Meet-in-the-Filter and Dynamic Counting with Applications to Speck

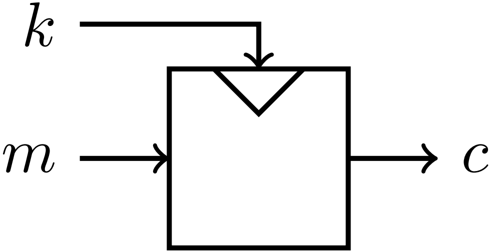
Alex Biryukov[1], Luan Cardoso dos Santos[1], Je Sen Teh[1,2],
Aleksei Udovenko[1], Vesselin Velichkov[3],

[1] SnT, University of Luxembourg
[2] University Sains Malaysia
[3] University of Edinburgh

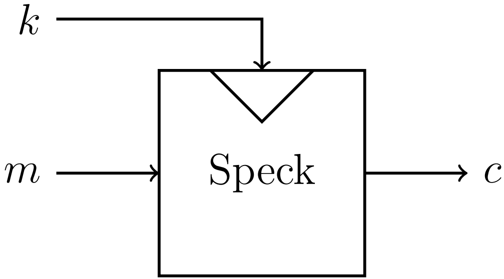
Applied Cryptography and Network Security 2023
19th June 2023

Plan

Symmetric-key Encryption



Symmetric-key Encryption



Differential Cryptanalysis

$$\Delta_{IN} \xrightarrow[\text{differential}]{r \text{ rounds}} \Delta_{OUT}$$

Differential Cryptanalysis

$$\Delta_{IN} \xrightarrow[\text{differential}]{r \text{ rounds}} \Delta_{OUT} \xleftarrow[\text{key recovery}]{k \text{ rounds}} \Delta_C, C_1, C_2$$

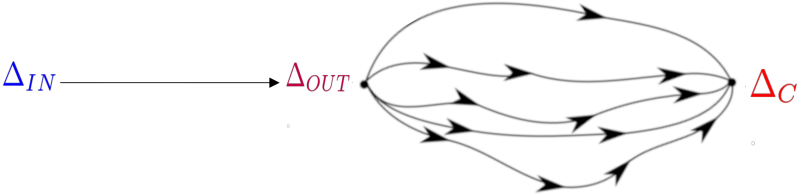
Differential Cryptanalysis

$$\Delta_{IN} \xrightarrow[\text{differential}]{r \text{ rounds}} \Delta_{OUT} \xleftarrow[\text{key recovery}]{k \text{ rounds}} \Delta_C, C_1, C_2$$

How to find key candidates efficiently?

Plan

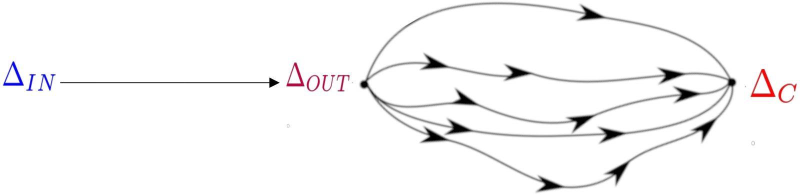
High-level idea



Two-step process:

1. compute most probable **trails** $\Delta_{OUT} \rightarrow \Delta_C$

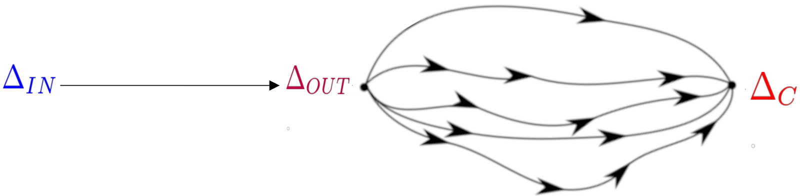
High-level idea



Two-step process:

1. compute most probable **trails** $\Delta_{OUT} \rightarrow \Delta_C$
2. run **trail-assisted** key recovery

High-level idea

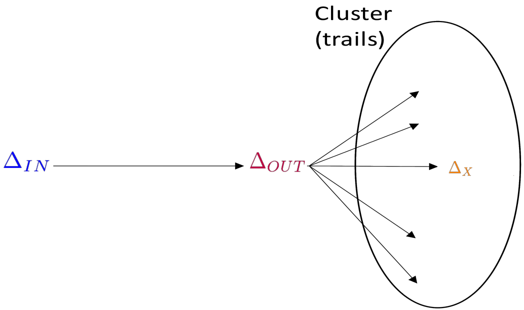


Two-step process:

1. compute most probable **trails** $\Delta_{OUT} \to \Delta_C$
2. run **trail-assisted** key recovery

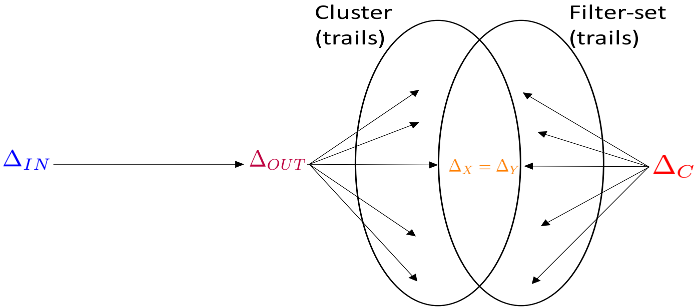
Motivation: alternative to Neural distinguishers (Gohr 2019)

