Separable Statistics and Multivariate Linear Cryptanalysis

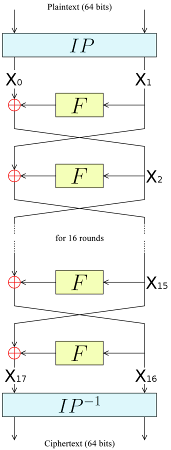
Stian Fauskanger[1] Igor Semaev[2]

Norwegian Defence Research Establishment (FFI), PB 25, 2027 Kjeller, Norway

Department of Informatics, University of Bergen, Bergen, Norway

Boolean Functions and their Applications (BFA), July, 2017

Vector of Internal Bits from Cipher



We define

$$A = (X_{16}[24, 18, 7, 29], X_{15}[16, 15, 14, 13, 12, 11], X_2[24, 18, 7, 29]).$$

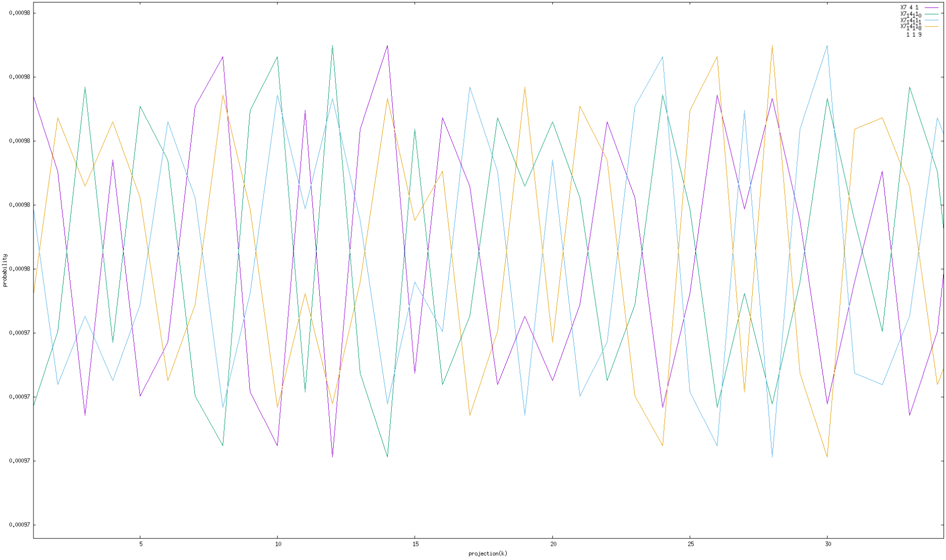
The probability distribution of A depends on somme 7-bit $\tilde{k}$. We know (approximately) the probability distribution of A:

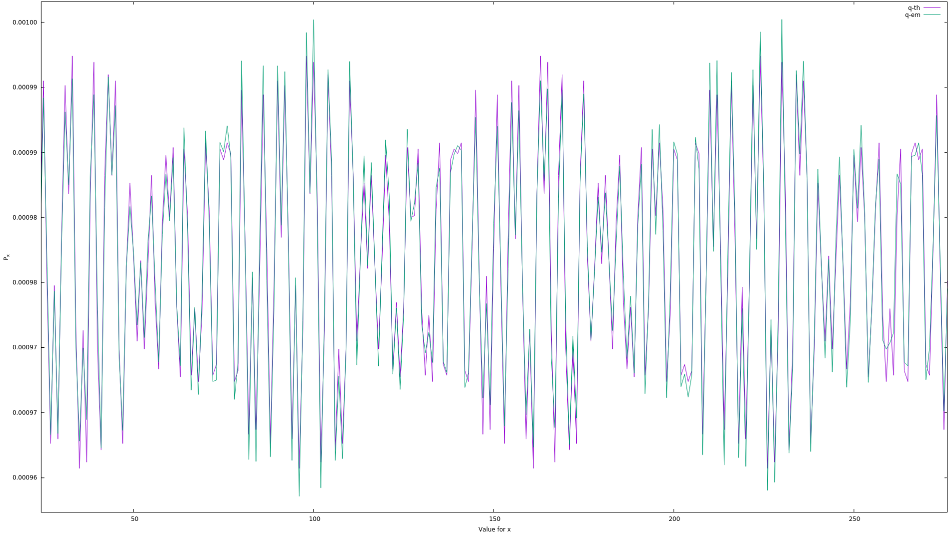
$$p(k) = (p_0, \dots, p_{2^{14}-1}),$$

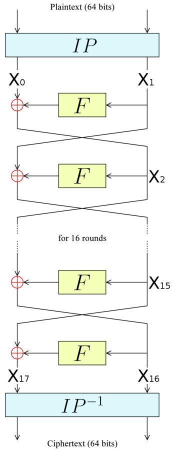
where

$$p_i = \mathbf{Pr}\left(A = i \;\middle|\; \tilde{k} = k\right).$$

Original image src (without variable names): wikimedia.org


0,00098
0,00098
0,00098
0,00098
0,00098
0,00097
0,00097
0,00097
0,00097
5
10
15
20
25
30
projection(k)
probability
X7 4 1
X714110
X714111
X714118
1 1 9

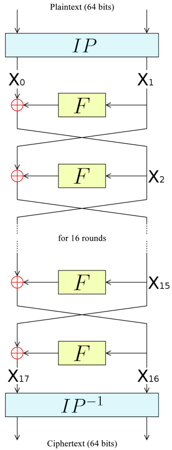

q-th
q-em
0.00100
0.00099
0.00099
0.00098
0.00098
0.00097
0.00097
0.00096
P x
50
100
150
200
250
Value for x



$$A = (X_{16}[24, 18, 7, 29], X_{15}[16, 15, 14, 13, 12, 11], X_2[24, 18, 7, 29]).$$

We want to use A in a known plaintext attack on DES but X_2 and X_{15} is not part of the plaintext or ciphertext. We can, however, compute the relevant bits in X_2 and X_{15} from X_0, X_1, X_{16}, X_{17} and some 42-bit $\bar{k}$.

Original image src (without variable names): wikimedia.org



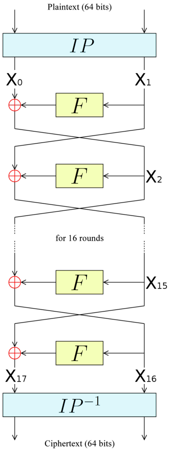
$$A = (X_{16}[24, 18, 7, 29], X_{15}[16, 15, 14, 13, 12, 11], X_2[24, 18, 7, 29]).$$

We want to use A in a known plaintext attack on DES but X_2 and X_{15} is not part of the plaintext or ciphertext. We can, however, compute the relevant bits in X_2 and X_{15} from X_0, X_1, X_{16}, X_{17} and some 42-bit $\bar{k}$.

Problem

$k \cup \bar{k} = 45$. We want time and data complexity to be $< 2^{43}$. Using the above vector in multivariate linear cryptanalysis [Hermelin et al.] would require that we rank 2^{45} key-candidates.

Original image src (without variable names): wikimedia.org



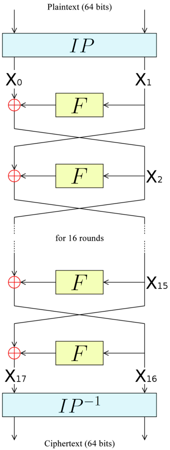
Instead of using A, we use 10-bit projections of A:

$$
\begin{aligned}
A^{(j)} &= (X_{16}[24, 18, 7, 29], X_{15}[a_j, b_j], X_2[24, 18, 7, 29]), \\
a_j, b_j &\in \{16, 15, 14, 13, 12, 11\}, \\
a_j &> b_j, \\
(a_j, b_j) &\neq (16, 11).
\end{aligned}
$$

There are 14 projections, $A^{(1)}, \ldots, A^{(14)}$. The probability distribution of $A^{(j)}$ can be computed from the probability distribution of A, and depends on some 2- or 3-bit $\tilde{k}^{(j)}$.

