Advanced Cryptography — Midterm Exam
Solution

Serge Vaudenay

18.4.2019

- duration: 1h45
- any document allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will **<u>not</u>** answer any technical question during the exam
- readability and style of writing will be part of the grade

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

1 On Various Equivalent Indistinguishability Notions

In this exercise, we consider two games $\Gamma_0(1^s)$ and $\Gamma_1(1^s)$ which can be played by an adversary $\mathcal{A}$. We assume that Γ_0 and Γ_1 are such that they output c if and only if $\mathcal{A}$ outputs a final message c. We define

$$\begin{aligned}\mathsf{Adv}_1^{\mathcal{A}}(s) &= \Pr[\Gamma_1(1^s,\mathcal{A}) \to 1] - \Pr[\Gamma_0(1^s,\mathcal{A}) \to 1] \\ \mathsf{Adv}_2^{\mathcal{A}}(s) &= |\Pr[\Gamma_1(1^s,\mathcal{A}) \to 1] - \Pr[\Gamma_0(1^s,\mathcal{A}) \to 1]| \\ \mathsf{Adv}_3^{\mathcal{A}}(s) &= \frac{1}{2} - \Pr[\Gamma'(1^s,\mathcal{A}) \to 1]\end{aligned}$$

where Γ' is a bit-guessing game defined by

Game $\Gamma'(1^s,\mathcal{A})$:

```
1: picks $b \in \{0,1\}$ uniformly at random
2: if $b = 0$ then
3:     simulate $\Gamma_0(1^s,\mathcal{A})$ which returns $c$
4: else
5:     simulate $\Gamma_1(1^s,\mathcal{A})$ which returns $c$
6: end if
7: $c' = 1_{c=1}$                              ▷ this forces $c'$ to be 0 or 1
8: return $1_{b=c'}$
```

Given a positive function $g(s)$, we define three notions of g-indistinguishability by

g-IND_i: "for any p.p.t. algorithm $\mathcal{A}$, $\exists s_0 \quad \forall s \geq s_0 \quad \mathsf{Adv}_i^{\mathcal{A}}(s) \leq g(s)$"

Q.1 Prove that g-IND_1 is equivalent to g-IND_2.
Warning: there are two directions in an equivalence!

Clearly, if $|f(s)| \leq g(s)$, then $f(s) \leq g(s)$. Since $\mathsf{Adv}_2^{\mathcal{A}} = |\mathsf{Adv}_1^{\mathcal{A}}|$, we deduce that g-$\mathsf{IND}_2 \Longrightarrow g$-$\mathsf{IND}_1$.
Given $\mathcal{A}$, we define a new adversary $\mathcal{B}$ which essentially simulates $\mathcal{A}$ until $\mathcal{A}$ returns c, in which case $\mathcal{B}$ returns $1_{c \neq 1}$. Clearly, $\mathsf{Adv}_1^{\mathcal{B}} = -\mathsf{Adv}_1^{\mathcal{A}}$.
If g-IND_1 holds, then $\mathsf{Adv}_1^{\mathcal{B}}$ must be lower than $g(s)$ for sufficiently large s, which means that $-\mathsf{Adv}_1^{\mathcal{A}}$ is lower than $g(s)$ for sufficiently large s.
If $f(s) \leq g(s)$ and $f'(s) \leq g(s)$, then $\max(f(s), f'(s)) \leq g(s)$. Since $\mathsf{Adv}_2^{\mathcal{A}} = \max(\mathsf{Adv}_1^{\mathcal{A}}, -\mathsf{Adv}_1^{\mathcal{A}})$, then $\mathsf{Adv}_2^{\mathcal{A}}(s) \leq g(s)$ for large enough s. Hence, g-IND_2 holds as well.
***Feedback from the exam.** Some students made mistakes with inequalities such as "since $\mathsf{Adv}(s) \leq g(s)$ and $g(s)$ is positive, we deduce $|\mathsf{Adv}(s)| \leq g(s)$".*

Q.2 Prove that g-IND_1 is equivalent to $\frac{g}{2}$-IND_3.

