

Information-Combining Differential Fault Attacks on DEFAULT

Marcel Nageler Christoph Dobraunig Maria Eichlseder

FSE Workshop 2022

☰ Outline

igsaw puzzle piece icon Background

link icon Attacks on Simple Key Schedule

double equals sign icon Equivalent Keys

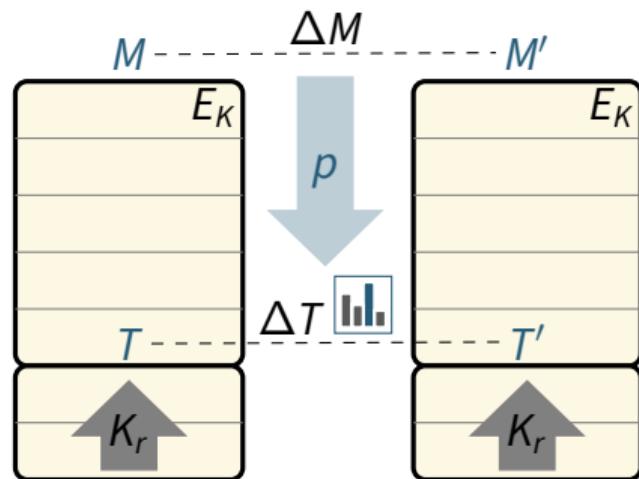
link icon Attacks on Strong Key Schedule

Background

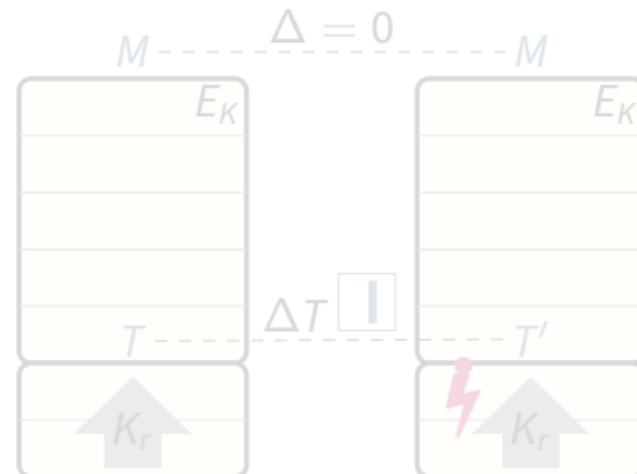


How DEFAULT protects against DFA

Differential Fault Attacks (DFA)

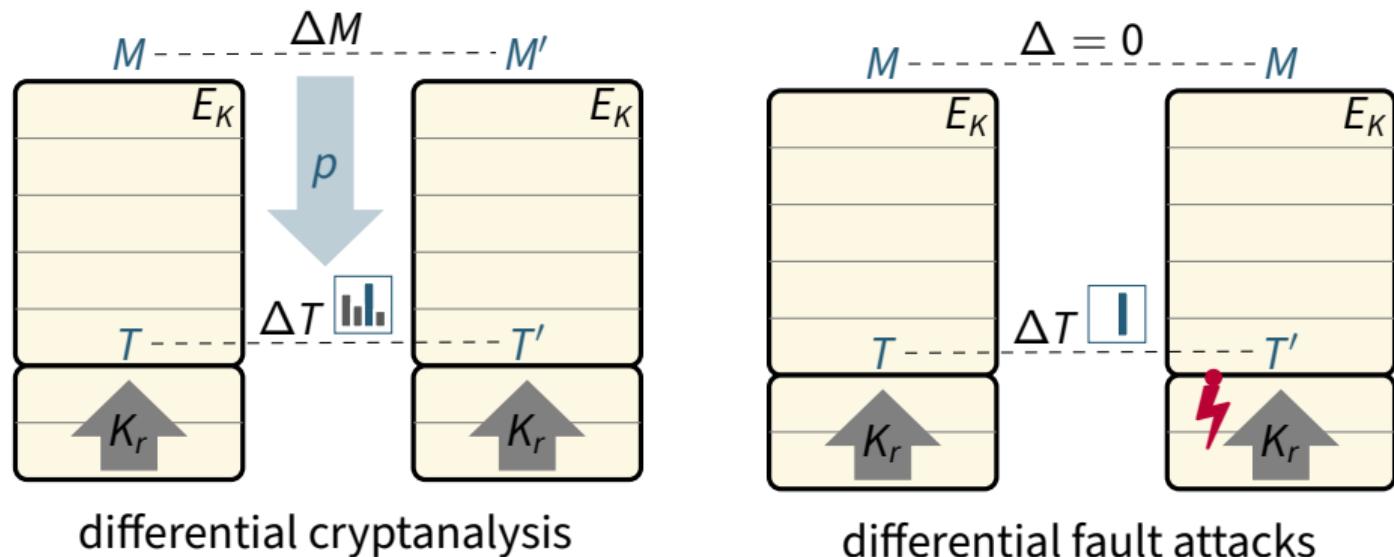


differential cryptanalysis

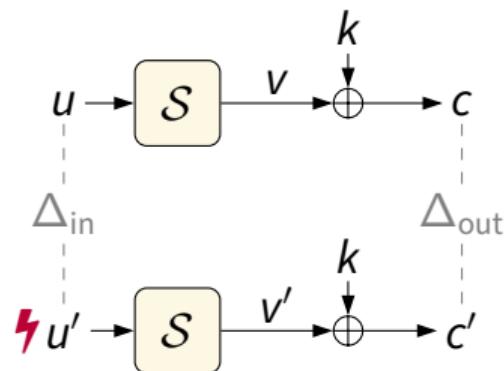


differential fault attacks

Differential Fault Attacks (DFA)



Differential Fault Attacks (DFA)



- Transition $\Delta_{in} \rightarrow \Delta_{out}$ limits (u, v) and k .
- DDT $[\Delta_{in}, \Delta_{out}]$ defines number of solutions.
- Thus, small entries benefit attacker.

DEFAULT

- Design Strategy by Baksi et al. at ASIACRYPT 2021 [BBB+21b]
- Intrinsic resistance against DFA thanks to special S-boxes with linear structures
- Block Cipher DEFAULT:
 - 128-bit classical security
 - 64-bit DFA security
 - simple [BBB+21a], and strong key schedule [BBB+21b]

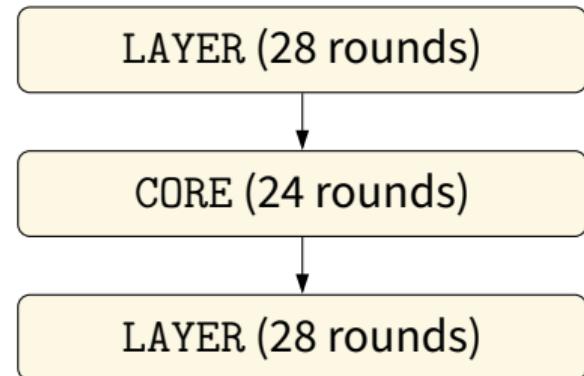


Figure: Design of DEFAULT

DEFAULT

- Design Strategy by Baksi et al. at ASIACRYPT 2021 [BBB+21b]
- Intrinsic resistance against DFA thanks to special S-boxes with linear structures
- Block Cipher DEFAULT:
 - 128-bit classical security
 - 64-bit DFA security
 - simple [BBB+21a], and strong key schedule [BBB+21b]

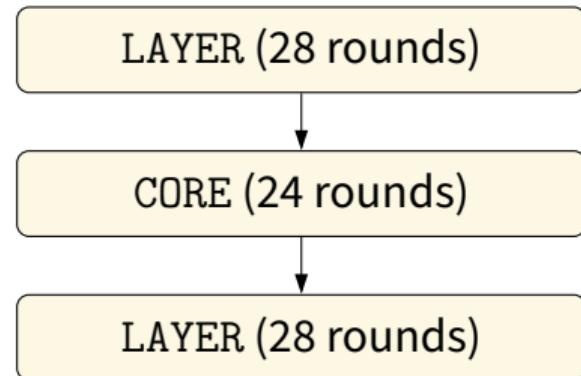


Figure: Design of DEFAULT

DEFAULT

- Design Strategy by Baksi et al. at ASIACRYPT 2021 [BBB+21b]
- Intrinsic resistance against DFA thanks to special S-boxes with linear structures
- Block Cipher DEFAULT:
 - 128-bit classical security
 - 64-bit DFA security
 - simple [BBB+21a], and strong key schedule [BBB+21b]

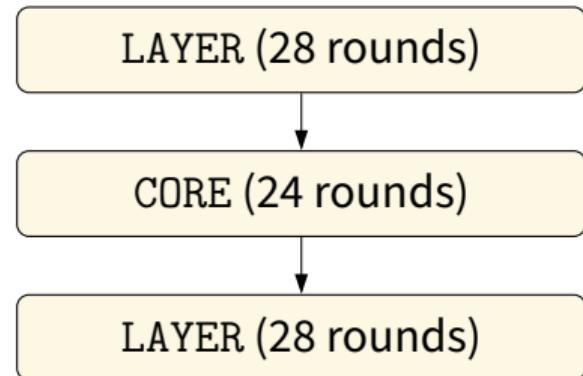


Figure: Design of DEFAULT

DEFAULT

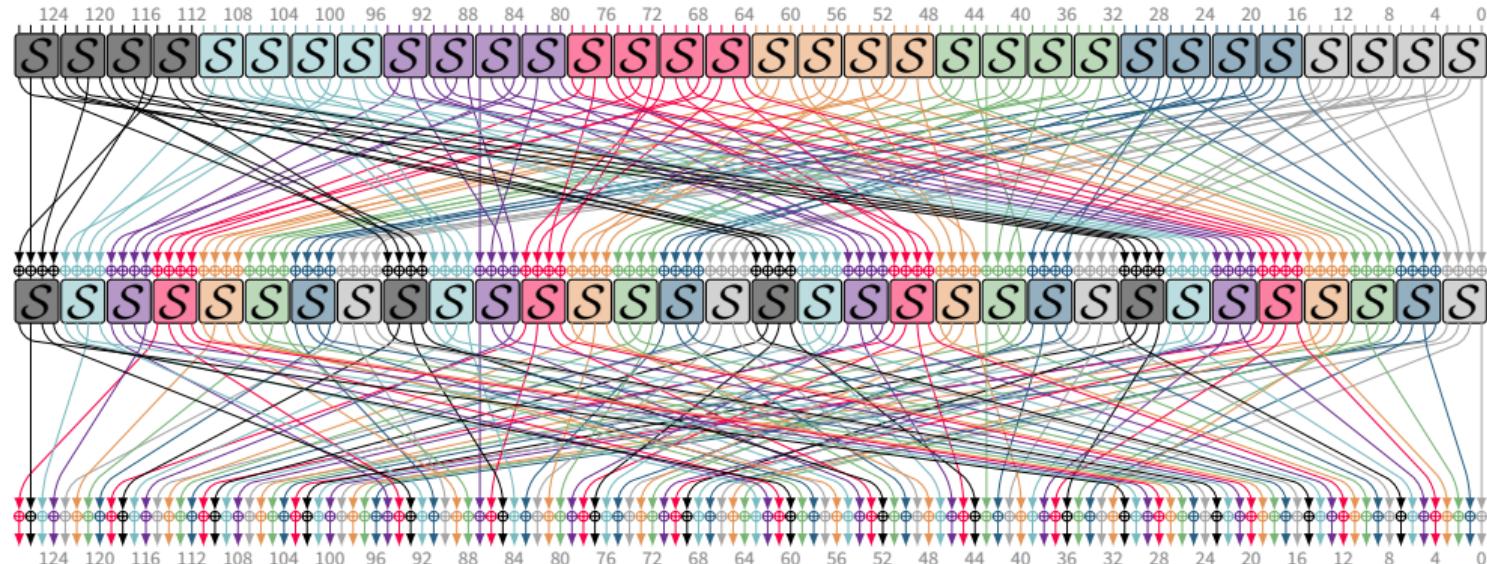
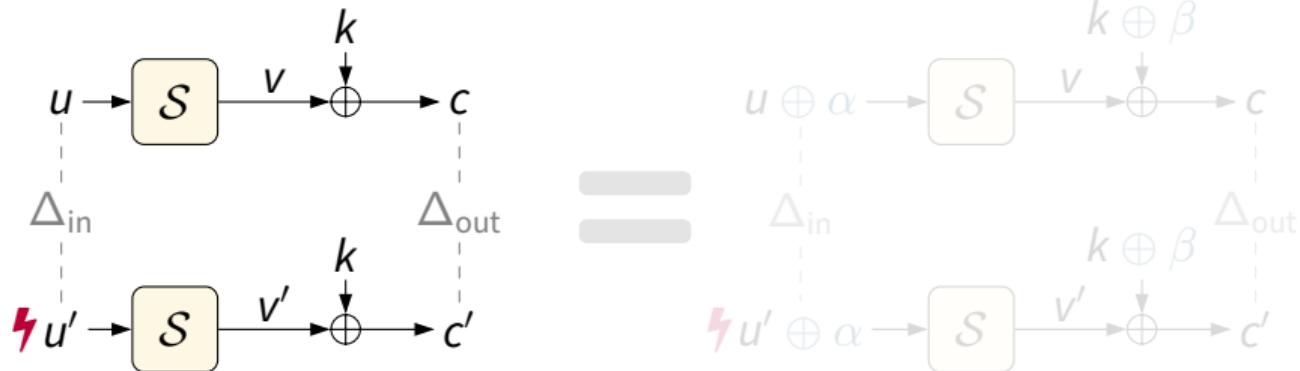


