yubico
the key to the cloud

YubiKey Integration for Full Disk Encryption

Pre-Boot Authentication

Version 1.2

May 7, 2012

Introduction

Yubico is the leading provider of simple, open online identity protection. The company's flagship product, the YubiKey®, uniquely combines driverless USB hardware with open source software. More than a million users in 100 countries rely on YubiKey strong two-factor authentication for securing access to computers, mobile devices, networks and online services. Customers range from individual Internet users to e-governments and Fortune 500 companies. Founded in 2007, Yubico is privately held with offices in California, Sweden and UK.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. Yubico shall have no liability for any error or damages of any kind resulting from the use of this document.

The Yubico Software referenced in this document is licensed to you under the terms and conditions accompanying the software or as otherwise agreed between you or the company that you are representing.

Trademarks

Yubico and YubiKey are trademarks of Yubico Inc.

Contact Information

Yubico Inc
228 Hamilton Avenue, 3rd Floor
Palo Alto, CA 94301
USA
info@yubico.com

Contents

1 Document Information

1.1 Purpose

The purpose of this document is to describe how Yubikey USB based two-factor authentication device can be implemented in a Preboot environment, in particular pre-boot for Full Disk Encryption (FDE) products.

An important advantage of the proposed implementation is that it does not require any network connectivity which is not yet commonly available in PBA environments and there are still some situations where functionality in offline mode is required (e.g. when there is no network coverage or on an airplane).

1.2 Audience

This document is intended for technical designers and implementers interested in implementing a YubiKey based strong two factor Pre-Boot Authentication solution for Full Disk Encryption products.

1.3 References

This document assumes that the reader is already familiar with the YubiKey features, technical details and the supporting Yubico software and tools listed below: (There is however also a short introduction to Yubikey in the Background section below.)

1) Information about the YubiKey

   http://www.yubico.com/yubikey

2) The"YubiKey Manual" (which provides overview and usage information, including Challenge-Response capabilities) and other user documentation can be found at

   http://www.yubico.com/support/documentation

3) YubiKey Personalization Tool

   http://www.yubico.com/personalization-tool

4) YubiKey Challenge-Response Tools

   http://www.yubico.com/challenge-response-tools

5) YubiKey Personalization cross-platform library

   https://github.com/Yubico/yubikey-personalization

1.4 Document History

Date	Version	Author	Activity
2011-02-15	1.0	KL and SP	First release
2011-03-10	1.1	KL and SP	Updated multiple sections
2012-05-08	1.2	Zwee	Changed document template

1.5 Definitions

Term	Definition
OTP	One Time Password
FDE aka WDE	Full Disk Encryption aka Whole Disk Encryption
PBA	Pre-Boot-Authentication
DEK	Disk Encryption Key
SED	Self-Encrypting Drive

2 Background

Yubico is a security company founded in 2007, with offices located in London, Stockholm Sweden and in Sunnyvale California.

Yubico's mission is to "make Internet identification secure, easy, and affordable for everyone". The Company offers a physical authentication device, the YubiKey, which can be used to provide secure authentication to resources like computing systems, web services and various applications.

Figure 1: The YubiKey

The YubiKey device from Yubico is a tiny key-sized one-button authentication device that acts as a USB keyboard. It does not require any readers and/or drivers (unlike smart cards for example) for its operation and works with most popular platforms and operating systems including Windows, Unix/Linux and Mac OS.

The YubiKey is very convenient to use and friendly to the environment; weighs just 2 grams, ultra-thin, crush-resistant, waterproof, battery-free, no moving parts and with virtually unlimited lifetime.

The feature rich YubiKey 2.2 has two configuration slots each of which can be independently configured in one of the following four modes of operation:

1. YubiKey OTP: 12 character ID + 32 character One Time Password
2. OATH 6 or 8 digit One Time Password (RFC 4226)
3. 1 to 38 character static pass code
4. Challenge-Response functionality, using client software

Yubico also offers supporting open source software components that can be directly used or further extended to meet custom authentication needs.

All these factors have resulted in rapidly growing adoption of YubiKey in consumer as well as Enterprise applications and markets.

The Challenge-Response mode newly introduced in YubiKey 2.2 opens up interesting new opportunities of using the YubiKey based authentication in offline mode where no network connectivity is available.

3 Introduction

This Application Note explains how FDE product vendors can use some of the open source software components offered by Yubico to quickly and easily implement YubiKey based strong two-factor Authentication capabilities into their products. The Yubikey will be the second factor – something you have (used in combination with a User name and Password – something you know).

The document covers integration into Pre-Boot Environment that may or may not have network access as well as briefly covers integration with the host native operating system. Yubikey One Time Password (OTP) device operating in standard “event” based OTP mode (generates an OTP each time the button on the device is pressed) requires network connection in order to get the OTP validated by a Yubico Validation Server. However, this mode will not easily work in an offline situation. A good way to implement a strong authentication technology that needs to work in both online and offline (network connected and without connection) is to use so called Challenge-Response technique (explained in detail in section 3.2 of this document). Yubikey can be programmed to work in Challenge- Response mode (HMAC-SHA1).

Before proceeding further, it would be helpful review some FDE specific considerations.

3.1 FDE Special Considerations

Full-Disk-Encryption (FDE) technology protect the data at rest by transparently (to the user) and at low level encrypt all data stored on the hard drive of a protected computers and is considered the most complete protection of such data.

We will now first at a high level walk through some concepts that are naturally fully known by all FDE vendors but the walkthrough helps to understand how Yubikey needs to be implemented in order to have a successful integration so bear with us.

