Technische Universität München (I7)
Dr. M. Luttenberger / M. Schlund

Winter term 2014/2015
Tuesday 25th November, 2014

Solution

Cryptography – Homework 3

Discussed on Tuesday, 18th of November, 2014.

For questions regarding the exercises, please send an email to schlund@model.in.tum.de or just drop by at room 03.11.055

Exercise 3.1 **Pseudorandom functions!**

(a) Show that PRFs with $l_{\text{out}}(n) \cdot 2^{l_{\text{in}}(n)} \leq n$ exist (unconditionally!).

(b) Let G be a PRG of stretch $l_G(n) = 2n$.

Split $G(k) =: G_0(k)||G_1(k)$ into two n bit strings.

i) Set $F_k^{(1)}(0) := G_0(k)$ and $F_k^{(1)}(1) := G_1(k)$. Show: $F^{(1)}$ is a PRF with $l_{\text{in}}(n) = 1$ and $l_{\text{out}}(n) = n$.

ii) (*) Set $F_k^{(2)}(x_1x_2) := G_{x_2}(F_k^{(1)}(x_1))$ for $x_1x_2 \in \{0,1\}^2$. Show: $F^{(2)}$ is a PRF with $l_{\text{in}}(n) = 2$ and $l_{\text{out}}(n) = n$.

Solution:

(a) Since $l_{\text{out}}(n) \cdot 2^{l_{\text{in}}(n)} \leq n$ we can decompose any $k \in \{0,1\}^n$ into (at least) $N = 2^{l_{\text{in}}}$ parts of size l_{out}: $k = k_{\lfloor 0 \rceil}|| \cdots ||k_{\lfloor N \rceil}||rest$. Set $F_k(x) = k_x$. Then F_k is clearly indistinguishable from a random function and thus a PRF.

(b) i) If F could be distinguished from a RO by some $\mathcal{A}$, then G would not be a PRG. Consider the following distinguisher $\mathcal{D}$: On input $y = y_0y_1$ with $y_i \in \{0,1\}^n$ we would set up an oracle $\mathcal{O}$, which on input $x \in \{0,1\}$ returns y_x. We run $\mathcal{A}$ on $\mathcal{O}$ which gives us an answer r. We simply return r as our guess for b. It is easy to see that $\mathcal{D}$ has the same advantage as $\mathcal{A}$. Thus $F_k^{(1)}$ is a PRF.

ii) The core observation is, that for $k = k_0k_1$ with $k_0, k_1 \xleftarrow{u} \{0,1\}^n$ independently, $G_0(k_0)||G_1(k_0)||G_0(k_1)||G_1(k_1)$ is indistinguishable from *both* $y \xleftarrow{u} \{0,1\}^{4n}$ and also $G_0(G_0(k))||G_1(G_0(k))||G_0(G_1(k))||G_1(G_1(k))$ for $k \xleftarrow{u} \{0,1\}^n$. Thus we can prove (in two steps) that $G_0(k_0)||G_1(k_0)||G_0(k_1)||G_1(k_1)$ is a PRG of stretch $2n$ and

$G_0(G_0(k))||G_1(G_0(k))||G_0(G_1(k))||G_1(G_1(k))$ is a PRG of stretch $4n$. From this it follows easily that $F^{(2)}$ is a PRF.

This is an instance of a "hybrid"-argument: We take an "intermediate" pseudorandom-sequence which we compare to both, a truly random string and to the image of $F^{(2)}$.

Gory details (Of the first step)

First, we show that a PRG for two independent seeds, $G'(k_0||k_1) := G(k_0)||G(k_1)$ is again a PRG of stretch $2n$, given a PRG G of stretch $2n$.

