Advanced Cryptography — Midterm Exam
Solution

Serge Vaudenay

4.5.2010

- all documents are allowed
- a pocket calculator is allowed
- communication devices are not allowed
- answers to the exercises must be provided on a separate sheet
- readability and style of writing will be part of the grade
- do not forget to put your name on your copy!

1 RC4 Biases

This exercise is partly inspired from Mantin-Shamir, A Practical Attack on Broadcast RC4, *published in the proceedings of FSE 2001, LNCS vol. 2355, Springer.*

The RC4 pseudorandom number generator is defined by a state and an algorithm which update the state and produces an output byte. In RC4, a state is defined by

- two indices i and j in $\mathbf{Z}_{256}$;
- one permutation S of $\mathbf{Z}_{256}$.

By abuse of notation we write $S(x)$ for an arbitrary integer x as for $S(x \bmod 256)$. The state update and output algorithm works as follows:

$i \leftarrow i + 1$
$j \leftarrow j + S(i)$
exchange the values at position $S(i)$ and $S(j)$ in table S
output $z_i = S(S(i) + S(j))$

Q.1 Assume that the initial S is a random permutation with uniform distribution and that i and j are set to 0.

Q.1a What is the probability that $[S(1) \neq 2 \text{ and } S(2) = 0]$?

It is $\frac{1}{N} \times \frac{N-2}{N-1}$ *with* $N = 256$.

Q.1b If $S(1) \neq 2$ and $S(2) = 0$ hold, show that the second output z_2 is always 0.

Let $S(1) = x$ *and* $S(x) = y$ *initially. At the first iteration,* i *is set to 1,* j *is set to* x*, and* $S(1)$ *and* $S(x)$ *are exchanged. There values become* y *and* x *respectively. Then,* i *is set to 2,* j *is set to* x *again, and* $S(2)$ *and* $S(x)$ *are exchanged. There values become* x *and* 0 *respectively. The output is* $S(x)$ *which is 0.*

Q.1c In other cases, show that $z_2 = 0$ with probability close to $\frac{1}{256}$.
Hint: a 2-line heuristic argument is fine for this question (and this question only).

Ideally, we would have to study many cases. In practice, we just assume that the number is truly random in that case so $\Pr[z_2 = 0] \approx \frac{1}{N} = \frac{1}{256}$.
(OK, the same argument could have applied in the previous case which would have contradicted the result. Nevertheless, $\frac{1}{256}$ *is a probability which is confirmed experimentally.)*

Q.1d Deduce $p = \Pr[z_2 = 0]$. What do you think of this probability?

Clearly, $p = \frac{1}{N} \times \frac{N-2}{N} + \frac{1}{N} \approx \frac{2}{N}$. *This is twice that what we should expect. This is a deviant property.*

Q.2 Let N be an integer. Let now consider a random generator which generate a byte Z such that $\Pr[Z = 0] = p \gg \frac{1}{N}$ and $\Pr[Z = x] = \frac{1-p}{N-1}$ for $0 < x < N$. In every question below, treat the general case then apply it to an example with $N = 256$ and $p = \frac{2}{N}$.

Q.2a Describe a best distinguisher between Z and an unbiased generator based on n samples? Show that there is one making the output 1 if and only if $\frac{k}{n} \geq \tau$ where k is the number of occurrences of $Z = 0$ in the n samples, and

$$\tau = \frac{1}{1 + \frac{\log(pN)}{\log \frac{1-\frac{1}{N}}{1-p}}}$$

is a threshold.
Treat the general case then compute τ in the example.

The best distinguisher takes the decision corresponding to the maximum likelihood probability of occurrence of the sample vector. Given a sample vector $z_1, \ldots, z_n$*, it computes the probability of getting* $z_1, \ldots, z_n$ *under the two distributions and takes the one with higher probability. Since samples are independent, this is the product of all* $\Pr[Z = z_i]$. *In the uniform case, this is* $\frac{1}{N^n}$. *In the biased case, this is* $p^k \left(\frac{1-p}{N-1}\right)^{n-k}$ *where* $k = \#\{i; z_i = 0\}$. *We have*

