Segurança de Sistemas – RSA – SHA – AES – Assinatura Digital
Prof. Avelino Zorzo – Escola Politécnica/PUCRS

Este trabalho é dividido em 2 partes. Cada uma destas partes é uma simplificação de algum dos assuntos vistos em aula. Elas não devem ser utilizadas para qualquer fim profissional, tem somente o intuito de demonstrar como alguns algoritmos funcionam de maneira simplificada. Para uso profissional, usar as recomendações ou padrões definidos publicamente. Valores em hexadecimal, representam os valores dos bytes, ou seja, 41 representa o caractere ASCII 'A' ou o valor decimal 65. **Este exemplo não é seguro** contra, por exemplo, "*man-in-the-middle attack*".

Usar a chave pública do professor $pk_p = (e_p, N_p)$ onde:

e_p = 2E76A0094D4CEE0AC516CA162973C895 (em hexadecimal)
N_p =
1985008F25A025097712D26B5A322982B6EBAFA5826B6EDA3B91F78B7BD63981382581218D33A9983E4E14D4B26113AA2A83BBCCF
DE24310AEE3362B6100D06CC1EA429018A0FF3614C077F59DE55AADF449AF01E42ED6545127DC1A97954B89729249C6060BA4BD3A5
9490839072929C0304B2D7CBBA368AEBC4878A6F0DA3FE58CECDA638A506C723BDCBAB8C355F83C0839BF1457A3B6B89307D672BB
F530C93F022E693116FE4A5703A665C6010B5192F6D1FAB64B5795876B2164C86ABD7650AEDAF5B6AFCAC0438437BB3BDF5399D80F
8D9963B5414EAFBFA1AA2DD0D24988ACECA8D50047E5A78082295A987369A67D3E54FFB7996CBE2C5EAD794391 (em hexadecimal)

Cuidado: se algum valor hexadecimal começar com {9, A, B, C, D, E, F}, isto significa que o bit mais significativo é 1 e indica um número negativo em algumas linguagens. Se acontecer, incluir um byte 0 na frente.

PARTE 1

Receber a chave pública do professor

Gerar dois números primos p e q com no mínimo 1024 bits
Calcular $N_a = p.q$
Calcular $L = (p-1).(q-1)$ → função φ de Euler
Encontrar um e_a que seja primo relativo de L, ou seja, MDC(e_a, L) = 1
Calcular o inverso d_a de e_a em Z_L, ou seja, $d_a \,.\, e_a = 1$ em Z_L
Guardar a chave pública $pk_a = (e_a, N_a)$ e a chave privada $sk_a = (d_a, N_a)$

Gerar chaves assimétricas

Escolher um valor aleatório s de 128 bit → chave a ser usada no AES
Guardar s
Calcular $x = s^{ep} \mod N_p$ → cifra a chave usando a chave pública do professor
Calcular $sig_x = x^{da} \mod N_a$ → assina a mensagem usando a chave privada do aluno

Gerar chave simétrica
Cifrar chave simétrica
Assinar texto cifrado

Enviar (x, sig_x, pk_a) para o professor por email ou whatsapp → todos os valores em hexadecimal

Envio

pk_p

x, sig_x, pk_a

$p, q, N=p.q; L = (p-1).(q-1)$

$sk_a = (e_a, N_a)$
$pk_a = (d_a, N_a)$

$s \leftarrow$ 128 bits aleatórios

$x = s^{ep} \mod N_p$

$sig_x = x^{da} \mod N_p$

$p, q, N=p.q; L = (p-1).(q-1)$

$sk_p \rightarrow (d_p, N_p)$
$pk_p \rightarrow (e_p, N_p)$

PARTE 2

Receber do professor:

- uma mensagem *c* cifrada → em hexadecimal com IV como 16 primeiros bytes
- uma assinatura sig_c para a mensagem c → em hexadecimal

Calcular h_c = SHA256(*c*)
Verificar se $h_c = sig_c^{ep} \bmod N_p$
Se sim, decifrar a mensagem *c* com AES (chave *s*, CBC, PKCS), tendo $m = AES^{-1}(c, s)$.
Se não, enviar mensagem de erro para o professor por email ou whatsapp.

Verificar assinatura
Decifrar mensagem

Inverter a mensagem *m* decifrada gerando m_{inv}. Exemplo, se *m* = "pucrs", então m_{inv} "srcup"
Gerar um IV aleatório
Cifrar m_{inv} usando AES (chave *s*, CBC, PKCS), tendo c_{inv} = concatenar(IV, AES(m_{inv}, *s*))

Calcular h_{inv} = SHA256(c_{inv})
Calcular $sig_{hinv} = h_{inv}^{da} \bmod N_a$

Processar mensagem
Cifrar mensagem
Assinar mensagem cifrada

Enviar (c_{inv}, sig_{hinv}) para o professor→ todos os valores em hexadecimal.

Envio

c, sig_c

c_{inv}, sig_{inv}

Se SHA(c) = $sig_c^{ep} \bmod N_p$
Então
m = AES-^{1}(c, s)

m_{inv} = inverter m
c_{inv} = AES(m_{inv}, s)

sig_c =SHA(c_{inv})$^{da} \bmod N_a$

Senão ERRO

c = AES(m, s)

sig_c =SHA(c)$^{dp} \bmod N_p$

Se SHA(c_{inv}) = $sig_{inv}^{ea} \bmod N_a$
Então
m_{inv} = AES-^{1}(c_{inv}, s)
TERMINOU
Senão ERRO

Submeter o código, com cada parte bem definida, e com todo o exemplo (valores) no início como comentário.