Cryptonet – 2009-April-28 – Linear cryptanalysis

→ Big picture:

↳ Known plaintext attack w/ many plain/ciphertext pairs from the same K.

↳ Recall the last step of encryption for SPN's:

$$y = v^n \oplus K_{n+1}$$

Thus, the first step of decryption is:

$$v^n = y \oplus K_{n+1}$$

We have y (many such y's). What if we try different K_{n+1} values, and then look for a probabilistic relationship between v^n and x? That is, we seek the K_{n+1} s.t. the most bits of v^n are not random (i.e., not $p = 1/2$).

→ Background probability theory

↳ Piling-up Lemma:

↳ Assume a sequence of bits $x_1, x_2, \ldots$, and a corresponding sequence of probabilities $p_1, p_2, \ldots$, s.t.: $Pr(x_i = 0) = p_i$ and $Pr(x_i = 1) = (1 - p_i)$

↳ If $i \neq j$, then if x_i is independent of x_j then:

$$Pr(x_i = 0, x_j = 0) = p_i p_j$$
$$Pr(x_i = 0, x_j = 1) = p_i (1 - p_j)$$
$$Pr(x_i = 1, x_j = 0) = (1 - p_i) p_j$$
$$Pr(x_i = 1, x_j = 1) = (1 - p_i)(1 - p_j)$$

Cryptonot — 2009-April-20 — Linear cryptanalysis

↳ Using the above, we can compute the probability of $x_i \oplus x_j$:

$$\Pr(x_i \oplus x_j = 0) = \Pr(x_i = 0, x_j = 0) + \Pr(x_i = 1, x_j = 1)$$
$$= p_i p_j + (1-p_i)(1-p_j)$$

$$\Pr(x_i \oplus x_j = 1) = \Pr(x_i = 0, x_j = 1) + \Pr(x_i = 1, x_j = 0)$$
$$p_i(1-p_j) + (1-p_i)p_j$$

↳ Let the <u>bias</u> of x_i be $\epsilon_i = p_i - 1/2$, $\Rightarrow$

$$-1/2 \le \epsilon_i \le 1/2$$

$$\Pr(x_i = 0) = 1/2 + \epsilon_i \quad \text{and} \quad \Pr(x_i = 1) = 1/2 - \epsilon_i$$

↳ ~~For a~~ Not proven but true:

↳ ~~The~~ Let the bias of $x_i \oplus x_j = \epsilon_{i,j} = 2\epsilon_i \epsilon_j$.

↳ More generally, the Piling-up Lemma:

The bias of $x_{i_1} \oplus x_{i_2} \oplus \dots \oplus x_{i_k} = \bigoplus_{j=1}^{k} x_{i_j}$ is

$$\epsilon_{i_1, i_2, \dots, i_k} = 2^{k-1} \prod_{j=1}^{k} \epsilon_{i_j}$$

↳ Corollary, also not proven:

$$\exists j : 1 \le j \le k : \epsilon_{i_j} = 0 \Rightarrow \epsilon_{i_0, i_1, \dots, i_k} = 0$$

↳ Foreshadowing: Recall that ~~we~~ this lemma is derived using the assumption of independence of the bits. We will seek cases where the <u>actual</u> bias does not match the Piling-up results.

Cryptonet — 2009 - April - 20 — Linear cryptanalysis

→ Linear approximations of S-boxes:

↳ Assume an S-box $\Pi_S : \mathbb{Z}_2^m \to \mathbb{Z}_2^n$ ~~where~~

↳ Also assume $x = (x_1, x_2, \ldots, x_m)$ where ~~each x_i~~
$\forall i : 1 \le i \le m : \epsilon_i = 0$ and each x_i is independent.

↳ $y = (y_1, y_2, \ldots, y_n)$, with no such assumptions about randomness or independence.

↳ Given the use of S-boxes for substitution:

~~$(y_1, \ldots, y_n) \neq \Pi_S(x_1,$~~

$$y \neq \Pi_S(x) \Longrightarrow Pr(x,y) = Pr(x_1, \ldots, x_m, y_1, \ldots, y_n) = 0$$

$$y = \Pi_S(x) \Longrightarrow Pr(x,y) = 2^{-m}.$$

↳ Why? $Pr(x) = 2^{-m}$ and $Pr(y|x) = 1$ via Bayes Theorem, so
~~$Pr(x,y)$~~

↳ We want, for $\left(\bigoplus_{\alpha=1}^{k} x_{i_\alpha}\right) \oplus \left(\bigoplus_{\beta=1}^{\ell} y_{j_\beta}\right)$, what is its bias?

↳ Why? If we can find a set of bits whose bias is furthest from 0, we may be able to mount an attack. We want a <u>weak</u> S-box.

↳ Bias:

$$\left(2^{k-1} \prod_{\alpha=1}^{k} \epsilon_{i_\alpha}\right)\left(2^{\ell-1} \prod_{\beta=1}^{\ell} \epsilon_{j_\beta}\right) = \epsilon_{i_1, \ldots, i_k, j_1, \ldots, j_\ell}$$

↳ ~~Recall that, for an SPN for a given ℓ and $\Pi_S : \mathbb{Z}_2^\ell \to \mathbb{Z}_2^\ell$,~~
~~the~~

Cryptonet — 2009-April-20 — Linear Cryptanalysis

↳ We can express any combination of x_i and y_j values with a set of $\mathbb{Z}_2$ vectors $a = (a_1, a_2, \ldots, a_m)$ and $b = (b_1, b_2, \ldots, b_n)$ s.t.:

$$\left(\bigoplus_{i=1}^{m} a_i x_i\right) \oplus \left(\bigoplus_{j=1}^{n} b_j y_j\right)$$

where a and b select x_i and y_j values.

↳ We can, for a given S-box, compute the probability (and thus the bias) for all possible m and n.

↳ For an SPN using a block size $l'm'$, $l' = m = n$, so there are $2^{2l'} = 2^{l'+1}$ possible a and b combinations.

↳ Let the <u>linear approximation</u> $N_L(a,b)$ denote the number of ~~8-tuple~~ mn-tuples of x and y elements s.t.:

$$(y_1, y_2, \ldots, y_n) = \pi_S(x_1, x_2, \ldots, x_m)$$

<u>and</u>

$$\left(\bigoplus_{i=1}^{m} a_i x_i\right) \oplus \left(\bigoplus_{j=1}^{n} b_j y_j\right) = 0$$

So, the bias for a given a and b is:

$$\epsilon(a,b) = \frac{N_L(a,b)}{2^m} - \frac{1}{2}$$

← # of entries in π_S, one per input

$$= \frac{N_L(a,b)}{2^m} - \frac{2^{m-1}}{2^m}$$

$$= \frac{N_L(a,b) - 2^{m-1}}{2^m}$$

→ Try an example? → Make an $2^m 2^n$ table of ~~each~~ $N_L(a,b)$ for each a, b. <u>A linear approx. table</u>

Cryptonet — 2009-April-20 — Linear cryptanalysis

For ~~an~~ our example last time, ~~an an SPN~~ $\ell = m = 4$, a graphical SPN: