Entrust Technologies White Paper

Entrust Overview

Author: Ian Curry
Date: October 1996
Version: 1.0

1. Introduction

This paper provides an overview of the Entrust™ family of network security products. Entrust is designed to be the single security infrastructure for organizations; as such, it provides numerous security services to system administrators, network users, and applications. In particular, Entrust provides automatic and transparent key management so that neither application developers nor end-users need to understand the details of cryptography to take advantage of Entrust security services.

As an unbiased vendor in the software industry, Entrust Technologies ensures that Entrust meets the following requirements:

- Entrust is based on open standards; for example, the X.509 certificate standard, the Generic Security Service (GSS) programming interface standard, the Simple Public-Key Mechanism (SPKM) standard, and the Lightweight Directory Access Protocol (LDAP) standard.

- Entrust is highly scalable. As the single security architecture for large organizations, all aspects of Entrust meet the requirements of large-scale networks.

- Entrust provides cross-platform and cross-application security so that security requirements do not "lock" customers into specific platforms and applications. Some of the supported operating systems are Windows (3.1x, 95, and NT), HP-UX, Solaris, SunOS, AIX, Digital UNIX, and Macintosh.

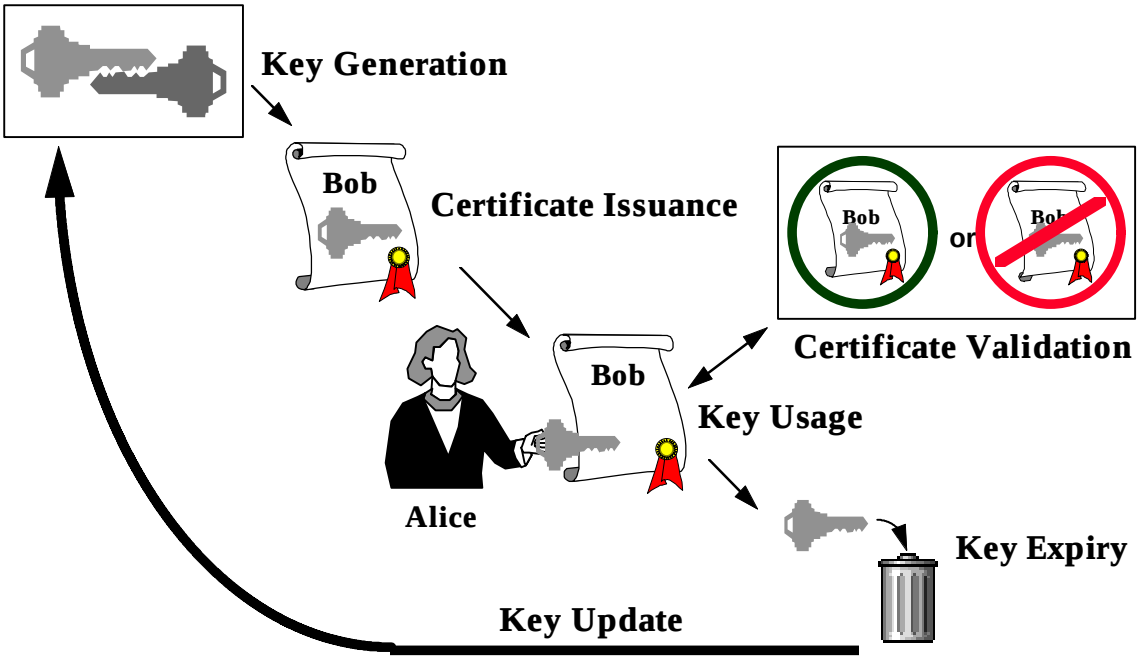
- Entrust is algorithm independent. The product family supports numerous public-key algorithms (RSA, DSA, Diffie-Hellman), hashing algorithms (SHA-1, MD5), and symmetric algorithms (CAST, Triple-DES, DES, RC4, RC2).

The first section of this paper briefly describes some of the critical issues in key lifecycle management. The second section describes the Entrust system architecture and how the various Entrust components work together to provide key lifecycle management. This section also describes the cryptographic algorithms and key lengths that Entrust supports.

2. Key lifecycle management

The term *key lifecycle management* refers to the process of managing cryptographic keys automatically, transparently, and securely over their lifetimes. For security reasons, cryptographic keys must have a well-defined life span. In addition, it is unreasonable to expect users within organizations to understand cryptography well enough to manage keys securely. Consequently, keys must be managed automatically and transparently over their lifecycles. The central goal of Entrust is to provide key lifecycle management.

The diagram below shows the critical phases in the lifecycle of a public-key pair: key generation, certificate issuance, key usage and certificate validation, key expiry, and key update.



