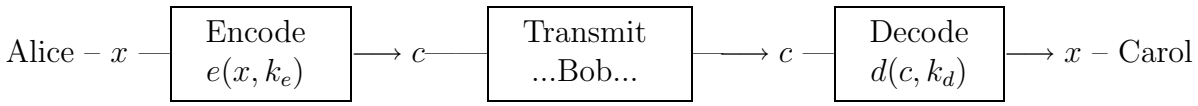
10.4

Cryptography

P. Danziger

1 Cipher Schemes

A cryptographic scheme is an example of a code. The special requirement is that the encoded message be difficult to retrieve without some special piece of information, usually referred to as a *key*. The key used for encoding may or may not be the same as the key used for decoding.
We presume that we are sending a secret message from Alice to Carol. An adversary, Bob, has access to the message stream, but not to the original messages. A fundamental tenet of cryptography is that Bob knows the algorithm that is being used for encryption/decryption, but not the keys. Thus the security of a cryptographic scheme relies on the difficulty of determining a decryption key given an encrypted message. Not on a lack of knowledge about the method of encryption.
Alice starts with a *plaintext* message x. This is then encoded by some *encoding function*, using an encoding key k_e, to an encrypted form c, called a *ciphertext*. Alice then sends the ciphertext over the channel, where it can be seen by Bob and received by Carol. Carol then uses the decoding key k_d and a *decoding function* to obtain the plaintext sent by Alice.
The original plaintext message x is over some alphabet Σ_P and the ciphertext message c is over another alphabet Σ_C. In practice we often use the same alphabet for both, so $\Sigma_P = \Sigma_C$.
In addition we have the key alphabet Σ_K, from which the keys may be chosen. This is also often the same as the message alphabet, but is often an integer.
The encoding function is a function which takes a key and a message to produce a ciphertext version of the message. $e : \Sigma_P \times \Sigma_K \longrightarrow \Sigma_C$, $e(x, k_e) = c$.
Similarly, the decoding function, d, takes a ciphertext message, c, and produces the original plaintext message x, using the decoding key k_d. $d : \Sigma_C \times \Sigma_K \longrightarrow \Sigma_P$, $d(c, k_d) = x$.



Example 1 Rot13

This is not really a cipher, as there is no key, but it serves well as a first example.
The plaintext and ciphertext alphabets are the same, the letters from a to z. So $\Sigma_P = \Sigma_C =$[a-z].
Each letter is assigned a number in sequence, starting from 0. So

$$\begin{aligned} a &\longrightarrow 0 \\ b &\longrightarrow 1 \\ c &\longrightarrow 2 \\ &\text{etc...} \end{aligned}$$

The encoding is done one character at a time, the encoding function is

$$e(x) = x + 13 \bmod 26$$

So for example "hello world" is translated to "uryyb jbeyq".
The decoding function is then $d(x) = x - 13 \bmod 26$, but since $-13 \equiv 13 \pmod{26}$, the encoding and decoding functions are the same, $d(x) = e(x)$.
Rot13 provides no security, it is trivial to decrypt, but it was widely used to obfuscate potentially offensive usenet postings.
Rot13 is an example of a wider class of ciphers known as Caesar Ciphers since these were used by the Romans.

Example 2 Caesar Ciphers

As above the plaintext and ciphertext alphabets are the same, the letters from a to z. So $\Sigma_P = \Sigma_C =$[a-z], and each letter is assigned a number in sequence, starting from 0. Again the encoding is done one character at a time.
The key is an integer, k, from 0 to 25 (Julius Caeser used $k = 3$). The encoding function is

$$e(x, k) = x + k \bmod 26$$

The decoding function is

$$d(x, k) = x - k \bmod 26 = x + (26 - k) \bmod 26$$

Rot13 is therefore a Caesar cipher with $k = 13$.
The Caesar ciphers main weakness is that it is a substitution cipher, and so is susceptible to frequency analysis. Specifically, since the same letter is always translated to the same symbol, we can analyze the frequency with which letters appear. Given a sufficiently long text it is possible to reconstruct the message by assuming that the most frequently occuring symbol is the most frequently occuring letter in the language of the plaintext ('e' for English) and so on.
Frequency analysis becomes even more powerful if we know some phrase or sequence of letters in the plaintext. Allied codebreakers in the second world war were initially able to break the German codes because, though the Germans had a new key for each day, they always started the day with the weather report. The codebreakers were able to use the limited vocabulary of this report, together with knowledge of the prevailing weather that was described to find the daily key.
A variation of the Caesar cipher, called the book cipher avoids this problem.

