Entrust Technologies White Paper

Entrust® Key Management Overview

Author: Ian Curry
Date: April 1996
Version: 1.4

1. Introduction

As products and services move from paper-based to electronic transactions in a distributed computing environment, there is a growing need for information security. The dramatic benefits of reduced costs, efficiency, and quality of service improvements make it imperative that these security hurdles be overcome. Information security in the electronic world is based on cryptographic techniques and algorithms that use keys.

The central question that businesses must face, however, is the following: "Where do cryptographic keys come from and how does a business manage those keys effectively and securely to protect its information?" To answer that question, any key management infrastructure must address the following key management concerns:

- key generation, backup, update, and revocation
- establishment and maintenance of trust relationships among security domains
- scalability and affordability
- application integration and user transparency

The Entrust product family is specifically designed to address these issues. Entrust provides automatic and transparent key management services to end-users so they can easily take advantage of encryption and digital signature services.

Entrust is based on the concepts of public-key cryptography. As such, Entrust allows organizations to have a single, highly-scalable security infrastructure to which all applications connect for their individual security requirements. With Entrust, organizations can administer security once for a wide range of applications rather than administering separate security solutions for individual applications.

2. The key management problem

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 preventing 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. While it is safe to send encrypted messages without fear of interception (because an interceptor is unlikely to be able to decipher the message), there always remains the problem of how to securely transfer the key to the receivers 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. 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.

Consequently, 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?"

Entrust combines symmetric and public-key cryptography to provide answers to key management questions such as those listed above.

3. Entrust

An important goal of Entrust is to provide transparent key management to end users, thereby making true information security attainable without requiring users to manage their keys. The central goal of Entrust is to provide a single security infrastructure that transparently manages keys across all of the applications in the network.

To achieve these goals, an Entrust system consists of a number of components that work together to provide a secure environment. The following discussion introduces each component of the system.

Figure 1 provides an overview of the Entrust product architecture.

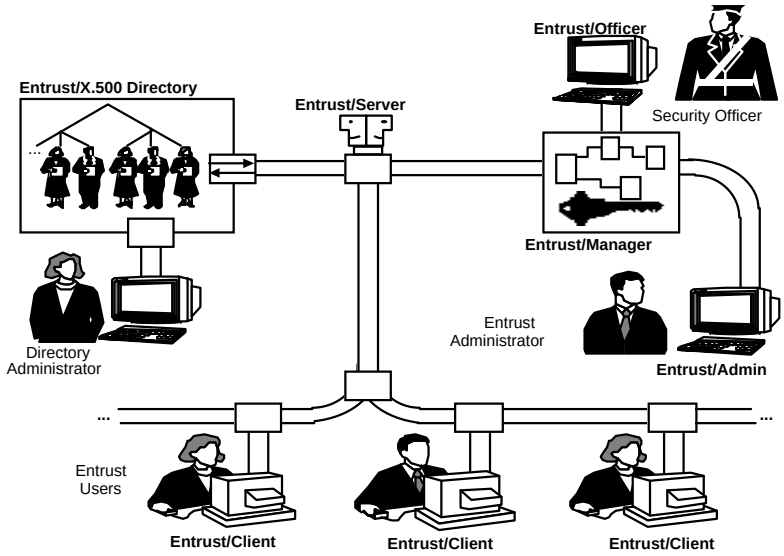

Figure 1: Entrust System Architecture

Entrust/Manager Entrust/Manager is the Certification Authority (CA) for the system. The CA is responsible for issuing all certificates in the security domain In addition to being a CA, Entrust/Manager holds a secure database containing users' encryption key pair histories.

Entrust/Officer Using Entrust/Officer, Security Officers specify the security policies governing the day-to-day operation of the system. These security policies are implemented by Entrust/Manager and Entrust/Client.

Entrust/Admin Using Entrust/Admin, Administrators create, recover, and delete users. In addition, Administrators revoke untrusted certificates.

Entrust/X.500 Directory

Entrust/X.500 Directory (referred to as the Directory) is a directory system supporting the X.500 standard. Entrust uses the Directory to store users' encryption public key certificates. Storing encryption certificates in a Directory allows Entrust-aware applications to easily access public key certificates so that users can encrypt information for each other. In addition to encryption certificates, Entrust stores certificate revocation lists (CRLs) in the Directory.

Entrust/Server Entrust/Server represents the central focus of communications within an Entrust system. Entrust/Server is a software process that manages all communications among Entrust/Manager, Entrust/X.500 Directory, and Entrust/Client.

Entrust/Client Entrust/Client is an application that presents a simple graphical user interface to Entrust users so they can perform encryption and digital signature operations on files.

4. Key management

With an understanding of the components of Entrust, it is now possible to discuss how these components work together to provide key management. Without question, key management is a complex subject. This section provides only a brief introduction to some important key management concepts in Entrust.

4.1 Creating Entrust Users

Entrust Administrators use Entrust/Admin to enable users, thereby making those users official subscribers to Entrust. Only users who have entries in the Directory can become Entrust users. Once enabled, users are able to run Entrust/Client.

