Automated Search for Full Impossible Differential, Zero-Correlation, and Integral Attacks

Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder

ISC-Webinar 2022 - Virtual

> hsn.hadipour@gmail.com

TU Graz

SCIENCE
PASSION
TECHNOLOGY

Outline

Introduction

Cryptographic Primitives

- Symmetric-Key Primitives
  - Block ciphers (AES [DR99], CLEFIA [Shi+07], SKINNY [Bei+16])
  - Stream ciphers (Trivium[CP08], ZUC [ETS11], Enocoro-128v2 [WOK10])
  - Unkeyed primitives (Ascon [Dob+16], Keccak [Ber+13])
- Public-Key Primitives
  - Public-key encryption algorithms (RSA, ECC)
  - Digital signature algorithms (RSA, ECC)
  - Key-exchange protocols (DH)

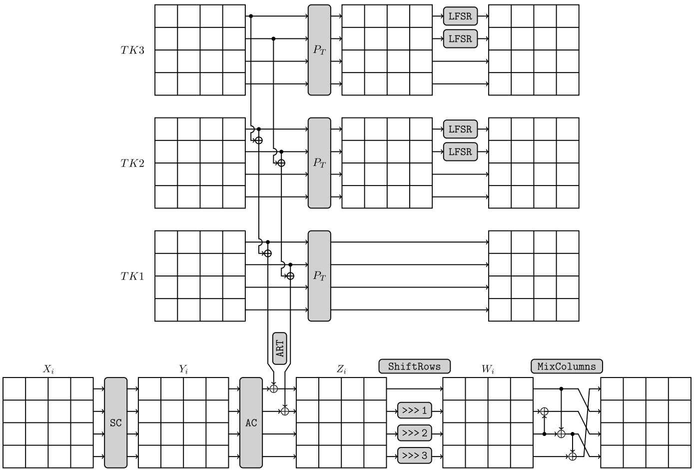
Cryptographic Primitives

- Symmetric-Key Primitives
  - Block ciphers (AES [DR99], CLEFIA [Shi+07], SKINNY [Bei+16])
  - Stream ciphers (Trivium[CP08], ZUC [ETS11], Enocoro-128v2 [WOK10])
  - Unkeyed primitives (Ascon [Dob+16], Keccak [Ber+13])
- Public-Key Primitives
  - Public-key encryption algorithms (RSA, ECC)
  - Digital signature algorithms (RSA, ECC)
  - Key-exchange protocols (DH)

Cryptographic Primitives

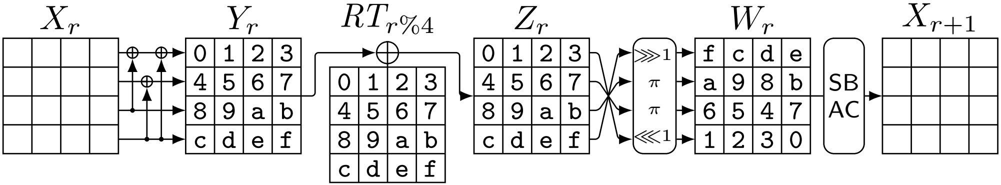
- Symmetric-Key Primitives
  - Block ciphers (AES [DR99], CLEFIA [Shi+07], SKINNY [Bei+16])
  - Stream ciphers (Trivium[CP08], ZUC [ETS11], Enocoro-128v2 [WOK10])
  - Unkeyed primitives (Ascon [Dob+16], Keccak [Ber+13])
- Public-Key Primitives
  - Public-key encryption algorithms (RSA, ECC)
  - Digital signature algorithms (RSA, ECC)
  - Key-exchange protocols (DH)

SKINNY Family of Twekable Block Ciphers [Bei+16]



Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder

CRAFT Tweakable Block Cipher[Bei+19]



- Tweakey schedule:

$$TK_0 = K_0 \oplus T,\ \ TK_1 = K_1 \oplus T,\ \ TK_2 = K_0 \oplus Q(T),\ \ TK_3 = K_1 \oplus Q(T)$$

- Q: a permutation on the position of nibbles
- Round tweakey: $TK_{i\%4}$

Symmetric-Key V.S. Public-Key Cryptography

- Symmetric-key primitives are faster, but require a pre-shared key
- Public-key primitives are slower, but require no pre-shared key
- Modern cryptographic systems employ both types of primitives
- Public-key cryptography is used to securely establish a common key
- Symmetric-key cryptography secures the transactions with a common key

Cryptanalysis

- Complexity-theoretic approach (Public-Key primitives)
- Cryptanalytic approach (Symmetric-Key primitives)

Cryptanalysis

- Complexity-theoretic approach (Public-Key primitives)
- Cryptanalytic approach (Symmetric-Key primitives)
  - Linear attack on DES [Mat93]
  - Differential analysis of AES-256 in the related-key setting [BKN09]
  - Integral analysis based on division property on full MISTY [Tod15]
  - Cube attack against reduced round of SHA-3 [Hua+17]

Automated Methods in Cryptanalysis

Mounting cryptanalytic attacks against symmetric-key primitives:

