3

Key Distribution

Note: Sections 3.1 and 3.2 are based on the first edition of "Cryptography Theory and Practice", which was published in 1995, and they have not been updated for this course.

3.1 Introduction

We have observed that public-key systems have the advantage over secret-key systems that a secure channel is not needed to exchange a key. But, unfortunately, most public-key cryptosystems are much slower than secret-key systems such as *AES*, for example. So, in practice, secret-key systems are usually used to encrypt "long" messages. But then we come back to the problem of exchanging keys.

In this chapter, we discuss several approaches to the problem of establishing secret keys. We will distinguish between key distribution and key agreement. *Key distribution* is defined to be a mechanism whereby one party chooses a secret key and then transmits it to another party or parties. *Key agreement* denotes a protocol whereby two (or more) parties jointly establish a secret key by communicating over a public channel. In a key agreement scheme, the value of the key is determined as a function of inputs provided by both parties.

As our setting, we have an insecure network of n users. In some of our schemes, we will have a *trusted authority* (denoted by TA) that is reponsible for such things as verifying the identities of users, choosing and transmitting keys to users, etc.

Since the network is insecure, we need to protect against potential opponents. Our opponent, Oscar, might be a *passive adversary*, which means that his actions are restricted to eavesdropping on messages that are transmitted over the channel. On the other hand, we might want to guard against the possibility that Oscar is an *active adversary*. An active adversary can do various types of nasty things such as the following:

1. alter messages that he observes being transmitted over the network

2. save messages for reuse at a later time
3. attempt to masquerade as various users in the network.

The objective of an active adversary might be one of the following:

1. to fool U and V into accepting an "invalid" key as valid (an invalid key could be an old key that has expired, or a key chosen by the adversary, to mention two possibilities)
2. to make U or V believe that they have exchanged a key with other when they have not.

The objective of a key distribution or key agreement protocol is that, at the end of the protocol, the two parties involved both have possession of the same key K, and the value of K is not known to any other party (except possibly the TA). Certainly it is much more difficult design a protocol providing this type of security in the presence of an active adversary as opposed to a passive one.

We first consider the idea of *key predistribution* in Section 3.2. For every pair of users $\{\mathrm{U}, \mathrm{V}\}$, the TA chooses a random key $K_{\mathrm{U,V}} = K_{\mathrm{V,U}}$ and transmits it "off-band" to U and V over a secure channel. (That is, the transmission of keys does not take place over the network, since the network is not secure.) This approach is unconditionally secure, but it requires a secure channel between the TA and every user in the network. But of possibly even more significance is the fact that each user must store $n - 1$ keys, and the TA needs to transmit a total of $\binom{n}{2}$ keys securely (this is sometimes called the "n^2 problem"). Even for relatively small networks, this can become prohibitively expensive, and thus it is not really a practical solution.

In Section 3.2.1, we discuss an interesting unconditionally secure key predistribution scheme, due to Blom, that allows a reduction in the amount of secret information to be stored by the users in the network. We also present in Section 3.3 a computationally secure key predistribution scheme based on the discrete logarithm problem.

A more practical approach can be described as *on-line key distribution by TA*. In such a scheme, the TA acts as a *key server*. The TA shares a secret key K_{U} with every user U in the network. When U wishes to communicate with V, she requests a *session key* from the TA. The TA generates a session key K and sends it in encrypted form for U and V to decrypt. The well-known **Kerberos** system, which we describe in Section 3.4, is based on this approach.

If it is impractical or undesirable to have an on-line TA, then a common approach is to use a *key agreement protocol*. In a key agreement protocol, U and V jointly choose a key by communicating over a public channel. This remarkable idea is due to Diffie and Hellman, and (independently) to Merkle. We describe a few of the more popular key agreement protocols. A variation of the original protocol of Diffie and Hellman, modified to protect against an active adversary, is presented in Section 4.1.1. Two other interesting protocols are also discussed: the

MTI scheme is presented in Section 4.1.2 and the **Girault** scheme is covered in Section 4.1.3.

3.2 Key Predistribution

In the basic method, the TA generates $\binom{n}{2}$ keys, and gives each key to a unique pair of users in a network of n users. As mentioned above, we require a secure channel between the TA and each user to transmit these keys. This is a significant improvement over each pair of users independently exchanging keys over a secure channel, since the number of secure channels required has been reduced from $\binom{n}{2}$ to n. But if n is large, this solution is not very practical, both in terms of the amount of information to be transmitted securely, and in the amount of information that each user must store securely (namely, the secret keys of the other other $n-1$ users).

