Cube-like attack against nonce-misused Ascon

Journées Codage & Cryptographie (JC2) 2022

Jules Baudrin
joint work with Anne Canteaut & Léo Perrin (Inria, COSMIQ)

April 2022

GT CODAGE & CRYPTOGRAPHIE

Contact: jules.baudrin@inria.fr

A few words about the context

Lightweight symmetric cryptography

- Internet of Things: new usages, **new security needs**
- **Lightweightness**: "*Best*" trade-off between size, speed and security according to future usages
- Many different usages = many different constraints (hardware, software, which measure units...)

International standardization

- **CAESAR** competition (2013 – 2019)
- Current **NIST standardization process** (2018 –)

▶ Our target: **Ascon**, one of the CAESAR winners, one of the finalists in the NIST LWC process

Ascon [DEMS19]

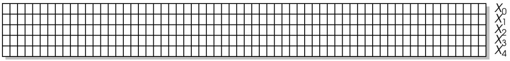
- **Authenticated encryption**: confidentiality/authenticity/integrity all-in-one in a single primitive

- Two main steps in the design:
    - A choice of a **mode of operation**: abstract construction with generic functions
    - A choice of an **instantiation** of the mode with carefully-chosen primitives

- In the case of Ascon:
    - Duplex Sponge mode
    - **bijection** $p\colon \mathbb{F}_2^{320} \to \mathbb{F}_2^{320}$: main object studied here

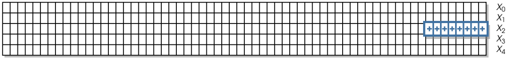
A confusion/diffusion structure…

The state

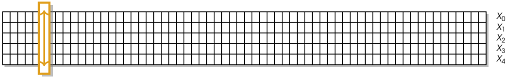


$$p = p_L \circ p_S \circ p_C$$

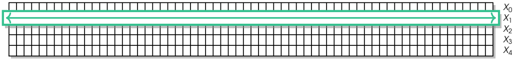
The constant addition p_C



The substitution layer p_S



The linear layer p_L



…studied algebraically

$$\begin{aligned}
y_0 &= \mathbf{x_4 x_1} + x_3 + \mathbf{x_2 x_1} + x_2 + \mathbf{x_1 x_0} + x_1 + x_0 \\
y_1 &= x_4 + \mathbf{x_3 x_2} + \mathbf{x_3 x_1} + x_3 + \mathbf{x_2 x_1} + x_2 + x_1 + x_0 \\
y_2 &= \mathbf{x_4 x_3} + x_4 + x_2 + x_1 + 1 \\
y_3 &= \mathbf{x_4 x_0} + x_4 + \mathbf{x_3 x_0} + x_3 + x_2 + x_1 + x_0 \\
y_4 &= \mathbf{x_4 x_1} + x_4 + x_3 + \mathbf{x_1 x_0} + x_1
\end{aligned}$$

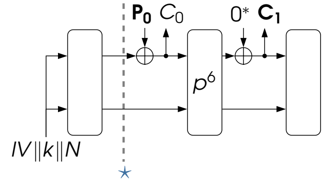
Algebraic Normal Form (ANF) of the S-box

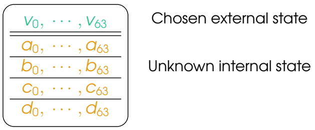
$$\begin{aligned}
X_0 &= X_0 \oplus (X_0 \ggg 19) \oplus (X_0 \ggg 28) \\
X_1 &= X_1 \oplus (X_1 \ggg 61) \oplus (X_1 \ggg 39) \\
X_2 &= X_2 \oplus (X_2 \ggg 1) \oplus (X_2 \ggg 6) \\
X_3 &= X_3 \oplus (X_3 \ggg 10) \oplus (X_3 \ggg 17) \\
X_4 &= X_4 \oplus (X_4 \ggg 7) \oplus (X_4 \ggg 41)
\end{aligned}$$

ANF of the linear layer p_L

The nonce-misuse scenario

Simplified setting





*After initialization

- Many reuse of the **same** (k, N) **pair**
- Chosen-plaintexts attack
- **If** the whole state is recovered, confidentiality is compromised

