At this point, both Alice and Bob know that they are sharing the same random key *k*. To see that this system is immune to offline dictionary attacks, look at each exchange. Because the data sent in each exchange is randomly selected and never visible to the attacker in plaintext form, the attacker cannot know when she has correctly deciphered the message.

13.7 Biometrics

Identification by physical characteristics is as old as humanity. Recognizing people by their voices or appearance, and impersonating people by assuming their appearance, was widely known in classical times. Efforts to find physical characteristics that uniquely identify people include the Bertillion cranial maps, fingerprints, and DNA sampling. Using such a feature to identify people for a computer would ideally eliminate errors in authentication.

Biometrics is the automated measurement of biological or behavioral features that identify a person [1343]. When a user is given an account, the system administration takes a set of measurements that identify that user to an acceptable degree of error. Whenever the user accesses the system, the biometric authentication mechanism verifies the identity. Lawton [1143] points out that this is considerably easier than identifying the user because no searching is required. A comparison to the known data for the claimed user's identity will either verify or reject the claim. Characteristics used are fingerprints, voice characteristics, eyes, facial features, keystroke dynamics, and other personal attributes [957].

Because biometrics are measurements of the characteristics of the individual, people are tempted to believe that attackers cannot pose as authorized users on systems that use biometrics. Several assumptions underlie this belief [1567, 1686].

1. The biometric data is initialized properly. This means that the biometric data is that of the person whose identity it is bound to. If, for example, Ann's fingerprint is listed as being Penny's, then the biometric device will incorrectly identify Ann as Penny.
2. The biometric device is accurate in the environment in which it is used. For example, if a fingerprint scanner is under observation, having it scan a mask of another person's finger would be detected. But if it is not under observation, such a trick might not be detected and the unauthorized user might gain access.
3. The methods and algorithms by which the input biometric is compared to the stored biometrics only return a successful match when the two biometrics belong to the same person. The problem here is that most biometrics vary between measurements. The comparison must take these variations into account. If the algorithm requires too precise a match, the biometric validator might return a false negative. Conversely, if the

algorithm accepts too large a variance, the validator might return a false positive. Either of these situations compromises security.

4. The stored biometric data and the software validating the biometric input has not been corrupted. If the former has been, Ann's stored biometric data may be replaced with Penny's, so Penny is incorrectly identified as Ann. Corrupting the software can cause it to return a match when there is no match, or vice versa.
5. The transmission from the biometric device to the computer's analysis process is tamperproof. Otherwise, one could corrupt the transmission, causing a variety of security problems.
6. The transmission from the biometric device to the computer's analysis process is not a replay. Otherwise, one could record a legitimate authentication and replay it later to gain access.

If any of these assumptions do not hold, then an attacker can authenticate as someone else. Thus, biometric mechanisms must ensure these assumptions are satisfied to the greatest possible degree. Exercise 13 explores these in more detail.

We now briefly explore some commonly used biometrics.

13.7.1 Fingerprints

Fingerprints can be measured in a variety of ways, and new ones are being developed. Two examples will show how they work.

Optical devices use cameras. The finger is placed onto a clear, lighted surface, usually a prism. The prism is illuminated, and the light reflects off the surface where the fingerprint rests to a camera. The fingerprint ridges obscure the rays from the light source, causing the ridges to appear as dark parts of the image [916]. Feature extraction methods then build a representation of the fingerprint that can be stored.

A capacitative technique uses the differences in electrical charges of the whorls on the finger to detect those parts of the finger touching a chip and those raised. The data is converted into a graph in which ridges are represented by vertices and vertices corresponding to adjacent ridges are connected. Each vertex has a number approximating the length of the corresponding ridge. At this point, determining matches becomes a problem of graph matching [949]. This problem is similar to the classical graph isomorphism problem, but because of imprecision in measurements, the graph generated from the fingerprint may have different numbers of edges and vertices. Thus, the matching algorithm is an approximation.

Example: Apple's Touch ID system on iPhones, iPads, and some laptops uses a capacitative system to gather data from the finger pressing on the home button [2114].

