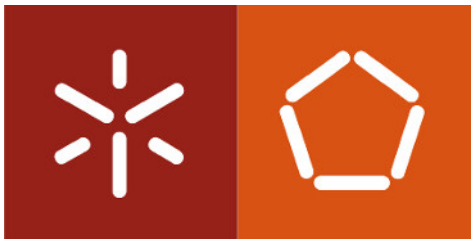

Mestrado em Engenharia Informática
Universidade do Minho

Engenharia de Segurança

Projeto 3 - Reverse Engineer da Aplicação CMD-SOAP

João Miranda - PG41845
Sandro Cruz - PG41906

6 de Julho de 2020

Conteúdo

Capítulo 1

Introdução

1.1 Contextualização

Neste projecto foi nos pedido para fazer-mos o *reverse engineering* da aplicação CMD_SOAP, que foi desenvolvida em Python 3 e que tivemos de passar para C++. A aplicação CMD_SOAP é uma aplicação de linha de comandos, que utilizando as credencias da chave móvel digital é possível testar as operações do serviço SCMD (Signature CMD).
Dadas as diferenças entre as linguagens, Python 3 e C++, levou a que aparecessem bastantes desafios e complicações, que iremos abordar e explicar neste relatório.

1.2 Objetivos

Os principais objectivos do projecto foram:

- Desenvolver software seguro utilizando técnicas que aprendemos ao longo desta unidade curricular.
- Criar uma aplicação em que seja possível executar testes ao software.

Capítulo 2

Noções Essenciais

2.1 Chave Móvel Digital

A Chave Móvel Digital (CMD) é um meio de autenticação e assinatura digital certificado pelo Estado português. Permite ao utilizador aceder a vários portais públicos ou privados, e assinar documentos digitais, com um único login. A Chave Móvel Digital associa um número de telemóvel ao número de identificação civil para um cidadão português, e o número de passaporte ou título/cartão de residência para um cidadão estrangeiro.(1)

2.2 Serviço SCMD

O serviço SCMD consiste na Signature CMD ou seja a utilização da chave móvel digital para fazer a assinatura de um documento, no caso em questão é utilizada o web service em SOAP disponibilizada pelo estado português para este efeito.

2.3 SOAP

SOAP ou Protocolo Simples de Acesso a Objetos, é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída. Baseia na Linguagem de Marcação Extensível (XML) para o formato das mensagens, e normalmente baseia-se em outros protocolos da camada de aplicação, mais notavelmente na chamada de procedimentos remotos (RPC) e Protocolo de transferência de hipertexto (HTTP), para negociação e transmissão de mensagens.SOAP pode formar a camada base de uma *Web services*, fornecendo uma *framework* básica das mensagens sob o qual se podem construir os serviços Web.(2)

2.4 OpenSSL

OpenSSL é uma implementação de código aberto dos protocolos SSL e TLS. A biblioteca (escrita na linguagem C) implementa as funções básicas de criptografia e disponibiliza várias funções utilitárias. Também estão disponíveis *wrappers* que permitem o uso desta biblioteca em várias outras linguagens.(3)

2.5 Base 64

Base64 é um método para codificação de dados para transferência na Internet (codificação MIME para transferência de conteúdo). É utilizado frequentemente para transmitir dados binários por meios de transmissão que lidam apenas com texto, como por exemplo para enviar arquivos anexos por e-mail. É constituído por 64 caracteres ([A-Z],[a-z],[0-9], "/"e "+") que deram origem ao seu nome. O carácter "="é utilizado como um sufixo especial e a especificação original (RFC 989) definiu que o símbolo "*"pode ser utilizado para delimitar dados convertidos, mas não criptografados, dentro de um stream.(4)