Cube attack principle

f_j denotes the jth output coordinate. Instead of $f_j \in \mathbb{F}_2[v_0, \cdots, v_{63}, a_0, \cdots, d_{63}]$, we separate public variables from secret variables:

$$f_j \in \mathbb{F}_2[a_0, \cdots, d_{63}][v_0, \cdots, v_{63}] \quad f_j = \sum_{(u_0, \cdots, u_{63}) \in \mathbb{F}_2^{64}} \alpha_{u,\,j} \left(\prod_{i=0}^{63} {v_i}^{u_i} \right)$$

where $\alpha_{u,\,j} \in \mathbb{F}_2[a_0, \cdots, d_{63}]$.

Cube attack principle

f_j denotes the jth output coordinate. Instead of $f_j \in \mathbb{F}_2[v_0, \cdots, v_{63}, a_0, \cdots, d_{63}]$, we separate public variables from secret variables:

$f_j \in \mathbb{F}_2[a_0, \cdots, d_{63}][v_0, \cdots, v_{63}]$ $\quad f_j = \sum_{(u_0, \cdots, u_{63}) \in \mathbb{F}_2^{64}} \alpha_{u,\, j} \left(\prod_{i=0}^{63} v_i^{u_i} \right)$

where $\alpha_{u,\, j} \in \mathbb{F}_2[a_0, \cdots, d_{63}]$.

Polynomial **expression** of $\alpha_{u,\, j}$ + **value** of $\alpha_{u,\, j}$ =
equation in the unknown variables $\simeq$
recovery of some information (if easily-solvable)

0. Select a monomial (**cube**) in f_j and target its coefficient: $\alpha_{u,\, j}$
1. **Offline phase**: recovery of the algebraic expression of $\alpha_{u,\, j}$
2. **Online phase**: recovery of the value of $\alpha_{u,\, j}$:
   $\alpha_{u,\, j} = \sum_{v \preccurlyeq u} f(v)$ (**chosen queries**).

Recovery of the polynomial: main problems

Problem 0: impossible access to the full ANF

$p \circ \cdots \circ p$: 6 iterations, 256 unknown variables.
S-box layer squares the number of terms. Linear layer triples it.
Impossible.

Recovery of the polynomial: main problems

Problem 0: impossible access to the full ANF

$p \circ \cdots \circ p$: 6 iterations, 256 unknown variables.
S-box layer squares the number of terms. Linear layer triples it.
Impossible.

Pb. 1: impossible access to a given $\alpha_{u,\,j}$ expression

Finding $\alpha_{u,\,j}$ for fixed u and j. **Too many combinatorial possibilities to track!**

Recovery of the polynomial: main problems

Problem 0: impossible access to the full ANF

$p \circ \cdots \circ p$: 6 iterations, 256 unknown variables.
S-box layer squares the number of terms. Linear layer triples it.
Impossible.

Pb. 1: impossible access to a given $\alpha_{u,\,j}$ expression

Finding $\alpha_{u,\,j}$ for fixed u and j. **Too many combinatorial possibilities to track!**

$v_0 v_1 = v_0 \times v_1 = (v_0 v_1) \times 1 = (v_0 v_1) \times v_0 = (v_0 v_1) \times v_1 = (v_0 v_1) \times (v_0 v_1)$

Recovery of the polynomial: main problems

Problem 0: impossible access to the full ANF

$p \circ \cdots \circ p$: 6 iterations, 256 unknown variables.
S-box layer squares the number of terms. Linear layer triples it.
Impossible.

Pb. 1: impossible access to a given $\alpha_{u,j}$ expression

Finding $\alpha_{u,j}$ for fixed u and j. **Too many combinatorial possibilities to track!**

$$v_0 v_1 = v_0 \times v_1 = (v_0 v_1) \times 1 = (v_0 v_1) \times v_0 = (v_0 v_1) \times v_1 = (v_0 v_1) \times (v_0 v_1)$$

Pb. 2: Finding exploitable $\alpha_{u,j}$

We need to be able to solve the system!

Recovery of the polynomial: main problems

Problem 0: impossible access to the full ANF

$p \circ \cdots \circ p$: 6 iterations, 256 unknown variables.
S-box layer squares the number of terms. Linear layer triples it.
Impossible.

