Family Name: . . . .

First Name: . . . .

Section: . . . .

Advanced Cryptography

Final Exam

Solutions

July 18th, 2006

Start at 9:15, End at 12:00

This document consists of 12 pages.

Instructions

Electronic devices are *not* allowed.

Answers must be written on the exercises sheet.

This exam contains 1 exercise.

Answers can be either in French or English.

Questions of any kind will certainly *not* be answered.
Potential errors in these sheets are part of the exam.

You have to put your full name on *each* page and you have to do it *now*.

An RSA Variant with Public Exponent 3

In this problem, we consider a special variant of RSA with public exponent e that is *not* coprime with $\varphi(N)$. For simplicity, we focus on $e = 3$. More precisely, key generation works as follows:

- pick r_1 of $\frac{s}{2}$ bits at random until $p = 9r_1 - 2$ is prime
- pick r_2 of $\frac{s}{2}$ bits at random until $q = 3r_2 - 1$ is prime
- take $N = pq$, $e = 3$
- public key is (N, e), secret key is (p, q)

Cubic Residuosity

1. Let $x \in \mathbf{Z}_q^*$.

   How many cubic roots can we have?

   How to compute cubic roots in $\mathbf{Z}_q^*$?

Consider α be an element of $\mathbf{Z}_q^*$. Let β defined by $\beta = \alpha^3 \bmod q$. We are searching all possible values γ such that $\beta^\gamma \equiv \alpha^{3\gamma} \equiv \alpha \pmod q$. Since 3 is invertible in $\mathbf{Z}_{q-1}^*$, we have a bijection, i.e., each element $x \in \mathbf{Z}_q^*$ has *exactly one* cubic root and it can be computed by

$$x^{3^{-1} \bmod q-1}.$$

2. Let $x \in \mathbf{Z}_p^*$.

   Show that $(x^3)^{\frac{p+2}{9}}$ is a cubic root of x^3.

Let y be a cubic residue, i.e., $y = x^3 \pmod p$. We have to show that $y^{\frac{p+2}{9}}$ is a cubic root of y, i.e.,

$$\left(y^{\frac{p+2}{9}}\right)^3 \equiv y \pmod p$$

We can write

$$\begin{aligned}\left((x^3)^{\frac{p+2}{9}}\right)^3 &\equiv x^{p+2} \pmod p\\ &\equiv x^{p-1} \cdot x^3 \pmod p\\ &\equiv x^3 \pmod p\end{aligned}$$

since $x^{p-1} \equiv 1 \pmod p$.

3. Given $x \in \mathbf{Z}_p^*$, how many cubic roots can we have in $\mathbf{Z}_p^*$?

Since p is prime, the group $\mathbf{Z}_p^*$ is cyclic and can be represented by $\{g, g^2, \ldots, g^{p-1} = 1\}$ where g is a generator.
After the mapping $x \mapsto y = x^3 \pmod{p}$, y is an element of a group G. Now we can map each element of $\mathbf{Z}_p^*$ into G and we can write

$$G = \{g^3, g^6, \ldots, g^{3 \cdot \frac{(p-1)}{3}} = 1, \ldots, g^{3 \cdot 2\frac{(p-1)}{3}} = 1, \ldots, g^{3\cdot(p-1)} = 1\}$$

since 3 divides $p-1$ ($p - 1 = 3 \cdot (3r_1 - 1)$).
An element x of $\mathbf{Z}_p^*$ has *exactly three* roots if and only if x belongs to the subgroup G. Note that this condition can be checked using $x^{3r_1-1} \equiv 1 \pmod{p}$. Otherwise, the element x has *no* root.

4. By using the Jacobi symbol and its computation rules, prove that -3 is a quadratic residue in $\mathbf{Z}_p^*$.

We have to compute the Jacobi symbol of -3 in $\mathbf{Z}_p^*$. So, we have to compute $\left(\frac{-3}{p}\right)$.
We can directly write:

$$(\frac{-3}{p}) = (\frac{-1}{p})(\frac{3}{p})$$

We are trying to inverse the jacobi symbol. So,

- $\left(\frac{3}{p}\right) = \left(\frac{p}{3}\right)$ if $p \equiv 1 \pmod{4}$
- $\left(\frac{3}{p}\right) = -\left(\frac{p}{3}\right)$ if $p \not\equiv 1 \pmod{4}$.

