NFC 2Go Starter Kit For Brand Protection

User Guide

About this document

Scope and purpose

The scope of this document is limited to the NFC 2Go starter kit for brand protection, its hardware and software components.

The purpose of this document is to assist users get started with NFC 2Go starter kit for brand protection.

Intended audience

This document is primarily intended for engineering teams at customers, solution providers and system integrators who want to understand and evaluate the NFC 2Go starter kit for brand protection, and develop custom prototypes to meet application requirements.

When reading this document, you should have

- The NFC 2Go starter kit for brand protection
- A basic technical understanding of mobile and cloud application development

Table of contents

Table of contents

List of tables

List of figures

1 Introduction

Infineon provides a Secured NFC tag-based solution for brand protection that performs product authentication, and includes consumer engagement.

The term “near-field communication” (NFC) refers to a technology that allows for the contactless transfer of information over short ranges. This allows the wireless transfer of data from one device (e.g., mobile phone/ reader) to another device such as NFC tags, cards etc.

1.1 NFC brand protection

Infineon offers a number of products that allow configuration as NFC passive tags. NFC-enabled mobile phones can read such tags and trigger specific actions depending on the information read out from the tag. Secured NFC tags can be used for the brand protection of products.

The Secured NFC tag, which is used for brand protection, is directly embedded into the product and contains a data set that can be used to identify the individual product and verify the originality of the data set. The Secured NFC tag implements truly secure and convenient brand authentication using secure cryptography that is difficult to clone and counterfeit.

As a result, the Secured NFC tag can be used to achieve the following:

- Validate the originality of the product
- Store and retrieve more information about the product and its specifications
  - Service information of the product (such as warranty details, etc.)
  - NFC records (such as URI, text, smart poster, vCard, etc.)

Figure 1 illustrates the brand authentication using an NFC-enabled phone with the Infineon NFC verifier application and a product with an embedded Infineon Secured NFC tag.

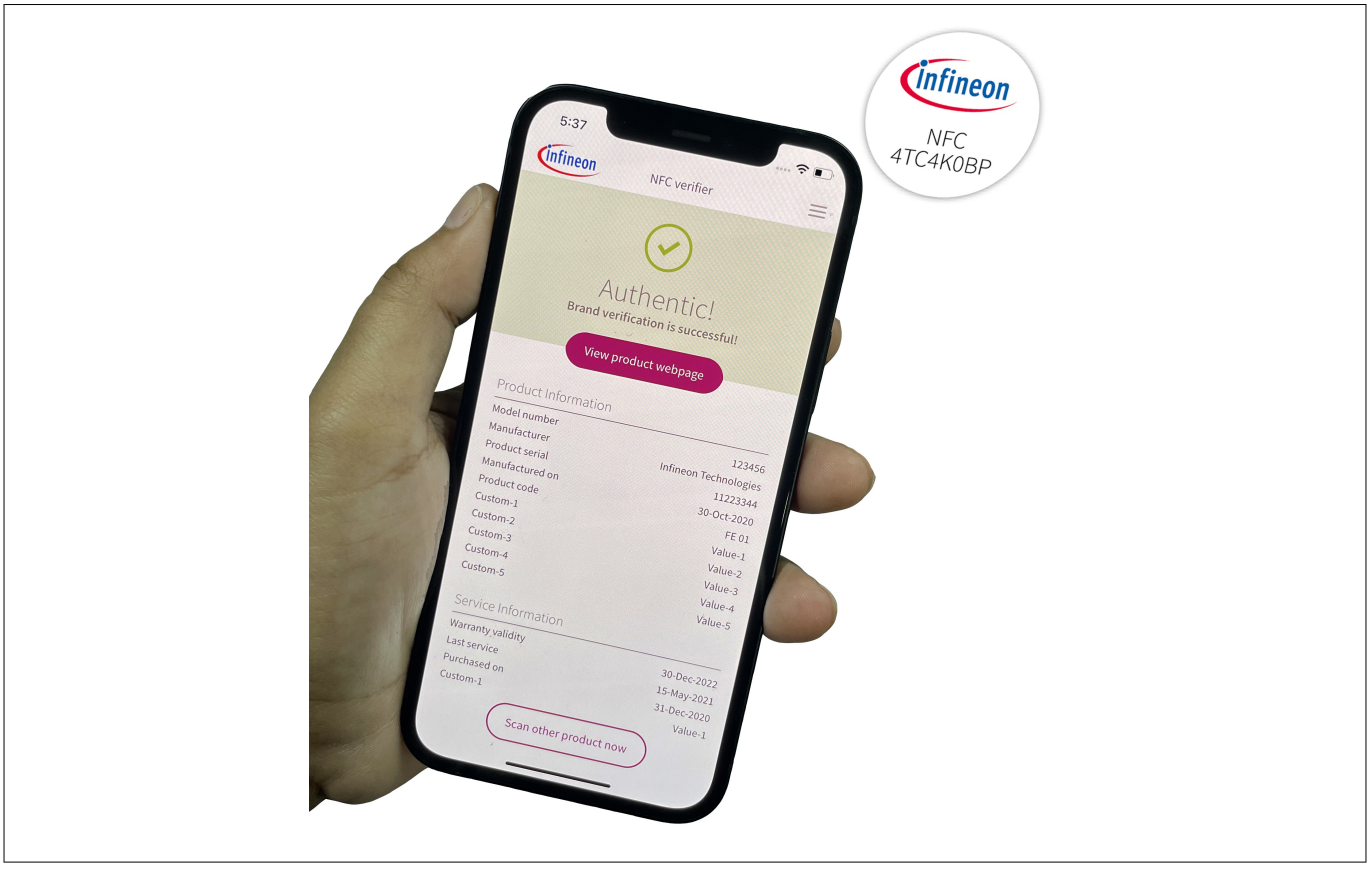

Figure 1 Brand verification using an NFC-enabled phone

2 Product overview

This chapter describes the NFC 2Go starter kit for brand protection, its contents and the main approach for implementing the brand protection solution.

2.1 NFC 2Go starter kit for brand protection

Infineon's NFC 2Go starter kit for brand protection offers a brand protection solution that validates the authenticity of the products via NFC communication.

The main focus of this starter kit is to allow users to understand the following:

- NFC-based trustworthy solutions for brand protection
- Various Infineon NFC products and their features
- Ease of use and integration into different types of goods
- Quick implementation of prototypes for brand protection

This starter kit demonstrates an NFC-based brand protection solution and also provides open-source application software examples that can be used to implement a customized brand protection solution to meet the application needs.

Figure 2 NFC 2Go starter kit for brand protection

2.2 Starter kit contents

Infineon's NFC 2Go starter kit for brand protection provides an evaluation environment that includes:

- 8 pieces of Secured NFC tags (NFC4TC4K0) in sticker form with contactless interface
- The "Infineon NFC verifier" mobile application for iOS and Android. This application can be downloaded from Apple App Store [2] or Google Play Store [3]

- The mobile and cloud software that supports cloud-based brand verification solution. This software is an open-source and hosted on GitHub [1]
- "Infineon secured NFC tag IDE", a Windows-based PC tool that reads and writes keys and information into Secured NFC tags. This tool is hosted on Infineon Development Center [4]

The following are the hardware requirements for using this starter kit:

- An NFC-enabled iPhone 8 and higher (iOS 13 and above)
- An NFC-enabled Android phone (Android 5 and above)
- Supported "Secured NFC tags" are as follows:
  - NFC4TC304
  - NFC4TC1K0
  - NFC4TC2K0
  - NFC4TC4K0
- PC (Windows 10 and above)
  - JRE V1.8 and above (32-bit variant)
- PC/SC reader

2.3 Cloud-based brand protection

Cloud-based brand protection is an approach that uses cloud services to authenticate the key present in the Secured NFC tag. This approach uses symmetric cryptography in which an AES-128 bit master key is stored in the cloud and the diversified version of the key (unique for each tag) is stored in the Secured NFC tag. To ensure the authenticity of the brand, a mutual authentication takes place between the Secured NFC tag and the cloud service during brand verification.

Figure 3 shows the system architecture of the cloud-based brand protection system.

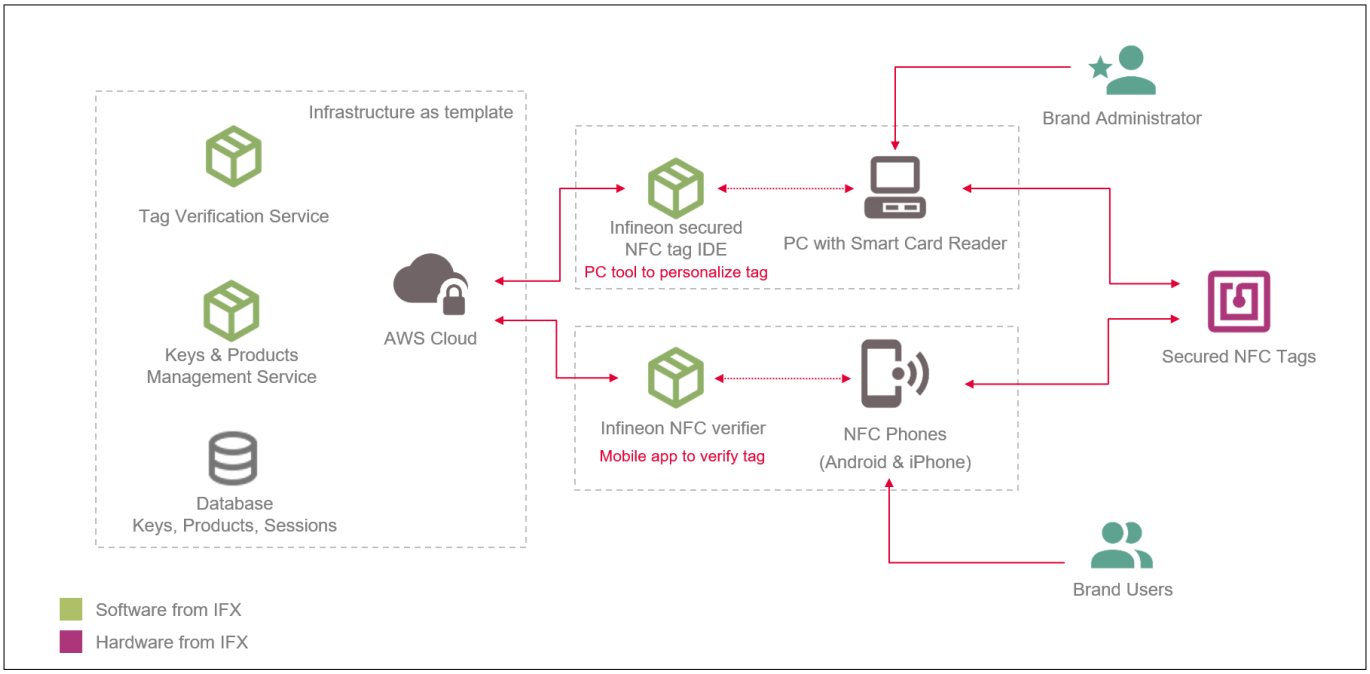

Figure 3 System architecture

This system consists of the following components:

- Secured NFC tag that will be embedded in the final product
- Mobile application (Infineon NFC verifier) that performs brand verification

- PC tool (Infineon secured NFC tag IDE) that personalizes Secured NFC tag content and cloud content
- Cloud services to facilitate cloud-based brand verification

The master keys in the cloud are accessible to the brand administrator. Using the Infineon secured NFC tag IDE tool, the brand administrator can personalize the keys into the Secured NFC tag and can also update the product information. The brand user can verify the authenticity of the tag by tapping an NFC enabled mobile phone against the Secured NFC tag using the Infineon NFC verifier application.

3 Product usage

This chapter explains how to use the starter kit's contents, which include a mobile application for iOS and Android (Infineon NFC verifier) and a Windows-based PC tool (Infineon secured NFC tag IDE).

3.1 Working with the Infineon NFC verifier application

The Infineon NFC verifier is a mobile application that is typically used by the end users of the brand to verify the authenticity of the product and to read the product information.

The Infineon NFC verifier application supports the following key features:

- Performs authentication with the cloud service
- Assists in the reading of brand protection application profiles A10, B10 and B20
- Displays product and service information based on the profile present in the scanned tag
- The mobile phone auto-launches the application when a Secured NFC tag that is configured with a URL record referring to the landing webpage URL, is scanned in the background/home screen

3.1.1 Installing the application

The Infineon NFC verifier - mobile applications for evaluating this starter kit are available for download from the following mobile application stores listed below:

- Infineon NFC verifier - iOS application: Apple App Store [2]
- Infineon NFC verifier - Android application: Google Play Store [3]

The steps for installing these applications are as follows:

1. Open the "App Store" on an iOS device or the "Play Store" on an Android device.
2. Search for the "Infineon NFC verifier" application from Infineon Technologies.
3. Install the application.
4. To launch the application, press the "open" button.

3.1.2 Performing brand verification with pre-configured tags

The Secured NFC tags available in the starter kit are pre-configured with default keys, product and service information. Therefore, it works out of the box with the Infineon NFC verifier application for quick evaluations.

Note: Before using this application, make sure that the phone's NFC is turned on in the settings and that an internet connection is available.

The steps for verifying the authenticity of the tag are as follows:

1. Open the Infineon NFC verifier application, this will scan for the Secured NFC tag upon launch.

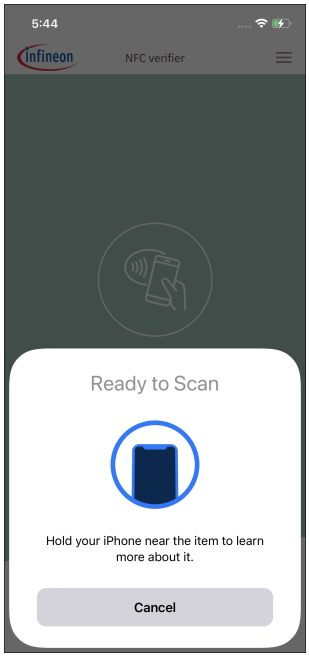

Figure 4 Scan mode

2. If the application is not already in scan mode, tap "Scan & Verify" to scan for the Secured NFC tag.

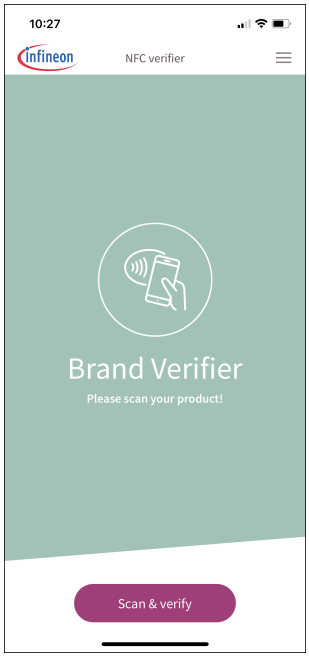

Figure 5 Home screen

3. Place the Secured NFC tag near the phone's NFC antenna area.

4. Once the Secured NFC tag is identified, the application performs brand verification with the cloud service and displays the result.

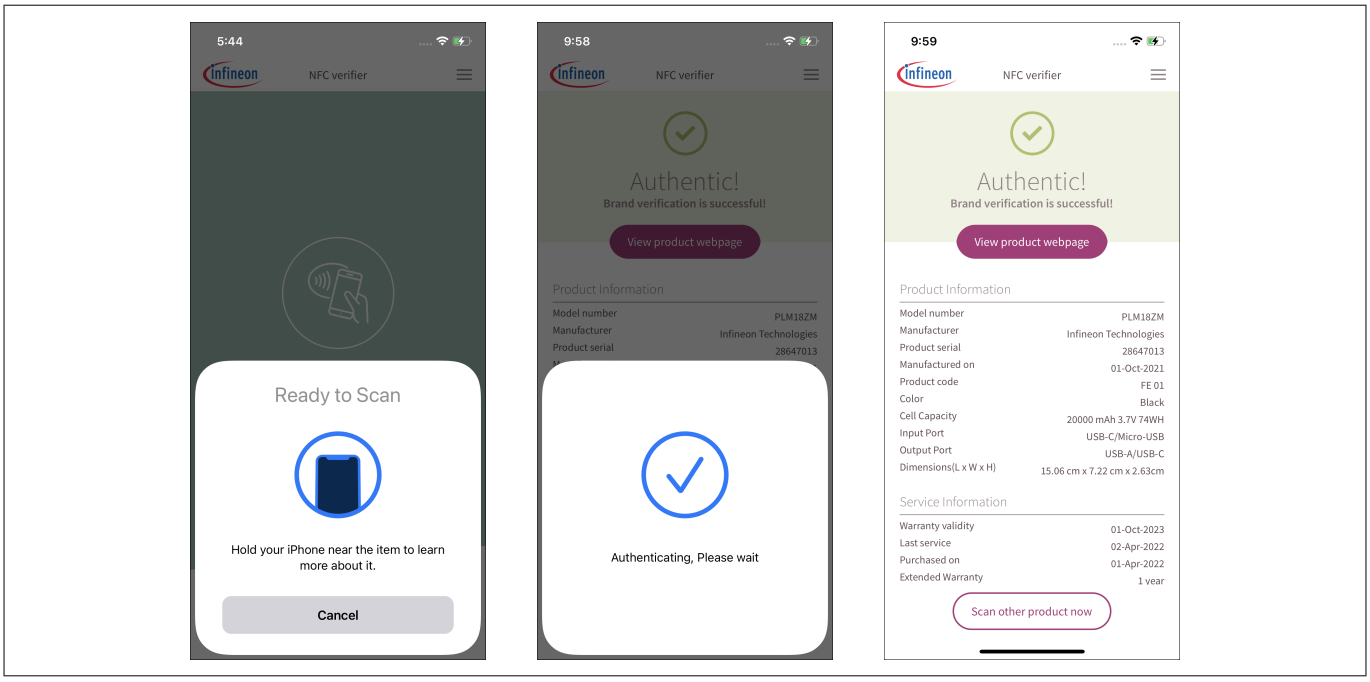

Figure 6 Authentication using the application

5. If the tag is authentic, the application reads and displays the product and service information from the tag.
6. If the tag contains a product URL, clicking "View product webpage" will launch the product's webpage (refer to Chapter 3.1.4.1).