Pb. 1: impossible access to a given $\alpha_{u,\,j}$ expression

Finding $\alpha_{u,\,j}$ for fixed u and j. **Too many combinatorial possibilities to track!**

$$v_0 v_1 = v_0 \times v_1 = (v_0 v_1) \times 1 = (v_0 v_1) \times v_0 = (v_0 v_1) \times v_1 = (v_0 v_1) \times (v_0 v_1)$$

Pb. 2: Finding exploitable $\alpha_{u,\,j}$

We need to be able to solve the system!

- Highest-degree terms (2^{t-1} at round t) are easier to study.

Strong constraint: products of two former highest-degree terms.

$$v_0 v_1 = v_0 \times v_1 = \cancel{(v_0 v_1) \times 1} = \cancel{(v_0 v_1) \times v_0} = \cancel{(v_0 v_1) \times v_1} = \cancel{(v_0 v_1) \times (v_0 v_1)}$$

Highest-degree terms in theory

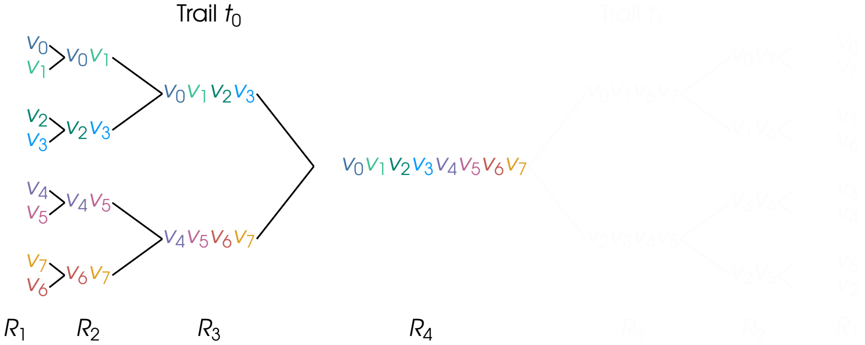
Strong constraint: products of two former highest-degree terms.

$$v_0 v_1 = v_0 \times v_1 = \cancel{(v_0 v_1) \times 1} = \cancel{(v_0 v_1) \times v_0} = \cancel{(v_0 v_1) \times v_1} = \cancel{(v_0 v_1) \times (v_0 v_1)}$$

Highest-degree terms in theory

Strong constraint: products of two former highest-degree terms.

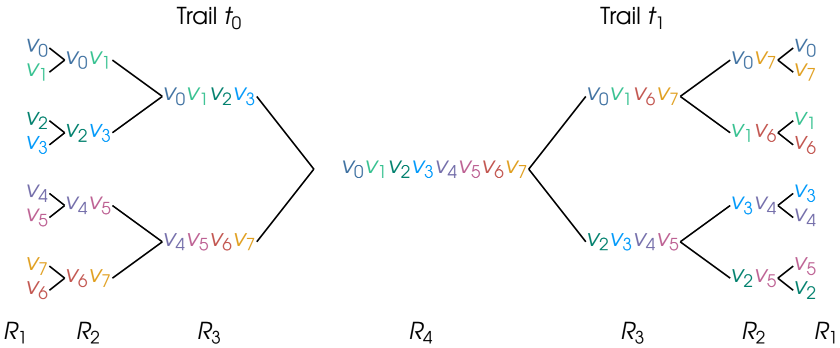
$$v_0v_1 = v_0 \times v_1 = \cancel{(v_0v_1) \times 1} = \cancel{(v_0v_1) \times v_0} = \cancel{(v_0v_1) \times v_1} = \cancel{(v_0v_1) \times (v_0v_1)}$$



