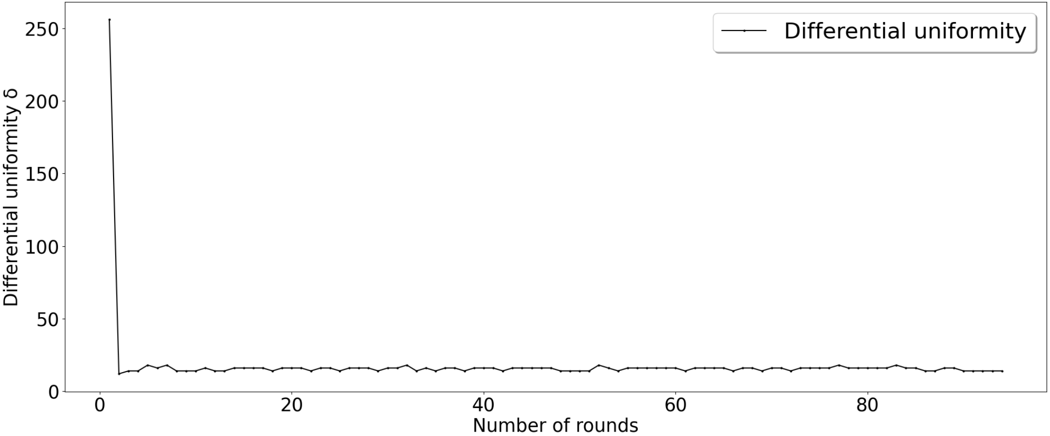
Propagation of Subspaces in Primitives with Monomial Sboxes: Applications to Rescue and Variants of the AES

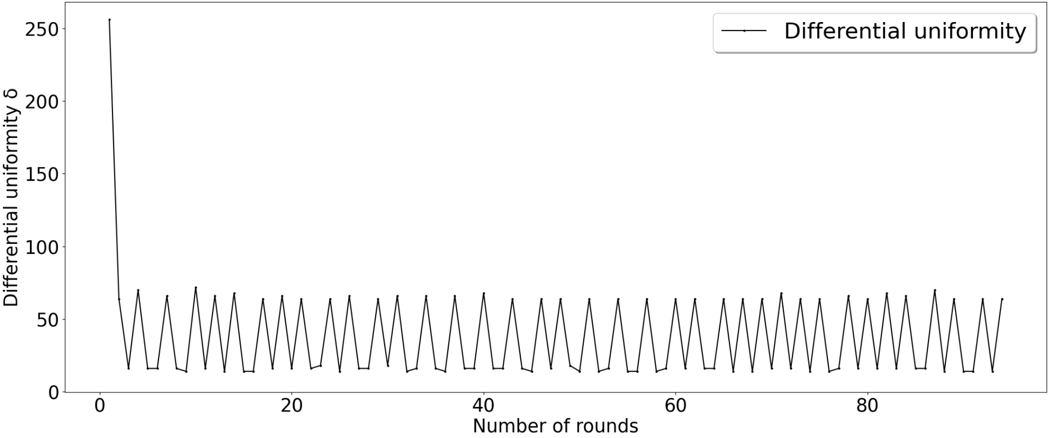
Aurélien Boeuf[1], Anne Canteaut[1], Léo Perrin[1]

[1]Inria Paris

FSE 2024, Leuven, Belgium



Wide-trail strategy!



Wide-trail strategy...?

Arithmetization-Oriented Symmetric Primitives

Arithmetization-Oriented Symmetric Primitives

- Advanced protocols (Zero-Knowledge proofs, MPC, FHE...) need primitives with a "simple" arithmetic description (e.g. using $x \mapsto x^3$ as the main nonlinear function), sometimes over $\mathbb{F}_q$ for a specific large q.

Arithmetization-Oriented Symmetric Primitives

- Advanced protocols (Zero-Knowledge proofs, MPC, FHE...) need primitives with a “simple” arithmetic description (e.g. using $x \mapsto x^3$ as the main nonlinear function), sometimes over $\mathbb{F}_q$ for a specific large q.

Classic	**Arithmetization-Oriented**
Binary operations	Arithmetic operations
Algebraically complex (for cheap)	Algebraically simple
Small field ($\mathbb{F}_{2^8}$)	Large (sometimes prime) field ($\mathbb{F}_q$)
e.g. AES, SHA-3	e.g. MiMC, Rescue

Arithmetization for Zero-Knowledge

- Implemented using "constraint systems" (R1CS, AIR, Plonk...).
- **Less constraints = Better performance**.

