2.1 Introduction

This chapter introduces the basic concepts in **cryptography**. Our aim will be to demystify all the complicated terms related to this technology. After we are through with this chapter, we shall be ready to understand computer-based security solutions and issues that follow in later chapters.

> ***Cryptography** is the art and science of achieving security by encoding messages to make them non-readable.*

Figure 2.1 shows the conceptual view of cryptography.

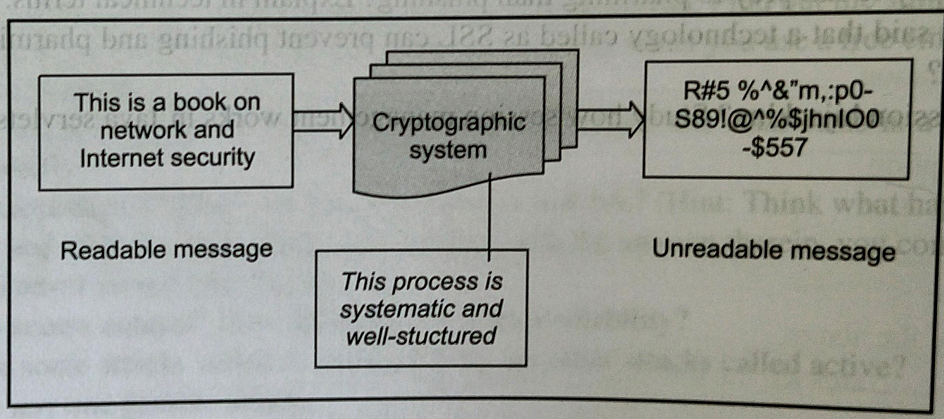

Fig. 2.1 *Cryptographic system*

Some more terms need to be introduced in this context.

> ***Cryptanalysis** is the technique of decoding messages from a non-readable format back to readable format without knowing how they were initially converted from readable format to non-readable format.*

In other words, it is like *breaking a code*. This concept is shown in Fig. 2.2.

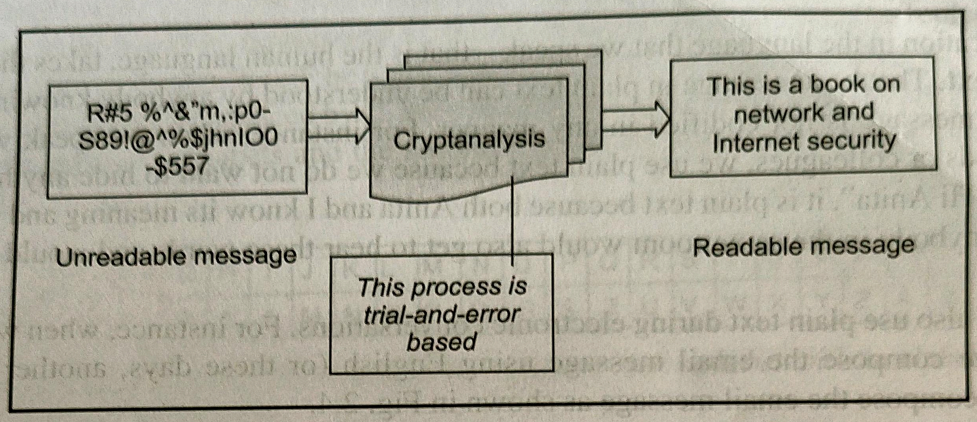

Fig. 2.2 *Cryptanalysis*

***Cryptology** is a combination of cryptography and cryptanalysis.*

This concept is shown in Fig. 2.3.

