2

Identification Schemes and Entity Authentication

2.1 Introduction

Suppose you want to prove your identity to someone else. It is sometimes said that this can be done in one of three ways, namely, based on what you are, what you have or what you know. "What you are" refers to physical attributes; "what you have" refers to documents, or credentials; and "what you know" encompasses passwords, personal information, etc. These techniques are described in more detail now.

physical attributes
People often identify other people already known to them by their appearance. This could include family and friends as well as famous celebrities. Specific features used for purposes include sex, height, weight, racial origin, eye color, hair color, etc. Attributes that are unique to an individual are often more useful; these include fingerprints or retina scans.

credentials
A "credential" is defined, in the diplomatic usage of the word, as a letter of introduction. Trusted documents or cards such as driver's licences and passports function as credentials in many situations. Note that credentials often include pictures, which enables physical identification, as described above.

knowledge
Knowledge is often used for identification when the person being identified is not in the same physical location as the the person or entity performing the identification. In the context of identification, knowledge could be a password or PIN (personal identification number), or "your mother's maiden name" (a favorite of credit card companies). The difficulty with using knowledge for identification is that such knowledge may not be secret

in the first place, and, moreover, it is usually revealed as part of the identification process. This allows for possible future impersonation of the person being identified, which is not a good thing!

Let's consider some common, everyday situations where it is necessary to electronically "prove" one's identity. Some typical scenarios are as follows:

telephone "calling cards"
To charge long-distance telephone calls (using a calling card), one requires only the knowledge of the telephone number that is being billed for the call, together with a four-digit PIN.

remote login
To do a remote login to a computer over a network, it suffices to know a valid user name and the corresponding password.

credit card purchases
When a purchase is made at a store using a credit card, the sales clerk is supposed to verify that the customer's signature matches the signature on the back of the card. This provides a weak form of identification, as many signatures can be forged without too much difficulty. Some credit cards also have a photograph of the owner, which provides an additional level of authentication. However, there are also situations where a credit card can be used without even these nominal levels of identification, for example, to purchase gasoline at the gas pump, without interacting with an attendant.

credit card purchases without the credit card
In many situations, possession of the actual credit card is not required in order to use it. For example, to charge purchases made over the telephone or the internet to a credit card, all that usually is necessary is a valid credit card number (and perhaps the expiry date). This clearly does not provide any real security, because there is no meaningful authentication of identity being performed. Indeed, it is a simple matter for a dishonest person to collect valid credit card numbers and then use them in situations where the actual card is not required to carry out the transaction.

bank machine withdrawals
To withdraw money from an automated teller machine (or ATM), we use a card together with a four-digit personal identification number (PIN). The card contains the owner's name and information about his or her bank accounts. The purpose of the PIN is to protect against fraudulent use of the card by someone else. The assumption is that the only person who knows the correct PIN is the owner of the card.

In practice, these types of schemes are not usually implemented in a secure way. In the protocols performed over the telephone, any eavesdropper can use the identifying information for their own purposes. This could include the person who is the recipient of the information; many credit card "scams" operate in this way. An ATM card is somewhat more secure, but there are still weaknesses. For example, someone monitoring the communication line can obtain all the information encoded on the card's magnetic strip, as well as the PIN. This could perhaps allow an imposter to gain access to a bank account. Finally, remote computer login is a serious problem due to the fact that user IDs and passwords are transmitted over the network in unencrypted form. Thus they are vulnerable to anyone who is eavesdropping on the computer network.

The objective of an identification scheme would be that someone "listening in" as Alice identifies herself to Bob, say, should not subsequently be able to misrepresent herself as Alice. The attack model is that the adversary can observe all the information being transmitted between Alice and Bob, and the adversarial goal is to be able to impersonate Alice. Furthermore, we may even try to guard against the possibility that Bob himself might try to impersonate Alice after she has identified herself to him. Ultimately, we would like to devise "zero-knowledge" schemes whereby Alice can prove her identity electronically, without "giving away" the knowledge that is used as her identifying information.

Several such identification schemes have been discovered. One practical objective is to find a scheme that is simple enough that it can be implemented on a smart card, which is essentially a credit card equipped with a chip that can perform arithmetic computations. Hence, both the amount of computation and the memory requirements should be kept as small as possible. Such a card would be a more secure alternative to current ATM cards. However, it is important to note that the "extra" security pertains to someone monitoring the communication line. Since it is the card that is "proving" its identity, we have no extra protection against a lost card. It would still be necessary to include a PIN in order to establish that it is the real owner of the card who is initiating the identification protocol.

