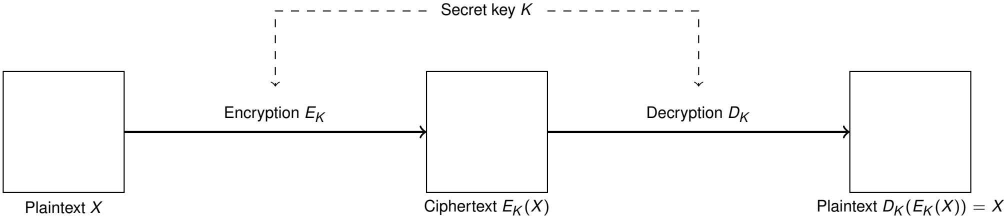
From ModRef 2014 to ModRef 2024: Ten years of CP models for solving differential cryptanalysis problems

Christine Solnon, CITI, INSA Lyon / INRIA

Collaboration with F. Delobel, P. Derbez, D. Gerault, A. Gontier, P. Lafourcade, L. Libralesso, M. Minier, C. Prud'homme, L. Rouquette

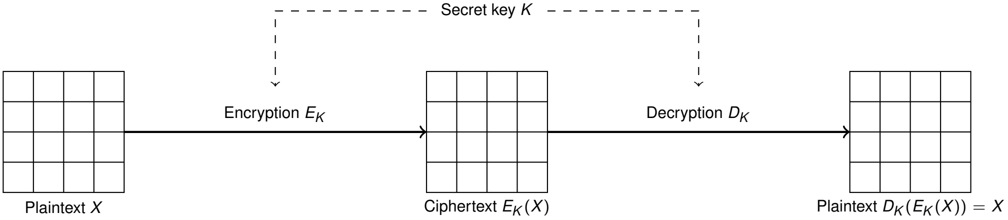
From ModRef 2014 to ModRef 2024

Symmetric Ciphers



- Same secret key K used for encryption and decryption
  $\rightsquigarrow D_K = E_K^{-1}$
- Plaintext and ciphertext are split into blocks
  $\rightsquigarrow$ Typically: 1 block = 4×4 bytes = 128 bits

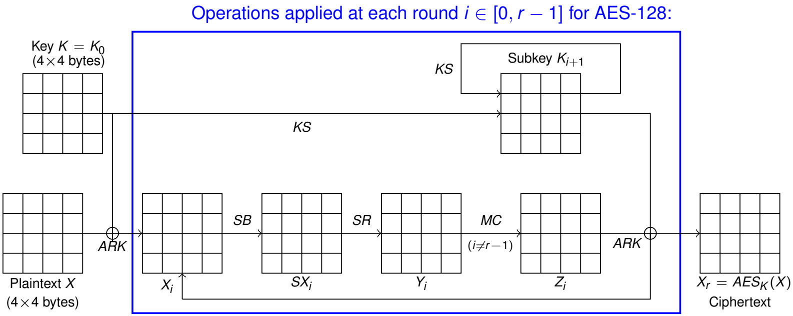
Symmetric Block Ciphers



- Same secret key K used for encryption and decryption
  $\rightsquigarrow D_K = E_K^{-1}$
- Plaintext and ciphertext are split into blocks
  $\rightsquigarrow$ Typically: 1 block = 4×4 bytes = 128 bits

AES-128: Advanced Encryption Standard with 128-bit keys

⇝ Standard block cipher since 2001



Initialization:

- $X_0 = ARK(X, K)$
- $K_0 = K$

For each round $i \in [0, r-1]$:

- $SX_i = SB(X_i)$
- $Y_i = SR(SX_i)$
- $Z_i = MC(Y_i)$
- $X_{i+1} = ARK(Z_i, K_{i+1})$
  with $K_{i+1} = KS(K_i)$

Return X_r

Cryptanalysis

Goal: Analyse ciphers to detect weaknesses

Confidentiality: Is it possible to retrieve the plaintext (under some given attack conditions)?

This must be done for each new cipher...

...and new ciphers are designed every year!

Examples of symmetric block ciphers:

AES, Craft, Deoxys, Gift, Midori, Present, Skinny, Simon, Speck, ...

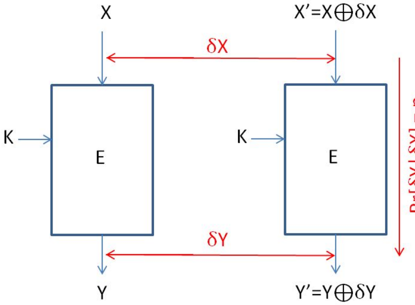
Differential Cryptanalysis [BS91]

How to inject differences with eXclusive OR (XOR)?

- Notation: $\oplus$ = XOR operator (i.e., $0 \oplus 0 = 1 \oplus 1 = 0$ and $0 \oplus 1 = 1 \oplus 0 = 1$)
  $\rightsquigarrow$ Extended to bitstrings (e.g., $00110 \oplus 01101 = 01011$)
- To inject a difference at bit k of bitstring M, XOR M with bitstring with only one '1' at position k

Differential cryptanalysis exploits differences to recover the key:

- Let $\delta X = X \oplus X'$ be an input plaintext difference
- Let $\delta Y = E_K(X) \oplus E_K(X')$ be the output difference
- The cipher is weak if $\exists\ \delta X$ and δY such that $Pr[\delta Y|\delta X] >> 2^{-|K|}$
  $\rightsquigarrow$ Key recovery in $\mathcal{O}(1/Pr[\delta Y|\delta X])$

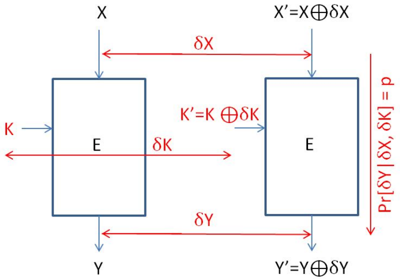


[BS91] E. Biham and A. Shamir: *Differential cryptoanalysis of feal and n-hash*. In EUROCRYPT 1991

Related-Key Attack [Bih93]

Inject differences in texts and keys:

- Let $\delta X = X \oplus X'$ be an input plaintext difference
- Let $\delta K = K \oplus K'$ be an input key difference
- Let $\delta Y = E_K(X) \oplus E_{K'}(X')$ be the output difference
- The cipher is weak if $\exists\ \delta X, \delta K$, and δY such that $Pr[\delta Y|\delta X, \delta K] >> 2^{-|K|}$
  $\rightsquigarrow$ Key recovery in $\mathcal{O}(1/Pr[\delta Y|\delta X, \delta K])$

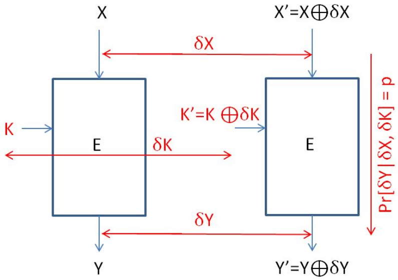


[Bih93] E. Biham: *New types of cryptoanalytic attacks using related keys*. In EUROCRYPT 1993

Related-Key Attack [Bih93]

Inject differences in texts and keys:

- Let $\delta X = X \oplus X'$ be an input plaintext difference
- Let $\delta K = K \oplus K'$ be an input key difference
- Let $\delta Y = E_K(X) \oplus E_{K'}(X')$ be the output difference
- The cipher is weak if $\exists\ \delta X, \delta K$, and δY such that $Pr[\delta Y | \delta X, \delta K] >> 2^{-|K|}$
  $\rightsquigarrow$ Key recovery in $\mathcal{O}(1/Pr[\delta Y | \delta X, \delta K])$



Differential Characteristic:

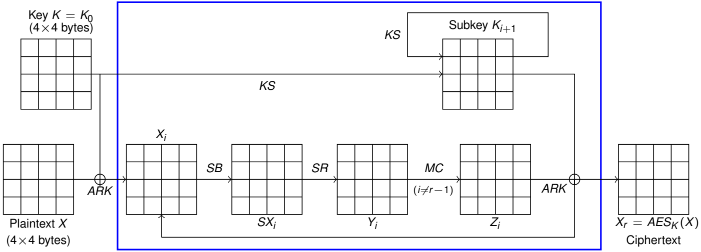
Plaintext and key differences for each round of the ciphering process

Goal:

Compute a differential characteristic the probability of which is maximal

[Bih93] E. Biham: *New types of cryptoanalytic attacks using related keys*. In EUROCRYPT 1993

Example: Differential Characteristic for AES-128

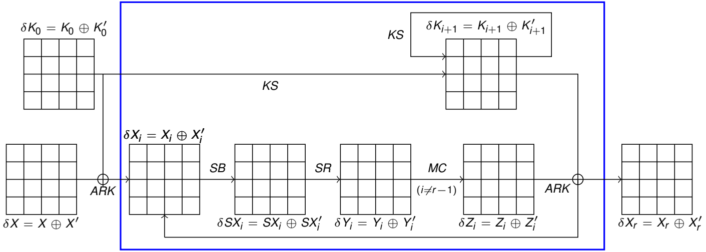


Notations for bytes (during ciphering):

- $K_{i,j,k}$ = byte at column j and row k of subkey at round i
- $X_{i,j,k}$ = byte at column j and row k of text at round i
- Same for $SX_{i,j,k}$, $Y_{i,j,k}$, ...

