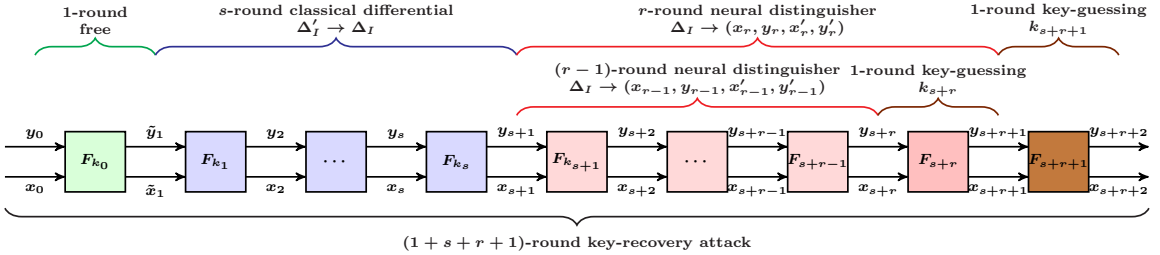
Enhancing Differential-Neural Cryptanalysis

Zhenzhen Bao Jian Guo Meicheng Liu Li Ma Yi Tu

5–9 December 2022, Asiacrypt

Outline

Differential-Neural Cryptanalysis [C:Gohr19]



Components of the key-recovery attacks

- 1-round free on the top
- s-round classical differential $\mathcal{CD}$ $\Delta_{I'} \to \Delta_I$
- r-round and $r-1$-round neural distinguisher $\mathcal{ND}$ trained with difference Δ_I
- 1-round key-guessing for the last and 1-round key-guessing for the second last subkey

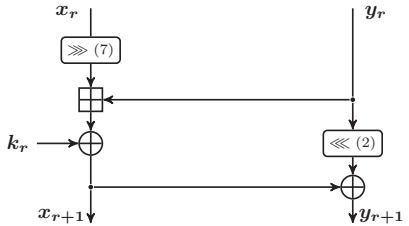
Differential-based Neural Distinguishers [C:Gohr19]

Task: distinguishing two types of ciphertext pairs

$$\text{Positive } (C, C'), \quad Y = 1, \quad \text{where } (C, C') \xleftarrow{\texttt{Enc}} \left((P, P') \mid P \leftarrow_{\$}, P' = P \oplus \Delta_I\right)$$

$$\text{Negative } (C, C'), \quad Y = 0, \quad \text{where } (C, C') \xleftarrow{\texttt{Enc}} \left((P, P') \mid P \leftarrow_{\$}, P' \leftarrow_{\$}\right)$$

No.	w	Train X	Y
0	x	1 0 1 1 0 1 0 0 1 1 1 0 0 1 0 1	1
	y	0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1	
	x'	0 0 1 1 0 0 1 1 0 0 0 1 1 1 0 1	
	y'	0 1 0 1 1 0 0 0 1 1 0 1 0 1 0 0	
1	x	0 1 0 0 1 1 1 0 0 0 0 0 1 0 0 0	0
	y	0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0	
	x'	1 0 0 1 1 1 1 0 0 1 1 1 0 1 0 0	
	y'	0 0 0 0 0 1 0 0 1 1 0 0 1 1 1 0	
2	x	1 0 0 0 1 0 0 0 0 1 1 0 1 0 0 0	1
	y	0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0	
	x'	0 0 1 0 1 1 0 1 0 1 0 0 1 0 1 1	
	y'	0 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0	
3	x	1 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1	0
	y	1 1 0 0 1 0 0 1 1 1 1 0 1 1 1 1	
	x'	1 0 1 0 1 1 1 0 0 1 0 1 0 1 0 0	
	y'	1 1 0 1 1 0 0 1 1 0 0 0 1 0 1 0	
4	x	0 0 1 1 1 0 0 0 1 1 0 0 1 0 1 0	1
	y	0 1 1 0 0 0 0 1 1 0 1 1 0 0 0 1	
	x'	1 1 0 1 1 0 1 0 0 0 1 1 0 1 0 1	
	y'	1 1 0 0 1 0 1 1 1 1 0 1 0 1 1 1	
5	x	0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0	1
	y	0 1 1 0 1 0 0 1 0 1 0 1 1 1 1 0	
	x'	1 0 1 1 1 0 0 1 1 0 0 0 0 0 0 1	
	y'	1 1 0 1 1 0 1 0 0 0 0 1 0 0 1 0	
6	x	1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0	0
	y	1 0 1 1 1 1 0 1 1 0 1 1 1 1 0 1	
	x'	0 1 0 0 0 1 0 0 1 1 1 1 0 1 1 1	
	y'	0 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0	
7	x	1 1 1 1 0 1 1 1 1 1 1 0 0 1 1 0	0
	y	0 1 1 0 1 0 1 0 0 1 1 1 1 0 1 0	
	x'	1 1 1 0 1 0 1 1 0 0 1 1 1 1 0 1	
	y'	1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0	



one round of SPECK32/64

Algorithm 1: Encryption of SPECK32/64

Input: $P := (x_0, y_0)$, $\{k_0, \cdots, k_{21}\}$
Output: $C = (x_{22}, y_{22})$
for $r = 0$ *to* 21 **do**
  $x_{r+1} \leftarrow x_r^{\ggg 7} \boxplus y_r \oplus k_r$
  $y_{r+1} \leftarrow y_r^{\lll 2} \oplus x_{r+1}$
end

No. w		Train X	Y
0	x	1 0 1 1 0 1 0 0 1 1 1 0 0 1 0 1	1
	y	0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1	
	x'	0 0 1 1 0 0 1 1 0 0 0 1 1 1 0 1	
	y'	0 1 0 1 1 0 0 0 1 1 0 1 0 1 0 0	
1	x	0 1 0 0 1 1 1 0 0 0 0 0 1 0 0 0	0
	y	0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0	
	x'	1 0 0 1 1 1 1 0 0 1 1 1 0 1 0 0	
	y'	0 0 0 0 0 1 0 0 1 1 0 0 1 1 1 0	
2	x	1 0 0 0 1 0 0 0 0 1 1 0 1 0 0 0	1
	y	0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0	
	x'	0 0 1 0 1 1 0 1 0 1 0 0 1 0 1 1	
	y'	0 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0	
3	x	1 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1	0
	y	1 1 0 0 1 0 0 1 1 1 1 0 1 1 1 1	
	x'	1 0 1 0 1 1 1 0 0 1 0 1 0 1 0 0	
	y'	1 1 0 1 1 0 0 1 1 0 0 0 1 0 1 0	
4	x	0 0 1 1 1 0 0 0 1 1 0 0 1 0 1 0	1
	y	0 1 1 0 0 0 0 1 1 0 1 1 0 0 0 1	
	x'	1 1 0 1 1 0 1 0 0 0 1 1 0 1 0 1	
	y'	1 1 0 0 1 0 1 1 1 1 0 1 0 1 1 1	
5	x	0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0	1
	y	0 1 1 0 1 0 0 1 0 1 0 1 1 1 1 0	
	x'	1 0 1 1 1 0 0 1 1 0 0 0 0 0 0 1	
	y'	1 1 0 1 1 0 1 0 0 0 0 1 0 0 1 0	
6	x	1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0	0
	y	1 0 1 1 1 1 0 1 1 0 1 1 1 1 0 1	
	x'	0 1 0 0 0 1 0 0 1 1 1 0 1 1 1 1	
	y'	0 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0	
7	x	1 1 1 1 0 1 1 1 1 1 1 0 0 1 1 0	0
	y	0 1 1 0 1 0 1 0 0 1 1 1 1 0 1 0	
	x'	1 1 1 0 1 0 1 1 0 0 1 1 1 0 1 0	
	y'	1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0	

Training schemes

- Basic training
- KeyAveraging algorithm-based
- Staged training

No.	w	Verification X	Z	Y	
0	x	0 1 0 0 1 1 1 0 1 1 0 0 0 0 1 1	0.35	0	TN
	y	0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0			
	x'	1 1 0 0 1 0 0 1 0 0 1 1 1 1 0 0			
	y'	0 0 1 0 1 0 1 0 0 0 0 0 1 1 0 0			
1	x	0 0 0 1 1 1 1 1 1 1 0 0 1 1 0 1 0	0.67	0	FP
	y	1 1 1 0 0 0 1 1 0 1 1 1 1 1 1 0			
	x'	1 0 1 1 0 0 1 1 1 1 0 1 1 0 1 1			
	y'	0 1 1 1 1 0 0 0 0 1 1 0 0 0 1 1 1			
2	x	0 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 0	0.74	1	TP
	y	1 1 0 0 1 0 0 0 1 1 0 1 1 0 0 1			
	x'	1 0 1 0 0 1 0 0 0 1 1 1 1 1 1 0			
	y'	0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1			
3	x	0 0 1 0 0 1 1 0 0 0 1 1 1 1 0 1	0.63	0	FP
	y	1 1 0 0 0 1 0 1 0 1 0 0 0 1 1 1			
	x'	0 1 0 0 1 0 0 0 0 0 1 0 0 1 1 1			
	y'	1 1 0 1 0 0 0 1 0 0 0 0 1 0 1 1			
4	x	0 1 1 1 1 1 0 0 0 0 0 1 0 1 0 0	0.46	1	FN
	y	1 1 1 0 1 0 0 0 1 1 1 0 0 0 1 0			
	x'	0 0 0 0 0 1 1 0 0 0 1 1 0 1 1 1			
	y'	1 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0			
5	x	0 1 1 1 0 1 1 1 0 0 1 1 0 1 1 1	0.42	0	TN
	y	0 0 1 1 1 1 1 0 1 0 0 0 1 1 1 0			
	x'	1 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1			
	y'	1 0 0 0 0 1 1 0 0 0 1 0 1 0 0 1			
6	x	0 1 0 0 1 1 1 1 1 0 0 0 1 0 0 0	0.66	1	TP
	y	0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 0			
	x'	1 0 0 0 0 1 1 0 0 1 1 1 1 1 0 1			
	y'	0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 1			
7	x	1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 1	0.77	1	TP
	y	1 1 0 1 1 0 0 1 0 0 1 0 0 0 0 0			
	x'	0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0			
	y'	1 0 0 0 1 1 1 0 0 0 0 1 1 0 0 0			

```
import numpy as np

def evaluate_tiny(net,X,Y):
    Z = net.predict(X,batch_size=10000).flatten();
    Zbin = (Z > 0.5);
    diff = Y - Z; mse = np.mean(diff*diff);
    n = len(Z);
    n0 = np.sum(Y==0); n1 = np.sum(Y==1);
    acc = np.sum(Zbin == Y) / n;
    tpr = np.sum(Zbin[Y==1]) / n1;
    tnr = np.sum(Zbin[Y==0] == 0) / n0;
    return (acc, tpr, tnr, mse)
```

acc: 0.625, tpr: 0.75, tnr: 0.5, mse: 0.20905

Evaluation

Table: Accuracy of Gohr's neural distinguishers on SPECK32/64 [**C:Gohr19**]

#R	Name	Accuracy	True Positive Rate	True Negative Rate
5	$\mathcal{DD}^{\text{SPECK}_{5R}}$	0.911	0.877	0.947
5	$\mathcal{ND}^{\text{SPECK}_{5R}}$	$0.929 \pm 5.13 \times 10^{-4}$	$0.904 \pm 8.33 \times 10^{-4}$	$0.954 \pm 5.91 \times 10^{-4}$
6	$\mathcal{DD}^{\text{SPECK}_{6R}}$	0.758	0.680	0.837
6	$\mathcal{ND}^{\text{SPECK}_{6R}}$	$0.788 \pm 8.17 \times 10^{-4}$	$0.724 \pm 1.26 \times 10^{-3}$	$0.853 \pm 1.00 \times 10^{-3}$
7	$\mathcal{DD}^{\text{SPECK}_{7R}}$	0.591	0.543	0.640
7	$\mathcal{ND}^{\text{SPECK}_{7R}}$	$0.616 \pm 9.70 \times 10^{-4}$	$0.533 \pm 1.41 \times 10^{-3}$	$0.699 \pm 1.30 \times 10^{-3}$
8	$\mathcal{DD}^{\text{SPECK}_{8R}}$	0.512	0.496	0.527
8	$\mathcal{ND}^{\text{SPECK}_{8R}}$	$0.514 \pm 1.00 \times 10^{-3}$	$0.519 \pm 1.41 \times 10^{-3}$	$0.508 \pm 1.42 \times 10^{-3}$

Results [C:Gohr19]

Target	#R	Time (#Enc)	Data (#CP)	Succ. Rate	Weak keys	Configure	Ref.
SPECK32/64	11	2^{46}	2^{14}	-	2^{64}	1+6 +4	[SAC:Dinur14]
		$2^{38^\star}$	$2^{13.6}$	0.52	2^{64}	1+2 +7+1	[C:Gohr19]
	12	2^{51}	2^{19}	-	2^{64}	1+7 +4	[SAC:Dinur14]
		$2^{43.40^\star}$	$2^{22.97}$	0.40	2^{64}	1+2 +8+1	[C:Gohr19]
	13	2^{57}	2^{25}	-	2^{64}	1+8 +4	[SAC:Dinur14]

- Not available.

* Under the assumption that one second equals the time of 2^{28} executions of SPECK32/64 on a CPU.

