Chapter 19 Zero-Knowledge Techniques

19.1 The Basic Setup

A few years ago, it was reported that some thieves set up a fake automatic teller machine at a shopping mall. When a person inserted a bank card and typed in an identification number, the machine recorded the information but responded with the message that it could not accept the card. The thieves then made counterfeit bank cards and went to legitimate teller machines and withdrew cash, using the identification numbers they had obtained.

How can this be avoided? There are several situations where someone reveals a secret identification number or password in order to complete a transaction. Anyone who obtains this secret number, plus some (almost public) identification information (for example, the information on a bank card), can masquerade as this person. What is needed is a way to use the secret number without giving any information that can be reused by an eavesdropper. This is where zero-knowledge techniques come in.

The basic challenge-response protocol is best illustrated by an example due to Quisquater, Guillou, and Berson [Quisquater et al.]. Suppose there is a tunnel with a door, as in Figure 19.1. Peggy (the prover) wants to prove to Victor (the verifier) that she can go through the door without giving any information to Victor about how she does it. She doesn't even want to let Victor know which direction she can pass through the door (otherwise, she

could simply walk down one side and emerge from the other). They proceed as follows. Peggy enters the tunnel and goes down either the left side or the right side of the tunnel. Victor waits outside for a minute, then comes in and stands at point B. He calls out “Left” or “Right” to Peggy. Peggy then comes to point B by the left or right tunnel, as requested. This entire protocol is repeated several times, until Victor is satisfied. In each round, Peggy chooses which side she will go down, and Victor randomly chooses which side he will request.

Figure 19.1 The Tunnel Used in the Zero-Knowledge Protocol

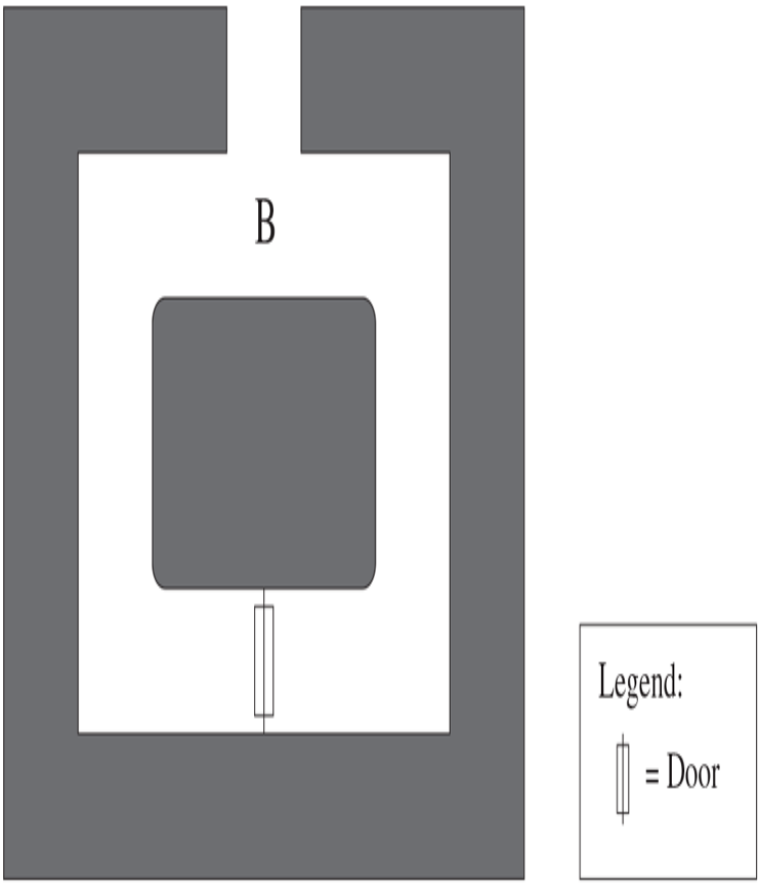

Figure 19.1 Full Alternative Text

Since Peggy must choose to go down the left or right side before she knows what Victor will say, she has only a 50% chance of fooling Victor if she doesn't know how to go through the door. Therefore, if Peggy comes out the correct side for each of 10 repetitions, there is only one chance in $2^{10} = 1024$ that Peggy doesn't know how to go through the door. At this point, Victor is probably convinced, though he could try a few more times just to be sure.