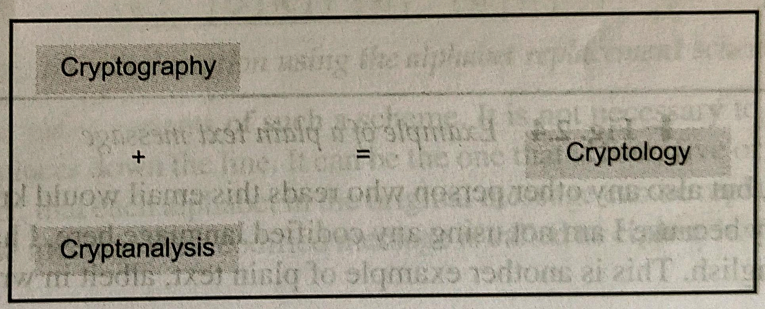

Fig. 2.3 *Cryptography + Cryptanalysis = Cryptology*

In the early days, cryptography used to be performed by using manual techniques. The basic framework of performing cryptography has remained more or less the same, of course, with a lot of improvements in the actual implementation. More importantly, computers now perform these cryptographic functions/algorithms, thus making the process a lot faster and secure. This chapter, however, discusses the basic methods of achieving cryptography without referring to computers.

The basic concepts in cryptography are introduced first. We then proceed to discuss how we can make messages illegible and thus, secure. This can be done in many ways. We discuss all these approaches in this chapter. Modern computer-based cryptography solutions have actually evolved based on these premises. This chapter touches upon all these cryptographic algorithms. We also discuss the relative advantages and disadvantages of the various algorithms, as and when applicable.

Some cryptographic algorithms are very trivial to understand, replicate and therefore, crack. Some other cryptographic algorithms are highly complicated and therefore, difficult to crack. The rest are somewhere in the middle. A detailed discussion of these is highly essential in cementing our concepts that we shall keep referring to when we actually discuss computer-based cryptography solutions in later chapters.

2.2 Plain Text and Cipher Text

Any communication in the language that we speak – that is the human language, takes the form of **plain text** or **clear text**. That is, a message in plain text can be understood by anybody knowing the language as long as the message is not codified in any manner. For instance, when we speak with our family members, friends or colleagues, we use plain text because we do not want to hide anything from them. Suppose I say "Hi Anita", it is plain text because both Anita and I know its meaning and intention. More significantly, anybody in the same room would also get to hear these words and would know that I am greeting Anita.

Notably, we also use plain text during electronic conversations. For instance, when we send an email to someone, we compose the email message using English (or these days, another) language. For instance, I can compose the email message as shown in Fig. 2.4.

Hi Amit

Hope you are doing fine. How about meeting at the train station this Friday at 5 pm?
Please let me know if it is ok with you.

Regards,

Atul

Fig. 2.4 *Example of a plain text message*

Now, not only Amit, but also any other person who reads this email would know what I have written. As before, this is simply because I am not using any codified language here. I have composed my email message using plain English. This is another example of plain text, albeit in written form.

> *Clear text or plain text signifies a message that can be understood by the sender, the recipient and also by anyone else who gets an access to that message.*

In normal life, we do not bother much about the fact that someone could be overhearing us. In most cases, that makes little difference to us because the person overhearing us can do little damage by using the overheard information. After all, we do not reveal many secrets in our day-to-day lives.

However, there are situations where we are concerned about the secrecy of our conversations. For instance, suppose that I am interested in knowing my bank account's balance and hence I call up my phone banker from my office. The phone banker would generally ask a secret question (e.g. What is your grandmother's maiden name?) whose answer only I know. This is to ascertain that someone else is not posing as me. Now, when I give the answer to the secret question (e.g. Leela), I generally speak in low voice or better yet, initially call up from a phone that is isolated. This ensures that only the intended recipient (the phone banker) gets to know the correct answer.