We have that

$$\Pr[\Gamma' \to 1] = (1 - \Pr[\Gamma_0 \to 1]) \Pr[b = 0] + \Pr[\Gamma_1 \to 1] \Pr[b = 1] = \frac{1}{2} - \frac{1}{2}\mathsf{Adv}_1^{\mathcal{A}}$$

hence $\mathsf{Adv}_3^{\mathcal{A}} = \frac{1}{2}\mathsf{Adv}_1^{\mathcal{A}}$. We deduce that g-IND_1 and $\frac{g}{2}$-IND_3 are equivalent.
***Feedback from the exam.** A majority of students wrote proofs by contraposition and things like "assume it is not *-secure so we have an adversary playing the *-game with non-negligible advantage". First of all, writing "non-negligible" must be avoided because in the literature, there is a notion of what non-negligible means, and it is not the negation of being negligible... Better write "not negligible". Second, manipulating negation of the negligible notion leads to big logical mistakes in a vast majority of cases so it is safer to avoid to negate it. The reason of mistakes comes from the negligible notion being defined by a chain of different quantifiers: $f(s)$ is negligible if for all $c > 0$, we have $f(s) = \mathsf{O}(\frac{1}{s^c})$. This is equivalent to*

$$\forall c > 0 \quad \exists s_0 \quad \forall s > s_0 \quad f(s) < \frac{1}{s^c}$$

The negation of it is

$$\exists c > 0 \quad \forall s_0 \quad \exists s > s_0 \quad f(s) \geq \frac{1}{s^c}$$

Writing $f(s) > \mathsf{O}(\frac{1}{s^c})$ does not make much sense and would not lead to the above meaning almost surely. It is advised to avoid proofs by contraposition. In all cases we have seen in the exam, the proof could be written without negations.

2 Goldwasser-Micali Cryptosystem

We define the GM cryptosystem over the message space $\{0,1\}$ as follows:

```
Gen(1^s):
 1: generate two different prime numbers $p$ and $q$ of $s$ bits
 2: $N = pq$
 3: pick $x \in \mathbf{Z}_N^*$ such that $(x/p) = (x/q) = -1$
 4: $\mathsf{pk} = (x, N)$, $\mathsf{sk} = p$
 5: return pk and sk
Enc(pk, b):
 6: parse $\mathsf{pk} = (x, N)$
 7: pick $r \in \mathbf{Z}_N^*$ uniformly at random
 8: $\mathsf{ct} = r^2 x^b \bmod N$
 9: return ct
Dec(sk, ct):
10: set $p = \mathsf{sk}$
11: $\sigma = (\mathsf{ct}/p)$
12: return $1_{\sigma=-1}$
```

Q.1 Prove that GM is public-key cryptosystem and that it is correct.
Hint: triple-check all what you must prove in this question!

We first observe that we have a tuple $(\mathsf{Gen}, \{0,1\}, \mathsf{Enc}, \mathsf{Dec})$ *as in the definition of a PKC. We can see that the algorithms are* polynomially bounded*. We can see that* Dec *is* deterministic*. What remains to be proven is the* correctness *property.*
Assume that we generated p, q, N, x as specified and that we encrypted some b with r as specified. During decryption, we have

$$\sigma = \left(\frac{\mathsf{ct}}{p}\right) = \left(\frac{r^2x^b}{p}\right) = \left(\frac{r}{p}\right)^2 \left(\frac{x}{p}\right)^b = (-1)^b$$

Hence, decryption returns $1_{(-1)^b=-1} = b$. The PKC is correct.
Feedback from the exam. *Almost all students made correctness right but a really small minority scored all points in this question. Almost everyone forgot to say that decryption is deterministic and many forgot to say that algorithms are polynomially bounded.*

Q.2 Prove that the key-recovery problem (KR-CPA) is equivalent to some well-known problem.

