山东大学密码技术与信息安全教育部重点实验室
Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University

On the Usage of Deterministic (Related-Key) Truncated Differentials and Multidimensional Linear Approximations for SPN Ciphers

Ling Sun[1], David Gerault[2], Wei Wang[1], Meiqin Wang[1](✉)

1. Shandong University, Jinan & Qingdao, China
2. Nanyang Technological University, Singapore

FSE 2020 @ November, 2020

SHANDONG UNIVERSITY

Outline

Background & Contributions

Automatic Search

- Automatic tools for cryptanalysis obtained rapid development.
- Few works concentrated on the deterministic TD/MDLA.

Essential Problems

- The optimality of TD/MDLA must be confirmed via an exhaustive search.
- The incomplete search is also a long-term problem for optimal ID/ZCLA.

Contributions

- An automatic tool for the search of deterministic (RK) TDs and MDLAs.
- Improved related-key differential-linear attack on AES-192.
- Constructing (RK) IDs with TDs and ZCLAs with MDLAs.
  - Provable security against ID attack of `SKINNY` and `Midori64`.

Outline

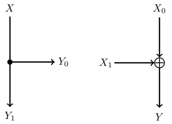
Preliminaries

Basics of Differential and Linear Cryptanalyses

- The **difference** of the state $\Delta X = (\Delta X_0, \Delta X_1, \ldots, \Delta X_{\ell-1})$, $\Delta X_i \in \mathbb{F}_{2^s}$.
- The **differential pattern** $\Delta_X = (\Delta_{X_0}, \Delta_{X_1}, \ldots, \Delta_{X_{\ell-1}})$.
  - **zero** differential pattern (Z).
  - **nonzero fixed** differential pattern (N).
  - **nonzero varied** differential pattern (N*).
  - **varied** differential pattern (U).

Lemma 1 (Branching)

$$\Delta_{Y_0} = \Delta_{Y_1} = \Delta_X.$$



Lemma 2 (XOR)

$$(\Delta_{X_0}, \Delta_{X_1}) \rightarrow \Delta_Y.$$

Δ_Y		Δ_{X_1}				
		Z	N	N $\oplus$ N*	N*	U
Δ_{X_0}	Z	Z	N	N $\oplus$ N*	N*	U
	N	N	Z/N	N*/N $\oplus$ N*	N $\oplus$ N*	U
	N $\oplus$ N*	N $\oplus$ N*	N*/N $\oplus$ N*	U	U	U
	N*	N*	N $\oplus$ N*	U	U	U
	U	U	U	U	U	U

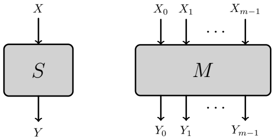
Preliminaries

Basics of Differential and Linear Cryptanalyses

Lemma 3 (S-box)

$$\Delta_X \to \Delta_Y .$$

Δ_X	Z	N	N $\oplus$ N*	N*	U
Δ_Y	Z	N*	U	N*	U



Lemma 4 (MDS matrix)

$$\Delta_X \to \Delta_Y .$$

Δ_X	(Z, Z, …, Z)	(Z, …, Z, N/N*, Z, …, Z)	Remaining cases
Δ_Y	(Z, Z, …, Z)	(N*, N*, …, N*)	(U, U, …, U)

- The **linear mask** of the state $\Gamma X = (\Gamma X_0, \Gamma X_1, \ldots, \Gamma X_{\ell-1})$, $\Gamma X_i \in \mathbb{F}_{2^s}$.
- The **linear pattern** $\Gamma_X = (\Gamma_{X_0}, \Gamma_{X_1}, \ldots, \Gamma_{X_{\ell-1}})$.
  - **zero** linear pattern (Z).
  - **nonzero fixed** linear pattern (N).
  - **nonzero varied** linear pattern (N*).
  - **varied** linear pattern (U).

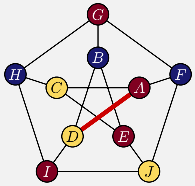
Preliminaries

Constraint Satisfaction Problem

Definition 1 (Constraint satisfaction problem @ SGL+17)

A **constraint satisfaction problem (CSP)** is represented as a triple $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$.

- $\mathcal{X} = \{x_0, x_1, \ldots, x_{n-1}\}$ is a set of variables.
- $\mathcal{D} = \{\mathcal{D}(x_0), \mathcal{D}(x_1), \ldots, \mathcal{D}(x_{n-1})\}$ is a set of nonempty sets.
- $\mathcal{C} = \{\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{m-1}\}$ stands for a set of constraints.



