Cryptanalysis of `Monolith` using rebound attacks

Pierre Galissant [1]   Guilhem Jazeron [1]   Léo Perrin [1]

[1]INRIA Paris

March 31, 2025

Outline

Plan of this Section

1. The Monolith family of permutations
2. 4-round rebound attack on Monolith-64
3. Open problems

Context

- Arithmetization-oriented family of permutations : efficient in *incrementally verifiable computation* (IVC) schemes that allow lookups.
- Family composed of 4 permutations $f : \mathbb{F}_p^t \to \mathbb{F}_p^t$.
- Can be turned into a hash function (sponge construction) or compression function $(x \in \mathbb{F}_p^t \mapsto \mathrm{Tr}_{t/2}(f(x) + x))$
- `Monolith-64` claims 128 bits of security, `Monolith-31` claims 124 bits of security.

The Monolith design

SPN design with :

- A MDS matrix : **diffusion**.
- A partial layer of Split-and-Lookup SBoxes (Bar) : **efficient** (lookup tables), **high algebraic degree**.
- A generalized Feistel layer where for each branch, $y_i = x_i + x_{i-1}^2$: very strong **differential properties** (PN).
- Addition of round constants (AddC).
- 6 **rounds** for all versions.

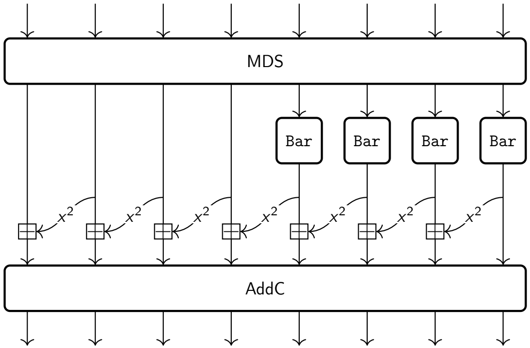


Figure: The Monolith-64 with 8 branches round function

Monolith parameters

	Monolith-64 $p = p_{\text{goldlilocks}} = 2^{64} - 2^{32} + 1$	Monolith-31 $p = p_{\text{mersenne}} = 2^{31} - 1$
Compression function	$t = 8$ $u = 4$	$t = 16$ $u = 8$
Hash function	$t = 12$ $u = 4$	$t = 24$ $u = 8$

Table: Overview of the 4 instances of Monolith
t : number of branches
u : number of Bar SBoxes per round.

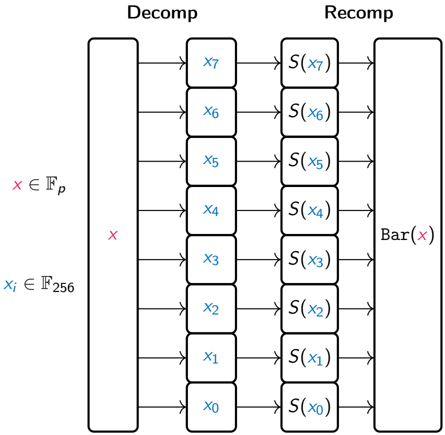
In orange, the instance upon which we will describe an attack.

Split-and-Lookups

The Split-and-Lookup construction

Example of $p_{\text{goldlilocks}}$

1. $x \in \mathbb{F}_p = x_0 + 2^8 x_1 + \cdots + 2^{56} x_7$
2. Apply a small SBox $S : \mathbb{F}_{256} \to \mathbb{F}_{256}$ to each x_i
3. Obtain the ouput $\texttt{Bar}(x) = S(x_0) + 2^8 S(x_1) + \cdots + 2^{56} S(x_7)$



