Clustering Effect in Simon and Simeck

Gaëtan Leurent[1], Clara Pernot[1] and André Schrottenloher[2]

[1]Inria, Paris
[2]CWI, Amsterdam

November 2021

Overview

Introduction of two lightweight block ciphers by NSA researchers in 2013:

- **Simon** optimized in hardware [BTSWSW, DAC'15]
- **Speck** optimized in software [BTSWSW, DAC'15]

Overview

Introduction of two lightweight block ciphers by NSA researchers in 2013:

- **Simon** optimized in hardware [BTSWSW, DAC'15]
- **Speck** optimized in software [BTSWSW, DAC'15]

Attempt of ISO standardization...
But some experts were **suspicious** about:

→ the absence of rationale

→ NSA's previous involvement in the creation and promotion of backdoored cryptographic algorithms

→ the lack of clear need for standardisation of the new ciphers

More than **70 papers** study **Simon** and **Speck**!

Overview

Introduction of two lightweight block ciphers by NSA researchers in 2013:

- **Simon** optimized in hardware [BTSWSW, DAC'15]
- **Speck** optimized in software [BTSWSW, DAC'15]

Attempt of ISO standardization...
But some experts were **suspicious** about:

→ the absence of rationale

→ NSA's previous involvement in the creation and promotion of backdoored cryptographic algorithms

→ the lack of clear need for standardisation of the new ciphers

More than **70 papers** study **Simon** and **Speck**!

⇒ A variant of **Simon** and **Speck**: **Simeck**. [YZSAG, CHES'15]

Summary of previous and new attacks

Cipher	Rounds	Attacked	Ref	Note
Simeck48/96	36	30	[QCW'16]	Linear † ‡
		32	**New**	Linear
Simeck64/128	44	37	[QCW'16]	Linear † ‡
		42	**New**	Linear
Simon96/96	52	37	[WWJZ'18]	Differential
		43	**New**	Linear
Simon96/144	54	38	[CW'16]	Linear
		45	**New**	Linear
Simon128/128	68	50	[WWJZ'18]	Differential
		53	**New**	Linear
Simon128/192	69	51	[WWJZ'18]	Differential
		55	**New**	Linear
Simon128/256	72	53	[CW'16]	Linear
		56	**New**	Linear

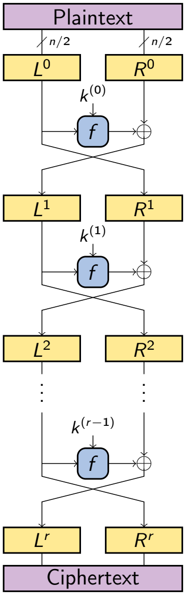
† The advantage is too low to do a key recovery.

‡ Attack use the duality between linear and differential distinguishers.

Table of contents

Table of contents

Feistel cipher



A **Feistel network** is characterized by:

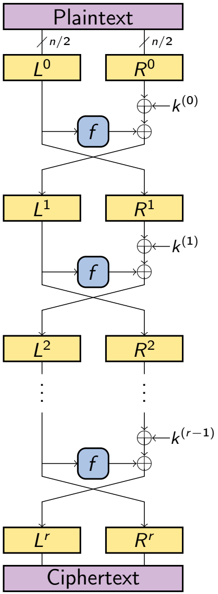
- its block size: n
- its key size: κ
- its number of round: r
- its round function: f

For each round $i = 0, \ldots, r-1$:

$$\begin{cases} R^{i+1} & = & L^i \\ L^{i+1} & = & R^i \oplus f(L^i, k^{(i)}) \end{cases}$$

Example: Data Encryption Standard (DES).

Feistel cipher



A **Feistel network** is characterized by:

- its block size: n
- its key size: κ
- its number of round: r
- its round function: f

For each round $i = 0, \ldots, r-1$:

$$\begin{cases} R^{i+1} & = & L^i \\ L^{i+1} & = & R^i \oplus f(L^i, k^{(i)}) \end{cases}$$

Example: Data Encryption Standard (DES).

Simon, Speck and Simeck

→ **Simon** is a Feistel network with a **quadratic** round function:

$$f(x) = ((x \lll 8) \wedge (x \lll 1)) \oplus (x \lll 2)$$

and a linear key schedule. [BTSWSW'15]

→ **Speck** is an Add-Rotate-XOR (ARX) cipher:

$$R_k(x, y) = \big(((x \lll \alpha) \boxplus y) \oplus k, (y \lll \beta) \oplus ((x \lll \alpha) \boxplus y) \oplus k\big)$$

which reuses its **round function** R_k in the **key schedule**.
[BTSWSW'15]

Simon, Speck and Simeck

→ **Simon** is a Feistel network with a **quadratic** round function:

$$f(x) = ((x \lll 8) \wedge (x \lll 1)) \oplus (x \lll 2)$$

and a linear key schedule. [BTSWSW'15]

→ **Speck** is an Add-Rotate-XOR (ARX) cipher:

$$R_k(x, y) = \big(((x \lll \alpha) \boxplus y) \oplus k, (y \lll \beta) \oplus ((x \lll \alpha) \boxplus y) \oplus k\big)$$

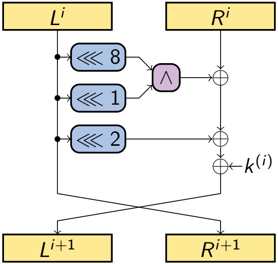
which reuses its **round function** R_k in the **key schedule**.
[BTSWSW'15]

→ **Simeck** is a Feistel network with a **quadratic** round function:

$$f(x) = ((x \lll 5) \wedge x) \oplus (x \lll 1)$$

which reuses its **round function** f in the **key schedule**. [YZSAG'15]

Simon and Simeck

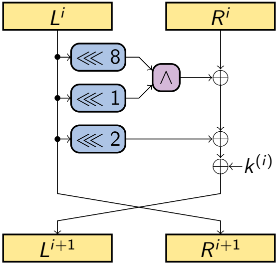


Simon round function

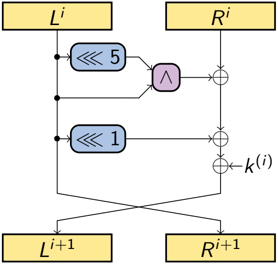
n (block size)	32	48		64		96		128		
κ (key size)	64	72	96	96	128	96	144	128	192	256
r (rounds)	32	36	36	42	44	52	54	68	69	72

$\rightarrow$ Linear key schedule.

Simon and Simeck



Simon round function



Simeck round function

n (block size)	32	48		64		96		128		
κ (key size)	64	72	96	96	128	96	144	128	192	256
r (rounds)	32	36	36	42	44	52	54	68	69	72

n	32	48	64
κ	64	96	128
r	32	36	44

→ Linear key schedule.

→ Non-linear key schedule which reuses f.

Table of contents

Differential Cryptanalysis [BS, CRYPTO'90]

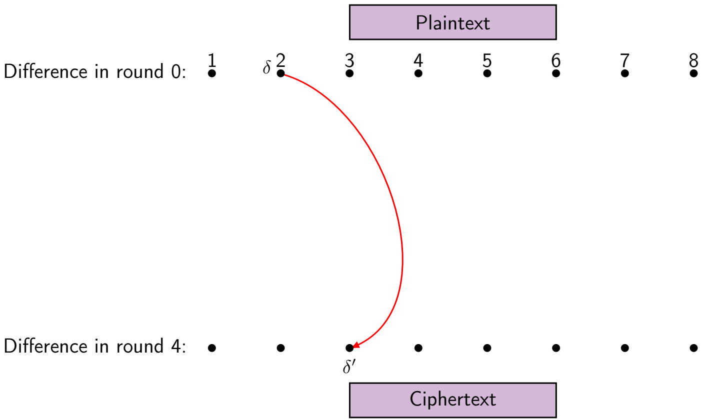
Plaintext

Difference in round 0: 1 2 3 4 5 6 7 8

Difference in round 4:

Ciphertext

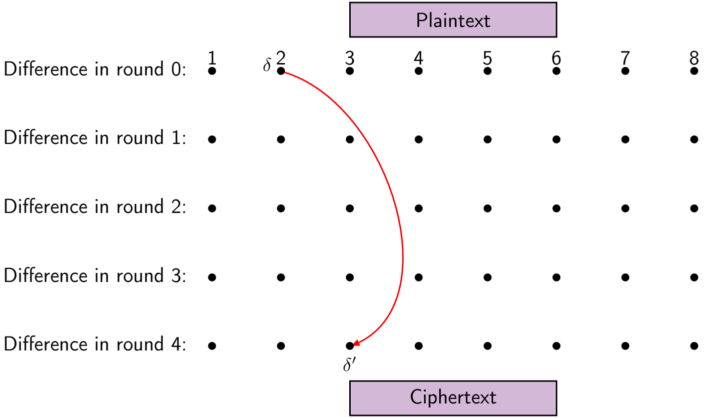
Differential Cryptanalysis [BS, CRYPTO'90]



A **differential** is a pair (δ, δ') such that:

$$\boxed{\Pr_{k,x}[E_k(x) \oplus E_k(x \oplus \delta) = \delta'] \gg 2^{-n}}$$

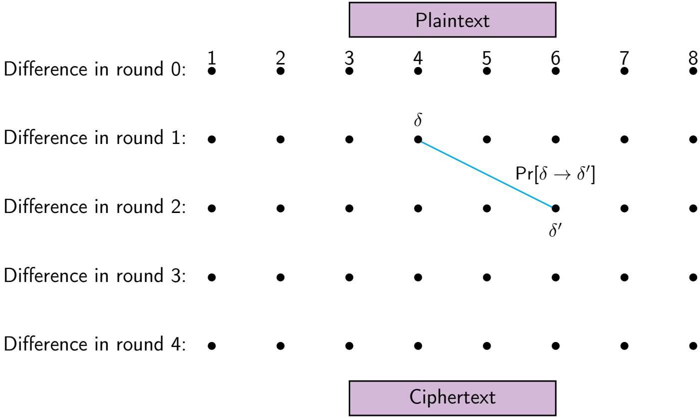
Differential Cryptanalysis [BS, CRYPTO'90]



A **differential** is a pair (δ, δ') such that:

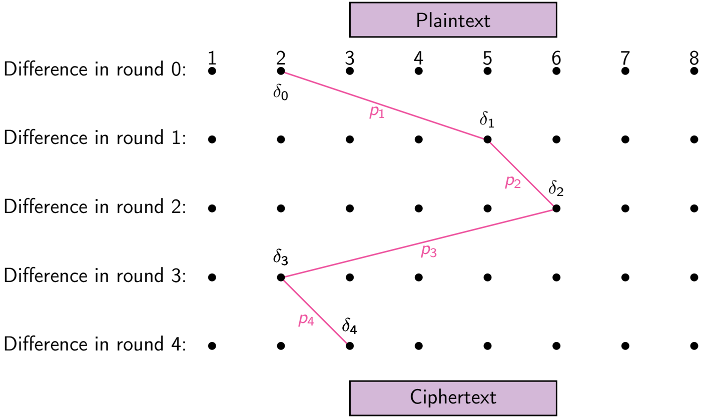
$$\boxed{\Pr_{k,x}[E_k(x) \oplus E_k(x \oplus \delta) = \delta'] \gg 2^{-n}}$$

Differential Cryptanalysis [BS, CRYPTO'90]



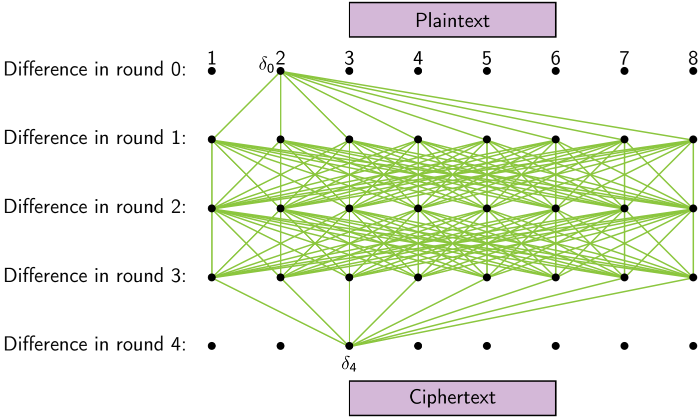
$$\boxed{\Pr[\delta \to \delta'] = \Pr_x[R(x) \oplus R(x \oplus \delta) = \delta']}$$

Differential Cryptanalysis [BS, CRYPTO'90]



$$\boxed{\Pr[\delta_0 \to \delta_1 \to \ldots \to \delta_4] = \mathbf{p_1} \times \mathbf{p_2} \times \mathbf{p_3} \times \mathbf{p_4}}$$

Differential Cryptanalysis [BS, CRYPTO'90]



$$\boxed{Pr[\delta_0 \overset{4}{\rightsquigarrow} \delta_4] = \sum_{\delta_1,\delta_2,\delta_3} \prod_{i=1}^{4} \mathrm{Pr}[\delta_{i-1} \to \delta_i]}$$

Differential Cryptanalysis

Differential: a pair (δ, δ') such that

$$\Pr_{k,x}[E_k(x) \oplus E_k(x \oplus \delta) = \delta'] \gg 2^{-n}$$

With independent round keys:

$\rightarrow$ for 1 round:

$$\Pr[\delta \rightarrow \delta'] = \Pr_x[R(x) \oplus R(x \oplus \delta) = \delta']$$

$\rightarrow$ for r rounds:

$$\boxed{\Pr[\delta_0 \overset{r}{\rightsquigarrow} \delta_r] = \sum_{\delta_1, \delta_2, \ldots \delta_{r-1}} \prod_{i=1}^{r} \Pr[\delta_{i-1} \rightarrow \delta_i]}$$

Differential Cryptanalysis

Differential: a pair (δ, δ') such that

$$\Pr_{k,x}[E_k(x) \oplus E_k(x \oplus \delta) = \delta'] \gg 2^{-n}$$

With independent round keys:

$\rightarrow$ for 1 round:

$$\Pr[\delta \rightarrow \delta'] = \Pr_x[R(x) \oplus R(x \oplus \delta) = \delta']$$

$\rightarrow$ for r rounds:

$$\boxed{\Pr[\delta_0 \overset{r}{\rightsquigarrow} \delta_r] = \sum_{\delta_1, \delta_2, \dots \delta_{r-1}} \prod_{i=1}^{r} \Pr[\delta_{i-1} \rightarrow \delta_i]}$$

Linear Cryptanalysis

Linear Approx: a pair (α, α') such that

$$|\Pr_x[x \cdot \alpha = E_k(x) \cdot \alpha'] - 1/2| \gg 2^{-n/2}$$

With independent round keys:

$\rightarrow$ for 1 round:

$$c(\alpha \rightarrow \alpha') = 2 \Pr_x[x \cdot \alpha = R(x) \cdot \alpha'] - 1$$

$\rightarrow$ for r rounds:

$$\boxed{\mathrm{ELP}(\alpha_0 \overset{r}{\rightsquigarrow} \alpha_r) = \sum_{\alpha_1, \alpha_2, \dots \alpha_{r-1}} \prod_{i=1}^{r} c^2(\alpha_{i-1} \rightarrow \alpha_i)}$$

Differential and Linear Distinguishers

- **Differential distinguisher:**

We collect $D = \mathcal{O}(1/\Pr[\delta \rightsquigarrow \delta'])$ pairs $(P, P \oplus \delta)$ and compute:

$$Q = \#\{P : E(P) \oplus E(P \oplus \delta) = \delta'\}$$

→ $Q \approx D \times \Pr[\delta \rightsquigarrow \delta']$ for the cipher

→ $Q \approx D \times 2^{-n}$ for a random permutation

Differential and Linear Distinguishers

- **Differential distinguisher:**

We collect $D = \mathcal{O}(1/\Pr[\delta \rightsquigarrow \delta'])$ pairs $(P, P \oplus \delta)$ and compute:

$$Q = \#\{P : E(P) \oplus E(P \oplus \delta) = \delta'\}$$

→ $Q \approx D \times \Pr[\delta \rightsquigarrow \delta']$ for the cipher
→ $Q \approx D \times 2^{-n}$ for a random permutation

- **Linear distinguisher:**

We collect $D = \mathcal{O}(1/\mathrm{ELP}[\alpha \rightsquigarrow \alpha'])$ pairs (P, C) and compute:

$$Q = (\#\{P, C : P \cdot \alpha \oplus C \cdot \alpha' = 0\} - \#\{P, C : P \cdot \alpha \oplus C \cdot \alpha' = 1\})$$

→ $Q^2 \approx D \times ELP[\alpha \rightsquigarrow \alpha']$ for the cipher
→ $Q^2 \approx D \times 2^{-n}$ for a random permutation

Differential and Linear Distinguishers

- **Differential distinguisher:**

We collect $D = \mathcal{O}(1/\Pr[\delta \rightsquigarrow \delta'])$ pairs $(P, P \oplus \delta)$ and compute:

$$Q = \#\{P : E(P) \oplus E(P \oplus \delta) = \delta'\}$$

$\rightarrow$ $Q \approx D \times \Pr[\delta \rightsquigarrow \delta']$ for the cipher
$\rightarrow$ $Q \approx D \times 2^{-n}$ for a random permutation

- **Linear distinguisher:**

We collect $D = \mathcal{O}(1/\mathrm{ELP}[\alpha \rightsquigarrow \alpha'])$ pairs (P, C) and compute:

$$Q = (\#\{P, C : P \cdot \alpha \oplus C \cdot \alpha' = 0\} - \#\{P, C : P \cdot \alpha \oplus C \cdot \alpha' = 1\})$$

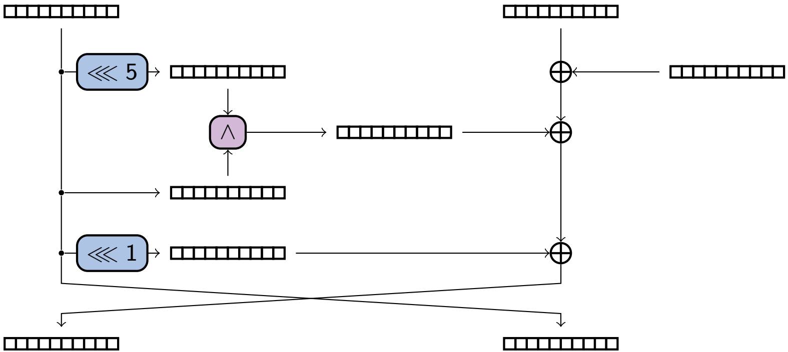
$\rightarrow$ $Q^2 \approx D \times ELP[\alpha \rightsquigarrow \alpha']$ for the cipher
$\rightarrow$ $Q^2 \approx D \times 2^{-n}$ for a random permutation

How to find stronger distinguishers for Simon and Simeck?

