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

Bergen Workshop, June 2023

Introduction

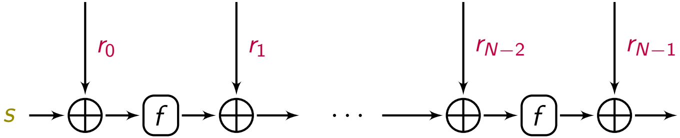
Affine Space Chains

Weak Designs and Curious Designs

Conclusion

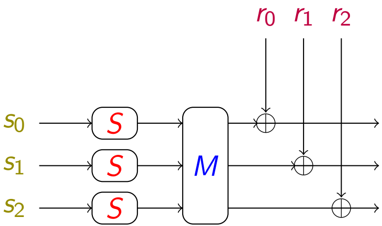
Which Symmetric Primitives?

Which Symmetric Primitives?



The ever-popular Block Cipher construction.

Which Round Function?



The round function of an SPN (Substitution-Permutation Network) Block Cipher. Design basis for the AES, very popular.

Arithmetization-Oriented Symmetric Primitives

- Term coined for the first time in a 2020 paper from Aly et al.
- Symmetric primitives with a “simple” arithmetic description.
- Minimize verification cost in Zero-Knowledge schemes and other advanced protocols.
- Generally defined over a large finite field $\mathbb{F}_q$. ($q \geq 2^{64}$ or so.)
- Heavy use of monomials for nonlinear functions as random permutations are hard to analyze.

Arithmetization-Oriented Symmetric Primitives

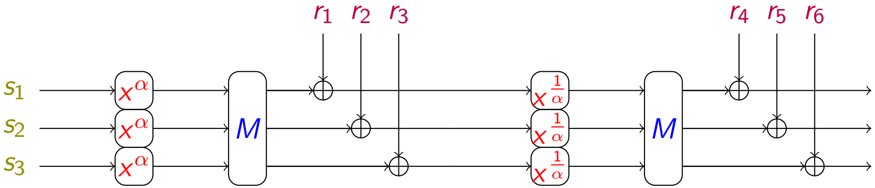
- Term coined for the first time in a 2020 paper from Aly et al.
- Symmetric primitives with a “simple” arithmetic description.
- Minimize verification cost in Zero-Knowledge schemes and other advanced protocols.
- Generally defined over a large finite field $\mathbb{F}_q$. ($q \geq 2^{64}$ or so.)
- Heavy use of monomials for nonlinear functions as random permutations are hard to analyze.

Example

Primitive using the nonlinear component $S : x \mapsto x^3$ (Mimc and variants, Rescue...).

Rescue [AABDS'20]

- Defined in $\mathbb{F}_p \cong \mathbb{Z}/p\mathbb{Z}$ with p prime $\simeq 2^{64}$.
- The S-box alternates between $S : x \mapsto x^\alpha$ and S^{-1} where α is the smallest s.t. S is a permutation.
- Defined for any MDS matrix M and round constants r_i.



2 rounds of Rescue (repeated $N \approx 10$ times).

Rescue's Design Choices

Rescue's Design Choices

- Alternate x^{α} and $x^{\frac{1}{\alpha}}$ for resistance against algebraic attacks.

Rescue's Design Choices

- Alternate x^{α} and $x^{\frac{1}{\alpha}}$ for resistance against algebraic attacks.
- x^{α} has good cryptographic properties (APN for $\alpha = 3$).

Rescue's Design Choices

- Alternate x^α and $x^{\frac{1}{\alpha}}$ for resistance against algebraic attacks.
- x^α has good cryptographic properties (APN for $\alpha = 3$).
- Wide-trail strategy is used, like in the AES, as a security argument.

Rescue's Design Choices

- Alternate x^α and $x^{\frac{1}{\alpha}}$ for resistance against algebraic attacks.
- x^α has good cryptographic properties (APN for $\alpha = 3$).
- Wide-trail strategy is used, like in the AES, as a security argument.
- For the Sbox, having a monomial followed by an affine transformation of the representation like in the AES may be nice, but... no subfield in $\mathbb{F}_p$.