FDE solutions normally encrypt all sectors on the hard disk, transparently to the native operating system, either by the use of an encryption driver or the encryption/decryption may be performed in hardware (in the computer chipset or on the disk itself – Self Encrypting Drive - SED). Both methods normally use a single key for encrypting sectors on the disk, the Disk Encryption Key (DEK) for software solutions and Data Encryption Key (DEK) for SED.

Most FDE solutions implement Pre-Boot Authentication by loading a hardened, lightweight operating system kernel (when the computer boots) called a Pre Boot Authentication environment (PBA) which is responsible for initial user authentication.

Therefore, in order to gain access the computer at power up the user must successfully authenticate to the PBA to generate (or unlock) the encryption key (DEK) and the PBA system will provide the encryption driver with the DEK for software solutions and for SED hardware it

PBA environments most commonly has no or limited network access capabilities and authentication would still have to work offline so any two- factor authentication solution will have to be self-sufficient and be able to authenticate the user without relying on a network connection to access an authentication server.

There are various two-factor authentication options available for many commercial FDE products. However, most of those options require installation of additional hardware e.g.

smart card readers and/or special drivers. YubiKey on the other hand is simple to use and has following distinct advantages:

a. Does not require any additional hardware to be installed
b. Uses standard USB port that is virtually ubiquitous on all personal computers used today and does not require
c. No special drivers need to be implemented in the pre-boot environment (only support for regular USB keyboards is required which is already built-in in virtually all computing systems used today)
d. YubiKey in Challenge/Response mode does not require network access in the pre-boot environment

The sections below will walk us through how two-factor authentication using Yubikey in Challenge/Response mode can be implemented to work seamlessly with FDE implementations.

Note: We did not discuss TPM (Trusted Platform Module) in the section above or below but it is an important component that should be mentioned when discussing PBA security. A TPM chip (currently found in more expensive business class computers) can simplified be thought of as a smartcard built into the computers motherboard that can be used in addition to the security methods described. A TPM chip does not replace a YubiKey; instead it complements the YubiKey by providing yet another layer of security such as providing an integrity check of the boot environment and more. A TPM chip cannot - unlike the YubiKey - be separated from the computer and only YubiKey can therefore be brought from computer to computer securing access on multiple computers using the same YubiKey and because YubiKey can be removed from the computer when not used it adds a significant hurdle for anyone trying to gain access to the computer if the computer is lost or stolen.

3.2 YubiKey Challenge-Response (HMAC-SHA1) mode

When a YubiKey 2.2 is programmed in Challenge-Response HMAC-SHA1 mode, there is support for programmatic interaction with the YubiKey. By using a client side software API, a 0 to 64 byte Challenge can be sent to the YubiKey as part of a request. The YubiKey computes HMAC-SHA1 on the Challenge using a 20 byte shared secret that is programmed into the YubiKey and the calculated digest i.e. Response is read via an API call (rather than by the means of recording keystrokes). This method provides additional security as the program calling the API can implement its own algorithm to generate a Challenge every time a request is made.

The following diagram illustrates the working of YubiKey 2.2 in Challenge- Response HMAC-SHA1 mode.

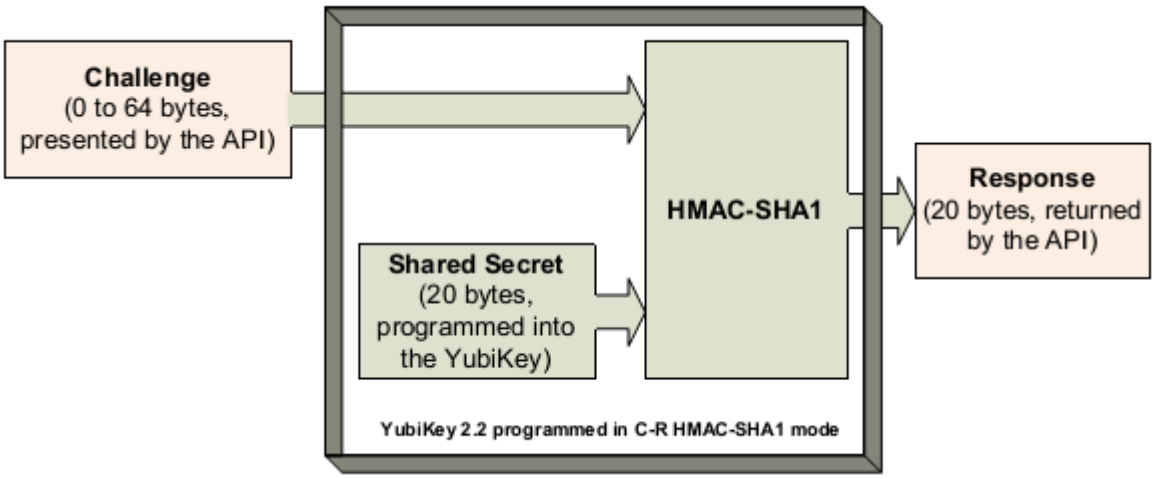

Figure 2: Working of Challenge-Response HMAC-SHA1 mode

Please refer to the following links for additional technical details on YubiKey Challenge-Response mode and the COM API:

http://static.yubico.com/var/uploads/pdfs/YubiKey_Manual_2010-09-16.pdf

http://www.yubico.com/challenge-response-tools

4 YubiKey two factor authentication for FDE

4.1 Design Overview

This section describes the suggested design for implementing YubiKey two-factor authentication in PBA for FDE products. We understand that there are many other ways to implement support so we recommend using the suggested implementation as a starting point that is proven to work.