Table of contents

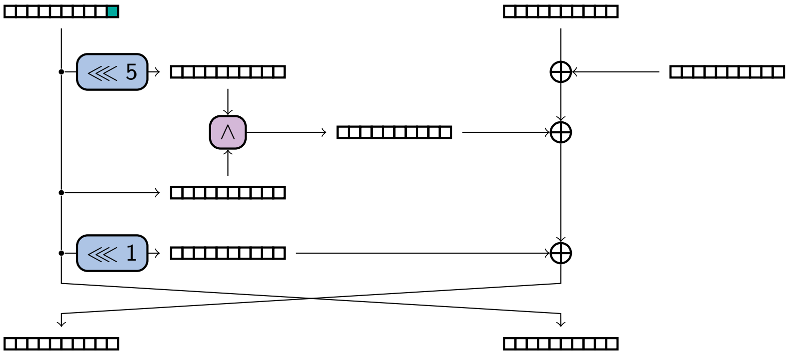
Table of contents

Probability of transition through f



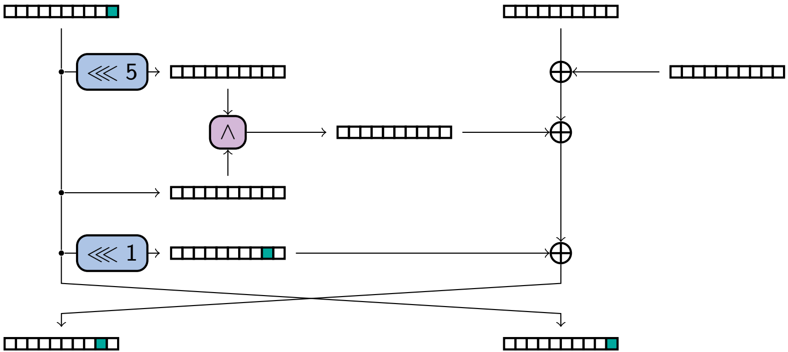
Probability of transition through f

Consider a difference $\alpha = 1$ on the left part:



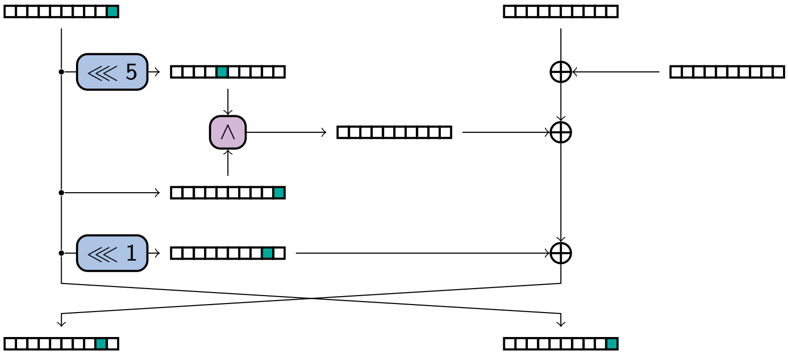
Probability of transition through f

Consider a difference $\alpha = 1$ on the left part:



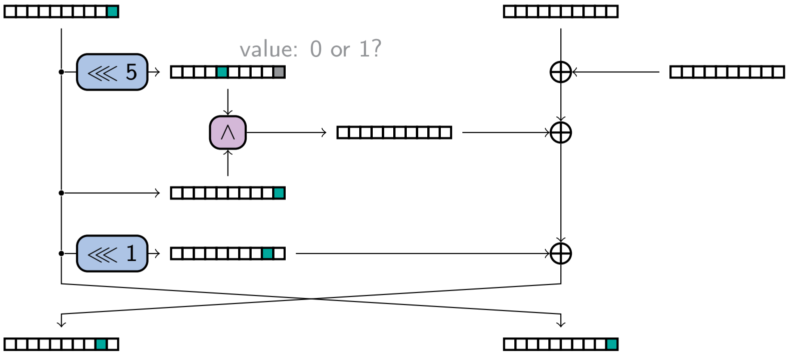
Probability of transition through f

Consider a difference $\alpha = 1$ on the left part:



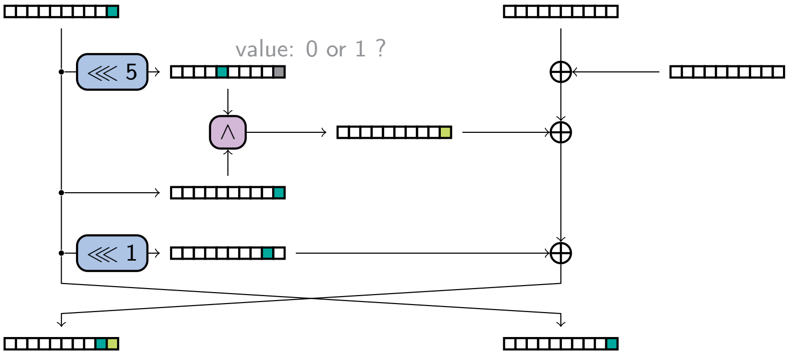
Probability of transition through f

Consider a difference $\alpha = 1$ on the left part:



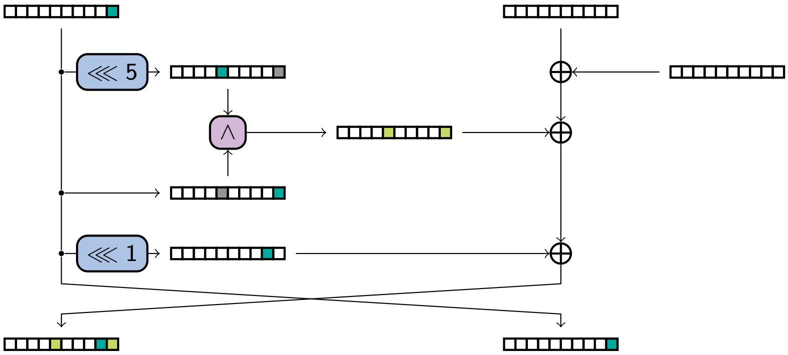
Probability of transition through f

Consider a difference $\alpha = 1$ on the left part:



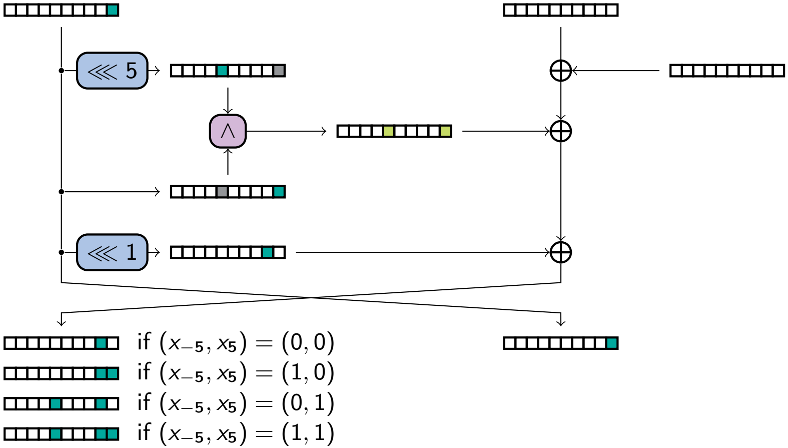
Probability of transition through f

Consider a difference $\alpha = 1$ on the left part:



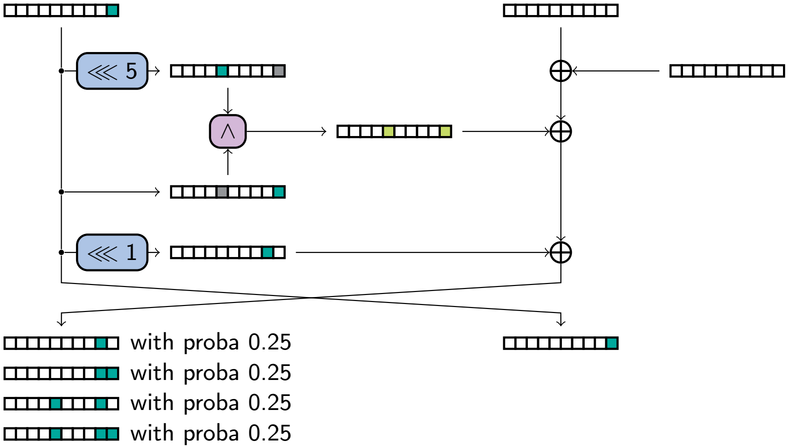
Probability of transition through f

Consider a difference $\alpha = 1$ on the left part:



Probability of transition through f

Consider a difference $\alpha = 1$ on the left part:



Probability of transition through f

Since f is **quadratic**, the **exact probability of transitions** can be computed efficiently for **Simon** and **Simeck**: [KLT, CRYPTO'15]

$$\Pr[(\delta_L, \delta_R) \to (\delta'_L, \delta'_R)] = \begin{cases} 2^{-\dim(U_{\delta_L})} & \text{if } \delta_L = \delta'_R \text{ and } \delta_R \oplus \delta'_L \in U_{\delta_L} \\ 0 & \text{otherwise} \end{cases}$$

$$U_\delta = \mathrm{Img}\left(x \mapsto f(x) \oplus f(x \oplus \delta) \oplus f(\delta)\right) \oplus f(\delta)$$

Table of contents

A class of high probability trails

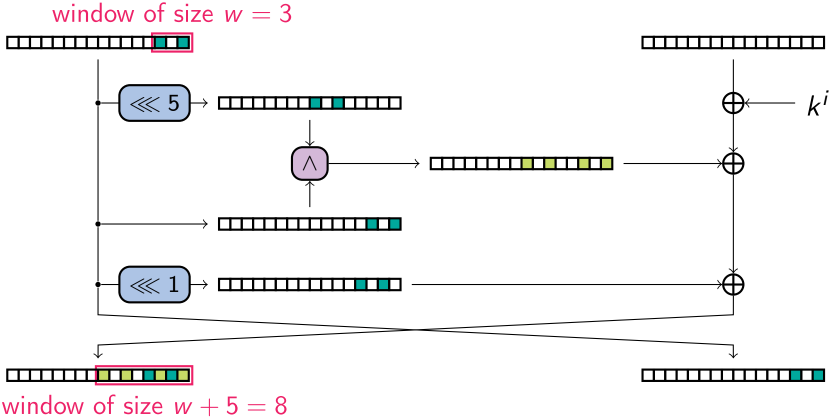
We know how to compute $\Pr[(\delta_L, \delta_R) \to (\delta'_L, \delta'_R)]$ **easily** now...

$\to$ But computing $\Pr[(\delta_L, \delta_R) \overset{r}{\rightsquigarrow} (\delta'_L, \delta'_R)]$ remains **hard**!

A class of high probability trails

We know how to compute $\Pr[(\delta_L, \delta_R) \to (\delta'_L, \delta'_R)]$ **easily** now...
$\to$ But computing $\Pr[(\delta_L, \delta_R) \overset{r}{\rightsquigarrow} (\delta'_L, \delta'_R)]$ remains **hard**!

Observation: Simeck diffusion in the **worst** case

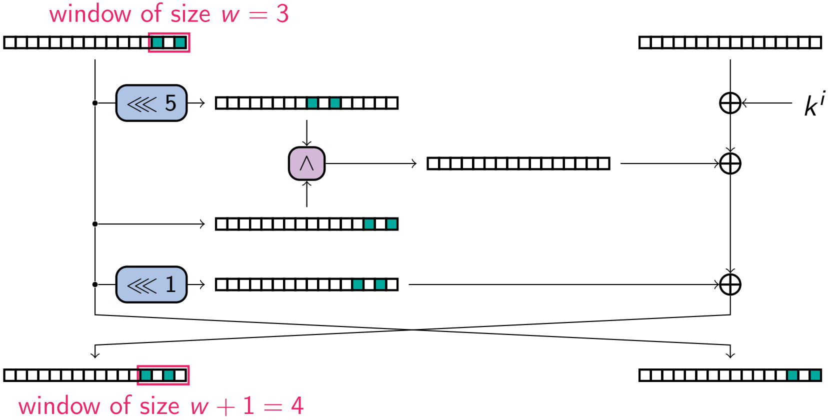


A class of high probability trails

We know how to compute $\Pr[(\delta_L, \delta_R) \to (\delta'_L, \delta'_R)]$ **easily** now...

$\to$ But computing $\Pr[(\delta_L, \delta_R) \overset{r}{\rightsquigarrow} (\delta'_L, \delta'_R)]$ remains **hard**!

Observation: Simeck diffusion in the **best** case

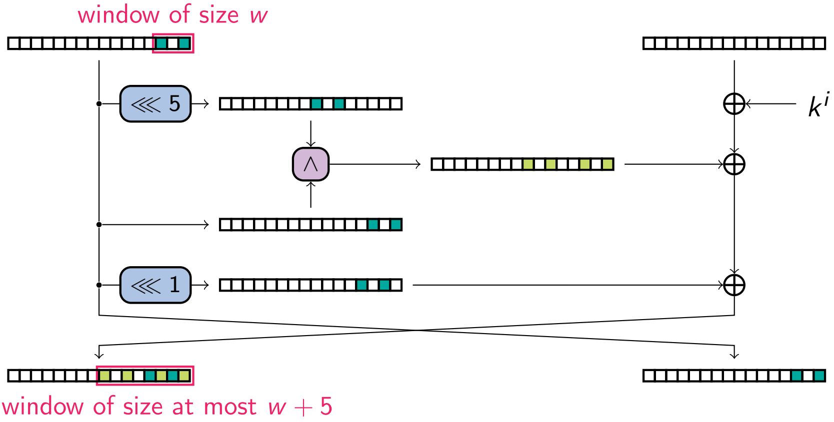


A class of high probability trails

We know how to compute $\Pr[(\delta_L, \delta_R) \to (\delta'_L, \delta'_R)]$ **easily** now...

