, 12]$. Similarly, guessing the values of $V_6[1, 6, 9, 10, 11, 15] \parallel V_7[1, 9, 13]$, we can compute the ordered sequence $\Delta V_8[1]$.

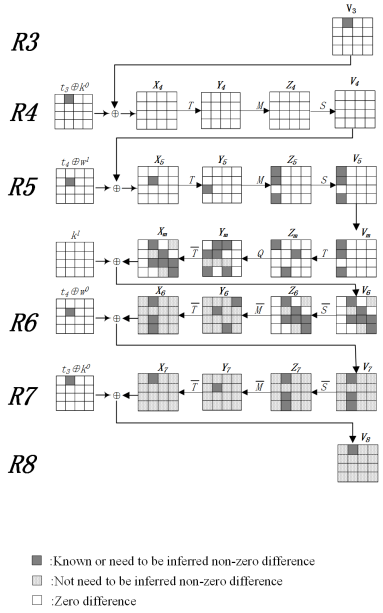


FIGURE 5. A 4.5-round distinguisher of QARMA-128.

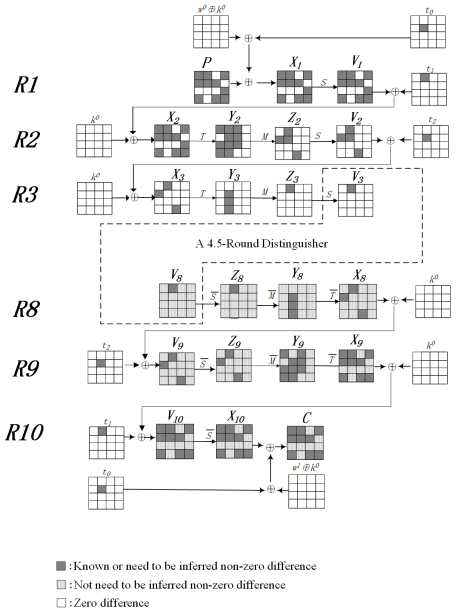


FIGURE 6. The Meet-in-the-middle attack on 10-round QARMA-128.

B. THE ATTACKING PROCEDURE OF 10-ROUND QARMA-128

By appending 3 rounds at the top and 2.5 rounds at the bottom of the 4.5-round distinguisher in Proposition 2, we build a 10-round attacking path of QARMA-128. Guess the related round subkeys of the first 3 rounds and the last 2.5 rounds, and compute the δ-set and the corresponding values of $\Delta V_8[1]$. If these values can match with the precomputation table, this guessed round subkeys may be correct. Otherwise, it will be removed from the key space. Finally, the correct subkeys will be retrieved. The detailed information can be found in Figure 6. In the following, we will present the meet-in-the-middle attack on 10-round QARMA-128.

The Offline Phase: Build a precomputation table H to store all the 2^{96} ordered sequences $\Delta V_8[1]$ in Proposition 2.

The Online Phase: In the online phase, choose a set of 2^{80} plaintexts where $P[0, 1, 3, 4, 5, 6, 11, 12, 14, 15]$ takes all the possible values while the other six bytes are constants. Ask for the encryption of each plaintext in the set under 2^8 tweaks where $t_0[5]$ takes all 2^8 possible values while the other 15 bytes are constants. In total, this attack requires 2^{88} plaintext-tweak combinations. For these pairs, execute the following steps.

