Cryptanalysis of Modern Symmetric-Key Block Ciphers

[Based on “A Tutorial on Linear and Differential Cryptanalysis” by Howard Heys.]

Modern block ciphers (like DES and AES):
- proceed in rounds
- each round has its own round key or subkey
- the subkeys are computed from the master key by the key schedule

A simpler modern-type block cipher for now: the substitution-permutation network (similar to DES and AES but simplified structure)
Substitution-Permutation Networks (SPN)

- consists of a number of rounds, each round (except the last), consists of XOR-ing the subkey (this is sometimes called key mixing), substitutions, and a permutation - typically subkeys are derived from the master key but here they are randomly generated and unrelated.

Let $\ell$ and $m$ be positive integers. The block length of the cipher is $\ell m$.

We will use one substitution (also called an S-box)

\[ \pi_S: \{0,1\}^\ell \rightarrow \{0,1\}^\ell \]

\[ \pi_S: (0,0) \rightarrow (1,0) \\
(0,1) \rightarrow (0,0) \\
(1,0) \rightarrow (1,1) \\
(1,1) \rightarrow (0,1) \]

and one permutation

\[ \pi_P: \{1,..., \ell m\} \rightarrow \{1,..., \ell m\}. \]
Substitution-Permutation Networks (SPN)

In each round:
- XOR with the round key,
- split the current string into m strings of length ℓ, apply $\pi_S$ to each of these m strings
- if this is not the last round, perform permutation $\pi_P$; if it is the last round, XOR with the round key $K_{R+1}$ where R is the number of rounds

For example, if $\ell=2$, $m=3$, $\pi_S$ and $\pi_P$ (see below), suppose the string before the round is 100011 and the round key is 100100 - what is the resulting string after this round?

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_S(x)$</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>x</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_P(x)$</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>
More on SPNs

- simple and very efficient, both in hardware and in software (assuming the S-boxes are not too large)
- decryption analogous to encryption (reverse each operation)
- very successful: DES and AES are variations on SPNs
- the first and last operations are XORing with subkeys (called whitening) - makes attacks harder

Figure 1 (Heys' tutorial): an example SPN that we will cryptanalyze
Attacks on SPNs

- linear cryptanalysis and differential cryptanalysis
- both: known-plaintext, and they require a lot of plaintext-ciphertext pairs

Linear cryptanalysis:
Find a linear relationship between a subset of the plaintext bits and a subset of the ciphertext bits; this relationship should hold with probability bounded away from $\frac{1}{2}$ (the further away from $\frac{1}{2}$, the better). This probability, minus $\frac{1}{2}$, is called the probability bias.

Note:
In SPNs, all computations are linear, except for the S-boxes. Also, recall that linear cryptosystems are vulnerable to known-plaintext attacks.
Linear Approximations of S-boxes

The S-box from Figure 1:

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>4</td>
<td>D</td>
<td>1</td>
<td>2</td>
<td>F</td>
<td>B</td>
<td>8</td>
<td>3</td>
<td>A</td>
<td>6</td>
<td>C</td>
<td>5</td>
<td>9</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>

Understanding the table: $\ell=4$, the possible 4-bit strings are given in HEX.

Let $X_1, X_2, X_3, X_4$ be random variables for the input bits (independent, uniform), and let $Y_1, Y_2, Y_3, Y_4$ be random variables for the output bits.
### Linear Approximations of S-boxes

**The S-box from Figure 1:**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>4</td>
<td>D</td>
<td>1</td>
<td>2</td>
<td>F</td>
<td>B</td>
<td>8</td>
<td>3</td>
<td>A</td>
<td>6</td>
<td>C</td>
<td>5</td>
<td>9</td>
<td>0</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Consider the linear equation:

\[ X_2 \oplus X_3 \oplus Y_1 \oplus Y_3 \oplus Y_4 = 0, \text{ or, equivalently } X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4. \tag{*} \]

This equation holds for 12 or the 16 possible input values \(X_1, X_2, X_3, X_4\). What is the probability bias of this equation?

The equation holds w. prob. \( \frac{12}{16} \) → the bias is \( \frac{12}{16} - \frac{1}{2} = \frac{1}{4} \).
Linear Approximations of S-boxes

The S-box from Figure 1:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td>4</td>
<td>D</td>
<td>1</td>
<td>2</td>
<td>F</td>
<td>B</td>
<td>8</td>
<td>3</td>
<td>A</td>
<td>6</td>
<td>C</td>
<td>5</td>
<td>9</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>

Consider the linear equation:

\[ X_1 \oplus X_4 = Y_2 \]

What is the probability bias of this equation? 0

(do yourself)
Linear Approximations of S-boxes

The S-box from Figure 1:

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>4</td>
<td>D</td>
<td>1</td>
<td>2</td>
<td>F</td>
<td>B</td>
<td>8</td>
<td>3</td>
<td>A</td>
<td>6</td>
<td>C</td>
<td>5</td>
<td>9</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>

Consider the linear equation:

\[ X_3 \oplus X_4 = Y_1 \oplus Y_4 \]

What is the probability bias of this equation?

\[ -\frac{3}{8} \]

Negatively.
Linear Approximations of S-boxes

The S-box from Figure 1:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>4</td>
<td>D</td>
<td>1</td>
<td>2</td>
<td>F</td>
<td>B</td>
<td>8</td>
<td>3</td>
<td>A</td>
<td>6</td>
<td>C</td>
<td>5</td>
<td>9</td>
<td>0</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

We can compute the probability biases for all linear equations relating the $X_i$'s and the $Y_i$'s. I.e. for any $a_i, b_i \in \{0, 1\}$, we can compute the bias of the equation

$$a_1X_1 \oplus a_2X_2 \oplus a_3X_3 \oplus a_4X_4 = b_1Y_1 \oplus b_2Y_2 \oplus b_3Y_3 \oplus b_4Y_4.$$ 

See Tables 3 and 4 in Heys's tutorial.

Next task: combining the linear approximations of the S-boxes to get a linear approximation of the entire SPN.
Piling-up Lemma

We will combine S-box approximations... What happens to the biases?

Piling-up Lemma: For $k$ independent random variables $X_1, X_2, ..., X_k$ where $X_i = 0$ has bias $\epsilon_i$, the equation $X_1 \oplus ... \oplus X_k = 0$ has bias $2^{k-1} \prod_{i=1}^{k} \epsilon_i$.

Note: lemma by Matsui, inventor of linear cryptanalysis

Proving the lemma for $k=2$:

\[
\begin{align*}
\Pr(X_1 = 0) &= p_1, \\
\Pr(X_2 = 1) &= p_2,
\end{align*}
\]

\[
\begin{align*}
\epsilon_1 &= p_i - \frac{1}{2}, \\
\epsilon_2 &= p_{i+1} - \frac{1}{2}.
\end{align*}
\]

\[
\begin{align*}
\Pr(X_1 \oplus X_2 = 0) &= \Pr(X_1 = 0 \text{ and } X_2 = 0) + \Pr(X_1 = 1 \text{ and } X_2 = 1) \\
&= \Pr(X_1 = 0) \cdot \Pr(X_2 = 0) + \Pr(X_1 = 1) \cdot \Pr(X_2 = 1) \\
&= (1-p_i)(1-p_{i+1}) + p_ip_{i+1} = \left(\frac{1}{2} - \epsilon_1\right) \left(\frac{1}{2} - \epsilon_2\right) + \left(\epsilon_i + \frac{1}{2}\right) \left(\epsilon_{i+1} + \frac{1}{2}\right) \\
&= \frac{1}{4} - \frac{\epsilon_1}{2} - \frac{\epsilon_2}{2} + \epsilon_1 \epsilon_2 + \frac{1}{4} + \frac{\epsilon_i}{2} + \frac{\epsilon_{i+1}}{2} + \epsilon_i \epsilon_{i+1} = \frac{1}{2} + 2\epsilon_1 \epsilon_2
\end{align*}
\]
We will combine S-box approximations... What happens to the biases?

Piling-up Lemma:
For \( k \) independent random variables \( X_1, X_2, ..., X_k \) where \( X_i = 0 \) has bias \( \epsilon_i \), the equation \( X_1 \oplus ... \oplus X_k = 0 \) has bias \( 2^{k-1} \prod_{i=1}^{k} \epsilon_i \).

Note: lemma by Matsui, inventor of linear cryptanalysis

Give a simple example that shows that the assumption that the \( X_i \)'s are independent is necessary.

\[
X_1 = X_2 \quad X_1 \oplus X_2 = 0 \quad \text{always true} \quad \Pr(X_1 \oplus X_2 = 0) = 1
\]

bias = \( \frac{1}{2} \)

not 2 \( \cdot \) 0 \( \cdot \) 0
Linear Approximation for the Cipher

Recall the SPN from Figure 1 (also see Figure 3; we do not do the last round on this slide).

Our approximation will involve S-boxes $S_{12}$, $S_{22}$, $S_{32}$, and $S_{34}$. We call them the active S-boxes.

We will use the following approximations of these S-boxes:

- $S_{12}$: $X_1 \oplus X_3 \oplus X_4 = Y_2$ bias $\frac{1}{4}$
- $S_{22}$: $X_2 = Y_2 \oplus Y_4$ bias $-\frac{1}{4}$
- $S_{32}$: $X_2 = Y_2 \oplus Y_4$ bias $-\frac{1}{4}$
- $S_{34}$: $X_2 = Y_2 \oplus Y_4$ bias $-\frac{1}{4}$
Linear Approximation for the Cipher

Let $P_i$ be the random variable for the $i$-th plaintext bit, let $U_{r,i}$ be the random variable for the $i$-th input bit to the round $r$ S-boxes, let $V_{r,i}$ be the random variable for the $i$-th output bit of the round $r$ S-boxes, and let $K_{r,i}$ be the $i$-th bit of the $r$-th subkey.

Let $T_1, T_2, T_3, T_4$ be random variables such that

\[ T_1 = U_{1,5} \oplus U_{1,7} \oplus U_{1,8} \oplus V_{1,6} \]
\[ T_2 = U_{2,6} \oplus V_{2,6} \oplus V_{2,8} \]
\[ T_3 = U_{3,6} \oplus V_{3,6} \oplus V_{3,8} \]
\[ T_4 = U_{3,14} \oplus V_{3,14} \oplus V_{3,16} \]

What are the biases of $T_i=0$ for $i \in \{1,2,3,4\}$?
Linear Approximation for the Cipher

Let $P_i$ be the random variable for the $i$-th plaintext bit, let $U_{r,i}$ be the random variable for the $i$-th input bit to the round $r$ S-boxes, let $V_{r,i}$ be the random variable for the $i$-th output bit of the round $r$ S-boxes, and let $K_{r,i}$ be the $i$-th bit of the $r$-th subkey.

Let $T_1, T_2, T_3, T_4$ be random variables such that

\[
\begin{align*}
T_1 &= U_{1,5} \oplus U_{1,7} \oplus U_{1,8} \oplus V_{1,6} \\
T_2 &= U_{2,6} \oplus V_{2,6} \oplus V_{2,8} \\
T_3 &= U_{3,6} \oplus V_{3,6} \oplus V_{3,8} \\
T_4 &= U_{3,14} \oplus V_{3,14} \oplus V_{3,16}
\end{align*}
\]

Note: the $T_i$'s are not independent but pretending that they are independent works well in practice.
Linear Approximation for the Cipher

Let $P_i$ be the random variable for the $i$-th plaintext bit, let $U_{r,i}$ be the random variable for the $i$-th input bit to the round $r$ S-boxes, let $V_{r,i}$ be the random variable for the $i$-th output bit of the round $r$ S-boxes, and let $K_{r,i}$ be the $i$-th bit of the $r$-th subkey.

Let $T_1, T_2, T_3, T_4$ be random variables such that

\[
T_1 = U_{1,5} \oplus U_{1,7} \oplus U_{1,8} \oplus V_{1,6}
\]
\[
T_2 = U_{2,6} \oplus V_{2,6} \oplus V_{2,8}
\]
\[
T_3 = U_{3,6} \oplus V_{3,6} \oplus V_{3,8}
\]
\[
T_4 = U_{3,14} \oplus V_{3,14} \oplus V_{3,16}
\]

Applying the Piling-up Lemma: what is the bias of $T_1 \oplus T_2 \oplus T_3 \oplus T_4 = 0$?

\[
2^{\frac{h-1}{2}} \cdot \frac{1}{2^h} \cdot \left(\frac{1}{2^3}\right)^3 = \frac{1}{32} \cdot 8 = 0.03125
\]
Linear Approximation for the Cipher

Expressing $T_1 \oplus T_2 \oplus T_3 \oplus T_4$ as the XOR of plaintext bits, subkey bits, and bits of the input (straightforward but tedious):

$$T_1 \oplus T_2 \oplus T_3 \oplus T_4 = P_5 \oplus P_7 \oplus P_8 \oplus U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14} \oplus K_{4,16}$$

For fixed key bits, their XOR-sum is either 0 or 1. Then the bias of

$$P_5 \oplus P_7 \oplus P_8 \oplus U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} = 0$$

is either $-\frac{1}{32}$ or $\frac{1}{32}$.\[\text{\small if the XOR of the key bits is 0, then this bias is } \frac{1}{32}]
Extracting Key Bits

Recall: we are performing a known-plaintext attack, and we assume that we have a large pool of plaintext-ciphertext pairs (all encrypted with the same key).

How to use our linear approximation to determine a part of subkey $K_5$?

We will partially decrypt each ciphertext, and see if our linear approximation

$$P_5 \oplus P_7 \oplus P_8 \oplus U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} = 0$$

holds or not.
In particular, we will go through all possible $2^8$ possibilities for the subkey bits $K_{5,5}, K_{5,6}, K_{5,7}, K_{5,8}, K_{5,13}, K_{5,14}, K_{5,15}, K_{5,16}$.

For each candidate subkey, compute the bias of

$$P_5 \oplus P_7 \oplus P_8 \oplus U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} = 0$$

(described on the next slide).

We are looking for a subkey for which the bias is the closest to $1/32$ or $-1/32$. 
Extracting Key Bits

How to compute the bias for a specific candidate subkey? For each plaintext-ciphertext pair, partially decrypt the ciphertext (in our case, XOR with the candidate subkey, then invert the two S-boxes to get $U_{4,5}, U_{4,6}, U_{4,7}, U_{4,8}, U_{4,13}, U_{4,14}, U_{4,15}, U_{4,16}$), then compute the value of

$$P_5 \oplus P_7 \oplus P_8 \oplus U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} = 0 \; (\star)$$

Determine the fraction of plaintext-ciphertext pairs for which this value is 0, subtract $\frac{1}{2}$ to get the bias (see Table 5). 

```
best bias = undefined.
for all 2^8 subkey possib. for K_5,5 ... K_5,16 
  count = 0
for all plaintext/ciphertext pairs (suppose there are b such pairs)
  XOR the key with the ciphertext to get V_{4,5} ... V_{4,16}
  Run the S-boxes backward to get U_{4,5} ... U_{4,16}
  Check if the equation (\star) holds \to if yes, count++
bias = count / b - 1/2
```

Choose the subkey for which $|\text{bias}|$ is the closest to $\frac{1}{32}$. 
Extracting Key Bits

How many plaintext-ciphertext pairs do we need? If the bias is $\epsilon$ (for us $|\epsilon| = 1/32$), we need about $c\epsilon^{-2}$ pairs for some “small” constant $c$. For our example $c=8$ is sufficient.

How many pairs do we need for our example?

$$c \cdot \frac{1}{\epsilon^2} = c \cdot (32)^2 = 8 \cdot 32^2$$

Questions:

- What are some disadvantages of linear cryptanalysis?

- How can you make your SPN more secure against linear cryptanalysis?