A Framework for Faster Key Search Using Related-key Higher-order Differential Properties

Christoph Dobraunig[1] **Farokhlagha Moazami** [2] Christian Rechberger[3] Hadi Soleimany[2]

[1]Radboud University, Netherlands

[2]**Cyberspace Research Institute, Shahid Beheshti University, Iran**

[3]Graz University of Technology, Austria

February 2, 2020

Outline

1. Related-key Cryptanalysis

Outline

1. Related-key Cryptanalysis
2. Higher-order Differential Characteristic

Outline

1. Related-key Cryptanalysis
2. Higher-order Differential Characteristic
3. Challenge of Utilizing Related-key Higher-Order Differential Distinguisher

Outline

1. Related-key Cryptanalysis
2. Higher-order Differential Characteristic
3. Challenge of Utilizing Related-key Higher-Order Differential Distinguisher
4. Overview of our Technique

Outline

1. Related-key Cryptanalysis
2. Higher-order Differential Characteristic
3. Challenge of Utilizing Related-key Higher-Order Differential Distinguisher
4. Overview of our Technique
5. Framework for a Cipher with Degree d

Outline

1. Related-key Cryptanalysis
2. Higher-order Differential Characteristic
3. Challenge of Utilizing Related-key Higher-Order Differential Distinguisher
4. Overview of our Technique
5. Framework for a Cipher with Degree d
6. Results on Agrasta

Related-key Cryptanalysis

- In the **related key attack** the attacker aims to derive some information about the key based on an assumption that he is capable of obtaining the encryption of plaintexts or the decryption of ciphertexts under several keys so that the **relation** between the keys is **chosen** by the **attacker**.

Related-key Cryptanalysis

- In the **related key attack** the attacker aims to derive some information about the key based on an assumption that he is capable of obtaining the encryption of plaintexts or the decryption of ciphertexts under several keys so that the **relation** between the keys is **chosen** by the **attacker**.

- **Related-key** model is usually **not** a **realistic** model hence some designers do not take this type of attacks into account.

Related-key Cryptanalysis

- In the **related key attack** the attacker aims to derive some information about the key based on an assumption that he is capable of obtaining the encryption of plaintexts or the decryption of ciphertexts under several keys so that the **relation** between the keys is **chosen** by the **attacker**.
- **Related-key** model is usually **not** a **realistic** model hence some designers do not take this type of attacks into account.
- A natural question is how well the **related-key** distinguishers can lead to **serious vulnerabilities** in practice?

Deterministic Related-key Differential Cryptanalysis

- For the first time Deffie and Helmann introduced a deterministic relation between the encryption of DES under the **key K** and **related key** $\bar{K}$ namely for every K and P if $C = E_K(P)$ then $\bar{C} = E_{\bar{K}}(\bar{P})$.

Deterministic Related-key Differential Cryptanalysis

- For the first time Deffie and Helmann introduced a deterministic relation between the encryption of DES under the **key K** and **related key** $\bar{K}$ namely for every K and P if $C = E_K(P)$ then $\bar{C} = E_{\bar{K}}(\bar{P})$.

- Hence, $E_K(P) \oplus E_{\bar{K}}(\bar{P}) = C \oplus \bar{C}$

Deterministic Related-key Differential Cryptanalysis

- For the first time Deffie and Helmann introduced a deterministic relation between the encryption of DES under the **key K** and **related key** $\bar{K}$ namely for every K and P if $C = E_K(P)$ then $\bar{C} = E_{\bar{K}}(\bar{P})$.
- Hence, $E_K(P) \oplus E_{\bar{K}}(\bar{P}) = C \oplus \bar{C}$
- We denote the set $\mathcal{K}_0$ as the subspace of all n-bit keys in which the LSB is zero: $\mathcal{K}_0 = \{K \in \mathbb{F}_2^n, LSB(K) = 0\}$.

- Hence, $E_K(P) \oplus E_{\bar{K}}(\bar{P}) = C \oplus \bar{C}$

- We denote the set $\mathcal{K}_0$ as the subspace of all n-bit keys in which the LSB is zero: $\mathcal{K}_0 = \{K \in \mathbb{F}_2^n, LSB(K) = 0\}$.

1. Ask for the encryption of P and the encryption of $\bar{P}$ under an unknown key and save them as C and C^* respectively.

- Hence, $E_K(P) \oplus E_{\bar{K}}(\bar{P}) = C \oplus \bar{C}$

- We denote the set $\mathcal{K}_0$ as the subspace of all n-bit keys in which the LSB is zero: $\mathcal{K}_0 = \{K \in \mathbb{F}_2^n, LSB(K) = 0\}$.