A technique that uses finger vein biometrics captures an image of the finger using an infrared camera. This shows the veins and shades produced by other internal structures such as bones and muscles. An image of the attributes is then extracted. One way is to use an adaptive technique that determines the threshold for the light parts of the image (veins, etc.), clarifying the image. Then various noise reduction algorithms remove irregularities, and the resulting image is translated into the storage form. Experiments show this method achieves a very high identification rate, with few false recognitions [1399]. Another method locates the valley-like structures in the image (the veins), and compares the number of matching pixels in the image with the number of pixels in the patterns. This method also shows a low error rate in experiments [1789].

Like other authentication mechanisms, fingerprints can be spoofed [1920], leading to the development of countermeasures [1249].

13.7.2 Voices

Authentication by voice, also called *speaker verification* or *speaker recognition*, involves recognition of a speaker's voice characteristics [343] or verbal information verification [1170, 1171]. The former uses statistical techniques to test the hypothesis that the speaker's identity is as claimed. The system is first trained on fixed passphrases or phonemes that can be combined. To authenticate, either the speaker says the passphrase or repeats a word (or set of words) composed of the learned phonemes. Verbal information verification deals with the contents of utterances. The system asks a set of questions such as "What is your mother's maiden name?" and "In which city were you born?" It then checks that the answers spoken are the same as the answers recorded in its database. The key difference is that speaker verification techniques are speaker-dependent, but verbal information verification techniques are speaker-independent, relying only on the content of the answers [1172].

Voice recognition systems are particularly vulnerable to replay attacks in which an adversary records, and later replays, the authorized user's voice. One detection method, designed for mobile phones, uses the difference in time that a voice reaches two microphones in the phone. The user says a passphrase that contains phonemes that produce known differences in time based on the placement of the phone and the user's voice. Experiments show the differences are not the same when the speech is replayed [2091].

13.7.3 Eyes

Authentication by eye characteristics uses the iris and the retina. Patterns within the iris are unique for each person. So one verification approach is to compare the patterns statistically and ask whether the differences are random [503, 504, 2005]. Retinal scans rely on the uniqueness of the patterns made by blood vessels at the back of the eye. This requires a laser beaming onto the retina, which is highly intrusive [1251]. This method is typically used only in the most secure facilities [1143].

The availability of eye tracking devices has led to the study of eye motion as an authentication mechanism. The device tracks specific features of the eyes, such as statistics about the pupil size, the speed of eye motion and the length of time of lack of motion, and the steadiness of the gaze. The more features used in the analysis, the more accurate the identification, and using all features enabled attackers to be detected quickly (over 90% in 40 seconds), with few false negatives [612]. A variant of eye motion uses eye gestures, in which the user moves her eyes in a particular way and that motion is compared to a predetermined shape [518].

13.7.4 Faces

Face recognition consists of several steps. First, the face is located. If the user places her face in a predetermined position (for example, by resting her chin on a support), the problem becomes somewhat easier. However, facial features such as hair and glasses may make the recognition harder. Techniques for doing this include the use of neural networks [1178, 1614] and templates [2074]. The resulting image is then compared with the relevant image in the database. The correlation is affected by the differences in the lighting between the current image and the reference image, by distortion, by "noise," and by the view of the face. The correlation mechanism must be "trained." An alternative approach is to focus on the facial features such as the distance between the nose and the chin, and the angle of the line drawn from one to the other [1645].

Techniques have been developed to detect spoofing attacks on facial recognition systems, called "presentation attacks" [1557]. An interesting problem is that many of the data sets used to train facial recognition systems are biased, resulting in higher error rates for those whom the data set is biased against [315, 768].

13.7.5 Keystrokes

Keystroke dynamics requires a signature based on keystroke intervals, keystroke pressure, keystroke duration, and where the key is struck (on the edge or in the middle). This signature is believed to be unique in the same way that written signatures are unique [982, 1502]. Keystroke recognition can be both static and dynamic. Static recognition is done once, at authentication time, and usually involves typing a fixed, known string [67, 1373]. Once authentication has been completed, an attacker can capture the connection (or take over the terminal) without detection. Dynamic recognition is done throughout the session, so the aforementioned attack is not feasible. However, the signature must be chosen so that variations within an individual's session do not cause the authentication to fail. For example, keystroke intervals may vary widely, and the dynamic recognition mechanism must take this into account. The statistics gathered from a user's typing are then run through statistical tests (which may discard some data as invalid, depending on the technique used) that account for acceptable variance in the data [835].