2.6 C++ vs Python

Tanto o Python como o C++ são linguagens de programação para uso geral, mas ambas diferem muito uma da outra. O C++ é uma linguagem descendente da linguagem C com vários paradigmas e fornece o recurso de compilação, em que cada programa em que ser compilado para cada sistema operativo de forma a correr o código, enquanto que o Python é uma linguagem de uso geral e uma das linguagens de mais alto nível, em que pode correr em qualquer máquina desde que tenha o Python instalado.
No C ++, o programador precisa declarar o tipo de dados antes de usá-lo. Portanto, é menos ambíguo com relação ao que os códigos fazem e o tratamento de erros se torna mais fácil que o python. Ao escrever o código em Python, o programador não precisa mencionar o tipo de dados antes de usá-los, consequentemente, tornando o tamanho do código mais curto e fácil de manter. Por exemplo, em C ++, um programador deve declarar int a = 5 enquanto em Python a = 5 é suficiente. O C ++ também é chamado de linguagem de programação de nível intermediário, pois é desenvolvido usando recursos de linguagem de nível baixo e alto. O C ++ também suporta funcionalidades orientadas a objetos, como o conceito de classes, sobrecarga de operadores, múltiplas heranças, funções virtuais, manipulação de exceções etc.(5)

Capítulo 3

Implementação do Sistema

3.1 Caracterização Geral do Sistema

O sistema CMD-SOAP - Teste das operações do serviço SCMD (Signature CMD) caracteriza-se por um conjunto de operações do serviço SCMD (Signature CMD), de forma a poder ser feita uma assinatura digital num ficheiro pretendido. Este sistema baseou-se numa aproximação do programa elaborado em *Python*, tentando-se sempre uma aplicação da coerência pretendida.

3.1.1 Cliente

O lado do cliente consiste no envio de pedidos para uma destas operações e a obtenção de uma resposta positiva em relação a estes. Assim, o cliente pode obter todos os pedidos que deseja, validando corretamente a sua identidade (autenticidade).

3.1.2 Servidor

O servidor é baseado em *Soap*, onde os pedidos são recebidos e respondidos sob a forma de XML, de forma a poder-se processar os dados relativos ao utilizador (Cliente). Os pedidos são feitos para o wsdl que é o *Frontend* de autenticação da AMA.

3.2 Bibliotecas Utilizadas

3.2.1 ArgParse

É uma biblioteca que facilita a validação e a análise na entrada dada pelo utilizador do programa. Facilita a criação de interfaces da linha de comandos. O módulo argparse também gera automaticamente mensagens de ajuda e uso e emite erros quando os usuários fornecem argumentos inválidos ao programa.

3.2.2 Base 64

Como já foi abordado anteriormente, esta biblioteca explora a transmissão de dados binários por meios de transmissão que lidam apenas com texto.

3.2.3 OpenSSL

Também já foi abordado anteriormente, e neste caso serviu para a obtenção do nome comum (CN) dos certificados, para o *hash256* do ficheiro fornecido pelo utilizador, e para a validação da assinatura respetiva. O *header* que utiliza os métodos criptográficos é denominado de **crypto.h**. De seguida, são explicados os métodos mais relevantes.

std::map<string, string> subject(X509* x509)

Este método recebe um certificado em formado X509 e tem como função fazer o tratamento do *subject* deste.

std::string parseCN(X509* x509)

Utiliza o método anterior mas retira unicamente um argumento do mapa de *subject* dos certificados, sendo neste caso o nome comum (CN).

vector<unsigned char> sha256(const string str)

É a operação de hash 256 de uma string dada como argumento. Devolde no formato de array de *bytes*.

bool verifySignature(RSA* publicRSA, std::string plainText, char* signatureBase64)

Este método faz a verificação da assinatura recebendo como argumentos a chave pública RSA, a mensagem, e a assinatura em codificação base 64.

RSA* getPub(X509* x509)

Dando como argumento um certificado X509, devolve a chave pública no objeto RSA.