$$\begin{aligned}
p^k \left(\frac{1-p}{N-1}\right)^{n-k} \geq \frac{1}{N^n} &\iff (Np)^k \left(N\frac{1-p}{N-1}\right)^{n-k} \geq 1 \\
&\iff k\log(Np) + (n-k)\log\left(N\frac{1-p}{N-1}\right) \geq 0 \\
&\iff k\log\left(p\frac{N-1}{1-p}\right) \geq n\log\left(\frac{1}{N} \times \frac{N-1}{1-p}\right) \\
&\iff \frac{k}{n} \geq \frac{\log\frac{1-\frac{1}{N}}{1-p}}{\log\left(p\frac{N-1}{1-p}\right)} = \frac{1}{1 + \frac{\log(pN)}{\log\frac{1-\frac{1}{N}}{1-p}}}
\end{aligned}$$

which is of form $\frac{k}{n} > \tau$.
In the example, we have $N = 256$ *and* $p = \frac{1}{128}$ *so* $\tau = 0.00563680 \approx \frac{1}{177}$.

Q.2b Give a simpler formula to estimate τ when $\frac{1}{N} \ll 1$ and $p \ll 1$.
Treat the general case then compute the estimate in the example.

If p and $\frac{1}{N}$ are small we have $\ln\frac{1-\frac{1}{N}}{1-p} \approx p - \frac{1}{N}$ and $\ln(pN)$ is non-negligible. So,

$$\begin{aligned}\tau &= \frac{\log\frac{1-\frac{1}{N}}{1-p}}{\log\left(p\frac{N-1}{1-p}\right)} \\ &= \frac{1}{\frac{\ln(pN)}{\ln\frac{1-\frac{1}{N}}{1-p}}+1} \\ &\approx \frac{1}{\frac{\ln(pN)}{p-\frac{1}{N}}+1}\end{aligned}$$

If $pN-1$ is negligible, we have $\tau \approx \frac{1}{N}$. In general, $\ln(pN)$ is not negligible and we have $\tau \approx \frac{p-\frac{1}{N}}{\ln(pN)}$.
The example is in the latter case. We obtain $\tau \approx 0.00563552$ which is not a so bad approximation.

Q.2c What is the best advantage of a distinguisher when limited $n = 1$?
Treat the general case then compute the advantage in the example.

With $n = 1$, the output is 1 iff $k = 1$. So, the advantage is $\mathsf{Adv} = p - \frac{1}{N}$.
In our example, this is $\mathsf{Adv} = \frac{1}{256}$.

Q.2d What is the best advantage of a distinguisher when limited $n = 2$? (Assume that $\tau \leq \frac{1}{2}$.)
Treat the general case then compute the advantage in the example.
Hint: reduce to computing $\Pr[k = 0]$.

With $n = 2$, the output is 1 iff $k \geq 2\tau$. For τ smaller than $\frac{1}{2}$, this holds iff $k \geq 1$. So, $\mathsf{Adv} = (2p(1-p)+p^2) - \left(\frac{2}{N}(1-\frac{1}{N})+\frac{1}{N^2}\right)$. This simplifies to $\mathsf{Adv} \approx 2(p-\frac{1}{N})$.
In our example, this is $\mathsf{Adv} \approx \frac{1}{128}$.

Q.2e What is the best advantage of a distinguisher when limited $n = \lfloor\frac{1}{\tau}\rfloor$?
Treat the general case then compute the advantage in the example.

The output is 1 iff $k \neq 0$. So, $\mathsf{Adv} = (1-(1-p)^n) - \left(1-(1-\frac{1}{N})^n\right)$ with $n = \lfloor\frac{1}{\tau}\rfloor$.
This yields $\mathsf{Adv} = \left(1-\frac{1}{N}\right)^n - (1-p)^n$.
In our example, this is $\mathsf{Adv} \approx \frac{1}{4}$ with $n = 177$.

Q.2f Show that the Chernoff information between the two distributions is

$$C = -\tau\log_2\frac{p}{\tau} - (1-\tau)\log_2\frac{1-p}{1-\tau}$$

where $\alpha = \frac{1/N}{p}$, $\beta = \frac{1-1/N}{1-p}$.
Treat the general case then compute the Chernoff information in the example.
Hint: show that $C = -\log_2\min\left(p\alpha^\lambda + (1-p)\beta^\lambda\right)$ and that the minimum is reached when

$$\left(\frac{\alpha}{\beta}\right)^\lambda = \frac{1-p}{p} \times \frac{1}{\frac{1}{\tau}-1}$$

and deduce the optimal λ to compute C.

Hint[2]: if you are afraid of manipulating ugly formulae, consider skipping this question.