Updated Results [This]

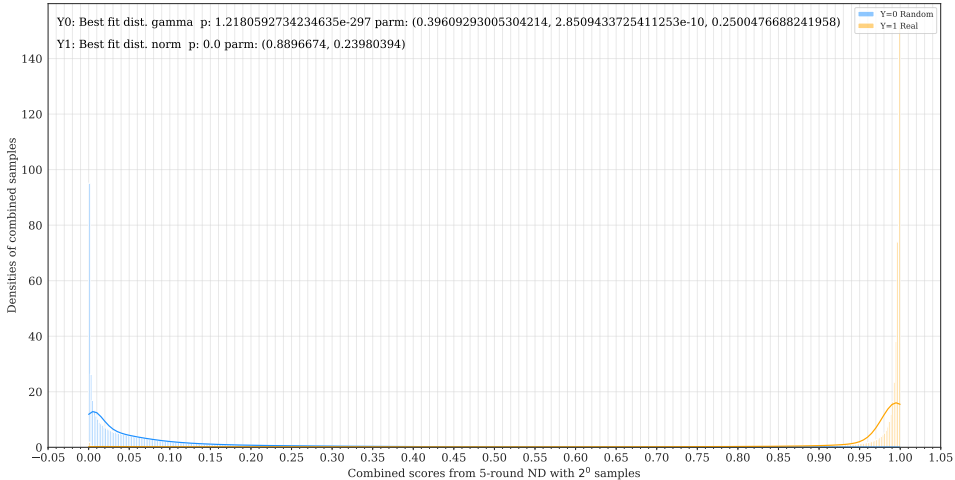
Target	#R	Time (#Enc)	Data (#CP)	Succ. Rate	Weak keys	Configure	Ref.
Speck32/64	11	2^{46}	2^{14}	-	2^{64}	1+6 +4	[SAC:Dinur14]
		$2^{38^\star}$	$2^{13.6}$	0.52	2^{64}	1+2 +7+1	[C:Gohr19]
	12	2^{51}	2^{19}	-	2^{64}	1+7 +4	[SAC:Dinur14]
		$2^{43.40^\star}$	$2^{22.97}$	0.40	2^{64}	1+2 +8+1	[C:Gohr19]
		$2^{44.89^\star}$	$2^{22.00}$	**0.86**	2^{64}	1+2 +8+1	[This]
		$\mathbf{2^{42.97^\star}}$	$\mathbf{2^{18.58}}$	**0.83**	2^{63}	1+3 +7+1	[This]
	13	2^{57}	2^{25}	-	2^{64}	1+8 +4	[SAC:Dinur14]
		$\mathbf{2^{48.67^\star+r}}$	2^{29}	**0.82**	2^{63}	1+3 +8+1	[This]
	14	$2^{62.47}$	$2^{30.47}$	-	2^{64}	1+9 +4	[EPRINT:SonHuaYan16]

- Not available. * Under the assumption that one second equals the time of 2^{28} executions of Speck32/64 or Simon32/64 on a CPU.
$r : \log_2(cpu/gpu)$, where *cpu* and *gpu* are the CPU and GPU time running an attack, respectively. In our computing systems, $r = 2.4$ (The worse case execution time of the core of the 12-round attack on Speck32/64 (without guessing the one key bit of k_0) took 6637 and 1265 seconds on CPU and GPU, respectively).

Outline

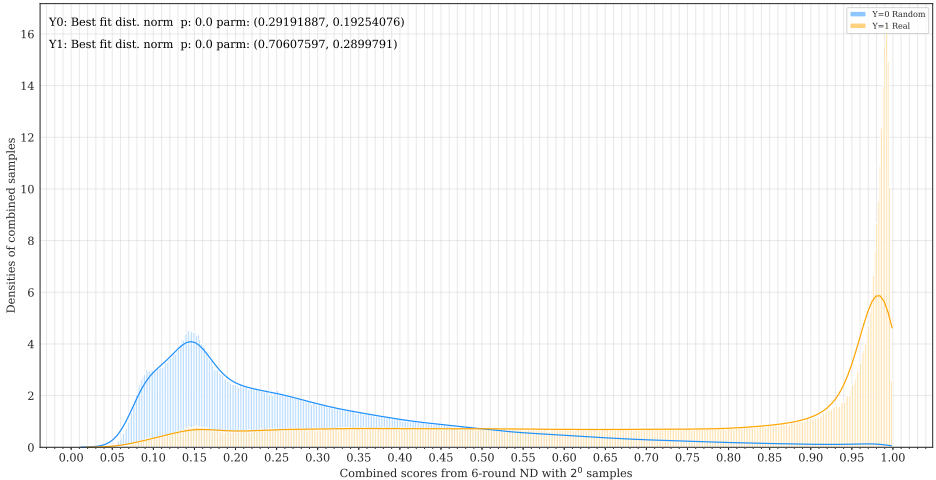
The Motivation

Distribution of responses from $\mathcal{ND}^{\text{SPECK}_{5R}}$



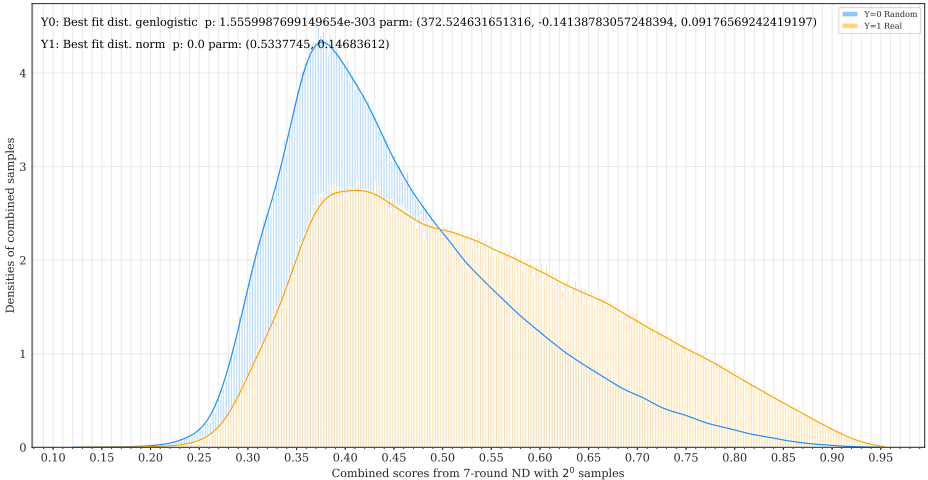
The Motivation

Distribution of responses from $\mathcal{ND}^{\text{SPECK}_{6R}}$



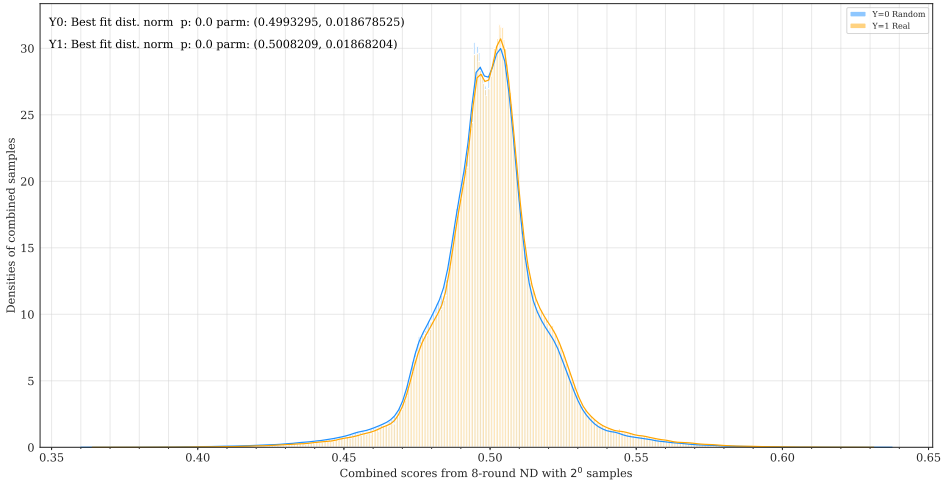
The Motivation

Distribution of responses from $\mathcal{ND}^{\text{SPECK}_{7R}}$



The Motivation

Distribution of responses from $\mathcal{ND}^{\text{SPECK}_{8R}}$



Using Multiple Samples of Same Attribute to Boost Signal [C:Gohr19]

CT	X_0	Z_0	X_1	Z_1	X_2	Z_2	X_3	Z_3	S	Y	
$\mathcal{C}_0$	$C_{0,0}$	0.49	$C_{0,1}$	0.26	$C_{0,2}$	0.16	$C_{0,3}$	0.21	0.26	0	TN
$\mathcal{C}_1$	$C_{1,0}$	0.19	$C_{1,1}$	0.84	$C_{1,2}$	0.13	$C_{1,3}$	0.08	0.26	0	TN
$\mathcal{C}_2$	$C_{2,0}$	0.15	$C_{2,1}$	0.98	$C_{2,2}$	0.97	$C_{2,3}$	0.24	0.75	1	TP
$\mathcal{C}_3$	$C_{3,0}$	0.20	$C_{3,1}$	0.65	$C_{3,2}$	0.98	$C_{3,3}$	0.20	0.61	0	FN
$\mathcal{C}_4$	$C_{4,0}$	0.22	$C_{4,1}$	0.99	$C_{4,2}$	0.16	$C_{4,3}$	0.84	0.68	1	TP
$\mathcal{C}_5$	$C_{5,0}$	0.32	$C_{5,1}$	0.17	$C_{5,2}$	0.14	$C_{5,3}$	0.28	0.22	0	TN
$\mathcal{C}_6$	$C_{6,0}$	0.18	$C_{6,1}$	0.97	$C_{6,2}$	0.99	$C_{6,3}$	0.48	0.83	1	TP
$\mathcal{C}_7$	$C_{7,0}$	0.52	$C_{7,1}$	0.98	$C_{7,2}$	1.00	$C_{7,3}$	0.98	0.97	1	TP

$$S = \frac{1}{1 + e^{-\frac{1}{n_b} \sum_{i=0}^{n_b - 1} \log \frac{Z_i}{1 - Z_i}}}$$

Using Multiple Samples of Same Attribute to Boost Signal [C:Gohr19]

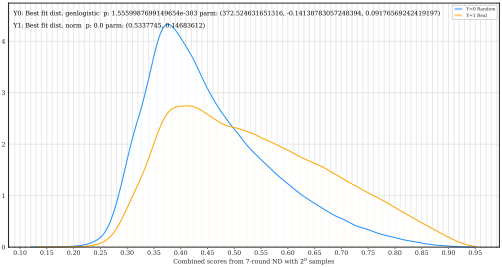
CT	X_0	Z_0	X_1	Z_1	X_2	Z_2	X_3	Z_3	S	Y	
$\mathcal{C}_0$	$C_{0,0}$	0.49	$C_{0,1}$	0.26	$C_{0,2}$	0.16	$C_{0,3}$	0.21	0.26	0	TN
$\mathcal{C}_1$	$C_{1,0}$	0.19	$C_{1,1}$	0.84	$C_{1,2}$	0.13	$C_{1,3}$	0.08	0.26	0	TN
$\mathcal{C}_2$	$C_{2,0}$	0.15	$C_{2,1}$	0.98	$C_{2,2}$	0.97	$C_{2,3}$	0.24	0.75	1	TP
$\mathcal{C}_3$	$C_{3,0}$	0.20	$C_{3,1}$	0.65	$C_{3,2}$	0.98	$C_{3,3}$	0.20	0.61	0	FN
$\mathcal{C}_4$	$C_{4,0}$	0.22	$C_{4,1}$	0.99	$C_{4,2}$	0.16	$C_{4,3}$	0.84	0.68	1	TP
$\mathcal{C}_5$	$C_{5,0}$	0.32	$C_{5,1}$	0.17	$C_{5,2}$	0.14	$C_{5,3}$	0.28	0.22	0	TN
$\mathcal{C}_6$	$C_{6,0}$	0.18	$C_{6,1}$	0.97	$C_{6,2}$	0.99	$C_{6,3}$	0.48	0.83	1	TP
$\mathcal{C}_7$	$C_{7,0}$	0.52	$C_{7,1}$	0.98	$C_{7,2}$	1.00	$C_{7,3}$	0.98	0.97	1	TP

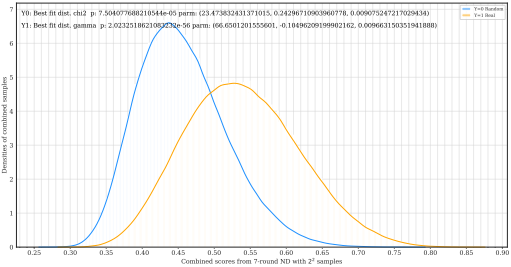
$$S = \frac{1}{1 + e^{-\frac{1}{n_b}\sum_{i=0}^{n_b-1} \log \frac{Z_i}{1-Z_i}}}$$

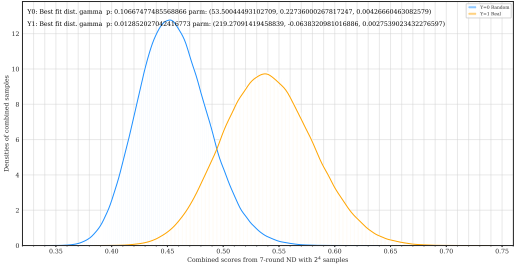
```
import numpy as np

def evaluate_multi(net, n_blocks, n_total, X, Y):
    Z = net.predict(X,batch_size=10000)
    Z = np.log(Z / (1 - Z));
    Z = np.reshape(Z, (n_total, n_blocks))
    Z = np.mean(Z, axis=1);
    Z = 1/(1 + np.exp(-Z))
    Z = Z.flatten();
    Zbin = (Z > 0.5);
    diff = Y - Z; mse = np.mean(diff*diff);
    n = len(Z); n0 = np.sum(Y==0); n1 = np.sum(Y==1);
    acc = np.sum(Zbin == Y) / n;
    tpr = np.sum(Zbin[Y==1]) / n1;
    tnr = np.sum(Zbin[Y==0] == 0) / n0;
    return (acc, tpr, tnr, mse)
```

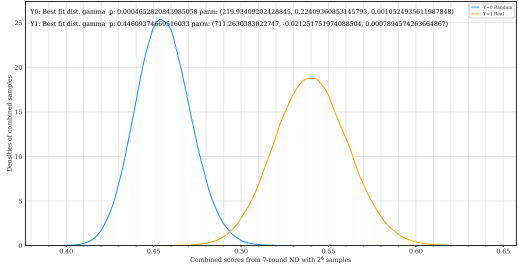
acc: 0.875, tpr: 1.0, tnr: 0.75, mse: 0.0938
↑ ↑ ↑ ↑
acc: 0.625, tpr: 0.75, tnr: 0.5, mse: 0.20905

