Cryptonat - 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:

\[ y = x^n + K_{n+1} \]

Thus, the first step of decryption is:

\[ x^n = y \oplus K_{n+1} \]

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

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.:

\[ \Pr(x_i = 0) = \frac{p_i}{1 - p_i} \]

- If \( i \neq j \) if \( i \neq j \), then \( x_i \) is independent of \( x_j \) then:

\[
\begin{align*}
\Pr(x_i = 0, x_j = 0) &= p_i p_j \\
\Pr(x_i = 0, x_j = 1) &= p_i (1 - p_j) \\
\Pr(x_i = 1, x_j = 0) &= (1 - p_i) p_j \\
\Pr(x_i = 1, x_j = 1) &= (1 - p_i)(1 - p_j)
\end{align*}
\]
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 bias of $x_i$ be $\epsilon_i = p_i - \frac{1}{2}$, $\epsilon_i \leq \frac{1}{2}$

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

**Lemma (Not proven but true):**

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 the $x_i \oplus x_{i_2} \oplus \ldots \oplus x_{i_k}$ is

$$\epsilon = \epsilon_{i_1, i_2, \ldots, i_k} = 2^{k-1} \prod_{i=1}^{k} \epsilon_{i_j}$$

**Corollary (also not proven):**

If $1 \leq j \leq k; \epsilon_{i_j} = 0 \Rightarrow \epsilon_{i_0, i_1, \ldots, i_k} = 0$

**Foreshadowing:** Recall that this lemma is derived using the assumption of independence of the bits. We will seek cases where the actual bias does not match the piling-up results.
Cryptography - 2009 - April 20 - Linear cryptanalysis

→ Linear approximations of S-boxes:

→ Assume an S-box \( T_s : \mathbb{Z}_2^m \rightarrow \mathbb{Z}_2^n \) where

→ Also assume \( x = (x_1, x_2, \ldots, x_m) \) where each \( x_i \), \( 1 \leq i \leq m \), is \( E_i = 0 \) 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:

\[
\begin{align*}
(y_1, y_2, \ldots, y_n) & \neq T_s(x) \\
y \neq T_s(x) & \Rightarrow Pr(x, y) = Pr(x_1, y_1, x_2, y_2, \ldots, x_m, y_m) = 0 \\
y = T_s(x) & \Rightarrow Pr(x, y) = 2^{-m}.
\end{align*}
\]

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

\[ Pr(x, y) = 2^{-m} \]

→ We want \( \mathbb{E} [ (\bigoplus_{i=1}^k X_{i\alpha}) \oplus (\bigoplus_{\beta=1}^l Y_{j\beta}) ] \), what is its bias?

→ Why? If we can find a set of bits whose bias is very close to \( \phi \), we may be able to mount an attack. We want a weak S-box bias:

\[
\begin{align*}
(\sum_{\alpha=1}^{k-1} 2^{k-1-\alpha}) E_{i\alpha} & \bigoplus_{\beta=1}^l 2^{l-1-\beta} E_{j\beta} = E_i \bigoplus_{j,k} E_{i,j,k} \\
\end{align*}
\]

→ Recall that, for an SPN, \( \xi \) is a given \( \xi \) and \( T_s : \mathbb{Z}_2^m \rightarrow \mathbb{Z}_2^n \), then
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.:

\[
\left( \sum_{i=1}^{m} a_i x_i \right) + \left( \sum_{j=1}^{n} b_j y_j \right)
\]

where $a$ and $b$ select $x_i$ and $y_j$ values.

To find the bias $\varepsilon$ for a given $a$ and $b$ for all possible $m$ and $n$:

For an SPN using block size $\ell \times m$, $\ell' = m = n$, so there are $Z^{m-n} = 2^{m+1}$ possible $a$ and $b$ combinations.

Let the linear approximation $N_L(a, b)$ denote the number of $m,n$-tuples $st x$ and $y$ elements st.:

\[
(y_1, y_2, \ldots, y_n) = \Pi_{i=1}^{m} (x_1, x_2, \ldots, x_m)
\]

and

\[
\left( \sum_{i=1}^{m} a_i x_i \right) + \left( \sum_{j=1}^{n} b_j y_j \right) = 0
\]

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

\[
\varepsilon(a, b) = \frac{N_L(a, b)}{Z^m - 2^{m-1}} - \frac{1}{2^m}
\]

$Z^m$ - # of entries in $\Pi_i$, one per input

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

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

Try an example? Make a $2^m \times 2^n$ table of each $N_L(a, b)$ for each $a, b$. A linear approx.
For our example last time, an an SPN $l = m = 4$, a graphical SPN:

\[ X_{l \times m} \]

\[ X \]

\[ U' \]

\[ V' \]

\[ W' \]

\[ U^2 \]

\[ V^2 \]

\[ W^2 \]

\[ S_1' \]

\[ S_2' \]

\[ S_3' \]

\[ S_4' \]

\[ K_1 \]

\[ U^4 \]

\[ V^4 \]

\[ Y_{l \times m} \]