The QARMA Block Cipher Family

Roberto Avanzi

Qualcomm Product Security Germany

Tokyo, March 7, 2017

For industry, developing a new cipher is *expensive**. Deploying it is *risky*:

With great power comes great responsibility.

Hence, motivation must come from very strong use cases, …

* Because qualified human resources are expensive. And, by the way, QPSI is hiring…

For industry, developing a new cipher is *expensive*[*]. Deploying it is *risky*:

With great power comes great responsibility.

Hence, motivation must come from very strong use cases, ...

[*] Because qualified human resources are expensive. And, by the way, QPSI is hiring...

For industry, developing a new cipher is *expensive**. Deploying it is *risky*:

With great power comes great responsibility.

Hence, motivation must come from very strong use cases, ...

* Because qualified human resources are expensive. And, by the way, QPSI is hiring...

(... use cases) where "transparent" performance is the difference between possible customer acceptance and outright feature rejection:

Memory Encryption

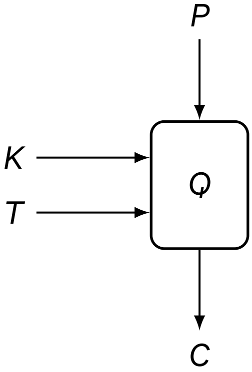
Software Security

(… use cases) where "transparent" performance is the difference between possible customer acceptance and outright feature rejection:

Memory Encryption

Software Security

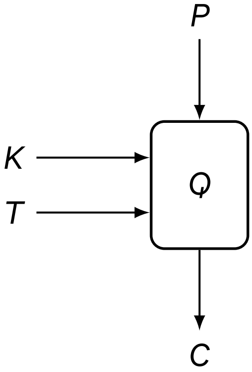
Tweakable Block Ciphers and applications



- **Memory encryption**: Just directly use address/nonce as tweak; no expensive XEX-like whitening value derivation:
  Reduced initial latency – direct impact on performance!

- **Software security**: SW exploits that manipulate *pointers*.
  Mitigations: Encrypt or hash these pointers...
  But: Decipher before use and/or increased memory traffic...
  Note: ARMv8 has 64-bit pointers and 52-bit address space
  Idea: Use a TBC to compute tag, truncated to just a few bits,
    key set by higher execution environment
    tweak = pointer's context
  then insert the tag in unused bits of the pointer!

Tweakable Block Ciphers and applications



- **Memory encryption**: Just directly use address/nonce as tweak; no expensive XEX-like whitening value derivation:

  Reduced initial latency – direct impact on performance!

- **Software security**: SW exploits that manipulate *pointers*.

  Mitigations: Encrypt or hash these pointers...

  But: Decipher before use and/or increased memory traffic...

  Note: ARMv8 has 64-bit pointers and 52-bit address space

  Idea: Use a TBC to compute tag, truncated to just a few bits,

  key set by higher execution environment
  tweak = pointer's context

  then insert the tag in unused bits of the pointer!

We had a look at all generic constructions and available primitives, but…

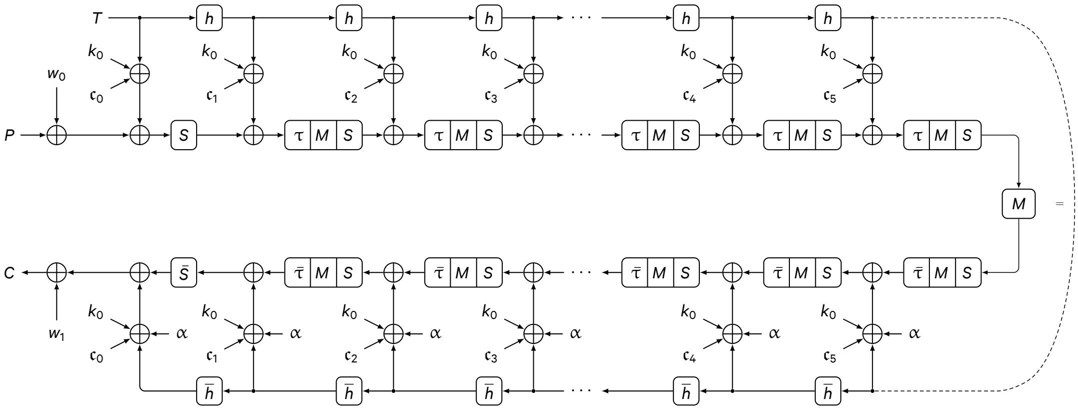
… they were all too large or too slow.

Timing requirements point to a "real TBC"
with low latency but no critical restrictions on total area.

We want a cipher that goes well fully unrolled, pipelined.

… a "TWEAKED-PRINCE," a bit fatter, but not much taller,
than PRINCE.

I took the train from Munich to Bochum ... and MANTIS was born



Maria Eichlseder described it so well I could only do worse...

Texts / tweak / state = vectors of sixteen 4-bit cells / 4×4 matrices

τ, h = Cell Shuffles, M = Involutory Almost MDS 4×4 matrix, S = S-Box layer

$\tau \circ M \circ S$ related to MIDORI round function – lighter than PRINCE's to offset the additional rounds.

Beyond MANTIS

I had second thoughts about the 4-round SuperBox in the middle,
some partners about the (re)use of MIDORI components.

So I had to go back to the drawing board. Boring: spice it with mathematics.

1. New structure
2. Better diffusion matrices
3. Better S-Boxes (and new heuristics to find them)
4. Provide a 128-bit variant with 256-bit key

Shortly after that, security margins of MANTIS eroded a bit.

Outcome: MANTIS has a new cousin …

… a cipher partly designed on the slopes of the Mt. Carmel …

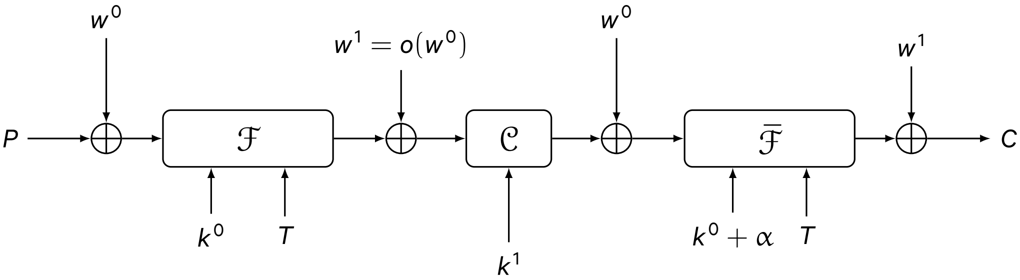
$$\underset{\substack{\downarrow\\ \text{Qualcomm}}}{\mathtt{Q}} + \overbrace{\mathtt{A} + \mathtt{R} + \mathtt{M}}^{\text{ARM}} + \overset{\substack{\text{Authenticator}\\ \uparrow}}{\mathtt{A}}$$

Roberto M. Avanzi

(and it might badly affect my karma)

1. New structure

QARMA has a new Structure

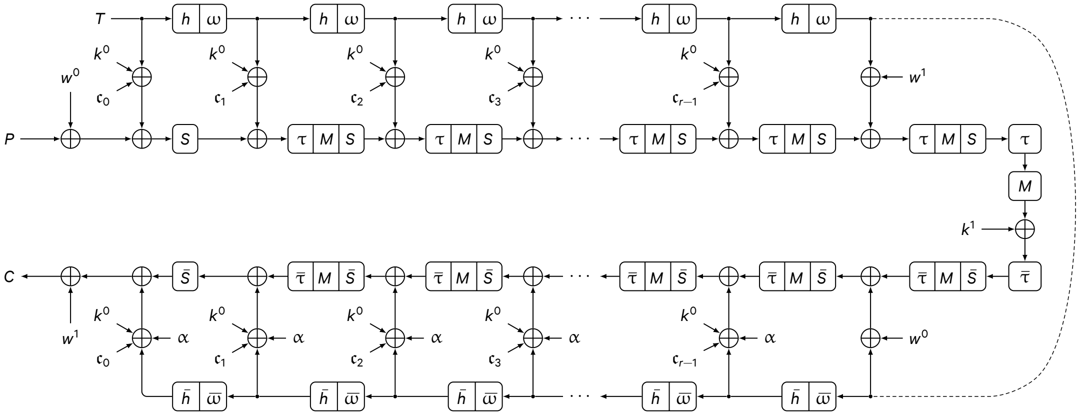


Whitening key derivation is s.t. $w^0 \mapsto w^1$ and $w^0 \mapsto w^0 + w^1$ both 1-1 (orthomorphism)

It is a 3-round, 2-key, alternating-key (non ideal) Even-Mansour scheme

(TD tradeoff may increase from $TD \geqslant^{n-\epsilon}$ to $TD \geqslant 2^{\frac{3}{2}n-\epsilon}$)

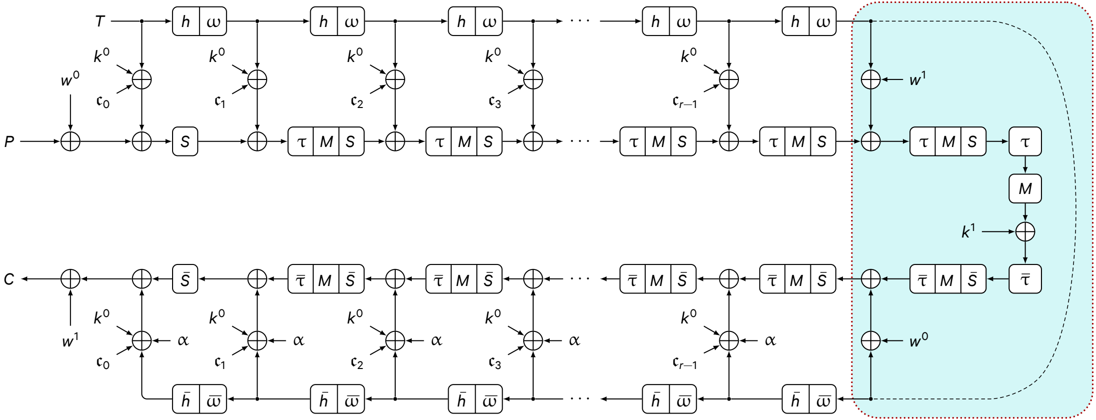
QARMA Encryption



Texts / tweak / state = vectors of sixteen 4-bit cells / 4×4 matrices

τ, h = Cell Shuffles; M = involutory Almost MDS 4×4 matrix; S = S-Box layer; ω = LSFR on 7/16 cells