On the same lines, suppose that my email to my friend Amit shown earlier is confidential for some reason. Therefore, I do not want anyone else to understand what I have written even if she is able to access the email by using some means, before it reaches Amit. How do I ensure this? This is exactly the problem that small children face. Many times, they want to communicate in such a manner that their little secrets are hidden from the elderly. What do they do in order to achieve this? Usually the simplest

trick that they use is a code language. For instance, they replace each alphabet in their conversation with another one. As an example, they replace each alphabet with the alphabet that is actually three alphabets down the order. So, each A will be replaced by D, B will be replaced by E, C will be replaced by F and so on. To complete the cycle, each W will be replaced by Z, each X will be replaced by A, each Y will be replaced by B and each Z will be replaced by C. We can summarize this scheme as shown in Fig. 2.5. The first row shows the original alphabets and the second row shows what each original alphabet will be replaced with.

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

Fig. 2.5 *A scheme for codifying messages by replacing each alphabet with an alphabet three places down the line*

Thus, using the scheme of replacing each alphabet with the one that is three places down the line, a message *I love you* shall become *L ORYH BRX* as shown in Fig. 2.6.

I		L	O	V	E		Y	O	U
L		O	R	Y	H		B	R	X

Fig. 2.6 *Codification using the alphabet replacement scheme*

Of course, there can be many variants of such a scheme. It is not necessary to replace each alphabet with the one that is three places down the line. It can be the one that is four, five or more places down the line. The point is, however, that each alphabet in the original message can be replaced by another to hide the original contents of the message. The codified message is called as **cipher text**. Cipher means a code or a secret message.

When a plain text message is codified using any suitable scheme, the resulting message is called as cipher text.

Based on these concepts, let us put these terms into a diagrammatic representation, as shown in Fig. 2.7.

Let us now write our original email message and the resulting cipher text by using the alphabet-replacing scheme, as shown in Fig. 2.8. This will clarify the idea further.

As shown in Fig. 2.9, there are two primary ways in which a plain text message can be codified to obtain the corresponding cipher text: **Substitution** and **Transposition**.

Let us discuss these two approaches now. Note that when the two approaches are used together, we call the technique as **product cipher**.

Fig. 2.45 *Cryptography techniques*

Symmetric Key Cryptography involves the usage of the same key for encryption and decryption. Asymmetric Key Cryptography involves the usage of one key for encryption and another, different key for decryption.

2.6 Symmetric and Asymmetric Key Cryptography

2.6.1 Symmetric Key Cryptography and the Problem of Key Distribution

Before we discuss computer-based symmetric and asymmetric key cryptographic algorithms (in the next few chapters), we need to understand why we need two different types of cryptographic algorithms in the first place. To understand this, let us consider a simple problem statement.

> *Person A wants to send a highly confidential letter to another person B. A and B both reside in the same city, but are separated by a few miles and for some reason, cannot meet each other.*

Let us now see how we can tackle this problem. The simplest solution would appear to be that A puts the confidential letter in an envelope, seals it and sends it by post. A hopes that no one opens it before it reaches B. This is shown in Fig. 2.46.

Clearly, this solution does not seem to be acceptable. What is the guarantee that an unscrupulous person does not obtain and open the envelope before it reaches B? Sending the envelope by registered post or courier might slightly improve the situation, but will not guarantee that the envelope does not get opened before it reaches B. After all, someone can open the envelope, read the confidential letter and re-seal the envelope!

Another option is to send the envelope via a hand-delivery mechanism. Here, A hands the envelope over to another person P, who personally hand-delivers the envelope to B. This seems to be a slightly better solution. However, it is still not fool proof.

Consequently, A comes up with another idea. A now puts the envelope inside a box, seals that box with a highly secure lock and sends the box to B (through the mechanism of post/courier/hand-delivery). Since the lock is highly secure, nobody can open the box while in transit and therefore, open the envelope. Consequently, nobody will be able to read/access the highly confidential letter! The problem is resolved! If we think about it, we will realize that the problem indeed seems to be resolved. However,