Figure 7 shows the NFC range present in the top back edge of most iPhones.

Figure 7 NFC-enabled iPhone

Figure 8 shows the NFC range present in the back body of most Android phones, near the camera or in the center.

Note: The antenna locations mentioned here are only for reference purposes. Based on the manufacturer's design, it may be located elsewhere."

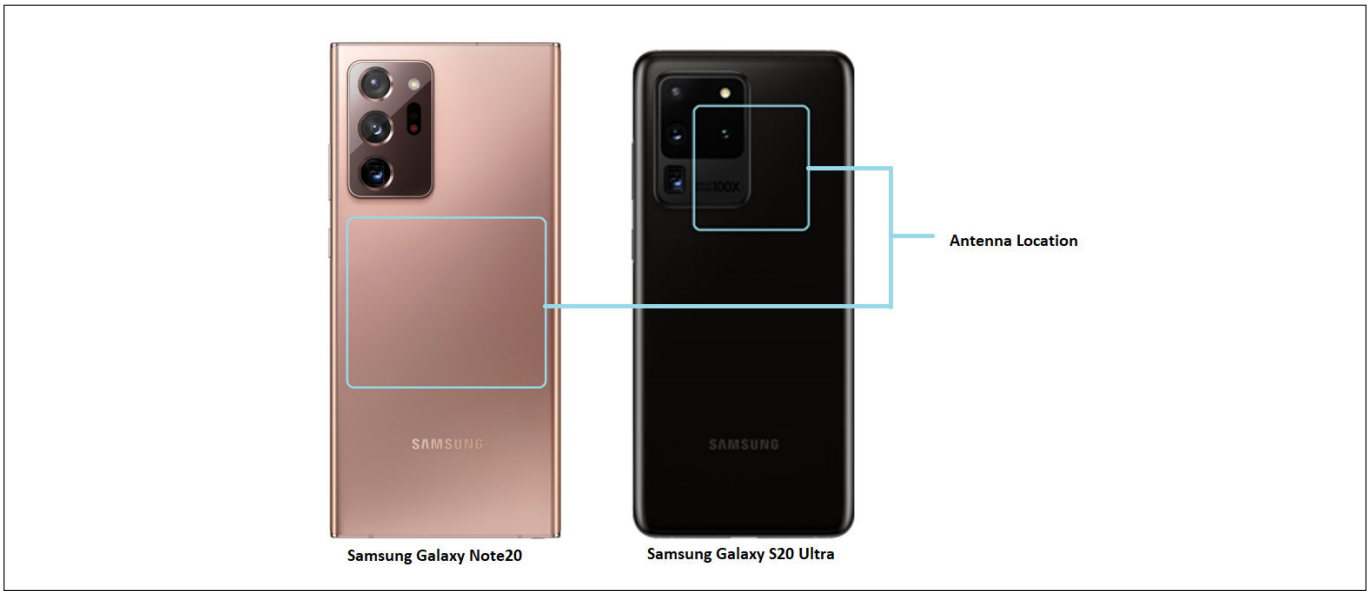

Figure 8 NFC-enabled Android phone

3.1.3 Product and service information

The brand protection application profiles allow for the storage of information about the product and its services, based on the memory supported by the Secured NFC tag. Table 1 includes information about each of these application profiles.

Figure 9 displays the product and service information read-out from a tag.

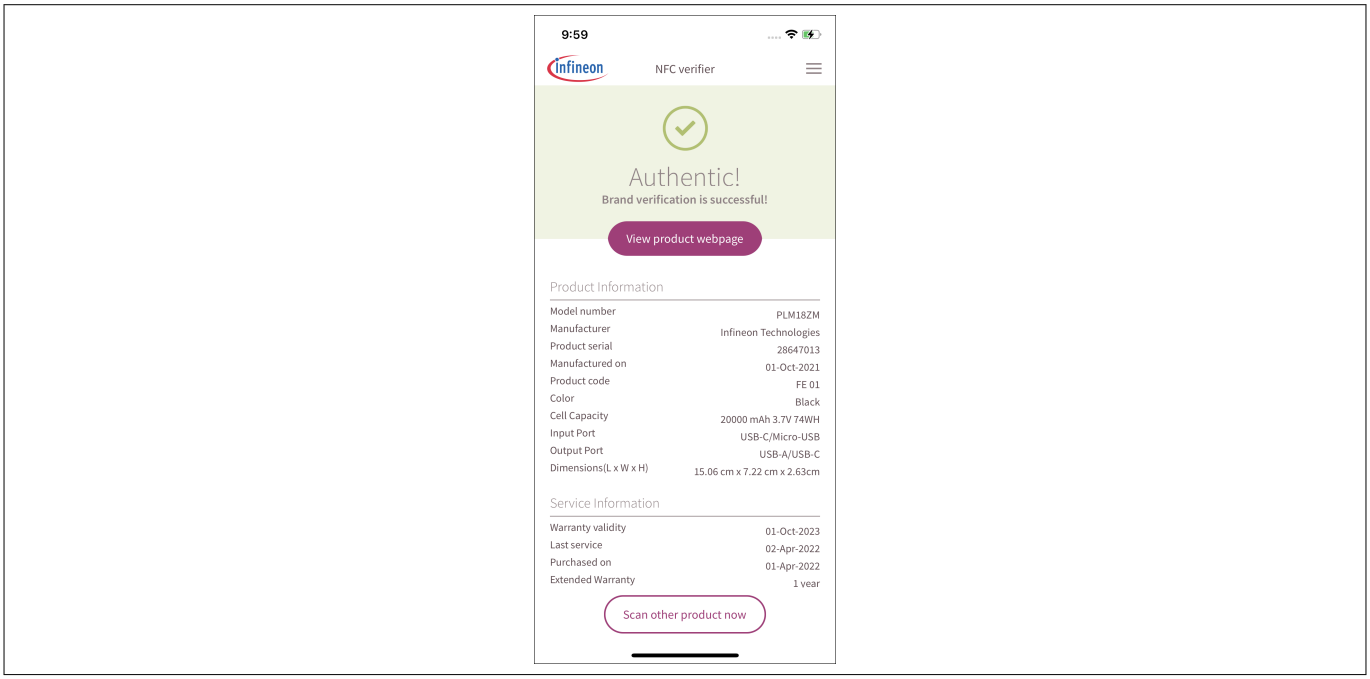

Figure 9 Product and service information of the product

These profiles allow users to store product information such as model number, manufacturer's name, product serial, manufactured date, product code and other custom product information.

These profiles also allow users to store service information such as warranty validity date, last service date, purchase date and other custom service information.

For more information on the brand protection application profiles, refer to Chapter 4.2.

3.1.4 Other features

This section describes the menu options and other features available in this application.

3.1.4.1 Product webpage

The product can be linked to an URL that can be opened by scanning the Secured NFC tag. The URL for the product webpage can be loaded into the NDEF message. Therefore, even if the Infineon NFC verifier mobile application is not installed, this URL can be opened.

After the Secured NFC tag is scanned in the Infineon NFC verifier application, it displays the information of the tag and a link to the product's webpage. Click "View Product webpage" to open the product URL in the browser.

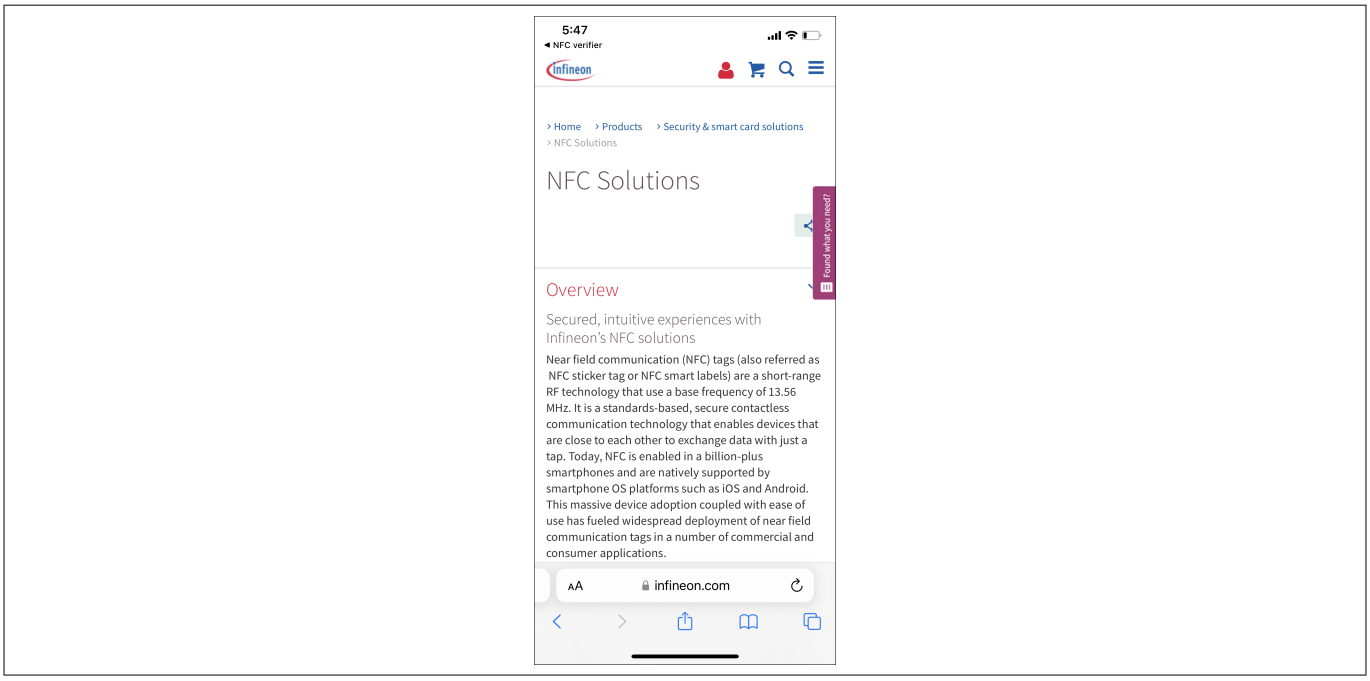

Figure 10 Product webpage

3.1.4.2 Recent transaction log

The recent transaction log screen displays the communication that takes place during brand verification between the mobile application, the Secured NFC tag and the cloud services. This screen stores and displays the logs of the most recent transaction. Older logs are automatically deleted.

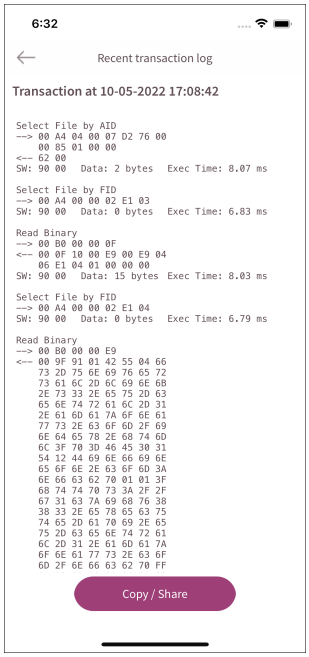

Figure 11 Recent transaction log screen

The following details can be found in the recent transaction log:

- APDU communication between Secured NFC tag and phone
- Cloud service communication - Request and response

To share the transaction log, click the Copy/Share option and choose the appropriate application to share.

3.1.4.3 Main menu screen

The main menu can be opened by selecting the menu icon (3 horizontal lines) on the top right corner. With the help of the main menu, the additional functions of this application are performed.

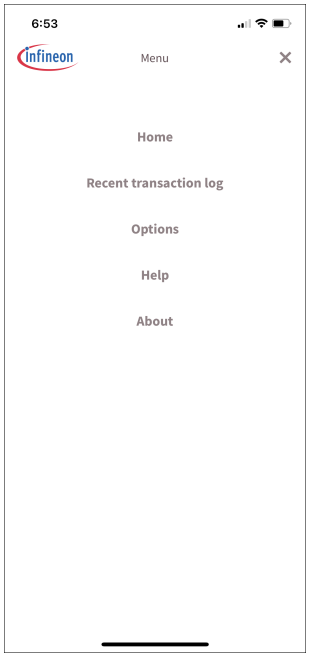

Figure 12 Main menu screen

Recent transaction log screen

For more information on the recent transaction log screen, please navigate to Chapter 3.1.4.2

Options screen

The “auto-open webpage” function is available in the options menu. When this option is enabled, the product webpage will be loaded automatically after the time specified by the 'Product webpage auto-open after (Sec)' value, which can be set up to 15 seconds.

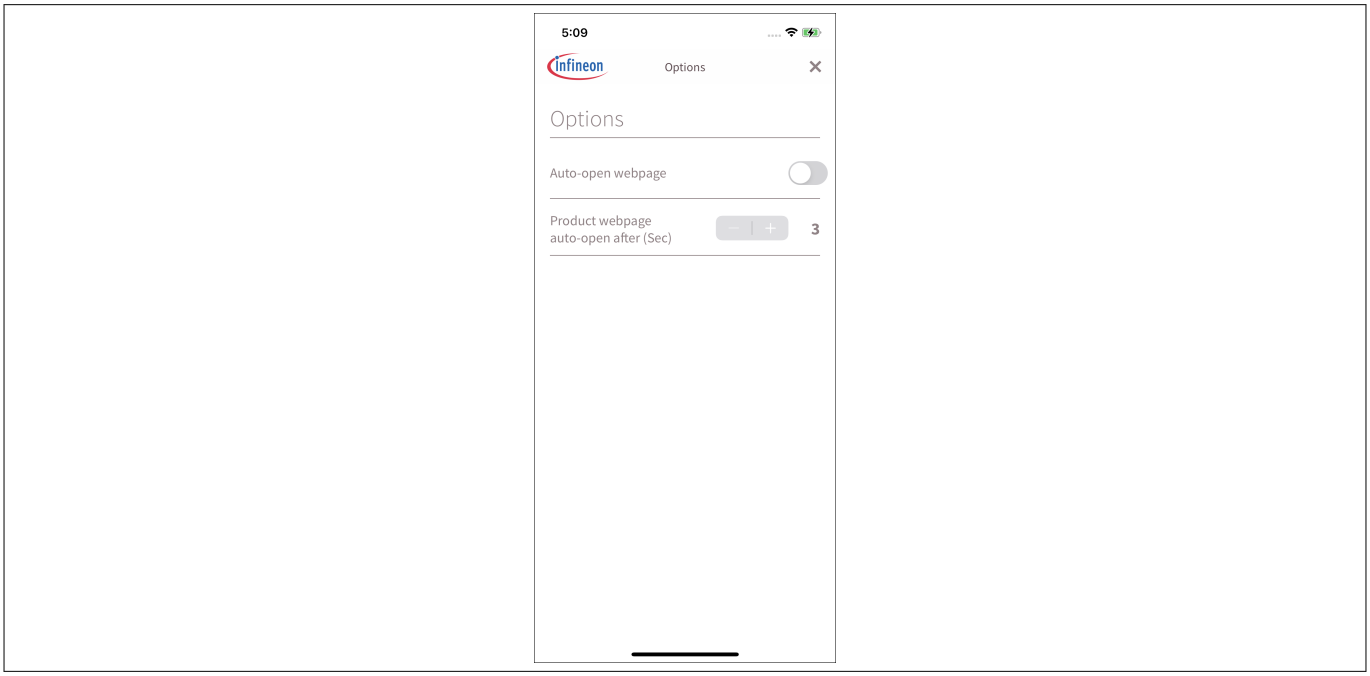

Figure 13 Options screen

Help screen

The help menu provides user assistance inside the mobile application. On clicking the "View user manual", it displays the user guide, which is hosted on GitHub [1].

The inbuilt "Getting Started" files can also be accessed from the help screen. Hyperlinks at the bottom of the help view allow switching among pages, and clicking any topic will display its contents.

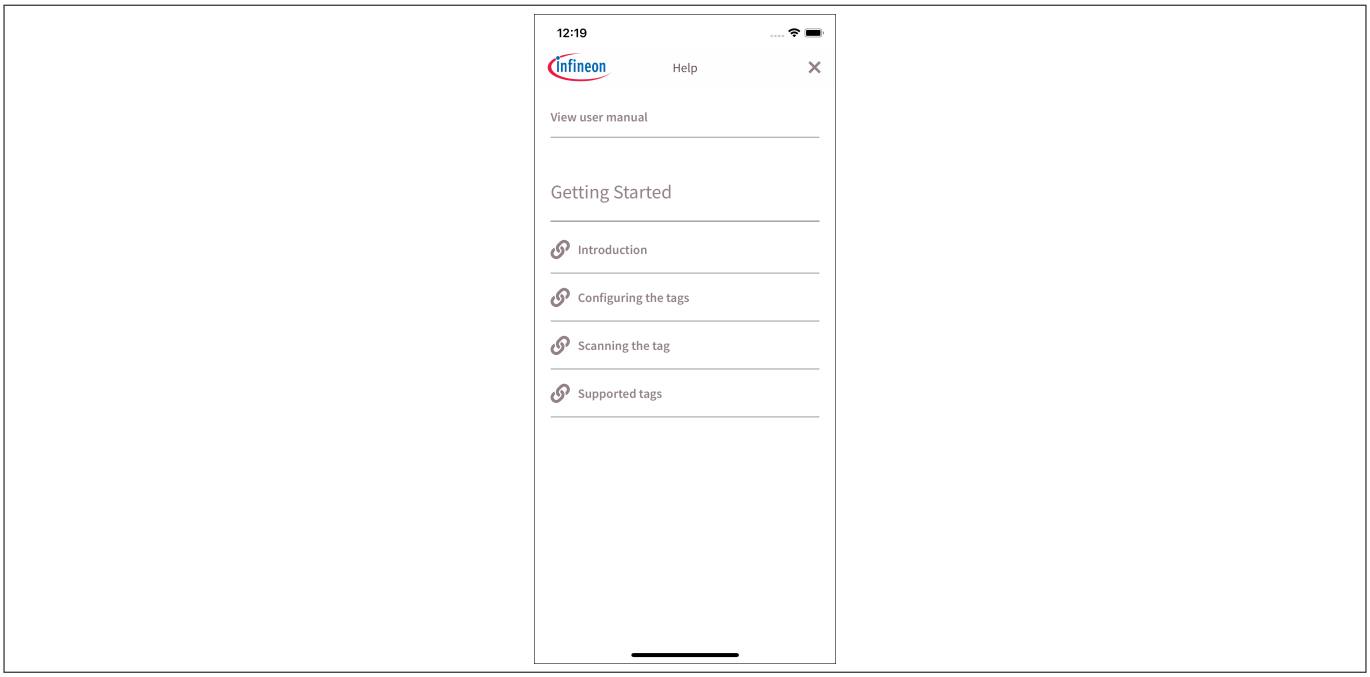

Figure 14 Help screen

About screen

The about menu contains the software details such as the Infineon NFC verifier application version, licenses and other product related information.

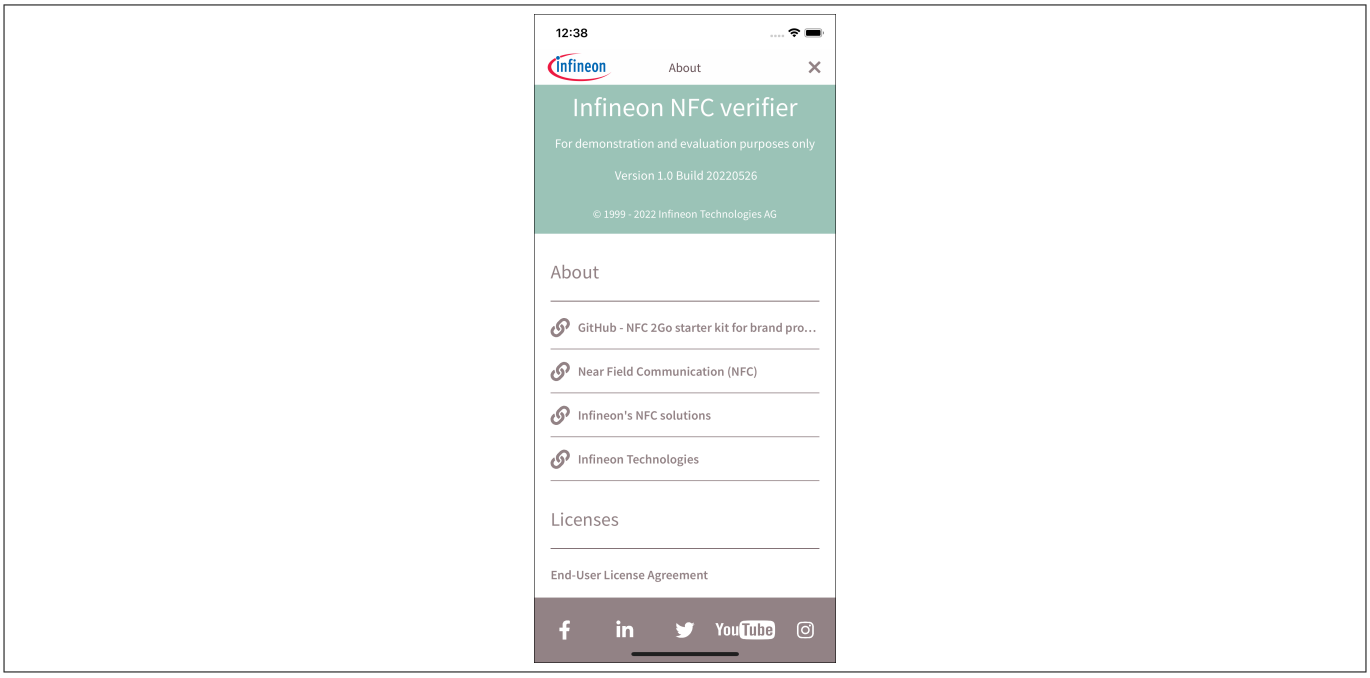

Figure 15 About screen

3.2 Working with the Infineon secured NFC tag IDE tool

The Infineon secured NFC tag IDE is a Windows-based tool that is used by the brand administrators to personalize the Secured NFC tags with cloud-based service.

The Infineon secured NFC tag IDE tool supports the following key features:

- Imports brand protection application profiles into the tag
- Stores the product information and service information in the tag
- Personalizes the brand verification keys and product webpage in the cloud

Note: Brand verification can also be performed using the Infineon secured NFC tag IDE tool to validate the brand verification data on the personalized tag.

3.2.1 Installing the tool

The Infineon secured NFC tag IDE - PC tool for evaluating this starter kit is available for download from the Infineon Development Center [4].

The steps for installing the tool are as follows:

1. Download and extract the zip file.
2. Double-click the installer to launch the install dialog.
3. Choose "Next" in the welcome screen.
4. Read the license agreement, then choose "I accept the terms of the license agreement" and click "Next" to proceed.
5. Click "Next" to install in the default installation directory or choose other location.
6. To begin installation, click "Install".
7. When the installation is finished, click "Finish".
8. After installation, the Infineon secured NFC tag IDE tool can be accessed from the "Infineon secured NFC tag IDE" icon in the desktop, or from the start menu programs.

3.2.2 Personalizing brand protection application profiles

Once the Infineon secured NFC tag IDE tool is installed, follow the steps below:

1. Connect a contactless PC/SC reader to your PC.

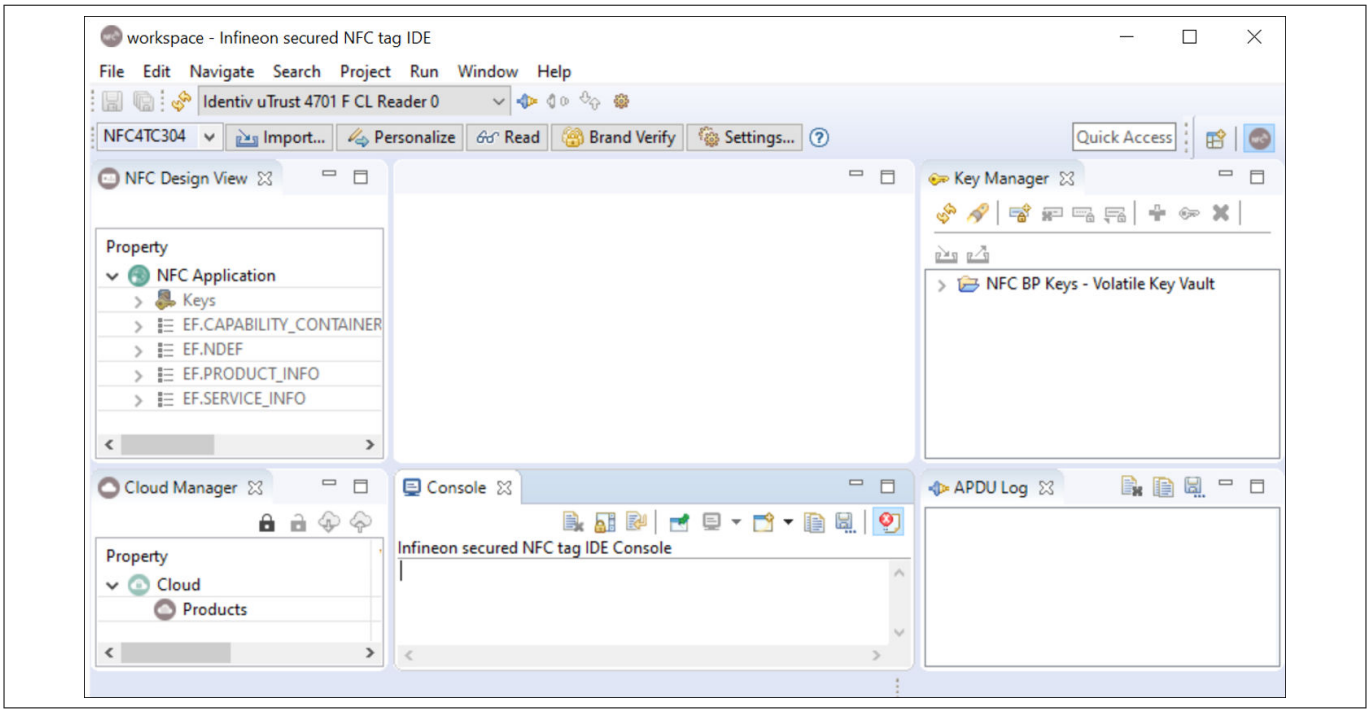

Figure 16 Connecting the reader to the PC

2. Open the "Infineon secured NFC tag IDE" tool.
3. After selecting the workspace, click "OK".
4. Choose a contactless reader from the list of readers.
5. Place the Secured NFC tag in the contactless interface of the reader.
6. Click the "Connect to card reader" icon to connect to the Secured NFC tag.
7. The ATR of the Secured NFC tag will be displayed in the "APDU Log" section on successful connection.

Personalizing brand protection application profiles into the Secured NFC tag

To personalize brand protection application profiles into the Secured NFC tag, follow the steps below:

1. Choose the product type of the Secured NFC tag.

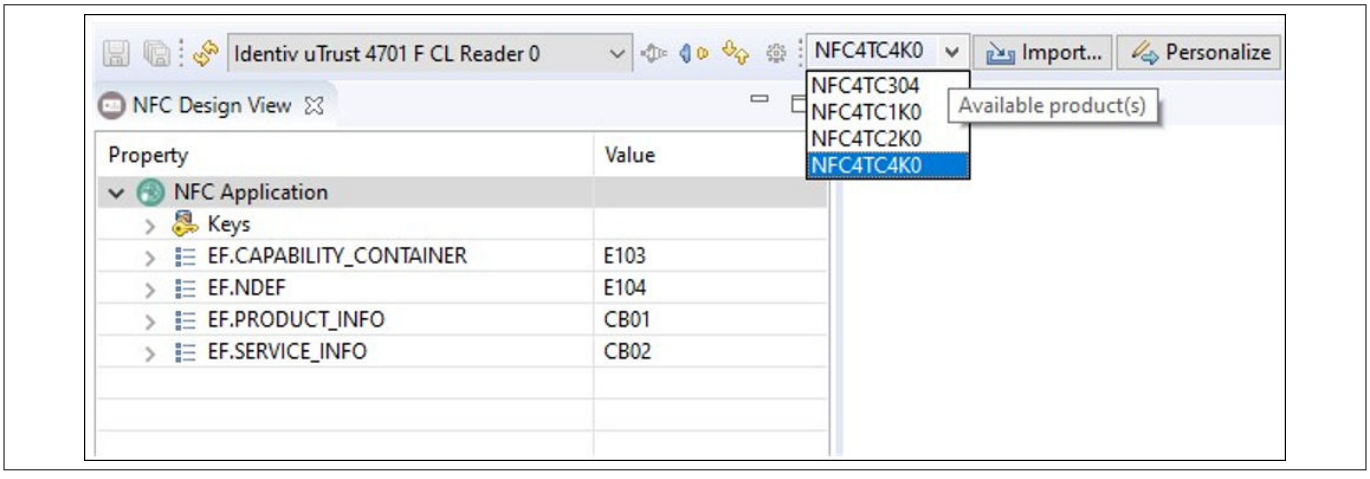

Figure 17 Selecting the product from the drop-down list

2. Right-click the "NFC application", select "Import" and then choose the appropriate profile. The selected profile will be loaded into the model.

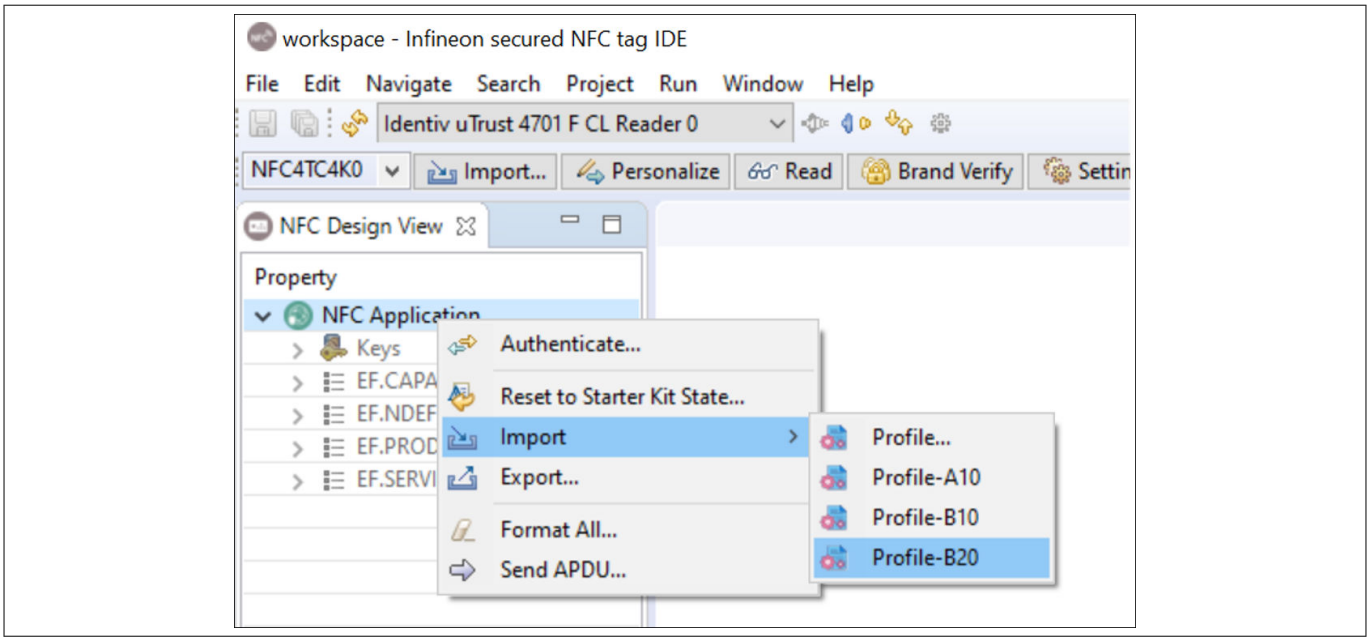

Figure 18 Importing the profile into the model

3. Click "Personalize" to personalize user data in the Secured NFC tag.

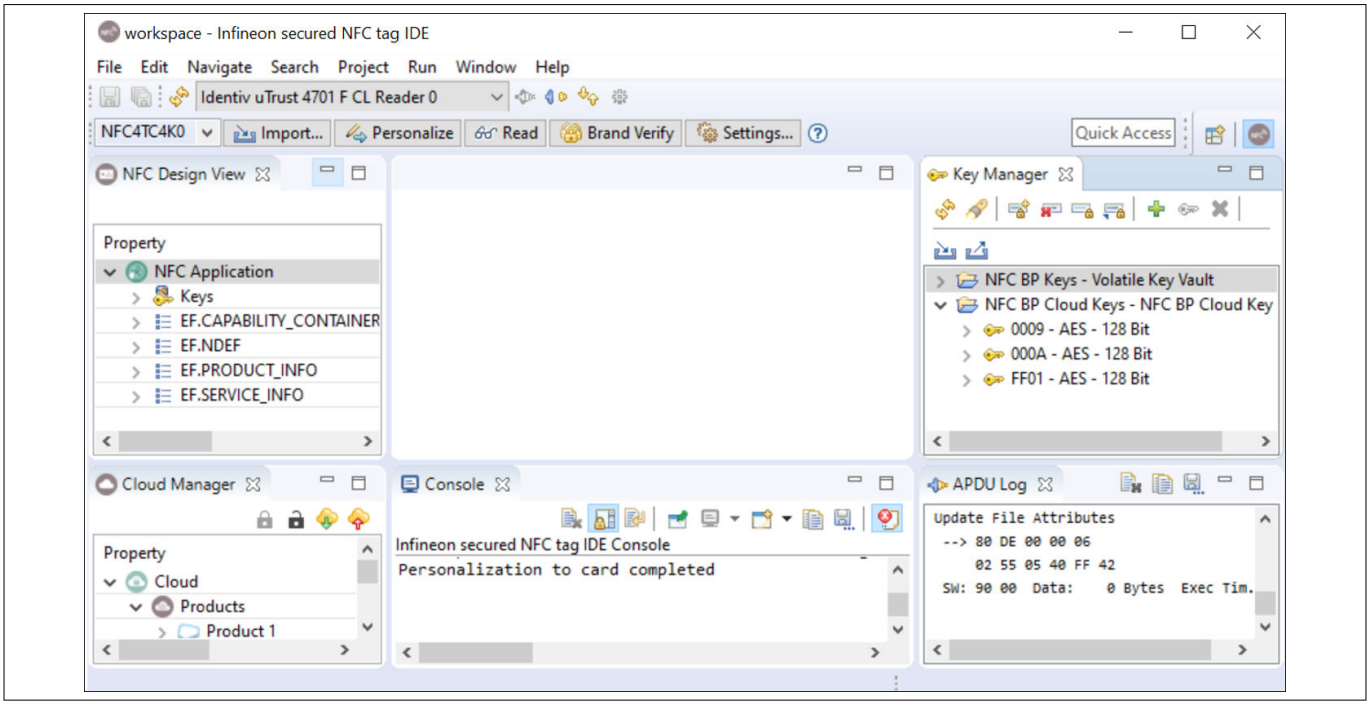

Figure 19 Personalizing the Secured NFC tag

Updating the Secured NFC tag's content

The steps for updating the Secured NFC tag's content are as follows:

1. Double-click any of the "EF.PRODUCT_INFO" or "EF.SERVICE_INFO" or "EF.NDEF" records to open the edit pane.

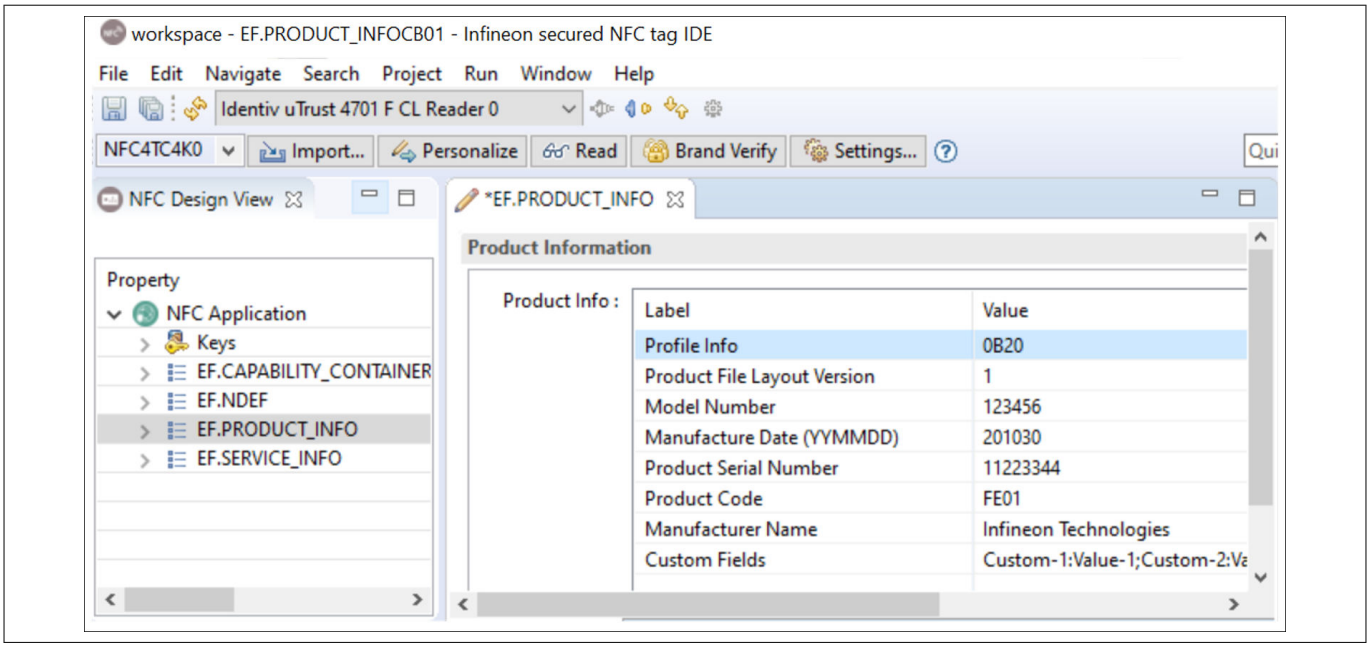

Figure 20 Updating the product information

2. Make the necessary changes in the edit pane.
3. Click the "Save" icon to save the changes to the model in the NFC Design View.
4. Click "Personalize" to update the content in the Secured NFC tag.

3.2.3 Personalizing the keys

Registering for myInfineon account

To access the starter kit evaluation cloud platform, an Infineon account is required. To get a new account,

1. Open Infineon website (www.infineon.com).
2. Navigate to the "myInfineon" menu in the browser and select "Register for myInfineon".
3. Fill out your account details and sign up for an account.
4. An email with a verification code will be sent to the registered email address. Click the "Activate now" button in the email to activate the account.

Updating brand verification key in the cloud

The steps for updating the brand verification keys in the cloud are as follows:

1. Click the "Login to server" icon in the "Cloud Manager" pane.
2. Enter your username and password for the myInfineon account.

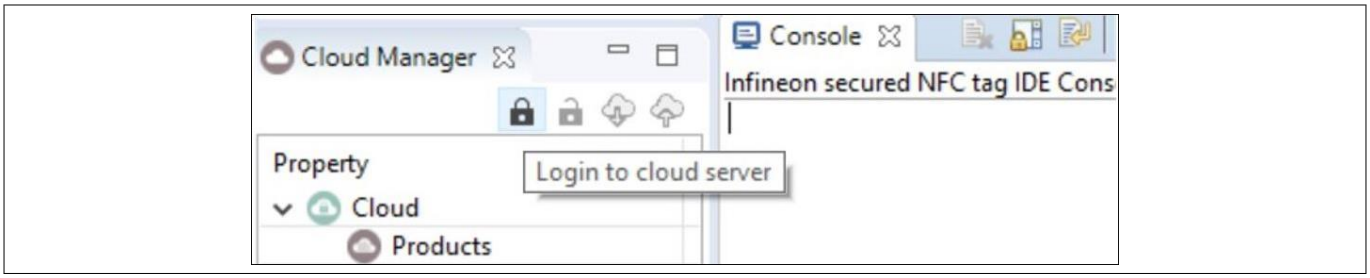

Figure 21 Logging into the cloud server

3. When the login is successful, the keys and product information are retrieved from the cloud.

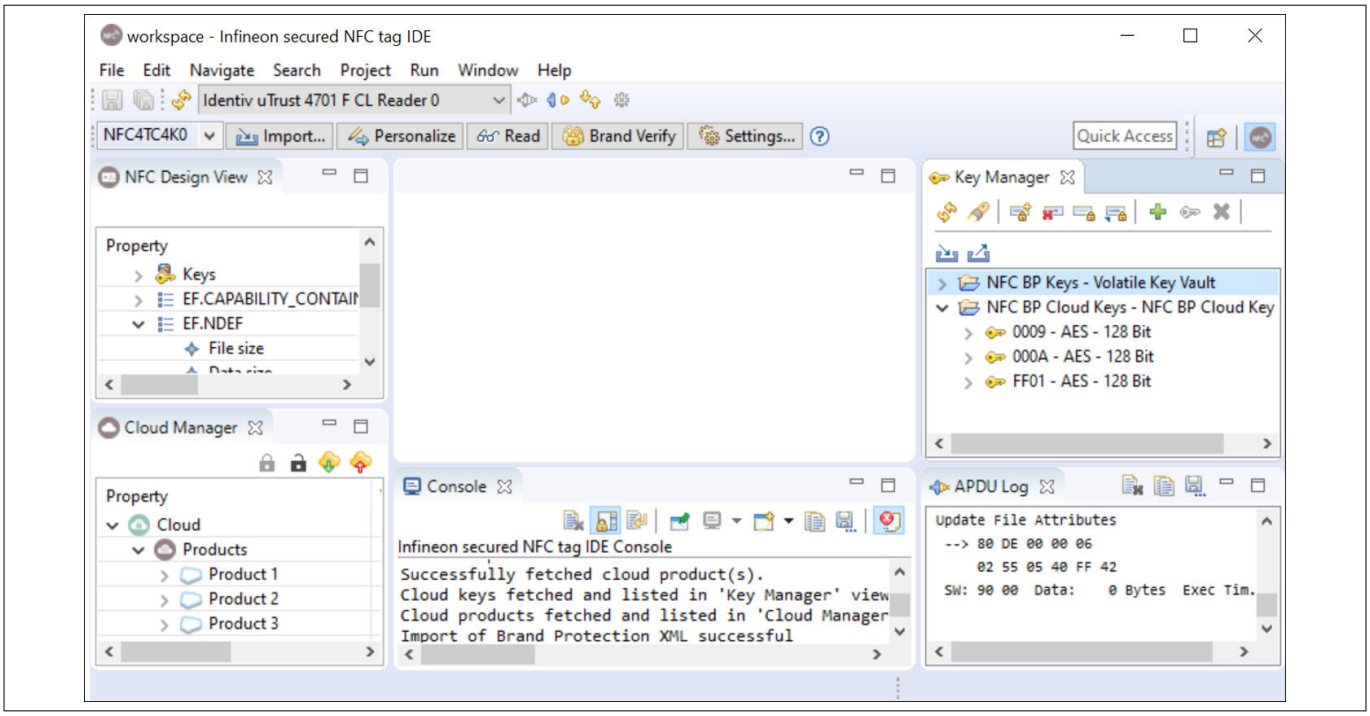

Figure 22 Successful login to the cloud server

4. Right-click the key to be edited and select the "Edit key".
5. Enter the key value in the "New component value" field and choose "Set value".
6. Click "OK" to confirm overwriting the existing key.

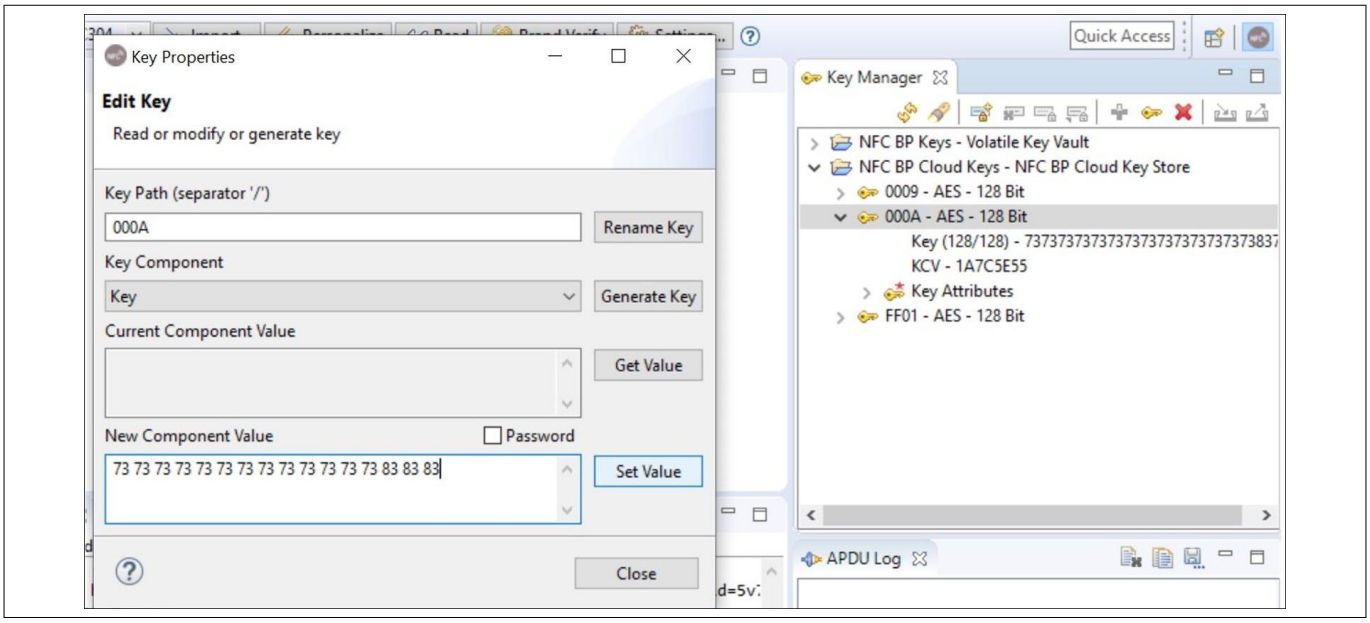

Figure 23 Editing the keys in the key properties view

7. Wait for the console to display a success message indicating that the key is updated in the cloud.

Updating the brand verification key in the Secured NFC tag

Note: When the key in the Secured NFC tag is updated, the user must manually update the key-label in the brand protection record. This key-label update is not performed automatically by Infineon secured NFC tag IDE.

To use the updated key in the tag, follow the steps below:

1. In NFC Design View, expand the "Keys" under the NFC application.
2. Right-click the verification key and select the "Edit Key".

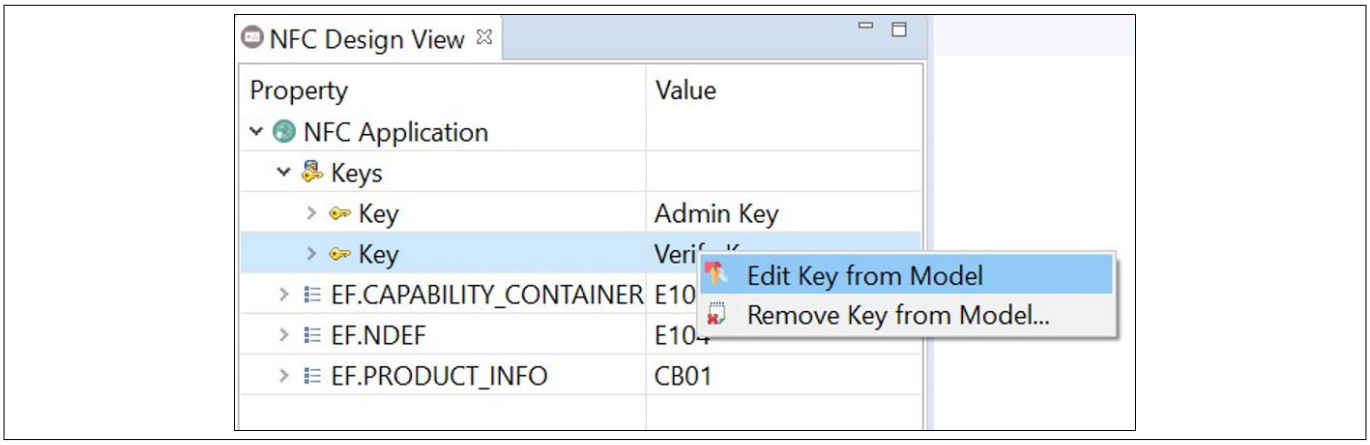

Figure 24 Editing the keys in the NFC application

3. Click "KeyVault" and select the key to be updated.

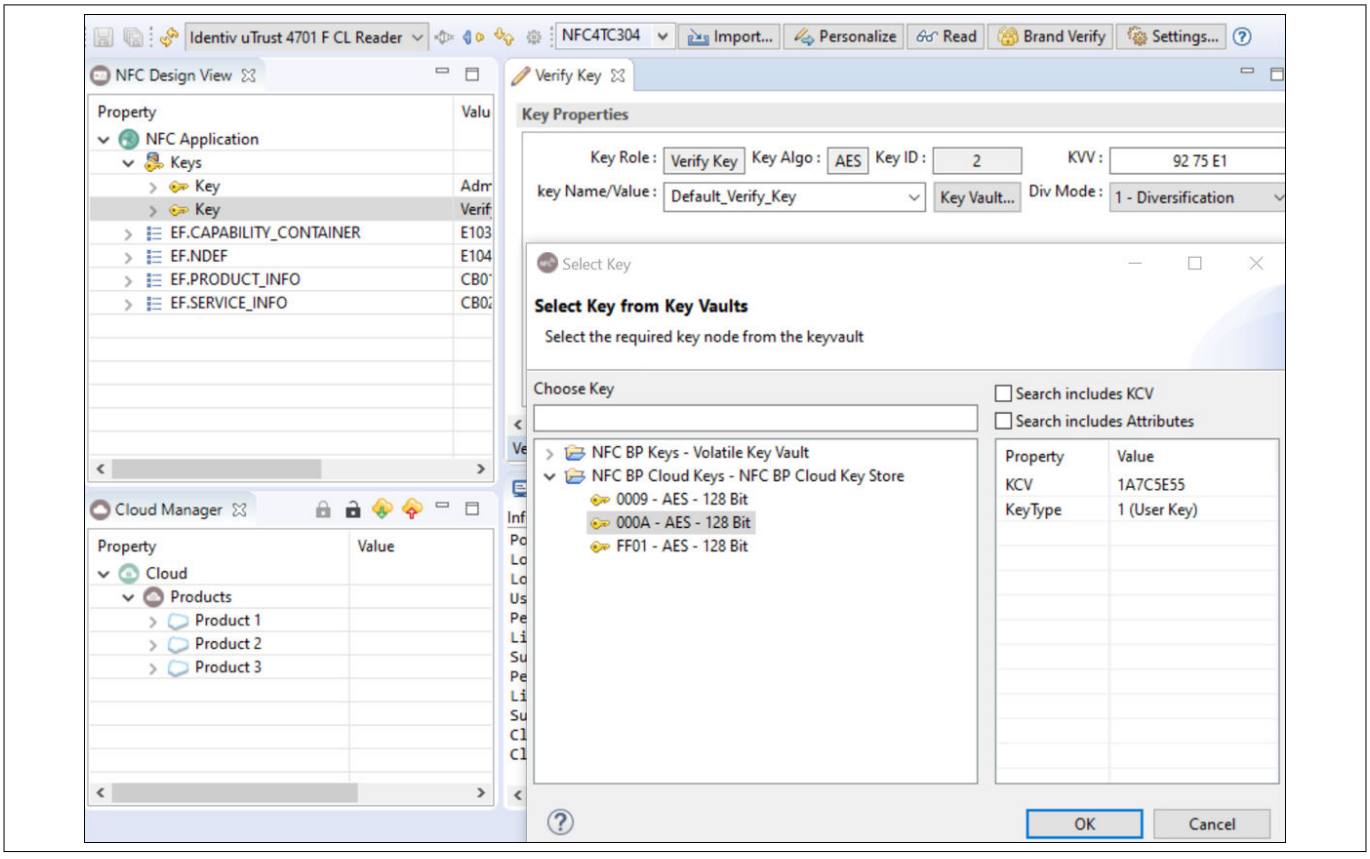

Figure 25 Selecting the key from Key Vaults

4. Click the "Save" icon to save this information to the model.
5. Expand the "EF.NDEF" and "NFC-Records".
6. Double-click the "Brand Protection" record.
7. Edit the Key Label with the key label of the verification key to be updated.

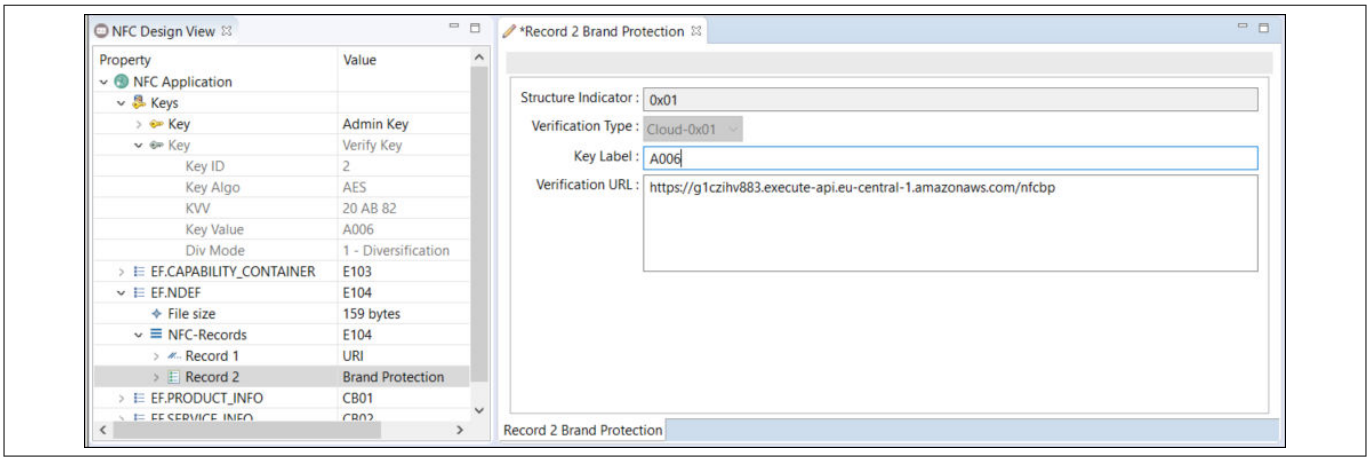

Figure 26 Editing the Brand Protection record

8. Click the "Save" icon to save the information to the model.
9. Click "Personalize" to update the key in the Secured NFC tag.

3.2.4 Personalizing the product webpage link

Updating product webpage URL in the cloud

To update the product webpage link in the cloud, then follow the steps below:

1. Login into the cloud manager.
2. Right-click the product to be edited and select "Edit Product".
3. Edit the "Product webpage" with the product webpage URL.

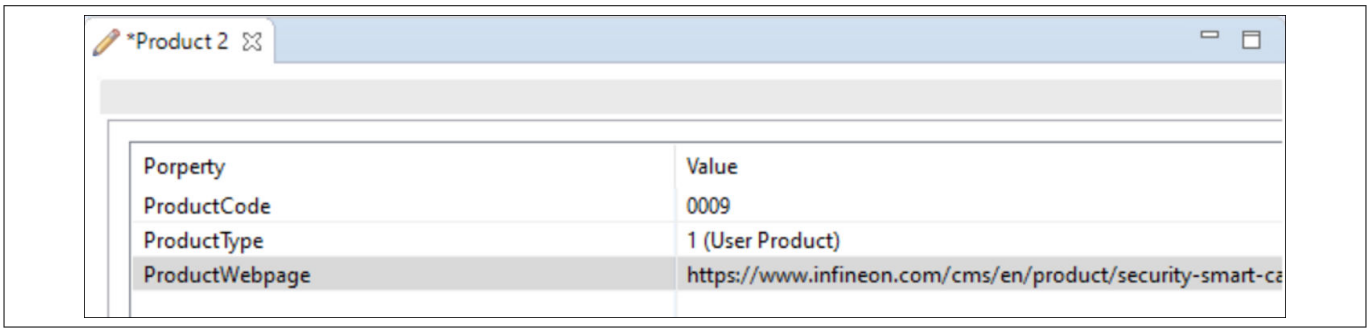

Figure 27 Editing the URL of the product

4. Click "Save" icon to save the changes to the model.
5. Click "Save product to cloud" icon to save the products to cloud.
6. Wait until the success message appears in the console.

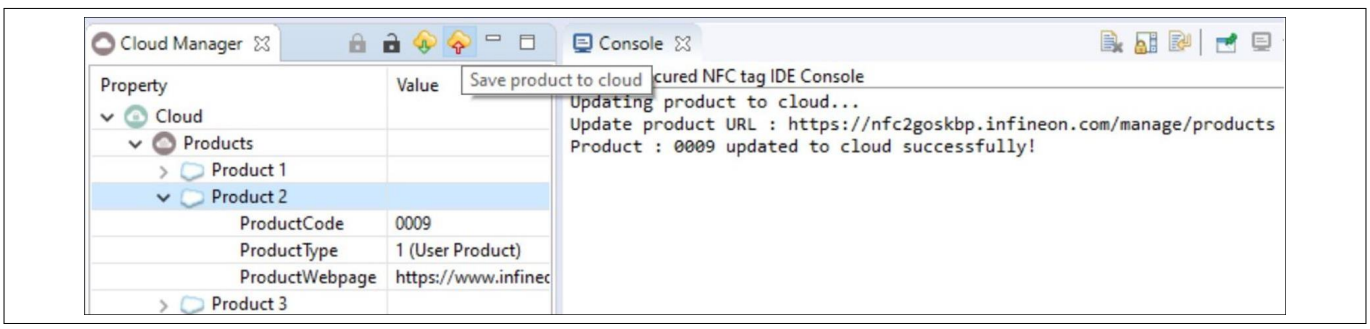

Figure 28 Updating product information into the cloud

Configuring reference to product webpage URL in the Secured NFC tag

Note: The product code in the landing webpage URL must be manually updated. Infineon secured NFC tag IDE does not automatically update the landing page URL with product code.

To update this product webpage in the tag, then follow the steps below:

1. In the NFC Design View, expand the "EF.NDEF" and "NFC-Records"
2. Double-click the "URI" record.
3. Edit the URI with the URL of the starter kit landing webpage.
4. Append a query parameter "?p=<PROD_CODE>" at the end of the URL. Replace <PROD_CODE> with the product code of the product.
5. Click the "Save" icon to save the changes to the model.
6. Click "Personalize" to update the information in the Secured NFC tag.

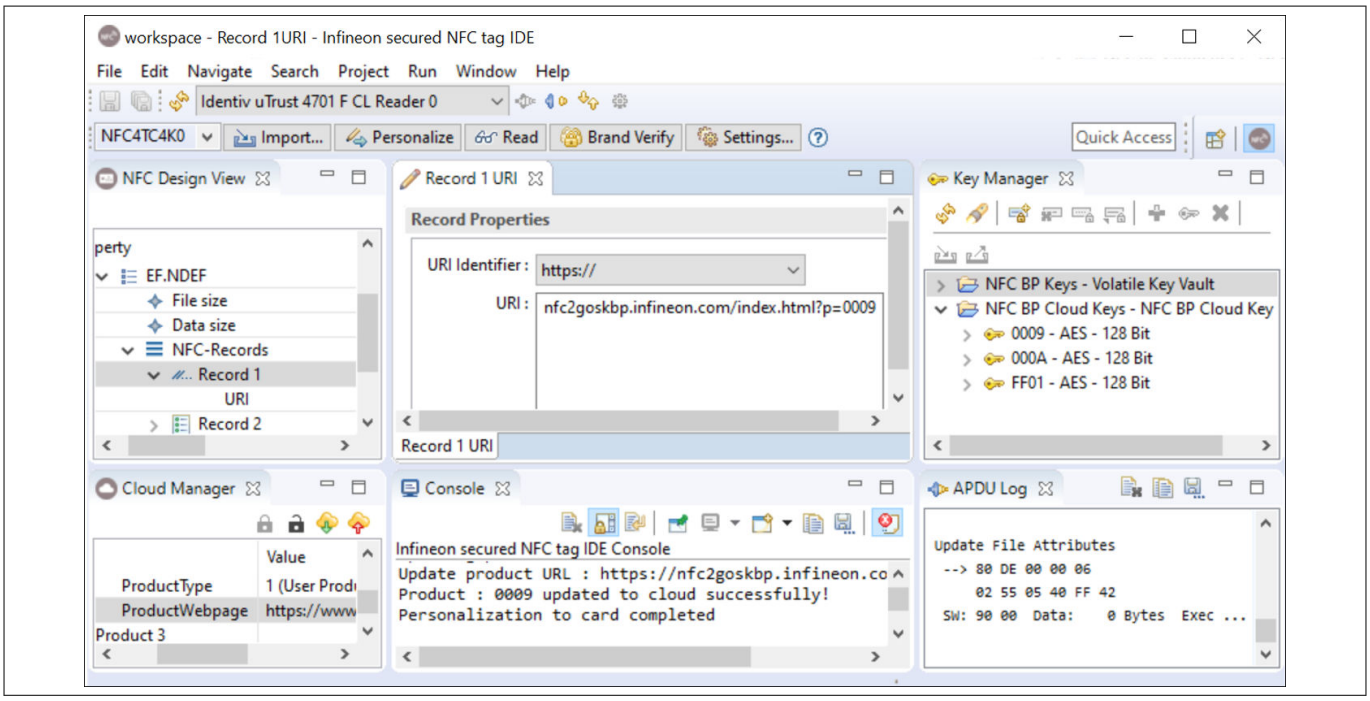

Figure 29 Editing URI record

4 Developing solutions using the starter kit

The starter kit provides the source code for developing quick prototype solutions for brand protection.
The requirements for developing custom solutions are as follows:

- Amazon Web Services (AWS) cloud account
- Mac with Xcode (V12 or higher) for iOS application development
- Android Studio (V4.2 or higher) for Android application development

4.1 Brand verification process

During brand verification, the verification key in the Secured NFC tag is mutually authenticated against the verification key in the cloud. The mobile application serves as a tunnel between the Secured NFC tag and the brand verification cloud service for authentication.

Figure 30 shows the high level interaction between the Secured NFC tag, the Infineon NFC verifier mobile application and the brand verification cloud service during the brand verification process.

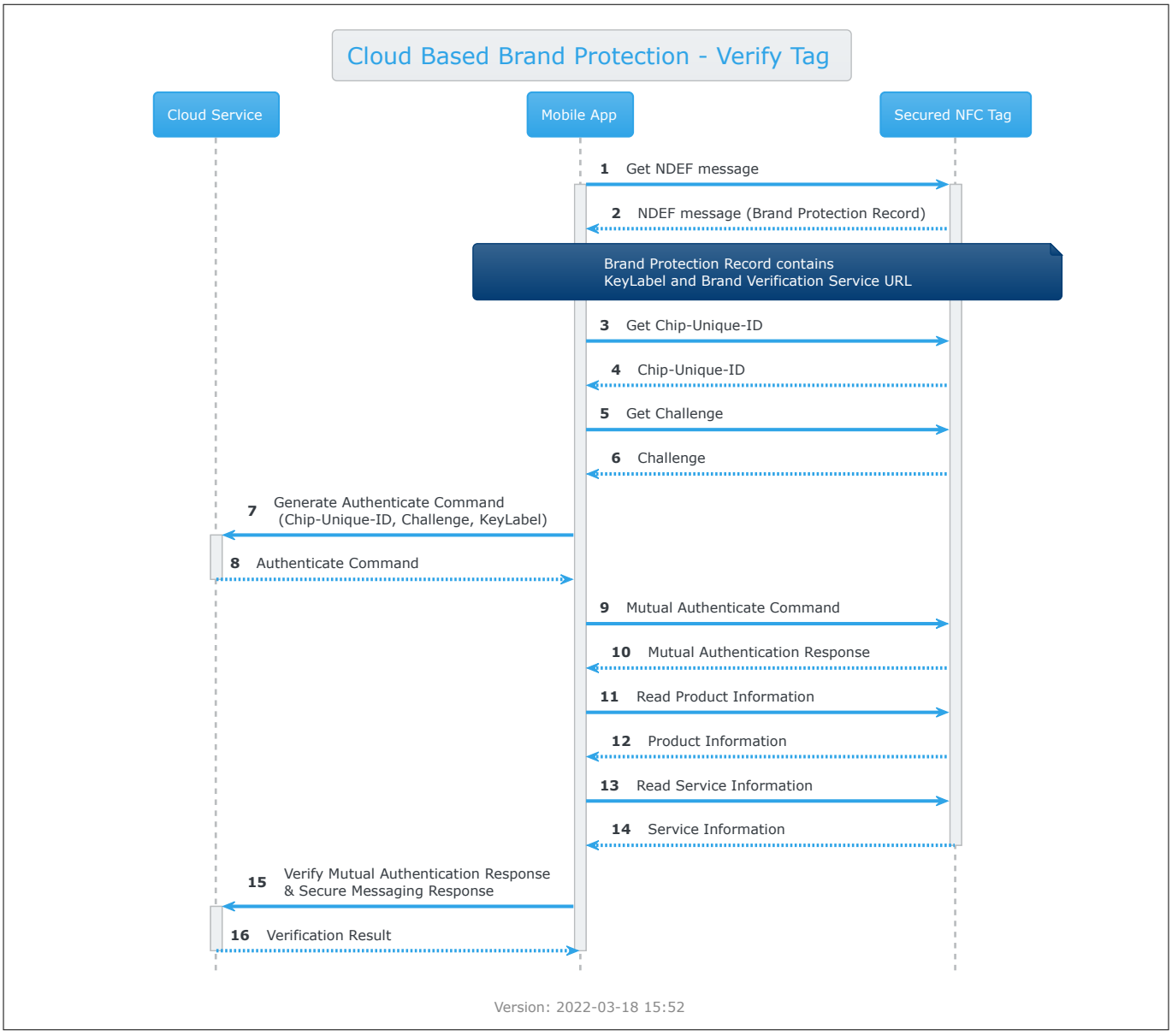

Figure 30 Cloud-based brand verification

Exchange of commands between the Infineon NFC verifier mobile application and the Secured NFC tag:

The Infineon NFC verifier mobile application and the Secured NFC tag communicates via application protocol data units (APDUs). When the Secured NFC tag is placed near the range of the phone's NFC reader during the brand verification operation, the exchange of APDU command and response takes place.

This operation is split into two parts. They are as follows:

1. NFC Forum Type 4 Tag operation:

This operation reads the NDEF message from the Secured NFC tag, allowing the content of brand verification record to be extracted.