Thus, it is of interest to try to reduce the amount of information that needs to be transmitted and stored, while still allowing each pair of users U and V to be able to (independently) compute a secret key $K_{\mathrm{U,V}}$. An elegant scheme to accomplish this, called the **Blom Key Predistribution Scheme**, is discussed in the next subsection.

3.2.1 Blom's Scheme

As above, we suppose that we have a network of n users. For convenience, we suppose that keys are chosen from a finite field $\mathbb{Z}_p$, where $p \geq n$ is prime. Let k be an integer, $1 \leq k \leq n-2$. The value k is the largest size coalition against which the scheme will remain secure. In the **Blom Scheme**, the TA will transmit $k+1$ elements of $\mathbb{Z}_p$ to each user over a secure channel (as opposed to $n-1$ in the basic key predistribution scheme). Each pair of users U and V will be able to compute a key $K_{\mathrm{U,V}} = K_{\mathrm{V,U}}$, as before. The security condition is as follows: any set of at most k users disjoint from $\{\mathrm{U}, \mathrm{V}\}$ must be unable to determine any information about $K_{\mathrm{U,V}}$ (note that we are speaking here about unconditional security).

We first present the special case of Blom's scheme where $k = 1$. Here, the TA will transmit two elements of $\mathbb{Z}_p$ to each user over a secure channel, and any individual user W will be unable to determine any information about $K_{\mathrm{U,V}}$ if $\mathrm{W} \neq \mathrm{U}, \mathrm{V}$. Blom's scheme is presented in Figure 3.1. We illustrate the **Blom Scheme** with $k = 1$ in the following example.

Example 3.1 Suppose the three users are U, V and W, $p = 17$, and their public elements are $r_{\mathrm{U}} = 12$, $r_{\mathrm{V}} = 7$ and $r_{\mathrm{W}} = 1$. Suppose that the TA chooses $a = 8$, $b = 7$ and $c = 2$, so the polynomial f is

$$f(x, y) = 8 + 7(x + y) + 2xy.$$

The g polynomials are as follows:

$$g_{\mathrm{U}}(x) = 7 + 14x$$
$$g_{\mathrm{V}}(x) = 6 + 4x$$
$$g_{\mathrm{W}}(x) = 15 + 9x.$$

The three keys are thus

$$K_{\mathrm{U,V}} = 3$$
$$K_{\mathrm{U,W}} = 4$$
$$K_{\mathrm{V,W}} = 10.$$

U would compute $K_{\mathrm{U,V}}$ as

$$g_{\mathrm{U}}(r_{\mathrm{V}}) = 7 + 14 \times 7 \bmod 17 = 3$$

V would compute $K_{\mathrm{U,V}}$ as

$$g_{\mathrm{V}}(r_{\mathrm{U}}) = 6 + 4 \times 12 \bmod 17 = 3.$$

We leave the computation of the other keys as an exercise for the reader. ▯

We now prove that no one user can determine any information about the key of two other users.

THEOREM 3.1 *The* **Blom Scheme** *with* $k = 1$ *is unconditionally secure against any individual user.*

PROOF Let's suppose that user W wants to try to compute the key

$$K_{\mathrm{U,V}} = a + b(r_{\mathrm{U}} + r_{\mathrm{V}}) + cr_{\mathrm{U}}r_{\mathrm{V}} \bmod p.$$

The values r_{U} and r_{V} are public, but a, b and c are unknown. W does know the values

$$a_{\mathrm{W}} = a + br_{\mathrm{W}} \bmod p$$

and

$$b_{\mathrm{W}} = b + cr_{\mathrm{W}} \bmod p$$

since these are the coefficients of the polynomial $g_{\mathrm{W}}(x)$ that was sent to W by the TA.

What we will do is show that the information known by W is consistent with any possible value $\ell \in \mathbb{Z}_p$ of the key $K_{\mathrm{U,V}}$. Hence, W cannot rule out any values for $K_{\mathrm{U,V}}$. Consider the following matrix equation (in $\mathbb{Z}_p$):

$$\begin{pmatrix} 1 & r_{\mathrm{U}} + r_{\mathrm{V}} & r_{\mathrm{U}}r_{\mathrm{V}} \\ 1 & r_{\mathrm{W}} & 0 \\ 0 & 1 & r_{\mathrm{W}} \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \ell \\ a_{\mathrm{W}} \\ b_{\mathrm{W}} \end{pmatrix}.$$

The first equation represents the hypothesis that $K_{\mathrm{U,V}} = \ell$; the second and third equations contain the information that W knows about a, b and c from $g_{\mathrm{W}}(x)$.

