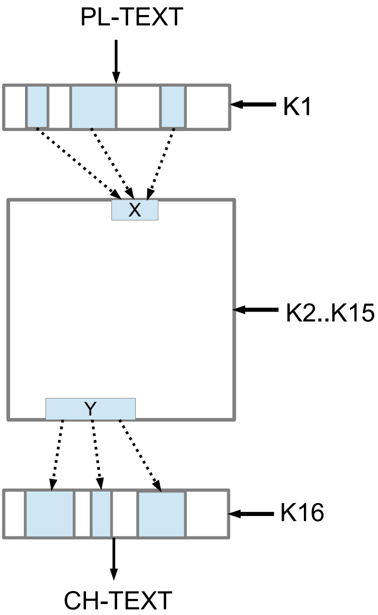
Separable Statistics in Linear Cryptanalysis

Igor Semaev,
Univ. of Bergen, Norway

joint work with Stian Fauskanger

5 September 2017, MMC workshop

Round Block Cipher Cryptanalysis



Logarithmic Likelihood Ratio(LLR) Statistic

- To distinguish two distributions with densities $P(x), Q(x)$
- by independent observations $\nu_1, .., \nu_n$
- Most powerful criteria(Neyman-Pearson lemma):
- accept $P(x)$ if

$$\sum_{i=1}^{n} \ln \frac{P(\nu_i)}{Q(\nu_i)} > \mathit{threshold}$$

- left hand side function is called LLR statistic

LLR Statistic for large (X, Y)?

- Approximate distribution of (X, Y) depends on some bits of $K2, .., K15$
- Observation on (X, Y) depends on some bits of $K1, K16$
- $\bar{K}$ key-bits which affect distribution and observation
- For large (X, Y) LLR statistic depends on many key-bits $\bar{K}$
- Conventional Multivariate Linear Cryptanalysis not efficient:
- $2^{|\bar{K}|}$ computations of the statistic to range the values of $\bar{K}$
- **Our work**: $<< 2^{|\bar{K}|}(\approx 10^3$ times faster in DES)
- by using a new statistic
- which reflects the structure of the round function
- that has a price to pay, but trade-off is positive

LLRs for Projections

- $(h_1, .., h_m)$ some linear projections of (X, Y) such that
- distr/observ of h_i depends on a lower number of key-bits $\bar{K}_i$
- happens for modern ciphers with small S-boxes
- Vector $(LLR_1, .., LLR_m)$ asymptotically distributed
- $\mathbf{N}(n\mu, nC)$ if the value of $\bar{K}$ is correct
- and close to $\mathbf{N}(-n\mu, nC)$ if the value of $\bar{K}$ is incorrect
- mean vector μ, covariance matrix C, number of plain-texts n

Separable Statistics

- LLR statistic S to distinguish two normal distributions
- quadratic, but in our case degenerates to linear
- $S(\bar{K},\nu) = \sum_{i=1}^{m} S_i(\bar{K}_i,\nu_i)$, where $S_i = \omega_i \, LLR_i$
- ω_i weights, ν observation on (X,Y), and ν_i observation on h_i
- S distributed $\mathbf{N}(a,a)$ if $\bar{K} = k$ correct
- close to $\mathbf{N}(-a,a)$ if $\bar{K} = k$ incorrect, for an explicit a
- For polynomial schemes the theory of separable statistics was developed by Ivchenko, Medvedev,.. in 1970-s
- Problem: find $\bar{K} = k$ such that $S(k,\nu) > \textit{threshold}$ without brute force

Reconstruct a set of $\bar{K}$-candidates k

- find solutions $\bar{K} = k$ to (linear for DES) equations

$$\begin{cases} \bar{K}_i & = k_i \quad \text{with weight } S_i(k_i, \nu_i) \\ i & = 1, .., m \end{cases}$$

