Utimaco

Double Key Encryption (DKE)

Integration Guide

CryptoServer HSM

4.45 or higher and CP5-V5.1.1.1

utimaco®

Imprint

Copyright 2026	Utimaco IS GmbH Krefelder Straße 220 52070 Aachen Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226) EMEA +49 800-627-3081 APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	1.0.0
Date	2026-03-31
Status	PUBLISHED
Document No.	IG-2026-0022

All rights reserved	No part of this documentation may be reproduced in any form (printing, photocopy or according to any other process) without the written approval of Utimaco IS GmbH or be processed, reproduced or distributed using electronic systems. Utimaco IS GmbH reserves the right to modify or amend the documentation at any time without prior notice. Utimaco IS GmbH assumes no liability for typographical errors and damages incurred due to them. All trademarks and registered trademarks are the property of their respective owners.

Table of Contents

1 Introduction 5
1.1 About This Guide 5
1.2 Abbreviations 5
1.3 Document Conventions 6
1.4 Target Audience 7
2 Overview 8
2.1 Microsoft DKE 8
2.2 Utimaco SecurityServer HSM 8
2.3 Utimaco CryptoServer CP5 HSM 8
2.4 Utimaco DKE Anchor Manager_1 8
2.5 Utimaco DKE Anchor Service 9
3 Integration Requirements and Prerequisites 10
3.1 Minimum System Requirements 10
3.2 Tested Versions 10
3.3 Software Requirements 10
3.4 Hardware Requirements 11
3.5 Prerequisites 11
4 Utimaco PKCS#11 Configuration 13
4.1 On Linux 13
4.2 On Windows 13
4.3 Update cs_pkcs11_R<2/3>.cfg file 14
5 Download the Utimaco DKE Anchor Package 16
6 Generate PKCS12 Keystore for WebServer 17
7 Install DKE Anchor Manager and DKE Anchor Service 20
7.1 On Linux_2 20
7.2 On Windows_2 21
8 Post Installation Configuration 25
8.1 PKCS12 KeyStore 25
8.2 Configure the application.yml file 25
9 Utimaco DKE Anchor Manager 32
9.1 Execute the DKE Anchor Manager on Linux 32

9.1.1 Initialize and Authorize Key for CP5 ... 35
9.1.2 Generate a New Version of a Key for CP5 ... 39
9.1.3 Delete a User for a Key ... 41
9.2 Execute the DKE Anchor Manager on Windows ... 42
9.2.1 Initialize and Authorize Keys for CP5 ... 46
9.2.2 Generating a New Version of a Key for CP5 ... 50
9.2.3 Delete a User for a Key ... 52
10 Utimaco DKE Anchor Service_2 ... 54
10.1 Set the Environment Variable ... 54
10.1.1 On Linux ... 54
10.1.2 On Windows ... 54
10.2 Set the Environment Variable for Java ... 55
10.2.1 On Linux ... 55
10.2.2 On Windows ... 55
10.3 Start the DKE Anchor Service ... 56
10.3.1 On Linux ... 56
10.3.2 On Windows ... 57
10.4 Stop the DKE Anchor Service ... 57
10.4.1 On Linux ... 57
10.4.2 On Windows ... 58
10.5 Logging ... 58
11 Azure DKE Configurations ... 59
11.1 Register DKE Anchor Service with Azure ... 59
11.2 Create Sensitivity Labels using DKE ... 61
11.3 Publish the Labels ... 67
11.4 Enable DKE in your Client ... 72
11.5 Install the Azure Information Protection Unified Labeling Client ... 72
12 Verify Encryption and Decryption with DKE on Microsoft 365 Client ... 73
13 Troubleshooting ... 76
14 Further Information ... 80
15 References ... 81
16 Contact and Support Information ... 82

1 Introduction

This guide is part of the information and support provided by Utimaco. Additional documentation produced to support your Utimaco SecurityServer product can be found in the document directory of the Utimaco SecurityServer product bundle.

All Utimaco SecurityServer product documentation is available from Utimaco's website at https://utimaco.com.

1.1 About This Guide

This guide provides an integration guide explaining how to integrate an

Utimaco SecurityServer or CP5 Hardware Security Module (HSM) with Microsoft DKE.

1.2 Abbreviations

Abbreviation	Meaning
AIP	Azure Information Protection
API	Application Programming Interface
Azure AD	Azure Active Directory
CA	Certificate Authority
CSAR	Cloud Service Architecture
CSR	Certificate Signing Request
DKE	Double Key Encryption
FQDN	Fully Qualified Domain Name

Abbreviation	Meaning
GUI	Graphical User Interface
HSM	Hardware Security Module
IP	Internet Protocol
LAN	Local Area Network
PKCS#11	Public-Key Cryptography Standard #11
RSA	Rivest-Shamir-Adleman
URL	Uniform Resource Locator

Table 1: Abbreviations

1.3 Document Conventions

The following conventions are used in this guide:

Convention	Use	Example
Bold	Items of the Graphical User Interface (GUI), e.g., menu options	Press **OK**
`Monospaced`	Code that is given for explanation or as an example, file paths	`chsm-create`

Convention	Use	Example
Italic	References and important terms	See *Sample Chapter* in the *CryptoServer - Sample Manual*

Table 2: Document conventions

We use special icons to highlight the most important notes and information.

Here you will find important safety information.

Here you will find additional notes or supplementary information.

This message indicates the expected result after the successful execution of an instruction.

1.4 Target Audience

This guide is intended for administrators of Azure and Utimaco HSMs.

2 Overview

2.1 Microsoft DKE

Double Key Encryption (DKE) uses two keys together to access protected content. Microsoft stores one key in Microsoft Azure, and the user holds the other key. The user maintains full control of one of their keys using the Double Key Encryption service. The user applies protection using the Azure Information Protection unified labeling client to their highly sensitive content.

Double Key Encryption supports both cloud and on-premises deployments. These deployments help to ensure that encrypted data remains opaque wherever the user stores the protected data.

2.2 Utimaco SecurityServer HSM

SecurityServer is a hardware security module developed by Utimaco IS GmbH. SecurityServer is a physically protected specialized computer unit designed to perform sensitive cryptographic tasks and to securely manage as well as store cryptographic keys and data. It can be used as a universal, independent security component for heterogeneous computer systems.

2.3 Utimaco CryptoServer CP5 HSM

The CryptoServer CP5 is a hardware security module developed and manufactured by Utimaco IS GmbH. It is intended to be used as cryptographic module by trust service providers (TSP) supporting electronic signature and electronic sealing operations, certificate issuance and revocation, time stamp operations, and authentication services as identified in eIDAS Regulation (EU) No. 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market [Regulation].

2.4 Utimaco DKE Anchor Manager_1

Utimaco DKE Anchor Manager is a tool to create the key(s) and protect it using

Utimaco HSM. The public part of the key(s) will be used by Azure Information Protection to encrypt the customer's sensitive data; the private part will be used by Utimaco DKE Anchor service to decrypt the data after checking user authorization. This tool also helps customer to maintain versions of the key and add the authorized user list to DKE Anchor service. DKE Anchor Manager can securely generate, store, and protect encryption keys in the Utimaco HSM. DKE Anchor Manager uses PKCS#11 to interact with the Utimaco HSMs.

2.5 Utimaco DKE Anchor Service

Utimaco DKE Anchor service helps customer to protect their sensitive information in Microsoft 365. This service delivers public keys and decrypts data using the HSM. It can be deployed either on-premises or in the Cloud (Windows or Linux platform).

The DKE Anchor service is a web server with a public URL, providing services to get the public key and to decrypt data, which it needs to talk to Azure to verify the authentication token and to get some user information. This service also uses PKCS#11 to interact with the HSM. The AIP client gets the sensitivity labels from Azure, presents them to Microsoft 365 applications, and talks to the DKE Anchor service to get the public key and decrypt data. In Azure, the DKE Anchor service is registered, sensitivity labels are defined, and bound to a DKE Anchor service key URL.

3 Integration Requirements and Prerequisites

Ensure the system environment you will be using meets the following hardware and software requirements.

3.1 Minimum System Requirements

Operating System	CPU	RAM
Windows Server 2019	2 CPU	8 GB Memory
CentOS 8.2	2 CPU	8 GB Memory

Table 3: Minimum System Requirements

3.2 Tested Versions

The integrations that have been successfully tested with the Utimaco HSM with DKE:

Operating System	Utimaco Security Server Version	Utimaco HSM
Windows Server 2019 RHEL 8/CentOS 8	SecurityServer 4.45. CryptoServerCP5-V5.1.1.1	CryptoServer CSe-Series/Se-Series CryptoServer CP5

Table 4: Tested Versions

3.3 Software Requirements

Software	Software Requirements
Java	Version 11 or higher

Software	Software Requirements
HSM Utility	PKCS#11 Tool Version 2 (p11tool2)
HSM Interfaces	SecurityServer PKCS#11 Provider

Table 5: Software Requirements

3.4 Hardware Requirements

Hardware	Hardware Requirements
Utimaco LAN HSM	CryptoServer CSe-Series/Se-Series LAN with firmware SecurityServer 4.45 or higher CryptoServerCP5-V5.1.1.1
Utimaco PCI-e HSM	CryptoServer CSe-Series/Se-Series PCI-e with firmware SecurityServer 4.45 or higher CryptoServerCP5-V5.1.1.1

Table 6: Hardware Requirements

Setup an account on the Utimaco support portal and request download access at the following URL. https://support.hsm.utimaco.com/

3.5 Prerequisites

The following requirements must be met, to install and run the Utimaco DKE Anchor Manager and DKE Anchor Service.

- Java version must be 11.0.12 or higher.
- Utimaco SecurityServer or CP5 HSM is set up and configured.
  See the SecurityServer or CP5 documentation to set up the HSM.
- MBK must be created and stored on each HSM.
  See the SecurityServer or CP5 documentation to set up the MBK.
- SecurityServer or CP5 Default Admin should be replaced with a new admin user.

- Operating system as specified in Tested Versions.
- SecurityServer as specified in Tested Versions.
- PKCS#11 library is set up and configured for your environment.
  See the SecurityServer or CP5 documentation to set up and configure the PKCS#11 library.
- The DKE Anchor Service needs to run on a host
  - which has a FQDN (the app service URL)
  - which has a verified/registered domain
  - with https port 443 allowed through the firewall and reachable from all AIP clients
- The DKE Anchor Service application should be in the root path and no sub path should be specified.
- DKE requires a Microsoft 365 E5 license, and it works only for “Microsoft Office Apps for enterprise version 2009 or later (Desktop versions of Word, PowerPoint, and Excel) on Windows”.
- The custom domains should be in verified state. For more details see https://learn.microsoft.com/en-us/azure/active-directory/fundamentals/addcustom-domain
- Microsoft 365 requires an internet connection. The list of FQDN or wildcard domains and IP addresses should be allowed from the customer network to Microsoft 365 as mentioned by Microsoft, see ID 56 under https://docs.microsoft.com/enus/microsofst-365/enterprise/urls-and-ip-address-ranges?view=o365worldwide#microsoft-365-common-and-office-online.

4 Utimaco PKCS#11 Configuration

This section describes the process of configuring the PKCS#11 Provider for Utimaco DKE Anchor Service.

4.1 On Linux

Create the directory `/etc/utimaco`. We will copy the Utimaco PKCS#11 configuration file `cs_pkcs11_R<2/3>.cfg` to this directory. It is located in your CryptoServer directory `Linux/x86-64/Crypto_APIS/PKCS11_R<2/3>/sample`.

›_ Console

```
# mkdir /etc/utimaco

# cd <install directory>/Software/Linux/x86-

64/Crypto_APIs/PKCS11_R<2/3>/sample

# cp cs_pkcs11_R<2/3>.cfg /etc/utimaco # cd /etc/utimaco
```

Also see following command to set environment variable

›_ Console

```
# export CS_PKCS11_R<2/3>_CFG=/etc/utimaco/cs_pkcs11_R<2/3>.cfg
```

Use CS_PKCS11_R2 or CS_PKCS11_R3 based on the version of Utimaco PKCS#11 library.

4.2 On Windows

Under Windows, `cs_pkcs11_R<2/3>.cfg` is automatically generated as part of the CryptoServer software installation and is available under `C:\ProgramData\Utimaco\PKCS11_R<2/3>`.

4.3 Update cs_pkcs11_R<2/3>.cfg file

Example Values:

›_ Console

```
[Global]
# For unix:
# Logpath = /tmp
# For windows:
Logpath = C:/ProgramData/Utimaco/PKCS11_R<2/3>
# Loglevel (0 = NONE; 1 = ERROR; 2 = WARNING; 3 = INFO; 4 = TRACE)
Logging = 1

# Prevents expiring session after inactivity of 15 minutes
KeepAlive = true

# Set the Device to connect with
[CryptoServer]
# Device specifier
Device = <HSM_IP>
```

Use CS_PKCS11_R2 or CS_PKCS11_R3 based on the version of UtimacoPKCS#11 library.
For more information regarding the commands and command parameters please check the CryptoServer documentation. The device may be a CryptoServer (PCIe or LAN) device.
The device line will follow one of these patterns, based on the HSM form-factor:
`Device = 288@<HSM IP address>` Hardware (LAN) HSM
OR
`Device = /dev/cs2.0Hardware` (PCIe) HSM

To make your testing easier, it would be good to enable the PKCS#11 log file. That can be enabled by editing the logging loglevel. Specify the `LogPath` and set `Loglevel` to `1`. For testing you may want to increase it to `4`.

The `LogPath` points to a writable directory, not to a file. If you encounter problems, check the log file `cs_pkcs11_R2.log` in the under `LogPath` defined directory. When you are done testing, should change `Loglevel` to 1 or 2. This will limit the logging to critical and important messages.

5 Download the Utimaco DKE Anchor Package

Download the Utimaco DKE Anchor package from the support portal to set up and configure the DKE Anchor Service and DKE Anchor Manager.

6 Generate PKCS12 Keystore for WebServer

The PKCS12 keystore is used by the web server to enable SSL capabilities.

Follow the steps below to generate a pkcs12 keystore using the `keytool` command.

1. Generate a pkcs12 keystore.

›_ Console

```
# keytool -genkey -alias somealias -keystore keystore.p12 -storetype PKCS12
keyalg RSA -storepass somepass -validity 730 -keysize 4096
```

```
-genkey     Generate keystore
-alias      Alias of the generated private key entry
-keystore   Keystore file to be created
-storetype  Type of keystore. PKCS12 in this example
-keyalg     Key algorithm of key entry to be generated
-storepass  Password to set on both the key entry and keystore
-validity   Validity of the certificate associated with the key entry
-keysize         Size of the generated private key in bits
```

```
# keytool -genkey -alias dkeservice -keystore dkestore.p12 -storetype PKCS12 -
keyalg RSA -storepass 123456 -validity 730 -keysize 4096
What is your first and last name?
  [Unknown]:  dkeservice.utimaco.com
What is the name of your organizational unit?
  [Unknown]:  Utimaco
What is the name of your organization?
  [Unknown]:  Utimaco
What is the name of your City or Locality?
  [Unknown]:  Campbell
What is the name of your State or Province?
  [Unknown]:  CA
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=dkeservice, OU=IT, O=Utimaco, L=Campbell, ST=CA, C=US correct?
            [no]:  yes
```

Figure 1 : Keytool: genkey output

2. Generate the certificate signing request.

›_ Console

```
# keytool -certreq -alias dkeservice -keystore dkestore.p12 -file
dkeservice.csr

Enter keystore password:
```

Provide the keystore password when prompted.

3. Get the CSR signed from Certificate Authority.
4. Use the keytool to import the CA certificate into the keystore.

›_ Console

```
# keytool -import -alias CaRoot -keystore dkestore.p12 -file ca.cer
```

Provide the keystore password when prompted.

```
# keytool -import -alias CaRoot -keystore dkestore.p12 -file ca.cer
Enter keystore password:
Owner: CN=utimaco.com-CA, DC=utimaco, DC=com
Issuer: CN=utimaco.com-CA, DC=utmaco, DC=com
Serial number: 21247f2d7922338b40b229ae48747ab8
Valid from: Fri Jan 14 06:18:48 UTC 2022 until: Thu Jan 14 06:28:48
UTC 2027
Certificate fingerprints:
SHA1: B2:52:FC:04:B1:C5:28:78:EB:8B:24:C9:85:02:9A:83:E7:34:F9:FA
SHA256:5F:27:8C:7E:B5:6D:E9:F1:2A:AC:40:3D:85:40:FB:40:6D:25:15:41
:F1:BB:1E:48:29:06:51:5A:16:AC:13:45
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3
....
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

Figure 2 : Keytool: import the CA certificate

5. Use the keytool to import the signed certificate for the associated alias into the keystore.

›_ Console

```
# keytool -import -alias dkeservice -keystore dkestore.p12 -file dkeservice.cer
```

Provide the keystore password when prompted.