3.2.4 Métodos para as operações do serviço SCMD

Os métodos para as operações foram feitos com base no *curl*, sendo que obtivemos dificuldades na utilização do *gSoap* que era a primeira opção para a obtenção dos pedidos *soap*. Como não foi possível prosseguir o método principal, fomos pela melhor alternativa que seria aceder através do *curl* e do envio de uma mensagem de pedido em formato *XML*. Sendo assim no ficheiro **cmd_soap_msg.h** estão dispostas todas as operações numa classe pretendidas para o teste principal deste programa. Os métodos dispostos recebem os argumentos necessários

para se efetuarem os pedidos tais como o número de telemóvel, o pin e o nome do ficheiro. Para o identificador da aplicação foi necessário fazer uma codificação em base 64.

vector<string> getcertificates(string applicationId, string userId)

Método que recebe como argumentos o identificador da aplicação e do utilizador e faz um pedido *Soap* para o endereço de autenticação da AMA. Retorna um vetor que contém os certificados referentes ao utilizador.

vector<string> certccMovelSign(string applicationId, string userId, string pin, string docHash, string docName)

Método que recebe o identificador da aplicação, do utilizador, o pin, o hash de um documento introduzido e o nome deste. Desta vez, faz um pedido para receber o identificador de processo. É recebido um vetor com os valores do identificador de processo, e o código (caso seja 200 quer dizer que o pedido foi tratado com sucesso).

vector<string> validateotp(string applicationId, string otp, string processId)

Este método recebe o identificador da aplicação, o código OTP recebido via SMS para o número de telemóvel, e o identificador do processo que foi enviado pelo Servidor ao fazer-se o pedido do método anteriormente verificado (certcc-MovelSign). É retornado um vetor com a assinatura, o código e a mensagem relativamente ao código descrito.

std::string exec(const char* cmd)

Método que executa um comando pretendido, neste caso foi necessário para a utilização do *curl*, onde se fez uma comunicação entre processos (via *pipes*). Retorna o resultado em formato String.

3.3 Instalação e Configuração

- Sistema Operativo: Linux - Ubuntu 18.04.2 LTS
- Compilador: g++ 7.5.0
- Dependências de entrada: -lcrypto (OpenSSL), -std=gnu++11

Para compilar o programa foi efetuado o comando:

```
$ g++ -o test_cmd_wsdl test_cmd_wsdl.cpp -std=gnu++11 -lcrypto
```

A execução do programa pode ser efetuada das seguintes formas:

- Para ajuda/obtenção dos certificados:

```
$ ./test_cmd_wsdl gc -h
```

```
$ ./test_cmd_wsdl gc "+XXX NNNNNNNNN"
```

- Para a ajuda ou pedido de assinatura:

```
$ ./test_cmd_wsdl ms -h
```

```
$ ./test_cmd_wsdl ms "+XXX NNNNNNNNN" "PPPP"
```

```
$ ./test_cmd_wsdl ms "file_name" "+XXX NNNNNNNNN" "PPPP"
```

- Para a ajuda/validação do código OTP:

```
$ ./test_cmd_wsdl otp -h
```

```
$ ./test_cmd_wsdl otp "otp_code" "processId"
```

- Para o teste de todas as operações:

```
$ ./test_cmd_wsdl test -h
```

```
$ ./test_cmd_wsdl test "file_name" "+XXX NNNNNNNNN" "PPPP
    "
```

- Para saber a versão do CMD:

```
$ ./test_cmd_wsdl -V
```

3.4 Funcionamento do Sistema

O sistema está divido nas operações que comunicam com o *Soap*, sendo dividido nas fases seguintes:

- Uma primeira fase, em que se obtêm os certificados do utilizador, através do seu número de telemóvel. Os certificados devolvidos correspondem ao do utilizador, ao certificado de autoridade (CA), e o certificado de raíz (Root).
- Numa segunda fase, é pretendido fazer um pedido para a obtenção do identificador do processo, sendo dado o número de telemóvel e o pin do utilizador.

- Por fim, é feito um pedido para a obtenção do código OTP para a validação do utilizador que recebe um código via SMS para o número declarado.