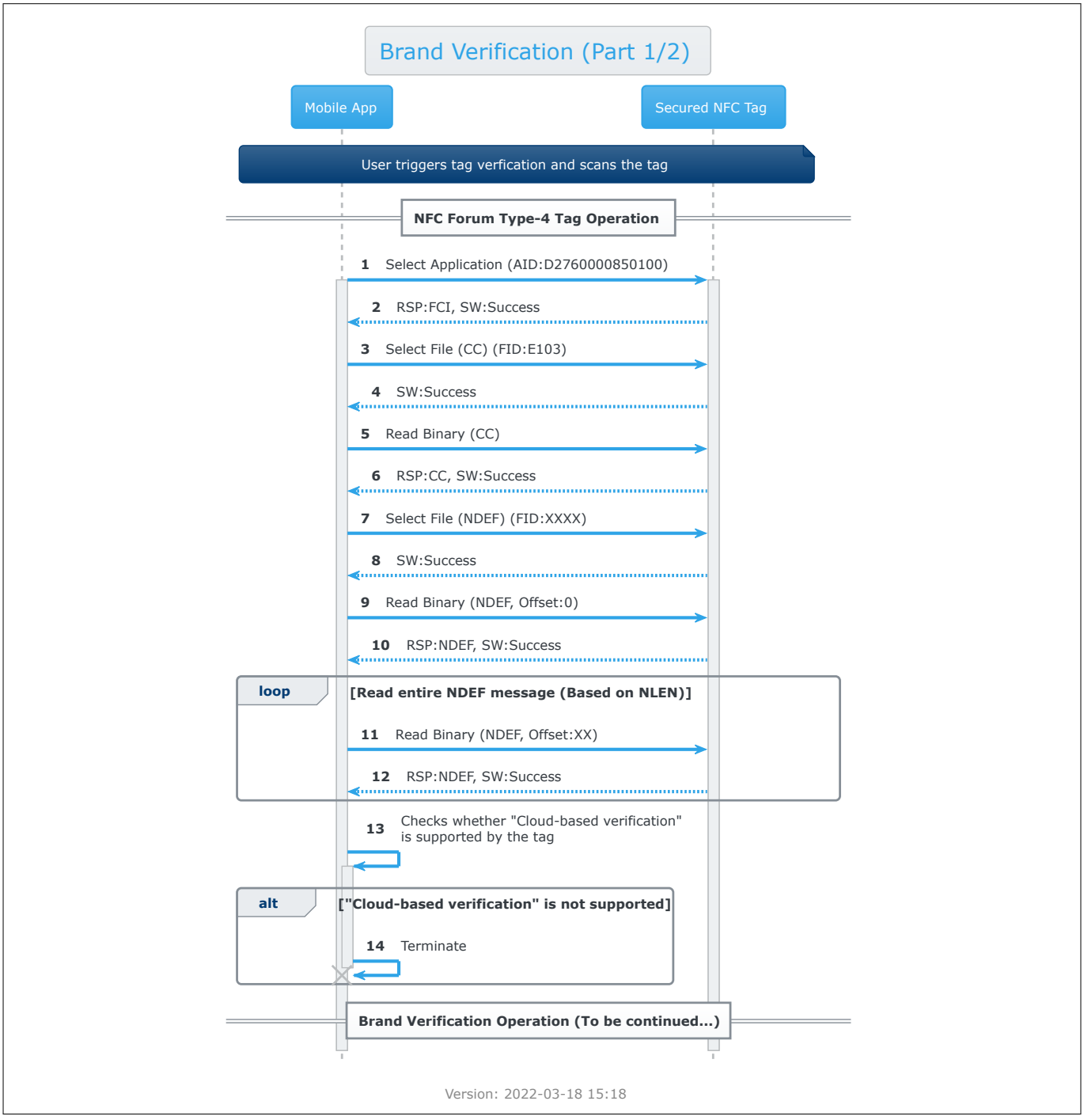

Figure 31 APDU-level communication during the NFC Forum Type 4 Tag operation

2. Brand verification operation:

This operation involves proprietary APDU command-response sequences to read the ChipID, to trigger the generation of the challenge, and to perform the mutual authentication.

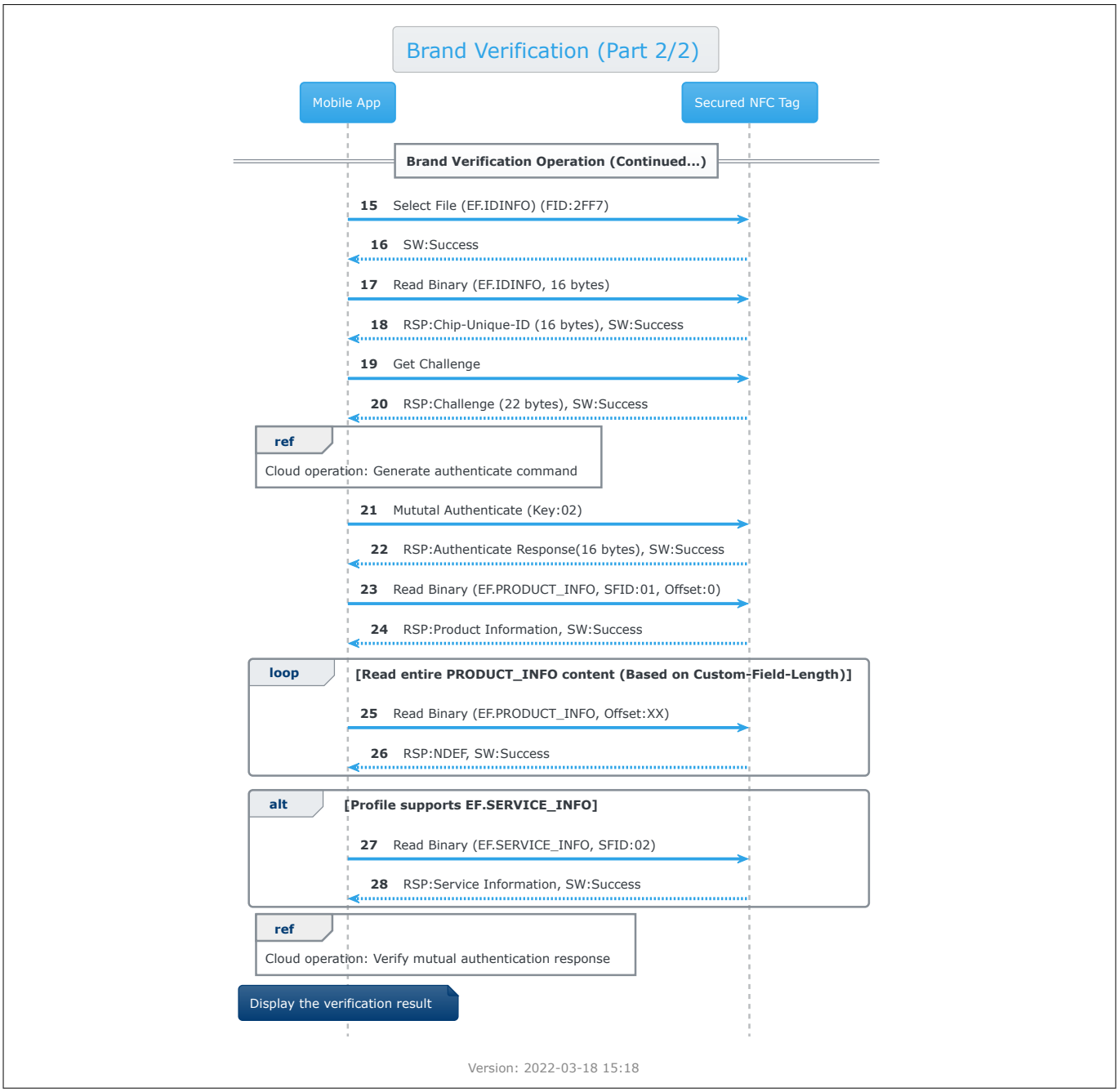

Figure 32 APDU-level communication during the brand verification operation

Communication between the Infineon NFC verifier mobile application and the brand verification cloud service:

The brand verification cloud service can be accessed through standard HTTP API requests from the mobile application. The request and response are in JSON format, which the mobile application creates and parses.

For more information on API details, refer to Chapter 4.5.

4.2 Tag file system

The Secured NFC tag contains a brand protection application that is based on NFC Forum Type 4 Tag file structure, customized with proprietary files and authentication functionality for brand protection use case. This application is based on Mapping Version 1.0 of the NFC Forum Type 4 Tag structure and therefore, it uses the application identifier (AID) "D2 76 00 00 85 01 00"$_H$.

The files in the brand-protection application are as follows:

- Capability Container (CC) file (FileID: E103$_H$)
- NDEF file (FileID: E104$_H$)
- Product information file (FileID: CB01$_H$, SFID: 01$_H$)
- Service information file (FileID: CB02$_H$, SFID: 02$_H$)

The presence of these files and their sizes are defined as per brand protection application profiles.

The keys stored in the Secured NFC tag are described in the "Authentication Keys" section.

Brand protection application profiles:

This starter kit predefines a set of brand protection application profiles which can be used in the Secured NFC tag based on the application use case and the memory size supported by the Secured NFC tag.

The application profiles defined in this starter kit are as follows:

- Profile A10
- Profile B10
- Profile B20

Table 1 shows the file sizes of each file of these profiles.

Table 1 File size of the predefined application profiles

Elementary Files	Profile A10	Profile B10	Profile B20
CC file	15	15	15
NDEF file	224	256	256
Product info file	64	624	3696
Service info file	File not present	128	128
Total	303	1023	4095

Note: All sizes are indicated in bytes in Table 1

Table 2 summarizes the supported profiles for the Secured NFC tag products.

Table 2 Supported profiles for Secured NFC tag products

Applicable product	Profile A10	Profile B10	Profile B20
NFC4TC304	Supported	Not supported	Not supported
NFC4TC1K0	Supported	Supported	Not supported
NFC4TC2K0	Supported	Supported	Not supported
NFC4TC4K0	Supported	Supported	Supported

4.2.1 CC and NDEF files

The CC and NDEF files follow the NFC Forum Type 4 Tag NDEF application definition. The NDEF file contains an NDEF message with two records. They are as follows:

1. URI record
2. Brand protection record

Note: Additional records can be added based on application needs.

4.2.1.1 URI record

The URI record is an NFC Forum Well Known Type URI record that contains an URL. When the Secured NFC tag is tapped on an NFC-enabled device, this URL is expected to be opened in the browser of the device.

This record is configured with the URL of the starter kit landing webpage, which allows users to download the brand verification application and view the product webpage. This URL record can be configured to meet the application requirements.

Table 3 shows the structure of the URI record.

Table 3 URI record structure

Length (bytes)	**Payload header**	**Type**
1	TNF: Well Known (01_H), short record (SR)	Byte
1	Type length	Byte
1	Payload length	Byte
1	Type (U)	String
Length (bytes)	**Payload**	**Type**
1	Protocol field (https://)	Byte
n	URI field	String

For more details on the URI record, refer to NFC Forum URI Record Specification [13].

4.2.1.2 Brand protection record

The brand protection record is an external type record that stores the information used for brand verification. This record stores the brand verification methods that are supported, and the verification data for each method.

Note: Only the cloud based verification method is defined in this record, and the other verification types will be defined in future.

This external type record is defined with the type: **infineon.com:nfcbp**

Table 4 shows the structure of this record.

Table 4 Brand protection record structure

Length (bytes)	**Payload header**	**Type**
1	TNF: External type (04_H), short record (SR)	Byte
1	Type length	Byte
1	Payload length	Byte

(table continues...)

Table 4 (continued) Brand protection record structure

18	Type ("infineon.com:nfcbp")	String
Length (bytes)	**Payload**	**Type**
1	Structure indicator: 01_H	Byte
1	Verification type: • Cloud: 01_H • PKI: 02_H • Blockchain: 04_H	Byte
	#1: Cloud verification data	
1	Length of verification URL (n)	Byte
n	Verification URL	String
2	Key label	Bytes
	#2: PKI data	
	To be defined in future releases	
	#3: Blockchain data	
	To be defined in future releases	

The structure indicator field indicates the structure in which the brand protection record payload is formatted. The verification type indicates the brand verification methods that are supported. This version only supports cloud-based verification, and therefore PKI data (#2) and Blockchain data (#3) fields are not populated.

Cloud verification data must be present in the first order if cloud-based verification is supported.

Verification URL: Refer to Chapter 4.5.1.1 for the URL of the brand verification cloud service endpoint.

Key label: This is the key label of the cloud-referenced brand verification key that will be used for verification.

4.2.2 Product information file

The product information file is a proprietary file used to store the details of the product. This file is intended to be updated by the brand administrator during the product's manufacturing process. The Infineon NFC verifier application can only read this information and display it to the users.

Table 5 displays a set of product information and its data structure for each profile defined in the starter kit.

Table 5 Product information and its data structure

Profile A10	**Profile B10**	**Profile B20**	**Product info**	
Length (bytes)	**Length (bytes)**	**Length (bytes)**	**Value**	**Type**
2	2	2	Profile Info (ID, Version)	Bytes
1	1	1	Product File Layout Version: 01_H	Byte
8	8	8	Model Number	String
3	3	3	Manufacture Date	Date (yymmdd)
8	8	8	Product Serial Number	String
2	2	2	Product Code	Bytes

(table continues...)

Table 5 (continued) Product information and its data structure

21	64	64	Manufacturer's Name	1 byte length + String
19	536	3608	Custom Field	2 byte length + Delimited String (: to separate label-value) (; to separate fields)

Note:
- *If the length of the string value field is less than the specified length of the field, the remaining characters are filled with spaces*
- *If the length of a byte array is less than the defined size, the remaining bytes are filled with 00_H*

The profile info field indicates the loaded profile. If profile B20 is loaded, the profile info contains $0B20_H$. The product file layout version specifies the structure in which the product information file is formatted.

In addition to the predefined fields, the product information file supports custom fields. Custom fields are stored as label-value pairs separated by a colon (:) and the fields separated by a semi-colon (;). This field is prefixed with the length of the custom field data, which is represented as 2 bytes.

4.2.3 Service information file

The service information file is a proprietary file used to store the usage details of the product. This file is intended to be updated by the brand administrator after the purchase of the product by the users. The Infineon NFC verifier application can read this information and display it to the users.

Table 6 displays a set of service information and its data structure for each profile defined in the starter kit.

Table 6 Service information and its data structure

Profile A10	Profile B10	Profile B20	Service info	
	Length (bytes)	Length (bytes)	Value	Type
File not present	1	1	Service File Layout Version: 01_H	Byte
	3	3	Purchase Date	Date (yymmdd)
	3	3	Warranty Validity Date	Date (yymmdd)
	3	3	Last Service Date	Date (yymmdd)
	118	118	Custom Field	2 byte length + Delimited String (: to separate label-value) (; to separate fields)

In addition to the predefined fields, the service information file also supports custom fields. Custom fields are stored as label-value pairs separated by a colon (:) and the fields separated by a semi-colon (;). This field is prefixed with the length of the custom field data, which is represented as 2 bytes.

Note: The data structure of the product information file and service information file can be customized based on application needs, and the respective logic to parse this data structure in the terminal applications such as mobile application and tools should be updated.

4.3 Keys in the Secured NFC tag

The brand protection application in the Secured NFC tag contains the following keys:

1. Administration key
2. Brand verification key

The administration key is used to perform tag administration operations such as create/delete application and files, update keys and files and so on.

The brand verification key is used to authenticate the product during the brand verification operation. Figure 33 shows the relationship between the brand verification key in the Secured NFC tag and the cloud database.

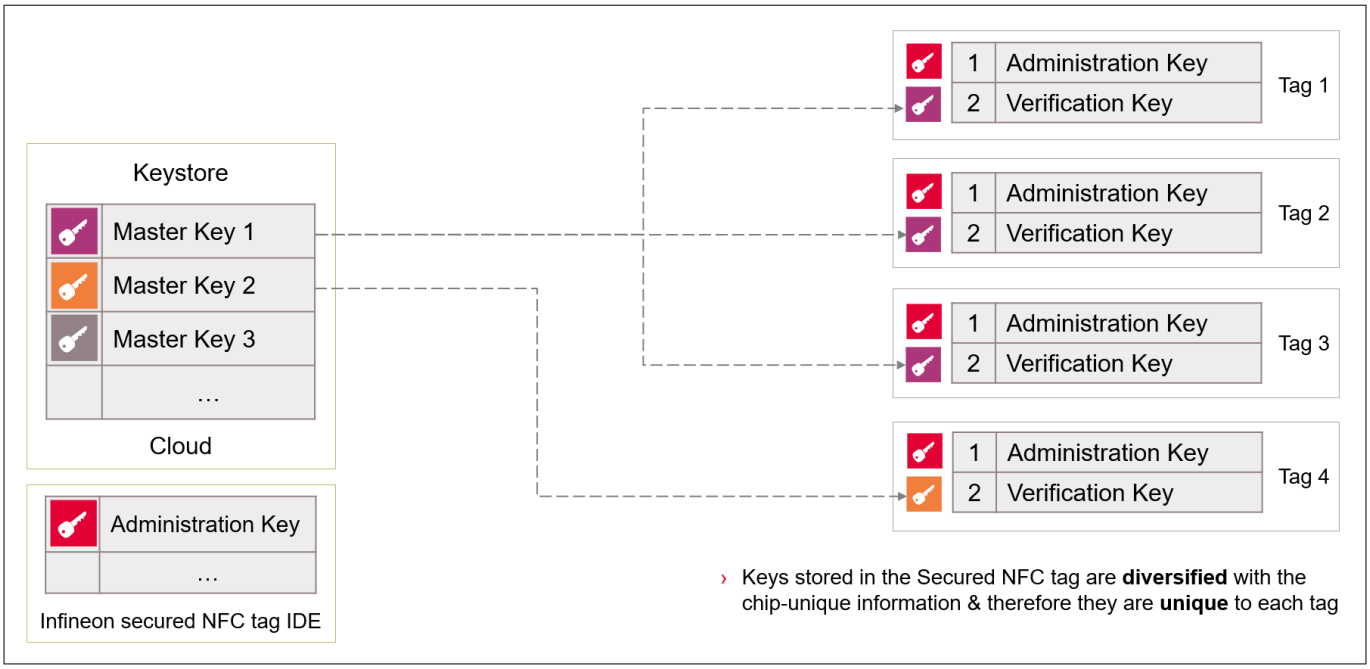

Figure 33 Brand verification key

Both the keys are diversified with the chip-unique information from the tag, and therefore each tag will have a unique administration key and a unique brand verification key.