- such that $S(k, \nu) = \sum_{i=1}^{m} S_i(k_i, \nu_i) > \textit{threshold}$
- the system is sparse: $|\bar{K}|$ is large, but $|\bar{K}_i| << |\bar{K}|$
- Walking over a search tree
- Algorithm first appears in I. Semaev, *New Results in the Linear Cryptanalysis of DES*, Crypt. ePrint Arch., 361, May 2014
- We compute success rate and the number of wrong solutions
- that is $\bar{K}$-candidates to brute force

Reconstruction Toy Example

S_1	0.1	0.2	0.3	0.1
$x_1 + x_2$	0	0	1	1
x_3	0	1	0	1

S_2	0.5	0.1
$x_1 + x_3$	0	1

S_3	0.4	0.5	0.7	0.1
x_1	0	0	1	1
$x_2 + x_3$	0	1	0	1

find x_1, x_2, x_3 s.t.

$$S(x_1, x_2, x_3) = S_1(x_1 + x_2, x_3) + S_2(x_1 + x_3) + S_3(x_1, x_2 + x_3) > 1$$

Solutions $010, 111$

Implementation for 16-Round DES

- 2 strings of 14 internal bits each(or a 28-bit string)
- 54 key-bits involved
- we use 28 of 10-bit projections, each involves $\approx$ 20 key-bits
- two separable statistics, one for each 14-bit string
- success probability 0.85(theoretically)
- number of (56-bit key)-candidates is $2^{41.8}$(theoretically&empirically) for $n = 2^{41.8}$
- search tree complexity is about the same

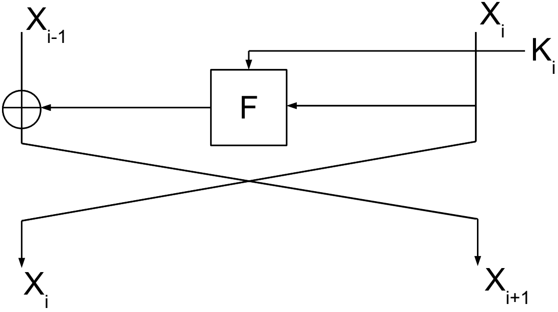
Further Talk Outline

- Formulae for internal bits probability distribution
- Construction of the statistic S
- Search tree algorithm
- Implementation details for 16-round DES

Probability of events in encryption(a priori distribution)

- Z vector of some internal bits in the encryption algorithm
- we want to compute $\mathbf{Pr}(Z = A)$ over all possible A
- that makes a distribution of Z
- More generally, $\mathbf{Pr}(\mathcal{E})$ for some event $\mathcal{E}$ in the encryption

Notation: one Feistel round



- in DES
- X_{i-1}, X_i are 32-bit blocks
- K_i is 48-bit round key
- sub-key of the main 56-bit key

Prob. Description of r-round Feistel (for SPN similar)

- $X_0, X_1, \ldots, X_{r+1}$ random independently uniformly generated m-bit blocks
- Main event $\mathcal{C}$ defines DES:

$$X_{i-1} \oplus X_{i+1} = F_i(X_i, K_i), \quad i = 1, \ldots, r$$

  $K_1, \ldots, K_r$ fixed round keys
- Then

$$\mathbf{Pr}(\mathcal{E}|\mathcal{C}) = \frac{\mathbf{Pr}(\mathcal{E}\mathcal{C})}{\mathbf{Pr}(\mathcal{C})} = 2^{mr}\mathbf{Pr}(\mathcal{E}\mathcal{C}).$$

- likely depends on all key-bits.

