Personal Code

Declaración de Seguridad

v1.3

02/09/2021

Creado por

Control de versiones

Versión	Descripción	Fecha
0.1	Versión inicial	12-05-2020
0.2	Pequeños cambios relacionados con el scope del SDK de validación	15-05-2020
0.3	Refinamiento en secciones *2.1.Descripción funcional del TOE 3.1.Descripción del entorno de ejecución*	10-06-2020
0.4	Inclusión de pequeños cambios en la descripción de las partes del TOE e inclusión de los manuales del producto	19-06-2020
0.5	Aplicación de cambios a raíz de los comentarios realizados de CPSTIC.	25-06-2020
0.6	Actualización de la descripción del proceso de generación de claves y de la sección *4.1.Hipótesis sobre el entorno de ejecución.* Se incluyen, además, amenazas y requisitos relacionados con biometría. Se incluye una sección para reflejar el ciclo de vida de las claves del TOE.	30-11-2020
0.7	Aplicación de cambios a raíz de los comentarios realizados por CPSTIC.	15-02-2021
0.8	Se excluye la funcionalidad biométrica del producto como parte del problema de seguridad.	17-02-2021
1.0	Se han implementado los cambios necesarios para solventar las no-conformidades emitidas por el laboratorio.	20-04-2021
1.1	Se ha actualizado la versión del producto y de los manuales.	30-04-2021
1.2	Se ha actualizado la versión del producto, se ha incluido la biblioteca Chilkat como parte del TOE y se ha incluido el uso de *ConfuserEx* como herramienta de ofuscación.	17-05-2021
1.3	Se ha actualizado el listado de librerías de terceros del requisito REQ.17, se ha eliminado la	02-09-2021

	mención a la utilidad Digital Persona y se han actualizado las versiones de los manuales.	

Tabla de contenidos

1 Introducción

El presente documento es la Declaración de Seguridad para el TOE Personal Code. En el mismo se abordarán los siguientes puntos:

1. Información acerca de este documento.
2. Descripción de la funcionalidad de la solución y casos de uso, que permite a las organizaciones o servicios públicos hacer frente a las amenazas y riesgos que se describen en el documento.
3. Descripción del entorno de ejecución.
4. Análisis de amenazas y riesgos del producto.
5. Requisitos fundamentales de seguridad que esta solución cubre y cómo los cubre para implementar cada función de seguridad.
6. Acrónimos, Glosario de términos y Documentos referenciados.

1.1 Información del patrocinador, TOE y evaluación

Datos del patrocinador	HuBOX
Datos del desarrollador	HuBOX
Datos de contacto del desarrollador	Séneca 134, Piso3, Oficina B Polanco Chapultepec, Del. Miguel Hidalgo, Ciudad de México fperezm@hubox.com
Nombre del TOE	Personal Code
Versión del TOE	2020.3.2
Tipo de evaluación	LINCE
Manuales de uso del producto y su versión	Véase sección *2.2.Identificación de las guías de uso, instalación y configuración seguras del TOE*.
Taxonomía del producto según CCN-STIC-140	NA

1.2 Metodología y criterios de evaluación

CCN-STIC-2001	Definición de la Certificación Nacional Esencial de Seguridad. (Enero 2020)
CCN-STIC-2002	Metodología de Evaluación para la Certificación Nacional Esencial de Seguridad. (Enero 2020)
CCN-STIC-2003	Plantilla para la Declaración de Seguridad de la Certificación Nacional Esencial de Seguridad. (Enero 2020)

2 Descripción del TOE

2.1 Descripción funcional del TOE

El producto objeto de la evaluación o TOE (*Target Of Evaluation*) consiste en un SDK de generación de claves criptográficas para la inclusión de información confidencial que identifique de forma unívoca a un usuario determinado que utilice una aplicación final. Además, el TOE a su vez, está formado de una SDK de validación que permite realizar, en términos generales, el procedimiento inverso, validando la autenticidad de la información confidencial generada por el SDK de generación y extrayendo la información propiamente dicha. La funcionalidad del TOE consiste en la generación de dos códigos QR para la inserción de los datos personales de un usuario determinado y su posterior validación de dicha información confidencial. Estos datos personales son aquellos que le permiten identificarlo de forma unívoca ante cualquier entidad.

De acuerdo con esta descripción, el TOE consiste en una solución compuesta por dos fases principales: generación y validación. Cada una de estas fases tiene su importancia dentro de la solución final ofrecida, pero, la generación de claves y su correspondiente implementación durante la generación de un código QR bidimensional, consisten en la fase con mayor relevancia debido a que es aquí donde se proporcionan las medidas de seguridad suficientes para proteger así la confidencialidad de la información que se pretende cifrar. Es necesario hacer mención a que la generación de claves se realizará utilizando la biblioteca Chilkat, mediante el uso de las distintas funciones que incluye. No obstante, el producto dispone de las bibliotecas necesarias para llevar a cabo la comunicación con un HSM físico y hacer uso de sus funcionalidades con la finalidad de adaptarse a las necesidades de su entorno de operación. Sin embargo, estas bibliotecas y su correspondiente comunicación con un HSM no se consideran parte del TOE y por tanto están fuera del ámbito de esta declaración de seguridad.

El objetivo de cada una de las partes de la solución final y por lo tanto del propio TOE es solucionar los siguientes problemas asociados a este tipo de implantaciones:

- Falsificación/Alteración.
- Suplantación de Identidad.
- Validación del documento.
- Uso en Dispositivos Móviles.
- Vida útil del documento.
- Reposición/Reimpresión.
- Protección de datos.

Para lograr estos objetivos, el TOE dispone de un encapsulamiento del SDK en un archivo compilado de extensión *.dll* y sufre una ofuscación del código manual y automático mediante el uso de la herramienta *ConfuserEx,* la cual permite transformar el código de forma que sea extremadamente difícil llevar a cabo un proceso de ingeniería inversa. Además, el TOE dispone de los siguientes niveles de protección:

- ***Optimización*.** En este nivel de protección se pretende disponer de un código fuente de calidad para evitar ataques asociados a un código fuente no optimizado. Para ello se llevan a cabo los siguientes pasos:
    - Eliminación de código redundante, código de registro y metadatos, recursos no utilizados y bibliotecas nativas.

