12. PUBLIC KEY CRYPTOGRAPHY

§ 12.1. The RSA Code: How it Works

The need for privacy in sending messages from one person to another has greatly increased with the introduction of electronic communication. There are many systems that have been developed over the years, with varying degrees of security. **Public-key cryptography** refers to those systems where members of a large universe of users (the public) want to be able to send a message to any other user of the system. The system is operated by an operator who issues to each user certain encoding and decoding keys.

Although messages will normally be alphanumeric they can be converted to sequences of large numbers by some straight-forward conversion process. We'll assume that a 'message' is just one of these large numbers.

Clifford Cox in 2015

The method described here is called the RSA code after the three people who devised it, Ron Rivest, Adi Shanir and Leonard Adleman in 1977. Actually it was devised earlier, in 1973, by Clifford Cocks working for the British Intelligence Agency GCHQ but it remained

secret until 1997 when it was declassified. This mirrors the invention of the computer, which was assumed to be by the Americans until British classified documents were declassified many years later, showing that the British (a team led by Alan Turing) had got there first.

SETTING UP

For each user the operator chooses two large prime numbers p, q and computes $n = pq$. This is the **modulus** for that user. In practice p, q would be very large primes with something like 100 digits each so n would have a couple of hundred digits.

Then the operator computes, for each user:

$$\varphi(n) = (p - 1)(q - 1).$$

Secondly, for each user, the operator chooses an **encoding number**, e, that is coprime to $\varphi(n)$.

The numbers n and e for each user are made public, in some sort of directory, but the values of p, q and $\varphi(n)$ for each user are kept completely secret.

Traditionally, in describing public key systems, the sender of a message is known as Alice

and the recipient is called Bob. Any third person who might intercept the message is known as Eve.

In principle Eve could discover the values of p and q for any user by simply factorising their modulus. But in practice these numbers are so large that it's computationally infeasible to do this. And unless the values of p and q could be found there would be no way of computing $\varphi(n)$.

Finally, for each user, the operator calculates the inverse of their encoding number modulo their modulus. This is a number d so that $ed \equiv 1 (\text{mod } n)$.

These numbers, d, are the so-called **decoding numbers**. Each user has one and each user is informed only of their own decoding number.

SENDING A MESSAGE

Suppose Alice wants to send a message m to Bob. [Only messages that are coprime to Bob's modulus n are possible so whatever method is used to convert symbols to numbers, the multiples of Bob's primes p and q must be avoided. This is not difficult.]

Alice looks up the directory for user Bob's encoding number e and modulus n and calculates $m^e \pmod{n}$.

RECEIVING A MESSAGE

Bob takes the encoded message m^e and calculates $(m^e)^d \pmod{n}$.

Example 11: In an RSA Public Key cryptographic system a certain user is given the modulus 95 and an encoding number of 65. The operator knows that 95 is the product of the two primes 5 and 19. Find the corresponding decoding number.

Solution: $\varphi(95) = \varphi(5.19) = 4.18 = 72$. We now verify that 65 is coprime with 72.

$$\begin{array}{rrr} 1 & 9 & 3 \\ 65)\,72 & 7)\,65 & 2)\,7 \\ \underline{65} & \underline{63} & \underline{6} \\ 7 & 2 & 1 \end{array}$$

So $1 = 7 - 2.3$

$= 7 - (65 - 7.9)3 = 7.28 - 65.3$

$= (72 - 65)28 - 65.3 = 72.28 - 65.31$

Modulo 72 this becomes $1 = -65.31$ so the inverse of 65 is $-31 = 41$. This is the corresponding decoding number.