The Chernoff information is $-\log_2 \min_{\lambda\in]0,1[} f(\lambda)$ *where*

$$f(\lambda) = p^{1-\lambda}\frac{1}{N^\lambda} + (N-1)\left(\frac{1-p}{N-1}\right)^{1-\lambda}\frac{1}{N^\lambda}$$

This can be written $f(\lambda) = p\alpha^\lambda + (1-p)\beta^\lambda$ *where* $\alpha = \frac{1/N}{p}$ *and* $\beta = \frac{1-1/N}{1-p}$. *Since* f *is convex and* $f(0) = f(1) = 1$, f *reaches a single minimum which is in* $]0,1[$. *The minimum is reached when the derivative vanishes, which leads us to*

$$\left(\frac{\alpha}{\beta}\right)^\lambda = -\frac{1-p}{p}\times\frac{\ln\beta}{\ln\alpha} = \frac{1-p}{p}\times\frac{1}{\frac{1}{\tau}-1}$$

which yields $\lambda = \log(\frac{\tau}{1-\tau}\times\frac{1-p}{p})/\log(\frac{\alpha}{\beta})$. *We deduce* $\alpha^\lambda = \left(\frac{\tau}{1-\tau}\times\frac{1-p}{p}\right)^{1-\tau}$ *and* $\beta^\lambda = \left(\frac{\tau}{1-\tau}\times\frac{1-p}{p}\right)^{-\tau}$. *So,* $f(\lambda) = \left(\frac{\tau}{1-\tau}\times\frac{1-p}{p}\right)^{-\tau}\frac{1-p}{1-\tau} = \left(\frac{p}{\tau}\right)^\tau\left(\frac{1-p}{1-\tau}\right)^{1-\tau}$. *We obtain the result.*
In our case we obtain $C = 0.000487898 \approx \frac{1}{2\,050}$.

Q.2g Deduce an approximation for the required number of samples to distinguish the two distributions.

Treat the general case then compute this number in the example. Discuss the validity of the approximation.

Based on the Sanov theorem we approximate it to $1/C$.
In our example, this is 2 050. *This is quite pessimistic since we reach a pretty good advantage with 177 samples.*

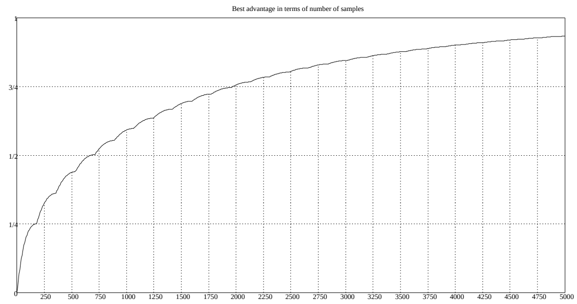
Q.2h Compute the Squared Euclidean imbalance between the two distributions and compare with the Chernoff information.

Treat the general case then compute the SEI in the example. Discuss the validity of the approximation for the Chernoff information.

We have

$$\mathsf{SEI} = N\left(p - \frac{1}{N}\right)^2 + N(N-1)\left(\frac{1-p}{N-1} - \frac{1}{N}\right)^2 = \frac{N^2}{N-1}\left(p - \frac{1}{N}\right)^2$$

In our case, this is $\mathsf{SEI} = \frac{1}{255}$*. We shall have* $C \approx \mathsf{SEI}/8\ln 2$*. In our case,* $8\ln 2/\mathsf{SEI} \approx 1\,414$*. So, the* $C \approx \mathsf{SEI}/8\ln 2$ *approximation is not very good in our example. Actually, this estimate is also pessimistic. We can compute the exact advantage for arbitrary* n *and we get that it becomes higher than* $\frac{1}{2}$ *as soon as* $n \geq 671$*. The bad estimates may come from the fact that the best advantage increases very smoothly after reaching* $\frac{1}{4}$ *as shown by the following graph.*



2 Breaking RSA with Low d Exponent

This exercise is inspired from Wiener, Cryptanalysis of Short RSA Secret Exponents, *published in IEEE Transactions on Information Theory vol. 36 in 1990.*

In this exercise we assume some RSA public key (N, e) and a secret key d such that $ed \bmod \varphi(N) = 1$. We let $pq = N$ be the factorization of N into primes. We assume that p and q are roughly of same length, i.e. $\frac{1}{c}\sqrt{N} \leq p, q \leq c\sqrt{N}$ for some $c \geq 1$ (e.g. $c = 2$). We assume that d is short so that $d \leq N^\alpha$ with $\alpha < \frac{1}{4}$. We will assume $N^\alpha \leq \frac{1}{c}N^{\frac{1}{4}}$. The objective of the exercise is to show that we can recover d from N and e in polynomial time.