$\rightsquigarrow$ Every byte has a value in $[0, 255]$

Example: Differential Characteristic for AES-128



Notations for bytes (during ciphering):

- $K_{i,j,k}$ = byte at column j and row k of subkey at round i
- $X_{i,j,k}$ = byte at column j and row k of text at round i
- Same for $SX_{i,j,k}$, $Y_{i,j,k}$, ...

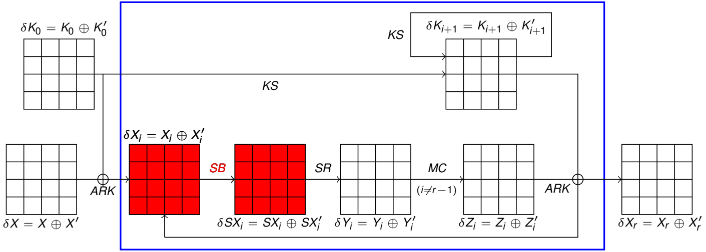
$\rightsquigarrow$ Every byte has a value in $[0, 255]$

Notations for differential bytes (in differential characteristics):

- $\delta K_{i,j,k} = K_{i,j,k} \oplus K'_{i,j,k}$
- $\delta X_{i,j,k} = X_{i,j,k} \oplus X'_{i,j,k}$
- Same for $\delta SX_{i,j,k}$, $\delta Y_{i,j,k}$, ...

$\rightsquigarrow$ Every differential byte has a value in $[0, 255]$

Example: Differential Characteristic for AES-128



SB operator for ciphering:

$$SX_{i,j,k} = s(X_{i,j,k})$$

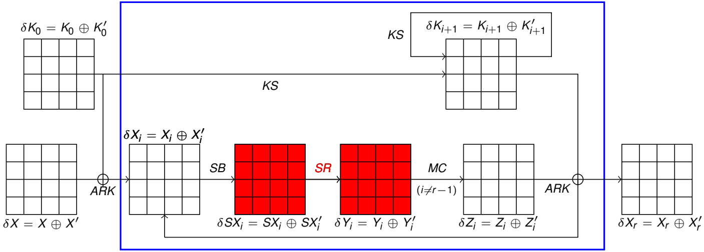
where $s : [0, 255] \rightarrow [0, 255]$ is a bijection defined by a look-up table

SB constraint for differential characteristic:

$$(\delta X_{i,j,k}, \delta SX_{i,j,k}) \in T_{sbox}$$

where $T_{sbox} = \{(a \oplus a', s(a) \oplus s(a')) \mid a, a' \in [0, 255]\}$

Example: Differential Characteristic for AES-128



SR operator for ciphering:

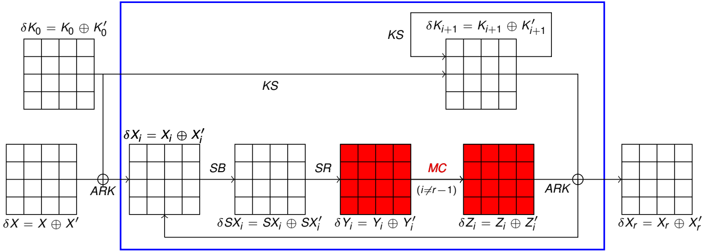
$$Y_{i,j,k} = SX_{i,j,(k+j)\%4}$$

⇝ Simple byte shifting

SR constraint for differential characteristic:

$$\delta Y_{i,j,k} = \delta SX_{i,j,(k+j)\%4}$$

Example: Differential Characteristic for AES-128



MC operator for ciphering:

$$
\begin{aligned}
Z_{i,j,k} &= M_{j,0} \otimes Y_{i,0,k} \\
&\oplus M_{j,1} \otimes Y_{i,1,k} \\
&\oplus M_{j,2} \otimes Y_{i,2,k} \\
&\oplus M_{j,3} \otimes Y_{i,3,k}
\end{aligned}
$$

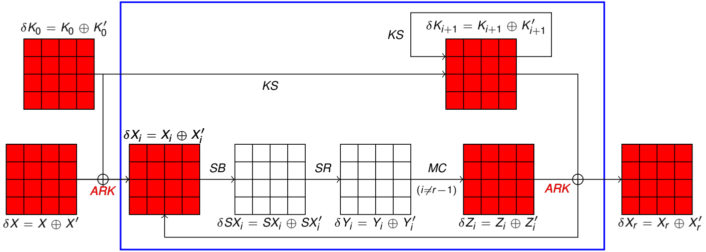
Where M is a given 4×4 matrix, and $\otimes$ is a finite field multiplication operator

MC constraint for differential characteristic:

$$
\begin{aligned}
\delta Z_{i,j,k} &= M_{j,0} \otimes \delta Y_{i,0,k} \\
&\oplus M_{j,1} \otimes \delta Y_{i,1,k} \\
&\oplus M_{j,2} \otimes \delta Y_{i,2,k} \\
&\oplus M_{j,3} \otimes \delta Y_{i,3,k}
\end{aligned}
$$

Because $(a \otimes b) \oplus (a \otimes b') = a \otimes (b \oplus b')$

Example: Differential Characteristic for AES-128



ARK operator for ciphering:

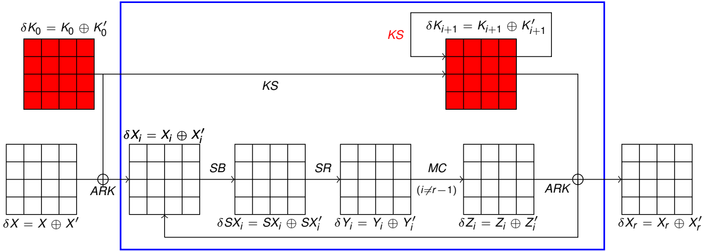
- $X_{0,j,k} = K_{0,j,k} \oplus X_{j,k}$
- $X_{i+1,j,k} = K_{i+1,j,k} \oplus Z_{i,j,k}$

ARK constraint for differential characteristic:

- $\delta X_{0,j,k} = \delta K_{0,j,k} \oplus \delta X_{j,k}$
- $\delta X_{i+1,j,k} = \delta K_{i,j,k} \oplus \delta Z_{i,j,k}$

because $(a \oplus b) \oplus (a' \oplus b') = (a \oplus a') \oplus (b \oplus b')$

Example: Differential Characteristic for AES-128



KS operator for ciphering:

- Row 0:
  $K_{i+1,j,0} = SK_{i,(j+1)\%4,3} \oplus K_{i,j,0}$
  where $SK_{i,j,3} = s(K_{i,j,3})$
- Row $k > 0$:
  $K_{i+1,j,k} = K_{i+1,j,k-1} \oplus K_{i,j,k}$

KS constraint for differential characteristic:

- Row 0:
  $\delta K_{i+1,j,0} = \delta SK_{i,(j+1)\%4,3} \oplus \delta K_{i,j,0}$
  where $(\delta K_{i,j,3}, \delta SK_{i,j,3}) \in T_{sbox}$
- Row $k > 0$: $\delta K_{i+1,j,k} = \delta K_{i+1,j,k-1} \oplus \delta K_{i,j,k}$

Full model for computing differential characteristics for AES-128

- SB: $\forall i \in [0, r-1], \forall j, k \in [0,3], (\delta X_{i,j,k}, \delta SX_{i,j,k}) \in T_{sbox}$
- SR: $\forall i \in [0, r-1], \forall j, k \in [0,3], \delta Y_{i,j,k} = \delta SX_{i,j,(k+j)\%4}$
- MC:
  $\forall i \in [0, r-2], \forall j, k \in [0,3], \delta Z_{i,j,k} = M_{j,0} \otimes \delta Y_{i,0,k} \oplus M_{j,1} \otimes \delta Y_{i,1,k} \oplus M_{j,2} \otimes \delta Y_{i,2,k} \oplus M_{j,3} \otimes \delta Y_{i,3,k}$
- ARK:
  - $\forall j, k \in [0,3], \delta X_{0,j,k} = \delta K_{0,j,k} \oplus \delta X_{j,k}$
  - $\forall i \in [0, r-1], \forall j, k \in [0,3], \delta X_{i+1,j,k} = \delta K_{i,j,k} \oplus \delta Z_{i,j,k}$
- SK:
  - $\forall i \in [0, r-1], \forall j \in [0,3], \delta K_{i+1,j,0} = \delta SK_{i,(j+1)\%4,3} \oplus \delta K_{i,j,0}$
  - $\forall i \in [0, r-1], \forall j \in [0,3], (\delta K_{i,j,3}, \delta SK_{i,j,3}) \in T_{sbox}$
  - $\forall i \in [0, r-1], \forall j \in [0,3], \forall k \in [1,3], \delta K_{i+1,j,k} = \delta K_{i+1,j,k-1} \oplus \delta K_{i,j,k}$

How to transform this model into a CP model?