Function → Arithmetic circuit → Set of constraints

Arithmetization for Zero-Knowledge

- Implemented using “constraint systems” (R1CS, AIR, Plonk...).
- **Less constraints = Better performance**.

Function → Arithmetic circuit → Set of constraints

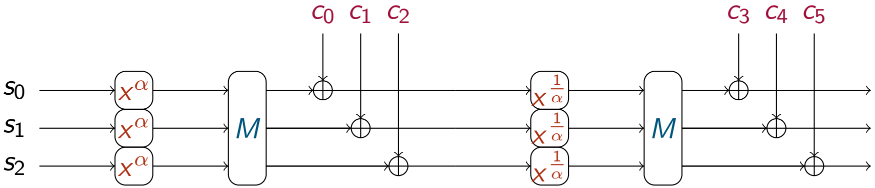
	$\rightarrow \boxed{x^3} \rightarrow$	$\rightarrow$ **AES** S-box $\rightarrow$	$\rightarrow \boxed{x^{1/3}} \rightarrow$
Security	**Low** degree	**High** degree	**High** degree
Performance	**Few** constraints	**Many** constraints	**Few** constraints

(Because low degree inverse:

$y = x^{\alpha}$ vs $y^{\alpha} = x$)

Rescue-Prime

- Defined in $\mathbb{F}_p$ with p prime $> 2^{64}$. Here we focus on $m = 3$, $c = 1$ and $p \approx 2^{256}$.



Two steps of Rescue for $m = 3$ (repeated $N \geq 8$ times).

- Defined for any MDS matrix M and round constants c_i.

Rescue's Design Choices

- Alternate x^{α} and $x^{\frac{1}{\alpha}}$ for resistance against algebraic attacks.
- Low verification cost, high degree overall.
- x^{α} has good cryptographic properties (APN for $\alpha = 3$).
- The standard wide-trail strategy is used.

Main motivation: Are the usual security arguments sufficient?

Differential Uniformity

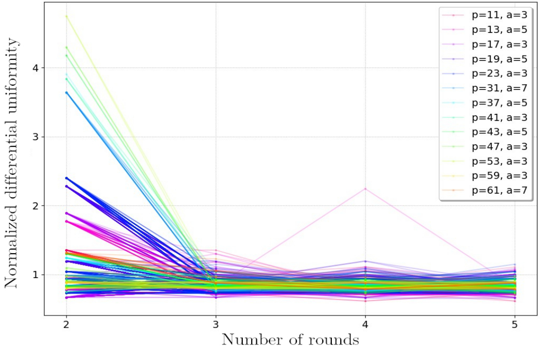
Definition

Differential uniformity of a function F:

$$\delta(F) = \max_{\sigma \neq 0, \beta} \left|\{F(x+\sigma) - F(x) = \beta \ \text{ s.t. } x \in (\mathbb{F}_p)^m\}\right|$$

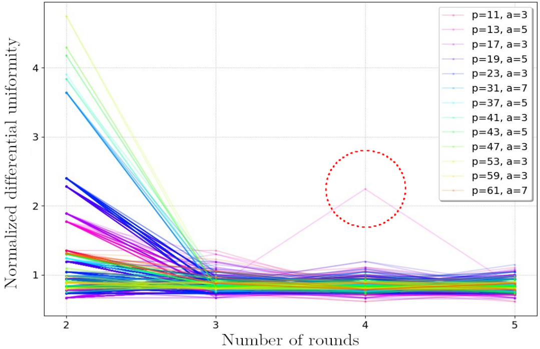
→ This quantity must be minimized.

High Differential Uniformities in Rescue



Graph taken from `eprint.iacr.org/2020/820`.

High Differential Uniformities in Rescue



Graph taken from `eprint.iacr.org/2020/820`.