- requires tracing the propagation of a certain property at the bit-level
- implies solving a hard combinatorial optimization problem
- is very time-consuming
- is potentially an error-prone process

Automated Methods in Cryptanalysis

Getting the help or using of machines to find, build or optimize the attacks

Automated Methods in Cryptanalysis

Mounting cryptanalytic attacks against symmetric-key primitives:

- requires tracing the propagation of a certain property at the bit-level
- implies solving a hard combinatorial optimization problem
- is very time-consuming
- is potentially an error-prone process

Automated Methods in Cryptanalysis

Getting the help or using of machines to find, build or optimize the attacks

Different Approaches for Automatic Cryptanalysis

- Dedicated algorithms
- Constraint Satisfaction/Optimization Problem (CSP/COP)
  - CP
    - MILP
    - SAT
    - SMT
- Artificial Intelligence (AI)

Different Approaches for Automatic Cryptanalysis

- Dedicated algorithms
- Constraint Satisfaction/Optimization Problem (CSP/COP)
  - CP
    - MILP
    - SAT
    - SMT
- Artificial Intelligence (AI)

Constraint Programming (CP)

Constraint Programming (CP)

- In **CP** we specify the properties of the solution to be found:
  - We define a set of variables: $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$
  - We specify the domain of each variable: $\mathbb{F}_2, \mathbb{Z}, \mathbb{R}, \ldots$
  - We define a set of constraints: $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_2\}$
  - We define an objective function (if it is required)
- MILP and SAT are special cases of CP

Constraint Programming (CP)

- In **CP** we specify the properties of the solution to be found:
  - We define a set of variables: $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$
  - We specify the domain of each variable: $\mathbb{F}_2, \mathbb{Z}, \mathbb{R}, \ldots$
  - We define a set of constraints: $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_2\}$
  - We define an objective function (if it is required)
- MILP and SAT are special cases of CP

Constraint Programming (CP)

- In **CP** we specify the properties of the solution to be found:
  - We define a set of variables: $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$
  - We specify the domain of each variable: $\mathbb{F}_2, \mathbb{Z}, \mathbb{R}, \ldots$
  - We define a set of constraints: $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_2\}$
  - We define an objective function (if it is required)
- MILP and SAT are special cases of CP

Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder

Constraint Programming (CP)

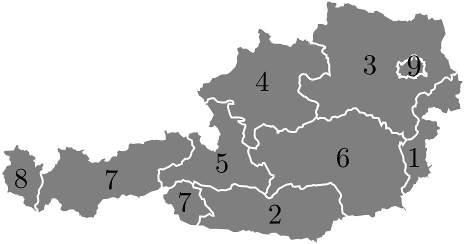
- In **CP** we specify the properties of the solution to be found:
  - We define a set of variables: $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$
  - We specify the domain of each variable: $\mathbb{F}_2, \mathbb{Z}, \mathbb{R}, \ldots$
  - We define a set of constraints: $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_2\}$
  - We define an objective function (if it is required)
- MILP and SAT are special cases of CP

Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder

Constraint Programming (CP)

- In **CP** we specify the properties of the solution to be found:
  - We define a set of variables: $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$
  - We specify the domain of each variable: $\mathbb{F}_2, \mathbb{Z}, \mathbb{R}, \ldots$
  - We define a set of constraints: $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_2\}$
  - We define an objective function (if it is required)
- MILP and SAT are special cases of CP

Constraint Satisfaction/Optimization Problem (CSP/COP)- Example

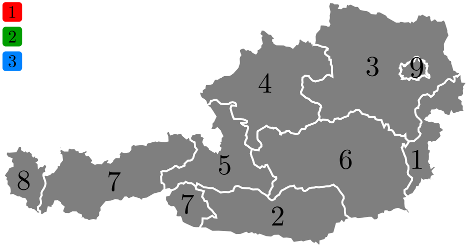



```
int: nc = 3;
array[1..9] of var 1..nc: r;
constraint r[1] != r[3]; constraint r[1] != r[6];
constraint r[2] != r[5]; constraint r[2] != r[6];
constraint r[2] != r[7]; constraint r[3] != r[9];
constraint r[3] != r[6]; constraint r[3] != r[4];
constraint r[4] != r[6]; constraint r[4] != r[5];
constraint r[5] != r[6]; constraint r[5] != r[7];
constraint r[7] != r[8];
solve satisfy;

r = [3, 3, 2, 3, 2, 1, 1, 2, 1];
```


Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder

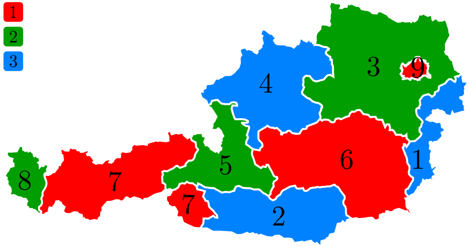
Constraint Satisfaction/Optimization Problem (CSP/COP)- Example




```
int: nc = 3;
array[1..9] of var 1..nc: r;
constraint r[1] != r[3]; constraint r[1] != r[6];
constraint r[2] != r[5]; constraint r[2] != r[6];
constraint r[2] != r[7]; constraint r[3] != r[9];
constraint r[3] != r[6]; constraint r[3] != r[4];
constraint r[4] != r[6]; constraint r[4] != r[5];
constraint r[5] != r[6]; constraint r[5] != r[7];
constraint r[7] != r[8];
solve satisfy;
```

```
r = [3, 3, 2, 3, 2, 1, 1, 2, 1];
```