Suppose Eve is watching the proceedings on a video monitor carried by Victor. She will not be able to use anything she sees to convince Victor or anyone else that she, too, can go through the door. Moreover, she might not even be convinced that Peggy can go through the door. After all, Peggy and Victor could have planned the sequence of rights and lefts ahead of time. By this reasoning, there is no useful information that Victor obtains that can be transmitted to anyone.

Note that there is never a proof, in a strict mathematical sense, that Peggy can go through the door. But there is overwhelming evidence, obtained through a series of challenges and responses. This is a feature of zero-knowledge "proofs."

There are several mathematical versions of this procedure, but we'll concentrate on one of them. Let $n = pq$ be the product of two large primes. Let y be a square mod n with $\gcd(y,\ n) = 1$. Recall that finding square roots mod n is hard; in fact, finding square roots mod n is equivalent to factoring n (see Section 3.9). However, Peggy claims to know a square root s of y. Victor wants to verify this, but Peggy does not want to reveal s. Here is the method:

1. Peggy chooses a random number r_1 and lets $r_2 \equiv s r_1^{-1} \pmod{n}$, so

$$r_1 r_2 \equiv s \pmod{n}.$$

(We may assume that $\gcd(r_1, n) = 1$, so $r_1^{-1} \pmod n$ exists; otherwise, Peggy has factored n.) She computes

$$x_1 \equiv r_1^2, \quad x_2 \equiv r_2^2 \pmod n$$

and sends x_1 and x_2 to Victor.

2. Victor checks that $x_1 x_2 \equiv y \pmod n$, then chooses either x_1 or x_2 and asks Peggy to supply a square root of it. He checks that it is an actual square root.

3. The first two steps are repeated several times, until Victor is convinced.

Of course, if Peggy knows s, the procedure proceeds without problems. But what if Peggy doesn't know a square root of y? She can still send Victor two numbers x_1 and x_2 with $x_1 x_2 \equiv y$. If she knows a square root of x_1 and a square root of x_2, then she knows a square root of $y \equiv x_1 x_2$. Therefore, for at least one of them, she does not know a square root. At least half the time, Victor is going to ask her for a square root she doesn't know. Since computing square roots is hard, she is not able to produce the desired answer, and therefore Victor finds out that she doesn't know s.

Suppose, however, that Peggy predicts correctly that Victor will ask for a square root of x_2. Then she chooses a random r_2, computes $x_2 \equiv r_2^2 \pmod n$, and lets $x_1 \equiv y x_2^{-1} \pmod n$. She sends x_1 and x_2 to Victor, and everything works. This method gives Peggy a 50% chance of fooling Victor on any given round, but it requires her to guess which number Victor will request each time. As soon as she fails, Victor will find out that she doesn't know s.

If Victor verifies that Peggy knows a square root, does he obtain any information that can be used by someone else? No, since in any step he is only learning the square root of a random square, not a square root of y. Of course, if Peggy uses the same random numbers more than once, he could find out the square roots of both x_1

and x_2 and hence a square root of y. So Peggy should be careful in her choice of random numbers.

Suppose Eve is listening. She also will only learn square roots of random numbers. If she tries to use the same sequence of random numbers to masquerade as Peggy, she needs to be asked for the square roots of exactly the same sequence of x_1's and x_2's. If Victor asks for a square root of an x_1 in place of an x_2 at one step, for example, Eve will not be able to supply it.

19.2 The Feige-Fiat-Shamir Identification Scheme

The preceding protocol requires several communications between Peggy and Victor. The Feige-Fiat-Shamir method reduces this number and uses a type of parallel verification. This then is used as the basis of an identification scheme.