- Introduce a table for the ternary XOR relation: $T_{\oplus} = \{(a, b, a \oplus b) \mid a, b \in [0, 255]\}$
- Decompose MC into relations of smaller arity

Full model for computing differential characteristics for AES-128

- SB: $\forall i \in [0, r-1], \forall j, k \in [0,3], (\delta X_{i,j,k}, \delta SX_{i,j,k}) \in T_{sbox}$
- SR: $\forall i \in [0, r-1], \forall j, k \in [0,3], \delta Y_{i,j,k} = \delta SX_{i,j,(k+j)\%4}$
- MC:
  $\forall i \in [0, r-2], \forall j, k \in [0,3], \delta Z_{i,j,k} = M_{j,0} \otimes \delta Y_{i,0,k} \oplus M_{j,1} \otimes \delta Y_{i,1,k} \oplus M_{j,2} \otimes \delta Y_{i,2,k} \oplus M_{j,3} \otimes \delta Y_{i,3,k}$
- ARK:
  - $\forall j, k \in [0,3], \delta X_{0,j,k} = \delta K_{0,j,k} \oplus \delta X_{j,k}$
  - $\forall i \in [0, r-1], \forall j, k \in [0,3], \delta X_{i+1,j,k} = \delta K_{i,j,k} \oplus \delta Z_{i,j,k}$
- SK:
  - $\forall i \in [0, r-1], \forall j \in [0,3], \delta K_{i+1,j,0} = \delta SK_{i,(j+1)\%4,3} \oplus \delta K_{i,j,0}$
  - $\forall i \in [0, r-1], \forall j \in [0,3], (\delta K_{i,j,3}, \delta SK_{i,j,3}) \in T_{sbox}$
  - $\forall i \in [0, r-1], \forall j \in [0,3], \forall k \in [1,3], \delta K_{i+1,j,k} = \delta K_{i+1,j,k-1} \oplus \delta K_{i,j,k}$

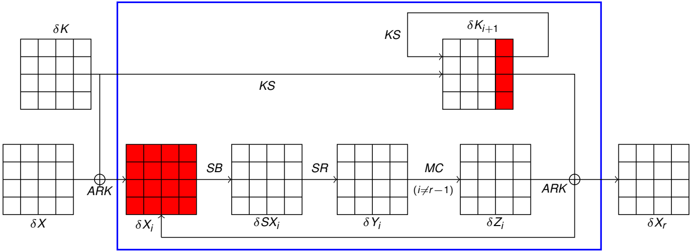
How to transform this model into a CP model?

- Introduce a table for the ternary XOR relation: $T_{\oplus} = \{(a, b, a \oplus b) \mid a, b \in [0, 255]\}$
- Decompose MC into relations of smaller arity

CP model for computing differential characteristics for AES-128

- SB: $\forall i \in [0, r-1], \forall j, k \in [0, 3], (\delta X_{i,j,k}, \delta SX_{i,j,k}) \in T_{sbox}$
- SR: $\forall i \in [0, r-1], \forall j, k \in [0, 3], \delta Y_{i,j,k} = \delta SX_{i,j,(k+j)\%4}$
- MC: $\forall i \in [0, r-2], \forall j, k \in [0, 3]$,
  - $(\delta Y_{i,x,k}, A_x) \in T_x$ where $T_x = \{(y, y \otimes M_x) \mid y \in [0, 255]\} \quad \forall x \in \{(j, 0), (j, 1), (j, 2), (j, 3)\}$
  - $(A_{j,0}, A_{j,1}, B) \in T_\oplus$
  - $(A_{j,2}, A_{j,3}, C) \in T_\oplus$
  - $(B, C, \delta Z_{i,j,k}) \in T_\oplus$
- ARK:
  - $\forall j, k \in [0, 3], (\delta X_{0,j,k}, \delta K_{0,j,k}, \delta X_{j,k}) \in T_\oplus$
  - $\forall i \in [0, r-1], \forall j, k \in [0, 3], (\delta X_{i+1,j,k}, \delta K_{i,j,k}, \delta Z_{i,j,k}) \in T_\oplus$
- SK:
  - $\forall i \in [0, r-1], \forall j \in [0, 3], (\delta K_{i+1,j,0}, \delta SK_{i,(j+1)\%4,3}, \delta K_{i,j,0}) \in T_\oplus$
  - $\forall i \in [0, r-1], \forall j \in [0, 3], (\delta K_{i,j,3}, \delta SK_{i,j,3}) \in T_{sbox}$
  - $\forall i \in [0, r-1], \forall j \in [0, 3], \forall k \in [1, 3], (\delta K_{i+1,j,k}, \delta K_{i+1,j,k-1}, \delta K_{i,j,k}) \in T_\oplus$

Probability of a differential characteristic



- ARK, SR, MC: output differences are computed from input differences with probability 1
- SB: probability of observing an output difference δ_{out} given an input difference δ_{in}
  - When $\delta_{in} = \delta_{out} = 0$: $p(\delta_{out}|\delta_{in}) = 1$
  - Otherwise: $p(\delta_{out}|\delta_{in}) \in \{0, 2^{-7}, 2^{-6}\}$

$\rightsquigarrow$ Introduce a variable $P_{\delta A}$ for each differential byte that passes through *SB* (in red)
$\rightsquigarrow$ Relate $P_{\delta A}$ with δA and δSA: $(\delta A, \delta SA, P_{\delta A}) \in T_{sbox}$ where

$$T_{sbox} = \{(\delta_{in}, \delta_{out}, \log_2(p(\delta_{out}|\delta_{in}))) \mid \delta_{in}, \delta_{out} \in [0, 255], p(\delta_{out}|\delta_{in} > 0\}$$

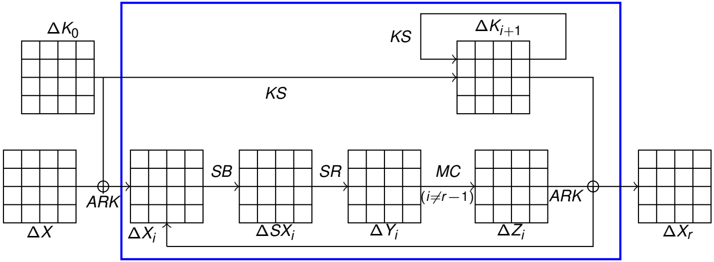
CP model for computing maximal differential characteristics for AES-128

- Maximize $\sum_{i,j,k} P_{\delta X_{i,j,k}} + \sum_{i,j} P_{\delta K_{i,j,3}}$
- SB: $\forall i \in [0, r-1], \forall j, k \in [0,3], (\delta X_{i,j,k}, \delta SX_{i,j,k}, P_{\delta X_{i,j,k}}) \in T_{sbox}$
- SR: $\forall i \in [0, r-1], \forall j, k \in [0,3], \delta Y_{i,j,k} = \delta SX_{i,j,(k+j)\%4}$
- MC: $\forall i \in [0, r-2], \forall j, k \in [0,3]$,
  - $(\delta Y_{i,x,k}, A_x) \in T_x$ where $T_x = \{(y, y \otimes M_x) \mid y \in [0,255]\} \quad \forall x \in \{(j,0),(j,1),(j,2),(j,3)\}$
  - $(A_{j,0}, A_{j,1}, B) \in T_\oplus$
  - $(A_{j,2}, A_{j,3}, C) \in T_\oplus$
  - $(B, C, \delta Z_{i,j,k}) \in T_\oplus$
- ARK:
  - $\forall j, k \in [0,3], (\delta X_{0,j,k}, \delta K_{0,j,k}, \delta X_{j,k}) \in T_\oplus$
  - $\forall i \in [0, r-1], \forall j, k \in [0,3], (\delta X_{i+1,j,k}, \delta K_{i,j,k}, \delta Z_{i,j,k}) \in T_\oplus$
- SK:
  - $\forall i \in [0, r-1], \forall j \in [0,3], (\delta K_{i+1,j,0}, \delta SK_{i,(j+1)\%4,3}, \delta K_{i,j,0}) \in T_\oplus$
  - $\forall i \in [0, r-1], \forall j \in [0,3], (\delta K_{i,j,3}, \delta SK_{i,j,3}, P_{\delta K_{i,j,3}}) \in T_{sbox}$
  - $\forall i \in [0, r-1], \forall j \in [0,3], \forall k \in [1,3], (\delta K_{i+1,j,k}, \delta K_{i+1,j,k-1}, \delta K_{i,j,k}) \in T_\oplus$