1. Ask for the encryption of P and the encryption of $\bar{P}$ under an unknown key and save them as C and C^* respectively.

2. **For all** $K \in \mathcal{K}_0$
   - Compute $E_K(P)$; if it is equal to C, return K.
   - else, if $C \oplus \bar{C} = C^*$ return $\bar{K}$.

- Hence, $E_K(P) \oplus E_{\bar{K}}(\bar{P}) = C \oplus \bar{C}$

- We denote the set $\mathcal{K}_0$ as the subspace of all n-bit keys in which the LSB is zero: $\mathcal{K}_0 = \{K \in \mathbb{F}_2^n, LSB(K) = 0\}$.

1. Ask for the encryption of P and the encryption of $\bar{P}$ under an unknown key and save them as C and C^* respectively.

2. **For all** $K \in \mathcal{K}_0$
   - Compute $E_K(P)$; if it is equal to C, return K.
   - else, if $C \oplus \bar{C} = C^*$ return $\bar{K}$.

This property **decrease the security** of DES with **one bit**.

- Hence, we can exploit **related-key differential characteristics** to obtain the secret key **faster** than exhaustive search in the **single-key model**.

- Hence, we can exploit **related-key differential characteristics** to obtain the secret key **faster** than exhaustive search in the **single-key model**.
- In general, assume $E_K(P) \oplus E_{K \oplus \Delta}(P \oplus \Delta') = \Delta''$ holds for an arbitrary n-bit block cipher with an m-bit key.

- Hence, we can exploit **related-key differential characteristics** to obtain the secret key **faster** than exhaustive search in the **single-key model**.
- In general, assume $E_K(P) \oplus E_{K \oplus \Delta}(P \oplus \Delta') = \Delta''$ holds for an arbitrary n-bit block cipher with an m-bit key.

  We denote the set $\mathcal{K}_0$ as the subspace of all n-bit keys in which the LSB is zero: $\mathcal{K}_0 = \{K \in \mathbb{F}_2^n, LSB(K) = 0\}$.

- Hence, we can exploit **related-key differential characteristics** to obtain the secret key **faster** than exhaustive search in the **single-key model**.

- In general, assume $E_K(P) \oplus E_{K \oplus \Delta}(P \oplus \Delta') = \Delta''$ holds for an arbitrary n-bit block cipher with an m-bit key.

  We denote the set $\mathcal{K}_0$ as the subspace of all n-bit keys in which the LSB is zero: $\mathcal{K}_0 = \{K \in \mathbb{F}_2^n, LSB(K) = 0\}$.

1. Ask for the encryption of P and the encryption of $P^* = P \oplus \Delta'$ under an unknown key and save them as C and C^* respectively.

- Hence, we can exploit **related-key differential characteristics** to obtain the secret key **faster** than exhaustive search in the **single-key model**.

- In general, assume $E_K(P) \oplus E_{K \oplus \Delta}(P \oplus \Delta') = \Delta''$ holds for an arbitrary n-bit block cipher with an m-bit key.

  We denote the set $\mathcal{K}_0$ as the subspace of all n-bit keys in which the LSB is zero: $\mathcal{K}_0 = \{K \in \mathbb{F}_2^n, LSB(K) = 0\}$.

1. Ask for the encryption of P and the encryption of $P^* = P \oplus \Delta'$ under an unknown key and save them as C and C^* respectively.

2. **For all** $K \in \mathcal{K}_0$
   - Compute $E_K(P)$; if it is equal to C, return K.
   - else, if $C \oplus \Delta'' = C^*$ return $K \oplus \Delta$.

Higher-order Differential Characteristic

For a function $f : S \to T$, the **derivative** at a **point** $a_1 \in S$ is defined as

$$\Delta_{(a_1)} f(x) = f(x + a_1) - f(x).$$

Higher-order Differential Characteristic

For a function $f : S \to T$, the **derivative** at a **point** $a_1 \in S$ is defined as

$$\Delta_{(a_1)} f(x) = f(x + a_1) - f(x).$$

We can recursively define the i^{th} **derivate** of f as

$$\Delta_{a_1,\ldots,a_i} f(x) = \Delta_{a_i}(\Delta_{a_1,\ldots,a_{i-1}} f(x)).$$

Higher-order Differential Characteristic (Simplified)

Let us assume the values $a_1, ..., a_i$ be linearly independent.

$$\Delta_{a_1,...,a_i} f(x) = \sum_{c \in L(a_1,...,a_i)} f(x \oplus c),$$

where $L[a_1, \ldots, a_i]$ is the set of all 2^i possible linear combinations of $a_1, \ldots, a_i$.