Some of the important issues in key lifecycle management handled by Entrust are as follows:

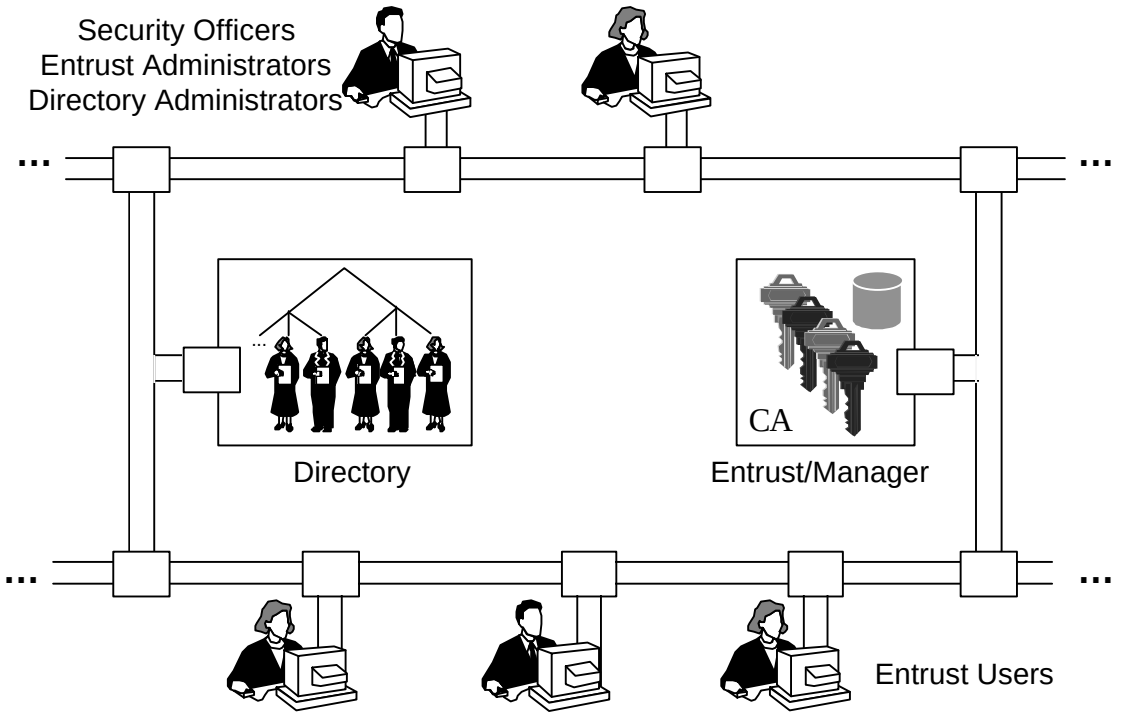
- use of a proven random number generator to generate secure public-key pairs and symmetric keys
- secure storage of private keys
- use of two key pairs, one for encryption and one for digital signature, to meet the conflicting requirements of key backup and non-repudiation
- support for backup of encryption key pairs, so users can recover their keys and encrypted information when they forget their passwords

- use of a Certification Authority (CA) to issue public key certificates
- use of a scalable directory system to hold public key certificates so users can easily encrypt data for each other
- use of a scalable certificate revocation system so users know whether or not to trust a certificate before it is used
- the capability to cross-certify with other domains
- automatic and transparent update before keys expire so users maintain service without interruption

3. Entrust system architecture

Entrust is a family of products that meet the requirements of a public-key infrastructure. To meet those requirements, the Entrust components work together to transparently and automatically manage keys and certificates for users and applications.

The following diagram shows the typical network configuration for an Entrust system.



Entrust provides security for client-server networks. As such, an Entrust system consists of both server-side and client-side products. The server-side products are often collectively referred to as the *Entrust infrastructure*. The next sections of this paper describe the infrastructure and client-side architecture of an Entrust system.

3.1 The Entrust infrastructure

The Entrust infrastructure consists of three components: Entrust/Manager, the Directory, and Entrust/Admin.

Entrust/Manager is the central component of an Entrust system. As the Certification Authority (CA), Entrust/Manager signs all certificates in the system. Entrust/Manager also contains a secure database to back up users' encryption key pairs.

Client-side workstations rarely communicate with Entrust/Manager. At first, users communicate with Entrust/Manager to get enabled on the system. Once users are enabled, they communicate with Entrust/Manager only during key update operations. Key update operations typically occur once every one or two years (or more).

Besides handling user initialization and key update requests, Entrust/Manager issues certificate revocation lists (CRLs). Entrust/Manager also performs cross-certification operations with other trusted Certification Authorities.