Rescue's Design Choices

- Alternate x^α and $x^{\frac{1}{\alpha}}$ for resistance against algebraic attacks.
- x^α has good cryptographic properties (APN for $\alpha = 3$).
- Wide-trail strategy is used, like in the AES, as a security argument.
- For the Sbox, having a monomial followed by an affine transformation of the representation like in the AES may be nice, but... no subfield in $\mathbb{F}_p$.

Main motivation: Are the usual security arguments sufficient?

Differential Uniformity

Definition

Differential uniformity of a function F:

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

Differential Uniformity

Definition

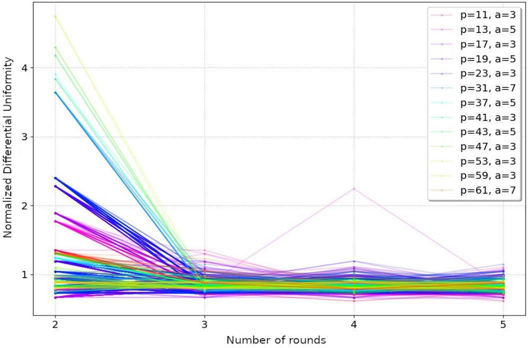
Differential uniformity of a function F:

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

$\rightarrow$ This quantity must be minimized.

High Differential Uniformities in Rescue

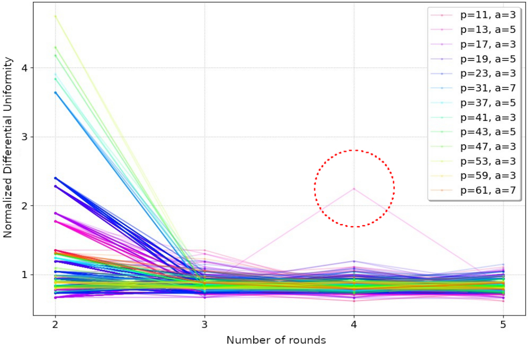
Wide-trail strategy: δ should quickly decrease towards the average random permutation differential uniformity.



Graph taken from [BCLNPW'20], *On the security of the Rescue hash function*. Cryptology ePrint Archive, Paper 2020/820.

High Differential Uniformities in Rescue

Wide-trail strategy: δ should quickly decrease towards the average random permutation differential uniformity.



Graph taken from [BCLNPW'20], *On the security of the Rescue hash function*. Cryptology ePrint Archive, Paper 2020/820

High Differential Uniformities in Rescue

The cause? Affine spaces of dimension 1 nicely mapping from one to another.

$$\begin{pmatrix} z \\ X \end{pmatrix} \xrightarrow{\quad 2 \text{ rounds} \quad} \begin{pmatrix} aX + b \\ cX + d \end{pmatrix} \xrightarrow{\quad 2 \text{ rounds} \quad} \begin{pmatrix} eX + f \\ gX + h \end{pmatrix}$$

High Differential Uniformities in Rescue

The cause? Affine spaces of dimension 1 nicely mapping from one to another.

$$\begin{pmatrix} z \\ X \end{pmatrix} \xrightarrow{\quad 2 \text{ rounds} \quad} \begin{pmatrix} aX + b \\ cX + d \end{pmatrix} \xrightarrow{\quad 2 \text{ rounds} \quad} \begin{pmatrix} eX + f \\ gX + h \end{pmatrix}$$

- 1 round or 3 rounds: the function is not affine.
- Because p is big ($\geq 2^{64}$), affine spaces of dim 1 are also big.

High Differential Uniformities in Rescue

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

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

High Differential Uniformities in Rescue

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

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

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

$$= \begin{pmatrix} e \\ g \end{pmatrix} = \beta$$

High Differential Uniformities in Rescue

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

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

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

$\rightarrow \delta(F) \geq p$

Structure of our work

Introduction

Affine Space Chains

Weak Designs and Curious Designs

Conclusion

Affine Space Chains

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

$$\boldsymbol{a}_0 + \langle \boldsymbol{v}_0 \rangle \longrightarrow \boldsymbol{a}_1 + \langle \boldsymbol{v}_1 \rangle \longrightarrow \ldots \longrightarrow \boldsymbol{a}_N + \langle \boldsymbol{v}_N \rangle$$

Separable Affine Spaces

Definition

An affine space of dimension 1 is separable if and only if there exists a representation of it denoted $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$ such that:

$$\forall\, 1 \leq i \leq m\,,\;\; a_i \cdot v_i = 0\;.$$

or, equivalently, $\mathrm{supp}(\boldsymbol{v}) \cap \mathrm{supp}(\boldsymbol{a}) = \emptyset$.