Split-and-Lookups

The Split-and-Lookup construction

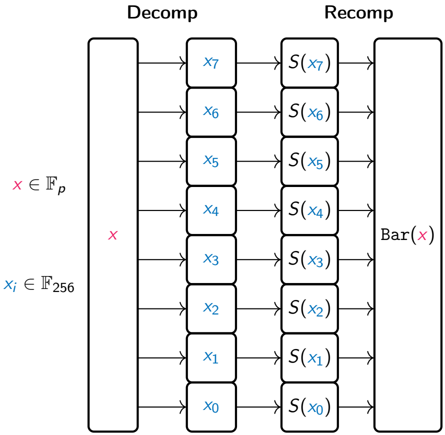
Example of $p_{\text{goldlilocks}}$

1. $x \in \mathbb{F}_p = x_0 + 2^8 x_1 + \cdots + 2^{56} x_7$
2. Apply a small SBox $S : \mathbb{F}_{256} \to \mathbb{F}_{256}$ to each x_i
3. Obtain the ouput
   $\mathtt{Bar}(x) = S(x_0) + 2^8 S(x_1) + \cdots + 2^{56} S(x_7)$

In Monolith-64

High algebraic degree ! But...

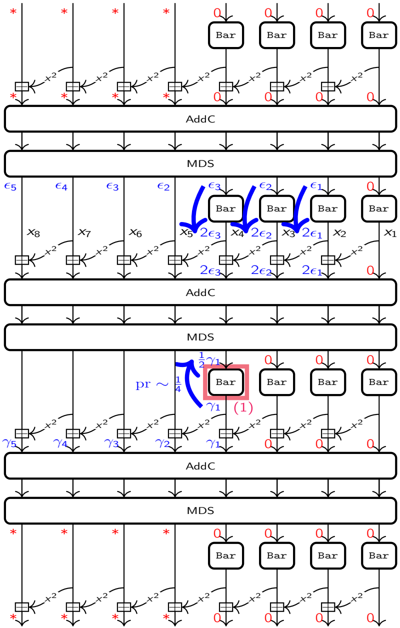
- $S(x) = (x \oplus [(\overline{x} \lll 1) \wedge (x \lll 2) \wedge (x \lll 3)]) \lll 1$
- $S(x) = 2x$ with good probability.
- Hence, $\mathtt{Bar}(x) = 2x$ with probability $\sim 2^{-22}$
- **Weak differential properties:** $1 \xrightarrow{\mathtt{Bar}} 2$ with probability $\frac{62}{256} \sim \frac{1}{4}$



Plan of this Section

Overview of the attacks

Limited birthday attack over 4 rounds: maps a dimension 4 subspace of $\mathbb{F}_p^8$ to a dimension 4 subspace.

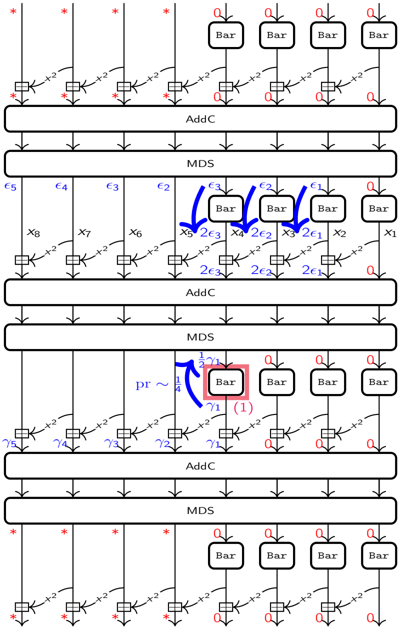


Overview of the attacks

Limited birthday attack over 4 rounds: maps a dimension 4 subspace of $\mathbb{F}_p^8$ to a dimension 4 subspace.

Description of the attack

1. Choose γ_1 with $\frac{1}{2}\gamma_1 \xrightarrow{Bar} \gamma_1$ with probability $\sim \frac{1}{4}$

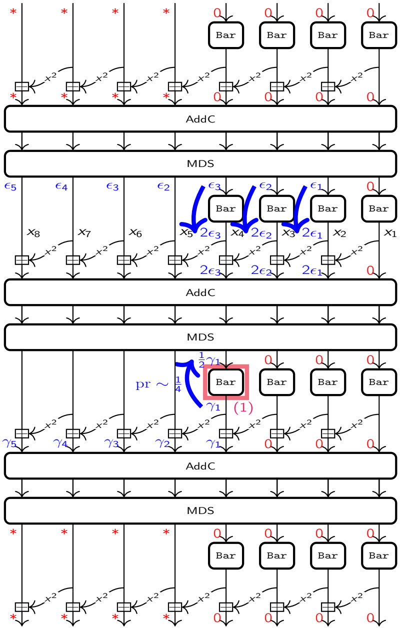


Overview of the attacks

Limited birthday attack over 4 rounds: maps a dimension 4 subspace of $\mathbb{F}_p^8$ to a dimension 4 subspace.

Description of the attack

1. Choose γ_1 with $\frac{1}{2}\gamma_1 \xrightarrow{Bar} \gamma_1$ with probability $\sim \frac{1}{4}$
2. Choose $\epsilon_1, \epsilon_2, \epsilon_3$ with $\epsilon_i \xrightarrow{Bar} 2\epsilon_i$ with probability at least T.

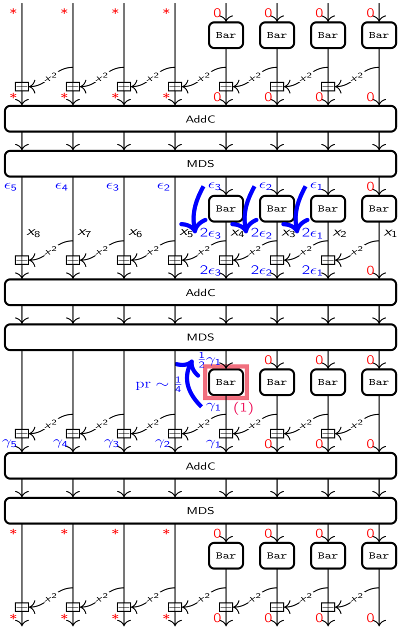


Overview of the attacks

Limited birthday attack over 4 rounds: maps a dimension 4 subspace of $\mathbb{F}_p^8$ to a dimension 4 subspace.

Description of the attack

1. Choose γ_1 with $\frac{1}{2}\gamma_1 \xrightarrow{Bar} \gamma_1$ with probability $\sim \frac{1}{4}$
2. Choose $\epsilon_1, \epsilon_2, \epsilon_3$ with $\epsilon_i \xrightarrow{Bar} 2\epsilon_i$ with probability at least T.
3. Assume the SBox (1) behaves as $x \mapsto 2x$, with probability 2^{-22}.

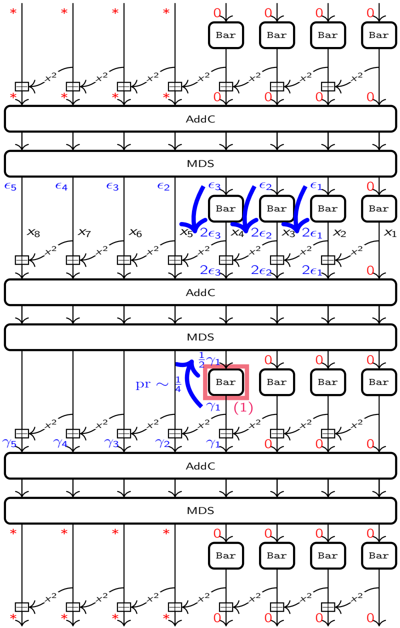


Overview of the attacks

Limited birthday attack over 4 rounds: maps a dimension 4 subspace of $\mathbb{F}_p^8$ to a dimension 4 subspace.