Figure: Two rounds of DEFAULT-LAYER, illustrating the S-box grouping.

Table: Differential Distribution Table of the S-box \mathcal{S} with linear structures

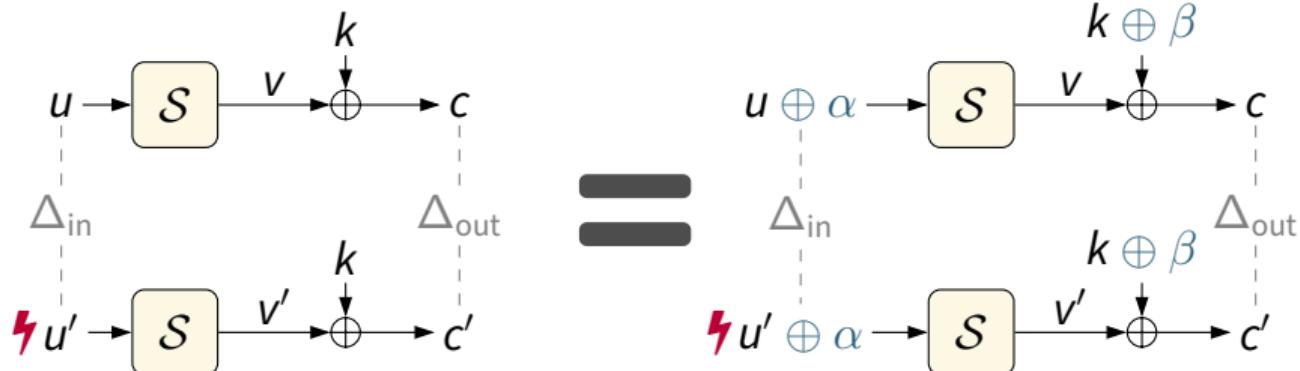
$\Delta_{\text{in}} \setminus \Delta_{\text{out}}$	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16
1	.	.	.	8	8
2	8	8	.	.
3	8	8	.
4	8	8	.	.
5	8	8	.	.
6	16
7	.	.	.	8	8
8	8	8	.	.	.
9	16
a	.	8	8
b	.	.	8	8
c	.	8	8
d	.	.	8	8
e	8	8
f	16

DFA: Effect of Linear Structures [BBB+21b]



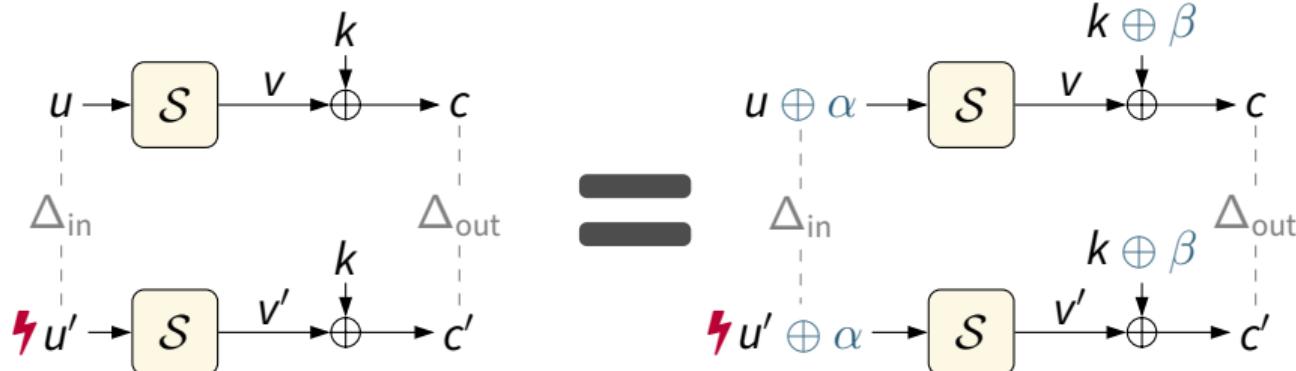
- Each linear structure $\alpha \rightarrow \beta$ leads to indistinguishable keys.
- If k is a valid key so is $k \oplus \beta$.
- 4 linear structures per S-box $\rightarrow 4^{32} = 2^{64}$ candidates

DFA: Effect of Linear Structures [BBB+21b]



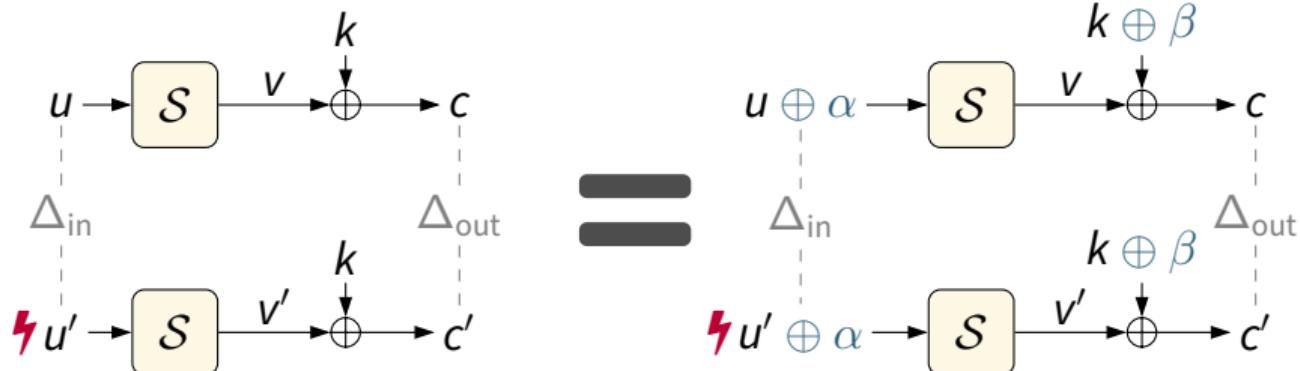
- Each linear structure $\alpha \rightarrow \beta$ leads to indistinguishable keys.
- If k is a valid key so is $k \oplus \beta$.
- 4 linear structures per S-box $\rightarrow 4^{32} = 2^{64}$ candidates

DFA: Effect of Linear Structures [BBB+21b]



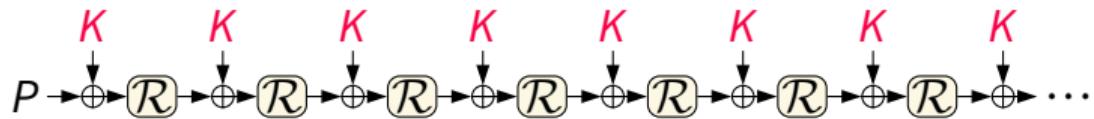
- Each linear structure $\alpha \rightarrow \beta$ leads to indistinguishable keys.
- If k is a valid key so is $k \oplus \beta$.
- 4 linear structures per S-box $\rightarrow 4^{32} = 2^{64}$ candidates

DFA: Effect of Linear Structures [BBB+21b]



- Each linear structure $\alpha \rightarrow \beta$ leads to indistinguishable keys.
- If k is a valid key so is $k \oplus \beta$.
- 4 linear structures per S-box $\rightarrow 4^{32} = 2^{64}$ candidates

Attacks on Simple Key Schedule



Attack on Simple Key Schedule



- 💡 Intuition behind simple key schedule: learn same information each round
- ❗ Combine (linear) information across rounds
- 📋 Three step attack
 - 1. Attack 1 round → 2^{64} keys remain
 - 2. Attack 2 rounds → 2^{32} keys remain
 - 3. Attack 3 rounds → 2^{16} keys remain

Attack on Simple Key Schedule



- 💡 Intuition behind simple key schedule: learn same information each round
- ❗ Combine (linear) information across rounds
- 📋 Three step attack
 - 1. Attack 1 round → 2^{64} keys remain
 - 2. Attack 2 rounds → 2^{32} keys remain
 - 3. Attack 3 rounds → 2^{16} keys remain

Attack on Simple Key Schedule



- 💡 Intuition behind simple key schedule: learn same information each round
- ❗ Combine (linear) information across rounds
- ☰ Three step attack
 - 1. Attack 1 round → 2^{64} keys remain
 - 2. Attack 2 rounds → 2^{32} keys remain
 - 3. Attack 3 rounds → 2^{16} keys remain