    - Optimización del código y de los recursos.
- ***Hardening*.** En este nivel de protección se pretende proporcionar al TOE de una capa adicional de protección consiguiendo lo siguiente:
    - Ofuscación de instrucciones aritméticas, control de flujo, código nativo y nombres de bibliotecas, recursos y métodos de llamada del SDK.
    - Encriptación de clases, cadenas, *assets*, archivos de recursos y bibliotecas nativas.
    - Empaquetado del código con *bytecode* combinado.
- ***Autoprotección en tiempo de ejecución*.** Este nivel de protección está destinado a evitar cualquier tipo de ataque en tiempo de ejecución, detectando cualquier anomalía que pueda afectar a la seguridad y comportamiento normal del TOE:
    - Detección de herramientas de depuración, protección contra herramientas de desensamblado (*ILdasm*), generación de proxy dinámico (interno y externo), protección contra *reflection* y *tampering*.

Desde el punto de vista de la funcionalidad del producto, es necesario tener en cuenta, en términos generales, el procedimiento de generación de claves y de cifrado de información. En el siguiente diagrama se puede observar las distintas fases:

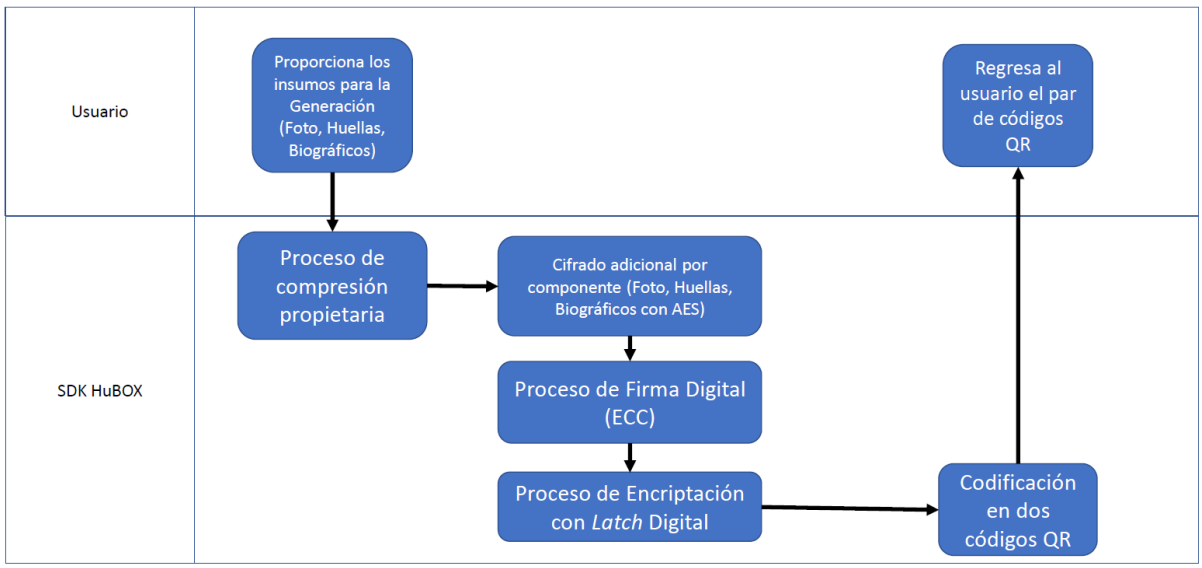

Ilustración 1. Fases del proceso de cifrado de información.

Cuya descripción general se procede a realizar a continuación:

- ***Generación de insumos*.** Esta fase independiente del TOE, consiste en la recopilación de la información relacionada con los datos del usuario.
- ***Proceso de compresión.*** El TOE lleva a cabo un proceso de formateo o preparación de la información seguido de una compresión de la información resultante.
- ***Cifrado adicional independiente*.** En esta fase, se lleva a cabo un cifrado independiente adicional por cada tipo de insumo que se esté cifrando, utilizando el algoritmo AES para ello. Este cifrado proporciona el privilegio de acceso exclusivo a la información contenida y una protección en múltiples niveles.
- ***Firma digital*.** En esta fase, se lleva a cabo el firmado de los bytes cifrados utilizando el paradigma de criptografía de curva elíptica, proporcionando de este modo autenticidad y evitando la falsificación y no repudio.

- ***Cifrado por capas.*** Después de llevar a cabo el firmado de la información, se procede a cifrar los bytes generados en dos etapas: la primera se realiza sobre los primeros bytes de la cadena; en la segunda etapa se lleva a cabo un cifrado de los últimos bytes de la cadena. En ambas etapas el algoritmo utilizado es RSA. Al final de este proceso de dos etapas se puede deducir la existencia de un grupo de bytes que han sido cifrados dos veces, produciendo así un solapamiento de capas de cifrado o *Latch Digital*.
- ***Posicionamiento de la información cifrada.*** Posteriormente, los bytes cifrados son posicionados y reordenados con la finalidad de separar la información que se incluirán en los códigos QR. Adicionalmente, se añade la información necesaria para su validación una vez organizados los bytes.
- ***Construcción de los códigos QR.*** Finalmente, se codifica la información en una imagen que contiene los códigos bidimensionales tipo QR estándar, los cuales, a su vez, contienen la información cifrada, comprimida y posicionada de los datos biográficos y biométricos del solicitante.