Whereas Entrust/Manager acts as the Certification Authority for the system, the Directory is a publicly-accessible storage method for users' encryption public key certificates and CRLs. Since users must access encryption public key certificates during encryption operations, client-side workstations communicate with the Directory on a frequent basis. In addition, client-side workstations need to retrieve CRLs from the Directory to verify the trustworthiness of certificates. Certificates must be verified when users encrypt data for each other and when they verify digital signatures.

Many products meet Entrust's requirements for a Directory. Client-side workstations, Entrust/Manager, and Entrust/Admin all communicate with the Directory using the Lightweight Directory Access Protocol (LDAP). LDAP is now widely-recognized as the leading directory protocol and, as such, is supported by a wide variety of directory products. Entrust should be able to work with any product that fully supports LDAP.

Directories based on the X.500 standard are traditional examples of highly-scalable directory systems, and many X.500 directories support LDAP. The features inherent in the X.500 standard (for example, scalability, robustness, and performance) make these directory products well-suited to the requirements of a public-key infrastructure such as Entrust. Entrust Technologies currently has Entrust customers working with at least six different X.500 products. While Entrust is proven to work with numerous X.500 products, the product family is also proven to work with non-X.500, LDAP-compliant products.

Entrust/Admin, the third component of the Entrust infrastructure, is not a server process like Entrust/Manager and the Directory. Rather,

Entrust/Admin provides administrative capabilities to three types of personnel: Security Officers, Entrust Administrators, and Directory Administrators.

Security Officers define the high-level security policies governing the operation of an Entrust system. For instance, they define the default lifetimes for encryption and signing key pairs, and the frequency with which CRLs are distributed from Entrust/Manager to the Directory. In addition, Security Officers execute cross-certification operations with other Certification Authorities.

Entrust Administrators perform the day-to-day tasks in administering the system. These tasks include creating and deleting users, changing user names, helping users recover their keys when they forget their passwords, and revoking users' certificates when circumstances warrant such action. Many of the tasks that Administrators do can be automated and done in bulk to reduce administrative effort.

Directory Administrators use Entrust/Admin to perform administrative tasks associated with the Directory; for example, adding users to the Directory. Directory Administrator tasks, like those of Entrust Administrators, can generally be done in bulk.

One person can be simultaneously a Security Officer, an Entrust Administrator, and a Directory Administrator. However, Entrust distinguishes among these roles so that particular individuals can be given different privileges to run a limited set of functions.

3.2 The Entrust client-side architecture

In the context of this paper, the term *client-side workstation* refers to any workstation other than those supporting the Entrust infrastructure. In this context, an application server (for instance, a database server) is referred to as a "client-side workstation" because application servers require the same services from the Entrust infrastructure as application clients.

The client-side architecture of Entrust consists of the following components: Entrust/Toolkit, Entrust/Engine, and Entrust/Client.

Entrust/Toolkit is a family of standards-based, high-level security application programming interfaces (APIs). These APIs provide security services, including full key lifecycle management, to a broad spectrum of applications. For instance, the EntrustFile and GSS-IDUP-API Toolkits provide security services to store-and-forward applications such electronic mail, electronic forms, and electronic data interchange (EDI). Variants of these interfaces provide applications with different security and data formatting standards, such as S/MIME and MSP. The EntrustSession Toolkit uses the standard GSS-API to provide security services to real-time, transaction-oriented applications such as Web browsers and servers.