```
# keytool -import -alias dkeservice -keystore dkestore.p12 -file dkeservice.cer
Enter keystore password:
Certificate reply was installed in keystore
```

Figure 3 : Keytool: import signed certificate

7 Install DKE Anchor Manager and DKE Anchor Service

7.1 On Linux_2

To set up the DKE Anchor Manager and DKE Anchor Service on a Linux platform follow the these steps as a root user.

1. Create a folder `/utimaco/dke` under `/opt` .

›_ Console

```
# mkdir -p  /opt/utimaco/dke
```

2. Copy the `utimaco-dke-anchor-linux-1.0.2.tar` you file downloaded from the Utimaco Support portal to your Linux Server and go to this folder.
3. Extract the files to `/opt/utimaco/dke/` folder:

›_ Console

```
# tar -xvf utimaco-dke-anchor-linux-1.0.2.tar -C /opt/utimaco/dke/
```

4. Provide executable permission

›_ Console

```
# chmod 755 /opt/utimaco/dke/*
```

5. Move the `utimaco-dke-anchor.service` file to `/etc/systemd/system/` and reload the service daemon:

›_ Console

```
# mv /opt/utimaco/dke/utimaco-dke-anchor.service /etc/systemd/system/ #
systemctl daemon-reload
```

6. To start the Utimaco Anchor Service at system reboot, use the following command:

›_ Console

```
# systemctl enable utimaco-dke-anchor.service
```

7.2 On Windows_2

To set up the DKE Anchor Manager and DKE Anchor Service on Windows, follow these steps.

You must have Windows administrator rights to perform the installation.

1. Copy the `UtimacoDKEAnchor-1.0.2.msi` you downloaded from Utimaco Support portal to a appropriate location on the Windows Server.
2. Run `UtimacoDKEAnchor-1.0.2.msi` with administrative privileges. Double-click on `UtimacoDKEAnchor-1.0.2.msi` to start the installation process.
   The **Utimaco DKE Anchor Setup** dialog will be displayed.
3. Click on **Next**.
   To change the default installation path, click on the **Browse** button and choose the desired installation location.
   Click on **Next**.

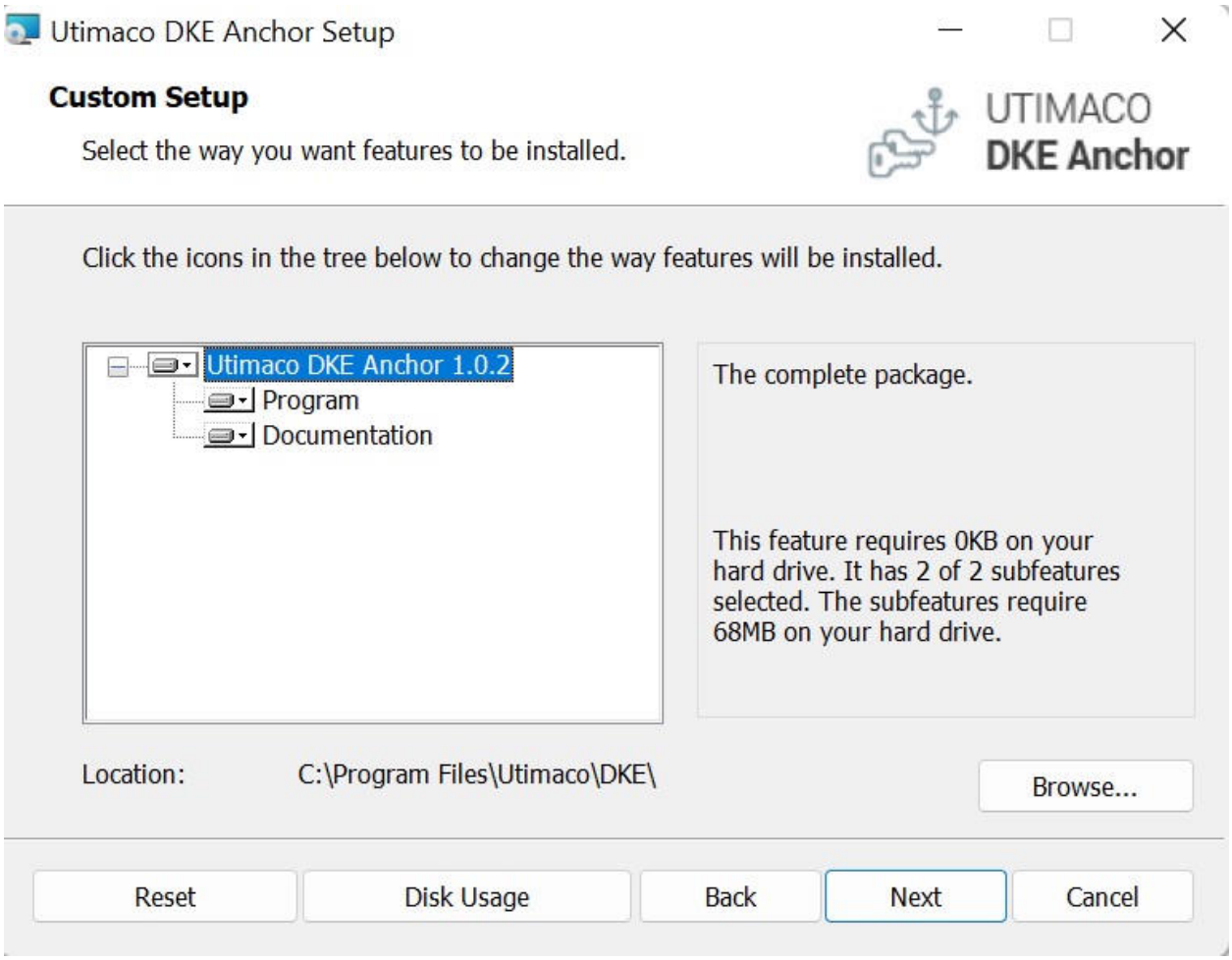


Figure 4 : Windows Custom Setup dialog

4. In the **Ready to install Utimao DKE Anchor** dialog, click on **Install** to start installation process. When you're prompted to allow this app from an unknown publisher, click on **Yes**.

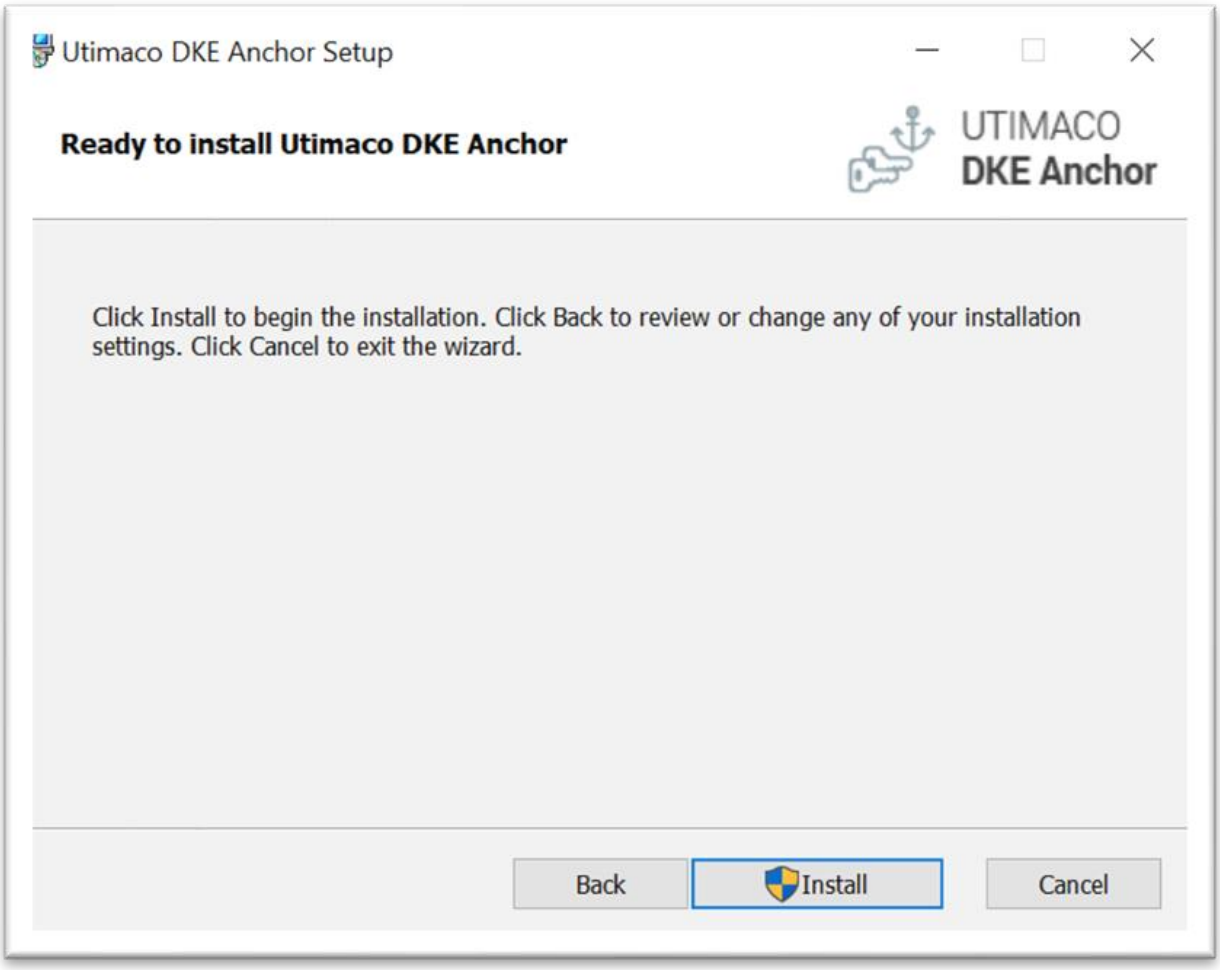


Figure 5 : Windows Ready to Install dialog

5. Click on **Finish** to complete the installation.

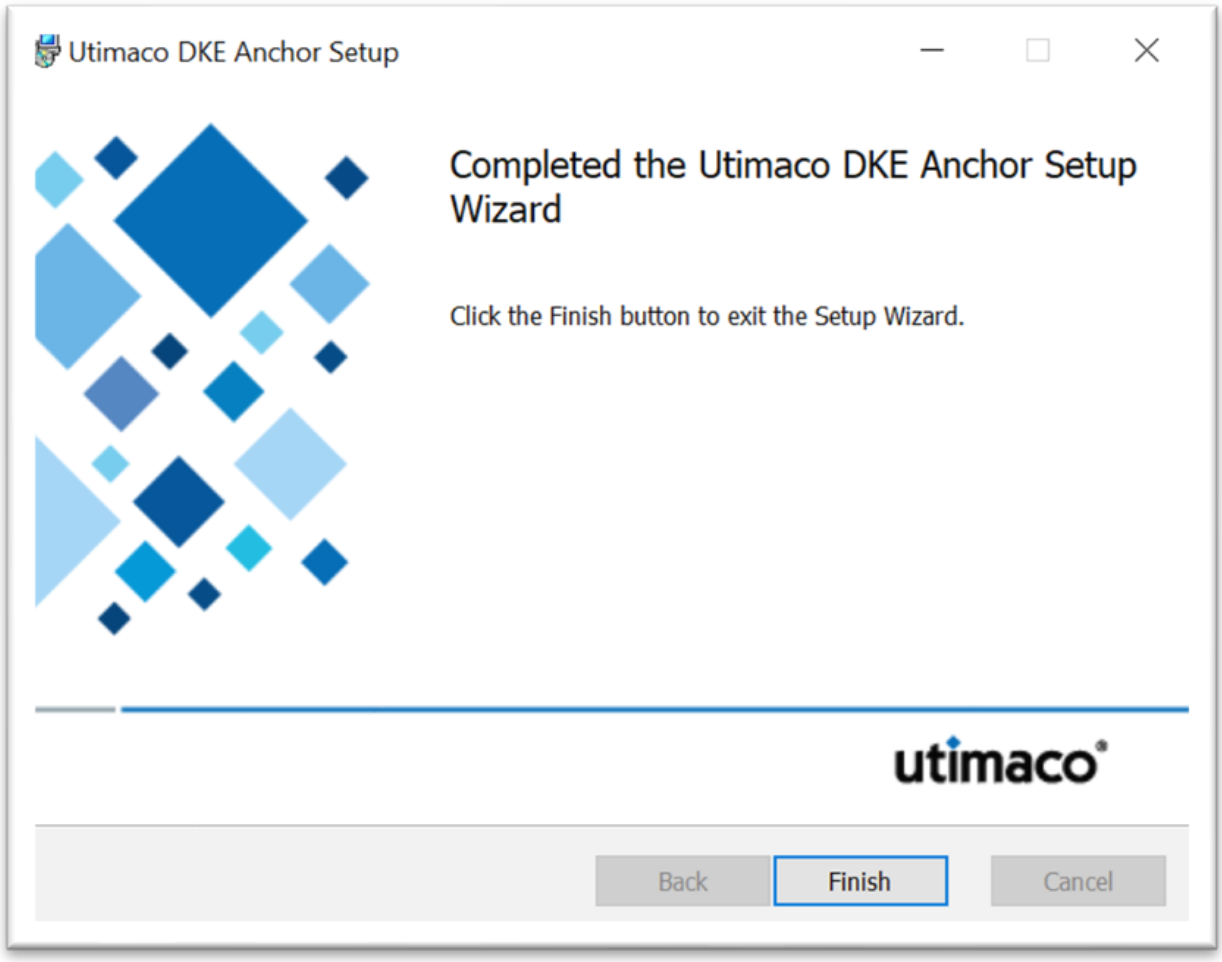


Figure 6 : Windows Completed Setup Wizzard dialog

Shortcuts on the Windows desktop and entries in the Windows Program Menu are created:

- Start Utimaco DKE Service
- Stop Utimaco DKE Service
- The **Installation guide** shortcut will only be available in the Program Menu if you selected the install option to add it.

8 Post Installation Configuration

8.1 PKCS12 KeyStore

Copy PKCS12 Keystore (refer Generate PKCS12 Keystore for WebServer) file to

Utimaco Installation folder as mentioned in Section 7.2, step 3 for windows and for Linux /opt/utimaco/dke

8.2 Configure the application.yml file

Update the "application.yml" in Utimaco Installation directory, which consist of following configuration parameters. Ensure that you have updated each parameter as per your environment.

Parameters	Value
`tenant-id`	Tenant ID for your azure account. You can locate your tenant ID by going to the Azure portal and viewing the tenant properties Default common
`validIssuers`	Locate the `validIssuers` setting and replace `<tenant-id>` with your tenant ID. If you want to enable external B2B access to your keystore, you will also need to include these external tenants as part of the valid issuers' list. See: https://techcommunity.microsoft.com/t5/security-compliance-andidentity/implement-dke-b2b-scenarios/ba-p/2193201
`jwtAudience`	DKE Anchor service URL without trailing slash

Parameters	Value
`pkcs11Lib`	The absolute path of pkcs11 library from cryptoserver Examples: ▪ Windows: `C:\\Utimaco\\SecurityServer\\Lib\\cs_pkcs11_R<2/3>.dll` ▪ Linux: `/opt/utimaco/lib/libcs_pkcs11_R<2/3>.so`
`slotId`	Change the pkcs11 slot accordingly Default: `0`
`pkcsHandleMaxPoolSize`	Maximum pool size for PKCS Handles Default 10
`userPIN`	The PKCS11 slot user password
`cacheDays`	Number of days for public key cache Default: `30`
`cacheDateTimeFormat`	Date time format for cache filed in public key api response Default: `yyyy-MM-dd'T'HH:mm:ss`
`key-store`	The keystore file in p12 format which is used for Tomcat Server
`key-store-password`	Password used to access the key store
`key-password`	Password used to access the key in the key store.

The sample `application.yml` file for Windows:

application.yml

```
---
azure:
   activedirectory:
       tenantid: "common"
       instance: "https://login.microsoftonline.com/"
       authority": "https://login.microsoftonline.com/common/v2.0"
       resource-server:
          principalClaimName: preferred_username
       tokenValidationParameters:
          validIssuers: ["https://sts.windows.net/
c7741b86-6a28-4c2f0-8c00-5aa7157d537b/"]
app:
    jwtAudience: "https://dkeservice.smv3q.onmicrosoft.com"
    allowedHosts: "*"
    jwtAuthorization: "https://login.windows.net/common/oauth2/authorize"
    userPrincipalAttribute: upn
    pkcs11Lib: " "C:\\Program
Files\\Utimaco\\SecurityServer\\Lib\\cs_pkcs11_R<2/3>.dll""
    slotId: 0
    pkcsHandleMaxPoolSize: 10
    userPIN: 123456
    cacheDays: 30
    cacheDateTimeFormat: yyyy-MM-dd'T'HH:mm:ss

server:
    port: 443
    ssl:
       key-store: dkeservice.p12
       Key-store-password: changeit
       key-store-type: pkcs12
       key-password: changeit
```

application.yml

```
spring:
    application:
       name: DKEService
    main:
       banner-mode: "off"
    mvc:
       log-request-details: false
logging:
  level:
       org.springframework.web: INFO
   file.name: dke-service.log
   pattern.file: "%d{${LOG_DATEFORMAT_PATTERN:yyyy-MM-dd HH:mm:ss.SSS}} $
{LOG_LEVEL_PATTERN:-%5p} ${PID:- } --- [%t] %-40.40logger{39} :
%m%n${LOG_EXCEPTION_CONVERSION_WORD:-%wEx{short}}"
```

Use cs_pkcs11_R2 or cs_pkcs11_R3 based on the version of the Utimaco PKCS#11 library. You can also modify slot id and PIN according to your requirement.

The sample `application.yml` file for Linux:

application.yml

```
---
azure:
    activedirectory:
        tenant-id: "common"
        instance: "https://login.microsoftonline.com/"
        authority": "https://login.microsoftonline.com/common/v2.0"
        resource-server:
            principalClaimName: preferred_username
               tokenValidationParameters:
            validIssuers: ["https://sts.windows.net/1f4beacd-b7aa-49b2-
aaa1-b8525cb257e0/"]
app:
    jwtAudience: "https://dkelinux.pls786.onmicrosoft.com"
    allowedHosts: "*"
    jwtAuthorization: "https://login.windows.net/common/oauth2/authorize"
    userPrincipalAttribute: upn
    pkcs11Lib: "/opt/utimaco/lib/libcs_pkcs11_R<2/3>.so"
    slotId: 0
    pkcsHandleMaxPoolSize: 10
    userPIN: 123456
    cacheDays: 30
    cacheDateTimeFormat: yyyy-MM-dd'T'HH:mm:ss

server:
    port: 443
    ssl:
        key-store: dkelinuxstore.p12
        key-store-password: 123456
        key-store-type: pkcs12
        key-password: 123456
```

application.yml