Por otra parte, una vez que se han generado los códigos QR con la información cifrada, el TOE puede llevar a cabo el proceso inverso en el cual realiza una verificación de la información incluida en dichos códigos. El proceso de validación, por tanto, consiste en los siguientes pasos:

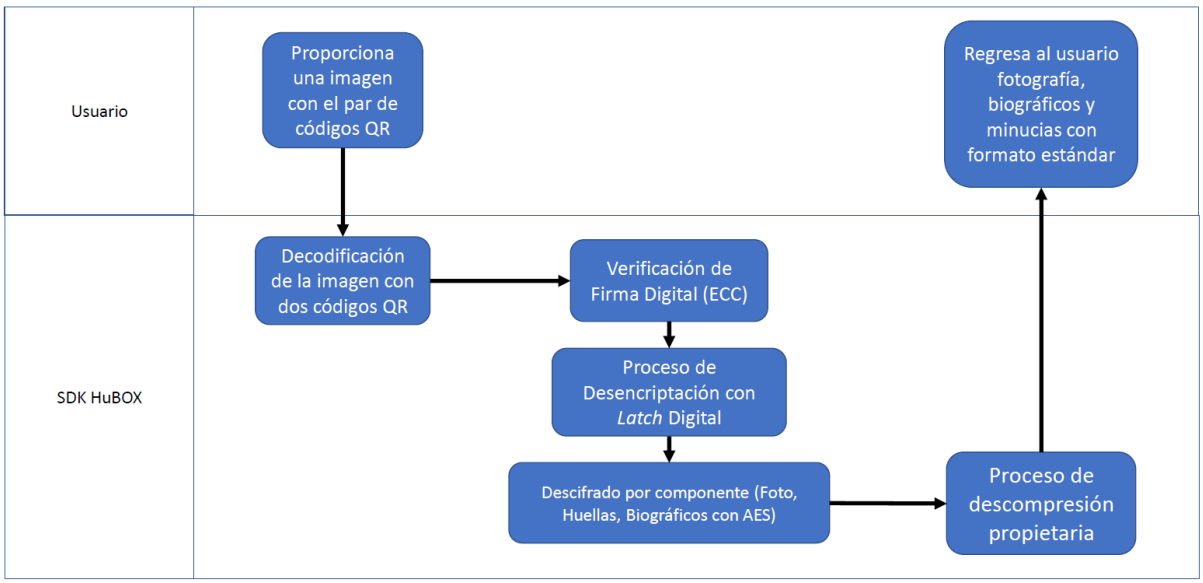

Ilustración 2. Fases del proceso de validación de información.

En este caso, las distintas fases de validación de los códigos QR se describen a continuación:

- **Verificación de los códigos QR.** A través de la aplicación de escritorio que hace uso del SDK de validación, se lleva a cabo un escaneo de los códigos QR. En este punto, se decodifica la información en una imagen que contiene los códigos bidimensionales tipo QR estándar, los cuales, a su vez, contienen la información cifrada, comprimida y posicionada de los datos biográficos y biométricos del solicitante.
- **Posicionamiento de la información cifrada.** Posteriormente, los bytes descifrados son posicionados y reordenados con la finalidad de conjuntar la información contenida en los códigos QR a descifrar.
- **Segunda capa de cifrado final.** Debido a que, durante la generación de los códigos se realiza la segunda capa de cifrado desde el final de la cadena, se desencripta desde los bits menos

significativos o LSB las llaves públicas correspondientes a la segunda capa de cifrado de generación.

- **Latch Digital.** Es el resultado de la aplicación de ambas capas de cifrado durante el proceso de generación, es decir, la cadena de bytes que termina con la superposición de N bytes, y que conlleva, por lo tanto, un proceso de desencriptación ordenado de ambas partes de la información cifrada con su llave pública correspondiente.
- **Primera capa de cifrado final.** Una vez desencriptada la información binaria de los bits menos significativos o LSB, se procede a descifrar los bytes restantes (propiamente a los bytes más significativos o MSB) mediante su llave pública correspondiente.
- **Firma digital de la información.** En este proceso se corrobora el firmado digital de la información mediante el paradigma de criptografía de curva elíptica, aplicando la llave pública correspondiente.
- **Descifrado independiente por elemento.** Posteriormente se realiza el descifrado (mediante el algoritmo AES) independiente para cada parámetro al cual se le aplicó el proceso de cifrado adicional (Fotografía, datos biográficos, huellas dactilares).
- **Descompresión de la información.** La información recuperada de los procedimientos anteriores, se somete a un proceso de descompresión de datos propietaria de HuBOX, tratando la información de forma separada de acuerdo con lo previamente codificado por la SDK de generación.

Una vez finalizado el proceso descrito, se procesa la información de forma que se recupera la información binaria de cada parámetro y, por lo tanto, proporcionando al usuario final de garantías de la no suplantación de la identidad.

Estas distintas fases hacen posible llevar a cabo la función principal del TOE con garantías de seguridad para la protección de los datos de identificación personales proporcionados en primera instancia. En última instancia, mediante la generación de QRs se ofrece:

- **Confidencialidad.** Protegiendo la información de carácter personal.
- **Autenticidad.** Mediante esta técnica se evita la falsificación de la información, identificando unívocamente los datos personales del usuario.
- **Integridad.** Proporciona un encapsulado seguro de los datos, protegiéndolo frente a modificaciones no deseadas.
- **No repudio.** Asegurando de su validez y procedencia tanto al emisor como el receptor del documento, así como al portador del mismo.

2.2 Identificación de las guías de uso, instalación y configuración seguras del TOE

En la siguiente tabla, se listan los documentos y guía de usuario necesarios para poder llevar a cabo la configuración del TOE de forma adecuada.

Tipo de documento	Nombre del documento	Versión del documento
Manual de instalación	Manual de configuración de ambiente	v2.1
Manual de usuario	Manual de usuario SDK Generación	v2.0
Manual de usuario	Manual de usuario SDK Validación	v2.1
Manual de usuario	Instalación Neurotechnology Trial	V2.1

2.3 Descripción del modo de uso del TOE

Para obtener las funciones de seguridad descritas en la sección *5. Funciones de seguridad* de este documento es necesario seguir los pasos incluidos en la guía de instalación del producto. Con carácter adicional, de acuerdo con la hipótesis *H.Plataforma cifrada* incluida en la sección *4.1.Hipótesis sobre el entorno de ejecución*, se precisa el uso de herramientas de cifrado de disco como *Bitlocker* en las máquinas donde se ejecuten las partes que conforman el TOE.