Two step solving process [Knu95]

Step 1: Compute an optimal Truncated Differential Characteristic (TDC)

- Each differential byte $\delta B = B \oplus B'$ is abstracted to a boolean ΔB
  $\rightsquigarrow \Delta B = 0$ if $B = B'$; $\Delta B = 1$ if $B \neq B'$
- Minimise the number of boolean variables $\Delta X_{i,j,k}$ and $\Delta K_{i,j,3}$ set to 1:
  - If $\delta X_{i,j,k} = 0$ then $\delta SX_{i,j,k} = 0$ and $p(\delta SX_{i,j,k} | \delta X_{i,j,k}) = 1$
  - Otherwise $p(\delta SX_{i,j,k}) | \delta X_{i,j,k}) \in \{0, 2^{-7}, 2^{-6}\}$

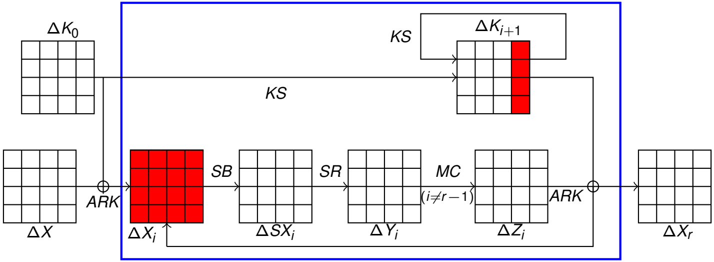


[Knu95] L. Knudsen: *Truncated and higher order differentials*. In Fast Software Encryption 1995

Two step solving process [Knu95]

Step 1: Compute an optimal Truncated Differential Characteristic (TDC)

- Each differential byte $\delta B = B \oplus B'$ is abstracted to a boolean ΔB
  $\rightsquigarrow \Delta B = 0$ if $B = B'$; $\Delta B = 1$ if $B \neq B'$
- Minimise the number of boolean variables $\Delta X_{i,j,k}$ and $\Delta K_{i,j,3}$ set to 1:
  - If $\delta X_{i,j,k} = 0$ then $\delta SX_{i,j,k} = 0$ and $p(\delta SX_{i,j,k} | \delta X_{i,j,k}) = 1$
  - Otherwise $p(\delta SX_{i,j,k}) | \delta X_{i,j,k}) \in \{0, 2^{-7}, 2^{-6}\}$



[Knu95] L. Knudsen: *Truncated and higher order differentials*. In Fast Software Encryption 1995

Two step solving process [Knu95]

Step 1: Compute an optimal Truncated Differential Characteristic (TDC)

- Each differential byte $\delta B = B \oplus B'$ is abstracted to a boolean ΔB
  $\rightsquigarrow \Delta B = 0$ if $B = B'$; $\Delta B = 1$ if $B \neq B'$
- Minimise the number of boolean variables $\Delta X_{i,j,k}$ and $\Delta K_{i,j,3}$ set to 1:
  - If $\delta X_{i,j,k} = 0$ then $\delta SX_{i,j,k} = 0$ and $p(\delta SX_{i,j,k}|\delta X_{i,j,k}) = 1$
  - Otherwise $p(\delta SX_{i,j,k})|\delta X_{i,j,k}) \in \{0, 2^{-7}, 2^{-6}\}$

Step 2: Use the optimal TDC to tighten domains

- For each boolean ΔB: If $\Delta B = 0$ then set δB to 0; otherwise set the domain of δB to $[1, 255]$
  - If no solution: The TDC is byte-inconsistent
  - If there are solutions: Search for the differential characteristic with maximal probability

---

[Knu95] L. Knudsen: *Truncated and higher order differentials*. In Fast Software Encryption 1995

Overview of the complete process

1. Initialize p_{max} to 0
2. Search for a TDC that minimizes $v = \sum_{i,j,k} \Delta X_{i,j,k} + \sum_{i,j} \Delta K_{i,j,3}$ (Step1opt)
3. If $2^{-6*v} < 2^{-|K|}$ then Stop (the cipher is indistinguishable from random)
4. Enumerate all TDCs s.t. $v = \sum_{i,j,k} \Delta X_{i,j,k} + \sum_{i,j} \Delta K_{i,j,3}$ (Step1enum)
   - For each TDC, search for a maximal differential characteristic (Step2)
     $\rightsquigarrow$ Update p_{max} if a greater probability is found
5. If $p_{max} < 2^{-6*(v+1)}$ then increment v and go to (3)
6. return p_{max} and the corresponding differential characteristic

Existing dedicated approaches for Step1

[BN10]: Branch & Bound

- $|K| = 128$: Several days of CPU time
- $|K| = 192$: Several weeks of CPU time

[FJP13]: Dynamic Programming

- $|K| = 128$: 30mn of CPU time (on 12 cores)
  ... but memory complexity in $\mathcal{O}(2^{32})$ = 60 GB
- Cannot be extended to $|K| = 192$ or 256

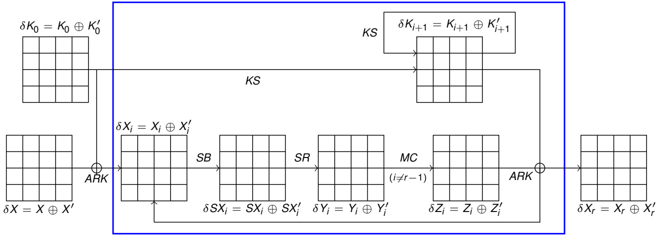
In both cases: Difficult and time-consuming programming work

- Checking the correctness of the program is not straightforward
- Nothing is said about Step 2

[BN10] Biryukov, Nikolic: *Automatic search for related-key differential characteristics in byte-oriented block ciphers: Application to AES, camellia, khazad and others*. In Advances in Cryptology 2010

[FJP13] Fouque, Jean, Peyrin: *Structural evaluation of AES and chosen-key distinguisher of 9-round AES-128*. In CRYPTO

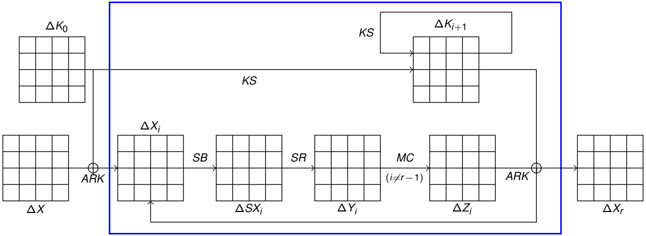
From ModRef 2014 to ModRef 2024



Byte var. for differential characteristics:

- $\delta K_{i,j,k} = K_{i,j,k} \oplus K'_{i,j,k}$
- $\delta X_{i,j,k} = X_{i,j,k} \oplus X'_{i,j,k}$
- Same for $\delta SX_{i,j,k}, \delta Y_{i,j,k}$, ...

$\rightsquigarrow$ Domain = $[0, 255]$



Byte var. for differential characteristics:

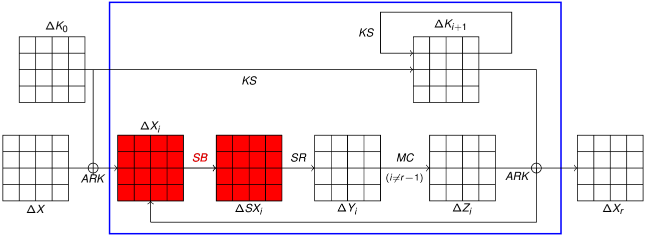
- $\delta K_{i,j,k} = K_{i,j,k} \oplus K'_{i,j,k}$
- $\delta X_{i,j,k} = X_{i,j,k} \oplus X'_{i,j,k}$
- Same for $\delta SX_{i,j,k}, \delta Y_{i,j,k}$, ...

$\rightsquigarrow$ Domain = $[0, 255]$

Boolean variables for TDC:

- $\Delta K_{i,j,k} = 0$ if $K_{i,j,k} = K'_{i,j,k}$; 1 otherwise
- $\Delta X_{i,j,k} = 0$ if $X_{i,j,k} = X'_{i,j,k}$; 1 otherwise
- Same for $\Delta SX_{i,j,k}, \Delta Y_{i,j,k}$, ...