Example 3 Book Cipher

Before communicating Alice and Carol choose a book, or specifically a piece of text of which they each have identical copies, this is the key. Suppose that the text has characters (ignoring spaces and punctuation) $a_1a_2a_3a_4\ldots$, and the plaintext message that we wish to send is $b_1b_2b_3b_4$. We form the ciphertext by adding mod 26.

$$c_i = a_i + b_i \bmod 26, \quad 1 \leq i \leq 4.$$

The use of a cipher "block" scrambles the letters and so stops frequency analysis.

2 Public and Private keys

Caesar and Book ciphers are an example of a private key cryptosystem. Alice and Carol must meet secretly, somewhere they believe Bob cannot hear and decide on a key pair for their messages. The most common private key systems in use are DES (Data Encryption Standard) now outdated, and the more modern AES (Advanced Encryption Standard aka Rijndael).
Clearly this is a problem, particularly in net based applications where all communication happens over an inherently insecure channel. The answer is public key encryption. In a public key system, the encryption key is public, but the decryption key is kept private.
Thus for Alice to send a message to Carol, she first encrypts it with Carol's (public) encryption key. She then sends the ciphertext to Carol who decrypts it with her secret decryption key. Bob in the middle is none the wiser, since he still lacks the decryption key.
In a public key encryption scheme it is vital that knowledge of the encryption key gives no information about the decryption key. This is clearly not the case for the Caesar ciphers above. Without any extra knowledge there are 26 possibilities for keys. However if we know the encryption key, it is easy to find the decryption key.
The most common public key encryption schemes are RSA (named after its inventors Ron Rivest, Adi Shamir and Leonard Adleman) and ECC (Elliptic Curve Cryptography). We will be discussing RSA.
In practice public key systems are more complex, and hence much slower than private key ones. Also RSA is a substitution cipher and so susceptible to frequency analysis attacks if the message is long enough. Thus the usual protocol is to use public key encryption to negotiate a private key which is then used for the rest of the transmission.

2.1 RSA

The security of RSA relies on the difficulty of solving the discrete logarithm problem. Specifically, given integers x, y and n, find d such that $y = x^d \pmod{n}$.
In addition, RSA relies on the difficulty of factoring large numbers. In particular it is possible to verify pimality of numbers which are several hundred digits long, but it is infeasible to factor their product.
RSA starts by choosing two (large) primes, p and q, we then let $N = pq$ and let $M = (p-1)(q-1)$. The number N is made public, but the primes p and q and the number M are kept secret. In practice p and q must be large enough that it is non trivial to factor their product.
The (public) encryption key, e, is a number between 1 and $(p-1)(q-1)$, which is relatively prime to $(p-1)(q-1)$, i.e. $\gcd(e,(p-1)(q-1)) = 1$. This ensures that e has a multiplicative inverse $\bmod (p-1)(q-1)$.
Each block of plaintext is a number from 1 to pq. The ciphertext C for the plaintext x is

$$C = x^e \pmod{N}$$

The decryption key, d, is the multiplicative inverse of e modulo $(p-1)(q-1)$. To decrypt we do the following on the ciphertext C

$$x = C^d \pmod{N}$$

Example 4

In this example we will take $p = 7$ and $q = 11$ in order to keep the calculations manageable. (Note that in practice we must choose p and q large enough that it is non trivial to factor their product.) $N = pq = 77$, $M = (p-1)(q-1) = 60$. So we choose e between 1 and 59.

1. Determine whether $e = 21$ is a valid encryption key.

   For e to be a valid key we must have $\gcd(M, e) \neq 1$. We use the Euclidean algorithm to find $\gcd(60, 21)$.

   $$\begin{array}{rcll} \gcd(60,21) & & & 60 = 21 \times 2 + 18 \\ & = & \gcd(21,18) & 21 = 18 + 3 \\ & = & \gcd(18,3) & 18 = 3 \times 6 + 0 \\ & = & \gcd(3,0) \qquad = 3 & \end{array}$$

   So $e = 21$ is a bad key as $\gcd(M, e) \neq 1$.

2. Determine whether $e = 23$ is a valid encryption key.

   $$\begin{array}{rcll} \gcd(60,23) & & & (60 = 23 \times 2 + 14) \\ & = & \gcd(23,14) & (23 = 14 + 9) \\ & = & \gcd(14,9) & (14 = 9 + 5) \\ & = & \gcd(9,5) & (9 = 5 + 4) \\ & = & \gcd(5,4) & (5 = 4 + 1) \\ & = & \gcd(4,1) & \\ & = & 1 & \end{array}$$

   $\gcd(60, 23) = 1$, so is a valid key.