1) Select a plaintext P^0 and its corresponding tweak T^0. Deduce the value $\Delta t_0, \Delta t_1, \Delta t_2$ by T^0. Guess the value of $(w^0 \oplus k^0)[0, 1, 3, 4, 5, 6, 11, 12, 14, 15]$. Encrypt one round to obtain the value of $V_1[0, 1, 3, 4, 5, 6, 11, 12, 14, 15]$.
2) Guess the value of $k^0[0, 1, 3, 4, 5, 6, 11, 12, 14, 15]$ and calculate the value of $V_3^0[1]$.
3) According to the δ-set, change the value of $V_3[1]$ to obtain the sequence $V_3^i[1](1 \le i \le 31)$. Since $\Delta V_3^i[1] = \Delta t_3^i[1]$, the value of corresponding tweak $T^i(1 \le i \le 31)$ can be known for each $V_3^i[1](1 \le i \le 31)$. By the guessed subkeys, we can compute the set of plaintexts $P^1, P^2, \ldots, P^{31}$.
4) Encrypt these plaintexts $P^0, P^1, \ldots, P^{31}$ to obtain the corresponding ciphertexts $C^0, C^1, \ldots, C^{31}$.
5) Because $w^1 = (w^0 \ggg 1) \oplus (w^0 \gg (16b - 1))$, we can calculate the values of $w^1[0, 1, 3, 4, 5, 6, 12, 14, 15]$ by the values of $(w^0 \oplus k^0)[0, 1, 3, 4, 5, 6, 11, 12, 14, 15] \| k^0[0, 1, 3, 4, 5, 6, 11, 12, 14, 15] \| w^0[2]|_7 \| w^0[13] |_7$. Thus, the value of $(w^1 \oplus k^0)[0, 1, 3, 4, 5, 6, 12, 14, 15]$ can be computed. Decrypt these ciphertexts $C^0, C^1, \ldots, C^{31}$ to obtain the value of $V_{10}[0, 1, 3, 4, 5, 6, 12, 14, 15]$.
6) Since the value of $k^0[0, 1, 3, 4, 5, 6, 12, 14, 15]$ has been guessed, we can calculate the value of $Z_9[1, 4, 14] \| \Delta Z_9[1, 4, 14] \| Z_8[1] \| \Delta Z_8[1]$. Finally, we can obtain the ordered sequence $\Delta V_8[1]$.
7) Verify whether this ordered sequence satisfies the precomputation table H or not.

In the online phase, we guess 162 bits in total, i.e., $(w^0 \oplus k^0)[0, 1, 3, 4, 5, 6, 11, 12, 14, 15] \| k^0[0, 1, 3, 4, 5, 6, 11, 12, 14, 15] \| w^0[2] |_7 \| w^0[13] |_7$. Thus about $1 + 2^{162} \times 2^{96}/2^{248} \approx 2^{10}$ round subkeys are left after the whole attack. Finally, retrieve the master key by the exhaustive search.

C. COMPLEXITY ANALYSIS

The upper data complexity of this attack is about 2^{88} plaintext-tweak combinations. The memory complexity is dominated by the size of precomputation table, i.e., $2^{96} \times 31 \times 8/2^7 \approx 2^{97}$ 128-bit blocks. The time complexity of offline phase is about $2^{96} \times 32 \times 12/160 \approx 2^{97.26}$ encryptions, and the time complexity of online phase is about $2^{162} \times 32 \times 28/160 \approx 2^{164.48}$ encryptions. Finally, we select two

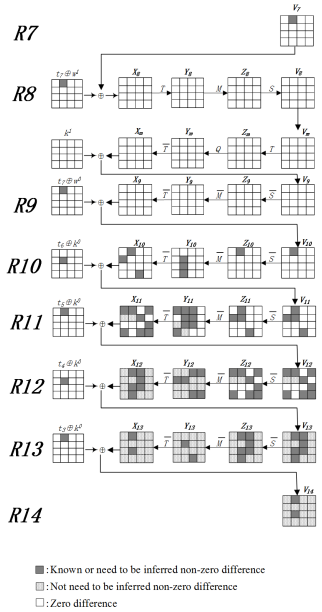


FIGURE 7. A 6.5-round distinguisher of QARMA-128.

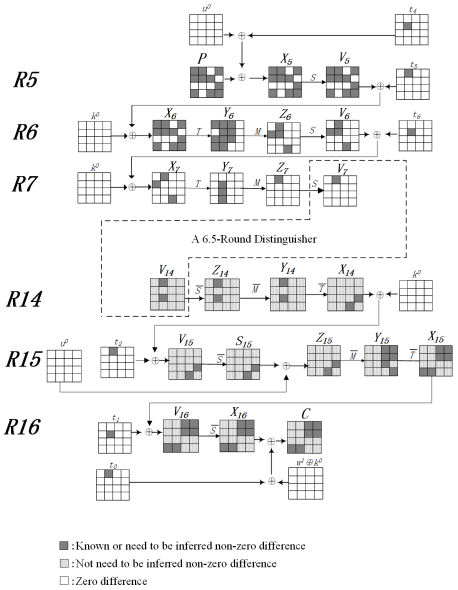


FIGURE 8. The Meet-in-the-middle attack on 12-round QARMA-128.

plaintext-ciphertext to search the master key. Thus the total time complexity is $2^{97.26} + 2^{164.48} + 2 \times 2^{96} \times 2^{10} \approx 2^{164.48}$ encryptions, which are less than $2^{256-88} = 2^{168}$ encryptions by [4].

V. THE MEET-IN-THE-MIDDLE ATTACK ON 12-ROUND QARMA-128

In this section, we first construct a 6.5-round meet-in-the-middle distinguisher in the offline phase. By appending 3 rounds at its top and 2.5 rounds at its bottom, we attack 12 rounds of QARMA-128. The attacking procedure is similar to the attack in section IV. The detailed information can be found in the following.

A. A 6.5-ROUND MEET-IN-MIDDLE DISTINGUISHER OF QARMA-128

The 6.5-round meet-in-the-middle distinguisher of QARMA-128 consists of two rounds of central round function and 4.5-round backward round functions. The detailed information can be found in Figure 7.

Proposition 3: Let the first 6 bits of $v_7[1]$ be active. These all 64 values form the δ'-set, denoted by $\{v_7^0, v_7^1, \ldots, v_7^{63}\}$. Consider the encryptions of this δ'-set through 6.5-round QARMA-128. Then the corresponding 504-bit ordered sequence $\Delta e \triangleq (e^1 \oplus e^0, e^2 \oplus e^0, \ldots, e^{63} \oplus e^0)$ only takes about 2^{152} values out of 2^{504} theoretical values, where $e^i \triangleq V_{14}^i[1] \oplus V_{14}^i[9]$ for $0 \le i \le 63$. The detailed information can be found in Figure 7.

Proof: In the following, we will show that the ordered sequence Δe is determined by 19 bytes, i.e., $V_{10}[1] \parallel V_{11}[1, 4, 5, 14] \parallel V_{12}[0, 1, 3, 6, 11, 12, 14, 15] \parallel V_{13}[1, 2, 6, 9, 10, 13]$. Specifically, select some corresponding tweaks carefully such that $\Delta t_7^i = \Delta V_7^i$. Then we can obtain $\Delta X_8^i = \Delta X_m^i = \Delta X_9^i = 0$ and $\Delta V_{10}[1] = \Delta t_7[1]$. Next, guessing the value of $V_{10}[1]$, we can obtain the value of $\Delta Z_{10}[1]$. Similarly, guessing the values of $V_{11}[1, 4, 5, 14] \parallel V_{12}[0, 1, 3, 6, 11, 12, 14, 15] \parallel V_{13}[1, 2, 6, 9, 10, 13]$, we can compute the ordered sequence Δe.

B. THE ATTACKING PROCEDURE OF 12-ROUND QARMA-128

By appending 3 rounds at the top and 2.5 rounds at the bottom of the 6.5-round distinguisher in Proposition 3, we build a 12-round attacking path of QARMA-128. The detailed information can be found in Figure 8. Because the operations ATK, SC and MC are linear, we equivalently swap ATK and SC with MC in our attack so as to reduce the time complexity. Let u^0 be $M(T(k^0))$, which is the equivalent subkey. In the following, we will present the meet-in-the-middle attack on 12-round QARMA-128 including the offline and online phases.