$\rightsquigarrow$ Domain = $\{0, 1\}$



SB constraint for differential characteristics:

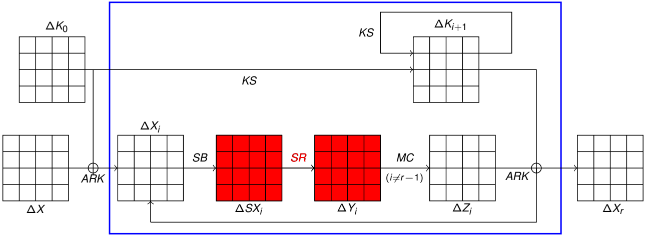
$$(\delta X_{i,j,k}, \delta SX_{i,j,k}, P_{\delta X_{i,j,k}}) \in T_{sbox}$$

where $T_{sbox} = \{(\delta_{in}, \delta_{out}, -\log_2(p(\delta_{out}|\delta_{in})))\}$

- either $\delta_{in} = \delta_{out} = 0$ and $p(\delta_{out}|\delta_{in}) = 1$
- or $\delta_{in} \neq 0$, $\delta_{out} \neq 0$ and $p(\delta_{out}|\delta_{in}) \in \{2^{-6}, 2^{-7}\}$

SB constraint for TDC:

$$\Delta SX_{i,j,k} = \Delta X_{i,j,k}$$

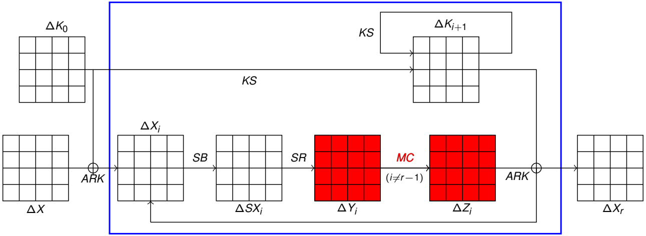


SR constraint for differential characteristics:

$$\delta Y_{i,j,k} = \delta SX_{i,j,(k+j)\%4}$$

SR constraint for TDC:

$$\Delta Y_{i,j,k} = \Delta SX_{i,j,(k+j)\%4}$$



MC constraint for differential characteristics:

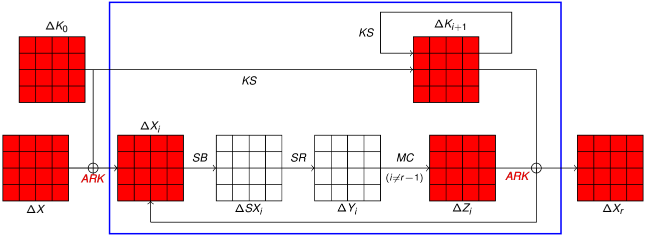
$$\begin{aligned} \delta Z_{i,j,k} &= M_{j,0} \otimes \delta Y_{i,0,k} \\ &\oplus M_{j,1} \otimes \delta Y_{i,1,k} \\ &\oplus M_{j,2} \otimes \delta Y_{i,2,k} \\ &\oplus M_{j,3} \otimes \delta Y_{i,3,k} \end{aligned}$$

MC constraint for TDC:

$$\sum_{j=0}^{3} \Delta Y_{i,j,k} + \Delta Z_{i,j,k} \in \{0, 5, 6, 7, 8\}$$

MDS property:

$$\sum_{j=0}^{3} (\delta Y_{i,j,k} \neq 0) + (\delta Z_{i,j,k} \neq 0) \in \{0, 5, 6, 7, 8\}$$



ARK constraint for differential characteristics:

- $\delta X_{0,j,k} = \delta K_{0,j,k} \oplus \delta X_{j,k}$
- $\delta X_{i+1,j,k} = \delta K_{i,j,k} \oplus \delta Z_{i,j,k}$

ARK constraint for TDC:

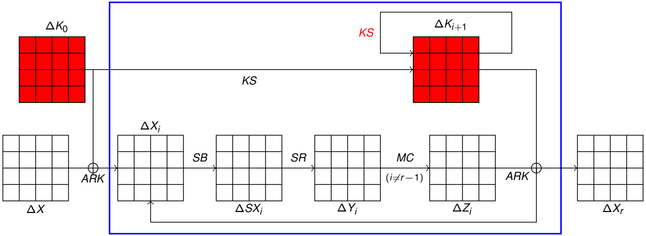
- $\Delta X_{0,j,k} + \Delta K_{0,j,k} + \Delta X_{j,k} \neq 1$
- $\Delta X_{i+1,j,k} + \Delta K_{i+1,j,k} + \Delta Z_{i,j,k} \neq 1$

XOR at the byte level:

- $0 \oplus 0 = 0$
- $0 \oplus x = x, \forall x \in [1, 255]$
- $x \oplus 0 = x, \forall x \in [1, 255]$
- $x \oplus x = 0, \forall x \in [1, 255]$
- $x \oplus y \neq 0, \forall x, y \in [1, 255]$ if $x \neq y$

$\Delta B_1 = \Delta B_2 \oplus \Delta B_3$ at the boolean level:

$$(\Delta B_1, \Delta B_2, \Delta B_3) \in \{ \begin{array}{l} (0, \ 0, \ 0), \\ (0, \ 1, \ 1), \\ (1, \ 0, \ 1), \\ (1, \ 1, \ 0), \\ (1, \ 1, \ 1)\} \end{array}$$



KS constraint for differential characteristics:

- $\delta K_{i+1,j,0} = \delta SK_{i,(j+1)\%4,3} \oplus \delta K_{i,j,0}$
- $(\delta K_{i,j,3}, \delta SK_{i,j,3}, P_{K_{i,j,3}}) \in T_{sbox}$
- $\delta K_{i+1,j,k} = \delta K_{i+1,j,k-1} \oplus \delta K_{i,j,k}$

KS constraint for TDC:

- $\Delta K_{i+1,j,0} + \Delta SK_{i,(j+1)\%4,3} + \Delta K_{i,j,0} \neq 1$
- $\Delta SK_{i,j,3} = \Delta K_{i,j,3}$
- $\Delta K_{i+1,j,k} + \Delta K_{i+1,j,k-1} + \Delta K_{i,j,k} \neq 1$

First CP model for Step1 [MSR14]

- Objective function: $v = \sum_{i,j,k} \Delta X_{i,j,k} + \sum_{i,j} \Delta K_{i,j,3}$
- SB: $\forall i \in [0, r-1], \forall j, k \in [0,3], \Delta X_{i,j,k} = \Delta SX_{i,j,k}$
- SR: $\forall i \in [0, r-1], \forall j, k \in [0,3], \Delta Y_{i,j,k} = \Delta SX_{i,j,(k+j)\%4}$
- MC: $\forall i \in [0, r-2], \forall j, k \in [0,3], \sum_{j=0}^{3} \Delta Y_{i,j,k} + \Delta Z_{i,j,k} \in \{0,5,6,7,8\}$
- ARK:
  - $\forall j, k \in [0,3], \Delta X_{0,j,k} + \Delta K_{0,j,k} + \Delta X_{j,k} \neq 1$
  - $\forall i \in [0, r-1], \forall j, k \in [0,3], \Delta X_{i+1,j,k} + \Delta K_{i,j,k} + \Delta Z_{i,j,k} \neq 1$
- SK:
  - $\forall i \in [0, r-1], \forall j \in [0,3], \Delta K_{i+1,j,0} + \Delta SK_{i,(j+1)\%4,3} + \Delta K_{i,j,0} \neq 1$
  - $\forall i \in [0, r-1], \forall j \in [0,3], \Delta K_{i,j,3} = \Delta SK_{i,j,3}$
  - $\forall i \in [0, r-1], \forall j \in [0,3], \forall k \in [1,3], \Delta K_{i+1,j,k} + \Delta K_{i+1,j,k-1} + \Delta K_{i,j,k} \neq 1$

Ordering heuristics:

- First choose variables that occur in the objective function
- First assign them to 0

[MSR14] M. Minier, C. Solnon, J. Reboul: *Solving a Symmetric Key Cryptographic Problem with CP*. In ModRef 2014

Experimental results for enumerating all TDCs for AES-128

r	v	Byte sol.	Bool. sol.	Gecode Time	Gecode CP	Choco 4 Time	Choco 4 CP	Chuffed Time	Chuffed CP
3	2	0	0	**0.0**	$9e^1$	**0.0**	$4e^1$	**0.0**	$5e^1$
3	3	0	$5e^2$	0.1	$2e^3$	0.4	$2e^3$	**0.0**	$7e^2$
3	4	0	$5e^3$	1.3	$2e^4$	1.8	$1e^4$	**0.2**	$5e^3$
3	5	2	$2e^4$	6.0	$6e^4$	5.1	$5e^4$	**0.9**	$2e^4$
4	8	0	0	**0.2**	$2e^4$	0.6	$1e^4$	0.3	$8e^3$
4	9	0	$2e^4$	7.1	$1e^5$	5.4	$7e^4$	**1.4**	$4e^4$
4	10	0	$6e^6$	-	-	1161.2	$2e^7$	**113.5**	$6e^6$
4	11	0	$9e^7$	-	-	-	-	**1974.5**	$9e^7$
4	12	2	-	-	-	-	-	-	-

- r = Number of rounds
- v = Number of differences that pass through SB (active S-boxes)
- CP = number of choice points in the search tree
  $\rightsquigarrow$ Chuffed explores less choice points and is faster

Problem of this first model: Most TDCs can't be concretised to differential characteristics

From ModRef 2014 to ModRef 2024

New variables to model byte equalities [GMS16]

What's wrong with the first CP model?