3. Given the encryption key $e = 23$ find the corresponding decryption key d.

   We calculate $d = e^{-1} \pmod{M}$ using the information above.

   $$\begin{array}{rcll} 1 & = & 5 - \underline{4} & (5 = 4 + 1) \\ & = & 5 - (9 - 5) & (4 = 9 - 5) \\ & = & 2 \times (14 - 9) - 9 & (5 = 14 - 9) \\ & = & 2 \times 14 - 3 \times (23 - 14) & (9 = 23 - 14) \\ & = & 5 \times (60 - 2 \times 23) - 3 \times 23 & (14 = 60 - 2 \times 23) \end{array}$$

   $$\begin{array}{rcll} & & & \\ & = & 2 \times \underline{5} - 9 & \\ & = & 2 \times 14 - 3 \times \underline{9} & \\ & = & 5 \times \underline{14} - 3 \times 23 & \\ & = & 5 \times 60 - 13 \times 23 & \end{array}$$

   So $1 = 5 \times 60 - 13 \times 23$ and $d = 23^{-1} \equiv -13 \equiv 47 \pmod{60}$.

   So $d = 47$ is the decryption key.

$e = 23$ and $N = 66$ are made public, but d and M are kept secret.

4. Suppose that Alice is using the encryption key $e = 23$ above and she wishes to encrypt the plaintext message $x = 5$, find the corresponding ciphertext C.

$C = 5^{23} \pmod{77}$. We use the fastpower algorithm to compute C, note that $23 = 10111$ binary.

level	power	b	action	return
1	23	10111	$\times a$	59
2	22	10110	square	58
3	11	1011	$\times a$	38
4	10	1010	square	23
5	5	101	$\times a$	45
6	4	100	square	9
7	2	10	square	25
8	1	1	$\times a$	5
9	0	0	-	1

All calculations are (mod 77).
Specifically, $25^2 = 625 \equiv 9$, $5 \times 9 \equiv 45$, $45^2 = 2025 \equiv 23$, $5 \times 23 = 115 \equiv 38$, $38^2 = 1444 \equiv 58$, $5 \times 58 = 290 \equiv 59 \pmod{77}$.

So $C = 59$ is the value sent over the channel.

5. Now Carol decrypts the received message $C = 59$ using her secret decryption key $d = 47$. Find the value of the original plaintext x.

   $x = 59^{47} \pmod{77}$. Note that $47 = 101111$ binary.

level	power	b	action	return
1	47	101111	$\times a$	5
2	46	101110	square	4
3	23	10111	$\times a$	75 ($\equiv -2$)
4	22	10110	square	60
5	11	1011	$\times a$	26
6	10	1010	square	67
7	5	101	$\times a$	12
8	4	100	square	25
9	2	10	square	16
10	1	1	$\times a$	59
11	0	0	-	1

   All calculations are (mod 77). Specifically,
   $59^2 = 3481 \equiv 16$, $16^2 = 256 \equiv 25$, $59 \times 25 = 1475 \equiv 12$, $12^2 = 144 \equiv 67$ $59 \times 67 = 3953 \equiv 26$, $26^2 = 676 \equiv 60$ $59 \times 60 = 3540 \equiv 75$, $75^2 = 5625 \equiv 4$ $59 \times 4 = 236 \equiv 5 \pmod{77}$.

   So $x = 5$.

In the absence of knowledge about p, q or M, Bob must try all 77 possible decryption keys.
Note how the Euclidean gcd algorithm, the algorithm for finding inverses mod n and the fastpower algorithm are all essential to being able to work in these environments.

3 Why does RSA work?

We find the dencryption key by finding the inverse of the encryption key mod M, but all arithmetic is done mod N, why does this work? Before answering this question we will need the following result.

3.1 The Chinese Remainder Theorem

A common situation is that we know some information about an integer modulo several different integers. The Chinese Remainder Theorem lets us put this information together in a useful form. For example, suppose that we know that an integer x is odd (so $x \equiv 1 \bmod 2$) and that it is 2 mod 3. These two pieces of information can be put together into on statement about the modularity of x mod 6, namely $x \equiv 5 \bmod 6$. Note that $6 = 2 \times 3$, also 2 and 3 are relatively prime.

Example 5

Consider that if $x \equiv 2 \pmod 3$ then $x = 2$ or $5 \pmod 6$.
Also if $x \equiv 1 \pmod 2$ then $x = 1, 3$ or $5 \pmod 6$. (Exercise prove these two statements.)
The only way both can be true is if $x \equiv 5 \pmod 6$.
The Chinese Remainder Theorem give us a general way of dealing with statements of this kind.

Theorem 6 (The Chinese Remainder Theorem) *If $n_1, n_2, \ldots, n_k$ are k pairwise relatively prime integers (so $\gcd(n_i, n_j) = 1$, when $i \neq j$), then the system of congruences*

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ &\vdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

has a unique solution modulo $n = n_1 n_2 \ldots n_k$.
Furthermore, if $N_j = \frac{\Pi_{i=1}^{k} n_i}{n_j}$ (the products of all the n_i's except n_j) and for each j, z_j is a solution of $N_j z_j \equiv a_j \pmod{n_j}$, then the solution is given by $x \equiv \sum_{j=1}^{k} N_j z_j \pmod n$.