The Offline Phase: We first build a precomputation table H' to store all the 2^{152} ordered sequences Δe in Proposition 3.

The Online Phase: In the online phase, choose a set of 2^{80} plaintexts where $P[0, 1, 3, 4, 5, 6, 11, 12, 14, 15]$ takes all the possible values while the other six bytes are constants. Ask for the encryption of each plaintext in the set under 2^8 tweaks where $t_5[1]$ takes all the 2^8 possible values while the other 15 bytes are constants. In total, this attack requires 2^{88} plaintext-tweak combinations. For each plaintext-ciphertext pair, execute the following steps.

1) Select a plaintext P^0 and its corresponding tweak T^0. Deduce the value Δt_5 and Δt_6 by T^0. Guess the value of $k^0[0, 1, 3, 4, 5, 6, 11, 12, 14, 15]$. Encrypt one round to obtain the value of Z_6.
2) According to the value of $k^0[0, 1, 3, 4, 5, 6, 11, 12, 14, 15]$, calculate the value of $V_7^0[1]$.

3) According to the δ'-set, change the values of $V_7[1]$ to obtain the sequence $V_7^i[1](1 \leq i \leq 63)$. Since $\Delta V_7^i[1] = \Delta t_7^i[1]$, we can know the values of corresponding tweaks $T^i(1 \leq i \leq 63)$ for each $V_7^i[1](1 \leq i \leq 63)$. By the guessed subkeys, we can compute the set of plaintexts $P^1, P^2, \ldots, P^{63}$.
4) Encrypt these plaintexts $P^0, P^1, \ldots, P^{63}$ to obtain the corresponding ciphertexts $C^0, C^1, \ldots, C^{63}$.
5) Guess the value of $(w^1 \oplus k^0)[2, 3, 6, 7, 12, 13]$. Decrypt these ciphertexts $C^0, C^1, \ldots, C^{63}$ to obtain the value of $Z_{15}[11, 14]$.
6) The value of $u^0[14]$ can be calculated by the value of $k_0[3, 6, 12]$. Guess the value of $u^0[11]$ to calculate the value of $V_{15}[11, 14]$. According to the value of $k^0[11, 14]$, we can calculate the value of $Y_{14}[1, 9] \parallel \Delta Y_{14}[1, 9]$. By Property 2, we can calculate the value of $\rho^4 \Delta Y_{14}[1] \oplus \Delta Y_{14}[9] \parallel \rho^4 \Delta Z_{14}[1] \oplus \Delta Z_{14}[9]$.
7) Guess the value of $\Delta Z_{14}[1] \parallel Z_{14}[1]$. Calculate the value of $\Delta Z_{14}[9] \parallel Z_{14}[9]$. Then $\Delta V_{14}[1, 9]$ can be computed. Finally, the ordered sequence Δe can be calculated.
8) Verify whether this ordered sequence satisfy the precomputation table H' or not.

In the online phase, we guess 152 bits in total, i.e., $k^0[0, 1, 3, 4, 5, 6, 11, 12, 14, 15] \parallel u^0[11] \parallel (w^1 \oplus k^0)[2, 3, 6, 7, 12, 13] \parallel \Delta Z_{14}[1] \parallel Z_{14}[1]$. Thus about $1 + 2^{152} \times 2^{152}/2^{504} \approx 1$ round subkeys are left after the whole attack. Finally, retrieve the master key by the exhaustive search.

