Cryptonet – 2009-April-22 – Linear Cryptanalysis II

→ Recall that we've developed tools to determine the bias for subsets of S-box input and output bits.

→ Show the building of a linear approximation table for a $\pi_S : \mathbb{Z}_2^2 \to \mathbb{Z}_2^2$:

x_1	x_2	y_1	y_2
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	1

a \ b	0	1	2	3
0	4	2	2	2
1	2	2	4	2
2	2	0	2	2
3	2	2	2	0

→ Attacking an SPN

↳ First, lay out a specific $\ell = m = 4$ SPN taken from Stinson:

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(x)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

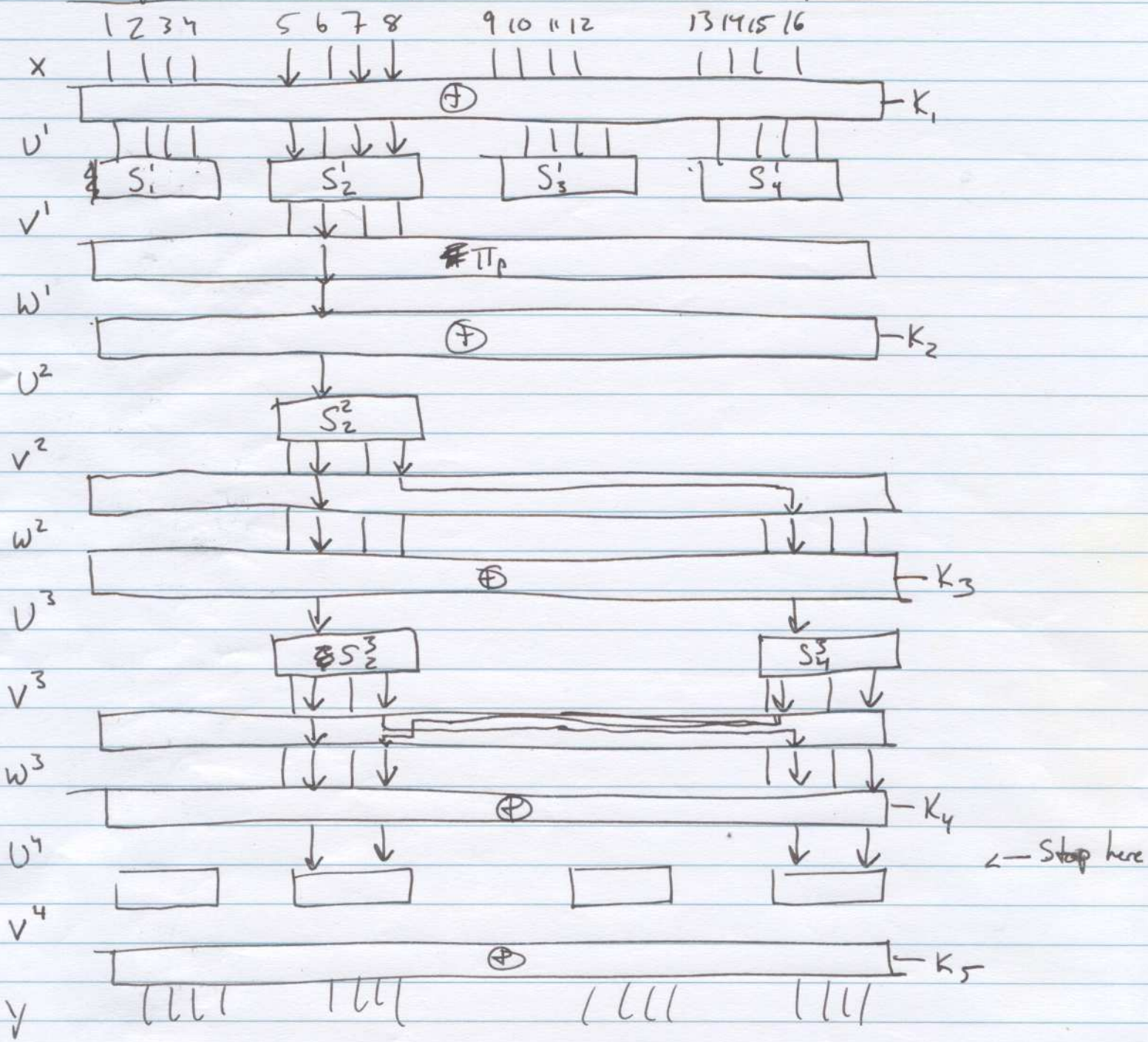
x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_P(x)$	1	5	9	D	2	6	A	E	3	7	B	F				

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_P(x)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16	

↳ Page 2 shows the graphical representation of the SPN, particularly the active S-boxes.

↳ How did we arrive at this set of active elements? On page 3, show the elements of the linear approximation table — we chose some of those for which $N_L(a,b)$ is largest (12) ⇒ the greatest magnitude bias.

Cryptonet — 2009-April-22 — Linear cryptanalysis II



↑ The <u>active</u> elements of our linear approximation for an SPN.