Effect of the Boosting of Signal

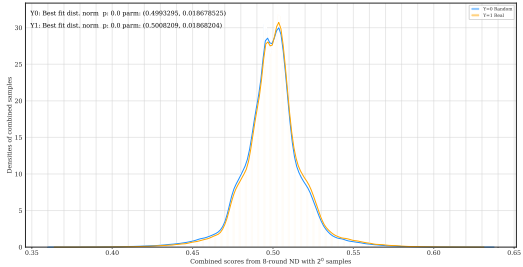


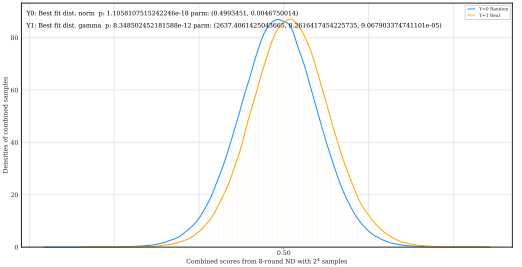


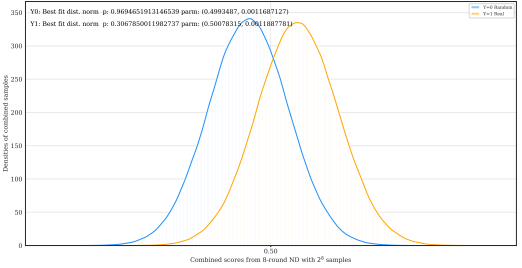


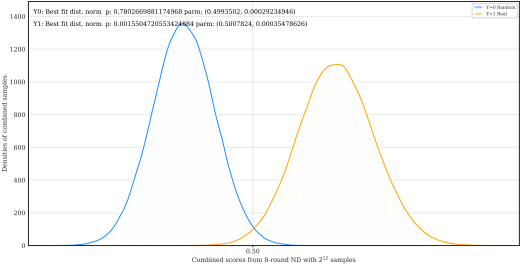


Effect of the Boosting of Signal

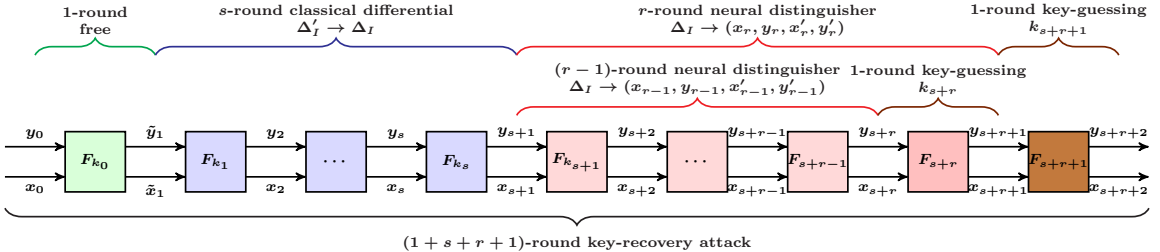








Differential-Neural Cryptanalysis



Components of the key-recovery attacks

- 1-round free on the top
- s-round classical differential $\mathcal{CD}$ $\Delta_{I'} \to \Delta_I$
- r-round and $r-1$-round neural distinguisher $\mathcal{ND}$ trained with difference Δ_I
- 1-round key-guessing for the last and 1-round key-guessing for the second last subkey

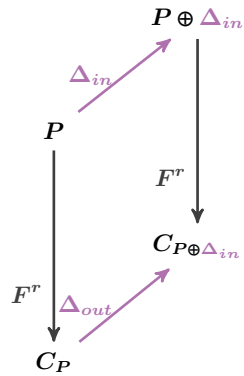
Neutral Bits of $\mathcal{CD}$

Neutral bits, NBs [C:BihChe04]

The i-th bit is a *neutral bit* of the differential $\Delta_{in} \to \Delta_{out}$, if for any conforming pair (P, P'), $(P \oplus e_i, P' \oplus e_i)$ is also a conforming pair, where, $e_0, e_1, \ldots, e_{n-1}$ are the standard basis of $\mathbb{F}_2^n$.

Related concepts

- Message modification [**EC:WanYu05**]
- Tunnels [**EPRINT:Klima06a**]
- Boomerangs [**C:JouPey07**]
- Probabilistic neutral bits [**FSE:AFKMR08**]
- Free bits [**AC:KneMeiNay10**]



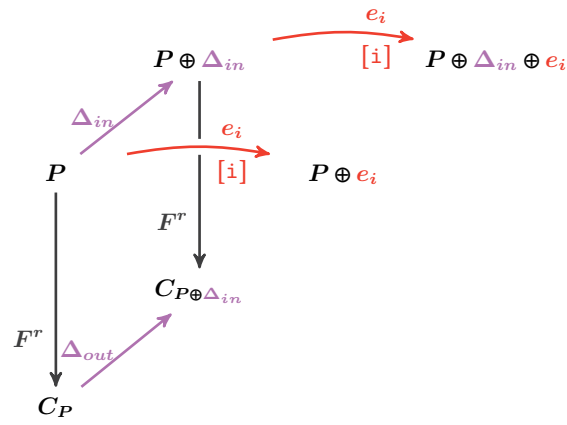
Neutral Bits of $\mathcal{CD}$

Neutral bits, NBs [C:BihChe04]

The i-th bit is a *neutral bit* of the differential $\Delta_{in} \to \Delta_{out}$, if for any conforming pair (P, P'), $(P \oplus e_i, P' \oplus e_i)$ is also a conforming pair, where, $e_0, e_1, \ldots, e_{n-1}$ are the standard basis of $\mathbb{F}_2^n$.

Related concepts

- Message modification [**EC:WanYu05**]
- Tunnels [**EPRINT:Klima06a**]
- Boomerangs [**C:JouPey07**]
- Probabilistic neutral bits [**FSE:AFKMR08**]
- Free bits [**AC:KneMeiNay10**]



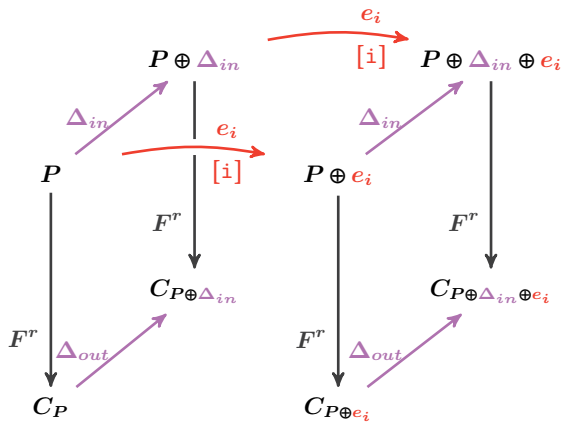
Neutral Bits of $\mathcal{CD}$

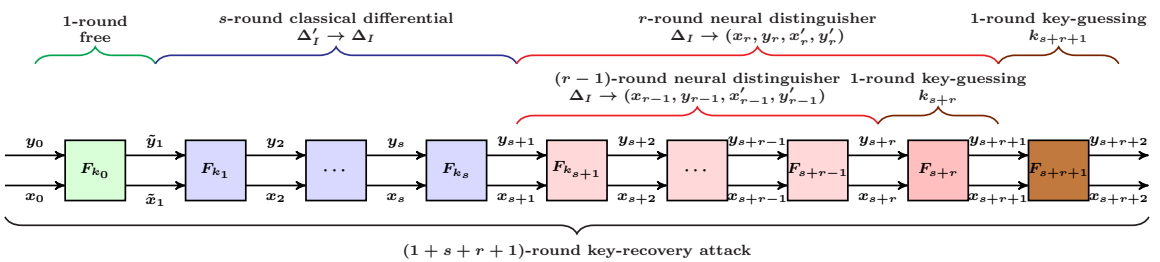
Neutral bits, NBs [C:BihChe04]

The i-th bit is a *neutral bit* of the differential $\Delta_{in} \to \Delta_{out}$, if for any conforming pair (P, P'), $(P \oplus e_i, P' \oplus e_i)$ is also a conforming pair, where, $e_0, e_1, \ldots, e_{n-1}$ are the standard basis of $\mathbb{F}_2^n$.

Related concepts

- Message modification [**EC:WanYu05**]
- Tunnels [**EPRINT:Klima06a**]
- Boomerangs [**C:JouPey07**]
- Probabilistic neutral bits [**FSE:AFKMR08**]
- Free bits [**AC:KneMeiNay10**]

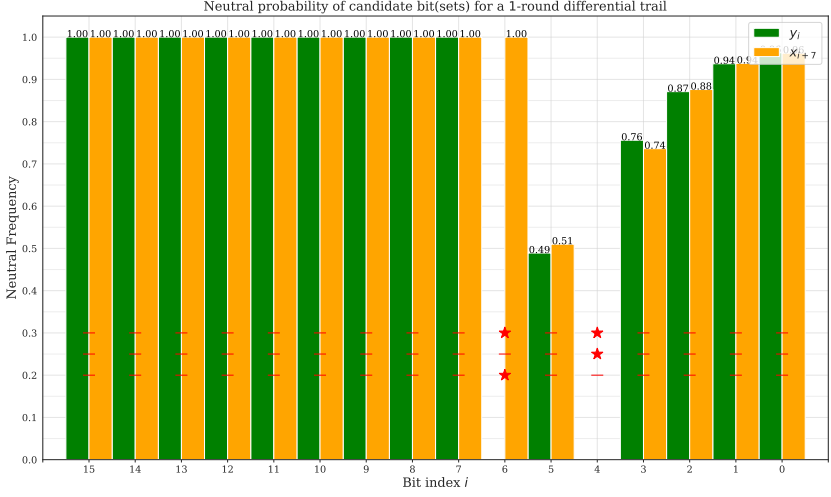




Ciphertext Structures: $\boldsymbol{b}$ neutral bits $\Rightarrow \mathbf{2}^{\boldsymbol{b}}$ ciphertext pairs per structure

$\mathcal{C}_1$	$\{(C_{1,0}, C'_{1,0})$	$(C_{1,1}, C'_{1,1})$	$\cdots$	$(C_{1,2^b-1}, C'_{1,2^b-1})\}$	0
$\mathcal{C}_2$	$\{(C_{2,0}, C'_{2,0})$	$(C_{2,1}, C'_{2,1})$	$\cdots$	$(C_{2,2^b-1}, C'_{2,2^b-1})\}$	0
$\mathcal{C}_3$	$\{(C_{3,0}, C'_{3,0})$	$(C_{3,1}, C'_{3,1})$	$\cdots$	$(C_{3,2^b-1}, C'_{3,2^b-1})\}$	0
$\mathcal{C}_4$	$\{(C_{4,0}, C'_{4,0})$	$(C_{4,1}, C'_{4,1})$	$\cdots$	$(C_{4,2^b-1}, C'_{4,2^b-1})\}$	1
$\mathcal{C}_5$	$\{(C_{5,0}, C'_{5,0})$	$(C_{5,1}, C'_{5,1})$	$\cdots$	$(C_{5,2^b-1}, C'_{5,2^b-1})\}$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathcal{C}_{n_{cts}}$	$\{(C_{n_{cts},0}, C'_{n_{cts},0}(C_{n_{cts},1}, C'_{n_{cts},1})$		$\cdots$	$(C_{n_{cts},2^b-1}, C'_{n_{cts},2^b-1})\}$	0

Neutral Bits of $\mathcal{CD}$ - **NB**s and **PNB**s

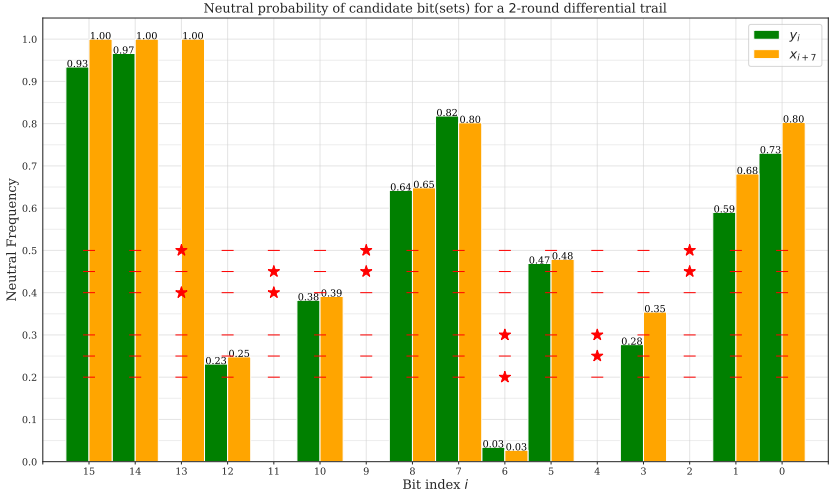


(2800 0010)

↓

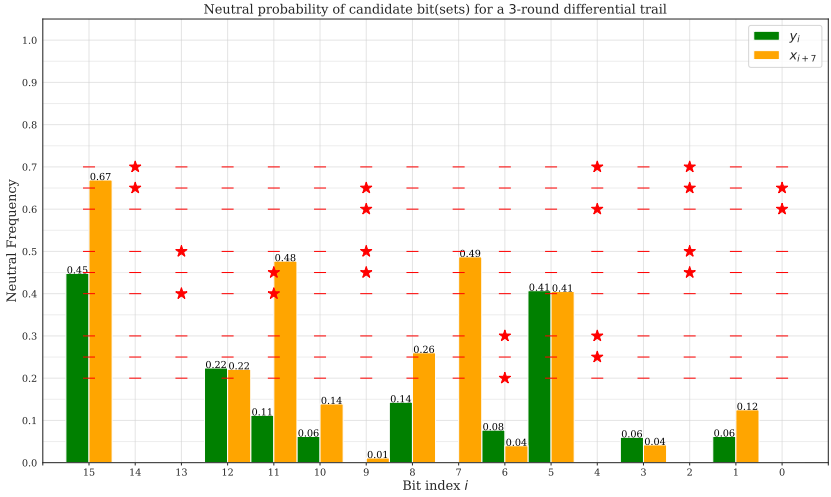
(0040 0000)