O objetivo principal é obter a assinatura e a realização da sua verificação. As operações podem ser efetuadas de forma individual, caso o utilizador opte por chamar a aplicação apenas para um dos serviços disponíveis (GetCertificate, CCMovelSign, ValidateOtp). De igual forma, pode fazer recorrer a uma análise através da flag -h ou –help onde é disponibilizada toda a informação acerca do programa e de que forma o pode correr. A sua versão está indicada através da flag -V ou –version.

3.5 Resultados do Sistema

3.5.1 Menus de ajuda/versão

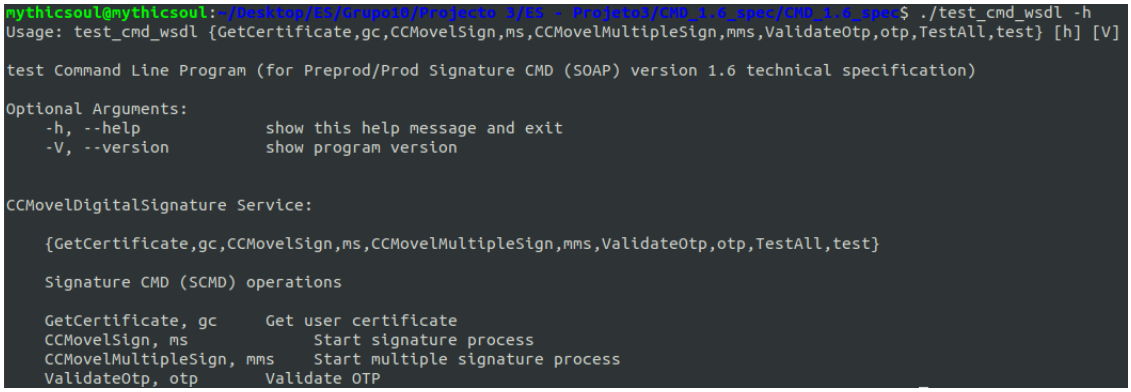

Figura 3.1: Menu de ajuda geral.

Figura 3.2: Versão do programa.

3.5.2 GetCertificate (gc)

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projecto 3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl gc -h
Usage: test_cmd_wsdl GetCertificate [-h] [-applicationId APPLICATIONID] [-prod] [-D] user

Get user certificate

Positional Arguments:
    user                    user phone number (+XXX NNNNNNNNN)