C. COMPLEXITY ANALYSIS

The upper data complexity of this attack is about 2^{88} plaintext-tweak combinations. The memory complexity is dominated by the size of precomputation table, i.e., $2^{152} \times 63 \times 8/2^7 \approx 2^{154}$ 128-bit blocks. The time complexities of offline and online phases are about $2^{152} \times 64 \times 19/192 \approx 2^{154.66}$ encryptions and $2^{152} \times 64 \times 25/192 \approx 2^{155.06}$ encryptions, respectively. Finally, we select two plaintext-ciphertexts to search the master key. Thus the total time complexity is $2^{154.66} + 2^{155.06} + 2 \times 2^{152} \times 1 \approx 2^{156.06}$ encryptions, which is less than $2^{256-88} = 2^{168}$ encryptions by [4].

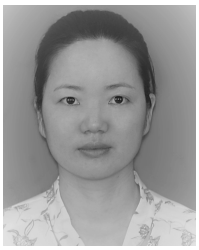
VI. CONCLUSION

This paper proposes four cryptanalytic results of QAMA-64/128 against impossible differential or meet-in-the-middle attacks. First, construct a 4.5-round impossible differential and mount an impossible differential attack on 11-round QARMA-64/128, which can recover 16 more bits of master key of QARMA-64 with the almost complexities compared with the paper [31]. Second, build a 4.5-round meet-in-the-middle distinguisher to propose an attack on 10-round symmetric QARMA-128 with less memory complexity. This attack requires 2^{88} chosen plaintext-tweak combinations, $2^{164.48}$ encryptions and 2^{97} blocks, respectively. Compared with paper [33], the memory complexity can be reduced by 2^{-48} times. Third, construct a 6.5-round meet-in-the-middle distinguisher to attack 12-round asymmetric QARMA-128 with 2^{88} chosen plaintext-tweak combinations, $2^{156.06}$ encryptions and 2^{154} blocks, which is the best known results on QARMA-128 with the same reduced-version strucutre. In a word, our cryptanalytic results show the lower security bounds of QARMA.

REFERENCES

[1] M. Liskov, R. L. Rivest, and D. Wagner, "Tweakable block ciphers," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 2002, pp. 31–46.
[2] T. Krovetz and P. Rogaway, "The software performance of authenticated-encryption modes," in *Proc. FSE*, Lyngby, Denmark, 2014, pp. 306–327.
[3] T. Peyrin and Y. Seurin, "Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers," in *Proc. CRYPTO*, Sacramento, CA, USA, 2016, pp. 33–63.
[4] R. Avanzi, "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," *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 1, pp. 4–44, Mar. 2017.
[5] L. R. Knudsen, "DEAL a 128-bit block cipher," *Complexity*, vol. 258, no. 2, p. 216, 1998.
[6] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," in *Proc. EUROCRYPT*, Prague, Czech Republic, 1999, pp. 12–23.
[7] J. Lu and J. Kim, "Improving the efficiency of impossible differential cryptanalysis of reduced Camellia and MISTY1," in *Proc. CT-RSA*, San Francisco, CA, USA, 2008, pp. 370–386.
[8] C. Boura, M. Naya-Plasencia, and V. Suder, "Scrutinizing and improving impossible differential attacks: Applications to CLEFIA, Camellia, LBlock and Simon," in *Proc. ASIACRYPT*, Kaoshiung, Taiwan, 2014 pp. 179–199.
[9] M. Tolba, A. Abdelkhalek, and A. M. Youssef, "Impossible differential cryptanalysis of reduced-round skinny," in *Proc. AFRICACRYPT*, Dakar, Senegal, 2017, pp. 117–134.
[10] J. Kim, S. Hong, and J. Lim, "Impossible differential cryptanalysis using matrix method," *Discrete Math.*, vol. 310, no. 5, pp. 988–1002, Mar. 2010.
[11] Y. Luo, X. Lai, Z. Wu, and G. Gong, "A unified method for finding impossible differentials of block cipher structures," *Inf. Sci.*, vol. 263, pp. 211–220, Apr. 2014.
[12] Y. Luo and X. Lai, "Improvement for finding impossible differentials of block cipher structures," *Secur. Commun. Netw.*, vol. 2017, Aug. 2017, Art. no. 5980251, doi: 10.1155/2017/5980251.
[13] Y. Sasaki and Y. Todo, "New impossible differential search tool from design and cryptanalysis aspects," in *Proc. EUROCRYPT*, Paris, France, 2017, pp. 185–215.
[14] Y. L. Ding, X. L. Wang, and N. Wang, "Improved automatic search of impossible differentials for Camellia with FL/FL^{-1} layers," *Sci. China Inf. Sci.*, vol. 61, Aug. 2017, Art. no. 038103, doi: 10.1007/s11432-016-9104-3.
[15] W. Diffie and M. Hellman, "Special feature exhaustive cryptanalysis of the NBS data encryption standard," *Computer*, vol. 10, no. 6, pp. 74–84, Jun. 1977.
[16] H. Demirci and A. A. Selçuk, "A meet-in-the-middle attack on 8-round AES," in *Proc. Fast Softw. Encryption*, Lausanne, Switzerland, 2008, pp. 116–126.
[17] O. Dunkelman, N. Keller, and A. Shamir, "Improved single-key attacks on 8-round AES-192 and AES-256," *J. Cryptol.*, vol. 28, no. 3, pp. 397–422, Jul. 2015.
[18] P. Derbez, P. A. Fouque, and J. Jean, "Improved key recovery attacks on reduced-round AES in the single-key setting," in *Proc. EUROCRYPT*, Athens, Greece, 2013, pp. 371–387.
[19] L. Li, K. Jia, and X. Y. Wang, "Improved single-key attacks on 9-round AES-192/256," in *Proc. Fast Softw. Encryption*, London, U.K., 2014, pp. 127–148.
[20] J. Guo, J. Jean, I. Nikolić, and Y. Sasaki, "Meet-in-the-middle attacks on generic feistel constructions," in *Proc. ASIACRYPT*, Kaoshiung, Taiwan, 2014 pp. 458–477.

