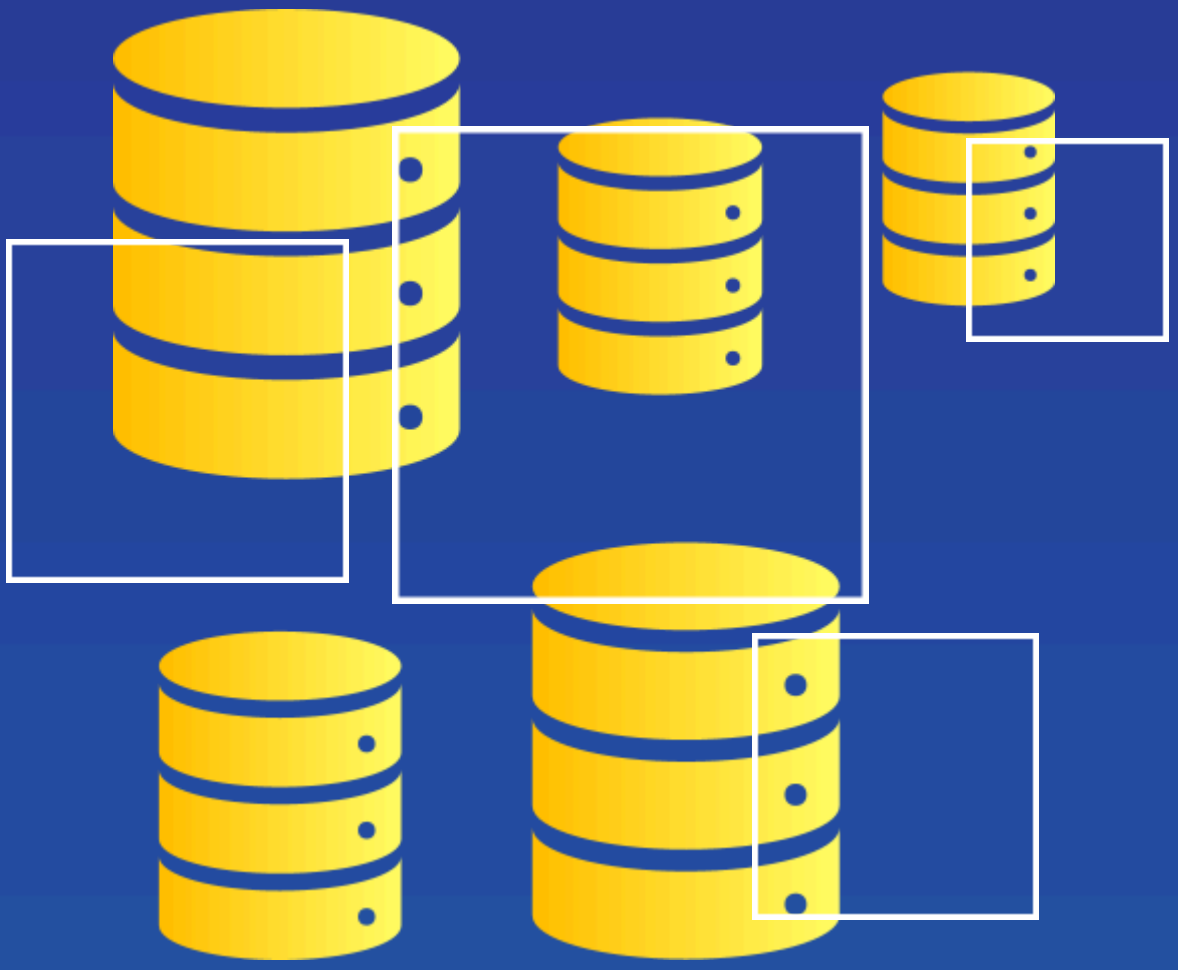
CRIPTOGRAFÍA Y SEGURIDAD



Nombre alumno 1:Adrián Vázquez Santiago

NIA 1:100522300

Correo 1:100522300@alumnos.uc3m.es

Nombre alumno 2:Ivan ciller Lopez

NIA 2:100522245

Correo 2:100522245@alumnos.uc3m.es

Introducción

En esta práctica se ha desarrollado una aplicación de gestión de vuelos que incorpora mecanismos de seguridad basados en la librería cryptography y openssl. Además de las funciones habituales de registro, inicio de sesión y creación de reservas, el sistema integra tres componentes esenciales: firma digital, gestión segura de claves e infraestructura de clave pública (PKI).