Description of the attack

1. Choose γ_1 with $\frac{1}{2}\gamma_1 \xrightarrow{Bar} \gamma_1$ with probability $\sim \frac{1}{4}$
2. Choose $\epsilon_1, \epsilon_2, \epsilon_3$ with $\epsilon_i \xrightarrow{Bar} 2\epsilon_i$ with probability at least T.
3. Assume the SBox (1) behaves as $x \mapsto 2x$, with probability 2^{-22}.
4. Choose $\epsilon_2, \dots, \epsilon_5$ such that the 4 last branches through M^{-1} are 0, and $\gamma_2, \dots, \gamma_5$ such that the last 4 branches through M are 0

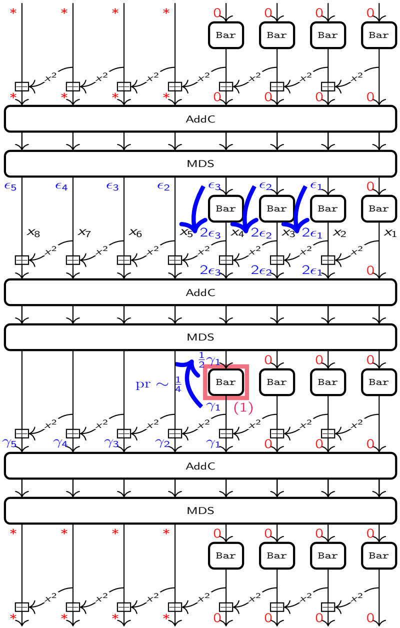


Overview of the attacks

Limited birthday attack over 4 rounds: maps a dimension 4 subspace of $\mathbb{F}_p^8$ to a dimension 4 subspace.

Description of the attack

1. Choose γ_1 with $\frac{1}{2}\gamma_1 \xrightarrow{Bar} \gamma_1$ with probability $\sim \frac{1}{4}$
2. Choose $\epsilon_1, \epsilon_2, \epsilon_3$ with $\epsilon_i \xrightarrow{Bar} 2\epsilon_i$ with probability at least T.
3. Assume the SBox (1) behaves as $x \mapsto 2x$, with probability 2^{-22}.
4. Choose $\epsilon_2, \ldots, \epsilon_5$ such that the 4 last branches through M^{-1} are 0, and $\gamma_2, \ldots, \gamma_5$ such that the last 4 branches through M are 0
5. Solve the polynomial system of degree 3 in $x_1 \ldots x_8$ to make differentials match.



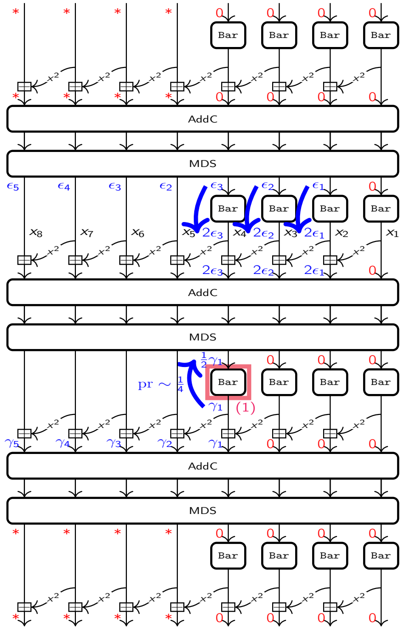
Overview of the attacks

Limited birthday attack over 4 rounds: maps a dimension 4 subspace of $\mathbb{F}_p^8$ to a dimension 4 subspace.

Description of the attack

1. Choose γ_1 with $\frac{1}{2}\gamma_1 \xrightarrow{Bar} \gamma_1$ with probability $\sim \frac{1}{4}$
2. Choose $\epsilon_1, \epsilon_2, \epsilon_3$ with $\epsilon_i \xrightarrow{Bar} 2\epsilon_i$ with probability at least T.
3. Assume the SBox (1) behaves as $x \mapsto 2x$, with probability 2^{-22}.
4. Choose $\epsilon_2, \ldots, \epsilon_5$ such that the 4 last branches through M^{-1} are 0, and $\gamma_2, \ldots, \gamma_5$ such that the last 4 branches through M are 0
5. Solve the polynomial system of degree 3 in $x_1 \ldots x_8$ to make differentials match.

Need to **repeat** these steps $2^{22} \cdot 2^2 \cdot T^{-3}$ to ensure the assumptions hold.



Generating enough differentials

The problem...