Q.1

Q.1a Show that there exists an integer k such that $ed = k(N - p - q + 1) + 1$.

Since $ed \bmod \varphi(N) = 1$ *there exists* k *such that* $ed = k\varphi(N) + 1$. *We observe that* $\varphi(N) = (p-1)(q-1) = N - p - q + 1$ *and conclude.*

Q.1b Show that $\mathsf{gcd}(k, d) = 1$ and

$$0 \leq \frac{k}{d} - \frac{e}{N} = \frac{k}{d} \times \frac{p + q - 1 - \frac{1}{k}}{N}$$

Due to $ed = k\varphi(N) + 1$, $\mathsf{gcd}(k, d)$ *divides* d *and* k *so it must divides 1 as well. Therefore,* $\mathsf{gcd}(k, d) = 1$. *We divide the equation* $ed = k(N - p - q + 1) + 1$ *by* dN. *The equality comes from straightforward computation. The inequality comes from that* $p + q - 1 - \frac{1}{k} \geq 0$.

Q.1c Deduce that if d is such that $d \leq \frac{1}{c}N^{\frac{1}{4}}$ and $c \geq 3$, then $0 \leq \frac{k}{d} - \frac{e}{N} \leq \frac{2}{3d^2}$.
Hint: show that $\frac{p+q}{N} \leq \frac{2}{3d^2}$ and use the result from Q.1b.

Since $e < \varphi(N)$, *we have that* $\frac{k}{d} = \frac{e - \frac{1}{d}}{\varphi(N)} < 1$. *We have* $$\frac{p+q}{N} = \frac{1}{\sqrt{N}}\left(\frac{p}{\sqrt{N}} + \frac{\sqrt{N}}{p}\right) \leq \frac{2c}{\sqrt{N}} \leq \frac{2c^2}{3\sqrt{N}} \leq \frac{2}{3d^2}$$ *for* $c \geq 3$. *So,* $0 \leq \frac{k}{d} - \frac{e}{N} = \frac{k}{d} \times \frac{p+q-1-\frac{1}{k}}{N} \leq \frac{2}{3d^2}$.

In the remaining part of the exercise, we will consider an arbitrary rational number x such that there exist integers μ and ν such that $\mathsf{gcd}(\mu, \nu) = 1$ and $0 \leq \frac{\mu}{\nu} - x \leq \frac{2}{3\nu^2}$. We will show that we can build an algorithm making from x a list of rational numbers containing $\frac{\mu}{\nu}$ in polynomial time in the bitlengths of x. (Note that the bitlength of a rational number is the cumulated bitlength of its numerator and denominator.)

Q.2 Under the assumption that this algorithm is found, deduce that we can recover d in polynomial time.

Let $x = \frac{e}{N}$, $\mu = k$ and $\nu = d$. Thanks to Q.1, these satisfy the hypothesis $0 \leq \frac{\mu}{\nu} - x \leq \frac{2}{3\nu^2}$ and $\mathsf{gcd}(\mu, \nu) = 1$. Using the algorithm (to be found), we list values including $\frac{\mu}{\nu}$ whose least denominator is d. Once we have a candidate for d, we can check it to see if it decrypts RSA encryptions. So, we isolate the correct value. The complexity is polynomial in terms of the bitlength of N.

Q.3 How can we factor N from e and d?

We write $ed - 1 = 2^t s$ with s odd, pick a random x and raise $x^t \bmod N$. If it is 1, we try again. Otherwise, we repeatedly square modulo N until it becomes stationary. If the limit is not 1, then x is not invertible so $\mathsf{gcd}(N, x)$ yields a p or q. Otherwise, we take the last value y different from 1. If $y = N - 1$, we try again. Otherwise, $\mathsf{gcd}(N, x - 1)$ yields a p or q.