Neutral Bits of $\mathcal{CD}$ - **NB**s and **PNB**s



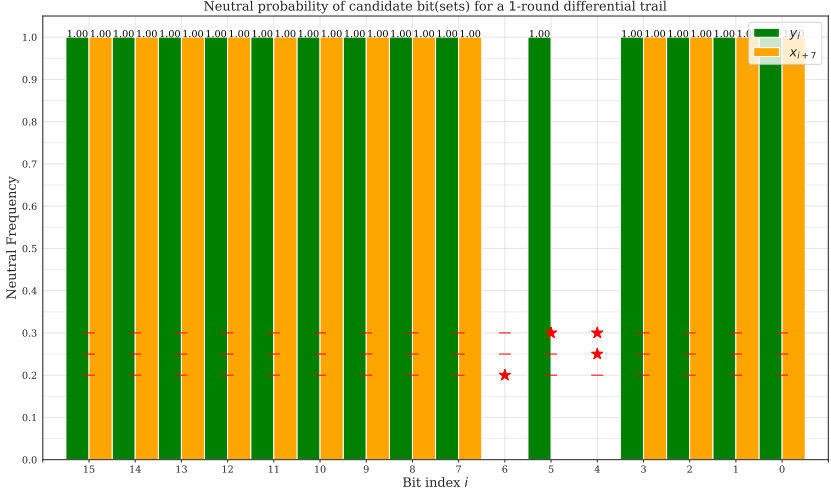
(0211 0A04)
↓
(2800 0010)
↓
(0040 0000)

Neutral Bits of $\mathcal{CD}$ - **NB**s and **PNB**s



(0A20 4205)
↓
(0211 0A04)
↓
(2800 0010)
↓
(0040 0000)

Neutral Bits of $\mathcal{CD}$ - **NB**s and **PNB**s

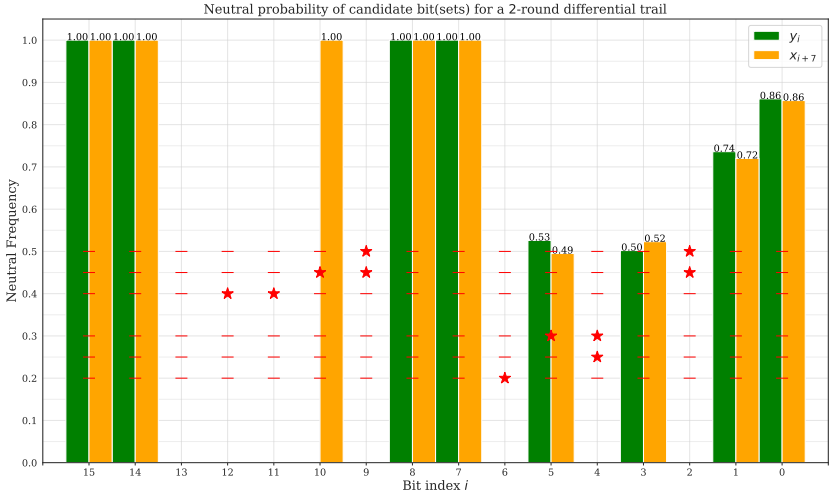


(1800 0010)

↓

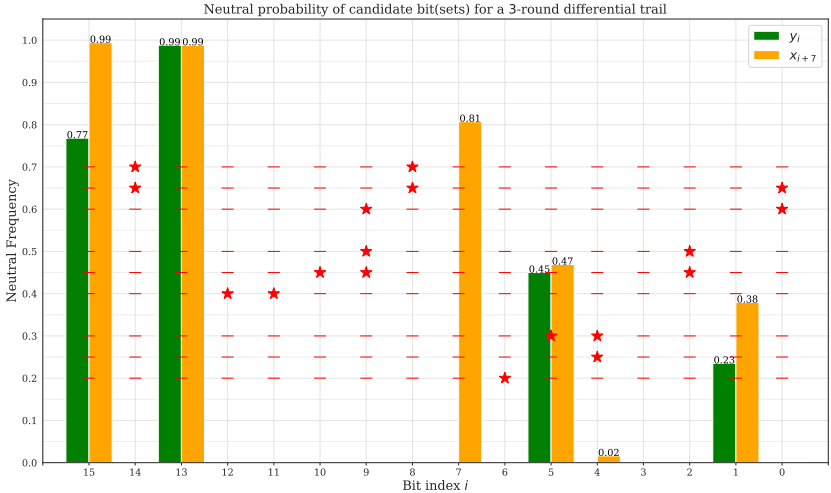
(0040 0000)

Neutral Bits of $\mathcal{CD}$ - **NB**s and **PNB**s



(0201 0604)
↓
(1800 0010)
↓
(0040 0000)

Neutral Bits of $\mathcal{CD}$ - **NB**s and **PNB**s



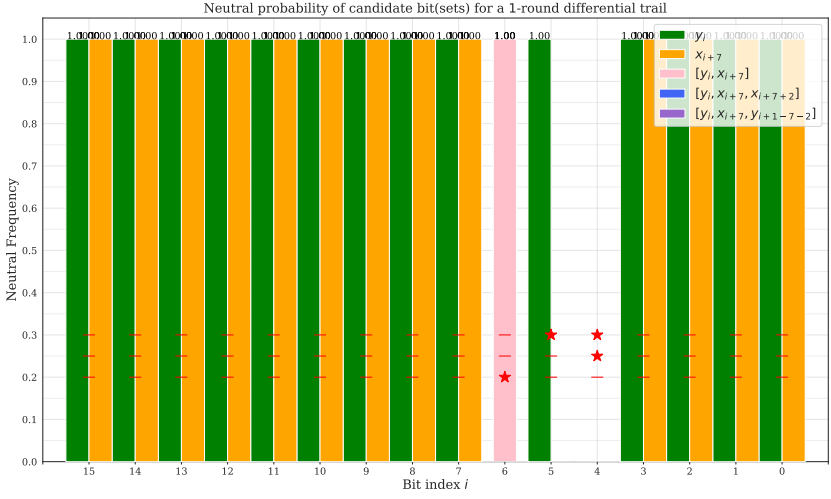
(8020 4101)
↓
(0201 0604)
↓
(1800 0010)
↓
(0040 0000)

Generalized Neutral Bits of $\mathcal{CD}$ - **SNBS**s

Simultaneous-neutral bit-sets, SNBSs [C:BihChe04]

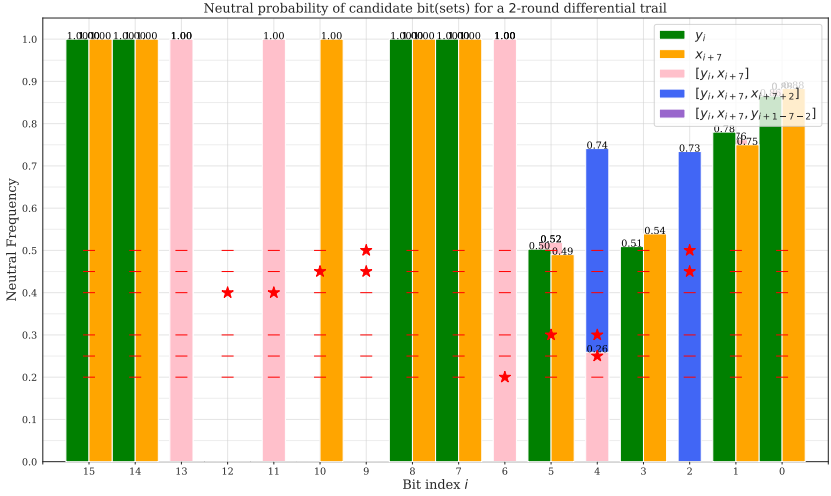
- Let $I_s = \{i_1, i_2, \ldots, i_s\}$ be a set of bit indices. Denote $f_{I_s} = \bigoplus_{i \in I_s} e_i$.
- The bit-set I_s is a simultaneous-neutral bit-set for the differential $\Delta_{in} \to \Delta_{out}$, if for any conforming pair (P, P'), $(P \oplus f_{I_s}, P' \oplus f_{I_s})$ is also a conforming pair, while for any subsets of I_s, the conformability of the resulted pair does not always hold.

Generalized Neutral Bits of $\mathcal{CD}$ - **SNBS**s



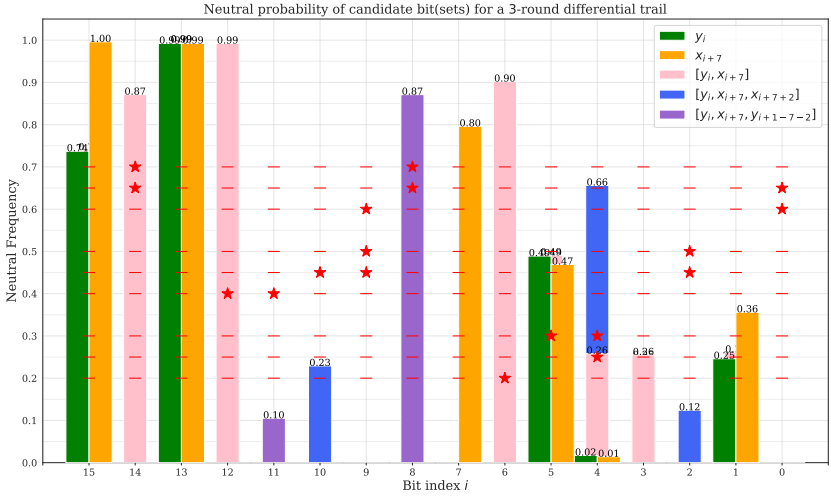
(1800 0010)
↓
(0040 0000)

Generalized Neutral Bits of $\mathcal{CD}$ - **SNBS**s



(0201 0604)
↓
(1800 0010)
↓
(0040 0000)

Generalized Neutral Bits of $\mathcal{CD}$ - **SNBS**s



(8020 4101)
↓
(0201 0604)
↓
(1800 0010)
↓
(0040 0000)

Generalized Neutral Bits of $\mathcal{CD}$ - **CSNBS**s

Conditional (simultaneous-) neutral bit(-set)s, CSNBSs

- Let $I_s = \{i_1, i_2, \ldots, i_s\}$ be a set of bit indices. Denote $f_{I_s} = \bigoplus_{i \in I_s} e_i$.
- Let $\mathcal{C}$ be a set of constraints on the value of an input P, and $\mathcal{P}_\mathcal{C}$ be the set of inputs that fulfill the constraints $\mathcal{C}$.
- The bit-set I_s is a conditional simultaneous-neutral bit-set for the differential $\Delta_{in} \rightarrow \Delta_{out}$, if for any conforming pair (P, P') where $P \in \mathcal{P}_\mathcal{C}$, $(P \oplus f_{I_s}, P' \oplus f_{I_s})$ is also a conforming pair.

Concerned parameters: value of bits in involved variables ($i \in \{0, \cdots, n-1\}$)

- x
- y
- $x^{\ggg 7} \oplus y$

Posterior and prior probability

$$\begin{cases} \Pr([y_5, x_{12}] \text{ is } \mathtt{N}) \approx 0.49 \\ \Pr(y_5 \oplus x_{12} = 1 \mid [y_5, x_{12}] \text{ is } \mathtt{N}) \approx 1.00 \\ \Pr(y_5 \oplus x_{12} = 1) \approx 0.49 \end{cases} \quad \begin{cases} \Pr([y_{15}, x_6, x_8] \text{ is } \mathtt{N}) \approx 0.51 \\ \Pr(y_1 = 0 \mid [y_{15}, x_6, x_8] \text{ is } \mathtt{N}) \approx 1.00 \\ \Pr(y_1 = 0) \approx 0.51 \end{cases} \quad \begin{cases} \Pr([y_4, x_{11}, x_{13}] \text{ is } \mathtt{N}) \approx 0.69 \\ \Pr(y_4 \oplus x_{11} = 1 \mid [y_4, x_{11}, x_{13}] \text{ is } \mathtt{N}) \approx 0.67 \\ \Pr(y_4 \oplus x_{11} = 1) \approx 0.51 \end{cases}$$

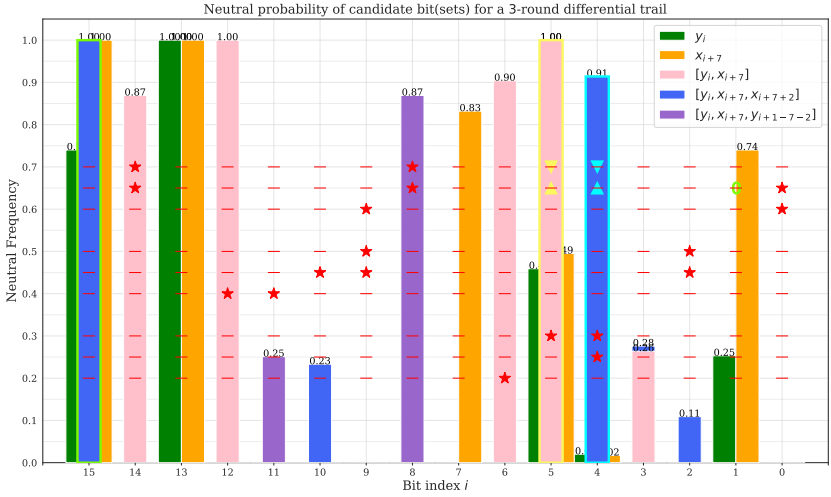
Likelihood

$$\Pr([y_5, x_{12}] \text{ is } \mathtt{N} \mid y_5 \oplus x_{12} = 1) = \frac{\Pr(y_5 \oplus x_{12} = 1 \mid [y_5, x_{12}] \text{ is } \mathtt{N}) \cdot \Pr([y_5, x_{12}] \text{ is } \mathtt{N})}{\Pr(y_5 \oplus x_{12} = 1)} \approx \frac{1.00 \cdot 0.49}{0.49} = 1.00$$

$$\Pr([y_{15}, x_6, x_8] \text{ is } \mathtt{N} \mid y_1 = 0) = \frac{\Pr(y_1 = 0 \mid [y_{15}, x_6, x_8] \text{ is } \mathtt{N}) \cdot \Pr([y_{15}, x_6, x_8] \text{ is } \mathtt{N})}{\Pr(y_1 = 0)} \approx \frac{1.00 \cdot 0.51}{0.51} = 1.00$$

$$\Pr([y_4, x_{11}, x_{13}] \text{ is } \mathtt{N} \mid y_4 \oplus x_{11} = 1) = \frac{\Pr(y_4 \oplus x_{11} = 1 \mid [y_4, x_{11}, x_{13}] \text{ is } \mathtt{N}) \cdot \Pr([y_4, x_{11}, x_{13}] \text{ is } \mathtt{N})}{\Pr(y_4 \oplus x_{11} = 1)} \approx \frac{0.67 \cdot 0.69}{0.51} = 0.91$$