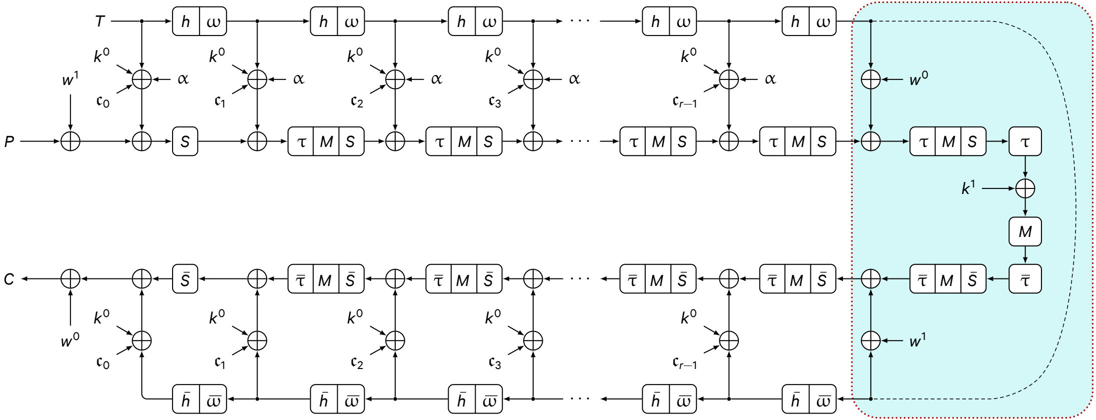
QARMA Encryption



Texts / tweak / state = vectors of sixteen 4-bit cells / 4×4 matrices

τ, h = Cell Shuffles; M = involutory Almost MDS 4×4 matrix; S = S-Box layer; ω = LSFR on 7/16 cells

QARMA Decryption

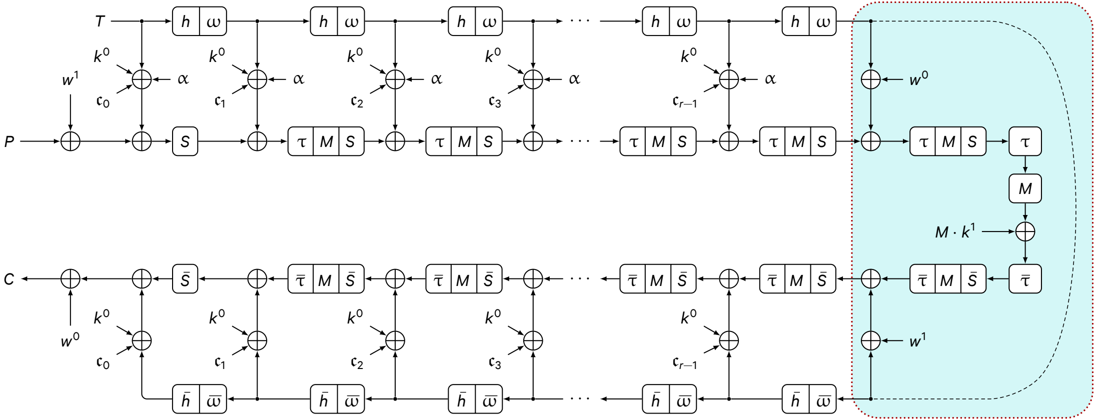


Texts / tweak / state = vectors of sixteen 4-bit cells / 4 × 4 matrices

τ, h = Cell Shuffles; M = involutory Almost MDS 4 × 4 matrix; S = S-Box layer; ω = LSFR on 7/16 cells

Decrypt with: $k^0 \mapsto k^0 \oplus \alpha$, swap w^0 and w^1

QARMA Decryption

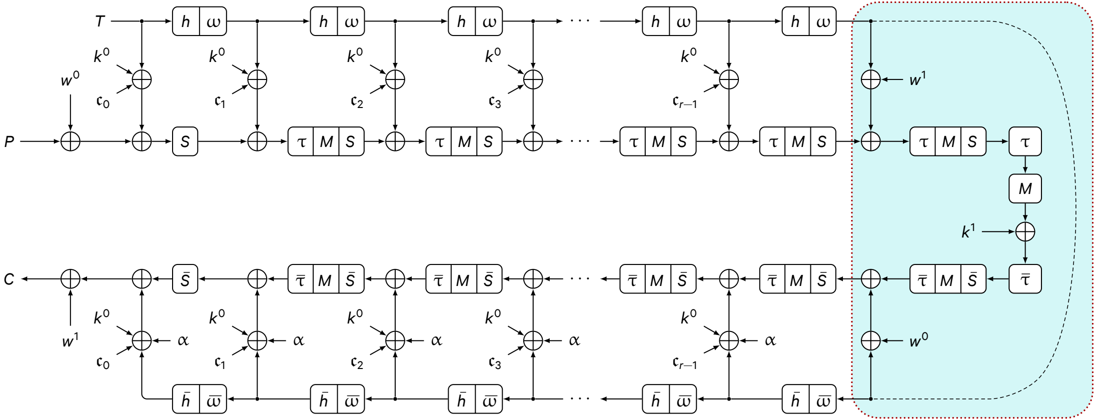


Texts / tweak / state = vectors of sixteen 4-bit cells / 4 × 4 matrices

τ, h = Cell Shuffles; M = involutory Almost MDS 4 × 4 matrix; S = S-Box layer; ω = LSFR on 7/16 cells