We describe how to build a possible distinguisher $\mathcal{D}$ for G, if we had some distinguisher $\mathcal{D}'$ for G':

i. Input: $y \in \{0,1\}^{2n}$

ii. Generate a random $b' \xleftarrow{u} \{0,1\}$

A. If $b' = 0$: generate $y' \xleftarrow{u} \{0,1\}^{2n}$ and give yy' to A.

B. If $b' = 1$: generate $x' \xleftarrow{u} \{0,1\}^n$, set $y' := G(x')$ and give $y'y$ to $\mathcal{D}'$ (note that we change the order here!).

iii. Get back r' from A and output it.

There are now four cases: $[b = 0, b' = 0]$, $[b = 0, b' = 1]$, $[b = 1, b' = 0]$, $[b = 1, b' = 1]$.

$b = 0, b' = 0$: $\Pr_{b=0,b'=0}\left[\mathsf{Win}_{n,G}^{\text{INDPRG}}(\mathcal{D})\right] = \Pr_{b=0,b'=0}[\mathcal{D}(yy') = 0] = \Pr_{b=0,b'=0}[\mathcal{D}'(yy') = 0] = \Pr_{b=0,b'=0}\left[\mathsf{Win}_{n,G}^{\text{INDPRG}}(\mathcal{D}')\right]$

$b = 0, b' = 1$: $\Pr_{b=0,b'=1}\left[\mathsf{Win}_{n,G}^{\text{INDPRG}}(\mathcal{D})\right] = \Pr_{b=0,b'=1}[\mathcal{D}(G(x')y) = 0] = \Pr_{b=0,b'=1}[\mathcal{D}'(G(x')y) = 0] =: q$

$b=1, b'=0$: $\Pr_{b=1,b'=0}\left[\mathsf{Win}^{\text{INDPRG}}_{n,G}(\mathcal{D})\right] = \Pr_{b=1,b'=0}[\mathcal{D}(G(x)y') = 1] = \Pr_{b=1,b'=0}[\mathcal{D}'(G(x)y') = 1] = 1-q$

$b=0, b'=0$: $\Pr_{b=1,b'=1}\left[\mathsf{Win}^{\text{INDPRG}}_{n,G}(\mathcal{D})\right] = \Pr_{b=1,b'=1}[\mathcal{D}(G(x')G(x)) = 0] =$

$= \Pr_{b=1,b'=1}[\mathcal{D}'(G(x')G(x)) = 1] = \Pr_{b=1,b'=1}\left[\mathsf{Win}^{\text{INDPRG}}_{n,G}(\mathcal{D}')\right]$

Together:

$$\Pr\left[\mathsf{Win}^{\text{INDPRG}}_{n,G}(\mathcal{D})\right] = 1/4\Pr\left[\mathsf{Win}^{\text{INDPRG}}_{n,G}(\mathcal{D}')\right] + 1/4q + 1/4(1-q) + 1/4\Pr\left[\mathsf{Win}^{\text{INDPRG}}_{n,G}(\mathcal{D}')\right] = 1/2\Pr\left[\mathsf{Win}^{\text{INDPRG}}_{n,G}(\mathcal{D}')\right] + 1/4$$

Now

$$\left|\Pr\left[\mathsf{Win}^{\text{INDPRG}}_{n,G}(\mathcal{D})\right] - 1/2\right| = \left|1/2\Pr\left[\mathsf{Win}^{\text{INDPRG}}_{n,G}(\mathcal{D}')\right] + 1/4 - 1/2\right| = 1/2\left|\Pr\left[\mathsf{Win}^{\text{INDPRG}}_{n,G}(\mathcal{D}')\right] - 1/2\right|,$$

thus the advantage of $\mathcal{D}'$ is twice the advantage of $\mathcal{D}$ (and therefore still negligible).

(For the general construction the reasoning is much more complicated since now there is an exponential number of pseudorandom parts — so this simple approach does not work)

Exercise 3.2 **rCBC and the BEAST-Attack (on CBC with chained IVs)**

Let F be a PRP.

(a) Show that F-rCBC is not CCA-secure.

(b) Show that F-CBC-CIV (with chained IV—see lecture slides) is *not* CPA-secure.