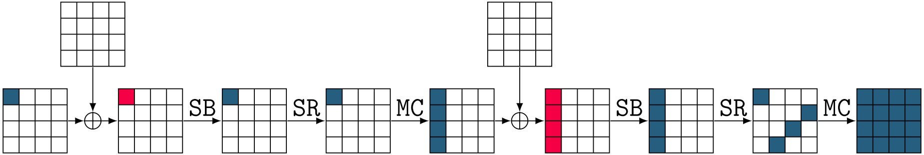
Constraint Satisfaction/Optimization Problem (CSP/COP)- Example




```
int: nc = 3;
array[1..9] of var 1..nc: r;
constraint r[1] != r[3]; constraint r[1] != r[6];
constraint r[2] != r[5]; constraint r[2] != r[6];
constraint r[2] != r[7]; constraint r[3] != r[9];
constraint r[3] != r[6]; constraint r[3] != r[4];
constraint r[4] != r[6]; constraint r[4] != r[5];
constraint r[5] != r[6]; constraint r[5] != r[7];
constraint r[7] != r[8];
solve satisfy;

r = [3, 3, 2, 3, 2, 1, 1, 2, 1];
```

Truncated Differential Trail for AES with Minimum Number of Active S-boxes



Variables:

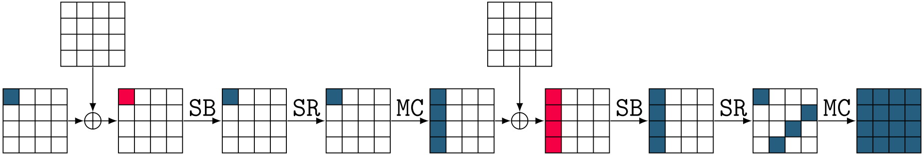
- $s_{r,i,j} \in \{0, 1\}$ is S-box in row i, column j, round r active?
- $m_{r,j} \in \{0, 1\}$ is Mix-columns j in round r active?

Objective function and constraints:

- $\min \sum_{r,i,j} s_{r,i,j}$
- $5 \cdot M_{r,j} \leq \sum_i s_{r,i,(i+j)\%4} + \sum_i s_{r+1,i,j} \leq 8 \cdot M_{r,j}; \quad \sum_{i,j} s_{0,i,j} \geq 1$

Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder

Truncated Differential Trail for AES with Minimum Number of Active S-boxes



Variables:

- $s_{r,i,j} \in \{0,1\}$ is S-box in row i, column j, round r active?
- $m_{r,j} \in \{0,1\}$ is Mix-columns j in round r active?

Objective function and constraints:

- $\min \sum_{r,i,j} s_{r,i,j}$
- $5 \cdot M_{r,j} \leq \sum_i s_{r,i,(i+j)\%4} + \sum_i s_{r+1,i,j} \leq 8 \cdot M_{r,j}; \quad \sum_{i,j} s_{0,i,j} \geq 1$

Impossible Differential Attack (ID)

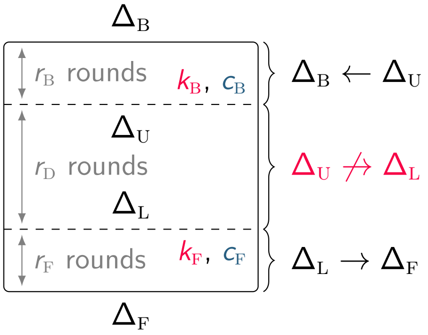
Impossible Differential (ID) Attack

- Core idea: Exploit an impossible event in block cipher to retrieve the secret key

Impossible Differential (ID) Attack

- Core idea: Exploit an impossible event in block cipher to retrieve the secret key

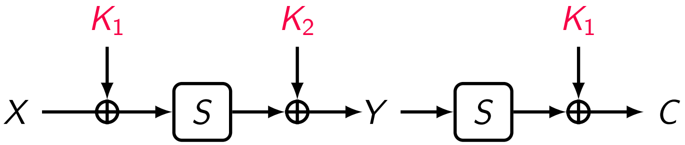
- Find an impossible differential $\Delta_{\mathrm{U}} \not\rightarrow \Delta_{\mathrm{L}}$
- Use it as a **distinguisher** to retrieve the key
  - Keys that suggest $(\Delta_{\mathrm{U}}, \Delta_{\mathrm{L}})$ are wrong
  - Discard as many wrong keys as possible
- Brute force the remaining key candidates