An initial observation is that any identification protocol should involve randomization in some way. If the information that Alice transmits to Bob to identify herself never changes, then the scheme is insecure in the model we introduced above. Therefore most "secure" identification schemes include "random challenges" in them. This concept is explored more deeply in the next section.

2.2 Challenge-and Response in the Secret-key Setting

In later sections, we will describe some of the more popular zero-knowledge identification schemes. First, we look at identification in the secret-key setting, where Alice and Bob both have the same secret key. We begin by examining a very simple (but insecure) scheme that can be based on any message authentication code, e.g., the *Advanced Encryption Standard* in CBC mode. The protocol, which is described as Protocol 2.1, is called *challenge-and-response*. In it, we assume that Alice is identifying herself to Bob, and their common secret key is denoted by K.

(Bob can also identify himself to Alice, by interchanging the roles of Alice and Bob in the protocol.) As usual, the message authentication code MAC_K is used to compute authentication tags.

Protocol 2.1: INSECURE CHALLENGE-AND-RESPONSE

1. Bob chooses a random *challenge*, r, which he sends to Alice.
2. Alice computes
$$y = MAC_K(r)$$
and sends y to Bob.
3. Bob computes
$$y' = MAC_K(r).$$
If $y' = y$, then Bob outputs "accept"; otherwise, Bob outputs "reject".

We will often depict interactive protocols in diagrammatic fashion, as follows:

$$\begin{array}{ccc} & \xleftarrow{\hspace{7em} r \hspace{7em}} & \\ \text{Alice} & & \text{Bob} \\ & \xrightarrow{\hspace{4em} y = MAC_K(r) \hspace{4em}} & \end{array}$$

Before analyzing the weakness of this protocol, let us define some basic terminology related to interactive protocols. In general, an *interactive protocol* will comprise two or more parties that are communicating with each other. Each party is an algorithm that alternately sends and receives information. Each run of a protocol will be called a *session*. Each step within a session of the protocol is called a *flow*; each flow consists of information transmitted from one party to another party. (Protocol 2.1 consists of two flows, the first one being from Bob to Alice, and the second one being from Alice to Bob.)

The above scheme is insecure, even if the message authentication code is secure. It is susceptible to a *parallel session attack*, wherein Oscar impersonates Alice. The attack works as follows:

$$\begin{array}{ccc} & \xleftarrow{\hspace{7em} r \hspace{7em}} & \\ \text{Oscar} & \boxed{\begin{array}{c} \xrightarrow{\hspace{6em} r \hspace{6em}} \\ \xleftarrow{\hspace{3em} y = MAC_K(r) \hspace{3em}} \end{array}} & \text{Bob} \\ & \xrightarrow{\hspace{7em} y \hspace{7em}} & \end{array}$$

Within the first session (in which it is supposed that Alice is identifying herself to Bob), Oscar initiates a second session in which he asks Bob to identify himself. This second session is boxed in the above diagram. In this second session, Oscar gives Bob the same challenge that he received from Bob in the first session. Once he receives Bob's response, Oscar resumes the first session, in which he relays Bob's response back to him. Thus Oscar is able to successfully complete the first session!

The reader might object to the premise that parallel sessions constitute a realistic threat. However, there are scenarios in which parallel sessions might be reasonable and/or desirable, and it would seem to be prudent to design an identification protocol to withstand such attacks. Here is one easy way to rectify the problem:

Protocol 2.2: (SECURE) CHALLENGE-AND-RESPONSE

1. Bob chooses a random challenge, r, which he sends to Alice.
2. Alice computes
$$y = MAC_K(Alice \parallel r)$$
and sends y to Bob.
3. Bob computes
$$y' = MAC_K(Alice \parallel r).$$
If $y' = y$, then Bob outputs "accept"; otherwise, Bob outputs "reject".

In Protocol 2.2, we will assume that the random challenge is a bitstring of a specifed, predetermined length. We also assume that the identity string ($Alice$ or Bob, depending on whose identity is being authenticated) is also a bitstring of a specified length, formatted in some standard, fixed manner.

We argue that a parallel session attack cannot be carried out against Protocol 2.2. If Oscar attampted to mount the same attack as before, he would receive the value $MAC_K(Bob \parallel r)$ from Bob in the second session. This is of no help in computing the value $MAC_K(Alice \parallel r)$ required in the first session to successfully respond to Bob's challenge.

