Entrust®

Key Update and the Complete Story on the Need for Two Key Pairs

Date: August 2000
Version: 1.2

Introduction

Support for two key pairs is a fundamental requirement for public-key infrastructures (PKIs). One key pair is for encryption operations; the other key pair is for digital signature operations. This paper describes the business requirements which preclude using a single key pair for encryption and digital signature operations.

Since the first release of the Entrust® product family in 1994, Entrust has provided comprehensive management of two key pairs (sometimes called "dual-key support"). There are three business requirements related to supporting two key pairs per user:

- support for key backup and non-repudiation
- support of different algorithms for encryption and digital signature
- support for updating encryption key pairs and managing decryption key histories, and updating signing key pairs and destroying signing keys

The paper discusses important issues related to key update and two key pairs, and the hidden costs of other vendors' "PKI" products. Other vendors, particularly certificate issuance service providers, do not discuss these hidden costs because they only provide a small portion of what is needed for a comprehensive PKI solution.

Note: While Entrust provides managed support for two key pairs, the system also supports applications (for example, popular Web browsers and servers) that only implement basic single-key-pair support.

This paper assumes the reader has a basic understanding of public-key cryptography. To get a brief overview of cryptography, refer to the White Paper titled *An Introduction to Cryptography*, available on the Entrust Web site at *http://www.entrust.com*.

PKI - Starting Right from the Beginning

Customers want applications to function according to customer requirements—not vice versa. Entrust allows customers to control management of their keys and benefit from consistent behavior across applications. This consistency provides security, ease of use, and low-cost operation. Policies for managing keys rest in the hands of administrators, and out of the hands of users and application developers.

Customers want applications to function according to customer requirements—not vice versa. Entrust puts administrators in control of how keys are managed across applications and platforms. Key management is transparent to users and application developers.

Entrust provides a complete solution —today— that is applicable over the long term as customers expand the uses of their PKIs to new applications. Administrators and users appreciate this solution because it ensures consistency across applications and platforms, providing benefits such as secure single sign-on.

Application developers appreciate the solution because it removes complexity from their applications. For application developers, Entrust improves time-to-market and product quality, and decreases maintenance costs. Most importantly, Entrust provides customers with what they want—a flexible PKI that behaves consistently across applications and platforms.

Entrust provides a mature, proven solution. For example, even though Entrust has supported the concept of two key pairs since 1994, most other "PKI" products still do not support two key pairs. A few vendors claim they will support two key pairs, but on closer examination, these planned products frequently do not meet the necessary requirements and none provides consistent behavior and support across applications and platforms like Entrust does. Other vendors' unmanaged products force customers to function according to individual application requirements—exactly what customers do not want.

Unlike other products, it does not cost extra to use two key pairs with Entrust. Entrust provides a complete solution today—without the hidden costs of other solutions—that will be applicable as customers expand the uses of their PKIs to new applications.

Unlike other products, it does not cost extra to use two key pairs with Entrust. This is because Entrust provides software that manages keys and certificates for customers, as opposed to other vendors who primarily sell limited certificate issuance services or products. With those services or products, customers will have to pay substantially more to use applications that support two key pairs.

The remaining sections of this paper explain why support of two key pairs is a fundamental requirement for PKIs.

Support for Key Backup and Non-repudiation

The first requirement driving the need for two key pairs is support for key backup and non-repudiation. It is not possible to meet these two requirements with single-key-pair systems.

Key Backup and Recovery

An organization must be able to retrieve encrypted data when users can no longer access their decryption keys. This means that the organization to which the user belongs requires a system for backing up and recovering the decryption keys. There are two reasons why key backup and recovery are important to organizations.

The first reason is that users forget passwords. It is potentially catastrophic for an organization to lose data when users forget the passwords required to access their decryption keys. Valuable information can be lost forever if there is no ability to securely recover those keys. Furthermore, unless users know they can always recover their encrypted data (even if they forget their passwords), some users will not encrypt their most valuable and sensitive information for fear of losing it—even though that information needs to be protected the most.

Key recovery is required when users forget passwords or lose access to their keys. Without key recovery, encrypted data can be lost forever. Entrust provides a unique solution for key backup that works across applications and platforms.

The second reason is that users may lose, break, or corrupt the devices in which their decryption keys are stored. For instance, if a user's decryption keys are stored on a magnetic card, the magnetic field on the card can become corrupted. Alternatively, a user's smart card might get lost or physically broken. Again, permanent loss of those decryption keys can be disastrous. Users are prevented from recovering encrypted data unless their decryption keys are backed up.

Entrust provides a unique, proven solution for key backup and recovery. Unlike other vendors' proposed solutions, the system ensures keys are properly backed up across applications. With announced key backup schemes from other vendors, customers would have to determine whether or

not each individual application properly supports key backup. Each application would have to be evaluated and tested separately, and users would have to determine how each application backs up its keys. Worst of all, with the backup schemes proposed by one certificate issuance service provider, customers would be *completely locked in* because *only* the vendor (not the customer) will hold the keys ultimately required for recovery operations. Thus, customers would have to contact the vendor and receive their cooperation before recovering any keys. With Entrust, however, customers know —for sure— that keys are properly backed up for all Entrust-Ready applications and can only be recovered under policies established by the customer. These policies include the number of administrators required to authorize key recovery operations.

Which keys require backup?

Earlier, this paper introduced the notion of different functions for key pairs. One key pair is used for encrypting and decrypting data. This is called the "encryption key pair". Another key pair is used for digitally signing data and verifying signatures. This is called the "signing key pair". Note that there is no discussion above regarding backup and recovery of signing key pairs. The only keys requiring backup are users' decryption keys. As long as a trusted agent (for example, a central authority inside the organization) securely backs up users' decryption keys, data can always be recovered. However, signing keys have different requirements from decryption keys. In fact, as the next section describes, backing up signing keys destroys the basic requirement of non-repudiation in a PKI.

Support for Non-Repudiation

Repudiation occurs when an individual denies involvement in a prior transaction. For example, when someone claims a credit card is stolen, this means that he or she is disclaiming liability for transactions that occur with that card anytime after reporting the theft. Non-repudiation means that an individual cannot successfully deny involvement in a legitimate transaction. In the paper-world, individuals' signatures legally bind them to their transactions (for example, credit card charges, business contracts, …). The signature prevents repudiation of those transactions. In the electronic world, *digital signatures* support non-repudiation because they are the electronic replacement for pen-based signatures.

The signing private key

Non-repudiation requires that users generate and control access to signing keys. Unlike decryption keys, backing up signing keys is neither required nor acceptable.

The most basic requirement for non-repudiation is that the key used to create digital signatures—the signing private key—be generated and securely stored under the control of the user at all times. It is not acceptable to back up the signing key.

Another basic requirement for non-repudiation is to ensure that the signing key is protected by a strong local password and/or is maintained on a protective device (for example, a smart card) that the user carries. With Entrust, password rules are configurable so that customers can enforce passwords that meet the security requirements of their environment. Users' local passwords are never stored on disk and never go over the network.

Since popular browsers currently do not ensure that users protect their signing keys with even a basic password, it is not possible to claim that these applications support non-repudiation. Certificate issuance service providers,

who do not customarily provide user software and yet claim to support non-repudiation, also cannot ensure that users' signing keys are properly protected; this is the most basic issue that refutes these service providers' claims to support non-repudiation.

Unlike encryption key pairs, there is no technical or business requirement to restore previous signing key pairs when users forget their passwords or lose access to their keys. In these situations, the secure solution is to create new signing key pairs. Signatures created with previous private keys can still be verified because the corresponding public keys are actually stored with the signed data.

The need for two key pairs

It is challenging for most PKI vendors to simultaneously support key backup and recovery *and* non-repudiation. To support key backup and recovery, the decryption keys must be backed up securely. To support non-repudiation, the keys used for digitally signing cannot be backed up and must be under control of the user.

Thus, to meet these requirements, a PKI must support two key pairs for each user. Entrust provides this capability because each user has one current key pair for encryption and decryption, and a second key pair for digital signature and signature verification.

While other PKI vendors claim to support two key pairs, Entrust has provided comprehensive management of two key pairs since its initial release in 1994. Other PKI vendors frequently only issue certificates to applications and require each individual application developer to support two key pairs. There is little chance that general application developers, the vast majority of whom are not experts in cryptography and data security, will support two key pairs properly—if at all. Entrust removes the complexity of managing two key pairs from application developers and users.

Key Update

The second requirement for two key pairs relates to updating key pairs over time. The term *key update* refers to the process of creating completely new key pairs for existing users—meaning that new public keys and new private keys are generated.

With Entrust, key update is a zero-cost operation because it occurs automatically and transparently, without requiring intervention by administrators or users. Like support for multiple algorithms, key update is an inherent capability customers obtain when they license Entrust software. There are no extra or hidden costs to update keys.

To ensure users do not experience unexpected "down-time" when keys expire, Entrust automatically updates key pairs before they expire. This process also avoids expensive support calls from users who should not have to experience denial-of-service due to expired keys.

Key update refers to the process of creating completely new key pairs for users. With Entrust, key update is a zero-cost operation because it occurs transparently to users and administrators.

Entrust believes that customers should choose the frequency of updating key pairs according to security policy, and not due to the costs or administrative overhead required in other solutions. Some other vendors provide "certificate renewal" tools at additional cost. However, these tools only update certificate lifetimes and do not address the fundamental security requirement to change the actual key pair itself. Since other products

do not provide key update, it is difficult to determine how much it would cost to achieve similar functionality with those products.

The Difference between Key Update and "Certificate Renewal"

Key update is a more sophisticated and valuable feature than "certificate renewal". Certificate renewal refers to a process in which an existing public key is extracted from one certificate and put into a new certificate with a different lifetime. Compared to key update, certificate renewal is a simple process because it does not involve management of a completely new key pair.

Unlike key update, however, certificate renewal does not benefit users. For instance,

- certificate renewal *only* works when browsers connect to specific, trustworthy Web sites that are enabled to re-issue certificates. Therefore, certificate renewal is limited to Web browsers and, more specifically, is only done if the user connects to a designated, trustworthy Web site—certificate renewal is not done if users only use the browser for e-mail or do not connect to one of these special Web sites.
- there is no cryptographic justification or advantage to perform certificate renewal. If a public key is intended to be used for 36 months, then the public key should be issued in a certificate with a lifetime of 36 months. Some certificate issuance service providers require certificate renewal because they usually issue certificates for only one year at a time to enforce an annual subscription service.
- from an administrative standpoint, certificate revocation gets much more complicated with certificate renewal. Since the same public key is put into multiple certificates, each of those certificates needs to be tracked and revoked if the single corresponding private key is compromised. It is unlikely that many software vendors would implement this properly.
- if the certificate is for encryption, then the certificate will be distributed to others during its lifetime so those people can encrypt for the certificate's owner. When the certificate is "renewed", it will have to be re-distributed to all those people because the old certificate will have expired. This process is difficult and cumbersome for users. Because certificate issuance service providers usually issue certificates for only one year, each user has to go through this process on an annual basis. (This process is analogous to changing your phone number or mailing address on an annual basis—imagine how difficult communicating would be if your phone number or mailing address changed on a yearly basis and there was no automated system to inform others of these changes!)

"Certificate renewal" is not required and does not benefit users—in fact, it can make their lives more difficult and cumbersome than necessary.

Unlike "certificate renewal" which is limited to Web browsers, Entrust's zero-cost key update process works transparently and simultaneously across all Entrust-Ready applications (for example, e-mail, disk encryption, e-forms, databases, remote access, Web applications, ERP applications, …).

Benefits of Key Update

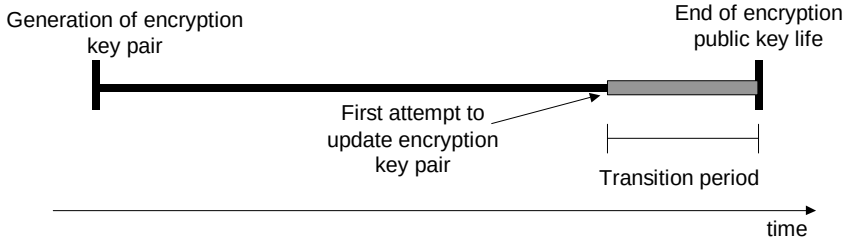
On the one hand, certificate renewal is not required and does not benefit users—in fact, it can make their lives more difficult and cumbersome than

necessary. On the other hand, key update is necessary and benefits users in the following ways:

- key update provides an automated mechanism for restricting the amount of data which may be exposed when a private key is compromised. Sound security policies restrict data exposure if compromises occur; therefore, updating encryption and signing key pairs is a fundamental component of sound security practices.
- key update provides a transparent way to change algorithms and/or key lengths (for example, a user wants to change from 1024-bit RSA to 2048-bit RSA). Changing algorithms or key lengths in an automated and transparent manner, as in Entrust, is necessary because cryptography and key management are complex issues that should be hidden from users.
- when a certificate is revoked, key update provides a method for issuing and managing the new key pair that may need to be issued.
- from an administrative standpoint, key update also eliminates the complexities of certificate revocation created by "certificate renewal" systems (described above).
- when done prior to key expiration (as in Entrust), transparent key update protects users from unexpected and costly "down-time". Since expired keys should not be accepted for authentication or encryption, they need to be updated prior to expiration.
- key update provides the only secure mechanism to change important policy information in certificates. When users' privileges change, as reflected in their certificate policies, they must get completely new key pairs and certificates through key update. "Certificate renewal" systems put the same public key in a different certificate and, therefore, do not provide a secure way to change important information like certificate policies. If certificate renewal were used to change a certificate policy for digital signatures, for example, it would be difficult (and perhaps impossible) to determine whether a signature was created in conjunction with the policy in an older certificate or the new certificate—clearly an untenable situation for valuable e-commerce transactions.

Encryption Key Pair Update and Key Histories

With Entrust, customers choose the lifetimes of their key pairs and control how key update occurs. Automatic and transparent update of encryption key pairs is driven by the expiration of the encryption public key. The following shows a sample lifeline of an encryption key pair:



The *transition period* represents the time between the first attempt at updating the key pair and the official expiration date of the encryption public key. Entrust provides a transition period of one hundred days (or 50% of the key's lifetime, whichever is less). Attempting to update the encryption key pair well

before it expires ensures a successful update process. If the user is off-line, for example, on the first update request, the system continues to function normally. In such a case, Entrust requests another update process the next time the user logs in.

When encryption key pairs are updated, the decryption private keys are retained and managed in the users' key histories. These keys are retained to decrypt any data encrypted with the corresponding public keys. Securely and transparently managing users' histories of decryption private keys is a fundamental requirement of a PKI.

Managing histories of decryption private keys

The *key history* ensures that users can easily access any of their encrypted data. (When data is encrypted with a user's encryption public key, only the corresponding decryption private key—the paired key—can be used for decrypting). To ensure ease of use and transparency, PKI software must automatically manage users' histories of decryption keys across applications.

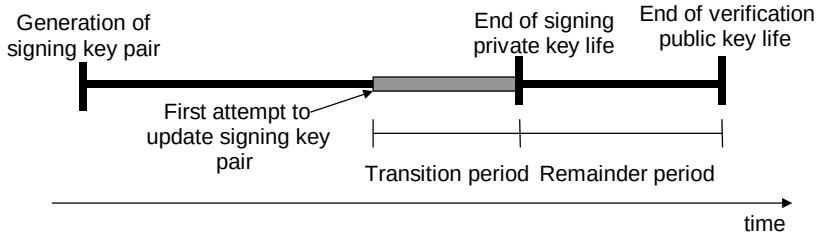
To ensure ease of use and transparency, the PKI software must automatically manage users' histories of decryption keys across applications.

The key history must also be securely managed by the key backup and recovery system. This ensures that encrypted data can always be recovered, regardless of which of the user's encryption public keys was used to originally encrypt the data (and, by extension, regardless of *when* the data was encrypted).

Signing Key Pair Update and Destruction of Signing Keys

Unlike update of encryption key pairs, which is driven by the expiration of the *public* key, update of signing key pairs is driven by the expiration of the *private* key. The two parts of a signing key pair are known as the *verification public key* and the *signing private key*.

The following shows a sample lifeline of a signing key pair:



When a new signing key pair is generated during key update, the previous signing private key is securely destroyed. This destruction ensures that nobody can recover the private key, further enhancing support for non-repudiation. There is no need to retain this private key once it expires.

The time between the expiration of the private and public keys is known as the *remainder period*. Even though the private key has expired, users can still validate signatures during the remainder period with the verification public key.

In single-key-pair products, the need to destroy the private key when updating the signing key pair conflicts with the need to retain the private key when updating the encryption key pair. A single-key-pair product cannot both retain the private key and destroy it; only a two-key-pair solution like Entrust can meet these needs.

In single-key-pair products, the need to destroy the private key for signing conflicts with the need to retain the private key for decrypting. Only a two-key-pair solution like Entrust can meet these needs.

Summary

This paper discussed the following reasons why PKIs need to support two key pairs:

- *support for key backup and non-repudiation*

To support data recovery, private keys for decrypting must be backed up securely. To support non-repudiation, private keys for signing must not be backed up. These conflicting requirements can only be solved by a comprehensive PKI that supports two key pairs in this manner.

- *update of encryption and signing key pairs*

When updating encryption key pairs, the private keys for decryption need to be retained and managed. When updating signing key pairs, the private keys for digitally signing must be securely destroyed to prevent future recovery. These conflicting requirements can only be solved by a comprehensive PKI that supports two key pairs.

As discussed, encryption and signing key pairs are fundamentally different entities that have distinct business requirements. These business requirements can only be met with a comprehensive PKI that works across applications and platforms.

This paper also described how Entrust provides automatic and transparent key update at no cost to administrators, users, or application developers. Entrust's solution for key update and managing two key pairs has been proven to work effectively for customers and application development partners since 1994. Finally, this paper discussed the business benefits of key update and the hidden costs and inadequacies of other vendors' "certificate renewal" schemes.

Like many features in Entrust, key update and support for two key pairs put the customer in control. Entrust's unique key and certificate management solution ensures applications meet customer requirements—not vice versa.