3 Entorno de ejecución

3.1 Descripción del entorno de ejecución

Tal y como se ha mencionado anteriormente, el TOE forma parte de una solución compleja en la que se proporciona un servicio de validación de documentación de identificación asociada a un usuario. Cabe indicar que el producto hace uso de la tecnología *.NET Framework 4.8* (tanto para el SDK de generación como para el SDK de validación) para la puesta en marcha de sus funcionalidades. Las distintas fases involucradas en el entorno de ejecución del TOE se muestran en la siguiente ilustración:

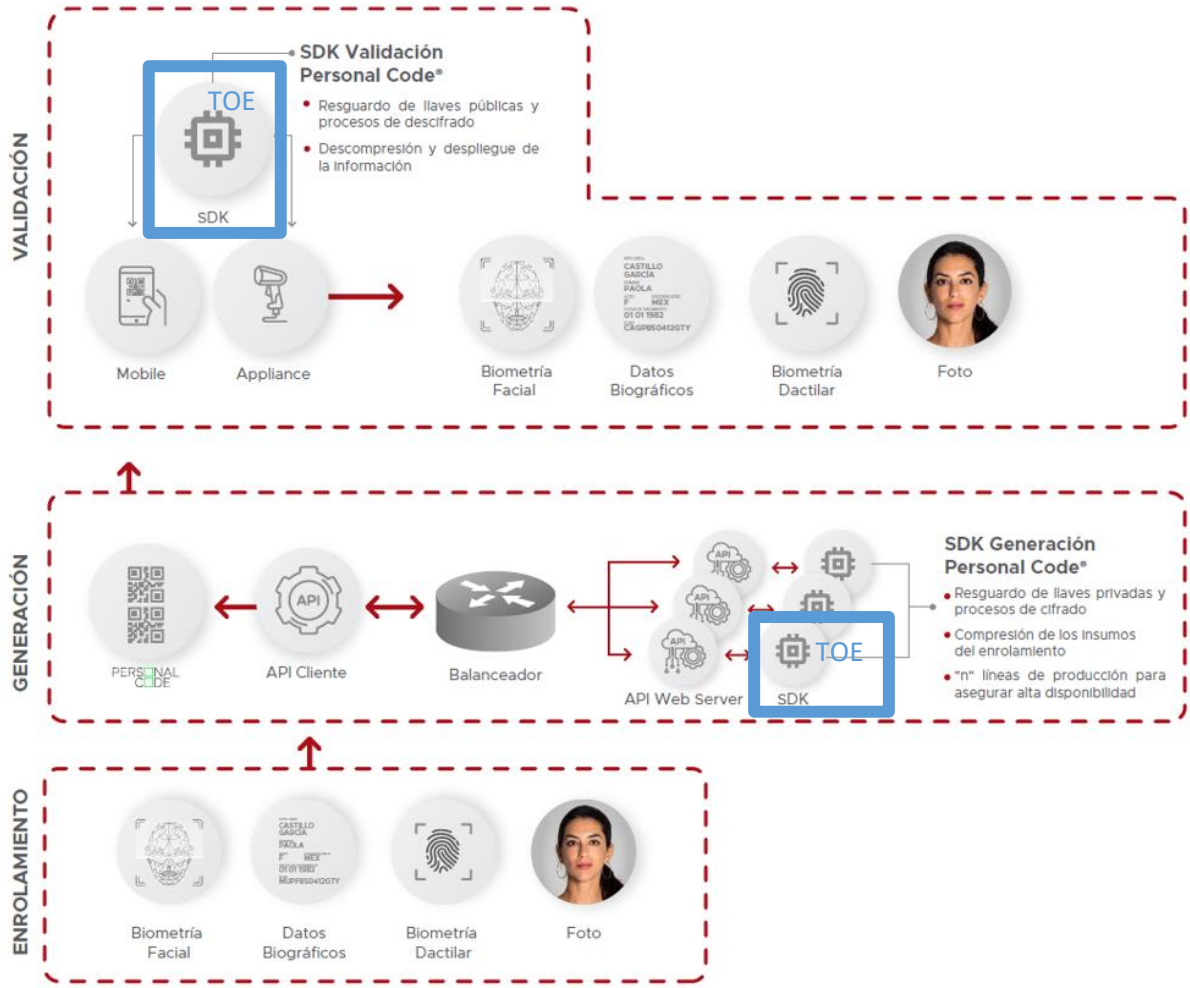

Ilustración 3. Fases del entorno de ejecución

Cabe señalar, de acuerdo con la imagen anterior, la distribución de dos componentes distintos, separados físicamente, que conforman el propio TOE. La finalidad de cada una de las fases se describe a continuación:

- **Enrolamiento.** Esta parte del proceso consiste en recabar información relacionada con los datos de usuario que se pretende utilizar para identificar al propio usuario. Los datos se recopilan a través del API cliente el cual hace el envío de información al SDK de certificación. En este punto, se hace la extracción de las minucias biométricas del rostro y de las huellas

dactilares y se envían al SDK de generación. Tal y como puede observarse, esta fase no forma parte del TOE objeto de la presente declaración de seguridad.

- **Generación.** Esta es la fase en la que interviene el funcionamiento del TOE. En términos generales, consiste en cifrar y alojar la información en códigos QR para que puedan ser validados por una aplicación de escritorio o aplicación móvil. Esta información se almacena en una base de datos interna la cual se encuentra alojada en una red interna protegida para dotar, a la información, de medidas de protección adicionales al cifrado implementado por el propio TOE. De esta forma se puede evitar la fuga y corrupción de los propios datos procesados. Hay que destacar la posibilidad de uso de un HSM por parte del TOE. Este HSM tiene un carácter opcional y, por lo tanto, no está contemplado en el ámbito de la evaluación y dependerá del entorno de operación donde el TOE se encuentre en ejecución.
- **Validación.** Esta parte del proceso consiste en, mediante el uso de una aplicación móvil o de un dispositivo que escanee los códigos QR (*appliance*) donde se encuentra en ejecución la segunda parte del TOE (SDK de validación), descifrando la información contenida en los códigos QR generados en el paso anterior de manera que se pueda comprobar, de forma precisa, los datos utilizados para la identificación unívoca de un usuario determinado. Hay que tener en cuenta que, solamente será considerada como parte del TOE la SDK de validación destinada a sistemas operativos de escritorio Windows. El SDK de validación se encarga de realiza una comparación biométrica de la imagen tomada y los datos extraídos del código QR.

