Improvements of Differential Meet-in-the-Middle Cryptanalysis

Zahra Ahmadian[1] Akram Khalesi[1] Dounia M'Foukh[2] Hossein Moghimi[1] María Naya-Plasencia[2]

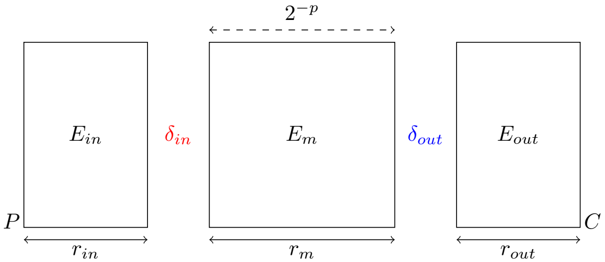
[1]Shahid Beheshti University

[2]Inria Paris

European Research Council
Established by the European Commission

Table of contents

Differential Meet-in-the-Middle [BDD+23]

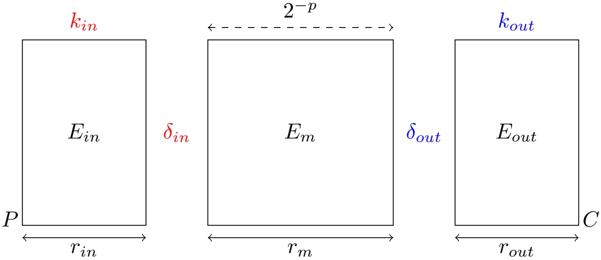


We generate 2^p pairs (P, C).

$$P \rightarrow 2^{|k_{in}|} P' \text{ and } C \rightarrow 2^{|k_{out}|} C'.$$

We keep the candidates $(P, P', C, C', k_{in}, k_{out})$ such that $P' = E^{-1}(C')$.

Differential Meet-in-the-Middle [BDD+23]

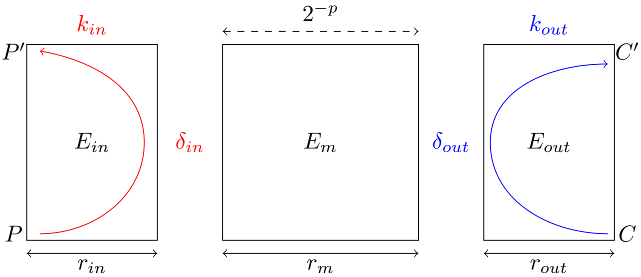


We generate 2^p pairs (P, C).

$$P \to 2^{|k_{in}|} P' \text{ and } C \to 2^{|k_{out}|} C'.$$

We keep the candidates $(P, P', C, C', k_{in}, k_{out})$ such that $P' = E^{-1}(C')$.

Differential Meet-in-the-Middle [BDD+23]



We generate 2^p pairs (P, C).

$$P \rightarrow 2^{|k_{in}|} P' \text{ and } C \rightarrow 2^{|k_{out}|} C'.$$

We keep the candidates $(P, P', C, C', k_{in}, k_{out})$ such that $P' = E^{-1}(C')$.

Extensions of the attack in the original paper

Reducing Data complexity :

- ⇝ Impose x bits conditions on the plaintexts P and P'.
- ⇝ Useful in the case that the whole codebook is needed.
- ⇝ The time complexity is compensated :

$$\mathscr{T} = 2^p(2^{|k_{in}|} + 2^{|k_{out}|} + 2^{|k_{in}|+|k_{out}|-|k_{in}\cap k_{out}|-n}).$$

The optimal number of bits conditions is given by the following bound :

$$\rightsquigarrow p + x \leq n - x \qquad \Longrightarrow \qquad x = \frac{n-p}{2}.$$

Finally, the Data complexity becomes : $\mathscr{D} = 2^{n-x}$.

