INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

SEGURANÇA INFORMÁTICA

LICENCIATURA ENGENHARIA INFORMÁTICA E COMPUTADORES

FASE DE EXERCÍCIOS
Fase 1

Autores:
43552 - Samuel Costa
43320 - André Mendes

Docente:
José Simão

21 de Outubro de 2019

Conteúdo

1 Introdução

O trabalho realizado para este fase pretende que os temas desenvolvidos durante as aulas sejam postos em prática. Para esta fase os exercícios focaram essencialmente a primeira parte da matéria.

- Esquemas e Primitivas Criptográficas
- Java Cryptographic Architecture (JCA)
- Certificados digitais e Infraestrutura de Chave Pública
- Protocolo Criptográfico Transport Layer Security (SSL/TLS)

Neste trabalho prático pretendemos responder aos vários exercícios propostos e implementar uma demonstração dos pontos referidos.

2 Primeira série de exercícios

2.1 Exercício 1

Ambos os esquemas criptográficos visam garantir a autenticidade da mensagem adicionando à própria mensagem um numero de bits de forma a identificar o emissor da mesma, no caso do MAC uma marca ou uma assinatura no caso da assinatura digital, sendo essa autenticidade verificada no recetor. Ambos os esquemas utilizam chaves privadas para "marcar" a mensagem, sendo que no caso da assinatura digital é usado uma chave publica para autenticar essa mesma mensagem, já no esquema MAC é usada uma chave simétrica.

2.2 Exercício 2

A organização da função de hash reduz uma mensagem de tamanho variável m a um bloco de tamanho n ou seja, a dimensão de yL = dimensão de $m1$, logo n bits = dimensão de um bloco «< dimensão da mensagem, pelo que existe m e m', tal que $H(m) = H(m')$. O diagrama seguinte pretende ilustrar a organização da aplicação da primitiva de cifra, conforme explicitado:

2.3 Exercício 3

Apesar de não ter acesso ao texto em claro, o que facilitaria o ataque, o atacante tem acesso ao texto cifrado. Sabendo que os mesmos blocos de texto em claro produzem as mesmas cifras, então um oráculo de encriptação pode ser usado. Se o atacante pode submeter texto em claro para encriptação, então também pode verificar hipóteses sobre o texto em claro que corresponde ao texto cifrado.

2.4 Exercício 4

Comparando os dois esquemas seria expectável que a primitiva AES fosse em todos os casos mais dificil de criptoanalizar que um sistema com primitiva DES, no entanto essa analise tem de ser feita também relativamente ao modo de operação que cada uma pode usar, como por exemplo no caso especifico de um sistema AES ter usado um modo de operação ECB (Electronic Code Book) e a sua mensagem ser de alguma forma curta e padronizada pode mais facilmente ser quebrada que uma DES que usou o mode de operação CBC (Cipher Block Chain).

2.5 Exercício 5

A JCA contem uma arquitectura baseada em providers e uma série de APIs para assinaturas digitais, hashes, certificados e a sua validação, encriptação, geração e gestão de chaves. Essas APIs permitem integrar segurança facilmente no código. Uma engine class fornece a interface para um serviço criptográfico especifico, ficando independente de um algoritmo criptografico particular ou de um provider. As engine classes fornecem operações criptograficas ou geradores ou conversores de material criptografico, ou objectos que encapsulam dados criptograficos e podem ser usados em niveis mais elevados de abstração. Esta opção de arquitectura privilegia a independencia de implementação e a interoperabilidade, já que as aplicações não precisam de implementar algoritmos de segurança e os providers são interoperáveis em relação às aplicações.

2.6 Exercício 6.1

Um certificado é composto pelo seu corpo (campos, extensões e propriedades) e por uma assinatura. A verificação da assinatura permite atestar da autenticidade da mensagem. Como o processo de verificação da autenticidade calcula um hash do corpo do certificado, não deve ser possível que mensagens diferentes produzam o mesmo hash (2^a pré-imagem), o que significaria que um certificado podia ser alterado no seu corpo (por exemplo, no seu emissor ou chave pública) e a verificação da assinatura retornar true.

2.7 Exercício 6.2

Para validar a assinatura do certificado C são usados o seu corpo, a assinatura presentes no certificado e a chave pública do emissor. Para validar os outros certificados intermédios na cadeia de certificação são usados os mesmos elementos. Para validar a assinatura do certificado raíz, é usada a sua chave pública, uma vez que nesse caso o certificado é assinado pelo emissor (auto-assinado). Portanto, nenhuma chave privada é usada para na validação de certificados X.509.

2.8 Exercício 7

Para este exercício foram realizadas experiências segundo o *Labs for Security Eduction*.
Na alnea 1 do ponto *"2.4 Task 4: Padding"* foram testadas os modos propostos e é possivel concluir que ECB e CBF, ambos usam *padding*, ao contrário dos modos CFB e OFB.
Para determinar o uso de *padding* nos varios modos foi usado um ficheiro de 5 *bytes* e depois foi feita a sua decifra usando o comando *nopad* e também um *hexdump* para verificar o seu conteúdo. Foi também possível verificar que o tamanho do ficheiro no caso dos modos que usam *padding* era sempre uma potência de base 2.

Na alínea 2 exercício usando a *-aes-128-cbc* para encriptar os 3 ficheiros propostos foi possível reparar que os ficheiros todos os ficheiros tiveram *padding*. O de 5 e o de 10 *bytes* tiveram um preenchimento de *padding* até aos 16 *bytes*, sendo que o de 16 ao ser cifrado obtinha o seu preenchimento completo então levou mais 16 *bytes* de *padding*.
Os valores de preenchimento pode ser verificados com o *hexdump* dos três ficheiros.

```
D:\Isel\1920i\Segurança Informatica\isel-leic-si-series\Serie1\Entrega\Lab\Alinea2>hexdump -C f1decipher_pad.txt
000000  31 32 33 34 35 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b 0b  12345...........

D:\Isel\1920i\Segurança Informatica\isel-leic-si-series\Serie1\Entrega\Lab\Alinea2>hexdump -C f2decipher_pad.txt
000000  31 32 33 34 35 36 37 38 39 61 06 06 06 06 06 06  123456789a......

D:\Isel\1920i\Segurança Informatica\isel-leic-si-series\Serie1\Entrega\Lab\Alinea2>hexdump -C f3decipher_pad.txt
000000  30 31 32 33 34 35 36 37 38 39 61 62 63 64 65 66  0123456789abcdef
000010  10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10  ................
```

Figura 1: hexdump on files

2.9 Exercício 8.1

A função de decifra e verificação é dada por: T(k2)(D(k1)(c)||t). Ou seja, aplicando a primitiva de decifra ao criptograma, e de seguida verificando a assinatura contra a mensagem e a marca.

2.10 Exercício 8.2

Foi implementada a aplicação Mac-then-encrypt usando a JCA.

2.11 Exercício 9

A fazer ...

3 Conclusão