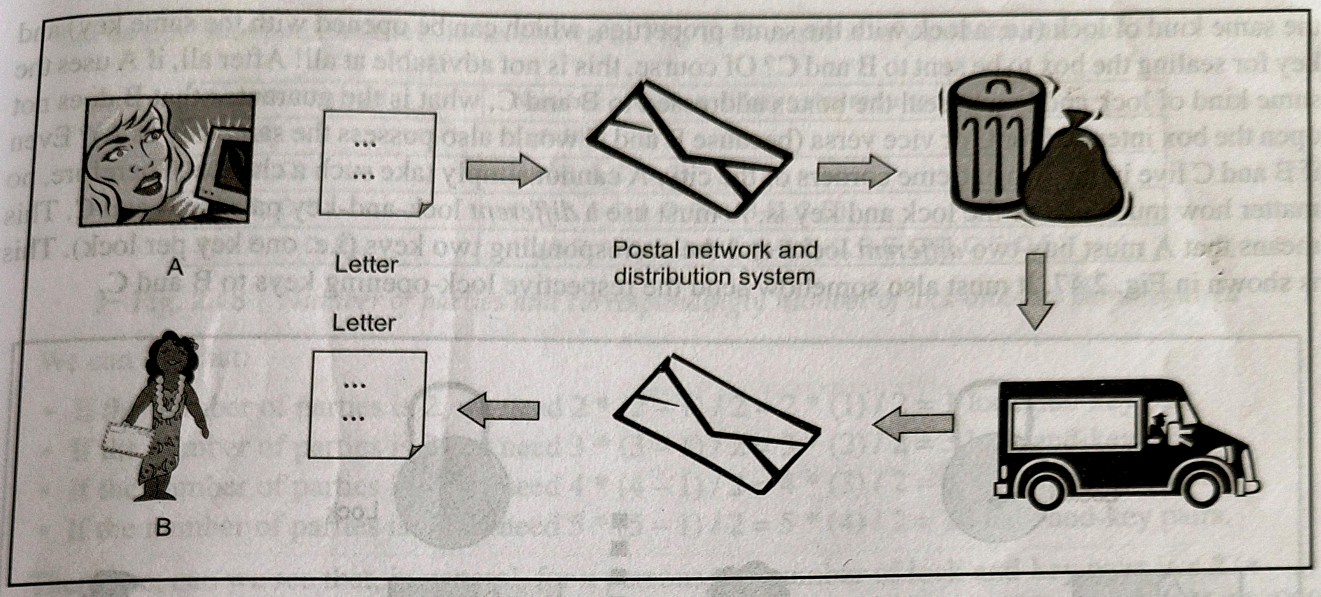

Fig. 2.46 *Simplest way to send a confidential letter*

this solution has given birth to a new problem. How on earth can the intended recipient (B) now open the box and therefore, the envelope? This solution has not only prevented unauthorized access to the letter, but also the authorized access. That is, even B would not be able to open the lock. This defeats the purpose of sending the letter in this manner, in the first place.

What if A also send the key of the lock along with the box, so that B can open the lock and get access to the envelope inside the box and hence, the letter? This seems absurd. If the key travels with the box, anybody who has access to the box in transit (e.g. P) can unlock and open the box.

Therefore, A now comes up with an improved plan. A decides that the locked box should travel to B as discussed (by post/courier/hand-delivery). However, she will not send the key used to lock the box along with the box. Instead, she will decide a place and a time to meet B in person, meet B at that time and hand over the key personally to B. This will ensure that the key does not land up in the wrong hands and that only B can access the confidential letter! This now seems to be a full-proof solution! Is it, really?

If A can meet B in person to hand over the key, she can as well hand the confidential letter to B in person! Why have all these additional worries and overheads? Remember that the whole problem started because A and B cannot, for some reason, meet in person!

As a result, we will observe that no solution is completely acceptable. Either it is not fool proof or is not practically possible. This is the problem of **key distribution** or **key exchange**. Since the sender and the receiver will use the same key to lock and unlock, this is called as *symmetric key operation* (when used in the context of cryptography, this operation is called as **symmetric key cryptography**). Thus, we observe that the key distribution problem is inherently linked with the symmetric key operation.