We know that $(-1)^{\frac{p-1}{2}}$ is equal to 1 or -1:

- We note that $(-1)^{\frac{p-1}{2}} = 1$ when $\frac{p-1}{2}$ is **even** which is the case when $4 \mid p-1$. This implies that $p - 1 \equiv 0 \pmod{4}$ and so $p \equiv 1 \pmod{4}$. In that case, we conclude that $\left(\frac{3}{p}\right) = \left(\frac{p}{3}\right)$. Recall that $p = 9r_1 - 2$ and so $p \equiv 1 \pmod{3}$. Finally, we have $\left(\frac{p}{3}\right) = \left(\frac{1}{3}\right) = 1$.
- As opposite, we note that $(-1)^{\frac{p-1}{2}} = -1$ when $\frac{p-1}{2}$ is **odd** which is the case when $4 \nmid p-1$. This implies that $p-1 \not\equiv 0 \pmod{4}$ and so $p \not\equiv 1 \pmod{4}$. We conclude that $p \equiv 3 \pmod{4}$ since p can not be even which exclude the 0 and 2. In that case, we conclude that $\left(\frac{-3}{p}\right) = -\left(\frac{p}{3}\right) = -1$.

In conclusion, we have $\left(\frac{-3}{p}\right) = \left(\frac{-1}{p}\right)\left(\frac{3}{p}\right) = 1 \times 1 = 1$ when $p \equiv 1 \pmod{4}$, and $\left(\frac{-3}{p}\right) = \left(\frac{-1}{p}\right)\left(\frac{3}{p}\right) = (-1) \times (-1) = 1$ when $p \not\equiv 1 \pmod{4}$.
Finally, we can write

$$(\frac{-3}{p}) = 1$$

which proves that -3 in a quadratic residue in $\mathbf{Z}_p^*$.

5. Let $j = \frac{\theta-1}{2} \bmod p$ where θ is a square root of -3.

   Show that $j^3 \bmod p = 1$.

We simply write

$$\begin{aligned} j^3 &= \frac{(\theta-1)^3}{2^3} \\ &= \frac{\theta^3 - 3\theta^2 + 3\theta - 1}{8} \\ &\overset{\theta^2\equiv-3 \pmod p}{=} \frac{-3\theta + 9 + 3\theta - 1}{8} \\ &= 1 \end{aligned}$$

6. Deduce all cubic roots of 1 in $\mathbf{Z}_p^*$.

The first root is straightforward, it is 1. The second comes from the previous point and it is j. The last one is j^2 since $(j^2)^3 \equiv (j^3)^2 \equiv 1 \pmod p$.

In summary, the three roots are $1, j, j^2$.

7. Deduce a way to compute all cubic roots of cubic residues in $\mathbf{Z}_p^*$.

Let y be a cubic residue in $\mathbf{Z}_p^*$. Using a previous result, we know how to find the first root, i.e., by computing $x_1 = y^{\frac{p+2}{9}} \bmod p$. Then, we find the two other roots using the result of the previous point, i.e., $x_2 = j \cdot x_1$ and $x_3 = j \cdot x_2$.

8. By using the Chinese Remainder Theorem, tell how many cubic roots cubic residues have in $\mathbf{Z}_N^*$ and how to compute them.

Using the CRT Theorem, we know that $\mathbf{Z}_N^*$ is isomorph to $\mathbf{Z}_p^* \times \mathbf{Z}_q^*$. An element y of $\mathbf{Z}_N^*$ can be written $(y_p, y_q) \in \mathbf{Z}_p^* \times \mathbf{Z}_q^*$ where $y_p = y \bmod p$ and $y_q = y \bmod q$.

We know that a cubic residue in $\mathbf{Z}_p^*$ has three roots and a cubic residue in $\mathbf{Z}_q^*$ has one root. We conclude that a cubic residue in $\mathbf{Z}_N^*$ has three roots.

We first compute y_p and its three roots in $\mathbf{Z}_p^*$, i.e. $x_{p,1}, x_{p,2}, x_{p,3}$, then we compute y_q and its unique root in $\mathbf{Z}_q^*$, i.e., x_q.
The three roots of y in $\mathbf{Z}_N^*$ are obtained with

$$x_i = x_{p,i} \cdot q \cdot (q^{-1} \bmod p) + x_q \cdot p \cdot (p^{-1} \bmod q) \bmod N$$

where $i = 1, 2, 3$.

We now denote $\mathsf{Root}(y, p, q)$ the function mapping any $y \in \mathbf{Z}_N^*$ to the set of all its cubic roots using the secret key. This function will be used throughout this problem.

Complexity of Cubic Roots

1. If $x, y \in \mathbf{Z}_N^*$ are such that $x \not\equiv y \pmod N$ and $x^3 \equiv y^3 \pmod N$, show that $\mathsf{gcd}(x - y, N) = q$.