La aplicación genera y almacena claves RSA para cada usuario, crea solicitudes de certificado (CSR) y utiliza una Autoridad de Certificación Raíz (AC1) para firmar los certificados X.509 que autentican sus claves públicas. La Autoridad de Certificación Raíz (AC1) utilizada en la aplicación es un certificado autofirmado. Esto significa que no existe una entidad superior que respalde su autenticidad, sino que se confía en ella de forma explícita por ser la raíz de la infraestructura. A partir de estos certificados, la aplicación puede verificar firmas digitales y cifrar información de manera segura, garantizando la confidencialidad, integridad, autenticidad y no repudio de las reservas.

Los apartados de esta memoria describen el uso de la firma digital, la implementación de la PKI y la gestión de claves dentro de la aplicación, justificando las decisiones de diseño adoptadas y mostrando cómo estas técnicas han sido integradas en el código para crear un entorno seguro y funcional.

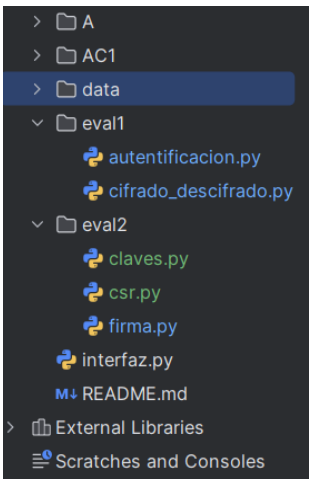
Preguntas

1-¿Cuál es el propósito de la aplicación?

El propósito de la aplicación es proporcionar una aplicación segura para gestionar vuelos y reservas, garantizando la confidencialidad, integridad y autenticidad de las operaciones realizadas por los usuarios. Para ello, la aplicación implementa varios mecanismos criptográficos:

- **Autenticación de usuarios** mediante contraseñas protegidas con *Scrypt*.
- **Generación y gestión de claves RSA** de cada usuario.
- **Creación de solicitudes de certificado (CSR)** y obtención de certificados firmados por la **Autoridad de Certificación Raíz (AC1)**.
- **Cifrado híbrido** para proteger las reservas (AES-GCM + RSA-OAEP).
- **Firma digital y verificación** para garantizar la integridad de las reservas.
- **Uso de certificados X.509** para autenticar claves públicas.

El sistema combina una infraestructura completa de PKI con técnicas modernas de cifrado simétrico y asimétrico para garantizar la seguridad de extremo a extremo de todos los datos gestionados.



2-¿Para qué se usa la firma digital?

La firma digital está integrada en la aplicación como un mecanismo esencial para garantizar la **integridad** y **autenticidad** de las reservas generadas por los usuarios. Aunque el sistema cifra estas reservas para proteger su contenido, el cifrado no impide que un tercero pueda intentar modificarlas; por este motivo, la firma digital actúa como un "sello" que certifica tanto su origen como su invariabilidad desde el momento en que fueron generadas.

Cuando un usuario crea una reserva, el sistema produce una representación completa del contenido en texto plano y, antes de cifrar, la firma utilizando la clave privada del propio usuario. De este modo, cada reserva queda vinculada a la identidad del autor. Es importante recalcar que, aunque la reserva se cifra mediante un esquema híbrido, el cifrado por sí mismo no garantiza la integridad del contenido. Un atacante podría modificar el texto cifrado o intentar sustituir la clave simétrica cifrada. La firma digital evita este tipo de ataques porque vincula el contenido original con la clave privada del usuario.

En la fase de comprobación, el sistema reconstruye la información descifrada y verifica la firma mediante la clave pública contenida en el certificado del usuario. Si alguien hubiese alterado cualquier parte del contenido, la verificación fallará inmediatamente.