Let us now imagine that not only A and B but also thousands of people want to send such confidential letters securely to each other. What would happen if they decide to go for symmetric key operation? If we examine this approach more closely, we can see that it has one big drawback if the number of people that want to avail of its services is very large.

We will start with small numbers and then inspect this scheme for a larger number of participants. For instance, let us assume that A now wants to communicate with two persons, B and C securely. Can A use

the same kind of lock (i.e. a lock with the same properties, which can be opened with the same key) and key for sealing the box to be sent to B and C? Of course, this is not advisable at all! After all, if A uses the same kind of lock and key to seal the boxes addressed to B and C, what is the guarantee that B does not open the box intended for C or vice versa (because B and C would also possess the same key as A)? Even if B and C live in the two extreme corners of the city, A cannot simply take such a chance! Therefore, no matter how much secure the lock and key is, A must use a *different* lock-and-key pair for B and C. This means that A must buy two *different* locks and the corresponding two keys (i.e. one key per lock). This is shown in Fig. 2.47. It must also somehow send the respective lock-opening keys to B and C.

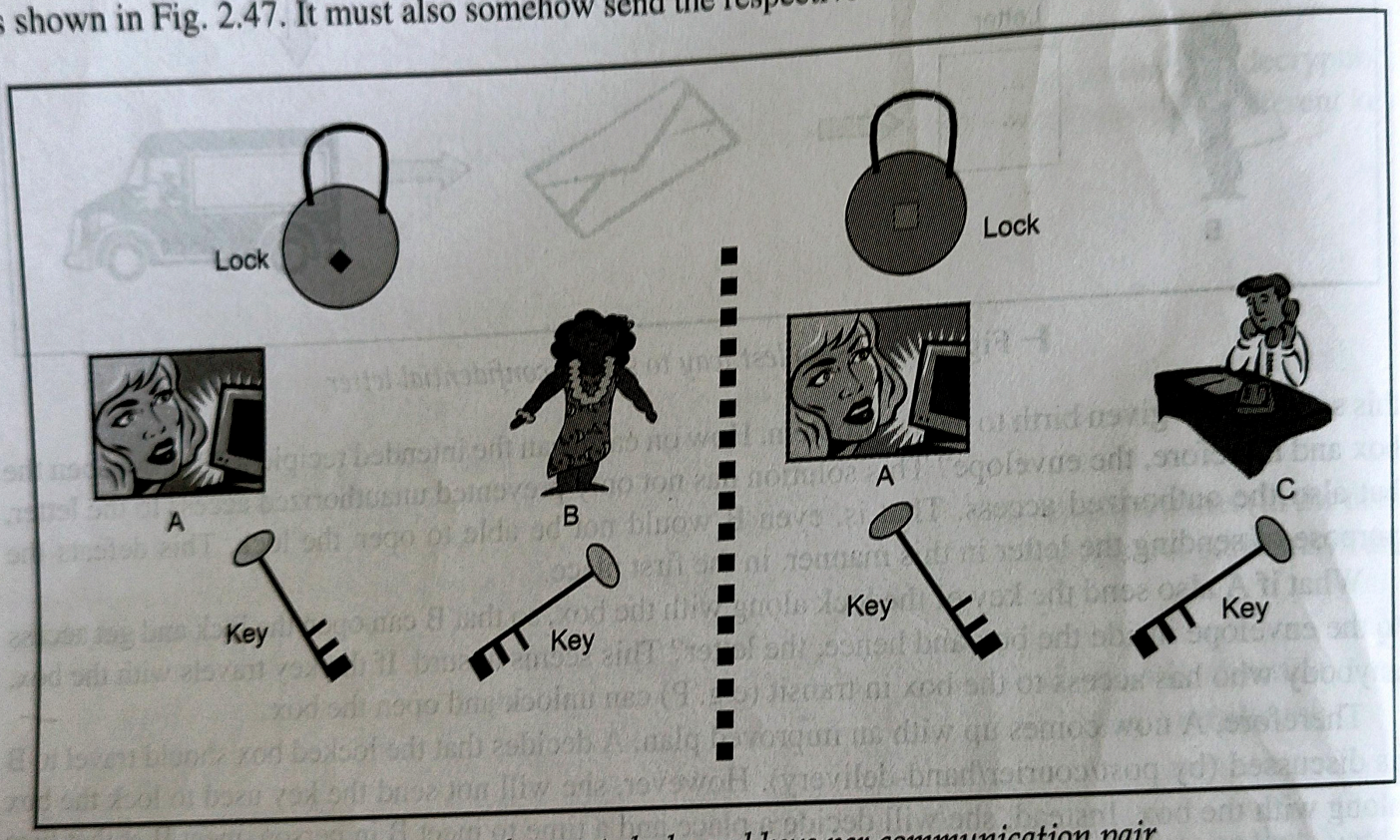

Fig. 2.47 *Use of separate locks and keys per communication pair*

Thus, we have the following situation:

- When A wanted to communicate only with B, we needed one lock-and-key pair (A-B).
- When A wants to communicate with B and C, we need two lock-and-key pairs (A-B and A-C). Thus, we need one lock-and-key pair per person with whom A wants to communicate. If B also wants to communicate with C, we have B-C as the third communicating pair, requiring its own lock-and-key pair. Thus, we would need three lock-and-key pairs to serve the needs of three communicating pairs.
- Let us consider the participation of a fourth person D. Let us also imagine that all of the four persons (A, B, C and D) want to be able to communicate with each other securely. Thus, we have six communicating pairs, namely A-B, A-C, A-D, B-C, B-D and C-D. Thus, we need six lock-and-key pairs, one per communicating pair, to serve the needs of four communicating pairs.