Here is the KR-CPA game:

Game KR-CPA(1^s)
```
1: $\mathsf{Gen}(1^s) \rightarrow (\mathsf{pk}, \mathsf{sk})$
2: $\mathcal{A}(\mathsf{pk}) \rightarrow k$
3: return $1_{k=\mathsf{sk}}$
```

It means that given x and N, $\mathcal{A}$ must recover p. Clearly, recovering p implies factoring N completely since we can easily divide N by p.
Similarly, the factoring problem is:

Game FACT(1^s)
```
1: generate two different prime numbers $p$ and $q$ of $s$ bits
2: $N = pq$
3: $\mathcal{A}(N) \rightarrow F$
4: return $1_{F=\{p,q\}}$
```

Factoring N with probability ρ implies recovering p, at least with probability $\frac{\rho}{2}$. Hence, if KR-CPA is hard, $\frac{\rho}{2}$ is negligible, hence ρ is negligible. Hence, KR-CPA security implies the hardness of factoring.
Conversely, if factoring is hard, let us take a KR-CPA adversary $\mathcal{A}$ with advantage ρ. We can pick a random x such that $(x/N) = +1$. With probability $\frac{1}{2}$, (x, N) forms a public key with correct distribution and we can feed $\mathcal{A}$ with (x, N) and check if it returns the factorization of N. If (x, N) does not form a valid public key, it is not clear what happens. Nevertheless, it factors with probability at least $\frac{\rho}{2}$. If factoring is hard, $\frac{\rho}{2}$ is negligible, hence ρ is negligible. Hence, the hardness of factoring implies KR-CPA security.
Finally, KR-CPA security is equivalent to the hardness of factoring.
Feedback from the exam. *Some students did not see there is an x to provide to the KR-CPA oracle. Many students did this question correct but forgot to discuss what happens if the x which is provided is an incorrect public key (namely: if it is a quadratic residue).*

Q.3 We define the following game which depends on a bit b:

Game $\Gamma_b(1^s, \mathcal{A})$:
```
1: $\mathsf{Gen}(1^s) \rightarrow (\mathsf{pk}, \mathsf{sk})$
2: $\mathsf{Enc}(\mathsf{pk}, b) \rightarrow \mathsf{ct}$
3: $\mathcal{A}(\mathsf{pk}, \mathsf{ct}) \rightarrow c$
4: return $c$
```

We say that GM is Γ-secure if for every p.p.t. $\mathcal{A}$, $\Pr[\Gamma_1(1^s, \mathcal{A}) \rightarrow 1] - \Pr[\Gamma_0(1^s, \mathcal{A}) \rightarrow 1]$ is a negligible function of s.

Prove that IND-CPA security and Γ-security are equivalent for GM.

The message domain is $\{0,1\}$. IND-CPA adversaries proposing the pair of messages to be $m_0 = m_1$ are not interesting because their advantage is always 0. Any adversary proposing the pair $m_0 = 1$ and $m_1 = 0$ has the opposite advantage of the one doing the same but proposing $m_0 = 0$ and $m_1 = 1$. Hence, we can focus without loss of generality on an adversary proposing $m_0 = 0$ and $m_1 = 1$. The IND-CPA game becomes equivalent to Γ.

More precisely, any adversary $\mathcal{A}$ playing the Γ-game can be transformed into an adversary $\mathcal{B}$ playing the IND-CPA game with same advantage: in the first part of $\mathcal{B}$, the adversary proposes the challenge plaintexts $(0,1)$. In the second part of $\mathcal{B}$, he does exactly like $\mathcal{A}$. Conversely, any adversary $\mathcal{B}$ playing the IND-CPA game can be transformed into an adversary $\mathcal{A}$ playing the Γ-game:

```
Adversary $\mathcal{A}(\mathsf{pk}, \mathsf{ct})$
1: $\mathcal{B}_1(\mathsf{pk}) \to (\mathsf{pt}_0, \mathsf{pt}_1, \mathsf{st})$
2: if $\mathsf{pt}_0 = \mathsf{pt}_1$ then
3:     pick $z \overset{\$}{\in} \{0,1\}$
4: else
5:     $\mathcal{B}_2(\mathsf{pk}, \mathsf{ct}) \to z_0$
6:     $z \leftarrow z_0 \oplus \mathsf{pt}_0$          ▷ flip $z$ if challenge plaintexts were $(1,0)$
7: end if
```