```
spring:
    application:
        name: DKEService
     main:
        banner-mode: "off"
    mvc:
        log-request-details: false
logging:
    level:
        org.springframework.web: INFO
    file.name: dke-service.log
    pattern.file: "%d{${LOG_DATEFORMAT_PATTERN:yyyy-MM-dd HH:mm:ss.SSS}} ${LOG_LEVEL_PATTERN:-%5p} ${PID:-} ---[%t] %-40.40logger{39} : %m%n${LOG_EXCEPTION_CONVERSION_WORD:-%wEx{short}}"
```

Use cs_pkcs11_R2 or cs_pkcs11_R3 based on the version of the Utimaco PKCS#11 library. You can also modify slot id and PIN according to your requirements.

9 Utimaco DKE Anchor Manager

In DKE, keys exist in several versions. When saving a DKE protected document the latest version is used. Since the version ID is stored with the document, documents protected with older versions can still be opened.

In Utimaco DKE access permissions can be managed for each version individually, but the commands also allow to change permissions for all versions of a key simultaneously. Currently, its only possible to set permissions for users individually, where users are identified by their email address in Azure AD.

9.1 Execute the DKE Anchor Manager on Linux

1. Execute the following command to list the available option of DKE Anchor Manager

›_ Console

```
# ./dkemanager.sh
```

```
[root@dkelinux dke]# ./dkemanager.sh
No command or more than one command given. Try 'help'.
Available commands:
key-new --name=<keyname>
  Creates new key with given keyname. Permissions are empty.
key-list
  Lists all available key names.
version-new --name=<keyname>
  Creates a new version for the key keyname and activates it.
  Permissions are copied from the currently active version.
version-list --name=<keyname>
  Lists all available versions of key keyname
permission-add --name=<keyname> [--version=<version>] --mailaddress=<address>
permission-remove --name=<keyname> [--version=<version>] --mailaddress=<address>
  Adds or removes permissions key keyname. If no version is given this affects
  all key versions. --version parameter can be repeated.
  --mailaddress can be repeated and adds/removes the given mail address to/from
  the permission list.
permission-list --name=<keyname> [--version=<version>]
  Lists the permissions for key keyname. If no version is given permissions
  of all versions of this key are listed.
[root@dkelinux dke]#
```

Figure 7 : Utimaco DKE Manager available options

2. Generate a Key.
   This will generate an RSA 2048-bit key, which will be used by DKE Anchor Service.

›_ Console

```
# ./dkemanager.sh key-new --name=<keyname>
```

```
Example
```

```
#./dkemanager.sh key-new --name=DKE_Key
```

```
[root@dkelinux dke]# ./dkemanager.sh key-new --name=DKE_Key
Key created.
```

Figure 8 : Utimaco DKE Key creation

3. List the generated keys.

›_ Console

```
# ./dkemanager.sh key-list
```

```
[root@dkelinux dke]# ./dkemanager.sh key-list
Found 1 key(s):
DKE_Key
[root@dkelinux dke]#
```

Figure 9 : dkemanager key-list output

4. Add Authorized Users to this key.

›_ Console

```
# ./dkemanager.sh permission-add --name=<keyname> [--version=<version>]
-mailaddress=<address>

# If you would like to add the users to all available versions of key,
don't pass the version parameter in the command

Example

# ./dkemanager.sh permission-add --name=DKE_Key
-mailaddress=testuser1@pls786.onmicrosoft.com

#./dkemanager.sh permission-add --name=DKE_Key
-mailaddress=testuser3@pls786.onmicrosoft.com

# If you would like to add the users to specific available versions of key,
pass the version parameter in the command

Example:

# ./dkemanager.sh permission-add --name=DKE_Key --version=a257be08-
e77b-48aa-

99bb-e5dac3b646c3 --mailaddress=testuser2@pls786.onmicrosoft.com
```

```
[root@dkelinux dke]# ./dkemanager.sh permission-add --name=DKE_Key --mailaddress=testuser1@pls786.onmicrosoft.com
Added 1 permission(s) to 1 of 1 key version(s).
[root@dkelinux dke]# ./dkemanager.sh permission-add --name=DKE_Key --mailaddress=testuser3@pls786.onmicrosoft.com
Added 1 permission(s) to 1 of 1 key version(s).
```

Figure 10 : Add authorized users to use the DKE_Key

```
[root@dkelinux dke]# ./dkemanager.sh permission-add --name=DKE_Key --version=a257be08-e77b-48aa-99bb-e5dac3b646c3 --mailaddres
s=testuser2@pls786.onmicrosoft.com
Added 1 permission(s) to 1 of 1 key version(s).
```

Figure 11 : Add authorized user to uses the DKE_Key with version

5. List Authorized Users for Key

›_ Console

```
# ./dkemanager.sh permission-list --name=<keyname> [--version=<version>]

# If you would like to list the users of all available versions of key,
don't pass the version parameter in the command

Example:

# ./dkemanager.sh permission-list --name=DKE_Key

# If you would like to list users of specific versions of key,

Example:

# ./dkemanager.sh permission-list --name=DKE_Key --version=a257be08-
e77b-48aa-

99bb-e5dac3b646c3
```

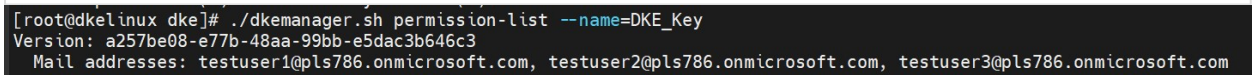


Figure 12 : List authorized users for DKE_Key

Figure 13 : List authorized user for unique version of DKE_Key

9.1.1 Initialize and Authorize Key for CP5

Skip this section if you are not using CP5 HSM.

In CP5 a private key cannot be used without being authorized. For authorizing a key, you first need to initialize the private key with an authorization key then set the authorization with the `AuthorizeKey` command.

1. List keys using the cxitool.

›_ Console

```
# cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 listkeys
```

```
[root@dkelinux dke]# cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 listkeys

idx algo  size type     group                    name                                spec
-------------------------------------------------------------------------------------
1   RAW   0    data     SLOT_0000                                                    1
2   RSA   2048 pub      SLOT_0000                                                    2
3   RSA   2048 prv      SLOT_0000                                                    3

[root@dkelinux dke]#
```

Figure 14 : cxitool listkeys output

Note down spec id for RSA private key. In above case it is 3.

2. Check the status of the key initialization using the cxitool.

›_ Console

```
# cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 group="SLOT_0000"
spec=3 Keyinfo
```

```
[root@dkelinux dke]# cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 group="SLOT_0000" spec=3 Keyinfo
Group          : SLOT_0000
Name           :
Specifier      : 3
Algo           : RSA
Size           : 2048
Export         : 0x00000000
  Extractable  : 0
  Sensitive    : 1
Usage          : 0x00000001
  Encrypt      : 0
  Decrypt      : 1
  Sign         : 0
  Verify       : 0
  Verify_Rec.  : 0
  Wrap         : 0
  Unwrap       : 0
  Derive       : 0
BlockLen       : 256
Type           : prv
DateGen        : 20230119
DateExp        :
Label          : DKE_Key
Modulus:
     0  d07d2284 3986bde7  62fdd34c 672cda2d | }" 9   b  Lg, ⊣
    10  7a77fb86 6c6280e1  0aefff94 0724f48d |zw  lb       $  |
    20  24f1519e a12b07fb  94856127 8be6eb7e |$ Q  +    a'   ~|
    30  271f1a27 862e519f  7002119a a5747732 |'   ' .Q p    tw2|
    40  becb7e16 6029552e  056566df 9dc38e79 |  ~ `)U. ef    y|
    50  b8237da1 98b49ae4  6eec7d99 5f5989f2 | #}     n } _Y  |
    60  60a4d652 cfcd7fdb  a05804a5 bf2e7b2e |`  R     X  .{.|
    70  eab5ec95 efc6ab77  412f2136 47954bc1 |       wA/!6G K |
    80  7f1eddd5 196f70f5  1e915031 3a09256d |     op   P1: %m|
    90  76a929fa ff59c39c  d745ffd6 e8b080f7 |v )  Y   E      |
    a0  c22a6b04 7fdc6e75  e248c106 4bd85b90 | *k   nu H  K [ |
    b0  e48ec590 f1726809  be8ec39d e75729ac |     rh      W) |
    c0  1fcde2fb eee4421e  ed45a8b9 e74a3b1b |      B  E   J; |
    d0  8a546cdf 4c285ad0  bd0b8363 080b03ab | Tl L(Z    c    |
    e0  62a1d00a 1f2638aa  b5debe9d db41ded4 |b    &8      A  |
    f0  189448ba ea0f67d6  7300b807 a73379a7 |  H   g s    3y |
Exponent:
     0  010001                              |               |
Initialized    : False
```

Figure 15 : Key initialized status

3. The initialized status is false.
   Run the command below to generate a user authentication key file `ka.key`, which contains an RSA key pair of the speciied size. This authentication key will be used to initialize the DKE_Key in the next step.

›_ Console

```
# ./csadm genkey=ka.key,2048,"USR_0000" generating RSA key: ka.key, 2048 bits, owner: USR_0000
```

4. Initialize the DKE_Key with the `ka.key` file using the cxitool.

›_ Console

```
# cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 group="SLOT_0000"
Spec=3 KeyFile=ka.key InitializeKey
```

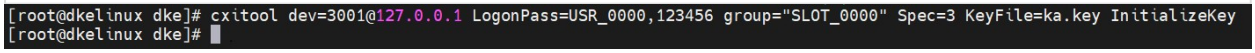


Figure 16 : Initialize the DKE_Key with the ka.key file

5. Authorize DKE_Key with the Authentication Key file `ka.key`.

›_ Console

```
#cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 group="SLOT_0000"
Spec=3 KeyFile=ka.key AuthorizeKey=1000000
```

6. Check the initialization status of keys.

›_ Console

```
# cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 group="SLOT_0000"
spec=3 Keyinfo
```

```
[root@dkelinux dke]# cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 group="SLOT_0000" spec=3 Keyinfo
Group          : SLOT_0000
Name           :
Specifier      : 3
Algo           : RSA
Size           : 2048
Export         : 0x00000000
  Extractable  : 0
  Sensitive    : 1
Usage          : 0x00000001
  Encrypt      : 0
  Decrypt      : 1
  Sign         : 0
  Verify       : 0
  Verify_Rec.  : 0
  Wrap         : 0
  Unwrap       : 0
  Derive       : 0
BlockLen       : 256
Type           : prv
DateGen        : 20230119
DateExp        :
Label          : DKE_Key
Modulus:
     0  d07d2284 3986bde7  62fdd34c 672cda2d | }" 9   b  Lg, ⊣
    10  7a77fb86 6c6280e1  0aefff94 0724f48d |zw  lb       $  |
    20  24f1519e a12b07fb  94856127 8be6eb7e |$ Q  +    a'   ~|
    30  271f1a27 862e519f  7002119a a5747732 |'  ' .Q p    tw2|
    40  becb7e16 6029552e  056566df 9dc38e79 |  ~ `)U. ef    y|
    50  b8237da1 98b49ae4  6eec7d99 5f5989f2 | #}     n } _Y  |
    60  60a4d652 cfcd7fdb  a05804a5 bf2e7b2e |`  R     X   .{.|
    70  eab5ec95 efc6ab77  412f2136 47954bc1 |       wA/!6G K |
    80  7f1eddd5 196f70f5  1e915031 3a09256d |     op   P1: %m|
    90  76a929fa ff59c39c  d745ffd6 e8b080f7 |v )  Y   E      |
    a0  c22a6b04 7fdc6e75  e248c106 4bd85b90 | *k   nu H  K [ |
    b0  e48ec590 f1726809  be8ec39d e75729ac |     rh      W) |
    c0  1fcde2fb eee4421e  ed45a8b9 e74a3b1b |      B  E   J; |
    d0  8a546cdf 4c285ad0  bd0b8363 080b03ab | Tl L(Z    c    |
    e0  62a1d00a 1f2638aa  b5debe9d db41ded4 |b    &8      A  |
    f0  189448ba ea0f67d6  7300b807 a73379a7 |  H   g s    3y |
Exponent:
     0  010001                              |               |
Initialized    : True
Assigned       : True
Protocol       : PSS
Auth Failures  : 0
Remaining Ops  : 1000000
[root@dkelinux dke]#
```

Figure 17 : Initialization status of keys

The initialized status is true.

9.1.2 Generate a New Version of a Key for CP5

Skip this section if you are not using CP5 HSM.

1. List all previous versions of a DKE_Key.

›_ Console

```
# ./dkemanager.sh version-list --name=<keyname>
```

```
Example
```

```
# ./dkemanager.sh version-list --name=DKE_Key
```

```
[root@dkelinux dke]# ./dkemanager.sh version-list --name=DKE_Key
Version UUID (* active)                Creation Date
------------------------------------------------------
a257be08-e77b-48aa-99bb-e5dac3b646c3 * 2023-01-18
```

Figure 18 : List version of DKE_Key

2. Generate a new version of a key.

›_ Console

```
# ./dkemanager.sh version-new --name=<keyname>
```

```
Example
```

```
./dkemanager.sh version-new --name=DKE_Key
```

```
[root@dkelinux dke]# ./dkemanager.sh version-new --name=DKE_Key
New key version created and activated. UUID: eb40b00a-c5d2-4b22-9c3f-c0f4761ac6db
```

Figure 19 : List version of DKE_Key

3. List the updated versions of a key.

›_ Console

```
# ./dkemanager.sh version-list --name=<keyname>
```

Example

```
[root@dkelinux dke]# ./dkemanager.sh version-list --name=DKE_Key
```

```
[root@dkelinux dke]# ./dkemanager.sh version-list --name=DKE_Key
Version UUID (* active)                  Creation Date
-------------------------------------------------------
a257be08-e77b-48aa-99bb-e5dac3b646c3   2023-01-18
eb40b00a-c5d2-4b22-9c3f-c0f4761ac6db * 2023-01-18
```

Figure 20 : List all versions of DKE_Key

4. Reinitialize and reauthorize the new version of a key for CP5 by following the steps in Initialize and Authorize Key for CP5.

For CP5: If you have generated a new version of the key, you need to initialize and authorize this new version of the key again by following the steps in Initialize and Authorize Key for CP5.

9.1.3 Delete a User for a Key

If you want to delete a user for a key, run the following command:

›_ Console

```
# ./dkemanager.sh permission-remove --name=<keyname> [--version=<version>]
-mailaddress=<address>

# If you would like to delete the user of all available versions of key,

Example:

# ./dkemanager.sh permission-remove --name=DKE_Key --mailaddress=
testuser1@pls786.onmicrosoft.com

# If you would like to delete the user of specific versions of key,

Example:

# ./dkemanager.sh  permission-remove --name=DKE_Key

  --version= a257be08-e77b-48aa-99bb-e5dac3b646c3

  --mailaddress= testuser3@pls786.onmicrosoft.com
```

```
[root@dkelinux dke]#  ./dkemanager.sh permission-remove --name=DKE_Key --mailaddress=testuser1@pls786.onmicrosoft.com
Removed 2 permission(s) from 2 of 2 key version(s).
```

Figure 21 : Deleting the user permission from a DKE_Key

[root@dkelinux dke]# ./dkemanager.sh permission-remove --name=DKE_Key --mailaddress=testuser1@pls786.onmicrosoft.com
Removed 2 permission(s) from 2 of 2 key version(s).

Figure 22 : Deleting the user permission from a DKE_Key with specific version

9.2 Execute the DKE Anchor Manager on Windows

1. Run the following command:

›_ Console

```
C:\Program Files\Utimaco\DKE>cd <Utimaco install directory> C:\Program
Files\Utimaco\DKE>dkemanager.bat
```

```
C:\Program Files\Utimaco\DKE>dkemanager.bat help
Available commands:
key-new --name=<keyname>
Creates new key with given keyname. Permissions are empty.
key-list
Lists all available key names.
version-new --name=<keyname>
Creates a new version for the key keyname and activates it.
Permissions are copied from the currently active version.
version-list --name=<keyname>
Lists all available versions of key keyname
permission-add --name=<keyname> [--version=<version>] --mailaddress=<address>
permission-remove --name=<keyname> [--version=<version>] --mailaddress=<address>
Adds or removes permissions key keyname. If no version is given this affects
all key versions. --version parameter can be repeated.
--mailaddress can be repeated and adds/removes the given mail address to/from
the permission list.
permission-list --name=<keyname> [--version=<version>]
Lists the permissions for key keyname. If no version is given permissions
of all versions of this key are listed.
```

Figure 23 : Utimaco DKE Manager available options

2. Generate a key:
   This will generate an RSA 2048 bit key, that will be used by the DKE Anchor Service.

›_ Console

```
C:\Program Files\Utimaco\DKE>dkemanager.bat key-new --name=<keyname>
```

```
Example
```

```
 C:\Program Files\Utimaco\DKE>dkemanager.bat key-new --name=DKE_Key
```

```
C:\Program Files\Utimaco\DKE>dkemanager.bat key-new --name=DKE_Key
Key created.
C:\Program Files\Utimaco\DKE>
```

Figure 24 : Generating a DKE_key

3. List the generated keys:

›_ Console

```
C:\Program Files\Utimaco\DKE>dkemanager.bat key-list
```

```
C:\Program Files\Utimaco\DKE>dkemanager.bat key-list
Found 1 key(s):
DKE_Key
C:\Program Files\Utimaco\DKE>
```

Figure 25 : Utimaco DKE Manager available options

4. Add authorized user to a key:

›_ Console

```
C:\Program Files\Utimaco\DKE>dkemanager.bat permission-add --name=<keyname>
[-version=<version>] --mailaddress=<address>


# If you would like to add the users to all available versions of key,
don't pass the version parameter in the command

Example:

C:\Program Files\Utimaco\DKE>dkemanager.bat permission-add --name=DKE_Key
-mailaddress=testuser1@pls786.onmicrosoft.com

C:\Program Files\Utimaco\DKE>dkemanager.bat permission-add --name=DKE_Key
-mailaddress=testuser3@pls786.onmicrosoft.com


# If you would like to add the users to specific available versions of key,
pass the version parameter in the command

Example:

C:\Program Files\Utimaco\DKE>dkemanager.bat permission-add --name=DKE_Key
-version=4f420b97-7a81-4340-ab4a-55f973262c47 --mailaddress=
testuser2@pls786.onmicrosoft.com
```

```
C:\Program Files\Utimaco\DKE>dkemanager.bat permission-add --name=DKE_Key --mailaddress=testuser1@pls786.onmicrosoft.com
Added 1 permission(s) to 1 of 1 key version(s).
C:\Program Files\Utimaco\DKE>
C:\Program Files\Utimaco\DKE>dkemanager.bat permission-add --name=DKE_Key --mailaddress=testuser3@pls786.onmicrosoft.com
Added 1 permission(s) to 1 of 1 key version(s).
C:\Program Files\Utimaco\DKE>
```

Figure 26 : Add authorized user to use the DKE_Key

```
C:\Program Files\Utimaco\DKE> dkemanager.bat permission-add --name=DKE_Key --version=4f420b97-7a81-4340-ab4a-55f973262c47 --mailaddress=
testuser2@pls786.onmicrosoft.com
Added 1 permission(s) to 1 of 1 key version(s).
C:\Program Files\Utimaco\DKE>
```

Figure 27 : Add authorized user to use the DKE_Key with version

5. List authorized user for a key:

›_ Console

```
C:\Program Files\Utimaco\DKE>dkemanager.bat permission-list --name=<keyname> [--version=<version>]

# If you would like to list the users of all available versions of key, don't pass the version parameter in the command

Example:

C:\Program Files\Utimaco\DKE>dkemanager.bat permission-list --name=DKE_Key

# If you would like to list users of specific versions of key,  Example:


C:\Program Files\Utimaco\DKE>dkemanager.bat permission-list --name=DKE_Key -version=4f420b97-7a81-4340-ab4a-55f973262c47
```

```
C:\Program Files\Utimaco\DKE>dkemanager.bat permission-list --name=DKE_Key
Version: 4f420b97-7a81-4340-ab4a-55f973262c47
Mail addresses: testuser1@pls786.onmicrosoft.com , testuser2@pls786.onmicrosoft.com, testuser3@pls786.onmicrosoft.com
C:\Program Files\Utimaco\DKE>
```

Figure 28 : List authorized user for DKE_Key

9.2.1 Initialize and Authorize Keys for CP5

Skip this section if you are not using CP5 HSM.

In CP5 a private key cannot be used without being authorized. For authorizing a key, you first need to initialize the private key with an authorization key, then set the authorization with the `AuthorizeKey` command.

1. List Keys using the cxitool:

›_ Console

```
C:\Program Files\Utimaco\DKE>cxitool dev=3001@127.0.0.1
LogonPass=USR_0000,123456 listkeys
```

```
C:\Program Files\Utimaco\DKE> cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 listkeys
idx algo  size type     group         name            spec
-----------------------------------------------------------
1   RAW   0    data     SLOT_0000                     1
2   RSA   2048 pub      SLOT_0000                     2
3   RSA   2048 prv      SLOT_0000                     3
C:\Program Files\Utimaco\DKE>
```

Figure 29 : cxitool listkeys output

Note down the spec id for RSA private key. In our example it is 3.

2. Check Status of Key Initialization using cxitool:

›_ Console

```
C:\Program Files\Utimaco\DKE>cxitool dev=3001@127.0.0.1

LogonPass=USR_0000,123456 group="SLOT_0000" spec=3 Keyinfo
```

```
C:\Program Files\Utimaco\DKE>cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 group="SLOT_0000" spec=3
Keyinfo
Group         : SLOT_0000
Name          :
Specifier     : 3
Algo          : RSA
Size          : 2048
Export        : 0x00000000
  Extractable : 0
  Sensitive   : 1
Usage         : 0x00000001
  Encrypt     : 0
  Decrypt     : 1
  Sign        : 0
  Verify      : 0
  Verify_Rec. : 0
  Wrap        : 0
  Unwrap      : 0
  Derive      : 0
BlockLen      : 256
Type          : prv
DateGen       : 20221116
DateExp       :
Label         : DKE_Key
Modulus:
     0  cc6f920e 87e64ed3  717ae5a2 75239461 | o    N qz  u# a|
    10  602ff551 d6e7276c  a99c118c b4993fce |`/ Q  'l      ? |
    20  ff9dcc09 5a5fe2b2  e3ecb381 1cbba584 |    Z_          |
    30  bc4ddf5f 413d79cf  831c2b5b 2937e038 | M _A=y   +[)7 8|
    40  05009072 4d28e2ea  25010da7 ebf10504 |   rM(  %       |
    50  7481dd0e 5e8b8ab3  edb14b53 9e3383b2 |t   ^     KS 3  |
    60  f407d3cb 75bb8039  a73899b8 68adc71c |    u  9 8  h   |
    70  8e6c0c33 76e9ca16  d118a85c 85d3313a | l 3v      \  1:|
    80  ac7c5ded f05af546  aaa00959 b52aa359 | |]  Z F   Y * Y|
    90  587e6ae2 dae674e3  e3707ccd 81b9b4eb |X~j   t  p|     |
    a0  a44bd77e ae8a9022  60589362 07eaf016 | K ~   "`X b    |
    b0  97ad91f8 d76b1872  75ce3bd6 cf590e20 |     k ru ;  Y  |
    c0  3f538bd3 2e3dc7eb  b26004f6 1029ab75 |?S  .=   `   ) u|
    d0  f5e4d9cf 2e8046f0  fcf8a826 060c41aa |    . F    &  A |
    e0  26a95577 b3614a2d  29ef11e2 6af2c222 |& Uw aJ-)   j  "|
    f0  0c3d28f0 0c6f0204  d0fc3e1f 0b9e0f49 | =(  o    >    I|
Exponent:
     0  010001                              |               |
Initialized   : False
```

Figure 30 : Key initialized status

As you can see key initialized status is false.

3. Run the command below to generate the user authentication key file ka.key, that contains an RSA key pair of the given size. This authentication key will be used to initialize the DKE_Key in the next step.

›_ Console

```
C:\Program Files\Utimaco\DKE>csadm genkey=ka.key,2048,"USR_0000" generating
RSA key: ka.key, 2048 bits, owner: USR_0000
```

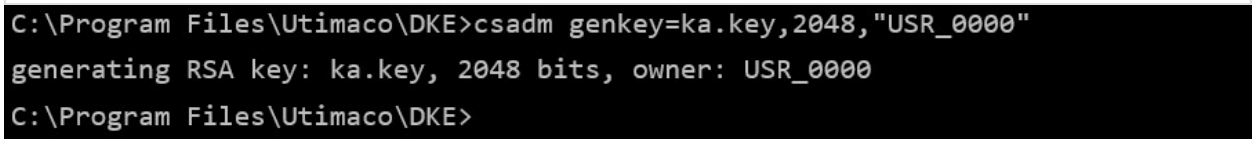


Figure 31 : cxitool listkeys output

4. Initialize the DKE_Key with ka.key file using the cxitool:

›_ Console

```
C:\Program Files\Utimaco\DKE>cxitool dev=3001@127.0.0.1

LogonPass=USR_0000,123456 group="SLOT_0000" Spec=3 KeyFile=ka.key
InitializeKey
```

Figure 32 : Initialize the DKE_Key with the ka.key file

5. Authorize the DKE_Key with the authentication key file ka.key:

›_ Console

```
C:\Program Files\Utimaco\DKE>cxitool dev=3001@127.0.0.1

LogonPass=USR_0000,123456 group="SLOT_0000" Spec=3 KeyFile=ka.key
AuthorizeKey=1000000
```

Figure 33 : cxitool listkeys output

6. Check the initialization status of the key:

›_ Console

```
C:\Program Files\Utimaco\DKE>cxitool dev=3001@127.0.0.1

LogonPass=USR_0000,123456 group="SLOT_0000" spec=3 Keyinfo
```

```
     0  cc6f920e 87e64ed3  717ae5a2 75239461 | o    N qz  u# a|
    10  602ff551 d6e7276c  a99c118c b4993fce |`/ Q  'l     ? |
    20  ff9dcc09 5a5fe2b2  e3ecb381 1cbba584 |    Z_         |
    30  bc4ddf5f 413d79cf  831c2b5b 2937e038 | M _A=y   +[)7 8|
    40  05009072 4d28e2ea  25010da7 ebf10504 |   rM(  %      |
    50  7481dd0e 5e8b8ab3  edb14b53 9e3383b2 |t   ^     KS 3  |
    60  f407d3cb 75bb8039  a73899b8 68adc71c |    u  9 8  h   |
    70  8e6c0c33 76e9ca16  d118a85c 85d3313a | l 3v      \  1:|
    80  ac7c5ded f05af546  aaa00959 b52aa359 | |]  Z F   Y * Y|
    90  587e6ae2 dae674e3  e3707ccd 81b9b4eb |X~j   t  p|     |
    a0  a44bd77e ae8a9022  60589362 07eaf016 | K ~   "`X b    |
    b0  97ad91f8 d76b1872  75ce3bd6 cf590e20 |     k ru ;  Y  |
    c0  3f538bd3 2e3dc7eb  b26004f6 1029ab75 |?S  .=   `   ) u|
    d0  f5e4d9cf 2e8046f0  fcf8a826 060c41aa |    . F    &  A |
    e0  26a95577 b3614a2d  29ef11e2 6af2c222 |& Uw aJ-)   j  "|
    f0  0c3d28f0 0c6f0204  d0fc3e1f 0b9e0f49 | =(  o    >    I|
Exponent:
     0  010001                               |               |
Initialized   : True
Assigned      : True
Protocol      : PSS
Auth Failures : 0
Remaining Ops : 1000000
C:\Program Files\Utimaco\DKE>
```

Figure 34 : Initialization status of the key

The key initialized status is true.

9.2.2 Generating a New Version of a Key for CP5

Skip this section if you are not using CP5 HSM.

1. List all the previous versions of the DKE_Key:

›_ Console

```
C:\Program Files\Utimaco\DKE>dkemanager.bat version-list --name=<keyname>
```

```
Example
```

```
C:\Program Files\Utimaco\DKE>dkemanager.bat version-list --name=DKE_Key
```

```
C:\Program Files\Utimaco\DKE> dkemanager.bat version-list --name=DKE_Key
Version UUID (* active)                Creation Date
-------------------------------------------------
4f420b97-7a81-4340-ab4a-55f973262c47 * 2022-11-16
C:\Program Files\Utimaco\DKE>
```

Figure 35 : List the version of a DKE_Key

2. Generate a new version of DKE_key with below command:

›_ Console

```
C:\Program Files\Utimaco\DKE>dkemanager.bat version-new --name=<keyname>
```

```
Example
```

```
C:\Program Files\Utimaco\DKE>dkemanager.bat version-new --name=DKE_Key
```

```
C:\Program Files\Utimaco\DKE>dkemanager.bat version-new --name=DKE_Key
New key version created and activated. UUID: 80332081-92c1-46d7-81f2-16e650369b33
C:\Program Files\Utimaco\DKE>
```

Figure 36 : List the version of a DKE_Key

3. List the Updated Versions of the Key:

›_ Console

```
C:\Program Files\Utimaco\DKE>dkemanager.bat version-list --name=<keyname>
```

Example

```
C:\Program Files\Utimaco\DKE>dkemanager.bat version-list --name=DKE_Key
```

```
C:\Program Files\Utimaco\DKE> dkemanager.bat version-list --name=DKE_Key
Version UUID (* active)                    Creation Date
-------------------------------------------------
4f420b97-7a81-4340-ab4a-55f973262c47       2022-02-22
80332081-92c1-46d7-81f2-16e650369b33 *     2022-02-22
C:\Program Files\Utimaco\DKE>
```

Figure 37 : List the version of a DKE_Key

1. Reinitialize and re-authorize the new version of the key for CP5 by following the steps in Initialize and Authorize Keys for CP5.

For CP5: if you have generated a new version of the key, you need to initialize and authorize this new version again by following the steps in Initialize and Authorize Keys for CP5.

9.2.3 Delete a User for a Key

To delete a user for a key, run the following command:

›_ Console

```
C:\Program Files\Utimaco\DKE> dkemanager.bat permission-remove
-name=<keyname> [--version=<version>] --mailaddress=<address>

Example

# If you would like to delete the user of all available versions of key,

C:\Program Files\Utimaco\DKE>dkemanager.bat permission-remove --
name=DKE_Key -mailaddress= testuser1@pls786.onmicrosoft.com


# If you would like to delete the user of specific versions of key,

C:\Program Files\Utimaco\DKE>dkemanager.bat  permission-remove --
name=DKE_Key

--version=4f420b97-7a81-4340-ab4a-55f973262c47

--mailaddress= testuser3@pls786.onmicrosoft.com
```

```
C:\Program Files\Utimaco\DKE>dkemanager.bat permission-remove --name=DKE_Key --mailaddress=testuser1@pls786.onmicrosoft.com
Removed 2 permission(s) from 2 of 2 key version(s).
C:\Program Files\Utimaco\DKE>
```

Figure 38 : Deleting user permission from the DKE_Key

```
C:\Program Files\Utimaco\DKE>dkemanager.bat  permission-remove --name=DKE_Key --version=4f420b97-7a81-4340-ab4a-55f973262c47
--mailaddress=testuser3@pls786.onmicrosoft.com
Removed 1 permission(s) from 1 of 1 key version(s).
C:\Program Files\Utimaco\DKE>
```

Figure 39 : Deleting user permission from the DKE_Key

10 Utimaco DKE Anchor Service_2

10.1 Set the Environment Variable

10.1.1 On Linux

1. Generate the HSMAuth key:

›_ Console

```
# csadm dev=3001@127.0.0.1 GetHSMAuthKey > /opt/utimaco/bin/HSMauth.key
```

2. Edit / `etc/systemd/system/utimaco-dke-anchor.service` file and add following entries under service section.

›_ Console

```
[Service]

Environment="CS_AUTH_KEYS=/opt/utimaco/bin/HSMauth.key"
```

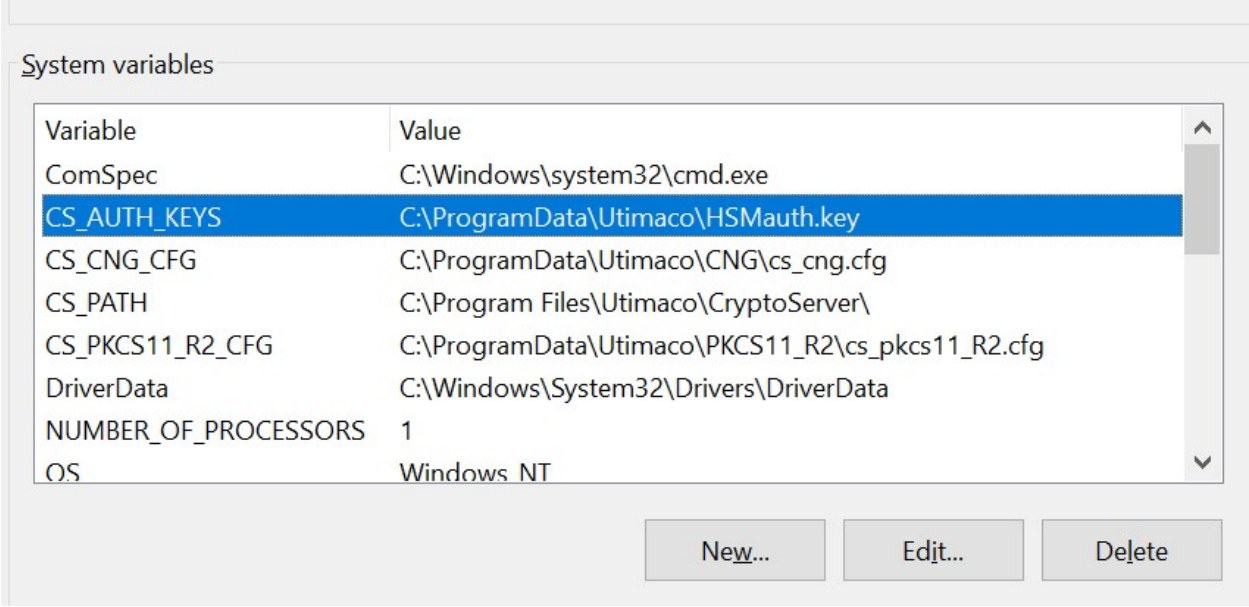
10.1.2 On Windows

1. Generate the HSMAuth key:

›_ Console

```
C:\Program Files\Utimaco\DKE>csadm dev=3001@127.0.0.1 GetHSMAuthKey > C:
\ProgramData\Utimaco\HSMauth.key
```

2. Set the HSMAuth key path under system variable as below:



10.2 Set the Environment Variable for Java

10.2.1 On Linux

Edit / `etc/systemd/system/utimaco-dke-anchor.service` file and add following entries under service section.

›_ Console

```
[Service]
Environment="JAVA_HOME=/opt/jdk-11.0.13"
Environment="PATH=/opt/jdk-11.0.13/bin:$PATH"
```

10.2.2 On Windows

Set the java environment variable:

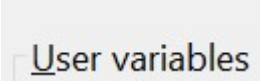
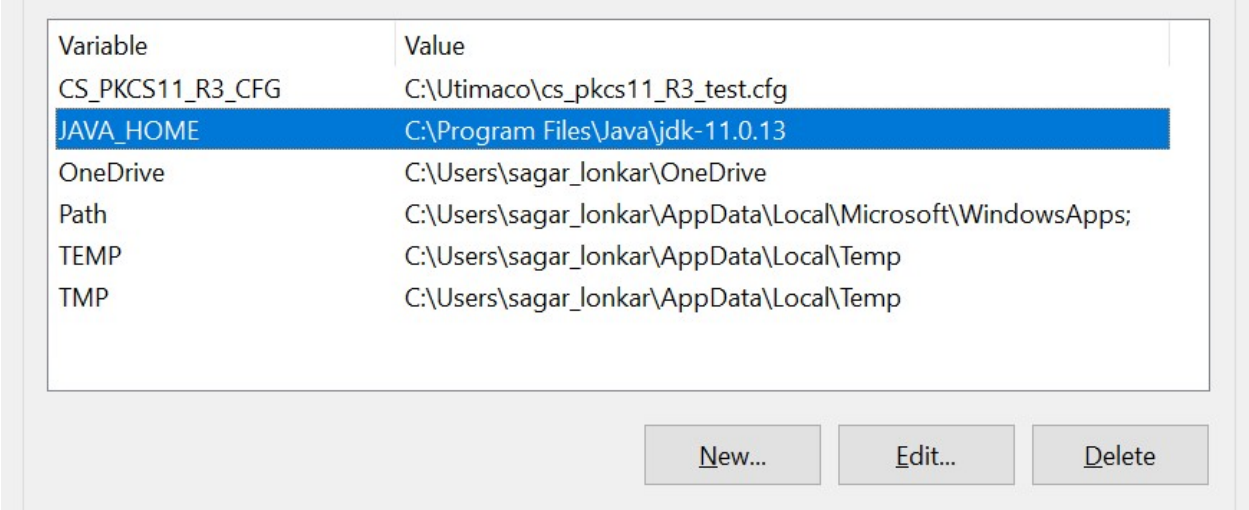


Figure 40 : Java home path



10.3 Start the DKE Anchor Service

10.3.1 On Linux

To Start Utimaco DKE Anchor Service, run the following command:

›_ Console

```
# systemctl start utimaco-dke-anchor.service
```

To Verify the status of the Utimaco DKE Anchor Service, run the following command:

›_ Console

```
# systemctl status utimaco-dke-anchor.service
```

```
[root@dkelinux dke]#  systemctl status utimaco-dke-anchor.service
● utimaco-dke-anchor.service - Utimaco DKE Anchor Service
   Loaded: loaded (/etc/systemd/system/utimaco-dke-anchor.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2022-03-23 13:32:20 UTC; 4s ago
 Main PID: 21504 (dkeservice.sh)
    Tasks: 24 (limit: 50737)
   Memory: 335.9M
   CGroup: /system.slice/utimaco-dke-anchor.service
           ├─21504 /bin/bash /opt/utimaco/dke/dkeservice.sh
           └─21505 java -jar dke-service-1.0.0.jar
```

Figure 41 : Utimaco Anchor Service Status

10.3.2 On Windows

Click the **Start Utimaco DKE Anchor Service** shortcut from desktop or from the start menu.
To verify if the Utimaco DKE Anchor service is running, verify the status of **Utimaco DKE Anchor** in Windows Services as follows:

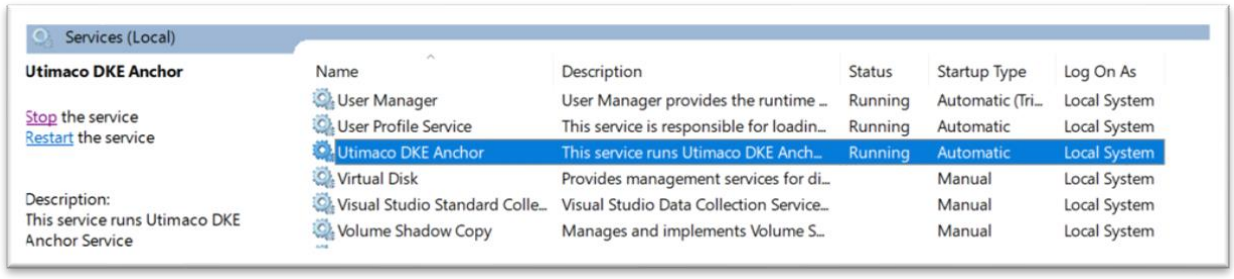


Figure 42 : Windows Services

Also check the dke-service.log in the Utimaco installation location as follows:

Figure 43 : DKE Anchor service log

10.4 Stop the DKE Anchor Service

10.4.1 On Linux

To stop the Utimaco DKE Anchor Service, run the following command:

›_ Console

```
# systemctl stop utimaco-dke-anchor.service
```

Client connectivity between DKE Anchor service and Azure AD:
For acquiring the public key and for decrypting existing keys, clients need to be able to reach the DKE Anchor service. To allow authentication, clients also require access to Azure AD. Ensure that required ports are enabled.

10.4.2 On Windows

Click on the **Stop Utimaco DKE Service** shortcut on the desktop or from the start menu.

10.5 Logging

When Utimaco the DKE Anchor Service & Manager start, they create the following log files in the installation directory:

- `dke-service.log:` Application log file for DKE Service with all information & error log
- `dke-service-audit.log:` Audit log file for DKE Service
- `dke-manager.log:` Application log file for DKE ;anager with all information & error log
- `dke-manager-audit.log:` Audit log file for DKE Manager

Logs are rotated on daily basis within same directory with file name like, `dke-service.log.2022-04-01.0.gz`.

11 Azure DKE Configurations

11.1 Register DKE Anchor Service with Azure

The following steps enable you to register your DKE Anchor Service. Registering your DKE Anchor Service is the last step in deploying DKE before you can start creating labels.

To register the DKE Anchor Service:

1. In your browser, open the Microsoft Azure portal, and go to **All Services** > **Identity** > **App registrations**.
2. Select **New registration** and enter a meaningful name.
3. Select an account type. Either single or multi-tenant.
   Example:

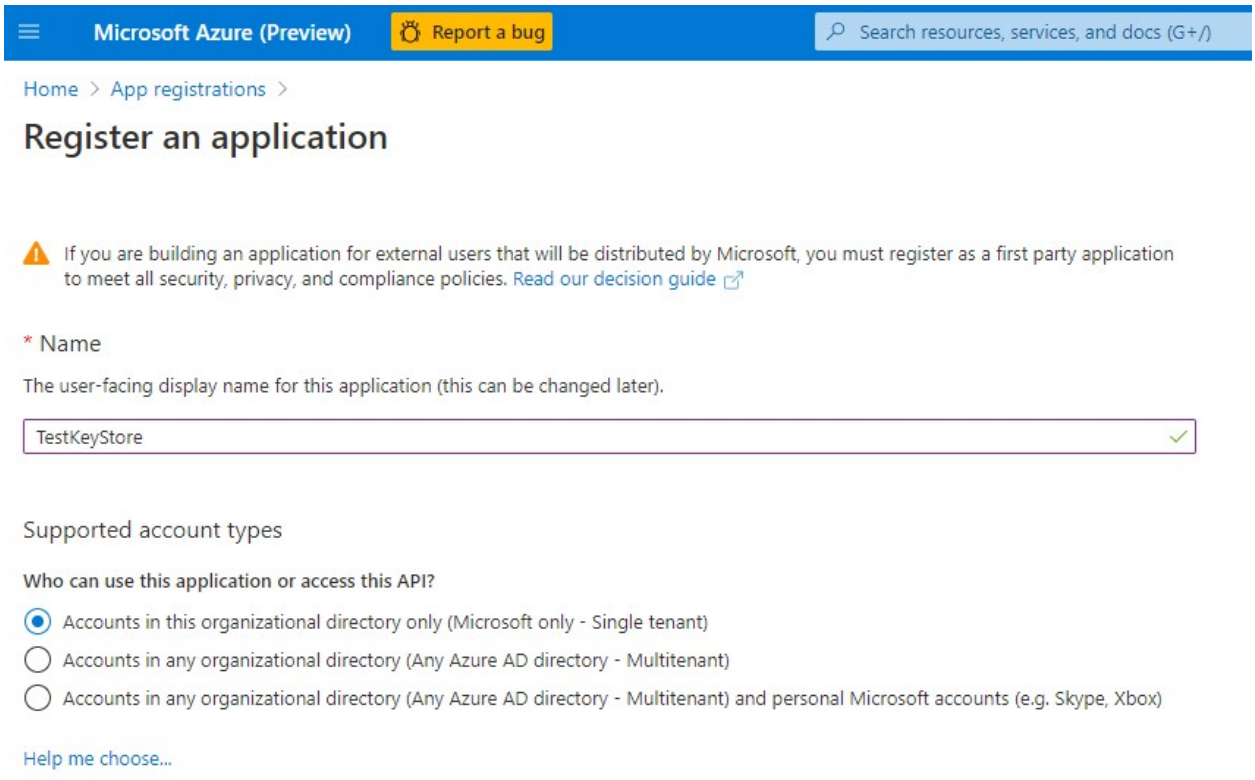


Figure 44 : Register an application

4. At the bottom of the page, select **Register** to create the new App registration.
5. In your new **App registration**, in the left pane, under **Manage**, select **Authentication**.

6. Select **Add a platform**.
7. On the Configure platforms popup, select **Web**.
8. Under Redirect URIs, enter the URI of your double key encryption service. Enter the App Service URL, including both the hostname and domain and click on **Configure** button.
9. For example: https://dkeservice.utimaco.com.

The DKE Anchor Service URL needs to be based on the root level, i.e., not a sub directory. Working example: https://dkeservice.utimaco.com/mykey
This example does not work: https://webserver.utimaco.com/dkeservice/mykey

The DKE URL must not contain a trailing slash:
Working example: https://dkeservice.utimaco.com/mykey
This example does not work: https://dkeservice.utimaco.com/mykey/

10. Under **Implicit grant**, select the **ID tokens** checkbox.
11. Select **Save** to save your changes.
12. On the left pane, select **Expose an API**, next to **Application ID URI**, click on **Set** and enter your app service URL, including both hostname and domain and click on **Save**.
13. Still on the **Expose an API** page, in the Scopes defined by this API area, select **Add a scope**.
    In the new scope:
    - Define the scope name as **user_impersonation**
    - Select the administrators and users who can consent
    - Define any remaining values required
    - Select Select **Save** at the top to save your changes
14. Still on the **Expose an API** page, in the **Authorized client applications** area, select **Add a client application**.
    In the new client application:
    - Define the Client ID as d3590ed6-52b3-4102-aeff-aad2292ab01c.
      This value is the Microsoft Office client ID and enables Office to obtain an access token for your key store.
    - Under **Authorized scopes**, select the **user_impersonation** scope.
    - Select **Save** at the top to save your changes

- Repeat these steps, but this time, define the client ID as c00e9d32-3c8d-4a7d-832b-029040e7db99.

This value is the Azure Information Protection unified labeling client ID

Alternatively, follow the link below from Microsoft to register your key store on Azure: https://docs.microsoft.com/en-us/microsoft-365/compliance/double-keyencryption?view=o365-worldwide#register-your-key-store.

11.2 Create Sensitivity Labels using DKE

1. In the Microsoft 365 compliance center, Go to the **Information Protection** section.
2. On the **Labels** page, select **+ Create a label** to start the new sensitivity label configuration.
3. For example, from the Microsoft 365 compliance center:

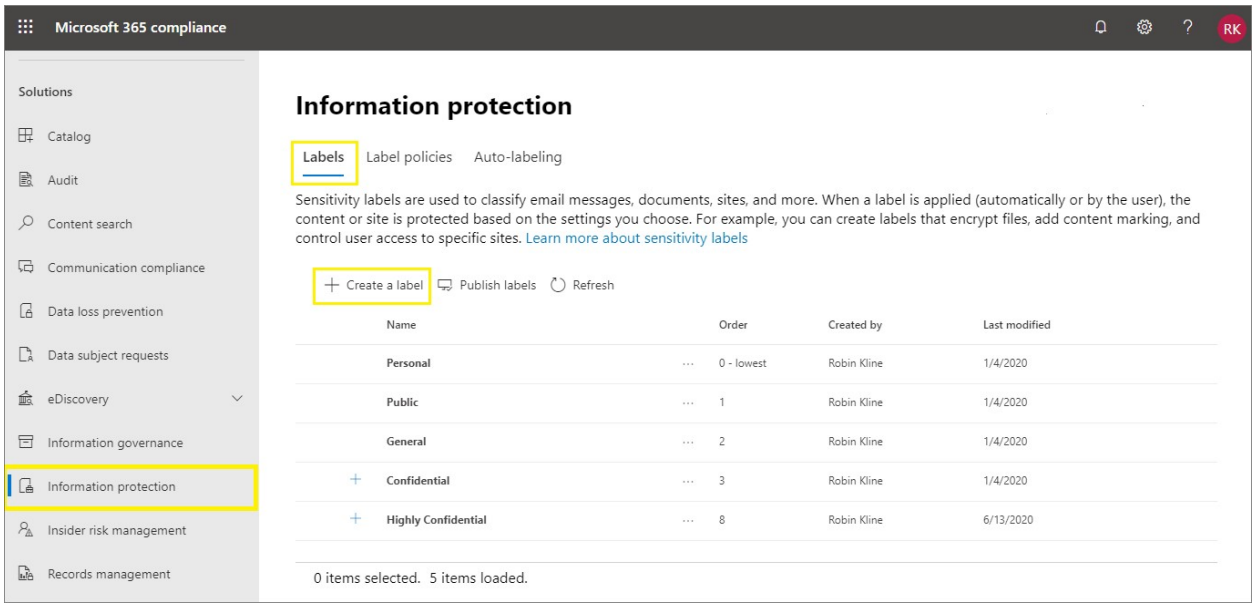


Figure 45 : Create sensitivity label

4. To name and create a tooltip for your label, specify the following:
   - Label name - visible in Microsoft 365 compliance center
   - Display name - visible to users in e.g., Microsoft Word
   - Description for users - the tooltip that appears when users hover over the sensitivity label

- Description for admins - the description available from the Microsoft 365 compliance center

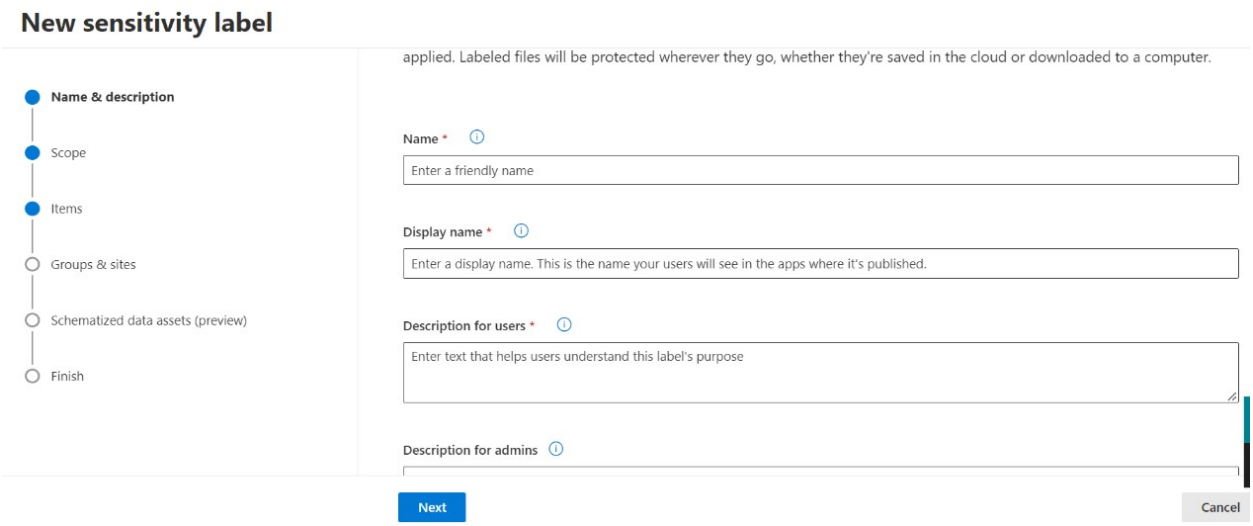


Figure 46 : Name and create a tooltip for your label

5. Click on **Next.**

6. Define the scope of the label, e.g., apply the label to files & emails only.

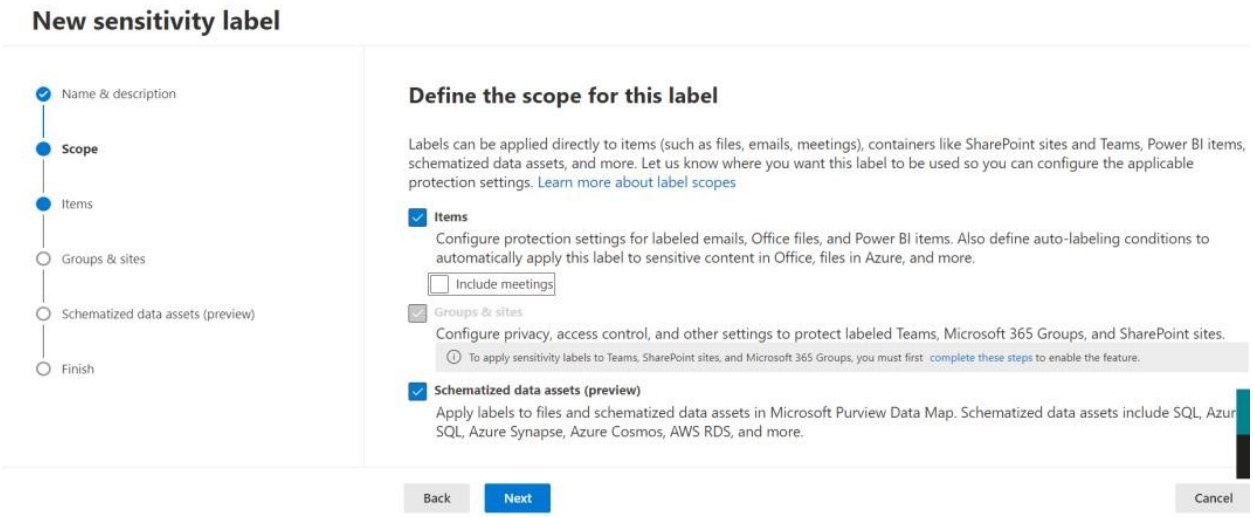


Figure 47 : Define the scope for a label

7. Click on **Next**.

8. Select the protection settings for files and emails.

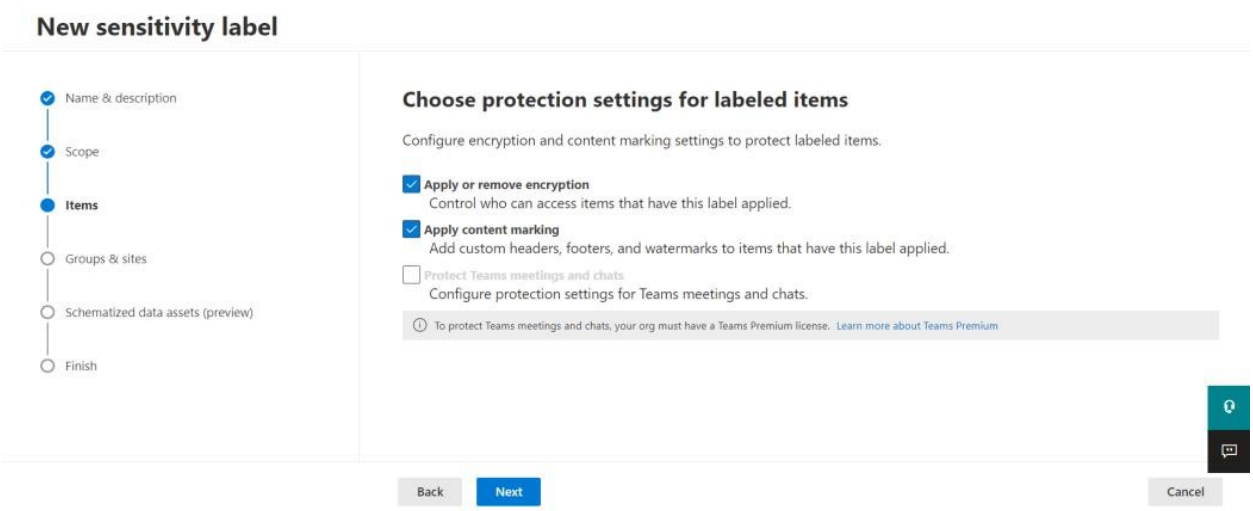


Figure 48 : Select protection settings for files and emails

9. In the encryption step, you can set permissions to content with the label applied. Since the primary reason for this procedure is to secure the files, you'll want to configure encryption settings.