Core Idea of ID Attack

X	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$\mathcal{S}(X)$	c	5	6	b	9	0	a	d	3	e	f	8	4	7	1	2

$$C = S(S(X \oplus K_1) \oplus K_2) \oplus K_1$$



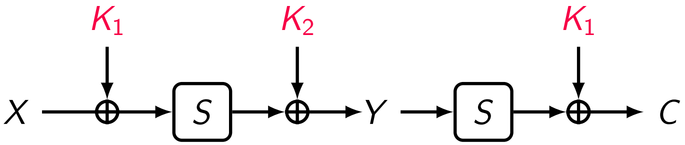
- Impossible differential: if $\Delta X = \mathtt{f}$ then $\Delta Y \notin I = \{0, 2, 3, 5, 6, 7, 8, 9, a, b, c, d\}$
- Filter wrong keys: $S^{-1}(C \oplus K_1) \oplus S^{-1}(C' \oplus K_1) \notin I$ where $P \oplus P' = \mathtt{f}$

Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder

Core Idea of ID Attack

X	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$\mathcal{S}(X)$	c	5	6	b	9	0	a	d	3	e	f	8	4	7	1	2

$$C = S(S(X \oplus K_1) \oplus K_2) \oplus K_1$$



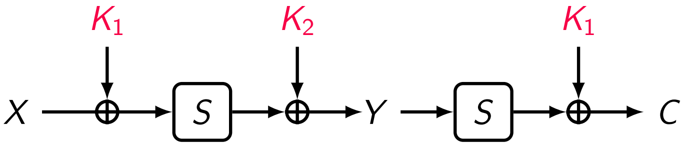
- Impossible differential: if $\Delta X = \mathtt{f}$ then $\Delta Y \notin I = \{0, 2, 3, 5, 6, 7, 8, 9, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$
- Filter wrong keys: $S^{-1}(C \oplus K_1) \oplus S^{-1}(C' \oplus K_1) \notin I$ where $P \oplus P' = \mathtt{f}$

Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder

Core Idea of ID Attack

X	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$\mathcal{S}(X)$	c	5	6	b	9	0	a	d	3	e	f	8	4	7	1	2

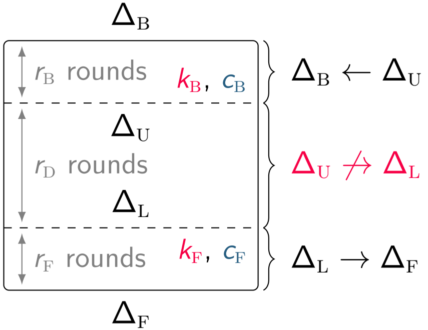
$$C = S(S(X \oplus K_1) \oplus K_2) \oplus K_1$$



- Impossible differential: if $\Delta X = \mathtt{f}$ then $\Delta Y \notin I = \{0, 2, 3, 5, 6, 7, 8, 9, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$
- Filter wrong keys: $S^{-1}(C \oplus K_1) \oplus S^{-1}(C' \oplus K_1) \notin I$ where $P \oplus P' = \mathtt{f}$

Notations

- $\Pr(\Delta_{\mathrm{U}} \to \Delta_{\mathrm{B}}) = 1,\ \Pr(\Delta_{\mathrm{U}} \leftarrow \Delta_{\mathrm{B}}) = 2^{-c_{\mathrm{B}}}$
- $\Pr(\Delta_{\mathrm{L}} \to \Delta_{\mathrm{F}}) = 1,\ \Pr(\Delta_{\mathrm{L}} \leftarrow \Delta_{\mathrm{F}}) = 2^{-c_{\mathrm{F}}}$
- k_{B} is involved to check $\Delta_{\mathrm{B}} \to \Delta_{\mathrm{U}}$
- k_{F} is involved to check $\Delta_{\mathrm{F}} \to \Delta_{\mathrm{L}}$



Key Recovery of ID Attack

- *Pair Generation.* Generate N pairs satisfying the input/output activeness pattern
- *Guess-and-Filter.* For each pair:
  - Guess the involved keys and partially encrypt (decrypt)
  - Discard the key if it yields the impossible differential
- *Exhaustive Search.* Brute force the remaining key candidates

Key Recovery of ID Attack

- *Pair Generation.* Generate N pairs satisfying the input/output activeness pattern
- *Guess-and-Filter.* For each pair:
  - Guess the involved keys and partially encrypt (decrypt)
  - Discard the key if it yields the impossible differential
- *Exhaustive Search.* Brute force the remaining key candidates

Key Recovery of ID Attack

- *Pair Generation.* Generate N pairs satisfying the input/output activeness pattern
- *Guess-and-Filter.* For each pair:
  - Guess the involved keys and partially encrypt (decrypt)
  - Discard the key if it yields the impossible differential
- *Exhaustive Search.* Brute force the remaining key candidates

Key Recovery of ID Attack

- *Pair Generation.* Generate N pairs satisfying the input/output activeness pattern
- *Guess-and-Filter.* For each pair:
  - Guess the involved keys and partially encrypt (decrypt)
  - Discard the key if it yields the impossible differential
- *Exhaustive Search.* Brute force the remaining key candidates