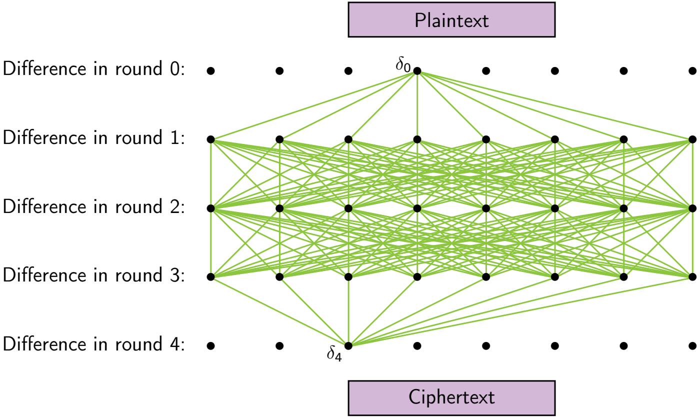
$\to$ But computing $\Pr[(\delta_L, \delta_R) \overset{r}{\rightsquigarrow} (\delta'_L, \delta'_R)]$ remains **hard**!

Conclusion: Simeck has a relatively slow diffusion!



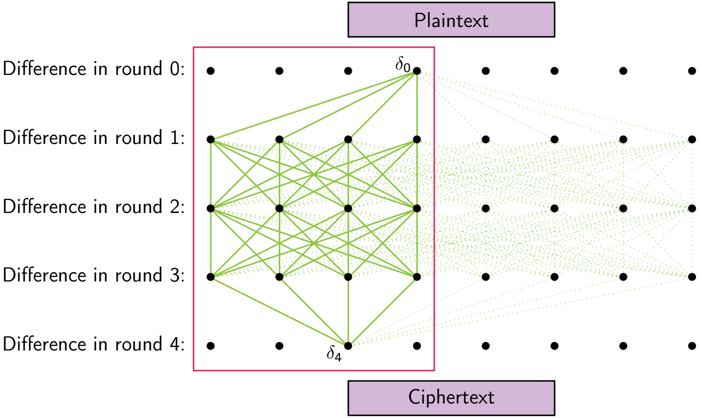
A class of high probability trails

Our idea is to focus on trails that are only active in a window of w bits:



A class of high probability trails

Our idea is to focus on trails that are only active in a window of w bits:



A class of high probability trails

- w: the size of the window ($w \leq n/2$).
- Δ_w: the vector space of differences active only in the w LSBs.
- Δ_w^2: the product $\Delta_w \times \Delta_w$ where the two words are considered.

A class of high probability trails

- w: the size of the window ($w \leq n/2$).
- Δ_w: the vector space of differences active only in the w LSBs.
- Δ_w^2: the product $\Delta_w \times \Delta_w$ where the two words are considered.

A lower bound of the probability of the differential (δ_0, δ_r) is computed by summing over all characteristics with intermediate differences in Δ_w^2:

$$\Pr[\delta_0 \underset{w}{\overset{r}{\rightsquigarrow}} \delta_r] = \sum_{\delta_1, \delta_2, \ldots \delta_{r-1} \in \Delta_w^2} \prod_{i=1}^{r} \Pr[\delta_{i-1} \to \delta_i] \leq \Pr[\delta_0 \overset{r}{\rightsquigarrow} \delta_r]$$

A class of high probability trails

- w: the size of the window ($w \leq n/2$).
- Δ_w: the vector space of differences active only in the w LSBs.
- Δ_w^2: the product $\Delta_w \times \Delta_w$ where the two words are considered.

A lower bound of the probability of the differential (δ_0, δ_r) is computed by summing over all characteristics with intermediate differences in Δ_w^2:

$$\Pr[\delta_0 \underset{w}{\overset{r}{\rightsquigarrow}} \delta_r] = \sum_{\delta_1, \delta_2, \ldots \delta_{r-1} \in \Delta_w^2} \prod_{i=1}^{r} \Pr[\delta_{i-1} \to \delta_i] \leq \Pr[\delta_0 \overset{r}{\rightsquigarrow} \delta_r]$$

For $w = 18$ and $r = 30$: a week on a 48-core machine using 1TB of RAM

Our results

$\rightarrow$ **Tighter lower bound** for existing differentials (with $w = 18$):

Rounds	Differential	Proba (previous)	Reference	Proba (new)
26	$(0,11) \rightarrow (22,1)$	$2^{-60.02}$	[Kölbl, Roy, 16]	$2^{-54.16}$
26	$(0,11) \rightarrow (2,1)$	$2^{-60.09}$	[Qin, Chen, Wang, 16]	$2^{-54.16}$
27	$(0,11) \rightarrow (5,2)$	$2^{-61.49}$	[Liu, Li, Wang, 17]	$2^{-56.06}$
27	$(0,11) \rightarrow (5,2)$	$2^{-60.75}$	[Huang, Wang, Zhang, 18]	"
28	$(0,11) \rightarrow (\mathtt{A8},5)$	$2^{-63.91}$	[Huang, Wang, Zhang, 18]	$2^{-59.16}$

Our results

→ Tighter lower bound for existing differentials (with $w = 18$):

Rounds	Differential	Proba (previous)	Reference	Proba (new)
26	$(0, 11) \rightarrow (22, 1)$	$2^{-60.02}$	[Kölbl, Roy, 16]	$2^{-54.16}$
26	$(0, 11) \rightarrow (2, 1)$	$2^{-60.09}$	[Qin, Chen, Wang, 16]	$2^{-54.16}$
27	$(0, 11) \rightarrow (5, 2)$	$2^{-61.49}$	[Liu, Li, Wang, 17]	$2^{-56.06}$
27	$(0, 11) \rightarrow (5, 2)$	$2^{-60.75}$	[Huang, Wang, Zhang, 18]	"
28	$(0, 11) \rightarrow (\mathtt{A8}, 5)$	$2^{-63.91}$	[Huang, Wang, Zhang, 18]	$2^{-59.16}$

→ The best characteristics we identified are a set of 64 characteristics:

$$\{(1,2),(1,3),(1,22),(1,23),(2,5),(2,7),(2,45),(2,47)\}$$
$$\rightarrow$$
$$\{(2,1),(3,1),(22,1),(23,1),(5,2),(7,2),(45,2),(47,2)\}$$

⇒ However, $(0, 1) \rightarrow (1, 0)$ is almost as good and will lead to a more efficient key recovery because it has fewer active bits!

Differentials with high probabilities

log_2 of the probability of differentials for **Simeck** (using $w = 18$):

	Differential	
Rounds	$(0, 1) \rightarrow (1, 0)$	$(1, 2) \rightarrow (2, 1)$
10	$-\infty$	$-\infty$
11	-23.25	-27.25
12	-26.40	-26.17
13	-28.02	-26.90
14	-30.06	-29.59
15	-31.93	-31.37
$\vdots$	$\vdots$	$\vdots$
20	-41.75	-41.26
$\vdots$	$\vdots$	$\vdots$
25	-51.01	-50.54
$\vdots$	$\vdots$	$\vdots$
30	-60.41	-59.92
31	-62.29	-61.81
32	-64.17	-63.69

Differentials with high probabilities

How does our lower bound vary depending on the size of the window w?