Decrypt with: $k^0 \mapsto k^0 \oplus \alpha$, swap w^0 and w^1, replace $k^1 \mapsto M \cdot k^1$

QARMA Encryption

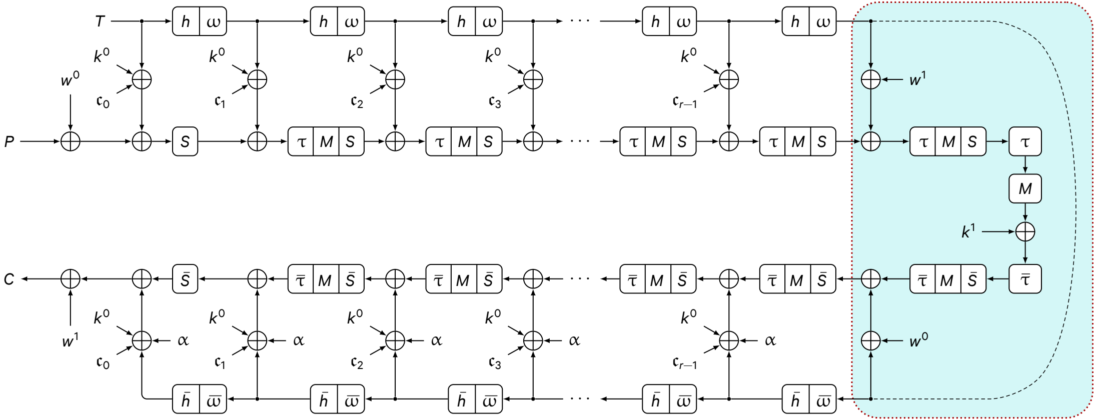


Texts / tweak / state = vectors of sixteen 4-bit cells / 4×4 matrices

τ, h = Cell Shuffles; M = involutory Almost MDS 4×4 matrix; S = S-Box layer; ω = LSFR on 7/16 cells

Decrypt with: $k^0 \mapsto k^0 \oplus \alpha$, swap w^0 and w^1, replace $k^1 \mapsto M \cdot k^1$

Impact of new central construction



Use of whitening key(s) instead of core key thwarts reflection attacks

Non involutory, keyed *Pseudo-Reflector* also makes reflection attacks more difficult

τ and $\bar{\tau}$ around it improve diffusion, kill 4-round SuperBox

2. Better diffusion matrices

MIDORI and MANTIS: Almost MDS Matrix circ(0, 1, 1, 1)

Represent state as a 4×4 matrix:

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

Diffusion layer based on Almost MDS matrix

$$M = \mathrm{circ}(\,0\,,\,1\,,\,1\,,\,1\,) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} .$$

A feature of the MIDORI matrix

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} v_0 \\ v_1 \\ \cdots \\ \cdots \end{pmatrix} = \begin{pmatrix} v_1 \oplus \cdots \\ v_0 \oplus \cdots \\ (v_0 \oplus v_1) \oplus \cdots \\ (v_0 \oplus v_1) \oplus \cdots \end{pmatrix}$$

Two S-Boxes copied, same addition twice – characteristics propagate unchanged and easily controlled.

QARMA: Almost MDS Matrix over a ring that encodes circular rotations

Cell values = vector space $\mathbb{F}_2^m$ ($m = 4$ or 8) with basis $\{\rho^{m-1}, \ldots, \rho^2, \rho, 1\}$ and $\rho^m = 1$. So we have a ring $R = \mathbb{F}_2[\rho]$ where ρ "=" *circular rotation to the left by one place*. We consider matrices over R of form

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

These matrices are as expensive (area, latency) as the $\{0, 1\}$-matrices.

We classify them (see paper): e.g. the involutory ones.

Choice of Matrices for QARMA

Example with $m = 8$ (involutory):

$$\begin{pmatrix} 0 & \rho^1 & \rho^4 & \rho^5 \\ \rho^5 & 0 & \rho^1 & \rho^4 \\ \rho^4 & \rho^5 & 0 & \rho^1 \\ \rho^1 & \rho^4 & \rho^5 & 0 \end{pmatrix} \times \begin{pmatrix} v_0 \\ v_1 \\ \cdots \\ \cdots \end{pmatrix} = \begin{pmatrix} (v_1 \lll 1) \oplus \cdots \\ (v_0 \lll 5) \oplus \cdots \\ \boxed{(v_0 \lll 4) \oplus (v_1 \lll 5)} \oplus \cdots \\ \boxed{(v_0 \lll 1) \oplus (v_1 \lll 4)} \oplus \cdots \end{pmatrix} \begin{matrix} \\ \\ \Delta = 1 \\ \Delta = 3 \end{matrix}$$

Then next S-Box layer more likely to disrupt characteristics (linear, differential, etc), or at least to avoid copy-and-paste.

Select values heuristically by minimising differentials over 1.5 rounds.