Approximatie Probabilistic Description

- We want **approximate** probability of $\mathcal{E}$ in the encryption
- Choose a larger event $\mathcal{C}_\alpha \supseteq \mathcal{C}$:
- $$\mathbf{Pr}(\mathcal{E}|\mathcal{C}) \approx \mathbf{Pr}(\mathcal{E}|\mathcal{C}_\alpha) = \frac{\mathbf{Pr}(\mathcal{E}\mathcal{C}_\alpha)}{\mathbf{Pr}(\mathcal{C}_\alpha)}$$
- $\mathbf{Pr}(\mathcal{E}|\mathcal{C}_\alpha)$ may depend on a lower number of key-bits
- Easier to compute and use

How to Choose $\mathcal{C}_\alpha$

- To compute the distribution of the random variable

$$Z = X_0[\alpha_1], X_1[\alpha_2 \cup \beta_1], X_r[\alpha_{r-1} \cup \beta_r], X_{r+1}[\alpha_r]$$

- ($X[\alpha]$ sub-vector of X defined by α), we choose trail

$$X_i[\beta_i], F_i[\alpha_i], \quad i = 1, \dots, r$$

- and event $\mathcal{C}_\alpha$:

$$X_{i-1}[\alpha_i] \oplus X_{i+1}[\alpha_i] = F_i(X_i, K_i)[\alpha_i], \quad i = 1, \dots, r.$$

- $\mathbf{Pr}(\mathcal{C}_\alpha) = 2^{-\sum_{i=1}^{r} |\alpha_i|}$

Regular trails

- trail

$$X_i[\beta_i], F_i[\alpha_i], \quad i = 1, \ldots, n$$

- is called regular if

$$\gamma_i \cap (\alpha_{i-1} \cup \alpha_{i+1}) \subseteq \beta_i \subseteq \gamma_i, \quad i = 1, \ldots, n.$$

- $X_i[\gamma_i]$ input bits relevant to $F_i[\alpha_i]$
- For regular trails $\mathbf{Pr}(Z = A|\mathcal{C}_\alpha)$ is computed with a convolution-type formula, only depends on α_i

Convolution Formula

- $Z = X_0[\alpha_1], X_1[\alpha_2 \cup \beta_1], X_r[\alpha_{r-1} \cup \beta_r], X_{r+1}[\alpha_r]$
- $\mathbf{Pr}(Z = A_0, A_1, A_r, A_{r+1} | \mathcal{C}_\alpha) =$

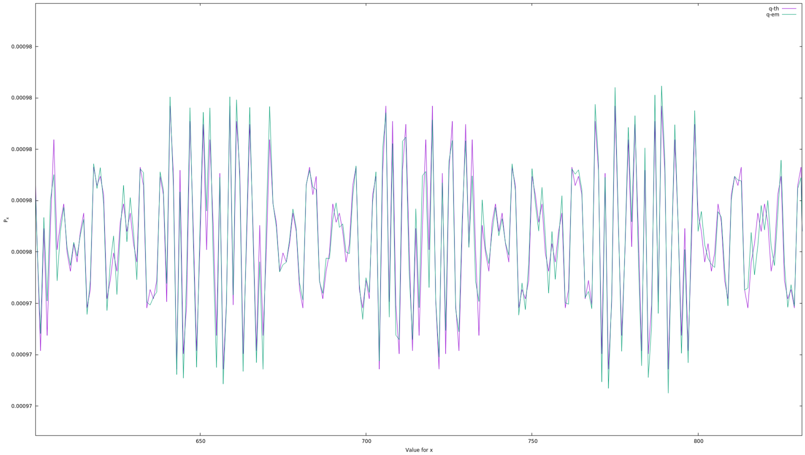
$$\frac{2^{\sum_{i=2}^{r-1} |\alpha_i|}}{2^{\sum_{i=1}^{r} |(\alpha_{i-1} \cup \alpha_{i+1}) \setminus \beta_i|}} \sum_{A_2,\ldots,A_{r-1}} \prod_{i=1}^{r} \mathbf{q}_i(A_i[\beta_i], (A_{i-1} \oplus A_{i+1})[\alpha_i], k_i),$$

- probability distribution of round sub-vectors

$$\mathbf{q}_i(b, a, k) = \mathbf{Pr}(X_i[\beta_i] = b, F_i[\alpha_i] = a \,|\, K_i[\delta_i] = k_i)$$