In what follows we forget about RSA and its settings. We only consider the positive rational number x. Given a sequence of integers (or real numbers in Q.4a) $a_0, a_1, \ldots$ such that $a_i > 0$ for all i and an integer n, we define the following notation:

$$[a_0, a_1, \ldots, a_n] = a_0 + \frac{1}{a_1 + \frac{1}{\ldots + \frac{1}{a_n}}}$$

Q.4 Let u and v be the sequences defined by

$$\begin{array}{lll} u_n = a_n u_{n-1} + u_{n-2} & u_{-1} = 1 & u_{-2} = 0 \\ v_n = a_n v_{n-1} + v_{n-2} & v_{-1} = 0 & v_{-2} = 1 \end{array}$$

Q.4a Show that $[a_0, a_1, \ldots, a_n] = \frac{u_n}{v_n}$ for all n.
Hint: first show that $[a_0, a_1, \ldots, a_{n-1}, x] = \frac{x u_{n-1} + u_{n-2}}{x v_{n-1} + v_{n-2}}$ for all real number x and all n.
Hint^2: try with $x = a_n + \frac{1}{x'}$.

The hint equation clearly holds for $n = 0$. Assume that it holds for n and we show it for $n + 1$. We have $[a_0, a_1, \ldots, a_{n-1}, a_n, x] = [a_0, a_1, \ldots, a_{n-1}, [a_n, x]]$ so we let $x' = [a_n, x] = a_n + \frac{1}{x}$ and we have $[a_0, a_1, \ldots, a_{n-1}, a_n, x] = \frac{x' u_{n-1} + u_{n-2}}{x' v_{n-1} + v_{n-2}} = \frac{x u_n + u_{n-1}}{x v_n + v_{n-1}}$. So, the hint equation is shown. We apply it with $x = a_n$ and we are done.

Q.4b Show that $u_n v_{n-1} - u_{n-1} v_n = -(-1)^n$ for all n.

It holds by induction: it is quite clear for $n = -2, -1$. If it holds for $n - 1$ and $n - 2$, then

$$\begin{aligned} u_n v_{n-1} - u_{n-1} v_n &= (a_n u_{n-1} + u_{n-2}) v_{n-1} - u_{n-1}(a_n v_{n-1} + v_{n-2}) \\ &= u_{n-2} v_{n-1} - u_{n-1} v_{n-2} \\ &= (-1)^{n-1} \\ &= -(-1)^n \end{aligned}$$

so it holds for n as well.

Q.4c From now on, we assume that the a_i's are integers. Deduce that $\mathsf{gcd}(u_n, v_n) = 1$ and that $\frac{u_n}{v_n} - \frac{u_{n-1}}{v_{n-1}} = \frac{-(-1)^n}{v_n v_{n-1}}$ for all n.

We can divide the previous equation by $\mathsf{gcd}(u_n, v_n)$ *and get an integer equal to* $-(-1)^n/\mathsf{gcd}(u_n, v_n)$*. So* $\mathsf{gcd}(u_n, v_n)$ *divides 1 so it must be 1. We divide the equation by* $v_n v_{n-1}$ *and obtain the result.*

Q.5 Let $x \geq 0$ be a real number. We define the $a_0, a_1, \ldots$ sequence of *integers* which can be either finite of infinite as follows:

let $r \leftarrow x$ and $n \leftarrow -1$
loop
  let $n \leftarrow n + 1$
  let $a_n = \lfloor r \rfloor$
  exit if $r = a_n$
  let $r \leftarrow \frac{1}{r - a_n}$
end loop

We define the sequences u and v from a as before.

Q.5a Show that for all n, x is between $\frac{u_n}{v_n}$ and $\frac{u_{n-1}}{v_{n-1}}$.
Hint: show that $[a_0, a_1, \ldots, a_n, r] = x$ at every iteration of the loop.

We show by induction that $[a_0, a_1, \ldots, a_n, r] = x$ *every time we enter into the loop. Then, we deduce that* $[a_0, a_1, \ldots, a_n] \leq x$ *when* n *is even and* $[a_0, a_1, \ldots, a_n] \geq x$ *when* n *is odd. So, the* $\frac{u_n}{v_n}$ *is alternating lower and higher than* x*.*

Q.5b Show that when x is rational, the algorithm terminates and $x = [a_0, \ldots, a_n]$ when it stops.
Hint: show that $\frac{1}{\nu v_n} \leq \left| x - \frac{u_n}{v_n} \right| \leq \frac{1}{v_{n-1} v_n}$ if $x \neq \frac{u_n}{v_n}$.