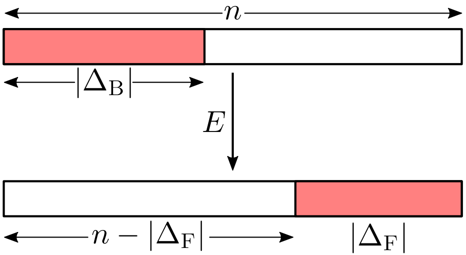
Pair Generation

- Generate N pairs satisfying the input/output activeness pattern
- We construct the pairs by using the plaintext (ciphertext) structures
- For more than one structures

$$N2^{n+1-|\Delta_{\mathrm{B}}|-|\Delta_{\mathrm{F}}|}$$

- For one structure

$$\min_{\Delta \in \{\Delta_{\mathrm{B}}, \Delta_{\mathrm{F}}\}} \left\{ \sqrt{N2^{n+1-|\Delta|}} \right\}$$



Complexity Analysis of ID Attack

- Generate N pairs

$$T_0 = \max\left\{\min_{\Delta\in\{\Delta_{\mathrm{B}},\Delta_{\mathrm{F}}\}}\left\{\sqrt{N2^{n+1-|\Delta|}}\right\}, N2^{n+1-|\Delta_{\mathrm{B}}|-|\Delta_{\mathrm{F}}|}\right\}$$

- Guess and filter

$$T_1 + T_2 = N + 2^{|k_{\mathrm{B}}\cup k_{\mathrm{F}}|}\frac{N}{2^{c_{\mathrm{B}}+c_{\mathrm{F}}}}$$

- Brute force

$$T_3 = 2^{k-|k_{\mathrm{B}}\cup k_{\mathrm{F}}|}\cdot P \cdot 2^{|k_{\mathrm{B}}\cup k_{\mathrm{F}}|} = 2^k \cdot P,\ P = \left(1 - 2^{-(c_{\mathrm{B}}+c_{\mathrm{F}})}\right)^N, P < 2^{-1}$$

$$T_{tot} = (T_0 + (T_1 + T_2)\,C_{E'} + T_3)\,C_E$$

Complexity Analysis of ID Attack

- Generate N pairs

$$T_0 = \max\left\{\min_{\Delta\in\{\Delta_{\mathrm{B}},\Delta_{\mathrm{F}}\}}\left\{\sqrt{N2^{n+1-|\Delta|}}\right\}, N2^{n+1-|\Delta_{\mathrm{B}}|-|\Delta_{\mathrm{F}}|}\right\}$$

- Guess and filter

$$T_1 + T_2 = N + 2^{|k_{\mathrm{B}}\cup k_{\mathrm{F}}|}\frac{N}{2^{c_{\mathrm{B}}+c_{\mathrm{F}}}}$$

- Brute force

$$T_3 = 2^{k-|k_{\mathrm{B}}\cup k_{\mathrm{F}}|}\cdot P\cdot 2^{|k_{\mathrm{B}}\cup k_{\mathrm{F}}|} = 2^k\cdot P,\ P = \left(1-2^{-(c_{\mathrm{B}}+c_{\mathrm{F}})}\right)^N, P < 2^{-1}$$

$$T_{tot} = (T_0 + (T_1 + T_2)\,C_{E'} + T_3)\,C_E$$

Complexity Analysis of ID Attack

- Generate N pairs

$$T_0 = \max\left\{\min_{\Delta \in \{\Delta_{\mathrm{B}}, \Delta_{\mathrm{F}}\}}\left\{\sqrt{N 2^{n+1-|\Delta|}}\right\}, N 2^{n+1-|\Delta_{\mathrm{B}}|-|\Delta_{\mathrm{F}}|}\right\}$$

- Guess and filter

$$T_1 + T_2 = N + 2^{|k_{\mathrm{B}} \cup k_{\mathrm{F}}|} \frac{N}{2^{c_{\mathrm{B}} + c_{\mathrm{F}}}}$$

- Brute force

$$T_3 = 2^{k - |k_{\mathrm{B}} \cup k_{\mathrm{F}}|} \cdot P \cdot 2^{|k_{\mathrm{B}} \cup k_{\mathrm{F}}|} = 2^k \cdot P, \;\; P = \left(1 - 2^{-(c_{\mathrm{B}} + c_{\mathrm{F}})}\right)^N, P < 2^{-1}$$

$$T_{tot} = (T_0 + (T_1 + T_2)\, C_{E'} + T_3)\, C_E$$

Complexity Analysis of ID Attack

- Generate N pairs

$$T_0 = \max\left\{\min_{\Delta \in \{\Delta_{\mathrm{B}}, \Delta_{\mathrm{F}}\}}\left\{\sqrt{N 2^{n+1-|\Delta|}}\right\}, N 2^{n+1-|\Delta_{\mathrm{B}}|-|\Delta_{\mathrm{F}}|}\right\}$$

- Guess and filter

$$T_1 + T_2 = N + 2^{|k_{\mathrm{B}} \cup k_{\mathrm{F}}|} \frac{N}{2^{c_{\mathrm{B}} + c_{\mathrm{F}}}}$$

