Product Systems, Substitution-Permutation Networks, and Linear and Differential Analysis
Cryptology, lecture 3

Nils Andersen

Stinson, Section 2.7–3.4

Tuesday, February 12th, 2008
1 Composition
   - Product

2 Iterated Ciphers
   - Iteration
   - Substitution-Permutation Networks
   - Feistel Networks
   - Exclusive Or

3 Linear Cryptanalysis
   - The Bias of a Random Variable
   - Linear Approximation
   - Linear Attack

4 Differential Cryptanalysis
   - Differential Approximation
   - Differential Attack
1 Composition
   - Product

2 Iterated Ciphers
   - Iteration
   - Substitution-Permutation Networks
   - Feistel Networks
   - Exclusive Or

3 Linear Cryptanalysis
   - The Bias of a Random Variable
   - Linear Approximation
   - Linear Attack

4 Differential Cryptanalysis
   - Differential Approximation
   - Differential Attack
1. Composition
   - Product

2. Iterated Ciphers
   - Iteration
   - Substitution-Permutation Networks
   - Feistel Networks
   - Exclusive Or

3. Linear Cryptanalysis
   - The Bias of a Random Variable
   - Linear Approximation
   - Linear Attack

4. Differential Cryptanalysis
   - Differential Approximation
   - Differential Attack
1. Composition
   - Product

2. Iterated Ciphers
   - Iteration
   - Substitution-Permutation Networks
   - Feistel Networks
   - Exclusive Or

3. Linear Cryptanalysis
   - The Bias of a Random Variable
   - Linear Approximation
   - Linear Attack

4. Differential Cryptanalysis
   - Differential Approximation
   - Differential Attack
Given

- \( S_1 = (P_1, M, K_1, E_1, D_1, Pr_1) \)
- \( S_2 = (M, C_2, K_2, E_2, D_2, Pr_2) \)

(Note: \( C_1 = P_2 = M \))

We write \( e(x, K) \) also as \( e_K(x) \).

Define

- \( S_1 \times S_2 = (P_1, C_2, K_1 \times K_2, E, D, Pr) \)
- \( e_{(K_1, K_2)} = e_{K_1} \circ e_{K_2} = e_{K_2} \circ e_{K_1} \)
- \( d_{(K_1, K_2)} = d_{K_2} \circ d_{K_1} = d_{K_1} \circ d_{K_2} \)
- \( Pr[(K_1, K_2)] = Pr_1(K_1) \cdot Pr_2(K_2) \)

Amounting to “serial composition”

\[
\begin{array}{c}
\xrightarrow{x} \quad S_1 \quad \xrightarrow{e_{K_1}(x)} \quad S_2 \quad \xrightarrow{e_{K_2}(e_{K_1}(x))}
\end{array}
\]
Given

- \( S_1 = (P_1, M, K_1, E_1, D_1, Pr_1) \)
- \( S_2 = (M, C_2, K_2, E_2, D_2, Pr_2) \)

(Note: \( C_1 = P_2 = M \))

We write \( e(x, K) \) also as \( e_K(x) \).

Define

- \( S_1 \times S_2 = (P_1, C_2, K_1 \times K_2, E, D, Pr) \)
- \( e_{(K_1,K_2)} = e_{K_2} \circ e_{K_1} \)
- \( d_{(K_1,K_2)} = d_{K_2} \circ d_{K_1} \)
- \( Pr[(K_1, K_2)] = Pr_1(K_1) \cdot Pr_2(K_2) \)

Amounting to “serial composition”

\[
\begin{align*}
\text{Given} & \quad \text{Define} \\
S_1 & \quad S_1 \times S_2 = \\
\text{e}_{K_1}(x) & \quad (P_1, C_2, K_1 \times K_2, E, D, Pr) \\
\text{Amounting to “serial composition”} & \quad \text{e}_{K_2}(\text{e}_{K_1}(x))
\end{align*}
\]
Properties

Multiplication of ciphers
- is associative, \( S_1 \times (S_2 \times S_3) = (S_1 \times S_2) \times S_3 \)
- is not (necessarily) commutative, \( S_2 \times S_1 \not\approx S_1 \times S_2 \)
- is endomorphic, if \( P = C \)
- (assume \( S \) endomorphic) is idempotent, if \( S \times S = S^2 = S \)

Examples

The **Shift**, **Substitution**, **Multiplicative**, **Affine**, **Hill**, **Vigenère** (*cf.* Exercise 2.20) and **Permutation Ciphers** are all idempotent.
Iteration

For a non-idempotent cipher \( S \) we might hope for the iterated cipher \( S \times \ldots \times S = S^n \) to be safer than just \( S \) by itself.

- If \( S \) and \( T \) are both idempotent and commute, then \( S \times T \) is also idempotent.

**Proof.**

\[
(S \times T) \times (S \times T) = S \times (T \times S) \times T = S \times (S \times T) \times T = (S \times S) \times (T \times T) = S \times T
\]
Iteration

For a non-idempotent cipher $S$ we might hope for the iterated cipher $S \times \ldots \times S = S^n$ to be safer than just $S$ by itself.

- If $S$ and $T$ are both idempotent and commute, then $S \times T$ is also idempotent.

Proof.

$$(S \times T) \times (S \times T) = S \times (T \times S) \times T = S \times (S \times T) \times T =$$

$$(S \times S) \times (T \times T) = S \times T$$

- Can you think of a non-idempotent cipher?
Iteration

For a non-idempotent cipher $S$ we might hope for the iterated cipher $S \times \ldots \times S = S^n$ to be safer than just $S$ by itself.

- If $S$ and $T$ are both idempotent and commute, then $S \times T$ is also idempotent.

Proof.

\[(S \times T) \times (S \times T) = S \times (T \times S) \times T = S \times (S \times T) \times T = (S \times S) \times (T \times T) = S \times T\]

Can you think of a non-idempotent cipher?
Iteration

For a non-idempotent cipher $S$ we might hope for the iterated cipher $S \times \ldots \times S = S^n$ to be safer than just $S$ by itself.

- If $S$ and $T$ are both idempotent and commute, then $S \times T$ is also idempotent.

Proof.

\[(S \times T) \times (S \times T) = S \times (T \times S) \times T = S \times (S \times T) \times T = (S \times S) \times (T \times T) = S \times T\]

- Can you think of a non-idempotent cipher?
Setting up an Iterated Cipher

- \( \mathcal{P} = \mathcal{C} \) are the states.
- The number of rounds \( \text{Nr} \in \mathbb{N} \).
- A key schedule \( K \mapsto (K^1, \ldots, K^{\text{Nr}}) \) assigning subkeys \( K^i \) (or round keys) to a given key \( K \).
  (Note: superscript-indexing.)
- A round function \( g : \mathcal{P} \times \mathcal{K} \to \mathcal{P} \) possessing an inverse \( g^{-1} \),
  i.e. \( g^{-1}(g(w, k), k) = w \) for each state \( w \) and subkey \( k \).

Encryption

\[
\begin{align*}
  w^0 & \leftarrow x \\
  w^i & \leftarrow g(w^{i-1}, K^i) \\
  y & \leftarrow w^{\text{Nr}}
\end{align*}
\]

\( i = 1, \ldots, \text{Nr} \)

Decryption

\[
\begin{align*}
  w^{\text{Nr}} & \leftarrow y \\
  w^{i-1} & \leftarrow g^{-1}(w^i, K^i) \\
  x & \leftarrow w^0
\end{align*}
\]

\( i = \text{Nr}, \ldots, 1 \)
Setting up a Substitution-Permutation Network

- For $\ell, m \in \mathbb{N}_1$, an $\ell m$-bit word $x = (x_1, \ldots, x_{m\ell})$ is split into $m \ell$-bit blocks (note the use of base 1 numbering):
  \[
x = x_{\langle 1 \rangle} \parallel \cdots \parallel x_{\langle m \rangle}
  \]
  \[
x_{\langle i \rangle} = (x_{(i-1)\ell+1}, \ldots, x_{i\ell})
  \]

- An $\ell$-bit block is permuted by (“substitution”)
  \[
  \pi_S : \{0, 1\}^\ell \to \{0, 1\}^\ell
  \]

- A complete $\ell m$-bit word is permuted by
  \[
  \pi_P : \{1, \ldots, \ell m\} \to \{1, \ldots, \ell m\}
  \]

- The key schedule maps $\mathcal{K}$ into $\left(\{0, 1\}^{\ell m}\right)^{N_{r+1}}$
Employing an SPN

The first $N_r - 1$ rounds consist of a subkey addition, $m$ substitutions (in parallel) and a permutation. The $N_r$-th round has subkey addition and substitution only, and finally $K^{N_r+1}$ is added.

Encryption

\[
\begin{align*}
w^0 & \leftarrow x \\
\text{for} & \ r \ \text{to} \ N_r - 1 \ \text{do} \\
& u^r \leftarrow w^{r-1} \oplus K^r \\
& \text{for} \ i \ \text{to} \ m \ \text{do} \ v^r_{\langle i \rangle} \leftarrow \pi_S(u^r_{\langle i \rangle}) \od \wedge \\
& w^r \leftarrow (v^r_{\pi_P(1)}, \ldots, v^r_{\pi_P(\ell m)}) \\
\text{od} \\
& u^{N_r} \leftarrow w^{N_r-1} \oplus K^{N_r} \\
& \text{for} \ i \ \text{to} \ m \ \text{do} \ v^{N_r}_{\langle i \rangle} \leftarrow \pi_S(u^{N_r}_{\langle i \rangle}) \od \\
y \leftarrow v^{N_r} \oplus K^{N_r+1}
\end{align*}
\]

Decryption

Solve Exercise 3.1.
### Example 3.1

All of the \( S_j^i \) use \( \pi_S \):

<table>
<thead>
<tr>
<th>( z )</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>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi_S(z) )</td>
<td>( E )</td>
<td>4</td>
<td>( D )</td>
<td>1</td>
<td>2</td>
<td>( F )</td>
<td>( B )</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( z )</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>( \pi_S(z) )</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>

<table>
<thead>
<tr>
<th>( z )</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>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi_P(z) )</td>
<td>1</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>2</td>
<td>6</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( z )</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi_P(z) )</td>
<td>3</td>
<td>7</td>
<td>11</td>
<td>15</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>16</td>
</tr>
</tbody>
</table>
Example 3.1

\[
\begin{align*}
K^1 &= 0011\ 1010\ 1001\ 0100 \\
K^2 &= 1010\ 1001\ 0100\ 1101 \\
K^3 &= 1001\ 0100\ 1101\ 0110 \\
K^4 &= 0100\ 1101\ 0110\ 0011 \\
K^5 &= 1101\ 0110\ 0011\ 1111
\end{align*}
\]

For the homework, please use \( \pi P' \):

\[
\begin{array}{c|cccccccc}
\text{z} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
\pi P'(z) & 1 & 5 & 9 & 13 & 14 & 2 & 6 & 10 \\
\text{z} & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
\hline
\pi P'(z) & 11 & 15 & 3 & 7 & 8 & 12 & 16 & 4
\end{array}
\]

FIGURE 3.1
A substitution-permutation network
Special case: a Feistel Cipher

Suggested 1973 by Horst Feistel for IBM’s “Lucipher”: Each state $w^i$ is split into a left and a right half, $w^i = L^i \parallel R^i$. For some $f$ (satisfying no special requirements!) each round encrypts by

\[
\begin{align*}
L^i &= R^{i-1} \\
R^i &= L^{i-1} \oplus f(R^{i-1}, K^i)
\end{align*}
\]

Decryption

Solve Exercise 3.2.
Binary (Boolean) algebra

0 = False, 1 = True. Inclusive disjunction (either this or that or both, (Lat.:) vel . . . vel . . . ) = ∨ = |  
Conjunction (this as well as that, (Lat.:) . . . et . . . ) = ∧ = & = .  
Exclusive disjunction (either this or that, but not both, (Lat.:) aut . . . aut . . . ) = non-equivalence (≠) = sum modulo 2 = ⊕

Facts

<table>
<thead>
<tr>
<th>⊕</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
a \oplus (b \oplus c) &= (a \oplus b) \oplus c & b \oplus a &= a \oplus b \\
a \oplus 0 &= 0 \oplus a = a & a \oplus a &= 0 \\
a \land (b \oplus c) &= (a \land b) \oplus (a \land c) & a \oplus \lnot a &= 1 \\
a \oplus 1 &= 1 \oplus a = \lnot a \\
a = b \oplus c &\iff a \oplus b \oplus c = 0 \\
a \oplus b &= (a \land \lnot b) \lor (\lnot a \land b) = (a \lor b) \land (\lnot a \lor \lnot b)
\end{align*}
\]
Binary Random Variable

A binary random variable $X$ assumes two values only, which we shall denote 0 and 1. It thus only distinguishes two probabilities, $\Pr[X = 0] = p$ and $\Pr[X = 1] = 1 - p$.

Bias

The bias of $X$ is defined to be $\text{Bi}[X] = p - \frac{1}{2}$. $X$ is fair if $\text{Bi}[X] = 0$.

Facts

\[
\begin{align*}
\Pr[X = 0] &= \frac{1}{2} + \text{Bi}[X] \\
\Pr[X = 1] &= \frac{1}{2} - \text{Bi}[X] \\
-\frac{1}{2} &\leq \text{Bi}[X] \leq \frac{1}{2} \\
\text{Bi}[X \oplus 1] &= -\text{Bi}[X]
\end{align*}
\]
Lemma (Piling-up Lemma)

For $k$ mutually independent random variables $X_1, \ldots, X_k$, 
$$\text{Bi}[X_1 \oplus \ldots \oplus X_k] = 2^{k-1} \prod_{i=1}^{k} \text{Bi}[X_i].$$

Proof.

By induction over $k$; obvious for $k = 1$. Let $Y$ denote $\bigoplus_{i=1}^{k-1} X_i$, and assume validity for $k-1$: $\text{Bi}[Y] = 2^{k-2} \prod_{i=1}^{k-1} \text{Bi}[X_i]$.

$$\Pr[\bigoplus_{i=1}^{k} X_i = 0] = \Pr[Y = 0] \Pr[X_k = 0] + \Pr[Y = 1] \Pr[X_k = 1]$$

$$= \left(\frac{1}{2} + \text{Bi}[Y]\right)\left(\frac{1}{2} + \text{Bi}[X_k]\right) + \left(\frac{1}{2} - \text{Bi}[Y]\right)\left(\frac{1}{2} - \text{Bi}[X_k]\right)$$

$$= \frac{1}{2} + 2\text{Bi}[Y]\text{Bi}[X_k],$$

so $\text{Bi}[\bigoplus_{i=1}^{k} X_i] = 2^{k-1} \prod_{i=1}^{k} \text{Bi}[X_i]$.\hfill \square$

Corollary

$X_1 \oplus \ldots \oplus X_k$ is fair if and only if one of the terms $X_i$ is fair.
Lemma (Piling-up Lemma)

For $k$ mutually independent random variables $X_1, \ldots, X_k$, $\text{Bi}[X_1 \oplus \ldots \oplus X_k] = 2^{k-1} \prod_{i=1}^{k} \text{Bi}[X_i]$.

Proof.

By induction over $k$; obvious for $k = 1$. Let $Y$ denote $\bigoplus_{i=1}^{k-1} X_i$, and assume validity for $k - 1$: $\text{Bi}[Y] = 2^{k-2} \prod_{i=1}^{k-1} \text{Bi}[X_i]$.

$$\Pr[\bigoplus_{i=1}^{k} X_i = 0] = \Pr[Y = 0] \Pr[X_k = 0] + \Pr[Y = 1] \Pr[X_k = 1]$$

$$= (\frac{1}{2} + \text{Bi}[Y])(\frac{1}{2} + \text{Bi}[X_k]) + (\frac{1}{2} - \text{Bi}[Y])(\frac{1}{2} - \text{Bi}[X_k])$$

$$= \frac{1}{2} + 2\text{Bi}[Y]\text{Bi}[X_k], \text{ so } \text{Bi}[\bigoplus_{i=1}^{k} X_i] = 2^{k-1} \prod_{i=1}^{k} \text{Bi}[X_i]$$

Corollary

$X_1 \oplus \ldots \oplus X_k$ is fair if and only if one of the terms $X_i$ is fair.
Linear Approximation of an S-box

For $\pi_S : \{0, 1\}^m \rightarrow \{0, 1\}^n$, try to find relations like

$Y_2 \approx X_1 \oplus X_3 \oplus X_4$; equivalently: a biased random variable

$Z(a_1, \ldots, a_m), (b_1, \ldots, b_n)(X, Y) = (\bigoplus_{i=1}^m a_i X_i) \oplus (\bigoplus_{i=1}^n b_i Y_i), Y = \pi_S(X)$.

### Linear Approximation Table

<table>
<thead>
<tr>
<th>$a$</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>0</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>14</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>10</td>
<td>4</td>
<td>10</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>12</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>6</td>
<td>4</td>
<td>10</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>10</td>
<td>12</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>4</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>12</td>
<td>10</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>10</td>
<td>6</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>6</td>
<td>10</td>
<td>8</td>
<td>12</td>
<td>10</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>8</td>
<td>12</td>
<td>6</td>
<td>10</td>
<td>4</td>
<td>10</td>
<td>6</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>8</td>
<td>12</td>
<td>8</td>
<td>4</td>
<td>12</td>
<td>8</td>
<td>12</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>3</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>8</td>
<td>6</td>
<td>12</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>12</td>
<td>10</td>
<td>4</td>
<td>6</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>10</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>12</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIGURE 3.2**
Linear approximation table: values of $N_L(a, b)$

Solve Exercise 3.12.

For each $a = (a_1, \ldots, a_m)$, $b = (b_1, \ldots, b_n)$, $N_L(a, b)$ is the number of $x$ such that $Z_{a,b}(x, y) = 0$, $y = \pi_S(x)$.

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

Nils Andersen

Composition, SPN, Linear/Differential Analysis
Linear Attack

Assume available a large number of corresponding plaintext-ciphertext pairs for the same unknown key. (Known-plaintext attack.) To find subkey $K^{Nr+1}$, combine biased variables like $X_1 \oplus X_3 \oplus X_4 \oplus Y_2$ and $X_2 \oplus Y_2 \oplus Y_4$ into a biased sum of plaintext bits, subkey bits and bits from $u^{Nr}$.

In Example 3.1, use

$$T_1 = U^1_5 \oplus U^1_7 \oplus U^1_8 \oplus V^1_6 \quad \text{Bi}[T_1] = \frac{1}{4}$$

$$T_2 = U^2_6 \oplus V^2_6 \oplus V^2_8 \quad \text{Bi}[T_2] = -\frac{1}{4}$$

$$T_3 = U^3_6 \oplus V^3_6 \oplus V^3_8 \quad \text{Bi}[T_3] = -\frac{1}{4}$$

$$T_4 = U^3_{14} \oplus V^3_{14} \oplus V^3_{16} \quad \text{Bi}[T_4] = -\frac{1}{4}$$
Linear Attack

Assuming (which is false!) linear independence among $T_1, T_2, T_3, T_4$, by the piling-up lemma on $T = T_1 \oplus T_2 \oplus T_3 \oplus T_4$ we obtain $\text{Bi}[T] = 2^3 \left( \frac{1}{4} \right) \left( -\frac{1}{4} \right)^3 = -\frac{1}{32}$. Insertion gives $T$

$$T = X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4 \oplus L$$

where $L = K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3 \oplus K_6^4 \oplus K_8^4 \oplus K_{14}^4 \oplus K_{16}^4$. Since $L$ is constant (0 or 1), we find

$$\text{Bi}[X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4] = \pm \frac{1}{32}$$
The final biased random variable only involves bits from the plaintext and from \(u^{Nr}\). These bits \(u^{Nr}_i\) participate in certain S-box inputs \(u^{Nr}_{j\langle j\rangle}\), with outputs \(v^{Nr}_{j\langle j\rangle}\) to be x-ored with \(M\) subkey bits \(K^{Nr+1}_{j\langle j\rangle}\). (In our example: the 8 subkey bits \(K^5_{\langle 2\rangle}\) and \(K^5_{\langle 4\rangle}\).)

Note that tracing the network backwards, \(u^{Nr}\) can be obtained from \(y\) and \(K^{Nr+1}\).

For each of the \(2^M\) subkey bit possibilities, count the number of cases in the test sample of \(N\) plaintext-ciphertext pairs where the biased variable is 0. Hopefully, for most of the subkey bit possibilities, this number will be close to \(N/2\), and for sufficiently large \(N\) (inversely proportional to the square of the bias), the correct subkey bits stand out.
Definition

For an input \( x - or \ x' = x \oplus x^* \) and an S-box \( \pi_S : \{0, 1\}^m \rightarrow \{0, 1\}^n \), the output x-or is

\[
y' = \pi_S(x) \oplus \pi_S(x^*).
\]

Definition

\[
\Delta(x') = \{(x, x^*) \mid x \oplus x^* = x'\}
\]

Differential Approximation Table

\[
\begin{array}{cccccccccccc}
\alpha' & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & A & B & C & D & E & F \\
\hline
0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 2 & 4 & 0 & 4 & 2 & 0 & 0 \\
2 & 0 & 0 & 0 & 2 & 0 & 6 & 2 & 2 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \\
3 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 4 & 2 & 0 & 2 & 0 & 0 \\
4 & 0 & 0 & 0 & 2 & 0 & 0 & 6 & 0 & 0 & 0 & 2 & 0 & 4 & 2 & 0 \\
5 & 0 & 4 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 4 & 0 & 2 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 2 \\
7 & 0 & 0 & 2 & 2 & 2 & 0 & 2 & 0 & 2 & 0 & 2 & 2 & 0 & 0 & 0 \\
8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 4 & 0 & 4 \\
9 & 0 & 0 & 2 & 0 & 2 & 2 & 0 & 4 & 2 & 0 & 2 & 2 & 0 & 0 & 0 \\
A & 0 & 0 & 2 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
B & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 2 \\
C & 0 & 2 & 2 & 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 \\
D & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 4 & 2 & 0 & 2 & 0 & 2 & 0 & 2 \\
E & 0 & 2 & 4 & 2 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
F & 0 & 2 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 4 & 0 & 2 & 0 & 0 & 0 \\
\end{array}
\]

\[
N_D(x', y') = |\{(x, x^*) \in \Delta(x') \mid \pi_S(x) \oplus \pi_S(x^*) = y'\}|.
\]

\[
R_D(x', y') = 2^{-m}N_D(x', y') \text{ approximates the probability of output x-or } y' \text{ given input x-or } x'.
\]
Differential Attack

Combine large propagation ratios into a differential trail through the network, arranging for the output x-or from each round to be the input x-or of the next round. The full trail leads from a particular input x-or $x'$ to a particular state x-or $(u^{Nr})'$. Assuming (which may not be mathematically valid) independence among the layers of the network, the propagation ratio $\varepsilon$ of the full differential trail may be computed by multiplication, and this will also be larger than for a random distribution.
Example 3.1

In $S^1_2$, use $R_P(1011, 0010) = 12$.
In $S^2_3$, use $R_P(0100, 0110) = 13$.
In $S^3_2$, use $R_P(0010, 0101) = 40$.
In $S^3_3$, use $R_P(0010, 0101) = 40$.

Giving $\frac{1}{2} \cdot \left(\frac{3}{8}\right)^3 = \frac{27}{1024}$ as propagation ratio for the full differential trail.
Those positions of $u^{Nr}$ where the corresponding position of $(u^{Nr})'$ is 1, depend on $M$ particular bits of $K^{Nr+1}$.

Assume a large collection of corresponding values $(x, x^*, y, y^*)$ have been obtained. (i.e.: ciphertexts $y$ and $y^*$ correspond to $x$ and $x^*$, respectively, all encodings employ the same (unknown) key, and each quadruple has the input $x$-or $x \oplus x^* = x'$ of the full differential trail.) (Chosen-plaintext attack.)

For each of the $2^M$ possibilities for the subkey bits in question, the following is done: The collection is run through; from $y$ and $y^*$ and the suggested subkey bits, $u^{Nr}$ and $(u^{Nr})^*$ are reconstructed, and the percentage with the correct sum $(u^{Nr})'$ is computed.

If the collection is sufficiently large (inversely proportional to propagation ratio $\varepsilon$), the $M$ correct subkey bits will stand out.