Extensions of the attack in the original paper

Reducing Data complexity :

- $\rightsquigarrow$ Impose x bits conditions on the plaintexts P and P'.
- $\rightsquigarrow$ Useful in the case that the whole codebook is needed.
- $\rightsquigarrow$ The time complexity is compensated :

$$\mathscr{T} = 2^{p+x}(2^{|k_{in}|-x} + 2^{|k_{out}|-x} + 2^{|k_{in}|+|k_{out}|-|k_{in}\cap k_{out}|-(n-x)-2x}).$$

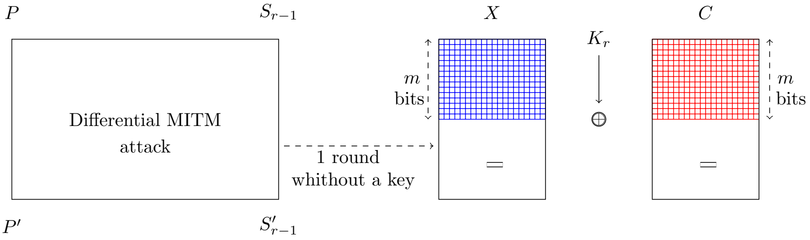
The optimal number of bits conditions is given by the following bound :

$$\rightsquigarrow p + x \leq n - x \qquad \Longrightarrow \qquad x = \frac{n-p}{2}.$$

Finally, the Data complexity becomes : $\mathscr{D} = 2^{n-x}$.

Extensions of the attack in the original paper

Parallel Partitions :



- ⇝ One round for free in the best case.
- ⇝ Round key addition applied on part of the cipher.

Our new results

Improvements of the differential MITM attack

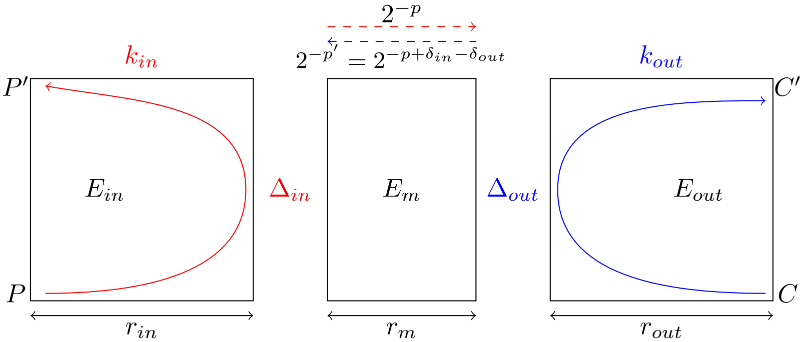
1. Extension to truncated differential MITM attack,
2. State-test technique,
3. Probability in the key recovery part,
4. Improved structures.

Applications of our improvements

1. 23 rounds of SKINNY-64-192,
2. 25 rounds of SKINNY-128-384,
3. 23 rounds out of 31 rounds of CRAFT.

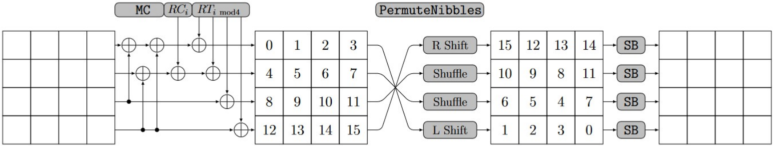
Truncated differential MITM

Instead of fixed differences δ_{in} and δ_{out}, we consider sets of differences Δ_{in} and Δ_{out}.



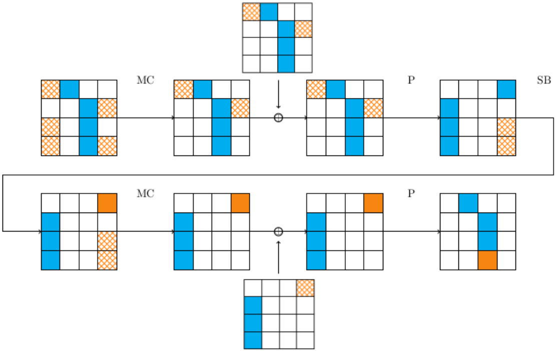
Description of CRAFT

CRAFT [BLMR19], published in ToSC in 2019, is a lightweight tweakable block cipher operating on a 64-bit block, a 128-bit key $(K_0||K_1)$, and a 64-bit tweak T.