Esta incorporación de la firma digital no solo fortalece la seguridad del sistema, sino que permite a cada usuario confiar en que sus reservas no podrán ser manipuladas sin que el sistema lo detecte.

```
def firmar_mensaje(mensaje: bytes,rsa_private_pem: bytes,passphrase: bytes | None = None) -> bytes:
    """
    Genera una firma digital RSA del mensaje.
    """
```

```
def verificar_firma_mensaje(mensaje: bytes,firma: bytes,rsa_public_pem: bytes) -> bool:
    """
    Se verifica que la firma corresponda al mensaje.
        - Si la firma coincide → el mensaje es auténtico y no ha sido modificado.
        - Si falla → se lanza una excepción, capturada y devuelta como False.
    """
```

3-¿Qué algoritmos ha utilizado y por qué?

El sistema emplea un esquema de firma digital basado en **RSA** combinado con un resumen criptográfico seguro. Esta elección responde a varias razones:

- RSA es un algoritmo ampliamente utilizado en sistemas reales, completamente compatible con los certificados X.509 y con la infraestructura de clave pública manejada por OpenSSL.
- Es una solución estándar para procesos de firma digital, fiable y adecuada para un entorno donde se simula una autoridad de certificación.
- Permite integrar de forma natural la firma digital con la gestión de certificados y la verificación posterior dentro de la aplicación.

No se ha considerado necesario utilizar alternativas más complejas o específicas, ya que RSA proporciona un equilibrio adecuado entre robustez, compatibilidad y facilidad de integración con la PKI creada para la práctica.

4-¿Cómo se gestionan y almacenan las claves y las firmas?

La aplicación incorpora un enfoque estructurado para la gestión de claves, prestando especial atención a la protección de la clave privada del usuario.

Durante el proceso de registro, se genera para cada usuario un par de claves RSA. La clave privada se guarda en un fichero protegido mediante cifrado derivado de la contraseña del propio usuario,de este modo, aunque alguien acceda al fichero, no podrá utilizarlo sin conocer dicha contraseña.El componente que realmente importa respecto a la clave pública es la incluida dentro de su certificado,del certificado del usuario (X.509).Esto garantiza que la clave utilizada para la verificación está autenticada por la AC1, evitando ataques donde un usuario suplante una clave pública distinta de la registrada en la autoridad.

En lo relativo a las firmas generadas, estas nunca se almacenan por separado ni en texto claro. Tras firmar la reserva, la firma se integra dentro del objeto cifrado que representa dicha reserva. Así, tanto la información firmada como la propia firma quedan protegidas dentro del mismo mecanismo de cifrado híbrido, garantizando confidencialidad e integridad de forma conjunta.

5-¿Cómo se generan los certificados de clave pública?

La aplicación genera certificados siguiendo el flujo habitual de una infraestructura de clave pública. Después de crear su par de claves, el usuario produce automáticamente una solicitud de certificado (Certificate Signing Request, CSR). Este documento contiene su clave pública y los atributos necesarios para identificarlo (país,organización,nombre,correo).

La solicitud se envía a la **Autoridad de Certificación Raíz (AC1)**, qué es la encargada de firmarla. Una vez la AC1 firma el CSR, se genera un certificado X.509 válido, que queda almacenado en la carpeta del propio usuario. La aplicación trabaja siempre con este certificado ya firmado, del cual extrae la clave pública para verificar firmas y cifrar reservas.

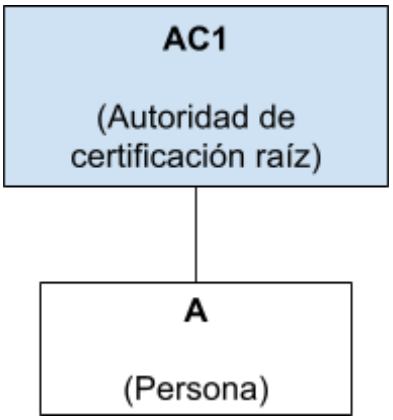
El proceso de generación del certificado se realiza a través de herramientas OpenSSL. De este modo, la práctica reproduce el comportamiento de una CA real, con su directorio de certificados emitidos, índice de solicitudes y archivo de números de serie.

6-¿Qué jerarquía de autoridades de certificación se ha desplegado? ¿Por qué ha escogido esta configuración y no otra?