Highest-degree terms in theory

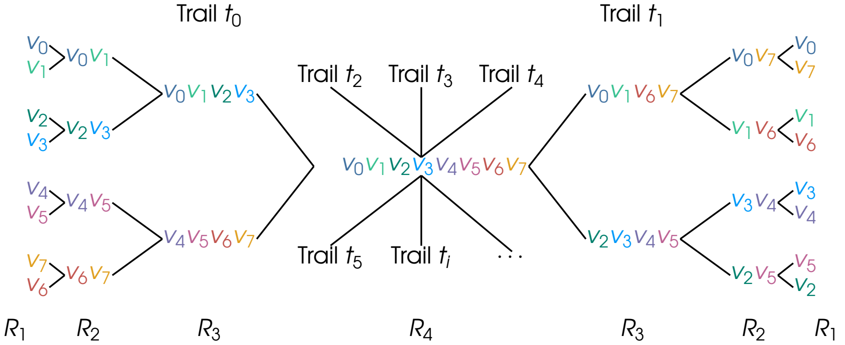
Strong constraint: products of two former highest-degree terms.

$$v_0v_1 = v_0 \times v_1 = \cancel{(v_0v_1) \times 1} = \cancel{(v_0v_1) \times v_0} = \cancel{(v_0v_1) \times v_1} = \cancel{(v_0v_1) \times (v_0v_1)}$$



Strong constraint: products of two former highest-degree terms.

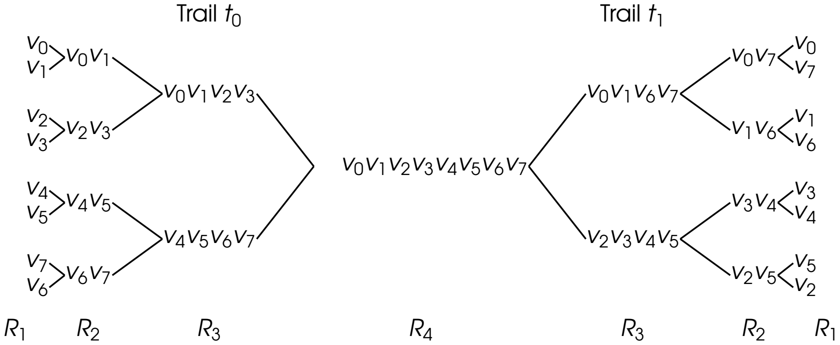
$$v_0v_1 = v_0 \times v_1 = \cancel{(v_0v_1) \times 1} = \cancel{(v_0v_1) \times v_0} = \cancel{(v_0v_1) \times v_1} = \cancel{(v_0v_1) \times (v_0v_1)}$$



Highest-degree terms in theory

Strong constraint: products of two former highest-degree terms.

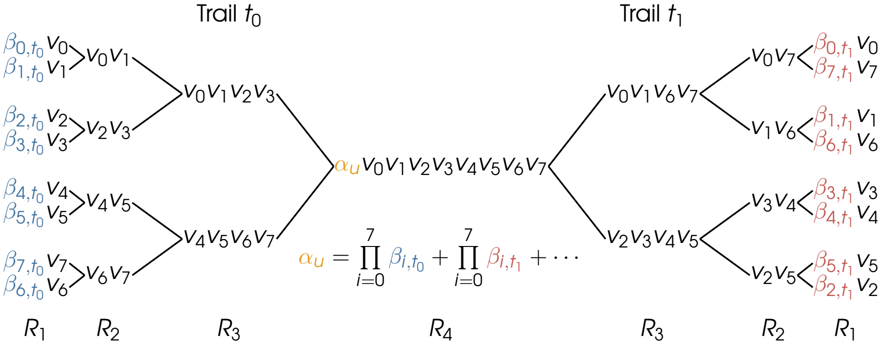
$$v_0v_1 = v_0 \times v_1 = \cancel{(v_0v_1) \times 1} = \cancel{(v_0v_1) \times v_0} = \cancel{(v_0v_1) \times v_1} = \cancel{(v_0v_1) \times (v_0v_1)}$$



Highest-degree terms in theory

Strong constraint: products of two former highest-degree terms.

$$v_0v_1 = v_0 \times v_1 = \cancel{(v_0v_1) \times 1} = \cancel{(v_0v_1) \times v_0} = \cancel{(v_0v_1) \times v_1} = \cancel{(v_0v_1) \times (v_0v_1)}$$



- Fewer combinatorial choices
- Known structure of α_u: sum of products of former coefficients

Highest-degree terms in **practice**

For $r = 6$

- Still costly to recover the polynomial expressions: computations have to be done round after round.

- The polynomials look horrible!