We write $x = \frac{\mu}{\nu}$ *with* $\mathsf{gcd}(\mu, \nu) = 1$*. We have* $\left| x - \frac{u_n}{v_n} \right| \leq \left| \frac{u_{n-1}}{v_{n-1}} - \frac{u_n}{v_n} \right| = \frac{1}{v_{n-1} v_n}$ *thanks to last two questions. On the other hand, if* $x \neq \frac{u_n}{v_n}$*, we have that* $\left| x - \frac{u_n}{v_n} \right| \geq \frac{1}{\nu v_n}$ *so* $\frac{1}{\nu} \leq \frac{1}{v_{n-1}}$ *so* $v_{n-1} \leq \nu$*. However,* v_n *strictly increases, so it must become higher* ν*. So, the algorithm terminates.*
The property $[a_0, a_1, \ldots, a_n, r] = x$ *at the entrance of the very last iteration leads to increasing* n *and assigning* r *to the final* a_n*. So,* $[a_0, a_1, \ldots, a_{n-1}, a_n] = x$ *at the end.*

Q.5c Deduce that the algorithm terminates if and only if x is rational.

We have shown that the algorithm terminates when x *is rational. Conversely, if the algorithm terminates, we can write* x *as a rational expression in terms of integers, so* x *must be rational.*

Q.5d Show that every positive rational number can be written $[a_0, a_1, \ldots, a_n]$ with a_i positive integers, $a_i \neq 0$ for $i > 0$, and $a_n \geq 2$ in the $n > 0$ case.

We observe that if $r > 1$ *in the loop, then* a_n *cannot be 0 and the new* r *must satisfy* $r > 1$ *again. If* $0 \leq x \leq 1$*, then* $a_0 = 0$ *but the new* r *satisfies* $r > 1$ *so only* a_0 *can be 0. Finally, the last* a_n *cannot be equal to 1. Otherwise, it would mean that* $r = a_{n-1} + 1$ *in the previous iteration so the computation of* a_{n-1} *would be wrong.*

Q.6

Q.6a Show that

$$[a_0, \ldots, a_{n-1}, a_n + \delta] - [a_0, \ldots, a_{n-1}, a_n] = \frac{\delta(-1)^n}{v_n(v_n + \delta v_{n-1})}$$

Hint: sorry, no hint here.

Note that if we prove it for every positive integer δ, we have two rational functions of δ which match for infinitely many δ's, so they match for all of them. We have

$$[a_0, \ldots, a_{n-1}, a'_n] = \frac{u'_n}{v'_n} = \frac{a'_n u'_{n-1} + u'_{n-2}}{a'_n v'_{n-1} + v'_{n-2}} = \frac{a'_n u_{n-1} + u_{n-2}}{a'_n v_{n-1} + v_{n-2}} = \frac{(a_n + \delta)u_{n-1} + u_{n-2}}{(a_n + \delta)v_{n-1} + v_{n-2}}$$

where $a'_n = a_n + \delta$ and u' and v' are the sequence obtained with this new a'_n. With $\delta = 0$ we have

$$[a_0, \ldots, a_{n-1}, a_n] = \frac{a_n u_{n-1} + u_{n-2}}{a_n v_{n-1} + v_{n-2}}$$

So, by computing the difference, regrouping, and using $u_{n-1}v_{n-2} - u_{n-2}v_{n-1} = (-1)^n$, we obtain the result.

Q.6b Assume that $\mathsf{gcd}(\mu, \nu) = 1$. We denote $x' = [a'_0, \ldots, a'_n]$ the result from the algorithm with $x' = \frac{\mu}{\nu}$ instead of x. Prove that if $0 \leq x - \frac{\mu}{\nu} \leq \frac{2}{3\nu^2}$, then $a_i = a'_i$ for $i < n$ and $a_n = a'_n - (n + 1 \bmod 2)$.
Hint: skip this question.