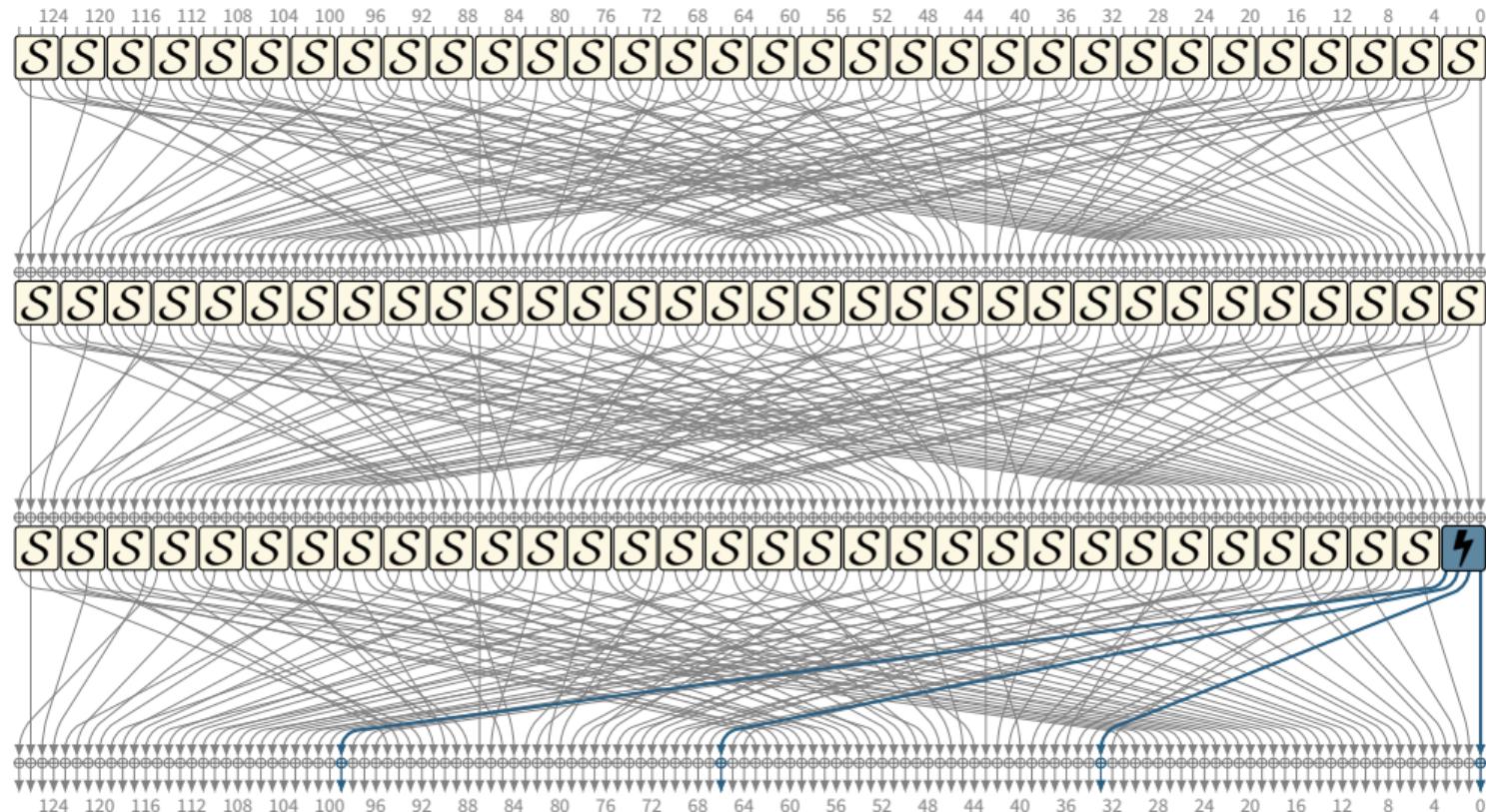


Figure: Attack on Simple Key Schedule ↗ (Step 1: 2^{64} keys remain)

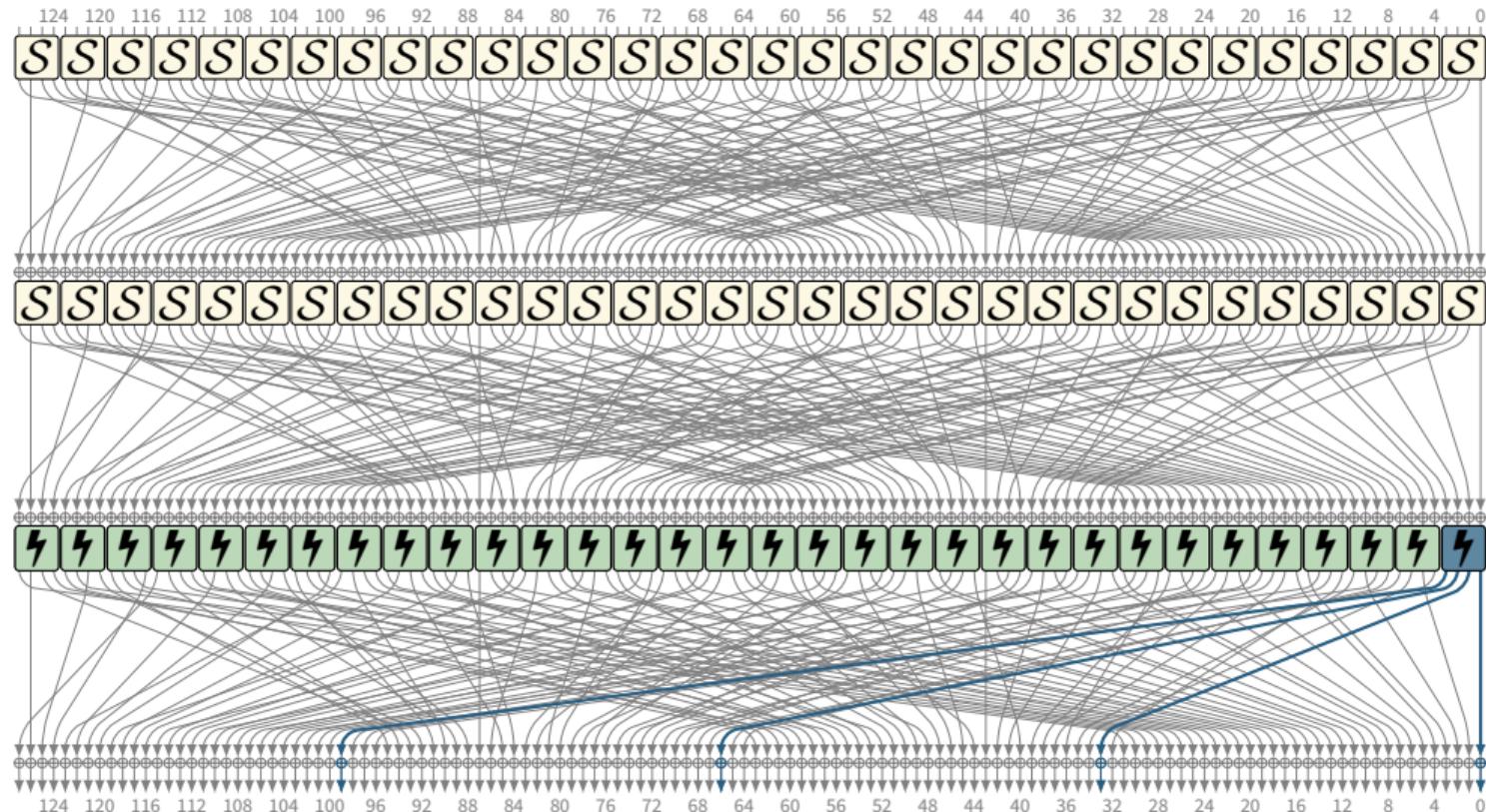


Figure: Attack on Simple Key Schedule ↗ (Step 1: 2^{64} keys remain)



Figure: Attack on Simple Key Schedule (Step 2: 2^{32} keys remain)

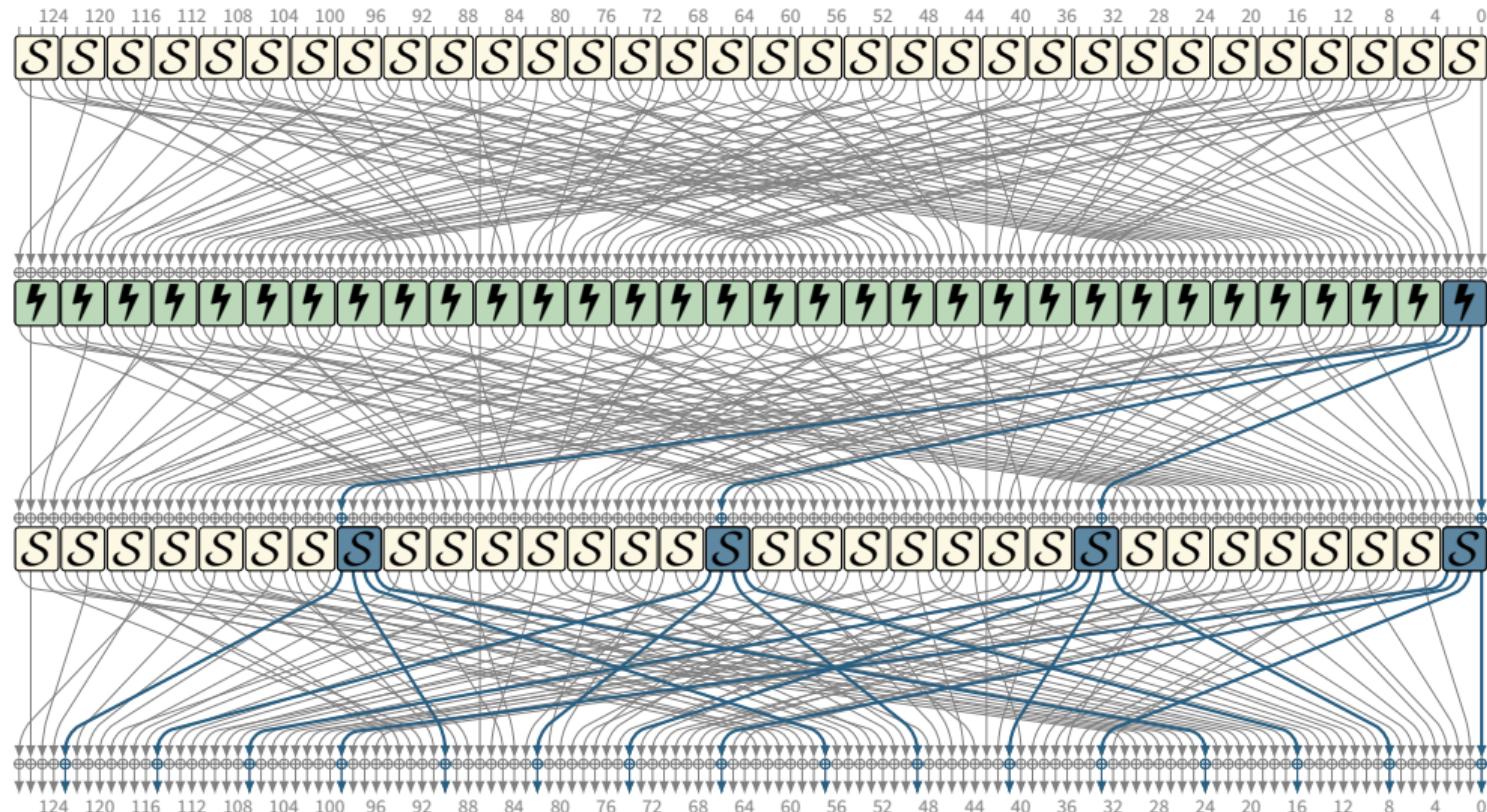


Figure: Attack on Simple Key Schedule ↗ (Step 2: 2^{32} keys remain)

