1) SPA of RSA Using CRT

a) How can we learn the secret key?

When x is zero, the equation is trivially false, and the most significant bits of x are trivially zero.

We can now increment x and look at the equation every time.

- If $p > q$, then the rhs of the equation will wrap around to zero earlier, making the equation true. Since this will happen exactly when $x = q$, we learned q.
- If $p < q$, then the lhs will wrap around earlier, leaving the equation false. We keep incrementing x until the rhs wraps around. Since the lhs wrapped earlier, it is now larger, making the equation true. Since this happenes exactly when $x = q$, we learned q.

Because $\log_2 p \approx \log_2 q$ and $n = p \times q$, the first half of the bits of x should still be zero at this point.

From n and q we can calculate the missing p and d.

b) Can padding verification protect against active attacks?

I assume yes, since arbitrary changes during decryption will probably invalidate the padding.

2) No Covert Channel?

a) Can you still think of a covert channel?

The number/frequency of failing computers, and thus of replacement purchases, could be used to extract information over long times. This requires malware on all computers in the datacenter.

E.g. a bit could be transmitted by breaking/buying two/three computers at once for a 0 or 1.

b) What is the capacity of this channel?

$\frac{1b}{25\,\text{years}} \times computers$ to not attract attention (due to the average failing rate), or $\frac{1b}{10\,\text{years}} \times computers$ if we can afford suspicion.

3) PINs, Pollen, Probability (Part II)

I assume this means that a number appears twice in the PIN.

a) How many trials without prior knowledge?

$\frac{5!\times 4}{2\times 2} = \frac{480}{2} = 120$ attempts on average: $5!$ for the permutations, this 4 times because we don't know which key was reused.

b) What can be achieved using pollen?

The average number of tries is $\frac{(4!\times 2)\times 4+(2!\times 2)\times 4+(3!\times 2)\times 4}{5*2} = \frac{256}{10} = 25.6$

- Dirty key is first or last: $4!$ permutations
- Dirty key is second or fourth: $3!$ permutations
- Dirty key is middle: $2!$ permutations

However, this is not yet adapted to the fact that we learn less information because one key is reused…

c) What to do if only 3 keys are worn off?

Two cases:

- different keys re-used: $\binom{5\times 3\times 3}{2,2} = \frac{5!\times 9}{4} = 270$ combinations, so 135 attempts on average.
- same key re-used: $\binom{5\times 3}{3} = \frac{5!\times 9}{3!} = 60$ combinations, so 30 attempts on average.

This normalizes to $\frac{135+30}{2} = 82.5$ attempts on average.

It may now be better to only clean one key.

4) Game Theory

Utility functions:

$u1(ag, ag) = 0$

$u1(ag, mc) = 0.2 \times 0.25 \times -7$ (1 gets fired only if forensics don't detect 2)

$u1(mc, ag) = 1.5 + 0.2 \times -7$ (1 gets fired no matter what forensics show)

$u1(mc, mc) = 1.5 + 0.2 \times -7 + 0.2 \times 0.25 \times -7$ (assuming compromise of the keys is independent)

$u2(ag, ag) = 0$

$u2(ag, mc) = 1.5 + 0.2 \times -7$

$u2(mc, ag) = 0.2 \times 0.25 \times -7$

$u2(mc, mc) = 1.5 + 0.2 \times -7 + 0.2 \times 0.25 \times -7$

Matrix form:

	2 air-gapped	2 minecraft
1 air-gapped	$(0, 0)$	$(-0.35, 0.1)$
1 minecraft	$(0.1, -0.35)$	$(-0.25, -0.25)$

Nash Equilibria With pure strategies, both playing minecraft, because switching won't help an individual admin. Prisoner's dilemma doesn't have additional nash equilibria with mixed strategies.

Social Optima The social optimum is for both admins to air-gap their system.

Which game applies? The prisoners dilemma: Social optimum if both air-gap, but (slight) advantage if one is the only one to play minecraft.

I would definitely not apply for this job, the chance of getting fired because my colleague messed up is too high.