Chapter 13
Public Key Cryptography

In this short chapter, we talk a bit about cryptography. First, we discuss some classical sorts of private key methods, and their limitations in the modern world. We then look at the first public key cryptographic method.

13.1 Private Key Cryptography

Countless methods of encrypting messages have been invented over the centuries, and we will not attempt to give an exhaustive list here. Let us discuss a few well-known codes.

One ancient method is known as the **Caesar cipher**. It could not be much simpler! Each letter in the alphabet is shifted forward three letters. Thus, A becomes D, B becomes E, and Z becomes C. If we wish to send the message HOWDY[1] then our encrypted message is KRZGB. Decrypting the message is equally simple; the recipient shifts each letter back three positions.

This is not a particularly good code. An opponent who knew that we were using a Caesar cipher could read any intercepted message instantly. We could complicate things a bit by selecting a positive integer k as a **key**. Instead of shifting letters 3 positions ahead, we would shift them k positions ahead. Decryption would then be a matter of shifting back k positions. We call this an **additive cipher**. This is better, but not much. There are only 26 possible keys (really 25, as one of them will just leave the message unencrypted). It would not take an opponent long to try all of the possible keys on an intercepted encrypted message, and see which one gives sensible text.

But we can be more sophisticated than that. Let ρ be any permutation of the set of letters of the alphabet. We can then encrypt text by replacing each letter with its

[1]The author acknowledges that the circumstances under which this message would need to be sent secretly are few and far between.

image under ρ. This is called a **simple substitution cipher**. For instance, suppose we use Table 13.1.

Table 13.1 Encryption table for a simple substitution cipher

original text	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
encrypted text	R	V	C	X	N	O	A	Y	W	B	K	U	J	E	T	D	I	L	Q	M	Z	F	S	P	G	H

Encrypting HOWDY, we obtain YTSXG. To decrypt, we apply the inverse of ρ. To put this another way, we flip the rows of the table and, for the sake of convenience, sort by the encrypted letter rather than the original letter, as in Table 13.2.

Table 13.2 Decryption table for the simple substitution cipher from Table 13.1

encrypted text	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
original text	G	J	C	P	N	V	Y	Z	Q	M	K	R	T	E	F	X	S	A	W	O	L	B	I	D	H	U

Thus, YTSXG decrypts to HOWDY. This is a vast improvement over the additive cipher in terms of security, because the number of possible keys is 26!. Even for a computer, that is a huge number of permutations to consider. It does come at the cost of having a larger key to exchange. Also, if a substantial amount of text is intercepted, this cipher is vulnerable to frequency analysis. That is, in English text, some letters occur much more frequently than others. For instance, E is by far the most common letter, T is the second most common, and so forth. An opponent could look for the most common letters in our text and make an educated guess that those represent T and E. If another moderately common letter occurs between them frequently, it might just be THE. Proceeding in this way, the code could be cracked.

Can anything be done about this? There is always the **one time pad**. This is, in fact, an unbreakable cipher. It is also quite simple. The key is a string of random letters, at least as long as our message to be encrypted. We then assign a numerical value to each letter. We let A be 0, B be 1, and so on, letting Z be 25. To encrypt, we add the value of the first letter of our message to the value of the first letter of our key in $\mathbb{Z}_{26}$. We then do the same with the second letter of our message and the second letter of our key, until we reach the end of our message. Each of our sums is then converted back to a letter.

For instance, say we wanted to encrypt HOWDY, and our randomly selected key was NCVBT. Now, H is 7 and N is 13, so the sum is 20, which is U. Similarly, O and C are 14 and 2 respectively, giving a sum of 16, which is Q. Next, W is 22 and V is 21, giving a sum of 17, which is R. Now, D and B give $3 + 1 = 4$, and hence E, and Y and T give $24 + 19 = 17$, which is R. Thus, our encrypted text is UQRER. To decrypt, we subtract the value of the key letter from the corresponding encrypted letter value. For this message, $20 - 13 = 7$, $16 - 2 = 14$, $17 - 21 = 22$, $4 - 1 = 3$ and $17 - 19 = 24$, and we obtain HOWDY.

Assuming that the key is truly random, is only used once and is kept secret, an opponent who intercepts an encrypted message will be unable to determine anything more than the length of the message. The difficulty with this cipher is that the participants must have the ability to exchange a very large key secretly. In general, anyone who can do that may not need a code! For certain purposes, though, it is ideal. For example, if two people are able to meet once, exchange a briefcase full of random letters, and then leave for distant cities, they will be able to exchange messages while they are apart.

In the internet age, the problem is that most encrypted messages are sent between two distant computers, and the computers can never meet secretly to exchange information. None of the schemes we have discussed are suitable. These are all **private key** methods. That is, the key used must be kept secret. Any opponent who discovered it could easily decrypt an intercepted message, since the ability to encrypt implies the ability to decrypt. Modern codes use **public key** schemes. The key can be released to an opponent without fear, because in these methods, it is quite possible to be able to encrypt and not be able to decrypt.

The next section is devoted to a discussion of the first such scheme.