Solution:

(a) A possible CCA-attack:

- Generate two messages $m_0 = 0^n$ and $m_1 = 1^n$ and send them to Alice/Bob.
- Receive a ciphertext $c = c^{(0)}||c^{(1)}$ (where $c^{(0)} = IV$) and compute $\tilde{c} = \overline{c^{(0)}}||c^{(1)}$ (just negate the first n bits).
- Since $\tilde{c} \neq c$ we are allowed to query the decryption oracle Dec_k on $\tilde{c}$ and this yields $\mathsf{Dec}_k(\tilde{c}) = m_b \oplus IV \oplus \overline{IV} = m_b \oplus 1^n = \overline{m_b}$. Hence, $\overline{\mathsf{Dec}_k(\tilde{c})} = m_b$.
- If $m_b == 1^n$ output "1" else output "0".

This attack is able to recover m_b and hence succeeds with probability 1.

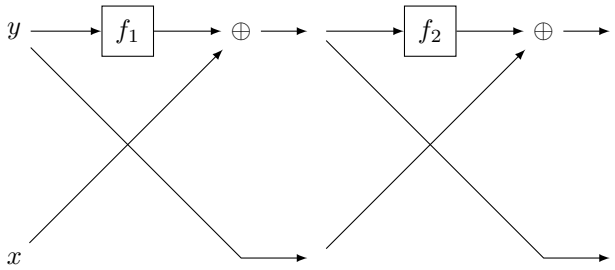
(b) We describe a CPA-attack:

- Before crafting our two messages we "set up" the Enc_k Algorithm with two queries (i.e. we force it into some state where we can predict the next IV it will use!)
- First query Enc_k on 0^n and get back $\mathsf{Enc}_k(0^n) = IV||\underbrace{F_k(IV)}_{=:IV'}$
- Then query Enc_k on IV' and receive $\mathsf{Enc}_k(IV') = IV'||\underbrace{F_k(0^n)}_{=:IV''}$
- Now we build our two messages: $m_0 = IV''$ and $m_1 = \overline{IV''}$ and send them to Alice& Bob.
- From Alice& Bob we get back a ciphertext $c = c^{(0)}||c^{(1)}$. If c is of the form $c = x||x$ we output 0 else we output 1.

Analysis:

- If $b = 0$ (i.e. $m_0 = IV''$ was chosen to be encrypted), then $c = IV''||F_k(IV'' \oplus IV'') = IV''||IV''$, so in this case we output 0 correctly with probability 1!
- If $b = 1$ (i.e. $m_0 = \overline{IV''}$ was chosen to be encrypted), then $c = IV''||F_k(IV'' \oplus \overline{IV''}) = IV''||F_k(1^n)$. Hence, we output 0 incorrectly (i.e. we loose) only if $IV'' = F_k(1^n)$, i.e. $F_k(0^n) = F_k(1^n)$ which is impossible, since F_k is a PRP and in particular injective! So also in this case our distinguisher wins with probability 1!

Exercise 3.3 **Feistel networks**



Let $f : \{0,1\}^* \to \{0,1\}^*$ be some function s.t. $|f(x)| = |x|$ for all $x \in \{0,1\}^*$. A *single-round Feistel network* FN_f is defined by

$$\mathsf{FN}_f(x||y) := y||x \oplus f(y) \text{ for all } x, y \in \{0,1\}^* \text{ with } |x| = |y| .$$

Similarly, given functions $f_1, \ldots, f_j$ a *j-round Feistel network* is inductively defined by

$$\mathsf{FN}_{f_1,f_2,\ldots,f_j}(x||y) := \mathsf{FN}_{f_j}(\mathsf{FN}_{f_1,f_2\ldots,f_{j-1}}(x||y))$$

(a) Show that independent of the choice of $f_1, \ldots, f_j$ the function $\mathsf{FN}_{f_1,\ldots,f_j}$ is invertible if $f_1, \ldots, f_j$ are known.