Choice of Matrices for QARMA

Example with $m = 8$ (involutory):

$$\begin{pmatrix} 0 & \rho^1 & \rho^4 & \rho^5 \\ \rho^5 & 0 & \rho^1 & \rho^4 \\ \rho^4 & \rho^5 & 0 & \rho^1 \\ \rho^1 & \rho^4 & \rho^5 & 0 \end{pmatrix} \times \begin{pmatrix} v_0 \\ v_1 \\ \cdots \\ \cdots \end{pmatrix} = \begin{pmatrix} (v_1 \lll 1) \oplus \cdots \\ (v_0 \lll 5) \oplus \cdots \\ \boxed{(v_0 \lll 4) \oplus (v_1 \lll 5)} \oplus \cdots \\ \boxed{(v_0 \lll 1) \oplus (v_1 \lll 4)} \oplus \cdots \end{pmatrix} \begin{array}{l} \\ \\ \Delta = 1 \\ \Delta = 3 \end{array}$$

Then next S-Box layer more likely to disrupt characteristics (linear, differential, etc), or at least to avoid copy-and-paste.

Select values heuristically by minimising differentials over 1.5 rounds.

3. Fantastic S-Boxes

and where to find them

S-Box Search Heuristics

Most important property in our context: total latency

Logic synthesis of a circuit is expensive and slow.
Cannot synthesise billions of S-boxes.

Idea: apply crude heuristics based on Quine-McCluskey to bound the depth of individual output bits. Take max. Minimise it.

Use variant of Prissette's algorithm to enumerate involutions with a predetermined subset of fixed points.

S-Box Search Heuristics

Most important property in our context: total latency

Logic synthesis of a circuit is expensive and slow.
Cannot synthesise billions of S-boxes.

Idea: apply crude heuristics based on Quine-McCluskey to bound the depth of individual output bits. Take max. Minimise it.

Use variant of Prissette's algorithm to enumerate involutions with a predetermined subset of fixed points.

The Three S-Boxes

S-Box	MIDORI	PRINCE		QARMA			
		Direct	Inverse	σ_0	σ_1	σ_2 Direct	σ_2 Inverse
Max. prob. of a differential	1/4	1/4	1/4	1/4	1/4	1/4	1/4
# with max. probability	24	15	15	18	15	15	15
Max. bias of a lin. approx.	1/4	1/4	1/4	1/4	1/4	1/4	1/4
# with max. bias	36	30	30	32	30	30	30
Algebraic Degree	3	3	3	3	3	3	3
# components of deg 3, 2	12, 3	15, 0	15, 0	14, 1	15, 0	15, 0	15, 0
Fixed Points	4	0	0	2	0	0	0
Minimal depth (GE)	3.5	5	4.5	3.5	4	4.5	4
Minimal area (GE)	12.8	20.2	19	14.17	16.5	20.2	19

The Three S-Boxes

S-Box	MIDORI	PRINCE Direct	PRINCE Inverse	QARMA σ_0	QARMA σ_1	QARMA σ_2 Direct	QARMA σ_2 Inverse
Max. prob. of a differential	1/4	1/4	1/4	1/4	1/4	1/4	1/4
# with max. probability	24	15	15	18	15	15	15
Max. bias of a lin. approx.	1/4	1/4	1/4	1/4	1/4	1/4	1/4
# with max. bias	36	30	30	32	30	30	30
Algebraic Degree	3	3	3	3	3	3	3
# components of deg 3, 2	12, 3	15, 0	15, 0	14, 1	15, 0	15, 0	15, 0
Fixed Points	4	0	0	2	0	0	0
Minimal depth (GE)	3.5	5	4.5	3.5	4	4.5	4
Minimal area (GE)	12.8	20.2	19	14.17	16.5	20.2	19

σ_0 is similar to MIDORI's S-Box but is has better cryptographic properties (all parameters that can be improved are improved), same latency, and slightly larger area

The Three S-Boxes

S-Box	MIDORI	PRINCE		QARMA			
		Direct	Inverse	σ_0	σ_1	σ_2 Direct	σ_2 Inverse
Max. prob. of a differential	1/4	1/4	1/4	1/4	1/4	1/4	1/4
# with max. probability	24	15	15	18	15	15	15
Max. bias of a lin. approx.	1/4	1/4	1/4	1/4	1/4	1/4	1/4
# with max. bias	36	30	30	32	30	30	30
Algebraic Degree	3	3	3	3	3	3	3
# components of deg 3, 2	12, 3	15, 0	15, 0	14, 1	15, 0	15, 0	15, 0
Fixed Points	4	0	0	2	0	0	0
Minimal depth (GE)	3.5	5	4.5	3.5	4	4.5	4
Minimal area (GE)	12.8	20.2	19	14.17	16.5	20.2	19

σ_1 is optimal *and* involutory, and has properties that may make side channel attacks more difficult

