Part A: RSA

1. A correctness proof of RSA shows that $c = m^e(ModN)$ then $m = c^d(ModN)$ using these steps:

   - $c = m^e(ModN)$
   - $m = c^d(ModN)$
   - $m = (m^e)^d(ModN)$
   - $m = m^{ed}(ModN)$
   - $ed = 1(Mod\Phi(N)) = 1 + k\Phi(N)$
   - $m = m^{1+k\Phi(N)}(ModN)$
   - $m = mm^{\Phi(N)^k}(ModN)$
   - Euler's Theorem: $a^{\Phi(N)} = 1(ModN)$
   - $m = m1^k(ModN)$
   - $m = m(ModN)$

2. Using gmpy we can calculate the message like this:

```
N = gmpy.mpz(26085133416023792110786950746751186556 9);
d = gmpy.mpz(11419990338673736177884281093720685329 1);
c = gmpy.mpz(25659792217239235040146736902131445688 5);

m = pow(c,d,N);
```

3. Now suppose you were given the following:

   $\Phi(N) = 260851334160237921075365462971131061444$

   What are p, q and e?

   (Hint: can you write a quadratic equation with p and q as the roots?)

```
phi = gmpy.mpz(260851334160237921075365462971131061444);
a = gmpy.mpz(1);
b = -(N-phi+1);
c = N;

p = (-b+gmpy.sqrt(pow(b,2)-4*a*c))/(2*a);
q = (-b-gmpy.sqrt(pow(b,2)-4*a*c))/(2*a);

(a0,e,a1) = gmpy.gcdext(d,phi);
e = e%phi;
```

Part B: Vinegar Cipher

1. The Vinegar cipher performs these substitutions:

i	0	1	2	3	4	5	6	7
0	2	3	4	5	6	7	0	1
1	3	4	5	6	7	0	1	2
2	5	6	7	0	1	2	3	4

   where i is the position in each block of 3 letters.

   (a) `123 452 034 521` is mapped to `350 607 261 756`.

(b) The cipher preserves frequency distributions of symbols in the plaintext. Over longer plaintexts, we can use this for cryptanalysis: determine the block size by finding repeated patterns in the ciphertext, then consider block positions independently, using frequency analysis.

(c) The only way to make an "unbreakable" cipher is using a one-time pad: use a key (and block-size) as long as the message. The disadvantage is that the key must be random and never re-used.

Part C: DLP and ElGamal

1. (a) $Z^*[10] = \{1, 3, 7, 9\}$ and the powers of its elements are:

$a \in Z^*[10]$	1	3	7	9
a^2	1	9	9	1
a^3	1	7	3	9
a^4	1	1	1	1

thus 3 and 7 are generators

(b) The security of *Diffie-Hellman key agreement* relies on the difficulty of solving the discrete logarithm problem. Modular exponentiation is an example of a *(candidate) one-way function*

(c) i. The immediate disadvantage of the ElGamal scheme compared to RSA is that the ciphertext twice as long as the plaintext (two numbers instead of one).

ii. Because of the difficulty of the DLP, we might suppose that an eavesdropper Eve cannot recover the random number r from $g = \alpha^r \bmod p$ nor of course Bob's private key from $\beta = \alpha^k \bmod p$. Because r is random, so is β^r (a random number raised to a random power), and the message multiplied by a random number is random again; therefore we can suppose that $m\beta^r$ gives Eve no knowledge about m.

iii. Suppose Alice re-uses the same r for a further message m', sending $E(m') = (g, d')$. If Eve has managed to find out the plaintext m then she can find out m' too as $m' = d'md^{-1} \bmod p$, which works because

$$d'md^{-1} \equiv (m'\beta^r)m(m\beta^r)^{-1} \equiv (m'\beta^r)mm^{-1}(\beta^r)^{-1} \equiv m' \pmod{p}.$$

(d) If α is a non-generator, then the equation $\beta \equiv \alpha^x \bmod p$ may not have solutions (e.g. $\alpha = 9$, $\beta = 3$ for $p = 10$ in the table above). The exponentiation function is therefore not a bijection. However, this does not immediately prevent its use in ElGamal encryption, because we use multiplication rather than exponentiation to encrypt.

Part D: Security Situations

The following situations require information to be transfered over an insecure channel. Describe and discuss any security issues that arise (eg: how easy is Eve's job) and suggest a system to provide adequate security:

1. The canonical way to forward terminals is with `ssh`.

   You should understand roughly what is behind it. It includes public key schemes for identifying host machines and exchanging session keys (and optionally, to use for authenticating users). It has a symmetric scheme for

sending the data (more efficient). You might be interested to know that the specification calls for a block cipher (rather than a stream cipher) and they have to rekey every gigabyte (http://www.ietf.org/rfc/rfc4344.txt).

2. For online transactions, the main concern is identity. You want to be very sure you are actually sending your card details to who you intend. Hence we want to use a public key scheme where you distribute the public keys, signed by a trusted third party. Third party signatures are verified using root certificates built into browsers. We'll talk about this public key infrastructure in later lectures.

3. For James Bond to send sensitive information back to MI6: although he probably has access to some fairly hefty cryptography, the question really is about how he delivers the data. If he sends the data back over the Internet it is very easy and cheap to intercept which may leak information such as his location.

   A better choice might be for James to print out the data and post it through the public physical post system. It costs a lot more for an intercept, hence all messages might not be scanned. If the message was intercepted it might be days after it was sent allowing any information leaked from the message (e.g., his location) to no longer be relevant.

4. Luke has run into this a couple of times: he wants to tell his parents a password but doesn't want to say it out loud or send it in an E-mail. Using standard encryption tools is tricky being on holiday and using a public computer.

   In this case a simple method such as substitution could provide adequate security as it's only used for short pieces of text that ideally follow no pattern (hence frequency analysis wouldn't be effective). For example, write a little substitution scheme on a bit of paper to leave at home, and read the ciphertext over the phone.

5. Top level diplomatic secrets may be sometimes still be handled by a perfectly secure method, exploiting massive resources for exchanging limited amounts of key material. Hence a one-time pad can be used and the key distribution problem can be solved by physically delivering the keys.