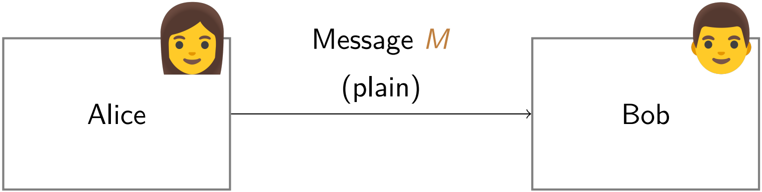
Propagation of Subspaces in Primitives with Monomial Sboxes: Applications to Rescue and Variants of the AES

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

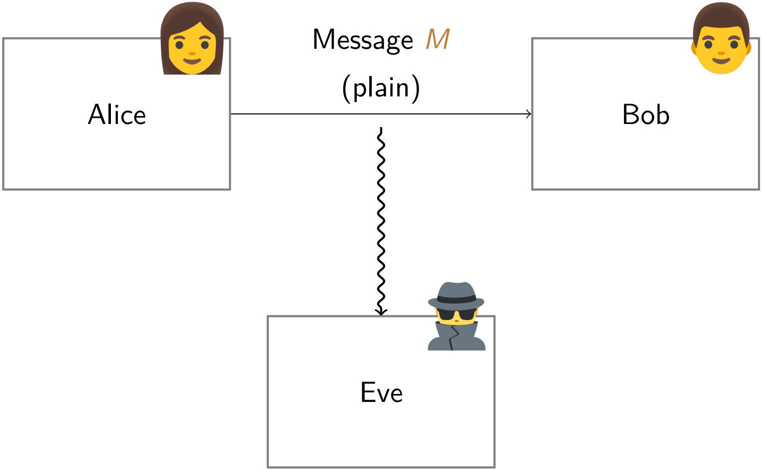
[1]Inria Paris

ALMASTY Seminar, Jussieu

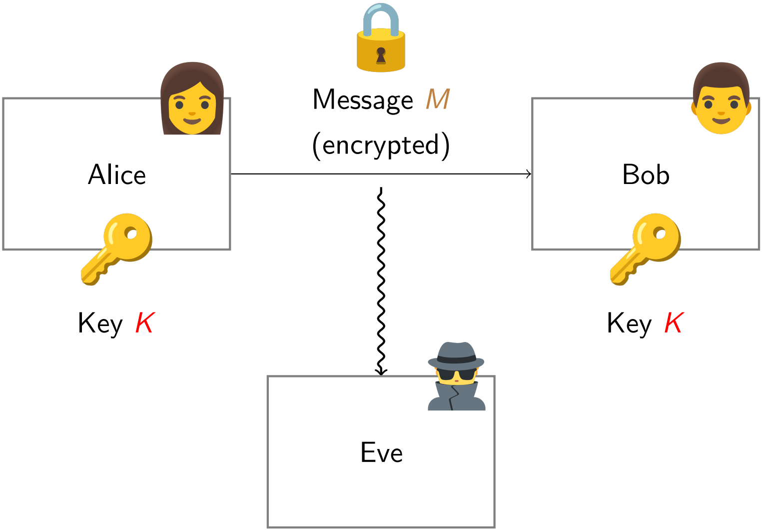
What is Symmetric Cryptography?



What is Symmetric Cryptography?

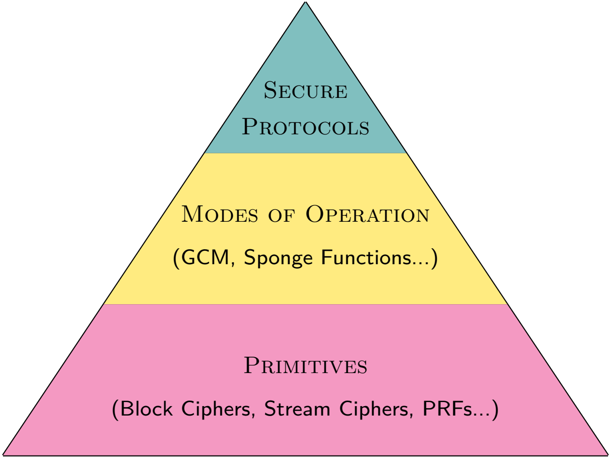


What is Symmetric Cryptography?