The Three S-Boxes

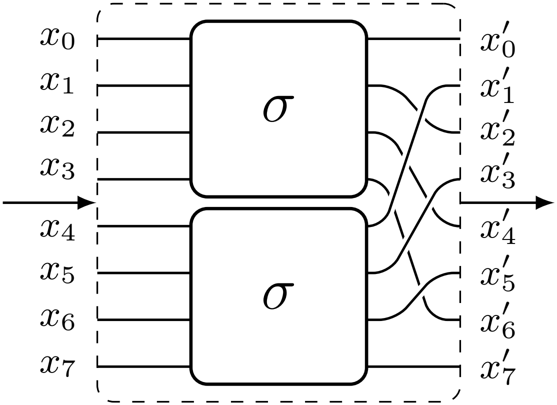
S-Box	MIDORI	PRINCE		QARMA			
		Direct	Inverse	σ_0	σ_1	σ_2 Direct	σ_2 Inverse
Max. prob. of a differential	1/4	1/4	1/4	1/4	1/4	1/4	1/4
# with max. probability	24	15	15	18	15	15	15
Max. bias of a lin. approx.	1/4	1/4	1/4	1/4	1/4	1/4	1/4
# with max. bias	36	30	30	32	30	30	30
Algebraic Degree	3	3	3	3	3	3	3
# components of deg 3, 2	12, 3	15, 0	15, 0	14, 1	15, 0	15, 0	15, 0
Fixed Points	4	0	0	2	0	0	0
Minimal depth (GE)	3.5	5	4.5	3.5	4	4.5	4
Minimal area (GE)	12.8	20.2	19	14.17	16.5	20.2	19