From the exam in 2010:

(b) Let F be a PRF of key and block length n and $P_{k_1,k_2}(x||y) := \mathrm{FN}_{F_{k_1},F_{k_2}}(x||y)$ be a two-round Feistel network using F.

i) Compute $P_{k_1,k_2}(0^n||y)$ and $P_{k_1,k_2}(F_{k_1}(0^n) \oplus z||0^n)$.

ii) Show that PPT-Eve can compute $P^{-1}_{k_1,k_2}$ when given oracle access to P_{k_1,k_2}.

(c) Is $\mathrm{FN}_{F_{k_1},F_{k_2},F_{k_3}}$ with three independent keys $k_1, k_2, k_3 \overset{u}{\in} \{0,1\}^n$ a PRP? Is it a PRF? (y/n)

Solution:

(a) First we recall how we can invert a single-round Feistel network FN_f with $f \in \{f_1, f_2\}$. Since $\mathsf{FN}_f(x, y) := (y, x \oplus f(y))$, it holds that $\mathsf{FN}_f(a \oplus f(b), b) = (b, a \oplus f(b) \oplus f(b)) = (b, a)$. Informally speaking, we have to "swap" the inputs and the outputs of the Feistel network to get the inverse function, see Fig. 1. This can be generalized for two-round Feistel networks by plugging the inverses of the two single-round Feistel networks together in reverse order, (see the Figure). Note that we do not need the "double twist" between the two single-round networks, since they cancel each other out.

(And using induction we can generalize that to arbitrary rounds).

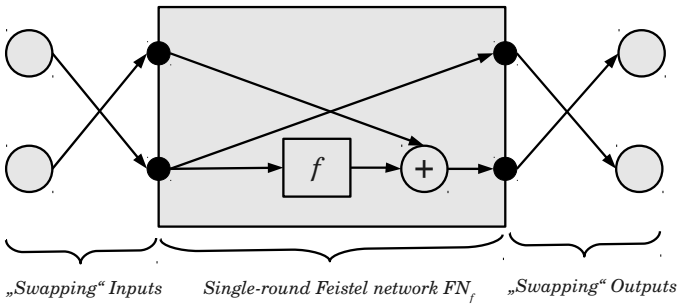


Figure 1: Inverting FN_f.

(b) See the slides for an illustration:

Result of first round: $(y, F_{k_1}(y) \oplus x)$.

Result of second round: $(F_{k_1}(y) \oplus x, F_{k_2}(F_{k_1}(y) \oplus x) \oplus y)$.

i) $P_{k_1,k_2}(0^n, y) = (F_{k_1}(y), F_{k_2}(F_{k_1}(y)) \oplus y)$.

$P_{k_1,k_2}(F_{k_1}(0^n) \oplus z, 0^n) = (z, F_{k_2}(z))$.

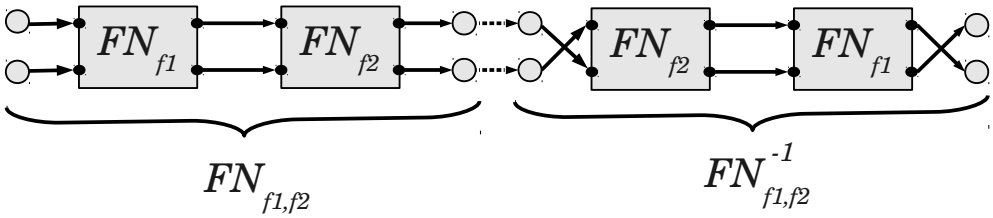


Figure 2: FN_{f_1,f_2} on the left side, $\mathsf{FN}^{-1}_{f_1,f_2}$ on the right. If we plug them together as in the figure, the resulting circuit realizes the identity map.

ii) By the preceding result, Eve can compute F_{k_1}, F_{k_2} by quering her oracle at most twice. Any Feistel network can be efficiently inverted if the round functions can be efficiently computed.

(Note that Eve is not given access to k so the important observation is that she can trick the oracle into supplying the required information.)

(c) i) Yes (see the result regarding FNs in the slides).

ii) Yes (see the result that any PRP is also a PRF).