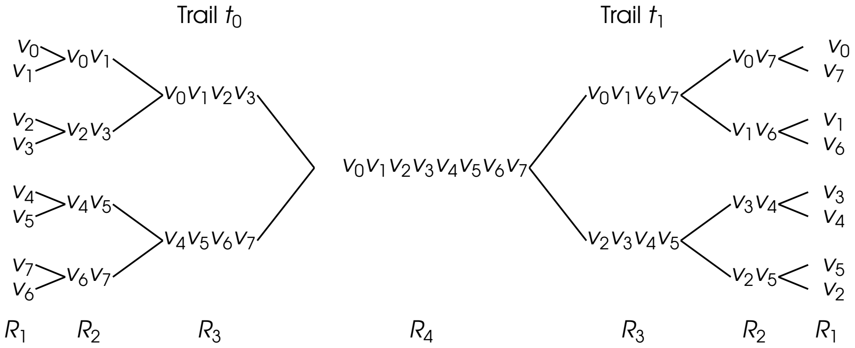
- ► Need for a cheaper and easier recovery: **conditional cubes** [HWX$^+$17, LDW17]

Conditional cubes

- We look for α_u with a simple divisor: β_0.
- Even **without the full knowledge** of α_u we know that:
  $\alpha_u = 1 \implies \beta_0 = 1$.
- If β_0 is linear, the **system** will be **linear**.

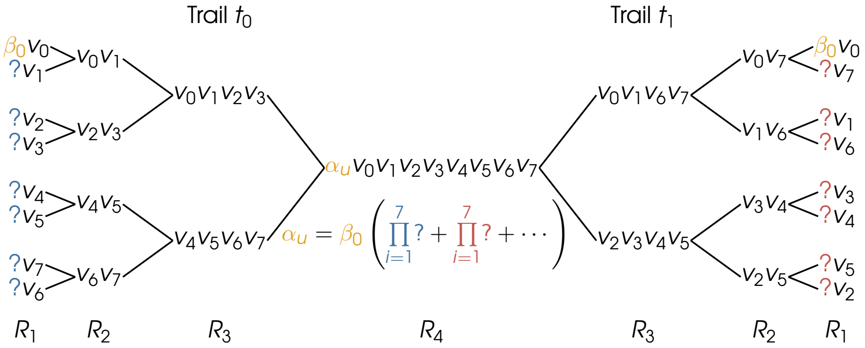
Conditional cubes

- We look for α_u with a simple divisor: β_0.
- Even **without the full knowledge** of α_u we know that: $\alpha_u = 1 \implies \beta_0 = 1$.
- If β_0 is linear, the **system** will be **linear**.



Conditional cubes

- We look for α_u with a simple divisor: β_0.
- Even **without the full knowledge** of α_u we know that: $\alpha_u = 1 \implies \beta_0 = 1$.
- If β_0 is linear, the **system** will be **linear**.



Choice of the cube: forcing some linear divisors

Study of the first rounds: Column C_0 after the first S-box layer

$$
\begin{array}{|c|}
\hline
v_0 \\
\hline
a_0 \\
b_0 \\
c_0 \\
d_0 \\
\hline
\end{array}
\xrightarrow{S_1}
\begin{array}{|c|}
\hline
(a_0+1)v_0 + \cdots \\
\hline
v_0 + \cdots \\
\hline
\cdots \\
\hline
(c_0+d_0+1)v_0 + \cdots \\
\hline
a_0 v_0 + \cdots \\
\hline
\end{array}
\quad
\begin{array}{l}
\longleftarrow \beta_0 := a_0 + 1 \\
\\
\\
\longleftarrow \gamma_0 := c_0 + d_0 + 1 \\
\\
\end{array}
$$

$\alpha_{u,\,j} = \beta_0(\dots) + \gamma_0(\dots)$ for all output coordinate indices $j \in [\![0, 63]\!]$.