4.1.1 Assumptions

1. A Pre-Boot-Authentication module is already implemented by the FDE vendor

2. The PBA module is written in a programming language that can interface with standard C libraries provided by Yubico

3. The PBA module is responsible for handling the encryption and validating the integrity of important secret parameters required by Yubico libraries.

4.1.2 Important data structures

The implementation scheme explained in this document uses the following important data structures that will be referred in the following sections:

Figure 3: User Authentication Record

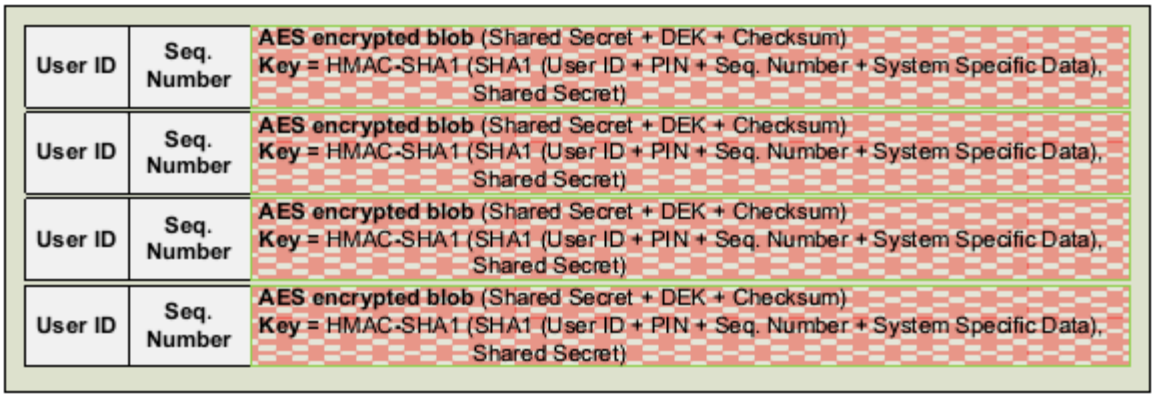

Figure 4: User Authentication Database

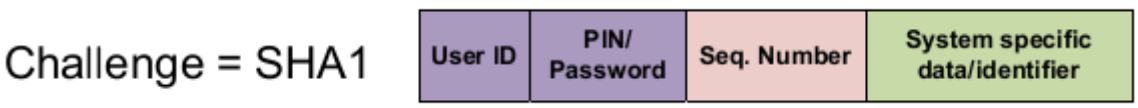

Figure 5: Generating a Challenge

4.1.3 YubiKey provisioning

An important and first step in the process is to provision and assign a YubiKey to a user. When a YubiKey is assigned to a user for PBA, the following steps will be performed and a new User Authentication Record (Figure 3 above) will be created and stored in the User Authentication Database (Figure 4 above):

Flow for provisioning:

1. The user will be asked to insert a YubiKey that will be used for PBA
2. The user will be required to provide a valid User ID and PIN/Password
3. The PBA module will perform its own authentication check on the User ID + PIN/Password provided by the user and if successful, the PBA process will proceed with YubiKey provisioning
4. FDE software vendor can select if the user needs to touch the YubiKey button at the time of PBA or the Challenge-Response mechanism can be completed under software control when the YubiKey is inserted a USB port (meaning authenticating without requiring the user to press the button)
5. A random 20 byte Shared Secret will be generated
6. The YubiKey will be programmed for Challenge-Response HMAC-SHA1 mode with the above shared secret.
7. An internal counter called Sequence Number will be initialized to a random, non-zero value
8. Cryptographic hash (SHA1) will be computed on (User ID + PIN + Sequence Number + some system specific information e.g. serial number of the hard drive) (see Figure 5 above)
9. This hash will be used as a challenge to compute HMAC-SHA1 response based on the Shared Secret in step 4 above (but without presenting the challenge to the YubiKey)
10. A User Authentication Record (see Figure 3 and Figure 4 above) will be created with the following information:
    a. User ID
    b. Sequence Number
    c. AES 256 encrypted blob (Shared Secret + Disk Encryption Key for the user + Checksum - CRC 16) using the response received in step 8 as the encryption key
11. The newly created User Authentication Record will be added to the User Authentication Database (see Figure 4 above).
12. The User Authentication Database will then be passed back to the PBA module for secure permanent storage on the hard drive.
13. This User Authentication Database needs to be provided by the PBA to YubiKey authentication module every time at the time of initialization (startup)
14. For any change to the User Authentication Database, the YubiKey authentication module will call a call-back function installed by the PBA module at the time of initialization and the PBA module will be responsible for saving the data in secure permanent storage.

The following high-level flow diagrams summarize the description below.

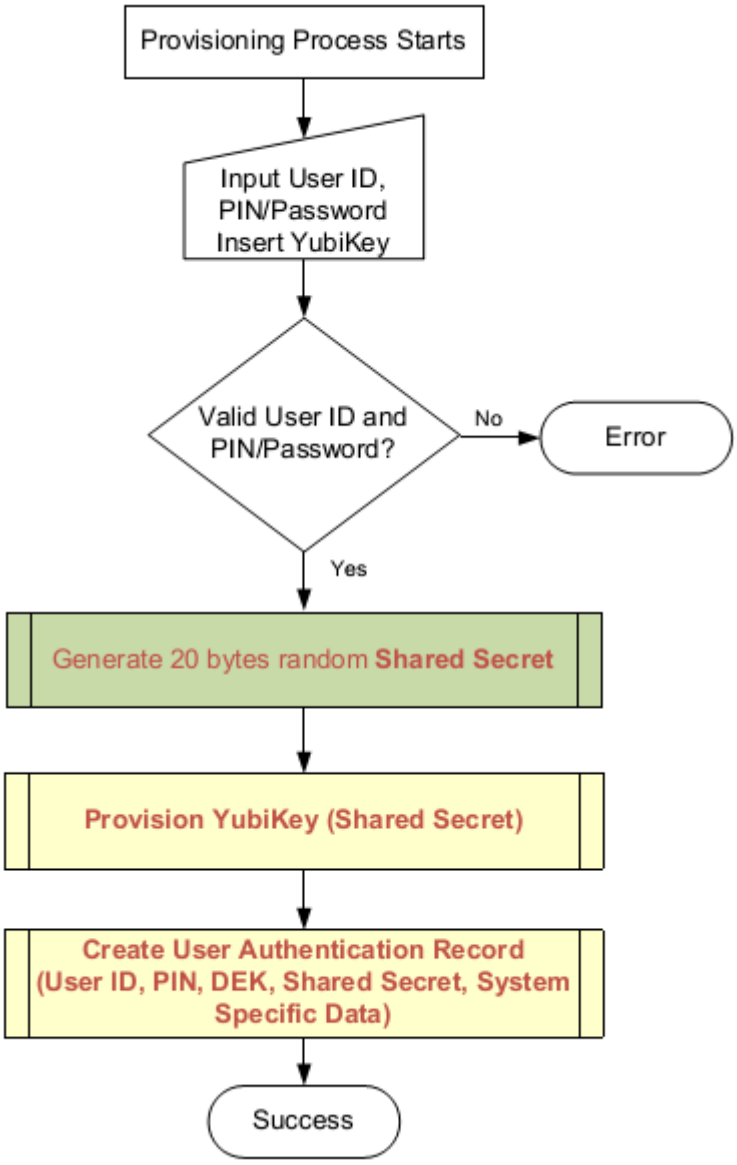

Figure 6: User Provisioning Process

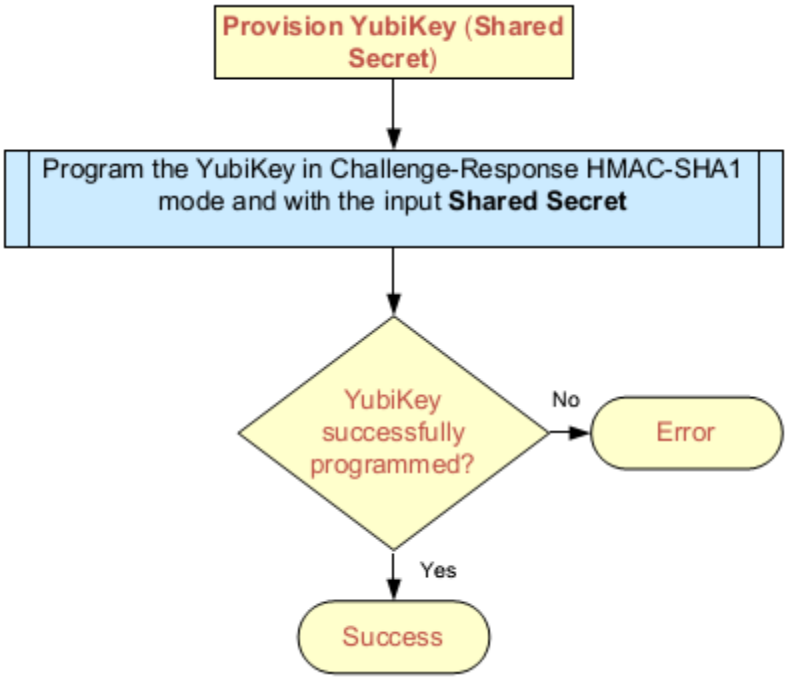

Figure 7: YubiKey Provisioning Process

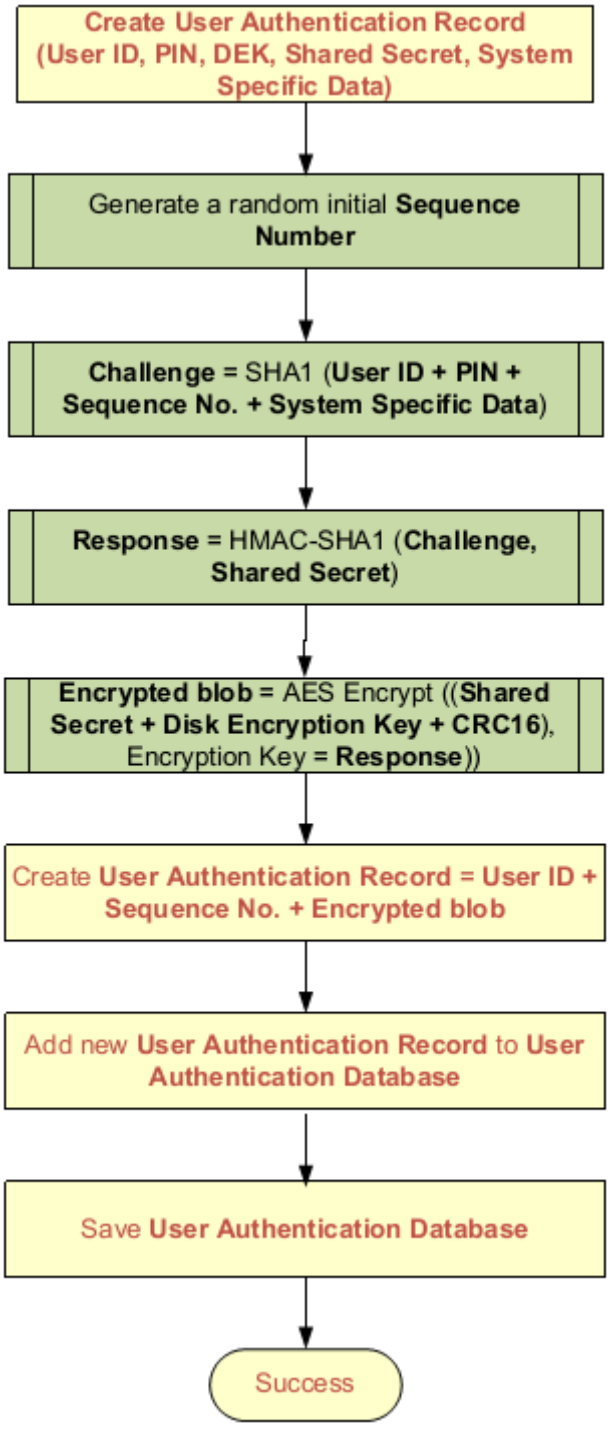

Figure 8: Creation of User Authentication Record

4.1.4 Two factor Pre-Boot-Authentication process

When the PBA module initiates the PBA process, it will initialize the YubiKey authentication module by passing the User Authentication Database created at the time of provisioning.

Thereafter it will prompt the user for User ID, PIN and YubiKey.

Once the user provides the above inputs the following checks will be performed:

1. A specific **User Authentication Record** will be identified based on the User ID provided by the user

2. A hash (SHA1) will be computed on (User ID + PIN + Sequence Number + some system specific information) and used as a Challenge

3. The Challenge will be presented to the YubiKey using YubiKey personalization library APIs and a Response will be received. Based on the implementation choice of the vendor the user may be required to touch the YubiKey button to generate a Response when the Challenge is presented.

4. The Response will be used to decrypt the encrypted portion of the User Authentication Record and if the **Checksum** matches, the decrypted DEK is successfully retrieved. A new sequence number will be calculated (either incremented by 1 or selecting a new random value). Please note, instead of the Checksum, FDE vendors can naturally choose to implement a cryptographic hash algorithm or simply use a known value that when decrypted is used to verify successful decryption of the encrypted portion of the User Authentication Record.

5. Steps 7 through 9 described in section 4.1.3 above will be repeated and the **User Authentication Record** in the **User Authentication Database** will be updated and saved in PBA secure permanent storage.

6. The PBA module will perform its own authentication check on the User ID + PIN/Password provided by the user and if successful, the PBA process will use the decrypted DEK to continue with the native OS boot process.

The following high-level flow diagrams summarize the above discussion.

