Conditional Linear Cryptanalysis
Cryptanalytic Techniques

- Differential and linear cryptanalysis are two major generic techniques for assessing the strength and vulnerabilities of block ciphers.
- These techniques have various extensions which can improve their success in various cases.
- Along with Davies’ attack, they are the best attacks against the Data Encryption Standard (DES).

<table>
<thead>
<tr>
<th>Technique</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differential Cryptanalysis</td>
<td>$2^{47}$</td>
</tr>
<tr>
<td>Linear Cryptanalysis</td>
<td>$2^{43}$</td>
</tr>
<tr>
<td>Improved Davies’ Attack</td>
<td>$2^{50}$</td>
</tr>
<tr>
<td>Conditional Linear Cryptanalysis</td>
<td>$\leq2^{42}$</td>
</tr>
</tbody>
</table>

- Today, we will show a new extension that reduces this complexity further.
Linear Cryptanalysis uses statistical approximations that approximate parity of subsets of bits of the plaintext, ciphertext, and the subkeys.

- E.g., (second bit of the plaintext) XOR (fifth bit of ciphertext) XOR (keys bits) = 0

Each approximation has a probability, \( p \), to hold

- Which is the fraction of plaintexts whose encryption follow the approximation
- In random cases, the parity is expected to be \( \frac{1}{2} \), or close to \( \frac{1}{2} \)

The ability if distinguish whether an approximation holds highly depends on the distance of the probability from \( \frac{1}{2} \)

Let the bias be \( \varepsilon = p - \frac{1}{2} \)

- Range: -\( \frac{1}{2} \) to +\( \frac{1}{2} \)
- The higher the (absolute value of the) bias, the easier to distinguish
Linear Approximations

- A one-round approximation is a tuple \((\lambda_P, \lambda_C, \lambda_K)\)
  - \(\lambda_P\) is a subset of bits of the plaintext
  - \(\lambda_C\) is a subset of bits of the ciphertext
  - \(\lambda_K\) is a subset of bits of the key (or the subkeys)

- The **probability of the approximation** is the probability that \(P\lambda_P \oplus C\lambda_C \oplus K\lambda_K = 0\)
  - Denoted by \(p = \frac{1}{2} + \epsilon\)
  - \(\epsilon\) is called the **bias of the approximation**

- We are usually interested in the highest \(|\epsilon|\)
  - \(\epsilon = \pm \frac{1}{2}\) means the approximation is linear
  - \(\epsilon = 0\) means that the approximation is mostly useless
Example

- The best non-trivial approximation of S5

\[ \lambda_p = 21 04 00 80 00 00 00 x \]

\[ \lambda_c = 21 04 00 80 00 00 08 00 x \]

- It approximates the second bit of input to the XOR of the four output bits.
  - In 12 cases: \( P \lambda_p \oplus C \lambda_c \oplus K \lambda_K = 0 \)
  - In 52 cases: \( P \lambda_p \oplus C \lambda_c \oplus K \lambda_K = 1 \)

\[ \frac{1}{2} + \varepsilon = \frac{12}{64} \]

\[ \varepsilon = \frac{-20}{64} \]
The Piling Up Lemma

- An $n_1$-round approximation $\lambda_1 = (\lambda^1_P, \lambda^1_C, \lambda^1_K)$ with bias $\varepsilon_1$ can be concatenated with an $n_2$-round approximation $\lambda_2 = (\lambda^2_P, \lambda^2_C, \lambda^2_K)$ with bias $\varepsilon_2$ if $\lambda^1_C$ equals the swapped value of the two halves of $\lambda^2_P$.

- The concatenation of the approximations is the $(n_1 + n_2)$-round approximation $\lambda = (\lambda^1_P, \lambda^2_C, \lambda^1_K \oplus \lambda^2_K)$.

- The bias of $\lambda$ is $\varepsilon = 2\varepsilon_1 \varepsilon_2$. 

![Diagram of the Piling Up Lemma](image-url)
The parity approximation of $\lambda$ holds

- If both the parity approximations of $\lambda_1$ and $\lambda_2$ hold, or
- If both do not hold

Therefore, the probability that $\lambda$ satisfies the parity approximation $\lambda$ is

$$\left(\frac{1}{2} + \varepsilon_1\right)\left(\frac{1}{2} + \varepsilon_2\right) + \left(\frac{1}{2} - \varepsilon_1\right)\left(\frac{1}{2} - \varepsilon_2\right) = \frac{1}{4} + \frac{1}{2}\varepsilon_1 + \frac{1}{2}\varepsilon_2 + \varepsilon_1\varepsilon_2 + \frac{1}{4} - \frac{1}{2}\varepsilon_1 - \frac{1}{2}\varepsilon_2 + \varepsilon_1\varepsilon_2 = \frac{1}{2} + 2\varepsilon_1\varepsilon_2$$
This linear approximation has probability \( \frac{1}{2} + 2\left(\frac{-20}{64}\right)^2 = \frac{1}{2} + \frac{25}{128} \).
Algorithm 1

- This algorithm finds the parity bits of the key involved in the approximation (total of one parity).
- Given $\lambda=(\lambda_P,\lambda_C,\lambda_K)$, $\varepsilon(\lambda)$, and $N$ plaintexts and their ciphertexts, the algorithm counts the number $M$ of plaintexts satisfying
  \[ P\lambda_P \oplus C\lambda_C = 0 \]
  - Recall that $P\lambda_P \oplus C\lambda_C \oplus K\lambda_K = 0$ holds with probability $p = \frac{1}{2} + \varepsilon$.
- The algorithm guesses that the parity of the key bits $K\lambda_K$ is
  \[
  \begin{array}{|c|c|c|}
  \hline
  & \varepsilon > 0 & \varepsilon < 0 \\
  \hline
  M > \frac{N}{2} & 0 & 1 \\
  \hline
  M < \frac{N}{2} & 1 & 0 \\
  \hline
  \end{array}
  \]
- This algorithm finds only one parity bit of the key.
- The success rate of the algorithm grows as the number of plaintexts $N$ increases, and as the value of $|\varepsilon|$ increases.
- For a high probability of success, $N \approx \frac{1}{\varepsilon^2}$ or higher.
Matsui’s Best Approximation (14 rounds)
Algorithm 2

- This algorithm finds the parity bits of the key involved in the approximation as well as bits of the first and last subkeys.
- Given $\lambda=(\lambda_p, \lambda_c, \lambda_K), \varepsilon(\lambda)$, and $N$ plaintexts and their corresponding ciphertexts.
- The algorithm partially encrypts/decrypts the first and the last round by all the possible values of the key bits entering the active S boxes in both the first and last round.
  - Total trial of 12 bits (6+6)
- For each trial key value $i$, we denote by $P_i$ the data after the first round, and by $C_i$ the data before the last round.
- For each trial key value $i$ the algorithm counts the number $M_i$ of plaintexts satisfying $P_i\lambda_p \oplus C_i\lambda_c = 0$.
- The algorithm deduces that the real value of the tried key bits is the $i$ that maximizes $|M_i - \frac{N}{2}|$.
- For a high probability of success, $N \approx c \frac{1}{\varepsilon^2}$ or higher.
  - For some small constant $c$ (e.g., 4 or 8).
Linear Cryptanalysis of the Full 16-Round DES

- Matsui uses the best 14-round approximation with probability $\approx \frac{1}{2} - 2^{-20.75}$
- The attack requires about $2^{43}$ known plaintexts
Conditional Linear Cryptanalysis

- Using conditions to discard data that reduces that bias
  - So the bias of the remaining data increase or decrease
- Conditions can be by any observable data available to the cryptanalyst
  - Plaintexts, ciphertexts, and formulae on them
- Such as (e.g., in Feistel ciphers)
  - Validity of other linear approximations
  - Inputs of $F$ in the first and last rounds
  - XORs of the outputs of $F$ in all even rounds (or all odd rounds)
  - Etc.
- Conditions may be by a single (parity) bit of the above, or by several
  - Including by a distribution of data by several bits, or
  - selection of several cases from such a distribution
A Short Approximation Conditioned by a Containing One

- Assume that we have an $n$-round approximation $\lambda$ that is the concatenations of two shorter approximations $\lambda_1, \lambda_2$ where the biases of $\lambda_1$ and $\lambda_2$ are exactly the same (with same sign).
- Suppose that we know that $\lambda$ holds, what is the bias of $\lambda_1$ in this case?
  - We call it the conditional bias.
- In other words:
  - Discard all the data that does not follow the predictions of $\lambda$.
  - What is the bias of $\lambda_1$ in the remaining data?
- Answer: about twice the original biases.
A Simple Example