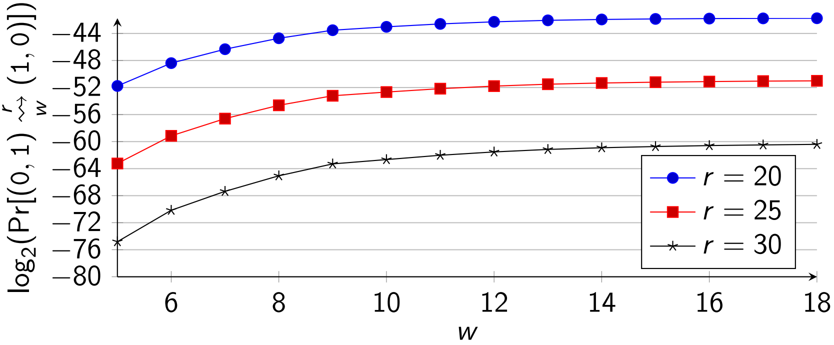


Table of contents

Stronger Linear distinguishers for Simon-like ciphers

By applying the same reasoning to linear cryptanalysis, a set of 64 (almost) **optimal trails** is obtained:

$$\{(20, 40), (22, 40), (60, 40), (62, 40), (50, 20), (51, 20), (70, 20), (71, 20)\}$$
$$\rightarrow$$
$$\{(40, 20), (40, 22), (40, 60), (40, 62), (20, 50), (20, 51), (20, 70), (20, 71)\}$$

Stronger Linear distinguishers for Simon-like ciphers

By applying the same reasoning to linear cryptanalysis, a set of 64 (almost) **optimal trails** is obtained:

$$\{(20,40),(22,40),(60,40),(62,40),(50,20),(51,20),(70,20),(71,20)\}$$
$$\rightarrow$$
$$\{(40,20),(40,22),(40,60),(40,62),(20,50),(20,51),(20,70),(20,71)\}$$

$\rightarrow$ They are bit-reversed versions of the optimal differential characteristics.

Stronger Linear distinguishers for Simon-like ciphers

By applying the same reasoning to linear cryptanalysis, a set of 64 (almost) **optimal trails** is obtained:

$$\{(20,40),(22,40),(60,40),(62,40),(50,20),(51,20),(70,20),(71,20)\}$$
$$\rightarrow$$
$$\{(40,20),(40,22),(40,60),(40,62),(20,50),(20,51),(20,70),(20,71)\}$$

$\rightarrow$ They are bit-reversed versions of the optimal differential characteristics.

$\rightarrow$ For key recovery attack, the preference is given to $(1,0) \rightarrow (0,1)$.

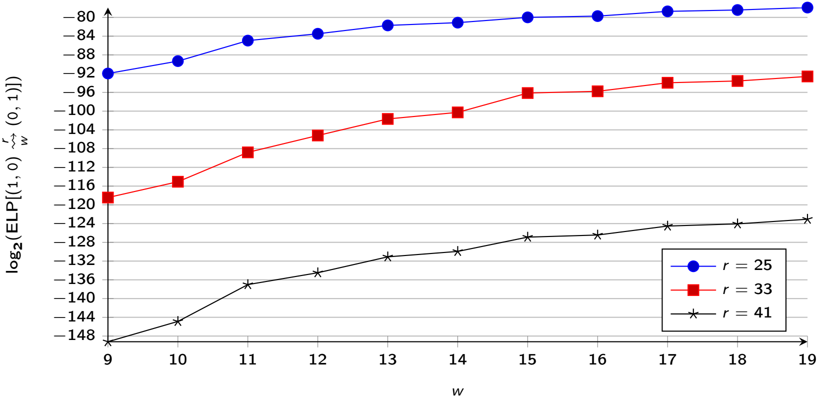
Lower bound of linear and differential distinguishers

Comparison of the **probability** of differentials and the linear potential of linear approximations for Simeck ($\log_2$, using $w = 18$). We also give the total number of trails included in the bound in parenthesis ($\log_2$):

	Differential			Linear		
Rounds	$(0,1) \to (1,0)$		$(1,2) \to (2,1)$	$(1,0) \to (0,1)$		$(1,2) \to (2,1)$
10	$-\infty$		$-\infty$	$-\infty$		$-\infty$
11	−23.25	(28.0)	−27.25	−23.81	(23.9)	−27.81
12	−26.40	(36.2)	−26.17	−26.39	(31.7)	−26.68
13	−28.02	(47.2)	−26.90	−27.98	(42.0)	−27.31
14	−30.06	(58.2)	−29.59	−29.95	(52.5)	−29.56
15	−31.93	(70.8)	−31.37	−31.86	(64.9)	−31.29
⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	−41.75	(131.9)	−41.26	−41.74	(124.5)	−41.25
⋮	⋮	⋮	⋮	⋮	⋮	⋮
25	−51.01	(192.9)	−50.54	−51.00	(184.1)	−50.56
⋮	⋮	⋮	⋮	⋮	⋮	⋮
30	−60.41	(254.0)	−59.92	−60.36	(243.6)	−59.86
31	−62.29	(266.2)	−61.81	−62.24	(255.5)	−61.75
32	−64.17	(278.4)	−63.69	−64.12	(267.4)	−63.63
33	−66.05	(290.6)	−65.57	−66.00	(279.3)	−65.51

What about Simon?

We also apply the same strategy against **Simon**, but the bound we obtain is **not as tight** as for Simeck: the linear potential still increases significantly with the window size w.



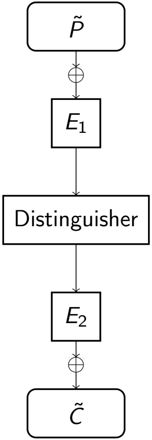
Effect of w on the probability of Simon linear hulls.

Table of contents

Key Recovery

Distinguisher

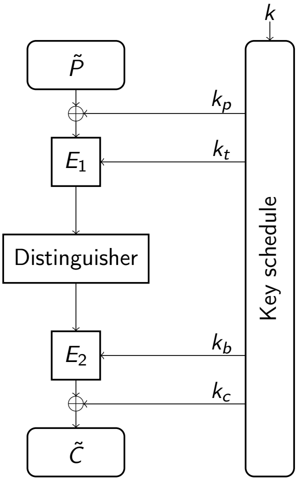
Key Recovery



General description of a cipher.