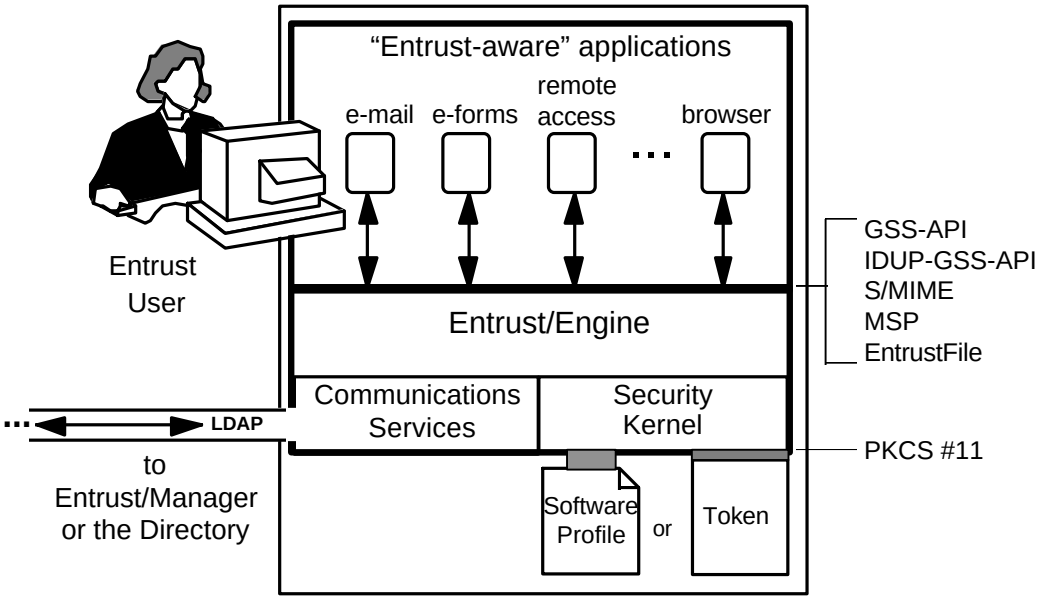
Key management is the most difficult aspect of adding security to an application. Entrust/Toolkit hides the complexities of key management from application developers so they do not need to hire security experts to make their applications secure. Moreover, because Entrust/Toolkit handles the most complex aspects of security for application developers, it prevents developers from making serious security errors. Finally, the high-level nature of the programming interfaces means developers can add security to their products quickly and obtain time-to-market advantages over competitors.

Most important, however, is that developers producing "Entrust-aware" applications do not increase administrative overhead for their customers. Because Entrust-aware applications use the Entrust infrastructure for security services, customers only administer security *once* for all Entrust-aware applications.

Entrust/Engine is the run-time software implementing the interfaces in Entrust/Toolkit. For each variant of Entrust/Toolkit, there is a corresponding variant of Entrust/Engine. For example, the EntrustSession Engine implements the functions provided in the EntrustSession Toolkit.

When Entrust is installed on client-side workstations, all variants of Entrust/Engine are installed. As a result, any Entrust-aware application is automatically enabled to provide services to users without requiring additional software.

The following diagram shows the software architecture on client-side workstations running "Entrust-aware" applications. The labels on the right side of the diagram represent the standard interfaces and formats currently supported by Entrust.



Entrust-aware applications, shown at the top of the diagram, communicate with Entrust/Manager and the Directory through Entrust/Engine. Besides communicating with Entrust/Manager and the Directory, Entrust/Engine provides access to the user's chosen method for storing keys. Entrust currently supports two methods for storing keys securely.

Hardware key storage devices, generically known as *tokens*, are optional security components such as PC cards (formerly known as PCMCIA cards) that contain cryptographic keys or algorithms (or both) for use in environments implementing strict security standards. Entrust supports the PKCS #11 standard interface to tokens and currently works with two PC card devices: Entrust/Token from Entrust Technologies and the *iPower* token from National Semiconductor. For environments not requiring tamper-proof tokens, secure software profiles can be used to store users' keys. Although smart cards are not commercially supported at this time, various demonstrations have shown interworking between Entrust and smart cards.

The symmetric and public-key algorithms in Entrust/Engine are stored in the security kernel. The Entrust security kernel is the first and only software product validated by the US and Canadian Governments to the FIPS 140-1 standard. This validation means that Entrust is approved for securing any sensitive but unclassified information in the US and Canadian Governments. Validation offers Entrust customers the comfort and confidence that an independent, government-approved agency has closely scrutinized Entrust's cryptographic software and agrees that it complies with the strictest security requirements. The validation process typically takes between twelve and eighteen months.

With respect to public-key algorithms, the security kernel supports the RSA algorithm for encryption and digital signature. The security kernel also supports the Digital Signature Algorithm (DSA) for digital signature and the Diffie-Hellman algorithm for key exchange. Entrust uses 1024-bit public-key pairs for both encryption and digital signature.

In terms of symmetric encryption algorithms, the security kernel supports CAST, Triple-DES, DES, RC4, and RC2. The domestic variant of Entrust uses CAST with 128-bit keys.

Entrust/Client, the final piece of Entrust software on client-side workstations, is an Entrust-aware application that allows users to easily encrypt and digitally sign files. Entrust/Client supports secure e-mail over Microsoft Exchange™, Microsoft Mail™, cc:Mail™, and QuickMail™, and the ability to securely delete files. Entrust/Client on Windows and UNIX comes with a command-line interface, making it straightforward to integrate Entrust/Client into applications that support macros (for example, Microsoft Word™). On the Macintosh®, Entrust/Client supports AppleScript™, a high-level macro language which lets users write simple scripts for numerous functions (for

example, automatically encrypting one or more folders on a disk at Shut Down).