Universidade da Beira Interior
Faculdade de Engenharia
Departamento de Informática

Segurança Informática
Guia para Aula Laboratorial 8
Licenciatura em Engenharia Informática
Licenciatura em Informática Web
Licenciatura em Tecnologias e Sistemas da Informação

Computer Security
Guide for Laboratory Class 8
Degree in Computer Science and Engineering
Degree in Web Informatics
Degree in Information Technologies and Systems

Sumário
Análise da função alçapão RSA e do procedimento para geração de pares de chaves. Elaboração passo-a-passo de uma assinatura digital usando a função alçapão RSA e a função resumo SHA256.

Summary
Analysis of the RSA trapdoor function and of the key pair generation procedure. Step-by-step elaboration of a digital signature using the RSA trapdoor function and the SHA256 hash function.

Pré-requisitos:
Algumas das tarefas propostas a seguir requerem o uso de *software* para efetuar cálculos, o acesso a um sistema com interpretador de programas escritos em linguagem de programação Python e que disponibilize a ferramenta OpenSSL. Sugere-se, assim, o uso de uma distribuição comum de Linux, onde todas estas condições estarão provavelmente preenchidas.

1 Função Alçapão RSA
RSA Trapdoor Function

A utilização da função alçapão RSA pressupõe a utilização de um algoritmo para geração de um par de chaves, normalmente designadas por chave pública (pk) e chave privada (sk). A geração é feita da seguinte forma:

1. Geram-se dois números primos p e q grandes (tipicamente com mais do que 1024 bits) e calcula-se $N = p \times q$;
2. Calcula-se $\phi(N) = (p-1) \times (q-1)$ e procura-se um par de números (e, d) tal que
$$e \times d = 1 \bmod \phi(N).$$
3. A chave pública é o par (e, N) e a chave privada é d.

Tarefa 1 *Task 1*

Escolha 2 números primos entre 12 e 42 e calcule N e $\phi(N)$:

☐ !!Não consigo fazer isto!!
☑ $p =$ 31 $q =$ 23 $N =$ 713 $\phi(N) =$ 660

Use o *libreoffice spreedsheet* ou Python para encontrar dois números e e d que satisfaçam

$$e \times d = 1 \bmod \phi(N),$$

i.e., $e \times d \bmod \phi(N)$ tem de dar 1.
☐ ***Red Alert!*** Não consigo fazer isto!
☑ Isto é canja de galinha: $e =$ 7 $d =$ 283

Depois de fazer o exercício anterior, preencha os espaços em baixo com a chave pública e com a chave privada:
☑ $pk =$ (7,713) $sk =$ (283)

Tarefa 2 *Task 2*

O algoritmo de cifra **em modo livro da escola** (que não se deve utilizar diretamente) é dado por

$$E(pk, m) = m^{pk} \bmod N,$$

enquanto que o algoritmo de decifra é dado por

$$E(sk, c) = c^{sk} \bmod N.$$

Dê a sua chave pública a um(a) colega e peça-lhe que cifre um número **aleatório** qualquer entre 1 e

221. Peça-lhe também que lhe transmita o criptograma oralmente. **Sugestão: use o *wolfram alpha* para o ajudar a fazer estes cálculos. Q1.: Precisou de algum cuidado especial quando transmitia a chave pública?**
☐ Sim, tive de lhe dizer baixinho ao ouvido.
☐ Não, mas devia tê-la cifrado antes de a transmitir.
☑ Não, e não há qualquer problema de segurança nisso.

Q2.: Como é que o seu(ua) colega sabe que a chave pública que vai usar para cifrar a mensagem que está prestes a enviar-lhe é sua?
☑ Porque eu lha dei diretamente.
☐ Ele(a) nunca vai ter essa certeza.

Q3.: E se lhe tivesse enviado a chave pública pela Internet, respondia à questão anterior da mesma forma?
☑ Hummm.... não.
☐ Hummm.... sim.
☐ Respondia que sim, mas com alguma hesitação.

Tarefa 3 *Task 3*

Decifre o criptograma que recebeu do(a) colega. Escreva a fórmula que utilizou e o resultado:

Sugestão: use o *wolfram alpha* para o ajudar a fazer estes cálculos.

Q4.: Qual a chave usada para cifrar?
☑ A chave pública. ☐ A chave privada.
☐ A chave de sessão. ☐ A chave do carro.

Q5.: Qual a chave usada para decifrar?
☐ A chave pública. ☑ A chave privada.
☐ A chave de sessão. ☐ A chave do carro.

Tarefa 4 *Task 4*

O *OpenSSL* permite gerar pares de chaves RSA para um ou mais ficheiros. Indique e comente os comandos que pode usar para este efeito, bem como o formato dos ficheiros assim criados:

- Comando para gerar chaves RSA para o ficheiro `sk-and-pk.pem`.

  `$ openssl` genrsa -out sk-and-pk.pem 1024.

- Comando para extrair a chave pública do ficheiro `sk-and-pk.pem` para o ficheiro `pk-nome-aluno.pem`.

  `$ openssl` rsa -in sk-and-pk.pem -pubout -out pk-nome-aluno.pem.

- Comando para ver a chave pública no ecrã.

  `$ openssl` openssl rsa -in pk-pedro.pem -pubin -text.

- Comando para gerar as chaves, mas protegidas por cifra simétrica (esta tarefa é opcional na aula).

  `$ openssl` _____.

Nota: guarde as chaves que gerou numa *pen* ou envie-as para o seu *e-mail*, pois podem vir a ser úteis mais tarde.

Tarefa 5 *Task 5*

Gere um valor aleatório com 128 bits (16 bytes) em hexadecimal e coloque-o no ficheiro de texto `secret.key`. Peça a um(a) colega que lhe envie a sua chave pública e procure construir o comando *OpenSSL* que lhe permita cifrar esse ficheiro com a chave pública que recebeu, a partir da seguinte sugestão:

`$ openssl rsautl` -encrypt -in secret.key -out secret.rsa -inkey pk-pedro.pem -pubin

Q6.: Conseguiu cifrar?
☐ Claro que consegui.
☑ Neste caso consegui, porque 128 bits é menor do que o tamanho do módulo que estou a usar.
☐ Não, não consegui.

No final, envie o criptograma resultante de novo para o(a) colega.

Tarefa 6 *Task 6*

Engendre e emita o comando que lhe permite decifrar o criptograma recebido do colega. No final, verifique se o que decifrou é igual ao que foi realmente cifrado, conversando com o(a) colega.

Comando: `$ openssl rsautl` -decrypt -in secret.rsa -out secret.txt -inkey skandpk.pem

Tarefa 7 ***Task 7***

Q7.: Qual o tamanho do módulo do par de chaves que gerou antes?

☐ 123 ☐ 256 ☐ 512 ☑ 1024 ☐ 1025 ☐ 2048

openssl rsa -in sk-and-pk.pem -text

Volte a repetir a tarefa na secção **??** mas, desta feita, tente cifrar um ficheiro chamado `Portugal.txt` contendo **o segundo parágrafo** do texto em `http://en.wikipedia.org/wiki/Economic_history_of_Portugal`. Se tiver tempo, aproveite para ler o parágrafo...

Q8.: Conseguiu cifrar o ficheiro?

☐ Não. Portugal é grande demais para poder ser cifrado com esta cifra.

☑ Não. O texto é grande demais e o *OpenSSL* não deixa cifrar por blocos usando o RSA, visto que este modo (livro de escola), por sí só, já é inseguro.

☐ Consegui sem problemas.

Note que, para além da razão apontada anteriormente, deve levar em conta de que **a criptografia assimétrica é tipicamente mais lenta ou acarreta mais custos computacionais** do que a criptografia simétrica.

2 Cifra à Moda do Port... do PGP

Encryption a la PGP

O PGP será alvo de estudo um pouco mais à frente, mas nesta secção vai-se explorar um pouco o seu funcionamento.

Tarefa 8 ***Task 8***

Q9.: O que significa o acrónimo PGP?

P retty G ood P rivacy

Cifre o ficheiro com um algoritmo de cifra simétrica (e.g., o AES) e chave de cifra à sua escolha. Reveja as tarefas feitas anteriormente, nomeadamente a que se refere à troca de um segredo de 16 bytes usando RSA. Engendre um modo de usar a criptografia de chave pública RSA para colmatar o problema de trocar a chave de cifra simétrica. Se necessário, discuta isto com o Professor.

Cifragem:

`$ openssl`

`$ openssl`

Decifragem:

`$ openssl`

`$ openssl`

`$ cat`

3 Assinatura Digital

Digital Signature

Q10.: Ainda tem o par de chaves pública e privada gerada anteriormente?

☐ Ups...

☐ Não cheguei a fazer isto!

☑ Sou um(a) fulano(moça) certinho(a), e tenho tudo guardadinho.

Recupere o ficheiro `Portugal.txt`. Como é um ficheiro com informação que não deve ser alterada e com a qual o Prof. pressupõe que concorda plenamente, convém assiná-lo digitalmente. Para isso, simule o algoritmo conhecido por `SHA256-with-RSA`:

1. Calcule o valor do SHA256 do ficheiro `Portugal.txt` para o ficheiro `Portugal.sha256`.
2. Cifre o ficheiro `Portugal.sha256` com a chave privada e guarde o criptograma em `Portugal.sig`.

Tarefa 9 ***Task 9***

Envie a assinatura e o ficheiro original para o(a) seu(ua) colega. Quando receber o respetivo par, verifique a assinatura. Quais os três comandos que deve usar para fazer esta verificação:

`$ openssl` dgst -sha256 Portugal.txt

```
$ openssl
rsautl -verify -in Portugal.sig -out hash2 -inkey pk-and-sk.pem -pubin

$ diff Portugal.txt hash2
```

Tarefa 10 ***Task 10***

Procure descobrir se há forma de calcular a assinatura digital de uma forma integrada no *OpenSSL*, i.e., com um só comando.

```
$ openssl
dgst -sha256 -sign pk-and-sk.pem Portugal.txt

openssl dgst -sha256 -verify pk-pedro.pem -signature Portugal.sig Portugal.txt
```

Tarefa 11 ***Task 11***

Finalmente, considere responder às questões incluídas em baixo para melhorar o seu entendimento em relação às semelhanças, mas sobretudo às diferenças, dos conceitos de assinatura digital e código de autenticação de mensagens.

Assinatura Digital	*Message Authentication Code*
Q11.: Quem é que pode assinar determinada mensagem?	**Q12.: Quem é que pode fazer o MAC de determinada mensagem?**
☑ Só quem possui a chave privada. ☐ Só quem possui a chave pública. ☐ Qualquer entidade. ☐ Só quem possui a chave simétrica secreta.	☐ Só quem possui a chave privada. ☐ Só quem possui a chave pública. ☐ Qualquer entidade. ☑ Só quem possui a chave simétrica secreta.
Q13.: Quem é que pode verificar a assinatura de determinada mensagem?	**Q14.: Quem é que pode fazer a verificação do MAC?**
☐ Só quem possui a chave privada. ☑ Só quem possui a chave pública. ☑ Qualquer entidade. ☐ Só quem possui a chave simétrica secreta.	☐ Só quem possui a chave privada. ☐ Só quem possui a chave pública. ☐ Qualquer entidade. ☑ Só quem possui a chave simétrica secreta.
Q15.: Quais são os objectivos de uma assinatura digital?	**Q16.: Quais são os propósitos de um MAC?**
☑ Autenticação da origem da informação. ☑ Integridade. ☑ Não repúdio. ☐ Confidencialidade. ☑ Autenticidade. ☑ Dificuldade de falsificação.	☑ Autenticação da origem da informação. ☑ Integridade. ☐ Não repúdio. ☐ Confidencialidade. ☐ Autenticidade. ☐ Dificuldade de falsificação.