The two conditions means that x and y are two *different* roots of the *same* cubic residue.
Using the CRT theorem and using the fact that there is only one root in $\mathbf{Z}_q^*$, i.e. $x \equiv y \bmod q$, we deduce that $x \not\equiv y \bmod p$.
Thus, q divides $x - y$ but p does not divides $x - y$ and finally we obtain that $\mathsf{gcd}(x - y, pq) = q$.

2. Deduce that an oracle who can extract one cubic root from a cubic residue in $\mathbf{Z}_N^*$ can be used to factor N.

The adversary simply picks a $x \in \mathbf{Z}_N^*$ and sumbits $y = x^3$ to the oracle. The oracle returns a root x' of y. Since there is three roots, with probability $2/3$, x' is different of x and thus we can factorize N using the previous result, i.e. by computing $\mathsf{gcd}(x - x', N) = q$.

Raw Encryption and Decryption

We consider the message space $\mathbf{Z}_N^*$. Encryption is made as in RSA, by raising to the power e modulo N.

1. Show that decryption is ambiguous.

Let x be the plaintext. The cyphertext is $y = x^3 \bmod N$.

We can decrypt the ciphertext using the secret key by computing

$$x = \mathsf{Root}(y, p, q)$$

Note that the function Root returns 3 values and thus the decryption is not deterministic.

2. Devise a chosen ciphertext attack.

In a ciphertext attack, we suppose that we have access to a decryption oracle. In our case, the decryption oracle returns a cubic roots of the ciphertext. Thus, we can use this oracle to perform the same attack as before, i.e. : The adversary picks $x \in \mathbf{Z}_N^*$ and sumbits $y = x^3$ to the decryption oracle. The decryption oracle returns a plaintext x'. With probability $2/3$, x' is different of x and we have two different roots of y. We can factorize N by computing $\mathsf{gcd}(x - x', N)$ which yields q.

Encryption and Decryption on a Reduced Space

Let n be such that $2^n \ll N$. Let F be a random injection from $\{0,1\}^n$ to $\mathbf{Z}_N^*$ which is easy to invert. We now consider the message space $\{0,1\}^n$. We define the encryption of x by $F(x)^e \bmod N$.

1. How can we decrypt now?

Let x be the plaintext. The cyphertext is $y = F(x)^3 \bmod N$.
We can decrypt the ciphertext using the secret key by computing

$$x = F^{-}\left[\mathsf{Root}(y,p,q)\right]$$

Note that the function Root returns 3 values, let it $F(x), \alpha, \beta$. If the $F^{-1}(\alpha)$ and/or $F^{-1}(\beta)$ is/are defined, the decryption is not deterministic.

2. What is the probability (over the choice of F) that there exists x such that decrypting the encryption of x does not produce x?

Using the same notations as before, we have to compute the probability that $F^{-1}(\alpha)$ and/or $F^{-1}(\beta)$ exist.

$$\begin{aligned} p &= \Pr[F^{-1}(x') \text{ and/or } F^{-1}(x') \text{ is defined}] \\ &= 1 - \Pr[F^{-1}(x') \text{ and } F^{-1}(x') \text{ are not defined}] \\ &\approx 1 - \Pr[F^{-1}(x') \text{ is not defined}]^2 \\ &= 1 - \left(1 - \Pr[F^{-1}(x') \text{ is defined}]\right)^2 \\ &\approx 1 - \left(1 - \frac{2^n}{N}\right)^2 \end{aligned}$$

3. Show that key recovery is equivalent to factoring numbers like N.

We saw that with two roots we can factorize N. Since no other root than x is defined (with very high probability), we cannot use the gcd tricks and we have to factorize N to recover the secret key p or q.

4. What can we now say about the decryption problem?

Now, the decrpytion is not ambiguous (with very high probability) since for each cubic residue only one root is defined (with very high probability).

5. Give at least one Boolean function on the plaintext that is not a hard core bit.

We use the Jacobi relation between plaintext/ciphertext pairs that exists for textbook RSA. Indeed, we can write

$$\left(\frac{y}{N}\right) = \left(\frac{F(x)^e}{N}\right) = \left(\frac{F(x)}{N}\right)^e = \left(\frac{F(x)}{N}\right)$$

Probabilistic Variant

Let n and k be integers such that $k < n$. We now consider that the message space is a binary code of length n and dimension k. We consider a symmetric encryption scheme over the plaintext/ciphertext space $\{0,1\}^n$ and keyspace $\mathcal{K}$ defined by SymEnc and SymDec algorithms. Let H be a random function from $\mathbf{Z}_N^*$ to $\mathcal{K}$. To encrypt a codeword x, we first pick a random $r \in \mathbf{Z}_N^*$ and we compute $y = \mathsf{SymEnc}_{H(r)}(x)$ and $z = r^e \bmod N$. The ciphertext is (y, z).