Choice of the cube: forcing some linear divisors

Study of the first rounds: Column C_0 after the first S-box layer

$$\begin{array}{|c|}\hline v_0 \\ \hline a_0 \\ b_0 \\ c_0 \\ d_0 \\ \hline\end{array} \xrightarrow[S_1]{} \begin{array}{|c|}\hline (a_0+1)v_0 + \cdots \\ \hline v_0 + \cdots \\ \hline \cdots \\ \hline (c_0+d_0+1)v_0 + \cdots \\ \hline a_0 v_0 + \cdots \\ \hline\end{array} \quad \begin{array}{l} \longleftarrow \beta_0 := a_0 + 1 \\ \\ \\ \longleftarrow \gamma_0 := c_0 + d_0 + 1 \\ \\ \end{array}$$

$\alpha_{u,\,j} = \beta_0(\dots) + \gamma_0(\dots)$ for all output coordinate indices $j \in [\![0, 63]\!]$.

- $(\alpha_{u,0}, \cdots, \alpha_{u,63}) \neq (0, \cdots, 0) \implies \beta_0 = 1$ or $\gamma_0 = 1$
- In practice, reciprocal also true! $\forall\, j,\ \alpha_{u,\,j} = 0 \implies \beta_0 = 0$ and $\gamma_0 = 0$

Choice of the cube: forcing some linear divisors

Study of the first rounds: Column C_0 after the first S-box layer

$$
\begin{array}{|c|}
\hline
v_0 \\
\hline
a_0 \\
b_0 \\
c_0 \\
d_0 \\
\hline
\end{array}
\xrightarrow{S_1}
\begin{array}{|c|}
\hline
(a_0+1)v_0+\cdots \\
\hline
v_0+\cdots \\
\hline
\cdots \\
\hline
(c_0+d_0+1)v_0+\cdots \\
\hline
a_0 v_0+\cdots \\
\hline
\end{array}
\quad
\begin{array}{l}
\longleftarrow \beta_0 := a_0+1 \\
\\
\\
\longleftarrow \gamma_0 := c_0+d_0+1 \\
\\
\end{array}
$$

$\alpha_{u,\,j} = \beta_0(\dots) + \gamma_0(\dots)$ for all output coordinate indices $j \in [\![0, 63]\!]$.

- $(\alpha_{u,0}, \cdots, \alpha_{u,63}) \neq (0, \cdots, 0) \implies \beta_0 = 1$ or $\gamma_0 = 1$
- In practice, reciprocal also true! $\forall\, j,\ \alpha_{u,\,j} = 0 \implies \beta_0 = 0$ and $\gamma_0 = 0$

An effective attack in 3 steps

1. Conditional cube attack: recovery of all $c_i + d_i + 1$ and some a_i
2. Cube attack: recovery of remaining a_i (adaptive step)
3. Cube attack: recovery of all b_i and c_i (target **sub-leading** terms)

Conclusion

- Looking at diffusion through the ANF.
- Effective full-state recovery on the full 6-round encryption: 2^{40} in time and data.
- Does not threaten Ascon (and Isap) directly.
- Good reminder that **a nonce is not a constant**!
- Importance of studying misused ciphers.

Main questions

- Can theoretical arguments underpin the "in practice it works" parts of the study?
- Are 6 rounds enough for encryption? (No cube attacks seem feasible on 7 rounds)

Conclusion

- Looking at diffusion through the ANF.
- Effective full-state recovery on the full 6-round encryption: 2^{40} in time and data.
- Does not threaten Ascon (and Isap) directly.
- Good reminder that **a nonce is not a constant**!
- Importance of studying misused ciphers.

Main questions

- Can theoretical arguments underpin the “in practice it works” parts of the study?
- Are 6 rounds enough for encryption? (No cube attacks seem feasible on 7 rounds)

Thank you for your attention!

Bibliography

Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer.
Ascon v1.2.
Technical report, National Institute of Standards and Technology, 2019.
https://csrc.nist.gov/Projects/lightweight-cryptography/finalists.

Senyang Huang, Xiaoyun Wang, Guangwu Xu, Meiqin Wang, and Jingyuan Zhao.
Conditional cube attack on reduced-round Keccak sponge function.
In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 259–288, Paris, France, April 30 – May 4, 2017. Springer, Heidelberg, Germany.

Zheng Li, Xiaoyang Dong, and Xiaoyun Wang.
Conditional cube attack on round-reduced ASCON.
IACR Trans. Symm. Cryptol., 2017(1):175–202, 2017.