Example 12:

Alice: $p_1 = 5$, $q_1 = 19$, $n_1 = 95$, $\varphi(n_1) = 72$, $e_1 = 65$, $d_1 = 41$. N.B. $65.41 \equiv 2665 \equiv 1 \pmod{72}$

Bob: $p_2 = 7$, $q_2 = 11$, $n_2 = 77$, $\varphi(n_2) = 60$, $e_2 = 53$, $d_2 = 17$. N.B. $53.17 \equiv 901 \equiv 1 \pmod{60}$

OPERATOR KNOWS	
A	**B**
$p_1 = 5$	$p_2 = 7$
$q_1 = 15$	$q_2 = 11$
$n_1 = 95$	$n_2 = 77$
$\varphi(n_1) = 72$	$\varphi(n_2) = 60$
$e_1 = 65$	$e_2 = 53$
$d_1 = 41$	$d_2 = 17$

Eve KNOWS	
A	**B**
$n_1 = 95$	$n_2 = 77$
$e_1 = 65$	$e_2 = 53$

Alice KNOWS	
A	**B**
$n_1 = 95$	$n_2 = 77$
$e_1 = 65$	$e_2 = 53$
$d_1 = 41$	

Bob KNOWS	
A	**B**
$n_1 = 95$	$n_2 = 77$
$e_1 = 65$	$e_2 = 53$
	$d_2 = 17$

Suppose Alice wants to send the message $m = 4$ to Bob. She looks up her directory and finds $e_2 = 53$ for Bob. She therefore calculates m′ as follows:

$$m' \equiv 4^{53} \pmod{77}$$
$$4^1 \equiv 4$$
$$4^2 \equiv 16$$
$$4^4 \equiv 256 \equiv 25$$
$$4^8 \equiv 625 \equiv 9$$

$4^{16} \equiv 81 \equiv 4$

$4^{32} \equiv 16$

Now $4^{53} = 4^{32+16+4+1} = 4^{32}.4^{16}.4^4.4^1$

$\equiv 16 \,.\, 4 \,.\, 25 \,.\, 4 \equiv 6400 \equiv 9 \pmod{77}$

Hence $m' = 9$ and this is what gets transmitted to Bob.

Bob receives $m' = 9$. He uses his own decoding number, $d_2 = 17$ to calculate:

$m'' \equiv 9^{17} \pmod{77}$

$9^1 \equiv 9$

$9^2 \equiv 81 \equiv 4$

$9^4 \equiv 16$

$9^8 \equiv 256 \equiv 25$

$9^{16} \equiv 625 \equiv 9$

So $9^{17} \equiv 9.9 \equiv 81 \equiv 4 \pmod{77}$.

Thus $m'' = 4 = m$. This is the original message.

§ 12.2. The RSA Code: Why it Works

Suppose m is the original message, n is the modulus of the recipient, e is the encoding number of recipient and d is the decoding number of recipient.

Then $m' \equiv m^e \pmod{n}$

$m'' \equiv (m')^d \equiv m^{ed} \pmod{n}$.

Now d was chosen so that $ed \equiv 1 \pmod{\varphi(n)}$.

So $ed \equiv k.\varphi(n) + 1$.

Thus $m'' \equiv m^{ed} \equiv m^{k\varphi(n)+1} \pmod{n} \equiv (m^{\varphi(n)})^k.m \pmod{n} \equiv 1^k.m \pmod{n} \equiv m \pmod{n}$.
Hence the original message is recovered.

§ 12.3. The RSA Code: Is it Secure?

Once $\varphi(n)$ is known it's relatively easy for anybody to compute another user's decoding number and therefore read his electronic mail. The security of the code therefore lies in the difficulty of calculating $\varphi(n)$.

Of course the operator has no difficulty in computing $\varphi(n)$ because the operator knows the primes p, q. The users know $n = pq$ and so in principle all they have to do is factorise it. This is no real difficulty if n has only 10 digits but it is not possible, with current technology to factorise a typical 500 digit number.

It is a very elementary programming exercise to write a factorisation program along the simple-minded lines of trying every possible factor. Even though there are much more sophisticated methods available, they are all just clever variations on this simple-minded approach. Whatever method is used, the number of steps involved grows exponentially with the number of digits.

A 200 digit number (a product of two very large primes) has been factorised in recent years, but it took several months, using a large number of powerful computers working in parallel. Perhaps in another 20 years improvements in hardware and software might reduce this to 1 week of computing time but for some

time the code is safe. And of course in 20 years time when it might be possible to factorise 400 digit numbers it will be possible to generate 1000 digit primes and so use a 2000 digit value of n.