Overview of Rescue

Affine Space Chains

Weak Designs and Weird Designs

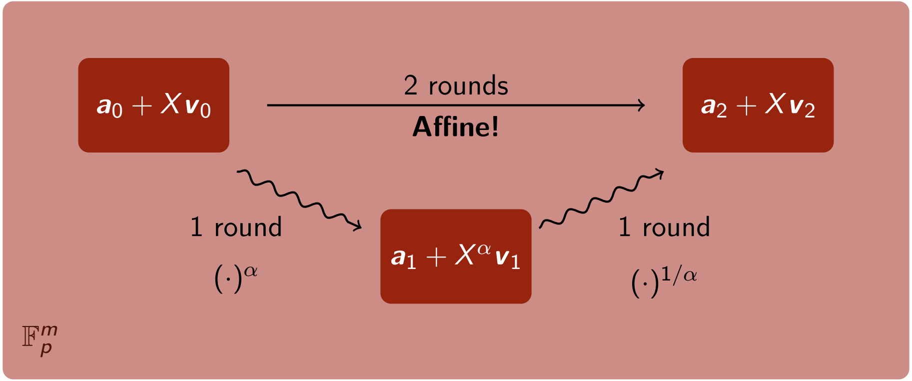
Conclusion

High Differential Uniformities in Rescue

- The cause? Affine spaces of dimension 1 mapped from one to another.
- Write elements of $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$ as $\boldsymbol{a} + X\boldsymbol{v}$ ($X \in \mathbb{F}_p$).

High Differential Uniformities in Rescue

- The cause? Affine spaces of dimension 1 mapped from one to another.
- Write elements of $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$ as $\boldsymbol{a} + X\boldsymbol{v}$ ($X \in \mathbb{F}_p$).



High Differential Uniformities in Rescue

$$\delta(F) = \max_{\sigma \neq 0, \beta} |\{F(x+\sigma) - F(x) = \beta \text{ s.t. } x \in (\mathbb{F}_p)^m\}|.$$

$$\forall X \in \mathbb{F}_p, F\begin{pmatrix} a \\ X \end{pmatrix} = \begin{pmatrix} eX + f \\ gX + h \end{pmatrix}.$$

High Differential Uniformities in Rescue

$$\delta(F) = \max_{\sigma \neq 0, \beta} |\{F(x+\sigma) - F(x) = \beta \ \text{ s.t. } \ x \in (\mathbb{F}_p)^m\}|.$$

$$\forall X \in \mathbb{F}_p, F\begin{pmatrix} a \\ X \end{pmatrix} = \begin{pmatrix} eX + f \\ gX + h \end{pmatrix}.$$

$$F\begin{pmatrix} a \\ X+1 \end{pmatrix} - F\begin{pmatrix} a \\ X \end{pmatrix} = \begin{pmatrix} e(X+1)+f \\ g(X+1)+h \end{pmatrix} - \begin{pmatrix} eX+f \\ gX+h \end{pmatrix}$$
$$= \begin{pmatrix} e \\ g \end{pmatrix} = \beta$$

$$\implies \quad \boxed{\delta(F) \geq p}$$

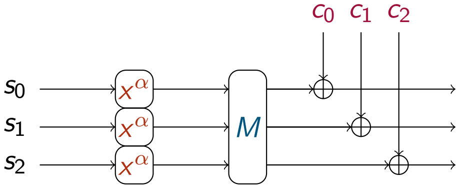

High Differential Uniformities in Rescue
Affine Space Chains

Affine Space Chains

Note $\boldsymbol{a} + \left\langle \boldsymbol{v} \right\rangle := \{\boldsymbol{a} + X\boldsymbol{v} \text{ such that } X \in \mathbb{F}_p\}$.

$$\boldsymbol{a}_0 + \langle \boldsymbol{v}_0 \rangle \xrightarrow{f} \boldsymbol{a}_1 + \langle \boldsymbol{v}_1 \rangle \xrightarrow{f} \dots \xrightarrow{f} \boldsymbol{a}_N + \langle \boldsymbol{v}_N \rangle$$

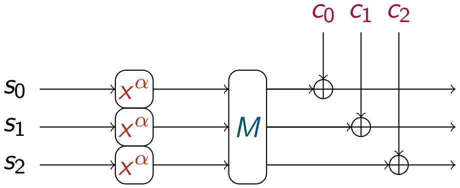
Main Idea



Rescue round.

Write elements of $\begin{pmatrix} 0 \\ 0 \\ a \end{pmatrix} + \left\langle \begin{pmatrix} 1 \\ v \\ 0 \end{pmatrix} \right\rangle$ as $\begin{pmatrix} s_0 \\ s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} X \\ vX \\ a \end{pmatrix}$.

Main Idea



Rescue round.