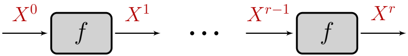
- $\mathcal{X} = \{A, B, \ldots, J\}$.
- $\mathcal{D} = \{\mathcal{D}(A), \mathcal{D}(B), \ldots, \mathcal{D}(J)\}$.
  - $\mathcal{D}(\cdot) = \{$"red", "yellow", "blue"$\}$.
- $\mathcal{C} = \{\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{14}\}, \mathcal{C}_* = \langle \mathcal{X}_*, \mathcal{R}_* \rangle$.
  - $\mathcal{C}_* = \langle \{A, D\}, A \neq D \rangle$.

- SAT/SMT problems can be viewed as individual cases of the CSP.
- The CSP can describe much harder cases.
- Many CP solvers are available to solve problems of practical interest.

Outline

Finding Deterministic TDs and MDLAs

Step 1: Initialising Variables



- δ_{X_i}: pattern Δ_{X_i}.

$$\delta_{X_i} = \begin{cases} 0, & \text{if } \Delta_{X_i} = \mathsf{Z} \\ 1, & \text{if } \Delta_{X_i} = \mathsf{N} \\ 2, & \text{if } \Delta_{X_i} = \mathsf{N}^* \\ 3, & \text{if } \Delta_{X_i} = \mathsf{U} \end{cases}.$$

- ζ_{X_i}: s-bit difference ΔX_i.

$$\zeta_{X_i} \in \begin{cases} \{0\}, & \text{if } \delta_{X_i} = 0 \\ \{1, 2, \ldots, 2^s - 1\}, & \text{if } \delta_{X_i} = 1 \\ \{-1\}, & \text{if } \delta_{X_i} = 2 \\ \{-2\}, & \text{if } \delta_{X_i} = 3 \end{cases}.$$

Model 1 (Relation between δ_{X_i} and ζ_{X_i})

The following expression will ensure that ζ_{X_i} falls into the correct range.

$$\texttt{if } \delta_{X_i} = 0 \texttt{ then } \zeta_{X_i} = 0$$
$$\texttt{elseif } \delta_{X_i} = 1 \texttt{ then } \zeta_{X_i} > 0$$
$$\texttt{elseif } \delta_{X_i} = 2 \texttt{ then } \zeta_{X_i} = -1$$
$$\texttt{else } \zeta_{X_i} = -2 \texttt{ endif}$$

Finding Deterministic TDs and MDLAs

Step 2: Propagating Differential Patterns

Model 2 (Branching)

The constraint restricts the pattern propagation for the Branching operation.

$$\delta_{Y_0} = \delta_X \text{ and } \zeta_{Y_0} = \zeta_X \text{ and } \delta_{Y_1} = \delta_X \text{ and } \zeta_{Y_1} = \zeta_X$$

Model 3 (XOR)

The constraint restricts the pattern propagation for the XOR operation.

$$
\begin{aligned}
&\texttt{if } \delta_{X_0} + \delta_{X_1} > 2 \texttt{ then } \delta_Y = 3 \texttt{ and } \zeta_Y = -2 \\
&\texttt{elseif } \delta_{X_0} + \delta_{X_1} = 1 \texttt{ then } \delta_Y = 1 \texttt{ and } \zeta_Y = \zeta_{X_0} + \zeta_{X_1} \\
&\texttt{elseif } \delta_{X_0} = \delta_{X_1} = 0 \texttt{ then } \delta_Y = 0 \texttt{ and } \zeta_Y = 0 \\
&\texttt{elseif } \zeta_{X_0} + \zeta_{X_1} < 0 \texttt{ then } \delta_Y = 2 \texttt{ and } \zeta_Y = -1 \\
&\texttt{elseif } \zeta_{X_0} = \zeta_{X_1} \texttt{ then } \delta_Y = 0 \texttt{ and } \zeta_Y = 0 \\
&\texttt{else } \delta_Y = 1 \texttt{ and } \zeta_Y = \zeta_{X_0} \oplus \zeta_{X_1} \texttt{ endif}
\end{aligned}
$$

Finding Deterministic TDs and MDLAs

Step 2: Propagating Differential Patterns

Model 4 (S-box)

The constraint restricts the pattern propagation for the S-box.

$$\delta_Y \neq 1 \text{ and } \delta_X + \delta_Y \in \{0, 3, 4, 6\} \text{ and } \delta_Y \geqslant \delta_X \text{ and } \delta_Y - \delta_X \leqslant 1$$

Model 5 (MDS matrix)

The constraint restricts the pattern propagation for the MDS matrix.

