Advanced Cryptography — Final Exam

Serge Vaudenay

16.6.2009

- 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 the sheet!

1 A Distinguisher

We consider an oracle A which, upon a query x which is a vector of k bits, behaves as follows:

Input: x
1: compute the vector $\bar{x}$ by flipping all bits of x
2: set $u = \bar{x}\|x$
3: pick a random permutation σ over $\{1, 2, \ldots, 2k\}$
4: apply transposition σ on u to get a vector v
namely, if $u = u_{2k}\|\cdots\|u_2\|u_1$ we have $v = u_{\sigma(2k)}\|\cdots\|u_{\sigma(2)}\|u_{\sigma(1)}$
5: set y to the k rightmost bits of v
Output: y

We denote $y = A(x)$. (We stress that $A(x)$ is a random variable.)

1. Given a random variable X we define its distribution function $P_X(x) = \Pr[X = x]$. Show that for any x and y we have
$$P_{A(x)}(y) = \frac{\binom{k}{k-w(y)}}{\binom{2k}{k}}$$
where $w(y)$ is the Hamming weight of y (i.e. the number of bits set to 1 in y). Deduce it does not depend on x.

> *Before permutation σ, the $2k$-bit string u has a Hamming weight of k. After permutation σ, the obtained $2k$-bit string v is a random string uniformly distributed among all those with Hamming weight k. We have to compute the probability that a random $2k$-bit string v ends with the y half. For this, we count how many strings exist and we divide by the total number of possible strings. There are $\binom{2k}{k}$ possible strings in total. To complete the v half to get a string with weight k, we have to pick a string of length k and weight $k - w(y)$. Clearly, we have $\binom{k}{k-w(y)}$. We deduce the formula for $P_{A(x)}(y)$ and we observe it does not depend on x and only depends on $w(y)$.*

As an application, compute the table of $P_{A(x)}$ with $k = 2$.

For $y = 0$ we have $P_{A(x)} = \frac{1}{6}$, for $y = 01$ and $y = 10$, we have $P_{A(x)} = \frac{1}{3}$, for $y = 11$, we have $P_{A(x)} = \frac{1}{6}$.

2. Deduce the best advantage of a distinguisher limited to a single query x for distinguishing A from a random oracle.

The best advantage is given by the statistical distance, so

$$\mathsf{BestAdv}_1 = \frac{1}{2}\sum_y \left| \frac{\binom{k}{k-w(y)}}{\binom{2k}{k}} - 2^{-k} \right|$$

We can group the y's by Hamming weight and obtain

$$\mathsf{BestAdv}_1 = \frac{1}{2}\sum_{w=0}^{k} \binom{k}{w} \left| \frac{\binom{k}{k-w}}{\binom{2k}{k}} - 2^{-k} \right|$$

For $k = 2$, compute the advantage.

We have

$$\mathsf{BestAdv}_1 = \frac{1}{2}\left(\left|\frac{1}{6} - \frac{1}{4}\right| + 2\left|\frac{1}{3} - \frac{1}{4}\right| + \left|\frac{1}{6} - \frac{1}{4}\right| \right) = \frac{1}{6}$$

which is pretty large. So, this construction introduces a significant bias.

3. Given a function $f : \{0,1\}^k \to \mathbf{R}$ we define its discrete Fourier transform

$$\hat{f}(a) = \sum_x (-1)^{a \cdot x} f(x)$$

Let r be the Hamming weight of the bitwise AND of a and x and let s be such that $r + s$ is the Hamming weight of x. Show that $a \cdot x$ can be expressed as a function in terms of r and s. By grouping the x's with same values of r and s in the sum, show that there is a function g such that $\hat{P}_{A(x)}(a) = g(w(a))$.

We have $a \cdot x = r \bmod 2$. Since $P_{A(x)}(y)$ is a function of $w(y)$ then it is a function of $r+s$. The number of y's given r and s is $\binom{w(a)}{r}\binom{k-w(a)}{s}$. We have

$$\hat{P}_{A(x)}(a) = \sum_y (-1)^{a\cdot y} \frac{\binom{k}{k-w(y)}}{\binom{2k}{k}}$$

So,

$$\hat{P}_{A(x)}(a) = \sum_{r=0}^{w(a)} \sum_{s=0}^{k-w(a)} \binom{w(a)}{r} \binom{k-w(a)}{s} (-1)^r \frac{\binom{k}{k-r-s}}{\binom{2k}{k}}$$

We thus have $\hat{P}_{A(x)}(a) = g(w(a))$ for

$$g(w) = \sum_{r=0}^{w} \sum_{s=0}^{k-w} \binom{w}{r} \binom{k-w}{s} (-1)^r \frac{\binom{k}{k-r-s}}{\binom{2k}{k}}$$

Compute the table of $\hat{P}_{A(x)}$ for $k=2$.