Optional Arguments:
    -h, --help              show this help message and exit
    -applicationId APPLICATIONID        CMD ApplicationId
    -prod                   Use production SCMD service (preproduction SCMD service used by default)
    -D, --debug             show debug information
```

Figura 3.3: Menu de ajuda - GetCertificate.

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projecto 3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl gc "+351 914307721"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  9824  100  9493  100   331  65020   2267 --:--:-- --:--:-- --:--:-- 67287
-----BEGIN CERTIFICATE-----
MIIJ+TCCB+GgAwIBAgIIB1ukz4xOgoQwDQYJKoZIhvcNAQELBQAwgekxCzAJBgNVBAYTAlBUMUIw
QAYDVQQKDDlBTUEgLSBBR8OKTkNJQSBQQVJBIEEgTU9ERVJOSVpBw4fDg08gQURNSU5JU1RSQVRJ
VkEgSS4gUC4xHDAaBgNVBAsME0NhcnTDo28gZGUgQ2lkYWTDo28xFDASBgNVBAsMC3N1YkVDRXN0
YWRvMWIwYAYDVQQDDFlFQyBkZSBDaGF2ZSBNw7N2ZWwgRGlnaXRhbCBkZSBBc3NpbmF0dXJhIERp
Z2l0YWwgUXVhbGlmaWNhZGEgZG8gQ2FydMOjbyBkZSBDaWRhZMOjbyAwMDAwMzAeFw0yMDA3MDUx
NzAxMzNaFw0yMjA5MjgyMTU5MDBaMIIBDTELMAkGA1UEBhMCUFQxHDAaBgNVBAoME0NhcnTDo28g
ZGUgQ2lkYWTDo28xQzBBBgNVBAsMOkNoYXZlIE3DsBCBEaWdpdGFsIGRlIEFzc2luYXR1cmEg
UXVhbGlmaWNhZGEgZG8gQ2lkYWTDo28xETAPBgNVBAsMCENpZGFkw6NvMR0wGwYDVQQLDBRSZW1v
dGVRU0NETWFuYWdlbWVudDEVMBMGA1UEBAwMTWFjaGFkbyBDcnV6MRcwFQYDVQQqDA5TYW5kcm8g
RW1hbnVlbDETMBEGA1UEBRMKQkkxNTMzNzM2OTEkMCIGA1UEAwwbU2FuZHJvIEVtYW51ZWwgTWFj
aGFkbyBDcnV6MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEAy4RCKT4vZXIlJ3vMr5Me
lhpj4HoxHPyFSSHrWnbsK4/ILdQss48m69k8snrAfB9cDWSWb6DS3/aaEfQau6HII3G6wTQDHSjh
S50hf1iMAa99QjLu/zhghxHCPpjF2C+FD3/Mec+SqY2wadKCkSk90bdlk2vy2CovKUGHvZ+Rckdu
VEAbyG9RQAZEQ3XiE9Ej5yNePkF4+tq46UKPIwAOKZuXtEZw3hqLhXQnfEcOPQ4sNS+o9/H1hGsZ
Zvq7G3q/8i4OJl824lE18luDsuL+BULjWPZOW7hRQ0AJpO8WPIjXPd6qPYGfcU5pUi2eMp+xQ1W/
FgnM6g4QKbNrxGW37UxeR0cZ64tW8t+kbuLwSgVdXqJCNaCbGBZEmE/OwGcqljKC8NqZZPwVOJJM
Dn/CG2Qt6WKuqAJI+4TmXdjDB9MXaiLHUczwfz7UUMRro/6Q4ZjyRSdvhKQ5pr/T3d7K28eL0VTk
+aioSEv+4JMZp4jDu+y6rUxHmfCdj/kfsq5jAgMBAAGjggP8MIID+DAMBgNVHRMBAf8EAjAAMB8G
A1UdIwQYMBaAFGrG5bN6vAZ07+eQppbYcMO3moOkMEsGCCsGAQUFBwEBBD8wPTA7BggrBgEFBQcw
AYYvaHR0cDovL29jc3AuY21kLmNhcnRhb2RlY2lkYWRhby5wdC9wdWJsaWNvL29jc3AwAYDVR0u