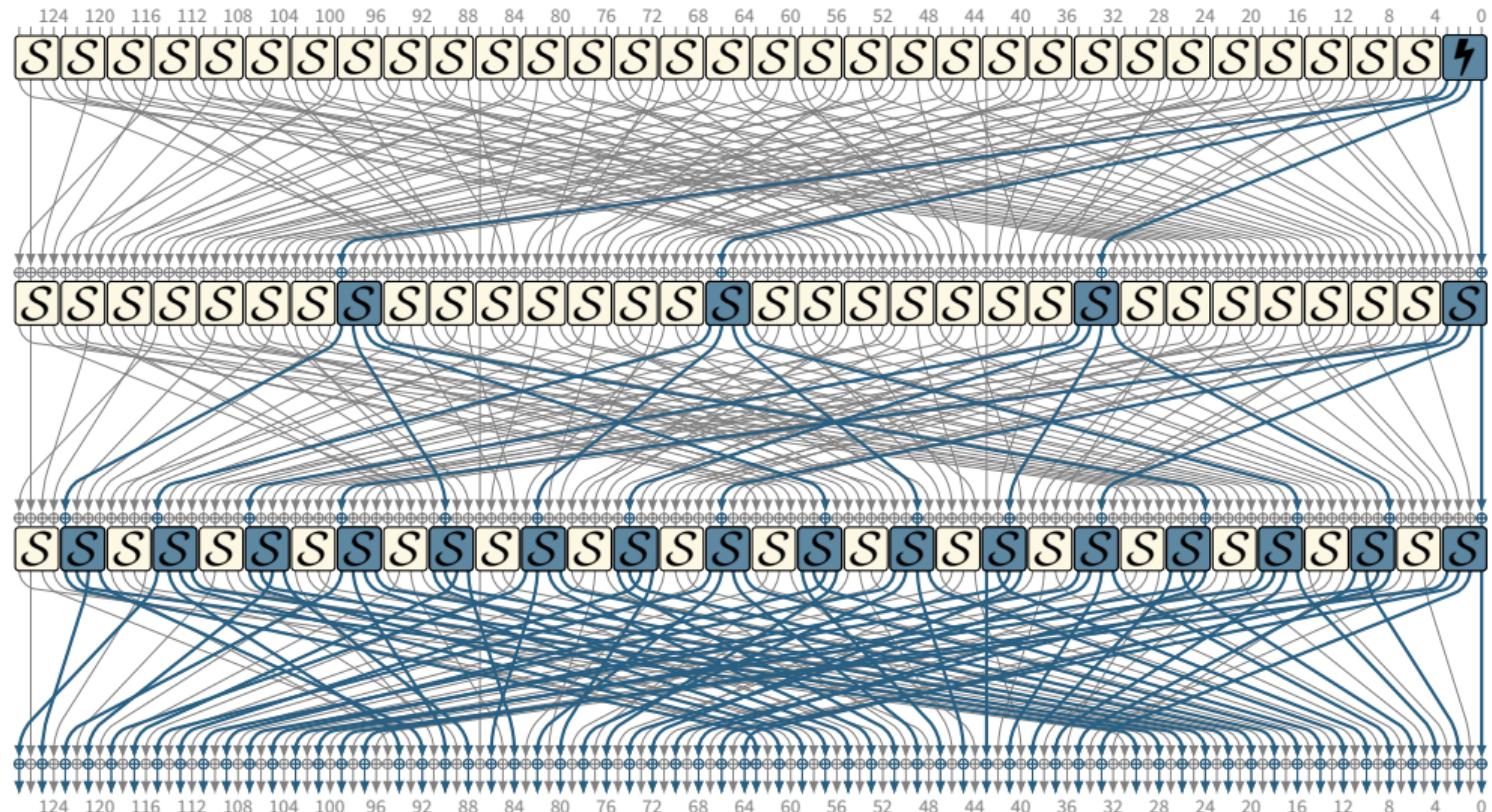


Figure: Attack on Simple Key Schedule (Step 3: 2^{16} keys remain)

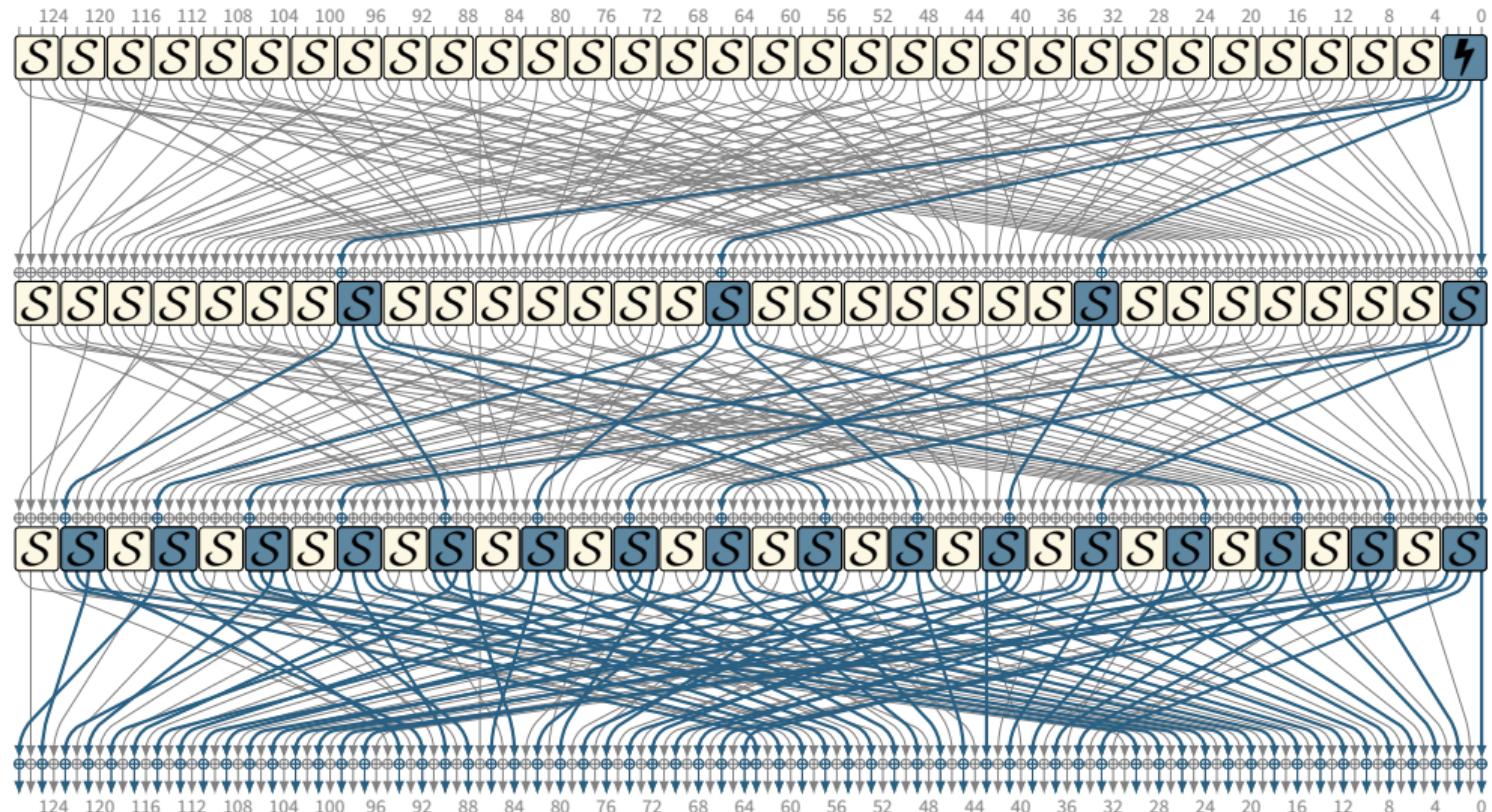


Figure: Attack on Simple Key Schedule ↗ (Step 3: 2^{16} keys remain)

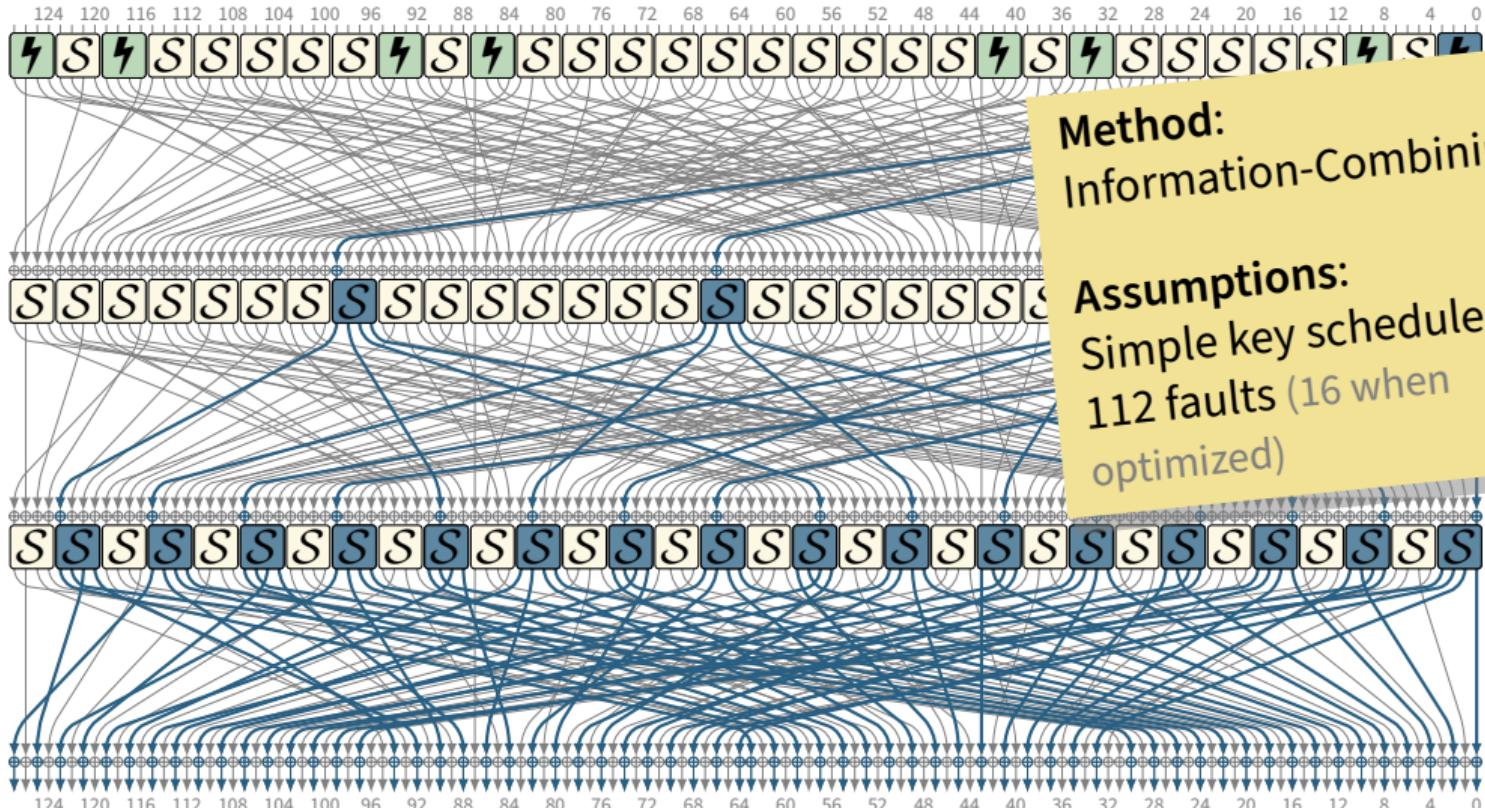


Figure: Attack on Simple Key Schedule (Step 3: 2^{16} keys remain)

Equivalent Keys



What if consecutive keys are independent?