Meet-in-the-Filter



1. precompute the cluster of *trails* $\Delta_{OUT} \rightarrow \Delta_X$

Meet-in-the-Filter



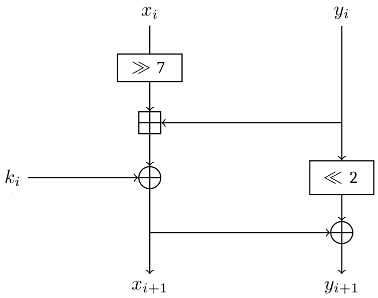
1. precompute the cluster of *trails* $\Delta_{OUT} \rightarrow \Delta_X$
2. (online) for each *observed* Δ_C:
    1. compute the filter-set of *trails* $\Delta_Y \rightarrow \Delta_C$
    2. intersect to get trails $\Delta_{OUT} \rightarrow (\Delta_X = \Delta_Y) \rightarrow \Delta_C$

Previous works

1. Differential meet-in-the-middle, e.g. on LowMC (Rechberger, Soleimany, and Tiessen 2018)
2. Trail-assisted key-recovery, e.g. on Speck (Dinur 2014)

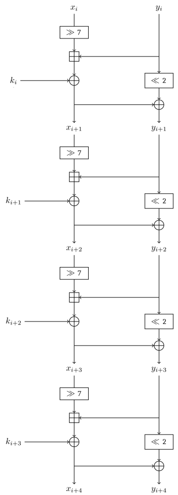
Block-cipher family Speck

- Designed by NSA (2014)
- Simple ARX structure
- Block size: **32**, 48, 64, . . . (2 words)
- Key size: **64**, 72, 96, . . . (2-4 words)
- Speck32:
  $2 \times$ 16-bit words state
  $4 \times$ 16-bit words master key
  22 rounds

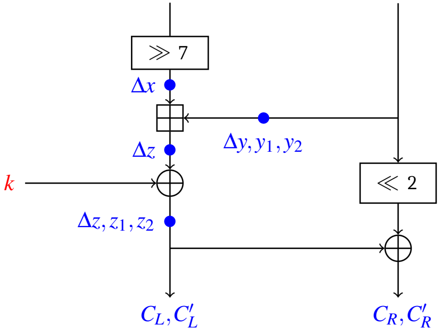


Block-cipher family Speck

- Designed by NSA (2014)
- Simple ARX structure
- Block size: **32**, 48, 64, . . . (2 words)
- Key size: **64**, 72, 96, . . . (2-4 words)
- Speck32:
  $2 \times$ 16-bit words state
  $4 \times$ 16-bit words master key
  22 rounds



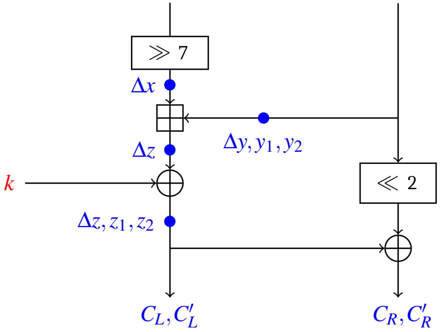
Recursive key recovery (Single-Trail)



Recursive key recovery (Single-Trail)

Procedure:

1. for each ciphertext pair C, C':
2. for each suggested MiF trail τ:
3. recover the last 4 subkeys k recursively bit-by-bit
4. criteria: conformance to the trail τ
5. use key schedule and full trail to test candidates



Recursive key recovery (Single-Trail)

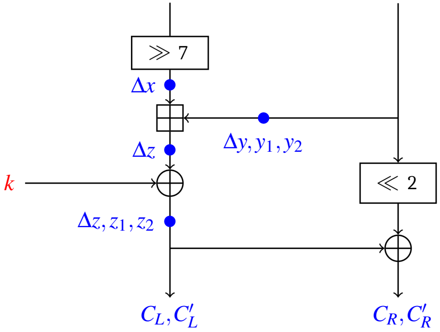
Procedure:

1. for each ciphertext pair C, C':
2. for each suggested MiF trail τ:
3. recover the last 4 subkeys k recursively bit-by-bit
4. criteria: conformance to the trail τ
5. use key schedule and full trail to test candidates

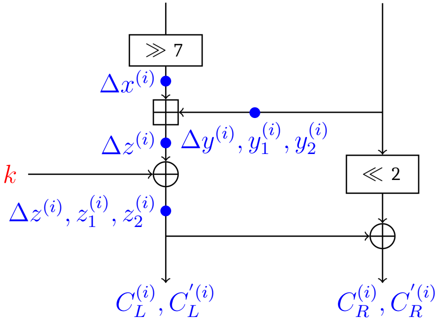
[+] online, memoryless

[+] simple to analyze (Biryukov, Teh, and Udovenko 2023)

[-] limited by the S/N ratio



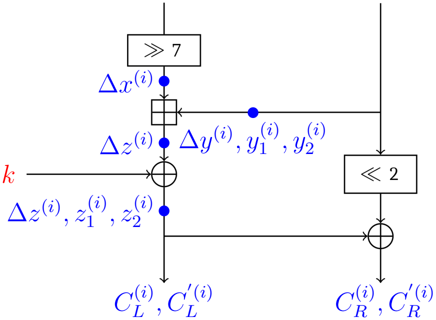
Recursive key recovery (Dynamic Counting)



Recursive key recovery (Dynamic Counting)

Procedure:

1. recursively guess key bits k,
   partially decrypting all available ciphertext pairs
   $C^{(i)}, C'^{(i)}$
2. criteria 1: conformance to available MitF trails
3. criteria 2: $\geq c$ ct pairs alive (e.g. $c = 2, 3, 4, 5$)

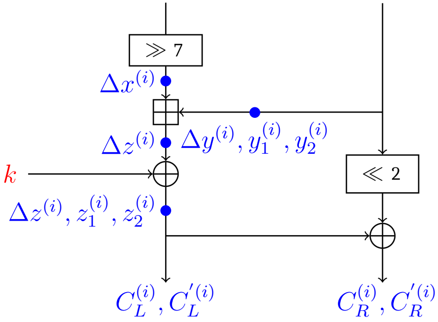


Recursive key recovery (Dynamic Counting)