Para llevar a cabo el proceso de enrolamiento y el de validación de los datos biométricos, se hace uso de la utilidad *Neurotechnology.* En el caso del enrolamiento, esta utilidad estará incorporada en el componente *API Cliente* del entorno de operación mostrado en *Ilustración 3. Fases del entorno de ejecución.* Tal y como se ha mencionado en la fase de enrolamiento, esta utilidad forma parte del entorno operacional del producto, pero no forma parte del TOE objeto de la presente declaración de seguridad.

Antes de comenzar con la descripción del proceso de puesta en marcha del entorno de operación del TOE, es necesario mencionar la descripción del proceso de generación de claves. Esta generación de claves se llevará a cabo en un equipo protegido y cifrado en la sede de HuBOX que tendrá instalado el programa de generación de llaves HBX que será el que haga uso, a su vez, de la biblioteca *Chilkat*. Por otra parte, se incluye la existencia de un Hardware ID, el cual consiste en un identificador único que hace posible que el TOE reconozca la máquina en la que se encuentra en ejecución, restringiendo su ejecución a dicha máquina. Para el correcto funcionamiento del SDK de generación, es necesario tener en cuenta los siguientes pasos:

- En primer lugar, se lleva a cabo la **generación del juego de claves** de procesos criptográficos que utilizará el TOE para la generación de códigos QR con información cifrada, en el caso del SDK de generación, y de su correspondiente validación, por parte del SDK de validación. De cada par de claves generado (para criptografía asimétrica), se destinará la clave privada para el SDK de generación y la correspondiente clave pública para el SDK de validación (excepto para la clave privada del algoritmo simétrico, que será la misma en ambas partes). Para el caso del SDK de generación, estas claves se almacenan, junto con su tiempo de vigencia asociado y el Hardware ID en un fichero. El tiempo de vigencia de las claves está establecido en un año y, una vez transcurrido dicho tiempo, será necesario el borrado de las claves y una nueva generación de las mismas en ambos SDK.

- A continuación, se inicia el proceso de **generación de claves de envoltura**. Las claves generadas con los algoritmos RSA y AES serán las que cifrarán el fichero de claves generado en el paso anterior. Para el caso de RSA, se hará uso de la clave privada para el cifrado.
- Una vez que se han generado las claves de envoltura y se ha cifrado el fichero de claves, se incluirá la correspondiente clave pública RSA y la clave AES en el código fuente del SDK de generación. Esto permitirá al SDK de generación descifrar el fichero de claves para llevar a cabo la generación de un código QR con la información cifrada.
- En este punto, se llevará a cabo la compilación habitual de un SDK con el entorno *.NET*.
- El resultado generado se ofuscará con la herramienta *ConfuserEx*, generando a su vez, un fichero *.dll* con la información ofuscada y protegida adecuadamente. El resultado de esta operación será el que proporcione la funcionalidad de generación de QRs de una forma segura de acuerdo a lo reflejado por la *Ilustración 3. Fases del entorno de ejecución.*

Del mismo modo, el SDK de validación se encuentra sometido a una serie de pasos previos a su correspondiente despliegue en el entorno de operación del TOE:

- Teniendo en cuenta el paso de generación de claves descrito para la SDK de generación, se incluyen las claves públicas (o privada para el caso de criptografía simétrica) en un fichero de claves. En este paso, hay que tener en cuenta la inclusión del correspondiente Hardware ID en el fichero para asociar el SDK de validación exclusivamente a una máquina y, además, se incluye el tiempo de vigencia de las claves (1 año de vigencia).
- A continuación, al igual que sucede para el SDK de generación, se hace uso de las llaves de envoltura generadas. Por tanto, se cifra el fichero de claves generado en el paso anterior con la clave privada RSA y la clave AES.
- El siguiente paso es el de incluir la clave pública RSA y la clave simétrica AES (claves de envoltura) en el código fuente del SDK de validación. A partir de este punto se realizará el proceso de compilación habitual adaptado al framework .NET para sistemas de escritorio Windows.
- El resultado generado se ofuscará con la herramienta *ConfuserEx*, para generar un fichero *.dll* con la información ofuscada y protegida.

Al final de este proceso se dispondrá de un SDK de validación para llevar a cabo el proceso de validación, de forma segura, de los códigos QR generados por el SDK de generación.

Con carácter adicional, se ha incluido más información sobre el ciclo de vida de las claves criptográficas utilizadas por el producto en un anexo [DS-ANEXO-001].

4 Problema de seguridad

4.1 Hipótesis sobre el entorno de ejecución

Debido a que las dos partes principales del TOE (SDK de Generación y SDK de Validación) tienen un propósito diferente, se hará una distinción de las hipótesis sobre el entorno de ejecución.

4.1.1 Hipótesis SDK Generación

Código	Descripción
H.Acceso físico	El producto deberá instalarse en un área donde el acceso sólo sea posible para el personal autorizado y con condiciones ambientales adecuadas.
H.Plataforma segura	El producto se ejecutará sobre una plataforma confiable, incluyendo el sistema operativo o cualquier entorno de ejecución sobre el que se utilice.
H.Plataforma cifrada	El producto se ejecutará en una plataforma que implemente cifrado de disco para proteger en reposo los parámetros sensibles de seguridad.
H.Administración	El administrador será un miembro de plena confianza y que vela por los mejores intereses en materia de seguridad de la empresa/administración. Por ello se asume que dicha persona estará capacitada, formada y carecerá de cualquier intención dañina al administrar el producto.
H.Actualizaciones	El software del producto será actualizado conforme aparezcan actualizaciones que corrijan vulnerabilidades conocidas.
H.Criptografía confiable	Toda la criptografía implementada por el entorno operacional que vaya a ser utilizada por el producto deberá cumplir los requisitos descritos en este documento. Esto incluye la generación de factores de autenticación externos.