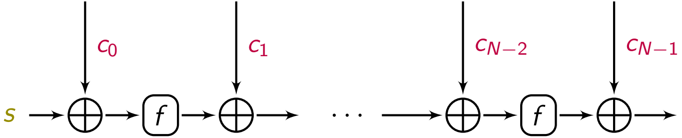
What is a Symmetric Primitive?

"Security": confidentiality, authentication, integrity...



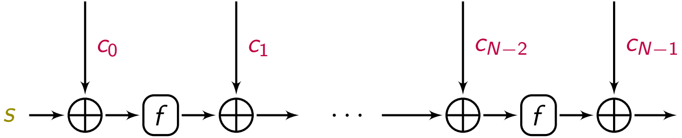
What is a Symmetric Primitive?

What is a Symmetric Primitive?



The ever-popular Block Cipher construction.

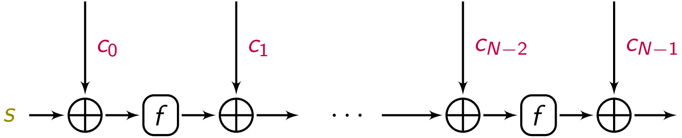
What is a Symmetric Primitive?



The ever-popular Block Cipher construction.

- Key-dependent $c_i(K)$: family of permutations E_K.

What is a Symmetric Primitive?



The ever-popular Block Cipher construction.

- Key-dependent $c_i(K)$: family of permutations E_K.
- Fixed, public c_i: pseudo-random permutation (useful for **hash functions**, PRFs, XOFs...)

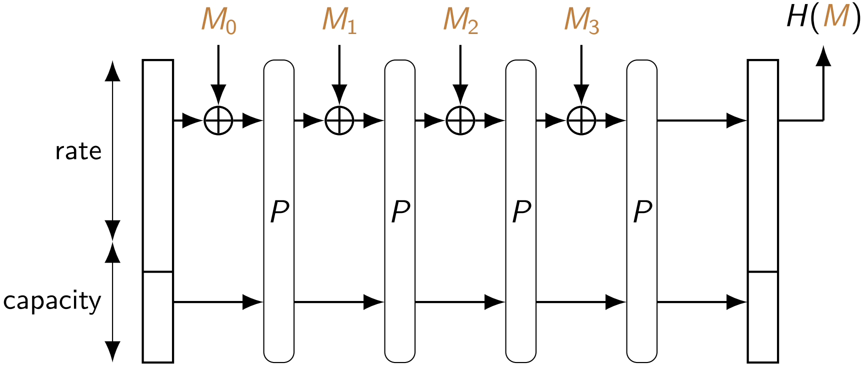
What is a Hash Function?

Definition

A hash function is a function that maps an input of **any size** in $\mathbb{F}_q$ to an element of $\mathbb{F}_q^r$ for a **fixed** integer r.

- **collision resistance**: hard to find $x,\ y$ such that $H(x) = H(y)$.
- **preimage resistance**: given $y \in \mathbb{F}_q^r$, hard to find x such that $H(x) = y$.
- **second preimage resistance**: given x, hard to find x' such that $H(x) = H(x')$.

Sponge Hash Functions



A sponge construction, originally designed for the standard **SHA-3**.
P is, for example, a **fixed-key Block Cipher**.

Arithmetization-Oriented Symmetric Primitives

Arithmetization-Oriented Symmetric Primitives

- Advanced protocols (Zero-Knowledge proofs, MPC, FHE...) call for primitives with a "simple" arithmetic description (unlike the **AES** or **SHA-3**), sometimes over $\mathbb{F}_p$ for a large p.