- Consider the following three-round linear approximation $\lambda$
  - Remark: this is the best three-round approximation
- We denote the first two rounds of $\lambda$ by $\lambda_1$, and the last round of $\lambda$ by $\lambda_2$
  - The probability of both $\lambda_1$ and $\lambda_2$ is $\frac{1}{2} - 0.31$
  - The probability of $\lambda$ is $\frac{1}{2} + 0.19$
  - The probability of $\lambda_1$ given that $\lambda$ satisfies the parity approximation is $\frac{1}{2} - 0.45$
- Hidden assumption
  - Both parities of involved key bits of $\lambda_1$ and $\lambda_2$ are the same
  - Home exercise: bypass this assumption
Why the Conditional Bias Increases?

Consider the four cases:

<table>
<thead>
<tr>
<th></th>
<th>$\lambda_2$ holds</th>
<th>$\frac{1}{2} + \varepsilon$</th>
<th>$\lambda_2$ fails</th>
<th>$\frac{1}{2} - \varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$ holds</td>
<td>$\lambda$ holds</td>
<td>$\frac{1}{2} + \varepsilon$</td>
<td>$\lambda$ fails - data discarded</td>
<td>$(\frac{1}{2} + \varepsilon)(\frac{1}{2} - \varepsilon) = \frac{1}{4} - \varepsilon^2$</td>
</tr>
<tr>
<td>$\frac{1}{2} + \varepsilon$</td>
<td>$(\frac{1}{2} + \varepsilon)^2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\lambda_1$ fails</td>
<td>$\lambda$ fails - data discarded</td>
<td>$\frac{1}{2} - \varepsilon$</td>
<td>$\lambda$ holds</td>
<td>$(\frac{1}{2} - \varepsilon)^2$</td>
</tr>
<tr>
<td>$\frac{1}{2} - \varepsilon$</td>
<td>$(\frac{1}{2} - \varepsilon)(\frac{1}{2} + \varepsilon) = \frac{1}{4} - \varepsilon^2$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The two red cases have exactly the same probabilities, but in one $\lambda_1$ holds, and in the other it fails $\Rightarrow$ the bias of $\lambda_1$ in the discards data is exactly 0.

Since the original bias is the weighted average of both, the conditional bias is higher than the original bias.

In cases $\varepsilon$ is very small, the discarded data is almost half the data $\Rightarrow$ the conditional bias becomes about twice of the original.
Why the Conditional Bias Increases?  
An Accurate Calculation

- Consider $\lambda$ an $n$-round linear approximation, which is concatenation of two linear approximations $\lambda_1$ and $\lambda_2$, where the probability of both $\lambda_1$ and $\lambda_2$ is $\frac{1}{2} + \varepsilon$.

- The probability of $\lambda$ is
  - $(\frac{1}{2} + \varepsilon)^2 + (\frac{1}{2} - \varepsilon)^2 = \frac{1}{2} + 2\varepsilon^2$

- The probability that both $\lambda_1$ and $\lambda_2$ satisfy the parity approximations is
  - $(\frac{1}{2} + \varepsilon)^2 = \frac{1}{4} + \varepsilon + \varepsilon^2$
Why the Conditional Bias Increases?
An Accurate Calculation

- Given the fact that $\lambda$ satisfies the parity approximation, the probability $\frac{1}{2} + \epsilon^*$ that $\lambda_1$ satisfies the parity approximation is:

$$\frac{1}{2} + \epsilon^* = \frac{1}{4} + \epsilon + \epsilon^2 = \frac{1}{2 + 2\epsilon^2} + \frac{\epsilon}{2 + 2\epsilon^2} = \frac{1}{2} + \frac{\epsilon}{2 + 2\epsilon^2}$$

- Therefore, $\frac{\epsilon^*}{\epsilon} = \frac{1}{2 + 2\epsilon^2} = 2 + o(\epsilon^2)$

- In most cases $\epsilon$ is very small $\Rightarrow \epsilon^* \approx 2\epsilon$
A Second Example