- $K_i[\delta_i]$ key-bits relevant to $F_i[\alpha_i]$
- Corollary: compute iteratively by splitting encryption into two parts. Few seconds for 14-round DES

Theoretical(red) vs Empirical(green) Distributions



- $X_2[24, 18, 7, 29], X_7[16, 14], X_8[24, 18, 7, 29]$
- Emp. with 2^{39} random pl-texts for one randomly chosen key

Approximate Distribution of a Vector from 14-round DES

- $X_2[24, 18, 7, 29], X_{15}[16, 15, .., 11], X_{16}[24, 18, 7, 29]$
- computed with the trail

round i	β_i, α_i
$2, 6, 10, 14$	$\emptyset, \emptyset$
$3, 5, 7, 9, 11, 13$	$\{15\}, \{24, 18, 7, 29\}$
$4, 8, 12$	$\{29\}, \{15\}$
15	$\{16, \ldots, 11\}, \{24, 18, 7, 29\}$

- depends on 7 key-bits:

$$K_{\{3,5,7,9,11,13\}}[22] \oplus K_{\{4,8,12\}}[44], K_{15}[23, 22, 21, 20, 19, 18].$$

- notation $K_{\{4,8,12\}}[44] = K_4[44] \oplus K_8[44] \oplus K_{12}[44]$

Another Approximation to the Same Distribution

- same $X_2[24, 18, 7, 29], X_{15}[16, 15, .., 11], X_{16}[24, 18, 7, 29]$
- with another trail

round i	β_i, α_i
2	$\emptyset, \emptyset$
$3, 5, 7, 9, 11, 13$	$\{16, 15, 14\}, \{24, 18, 7, 29\}$
$4, 6, 8, 10, 12, 14$	$\{29, 24\}, \{16, 15, 14\}$
15	$\{16, \ldots, 11\}, \{24, 18, 7, 29\}$

- different distribution
- quadratic imbalance is negligibly larger
- but depends on a much larger number of the key-bits

Conventional LLR statistic

- We use 28 internal bits in the analysis of DES:

$$X_2[24,18,7,29], X_{15}[16,15,..,11], X_{16}[24,18,7,29]$$
$$X_1[24,18,7,29], X_2[16,15,..,11], X_{15}[24,18,7,29]$$

- distribution and observation depend on available plain-text/cipher-text and 54 key-bits
- conventional LLR statistic takes 2^{54} computations
- no advantage over Matsui's 2^{43} complexity for breaking DES

Attack

- We used 28 projections($i, j \in \{16, .., 11\}$):

$$X_2[24, 18, 7, 29], X_{15}[i, j], X_{16}[24, 18, 7, 29]$$
$$X_1[24, 18, 7, 29], X_2[i, j], X_{15}[24, 18, 7, 29]$$

- except $i = 16, j = 11$, where the distributions are uniform
- For each projection LLR statistic depends on ($\leq$21) key-bits
- We constructed two new separable statistics for two independent bunches of the projections
- and combined ($\leq$ 21)-bit values to find a number of candidates for 54-bit sub-key
- brute force those candidates

Separable Statistics in Details

- observation $\nu = (\nu_1, \ldots, \nu_m)$ on m projections $(h_1, .., h_m)$
- ν_i depends on plain/cipher-texts and $\bar{K}_i$
- best statistic is approx. separable: $S(\bar{K}, \nu) = \sum_{i=1}^{m} S_i(\bar{K}_i, \nu_i)$
- $S_i(\bar{K}_i, \nu_i)$ weighted LLR statistics for $h_i(\mathtt{x})$
- Construct $\bar{K}$-values (s.t. $\sum_{i=1}^{m} S_i(\bar{K}_i, \nu_i) >$ threshold) from $\bar{K}_i$-values
- One computes error probabilities etc., details are below