Generalized Neutral Bits of $\mathcal{CD}$ - **CSNBS**s



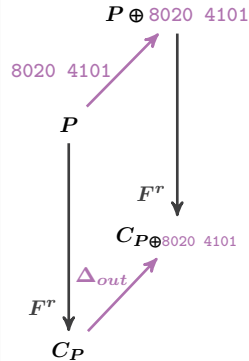
(8020 4101)
↓
(0201 0604)
↓
(1800 0010)
↓
(0040 0000)

Switching Bits for Adjoining Differentials - **SBfAD**s

Switching bits for adjoining differentials, SBfADs

The j-th bit is a *switching bit* of two differentials $\delta_1 = \Delta_{in_1} \rightarrow \Delta_{out}$ and $\delta_2 = \Delta_{in_2} \rightarrow \Delta_{out}$, if for any conforming pair $(P, P \oplus \Delta_{in_1})$ of δ_1, flipping the j-th bit and adjusting the input difference, the resulted pair $(P \oplus e_j, P \oplus e_j \oplus \Delta_{in_2})$ conforms to δ_2 under the same key. We call δ_1 and δ_2 *adjoining differentials*.

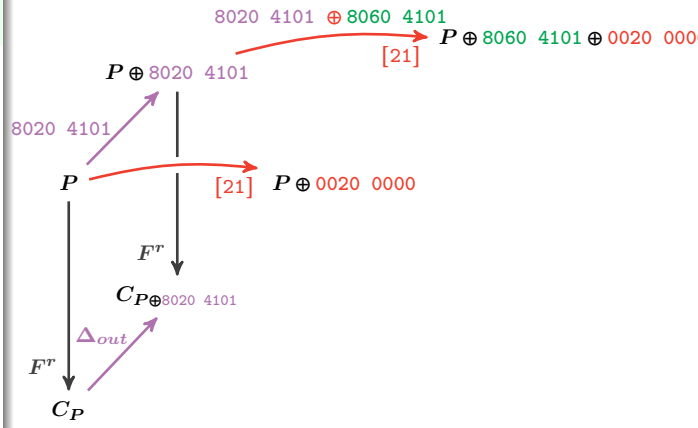
A SBfAD can play the same role as a NB/SNBS for doubling the size of a ciphertext structure.



Switching Bits for Adjoining Differentials - **SBfAD**s

Switching bits for adjoining differentials, SBfADs

The j-th bit is a *switching bit* of two differentials $\delta_1 = \Delta_{in_1} \rightarrow \Delta_{out}$ and $\delta_2 = \Delta_{in_2} \rightarrow \Delta_{out}$, if for any conforming pair $(P, P \oplus \Delta_{in_1})$ of δ_1, flipping the j-th bit and adjusting the input difference, the resulted pair $(P \oplus e_j, P \oplus e_j \oplus \Delta_{in_2})$ conforms to δ_2 under the same key. We call δ_1 and δ_2 *adjoining differentials*.
A SBfAD can play the same role as a NB/SNBS for doubling the size of a ciphertext structure.

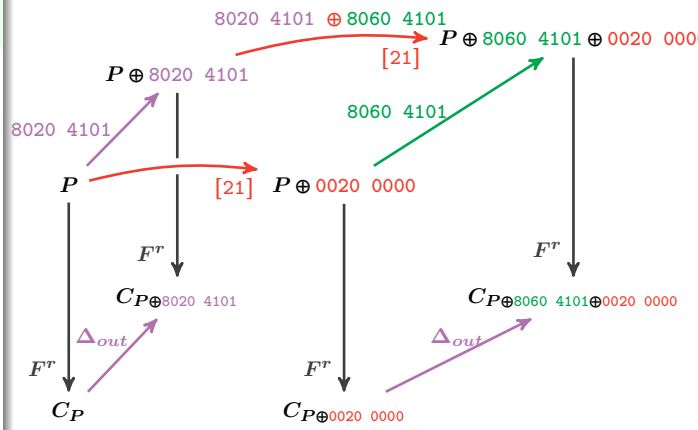


Switching Bits for Adjoining Differentials - **SBfAD**s

Switching bits for adjoining differentials, SBfADs

The j-th bit is a *switching bit* of two differentials $\delta_1 = \Delta_{in_1} \rightarrow \Delta_{out}$ and $\delta_2 = \Delta_{in_2} \rightarrow \Delta_{out}$, if for any conforming pair $(P, P \oplus \Delta_{in_1})$ of δ_1, flipping the j-th bit and adjusting the input difference, the resulted pair $(P \oplus e_j, P \oplus e_j \oplus \Delta_{in_2})$ conforms to δ_2 under the same key. We call δ_1 and δ_2 *adjoining differentials*.

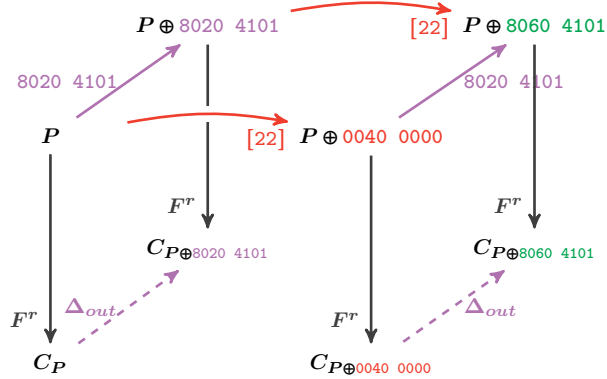
A SBfAD can play the same role as a NB/SNBS for doubling the size of a ciphertext structure.



Paired Differentials Sharing the Same Neutral Bits

Paired differentials

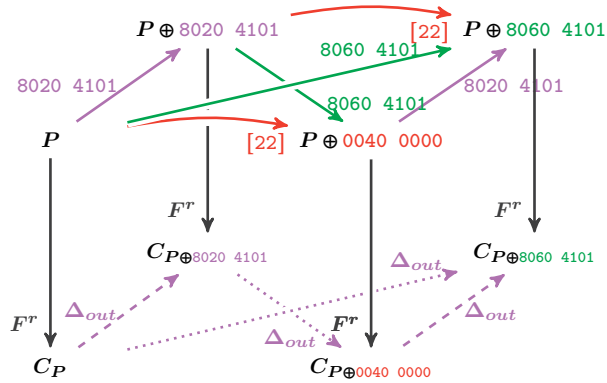
Let $\delta_1 = \Delta_{in_1} \rightarrow \Delta_{out}$ and $\delta_2 = \Delta_{in_2} \rightarrow \Delta_{out}$ be two differentials with the same output difference and with input differences satisfying $\Delta_{in_1} \oplus \Delta_{in_2} = e_i$. Suppose i is a NB/SNBS for both δ_1 and δ_2. Then, once a pair of input pair $\{(P, P \oplus \Delta_{in_1}), (P \oplus e_i, P \oplus \Delta_{in_1} \oplus e_i)\}$ is generated for δ_1, one can re-pair the inputs as $\{(P, P \oplus \Delta_{in_1} \oplus e_i), (P \oplus \Delta_{in_1}, P \oplus e_i)\}$ and obtain a pair of input pair for δ_2. Thus, re-pairing the corresponding ciphertext pairs doubling the number of ciphertext structures.



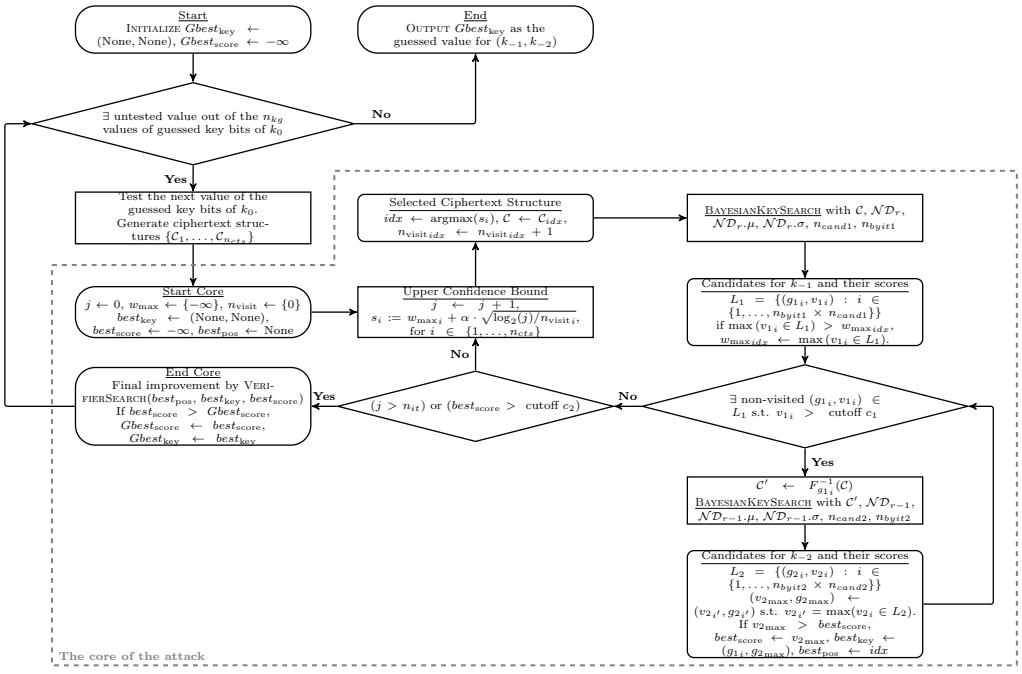
Paired Differentials Sharing the Same Neutral Bits

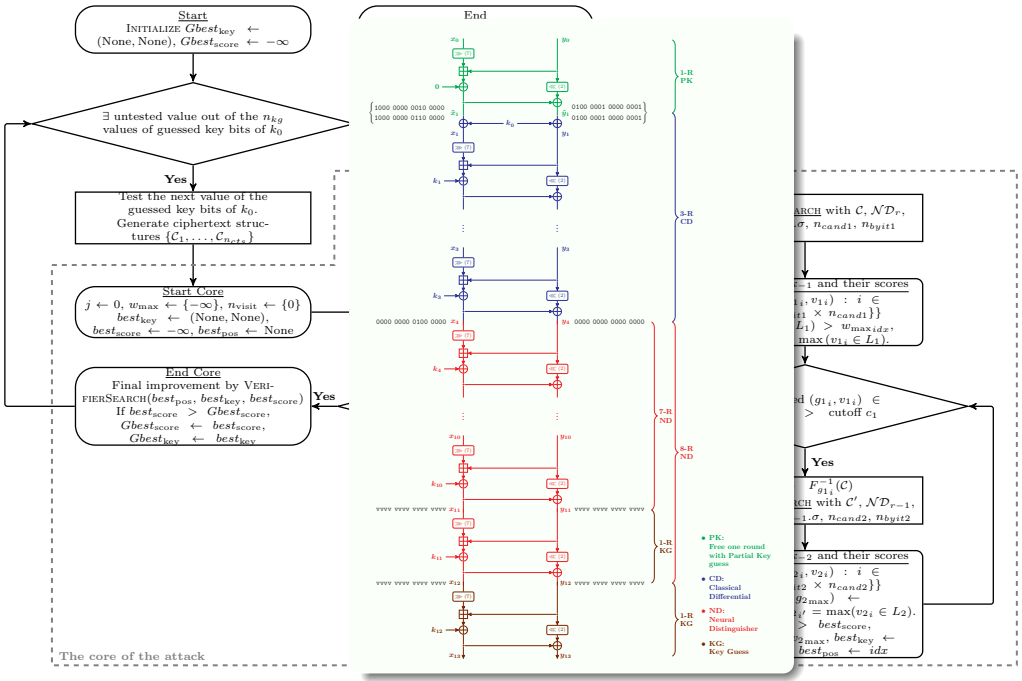
Paired differentials

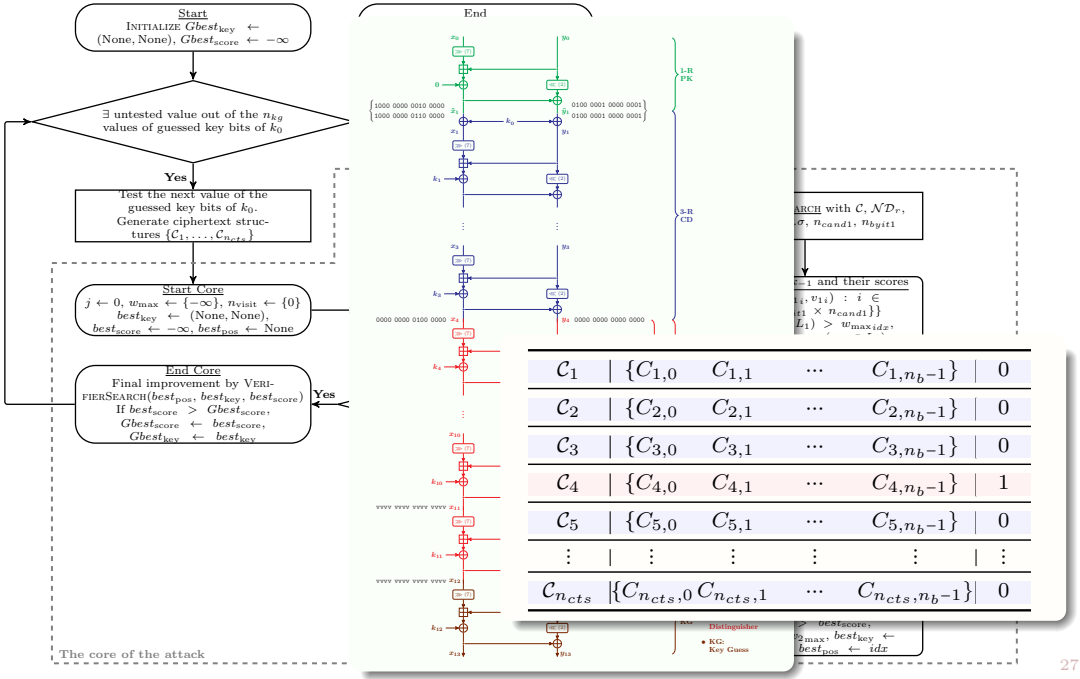
Let $\delta_1 = \Delta_{in_1} \rightarrow \Delta_{out}$ and $\delta_2 = \Delta_{in_2} \rightarrow \Delta_{out}$ be two differentials with the same output difference and with input differences satisfying $\Delta_{in_1} \oplus \Delta_{in_2} = e_i$. Suppose i is a NB/SNBS for both δ_1 and δ_2. Then, once a pair of input pair $\{(P, P \oplus \Delta_{in_1}), (P \oplus e_i, P \oplus \Delta_{in_1} \oplus e_i)\}$ is generated for δ_1, one can re-pair the inputs as $\{(P, P \oplus \Delta_{in_1} \oplus e_i), (P \oplus \Delta_{in_1}, P \oplus e_i)\}$ and obtain a pair of input pair for δ_2. Thus, re-pairing the corresponding ciphertext pairs doubling the number of ciphertext structures.



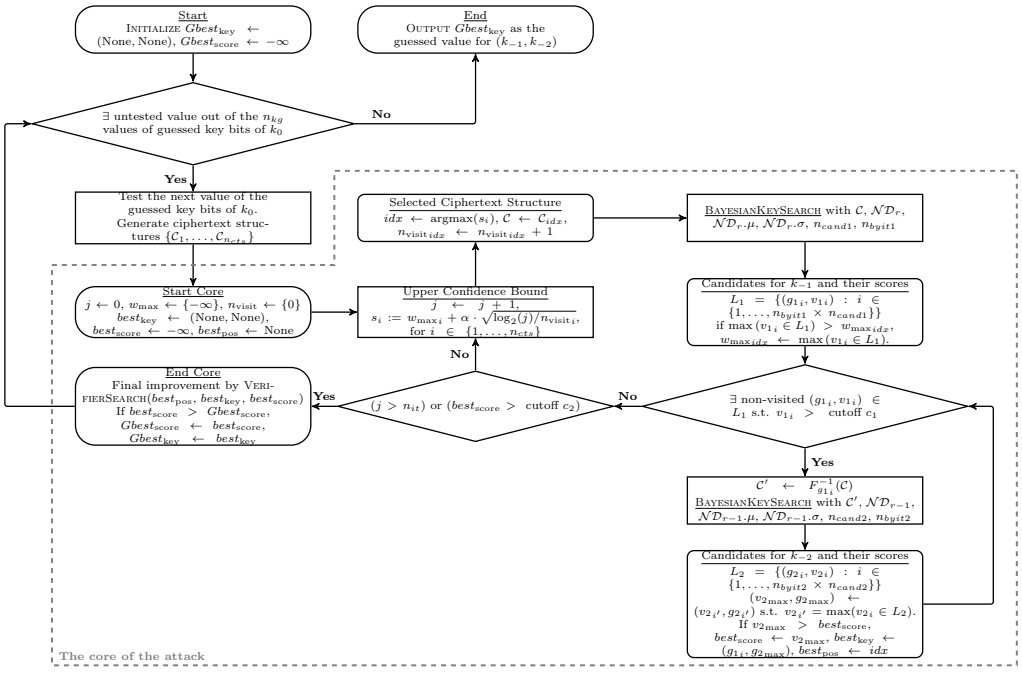
Outline


Start
Initialize $Gbest_{\text{key}} \leftarrow$ (None, None), $Gbest_{\text{score}} \leftarrow -\infty$
End
Output $Gbest_{\text{key}}$ as the guessed value for (k_{-1}, k_{-2})
$\exists$ untested value out of the n_{kg} values of guessed key bits of k_0
No
Yes
Test the next value of the guessed key bits of k_0. Generate ciphertext structures $\{\mathcal{C}_1, \ldots, \mathcal{C}_{n_{cts}}\}$
Selected Ciphertext Structure
$idx \leftarrow \text{argmax}(s_i)$, $\mathcal{C} \leftarrow \mathcal{C}_{idx}$, $n_{\text{visit}\,idx} \leftarrow n_{\text{visit}\,idx} + 1$
BayesianKeySearch with $\mathcal{C}$, $\mathcal{ND}_r$, $\mathcal{ND}_r.\mu$, $\mathcal{ND}_r.\sigma$, n_{cand1}, n_{byit1}
Candidates for k_{-1} and their scores
$L_1 = \{(g_{1_i}, v_{1_i}) : i \in \{1, \ldots, n_{byit1} \times n_{cand1}\}\}$
if $\max(v_{1_i} \in L_1) > w_{\max idx}$, $w_{\max idx} \leftarrow \max(v_{1_i} \in L_1)$.
Start Core
$j \leftarrow 0$, $w_{\max} \leftarrow \{-\infty\}$, $n_{\text{visit}} \leftarrow \{0\}$
$best_{\text{key}} \leftarrow$ (None, None), $best_{\text{score}} \leftarrow -\infty$, $best_{\text{pos}} \leftarrow$ None
Upper Confidence Bound
$j \leftarrow j + 1$,
$s_i := w_{\max i} + \alpha \cdot \sqrt{\log_2(j)/n_{\text{visit}\,i}}$, for $i \in \{1, \ldots, n_{cts}\}$
No
$\exists$ non-visited $(g_{1_i}, v_{1_i}) \in L_1$ s.t. $v_{1_i} >$ cutoff c_1
No
$(j > n_{it})$ or $(best_{\text{score}} >$ cutoff $c_2)$
Yes
End Core
Final improvement by VeriFierSearch($best_{\text{pos}}$, $best_{\text{key}}$, $best_{\text{score}}$)
If $best_{\text{score}} > Gbest_{\text{score}}$, $Gbest_{\text{score}} \leftarrow best_{\text{score}}$, $Gbest_{\text{key}} \leftarrow best_{\text{key}}$
Yes
$\mathcal{C}' \leftarrow F_{g1_i}^{-1}(\mathcal{C})$
BayesianKeySearch with $\mathcal{C}'$, $\mathcal{ND}_{r-1}$, $\mathcal{ND}_{r-1}.\mu$, $\mathcal{ND}_{r-1}.\sigma$, n_{cand2}, n_{byit2}
Candidates for k_{-2} and their scores
$L_2 = \{(g_{2_i}, v_{2_i}) : i \in \{1, \ldots, n_{byit2} \times n_{cand2}\}\}$
$(v_{2\max}, g_{2\max}) \leftarrow (v_{2_{i'}}, g_{2_{i'}})$ s.t. $v_{2_{i'}} = \max(v_{2_i} \in L_2)$.
If $v_{2\max} > best_{\text{score}}$, $best_{\text{score}} \leftarrow v_{2\max}$, $best_{\text{key}} \leftarrow (g_{1_i}, g_{2\max})$, $best_{\text{pos}} \leftarrow idx$
The core of the attack


Start
INITIALIZE $Gbest_{key} \leftarrow$ (None, None), $Gbest_{score} \leftarrow -\infty$
$\exists$ untested value out of the n_{kg} values of guessed key bits of k_0
Yes
Test the next value of the guessed key bits of k_0. Generate ciphertext structures $\{\mathcal{C}_1, \ldots, \mathcal{C}_{n_{cts}}\}$
Start Core
$j \leftarrow 0$, $w_{max} \leftarrow \{-\infty\}$, $n_{visit} \leftarrow \{0\}$
$best_{key} \leftarrow$ (None, None), $best_{score} \leftarrow -\infty$, $best_{pos} \leftarrow$ None
End Core
Final improvement by VERIFIERSEARCH($best_{pos}$, $best_{key}$, $best_{score}$)
If $best_{score} > Gbest_{score}$, $Gbest_{score} \leftarrow best_{score}$, $Gbest_{key} \leftarrow best_{key}$
Yes
The core of the attack
End
1-R PK
3-R CD
7-R ND
8-R ND
1-R KG
1-R KG
PK: Free one round with Partial Key guess
CD: Classical Differential
ND: Neural Distinguisher
KG: Key Guess
ARCH with $\mathcal{C}$, $\mathcal{NC}_r$, σ, n_{cand1}, n_{byit1}
-1 and their scores
$_{1i}, v_{1i}) : i \in$
$_{it1} \times n_{cand1}\}\}$
$L_1) > w_{max idx}$,
$\max(v_{1i} \in L_1)$.
$d(g_{1i}, v_{1i}) \in$
$>$ cutoff c_1
Yes
$F_{g1_i}^{-1}(\mathcal{C})$
RCH with $\mathcal{C}'$, $\mathcal{NC}_{r-1}$,
$1.\sigma$, n_{cand2}, n_{byit2}
2 and their scores
$_i, v_{2i}) : i \in$
$_{it2} \times n_{cand2}\}\}$
$g_{2max}) \leftarrow$
$_{i'} = \max(v_{2i} \in L_2)$.
$> best_{score}$,
$_{2max}$, $best_{key} \leftarrow$
$best_{pos} \leftarrow idx$



$\mathcal{C}_1$	$\{C_{1,0}$	$C_{1,1}$	$\cdots$	$C_{1,n_b-1}\}$	0
$\mathcal{C}_2$	$\{C_{2,0}$	$C_{2,1}$	$\cdots$	$C_{2,n_b-1}\}$	0
$\mathcal{C}_3$	$\{C_{3,0}$	$C_{3,1}$	$\cdots$	$C_{3,n_b-1}\}$	0
$\mathcal{C}_4$	$\{C_{4,0}$	$C_{4,1}$	$\cdots$	$C_{4,n_b-1}\}$	1
$\mathcal{C}_5$	$\{C_{5,0}$	$C_{5,1}$	$\cdots$	$C_{5,n_b-1}\}$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathcal{C}_{n_{cts}}$	$\{C_{n_{cts},0}$	$C_{n_{cts},1}$	$\cdots$	$C_{n_{cts},n_b-1}\}$	0

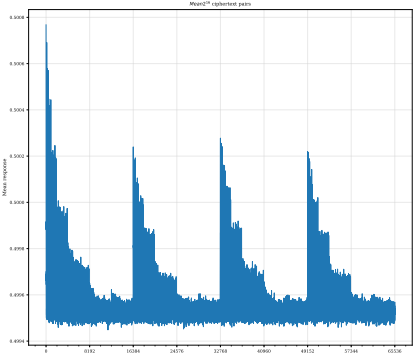

Start
Initialize $Gbest_{\text{key}} \leftarrow$ (None, None), $Gbest_{\text{score}} \leftarrow -\infty$
End
Output $Gbest_{\text{key}}$ as the guessed value for (k_{-1}, k_{-2})
$\exists$ untested value out of the n_{kg} values of guessed key bits of k_0
No
Yes
Test the next value of the guessed key bits of k_0. Generate ciphertext structures $\{\mathcal{C}_1, \ldots, \mathcal{C}_{n_{cts}}\}$
Selected Ciphertext Structure
$idx \leftarrow \text{argmax}(s_i)$, $\mathcal{C} \leftarrow \mathcal{C}_{idx}$, $n_{\text{visit}\,idx} \leftarrow n_{\text{visit}\,idx} + 1$
BayesianKeySearch with $\mathcal{C}$, $\mathcal{ND}_r$, $\mathcal{ND}_r.\mu$, $\mathcal{ND}_r.\sigma$, n_{cand1}, n_{byit1}
Candidates for k_{-1} and their scores
$L_1 = \{(g_{1_i}, v_{1_i}) : i \in \{1, \ldots, n_{byit1} \times n_{cand1}\}\}$
if $\max(v_{1_i} \in L_1) > w_{\max idx}$, $w_{\max idx} \leftarrow \max(v_{1_i} \in L_1)$.
Start Core
$j \leftarrow 0$, $w_{\max} \leftarrow \{-\infty\}$, $n_{\text{visit}} \leftarrow \{0\}$
$best_{\text{key}} \leftarrow$ (None, None), $best_{\text{score}} \leftarrow -\infty$, $best_{\text{pos}} \leftarrow$ None
Upper Confidence Bound
$j \leftarrow j + 1$,
$s_i := w_{\max i} + \alpha \cdot \sqrt{\log_2(j)/n_{\text{visit}\,i}}$, for $i \in \{1, \ldots, n_{cts}\}$
No
$\exists$ non-visited $(g_{1_i}, v_{1_i}) \in L_1$ s.t. $v_{1_i} >$ cutoff c_1
No
$(j > n_{it})$ or $(best_{\text{score}} >$ cutoff $c_2)$
Yes
End Core
Final improvement by VeriFierSearch($best_{\text{pos}}$, $best_{\text{key}}$, $best_{\text{score}}$)
If $best_{\text{score}} > Gbest_{\text{score}}$, $Gbest_{\text{score}} \leftarrow best_{\text{score}}$, $Gbest_{\text{key}} \leftarrow best_{\text{key}}$
Yes
$\mathcal{C}' \leftarrow F_{g1_i}^{-1}(\mathcal{C})$
BayesianKeySearch with $\mathcal{C}'$, $\mathcal{ND}_{r-1}$, $\mathcal{ND}_{r-1}.\mu$, $\mathcal{ND}_{r-1}.\sigma$, n_{cand2}, n_{byit2}
Candidates for k_{-2} and their scores
$L_2 = \{(g_{2_i}, v_{2_i}) : i \in \{1, \ldots, n_{byit2} \times n_{cand2}\}\}$
$(v_{2\max}, g_{2\max}) \leftarrow (v_{2_{i'}}, g_{2_{i'}})$ s.t. $v_{2_{i'}} = \max(v_{2_i} \in L_2)$.
If $v_{2\max} > best_{\text{score}}$, $best_{\text{score}} \leftarrow v_{2\max}$, $best_{\text{key}} \leftarrow (g_{1_i}, g_{2\max})$, $best_{\text{pos}} \leftarrow idx$
The core of the attack