Procedure:

1. recursively guess key bits k,
   partially decrypting all available ciphertext pairs
   $C^{(i)}, C'^{(i)}$
2. criteria 1: conformance to available MitF trails
3. criteria 2: $\geq c$ ct pairs alive (e.g. $c = 2, 3, 4, 5$)

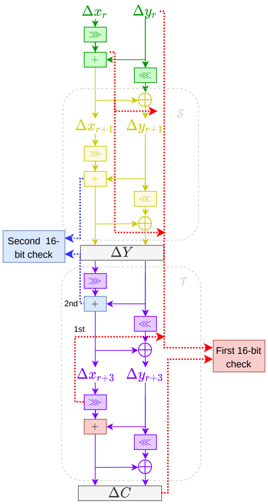
[+] faster attack (stronger filtering)
[-] $\times c$ more data
[-] needs full dataset (memory usage)
[-] harder to analyze



Plan

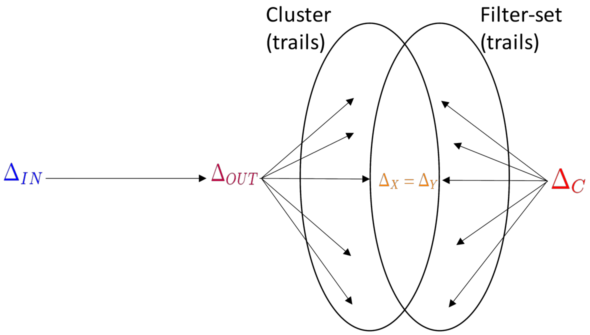
Technical details

- Trail search: our implementation of (Huang and Wang 2019)
- Meet-in-the-middle optimization (fast 1-branch matching)



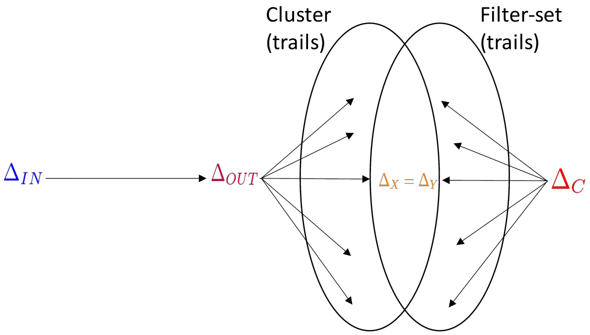
Parameter space

1. # rounds (differential/cluster/filter)



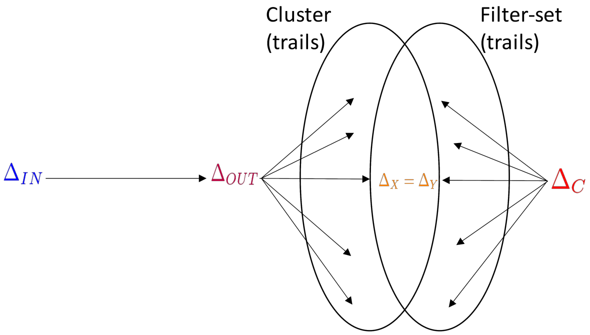
Parameter space

1. \# rounds (differential/cluster/filter)
   - 1+6+2+2=11



Parameter space

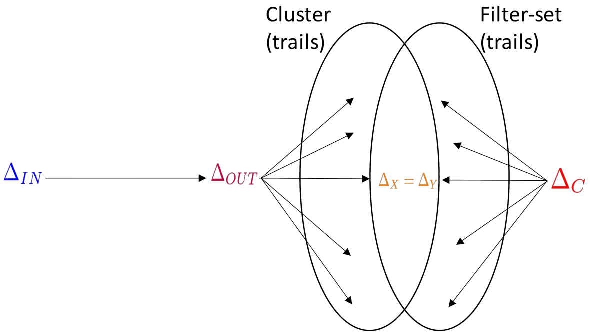
1. # rounds (differential/cluster/filter)
   - 1+6+2+2=11
   - 1+0+8+2=11



Parameter space

1. # rounds (differential/cluster/filter)
   - 1+6+2+2=11
   - 1+0+8+2=11

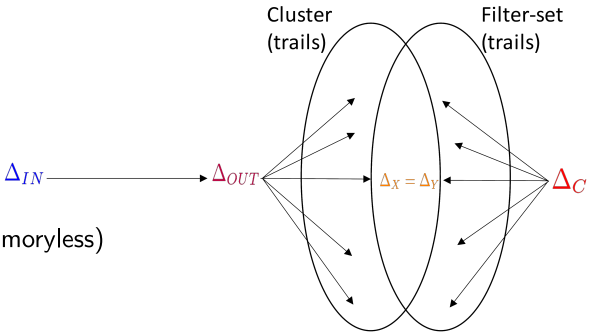
2. counting factor c



Parameter space

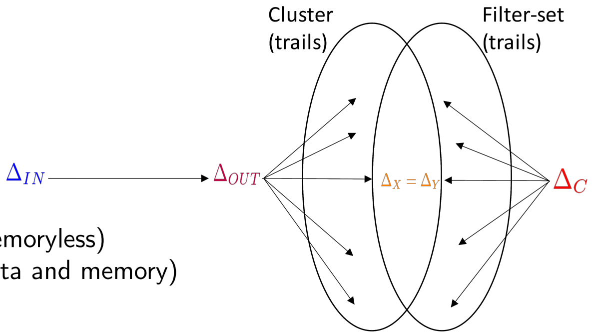
1. # rounds (differential/cluster/filter)
   - 1+6+2+2=11
   - 1+0+8+2=11