10. In most cases, you will use the **Never expires** option, since it will let users access files without any time limit.
11. You can also allow or deny offline access to a file. Without the offline access, a user will need to be re-authenticated each time they want to open a document or email.
12. Choose which users or groups should be able to access documents with the label set. Clicking **Assign permissions** opens a wizard on the right side.
13. Finally, selectthe **User Double Key Encryption** checkbox and enter the correct DKE URL along with key.

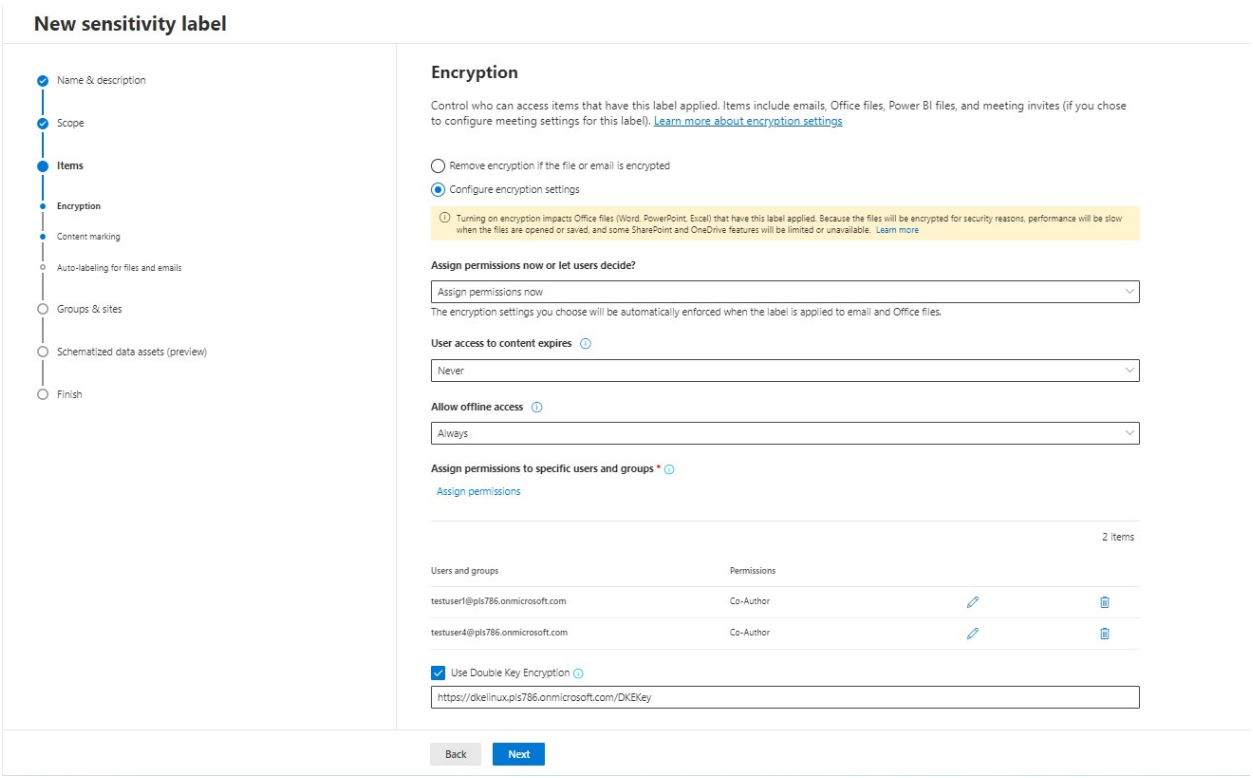


Figure 49 : Files and emails access

For example: https://dkeservice.utimaco.com/KEYNAME
Where KEYNAME is the label of the key which is generated using the DKE Anchor Manager.

The sensitivity label used for DKE protection needs to provide sufficient permissions for all intended recipients of the documents.

14. Content marking lets you add clear information about the applied labels directly to a document.
    You can choose if you want to add a watermark, a header and a footer, and what it should display on Microsoft word document.

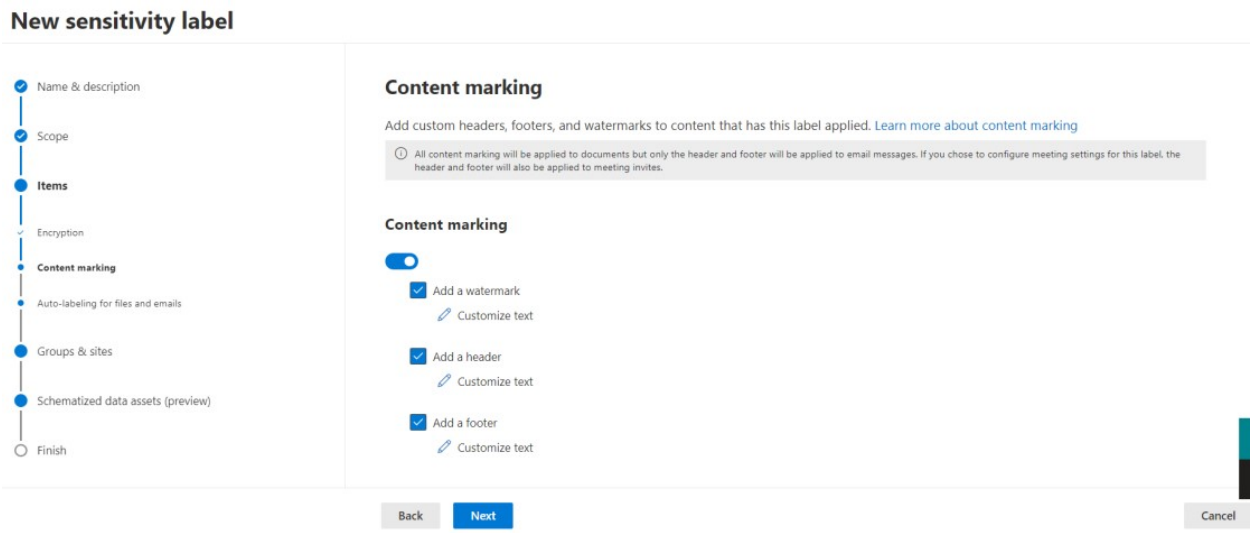


Figure 50 : Content marking

15. In the next step, you can turn on auto-labeling. That way certain groups or documents created in certain sites can be automatically marked with your label. This may be useful if you want to make sure nobody forgets about securing documents. On the other hand, auto-labeling can create problems if you intend to share some documents outside your organization.

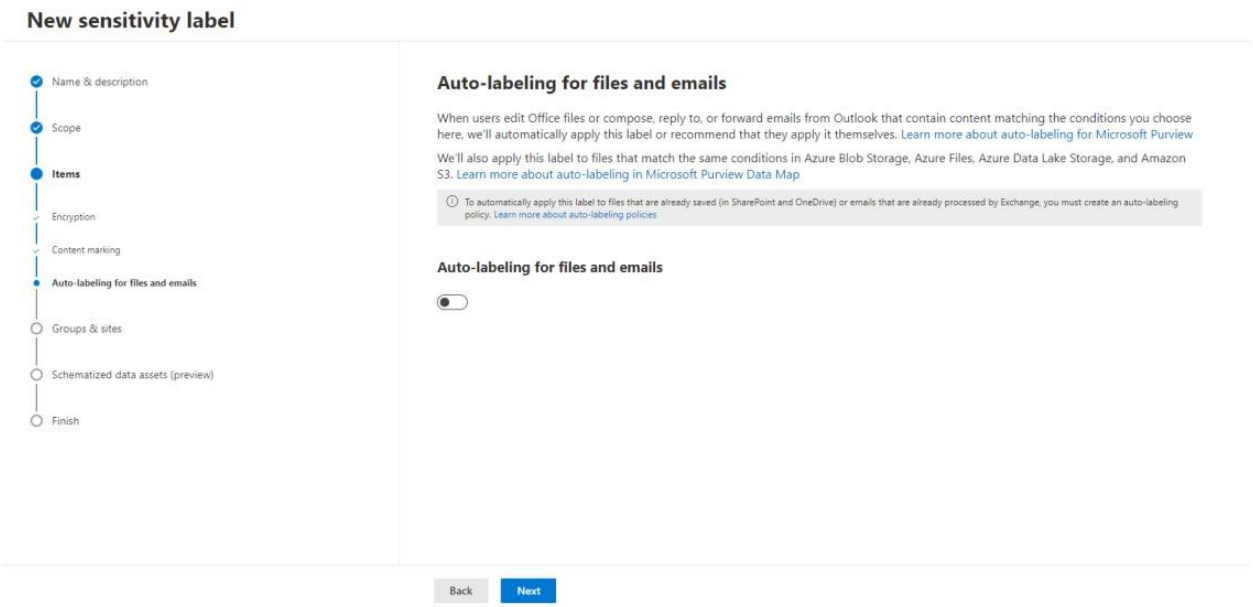


Figure 51 : Auto-labelling for files and emails

16. Click **Next**.
    You can define protection settings for groups and sites. Those settings, unlike the previous ones, apply to teams, groups or sites and not the documents stored in them

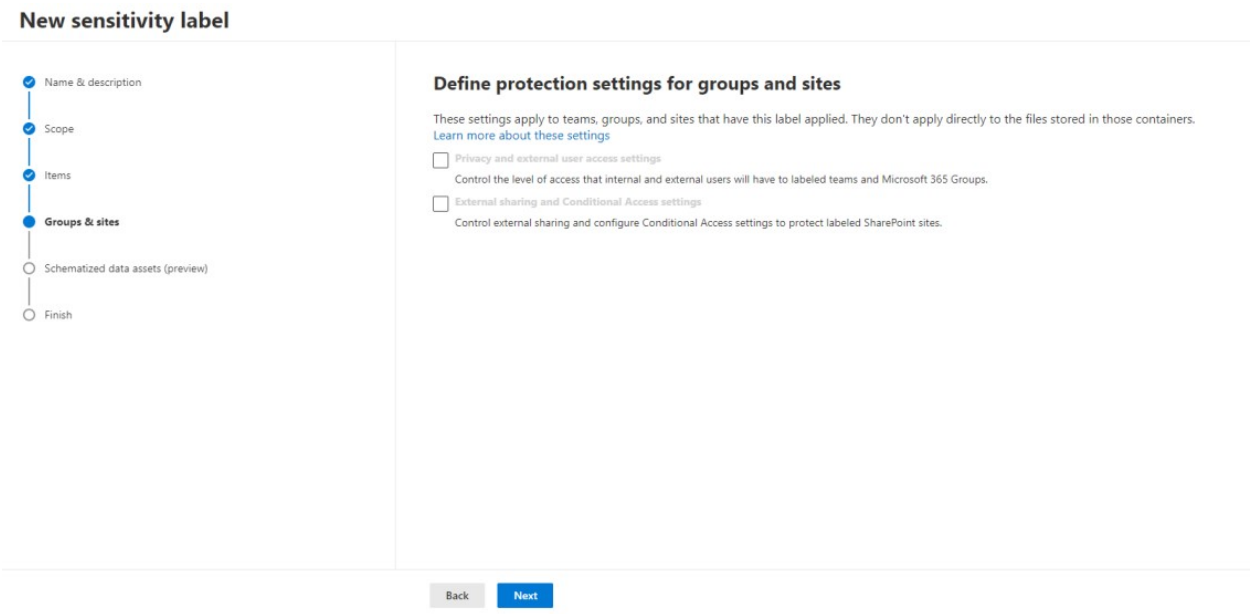
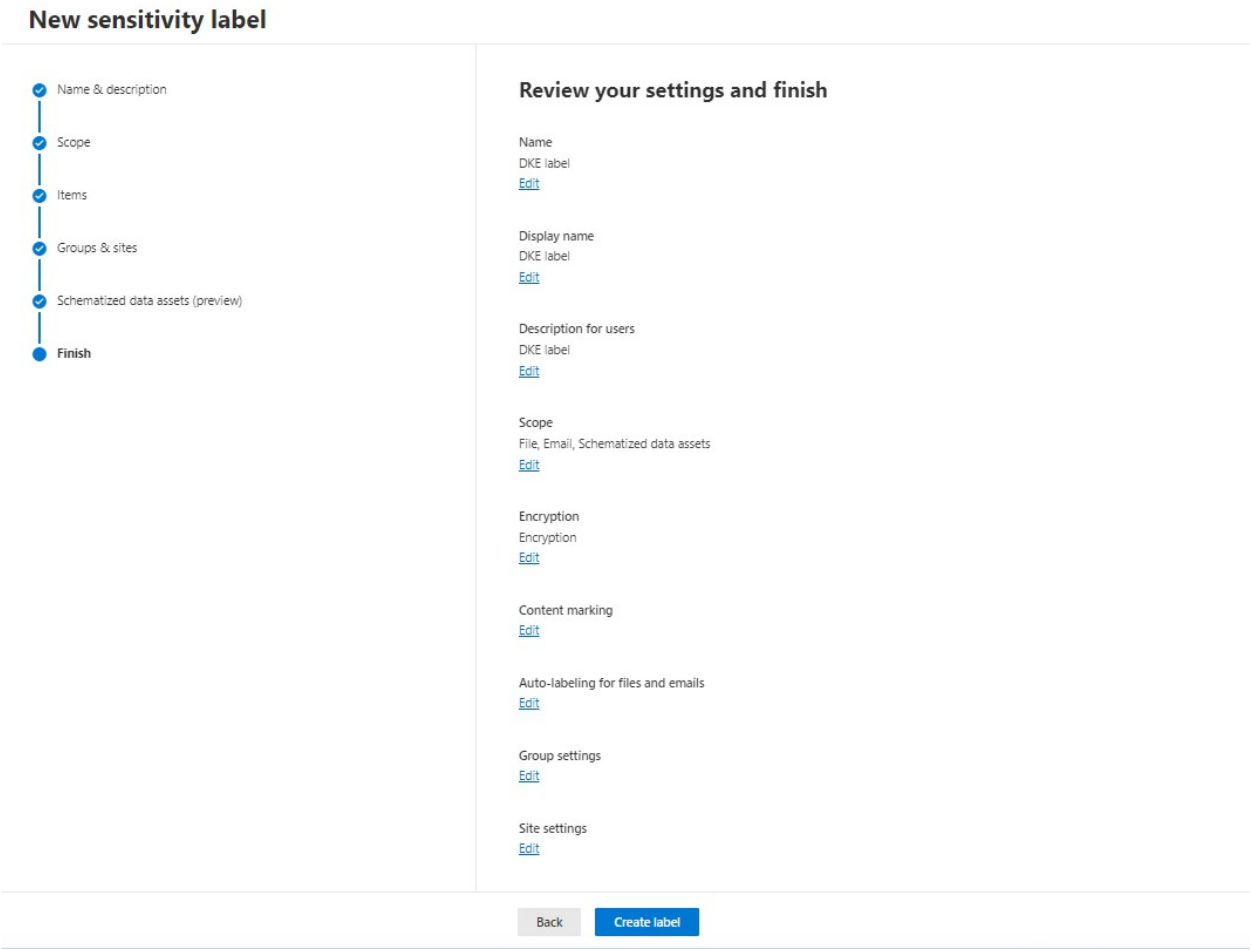


Figure 52 : Define protection settings for groups and sites

17. The last step is revising the configuration. Revise your settings, click **Create** label and then **Done**.



11.3 Publish the Labels

1. To publish the label, go to **Label policies** and click **Publish label**.

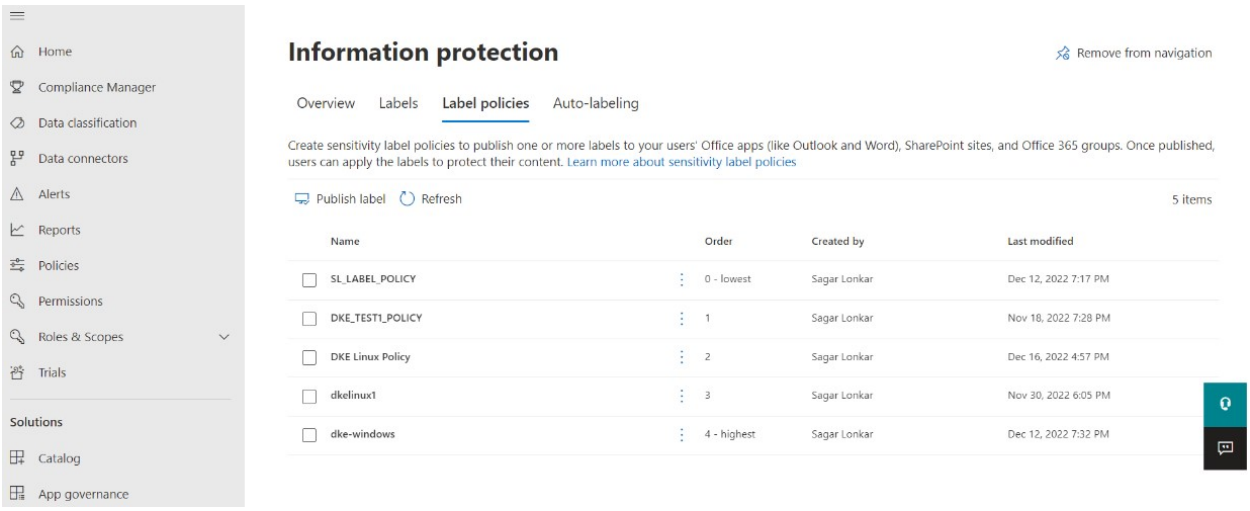


Figure 53 : Information Protection

2. Click **Choose sensitivity labels to publish** and pick the label you've created earlier. Click **Add** and **Next**.

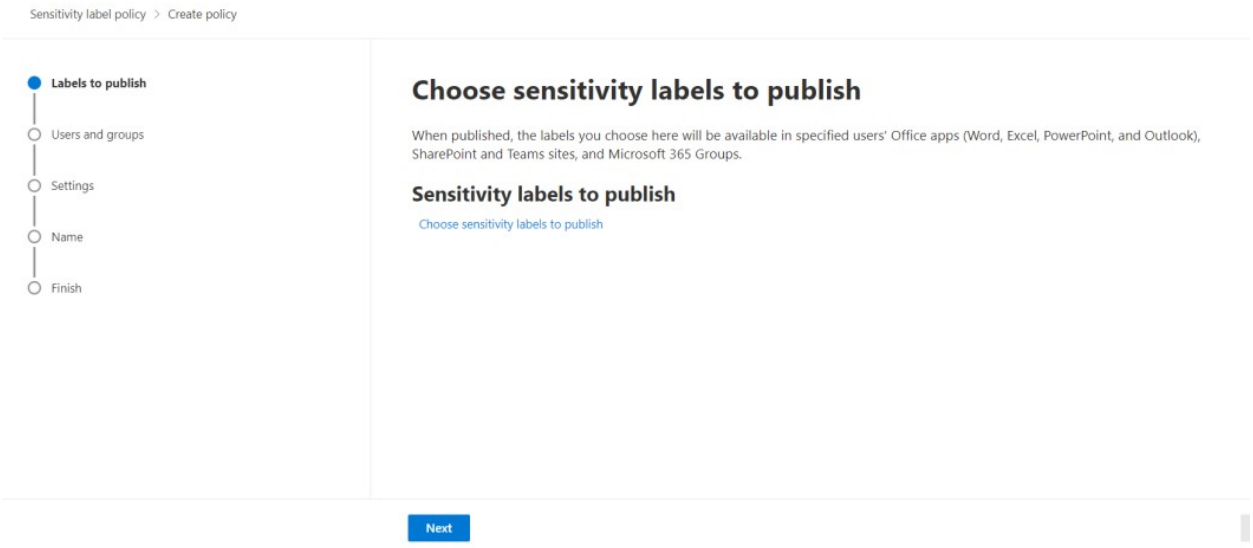


Figure 54 : Choose sensitivity labels to publish

3. Choose which groups or users should have the label available. Again, click **Done** and **Next**

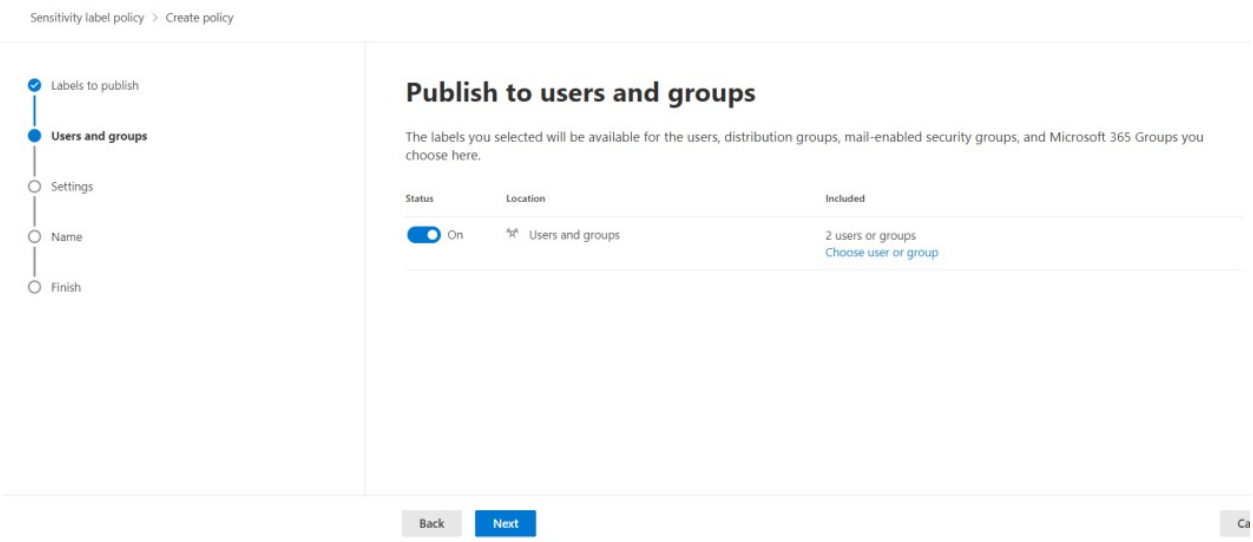


Figure 55 : Publish to users and groups

4. Next, you can choose to use various policy settings:

- Users must provide a justification to remove the label or lower its classification
- Require users to apply the label to their emails and documents
- Require users to apply the label to their Power BI content
- Provide users with a link to a custom help page

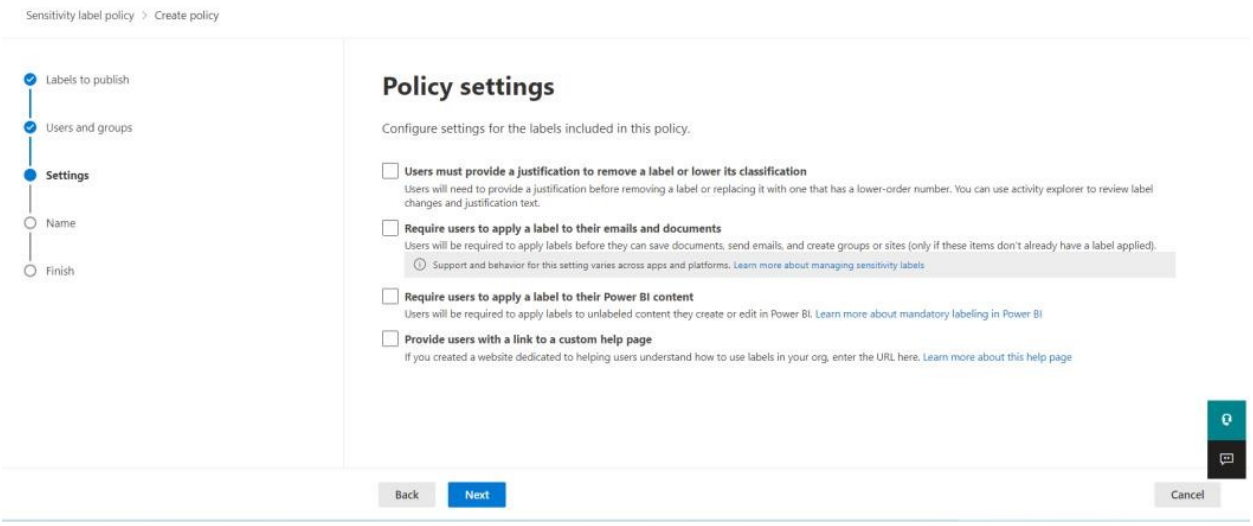


Figure 56 : Policy settings

5. The next step allows you to apply the default label to documents. If you leave the default None option, users will have the choice to apply the label or use the document without

enhanced protection. The next steps of the wizard let you configure the default labels for emails and Power BI.

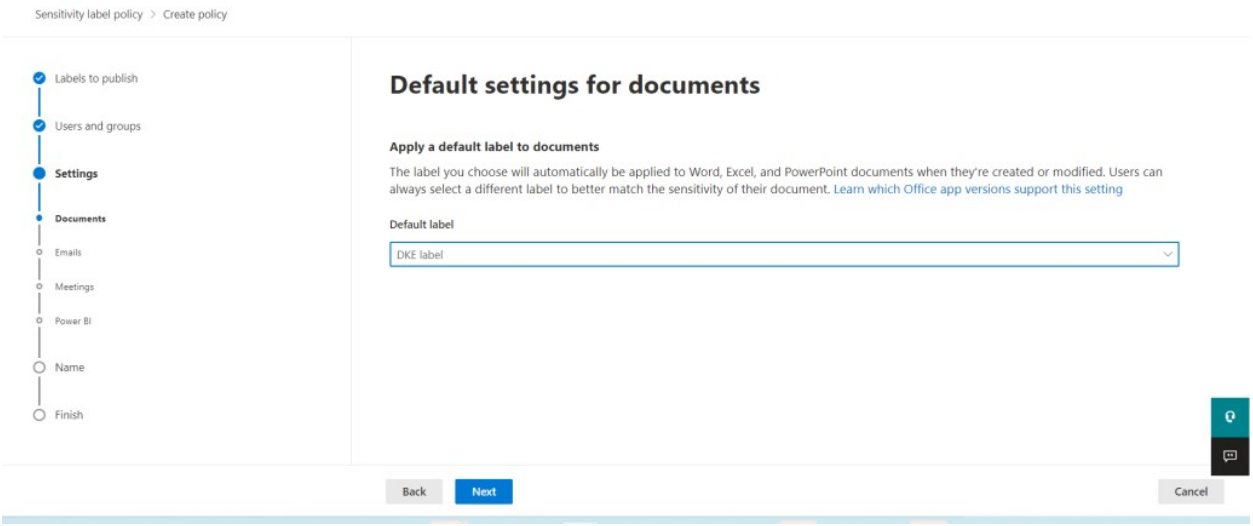


Figure 57 : Apply a default label to documents

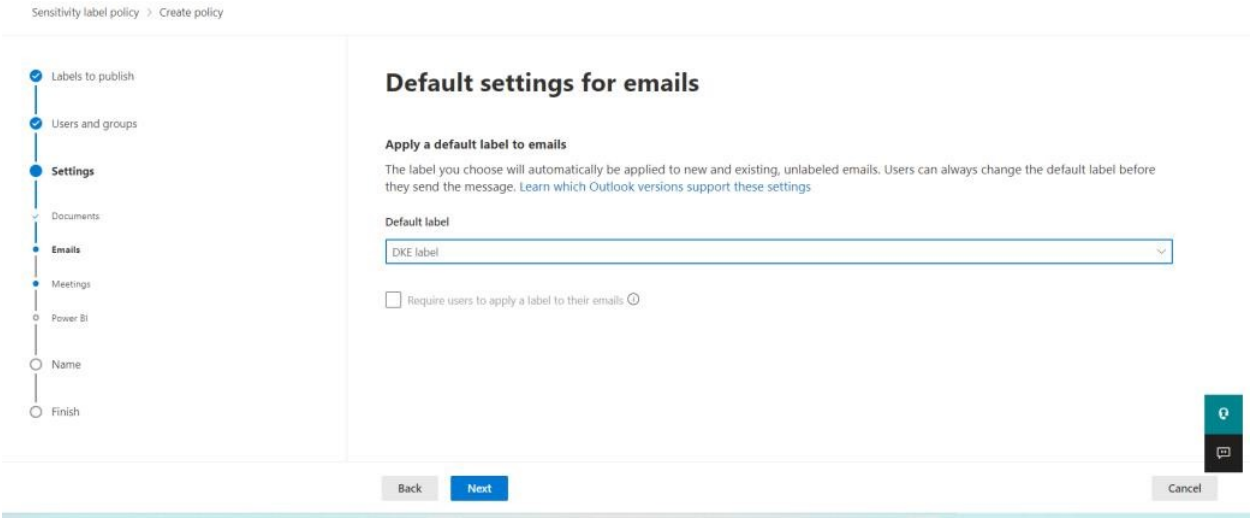
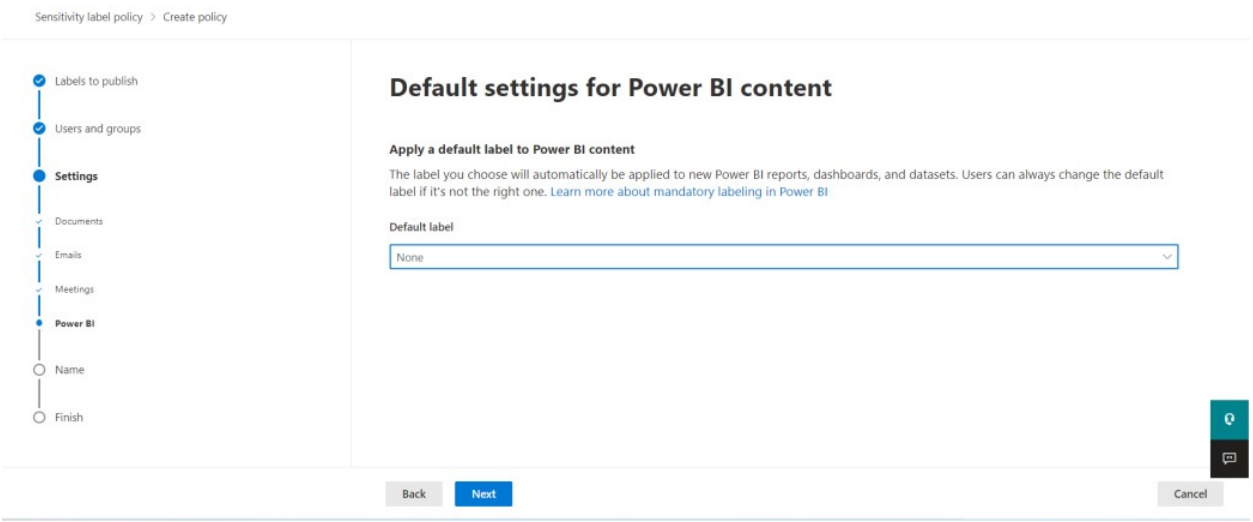


Figure 58 : Apply a default label to emails



6. Now you can name your label policy and provide a description

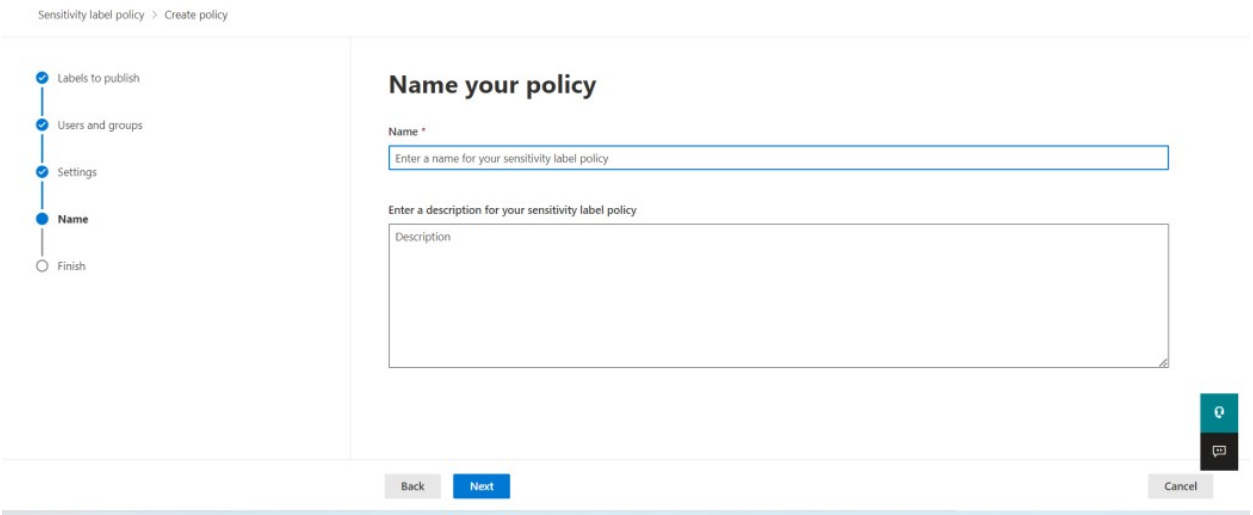


Figure 59 : Name your policy

7. Finally, review your policy. Click **Done** when everything is set

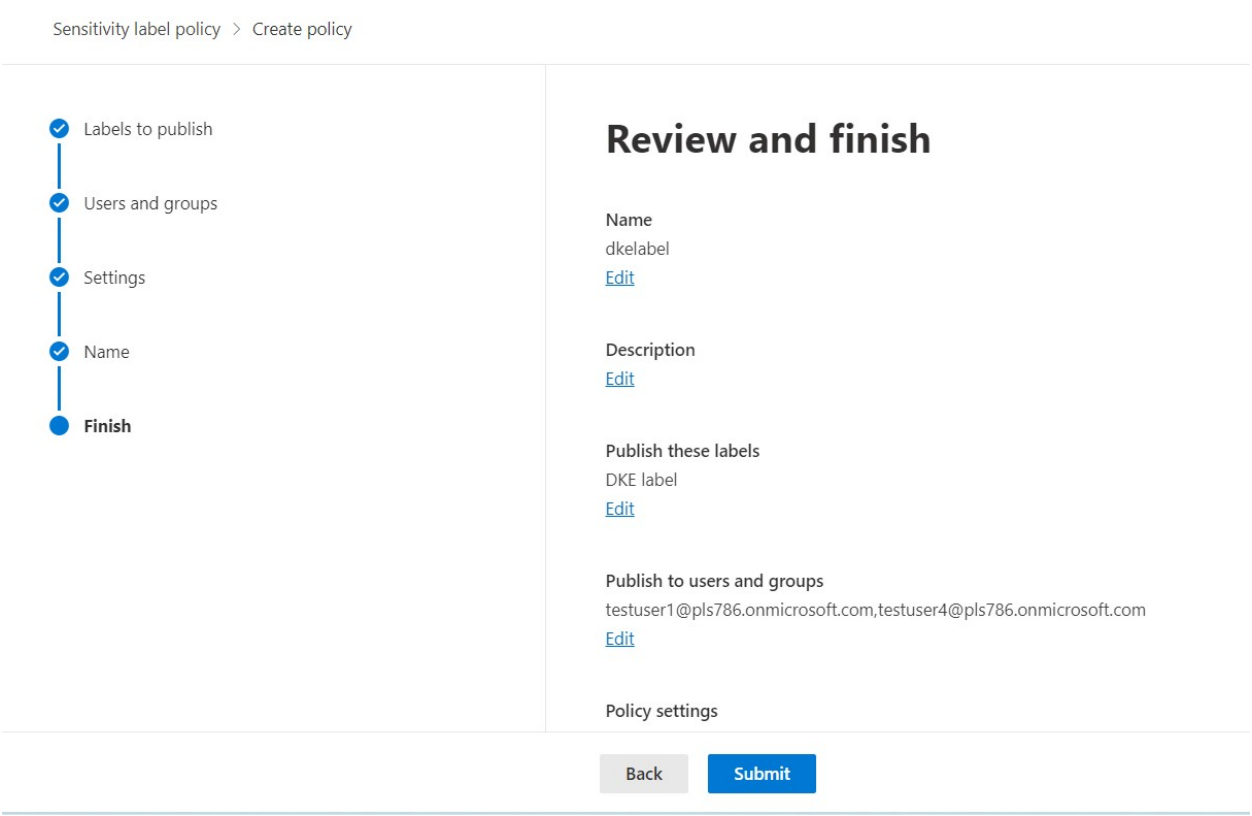


Figure 60 : Information Protection

It may take up to 24 hours for the clients to refresh with the new labels.

11.4 Enable DKE in your Client

As an Administrator user, execute the `dke.reg` file, which is part of the software package, on each client machine to update the registry.

11.5 Install the Azure Information Protection Unified Labeling Client

Ensure that the Azure Information Protection Unified Labeling Client is installed on each client machine. Use this URL to download it:

https://www.microsoft.com/en-us/download/details.aspx?id=53018

12 Verify Encryption and Decryption with DKE on Microsoft 365 Client

1. Log in to a client machine as a user to whom access has been provided as described in Execute the DKE Anchor Manager on Linux, Create Sensitivity Labels using DKE, and Publish the Labels.

Make sure before creating any file delete the cache folder data from both client machine on this location: `C:\Users\testuser4\AppData\Local\Microsoft\MSIP and MSIPC`.

2. Create a word document and assign sensitivity labels as created earlier, for example DKE.

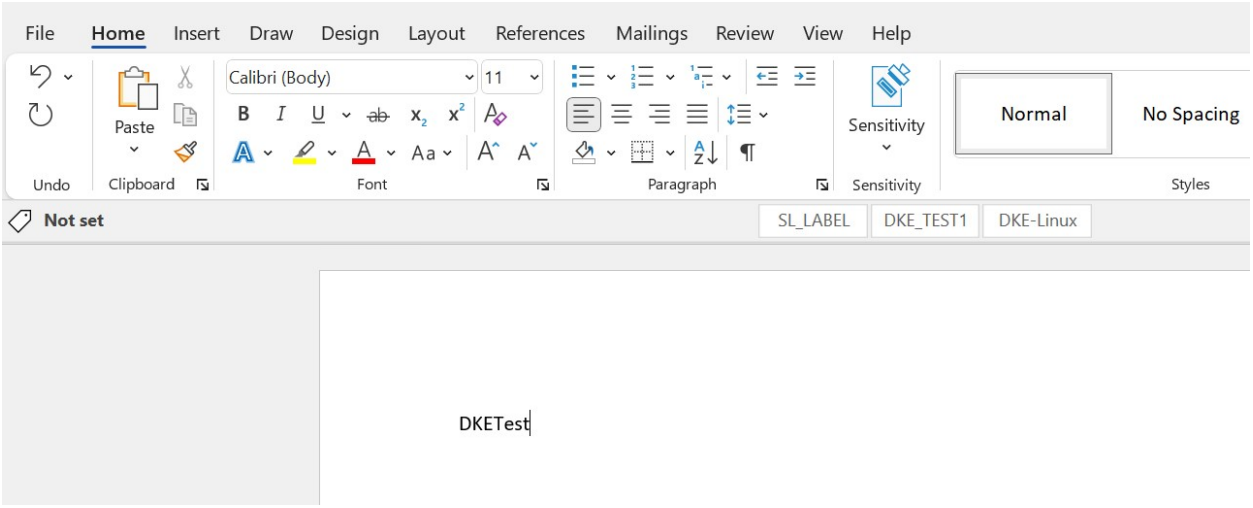


Figure 61 : Word doument

3. Save the file to any location.

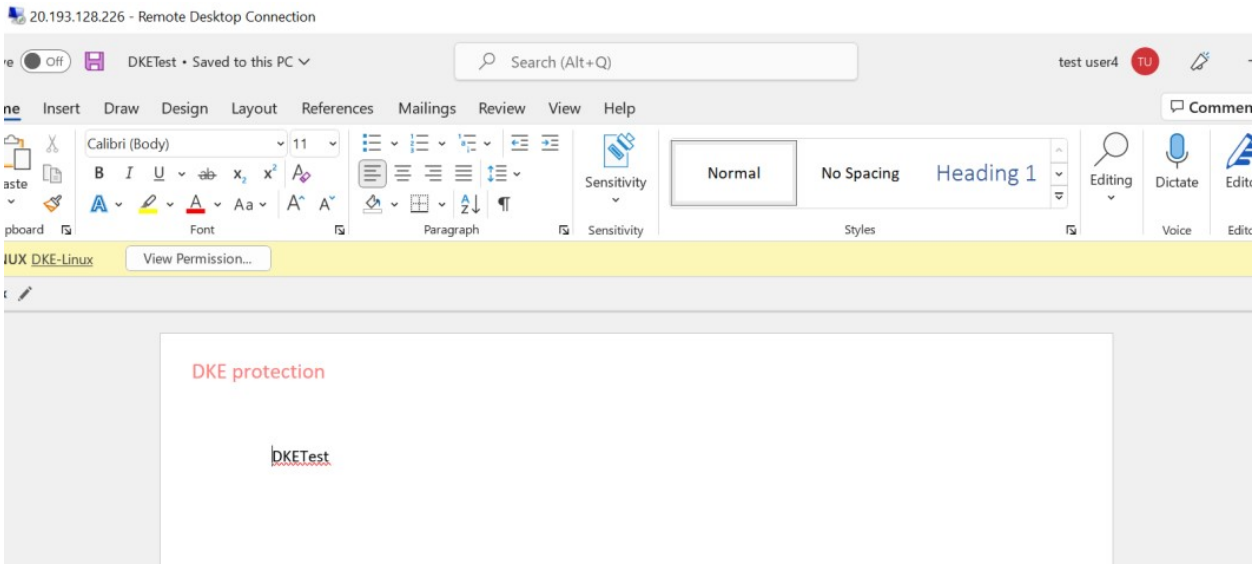


Figure 62 : Encrypted Word document

4. Copy this file to another client machine and user with allowed access to this sensitivity label and DKE.

5. Open the file and verify that this authorized user is able to open the document.

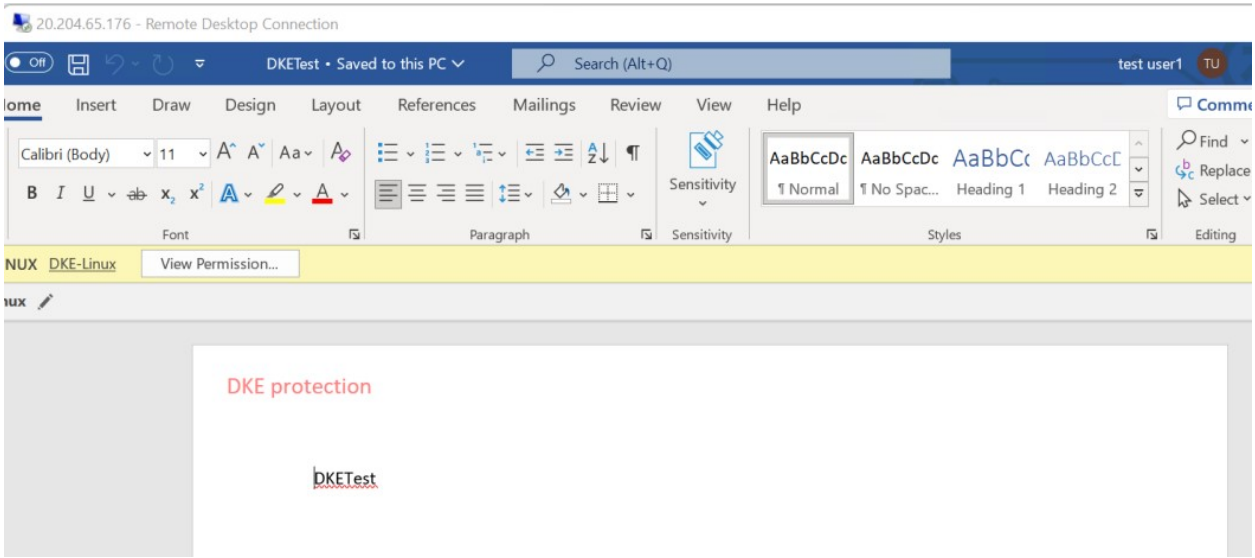


Figure 63 : Encrypted Word document

6. Now copy the same file to another client machine with user who is not authorized to this sensitivity label and DKE