$$\texttt{if} \sum_{i=0}^{m-1} \delta_{X_i} \equiv 0 \texttt{ then } \delta_{Y_0} = \delta_{Y_1} = \cdots = \delta_{Y_{m-1}} = 0$$

$$\texttt{elseif} \sum_{i=0}^{m-1} \delta_{X_i} \equiv 1 \texttt{ then } \delta_{Y_0} = \delta_{Y_1} = \cdots = \delta_{Y_{m-1}} = 2$$

$$\texttt{elseif} \sum_{i=0}^{m-1} \delta_{X_i} \equiv 2 \texttt{ and } \sum_{i=0}^{m-1} \zeta_{X_i} < 0 \texttt{ then } \delta_{Y_0} = \delta_{Y_1} = \cdots = \delta_{Y_{m-1}} = 2$$

$$\texttt{else } \delta_{Y_0} = \delta_{Y_1} = \cdots = \delta_{Y_{m-1}} = 3 \texttt{ endif}$$

Finding Deterministic TDs and MDLAs

Step 3: Clarifying the Searching Scopes of the Input Patterns

Old-fashion

- **Fix** the input pattern as a predetermined value.
- The optimal TD requests an **exhaustive search** over all possible patterns.
- The program should be implemented for about 2^ℓ times.

New-fashion

- Do **not fix** the format of the input pattern.
- Denote $(X_0^0, X_1^0, \ldots, X_{\ell-1}^0)$ the input state. Add $\sum_{i=0}^{\ell-1} \delta_{X_i^0} \neq 0$.
- The CP solver will **automatically traverse** all possible input patterns.
- To ensure the existence of R-round TDs/MDLAs, at most, we invoke the searching program for $3 \cdot R \cdot \ell$ times.

- The number of runs to search for the optimal ID of Minalpher-P is **reduced** from 2^{128} to $2^{10.9}$.

Finding Deterministic TDs and MDLAs

Step 4: Clarifying the Searching Scopes of the Output Patterns

- The output differential patterns we are interested in are Z, N and N*.
  - ΔX_i^r being zero corresponds to $\delta_{X_i^r} = 0$.
  - ΔX_i^r being nonzero and fixed corresponds to $\delta_{X_i^r} = 1$.
  - ΔX_i^r being any value except zero corresponds to $\delta_{X_i^r} = 2$.

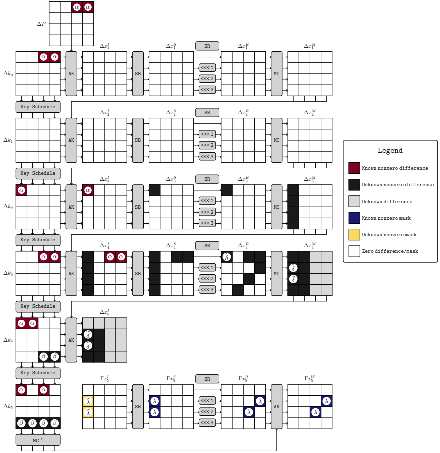
Generalisation

- The method for the search of TDs can be adjusted to search for **MDLAs**.
- For ciphers with word-oriented key schedules, this method can be applied to search for **related-key truncated differentials**.

Outline

Related-Key Differential-Linear Attack on AES-192

Improved RK DL Attack on AES-192



Previous distinguishing property

$$\lambda \cdot (\Delta x_5^W[1,3] \oplus \Delta x_5^W[2,2]) = 0$$

- The bias is about 2^{-9}.

New distinguishing property

$$\lambda \cdot \Delta x_5^W[1,3] = 0$$

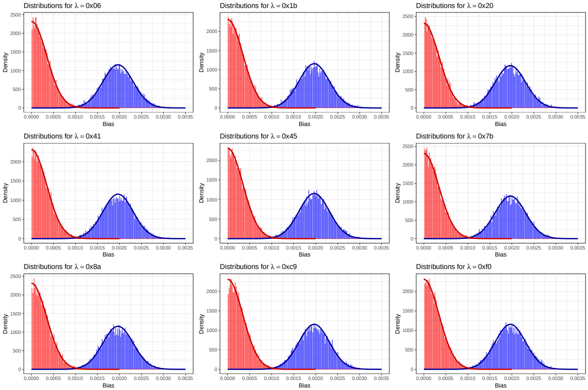
- The bias is about $2^{-8.99}$.

- The biases are almost the same.
- The complexity of the distinguishing attack basically remains unchanged.
- The complexity of the key-recovery attack drops.

Related-Key Differential-Linear Attack on AES-192

Improved RK DL Attack on AES-192

- Given N pairs of plaintexts, Σ records the number of good pairs.
- For the real cipher, $|\Sigma/N - 0.5|$ follows the distribution $\mathcal{N}(\varepsilon, 1/4N)$.
- Otherwise, $|\Sigma/N - 0.5|$ follows the distribution $\overline{\mathcal{N}}(0, 1/4N)$.



- The key-recovery attack requires $2^{21.3}$ chosen plaintexts.
- The time complexity is reduced from 2^{187} to $2^{170.5}$.