Recall: Effect of Linear Structures [BBB+21b]

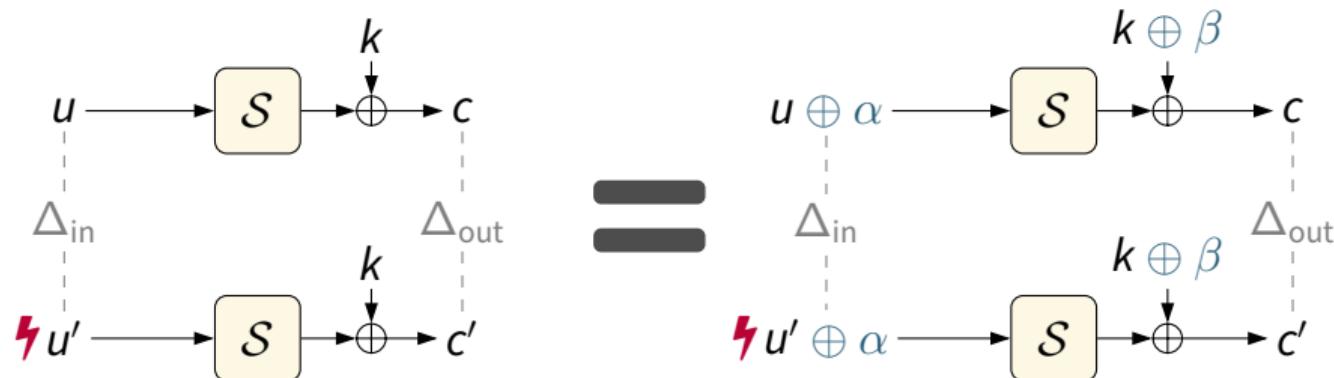


Figure: Each Linear Structure $\alpha \rightarrow \beta$ leads to indistinguishable keys

Linear Structures Imply Equivalent Keys



Figure: Each Linear Structure $\alpha \rightarrow \beta$ leads to equivalent keys

Linear Structures Imply Equivalent Keys

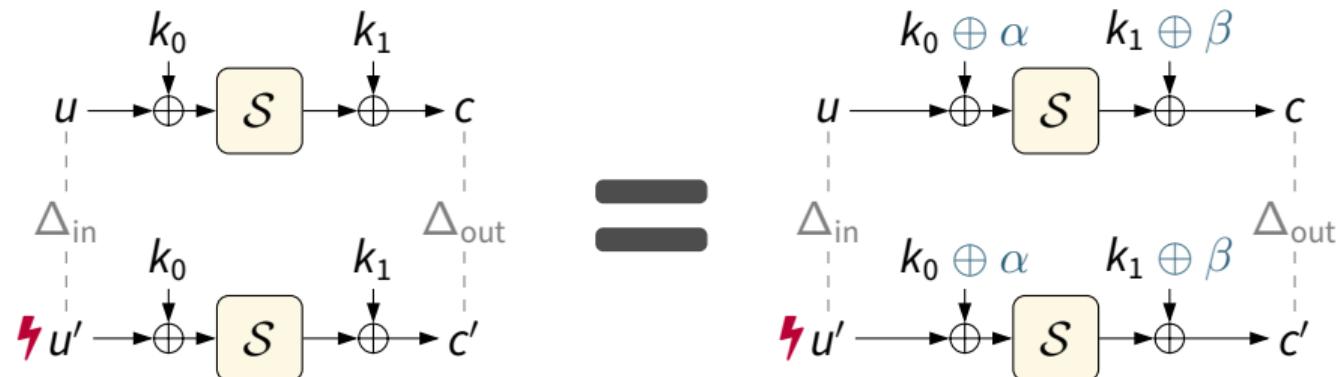


Figure: Each Linear Structure $\alpha \rightarrow \beta$ leads to equivalent keys

Equivalent Keys: Multiple Rounds

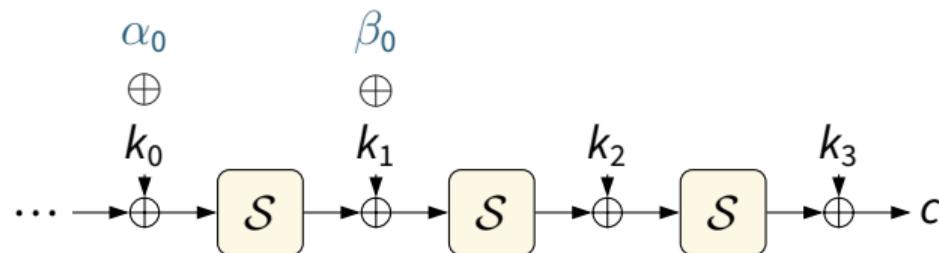


Figure: Equivalent keys for a toy cipher.

- = Group keys into equivalence classes,
each corresponding to identical behavior.
- 💡 Idea: Pick a representative \bar{k} from each equivalence class

Equivalent Keys: Multiple Rounds

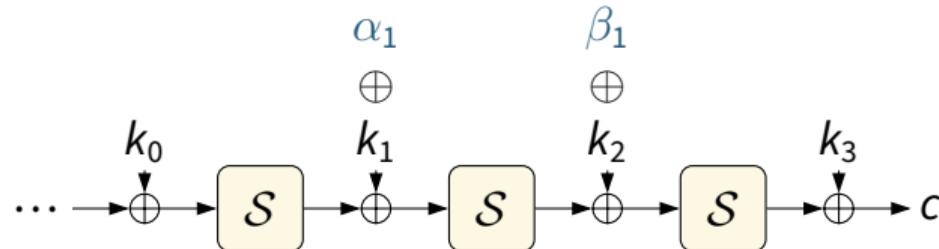


Figure: Equivalent keys for a toy cipher.

- = Group keys into equivalence classes,
each corresponding to identical behavior.
- 💡 Idea: Pick a representative \bar{k} from each equivalence class

Equivalent Keys: Multiple Rounds

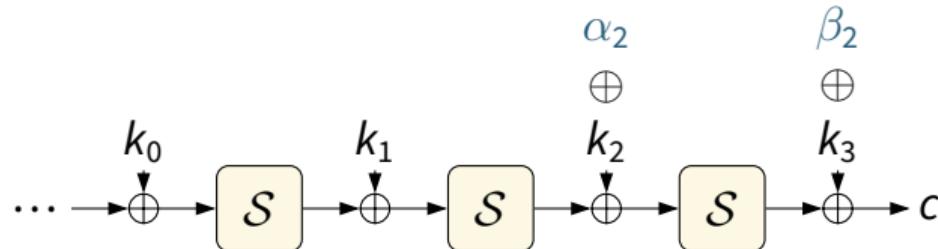


Figure: Equivalent keys for a toy cipher.

- = Group keys into equivalence classes,
each corresponding to identical behavior.
- 💡 Idea: Pick a representative \bar{k} from each equivalence class

Equivalent Keys: Multiple Rounds

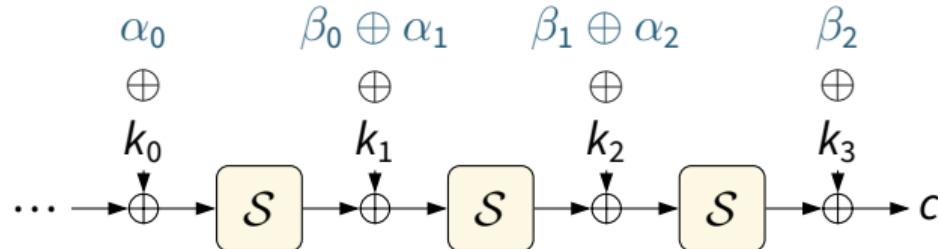


Figure: Equivalent keys for a toy cipher.

- = Group keys into equivalence classes,
each corresponding to identical behavior.
- 💡 Idea: Pick a representative \bar{k} from each equivalence class

Equivalent Keys: Multiple Rounds

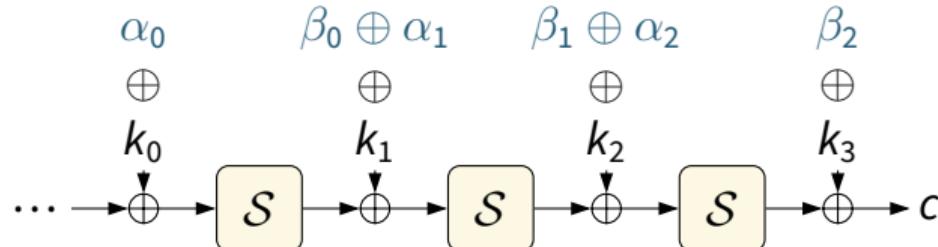


Figure: Equivalent keys for a toy cipher.

- = Group keys into equivalence classes, each corresponding to identical behavior.
- 💡 Idea: Pick a representative \bar{k} from each equivalence class

Equivalent Keys: Multiple Rounds

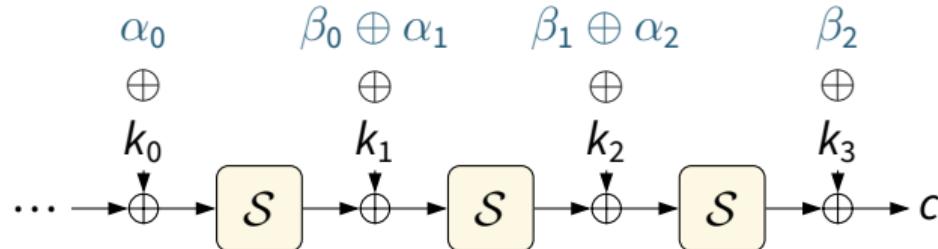


Figure: Equivalent keys for a toy cipher.

- = Group keys into equivalence classes, each corresponding to identical behavior.
- 💡 Idea: Pick a representative \bar{k} from each equivalence class

Normalized Keys

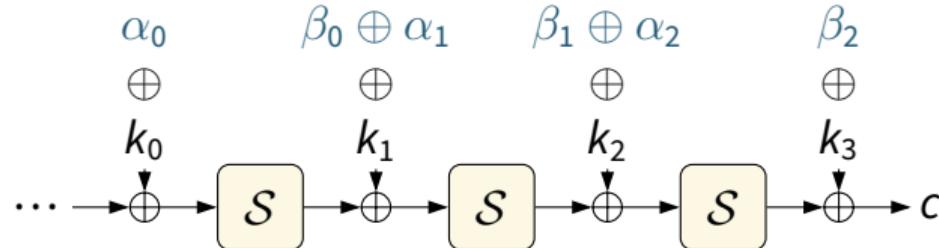


Figure: Equivalent keys for a toy cipher.

- ?
- How to pick representatives?
- ⟳ Reduce candidates for initial key guesses
- ⚡ For n round keys, $n - 1$ normalized keys can be recovered uniquely

Normalized Keys

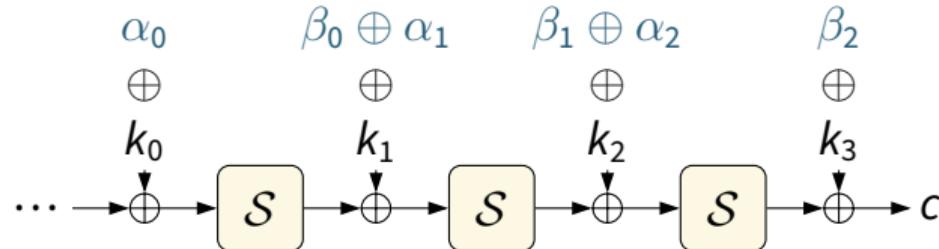


Figure: Equivalent keys for a toy cipher.