4.1.2 Hipótesis SDK de Validación

Código	Descripción
H.Plataforma segura	El producto se ejecutará sobre una plataforma confiable, incluyendo el sistema operativo o cualquier entorno de ejecución sobre el que se utilice.
H.Plataforma cifrada	El producto se ejecutará en una plataforma que implemente cifrado de disco para proteger en reposo los parámetros sensibles de seguridad.
H.Administración	El administrador será un miembro de plena confianza y que vela por los mejores intereses en materia de seguridad de la empresa/administración. Por ello se asume que dicha persona estará capacitada, formada y carecerá de cualquier intención dañina al administrar el producto.
H.Actualizaciones	El software del producto será actualizado conforme aparezcan actualizaciones que corrijan vulnerabilidades conocidas.
H.Criptografía confiable	Toda la criptografía implementada por el entorno operacional que vaya a ser utilizada por el producto deberá cumplir los requisitos descritos en este documento. Esto incluye la generación de factores de autenticación externos.

4.2 Activos sensibles a proteger

A continuación, se listan los activos sensibles del producto que se deben de proteger y que a su vez están bajo amenaza.

Código	Descripción	Dimensiones de seguridad
AC.Criptografía	Claves y material criptográfico gestionado por el producto.	Confidencialidad, Integridad, Disponibilidad, Autenticidad, Trazabilidad.
AC.Datos	Datos del usuario que vayan a ser cifrados por el producto	Confidencialidad, Integridad, Disponibilidad, Autenticidad, Trazabilidad.
AC.Configuración	Datos de configuración del producto	Confidencialidad, Integridad, Disponibilidad, Autenticidad, Trazabilidad.

4.3 Descripción de las amenazas

Código	Descripción
A.Acceso no autorizado	Un atacante consigue acceder al dispositivo/herramienta y a los datos contenidos en este. (*AC.Criptografía, AC.Datos, AC.Configuración*)
A.Compromiso del material de cifra	Un atacante consigue acceder a claves, factores de autenticación, números aleatorios o cualquier otro factor que contribuya a crear las claves o factores de autenticación. Esto le permitirá al atacante el acceso a los datos almacenados. (*AC.Criptografía, AC.Datos, AC.Configuración*)
A.Obtención de claves	Un atacante podría llegar a conseguir el acceso a las claves de cifrado, y por lo tanto a la información, mediante la ejecución de aplicaciones software. Esto se produce cuando se eligen algoritmos débiles o con un reducido espacio de claves. (*AC.Criptografía, AC.Datos, AC.Configuración*)
A.Acceso a información en claro	Un atacante podría tener acceso a zonas de memoria que contengan texto plano y que no hayan sido correctamente borradas o zonas conocidas donde se aloje código ejecutable. (*AC.Criptografía, AC.Datos, AC.Configuración*)

5 Funciones de seguridad

A continuación, se incluirá una especificación de las funciones de seguridad que el producto implementa, teniendo en cuenta que esta declaración de seguridad no declara conformidad frente a ninguna de las taxonomías existentes definidas por el catálogo, en el momento de su redacción.

5.1 FS. Requisitos SDK Generación

5.1.1 FS. Requisitos específicos

Esta función de seguridad mitiga la amenaza ***A.Compromiso del material de cifra, A.Obtención de claves, A.Acceso a la información en claro.***

Requisito	Descripción	Justificación
REQ.1	Todos los algoritmos de cifrado simétrico, asimétrico, protocolos de acuerdo de clave y funciones resumen que utilice el producto para llevar a cabo operaciones de cifrado y firma, deberán encontrarse dentro de los acreditados por el CCN para su uso en ENS. El listado de dichos algoritmos se encuentra recogido en la CCN-STIC-807 Criptología de empleo en el ENS (Categoría ALTA).	El producto hace uso de algoritmos de cifrado simétrico como AES-256-CBC, uso de criptosistema de clave pública RSA de 8192 bits y algoritmos de curva elíptica ECC de 256 bits para firma. Además, el producto hace uso del algoritmo SHA-256 bits como función resumen y el algoritmo ECDSA con claves de 256 bits para la verificación de firma digital. Por lo tanto, el producto hace uso de algoritmos que se encuentran recogidos en la guía CCN-STIC-807 Criptología de empleo en el ENS para categoría ALTA.
REQ.2	El producto deberá impedir el acceso en claro a los parámetros de seguridad críticos del sistema (claves simétricas y claves privadas).	El producto realiza el cifrado mediante el algoritmo de cifrado asimétrico RSA de 8192 bits y algoritmo de cifrado simétrico AES de 256 bits de las claves que se insertan en el código fuente de la SDK de generación. De este modo, se establece una protección del material criptográfico que utiliza el producto para proteger la información. Por otra parte, las claves privadas utilizadas por el producto se guardan incrustadas en la biblioteca *.dll* para que

		pueda ser accesible por la lógica del producto. La protección de las claves privadas almacenadas recae sobre el entorno que está cifrado haciendo uso de Bitlocker. Después se ofusca, como medida adicional de protección para las claves cuando están en uso, el código de la propia biblioteca para que no pueda ser leído por un usuario no autorizado. Como medida de protección adicional, el producto asocia un Hardware ID en el momento de la instalación para que sea posible asociarlo de forma unívoca a un equipo solamente, de forma que, si se intenta hacer una instalación en otro equipo, el TOE quede en un estado inutilizable.
REQ.3	Los datos personales asociados a un usuario que son procesados por el producto, se cifrarán con la finalidad de proteger su confidencialidad.	El producto hace uso de algoritmos de cifrado como AES y RSA con la finalidad de proporcionar de varias capas de protección a la información, preservando así su confidencialidad.
REQ.4	Los datos personales asociados a un usuario procesados por el producto, se firmarán mediante el uso de firma digital, con la finalidad de proteger la integridad de dichos datos.	El producto hace uso de algoritmos de curva elíptica como ECDSA con el cual lleva a cabo un firmado de la información personal que procesa. Dicho proceso de firmado le proporciona a la información de autenticidad, garantizando que su emisión fue realizada por una entidad autorizada y por lo tanto evitando la falsificación y no repudio de la información incluida en los códigos QR.
REQ.5	Todos los datos personales asociados a un usuario que el producto almacene en claro serán borrados cuando finalice su uso utilizando un método de borrado por sobreescritura.	El producto no almacena información referente a los datos personales asociados a un usuario. El producto, una vez que recibe los datos los integra en el código QR, cifrándolos y

		preservando así su confidencialidad.