Original image src (without variable names): wikimedia.org

Computing $A^{(j)}$ from Observation



$$A^{(j)} = (X_{16}[24, 18, 7, 29], X_{15}[a_j, b_j], X_2[24, 18, 7, 29]).$$

Like before, we want to use $A^{(j)}$ in a known plaintext attack but X_2 and X_{15} is not part of the plaintext or ciphertext. We can, however, compute the relevant bits in X_2 and X_{15} from X_0, X_1, X_{16}, X_{17} and some 18-bit $\bar{k}^{(j)}$.

In total $A^{(j)}$ depends on 18-21 key-bits, denoted by $\mathtt{K}^{(j)} = \bar{k}^{(j)} \cup \tilde{k}^{(j)}$. 18 key-bits are needed to compute $A^{(j)}$ from a plaintext-ciphertext pair, and the distribution of $A^{(j)}$ depends on 2-3, possibly overlapping, key-bits.

Original image src (without variable names): wikimedia.org

Random Vectors Based on Plaintext-Ciphertext Pairs

We observe n plaintext/ciphertext pairs all encrypted using the same key. We run over all plaintext-ciphertext pairs and compute the number of occurrences for each possible value of $A^{(j)}$ for all $\bar{k}^{(j)}$. We define a random vector (observation vector) for each $\bar{k}^{(j)}$

$$V^{(j)}(k) = (v_0^{(j)}, ..., v_{2^{10}-1}^{(j)}),$$

where $v_i^{(j)}$ is the number of times $A^{(j)} = i$ assuming $\bar{k}^{(j)} = k$.

$$V^{(j)}(k) = (v_0^{(j)}, \dots, v_{2^{10}-1}^{(j)})$$

is a random vector that follows multinomial distribution with n samples and some vector of probabilities, q. We have that:

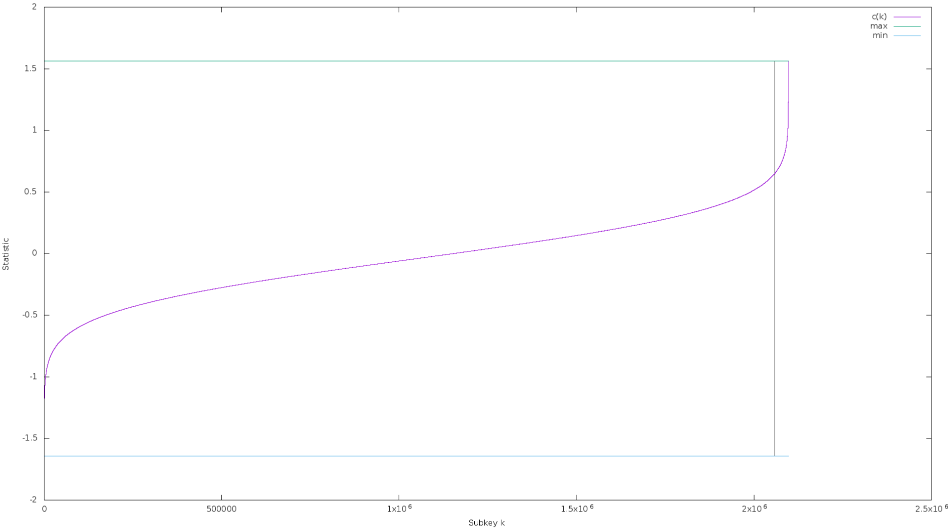
	guess of $\mathtt{K}^{(j)}$ correct	guess of $\mathtt{K}^{(j)}$ incorrect
$q =$	$p^{(j)}$	$(2^{-10}, \dots, 2^{-10})$
$E[v_i^{(j)}] =$	$n \times p_i^{(j)}$	$n \times 2^{-10}$
$Var[v_i^{(j)}] =$	$n \times p_i^{(j)} \times (1 - p_i^{(j)})$	$n \times 2^{-10} \times (1 - 2^{-10})$
$Cov[v_i^{(j)}, v_j^{(j)}] =$	$n \times p_i^{(j)} \times p_j^{(j)}$	$n \times 2^{-20}$