- Based on Matsui’s best approximation
  - 8-round iterative approximation
- We denote the first four rounds by $\lambda_1$, and the last four rounds by $\lambda_2$
  - $\varepsilon(\lambda_1) = \varepsilon(\lambda_2) \approx 2^{-14}$
- The approximations in $\lambda_1$ are the same as in $\lambda_2$
  - but in a different order of rounds
  - i.e., $\lambda_1$ and $\lambda_2$ have the same total biases
- $\varepsilon(\lambda_1 | \lambda) \approx 2\varepsilon(\lambda_1) \approx 2^{-13}$
Remarks

- Same holds for $\lambda_2$, it has the same conditional bias as $\lambda_1$
- Same holds for any “approximation” $\lambda_1$ combining any subset of rounds, where the rest of the rounds form $\lambda_2$, and both have the same biases
  - I.e., also holds for non-consecutive rounds
  - E.g., $\lambda_1$ contains all rounds divisible by 4, while $\lambda_2$ contains the rest
- Same phenomenon can be applied to cases with different biases
  - But the factor reduces below 2
A Case of Single Round

- The best non-trivial approximation of S5:

- It approximates the second bit of input to the XOR of the four output bits
  - Probability $\frac{1}{2} \cdot \frac{20}{64}$
  - I.e., 12 cases with equality (parity 0 of the 5 bits), 52 cases with inequality (parity 1)
A Case of Single Round

- Conditioning on all the four output bits of S5 (16 cases) we get

<table>
<thead>
<tr>
<th>0000</th>
<th>0001</th>
<th>0010</th>
<th>0011</th>
<th>0100</th>
<th>0101</th>
<th>0110</th>
<th>0111</th>
<th>1000</th>
<th>1001</th>
<th>1010</th>
<th>1011</th>
<th>1100</th>
<th>1101</th>
<th>1110</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.5</td>
<td>-0.5</td>
<td>-0.5</td>
<td>-0.5</td>
<td>-0.5</td>
<td>-0.5</td>
<td>0</td>
<td>-0.5</td>
<td>0</td>
<td>-0.5</td>
<td>0.25</td>
<td>-0.25</td>
<td>0</td>
<td>-0.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Consider a condition on the LSB of the four output bits of S5 (a single bit)

<table>
<thead>
<tr>
<th>Condition</th>
<th>0</th>
<th>1</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>12</td>
<td>52</td>
<td>-20/64</td>
</tr>
<tr>
<td>LSB=0</td>
<td>11</td>
<td>21</td>
<td>-5/32=-10/64</td>
</tr>
<tr>
<td>LSB=1</td>
<td>1</td>
<td>31</td>
<td>-15/32=-30/64</td>
</tr>
</tbody>
</table>

\[ P\lambda_p \oplus C\lambda_c \oplus K\lambda_k = \]
A Case of Single Round

- Scan from Adi Shamir's CRYPTO'85 paper
  - He circled the values with an even parity of the four output bits

The 12 VS. the 52

- $1$ vs. $31$ for $\text{LSB}=1$, and $11$ vs. $21$ for $\text{LSB}=0$
Another Type of Conditions

- In Feistel ciphers we can compute various linear combinations of internal bits directly from the plaintext and ciphertext.
- In particular, we can compute the XOR of the outputs of the F function of the odd rounds, and similarly of the even rounds.
Another Type of Conditions

- We can condition on the XOR of plaintext and ciphertext bits
  - even more than one bit at a time
- For example, on \( P_L \oplus C_R = \bigoplus_{r \text{ is odd}} Y^r \)
  - which is the XOR of the output of \( F \) in all odd rounds
- Consider any one of these bits as a linear approximation
  - E.g., \( P_{L,17} \oplus C_{R,17} = 0 \)
    - Equivalent to \( Y_{17}^1 \oplus Y_{17}^3 = 0 \)
    - Such approximations are expected to have bias 0
- But they are very useful as conditions to other approximations
A Four-Round Example

- Consider four successive rounds taken from Matsui's best linear approximation
- This approximation uses three active S boxes:
  - S5 on the first and third rounds, and
  - S1 on the fourth round
- Both odd rounds have the same active S box. 
A Four-Round Example

- Conditioning on all the four XOR output bits of $S_5$ (16 cases) we get

<table>
<thead>
<tr>
<th>0000</th>
<th>0001</th>
<th>0010</th>
<th>0011</th>
<th>0100</th>
<th>0101</th>
<th>0110</th>
<th>0111</th>
<th>1000</th>
<th>1001</th>
<th>1010</th>
<th>1011</th>
<th>1100</th>
<th>1101</th>
<th>1110</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.008</td>
<td>0.008</td>
<td>0</td>
<td>0.009</td>
<td>0</td>
<td>0.009</td>
<td>0</td>
<td>0.014</td>
<td>0</td>
<td>0.014</td>
<td>0</td>
<td>0.015</td>
<td>0</td>
<td>0.015</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Notice that this condition is based on the XOR of both odd rounds
  - Not just on one of them