5.1.2 FS. Instalación confiable

Esta función de seguridad mitiga la amenaza *A.Acceso a la información en claro, A.Compromiso del material de cifra.*

Requisito	Descripción	Justificación
REQ.6	El producto no descargará ni modificará su propio código binario.	El producto no lleva a cabo ningún cambio en su código binario una vez que ha sido instalado o durante su ejecución.
REQ.7	El producto solamente utilizará las bibliotecas de terceras partes declaradas por el fabricante.	El producto solamente hace uso de las siguientes bibliotecas declaradas por el fabricante**.** • Aurigma.GraphicsMill.dll • Pkcs11Interop.dll • ChilkatDotNet48.dll • DlibDotNet.dll • DlibDotNetNative.dll • DlibDotNetNativeDnn.dll • dpfj.dll • dpfpdd.dll • dpfr6.dll • DPUruNet.dll • ml_0.bin • nex_sdk.dll • wsq2iso.exe
REQ.8	El producto debe ofrecer la posibilidad de consultar la versión actual del software.	El producto permite consultar la versión actual en ejecución. Para ello, es necesario ver la versión del producto en el campo *Propiedades → Detalles* del fichero *HuBOX_AC.dll*

5.1.3 FS. Capacidad anti-explotación

Esta función de seguridad mitiga la amenaza *A.Acceso no autorizado, A.Acceso a la información en claro, A.Compromiso del material de cifra.*

Requisito	Descripción	Justificación
REQ.9	El producto se auto-protegerá cuando se encuentre en ejecución, de tal forma que tenga acceso exclusivo a su zona de memoria asignada.	El producto hace uso de mecanismos de protección para prevenir ataques durante su ejecución. Esta protección está basada en el flag de compilación

		ASLR (*Address Space Layout Randomization*). Esta flag de compilación está asociada al proceso de compilación de aplicaciones de escritorio para el sistema operativo Windows. Por otro lado, el sistema operativo Windows es el responsable de que cada proceso tenga acceso exclusivo a su zona de memoria, es decir, el código de un proceso no puede acceder a la memoria física que está siendo usada por otro proceso.
REQ.10	El producto y/o su entorno operacional deben de disponer de medidas de protección frente ataques de ingeniería inversa.	El producto hace uso de la herramienta *ConfuserEx* como ofuscador de código como medida de protección frente a ataques de ingeniería inversa. Además, esta herramienta ofrece también una protección contra herramientas de desensamblado (como *ILdasm*).
REQ.11	El producto y/o su entorno operacional deben de disponer de mecanismos de protección frente ataques *tampering*, con la finalidad de evitar modificaciones no autorizadas de su código fuente.	El uso de la herramienta *ConfuserEx*, utilizado durante la generación del código fuente del producto, provee a dicho código de una capa de protección que evita la modificación no autorizada de su código fuente, así como, permite la detección de este tipo de ataques en caso de que hayan ocurrido.

5.2 FS. Requisitos SDK Validación

5.2.1 FS. Requisitos específicos

Esta función de seguridad mitiga la amenaza ***A.Compromiso del material de cifra, A.Obtención de claves, A.Acceso a la información en claro.***

Requisito	Descripción	Justificación
REQ.12	Todos los algoritmos de cifrado simétrico, asimétrico, protocolos de acuerdo de clave y funciones resumen que utilice el producto para llevar a cabo operaciones de descifrado, verificación y validación deberán encontrarse dentro de los acreditados por el CCN para su uso en ENS. El listado de dichos algoritmos se	El producto hace uso de algoritmos de cifrado simétrico como AES-CBC-256 bits, uso de criptosistema de clave pública RSA de 8192 bits y algoritmos de curva elíptica ECC de 256 bits para firma. Además, el producto hace

	encuentra recogido en la CCN-STIC-807 Criptología de empleo en el ENS (Categoría ALTA).	uso del algoritmo SHA-256 bits como función resumen y el algoritmo ECDSA con claves de 256 bits para la verificación de firma digital. Por lo tanto, el producto hace uso de algoritmos que se encuentran recogidos en la guía CCN-STIC-807 Criptología de empleo en el ENS (Categoría ALTA).
REQ.13	El producto deberá impedir el acceso en claro a los parámetros de seguridad críticos del sistema (claves simétricas y claves privadas).	El producto realiza el cifrado mediante el algoritmo de cifrado asimétrico RSA de 8192bits y simétrico de AES-CBC-256 bits de las claves que se insertan en el código fuente de la SDK de validación. De este modo, se establece una protección del material criptográfico que utiliza el producto para proteger la información. Por otra parte, las claves privadas utilizadas por el producto se guardan incrustadas en la biblioteca *.dll* para que pueda ser accesible por la lógica del producto. Después se ofusca el código de la propia biblioteca para que no pueda ser leído por un usuario no autorizado. En cada actualización, además, se necesita generar una nueva clave antes de incrustarla. Mediante la biblioteca criptográfica del producto se puede llevar a cabo dicho procedimiento. Además, todo este material criptográfico se encuentra protegido mediante el uso del algoritmo de cifrado asimétrico RSA de 8192 bits y simétrico AES-CBC-256 para añadir una capa adicional de protección. Como medida de protección adicional, el producto asocia un Hardware ID en el momento de la instalación para que sea posible asociarlo de forma unívoca a un equipo solamente, de forma que, si se intenta hacer una instalación

		en otro equipo, el TOE quede en un estado inutilizable.
REQ.14	Los datos personales asociados a un usuario que son procesados por el producto, se descifrarán con la finalidad de verificar su confidencialidad.	El producto dispone de los mecanismos necesarios para el descifrado de la información codificada en los códigos QR. De esta forma se puede verificar la integridad, mediante el procedimiento inherente a la utilización de los algoritmos AES y RSA.
REQ.15	Los datos personales asociados a un usuario procesados por el producto, se verificarán mediante el uso de firma digital o un mecanismo similar, con la finalidad de comprobar la integridad de dichos datos.	El producto dispone de los mecanismos necesarios para llevar a cabo la verificación de la información incluida en los códigos QR mediante su firma digital. Para ello hace uso del procedimiento inherente a la utilización al algoritmo de curva elíptica ECC.