2. counting factor c
   - $c = 1$ single-trail analysis (online, memoryless)



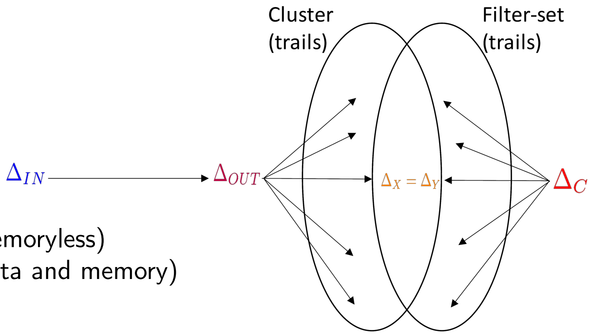
Parameter space

1. # rounds (differential/cluster/filter)
   - 1+6+2+2=11
   - 1+0+8+2=11
2. counting factor c
   - $c = 1$ single-trail analysis (online, memoryless)
   - $c = 2, 3, 4, 5$ better attacks (more data and memory)



Parameter space

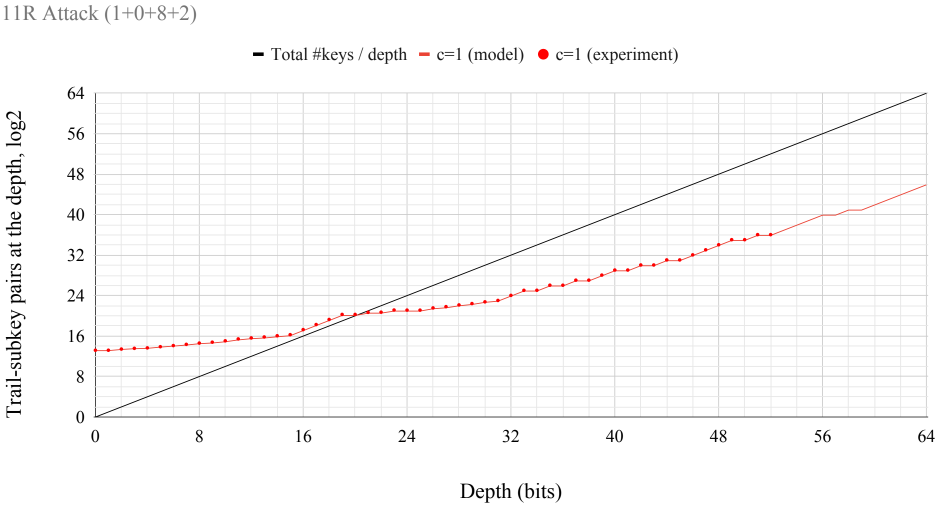
1. # rounds (differential/cluster/filter)
   - 1+6+2+2=11
   - 1+0+8+2=11
2. counting factor c
   - $c = 1$ single-trail analysis (online, memoryless)
   - $c = 2, 3, 4, 5$ better attacks (more data and memory)
3. cluster/filter max weight
   - maximize to avoid signal loss
   - constraint: feasible #trails, low overhead