Arithmetization-Oriented Symmetric Primitives

- Advanced protocols (Zero-Knowledge proofs, MPC, FHE...) call for primitives with a "simple" arithmetic description (unlike the **AES** or **SHA-3**), sometimes over $\mathbb{F}_p$ for a large p.

Classic: binary operations, algebraically complex nonlinear layers over a small field ($\mathbb{F}_{2^8}$) $\rightsquigarrow$ **AOP**: arithmetic operations, algebraically simple nonlinear layers over a large (sometimes prime) field $\mathbb{F}_q$, $q \geq 2^{64}$.

Arithmetization-Oriented Symmetric Primitives

- Advanced protocols (Zero-Knowledge proofs, MPC, FHE...) call for primitives with a "simple" arithmetic description (unlike the **AES** or **SHA-3**), sometimes over $\mathbb{F}_p$ for a large p.

Classic: binary operations, algebraically complex nonlinear layers over a small field ($\mathbb{F}_{2^8}$) $\rightsquigarrow$ **AOP**: arithmetic operations, algebraically simple nonlinear layers over a large (sometimes prime) field $\mathbb{F}_q$, $q \geq 2^{64}$.

Example

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

Arithmetization for Zero-Knowledge

- **Zero-Knowledge proof**: prove that a statement on my private data is true, and reveal nothing else.

Arithmetization for Zero-Knowledge

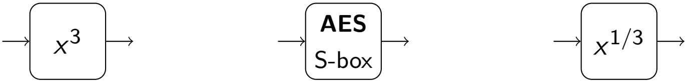
- **Zero-Knowledge proof**: prove that a statement on my private data is true, and reveal nothing else.
- Implemented using "constraint systems" (R1CS, AIR, Plonk...). **Less constraints = Better performance**.

Function → Arithmetic circuit → Set of constraints

Arithmetization for Zero-Knowledge

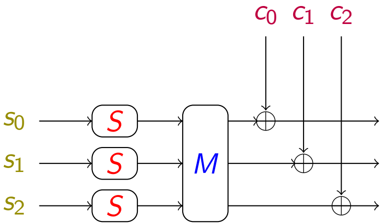
- **Zero-Knowledge proof**: prove that a statement on my private data is true, and reveal nothing else.
- Implemented using "constraint systems" (R1CS, AIR, Plonk...). **Less constraints = Better performance**.

Function → Arithmetic circuit → Set of constraints



x^3	AES S-box	$x^{1/3}$
Low degree	**High** degree	**High** degree
Few constraints	**Many** constraints	**Few** constraints
		(Because low degree inverse)

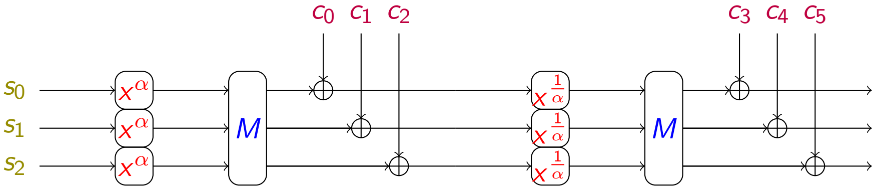
A Typical Round Function



The round function of an SPN Block Cipher. Design basis for the **AES**.

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

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.
- Low verification cost, high degree overall.

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 $\alpha = 3$).

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 $\alpha = 3$).
- 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.
- Low verification cost, high degree overall.
- 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.
- Low verification cost, high degree overall.
- x^α has good cryptographic properties (APN for $\alpha = 3$).
- Wide-trail strategy is used, like in the **AES**, as a security argument.

Main motivation: Are the usual security arguments sufficient?

