Advanced Cryptography: Lightweight Cryptography

MICHAEL NÜSKEN, DANIEL LOEBENBERGER

5. Exercise sheet
Hand in solutions until Monday, 28 November 2011, 23:59:59h.

If you find any errors in the sheets, do not hesitate to write an email to the mailing list `11ws-ac@lists.bit.uni-bonn.de`.

Exercise 5.1 (From the lecture). (5 points)

Show that for any of the DES S-boxes S_i, we have due to Coppersmith' criterion S-3 that for all β we have [5]

$$\begin{aligned}\mathrm{bias}_{S_i}(000001, \beta) &= 0,\\ \mathrm{bias}_{S_i}(100000, \beta) &= 0,\\ \mathrm{bias}_{S_i}(100001, \beta) &= 0.\end{aligned}$$

Recall that $2^6 \,\mathrm{bias}_S(\alpha, \beta) = \sum_{x \in \{0,1\}^6} (-1)^{\langle \beta \,|\, S(x)\rangle + \langle \alpha \,|\, x\rangle}$.

Exercise 5.2 (Partially defined S-boxes). (25+25 points)

In this exercise you are yet again going to explore the joy of constructing S-boxes. For the construction you will iteratively extend a partially defined S-box

$$S\colon \{0,1\}^6 \longrightarrow \{0,1\}^4 \uplus \{\perp\}$$

until you obtain one that is defined for all possible input values. We use the symbol $\perp$ to mark undefined values. Assume you have an input value $x_0 \in \{0,1\}^6$ for which the output of S is undefined, i.e. $S(x_0) = \perp$ and an output value $y_0 \in \{0,1\}^4$. Define the extended S-box as

$$S'\colon \begin{array}{ccc} \{0,1\}^6 & \longrightarrow & \{0,1\}^4 \cup \{\perp\}, \\ x & \longmapsto & \begin{cases} S(x) & x \neq x_0 \\ y_0 & x = x_0 \end{cases}. \end{array}$$

As a notation for this you may use: $S + (x_0 \mapsto y_0) := S'$. We define the differential table diff_S for such a partially defined S-box by

$$\mathrm{diff}_S\colon \begin{array}{ccc} \{0,1\}^6 \times \{0,1\}^4 & \longrightarrow & \mathbb{N}, \\ (\Delta x, \Delta y) & \longmapsto & \#\left\{ x \in \{0,1\}^6 \;\middle|\; S(x) \oplus S(x \oplus \Delta x) = \Delta y \right\} \end{array}$$

where we extend $\oplus$ by $y \oplus \perp := \perp$, $\perp \oplus y := \perp$, $\perp \oplus \perp := \perp$ to handle the undefined positions.

6 (i) Assume that already you have the differential table diff_S for the partially defined S-box S. Find a simple representation for the differential table $\mathrm{diff}_{S'}$ in terms of the values of the differential table diff_S.

Prove as a corollary that $\mathrm{diff}_{S'}(\Delta x, \Delta y) \geq \mathrm{diff}_S(\Delta x, \Delta y)$ for all Δx, Δy.

Hint: Consider the difference $\Delta\mathrm{diff}_S^{x_0 \mapsto y_0} := \mathrm{diff}_{S'} - \mathrm{diff}_S$ of both tables.

4 (ii) Assume a table maxdiff of bounds for the final differential table is given, that is, we require $\mathrm{diff}_S(\Delta x, \Delta y) \leq \mathrm{maxdiff}(\Delta x, \Delta y)$. We wish to find for all x_0 the set $\mathrm{options}_S(x_0)$ of all possible options y_0 for which the extended S-box S' is still allowed, that is, its differential table $\mathrm{diff}_{S'}$ is still component-wise less than or equal to maxdiff.

Prove that $\mathrm{options}_S$ only depends on $\mathrm{maxdiff} - \mathrm{diff}_S$, which describes the remaining margin for differences.

Use the results of (i) to find a method that is able to compute one such set in 2^6 operations (rather than 2^{10} with the trivial criterion!).

+5 (iii) Assume you have the table of all possible options for extending S-box S. Try to find a fast algorithm to update the options (rather than recomputing them), that is, compute the table of options for S-box S' based on the data available, which are $\mathrm{maxdiff} - \mathrm{diff}_S$, $\Delta\mathrm{diff}_S^{x_0 \mapsto y_0}$ and $\mathrm{options}_S$.

5 (iv) In the lecture we listed some properties that your final S-box should have: Coppersmith' S-3, S-4, S-5, S-7 and Poschmann's condition C-1. Produce the table $\mathrm{maxdiff}(\Delta x, \Delta y)$ that is implied by those conditions.

Hint: an algorithmic description will be helpful for the next item.

10 (v) Try to generate an S-box fulfilling Coppersmith' S-3, S-4, S-5, S-7 and Poschmann's C1 in the iterative way described above starting from a totally undefined S-box. Experiment with different strategies to select possible options for extension of your current intermediate box.

You do not necessarily have to use the result from (iii), but it speeds up things.

+5 (vi) Compute the linear table bias_S for the final S-box you found.

+15 (vii) So far we have derived a strategy to deal with diff_S in this exercise.

Derive a similar strategy to compute bias_S iteratively so that we can update with every newly chosen value. This shall enable us to check conditions on this table simultaneously.

Assume you are given a table maxbias of maximally allowed absolute values for the biases, that is, we want $|\mathrm{bias}_S(\alpha, \beta)| \leq \mathrm{maxbias}(\alpha, \beta)$ for the final S-Box. ~~Adapt the definition of $\mathrm{options}_S$ to cover also these conditions and again find a fast algorithm to update options.~~ Use a suitable penalty (a function whose values increase when $|\mathrm{bias}_S(\alpha, \beta)| - \mathrm{maxbias}_S(\alpha, \beta)$ gets larger) to decide in which order to consider alternatives.

Use Poschmann's condition C2 ($\mathrm{bias}_S(\alpha, \beta) \leq 28$ for all α, β) and C3 ($\mathrm{bias}_S(\alpha, \beta) \leq 4$ for all α, β of Hamming weight 1) to define maxbias.

Generate an S-box within these differential and bias bounds.