13.7.6 Combinations

Several researchers have combined some of the techniques described above to improve the accuracy of biometric authentication. Lumini and Nanni [1217] provide an overview of techniques used to do this. Dieckmann, Plankensteiner, and Wagner [562] combined voice sounds and lip motion with the facial image. Duc et al. [592] describe a "supervisor module" for melding voice and face recognition with a success rate of 99.5%. Lu et al. [1209] combined mouse and eye movement. The results of experiments involving fusions of biometric characteristics indicate that a higher degree of accuracy can be attained than when only a single characteristic is used.

13.8 Location

Denning and MacDoran [544] suggest an innovative approach to authentication. They reason that if a user claims to be Anna, who is at that moment working in a bank in California but is also logging in from Russia at the same time, the user is impersonating Anna. Their scheme is based on the Global Positioning System (GPS), which can pinpoint a location to within a few meters. The physical location of an entity is described by a location signature derived from the GPS satellites. Each location (to within a few meters) and time (to within a few milliseconds) is unique, and hence form a location signature. This signature is transmitted to authenticate the user. The host also has a location signature sensor (LSS) and obtains a similar signature for the user. If the signatures disagree, the authentication fails.

If the LSS is stolen, the thief would have to log in from an authorized geographic location. Because the signature is generated from GPS data, which changes with respect to time, location, and a variety of vagaries resulting from the nature of the electromagnetic waves used to establish position, any such signature would be unique and could not be forged. Moreover, if intercepted, it could not be replayed except within the window of temporal uniqueness.

This technique can also restrict the locations from which an authorized user can access the system.

EXAMPLE: Suppose Anna is an employee of a bank in California. The bank uses location-based authentication to verify logins. Anna's LSS is stolen, and the thief takes it to New York. From there, the thief tries to access the bank's computer.

Anna's LSS generates a signature and transmits it to the bank. The bank's LSS determines that Anna's LSS is in New York and is supplying a correct signature. However, Anna is not authorized to access the bank's computer from New York, so the authentication is rejected. If the thief tries to forge a message indicating that Anna is connecting from inside California, the host's LSS would report that Anna was at a different location and would reject the connection.

An interesting point is that the authentication can be done continuously. The LSS simply intermingles signature data with the transmitted data, and the host checks it. If the connection were hijacked, the data from the LSS would be lost.

A mobile phone or other mobile computer may be used as an LSS [2089].

13.9 Multifactor Authentication

Authentication methods can be combined, or multiple methods can be used. Multifactor authentication uses two different forms of authentication to validate identity.

EXAMPLE: A mechanism that asks first for a password and then requires the user to enter a sequence of numbers sent to a smartphone is multifactor as it uses what the entity knows (the password) and what the entity has (the smartphone). A mechanism that asks for a password and then the answer to a question is not multifactor, as it uses only what the entity knows.

The widespread use of cell phones and other portable computing media, coupled with the growth of attacks on authentication systems, has encouraged the use of two-factor authentication. For example, many banks, particularly in Europe and Asia, use multifactor authentication [1052]. More commonly, many social networking web providers are encouraging users to adopt it.

EXAMPLE: Google provides a two-factor authentication protocol (called "2-Step Verification") [825,2162]. After a user supplies a login name and password, Google sends a six-digit code to a prearranged phone number or Google's mobile app. The user retrieves this code and enters it on the web page. If the number matches what was sent, authentication succeeds.

This method requires that the user have two factors. If the phone number is used, the user must have immediate access to the phone. This is usually a cell phone, which most people carry with them; but it can be any phone. Google can send the code by voice or text. When sent to the app, the user must have the device with the app.

Techniques using multiple methods assign one or more authentication methods to each entity. The entity must authenticate using the specific method,