Separable Affine Spaces

Definition

An affine space of dimension 1 is separable if and only if there exists a representation of it denoted $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$ such that:

$$\forall\, 1 \leq i \leq m\,,\ a_i \cdot v_i = 0\ .$$

or, equivalently, $\operatorname{supp}(\boldsymbol{v}) \cap \operatorname{supp}(\boldsymbol{a}) = \emptyset$.

Examples

- $\begin{pmatrix} a \\ 0 \end{pmatrix} + \left\langle \begin{pmatrix} 0 \\ b \end{pmatrix} \right\rangle$ is a separable affine space for all a and b.

Separable Affine Spaces

Definition

An affine space of dimension 1 is separable if and only if there exists a representation of it denoted $\boldsymbol{a} + \langle \boldsymbol{v} \rangle$ such that:

$$\forall\, 1 \leq i \leq m\,,\ a_i \cdot v_i = 0\ .$$

or, equivalently, $\mathrm{supp}(\boldsymbol{v}) \cap \mathrm{supp}(\boldsymbol{a}) = \emptyset$.

Examples

- $\begin{pmatrix} a \\ 0 \end{pmatrix} + \left\langle \begin{pmatrix} 0 \\ b \end{pmatrix} \right\rangle$ is a separable affine space for all a and b.
- $\begin{pmatrix} 0 \\ 1 \end{pmatrix} + \left\langle \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\rangle$ is not.

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. Also, the image is still separable if and only if there exists* λ *in* $\mathbb{F}_p$ *such that:*

Main Result

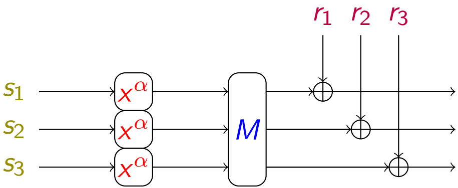
Theorem

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

$$\forall i \in \mathrm{supp}(M \circ S)(\boldsymbol{v}),$$

$$r_i = \lambda (M \circ S)(\boldsymbol{v})_i - (M \circ S)(\boldsymbol{a})_i$$

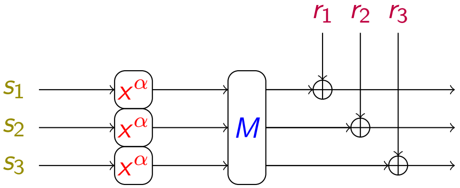
Main Result - Sketch of Proof



Rescue round.

Write elements of $\begin{pmatrix}0\\0\\a\end{pmatrix} + \left\langle \begin{pmatrix}1\\v\\0\end{pmatrix} \right\rangle$ as $\begin{pmatrix}s_1\\s_2\\s_3\end{pmatrix} = \begin{pmatrix}X\\vX\\a\end{pmatrix}$.

Main Result - Sketch of Proof

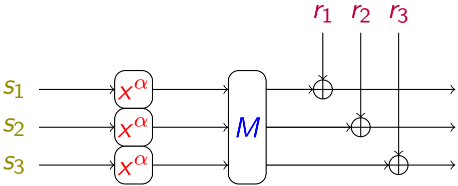


Rescue round.

$$\begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} X \\ vX \\ a \end{pmatrix} \longrightarrow \begin{pmatrix} X^\alpha \\ v^\alpha X^\alpha \\ a^\alpha \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ a^\alpha \end{pmatrix} + X^\alpha \begin{pmatrix} 1 \\ v^\alpha \\ 0 \end{pmatrix}$$