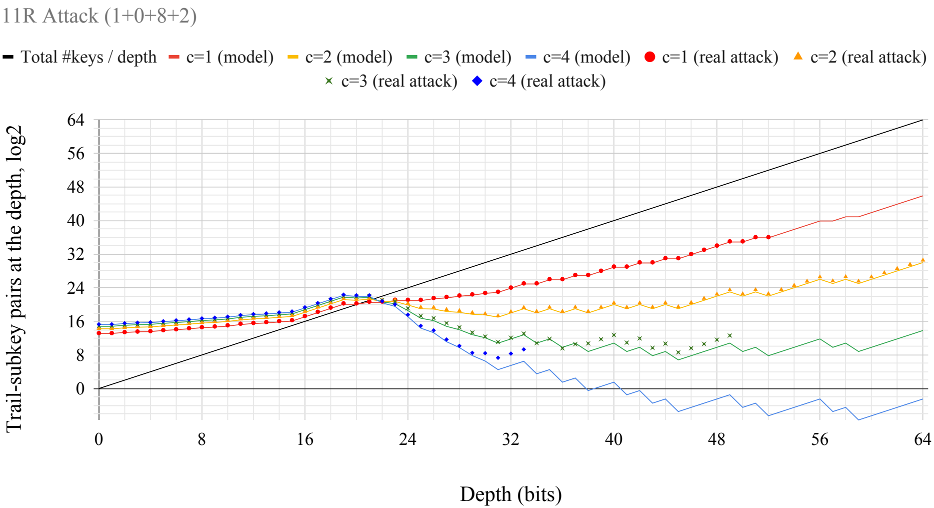
Massive search

1	#roun	split	CW	p.diff	prefix-diff	samp pai	samp tra	log(|S|)	Dq	D|C=1	MiF Time	T_rec	T_keys	T	D|C=2	MiF Time	T_rec	T_keys	T	D|C=3	MiF Time	T_rec	T_keys	T
1776	13	1+8+2+2	43|30	25.00	7c48:b0f8-870a:9720 (2	12.58	19.93	15.01	-27.63	28.63	36.42	55.69	54.27	56.15	29.74	37.52	54.31	49.44	54.35	30.34	38.13	54.20	44.31	54.20
1777	13	1+8+2+2	43|30	25.00	7c48:b0f8-8d0a:9d20 (2	12.51	19.93	15.06	-27.53	28.53	36.36	55.86	54.26	56.27	29.63	37.47	54.50	49.42	54.54	30.24	38.07	54.33	44.28	54.33
1778	13	1+8+2+2	43|30	25.00	7c49:b0f8-850a:9520 (2	11.98	19.93	15.71	-26.74	27.74	36.14	55.72	54.33	56.19	28.84	37.24	54.01	49.56	54.07	29.45	37.85	53.66	44.49	53.67
1779	13	1+8+2+2	43|30	25.00	7c58:b0f8-830a:9320 (2	12.28	19.93	15.29	-27.10	28.10	36.13	55.70	54.28	56.16	29.20	37.23	54.37	49.45	54.42	29.80	37.83	54.13	44.32	54.13
1780	13	1+8+2+2	43|30	25.00	7c58:b0f8-8506:952c (2	12.39	19.93	15.24	-27.31	28.31	36.29	55.82	54.31	56.26	29.41	37.39	54.54	49.51	54.58	30.01	37.99	54.31	44.41	54.31
1781	13	1+8+2+2	43|30	25.00	7c58:b0f8-850e:9524 (2	12.55	19.93	15.17	-27.41	28.41	36.33	55.85	54.28	56.27	29.51	37.43	54.43	49.45	54.48	30.11	38.03	54.22	44.33	54.23
1782	13	1+8+2+2	43|30	25.00	7c58:b0f8-851a:9530 (2	12.74	19.93	14.93	-27.70	28.70	36.42	56.24	54.30	56.58	29.80	37.52	54.95	49.49	54.99	30.40	38.12	54.70	44.38	54.70
1783	13	1+8+2+2	43|30	25.00	7c58:b0f8-870a:9720 (2	12.53	19.93	15.01	-27.63	28.63	36.42	55.70	54.28	56.16	29.74	37.52	54.32	49.46	54.36	30.34	38.13	54.21	44.34	54.21
1784	13	1+8+2+2	43|30	25.00	7c58:b0f8-8d0a:9d20 (2	12.48	19.93	15.06	-27.53	28.53	36.36	55.89	54.29	56.30	29.63	37.47	54.54	49.47	54.58	30.24	38.07	54.37	44.36	54.37
1785	13	1+8+2+2	43|30	25.00	7c59:b0f8-850a:9520 (2	11.95	19.93	15.71	-26.74	27.74	36.14	55.68	54.28	56.14	28.84	37.24	53.95	49.45	54.01	29.45	37.85	53.60	44.33	53.60
1786	13	1+8+2+2	43|30	25.00	7c68:b0f8-850a:9520 (2	11.94	19.93	15.71	-26.74	27.74	36.14	55.68	54.28	56.15	28.84	37.24	53.96	49.46	54.02	29.45	37.85	53.61	44.34	53.62
1787	13	1+8+2+2	43|30	25.00	7c78:b0f8-850a:9520 (2	11.97	19.93	15.71	-26.74	27.74	36.14	55.64	54.25	56.11	28.84	37.24	53.89	49.39	53.96	29.45	37.85	53.55	44.23	53.55
1788	13	1+8+2+2	43|30	25.00	8020:4101-802a:d4a8 (2	12.59	19.93	14.98	-27.46	28.46	36.22	56.21	54.28	56.55	29.56	37.32	55.16	49.46	55.18	30.17	37.93	55.16	44.34	55.16
1789	13	1+8+2+2	43|30	25.00	8021:4101-802a:d4a8 (2	12.62	19.93	14.98	-27.46	28.46	36.22	56.23	54.30	56.57	29.56	37.32	55.19	49.50	55.22	30.17	37.93	55.19	44.40	55.19