For $a=0$ we have $\hat{P}_{A(x)} = g(0) = 1$, for $a = 01$ and $a = 10$, we have $\hat{P}_{A(x)} = g(1) = 0$, for $a = 11$, we have $\hat{P}_{A(x)} = g(2) = -\frac{1}{3}$.

To fix the bias, we consider the following oracle B.

Input: x
1: **for** i=1 to r **do**
2: query $A(x)$ and get y_i
3: **end for**
4: set $y = y_1 \oplus \cdots \oplus y_r$
Output: y

Again, we denote $B(x)$ the random output from x.

4. Given two independent random variables X and Y, show that

$$P_{X\oplus Y}(z) = \sum_{x,y \text{ s.t. } x\oplus y = z} P_X(x) P_Y(y)$$

By definition we have

$$P_{X\oplus Y}(z) = \Pr[X \oplus Y = z] = \sum_{x,y \text{ s.t. } x\oplus y = z} \Pr[X = x \text{ and } Y = y]$$

and since X and Y are independent we obtain the announced result.

Deduce that

$$P_{B(x)}(y) = \sum_{\substack{y_1,\ldots,y_r \text{ s.t.} \\ y_1\oplus\cdots\oplus y_r = y}} \prod_{i=1}^{r} P_{A(x)}(y_i)$$

We prove it by induction on r. Clearly, it is true for $r = 1$. If it is true for $r - 1$ we prove it for r by letting X be the XOR of the $r - 1$ first y_i's and Y be the last y_r.

If we had to compute the table of $P_{B(x)}$ form this formula, what would be the complexity, roughly? Is it doable for $k = 10$ and $r = 10$?

We would have to sum $2^{k(r-1)}$ terms. For $k = 10$ and $r = 10$ this would be infeasible.

5. Show that for all a we have

$$\hat{P}_{X\oplus Y}(a) = \hat{P}_X(a) \times \hat{P}_Y(a)$$

i.e. the discrete Fourier transform of the distribution of $X \oplus Y$ is obtained by multiplying the discrete Fourier transforms of X and Y.

We have

$$\hat{P}_{X\oplus Y}(a) = \sum_z (-1)^{a\cdot z} P_{X\oplus Y}(z) = \sum_z (-1)^{a\cdot z} \sum_{x,y \text{ s.t. } x\oplus y=z} P_X(x)P_Y(y)$$

We rewrite it into

$$\hat{P}_{X\oplus Y}(a) = \sum_z \sum_{x,y \text{ s.t. } x\oplus y=z} (-1)^{a\cdot x} P_X(x)(-1)^{a\cdot y} P_Y(y)$$

Since z does not appear anymore, the inner sum finally sums over all x and y. We obtain

$$\hat{P}_{X\oplus Y}(a) = \sum_{x,y} (-1)^{a\cdot x} P_X(x)(-1)^{a\cdot y} P_Y(y)$$

which clearly factors into $\hat{P}_X(a)\hat{P}_Y(a)$.

Deduce that

$$\hat{P}_{B(x)}(a) = \left(\hat{P}_{A(x)}(a)\right)^r$$

Again, this is proven by induction on r.

If we had to compute the table of $\hat{P}_{B(x)}$ form this formula, what would be the complexity, roughly? Is it doable for $k = 10$ and $r = 10$? How about $k = 128$ and $r = 10$?

We would have to compute the table of $\hat{P}_{A(x)}$ and to raise its 2^k terms to the power r. There are efficient algorithms to compute the discrete Fourier transform. For $k = 10$ and $r = 10$ this would be easy. For $k = 128$ we cannot even store the table so it would be impossible.