- In order for all the assumptions to hold, we need to repeat the attack enough times.
- Trade-off: if T, the probability that the differentials go through Bar is high, the number N of differentials will be low (few good differentials), and $2^{-22} \cdot 2^{-2} \cdot p^3 \cdot N \ll 1$

The solution !

- We deliberately choose to active 3 Bar SBoxes to have more differentials to pass the 2^{-22} probability that $\mathtt{Bar}(x) = 2x$
- Bars $\approx$ parallel application of 8 small SBoxes S operating on $\mathbb{F}_{256}$
- Then we can choose activation patterns *inside* Bars.

Generating enough differentials

The solution, continued

1. Let T be some threshold probability
2. X_1 be the set of differentials through the small S that have probability $\geq T$, X_2 those with probability $\geq T^{1/2}$, X_3 those with probability $\geq T^{1/3}$
3. Then we take, on the big Bar SBox:
   - Differentials from X_1 activating one small S,
   - Pairs of differentials from X_2 activating two small S,
   - etc...
4. In total, we generate:

$$\binom{8}{1} \cdot (\#X_1)^1 + \binom{8}{2} \cdot (\#X_2)^2 + \binom{8}{3} \cdot (\#X_3)^3$$

differentials with probability $\geq T$ on Bar.

Generating enough differentials

The solution, continued

1. Let T be some threshold probability
2. X_1 be the set of differentials through the small S that have probability $\geq T$, X_2 those with probability $\geq T^{1/2}$, X_3 those with probability $\geq T^{1/3}$
3. Then we take, on the big Bar SBox:
   - Differentials from X_1 activating one small S,
   - Pairs of differentials from X_2 activating two small S,
   - etc...
4. In total, we generate:

$$\binom{8}{1} \cdot (\#X_1)^1 + \binom{8}{2} \cdot (\#X_2)^2 + \binom{8}{3} \cdot (\#X_3)^3$$

differentials with probability $\geq T$ on Bar.

- $T = \frac{1}{64}$: 2^{42} differentials, the attack succeeds with probability 0.59
- $T = \frac{1}{256}$: 2^{52} differentials, the attacks succeeds with probability 0.99999

Complexity of the attacks

The complexity is caused by two factors:

- Repeat the solving step: 2^{52} times.
- The complexity of solving a system of polynomial 8 polynomial equations of degree 3 using Gröbner basis: upper bounded by $\mathcal{O}(2^{39})$ using generic formulas.

In total: $\mathcal{O}(2^{91})$.

The *generic attack* for mapping a 4-dimensional subspace to a 4-dimensional subspace is in $\mathcal{O}(2^{128})$ with words of size 64 bits.

Plan of this Section

Open problems

How does this attack extend to more branches ?

- With 12 branches, more degrees of freedom.
- How much stronger does this attack become ?

Open problems

How does this attack extend to more branches ?

- With 12 branches, more degrees of freedom.
- How much stronger does this attack become ?

`Monolith-31` ?

- For `Monolith-31`, twice as many branches: how does the attack adapt ?
- Is doubling the number of branches really the good way to keep the same security level ?

Open problems

How does this attack extend to more branches ?

- With 12 branches, more degrees of freedom.
- How much stronger does this attack become ?

`Monolith-31` ?

- For `Monolith-31`, twice as many branches: how does the attack adapt ?
- Is doubling the number of branches really the good way to keep the same security level ?

Gröbner basis computation time

- In practice, systems generated from `Monolith` take much less time to solve than generic ones.
- Why ? How to theoretically assess the complexity of solving such systems ?

Open problems

How does this attack extend to more branches ?

- With 12 branches, more degrees of freedom.
- How much stronger does this attack become ?

`Monolith-31` ?

- For `Monolith-31`, twice as many branches: how does the attack adapt ?
- Is doubling the number of branches really the good way to keep the same security level ?

Gröbner basis computation time

- In practice, systems generated from `Monolith` take much less time to solve than generic ones.
- Why ? How to theoretically assess the complexity of solving such systems ?

Thank you ! Questions ?