Instituto Superior de Engenharia de Lisboa
LEIC, LEIRT, LEIM
Segurança Informática
Primeira série de exercícios, Semestre de Inverno de 20/21
Entregar até 11 de novembro de 2020

1. No contexto dos esquemas de cifra simétrica, apresente duas vantagens do modo de operação CTR (*counter mode*) em relação ao modo CBC.

2. Considere um novo esquema criptográfico AE. O objectivo é fazer uma cifra simétrica com garantias de integridade, ou seja, caso os criptogramas sejam modificados no canal de comunicação, tal seria detetado pelo destinatário.

   As funções AE_e e AE_d realizam a cifra e decifra autenticada, sendo E uma primitiva de cifra simétrica, H uma função de *hash* criptográfica e $||$ a concatenação de *bytes*.

   $AE_e(k)(m) = E(k)(m) \;||\; H(E(k)(m))$

   $AE_d(k)(c,h) = (\textbf{se } H(c) == h \textbf{ então } m = D(k)(c) \textbf{ senão} \text{ falha de integridade})$

   Note que a função de decifra opera sobre criptogramas (c) e o valor de hash (h) que foram colocados no canal de comunicação pela função de cifra.

   Descreva de que forma pode ser comprometida a propriedade de integridade do esquema.

3. Na biblioteca JCA, como é que as *engine classes* (ex: Cipher, Signature, Mac) possibilitam a aplicação incremental das respetivas proteções? Qual a vantagem de aplicar proteções incrementalmente?

4. Considere os certificados digitais X.509 e as infra-estruturas de chave pública:

   4.1. A assinatura de um certificado folha tem em conta toda a cadeia de certificados?

   4.2. Existem campos num certificado que estejam protegidos por um esquema de cifra (simétrica ou assimétrica)?

   4.3. Considere dois sistemas informáticos cujas comunicações estão cifradas usando cifra assimétrica, após troca de chaves públicas em certificados X.509. A realização de um ataque de *man-in-the-middle* implica conseguir alterar algo do lado cliente?

5. Considere o enunciado do laboratório "Crypto Lab" disponível em `https://seedsecuritylabs.org/Labs_16.04/Crypto/Crypto_Encryption/`. Realize a tarefa "Task 4: Padding". Descreva sucintamente os resultados.

6. Usando a JCA, desenvolva uma aplicação para cifrar ou decifrar um ficheiro, usando um esquema simétrico. A aplicação deve receber na linha de comandos o ficheiro (em claro ou cifrado) e a indicação do tipo de proteção a realizar (cifra ou decifra).

   Quando executada para cifra, a aplicação gera a chave e guarda-a em ficheiro próprio. Valoriza-se o uso de `Keystore` para armazenar a chave simétrica. Assuma que a *password* é transportada num canal seguro.

7. Realize uma aplicação de consola para assinar e verificar objectos *JSON Web Token* (JWT) [1], transportados numa estrutura *JSON Web Signature* (JWS) [2]. O site `https://jwt.io/` pode ser usado para produzir ou validar objectos JWS.

   A aplicação deve, no mínimo, suportar assinatura digital com os algoritmos "RS256" e "HS256". Todos os comandos da aplicação devem ser indicados na linha de comandos. Use a biblioteca Apache Commons [3] para realizar o base64URL *encoding*/*decoding* necessário à criação e verificação da assinatura. Para assinar/verificar JWS com assinatura RS256 use o material criptográfico (chaves e certificados) fornecidos em anexo.

   Valoriza-se que a aplicação valide o certificado usado para verificar a assinatura digital.

14 de outubro de 2020 (editado a 2 de novembro de 2020)

Referências

[1] `https://tools.ietf.org/html/draft-ietf-oauth-json-web-token-27`

[2] `https://tools.ietf.org/html/draft-ietf-jose-json-web-signature-33`

[3] `https://commons.apache.org/`