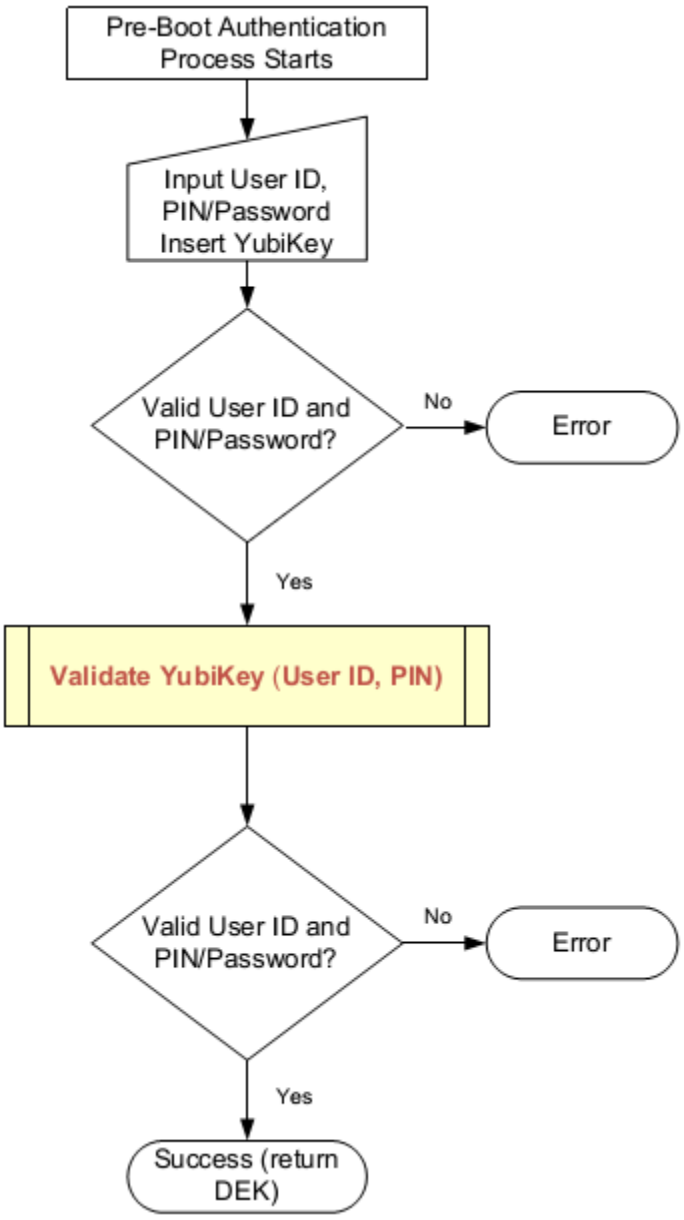

Figure 9: Typical PBA process

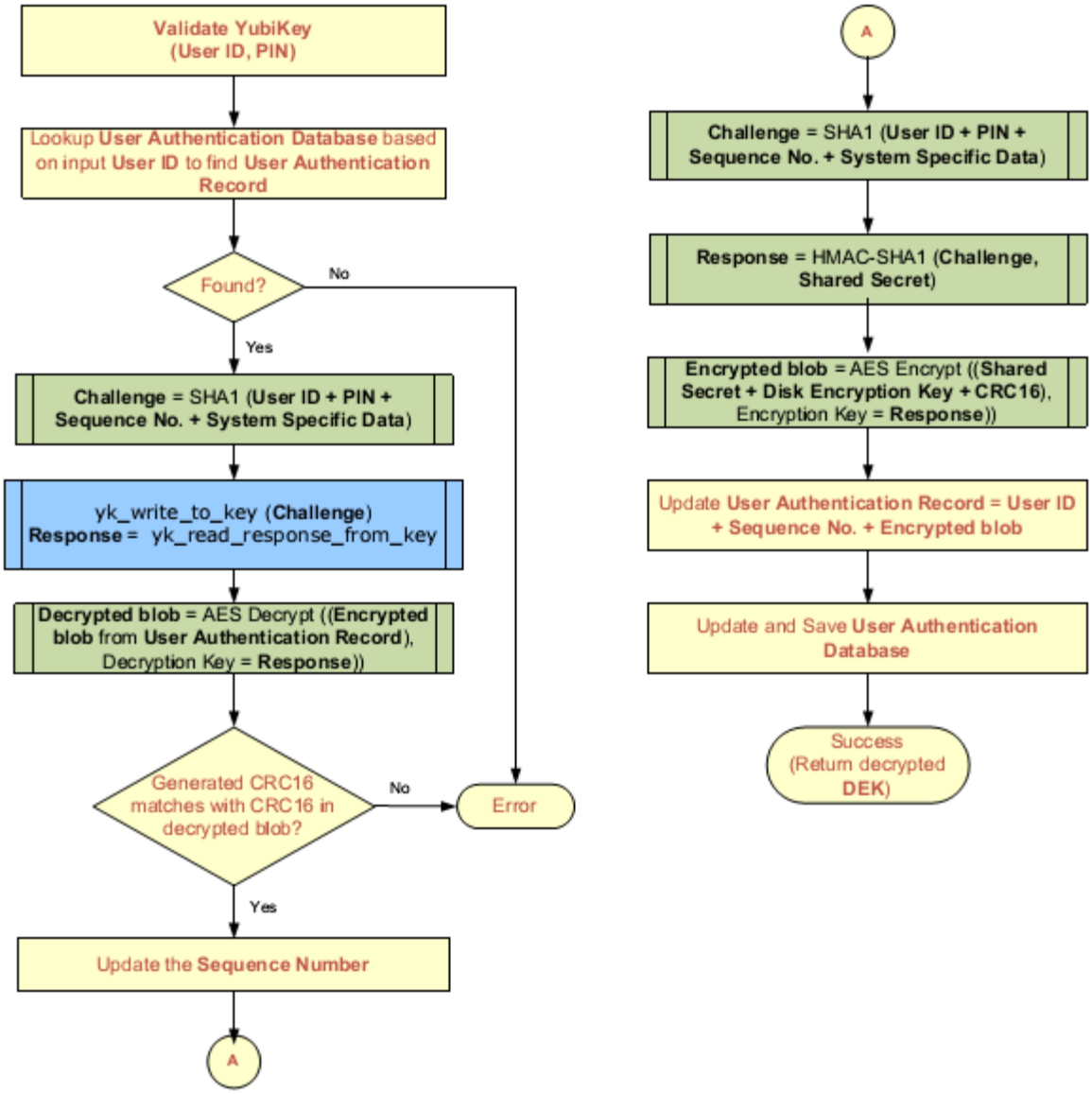

Figure 10: Validation of YubiKey as part of two factor PBA

4.1.5 Alternate security model

The design explained above keeps an encrypted copy of the Shared Secret programmed into the YubiKeys for users provisioned on a computer. The Shared Secret is generally well protected by encrypting it with a key that is cryptographically derived based on 3 distinct data sources:

a. User inputs (User ID and PIN/Password/Pass Phrase)
b. A Sequence Number that changes with each successful authentication
c. System specific data

This applies always when the User Authentication Records are stored on a permanent storage and most of the times when the User Authentication Database is loaded in the memory during the PBA process. However, there is still a brief interval of time when the key is held un-encrypted in computer memory (RAM) (Refer to Step 8 in Section 4.1.3 and to Step 6 in Section 4.1.4 above).

The advantage of this scheme isthat when updating the **User Authentication Record** in Step 6 of Section 4.1.4 above, the YubiKey authentication module does not send the new Challenge to the YubiKey thereby leaving no scope for USB man-in-the-middle (MIM) attack.

However, if brief exposure of unencrypted **Shared Secret** in computer memory is a concern, **User Authentication Record** can be modified to not store the **Shared Secret** and Step 8 of Section 4.1.3 and Step 6 of Section 4.1.4 could then be modified to present the new Challenge to the YubiKey and use the Response to AES encrypt part of the **User Authentication Record** to prepare it for the next authentication attempt.

Several challenges and responses can be pre-calculated in advance in order to make the challenge appear to be random.

4.1.6 Multi-user support

The above design supports provisioning of multiple users for the PBA process. Every time a new user is provisioned (identified by a new User ID), a new **User Authentication Record** will be created in the **User Authentication Database**.

4.1.7 Support for Windows Screen Saver

The design explained in this document will also work for implementing YubiKey two-factor authentication with Windows screen saver.

4.1.8 Backing up the Shared Secret on a Server

FDE products designed for large enterprise deploymentsneed to implement a mechanism to backup User Authentication Records on a Server/centralized repository from where it can be redistributed to other systems for situations where users should be able to authenticate to multiple computers (e.g. for administrators).

4.1.9 Handling Password Change and Password Synchronization

As the user PIN/Password (in the primary suggested implementation) is used in the process of recreating the Yubikey Challenge and in turn the Yubikey Response is used in the process of securing the User Authentication Record, then a Password change and Password Synchronization processes need to update the User Authentication Record with a new User Password.

4.1.10 Enabling a user to authenticate to multiple computers

Most FDE product vendors allows users to authenticate to multiple computers, the User Authentication Records for those users need to be propagated to the computers to which the user/a group of users can have access. The FDE vendor may have to implement additional functionality and interfaces on top of the libraries provided by Yubico for this purpose e.g. to accept the User Authentication Records from a Server/central repository and add them to the local User Authentication Database after additional processing.

4.1.11 Handling of lost/damaged/unavailable YubiKeys

Most FDE product vendors have already implemented robust and proven mechanisms ranging from Challenge-Response based help-desk or emergency recovery file or alike to recover from the lost passwords. The same mechanisms should continue to apply in this case as well except in some cases any specific extensions/changes that might be needed to allow the user to be assigned a new YubiKey in case the previous YubiKey is lost or damaged. Elaboration on product specific extensions/changes is beyond the scope of this document but vendors can contact Yubico to discuss possible approaches.

The following diagram illustrates some of these concepts applicable in an enterprise deployment.

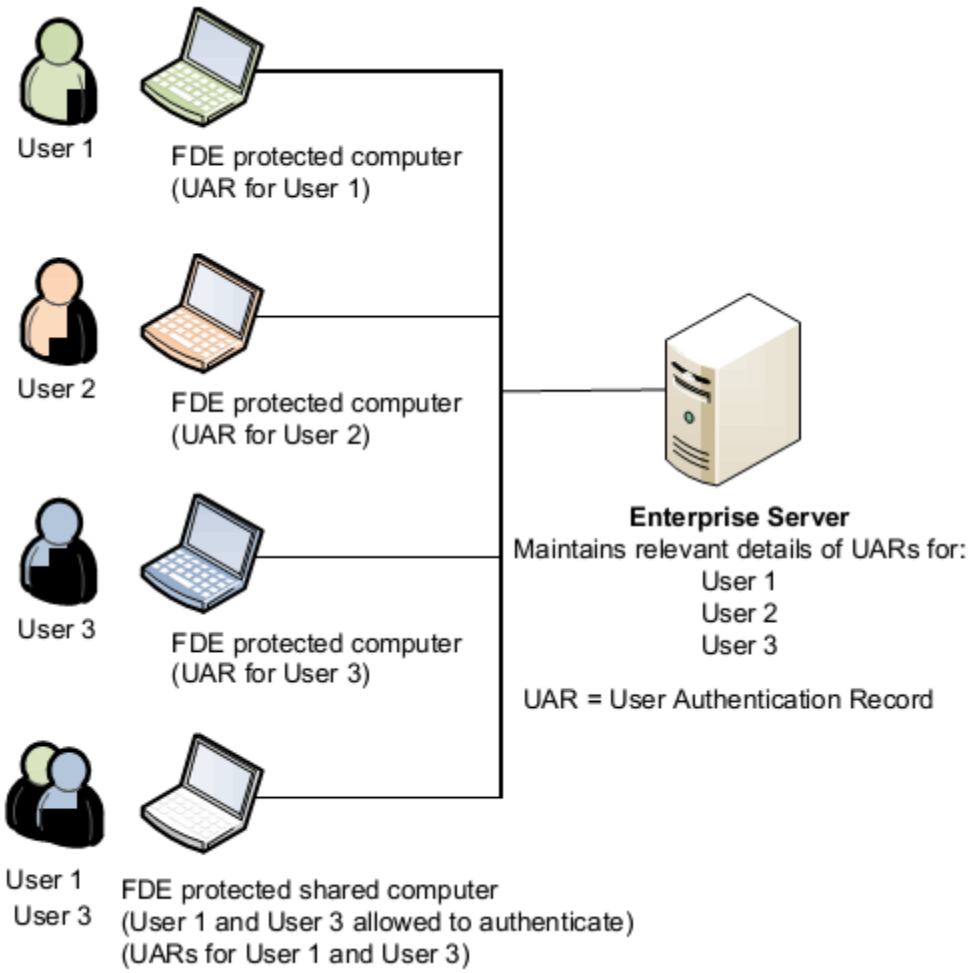

Figure 11: Typical enterprise deployment

5 Implementation Details

This section explains the software modules offered by Yubico, external dependencies and related implementation details that can be used for implementation of YubiKey support.

The high-level flow diagrams in the previous chapter used color coding for various functions to visually depict the containers for those functions.

- Blocks with white background are typically implemented in the PBA module by the FDE vendor
- Blocks with green background come from a Crypto library explained later in this chapter
- Blocks with blue background come from YubiKey-personalization library explained later in this chapter
- Finally blocks with yellow background are the ones that will be implemented as YubiKey Challenge-Response validation library by FDE vendor for YubiKey PBA integration.

Software Modules Offered by Yubico:

5.1 YubiKey personalization library

Yubico offers “yubikey-personalization” library (https://github.com/Yubico/yubikey-personalization) written in C that works on Windows, Linux and Mac OS platforms. This library should be used:

1. At the time of user provisioning for programming a YubiKey in C-R HMAC SHA1 mode with the shared secret
2. At the time of PBA process, present a challenge and receive a response from the YubiKey

5.2 YubiKey challenge-response (C-R) validation library

This library needs to be built by the FDE vendor based on the design explained in this document and the FDE product and platform specific needs. E.g. In most enterprise deployments, where synchronization with a Server is required (such as for backup, recovery purpose and for allowing users to authenticate with multiple computers) the same library can be used in multiple applications on multiple platforms e.g.:

a. In Pre-Boot Environment for PBA
b. In Windows environment for user authentication when screen saver is active
c. On Server side and other supported platforms for User Authentication and provisioning etc.

FDE vendor should implement the following functions in the YubiKey C-R validation library:

(Note: This section provides only a guideline by listing high-level set of functions that can be implemented in such a library and does not present a comprehensive list which will depend on the FDE vendor and product specific details.)

1. Initialization
   This function accepts inputs parameters from the calling module (e.g. from PBA module for Pre-Boot-Authentication or Windows for authenticating a user when screen saver is active). Some important input parameters are:
   a. User Authentication Database
   b. System Specific Data
   c. Call back function to be called when the User Authentication Database is updated and needs to be stored in the PBA secure permanent storage (the User Auth. Database gets updated in cases such as successful user authentication, User Authentication Record is created or deleted etc.)
2. Provision new User and YubiKey

   This function accepts User ID, PIN and a YubiKey to be inserted in a USB port. If User ID and PIN are valid, the YubiKey is programmed in C-R HMAC-SHA1 mode and a new User Authentication Record is created.

3. Create User Authentication Record

   This function accepts User ID, PIN, DEK, Shared Secret and System Specific Data and creates a new User Authentication Record

4. Validate YubiKey

   This function accepts inputs parameters from the calling module and validates if the User is using the same YubiKey assigned to the user. Some important input parameters are:
   a. User ID
   b. PIN

5. Change Password

   This function updates the **User Authentication Record** in case a user changed PBA PIN/Password. Please note that this function is not responsible of validating the PIN/Password policy enforcements and confirmation of new password etc. which would be the responsibility of PBA module developed by the FDE vendor.
   a. User ID
   b. Original Password
   c. New Password

6. Synchronize User Authentication Database

   This function is responsible for synchronizing the User Authentication Database (or relevant parts of it) with the central management Server. This function should identify any new User Authentication Records created and report them to the server and accept the records provided by the server and add them to the local User Authentication Database for the users provisioned on the current computer.

7. Terminate

   This function terminates the current active session with the calling module; the PBA is notified to store the array of user records and then all internal data structures are released.

5.3 Third party module dependencies

In most cases the FDE vendors will already have a supporting crypto library that is capable of:

1. Random number generation
2. AES encryption and decryption
3. SHA1 hashing
4. CRC16

In case a library is not available from the vendor, several open source C implementations are available for the above functions.

Vendors currently looking for a FIPS validated library are encouraged to consider OpenSSL project (www.openssl.org), which is a free library.

5.4 Target platform considerations

There are no target specific considerations at this time.

6 Checklist

The checklist in this section lists important considerations and design decisions to be made when implementing YubiKey two factor authentication for FDE Pre-Boot-Authentication so that YubiKey integration is a well though-out and functions smoothly.

The information in the checklist should also be used when registering the solution on the Yubico Wiki Partner page. It will help customers to quickly get an overview how the YubiKey implementation has been carried out in order for them to gage how well this solution to work with other YubiKey enabled solutions and applications that they may already use or planning to use.

No.	Consideration	Description	Your Selection/Design Decision
1	YubiKey user initiated authorization required for the PBA?	Users must touch the YubiKey button during the Pre-Boot- Authentication process?	□ Yes □ No No means the Challenge-Response mechanism will work under software control when a YubiKey is connected to a USB port and user is not required to touch the button.
2	Alternate Security Model	Refer section 4.1.5	□ Yes □ No
3	Multi-user support?	Different user names and Yubikeys can be used for logon in PBA mode (i.e. not singe user)	□ Yes □ No
4 a	Support for YubiKey Challenge-Response authentication to **unlock Windows screen saver**	Requires C/R User Authentication Records be accessible in the PBA mode and Windows mode	□ Yes □ No Design details:
4 b	Support for YubiKey Challenge-Response authentication at **Windows login**	Requires C/R User Authentication Records be accessible in the PBA mode and Windows mode	□ Yes □ No Design details:
5	Backing up the Shared Secret on a server	Useful for enterprise deployments, when a user needs to authenticate to more than one computers with the same YubiKey	□ Yes □ No Design details:
6	Handling password changes and synchronization		□ Yes □ No Design details:
7	Enabling a user to authenticate to multiple computers using the	User records must be synchronized between computers	□ Yes □ No Design details:

	same Username, PW and YubiKey		
8	Handling lost/damaged YubiKeys	Mechanisms for temporarily allow a user to gain access by other means and how to set up a new YubiKey in the field	□ Yes □ No Design details:
9	Selection of System Specific Information	System Specific Information is used to cryptographically secure part of the User Authentication Records. Select information that is fairly unique for different computer systems but at the same time the information must be consistently retrievable on a system by making simple system calls available in the pre- boot environment.	Design details:
10	Selection of crypto library	Indicate what crypto library that has been used. If FIPS validated library specify cert number.	Design details:

