Instituto Superior de Engenharia de Lisboa

Licenciatura/Mestrado em Engenharia Informática e de Computadores

Segurança Informática

Primeira série de exercícios, Semestre de Inverno de 16/17

Data de entrega: 9 de Novembro de 2016

---

1. Qual a motivação para os esquemas MAC (*Message Authentication Code*), dado que os esquemas de cifra simétrica já fornecem confidencialidade?

2. No contexto das funções de *hash*, o que é a propriedade da resistência ao cálculo da segunda pré-imagem (falsificação selectiva)? Forneça um exemplo de utilização onde esta propriedade é usada.

3. Quando o espaço de mensagens a cifrar é muito pequeno (e.g. PINs de 4 dígitos), é fundamental que o esquema assimétrico de cifra utilizado tenha a função E não determinística. Justifique.

4. Com o objectivo de eliminar o problema do modo ECB quanto à passagem de padrões do texto em claro para o texto cifrado, foi definido o seguinte modo de operação:

   - $m = m_1, \ldots, m_L$ a divisão da mensagem m nos blocos m_i.
   - R é um número aleatório.
   - $c_0 = R, c_i = E(k)(m_i \oplus R)$, para $i = 1, \ldots, L$, onde $\oplus$ denota a operação de ou-exclusivo bit a bit e E uma primitiva de cifra em bloco.

   Este modo de operação cumpre o objectivo?

5. Na biblioteca JCA, porque razão a classe `MAC` não precisa de ter um método `verify`, semelhante ao existente na classe `Signature`?

6. Considere a infra-estrutura de certificados X.509.

   6.1. Foi atribuído à *Alice* o certificado C_{Alice} e a respectiva chave privada. Justifique se a *Alice* consegue ou não alterar o campo com a data de validade do certificado, mantendo a sua validade?

   6.2. Quais as consequências se uma aplicação consumidora de certificados ignorar a extensão *basic constraints*?

   6.3. Considere que o servidor S usa a CA_0 para a emissão dos seus certificados. Em que situações o ataque a uma autoridade de certificação diferente de CA_0 pode criar as condições para ataques de *Man In The Middle* entre clientes e S.

   6.4. Qual a diferença entre um ficheiro `.pfx` e um ficheiro `.cer`?

7. Realize uma aplicação de consola para proteger *JSON Web Tokens* (JWT) [3] usando a biblioteca JCA [1]. A proteção consiste no uso de cifra/decifra autenticada, com transporte de chaves usando criptografia assimétrica, tal como exemplificado na secção 3.3 do RFC 7516 [4].

   Sugere-se a seguinte abordagem:

   - Realize e teste um componente para cifrar e decifrar *strings* arbitrárias usando o algoritmo AES em modo GCM [1, 2].
   - Realize e teste um componente para cifrar e decifrar uma chave AES usando o algoritmo RSA.
   - Realize e teste um componente para obter a chave pública de um certificado X.509 validado e uma chave privada de um *keystore*.
   - Use os componentes anteriores para realizar e testar o sistema pretendido.

   Use o material criptográficos presente nos certificados e *keystores* do anexo `certificates.zip`.

Referências

[1] https://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html, visitado 8 outubro de 2016.

[2] http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf, visitado 8 outubro de 2016.

[3] https://tools.ietf.org/html/rfc7519, visitado 8 outubro de 2016.

[4] https://tools.ietf.org/html/rfc7516, visitado 8 outubro 2016.

ISEL, 11 de Outubro de 2016.