Cryptonet — 2009-April-22 — Linear cryptanalysis II

↳ The used elements of our linear approximation table:

a \ b	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4						4										
5																
6																
7																
8																
9																
A																
B					12											
C																
D																
E																
F																

↳ By choosing these two biased values, we can construct variables that are the XOR combination of S-box inputs and outputs at different parts of the network:

$$t_1 = u_5^1 \oplus u_7^1 \oplus u_8^1 \oplus v_6^1 \implies \epsilon = 1/4$$

$$t_2 = u_6^2 \oplus v_6^2 \oplus v_8^2 \implies \epsilon = -1/4$$

$$t_3 = u_6^3 \oplus v_6^3 \oplus v_8^3 \implies \epsilon = -1/4$$

$$t_4 = u_{14}^3 \oplus v_{14}^3 \oplus v_{16}^3 \implies \epsilon = -1/4$$

↳ Correlate these to the graphical SPN.

Cryptonet — 2009-April-20 — Linear cryptanalysis II

↳ Why these?
↳ Their biases ~~is~~ are large.
↳ Their combined XOR will allow us to cancel intermediate values, leaving only x, u^4, and K_i ~~bits~~ variables.

↳ ~~Let~~ Assume that the t values are independent. This assumption is false, but is sufficient as an approximation. Apply the piling-up lemma to $t_1 \oplus t_2 \oplus t_3 \oplus t_4$:

$$\epsilon_{t_1,t_2,t_3,t_4} = 2^3\left(\frac{1}{4}\right)\left(\frac{-1}{4}\right)^3 = \frac{-1}{32}$$ ← important that this value is not 0.

↳ As promised, simplify that xor to x, u^4, and K_i bits:

$$t_1 = u_5^1 \oplus u_7^1 \oplus u_8^1 \oplus v_6^1 = x_5 \oplus K_5^1 \oplus x_7 \oplus K_7^1 \oplus x_8 \oplus K_8^1 \oplus v_6^1$$

$$t_2 = u_6^2 \oplus v_6^2 \oplus v_8^2 = v_6^1 \oplus K_6^2 \oplus v_6^2 \oplus v_8^2$$

$$t_3 = u_6^3 \oplus v_6^3 \oplus v_8^3 = v_6^2 \oplus K_6^3 \oplus v_6^3 \oplus v_8^3$$

$$t_4 = u_{14}^3 \oplus v_{14}^3 \oplus v_{16}^3 = v_8^2 \oplus K_{14}^3 \oplus v_{14}^3 \oplus v_{16}^3$$

So, $t_1 \oplus t_2 \oplus t_3 \oplus t_4 = x_5 \oplus K_5^1 \oplus x_7 \oplus K_7^1 \oplus x_8 \oplus K_8^1 \oplus \cancel{v_6^1} \oplus$

$\cancel{v_6^1} \oplus K_6^2 \oplus \cancel{v_6^2} \oplus \cancel{v_8^2} \oplus$

since $a \oplus a = 0 \Rightarrow$

$\cancel{v_6^2} \oplus K_6^3 \oplus v_6^3 \oplus v_8^3 \oplus$

$\cancel{v_8^2} \oplus K_{14}^3 \oplus v_{14}^3 \oplus v_{16}^3$

$= x_5 \oplus x_7 \oplus x_8 \oplus$

$v_6^3 \oplus v_8^3 \oplus v_{14}^3 \oplus v_{16}^3 \oplus$

$K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3$

Cryptonet — 2009-April-22 — Linear ~~approxim~~ cryptanalysis II

↳ More simplification:

$$v_6^3 = u_6^4 \oplus k_6^4 \qquad v_{14}^3 = u_8^4 \oplus k_8^4$$

$$v_8^3 = u_{14}^4 \oplus k_{14}^4 \qquad v_{16}^3 = u_{16}^4 \oplus k_{16}^4$$

So: $\bigoplus_{i=1}^{4} t_i = x_5 \oplus x_7 \oplus x_8 \oplus$

$u_6^4 \oplus u_8^4 \oplus u_{14}^4 \oplus u_{16}^4 \oplus$

$k_5^1 \oplus k_7^1 \oplus k_8^1 \oplus k_6^2 \oplus k_6^3 \oplus k_{14}^3 \oplus k_6^4 \oplus k_8^4 \oplus k_{14}^4 \oplus k_{16}^4$

x, u^4, and k bits only!

↳ For a given key, the k bits, XOR-ed, $= 0$ or 1. So, the remainder of the expression, for a given key, is $\pm 1/32$.

↳ Our goal: Use ~~our~~ our plaintext/ciphertext pairs to determine 8 bits of k^5 — those XOR-d w/ the output of S_4^2 and S_4^4, ~~about~~ into which we have active inputs.

↳ There are ~~256~~ 2^8 possible values for these ~~eight~~ 8 bits of k^5, so we have 256 candidate subkeys.

↳ For each (x, y) of our plain-/cipher-text pairs, attempt each of the 256 candidate subkeys k_c.

↳ For each x, y, k_c, we can perform ~~de~~ a partial decryption from y to u^4.

↳ We can then compute: $x_5 \oplus x_7 \oplus x_8 \oplus u_6^4 \oplus u_8^4 \oplus u_{14}^4 \oplus u_{16}^4$

↳ ~~For~~ Keep a counter for each ~~cand~~ k_c. If the above $= 0$, increment the counter.

↳ At the end, most ~~keys~~ k_c counters $= T/2$, while the ~~correct~~ correct k_c counter $= T/2 \pm T/32$.