6. For any function $f : \{0,1\}^k \to \mathbf{R}$ such that $\sum_x f(x) = 1$, show that

$$\sum_x \left(f(x) - 2^{-k}\right)^2 = 2^{-k} \sum_{a\neq 0} \left(\hat{f}(a)\right)^2$$

Hint: think about Parseval.

By expanding the left-hand side we obtain

$$\sum_x f(x)^2 - 2^{-k}$$

We notice that $\hat{f}(a) = \sum_x f(x) = 1$, *so the equation is equivalent to*

$$\sum_x f(x)^2 = 2^{-k} \sum_a \left(\hat{f}(a)\right)^2$$

To prove it, we start by the right-hand side sum. We have

$$\sum_a \left(\hat{f}(a)\right)^2 = \sum_a \sum_x \sum_y (-1)^{a\cdot(x\oplus y)} f(x) f(y)$$

We swap the sums and obtain

$$\sum_a \left(\hat{f}(a)\right)^2 = \sum_x \sum_y f(x) f(y) \sum_a (-1)^{a\cdot(x\oplus y)}$$

The inner sum is always 0 when $x \neq y$ *and equals* 2^k *otherwise. Hence,*

$$\sum_a \left(\hat{f}(a)\right)^2 = 2^k \sum_x f(x)^2$$

which is what we wanted to prove.

7. Deduce that the square Euclidean imbalance of $B(x)$ is

$$\mathsf{SEI}(B(x)) = \sum_{a\neq 0} \left(\hat{P}_{A(x)}(a)\right)^{2r}$$

By definition, we have

$$\mathsf{SEI}(B(x)) = 2^k \sum_y \left(P_{B(x)}(y) - 2^{-k}\right)^2$$

Thanks to the previous question, we obtain

$$\mathsf{SEI}(B(x)) = \sum_{a\neq 0} \left(\hat{P}_{B(x)}(a)\right)^2$$

We conclude be recalling our previous result $\hat{P}_{B(x)}(a) = \left(\hat{P}_{A(x)}(a)\right)^r$.

Finally deduce that

$$\mathsf{SEI}(B(x)) = \sum_{w=1}^{k} \binom{k}{w} (g(w))^{2r}$$

Is it feasible to compute it for $k = 128$ and $r = 10$?

In the previous equation we just group all a's by their Hamming weight w and recall $\hat{P}_{A(x)}(a) = g(w(a))$. To compute it we first have to make the table of g which is a double sum with at most k terms in each sum so we have less than k^2 terms to sum. Then, the above sum is over k terms, so this is easy to compute.

8. Deduce an estimate on the number of samples to distinguish $B(x)$ from a uniformly distributed random variable.

The number of sample is within the order of magnitude of the inverse of the Chernoff information which is roughly the SEI over $8\ln 2$. Hence, the number of sample is

$$q \approx \frac{8\ln 2}{\sum_{w=1}^{k}\binom{k}{w}(g(w))^{2r}}$$

9. As an application, compute this estimate for $k = 2$. How large r must be so that this is higher than 2^{80}?

We have seen that $g(1) = 0$ and $g(2) = -\frac{1}{3}$. Hence, $q \approx 8\ln 2 \times 9^r$. We need $r \geq 24$ to have $q \geq 2^{80}$.

2 Σ-Protocol for Cubic Residues

We consider an integer $n = p \times q$ where p and q are two primes numbers, 3 divides $p-1$ but not $q-1$.

1. Show that -3 is a quadratic residue modulo p.

We use the properties of the Jacobi symbol. We have $\left(\frac{-3}{p}\right) = \left(\frac{p}{-3}\right) = \left(\frac{1}{-3}\right) = +1$ so -3 is a quadratic residue modulo p.

2. Deduce that $X^2 + X + 1$ has 2 roots in $\mathbf{Z}_p$.

The discriminant of X^2+X+1 is -3. Let $-3 \equiv u^2 \pmod p$. Therefore, X^2+X+1 has two square roots $(-1 \pm u)/2 \bmod p$.
Alternately, we have $X^2 + X + 1 = (X + \frac{1}{2})^2 - \frac{u^2}{4} = (X - \frac{-1+u}{2})(X + \frac{-1-u}{2})$ from which we deduce the two roots.