The preceding discussion may convince the reader that the parallel session attack cannot be mounted against Protocol 2.2, but it does not present a proof of security against all possible attacks. We will give an informal proof of security, but first we have to explicitly list all the assumptions we make:

1. The secret key, K, is known only to Alice and Bob.
2. Alice and Bob both have perfect random number generators which they use to determine their challenges. Therefore, we assume that the probability that the same challenge occurs in two different sessions is very small.
3. The message authentication code is secure. In other words, the probability that Oscar can correctly compute $MAC_K(x)$ is very small, even when he

is given many other MACs, say $MAC_K(x_i)$, $i = 1, 2, \ldots$, provided that $x \neq x_i$ for any i.

Oscar may observe several sessions between Alice and Bob. Oscar's goal is to deceive Bob, i.e., to get Bob to output "accept" in a session in which Alice is not taking part, or in which Oscar modifies messages in the channel. (If Oscar does not change any messages in the channel, then he is not an active adversary, nor is this regarded as an attack.) We show that Oscar will not succeed in deceiving Bob in this way, except with small probability, when the above assumptions are valid. This is done fairly easily by analyzing the structure of the protocol.

Suppose that Bob outputs "accept". Then $y = MAC_K(Alice \parallel r)$, where y is the value he receives in the second flow and r was his challenge from the first flow of the protocol. We claim that, with high probability, this value y must have been constructed by Alice (i.e., it cannot have been constructed by Oscar) in response to the challenge r from the first flow of the protocol. This claim can be verified by considering all possible cases that could arise, and eliminating them in turn:

1. Suppose the value $y = MAC_K(Alice \parallel r)$ was previously constructed by Bob himself in some previous session. However, Bob only computes MACs of the form $MAC_K(Bob \parallel r)$, so he would not have created y himself.
2. Suppose the value y was previously constructed by Alice in some earlier session. The challenge r is assumed to be a challenge newly created by Bob, so Bob (probably) would not have issued the same challenge in some other session.
3. Suppose the value y is a new MAC that is constructed by Oscar. Assuming that the message authentication code is secure, Oscar cannot do this, except with a very small probability.

The informal proof given above can easily be made more precise. If we can prove an explicit, precise statement of the security of the underlying MAC, then we can give a precise security guarantee for the identification protocol. On the other hand, if we make an assumption about the MAC's security, then we can provide a security result for the identification protocol that depends on this assumption (this is the usual model of provable security, as described in Chapter 1). The security guarantees for the identification protocol pertain to the probability that the adversary can fool Bob into accepting, when the adversary is an active participant in the protocol.

A MAC is said to be (q, ϵ)-*secure* if the adversary cannot construct a valid MAC for any new message with probability greater than ϵ, given that the adversary has previously seen valid MACs for at most q messages. (As usual, we assuming a fixed key, K, whose value is not known to the adversary, is used to construct all the MACs.) In fact, unconditionally secure MACs of this type exist for any desired values of q and ϵ. These unconditionally secure MACs typically require fairly large keys (especially if q is large). As a consequence, the MACs that cannot be proved to be secure are more often used in practice. In this situation, an assumption about the security of the MAC is necessary. This assumption would take a similar form, but include time as an explicit parameter: A MAC is said to be (q, ϵ, T)-*secure* if the adversary cannot construct a valid MAC for any new message with probability greater than ϵ, given that his computation time is at most T and given that he has previously seen valid MACs for at most q messages.

To simplify the argument, suppose we base the identification protocol on an unconditionally secure MAC, as described above. Then the resulting identification protocol will also be unconditionally secure, provided that the adversary has access to at most q valid MACs during some collection of sessions that all use the same MAC key. We need to quantify one additional parameter, namely, the size of the random challenge used in the protocol. We will suppose that the random challenge is a (positive) k-bit integer. Under these conditions, we can easily give an upper bound on the adversary's probability of deceiving Bob. We consider the same three cases as before:

1. As argued above, the value $y = MAC_K(Alice \parallel r)$ cannot have been previously constructed by Bob himself in some other session.
2. Suppose the value y was previously constructed by Alice in some other session. The challenge r is assumed to be a challenge newly created by Bob. The probability that Bob already used r in a specific previous session is $1/2^k$. There are at most q previous sessions under consideration, so the probability that r was used as challenge in one of these previous sessions is at most $q/2^k$. If this happens, then the adversary can re-use a MAC from a previous session.
3. Suppose the value y is a new MAC that is constructed by Oscar. Then, Oscar will be successful in his deception with probability at most ϵ; this follows from the security of the message authentication code being used.

Summing up, Oscar's probability of deceiving Bob is at most $q/2^k + \epsilon$. We therefore have established the security of the identification scheme as a function of the security of the underlying primitives.