Cryptography and Security — Final Exam
Solution

Serge Vaudenay

22.1.2014

- duration: 3h00
- no document is allowed except one two-sided sheet
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will *not* answer any technical question during the exam
- the answers to each exercise must be provided on separate sheets
- readability and style of writing will be part of the grade
- do not forget to put your name on every sheet!

The exam grade follows a linear scale in which each question has the same weight.

1 AES Arithmetics

We consider all polynomials in terms of x, where x is a solution to $z^8 + z^4 + z^3 + z + 1 = 0$, with coefficients in $\mathbf{Z}_2$. This is supposed to define $\mathsf{GF}(2^8)$. For convenience, each element is represented in hexadecimal by a number whose binary expansion lists the binary coefficients of the monomials from the one of highest degree to the one of lowest degree. For instance, `0xa3` represents $x^7 + x^5 + x + 1$.

Q.1 Compute $\texttt{0x95} + \texttt{0x54}$.

We have $\texttt{0x95} = x^7 + x^4 + x^2 + 1$ *and* $\texttt{0x54} = x^6 + x^4 + x^2$. *So,*

$$\begin{aligned}\texttt{0x95} + \texttt{0x54} &= x^7 + x^4 + x^2 + 1 + x^6 + x^4 + x^2 \\ &= x^7 + x^6 + 1 \\ &= \texttt{0xc1}\end{aligned}$$

Q.2 Compute $\texttt{0x3c} \times \texttt{0x18}$.

We have $\texttt{0x3c} = x^5 + x^4 + x^3 + x^2$ *and* $\texttt{0x18} = x^4 + x^3$. *So,*

$$\begin{aligned}\texttt{0x3c} \times \texttt{0x18} &= (x^5 + x^4 + x^3 + x^2)(x^4 + x^3) \\ &= xx^8 + x^5 \\ &= x(x^4 + x^3 + x + 1) + x^5 \\ &= x^4 + x^2 + x \\ &= \texttt{0x16}\end{aligned}$$

Q.3 Compute $(\texttt{0x02})^{-1}$.

We have

$$\begin{aligned}\frac{1}{\texttt{0x02}} &= \frac{1}{x}\\ &= \frac{x^8+x^4+x^3+x}{x}\\ &= x^7+x^3+x^2+1\\ &= \texttt{0x8d}\end{aligned}$$

Q.4 Show that $z^{255}=1$ for all $z\neq 0$ in $\mathsf{GF}(2^8)$.

We have defined a finite field of 256 elements. Its multiplicative group consists of all nonzero elements and has 255 elements. Due to the Lagrange theorem, we have $z^{255}=1$ *for all* $z\neq 0$.

Q.5 Compute $(\texttt{0x02})^{254}$.

Clearly, $(\texttt{0x02})^{254} = \frac{(\texttt{0x02})^{255}}{\texttt{0x02}} = \frac{1}{\texttt{0x02}} = \texttt{0x8d}$.

Q.6 Give the hexadecimal representation of at least four solutions (including x) to $z^8+z^4+z^3+z+1=0$.
HINT: squaring is a linear operation!

If y *satisfies* $y^8+y^4+y^3+y+1=0$, *we note that*

$$(y^2)^8+(y^2)^4+(y^2)^3+(y^2)+1=(y^8+y^4+y^3+y+1)^2=0$$

So, the roots are the iterated squares: $x=\texttt{0x02}$, $x^2=\texttt{0x04}$, $x^4=\texttt{0x10}$, $x^8=\texttt{0x1b}$, *etc.*

Q.7 We recall that the trace function is defined by

$$\mathsf{tr}(z)=\sum_{i=0}^{7} z^{2^i}$$

Based on the previous question, compute $\mathsf{tr}(x)$.

Since the roots of $z^8+z^4+z^3+z+1$ *are the* x^{2^i}, *we have*

$$z^8+z^4+z^3+z+1=\prod_{i=0}^{7}\left(z-x^{2^i}\right)$$

So, the sum of the x^{2^i} *is the coefficient of* z^7. *Hence,* $\mathsf{tr}(x)=0$.

2 Distribution of Birthdays

This semester, we had $M = 81$ registered students. We assume their birthdays are *a priori* uniformly distributed and independent, in a calendar of $N = 365$ days. (Indeed, no student is born on a February 29). In what follows, the *a priori* probabilities refers to the situation *before* we look at the actual birthdays (which we will do in Q.5).

Q.1 What was the *a priori* probability that a given student is born on a January 22?

It is $1/N$ due to uniform distribution.

Q.2 What is the *a priori* probability that your right neighbor shares with you the same birthday? (Consider the left neighbor if you have no right neighbor.)