While ever it is much easier to generate a k digit prime than it is to factorise a $2k$ digit number this cryptosystem can always stay one step ahead of would-be crackers. The RSA algorithm is currently used whenever sensitive data, such as account numbers and passwords, have to be transmitted electronically.

One weakness with the above system is that the operator knows everything. This is the current situation. The message is transmitted to the operator who then encodes it and sends the encrypted message to the recipient. A new protocol is E2EE – End to End Encryption. The message is encrypted in the user's device and sent, via the operator, to the recipient where it is decrypted. Using a modified form of the above algorithm, the operator, or anyone who might intercept the transmission, is unable to decrypt the message.

Hence if some government agency asks the operator to hand over a decrypted message they are, correctly, able to affirm that this is not possible. In 2016 such E2EE systems are starting to be implemented. However some governments are expected to resist these developments and may legislate against them.

§ 12.4. Cracking the RSA Code

As described above, the RSA code seems secure. But it doesn't pay to be too complacent. For example it might have occurred to you that the system could be simplified by using the same modulus for all users. After all, if it can't be factorised, why not?

Example 13:

OPERATOR KNOWS

$p = 37$	
$q = 73$	
$n = 2701$	
$\varphi(n) = 2592$	
A	**B**
$e_1 = 125$	$e_2 = 325$
$d_1 = 1493$	$d_2 = 973$

Eve KNOWS

$n = 2701$	
A	**B**
$e_1 = 125$	$e_2 = 15$

Alice KNOWS

$n = 2701$	
A	**B**
$e_1 = 125$	$e_2 = 15$
$d_1 = 1493$	

Bob KNOWS

$n = 2701$	
A	**B**
$e_1 = 125$	$e_2 = 325$
	$d_2 = 973$

Eve doesn't know $\varphi(n)$. (We'll assume that n is so large that she can't factorise it.) But she does know that $e_1 d_1 \equiv 1 (\text{mod } \varphi(n))$ so that $\varphi(n)$ divides:
$125.1493 - 1 = 186624$.
This means that $186624 = k\varphi(n)$, for some integer k, giving $k = \dfrac{186624}{\varphi(n)}$.
Now $\varphi(n) < n$ but for large n will be close to n.
So $k > \dfrac{186624}{n} = \dfrac{186624}{2701} \approx 69.09$, but it will only be a little bigger.
Try $k = 70$. This gives $\varphi(n) = \dfrac{186624}{70} \approx 2666.057$.
This is not an integer.
Try $k = 71$. This gives $\varphi(n) = \dfrac{186624}{71} \approx 2628.507$.
This is not an integer.
Try $k = 72$. This gives $\varphi(n) = \dfrac{186624}{72} = 2592$.
This is an integer and so is probably the correct value. To be completely sure one can use $\varphi(n)$ to factorise n.

In this example, if p, q are the factors of n we know that we know that:
$pq = 2701$ and
$(p - 1)(q - 1) = \varphi(n) = 2592$
Subtracting, we get $p + q - 1 = 109$, so $p + q = 110$.

So p, q are the roots of a quadratic equation, and given the sum of the roots and the product of the roots the quadratic must be $x^2 - 110x + 2701$.

Solving, we get $x = \dfrac{110 \pm \sqrt{110^2 - 4.2701}}{2}$

$$= \frac{110 \pm \sqrt{1296}}{2}$$

$$= \frac{110 \pm 36}{2}$$

$$= 73, 37.$$

These are the factors of 2701.

Once Eve has discovered p and q for Bob she can work out his $\varphi(n)$ and she can then work out his decoding number and so, if she intercepts any message to Bob, she can decode it. But this has come about only if the same modulus is used for every user.

§ 12.5. Signature Verification

Another problem with computer security is to be able to guarantee that a particular message has come from whoever it's supposed to. If I send a message to your bank, masquerading as you, and request that your balance be transferred into a certain Swiss bank account, it would be comforting to know that your bank could tell that the request hadn't come from you. Of course I'd need to have somehow obtained your account details and

password, but that's not impossible. Signature verification is an additional security measure.