BGEwXzBdoFugWYZXaHR0cDovL3BraS5jYXJ0YW9kZWNpZGFkYW8ucHQvcHVibGljby9scmMvY2Nf
c3ViLWVjXZNpZGFkYW9fY21kX2NybDAwMDNfZGVsdGFfcDAwMDEuY3JsMIIBOwYDVR0gBIIBMjCC
AS4wNgYIYIRsAQEBAgowKjAoBggrBgEFBQcCARYcaHR0cDovL3d3dy5zY2VlLmdvdi5wdC9wY2Vy
dDAJBgcEAIvsQAECMG4GDGCEbAEBAQIEAwABATBeMFwGCCsGAQUFBwIBFlBodHRwOi8vcGtpLmNh
cnRhb2RlY2lkYWRhby5wdC9wdWJsaWNvL3BvbGl0aWNhcy9wYy9jY19zdWItZWNfY2lkYWRhb19j
bWRfcGMuaHRtbDBvBgtghGwBAQECBAMABzBgMF4GCCsGAQUFBwIBFlJodHRwOi8vcGtpLmNhcnRh
b2RlY2lkYWRhby5wdC9wdWJsaWNvL3BvbGl0aWNhcy9kcGMvY2Nfc3ViLWVjX2NpZGFkYW9fY21k
X2RwYy5odG1sMAgGBgQAj3oBAjAoBgNVHQkEITAfMB0GCCsGAQUFBwkBMREYDzE5OTcxMDI3MTIw
MDAwWjCCARIGCCsGAQUFBwEDBIIBBDCCAQAwCAYGBACORgEBMAgGBgQAjkYBBDBaBgcEAI5GAQYB
DE9DZXJ0aWZpY2F0ZSBmb3IgZWxlY3Ryb25pYyBzaWduYXR1cmVzIGFzIGRlZmluZWQgaW4gUmVn
dWxhdGlvbiAoRVUpIE5vIDkxMC8yMDE0MIGNBgYEAISGAQUwgYIwPxY5aHR0cHM6Ly9wa2kuY2Fy
dGFvZGVjaWRhZGFvLnB0L3B1YmxpY28vcG9saXRpY2FzL2Nwcy5odG1sEwJQVDA/FjlodHRwczov
L3BraS5jYXJ0YW9kZWNpZGFkYW8ucHQvcHVibGljby9wb2xpdGljYXMvY3BzLmh0bWwTAkVOMGIG
A1UdHwRbMFkwV6BVoFOGUWh0dHA6Ly9wa2kuY2FydGFvZGVjaWRhZGFvLnB0L3B1YmxpY28vbHJj
L2NjX3N1Yi1lY19jaWRhZGFvX2NtZF9jcmwwMDAzX3AwMDAxLmNybDAdBgNVHQ4EFgQUDlizB3+n
o4lkshtb/xVgtePTPWMwDgYDVR0PAQH/BAQDAgZAMA0GCSqGSIb3DQEBCwUAA4ICAQCMT9R3tb4s
zgtJNj3rX7WeCZWxDlTG6ooZYyk94mZe+Qm1Q+FNHcJL7JL9eAI0NicVE0zPz0+x0zG/Kpiz4v+w
hU5zyw3qspSClhiMeFvIv6gFNHhfqam0SxC1Zldfli/b6GS8OUNkxgb5RSNku/aZtlslk9VvOq0U
3M0vXyPQIO5T+wzfQmky2CaFehuE8kqUyHYi4UAdBGNnZUO/mcqsBK/9N2zXp9LY18LB8HlUmAeo
O8GcYRMPJ0kLGKPVDhJbpLEWTGVHz3nQMNByBOKxFBFeWBA2w20/x7q0ddizbsoMFMaGuEAbDLbn
jcZ8A1al0fzUtC1AD/JSYmyKg/QjHlB+dBEEaLiVMJkGiQjOYP0h4a2mn2E13HN3L5jPLYztAGEP
9nEnIwEN7Gaijq6dQgn7rj/dCsiglqzOMTwjEnCfDdfwzasr6SmNEFg6FoXofTFLLJyEh/+Kud1f
csQ3cFOkmRqUXnTejNTTBteI2kUOWLRk3JIbfho+sZyz3oQs+OFnVmxsVSw9HmNf5vJAawms8SX+
Y/PouIAdsG+1QwZqXfkTOFBrybZ6qcXiOuwAx1ZGG5/bq+uJbGU8n8qM0a6Go+S8lyIpOMTT4hbC
A8uKVNcx90o/IzmjMhzdSQmN9G1Zm1QB41LBimCYnn61CJ4gtgG5k6UgsHbe4hn0tA==
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIGuTCCBKGgAwIBAgIQMOKuZfn7OuJedMSBZDJHMzANBgkqhkiG9w0BAQsFADBiMQswCQYDVQQG
EwJQVDE4MDYGA1UECgwvU2lzdGVtYSBkZSBDZXJ0aWZpY2HDp8OjbyBFbGV0csOzbmljYSBkbyBF
```