- If E is the fifth person joining this group, we have ten communicating pairs, namely A-B, A-C, A-D, A-E, B-C, B-D, B-E, C-D, C-E and D-E. Thus, we would need ten lock-and-key pairs to make secure communication between all these pairs possible.

Let us now tabulate these results as shown in Fig. 2.48 to see if any pattern emerges.

Parties involved	Number of lock-and-key pairs required
2 (A, B)	1 (A-B)
3 (A, B, C)	3 (A-B, A-C, B-C)
4 (A, B, C, D)	6 (A-B, A-C, A-D, B-C, B-D, C-D)
5 (A, B, C, D, E)	10 (A-B, A-C, A-D, A-E, B-C, B-D, B-E, C-D, C-E, D-E)

Fig. 2.48 *Number of parties and correspondingly number of lock-and-key pairs required*

We can see that:

- If the number of parties is 2, we need 2 * (2 – 1) / 2 = 2 * (1) / 2 = 1 lock-and-key pair.
- If the number of parties is 3, we need 3 * (3 – 1) / 2 = 3 * (2) / 2 = 3 lock-and-key pairs.
- If the number of parties is 4, we need 4 * (4 – 1) / 2 = 4 * (3) / 2 = 6 lock-and-key pairs.
- If the number of parties is 5, we need 5 * (5 – 1) / 2 = 5 * (4) / 2 = 10 lock-and-key pairs.

Therefore, can we see that, in general, for n persons, the number of lock-and-key pairs is $n * (n - 1) / 2$? Now, if we have about 1,000 persons in this scheme, we will have 1000 * (1000 – 1) / 2 = 1000 * (999) / 2 = 99,9000 / 2 = 499,500 lock-and-key pairs!

Moreover, we must keep in mind that a record of which lock-and-key pair was issued to which communicating pair must be maintained by somebody. Let us call this somebody as T. This is required because it is quite possible that some persons might lose the lock or key or both. In such cases, T must ensure that the proper duplicate key is issued or that the lock is replaced with an exact replica of the lock or that a different lock and key pair is issued (for security reasons), depending on the situation. This is quite a bit of task! Also, who is T, after all? T must be highly trustworthy and accessible to everybody. This is because each communicating pair has to approach T to obtain the lock-and-key pair. This is quite a tedious and time-consuming process!

work, an alternative emerges.