Exercises

Spaces and punctuation have been deleted from all messages to be encrypted or decrypted.

13.1. Encrypt the following message using a Caesar cipher:
THETREASUREISBURIEDTWENTYPACESNORTHOFTHEPALMTREE

13.2. A message in English has been encrypted using an additive cipher. Decrypt it.
BPMBQUMPIAKWUMBPMEITZCAAIQLBWBITSWNUIVGBPQVOA

13.3. Let us define a multiplicative cipher as follows. Assign the letters of the alphabet numerical values as usual (A is 0, B is 1, Z is 25), and choose a positive integer k as a key. Then if a letter with value v appears in the text, encrypt it as kv, with the multiplication taking place in $\mathbb{Z}_{26}$. Which values of k will produce a valid cipher?

13.4. A message was encrypted using a multiplicative cipher, as in the preceding problem, with $k = 7$. Decrypt it.
WUGCDEGCWERCHCZECRCVAWGANMAWWE
FEGBUWWEHZCDXENQWHCJUPCHPCASJAWD

13.5. We establish a simple substitution cipher using the following table.

original text	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
encrypted text	V	Y	Z	X	E	N	A	W	R	I	O	P	C	S	B	D	F	G	H	J	K	L	M	Q	T	U

Encrypt the following message:
TRANSFERTENMILLIONDOLLARSONTUESDAY

13.6. Using the same simple substitution cipher as in the preceding problem, a message is encrypted. Decrypt it.

YEJJEGJWGEEWBKGHJBBHBBSJWVSVCRSKJEJBBPVJE

13.7. Making use of the key

RQPKDFOCMWODKFJDKSKDFVKQUYCHTISOXETX,

encrypt the following message with a one time pad.

THEDOCUMENTSAREHIDDENBEHINDTHERENOIR

13.8. Making use of the key

ICOWLDIFNSXZIEEOWPAMWRUSDMFJEJFJBAWUQH,

the following message was encrypted with a one time pad. Decrypt it.

EJSJLQOWLULTVXXCBDUDSYYFYQWHEWLAZSSYQY

13.2 The RSA Scheme

The **RSA Scheme** is a public key cipher first described by Ronald L. Rivest, Adi Shamir and Leonard N. Adleman in 1977. In fact, an equivalent system was created by Clifford C. Cocks in 1973, but his work was classified and not made public for more than two decades.

For convenience, let us say that Bob will be sending messages to June. In this case, it is June who creates the cipher. She selects two large distinct primes p and q, and lets $n = pq$. By Theorem 3.19, $\varphi(n) = (p-1)(q-1)$. June chooses a number e, with $1 < e < \varphi(n)$, such that $(e, \varphi(n)) = 1$. The public key consists of the numbers e and n. She sends these to Bob, without worrying about whether they are intercepted. She does not, however, tell anyone what p and q are!

Bob prepares to send a message m which must be an integer with $0 \leq m < n$. (We will discuss how to convert text to this format shortly.) Bob calculates

$$m^e \equiv a \pmod{n},$$

where $0 \leq a < n$. He then sends the encrypted message a to June.

How does June decrypt? The number e was chosen so that $e \in U(\varphi(n))$. Let d be the inverse of e in this group. To put that another way,

$$de \equiv 1 \pmod{\varphi(n)}.$$

But now we have the following theorem.

Theorem 13.1. *Let p and q be distinct primes and $n = pq$. If $k \equiv 1 \pmod{\varphi(n)}$, then for any integer b, we have*

$$b^k \equiv b \pmod{n}.$$

Proof. First suppose that $(b, n) = 1$. Then $b \in U(n)$. But by Theorem 3.17, $|U(n)| = \varphi(n)$. Thus, by Corollary 3.5, $b^{\varphi(n)} \equiv 1 \pmod{n}$. Now, $\varphi(n)|(k-1)$, and hence $b^{k-1} \equiv 1 \pmod{n}$. Thus, $b^k \equiv b \pmod{n}$.

Now, suppose that exactly one of $\{p, q\}$ divides b. Without loss of generality, say $p|b$ but $q \nmid b$. Then $b \in U(q)$. As $|U(q)| = \varphi(q) = q - 1$, we see that $b^{q-1} \equiv 1 \pmod{q}$. Thus, $b^{(p-1)(q-1)} \equiv 1 \pmod{q}$ and hence, as above, $b^k \equiv b \pmod{q}$. Also, $b \equiv b^k \equiv 0 \pmod{p}$. Thus, p and q both divide $b^k - b$. Since p and q are relatively prime, Corollary 2.3 tells us that $n = pq|(b^k - b)$, as required.

Finally, if both p and q divide b, then $b \equiv b^k \equiv 0 \pmod{n}$. □

Therefore, to decrypt, June calculates

$$a^d \equiv (m^e)^d \equiv m \pmod{n}.$$

The scheme works because June is the only one who knows d. In order to calculate d, an opponent would need to find $\varphi(n)$. But knowing that means being able to calculate p and q (see Exercise 13.10). And it is precisely upon the difficulty of this problem that the security of the system rests. To be sure, if Bob and June were foolish enough to use $n = 143$, an opponent would be able to find p and q instantly. But what if n had 300 digits? Factoring that into two primes of roughly 150 digits each is certainly beyond human abilities, and even for a computer, it is going to take a very long time. In theory, the cipher would be breakable. But any system that will take a fast computer a trillion years to crack is good enough for most purposes.