Separable Statistic Construction

- $\mathtt{x}$ may have distribution Q or P. Projection $h_i(\mathtt{x})$ may have Q_i or P_i $i = 1, .., m$
- n plain/cipher-texts
- LLR statistic for h_i: $LLR_i = \sum_b \nu_{ib} \ln\left(\frac{q_{ib}}{p_{ib}}\right)$
- $(LLR_1, \ldots, LLR_m)$ normally distributed
- $\mathbf{N}(n\mu_Q, nC_Q)$ or $\mathbf{N}(n\mu_P, nC_P)$
- If Q is close to P, then $\mu_Q \approx -\mu_P$(follows from Baigneres et al. 2004) and $C_Q \approx C_P$(this work)
- We get $\mathbf{N}(n\mu, nC)$ or $\mathbf{N}(-n\mu, nC)$

Construct Separable Statistics 1

- assume non-singular C, always the case in our analysis of DES
- To distinguish $\mathbf{N}(-n\mu, nC), \mathbf{N}(n\mu, nC)$ we use LLR statistic S
- which degenerates to linear

$$S = (\frac{C^{-1}\mu}{n})(LLR_1, \ldots, LLR_m)^T$$

- So that $S(\bar{K}, \nu) = \sum_{i=1}^{m} S_i(\bar{K}_i, \nu_i)$, where $S_i = \omega_i LLR_i$
- weights ω_i entries of the vector $\frac{C^{-1}\mu}{n}$

Covariance Matrix C for Linear Projections

- random variable $\mathtt{x}$ may have uniform P or a distribution Q close to P
- assume m linear projections $h_i(\mathtt{x})$
- $\text{rank}(h_i)$ is r_i and $\text{rank}(h_i, h_j)$ is r_{ij}
- then
$$C = \left[(2^{r_i + r_j - r_{ij}} - 1)\mu_i \mu_j\right]_{ij}$$
- easy to compute and check singularity of C

Distribution of the Main Statistic S

- Assume P is close to Q
- if x follows Q
- then S has distribution $\mathbf{N}(a, a)$
- if x follows P
- then S has distribution close to $\mathbf{N}(-a, a)$
- $a = \mu C^{-1} \mu$

Critical Region

- Decide $\bar{K} = k$ correct if $S(\nu, k) > z$(threshold)
- Success probability

$$\beta = \mathbf{Pr}(S(k, \nu) > z | \bar{K} = k \,\text{correct})$$

- The number of $\bar{K}$-candidates to brute force $\alpha 2^{|\bar{K}|}$, where

$$\alpha = \mathbf{Pr}(S(k, \nu) > z | \bar{K} = k \,\text{incorrect})$$

- We need an algorithm to construct $\bar{K}$-candidates

Constructing $\bar{K}$-candidates

- $\bar{K}_i$ has $2^{|\bar{K}_i|}$ values k_i, keep their weights $S_i(k_i, \nu_i)$
- combine k_i s.t.
  1. $\sum_i S_i(k_i, \nu_i) > z$
  2. $\begin{cases} \bar{K}_i & = k_i \\ i & = 1, .., m \end{cases}$ is consistent.
  3. Solution is a $\bar{K}$-candidate
- by walking over a search tree

Precomputation

- Space generated by linear functions $\bar{K}_i$

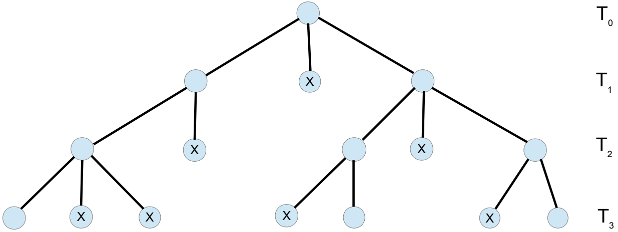
$$\langle \bar{K} \rangle = \langle \bar{K}_1, \ldots, \bar{K}_m \rangle$$

- Precompute sequence of subspaces

$$0 = \langle T_0 \rangle \subset \langle T_1 \rangle \subset \langle T_2 \rangle \subset \ldots \subset \langle T_p \rangle = \langle \bar{K} \rangle.$$