- Brute force

$$T_3 = 2^{k - |k_{\mathrm{B}} \cup k_{\mathrm{F}}|} \cdot P \cdot 2^{|k_{\mathrm{B}} \cup k_{\mathrm{F}}|} = 2^k \cdot P, \;\; P = \left(1 - 2^{-(c_{\mathrm{B}} + c_{\mathrm{F}})}\right)^N, P < 2^{-1}$$

$$T_{tot} = (T_0 + (T_1 + T_2)\, C_{E'} + T_3)\, C_E$$

Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder

Reformulating the Complexity of ID Attack

- Let $P = 2^{-g},\ 1 < g \leq |k_{\mathrm{B}} \cup k_{\mathrm{F}}|$
- $P = \left(1 - 2^{-(c_{\mathrm{B}}+c_{\mathrm{F}})}\right)^{N}$, thus $N = 2^{c_{\mathrm{B}}+c_{\mathrm{F}}+\log_2(g)-0.53}$
- CP-friendly formulation:

$$T_0 = \max\left\{\begin{array}{l}\min\limits_{\Delta\in\{\Delta_{\mathrm{B}},\Delta_{\mathrm{F}}\}}\{2^{\frac{c_{\mathrm{B}}+c_{\mathrm{F}}+n+1-|\Delta|+LG(g)}{2}}\},\\ 2^{c_{\mathrm{B}}+c_{\mathrm{F}}+n+1-|\Delta_{\mathrm{B}}|-|\Delta_{\mathrm{F}}|+LG(g)}\end{array}\right\},\ \ T_0 < 2^n,$$

$$T_1 = 2^{c_{\mathrm{B}}+c_{\mathrm{F}}+LG(g)},\ \ T_2 = 2^{|k_{\mathrm{B}}\cup k_{\mathrm{F}}|+LG(g)},\ \ T_3 = 2^{k-g},$$

$$T_{\mathrm{tot}} = (T_0 + (T_1 + T_2)\,C_{E'} + T_3)\,C_E,\ \ T_{\mathrm{tot}} < 2^k,$$

Reformulating the Complexity of ID Attack

- Let $P = 2^{-g},\ 1 < g \leq |k_{\mathrm{B}} \cup k_{\mathrm{F}}|$
- $P = \left(1 - 2^{-(c_{\mathrm{B}}+c_{\mathrm{F}})}\right)^N$, thus $N = 2^{c_{\mathrm{B}}+c_{\mathrm{F}}+\log_2(g)-0.53}$
- CP-friendly formulation:

$$T_0 = \max\left\{\begin{array}{l} \min\limits_{\Delta\in\{\Delta_{\mathrm{B}},\Delta_{\mathrm{F}}\}}\{2^{\frac{c_{\mathrm{B}}+c_{\mathrm{F}}+n+1-|\Delta|+LG(g)}{2}}\}, \\ 2^{c_{\mathrm{B}}+c_{\mathrm{F}}+n+1-|\Delta_{\mathrm{B}}|-|\Delta_{\mathrm{F}}|+LG(g)} \end{array}\right\},\quad T_0 < 2^n,$$

$$T_1 = 2^{c_{\mathrm{B}}+c_{\mathrm{F}}+LG(g)},\quad T_2 = 2^{|k_{\mathrm{B}}\cup k_{\mathrm{F}}|+LG(g)},\quad T_3 = 2^{k-g},$$

$$T_{tot} = \left(T_0 + \left(T_1 + T_2\right) C_{E'} + T_3\right) C_E,\quad T_{tot} < 2^k,$$

Reformulating the Complexity of ID Attack

- Let $P = 2^{-g},\ 1 < g \leq |k_{\mathrm{B}} \cup k_{\mathrm{F}}|$
- $P = \left(1 - 2^{-(c_{\mathrm{B}}+c_{\mathrm{F}})}\right)^{N}$, thus $N = 2^{c_{\mathrm{B}}+c_{\mathrm{F}}+\log_2(g)-0.53}$
- CP-friendly formulation:

$$T_0 = \max\left\{\begin{array}{l}\min\limits_{\Delta\in\{\Delta_{\mathrm{B}},\Delta_{\mathrm{F}}\}}\{2^{\frac{c_{\mathrm{B}}+c_{\mathrm{F}}+n+1-|\Delta|+LG(g)}{2}}\},\\ 2^{c_{\mathrm{B}}+c_{\mathrm{F}}+n+1-|\Delta_{\mathrm{B}}|-|\Delta_{\mathrm{F}}|+LG(g)}\end{array}\right\},\ \ T_0 < 2^n,$$

$$T_1 = 2^{c_{\mathrm{B}}+c_{\mathrm{F}}+LG(g)},\ \ T_2 = 2^{|k_{\mathrm{B}}\cup k_{\mathrm{F}}|+LG(g)},\ \ T_3 = 2^{k-g},$$

$$T_{\mathrm{tot}} = (\,T_0 + (\,T_1 + T_2)\,C_{E'} + T_3)\,C_E,\ \ T_{\mathrm{tot}} < 2^k,$$