7. Try to open the document. You will get error with below permission message.

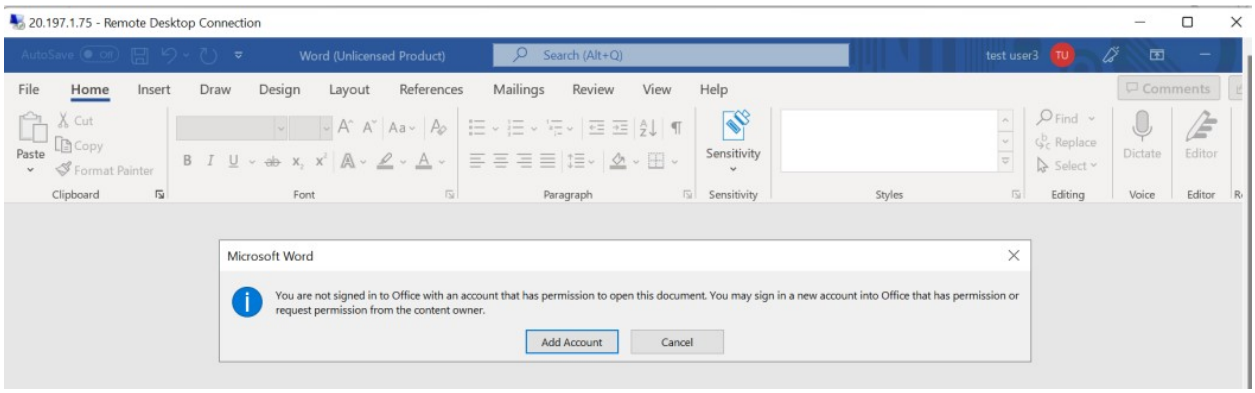


Figure 64 : Access denied message

13 Troubleshooting

Error	Diagnosis
Error: Failed to attach external HSM client library. Please check if you specified the vendor provided PKCS#11 library path correctly	Verify whether the correct Path to PKCS#11 library path is specified in pkcs11Lib property of application,yml
CKR_USER_PIN_NOT_INITIALIZED	PKCS#11 Slot is not initialized.
CKR_PIN_INCORRECT	Check userPIN property in application,yml
Bearer Token Error	Go to Control Panel / Network and Internet / Internet Options. In Internet Options, go to Security / LOCAL INTRANET / Sites / Advanced, and add the DKE URL to that. Please note, this is LOCAL INTRANET sites, not Trusted sites. It must be LOCAL INTRANET
Http timeout error	Increase pkcsHandleMaxPoolSize in application.yaml and restart DKE Anchor service
Despite having ample space on a disk (or on OneDrive), the following message is shown when saving a DKE protected document: «Word cannot save or create this file. Make sure the disk you want to save the file on it is not full, write-protected, or damaged. »	

Error	Diagnosis
Potential Issue: The client is not configured to use DKE.	1. Re-check the Office version 2. Verify the DKE registry keys have been imported on the client
Potential Issue: The client cannot reach the DKE Anchor service	On the client, try opening the DKE-URL configured in the sensitivity label. If that fails, fix the network issue as needed.
You are not signed into Office with an account that has permission to open this document. You may sign in a new account into Office that has permission or request permission from the content owner	
Potential Issue: The user hasn't been granted permission in the sensitivity label.	During tests, try granting the whole tenant access in the sensitivity label permissions
Potential Issue: The DKE Anchor service URL contains a sub-folder	Verify that the DKE URL consists of the FQDN only
Potential Issue: The web application isn't configured correctly.	Check the settings in the application.yml
An unknown error occurred. If this problem persists, contact your administrator or help desk	
Potential Issue: The client doesn't have the correct Office version installed.	Re-check the Office version

Error	Diagnosis
Error	Diagnosis
Potential Issue: The AIP client is not registered in the web application	Check whether the client ID for the AIP client has also been registered in the web application

Error	Diagnosis
2022-12-07 10:11:24.742 ERROR 1292080 --- [-nio-443-exec-1] c.u.d.s.JWTAuthenticationEntryPoint: Inside Unauthorized processing 2022-12-07 10:11:24.758 INFO 1292080 --- [-nio-443-exec-9] c.utimaco.dke.controller.DkeController: Entering method to decode data for DKE_Key,8f0c4c01-7f05-47ca-adfe-f45030046a31 2022-12-07 10:11:24.758 INFO 1292080 --- [-nio-443-exec-9] c.utimaco.dke.controller.DkeController: Authorization header found, Identifying user identity	Step 1: delete your cache files on both client machine from C:\Users\testuser4\AppData\Local\Microsoft\MSIP and MSIPC Step 2: Check the Keyinfo command status and make sure Remaining Ops value will be more than 99999 Command: cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 group="SLOT_0000" spec=3 Keyinfo Step 3: If value of remaining ops is 0 run below command: cxitool dev=3001@127.0.0.1 LogonPass=USR_0000,123456 group="SLOT_0000" Spec=3 KeyFile=ka.key AuthorizeKey=1000000

Table 7: Errors and its diagnosis

For more troubleshooting see https://www.drware.com/dke-troubleshooting/

14 Further Information

This document is a part of the information and support provided by Utimaco IS GmbH. Additional documentation can be found on the product CD in the Documentation directory.

For the Microsoft DKE documentation, see https://learn.microsoft.com/enus/microsoft-365/compliance/double-key-encryption?view=o365-worldwide

All CryptoServer product documentation is also available on the Utimaco IS GmbH website: https://utimaco.com/

15 References

Reference	Title/Company	Document No.
[CSADMIN]	CryptoServer – csadm Manual/Utimaco IS GmbH	2009-0003
[CSTrSh]	CryptoServer Troubleshooting/Utimaco IS GmbH	M011-0008-en
[CSADMIN2]	CryptoServer_csadm_Manual_Systemadministrators.pdf	2009-0003
[CSP11Tool2]	CryptoServer_p11tool2_Manual.pdf	2012-0004
[CSPKCSM]	CryptoServer - PKCS#11 P11CAT Manual	M013-0001-en
[CSLAN5]	CryptoServerLAN_Manual_Systemadministrators.pdf	2018-0004

16 Contact and Support Information

You can reach us from Monday to Friday, 09.00 a.m. to 05.00 p.m., Central European Time (CET).

Utimaco IS GmbH
Krefelder Straße 220
52070 Aachen
Germany

RMA Query

If you need to send the device back to Utimaco IS GmbH, please open a new RMA case (Return Merchandise Authorization). We request that you use the following web address. RMA cases cannot be opened by email or phone.

https://support.hsm.utimaco.com/support/rma/new

Other Support Queries

- Mail (preferred contact method)
  support@utimaco.com
  Attach the diagnostic information to your email.
- Web portal
  https://support.hsm.utimaco.com/support/cases/new/
  The diagnostic information will be requested in our response if necessary.
- By phone
  AMERICAS +1-844-UTIMACO (+1 844-884-6226)
  EMEA +49 800-627-3081
  APAC +81 800-919-1301
  The diagnostic information will be requested in our response if necessary.