Para esta práctica se ha desplegado únicamente una **Autoridad de Certificación Raíz (AC1)**. No se han creado autoridades subordinadas. Esta decisión responde principalmente a dos motivos:

1. **Simplicidad y foco en los objetivos de la práctica.**
   La introducción de autoridades subordinadas complicaría la estructura sin aportar beneficios concretos para los requisitos planteados.
2. **Coherencia operativa.**
   Una única autoridad es suficiente para emitir certificados a todos los usuarios y verificar posteriormente sus firmas dentro del sistema.
3. **Recomendaciones realizadas en el aula.**
   Aunque una PKI puede contener múltiples autoridades subordinadas, en este proyecto no aporta beneficios prácticos. El profesor recomendó utilizar únicamente una AC raíz para centrar el trabajo en los aspectos esenciales: firma digital y verificación mediante certificados. Una única AC resulta suficiente para emitir, mantener y validar todos los certificados del sistema.A demás confiamos únicamente en esta AC raíz

La AC1 cumple todas las funciones requeridas: firma los certificados de los usuarios, proporciona el certificado raíz para las verificaciones y mantiene el registro interno de certificados emitidos.A continuación se muestra un esquema para representar visualmente lo explicado



8-¿Cómo se ha implementado?

La Autoridad de Certificación Raíz (AC1) se ha implementado utilizando OpenSSL siguiendo la estructura estándar de una autoridad X.509. Para ello se ha creado un directorio específico que contiene su clave privada, su certificado autofirmado, el fichero de configuración (openssl_AC1.cnf) y los archivos de control necesarios para la emisión de certificados (index.txt, serial, carpetas de solicitudes y certificados emitidos).

Cuando un usuario se registra, la aplicación genera localmente su par de claves RSA y crea automáticamente una solicitud de certificado (CSR) con la información de identidad del usuario. Este CSR se almacena en la carpeta de solicitudes de AC1 y es firmado mediante el comando openssl ca, empleando la clave privada de la autoridad y la configuración definida en el fichero de la AC. El resultado es un certificado X.509 válido que vincula la clave pública del usuario con su identidad y queda registrado en la base de datos interna de AC1.

Una vez emitido, el certificado se entrega al usuario y se almacena junto a sus claves en la aplicación. Desde ese momento se utiliza para dos operaciones esenciales: la verificación de firmas digitales y la extracción de la clave pública durante los procesos de cifrado asimétrico del sistema.

Además, la aplicación incorpora un paso adicional para garantizar la autenticidad del certificado antes de utilizarlo. Concretamente, se emplea el método verify_directly_issued_by() de la librería **cryptography**, que verifica tres condiciones:

1. que la firma del certificado del usuario puede comprobarse correctamente con la clave pública de la AC1;
2. que el el certificado del usuario debe indicar que ha sido emitido por la misma autoridad cuya identidad aparece en el certificado de la AC1, sin diferencias en ningún atributo.;
3. que los algoritmos indicados en el certificado son compatibles con los que soporta la autoridad.

Este método puede fallar si existe cualquier discrepancia en los atributos de la identidad (por ejemplo, en stateOrProvinceName u organizationName), o si la configuración de la CA no coincide con la utilizada al generar los CSR. Por ello, fue necesario mantener una coherencia estricta entre los valores incluidos en el CSR y los definidos en el fichero de configuración de OpenSSL,estableciendo estos valores como “optional”.Este comportamiento refleja fielmente cómo funcionan las validaciones formales en una PKI real.

9-¿En qué momento se utilizan los certificados y para qué?

Los certificados intervienen en dos momentos fundamentales del funcionamiento de la aplicación:

1) **Durante la verificación de firmas digitales-**Cuando un usuario abre una reserva cifrada, el sistema extrae la clave pública desde su certificado y la utiliza para comprobar si la firma almacenada coincide con los datos descifrados. Es el certificado quien proporciona la garantía formal de que esa clave pública pertenece realmente al usuario.

2) **Durante el cifrado híbrido de reservas-**La clave pública usada para cifrar la clave simétrica de la reserva también se obtiene del certificado X.509. Esto asegura que la información cifrada sólo podrá ser descifrada por quien posee la clave privada correspondiente, es decir, el propio usuario.

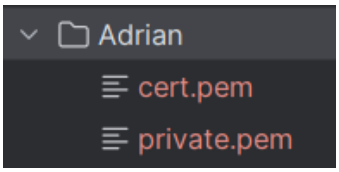
En ambos casos, el certificado actúa como vínculo entre la identidad del usuario y su clave pública, reforzando la confianza global del sistema.Gracias a la firma de la AC1, la aplicación puede confiar en que la clave pública asociada al usuario es auténtica y pertenece realmente a él. La verificación del certificado garantiza que no se ha modificado y que ha sido emitido por una autoridad confiable.