- ? How to pick representatives?
- ➡ Reduce candidates for initial key guesses
- ⚡ For n round keys, $n - 1$ normalized keys can be recovered uniquely

Normalized Keys

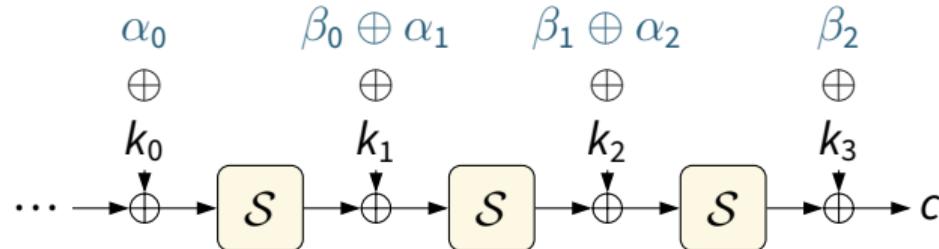


Figure: Equivalent keys for a toy cipher.

- ? How to pick representatives?
- ➡ Reduce candidates for initial key guesses
- ⚡ For n round keys, $n - 1$ normalized keys can be recovered uniquely

Equivalent Keys in DEFAULT

- 4 linear structures $\alpha \rightarrow \beta$ per S-box
 - $0 \rightarrow 0, 6 \rightarrow a, 9 \rightarrow f, f \rightarrow 5$
- 2^{64} linear structures per round
- For n independent keys
 - There are 2^{128n} keys
 - but we get 2^{64n+64} equivalence classes (distinct keyed permutations)

Equivalent Keys in DEFAULT

- 4 linear structures $\alpha \rightarrow \beta$ per S-box
 - $0 \rightarrow 0, 6 \rightarrow a, 9 \rightarrow f, f \rightarrow 5$
- 2^{64} linear structures per round
- For n independent keys
 - There are 2^{128n} keys
 - but we get 2^{64n+64} equivalence classes (distinct keyed permutations)

Equivalent Keys in DEFAULT

- 4 linear structures $\alpha \rightarrow \beta$ per S-box
 - $0 \rightarrow 0, 6 \rightarrow a, 9 \rightarrow f, f \rightarrow 5$
- 2^{64} linear structures per round
- For n independent keys
 - There are 2^{128n} keys
 - but we get 2^{64n+64} equivalence classes (distinct keyed permutations)

Example: DEFAULT

$K_0 : 922d8799645a197240612627adac008c$

$K_1 : fd034fb83d3f82087ecb3d36ebd5b311$

$K_2 : 6c6ebe434de58a603140168a0cbcea2f$

$K_3 : ab6472a5fc49ba97a6504da4acaa8113$

(a) Sequence of round keys K .

$\bar{K}_0 : 02221100023310122001202132330013$

$\bar{K}_1 : 22312310332022020103310210031312$

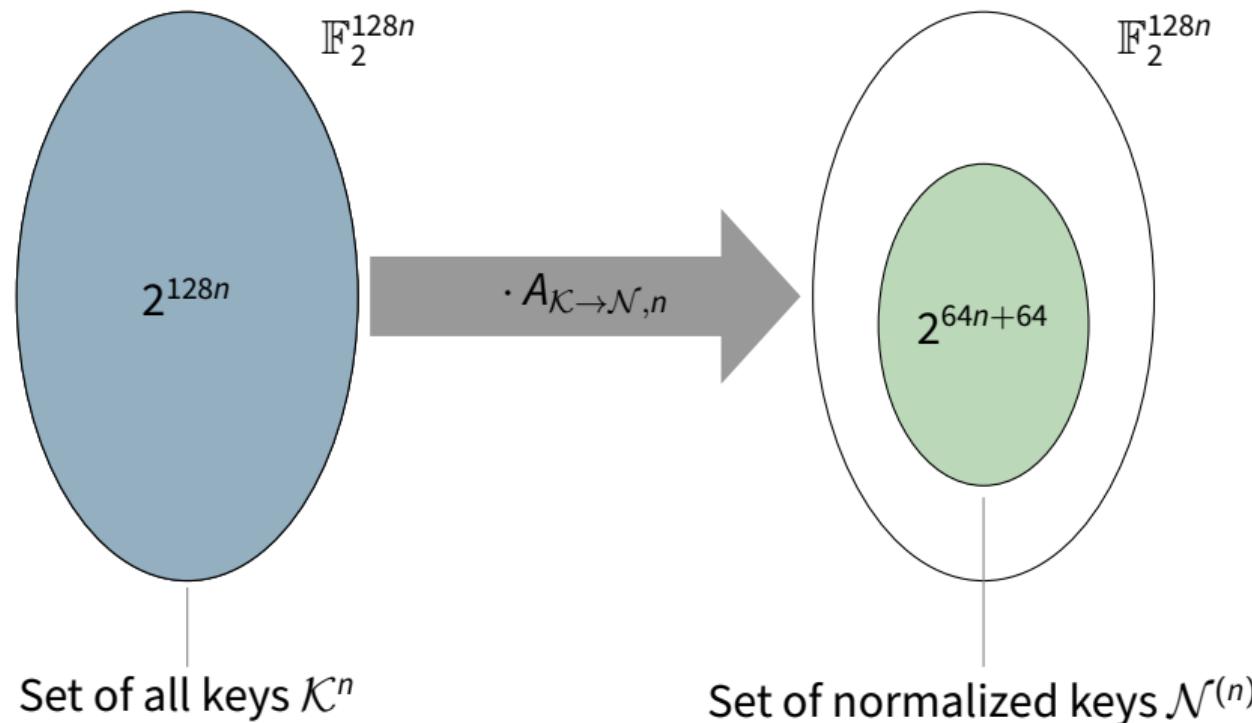
$\bar{K}_2 : 31012322123322300020111133332110$

$\bar{K}_3 : 1f95f3c6f75987f847a46938a2ea468c$

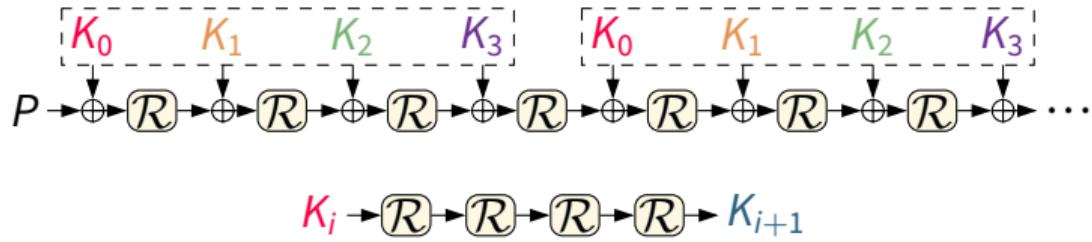
(b) Equivalent normalized key \bar{K} .

Figure: Sequence of round keys and its normalized equivalent.

Normalizing Keys is a Linear Operation

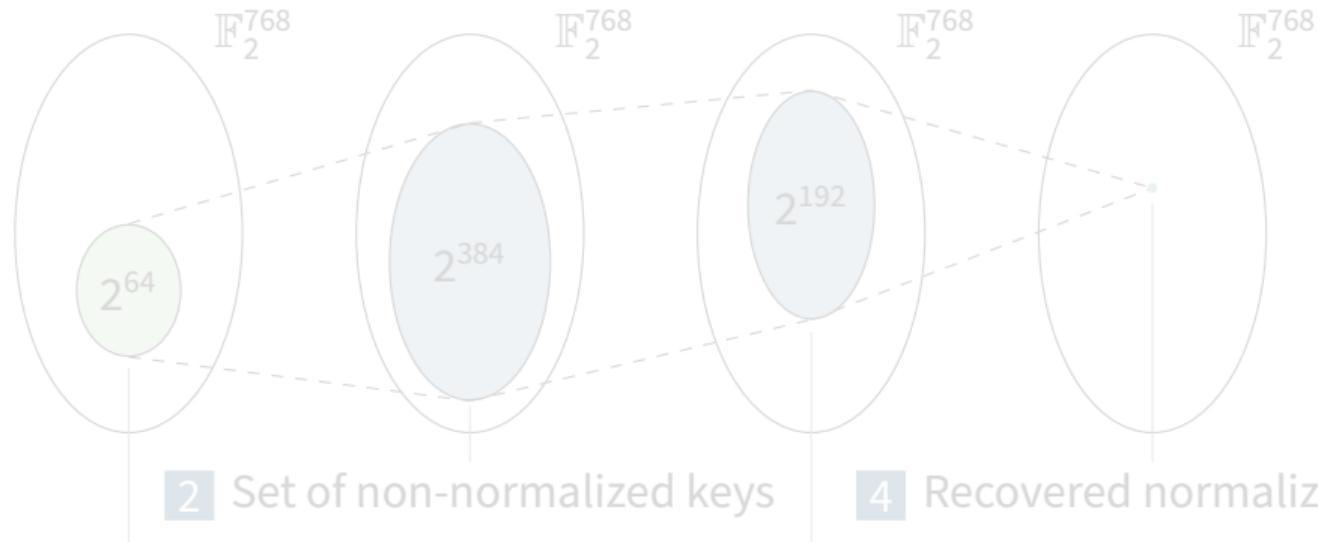


Attacks on Strong Key Schedule



Linear Spaces in the Attack on DEFAULT-LAYER

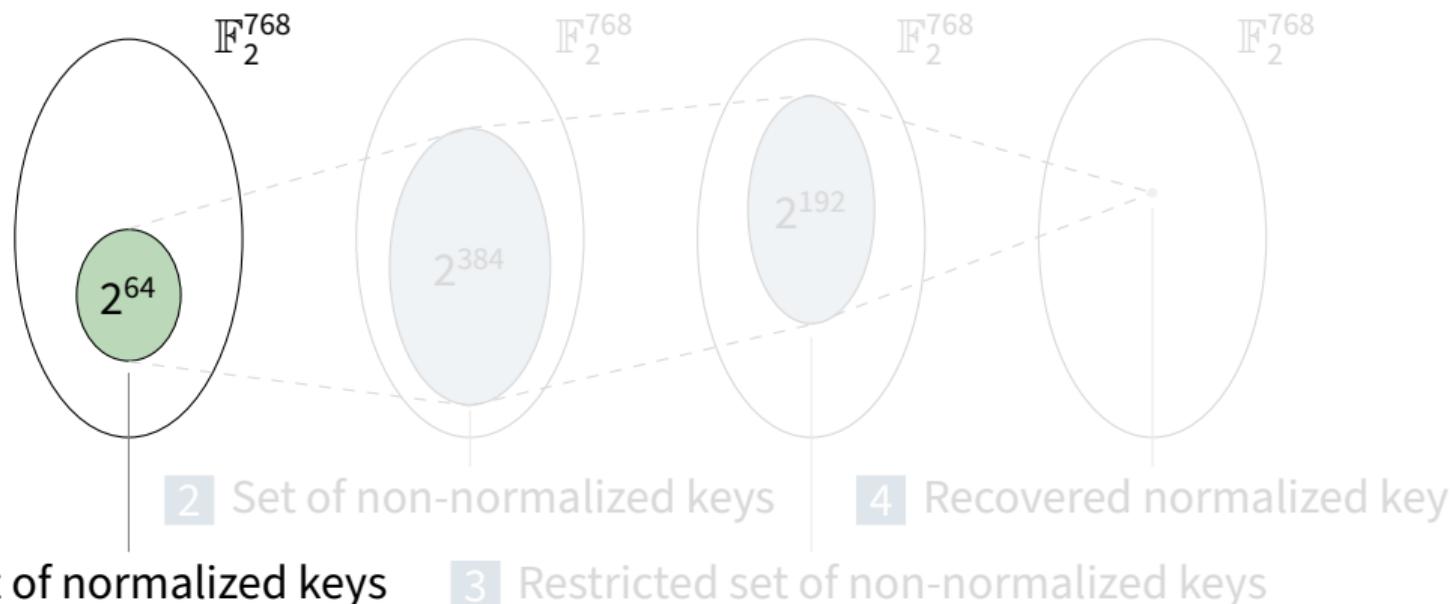
💡 Pretend there are 6 independent keys K_0, \dots, K_5