It is again $1/N$ due to uniform distribution and independence.

Q.3 For a given student, what is the *a priori* probability that there are *exactly* two others students in the class sharing the same birthday with him?

The probability to be in a 3-collisions is $\binom{M-1}{2}\frac{1}{N^2}\left(1-\frac{1}{N}\right)^{M-3} \approx 1.9\%$.
We could do the same computation for a 2-collision: the probability that exactly one other student share his birthday is $\binom{M-1}{1}\frac{1}{N}\left(1-\frac{1}{N}\right)^{M-2} \approx 18\%$.
The probability that no student share his birthday is $\left(1-\frac{1}{N}\right)^{M-1} \approx 80\%$.

Q.4 What was the *a priori* expected number of unordered pairs of different students with the same birthday? Do the same for unordered triplets of students.
HINT: the number of pairs of students with the same birthday is

$$\sum_{\mathsf{pair}} 1_{\mathsf{the\ students\ in\ pair\ share\ the\ same\ birthday}}$$

We have $\binom{M}{2}$ pairs of students. Each pair share the same birthday with probability $1/N$. So, is should be $\binom{M}{2}\frac{1}{N} \approx 8.9$.
For triplets, this is $\binom{M}{3}\frac{1}{N^2} \approx 0.6$.

Q.5 We observed

- 3 students are born on a March 2,
- 3 students are born on a May 5,
- 3 students are born on a July 4,
- 2 students are born on a April 14,
- 2 students are born on a May 9,
- 2 students are born on a June 2,
- 2 students are born on a June 13,
- 2 students are born on a August 5,
- 2 students are born on a November 1,

- 60 students have a unique birthday.

From the observation, how many unordered pairs of different students have the same birthday? Do the same for unordered triplets of students. How to explain the discrepancy?

The pairs are from the pool of 2-collisions and 3-collisions. In a 3-collision, we have 3 pairs. In a 2-collision, we have a single pair. So, we have $3 \times 3 + 6 \times 1 = 15$ *pairs of students with the same birthday. There is a gap between 15 and* 8.9.
Clearly, we have only 3 triplets of students sharing the same birthday. There is a gap between 3 and 0.6.
Most probably, the distribution of birthdays in the class is not uniform. This can explain the discrepancy.

Q.6 If each student independently selects a numeric PIN code of fixed length with uniform distribution, what is the minimal length (in digits) of the PIN code so that the probability of having two students selecting the same PIN code is lower than 1%?

We have $\Pr[\mathsf{collision}] \leq \binom{M}{2}\frac{1}{N}$ *with* $N = 10^\ell$. *We can solve* $\binom{M}{2}\frac{1}{10^\ell} \leq 0.01$ *and obtain* $\ell \geq 2 + \log_{10}\binom{M}{2} \approx 5.5$. *So,* $\ell = 6$ *digits are enough for* $\Pr[\mathsf{collision}] \leq 1\%$.
With $\ell = 5$ *digits, we have*

$$\Pr[\mathsf{collision}] = 1 - \prod_{i=0}^{M-1}\left(1 - \frac{i}{N}\right) \approx 0.03$$

so 5 digits are not enough. Note that we obtain the same result by using the approximation

$$\Pr[\mathsf{collision}] \approx 1 - e^{-\frac{M^2}{2N}}$$

3 (Non-Uniform) Attack on P256

> *The following exercise is inspired from* Non-Uniform Cracks in the Concrete: the Power of free Precomputation *by Bernstein and Lange, published in the proceedings of Asiacrypt'13 pp. 321–340, LNCS vol. 8270, Springer 2013.*

In this exercise, we consider the elliptic curve P256 defined with order n and a generator G. The integer n is a prime number of 256 bits. Given a point Q of the curve, we want to find $x \in \mathbf{Z}_n$ such that $Q = xG$.

Note that this exercise is not specific to P256 *but could apply to any cyclic group (with additive notations) of order n.*

We let H be a random function from the curve to $\mathbf{Z}_n$ and define $w(P) = P + H(P)G$ for a point P of the curve. Given a point P_0, the sequence defined by $P_i = w(P_{i-1})$ for $i > 0$ is called a *random walk* starting from P_0. We also consider a random Boolean function D. A point P such that $D(P) = 1$ is called a *distinguished point.* We assume that for all points P and P', all random variables $H(P)$ and $D(P')$ are independent, $H(P)$ is uniformly distributed in the curve, and $\Pr[D(P') = 1] = \frac{1}{t}$.