We let E_{uv} be the event $[\mathsf{pt}_0 = u, \mathsf{pt}_1 = v]$. For any u, we have $\Pr[\mathsf{INDCPA}_1(1^s, \mathcal{B}) \to 1 | E_{uu}] = \Pr[\mathsf{INDCPA}_0(1^s, \mathcal{B}) \to 1 | E_{uu}]$ because the outcome is independent from b (the encrypted plaintext is always u, no matter the value of b). We have $\Pr[\Gamma_1(1^s, \mathcal{A}) \to 1 | E_{uu}] = \Pr[\Gamma_0(1^s, \mathcal{A}) \to 1 | E_{uu}] = \frac{1}{2}$. Furthermore, we have that for all b, $\Pr[\Gamma_b(1^s, \mathcal{A}) \to 1 | E_{01}] = \Pr[\mathsf{INDCPA}_b(1^s, \mathcal{B}) \to 1 | E_{01}]$ and $\Pr[\Gamma_b(1^s, \mathcal{A}) \to 1 | E_{10}] = 1 - \Pr[\mathsf{INDCPA}_{1-b}(1^s, \mathcal{B}) \to 1 | E_{10}]$. Hence,

$$
\begin{aligned}
&\mathsf{Adv}_{\mathcal{A}} \\
&= \Pr[\Gamma_1(1^s, \mathcal{A}) \to 1] - \Pr[\Gamma_0(1^s, \mathcal{A}) \to 1] \\
&= \sum_{u,v} \left(\Pr[\Gamma_1(1^s, \mathcal{A}) \to 1 | E_{uv}] - \Pr[\Gamma_0(1^s, \mathcal{A}) \to 1 | E_{uv}]\right) \Pr[E_{u,v}] \\
&= \sum_{u,v} \left(\Pr[\mathsf{INDCPA}_1(1^s, \mathcal{B}) \to 1 | E_{uv}] - \Pr[\mathsf{INDCPA}_0(1^s, \mathcal{B}) \to 1 | E_{uv}]\right) \Pr[E_{u,v}] \\
&= \mathsf{Adv}_{\mathcal{B}}
\end{aligned}
$$

Feedback from the exam. *The difficulty in this question is to write down the reduction correctly. We must explain why an IND-CPA adversary giving equal challenge plaintexts would be "stupid" (students said so without explaining). Actually, "stupidity" should clearly come from the math by having a null advantage. We must also explain that if the adversary gives $(1,0)$ instead of $(0,1)$, we can just flip the answer to reduce to a Γ adversary in all cases.*

Q.4 We define the following game which depends on a bit b:

Game $\mathsf{QR}_b(1^s, \mathcal{A})$:

```
1: generate two different prime numbers $p$ and $q$ of $s$ bits
2: $N = pq$
3: pick $x \in \mathbf{Z}_N^*$ such that $(x/p) = (x/q) = (-1)^b$
4: $\mathcal{A}(x, N) \to c$
5: **return** $c$
```

We define $\mathsf{Adv}^{\mathcal{A}}(s) = \Pr[\mathsf{QR}_1(1^s, \mathcal{A}) \to 1] - \Pr[\mathsf{QR}_0(1^s, \mathcal{A}) \to 1]$. We say that the QR problem is hard if for every p.p.t. $\mathcal{A}$, $\mathsf{Adv}^{\mathcal{A}}$ is a negligible function.
Prove that the IND-CPA security of GM implies the QR hardness.