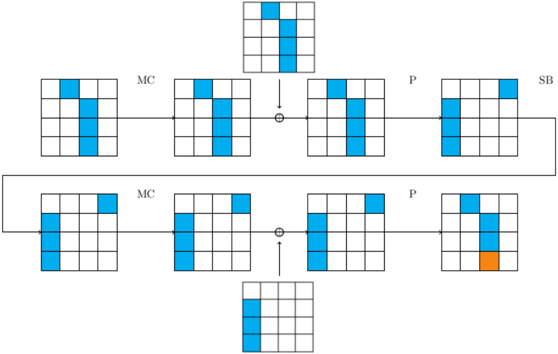
The last round is composed of only the *MixColumn*, *AddConstant* and *AddTweakey* operations.

1. State-test technique



- ⇝ Technique used previously in the context of the MITM and impossible differential attacks in [DSP07,BNS14],
- ⇝ Gives non-linear equations over the key bits,
- ⇝ Reduces the size of k_{in} and k_{out}.

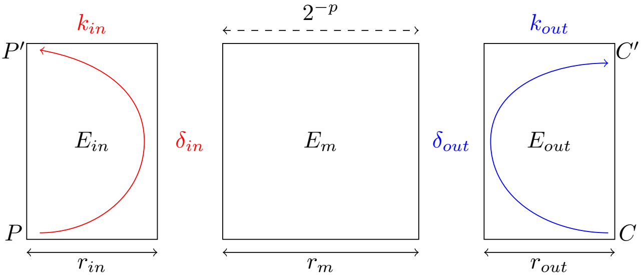
1. State-test technique



- ⇝ Technique used previously in the context of the MITM and impossible differential attacks in [DSP07,BNS14],
- ⇝ Gives non-linear equations over the key bits,
- ⇝ Reduces the size of k_{in} and k_{out}.

2. Probalistic key recovery

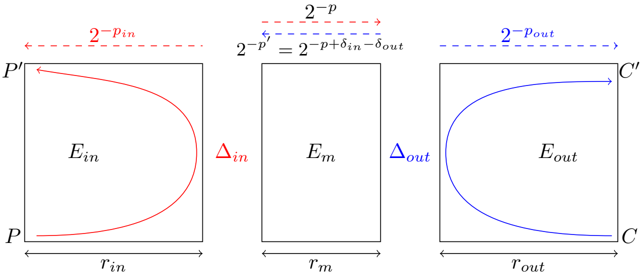
Classical case :



- ⇝ Propagate the differences with probability one.
- ⇝ Usually the whole state is active after a few rounds.

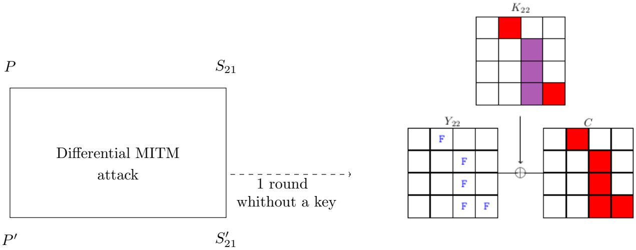
2. Probalistic key recovery

Probabilistic Key Recovery :



- ⇝ Probability of the overall attack become $2^{-p-p_{in}-p_{out}}$.
- ⇝ Higher Data is needed.
- ⇝ The number of candidate pairs of each side decreases by $2^{p_{in}}$ and $2^{p_{out}}$ respectively thus the time complexity does not increase.
- ⇝ Limits the propagation of the differences thus the size of k_{in} and k_{out} decreases.

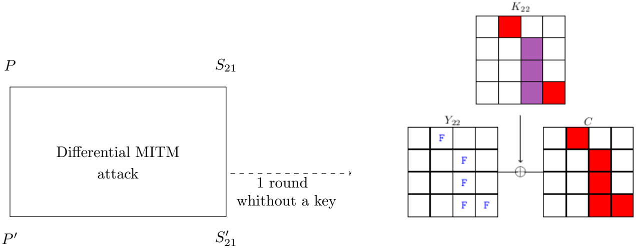
3. Example of the improved Parallel Partitioning



Extend the original parallel partitioning from [BBD+23]:

- ⇝ To 2 rounds for ciphers with partial-state round key addition (SKINNY).
- ⇝ To 1 round for ciphers with whole-state round key addition (CRAFT).

3. Example of the improved Parallel Partitioning



The ■ are known for both the upper and lower parts. The ■ are only known for the upper part.