XOR constraints do not propagate equality relationships at the byte level

Example:

- At byte level: $(\delta a \oplus \delta b \oplus \delta c = 0) \wedge (\delta a = \delta b) \Rightarrow (\delta c = 0)$
- At Boolean level: $\Delta a + \Delta b + \Delta c \neq 1 \wedge (\Delta a = \Delta b) \not\Rightarrow (\Delta c = 0)$

New variables and constraints to model byte equalities:

- For each couple of differential bytes $(\delta A, \delta B)$: $diff_{\delta A,\delta B} = 1 \Leftrightarrow \delta A \neq \delta B$
- Symmetry: $diff_{\delta A,\delta B} = diff_{\delta B,\delta A}$
- Transitivity: $diff_{\delta A,\delta B} + diff_{\delta B,\delta C} + diff_{\delta A,\delta C} \neq 1$
- Relation with Δ variables: $diff_{\delta A,\delta B} + \Delta A + \Delta B \neq 1$

Too expensive (and useless) to maintain all relationships
$\rightsquigarrow$ Limit to byte couples in a same row of a same group $(\delta K, \delta Y$, and $\delta Z)$

[GMS16] D. Gerault, M. Minier, C. Solnon: *CP models for chosen key differential cryptanalysis*. In CP 2016

Revisiting the XOR constraint

Definition of XOR in the first CP model: $\Delta B_1 + \Delta B_2 + \Delta B_3 \neq 1$

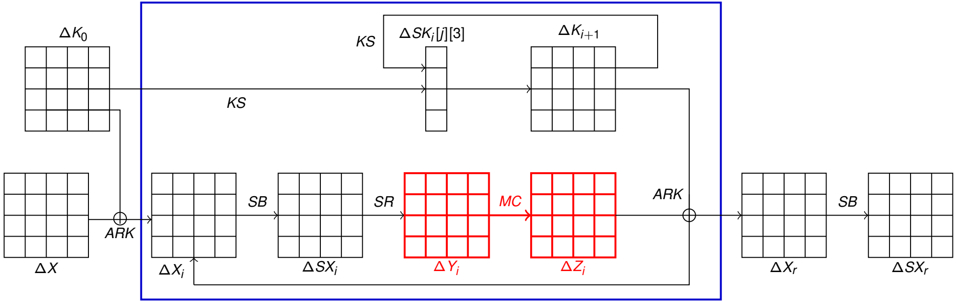
Can we strengthen it by exploiting byte equalities?
Yes, because: $\Delta B_1 = 0 \Leftrightarrow \delta B_2 = \delta B_3$

New definition of XOR: Replace $\Delta B_1 + \Delta B_2 + \Delta B_3 \neq 1$ **with**

$(diff_{\delta B_1, \delta B_2} = \Delta B_3) \wedge (diff_{\delta B_1, \delta B_3} = \Delta B_2) \wedge (diff_{\delta B_2, \delta B_3} = \Delta B_1)$

$\rightsquigarrow$ Every XOR constraint "removes" 3 Boolean variables

Propagation of MDS between different columns



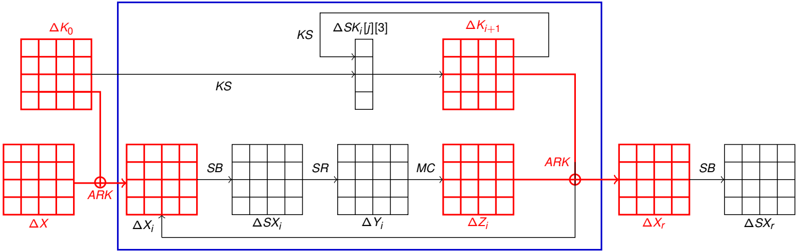
MDS also holds when XORing different columns of δY and δZ:

$\forall i_1, i_2 \in [0, r-2], \forall k_1, k_2 \in [0, 3]$, we have:
$\sum_{j=0}^{3} (\delta Y_{i_1,j,k_1} \oplus \delta Y_{i_2,j,k_2} \neq 0) + (\delta Z_{i_1,j,k_1} \oplus \delta Z_{i_2,j,k_2} \neq 0) \in \{0, 5, 6, 7, 8\}$

New constraints to propagate MDS between different columns:

$\forall i_1, i_2 \in [0, r-2], \forall k_1, k_2 \in [0, 3]$,
$\sum_{j=0}^{3} diff_{\delta Y_{i_1,j,k_1}, \delta Y_{i_2,j,k_2}} + diff_{\delta Z_{i_1,j,k_1}, \delta Z_{i_2,j,k_2}} \in \{0, 5, 6, 7, 8\}$

Propagation of ARK at the byte level



ARK implies the following equations: $\forall i_1, i_2 \in [0, r-2], \forall j, k_1, k_2 \in [0,3]$**:**

$\delta K_{i_1+1,j,k_1} \oplus \delta Z_{i_1,j,k_1} = \delta X_{i_1+1,j,k_1}$ and $\delta K_{i_2+1,j,k_2} \oplus \delta Z_{i_2,j,k_2} = \delta X_{i_2+1,j,k_2}$

By xoring these two equations, we infer that:

$(\delta K_{i_1+1,j,k_1} \neq \delta K_{i_2+1,j,k_2}) + (\delta Z_{i_1,j,k_1} \neq \delta Z_{i_2,j,k_2}) + (\delta X_{i_1+1,j,k_1} \neq \delta X_{i_2+1,j,k_2}) \neq 1$

Corresponding constraint: $\forall i_1, i_2 \in [0, r-2], \forall j, k_1, k_2 \in [0,3]$**:**

$diff_{\delta K_{i_1+1,j,k_1}, \delta K_{i_2+1,j,k_2}} + diff_{\delta Z_{i_1,j,k_1}, \delta Z_{i_2,j,k_2}} + \Delta X_{i_1+1,j,k_1} + \Delta X_{i_2+1,j,k_2} \neq 1$

(because $(\Delta X_{i_1+1,j,k_1} + \Delta X_{i_2+1,j,k_2} = 1) \Rightarrow (\delta X_{i_1+1,j,k_1} \neq \delta X_{i_2+1,j,k_2})$)

Experimental results [GMS16]

$\|K\|$	r	Step1-opt v^*	t	Step1-enum $\#T$	t
128	3	5	4	4	6
128	4	12	21	8	74
128	5	17	44	1113	32340
192	3	1	3	15	16
192	4	4	8	4	12
192	5	5	14	2	13
192	6	10	34	6	65
192	7	13	72	4	98
192	8	18	205	8	752
192	9	24	2527	240	43359
192	10	27	3715	27548	$>$ 2 weeks
256	3	1	3	33	39
256	4	3	8	14	38
256	5	3	13	4	21
256	6	5	25	3	29
256	7	5	48	1	22
256	8	10	61	3	76
256	9	15	172	16	705
256	10	16	236	4	385
256	11	20	488	4	705
256	12	20	625	4	1228
256	13	24	1621	4	1910
256	14	24	2179	4	1722

- MiniZinc model solved with Picat-SAT
- $|K|$ = size of key (in bits)
- r = number of rounds
  $\rightsquigarrow$ Stop when $p_{max} \geq 2^{|K|}$
- $v*$ = objective function value
- t = time in seconds
- $\#T$ = number of TDCs

One instance is still out of reach!