5.2.2 FS. Instalación confiable

Esta función de seguridad mitiga la amenaza ***A.Acceso a la información en claro, A.Compromiso del material de cifra.***

Requisito	Descripción	Justificación
REQ.16	El producto no descargará ni modificará su propio código binario.	El producto no lleva a cabo ningún cambio en su código binario una vez que ha sido instalado o durante su ejecución.
REQ.17	El producto solamente utilizará las bibliotecas de terceras partes declaradas por el fabricante.	El producto solamente hace uso de las siguientes bibliotecas declaradas por el fabricante**.** • Aspose.BarCode.dll • Aurigma.GraphicsMill.dll • Pkcs11Interop.dll • ChilkatDotNet48.dll • Neurotec.dll • Neurotec.Accelerator.Admin.Rest.dll • Neurotec.Biometrics.dll • Neurotec.Biometrics.Client.dll • Neurotec.Biometrics.Gui.dll • Neurotec.Devices.dll • Neurotec.Gui.dll • Neurotec.Licensing.dll

		• Neurotec.Media.dll • Neurotec.Media.Processing.dll
REQ.18	El producto debe ofrecer la posibilidad de consultar la versión actual del software.	El producto permite consultar la versión actual en ejecución. Para ello, es necesario ver la versión del producto en el campo *Propiedades → Detalles* del fichero *HuBOX_AV.dll*

5.2.3 FS. Capacidad anti-explotación

Esta función de seguridad mitiga la amenaza ***A.Acceso no autorizado, A.Acceso a la información en claro, A.Compromiso del material de cifra.***

Requisito	Descripción	Justificación
REQ.19	El producto se auto-protegerá cuando se encuentre en ejecución, de tal forma que tenga acceso exclusivo a su zona de memoria asignada.	El producto hace uso de mecanismos de protección para prevenir ataques durante su ejecución. Esta protección está basada en el flag de compilación, ASLR (*Address Space Layout Randomization*). Esta flag de compilación está asociada al proceso de compilación de aplicaciones de escritorio para el sistema operativo Windows. Por otro lado, el sistema operativo Windows es el responsable de que cada proceso tenga acceso exclusivo a su zona de memoria, es decir, el código de un proceso no puede acceder a la memoria física que está siendo usada por otro proceso.
REQ.20	El producto y/o su entorno operacional deben de disponer de medidas de protección frente ataques de ingeniería inversa.	El producto hace uso de la herramienta *ConfuserEx* como ofuscador de código como medida de protección frente a ataques de ingeniería inversa. Además, esta herramienta ofrece también una protección contra herramientas de desensamblado (como *ILdasm*). Esta capa de protección adicional se realiza en el SDK de validación para sistemas operativos de escritorio Windows.

REQ.21	El producto y/o su entorno operacional deben de disponer de mecanismos de protección frente ataques *tampering*, con la finalidad de evitar modificaciones no autorizadas de su código fuente.	El uso de la herramienta *ConfuserEx*, utilizado durante la generación del código fuente del producto, provee a dicho código de una capa de protección que evita la modificación no autorizada de su código fuente, así como, permite la detección de este tipo de ataques en caso de que hayan ocurrido.

6 Acrónimos

La siguiente tabla muestra los acrónimos usados en este documento.

Acrónimo	Significado
CCN	Centro Criptológico Nacional.
CNI	Centro Nacional de Inteligencia.
ENS	Esquema Nacional de Seguridad.
LINCE	Certificación Nacional de Seguridad.
MCF	Módulo de revisión de Código Fuente.
MEC	Módulo de Evaluación Criptográfica.
MITM	Man In The Middle
SDK	Kit de Desarrollo de Software (*Software Development Kit*)
ST	Security Target.
STIC	Seguridad de las Tecnologías de la Información y la Comunicación.
TIC	Tecnologías de la Información y la Comunicación.
TOE	Target Of Evaluation - Objeto a evaluar.
TSS	TOE Summary Specification.
HSM	*Hardware Security Module*
ECDSA	*Elliptic Curve Digital Signature Algorithm*
AES	*Advanced Encryption Standard*
RSA	Rivest, Shamir y Adleman
ECC	*Elliptic curve* cryptography
SHA	*Secure Hash Algorithm*
QR	Quick Response

Tabla 1 *Acrónimos*

7 Glosario de términos

Término	Significado
Entorno operacional	Entorno en el que se opera el TOE.
Declaración de Seguridad	Declaración de las necesidades de seguridad para un TOE específicamente identificado con independencia de la implementación.
Target Of Evaluation	Conjunto de software, firmware y/o hardware objeto de la evaluación.

Tabla 2 Glosario de términos

8 Documentos referenciados

La siguiente tabla muestra los documentos que han sido referenciados.

Referencia	Documento
CCN-STIC-2001	Metodología de Evaluación para la Certificación Nacional Esencial de Seguridad (LINCE).
CCN-STIC-2002	Metodología de Evaluación para la Certificación Nacional Esencial de Seguridad (LINCE).
CCN-STIC-2003	Plantilla para la Declaración de Seguridad de la Certificación Nacional Esencial de Seguridad (LINCE).
CCN-STIC-2004	Plantilla del Informe Técnico de Evaluación de la Certificación Nacional Esencial de Seguridad (LINCE).
DS-ANEXO-001	Anexo 1. Ciclo de vida de las claves criptográficas.

Tabla 3 *Lista de referencias*