Figura 3.4: Utilização da operação GetCertificate.

3.5.3 CCMovelSign (ms)

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projecto 3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl ms -h
Usage: test_cmd_wsdl CCMovelSign [-h] [-applicationId APPLICATIONID] [-prod] [-D] file user pin

Start signature process

Positional Arguments:
    file                  file input name (optional)
    user                  user phone number (+XXX NNNNNNNNN)
    pin                   CMD signature PIN

Optional Arguments:
    -h, --help            show this help message and exit
    -applicationId APPLICATIONID        CMD ApplicationId
    -prod                 Use production SCMD service (preproduction SCMD service used by default)
    -D, --debug           show debug information
```

Figura 3.5: Menu de ajuda - CCMovelSign.

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projecto 3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl ms LICENSE "+351 914307721" "0803"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1126  100   548  100   578   4281   4515 --:--:-- --:--:-- --:--:--  8796
ProcessID devolvido pela operação CCMovelSign: 8ac5b198-82ba-4107-b63c-0e47f2c095d8
```

Figura 3.6: Utilização da operação CCMovelSign.

3.5.4 ValidateOtp (otp)

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projecto 3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl otp -h
Usage: test_cmd_wsdl ValidateOtp [-h] [-applicationId APPLICATIONID] [-prod] [-D] OTP ProcessId

Validate OTP

Positional Arguments:
    OTP                   OTP received in your device
    ProcessId             ProcessID received in the answer of the CCMovelSign/CCMovelMultipleSign command

Optional Arguments:
    -h, --help            show this help message and exit
    -applicationId APPLICATIONID        CMD ApplicationId
    -prod                 Use production SCMD service (preproduction SCMD service used by default)
    -D, --debug           show debug information
```

Figura 3.7: Menu de ajuda - ValidateOtp.

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projecto 3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl otp "434815" "8ac5b198-82ba-4107-b63c-0e47f2c095d8"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1544  100  1154  100   390    544    183  0:00:02  0:00:02 --:--:--   727
O OTP foi validado!
```

Figura 3.8: Utilização da operação ValidateOtp.

3.5.5 TestAll (test)

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projecto 3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl test -h
Usage: test_cmd_wsdl TestAll [-h] [-applicationId APPLICATIONID] [-prod] [-D] file user pin

Automatically test all commands

Positional Arguments:
    file                    file
    user                    user phone number (+XXX NNNNNNNNN)
    pin                     CMD signature PIN

Optional Arguments:
    -h, --help              show this help message and exit
    -applicationId APPLICATIONID        CMD ApplicationId
    -prod                   Use production SCMD service (preproduction SCMD service used by default)
    -D, --debug             show debug information
```

Figura 3.9: Menu de ajuda - TestAll.

```
mythicsoul@mythicsoul:~/Desktop/ES/Grupo10/Projecto 3/ES - Projeto3/CMD_1.6_spec/CMD_1.6_spec$ ./test_cmd_wsdl test LICENSE "+351 914307721" "0803"
test Command Line Program (for Preprod/Prod Signature CMD (SOAP) version 1.6 technical specification)
version: 1.0