Higher-order Differential Characteristic (Simplified)

Let us assume the values $a_1, ..., a_i$ be linearly independent.

$$\Delta_{a_1,...,a_i} f(x) = \sum_{c \in L(a_1,...,a_i)} f(x \oplus c),$$

where $L[a_1, \ldots, a_i]$ is the set of all 2^i possible linear combinations of $a_1, \ldots, a_i$.

If the **algebraic degree** of a **function** f with respect to the input x is at most d, then

$$\Delta_{a_1,...,a_{d+1}} f(x) = \sum_{c \in L(a_1,...,a_{d+1})} f(x \oplus c) = 0$$

The Challenge

How we can **utilize** Related-key Higher-Order Differential Distinguisher?

The Challenge

How we can **utilize** Related-key Higher-Order Differential Distinguisher?

- Let us assume that there exists a**related-key higher-order differential distinguisher** for a cipher with degree d.

The Challenge

How we can **utilize** Related-key Higher-Order Differential Distinguisher?

- Let us assume that there exists a**related-key higher-order differential distinguisher** for a cipher with degree d.
- The **key space** can simply be **partitioned** into 2^{k-d-1} sets of the same size 2^{d+1}.

The Challenge

How we can **utilize** Related-key Higher-Order Differential Distinguisher?

- Let us assume that there exists a**related-key higher-order differential distinguisher** for a cipher with degree d.
- The **key space** can simply be **partitioned** into 2^{k-d-1} sets of the same size 2^{d+1}.
- We compute the corresponding ciphertexts for $2^{d+1} - 1$ keys in one set.

- Then we obtain the corresponding ciphertext for the **remaining key** in the set by summation of the computed ciphertexts.

- Then we obtain the corresponding ciphertext for the **remaining key** in the set by summation of the computed ciphertexts.

- In this method, the adversary evaluate the **encryption function** $2^{d+1} - 1$ **times** under different keys and check **one** more possible key for **almost free**.

- Then we obtain the corresponding ciphertext for the **remaining key** in the set by summation of the computed ciphertexts.

- In this method, the adversary evaluate the **encryption function** $2^{d+1} - 1$ **times** under different keys and check **one** more possible key for **almost free**.

- The time complexity of the described method is $2^{k-d-1} \cdot (2^{d+1} - 1)$ **full encryptions** and $2^{k-d-1} \cdot 2^{d+1}$ **memory accesses**.

- Then we obtain the corresponding ciphertext for the **remaining key** in the set by summation of the computed ciphertexts.

- In this method, the adversary evaluate the **encryption function** $2^{d+1} - 1$ **times** under different keys and check **one** more possible key for **almost free**.

- The time complexity of the described method is $2^{k-d-1} \cdot (2^{d+1} - 1)$ **full encryptions** and $2^{k-d-1} \cdot 2^{d+1}$ **memory accesses**.

- Obviously, this conventional technique **cannot** be exploited to **beat exhaustive search**.

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	0000	Guess
1	0001	Guess
2	0010	Guess

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	0000	Guess
1	0001	Guess
2	0010	Guess
3	0011	

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	**0000**	Guess
1	**0001**	Guess
2	**0010**	Guess
3	**0011**	**0, 1, 2** free

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	0000	Guess
1	0001	Guess
2	0010	Guess
3	0011	0, 1, 2 free
4	0100	Guess

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	0000	Guess
1	0001	Guess
2	0010	Guess
3	0011	0, 1, 2 free
4	0100	Guess
5	0101	

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	**0000**	Guess
1	**0001**	Guess
2	0010	Guess
3	0011	0, 1, 2 free
4	**0100**	Guess
5	**0101**	**0, 1, 4** free

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	0000	Guess
1	0001	Guess
2	0010	Guess
3	0011	0, 1, 2 free
4	0100	Guess
5	0101	0, 1, 4 free
6	0110	

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	**0000**	Guess
1	0001	Guess
2	**0010**	Guess
3	0011	0, 1, 2 free
4	**0100**	Guess
5	0101	0, 1, 4 free
6	**0110**	**0, 2, 4** free

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	0000	Guess
1	0001	Guess
2	0010	Guess
3	0011	0, 1, 2 free
4	0100	Guess
5	0101	0, 1, 4 free
6	0110	0, 2, 4 free
7	0111	