SWP

Example 7

1. Consider the example given above. We have that $x \equiv 1 \bmod 2$ and $x \equiv 2 \bmod 3$.
   So $n_1 = 2,\ a_1 = 1$ and $n_2 = 3,\ a_2 = 2$.
   Now $N_1 = n_2 = 3$ and $N_2 = n_1 = 2$.
   z_1 is a solution to $3z_1 \equiv 1 \bmod 2$, so $z_1 = 1$ is a possibility.
   z_2 is a solution to $2z_2 \equiv 2 \bmod 3$, so $z_2 = 1$ is a possibility.
   So $x \equiv N_1 z_1 + N_2 z_2 = 3 \times 1 + 2 \times 1 = 5$.

2. Suppose that we are given that for some number x:

$$\begin{aligned} x &\equiv 1 \pmod 3 \\ x &\equiv 3 \pmod 4 \\ x &\equiv 1 \pmod 5 \end{aligned}$$

Find x mod 60. Your answer should be an integer between 0 and 59.

For this problem we take $n_1 = 3$, $n_2 = 4$, $n_3 = 5$, note that these are pairwise relatively prime. Thus,

$$\begin{aligned} N_1 &= 4 \times 5 = 20, \quad a_1 = 1, \\ N_2 &= 3 \times 5 = 15, \quad a_2 = 3, \\ N_2 &= 4 \times 5 = 12, \quad a_3 = 1, \end{aligned}$$

$$\begin{aligned} 20z_1 &\equiv 1 \pmod 3 \Rightarrow z_1 = 2 \text{ is a solution.} \\ 15z_2 &\equiv 3 \pmod 4 \Rightarrow z_2 = 1 \text{ is a solution.} \\ 12z_3 &\equiv 1 \pmod 5 \Rightarrow z_3 = 3 \text{ is a solution.} \end{aligned}$$

Now

$$x = \sum_{i=1}^{3} N_i z_1 = 40 + 15 + 36 \equiv 31 \pmod{60}$$

3. Suppose that we have some number, x of balls. When we place them in cartons which take 3 balls each, there is 1 ball left over. When we place them in cartons which take 4 balls each, there are 3 balls left over. When we place them in cartons which take 5 balls each, there is 1 ball left over. How many balls will be left over when we place them in cartons which can hold 60 balls each.

This is actually the same as the last question. When we place the balls in the size 3 cartons we get that $x = 3k + 1$, the size four that $x = 4k' + 3$ and the size 5 cartons that $x = 5k'' + 1$. Thus

$$\begin{aligned} x &\equiv 1 \pmod 3 \\ x &\equiv 3 \pmod 4 \\ x &\equiv 1 \pmod 5 \end{aligned}$$

and the problem has the same solution as above, namely $x \equiv 31 \pmod{60}$.

We note a special case of the Chinese Remainder Theorem which we will find useful.

Corollary 8 *Given relatively prime integers, p and q, and $a \in \mathbb{Z}$, if $x \equiv a \bmod p$ and $x \equiv a \bmod q$ then $x \equiv a \bmod pq$.*

3.2 RSA Investigated

We encrypt a message x to $c \equiv x^e \pmod N$. Now

$$x \equiv c^d \equiv (x^e \bmod N)^d \equiv x^{ed} \pmod N$$

So we must have that $x \equiv x^{ed} \pmod{N}$, but d was chosen to be the inverse of $e \bmod M$, where $M = (p-1)(q-1)$. Thus

$$ed \equiv 1 \pmod{(p-1)(q-1)},$$

which is the same as saying

$$ed = 1 + (p-1)(q-1)k \text{ for some } k \in \mathbb{Z}.$$

Thus

$$x^{ed} = x^{1+(p-1)(q-1)k} = x(x^{p-1})^{(q-1)k} = x(x^{q-1})^{(p-1)k}$$

Now, provided that $p \nmid x$, Fermat's Little Theorem tells use that $x^{p-1} \equiv 1 \pmod{p}$. Similarly, provided that $q \nmid x$, we have $x^{q-1} \equiv 1 \pmod{q}$. Thus,

$$x^{ed} = x(x^{q-1})^{(p-1)k} \equiv x(1)^{(p-1)k} = x \pmod{q}.$$

Similarly,

$$x^{ed} = x(x^{p-1})^{(q-1)k} \equiv x(1)^{(q-1)k} = x \pmod{p}.$$

So $x^{ed} \equiv x \bmod p$ and $x^{ed} \equiv x \bmod q$. So by the Chinese Remainder Theorem

$$x^{ed} \equiv x \pmod{pq}$$

as required.
Note that we require a preprocessing stage so that the unencrypted x is relatively prime to $N = pq$.