1790	13	1+8+2+2	43|30	25.00	8060:4101-802a:d4a8 (2	12.61	19.93	14.98	-27.46	28.46	36.22	56.23	54.30	56.57	29.56	37.32	55.17	49.50	55.20	30.17	37.93	55.17	44.39	55.17
1791	13	1+8+2+2	43|30	25.00	8061:4101-802a:d4a8 (2	12.63	19.93	14.98	-27.46	28.46	36.22	56.25	54.33	56.59	29.56	37.32	55.21	49.54	55.24	30.17	37.93	55.21	44.46	55.22
1792	13	1+8+2+2	43|30	25.00	8148:8100-a850:0952 (2	12.70	19.93	14.94	-27.52	28.52	36.25	55.85	54.25	56.26	29.62	37.35	54.12	49.40	54.17	30.22	37.95	53.68	44.25	53.68
1793	13	1+8+2+2	43|30	25.00	9428:5008-850a:9520 (2	11.93	19.93	15.71	-26.74	27.74	36.14	55.68	54.29	56.14	28.84	37.24	53.94	49.47	54.01	29.45	37.85	53.60	44.35	53.60
1794	13	1+8+2+2	43|30	25.00	9468:5008-850a:9520 (2	11.92	19.93	15.71	-26.74	27.74	36.14	55.67	54.27	56.13	28.84	37.24	53.93	49.43	53.99	29.45	37.85	53.59	44.30	53.59
1795	13	1+8+2+2	43|30	25.00	94c8:1008-850a:9520 (2	11.92	19.93	15.71	-26.74	27.74	36.14	55.69	54.31	56.16	28.84	37.24	53.97	49.51	54.04	29.45	37.85	53.62	44.42	53.63
1796	13	1+8+2+2	43|30	25.00	f448:10f8-850a:9520 (2	11.93	19.93	15.71	-26.74	27.74	36.14	55.66	54.26	56.12	28.84	37.24	53.92	49.41	53.99	29.45	37.85	53.58	44.27	53.59
1797	13	1+8+2+2	43|30	25.00	f449:b1f8-850a:9520 (2	11.92	19.93	15.71	-26.74	27.74	36.14	55.70	54.30	56.17	28.84	37.24	53.99	49.49	54.05	29.45	37.85	53.65	44.39	53.65
1798	13	1+8+2+2	43|30	25.00	f44b:b1f8-850a:9520 (2	11.91	19.93	15.71	-26.74	27.74	36.14	55.69	54.30	56.16	28.84	37.24	53.97	49.49	54.04	29.45	37.85	53.63	44.39	53.63
1799	13	1+8+2+2	43|30	25.00	f459:b1f8-850a:9520 (2	11.96	19.93	15.71	-26.74	27.74	36.14	55.69	54.30	56.16	28.84	37.24	53.97	49.50	54.03	29.45	37.85	53.62	44.40	53.62
1800	13	1+8+2+2	43|30	25.00	f45b:b1f8-850a:9520 (2	11.95	19.93	15.71	-26.74	27.74	36.14	55.69	54.28	56.15	28.84	37.24	53.97	49.45	54.03	29.45	37.85	53.62	44.33	53.62
1801	13	1+8+2+2	43|30	25.00	fc48:10f8-850a:9520 (2	11.91	19.93	15.71	-26.74	27.74	36.14	55.68	54.30	56.15	28.84	37.24	53.95	49.50	54.02	29.45	37.85	53.61	44.40	53.61
1802	13	1+8+2+2	43|30	25.00	fc49:b1f8-850a:9520 (2	11.90	19.93	15.71	-26.74	27.74	36.14	55.70	54.31	56.17	28.84	37.24	53.99	49.51	54.05	29.45	37.85	53.64	44.42	53.64
1803	13	1+8+2+2	43|30	25.00	fc4b:b1f8-850a:9520 (2	11.97	19.93	15.71	-26.74	27.74	36.14	55.68	54.28	56.14	28.84	37.24	53.95	49.46	54.01	29.45	37.85	53.60	44.34	53.60
1804	13	1+8+2+2	43|30	25.00	fc59:b1f8-850a:9520 (2	11.95	19.93	15.71	-26.74	27.74	36.14	55.67	54.28	56.14	28.84	37.24	53.94	49.45	54.00	29.45	37.85	53.60	44.32	53.60
1805	13	1+8+2+2	43|30	25.00	fc5b:b1f8-850a:9520 (2	11.95	19.93	15.71	-26.74	27.74	36.14	55.69	54.29	56.16	28.84	37.24	53.98	49.48	54.04	29.45	37.85	53.63	44.37	53.64