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} X \\ vX \\ a \end{pmatrix} \longrightarrow \begin{pmatrix} X^\alpha \\ v^\alpha X^\alpha \\ a^\alpha \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ a^\alpha \end{pmatrix} + X^\alpha \begin{pmatrix} 1 \\ v^\alpha \\ 0 \end{pmatrix}$$

This is the most important part! It only relies on the fact that the Sbox is a monomial.

Separable Affine Spaces

Definition

An affine space of dimension 1 E is separable if there exists $\boldsymbol{a}$ and $\boldsymbol{v}$ such that:

$$E = \boldsymbol{a} + \langle \boldsymbol{v} \rangle \quad \text{and} \quad \forall\, 0 \leq i \leq m-1\,, \quad a_i \cdot v_i = 0\,.$$

Equivalently, $$E = \boldsymbol{a} + \langle \boldsymbol{v} \rangle \quad \text{and} \quad \operatorname{supp}(\boldsymbol{a}) \cap \operatorname{supp}(\boldsymbol{v}) = \varnothing.$$

Separable Affine Spaces - Examples

- $\begin{pmatrix}1\\0\end{pmatrix} + \left\langle \begin{pmatrix}0\\1\end{pmatrix} \right\rangle$ **is a separable affine space** (these $\boldsymbol{a}$ and $\boldsymbol{v}$ have disjoint supports).

Separable Affine Spaces - Examples

- $\begin{pmatrix}1\\0\end{pmatrix} + \left\langle \begin{pmatrix}0\\1\end{pmatrix} \right\rangle$ **is a separable affine space** (these $\boldsymbol{a}$ and $\boldsymbol{v}$ have disjoint supports).
- $\begin{pmatrix}0\\1\end{pmatrix} + \left\langle \begin{pmatrix}1\\1\end{pmatrix} \right\rangle$ **is not**: its representants are the $\begin{pmatrix}\lambda\\\lambda+1\end{pmatrix} + \left\langle \begin{pmatrix}\mu\\\mu\end{pmatrix} \right\rangle$ with $\mu \neq 0$.

$\implies$ We would need $\begin{pmatrix}\lambda\\\lambda+1\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$: not possible!

Separable Affine Spaces - Examples

- $\begin{pmatrix}1\\0\end{pmatrix} + \left\langle \begin{pmatrix}0\\1\end{pmatrix} \right\rangle$ **is a separable affine space** (these $\boldsymbol{a}$ and $\boldsymbol{v}$ have disjoint supports).
- $\begin{pmatrix}0\\1\end{pmatrix} + \left\langle \begin{pmatrix}1\\1\end{pmatrix} \right\rangle$ **is not**: its representants are the $\begin{pmatrix}\lambda\\\lambda+1\end{pmatrix} + \left\langle \begin{pmatrix}\mu\\\mu\end{pmatrix} \right\rangle$ with $\mu \neq 0$.

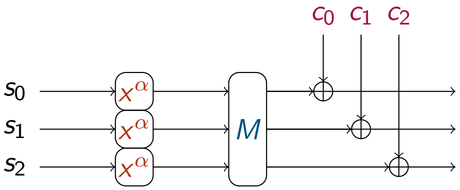
  $\Longrightarrow$ We would need $\begin{pmatrix}\lambda\\\lambda+1\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$: not possible!
- $\begin{pmatrix}1\\2\end{pmatrix} + \left\langle \begin{pmatrix}-1\\0\end{pmatrix} \right\rangle$

Separable Affine Spaces - Examples

- $\begin{pmatrix}1\\0\end{pmatrix} + \left\langle \begin{pmatrix}0\\1\end{pmatrix} \right\rangle$ **is a separable affine space** (these ***a*** and ***v*** have disjoint supports).
- $\begin{pmatrix}0\\1\end{pmatrix} + \left\langle \begin{pmatrix}1\\1\end{pmatrix} \right\rangle$ **is not**: its representants are the $\begin{pmatrix}\lambda\\\lambda+1\end{pmatrix} + \left\langle \begin{pmatrix}\mu\\\mu\end{pmatrix} \right\rangle$ with $\mu \neq 0$.