Signature verification uses the RSA system in reverse. If I want to send a message to you, in such a way that you could be sure that it has indeed come from me, I would encode it using my *decoding number* instead of your encoding number. When you receive it you decode it using my encoding number. If it comes out as a recognisable message then it must have come from me.

If the original message is m, and my modulus is n and my decoding number is d then I calculate $m' \equiv m^d$ modulo n. When you receive m' you calculate $(m')^e$.
But $(m')^e \equiv (m^d)^e \equiv m^{ed} \equiv m$.

But how would you know that m was the correct original message? With a short, cryptic message you might not. But with a much longer message, the fact that it made sense when converted to alphanumeric characters would guarantee its validity. If someone else attempted to encode the message to send money from your account to some Swiss bank account, and used the wrong decoding number (remember that only I know my own decoding number) the output after decoding would be gibberish.

Example 14:

OPERATOR KNOWS

A	B
$p_1 = 5$	$p_2 = 7$
$q_1 = 15$	$q_2 = 11$
$n_1 = 95$	$n_2 = 77$
$\varphi(n_1) = 72$	$\varphi(n_2) = 60$
$e_1 = 65$	$e_2 = 53$
$d_1 = 41$	$d_2 = 17$

Eve KNOWS

A	B
$n_1 = 95$	$n_2 = 77$
$e_1 = 65$	$e_2 = 53$

Alice KNOWS

A	B
$n_1 = 95$	$n_2 = 77$
$e_1 = 65$	$e_2 = 53$
$d_1 = 41$	

Bob KNOWS

A	B
$n_1 = 95$	$n_2 = 77$
$e_1 = 65$	$e_2 = 53$
	$d_2 = 17$

Suppose Alice wants to send the message $m = 2$ to Bob so that Bob can ensure that it came from Alice. Then Alice computes $2^{41} \pmod{95}$.

$2^2 = 4$
$2^4 = 16$
$2^8 \equiv 66$
$2^{16} \equiv 81$
$2^{32} \equiv 6$

Hence $2^{41} = 2^{32}.2^8.2 \equiv 6.66.2 \equiv 32$.

She transmits this message, 32 and when Bob receives this he calculates 32^{65}.

$32^2 \equiv 74$

$32^4 \equiv 61$

$32^8 \equiv 16$

$32^{16} \equiv 66$

$32^{32} \equiv 81$

$32^{64} \equiv 6$

Hence $32^{65} \equiv 6.32 \equiv 2$. If Eve had sent the message, using a wrong encoding number, it would have been decoded as some other number, such as 7. How does Bob know that it wasn't Eve who sent the message? After all 7 is just as plausible as 2.

But remember that, in practice, the messages will be converted as quite large numbers. So a message that had come from Alice might be decrypted by Bob as:
PLEASE SEND $1000 TO BANK ACCOUNT 51238.
If Eve had sent this, with 51238 as her own account, Bob might decrypt it as:
NGDR JU2CDF NK7RC G9KHB LQXZYQ.

EXERCISES FOR CHAPTER 12

EXERCISES 12A (Public Key Cryptography)

Ex 12A1: (a) You are a user in a Public Key Cryptographic System based on the RSA system. You wish to send the message 12 to another user whose modulus is 527 and whose encoding number is 113. What is the encoded message that you would send?
(b) Factorise 527 into primes (in practical Public Key Systems this would be the 'stumbling block'). Use this factorisation to compute the decoding number for the other user.
(c) Now use that other user's decoding number to decode the message sent in part (a).

Ex 12A2: Suppose you are setting up a Public Key Cryptographic system, using the RSA algorithm and you choose to use the same modulus 4331 = 61.71 for all users. (This is risky, as we've seen in §12.4, even for large primes.) In choosing encoding numbers for the users you'd obviously avoid $e = 1$. Suppose that encoding numbers that are prime or perfect squares are considered to be unlucky and are avoided. Choose the smallest suitable encoding number for a user and calculate the corresponding decoding number.