2.6.3 Asymmetric Key Operation

In this scheme, A and B do not have to jointly approach T for a lock-and-key pair. Instead, B alone approaches T, obtains a lock and a key (K1) that can seal the lock and sends the lock and key K1 to A. B tells A that A can use that lock and key to seal the box before sending the sealed box to B. How can B open the lock, then?

An interesting property of this scheme is that B possesses a *different but related* key (K2), which is obtained by B from T along with the lock and key K1, only which can open the lock. It is guaranteed that no other key and of course, including the one used by A (i.e. K1) for locking, can open the lock. Since one key (K1) is used for locking and *another, different* key (K2) is used for unlocking; we will call this scheme as *asymmetric key operation*. Also, T is clearly defined here as a **trusted third party**. T is certified as a highly trustworthy and efficient agency by the government.

This means that B possesses a **key pair** (i.e. two keys K1 and K2). One key (i.e. K1) can be used for locking and only the corresponding other key (i.e. K2) from the key pair can be used for unlocking. Thus B can send the lock and key K1 to anybody (e.g. A) who wants to send anything securely to B. B would request the sender (e.g. A) to use that lock and key K1 to seal the contents. B can then open the seal using the key K2. Since the key K1 is meant for locking and is available to the general public, we shall call K1 as **public key**. Note that K1 need not be secret – in fact, it *should not be* secret! Thus, unlike what happens in the case of symmetric key operation, the (locking) key need not be guarded secretly now. The other key K2 is meant for unlocking and is strictly held secret/private by A. Therefore, we shall call it as **private key** or **secret key**.

This is shown in Fig. 2.59.

