Problemas

1. Funções do middleware chamadas em contextos diferentes têm comportamentos diferentes. Exemplo:

- `PTEID_Pins.getPinByPinRef` no construtor da classe `CitizenCard` funciona como esperado
- `PTEID_Pins.getPinByPinRef` no método `CitizenCard.login()` (chamado posteriormente) lança `PTEID_Exception`

2. Pretende-se criar uma ligação SSL utilizando os certificados do cartão, e para tal recorre-se funções comuns das bibliotecas de segurança do java, implementadas pelo provider de SunPKCS11 que utiliza a implementação da libpteidpkcs11. A inicialização é a seguinte:

Construtor da classe `CitizenCard`

```
// inicialização do provider
String pkcs11Config = "--name=cartao-cidadao\nlibrary=path/to/libpteidpkcs11.so\n"
pkcs11Config+= "slotListIndex={{slot}}\nattributes=compatibility";
try {
    sunpkcs11 = Security.getProvider("SunPKCS11").configure(pkcs11Config);
    Security.addProvider(sunpkcs11);
} catch (Exception e) { ... }
```

Todas as chamadas funções ou métodos que utilizam o provider `sunpkcs11` (como `KeyStore.getInstance` ou `SSLSocket.startHandshake()`) posteriores à sua inicialização e fora do método onde foi inicializado lançam a exceção: `ProviderException: "Token has been removed"`

Um exemplo é a criação de uma Keystore PKCS11:

```
KeyStore.PasswordProtection pp = new KeyStore.PasswordProtection(pin.toCharArray());
KeyStore ks = null;
try {
    ks = KeyStore.getInstance("PKCS11", sunpkcs11);
    ks.load(null, null);
    this.keyManagerFactory = KeyManagerFactory.getInstance("SunX509");
    this.keyManagerFactory.init(ks, null);

    this.tmf = TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
    tmf.init((KeyStore) null);
} catch { ... }
```

Isto levou-nos a pensar que podia ser um problema de concorrência ou de permissões relativas a Threads mas testámos isoladamente estes métodos em cenários multi-threaded e funcionaram como esperado.

Nos casos em que estas funções funcionam e tentamos estabelecer a ligação SSL observam-se comportamentos diferentes com cartões diferentes:

- Estão a ser testados dois cartões distintos, um emitido em Julho de 2016 e outro em Junho de 2019.
- No cartão mais velho, a ligação é estabelecida com sucesso.
- No cartão mais recente, ao invocar `SSLSocket.startHandshake()` são lançadas as exceções:

```
SSLHandshakeException: Cannot produce CertificateVerify signature
Caused by: java.security.InvalidKeyException: RSA key must be at most 2048 bits
```

Assinado usando o Cartão de Cidadão