[21] J. Guo, J. Jean, I. Nikolić, and A. Shamir, "Extended meet-in-the-middle attacks on some Feistel constructions," *Des., Codes Cryptogr.*, vol. 80, no. 3, pp. 587–618, Sep. 2016.
[22] J. Guo, J. Jean, I. Nikolić, and A. Shamir, "Meet-in-the-middle attacks on classes of contracting and expanding feistel constructions," *IACR Trans. Symmetric Cryptol.*, vol. 2016, no. 2, pp. 307–337, Feb. 2017.
[23] W.-L. Wu, W.-T. Zhang, and D.-G. Feng, "Impossible differential cryptanalysis of reduced-round ARIA and Camellia," *J. Comput. Sci. Technol.*, vol. 22, no. 3, pp. 449–456, May 2007.
[24] L. Li, K. Jia, X. Wang, and X. Dong, "Meet-in-the-middle technique for truncated differential and its applications to CLEFIA and Camellia," in *Proc. Fast Softw. Encryption*, İstanbul, Turkey, 2015, pp. 48–70.
[25] Y. Liu, L. Li, and D. Gu, "New observations on impossible differential cryptanalysis of reduced-round Camellia," in *Proc. Fast Softw. Encryption*, Washington, DC, USA, 2012, pp. 90–109.
[26] Y. Liu, A. Yang, B. Dai, W. Li, Z. Liu, D. Gu, and Z. Zeng, "Improved meet-in-the middle attacks on reduced-round TWINE-128," *Comput. J.*, vol. 61, no. 8, pp. 1252–1258, Aug. 2018.
[27] A. Biryukov, P. Derbez, and P. Léo, "Differential analysis and meet-in-the-middle attack against round-reduced TWINE," in *Proc. Fast Softw. Encryption*, Istanbul, Turkey, 2015, pp. 3–27.
[28] Y. Liu, L. Cheng, and Z. Liu, "Improved meet-in-the-middle attacks on reduced-round Piccolo," *Sci. China Inf. Sci.*, vol. 61, Nov. 2017, Art. no. 032108, doi: 10.1007/s11432-016-9157-y.
[29] R. Li and C. Jin, "Meet-in-the-middle attacks on 10-round AES-256," *Des., Codes Cryptogr.*, vol. 80, no. 3, pp. 459–471, Sep. 2016.
[30] R. Zong and X. Y. Dong. *Meet-in-the-Middle Attack on QARMA Block Cipher*. Accessed: Dec. 18, 2016. [Online]. Available: http://eprint.iacr.org/2016/1160
[31] R. Zong, X. Y. Dong, and X. Y. Wang. *MILP-Aided Related-Tweak/Key Impossible Differential Attack and Its Applications to QARMA, Joltik-BC*. Accessed: Feb. 6, 2018. [Online]. Available: https://eprint.iacr.org/2018/142
[32] M. Z. Li, K. Hu, and M. Q. Wang, "Related-tweak statistical saturation cryptanalysis and its application on QARMA," *IACR Trans. Symmetric Cryptol.*, vol. 2019, no. 1, pp. 236–263, Mar. 2019.
[33] R. Li and C. Jin, "Meet-in-the-middle attacks on reduced-round QARMA-64/128," *Comput. J.*, vol. 61, no. 8, pp. 1158–1165, Aug. 2018.