- For each i, j
- precompute function $d_{ji}(B) = \max_{\{k_i | T_j = B\}} S_i(k_i)$
- d_{ji} has $2^{\dim(<T_j> \cap <\bar{K}_i>)}$ values, may be kept
- search tree algorithm below

Search Tree



- $0 = \langle T_0 \rangle \subset \langle T_1 \rangle \subset \langle T_2 \rangle \subset \langle T_3 \rangle = \langle \bar{K}_1, .., \bar{K}_m \rangle$
- Continue a branch from level j, where $T_j = B$, to level $j+1$ if

$$\sum_{i=1}^{m} d_{ji}(B) > z$$

- Otherwise cut and backtrack
- Tree complexity is the number of nodes

Formal Algorithm

- Start with $j = 1$, recursive step:
- value of $T_{j-1} \subset T_j$ determined, find a value for T_j
- Take any T_j-value B that extends the value of T_{j-1}
- For each i look up $d_{ji}(B)$
- Check $\sum_{i=1}^{m} d_{ji}(B) > z$, if yes
- and $j < p$, then $j \leftarrow j + 1$ and repeat,
- If $j = p$, then as $\langle T_p \rangle = \langle \bar{K} \rangle$, a $\bar{K}$-candidate is found.
- Otherwise, take another value for T_j or backtrack

Justification and Success Probability

- Obviously,
- $\sum_{i=1}^{m} S_i(k_i, \nu_i) > z$, where $\bar{K}_i = k_i, i = 1, .., m$ are consistent,
- implies $\sum_{i=1}^{m} d_{ji}(B) > z$ for every j and B(value of T_j)
- We won't miss the correct key-value of $\bar{K}$,
- Success probability is still β computed earlier

Complexity

- The number of $\bar{K}$-candidates is $\alpha 2^{|\bar{K}|}$
- the number of cipher-keys to brute force

$$(\alpha 2^{|\bar{K}|}) \times 2^{\text{keysize}-|\bar{K}|} = \alpha 2^{\text{keysize}}$$

- The number of nodes in the search tree,
- experimentally for DES, is comparable with $\alpha 2^{\text{keysize}}$
- Constructing one node is easy:
- few XORs and additions of low precision real numbers

Back to 16-round DES

- By DES symmetry we can use two 14-bit vectors:

$$X_2[24, 18, 7, 29], X_{15}[16, 15, .., 11], X_{16}[24, 18, 7, 29]$$
$$X_1[24, 18, 7, 29], X_2[16, 15, .., 11], X_{15}[24, 18, 7, 29]$$

- considered independent as they incorporate different bits
- 14 dependent 10-bit projections from each, 28 in all
- two separable statistics independently distributed are used

How it Looks for One Projection

- projection h_1:

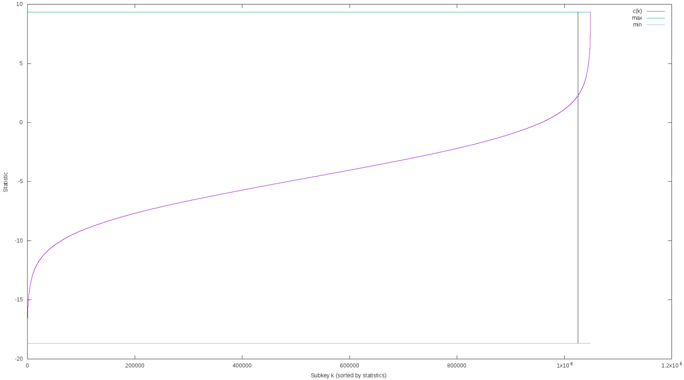
$$X_2[24, 18, 7, 29], X_{15}[16, 15], X_{16}[24, 18, 7, 29]$$