Let δ be such that $[a'_0, \ldots, a'_{n-1}, a'_n+\delta] = x$. Note that δ has the same sign as $(-1)^{n+1}$ since $x \leq \frac{\mu}{\nu}$. Since $x - \frac{\mu}{\nu} = \frac{\delta(-1)^n}{v'_n(v'_n+\delta v'_{n-1})}$, $v'_n + \delta v'_{n-1} > 0$. We have

$$\frac{2}{3\nu^2} \geq \left| x - \frac{\mu}{\nu} \right| = \frac{|\delta|}{v'_n(v'_n + \delta v'_{n-1})}$$

By construction, we have $v'_n = \nu$.
If $\delta < 0$ (n even), we deduce $|\delta| \leq \frac{2}{3}$. So, we can write $[a'_0, \ldots, a'_{n-1}, a'_n - 1 + \delta'] = x$ with $1 > \delta' > 0$.
If $\delta > 0$ (n odd), since $v'_n = a'_n v'_{n-1} + v'_{n-2}$ and $a'_n \geq 2$ we have $v'_{n-1} < \frac{1}{2}v'_n$. So,

$$\frac{2}{3} \geq \frac{\delta}{1 + \frac{1}{2}\delta}$$

which leads us to $\delta < 1$. So, we can write $[a'_0, \ldots, a'_{n-1}, a'_n + \delta] = x$ with $1 > \delta \geq 0$. This shows that running the algorithm on x will output $a_i = a'_i$ for $i = 0, \ldots, n-1$ and $a_n = a'_n - 1$ or a'_n depending on the parity of n.

Q.6c Deduce that if $0 \leq \frac{\mu}{\nu} - x \leq \frac{2}{3\nu^2}$ there exists n such that $\frac{\mu}{\nu} = [a_0, \ldots, a_n + (n+1 \bmod 2)]$.
Hint: assume you did the previous question.

We apply the previous result with n set to the number of iteration for $\frac{\mu}{\nu}$.

Q.7 By observing that v_n grows faster than a Fibonacci sequence, show that if x is rational, the number of iterations of the previous algorithm is linearly bounded in terms of the bitlength of x.

Let w_n be defined by $w_n = w_{n-1} + w_{n-2}$, $w_{-1} = 0$, and $w_{-2} = 1$. By induction we have that $v_n \geq w_n$ for all n. until the end of the sequence. This Fibonacci sequence has terms in $\left(\frac{1\pm\sqrt{5}}{2}\right)^n$. We can actually show that $w_n \geq \frac{1}{5}\left(\frac{1+\sqrt{5}}{2}\right)^{n+3}$. So, v_n grows exponentially fast and reaches the size of the denominator within a linear number of iterations n.

Q.8

Q.8a Wrap up: show that if x is rational and if there exists μ and ν such that $0 \leq \frac{\mu}{\nu} - x \leq \frac{2}{3\nu^2}$, we can make from x a list of rational numbers containing $\frac{\mu}{\nu}$ in polynomial time.
Hint: just write the algorithm with an explanation about the odd n case.

The following algorithm prints the sequence of all $[a_0, \ldots, a_{n-1}, a_n + (n \bmod 2)]$.

let $r \leftarrow x$, $n \leftarrow -1$
let $u_{-1} = 1$, $u_{-2} = 0$, $v_{-1} = 0$, $v_{-2} = 1$
loop
  let $n \leftarrow n + 1$
  let $a_n = \lfloor r \rfloor$
  let $u_n = a_n u_{n-1} + u_{n-2}$
  let $v_n = a_n v_{n-1} + v_{n-2}$
  if $n \bmod 2 = 1$ ***then***
    print u_n / v_n
  else
    print $(u_n + u_{n-1})/(v_n + v_{n-1})$
  end if
  exit if $r = a_n$
  let $r \leftarrow \frac{1}{r - a_n}$
end loop

In the even n case, we replace a_n by $a_n + 1$ so u_n by $u_n + u_{n-1}$ due to $u_n = a_n u_{n-1} + u_{n-2}$, and the same for v_n.

Q.8b Show that the following algorithm breaks RSA within a linear number of iterations.

let $r \leftarrow \frac{e}{N}$, $n \leftarrow -1$
let $v_{-1} = 0$, $v_{-2} = 1$
let $\rho \leftarrow$ random
loop
  let $n \leftarrow n + 1$
  let $a_n = \lfloor r \rfloor$
  let $v_n = a_n v_{n-1} + v_{n-2}$
  let $r \leftarrow \frac{1}{r - a_n}$
  let $d \leftarrow v_n + v_{n-1} \times (n + 1 \bmod 2)$
  if $\rho^{ed-1} \bmod N = 1$ **then**
    print d, factor N, and exit
  end if
end loop

For the RSA attack the computation of u is useless. Instead of printing $v_n + v_{n-1}(n \bmod 2)$ we check if d can decrypt $\rho^2 \bmod N$ and factor N if this is the case.