+++ Test All inicializado +++

 0% ... Leitura de argumentos da linha de comando - file: LICENSE user: +351 914307721 pin: 0803
10% ... A contactar servidor SOAP CMD para operação GetCertificate
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  9824  100  9493  100   331  81836   2853 --:--:-- --:--:-- --:--:-- 84689
20% ... Certificado emitido para "Sandro Emanuel Machado Cruz" pela Entidade de Certificação "EC de Chave Móvel Digital de Assinatura Digital Qualificada do Cartão de Cidadão 00003" n
a hierarquia do "Cartão de Cidadão 006"
30% ... Leitura do ficheiro LICENSE
40% ... Geração de hash do ficheiro LICENSE
50% ... Hash gerada (em base64): OXLcl0T2SZ8Pmy2/dmlvKuetivmyPd5m1q+Gyd+zaYY=
60% ... A contactar servidor SOAP CMD para operação CCMovelSign
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1126  100   548  100   578   4314   4551 --:--:-- --:--:-- --:--:--  8866
70% ... ProcessID devolvido pela operação CCMovelSign: 7eba6374-7865-4378-9453-a1a125dc4bf2
80% ... A iniciar operação ValidateOtp
Introduza o OTP recebido no seu dispositivo: 934783
90% ... A contactar servidor SOAP CMD para operação ValidateOtp
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1544  100  1154  100   390    537    181  0:00:02  0:00:02 --:--:--   719
100% ... Assinatura (em base 64) devolvida pela operação ValidateOtp: lZqAEhn6iG2dHShNPdfz2PjDgRe1dMrQeANYuGQcRRrGtVv7vlD+UBA9OnMdeQQpXtVBVJBWp3Myo0Y9MroTYTaQchjoh8BWGKsrRszFCp7b2+qsv
FYeeneODIUdvBVqY1epNRjnUgLxYuNeS7N4xsz2xpTn+QGxzEpyEkYdYO8/LDsdKSfu8Q5C44Cl5gCoWJ7DL1YFrdipjFhmWytGzhJ5yr4QdXP/UFtVrk1bdFcDUHilRNnKlZWHXQ+akYVuKua7XoUsulo8NZetr2HyxCddgFWBEBxwOzEAA4Ve
VwjbgySJSsM/fTSLwkIE9sO0kiblP18BBe6SloDZBswNKkhVABZ1b09G1Dv6MU+907t/072kjCHHkkh/qBy3vVSH209ngQQ1yGaPhnqgkQhMnsR3IKghXv8y4j3njzMHVP7AZ7mPKqwTRd7EjFaoy5+8BLMdknI18XRYBrlgkPOA9qD5Amu2mtD
8piaGAwerDfrXzoEGT5WtialvUU4bk4lH
110% ... A validar assinatura ...
Falha na verificação da assinatura

+++ Test All finalizado +++
```

Figura 3.10: Utilização da operação TestAll.

Capítulo 4

Considerações Finais

Devido a algumas dificuldades obtidas, não foi conseguida a verificação da assinatura, sendo que foi tentada de várias formas através do OpenSSL. As operações foram feitas com base em pedidos feitos pelo utilizador (Cliente) e as suas respostas. Relativamente ao método principal (test ou TestAll) foi realizado na sua completude à exceção da verificação que nunca deu como se pretendia, que era o sucesso desta. A chave pública que é obtida do certificado tem um tamanho de 384 bytes e corresponde a uma chave de 3072 bits.

Capítulo 5

Conclusões

Apesar de não termos conseguido obter um programa completamente funcional não estamos desapontados com os resultados obtidos, dadas as grandes dificuldades que tivemos nas comunicações com os *web services* em SOAP e as outras complicações que aparecem quando se tenta fazer *reverse engeneering* de um programa de uma linguagem para a outra linguagem que é quase completamente diferente, como por exemplo o problema crítico que tivemos com as chaves de Base 64.
Em geral foi um trabalho interessante, mas que pareceu que estávamos em desvantagem simplesmente por causa da linguagem em que tivemos de desenvolver este programa, pois deparámo-nos com faltas de bibliotecas ou de recursos online que nos pudessem ajudar ou informar em como resolver alguns dos problemas que encontramos ao longo deste projeto e, se tivéssemos a utilizar outra linguagem estes problemas não eram tão significativos ou até não existiam, mas a extensão da data de entrega foi uma boa ajuda para tentar resolver os problemas encontrados.

Referências

[1] "O Que é a Chave Móvel Digital?" A Chave Móvel Digital, www.autenticacao.gov.pt/a-chave-movel-digital.

[2] "SOAP." Wikipedia, Wikimedia Foundation, 12 July 2019, pt.wikipedia.org/wiki/SOAP.

[3] "OpenSSL." Wikipedia, Wikimedia Foundation, 11 May 2019, pt.wikipedia.org/wiki/OpenSSL.

[4] "Base64." Wikipedia, Wikimedia Foundation, 3 July 2020, pt.wikipedia.org/wiki/Base64.

[5] "Python vs C : Find Out The 9 Essential Differences." EDUCBA, 24 Apr. 2020, www.educba.com/python-vs-c-plus-plus/.