The determinant of the coefficient matrix is

$$r_{\mathrm{W}}{}^2 + r_{\mathrm{U}} r_{\mathrm{V}} - (r_{\mathrm{U}} + r_{\mathrm{V}}) r_{\mathrm{W}} = (r_{\mathrm{W}} - r_{\mathrm{U}})(r_{\mathrm{W}} - r_{\mathrm{V}}),$$

where all arithmetic is done in $\mathbb{Z}_p$. Since $r_{\mathrm{W}} \neq r_{\mathrm{U}}$ and $r_{\mathrm{W}} \neq r_{\mathrm{V}}$, it follows that the coefficient matrix has non-zero determinant, and hence the matrix equation has a unique solution for a, b, c. In other words, any possible value ℓ of $K_{\mathrm{U,V}}$ is consistent with the information known to W. ▮

On the other hand, a coalition of two users, say $\{\mathrm{W}, \mathrm{X}\}$, will be able to determine any key $K_{\mathrm{U,V}}$ where $\{\mathrm{W}, \mathrm{X}\} \cap \{\mathrm{U}, \mathrm{V}\} = \emptyset$. W and X together know that

$$\begin{aligned}
a_{\mathrm{W}} &= a + b r_{\mathrm{W}} \\
b_{\mathrm{W}} &= b + c r_{\mathrm{W}} \\
a_{\mathrm{X}} &= a + b r_{\mathrm{X}} \\
b_{\mathrm{X}} &= b + c r_{\mathrm{X}}.
\end{aligned}$$

Thus they have four equations in three unknowns, and they can easily compute a unique solution for a, b and c. Once they know a, b and c, they can form the polynomial $f(x, y)$ and compute any key they wish.

It is straightforward to generalize the scheme to remain secure against coalitions of size k. The only thing that changes is step 2. The TA will use a polynomial $f(x, y)$ having the form

$$f(x, y) = \sum_{i=0}^{k} \sum_{j=0}^{k} a_{i,j}\, x^i y^j \bmod p,$$

where $a_{i,j} \in \mathbb{Z}_p$ ($0 \leq i \leq k, 0 \leq j \leq k$), and $a_{i,j} = a_{j,i}$ for all i, j. The remainder of the protocol is unchanged.

FIGURE 3.1
Blom's key distribution scheme ($k = 1$)

1. A prime number p is made public, and for each user U, an element $r_{\mathrm{U}} \in \mathbb{Z}_p$ is made public. The elements r_{U} must be distinct.
2. The TA chooses three random elements $a, b, c \in \mathbb{Z}_p$ (not necessarily distinct), and forms the polynomial

$$f(x, y) = a + b(x + y) + cxy \bmod p.$$

3. For each user U, the TA computes the polynomial

$$g_{\mathrm{U}}(x) = f(x, r_{\mathrm{U}}) \bmod p$$

and transmits $g_{\mathrm{U}}(x)$ to U over a secure channel. Note that $g_{\mathrm{U}}(x)$ is a linear polynomial in x, so it can be written as

$$g_{\mathrm{U}}(x) = a_{\mathrm{U}} + b_{\mathrm{U}}x,$$

where

$$a_{\mathrm{U}} = a + br_{\mathrm{U}} \bmod p$$

and

$$b_{\mathrm{U}} = b + cr_{\mathrm{U}} \bmod p.$$

4. If U and V want to communicate, then they use the common key

$$K_{\mathrm{U,V}} = K_{\mathrm{V,U}} = f(r_{\mathrm{U}}, r_{\mathrm{V}}) = a + b(r_{\mathrm{U}} + r_{\mathrm{V}}) + cr_{\mathrm{U}}r_{\mathrm{V}} \bmod p,$$

where U computes $K_{\mathrm{U,V}}$ as

$$f(r_{\mathrm{U}}, r_{\mathrm{V}}) = g_{\mathrm{U}}(r_{\mathrm{V}})$$

and V computes $K_{\mathrm{U,V}}$ as

$$f(r_{\mathrm{U}}, r_{\mathrm{V}}) = g_{\mathrm{V}}(r_{\mathrm{U}}).$$