4.4 Cloud architecture

Brand verification service:

The brand verification service is the authentication service that provides HTTP APIs to verify the authenticity of the Secured NFC tag. It generates the authentication command to be sent to the tag and verifies the authentication response from the tag using a Java-based authentication library. This service is used by the Infineon NFC verifier application or the Infineon secured NFC tag IDE tool during the brand verification process.

Key and product management services:

The key and product management services provide HTTP APIs to manage the keys and product details stored in the cloud. These services are used by the Infineon secured NFC tag IDE tool to list and update keys and product information in the cloud.

Starter kit landing webpage:

The starter kit landing webpage includes links to download the Infineon NFC verifier application from the app stores and a link to view the product webpage URL.

This URL [5] is pre-configured in the starter kit Secured NFC tags.

Figure 34 shows the architecture of the cloud.

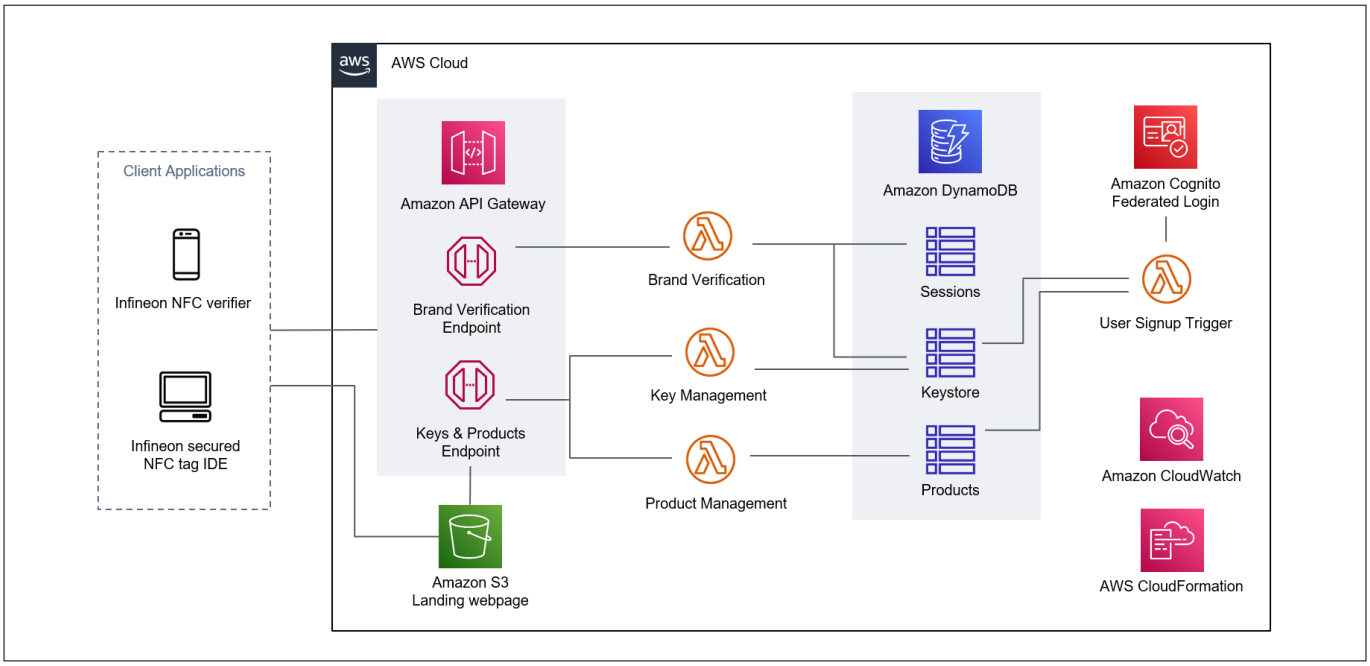

Figure 34 Cloud architecture

Figure 35 shows the relation between the landing webpage and the product webpage.

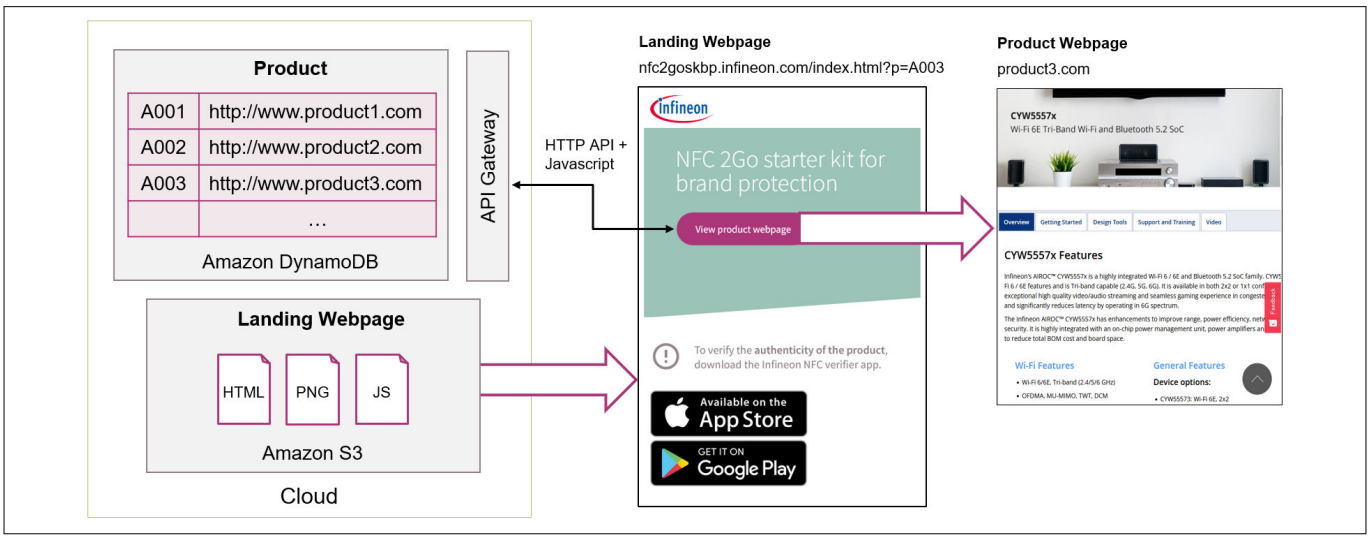

Figure 35 Landing webpage and product webpage

The landing webpage is a simple webpage with links to download the app from stores and a link to the product webpage. The use of a landing webpage in the tag allows the users, who do not have the Infineon NFC verifier mobile application, to install it. However, based on the application specific requirements, the direct end-product URL can be stored in the tag instead of this landing webpage.

The URL of the landing webpage includes a URL parameter (p) for entering the product code (example: https://nfc2goskbp.infineon.com/index.html?p=FE01). The landing webpage contains JavaScript, which retrieves the product code from the URL parameter (p) and invokes an API to retrieve the product webpage URL that is used in the "View product webpage" link.

4.5 Server-side APIs for brand verification

4.5.1 Brand verification API

The brand verification includes generation of mutual authentication data in the cloud and verification of the tag response in the cloud.

4.5.1.1 Brand verification service endpoint

For the NFC brand verification service endpoint URL, refer to [6].

Notes:

1. *This service endpoint URL should be retrieved from the brand protection record available in the tag.*
2. *These APIs do not need AWS Cognito user authentication.*

4.5.1.2 Secure messaging – Generate mutual authentication data

This API is used to generate the mutual authentication data based on the information received from the tag.

Table 7 API – Generate mutual authentication data

	Detail			
API	sm/generate-ma			
Request Method	POST			
Path Parameters	None			
Request Type	application/json			
Request Parameters	ChipID	String	16 bytes	Chip-unique ID from the EF.IDINFO
	Challenge	String	22 bytes	Response data of the Get-Challenge command
	KeyLabel	String	2 bytes	Parsed key label from the tag

Request Sample	<pre>{
	"ChipID": "05 02 05 6D DE BB 82 03 E9 61 FC 8D AF 21 18 00",
	"Challenge": "92 83 6C EA 06 B3 B9 12 1C 56 3C C4 54 40 B8 15 DC 9D E3 92 C4 4D",
	"KeyLabel": "FF01"
	}</pre>			

Response Type	application/json			
Response Parameters	Version	String	5	Version of the API
	SessionID	String	36	Session ID of the verification session
	CommandData	String	38 bytes	Mutual authentication command data

(table continues...)

Table 7 (continued) API – Generate mutual authentication data

	Detail

Response Sample	{
	"Version": "01.08",
	"SessionID": "fe606ab6-769d-4a25-8909-7f850a77caae",
	"CommandData": "2D424ED9260CB9657E0E185B8F59ABDC65EC99A55674439
	1C3BD5F541D030FE89224A6E3FC6D"
	}

4.5.1.3 Secure messaging – Verify mutual authentication response

This API is used to verify the mutual authentication response received from the tag for the mutual authenticate command.

Table 8 API – Verify mutual authentication response

	Detail			
API	sm/verify-ma			
Request Method	POST			
Path Parameters	None			
Request Type	application/json			
Request Parameters	SessionID	String	36	Session ID of the verification session
	MutualAuthResponse	String	18 bytes	Response of the mutual authenticate command, including the status word

Request Sample	{
	"SessionID": "fe606ab6-769d-4a25-8909-7f850a77caae",
	"MutualAuthResponse": "1A C4 E7 08 E5 D6 C6 4F F1 F8 F9 6B 66 0B 3F 67 90 00"
	}			

Response Type	application/json			
Response Parameters	Version	String	5	Version of the API
	AuthResult	String	7	“SUCCESS” or “FAILURE”

Response Sample	{
	"Version": "01.08",
	"AuthResult": "SUCCESS"
	}			

4.5.2 Keys and products API

These APIs are secured with AWS Cognito user authentication. Therefore, a valid token is necessary to invoke these APIs.

4.5.2.1 Authorization service

For the authorization service URL, refer to [8].

This authorization service is an AWS Cognito service that federates user authentication with Infineon's Single-Sign-On (SSO) system. Thus, users who registered for a myInfineon account (refer to Chapter 3.2.3) will be able to login with this account.

Note: This federated Infineon SSO service is only configured in the Infineon provided evaluation environment. This configuration is not available in the CloudFormation templates provided. Therefore, user account registration/sign-up must be implemented based on application-specific requirements.

The Amazon Cognito Hosted UI provides an OAuth 2.0 compliant authorization server. It includes default implementation for the login workflow. To login, the Infineon secured NFC tag IDE application redirects to the Hosted UI, which handles the login workflow and returns the authorization tokens in the resulting URL upon successful authentication. The URL below shows the format of the resulting URL with placeholders for the server generated information.

```
https://localhost:8081/
index.html#access_token=<access_token>&id_token=<id_token>&token_type=Bearer&expires_in=<token_e
xpiry>
```

Note: The order of the URL query parameters can change.

Access token: A JSON web token (JWT) is used to represent the access token. The purpose of the access token is to authorize API operations.

ID token: A JSON web token (JWT) represents the ID token. This contains claims about the identity of the authenticated user, such as name, etc. If required, this identity information can be used within the application.

Token type: The type of token that is issued. Bearer Tokens are the most common type of access token used with OAuth 2.0.

Token expiry: The token expiry time is denoted in seconds. The application can choose to refresh the token before it expires, or it can perform another login to obtain a new token after the token expires.

4.5.2.2 Keys and products service endpoint

For the keys and products service endpoint URL, refer to [7].

4.5.2.3 List keys API

This API will list the keys accessible to the logged in user.

Keys are classified into the following types:

1. **Infineon key (KeyType: 0):** This is the default key created by Infineon and cannot be edited by the user.
2. **User key (KeyType: 1):** This key can be edited by the user.

Table 9 API – List keys

	Detail			
API	keys			
Request Method	GET			
Path Parameters	None			
Request Type	-			
Request Headers	Authorization: <access-token>			
Request Parameters	-			
Request Sample	-			
Response Type	application/json			
Response Parameters	KeyLabel	String	2 bytes	Key label used as reference for the key
	KeyValue	String	16 bytes	Key value
	KeyType	String	1 byte	Infineon Key (0) or User Key (1)

Response Sample	[{
	"KeyLabel" : "A001",
	"KeyValue" : "73737373737373737373737373737373",
	"KeyType" : "1"
	},{
	"KeyLabel" : "A002",
	"KeyValue" : "73737373737373737373737373737373",
	"KeyType" : "1"
	
	},{
	"KeyLabel" : "FF01",
	"KeyValue" : "B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF",
	"KeyType" : "0"
	}]			

4.5.2.4 Update key API

This API is used to update the user key. This API does not allow users to update Infineon Key (KeyType: 0) or the keys of other users.

Table 10 API – Update key

	Detail
API	keys

(table continues...)

Table 10 (continued) API – Update key

	Detail			
Request Method	POST			
Path Parameters	None			
Request Headers	Authorization: <access-token>			
Request Type	application/json			
Request Parameters	KeyLabel	String	2 bytes	Key label for the key which has to be updated
	KeyValue	String	16 bytes	New key value to be updated

Request Sample	<pre>{
	"KeyLabel": "A001",
	"KeyValue": "83838383838383838383838383838383"
	}</pre>			

Response Type	-			
Response Code	200 (Success)			

4.5.2.5 List products API

This API will list the products accessible to the logged in user.

Product webpage URLs are classified into the following types:

1. **Infineon product (ProductType: 0):** The product webpage URL which is created default by Infineon and cannot be edited by the user.
2. **User product (ProductType: 1):** This product webpage URL can be edited by the user.

Table 11 API – List products

	Detail
API	products
Request Method	GET
Path Parameters	None
Request Type	-
Request Headers	Authorization: <access-token>
Request Parameters	-

(table continues...)

Table 11 (continued) API – List products

	Detail			
Request Sample	-			
Response Type	application/json			
Response Parameters	ProductWebpage	String	Up to 1024 bytes	Product webpage URL
	ProductCode	String	2 bytes	Product code
	ProductType	String	1 byte	Infineon product URL (0) or user product URL (1)

Response Sample	<pre>[
	  {
	  "ProductWebpage": "https://www.infineon.com/cms/en/discoveries/near-field-communication/",
	  "ProductCode": "FE01",
	  "ProductType": "0"
	  },
	  {
	  "ProductWebpage": "https://www.infineon.com",
	  "ProductCode": "A006",
	  "ProductType": "1"
	  },
	  {
	  "ProductWebpage": "https://www.infineon.com/cms/en/discoveries/near-field-communication/",
	  "ProductCode": "A007",
	  "ProductType": "1"
	  }
	]</pre>			

4.5.2.6 Update product API

This API is used to update the user product webpage. This API does not allow users to update Infineon product (ProductType:0) or the product webpage of other users.

Table 12 API – Update product

	Detail
API	products
Request Method	POST
Path Parameters	None

(table continues...)

Table 12 (continued) API – Update product

	Detail			
Request Headers	Authorization: <access-token>			
Request Type	application/json			
Request Parameters	ProductCode	String	2 bytes	Product code for which the webpage must be updated
	ProductWebpage	String	Up to 1024 bytes	New product webpage

Request Sample	<pre>{
	"ProductCode": "A006",
	"ProductWebpage": "https://www.infineon.com/cms/en/discoveries/near-field-communication/"
	}</pre>			

Response Type	-			
Response Code	200 (Success)			

Note: The API does not validate the webpage URL format. The client shall perform basic validations on the URL format before invoking the update API.

4.5.2.7 Status and error codes

The following standard HTTP status codes are commonly responded by the cloud server.

Table 13 Standard HTTP status codes

HTTP status code	Detail
200	OK Standard response for successful HTTP requests
400	Bad request The server cannot or will not process the request due to an apparent client error (for example, malformed request syntax, size too large, invalid request message framing, or deceptive request routing)
401	Unauthorized The user does not have valid authentication credentials for the target resource
404	Not found The requested resource could not be found in the server
405	Method not allowed A request method is not supported for the requested resource. E.g. GET, POST
500	Internal server error Operation failed while processing the request in the server

For more information on HTTP status codes, refer to [14].

The following are the error codes responded by the cloud server in case of internal errors.