Differential Uniformity

Definition

Differential uniformity of a function F:

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

Differential Uniformity

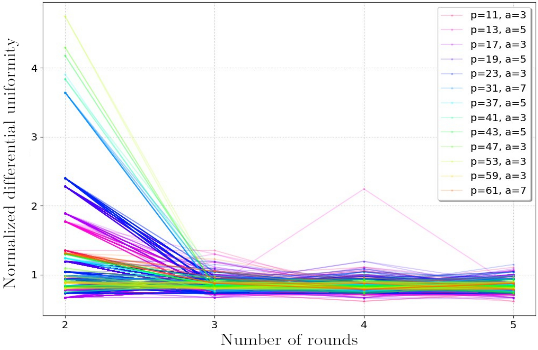
Definition

Differential uniformity of a function F:

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

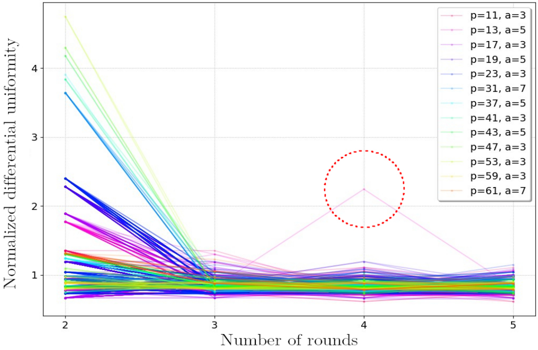
$\rightarrow$ This quantity must be minimized.

High Differential Uniformities in Rescue



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

High Differential Uniformities in Rescue



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

High Differential Uniformities in Rescue

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

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

High Differential Uniformities in Rescue

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

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

- 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

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

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

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

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

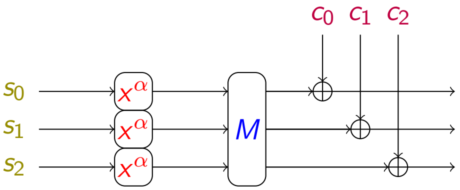
Structure of our work

Affine Space Chains

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

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

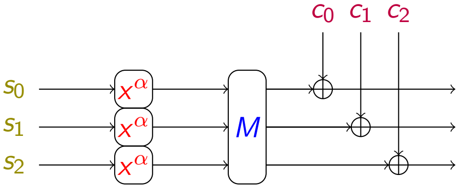
Main Observation



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 Observation



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 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, $\mathrm{supp}(\boldsymbol{v}) \cap \mathrm{supp}(\boldsymbol{a}) = \varnothing$.

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, $\mathrm{supp}(\boldsymbol{v}) \cap \mathrm{supp}(\boldsymbol{a}) = \varnothing$.

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:

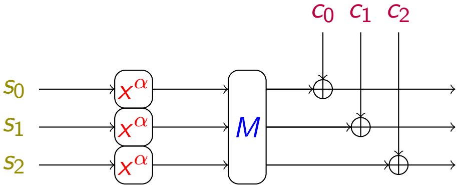
$$\forall\, 1 \leq i \leq m\,,\ a_i \cdot v_i = 0\,.$$

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

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.

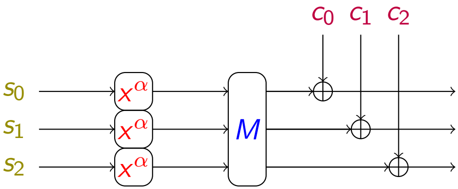
Main Observation



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 Observation



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 Observation

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

Main Observation

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

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. 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:

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

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

Introduction

The Rescue Family

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*

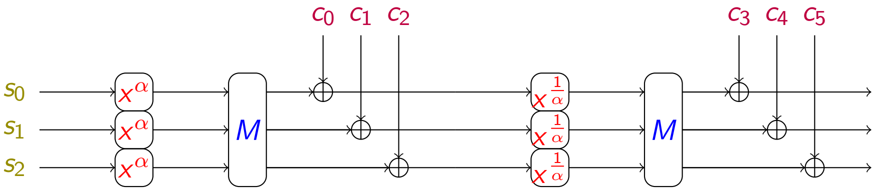
Our Designs

- 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 c are carefully chosen to impose one affine space chain over the whole permutation.