1 Set of normalized keys 3 Restricted set of non-normalized keys

Linear Spaces in the Attack on DEFAULT-LAYER

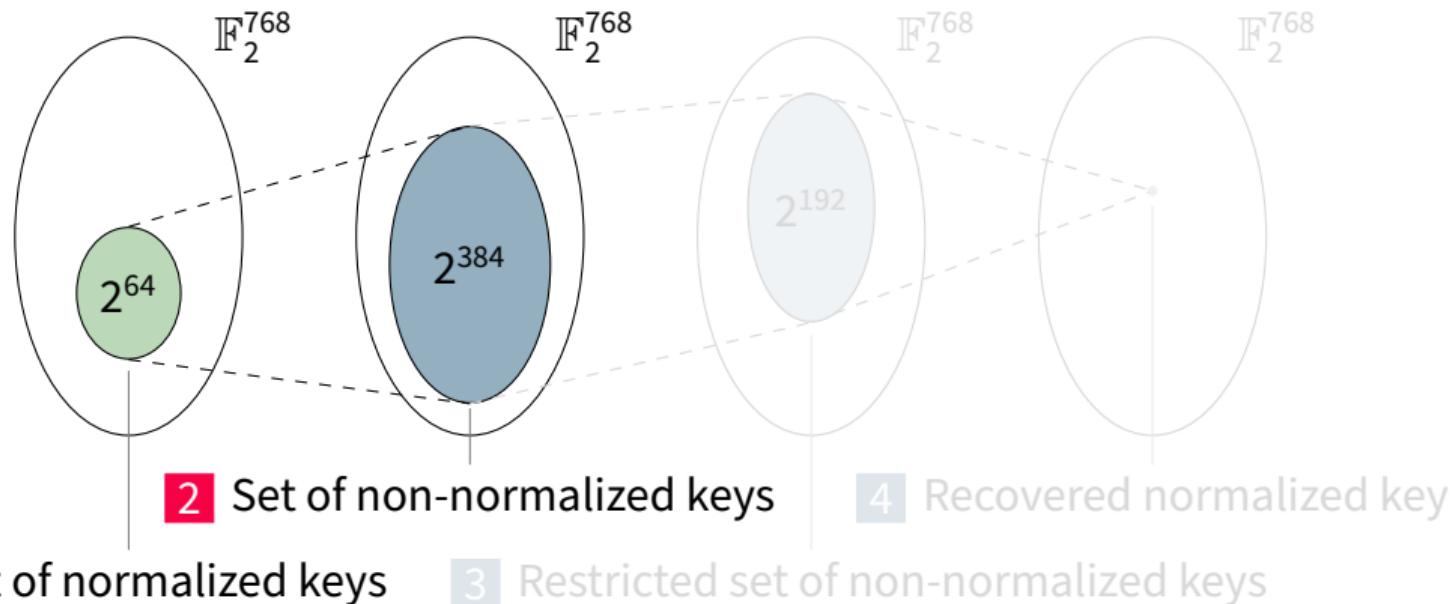
- 1 Recover 2^{64} normalized keys $\bar{K}_0, \dots, \bar{K}_5$ using DFA



- 1 Set of normalized keys
- 3 Restricted set of non-normalized keys

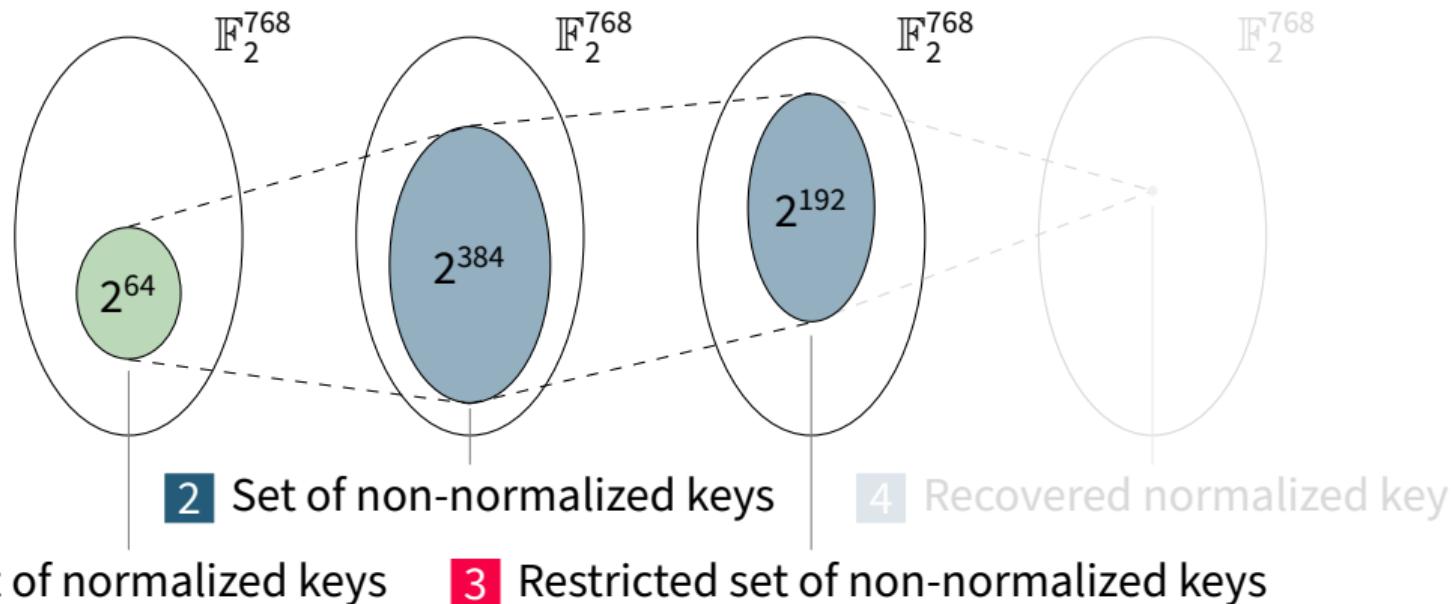
Linear Spaces in the Attack on DEFAULT-LAYER

2 Convert to linear space of non-normalized keys



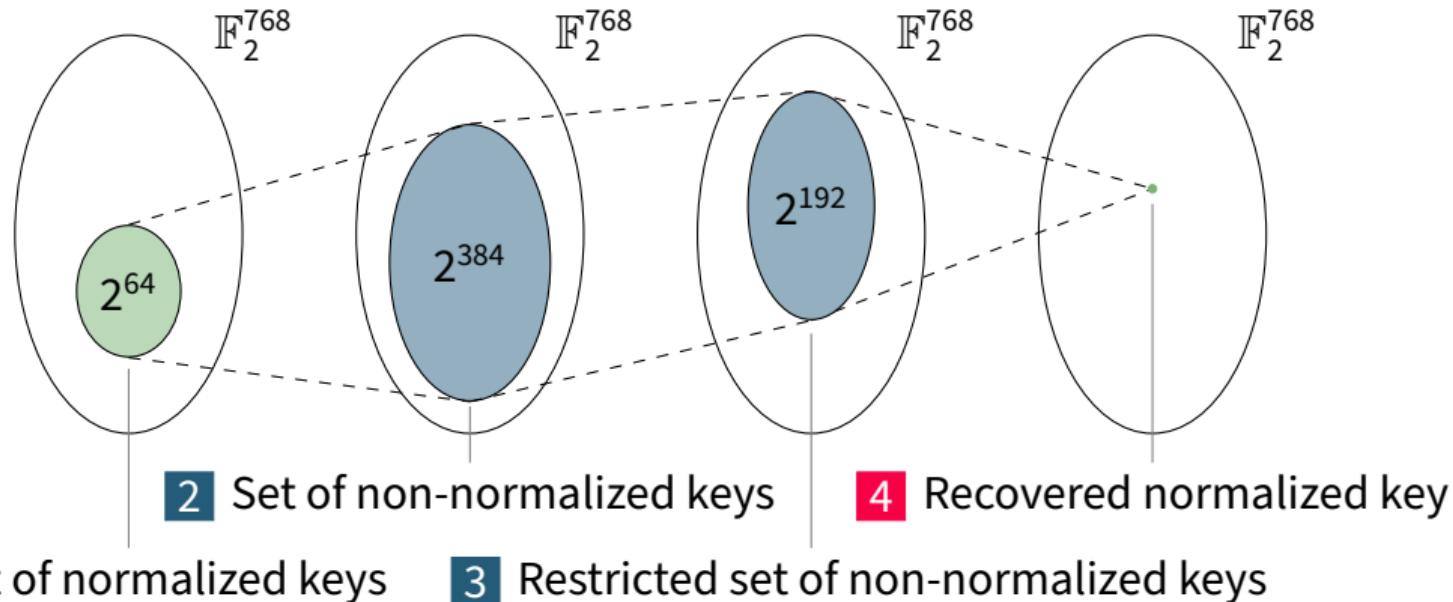
Linear Spaces in the Attack on DEFAULT-LAYER

- 3 Add condition that $K_0 = K_4$ and $K_1 = K_5$



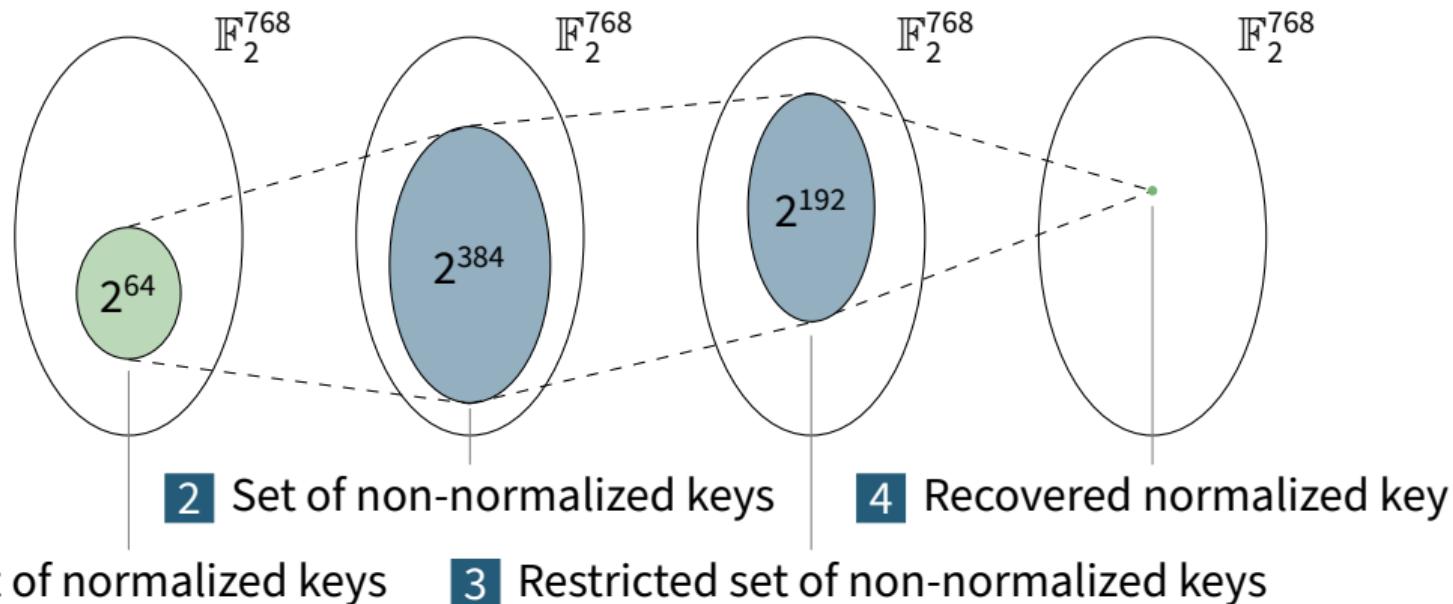
Linear Spaces in the Attack on DEFAULT-LAYER