Again, let $n = pq$ be the product of two large primes. Peggy has secret numbers $s_1, \ldots, s_k$. Let $v_i \equiv s_i^{-2} \pmod{n}$ (we assume $\gcd(s_i, n) = 1$). The numbers v_i are sent to Victor. Victor will try to verify that Peggy knows the numbers $s_1, \ldots, s_k$. Peggy and Victor proceed as follows:

1. Peggy chooses a random integer r, computes $x \equiv r^2 \pmod{n}$ and sends x to Victor.
2. Victor chooses numbers $b_1, \ldots, b_k$ with each $b_i \in \{0, 1\}$. He sends these to Peggy.
3. Peggy computes $y \equiv r s_1^{b_1} s_2^{b_2} \cdots s_k^{b_k} \pmod{n}$ and sends y to Victor.
4. Victor checks that $x \equiv y^2 v_1^{b_1} v_2^{b_2} \cdots v_k^{b_k} \pmod{n}$.
5. Steps 1 through 4 are repeated several times (each time with a different r).

Consider the case $k = 1$. Then Peggy is asked for either r or rs_1. These are two random numbers whose quotient is a square root of v_1. Therefore, this is essentially the same idea as the simplified scheme discussed previously, with quotients instead of products.

Now let's analyze the case of larger k. Suppose, for example, that Victor sends $b_1 = 1$, $b_2 = 1$, $b_4 = 1$, and all other $b_i = 0$. Then Peggy must produce $y \equiv r s_1 s_2 s_4$

, which is a square root of $xv_1^{-1}v_2^{-1}v_4^{-1}$. In fact, in each round, Victor is asking for a square root of a number of the form $xv_{i_1}^{-1}v_{i_2}^{-1}\cdots v_{i_j}^{-1}$. Peggy can supply a square root if she knows r, s_{i_1}, $\ldots$, s_{i_j}. If she doesn't, she will have a hard time computing a square root.

If Peggy doesn't know any of the numbers $s_1, \ldots, s_k$ (the likely scenario also if someone other than Peggy is pretending to be Peggy), she could guess the string of bits that Victor will send. Suppose she guesses correctly, before she sends x. She lets y be a random number and declares $x \equiv y^2 v_1^{b_1} v_2^{b_2} \cdots v_k^{b_k} \pmod{n}$. When Victor sends the string of bits, Peggy sends back the value of y. Of course, the verification congruence is satisfied. But if Peggy guesses incorrectly, she will need to modify her choice of y, which means she will need some square roots of v_i's.

For example, suppose Peggy is able to supply the correct response when $b_1 = 1$, $b_2 = 1$, $b_4 = 1$, and all other $b_i = 0$. This could be accomplished by guessing the bits and using the preceding method of choosing x. However, suppose Victor sends $b_1 = 1$, $b_3 = 1$, and all other $b_i = 0$. Then Peggy will be ready to supply a square root of $xv_1^{-1}v_2^{-1}v_4^{-1}$ but will be asked to supply a square root of $xv_1^{-1}v_3^{-1}$. This, combined with what she knows, is equivalent to knowing a square root of $v_2^{-1}v_3v_4^{-1}$, which she is not able to compute. In an extreme case, Victor could send all bits equal to 0, which means Peggy must supply a square root of x. With Peggy's guess as before, this means she would know a square root of $v_1v_2v_4$. In summary, if Peggy's guess is not correct, she will need to know the square root of a nonempty product of v_i's, which she cannot compute. Therefore, there are 2^k possible strings of bits that Victor can send, and only one will allow Peggy to fool Victor. In one iteration of the protocol, the chances are only one in 2^k that Victor will be fooled. If the procedure is repeated t times, the

chances are 1 in 2^{kt} that Victor is fooled. Recommended values are $k = 5$ and $t = 4$. Note that this gives the same probability as 20 iterations of the earlier scheme, so the present procedure is more efficient in terms of communication between Peggy and Victor. Of course, Victor has not obtained as strong a verification that Peggy knows, for example, s_1, but he is very certain that Eve is not masquerading as Peggy, since Eve should not know any of the s_i's.

There is an interesting feature of how the numbers are arranged in Steps 1 and 4. It is possible for Peggy to use a cryptographic hash function and send the hash of x after computing it in Step 1. After Victor computes $y^2 v_1^{b_1} v_2^{b_2} \cdots v_k^{b_k} \pmod{n}$ in Step 4, he can compute the hash of this number and compare with the hash sent by Peggy. The hash function is assumed to be collision resistant, so Victor can be confident that the the congruence in Step 4 is satisfied. Since the output of the hash function is probably shorter than x (for example, 256 bits for the hash, compared to 2048 bits for x), this saves a few bits of transmission.

The preceding can be used to design an identification scheme. Let I be a string that includes Peggy's name, birth date, and any other information deemed appropriate. Let H be a public hash function. A trusted authority Arthur (the bank, a passport agency, ...) chooses $n = pq$ to be the product of two large primes. Arthur computes $H(I \parallel j)$ for some small values of j, where $I \parallel j$ means j is appended to I. Using his knowledge of p, q, he can determine which of these numbers $H(I \parallel j)$ have square roots mod n and calculate a square root for each such number. This yields numbers $v_1 = H(I \parallel j_1),\ \ldots,\ v_k = H(I \parallel j_k)$ and square roots $s_1,\ \ldots,\ s_k$. The numbers I, n, $j_1,\ \ldots,\ j_k$ are made public. Arthur gives the numbers $s_1,\ \ldots,\ s_k$ to Peggy, who keeps them secret. The prime numbers p, q are discarded once the square

roots are calculated. Likewise, Arthur does not need to store $s_1, \ldots, s_k$ once they are given to Peggy. These two facts add to the security, since someone who breaks into Arthur's computer cannot compromise Peggy's security. Moreover, a different n can be used for each person, so it is hard to compromise the security of more than one individual at a time.

Note that since approximately half the numbers mod p and half the numbers mod q have square roots, the Chinese remainder theorem implies that approximately 1/4 of the numbers mod n have square roots. Therefore, each $H(I \parallel j)$ has a 1/4 probability of having a square root mod n. This means that Arthur should be able to produce the necessary numbers $j_1, \ldots, j_k$ quickly.

Peggy goes to an automatic teller machine, for example. The machine reads I from Peggy's card. It downloads $n, j_1, \ldots, j_k$ from a database and calculates $v_i = H(I \parallel j_i)$ for $1 \leq i \leq k$. It then performs the preceding procedure to verify that Peggy knows $s_1, \ldots, s_k$. After a few iterations, the machine is convinced that the person is Peggy and allows her to withdraw cash. A naive implementation would require a lot of typing on Peggy's part, but at least Eve won't get Peggy's secret numbers. A better implementation would use chips embedded in the card and store some information in such a way that it cannot be extracted.

If Eve obtains the communications used in the transaction, she cannot determine Peggy's secret numbers. In fact, because of the zero-knowledge nature of the protocol, Eve obtains no information on the secret numbers $s_1, \ldots, s_k$ that can be reused in future transactions.

19.3 Exercises

1. Consider the diagram of tunnels in Figure 19.2. Suppose each of the four doors to the central chamber is locked so that a key is needed to enter, but no key is needed to exit. Peggy claims she has the key to one of the doors. Devise a zero-knowledge protocol in which Peggy proves to Victor that she can enter the central chamber. Victor should obtain no knowledge of which door Peggy can unlock.