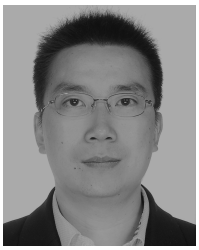
YA LIU was born in Maanshan, Anhui, China, in 1983. She received the B.S. and M.S. degrees in mathematics from Anhui Normal University, in 2004 and 2007, respectively, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, in 2013. She is currently an Associate Professor with the Department of Computer Science and Engineering, University of Shanghai for Science and Technology, Shanghai, China. Her research interests include applied cryptography, network security, cloud computing, the design and analysis of symmetric ciphers, and computational number theory.

TIANDE ZANG was born in Qingdao, Shandong, China, in 1995. He received the B.S. degree from the Qilu University of Technology, Jinan, China. He is currently pursuing the M.S. degree in computer technology with the University of Shanghai for Science and Technology, Shanghai, China. His research interest mainly includes the designs of analysis of symmetric ciphers.

DAWU GU received the M.S. and Ph.D. degrees from Xidian University, China, in 1995 and 1998, respectively. He is currently a Professor with the Computer Science and Engineering Department, Shanghai Jiao Tong University. His research interests include cryptology and computer security. He serves as a technical committee member of China Association of Cryptologic Research (CACR) and China Computer Federation (CCF), and a member of ACM, IACR, and IEICE. He received the New Century Excellent Talent Program made by Ministry of Education of China, in 2005. He has received more than 100 scientific papers in academic journals and conferences. He has been invited as Chair and TPC member for many international conferences, such as E-Forensics, ISPEC, ICIS, ACSA, and CNCC.

FENGYU ZHAO received the M.S. degree from the Nanjing University of Aeronautics and Astronautics, in 1989, and the Ph.D. degree from Fudan University, in 2010. He is currently a Professor with the Department of Computer Science and Engineering, University of Shanghai for Science and Technology. His research interests mainly include software engineer and web security.

WEI LI received the M.S. and Ph.D. degrees from Shanghai Jiao Tong University, in 2005 and 2009, respectively. She is currently a Professor with the School of Computer Science and Engineering, Donghua University. Her research interest mainly includes the design and analysis of symmetric ciphers. She serves as a member of China Association of Cryptologic Research (CACR), China Computer Federation (CCF), and ACM.



ZHIQIANG LIU received the B.S. and Ph.D. degrees from Shanghai Jiao Tong University, in 1998 and 2012, respectively. From 2001 to 2008, he worked in ZTE, Alcatel, and VLI in the realm of Next Generation Network (NGN)/IP Multimedia Subsystem (IMS). He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests mainly include block chain and symmetric ciphers.