Ex 12A3: You are Alice Turnbull, a user of a Public Key Cryptographic system and you've been issued with this directory of moduli and encoding keys for all users.

DIRECTORY		
USER	**modulus**	**encoding number**
.....	……..	……
TURING Alan	4343	1573
TURNBULL Alice	7259	3907
TURTLE Bob	9301	129
.....	……..	……

You are also issued with a secret decoding number:

TOP SECRET
Alice Turnbull:
Your Decoding number is 4003.
DO NOT REVEAL THIS TO ANYONE

(a) You wish to send the message 2195 to Alan Turing. What should you send?

(b) You receive the message 157, apparently from Bob Turtle. Decode it.

(c) The message includes the information in 'plain text' to the effect that it has come from Bob Turtle and, when the main message is decoded it should read 8143. Was it a forgery?

Ex 12A4: You are setting up an RSA Public Key coding system and, for subscriber A, you've chosen the primes $p = 31$, $q = 83$ and the encoding number $e = 77$.
(i) Calculate the corresponding decoding number, d.
(ii) Subscriber B wishes to send the message $m = 14$ to A. What does she send to A?
(iii) B receives the message 14 from A, who claims that it is his encoding number encoded by his decoding number (using his modulus). Check whether this message did come from A.

Ex 12A5: An RSA Public Key coding system works with modulus 391 which is the product of two primes. A message, represented by the number 20, is sent to someone whose encoding key is 53.
(i) What is the corresponding encoded message?
(ii) The corresponding decoding key is 93. Use this information to find the two prime factors, p and q, of 391.
(iii) Check that $\varphi(391) = (p - 1)(q - 1)$.
(iv) Use this to find the decoding number for someone who has an encoding key of 91.

Ex 12A6: An RSA Public Key system works with the same modulus, 673627, for all users. It is the product of two primes. A message, represented by the number 3, is sent to someone whose encoding key is 13.
(i) What is the corresponding encoded message?

(ii) The corresponding decoding key is 103381. Use this information to find the two prime factors, p and q of 673627.
(iii) Check that $\varphi(673627) = (p - 1)(q - 1)$.
(iv) Use this to find the decoding number for someone else, with an encoding key of 7.

Ex 12A7: An RSA Public Key system assigns to a user the modulus 42547, the encoding key 77 and decoding key 41573. Use this information to find the two prime factors of 42547.

SOLUTIONS FOR CHAPTER 12

Ex 12A1: (a) Send 12^{113} (mod 527). Now in $\mathbb{Z}_{527}$:

$12^2 = 144$;
$12^4 = 20736 = 183$;
$12^8 = 33489 = 288$;
$12^{16} = 82944 = 205$;
$12^{32} = 42025 = 392$;
$12^{64} = 153664 = 307$;

$\therefore\ 12^{113} = 12^{64}.12^{32}.12^{16}.12$
$= 307.392.205.12$
$= 120344.2460$
$= 188.352$
$= 66176 = 301.$

So the transmitted message is 301.

(b) $527 = 17.31$ and so $\varphi(527) = 16.30 = 480$.

Now $480 = 113.4 + 28$ and

$113 = 28.4 + 1$.

Hence $1 = 113 - (480 - 113.4).4$
$= 17.113 - 480.4$

and so in $\mathbb{Z}_{480}$, $1 = 17.113$.

Thus the decoding number is 17.

CHECK: If we now decode the message 301 using this decoding number we get
301^{17}(mod 527). Using the same method as in (a) we get 12, the original message.