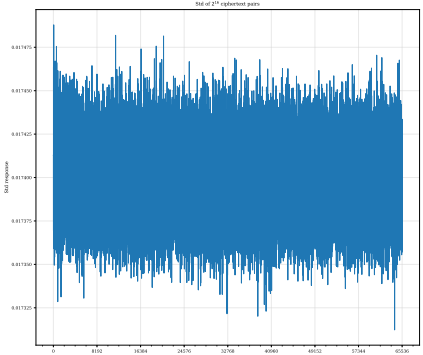
Algorithm 2: BayesianKeySearch Algorithm [C:Gohr19]

Input: Ciphertext structure $\mathcal{C} := \{C_0, \cdots, C_{n_b-1}\}$, an $\mathcal{ND}$ and its wrong key response profile μ and σ, the number n_{cand} of candidates to be generated within each iteration, the number n_{byit} of iterations

Output: The list L of tuples of recommended keys and their scores

$S := \{k_0, \ldots, k_{n_{cand}-1}\} \leftarrow$ choose n_{cand} values at random without replacement from the set of all subkeys.
$L \leftarrow \{\}$
for $t = 1$ *to* n_{byit} **do**
  for $\forall k_i \in S$ **do**
    for $j = 0$ *to* $n_b - 1$ **do**
      $C'_{j,k_i} = F_{k_i}^{-1}(C_j)$
      $v_{j,k_i} = \mathcal{ND}(C'_{j,k_i}), \quad s_{j,k_i} = \log_2(v_{j,k_i}/(1 - v_{j,k_i}))$
    end
    $s_{k_i} = \sum_{j=0}^{n_b-1} s_{j,k_i}$; /* the combined score of k_i */
    $L \leftarrow L \| (k_i, s_{k_i})$
    $m_{k_i} = \sum_{j=0}^{n_b-1} v_{j,k_i}/n_b$
  end
  for $k \in \{0, 1, \cdots, 2^{16} - 1\}$ **do**
    $\lambda_k = \sum_{i=0}^{n_{cand}-1} (m_{k_i} - \mu_{k_i \oplus k})^2 / \sigma^2_{k_i \oplus k}$
  end
  $S \leftarrow \text{argsort}_k(\lambda)[0 : n_{cand} - 1]$; /* Pick n_{cand} keys with the n_{cand} smallest scores */
end
return L





for $k \in \{0, 1, \cdots, 2^{16} - 1\}$ **do**

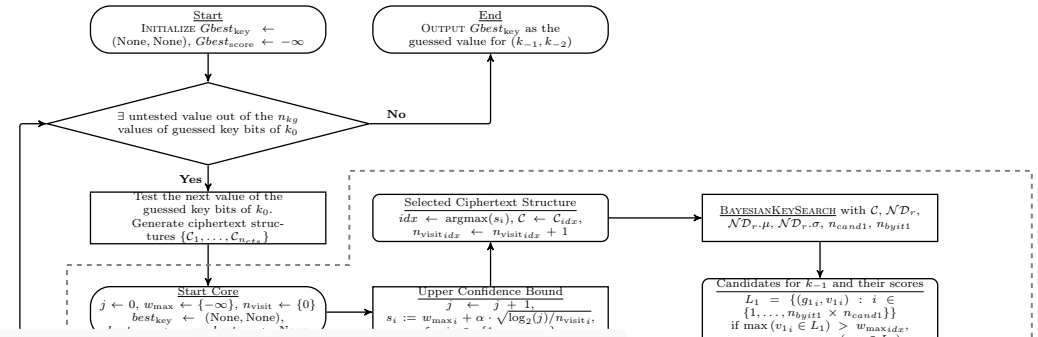
  $\lambda_k = \sum_{i=0}^{n_{cand}-1} (m_{k_i} - \mu_{k_i \oplus k})^2 / \sigma^2_{k_i \oplus k}$

end

$S \leftarrow \text{argsort}_k(\lambda)[0 : n_{cand} - 1]$; `/* Pick` n_{cand} `keys with the` n_{cand} `smallest scores */`

end

return L



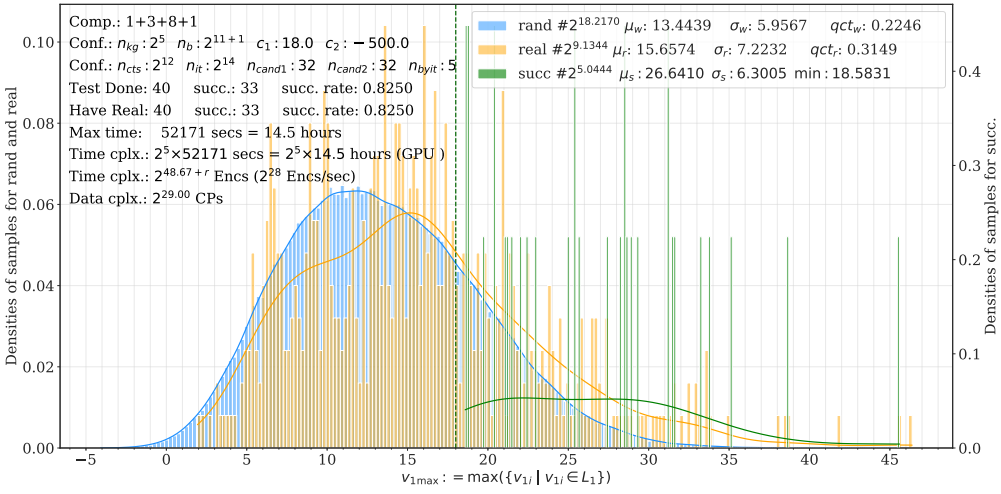
Algorithm 2: BayesianKeySearch Algorithm [C:Gohr19]

Input: Ciphertext structure $\mathcal{C} := \{C_0, \cdots, C_{n_b-1}\}$, an $\mathcal{ND}$ and its wrong key response profile μ and σ, the number n_{cand} of candidates to be generated within each iteration, the number n_{byit} of iterations

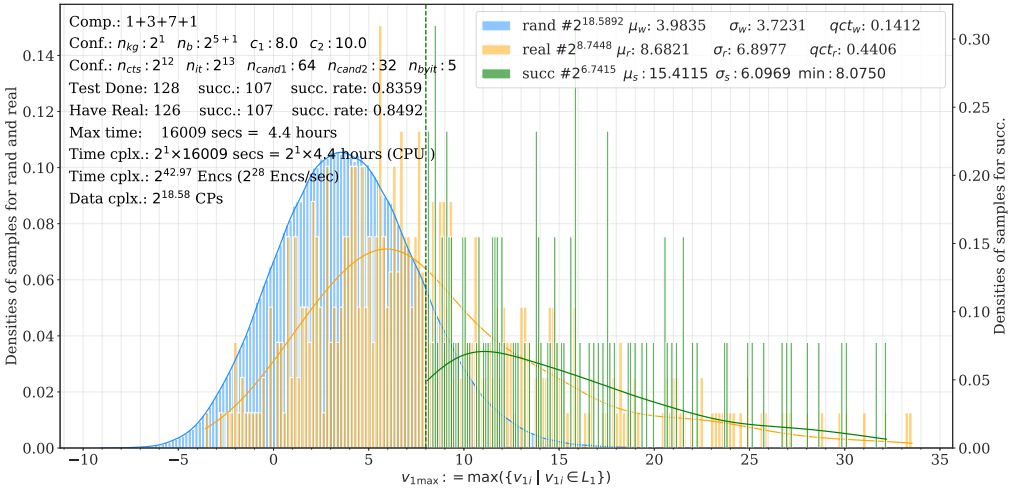
Output: The list L of tuples of recommended keys and their scores

$S := \{k_0, \ldots, k_{n_{cand}-1}\} \leftarrow$ choose n_{cand} values at random without replacement from the set of all subkeys.
$L \leftarrow \{\}$
for $t = 1$ *to* n_{byit} **do**
  for $\forall k_i \in S$ **do**
    for $j = 0$ *to* $n_b - 1$ **do**
      $C'_{j,k_i} = F^{-1}_{k_i}(C_j)$
      $v_{j,k_i} = \mathcal{ND}(C'_{j,k_i})$, $s_{j,k_i} = \log_2(v_{j,k_i}/(1 - v_{j,k_i}))$
    end
    $s_{k_i} = \sum_{j=0}^{n_b-1} s_{j,k_i}$; /* the combined score of k_i */
    $L \leftarrow L \| (k_i, s_{k_i})$
    $m_{k_i} = \sum_{j=0}^{n_b-1} v_{j,k_i}/n_b$
  end
  for $k \in \{0, 1, \cdots, 2^{16} - 1\}$ **do**
    $\lambda_k = \sum_{i=0}^{n_{cand}-1} (m_{k_i} - \mu_{k_i \oplus k})^2 / \sigma^2_{k_i \oplus k}$
  end
  $S \leftarrow \text{argsort}_k(\lambda)[0 : n_{cand} - 1]$; /* Pick n_{cand} keys with the n_{cand} smallest scores */
end

Attack information and distributions of $v_{1\max}$ for attack $\mathcal{A}_I^{\text{SPECK}_{13R}}$



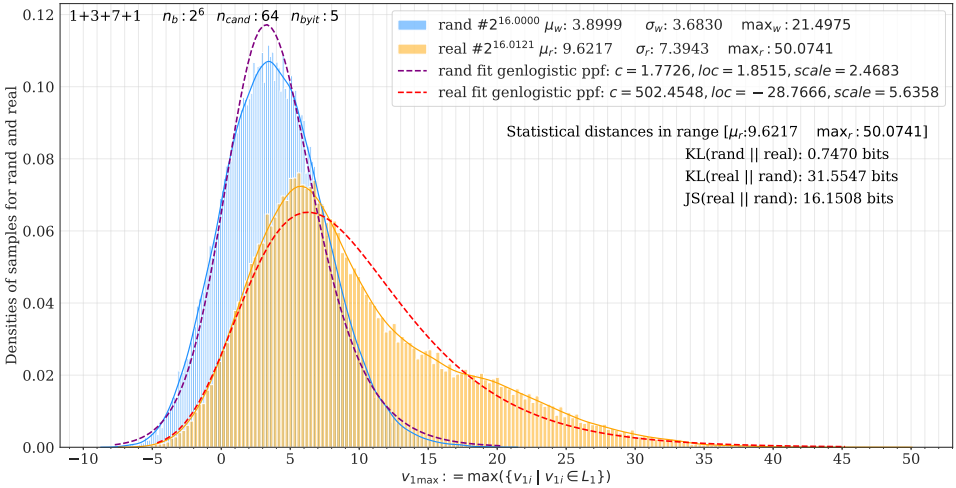
Attack information and distributions of $v_{1\max}$ for attack $\mathcal{A}_{II}^{\text{SPECK}_{12R}}$



Outline

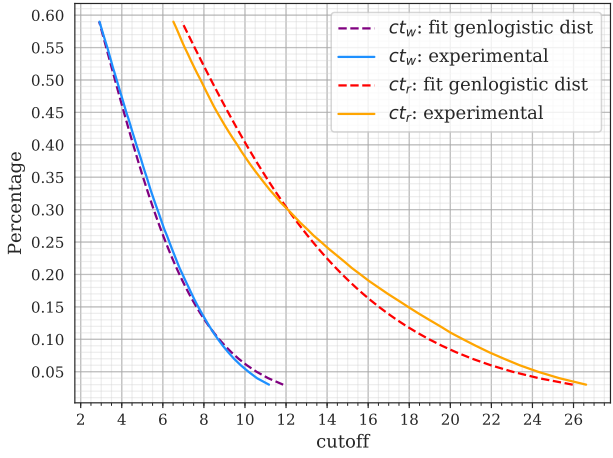
Investigations on $\mathcal{D}_r^{v_{1\max}}$ and $\mathcal{D}_w^{v_{1\max}}$ for attack $\mathcal{A}^{\text{Speck}_{12R}}$

Sampling 2^{16} correct/wrong ciphertext structures to study the distributions $\mathcal{D}_r^{v_{1\max}}$ and $\mathcal{D}_w^{v_{1\max}}$ involved in attack $\mathcal{A}^{\text{Speck}_{12R}}$



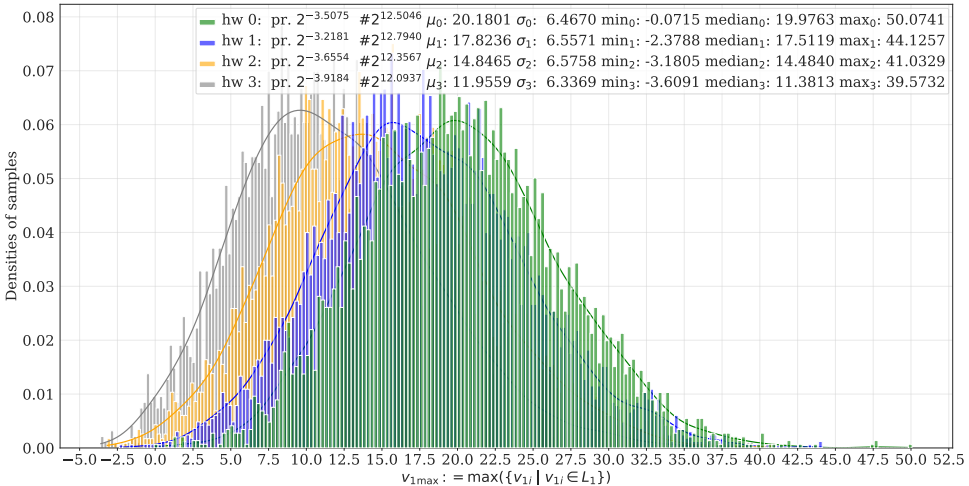
Investigations on $\mathcal{D}_r^{v_1\max}$ and $\mathcal{D}_w^{v_1\max}$ for attack $\mathcal{A}^{\text{Speck}_{12R}}$