Let consider a QR adversary $\mathcal{A}$. Following QR, $\mathcal{A}$ has input (y, N).
We define $\mathcal{B}(x, N, y) = \mathcal{A}(y, N)$ to play the Γ game. In Γ, we have $y = r^2x^b$ which has the same distribution as x in the QR game. Hence, $\mathcal{A}$ and $\mathcal{B}$ have the same advantage. If GM is IND-CPA secure, then it is Γ-secure, thus the advantage of $\mathcal{B}$ must be negligible. We deduce that the advantage of $\mathcal{A}$ is negligible as well. Hence, QR is hard.
Feedback from the exam. *Again, we had many proofs by contraposition which should be avoided. The only trick was to use Γ-security (hence the result from last question). Not using it implies redoing somehow the equivalence in the last question.*

Q.5 Prove that the IND-CPA security of GM is equivalent to the hardness of QR.

What remains to be proven is that if the QR problem is hard, then GM is IND-CPA secure. We know that IND-CPA security and Γ security are equivalent. So, we can just prove the Γ security under the assumption that QR is hard.
Let consider a Γ adversary $\mathcal{A}$. Following Γ, $\mathcal{A}$ has input (x, N, r^2x^b). We define a QR adversary as follows:

$\mathcal{B}(y, N)$:

```
1: pick a random $x \in \mathbf{Z}_N^*$ such that $(x/N) = +1$
2: $\mathcal{A}(x, N, y) \to c$
3: **return** $c$
```

Let E be the event that x is not a quadratic residue. When E holds (this happens with probability $\frac{1}{2}$), then (x, N, y) has the same distribution at in the Γ game. When $\neg E$ holds, the distribution is the same as in the following QR adversary:

$\mathcal{C}(y, N)$:

```
1: pick a random $r \in \mathbf{Z}_N^*$
2: $x = r^2 \bmod N$
3: $\mathcal{A}(x, N, y) \to c$
4: **return** $c$
```

Hence, $\Pr[\mathsf{QR}_b(1^s, \mathcal{B}) \to 1] = \frac{1}{2}\Pr[\Gamma_b(1^s, \mathcal{A}) \to 1] + \frac{1}{2}\Pr[\mathsf{QR}_b(1^s, \mathcal{C}) \to 1]$. *Therefore,* $\mathsf{Adv}^{\mathcal{B}} = \frac{1}{2}\mathsf{Adv}^{\mathcal{A}} + \frac{1}{2}\mathsf{Adv}^{\mathcal{C}}$. *If QR is hard, we know that both* $\mathsf{Adv}^{\mathcal{B}}$ *and* $\mathsf{Adv}^{\mathcal{C}}$ *are negligible. Hence,* $\mathsf{Adv}^{\mathcal{A}}$ *is negligible as well.*

3 A Weird Signcryption

We consider the plain RSA cryptosystem ($\mathsf{RSA.Gen}, \mathsf{RSA.Enc}, \mathsf{RSA.Dec}$) and a digital signature scheme ($\mathsf{DS.Gen}, \mathsf{DS.Sign}, \mathsf{DS.Ver}$). We construct a *signcryption* scheme as follows:

```
SC.Gen:                                              ▷ generate a key pair for a user
 1: RSA.Gen → (ek, dk)                               ▷ encryption key and decryption key
 2: DS.Gen → (sk, vk)                                ▷ signing key and verification key
 3: pubk ← (ek, vk)                                  ▷ public key of user
 4: privk ← (dk, sk)                                 ▷ private key of user
 5: return (pubk, privk)
SC.Send($\mathsf{pubk}_B$, $\mathsf{privk}_A$, pt):  ▷ user $A$ sends a message to user $B$
 6: parse $\mathsf{pubk}_B = (\mathsf{ek}_B, \mathsf{vk}_B)$
 7: parse $\mathsf{privk}_A = (\mathsf{dk}_A, \mathsf{sk}_A)$
 8: $\mathsf{ct} \leftarrow \mathsf{RSA.Enc}(\mathsf{ek}_B, \mathsf{pt})$
 9: $\sigma \leftarrow \mathsf{DS.Sign}(\mathsf{sk}_A, \mathsf{ct})$
10: return $(\mathsf{ct}, \sigma)$
```