We consider the following algorithm defined by the parameters t and m, where the curve is assumed to be hard coded:

```
Precomp:
clear the list $L$
for $i = 1$ to $m$ do
  pick $a \in \mathbf{Z}_n$ at random and set $P_0 = aG$
  compute the random walk $P_0, \ldots, P_{\ell-1}$ starting from $P_0$ until either it loops (i.e., $P_{\ell-1} \in \{P_0, \ldots, P_{\ell-2}\}$) or it reaches a distinguished point (i.e., $D(P_{\ell-1}) = 1$)
  if the random walk loops then
    abort
  end if
  if not already there, insert $P_{\ell-1}$ and its logarithm in the list $L$
end for
output $L$
```

(In the abort case, we can just restart with new functions H and D.)

We also consider the following algorithm where the curve and the list L from Precomp are assumed to be hard coded:

```
Dlog($Q$):
loop
  pick $a \in \mathbf{Z}_n$ at random and set $P_0 = aG + Q$
  compute the random walk $P_0, \ldots, P_{\ell-1}$ until it loops or it reaches a distinguished point $P_{\ell-1}$
  if the random walk did not loop then
    if there exists $(P_{\ell-1}, b) \in L$ then
      stop
    end if
  end if
end loop
```

Q.1 Show that in the Precomp algorithm, every point P_i, $i = 0, \ldots, \ell-1$ has an easy-to-compute discrete logarithm.

The starting point P_0 has a known discrete logarithm a. If P_i is in the path and its discrete logarithm a_i is known, its successor $P_{i+1} = w(P_i) = P_i + H(P_i)G$ has a known discrete logarithm $a_i + H(P_i)$. So, all points in the path have a known discrete logarithm.

Q.2 How to slightly modify the algorithm Dlog so that when it stops, it gives the discrete logarithm of Q?

In each step of the random walk, we set $a \leftarrow a + H(P_i)$ so that at every time, we have $P_i = aP + Q$. If we find P_i in the list with discrete logarithm b, we just yield $b - a$ as the discrete logarithm of Q.

Q.3 (Number of iterations.)
We assume that Dlog never finds a looping random walk (i.e., the condition in Step 4 is always *true*). We further assume that Precomp visited at least $k\frac{n}{t}$ points (for some value k, where t is the parameter of the algorithm). Show that the expected number of iterations of the loop in Dlog (Steps 2–8) is at most $1 + \frac{1}{k}$.
HINT: observe that the random walk can be described by the following process: we pick a point at random and distinguishe the events that A: the point is already visited, B: the point is not already visited but distinguished, C: the point is not already visited nor distinguished. In the case of A, the random walk continues until a visited distinguished point and succeed. In the case of B, the random walk failed and the algorithm iterate. In the case of C, the random walk continues with a new point. No matter the stage of the random walk, $\Pr[A]/\Pr[B]$ is constant and at least k.

An iteration repeatedly picks a new point. If new, this new point is distinguished with probability $\frac{1}{t}$. But the point is not new with probability at least $\frac{\#\text{visited}}{n} \geq k\frac{1}{t}$. So, each new point has a probability to be new and distinguished with less than $\frac{1}{k}$ times the probability of being not new. So, we reach a distinguished point from L with probability at least $\frac{k}{k+1}$ and iterate with a probability less than $\frac{1}{k+1}$. So, the expected number of iterations is at most $1 + \frac{1}{k}$.

Q.4 (No long walk.)
Given $c > 1$, we let $\lambda = \lceil ct \ln n \rceil$. A *long walk* is a random walk $P_0, \ldots, P_{\lambda-1}$ which has no distinguished point. Show that the probability that there exists a long walk is bounded by n^{1-c}.
HINT: show that for each P_0, the probability that $P_0, \ldots, P_{\lambda-1}$ is a long walk is bounded by n^{-c}.
HINT: $\ln\left(1 - \frac{1}{t}\right) \leq -\frac{1}{t}$ for $t > 1$.

Given P_0, the function H defines $P_1, \ldots, P_{\lambda-1}$. The probability (over D) that there are no distinguished point in this list is $\left(1 - \frac{1}{t}\right)^{\lambda}$. So, the overall probability is bounded by

$$n\left(1 - \frac{1}{t}\right)^{\lceil ct \ln n \rceil} \leq n^{1 + ct \ln\left(1 - \frac{1}{t}\right)} \leq n^{1-c}$$

since $\ln\left(1 - \frac{1}{t}\right) \leq -\frac{1}{t}$ *for* $t > 1$.

In what follows, we take $c = 2$ and we assume there exists no path of length λ with no distinguished point.