σ_2 comes from the PRINCE selection

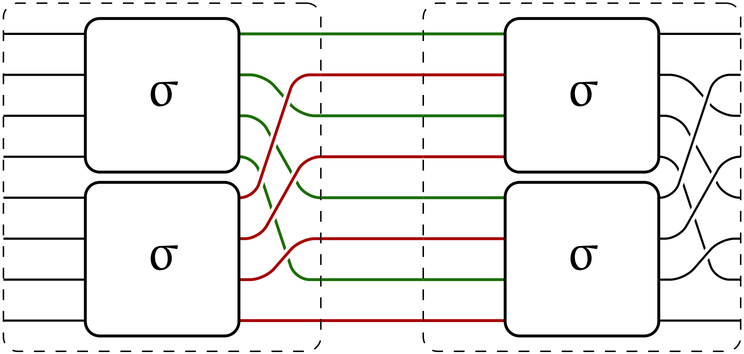
4. A 128-bit cipher

with a 256-bit key

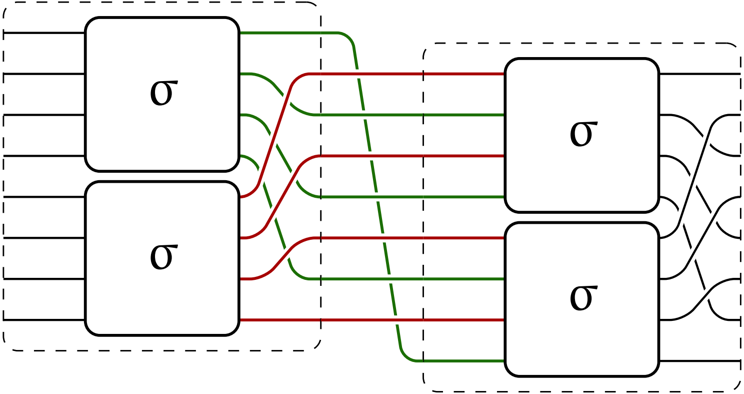
The 8-bit S-Box for QARMA-128



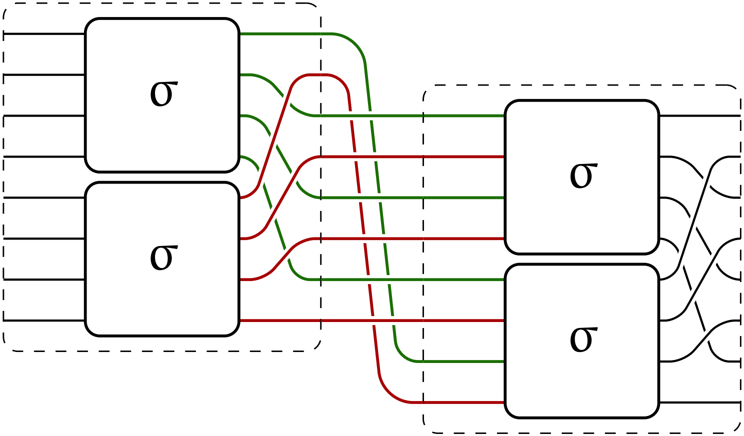
The 8-bit S-Box for QARMA-128



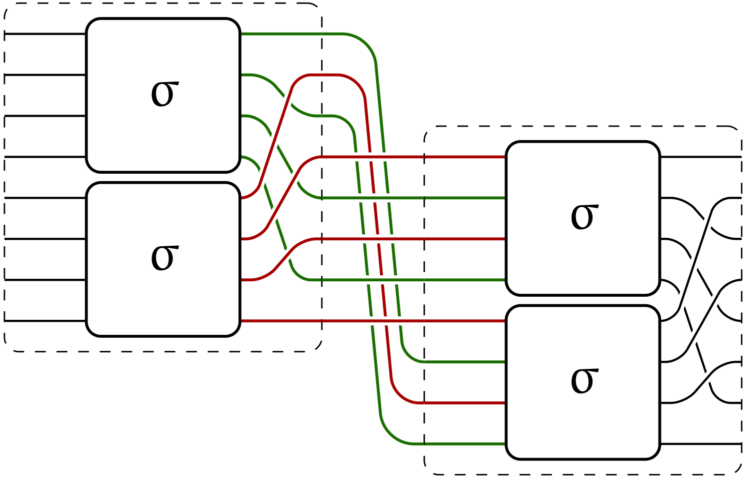
The 8-bit S-Box for QARMA-128



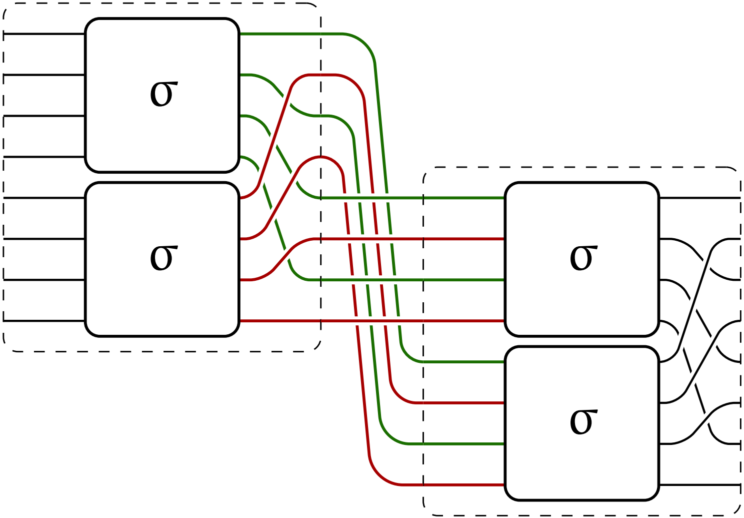
The 8-bit S-Box for QARMA-128



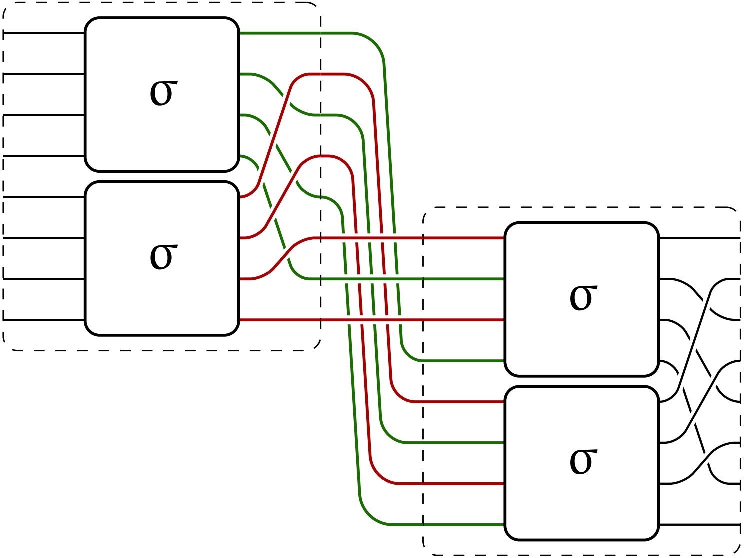
The 8-bit S-Box for QARMA-128



The 8-bit S-Box for QARMA-128



The 8-bit S-Box for QARMA-128



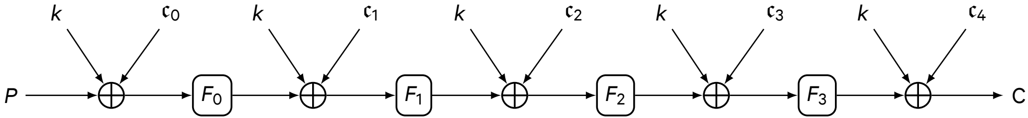
Security Analysis

Considered attacks (designing block ciphers is horrible, horrible)

- Linear and differential cryptanalysis (MILP models, following Beierle)
- —, under related tweak model (MILP models, following Beierle)
- Reflection Attacks (follows from structure)
- Generic attacks on Even-Mansour schemes (follows from structure)
- Slide attacks (follows from round heterogeneity)
- Meet-in-the-middle attacks (following MIDORI)
- Invariant subspace attacks (new heuristic arguments)
- Algebraic cryptanalysis (count equations and variables)
- Impossible differential & zero correlation linear cryptanalysis (method: Sun et al. EC ’16)
- Higher order differential cryptanalysis (boomerang, integral) (following MIDORI)

Invariant Subspace Attacks

These are subtle attacks and focus of very recent research.



Suppose there is a vector space $\mathcal{V}$, s.t. $F_i(b + \mathcal{V}) = a + \mathcal{V}$ for all i.

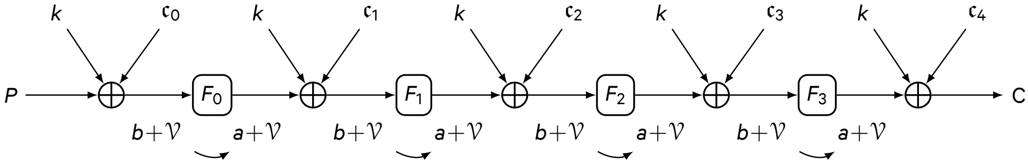
Note: $\mathcal{V}$ contains all $c_i + c_j$ ...

Distinguisher: if $P \in a + \mathcal{V}$ and $C \in b + \mathcal{V}$, then $k \in a + b + c_i + \mathcal{V}$ (likely).

We want $\mathcal{V}$ very small or very large ($\supseteq$ almost whole space).