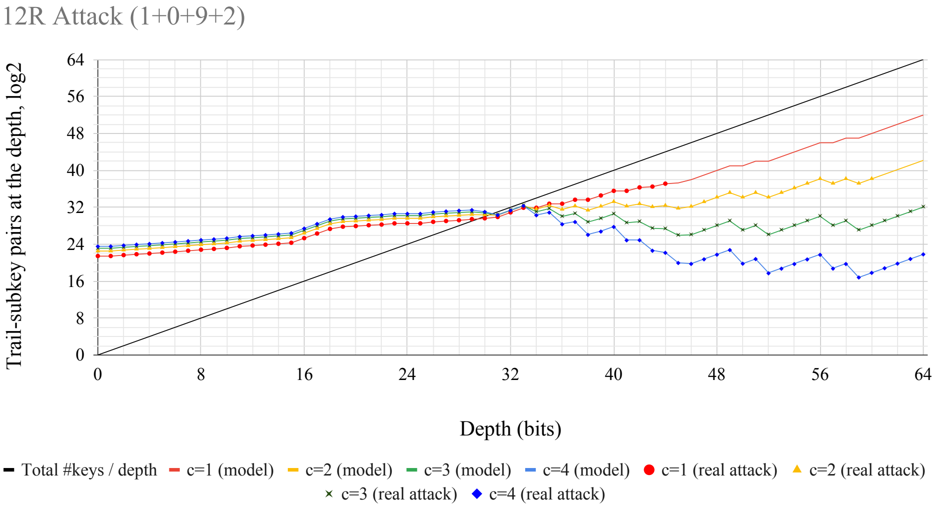
Example analysis



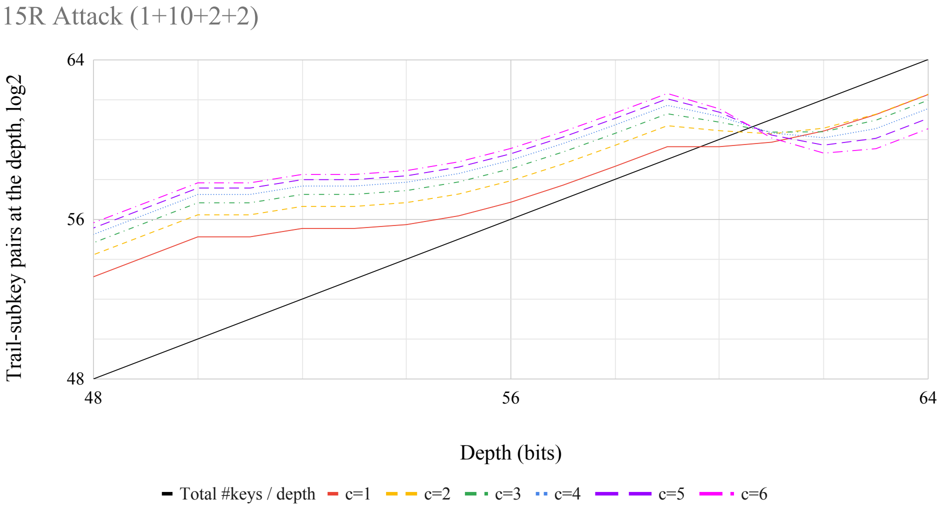
Example analysis



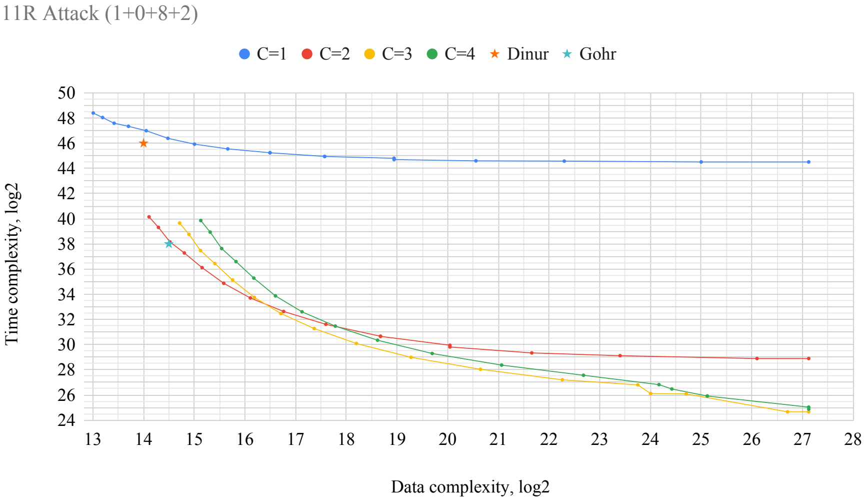
Example analysis



Example analysis



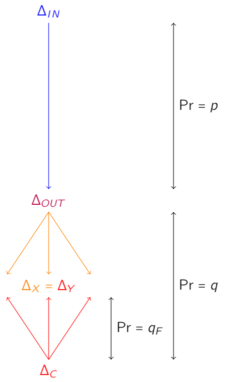
Some results



Plan

Conclusions

- Meet-in-the-Filter is a very versatile framework for differential key recovery
- See ia.cr/2022/673 (ACNS 2023) for:
  1. theoretical framework
  2. analysis techniques
  3. attacks on Speck64/128
- See ia.cr/2023/851 (SAC 2022) for:
  1. simpler theory for $c = 1$
  2. plaintext structures + key bridging
  3. attacks on CHAM and KATAN



References I

Biham, Eli and Adi Shamir (1993). *Differential Cryptanalysis of the Data Encryption Standard*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387979301.

Biryukov, Alex, Je Sen Teh, and Aleksei Udovenko (2023). "Advancing the Meet-in-the-Filter Technique: Applications to CHAM and KATAN". In: *Selected Areas in Cryptography 2022*. Lecture Notes in Computer Science. To appear. Springer.

Dinur, Itai (Aug. 2014). "Improved Differential Cryptanalysis of Round-Reduced Speck". In: *SAC 2014*. Ed. by Antoine Joux and Amr M. Youssef. Vol. 8781. LNCS. Springer, Heidelberg, pp. 147–164. DOI: 10.1007/978-3-319-13051-4_9.

Gohr, Aron (2019). "Improving Attacks on Round-Reduced Speck32/64 Using Deep Learning". In: *CRYPTO 2019*. Vol. 11693. LNCS. Springer, pp. 150–179.

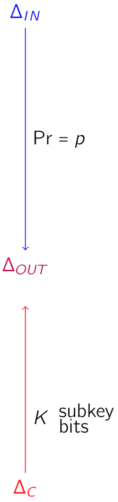
References II

Huang, Mingjiang and Liming Wang (2019). "Automatic Tool for Searching for Differential Characteristics in ARX Ciphers and Applications". In: *INDOCRYPT 2019*. Vol. 11898. LNCS. Springer, pp. 115–138.

Rechberger, Christian, Hadi Soleimany, and Tyge Tiessen (2018). "Cryptanalysis of Low-Data Instances of Full LowMCv2". In: *IACR Trans. Symm. Cryptol.* 2018.3, pp. 163–181. ISSN: 2519-173X. DOI: 10.13154/tosc.v2018.i3.163-181.

Signal/Noise Ratio (Biham and Shamir 1993)

- When is the differential attack meaningful?

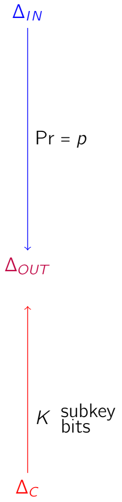


Signal/Noise Ratio (Biham and Shamir 1993)

- When is the differential attack meaningful?
- Signal/Noise ratio:

$$S/N = \frac{2^K p}{w}, \qquad \begin{aligned} p &= \Pr[\Delta_{IN} \to \Delta_{OUT}] \text{ (main differential)} \\ K &= \text{guessed subkeys size} \\ w &= \text{avg \# subkey candidates / pair} \end{aligned}$$

- Faster than K-bit exhaustive search by a factor (S/N)



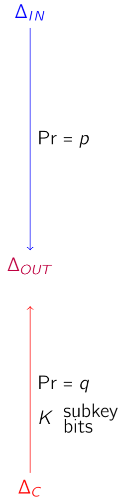
Signal/Noise Ratio (Biham and Shamir 1993)

- When is the differential attack meaningful?
- Signal/Noise ratio:

$$S/N = \frac{2^K p}{w}, \qquad \begin{aligned} p &= \Pr[\Delta_{IN} \to \Delta_{OUT}] \text{ (main differential)} \\ K &= \text{guessed subkeys size} \\ w &= \text{avg \# subkey candidates / pair} \end{aligned}$$

- Faster than K-bit exhaustive search by a factor (S/N)
- Consider observed difference Δ_C:

$$w = 2^K q, \text{ where } q = \Pr[\Delta_{OUT} \to \Delta_C] \quad \text{(MiF trail)}$$