Figure 19.2 Diagram for Exercise 1

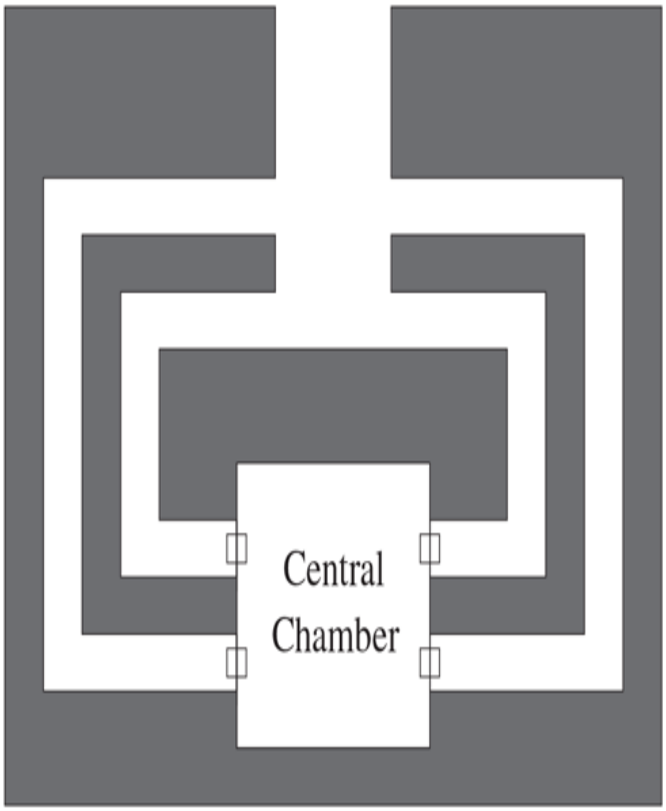

Figure 19.2 Full Alternative Text

2. Suppose p is a large prime, α is a primitive root, and $\beta \equiv \alpha^a \pmod{p}$. The numbers p, α, β are public. Peggy wants

to prove to Victor that she knows a without revealing it. They do the following:

1. Peggy chooses a random number $r \pmod{p-1}$.
2. Peggy computes $h_1 \equiv \alpha^r \pmod{p}$ and $h_2 \equiv \alpha^{a-r} \pmod{p}$ and sends h_1, h_2 to Victor.
3. Victor chooses $i = 1$ or $i = 2$ and asks Peggy to send either $r_1 = r$ or $r_2 = a - r \pmod{p-1}$.
4. Victor checks that $h_1 h_2 \equiv \beta \pmod{p}$ and that $h_i \equiv \alpha^{r_i} \pmod{p}$.

They repeat this procedure t times, for some specified t.

1. Suppose Peggy does not know a. Why will she usually be unable to produce numbers that convince Victor?
2. If Peggy does not know a, what is the probability that Peggy can convince Victor that she knows a?
3. Suppose naive Nelson tries a variant. He wants to convince Victor that he knows a, so he chooses a random r as before, but does not send h_1, h_2. Victor asks for r_i and Nelson sends it. They do this several times. Why is Victor not convinced of anything? What is the essential difference between Nelson's scheme and Peggy's scheme that causes this?

3. Naive Nelson thinks he understands zero-knowledge protocols. He wants to prove to Victor that he knows the factorization of n (which equals pq for two large primes p and q) without revealing this factorization to Victor or anyone else. Nelson devises the following procedure: Victor chooses a random integer x mod n, computes $y \equiv x^2 \pmod{n}$, and sends y to Nelson. Nelson computes a square root s of y mod n and sends s to Victor. Victor checks that $s^2 \equiv y \pmod{n}$. Victor repeats this 20 times.

    1. Describe how Nelson computes s. You may assume that p and q are $\equiv 3 \pmod{4}$ (see Section 3.9).
    2. Explain how Victor can use this procedure to have a high probability of finding the factorization of n. (Therefore, this is not a zero-knowledge protocol.)
    3. Suppose Eve is eavesdropping and hears the values of each y and s. Is it likely that Eve obtains any useful information? (Assume no value of y repeats.)

4. Exercise 2 gave a zero-knowledge proof that Peggy knows a discrete logarithm. Here is another method. Suppose p is a large prime, α is a primitive root, and $\beta \equiv \alpha^a \pmod{p}$. The numbers

p, α, β are public. Peggy wants to prove to Victor that she knows a without revealing it. They do the following:

1. Peggy chooses a random integer k with $1 \leq k < p-1$, computes $\gamma \equiv \alpha^k \pmod{p}$, and sends γ to Victor.
2. Victor chooses a random integer r with $1 \leq r < p-1$ and sends r to Peggy.
3. Peggy computes $y \equiv k - ar \pmod{p-1}$ and sends y to Victor.
4. Victor checks whether $\gamma \equiv \alpha^y \beta^r \pmod{p}$. If so, he believes that Peggy knows a.

1. Show that the verification equation holds if the procedure is followed correctly.
2. Does Victor obtain any information that will allow him to compute a?
3. Suppose Eve finds out the values of γ, r, and y. Will she be able to determine a?
4. Suppose Peggy repeats the procedure with the same value of k, but Victor uses different values r_1 and r_2. How can Eve, who has listened to all communications between Victor and Peggy, determine a?