  $\Longrightarrow$ We would need $\begin{pmatrix}\lambda\\\lambda+1\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$: not possible!
- $\begin{pmatrix}1\\2\end{pmatrix} + \left\langle \begin{pmatrix}-1\\0\end{pmatrix} \right\rangle$ **is separable**: it is also represented by $\begin{pmatrix}0\\2\end{pmatrix} + \left\langle \begin{pmatrix}-1\\0\end{pmatrix} \right\rangle$.

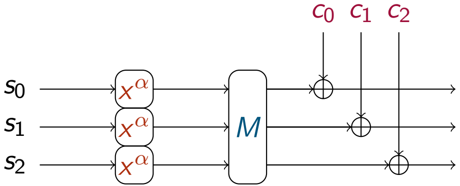
Main Idea



Rescue round.

$$\begin{pmatrix} 0 \\ 0 \\ a^\alpha \end{pmatrix} + X^\alpha \begin{pmatrix} 1 \\ v^\alpha \\ 0 \end{pmatrix} \longrightarrow M \begin{pmatrix} 0 \\ 0 \\ a^\alpha \end{pmatrix} + X^\alpha M \begin{pmatrix} 1 \\ v^\alpha \\ 0 \end{pmatrix}$$

Main Idea



Rescue round.

$$M\begin{pmatrix}0\\0\\a^\alpha\end{pmatrix} + X^\alpha M\begin{pmatrix}1\\v^\alpha\\0\end{pmatrix} \longrightarrow M\begin{pmatrix}0\\0\\a^\alpha\end{pmatrix} + \begin{pmatrix}c_0\\c_1\\c_2\end{pmatrix} + X^\alpha M\begin{pmatrix}1\\v^\alpha\\0\end{pmatrix}$$

Main Idea

$$M \begin{pmatrix} 0 \\ 0 \\ a^{\alpha} \end{pmatrix} + \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} + \left\langle M \begin{pmatrix} 1 \\ v^{\alpha} \\ 0 \end{pmatrix} \right\rangle$$

Main Idea

$$M\begin{pmatrix}0\\0\\a^{\alpha}\end{pmatrix}+\begin{pmatrix}c_1\\c_2\\c_3\end{pmatrix}+\left\langle M\begin{pmatrix}1\\v^{\alpha}\\0\end{pmatrix}\right\rangle$$

For this space to be separable, we need that there exists $\lambda \in \mathbb{F}_p$ such that

$$M\begin{pmatrix}1\\v^{\alpha}\\0\end{pmatrix} \text{ and } M\begin{pmatrix}0\\0\\a^{\alpha}\end{pmatrix}+\begin{pmatrix}c_1\\c_2\\c_3\end{pmatrix}+\lambda M\begin{pmatrix}1\\v^{\alpha}\\0\end{pmatrix}$$

have disjoint supports.

Main Result

Theorem

- *The image of a separable affine space* $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$ *by a round of a monomial SPN is an affine space.*

Main Result

Theorem

- *The image of a separable affine space* $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$ *by a round of a monomial SPN is an affine space.*
- *This image is still separable if and only if there exists* λ *in* $\mathbb{F}_p$ *such that:*

$$\forall i \in \mathrm{supp}(M \circ S)(\boldsymbol{v}), \quad c_i = \lambda (M \circ S)(\boldsymbol{v})_i - (M \circ S)(\boldsymbol{a})_i$$

Overview of Rescue

Affine Space Chains

Weak Designs and Weird Designs

Conclusion

Our Designs

To illustrate the limits of classical arguments in the AO context:

- Stir, a weak instance of Rescue.
- Snare, a tweakable cipher with a secret weak tweak. Directly based on the MALICIOUS framework (Peyrin & Wang, 2020).
- AES-like ciphers where we can introduce and control differential uniformity spikes.

Our Designs

To illustrate the limits of classical arguments in the AO context:

- Stir, a weak instance of Rescue.
- Snare, a tweakable cipher with a secret weak tweak. Directly based on the MALICIOUS framework (Peyrin & Wang, 2020).
- AES-like ciphers where we can introduce and control differential uniformity spikes.

Our Designs

To illustrate the limits of classical arguments in the AO context:

- Stir, a weak instance of Rescue.
- Snare, a tweakable cipher with a secret weak tweak. Directly based on the MALICIOUS framework (Peyrin & Wang, 2020).
- AES-like ciphers where we can introduce and control differential uniformity spikes.