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

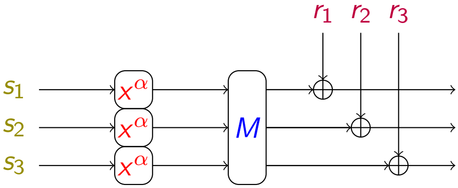
Main Result - Sketch of Proof



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 Result - Sketch of Proof



Rescue round.

$$M\begin{pmatrix}0\\0\\a^\alpha\end{pmatrix}+X^\alpha M\begin{pmatrix}1\\v^\alpha\\0\end{pmatrix}\longrightarrow M\begin{pmatrix}0\\0\\a^\alpha\end{pmatrix}+\begin{pmatrix}r_1\\r_2\\r_3\end{pmatrix}+X^\alpha M\begin{pmatrix}1\\v^\alpha\\0\end{pmatrix}$$

Main Result - Sketch of Proof

$$M \begin{pmatrix} 0 \\ 0 \\ a^{\alpha} \end{pmatrix} + \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} + \left\langle M \begin{pmatrix} 1 \\ v^{\alpha} \\ 0 \end{pmatrix} \right\rangle$$

Main Result - Sketch of Proof

$$M\begin{pmatrix}0\\0\\a^{\alpha}\end{pmatrix}+\begin{pmatrix}r_1\\r_2\\r_3\end{pmatrix}+\left\langle M\begin{pmatrix}1\\v^{\alpha}\\0\end{pmatrix}\right\rangle$$

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

$$M\begin{pmatrix}1\\v^{\alpha}\\0\end{pmatrix} \text{ and } M\begin{pmatrix}0\\0\\a^{\alpha}\end{pmatrix}+\begin{pmatrix}r_1\\r_2\\r_3\end{pmatrix}+\lambda M\begin{pmatrix}1\\v^{\alpha}\\0\end{pmatrix}$$

have disjoint supports. □

Introduction

Affine Space Chains

Weak Designs and Curious Designs

Conclusion

Our Designs

- Stir, a weak instance of Rescue.

[1] Thomas Peyrin and Haoyang Wang, *The MALICIOUS Framework: Embedding Backdoors into Tweakable Block Ciphers*

Our Designs

- Stir, a weak instance of Rescue.
- Snare, a tweakable cipher with a secret weak tweak. Directly based on the MALICIOUS framework[1].

[1] Thomas Peyrin and Haoyang Wang, *The MALICIOUS Framework: Embedding Backdoors into Tweakable Block Ciphers*

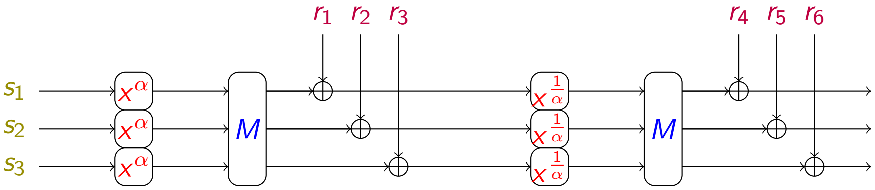
Our Designs

- Stir, a weak instance of Rescue.
- Snare, a tweakable cipher with a secret weak tweak. Directly based on the MALICIOUS framework[1].
- AES-like ciphers where we can introduce and control differential uniformity spikes.

[1] Thomas Peyrin and Haoyang Wang, *The MALICIOUS Framework: Embedding Backdoors into Tweakable Block Ciphers*

Stir

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

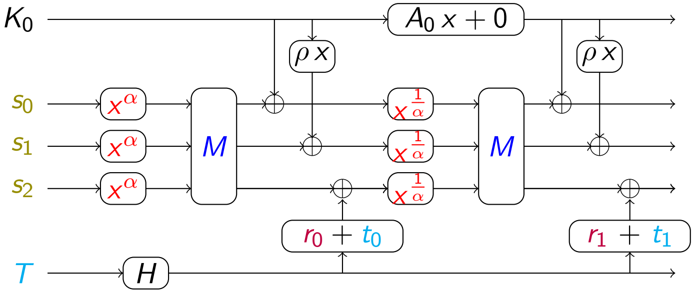


Stir

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

- Yields $p \approx 2^{64}$ solutions to the "CICO problem". This breaks security arguments in sponge constructions.