- For applying Matsui’s Algorithm 1 with our observation, we discard half of the known plaintexts, and use only the plaintexts in which the XOR of the LSB bits of $S_5$ is zero.
  - Their average bias is 0.0115
  - While the bias over all cases is 0.0057

- Using only these plaintexts increases the bias by a factor of two.
A Four-Round Example

- We need a quarter of the data
  - Compared to a regular linear attack with the same approximation
  - But this is after we discard half of the given data that fails the condition
- We need half of the original data
  - We discard half of it, and get the required quarter
An Eight-Round Example

- Consider eight successive rounds of Matsui's linear approximation
- We denote the first four rounds by $\lambda_1$, and the last four rounds by $\lambda_2$
- The rounds of $\lambda_1$ are the same rounds of $\lambda_2$, but in different order
An Eight-Round Example

- Conditioning $\lambda_1$ (or $\lambda_2$) on all the four XOR output bits of S5 (16 cases) we get

<table>
<thead>
<tr>
<th>XOR of the LSB bits of S5 is 0</th>
<th>XOR of the LSB bits of S5 is 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rounds 1,3</td>
<td>Rounds 1,3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XOR of the LSB bits of S5 is 0</th>
<th>XOR of the LSB bits of S5 is 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rounds 5,7</td>
<td>Rounds 5,7</td>
</tr>
</tbody>
</table>

- Consider the four cases
  - Let $X$ be the XOR of the LSB bits of the outputs of S5 in all odd rounds (1,3,5,7)
  - $X$ is known to the cryptanalyst

<table>
<thead>
<tr>
<th>XOR of the LSB bits of S5 is 0</th>
<th>XOR of the LSB bits of S5 is 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rounds 1,3</td>
<td>Rounds 1,3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XOR of the LSB bits of S5 is 0</th>
<th>XOR of the LSB bits of S5 is 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rounds 5,7</td>
<td>Rounds 5,7</td>
</tr>
</tbody>
</table>

$\Rightarrow \varepsilon(\lambda|X = 1) = 0$
We had just seen that $\varepsilon(\lambda|X = 1) = 0$.

Therefore $\varepsilon(\lambda|X = 0) \approx 2\varepsilon(\lambda)$.

And just as in the earlier example, we can save half of the data required by the attack.

A similar improvement on the conditional approximation holds for any multiple of four rounds.
Conditional Linear Cryptanalysis of the Full 16-Round DES

- The attack is based on two 14-round linear approximations, which provides candidates for 26 key bits:
  - $\lambda_1$ is the best 14-round linear approximation, same as Matsui uses
  - $\lambda_2$ differs from $\lambda_1$ (only) in the first round
Conditional Linear Cryptanalysis of the Full 16-Round DES

\[ \lambda_p = 21 04 00 80 \ 00 01 10 00, \]

\[ \lambda_1 \]

\[ \lambda_2 \]

\[ \lambda_c = 00 00 00 00 \ 01 04 00 80, \]
Conditional Linear Cryptanalysis of the Full 16-Round DES

- By the above approximations:
  - We can recover 13 key bits from the linear equation derived from $\lambda_1$
    - by applying the equation to fourteen consecutive rounds from the 2nd round to the 15th round
    - This is Matsui’s attack with $2^{43}$ known plaintexts
  - We can recover 19 key bits from the linear equation derived from $\lambda_2$
    - by applying the equation to fourteen consecutive rounds from the 2nd round to the 15th round
    - With about $2^{46}$ known plaintexts
  - Six bits are common to both
    - so the total is $13+19-6=26$
Conditional Linear Cryptanalysis of the Full 16-Round DES

- Conditioning $\lambda_1$ on all the four XOR output bits of S5 (16 cases) we get