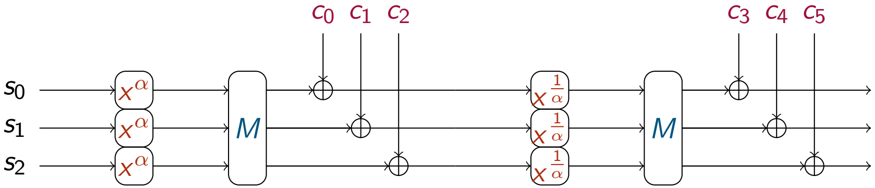
Our Designs

To illustrate the limits of classical arguments in the AO context:

- Stir, a weak instance of Rescue.
- Snare, a tweakable cipher with a secret weak tweak. Directly based on the MALICIOUS framework (Peyrin & Wang, 2020).
- AES-like ciphers where we can introduce and control differential uniformity spikes.

Stir

- Based on Rescue.
- MDS matrix M and round constants c are carefully chosen to impose one affine space chain over the whole permutation.



Stir

$$\begin{pmatrix}0\\0\\0\end{pmatrix} + \left\langle \begin{pmatrix}v_1\\v_2\\0\end{pmatrix} \right\rangle \longrightarrow \begin{pmatrix}0\\0\\a_3\end{pmatrix} + \left\langle \begin{pmatrix}v_1'\\v_2'\\0\end{pmatrix} \right\rangle \longrightarrow \ldots \longrightarrow \begin{pmatrix}0\\0\\0\end{pmatrix} + \left\langle \begin{pmatrix}v_1''\\v_2''\\0\end{pmatrix} \right\rangle$$

Yields $p \approx 2^{64}$ solutions to the CICO (Constrained Input Constrained Output) problem. This breaks security arguments in sponge constructions.

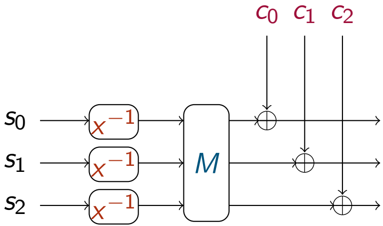
Affine Space Chain vs Affine Function

- Stir and Snare are based on affine space chains.
- Having an affine space chain doesn't mean that the function itself is affine.
- In the beginning we measured high differential uniformites because the function itself is affine on these subspaces.
- Can we recreate that?

$$\boldsymbol{a}_1 + X\boldsymbol{v}_1 \longrightarrow \boldsymbol{a}_2 + (X^\alpha + \lambda)\boldsymbol{v}_2 \longrightarrow \boldsymbol{a}_3 + (X^\alpha + \lambda)^{\frac{1}{\alpha}}\boldsymbol{v}_3$$

Morse Code with Differential Uniformity

- Same thing as Rescue, but with elements over $\mathbb{F}_{2^n}$ and the inverse function $x \mapsto x^{-1}$ as an Sbox.



Morse Code with Differential Uniformity

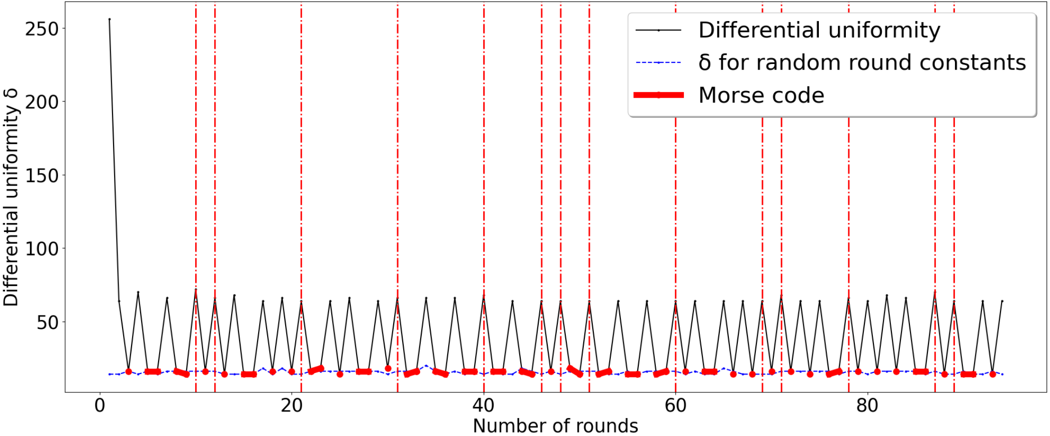
Idea: Same strategy as Stir, but make it so that the mapping from the input to output affine space is *itself* affine every 2 or 3 rounds!

Morse Code with Differential Uniformity

Idea: Same strategy as Stir, but make it so that the mapping from the input to output affine space is *itself* affine every 2 or 3 rounds!

- For a 2-round delay, the coefficient X of the affine space basis verifies $X \longrightarrow X^{-1} \longrightarrow X$ (Case $\lambda = 0$).
- For a 3-round delay we use the following identity in $\mathbb{F}_{2^n}$: $(X^{-1}+1)^{-1} = (X+1)^{-1}+1$.
- High differential uniformity every 2 or 3 rounds (controlled by our choices of c_i).

Morse Code with Differential Uniformity



This differential uniformity graph spells “.-- . .-.. -.-. --- -- . - --- .-.. . ..- ...- . -.” (**WELCOMETOLEUVEN**) over 94 rounds ($m = 2$, $\mathbb{F}_{2^6}$).

Overview of Rescue

Affine Space Chains

Weak Designs and Weird Designs

Conclusion

Conclusion

Conclusion

- Bad choice of round constants may lead to affine space chains, but for random round constants this is unlikely.

Conclusion

- Bad choice of round constants may lead to affine space chains, but for random round constants this is unlikely.
- Our weak designs satisfy state-of-the art security arguments (APN Sbox, MDS matrix, wide-trail strategy...). Usual security arguments are not sufficient in the AO context.

Conclusion

- Bad choice of round constants may lead to affine space chains, but for random round constants this is unlikely.
- Our weak designs satisfy state-of-the art security arguments (APN Sbox, MDS matrix, wide-trail strategy...). Usual security arguments are not sufficient in the AO context.
- The principles behind these techniques are applicable to other AOPs, like **Arion**-π and **Griffin**, and were exploited to break them (see `eprint.iacr.org/2024/347` on "**Freelunch Attacks**").

Conclusion

- Bad choice of round constants may lead to affine space chains, but for random round constants this is unlikely.
- Our weak designs satisfy state-of-the art security arguments (APN Sbox, MDS matrix, wide-trail strategy...). Usual security arguments are not sufficient in the AO context.
- The principles behind these techniques are applicable to other AOPs, like **Arion**-π and **Griffin**, and were exploited to break them (see `eprint.iacr.org/2024/347` on "**Freelunch Attacks**").

Thank you for listening!

More on the CICO Problem

Definition (CICO Problem of size c)

Given a permutation P, find x of size $(n - c)$ such that $P(x \mid\mid 0^c) = (* \mid\mid 0^c)$.

More on the CICO Problem

Definition (CICO Problem of size c)

Given a permutation P, find x of size $(n - c)$ such that $P(x \mid\mid 0^c) = (* \mid\mid 0^c)$.

- Given a sponge construction of rate r and capacity c, solving the CICO problem of size c on its inner permutation gives a **collision**.

More on the CICO Problem

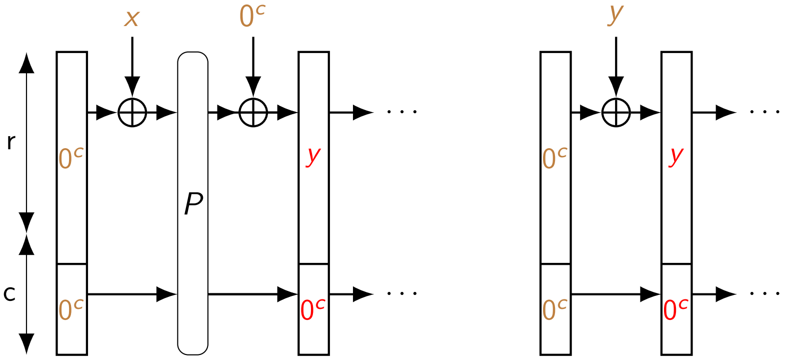
Definition (CICO Problem of size c)

Given a permutation P, find x of size $(n - c)$ such that $P(x \mid\mid 0^c) = (* \mid\mid 0^c)$.