3. Show that $X^3 - 1$ has exactly 3 different roots in $\mathbf{Z}_p$.

The polynomial $X^3 - 1$ cannot have more than 3 roots over the field $\mathbf{Z}_p$. Multiple roots must be roots of its derivative $3X^2$ which has only 0 as a root. So, $X^3 - s$ has no multiple roots when $s \in \mathbf{Z}_p^*$. The polynomial $X^3 - 1$ has root 1 and the roots of $X^2 + X + 1$. So, $X^3 - 1$ has exactly 3 roots.

Deduce that for all $s \in \mathbf{Z}_p^*$ the polynomial $X^3 - s$ has either no root or exactly 3 different roots.

We know it cannot have more than 3 roots. Assume it has one root θ. Let $1, \zeta, \zeta'$ be the 3 roots of $X^3 - 1$. We observe that $\theta, \theta\zeta, \theta\zeta'$ are 3 different roots of $X^3 - s$. So we have exactly 3 different roots.

4. By using the Chinese remainder theorem, show that any element of $\mathbf{Z}_n^*$ has either exactly 3 cubic roots or none. Those with cubic roots will be called *cubic residues*. We denote by CR_n the set of all cubic residues from $\mathbf{Z}_n^*$.

A number x is a cubic root of s modulo n iff it is a cubic root modulo p and modulo q. Since 3 is coprime with $\varphi(q)$, every residue has a unique cubic root modulo q. Hence, by using the Chinese remainder theorem we obtain that a number always has the same number of cubic roots modulo n and modulo p.

5. Inspire by the Fiat-Shamir Σ-protocol and propose a Σ-protocol for the relation

$$R((n, v), s) \Leftrightarrow vs^3 \bmod n = 1$$

Be careful to go through the check list which has been given in the course, describe all components of the Σ-protocol and prove it satisfies the required properties.

We propose

$$\begin{array}{ccc}
\textbf{\textit{Prover}} & & \textbf{\textit{Verifier}} \\
\textbf{witness}\textit{: } s & \textbf{input}\textit{: } (n,v) & \\
\textit{pick } r \in \mathbf{Z}_n^* & & \textit{pick } e \in \{0,1\} \\
x = r^3 \bmod n & \xrightarrow{\quad x \quad} & \\
 & \xleftarrow{\quad e \quad} & \\
y = rs^e \bmod n & \xrightarrow{\quad y \quad} & y^3 v^e \bmod n \stackrel{?}{=} x
\end{array}$$

By going through the checklist, we define:

- *the relation R is already defined*
- *the first prover function $\mathcal{P}(n,v;r) = r^3 \bmod n$*
- *the challenge domain $E = \{0,1\}$*
- *the second prover function $\mathcal{P}(n,v,e;r) = rs^e \bmod n$*
- *the verification function $V(n,v,x,e,y) \iff y^3 v^e \bmod n = x$*
- *the extractor algorithm $\mathcal{E}(n,v,x,e,y,e',y')$: since e and e' are different in $\{0,1\}$ we denote y_0 resp. y_1 the y or y' value corresponding to the challenge 0 resp. 1. We compute $z = y_1/y_0 \bmod n$.*
- *the simulator algorithm $\mathcal{S}(n,v,e;r)$: pick $y \in_U \mathbf{Z}_n^*$ form r and set $x = y^3 v^e \bmod n$.*

We can now prove all required properties:

- *(efficiency) all algorithms are polynomially bounded*
- *(completeness) for each $((n,v),s)$ in the language and a honestly generated transcript (x,e,y) then $V(n,v,x,e,y)$ holds.*
- *(special soundness) for each (n,v), if (x,e,y) and (x,e',y') are two accepting transcripts with same x, then $\mathcal{E}$ produces a witness. This comes from*

$$\left(\frac{y_1}{y_0}\right)^3 v \equiv \frac{y_1^3 v}{y_0^3} \equiv \frac{x}{x} \equiv 1 \pmod{n}$$

- *(honest verifier zero-knowledge) for a honest prover, y is always uniformly distributed (whatever e) and $x = y^3 v^e \bmod n$. For the simulator, this is the same. So, both transcripts have same distribution.*

3 The GQ Protocol

Σ-protocols are made with some components satisfying a list of requirements as explained in the course. We consider here Σ-protocols with the extra property of uniqueness of response: using the notations from the course, for each x, a, e, there exists a unique z such that the verification $V(x,a,e,z)$ holds.