- Some rounds are added **before** and/or **after** the distinguisher.

Key Recovery



General description of a cipher.

- Some rounds are added **before** and/or **after** the distinguisher.
- The statistic used by the distinguisher is Q, and it can be evaluated using a subset of the key: (k_p, k_t, k_b, k_c).
- The total number of guessed bits is κ_g with $\kappa_g < \kappa$.

Key Recovery

Algorithm Naive key recovery

for all $k = (k_p, k_t, k_b, k_c)$ **do**
  for all pairs in D **do**
    compute $Q(k)$
  if $Q(k) > s$ **then**
    k is a possible candidate

Complexity: $D \times 2^{\kappa_g}$ with κ_g the number of key bits of k.

Key Recovery

Algorithm Naive key recovery

---

for all $k = (k_p, k_t, k_b, k_c)$ **do**
  for all pairs in D **do**
    compute $Q(k)$
  if $Q(k) > s$ **then**
    k is a possible candidate

Complexity: $D \times 2^{\kappa_g}$ with κ_g the number of key bits of k.

This can be reduced to approximately $D + 2^{\kappa_g}$ using algorithm tricks:

- Dynamic key guessing for Differential Cryptanalysis [QHS'16, WWJZ'18]
- Fast Walsh Transform for Linear Cryptanalysis [CSQ'07, FN'20]

Overview of the attack

(0) Find an efficient distinguisher Q

(1) Find the subset of the key that need to be guessed to evaluate Q

(2) Rearrange operations to reduce the time complexity
from $D \times 2^{\kappa_g}$ to approximately $D + 2^{\kappa_g}$

Overview of the attack

Example: distinguisher over 30 rounds – Simeck64/128

Differential cryptanalysis | Linear cryptanalysis

(0) Find an efficient distinguisher Q

$(0,1) \to (1,0)$ p $= 2^{-60.41}$ | $(1,0) \to (0,1)$ p $= 2^{-60.36}$

(1) Find the subset of the key that need to be guessed to evaluate Q

3+7 rounds added with $\kappa_g = 123$ | 4+8 rounds added with $\kappa_g = 118$

(2) Rearrange operations to reduce the time complexity from $D \times 2^{\kappa_g}$ to approximately $D + 2^{\kappa_g}$

Dynamic key guessing | Fast Walsh Transform

Overview of the attack

Example: distinguisher over 30 rounds – Simeck64/128

Differential cryptanalysis | Linear cryptanalysis

(0) Find an efficient distinguisher Q

Differential: $(0,1) \to (1,0)$ $\mathsf{p} = 2^{-60.41}$

Linear: $(1,0) \to (0,1)$ $\mathsf{p} = 2^{-60.36}$

(1) Find the subset of the key that need to be guessed to evaluate Q

Differential: 3+7 rounds added with $\kappa_g = 123$

Linear: 4+8 rounds added with $\kappa_g = 118$

$\Rightarrow$ **main difference between differential and linear cryptanalysis!**

(2) Rearrange operations to reduce the time complexity from $D \times 2^{\kappa_g}$ to approximately $D + 2^{\kappa_g}$

Differential: Dynamic key guessing

Linear: Fast Walsh Transform

Linear VS Differential Key Recovery

Key bits	Differential		Linear	
Rounds	total	independent	total	independent
1	0	0	0	0
2	2	2	2	2
3	9	9	7	7
4	27	27	16	16
5	56	56	30	30
6	88	88	50	48
7	120	114	75	68
8			104	88

Comparison of the **number of bits** that have to be **guessed** for differential and linear attacks against Simeck64/128.

Results on Simeck

Cipher	Rounds	Attacked	Data	Time	Ref	Note
Simeck48/96	36	30	$2^{47.66}$	$2^{88.04}$	[QCW'16]	Linear † ‡
		32	2^{47}	$2^{90.9}$	**New**	Linear
Simeck64/128	44	37	$2^{63.09}$	$2^{121.25}$	[QCW'16]	Linear † ‡
		42	$2^{63.5}$	$2^{123.9}$	**New**	Linear

Summary of previous and new attacks against Simeck.

† The advantage is too low to do a key recovery.

‡ Attack use the duality between linear and differential distinguishers.

Results on Simon

Cipher	Rounds	Attacked	Data	Time	Ref	Note
Simon96/96	52	37	2^{95}	$2^{87.2}$	[WWJZ'18]	Diff.
		43	2^{94}	$2^{89.6}$	**New**	Linear
Simon96/144	54	38	$2^{95.2}$	2^{136}	[CW'16]	Linear
		45	2^{95}	$2^{136.5}$	**New**	Linear
Simon128/128	68	50	2^{127}	$2^{119.2}$	[WWJZ'18]	Diff.
		53	2^{127}	2^{121}	**New**	Linear
Simon128/192	69	51	2^{127}	$2^{183.2}$	[WWJZ'18]	Diff.
		55	2^{127}	$2^{185.2}$	**New**	Linear
Simon128/256	72	53	$2^{127.6}$	2^{249}	[CW'16]	Linear
		56	2^{126}	2^{249}	**New**	Linear

Summary of previous and new attacks against Simon.

Table of contents

Conclusion

- Better probabilities for existing differential and linear distinguishers using trails with all intermediate states in a window of w bits.

Conclusion

- Better probabilities for existing differential and linear distinguishers using trails with **all intermediate states in a window of w bits**.
- New distinguishers with the **minimum number of active bits**, implying **cheaper** key-recovery.

Conclusion

- Better probabilities for existing differential and linear distinguishers using trails with **all intermediate states in a window of w bits**.
- New distinguishers with the **minimum number of active bits**, implying **cheaper** key-recovery.
- Attacks on **42 out of 44** rounds for Simeck64/128 and **43 out of 52** rounds of Simon96/96 using advanced existing techniques.

Conclusion

- Better probabilities for existing differential and linear distinguishers using trails with **all intermediate states in a window of w bits**.
- New distinguishers with the **minimum number of active bits**, implying **cheaper** key-recovery.
- Attacks on **42 out of 44** rounds for Simeck64/128 and **43 out of 52** rounds of Simon96/96 using advanced existing techniques.
- **Further work** can probably improve our results concerning **Simon**...

Conclusion

- Better probabilities for existing differential and linear distinguishers using trails with **all intermediate states in a window of w bits**.
- New distinguishers with the **minimum number of active bits**, implying **cheaper** key-recovery.
- Attacks on **42 out of 44** rounds for Simeck64/128 and **43 out of 52** rounds of Simon96/96 using advanced existing techniques.
- **Further work** can probably improve our results concerning **Simon**...

For more details:

`https://eprint.iacr.org/2021/1198`