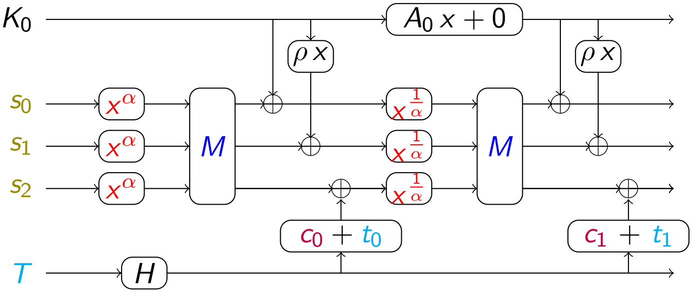
- Given a sponge construction of rate r and capacity c, solving the CICO problem of size c on its inner permutation gives a **collision**.
- There are variants (e.g. given y of size r, find x such that $P(x \mid\mid 0^c) = (y \mid\mid *)$.

Collision from the CICO Problem

- Suppose you know x such that $P(x \mid\mid 0^c) = (y \mid\mid 0^c)$.

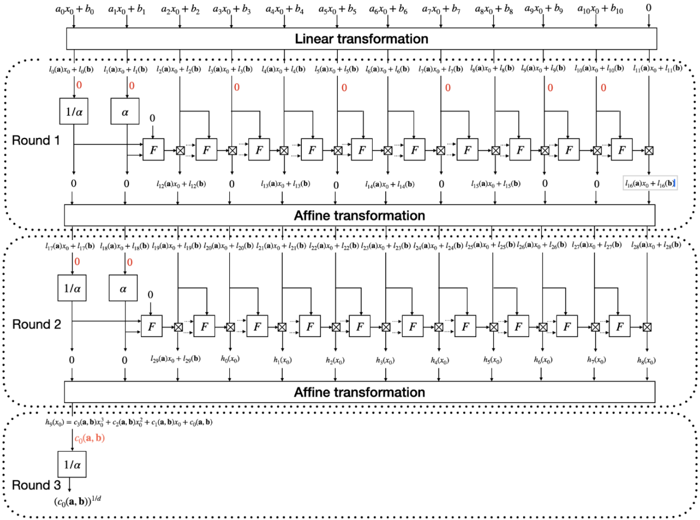


Snare

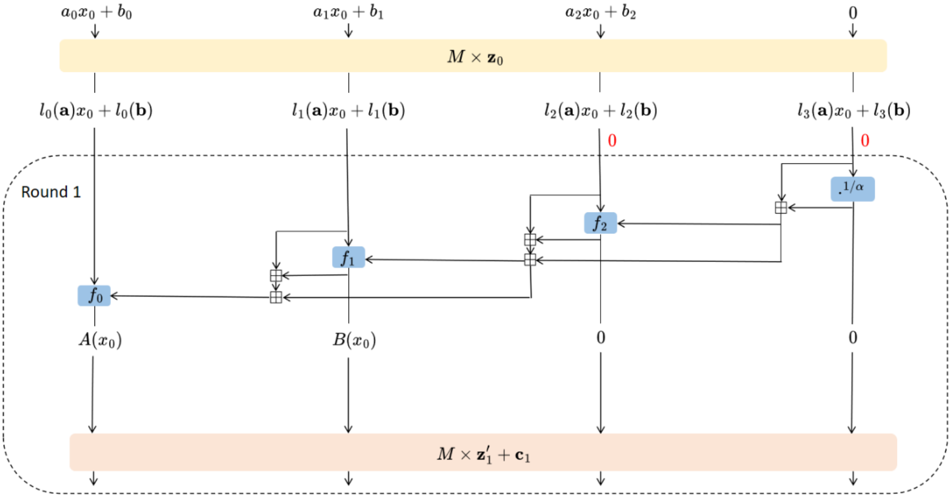


- H is an XOF (eXtendable Output Function), like **SHAKE256**.
- The t_i are the tweak hashes.

Griffin Trick



Arion Trick


$a_0x_0 + b_0$
$a_1x_0 + b_1$
$a_2x_0 + b_2$
0
$M \times \mathbf{z}_0$
$l_0(\mathbf{a})x_0 + l_0(\mathbf{b})$
$l_1(\mathbf{a})x_0 + l_1(\mathbf{b})$
$l_2(\mathbf{a})x_0 + l_2(\mathbf{b})$
$l_3(\mathbf{a})x_0 + l_3(\mathbf{b})$
0
0
Round 1
$\cdot^{1/\alpha}$
f_2
f_1
f_0
$A(x_0)$
$B(x_0)$
0
0
$M \times \mathbf{z}_1' + \mathbf{c}_1$