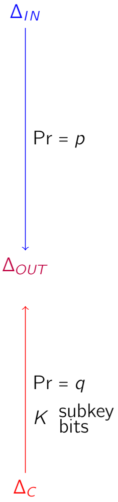
- When is the differential attack meaningful?
- Signal/Noise ratio:

$$S/N = \frac{2^K p}{w}, \qquad \begin{aligned} p &= \Pr[\Delta_{IN} \to \Delta_{OUT}] \text{ (main differential)} \\ K &= \text{guessed subkeys size} \\ w &= \text{avg \# subkey candidates / pair} \end{aligned}$$

- Faster than K-bit exhaustive search by a factor (S/N)
- Consider observed difference Δ_C:

$$w = 2^K q, \text{ where } q = \Pr[\Delta_{OUT} \to \Delta_C] \quad \text{(MiF trail)}$$

- Conclude $S/N = \frac{p}{q}$



Signal/Noise Ratio (Biham and Shamir 1993)

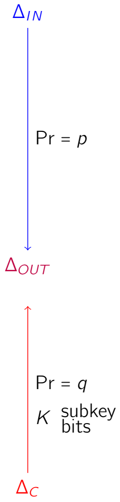
- When is the differential attack meaningful?
- Signal/Noise ratio:

$$S/N = \frac{2^K p}{w}, \qquad \begin{aligned} p &= \Pr[\Delta_{IN} \to \Delta_{OUT}] \text{ (main differential)} \\ K &= \text{guessed subkeys size} \\ w &= \text{avg \# subkey candidates / pair} \end{aligned}$$

- Faster than K-bit exhaustive search by a factor (S/N)
- Consider observed difference Δ_C:

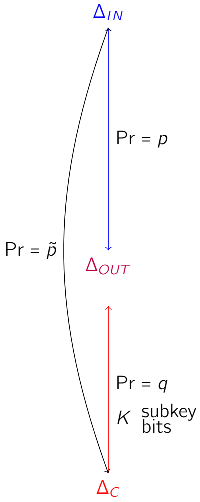
$$w = 2^K q, \text{ where } q = \Pr[\Delta_{OUT} \to \Delta_C] \quad \text{(MiF trail)}$$

- ~~Conclude~~ $\cancel{S/N = \frac{p}{q}}$ INCORRECT



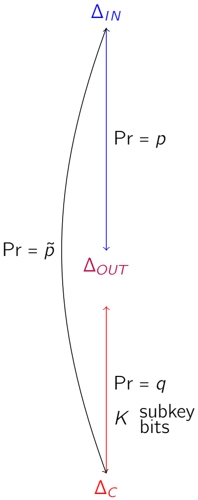
Gain

- define **gain** $g = \dfrac{\Pr[\text{a suggested key is the right one}]}{\Pr[\text{a random key is the right one}]}$



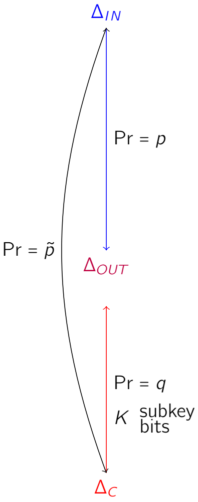
Gain

- define **gain** $g = \dfrac{\Pr[\text{a suggested key is the right one}]}{\Pr[\text{a random key is the right one}]}$
- we show that $g = \dfrac{p}{\tilde{p}} = \dfrac{\Pr[\Delta_{IN} \to \Delta_{OUT}]}{\Pr[\Delta_{IN} \to \Delta_C]} = S/N \cdot \dfrac{q}{\tilde{p}}$



Gain

- define **gain** $g = \dfrac{\Pr[\text{a suggested key is the right one}]}{\Pr[\text{a random key is the right one}]}$
- we show that $g = \dfrac{p}{\tilde{p}} = \dfrac{\Pr[\Delta_{IN} \to \Delta_{OUT}]}{\Pr[\Delta_{IN} \to \Delta_C]} = S/N \cdot \dfrac{q}{\tilde{p}}$



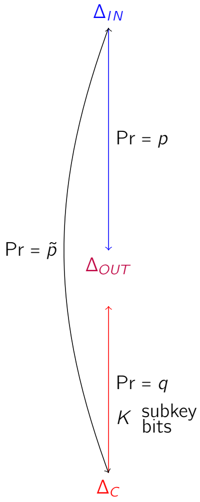
Gain

- define **gain** $g = \dfrac{\text{Pr[a suggested key is the right one]}}{\text{Pr[a random key is the right one]}}$

- we show that $g = \dfrac{p}{\tilde{p}} = \dfrac{\Pr[\Delta_{IN} \to \Delta_{OUT}]}{\Pr[\Delta_{IN} \to \Delta_{C}]} = S/N \cdot \dfrac{q}{\tilde{p}}$

- ciphertext-randomization hypothesis:

$$\tilde{p} = 2^{-|C|} \quad \Rightarrow \quad g = 2^{|C|} p$$



Gain

- define **gain** $g = \dfrac{\text{Pr[a suggested key is the right one]}}{\text{Pr[a random key is the right one]}}$
- we show that $g = \dfrac{p}{\tilde{p}} = \dfrac{\Pr[\Delta_{IN} \to \Delta_{OUT}]}{\Pr[\Delta_{IN} \to \Delta_{C}]} = S/N \cdot \dfrac{q}{\tilde{p}}$
- ciphertext-randomization hypothesis:

$$\tilde{p} = 2^{-|C|} \quad \Rightarrow \quad g = 2^{|C|} p$$

- (general limit of differential key recovery)