- $\bar{K}_1$ incorporates 20 unknowns

$$x_{63}, x_{61}, x_{60}, x_{53}, x_{46}, x_{42}, x_{39}, x_{36}, x_{31},$$
$$x_{30}, x_{27}, x_{26}, x_{25}, x_{22}, x_{21}, x_{12}, x_{10}, x_7, x_5,$$
$$x_{57} + x_{51} + x_{50} + x_{19} + x_{18} + x_{15} + x_{14}$$

  x_i key-bits of 56-bit DES key
- For each value $\bar{K}_1 = k_1$ the value of $S_1(k_1)$ is kept
- 2^{20} values

LLR_1-values for h_1



- $n = 2^{41.8}$, expected LLR_1 for correct $\bar{K}_1 = k_1$ is 4.6649, for incorrect -4.6638
- Experimental value for correct key 2.2668
- 23370 values higher than that
- Similar picture for other 27 projections h_i

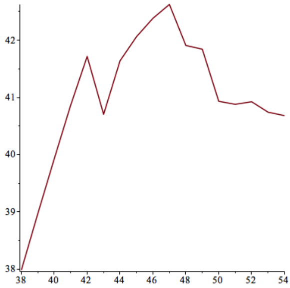
Constructing Search Tree

- T_j-sequence:
- $T_1 =< x_2 >, T_2 =< x_2, x_{19} >, T_3 =< x_2, x_{19}, x_{60} >, ..$
- x_2 appears in 14(maximal number) of $\bar{K}_i$, etc

$$x_2, x_{19}, x_{60}, x_{34}, x_{10}, x_{17}, x_{59}, x_{36}, x_{42}, x_{27}, x_{25},$$
$$x_{52}, x_{11}, x_{33}, x_{51}, x_9, x_{23}, x_{28}, x_5, x_{55}, x_{46}, x_{22},$$
$$x_{62}, x_{15}, x_{37}, x_{47}, x_7, x_{54}, x_{39}, x_{31}, x_{29}, x_{20}, x_{61},$$
$$x_{63}, x_{30}, x_{38}, x_{26}, x_{50}, x_1, x_{57}, x_{18}, x_{14}, x_{35}, x_{44},$$
$$x_3, x_{21}, x_{41}, x_{13}, x_4, x_{45}, x_{53}, x_6, x_{12}, x_{43}$$

Search Tree Complexity

- plain-texts $n = 2^{41.8}$, success rate 0.85



- in fig. examined values of T_j(tree nodes), $j = 38, ..54$, $\log_2$ scale
- $\#$ $\bar{K}$-candidates is $2^{39.8}$, $\#$ key to brute force $n = 2^{41.8}$
- overall number of nodes is $2^{45.5} << 2^{54}$. Constructing the nodes is faster(at least in bit operations) than brute force
- improvement over Matsui's result on DES($n = 2^{43}$, 0.85)

Possible Improvements

- Use another statistics for projections h_i. Let $\bar{K}_{0i} \subset \bar{K}_i$
- e.g., key-bits $\bar{K}_{0i}$ affect the distribution, then

$$LLR_i^*(\bar{K}_i \setminus \bar{K}_{0i}) = \max_{K_{0i}} LLR_i(\bar{K}_i)$$

- In practice better, in line with Matsui's analysis
- However the distribution of

$$(LLR_1^*, \ldots, LLR_m^*)$$

  is not well understood. Success probability is difficult to predict
- Experimentally for a truncated cipher and extrapolate?

Conclusions

- A method of computing joint distribution of encryption internal bites X, Y is presented
- We have realised that Multivariate Linear Analysis and its variations are inefficient for large X, Y. A solution to this problem is suggested
- based on a new statistic which reflects round function structure and a new search algorithm to find key-candidates which fall into critical region
- The method was applied to DES, gave an improvement over Matsui's results
- We were able to predict correctly success probability(8-round DES) and the number of final key-candidates(16-round DES)
- Complexity of the search algorithm is 10^3 times faster than brute force over all sub-keys which affect the statistic