Outline

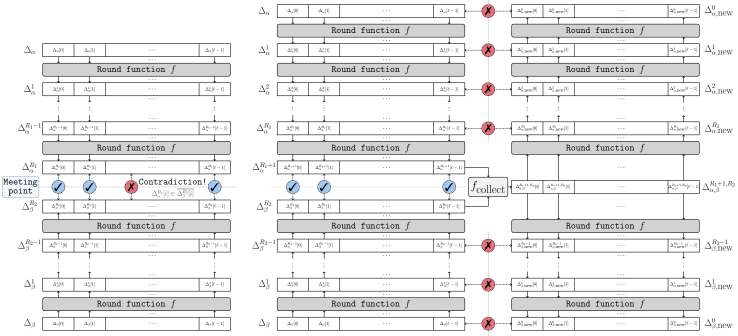
Miss-in-the-Middle approach

- Constructing two TDs $\Delta^{I_1} \xrightarrow[R_1\text{-round}]{} \Delta^{O_1}$ and $\Delta^{O_2} \xleftarrow[R_2\text{-round}]{} \Delta^{I_2}$.
- Checking the compatibility of the two output patterns Δ^{O_1} and Δ^{O_2}.

Distinctions between $\mathcal{U}$-method and our $\mathcal{U}^*$-method

- The way to implement the search.
- The set of differential patterns applied to yield contradictions.
  - The $\mathcal{U}$-method considers the set $\mathcal{U} = \{\mathrm{Z}, \mathrm{N}, \mathrm{N} \oplus \mathrm{N}^*, \mathrm{N}^*\}$.
  - The $\mathcal{U}^*$-method takes the smaller set $\mathcal{U}^* = \{\mathrm{Z}, \mathrm{N}, \mathrm{N}^*\}$.
- The searching scopes of the input and output patterns.

- Regarding SPN ciphers
  - The $\mathcal{U}^*$-method has almost the same performance as the $\mathcal{U}$-method.



(a) Type-I contradiction.

(b) Type-II contradiction.

Definition 2 (Message collecting function)

The message collecting function f_{collect} is a function over two differential patterns Δ_X and Δ_Y with $\Delta_Y \notin \overline{\Delta_X}$. The output $f_{\text{collect}}(\Delta_X, \Delta_Y)$ is a pattern that unifies information of two compatible differential patterns.

Comparison of All Tools Targeting (RK) IDs of SPN Ciphers

Method	Properties						
	P1: 1-property	P2: DDT	P3: truncated	P4: 8-bit S-box	P5: fixed	P6: exhaustive	P7: RK ID
$\mathcal{U}$-method	★		★	★	★	⯪	
UID-method			★	★	★	⯪	
Wu and Wang			★	★	★	⯪	
Sasaki and Todo		★	★	★	★	⯪	
Sun et al.		★			★		★
(Optimised) $\mathcal{U}^*$-method			★	★		★	★

- The source codes can be found at `https://github.com/Deterministic-TD-MDLA/auxiliary_material`.
- One processor Intel® Xeon® Gold 5118 CPU @ 2.30GHz.
- For SKINNY and Midori64, all programs finish in several seconds.
- For Minalpher-P, it takes several minutes to return the result.

Outline

Applications to SKINNY

Main results

- 12.5-round impossible differentials with the optimised $\mathcal{U}^*$-method.
- New 12.5-round related-tweakey impossible differentials for SKINNY-n-n.
- 11.5-round zero-correlation linear approximations.

Theorem 1 (Provable security of SKINNY against ID distinguishing attack)

Under the keyed (uniform) bijective S-box assumption, 13.5-round encryption of SKINNY is secure against impossible differentials with arbitrary nonzero input and output differences.

Theorem 2 (Provable security of SKINNY-n-n against RT IDs)

13.5-round SKINNY-n-n is secure against related-tweakey impossible differentials with arbitrary nonzero input and output differences under the following assumptions:

- the S-box satisfies keyed (uniform) bijective assumption;
- the difference of tweakey only has one active cell.

Applications to Midori64 and Minalpher-P

Main results

- 480 6.5-round impossible differentials for Midori64.
- 600 8.5-round impossible differentials for Minalpher-P.

Theorem 3 (Provable security of Midori64 against ID distinguishing attack)

Under the keyed (uniform) bijective S-box assumption, 7.5-round Midori64 is secure against impossible differentials with arbitrary nonzero input and output differences.

Outline

Conclusion

- An automatic tool for the search of deterministic (RK) TDs and MDLAs.
- Improved related-key differential-linear attack on AES-192.
- Constructing (RK) IDs with TDs and ZCLAs with MDLAs.
  - Provable security against ID attack of `SKINNY` and `Midori64`.

Discussion

- The centre of the paper is more the new technique.
- The tool may play an essential role in the designing phase of new ciphers.
- Constructing a unified framework involving the key-recovery approach.

Thank you for your attention!

Thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.