Instituto Superior de Engenharia de Lisboa

Licenciatura/Mestrado em Engenharia Informática e de Computadores

Segurança Informática

Primeira série de exercícios, Semestre de Inverno de 17/18

Data de entrega: 18 de outubro de 2017

---

1. Considere o esquema CI para confidencialidade e autenticidade de mensagens, onde $||$ representa a concatenação de bits e $X_{1..L}$ representa os primeiros L bits de X.

$$CI(m) = E_s(T(k_1)(m)_{1..L})(m)||T(k_1)(m)$$

   $E_s(k)(m)$ é um esquema simétrico de cifra e $T(k)(m)$ é um esquema de *message authentication code* (MAC). Porque motivo este esquema não cumpre os objectivos?

2. Num esquema de assinatura digital, qual o papel da função de *hash* e o da primitiva de assinatura?

3. Seja $E(k)(m)$ um esquema de cifra simétrico, que cifra m com a chave k, porque motivo o resultado de $E(k)(m_1||m_2)$ é diferente de $E(k)(m_1)||E(k)(m_2)$, sendo $||$ a concatenação?

4. Porque motivo o modo de operação *Galois Counter Mode* [1] não é vulnerável a ataques de *Vaudenay*?

5. Considere a infra-estrutura de certificados X.509 e a biblioteca JCA.

   5.1. Considere o certificado folha C e os intermédios $I_1, I_2, \ldots, I_n$. Alguma das chaves privadas dos certificados intermédios é usada para validar o certificado C?

   5.2. Que protecção tem a chave pública presente num certificado?

   5.3. Porque motivo é necessário garantir a integridade de um *keystore* que tenha apenas certificados auto-assinados?

6. Seja $h_k : \{0,1\}^* \rightarrow \{0,1\}^k$ a função de *hash* definida por: $h_k(x) = y_1 \ldots y_k$, onde $y_1 \ldots y_{160} = \mathbf{SHA1}(x)$.

   Sejam m_1 e m_2 os programas Java definidos nos ficheiros `BadApp.java` e `GoodApp.java` (presentes em anexo ao enunciado). Dois programas m e m' dizem-se equivalentes ($m \equiv m'$) se a sua execução produz o mesmo resultado observável.

   6.1. Calcule $h_k(m_1)$ e $h_k(m_2)$ para $k = 8, 16, 32$.

   6.2. Realize uma aplicação para encontrar um programa m' tal que $h_k(m') = h_k(m_2)$ e $m' \equiv m_1$. Considere $k = 8, 12, 16$. Realize 5 execuções da aplicação e apresente o número médio de operações h_k necessário para encontrar a colisão.

   6.3. Realize uma aplicação para encontrar um par (m'_1, m'_2) tal que $h_k(m'_1) = h_k(m'_2), m'_1 \equiv m_1$ e $m'_2 \equiv m_2$. Considere $k = 8, 16, 32$. Realize 5 execuções da aplicação e apresente o número médio de operações h_k necessário para encontrar a colisão.

7. Realize uma aplicação de consola para cifrar e decifrar ficheiros usando um esquema híbrido. Este tipo de esquema usa cifra assimétrica para transportar uma chave simétrica (gerada pela aplicação) que cifra o conteúdo do ficheiro. Independentemente da operação a realizar, a aplicação recebe como *input*: i) nome de ficheiro (com mensagem em claro ou cifrada); ii) a operação a realizar (`cifra` ou `decifra`).

   No modo `cifra` recebe: i) Certificado do destinatário; ii) *Keystore* para validar o certificado. Produz: i) Ficheiro com mensagem cifrada (C_f); ii) Ficheiro com metadados (IV e chave simétrica cifrada com a chave pública do destinatário). No modo `decifra` recebe: i) C_f; ii) metadados; iii) Chave privada do destinatário (ficheiro .pfx). Produz ficheiro com o texto em claro.

   Apresente tempos de execução para cifrar e decifrar o PDF do enunciado usando os algoritmos simétricos Blowfish e AES em combinação com o algoritmo assimétrico RSA [3]. Use o material criptográfico presente no anexo `certificates-keys.zip`.

Referências

[1] `https://en.wikipedia.org/wiki/Galois/Counter_Mode`, visitado em 7 de setembro de 2017

[2] `https://commons.apache.org/proper/commons-cli/`, visitado em 12 de setembro de 2017

[3] `http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#Cipher`, visitado em 10 de setembro de 2017

ISEL, 19 de setembro de 2017.
(editado a 23 de setembro de 2017)