Stir

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

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

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

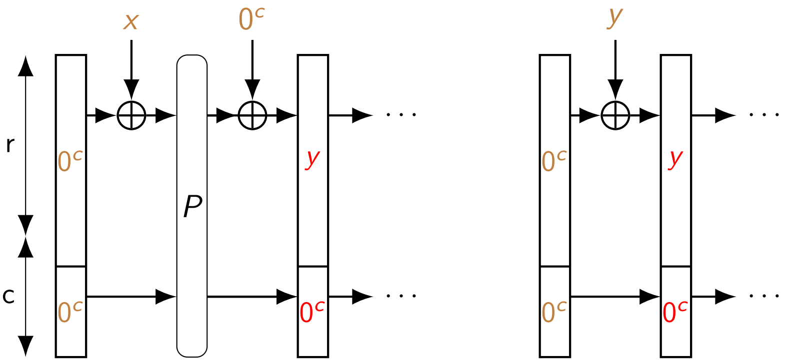
Definition (CICO Problem of size c)

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

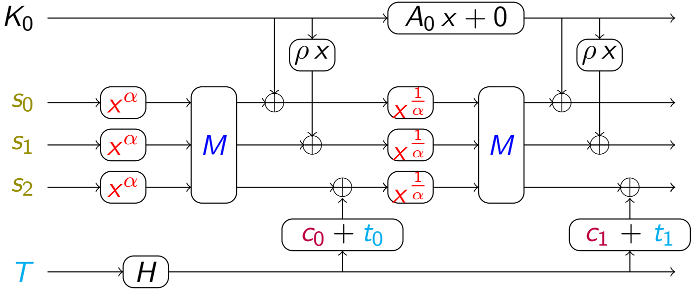
- Given a sponge construction of rate r and capacity c, solving the CICO problem of size c on its inner permutation gives a **collision**.
- There are variants (e.g. given y of size r, find x such that $P(x \mid\mid 0^c) = (y \mid\mid *)$.

Collision from the CICO Problem

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



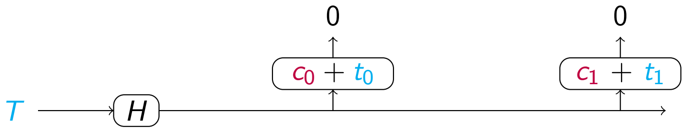
Snare



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

Snare

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



Snare

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

Snare

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

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.

→ 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?

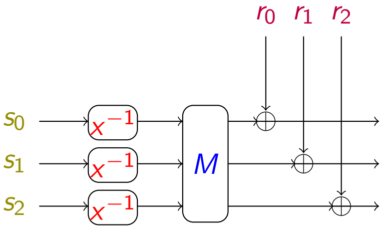
Affine Space Chain vs Affine Function

- 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

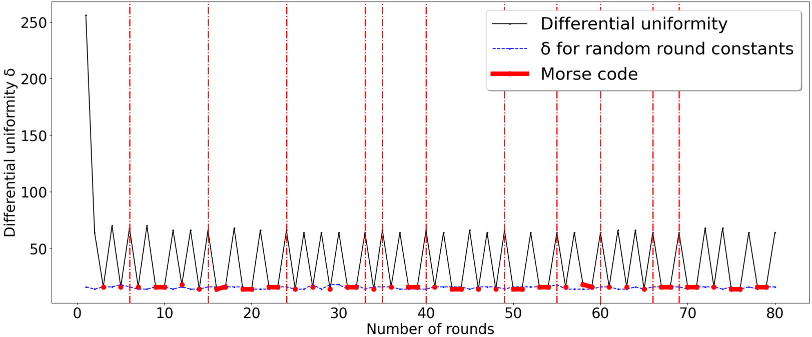
Idea: Same strategy as Snare, but make it so that the mapping from the input to output affine space is *itself* affine every 2 or 3 rounds!