Separable Statistics

We compute the statistic $c^{(j)}(\mathtt{K}^{(j)})$ for all possible realisations of $\mathtt{K}^{(j)}$ and for all j. $c^{(j)}(\mathtt{K}^{(j)})$ is the log-likelihood-ratio of a correct guess of $\mathtt{K}^{(j)}$, over an incorrect guess of of $\mathtt{K}^{(j)}$.

$$c^{(j)}(\mathtt{K}^{(j)}) = log_2\left(\prod_i \left(\frac{p_i^{(j)}}{2^{-10}}\right)^{v_i^{(j)}}\right) = \sum_i v_i^{(j)} \times (log_2(p_i^{(j)}) + 10).$$

There are $< 14 \times 2^{21}$ possible realisations of $\mathtt{K}^{(j)}$ in total. Computing $c^{(j)}(\mathtt{K}^{(j)})$ for all of them can be done efficiently using fast Walsh-Hadamard Transform. The complexity is $O(2^{37})$ operations using $O(2^{28})$ memory.



c(k)
max
min
Statistic
Subkey k
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
500000
1x10 6
1.5x10 6
2x10 6
2.5x10 6

Symmetry in DES Cipher

Because of symmetry in DES it's trivial to duplicate all previous work using both A and A', which we assume are statistically independent.

$$A = (X_{16}[24, 18, 7, 29], X_{15}[16, 15, 14, 13, 12, 11], X_2[24, 18, 7, 29]) \ ,$$
$$A' = (X_1[24, 18, 7, 29], X_2[16, 15, 14, 13, 12, 11], X_{15}[24, 18, 7, 29]) \ .$$

We use 14 10-bit projections from each of them. $A^{(1)}, \ldots, A^{(14)}$ are projections of A and $A^{(15)}, \ldots, A^{(28)}$ are projections of A'. We now have 28 sub-keys, $\mathtt{K}^{(1)}, \ldots, \mathtt{K}^{(28)}$, and a statistic associated to each possible key value. That is, we have $< 28 \times 2^{21}$ different $c^{(j)}(\mathtt{K}^{(j)})$.

Let $\mathtt{K}$ be a 54-bit sub-key of the 56-bit key in DES. $\mathtt{K}$ is the union of $\mathtt{K}^{(1)}, \ldots, \mathtt{K}^{(28)}$. We want to use the previous statistics to find a good key candidate for $\mathtt{K}$. We define two separable statistics

$$C(\mathtt{K}) = \sum_{j=1}^{14} w_j \times c^{(j)}(\mathtt{K}^{(j)}) \qquad \text{and} \qquad C'(\mathtt{K}) = \sum_{j=15}^{28} w_j \times c^{(j)}(\mathtt{K}^{(j)}).$$

We built a search tree from the statistics $c^{(j)}(\mathtt{K}^{(j)})$ and designed an algorithm that goes through the tree to find 54-bit key candidates, $\mathtt{K}$. A key candidate is accepted if $C(\mathtt{K}) > z$ and $C'(\mathtt{K}) > z$ simultaneous, for some optimal weights w_j and a parameter z. The remaining 2 key-bits are brute forced for each key candidate.

Complexity and Probability of Success

The complexity of our attack is measured by n (number of plaintext-ciphertext pairs), the number of nodes visited while traversing the search tree and the number of encryptions to brute force the remaining 2 key-bits for all candidates.

$C(\mathtt{K})$ and $C'(\mathtt{K})$ are normally distributed. We choose z so that $n/4$ candidates for $\mathtt{K}$ are accepted. n encryptions is then performed.

The probability that our attack is successfull is the probability that $C(\mathtt{K}) > z$ and $C'(\mathtt{K}) > z$ for correct $\mathtt{K}$.

In particular, we set $n = 2^{41.8}$ and z so that the expected number of accepted candidates is $2^{39.8}$. Running the full attack returned $2^{39.46}$ candidates while visiting $2^{45.78}$ nodes in the search tree. Visiting one node is a simpler operation than one DES encryption, so the total time and data complexity is about $2^{41.8}$ encryptions. We are working on reducing the number of nodes visited.

Thanks

Questions?