<table>
<thead>
<tr>
<th></th>
<th>0000</th>
<th>0001</th>
<th>0010</th>
<th>0011</th>
<th>0100</th>
<th>0101</th>
<th>0110</th>
<th>0111</th>
<th>1000</th>
<th>1001</th>
<th>1010</th>
<th>1011</th>
<th>1100</th>
<th>1101</th>
<th>1110</th>
<th>1111</th>
</tr>
</thead>
</table>

- Conditioning $\lambda_2$ on all the four XOR output bits of S5 (16 cases) we get

<table>
<thead>
<tr>
<th></th>
<th>0000</th>
<th>0001</th>
<th>0010</th>
<th>0011</th>
<th>0100</th>
<th>0101</th>
<th>0110</th>
<th>0111</th>
<th>1000</th>
<th>1001</th>
<th>1010</th>
<th>1011</th>
<th>1100</th>
<th>1101</th>
<th>1110</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-2^{-20.26}$</td>
<td>$2^{-23.13}$</td>
<td>$-2^{-20.26}$</td>
<td>$2^{-23.13}$</td>
<td>$-2^{-20.26}$</td>
<td>$2^{-23.13}$</td>
<td>$-2^{-20.26}$</td>
<td>$2^{-23}$</td>
<td>$-2^{-20.26}$</td>
<td>$2^{-23}$</td>
<td>$-2^{-20.26}$</td>
<td>$2^{-23}$</td>
<td>$-2^{-20.26}$</td>
<td>$2^{-23}$</td>
<td>$-2^{-20.26}$</td>
<td>$2^{-23}$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition</th>
<th>Bias</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>$\sim -2^{-20.75}$</td>
<td>$\sim -2^{-21.48}$</td>
</tr>
<tr>
<td>LSB=0</td>
<td>$\sim -2^{-21.74}$</td>
<td>$\sim -2^{-20.26}$</td>
</tr>
<tr>
<td>LSB=1</td>
<td>$\sim -2^{-20.16}$</td>
<td>$\sim 2^{-23.06}$</td>
</tr>
</tbody>
</table>

$\sim -2^{-20.21}$ average of the red entries
Conditional Linear Cryptanalysis of the Full 16-Round DES

- We apply a variant of Algorithm 2 as follows
- For each plaintext and the corresponding ciphertext, we partially decrypt S5 in the last round using all $2^6$ possible key candidates
  - For each guess, we get an estimate for the output of S5 in round 16
- According to the estimated value of the XOR of the LSB bits of S5 in the odd rounds, we decide which approximation to use
  - If the XOR of the LSB bits of S5 in the even rounds 2-14 is 0, we use $\lambda_2$, and partially encrypt S3 and S4 in the first round using all $2^{12}$ possible key candidates
  - If the XOR of the LSB bits of S5 is 1, we use $\lambda_1$, and partially encrypt S1 in the first round using all $2^6$ possible key candidates
  - We increment counters accordingly
- Analysis is more complex that just taking the maximal observed bias, as we count in two different arrays
  - But is can still be done with about $2^{42}$ known plaintexts
  - Using some auxiliary techniques
Applications to Other Types of Cryptanalysis

- The idea of conditioning approximation on other approximations can be extended to differential cryptanalysis
  - Conditioning characteristic (or differentials) on approximations
- But the improvement in complexity will be relatively small
  - I.e., only second order saving
- Because the increase in probability is linear with the decrease in number of required data
  - So still the same order of data is required before discarding the ones that fail the conditions
In this talk we showed that linear approximations are highly affected by conditioning them on other approximations.

The simplest case is conditioning an approximation on a containing, longer, approximation.

But it is also possible to condition on a “bad” approximation (e.g., with bias 0).

E.g., any parity of bits of the plaintext and Ciphertext, or several of such.

In particular, parity that relates to XOR of outputs of same S box in all even (or all odd) rounds.

And to condition a “bad” approximation (e.g., with bias 0) on another one.

Gaining a better bias.

We also showed that such conditional approximations can be combined in new ways to create unexpected conditional approximations.

Including ones we had no time to show here.
Summary

- And showed how to use such conditional approximations for attacks
  - Leading to the best current attack against DES
- The required data decreases linearly with the increase in the bias
  - Since the data (after discarding by the condition) decreases quadratically with the bias
- We tested most techniques with our test programs
Success Probability by Complexity (KPs&Time)
Known Plaintexts when Time is Fixed to $2^{50}$
Success Probability by Time for Various #KPs
Matsui’s Success Prob. by Time for Various #KPs
The End