From ModRef 2014 to ModRef 2024

Generation of new XOR equations [GMLS20]

What's wrong with the second model? Example coming from KS:

Let $A = K_{0,0,3}$, $B = K_{1,0,2}$, $C = K_{1,0,3}$, $D = K_{2,0,1}$, $E = K_{2,0,2}$, $F = K_{2,0,3}$. We have:
$(\delta A \oplus \delta B \oplus \delta C = 0) \wedge (\delta B \oplus \delta D \oplus \delta E = 0) \wedge (\delta C \oplus \delta E \oplus \delta F = 0)$

- At the byte level, $\delta D = \delta F = 0 \Rightarrow \delta A = 0$
- At the Boolean level, $\Delta D = \Delta F = 0 \not\Rightarrow \Delta A = 0$

Idea: Generate new XOR constraints to tighten the abstraction

From $\delta A_1 \oplus \ldots \oplus \delta A_n = 0$ and $\delta B_1 \oplus \ldots \oplus \delta B_m = 0$, we generate:
$\bigoplus_{C \in \{A_1,\ldots,A_n\} \cup \{B_1,\ldots,B_m\} \setminus \{A_1,\ldots,A_n\} \cap \{B_1,\ldots,B_m\}} \delta C = 0$

Example:

$(\delta A \oplus \delta B \oplus \delta C = 0) \wedge (\delta B \oplus \delta D \oplus \delta E = 0) \Rightarrow (\delta A \oplus \delta C \oplus \delta D \oplus \delta E = 0)$
$(\delta A \oplus \delta C \oplus \delta D \oplus \delta E = 0) \wedge (\delta C \oplus \delta E \oplus \delta F = 0) \Rightarrow (\delta A \oplus \delta D \oplus \delta F = 0)$

- At the Boolean level, $\Delta D = \Delta F = 0 \Rightarrow \Delta A = 0$

[GMLS20] Gerault, Lafourcade, Minier, Solnon: *Computing AES related-key differential characteristics with CP*. In AIJ 2020

Generation of new XOR equations [GMLS20]

What's wrong with the second model? Example coming from *KS*:

Let $A = K_{0,0,3}$, $B = K_{1,0,2}$, $C = K_{1,0,3}$, $D = K_{2,0,1}$, $E = K_{2,0,2}$, $F = K_{2,0,3}$. We have:
$(\delta A \oplus \delta B \oplus \delta C = 0) \wedge (\delta B \oplus \delta D \oplus \delta E = 0) \wedge (\delta C \oplus \delta E \oplus \delta F = 0)$

- At the byte level, $\delta D = \delta F = 0 \Rightarrow \delta A = 0$
- At the Boolean level, $\Delta D = \Delta F = 0 \not\Rightarrow \Delta A = 0$

Idea: Generate new XOR constraints to tighten the abstraction

From $\delta A_1 \oplus \ldots \oplus \delta A_n = 0$ and $\delta B_1 \oplus \ldots \oplus \delta B_m = 0$, we generate:
$\bigoplus_{C \in \{A_1,\ldots,A_n\} \cup \{B_1,\ldots,B_m\} \setminus \{A_1,\ldots,A_n\} \cap \{B_1,\ldots,B_m\}} \delta C = 0$

Example:

$(\delta A \oplus \delta B \oplus \delta C = 0) \wedge (\delta B \oplus \delta D \oplus \delta E = 0) \Rightarrow (\delta A \oplus \delta C \oplus \delta D \oplus \delta E = 0)$
$(\delta A \oplus \delta C \oplus \delta D \oplus \delta E = 0) \wedge (\delta C \oplus \delta E \oplus \delta F = 0) \Rightarrow (\delta A \oplus \delta D \oplus \delta F = 0)$

- At the Boolean level, $\Delta D = \Delta F = 0 \Rightarrow \Delta A = 0$

[GMLS20] Gerault, Lafourcade, Minier, Solnon: *Computing AES related-key differential characteristics with CP*. In AIJ 2020

Generation of new XOR equations (2/2)

Number of new equations for AES128:

- $r = 4$: 988
- $r = 5$: 16332
- $r = 6$: CPU time exceeds one hour

Number of new equations when limiting the size to 4:

	AES128	AES192	AES256
# Initial eq.	144	168	192
# new eq. with 3 bytes	122	168	144
# new eq. with 4 bytes	1104	1696	1256

- CPU time always smaller than 0.1s
- Proof of completeness by Jérémie Detrey

Experimental comparison of models 2 and 3

	Step1-opt				*Step1-enum*			
	Model 2		Model 3		Model 2		Model 3	
	v^*	t	v^*	t	$\#T$	t	$\#T$	t
AES-128-4	12	21	12	**14**	8	74	8	**38**
AES-128-5	17	44	17	**33**	1113	32340	1113	**22869**
AES-192-4	4	8	4	**5**	4	12	4	**7**
AES-192-5	5	14	5	**8**	2	13	2	**9**
AES-192-6	10	34	10	**18**	6	65	6	**45**
AES-192-7	13	72	13	**37**	4	98	4	**66**
AES-192-8	18	205	18	**73**	8	752	8	**333**
AES-192-9	24	2527	24	**520**	240	43359	240	**13524**
AES-192-10	27	3715	**29**	**3285**	27548	-	**602**	**216120**
AES-256-4	3	8	3	**7**	14	38	14	**25**
AES-256-5	3	13	3	**8**	4	21	4	**15**
AES-256-6	5	25	5	**17**	3	29	3	**20**
AES-256-7	5	48	5	**47**	1	22	1	**15**
AES-256-8	10	61	10	**49**	3	76	3	**52**
AES-256-9	15	172	15	**106**	16	705	16	**430**
AES-256-10	16	236	16	**112**	4	385	4	**224**
AES-256-11	20	488	20	**286**	4	705	4	**312**
AES-256-12	20	625	20	**140**	4	1228	4	**463**
AES-256-13	24	1621	24	**822**	4	1910	4	**597**
AES-256-14	24	2179	24	**682**	4	1722	4	**607**

From ModRef 2014 to ModRef 2024

Overview of the complete process (recall)

1. Initialize p_{max} to 0
2. Search for a TDC that minimizes $v = \sum_{i,j,k} \Delta X_{i,j,k} + \sum_{i,j} \Delta K_{i,j,3}$ (Step1opt)
3. If $2^{-6*v} < 2^{-|K|}$ then Stop (the cipher is indistinguishable from random)
4. Enumerate all TDCs s.t. $v = \sum_{i,j,k} \Delta X_{i,j,k} + \sum_{i,j} \Delta K_{i,j,3}$ (Step1enum)
   - For each TDC, search for a maximal differential characteristic (Step2)
     $\rightsquigarrow$ Update p_{max} if a greater probability is found
5. If $p_{max} < 2^{-6*(v+1)}$ then increment v and go to (3)
6. return p_{max} and the corresponding differential characteristic

Time for solving Step2 with Choco 3

	#Bool. sol.	#Byte sol.	p	t_2	$\frac{t_2}{\#\text{Bool. sol}}$
AES-128-4	8	8	2^{-75}	40	5
AES-128-5	1113	97	2^{-105}	235086	211.2
AES-192-4	4	4	2^{-24}	13	3.3
AES-192-5	2	2	2^{-30}	11	5.5
AES-192-6	6	6	2^{-60}	35	5.8
AES-192-7	4	4	2^{-78}	46	11.5
AES-192-8	8	8	2^{-108}	119	14.9
AES-192-9	240	80	2^{-146}	35254	146.9
AES-192-10	602	202	2^{-176}	55310	91.9
AES-256-4	14	14	2^{-18}	25	1.8
AES-256-5	4	4	2^{-18}	12	3
AES-256-6	3	3	2^{-30}	11	3.7
AES-256-7	1	1	2^{-30}	9	8.8
AES-256-8	3	1	2^{-60}	19	6.3
AES-256-9	16	16	2^{-92}	457	28.6
AES-256-10	4	4	2^{-98}	160	40
AES-256-11	4	4	2^{-122}	178	44.5
AES-256-12	4	4	2^{-122}	237	59.3
AES-256-13	4	4	2^{-146}	244	61
AES-256-14	4	4	2^{-146}	302	75.5

Some instances are challenging!

Can we improve this?

New two-step decomposition [GLMS20]

Problem with the existing decomposition:

- 3 instances (128-5, 192-9, and 192-10) have many Boolean solutions
- Step 2 is time consuming on these instances, even if each Boolean solution is processed rather quickly

New decomposition: Shift the frontier between Steps 1 and 2

- Modify the goal of *Step1-enum*:
  - Old goal = Enumerate all Boolean solutions
  - New goal = Only consider variables that pass through Sboxes
    $\rightsquigarrow$ Enumerate all consistent assignments of $\Delta X_i[j][k]$ and $\Delta K_i[j][3]$

[GMLS20] Gerault, Lafourcade, Minier, Solnon: *Computing AES related-key differential characteristics with CP*. In AIJ 2020

Experimental results

	New Step 1		New Step 2			
	$\#T$	t_1	$\#B$	t_2	$\frac{t_2}{\#T}$	$t_1 + t_2$
AES-128-4	1	8	1	13	12.6	35
AES-128-5	103	1409	27	52313	507.9	53755
AES-192-4	2	4	2	7	3.5	16
AES-192-5	1	4	1	4	3.8	16
AES-192-6	2	11	2	14	7.0	43
AES-192-7	1	17	1	7	7.4	61
AES-192-8	1	57	1	8	8.2	138
AES-192-9	3	386	3	109	36.3	1015
AES-192-10	7	13558	7	281	40.1	17124
AES-256-4	10	14	10	24	2.4	45
AES-256-5	4	10	4	15	3.8	33
AES-256-6	3	12	3	16	5.3	45
AES-256-7	1	8	1	7	7.4	62
AES-256-8	2	18	2	14	7.0	81
AES-256-9	4	63	4	69	17.3	238
AES-256-10	1	41	1	45	45.3	198
AES-256-11	1	77	1	28	27.8	391
AES-256-12	1	89	1	35	35.2	264
AES-256-13	1	140	1	46	46.0	1008
AES-256-14	1	97	1	35	34.8	814

All instances but 2 are solved in less than 1h

- AES-128-5 solved in less than 15h
- AES-192-10 solved in less than 5h

$\rightsquigarrow$ Clear improvement over [BN10] and [FJP13]

New results and attacks:

- AES-128-4: $p_{max} = 2^{-79}$, greater than the solution given in [BN10] and [FJP13] (2^{-81})
- AES-256-14: $p_{max} = 2^{-146}$, greater than the solution given in [BKN09] (2^{-154})
- Improvement of related-key distinguisher and related-key differential attack on the full AES-256 by a factor 64

Related CP models

- Computation of differential characteristics for other ciphers: MIDORI [GL16], SKINNY [DDH+21], Rijndael [RGM+22]
- Other differential cryptanalysis problems: Boomerang attacks on SKINNY [DDV20], Rijndael [RMS24], Rectangle attacks on WARP [LMR22]

Designing models is usually quite easy, but designing efficient models is much harder!

Can we automatically generate them?

[GL16] Gérault, Lafourcade: *Related-key cryptanalysis of MIDORI*. In INDOCRYPT, 2016

[DDV20] Delaune, Derbez, Vavrille: *Catching the Fastest Boomerangs: Application to SKINNY*. In IACR transactions on symmetric cryptology 2020

[DDH+21] Delaune, Derbez, Huynh, Minier, Mollimard, Prud'Homme: *Efficient methods to search for best differential characteristics on SKINNY*. In Applied Cryptography and Network Security 2021

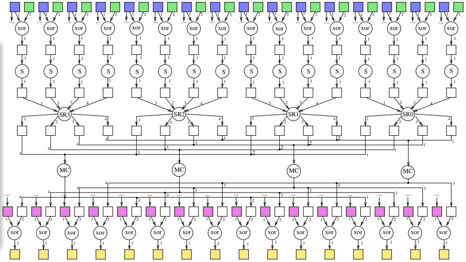
[RGM+22] Rouquette, Gérault, Minier, Solnon: *And rijndael? Automatic related-key differential analysis of Rijndael*. In AfricaCrypt 2022

[LMR22] Lallemand, Minier, Rouquette: *Automatic search of rectangle attacks on feistel ciphers: application to WARP*. In IACR Transactions on Symmetric Cryptology 2022

[RMS24] Rouquette, Minier, Solnon: *Automatic boomerang attacks search on Rijndael*. In Mathematical Cryptology 2024

From ModRef 2014 to ModRef 2024

Overview of Tagada `https://gitlab.com/tagada-framework/tagada`



Input: Description of the cipher by means of a DAG

- Vertices = Operators or Parameters (k-bit words)
  $\rightsquigarrow$ Executable functions associated with operators
- Arcs connect operators to their parameters

$\rightsquigarrow$ Correctness tested with initialisation vectors

Output:

- MiniZinc model for computing TDCs (Step1-opt and Step1-enum) [LDL+21]
- Choco model for computing a maximal DC given a TDC (Step2) [DDG+23]

[LDL+21] L. Libralesso, F. Delobel, P. Lafourcade, C. Solnon: *Automatic generation of declarative models for differential cryptanalysis*. In CP 2021

[DDG+23] F. Delobel, P. Derbez, A. Gontier, L. Rouquette, C. Solnon: *A CP-based Automatic Tool for Instantiating Truncated Differential Characteristics*. In INDOCRYPT 2023

Generation of MiniZing models for computing TDCs

1: Automatic generation of a table constraint for each operator o

- Generate the table of all consistent boolean tuples using the executable function of o

2: Simplify the DAG

- Merge equal parameters
- Suppress constant and free parameters

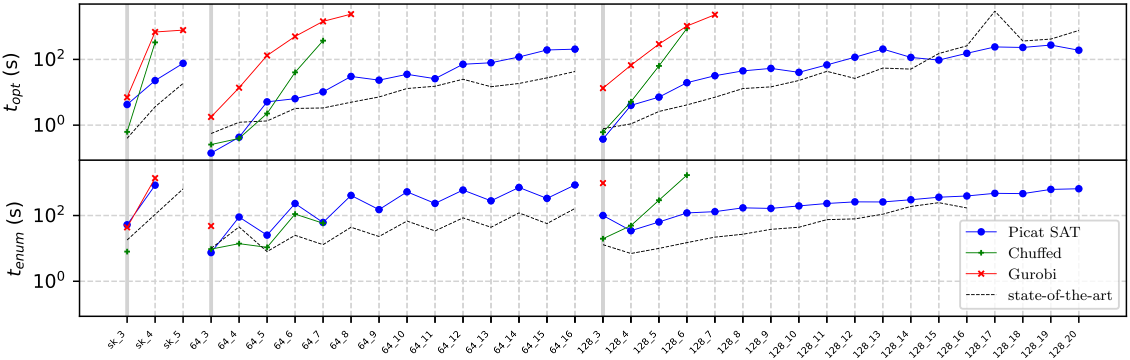
3: Extend the DAG to tighten the abstraction

- Generate *diff* variables
- Generate new XORs

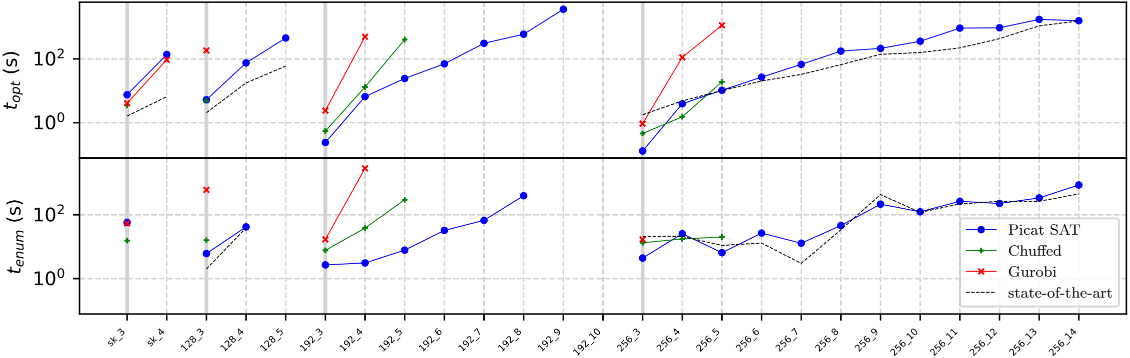
4: Generate a MiniZinc model from the DAG

- Declare a boolean variable for each parameter
- Post a constraint for each operator
- Declare an integer variable corresponding to the number of active S-boxes

Experimental results: Midori



Experimental results: AES



(See [LDL+21] for results on Skinny and Craft
and [DDG+23] for Step2 results on Midori, Warp, Twine, Skinny, and Rijndael)

[LDL+21] L. Libralesso, F. Delobel, P. Lafourcade, C. Solnon: *Automatic generation of declarative models for differential cryptanalysis*. In CP 2021

[DDG+23] F. Delobel, P. Derbez, A. Gontier, L. Rouquette, C. Solnon: *A CP-based Automatic Tool for Instantiating Truncated Differential Characteristics*. In INDOCRYPT 2023

From ModRef 2014 to ModRef 2024

Conclusion

Differential cryptanalysis is a very nice application for CP

- Step1 is easy to model with MiniZinc or XCSP3
  - Advanced constraints must be added to tighten the abstraction
  - Tagada can automatically infer very efficient models from cipher specifications
  - SAT solvers are more efficient than CP solvers
- Step2: Table constraints allow us to easily model non linear operators

Further work: Extensions of Tagada

- Other attacks: Boomerang, related-tweak, ...
- Use dynamic programming to solve Step1
- Study variable and value ordering heuristics

Further work: Certification

Can we automatically build mathematical proofs?