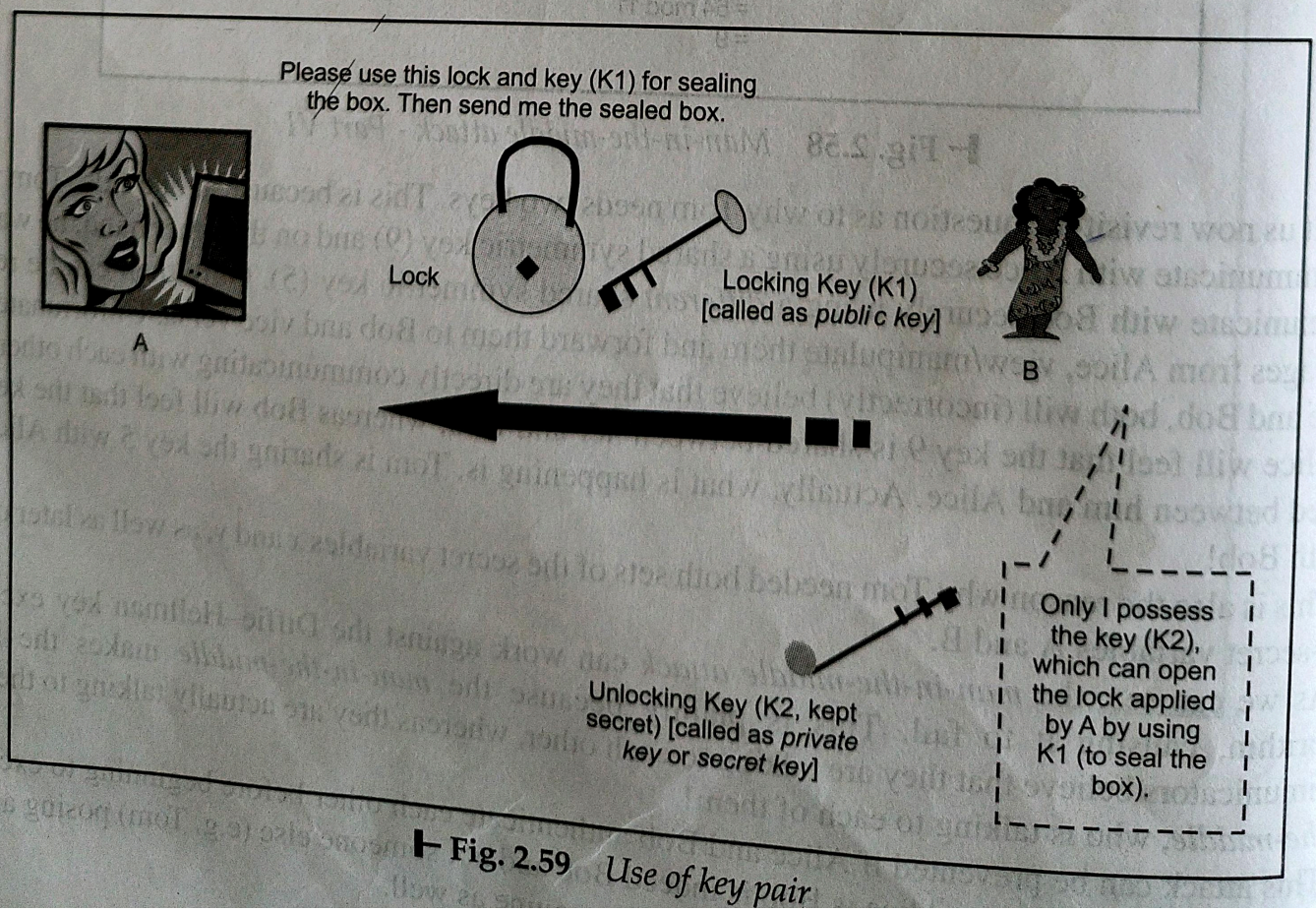

Fig. 2.59 *Use of key pair*

Note that if B wants to receive something securely from another person say C, B need *not* obtain a fresh lock-and-key pair. B can send *the same* lock-and-key (K1) pair (or a copy of the lock and B's *public key* K1, in case A and C want to send something securely to B at the same time) to C. Thus, C will also use the same lock and B's *public key* K1 to seal the contents before sending them to B. As before, B will use the corresponding *private key* K2 to open the lock. Extending this concept a Step further, if B wants to receive messages securely from 10,000 different persons, B can send the same lock-and-public key K1 (or their copies) to each one of them! It need not have 10,000 unique locks and keys (unlike the symmetric key approach)! Always, B's public key K1 will be used by the sender for locking and B's private key K2 will be used for unlocking by the receiver (i.e. B).

Clearly, this is an extremely convenient approach, as compared to symmetric key operation!

Let us now consider what happens if three persons A, B and C want to communicate with each other. That is, A, B and C must all be able to send/receive messages securely to/from each other. For this to be possible, all the three persons can obtain a lock-and-public key pair from the trusted third party (T). Whenever any one of them wants to receive a message securely from another person, she has to send her lock-and-public key to the sender. That is, when A wants to receive a message securely from B, A sends her lock and public key to B. B can use that to seal the message and send the sealed message to A. A can then use her private key to open the lock. Similarly, when B wants to receive a message securely from A, B sends her lock and public key to A, using which B can secure the message. Since only B has her own private key, she can open the lock and access the message.

Extending this basic idea, if 1,000 people want to be able to securely communicate with each other, only 1,000 locks, 1,000 public keys and the corresponding 1,000 private keys are required. This is in stark contrast to the symmetric key operation wherein for 1,000 participants, we needed 499,500 lock-and-key pairs (please refer to our earlier discussion).

Therefore, in general, when using asymmetric key operation; the recipient has to send the lock and her public key to the sender. The sender uses these to apply the lock and sends the sealed contents to the recipient. The recipient uses her private key to open the lock. Since only the recipient possesses the private key, all concerned are assured that only the intended recipient can open the lock.