Q.5 (No abort.)
In a given iteration of the algorithm, show that the probability that the random walk loops is bounded by $\frac{t^2}{n}$. Deduce that for $mt^2 \leq \frac{n}{2}$ (where t and m are the parameters of the algorithms), the algorithm Precomp aborts with a probability bounded by $\frac{1}{2}$.
HINT: $\sum_{\ell=1}^{+\infty} \left(1 - \frac{1}{t}\right)^{\ell-1} \frac{\ell}{n} = \frac{t^2}{n}$

For $\ell \geq 1$, the probability that the random walk loops after ℓ steps is the probability that any of the first $\ell - 1$ steps jumps onto a new point which is not distinguished (with probability $1 - \frac{1}{t}$), and that the last step jumps onto one of the previous $\ell - 1$ points. This is bounded by

$$\sum_{\ell=1}^{+\infty} \left(1 - \frac{1}{t}\right)^{\ell-1} \frac{\ell}{n}$$

which is equal to $\frac{t^2}{n}$. Since we make m random walks, the probability that the algorithm aborts is bounded by $\frac{mt^2}{n}$. So, for $mt^2 \leq \frac{n}{2}$, the algorithm Precomp *aborts with a probability bounded by $\frac{1}{2}$.*

Q.6 (Many visited points.)
Show that every time the algorithm computes a new point (i.e., starts a new walk or makes a new step), the probability to stop or to select an already visited point is between $\frac{1}{t}$ and $q = \frac{1}{t} + \left(1 - \frac{1}{t}\right) \frac{m\lambda}{n}$. Deduce that in each iteration, the probability that the random walk visits at least v new points is at least $(1 - q)^v$.

Let v be the number of already visited points. The new point is one of these with probability $\frac{v}{n}$. If not, the only reason to stop is to reach a distinguished point, which occurs with probability $\frac{1}{t}$. So, the probability is $\frac{v}{n} + \left(1 - \frac{v}{n}\right) \frac{1}{t}$. Since $0 \leq v \leq m\lambda$, it is between $\frac{1}{t}$ and $q = \frac{1}{t} + \left(1 - \frac{1}{t}\right) \frac{m\lambda}{n}$.
For an entire path, let q_i be the probability to reach an already visited point or to stop at the ith step. The probability that this path visits at least v points is thus $\sum_{i=1}^{v} (1 - q_1) \cdots (1 - q_i)$. We use $q_i \leq q$ to obtain that this is larger than $(1 - q)^v$.

By using the Chernoff bound, we deduce that for v such that $(1-q)^v > \frac{1}{2}$, more than half of the iterations visit less than v new points with a probability bounded by $e^{-2\left((1-q)^v - \frac{1}{2}\right)^2 m}$. In that case, the total number of visited points is at least $\frac{mv}{2}$, except with a probability bounded by this.
In what follows, we assume $m = \lfloor \frac{n}{\lambda t} \rfloor$ and we take $v = \lfloor \frac{t}{4} \rfloor$.

Q.7 By adjusting t, deduce that for each group generated by some G of order n, there *exists* an algorithm with precomputed data of size $\mathcal{O}(\sqrt[3]{n})$ and solving the discrete logarithm problem with time $\mathcal{O}(\sqrt[3]{n})$ multiplied by some polynomial in terms of $\log n$. (Here the big $\mathcal{O}$'s are when n tends towards infinity.)

We take $t = \lfloor \sqrt[3]{n} \rfloor$. So, L has size $m \sim \frac{n}{\lambda t} \sim \frac{\sqrt[3]{n}}{2 \ln n}$.
In Q.6, we have $q \sim \frac{2}{t}$, $v \sim \frac{t}{4}$, so $(1-q)^v \sim e^{-\frac{1}{2}} \approx 0.61 > \frac{1}{2}$. Thanks, to Q.6, Precomp visits at least $\frac{mv}{2} \sim \frac{mt}{8} = k\frac{n}{t}$ points except with negligible probability, with $k \sim \frac{1}{16 \ln n}$.
Following Q.3, one iteration of Dlog succeeds with probability $\frac{1}{1+16 \ln n}$. So, we have to iterate $\frac{1}{k}$ times.
Following Q.4, we assume no long walk. So, Dlog has a complexity bounded by $\frac{1}{k}\lambda \sim 32\sqrt[3]{n}(\ln n)^2$.
We have $mt^2 \sim \frac{n}{2 \ln n} < \frac{n}{2}$. So, thanks to Q.5, Precomp does not abort with probability at least $\frac{1}{2}$.
Similarly, none of the $\mathcal{O}(\log n)$ iterations of Dlog aborts, except with negligible probability.