Pruebas

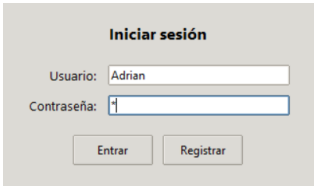
Prueba:Verificación de la firma digital	Resultado
El objetivo de esta prueba es comprobar que, al realizar una reserva, el sistema genera correctamente una firma digital RSA y que dicha firma puede ser verificada posteriormente. De esta forma validamos que la práctica cumple con los estándares de autenticidad e integridad requeridos.	

1. **Creación del usuario**
   Se inicia el proceso registrando un nuevo usuario dentro de la aplicación.Para poder realizar reservas, es imprescindible disponer de una cuenta registrada, ya que cada usuario posee su propio par de claves RSA.

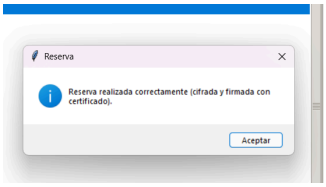
2. **Generación del certificado del usuario**
   Tras registrar el usuario, es necesario su certificado y se coloca en su carpeta correspondiente.Este certificado es necesario para poder firmar y posteriormente verificar las reservas.



3. **Inicio de sesión**
   Dentro de la interfaz gráfica, iniciamos sesión con el usuario recién creado para acceder al panel principal.

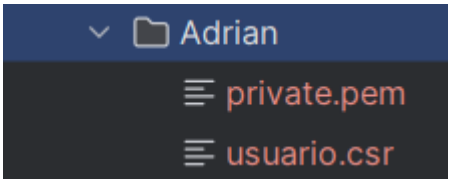


4. **Realización de una reserva**
   Se selecciona uno de los vuelos listados y se procede a generar una reserva.Durante este proceso:
   a. La información de la reserva se generaliza en JSON.
   b. Se firma digitalmente utilizando la clave privada RSA del usuario.
   c. El archivo resultante se cifra mediante el sistema de cifrado híbrido de la práctica.



5. **Verificación de la reserva generada**
   Finalmente, se accede al directorio de reservas del usuario y se comprueba que:
   a. El archivo generado incluye una firma digital.
   b. Al visualizar la reserva desde la interfaz, en la sección “Mis reservas”, el sistema muestra el estado “Firma: OK”, indicando que la verificación con el certificado del usuario ha sido satisfactoria.

   En caso contrario, si la firma fuera incorrecta o los datos hubieran sido manipulados, aparecería un estado “ERROR”, señalando un fallo en la integridad del mensaje.

Prueba:Generación del CSR	Resultado
Para esta prueba comprobaremos que una vez registrado un usuario en el sistema, la aplicación debe generar automáticamente un **CSR**	

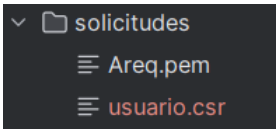
1. **Registro del usuario**
   Se inicia el procedimiento creando un nuevo usuario desde la interfaz.Durante el registro, el sistema:
   a. Calcula y almacena el hash seguro de la contraseña mediante Scrypt.
   b. Genera el par de claves RSA del usuario.
   c. Cifra la clave privada utilizando la propia contraseña.
   Este conjunto de operaciones es necesario para garantizar la seguridad del usuario dentro del sistema.

2. **Generación del certificado del usuario**
   Una vez generadas las claves RSA, la aplicación ejecuta automáticamente la función generar_csr_usuario().Esta función crea un CSR que contiene la identidad del usuario,incluyendo (país,provincia etc.)

3. **Verificación en los archivos**
   Tras finalizar el registro, se comprueba que dentro del directorio del usuario se ha creado el archivo junto a la clave privada cifrada (private.pem), este archivo confirma que el sistema ha generado correctamente la solicitud de certificado.

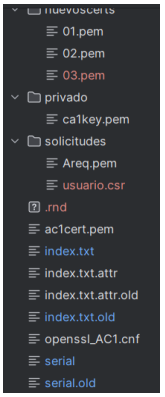
Prueba:Firma de certificados	Resultado
El objetivo de esta prueba es demostrar que el sistema permite firmar manualmente un certificado X.509 utilizando la herramienta OpenSSL, y que este proceso funciona correctamente a partir del CSR generado por el usuario en la prueba anterior.	