The first time users run Entrust/Client, they enter a reference number and an authorization code given to them by their Entrust Administrator. Once this entry is complete, Entrust/Manager and Entrust/Client set up a secure communications session to establish the user's initial encryption key pair. At that time, the Client also creates the user's signing key pair.

All private keys passed from Entrust/Manager to Entrust/Client are encrypted to prevent someone eavesdropping on the network from obtaining the keys.

Entrust/Client stores each user's encryption key pair and signing key pair (in an encrypted form) in the user's profile. Each user has a profile that stores the user's keys in addition to other Entrust information specific to that user.

Users' signing private keys, which are used to create digital signatures, are not backed up in the Entrust/Manager database. The signing private key is solely owned by the user, and no one else, including Entrust Administrators and Security Officers, has access to that key.

A copy of each user's encryption key pair is securely backed up in the Entrust/Manager database. Thus, users' encryption key pairs can be recovered if they are ever lost (for example, when users forget their passwords -- refer to section 4.3 for more information on recovering lost keys).

A copy of each user's encryption public key certificate is stored in the Directory. Storing encryption public key certificates in the Directory allows Entrust applications to access keying material so that users can easily encrypt information for each other.

4.2 Encrypting a File with Entrust

From a key management perspective, the difficulty in file encryption relates to specifying the recipients of the encrypted data. The recipients of an encrypted file are those individuals authorized to decrypt the file. There can be numerous recipients of any single encrypted file. The user encrypting a file has complete control over selecting recipients.

Fortunately for users, they do not need to be aware of other users' keys to select recipients and encrypt information for those recipients. When a user selects recipients, Entrust/Client obtains the encryption public key certificate of each recipient from the Directory. With Entrust's key management infrastructure supporting users, encrypting a file for multiple recipients is a simple task involving the selection of recipients' names from a menu. Without Entrust's key management, it would be difficult for a user to encrypt a file for multiple recipients without having a substantial understanding of cryptography.

4.3 Recovering Lost Keys

Many data security systems require that users have passwords to access their keys. However, with some information security systems, disaster strikes when users forget their passwords. When users of such systems forget their passwords, not only do they lose their keys -- they also lose all the information encrypted with those lost keys (forever). Entrust, however, provides an easy way to securely recover keys.

If users forget their Entrust passwords, they call their Entrust Administrator to request that their encryption key pairs be recovered. After setting up the users for key recovery using Entrust/Admin, the Entrust Administrator provides each recovered user with a new reference number and authorization code. Users then enter this information as part of the Recover User operation in Entrust/Client. This operation sets up a protected communication session between Entrust/Client and Entrust/Manager for encryption key pair recovery. When Entrust/Client recovers a user's encryption key pair, it automatically creates a new signing key pair.

4.4 Updating Keys

While users can recover their key pairs if they forget their passwords, there are some situations in which users may want to get new key pairs altogether. For instance, the security policy of an organization may dictate that users get new key pairs every two years. In this case, a Security Officer uses Entrust/Officer to specify that key update occurs every 24 months. The term key update refers to the automatic creation of new encryption key pairs and signing key pairs at specified times. Any Security Officer can specify the frequency of key updates.

Every time users log on to Entrust, the software checks to see if key update is required. If this is the case, Entrust/Client makes a request to Entrust/Manager for a new encryption key pair. Once the new encryption key pair is generated, Entrust/Manager delivers the new key pair in a protected communications session. Delivery of the new encryption key pair is *completely* transparent to the user.

Entrust/Client does not delete the old encryption key pair when a new encryption key pair is delivered to the user. If the old encryption key pairs were deleted, it would be impossible to decrypt information that was originally encrypted with the old keys. To address this problem, Entrust/Client maintains a full history of encryption key pairs. Thus,

when a user attempts to decrypt a file originally encrypted with old keys, Entrust simply looks back through the key history to find the appropriate key to decrypt the file.

The concept of a key history is also intimately related to the concept of key recovery. Whenever users recover keys, it is their entire history of encryption key pairs that is securely downloaded to Entrust/Client.

When key update occurs for a signing key pair, Entrust/Client automatically generates a new signing key pair for the user. The public part of the signing key pair (known as the verification public key) is securely communicated to Entrust/Manager. Entrust/Manager creates the user's verification public key certificate and returns the certificate to Entrust/Client.

4.5 Revoking Certificates

There are times when it is necessary to completely revoke a user's certificates (for example, when someone leaves the organization or a key becomes untrustworthy for any reason). In these situations, Entrust Administrators run a simple command to place the user's certificates on a certificate revocation list (CRL) stored in the Directory. To revoke a certificate, the CA places the serial number of the certificate containing the untrusted public key on a CRL. Every certificate in the system has a unique serial number.

Once a certificate is on a certificate revocation list, no application should use the key in the certificate to encrypt information for the certificate's owner. With Entrust, when a user verifies a digital signature associated with a revoked certificate, the user is notified that the certificate has been revoked and the signature should not be trusted. Thus, certificate revocation provides an efficient, centralized mechanism to notify all users of invalid certificates.