Ex 12A2: (a) $m = pq = 4331$ and $\varphi(m) = 60 \times 70 = 4200$. An encoding number needs to be any number that

is coprime with 4200. It therefore must have no factors of 2, 3, 5 or 7. Clearly $e = 1$ is unsuitable, and we're told to avoid primes and square values of e. The smallest suitable number is thus $e = 11.13 = 143$.
We now find the inverse of 143 modulo 4331.
$4200 = 143.29 + 53$
$143 = 53.2 + 37$
$53 = 37 + 16$
$37 = 16.2 + 5$
$16 = 5.3 + 1$ so
$1 = 16 - 5.3$
$= 16 - (37 - 16.2).3$
$= 16.7 - 37.3$
$= (53 - 37).7 - 37.3$
$= 53.7 - 37.10$
$= 53.7 - (143 - 53.2).10$
$= 53.27 - 143.10$
$= (4200 - 143.29).27 - 143.10$
$= 4200.27 - 143.793.$
So $d = -793 = 3407$.
Thus $e = 143$ and $d = 3407$.

Ex 12A3: (a) The encoded message is 2195^{1573} (mod 4343).
Now $1573 = 1024 + 512 + 32 + 4 + 1$
We prepare a table of values of 21952^{2^m} modulo 4343.

m	1	2	4	8	16	32
2195^m	2195	1638	3413	643	864	3843

m	64	128	256	512	1024
2195^m	2449	4261	2381	1546	1466

So 2195^{1573} = 2195.3413.3843.1546.1466
= 4203.3843.1546.1466
= 512.1546.1466
= 1126.1466
= 376.

(b) The decoded message is 157^{4003} (mod 7259).
Now 4003 = 2048 + 1024 + 512 + 256 + 128 + 32 + 2 + 1.

m	1	2	4	8	16	32
157^m	157	2872	2160	5322	6325	1276

m	64	128	256	512	1024	2048
157^m	2160	5322	6325	1276	2160	5322

So $157^{4003} \equiv$
5322.2160.1276.6325.5322.1276.2872.4003
$\equiv$ 4523.5951.3707.5619
$\equiv$ 1.3562
$\equiv$ 3562

(c) We must calculate 3562^{129} (mod 9301).

m	1	2	4	8	16	32	64	128
3562^m	3562	1280	1424	158	6362	6393	1855	8956

$3562^{129} \equiv 8956.3562 \equiv 8143.$
Hence we assume that it must have genuinely come from Andreas Turtle.

Ex 12A4:
(i) $n = 2573$. $\varphi(n) = 30.82 = 2460$.
So we must solve $77d \equiv 1 (\text{mod } 2460)$.

$2460 = 77.31 + 73$
$77 = 73.1 + 4$
$73 = 4.18 + 1$
$\therefore 1 = 73 - 4.18$
$= 73 - (77 - 73).18$
$= 73.19 - 77.18$
$= (2460 - 77.31).19 - 77.18$
$= 2460.19 - 77.607$
$\equiv -77.607 (\text{mod } 2460)$
$\therefore d \equiv -607 \equiv 1853 (\text{mod } 2460)$.
So the decoding key is 1853.

(ii) B sends $14^e (\text{mod } pq)$, that is, $14^{77} (\text{mod } 2573)$.
$14^2 \equiv 196$
$14^4 \equiv 2394$
$14^8 \equiv 1165$
$14^{16} \equiv 1254$
$14^{32} \equiv 413$
$14^{64} \equiv 751$
$77 = 64 + 8 + 4 + 1$

$\therefore\ 14^{77} \equiv 751.1165.2394.14$

$\equiv 95.67 \equiv 1219.$

So B sends A the coded message 1219.

(iii) If A had sent his encoding number, coded by his decoding number, then it should have been $77^d \pmod{2573}$.

That would mean that $77^d \equiv 14 \pmod{2573}$.

Then $77^{77d} \equiv 14^{77} \pmod{2573}$, that is $14^{77} \equiv 77 \pmod{2573}$.

But from (ii) we found that

$14^{77} \equiv 1219$, not 77.

Therefore either A got it wrong, or more likely, the message came from someone else pretending to be A.

Ex 12A5:

(i) The encoded message is $20^{53} \pmod{391}$.

$20^2 \equiv 400 \equiv 9$

$20^4 \equiv 81$

$20^8 \equiv 305$

$20^{16} \equiv 358$

$20^{32} \equiv 307$

$53 = 32 + 16 + 4 + 1$ so

$20^{53} = 307.358.81.20$

$= 35.56 = 5.$

So the encoded message is 5.