Percentage of samples passing various cutoffs

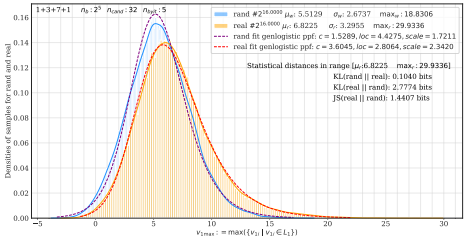


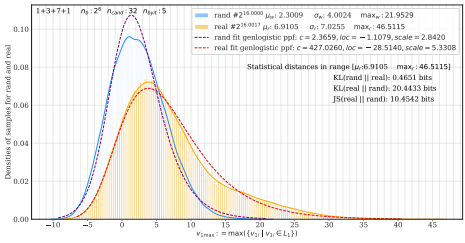
Investigations on $\mathcal{D}_r^{v_{1\max}}$ and $\mathcal{D}_w^{v_{1\max}}$ for attack $\mathcal{A}^{\text{Speck}_{12R}}$

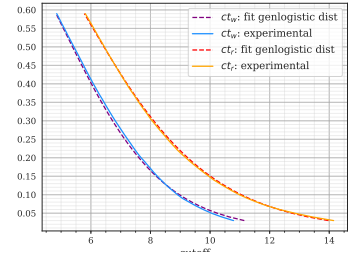
Distribution of combined responses using correct ciphertext structures when the corresponding recommended subkey has Hamming distance hw with the real subkey

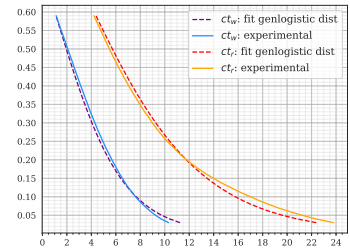


Influence on $\mathcal{D}_r^{v_1\max}$ and $\mathcal{D}_w^{v_1\max}$ of various parameters



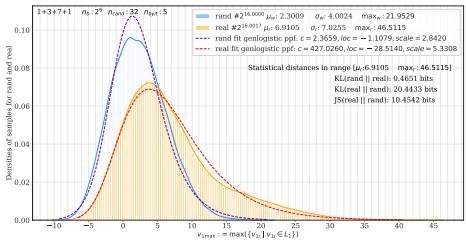


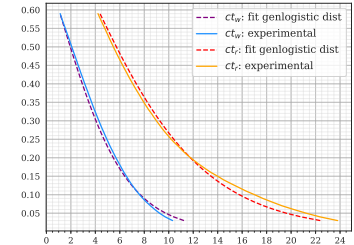


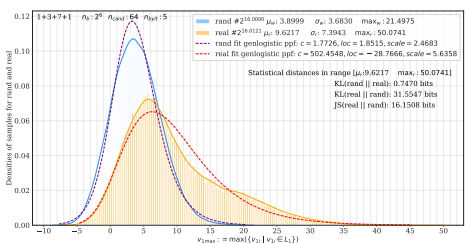


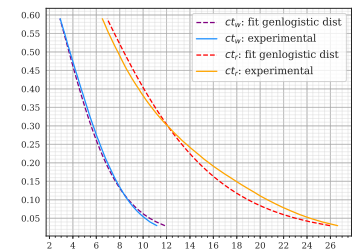
Distributions $\mathcal{D}_r^{v_1\max}$ and $\mathcal{D}_w^{v_1\max}$ when using 5 vs. 6 NBs

Influence on $\mathcal{D}_r^{v_1\max}$ and $\mathcal{D}_w^{v_1\max}$ of various parameters









Distributions $\mathcal{D}_r^{v_1\max}$ and $\mathcal{D}_w^{v_1\max}$ when $n_{cand} = 32$ vs. $n_{cand} = 64$

Rules of Thumb for Turning Parameters for the Key Recovery Attacks

Observation

Suppose in the above attack framework, the probability of the prepended differential is p, the number of ciphertext structures is n_{cts}. Denote the attack success probability by P_s.
Note that $P_s \leq 1 - (1 - p \cdot q)^{n_{cts}}$, where q is the probability for the response $v_{1\max}$ from a correct ciphertext structure pass the cutoff c_1, *i.e.*, $q = \Pr_{\mathcal{C}_r}[v_{1\max} \geq c_1]$, where $\mathcal{C}_r$ is space of correct ciphertext structures. Thus, the following relation should be fulfilled:

$$n_{cts} \geq \frac{\log_2(1 - P_s)}{\log_2(1 - p \cdot q)}.$$

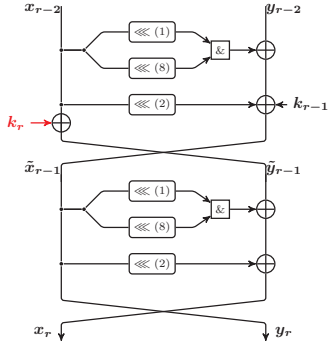
For given n_{cts}, p, and P_s, the cutoff c_1 should be chosen such that

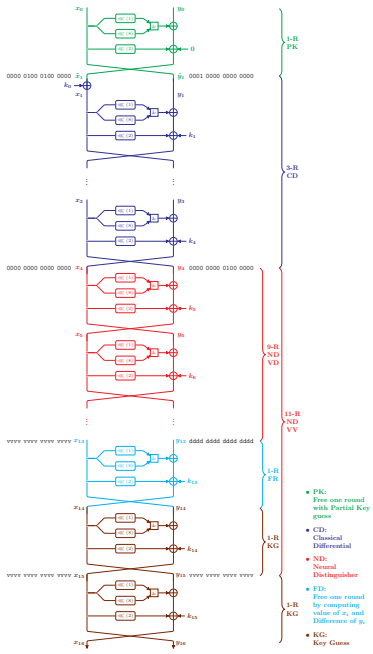
$$c_1 \leq Q(1 - \frac{1 - (1 - Ps)^{\frac{1}{n_{cts}}}}{p}),$$

where $Q(\cdot)$ is the quantile function of the distribution of $v_{1\max}$ corresponding to correct ciphertext structures, *i.e.*, $\mathcal{D}_r^{v_{1\max}}$.

Outline

#R	Name	Network	Accuracy	True Positive Rate	True Negative Rate
6	$\mathcal{DD}_{\mathbf{DD}}^{\text{SIMON}_{6R}}$	DDT	0.9918	0.9995	0.9841
7		ResNet	$0.9823 \pm 1.2 \times 10^{-4}$	$0.9996 \pm 2.7 \times 10^{-5}$	$0.9650 \pm 2.3 \times 10^{-4}$
	$\mathcal{ND}_{\mathbf{VV}}^{\text{SIMON}_{7R}}$	SENet	$0.9802 \pm 1.3 \times 10^{-4}$	$0.9987 \pm 4.2 \times 10^{-5}$	$0.9617 \pm 2.4 \times 10^{-4}$
7	$\mathcal{DD}_{\mathbf{DD}}^{\text{SIMON}_{7R}}$	DDT	0.8465	0.8641	0.8288
8	$\mathcal{ND}_{\mathbf{VV}}^{\text{SIMON}_{8R}}$	SENet	$0.8150 \pm 4.2 \times 10^{-4}$	$0.8418 \pm 5.5 \times 10^{-4}$	$0.7882 \pm 5.1 \times 10^{-4}$
		ResNet	$0.7912 \pm 4.2 \times 10^{-4}$	$0.8041 \pm 5.5 \times 10^{-4}$	$0.7783 \pm 6.2 \times 10^{-4}$
8	$\mathcal{DD}_{\mathbf{DD}}^{\text{SIMON}_{8R}}$	DDT	0.6628	0.5781	0.7476
8	$\mathcal{ND}_{\mathbf{VD}}^{\text{SIMON}_{8R}}$	SENet	$0.6587 \pm 4.8 \times 10^{-4}$	$0.5586 \pm 7.4 \times 10^{-4}$	$0.7588 \pm 5.6 \times 10^{-4}$
9	$\mathcal{ND}_{\mathbf{VV}}^{\text{SIMON}_{9R}}$	SENet	$0.6515 \pm 5.3 \times 10^{-4}$	$0.5334 \pm 7.0 \times 10^{-4}$	$0.7695 \pm 5.7 \times 10^{-4}$
		ResNet	$0.6296 \pm 4.5 \times 10^{-4}$	$0.5164 \pm 6.3 \times 10^{-4}$	$0.7429 \pm 5.5 \times 10^{-4}$
9	$\mathcal{DD}_{\mathbf{DD}}^{\text{SIMON}_{9R}}$	DDT	0.5683	0.4691	0.6674
9	$\mathcal{ND}_{\mathbf{VD}}^{\text{SIMON}_{9R}}$	SENet	$0.5657 \pm 4.9 \times 10^{-4}$	$0.4748 \pm 7.1 \times 10^{-4}$	$0.6565 \pm 6.6 \times 10^{-4}$
10	$\mathcal{ND}_{\mathbf{VV}}^{\text{SIMON}_{10R}}$ +	SENet	$0.5610 \pm 4.5 \times 10^{-4}$	$0.4761 \pm 6.0 \times 10^{-4}$	$0.6460 \pm 7.2 \times 10^{-4}$
10	$\mathcal{DD}_{\mathbf{DD}}^{\text{SIMON}_{10R}}$	DDT	0.5203	0.5002	0.5404
11	$\mathcal{ND}_{\mathbf{VV}}^{\text{SIMON}_{11R}}$	SENet	$0.5174 \pm 5.3 \times 10^{-4}$	$0.5041 \pm 7.1 \times 10^{-4}$	$0.5307 \pm 7.9 \times 10^{-4}$


x_{r-2}
y_{r-2}
(1)
(8)
(2)
&
k_r
k_{r-1}
$\tilde{x}_{r-1}$
$\tilde{y}_{r-1}$
(1)
(8)
(2)
&
x_r
y_r


0000 0100 0100 0000
0000 0000 0000 0000
0000 0000 0000 0100
vvvv vvvv vvvv vvvv
dddd dddd dddd dddd
vvvv vvvv vvvv vvvv
1-R PK
9-R CD
9-R ND VD
11-R ND VV
1-R FR
1-R KG
1-R KG
PK: Free one round with Partial Key guess
CD: Classical Differential
ND: Neural Distinguisher
FD: Free one round by computing value of x_i and Difference of y_i
KG: Key Guess

Outline

Target	#R	Time (#Enc)	Data (#CP)	Succ. Rate	Weak keys	Configure	Ref.
SPECK32/64	11	2^{46}	2^{14}	-	2^{64}	1+6 +4	[**SAC:Dinur14**]
		$2^{38^\star}$	$2^{13.6}$	0.52	2^{64}	1+2 +7+1	[**C:Gohr19**]
	12	2^{51}	2^{19}	-	2^{64}	1+7 +4	[**SAC:Dinur14**]
		$2^{43.40^\star}$	$2^{22.97}$	0.40	2^{64}	1+2 +8+1	[**C:Gohr19**]
		$2^{44.89^\star}$	$2^{22.00}$	**0.86**	2^{64}	1+2 +8+1	[**This**]
		$\mathbf{2^{42.97^\star}}$	$\mathbf{2^{18.58}}$	**0.83**	2^{63}	1+3 +7+1	[**This**]
	13	2^{57}	2^{25}	-	2^{64}	1+8 +4	[**SAC:Dinur14**]
		$\mathbf{2^{48.67^\star+r}}$	2^{29}	**0.82**	2^{63}	1+3 +8+1	[**This**]
	14	$2^{62.47}$	$2^{30.47}$	-	2^{64}	1+9 +4	[**EPRINT:SonHuaYan16**]
SIMON32/64	16	$2^{26.48}$	$2^{29.48}$	0.62	2^{64}	2+12 +2	[**EPRINT:AlkLau13**]
		$2^{41.81^\star+r}$	$\mathbf{2^{21}}$	0.49	2^{64}	1+3 +11+1	[**This**]
	18	$2^{46.00}$	$2^{31.2}$	0.63	2^{64}	1+13 +4	[**FSE:ALLW14**]
	21	$2^{55.25}$	$2^{31.0}$	-	2^{64}	4+13 +4	[**EPRINT:WWJZ14**]

- Not available. * Under the assumption that one second equals the time of 2^{28} executions of SPECK32/64 or SIMON32/64 on a CPU.

$r : \log_2(cpu/gpu)$, where *cpu* and *gpu* are the CPU and GPU time running an attack, respectively. In our computing systems, $r = 2.4$ (The worse case execution time of the core of the 12-round attack on SPECK32/64 (without guessing the one key bit of k_0) took 6637 and 1265 seconds on CPU and GPU, respectively).

Conclusions

- Differential-neural cryptanalysis should work in general on modern ciphers. Still, their advantages might be easier to show on ciphers whose differential-like properties can not be accurately evaluated using existing tools.
- Enhancing the connection between traditional cryptanalysis techniques and machine-learning approaches is helpful for achieving better cryptanalysis results.
- These generalized neutral bits are not intrinsically linked to neural network-based cryptanalysis but are expected to be useful for converting a wider range of weak distinguishers to competitive key-recovery attacks.
- The rules of thumb on turning parameters for the key-recovery phase are far from perfect. A rigorous theoretical model on the relation between attack parameters, attack complexity, and success probability is missing, the building of which is left as future work.

Thanks for your attention!