- ⇝ Fix the 5 `F` nibbles in Y_{22}..
- ⇝ Compute the corresponding ■ in C.
- ⇝ For all the possible values of the non fixed words, compute the 2^{44} possibles Y_{22} and the 2^{44} possibles C.
- ⇝ Proceed to the upper (resp. lower) part of the attack from the structures of C (resp. Y_{22}).

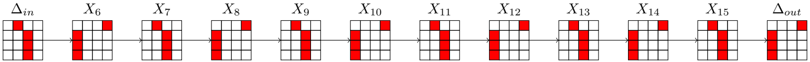
Results

Cipher	Rounds	Time	Data	Memory	Attack	Setting	Ref
CRAFT	19	$2^{114.68}$	2^{56}	2^{109}	DS-MITM	STK,CP	[MLC23]
	19	$2^{112.61}$	$2^{60.92}$	2^{72}	Rectangle	SK	[SZY+22]
	20	$2^{126.96}$	2^{56}	2^{109}	DS-MITM	STK,CP	[MLC23]
	21	$2^{106.53}$	$2^{60.99}$	2^{100}	ID	STK,CP	[HSE23]
	23	$2^{124.58}$	2^{60}	2^{101}	Tr-Diff-MITM	STK	New
SKINNY-64-192	23	2^{188}	2^{52}	2^{4}	MITM	STK	[DHS+21]
	23	2^{184}	2^{60}	2^{8}	MITM	STK	[BGST22]
	23	2^{188}	2^{28}	2^{4}	MITM	STK	[BGST22]
	23	2^{188}	2^{56}	2^{104}	Tr-Diff-MITM	STK	New
SKINNY-128-384	24	$2^{372.5}$	$2^{122.3}$	$2^{123.8}$	Diff-MITM	STK	[BDD+23]
	25	$2^{372.5}$	$2^{122.3}$	$2^{188.3}$	Diff-MITM	STK	[BDD+23]
	25	$2^{378.9}$	2^{117}	2^{165}	Diff-MITM	STK	New
	25	2^{366}	$2^{122.3}$	$2^{188.3}$	Diff-MITM	STK	New

Table: Summary of the best known cryptanalyses on `CRAFT`, `SKINNY-64-192` and `SKINNY-128-384` in the single-tweak setting.

Outline of the attack

- 23-round attack on CRAFT in the single-tweak setting
- We use the following truncated differential distinguisher over 11 rounds found via a MILP program.

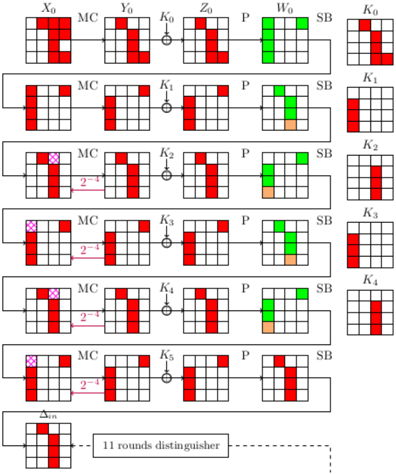


- Parameters :

p	p_{in}	p_{out}	s_{in}	s_{out}	δ_{in}	δ_{out}	$\|k_{in}\|$	$\|k_{out}\|$
44	16	12	16	12	16	16	32	32

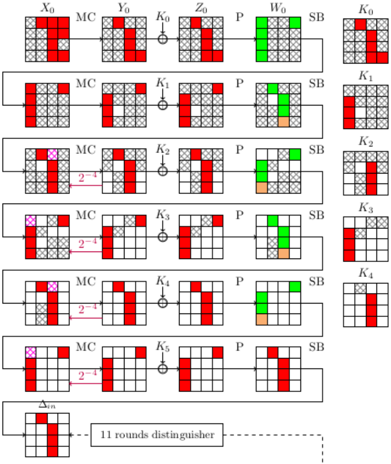
Upper part of the key recovery

- Truncated differential → No need to guess K_5.
- The ■ are the backward propagation of the differences.
- Value of ■ and ■ need to be known, ■ depends only on active bits but ■ also depends on non-active bits.
- In rounds 2,3,4,5 we pay a probability 2^{-4} to force the ⊠ to be non active.
- The ■ are the state test words that makes the value of ⊠ unnecessary to be determined.