1. Show that the Schnorr Σ-protocol provides uniqueness of response.

In the Schnorr protocol we have $ry^e = g^s$ in the group so the response s is the unique integer in $\mathbf{Z}_q$ such that $g^s = ry^e$.

Let (N, e) be an RSA public key. We consider the following GQ protocol with relation

$$R((N, e, X), x) \iff x^e \bmod N = X$$

$$\begin{array}{lcl}
\textbf{Prover} & & \textbf{Verifier} \\
\textbf{witness}: x \quad \textbf{input}: (N,e,X) & & \\
\text{pick } y \in \mathbf{Z}_N^* & & \text{pick } c \in \{0, 1, \ldots, 2^\ell - 1\} \\
Y \leftarrow y^e \bmod N & \xrightarrow{\quad Y \quad} & \\
 & \xleftarrow{\quad c \quad} & \\
z \leftarrow yx^c \bmod N & \xrightarrow{\quad z \quad} & z^e \stackrel{?}{\equiv} YX^c \pmod N
\end{array}$$

Warning: in the GQ protocol, notations are somewhat different from usual.

2. Assuming that GQ is a Σ-protocol, formalize all components except the extractor.

By going through the checklist, we define:
- *the relation R is already defined*
- *the first prover function $\mathcal{P}(N, e; r)$ generates y form r and output $Y = y^e \bmod N$*
- *the challenge domain $E = \{0, 1, \ldots, 2^\ell - 1\}$*
- *the second prover function $\mathcal{P}(N, e, c; r)$ computes y as before and $z = yx^c \bmod N$*
- *the verification function $V(N, e, Y, c, z) \iff z^e \equiv YX^c \pmod N$*
- *the extractor algorithm $\mathcal{E}(N, e, Y, c, z, c', z')$ is not asked in this question*
- *the simulator algorithm $\mathcal{S}(N, e, c; r)$: pick $z \in_U \mathbf{Z}_N^*$ form r and set $Y = z^e/X^c \pmod N$*

3. Show (except special soundness) that all properties are satisfied.

We can now prove all required properties:
- *(efficiency) all algorithms are polynomially bounded*
- *(completeness) for each $((N, e), x)$ in the language and a honestly generated transcript (Y, c, z) then $V(N, e, Y, c, z)$ holds.*
- *(special soundness) not asked in this question*
- *(honest verifier zero-knowledge) for a honest prover, z is always uniformly distributed (whatever c) and $Y = z^e/X^c \pmod N$. For the simulator, this is the same. So, both transcripts have same distribution.*

4. Show that GQ provides response uniqueness.

The response z must satisfy $z^e \equiv YX^c \pmod N$. Since (N, e) is a valid RSA key, there exists a secret key d and by raising the equation to the power d we have $z = Y^d X^{dc} \pmod N$, so z is unique.

5. When $\mathsf{gcd}(c_1 - c_2, e) = 1$, show that we can extract a witness from two transcripts (Y, c_1, z_1) and (Y, c_2, z_2).
Hint: use the extended Euclid algorithm to find two integers a and b such that $ae + b(c_1 - c_2) = 1$.

Let a and b from the extended Euclid algorithm be such that $ae + b(c_1 - c_2) = 1$. We have $z_1^e \equiv Y X^{c_1}$ and $z_2^e \equiv Y X^{c_2}$ so $X \equiv X^{ae} \frac{Y^b X^{bc_1}}{Y^b X^{bc_2}} \equiv X^{ae} \frac{z_1^{be}}{z_2^{be}}$ so $x = X^a \frac{z_1^b}{z_2^b} \bmod N$ satisfies $X \equiv x^e$.

6. Deduce that we have an extractor which might fail sometimes. Estimate the probability of failure for $e = 65\,537$.

When getting two transcripts with same Y the extractor $\mathcal{E}(N, e, Y, c, z, c', z')$ works by taking $X^a \frac{z_1^b}{z_2^b} \bmod N$ as above. It fails in the extended Euclid algorithm if $\mathsf{gcd}(c_1 - c_2, e) \neq 1$. For e prime, this is equivalent to e divides $c_1 - c_2$. For ℓ large and $e = 65\,537$, which is prime, the probability of this event is roughly $1/e$, which is small.