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	0000	Guess
1	0001	Guess
2	0010	Guess
3	0011	0, 1, 2 free
4	**0100**	Guess
5	**0101**	0, 1, 4 free
6	**0110**	0, 2, 4 free
7	**0111**	**4, 5, 6** free

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	0000	Guess
1	0001	Guess
2	0010	Guess
3	0011	0, 1, 2 free
4	0100	Guess
5	0101	0, 1, 4 free
6	0110	0, 2, 4 free
7	0111	4, 5, 6 free
8	1000	Guess

Overview of our Technique

Target key x	Binary representation	Set of key generates the target key
0	0000	Guess
1	0001	Guess
2	0010	Guess
3	0011	0, 1, 2 free
4	0100	Guess
5	0101	0, 1, 4 free
6	0110	0, 2, 4 free
7	0111	4, 5, 6 free
8	1000	Guess
9	1001	**0, 1, 8** free
10	1010	**0, 2, 8** free
11	1011	**1, 2, 8** free
12	1100	**0, 4, 8** free
13	1101	**1, 4, 8** free
14	1110	**2, 4, 8** free
15	1111	**12, 13, 14** free

Our Algorithm

Input: Known plaintext–ciphertext pair P, C
Output: Speeding up the exhaustive search.

1: **for** $x = 0$ to $2^k - 1$ **do**
2: **if** $wh(K_x) < d + 1$ **then**
3: Compute $C_x = E_x(P)$
4: **if** $C = C_x$ **then**
5: Return x
6: **end if**
7: **else**
8: **Consider $e_{i_1}, e_{i_2}, \ldots, e_{i_d}$ such that $c_{i_j} \neq 0$ in the binary representation of x**, for $0 \leq j \leq d$
9: Compute $C' = \bigoplus_{m \in L(e_{i_1}, e_{i_2}, \ldots, e_{i_d}, x) \setminus \{x\}} C_m$
10: **if** $C = C_x$ **then**
11: Return x
12: **end if**
13: **end if**
14: **end for**

Complexity of the algorithm

- The **time complexity** of the mentioned algorithm is $\sum_{i=0}^{d} \binom{k}{i}$ full encryptions for the **guessed keys**.

Complexity of the algorithm

- The **time complexity** of the mentioned algorithm is $\sum_{i=0}^{d} \binom{k}{i}$ full encryptions for the **guessed keys**.
- $2^{d+1} - 1$ **memory accesses** for each of the **remaining keys**.

Complexity of the algorithm

- The **time complexity** of the mentioned algorithm is $\sum_{i=0}^{d} \binom{k}{i}$ full encryptions for the **guessed keys**.
- $2^{d+1} - 1$ **memory accesses** for each of the **remaining keys**.
- The number of required **guessed keys** is surprising **small** but it requires **large memory** (order of 2^k).

Complexity of the algorithm

- The **time complexity** of the mentioned algorithm is $\sum_{i=0}^{d} \binom{k}{i}$ full encryptions for the **guessed keys**.
- $2^{d+1} - 1$ **memory accesses** for each of the **remaining keys**.
- The number of required **guessed keys** is surprising **small** but it requires **large memory** (order of 2^k).
- The cost of a memory access directly depends on the size of memory.