1. How to decrypt?

(a) We first have to recover r by using the secret key, i.e., $r = \mathsf{Root}(z, p, q)$. Note that we have three candidates for r, i.e. $\hat{r}, r', r''$ where $\hat{r}$ is the true candidate.

(b) For each candidate of r, we compute $x = \mathsf{SymDec}_{H(r)}(y)$ and we obtain three candidates for x, i.e. $\hat{x}, x', x''$ where $\hat{x}$ is the true candidate.

(c) We discard each candidate for x that is not a codeword.

2. Assuming that the symmetric encryption is ideal and that $\mathcal{K}$ is large enough, what is the probability that decryption is ambiguous?

Let $\hat{x}$, x', x'' the candidates for x in the previous question.

$$\begin{aligned}
\Pr[\text{ambiguous decryption}] &= \Pr[x' \text{ or } x'' \text{ is a codeword}]\\
&= 1 - \Pr[x' \text{ and } x'' \text{ are not codewords}]\\
&= 1 - \Pr[x' \text{ is not a codeword}]^2\\
&= 1 - \big(1 - \Pr[x' \text{ is a codeword}]\big)^2\\
&= 1 - \left(1 - \frac{2^k}{2^n}\right)^2
\end{aligned}$$

The last equality occurs since the symmetric encryption is ideal. Indeed, using a wrong key, i.e. $H(r')$ or $H(r'')$, we obtain uniformly distributed random n-bit strings x' or x''. Note that a random n-bit string is a codeword with probability $\frac{2^k}{2^n}$.

3. Recall what is an adversary against the semantic security.

It's an adversary plaing the following game:

$$\begin{array}{ccc}
\mathcal{A} & & \mathcal{C}\\
 & \xleftarrow{\qquad K_p \qquad} & (K_p, K_s) \leftarrow \mathsf{setup}()\\
 & \xrightarrow{\qquad x_0, x_1 \qquad} & \text{pick } b \in_U \{0,1\}\\
\hat{b} \leftarrow \text{guess } b & \xleftarrow{\qquad y \qquad} & y \leftarrow \mathsf{Enc}(K_s, x_b)
\end{array}$$

Winning condition: $\hat{b} = b$

4. Assume that we have an adversary $\mathcal{A}$ playing the semantic security game against our new cryptosystem. We assume that the symmetric encryption scheme is an ideal cipher, that is, $H(r)$ fully specifies a random permutation over $\{0,1\}^n$. We further assume that function H is only available through an oracle $\mathcal{O}$, that is, nobody can reliably compute $H(r)$ without querying the oracle $\mathcal{O}$ with r to get $H(r)$ in return. This way, $\mathcal{A}$ may query the oracle $\mathcal{O}$ while playing the semantic security game.

   (a) Show that if $\mathcal{A}$ does not query $\mathcal{O}$ with the r chosen by the challenger, the advantage of $\mathcal{A}$ in the semantic game is zero.

   The result is straightforward. Since $\mathcal{A}$ does not query $\mathcal{O}$ with r, he uses a wrong key for the decryption, i.e. with the SymDec algorithm. Tanks to the ideal decryption, we know that it returns a uniformly distributed random n-bit string $\bar{x}$ and thus $\mathcal{A}$ has no information on which x_i was encrypted. The proabaility that $\mathcal{A}$ guesses the right b is $1/2$ and we conclude that his advantage is 0.

   (b) By simulating $\mathcal{O}$ and several parts of the semantic game, deduce that if the advantage of $\mathcal{A}$ is ε, we can transform $\mathcal{A}$ in an algorithm which given $z = r^3 \bmod N$ for a random r can deduce r or other cubic roots of z with probability ε.

   If the adversary has an advantage of ε, this means that he queried $\mathcal{O}$ wwith the right r with probability ε (otherwise its advantage would be 0). We conclude that the adversary given (y, z) recovered r with probabiltiy ε which means that he found a root of z.

(c) Deduce that if the advantage of $\mathcal{A}$ is ε, we can factor N with probability ε.

Using the previous result, we know that $\mathcal{A}$ can deduce a cubic root r' from z. Now, runnig a ciphertext attack, i.e. we pick a r, comput $z = r^3$, and use $\mathcal{A}$ to find a root r' of z, we obtain another root of z with probability $2/3$. Using the gcd algorithm and the two roots as before, we can deduce q and thus factorize N.

(d) Deduce that if factoring N is hard, if the symmetric encryption is ideal, and if H is a random oracle, this cryptosystem is semantically secure.

Considering all above assumptions, we note that ε becomes very close to 0 and thus the cryptosystem is semantically secure.