Entrust Technologies

An Introduction to Cryptography

Author: Ian Curry
Date: December 1997
Version: 1.0

Cryptography

The concept of securing messages through cryptography has a long history. Indeed, Julius Caesar is credited with creating one of the earliest cryptographic systems to send military messages to his generals.

Throughout history, however, there has been one central problem limiting widespread use of cryptography. That problem is *key management*. In cryptographic systems, the term *key* refers to a numerical value used by an algorithm to alter information, making that information secure and visible only to individuals who have the corresponding key to recover the information. Consequently, the term key management refers to the secure administration of keys to provide them to users where and when they are required.

Historically, encryption systems used what is known as symmetric cryptography. Symmetric cryptography uses the same key for both encryption and decryption. Using symmetric cryptography, it is safe to send encrypted messages without fear of interception (because an interceptor is unlikely to be able to decipher the message); however, there always remains the difficult problem of how to securely transfer the key to the recipients of a message so that they can decrypt the message.

A major advance in cryptography occurred with the invention of public-key cryptography. The primary feature of public-key cryptography is that it removes the need to use the same key for encryption and decryption. With public-key cryptography, keys come in pairs of matched "public" and "private" keys. The public portion of the key pair can be distributed in a public manner without compromising the private portion, which must be kept secret by its owner. An operation (for example, encryption) done with the public key can only be undone with the corresponding private key.

Prior to the invention of public-key cryptography, it was essentially impossible to provide key management for large-scale networks. With symmetric cryptography, as the number of users increases on a network, the number of keys required to provide secure communications among those users increases rapidly. For example, a network of 100 users would require almost 5000 keys if it used only symmetric cryptography. Doubling such a network to 200 users increases the number of keys to almost 20,000. Thus, when only using symmetric cryptography, key management quickly becomes unwieldy even for relatively small-scale networks.

The invention of public-key cryptography was of central importance to the field of cryptography and provided answers to many key management problems for large-scale networks. For all its benefits, however, public-key cryptography did not provide a comprehensive solution to the key management problem. Indeed, the possibilities brought forth by public-key cryptography heightened the need for sophisticated key management systems to answer questions such as the following:

> "How can I easily encrypt a file once for a number of different people using public-key cryptography?"
>
> "If I lose my keys, how can I decrypt all of my files that were encrypted with those keys?"
>
> "How do I know that I really have Alice's public key and not the public key of someone pretending to be Alice?"
>
> "How can I know that a public key is still trustworthy?"

The Entrust® product family combines symmetric and public-key cryptography to provide answers to key management questions such as those listed above. For more information about the Entrust product family, refer to the *Entrust Overview* White Paper. For information regarding the requirements for public-key infrastructures, refer to the White Paper titled *Trusted Public-Key Infrastructures*. Both of these papers are available on the Entrust Technologies Web site at *http://www.entrust.com/library.htm*.

The next section provides an introduction to the mechanics of encryption and digital signature.

Encryption and digital signature explained

To better understand how cryptography is used to secure electronic communications, let's look at a process we are all familiar with: writing and sending a check.

Securing the electronic version

The simplest electronic version of the check can be a text file, created with a word processor, asking your bank to pay someone a specific sum. However, sending this check over an electronic network poses several security problems:

- since anyone could intercept and read the file, you need confidentiality.
- since someone else could create a similar counterfeit file, the bank needs to authenticate that it was actually you who created the file.
- since you could deny creating the file, the bank needs non-repudiation.
- since someone could alter the file, both you and the bank need data integrity.

To overcome these issues, Entrust performs a number of steps hidden behind a simple user interface. The first step is to "sign" the check with a digital signature.

Digital signature

The process of digitally signing starts by taking a mathematical summary (called a *hash code*) of the check. This hash code is a uniquely-identifying digital fingerprint of the check. If even a single bit of the check changes, the hash code will dramatically change. The next step in creating a digital signature is to sign the hash code with your private key. This signed hash code is then appended to the check.

How is this a signature? Well, the recipient of your check can verify the hash code sent by you, using your public key. At the same time, a new hash code can be created from the received check and compared with the original signed hash code. If the hash codes match, then the recipient has verified that the check has not been altered. The recipient also knows that only you could have sent the check because *only you have the private key that signed the original hash code.*

Confidentiality and encryption

Once the electronic check is digitally signed, it can be encrypted using a high-speed mathematical transformation with a key that will be used later

to decrypt the document. This is often referred to as a *symmetric key* system because the same key is used at both ends of the process.

As the check is sent over the network, it is unreadable without the key. The next challenge is to securely deliver the symmetric key to the bank.

Public-key cryptography for delivering symmetric keys

Public-key encryption is used to solve the problem of delivering the symmetric encryption key to the bank in a secure manner. To do so, you would encrypt the symmetric key using the bank's public key. Since only the bank has the corresponding private key, only the bank will be able to recover the symmetric key and decrypt the check.

Why use this combination of public-key and symmetric cryptography? The reason is simple. Public-key cryptography is relatively slow and is only suitable for encrypting small amounts of information – such as symmetric keys. Symmetric cryptography is much faster and is suitable for encrypting large amounts of information.

The following illustration describes what Entrust does behind the scenes to deliver the secure electronic check.