- 4 Normalize the remaining keys to get a single key



Linear Spaces in the Attack on DEFAULT-LAYER

5 Attack DEFAULT-CORE using classical DFA



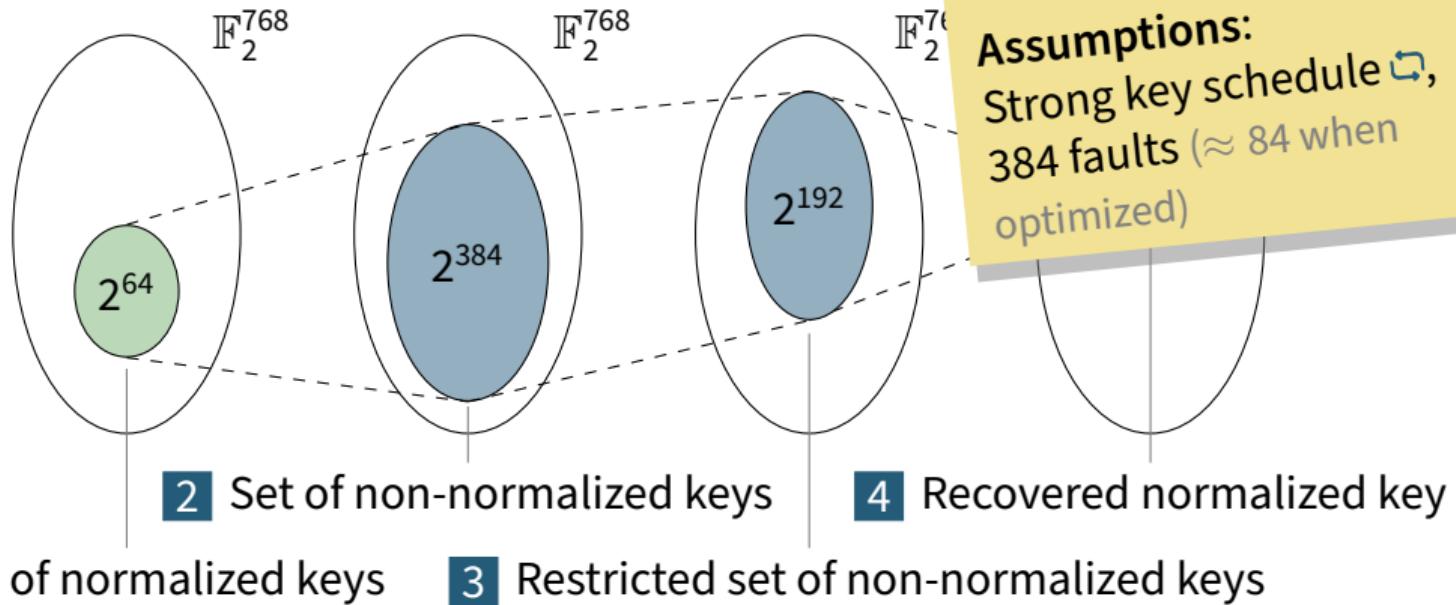
1 Set of normalized keys 3 Restricted set of non-normalized keys

Linear Spaces in the Attack on DEFAULT-LAYER

5 Attack DEFAULT-CORE using classical DFA

Method:
Normalized Keys,
Information-Combining

Assumptions:
Strong key schedule ↗,
384 faults (≈ 84 when
optimized)



Summary of Attacks on DEFAULT-LAYER

Approach	Faults	Time	Key Schedule		
			simple	strong	ideal
DFA [BBB+21b]	64	2^{64}			
Enc-Dec IC-DFA	16	$\leq 2^{39}$	✓		
Multi-round IC-DFA	16	$\leq 2^{20}$	✓		
Generic NK DFA	1728	2^0	✓	✓	✓
Enc-Dec IC-NK-DFA	288	2^{32}	✓*	✓	
Multi-round IC-NK-DFA	84 ± 15	2^0	✓*	✓	

Summary of Attacks on DEFAULT-LAYER

Approach	Faults	Time	Key Schedule		
			simple	strong	ideal
DFA [BBB+21b]	64	2^{64}			
Enc-Dec IC-DFA	16	$\leq 2^{39}$	✓		
Multi-round IC-DFA	16	$\leq 2^{20}$	✓		
Generic NK DFA	1728	2^0	✓	✓	✓
Enc-Dec IC-NK-DFA	288	2^{32}	✓*	✓	
Multi-round IC-NK-DFA	84 ± 15	2^0	✓*	✓	

Summary of Attacks on DEFAULT-LAYER

Approach	Faults	Time	Key Schedule		
			simple	strong	ideal
DFA [BBB+21b]	64	2^{64}			
Enc-Dec IC-DFA	16	$\leq 2^{39}$	✓		
Multi-round IC-DFA	16	$\leq 2^{20}$	✓		
Generic NK DFA	1728	2^0	✓	✓	✓
Enc-Dec IC-NK-DFA	288	2^{32}	✓*	✓	
Multi-round IC-NK-DFA	84 ± 15	2^0	✓*	✓	

Conclusion

- DEFAULT is an interesting concept to provide cipher-level fault resistance
- Cipher-level protection against DFA remains an open problem
- Substantial ideas beyond linear structures seem to be necessary

- <https://eprint.iacr.org/2021/1374>
- https://extgit.iaik.tugraz.at/castle/tool/dfa_on_default

Information-Combining Differential Fault Attacks on DEFAULT

Marcel Nageler Christoph Dobraunig Maria Eichlseder

FSE Workshop 2022

Table: Information learned when injecting a fault $\Delta_{\text{in}} / \Delta_{\text{out}}$.

Δ_{in}	Learned expression (Enc)	Δ_{out}	Learned expression (Dec)
0	1	0	1
1	$k_0 \oplus k_1 \oplus k_2 \oplus k_3$	1	$k_0 \oplus k_1 \oplus k_2 \oplus k_3$
2	$k_0 \oplus k_2$	2	$k_0 \oplus k_3$
3	$k_1 \oplus k_3$	3	$k_1 \oplus k_2$
4	$k_0 \oplus k_2$	4	$k_0 \oplus k_1 \oplus k_2 \oplus k_3$
5	$k_1 \oplus k_3$	5	1
6	1	6	$k_1 \oplus k_2$
7	$k_0 \oplus k_1 \oplus k_2 \oplus k_3$	7	$k_0 \oplus k_3$
8	$k_0 \oplus k_1 \oplus k_2 \oplus k_3$	8	$k_0 \oplus k_3$
9	1	9	$k_1 \oplus k_2$
a	$k_1 \oplus k_3$	a	1
b	$k_0 \oplus k_2$	b	$k_0 \oplus k_1 \oplus k_2 \oplus k_3$
c	$k_1 \oplus k_3$	c	$k_1 \oplus k_2$
d	$k_0 \oplus k_2$	d	$k_0 \oplus k_3$
e	$k_0 \oplus k_1 \oplus k_2 \oplus k_3$	e	$k_0 \oplus k_1 \oplus k_2 \oplus k_3$
f	1	f	1

Optimized Attack on Strong Key Schedule

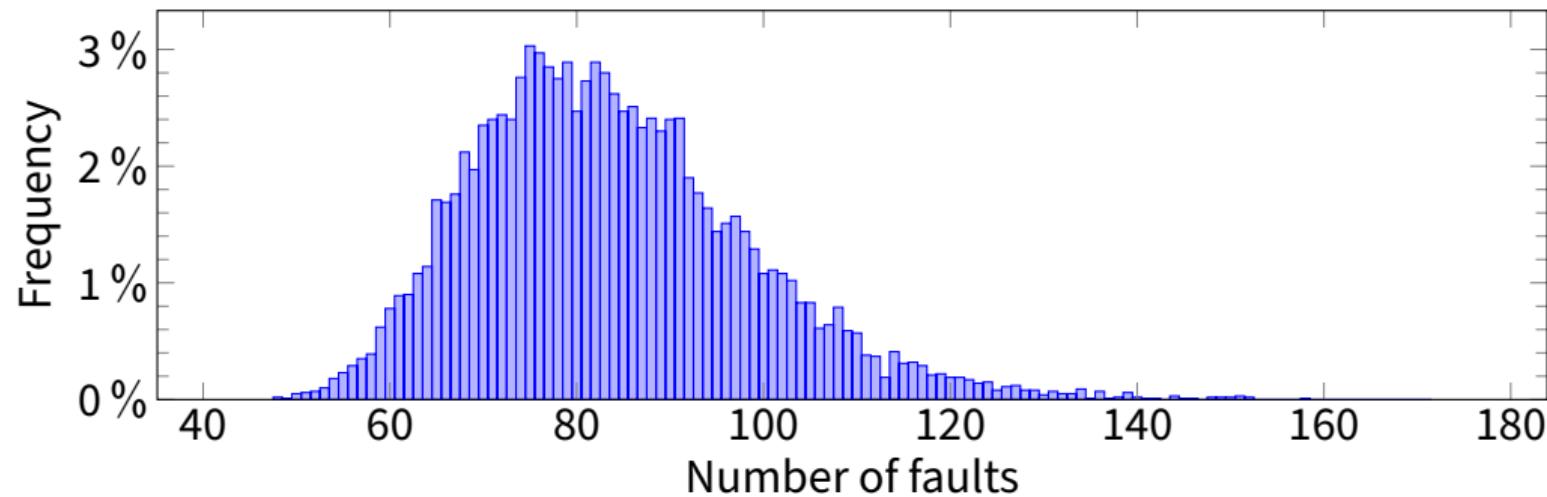


Figure: Number of faults needed to recover the key ($n = 10\,000$).

Bibliography I

- [BBB+21a] Anubhab Baksi, Shivam Bhasin, Jakub Breier, Mustafa Khairallah, Thomas Peyrin, Sumanta Sarkar, and Siang Meng Sim. **DEFAULT: Cipher Level Resistance Against Differential Fault Attack.** IACR Cryptology ePrint Archive, Report 2021/712. 2021.
URL: <https://eprint.iacr.org/2021/712/20210528:092448>.
- [BBB+21b] Anubhab Baksi, Shivam Bhasin, Jakub Breier, Mustafa Khairallah, Thomas Peyrin, Sumanta Sarkar, and Siang Meng Sim. **DEFAULT: Cipher Level Resistance Against Differential Fault Attack.** ASIACRYPT 2021. Vol. 13091. LNCS. Springer, 2021, pp. 124–156.