Modified Algorithm

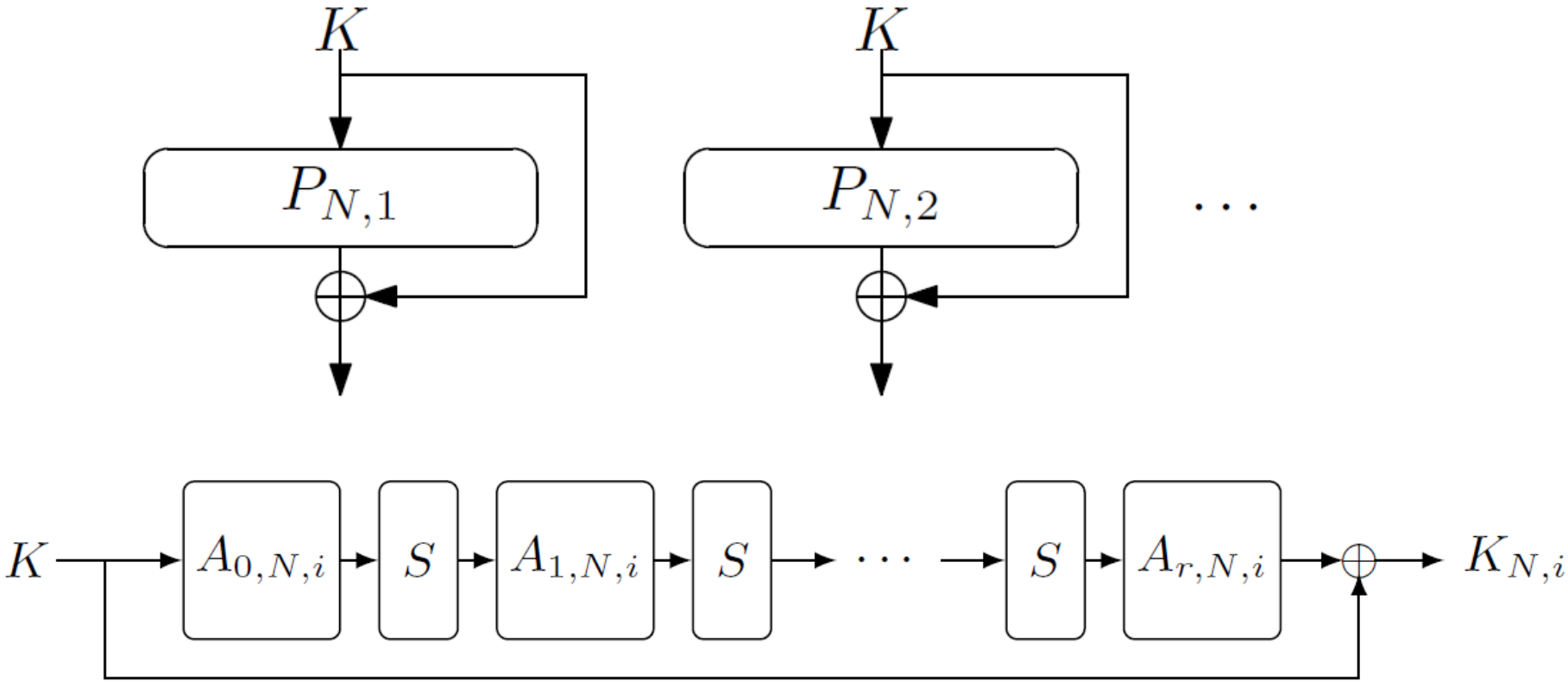
Input: Known plaintext–ciphertext pair P, C

Output: Speeding up the exhaustive search.

1: **for** $x = 0$ to $2^k - 1$ **do**
2: **if** $wh(K_x) < d + 1$ OR $wh(K_{xt}) < d + 1$ **then**
3: Compute $C_x = E_{K_x}(P)$
4: **if** $C = C'$ **then**
5: Return x
6: **end if**
7: **else**
8: Consider $e_{i_1}, e_{i_2}, \ldots, e_{i_d}$ such that $c_{i_j} \neq 0$ in the binary representation of x, for $0 \leq j \leq d$
9: Compute $C' = \bigoplus_{m \in L(e_{i_1}, e_{i_2}, \ldots, e_{i_d}, x) \setminus \{x\}} C_m$
10: **if** $C = C_x$ **then**
11: Return x
12: **end if**
13: **end if**
14: **end for**

Rasta and Agrasta

Presented at CRYPTO 2018



Results on Agrasta

Rounds	Degree	Data	t	Time		Memory
				Full encryption	MA per each obtaining key	
r	2^r	1	t	$2^{k-t}\sum_{i=0}^{d}\binom{t}{i}$	2^d	$O(2^{t+1})$
1	2	1	8	$2^{k-2.8}$	2^2	$O(2^9)$
2	4	1	12	$2^{k-2.36}$	2^4	$O(2^{13})$
2	4	1	16	$2^{k-4.71}$	2^4	$O(2^{17})$
3	8	1	20	$2^{k-2.1}$	2^8	$O(2^{21})$
3	8	1	24	$2^{k-3.78}$	2^8	$O(2^{25})$
4	16	1	35	$2^{k-1.83}$	2^{16}	$O(2^{36})$
4	16	1	40	$2^{k-3.12}$	2^{16}	$O(2^{41})$

Conclusion

- The effect of related-key cryptanalysis in the single-key model has been a long-standing open question.
- In this work we present a general formalization for utilizing related-key higher-order differential characteristic to assess the precise security of cryptographic primitives in the single-key model.
- The effectiveness of our method directly depends on the degree of the function.