How do we create our messages? Suppose that n has d digits. Then we will create a message m that is at most $d - 1$ digits long, so that $m < n$. If $d - 1$ is even, we will do this by grouping our message into blocks of $\frac{d-1}{2}$ letters; if it is odd, then the blocks will have size $\frac{d-2}{2}$. Use the same values for letters introduced in the previous section, but make each letter have two digits. Thus, A is 00, B is 01, and Z is 25. Put the numbers from one block together to form a message m. (We have to use the two-digit method. If we dropped the leading zeroes, we would not know if 123 meant 1, 2 and 3 (BCD), 1 and 23 (BX) or 12 and 3 (MD).) If the length of our text message does not split evenly into blocks of the appropriate length, then pad out the last block with random letters.

Example 13.1. June decides to create an RSA scheme using the primes $p = 113$ and $q = 137$. (Yes, these are much too small to produce a secure system. However, the author is far too lazy to perform calculations using 300-digit numbers, and these will suffice for an illustration.) Then $n = pq = 15481$ and $\varphi(n) = (p-1)(q-1) = 15232$. How can June find a suitable e? A prime larger than both p and q will certainly be relatively prime to $\varphi(n)$. June selects $e = 151$. She then sends the values of n and e to Bob.

As n has five digits, Bob knows that he must break his message into two-letter blocks. Suppose he wishes to send the message HOWDY. As the length is not a multiple of 2, he pads it out by adding a Q to the end. Now, HO is 0714, WD is 2203 and YQ is 2416. To encrypt the first message, Bob calculates

$$714^{151} \equiv 14628 \pmod{15481}.$$

Next, he calculates

$$2203^{151} \equiv 2494 \pmod{15481}$$

and

$$2416^{151} \equiv 8498 \pmod{15481}.$$

Bob sends June the messages 14628, 2494 and 8498.

June must calculate d. Since $(e, \varphi(n)) = 1$, we know that there exist $u, v \in \mathbb{Z}$ such that $eu + \varphi(n)v = 1$. Then d will be u (modulo $\varphi(n)$). The Euclidean algorithm shows us how to calculate d, and we find that $d = 807$. Thus, June calculates

$$14628^{807} \equiv 714 \pmod{15481},$$

$$2494^{807} \equiv 2203 \pmod{15481}$$

and

$$8498^{807} \equiv 2416 \pmod{15481}.$$

The original message was, therefore, 071422032416, which converts to HOWDYQ.

We should mention a couple of practical points. First, as any power of 0 is 0 and any power of 1 is 1, the messages 0 and 1 will not be encrypted using any RSA scheme. For that matter, since e will always be odd, $n - 1$ (which is -1 modulo n), will not change either. Given our method of encrypting English text, $n - 1$ will not arise, but 0 and 1 might. Can we do anything about it? Keep in mind that in the preceding example, we had $n = 15481$ but the possible messages would have been in the range of 0 to 2525. Would there be any harm in pushing them into the range of 2 to 2527? Surely not! Thus, we can agree to add 2 to every message before encrypting, and then subtract 2 after decrypting. (Do not do this in the exercises!)

Another point worth mentioning is that e should be reasonably large. To see why, note that in the example above, we could have used $e = 3$. This would be a problem if we sent a relatively small message. For example, if we had $m = 6$, the encrypted message would be $6^3 = 216$. No reduction modulo n takes place! An opponent who intercepted the message could simply take the cube root of 216 and recover the original message, without knowing anything about p and q. If e is large, we can ensure that this is avoided.

While modern ciphers are more complex than the RSA scheme, their security invariably rests upon the fact that it is very difficult to factor large numbers.

Exercises

Spaces have been deleted from all messages to be encrypted or decrypted. Where a letter is needed to pad out a block, use Q.

13.9. Suppose that someone foolishly used numbers as small as $n = 1961$ and $e = 43$ to create an RSA scheme. Crack the code by determining d.

13.10. If n is a product of two distinct primes, and both n and $\varphi(n)$ are known, show how to determine the two primes quickly. Illustrate the method using $n = 10961$, $\varphi(n) = 10752$.

13.11. Encrypt the message ALGEBRA using an RSA scheme with $n = 17399$ and $e = 149$.

13.12. Encrypt the message ABELIANGROUP using an RSA scheme with $n = 18203$ and $e = 191$.

13.13. Having set up an RSA scheme using $p = 103$, $q = 179$ and $e = 151$, we receive the following message: 2469, 7093, 14773, 10900, 143. Decrypt it.

13.14. Having set up an RSA scheme using $p = 89$, $q = 167$ and $e = 181$, we receive the following message: 13962, 8768, 7864, 4297, 12341. Decrypt it.