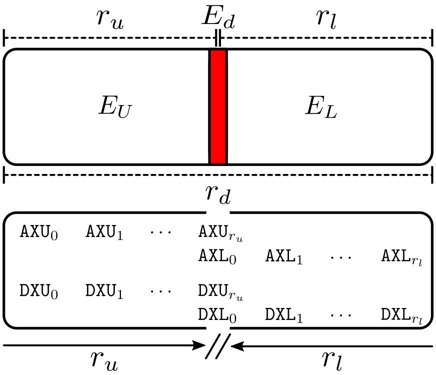
Previous Methods to Search for ID/ZC, and Integral Distinguishers

- Eprint 2016 (ID) [Cui+16]
- EUROCRYPT 2017 (ID, ZC) [ST17]
- ToSC 2017 (ID, ZC) [Sun+17]
- CRYPTO 2016 ($\mathcal{DC}$-MITM, ID) [DF16]
- ASIACRYPT 2016 (Division Property, Integral) [Xia+16]
- ToSC 2020 (ID, ZC) [Sun+20]

Our CP Model to Search for ID Attacks

Our CP Model for Finding ID Distinguishers (High-level View)

- Divide E_{D} into two parts: $E_{\mathrm{D}} = E_{\mathrm{L}} \circ E_{\mathrm{U}}$
- Model the deterministic truncated trails over E_{U} and E_{L} forward and backward
- Include new constraints for the meeting point to guarantee the contradiction



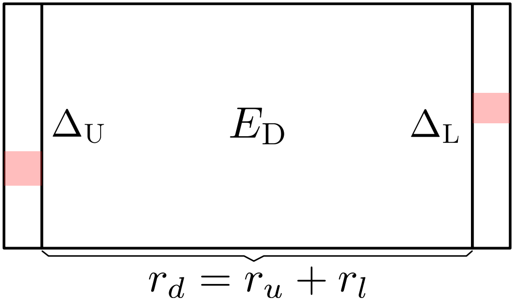
Demo 1

High-level View of Our Unified Model for Key Recovery Attacks

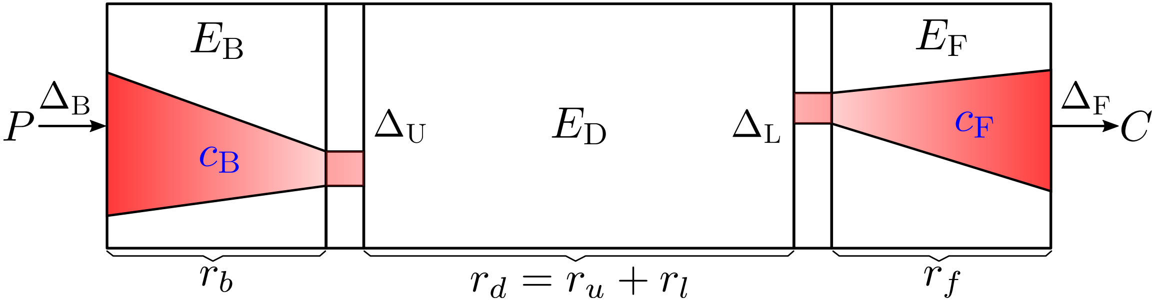
- Model the distinguisher
- Model the difference propagation through the outer parts
- Model the guess-and-determine
- Model the key bridging
- Model the complexity formula

Hosein Hadipour, Sadegh Sadeghi, and Maria Eichlseder

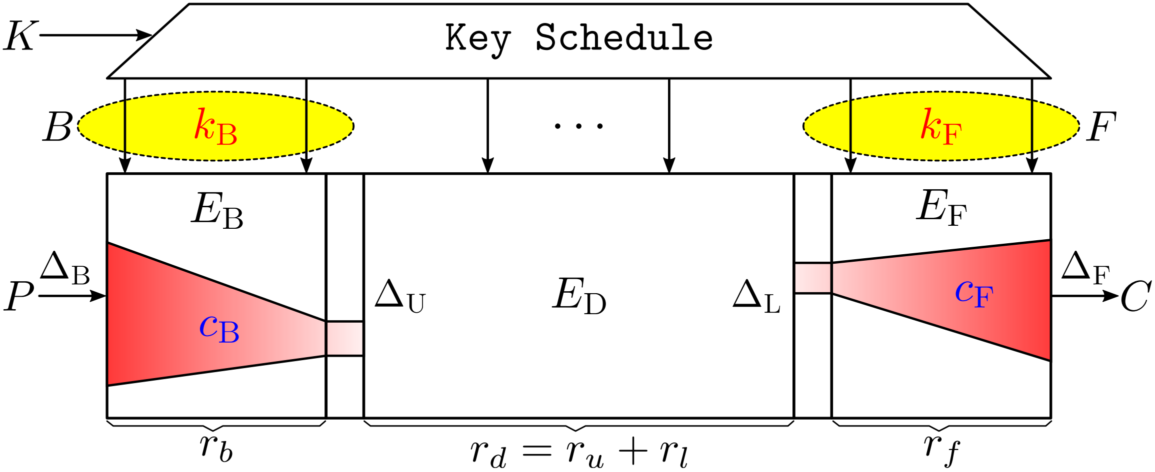
Our Unified CP Model for ID Attack



Our Unified CP Model for ID Attack



Our Unified CP Model for ID Attack



Demo 2

Conclusion

Our Main Contributions

- We introduced a unified CP model for full ID/ZC/Integral attacks
- We applied our method to SKINNY, CRAFT, SKINNYe, and SKINNYee and improved their ID/ZC/Integral attacks significantly
- Our method is generic and can be applied to other strongly aligned block ciphers, e.g., AES

Thanks for your attention!