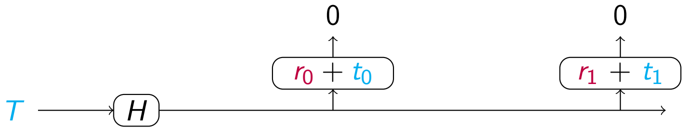
Snare



- H is some hash function, like SHAKE256.
- The t_i are the tweak hashes.

Snare

Idea: Choose $r_i = -H(T^*)_i$ for some secret tweak T^*.
$\rightarrow$ When $T = T^*$, r_i and t_i annihilate one another and an invariant vector space appears.



Snare

$$\left\langle \begin{pmatrix} 1 \\ \rho \\ 0 \end{pmatrix} \right\rangle \xrightarrow{\text{1 round}} \left\langle \begin{pmatrix} 1 \\ \rho \\ 0 \end{pmatrix} \right\rangle \longrightarrow \dots \longrightarrow \left\langle \begin{pmatrix} 1 \\ \rho \\ 0 \end{pmatrix} \right\rangle$$

Snare

$$\begin{pmatrix}1\\ \rho\\ 0\end{pmatrix} \xrightarrow{\text{1 round}} P_1(K_0)\begin{pmatrix}1\\ \rho\\ 0\end{pmatrix} \longrightarrow \dots \longrightarrow P_n(K_0)\begin{pmatrix}1\\ \rho\\ 0\end{pmatrix}$$

Snare

$$\begin{pmatrix} 1 \\ \rho \\ 0 \end{pmatrix} \xrightarrow{\text{1 round}} P_1(K_0) \begin{pmatrix} 1 \\ \rho \\ 0 \end{pmatrix} \longrightarrow \dots \longrightarrow P_n(K_0) \begin{pmatrix} 1 \\ \rho \\ 0 \end{pmatrix}$$

- Retrieve K_0 with multivariate polynomial solving (Gröbner bases), with m times less equations as the general case.

$\rightarrow$ Algebraic attack complexity put to the mth root!

Affine Space Chain vs Affine Function

- Last 2 designs 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?

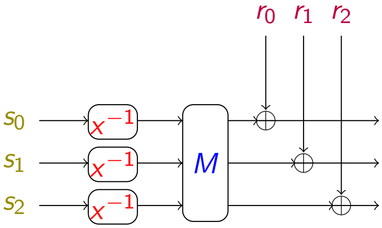
Affine Space Chain vs Affine Function

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



Morse Code with Differential Uniformity

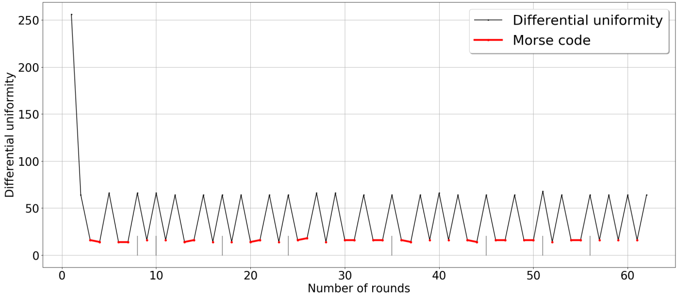
Idea: Same strategy as Snare, but make it so that the mapping from the input to output affine space is *itself* affine every 2 or 3 rounds!

Morse Code with Differential Uniformity

Idea: Same strategy as SNARE, 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 $\lambda = 0$).
- High differential uniformity every 2 or 3 rounds (controlled by our choices of r_i).

Morse Code with Differential Uniformity



This differential uniformity graph spells "-- . .-. .-. -.-- -..- -- .- ..." (MERRYXMAS) over 62 rounds ($m = 2$, $\mathbb{F}_{2^6}$).

Introduction

Affine Space Chains

Weak Designs and Curious Designs

Conclusion

Conclusion

Conclusion

- Bad choice of round constants may lead to high differential uniformities.

Conclusion

- Bad choice of round constants may lead to high differential uniformities.
- 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 high differential uniformities.
- 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.
- Look out for similar algebraic shenanigans in AO primitives.

Thank you for listening!

Questions?