so that A can send (ct, σ) to B. Once B obtains pt, he can show $\mathsf{proof} = (\mathsf{vk}_A, \mathsf{ek}_B, \mathsf{ct}, \sigma, \mathsf{pt})$ as a proof that A sent pt. We call this property *non-repudiation.*

Q.1 Describe the algorithm using $(\mathsf{pubk}_A, \mathsf{privk}_B)$ to receive (ct, σ) and compute pt, as well as the algorithm to verify the proof.

To receive the message, we use:

```
SC.Receive($\mathsf{pubk}_A$, $\mathsf{privk}_B$, ct, $\sigma$):
 1: parse $\mathsf{pubk}_A = (\mathsf{ek}_A, \mathsf{vk}_A)$
 2: parse $\mathsf{privk}_B = (\mathsf{dk}_B, \mathsf{sk}_B)$
 3: if DS.Ver($\mathsf{vk}_A$, ct, $\sigma$) then return $\perp$
 4: $\mathsf{pt} \leftarrow \mathsf{RSA.Dec}(\mathsf{dk}_B, \mathsf{ct})$
 5: return pt
```

To verify the proof:

```
SC.Verify($\mathsf{vk}_A$, $\mathsf{ek}_B$, ct, $\sigma$, pt):
 1: if DS.Ver($\mathsf{vk}_A$, ct, $\sigma$) then return reject
 2: if $\mathsf{RSA.Enc}(\mathsf{ek}_B, \mathsf{pt}) \neq \mathsf{ct}$ then return reject
 3: return accept
```

Feedback from the exam. *Several students wrote a* SC.Receive *which always returns the decryption of the ciphertext no matter if the signature is correct. They missed the point of having the ciphertext authenticated. It is not a good idea to return any information is the message is not well formed. Many students did not understand what they were supposed to do: they gave as* SC.Receive *a decryption algorithm and as* SC.Verify *a signature verification algorithm.*

Q.2 Given $(\mathsf{vk}_A, \mathsf{ct}, \sigma)$ such that $\mathsf{DS.Ver}(\mathsf{vk}_A, \mathsf{ct}, \sigma)$ is true and given an arbitrary pt, prove that we can easily find ek such that $(\mathsf{vk}_A, \mathsf{ek}, \mathsf{ct}, \sigma, \mathsf{pt})$ is a valid proof.

ek *consists of an exponent* e *and a modulus* N. *We can fix* e *arbitrarily and set* $N = \mathsf{pt}^e - \mathsf{ct}$. *We obtain* $\mathsf{RSA.Enc}(\mathsf{ek}, \mathsf{pt}) = \mathsf{ct}$. *Hence,* $(\mathsf{vk}_A, \mathsf{ek}, \mathsf{ct}, \sigma, \mathsf{pt})$ *is a valid proof.*

Q.3 Propose a fix to this problem so that we have non-repudiation.

One way could be to sign $(\mathsf{ct}, \mathsf{pubk}_B)$ *instead of* ct *alone. This way, the adversary would not be able to change* ek *a posteriori.*
Feedback from the exam. *Some students proposed to sign the plaintext instead of the ciphertext. One problem is that with the currently proposed structure, the signature is given in clear, which means that it leaks information about the plaintext. Clearly, we have no IND-CPA security this way. We could instead sign the plaintext and encrypt the plaintext and its signature. But there may be reasons why would like message reception to start with a signature verification. For instance, it could avoid malicious access to the decryption key and protect more against side channels.*