(ii) $53.93 = 4929 \equiv 1 \pmod{\varphi(391)}$.

Hence $4928 = \varphi(391)k$ for some k.

Now $\varphi(391) < 391$ so $k > 4928/391 \approx 12.60$.

Try $k = 13$. Then $\varphi(391) = 4928/13 \approx 379.077$. This is impossible.
Try $k = 14$. Then $\varphi(391) = 4928/14 = 352$. This is probably correct.
If p, q are the prime factors of 391 we therefore have $pq = 391$.
Because the numbers are small in this exercise it would be very easy to find p and q but let's pretend that this is not feasible, as would be the case with 200 digit primes.

So $pq = 391$ and
$\varphi(391) = (p - 1)(q - 1) = 352$.
Subtracting, we get $p + q - 1 = 39$ so $p + q = 40$.

So p, q are the roots o a quadratic where the sum of the roots is 40 and the product is 391.
This quadratic equation is $x^2 - 40x + 391 = 0$.
Solving, we get $x = \dfrac{40 \pm \sqrt{36}}{2} = 17$ and 23.

(iii) $(p - 1)(q - 1) = 16.22 = 352 = \varphi(391)$.

(iv) The decoding number for $e = 91$ would satisfy $91d \equiv 1 \pmod{352}$.
Now $352 = 91.3 + 79$
$91 = 79.1 + 12$
$79 = 12.6 + 7$
$12 = 7.1 + 5$
$7 = 5.1 + 2$

$5 = 2.2 + 1$

$\therefore\ 1 = 5 - 2.2$

$= 5 - (7 - 5).2$

$= 5.3 - 7.2$

$= (12 - 7).3 - 7.2$

$= 12.3 - 7.5$

$= 12.3 - (79 - 12.6).5$

$= 12.33 - 79.5$

$= (91 - 79).33 - 79.5$

$= 91.33 - 79.38$

$= 91.33 - (352 - 91.3).38$

$= 91.147 - 352.38 \equiv 91.147 \pmod{352}$.

Hence the decoding number is 147.

Ex 12A6:

(i) $3^{13} = 1594323 \equiv 247069 \pmod{673627}$.

So the encoded message is 247069.

(ii) Let $e = 13$, $d = 103381$.

Then $ed = 1343953 \equiv 1 \pmod{(p-1)(q-1)}$.

Hence $(p-1)(q-1)$ divides 134952.

We know also that $pq = 673627$.

Since $(p-1)(q-1)$ is less than pq but most likely has the same order of magnitude as pq it is likely that

$134952 = 2(p-1)(q-1)$, which gives

$(p-1)(q-1) = 671976$.

Hence $pq - p - q + 1 = 671976$ which gives

$p + q = pq + 1 - 671976$

$= 673627 + 1 - 671976 = 1652$.

We therefore have $p + q = 1652$ and $pq = 673627$. It follows that p, q are solutions to the quadratic equation $x^2 - 1652x + 673627 = 0$. The solutions of this quadratic are the integers 733 and 919. We can check that they are indeed the two factors of 673627.

(iii) $(p - 1)(q - 1) = 732.918 = 671976$.

(iv) If now $e = 7$ then the corresponding decoding key satisfies $7d \equiv 1 \pmod{671976}$.

The solution can be easily found to be 479983.

Ex 12A7: Let $e = 77$, $d = 41573$, $n = pq = 42547$.

Then $ed - 1 = 3201120 = k\varphi(n)$ for some integer k.

$$k = \frac{3201120}{\varphi(n)} > \frac{3201120}{42547} \approx 75.23.$$

Try $k = 76$: $\varphi(n) = 42120$. Since this is an integer it is probably the correct value.

$$pq = 42547$$

$$(p - 1)(q - 1) = 42120$$

$\therefore p + q - 1 = 427$ so $p + q = 428$.

Hence p, q are the roots of the quadratic $x^2 - 428x + 42547$.

So the prime factors of 42547 are 271, 157.