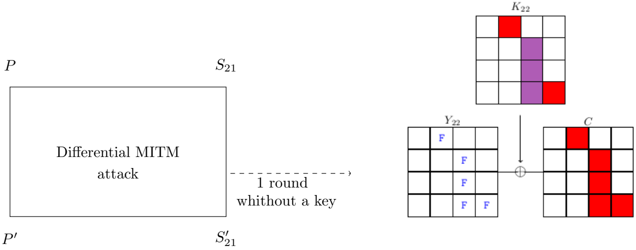


Upper part of the key recovery

- Truncated differential → No need to guess K_5.
- The ■ are the backward propagation of the differences.
- Value of ■ and ■ need to be known, ■ depends only on active bits but ■ also depends on non-active bits.
- In rounds 2,3,4,5 we pay a probability 2^{-4} to force the ⊠ to be non active.
- The ■ are the state test words that makes the value of ⊠ unnecessary to be determined.



Extension of one round



The ■ are known for both the upper and lower parts. The ■ are only known for the upper part.

- ⇝ Fix the 5 F nibbles in Y_{22}..
- ⇝ Compute the corresponding ■ in C.
- ⇝ For all the possible values of the non fixed words, compute the 2^{44} possibles Y_{22} and the 2^{44} possibles C.
- ⇝ Proceed to the upper (resp. lower) part of the attack from the structures of C (resp. Y_{22}).

Recovering the whole key

We have recovered 2^{108} candidates for 112 bits of information of the master key, including K_0 and 7 non-linear equations over the bits of K_1.

How to recover the rest of the key ?

First we can rewrite the equations given on rounds 4 and 18 as a function of 24 variables $x_1, \ldots, x_{24}$ which depend only on known information.

1. Store up to a table of size 2^s the solutions at a time.
2. Sort the table based on $x_1, \ldots, x_{24}$; we get 2^{96} groups of size $2^{s-96} = 2^y$ with the same solutions for equations 4 and 18.
3. For each candidate in each group, get the 2^{16} solutions for equations 3 and 19.
4. For each of the 2^{16} solutions we get only one match with solutions from equation 4 and 18.

Complexities

$\rightsquigarrow$ Time complexity to recover 2^{108} candidates for 112 bits of the master key

$$\begin{aligned} \mathscr{T} &= 2^{12} \times 2^{24}(2^{44} \times 2^{24} \times 2^{16-16} + 2^{44} \times 2^{20} \times 2^{16-12} + 2^{68+68-20-44}) \\ &= 2^{108}. \end{aligned}$$

$\rightsquigarrow$ Time complexity to recover the whole key

The time complexity to recover the whole key is finally

$$\mathscr{T} = 2^{108-s} 2^{96} \left(2^{20} + 2^{y} 2^{16}\right).$$

Complexities

⇝ Time complexity to recover 2^{108} candidates for 112 bits of the master key

$$\begin{aligned}\mathcal{T} &= 2^{12} \times 2^{24}(2^{44} \times 2^{24} \times 2^{16-16} + 2^{44} \times 2^{20} \times 2^{16-12} + 2^{68+68-20-44}) \\ &= 2^{108}.\end{aligned}$$

⇝ Time complexity to recover the whole key

The time complexity to recover the whole key is finally

$$\mathcal{T} = 2^{108-s} 2^{96} \left(2^{20} + 2^{y} 2^{16}\right).$$

⇝ Memory and data complexities $\mathcal{M} = 2^{s}$ to stock the list of solutions and $\mathcal{D} = 2^{60}$.

For $s = 101$:

$$\mathcal{T} = 2^{124.58},\ \mathcal{M} = 2^{101} \text{ and } \mathcal{D} = 2^{60}.$$

Conclusion

Conclusion

- $\rightsquigarrow$ New techniques which improve some best known attacks.
- $\rightsquigarrow$ Improved Differential MITM cryptanalysis.

Open questions and future works

- $\rightsquigarrow$ What are the limits of the state-test technique and the setting in which it is the most efficient.
- $\rightsquigarrow$ To automatise the parallel partitioning technique.

Conclusion

Conclusion

- ⇝ New techniques which improve some best known attacks.
- ⇝ Improved Differential MITM cryptanalysis.

Open questions and future works

- ⇝ What are the limits of the state-test technique and the setting in which it is the most efficient.
- ⇝ To automatise the parallel partitioning technique.

Thank you for your attention !