Invariant Subspace Attacks

These are subtle attacks and focus of very recent research.



Suppose there is a vector space $\mathcal{V}$, s.t. $F_i(b+\mathcal{V}) = a+\mathcal{V}$ for all i.

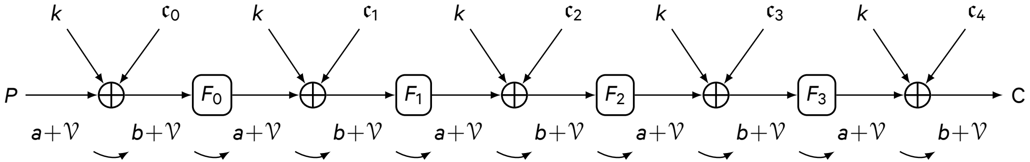
Note: $\mathcal{V}$ contains all $c_i + c_j$ ...

Distinguisher: if $P \in a+\mathcal{V}$ and $C \in b+\mathcal{V}$, then $k \in a+b+c_i+\mathcal{V}$ (likely).

We want $\mathcal{V}$ very small or very large ($\supseteq$ almost whole space).

Invariant Subspace Attacks

These are subtle attacks and focus of very recent research.



Suppose there is a vector space $\mathcal{V}$, s.t. $F_i(b+\mathcal{V}) = a+\mathcal{V}$ for all i.

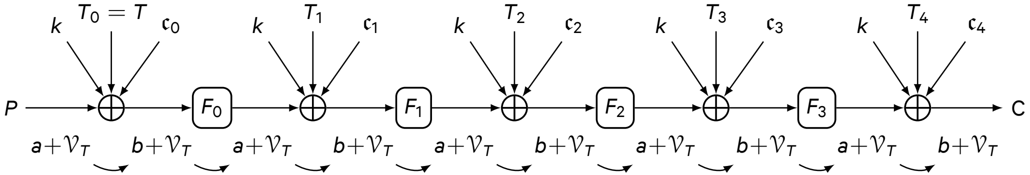
Note: $\mathcal{V}$ contains all $c_i + c_j$ ...

Distinguisher: if $P \in a+\mathcal{V}$ and $C \in b+\mathcal{V}$, then $k \in a+b+c_i+\mathcal{V}$ (likely).

We want $\mathcal{V}$ very small or very large ($\supseteq$ almost whole space).

Invariant Subspace Attacks

These are subtle attacks and focus of very recent research.



Suppose there is a vector space $\mathcal{V}$, s.t. $F_i(b+\mathcal{V}) = a+\mathcal{V}$ for all i.

Note: $\mathcal{V}_T$ contains all $(c_i + T_i) + (c_j + T_j)$ ...

Distinguisher: if $P \in a + \mathcal{V}$ and $C \in b + \mathcal{V}$, then $k \in a + b + c_i + T_i + \mathcal{V}_T$ (likely).

We want $\mathcal{V}_T$ very small or very large ($\supseteq$ almost whole space).

Invariant Subspaces - The importance of structure and diffusion matrices

Remark: in our case, any invariant subspace is invariant under τ, M and S.

Construct a $\mathcal{U} \subseteq \mathcal{V}$ by taking all $(c_i + T_i) + (c_j + T_j)$ and α, repeatedly applying τ and M.

Compute dimension of $\mathcal{U}$ for millions of random tweaks. Averages:

r	5	7
QARMA-64	60.32	63.02
MANTIS	46.92	55.37

r	8	11
QARMA-128	123.61	126.51
— with MIDORI matrix	92.17	107.17

These values vary with M. Their maximisation is part of the choice of M.

If we also take the S-box into account we always get the full space or codimension 1 (rare).

Implementation

Implementation (7nm FinFet)

Targeting	*Minimum Area*		*Minimum Delay*	
	Delay	Area	Delay	Area
Cipher	*ns*	GE	*ns*	GE
$\texttt{QARMA}_7$-64-σ_1	6.23	18362	3.25	34354
$\texttt{MANTIS}_7$	5.85	15831	2.94	27998
PRINCE	4.07	8702	2.12	20464
Mult. in $\mathbb{F}_{2^{64}}$	1.05	13083	0.44	16897

Implementation (7nm FinFet)

Targeting	*Minimum Area*		*Minimum Delay*	
	Delay	Area	Delay	Area
Cipher	*ns*	GE	*ns*	GE
$\mathtt{QARMA}_{11}$-128-σ_1	8.88	53872	4.80	96883
AES-128, pipelined*	15.67	71164	—	—
AES-256, pipelined*	21.99	101128	—	—
Mult. in $\mathbb{F}_{2^{128}}$	—	—	$\approx$ 0.5	$\approx$ 60K

* Note: The latency of one full AES round is 1.58 ns

Compare 2$\times$ AES plus one GFMULT to 1$\times$ QARMA-128

QARMA is placed in the public domain!

Standard for ARMv8.3-A pointer authentication: QARMA-64.

`https://community.arm.com/groups/processors/blog/2016/10/27/armv8-a-architecture-2016-additions`

`https://www.qualcomm.com/news/onq/2017/01/10/qualcomm-releases-whitepaper-detailing-pointer-authentication-armv83`

Ideal for memory encryption.

Analysis welcome! We can fix it if needed.

(For instance, Xiaoyang Dong et al: MITM on 10 rounds.)