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 c_i).

Morse Code with Differential Uniformity



This differential uniformity graph spells “–– . .–. .–. –.–– –..– –– .– . . .” (ILOVEALMASTY) over 80 rounds ($m = 2$, $\mathbb{F}_{2^6}$).

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

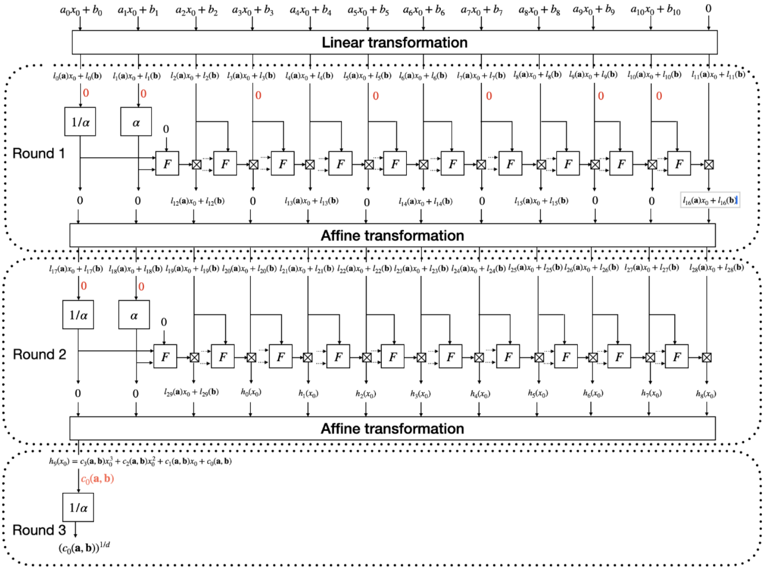
- Bad choice of round constants may lead to high differential uniformities.
- Our weak designs satisfy state-of-the art security arguments (APN Sbox, MDS matrix, wide-trail strategy...). Usual security arguments are not sufficient in the AO context.
- The principles behind these techniques are applicable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

Conclusion

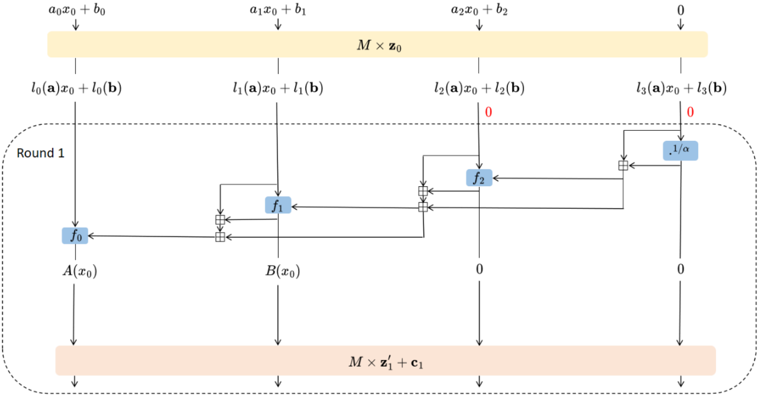
- Bad choice of round constants may lead to high differential uniformities.
- Our weak designs satisfy state-of-the art security arguments (APN Sbox, MDS matrix, wide-trail strategy...). Usual security arguments are not sufficient in the AO context.
- The principles behind these techniques are applicable 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!

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
$.^{1/\alpha}$
f_2
f_1
f_0
$A(x_0)$
$B(x_0)$
0
0
$M \times \mathbf{z}_1' + \mathbf{c}_1$