Table 14 Internal error codes

Internal error code	Detail
1002	Incorrect/invalid parameters in the request
1003	Session failure
1004	Failed to build mutual authentication command data

4.6 Server-side brand verification library

The brand verification library is a collection of Java Archive (JAR) files which is hosted as AWS Lambda service. This service can be accessed via HTTPS API endpoints created with AWS API gateway. This library is linked to the keystore and sessions database.

Figure 36 shows the data flow happening with this library during brand verification.

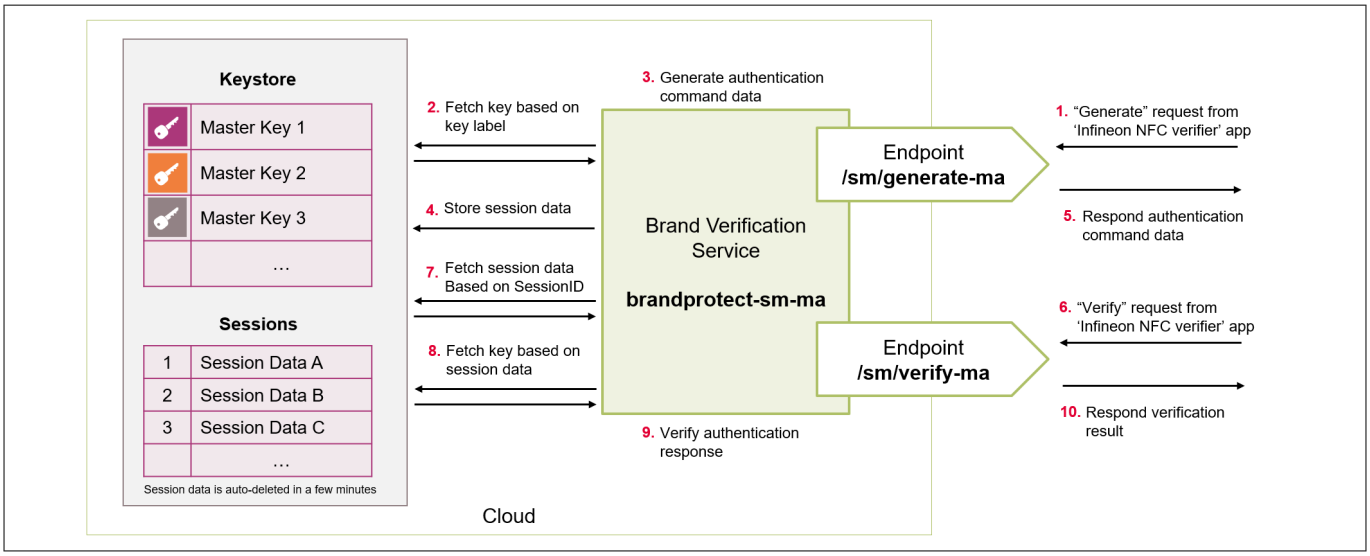

Figure 36 Brand verification process in the cloud

The library performs the following operations:

- Generate mutual authentication command data
- Verify mutual authentication command response

This library identifies the request based on the API invocation path. For more information on command parameters and response details, refer to Chapter 4.5.

4.7 Server-side database

This starter kit uses AWS DynamoDB for database operations. The keys, product detail and sessions are stored in the database.

Keystore

Keystore table stores the master keys in the cloud. These keys are referenced by a 2-byte key label, which is generated in the AWS cloud.

The starter kit uses additional fields such as KeyType and UserID. The KeyType is used to identify whether the key is a default key (KeyType: 0) or a user editable key (KeyType: 1). When a new user registers in the starter kit, a set of keys are created for them automatically.

Table 15 Keystore (stores master keys in the cloud)

S.no	Attribute	Type	Sample values
1	KeyLabel	String	A001
	KeyAlgoID	String	AES128
	KeyType	String	1
	KeyValue	String	73737373737373737373737373737373
	UserID	String	user1001@example.com

S - String type

Sessions

During the brand verification process, the state of the session must be maintained between the "generate" and "verify" requests. Session table stores this authentication session information in the cloud.

Session information (including SessionID) is deleted after a few minutes. With AWS DynamoDB, this can be configured by setting the SessionValidity as TTL field. The SessionValidity field denotes the session expiry date-time in seconds (Unix epoch time representation).

Table 16 Sessions (stores authentication session information in the cloud)

S.no	Attribute	Type	Sample values
1	SessionID	String	22e40ccb-c53d-405a-855b-bb15ba010e27
	SessionValidity	Number	1638584236 (in seconds)
	SessionData	String	ACED000573720038636F6D2E69…(Trimmed)
	KeyLabel	String	A001

Product

Product table stores the URL of the product webpage. This product webpage can be opened from the Infineon NFC verifier application or from the landing webpage when the tag is scanned. This URL is referenced by a 2-byte ProductCode, which is generated in the AWS cloud.

The starter kit uses additional fields such as ProductType and UserID. The ProductType is used to identify whether the product is a default product (ProductType: 0) or a user editable product (ProductType: 1). When a new user registers in the starter kit, a set of products are created for them automatically.

Table 17 Product (stores the URL of the product webpage)

S.no	Attribute	Type	Sample values
1	ProductCode	String	A006
	ProductType	String	1
	ProductWebpage	String	https://www.infineon.com/cms/en/product/security-smart-card-solutions/nfc-solutions/
	UserID	String	user1001@example.com

Note: Securing the keys and session information in the cloud are not the focus of this starter kit, and therefore stored as plain text in cloud database. It is recommended to secure them during production implementation.

***Attention*: Infineon highly recommends the user to take appropriate steps to secure the master keys and cloud services whereby loss includes theft, damage or any other event that could impact/ compromise the brand protection system, can be prevented.**

Note: In addition to these tables, the starter kit employs tables (such as counter, verifications, etc.) for analysis purposes.

4.8 Setting up the cloud environment

An AWS CloudFormation template is a JSON file that describes AWS infrastructure. This template can be used to create AWS infrastructure using the AWS CloudFormation service.

CloudFormation has the following advantages:

1. **Simplify infrastructure management:** The entire infrastructure is deployed by code templates.
2. **Quick replication of infrastructure:** AWS resources are hosted in a data center region preferred by the user. However, if the same infrastructure needs to be replicated in another AWS region or into a different AWS account, CloudFormation templates facilitate these processes.

4.8.1 Installing the cloud templates

The starter kit provides the CloudFormation templates in JSON format that can be used to deploy services into an AWS account.

Figure 37 illustrates the content of the CloudFormation templates.

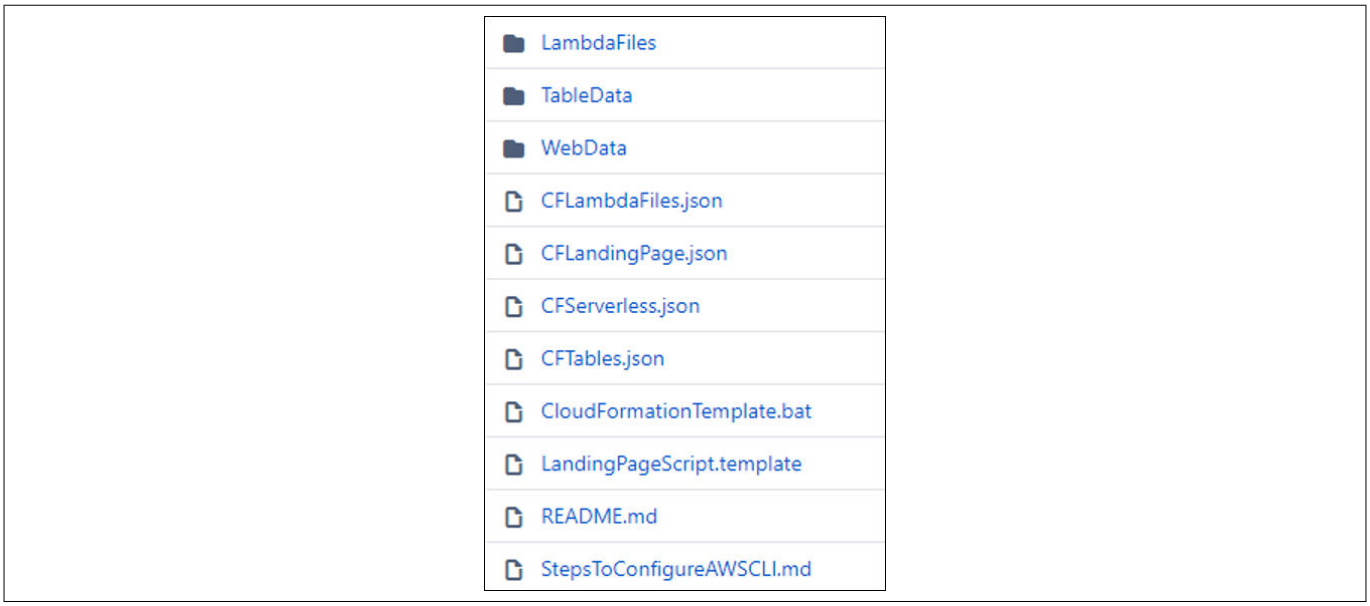

Figure 37 Content of the CloudFormation templates

The steps to deploy the CloudFormation templates are described in the README.md file [9].

The contents of the CloudFormation templates are as follows:

- **CloudFormationTemplate.bat:** This batch file executes the template files in the AWS account and creates the AWS infrastructure for brand protection.

- **LandingPageScript.template:** CloudFormationTemplate.bat uses this script to edit and upload the landing webpage.
- **StepsToConfigureAWSCLI.md:** This file contains the instructions to set-up the AWS CLI.
- **CFLandingPage.json:** This file creates an S3 bucket with the name "nfcbpsk-s3-bucket", which is used to store the source files of the landing webpage.
- **CFLambdaFiles.json:** It creates the "temp-nfcbpsk-s3-bucket-jar" for temporary storage used for the deployment of JAR files. This temporary bucket will be deleted after the deployment is completed.
- **CFServerless.json:** This template creates the brand verification services, products and keys management services, and provides APIs for these services.
- **CFTables.json:** This file generates the table structures required for brand protection.
- **LambdaFiles:** It contains the AWS Lambda services in the form of JAR libraries.
- **TableData folder:** It includes a predefined data set that is used to initialize the database tables with default values.
- **WebData folder:** It contains the source files of the landing webpage such as HTML, CSS, JS, etc.

4.9 Mobile application development

The Infineon NFC verifier mobile applications perform the basic functionality such as brand verification and reading the product information from the Secured NFC tag. This application can be used as a reference for building brand verifier applications.

This starter kit uses inbuilt libraries and API provided by the smartphone platform for APDU communication over NFC. The NFC mode used in this application is ISO/IEC 7816-based, and it exchanges APDUs with the Secured NFC tag.

The concept of deep-linking (in Android)/associated-domains (in iOS) is used to auto-launch the mobile application when the Secured NFC tag is scanned. The URL configured in the URI record of the Secured NFC tag is associated with the mobile application. The starter kit landing webpage URL is linked to the mobile application in the starter kit.

4.9.1 iOS application source

To open the Infineon NFC verifier iOS application in Xcode, follow the steps below:

1. Download the "NFCVerifier_iOS" project from the GitHub [1] repository.
2. Extract the zip file into the working folder.
3. Open the "NfcVerifier_iOS.xcodeproj" file in Xcode.
4. Choose the target device, click "Run" button to build, and run the application.

4.9.2 Android application source

To open the Infineon NFC verifier Android application in Android Studio, follow the steps below:

1. Download the "NFCVerifier_Android" project from the GitHub [1] repository.
2. Extract the zip file into the working folder.
3. Launch Android Studio.
4. Select "Open Existing Android Studio Project".
5. Select the extracted folder and click "OK"
6. Choose the target device, click "Run" button to build, and run the application.

References

Infineon

[1] NFC Brand Protection: *NFC 2Go Starter Kit For Brand Protection - GitHub* - https://github.com/Infineon/NFC-verifier

[2] NFC Brand Protection: *Infineon NFC verifier (iOS) - Available in Apple App Store* - https://apps.apple.com/us/developer/infineon-technologies-ag/id469396533/

[3] NFC Brand Protection: *Infineon NFC verifier (Android) - Available in Google Play Store* - https://play.google.com/store/apps/developer?id=Infineon+Technologies+AG

[4] NFC Brand Protection: *Infineon secured NFC tag IDE (PC tool) - Infineon Development Center* - https://softwaretools.infineon.com/

[5] NFC Brand Protection: *NFC 2Go Starter Kit For Brand Protection - Landing webpage* - https://nfc2goskbp.infineon.com/index.html

[6] NFC Brand Protection: *NFC 2Go Starter Kit For Brand Protection - Brand verification service endpoint* - https://nfc2goskbp.infineon.com/verify

[7] NFC Brand Protection: *NFC 2Go Starter Kit For Brand Protection - Keys and products service endpoint* - https://nfc2goskbp.infineon.com/manage/

[8] NFC Brand Protection: *NFC 2Go Starter Kit For Brand Protection - Authorization service* - https://nfcbpk.auth.eu-central-1.amazoncognito.com/login?client_id=5v7jdsjl2t2vqh7llhtioqhnf4&response_type=token&scope=email+openid+profile&redirect_uri=http://localhost:8081/index.html

[9] NFC Brand Protection: *NFC 2Go Starter Kit For Brand Protection - CloudFormation template* - https://github.com/Infineon/NFC-verifier/CloudService/AWSTemplates/README.md

NFC Forum

[10] NFC Forum™: *NFC Forum™ Type 4 Tag Technical Specification Version 1.0*

[11] NFC Forum™: *NFC Forum™ Type 4 Tag Operational Specification T4TOP 2.0*

[12] NFC Forum™: *NFC Forum™ NFC Data Exchange Format (NDEF)Technical Specification NDEF 1.0*

[13] NFC Forum™: *NFC Forum™ URI Record Type Definition Technical Specification RTD-URI 1.0*

Others

[14] Internet Assigned Numbers Authority: *Hypertext Transfer Protocol (HTTP) Status Code Registry* - https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml

Glossary

AES

Advanced Encryption Standard (AES)
The standard for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. The algorithm described by AES is a symmetric-key algorithm (i.e. the same key is used for both encryption and decryption).

APDU

application protocol data unit (APDU)
The communication unit between a smart card reader and a smart card.

API

application programming interface (API)

ATR

answer to reset (ATR)
A message conforming to ISO/IEC 7816 sent by the controller following a reset. It contains information on communication parameters, type and state of the chip.

AWS

Amazon Web Services (AWS)

AWS is a cloud computing platform provided by Amazon that includes offerings such as infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS).

EF

elementary file (EF)
A file system component containing (user) data.

FCI

file control information (FCI)

FID

file identifier (FID)
Used to reference an elementary file.

HTTPS

hypertext transport protocol secure (HTTPS)

HTTPS is an internet communication protocol that is used for secure communication. Data sent through HTTPS is secured using transport layer security (TLS).

IDE

integrated development environment (IDE)

A software application that combines multiple tools used for software development into a single environment.

iOS

iPhone OS (iOS)

iOS is a mobile operating system created and developed by Apple Inc. exclusively for its hardware.

JAR

Java archive (JAR)

JAR is a package file format typically used to aggregate many Java class files and associated metadata and resources (text, images, and so on) into one file for distribution.

JRE

Java runtime environment (JRE)

JRE, also known as Java runtime, is the part of the Java development kit (JDK) that contains and orchestrates the set of tools and minimum requirements for executing a Java application. The JRE specifically contains a Java class loader, which is responsible for loading classes and connecting them to the core Java class libraries.

JSON

Java script object notation (JSON)

JSON is a lightweight format for storing and transporting data. It is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays.

NDEF

NFC data exchange format (NDEF)

NDEF is a standardized data format specification by the NFC Forum which is used to describe how a set of information are to be encoded onto an NFC tag or to be exchanged between two active NFC devices.

NFC

near field communication (NFC)

NLEN

NDEF length (NLEN)

The NDEF length is a field in the NDEF message that indicates the size of the NDEF message.

PC/SC

personal computer/smart card (PC/SC)

PC/SC is a standard to interface computers with smartcards, available on most operating systems, including Windows and Linux.

PKI

public key infrastructure (PKI)

RFU

reserved for future use (RFU)

SFID

short file identifier (SFID)

SR

short record (SR)

The short record flag in the NDEF message is a 1-bit indicator that, when set, denotes that the payload length field is a single octet.

TNF

type name formats (TNF)

The record type string field of an NDEF record contains the name of the record type. Record type names are used by NDEF applications to identify the semantics and structure of the record content.

URI

uniform resource identifier (URI)

URI is a string of characters that uniquely identify a name or a resource on a network such as the internet.

URL

uniform resource locator (URL)

A URL is a unique identifier that is used to locate a resource on the internet. It is also referred to as a web address.

vCard

virtual card or virtual contact file (vCard)

vCard is a file format standard for electronic business cards. vCard format is the widely used format for contact data interchange, usually contains information such as contact's name, address, phone number, email, birthday, photographs, and so on.

Revision history

Reference	Description
Revision 1.0, 2022-06-08	
All	Initial release

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-06-08

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email:
CSSCustomerService@infineon.com

Document reference
IFX-ghr1636346375272

Important notice

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.