: https://github.com/hadipourh/talks

: https://ia.cr/2022/1147

Bibliography I

[Bei+16] Christof Beierle et al. **The SKINNY family of block ciphers and its low-latency variant MANTIS**. CRYPTO 2016. Springer. 2016, pp. 123–153. DOI: 10.1007/978-3-662-53008-5_5.

[Bei+19] Christof Beierle et al. **CRAFT: Lightweight Tweakable Block Cipher with Efficient Protection Against DFA Attacks**. *IACR Trans. Symmetric Cryptol.* 2019.1 (2019), pp. 5–45. DOI: 10.13154/tosc.v2019.i1.5-45.

[Ber+13] Guido Bertoni et al. **Keccak**. EUROCRYPT. Vol. 7881. Lecture Notes in Computer Science. Springer, 2013, pp. 313–314. DOI: 10.1007/978-3-642-38348-9_19.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. **Distinguisher and Related-Key Attack on the Full AES-256**. CRYPTO 2009. Vol. 5677. Lecture Notes in Computer Science. Springer, 2009, pp. 231–249.

[CP08] Christophe De Cannière and Bart Preneel. **Trivium**. *New stream cipher designs*. Springer, 2008, pp. 244–266.

Bibliography II

[Cui+16] Tingting Cui et al. **New Automatic Search Tool for Impossible Differentials and Zero-Correlation Linear Approximations**. Cryptology ePrint Archive, Paper 2016/689. https://eprint.iacr.org/2016/689. 2016. URL: https://eprint.iacr.org/2016/689.

[DF16] Patrick Derbez and Pierre-Alain Fouque. **Automatic Search of Meet-in-the-Middle and Impossible Differential Attacks**. CRYPTO 2016. Vol. 9815. Lecture Notes in Computer Science. Springer, 2016, pp. 157–184.

[Dob+16] Christoph Dobraunig et al. **Ascon v1.2**. *Submission to the CAESAR Competition* (2016).

[DR99] Joan Daemen and Vincent Rijmen. **AES proposal: Rijndael**. (1999). https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf.

[ETS11] ETSI/SAGE. **Specification of the 3GPP confidentiality and integrity algorithms 128-EEA3 and 128-EIA3: ZUC specification**. *ETSI/SAGE, Document 2, Version 1.6* (2011).

Bibliography III

[Hua+17] Senyang Huang et al. **Conditional Cube Attack on Reduced-Round Keccak Sponge Function**. EUROCRYPT 2017. Vol. 10211. Lecture Notes in Computer Science. 2017, pp. 259–288. DOI: 10.1007/978-3-319-56614-6_9.

[Mat93] Mitsuru Matsui. **Linear Cryptanalysis Method for DES Cipher**. EUROCRYPT 1993. Vol. 765. Lecture Notes in Computer Science. Springer, 1993, pp. 386–397. DOI: 10.1007/3-540-48285-7_33.

[Shi+07] Taizo Shirai et al. **The 128-Bit Blockcipher CLEFIA (Extended Abstract)**. FSE 2007. Vol. 4593. LNCS. Springer, 2007, pp. 181–195.

[ST17] Yu Sasaki and Yosuke Todo. **New Impossible Differential Search Tool from Design and Cryptanalysis Aspects**. EUROCRYPT 2017. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Cham: Springer International Publishing, 2017, pp. 185–215. DOI: 10.1007/978-3-319-56617-7_7.

Bibliography IV

[Sun+17] Siwei Sun et al. **Analysis of AES, SKINNY, and Others with Constraint Programming**. *IACR Transactions on Symmetric Cryptology* 2017.1 (Mar. 2017), pp. 281–306. DOI: 10.13154/tosc.v2017.i1.281-306.

[Sun+20] Ling Sun et al. **On the Usage of Deterministic (Related-Key) Truncated Differentials and Multidimensional Linear Approximations for SPN Ciphers**. *IACR Transactions on Symmetric Cryptology* 2020.3 (Sept. 2020), pp. 262–287. DOI: 10.13154/tosc.v2020.i3.262-287.

[Tod15] Yosuke Todo. **Integral Cryptanalysis on Full MISTY1**. CRYPTO 2015. Vol. 9215. Lecture Notes in Computer Science. Springer, 2015, pp. 413–432. DOI: 10.1007/s00145-016-9240-x.

[WOK10] D Watanabe, K Okamoto, and T Kaneko. **A Hardware-Oriented Light Weight Pseudo-Random Number Generator Enocoro-128v2. The 2010 Symposium on Cryptography and Information Security, SCIS 2010, 3D1-3**. (2010).

[Xia+16] Zejun Xiang et al. **Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers**. ASIACRYPT 2016. Vol. 10031. Lecture Notes in Computer Science. 2016, pp. 648–678. DOI: 10.1007/978-3-662-53887-6_24.