1. **Mover el CSR del usuario a la Autoridad Certificadora (AC1)**
   Tras registrar al usuario, el archivo .csr se desplaza manualmente a la carpeta AC1/solicitudes .Esta carpeta es utilizada por la autoridad de certificación para gestionar las solicitudes pendientes de firma.
   AC1 actúa como la entidad de confianza responsable de emitir y validar certificados dentro del sistema.



2. **Firma del certificado mediante OpenSSL**
   Una vez el CSR se encuentra en la carpeta de solicitudes, utilizamos los comandos apropiados de OpenSSL para realizar su firma.
   Como resultado del proceso:
   a. Se actualizan automáticamente los archivos **serial**, **serial.old**, **index.txt** e **index.txt.old**, utilizados por la AC para llevar el control de certificados emitidos.
   b. Se genera un nuevo archivo 03.pem correspondiente al certificado recién emitido

   Este archivo contiene el certificado X.509 firmado por la AC y vinculado inequívocamente al usuario que generó el CSR.



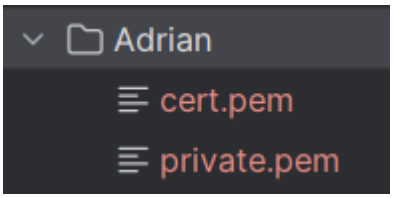
3. **Entrega del certificado al usuario**

   Finalmente, el certificado firmado se copia a la carpeta del usuario:Este paso es esencial:
   El certificado firmado habilita al usuario para realizar y verificar reservas, ya que la aplicación utiliza este certificado para validar que las firmas digitales pertenecen realmente a un usuario reconocido por la AC.

   Por qué es necesario:
   En esta práctica, el sistema está diseñado para que toda validación de firmas dependa del certificado emitido por la AC.Si un usuario no posee su cert.pem firmado:

   - no puede generar reservas válidas,
   - no puede verificar reservas antiguas,
   - y la aplicación indica errores al comprobar la firma.

   Esto incrementa la seguridad, garantizando que únicamente los usuarios con certificados válidos pueden operar correctamente.




```
PS C:\Users\Trace\PycharmProjects\CRIPTO\AC1> mv nuevoscerts\03.pem ..\data\keys\Adr
ian\cert.pem
```

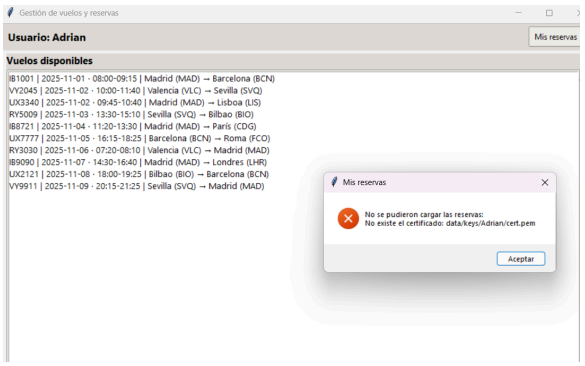
Información adicional de la prueba

Funcionalidades que proporciona que el usuario tenga el certificados

Una vez el usuario posee un certificado firmado por la AC:

- ☑ Puede realizar reservas, ya que la firma generada será verificable.
- ☑ Puede visualizar sus reservas descifradas.
- ☑ Puede demostrar su identidad digital ante el sistema.
- ☑ Puede validar la integridad de los datos firmados.

Sin el certificado, muchas de estas funciones fallan, lo cual confirma el correcto diseño de dependencia entre usuario y AC.



Errores probados durante la prueba

CSR modificado o no generado por la AC

Se probó a introducir un certificado inventado o no firmado por la AC.El sistema detectó correctamente la invalidez, mostrando un error al intentar verificar reservas.Este comportamiento confirma que solo los certificados emitidos por la AC son aceptados.

Configuración incorrecta del algoritmo de firma

Durante la prueba se observó que, si OpenSSL firmaba usando un algoritmo distinto a SHA-256, la aplicación no podía validar correctamente el certificado.Tras configurar OpenSSL para usar SHA256, la validación funcionó como se esperaba.

```
default_md  = sha256
```