The preceding procedure is the basis for the **Schnorr identification scheme**. Victor could be a bank and a could be Peggy's personal identification number. The bank stores β, and Peggy must prove she knows a to access her account. Alternatively, Victor could be a central computer and Peggy could be logging on to the computer through nonsecure telephone lines. Peggy's password is a, and the central computer stores β.

In the Schnorr scheme, p is usually chosen so that $p-1$ has a large prime factor q, and α, instead of being a primitive root, is taken to satisfy $\alpha^q \equiv 1 \pmod{p}$. The congruence defining y is then taken mod q. Moreover, r is taken to satisfy $1 \leq r \leq 2^t$ for some t, for example, $t = 40$.

5. Peggy claims that she knows an RSA plaintext. That is, n, e, c are public and Peggy claims that she knows m such that $m^e \equiv c \pmod{n}$. She wants to prove this to Victor using a zero-knowledge protocol. Peggy and Victor perform the following steps:

    1. Peggy chooses a random integer r_1 and computes $r_2 \equiv m \cdot r_1^{-1} \pmod{n}$ (assume that $\gcd(r_1, n) = 1$.)

2. Peggy computes $x_1 \equiv r_1^e \pmod{n}$ and $x_2 \equiv r_2^e \pmod{n}$ and sends $x_1,\ x_2$ to Victor.

3. Victor checks that $x_1 x_2 \equiv c \pmod{n}$.

Give the remaining steps of the protocol. Victor should be at least 99% convinced that Peggy is not lying.

6. 1. Suppose that p is a large prime, and $g,\ h \not\equiv 0 \pmod{p}$. Peggy wants to prove to Victor, using a zero-knowledge protocol, that she knows a value of x with $g^x \equiv h \pmod{p}$. Peggy and Victor do the following:

    1. Peggy chooses three random integers r_1, r_2, r_3 with $r_1 + r_2 + r_3 \equiv x \pmod{p-1}$.

    2. Peggy computes $h_i \equiv g^{r_i}$, for $i = 1, 2, 3$ and sends h_1, h_2, h_3 to Victor.

    3. Victor checks that $h_1 h_2 h_3 \equiv h \pmod{n}$.

    Design the remaining steps of this protocol so that Victor is at least 99% convinced that Peggy is not lying. (Note: There are two ways for Victor to proceed in Step 4. One has a higher probability of catching Peggy, if she is cheating, than the other.)

    2. Give a reasonable method for Peggy to choose the three random numbers such that $r_1 + r_2 + r_3 = \equiv x \pmod{p-1}$. (A method that doesn't work is "Choose three random numbers and see if their sum is x. If not, try again.)

7. Suppose that n is the product of two large primes, and that s is given. Peggy wants to prove to Victor, using a zero-knowledge protocol, that she knows a value of x with $x^2 \equiv s \pmod{n}$. Peggy and Victor do the following:

    1. Peggy chooses three random integers r_1, r_2, r_3 with $r_1 r_2 r_3 \equiv x \pmod{n}$.

    2. Peggy computes $x_i \equiv r_i^2$, for $i = 1, 2, 3$ and sends x_1, x_2, x_3 to Victor.

    3. Victor checks that $x_1 x_2 x_3 \equiv s \pmod{n}$.

    1. Design the remaining steps of this protocol so that Victor is at least 99% convinced that Peggy is not lying. (*Note:* There are two ways for Victor to proceed in Step 4. One

has a higher probability of catching Peggy, if she is cheating, than the other.)

2. Give a reasonable method for Peggy to choose the three random numbers such that $r_1 r_2 r_3 = \equiv x \pmod{n}$. (A method that doesn't work is "Choose three random numbers and see if their product is x. If not, try again.")

8. Peggy claims that she knows an RSA plaintext. That is, n, e, c are public and Peggy claims that she knows m such that $m^e \equiv c \pmod{n}$. Devise a zero-knowledge protocol similar to that used in Exercises 6 and 7 for Peggy to convince Victor that she knows m.