IBM Client Security Software Deployment Guide
Version 5.41

Updated: January 7, 2005

Fifth Edition (January 2005)

Preface

IT administrators must understand and plan for numerous factors when deploying IBM® Client Security Software. This guide is not intended to explain how to use the Embedded Security Subsystem Chip or Client Security Software; rather it is a guide for how to deploy the software to Embedded Security Chip-equipped computers across an enterprise.

Audience

This guide is intended for IT administrators, or those who are responsible for deploying IBM Client Security Software Version 5.41 (CSS) on computers in their organization. The guide is intended to provide the information required for installing IBM Client Security Software on one or many computers. Please read the *IBM Client Security Software Version 5.41 Administrator and User Guide* as a prerequisite before reading this manual. IBM provides the *IBM Client Security Software Version 5.41 Administrator and User Guide* and application helps which you can consult for information about using the application.

Product Publications

The following documents are available in the Client Security Software Version 5.41 library:

- *Client Security Software Version 5.41 Administrator and User Guide*,

  Provides information on setting up and using the security features provided with Client Security Software, and contains information about performing Client Security Software tasks, such as using UVM logon protection, setting up the Client Security screen saver, creating a digital certificate, and using the User Configuration Utility.
- *Client Security Software Version 5.41 Installation Guide*,

  Contains information about installing Client Security Software on IBM network computers which contain IBM embedded security chips.

Additional Information

You can obtain additional information and security product updates, when available, from the http://www.pc.ibm.com/us/security/ IBM Web site.

Contents

Preface iii
Audience iii
Product Publications iii
Additional Information iii

Chapter 1. Considerations before deploying IBM Client Security Software . 1
Requirements and specifications for deployment . . 1

Chapter 2. Installing Client Security Software 3
Standard installation. 3
Administrative installation. 3
Command line parameters. 4
Client Security Software custom public properties . . 5
Client Security Software installation features. . . . 6
Examples using Setup.exe 6

Chapter 3. How the Embedded Security Chip functions 9
Key-swapping hierarchy 11
Why key swapping? 12

Chapter 4. Key archiving considerations 13
Why an administrator key pair? 16

Chapter 5. IBM Client Security Software 27
Enrolling users and managing enrollment 27
Requiring a passphrase 28
Setting up a passphrase 28
Using a passphrase 29
TPM Initialization 32
Best Practices 33
User initialization 34
Personal Initialization 35
Deployment scenarios 36
Configuration file details 41

Chapter 6. Installing the Client Security component on a Tivoli Access Manager server 47
Prerequisites 47
Downloading and installing the Client Security component 47
Adding the Client Security components on the Tivoli Access Manager server 48
Establishing a secure connection between the IBM client and the Tivoli Access Manager server . . . 48
Configuring IBM clients 50
Prerequisites 50
Configuring the Tivoli Access Manager setup information 50
Setting and using the local-cache feature. . . . 51
Enabling Tivoli Access Manager to control IBM client objects 51
Troubleshooting charts. 53
Digital certificate troubleshooting information . . 53
Tivoli Access Manager troubleshooting information 53
Lotus Notes troubleshooting information . . . 54
Encryption troubleshooting information 54

Chapter 7. Installing third-party hardware device drivers to complement IBM Client Security Software 55

Chapter 8. Remotely deploying new or revised security policy files. 57

Appendix. Notices. 59
Non-IBM Web sites 59
Trademarks 60

Chapter 1. Considerations before deploying IBM Client Security Software

Central deployment of IBM Client Security Software Version 5.41 is accomplished through the Advanced Configuration Mode in the IBM Client Security Software setup wizard. IBM Client Security Software Version 5.4 does not support first generation (non-TCPA) security chips. Users of these systems must use Client Security Software Version 5.3.

There are various ways to deploy IBM Client Security Software (CSS), which uses the IBM Embedded Security Subsystem (ESS) hardware that is integrated into IBM personal computers. This document will help you determine how to deploy the ESS in your environment. It is important to look at the process of how your company deploys computers from image creation to the way the PC is given to an end user. This process will greatly influence how your company deploys ESS. The IBM ESS is composed of essentially two parts as shown in Figure 1:

1. Client Security Software
2. Embedded Security Chip

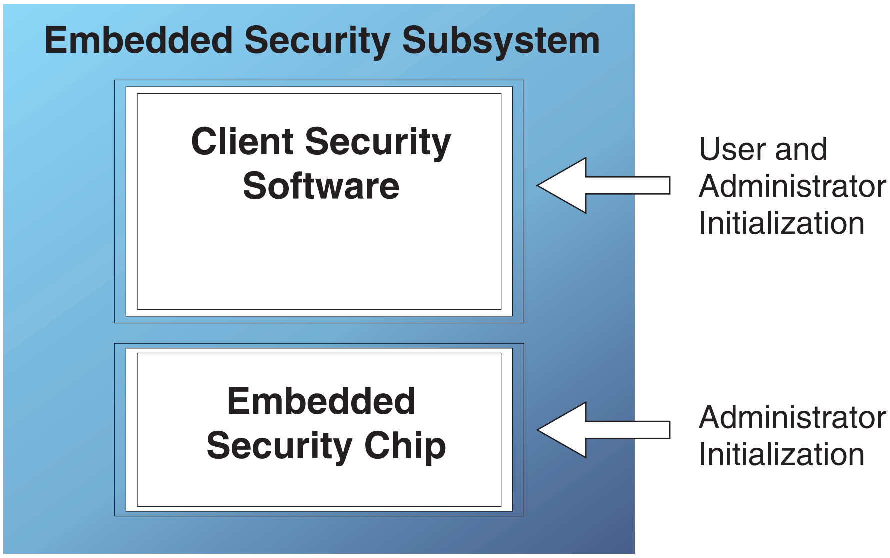
Embedded Security Subsystem
Client Security Software
Embedded Security Chip
User and Administrator Initialization
Administrator Initialization

Figure 1. IBM Embedded Security Subsystem components

Requirements and specifications for deployment

If you plan to install IBM Client Security Software on computers that are equipped with the Embedded Security Chip, plan on the following server storage and download requirements and installation times:

1. IBM PC with Embedded Security Chip
2. Server Storage requirement for installable code: approximately 12 MB
3. Average per-user server storage requirement for key archive data: 200 KB per user for archive storage

Chapter 2. Installing Client Security Software

This chapter describes two different ways to install Client Security Software, the standard installation, and the administrative installation.

Standard installation

The z046zis2018usaa.exe file is a self-extracting installation package that extracts the installation source files and launches the installation. This file accepts a set of command line parameters, which are described as follows. Command-line options that require a parameter must be specified with no space between the option and its parameter. For example, `z046zis2018usaa.exe /s /v"/qn REBOOT="R""` is valid, while `Setup.exe /s /v "/qn REBOOT="R""` is not (the **"/qn REBOOT="R""** is a parameter to the option **/v**. Quotation marks around an option's parameter are required only if the parameter contains spaces.

The default behavior of the installation when you are just running Setup.exe without any parameters, which runs the installation with a user interface, is to prompt for a reboot at the end of the installation. The default behavior when running the installation with no user interface is to do a reboot at the end of the installation. However, the reboot can be delayed with the REBOOT property as documented previously and in the example section.

/a This parameter causes the executable file to perform an administrative installation. An administrative installation copies your data files to a directory specified by the user, but does not create shortcuts, register COM servers, or create an uninstallation log.

/x This parameter causes the executable file to uninstall a previously installed product.

/s This parameter causes the executable file to run in silent mode.

/v This parameter is used to pass command line switches and values of public properties through to Msiexec.exe.

/w This parameter forces the executable file to wait until the installation is complete before exiting. If you are using this parameter in a batch file, you may want to precede the entire executable file command line argument with `start /WAIT`. A properly formatted example of this usage is as follows:

```
start /WAIT z046zis2018usaa.exe /w
```

Administrative installation

The Microsoft® Windows® Installer can perform an administrative installation of an application or product to a network for use by a workgroup or for customization. For the Client Security Software installation package, an administrative installation unpacks the installation source files to a specified location. To run an administrative installation the setup package needs to be executed from the command line using the **/a** parameter:

```
z046zis2018usaa.exe /a
```

A new location can be chosen which may include drives other than C:, such as other local drives and mapped network drives. New directories can also be created during this step.

If an administrative installation is run silently, the public property TARGETDIR can be set on the command line to specify the extract location:

```
Setup.exe /s /v"/qn TARGETDIR=F:\IBMCSS"
```

or

```
msiexec.exe /i "IBM Client Security Software.msi" /qn TARGERDIR=F:\IBMCSS
```

To install from the unpacked source after customizations are made, the user calls msiexec.exe from the command line. "Command line parameters" describes the available command line parameters that can be used with msiexec.exe as well as an example of how to use it. Public properties can also be set directly in the msiexec command line call.

Command line parameters

/i *package* **or** *product code*
Use this format to install the product:

```
msiexec /i "C:\WindowsFfolder\Profiles\UserName\Personal\MySetups\Othello\Trial
Version\Release\DiskImages\Disk1\productOthello Beta.msi"
```

Product code refers to the GUID that is automatically generated in the product code property of your product's project view.

Note: The previous example has been split into two lines in order to fit on the page. When entering this command, type it on one line.

/a *package*
The **/a** parameter allows users with administrator privileges to install a product onto the network.

/x *package* **or** *product code*
This parameter uninstalls a product.

/L [i|w|e|a|r|u|c|m|p|v|+] *logfile*
This parameter specifies the path to the log file. The following flags indicate which information to record in the log file:

- **i**

  Logs status messages
- **w**

  Logs non-fatal warning messages
- **e**

  Logs any error messages
- **a**

  Logs the commencement of action sequences
- **r**

  Logs action-specific records
- **u**

  Logs user requests
- **c**

  Logs initial user interface parameters
- **m**

  Logs out-of-memory messages
- **p**

Logs terminal settings

- **v**

  Logs the verbose output setting
- **+**

  Appends to an existing file
- *****

  Is a wildcard character that allows you to log all information except the verbose output setting

/? or /h

Either command displays Windows Installer copyright information

TRANSFORMS

Use the TRANSFORMS command line parameter to specify any transforms that you would like applied to your base package. Your transform command line call might look something like this:

```
msiexec /i "C:\WindowsFolder\Profiles\UserName\Personal\MySetups\Project Name\
Trial Version\My Release-1\DiskImages\Disk1\ProductName.msi"
TRANSFORMS="New Transform 1.mst"
```

Because you can separate multiple transforms with a semicolon, it is recommended that you do not use semicolons in the name of your transform, as the Windows Installer service will not interpret those correctly.

Note: The previous example has been split into three lines in order to fit on the page. When entering this command, type it on one line.

Properties

All public properties can be set or modified from the command line. Public properties are distinguished from private properties by the fact that they are in all capital letters. For example, COMPANYNAME is a public property.

To set a property from the command line, use the following syntax: `PROPERTY=VALUE`. If you wanted to change the value of *COMPANYNAME*, you would enter:

```
msiexec /i "C:\WindowsFolder\Profiles\UserName\Personal\MySetups\Project Name\
Trial Version\My Release-1\DiskImages\Disk1\ProductName.msi"
COMPANYNAME="InstallShield"
```

Note: The previous example has been split into three lines in order to fit on the page. When entering this command, type it on one line.

Client Security Software custom public properties

The installation package for Client Security Software contains a set of custom public properties that can be set on the command line when running the installation. The currently available custom public properties are:

INSTALLPWM

This property is used to control whether Password Manager gets installed during initial installation. Set to 1 to install Password Manager, set to 0 to not install Password Manager. Default value is 1.

CFGFILE

This property can be used during a silent installation to specify the location of a configuration file. The configuration file can contain the value of the existing password for the security chip. This allows the installation to complete with no user interaction even if there is already a password on the chip. For example:

```
CFGFILE=C:\csec.ini
```

Client Security Software installation features

The Client Security Software One-Click Installation contains two main features, *Security* (IBM Client Security Software) and *PWManager* (IBM Password Manager). By default both features are installed, however you have several options to run the installation so that only the Security feature will be installed (the Security feature is required, the PWManager feature is not required). If the user runs the installation with a user interface and IBM Password Manager Version 1.3 or lower is not already installed, they will be presented with a screen to choose whether to install IBM Client Security Software only or both IBM Client Security Software and IBM Password Manager. If the user is running the installation without a user interface (silent), they can control whether Password Manager gets installed by using the INSTALLPWM property (set to 0 to not install Password Manager). If the user chooses to install IBM Client Security only during the initial installation and later decides to add IBM Password Manager, they can do so by running the original source package again. If running the installation again with a user interface, they will be presented with a maintenance screen where they can choose the “Modify” button if Password Manager has not been installed yet. This will bring them to the screen where they can choose to reinstall Client Security Only or change their choice to install both IBM Client Security Software and IBM Password Manager. The user can also reinstall the product from the source with no user interface to add IBM Password Manager. Example commands to do this are listed as follows.

Examples using Setup.exe

Table 1 shows installation examples using z046zis2018usaa.exe.

Table 1. Installation examples using z046zis2018usaa.exe

Type	Example
Silent installation with reboot and end of installation	`z046zis2018usaa.exe /s /v/qn`
Silent installation with no reboot	`z046zis2018usaa.exe /s /v"/qn REBOOT="R""`
Silent installation with no reboot and Password Manager not installed	`z046zis2018usaa.exe /s /v"/qn REBOOT="R" INSTALLPWM=0"`
Silent installation with no reboot and specify Installation directory	`z046zis2018usaa.exe /s /v"/qn REBOOT="R" INSTALLDIR=C:\ibmcss"`
Silent installation with no reboot and specify configuration file	`z046zis2018usaa.exe /s /v"/qn REBOOT="R" CFGFILE=C:\csec.ini"`
Silent administrative installation	`z046zis2018usaa.exe /a`
Silent administrative installation specifying the extract location	`z046zis2018usaa.exe /a /s /v"/qn TARGETDIR="F:\CSS""`
Installation with no reboot and create an installation log in temp directory	`z046zis2018usaa.exe /v"REBOOT="R" /L*v %temp%\css.log"`
Silent reinstallation of product to add Password Manager	`z046zis2018usaa.exe /s /v"/qn ADDLOCAL=PWManager"`

Table 2 on page 7 shows installation examples using msiexec.exe.

Table 2. Installation using msiexec.exe

Type	**Example**
Installation with log file	`msiexec /i "C:\IBM Client Security Software.msi" /L*v %temp%\css.log`
Silent installation with no reboot	`msiexec /i "C:\IBM Client Security Software.msi" /qn REBOOT="R"`
Silent installation with no reboot and Password Manager not installed	`msiexec /i "C:\IBM Client Security Software.msi" /qn REBOOT="R" INSTALLPWM=0`
Silent reinstallation of product to add Password Manager	`msiexec /i "C:\IBM Client Security Software.msi" /qn ADDLOCAL=PWManager`

Chapter 3. How the Embedded Security Chip functions

The IBM Embedded Security Chip is represented graphically in Figure 2. There are three major components:

1. Administrator password
2. Hardware public key
3. Hardware private key

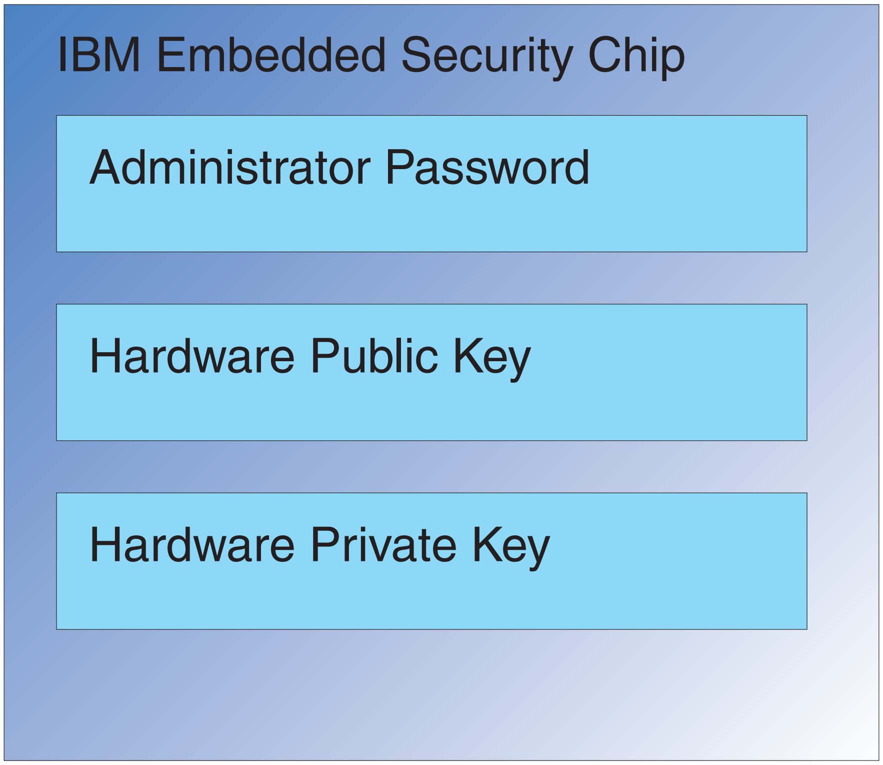

Figure 2. Data held in the IBM Embedded Security Chip

The hardware public and private keys are unique on every computer. The hardware private key can never be extracted from the chip. New key pairs can be generated in the following ways:

- Through the Client Security Software wizard
- Through the Administrator Utility
- Using scripts

The administrator uses the administrator password to access the following functions, including:

- Adding users
- Setting security policy
- Setting passphrase policy
- Enrolling smartcards
- Enrolling biometric devices

For example, an administrator might need to enable an additional user to take advantage of the Embedded Security Chip features and functions. The administrator password is set when the Client Security Software is installed. Details regarding how and when the administrator passwords are set are covered later in this document.

Important: Develop a strategy for maintaining administrator passwords, which must be established when first configuring ESS. It is possible for each computer with an Embedded Security Chip to have the same administrator password, if the IT administrator or security administrator so determines. Alternatively, each department or building can be assigned different administrator passwords.

The other components of the IBM Embedded Security Chip are the hardware public key and hardware private key. This RSA key pair is generated when the Client Security Software is configured.

Each computer will have a unique hardware public key and a unique private key. Random number capability on the IBM Embedded Security Chip ensures that each hardware key pair is statistically unique.

Figure 3 on page 11 describes two additional components of the IBM Embedded Security Chip. Understanding these two components is critical for effectively managing your IBM Embedded Security Subsystem infrastructure. Figure 3 on page 11 shows the administrator public and private keys as well as user public and private keys. The following is a summary of public and private keys.

- Public and private keys are considered a ″key pair.″
- The public and private keys are mathematically related such that:
  - Anything encrypted with the public key can only be decrypted with the private key.
  - Anything encrypted with the private key can only be decrypted with the public key.
  - Knowing the private key does not enable you to derive the public key.
  - Knowing the public key does not enable you to derive the private key.
  - The public key is generally made available to everyone.
- The private key must be aggressively protected.
- Public and private keys are the basis for public key infrastructure (PKI).

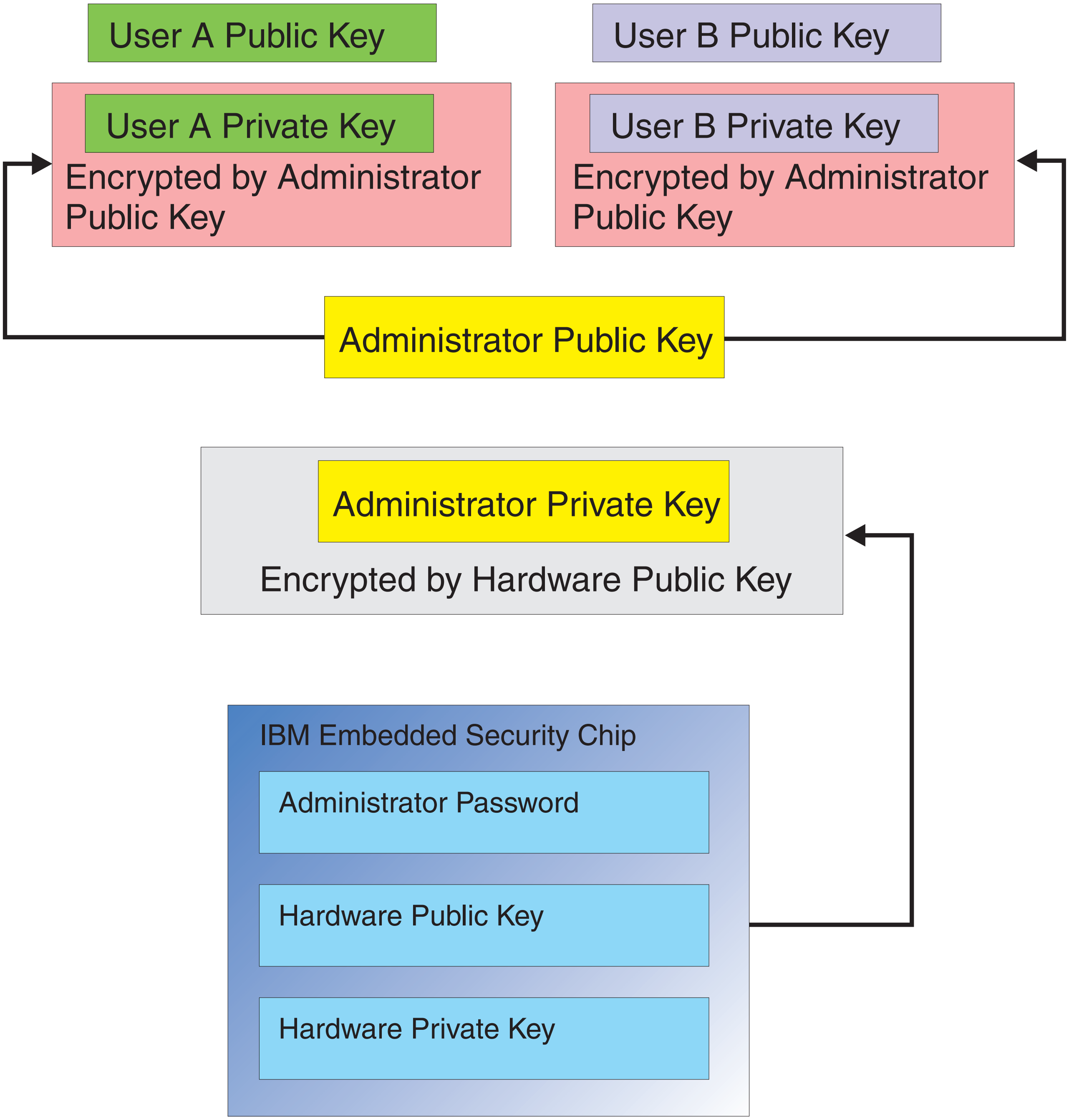
User A Public Key
User B Public Key
User A Private Key
Encrypted by Administrator Public Key
User B Private Key
Encrypted by Administrator Public Key
Administrator Public Key
Administrator Private Key
Encrypted by Hardware Public Key
IBM Embedded Security Chip
Administrator Password
Hardware Public Key
Hardware Private Key

Figure 3. Several layers of encryption provide strong security

Key-swapping hierarchy

Part of the IBM ESS architecture is a ″key-swapping″ hierarchy. The details of precisely how this works will be covered in the *IBM Client Security Software Administrator and User Guide*; however, this section introduces the concept as it applies to mass configuration, deployment, and management. In Figure 3, you can see the Hardware public and Hardware private key. As mentioned previously these keys are created by the Client Security Software and are statistically unique on each client. In Figure 2, the IBM Embedded Security Chip you can see the

Administrator public and private key pair. The Administrator public and private key pair can be unique on all computers or they can be the same on all clients or a subset of clients. The advantages and disadvantages will be discussed later in this document. The Administrator public and private keys perform the following:

- Protect user public and private keys
- Enable archiving and restoration of user credentials
- Enable user credential roaming, which is described in the *IBM Client Security Software Administrator and User Guide*

Why key swapping?

In the following sections you will read about users in the IBM ESS environment. The details of how to set up IBM Client Security Software and ESS to accommodate these users will be covered in those sections. Each user has a public and private key. The user's private key is encrypted with the Administrator public key. From Figure 3 on page 11, you can see that the Administrator private key is encrypted with the hardware public key. Why are these various private keys encrypted?

The reason goes back to the hierarchy mentioned earlier. Due to limited storage space in the IBM Embedded Security Chip, only a limited number of keys can be in the chip at any given time. The Hardware public and private keys are the only persistent (from boot to boot) keys in this scenario. In order to enable multiple keys and multiple users, IBM ESS implements a key swapping hierarchy. Whenever a key is needed it is ″swapped″ into the IBM Embedded Security Chip. By swapping the encrypted private keys into the chip, the private key can be decrypted and used only in the protected environment of the chip.

The Administrator private key is encrypted with the Hardware public key. The Hardware private key, which is only available in the chip, is used to decrypt the Administrator private key. After the Administrator private key is decrypted in the chip, a user's private key (encrypted with the Administrator public key) can be passed into the chip from the hard disk and decrypted with the Administrator private key. From Figure 3 on page 11, you can see that you can have multiple users' private keys encrypted with the Administrator public key. This provides the ability to set up as many users as necessary on a computer with the IBM ESS.

Chapter 4. Key archiving considerations

Passwords and keys work together, along with other optional authentication devices, to verify the identity of system users.

Figure 4 shows how the IBM Embedded Security Subsystem and Client Security Software work together. The Windows logon prompts User A to log on and User A does so. The IBM Client Security System determines who the current user is through information provided by the operating system. The Administrator private key, which is encrypted with the Hardware public key, is loaded into the Embedded Security chip.

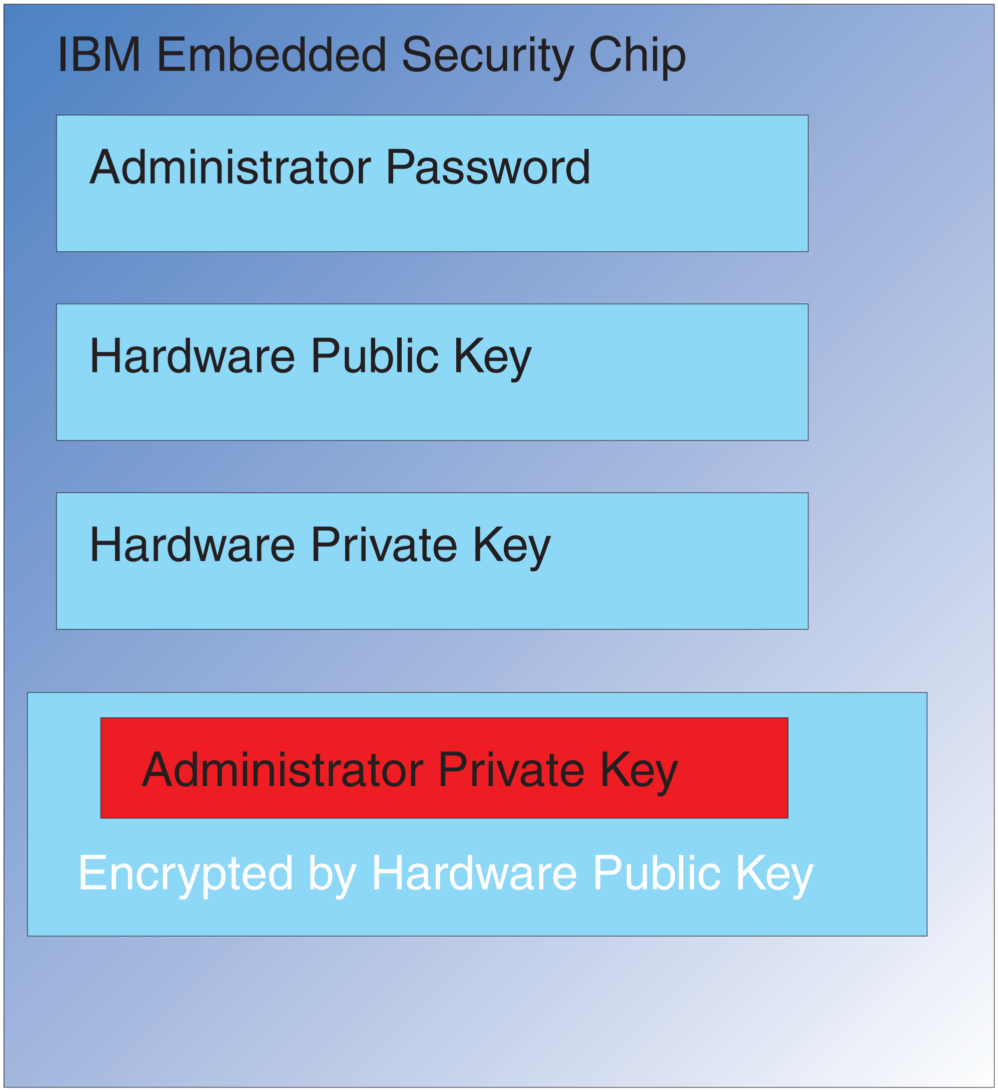
IBM Embedded Security Chip
Administrator Password
Hardware Public Key
Hardware Private Key
Administrator Private Key
Encrypted by Hardware Public Key

Figure 4. The Administrator private key, which is encrypted by the hardware public key, is loaded into the Embedded Security chip.

The Hardware private key (which is only available in the chip) decrypts the Administrator private key. Now the Administrator private key is available for use in the chip as shown in Figure 5.

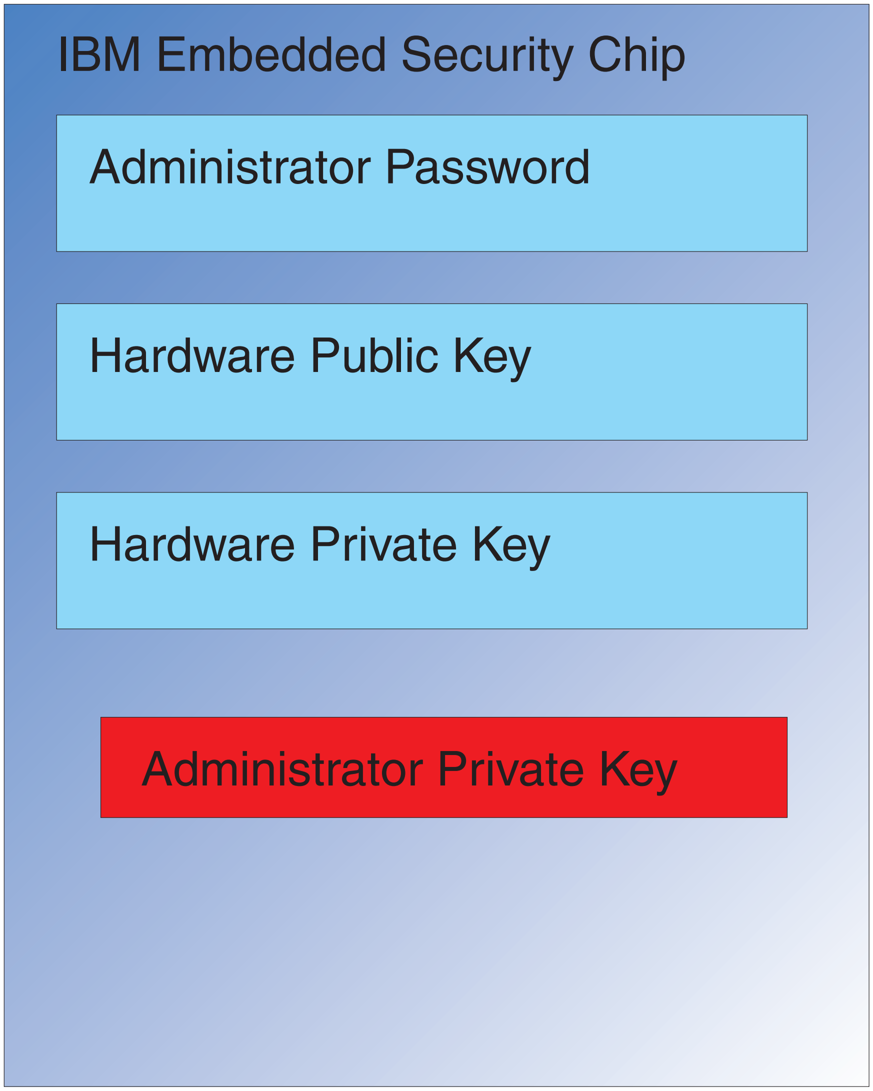

Figure 5. The Administrator private key is available for use in the security chip.

Because User A is logged onto the computer, User A's private key (encrypted with the Administrator public key) is passed into the chip as shown in Figure 6 on page 15.

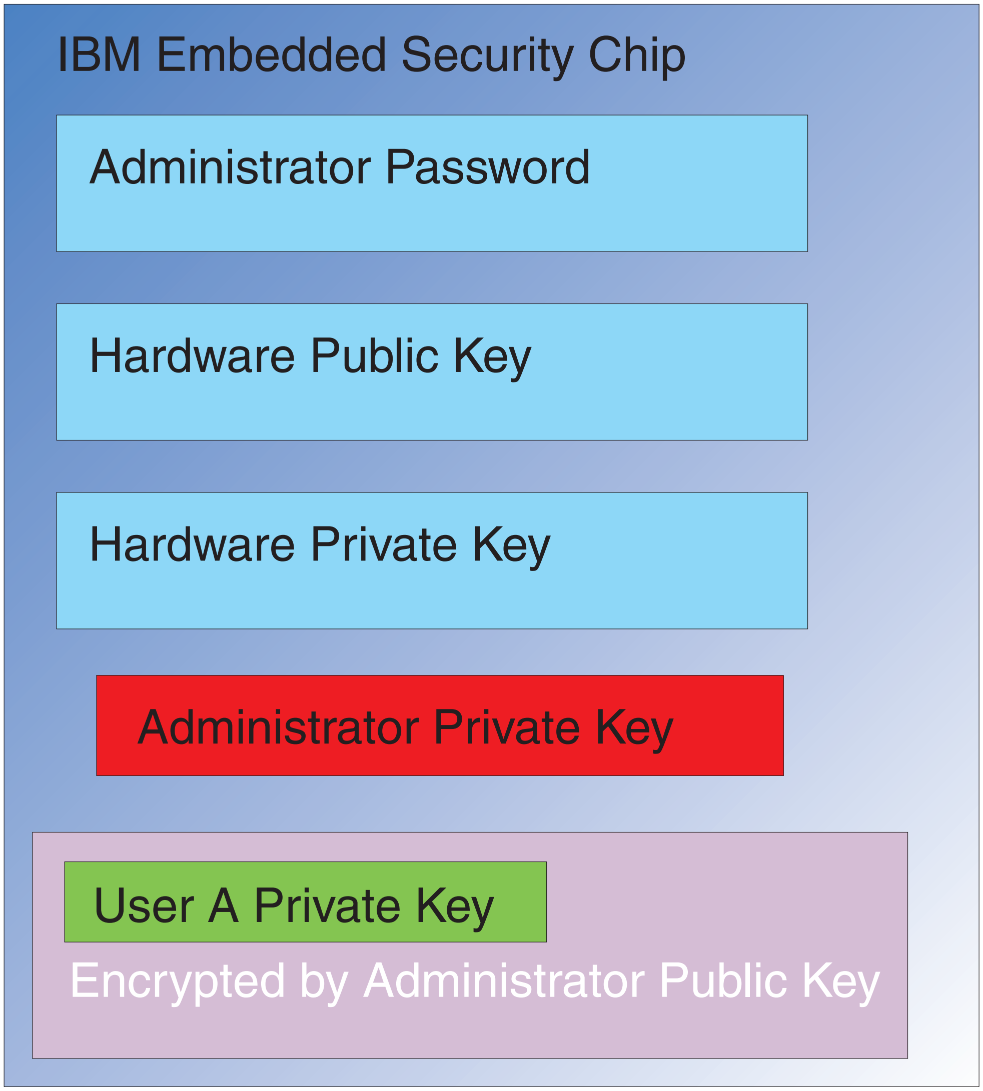
IBM Embedded Security Chip
Administrator Password
Hardware Public Key
Hardware Private Key
Administrator Private Key
User A Private Key
Encrypted by Administrator Public Key

Figure 6. User A's private key, which is encrypted by the Administrator public key, is passed into the security chip.

The Administrator private key is used to decrypt the User A's private key. Now User A's private key is ready for use as shown in Figure 7 on page 16.

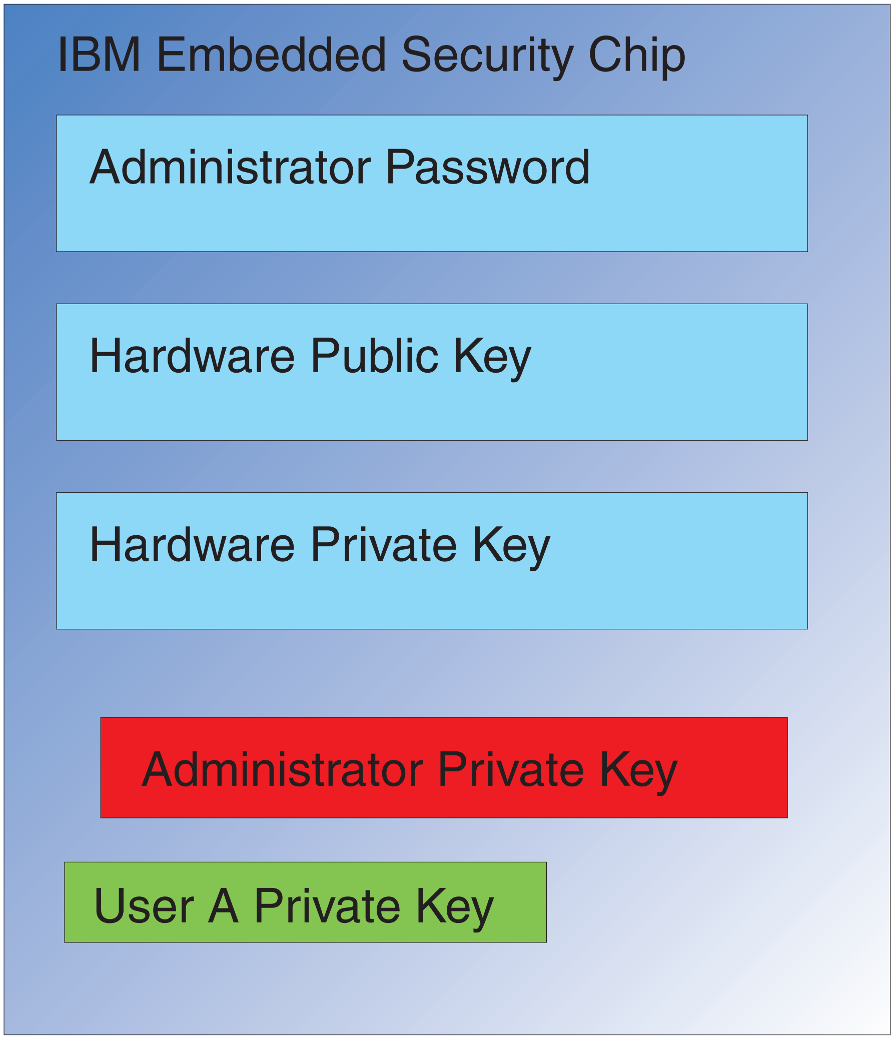

Figure 7. User A’s private key is ready for use.

There are several other keys that can be encrypted with the User A’s public key. An example would be a private key used for signing e-mail. When User A goes to send a signed e-mail the private key used for signing (encrypted with User A’s public key) would be passed into the chip. User A’s private key (already in the chip) would decrypt User A’s private signing key. Now User A’s private signing key is available in the chip to perform the desired operation, in this case creating a digital signature (encrypting a hash).

Note: The same process of moving keys in to and out of the chip would be used when User B logs onto the computer.

Why an administrator key pair?

The main reason to have an administrator key pair is for archive and restore capabilities. The Administrator key pair serves as an abstraction layer between the chip and the user credentials. The user-specific private key information is encrypted with the Administrator public key as shown in Figure 8 on page 17.

Important: Develop a strategy for maintaining administrator key pairs. It is possible for each computer with an Embedded Security Chip to have the same administrator key pair, if the IT administrator or security administrator so determines. Alternatively, each department or building can be assigned different administrator key pairs.

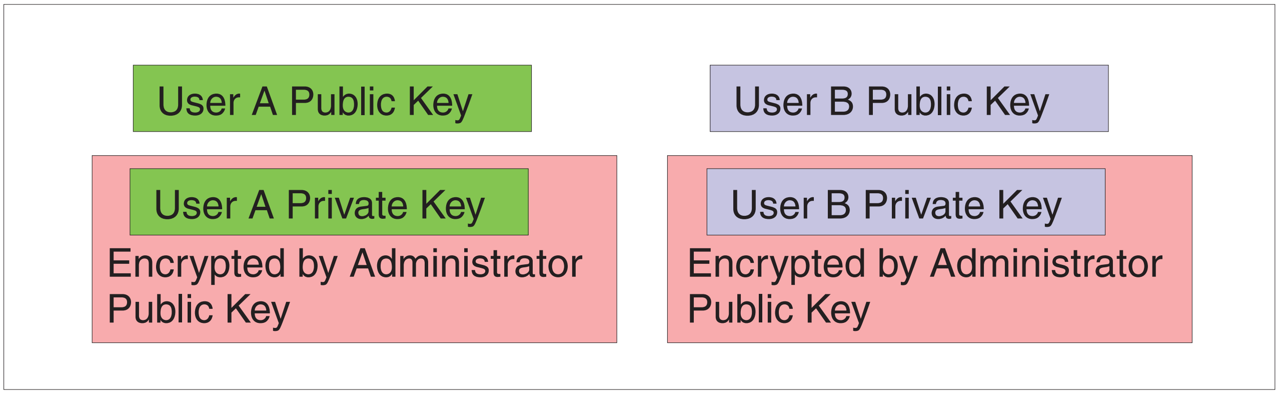
User A Public Key
User A Private Key
Encrypted by Administrator Public Key
User B Public Key
User B Private Key
Encrypted by Administrator Public Key

Figure 8. The user-specific private key information is encrypted with the Administrator public key.

Another reason to have an administrator key pair is to sign the client security policy file, thereby preventing anyone except the administrator from changing security policy. In order to achieve a high degree of security for client security policy file, you can split the administrator private key among up to five individuals. In such a case, the five individuals who hold part of the private key, must all be present to sign and encrypt files, such as the client security policy file. This prevents a single individual from unilaterally performing administrator functions. For information about splitting the administrator private key, see the Keysplit=1 setting in Table 6 on page 41.

During IBM Client Security Software initialization, administrator key pairs can either be created by the software or can be imported from an external file. If you want to use a common administrator key pair, you will specify the location of the necessary files during client installation.

This user specific information is backed up (written) to an administrator defined archive location as shown in Figure 8. This archive location can be any type of media that is physically or logically connected to the client. The IBM Client Security System installation section will discuss best practices for this archive location.

The Administrator public and private keys are not archived. The user data in the archive location is encrypted with the Administrator public key. Having the user archive data by itself does you no good if you do not have the Administrator private key to unlock the data. The Administrator public and private key are often referred to in IBM Client Security Software documentation as the ″Archive Key Pair.″

Note: Archive Private Key is not encrypted. Special care must be taken in storing and protecting the Archive Key Pair.

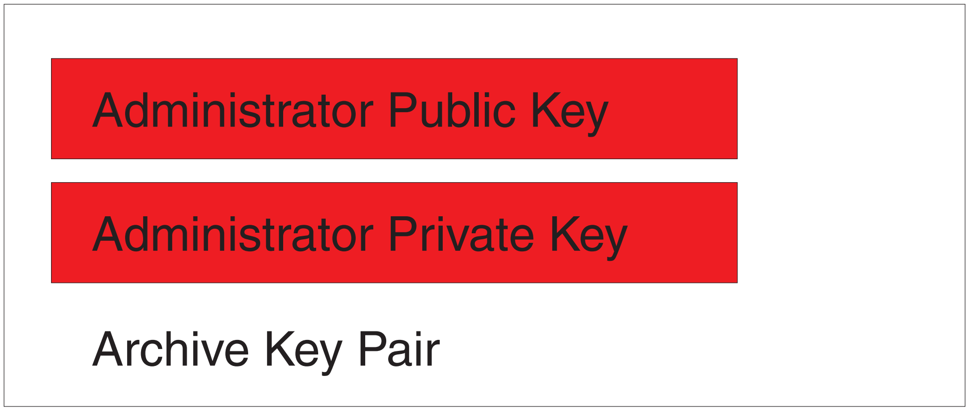

Figure 9. The Administrator public and private key make up the Archive Key Pair

As mentioned previously, one of the most important functions of the Administrator public and private keys is for backing up and restoring disk contents. This capability is shown in 10 through 15. The steps are as follows:

1. Client A, for some reason, becomes unusable to User A. In this example, the computer, Client A, is struck by lightning as shown in Figure 10 on page 19.

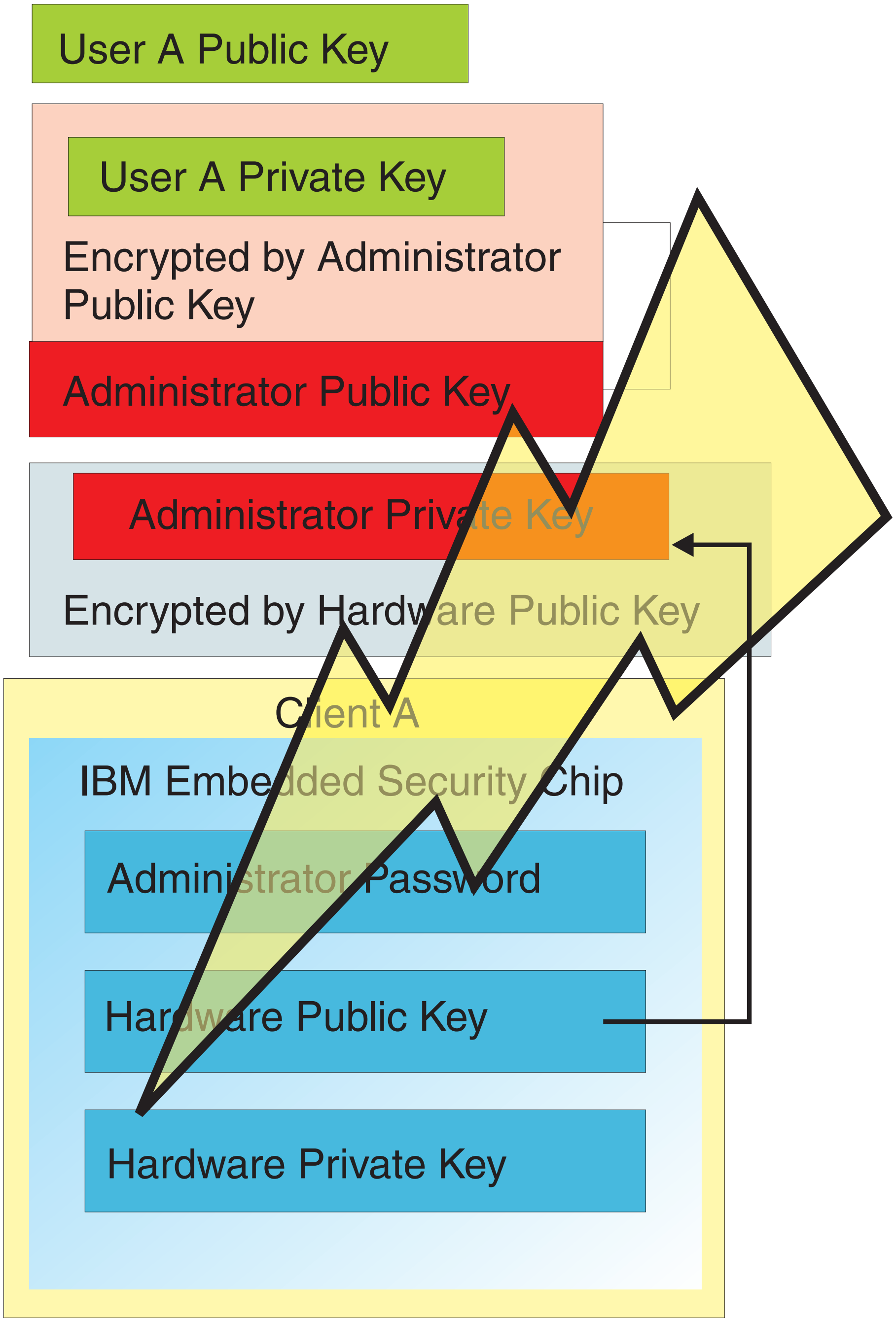

Figure 10. User A's computer is struck by lightning, making it unusable.

2. User A gets a new and improved IBM computer, called Client B as shown in Figure 11 on page 20. Client B is different from Client A in that the Hardware public and private keys are different from those of Client A. This difference is visually represented by the gray color keys in Client B and the green color keys in Client A. However, note the Administrator Password is the same in Client B as in Client A.

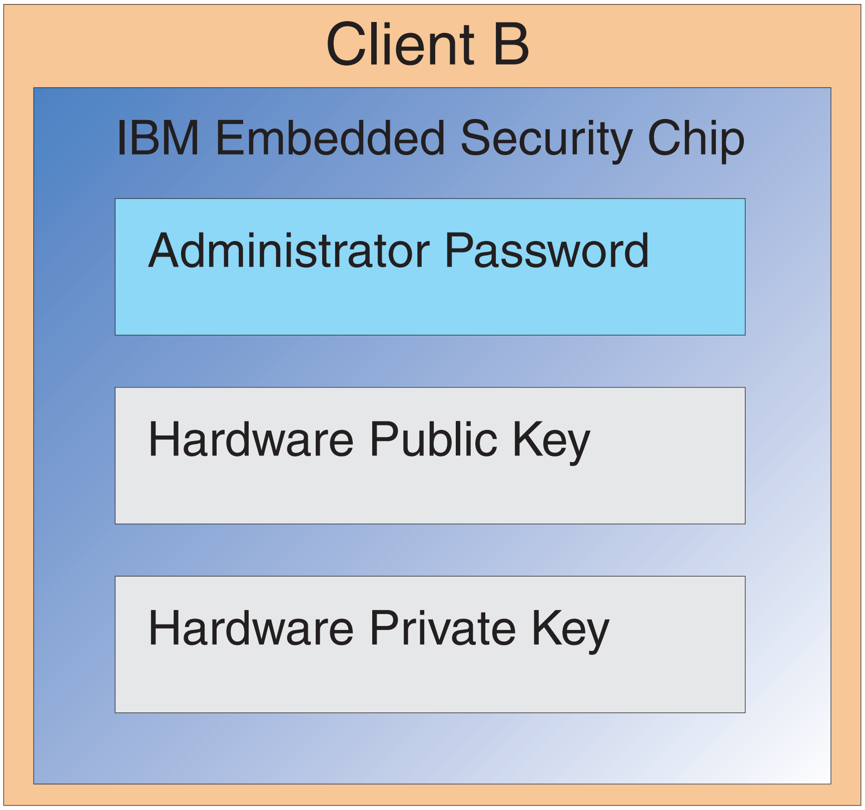

Figure 11. User A receives a new computer, Client B, with a new Embedded Security chip.

3. Client B now needs the same user credentials that were on Client A. This information was archived from Client A. If you look back at Figure 8 on page 17, you will recall that the user keys are encrypted with the Administrator Public Key and stored in the archive location. In order for the user's credentials to be available on Client B, the Administrator public and private keys must be transferred to this machine. Figure 12 shows Client B retrieving the Administrator public and private keys in order to recover user data from the archive location.

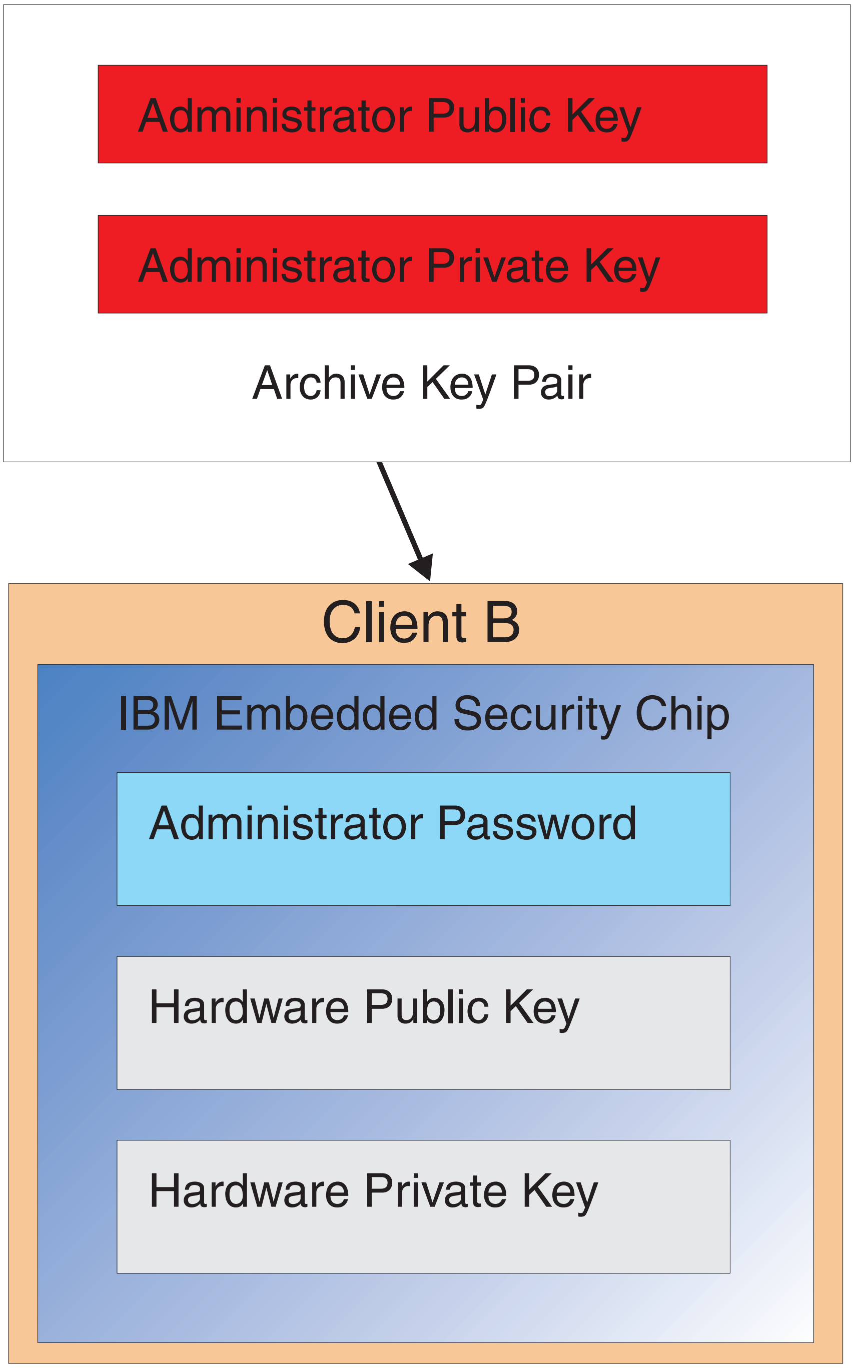

Figure 12. Client B retrieves the Administrator public and private keys from the archive location.

4. Figure 13 on page 22 shows the Administrator private key being encrypted with Hardware public key of Client B.

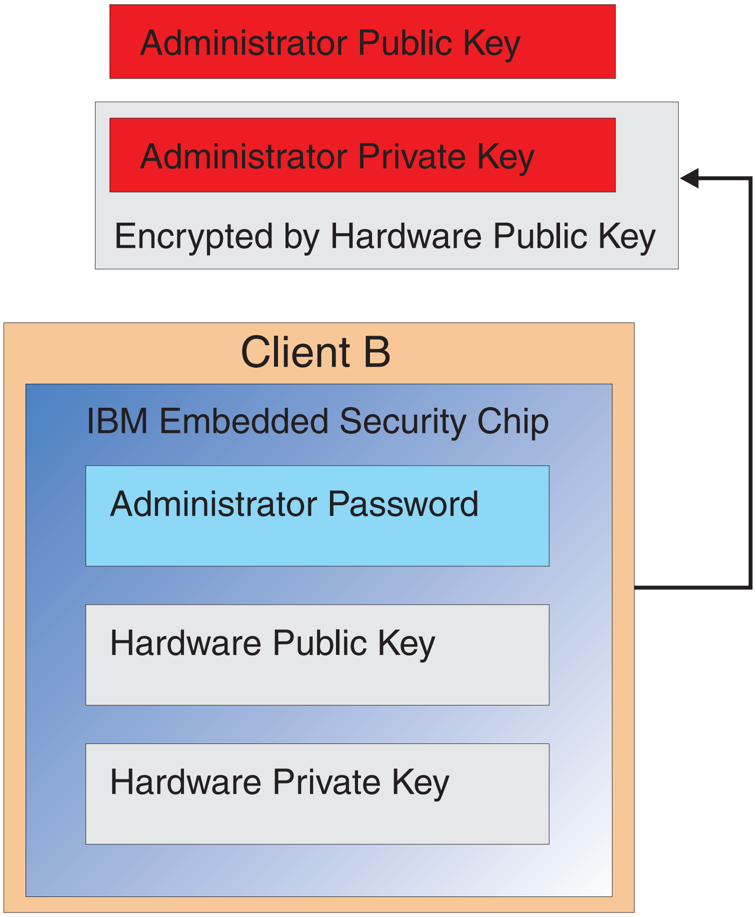

Figure 13. The Administrator private key is encrypted with the Client B hardware key.

Now that the Administrator private key is encrypted with the Hardware public key, the user's credentials can be brought down for User A on Client B as shown in Figure 14 on page 23.

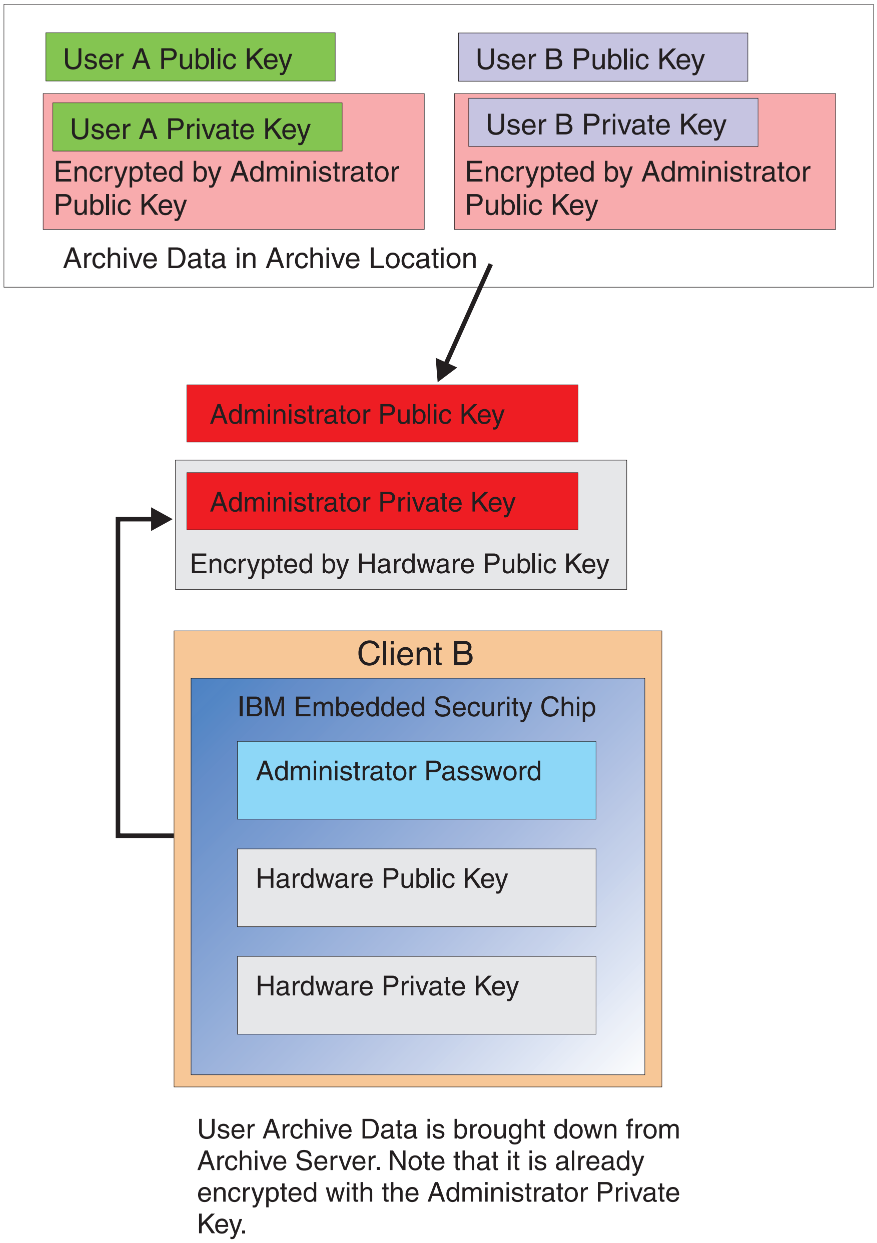
User A Public Key
User B Public Key
User A Private Key
Encrypted by Administrator Public Key
User B Private Key
Encrypted by Administrator Public Key
Archive Data in Archive Location
Administrator Public Key
Administrator Private Key
Encrypted by Hardware Public Key
Client B
IBM Embedded Security Chip
Administrator Password
Hardware Public Key
Hardware Private Key
User Archive Data is brought down from Archive Server. Note that it is already encrypted with the Administrator Private Key.

Figure 14. User A's credentials can be loaded on Client B after the Administrator private key has been encrypted.

Figure 15 on page 24 shows User A fully restored on Client B. User A's private key was encrypted with the Administrator public key while on the archive server. The Administrator public key is a 2048-bit RSA key and is virtually impossible to break. This means the archive location does not necessarily have to be protected or have strong Access Control. As long as the Archive key pair (the Administrator public and private keys) and more specifically the Administrator private key are kept secure the Archive location for user credentials can be essentially anywhere.

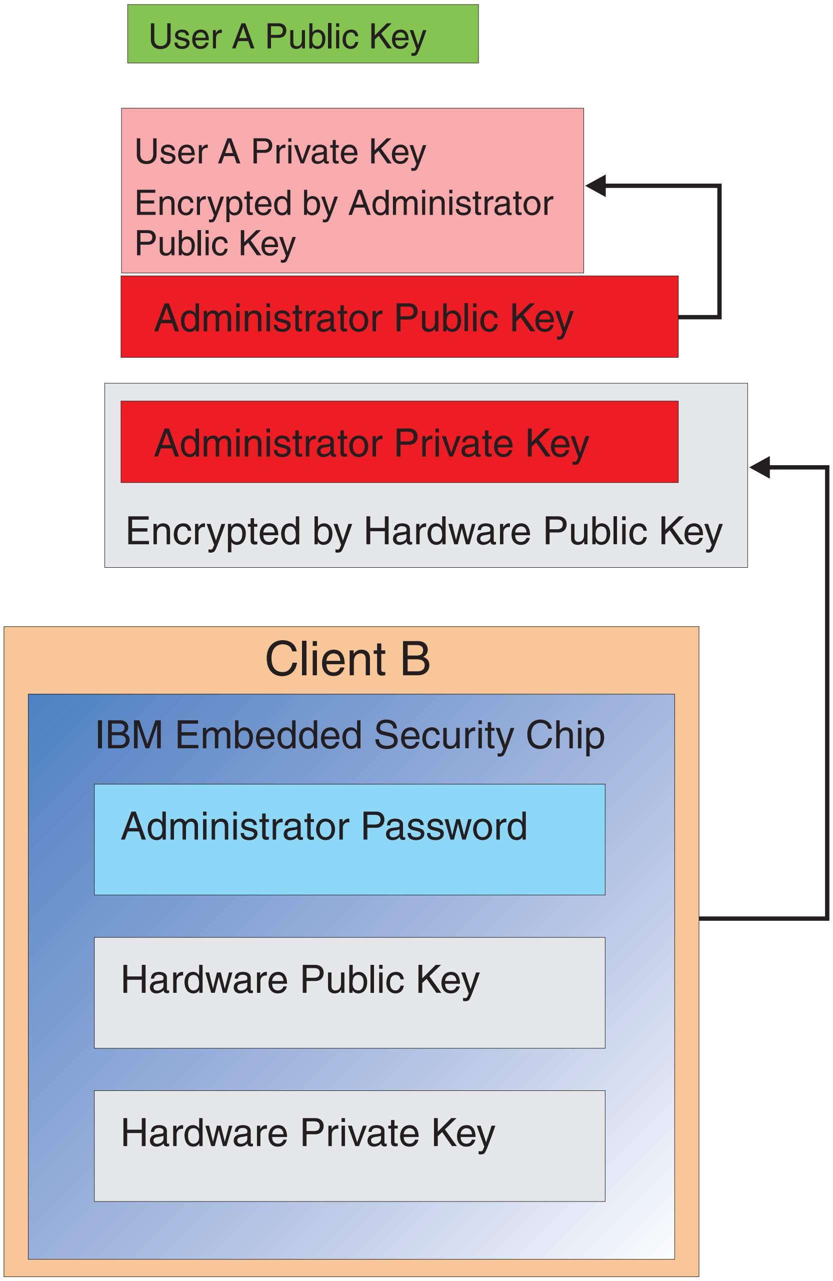

Figure 15. User A is fully restored on Client B.

The details (for example, of how the Administrator Password is set and where Archive locations should be) will be discussed in the software installation section of this document. Figure 16 shows an overview of the components in an ESS environment. The major points are that each client is unique from a hardware public and private key perspective, but has a common Administrator public and private Key. The Clients have a common archive location but this archive location could be for a segment or group of users.

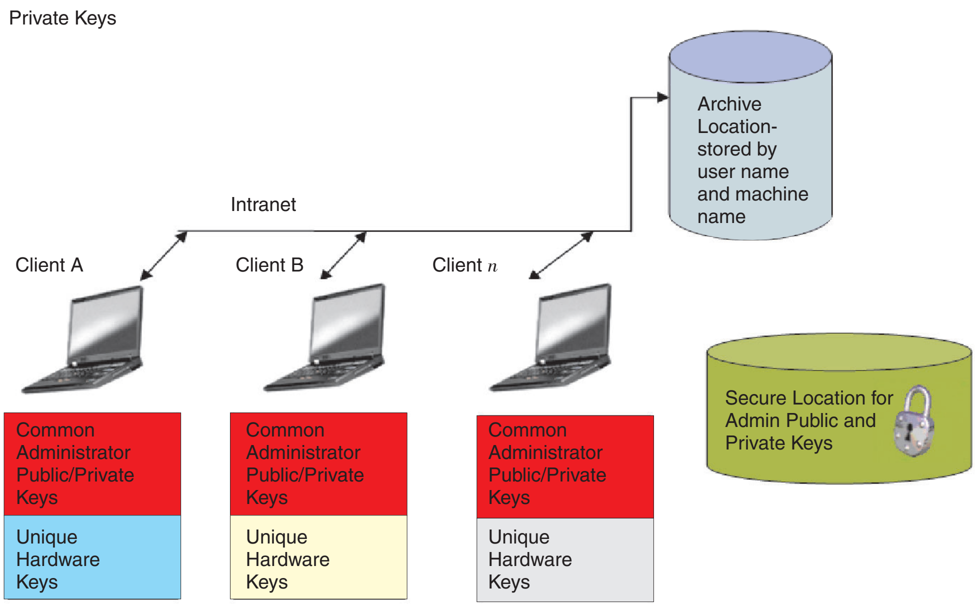
Private Keys
Archive Location- stored by user name and machine name
Intranet
Client A
Client B
Client *n*
Secure Location for Admin Public and Private Keys
Common Administrator Public/Private Keys
Unique Hardware Keys
Common Administrator Public/Private Keys
Unique Hardware Keys
Common Administrator Public/Private Keys
Unique Hardware Keys

Figure 16. Major components of the IBM Client Security System.

Consider the following example. The Human Resources Department could have a separate archive location from the Engineering department. Archiving is done on a user-name and computer name basis. The IBM Client Security Software will archive the users of a system to the defined archive location based on the user name and computer name as shown earlier in User A and User B. Also note the secure location for the Administrator public and private Keys.

Note: Each computer name and user name that will be archived in the same location must be unique. A duplicate computer name or user name will overwrite the previous archive of the same name.

Chapter 5. IBM Client Security Software

The IBM Client Security Software is the connection between applications and the IBM Embedded Security chip, as well as the interface to enroll users, set policy, and perform basic administration functions. The IBM Client Security System is essentially composed of the following components:

- Administrator Utility
- User Configuration Utility
- Administrator Console
- Installation Wizard
- User Verification Manager (UVM)
- Cryptographic Service Provider
- PKCS#11 module

The IBM Client Security System enables you to do several key functions:

- Enroll users
- Set Policy
- Set Passphrase Policy
- Reset forgotten passphrases
- Restore user credentials

For example, if User A logs onto the operating system, IBM Client Security System bases all decisions on the assumption that User A is logged on.

Note: Security Policy is machine based, not user based; the policy applies to all users of a single computer.

If User A attempts to leverage the IBM Embedded Security Subsystem, the IBM Client Security System will enforce security policies as set for User A on that computer, such as passphrase or fingerprint authentication. If the person logged on as User A cannot supply the correct passphrase or the correct fingerprint for authentication, IBM ESS will prohibit the user from performing the requested action.

Enrolling users and managing enrollment

IBM ESS users are simply Windows users who are enrolled in the IBM ESS environment. There are several ways users can enroll, which will be covered in detail later in this document. This section covers what happens when a user enrolls. Understanding what happens during this process will give you a better understanding of how IBM ESS works and ultimately how to successfully manage this in your environment.

Client Security software uses the User Verification Manager (UVM) to manage passphrases and other elements to authenticate system users. UVM software enables the following features:

- UVM client policy protection
- UVM system logon protection
- UVM Client Security screen saver protection

Each user in the IBM ESS environment has at least one personalization object associated with him or her that is used for authentication purposes. The minimum requirement is a passphrase. Every user in the UVM component of ESS (from the user perspective, UVM manages authentication and enforces security policy) environment must have a passphrase and this passphrase must be given a minimum of once per computer start-up. The following sections will explain why a passphrase is used, how to set one up, and how to use it.

Requiring a passphrase

Simply put, a passphrase is required for security purposes. Having a hardware element such as the IBM Embedded Security Subsystem is a tremendous benefit because it provides a secure, autonomous location for a user's credentials to be operated upon. However, the protection that a hardware chip provides is of little use if the authentication required to access the chip is weak. For example, consider that you have a hardware chip that performs security functions. However, the authentication required to invoke an action by the chip is a single digit. This leaves a potential hacker the choice of guessing a single numerical digit (0 though 9) to invoke actions with your credentials. The single-digit authentication weakens the security of the chip such that it provides little or no added benefit over a software-based solution. If you don't have strong authentication in conjunction with the hardware protection, you could have no security gain at all. The passphrase required by IBM ESS is used to authenticate a user before any actions take place with the user's credentials in the hardware. The UVM passphrase is only recoverable through the administrator key pair, therefore it cannot be retrieved from a stolen system.

Setting up a passphrase

Each user selects a passphrase to protect their credentials. In Chapter 3, "How the Embedded Security Chip functions," on page 9, you saw that a user's private key is encrypted with the administrator public key. The user's private key also has an associated passphrase. This passphrase is used to authenticate the user with his or her credentials. Figure 17 shows the passphrase plus the private key component encrypted with the administrator public key.

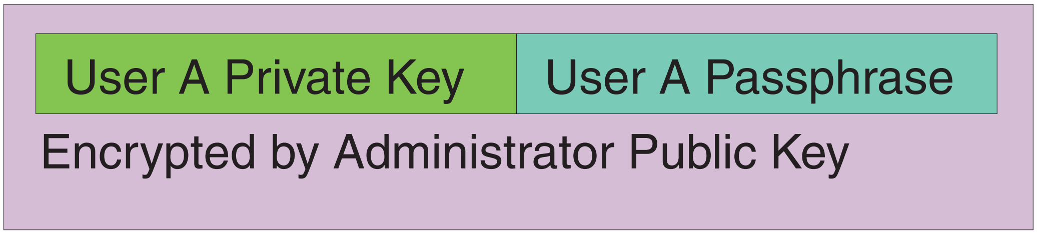

Figure 17. User A must provide the passphrase in order to perform any functions that require User A's private key.

The passphrase depicted in Figure 17 is selected by the user based on the existing policy, that is, the rules that are in place that control password creation such as number of characters, and number of days that the password is valid for. The passphrase is created when a user is enrolled into UVM. How this actually happens when rolling out IBM Client Security Software will be covered later in this document.

User A's private key is encrypted with the administrator public key, because decrypting the private key requires the administrator's private key. Therefore, if User A's passphrase is forgotten, the administrator can reset a new passphrase.

Using a passphrase

Figure 18 through Figure 20 on page 31, shows how the user passphrase is processed on the chip. A passphrase must always be used first and at least once per session. A passphrase is always required. You can choose to add additional authentication devices, but none of these can replace the initial user passphrase requirement. Briefly, the biometric or other authentication data are encrypted with the user's public key. Access to the private key is required to decrypt this additional security data.

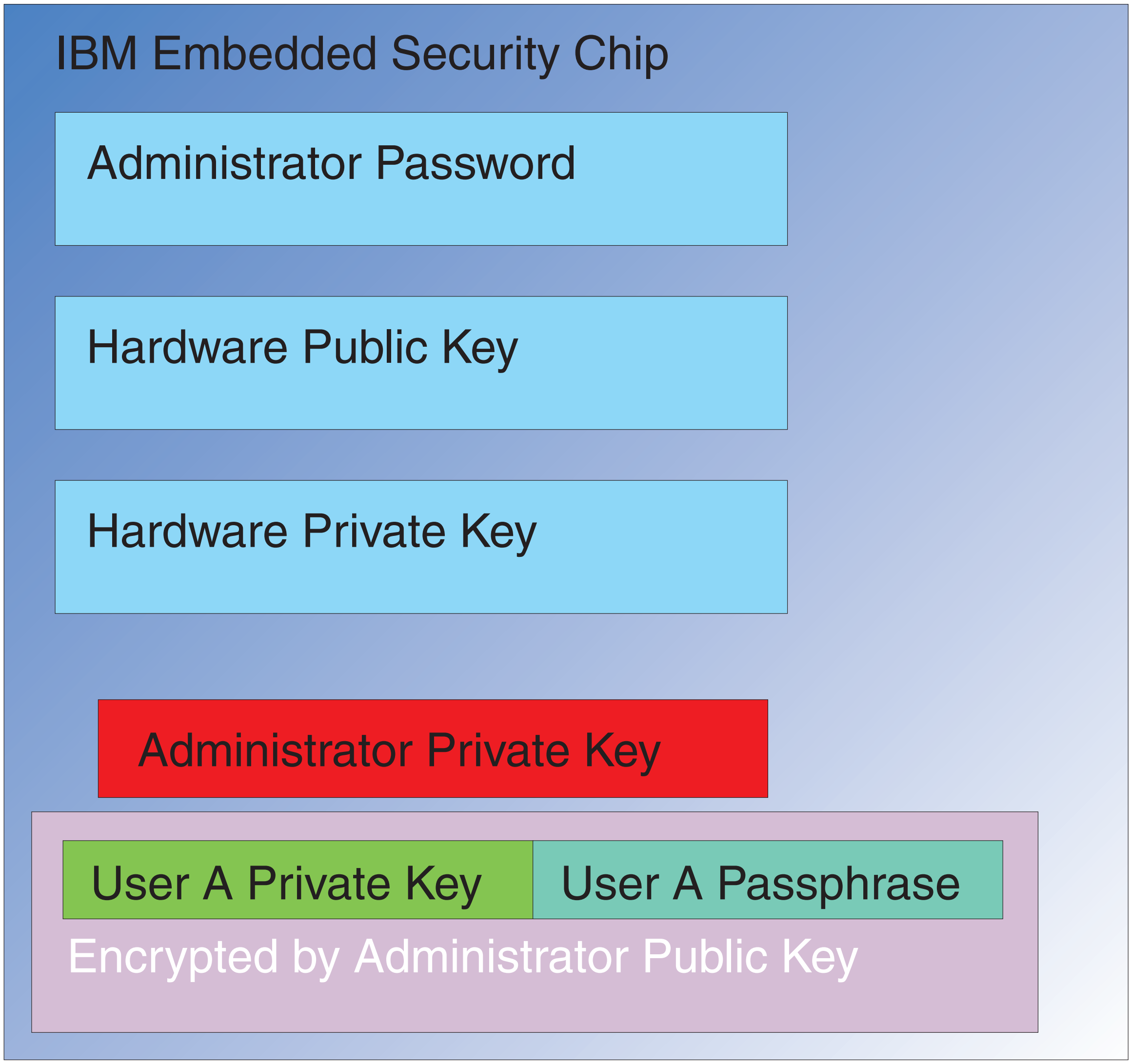

Figure 18. The Administrator's private key is decrypted in the chip.

Therefore, providing the passphrase at least once per session is required to decrypt the additional data. The credentials that constitute User A's Private Key and User A's Passphrase encrypted with the Administrator Public key is passed into the IBM Embedded Security Chip. The Administrator's private key is already decrypted in the chip as described earlier. The credentials are passed in as described in Figure 19 on page 30.

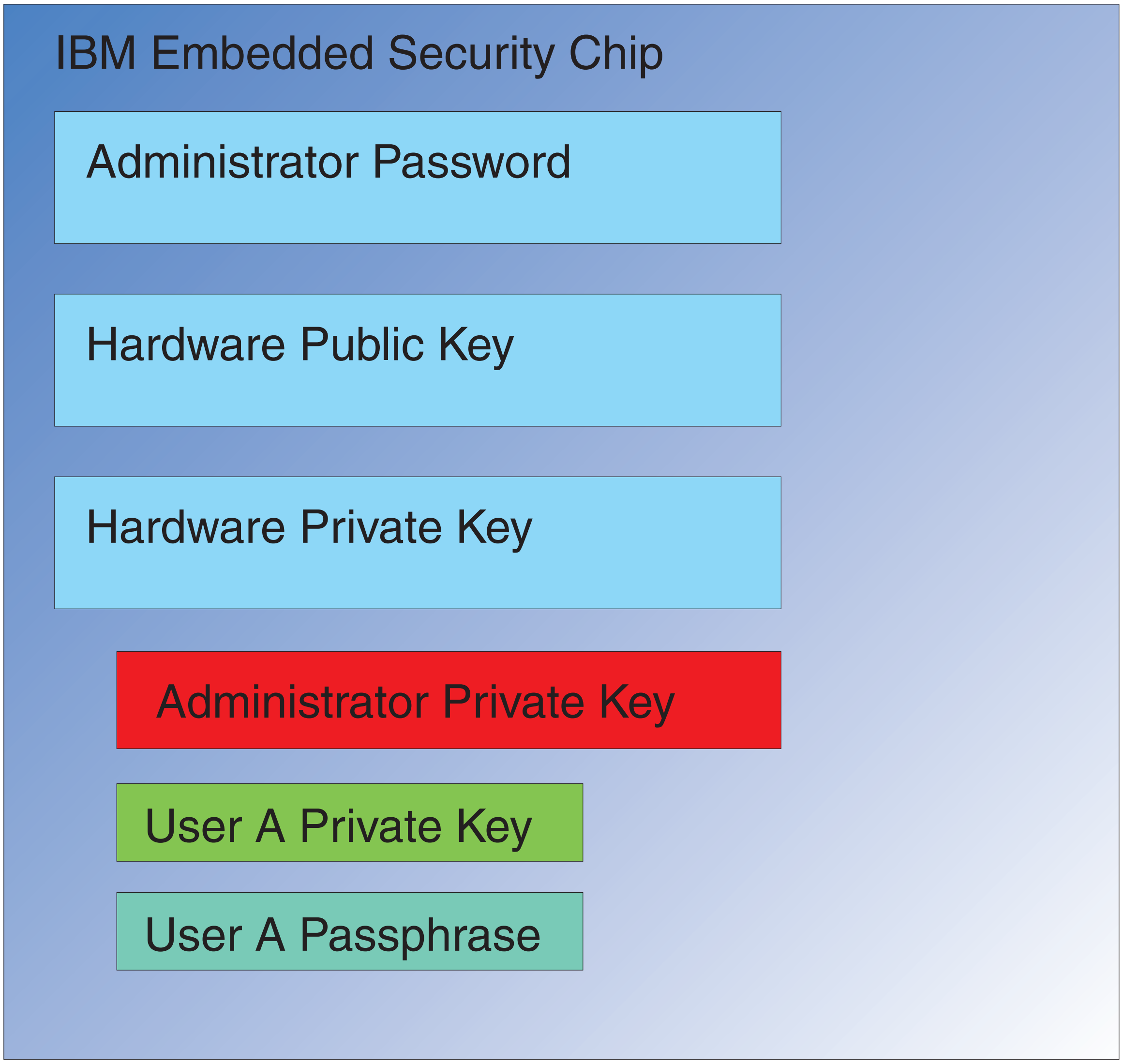

Figure 19. User A’s Private Key as well as User A’s passphrase are available in the chip.

The credentials are decrypted, making User A’s Private Key as well as User A’s passphrase available in the chip. When the currently logged-in user, identified by the IBM Client Security System as User A, attempts to use the credentials of User A, a passphrase dialog will open as shown in Figure 20 on page 31.

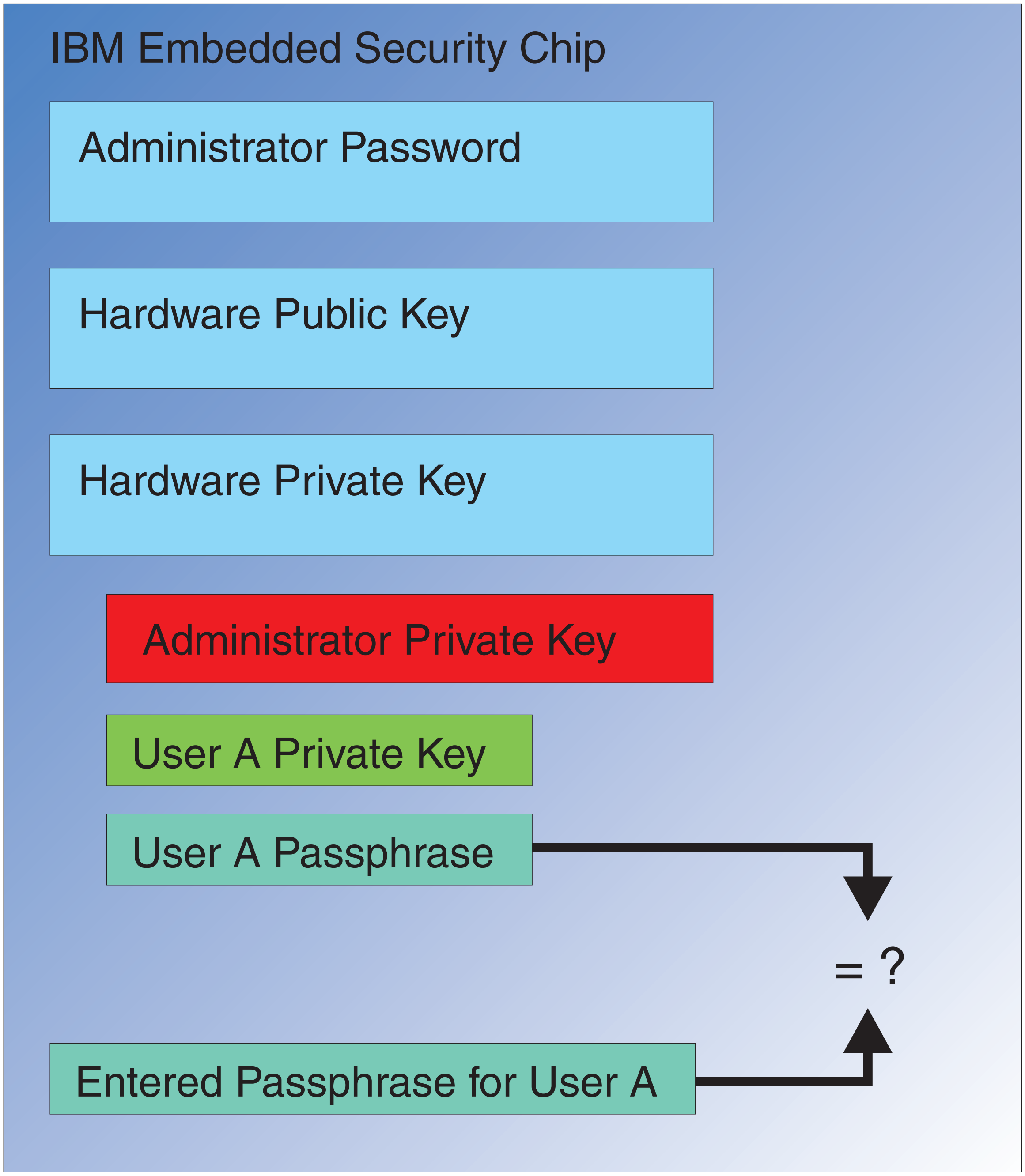

Figure 20. When User A attempts to use the credentials of User A then a passphrase dialog will open.

The typed passphrase is passed to the chip and compared to the decrypted passphrase value. If they match, then the credentials of User A can be used for various functions such as digital signatures or decrypting e-mails. This passphrase comparison is done in the secure environment of the chip. The chip has anti-hammering capabilities to detect repeated failed access attempts. Also User A's registered passphrase is never exposed outside of the chip. As part of the IBM Client Security Software installation, users are enrolled. Part of this enrollment process is the creation of the user's passphrase. The following section discusses the details of how this passphrase is set and how passphrase rules can be enforced.

Figure 1 on page 1 showed the IBM Embedded Security Chip as well as the IBM Client Security System. Figure 1 on page 1 also depicts Company initialization and user initialization. Company initialization is associated with the Embedded Security Subsystem and user initialization is associated with the IBM Client Security Software. The previous sections described the initialization that takes place to offer understanding of the general concept. The following sections will give more details on the process of initialization.

TPM Initialization

TPM initialization is essentially the process of adding hardware public and private keys and an administrator password. This process takes a generic machine as shipped from IBM and makes it unique for your enterprise. The following chart will show the methods for the initialization of public and private keys as well as Administrator passwords.

Table 3. Hardware initialization methods

Action	**Can be created in BIOS**	**Can be created manually by Administrator in CSS software**	**Can be created in a Script**
Hardware Public/Private Key Creation	No	Yes	Yes
Administrator Password Creation	On some TCPA-compatible clients, yes. Check for BIOS entry.	Yes	Yes

Table 3 demonstrates that the Hardware Public and Private keys are not created automatically when the software is installed. The Hardware Public and Private Key creation must be initiated manually in the software or by script. The administrator password can be created in BIOS, the IBM Client Security Software application, or by script. The chip controls the values set for the hardware public and private keys; you cannot set the values. Random-number capability in the chip is used to produce statistically random Public and Private key pairs. However, you do set the administrator password.

The administrator password, however, is different because the administrator must set this value. Several issues regarding the administrator password must be addressed:

- What will you set as the administrator password or passwords?
- Will you have more than one for various groups? If so, how will you logically make the determination of which computers have which password?
- Which administrator will have access to the password? If you have more than one password for separate groups of users, who will have access to which passwords?
- Will self-administered end users have access to the administrator password?

To make an effective decision regarding the items previously mentioned, it is important to understand what the administrator password enables you to do:

- Gain access to administrator utilities
- Add/remove users

- Define which IBM Client Security Software application/features can be used

Subsequent sections will explain the connection between the policy file and the administrator private key. Note for now that the administrator private key is required to change policy. Table 4 summarizes the abilities of having the administrator password and/or the administrator private key.

Table 4. Administrator actions based password and private key

Action	Administrator password	Administrator private key
Gain Access to Admin Utility	Yes	No
Add/Remove/Restore users	Yes	No
Define which CSS Application/features can be used	Yes	No
Define/Change policy	Yes	Yes
Create file to reset user's passphrase	Yes	Yes

TPM initialization also refers to the Administrator public and private key. From the preceding chart you can see the capabilities associated with this key. Give some thought to setting the Administrator public and private keys. This key pair can be unique for each computer or it can be the same for all machines. When the IBM Client Security Software is initialized the administrator will have the choice of using an existing key pair or creating a new key pair for the client. Once again, the usage model will determine what is best for your enterprise.

Best Practices

Large enterprises can use a unique key for each machine or a unique key for each department. For example, set an administrator password and/or administrator private key for all computers used in the human resources department, another for the engineering department, etc. You can also differentiate on a physical basis, such as by building or site location. Being able to determine which administrator private key to use when creating a passphrase reset file should be an easy process based upon who is requesting the reset. As Table 3 on page 32 and Table 5 on page 36 indicate, user and company, or hardware, initialization must also take place.

Setting security policy prior to deploying CSS

Security and authentication requirements will come from various interested parties in your organization. Although individuals with administrator access can make policy changes and ″push″ them to client computers (see Chapter 8, “Remotely deploying new or revised security policy files,” on page 57), configuring policy settings prior to deployment will provide best results. For additional information on setting policy, refer to ″Working with UVM Policy″ in the *Client Security Software Administrator's Guide*.

Preparing for forgotten passphrases or malfunctioning authentication devices

Users will inevitably forget a passphrase and there is the possibility that authentication devices, such as fingerprint biometric devices or SmartCards, will not work correctly.

Forgotten passphrase: The user's passphrase is not stored anywhere on the client hard disk or in the embedded security chip in human-readable form. It is kept secure in the user's mind and in one other location: the archive that is protected by the Administrator key pair. The administrator will need to decrypt the user's information held in the archive, using the Administrator private key. Then the Administrator can supply a new passphrase to the user.

When the user changes the passphrase, the new information will be archived in the specified archive location.

In the event that an authentication device malfunctions, you can configure IBM Client Security Software to present a **Click here to bypass** button. Clicking the bypass button solely challenges the user to type the passphrase successfully. Then the user can carry out secure tasks.

To configure CSS to show the bypass button, do the following:

1. In the CSEC.INI file (located in the root directory) locate the `AllowBypass= 0` entry. The default `0` value sets CSS to hide the bypass button.
2. Set the `AllowBypass` value to 1. The bypass button will show when CSS window challenges a user to provide authentication in addition to the passphrase.
3. Save the CSEC.INI file.

Notes:

1. In order to have this information archived, it is essential that the archive location be specified in the CSEC.INI file`kal=c:\jgk\archive`. Furthermore, if c:\jgk\archive is a network drive, that drive must be mapped on the client computer in order for the passphrase to be archived.
2. If you do not specify an archive location and that location is not mapped on the client computer, passphrases cannot be recovered.

User initialization

The IBM ESS provides the ability for several users to carry out independent and secure transactions on a single computer. These users must have a passphrase associated with them and may additionally have other authentication elements, such as fingerprints and/or smartcards. This is known as *Multi Factor Authorization*. User initialization is a critical step in configuring client computers to use the IBM ESS. User initialization is a two-part process:

1. Registration
2. Personalization

Registration

Registration is simply adding a user to, or registering a user with, the IBM Client Security System. In Figure 21 on page 35, you can see the User Verification Manger (UVM) component of the IBM Client Security Software. UVM controls each user's credentials as well as enforces policy.

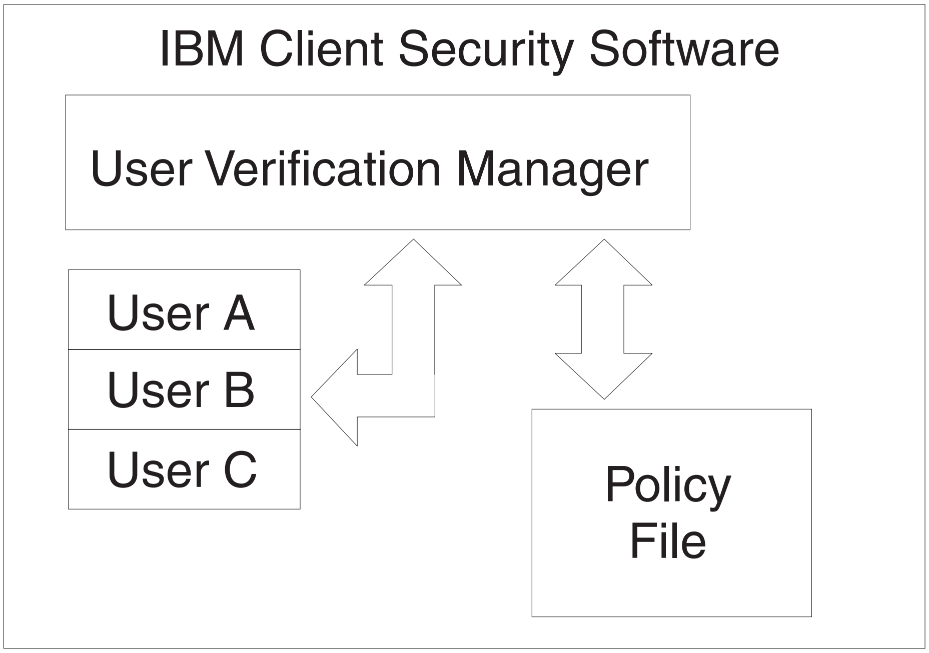

Figure 21. User Verification Manager controls each user's credentials and enforces security policies.

A policy file, such as that depicted in Figure 21, contains the authentication requirements for each user that UVM manages. UVM users are simply Windows users (local or Domain). UVM manages credentials based on who is currently logged onto the computer and operating system. For example, if User A logs into Windows and User A is also part of UVM, then UVM will enforce the policy when User A attempts to perform operations that require credentials. In another example, User A logs onto the computer. User A then goes into Microsoft Outlook and sends a digitally signed e-mail. The private key used to send that digitally signed e-mail is protected in the IBM Embedded Security Subsystem. Before UVM will permit that operation to be carried out, it will enforce policy as defined in the policy file. In this example, the requirement is for a passphrase to be authenticated before the operation is carried out. UVM will prompt the user for the passphrase and if it is verified correctly the private key operation will be carried out in the chip.

Personal Initialization

Personal initialization is simply setting an individual's personal UVM passphrase. Different people can perform the distinct parts of the process. The individual's UVM passphrase should be known to the individual only. However, if each individual does not perform the initialization process that person might need to perform an additional step. UVM can also be configured to force the user to change the passphrase the first time they log on.

For example, User A is initialized by the IT administrator. The IT Administrator selects User A from a Windows list of users (from a domain, for example). UVM asks for the UVM passphrase to be associated with User A. The IT administrator enters a ″default value″ of ″IT Admin Passphrase.″ To ensure security of the system, after User A receives the system he or she must customize the passphrase so no one could conduct secure transactions using the default passphrase.

Table 5. User initialization methods

Method	Command Process	Process Requirements
Manual	The Admin can manually personalize CSS for the user through the Admin Utility	Administrator must be present at each computer for setup.
Administrator configuration file	The Admin can create a configuration file, which contains an encrypted version of the Administrator Password. That file is sent to the user, who can then enroll individually without administrator intervention or presence.	User goes through setup process.
*.ini	The administrator creates a script that executes the.ini file and places a default or personalized password.	Admin or user presence optional.

Deployment scenarios

You are deploying 1,000 clients to 1,000 end users. One of the following might describe your approach to deployment:

- You know exactly which machine is going to which end user. For example, you know machine 1 is going to Bob, so you register Bob on machine 1. Bob must personalize (set his individual passphrase) when he receives the computer. Bob receives the computer, starts IBM Client Security Software, and then sets his passphrase.
- You do not know which machine is going to which user. You take client 1 and ship to end user X.

These two variable factors make deploying the IBM ESS different from deploying a typical application. However, there are several deployment options that provide flexibility in deploying IBM's ESS.

A typical flow diagram of PC delivery in your company may look like the following:

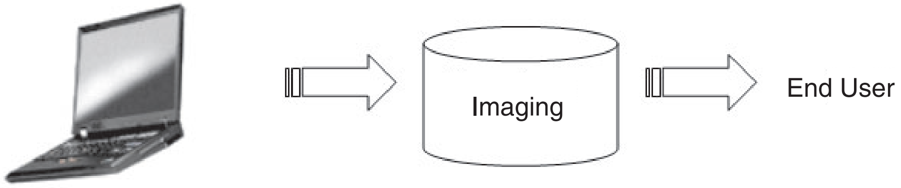

Figure 22. Typical PC deployment flow diagram

Six deployment scenarios

There are six deployment methods for IBM Client Security Software:

1. **Added component**—IBM Client Security Software code is not part of the disk image. It is installed, initialized, and personalized after computers have been deployed.
2. **Image component**—IBM Client Security Software code is part of the image, but is not installed. Neither company personalization nor user personalization has been initiated. (See Figure 23 on page 38.)
3. **Simple installation**—IBM Client Security Software is installed and has been personalized for the company or the end user. (See Figure 24 on page 39.)
4. **Partial personalization**—IBM Client Security Software is installed and company personalization has occurred, but end user personalization has not occurred. (See Figure 24 on page 39.)
5. **Temporary personalization**—IBM Client Security Software is installed and both company and user personalization has been set. The user will need to reset the user passphrase and, if required, provide other authentication information, such as fingerprint scans or smartcard association. (See Figure 25 on page 40.)
6. **Full personalization**—IBM Client Security Software is installed and both company and user personalization has been set. The administrator sets the user passphrase. If a fingerprint scan or other authentication is required, the user must provide that personalization. (See Figure 25 on page 40.)

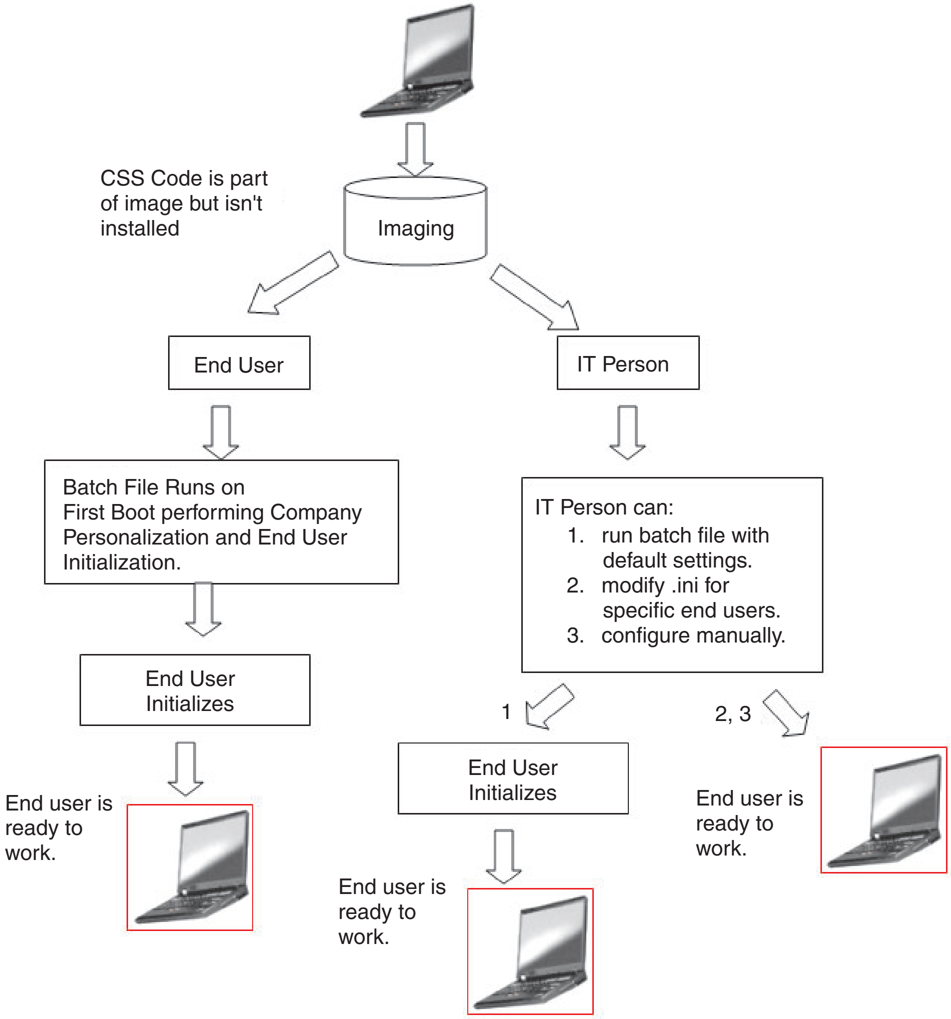
CSS Code is part of image but isn't installed
Imaging
End User
IT Person
Batch File Runs on First Boot performing Company Personalization and End User Initialization.
IT Person can:
1. run batch file with default settings.
2. modify .ini for specific end users.
3. configure manually.
End User Initializes
1
2, 3
End User Initializes
End user is ready to work.
End user is ready to work.
End user is ready to work.

Figure 23. IBM Client Security Software code is part of the image, but is not installed.

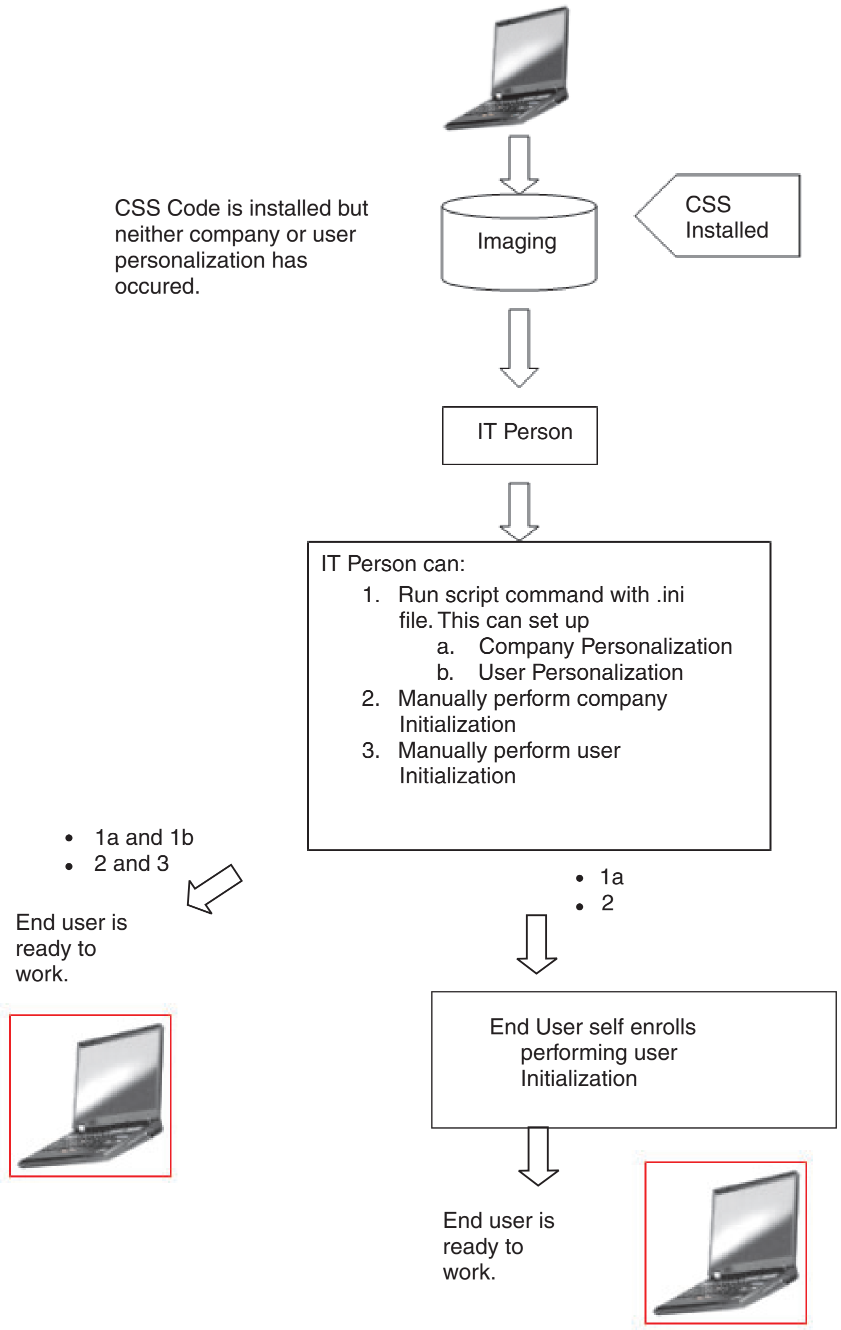
CSS Code is installed but neither company or user personalization has occured.
Imaging
CSS Installed
IT Person
IT Person can:
1. Run script command with .ini file. This can set up
a. Company Personalization
b. User Personalization
2. Manually perform company Initialization
3. Manually perform user Initialization
• 1a and 1b
• 2 and 3
End user is ready to work.
• 1a
• 2
End User self enrolls performing user Initialization
End user is ready to work.

Figure 24. IBM Client Security Software Code is installed but neither company or user personalization has occurred.

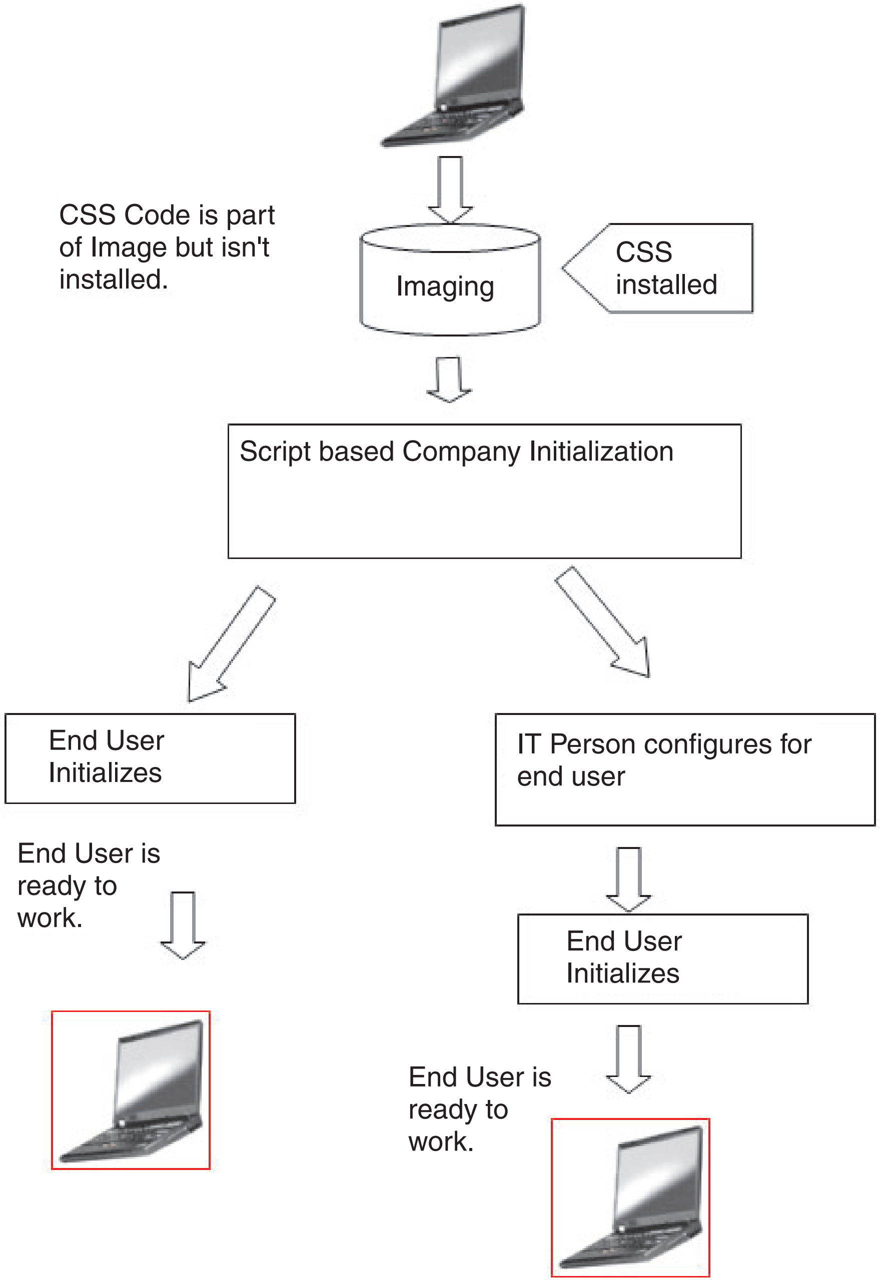
CSS Code is part of Image but isn't installed.
Imaging
CSS installed
Script based Company Initialization
End User Initializes
IT Person configures for end user
End User is ready to work.
End User Initializes
End User is ready to work.

Figure 25. IBM Client Security Software is installed and both company and user personalization has been set.

In scenario 1, IBM Client Security Software is deployed after the disk image is placed on the computer. The IBM Client Security Software installed and configured and the Embedded Security Chip is configured after the disk image has been installed.

Scenarios 2-6 represent various options of software deployment and configuration and chip configuration. Depending on your needs and your environment, you can select the scenario and the installation method the best meets your requirements.

Configuration file details

You can create the CSEC.INI file using the Client Security Wizard: CSECWIZ.EXE in the Security directory. After completing the wizard, mark the check box beside **Save settings, but do not configure subsystem. (Settings will be saved in C:\CSEC.INI)**.

Configuring

The cscc.ini file is essential when initiating a mass configuration. The file can be named anything, as long as it has a .ini extension. The following list details the settings and setting explanations for the .ini file you must create. Before you can open and revise the CSEC.INI file, you must first decrypt it, using CONSOLE.EXE in the Security folder.

Table 6. Client Security System configuration settings

[CSSSetup]	Section header for CSS setup.
suppw=bootup	BIOS Administrator/Supervisor password. Leave blank if not required.
hwpw=11111111	CSS hardware password. Must be eight characters. Always required. Must be correct if hardware password has already been set.
newkp=1	1 to generate a new administrator key pair 0 to use an existing administrator key pair.
keysplit=1	When newkp is 1, this determines the number of private key components. **Note:** If the existing keypair uses multiple private key parts, all private key parts must be stored in the same directory.
kpl=c:\jgk	Location of the administrator key pair when newkp is 1, if this is a network drive it must be mapped.
kal=c:\jgk\archive	Location of the user key archive, if this is a network drive it must be mapped.
pub=c:\jk\admin.key	Location of the administrator public key when using an existing administrator key pair, if this is a network drive it must be mapped.
pri=c:\jk\private1.key	Location of the administrator private key when using an existing administrator key pair, if this is a network drive it must be mapped.
wiz=0	Determines if this file was generated by the CSS setup wizard. This entry is not necessary. If you include it in the file the value should be 0.
clean=0	1 to delete the .ini file after initialization, 0 to leave the .ini file after initialization.
enableroaming=1	1 to enable roaming for the client, 0 to disable roaming for the client.

Table 6. Client Security System configuration settings (continued)

username= [promptcurrent]	[promptcurrent] to prompt the current user for the system registration password. [current] when the system registration password for the current user is provided by the sysregpwd entry and the current user has been authorized to register the system with the roaming server. [<specific user account>] if the designated user has been authorized to register the system with the roaming server and if the system registration password for that user is provided by the sysregpwd entry. Do not use this entry if the enableroaming value is 0, or if the enableroaming entry is not present.
sysregpwd=12345678	System registration password. Set this value to the correct password to enable the system to be registered with the roaming server. Do not include this entry if the username value is set to [promptcurrent], or if the username entry is not present.
[UVMEnrollment]	Section header for user enrollment.
enrollall=0	1 to enroll all local user accounts in UVM, 0 to enroll specific user accounts in UVM.
defaultuvmpw=top	When enrollall is 1, this will be the UVM passphrase for all users.
defaultwinpw=down	When enrollall is 1, this will be the Windows password registered with UVM for all users.
defaultppchange=0	When enrollall is 1, this will establish the UVM passphrase change policy for all users. 1 to require the user to change the UVM passphrase at next logon, 0 to not require the user to change the UVM passphrase at next logon.
defaultppexppolicy=1	When enrollall is 1, this will establish the UVM passphrase expiration policy for all users. 0 to indicate that the UVM passphase expires 1 to indicate that the UVM passphrase does not expire
defaultppexpdays=0	When enrollall is 1, this will establish the number of days until the UVM passphase expires for all users. When ppexppolicy is set to 0, set this value to establish the number of days until the UVM passphrase expires.
enrollusers=*x*, where x is the total number of users you will enroll on the computer.	The value in this statement specifies the total number of users that you will enroll. When enrollall is 0, this is the number of users that will be enrolled in UVM.
user1=jknox	Provide the information for each user to be enrolled starting with user 1. (There is no user 0.) User names must be the account names. In order to get the actual account name on XP, do the following 1. Start Computer Management (Device Manager). 2. Expand the Local Users and Groups node. 3. Open the Users folder. The items listed in the Name column are the account names.
user1uvmpw=chrome	Specify the UVM passphrase for user 1 UVM.

Table 6. Client Security System configuration settings (continued)

user1winpw=spinning	Specify the Windows passphrase for user 1 to be registered with UVM.
user1domain=0	Specify whether the account for user 1 is local or on the domain. 0 to indicate that this account is local, 1 to indicate that this account is on the domain.
user1ppchange=0	Specify whether user 1 will be required to change the UVM passphrase at next logon. 1 to require the user to change the UVM passphrase at next logon, 0 not to require the user to change the UVM passphrase at next logon.
user1ppexppolicy=1	Specify whether the UVM passphrase for user 1 expires. 0 to indicate that the UVM passphrase expires. 1 to indicate that the UVM passphrase does not expire.
user1ppexpdays=0	If user1ppexppolicy=0, set this value to indicate the number of days until the UVM passphase expires.
For each user provide a complete set of configuration settings in the order specified in the shaded portion of the table. Provide all parameters for one user, and then provide parameters for the next user. If for example enrollusers were set to 2, you would add the following group of configuration settings.	
user2=chrome	
user2uvmpw=left	
user2winpw=right	
user2domain=0	
user2ppchange=1	
user2ppexppolicy=0	
user2ppexpdays=90	
[UVMAppConfig]	Section header for UVM-aware application setup and UVM-aware module setup.
uvmlogon=0	1 to use UVM logon protection, 0 to use Windows logon.
entrust=0	1 to use UVM for entrust authentication, 0 to use entrust authentication.
notes=1	1 to use UVM protection for lotus notes, 0 to use notes password protection.
netscape=0	1 to sign and encrypt e-mails with the IBM PKCS#11 module, 0 to not sign and encrypt e-mails with the IBM PKCS#11 module.
passman=0	1 to use Password Manager, 0 to not use Password Manager
folderprotect=0	1 to use File and Folder Encryption, 0 to not use File and Folder Encryption.

Notes:

1. As IBM Client Security Software is enhanced and updated, the *.ini parameters may change.
2. If any files or paths are on a network drive, the drive must be mapped to a drive letter.

3. The CSEC.ini file must be encrypted for the software to load the contents. It must be encrypted via CONSOLE.EXE in the Security directory. The following command can also be used to encrypt an INI file via script. (Quotes are needed for long path names): *CSS installation folder*\console.exe /q /ini: *full path to an unencrypted ini file*
4. The following command runs the .ini file from the command line when the mass configuration is not performed along with a mass installation:

   CSS installation folder\acamucli /ccf:c:\csec.ini
5. The INI file supports the ability to add new users after the subsystem is configured which is useful for doing user enrollment. Run an INI file as described before, but do not include the ″pub=″ and ″pri=″ values. The code will assume user enrollment only and not reinitialize the subsystem.

IBM Client Security Software allows you to run the CSEC.INI file a second time without affecting the current Client Security Software installation. You might run this file a second time to enroll additional users, for example.

Table 7. Client Security System configuration settings when running a second time

[CSSSetup]	Section header for CSS setup.
suppw=	BIOS Administrator/Supervisor password. Leave blank if not required.
hwpw=11111111	CSS hardware password. Must be eight characters. Always required. Must be correct if hardware password has already been set.
newkp=0	Enter 0 to use an existing administrator key pair.
keysplit=1	When newkp is 1, this determines the number of private key components. **Note:** If the existing keypair uses multiple private key parts, all private key parts must be stored in the same directory.
pub=	Leave blank
pri=	Leave blank
kal=c:\archive	Location of the user key archive, if this is a network drive it must be mapped.
wiz=0	Determines if this file was generated by the CSS setup wizard. This entry is not necessary. If you include it in the file the value should be 0.
clean=0	Enter 0 to leave the .ini file after initialization.
enableroaming=0	Enter 0 to disable roaming for the client.
[UVMEnrollment]	Section header for user enrollment.
enrollall=0	1 to enroll all local user accounts in UVM, 0 to enroll specific user accounts in UVM.
enrollusers=1	The value in this statement specifies the total number of users that you will enroll.
user1=*eddy*	This is the name of the new user that is being enrolled.
user1uvmpw=pass1word	Specify the UVM passphrase for user 1 UVM.
user1winpw=	Specify the Windows passphrase for user 1 to be registered with UVM.
user1domain=0	Specify whether the account for user 1 is local or on the domain. 0 to indicate that this account is local, 1 to indicate that this account is on the domain.

Table 7. Client Security System configuration settings when running a second time (continued)

user1ppchange=0	Specify whether user 1 will be required to change the UVM passphrase at next logon. 1 to require the user to change the UVM passphrase at next logon, 0 not to require the user to change the UVM passphrase at next logon.
user1ppexppolicy=1	Specify whether the UVM passphrase for user 1 expires. 0 to indicate that the UVM passphrase expires. 1 to indicate that the UVM passphrase does not expire.
user1ppexpdays=0	If user1ppexppolicy=0, set this value to indicate the number of days until the UVM passphase expires.

Chapter 6. Installing the Client Security component on a Tivoli Access Manager server

Authenticating end users at the client level is an important security concern. Client Security Software provides the interface that is required to manage the security policy of an IBM client. This interface is part of the authenticating software, User Verification Manager (UVM), which is the main component of Client Security Software.

The UVM security policy for an IBM client can be managed in two ways:

- Locally, using a policy editor that resides on the IBM client
- Throughout an enterprise, using Tivoli® Access Manager

Before Client Security can be used with Tivoli Access Manager, the Client Security component of Tivoli Access Manager must be installed. This component can be downloaded from the http://www.pc.ibm.com/us/security/index.html IBM Web site.

Prerequisites

Before a secure connection can be established between the IBM Client and the Tivoli Access Manager server, the following components must be installed on the IBM Client:

- IBM Global Security Toolkit
- IBM SecureWay® Directory Client
- Tivoli Access Manager Runtime Environment

For detailed information about installing and using Tivoli Access Manager, see the documentation that is provided on the http://www.tivoli.com/products/index/secureway_policy_dir/index.htm Web site.

Downloading and installing the Client Security component

The Client Security component is available as a free download from the IBM Web site.

To download and install the Client Security component on the Tivoli Access Manager server and IBM client, complete the following procedure:

1. Using the information on the Web site, ensure that the IBM integrated security chip is on your system by matching your model number to one provided in the system requirements table; then click **Continue**.
2. Select the radio button that matches your Machine Type and click **Continue**.
3. Create a user ID, register with IBM by filling out the online form, and review the License Agreement; then click **Accept Licence**.

   You will automatically be redirected to the Client Security download page.
4. Follow the steps on the download page to install all necessary device drivers, readme files, software, reference documents, and additional utilities.
5. Install Client Security Software by completing the following procedure:

   a. From the Windows desktop, click **Start > Run**.
   b. In the Run field, type `d:\directory\csec53.exe`, where d:\directory\ is the drive letter and directory where the file is located.
   c. Click **OK**.

      The Welcome to the InstallShield Wizard for IBM Client Security Software window opens.
   d. Click **Next**.

      The wizard will extract the files and install the software. When the installation is complete, you will be given the option to restart your computer now or to wait until later.
   e. Select the appropriate radio button and click **OK**.
6. When the computer restarts, from the Windows desktop, click **Start > Run**.
7. In the Run field, type *`d:\directory`*`\TAMCSS.exe`, where *d:\directory*\ is the drive letter and directory where the file is located, or click **Browse** to locate the file.
8. Click **OK**.
9. Specify a destination folder and click **Unzip**.

   The wizard will extract the files to the specified folder. A message indicates that the files unzipped successfully.
10. Click **OK**.

Adding the Client Security components on the Tivoli Access Manager server

The pdadmin utility is a command-line tool that an administrator can use to perform most Tivoli Access Manager administration tasks. Multiple command execution enables an administrator to use a file that contains multiple pdadmin commands to perform a complete task or series of tasks. The communication between the pdadmin utility and the Management Server (pdmgrd) is secured over SSL. The pdadmin utility is installed as part of the Tivoli Access Manager Runtime Environment (PDRTE) package.

The pdadmin utility accepts a filename argument that identifies the location of such a file, for example:

```
MSDOS>pdadmin [-a admin-user][-p password ]file-pathname
```

The following command is an example of how to create the IBM Solutions object space, Client Security Actions, and individual ACL entries on the Tivoli Access Manager server:

```
MSDOS>pdadmin -a sec_master -p password C:\TAM_Add_ClientSecurity.txt
```

Refer to the *Tivoli Access Manager Base Administrator Guide* for more information about the pdadmin utility and its command syntax.

Establishing a secure connection between the IBM client and the Tivoli Access Manager server

The IBM Client must establish its own authenticated identity within the Tivoli Access Manager secure domain in order to request authorization decisions from the Tivoli Access Manager Authorization Service.

A unique identity must be created for the application in the Tivoli Access Manager secure domain. In order for the authenticated identity to perform authentication checks, the application must be a member of the remote-acl-users group. When the application wants to contact one of the secure domain services, it must first log in to the secure domain.

The svrsslcfg utility enables the IBM Client Security applications to communicate with the Tivoli Access Manager Management Server and Authorization Server.

The svrsslcfg utility enables the IBM Client Security applications to communicate with the Tivoli Access Manager Management server and the Authorization server.

The svrsslcfg utility performs the following tasks:

- Creates a user identity for the application. For example, DemoUser/HOSTNAME
- Creates an SSL key file for that user. For example, DemoUser.kdb and DemoUser.sth
- Adds the user to the remote-acl-users group

The following parameters are needed:

- **-f cfg_file** Configuration file path and name, use TAMCSS.conf
- **-d kdb_dir** The directory that is to contain the key ring database files for the server.
- **-n server_name** The actual Windows Username/UVM username of the intended IBM Client user.
- **-P admin_pwd** The Tivoli Access Manager Administrator password.
- **-s server_type** Must be specified as remote.
- **-S server_pwd** The password for the newly created user. This parameter is required.
- **-r port_num** Set the listening port number for the IBM Client. This is the parameter specified in the Tivoli Access Manager Runtime variable SSL Server Port for PD Management Server.
- **-e pwd_life** Set the password expiration time in number of days.

To establish a secure connection between the IBM client and the Tivoli Access Manager server, complete the following procedure:

1. Create a directory and move the TAMCSS.conf file to the new directory.

   For example, `MSDOS> mkdir C:\TAMCSS MSDOS> move C:\TAMCSS.conf C:\TAMCSS\`
2. Run svrsslcfg to create the user.

   MSDOS> svrsslcfg -config -f C:\TAMCSS\TAMCSS.conf -d C:\TAMCSS\ -n <server_name> - s remote -S <server_pwd> -P <admin_pwd> -e 365 -r 199

   Note: Replace <server_name> with the intended UVM username and hostname of the IBM client.For example: –n DemoUser/MyHostName. The IBM Client Hostname can be found by typing "hostname" at the MSDOS prompt. The svrsslcfg utility will create a valid entry in the Tivoli Access Manager server and provide a unique SSL key file for encrypted communication.
3. Run svrsslcfg to add the location of ivacld to the TAMCSS.conf file.

   By default, the PD Authorization server listens on port 7136. This can be verified by looking at the tcp_req_port parameter in the ivacld stanza of the ivacld.conf file on the Tivoli Access Manager server. It is important that you get the ivacld host name correct. Use the pdadmin server list command to obtain this information. The servers are named: **server_name-host_name**. The following is an example of running pdadmin server list:

```
MSDOS> pdadmin server list   ivacld-MyHost.ibm.com
```

The following command is then used to add a replica entry for the ivacld server displayed previously. It is assumed that ivacld is listening on the default port 7136.

```
svrsslcfg -add_replica  -f config file path -h host_name MSDOS>svrsslcfg
-add_replica  -f C:\TAMCSS\TAMCSS.conf -h MyHost.ibm.com
```

Configuring IBM clients

Before you can use Tivoli Access Manager to control the authentication objects for IBM clients, you must configure each client by using the Administrator Utility, a component that is provided with Client Security Software. This section contains prerequisites and instructions for configuring IBM clients.

Prerequisites

Make sure the following software is installed on the IBM client in the following order:

1. **Microsoft Windows supported operating system.** You can use Tivoli Access Manager to control the authentication requirements for IBM clients running Windows XP, Windows 2000, or Windows NT® Workstation 4.0.
2. **Client Security Software version 3.0 or later.** After you install the software and enable the IBM embedded Security Chip, you can use the Client Security Administrator Utility to set up user authentication and edit the UVM security policy. For comprehensive instructions on installing and using Client Security Software, see the *Client Security Software Installation Guide* and the *Client Security Software Administrator's Guide*.

Configuring the Tivoli Access Manager setup information

After Tivoli Access Manager has been installed on the local client, you can configure the Access Manager setup information by using the Administrator Utility, a software component that is provided by Client Security Software. The Access Manager setup information consists of the following settings:

- Selecting the full path to the Configuration File
- Selecting the Local Cache Refresh Interval

To configure the Tivoli Access Manager setup information on the IBM client, complete the following procedure:

1. Click **Start > Settings > Control Panel > IBM Embedded Security Subsystem**.
2. Type the Administrator Password, and click **OK**.

   After you enter your password, the Administrator Utility main window opens.
3. Click the **Configure Application Support and Policies** button.

   The UVM Application and Policy Configuration screen is displayed.
4. Select the **Replace the standard Windows logon with UVM's secure logon** check box.
5. Click the **Application Policy** button.
6. In the Tivoli Access Manager Setup Information area, select the full path to the TAMCSS.conf configuration file. For example, `C:\TAMCSS\TAMCSS.conf`

Tivoli Access Manager must be installed on the client for this area to be available.

7. Click the **Edit Policy** button.

   The Enter Administrator Password screen is displayed.
8. Type the Administrator Password in the provided field and click **OK**.

   The IBM UVM Policy screen is displayed.
9. Select the actions that you want Tivoli Access Manager to control from the Actions drop-down menu.
10. Select the Access Manager controls selected object check box so that a check appears in the box.
11. Click the **Apply** button.

    The changes take place at next cache refresh. If you want the changes to take place immediately, click the **Refresh Local Cache** button.

Setting and using the local-cache feature

After selecting the Tivoli Access Manager configuration file, the local cache refresh interval can be set. A local replica of the security policy information as managed by Tivoli Access Manager is maintained at the IBM client. You can schedule an automatic refresh of the local cache in increments of months (0-12) or days (0-30).

To set or refresh the local cache, complete the following procedure:

1. Click **Start > Settings > Control Panel > IBM Embedded Security Subsystem**.
2. Type the administrator password, and click **OK**.

   The Administrator Utility window opens. For complete information on using the Administrator Utility, see the *Client Security Software Administrator's Guide*.
3. In the Administrator Utility, click the **Configure Application Support and Policies** button and then the **Application Policy** button.

   The Modify Client Security Policy Configuration screen is displayed.
4. Do one of the following:
   - To refresh the local cache now, click **Refresh Local Cache**.
   - To set the automatic refresh rate, type the number of months (0-12) and days (0-30) in the fields provided, and click **Refresh Local Cache**. The local cache will refresh and the file expiration date will update to indicate when the next automatic refresh will take place.

Enabling Tivoli Access Manager to control IBM client objects

UVM policy is controlled through a global policy file. The global policy file, called a UVM-policy file, contains authentication requirements for actions that are performed on the IBM client system, such as logging on to the system, clearing the screen saver, or signing e-mail messages.

Before you can enable Tivoli Access Manager to control the authentication objects for an IBM client, use the UVM-policy editor to edit the UVM-policy file. The UVM-policy editor is part of the Administrator Utility.

Important: Enabling Tivoli Access Manager to control an object gives object control to the Tivoli Access Manager object space. If you do this, you must reinstall Client Security Software to re-establish local control over that object.

Editing a local UVM policy

Before attempting to edit the UVM policy for the local client, make sure at least one user is enrolled in UVM. Otherwise, an error message will be displayed when the policy editor attempts to open the local policy file.

You edit a local UVM policy and use it only on the client for which it was edited. If you installed Client Security in its default location, the local UVM policy is stored as \Program Files\IBM\Security\UVM_Policy\globalpolicy.gvm. Only a user who has been added to UVM can use the UVM-policy editor.

Note: If you set UVM policy to require fingerprints for an authentication object (such as the operating-system logon), users that are added to UVM must have their fingerprints registered to use that object.

To start the UVM-policy editor, complete the following Administrator Utility procedure:

1. Click the **Configure Application Support and Policies** button and then the **Application Policy** button.

   The Modify Client Security Policy Configuration screen is displayed.
2. Click the **Edit Policy** button.

   The Enter Administrator Password screen is displayed.
3. Type the Administrator Password in the provided field and click **OK**.

   The IBM UVM Policy screen is displayed.
4. On the Object Selection tab, click **Action** or **Object type**, and select the object for which you want to assign authentication requirements.

   Examples of valid actions include System Logon, System Unlock, and E-mail Decryption; an example of an object type is Acquire Digital Certificate.
5. For each object that you select, select **Tivoli Access Manager controls selected object** to enable Tivoli Access Manager for that object.

   Important: If you enable Tivoli Access Manager to control an object, you are giving control to the Tivoli Access Manager object space. If you later want to re-establish local control over that object, you must reinstall Client Security Software.

   Note: While you are editing UVM policy, you can view the policy summary information by clicking **Policy Summary**.
6. Click **Apply** to save your changes.
7. Click **OK** to exit.

Editing and using UVM policy for remote clients

To use UVM policy across multiple IBM clients, edit and save UVM policy for a remote client, and then copy the UVM-policy file to other IBM clients. If you install Client Security in its default location, the UVM-policy file will be stored as \Program Files\IBM\Security\UVM_Policy\remote\globalpolicy.gvm.

Copy the following files to other remote IBM clients that will use this UVM-policy:

- \IBM\Security\UVM_Policy\remote\globalpolicy.gvm
- \IBM\Security\UVM_Policy\remote\globalpolicy.gvm.sig

If you installed Client Security Software in its default location, the root directory for the preceding paths is \Program Files. Copy both files to the \IBM\Security\UVM_Policy\ directory path on the remote clients.

Troubleshooting charts

The following section contains troubleshooting charts that might be helpful if you experience problems with Client Security Software.

Digital certificate troubleshooting information

The following troubleshooting information might be helpful if you experience problems obtaining a digital certificate.

Problem Symptom	Possible Solution
UVM passphrase window or fingerprint authentication window displays multiple times during a digital certificate request	**Action**
The UVM security policy dictates that a user provide the UVM passphrase or fingerprint authentication before a digital certificate can be acquired. If the user tries to acquire a certificate, the authentication window that asks for the UVM passphrase or fingerprint scan displays more than once.	Type your UVM passphrase or scan your fingerprint each time the authentication window opens.
A VBScript or JavaScript™ error message displays	**Action**
When you request a digital certificate, an error message related to VBScript or JavaScript might display.	Restart the computer, and obtain the certificate again.

Tivoli Access Manager troubleshooting information

The following troubleshooting information might be helpful if you experience problems when using Tivoli Access Manager with Client Security Software.

Problem Symptom	Possible Solution
Local policy settings do not correspond to those on the server	**Action**
Tivoli Access Manager allows certain bit configurations that are not supported by UVM. Consequently, local policy requirements can override settings made by an administrator when configuring the PD server.	This is a known limitation.
Tivoli Access Manager setup settings are not accessible	**Action**
Tivoli Access Manager setup and local cache setup settings are not accessible on the Policy Setup page in the Administrator Utility.	Install the Tivoli Access Manager runtime Environment. If the Runtime Environment is not installed on the IBM client, the Tivoli Access Manager settings on the Policy Setup page will not be available.
A user’s control is valid for both the user and the group	**Action**
When configuring the Tivoli Access Manager server, if you define a user to a group, the user’s control is valid for both the user and the group if **Traverse bit** is on.	No action is required.

Lotus Notes troubleshooting information

The following troubleshooting information might be helpful if you experience problems with using Lotus Notes® with Client Security Software.

Problem Symptom	Possible Solution
After enabling UVM protection for Lotus® Notes, Notes is not able to finish its setup	**Action**
Lotus Notes is not able to finish setup after UVM protection is enabled using the Administrator Utility.	This is a known limitation. Lotus Notes must be configured and running before Lotus Notes support is enabled in the Administrator Utility.
An error message displays when you try to change the Notes password	**Action**
Changing the Notes password when using Client Security Software might display in an error message.	Retry the password change. If this does not work, restart the client.
An error message displays after you randomly-generate a password	**Action**
An error message might display when you do the following: • Use the Lotus Notes Configuration tool to set UVM protection for a Notes ID • Open Notes and use the function provided by Notes to change the password for Notes ID file • Close Notes immediately after you change the password	Click **OK** to close the error message. No other action is required. Contrary to the error message, the password has changed. The new password is a randomly-generated password created by Client Security Software. The Notes ID file is now encrypted with the randomly-generated password, and the user does not need a new User ID file. If the end user changes the password again, UVM will generate a new random password for the Notes ID.

Encryption troubleshooting information

The following troubleshooting information might be helpful if you experience problems when encrypting files using Client Security Software 3.0 or later.

Problem Symptom	Possible Solution
Previously encrypted files will not decrypt	**Action**
Files encrypted with previous versions of Client Security Software do not decrypt after upgrading to Client Security Software 3.0 or later.	This is a known limitation. You must decrypt all files that were encrypted using prior versions of Client Security Software *before* installing Client Security Software 3.0 or later. Client Security Software 3.0 cannot decrypt files that were encrypted using prior versions of Client Security Software because of changes in its file encryption implementation.

Chapter 7. Installing third-party hardware device drivers to complement IBM Client Security Software

With Client Security and third-party solutions, you can protect your entire infrastructure by integrating additional offerings, allowing you to tailor the level of protection for your computing environment.

The IBM Embedded Security Subsystem has been tested to comply with select security authentication hardware offerings from these organizations:

- Targus for fingerprint readers
- Gemplus for smart card solutions
- Ensure Technologies for proximity badges

Visit this Web site that contains links to these organizations to learn more about each organization's offerings: http://www.pc.ibm.com/us/security/

As with many components that are part of disk images, installation order is critically important. If you plan to deploy the authentication devices listed previously and their associated drivers and other software, IBM Client Security Software must be installed first. The drivers and software for these devices will not install correctly, if CSS is not placed on the hard disk before the device driver files.

For specific and up-to-date information about installing the software and drivers that enable authentication hardware, refer to the documentation that comes with the devices.

Chapter 8. Remotely deploying new or revised security policy files

Whether you are updating security policies or creating different policies for different computers, the IT administrator with signing authority can revise and deploy policy files. Edit the policy file, using ACAMUCLI.EXE. (You can also edit the policy by double-clicking the IBM Security Subsystem icon in the Control Panel.)

Sign the policy file according to on-screen instructions after you click Apply. (**Note:** if the administrator private key has been split, all components must be entered in order to sign the policy file.) The files you have edited are GLOBALPOLICY.GVM and GLOBPOLICY.GVM.SIG. Distribute these files to appropriate users the file, making sure that they are saved to the `C:\Documents and Settings\All Users\Application Data\ibmy` folder.

You can update passphrase policies remotely after deployment. Updating the passphrase policy file enables you to change the passphrase requirements when (or if) the user next changes their passphrase. The administrator can define a period of time, after which the user is forced to change the passphrase. This time period is defined during user enrollment or registration. An example would be as follows: The administrator enrolls a user, Jane, and the initial policy states that user Jane has to have an eight-character password that expires every 30 days. The administrator could update the policy file and require that the next time that Jane changes her passphrase, the new passphrase must now be 12 characters. The administrator could also change the expiration period. For example, instead of every 30 days the administrator could require Jane to change passphrases every 15 days. What happens in the following scenario? You are in day 10 of the 30-day passphrase ″life.″ A new passphrase policy file is sent to the client computer that states that the passphrase must be changed every 15 days. Does the passphrase expire in 5 days or in 20 days. The passphrase expires in 20 days as the original policy stated. The passphrase expiration policy goes into effect when the passphrase is set. The 15-day change policy will commence when Jane changes her passphrase after the 20 days.

If you want to change the required characteristics of the passphrase, follow the preceding instructions. Then distribute the following files from the SECURITY\UVM_POLICY folder: UVM_PP_POLICY.DAT and UVM_PP_POLICY.DAT.SIG.

Appendix. Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change IBM product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Non-IBM Web sites

Any references in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product, and use of those Web sites is at your own risk.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:
IBM
SecureWay
Tivoli

Microsoft, Windowsand Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.